

NUMERICAL SIMULATION OF THERMAL FLUID–STRUCTURE INTERACTION

Vom Fachbereich Maschinenbau
an der Technischen Universität Darmstadt
zur
Erlangung des Grades eines Doktor
genehmigte

Dissertation

vorgelegt von
Plamen Pironkov, M. Sc.
aus Sofia, Bulgarien

Berichterstatter:	Prof. Dr. rer. nat. Michael Schäfer
Mitberichterstatter:	Prof. Dr. -Ing. Johannes Janicka
Tag der Einreichung:	24. November 2009
Tag der mündlichen Prüfung:	13. Januar 2010

Darmstadt 2010
D17

Acknowledgements

This thesis presents the outcome of the B2 subproject of the DFG Graduiertenkolleg "Instationäre Systemmodellierung von Flugtriebwerken" program, in which I was involved, during the last three years as a PhD student under the supervision of Prof. rer. nat. Michael Schäfer and Prof. Dr. -Ing. Johannes Janicka. The work was done in the Chair of Numerical Methods in Mechanical Engineering (FNB) at the Technische Universität Darmstadt.

I owe my deepest gratitude to my supervisor Prof. Michael Schäfer for all the discussions, ideas and the constant support. Without him this thesis would not have been completed. I would like to thank also my second advisor Prof. Johannes Janicka for including me in the project and for sharing his incredible experience with me. Thank you both for the perfect working conditions and for your help in every aspect of the work.

Next I would like to thank Monika Müller and Anna Schwarz for their help and support in all administrative problems which I have experienced during the last three years in Germany. I owe special thanks to the FNB system administrator Michael Fladerer, not only for solving all my technical problems, but also for being a good friend and driver.

During all these years I benefited a lot from the cooperation with different colleagues from FNB and GRK. I would like to express my gratitude especially to Krastan Krastev, Hongtao Sun, Christoph Ulrich Scholler, Isabell Serwas, Saim Yigit, Marcus Heck, Steffen Bauer, Friederike Gau, Florian van de Loo, Stephen Sachs, Gerrit Becker, Johannes Siegmann, Julian Michaelis, Nima Aghajari, Tomasz Waclawczyk, Astrid Walle, Frank Flitz and Yu Du. Thank you all for the good atmosphere, the good humor and the good beer which we drunk together.

Three more colleagues I owe a special gratitude. At first place I thank Dörte Sternel for the incredible enthusiasm and support and for the good words in all the ups and downs in my work. Deep gratitude to Christine Baur for being the best room colleague, good friend and frequent victim of my sense of humor. I am extremely grateful to Michael Kornhaas, who was always next to me in the right moments as a friend, engineer, system administrator, colleague, housemate and motorbike tour companion. Thank you all!

Next I would like to thank my mother, father and grandmother for the incredible support, for the right words in the right moments, for the positive energy, for everything. You are the best family in the world!

Last but not least I would also like to thank all my fantastic friends all over the world, for their smile, humor, energy, support, and the atmosphere they create. Deep gratitude to Sveta for being with me all the time and for standing my sometimes hysteric temper. Special thanks to Dr. Radostin Pachamanov, Dr. Dimiter Trenev, Dr. Grigor Grigorov, Dr. Stefan Andreev and Dr. Jordan Rizov for being a guiding light for me in my PhD. Thank you Hristo, Fam, Vanio, Ogi, Tedi, Svetlio for all these warm evenings with a lot of beer.

Darmstadt 10.2009

Contents

1	Introduction	3
1.1	State of the art	4
1.2	Scope of this work	6
1.3	Organization of the thesis	6
2	Basic concepts	9
2.1	Governing equations	9
2.1.1	Equations of structural mechanics	9
2.1.2	Equations of fluid mechanics	12
2.1.3	Heat transfer	13
2.1.4	Turbulence modeling	13
2.1.5	Fluid–structure interaction	18
2.2	Numerical methods	20
2.2.1	Finite–volume methods	20
2.2.2	Pressure–correction scheme	22
2.2.3	Multigrid method	25
2.2.4	Parallelization	26
2.2.5	Finite–element methods	27
3	Thermal fluid–structure interaction	31
3.1	Strategies for FSI	31
3.2	Strategies for handling the heat transfer	32
3.3	An implicit partitioned algorithm for TFSI	34
3.4	Interpolation of the thermal boundary flux	36
3.5	Flux correction	39
3.6	Grid movements strategies	41
3.6.1	General information and notation	41
3.6.2	Distortion of the edges	42
3.6.3	Distortion of the faces	44
3.6.4	Distortion of the blocks	46
3.7	Multigrid techniques for thermal fluid–structure interaction	47
3.8	Parallelization of the thermal fluid–structure interaction	48

4	Results and discussion	51
4.1	Laminar TFSI	51
4.2	Turbulent TFSI	56
4.3	Numerical properties of the coupling scheme	63
4.4	Accelerating the computation	66
4.4.1	T(FSI) predictor	67
4.4.2	Uneven time coupling with estimation of the distortions	69
4.4.3	Multigrid technique	73
4.4.4	Parallelization	74
4.5	LES on moving grids	75
4.5.1	Test configuration	75
4.5.2	Grid distortions	77
4.5.3	Results and discussion	78
5	Conclusion and Outlook	83

Chapter 1

Introduction

Aviation activity has grown strongly over the last twenty years. This has largely been driven by broader economic growth, increased tourism, regulatory reform, and greater industry efficiency. The aviation industry predicts that in the next 10 years the number of aircrafts will increase twice. On the other hand, nowadays communities are increasingly concerned about the impact of aircraft noise and gas emissions on the environment. For example, the European commission published a report, named “European Aeronautics: a vision for 2020”, where as a main goal for this industry a 50% reduction of the noise and the CO₂ emissions is specified. Moreover, considering the constantly rising fuel prices, another important aspect is the fuel consumption reduction. All these facts show the need for rapid development in this area.

The numerical analysis plays an important role in the development of modern jet engines. The invention of the so called super computers, together with advanced numerical methods allows the numerical modelling of different aspects of an engine, thus minimizing the costs for design and construction. Typical applications are the optimization of the rotor blades of a turbo fan, simulation of the flame in a combustion chamber, or modeling the cooling/heating of the turbine walls. In most of the cases the underlying problems have a multiphysics nature. Thus, for their reasonable description a coupled simulation should be performed.

With the increase of the computer power during the last several years the simulation of a coupled fluid–structure interaction becomes a possible task. However, investigating only the interaction between fluid forces and structural deformations in a jet engine is not sufficient. In the whole jet engine the temperature is playing an important role. Nowadays the inlet turbine temperature, for example, is over 1300°C and in 1947 it was around 800°C. This progress resulted from approved turbine materials, but above all from advanced cooling methods. Therefore developing a model for the heat transport is essential for the accurate numerical simulation of such problems.

1.1 State of the art

The numerical simulation of fluid–structure interaction problems is one of the greatest challenges in the modern scientific computing. Typical examples arise in aeronautics, where air flow around an elastic aircraft or oscillations of airfoils in air flow are computed (e.g. Dowell et al. [21]), in turbomachinery, where the transfer of energy between a rotor and a fluid is simulated (e.g. Willcox et al. [106]), in bio–mechanics, where the elastic behaviour of micro–pumps or artificial membranes in blood flow is considered (Scotti and Finol [88] or Tezduyar et al. [100]), or even in the food industry (Nitin and Karwe [63]).

In FSI analysis, the method used to satisfy the geometrical compatibility and the equilibrium conditions on the interface between the fluid and the structure is one of the key factors. For this purpose, the direct (monolytical) coupling method, the partitioned coupling method (iterative method) and the constraint method can be distinguished.

In the direct coupling scheme the fluid and the structure are treated as elements of a single dynamic system, and all of the governing equations are integrated simultaneously, and interactively in the time-domain. The main difficulty originates from the fact that the spatial domain occupied by the fluid changes in time and the location of the boundary is usually an unknown itself that depends on the fluid flow and the solution of additional equations describing the motion/deformation of the body. For the boundary formulations different methods exist, for example moving reference frames (see Li et al. [50]), coordinate transformations (see Newman and Karniadakis [61]), non-boundary conforming formulations (see Mittal and Iaccarino [58]) and embedded-boundary formulations (see Yang et al. [112] or Kim et al. [44]).

Another possibility is to enforce the fully coupled interaction constraints using the augmented-Lagrangian method. This scheme is applied by Rifai et al. [77]. They employ the finite–element method, based on the Galerkin–Least–Squares (GLS) method with discontinuity capturing operators for the fluid. The arbitrary Lagrangian–Eulerian (ALE) method is utilized to account for deformable fluid domains. The finite element treatment of solids and structures is based on a three-field variational principle.

The limited flexibility is the main drawback of both the monolytical and the constraint methods. On the other hand the partitioned time marching algorithms are based on subsequent solutions of the fluid and structure subproblems and allow one to reuse existing computational codes. Depending on how the interface conditions are implied in the coupling procedure, explicit (loosely coupled) and implicit (strongly coupled) schemes are distinguished. By a loosely coupled method (see Farhat et al. [25]) the data transfer between the solvers is done only once per time step. For stability reasons, often a fully implicit formulation has to be used (see Tallec and Mouro [97]). In this case the data exchange at every time step is repeated until a certain convergence criterion is satisfied. Although the advantages of the partitioned coupling methods make them attractive compared with the monolytical coupling methods, it is known that the convergence problem is more likely to arise because of the so called artificially

added mass effect (see Causin et al. [13]). In order to overcome this problem, some ideas to enhance the coupling scheme have been proposed. Belytschko et al. [6] introduced a moderate amount of damping in the interface element in the structure solver to include the damping effect from a fluid. Stein [93] used multi-step and multi-iterations in a structural dynamic solver. Degroote et al. [18] proposed an interface artificial compressibility method which mitigates the incompressibility constraint by adding a source term to the continuity equation in the fluid domain adjacent to the fluid-structure interface. Constant and adaptive underrelaxation was employed by Yigit [113] who also investigated the effect of the spatial discretization on the convergence of the coupling procedure.

Similar strategies for handling the convective heat transport between the fluid and the structure can be employed. Teschauer [99] defines the solid as an additional flow region and calculates the temperature in the whole domain in an analogue to the direct coupling methods. This approach is also used by different authors for a wide range of multiphysics problems, such as heating of vehicles in hypersonic flow (see Johnson et al. [42]), heating and cooling of turbine blades in jet engines (see Sondak and Dorney [91]) and thermoelastic deformation of a structure due to aerodynamic heating (see Lee et al. [47]). Haupt et al. [35] proposes an iterative solution procedure, based on the classical Dirichlet-Neumann approach, where transient problems are handled with iterative staggered schemes. Roe et al. [79] analyses the stability of a partitioned thermally coupled fluid-structure interaction scheme with a moving interface. Two types of fluid and structural discretizations are investigated finite-difference/finite-difference, as well as the more traditional finite-volume/finite-element configuration.

Validation of the different numerical procedures for thermal fluid structure interaction can be rarely found in the literature. In most of the cases a pure FSI problem is investigated. In Yamamoto et al. [109] an elastic deformable structure is used in terms of an Euler-Bernoulli beam. For the studied configuration, consisting of a cantilever flexible cylinder immersed in a current, a comparison between numerical and experimental results is shown. Lienhart et al. [51] considered a more general case, employing an elastic, free vibrating structure with a full resolution of the structural problem. The temporal courses of front body angles, trailing edge coordinates, membrane deformations, and the flow field properties are compared with an experiment. In both situations temperature effects are not taken into consideration and the fluid flow is laminar.

On the other hand the heat transfer characteristics under single or multiple impinging turbulent jet flows are investigated numerically and experimentally by different authors during the last 30 years, without taking into account the effects of the fluid-structure interaction. Polat and Douglas [73] studied the numerical simulation of a turbulent flow field of impinging jets over a surface. Nitin and Karwe [63] used a numerical simulation to predict flow and thermal fields and calculate the surface heat-transfer coefficient for a single hot air jet impinging on a cookie-like product. The results from the numerical predictions are compared with experimental measurements. Craft et al. [16] applied different turbulence models to the numerical prediction of the turbulent imping-

ing jets discharged from a circular pipe and discussed the characteristics of the numerical approaches.

1.2 Scope of this work

The aim of this work is to develop and implement a framework for numerical simulation of fully coupled thermal fluid structure interaction. For the mechanical fluid–structure interaction coupling an implicit partitioned approach which has been already implemented and investigated in the Chair of Numerical Methods in Mechanical Engineering (FNB), is chosen. The scheme allows to use specifically designed and highly optimized codes on different fields and offers significant benefits in terms of efficiency. Furthermore it can be easily applied to various FSI problems which is a crucial aim of this work. For the flow field the in–house program FASTEST is used. The open–source finite–element solver FEAP is employed for the solid domain. The interpolation of the property values between the numerical grids is done by the quasi–standard coupling interface MpCCI.

Since in most of the practical situations the flow of interest is in the turbulent regime, special attention should be paid on the different turbulence models. Low and high Reynolds averaged Navier–Stokes models, as well as large eddy simulation models seems to be the natural choice. Advanced techniques for the grid distortions, especially in the near wall regions, are a crucial point for such FSI calculation. Different approaches for the mesh movement are implemented and investigated with respect to grid quality and efficiency.

The thermal fluid–structure interaction problems are highly nonlinear problems, thus a great amount of computational time is needed for their numerical simulation. Optimization, in terms of computational time and accuracy, of the coupled procedure is an important goal of this work. Special attention is paid to the parallelization of the whole framework.

Validation and verification of the coupling procedure is also an important aspect. Analytical test cases, as well as representative test configurations for which experimental data exist, are used. The numerical properties of the scheme are studied.

1.3 Organization of the thesis

In chapter 2 first the basic governing equations of the continuum mechanics are presented, together with different material models for both the structure and the fluid. The scalar transport equation which is used for the simulation of heat transfer is shown afterwards. The transformation of the Navier–Stokes equations for fluid flow in a turbulent regime are demonstrated. The low and high Reynolds RANS methods, as well as the Smagorinsky and its dynamic variant LES models are discussed. The first part of the chapter ends with the changes in the equations imposed by the arbitrary Lagrangian–Euler (ALE) de-

scription. In the second part of chapter 2 the numerical methods used in this work are introduced. The finite-volume method for solving the equations of the fluid dynamics are followed by the SIMPLE pressure-correction algorithm, for coupling of the velocity and the pressure. The principles of the geometrical multigrid method on moving grids for accelerating the computation, as well as techniques for parallelizing the calculation are then presented. At the end the finite-elements method which were used for solving the equations of the structural mechanics, are shown. Chapter 3 starts with a short overview of the existing strategies for thermal fluid-structure interaction followed by a description of the developed coupling algorithm. Special attention is paid to the grid distortion concept and to the flux correction scheme. Results for the validation and verification of the numerical procedure for laminar and turbulent fluid flow are presented in chapter 4. The numerical properties of the coupled scheme are then investigated. Based on this study different acceleration methods are proposed. At the end of the chapter results for applying a dynamic LES model on a moving grid framework are shown. In chapter 5 conclusions and outlook are given.

Chapter 2

Basic concepts

The mathematical fundamentals of this work are presented in this chapter. First the basic model equations of continuum mechanics are shown. As, in practical relevant cases, no analytical solution of those equations can be found, a numerical approach is needed. Therefore different numerical methods and their characteristics are discussed in the second part of the chapter.

2.1 Governing equations

Here the basic governing equations of the continuum mechanics are presented. The equations are derived from the fundamental conservation laws of mass, momentum, momentum of momentum and energy, together with problem specific material laws. The interested reader can find a detailed derivation of the equations in [55]. Based on these equations, several possibilities of modeling turbulence are then shown. At the end of the section the changes in these equations, needed for modeling a fluid–structured interaction are presented and discussed.

2.1.1 Equations of structural mechanics

General task of the structural mechanics is the computation of deformations, deflections, and internal forces or stresses within structures. Normally one distinguishes between linear and non–linear models, where the non–linearity can be of geometrical and/or physical nature. Geometrically linear problems are characterized by the linear strain–displacement relations, whereas physically linear problems are based on material models involving a linear relation between strains and stresses. In the following, the equations for two linear and one non–linear models are outlined. In the whole work we use the Einstein summation convention of summing on repeated indices.

Linear models

In this work only the linear elasticity and the linear thermo-elasticity models are used. The theory of linear elasticity is a physically and geometrically linear one and the equations governing such boundary value problems are based on three tensor equations (see [24]):

- Equation of motion (an expression of Newton's second law)

$$\frac{\partial \sigma_{ij}^s}{\partial x_j} + \rho_s f_i = \rho_s \frac{D^2 u_i}{Dt^2} \quad (2.1)$$

- Strain-displacement relation

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.2)$$

- Constitutive equation (Hooke's law)

$$\sigma_{ij}^s = C_{ijkl} \varepsilon_{kl} + \varepsilon_{ij}^{th}, \quad (2.3)$$

where x_i , $i = 1, 2, 3$ are the components of the position vector \mathbf{x} , t is the time, $\sigma_{ij}^s = \sigma_{ji}^s$ is the Cauchy stress tensor, f_i are the body forces, ρ_s is the mass density, u_i is the displacement vector, $\varepsilon_{ij} = \varepsilon_{ji}$ is the strain, ε^{th} is the thermal part of the strain tensor and C_{ijkl} is the elasticity tensor.

For isotropic materials the Cauchy stress tensor, as well as the infinitesimal strain tensor are symmetric and they can be expressed as six-dimensional vectors. In this case $C_{ijkl} = C_{jikl}$ and $C_{ijkl} = C_{ijlk}$. The model is defined by two independent elastic parameters for C and two parameters for ε^{th} . Here we take the Young's modulus E and the Poisson's ratio ν for the elastic parameters. The elasticity tensor can be written as:

$$C = \begin{bmatrix} \frac{1}{E} & -\frac{\nu}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\ -\frac{\nu}{E} & \frac{1}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\ -\frac{\nu}{E} & -\frac{\nu}{E} & \frac{1}{E} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G} \end{bmatrix} \quad (2.4)$$

with the shear modulus G related through

$$G = \frac{E}{2(1 + \nu)}. \quad (2.5)$$

The thermal strain is given by

$$\varepsilon^{th} = \begin{bmatrix} \alpha \\ \alpha \\ \alpha \\ 0 \\ 0 \\ 0 \end{bmatrix} \Delta T, \quad (2.6)$$

where α is the coefficient of linear thermal expansion and $\Delta T = T - T_0$, where T_0 is the temperature where the thermal strains vanish and T is the current value of the temperature.

Finite deformations

In the theory of finite deformations usually the Lagrangian formulation is used. This means that an observer is located in a fixed position in space and the deformation is given by a function χ :

$$x_i = \chi(X_j, t), \quad (2.7)$$

where X_j are the components of the position vector \mathbf{X} of a particle in the reference (undeformed) configuration and the current position of this particle is given by the spatial coordinates x_i (See Fig. 2.1).

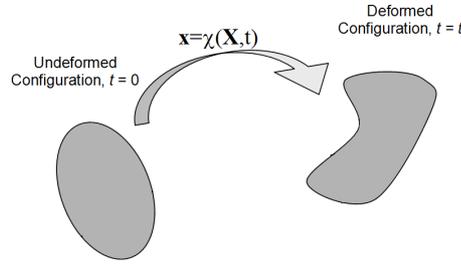


Figure 2.1: Deformation of continuum body

The actual displacements in this case can be written as:

$$u_i = x_i - X_i = \chi(X_j, t) - X_i. \quad (2.8)$$

Other important kinematic measure is the material deformation gradient tensor F_{ij} which characterizes the local deformation at a material point with position vector \mathbf{X} , i.e. deformation at neighboring points, by transforming (linear transformation) a material line element emanating from that point from the reference configuration to the current or deformed configuration:

$$F_{ij} = \frac{\partial x_i}{\partial X_j} = \frac{\partial \chi(X_i, t)}{\partial X_j}, \quad (2.9)$$

as well as the Green–Lagrange strain tensor:

$$E_{ij} = \frac{1}{2}(F_{ki}f_{kj} - \delta_{ij}), \quad (2.10)$$

with δ_{ij} the delta Kronecker symbol, defined as

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}. \quad (2.11)$$

The equation of motion can be written using the 2nd Piola–Kirchhoff stress tensor S_{ij} in the following manner:

$$\rho_s \frac{\partial^2 \chi(X_j, t)}{\partial t^2} = \frac{\partial S_{ji} F_{ij}}{\partial X_j} + \rho_s f_i. \quad (2.12)$$

In order to close the system of equations a material model is needed. In the case of finite deformations several models exist like Fung model, Neo–Hooke model, Mooney–Rivlin model, etc. More information can be found in Eschenauer et al. [24]. In this work we use the simplest hyper-elastic model, the Saint Venant–Kirchhoff model which is just an extension of the linear elastic material model to the nonlinear regime. For the 2nd Piola–Kirchhoff stress tensor the model states

$$S_{ij} = \lambda E_{kk} \delta_{ij} + 2\mu_s E_{ij}, \quad (2.13)$$

where the λ and μ_s are the Lamé parameters which can be expressed in terms of Young’s modulus and Poisson’s ratio as

$$E = \frac{\mu_s(3\lambda + 2\mu_s)}{\lambda + \mu_s} \quad \text{and} \quad \nu = \frac{\lambda}{2(\lambda + \mu_s)}. \quad (2.14)$$

The last step is prescribing proper boundary conditions which typically are of Dirichlet (prescribed deformation) or Neumann (prescribed stress) type.

2.1.2 Equations of fluid mechanics

The fluid mechanics characterize the flow behavior of liquids and gasses. In this work we concentrate only on one special class of problems – linear viscous isotropic fluids, known as Newtonian fluids which are characterized by the following material (Newton’s) law for the Cauchy stress tensor σ_{ij}^f

$$\sigma_{ij}^f = \mu_f \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} - \frac{2}{3} \frac{\partial v_k}{\partial x_k} \delta_{ij} \right) - p \delta_{ij}, \quad (2.15)$$

where v is the velocity vector with components v_i , $i = 1, 2, 3$, μ_f is the dynamic fluid viscosity and p is the pressure.

Moreover we restrict ourselves to the incompressible case which means that the divergence of the velocity is zero and the Cauchy stress tensor in that case transforms to

$$\sigma_{ij}^f = \mu_f \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) - p \delta_{ij}. \quad (2.16)$$

From the conservation laws of mass and momentum one can derive the so called Navier–Stokes equations for incompressible Newtonian flow which describe the fluid motion:

$$\frac{\partial(v_i)}{\partial x_i} = 0, \quad (2.17)$$

$$\frac{\partial(\rho_f v_i)}{\partial t} + \frac{\partial(\rho_f v_i v_j)}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\mu_f \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right] - \frac{\partial p}{\partial x_i} + \rho_f f_i, \quad (2.18)$$

with ρ_f the density of the fluid.

The Navier–Stokes equations are nonlinear partial differential equations in almost every real situation. Due to this fact, most problems are difficult or impossible to solve analytically. The nonlinearity of the equations is also the main contributor to the turbulence. Several ways of handling this phenomenon will be discussed in section 2.1.4. For detailed derivation of the Navier–Stokes equations and discussion on their properties see Markov [55] or Zaprianov [115].

2.1.3 Heat transfer

An important class of problems in the mechanical engineering represent problems in which heat transfer occurs. A general definition of heat transfer can be formulated as: the energy in transit due to a temperature difference. Classical transfer of thermal energy takes place only through conduction, convection, radiation or any combination of these. When a temperature gradient exists in a stationary medium which may be solid or fluid, the term conduction is used to refer to the heat transfer that will occur across the medium due to atomic and molecular activity. The law of heat conduction, also known as Fourier’s law, states that the time rate of heat transfer through a material is proportional to the negative gradient of the temperature:

$$q_i = -\kappa \frac{\partial T}{\partial x_i}, \quad (2.19)$$

where q is the heat flux, T the temperature and κ the heat conductivity of the material.

While the conduction represents the transfer of energy by vibrations at a molecular level, the convection is a mechanism of heat transfer occurring because of bulk motion of fluids and cannot be observed in solids. The last mode of heat transfer is the radiation which is the transfer of heat through electromagnetic radiation. In this work radiation will be not considered.

Using the Fourier’s (Eq. 2.19) and the energy conservation laws one can formulate the so called Convection–Diffusion equation for the temperature T :

$$\frac{\partial \rho c_p T}{\partial t} + \frac{\partial}{\partial x_i} \left(\rho c_p v_i T - \kappa \frac{\partial T}{\partial x_i} \right) = \rho Q, \quad (2.20)$$

with thermal source Q , specific heat c_p and density ρ . For solids and stationary medium the convective part disappears and the transport equation (Eq. 2.20) transforms to:

$$\frac{\partial \rho c_p T}{\partial t} - \frac{\partial}{\partial x_i} \left(\kappa \frac{\partial T}{\partial x_i} \right) = \rho Q \quad (2.21)$$

2.1.4 Turbulence modeling

Presenting the equations of the fluid dynamics, we mentioned the term turbulence. Turbulent flow is a fluid regime characterized by chaotic, stochastic

property changes. This includes low momentum diffusion, high momentum convection, and rapid variation of pressure and velocity in space and time. Flow that is not turbulent is called laminar. The Reynolds number Re is a dimensionless number defined as $Re = (U \cdot L)/\nu_f$, with characteristic (mean) velocity U , characteristic length L and kinematic viscosity ν_f . It gives a measure of the ratio of inertial forces to viscous forces and is used to characterize different flow regimes. Turbulent flow occurs at high Reynolds numbers (the exact value is problem dependent, see Pope [74]), where the inertial forces are dominant.

Turbulence causes the formation of eddies of many different length scales. Most of the kinetic energy of the turbulent motion is contained in the large scale structures. The energy "cascades" from these large scale structures to smaller scale structures by an inertial and essentially inviscid mechanism. This process continues, creating smaller and smaller structures which produces a hierarchy of eddies. Eventually this process creates structures that are small enough that molecular diffusion becomes important and viscous dissipation of energy finally takes place. The scale at which this happens is the Kolmogorov length scale named after the Soviet Russian mathematician A.N. Kolmogorov, whose work (see [45]) had a significant influence on this field.

Because of their chaotic characteristics, turbulent flows are hard to simulate. The whole range of spatial and temporal scales of the turbulence must be resolved and that's why extremely fine space and time discretization is needed. If properly resolved, the direct numerical solution of the Navier–Stokes equations (DNS) solution represents a real picture of turbulence containing all scales of motion, from the largest scales to the smallest dissipative scales. Since DNS resolves all sizes of the motion, the computational cost increases with the Reynolds number enormously, roughly with Re^3 , thus restricting it to low-to-moderate Reynolds number in relatively simple geometries. In contrast, in a turbulence model, equations are solved for some mean quantities. Such models are based on the Reynolds–averaged Navier–Stokes equations (RANS) and the Large–eddy simulation (LES).

Reynolds–averaged Navier–Stokes equations

The idea behind the Reynolds–averaged Navier–Stokes equations is to consider the velocity and the pressure fields as a random variables and to present them as sum of mean value $\langle * \rangle$ and fluctuation $*'$ (Reynolds decomposition):

$$v_i(x_j, t) = \langle v_i(x_j, t) \rangle + v'_i(x_j, t), \quad (2.22)$$

$$p(x_j, t) = \langle p(x_j, t) \rangle + p'(x_j, t), \quad (2.23)$$

$$T(x_j, t) = \langle T(x_j, t) \rangle + T'(x_j, t). \quad (2.24)$$

For steady flow the mean value, for instance for the pressure p , is defined by

$$\langle p(x_j) \rangle = \lim_{I_t \rightarrow \infty} \frac{1}{I_t} \int_0^{I_t} p(x_j, t) dt, \quad (2.25)$$

where t is the time and I_t is the averaging interval which should be larger than the time scale of the fluctuations. In case of unsteady flow ensemble averaging

should be used, i.e.

$$\langle p(x_j, t) \rangle = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N p(x_j, t), \quad (2.26)$$

where N is the number of tests and it must be large enough to eliminate the effects of the fluctuations.

