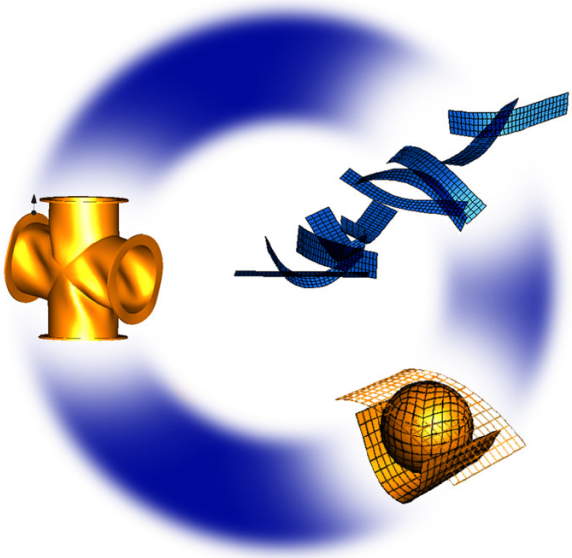
Technische Universität München

A Simple Yet Revealing Title

John Doe

Type of Thesis



Supervisor:
D. B. Cooper, M.Sc.
someone@lnm.mw.tum.de

Institute for Computational Mechanics
Prof. Dr.–Ing. W. A. Wall
Technische Universität München
Boltzmannstraße 15
85748 Garching b. München (Germany)

Own work declaration

Hereby I confirm that this is my own work and I referenced all sources and auxiliary means used and acknowledged any help that I have received from others as appropriate.

(location, date) (signature)

Abstract

This is the most often read section. It will determine if someone actually bothers to read more of what you've written. Only write this section after you have completed your report. The abstract consist of only one paragraph, mostly limited to 200-300 words. Literally, a summary of your work. Write a sentence or two about each of the main sections of your report, as discussed in this section. When summarizing results, make the reader aware of the most important results (including numbers when applicable) and important conclusions or questions that follow from these.

Zusammenfassung

german version of your abstract.

Contents

List of Figures

Nomenclature

Symbols

3-field formulations

T_p	Interface projection matrix from fluid to structure mesh	`ifaceprojFtoS`
$\boldsymbol{P}_F$	Fluid load TODO	`fload`
$\mathcal{D}$	State equation of the mesh	`EOSmesh`
$\boldsymbol{H}_2$	Second order Jacobian TODO	`jactwo`
χ	Additional fluid state variable introduced by the turbulence model	`turbulenceparam`
$\boldsymbol{x}$	Fluid mesh motion	`mms`
$\boldsymbol{P}_T$	Structure load TODO	`sload`
$\boldsymbol{u}$	Structure displacement	`structdisp`
S	Source term in the RANS equations	`turbulencesourc`
$\mathcal{S}$	State equation of the structure	`EOSstruct`
T_u	Interface projection matrix from structure to fluid mesh	`ifaceprojStoF`
$\boldsymbol{w}_{RANS}$	Augumentes fluid state vector in the RANS formulation	`fstaterans`
$\mathcal{F}$	State equation of the fluid	`EOSfluid`
$\boldsymbol{w}$	Fluid state vector	`fstate`

Fluid Mechanics

τ	Deviatoric fluid stress tensor	`fluidstresscomp`
$\boldsymbol{q}$	Heat flux vector	`heatflux`
v_x	Fluid velocity in x-direction	`fluidvelx`
v_y	Fluid velocity in y-direction	`fluidvely`
p	Pressure	`pres`
ν	Kinematic viscosity	`viscoskin`
k	Thermal conductivity of the fluid	`thermcond`
RE	Reynolds number	`reynolds`
e	Internal energy	`energyint`
μ	Dynamic viscosity	`viscosdyn`
q	Heat flux comopnenent	`heatfluxcomp`
γ	Specific heat ratio	`specheatratio`
v_z	Fluid velocity in z-direction	`fluidvelz`
I	Identity matrix	`eye`
$\boldsymbol{v}$	Fluid velocity vector	`fluidvel`
$\mathcal{G}$	Diffusive fluxes	`fluxesdiff`
E	Total energys	`energytot`
ρ	Density	`dens`
T	Fluid temperature	`temp`
$\mathcal{F}$	Convective fluxed	`fluxesconv`

Fluid Structure Interaction

$\mathcal{D}$	TODO	`mmsstateeq`

Optimization

$n_{\boldsymbol{g}}$	Number of non-equality constraints	`numneqctr`
$\boldsymbol{\gamma}$	Lagrange multipliers of the inequality constraints	`lagmultsneq`
$\boldsymbol{s}$	Vector of abstract optimization variables	`absvars`
$n_{\boldsymbol{h}}$	Number of equality constraints	`numeqctr`
$\boldsymbol{g}$	Non-equality constraints	`neqctr`
$\boldsymbol{a}$	Adjoint solutions	`adjoints`
s	Abstract optimazation variable	`absvar`
z	Target cost function	`costfunc`
$\boldsymbol{\eta}$	Lagrange multipliers of the equality constraints	`lagmultseq`
$\boldsymbol{d}$	Physical design parameters	`physvars`
ϵ^{SA}	Specified tolerance in the Sensitivity analysis	`tolsa`
L	Lagrangian function of the optimization problem	`Lagfunc`
q	Optimization criterium	`optcrit`
$\boldsymbol{h}$	Equality constraints	`eqctr`
$\boldsymbol{q}$	Vector of optimization criteria	`optcrits`

Sturctural Analysis

$\mathbf{K}$	Finite Element stiffness matrix	`stiffmat`
u	Displacement vector	`disp`
$\boldsymbol{x}$	Mesh motion	`motion`
d	Interface Displacement	`ifacedisp`
$\mathbf{u}$	Disrete displacement vector	`dispvec`
$\mathbf{d}$	Interface Displacent	`ifacedispvec`

Operators

Operators and Symbols

A^T	Transpose of a tensor	`T`
A^{-1}	Inverse of a tensor	`inv`

Representation of scalars, tensors and other quantities

$\boldsymbol{A}$	Tensor	`tensor`
A'	Time-averaged quantity	`fluc`
$A^{(n)}$	Iteration index in thestaggered algorithm	`its`
$\mathbf{A}$	discrete vector represenation of a scalar field	`dvec`
$\mathbf{A}$	discrete vector represenation of a tensor field	`dmat`
$\ddot{A}$	Second time derivative at a fixed refrence position	`dderivtime`
A	Scalar quantity	`scalar`
$A^{(k)}$	Iteration index in the optimization loop	`ito`
$\bar{A}$	Fictious entity	`fic`
$\dot{A}$	First time derivative at a fixed reference position	`derivtime`

Abbrevations

SQP Sequential Quatdratic Programming

GSE Gloabal Sensitivity Equations

SA Sensitivity Analysis

VOF Volume of Fluid Method

GFM Ghost Fluid Method

GFMP Ghost Fluid Method of the Poor

EOS Equation of State

JWL Jones-Wilkins-Lee

SG Stiffend Gas

MUSCL Monotonic Upwind scheme for Conservation Laws

FIVER Finite Volume Method with exact two-phase Riemann Integrals

RANS Reynolds Averaged Navier-Stokes

FV Finite Volumes

FD Finite Differences

FE Finite Elements

SA Sensitivity Analysis

LNM Lehrstul fuer Numerische Mechanik(Institute of Computational Mechanics)

TUM Technical University od Munich

CFD Computational Fluid Dynamics

ALE Arbitrary Lagrangian Eulerian

VOF Volume of Fluid Methods

DFP Davidon Fletcher Powell formula

BFGS Broyden Fletcher Goldfarb Shanno algorithm

LDR Lift over Drag Ratio

NSE Navier Stokes Equations

PDE Partial Differential Equation

1 Introduction

1.1 Motivation

Thanks to the advent of massively parallel high-performance computers, Computational Fluid Dynamics (CFD) is nowadays capable of accurately predicting the flow characteristics of a large variety of real-world problems. Typically, it involves numerical analysis and algorithms in order to analyze the characteristics of a given input setting.
In contrast to a pure scientific point of view, which put the main objective in gaining fundamental insight into the flow phenomenon, an engineer typically sees the computational methods as a mean of designing and improving his product. Therefore he is not only satisfied with an sufficiently accurate approximation of the flow variables around his design objective, one could imagine an aircraft or an automobile, but he seeks to use that information to improve his product. For trivial cases it might be intuitive to identify opportunities for improvement, in a highly complex system with mutual interactions, however, this is not only time consuming but it might even be not possible at all.

The effort thus is to automatically determine the gradient of specific variables of interest with respect to design parameters. More vividly speaking, this could be the change of lift of a given airfoil with respect to its front edge curvature or its twist. In fact there are not restrictions when it comes to the choice of the design parameters. In fact one can also think of purely abstract ones, as we will describe in Section 2.2. However, the appropriate choice of this variables is the key-point in getting a satisfactory results. Details of this will be discussed in Section 2.2. The numerical methodology of calculating this gradients is known as Sensitivity Analysis (SA).

There are different approaches to do SA. Most intuitively, one can approximate the sensitivity of a target value with respect to some design parameter by a simple Finite Differences (FD) approach. Figuratively speaking, for the airfoil example above, this would mean, that on could run two(or more) simulations with slightly different edge curvatures and then obtain the sensitivity of the lift with respect to that design decision as the difference of the absolute lift values of both simulation divided by the difference in absolute curvature. This approach is appealing, since it can be done with a standard flow solver. However, to get one sensitivity result, at least two simulations now have to be run. What is more, finite differencing not only introduces an approximation error, the appropriate choice of the stencil size is also difficult. Moreover, two slightly different meshes are required, or at least a mean to perturb the original one according to the design parameter choice.
Another approach is automatic differentiation. It is different form numerical differentiation in the sense that its error is only determined by the finite accuracy of the machine, but like finite differencing, automatic differentiation of a function works on a specific input parameter set. Therefore it has to be done repeatedly over and over again. An elegant alternative that solves both issues it symbolic derivation, which

means that the equations of interest are derived by specific variables in a fully general manner, leading to a symbolic formula for the derivatives. Once obtained, this formula can then simply be evaluated by the code. We therefore no longer need multiple function evaluations nor is it necessary to do the derivation process over and over again. The payoff lies in the complexity. On can imagine that in a complex 3D fluid simulation, potentially with viscous, inviscid and turbulence-closure terms, this derivation is cumbersome and involves multiple chain rules.

The main purpose of SA lies in optimization. Given the a set of design parameters on can imagine that the methodology above is able to give reliable values for out target function, e.g. airfoil lift, with respect to these parameters, e.g. airfoil thickness at different points. Based on this information it is now possible to slightly modify the airfoil, according to these new results. After that the while process starts again until it finally converges to the optimal configuration.

