

661 | Mai 2012

SCHRIFTENREIHE SCHIFFBAU

Arthur Stück

**Adjoint Navier–Stokes Methods for
Hydrodynamic Shape Optimisation**

TUHH

Technische Universität Hamburg-Harburg

Adjoint Navier–Stokes Methods for Hydrodynamic Shape Optimisation

Vom Promotionsausschuss der
Technischen Universität Hamburg-Harburg
zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)
genehmigte Dissertation

von
Arthur Stück
aus Schleswig

2011

1. Gutachter: Prof. Dr.-Ing. Thomas Rung
2. Gutachter: Prof. Dr. Rainald Löhner

Tag der mündlichen Prüfung: 16. November 2011

Bericht Nr. 661 der Schriftenreihe Schiffbau der
Technischen Universität Hamburg-Harburg
Schwarzenbergstraße 95c
D-21073 Hamburg
ISBN 978-3-89220-661-3

Abstract

Adjoint Navier–Stokes methods are presented for incompressible flow. The sensitivity derivative of scalar hydrodynamic objective functionals with respect to the shape is obtained at the cost of one flow-field computation, so that the numerical effort is practically independent of the number of shape parameters. The adjoint Navier–Stokes problem was derived on the level of partial differential equations (PDE) first. According to the frozen-turbulence assumption, variations of the turbulence field with respect to the shape control were neglected in the adjoint analysis. Second, consistent discretisation schemes were derived for the individual terms of the adjoint PDE on the basis of the primal, unstructured finite-volume discretisation. A unified, discrete formulation for the adjoint wall boundary condition and the sensitivity equation is presented that supports both low- and high-Reynolds number boundary treatments. The segregated pressure-correction scheme used to solve the primal problem was also pursued in the adjoint code. Reusing huge portions of the flow code led to a compact adjoint module, reduced coding effort and yielded a consistent implementation. Analytical adjoint solutions were tailored to validate the adjoint approach. Moreover, the adjoint-based sensitivity derivative was verified against the direct-differentiation method for both internal and external flow cases. The adjoint method was used for ship-hydrodynamic design optimisation to compute the sensitivity derivative of a wake objective functional with respect to the hull shape. The sensitivity map yields considerable insight into the design problem from the objective point of view. The adjoint-based sensitivity analysis was carried out at model scale to support a manual redesign of the hull and led to an improved wake field. An explicit, filtering-based preconditioning of the sensitivity derivative is first-order equivalent to the concept of “Sobolev-smoothing”. Particularly the sensitivity derivative obtained in conjunction with the low-Reynolds number treatment of boundary walls needed to be filtered before applied to the design surfaces. Guided by the adjoint-based sensitivity derivative, automatic shape optimisation runs were performed for 2D and 3D internal flow problems to reduce the power loss.

Acknowledgements

It has been a pleasure to work at the Institute for Fluid Dynamics and Ship Theory at Hamburg University of Technology under the supervision of Prof. Dr.-Ing. Thomas Rung, who gave me a great deal of latitude for my studies in the field of viscous CFD methods for hydrodynamic shape optimisation. He was continuously interested in my work, spent much of his time on inspiring discussions and enthusiastically came up with many valuable ideas. I thank Prof. Dr. Rainald Löhner for giving me the great opportunity to join his team as a visiting researcher at George Mason University and for being second assessor. Thanks are due to Prof. Dr.-Ing. Otto v. Estorff for chairing the examination board. Prof. Dr.-Ing. Heinrich Söding has probably initialised this study by confronting me with the “adjoint” technique in the first place—without ever using the term. I am very grateful for the valuable discussions we have had. I was lucky to work in the core of a growing group of young researchers developing viscous CFD methods for shape optimisation in research projects with partners from academia and industry. I have greatly enjoyed the company and the help of my colleagues at the institute. In particular, I am very thankful to Jörn Kröger for help in geometry manipulation and mesh generation for the ship-hull optimisation study; and to Jörg Brunswig and Manuel Manzke always being helpful in the range of Fortran and FreSCo⁺. Finally, I thank Anne for having patience with me-and-the-time-consuming-CFD and, most importantly, for her continuous support beyond the CFD.

Contents

Abstract	iii
Acknowledgements	v
Nomenclature	xi
1. Introduction	1
1.1. Background and Motivation	1
1.2. Starting Point and Direction	3
1.3. Present Contributions	4
1.4. Plan of the Thesis	5
2. Calculation of Constrained Derivatives	7
2.1. Finite Differencing	8
2.2. Complex-Step Differentiation	10
2.3. Continuous Differentiation	10
2.4. Discrete Differentiation	14
3. Review of Adjoint-based Shape Optimisation	19
3.1. Continuous and Discrete Differentiation	19
3.2. Primal and Adjoint Algorithms	21
3.3. Gradient Evaluation	23
3.4. Turbulence Treatment	24
3.5. Accuracy and Consistency	25
3.6. Integration into Shape Optimisation	27
3.7. Hydrodynamic and Marine Applications	28
4. Hydrodynamic Optimisation Problem	31
4.1. Governing Field Equations	31
4.2. Boundary Conditions	34
4.3. Hydrodynamic Objectives	35
5. Variation of the Optimisation Problem	39
5.1. Concept of Material Derivative	39
5.2. Variation of the Objective Functional	40
5.3. Variation of the Navier–Stokes Equations	43
5.4. Direct Evaluation of Constrained Derivatives	45

6. Continuous Adjoint Navier–Stokes Concept	49
6.1. Augmented Objective Functional	49
6.2. Adjoint Navier–Stokes Equations	50
6.3. Adjoint Gradient Equation	55
6.4. Complete Gradient Equation	56
6.5. Example: Interior vs. Exterior Evaluation of Forces	57
7. Finite-Volume Discretisation	61
7.1. Scalar Transport Equation	61
7.2. Momentum Equations	68
7.3. Pressure-Correction Scheme	70
7.4. Boundary Conditions	73
7.5. Solution Algorithm	78
7.6. Objective Functionals	80
7.7. Navier–Stokes Variation	80
8. Adjoint Discretisation	87
8.1. Scalar Transport Equation	88
8.2. Momentum Equations	92
8.3. Pressure-Correction Scheme	97
8.4. Boundary Conditions	100
8.5. Gradient Equation	104
8.6. Solution Algorithm	105
9. Verification and Validation Studies	109
9.1. Analytic Solution for Axis-Symmetric Couette Flow	109
9.2. Direct-Differentiation Method	117
10. CAD-free Geometry Concept for Shape Optimisation	137
10.1. Evaluation of Metric Terms	137
10.2. Handling of Constraints	137
10.3. Filtering of Derivatives	138
10.4. Mesh Adaptation	141
11. Applications	143
11.1. Sensitivity Analysis for Manual Wake Optimisation	143
11.2. Optimisation of a 2D T-Junction	153
11.3. Optimisation of a 3D Double-Bent Pipe	162
12. Outlook and Conclusions	171
A. Derivation of Adjoint Navier–Stokes Equations	175
A.1. Variation of the Navier–Stokes Equations	175
A.2. Integration by Parts	175

B. Advection Formulations ADV2/3	177
B.1. Cartesian Coordinates	177
B.2. Physical Cylinder Coordinates	177
B.3. Comparison of Advection Schemes	179
C. Differential Geometry	181
D. Two-Step Adjoint Pressure-Correction Scheme	183

Nomenclature

The subsequent glossary of terms is not exhaustive. Einstein's sum convention applies to small-type Latin subscripts unless declared differently. In symbolic notation, the number of underlines corresponds to the order of a tensor.

Lower-case Greek

α_ϕ	under-relaxation factor ($0 < \alpha_\phi < 1$)
$\beta, \underline{\beta}$	continuous or discrete (shape) control
β_ϕ	factor for UDS-CDS blending ($0 < \beta_\phi < 1$)
γ_ϕ	diffusion coefficient for scalar variable ϕ
δ	variation operator
δ_C	convective variation operator (with respect to the position)
δ_G	geometric variation operator denoting partial variation with respect to metrics (surface area, boundary unit vector)
δ_L	local variation operator (with respect to the flow)
δ_{LC}	material variation operator, consisting of convective and local variations ($\delta_{LC} = \delta_L + \delta_C$)
δn	boundary-normal position shift ($\delta n = n_i \delta x_i$)
$\delta \underline{x}$	position shift
δ_{ij}	Kronecker delta
ε	rate of dissipation of turbulent kinetic energy
λ	linear weighting factor for cell-face interpolation ($0 < \lambda < 1$)
μ	dynamic viscosity
μ_{eff}	effective viscosity ($\mu + \mu_T$)
μ_{\log}	auxiliary definition of effective viscosity in terms of logarithmic law of the wall
μ_T	eddy viscosity
ν	kinematic viscosity
$\underline{\underline{\pi}}$	hydrodynamic stress tensor
ρ	density
σ_ε	turbulent Prantl number for dissipation
σ_k	turbulent Prantl number for turbulent kinetic energy
$\underline{\underline{\tau}}$	viscous portion of hydrodynamic stress tensor
τ^w	wall shear stress magnitude
ϕ'	turbulent fluctuations
$\phi, \underline{\phi}$	continuous or discrete state variable
ω	specific dissipation rate or turbulent frequency

Upper-case Greek

Γ	boundary of flow domain
Γ_D	design surface(s) subject to the shape control (active boundaries)
Γ_O	part of the boundary carrying the objective functional
Ω	interior flow domain
Ω_C	part of the interior domain carrying the control
Ω_O	part of the interior domain carrying the objective functional

Lower-case Roman

\underline{d}	vector connecting CV centres P and N ($\underline{d} = \underline{x}_N - \underline{x}_P$)
\underline{d}^*	unit vector indicating the force component defined as objective functional
\underline{f}	specific body-force vector
g	acceleration of gravity
j	objective functional integrand
j_Γ	boundary-declared objective functional integrand
j_Ω	volume-declared objective functional integrand
k	turbulent kinetic energy
l_T	turbulent length scale
m	number of (discrete) control parameters
n	number of (discrete) constraints
n, t, s	system of boundary coordinates
$\underline{n}, \underline{t}, \underline{s}$	boundary unit vectors
n_O	number of objectives
p	pressure
s, \underline{s}	source term density
t	time
t_T	turbulent time scale
x_i	Cartesian coordinates/position vector
y	closest wall distance

Upper-case Roman

B	boundary condition operator
C_μ	turbulent viscosity constant in the k - ε equation
$D(\cdot)/Dt$	substantial derivative $(\partial/\partial t + \underline{U} \cdot \underline{\nabla})(\cdot)$
F, \underline{F}	cell-face fluxes
\underline{F}^*	hydrodynamic force vector
G	complete gradient ($G = G_L + G_C + G_G$)
G_C	convective gradient contribution associated with a convective variation of the flow
G_G	geometric gradient contribution due to changes of the boundary orientation and surface area
G_L	local gradient contribution associated with a local variation of the flow

\underline{I}	unit matrix [3×3]
J	objective function(al)
J_Γ	boundary-declared objective function(al)
J_Ω	volume-declared objective function(al)
N	Navier–Stokes operator
$\mathcal{O}(\cdot)$	order of
P	rate of production of turbulent kinetic energy
$\underline{\underline{P}}$	matrix projecting onto boundary surface normal ($\underline{\underline{P}} = \underline{\underline{nn}}$)
Q	residual of continuity equation
R_i	residuals of momentum equations
S, \underline{S}	CV source term ($S = \int_{\Delta\Omega} s \, d\Omega$)
$\underline{\underline{S}}$	rate of strain tensor ($2\underline{\underline{S}} = \underline{\nabla U} + (\underline{\nabla U})^T$)
\mathcal{S}	rate of strain ($\mathcal{S} = \sqrt{2\underline{\underline{S}}_{ij}\underline{\underline{S}}_{ij}}$)
$\underline{\underline{T}}$	matrix projecting onto boundary surface ($\underline{\underline{T}} = \underline{\underline{I}} - \underline{\underline{nn}}$)
$\underline{T}_{1,2}$	residual of turbulence equation(s)
\underline{U}	velocity vector
U^τ	shear velocity ($\tau^w = \rho [U^\tau]^2$)
$\underline{U}^{\text{trg}}$	target velocity for inverse design
$\langle \underline{U}'\underline{U}' \rangle$	Reynolds stress tensor

Sub- & Superscripts

i, j, k, l, \dots	small-type Roman indices according to tensor notation
B	boundary face
$B(P)$	set of boundary faces surrounding CV P
C	pointer for convective variations
D	downstream CV
DD	remote downstream CV
F	face index
F'	auxiliary point used for face interpolation
G	pointer for geometric variations
L	pointer for local variations
N	neighbouring CV
$N(P)$	set of immediate internal neighbours surrounding CV P separated by corresponding cell faces
n, s, t	superscripts for system of local boundary coordinates (normal, tangentials)
P	internal CV
PB	boundary face with outward oriented face-normal \underline{n}_{PB}
PN	internal CV face separating CVs P and N with face-normal $\underline{n}_{PN} = -\underline{n}_{NP}$
r, φ, a	superscripts for system of physical cylinder coordinates (radial, circumferential and axial)
U	upstream CV

UU remote upstream CV

Symbols & Specials

$\hat{\phi}$	adjoint variable to ϕ
$\langle \phi \rangle$	mean of ϕ
$\max(a, b)$	the greater of a and b
$\min(a, b)$	the lesser of a and b
$(\cdot) \cdot (\cdot)$	scalar product
$(\cdot) : (\cdot)$	double scalar product
$(\cdot) \times (\cdot)$	vector cross product
$\Delta(\cdot)$	difference operator
$\nabla(\cdot)$	gradient operator
$\nabla \cdot (\cdot)$	divergence operator
$\nabla \times (\cdot)$	curl operator
$\nabla^2(\cdot)$	Laplace operator

Abbreviations

1D, 2D, 3D	one-, two- or three-dimensional
AD	automatic differentiation (also algorithmic differentiation)
ADV0–3	different treatments of adjoint advection term
ALE	arbitrary Lagrangian Eulerian formulation
CAD	computer-aided design
CDS	central differencing scheme
CFD	computational fluid dynamics
CUI	cubic upwind interpolation scheme
CV	control volume
DDS	downwind differencing scheme
FD	finite differencing
FreSCo ⁺	in-house RANS solver: Free Surface Code
FV	finite volume
high-Re	high-Reynolds number turbulence modelling based on the logarithmic law-of-the-wall
LES	large eddy simulation
low-Re	numerical integration of boundary layers according to low-Reynolds number turbulence modelling
LUDS	linear upwind difference scheme
MDO	multidisciplinary design-optimisation
MPI	message passing interface
MUSCL	monotone upstream-centred schemes for conservation laws
PDE	partial differential equation
QUICK	quadratic upwind discretisation
RANS	Reynolds-averaged Navier–Stokes
SQP	sequential quadratic programming

TVD total variation diminishing
UDS upwind differencing scheme

1. Introduction

“Assuming that one has the ability to predict the performance, the question then arises of how to modify the design to improve the performance.” Jameson [61]

1.1. Background and Motivation

Questions related to shape design and optimisation have often challenged the development of new techniques in computational fluid dynamics. Today’s design in aerospace, automotive or maritime industry is based on computer aided engineering (CAE). It is a multidisciplinary design-optimisation (MDO) task, including computer-aided design (CAD), finite-element (FEM) analysis, computational fluid dynamics (CFD) and computer-aided manufacturing (CAM) amongst others. The industrial design process is dominated by tight economic constraints, limiting the costs for man-power, equipment, resources etc. In order to be competitive, the complete process chain needs to be robust and efficient. The performance of its weakest element determines the strength of the overall design chain.

In practical design optimisation, viscous CFD is a notorious bottleneck. Particularly for high Reynolds number flow, the turnaround-times from case set-up to result are high, owing to time-consuming user-input such as grid generation and—of course—computational costs. Typically, hundreds or thousands of CFD runs are carried out during an iterative optimisation when a few (or more) shape parameters are involved. The computational costs of a direct exploration of the entire design space quickly become prohibitive, so that the whole process chain is waiting for the CFD output. In the framework of optimisation, the CFD analysis is merely a link in the process chain, providing the fluid-dynamic information to guide the designer or the design engine. Unless the detailed information available from CFD can be fed into the optimisation process, it is useless.