Taking the mean of the continuity (Eq. 2.17), momentum (Eq. 2.18) and scalar transport (Eq. 2.20) we have:

$$\frac{\partial \rho \langle v_i \rangle}{\partial x_i} = 0 \quad (2.27)$$

$$\begin{aligned} \rho_f \frac{\partial \langle v_i \rangle}{\partial t} + \rho_f \langle v_j \rangle \frac{\partial \langle v_i \rangle}{\partial x_j} &= \mu_f \frac{\partial}{\partial x_j} \left[\left(\frac{\partial \langle v_i \rangle}{\partial x_j} + \frac{\partial \langle v_j \rangle}{\partial x_i} \right) \right] - \rho_f \frac{\partial \langle v'_i v'_j \rangle}{\partial x_j} \\ &\quad - \frac{\partial \langle p \rangle}{\partial x_i} + \rho_f f_i \end{aligned} \quad (2.28)$$

$$\frac{\partial \rho c_p \langle T \rangle}{\partial t} + \rho c_p \frac{\partial}{\partial x_i} (\langle v_i \rangle \langle T \rangle + \langle v'_i T' \rangle) = \frac{\partial}{\partial x_i} \left(\kappa \frac{\partial \langle T \rangle}{\partial x_i} \right) + \rho Q, \quad (2.29)$$

which are the same equations as the Navier–Stokes equations, except for the terms $\langle v'_i v'_j \rangle$ and $\langle v'_i T' \rangle$, called Reynolds stress and turbulent thermal stress respectively (for the derivation of the equations see Markov [55]). The presence of the later means that the equations are not closed and in order to do that we must introduce a turbulence model. A widely used model is the k – ε model which lays on the approximation, introduced by Boussinesq that the deviatoric Reynolds stress ($-\rho_f \langle v'_i v'_j \rangle + \frac{2}{3} \rho k \delta_{ij}$) is proportional to the mean rate of strain:

$$-\rho_f \langle v'_i v'_j \rangle + \frac{2}{3} \rho k \delta_{ij} = \rho \mu_T \left(\frac{\partial \langle v_i \rangle}{\partial x_j} + \frac{\partial \langle v_j \rangle}{\partial x_i} \right), \quad (2.30)$$

where the scalar coefficient μ_T is the turbulent viscosity and k is the turbulent kinetic energy ($k \equiv \frac{1}{2} \langle v'_i v'_i \rangle$). The equation (Eq. 2.30) is also known as turbulent–viscosity hypothesis. For the thermal stress the eddy–diffusion model is used:

$$-\rho_f \langle v'_i T' \rangle = \Gamma_T \frac{\partial \langle T \rangle}{\partial x_i}, \quad (2.31)$$

with Γ_T the turbulent diffusivity. The turbulent Prandtl number Pr_T can be used to relate Γ_T and μ_T , i.e. $\Gamma_T = \mu_T / Pr_T$.

The k – ε model belongs to the class of two–equation models. In addition to the both mentioned hypotheses it consists of model transport equation for two turbulence quantities – the turbulent kinetic energy k :

$$\begin{aligned} \rho_f \frac{\partial k}{\partial t} + \rho_f \langle v_j \rangle \frac{\partial k}{\partial x_j} &= \left[\mu_T \left(\frac{\partial \langle v_i \rangle}{\partial x_j} + \frac{\partial \langle v_j \rangle}{\partial x_i} \right) - \frac{2}{3} \rho_f k \delta_{ij} \right] \frac{\partial \langle v_i \rangle}{\partial x_j} \\ &\quad - \rho_f \varepsilon + \frac{\partial}{\partial x_j} \left[\left(\mu_f + \frac{\mu_T}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] \end{aligned} \quad (2.32)$$

and the dissipation ε :

$$\begin{aligned} \rho_f \frac{\partial \varepsilon}{\partial t} + \rho_f \langle v_j \rangle \frac{\partial \varepsilon}{\partial x_j} &= C_{\varepsilon 1} \frac{\varepsilon}{k} \left[\mu_T \left(\frac{\partial \langle v_i \rangle}{\partial x_j} + \frac{\partial \langle v_j \rangle}{\partial x_i} \right) - \frac{2}{3} \rho_f k \delta_{ij} \right] \frac{\partial \langle v_i \rangle}{\partial x_j} \\ &\quad - C_{\varepsilon 2} \rho_f \frac{\varepsilon^2}{k} + \frac{\partial}{\partial x_j} \left[\left(\mu_f + \frac{\mu_T}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_j} \right] \end{aligned} \quad (2.33)$$

and specification of the turbulent viscosity as:

$$\mu_T = \frac{C_\mu k^2}{\varepsilon}, \quad (2.34)$$

with the following model parameters:

$$C_\mu = 0.09, \quad C_{\varepsilon 1} = 1.44, \quad C_{\varepsilon 2} = 1.92, \quad \sigma_k = 1.0, \quad \sigma_\varepsilon = 1.3. \quad (2.35)$$

The equations (Eq. 2.27 – 2.35) form a closed system of equations for the unknowns $\langle v_i \rangle$, $\langle p \rangle$, k and ε which is ready to be numerically solved.

The presented k - ε model is not valid in the near wall regions. In this work two different strategies were used to overcome this problem – the wall functions model proposed by Launder [46], and the low Reynolds model proposed by Chien [14], where the damping functions f_1 , f_2 and f_3 are applied in the equation for the dissipation:

$$\begin{aligned} \rho_f \frac{\partial \bar{\varepsilon}}{\partial t} + \rho_f \langle v_j \rangle \frac{\partial \bar{\varepsilon}}{\partial x_j} &= \\ C_{\varepsilon 1} f_1 \frac{\bar{\varepsilon}}{k} \left[\mu_T \left(\frac{\partial \langle v_i \rangle}{\partial x_j} + \frac{\partial \langle v_j \rangle}{\partial x_i} \right) - \frac{2}{3} \rho_f k \delta_{ij} \right] \frac{\partial \langle v_i \rangle}{\partial x_j} \\ &\quad - C_{\varepsilon 2} f_2 \rho_f \frac{\bar{\varepsilon}^2}{k} + \rho_f f_3 + \frac{\partial}{\partial x_j} \left[\left(\mu_f + \frac{\mu_T}{\sigma_\varepsilon} \right) \frac{\partial \bar{\varepsilon}}{\partial x_j} \right], \end{aligned} \quad (2.36)$$

where $\varepsilon = \varepsilon_0 + \bar{\varepsilon}$ and $\varepsilon_0 = 2\mu_f \frac{k}{\delta^2}$. The model constants in that case are set as follows:

$$C_\mu = 0.09, \quad C_{\varepsilon 1} = 1.35, \quad C_{\varepsilon 2} = 1.80, \quad \sigma_k = 1.0, \quad \sigma_\varepsilon = 1.3. \quad (2.37)$$

Large-eddy simulation

The RANS models need much less computational efforts than the DNS, but it provides only averaged results. Another approach for simulating a turbulent flow which requires more effort than the RANS methods, but is able to predict instantaneous flow characteristics and resolve turbulent flow structures, is the large-eddy simulation (LES).

In large-eddy simulation, the larger three-dimensional unsteady turbulent motions are directly represented, whereas the effects of the smaller scale motions are modelled. For this purpose a filtering operation is defined to decompose the velocity into sum of a filtered and residual components:

$$\bar{v}_i(x, t) = \int G(x, x') v_i(x', t) dx', \quad (2.38)$$

where $G(x, x')$ is the filter kernel which is a localized function.

Filtering the Navier–Stokes equations (Eq. 2.17 – 2.18), one obtains a set of equations very similar in form to the RANS equations:

$$\frac{\partial \rho \bar{v}_i}{\partial x_i} = 0 \quad (2.39)$$

$$\rho_f \frac{\partial (\bar{v}_i)}{\partial t} + \frac{\partial \rho_f \bar{v}_i \bar{v}_j}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\mu_f \left(\frac{\partial \bar{v}_i}{\partial x_j} + \frac{\partial \bar{v}_j}{\partial x_i} \right) \right] - \frac{\partial \bar{p}}{\partial x_i} + \rho_f f_i, \quad (2.40)$$

where \bar{p} is the filtered pressure field. This equations differ from the Navier–Stokes equations because the filtered product $\bar{v}_i \bar{v}_j$ is not equal to the product of the filtered velocities $\bar{v}_i \bar{v}_j$. The difference is the subgrid scale Reynolds stress:

$$\tau_{ij}^s = \overline{v_i v_j} - \bar{v}_i \bar{v}_j, \quad (2.41)$$

which have to be modeled in order to close the equations. In this work only the simplest model, proposed by Smagorinski (1963) and its dynamic variant are used. The Smagorinsky model is an eddy viscosity model. It relates the anisotropic residual stress tensor with the filtered rate of strain:

$$\tau_{ij}^s - \frac{1}{3} \tau_{kk}^s \delta_{ij} = -\mu_t \left(\frac{\partial \bar{v}_i}{\partial x_j} + \frac{\partial \bar{v}_j}{\partial x_i} \right) = -2\mu_t \bar{S}_{ij} \quad (2.42)$$

The form of the subgrid scale eddy viscosity μ_t can be derived by dimensionless argument and is:

$$\mu_t = C_S^2 \rho \Delta^2 |\bar{S}|, \quad (2.43)$$

where $|\bar{S}| = (2\bar{S}_{ij}\bar{S}_{ij})^{1/2}$, C_S is the model parameter and $\Delta = (\Delta_1\Delta_2\Delta_3)^{1/3}$ is the filter length, with Δ_i the filter width in spatial direction x_i .

The dynamic model proposed by Germano et al. [31] is based on the idea that the smallest resolved scale motions can provide information that can be used to model the largest subgrid scale motions. The Smagorinsky constant in this case is determined dynamically from the results of the LES. Thus we can define it as a function of space and time $C_S = C_S(x, t)$. The procedure starts with filtering the velocity field \bar{v}_i by a filter broader than one used in the LES itself. A standard choice for its width is $\hat{\Delta} = 2\Delta$. Analogously as before, we define the subtest scale stresses τ_{ij}^t :

$$\tau_{ij}^t = \widehat{\overline{v_i v_j}} - \widehat{\bar{v}_i \bar{v}_j} \quad (2.44)$$

and approximate it using the Smagorinsky model as

$$\tau_{ij}^t - \frac{1}{3} \tau_{kk}^t \delta_{ij} = -2C_S^2 \rho \hat{\Delta}^2 |\widehat{\bar{S}}| \widehat{\bar{S}}_{ij}. \quad (2.45)$$

The two subgrid scale stress terms are related by the Germano identity

$$\mathcal{L}_{ij} = \tau_{ij}^t - \widehat{\tau_{ij}^s}, \quad (2.46)$$

where

$$\mathcal{L}_{ij} = \widehat{\overline{v_i v_j}} - \widehat{\overline{v_i}} \widehat{\overline{v_j}} \quad (2.47)$$

is the resolved turbulent stress. Inserting (2.42) and (2.45) in the Germano identity and employing the approximation

$$\widehat{C_S^2 A} = C_S^2 \widehat{A}, \quad (2.48)$$

where $A = \rho \Delta^2 |\overline{S}| \overline{S}_{ij}$, yields

$$\mathcal{L}_{ij} = 2C_S^2 M_{ij}, \quad (2.49)$$

where M_{ij} is defined by

$$M_{ij} = \widehat{A} - \rho \widehat{\Delta^2 |\overline{S}| \overline{S}_{ij}}. \quad (2.50)$$

For solving the system of equations (2.50) for the parameter $C_g = C_S^2$ the least-squares method is used, as proposed by Lilly [52]. As C_g may take also negative values a simple clipping is used, i.e. $C_g(x, t) = \max\{C_g(x, t), 0\}$.

2.1.5 Fluid–structure interaction

We consider a problem domain Ω consisting of a fluid part Ω_f and a solid part Ω_s which regarding the shape as well as the location of fluid and solid parts can be arbitrary. As the Lagrangian description of motion is used for Ω_s , we can apply the different models (e.g. linear–elasticity model, Saint Venant–Kirchhof, etc. see Section 2.1.1) without any changes.

The arbitrary Lagrangian–Euler (ALE) description of motion is used in the fluid domain. The deformation of Ω_f , caused by the deformation of the solid, is taken into account by the use of the velocity v_i^g , with which the surface of a control volume moves (more information can be found in Donea et al. [20]). The Navier–Stokes equations 2.17 and 2.18, together with the transport equation 2.20 then transform to:

$$\frac{\partial v_i^*}{\partial x_i} = 0 \quad (2.51)$$

$$\rho_f \frac{\partial v_i}{\partial t} + \rho_f v_j \frac{\partial v_i^*}{\partial x_j} = \frac{\partial}{\partial x_j} \sigma_{ij}^f + \rho_f f_i \quad (2.52)$$

$$\rho c_p \frac{\partial T}{\partial t} + \rho c_p \frac{\partial}{\partial x_i} v_i^* T = \frac{\partial}{\partial x_i} \left(\kappa_f \frac{\partial T}{\partial x_i} \right) + \rho q \quad (2.53)$$

where $v_i^* = v_i - v_i^g$.

The same modification is needed for the RANS equations. For the low-Reynolds model, for example, we get:

$$\begin{aligned} \rho_f \frac{\partial \langle v_i \rangle}{\partial t} + \rho_f \langle v_j \rangle \frac{\partial \langle v_i^* \rangle}{\partial x_j} = & \mu_f \frac{\partial}{\partial x_j} \left[\left(\frac{\partial \langle v_i \rangle}{\partial x_j} + \frac{\partial \langle v_j \rangle}{\partial x_i} \right) \right] \\ & - \rho_f \frac{\partial \langle v_i' v_j' \rangle}{\partial x_j} - \frac{\partial \langle p \rangle}{\partial x_i} + \rho_f f_i, \end{aligned} \quad (2.54)$$

$$\frac{\partial \rho_f c_p \langle T \rangle}{\partial t} + \rho_f c_p \frac{\partial}{\partial x_i} (\langle v_i^* \rangle \langle T \rangle + \langle v_i^* T' \rangle) = \frac{\partial}{\partial x_i} \left(\kappa \frac{\partial \langle T \rangle}{\partial x_i} \right) + \rho_f q. \quad (2.55)$$

$$\begin{aligned} \rho_f \frac{\partial k}{\partial t} + \rho_f \langle v_j^* \rangle \frac{\partial k}{\partial x_j} &= \left[\mu_T \left(\frac{\partial \langle v_i \rangle}{\partial x_j} + \frac{\partial \langle v_j \rangle}{\partial x_i} \right) - \frac{2}{3} \rho_f k \delta_{ij} \right] \frac{\partial \langle v_i \rangle}{\partial x_j} \\ &\quad - \rho_f \varepsilon + \frac{\partial}{\partial x_j} \left[\left(\mu_f + \frac{\mu_T}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] \end{aligned} \quad (2.56)$$

$$\begin{aligned} \rho_f \frac{\partial \varepsilon}{\partial t} + \rho_f \langle v_j^* \rangle \frac{\partial \varepsilon}{\partial x_j} &= C_{\varepsilon 1} f_1 \frac{\varepsilon}{k} \left[\mu_T \left(\frac{\partial \langle v_i \rangle}{\partial x_j} + \frac{\partial \langle v_j \rangle}{\partial x_i} \right) - \frac{2}{3} \rho_f k \delta_{ij} \right] \frac{\partial \langle v_i \rangle}{\partial x_j} \\ &\quad - C_{\varepsilon 2} f_2 \rho_f \frac{\varepsilon^2}{k} + \rho_f f_3 + \frac{\partial}{\partial x_j} \left[\left(\mu_f + \frac{\mu_T}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_j} \right]. \end{aligned} \quad (2.57)$$

The same additional terms appear exactly in the same manner in both, the equations of the high-Reynolds models and the LES.

The problem formulation has to be closed by prescribing suitable boundary and interface conditions. On solid and fluid boundaries Γ_s and Γ_f standard conditions as for individual solid and fluid problems can be prescribed. For the velocities and the stresses on a fluid-solid interface Γ_i we have the conditions:

$$\langle v_i \rangle = u_i^b \quad \text{and} \quad \sigma_{ij}^s = \langle \sigma_{ij}^f \rangle n_j, \quad (2.58)$$

where u_i^b is the interface velocity. In addition, the temperatures as well as the heat fluxes have to be identical on Γ_i . How these conditions are implied in the numerical scheme is discussed later in this work.

2.2 Numerical methods

The partial differential equations which were presented in the last chapter can not be solved analytically in most practical cases. In the last decades, with the appearing of the computers, the numerical methods for solving PDEs received a great attention. A lot of different methods were developed and investigated by different authors. The most widely used are the finite-difference methods, where the derivatives are approximated using a finite-difference formula, the spectral and pseudo-spectral methods which represents functions as a sum of particular basis functions, for example using Fourier or Chebyshev series, the finite-element methods, and the finite-volume methods. In this work the last two were used and they will be presented in this chapter. First the finite-volume method for solving the equations of the fluid dynamics will be introduced. Then the SIMPLE pressure-correction algorithm, for coupling of the velocity and pressure will be discussed, together with the geometrical multigrid method for accelerating the computation. At the end the finite-elements method which were used for solving the equations of the structural mechanics, will be described.

2.2.1 Finite-volume methods

For illustrating the finite-volume methods we will use the heat transport equation (Eq. 2.20). Here we assume that the velocity field is known. How the non-linear Navier-Stokes equations are solved and the velocity and the pressure fields are found we show in the next chapter. For more detailed information on the methods see Schäfer [84] or Ferziger and Perić [28].

We discretize the solution domain into a finite number of control volumes, as for the sake of simplicity we suppose that the grid is Cartesian. A typical control volume, together with the notation which is used is shown in Figure 2.2. The center of the CV is marked with \mathbf{P} and the neighbour CV's centers from top, bottom, east, west, south and north with \mathbf{T} , \mathbf{B} , \mathbf{E} , \mathbf{W} , \mathbf{S} and \mathbf{N} , respectively. With low-case letters the centers of the plane faces are denoted.

Integrating (Eq. 2.20) over all control volumes and applying the Gauss integral formula gives the equation:

$$\int_V \rho_f c_p \frac{\partial T}{\partial t} dV + \int_S \rho_f c_p T (v_i n_i) dS = \int_S \kappa \frac{\partial T}{\partial x_i} n_i dS + \int_V \rho_f Q dV, \quad (2.59)$$

where n_i are the components of the normal to the surface vector. For approximation of the surface integrals we first present them as sum of integrals over the six CV faces:

$$\int_S f dS = \sum_k \int_{S_k} f dS, \quad k = e, w, n, s, t, b, \quad (2.60)$$

where f is the component of the diffusive or convective flux vector. In what follows, only the east face will be considered. Analogous expressions for all faces can be derived straightforward.

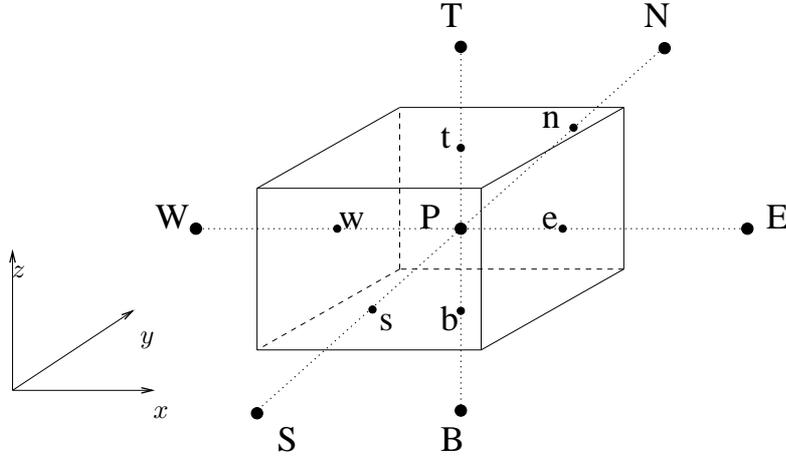


Figure 2.2: A typical CV and the notation used for a Cartesian 3D grid

Applying the midpoint rule for the surface integrals we get:

$$\int_{S_e} f dS_e \approx f_e S_e, \quad (2.61)$$

where f_e is the value of the function in the cell face center, and S_e is the area of the east face. Analogously the volume integrals are replaced by the product of the value of the integrand at the CV center f_P and the volume of the CV δV :

$$\int_V f dV \approx f_P \delta V. \quad (2.62)$$

The next step is approximation of the cell–face values in terms of CV center values. In this work three interpolation techniques are used – the upwind differencing scheme (UDS) (Eq. 2.63), the linear central difference scheme (CDS) Eq. 2.64, and the multi–dimensional linear interpolation (MULI) (Eq. 2.65), which unlike the first two preserves its second order even for noncartesian grids. The derivation of the formulas and the values of the coefficients λ_E , λ_{NS} and λ_{TB} can be found in Lehnhäuser [49].

$$f_e^{UDS} \approx \begin{cases} f_P & \text{if } (v_i n_i)_e > 0 \\ f_E & \text{if } (v_i n_i)_e < 0 \end{cases} \quad (2.63)$$

$$f_e^{CDS} \approx f_E \lambda_E + f_P (1 - \lambda_E) \quad (2.64)$$

$$f_e^{MULI} \approx \lambda_E f_E + (1 - \lambda_E) f_P + \lambda_{NS} (f_N - f_S) + \lambda_{TB} (f_T - f_B). \quad (2.65)$$

The simplest approximation of the gradients which is needed for the evaluation of the diffusive fluxes, is the linear interpolation:

$$\left(\frac{\partial f}{\partial x_i} \right)_e \approx \frac{f_E - f_P}{x_E - x_P}, \quad (2.66)$$

which for distorted grids is only a first order method. In contrast the Taylor-based interpolation (TBI) (see [48]) is always second order:

$$\begin{aligned} \left(\frac{\partial f}{\partial x_i} \right)_e &\approx \gamma_1 f_E + \gamma_2 f_P + \gamma_3 f_N + \gamma_4 f_S + \gamma_5 f_T + \gamma_6 f_B \\ &\quad + \gamma_7 f_{NE} + \gamma_8 f_{SE} + \gamma_9 f_{TE} + \gamma_{10} f_{BE}, \end{aligned} \quad (2.67)$$

where γ_1 – γ_{10} are interpolation factors.

The last step is the time discretization in Eq. 2.59. Here different strategies can be applied, as Runge–Kutta methods, Euler methods, etc (see [84]). In this work we use a three–point backward method which use the information from the previous two time steps:

$$\int_V \rho_f c_p \frac{\partial T}{\partial t} dV \approx \rho_f c_p \delta V \frac{\partial T}{\partial t} \Big|_P \approx \rho_f c_p \delta V \frac{3T_P^{n+1} - 4T_P^n + T_P^{n-1}}{2\Delta t}. \quad (2.68)$$

By summing all the flux approximations, the source terms and the time terms we derive an algebraic system for the temperature T in the CV centers:

$$a_P^T T_P - \sum_c a_c^T T_c = b^T. \quad (2.69)$$

2.2.2 Pressure–correction scheme

In this chapter we will address the application of the finite–volume methods for the numerical computation of incompressible fluid flows described by the Navier–Stokes equations (Eq. 2.17 – Eq. 2.18). The discretization principles which were presented in the previous chapter, can be directly applied to both, the momentum and continuity equations. As a result we get:

$$\sum_c \dot{m}_c = 0, \quad (2.70)$$

$$a_P^{v_i} v_{i,P} + \sum_c a_c^{v_i} v_{i,c} = \underbrace{-\delta V \left(\frac{\partial p}{\partial x_i} \right)_P}_{b^{v_i}} + S_{v_i}, \quad (2.71)$$

where $\dot{m}_c = (\rho v_i n_i)_c \delta S_c$ is the mass flux through the cell face c .

For incompressible flows the computation of the pressure is not trivial. A standard approach to solve the system of equations is to construct a coupled matrix A_{cpl} which contains the coefficients for both equations. As the gradient of the pressure contributes to each of the three momentum equations but on the other hand the continuity equation does not have a dominant variable and represent a kinematic constraint on the velocity field, rather than a dynamic equation, A_{cpl} is ill–conditioned. Furthermore the memory requirements in that case are enormous. Alternatively, there are artificial compressibility methods, where an additional pressure term is introduced in the continuity equation (for

more details see Hirsch [38]). Another approach, proposed by Patankar and Spalding [66] is to decouple the pressure and the velocity fields and to use a predictor–corrector scheme.

In this work a semi–implicit method for pressure–linked equations (SIMPLE) is applied. The idea behind this approach is to solve the Navier–Stokes equations independent from each other in an iterative process. First the momentum equation is solved using some predicted values for the pressure and for the mass fluxes (prediction step). The resulting values for the velocity field then does not satisfy the continuity equation. In the next (correction) step the new values for the velocity components and for the pressure are searched, such that the continuity equation is fulfilled. In what follows a short description of the SIMPLE pressure–correction algorithm is shown. For more details see Schäfer [84].

For the prediction step, we consider the discrete momentum equations with an estimated pressure field p^* :

$$a_P^{v_i} v_{i,P}^* + \sum_c a_c^{v_i} v_{i,c}^* = -\delta V \left(\frac{\partial p^*}{\partial x_i} \right)_P + S_{v_i} \quad (2.72)$$

The predicted velocity components v_i^* are then used for the calculation of the mass fluxes \dot{m}^* which substituted into the discrete continuity equation (Eq. 2.70) yields a mass source b_m

$$\sum_c \dot{m}_c^* = b_m. \quad (2.73)$$

The correction step starts with defining the corrections for the velocity and pressure:

$$v'_i = v^{k+1} - v_i^* \quad p' = p^{k+1} - p^*, \quad (2.74)$$

with $k + 1$ the actual iteration. By subtracting equations (Eq. 2.72) from (Eq. 2.71) we obtain the following relations for the velocity corrections :

$$v'_{i,c} = -\frac{1}{a_P^{v_i}} \sum_c a_c^{v_i} v'_{i,c} - \frac{\delta V}{a_P^{v_i}} \left(\frac{\partial p'}{\partial x_i} \right)_P. \quad (2.75)$$

Analogously for the continuity equation we get:

$$\sum_c \dot{m}'_c = -b_m. \quad (2.76)$$

Now we neglect the sum term in (Eq. 2.75) and insert the velocity corrections into (Eq. 2.76)

$$\sum_c \dot{m}'_c = \sum_c \rho v'_{i,c} n_i \delta S_c = \sum_c -\rho \left(\frac{\delta V}{a_P^{v_i}} \right)_c \left(\frac{\partial p'}{\partial x_i} \right)_c n_i \delta S_c = -b_m. \quad (2.77)$$

Applying one of the interpolation techniques, discussed in the previous chapter for all parameters on the cell face centers an equation for the pressure correction is obtained

$$a_P^p p'_P + \sum_c a_c^p p'_c = -b_m. \quad (2.78)$$

From the computed values for the pressure correction the velocity corrections can be obtained and the searched quantities v_i^{k+1} and p^{k+1} determined. The last step is the determination of the temperature from the equation

$$a_P^{T,k+1}T_P^{k+1} - \sum_c a_c^{T,k+1}T_c^{k+1} = b^{T,k+1}. \quad (2.79)$$

In case of a steady RANS calculation the scalar transport equations for the turbulent kinetic energy k and the dissipation ε should be also computed. A schematic representation of the pressure–correction scheme is shown in Figure 2.3.

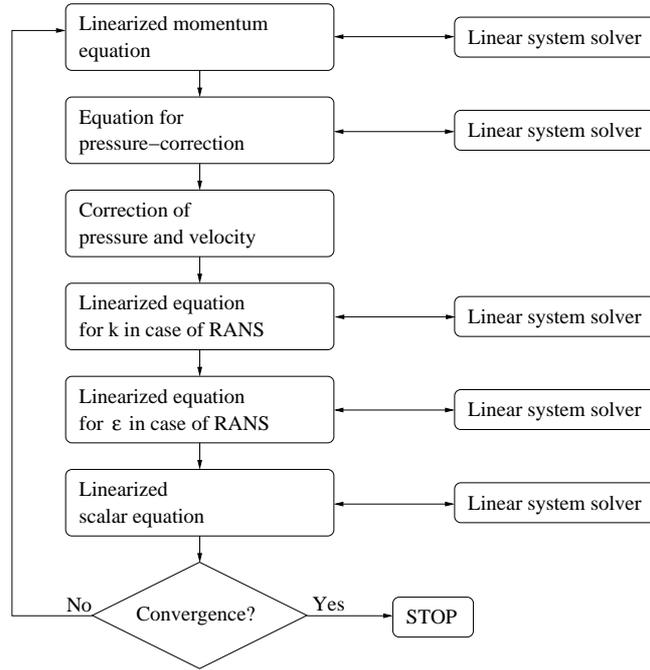


Figure 2.3: Schematic view of the SIMPLE pressure–correction method

In order to ensure the convergence of the pressure correction algorithm under-relaxation procedure is applied. This means that at every iteration the pressure p^{k+1} is corrected only with a certain portion $\alpha_p \in (0, 1]$ of the full pressure correction p'

$$p^{k+1} = p^* + \alpha_p p'. \quad (2.80)$$

Under-relaxation is also used for the velocity field and for the scalar quantities. Following Patankar [65] we take the algebraic equation for a generic variable ϕ at a point P , solved in an iterative process

$$a_P^\phi \phi_P^n + \sum_c a_c^\phi \phi_c^n = b^\phi. \quad (2.81)$$

At every iteration the new value of ϕ changes only with a fraction $\alpha_\phi \in (0, 1]$ of the calculated value i.e.

$$\phi^n = \phi^{n-1} + \alpha_\phi(\phi^{new} - \phi^{n-1}), \quad (2.82)$$

where ϕ^{new} is the result of (Eq. 2.80). After simple transformations we can write a new equation for the under-relaxed variable ϕ^n :

$$\frac{a_P^\phi}{\alpha_\phi} \phi_P^n + \sum_c a_c^\phi \phi_c^n = b^\phi + \frac{1 - \alpha_\phi}{\alpha_\phi} a_P^\phi \phi_P^{n-1} \quad (2.83)$$

2.2.3 Multigrid method

The use of different grid levels leads to a rapid reduction of the corresponding error frequencies. This is the idea on which the multigrid methods are built. In the last decades a variety of multigrid schemes were developed and investigated. An overview of the different multigrid approaches can be found in Trottenberg et al. [101] or Briggs [10]. In both works non-linear cases are also discussed. In Wesseling [105] multigrid techniques for the incompressible Navier–Stokes equations in combination with finite volume discretizations are presented. In this work the so-called full approximation scheme (FAS), proposed from Brandt [9], is used and will be briefly described in what follows. Detailed information on the realization and the properties of the method for the prediction of incompressible flows is given in Durst and Schäfer [22].