1.2 Objective

The main purpose of this project is the implementation of the SA method into the CFD code AERO-F [4].. Both analytical methods and numerical schemes for the computation of aeroelastic sensitivities in the presence of turbulent viscous flows will be considered. The scope of this thesis covers both, the direct 3 and adjoint approaches 4, within the context of Eulerian, Arbitrary Lagrangian Eulerian (ALE) and Embedded formulations 6.. The main effort of this thesis concentrates on the computation of the contributions of the viscous terms to the aforementioned sensitivities, as well as a generalization of the technique to embedded methods.

2 Optimization

A generic, aeroelastic optimization problem can be written as

$$q(\boldsymbol{s},\boldsymbol{u},\boldsymbol{w}) \begin{cases} \min\limits_{\boldsymbol{s}} z(\boldsymbol{s}) \\ \boldsymbol{h}(s) = \boldsymbol{0}, \quad \boldsymbol{h} \in \mathbb{R}^{n_h} \\ \boldsymbol{g}(s) > \boldsymbol{0}, \quad \boldsymbol{g} \in \mathbb{R}^{n_g} \end{cases} \tag{1}$$

eq:generic_optimization_problem_1

$$\boldsymbol{s} = \{\boldsymbol{s} \in \mathbb{R}^{n_s} | \boldsymbol{s}_L \leq \boldsymbol{s} \leq \boldsymbol{s}_U\} \tag{2}$$

eq:generic_optimization_problem_2

Here, z is an arbitrary cost function that should be minimized. Note that a maximization problem can easily be obtained by multiplying the cost function by a factor of -1. In aerodynamics, a typical example for a cost function would be the Lift over Drag Ratio (LDR) ratio of an airfoil. The cost-function is described in terms of so-called *abstract variables*. These can have some physical interpretation, but don't necessarily have to(see Section 2.2). Since typically, these optimization problems have no finite solution on an unbounded domain, some restrictions/conditions are introduced. In the airfoil example, we would probably specify a minimum lift that is required to support the configuration. Also a maximum stress for the structure will likely have be specified. These are classical examples of non-equality constraints, denoted by $\boldsymbol{g}(\boldsymbol{s})$ in the above formulation. Constraints can also be formulated in an equality sense, for instance geometrical restrictions due to the turbine suspensions. Furthermore, the abstract optimization variables $\boldsymbol{s}$ themselves are typically restricted by lower($\boldsymbol{s}_L$) and upper($\boldsymbol{s}_U$) bounds.
The combinations of objective function z and constraints $\boldsymbol{h}$ and $\boldsymbol{g}$ is typically denoted as optimization criteria q.

$$\begin{aligned} q &= q(\boldsymbol{s},\boldsymbol{u},\boldsymbol{w}) \\ \boldsymbol{u} &= \boldsymbol{u}(\boldsymbol{s}), \quad \boldsymbol{w} = \boldsymbol{w}(\boldsymbol{s}) \end{aligned} \tag{3}$$

eq:optimization_criteria

This thesis follows the *nested analysis and design approach*, meaning that we assume that $\boldsymbol{u}$ and $\boldsymbol{w}$ always satisfy the aeroelastic state equations. This means that the state equations are not included in the set of constraints, but the structural displacements $\boldsymbol{u}$ and the fluid state variables $\boldsymbol{w}$ are determined in each iteration.

As [9] write, the aeroelastic optimization problem can typically be solved by combining three different numerical approaches:

- Optimization Model
- Design Model
- Analysis Model

The optimization model describes the solution of the generic optimization problem(1)-(2). For this thesis, a Sequential Quatdratic Programming (SQP) has been chosen [3].
The design model links abstract optimization variables $\boldsymbol{s}$ to actual shapes, structures, geometries or aerodynamics design parameters. For this purpose SDESIGN,

a programm specifically written for SA purpose in the **FRG!** (**FRG!**), has been used during this thesis. Its basic concepts an ideas are described in [10].
Finally, the analysis model provides concepts of evaluation the optimization criteria. Typically, the optimization criteria depend on $\boldsymbol{u}$ and $\boldsymbol{w}$ which is why a coupled system of equations has to be solved in every design optimization process. The Sensitivity Analysis (SA) is also part of this model. Aeroelastic analysis and Sensitivity analysis are discussed in Section 2.4.

2.1 Optimization Model

Optimization problems are typically solved by gradient-based methods. The methods are divided into

- Primal
- Dual
- Penalty-barrier and
- Lagrange approaches

This thesis focuses on Lagrange approaches. For a thorough analysis and comparison of the different approaches, the interested reader is referred to [12].
The Lagrangian approach formulates the optimization problem(1)(2) as an extreme value problem of the Lagrangian:

eq:lagrangian_of_optimization

$$L(\boldsymbol{s}, \boldsymbol{\eta}, \boldsymbol{\gamma}) = z(\boldsymbol{s})) = \boldsymbol{\eta}^T \boldsymbol{h}(\boldsymbol{s}) + \boldsymbol{\gamma}^T \boldsymbol{g}(\boldsymbol{s}) \tag{4}$$

(5)

Here, $\boldsymbol{\eta}$ denote the Lagrange multipliers for the equality constraints and $\boldsymbol{\gamma}$ the Lagrange multipliers for the non-equality constraints. In fact, one can easily see that the original optimization problem can be obtained as saddle point of the Lagrangian:

eq:saddlepoint_optimization

$$\frac{\partial L}{\partial \boldsymbol{s}} = \frac{\partial z}{\partial \boldsymbol{s}} = \boldsymbol{\eta}^T \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{s}} + \boldsymbol{\gamma}^T \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{s}} \tag{6}$$

$$\frac{\partial L}{\partial \boldsymbol{\eta}} = \boldsymbol{h} = \boldsymbol{0} \tag{7}$$

$$\frac{\partial L}{\partial \boldsymbol{\gamma}} = \boldsymbol{\gamma}^T \boldsymbol{g} = \boldsymbol{0} \tag{8}$$

The SQP method, mention before uses a Newton-Rhapsodon algorithm to solve the above system.

eq:saddlepoint_newtonform

$$\begin{bmatrix} \frac{\partial^2 L}{\partial \boldsymbol{s}^2}^{(k)} & \frac{\partial \boldsymbol{g}}{\partial s}^{(k)} & \frac{\partial \boldsymbol{h}}{\partial s}^{(k)} \\ \boldsymbol{\gamma}^{(k)} \frac{\partial \boldsymbol{g}}{\partial s}^{(k)} & \boldsymbol{h}^{(k)} & \boldsymbol{0} \\ \frac{\partial \boldsymbol{h}}{\partial s}^{(k)} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{s} \\ \Delta \boldsymbol{\gamma} \\ \Delta \boldsymbol{\eta} \end{bmatrix} = - \begin{bmatrix} \frac{\partial L}{\partial \boldsymbol{s}}^{(k)} \\ \boldsymbol{\gamma}^{(k)T} \boldsymbol{g}^{(k)} \\ \boldsymbol{h}^{(k)} \end{bmatrix} \tag{9}$$

Here $(\cdot)^{(k)}$ denotes the iteration index of the optimization loop.

The linear Equations (9) can also be formulated as an equivalent quadratic minimization problem:

$$\min_{\boldsymbol{s}} \left(\frac{1}{2}\Delta \boldsymbol{s}^T \frac{\partial^2 L}{\partial \boldsymbol{s}^2}^{(k)} \Delta \boldsymbol{s} + \frac{\partial L}{\partial \boldsymbol{s}}^{(k)}\right) \qquad \texttt{eq:quadratic\_minimization\_problem\_1} \tag{10}$$

$$\frac{\partial \boldsymbol{g}}{\partial s}^{(k)} \Delta \boldsymbol{s} + \boldsymbol{g}^{(k)} \geq \boldsymbol{0} \qquad \texttt{eq:quadratic\_minimization\_problem\_2} \tag{11}$$

$$\frac{\partial \boldsymbol{h}}{\partial s}^{(k)} \Delta \boldsymbol{s} + \boldsymbol{h}^{(k)} = \boldsymbol{0} \qquad \texttt{eq:quadratic\_minimization\_problem\_3} \tag{12}$$

In the above formulation, the evaluation of the second derivative of the Lagrangian(Hessian of L) is the numerically most expensive part. Usually it is preferred to approximate it by a first-order information, for example by the Davidon Fletcher Powell formula (DFP) or by the Broyden Fletcher Goldfarb Shanno algorithm (BFGS) method. However, this simplification introduces an error that one should be aware of. Some correction methods have been proposed trying to minimize the error. In this thesis we adapt the one proposed by [9]:

$$\begin{bmatrix} \boldsymbol{s}^{(k+1)} \\ \boldsymbol{\eta}^{(k+1)} \\ \boldsymbol{\gamma}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}^{(k)} \\ \boldsymbol{\eta}^{(k)} \\ \boldsymbol{\gamma}^{(k)} \end{bmatrix} + \alpha^{(k)} \begin{bmatrix} \Delta \boldsymbol{s}^{(k)} \\ \Delta \boldsymbol{\eta}^{(k)} \\ \Delta \boldsymbol{\gamma}^{(k)} \end{bmatrix} \qquad \texttt{eq:correction\_step} \tag{13}$$

The appropriate step size $\alpha^{(k)}$ is determined by a line search. AS [9] write, due to this insufficiencies, the Lagrangian can not be used to measure an improvement. Instead, a merit function is introduced that is minimized by the line-search. A local minimum is reached by following a sequence of decreasing merit function values. Convergence of the process can be measured via the $\mathcal{L}_2$-norm of the residual of the Kuhn-Tucker conditions(6).
By construction, the constraints are satisfied only when the optimum point is reached.

2.2 Design Model

The design model represent the essential link between the described optimization model and the analysis model. Generally speaking, it relates physical design parameters to abstract ones.

$$\boldsymbol{d} = \boldsymbol{d}(\boldsymbol{s}), \quad \boldsymbol{d} \in \mathbb{R}^{n_{\boldsymbol{d}}} \qquad \texttt{eq:physvarsTOabsvars} \tag{14}$$

Here, $n_{\boldsymbol{d}}$ denotes the number of physical design parameters. To define a relation between the abstract optimization variables and the motion of the nodes, the following design model is introduced:

$$\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{s}) \tag{15}$$

As [9] explain, it is unpractical to identify an abstract variable directly with an increment of the coordinate of a grid point. Instead, two approaches, namely geometrical and mechanical can be adopted for constructing the generic design model.

Mechanical approach In the mechanical approach, the shape variation is identified with a superposition of fictitious structural deformations $\bar{\boldsymbol{u}}_j$ due to fictitious loads $\mu\bar{\boldsymbol{P}}_j$ and fictions support conditions

$$\boldsymbol{x} = \sum_j \bar{\boldsymbol{u}}_j = \sum_j \bar{\mathbf{K}}_j^{-1} \mu_j \bar{\boldsymbol{P}}_j \tag{16}$$ eq:mechanical_approach

where $\bar{\mathbf{K}}_j$ is a fictitious pseudo structure stiffness matrix representing the fluid domain compatible to the current fictions support conditions.