The best choice of optimisation strategies depends on the design task and state. Global techniques such as evolutionary algorithms [44, 17, 71] are usually preferred in the early stages of the optimisation. These strategies have to be robust in the first instance and take into account objective function *values* alone. Referred to as zero-order methods these approaches are appropriate if the CFD evaluation can be done within seconds on the computer system available. Otherwise gradient-based optimisation, if applicable, can significantly enhance the performance of optimisation runs evaluated by cost-intensive CFD methods. Generally, it requires a more or less smooth distribution of the objective function over the design space and a reasonable initialisation or starting point. The latter is usually available in advanced design stages or when the optimisation starts from a proven design. Standard gradient-based or deterministic techniques search locally around the current design state, wherein the direction of the steepest descent of the objective functional is indicated by the approximation of the sensitivity derivative or gradient.

In industry, the attention turns more and more from pure CFD simulation towards CFD-based optimisation. Cost-efficient gradient-based optimisation is a natural candidate in the context of viscous flow simulations. But besides the restriction to local optimisation, which is inherent to gradient-based optimisation, the industrial application has probably been impeded by the lacking availability of objective functional derivatives constrained by viscous flow. The development of tools for the computation of sensitivity derivatives considerably lags behind the development of flow solvers itself [116], particularly for pressure-based solution schemes widely used in maritime and automotive industry to solve viscous problems of incompressible flow. Since the CFD evaluation is usually associated with high numerical costs, efficient algorithms are required to compute the sensitivity derivatives with respect to the shape. Stability and numerical robustness are premises of equal importance.

The Jacobian matrix $\partial J_i / \partial \beta_j$, being the tensor of first derivatives of the output quantities or objective functions $J_i(\underline{\beta})$ with respect to the input or control variables β_j , can be calculated in several ways. These can be divided into *direct* and *adjoint* methods. Direct methods follow the straight-forward chain of influence, so that for each parameter variation a calculation of the flow variation is necessary which, in turn, is used to compute the variation of the objective functions of interest. When the computational costs are dominated by the calculation of the flow or state, the computational effort scales with the number of input parameters n_β . On the contrary, using the adjoint technique, which was pioneered by Pironneau [118, 119] and Jameson [61], only one dual problem has to be solved per objective J_i . A different classification of approaches to sensitivity analysis regards the order of differentiation and discretisation of the PDE-constrained optimisation problem [38, 104]. Accordingly, the methods are subdivided into *derive-then-discretise* (continuous) and *discretise-then-derive* (discrete) approaches. The issue is discussed in the following sections in the context of segregated pressure-based RANS solvers predominantly used in marine and automotive industry.

In the framework of hydrodynamic shape optimisation, sensitivity calculations can be a key ingredient to cost-efficient design. But like the CFD evaluation, the sensitivity analysis must be embedded in the design process, containing—among others—an optimisation routine or strategy, preconditioning algorithms, tools for an (automatic) geometry update, mesh adaptation or re-meshing and data interfaces. Sensitivity derivatives can support both manual and automatic optimisation strategies. However, industrial optimisation is often carried out manually for different reasons, some of which are:

- No systematic optimisation procedure is available for the problem at hand. Practical applications can be very different from each other and require different optimisation strategies.
- The CFD evaluation of different applications requires different numerical simulation techniques, all of which need to be controlled: pre-processing, validity of models, numerical parameters, etc.
- The mathematical description of design objectives, constraints and parametrisation can be cumbersome and very time-consuming. A general mathematical descrip-

tion of design constraints is often unavailable and design constraints may become “blurred” when they are considered in the objective function through a weighted penalty approach. Thus, an experienced designer can often outpace general-purpose design routines in practical engineering.

- Appropriate data interfaces are required, tailored to the particular optimisation task.

For many ship-hydrodynamic applications, the RANS equations have steady-state solutions in terms of the mean flow variables, which makes them very attractive in the optimisation context. Compared to scale-resolving simulations of turbulent flow, such as large or detached eddy simulations, the steady-state simulation saves computer resources and simplifies the optimisation case considerably. Steady RANS is often considered a reasonable versatility-robustness-efficiency tradeoff and enjoys a wide acceptance in the marine user community. Several general purpose RANS-codes are available and many companies have years of experience and user-competence.

Most workhorse CFD codes used in industry for incompressible RANS simulations are based on the finite-volume method. Implicit pressure-velocity coupling is often enforced by SIMPLE-type pressure-correction algorithms. Parallel 3D calculations on fully unstructured grids are industrial standard. The physics of the optimisation problem may involve additional transport equations that have to be solved along with the turbulent momentum-continuity problem, e.g. accounting for heat transfer or multi-phase flow.

1.2. Starting Point and Direction

The practical trigger for this study was to reduce the excessive numerical costs associated with a systematic sensitivity analysis based on viscous CFD in the face of many design parameters. The challenge is to compute the sensitivity derivative of hydrodynamic design objectives in the adjoint way, so that the computational effort is independent of the number of parameters involved.

The base-line version of the in-house RANS code FreSCo⁺ is based on the finite-volume method and supports 3D unstructured grids of arbitrarily shaped polyhedral cells. The governing flow equations are solved to second-order accuracy in time and space. Pressure-velocity coupling is achieved via a SIMPLE-type projection scheme. The method is implemented in parallel using the MPI protocol. For the sake of code efficiency, maintainability and user-friendliness (minimum of data interfaces required), the adjoint solver was drafted as an add-on module complementing the primal CFD solver. During my work, a so-called hybrid strategy of the continuous and the discrete adjoint method turned out to be promising in conjunction with segregated RANS solvers for the following reasons:

- The background knowledge of the adjoint PDE provides a high level of flexibility regarding adjoint discretisation and solution algorithm where it is required.
- Consistency is ensured by deriving the adjoint discretisation schemes from the corresponding primal schemes. The guidance of the discrete adjoint approach is very

welcome to obtain an appropriate discretisation for the adjoint PDE.

- Coding effort is reduced if the adjoint discretisation and solution schemes can be taken over from the primal implementation. With moderate modifications, the primal flow solver could be adapted to solve the corresponding adjoint problem.

1.3. Present Contributions

The specific contributions to adjoint-based sensitivity analysis for viscous, incompressible flow are summarised as follows:

- Going out from a self-contained derivation of the adjoint Navier–Stokes equations for incompressible flow, the adjoint discretisation schemes are devised term-wise from the primal schemes in a so-called hybrid adjoint approach:
 - The adjoint problem is specific to the considered objective functional, which is consistently taken into account in the adjoint discretisation.
 - The convection schemes need to be adapted to the inverse transport direction of the adjoint problem. Blended upstream/central-differencing schemes as well as limited higher-order schemes (MUSCL-based QUICK, LUDS, CUI schemes) are considered.
 - The linearisation of the primal convection term results in an extra term, referred to as advection term. Different discretisation schemes are derived, presented and discussed.
 - A unified, discrete formulation is presented to consistently treat wall boundaries in the adjoint problem. It supports wall boundary conditions based on both low- and high-Reynolds number (low-/high-Re) models. The adjoint low- and high-Re schemes apply to the adjoint wall boundary condition and the gradient equation.
 - Based on the primal, SIMPLE-type pressure-projection scheme, a corresponding adjoint pressure-correction scheme is derived, implemented and discussed.
- The adjoint method is implemented within a state-of-the-art finite-volume flow solver using unstructured grids of arbitrarily shaped polyhedral cells. The parallel implementation is based on the MPI protocol. A set of standard objective functionals is available. A flexible approach to handle further, user-defined objective functionals is provided through a user-coding interface.
- Analytic adjoint solutions are derived for an axis-symmetric Couette problem allowing to assess how far the numerical adjoint solution is a valid representation of the adjoint PDE and how good different adjoint convection and advection schemes approximate the analytic solution.

- The adjoint-based sensitivity derivative was verified against the direct-differentiation method. The hybrid-adjoint strategy was consistently used to discretise the linearised Navier–Stokes problem. The agreement of the sensitivity derivatives based on the direct-differentiation method and the adjoint method ranges from fair to good, depending on the application of internal or external flow.
- A gradient preconditioning method is presented based on explicit filtering of the sensitivity derivative. The method is first-order equivalent to the well-known “Sobolev-smoothing”. It is easy to implement and intuitive to use. The preconditioned steepest-descent approach guided by the adjoint-based derivative is applied to a selection of 2D and 3D test cases.
- The adjoint method is introduced to wake optimisation. The adjoint right-hand side contributions stemming from the wake objective functional are derived, discretised and discussed. The adjoint-based sensitivity map yields a detailed insight into the optimisation problem from the objective point of view.
- The adjoint-based sensitivity analysis is used to guide an automatic shape optimisation procedure to reduce the power loss in 2D and 3D cases of internal flow.

The adjoint general-purpose solver developed in this work is also applicable in other fields beyond shape design. Its formulation independent of the control makes the adjoint method attractive for other applications, such as topology optimisation, active and passive flow control (turbulence, separation, etc.), goal oriented error analysis, grid adaptation, convergence improvement, etc. The body of the adjoint implementation developed in this work can directly be used for these applications. Minor adaptations are necessary to tailor the adjoint post-processing to the particular type of control or to account for different objective functionals.

1.4. Plan of the Thesis

The basic concepts for the calculation of constrained derivatives are introduced in Chapter 2 followed by a review of adjoint-based shape optimisation presented in Chapter 3. The primal RANS-constrained hydrodynamic optimisation problem is described in Chapter 4. It is derived on a continuous level according to the concept of material derivative and subsequently recast in the adjoint form, cf. Chapters 5 and 6. The unstructured finite-volume method used to solve the primal system is outlined in Chapter 7. In Chapter 8, the attention turns to the adjoint numerics. The adjoint finite-volume discretisation is consistently devised from the primal schemes, and the SIMPLE-based pressure-correction scheme is adapted to solve the adjoint problem. Chapter 9 is concerned with the verification of the adjoint algorithm comparing the numerical results against analytic test cases and the direct-differentiation method. A collection of techniques required for automatic CAD-free shape optimisation is described in Chapter 10, including a filter-based preconditioning technique for sensitivity derivatives, the steepest-descent method, the

evaluation of surface metrics, constraint handling and a mesh-deformation approach. An automatic optimisation procedure is applied to simple optimisation test cases and a 3D duct in Chapter 11. The adjoint method has successfully been introduced to wake design as demonstrated for a generic container vessel. The thesis closes with an outlook and conclusions drawn in Chapter 12.

Parts of the derivation are confined to the appendix to keep the presentation concise. Throughout the analysis, Einstein's sum convention applies to small-type Latin subscripts, unless declared differently. When symbolic notation is used, the number of underlines corresponds to the order of a tensor. A glossary of terms and abbreviations is provided on pages xi ff.

2. Calculation of Constrained Derivatives

“A definition of sensitivity analysis: The study of how the uncertainty [variation] in the output of a mathematical model [objective functional] can be apportioned to different sources of uncertainty [variation] in the model input [shape parametrisation].”
Saltelli et al. [135]

Local optimisation methods are based on the derivative of the objective function(al) with respect to the control function or parametrisation. “Local” suggests that the derivative is obtained at a certain point in the design space. The expression “sensitivity analysis” is used for the calculation of constrained sensitivity derivatives. The existing concepts for the calculation of sensitivity derivatives can be classified as *direct methods* following the Jacobian method, or *adjoint methods* according to the method of Lagrange. A description of a generic, PDE-constrained optimisation problem is presented in the following.

The search for a shape β optimal from a hydrodynamic point of view can be considered as a PDE-constrained control or optimisation problem [118, 119, 61]. The costs are quantified through the integral, real-valued *objective functional*, which usually depends on the *state* ϕ and the *control* β . From a mathematical point of view the subdivision into state function and control function is arbitrary, from an engineering point of view it is usually obvious: In the context of hydrodynamic optimisation, the fluid flow is denoted as state and governed by the *equations of state* in the fluid domain Ω . In shape optimisation, the control (e.g. a shape parametrisation) is applied to a part of the domain boundary, $\Gamma_D \subset \Gamma$, called *design surface*. Alternatively, volume-based control functions can be applied within the *control domain* $\Omega_C \subset \Omega$, e.g. distributions of porosity for topology optimisation [6, 35, 47, 111, 34, 110] or body forces and mass sources for active or passive flow control [13, 10, 59, 11, 55, 8, 159, 9, 48, 154].

The objective functional is composed of boundary and volume contributions:

$$\text{Minimise } J(\phi) = \int_{\Omega_O} j_{\Omega}(\phi) \, d\Omega + \int_{\Gamma_O} j_{\Gamma} [C(\phi, \beta)] \, d\Gamma. \quad (2.1)$$

Fluid force components acting on the *objective surface* $\Gamma_O \subset \Gamma$ are typical representatives for boundary-based objectives. Volume-based criteria declared in the *objective volume* $\Omega_O \subset \Omega$ are, for instance, the energy dissipation per fluid volume or homogeneity objectives formulated in terms of the fluid velocity. The boundary-based operator C used in Eqn. (2.1) may be non-linear, e.g. a quadratic norm for the pressure deviation, and may contain derivatives of the state variables, e.g. in case of force objectives. In many cases the definition of the volume contribution j_{Ω} is non-linear. The objective functional (2.1) is subject to the non-linear equations of state, e.g. the Navier–Stokes equations governing the flow ϕ

$$N(\phi, \beta) = 0 \quad \text{in } \Omega \quad (2.2)$$

with the boundary conditions

$$B(\underline{\phi}, \underline{\beta}) = 0 \quad \text{on } \Gamma. \quad (2.3)$$

Also the boundary condition operator B can be non-linear, e.g. a convective velocity boundary condition. It is differential for Neumann or Robin boundary conditions.

In order to be solved numerically, this problem is turned into the discrete form:

$$\text{Minimise} \quad J(\underline{\phi}, \underline{\beta}) \quad (2.4)$$

$$\text{subject to} \quad N_j(\underline{\phi}, \underline{\beta}) = 0, \quad j = 1, \dots, n. \quad (2.5)$$

Like the constraints, the state variables and the control are now in vector form evaluated at discrete points, elements or volumes in the solution domain

$$\begin{aligned} \underline{\phi} &= (\phi_1, \dots, \phi_n) \\ \underline{\beta} &= (\beta_1, \dots, \beta_m). \end{aligned}$$

The solution is unique if the number of independent constraints matches the dimension of the state vector.

The objective functional is evaluated from the primal, non-linear system in a straightforward sequence:

$$\text{Set } \underline{\beta}. \quad \rightarrow \quad \text{Solve } N_j(\underline{\phi}, \underline{\beta}) = 0 \text{ for } \underline{\phi}. \quad \rightarrow \quad \text{Evaluate } J(\underline{\phi}, \underline{\beta}). \quad (2.6)$$

The sequence (2.6) can be traversed for the calculation of the constrained variation of the objective functional (receiver/output) J with respect to the control (input/sender) $\underline{\beta}$. Following the direction of (2.6) is referred to as *forward* or direct mode of sensitivity analysis. The dependence can be written as $\delta J[\delta \underline{\phi}(\delta \underline{\beta}), \delta \underline{\beta}] = \delta J(\delta \underline{\beta})$. Alternatively, this chain can be traversed in *reverse* which is done by adjoint methods.

Four popular techniques for the calculation of constrained sensitivity derivatives are presented in the following—being the methods of finite differencing (Section 2.1), complex-step differentiation (Section 2.2), continuous (Section 2.3) and discrete differentiation (Section 2.4).

2.1. Finite Differencing

Finite differencing (FD) is the simplest approach to calculate constrained objective function derivatives. It is a direct method, traversing the chain (2.6) in forward direction. A first-order approximation of the sensitivity derivative is the one-sided divided difference

$$\frac{dJ}{d\beta_\alpha} = \frac{J(\underline{\beta} \pm \varepsilon \underline{e}_\alpha) - J(\underline{\beta})}{\pm \varepsilon} + \mathcal{O}(\varepsilon), \quad (2.7)$$

where ε is a small perturbation applied to the α -component of the discrete control vector $\underline{\beta}$. A second-order FD approximation is obtained from

$$\frac{dJ}{d\beta_\alpha} = \frac{J(\underline{\beta} + \varepsilon \underline{e}_\alpha) - J(\underline{\beta} - \varepsilon \underline{e}_\alpha)}{2\varepsilon} + \mathcal{O}(\varepsilon^2). \quad (2.8)$$

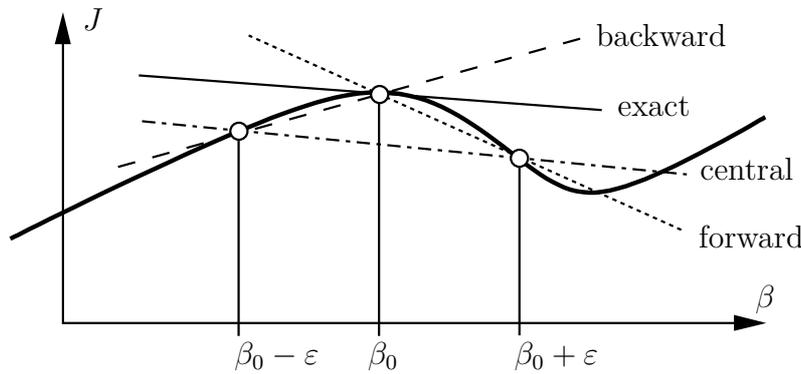


Figure 2.1.: Approximation of the objective functional derivative $[\partial J/\partial\beta]_{\beta_0}$ by divided differences

The finite-difference approximations (2.7) and (2.8) to the derivative are plotted in Figure 2.1. For the control vector $\underline{\beta}$ [m], a first-order approximation requires $m + 1$ objective function evaluations—one for the reference case plus m perturbed solutions. The second-order approximation (2.8) comes at the price of $2m$ objective evaluations, i.e. the computational effort scales with the number of parameters involved. The FD method is straight-forward and does not require an extra sensitivity code. However, an appropriate perturbation size ε has to be found:

- Too small a perturbation may lead to erroneous derivatives suffering from cancellation errors.
- In the case of too large perturbation steps the truncation errors in (2.7) and (2.8) falsify the finite-difference approximations to the derivative.