For our fluid domain Ω_f we consider a general notation

$$Nv = b^\Omega, \quad (2.84)$$

where N is assumed to be a nonlinear differential operator. We also assume that (Eq. 2.84) have been discretized on a given grid Ω_f^h by

$$N_h v_h = b_h, \quad (2.85)$$

After m SIMPLE iterations one obtains a solution \tilde{v}_h^m fulfilling:

$$N_h \tilde{v}_h^m = b_h - r_h^m, \quad (2.86)$$

with r_h the residual vector. We construct the defect equation on Ω_f^h as follows:

$$N_h(\tilde{v}_h^m + e_h) - N_h \tilde{v}_h^m = r_h^m, \quad \text{with } v_h = \tilde{v}_h^m + e_h. \quad (2.87)$$

For the transfer between the fine grid h and the coarser grid H we choose a restriction and an interpolation operator I_h^H and I_H^h see Trottenberg et al. [101]. Restricting (Eq.2.87) on Ω_f^H one gets:

$$N_H(\tilde{v}_H^m + \tilde{e}_H) - N_H \tilde{v}_H^m = r_H^m, \quad (2.88)$$

with $\tilde{v}_H^m = I_h^H \tilde{v}_h^m$ and $r_H^m = I_h^H r_h^m$. Solving this equation for the unknown \tilde{e}_H and interpolating the result on the fine grid ($\tilde{e}_h = I_H^h \tilde{e}_H$) gives the correction of the fine solution:

$$v_h^{m+1} = \tilde{v}_h^m + \tilde{e}_h. \quad (2.89)$$

The procedure can be recursively extended for more than two grids.

2.2.4 Parallelization

Despite the development of the computer technique nowadays, many problems are so large and/or complex that it is impractical or impossible to solve them on a single computer, especially given limited computer memory and enormous computational time needed. Parallel computing offers a solution to these problems.

Technically, there exist several different programming approaches for the implementation of parallel computing. Among the most popular are the use of parallelizing compilers or virtual shared memory. In both cases the code does not need to be changed and the whole parallelization work is done by the compiler or by the operating system. The disadvantage of both concepts is that the characteristic of the problem is not taken into account which leads to lower efficiency. The other approach, for which special changes in the source code are needed is the so called message passing. The data exchange between the different processors is performed by a message oriented communication, based on protocol like Parallel Virtual Machine (PVM) or Message Passing Interface (MPI). This way the best efficiency can be achieved and in this work the MPI standard is used.

Different concepts exist for parallelizing a continuum mechanical computations. In this work the grid partitioning technique which can be considered as the most frequently applied in practice, is used. The approach is based on the idea of the domain decomposition methods. The computational domain is splitted into non-overlapping subdomains, for which certain portion of the computation can be performed simultaneously by different processors. The problem formulation is extended with interface conditions which have to be fulfilled on the boundary between the subdomains, for which purpose data exchange between the processors is needed. In order to keep this communication effort as small as possible the idea of the “ghost cells” is used (for more detailed information on this approach see Schäfer [84]). Along the interface boundaries additional control volumes (ghost CV) are introduced (see Figure 2.4) which correspond to adjacent CVs of the neighbouring domain. Depending on the order of the used discretization scheme several layers of auxiliary CVs can be needed. The information in the ghost cells is actualized during exchange subroutines at suitable time points.

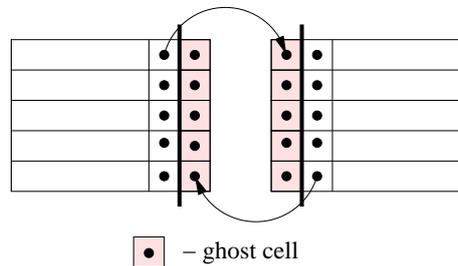


Figure 2.4: Ghost cells along subdomain interfaces

A global communication and synchronization over all subdomain is also needed by the parallel computation. At the beginning of the calculation the initial data (e.g. size of the subdomains, grid data, boundary conditions, etc.) have to be read from the user's input and distributed to all processors. The second global synchronization point is at the end of every SIMPLE iteration. In order to check whether the convergence criterion is satisfied the values of the residuals should be collected from all CPUs. At the end of the computation the result data have to be properly merged and stored for post processing. This is done by only one processor which gathers the needed information, saves the output and closes the whole calculation process.

The discussed additional processor communication is directly connected with the efficiency of the parallel algorithm. In order to assess the performance of the method we define the speed-up S_P and the efficiency E_P in the following manner:

$$S_P = \frac{T_1}{T_P} \quad \text{and} \quad E_P = \frac{T_1}{PT_P}, \quad (2.90)$$

where P is the number of processors, T_1 and T_P is the computing for the solution of the problem with one and P CPUs, respectively.

2.2.5 Finite-element methods

The finite-element methods (FEM) are nowadays, a standard tool for the numerical computation of solid mechanics problems. Here we outline the basic ideas of the methods. Detailed information can be found in Zienkiewicz et al. [118] or Wriggers [107].

For illustrating the methods the heat equation (Eq. 2.20) will be used, but this time for the solid domain, where the convective part drops out:

$$\rho_f c_p \frac{\partial T}{\partial t} - \rho Q + \kappa_s \frac{\partial}{\partial x_i} \frac{\partial T}{\partial x_i} = 0 \quad (2.91)$$

The finite-element methods are based on the variation form of the equations. As a beginning the weak form of the equations is constructed. We take arbitrary test functions $v(x_i)$ and multiply the (Eq. 2.91) by it:

$$v(x_i) \left(\rho_f c_p \frac{\partial T}{\partial t} - \rho Q + \kappa_s \frac{\partial}{\partial x_i} \frac{\partial T}{\partial x_i} \right) = 0 \quad (2.92)$$

Integrating (Eq.2.92) and applying the Green theorem and the integration by parts rule, yields:

$$\int_{\Omega_s} v(x_i) \left(\rho_f c_p \frac{\partial T}{\partial t} - \rho Q \right) d\Omega_s - \int_{\Omega_s} v(x_i) \kappa_s \frac{\partial T}{\partial x_i} d\Omega_s + \int_{\Gamma_s} v(x_i) \kappa_s \frac{\partial T}{\partial x_i} n_i d\Gamma_s = 0 \quad (2.93)$$

Equation (Eq.2.93) is the so called irreducible weak formulation of (Eq.2.91).

Now the solution of the weak formulation is searched in a finite dimensional subspace of Ω_s . By the FEM Ω_s is subdivided into nonoverlapping subdomains Ω_e , called finite elements

$$\Omega_s \approx \Omega_h = \sum_{e=1}^{N_{el}} \Omega_e, \quad (2.94)$$

where N_{el} is the number of nodes attached to an element. Integrals can now be summed over each element:

$$\begin{aligned} & \sum_{e=1}^{N_{el}} \int_{\Omega_h} v(x_i) \left(\rho_f c_p \frac{\partial T}{\partial t} - \rho Q \right) d\Omega_h - \\ & \sum_{e=1}^{N_{el}} \int_{\Omega_h} v(x_i) \kappa_s \frac{\partial T}{\partial x_i} d\Omega_h + \sum_{e=1}^{N_{el}} \int_{\Gamma_s} v(x_i) \kappa_s \frac{\partial T}{\partial x_i} n_i d\Gamma_s = 0 \end{aligned} \quad (2.95)$$

In the finite–element method isoparametric elements are used which satisfy the conditions

$$x_i = \sum_{I=1}^{N_{el}} N_I(\boldsymbol{\xi}) x_i^I \quad (2.96)$$

for coordinates and

$$T = \sum_{I=1}^{N_{el}} N_I(\boldsymbol{\xi}) T^I \quad (2.97)$$

for the temperature T . N_I are the so called shape functions for node I , x_i^I are values of the coordinates at node I and $\boldsymbol{\xi}$ are the natural coordinates of the element. For the shape functions the condition

$$\sum_{I=1}^{N_{el}} N_I(\boldsymbol{\xi}) = 1 \quad (2.98)$$

is fulfilled. In this work mainly 8 node brick elements are used, for which the local form function is given by

$$N_I(\boldsymbol{\xi}_i) = \frac{1}{8} (1 + \xi_1^I \xi_1) (1 + \xi_2^I \xi_2) (1 + \xi_3^I \xi_3), \quad I = 1, 8. \quad (2.99)$$

Following the Galerkin method the test functions are expressed as:

$$v = N_I(\boldsymbol{\xi}) v^I, \quad (2.100)$$

where v^I are arbitrary parameters.

For the derivatives of the shape functions we have:

$$\frac{\partial N_I}{\partial x_i} = \frac{\partial N_I}{\partial \xi_j} J_{ij}^{-1}, \quad (2.101)$$

with J_{ij} the Jacobian of the transformation $\mathbf{x}(x_i) \rightarrow \boldsymbol{\xi}(\xi_i)$. The gradients of the temperature T and the test functions v then can be expressed as:

$$\frac{\partial T}{\partial x_i} = \frac{\partial N_I}{\partial x_i} T^I, \quad \frac{\partial v}{\partial x_i} = \frac{\partial N_I}{\partial x_i} v^I. \quad (2.102)$$

With the above definitions the equation (Eq. 2.95) can be rewritten as

$$\sum_{e=1}^{N_{el}} (M_{IJ}) \frac{\partial T^J}{\partial t} + K_{IJ} T^J - F_I = 0, \quad (2.103)$$

with

$$M_{IJ} = \int_{\Omega_e} N_I \rho_s c_p N_J d\Omega_h \quad (2.104)$$

the element heat capacity matrix,

$$K_{IJ} = \int_{\Omega_e} \frac{\partial N_I}{\partial x_i} \kappa_s \frac{\partial N_J}{\partial x_i} d\Omega_h \quad (2.105)$$

the element conductivity matrix and

$$F_I = \int_{\Omega_e} \rho \frac{\partial N_I}{\partial x_i} Q d\Omega_h - \int_{\Gamma_s} \kappa_s N_I \frac{\partial T}{\partial x_i} n_i d\Gamma_s. \quad (2.106)$$

In matrix notation we can write

$$\mathbf{M}\dot{\mathbf{T}} + \mathbf{K}\mathbf{T} = \mathbf{F}. \quad (2.107)$$

Analogously one can derive a matrix equation for the equation of motion (Eq. 2.1):

$$\mathbf{M}\ddot{\mathbf{u}}_i + \mathbf{K}\mathbf{u}_i = \mathbf{F} \quad (2.108)$$

In the first case we have to solve the problem $\mathbf{R}(\dot{\mathbf{T}}, T) = 0$, whereas the second class is given by $\mathbf{R}(\ddot{\mathbf{u}}_i, \dot{\mathbf{u}}_i, \mathbf{u}_i) = 0$. \mathbf{R} is the residual, which in the case of the heat conduction problem is given by $\mathbf{R} = \mathbf{F} - \mathbf{K}\mathbf{T} - \mathbf{M}\dot{\mathbf{T}}$. For the first ODE which is of first order the backward Euler implicit method is used:

$$\dot{\mathbf{T}}^{n+1} \approx \frac{\mathbf{T}^{n+1} - \mathbf{T}^n}{\Delta t}. \quad (2.109)$$

For the second order differential equation which results from the finite deformations equation of motion (Eq. 2.108) the classical Newmark method (see [62]) is applied

$$u_i^{n+1} = u_i^n + \Delta t v_i^n + \frac{(\Delta t)^2}{2} [(1 - 2\beta) + 2\beta a_i^{n+1}], \quad (2.110)$$

$$v_i^{n+1} = v_i^n + \Delta t [(1 - \gamma) a_i^n + \gamma a_i^{n+1}], \quad (2.111)$$

where $a_i = \ddot{u}_i$ and $v_i = \dot{u}_i$. Heck [36] showed in his work, that the values $\beta = 0.5$ and $\gamma = 0.8$ are a good choice for the two parameters.

In case of non-linear models (e.g. finite deformations) a linearization of the equations is needed. In this work we use the iterative Newton–Raphson scheme (see Sendov and Popov [89]). The method starts with an initial guess and then the function is approximated by its tangent line:

$$J_F(u_i^n)(u_i^{n+1} - u_i^n) = -F(u_i^n). \quad (2.112)$$

Here F is the non-linear operator and $n + 1$ the current iteration.

Both the finite-volume and the finite-element methods results in algebraic systems of equations. More information how these are numerically solved can be found in Schäfer [84] and Zienkiewicz [118]. More detailed information on different iterative solvers is presented in Axelson [1].

Chapter 3

Thermal fluid–structure interaction

Thermal fluid–structure interaction (TFSI) occurs in many engineering applications, as for example in the design of aircraft turbines, car engines or in biomechanical applications. In all these cases a full set of interaction phenomenon between the fluid and the solid is observed. This includes flow acting on a solid (during drag, lift or movement), as well as solid acting on flow (during deformation or/and movement) and additional heat transfer in the whole system. Moreover, in most cases the fluid flow is turbulent. Developing a numerical model for simulating such problems is an important and complex task. In this chapter we present the developed numerical scheme for simulating thermal fluid–structure interaction. In the first two sections a short overview of the existing strategies is presented, together with a discussion of their properties. Then the chosen coupling algorithm is described.

3.1 Strategies for FSI

There exist two major techniques for numerical simulation of fluid–structure interaction. To the first one belong the direct coupling methods, also referred to by some other authors as monolithical coupling, in which the variable vector of the coupled system is solved simultaneously (see Hron and Turek [40], Matthies et al. [57], Heil [37], Blom [7] or Bathe and Zhang [3]). Although these methods show great stability, their use is limited mainly because of three reasons: the need for modification of the existing fluid and structure solvers (which is not always possible, the codes are often black boxes), less flexibility between both solvers and requirements of more storage and computational time for one iteration.

Partitioned approaches (for an overview see Felippa et al. [27]), on the other hand, allow to use specifically designed and highly optimized codes on different fields and offer significant benefits in terms of efficiency. The different solvers for

the structure and fluid dynamics are working separately and instead of one big, in most of the cases ill-posed system, smaller and better conditioned subsystems are solved.

The idea behind the partitioned approaches is based on the domain-decomposition technique (see Quarteroni and Valli [75]). The computation domain is divided into fluid and solid parts. In order to close the problem formulation, interface conditions are specified on the fluid–structure boundary. Depending on how these interface conditions are satisfied we distinguish loosely and strongly coupled partitioned schemes. Loose coupling requires only one solution of either field per time step in a sequentially staggered manner and is thus particularly appealing in terms of efficiency. Such methods are successfully applied and used in the field of aeroelasticity (see the work of Farhat et al. [25], [26] and the references therein or Piperno et al. [68], [69]) but in other applications, where large structural deformations appear, they can suffer from possible instabilities (see Causin et al. [13]).

For stability reasons, often a fully implicit formulation (also called strong coupling) has to be used, see Tallec and Mouro [97]. At every time step the equilibrium is satisfied jointly between fluid and structure, by applying an iterative procedure based on the Dirichlet–Neumann or Neumann–Dirichlet methods, borrowing the terminology from corresponding domain decomposition algorithms (see Quarteroni and Valli [75]). By Dirichlet–Neumann method we mean that in each iteration the fluid equations are solved with respect to primitive variables (v ; p) subject to Dirichlet boundary conditions at the interface (imposed displacements or velocities) and the structure equations subject to Neumann boundary conditions (imposed loads). The Neumann–Dirichlet method is carried out with a procedure dual to the one above. More detailed information for these approaches can be found in Matthies and Steindorf [56], Matthies et al. [57] or Yigit [113].

In this work an implicit partitioned solution approach which combines the advantages of weakly and strongly coupled schemes is used. As shown in Causin et al. [13] the use of Neumann–Dirichlet method is not of practical use because of its instability. That is why for the interface coupling procedure a Dirichlet–Neumann iteration method with constant or adaptive (see Mok [59]) underrelaxation is chosen. The method is realized on the basis of the finite-volume flow solver FASTEST (see [64]), the finite-element structural solver FEAP (see [98]) and the quasi-standard coupling interface MpCCI (see [83]). More detailed information on the implementation and the characteristics of the method can be found in Sieber [90].

3.2 Strategies for handling the heat transfer

Having chosen the method for the fluid–structure interaction, two different strategies can be implied for the calculation of the heat transfer. The first possibility is to calculate the temperature in the whole domain for every FSI iteration. In this case an extra thermo-mechanical coupling is needed (see

Teschauer [99]). As this is an analogue to the monolithical procedure for the force–displacement coupling, it suffers from the same drawbacks as discussed in the previous chapter. Having two highly optimized solvers for each domain of interest and partitioned coupling approach for the force–deformation coupling, a more logical way seems to be the non–overlapping Schwarz domain decomposition method. In this case the temperature transport equation is solved in every domain and on the boundary interface a iterative procedure is used which consists of prescribing alternately boundary conditions of Dirichlet and Neumann type. This algorithm is implemented and investigated in this work and here we will describe its mathematical idea.

Following Xu and Zou [108] we first split the whole computational domain of its natural fluid–solid interface, i.e. $\Omega = \Omega_f \cup \Omega_s$. We denote by $\Gamma_{\text{fsi}} = \partial\Omega_f \cap \partial\Omega_s$ the interface between the two subdomains. For the sake of simplicity in every computational subdomain we write the equation for the temperature as:

$$\begin{aligned} L_f T &= Q_f, \text{ with } T = t_f^0 \text{ on } \Gamma_f^d \text{ and } q = q_f^0 \text{ on } \Gamma_f^n \\ L_s T &= Q_s, \text{ with } T = t_s^0 \text{ on } \Gamma_s^d \text{ and } q = q_s^0 \text{ on } \Gamma_s^n, \end{aligned} \quad (3.1)$$

where L_f and L_s are some linear operators, Γ_f^d and Γ_s^d are the boundaries of both fluid and solid domains with prescribed temperature t_f^0 and t_s^0 (Dirichlet boundary conditions), respectively and Γ_f^n and Γ_s^n are the Neumann boundaries with prescribed flux q_f^0 and q_s^0 . In order to close the whole formulation the classical two steps iterative Schwarz alternating procedure is applied for prescribing interface conditions on the boundary Γ_{fsi} . For the iteration k the method can be written as:

STEP 1:

$$\begin{aligned} L_f T^{2k+1} &= Q_f \\ T^{2k+1} &= t_f^0 \quad \text{on } \Gamma_f^d \\ q^{2k+1} &= q_f^0 \quad \text{on } \Gamma_f^n \\ \lambda_1 T^{2k+1} + (1 - \lambda_1) q^{2k+1} &= \lambda_1 T^{2k} + (1 - \lambda_1) q^{2k} \quad \text{on } \Gamma_{\text{fsi}}, \end{aligned} \quad (3.2)$$

STEP 2:

$$\begin{aligned} L_s T^{2k+2} &= Q_s \\ T^{2k+2} &= t_s^0 \quad \text{on } \Gamma_s^d \\ q^{2k+2} &= q_s^0 \quad \text{on } \Gamma_s^n \\ \lambda_2 T^{2k+2} + (1 - \lambda_2) q^{2k+2} &= \lambda_2 T^{2k+1} + (1 - \lambda_2) q^{2k+1} \quad \text{on } \Gamma_{\text{fsi}}. \end{aligned} \quad (3.3)$$

Choosing the parameters λ_1 and λ_2 to be 1 and 0, respectively, we get a Dirichlet–Neumann alternative Schwarz decomposition method. The calculated temperature from the previous iteration on the fluid–solid interface is used as

a number of “temperature” iterations. Another variant of the scheme ITFSI2 with only one outer iteration cycle is given in Figure 3.2. Every time step consists of a nested iteration procedure for both the force–displacement and the thermal coupling. The fluid solver FASTEST starts first and calculates the flow and thermal field with the initial geometry. From this the friction and pressure forces F , as well as the normal thermal flux on the interacting walls q are computed and passed to the structural code as boundary conditions. The structural solver computes the deformations, with which then the fluid mesh is modified. The temperature at the coupling boundary is also computed and set as a boundary condition in FASTEST before the flow solver is started again.

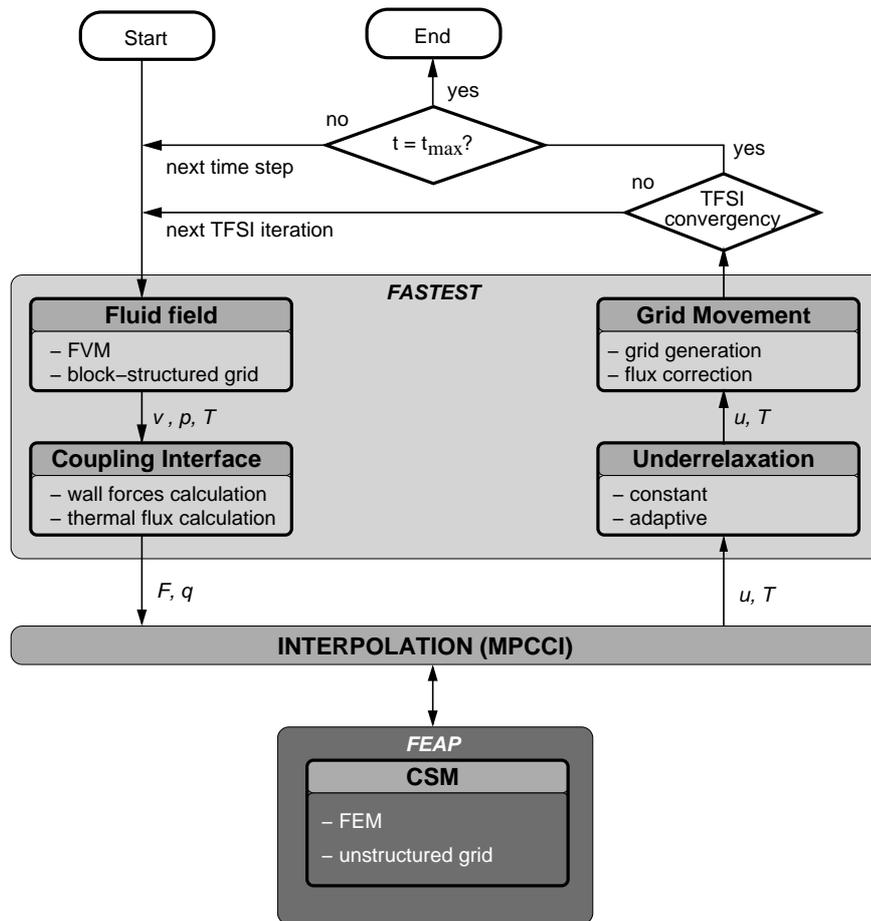


Figure 3.2: Flow chart of the coupling procedure ITFSI2

In all cases the transfer of the data between the two solvers is done via the quasi–standard coupling interface MpCCI, see [83], that controls the data communication and also carries out the interpolation of the data from the fluid

and solid grids.

The nested iteration procedure continues until reaching two convergence criteria ε_{FSI} and $\varepsilon_{\text{TEMP}}$, defined as follows:

$$\frac{\|u^{m-1} - u^m\|_\infty}{\|u^m\|_\infty} < \varepsilon_{\text{FSI}} \quad (3.4)$$

$$\frac{\|T^{m-1} - T^m\|_\infty}{\|T^m\|_\infty} < \varepsilon_{\text{TEMP}}, \quad (3.5)$$

where m is the number of the TFSI iteration and $\|\cdot\|$ denotes the uniform norm.

Underrelaxation is not needed for the alternating Schwarz procedure for the temperature coupling as shown, for example, in Gastaldi and Gastaldi (1994) [30]. Various test computations have shown that the force-deformation coupling scheme is rather sensitive with respect to the deformations. Here, situations that are far away from the physical equilibrium can arise which may lead to instabilities or even to divergence of the FSI iterations. This effect is known as artificially added mass effect and is very well documented in the literature (see Causin et al. [13] or Förster et al. [29]). These instabilities occur when the density of the fluid and the structure are comparable, or when the domain has a slender shape.

Over the past years different techniques for counteracting to this problem were developed. Degroote et al. [18] proposed an interface artificial compressibility method which mitigates the incompressibility constraint by adding a source term to the continuity equation in the fluid domain adjacent to the fluid-structure interface. This source term imitates the effect of the structure's displacement as a result of the fluid pressure and disappears when the coupling iterations have converged. Förster et al. [29], however, showed that for incompressible flow this approach only postpones the outset of the instability. Another new methodology is based on the reduced order methods (see Virendeels et al. [103]), however, this is not in the scope of this work. Here two types of underrelaxation are employed - the adaptive Aitken procedure (see Mok [59]) and a standard scheme. In both cases the actually computed displacements u_i^{act} are linearly weighted with the values u_i^{old} from the preceding iteration to give the new displacements u_i^{new} :

$$u_i^{\text{new}} = \alpha_{\text{FSI}} u_i^{\text{act}} + (1 - \alpha_{\text{FSI}}) u_i^{\text{old}}, \quad (3.6)$$

where $0 < \alpha_{\text{FSI}} \leq 1$. Note that the underrelaxation does not change the final converged result.

3.4 Interpolation of the thermal boundary flux

The wall forces and the normal thermal flux on the fluid-solid boundary are needed by the coupling algorithm. The friction and pressure forces F are calculated in the fluid domain as described in Ferziger and Perić (2002) [28] and

are then transferred to the solid solver as a boundary conditions. Depending on whether Dirichlet–Neumann or Neuman–Dirichlet Schwarz method is chosen for the thermal coupling the thermal flux should be calculated from the fluid solver or in the finite–element program FEAP. In the second case we just use the element shape functions for every boundary element (see Section 2.2.5). This yields:

$$q = \sum_{I=1}^{N_{el}} \frac{\partial N_I}{\partial x_i} T^I. \quad (3.7)$$

Because of the so called superconvergence (see Barlow [2] or Zhu and Lin [117]) most FEM codes calculate the gradients and stresses at the Gauss quadrature points of every element and different techniques for the recovery of the nodal values are then implied. Discussion over this subject can be found in Zienkiewicz et al. [118].

Here we will show how the thermal flux is interpolated in FASTEST in the case of Dirichlet–Neumann thermal coupling method. The aim is to have a second order grid independent interpolation, as in the author’s experience first order methods can be too diffusive and the error of the simulation can be too big.

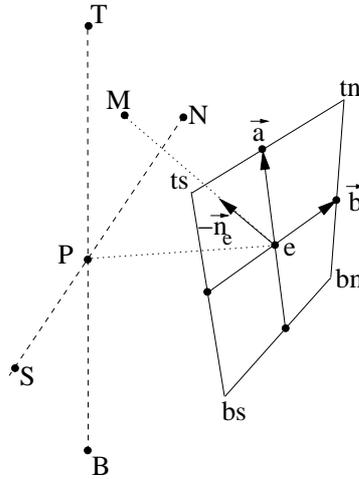


Figure 3.3: East side of an arbitrary control volume at the coupling interface (neighboring points are labeled according to the compass notation)

We will show the interpolation of the boundary normal thermal flux only for the east side of a boundary control volume, see Figure 3.3. The control volumes centers for the current control volume (P) and its neighbors from the top (T), bottom (B), south (S) and north (N) sides are denoted. ts , tn , bs , bn are the coordinates of the vertices of the east side and $e(x_e, y_e, z_e)$ is its center and \vec{n} is

the vector from e to P . The vectors \vec{a} and \vec{b} connect the centers of the edges and are used to determine the direction of the normal vector $n_e(n_x, n_y, n_z)$.

We need 3 points ($P_1(x_1, y_1, z_1)$, $P_2(x_2, y_2, z_2)$, $P_3(x_3, y_3, z_3)$) for a second order interpolation. As first one we always take P . Depending on the direction of the normal vector n_e for the other two we choose from T, B, N or S in the following manner:

$$P_2 \equiv \begin{cases} B & \text{if } \vec{p} \cdot \vec{a} \geq 0 \\ T & \text{if } \vec{p} \cdot \vec{a} < 0 \end{cases} \quad (3.8)$$

$$P_3 \equiv \begin{cases} S & \text{if } \vec{p} \cdot \vec{b} \geq 0 \\ N & \text{if } \vec{p} \cdot \vec{b} < 0 \end{cases} \quad (3.9)$$

Next we find the point $M(x, y, z)$ (see Figure 3.4) as intersection point between the normal vector n_e and the plane defined by P_1 , P_2 and P_3 :

$$\begin{aligned} x &= x_e + n_x t \\ y &= y_e + n_y t \\ z &= z_e + n_z t, \end{aligned} \quad (3.10)$$

where $t = \mathbf{A}/\mathbf{B}$ with:

$$\mathbf{A} = \begin{vmatrix} x_e - x_1 & y_e - y_1 & z_e - z_1 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{vmatrix} \quad (3.11)$$

$$\mathbf{B} = \begin{vmatrix} n_x & n_y & n_z \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{vmatrix} \quad (3.12)$$

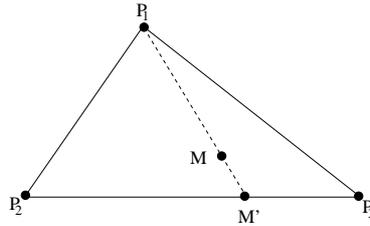


Figure 3.4: Modified bilinear interpolation using 3 points

The last step is a modified bilinear interpolation of the temperature in point M using 3 points P_1 , P_2 , P_3 . All four points lie in one plane. First a linear interpolation between P_2 and P_3 for the value of the temperature in point M' is applied (see Figure 3.4). The same procedure is repeated for the temperature

in point M using the values of the temperature in P_1 and M' . The thermal flux now can be approximated with a central differencing formula:

$$q_e = \lambda \frac{T_e - T_M}{\delta_e}, \quad (3.13)$$

where δ_e is the distance between M and e , T_e is the temperature in e and T_M is the interpolated temperature in point M . For laminar flow λ is just the conductivity of the fluid κ_f . In the case of turbulent flow λ is calculated as:

$$\lambda = \frac{\rho_f C_\mu^{1/4} \kappa_f k_P^{1/2} \delta_e}{\sigma_T^t [\ln(E y_P^+) + P_f \kappa_f]} \quad (3.14)$$

Here σ_T stands for the Prandtl number, σ_T^t is the turbulent Prandtl number, k_P is the value of the turbulent kinetic energy at the node P , y_P^+ is the dimensionless distance of point M from the wall and P_f is a function given by

$$P_f = 9.24 \left[\left(\frac{\sigma_T}{\sigma_T^t} \right)^{3/4} - 1 \right] \left[1 + 0.28 e^{(-0.007 \sigma_T / \sigma_T^t)} \right]. \quad (3.15)$$

The coefficient E is a constant which can be varied to simulate surface roughness (see Launder and Spalding (1974) [46]).