Geometrical approach The geometrical approach describes the geometry of the structure or the fluid boundary the design element concept. Here, the shape of a body $\boldsymbol{X}$ is approximated by so-called design elements as follows:

$$\boldsymbol{X} = \sum_j \phi_j(\xi) \hat{\boldsymbol{X}}_j \tag{17}$$ eq:geometrical_approach

Here, $\phi_j(\xi)$ is a shape function, $\hat{\boldsymbol{X}}_j$ is the vector of control nodes and ξ represents the reference coordinate. In the design element concept, the variation of the control nodes of the design element is used to vary the shape of the body. The variation of the control nodes position denoted as "mesh-motion" is given as $\boldsymbol{x} = \Delta\hat{\boldsymbol{X}}$

$$\boldsymbol{x} = \sum_j \phi_j(\xi) \hat{\boldsymbol{x}}_j \tag{18}$$ eq:mms

Just as in Finite Elements, where the shape-functions can not only be used to describe the solution field, but also the body's geometry and material parameters, the design element concept can be applied to prescribe parameter distributions and their variations in the optimization process. A simple sketch of the concept is provided in Figure 1
Both approaches have pros and cons, that are quickly discussed in [10], this thesis utilizes the geometrical approach solemnly.

Test abased acid a SD acid

2.3 Analysis Model

The Saddle point formulation of the Lagrangian

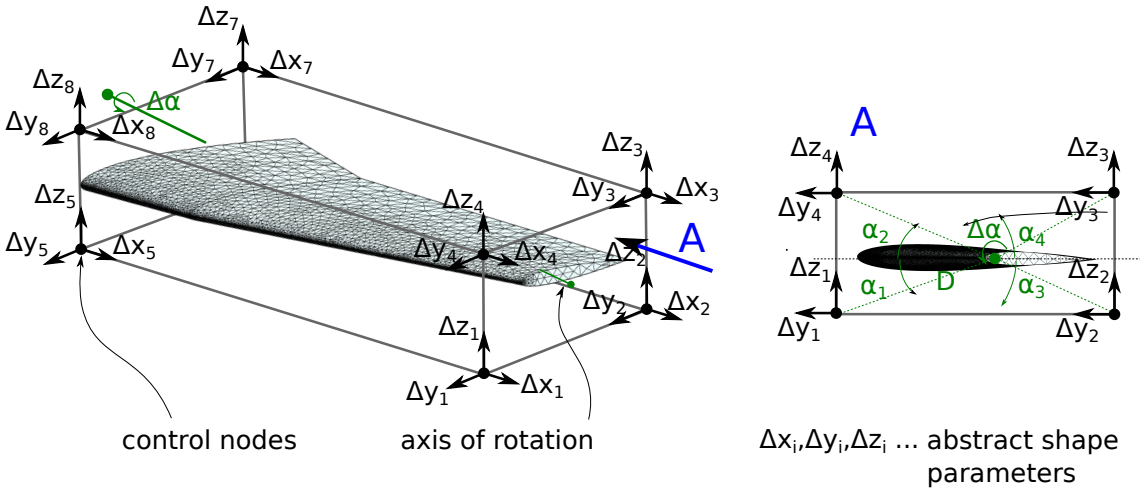


Figure 1: Geometrical approach for the design model as explained in Section 2.2. A NACA type airfoil is exemplarily considered here. The airfoil is embedded in a bounding box defined by eight so-called control nodes. The position of these control nodes can be varied, which alternates the shape of the airfoil according to some interpolation functions defined on the nodes. The 24 displacement unknowns are denoted as "abstract shape parameters". Abstract shape variables can be arbitrarily defined. As an example, a rotation axis is defined through the wing. ***REFORMOLATE: With only one independent abstract variable(α) the whole wing can now be rotated, e.g. to find the optimum angle of attack***

2.4 Aero elastic sensitivity analysis

The SA approach applied in this thesis is based on the work of [13], for deriving the Gloabal Sensitivity Equations (GSE) of coupled systems. As introduced by the authors of[**?**], we utilize the three-field formulation of [2].

The derivative of the optimization criterion q_j, as introduced in Equation (3), with respect to the optimization variable s_i gives:

$$\frac{dq_j}{ds_i} = \frac{\partial q_j}{\partial s_i} + \frac{\partial q_j}{\partial \boldsymbol{u}}\frac{d\boldsymbol{u}}{ds_i} + \frac{\partial q_j}{\partial \boldsymbol{x}}\frac{d\boldsymbol{x}}{ds_i} + \frac{\partial q_j}{\partial \boldsymbol{w}}\frac{d\boldsymbol{w}}{ds_i} \tag{19}$$

$$= \frac{\partial q_j}{\partial s_i} + \begin{bmatrix} \frac{\partial q_j}{\partial \boldsymbol{u}} \\ \frac{\partial q_j}{\partial \boldsymbol{x}} \\ \frac{\partial q_j}{\partial \boldsymbol{w}} \end{bmatrix}^T \cdot \begin{bmatrix} \frac{d\boldsymbol{u}}{ds_i} \\ \frac{d\boldsymbol{x}}{ds_i} \\ \frac{d\boldsymbol{w}}{ds_i} \end{bmatrix} \tag{20}$$

where the partial derivatives, $\frac{\partial q_j}{\partial \boldsymbol{u}}$, $\frac{\partial q_j}{\partial \boldsymbol{x}}$ and $\frac{\partial q_j}{\partial \boldsymbol{w}}$ can be directly evaluated within the discretized structure and fluid model through the relation between structural, aerodynamic design and abstract optimization parameters defied in the design model 2.2.

The cumbersome part are the derivatives $\frac{d\boldsymbol{u}}{ds_i}$, $\frac{d\boldsymbol{x}}{ds_i}$ and $\frac{d\boldsymbol{w}}{ds_i}$ To obtain them, the governing equations (**??**) have to be derived:

$$\begin{bmatrix} \frac{\partial \mathcal{S}}{\partial s_i} \\ \frac{\partial \mathcal{D}}{\partial s_i} \\ \frac{\partial \mathcal{F}}{\partial s_i} \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{\partial \mathcal{S}}{\partial \boldsymbol{u}} & \frac{\partial \mathcal{S}}{\partial \boldsymbol{x}} & \frac{\partial \mathcal{S}}{\partial \boldsymbol{w}} \\ \frac{\partial \mathcal{D}}{\partial \boldsymbol{u}} & \frac{\partial \mathcal{D}}{\partial \boldsymbol{x}} & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathcal{F}}{\partial \boldsymbol{x}} & \frac{\partial \mathcal{F}}{\partial \boldsymbol{w}} \end{bmatrix}}_{\boldsymbol{A}} \begin{bmatrix} \frac{\partial \boldsymbol{u}}{\partial s_i} \\ \frac{\partial \boldsymbol{x}}{\partial s_i} \\ \frac{\partial \boldsymbol{w}}{\partial s_i} \end{bmatrix} = \mathbf{0} \tag{21}$$

eq:governing_eqautions_derivative

In this equations $\frac{\partial \mathcal{S}}{\partial s_i}$ and $\frac{\partial \mathcal{F}}{\partial s_i}$ can be again directly evaluated using the relation specified in the design model. The matrix of first derivatives $\boldsymbol{A}$ is from now on denoted as the "Jacobian of the optimization problem".
Combining the previous two equations, it follows that the total derivative of the optimization criterion with respect to the abstract variables can be expressed as:

$$\frac{dq_j}{ds_i} = \frac{\partial q_j}{\partial s_i} - \underbrace{\begin{bmatrix} \frac{\partial q_j}{\partial \boldsymbol{u}} \\ \frac{\partial q_j}{\partial \boldsymbol{x}} \\ \frac{\partial q_j}{\partial \boldsymbol{w}} \end{bmatrix}^T}_{n_{\boldsymbol{q}} \times n_{eq}} \underbrace{\boldsymbol{A}^{-1}}_{n_{eq} \times n_{eq}} \underbrace{\begin{bmatrix} \frac{\partial \mathcal{S}}{\partial s_i} \\ \frac{\partial \mathcal{D}}{\partial s_i} \\ \frac{\partial \mathcal{F}}{\partial s_i} \end{bmatrix}}_{n_{eq} \times n_{\boldsymbol{s}}} \tag{22}$$

eq:totderiv_optcritBYabsvar

2.4.1 Direct vs. adjoint approach

Equation 22 suggests, that there are two alternatives to compute vector-matrix-vector product above.

Direct approach Firstly, one could first compute the derivatives of the aeroelastic response for each abstract variable and perform the matrix product with $\boldsymbol{A}$:

$$\begin{bmatrix} \frac{d\boldsymbol{u}}{ds_i} \\ \frac{d\boldsymbol{x}}{ds_i} \\ \frac{d\boldsymbol{w}}{ds_i} \end{bmatrix} = -\boldsymbol{A}^{-1} \begin{bmatrix} \frac{\partial \mathcal{S}}{\partial s_i} \\ \frac{\partial \mathcal{D}}{\partial s_i} \\ \frac{\partial \mathcal{F}}{\partial s_i} \end{bmatrix} \quad \text{and then} \quad \texttt{eq:direct\_approach} \tag{23}$$

$$\frac{dq_j}{ds_i} = \frac{\partial q_j}{\partial s_i} - \begin{bmatrix} \frac{\partial q_j}{\partial \boldsymbol{u}} \\ \frac{\partial q_j}{\partial \boldsymbol{x}} \\ \frac{\partial q_j}{\partial \boldsymbol{w}} \end{bmatrix}^T \begin{bmatrix} \frac{d\boldsymbol{u}}{ds_i} \\ \frac{d\boldsymbol{x}}{ds_i} \\ \frac{d\boldsymbol{w}}{ds_i} \end{bmatrix} \tag{24}$$

Where the total complexity can be approximated as $\mathcal{O}(n_{eq}^2 n_{\boldsymbol{s}} + n_{\boldsymbol{q}} n_{eq} n_{\boldsymbol{s}})$

Adjoint approach Secondly, one could also first compute the derivatives of the optimization criteria and multiply with the Jacobian before substituting this into Equation (22):

$$\begin{bmatrix} \boldsymbol{a_u} \\ \boldsymbol{a_x} \\ \boldsymbol{a_w} \end{bmatrix} = \boldsymbol{A}^{-T} \begin{bmatrix} \frac{\partial q_j}{\partial \boldsymbol{u}} \\ \frac{\partial q_j}{\partial \boldsymbol{x}} \\ \frac{\partial q_j}{\partial \boldsymbol{w}} \end{bmatrix} \quad \texttt{eq:firststep\_adjoint} \tag{25}$$

$$\frac{dq_j}{ds_i} = \frac{\partial q_j}{\partial s_i} - \begin{bmatrix} \boldsymbol{a_u} \\ \boldsymbol{a_x} \\ \boldsymbol{a_w} \end{bmatrix}_j^T \begin{bmatrix} \frac{\partial \mathcal{S}}{\partial s_i} \\ \frac{\partial \mathcal{D}}{\partial s_i} \\ \frac{\partial \mathcal{F}}{\partial s_i} \end{bmatrix} \tag{26}$$

Where the total complexity can be approximated as $\mathcal{O}(n_{eq}^2 n_{\boldsymbol{q}} + n_{\boldsymbol{q}} n_{eq} n_{\boldsymbol{s}})$

If one or the other approach is to be preferred depends in on the optimization setup, particularly the number of optimization criteria and the number of optimization variables. Looking at the orders above, one can conclude that if the number of abstract parameters $n_{\boldsymbol{s}}$ is smaller than the number of optimization criteria, the

direct approach is more efficient, otherwise the adjoint approach is to be preferred. Additionally the on can argue, that the relevant term in the orders above is the one with n_{eq}^2 since it dominates the sum.