The state solutions need to be well converged to obtain a good approximation of the gradient via divided differences. Some computation time may be saved by using the reference solution to initialise the calculation of the perturbed flow field.

Particularly in automatic optimisation strategies, the success and the efficiency of the optimisation may depend on the quality of the gradient approximation. The (combination of the) numerical parameters, such as perturbation step, convergence tolerance, data precision, etc., must be controlled carefully. Therefore the use of FD techniques—though they are simple from an implementational point of view and can be executed in a trivially parallel mode—is not straight-forward in practice. An enormous advantage of the FD method is that it does not require access to the source codes of the computer programs involved in the optimisation chain, e.g. CAD tools, mesh generator, flow solver, post-processor, etc. The whole process chain from input to output can be considered as a “black box” which is differentiated in one go (Mohammadi [94]). The FD technique is a popular means for the validation of alternative concepts of sensitivity analysis [e.g. 99, 42, 16, 76, 110, 155, 9, 171]. Alternatively, intermediate derivatives within the chain of

influence may be determined by finite differencing of proprietary CAD or grid generating software, which are subsequently multiplied to adjoint-based derivatives according to the chain-rule of differentiation to close the chain.

2.2. Complex-Step Differentiation

The variables depending on the input $\underline{\beta}$ are redefined in the computer program from real-valued to complex. A perturbation ε is applied to the imaginary part so that a second-order approximation of the desired sensitivity derivative can be calculated from the imaginary part of the function

$$\frac{dJ}{d\beta_\alpha} = \frac{\text{Im} [J(\underline{\beta} + i\varepsilon \underline{e}_\alpha)]}{\varepsilon} + \mathcal{O}(\varepsilon^2). \quad (2.9)$$

The corresponding real part contains the objective function value to second-order accuracy. Very small ε -values can be used since cancellation errors are avoided. The gradient approximation (2.9) still suffers from iteration errors. However, Newman III et al. [105] state that fully converged solutions are not necessary to obtain sufficiently accurate approximations of the derivative in conjunction with RANS-based optimisation. Flow and derivative are calculated together so that m evaluations are required for a complete evaluation of the gradient vector. Note that the source code of all computer programs involved between input $\underline{\beta}$ and output J must be available. Memory and storage requirements almost double as complex variables take the equivalent of two real-valued variables. Cusdin and Müller [16] report that a complex-step gradient computation takes over eight times the CPU time of a primal simulation for 2D compressible Euler and Navier–Stokes simulations. The implementation effort increases when operations are not declared or have a special meaning in the complex context. Martins et al. [92] suggest an automated, script-based precompilation of the input code. Overloading of real functions and operators that are incompatible with complex arguments by their complex counterpart is a convenient means to reduce the implementation effort. Müller and Cusdin [97] use the complex-step differentiation in combination with automatic differentiation to obtain the right-hand side terms for an adjoint solver. The absence of cancellation errors makes the method also attractive for verification studies, e.g. Nadarajah [101].

Alternatively, the PDE-constrained optimisation problem can be derived according to control theory either on the continuous or the discrete level as described in the following.

2.3. Continuous Differentiation

Linear changes in the objective functional, δJ , caused by a shape perturbation are attributed to

- a perturbation of the control, $\delta\beta$, and
- compatible flow changes $\delta\phi$ satisfying the variation of the PDE-constraints $\delta N = 0$.

A comma notation is used in the continuous analysis to denote the partial derivative, either with respect to the flow ϕ or with respect to the control β . The variation of the objective functional can either be obtained by the *Jacobian* (direct) or the *Lagrangian* (adjoint) calculus.

In order to prepare the following Navier–Stokes based analysis, the descriptions by Giles and Pierce [39] and Hartmann and Houston [51] are extended to distinguish parts of the boundary Γ or the internal domain Ω that are either subject to the control (Γ_D and Ω_C), carry the objective functional (Γ_O and Ω_O), or are combinations thereof ($\Gamma_O \cap \Gamma_D$ and $\Omega_O \cap \Omega_C$). The generic formulation derived for the gradient expression (G_Γ on Γ_D and G_Ω in Ω_C) is reused in Chapter 6. The variation of the objective functional can either be obtained by the *Jacobian* (direct) or the *Lagrangian* (adjoint) calculus, both of which are systematically compared. The issue is presented in a continuous formulation first, followed by a discussion of the corresponding discrete problem. The brief description provided here in a consistent notation serves as a prototype for the continuous adjoint Navier–Stokes analysis presented in Chapter 6, which is reconsidered on the discrete level in Chapter 8 to derive the adjoint discretisation.

Jacobian Method

In the direct or Jacobian method, the first variation of the objective is evaluated in a straight-forward sequence: First, the variation of the state $\delta\phi$ with respect to a change of the control $\delta\beta$ is calculated from the *variation of the equations of state* ($\delta N = 0$)

$$\begin{aligned} N_{,\phi} \delta\phi &= -N_{,\beta} \delta\beta & \text{in } \Omega_C \\ N_{,\phi} \delta\phi &= 0 & \text{in } \Omega \setminus \Omega_C \end{aligned} \quad (2.10)$$

subject to the *variation of the boundary conditions*

$$\begin{aligned} B_{,\phi} \delta\phi &= -B_{,\beta} \delta\beta & \text{on } \Gamma_D \\ B_{,\phi} \delta\phi &= 0 & \text{on } \Gamma \setminus \Gamma_D. \end{aligned} \quad (2.11)$$

For instance, for a generic Robin-boundary condition with the coefficients c_0 , c_1 and c_2

$$B(\phi) = c_0 + c_1\phi + c_2 \frac{\partial\phi}{\partial n} = 0 \quad \text{on } \Gamma_{\text{Robin}} \quad (2.12)$$

the full variation reads

$$B_{,\phi} \delta\phi = c_1 \delta\phi + c_2 \frac{\partial}{\partial n} (\delta\phi) \quad \text{on } \Gamma_{\text{Robin}}. \quad (2.13)$$

In the second step the PDE-constrained, first variation of the objective functional, namely $\delta J(\delta\phi(\delta\beta), \delta\beta)$, has to be evaluated:

$$\delta J = \delta_\beta J + \delta_\phi J. \quad (2.14)$$

The calculation of the leading right-hand side term of Eqn. (2.14), representing the partial variation with respect to the shape, is straight-forward and requires metric operations alone

$$\delta_\beta J = \delta_\beta \int_{\Gamma_o} j_\Gamma d\Gamma + \delta_\beta \int_{\Omega_o} j_\Omega d\Omega. \quad (2.15)$$

With the solution $\delta\phi$ obtained from Eqn. (2.10) the second right-hand side term of Eqn. (2.14), being the constrained variation with respect to the state, is determined

$$\delta_\phi J = \int_{\Omega_o} j_{\Omega,\phi} \delta\phi d\Omega + \int_{\Gamma_o} j_{\Gamma,C} C_{,\phi} \delta\phi d\Gamma, \quad (2.16)$$

using the chain-rule of differentiation

$$\delta_\phi j_\Gamma [C(\phi)] = j_{\Gamma,C} C_{,\phi} \delta\phi. \quad (2.17)$$

Since the Jacobian method works in direct mode—i.e. the control function β is perturbed, the corresponding change in the state $\delta\phi(\delta\beta)$ is evaluated from the linearised equations of state (2.10), and eventually the change in the objective functional $\delta J(\delta\phi(\delta\beta))$ is evaluated from Eqn. (2.14)—the effort for the sensitivity analysis scales with the number of perturbations. Alternatively, the method of Lagrange can be pursued, which formally turns the constrained optimisation problem into an unconstrained problem, so that the derivative of the objective functional with respect to the control function (infinite number of degrees of freedom) can be obtained from one calculation.

Method of Lagrange

The method of Lagrange starts out from the so-called Lagrange functional L . It is composed of the objective functional augmented by the constraints N , weighted by the Lagrange multipliers or *adjoint variables* $\hat{\phi}$:

$$\begin{aligned} L &= J_\Gamma + J_\Omega + \int_{\Omega} \hat{\phi} N(\phi, \beta) d\Omega \\ &= \int_{\Gamma_o} j_\Gamma d\Gamma + \int_{\Omega_o} j_\Omega d\Omega + \int_{\Omega} \hat{\phi} N(\phi, \beta) d\Omega. \end{aligned} \quad (2.18)$$

A complete linearisation with respect to β and ϕ yields

$$\begin{aligned} \delta L &= \delta_\beta J + \int_{\Gamma_o} j_{\Gamma,C} C_{,\phi} \delta\phi d\Gamma + \int_{\Omega_o} j_{\Omega,\phi} \delta\phi d\Omega \\ &\quad + \int_{\Omega} \hat{\phi} (N_{,\phi} \delta\phi + N_{,\beta} \delta\beta) d\Omega. \end{aligned} \quad (2.19)$$

The adjoint operators, $\hat{N}_{,\phi}$, $\hat{B}_{,\phi}$ and $\hat{C}_{,\phi}$ are derived via integration by parts of Eqn. (2.19):

$$\delta L = \delta_\beta J + \int_{\Omega_O} \delta\phi j_{\Omega,\phi} d\Omega + \int_{\Gamma_O} C_{,\phi} \delta\phi j_{\Gamma,C} d\Gamma \quad (2.20a)$$

$$+ \int_{\Omega} \delta\phi \left(\hat{N}_{,\phi} \hat{\phi} \right) d\Omega + \int_{\Gamma} (D_{,\phi} \delta\phi) \left(\hat{B}_{,\phi} \hat{\phi} \right) d\Gamma + \int_{\Gamma} \left(\hat{D}_{,\phi} \hat{\phi} \right) (B_{,\phi} \delta\phi) d\Gamma \quad (2.20b)$$

$$+ \int_{\Omega_C} \hat{\phi} (N_{,\beta} \delta\beta) d\Omega. \quad (2.20c)$$

The operator $D_{,\phi}$ and its adjoint complement $\hat{D}_{,\phi}$ arise from integration by parts. With (2.11), the last term of (2.20b) can be expressed in terms of the control β . Provided that the declaration of the objective functional satisfies

$$C_{,\phi} + D_{,\phi} = 0 \quad \text{on } \Gamma,^1 \quad (2.21)$$

Eqn. (2.20) can be reorganised to read

$$\delta L = \delta_\beta J + \int_{\Omega} \delta\phi \left(\hat{N}_{,\phi} \hat{\phi} \right) d\Omega + \int_{\Omega_O} \delta\phi j_{\Omega,\phi} d\Omega \quad (2.22a)$$

$$- \int_{\Gamma} (C_{,\phi} \delta\phi) \left(\hat{B}_{,\phi} \hat{\phi} \right) d\Gamma + \int_{\Gamma_O} (C_{,\phi} \delta\phi) j_{\Gamma,C} d\Gamma \quad (2.22b)$$

$$+ \int_{\Omega_C} \hat{\phi} (N_{,\beta} \delta\beta) d\Omega + \int_{\Gamma_D} \hat{C}_{,\phi} \hat{\phi} (B_{,\beta} \delta\beta) d\Gamma. \quad (2.22c)$$

The right-hand side terms of line (2.22a) depending on $\delta\phi$ are eliminated by satisfying the *adjoint equations*

$$\begin{aligned} \hat{N}_{,\phi} \hat{\phi} &= -j_{\Omega,\phi} & \text{in } \Omega_O \\ \hat{N}_{,\phi} \hat{\phi} &= 0 & \text{in } \Omega \setminus \Omega_O, \end{aligned} \quad (2.23)$$

and the terms of line (2.22b) are cancelled out by the *adjoint boundary conditions*

$$\begin{aligned} \hat{B}_{,\phi} \hat{\phi} &= j_{\Gamma,C} & \text{on } \Gamma_O \\ \hat{B}_{,\phi} \hat{\phi} &= 0 & \text{on } \Gamma \setminus \Gamma_O. \end{aligned} \quad (2.24)$$

Having solved the adjoint PDE above, the constrained first variation of the objective functional with respect to the state (2.16) can be calculated from the remainder (2.22c), which is independent of $\delta\phi$, viz.

$$\begin{aligned} \delta_\phi J &= \int_{\Omega_C} \hat{\phi} (N_{,\beta} \delta\beta) d\Omega + \int_{\Gamma_D} \left(\hat{C}_{,\phi} \hat{\phi} \right) (B_{,\beta} \delta\beta) d\Gamma \\ &= \int_{\Omega_C} G_\Omega \delta\beta d\Omega + \int_{\Gamma_D} G_\Gamma \delta\beta d\Gamma \end{aligned} \quad (2.25)$$

¹Strictly, a proportionality, $C_{,\phi} + \alpha D_{,\phi} = 0$, is sufficient. The constant multiplier α is omitted for the sake of brevity.

with

$$G_\Omega = \hat{\phi} N_{,\beta} \quad \text{in} \quad \Omega_C \quad \text{and} \quad G_\Gamma = \hat{C}_{,\phi} \hat{\phi} B_{,\beta} \quad \text{on} \quad \Gamma_D. \quad (2.26)$$

Note that the gradient expressions (2.26) are confined to the partial variation of J with respect to ϕ . The evaluation of the partial variation of J with respect to β involves metric operations alone which are not specific to either the Jacobi method or the method of Lagrange. It is omitted here for the sake of brevity.

Continuous Adjoint Identity

The first variations obtained from the Jacobian method (2.16) and the method of Lagrange (2.25) should agree (Giles and Pierce [39]):

$$\delta_\phi J = \int_{\Omega_O} \delta\phi j_{\Omega,\phi} d\Omega + \int_{\Gamma_O} (C_{,\phi} \delta\phi) j_{\Gamma,C} d\Gamma \quad (2.27a)$$

$$= \int_{\Omega_C} \hat{\phi} (N_{,\beta} \delta\beta) d\Omega + \int_{\Gamma_D} (\hat{C}_{,\phi} \hat{\phi}) (B_{,\beta} \delta\beta) d\Gamma. \quad (2.27b)$$

The adjoint field equations (2.23) and boundary conditions (2.24) are applied to (2.27a). The linearised equations of state (2.10) and its boundary conditions (2.11) are substituted into (2.27b). The resulting expressions define the *continuous formulation of the general adjoint identity*

$$\begin{aligned} & \int_{\Omega_O} \delta\phi (\hat{N}_{,\phi} \hat{\phi}) d\Omega + \int_{\Gamma_O} (C_{,\phi} \delta\phi) (\hat{B}_{,\phi} \hat{\phi}) d\Gamma \\ &= \int_{\Omega_C} \hat{\phi} (N_{,\phi} \delta\phi) d\Omega + \int_{\Gamma_D} (\hat{C}_{,\phi} \hat{\phi}) (B_{,\phi} \delta\phi) d\Gamma. \end{aligned} \quad (2.28)$$

Given that Eqn. (2.28) is satisfied, the corresponding first variations of the objective functionals calculated in the Jacobian and the Lagrangian approach are equivalent on the continuous level.