3.5 Flux correction

In Section 2.1.5 we presented the modified Navier–Stokes equations (2.51 – 2.53) for the case of moving grids. When the cell faces move, conservation of mass and all other conserved quantities is not necessarily ensured. Artificial mass sources may appear and accumulate with time, thus causing serious convergence problems.

For the arbitrary Lagrangian–Eulerian description, a discrete form of the space conservation law for a fluid control volume V_f with surface S_f :

$$\frac{d}{dt} \int_{V_f} = \int_{S_f} v^g \cdot n dS \quad (3.16)$$

should be taken into account (see Demirdzić and Perić [19]), in order to compute additional convective fluxes for the blocks that are moving. The volume difference between the new and the old CV can be expressed using the so called swept volumes, the volumes δV_c ($c = e, w, n, s, t, b$) which swept by the CV faces during the time step:

$$\frac{d}{dt} \int_{V_f} \approx \frac{V_f^n - V_f^{n-1}}{\Delta t_n} = \sum_c \frac{\delta V_c^n}{\Delta t_n}, \quad (3.17)$$

where n is the actual time and Δt_n is the time step.

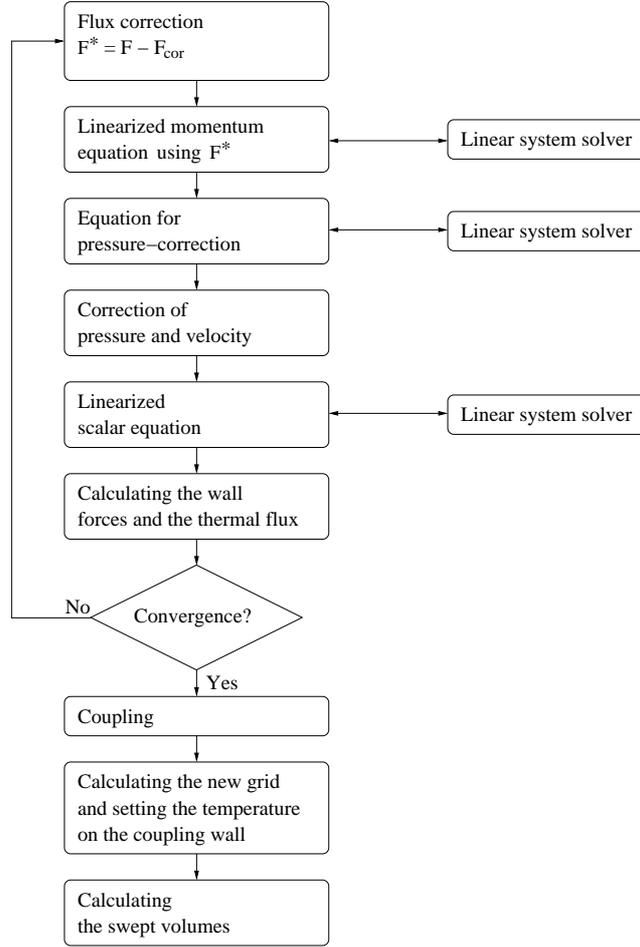


Figure 3.5: Schematic view of one TFSI iteration

Comparing equations 3.16 and 3.17, we see that the volume swept by the cell face c is:

$$\dot{V}_c = (v^g \cdot n)_c S_c = \frac{\delta V_c}{\Delta t}. \quad (3.18)$$

The mass flux through this cell face can therefore be calculated as:

$$\dot{m}_c = \int_{S_c} \rho_f (v - v^g) \cdot n dS \approx \rho_f (v \cdot n)_c S_c - \rho_f \dot{V}_c. \quad (3.19)$$

The last step is to introduce the flux correction term F_{cor}

$$F_{cor} = \sum_c \rho_f \dot{V}_c \quad (3.20)$$

into the SIMPLE pressure–correction scheme which is straightforward. A schematic view of one inner TFSI iteration can be seen in Figure 3.5.

Additionally, after every TFSI iteration the normalized distance to the nearest wall y^+ should be calculated in case of low-Reynolds RANS model. In case of dynamic Smagorinski model the filterwidth should be also updated.

3.6 Grid movements strategies

As already discussed, the fluid domain changes because of the deformation and/or movement of the solid. This leads to the need of changing the grid after every iteration of the coupling algorithm. Different strategies for managing the grid distortions are investigated from many authors during the last decade. Stein et al. [95] and Dwight [23] model the mesh as an elastic solid using the equations of linear elasticity. In both works special techniques are used in order to preserve the quality in boundary–layers and regions of high resolution. Liu et al. [53] generate Delaunay graph of the solution domain which can be moved easily during the geometric dynamic deformation, even for very large distortion. A one to one mapping between the Delaunay graph and the computational grid is maintained during the movement. The new computational grid after the dynamic movement is generated through the mapping while maintaining the primary qualities of the grid. In the work of Zhang et al. [116] the spring smoothing method is employed to perturb volume grids adapting to moving and/or deforming surfaces with relatively small displacement. In the spring smoothing, the grid nodes are assumed to be interconnected with a system of tension springs. Based on force balance, an equilibrium balance is sought to provide a smooth mesh. A more detailed and systematic investigation of this grid generation concepts can be found in the paper from Yang and Mavripilis [111]. Another grid distortion technique is the elliptic mesh generation which is based on the use of a composite mapping which consists of a nonlinear transfinite algebraic transformation and an elliptic transformation. The elliptic transformation is based on the Laplace equations for domains, or on Laplace–Beltrami equations for surfaces (see [92]).

In this work several different strategies based on linear interpolation, transfinite mapping and Laplace equations are used. The aim is to have a dynamic grid distortion procedure which preserves the grid quality even for large three dimensional structure deformations. In what follows, we present first some basis notations and ideas and then the mesh movement techniques.

3.6.1 General information and notation

As mentioned before for the fluid domain the in-house finite–volume solver FASTEST is used. FASTEST works on a block–structured grid which means that the grid topology should remain the same after the distortion procedure.

Let us consider a block in the 3 dimensional Cartesian space $X = (x, y, z)$. Without any restriction we take that the indices of the structured grid nodes

are $i = 1..N_i$, $j = 1..N_j$ and $k = 1..N_k$. Thus, every node in this block has the coordinates $\mathbf{x}_{i,j,k} = (x_{i,j,k}, y_{i,j,k}, z_{i,j,k})$, for every i, j, k . In a compass notation the faces are indexed from 1 to 6. The edges are also indexed with E_1-E_{12} .

The procedure starts with defining the edges of the distorted block. In order to avoid negative volumes not only linear interpolation was implemented but also a cubic spline interpolation which preserves the angle between the edge and the faces which it connects. The second step is distorting the block faces. Linear interpolation, transfinite interpolation, as well as, elliptic grid generation on minimal surface are implemented, as well as the possibility to move the face parallel to the opposite wall. At the end the new coordinates of the walls are used as boundary conditions for the distortion of the whole block. This is done via three dimensional linear interpolation, TFI or elliptic method. More information on every step of the process is shown in the following chapters.

3.6.2 Distortion of the edges

Three different interpolations are implemented for the edges. The first is a simple linear interpolation. Suppose that the edge is over the i direction and j and k are fixed to 1, i.e. the points $\mathbf{x}_{i,1,1}$, $i = 1, \dots, N_i$ are from this edge and $\mathbf{x}_{1,1,1}$ and $\mathbf{x}_{N_i,1,1}$ are its end points which have been already distorted. For the internal points we have:

$$\mathbf{x}_{i,1,1} = (\mathbf{x}_{N_i,1,1} - \mathbf{x}_{1,1,1})k_i + \mathbf{x}_{1,1,1}, \quad (3.21)$$

where

$$k_i = \frac{1}{len} \sum_{m=2}^i \|\mathbf{x}_{m,1,1} - \mathbf{x}_{m-1,1,1}\|, \quad len = \sum_{i=2}^{N_i} \|\mathbf{x}_{i,1,1} - \mathbf{x}_{i-1,1,1}\|. \quad (3.22)$$

For the interpolation with cubic spline the derivative vectors $\mathbf{T}_1(x, y, z)$ and $\mathbf{T}_2(x, y, z)$ at the end points of the edge are needed. We define:

$$\mathbf{a}_1 = -3\mathbf{x}_{1,1,1} + 3\mathbf{x}_{N_i,1,1} - 2\mathbf{T}_1 + \mathbf{T}_2 \quad (3.23)$$

$$\mathbf{a}_2 = 2\mathbf{x}_{1,1,1} - 2\mathbf{x}_{N_i,1,1} + \mathbf{T}_1 - \mathbf{T}_2. \quad (3.24)$$

The internal points we distort as:

$$\mathbf{x}_{i,1,1} = \mathbf{x}_{1,1,1} + \mathbf{a}_2 k_i^3 + \mathbf{a}_1 k_i^2 + \mathbf{T}_1 k_i. \quad (3.25)$$

The derivative vectors $\mathbf{T}_1(x, y, z)$ and $\mathbf{T}_2(x, y, z)$ are chosen, that the angles between the edge and the faces it connects remain the same as in the initial geometry. Here we will describe shortly how this is done only for the $\mathbf{T}_1(x, y, z)$. The second vector $\mathbf{T}_2(x, y, z)$ can be found analogously.

We first define the vectors \vec{a} , \vec{b} and \vec{c} in the initial geometry:

$$\vec{a} = \mathbf{x}_{1,2,1} - \mathbf{x}_{1,1,1}, \quad \vec{b} = \mathbf{x}_{1,1,2} - \mathbf{x}_{1,1,1}, \quad \vec{c} = \mathbf{x}_{2,1,1} - \mathbf{x}_{1,1,1}, \quad (3.26)$$

i.e. these are the vectors between the corner point and its neighbour edge nodes. The vector \vec{c} lies on the edge which should be distorted and \vec{a} and \vec{b} lie in the face A ($\mathbf{x}_{1,j,k} \in A$, $j = 1, \dots, N_j$ and $k = 1, \dots, N_k$). Vector \vec{e} we define as a vector product between \vec{a} and \vec{b} , thus it is orthogonal to A . \vec{a}_1 we take to be the projection of \vec{c} into \vec{a} , i.e.

$$\vec{a}_1 = \vec{a} \frac{(\vec{c} \cdot \vec{a})}{|\vec{a}|^2}. \quad (3.27)$$

Additionally two more vectors are defined:

$$\vec{f} = \vec{b} - \vec{a} \frac{\vec{b} \cdot \vec{a}}{|\vec{a}|^2}, \quad \vec{f}_1 = \vec{f} \frac{\vec{c} \cdot \vec{f}}{|\vec{f}|^2}. \quad (3.28)$$

The direction of the edge in 3-dimensional space can be uniquely defined by the three angles α , β and γ :

$$\cos \alpha = \frac{(\vec{a} \cdot \vec{c})}{|\vec{a}| |\vec{c}|}, \quad \cos \beta = \frac{(\vec{b} \cdot \vec{c})}{|\vec{b}| |\vec{c}|}, \quad \cos \gamma = \frac{(\vec{e} \cdot \vec{c})}{|\vec{e}| |\vec{c}|} \quad (3.29)$$

and those should be also preserved in the new geometry.

After one or more coupling iteration the face can be changed and the vectors \vec{a} and \vec{b} transform to \vec{a}' and \vec{b}' . Following the same procedure as before we define the following vector in the distorted geometry:

$$\vec{e}' = \vec{a}' \times \vec{b}', \quad \vec{a}_1' = \vec{a}' \frac{|\vec{a}_1|}{|\vec{a}'|} \text{sign}(\cos \alpha), \quad (3.30)$$

$$\vec{f}' = \vec{b}' - \vec{a}' \frac{\vec{b}' \cdot \vec{a}'}{|\vec{a}'|^2}, \quad \vec{f}_1' = \vec{f}' \frac{|\vec{f}_1|}{|\vec{f}'|} \text{sign}(\cos \beta), \quad (3.31)$$

where

$$\text{sign}(\psi) = \begin{cases} 1 & \text{if } \psi \geq 0 \\ -1 & \text{if } \psi < 0. \end{cases} \quad (3.32)$$

With those definitions the angles between the vector $\vec{d} = \vec{a}_1' + \vec{f}_1'$ and \vec{a}' and \vec{b}' are exactly α and β , respectively. In order to fulfill the last condition, namely to preserve the angle γ , we define $\vec{u} = (u_x, u_y, u_z)$:

$$\vec{u} = \frac{\vec{d} \times \vec{e}'}{|\vec{d} \times \vec{e}'|} \quad (3.33)$$

and use it to rotate \vec{e}' around it by an angle of γ . The rotation matrix with the given rotation axis and angle is defined as following:

$$\mathbf{M}_R = \begin{bmatrix} u_x^2 + (1 - u_x^2)c & u_x u_y (1 - c) - u_z s & u_x u_z (1 - c) + u_y s \\ u_x u_y (1 - c) + u_z s & u_y^2 + (1 - u_y^2)c & u_y u_z (1 - c) - u_x s \\ u_x u_z (1 - c) - u_y s & u_y u_z (1 - c) + u_x s & u_z^2 + (1 - u_z^2)c, \end{bmatrix} \quad (3.34)$$

where $c = \cos \gamma$ and $s = \sin \gamma$. The vector \mathbf{T}_1 is now defined as:

$$\mathbf{T}_1 = \frac{\mathbf{M}_R \vec{e}'}{|\mathbf{M}_R \vec{e}''|}. \quad (3.35)$$

The so defined vector preserves the angles α , β and γ without any restrictions for the both, initial and distort geometries.

3.6.3 Distortion of the faces

Three different types of faces are distinguished in the procedure. A block can have coupling faces, faces which should remain the same and faces which should be distorted. In the last case we suppose that the edges of the wall have been already changed. The simplest technique is to use linear interpolation, i.e. the Equations 3.21 – 3.22 along one of the parametrical directions. Another algebraic solution is the 2 dimensional transfinite interpolation.

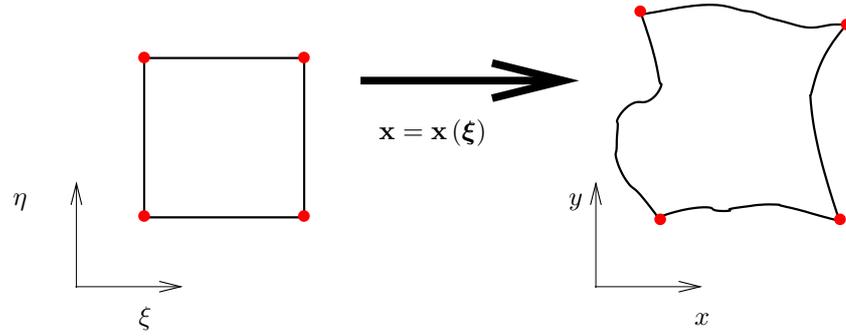


Figure 3.6: 2D TFI procedure. Computational domain (left) is mapped onto the Physical domain (right).

We consider a mapping between the physical domain in terms of (x, y, z) and the computational domain parametrized with (ξ, η, ζ) (see Figure 3.6). The vectors in the computational space will be represented by bold symbol as follows: $\boldsymbol{\xi} = (\xi_i)$, where $i = 1 \dots 3$. The mapping is one-to-one nevertheless that $0 \leq \xi \leq 1$, $0 \leq \eta \leq 1$ and ζ is a constant. We construct the mapping function using the formula of the linear 2D TFI:

$$\mathbf{x}(\boldsymbol{\xi}) = \mathbf{U} + \mathbf{V} - \mathbf{U} \cdot \mathbf{V}, \quad (3.36)$$

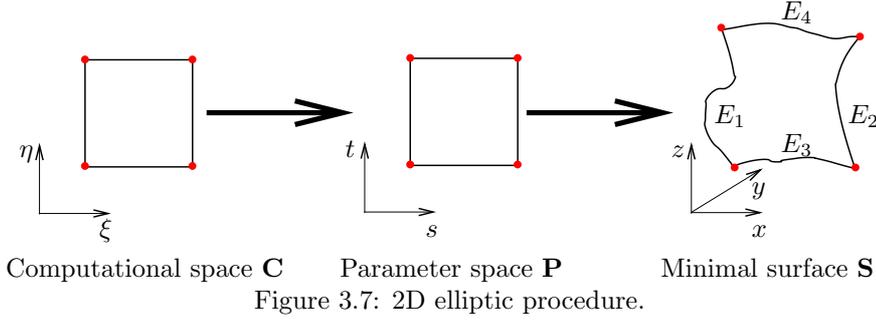
where

$$\mathbf{U} = (1 - \xi)\mathbf{x}(0, \eta) + (1 + \xi)\mathbf{x}(1, \eta), \quad (3.37)$$

$$\mathbf{V} = (1 - \eta)\mathbf{x}(\xi, 0) + (1 + \eta)\mathbf{x}(\xi, 1) \quad (3.38)$$

and $\mathbf{U} \cdot \mathbf{V}$ is the binary product of the two terms

$$\begin{aligned} \mathbf{U} \cdot \mathbf{V} &= (1 - \xi)(1 - \eta)\mathbf{x}(0, 0) \\ &+ (1 + \xi)(1 - \eta)\mathbf{x}(1, 0) \\ &+ (1 - \xi)(1 + \eta)\mathbf{x}(0, 1) \\ &+ (1 + \xi)(1 + \eta)\mathbf{x}(1, 1). \end{aligned} \quad (3.39)$$



The last implemented technique is an elliptic grid generation method on minimal surface. A minimal surface is defined as a surface bounded by four edges and with zero mean curvature. Following Spekreijse [92] we define a parameter system with two parameters (see Figure 3.7). We consider four curved edges E_1 , E_2 , E_3 and E_4 situated in the 3-dimensional physical space. The parameter space **P** we define as a unit square with Cartesian coordinates $\mathbf{s} = (s, t)$, where s is the normalized arclength along edges E_3 and E_4 and t – the normalized arclength along edges E_1 and E_2 . The mapping $\mathbf{s} : \mathbf{C} \rightarrow \mathbf{P}$ we define by two algebraic equations:

$$s = \mathbf{s}_{E_3}(\xi)(1 - t) + \mathbf{s}_{E_4}(\xi)t \quad (3.40)$$

$$t = \mathbf{t}_{E_1}(\eta)(1 - s) + \mathbf{t}_{E_2}(\eta)s \quad (3.41)$$

We introduce consequently the two covariant base vectors \mathbf{a}_1 and \mathbf{a}_2 , the contravariant base vectors \mathbf{a}^1 and \mathbf{a}^2 and the contravariant metric base vectors \mathbf{a}^{ij} , $i = 1, 2$, $j = 1, 2$:

$$\mathbf{a}_1 = \frac{\partial \mathbf{x}}{\partial \xi} \quad (3.42)$$

$$\mathbf{a}_2 = \frac{\partial \mathbf{x}}{\partial \eta} \quad (3.43)$$

$$(\mathbf{a}^i, \mathbf{a}_j) = \delta_{ij} \quad (3.44)$$

$$a^{ij} = (\mathbf{a}^i, \mathbf{a}^j) \quad (3.45)$$

We define three control functions:

$$\mathbf{P}_{11} = -\mathbf{T}^{-1} \begin{pmatrix} \mathbf{s}_{\xi\xi} \\ \mathbf{t}_{\xi\xi} \end{pmatrix}, \quad \mathbf{P}_{12} = -\mathbf{T}^{-1} \begin{pmatrix} \mathbf{s}_{\xi\eta} \\ \mathbf{t}_{\xi\eta} \end{pmatrix}, \quad \mathbf{P}_{22} = -\mathbf{T}^{-1} \begin{pmatrix} \mathbf{s}_{\eta\eta} \\ \mathbf{t}_{\eta\eta} \end{pmatrix}, \quad (3.46)$$

where

$$\mathbf{T} = \begin{pmatrix} \mathbf{s}_\xi & \mathbf{s}_\eta \\ \mathbf{t}_\xi & \mathbf{t}_\eta \end{pmatrix} \quad (3.47)$$

With these definitions we can write a Poisson equation for the grid coordinates:

$$\begin{aligned} a^{11}\mathbf{x}_{\xi\xi} + 2a^{12}\mathbf{x}_{\xi\eta} + a^{22}\mathbf{x}_{\eta\eta} + (a^{11}\mathbf{P}_{11}^1 + 2a^{12}\mathbf{P}_{12}^1 + a^{22}\mathbf{P}_{22}^1) \\ + (a^{11}\mathbf{P}_{11}^2 + 2a^{12}\mathbf{P}_{12}^2 + a^{22}\mathbf{P}_{22}^2) = 0. \end{aligned} \quad (3.48)$$

Together with the requirement which follows from the condition of minimal surface $\Delta\mathbf{x} = 0$ we have a closed set of equations for the grid coordinates. To solve this system we use the Picard iteration procedure. We first rewrite the Poisson equation as:

$$A\mathbf{x}_{\xi\xi} + 2B\mathbf{x}_{\xi\eta} + C\mathbf{x}_{\eta\eta} + D\mathbf{x}_\xi + E\mathbf{x}_\eta = 0, \quad (3.49)$$

where

$$\begin{aligned} A &= (\mathbf{x}_\eta, \mathbf{x}_\eta), \quad B = -(\mathbf{x}_\xi, \mathbf{x}_\eta), \quad C = (\mathbf{x}_\xi, \mathbf{x}_\xi) \\ D &= AP_{11}^1 + 2BP_{12}^1 + CP_{22}^1 \\ E &= AP_{11}^2 + 2BP_{12}^2 + CP_{22}^2 \end{aligned} \quad (3.50)$$

and then using a Picard iteration for the k^{th} iteration we get:

$$A^{k-1}\mathbf{x}_{\xi\xi}^k + 2B^{k-1}\mathbf{x}_{\xi\eta}^k + C^{k-1}\mathbf{x}_{\eta\eta}^k + D^{k-1}\mathbf{x}_\xi^k + E^{k-1}\mathbf{x}_\eta^k. \quad (3.51)$$

The iteration process contains the following steps:

- computing of the coefficients A^{k-1} , B^{k-1} , C^{k-1} , D^{k-1} , E^{k-1} using CDS.
- Discretize $\mathbf{x}_{\xi\xi}^k$, $\mathbf{x}_{\xi\eta}^k$, $\mathbf{x}_{\eta\eta}^k$, \mathbf{x}_ξ^k and \mathbf{x}_η^k using CDS.
- Solving the linear system of equations for the unknowns \mathbf{x}^k using the Gauss-Seidel method.

The process is repeated until fulfilling of a fixed convergence criterion.

3.6.4 Distortion of the blocks

The same three techniques as in the previous chapter, but this time in the, 3-dimensional space, are used for the distortions of the whole block. The Equations 3.21 – 3.22 applied in every parametrical directions give a linear distortion procedure. The transfinite interpolation (TFI) in 3D can be written as:

$$\mathbf{x}(\xi) = \mathbf{U}_1 + \mathbf{U}_2 + \mathbf{U}_3 - \mathbf{U}_1 \cdot \mathbf{U}_2 - \mathbf{U}_2 \cdot \mathbf{U}_3 - \mathbf{U}_3 \cdot \mathbf{U}_1 + \mathbf{U}_1 \cdot \mathbf{U}_2 \cdot \mathbf{U}_3, \quad (3.52)$$

where

$$\mathbf{U}_i = (1 - \xi_i) \mathbf{x}|_{\xi_i=0} + (1 + \xi_i) \mathbf{x}|_{\xi_i=1}. \quad (3.53)$$

The elliptic grid generation is obtained by solving the following Poisson equation:

$$\begin{aligned}
 & a^{11}\mathbf{x}_{\xi\xi} + 2a^{12}\mathbf{x}_{\xi\eta} + a^{13}\mathbf{x}_{\xi\zeta} + a^{22}\mathbf{x}_{\eta\eta} + 2a^{23}\mathbf{x}_{\eta\zeta} + a^{33}\mathbf{x}_{\zeta\zeta} \\
 & + (a^{11}\mathbf{P}_{11}^1 + 2a^{12}P_{12}^1 + 2a^{13}\mathbf{P}_{13}^1 + a^{22}\mathbf{P}_{22}^1 + 2a^{23}P_{23}^1 + 2a^{33}\mathbf{P}_{33}^1)\mathbf{x}_{\xi} \\
 & + (a^{11}\mathbf{P}_{11}^2 + 2a^{12}P_{12}^2 + 2a^{13}\mathbf{P}_{13}^2 + a^{22}\mathbf{P}_{22}^2 + 2a^{23}P_{23}^2 + 2a^{33}\mathbf{P}_{33}^2)\mathbf{x}_{\eta} \quad (3.54) \\
 & + (a^{11}\mathbf{P}_{11}^3 + 2a^{12}P_{12}^3 + 2a^{13}\mathbf{P}_{13}^3 + a^{22}\mathbf{P}_{22}^3 + 2a^{23}P_{23}^3 + 2a^{33}\mathbf{P}_{33}^3)\mathbf{x}_{\zeta} = 0,
 \end{aligned}$$

with the following control functions \mathbf{P}_{11} , \mathbf{P}_{12} , \mathbf{P}_{13} , \mathbf{P}_{22} , \mathbf{P}_{23} , \mathbf{P}_{33} :

$$\begin{aligned}
 \mathbf{P}_{11} &= -\mathbf{T}^{-1} \begin{pmatrix} s_{\xi\xi} \\ t_{\xi\xi} \\ u_{\xi\xi} \end{pmatrix}, & \mathbf{P}_{12} &= -\mathbf{T}^{-1} \begin{pmatrix} s_{\xi\eta} \\ t_{\xi\eta} \\ u_{\xi\eta} \end{pmatrix}, & \mathbf{P}_{13} &= -\mathbf{T}^{-1} \begin{pmatrix} s_{\xi\zeta} \\ t_{\xi\zeta} \\ u_{\xi\zeta} \end{pmatrix}, \\
 \mathbf{P}_{22} &= -\mathbf{T}^{-1} \begin{pmatrix} s_{\eta\eta} \\ t_{\eta\eta} \\ u_{\eta\eta} \end{pmatrix}, & \mathbf{P}_{23} &= -\mathbf{T}^{-1} \begin{pmatrix} s_{\eta\zeta} \\ t_{\eta\zeta} \\ u_{\eta\zeta} \end{pmatrix}, & \mathbf{P}_{33} &= -\mathbf{T}^{-1} \begin{pmatrix} s_{\zeta\zeta} \\ t_{\zeta\zeta} \\ u_{\zeta\zeta} \end{pmatrix}, \quad (3.55)
 \end{aligned}$$

where

$$\mathbf{T} = \begin{pmatrix} \mathbf{s}_{\xi} & \mathbf{s}_{\eta} & \mathbf{s}_{\zeta} \\ \mathbf{t}_{\xi} & \mathbf{t}_{\eta} & \mathbf{t}_{\zeta} \\ \mathbf{u}_{\xi} & \mathbf{u}_{\eta} & \mathbf{u}_{\zeta} \end{pmatrix} \quad (3.56)$$

and the computational and parametrical spaces \mathbf{C} and \mathbf{P} are 3–dimensional, i.e. $\boldsymbol{\xi} = (\xi, \eta, \zeta)$ and $\mathbf{s} = (s, t, u)$ and the algebraic mapping $\mathbf{s} : \mathbf{C} \rightarrow \mathbf{P}$ is defined as:

$$\begin{aligned}
 s &= s_{E_1}(\xi)(1-t)(1-u) + s_{E_2}(\xi)t(1-u) + s_{E_3}(\xi)(1-t)u + s_{E_4}(\xi)tu, \\
 t &= t_{E_5}(\eta)(1-s)(1-u) + t_{E_6}(\eta)s(1-u) + t_{E_7}(\eta)(1-s)u + t_{E_8}(\eta)su, \quad (3.57) \\
 u &= s_{E_9}(\zeta)(1-s)(1-t) + s_{E_{10}}(\zeta)s(1-t) + s_{E_{11}}(\zeta)(1-s)t + s_{E_{12}}(\zeta)st,
 \end{aligned}$$

with $E_1 - E_{12}$ the edges of the block. According to this bilinear transformation the new coordinates of the interior nodes of the 6 faces are computed and used as a boundary conditions. The edges are computed via a linear interpolation, thus closing the system of equations. As in 2–dimensional case a Picard iteration together with the Gauss–Seidel method are used for solving the resulting Poisson equation system.