3 Direct Sensitivity Analysis for the Euler equations

The A matrix for the direct approach in ALE formulation looks like

$$\boldsymbol{A} = \begin{bmatrix} \frac{\partial \mathcal{S}}{\partial \boldsymbol{u}} & \frac{\partial \mathcal{S}}{\partial \boldsymbol{x}} & \frac{\partial \mathcal{S}}{\partial \boldsymbol{w}} \\ \frac{\partial \mathcal{D}}{\partial \boldsymbol{u}} & \frac{\partial \mathcal{D}}{\partial \boldsymbol{x}} & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathcal{F}}{\partial \boldsymbol{x}} & \frac{\partial \mathcal{F}}{\partial \boldsymbol{w}} \end{bmatrix} = \begin{bmatrix} \mathbf{K} & \frac{\partial \boldsymbol{P}_T}{\partial \boldsymbol{x}} & \frac{\partial \boldsymbol{P}_T}{\partial \boldsymbol{w}} \\ \begin{bmatrix} \mathbf{K}_{\Omega\Gamma}\mathbf{T}_u \\ \mathbf{T}_u \end{bmatrix} & \begin{bmatrix} \mathbf{K}_{\Omega\Omega} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathbf{F}_2}{\partial \boldsymbol{x}_\Omega} & \boldsymbol{H}_2 \end{bmatrix} \quad \texttt{eq:Amatrix\_ALE} \tag{27}$$

Where, $\boldsymbol{H}_2$ is the Jacobian of the second order row flux. It shall be noted that constructing this Jacobian is not a trivial issue and takes up a lot of computational resources, especially for FV, as described in [2]. Investigation into whether this term can be approximated at first order were carried out in [9] and [10].
Furthermore, [11] considered replacing the two mesh motion related matrices $\frac{\partial \boldsymbol{P}_T}{\partial \boldsymbol{x}}$ and $\frac{\partial \mathbf{F}_2}{\partial \boldsymbol{x}_\Omega}$ by a transpirational boundary condition. The consequences of this approach are also investigated in [9] and [10].
This thesis, however, dow not use any of this simplifications.
The derivation of the sensitivities, can be achieved in a staggered scheme, very similar to the one, described in **??TODO**. It consists of five steps.

1) Update the structural displacement sensitivity to a new time step
BY differentiating equations (71) and (75) and applying an under relaxation, we can obtain

$$\frac{\mathrm{d}\boldsymbol{u}^{(n)}}{\mathrm{d}s_i} = (1-\theta)\frac{\mathrm{d}\boldsymbol{u}^{(n)}}{\mathrm{d}s_i} + \theta\frac{\mathrm{d}\bar{\boldsymbol{u}}}{\mathrm{d}s_i} \quad \texttt{eq:underrelax\_structdisp} \tag{28}$$

where $\bar{\boldsymbol{u}}$ is obtained from:

$$\mathbf{K}\frac{\mathrm{d}\bar{\boldsymbol{u}}}{\mathrm{d}s_i} = \frac{\partial TODO}{\partial s_i} + \frac{\partial \mathbf{T}_u{}^{(n)}}{\partial s_i} - \frac{\partial \mathbf{K}}{\partial s_i}\boldsymbol{u} \quad \texttt{eq:fictious\_structdisp} \tag{29}$$

2) Transfer sensitivity of structure motion to the interface

$$\frac{\mathrm{d}\boldsymbol{u}_T^{(n)}}{\mathrm{d}s_i} = \mathsf{T}_u \frac{\mathrm{d}\boldsymbol{u}^{(n)}}{\mathrm{d}s_i} \qquad \texttt{eq:interface\_projections} \tag{30}$$

3) Compute derivative of fluid mesh motion The fluid mesh motion is computed by solving the pseudo Dirichlet problem as described in [**?**]. By design, the fictions stiffness matrix $\bar{\mathsf{K}}$ does not depend on the abstract optimization variables $\boldsymbol{s}$

$$\bar{\mathsf{K}}_{\Omega\Omega} \frac{\mathrm{d}\boldsymbol{x}_\Omega^{(n)}}{\mathrm{d}s_i} = -\bar{\mathsf{K}}_{\Omega\Gamma} \frac{\mathrm{d}\boldsymbol{x}_\Gamma^{(n)}}{\mathrm{d}s_i} \qquad \texttt{eq:mms\_domain} \tag{31}$$

with

$$\frac{\mathrm{d}\boldsymbol{x}_\Gamma^{(n)}}{\mathrm{d}s_i} = \frac{\mathrm{d}\boldsymbol{x}_\Gamma^{(n)}}{\mathrm{d}s_i} \tag{32}$$

4) Compute the sensitivity of the fluid state variables The derivatives of the fluid state variables are computed by

$$\boldsymbol{H}_2 \frac{\mathrm{d}\boldsymbol{w}^{(n+1)}}{\mathrm{d}s_i} = \frac{\partial \boldsymbol{F}_2}{\partial s_i} - \frac{\partial \boldsymbol{F}_2}{\partial \boldsymbol{x}} \frac{\mathrm{d}\boldsymbol{x}^{(n)}}{\mathrm{d}s_i} \tag{33}$$

5) Compute the sensitivity of the structure load vector The derivative of the fluid load with respect to the abstract variables can be computed by the 3rd of Equations (23) with the definition of $\boldsymbol{A}$ as specified in (27).

$$\frac{\partial \boldsymbol{P}_F{}^{(n+1)}}{\partial s_i} = \frac{\partial \boldsymbol{P}_F{}^{(n+1)}}{\partial \boldsymbol{x}} \frac{\mathrm{d}\boldsymbol{x}^{(n)}}{\mathrm{d}s_i} + \frac{\partial \boldsymbol{P}_F{}^{(n+1)}}{\partial \boldsymbol{w}} \frac{\mathrm{d}\boldsymbol{w}^{(n)}}{\mathrm{d}s_i} \qquad \texttt{eq:deriv\_floadBYabsvar} \tag{34}$$

and compute project it onto the structure via

$$\frac{\partial \boldsymbol{P}_T{}^{(n+1)}}{\partial s_i} = \mathsf{T}_p \frac{\partial \boldsymbol{P}_F{}^{(n+1)}}{\partial s_i} \tag{35}$$

The convergence of the staggered algorithm can be monitored via

$$\left\| \mathsf{K} \frac{\mathrm{d}\bar{\boldsymbol{u}}^{(n+1)}}{\mathrm{d}s_i} - \frac{\partial TODO}{\partial s_i} - \frac{\partial \boldsymbol{P}_T{}^{(n+1)}}{\partial s_i} + \frac{\partial \mathsf{K}}{\partial s_i} \right\|_2 \leq \epsilon^{SA} \left\| \mathsf{K} \frac{\mathrm{d}\bar{\boldsymbol{u}}^{(0)}}{\mathrm{d}s_i} - \frac{\partial TODO}{\partial s_i} - \frac{\partial \boldsymbol{P}_T{}^{(0)}}{\partial s_i} + \frac{\partial \mathsf{K}}{\partial s_i} \right\|_2 \tag{36}$$

$$\left\| \boldsymbol{H}_2 \frac{\mathrm{d}\boldsymbol{w}^{(n+1)}}{\mathrm{d}s_i} + \frac{\partial \boldsymbol{F}_2}{\partial s_i} + \frac{\partial \boldsymbol{F}_2}{\partial \boldsymbol{x}} \frac{\mathrm{d}\boldsymbol{x}^{(n+1)}}{\mathrm{d}s_i} \right\|_2 \leq \epsilon^{SA} \left\| \boldsymbol{H}_2 \frac{\mathrm{d}\boldsymbol{w}^{(0)}}{\mathrm{d}s_i} + \frac{\partial \boldsymbol{F}_2}{\partial s_i} + \frac{\partial \boldsymbol{F}_2}{\partial \boldsymbol{x}} \frac{\mathrm{d}\boldsymbol{x}^{(0)}}{\mathrm{d}s_i} \right\|_2 \tag{37}$$

4 Adjoint Sensitivity analysis for the Euler equations

The adjoint SA follows the same scheme as the direct one
Equation (25) can be written as:

$$\begin{bmatrix} \mathbf{K} & \begin{bmatrix} \frac{\partial \boldsymbol{P}_T}{\partial \boldsymbol{x}_\Omega} & \frac{\partial \boldsymbol{P}_T}{\partial \boldsymbol{x}_\Gamma} \end{bmatrix} & \frac{\partial \boldsymbol{P}_T}{\partial \boldsymbol{w}} \\ \begin{bmatrix} \mathbf{K}_{\Omega\Gamma}\mathbf{T}_u \\ \mathbf{T}_u \end{bmatrix} & \begin{bmatrix} \mathbf{K}_{\Omega\Omega} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} & \mathbf{0} \\ \mathbf{0} & \begin{bmatrix} \frac{\partial \mathbf{F}_2}{\partial \boldsymbol{x}_\Omega} & \frac{\partial \mathbf{F}_2}{\partial \boldsymbol{x}_\Gamma} \end{bmatrix} & \boldsymbol{H}_2 \end{bmatrix}^T \begin{bmatrix} \boldsymbol{a_u} \\ \begin{bmatrix} \boldsymbol{a}_{\boldsymbol{x}_\Omega} \\ \boldsymbol{a}_{\boldsymbol{x}_\Gamma} \end{bmatrix} \\ \boldsymbol{a_w} \end{bmatrix} = \begin{bmatrix} \frac{\partial q_j}{\partial \boldsymbol{u}} \\ \begin{bmatrix} \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_\Omega} \\ \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_\Gamma} \end{bmatrix} \\ \frac{\partial q_j}{\partial \boldsymbol{w}} \end{bmatrix} \quad \text{eq:adjoint\_equation} \tag{38}$$