2.4. Discrete Differentiation

Alternatively, the sensitivity analysis can be derived on the basis of the discretisation, Eqns. (2.4) and (2.5), of the underlying PDE-constrained optimisation problem. The linear variation of the discrete objective is obtained from a truncated Taylor series expansion about a feasible point $(\underline{\phi}, \underline{\beta})$

$$\begin{aligned} \delta J &= \sum_{j=1}^n \delta\phi_j \frac{\partial J}{\partial \phi_j} + \sum_{j=1}^m \delta\beta_j \frac{\partial J}{\partial \beta_j} \\ &= \delta_\phi J + \delta_\beta J. \end{aligned} \quad (2.29)$$

Accordingly, the linear variation of the discrete constraints reads

$$\begin{aligned}
\delta N_i = 0 &= \sum_{j=1}^n \delta \phi_j \frac{\partial N_i}{\partial \phi_j} + \sum_{j=1}^m \delta \beta_j \frac{\partial N_i}{\partial \beta_j} \\
&= \sum_{j=1}^n A_{ij} \delta \phi_j + \sum_{j=1}^m C_{ij} \delta \beta_j \\
&= \sum_{j=1}^n A_{ij} \delta \phi_j - \delta s_i,
\end{aligned} \tag{2.30}$$

or, when rearranged

$$\sum_{j=1}^n A_{ij} \delta \phi_j = \delta s_i, \quad i = 1, \dots, n. \tag{2.31}$$

The matrices $\underline{\underline{A}}$, $\underline{\underline{C}}$ represent the *Jacobian* and the *control matrix*, respectively.

Jacobian Method

Assuming that the Jacobian matrix is non-singular, Eqn. (2.30) can be multiplied by $\underline{\underline{A}}^{-1}$

$$\delta \phi_i = - \sum_{j=1}^n \sum_{k=1}^m A_{ij}^{-1} C_{jk} \delta \beta_k, \quad i = 1, \dots, n, \tag{2.32}$$

in order to express the first variation of the objective function (2.29) in terms of $\delta \underline{\underline{\beta}}$:

$$\delta J = - \sum_{j=1}^n \sum_{l=1}^n \sum_{k=1}^m A_{jl}^{-1} C_{lk} \delta \beta_k \frac{\partial J}{\partial \phi_j} + \sum_{k=1}^m \delta \beta_k \frac{\partial J}{\partial \beta_k}. \tag{2.33}$$

The corresponding derivative form reads

$$\frac{\delta J}{\delta \beta_k} = - \sum_{j=1}^n \sum_{l=1}^n A_{jl}^{-1} C_{lk} \frac{\partial J}{\partial \phi_j} + \frac{\partial J}{\partial \beta_k}, \quad k = 1, \dots, m. \tag{2.34}$$

Expression (2.34) is the discrete form of the constrained derivative of J with respect to $\underline{\underline{\beta}}$. It represents the rate of change of J that is induced by explicit perturbations individually applied to β_i and the corresponding flow variations $\delta \underline{\underline{\phi}}(\delta \underline{\underline{\beta}})$ compatible with the hydrodynamic constraints $\underline{\underline{N}} = \underline{\underline{0}}$. Mind that the evaluation of the constrained derivative either requires the inversion of matrix $\underline{\underline{A}}$ according to Eqn. (2.34), or the direct calculation of $\delta \underline{\underline{\phi}}(\delta \underline{\underline{\beta}})$ from Eqn. (2.30), which is subsequently applied to (2.29). In either case, the numerical effort scales with the number of control parameters m .

Method of Lagrange

In the discrete adjoint approach the Lagrangian is constructed as

$$L = J(\underline{\beta}, \underline{\phi}) + \sum_{j=1}^n \hat{\phi}_j N_j, \quad (2.35)$$

with the discrete adjoint variables $\hat{\phi}$. By construction, the system of equations (2.5) corresponds to

$$\frac{\partial L}{\partial \hat{\phi}_j} = 0, \quad j = 1, \dots, n. \quad (2.36)$$

When the constraints are satisfied, i.e. $N_j = 0$, the Lagrangian matches the objective function value. The linear objective function variation

$$\begin{aligned} \delta L &= \sum_{j=1}^n \delta \phi_j \frac{\partial L}{\partial \phi_j} + \sum_{j=1}^m \delta \beta_j \frac{\partial L}{\partial \beta_j} \\ &= \sum_{j=1}^n \delta \phi_j \frac{\partial J}{\partial \phi_j} + \sum_{j=1}^m \delta \beta_j \frac{\partial J}{\partial \beta_j} + \sum_{k=1}^n \hat{\phi}_k \left(\sum_{j=1}^n \delta \phi_j \frac{\partial N_k}{\partial \phi_j} + \sum_{j=1}^m \delta \beta_j \frac{\partial N_k}{\partial \beta_j} \right) \end{aligned} \quad (2.37)$$

can be reorganised as follows:

$$\delta L = \sum_{j=1}^n \delta \phi_j \left(\frac{\partial J}{\partial \phi_j} + \sum_{k=1}^n \hat{\phi}_k \frac{\partial N_k}{\partial \phi_j} \right) + \sum_{j=1}^m \delta \beta_j \left(\frac{\partial J}{\partial \beta_j} + \sum_{k=1}^n \hat{\phi}_k \frac{\partial N_k}{\partial \beta_j} \right). \quad (2.38)$$

The first right-hand side expression can be eliminated for arbitrary $\delta \underline{\phi}$ by solving

$$\begin{aligned} \sum_{j=1}^n \frac{\partial N_j}{\partial \phi_i} \hat{\phi}_j &= -\frac{\partial J}{\partial \phi_i}, \quad \text{or} \\ \sum_{j=1}^n A_{ij}^T \hat{\phi}_j &= \hat{s}_i, \quad i = 1, \dots, n. \end{aligned} \quad (2.39)$$

The equation system (2.39) contains the transpose of the Jacobian matrix obtained by linearisation of the constraints with respect to the state variables, Eqn. (2.30). It is interesting to note, that the boundary terms associated with the adjoint boundary control problem appear as source terms in the discrete adjoint system (e.g. Nadarajah and Jameison [102], Nadarajah [101]). Having solved Eqn. (2.39) for the adjoint variables $\hat{\phi}$, it is possible to evaluate the constrained first variation of the objective function from the remainder of Eqn. (2.38), viz.

$$\delta J = \sum_{i=1}^m \sum_{j=1}^n \delta \beta_i \hat{\phi}_j \frac{\partial N_j}{\partial \beta_i}, \quad (2.40)$$

or in derivative form

$$\frac{\delta J}{\delta \beta_i} = \sum_{j=1}^n \hat{\phi}_j \frac{\partial N_j}{\partial \beta_i}, \quad i = 1, \dots, m. \quad (2.41)$$

The evaluation of the inner product (2.41) is inexpensive from a computational point of view. In the Lagrange method, the main numerical effort is spent solving the adjoint problem (2.39) for the adjoint multipliers $\hat{\phi}$. Each objective function is associated with a corresponding adjoint problem, so that the numerical effort scales with the number of objectives n_O . The computational costs are practically independent of the number of control parameters since the adjoint Eqn. (2.39) does not depend on the variation of the state $\delta \underline{\phi}$.

Discrete Adjoint Identity

Equivalence of the sensitivity derivatives obtained via the Jacobian and the Lagrangian calculus can also be postulated on the discrete level. A valid, consistent approximation to the adjoint PDE (2.23) and its boundary conditions (2.24) should also satisfy the discrete adjoint identity (Giles and Pierce [39]):

$$\delta_\phi J = \sum_{i=1}^n \delta \phi_i \frac{\partial J}{\partial \phi_i} = \sum_{i=1}^n \delta \phi_i \hat{s}_i \quad (2.42a)$$

$$= \sum_{i=1}^n \sum_{j=1}^n \delta \phi_i A_{ij}^T \hat{\phi}_j \quad (2.42b)$$

$$= \sum_{i=1}^n \sum_{j=1}^n \hat{\phi}_i A_{ij} \delta \phi_j \quad (2.42c)$$

$$= \sum_{i=1}^n \hat{\phi}_i \delta s_i. \quad (2.42d)$$

Confined to real numbers, the transposed matrix $\underline{\underline{A}}^T$ is equivalent to the adjoint $\underline{\underline{A}}$.

Automatic Differentiation

Algorithmic or automatic differentiation (AD) is a special case of discrete differentiation. The differentiation is carried out on the code level by source-to-source precompilation. The (nonlinear) input code is considered as a process of numerical operations

$$Z^{(n)}(X) = f^{(1)} \circ \dots \circ f^{(n)}, \quad (2.43)$$

connecting the input parameter(s) X with the output of the computer code $Z^{(n)}$, e.g. Giering and Kaminski [37]. It is derived according to the chain rule of differentiation

$$\frac{dZ^{(n)}}{dX} = \frac{\partial f^{(n)}}{\partial Z^{(n-1)}} \cdot \dots \cdot \frac{\partial f^{(1)}}{\partial Z^{(0)}}, \quad (2.44)$$

wherein

$$Z^{(i)}(X) = f^{(1)} \circ \dots \circ f^{(i)} \quad \text{and} \quad Z^{(0)} = X. \quad (2.45)$$

The chain (2.44) can either be traversed in

- forward mode starting from $f^{(1)}$, or in
- reverse mode from $f^{(n)}$.

These are the AD-equivalents to the direct or adjoint techniques introduced above. Giering and Kaminski [37] and Griewank [45] give a general introduction, supported by rules and examples illustrating the use of AD techniques. AD has been applied for fluid dynamic design optimisation in several studies [e.g. 94, 97, 41, 82, 16, 42, 43, 111, 69, 112, 9]. The non-commercial community portal Autodiff.org [3] gives an overview of recent trends and developments in AD.

In the context of CFD sensitivity analysis, some crucial AD-characteristics are:

- AD precompilers are specific to the programming language of the input code. The source code of the entire process chain (between input and output) must be available in order to be precompiled. Among others Fortran-based AD tools are TAF (Fortran77/90), TAC (C), TAC++ (C++ subset), ADIFOR (Fortran77) and TAPE-NADE (Fortran77/90 and C) and Treeverse (C, C++ subset).
- Application of AD to modern, object-oriented code structures is not straight-forward. However, features such as operator overloading are supported by some AD tools.
- Input and output of the computer program have to be declared to the AD pre-compiler. Often, manual code-preparation is required before the source code is differentiated “automatically”. Depending on the precompiler, non-differentiable code-statements or unsupported language features need to be eliminated or rewritten in the input code.
- Particularly in reverse mode some human intervention is necessary to derive efficient AD-code [94, 97, 16, 69, 9]. It requires a profound user-knowledge of AD principles, the specific AD precompiler and the structure of the input code.

3. Review of Adjoint-based Shape Optimisation

A selection of central issues observed in the context of adjoint CFD is discussed in the following. The focus is on incompressible hydrodynamics. Aerodynamic applications are considered insofar of interest for the current study since, in large part, the development of the adjoint method has been driven through the aerospace community.

3.1. Continuous and Discrete Differentiation

Adjoint CFD-codes can be subdivided into continuous and discrete adjoint approaches, which are obtained following the derive-then-discretise and the discretise-then-derive strategy, respectively. In the context of compressible aerodynamics, the pros and cons of either method have been discussed by several authors [38, 104, 99, 101, 116]. Though the solution strategy is usually density-based and (partially) coupled, many conclusions drawn in the context of compressible aerodynamics are also valid for pressure-based, segregated algorithms used in incompressible hydrodynamics.

The discrete adjoint system is derived from the discretisation of the primal by transposing the linearised operators. Thus it provides the exact derivative of the discrete objective function, so that the discrete adjoint identity (2.42) is satisfied. When the full Jacobian system matrix is available and stored in the reference code, the corresponding adjoint system matrix is obtained by transposition, cf. Section 2.4. This can be easily coded in a computer program. The right-hand side is assembled depending on the definition of the objective function. However, most non-trivial, coupled finite-volume solvers avoid to store the complete Jacobian matrices of the second-order discretisation [e.g. 96, 113, 109, 116]. Usually, only the adjacent neighbours are considered implicitly in (unstructured) face-based finite-volume schemes. If the Jacobian system matrix is unavailable in explicit algorithms or only a simplified lower-order estimate of the residual is available on the left-hand side, an explicit defect correction is required in the adjoint code [e.g. 38, 109, 113]. Nadarajah [101, Chapter 4] reports that the effort for a manual derivation of the discrete adjoint system grows rapidly with the size of the numerical stencil. This is obvious since the adjoint schemes have to be individually devised from the primal ones according to Eqns. (2.37) and (2.38). The adjoint schemes involving remote neighbours can be evaluated sequentially in reverse order as done by automatic differentiation, cf. Giles [38]. Alternatively, a simplified version of the original primal can be transposed so that the discrete derivative of a reduced description is calculated [21].

Some of the inconveniences and limits of hand-coded discrete adjoints can be circumvented through automatic differentiation in reverse mode [37, 45], a special case of the discrete adjoint method. Source-to-source transformation is generally independent of the underlying physics or solution algorithms. Adjoint codes generated by automatic differ-

entiation traverse the trajectory of the primal (input code) in reverse. Generally, the full trajectory of the primal state is required in the reverse run. Storing the trajectory to disk can be prohibitively expensive for large-scale computations. So-called check-pointing directives are used in order to trade memory storage for CPU-time [e.g. 37, 46, 163]. A number of states, so-called check-points, are stored in the primal run at given intervals. During the reverse run, the state is restored from the closest checkpoint and recomputed to the current (time) step by restarting the primal code. According to Christianson [12], convergent loops, such as iterations against steady state, may be linearised about the converged primal state so that the adjoint converges at the rate of the primal. Though the sensitivity code is derived on the basis of the input code alone, so-called AD-directives have to be added by the user in order to enhance the performance of the AD code. Memory and CPU consumptions of adjoint, AD-based CFD codes have been discussed and optimised by several authors [e.g. 97, 41, 16, 42]. A number of studies on aerodynamic, AD-based shape optimisation have been performed [94, 95, 41, 97, 16, 43]. Hydrodynamic applications of viscous CFD are presented by Othmer et al. [111], Özkaya and Gauger [112] and Carnarius et al. [9]. Huge software projects such as industrial general-purpose solvers typically use modern code structures and consist of different programming languages. Often external libraries are involved, sophisticated numerical toolkits for preconditioning and solving large systems of equations. For parallel algorithms, the corresponding parallel code structures have to be created automatically. A considerable amount of manual preparation of the input code can be required, as explained e.g. by Giering and Kaminski [37], Müller and Cusdin [97], and Cusdin and Müller [16]. The applicability of AD in reverse mode for the differentiation of the full design chain in one go is expected to be limited for such problems.

In the continuous adjoint approach, the adjoint PDE are derived from the PDE governing the primal problem according to Section 2.3. Subsequently, the adjoint PDE are discretised. The continuous adjoint formulation allows for a mathematical and physical interpretation of the adjoint problem, including field equations, boundary conditions and initialisation. The background knowledge of the adjoint PDE can also be helpful to obtain analytic solutions, e.g. for validation, or for the numerical treatment of discontinuities as discussed by Giles and Pierce [40]. The flexibility of the continuous adjoint method can considerably facilitate adjoint code development [67]. Adjoint discretisation schemes and iterative solution algorithms can be derived for the adjoint PDE independently of the primal schemes. Its flexibility is expected to be one of the main reasons why the continuous adjoint method is often used in conjunction with incompressible, segregated pressure-based solvers widely used in automotive and marine industry, see Section 3.2. When higher-order schemes are required, the continuous adjoint approach allows to freely derive and implement discretisation schemes of higher order. Special care is needed to provide a certain level of consistency of the discrete objective functional value and its derivative [102, 99, 21, 50, 39]. Consistency of the discretisation according to Eqn. (2.42), if not considered explicitly, will probably be violated. The formulation of adjoint boundary conditions is related to the definition of boundary-declared objective functionals. According to Eqn. (2.21), there are restrictions on the possible formulations of objective functionals that can be addressed through the continuous adjoint calculus, i.e. the declarations of the

objective functional and the state equations must be compatible. Some RANS closure models have a singularity next to the boundary or include complex source terms, impeding a continuous adjoint formulation. In primal CFD, the problem can be tackled in the discrete space, e.g. Wilcox [166, Sections 4.7 and 4.9], which directly leads to hybrids of continuous and discrete adjoint techniques. *Hybrid methods* can be pursued to combine the individual advantages [99] of the continuous and the discrete adjoint method. In the practical development of adjoint production codes, the consistency of the discrete method can be traded for the flexibility of the continuous approach. Certain formulations that cannot be handled in the continuous adjoint calculus can be addressed on the basis of the discrete. The knowledge of the underlying adjoint PDE allows to accept or justify certain inconsistencies which may facilitate the algorithmic implementation.

According to Kim et al. [74] and Jameson and Kim [64], the derivatives obtained through both the discrete and the continuous gradients should match in the *continuous limit* of infinitesimally fine mesh resolution. Ideally, the discretisation of the adjoint PDE satisfies the adjoint identity, Section 2.4, so that the derive-then-discretise method and the discretise-then-derive methods lead to the same adjoint discretisation.