3.7 Multigrid techniques for thermal fluid–structure interaction

Coupled fluid solid problems usually require a high computational effort, especially in three–dimensional cases. As these are only considered in this work, accelerating the calculations is an important aspect. In this thesis we use a multigrid method for fluid computations on moving grids and the partition

coupling procedure is executed only on the finest grid level. In the nonlinear multigrid algorithm special care has to be taken to the consistent treatment of the additional fluxes due to the grid movement. If this is done properly, the method shows the typical behavior with acceleration factors increasing with the grid size also known from uncoupled simulations. Yigit et al. [114] showed that acceleration factors up to 12 for the finest grid could be achieved for a typical fully coupled three–dimensional FSI simulation.

Recently different multigrid ideas for fluid–structure interaction were developed and studied. Zuijlen et al. [102] showed that coupling not only the structure with the fine flow mesh, but also with the coarse flow mesh (often present due to the multigrid scheme) leads to a significant efficiency improvement. As solving numerically the flow equations typically takes much longer than the structural mechanics equations and as multigrid is not standard in structure solvers, Zuijlen et al. [102] do not coarsen the structure or the interface. A similar strategy is applied by Heck [36], who also studied different multigrid coupling strategies but again with no coarsening of the structure mesh. On the other hand in Sachs and Schäfer [82] the mathematical basis for applying a full approximation scheme in the whole computational domain are presented. They state that in their future work this procedure will be implemented and studied in the ITFSI scheme.

3.8 Parallelization of the thermal fluid–structure interaction

In most engineering applications with thermal fluid structure interaction, the computational fluid dynamics part is much more time and resource intensive. Consider, for instance, an elastic beam in a turbulent channel flow. Only a few beam finite elements can be enough for properly resolving the mechanical behaviour of the structure. On the other hand, depending on the chosen numerical model, the Reynold’s number, etc, millions of control volumes can be needed by the fluid solver. That is the reason why in this work only the fluid code FASTEST is used in its parallel version. The structure mechanics solver, on the other hand, is working on only one processor. Special attention to the coupling is then needed.

The program FASTEST is a parallel block structured code. As discussed in the previous chapter, three global synchronization points are used: at the beginning of the computation, after every SIMPLE iteration and at the end of the program execution. In the case of a (thermal) fluid–structure interaction two more global data exchange procedures should be applied before and after the coupling between the two codes.

Consider the coupling procedure ITFSI2 (Figure 3.2). After calculating the thermal fluxes and the wall forces, the information should be transferred via an interpolation to FEAP. Two different procedures, responsible for managing this task, are implemented in this work. In the first case (1 to 1) one processor

collect the whole interface information and send it via MpCCI to the other code (see Figure 3.8(a)). The biggest advantage of this approach is its easy implementation. The other technique (n to 1) uses the parallel features of the black box coupling interface MpCCI which were implemented first in the version 3.0.5. Every processor which works on a block laying on the fluid–solid interface register itself in the global MPI world initialized by MpCCI (see Figure 3.8(b)). The quasi–standard interface MpCCI is also responsible for the synchronization between the different processors. In such a manner, less communication is needed for the coupling between the two codes.

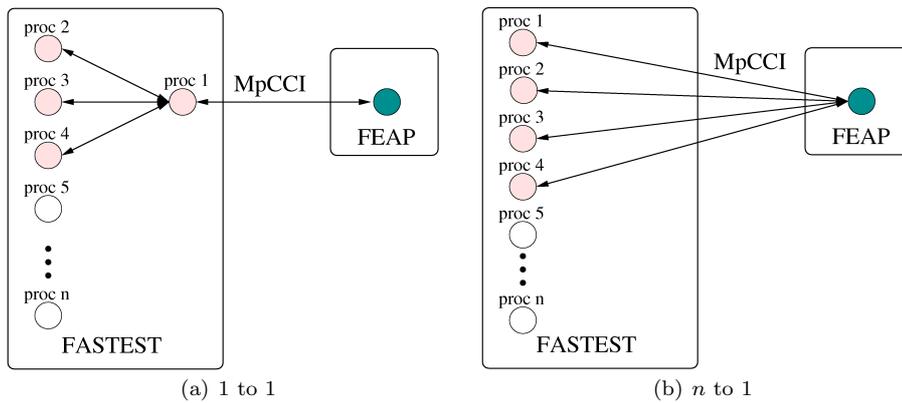


Figure 3.8: Parallelization strategies

Special attention should be also paid to the grid distortion procedure. The “ghost cells” idea can be also used here, but the existing subroutines can not be directly applied. The main difference is that, the finite–volume code needs the values of the interface quantities in the control volumes’ centers. The grid distortions, on the other hand, are presented with their nodal values. In this work a parallel grid distortion concept is developed and implemented in the fluid solver FASTEST.

Chapter 4

Results and discussion

In this chapter we present results for the developed and implemented numerical schemes for thermal fluid structure interaction. Several representative test cases, among which an analytical test case, a benchmark test case, a test case with reference solution and a test case with experimental results, are considered in order to validate the methods, to prove their accuracy and to investigate their numerical characteristics.

We first start with a validation and verification of the numerical scheme for simulating thermal fluid–structure interaction in the case of laminar fluid flow. The results from series of calculations with different RANS models are then presented. In the next two sections the numerical characteristics of the developed scheme are investigated and different implemented techniques for accelerating the computations are proposed. At the end the effect of the moving grids on a LES calculation is studied and the properties of a TFSI simulation with a dynamic LES model for the fluid flow are discussed.

4.1 Validation and verification of the laminar thermal fluid–structure interaction

For the validation and verification of the simulation scheme, when the fluid flow is laminar, two test cases are used. Figure 4.1 shows the geometry and the non-Cartesian numerical O-grid for a simple test case with a $1D$ analytical solution (TC1). The computational domain consists of two cubes. The left one is occupied by fluid and the other one is a solid block. The material properties for both, fluid and solid, are shown in Table 4.1. The temperature is prescribed at the left ($T_f = 300$ K) and at the right T_s ends of the domain. For T_s two different values are used, i.e. 310 K and 3000 K in order to investigate the influence of the thermal gradient. All other walls are treated as adiabatic walls. A no-slip boundary condition is used on every wall of the fluid domain. In the solid part of the domain all walls are fixed (no deformation allowed). As the

fluid is not in motion the analytical solution in the case $T_s = 310$ is given by

$$T_{\text{fluid}} = 98.68x + 300, \quad (4.1)$$

$$T_{\text{solid}} = 1.32x + 309,736, \quad (4.2)$$

where T_{fluid} and T_{solid} describe the temperature in the fluid and in the solid domains, respectively and x is the lateral direction. When $T_s = 3000$ the analytical solution for the temperature is

$$T_{\text{fluid}} = 26644.7 \frac{x}{m} + 300, \quad (4.3)$$

$$T_{\text{solid}} = 355.3 \frac{x}{m} + 2928,94. \quad (4.4)$$

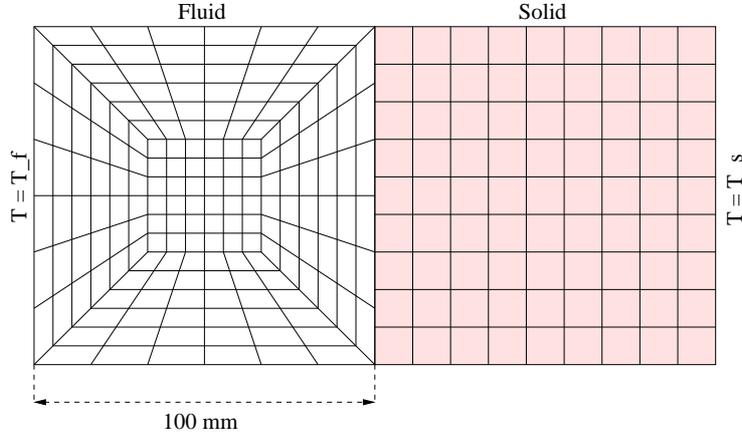


Figure 4.1: Geometry and numerical for TC1

The second test case TC2 (Figure 4.2) is a two-dimensional benchmark test case with finite deformation (see Schäfer and Turek [87]). An elastic beam, attached to a fixed cylinder is placed in a channel with laminar inflow with:

$$v_{in} = 35.693\check{v}_{in}\left(0.41 - \frac{y}{m}\right)\frac{y}{m}, \quad (4.5)$$

where y is the lateral direction and \check{v}_{in} is set so, that the Reynolds number is $Re = 100$. The front cylinder has a fixed temperature of 1000 K, thus acting as a thermal source. The inflow temperature of the fluid is set to $T_{in} = 300$ K. The material properties are chosen (see Table 4.1) that large deformations of the structure are possible. At the right end of the channel a zero gradient outflow boundary condition is applied. For all solid walls no-slip boundary conditions are used. The time step is set to 0.002 s. For the force-displacement coupling underrelaxation is employed with $\alpha_{\text{FSI}} = 0.1$.

Let us first consider the two coupling schemes ITFSI1 and ITFSI2. Both are used for the calculation of the two laminar test cases. For TC1 where no

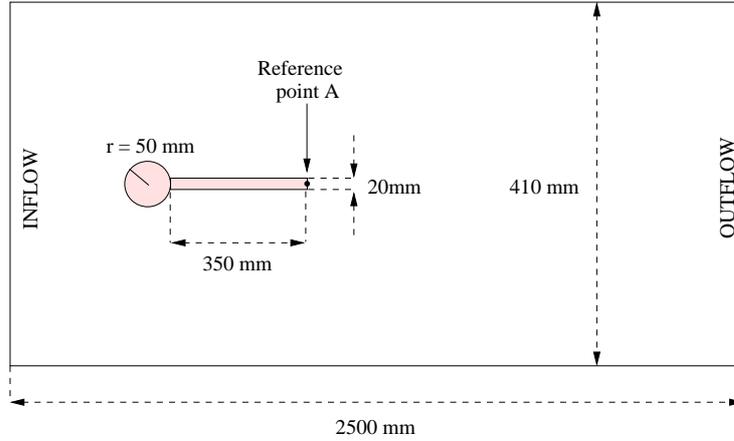


Figure 4.2: Geometry of TC2

deformations are taking place, the two schemes are in fact identical. For TC2 this is not the case. Every ITFSI1 force–displacement coupling iteration needs 2–3 outer thermal iterations. By ITFSI2 these thermal iterations are “hidden” behind the greater number of FSI iterations. In such a manner the ITFSI2 is much more effective. On the other hand no stability problems are observed using the second scheme. The values for the critical underrelaxation parameters are the same, even when the material parameters are temperature dependent. The results from the both schemes are also identical for every time step. Such the additional work of ITFSI1 can be considered as a not needed luxury. In author’s experience simulations of a turbulent TFSI using different RANS or LES models does not effect this observation. From now on only the second scheme will be used and all presented results are due to computations with ITFSI2.

In Table 4.2 the maximum difference between simulation and analytical results for the temperature on the coupling wall ($E_{coupling}$) and the temperature over the whole domain (E_{global}) for TC1 are shown for $T_s = 310$ K and $T_s = 3000$ K. Two different grid types – a Cartesian grid with 4096 control volumes (17 nodes in every spatial direction) and a non-Cartesian grid (O-grid) with similar number of control volumes (3584) are used for the fluid domain. Both, the Dirichlet–Neumann and the Neumann–Dirichlet methods for the thermal coupling are applied. The structure is discretized with 100 serendipity finite elements. In all cases 10^{-6} is used as a convergence criterion for both the thermal coupling and the CFD simulation. The results from the simulation are in a perfect agreement with the analytical solution. The maximal absolute error is in the field of 10^{-5} . For $T_s = 3000$ K the error is one order larger than for $T_s = 310$ K. Negligible difference in the accuracy of the two different Schwarz methods is observed.

In TC1 the mechanical FSI is very limited, as only very small deformations can take place. The second test case TC2, on the other hand, displays significant

Table 4.1: Material properties for TC1 and TC2

Property	TC1		TC2	
	Solid	Fluid	Solid	Fluid
E [Pa]	$2.1e^{11}$		$1.4e^{11}$	
ν	0.3		0.4	
α [K^{-1}]	$1.0e^{-5}$	$1.5e^{-4}$	$1.5e^{-5}$	$1.5e^{-4}$
κ [$W/(m \cdot K)$]	45.0	$2.5e^{-2}$	0.2	$0.6e^{-2}$
c [$J/(kg \cdot K)$]	450.0	1003	1700.0	4187
ρ [g/m^3]	7800	1.225	1	1
μ [$kg/(m \cdot s)$]	1	$1.8e^{-5}$	0.5	1

Table 4.2: Error of the simulation for different grids and boundary conditions for TC1

T_s	Grid	Dirichlet–Neumann		Neumann–Dirichlet	
		$E_{coupling}$	E_{global}	$E_{coupling}$	E_{global}
310	Cartesian	0.000036	0.000039	0.000038	0.000038
310	non–Cartesian	0.000053	0.000061	0.000047	0.000052
3000	Cartesian	0.00070	0.00073	0.00065	0.00069
3000	non–Cartesian	0.00091	0.00091	0.00089	0.00093

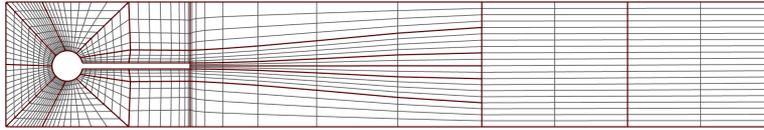


Figure 4.3: Grid configuration for TC2. Every 4th grid line is shown

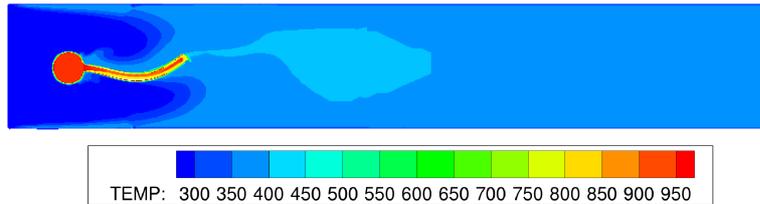


Figure 4.4: Temperature distribution (in [K]) in the middle of the geometry for TC2

displacements. The fluid problem is computed with the finite-volume solver FASTEST employing a second order Taylor-based interpolation (see Lehnhäuser and Schäfer [49]). Second order serendipity finite elements are used for the solid domain. The calculation is performed on 4 successively refined grids with approximately 12 500, 50 000, 200 000 and 800 000 control volumes, respectively. The finest grid is shown in Figure 4.3. The temperature distribution in a slice in the center of the geometry is shown in Figure 4.4. The reference results are taken from a simulation without any temperature (see Hron and Turek [39]), thus no quantitative comparison with the results from a full TFSI is possible. However, qualitatively we have a very good agreement. In both of the cases the structure oscillates periodically. The absolute velocity profile at periods $\frac{\pi}{2}$, $\frac{3\pi}{2}$ and 2π are plotted in Figure 4.5, Figure 4.6 and Figure 4.7. The difference

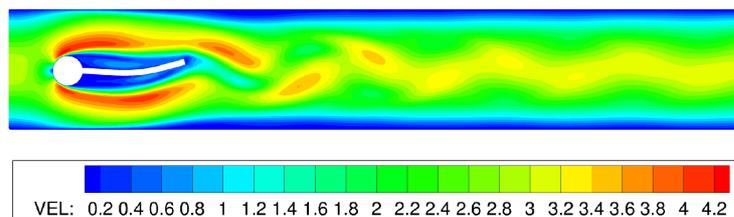


Figure 4.5: Absolute velocity (in [m/s]) at $\frac{\pi}{2}$ time period

of the frequency from reference and simulation results is less than 0.1%. The values for the drag and lift have the same courses over the time. Results for the

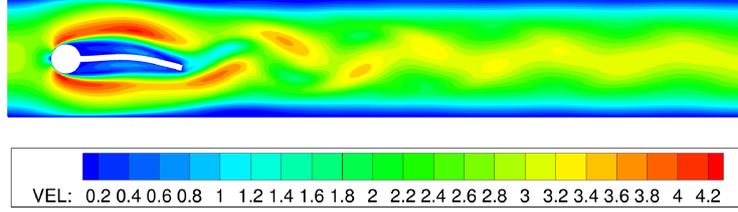


Figure 4.6: Absolute velocity (in [m/s]) at $\frac{3\pi}{2}$ time period

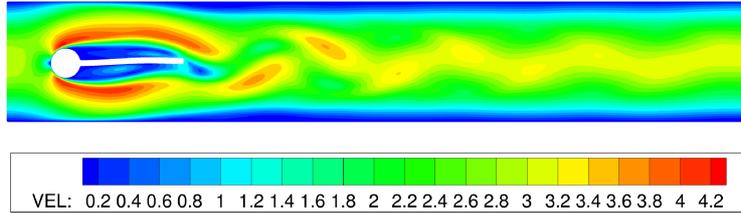


Figure 4.7: Absolute velocity (in [m/s]) at 2π time period

oscillation of the reference point A (see Figure 4.2), as well as the drag and the lift are shown in Figure 4.8. For the thermal coupling again the two variants of the Schwarz domain decomposition method are used and also here no significant difference in the results is observed.

4.2 Validation and verification of the turbulent thermal fluid–structure interaction

Verifying and validating the numerical procedure for turbulent thermal fluid–structure interaction is an important and complex task. In this work we investigate two different test configurations, for which reference results exist. The first one (TC3) is a rotational symmetric test case, composed of two opposed nozzles and a flat horizontal plate (see Figure 4.9). The second one (TC4) represents a simplified combustion chamber (see Figure 4.10). Here only the Neumann–Dirichlet alternating domain decomposition method is applied. The reason for that is the much better convergence of the flow solver in this case (discussion of this phenomena can be found in the next section). In author’s experience, also for the turbulent flows, the two different methods produce identical final results, but the Dirichlet–Neumann approach need much more computational time.

We start with a comparison between numerical and experimental results for

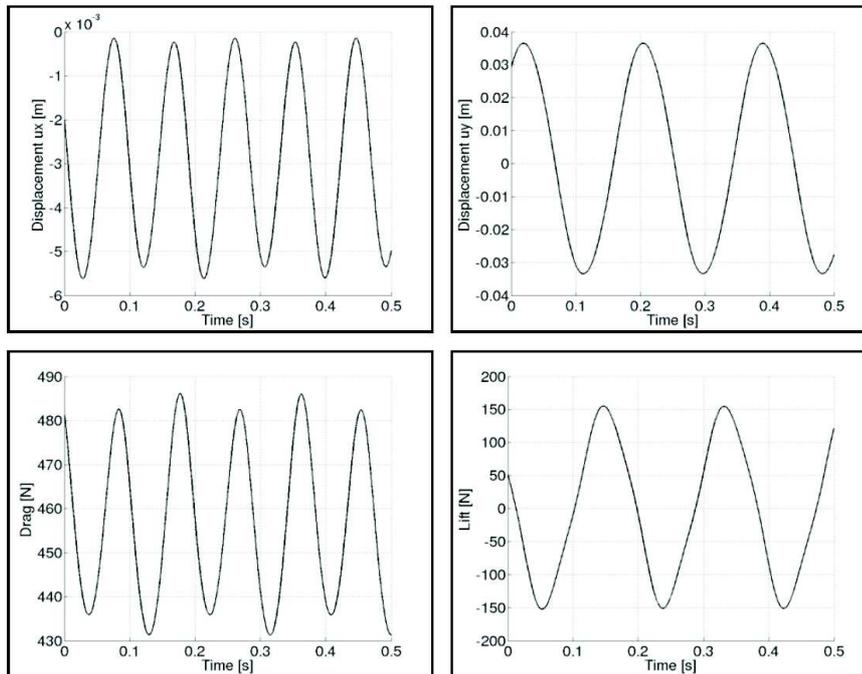


Figure 4.8: Displacement course of the reference point A, drag and lift for TC2

TC3. The investigated generic system consists of two opposed, vertical nozzles impinging onto a flat, horizontal copper plate. In the literature the heat transfer characteristics under single or multiple impinging turbulent jet flows are investigated numerically and experimentally by different authors during the last 30 years, without taking into account the effects of the fluid–structure interaction. Polat and Douglas [73] studied the numerical simulation of turbulent flow field of impinging jets over a surface. Nitin et al. [63] use a numerical simulation to predict flow and thermal fields and calculate the surface heat–transfer coefficient for a single hot air jet impinging on a cookie–like product. The results from the numerical predictions are compared with experimental measurements. Craft et al. [16] applied different turbulence models to the numerical prediction of the turbulent impinging jets discharged from a circular pipe and discussed the characteristics of the numerical approaches.

For the numerical simulation of TC3 exactly the same geometry which was used in the experiment from Brübach et al. [11] is employed (see Figure 4.9). A horizontal copper plate is heated from below by a non–reacting, turbulent air jet ($Re = 5000$, based on the bulk velocity and the jet’s diameter). To achieve well defined boundary conditions the plate is cooled by a turbulent cold jet from above. All outer walls of the domain are treated with outflow zero gradient boundary condition. The fluid is air and its material properties, as well as those for the copper plate, are shown in Table 4.3.

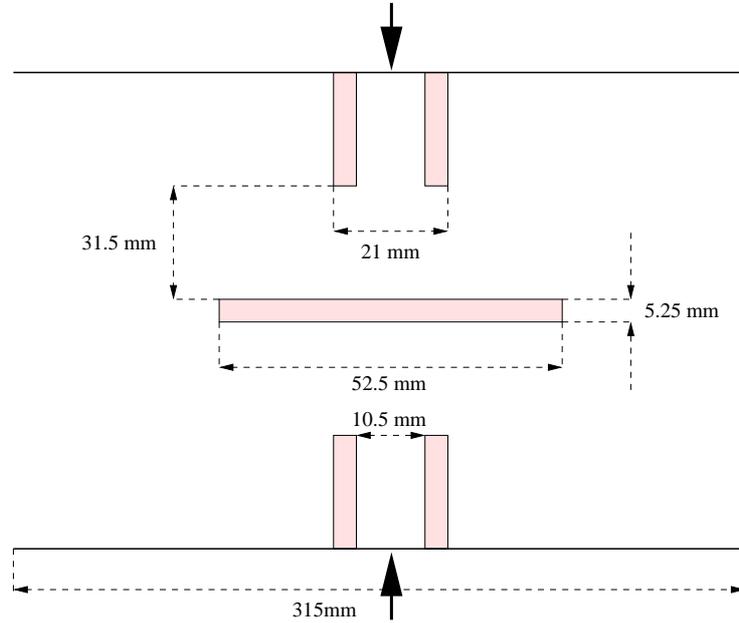


Figure 4.9: Geometry configuration for TC3

The radial and the axial length of the computational domain for the fluid are set to 301.5 mm and 130.25 mm, respectively. These dimensions are chosen in order to minimize the effects of the applied zero-gradient outflow boundary condition. This is important for the convergence of the calculation. As inlet boundary conditions a flow with 7.19 m/s axial velocity and 294 K temperature for the upper nozzle and a flow with 26.1 m/s axial velocity and 640 K temperature for the lower nozzle is used. The convergence criterion for the FSI and for the thermal coupling are set to 10^{-4} and 10^{-5} , respectively (see Eq. 3.4).

The fluid domain is discretized by 572 544 control volumes. As a low Reynolds $k - \varepsilon$ model is applied, an important characteristic is the normalized (dimensionless) distance to the nearest wall y^+ which should be in the interval $[0.1; 0.5]$ for the control volumes next to the solid walls (see [84]). Due to this restriction, the grid is refined near the solid walls, as shown in Figure 4.11. The structure is discretized with 12 000 linear solid hexahedrons.

The convergence of the whole coupling process is mostly dependent on the convergence of the flow solver. Strong underrelaxation especially for the turbulent kinetic energy k and for the dissipation ε (0.01 and 0.05 respectively) is employed. For reaching the convergence criterion of 10^{-4} for the velocity-pressure coupling around 10000 iterations are needed for the first 2 outer TFSI iteration. This number reduces dramatically for the remaining 5 TFSI iterations. Because of the characteristics of the chosen material, the deformations of the plate are extremely small ($\approx 10^{-2}$ mm) and they have no effect on the

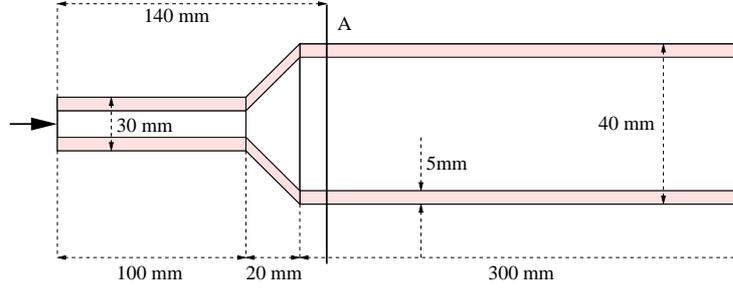


Figure 4.10: Geometrical setup for TC4

Table 4.3: Material properties used for TC3 and TC4.

Property	TC3		TC4	
	Solid	Fluid	Solid	Fluid
E [Pa]	$1.2e^{11}$		$2.1e^{11}$	
ν	0.34		0.3	
α [K^{-1}]	$1.7e^{-6}$	$1.5e^{-4}$	$1.1e^{-5}$	$1.5e^{-4}$
κ [$W/(m \cdot K)$]	401.0	$2.5e^{-2}$	45	$2.5e^{-2}$
c [$J/(kg \cdot K)$]	385.0	1003	450	1003
ρ [$J/(kg \cdot K)$]	8960	1.225	7800	1.225
μ [$kg/(m \cdot s)$]	1	$1.806e^{-5}$	1	$1.806e^{-5}$

convergence of the numerical procedure and on the heat transfer in the whole system. The temperature distribution in a 2D slice in the middle of the geometry can be seen in Figure 4.12. The results for the fluid field for the velocity in tangential direction (v_3), the pressure (p), the turbulent kinetic energy (k) and the dissipation (ε) are plotted in Figure 4.13.

The comparison of the simulation results with the experimental data starts with the temperature on the surface of the copper plate which can be seen in Figure 4.14. Due to the high heat conduction of the copper, the radial surface temperature profile is quite homogeneous. Except for one point the values from the numerical procedure are close to the experimental data and are always in the range of the shot-to-shot standard deviation. As expected, the temperature profile is absolutely symmetric in the simulation and almost symmetric in the experiment. The temperature maximum is located in the center of the plate, as the simulation tends to overpredict it, a phenomenon which we will clear later in this chapter.

Figure 4.15 compares the gas temperature in the middle of the geometry (radial position $r = 0$). The qualitative courses are similar. Near to the boundary layer the values from the simulation are slightly displaced from the experimen-

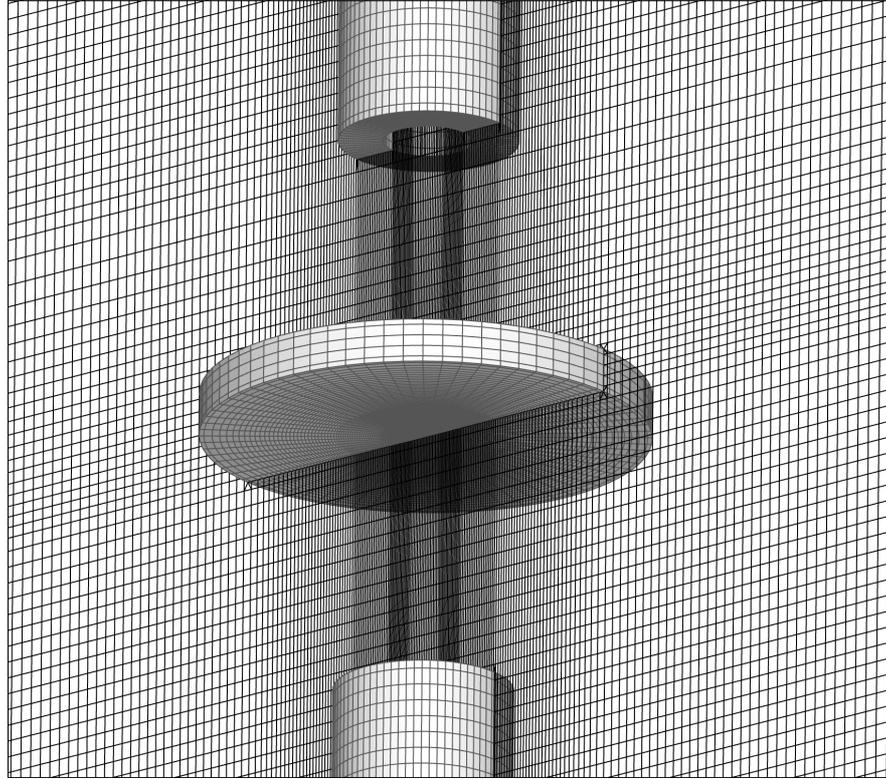


Figure 4.11: Discretization of the fluid domain.

tal data, probably due to the not exact positioning of the measurement devices (the difference is 0.1 mm). Near to the stagnation point one can observe that the numerically predicted temperature is higher than the measured one. This was observed also by the surface temperature comparison. In the authors experience the quality of the heat-transfer prediction in this region becomes worse as the Reynolds number increases. As reported in Craft et al. [16] the most probable reason for this is the overprediction of turbulent kinetic energy in the stagnation region, when eddy-viscosity models are applied. Both experiments and large eddy simulations (LES), show negative production of turbulent kinetic energy in the problematic zone. Such a phenomenon cannot be predicted by an eddy-viscosity model which always produces a positive production of turbulent kinetic energy. Craft et al. [16] proved that second-moment closure improved the results to some extent, depending on the model sophistication, primarily of the pressure strain term.