TODO check if the index j is really required here! A stated earlier, the matrices $\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}}$ and $\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{w}}$ can be computed analytically. As for $\boldsymbol{H}_2$, we follow the methodology outlined in [7] for evaluating ans storing it efficiently as the product of flux operators. Again the staggered procedure for solving the adjoint state problem shares the same computational kernels with the partitioned aeroelastic scheme described in [**?**]

1) Update the adjoint structure displacement to the new time step

$$\boldsymbol{a_u}^{(n+1)} = (1-\theta)\boldsymbol{a_u}^{(n)} + \theta \bar{\boldsymbol{a}_{\boldsymbol{u}}}^{(n+1)} \tag{39}$$

$$\tag{40}$$

where $\bar{\boldsymbol{a}_{\boldsymbol{u}}}^{(n+1)}$ is obtained from

$$\mathbf{K}\bar{\boldsymbol{a}_{\boldsymbol{u}}}^{(n+1)} = \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{u}} - \mathbf{K}_{\Omega\Gamma}\mathbf{T}_u \boldsymbol{a}_{\boldsymbol{x}\Omega}{}^{(n)} + \mathbf{T}_u \boldsymbol{a}_{\boldsymbol{x}\Gamma}{}^{(n)} \tag{41}$$

TODO derive this equation

2) Compute the adjoint fluid state by solving

$$\boldsymbol{H}_2{}^T \boldsymbol{a_w}^{(n+1)} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{w}} + \frac{\partial \boldsymbol{P}_T}{\partial \boldsymbol{w}^T} \boldsymbol{a_u}^{(n+1)} \tag{42}$$

TODO derive this equation Again, $\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{w}}$ is computed analytically from the relations defined in the design and aeroelastic model.

3) Compute adjoint mesh motion in domain and on the interface

$$\bar{\mathsf{K}}_{\Omega\Omega}\boldsymbol{a}_{\boldsymbol{x}\Omega}{}^{(n+1)} = \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_\Omega} - \frac{\partial \boldsymbol{P}_T}{\partial \boldsymbol{x}_\Omega}\boldsymbol{a}_{\boldsymbol{u}}{}^{(n+1)} - \frac{\partial \mathsf{F}_1{}^T}{\partial \boldsymbol{x}_\Omega}\boldsymbol{a}_{\boldsymbol{x}}{}^{(n+1)} \text{ in } \Omega \tag{43}$$

TODO check equations And the adjoint mesh motion on the interface is computed as $\boldsymbol{a}_{\boldsymbol{x}\Gamma}{}^{(n+1)}$

$$\boldsymbol{a}_{\boldsymbol{x}\Gamma}{}^{(n+1)} = \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{Gamma}} + \frac{\partial \boldsymbol{P}_T}{\partial \boldsymbol{x}_\Gamma}\boldsymbol{a}_{\boldsymbol{u}}{}^{(n+1)} - \frac{\partial \mathsf{F}_2{}^T}{\partial \boldsymbol{x})_\Gamma} \text{ on } \Gamma \tag{44}$$

where $\dfrac{\partial \boldsymbol{q}}{\partial \boldsymbol{w}}$ is computed analytically.

The convergence of the staggered adjoint optimization algorithm can be monitored via

$$\left\| \boldsymbol{R}_{\boldsymbol{q}} - \boldsymbol{A}^T \boldsymbol{a}^{(n+1)} \right\| \leq \epsilon^{SA} \left\| \boldsymbol{R}_{\boldsymbol{q}} \right\| \tag{45}$$

5 Eulerian and Lagrangian view

In time dependent analytical physics there are two basic concepts of how to view a system of interest: the Eulerian and the Lagrangian approach. It shall be stressed here, that Eulerian view and Euler equations are two totally separate things.
One view this issue from different perspectives, but what the difference points down to is the interpretation of the time derivative. Since we will be using the different derivatives frequently in this thesis, a short introduction shall be provided.

5.1 Material derivative

The material derivative of a property, is a derivative(rate of change), that follows a specific particle 'p'. That means that at every time instance the material derivative gives as the current value of the property of a specific particle. Since in general, a particle will move its position in time quite a lot, some mean is required to track the particles motion. The material derivative is also known as the Lagrangian concept, and is commonly used in Solid mechanics, since the motion of the solid particles is typically little, or at least nearly uniform over the body, which makes position tracking easy.

5.2 Eulerian derivative

The Eulerian derivative takes a different approach. It refers to a fixed frame of reference in the domain, and gives the rate of change at that specific position. Due to the particle movement, it typically refers to different particles at different times. Material and Eulerian derivatives can be linked via the so-called convective rate of change as:

$$\underbrace{\frac{D\boldsymbol{G}}{Dt}}_{\text{Lagrangian rate of change}} = \underbrace{\frac{\partial \boldsymbol{G}}{\partial t}}_{\text{Eulerian rate of change}} + \underbrace{\boldsymbol{v} \cdot \nabla \boldsymbol{G}}_{\text{Convective rate of change}} \tag{46}$$

In fluid mechanics the Eulerian concept is typically proffered.
However, when looking at fluid-structure interaction problems, the interface is typically moving. To account for that motion in the fluid, one either has to come up with some cell-intersection approach, or the fluid-mesh has to be deformed to.
An appealing and straightforward approach to do so, is the so called Arbitrary Lagrangian Eulerian(ALE) approach. As the name suggest, it is a hybrid between Eulerian and Lagrangian point of view. The ALE approach allows to utilize the best of both approaches. The mesh can either be fixed, moved with the continuum in an Eulerian manner, or be moved in some arbitrarily specified manner in-between. The ALE method can handle larger distortions than a pure Eulerian would, while stiff allowing to keep interface continuity between the structure an the fluid, such that no cell intersection will be necessary.
In this thesis, both Eulerian and ALE methods will be covered. In general, no recommendation on to prefer one or the other method can be given. Instead, one

should always be aware of the strength and weaknesses of either method and choose the appropriate one for a specific problem accordingly.

6 Basic equation of fluid mechanics

The physics of fluid motion are governed by the so-called Navier Stokes Equations (NSE). These equations were derived independently of one another by Claude-Louis Navier and George Gabriel Stokes as a generalization of the Euler-equations that now includes viscosity effects. In general one distinguished the incompressible and the compressible NSE. The validity of their application depends on the problem setup. As a rule of thumb, flows with Mach-numbers lower than 0.3 can usually be safely approximated by the incompressible NSE.

6.1 Incompressible Navier Stokes Equations

The incompressible NSE are derived by putting a few assumptions on the Cauchy stress tensor:

- Galilean invariance of the fluid stress tensor
- Isotropy of the fluid
- The stokes stress constitutive equation applies: $\boldsymbol{\tau} = 2\mu\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} = \frac{1}{2}(\nabla\boldsymbol{v}({}^{T}\nabla\boldsymbol{v}))$

In convective form, the incompressible NSE can be written as

$$\overbrace{\underbrace{\frac{\partial \boldsymbol{v}}{\partial t}}_{\text{Variation}} + \underbrace{(\boldsymbol{v}\cdot\nabla)\boldsymbol{v}}_{\text{Convection}}}^{\text{Inertia per volume}} - \overbrace{\underbrace{\nu\nabla^2\boldsymbol{v}}_{\text{Diffusion}} = -\underbrace{\nabla\boldsymbol{w}}_{\text{Internal source}}}^{\text{Divergence of stress}} + \underbrace{g}_{\text{Extrenal source}} \tag{47}$$

eq:nsg_incompressible

where $\boldsymbol{w}$ is the specific thermodynamic work. It shall be noted, that the pressure does not appear in this equations. Instead, it is recovered from the solved velocity field.

6.2 Compressible Navier Stokes Equations

The compressible NSE are derived by making the following assumptions:

- Galilean invariance of the stress tensor
- Linearity of the stress in the velocity gradient: $\boldsymbol{\tau}(\nabla\boldsymbol{v}) = \boldsymbol{C} : (\nabla\boldsymbol{v})$, where $\boldsymbol{C}$ is called the viscosity or elasticity tensor.
- Isotopy of the fluid

The compressible Navier stokes equations can be written as

$$\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{v} = -\frac{\partial 1}{\partial \rho} \nabla p + \nu \nabla^2 \boldsymbol{v} + \frac{1}{3} \nu \nabla (\nabla \cdot \boldsymbol{v}) + g \quad \text{eq:nsg\_compressible} \tag{48}$$

where it shall be noted that for the special case of an incompressible flow, the volume of the fluid elements is constant, resulting in a solenoidal velocity field. Thus $\nabla \cdot \boldsymbol{v} = \boldsymbol{0}$ and $\nabla p = 0$, which gives equation 47.

6.3 Conservative form

For the purpose of FV discretization one usually bring the equations into a so-called "conservative form".

$$\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}t} + \nabla \cdot f(\boldsymbol{\xi}) = \boldsymbol{0} \quad \text{eq:conservative\_form\_general} \tag{49}$$

(50)

This can then be transformed into an integral form using the divergence theorem

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_V \boldsymbol{\xi} dV + \int_{\partial V} f(\boldsymbol{\xi}) \cdot \boldsymbol{n} \, dS \quad \text{eq:conservation\_form\_integral} \tag{51}$$

This equations states that the rate of change of the integral of the quantity $\boldsymbol{\xi}$ over an arbitrary control volume V is equal to the negative of the "flux" through the boundary of the control volume ∂V. A simple choice for f would be $f(\xi) = \xi \boldsymbol{v}$, which mean that the quantity $\boldsymbol{\xi}$ follows the fluid field and gives the so-called "advection equation".
Although one form is derived from the other, these two are not equivalent. Particularly, it is possible to find solution to the integral equations that are non-differentiable and therefor not a solution to the differential form. This leads to so called "weak solutions" and is the cause for many numerical difficulties in the finite volume simulation of such problems.