3.2. Primal and Adjoint Algorithms

The most obvious strategy, for both the discrete and the continuous adjoint method, is to reuse the architecture and the data-structure of the primal solver in the adjoint code.¹ The features of the primal code are usually re-employed to solve the adjoint as regards the iterative solution strategy, the discretisation, the concepts for parallel computation, preconditioning, convergence acceleration, time-stepping, etc.

Unsteady adjoint solution schemes are required for unsteady optimisation problems. After integration by parts (continuous adjoint) or summation by parts (discrete adjoint) with respect to time, the direction of the time integration changes. Hence, the adjoint runs in reverse direction from the end of the primal simulation back to the initialisation of the primal, i.e. the adjoint “initialisation” is prescribed at the end of the primal simulation. As the adjoint is linearised about the unsteady primal, the flow solution has to be restored or recomputed every adjoint time-step, which is inconvenient and costly from a computational point of view. To optimise the shape of airfoil-sections in inviscid transonic flow for a time-varying angle of attack, Nadarajah [101] investigates three methods of different numerical effort: “unsteady-flow unsteady-adjoint”, “unsteady-flow steady-adjoint” and “time-averaged-flow steady-adjoint”. Nadarajah et al. [100] apply an adjoint method, formulated in the non-linear frequency domain, to optimise a wing in periodic three-dimensional inviscid flow. Several checkpointing strategies have been devised in the context of automatic or manual differentiation [e.g. 37, 46, 163]. A checkpointing algorithm for adaptive time-stepping schemes is proposed by Wang and Moin [164]. A strongly unsteady adjoint analysis for blast flows is carried out by Stück et al. [153] to aid in the design of shock-mitigation devices. A combination of measures is proposed by Stück et al. [154] to optimise the adjoint efficiency, including two-level check-pointing,

¹Unless the goal is a stand-alone or solver-independent adjoint solver.

data-compression and a reduced data precision together with a case-dependent memory management to reduce the I/O operations.

Implicit or explicit time-stepping schemes can also be used to drive the flow against the limit of steady-state. First-order Euler schemes are often applied if the time history is not important. The corresponding adjoint scheme can be linearised about the converged steady-state solution, so that the trajectory of the primal convergence does not need to be restored. Aerodynamic, density-based applications are usually formulated in a (partially) coupled system of conserved variables. The Jacobian system matrix is assembled of point-wise $[5 \times 5]$ sub-matrices, formulated in the conservative variables for density (ρ), density times velocity components (ρU_i) and density times specific energy (ρe). Jameson et al. [66] suggest a bundle of methods to efficiently iterate flow and adjoints to steady state. Based on a generalised Runge–Kutta scheme, in which the convective and diffusive terms are treated differently to enlarge the stability region, they use residual averaging and a multigrid procedures. Convergence is further accelerated by the use of locally varying time steps or matrix preconditioning. Further improvements are presented by Kim et al. [79]. Implicit methods generally allow for large time steps or efficient steady-state simulations. Nielsen et al. [109] and Dwight and Brezillon [23] apply implicit time-stepping. Implicit methods often use a simplified Jacobian on the left-hand side. To obtain an accurate linearisation in the linearised code, the implicit low-order approximations have to be compensated by explicit defect corrections. The hand-written discrete adjoint code for transonic Navier–Stokes flow presented by Nielsen et al. [109] ensures duality with the direct-differentiation method throughout the convergence history. This is achieved by a fully-consistent discretisation together with a reverse implementation of the implicit, extended line-Gauss–Seidel scheme used for the primal. The dual algorithm is manually devised from the primal so that the convergence rates are asymptotically equivalent. An overview of recent adjoint aerodynamics is given by Peter and Dwight [116].

An artificial compressibility term can be added to the continuity equation in order to solve incompressible flow problems. This strategy is pursued for adjoint-based shape optimisation by several research groups in combination with multigrid methods for convergence acceleration [90, 19, 68, 91].

So-called pressure-projection schemes, based on pressure- or pressure-correction algorithms, are predominantly used in industry to solve incompressible flow problems, e.g. Ferziger and Peric [30, Sections 7.4 and 8.8]. A Poisson-type pressure equation or pressure-correction equation is obtained by applying the divergence operator to the momentum equations either on a continuous or a discrete level. Usually, an incomplete linearisation is used. The individual transport equations for the velocity components, the pressure-correction scheme and turbulence variables are solved sequentially. If necessary, the sequence is extended to additional transport equations for energy, multi-phase flow, etc. The individual iteration matrices and right-hand side terms are updated during the iteration based on the “known” quantities. It is not straightforward to manually derive the exact adjoint algorithm for segregated schemes such that the primal convergence process is exactly traversed in reverse. It requires forward-reverse consistency of the solution sequence (momentum \rightarrow pressure \rightarrow turbulence \rightarrow multi-phase, etc.), including deferred corrections for second-order accuracy, pressure and velocity corrections, time-stepping scheme,

etc. The continuous adjoint technique generally allows to re-employ the segregated primal algorithm and its discretisation for the adjoint problem with slight modifications. A compact implementation is obtained by reusing huge portions of the flow solver in the adjoint code [147, 110, 155]. It reduces coding effort, avoids code redundancies and facilitates code maintenance. The functionality of complex production codes is provided in the adjoint code as regards the parallel concept, the unstructured bookkeeping or the discretisation schemes. It may be the main reason why many non-trivial hand-coded adjoint solvers for segregated primal codes follow the continuous adjoint approach [8, 110, 170]. Soto and Löhner [147] present an implicit pressure-correction approach based on the finite-element method for both continuous and discrete adjoint problems.

3.3. Gradient Evaluation

Adjoint methods eliminate the partial variation of the augmented objective functional with respect to the state via the adjoint variables found from the adjoint equations. The gradient can subsequently be evaluated from metric terms alone, either in the continuous or the discrete adjoint way, cf. Eqns. (2.26) and (2.41), respectively.

In the discrete adjoint approach, the evaluation of Eqn. (2.41) generally involves the calculation of the displacements of the interior mesh nodes with respect to the boundary deformations associated with every shape parameter; it is called *mesh sensitivity*. On structured grids the dependence can be evaluated at a moderate effort following the grid lines. On unstructured grids, the mesh sensitivity has to be evaluated through a mesh deformation approach. The application of finite-differencing or complex-step differentiation techniques is straight-forward, however, the effort scales with the number of parameters involved. For a large number of parameters the calculation of mesh sensitivities may become a bottleneck in the adjoint-based sensitivity analysis. Nielsen et al. [109] suggest to apply decomposition techniques to the coefficient matrix of the mesh deformation approach, so that the problem can efficiently be solved for several right-hand sides. An alternative approach is used by Nielsen and Park [108] who set up the discrete adjoint to the mesh deformation algorithm to compute the sensitivity derivatives at a computational effort that is practically independent of the number of shape parameters.

The continuous, PDE-constrained shape optimisation problem is *boundary-controlled* suggesting a boundary-based gradient evaluation. Based on a linear development of the boundary condition to a modified position, a boundary-based gradient expression is obtained, cf. Eqn. (2.26). A boundary-based gradient calculation was first shown by Enoksson and Weinerfeld [25], and subsequently pursued by several authors in the context of adjoint-based shape optimisation [e.g. 64, 149, 110, 170]. In conjunction with unstructured grids, a boundary-based gradient equation is particularly attractive since the volume mesh does not need to be adapted to the infinitesimal deformations.

3.4. Turbulence Treatment

In general, the complete flow field changes when the shape is modified. The considered hydrodynamic objective functional is, directly or indirectly, affected by changes in all state variables. The transport equations associated with the turbulence variables have to be derived in order to calculate the full gradient of the objective functional.

The assumption of *frozen-turbulence* is the most convenient simplification and is considered industrial standard for both continuous [143, 151, 110, 156] and discrete adjoint methods [106, 101]. The optimisation problem is partially linearised about the current state neglecting the variation of the turbulence field. Only changes of the mean flow are taken into account, described through the system for continuity and momentum for incompressible flow problems—or mass, momentum and energy for compressible flow problems. Accordingly, the gradient obtained by freezing the turbulence field is incomplete.

In the framework of the continuous adjoint method, the full variation and the adjoint problem are derived on a PDE level. Zero-, one- and two-equation turbulence models usually include several non-linear source terms described through turbulence quantities and the mean velocity field. The number of terms in the fully-linearised problem or the full adjoint problem can be significantly higher and the cross-coupling usually densifies. The character of the cross-coupling between the individual equations can be very different in primal and dual problems. Several limiters or physical constraints (e.g. positiveness, boundedness) and switches (e.g. based on the wall distance) are included in most turbulence models, for which a continuous differentiation is not obvious. Some RANS closure models have a singularity along the boundary, which is often circumvented by the declaration of “pseudo boundary conditions” for the turbulence variables in the control volume next to the boundary face [166]. Such techniques are difficult to translate in the continuous adjoint context. Only a few continuous adjoint RANS solvers feature a complete adjoint formulation. Zymaris et al. [170] have derived the adjoint Spalart–Allmaras turbulence model and the adjoint of the Wilcox k - ω turbulence model in conjunction with wall functions [171]. Both adjoint formulations are very complex in terms of the number of source terms and the adjoint cross-coupling. Also the gradient equation is lengthy (approximately ten boundary terms, in parts differential) and partly difficult to evaluate in a finite-volume environment. The authors identify the dominating terms in order to reduce the effort and show the differences compared to the frozen-turbulence prediction. They report that the effort of solving the adjoint problem is comparable to that of the primal RANS problem. The method is applied to 2D incompressible, internal-flow problems, where the power loss is considered as objective functional.

In the majority of fully turbulent adjoint methods, the hand-coded discrete-adjoint approach is used in combination with coupled, compressible solvers for aerodynamic flow. Several authors have derived and solved the compressible, discrete adjoint RANS problem for a Baldwin–Lomax zero-equation turbulence model [e.g. 74, 96, 117]. Examples for a discrete adjoint of a one-equation Spalart–Allmaras turbulence model with application to compressible flow problems can be found in [1, 2, 103, 109]. The adjoint system including the additional adjoint turbulence variable and the corresponding equation is solved in a tightly-coupled way; Nielsen et al. [109] use a strictly reverse implementation of the

primal code to solve the corresponding adjoint problem including the turbulence equation. Hand-coded discrete-adjoint implementations for two-equation RANS solvers of $k\text{-}\varepsilon/\omega$ -type feature two additional adjoint variables and the corresponding adjoint equations; such adjoint implementations for compressible, aerodynamic problems can be found in [73, 125, 124].

Using automatic differentiation, the turbulence procedures of a CFD-code can be differentiated all at once along with the main routines of the solver. The lengthy algebraic expressions associated with the discrete adjoint formulation are created more or less automatically following the chain rule of differentiation as described in Section 2.4. Parts of the turbulence model, which are indifferentiable in the continuous formulation—e.g. singularities in the boundary formulations of several turbulence models [166]—are algorithmically differentiable. Turbulent adjoint examples for aerodynamic applications are presented by Mohammadi [93] and Giles et al. [41]. Carnarius et al. [9] apply AD to a segregated, incompressible Navier–Stokes solver in conjunction with the SST $k\text{-}\omega$ turbulence model. The method is used to control the flow control around a rotating cylinder.

The adjoint method has also been used in conjunction with scale-resolving simulations of turbulent flow. The analysis is inherently unsteady and requires a backwards time-integration of the adjoint system. Chang and Collis [10] use adjoint-based LES for flow control to reduce the resistance of the bounding walls of a straight channel by means of wall transpiration. The adjoint system is build for the Smagorinsky subgrid-scale model. The adjoint time integration is carried out in reverse mode. Due to the strong coupling of the adjoint PDE system, the adjoint equations are more expensive to solve than the flow equations. Aeroacoustic adjoint LES is carried out for jet noise minimisation by Kim et al. [75] and Freund [32].

3.5. Accuracy and Consistency

The computed gradient is considered accurate if it is the exact, complete derivative of the primal discretisation, which is in principle assured by the discrete adjoint method. In that sense, a derivative based on a consistent discretisation—which may be of low order or suffer from substantial modelling insufficiency—is accurate. A different question is whether the adjoint discretisation obtained in the discretise-then-derive approach is a valid approximation to the adjoint PDE devised in the continuous (derive-then-discretise) approach. It requires that an adjoint PDE formulation exists and that the primal PDE and the objective functional are discretised consistently in the primal code. Following the continuous adjoint approach requires to choose an appropriate, stable discretisation for the adjoint PDE. When such an approximation is found, the question is how accurate it is compared to the gradient of the discrete objective function of primal solver. Direct comparisons of the continuous and the discrete adjoint method for aerodynamic inviscid and viscous cases have been performed by Nadarajah and Jameson [102, 99]. In a hybrid adjoint approach pursued in this work, the discretisation for the adjoint PDE, which has been derived beforehand, is found by analysis of the primal discretisation. In that way, the discretisation schemes for the individual terms of the adjoint PDE can be chosen

consistently with the primal schemes so that ideally both the discrete adjoint identity (2.42) and the continuous adjoint identity (2.28) are satisfied. Such schemes are denoted *adjoint-consistent* in this work, compare for instance Hartmann [50].

Ideally, both primal and adjoint problems share the same turbulence modelling. A major source of inconsistencies is the widely accepted assumption of frozen turbulence described above. Since the turbulence field is considered to be independent of the shape variations, an incomplete derivative is obtained. The resulting mismatch between objective functional values and derivatives can affect the performance of the optimisation. However, a continuous update of the primal turbulence field during the iterative optimisation identifies if the optimisation is led into the wrong direction; i.e. the optimisation does not converge any further.

In the shape optimisation process, gradient accuracy is required insofar it affects the performance of the selected optimisation strategy in terms of robustness and speed. Consistency of the discrete objective function and its derivative is a premise for an optimisation procedure to fully converge. Simple steepest descent methods have proven robust to poor gradient input [22]. When surrogate models are used, gradient errors can deteriorate the local model of the design space; Dwight and Han [24] observed a poor performance of gradient-based surrogate models—such as co-kriging [168, 77]—based on erroneous gradients.

Preconditioning of sensitivity derivatives is often used in combination with CAD-free optimisation to enhance the performance of the optimisation procedure, and to obtain shapes that are desirable from a technical point of view. The preconditioning is a manifest manipulation of the sensitivity derivative, posing the question whether an accurate derivative is required or only general trends should be reflected in the derivative. Smoothing techniques based on “Sobolev smoothing” [65, 138], filtering [152] or multigrid methods [69] damp out unwanted, high-frequency oscillations, which are often present in the mesh-based sensitivity derivative with respect to the shape. Such irregularities may also have a numerical origin: Sakamoto and Kawahara [134] note that they obtained a rough sensitivity derivative on unstructured grids using a finite-element discretisation, whereas the same approach yielded a smooth derivative on structured grids.

Pironneau [120], Laporte and Tallec [82] make a much more restrictive assumption. In a reduced gradient formulation only the partial variation with respect to the shape is considered. The partial variation with respect to the flow is neglected, so that no adjoint problem has to be solved at all. The validity and the success of such assumptions is considered to strongly depend on the optimisation problem at hand.

Gradient inaccuracies can be a consequence of incomplete iterations. However, the constraints represented by the flow equations are only required to be satisfied by the final converged solution of the optimisation process. This can be exploited in order to reduce the overall optimisation effort by limiting the number of flow and adjoint iterations per optimisation cycle. In the limit of one flow and adjoint iteration per design cycle, it leads to the one-shot (or piggyback) algorithm—i.e. the optimisation problem, the state and the dual are iterated in just one integrated loop [81, 36, 54]. Özkaya and Gauger [112] use a one-shot strategy to minimise the drag of a NACA4412 section in incompressible flow using the k - ω turbulence model. The CAD-free process chain is automatically differentiated in

reverse mode.

Approximation and modelling errors usually change while the design evolves during an optimisation. Re-generation or adaptation of the computational grid for instance, is accompanied by a change of the numerical error. Also the validity of the CFD model can be questionable in critical points. The appropriate numerical input parameters may differ in different locations in the design space. Automatic monitoring of such influences during the optimisation is a difficult task and depends on the particular application. Looking at the evolution of the objective functional alone, it is possible that the evaluation error is “optimised” by the procedure instead of the physical quantity of interest.