Figure 4.16 compares the fluid temperature along the radial position $r = 15$ mm. Away from the stagnation region the results from the simulation are in a very good agreement with the experimental data.

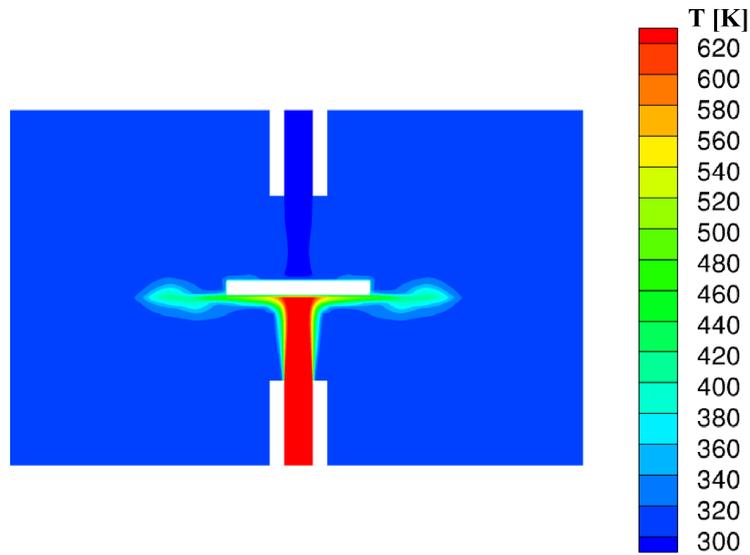
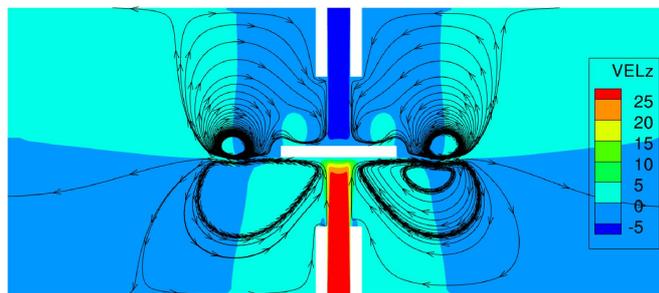


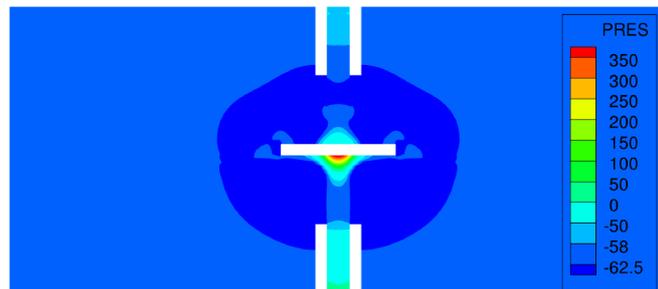
Figure 4.12: Temperature distribution (in [K]) in a center slice of the geometry

For the second test case TC4 a reference calculation with FASTEST with 2 flow regions exists where the deformations are not handled (explanation of the method can be found in Teschauer [99]). As the deformations which occur during the computation with the fully coupled scheme, are infinitesimal small and can be neglected, a comparison between the two calculations can be made. The geometry of the test configuration represents a simplified combustion chamber and is shown in Figure 4.10. The wall of the channel is 5mm thick steel. The fluid is set to be air. The material properties for both, fluid and solid, are shown in Table 4.3. Hot air with temperature 1000 K streams from the left with 4.9m/s . The Reynolds number in this case is $Re = 10\,000$. The temperature at the outside wall is set to 450 K . For the discretization of the fluid domain around $400\,000$ control volumes are used. The used turbulence model is the high-Reynolds $k-\varepsilon$ RANS model. The solid wall is discretized with 4080 brick finite elements.

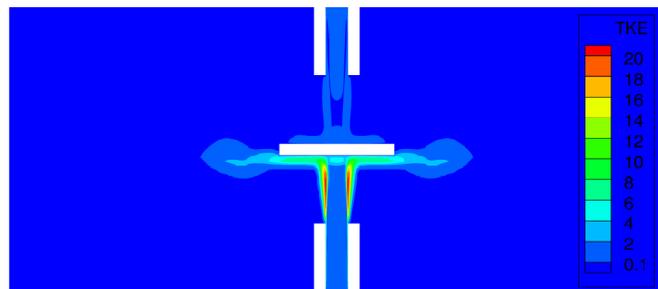
In Figure 4.17 the temperature distribution in a slice in the middle of the geometry is shown, as well as a plot of the temperature over the selected line A. As expected, by comparison with the reference solution no difference is observed. This, of course, is due to the fact that the material of the wall is so chosen, that no deformation appears. In author experience, even small deformation can have a great effect on the fluid flow, thus making the comparison meaningless.



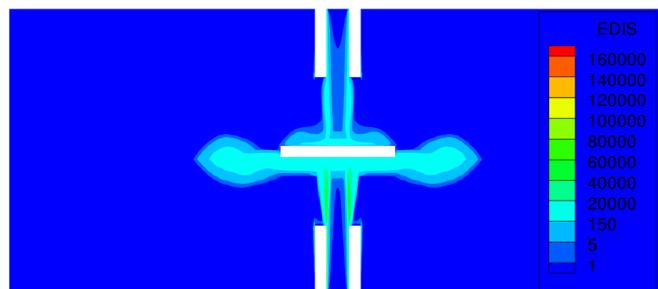
(a) Z velocity [m/s]



(b) Pressure [Pa]



(c) Turbulent kinetic energy [J/kg]



(d) Dissipation [W/kg]

Figure 4.13: Results for TC3

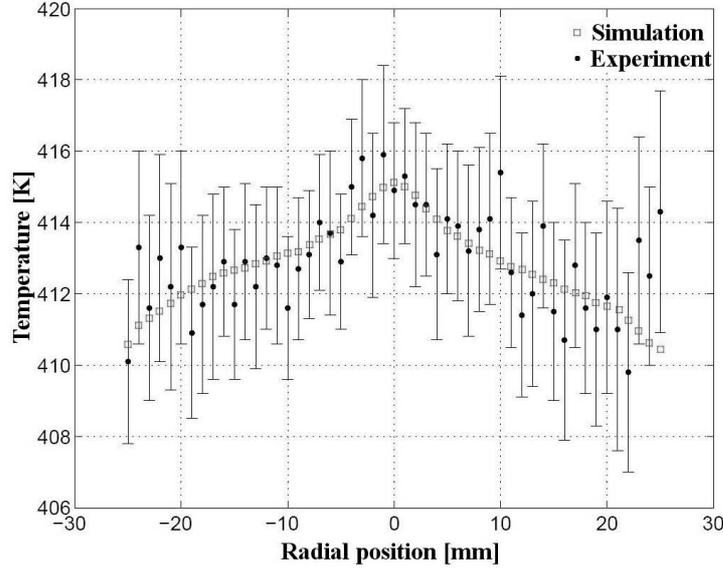


Figure 4.14: Comparison of the surface temperature. The error bars represent the shot-to-shot standard deviation for the experiment

4.3 Numerical properties of the coupling scheme

In this chapter we discuss the numerical properties of the coupling scheme. We first present results for the accuracy of the method and prove that it is second order accurate. Then the convergence characteristics of the algorithm are investigated and different possibilities for accelerating the calculation process are discussed.

In order to show that the results of the numerical simulation have 2^{nd} order of accuracy we use TC2, as the calculation is performed on 4 successively refined grids with approximately 12 500, 50 000, 200 000 and 800 000 control volumes, respectively. Using a Richardson extrapolation (see Schäfer (2006) [84]) for the two finest grids, a grid independent solution for the Nusselt number at the reference point A (see Figure 4.2) is found. By comparison with the grid independent solution, the relative solution error on all grids can be determined. In Figure 4.18 the relative error is plotted versus the grid spacing. The second order accuracy of the coupled method can be observed.

The investigation of the convergence properties of the scheme we start with the test cases TC1, TC3 and TC4, where the deformations of the structure are indivisible small and the convergence of the algorithm depends only on the alternating Schwarz decomposition method for the thermal coupling. Here again the Dirichlet–Neumann Schwarz coupling procedure is used. Table 4.4 shows

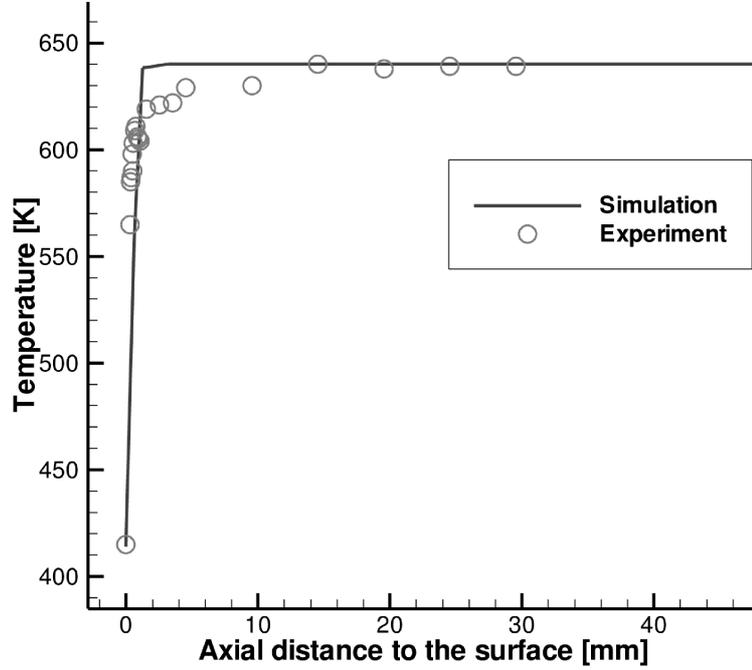


Figure 4.15: Comparison of the temperature at a radial position $r = 0$ mm.

the residuals of the thermal coupling $\|T^{m-1} - T^m\|_\infty / \|T^m\|_\infty$ for TC1 and TC3. TC1 is presented with four different grid configurations of the fluid mesh – Cartesian with 4096 CVs ($fine_1$), Cartesian with 512 CVs ($coarse_1$), non-Cartesian with 3586 CVs ($fine_2$) and non-Cartesian with 512 CVs ($coarse_1$). In all cases the boundary temperature for the solid is $T_s = 310$ K. A fine and a coarse grid with approximately 400 000 and 50 000 control volumes, respectively, are used for the simplified combustion chamber TC4. It can be seen that the rate of convergence is independent of the grid size and type, as also proved by Yang [110]. The number of outer TFSI iterations for reaching a defined convergence limit ϵ_{TEMP} stays almost constant for all investigated cases.

In Table 4.5 the values of the TFSI residual for TC1 with $T_s = 310$ K and $T_s = 3000$ K and TC3 with $T_{infl} = 640$ K and $T_{infl} = 860$ K are shown. It can be seen that the convergence rate of the scheme does not depend on the thermal gradient and it stays constant for all investigated cases. The only thing which influenced the number of TFSI iterations, needed for reaching a prescribed convergence criterion is the initial guess of the interface temperature. For TC1 with $T_s = 310$ K, for example, the initial value is closer to the solution than for the case where $T_s = 3000$ K and the difference can be seen in the residual of

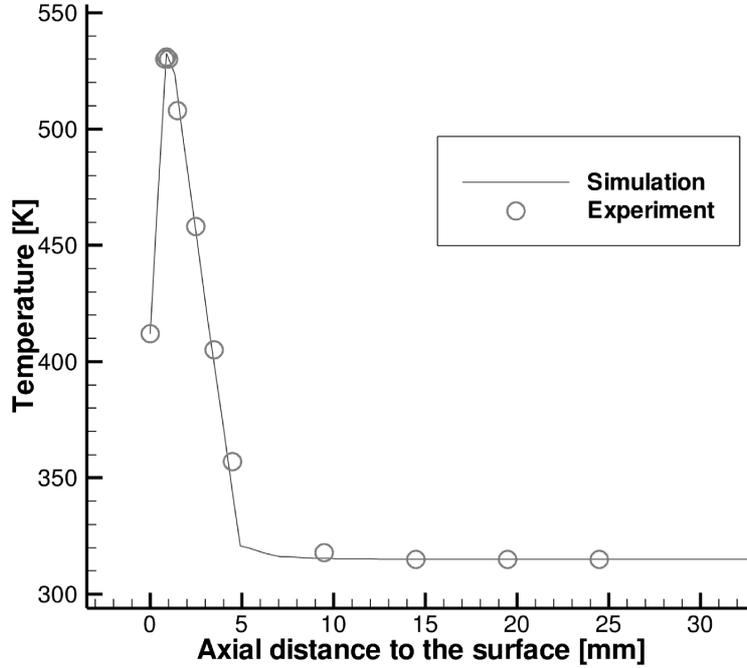
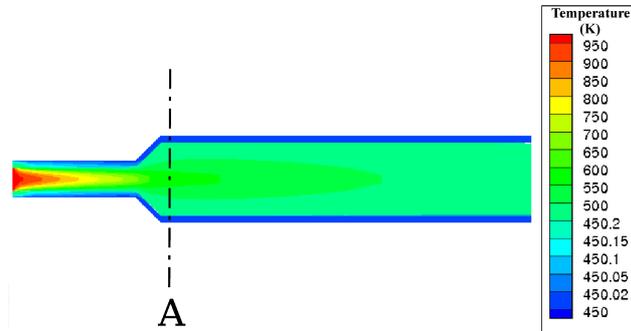


Figure 4.16: Comparison of the temperature at a radial position $r = 15$ mm.

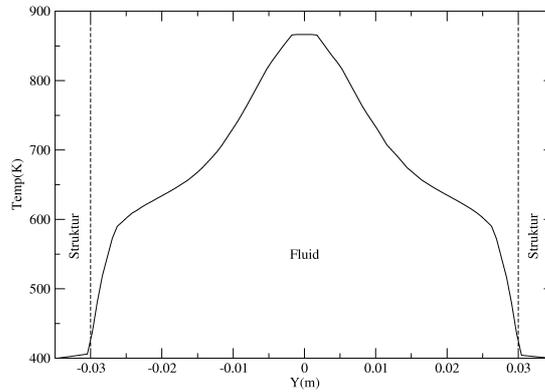
the first TFSI iteration.

In the second case TC2, where the structural deformations are playing a crucial role, the convergence of the coupling algorithm depends on both thermal and force–displacement coupling. In problems with large deformations, as here, normally much more iterations are needed by the mechanical coupling. For every time step around 300 TFSI outer iterations are needed for fulfilling the convergence criterion $\epsilon_{\text{FSI}} = 10^{-4}$ and for the thermal coupling only 6 iterations are needed. Thus the computations are strongly dominated by the force–displacement coupling algorithm. A deeper investigation of its convergence characteristics can be found in Heck (2008) [36].

For all of the considered problems most of the computation time is spent by the flow solver in particular when the flow is turbulent. In Table 4.6 we present results for the number of CFD iterations for every TFSI iteration for the test cases TC1 and TC3 in the case of Dirichlet–Neumann method for the thermal coupling. The results for the dual Neumann–Dirichlet method are shown in Table 4.7. In the second case a flux boundary conditions is used on the fluid–solid interface in the flow solver. As can be seen in this case much more SIMPLE iterations are needed to achieve convergence. The needed computation time is



(a) Temperature distribution (in [K]) in a slice in the middle of the channel



(b) Temperature distribution (in [K]) over the line A

Figure 4.17: Results for TC4

up to 4 times more than when the Dirichlet–Neumann approach is used. On the other hand, the same number of outer coupling iterations is needed by both of the methods. These observations show that the Neumann–Dirichlet coupling is, in fact, in no practical use.

With every new step less iterations are needed by the CFD solver. Much more iterations are needed when the grid is non–Cartesian.

4.4 Accelerating the computation

An important characteristic of a numerical scheme is its efficiency. Common problem for all (T)FSI applications is the great amount of computational time

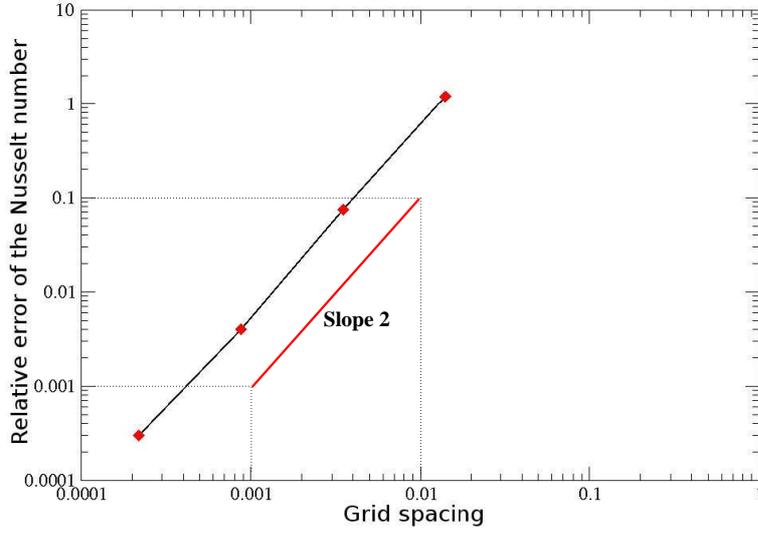


Figure 4.18: Convergence of the Nusselt numbers for successively refined grids

needed. In the first part of this chapter we already showed the advantages of ITFSI2 in comparison to ITFSI1 in this aspect. Here we will present and discuss several other techniques which were implemented and studied in this work.

4.4.1 T(FSI) predictor

As conclusions from our convergence study in the previous section we can state, that in case of finite deformations a great number of TFSI outer iterations are needed for reaching the convergence criterion. To reduce this number we use a FSI predictor which tries to predict the structural deformations before every time step (information can be found in Piperno [67]). Förster et al. [29] claimed

Table 4.4: Residual of the thermal coupling $\|T^{m-1} - T^m\|_\infty / \|T^m\|_\infty$ for different grids for TC1, TC3 and TC4 for every TFSI iteration it

it	TC1				TC3	TC4	
	fine ₁	coarse ₁	fine ₂	coarse ₂		fine	coarse
1	4.9	4.9	4.9	3	2	3	3
2	$6 \cdot 10^{-2}$	$6 \cdot 10^{-2}$	$6 \cdot 10^{-2}$	$6 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	$5 \cdot 10^{-2}$	$5 \cdot 10^{-2}$
3	$8 \cdot 10^{-4}$	$8 \cdot 10^{-4}$	$7 \cdot 10^{-4}$	$7 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$2 \cdot 10^{-4}$
4	$3 \cdot 10^{-5}$	$3 \cdot 10^{-5}$	$3 \cdot 10^{-5}$	$3 \cdot 10^{-5}$	$2 \cdot 10^{-6}$	$3 \cdot 10^{-5}$	$3 \cdot 10^{-5}$
5	$5 \cdot 10^{-6}$	$4 \cdot 10^{-6}$	$5 \cdot 10^{-6}$	$5 \cdot 10^{-6}$	$9 \cdot 10^{-8}$	$6 \cdot 10^{-6}$	$6 \cdot 10^{-6}$

Table 4.5: Residual of the thermal coupling $\|T^{m-1} - T^m\|_\infty / \|T^m\|_\infty$ for TC1 and TC3 for every TFSI iteration it

it	TC1		TC3	
	$T_s = 310$	$T_s = 3000$	$T_{infl} = 640$	$T_{infl} = 860$
1	4.9	1483	3	1
2	6.10^{-2}	18	5.10^{-2}	2.10^{-2}
3	8.10^{-4}	2.10^{-1}	1.10^{-4}	1.10^{-4}
4	3.10^{-5}	3.10^{-3}	3.10^{-5}	7.10^{-6}
5	5.10^{-6}	1.10^{-5}	6.10^{-6}	1.10^{-5}

Table 4.6: Number of CFD iterations for different grids for TC1 and TC3 for every TFSI iteration it . Dirichlet–Neumann thermal coupling applied.

it	TC1				TC3
	fine ₁	coarse ₁	fine ₂	coarse ₂	
1	2071	594	9950	2495	13435
2	2069	591	9939	2492	13428
3	1305	373	6174	1547	12761
4	534	153	2351	588	1593
5	30	11	66	21	340

that different predictors can lower the stability limit of the coupling scheme, i.e. the algorithm becomes more unstable and stronger underrelaxation for the force–displacement coupling is needed. However Yigit [113] showed that the acceleration due to a second order predictor can be up to 8 times in comparison to a calculation where no predictor is applied.

In this work three predictor techniques with different orders were implemented not only for the force–displacement coupling, but also for the thermal coupling:

$$\begin{aligned}
 d^{k+1} &= d^k && 0^{th} \text{ order,} \\
 d^{k+1} &= d^k + (d^k - d^{k-1}) && 1^{th} \text{ order,} \\
 d^{k+1} &= d^k + \left(\frac{3}{2}(d^k - d^{k-1}) - \frac{1}{2}(d^{k-1} - d^{k-2})\right) && 2^{nd} \text{ order,}
 \end{aligned} \tag{4.6}$$

where $k+1$ is the number of the next TFSI outer iteration and d the deviance of the current calculated values of interest (interface distortion, boundary temperature or boundary thermal flux) and their initial values. The number of TFSI iterations for TC2 and TC4 when different preconditioners are applied is shown in Table 4.8. The convergence criteria for both thermal and force–deformation coupling (see Eq. 3.4) are set to 10^{-6} . In all cases the underrelaxation param-

Table 4.7: Number of CFD iterations for different grids for TC1 and TC3 for every TFSI iteration it. Neumann–Dirichlet thermal coupling applied.

it	TC1				TC3
	fine ₁	coarse ₁	fine ₂	coarse ₂	
1	4121	936	18341	3987	19211
2	3978	920	17534	3389	19101
3	1901	531	10100	1932	17671
4	864	349	3989	876	3004
5	43	19	110	33	426

eter for the mechanical coupling α_{FSI} is set to 0.3. In case of the 1st order predictor the scheme suffers from instabilities and stronger underrelaxation is needed. That is why the number of needed TFSI iterations is almost the same as in the case of 0th order prediction. Such problems are not observed by applying the second order prediction technique. The acceleration achieved in this case is around 6 times. The acceleration factor due to predictor for the thermal coupling depends on the character of the problem. In cases where the temperature is strongly time dependent the computation time is reduced up to 5 times, as instead of 6–10 TFSI outer iterations only 2–3 are needed.

Table 4.8: Number of TFSI iterations by different preconditioners for TC2 and TC4

Test case	0 th order	1 st order	2 nd order
TC2	298	290	48
TC4	7	6	3

4.4.2 Uneven time coupling with estimation of the distortions

The thermal fluid–structure interaction is a typical multi–field problem. Every physical field (fluid or solid) is characterized by different temporal and spatial scales. By the chosen numerical partition approach both computational domains are handled separately. As it have been already discussed a block structured finite–volume mesh is needed by the FASTEST solver and an unstructured finite–element grid is used by FEAP. The two different spatial discretizations can be independently chosen so that they perfectly fit to the underlying numerical methods and the problem of interest. For example, several million control volumes can be needed for a LES calculation of a fluid flow in a channel but only

10 beam finite elements can be enough for a proper simulation of the dynamics of a solid beam in this channel.

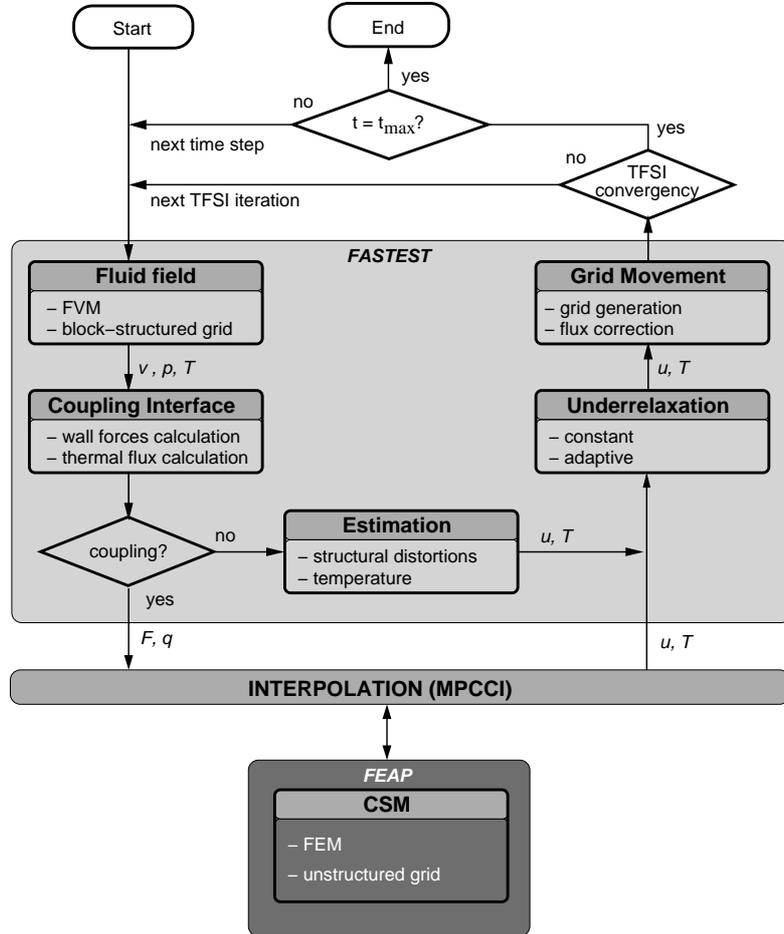


Figure 4.19: Flow chart of the coupling procedure ITFSI3

Similar problems exist also for the temporal discretization. Normally a very small time step is needed for properly resolving the flow motion scales, while for such a time period almost nothing happens to the structure. This observation can be used for optimizing the TFSI calculation. The main idea is to have two different time steps: Δt_f for FASTEST and Δt_s for FEAP, as $\Delta t_s = k_t \Delta t_f$. The coupling then is realized every k_t time steps. For the time steps where no interface transfer is performed, an estimation of the structural distortions and the temperature is applied. In fact here we can use the already discussed TFSI predictors' techniques. The schematic view of this modified procedure (ITFSI3) is shown in Figure 4.19.

Table 4.9: Material properties used for TC5.

Property	Solid	Fluid
E [Pa]	$1.2e^{11}$	
ν	0.34	
ρ [J/(kg · K)]	8960	1.225
μ [kg/(m · s)]	1	$1.e^{-9}$

In order to show the advantages of the scheme ITFSI3 the test geometry TC5 is used (see Figure 4.20). The test case is a three dimensional channel with a deformable obstacle inside it. We concentrate ourselves only on the structural deformation and no temperature is taken into account. The material properties are shown in Table 4.9.

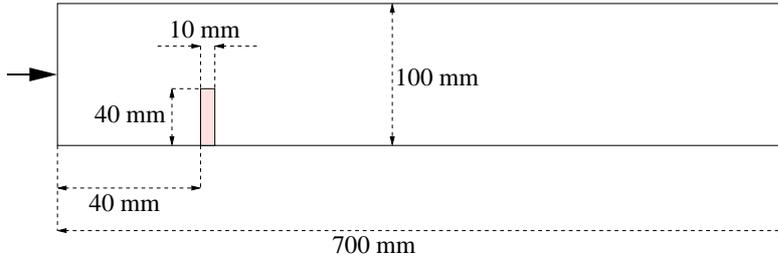


Figure 4.20: Geometry of the test configuration TC5

First the properties of the method are studied for a laminar flow ($Re = 100$). 4212 CVs are used for the spatial discretization of the fluid domain. The structure is discretized by 800 linear solid hexahedron finite elements. In order to avoid the locking effect enhanced strain formulation is employed. The time step size in the fluid code is fixed to 0.001 s. Three simulations with three different values for k_t are performed: $k_t = 3$, $k_t = 4$ and $k_t = 5$, i.e. the time step in FEAP is set to $\Delta t_s = 0.003$ s., $\Delta t_s = 0.004$ s. or $\Delta t_s = 0.005$ s. For every estimation's scheme we define E_x as the maximal difference in percent between the structural distortions calculated with the ITFSI2 and ITFSI3:

$$E_x = \|(u_{ITFSI3} - u_{ITFSI2})u_{ITFSI2}/100\|_{\infty}, \quad (4.7)$$

where x is the order of the scheme, i.e. $x = 0, 1, 2$.

The CPU times, as well as the maximal deviation E_x for 100 time steps simulations with different estimation of distortions schemes and coupling periods (k_t) are shown in Table 4.10. Employing a higher order prediction for the distortions leads, as expected, to better results, i.e. $E_2 < E_1 < E_0$. The computational time on the other hand stays almost constant. Higher order

Table 4.10: CPU time (in seconds) and E_x for 100 time steps with ITFSI2 and ITFSI3 with different estimation of distortion for TC5

k_t	ITFSI3						ITFSI2
	0 th order	E_0	1 st order	E_1	2 nd order	E_2	
3	3864	0.05	3856	0.012	3861	0.0017	4680
4	3556	0.07	3554	0.027	3558	0.0021	4680
5	3581	0.15	3585	0.047	3584	0.0033	4680

schemes can be also easily implemented but as in FASTEST the highest realized time scheme is of second order no greater improvement will be achieved.