This thesis focuses on the compressible NSE only. They can be brought to conservative form as:

$$\frac{\partial \boldsymbol{w}}{\partial t} + \nabla \cdot \mathcal{F}(\boldsymbol{w}) + \nabla \cdot \mathcal{G}(\boldsymbol{w}) = \boldsymbol{0} \quad \text{eq:nsg\_conservative form} \tag{52}$$

where comparing to the general conservative form, one can see that $\boldsymbol{\xi} = \boldsymbol{w}$ and $f(\boldsymbol{\xi}) = \mathcal{F}(\boldsymbol{w}) + \mathcal{G}(\boldsymbol{w})$

Here,

$$\boldsymbol{w} = \begin{bmatrix} \rho \\ \rho \boldsymbol{v} \\ E \end{bmatrix} \quad \text{eq:fstate\_definition} \tag{53}$$

is the so-called fluid state vector, with

$$\boldsymbol{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{54}$$ eq:fluidvel_definition

denoting the fluid velocity vector.
The total energy E can be computed as a sum of the so called internal energy and the kinematic energy due to fluid velocity as

$$\underbrace{E}_{\text{total energy}} = \underbrace{\rho e}_{\text{internal energy}} + \underbrace{\frac{1}{2}\boldsymbol{v}^T\boldsymbol{v}}_{\text{kinematic energy}} \tag{55}$$ eq:energytot

Bringing the compressible NSE (48) into the conservative form (52), one can derive $\mathcal{F}$ and $\mathcal{G}$ as

$$\mathcal{F} = (\frac{1}{\rho}\boldsymbol{w}\boldsymbol{w}^T + p + \boldsymbol{R}_3\boldsymbol{w}^T) \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_3 \\ \mathbf{0} \end{bmatrix} \tag{56}$$ eq:fluxesconv

$$= \boldsymbol{w}\boldsymbol{v}^T + p \begin{bmatrix} 0 \\ \mathbf{I} \\ \boldsymbol{v}^T \end{bmatrix} \tag{57}$$

$$= \begin{bmatrix} \rho\boldsymbol{v}^T \\ \rho(\boldsymbol{v}\boldsymbol{v}^T) + p\mathbf{I} \\ (E+p)\boldsymbol{v}^T \end{bmatrix} \tag{58}$$

$$= \begin{bmatrix} \rho v_x & \rho v_y & \rho v_z \\ p + \rho v_x^2 & \rho v_x v_y & \rho v_x v_z \\ \rho v_y v_x & p + \rho v_y^2 & \rho v_y v_z \\ \rho v_z v_x & \rho v_z v_y & p + \rho v_z^2 \\ v_x(E+p) & v_x(E+p) & v_x(E+p) \end{bmatrix} \tag{59}$$

depending on the algorithm, one or another form can be advantageous.

The diffusive fluxes can be written as

$$\mathcal{G} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\tau} \\ \boldsymbol{\tau}\boldsymbol{v} + \boldsymbol{q} \end{bmatrix} \tag{60}$$ eq:fluxes_diff

$$= \begin{bmatrix} 0 & 0 & 0 \\ -\tau_{xx} & -\tau_{yx} & -\tau_{zx} \\ -\tau_{xy} & -\tau_{yy} & -\tau_{zy} \\ -\tau_{xz} & -\tau_{yz} & -\tau_{zz} \\ -q_x - v_x\tau_{xx} - v_x\tau_{yx} - v_x\tau_{zx} & -q_y - v_x\tau_{xy} - v_x\tau_{yy} - v_x\tau_{zy} & -q_z - v_x\tau_{xz} - v_x\tau_{yz} - v_x\tau_{zz} \end{bmatrix}^T \tag{61}$$

The definition of the stress tensor depends of the fluid model used. For the simple case of a Newtonian fluid, it can be written as

$$\boldsymbol{\tau} = \mu(\nabla \boldsymbol{v} + (\nabla \boldsymbol{v})^T) + \lambda(\nabla \cdot \boldsymbol{v})\mathbf{I} \tag{62}$$ eq:fluidstress

For the relation between Temperature and heat flux an assumption has to be made. Most often, a simple Fourier's law is assumed

$$\boldsymbol{q} = -k\nabla T \tag{63}$$ eq:heatflux

6.4 Euler equations

The euler equations are a simplified form of the Navier-Stokes Equations (47)(48), where the viscous effects are neglected by setting $\mathcal{G} = \mathbf{0}$.

$$\frac{\partial \boldsymbol{w}}{\partial t} + \nabla \cdot \mathcal{F}(\boldsymbol{w}) = \mathbf{0} \tag{64}$$ eq:euler

The Euler equations are appropriate for a wide range of applications. A typical indicator is the Reynolds number, which describes the ratio between inertial forces and viscous forces:

$$RE = \frac{\rho \boldsymbol{v} L}{\mu} = \frac{\boldsymbol{v} L}{\nu} \tag{65}$$ eq:reynolds

where L describes a characteristic length.

A high Reynolds number thus indicates that the flow is dominated by inertial forces, thus the Euler Equations should give satisfying results. However, an Euler flow lacks the ability to represent stick wall boundary conditions, thus it is unable to re[resent boundary layers.

6.5 Equations of State

Looking at the above Equations, one might notice that the number of unknowns is greater than the number of Equations. Particularly, the pressure only appears in Equation (56) and is not linked to any other equations. This problem is solved by introducing an Equation of State(Equation of State (EOS)) that relates pressure, internal energy and density. The equation of state depends on the fluid model, some well-known ones are: Perfect Gas(PG), Stiffened Gas(Stiffend Gas (SG)), Jones-Wilkins-Lee (JWL).
For the simples one, PG, the EOS can be written as

$$p = (\gamma - 1)\rho e \tag{66}$$ eq:eos_pg

6.6 RANS Equations

The RANS Equations are time-averaged equations of motion for the fluid.

$$\boldsymbol{w} \to \bar{\boldsymbol{w}} = \lim_{T \to \infty} \frac{1}{T} \int_{t^0}^{t^0+T} \boldsymbol{w} dt \tag{67}$$ eq:rans_averaging

The main idea of the approach is to decompose an instantaneous quantity into time-averaged and fluctuating components

$$\boldsymbol{w} = \underbrace{\bar{\boldsymbol{w}}}_{time-average} + \underbrace{\boldsymbol{w}'}_{fluctuation} \tag{68}$$ eq:rans_decomposition

When substituting this decomposition back into the NSE(and injecting several other approximations), a closure problem induced by the arising non-linear Reynolds stress term $RE_{ij} = -v_i'\bar{v}_j'$ arises. Additional modeling is therefore requires to close the RANS equations, which has led to many different turbulence models.

Whatever turbulence model is chosen, the fluid-state vector is augmented by the m parameters of the turbulence model

$$\boldsymbol{w}_{RANS} \leftarrow \begin{bmatrix} \boldsymbol{w} \\ \chi_1 \\ \vdots \\ \chi_m \end{bmatrix} = \begin{bmatrix} \rho \\ \rho \boldsymbol{v}^T \\ E \\ \chi_1 \\ \vdots \\ \chi_m \end{bmatrix} \tag{69}$$

The RANS equation can then be written as

$$\frac{\partial \bar{\boldsymbol{w}}}{\partial t} + \nabla \cdot \mathcal{F}(\bar{\boldsymbol{w}}) + \nabla \cdot \mathcal{G}(\bar{\boldsymbol{w}}) = \mathsf{S}(\bar{\boldsymbol{w}}, \chi_1, \cdots, \chi_m) \tag{70}$$ eq:rans_eqautions

7 Aeroelastic Optimization

To describe the aeroelastic response of a mechanical system under fluid load, as well as the interaction between fluid and structure itself, a three-dimensional second order finite volume code [4] was coupled with a second order accurate finite element code[5] according to the popular three-field formulation of [2]. It's discrete steady state form can be written as follows:

$$\begin{aligned} &\text{state equation of the structure:} && \mathcal{S}(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{0} && \text{eq:3field\_structure} \quad (71)\\ &\text{governing equation of meshmotion:} && \mathcal{D}(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{x}) = \boldsymbol{0} && \text{eq:3field\_mesh} \quad (72)\\ &\text{state equation of the fluid:} && \mathcal{F}(\boldsymbol{s}, \boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{0} && \text{eq:3field\_fluid} \quad (73)\\ & && && (74) \end{aligned}$$

State equation of the structure: For a simple linear case, that is assumed in this thesis, the structure state equation can be written as

$$\mathcal{S} = \mathsf{K}\boldsymbol{u} - \boldsymbol{P}(\boldsymbol{x}, \boldsymbol{w}) \qquad \text{eq:eos\_struct} \quad (75)$$

where K represents the finite element stiffness matrix and $\boldsymbol{P}$ denotes the external load vector that combines aerodynamic loads $\boldsymbol{P}_F$ that are inflected to the structure by the fluid, and gravity loads $\boldsymbol{P}_0$

$$\boldsymbol{P} = \boldsymbol{P}_0 + \boldsymbol{P}_F \quad (76)$$

Governing equation of the mesh motion: In an Arbitrary Lagrangian Eulerian formulation, the fluid mesh deforms with the structure. A governing equation is thus required to describe that deformation; Typically a simple linear pseudo finite element approach or a spring analogy method [1] is used to do so. A Dirichlet problem is solved to move the mesh.

$$\mathcal{D} = \bar{\mathsf{K}}\boldsymbol{x} \quad \text{with} \quad \boldsymbol{x} = \boldsymbol{u} \text{ on } \Gamma_{F|S} \qquad \text{eq:eos\_mesh} \quad (77)$$

State equation of the fluid: The state equation of the fluid depends on the flow model used, as well as ton whether a Eulerian or Lagrangian approach is used. A quick overview is provided in Section 6.
Please note, that the frame of reference(Eulerian, Lagrangian, Arbitrary Lagrangian Eulerian) has nothing to do with the fluid equation type.

The state equation of the fluid(Equation three in (**??**) for can be expressed as

$\mathcal{F} =$

Eulerian	EULER	$\mathsf{F}_2(\boldsymbol{x}, \boldsymbol{w})$
	NSG	$\mathsf{F}_2(\boldsymbol{x}, \boldsymbol{w}) + \mathsf{G}_2(\boldsymbol{x}, \boldsymbol{w})$
	RANS	$\mathsf{F}_2(\boldsymbol{x}, \boldsymbol{w}) + \mathsf{G}_2(\boldsymbol{x}, \boldsymbol{w}) - \mathsf{S}(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{\chi})$
ALE	EULER	$(\mathsf{A}(\dot{\boldsymbol{x}})\boldsymbol{w}) + \mathsf{F}_2(\boldsymbol{x}, \boldsymbol{w})$
	NSG	$(\mathsf{A}(\dot{\boldsymbol{x}})\boldsymbol{w}) + \mathsf{F}_2(\boldsymbol{x}, \boldsymbol{w}) + \mathsf{G}_2(\boldsymbol{x}, \boldsymbol{w})$
	RANS	$(\mathsf{A}(\dot{\boldsymbol{x}})\boldsymbol{w}) + \mathsf{F}_2(\boldsymbol{x}, \boldsymbol{w}) + \mathsf{G}_2(\boldsymbol{x}, \boldsymbol{w}) - \mathsf{S}(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{\chi})$

(78)

where $\mathbf{F}_2$ denotes the vector of Roe fluxes resulting from a second-order finite volume discretization of the convective part of the NSE 8.2, $\mathbf{G}_2$ likewise denotes a second-order FV discretization oft the diffusive part, $\mathbf{A}$ denotes the matrix of cell volumes and $\mathbf{S}$ denotes the matrix resulting from the additional turbulence closures in the RANS equations(see Section**??**).
The above table also gives a very nice overview over the different equation types and the used simplifications. The Euler equations are obviously obtained by neglecting the diffusive part of the NSE. One can also see that the RANS equations are more difficult than the NSE themselves. However, as explained in **TODO**, the advantage is that RANS can operate on much coarser meshes that do not have to resolve the small scale turbulence phenomena, which makes the more numerically efficient overall in many applications.