3.6. Integration into Shape Optimisation

Both the continuous and the discrete adjoint method provide the derivative of the objective functional with respect to the position of the surface mesh nodes as an intermediate solution. For complex industrial shapes the discrete derivative contains a couple of (ten) thousand degrees of freedom on the design surface(s), offering a detailed insight into the design opportunities from the objective point of view. The “raw” sensitivity derivative with respect to the shape can be used to drive a deterministic, gradient-based shape optimisation in different manners:

Global shape functions, such as a number of Hicks–Henne basis functions [56], analytic descriptions of foil-sections (e.g. NACA) or free-form deformation techniques [139, 123, 130, 20], are associated with a predefined set of parameters. Noisy shapes are prevented by limiting the design space a priori. The global support inherent to many techniques is problematic in the context of complex 3D geometries. Soto et al. [150, 151] present a pseudo-shell approach based on *local shape functions* (finite-element method) that couples the perturbation of an arbitrary mesh point to the other points on the design surface. The coupling of points is inherently smooth, works in an implicit manner and suppresses undesirable modes in the shape.

If a *CAD-model* is involved and the connectivity between the CAD-parameters and the surface mesh is available, the mesh-based derivatives can be linked to the CAD-parametrisation using the chain rule of differentiation [89, 128]. Moreover, the grid sensitivity, representing the derivatives of the position of the interior mesh nodes with respect to the position of the boundary nodes, must be determined. Unless a reduced, boundary-based gradient formulation is used, the grid generator (re-meshing strategy) or the mesh-deformation algorithm—e.g. tension spring analogy [5, 162], torsion spring analogy [27, 98], linear elasticity analogy [107] or explicit approaches based on inverse distance weighting [82, 33, 167]—must be differentiated. An appropriate CAD parametrisation ensures suitable shapes—however, a rigid, low-dimensional setup limits the shapes that can be generated and different CAD-models may lead to different optimal shapes. The CAD-systems predominantly used in industry are proprietary, so that the underlying shape definitions are not available explicitly. This requires the numerical differentiation of the black-box CAD-method.

The most obvious strategy is to directly employ the mesh-based derivatives to the de-

sign surface [95, 65, 69, 138, 26]. The method is referred to as *CAD-free* since no external geometry description beyond the computational mesh is involved. The approach retains all information that is contained in the derivative. As the mesh nodes on the design surface are (independent) degrees of freedom, some regularisation requirements have to be fulfilled to provide sufficiently smooth shapes which are feasible and desirable from a technical point of view. The suitability of shapes is linked to the performance of the optimisation process in terms of robustness and speed. In local shape optimisation, the convergence rate of gradient iterations strongly depends on the (local) condition of the Hessian of the objective functional. Through preconditioning operations, the independent variables can be rescaled so that the condition number of the Hessian and thus the convergence rate of the optimisation algorithm increases. In this context, the Riesz representative of the objective functional derivative with respect to the shape is usually denoted as the gradient of the objective functional. It assumes smooth distributions for appropriate choices of inner products. Accordingly, different gradients are associated with different transformations applied to the derivative. Several preconditioning techniques are proposed to increase the level of regularity or smoothness of the derivative with respect to the shape. Jameson and Vassberg [65] apply an implicit, continuous smoothing operator to the derivative, based on elliptic second-order damping derived from an extended definition of the inner product (frequently called “Sobolev gradient”). Jaworski and Müller [69] compare several strategies including implicit, explicit Laplacian smoothing and multi-grid techniques in the context of adjoint-based CAD-free shape optimisation. Different smoothing operations based on shape Hessian preconditioners are investigated in conjunction with potential flow by Eppler et al. [26] and Euler flow by Schmidt et al. [138].

3.7. Hydrodynamic and Marine Applications

In the 1970s and 1980s potential flow techniques were explored for hydrodynamic and marine design. Potential flow simulations are attractive where the flow features of interest are predominantly irrotational and solenoidal, such as gravity waves or hydrodynamic lift. Early applications of optimal flow control for hydrodynamic shape optimisation are discussed by Pironneau [120]. Söding [140, 141, 142] presents a discrete adjoint potential-flow solver that calculates the sensitivity derivative of the resistance of a ship hull. The sensitivity derivative is calculated with respect to surface-normal deformations of the hull form. The influences of hydrodynamic trim and sinkage as well as modelled skin friction are considered in the sensitivity analysis. Guided by the sensitivity derivative, a manual, CAD-based hull redesign is carried out. Ragab [121, 122] uses a continuous adjoint potential flow method to optimise a simplified submarine operating near the free surface and a surface-piercing Wigley-like hull. He calculates the sensitivity derivative of the wave drag with respect to the hull shape, which is controlled via B-spline curves. Also the quadratic deviation from a target pressure distribution over the hull is considered in a so-called “inverse-design” approach. The shape optimisation is carried out by the steepest-descent algorithm.

Dreyer and Martinelli [19] solve the 3D incompressible Euler equations in rotating Cartesian coordinates by a pseudo-compressibility approach. The methodology is used in an “inverse-design” approach, applied to a rotor-stator propulsor configuration. Soto and Löhner [146] and Soto et al. [151] optimise the bow-shape of a Kriso-Container Ship (KCS) by means of an adjoint solver for incompressible Euler flow. The elevation of the free water-surface is taken into account in the primal state by solving a surface equation on the still-water surface. The free-surface elevation is neglected in the sensitivity analysis and the authors conclude, that it needs to be considered to further reduce the resistance force. In the same study, hydrofoils are optimised for different pressure-based criteria using the adjoint Euler method. The wave resistance of a surface-piercing Wigley-hull is addressed by Martinelli and Jameson [91]. They use a continuous adjoint, block-structured Euler method that accounts for the shape-dependency of the free water surface by an interface-tracking approach [28]. The optimisation follows the direction of steepest descent.

Hino [58] combines RANS-based discrete adjoint sensitivity analysis and sequential quadratic programming (SQP) to reduce the viscous drag of a generic tanker hull. Moreover, a Series-60 hull is optimised with respect to the drag. The RANS solver is based on structured multi-block grids. The dependence of both the turbulent viscosity and the free water-surface on the shape is neglected in the sensitivity analysis. A continuous adjoint method for incompressible Euler and RANS flow is presented by Martinelli and Cowles [90]. They calculate the shape sensitivity to minimise the deviation from a target pressure distribution for wings and wing-sections. The inverse design is carried out in a steepest-descent approach. Flow and adjoint solvers are based on structured grids and use the artificial-compressibility method. The sensitivity analysis assumes that the turbulence is “frozen” with respect to the shape. Stück et al. [155, 156] and [80] use the adjoint RANS method to calculate the derivative of a wake objective functional with respect to a local change of the hull displacement. In the adjoint analysis, the eddy viscosity is frozen with respect to the shape. Guided by the sensitivity derivative, a generic PanMax container vessel is redesigned manually to improve the quality of the velocity wake field. RANS-based finite differencing is applied by Tahara et al. [158] to calculate the desired sensitivity derivatives with respect to the shape. Bow and aft-ship of a surface combatant are optimised via an SQP algorithm taking into account the water surface. The authors conclude that the performance of the costly optimisation can be improved in combination with adjoint-based techniques.

4. Hydrodynamic Optimisation Problem

“From simulation to optimization...” Corliss et al. [14]

Design configurations are evaluated in terms of scalar, integral, hydrodynamic design criteria denoted as objective functionals. The integrand of the hydrodynamic objective functional depends on the flow which, in turn, is governed by the incompressible RANS equations for steady state. These are considered a reasonable efficiency vs. accuracy tradeoff for many hydrodynamic optimisation problems.

The RANS-constrained optimisation problem is described in the following including the governing field equations for momentum, continuity and turbulence, the corresponding boundary conditions and a base-line collection of hydrodynamic objective functionals. The presentation is not exhaustive. The focus is on the features that are of interest for the derivation of the variation of the Navier–Stokes problem and its adjoint counterpart in Chapters 5 and 6.

4.1. Governing Field Equations

Inside the flow domain Ω , the fluid flow is governed by the incompressible, steady-state RANS equations complemented by the turbulence equations.

Navier–Stokes Equations

The incompressible, steady-state RANS equations read

$$\rho \langle U_j \rangle \frac{\partial \langle U_i \rangle}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial \langle U_i \rangle}{\partial x_j} + \frac{\partial \langle U_j \rangle}{\partial x_i} \right) - \langle p \rangle \delta_{ij} - \rho \langle U'_i U'_j \rangle \right] + f_i \quad (4.1)$$

and

$$-\frac{\partial \langle U_i \rangle}{\partial x_i} = 0. \quad (4.2)$$

Eqns. (4.1) and (4.2) are written in terms of the Reynolds-averaged velocity and pressure values ($\langle \underline{U} \rangle, \langle p \rangle$), which are related to the corresponding fluctuating quantities (\underline{U}, p) and the fluctuations (\underline{U}', p') by

$$U_i = \langle U_i \rangle + U'_i \quad \text{and} \quad p = \langle p \rangle + p'. \quad (4.3)$$

The fluid density ρ , the dynamic and the kinematic viscosity, μ and $\nu = \mu/\rho$, are not subject to turbulent fluctuations. The Reynolds-stress tensor $\rho \langle \underline{U}' \underline{U}' \rangle$ is symmetric and consists of six independent, unknown entries. In order to compute the mean turbulent

flow, a prescription or modelling for the Reynolds-stresses in terms of the mean flow variables is required. The body-force vector \underline{f} represents, for example, the acceleration of gravity

$$f_i = -\rho g \delta_{i3}, \quad (4.4)$$

with the Kronecker delta δ_{ij} . According to the Boussinesq eddy-viscosity hypothesis, the deviatoric portion of the Reynolds stress tensor is proportional to the mean rate of strain

$$-\rho \langle U'_i U'_j \rangle + \frac{2}{3} \rho k \delta_{ij} = 2 \mu_T \langle S_{ij} \rangle, \quad (4.5)$$

where the turbulent kinetic energy is defined as half the trace of the Reynolds stress tensor

$$k = \frac{1}{2} \langle U'_i U'_i \rangle. \quad (4.6)$$

The strain-rate tensor represents the symmetric portion of the velocity gradient

$$\langle S_{ij} \rangle = \frac{1}{2} \left(\frac{\partial \langle U_i \rangle}{\partial x_j} + \frac{\partial \langle U_j \rangle}{\partial x_i} \right). \quad (4.7)$$

The proportionality constant μ_T introduced in Eqn. (4.5) is called eddy viscosity. Eqn. (4.1) can be rewritten to read

$$\rho \langle U_j \rangle \frac{\partial \langle U_i \rangle}{\partial x_j} = \frac{\partial}{\partial x_j} \left[2 \mu_{\text{eff}} \langle S_{ji} \rangle - \left(\langle p \rangle + \frac{2}{3} \rho k \right) \delta_{ij} \right] + f_i, \quad (4.8)$$

with the effective viscosity $\mu_{\text{eff}} = \mu + \mu_T$ and the modified mean pressure $(\langle p \rangle + \frac{2}{3} \rho k)$. In the following description, the Reynolds-averaged values are simply referred to as velocity or stress components for the sake of brevity. The modified mean pressure, $\langle p \rangle \leftarrow (\langle p \rangle + \frac{2}{3} \rho k)$, is denoted as pressure and the angular brackets are left out except for the Reynolds stress tensor.

A local residual formulation of the Reynolds-averaged equations for momentum and continuity, Eqns. (4.8) and (4.2), reads

$$R_i = \rho U_j \frac{\partial U_i}{\partial x_j} - \frac{\partial \pi_{ij}}{\partial x_j} - f_i = 0 \quad \text{in } \Omega \quad (4.9)$$

and

$$Q = -\frac{\partial U_i}{\partial x_i} = 0 \quad \text{in } \Omega. \quad (4.10)$$

The hydrodynamic stress tensor

$$\pi_{ij} = -p \delta_{ij} + \tau_{ij}, \quad (4.11)$$

can be decomposed into the pressure stress, p , and the viscous stress tensor

$$\tau_{ij} = 2 \mu_{\text{eff}} S_{ij}. \quad (4.12)$$

Turbulence Models

Depending on the turbulence model, the turbulent viscosity is determined from the turbulent length and time scales (l_T, t_T)

$$\nu_T = l_T^2 / t_T. \quad (4.13)$$

One of the most popular eddy-viscosity turbulence models is the two-equation k - ε model by Jones and Launder [70]. The eddy viscosity is determined from the turbulent length and time scales according to Table 4.1. The turbulent kinetic energy k and its dissipation rate ε are obtained from the transport equations

$$T_1 = \frac{Dk}{Dt} - \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_T}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] - P + \varepsilon = 0 \quad (4.14)$$

and

$$T_2 = \frac{D\varepsilon}{Dt} - \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_T}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_j} \right] - C_{\varepsilon 1} \frac{P\varepsilon}{k} + C_{\varepsilon 2} \frac{\varepsilon^2}{k} = 0. \quad (4.15)$$

The production of turbulent kinetic energy is

$$P = -\langle U'_i U'_j \rangle \frac{\partial U_i}{\partial x_j} = \nu_T \mathcal{S}^2, \quad (4.16)$$

with the rate of strain defined as $\mathcal{S} = \sqrt{2 S_{ij} S_{ij}}$. The involved closure coefficients are

$$C_{\varepsilon 1} = 1.44, \quad C_{\varepsilon 2} = 1.92, \quad C_\mu = 0.09, \quad \sigma_k = 1.0 \quad \text{and} \quad \sigma_\varepsilon = 1.3. \quad (4.17)$$

The Wilcox k - ω model [166] solves a transport equation for the specific dissipation rate or turbulent frequency ω in addition to the turbulent kinetic energy equation (4.14). The kinematic eddy viscosity is determined according to Table 4.1. The associated transport equations are

$$T_1 = \frac{Dk}{Dt} - \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_T}{\sigma^*} \right) \frac{\partial k}{\partial x_j} \right] - P + \beta^* k \omega = 0 \quad (4.18)$$

and

$$T_2 = \frac{D\omega}{Dt} - \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_T}{\sigma} \right) \frac{\partial \omega}{\partial x_j} \right] - \alpha \frac{\omega}{k} P + \beta \omega^2 = 0. \quad (4.19)$$

The model coefficients and auxiliary relations read:

$$\alpha = \frac{13}{25}; \quad \beta = \beta_o f_\beta; \quad \beta^* = \beta_o^* f_\beta; \quad \sigma = \sigma^* = 2; \quad (4.20)$$

$$\beta_o = \frac{9}{125}; \quad f_\beta = \frac{1 + 70\chi_\omega}{1 + 80\chi_\omega}; \quad \chi_\omega = \left| \frac{W_{ij} W_{jk} S_{ki}}{(\beta_o^* \omega)^3} \right|; \quad (4.21)$$

$$\beta_o^* = \frac{9}{100}; \quad f_{\beta^*} = \begin{cases} 1; & \chi_k \leq 0 \\ \frac{1+680\chi_k^2}{1+400\chi_k^2}; & \chi_k > 0 \end{cases}; \quad \chi_k = \frac{1}{\omega^3} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j}. \quad (4.22)$$

Table 4.1.: Length scale, time scale and eddy viscosity for k - ε and k - ω turbulence models

model	l_T	t_T	$\nu_T = l_T^2/t_T$
k - ε	$C_\mu^{3/4} k^{3/2}/\varepsilon$	$\sqrt{C_\mu} k/\varepsilon$	$C_\mu k^2/\varepsilon$
k - ω	$k^{1/2}/\omega$	$1/\omega$	k/ω

4.2. Boundary Conditions

The following description of boundary conditions is formulated in a local system of boundary coordinates (n, t, s) . The unit boundary vector \underline{n} points outwards of the fluid domain Ω . The system is completed by the in-plane unit vectors \underline{t} and $\underline{s} = \underline{t} \times \underline{n}$. The corresponding local velocity coordinates are defined by

$$\underline{U} = U^n \underline{n} + U^t \underline{t} + U^s \underline{s}. \quad (4.23)$$

On **no-slip walls**, the in-plane unit vector \underline{t} is aligned with the local direction of strain

$$\underline{t} = \underline{n} \cdot \underline{S} \cdot \underline{T} / |\underline{n} \cdot \underline{S} \cdot \underline{T}| \quad \text{with} \quad \underline{T} = \underline{I} - \underline{n} \underline{n}. \quad (4.24)$$

A zero relative velocity is prescribed

$$\underline{U} = \underline{U}|_{\text{wall}}, \quad (4.25)$$

which reads in the local system of boundary coordinates

$$U^t = U^t|_{\text{wall}}, \quad U^s = U^s|_{\text{wall}} \quad \text{and} \quad U^n = 0 \quad \text{on} \quad \Gamma_{\text{wall}}. \quad (4.26)$$

For negligible curvature, the incompressible equation of continuity formulated in the system of local boundary coordinates reads

$$\underline{\nabla} \cdot \underline{U} \approx \frac{\partial U^n}{\partial n} + \frac{\partial U^t}{\partial t} + \frac{\partial U^s}{\partial s} = 0 \quad \text{on} \quad \Gamma_{\text{wall}}. \quad (4.27)$$

Eqn. (4.26) implies

$$0 = \frac{\partial U^t}{\partial t} = \frac{\partial U^s}{\partial s}, \quad \text{i.e.} \quad 0 = \frac{\partial U^n}{\partial n} \quad \text{on} \quad \Gamma_{\text{wall}}. \quad (4.28)$$

In line with first-order boundary-layer theory, a Neumann boundary condition is imposed on the pressure

$$0 = \frac{\partial p}{\partial n} \quad \text{on} \quad \Gamma_{\text{wall}}. \quad (4.29)$$

The numerical approximation of boundary conditions for the turbulence quantities k , ε and ω is beyond the scope of this Chapter. In-depth descriptions are given by Wilcox [166, Sections 4.7 and 4.9] or Rung [131]. The approximations are asymptotically correct and in line with boundary-layer theory either

- (a) prescribing a boundary condition on the boundary, or
- (b) explicitly prescribing their values in the CV next to the wall, which circumvents the numerical integration of singular surface fluxes.