The more often the exchanging procedure is applied the less is the average difference to the reference results calculated with ITFSI2. The computational time is however not always less. As can be seen from Table 4.10 the case $k_t = 5$ needs more CPU time than $k_t = 4$. This is due to the fact that the convergence of the SIMPLE method is worse and also more outer iterations are needed by every coupling time step.

Similar results are obtained when the flow is in the turbulent regime. The simulations are done for $Re = 120\,000$ and the standard $k-\varepsilon$ model is employed. The fluid domain is discretized with 9 792 control volumes, thus the y^+ stays in the interval (30, 300). The same number of finite elements as in the laminar case is used. Only second order scheme for the estimation of the distortions is applied. The time step for the fluid solver is set to $\Delta t_f = 0.0005s$. On the initial geometry a standard fluid simulation (without FSI coupling) is first performed and its results are used as an initial solution for the coupled calculation. The CPU times and E_2 for 100 time steps are shown in Table 4.11.

Table 4.11: CPU time (in seconds) and E_2 for 100 time steps with ITFSI2 and ITFSI3 and turbulent flow

k_t	ITFSI3		ITFSI2
	CPU time	E_2	
3	7036	0.0018	8510
4	6458	0.0048	8510
5	6686	0.0074	8510

Finding the best time step sizes for both computational domains is problem dependent and is a question of experience. Employing the ITFSI3 scheme allows the user to choose different values, considering the specific of the two domains. In such a manner for different problems a great acceleration can be achieved.

4.4.3 Multigrid technique

As we have already commented, most computational time is needed by the pressure–velocity coupling procedure in the flow solver FASTEST. Common practice for reducing this time is parallelizing the code (the approach will be discussed in the next section), or using a multigrid method. Two calculations, with and without multigrid for TC4 are taken into account. In the first case a single grid with 400 000 control volumes is used and in the second a multigrid method with 3 grids with 400 000, 50 000 and 6 250 control volumes respectively is applied. In Figure 4.21 the number of fine grid iterations for one time step which contains 6 TFSI iterations is shown. The convergence criterion in both cases is set to 10^{-5} . Around 2300 fine grid iterations are needed in the single–grid calculation. For the multigrid case this number is significantly reduced to only 800 which is a factor of almost 3.

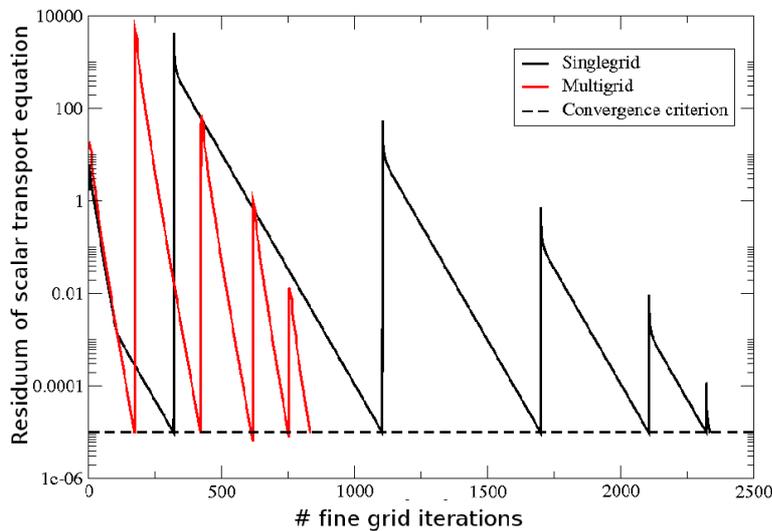


Figure 4.21: Acceleration due to multigrid calculation in the flow solver for TC4

Comparing the real computational times of the whole coupled simulation with and without multigrid technique the difference is not so great. A certain computational effort is needed for the coarser grid levels. Also no acceleration for the structural solver is employed. However, approximately 2 times less CPU time is needed by a multigrid computation.

As a conclusion we can state that the presented results show a great speedup of the calculation, nevertheless that the multigrid method is used only from the fluid code. It looks that a very promising approach should be a multigrid technique for the whole coupling simulation.

4.4.4 Parallelization

As discussed in the previous chapters, for the parallelization of the thermal fluid structure interaction we choose to perform a multiprocessor calculation only for the flow solver. The finite–element code FEAP is working sequentially and for the coupling two different techniques are implemented. The first we call “1 to 1” as here only one processor gathers the needed coupling information and takes part in the data exchange via MpCCI. With the further development of the MpCCI another option becomes also possible. First in MpCCI 2.0 Professional a new alternative communication scheme based on coupling servers was introduced. Instead of having all codes in one big MPI world, in the new approach each code uses its own MPI (MPICH or the native MPI of the platform where it is running) and is started independently of the other codes. In such a manner, it is possible for all processors on the coupling interface to communicate directly via MpCCI with the one processor FEAP job (“ n to 1” approach).

The simulations are carried out at the Hessian High Performance Computer (HHLR). The super computer has 15 shared memory multi–processor nodes (SMP), each with 32 Power6-CPU’s and 128 GB memory. The benchmark test case TC2 is used again in order to investigate the efficiency of the both parallel schemes. The fluid domain is discretized with a fine grid with almost 1 000 000 control volumes. Calculations with 1, 2, 4, 8, and 16 processors are performed. A special block splitting is used in order to assure 100% load balancing, as well as the same number of fluid–structure interface control volumes for every CPU. This means, that every processor should take the same part in the communication procedure. The results for the speed–up and for the parallel efficiency of both schemes are shown in Table 4.12.

Table 4.12: Speed–up and parallel efficiency for the two parallelization strategies

number of processors	S_P		E_P	
	1 to 1	n to 1	1 to 1	n to 1
2	1.86	1.87	93%	93.5%
4	3.34	3.38	83.6%	84.5%
8	6.06	6.15	75.7%	76.8%
16	9.96	10.15	62.2%	63.4%

The efficiency of the “ n to 1” scheme is with 1%-2% better than that of the “1 to 1” approach, as for more processor jobs the difference grows. This is due to the fact that the greater the number of CPU’s is, the more internal communication for collecting the whole interface information is needed. Something more, the test configuration is so constructed, that it is “the worse case” for the “1 to 1” method. In the practice it is almost not possible to have all processors on the coupling interface. In such cases the advantages of the “ n to 1” scheme will be relatively small. If however, the structural mechanics solver works also parallel

this approach will be without any competition. The fact that FEAP works on only 1 processor is the reason, why for jobs with more processors the parallel efficiency of the simulation drops significantly.

4.5 LES on moving grids

In most of the engineering fluid–structure interaction applications the flow of interest is in the turbulent regime. We have already shown results for the validation and the verification of the coupling procedure with different RANS methods. We also discussed some of the problems by using these methods for thermal simulations. Yigit [113] showed that employing a RANS approach for FSI simulations with great structure deformations produces results which are not in agreement with the experimental data. A standard way to overcome these problematic issues is to use a Large–Eddy Simulation (LES) or hybrid LES/RANS models. For a detailed discussion on these approaches see Kadavelil et al. [43].

In this section we present results for applying the dynamic Smagorinsky model in the context of moving grids. No temperature is taken into account. A fully coupled FSI simulation is also not in the scope of this thesis. Our interest here is concentrated on performing a one way explicit FSI coupling for a case with finite deformations. This means that prescribed deformations are interpolated on the fluid grid once at every time step and then a standard LES simulation is performed. For the determination of the Smagorinsky constant C_S the dynamic Germano procedure is applied after every change of the geometry. No back coupling (i.e. interface force calculation and exchange) is employed. The aim of this study is to show that the implemented TFSI parallel concept can be used with LES calculations on a super computer, without any restrictions.

4.5.1 Test configuration

The test configuration used in the experimental study of Gomes and Lienhart [33] is employed also in this work and is denoted here as TC6. The structure consists of a 0.04 mm thick stainless steel membrane attached to a aluminum cylindrical front body. At the trailing edge of the flexible sheet a rectangular stainless steel mass is located. Both the rear mass and the front body are considered rigid. The structure is free to rotate around an axis, located in the center point of the front cylinder. The structure is placed in the middle of a wind channel with length (x -direction) $L = 750$ mm, width (y -direction) $W = 240$ mm and height (z -direction) $H = 88$ mm. The center of the cylindrical front body is located 155 mm downstream from the inflow boundary of the physical domain. The detailed dimensions are represented in Figure 4.22.

The material properties of the chosen fluid are shown in Table 4.13.

As inflow boundary condition a block profile with a constant velocity $u_x = 0.68$ m/s in the streamwise direction is used. The correspondent Reynolds number, based on the diameter of the front cylinder, is about $Re = 15\ 000$. At

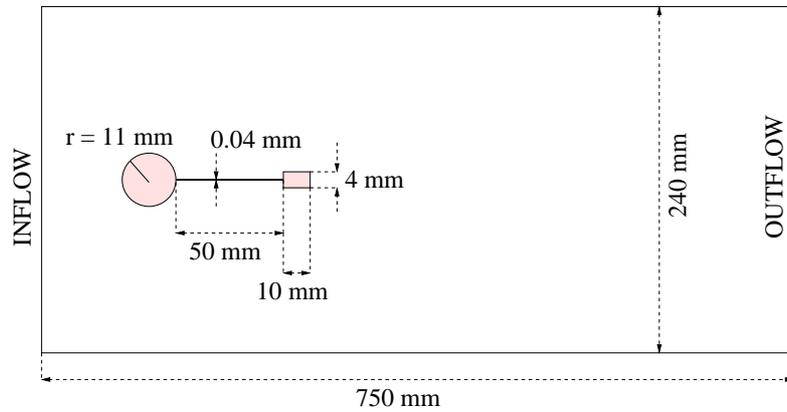


Figure 4.22: Geometry setup for TC6

Table 4.13: Material properties used for TC6.

Property	Value
ρ [$J/(kg \cdot K)$]	998
μ [$kg/(m \cdot s)$]	$9.7e^{-3}$

the right end of the geometry a zero-gradient boundary condition is applied. All walls are treated as no-slip walls. In z -direction periodic boundary conditions are employed. As in the experiment no deformations of the structure occur in this direction.

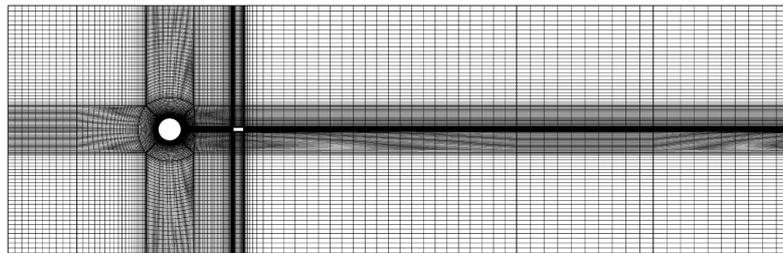


Figure 4.23: Discretization grid for TC6

The fluid code is explicitly one way coupled (1 iteration per time step) with the solid solver, in order to get a periodical movement of the structure. A prescribed periodic displacement of the trailing edge of the structure is set as a boundary condition in FEAP and no force coupling with FASTEST is employed. Exactly as in the experiment, the structure oscillates with frequency 4.45 Hz.

For the discretization of the fluid domain 1 069 056 CVs are used. The mesh of the initial geometry is shown in Figure 4.23. The time step is set to $2e - 4$, thus $CFL \approx 1$. No underrelaxation for the structural distortions is applied, as for the chosen time step size the method shows to be stable. CDS is chosen for the interpolation of the convective and the diffusive fluxes.

4.5.2 Grid distortions

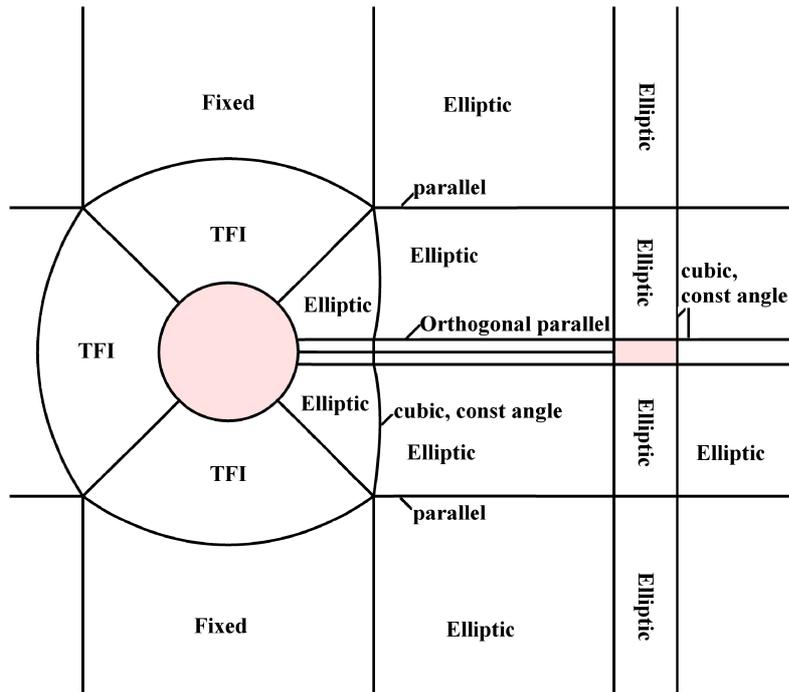


Figure 4.24: Block structure and grid distortion concept

In the case of moving boundaries and LES calculation special attention has to be paid on preserving the grid quality. As large three dimensional deformations occur this task is not trivial. The remeshing techniques, implemented in this work which were discussed in 3.6 are employed here. The block structure for the discretized fluid domain, as well as the concept for the grid distortions is shown in Figure 4.24.

For the blocks adjacent to the structure the elliptic method is employed for the faces and for the whole blocks. The walls which lay opposite to the coupling boundary are moved parallel and orthogonal to the interface. As those are used as boundary conditions for the following 3D elliptic grid generation the quality of the final grid (angles and distances between the nodes) is preserved on a certain appropriate level.

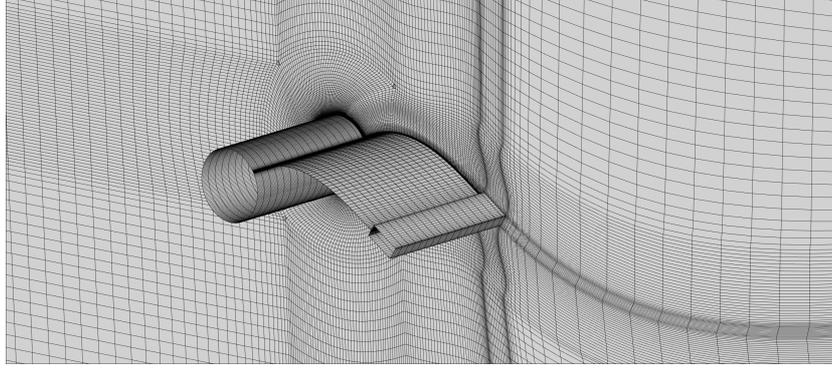


Figure 4.25: Overview of the distorted mesh

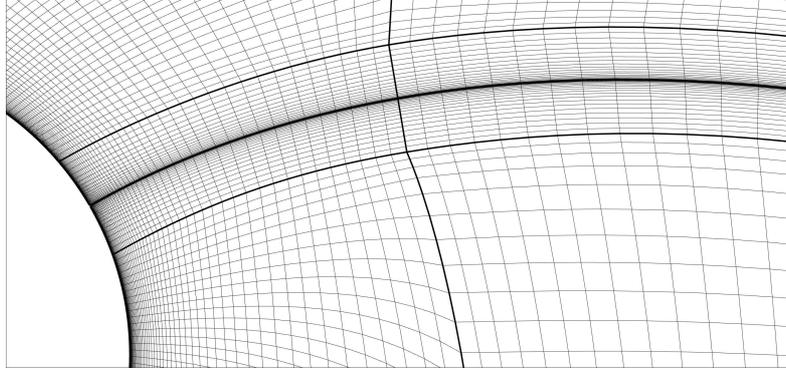


Figure 4.26: Distorted grid around the rotating cylinder

The discretization mesh for a chosen time step is shown in Figure 4.25. A detailed view of the grid around the front body cylinder and the stainless steel mass, attached at the trailing edge of the elastic structure is represented in Figure 4.26 and Figure 4.27. It can be seen that the mesh remains orthogonal in the near wall regions, nevertheless that no extra boundary conditions for the elliptic grid generation are employed as in the work of Yigit [113].

4.5.3 Results and discussion

Before starting the explicit one way coupling simulation an LES calculation with the default geometry is performed in order to get a fully developed turbulent initial flow field. The time series for the velocity in the streamwise direction in 3 different points in the fluid domain are shown in Figure 4.28

56 200 time steps were performed, thus 50 periods were simulated After which the calculation was manually canceled. In Figure 4.29 the velocity field in 3 different time-phase angles which is normed by the bulk velocity v_{bulk} is shown. The pressure field for the same periods ($\frac{\pi}{2}$, $\frac{3\pi}{2}$ and 2π) is represented.

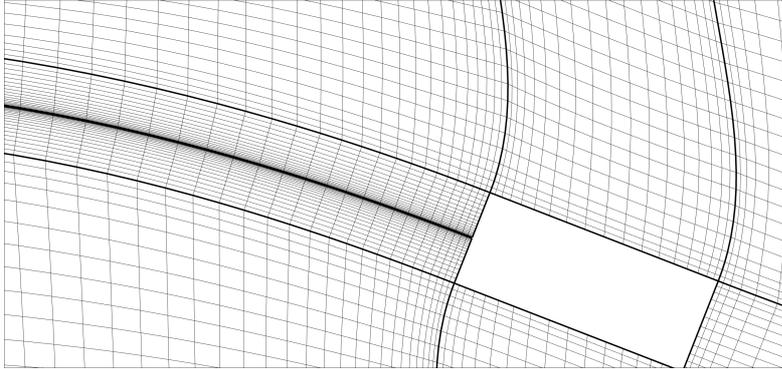


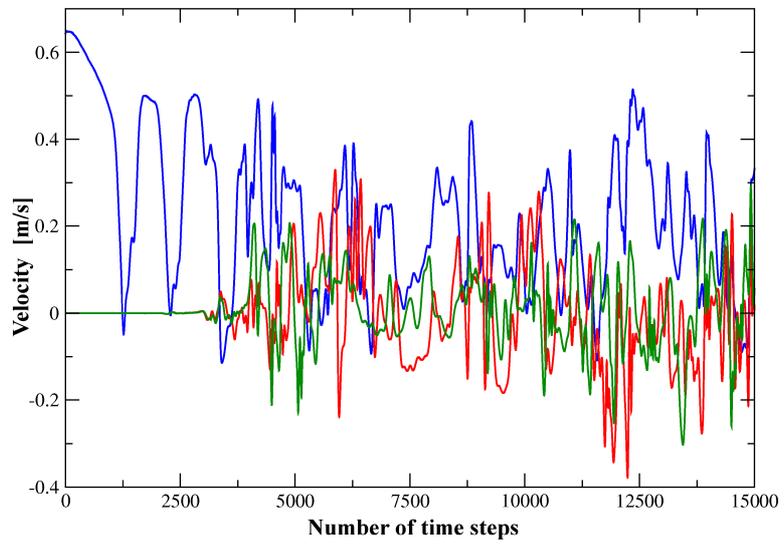
Figure 4.27: Distorted grid around the steel mass

No quantitative comparison with the experimental data is possible, as the prescribed fluctuations do not coincide with the real structure movement. This is due to the fact that only the trailing edge coordinates as well as the rotation angle of the front body cylinder were measured. Thus, without having a full coupled simulation the structure deformation is far away from the reality.

The most important conclusion from the performed simulation is that the TFSI framework with the implemented grid distortion procedure is fully functional and can be used without any restrictions with different turbulence models. The biggest problem, however is the great computational time which is needed. Around 2000 SIMPLE iterations are needed at every time step of the discussed computation. For one time step in this case around 10 minutes computational time on 14 processors is needed. However, one should keep in mind that this is only an explicit one way coupling with no underrelaxation of the structural distortions.

For a fully coupled simulation the number of outer iterations depends on the chosen underrelaxation FSI parameter. In author's experience using the Aitken method leads to fast divergence of the whole computation and only a strong constant underrelaxation ($\alpha_{FSI} \approx 0.005$) can damp the effect of the artificial added mass. Thus, around 2000 outer FSI iterations are needed in this case. Additionally the number of SIMPLE iterations stays almost constant after every FSI coupling iteration. In such a manner around 10 days are needed for one time step which made such a simulation not possible.

Acceleration due to techniques, discussed in the previous section, can reduce the needed computation time twice. However it is still not acceptable. Another idea is to implement and investigate the characteristics of a different time discretization schemes. For example the Crank–Nicolson method, applied to this case needs only 7 iterations per time step in the coupling procedure.

Figure 4.28: Time series for v for TC6

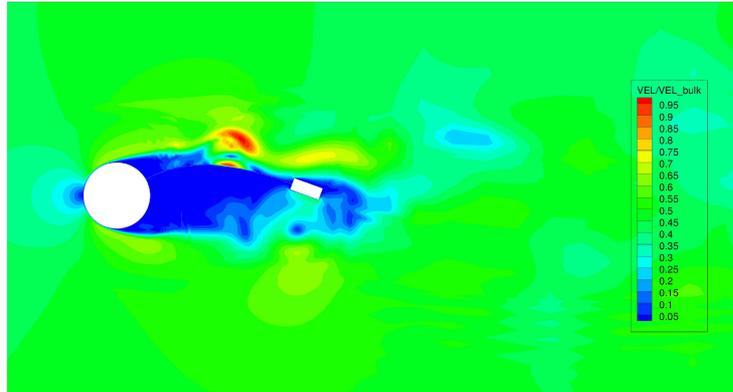
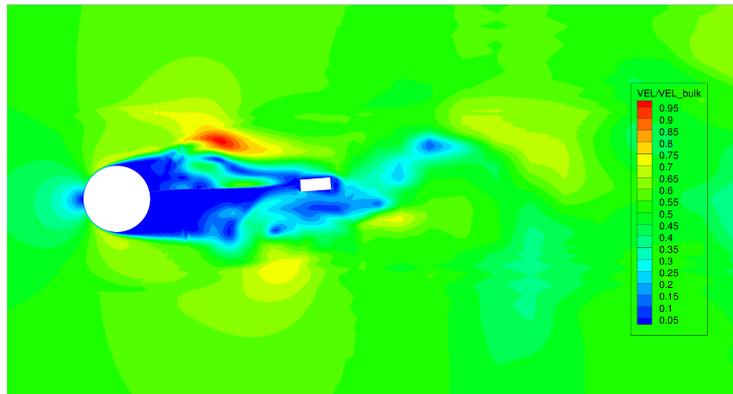
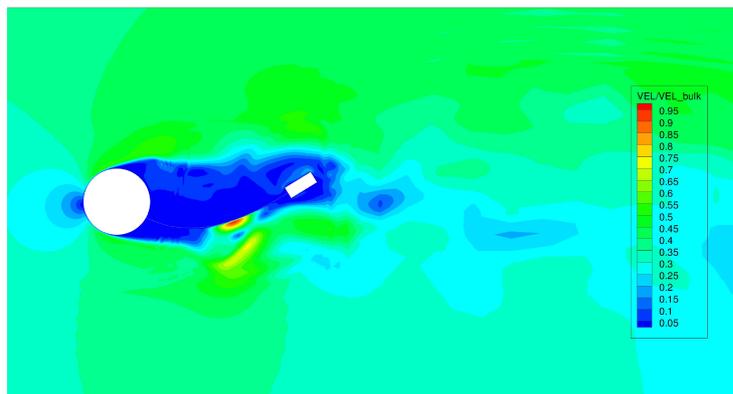
(a) phase angle $\frac{\pi}{2}$ (b) phase angle $\frac{3\pi}{2}$ (c) phase angle 2π

Figure 4.29: Normed velocity profile for TC6

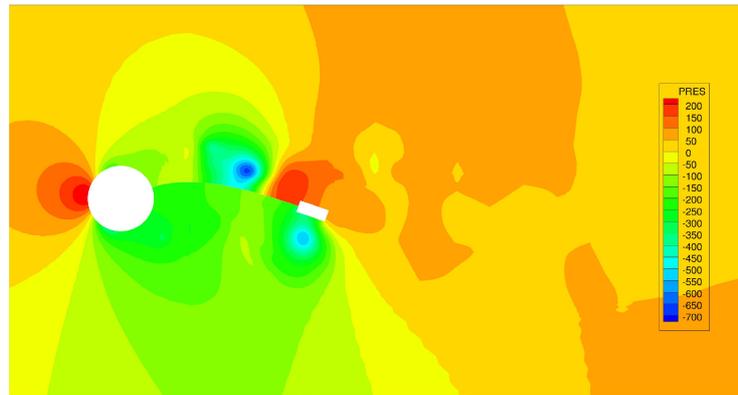
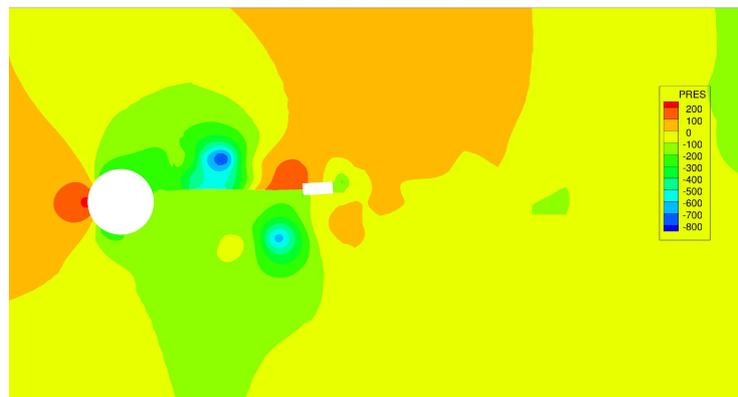
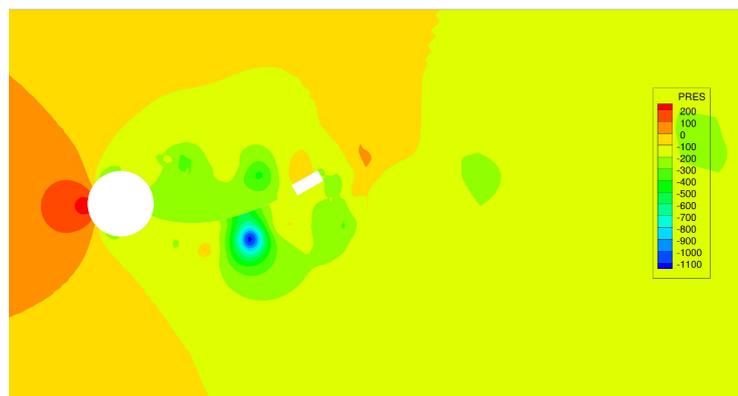
(a) phase angle $\frac{\pi}{2}$ (b) phase angle $\frac{3\pi}{2}$ (c) phase angle 2π

Figure 4.30: Pressure for different time–phase angles for TC6

Chapter 5

Conclusion and Outlook

The aim of this thesis was to develop and then to implement a framework for numerical simulation of fully coupled thermal fluid structure interaction, a phenomenon which occurs in a large set of engineering problems. As for such calculations normally a great amount of computational time is needed, an important aspect of this work was the optimization of the numerical procedure.

For the mechanical fluid–structure interaction coupling an implicit partitioned approach was chosen as it allows to use specifically designed and highly optimized codes on different fields and offers significant benefits in terms of efficiency. Furthermore it can be easily applied to various FSI problems which was the crucial aim of this thesis. For the flow field the in–house program FASTEST [64] was used. The open–source finite–element solver FEAP [98] was employed for the solid domain. The interpolation of the property values between the both numerical grids was done by the quasi–standard coupling interface MpCCI [83]. As a basis the work of Glück [32], Yigit [113] and Heck [36] was used.

The partitioned approach was also chosen for the thermal coupling. The Neumann–Dirichlet and Dirichlet–Neumann Schwarz domain decomposition methods were employed. For their realization a second–order differentiation technique, as well as variational consistent approach for the interpolation of the thermal flux in FASTEST and FEAP, respectively, were implemented. Both codes were extended with thermal flux coupling boundary conditions.

For the adaption of the CFD mesh to each current structural geometry, especially in the case of finite deformations, a new concept for the management of the grid distortions was developed and implemented in the flow solver FASTEST. Linear interpolation methods, trans–finite interpolation, as well as elliptic grid generation in two and three dimensional space were employed. Special techniques for handling the edge and face movement, like parallel forwarding of displacements and orthogonal wall shifting were used in order to keep the mesh quality which is an important factor by the numerical simulation of turbulent flows. Several test cases proved the efficiency and the robustness of the scheme.

The coupling framework was verified and validated for a laminar flow by a

simple test case with an analytical solution and by a benchmark test case. A test configuration for which experimental data exists was used together with different RANS models for validation of the scheme when the flow of interest is the turbulent regime. In all cases the algorithm showed very good results.