7.1 Staggered algorithm

The three-field couples system of equations **TODO**, can be solved very efficiently with an iterative staggered procedure, such as the one proposed by by [**?**]. This work utilizes the second-order staggered algorithm described in Figure 2.

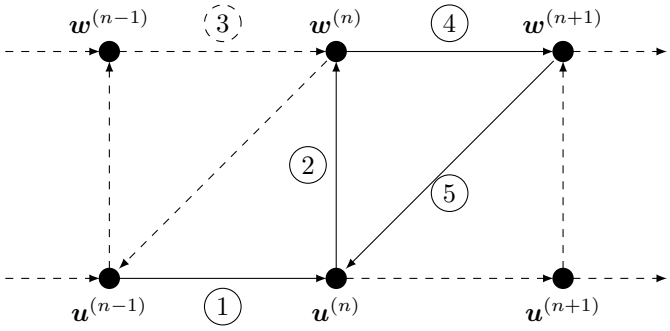


Figure 2: Scheme for the staggered algorithm. Th picture depicts the sequential structure($\boldsymbol{u}$) and fluid($\boldsymbol{w}$) solves. First, the structure is updated to a new time step. With this information, a corresponding fluid solution is obtained. After the fluid is progressed in time, the structure solutions is being corrected with that information. A detailed description of all steps follows below.

① Determining structural response For a given external load $\boldsymbol{P}^{(n)}$, the structural response $\boldsymbol{u}^{(n)}$ is determined by solving the state equation of the structure **TODO**. To increase stability, and under-relaxation is typically performed:

$$\boldsymbol{u}^{(n)} = (1-\theta)\boldsymbol{u}^{(n-1)} + \theta\tilde{\boldsymbol{u}} \tag{79}$$

$$\tilde{\boldsymbol{u}} = \mathbf{K}^{-1}\boldsymbol{P}^{(n)} \tag{80}$$

Typically, the relaxation factor is chosen as $0.5 \leq \theta \leq 0.9$

② Motion transfer to wet surface The motion of the wet surface of the structure has to be transformed to the fluid.

$$\boldsymbol{u}_T^{(n)} = \mathbf{T}_u \boldsymbol{u}^{(n)} \tag{81}$$

where $\mathbf{T}_u$ is a transfer matrix that also accounts for potentially non-matching meshes REF32.

③ Fluid mesh motion update The fluid mesh motion is then updated by solving an auxiliary pseudo Dirichlet problem on the fluid mesh. This step is the very heart of the ALE algorithm.

$$\bar{\mathbf{K}}\boldsymbol{x}^{(n)} = 0 \qquad \boldsymbol{x}^{(n)} = \boldsymbol{u}_T^{(n)} \text{ on } \Gamma_{F|S} \tag{82}$$

where the Dirichlet conditions are introduced into the system via a decomposition:

$$\bar{\mathbf{K}} = \begin{bmatrix} \bar{\mathbf{K}}_{\Omega\Omega} & \bar{\mathbf{K}}_{\Omega\Gamma} \\ \bar{\mathbf{K}}_{\Gamma\Omega} & \bar{\mathbf{K}}_{\Gamma\Gamma} \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_\Omega \\ \boldsymbol{x}_\Gamma \end{bmatrix} \tag{83}$$

Here, the subscript Γ denotes the fluid grid points at the FSI interface and the subscript Ω denotes the remainder. Hence, the fluid mesh is updated in two steps, by first transferring the structure motion to the fluid interface, and then solving the auxiliary pseudo Dirichlet problem:

$$\boldsymbol{x}_\Gamma^{(n)} = \mathbf{T}_u \boldsymbol{u}^{(n)} \tag{84}$$ eq:mms_update_1

$$\bar{\mathbf{K}}_{\Omega\Omega}\boldsymbol{x}_\Omega^{(n)} = -\bar{\mathbf{K}}_{\Omega\Gamma}\boldsymbol{x}^{(n)} \tag{85}$$ eq:mms_update_2

It shall be noted that an exact solution of the preceding equation is not required. In fact it is an arbitrarily chosen auxiliary pseudo-problem anyway. Therefor, a valid update of the mesh motion, that does not produce crossovers is enough. Such an update can be obtained at low cost by allying a few preconditioned conjugate gradient(PCG) iterations to Equation (85).

④ Update fluid state vector After the mesh has been deformed, the fluid state$\mathbf{w}$ vector can be updated accordingly.

Eulerian	EULER	$\mathbf{F}_2(\boldsymbol{w}^{(n)}) + \frac{\partial \mathbf{F}_2}{\partial \boldsymbol{w}}(\boldsymbol{w}^{(n+1)} - \boldsymbol{w}^{(n)})$
	NSG	1
	RANS	1
ALE	EULER	1
	NSG	1
	RANS	1

$= 0$ (86)

t

⑤ Determining structural responses TODO a lot of stuff is missing here!

8 The Finite Volume method

8.1 General

Like FD or Finite Elements (FE), the FV method is a mean to solving a Partial Differential Equation (PDE) by transforming it into a discrete algebraic form. In the field of fluid mechanics, the finite volume method is the most popular approach. Other than FD or FE, the finite volume is conservative by construction(we will explain what that means in Section **TODO**). Other than FD it can easily be implemented for unstructured grids. Compared to the FE method, where boundary conditions come naturally from the formulation, this causes some difficulties in FV. It shall also be mentioned that the introduction of stabilization schemes(like streamline upwinding), is much easier in an FV formulation. Overall, for CFD, FV has so far shown to be the best compromise between accuracy, stability and efficiency.

Finite volume methods are typically derived from the so called conservative form of a PDE. It is shown in Section 6.3, that the NSE equation can be brought to the same form. In general, the conservative form can be written as:

$$\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}t} + \nabla \cdot f(\boldsymbol{\xi}) = \mathbf{0} \tag{87}$$

$$\tag{88}$$

where $\boldsymbol{\xi}$ represents a vector of states and f is the so-called flux tensor.
After subdividing the domain into finite volumes, also called cells, on can write for each particular cell i

$$\int_{V_i} \frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}t} dV_i + \int_{V_i} \nabla \cdot f(\boldsymbol{\xi}) dV_i \tag{89}$$

After applying the divergence theorem to the second term this gives:

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{V_i} \boldsymbol{\xi} dV_i + \int_{\partial V_i} f(\boldsymbol{\xi}) \cdot \boldsymbol{n} \, dS_i \tag{90}$$

And after integration the first term to get the volume average

$$V_i \frac{\mathrm{d}\bar{\boldsymbol{\xi}}}{\mathrm{d}t} + \int_{\partial V_i} f(\boldsymbol{\xi}) \cdot \boldsymbol{n} \, dS_i \tag{91}$$

So that finally, the equation can be written as

$$\frac{\mathrm{d}\bar{\boldsymbol{\xi}}}{\mathrm{d}t} + \frac{1}{V_i} \int_{\partial V_i} f(\boldsymbol{\xi}) \cdot \boldsymbol{n} \, dS_i \tag{92}$$

which can easily be interpreted. The cell average of the conserved properties changes according to the total flux through the cells surface. Of course, the conservative value is defined as being constant within one cell, so there will be different values on faces or edges, depending on which side is one is looking at. There are different approaches on how to choose an appropriate value. And the choice made greatly effects the numerical properties. **Is this accurate? If so, provide more details.**

8.2 Finite Volume method for fluid mechanics

It has already been shown in Section REF, that the NSE can be brought into the conservative form(Equation REF). The flux has been shown to consist of a convective and a diffusive part(Equation REF). For this thesis, a Monotonic Upwind scheme for Conservation Laws (MUSCL) type FV framework for unstructured three-dimensional grids, as described in [8] has been used.

The formulation is derived by re-writing equation **??** in variational form by multiplying with a test function:

$$\int_\Omega \frac{\partial \boldsymbol{w}}{\partial t}\phi_i d\Omega + \int_\Omega \nabla \cdot \mathcal{F}(\boldsymbol{w})\phi_i d\Omega + \int_\Omega \nabla \cdot \mathcal{G}(\boldsymbol{w})\phi_i d\Omega = \mathbf{0} \tag{93}$$

where $\phi_i \in C^0(\Omega)$. For this formulation ϕ_i is chosen to be piecewise linear. Particularly, the test functions fulfill

$$\phi_i(\boldsymbol{X}_j) = \delta_{ij} \tag{94}$$

Now, Gauss divergence theorem is applied to the last part, giving

$$\int_\Omega \frac{\partial \boldsymbol{w}}{\partial t}\phi_i \; d\Omega + \int_\Omega \nabla \cdot \mathcal{F}(\boldsymbol{w})\phi_i \; d\Omega + (\int_\Gamma \mathcal{G}(\boldsymbol{w}) \cdot \boldsymbol{n} \; d\Gamma - \int_\Omega \mathcal{G}(\boldsymbol{w}) \cdot \nabla\phi_i \; d\Omega) = \mathbf{0} \tag{95}$$

In contrast to a real FE formulation, the contribution of the viscous flux at the far fields boundary is now neglected, as explained in [6]. How the boundary conditions at the far fields are taken care of will be explained at a proper place. We finally get the mixed FV-FE form by mass lumping the first two terms. Since we are using a vertex-centered approach here, mass lumping is equivalent to using a constant test function of ϕ_i on the dual cell. We therefor get:

$$\int_{\Omega_i} \frac{\partial \boldsymbol{w}}{\partial t} \; d\Omega + \int_{\Omega_i} \nabla \cdot \mathcal{F}(\boldsymbol{w}) \; d\Omega - \int_\Omega \mathcal{G}(\boldsymbol{w}) \cdot \nabla\phi_i \; d\Omega = \mathbf{0} \tag{96}$$

Please note that we have switched from an integral over the whole domain to an integral across the dual cells here. Finally by averaging $\boldsymbol{w}$ over the cell in the first term, we can derive the FV formulation as

$$\frac{\partial \boldsymbol{w}_i}{\partial t} + \frac{1}{\|\Omega_i\|}\int_{\Omega_i} \nabla \cdot \mathcal{F}(\boldsymbol{w}) \; d\Omega_i - \frac{1}{\|\Omega_i\|}\int_\Omega \mathcal{G}(\boldsymbol{w}) \cdot \nabla\phi_i \; d\Omega = \mathbf{0} \tag{97}$$

Where now, $\boldsymbol{w}_i$ denotes the average of $\boldsymbol{w}$ in the dual cell Ω_i, which is by construction the value of $\boldsymbol{w}$ at vertex i. The volume of cell Ω_i is denoted as $\|\Omega_i\|$ here. Finally, gauss divergence theorem is applied to the convective term, resulting in

$$\frac{\partial \boldsymbol{w}_i}{\partial t} + \frac{1}{\|\Omega_i\|}\int_{\partial\Omega_i} \mathcal{F}(\boldsymbol{w}) \cdot dS - \frac{1}{\|\Omega_i\|}\int_\Omega \mathcal{G}(\boldsymbol{w}) \cdot \nabla\phi_i \; d\Omega = \mathbf{0} \tag{98}$$

eq:nsg_basic_fv-form

As can be seen from picture 3, the dual cells themselves can have very random shapes. This makes the integration over the boundary of the second term in Equation (98) more cumbersome than in a primal approach. Also, the volume integral in the FE like expression has to be splitted into regular shapes subdomains, e.g. tetrahedra, such that standard integration rules like gauss rule can be applied.