For boundaries with a non-zero velocity, the in-plane unit vectors \underline{t} and \underline{s} are defined as

$$\underline{t} = \underline{T} \cdot \underline{U} / |\underline{T} \cdot \underline{U}| \quad \text{and} \quad \underline{s} = \underline{t} \times \underline{n}. \quad (4.30)$$

Under the assumption of zero curvature, the subsequent **symmetry** boundary condition is obtained

$$0 = U^n = \frac{\partial U^t}{\partial n} = \frac{\partial U^s}{\partial n} \quad \text{on} \quad \Gamma_{\text{sym}}. \quad (4.31)$$

A zero-gradient Neumann boundary condition is applied to the pressure, cf. Eqn. (4.29), and any further variables ϕ such as turbulence, heat, etc.:

$$0 = \frac{\partial \phi}{\partial n} \quad \text{on} \quad \Gamma_{\text{sym}}. \quad (4.32)$$

At the **inlet** a Dirichlet boundary condition, $\phi|_{\text{in}} = \mathcal{D}$, is defined for the velocity components, the turbulence variables and further scalar variables, along with a Neumann condition (4.29) for the pressure.

A prescribed pressure value at the outlet, $p|_{\text{out}} = \mathcal{D}$, is referred to as **pressure outlet** boundary condition. The normal derivatives of the remaining velocity, turbulence and scalar variables are assumed to vanish at the pressure outlet, see Eqn. (4.32).

4.3. Hydrodynamic Objectives

The hydrodynamic optimisation problems are formulated as minimisation problems unless described differently. Hence, a reduction of the objective functional leads to an improved design. The objective functionals are classified as boundary- or volume-based

$$J = J_{\Gamma} + J_{\Omega} = \int_{\Gamma_o} j_{\Gamma} d\Gamma + \int_{\Omega_o} j_{\Omega} d\Omega. \quad (4.33)$$

The boundary- and volume-based integrands j_{Γ} and j_{Ω} are integrated over the objective surface ($\Gamma_o \subset \Gamma$) or volume ($\Omega_o \subset \Omega$), respectively. The objective functional integrands depend on the flow which, in turn, is subject to the RANS equations

$$R_i = 0, \quad Q = 0 \quad \text{and} \quad T_{1,2} = 0 \quad \text{in} \quad \Omega.$$

A base-line collection of objective functionals is presented below, which is extended for the particular applications in Chapter 11.

Force Components

A typical representative of a boundary-declared objective functional is the hydrodynamic force component acting on the objective functional surface Γ_O . The force direction is defined by the unit vector \underline{d}^* :

$$J_\Gamma = \int_{\Gamma_O} j_\Gamma \, d\Gamma \quad \text{with} \quad j_\Gamma = -\pi_{ij} n_j d_i^*, \quad (4.34)$$

wherein $\underline{n} \, d\Gamma = d\Gamma$. Often the drag is to be minimised to improve the design. A lift-maximisation problem can be turned into a minimisation problem multiplying the direction vector by -1 .

Power Loss

A typical criterion used in combination with internal flow problems is the power loss in pipe networks or its components such as ducts, bends, diffusers, nozzles, valves, junctions etc. The budget of kinetic energy is governed by the energy equation which, for incompressible flow, is a consequence of the momentum equation. In the absence of heat sources and adiabatic boundary conditions, the differential energy equation is obtained by scalar multiplication of the momentum equation and the velocity:

$$\begin{aligned} 0 &= E = U_i R_i \\ &= U_i \left[\rho U_j \frac{\partial U_i}{\partial x_j} - \frac{\partial}{\partial x_j} (2\mu_{\text{eff}} S_{ij} - p \delta_{ij}) - f_i \right] \\ &= \frac{\partial}{\partial x_j} \left[U_j \left(\frac{\rho}{2} U_i^2 + p \right) - 2\mu_{\text{eff}} S_{ij} U_i \right] + 2\mu_{\text{eff}} S_{ij} S_{ij} - f_i U_i \quad \text{in } \Omega \end{aligned} \quad (4.35)$$

Integration over the flow volume Ω according to Gauss' theorem yields

$$0 = \oint d\Gamma_j \left[U_j \left(\frac{\rho}{2} U_i^2 + p \right) - 2\mu_{\text{eff}} S_{ij} U_i \right] + \int d\Omega [2\mu_{\text{eff}} S_{ij} S_{ij} - f_i U_i]. \quad (4.36)$$

In the absence of body forces, $\underline{f} = \underline{0}$, the volume-based dissipation of energy can be expressed in terms of the boundary fluxes by virtue of Eqn. (4.36). The corresponding objective functional declared over inlet and outlet boundaries, $\Gamma_O = \Gamma_{\text{in}} \cup \Gamma_{\text{out}}$, is

$$\begin{aligned} J_\Gamma &= \int 2\mu_{\text{eff}} S_{ij} S_{ij} \, d\Omega \\ &= - \int_{\Gamma_O} \left[U_j \left(\frac{\rho}{2} U_i^2 + p \right) - \underline{2\mu_{\text{eff}} S_{ij} U_i} \right] \, d\Gamma_j. \end{aligned} \quad (4.37)$$

The underlined term in Eqn. (4.37) represents the energy-transport by viscous and turbulent stresses over the boundaries with a non-zero velocity and is neglected in the following. According to the eddy-viscosity hypothesis, the mean energy is dissipated by two mechanisms:

- Direct viscous dissipation of kinetic energy which reappears as heat or kinetic energy of molecular motion;
- Shear production of turbulence by interaction of the Reynolds-stresses and the mean flow; this term generally results in a loss of mean kinetic energy or a gain (production) of turbulent kinetic energy. It reappears with inverse sign in the budget of the turbulent kinetic energy (4.14) and (4.18).

At high Reynolds numbers, the dissipation of mean kinetic energy in the primal budget of energy (4.36) is usually dominated by the production of turbulent kinetic energy k which, in turn, is dissipated by means of viscosity. The boundary-based power-loss criterion (4.37) can be formalised according to (4.33) with the objective surface(s) Γ_O being the in- and outflow faces

$$J_\Gamma = \int_{\Gamma_O} j_\Gamma \, d\Gamma \quad \text{with} \quad j_\Gamma = -n_j U_j \left(p + \frac{\rho}{2} U_i^2 \right). \quad (4.38)$$

Inverse Design Criteria

The quadratic deviation from a target velocity distribution $\underline{U}^{\text{trg}}$ is considered as a sample for a volume-declared objective functional

$$J_\Omega = \int_{\Omega_O} j_\Omega \, d\Omega \quad \text{with} \quad j_\Omega = \frac{1}{2} (U_i - U_i^{\text{trg}})^2. \quad (4.39)$$

The search for a shape that optimally reproduces a predefined flow is often called “inverse design”.

Weighted Criteria

Provided that the weights $\underline{\alpha}$ are fixed, a multi-objective optimisation problem can be turned into a scalar problem

$$J = \sum_{k=1}^{n_O} \alpha_k J_k \quad \text{with} \quad \sum_{k=1}^{n_O} \alpha_k = 1. \quad (4.40)$$

The individual contributions J_k may either share the same objective surface (volume), or may be declared on different surfaces (volumes).

Having defined the hydrodynamic integral objectives and the PDE constraints, the following sections focus on the calculation of the objective functional derivatives with respect to the boundary shape. The direct (Jacobian) and adjoint (Lagrangian) techniques introduced in Section 2 are applied to the RANS-based shape control problem in order to derive the continuous adjoint RANS problem.

5. Variation of the Optimisation Problem

This chapter provides the main building blocks for the subsequent derivation of the continuous adjoint Navier–Stokes equations in Chapter 6. The calculus of variations for shape optimisation is based on the concept of material derivative introduced in Section 5.1. The variation of the base-line objective functionals is described in Section 5.2, followed by the linearisation of the Navier–Stokes equations and boundary conditions (Section 5.3). The chapter closes with a brief description of a direct algorithm for Navier–Stokes based sensitivity analysis (Section 5.4).

5.1. Concept of Material Derivative

The shape is controlled via a “black-box” parametrisation based on a set of parameters $\underline{\beta} [m]$. The shape parametrisation itself is beyond the scope of this thesis. The approach presented here is generic insofar it is applicable to any shape representation that is continuous and differentiable. The parametrisation is assumed to produce only feasible shapes that do not overlap or intersect.

A small perturbation of the shape parametrisation, $\delta\underline{\beta} [m]$, implies a continuous shift $\delta\underline{x}$ of the design surface Γ_D and the fluid domain Ω :

$$\delta x_k = \sum_{i=1}^m \left(\delta\beta_i \frac{\partial x_k}{\partial \beta_i} \right) + \mathcal{O} \left(\sum_{i=1}^m \delta\beta_i^2 \right) \quad \text{on/in } \Gamma_D, \Omega. \quad (5.1)$$

The rest of the boundaries, $\Gamma \setminus \Gamma_D$, remains unchanged. The distribution of the shift vector $\delta\underline{x}$ corresponds to a particular realisation or combination of shape parameter variations. Imposing the linear, spatial perturbation (5.1) on the *old* position \underline{x}_0 yields the *new* position

$$\underline{x}_1 = \underline{x}_0 + \delta\underline{x} \quad \text{on/in } \Gamma_D, \Omega. \quad (5.2)$$

A truncated Taylor series expansion of the old flow $\phi^{(0)}$ about the old grid position \underline{x}_0 yields the old flow at the new position \underline{x}_1 , viz.

$$\phi^{(0)}|_{\underline{x}_1} = \phi^{(0)}|_{\underline{x}_0} + \delta x_j \frac{\partial \phi^{(0)}}{\partial x_j} \Big|_{\underline{x}_0} + \mathcal{O}(\delta x_j^2). \quad (5.3)$$

A modification of the domain induces a change of the flow. The new flow $\phi^{(1)}$ at the new position \underline{x}_1 can be approximated to second order:

$$\phi^{(1)}|_{\underline{x}_1} = \phi^{(0)}|_{\underline{x}_0} + \sum_{i=1}^m \delta\beta_i \left[\frac{\partial \phi}{\partial \beta_i} + \frac{\partial x_k}{\partial \beta_i} \frac{\partial \phi}{\partial x_k} \right]_{\underline{x}_0}^{(0)} + \mathcal{O} \left(\sum_{i=1}^m \delta\beta_i^2 \right). \quad (5.4)$$

The subsequent analysis is carried out on the old grid, thus the indicators \underline{x}_0 are left out for brevity. ϕ refers to the old flow $\phi^{(0)}$ unless declared differently. Two contributions can be identified from (5.4), being the *local* and the *convective* variations of ϕ . The former linearly accounts for the flow change at the old grid position, whereas the latter describes the spatial variation of the old flow due to a nodal position shift $\delta\underline{x}$:

$$\sum_{i=1}^m \delta\beta_i \left(\frac{\partial\phi}{\partial\beta_i} + \frac{\partial x_k}{\partial\beta_i} \frac{\partial\phi}{\partial x_k} \right) = \left(\delta + \delta x_k \frac{\partial}{\partial x_k} \right) \phi = \underbrace{\delta\phi}_{\text{local}} + \underbrace{\delta x_k \frac{\partial\phi}{\partial x_k}}_{\text{convective}}. \quad (5.5)$$

When the concept is applied to the gradient of a field variable, $\underline{\nabla}\phi(\underline{x})$, it leads to

$$\begin{aligned} \left(\delta + \delta x_k \frac{\partial}{\partial x_k} \right) \left(\frac{\partial\phi}{\partial x_i} \right) &= \frac{\partial\delta\phi}{\partial x_i} + \delta x_k \frac{\partial}{\partial x_k} \left(\frac{\partial\phi}{\partial x_i} \right) \\ &= \frac{\partial}{\partial x_i} \left(\delta\phi + \delta x_k \frac{\partial\phi}{\partial x_k} \right) - \frac{\partial\delta x_k}{\partial x_i} \frac{\partial\phi}{\partial x_k}. \end{aligned} \quad (5.6)$$

A development of the Laplacian reads

$$\begin{aligned} \left(\delta + \delta x_k \frac{\partial}{\partial x_k} \right) \left(\frac{\partial^2\phi}{\partial x_i^2} \right) &= \frac{\partial^2\delta\phi}{\partial x_i^2} + \delta x_k \frac{\partial}{\partial x_k} \left(\frac{\partial^2\phi}{\partial x_i^2} \right) \\ &= \frac{\partial^2}{\partial x_i^2} \left(\delta\phi + \delta x_k \frac{\partial\phi}{\partial x_k} \right) - \frac{\partial}{\partial x_i} \left(\frac{\partial\delta x_k}{\partial x_i} \frac{\partial\phi}{\partial x_k} \right) - \frac{\partial\delta x_k}{\partial x_i} \frac{\partial^2\phi}{\partial x_k \partial x_i}. \end{aligned} \quad (5.7)$$

This is known as the *concept of material derivative* [57, 18, 53, 144]. The nomenclature is analogous to the transport theorem with the substantial derivative defined as

$$\frac{D}{Dt}(\cdot) = \frac{\partial}{\partial t}(\cdot) + \underline{U} \cdot \underline{\nabla}(\cdot). \quad (5.8)$$

Both material and substantial derivatives, Eqns. (5.5) and (5.8), are evaluated at the original (or old) positions, which is characteristic of Eulerian, mesh-based methods.

5.2. Variation of the Objective Functional

Like the flow, the objective functional experiences a change through a perturbation of the boundary shape. The variation of the objective functional samples introduced in Section 4.3 is presented below.

An example for a *volume-based* objective functional is the inverse design criterion (4.39)

$$J_\Gamma = \int_{\Omega_O} j_\Omega \, d\Omega \quad \text{with} \quad j_\Omega = \frac{1}{2} (U_i - U_i^{\text{trg}})^2. \quad (5.9)$$

Given that the objective volume Ω_O does not undergo a spatial shift or deformation, the objective functional variation has a *local* character indicated by the $\delta_L(\cdot)$ notation; that is changes in the objective functional are induced by changes in the flow variables ($\delta\underline{U}, \delta p$):

$$\delta_L J_\Gamma = \int_{\Omega_O} \delta_L j_\Omega \, d\Omega \quad \text{with} \quad \delta_L j_\Omega = \delta U_i (U_i - U_i^{\text{trg}}). \quad (5.10)$$

Optimisation problems with *boundary-based* objective functionals can be subdivided in two basic scenarios:

- (a) Objective functional surface(s) and design surface(s) are separate, i.e. $\Gamma_O \cap \Gamma_D = \emptyset$.
- (b) The objective functional surface Γ_O is (in parts) subject to shape variations, i.e. $\Gamma_O \cap \Gamma_D \neq \emptyset$.