The numerical properties of the method were studied by a variety of test cases. We demonstrated that the Dirichlet–Neumann coupling which uses a differencing technique for the interpolation of the thermal boundary flux, produces the same results as the dual Neumann–Dirichlet method for all investigated cases. We discussed the convergence rate of the coupling algorithm and proved that it does not depend on the numerical grid and the boundary conditions. We also explained the influence of the chosen coupling scheme on the convergence of the finite–volume fluid solver and showed that the variationally consistent Neumann–Dirichlet coupling approach needs much more computational time.

Several techniques for improving the convergence characteristics and for accelerating the computational process with the scheme were implemented. The fluid code was extended by structural and thermal predictors with different orders. In the flow solver a multigrid method on moving grids was employed. Most of the test cases were carried out at the Hessian High Performance Computer (HHLR). For that reason the whole framework was parallelized. For different test cases the parallel efficiency of the program package went up to 95% in the case of a two processor calculation and up to 65% for a simulation with 16 processors.

The parallelization of the framework allowed employing a dynamic LES model in the flow solver. A calculation with a benchmark test case showed that such simulations could be very promising.

In summary, it can be noticed, that the FSI environment which was used in the Chair of Numerical Methods in Mechanical Engineering was extended by a thermal coupling procedure and the whole process was optimized. As a result a powerful program package that can be used for a complex real engineering problems was built.

However, we can not consider the work in this field as finished. A lot of other important aspects concerning thermal–fluid structure interaction are worth to be studied. At first place a dynamic model for the temperature calculation by an LES simulations can be applied. Different turbulence models as hybrid models or second closure RANS model can be implemented and investigated in terms of TFSI.

From a practical point of view an important task is the use of the developed program package for a complete engineering simulation. Together with a combustion model in the flow solver, the behavior of a jet engine combustion chamber or turbine can be predicted. The framework can be also employed, without any methodical changes also for problems in other fields, e.g. chemical engineering, medical technology or biotechnology.

An extension of the code with respect to an acoustic thermal fluid–structure interaction is also an interesting aspect for future work. In the Chair of Numerical Methods in Mechanical Engineering a version of FASTEST with an acoustic coupling is already developed and validated. Incorporating it in the

TFSI framework should be a straightforward task.

Finally, the coupled software package can be employed for optimization problems. A highly efficient optimization procedure for TFSI which uses higher order surfaces and the continuous sensitivity equation method seems to be a promising approach.

List of Figures

2.1	Deformation of continuum body	11
2.2	A typical CV and the notation used for a Cartesian 3D grid . . .	21
2.3	Schematic view of the SIMPLE pressure–correction method . . .	24
2.4	Ghost cells along subdomain interfaces	26
3.1	Flow chart of the coupling procedure ITFSI1	34
3.2	Flow chart of the coupling procedure ITFSI2	35
3.3	East side of an arbitrary control volume at the coupling interface (neighboring points are labeled according to the compass notation)	37
3.4	Modified bilinear interpolation using 3 points	38
3.5	Schematic view of one TFSI iteration	40
3.6	2D TFI procedure. Computational domain (left) is mapped onto the Physical domain (right).	44
3.7	2D elliptic procedure.	45
3.8	Parallelization strategies	49
4.1	Geometry and numerical for TC1	52
4.2	Geometry of TC2	53
4.3	Grid configuration for TC2. Every 4 th grid line is shown	55
4.4	Temperature distribution (in [K]) in the middle of the geometry for TC2	55
4.5	Absolute velocity (in [m/s]) at $\frac{\pi}{2}$ time period	55
4.6	Absolute velocity (in [m/s]) at $\frac{3\pi}{2}$ time period	56
4.7	Absolute velocity (in [m/s]) at 2π time period	56
4.8	Displacement course of the reference point A, drag and lift for TC2	57
4.9	Geometry configuration for TC3	58
4.10	Geometrical setup for TC4	59
4.11	Discretization of the fluid domain.	60
4.12	Temperature distribution (in [K]) in a center slice of the geometry	61
4.13	Results for TC3	62
4.14	Comparison of the surface temperature. The error bars represent the shot-to-shot standard deviation for the experiment	63
4.15	Comparison of the temperature at a radial position $r = 0$ mm. .	64
4.16	Comparison of the temperature at a radial position $r = 15$ mm. .	65

4.17	Results for TC4	66
4.18	Convergence of the Nusselt numbers for successively refined grids	67
4.19	Flow chart of the coupling procedure ITFSI3	70
4.20	Geometry of the test configuration TC5	71
4.21	Acceleration due to multigrid calculation in the flow solver for TC4	73
4.22	Geometry setup for TC6	76
4.23	Discretization grid for TC6	76
4.24	Block structure and grid distortion concept	77
4.25	Overview of the distorted mesh	78
4.26	Distorted grid around the rotating cylinder	78
4.27	Distorted grid around the steel mass	79
4.28	Time series for v for TC6	80
4.29	Normed velocity profile for TC6	81
4.30	Pressure for different time-phase angles for TC6	82

List of Tables

4.1	Material properties for TC1 and TC2	54
4.2	Error of the simulation for different grids and boundary conditions for TC1	54
4.3	Material properties used for TC3 and TC4.	59
4.4	Residual of the thermal coupling $\ T^{m-1} - T^m\ _\infty / \ T^m\ _\infty$ for different grids for TC1, TC3 and TC4 for every TFSI iteration it	67
4.5	Residual of the thermal coupling $\ T^{m-1} - T^m\ _\infty / \ T^m\ _\infty$ for TC1 and TC3 for every TFSI iteration it	68
4.6	Number of CFD iterations for different grids for TC1 and TC3 for every TFSI iteration it. Dirichlet–Neumann thermal coupling applied.	68
4.7	Number of CFD iterations for different grids for TC1 and TC3 for every TFSI iteration it. Neumann–Dirichlet thermal coupling applied.	69
4.8	Number of TFSI iterations by different preconditioners for TC2 and TC4	69
4.9	Material properties used for TC5.	71
4.10	CPU time (in seconds) and E_x for 100 time steps with ITFSI2 and ITFSI3 with different estimation of distortion for TC5	72
4.11	CPU time (in seconds) and E_2 for 100 time steps with ITFSI2 and ITFSI3 and turbulent flow	72
4.12	Speed-up and parallel efficiency for the two parallelization strategies	74
4.13	Material properties used for TC6.	76

Bibliography

- [1] O. Axelsson. *Iterative Solutions Methods*. Cambridge University Press, second edition, 1994.
- [2] J. Barlow. Optimal stress locations in finite element models. *International Journal for Numerical Methods in Engineering*, 10:243–251, 1976.
- [3] K. J. Bathe, H. Zhang. Finite element developments for general fluid flows with structural interactions. *International Journal for Numerical Methods in Engineering*, 60:213–245, 2004.
- [4] G. Becker, U. Falk and M. Schäfer. Shape optimization with higher-order surfaces in consideration of fluid-structure interaction. In S. Hartmann, A. Meister, M. Schäfer and S. Turek, editor, *International Workshop on Fluid-Structure Interaction: Theory, Numerics and Application*. Kassel Univ. Press, Kassel, 2009.
- [5] G. Becker, Y. Du, P. Pironkov and M. Schäfer. Quantitative comparison of fsi simulation approaches with respect to efficiency and accuracy. In *Computational Methods for Coupled Problems in Science and Engineering III*, Ischia, Italy, June 2009. CIMNE.
- [6] T. Belytschko, H.J. Yen and R. Mullen. Mixed methods for time integration. *Computer Methods in Applied Mechanics and Engineering*, 17/18:259–275, 1979.
- [7] F.J. Blom. A monolithical fluid-structure interaction algorithm applied to the piston problem. *Computer Methods in Applied Mechanics and Engineering*, 167:369–391, 1998.
- [8] P. Bontoux, E. Serre, E. Severac, R. Pasquetti, K. Krastev, P. Pironkov, M. Schäfer and I. Serwas. A pseudo-spectral approach for LES of complex flows. In *Euromech Colloquium 469: LES of Complex Flows*, pages 55–56. TU Dresden, 2005.
- [9] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computations*, 31:333–390, 1977.
- [10] W.L. Briggs. *A Multigrid Tutorial*. SIAM, 2000.

- [11] J. Brübach, E. van Veen and A. Dreizler. Combined phosphor and cars thermometry at the wall-gas interface of impinging flame and jet systems. *Experiments in Fluids*, 44:897–904, 2008.
- [12] H.J. Bungartz and M. Schäfer. *Fluid-Structure Interaction. Modeling, Simulation, Optimization*. Lecture Notes in Computational Science and Engineering. Springer Verlag, Berlin, 2006.
- [13] P. Causin, J. F. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Computer Methods in Applied Mechanics and Engineering*, 194:4506–4527, 2005.
- [14] K. Chien. Predictions of channel and boundary-layer flows with low-reynolds-number turbulence model. *AIAA Journal*, 20:33–38, 1982.
- [15] M. Ciofalo. Large-eddy simulation: A critical survey of models and applications. *Advances in Heat Transfer*, 25:321–419, 1994.
- [16] T.J. Craft, L.J.W. Graham and B. E. Launder. Impinging jet studies for turbulence model assessment ii. an examination of the performance of four turbulence models. *International Journal of Heat and Mass Transfer*, 36:2685–, 1993.
- [17] J. Degroote, P. Bruggeman, R. Haelterman and J. Vierendeels. Stability of a coupling technique for partitioned solvers in fsi applications. *Computers and Structures*, 86(23-24):2224–2234, 2008.
- [18] J. Degroote, S. Abigail, P. Bruggeman, R. Haelterman, P. Segers and J. Vierendeels. Simulation of fluid-structure interaction with the interface artificial compressibility method. 2009.
- [19] I. Demirdzić and M. Perić. Space conservation law in finite volume calculations of fluid flow. *International Journal for Numerical Methods in Fluids*, 8:1037–1050, 1989.
- [20] J. Donea, A. Huerta, J.Ph. Ponthot and A. Rodríguez-Ferran. *Arbitrary Lagrangian-Eulerian Methods*, volume 1 of *Encyclopedia of Computational Mechanics*. John Wiley & Sons, Inc., 2004.
- [21] E. Dowell, J. Thomas and K. Hall. Transonic limit cycle oscillation analysis using reduced order aerodynamic models. *Journal of Fluids and Structures*, 19:17–27, 2004.
- [22] F. Durst and M. Schfer. A parallel blockstructured multigrid method for the prediction of incompressible flows. *International Journal for Numerical Methods in Fluids*, 22:549–565, 1996.
- [23] R. Dwight. Robust mesh deformation using the linear elasticity equations. *Computational Fluid Dynamics*, 2006.

- [24] H. Eschenauer, N. Olhoff and W. Schnell. *Applied Structural Mechanics*. Springer Verlag, Berlin, 1997.
- [25] C. Farhat, K. G. van der Zee, P. Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 195:1973–2001, 2006.
- [26] C. Farhat, M. Lesoinne and P. Le Tallec. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 157:95–114, 1998.
- [27] C.A. Felippa, K.C. Park and C. Farhat. Partitioned analysis of coupled mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 190:3247–3270, 2001.
- [28] J.H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer Verlag, Berlin, 3rd. edition, 2002.
- [29] C. Frster, W. A. Wall and E. Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 196:1278–1293, 2007.
- [30] F. Gastaldi and L. Gastaldi. On a domain decomposition for the transport equation: theory and finite element approximation. *IMA Journal of Numerical Analysis*, 14:111–135, 1994.
- [31] M. Germano, U. Piomelli, P. Moin and W.H. Cabot. A dynamic subgrid-scale eddy viscosity model. *Physics Fluids A*, 3(7):1760–1765, 1991.
- [32] M. Glck. *Ein Beitrag zur Numerischen Simulation von Fluid-Struktur-Interaktionen – Grundlagenuntersuchungen und Anwendung auf Membrantragwerke*. PhD thesis, Technische Fakultt der Universitt Erlangen–Nrnberg, 2002. (In German).
- [33] J. Gomes and H. Lienhart. Reference test cases for fluid-structure interaction studies. In S. Hartmann, A. Meister, M. Schäfer and S. Turek, editor, *International Workshop on Fluid-Structure Interaction: Theory, Numerics and Application*. Kassel Univ. Press, Kassel, 2009.
- [34] W. Hackbusch. *Theorie und Numerik elliptischer Differentialgleichungen*. B. G. Teubner Stuttgart, second edition, 1986.
- [35] M.C. Haupt, R. Niesner and P. Horst. Coupling techniques for aero-thermo-elasticity. *Computational Methods for Coupled Problems in Science and Engineering*, pages 19–22, 2005.

- [36] M. Heck. *Mehrgitterverfahren zur effizienten numerischen Simulation von Fluid-Struktur-Wechselwirkungen*. PhD thesis, TU Darmstadt, 2008. (In German).
- [37] M. Heil. An efficient solver for the fully coupled solution of large-displacement fluid–structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 193:1–23, 2004.
- [38] C. Hirsch. *Numerical Computation of Internal and External Flows*. Wiley, Chichester, 1988.
- [39] J. Hron and S. Turek. A monolithic fem solver for ale formulation of fluid–structure interaction with configurations for numerical benchmarking. In *Proc. First International Conference on Computational Methods for Coupled Problems in Science and Engineering*, 2005.
- [40] J. Hron and S. Turek. *A Monolithic FEM/Multigrid Solver for an ALE Formulation of Fluid-Structure Interaction with Applications in Biomechanics*, volume 53 of *Lecture Notes in Computational Science and Engineering*. Springer Verlag, Berlin, 2006.
- [41] B. Hübner, E. Walhorn and D. Dinkler. A monolithic approach to fluid-structure interaction using space-time finite elements. *Computer Methods in Applied Mechanics and Engineering*, 193:2087–2104, 2004.
- [42] H.B. Johnson, T.G. Seipp and G.V. Gandler. Numerical study of hypersonic reacting boundary layer transition on cones. *Physics of Fluids*, 10(10):2676–2685, 1998.
- [43] G. Kadavelil, M. Kornhaas, S. Sarić, D.C. Sternel, S. Jakirlić and M. Schfer. Numerical and physical aspects in LES and hybrid LES/RANS of turbulent flow separation in a 3–d diffuser. In *Turbulence, Heat and Mass Transfer 6*. Begell House, Inc., 2009.
- [44] J. Kim, D. Kim and H. Choi. An immersed–boundary finite-volume method for simulations of flow in complex geometries. *Journal of Computational Physics*, 171:132–150, 2001.
- [45] A.N. Kolmogorov. Almost horizontal turbulence. *Russian Mathematical Surveys*, 59(2):197–202, 2004.
- [46] B.E. Launder and D.B. Spalding. The numerical computation of turbulent flows. *Numerical Mathematics and Advanced Applications*, pages 625–632, 1974.
- [47] I. Lee, J.H. Roh and I.K. Oh. Aerothermoelastic phenomena of aerospace and composite structures. *Journal of Thermal Stresses*, 26(6):525–546, 2003.

- [48] T. Lehnhäuser. *Eine effiziente numerische Methode zur Gestaltoptimierung von Strömungsgebieten*. PhD thesis, TU Darmstadt, 2003. (In German).
- [49] T. Lehnhäuser and M. Schäfer. Improved linear interpolation practice for finite-volume schemes on complex grids. *International Journal for Numerical Methods in Fluids*, 38:625–645, 2002.
- [50] L. Li, S.J. Sherwin and P.W. Bearman. A moving frame of reference algorithm for fluid/structure interaction of rotating and translating bodies. *International Journal for Numerical Methods in Fluids*, pages 187–206, 2002.
- [51] H. Lienhart, J.P. Gomes, M. Schäfer and S. Yigit. Experimental and numerical study on a fluid–structure interaction reference test case. *to appear*, 2009.
- [52] D.K. Lilly. A proposed modification of the germano subgrid-scale closure method. *Physics of Fluids A*, 4(3):633–635, 1992.
- [53] X. Liu, N. Qin and H. Xia. Fast dynamic grid deformation based on delaunay graph mapping. *Journal of Computational Physics*, 211(2):405–423, 2006.
- [54] K.Z. Markov. *Mathematical Modeling*. “St. Kliment Ohridski” University Press, Sofia, 2002. (In Bulgarian).
- [55] K.Z. Markov. *Continuum Mechanics*. “St. Kliment Ohridski” University Press, Sofia, 2003. (In Bulgarian).
- [56] H.G. Matthies and J. Steindorf. Partitioned but strongly coupled iteration schemes for non-linear fluid–structure interaction. *Computers and Structures*, 80:1991–1999, 2002.
- [57] H.G. Matthies, R. Niekamp and J. Steindorf. Algorithms for strong coupling procedures. *Computer Methods in Applied Mechanics and Engineering*, 195:2028–2049, 2006.
- [58] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37:239–261, 2005.
- [59] D.P. Mok. *Partitionierte Lösungstechnik in der Struktur- und der Fluid-Struktur-Interaktion*. PhD thesis, Institut für Baustatik der Universität Stuttgart, 2001. (In German).
- [60] M.J. Moran, H.N. Shapiro, B.R. Munson and D.P. DeWitt. *Introduction to Thermal Systems Engineering: Thermodynamics, Fluid Mechanics, and Heat Transfer*. John Wiley & Sons, Inc., 2003.

- [61] D.J. Newman and G. E. Karniadakis. A direct numerical simulation study of flow past a freely vibrating cable. *Journal of Fluid Mechanics*, 344:95–136, 1997.
- [62] N. Newmark. A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division*, 85:67–94, 1959.
- [63] N. Nitin and M.W. Karwe. Numerical simulation and experimental investigation of conjugate heat transfer between a turbulent hot air jet impinging on a cookie-shaped object. *Journal of Food Science*, pages 59–65, 2006.
- [64] Department of Numerical Methods in Mechanical Engineering. *FASTEST – Users’ Manual*. Technische Universität Darmstadt, 2004.
- [65] S.V. Patankar. *Numerical Heat Transfer and Fluid Flow*. McCraw-Hill, New York, 1980.
- [66] S.V. Patankar and D.B. Spalding. A calculation procedure for heat, mass and momentum transfer in three dimensional parabolic flows. *International Journal of Heat Mass Transfer*, 15:1787–1806, 1972.
- [67] S. Piperno. Explicit/implicit fluid/structure staggered procedures with a structural predictor and fluid subcycling for 2d inviscid aeroelastic simulations. *International Journal for Numerical Methods in Fluids*, 25:1207–1226, 1997.
- [68] S. Piperno, C. Farhat and B. Larrouturou. Partitioned procedures for the transient solution of coupled aeroelastic problems – part i: model problem, theory and two-dimensional application. *Computer Methods in Applied Mechanics and Engineering*, 124:79–112, 1995.
- [69] S. Piperno and C. Farhat. Partitioned procedures for the transient solution of coupled aeroelastic problems – part ii: energy transfer analysis and three-dimensional applications. *Computer Methods in Applied Mechanics and Engineering*, 190:3147–3170, 2001.
- [70] P. Pironkov and M. Schäfer . A second order coupling algorithm for thermal fluid–structure interaction. 2009. to appear.
- [71] P. Pironkov and M. Schäfer. Numerical study on thermal turbulent fluid–structure interaction. In *Proc. 20th International Symposium on Transport Phenomena*, 2009.
- [72] P. Pironkov, M. Schäfer and K. Krastev. Properties of discrete chebyshev collocation differential operators in curvilinear geometries. *Numerical Linear Algebra with Applications*, 2009. to appear.
- [73] S. Polat and W.J. Douglas. Heat transfer under multiple slot impinging on a permeable moving surface. *AICHE J.* 36, pages 1370–1378, 1990.

- [74] S.B. Pope. *Turbulent Flows*. Cambridge University Press, 4th. edition, 2005.
- [75] A. Quarteroni and A. Valli. *Domain decomposition methods for partial differential equations*, volume 1 of *Numerical Mathematics and Scientific Computation*. The Clarendon Press Oxford University Press, 1999.
- [76] J.R. Rice, E.A. Vavalis and Daoqi Yang. Convergence analysis of a non-overlapping domain decomposition method for elliptic pdes. *Journal of Computational and Applied Mathematics*, 87:4790–4797, 1997.
- [77] S.M. Rifai, Z. Johan, W.P. Wang, T.J.R. Hughes and R.M. Ferencz. Multi-physics simulation of flow-induced vibrations and aeroelasticity on parallel computing platforms. *Computer Methods in Applied Mechanics and Engineering*, 174:393–417, 1999.
- [78] W. Rodi, J.H. Ferziger, M. Breuer and M. Pourquié. Status of large eddy simulation: Results of a workshop. *Journal of Fluids Engineering – Transactions of the ASME*, 119:248–262, 1997.
- [79] B. Roe, A. Haselbacher and P.H. Geubelle. Stability of fluid-structure thermal simulations on moving grids. *International Journal for Numerical Methods in Fluids*, 54:1097–1117, 2007.
- [80] S. Rugonyi and K.J. Bathe. On the finite element analysis of fluid flows fully coupled with structural interactions. *Computing in Science and Engineering*, 2:195–212, 2001.
- [81] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, second edition, 1996. (freely available in electronic format).
- [82] S. Sachs and M. Schäfer. Simultaneous multigrid and multilevel coupling in fluid-structure interaction. In *Proc. First International Workshop on Computational Engineering*, 2009.
- [83] SCAI. *MpCCI 3.0.5 Documentation*, 2006.
- [84] M. Schäfer. *Computational Engineering. Introduction to Numerical Methods*. Springer Verlag, Berlin, 2006.
- [85] R. Schäfer. Thermisch-mechanisches Verhalten heisser Strukturen in der Wechselwirkung mit einem umströmenden Fluid, 2005. Forschungsbericht.
- [86] M. Schäfer and P. Pironkov. Efficient numerical simulation and optimization of thermally coupled fluid-solid problems. In *Proc. 19th International Symposium on Transport Phenomena*, 2008.
- [87] M. Schäfer and S. Turek. Benchmark computations of laminar flow around a cylinder. *Flow Simulation with High-Performance Computers II, Notes on Numerical Fluid Mechanics*, 52:547–566, 1996.

- [88] C. Scotti and E. Finol. Compliant biomechanics of abdominal aortic aneurysms: a fluid–structure interaction study. *Computers and Structures*, 85:1097–1130, 2007.
- [89] B. Sendov and V. Popov. *Numerical Methods*, volume I. “St. Kliment Ohridski” University Press, Sofia, second edition, 1995. (In Bulgarian).
- [90] R. Sieber. *Numerische Simulation technischer Strömungen mit Fluid–Struktur–Kopplung*. PhD thesis, TU Darmstadt, 2002. (In German).
- [91] S.L. Sondak and D.J. Dorney. Simulation of coupled unsteady flow and heat conduction in turbine stage. *AIAA Journal of Propulsion and Power*, 16(6):1141–1148, 2000.
- [92] S. P. Spekreijse. Elliptic grid generation based on laplace equations and algebraic transformations. *Journal of Computational Physics*, 118(1):38–61, 1995.
- [93] K.R. Stein. *Simulation and modeling techniques for parachute fluid–structure interactions*. PhD thesis, University of Minnesota, 1999.
- [94] K. Stein, R. Benney, V. Kalro, T.E. Tezduyar, J. Leonard, and M. Accorsi. Parachute fluid–structure interactions: 3-d computation. *Computer Methods in Applied Mechanics and Engineering*, 190:373–386, 2000.
- [95] K. Stein, T. Tezduyar and R. Benney. Mesh moving techniques for fluid–structure interactions with large displacements: Flow simulation and modeling. *Journal of applied mechanics*, 70:58–63, 2003.
- [96] D.C. Sternel, P. Pironkov, M. Kornhaas and M. Schäfer. Fluid–structure interaction in turbulent flow. In *Computational Methods for Coupled Problems in Science and Engineering III*, Ischia, Italy, June 2009. CIMNE.
- [97] P. Le Tallec and J. Mouro. Fluid structure interaction with large structural displacements. *Computer Methods in Applied Mechanics and Engineering*, 190:3039–3067, 2001.
- [98] R.L. Taylor. *FEAP – A Finite Element Analysis Program. Version 7.5 User Manual*, 2004.
- [99] I. Teschauer. *Numerische Simulation gekoppelter Fluid-Struktur Probleme mittels der Finite-Volumen-Methode*. PhD thesis, TU Darmstadt, 2001. (In German).
- [100] T. E. Tezduyar, M. Schwaab and S. Sathe. Sequentially-coupled arterial fluid–structure interaction (scafsi) technique. *Computer Methods in Applied Mechanics and Engineering*, 2008.
- [101] U. Trottenberg, C.W. Oosterlee and A. Schüller . *Multigrid*. Elsevier Academic Press, 2001.

- [102] A. van Zuijlen, S. Bosscher and H. Bijl. Two level algorithms for partitioned fluid–structure interaction computations. *Computer Methods in Applied Mechanics and Engineering*, 196:1458–1470, 2007.
- [103] J. Vierendeels, L. Lanoye, J. Degroote and P. Verdonck. Implicit coupling of partitioned fluid–structure interaction problems with reduced order models. *Computers and Structures*, 85:970–976, 2007.
- [104] E. P. Weidmann, W. Bauer, M. Maihöfer and U. Scholl. A new CFD and thermal analysis method based on MpCCI. In *Proc. 8th MpCCI User Forum*, pages 18–27. Fraunhofer Institute SCAI, 2007.
- [105] P. Wesseling. *An Introduction to Multigrid Methods*. R.T. Edwards, Inc., 2004.
- [106] K. Willcox, J. Paduano and J. Peraire. Low order aerodynamic models for aeroelastic control of turbomachines. In *40th AIAA ASME ASCE AHS ASC structures, structural dynamics and materials conference, St Louis, MO*, 1999.
- [107] P. Wriggers. *Nichtlineare Finite-Element-Methoden*. Springer Verlag, Berlin, 2001.
- [108] J. Xu and J. Zou. Non-overlapping domain decomposition methods. SIAM Review, 1998.
- [109] C.T. Yamamotoa, J.R. Meneghinib, F. Saltarab, R.A. Fregonesib and J.A. Ferrari. Numerical simulations of vortex-induced vibrations on flexible cylinders. *Journal of Fluids and Structures*, 19:467–489, 2004.
- [110] D. Yang. A parallel nonoverlapping schwarz domain decomposition method for elliptic interface problems. In *Proc. Eighth SIAM Conference on Parallel Processing for Scientific Computing*, 1997.
- [111] Z. Yang and D. Mavripilis. Unstructured dynamic meshes with higher-order time integration schemes for the unsteady navier-stokes equations. *AIAA Journal*, 2005.
- [112] J. Yang, S. Preidikman, E. Balaras. A strong coupling scheme for fluid–structure interaction problems in viscous incompressible flows. In *Proc. International Conference on Computational Methods for Coupled Problems in Science and Engineering*, CIMNE, Barcelona, 2005.
- [113] S. Yigit. *Phänomene der Fluid-Struktur-Wechselwirkung und deren numerische Berechnung*. PhD thesis, TU Darmstadt, 2008. (In German).
- [114] S. Yigit, M. Heck, D.C. Stempel and M. Schfer. Efficiency of fluid–structure interaction simulations with adaptive underrelaxation and multigrid acceleration. *International Journal of Multiphysics*, 1(1):85–99, 2007.

- [115] Z. Zaprianov. *Hydrodynamics*. “St. Kliment Ohridski” University Press, Sofia, 1995. (In Bulgarian).
- [116] S.J. Zhang, J. Liu, Y.S. Chen and X. Zhao. A dynamic mesh method for unstructured flow solvers. *AIAA Journal*, 2003.
- [117] Q.D. Zhu and Q. Lin. *Superconvergence Theory of the Finite Element Methods*. Hunan Science and Technology Press, Hunan, China, 1989.
- [118] O.C. Zienkiewicz, R.C. Taylor and J.Z. Zhu. *The finite element method. Its Basis and Fundamentals*. Elsevier Butterworth-Heinemann, 6th. edition, 2005.

Plamen Pironkov

Curriculum Vitae

Personal information

Name	Pironkov, Plamen
Address	Kesselhutweg 1, 64289 Darmstadt, Germany
E-Mail	pironkov@fnb.tu-darmstadt.de
Nationality	Bulgarian
Date of birth	05.12.1979
Gender	Male
Marital status	Single

Education

- 2002–2004 **M.Sc.**, *Sofia University, Faculty of Mathematics and Informatics*, Computational Mathematics.
Master thesis: *Numerical integration with two side approximations*, supervisor: Geno Nikolov
- 1998–2002 **B.Sc.**, *Sofia University, Faculty of Mathematics and Informatics*, Computer Sciences.
- 1993–1998 **Secondary School**, *National High School of Mathematics and Applied Sciences*, Mathematics and Computer Science.

Experience

- 2004–recent **Research associate**, *Technische Universität Darmstadt*, FNB, Dolivostrasse 15, 64293, Darmstadt, Germany.
Numerical methods for coupled problems. Spectral methods for CFD.
- 2003–2004 **Product manager**, *Efficient Software Systems Ltd.*, km. 7, Tzarigradsko shosse, IPAKT, 202 Sofia 1784, Bulgaria.
Management of software projects for business processes (BPMS), research, analysis and integration. Management consulting.
- 2001–2003 **Project manager, software developer**, *LSN Software and Telecommunication Services Ltd.*, km. 7, Tzarigradsko shosse, IPAKT, 202 Sofia 1784, Bulgaria.
Software development for GSM network planning. Operator–regulator policies.
- 1999–2001 **System operator**, *Eurofootball Ltd.*, Koloman Str. 1, 1618 Sofia, Bulgaria.
Development and administration of electronic bookmaking system.