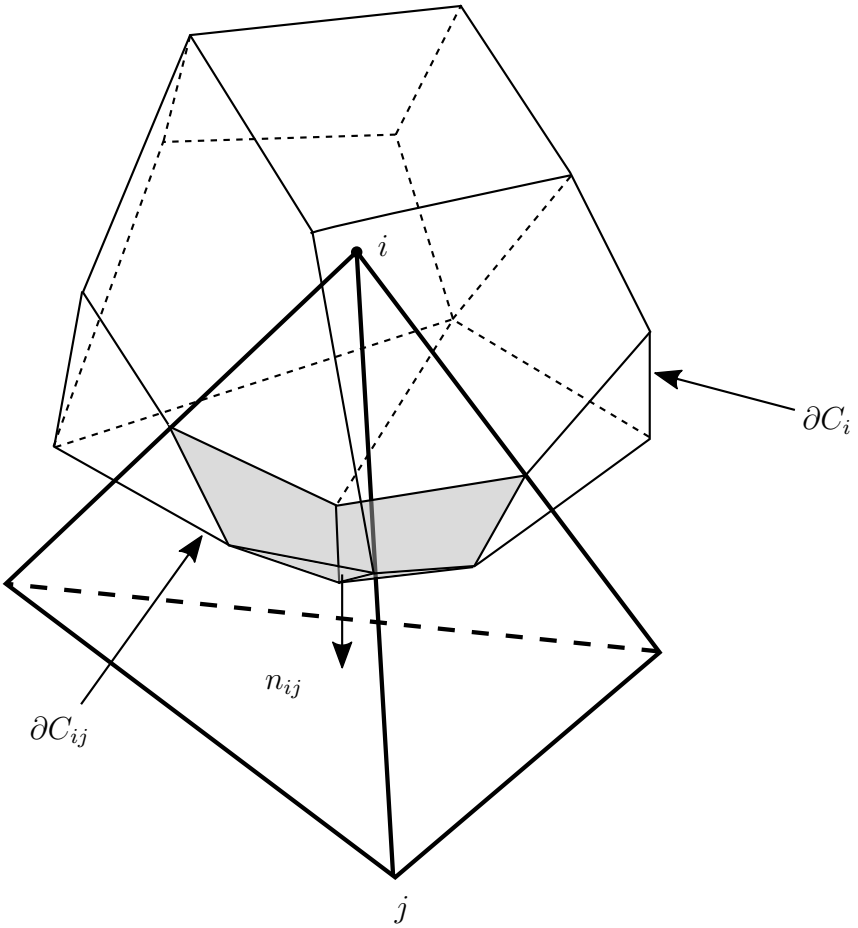


Figure 3: FV semi-discretization of an unstructured mesh. Vertex i is the center of dual cell C_i. The boundary of the dual cell is denoted as ∂C_{ij}

For a closer look into the convective fluxes integral, we decompose the boundary as $\partial\Omega_i = \sum_{i\in\kappa(i)} \partial\Omega_{ij}$, where $\kappa(i)$ denotes the set of vertices connected by an edge to vertex i.
In this thesis, the surface integral in equation REF is than approximated using a Riemann solver and a MUSCL [14] technique. This approximation can be written as

$$\int_{\partial\Omega_i} \mathcal{F}(\boldsymbol{w}) \cdot dS \approx \sum_{j\in\kappa(i)} \boldsymbol{\phi}_{ij}(\boldsymbol{w}_{ij}, \boldsymbol{w}ji, \boldsymbol{n}_{ij}) \tag{99}$$

where $\boldsymbol{\phi}_{ij}$ denotes the chosen numerical flux function along the edge $i-j$ and the two extrapolated fluid stated at the i and the j-side of the the intersection of the

cell boundary $\partial\Omega_{ij}$ and edge $i-j$ are denoted by $\boldsymbol{w}_{ij}$ and $\boldsymbol{w}_{ji}$ respectively. The area-weighted normal of edge $i-j$ is denoted as $\boldsymbol{n}_{ij}$.

As for the volume integral of the diffusive term in Equation (98), it shall be noted that the shape function is still denoted in the primal cell. Since the gradient of the test function is constant, as is the diffusive flux itself, the integral becomes a summation of the primal sub-tetrahedral contributions.

9 Fiver

It was already outlined in Section REF, that using an Eulerian approach int the context of an thermoelastic simulations leads to a so-called embedded formulation, where the interface of the structure mesh no longer coincides with the fluid mesh. This was appealing, since it allowed for the usage of fixed meshes and an Eulerian formulation, on the other hand as explained in Section REF, an error of the order $\mathcal{O}(()\frac{h}{2})$ is introduced.

A possible solution to this problem is the Finite Volume Method with exact two-phase Riemann Integrals (FIVER) method.

9.1 Level-set method

Since the interface in an embedded stimulation typically moves, an appropriate interface tracking approach is required.
For FIVER, the popular level set method was chosen.

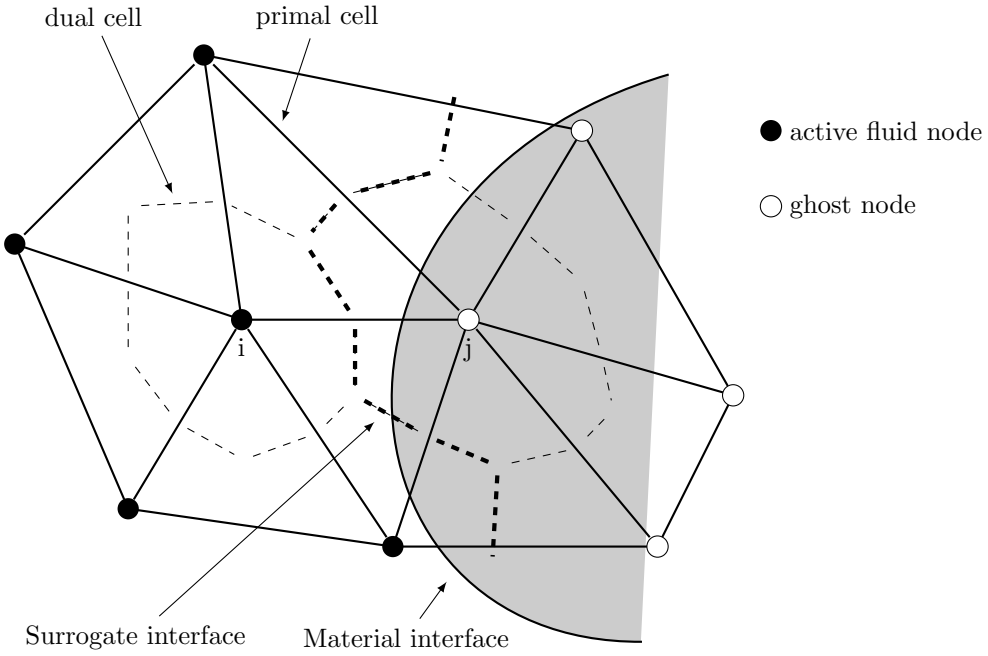


Figure 4: TODO

References

[1] C. Farhat, C. Degand, B. Koobus, and M. Lesoinne. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Computer Methods in Applied Mechanics and Engineering*, 163(1-4):231–245, 1998.

[2] Charbel Farhat, Michel Lesoinne, and Nathan Maman. Mixed explicit/implicit time integration of coupled aeroelastic problems: Three???field formulation, geometric conservation and distributed solution. *International Journal for Numerical Methods in Fluids*, 21(10):807–835, 1995.

[3] J. Frederic Bonnans, J. Charles Gilbert, Claude Lemar??chal, and Claudia A. Sagastiz??bal. Numerical optimization: Theoretical and practical aspects. *Numerical Optimization: Theoretical and Practical Aspects*, pages 1–494, 2006.

[4] Farhat Research Group. Aero-F.

[5] Farhat Research Group. Aero-S.

[6] V. Lakshminarayan, C. Farhat, and A. Main. An embedded boundary framework for compressible turbulent flow and fluid-structure computations on structured and unstructured grids. *International Journal for Numerical Methods in Fluids*, 76(6):366–395, oct 2014.

[7] Michel Lesoinne, Marcus Sarkis, Ulrich Hetmaniuk, and Charbel Farhat. A linearized method for the frequency analysis of three-dimensional uid / structure interaction problems in all ow regimes. 190:1–38, 2001.

[8] Alexander Main. Algorithmic Problems in Social and Geometric Influence a Dissertation Submitted To the Institute for Computational and Mathematical Engineering and the Committee on Graduate Studies of Stanford University in Partial Fulfillment of the Requirements for the. 2014.

[9] K. Maute, M. Nikbay, and C. Farhat. Coupled analytical sensitivity analysis and optimization of three-dimensional nonlinear aeroelastic systems. *AIAA Journal*, 39(11):2051–2061, 2001.

[10] K. Maute, M. Nikbay, and C. Farhat. Sensitivity analysis and design optimization of three-dimensional non-linear aeroelastic systems by the adjoint method. *International Journal for Numerical Methods in Engineering*, 2003.

[11] Serge Piperno and Charbel Farhat. Design of Efficient Partitioned Procedures for the Transient Solution of Aeroelastic Problems. *Revue Européenne des Éléments Finis*, 9(6-7):655–680, jan 2000.

[12] K. Schittkowski, C. Zillober, and R. Zotemantel. Numerical comparison of nonlinear programming algorithms for structural optimization. *Structural Optimization*, 7(1-2):1–19, 1994.

[13] Jaroslaw Sobieszczanski-Sobieski. Sensitivity of complex, internally coupled systems. *AIAA Journal*, 28(1):153–160, 1990.

[14] Bram van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *Journal of Computational Physics*, 32(1):101–136, 1979.