An example of type (a) is the power loss criterion (4.38). The corresponding variation reads

$$\delta J = \int_{\Omega} d\Omega \left[\underline{2 \delta \mu_T S_{ij} S_{ij}} + 4 \mu_{\text{eff}} S_{ij} \delta S_{ij} \right]. \quad (5.11)$$

If the turbulence field is frozen with respect to the shape control, the underlined term of Eqn. (5.11) is zero. In the boundary-based formulation (4.38), the objective functional is evaluated on inlet and outlet, $\Gamma_O = \Gamma_{\text{in}} \cup \Gamma_{\text{out}}$. When the shape control is confined to the channel walls $\subset \Gamma_D$, i.e. $\Gamma_O \cap \Gamma_D = \emptyset$, only local objective functional variations have to be evaluated

$$\begin{aligned} \delta_L J_{\Gamma} &= \int_{\Gamma_O} \delta_L j_{\Gamma} d\Gamma \\ \text{with} \quad \delta_L j_{\Gamma} &= -n_j \delta U_j \left(p + \frac{\rho}{2} U_i^2 \right) - n_j U_j (\delta p + \rho U_i \delta U_i) \\ &= -\delta U_i \left(\frac{\rho}{2} U_j^2 n_i + p n_i + \rho U_j U_i n_j \right) - \delta p (U_j n_j). \end{aligned} \quad (5.12)$$

A typical representative of scenario (b) is the hydrodynamic force criterion (4.34)

$$J_{\Gamma} = \int_{\Gamma_O} j_{\Gamma} d\Gamma \quad \text{with} \quad j_{\Gamma} = -\pi_{ij} n_j d_i^*. \quad (5.13)$$

The objective functional is declared on the body surface that is subject to the shape design, i.e. the objective functional surface and the design surface coincide fully or in parts, $\Gamma_O \cap \Gamma_D \neq \emptyset$. In a linear approach, three different contributions to the total change of the objective functional can be identified: (i) a *local change* $\delta_L J_{\Gamma}$ due to a local change of the flow ($\delta \underline{U}$, δp); (ii) a *convective shift* $\delta_C J_{\Gamma}$ accounting for a boundary-normal position shift $\delta \underline{n} = \underline{n} \cdot \delta \underline{x}$; and (iii) a *geometric change* $\delta_G J_{\Gamma}$ due to a change of the surface area and orientation:

$$\delta_{LCG} J = (\delta_L + \delta_C + \delta_G) J_{\Gamma}. \quad (5.14)$$

The local variation of the objective functional is

$$\delta_L J_{\Gamma} = \int_{\Gamma_O} \delta_L j_{\Gamma} d\Gamma \quad \text{with} \quad \delta_L j_{\Gamma} = (n_i \delta p - n_j \delta \tau_{ij}) d_i^*.$$

Convective and geometric variations of the objective functional are discussed in the following based on the force criterion (5.13).

Convective Derivative

For contour-normal perturbations, $\delta n = \underline{n} \cdot \delta \underline{x}$, the hydrodynamic stress tensor in the integrand of the force criterion (5.13) can be developed to the new position:

$$\delta_C J = \int_{\Gamma_O \cap \Gamma_D} \delta_C j_\Gamma \, d\Gamma \quad (5.15a)$$

$$= \int_{\Gamma_O \cap \Gamma_D} \delta n \frac{\partial}{\partial n} [n_i p - n_j \tau_{ij}] d_i^* \, d\Gamma. \quad (5.15b)$$

Eqn. (5.15b) can be rewritten as

$$\delta_C J = \int_{\Gamma_O \cap \Gamma_D} \delta n \, G_C \, d\Gamma, \quad (5.16)$$

with the convective derivative of the objective functional

$$G_C = \frac{\partial}{\partial n} [n_i p - n_j \tau_{ij}] d_i^* \quad \text{on } \Gamma_O \cap \Gamma_D. \quad (5.17)$$

For no-slip walls, the convective variation of the force objective is zero according to first-order boundary-layer theory, since both pressure and shear stress are assumed to be constant in the boundary-normal direction:

$$0 \approx \frac{\partial \tau^{nt}}{\partial n} \approx \frac{\partial p}{\partial n} \quad \text{on } \Gamma_D. \quad (5.18)$$

Geometric Derivative

The geometric variation of the objective functional accounts for variations of the surface area and orientation. It can be evaluated according to *differential geometry*, e.g. Hill [57], Dems and Mróz [18], Sokolowski and Zolésio [145]. Schmidt [137] calculates the geometric variation via a surface-divergence formulation

$$\delta_G J = d_i^* \delta_G \int_{\Gamma_O \cap \Gamma_D} \pi_{ij} n_j \, d\Gamma \quad (5.19a)$$

$$= d_i^* \int_{\Gamma_O \cap \Gamma_D} \pi_{ij} \delta(n_j) \, d\Gamma \quad (5.19b)$$

$$= \int_{\Gamma_O \cap \Gamma_D} \delta n \frac{\partial}{\partial x_j^\Gamma} (\pi_{ij} d_i^*) \, d\Gamma. \quad (5.19c)$$

The *surface divergence*, $\partial(\cdot)_j / \partial x_j^\Gamma$, is the projection of the corresponding Cartesian operator onto the boundary surface, cf. Appendix C. According to Schmidt [137], Eqn. (5.19c) can be rewritten in terms of the geometric derivative G_G

$$\delta_G J = \int_{\Gamma_O \cap \Gamma_D} \delta n \, G_G \, d\Gamma \quad (5.20)$$

with

$$G_G = \frac{\partial}{\partial x_j^\Gamma} (\pi_{ij} d_i^*) \quad \text{on } \Gamma_O \cap \Gamma_D. \quad (5.21)$$

A Note on Local, Geometric and Convective Variations

The convective and the geometric variations of the objective functional $\delta_{CG}J$ are “local” in character (not to be confused with the term “local variation” motivated by the concept of material derivative). It means that it is possible to locally evaluate the influence of a shape perturbation on the objective functional. From a computational point of view this is both convenient and cost-efficient. The opposite is the case for the local variations of the objective functional $\delta_L J$. The flow perturbation $(\delta \underline{U}, \delta p)$ is propagated through the domain by the variation of the Navier–Stokes equations derived in the next section. The elliptic character of the incompressible Navier–Stokes equations implies that a perturbation applied at any position in the flow domain causes a global change in the flow field. Assuming that the numerical effort spent for solving the non-linear and the linearised Navier–Stokes equations is comparable, it is prohibitively expensive to calculate the sensitivity derivatives with respect to many degrees of freedom in direct mode; the adjoint approach described in Chapter 6 is more efficient in that case.

5.3. Variation of the Navier–Stokes Equations

Following the concept of material derivative, the governing flow equations (4.9) and (4.10) have to be satisfied for a slightly modified domain, resulting in the variation of the field equations and boundary conditions.

Field Equations

A linearisation of the Navier–Stokes equations about a steady-state solution leads to

$$0 = \left(\delta + \delta x_j \frac{\partial}{\partial x_j} \right) R_i \quad \text{in } \Omega, \quad (5.22)$$

and

$$0 = \left(\delta + \delta x_j \frac{\partial}{\partial x_j} \right) Q \quad \text{in } \Omega. \quad (5.23)$$

Thanks to the material derivative approach, the variation of the Navier–Stokes equations is solved on the original domain, i.e. the non-deformed computational mesh. Since the primal Navier–Stokes equations are satisfied on the old grid, $R_i = Q = 0$, the respective gradients of the primal residuals vanish

$$0 = \delta x_k \frac{\partial R_i}{\partial x_k} = \delta x_k \frac{\partial Q}{\partial x_k} \quad \text{in } \Omega. \quad (5.24)$$

Therefore, Eqns. (5.22) and (5.23) reduce to

$$0 = \delta R_i = \delta Q \quad \text{in } \Omega. \quad (5.25)$$

The expanded form of Eqn. (5.25) is provided in Appendix A.1. Since the convective flow variation inside the domain does not need to be evaluated, a variation of the interior grid

is not necessary. For incompressible flow the density shows no variation. Also the body-force density \underline{f} does not undergo variations. A possible variation of the eddy viscosity with respect to the shape is not taken into consideration, neither in terms of a convective nor a local variation. The so-called frozen-turbulence assumption is common practice for complex applications [66, 149, 151, 110, 22, 116].

The variation of the boundary conditions for the Navier–Stokes equations are presented in the next section.

Boundary Conditions on Γ_D

The shape design problem is “boundary-controlled”. The shape variation is applied to the design surfaces Γ_D , which are subject to both convective and local variations $(\delta + \delta n \partial/\partial n)\phi$. The variation of the boundary conditions is developed so that the original boundary conditions are satisfied for the new boundary \underline{x}_1 in a linear sense.

A **Dirichlet** boundary condition postulated for the old flow $\phi^{(0)}$ at the old (non-perturbed) boundary position \underline{x}_0

$$\mathcal{D} = \phi^{(0)}|_{\underline{x}_0}, \quad (5.26)$$

also needs to be satisfied by the new flow $\phi^{(1)}$ on the new geometry \underline{x}_1 . For boundary-normal shape perturbations, $\delta n = \underline{n} \cdot \delta \underline{x}$, the boundary condition can be developed to second-order accuracy about the old position

$$\mathcal{D} = \phi^{(1)}|_{\underline{x}_1} \approx \left[\phi^{(0)} + \delta\phi + \delta n \frac{\partial\phi^{(0)}}{\partial n} \right]_{\underline{x}_0}. \quad (5.27)$$

Subtracting (5.27) from (5.26) yields

$$\delta\phi|_{\underline{x}_0} \approx -\delta n \frac{\partial\phi^{(0)}}{\partial n}|_{\underline{x}_0} \quad \text{on } \Gamma_D. \quad (5.28)$$

The subsequent analysis is carried out on the old grid. The pointers \underline{x}_0 and \underline{x}_1 are left out for the sake of brevity. ϕ refers to the old flow unless declared differently.

A **Neumann** boundary condition

$$\mathcal{N} = \frac{\partial\phi}{\partial n} \quad \text{on } \Gamma_D, \quad (5.29)$$

can be developed for a boundary-normal perturbation δn on the old grid according to Eqn. (5.6):

$$\frac{\partial\delta\phi}{\partial n} = -\delta n \frac{\partial}{\partial n} \left(\frac{\partial\phi}{\partial n} \right) \quad \text{on } \Gamma_D. \quad (5.30)$$

A **Robin**-type boundary condition

$$\mathcal{R} = \phi + \alpha \frac{\partial\phi}{\partial n} \quad \text{on } \Gamma_D, \quad (5.31)$$

subject to a boundary-normal perturbation δn reads

$$\delta\phi + \alpha \frac{\partial\delta\phi}{\partial n} = -\delta n \frac{\partial\phi}{\partial n} - \alpha \delta n \frac{\partial^2\phi}{\partial n^2} \quad \text{on } \Gamma_D. \quad (5.32)$$

The variation of the boundary conditions for the velocity vector is derived on the basis of a local, boundary-oriented system of coordinates (n, t, s) . The design surfaces Γ_D are confined to solid walls with a no-slip boundary condition, which can be linearly developed according to Eqn. (5.28):

$$\delta U_i = -\delta n \frac{\partial U_i}{\partial n} \quad \text{on } \Gamma_D. \quad (5.33)$$

By virtue of Eqn. (4.27), Eqn. (5.33) simplifies to

$$\delta U^t = -\delta n \frac{\partial U^t}{\partial n} \quad \text{and} \quad 0 = \delta U^n = \delta U^s$$

or

$$\delta U_i = -\delta n \frac{\partial U^t}{\partial n} t_i = -\delta n \frac{\partial U_i^t}{\partial n}. \quad (5.34)$$

Boundary Conditions on $\Gamma \setminus \Gamma_D$

The boundaries that are not subject to shape variations, $\Gamma \setminus \Gamma_D$, experience local variations alone.

Hence, the variation of a **Dirichlet** boundary condition is zero by definition

$$\mathcal{D} = \phi \quad \rightsquigarrow \quad 0 = \delta\phi \quad \text{on } \Gamma \setminus \Gamma_D. \quad (5.35)$$

For **Neumann** boundary conditions the prescribed gradient remains unchanged

$$\mathcal{N} = \frac{\partial\phi}{\partial n} \quad \rightsquigarrow \quad 0 = \frac{\partial\delta\phi}{\partial n} \quad \text{on } \Gamma \setminus \Gamma_D. \quad (5.36)$$

A **Robin**-type boundary condition implies

$$\mathcal{R} = \phi + \alpha \frac{\partial\phi}{\partial n} \quad \rightsquigarrow \quad 0 = \delta\phi + \alpha \frac{\partial\delta\phi}{\partial n} \quad \text{on } \Gamma \setminus \Gamma_D. \quad (5.37)$$

The variation of the boundary conditions derived above is compiled for a scalar variable $\delta\phi$ and for the variation of the Navier–Stokes problem $(\delta\mathbf{U}, \delta p)$ in Tables 5.1 and 5.2, respectively.

5.4. Direct Evaluation of Constrained Derivatives

A direct evaluation of the objective functional sensitivity according to the Jacobian calculus, cf. Section 2.3, requires the following steps:

- (i) Calculate the reference solution, $(\underline{U}, p)[\underline{\beta}]$, from the non-linear equations of state.

Table 5.1.: Variation of the boundary conditions for a scalar ϕ formulated on the *old* boundary

boundary	ϕ	$\delta\phi$
Dirichlet $\subset \Gamma_D$	$\mathcal{D} = \phi$	$\delta\phi = -\delta n \phi^{,n}$
Dirichlet $\not\subset \Gamma_D$	$\mathcal{D} = \phi$	$\delta\phi = 0$
Neumann $\subset \Gamma_D$	$\mathcal{N} = \phi^{,n}$	$\delta\phi^{,n} = -\delta n \phi^{,nn}$
Neumann $\not\subset \Gamma_D$	$\mathcal{N} = \phi^{,n}$	$\delta\phi^{,n} = 0$
Robin $\subset \Gamma_D$	$\mathcal{R} = \phi + \alpha \phi^{,n}$	$\delta\phi + \alpha \delta\phi^{,n} = -\delta n (\phi^{,n} + \alpha \phi^{,nn})$
Robin $\not\subset \Gamma_D$	$\mathcal{R} = \phi + \alpha \phi^{,n}$	$\delta\phi + \alpha \delta\phi^{,n} = 0$

Table 5.2.: Variation of the boundary conditions for the Navier–Stokes equations formulated on the *old* boundary

boundary	$\delta\mathcal{U}$	δp
no-slip wall $\subset \Gamma_D$	$\delta\mathcal{U} = -\delta n (\underline{n} \cdot \underline{\nabla} \mathcal{U}^t)$	$\delta p = -\delta n p^{,nn} = 0$
no-slip wall $\not\subset \Gamma_D$	$\delta\mathcal{U} = \underline{0}$	$\delta p^{,n} = 0$
symmetry	$\delta U^n = \delta U^{t,n} = 0$	$\delta p^{,n} = 0$
inflow	$\delta\mathcal{U} = \underline{0}$	$\delta p^{,n} = 0$
pressure outflow	$\underline{n} \cdot \underline{\nabla}(\delta\mathcal{U}) = \underline{0}$	$\delta p = 0$

- (ii) For $i = 1, \dots, m$, perturb the shape by $\delta\beta_i$.
- (iii) Evaluate the distribution of the boundary-normal offset $\delta n[\delta\beta_i]$ over the design surface Γ_D .
- (iv) Solve Eqns. (5.25) for the local flow variation, $(\delta\mathbf{U}, \delta p)[\delta\beta_i]$, subject to the variation of the boundary conditions (5.34).
- (v) Evaluate the variation of the objective functional, $\delta_{LGG}J[\delta\beta_i]$, according to Section 5.2.
- (vi) $i \leftarrow i + 1$ and go to (ii).

In order to compute the complete sensitivity vector $\delta J/\delta\beta_i$, the steps (ii) to (v) have to be carried out m times. Assuming that step (iv) dominates the computational effort, the costs of an evaluation of the full gradient vector scale with the number of design variables m . In the face of many parameters, the sensitivity analysis should be conducted in the adjoint or Lagrange manner as outlined in Section 2.3. The adjoint calculus is applied to the Navier–Stokes problem on a PDE level in the next chapter.

6. Continuous Adjoint Navier–Stokes Concept

“Additionally, the design method, which is greatly accelerated by the use of control theory, has been further enhanced by the use of a new continuous adjoint method, which reduce the volume integral part of the adjoint gradient formula to a surface integral [64], thus eliminating the dependence of the gradient formulas on the mesh perturbation.” Jameson [63]

The continuous adjoint Navier–Stokes problem is devised from the augmented objective functional or Lagrangian as described in Section 6.1. Its material derivative is re-organised and separated into two parts, either

- (i) depending on the variation of the state $(\delta \underline{U}, \delta p)$; or
- (ii) on the control δn , which is applied to the boundary Γ_D of the flow domain.

The former (i) defines the adjoint Navier–Stokes equations and boundary conditions, Section 6.2, whereas the adjoint sensitivity equation is obtained from (ii) as described in Section 6.3. Similar derivations of the adjoint Navier–Stokes equations for hydrodynamic shape optimisation are, for instance, presented by Soto and Löhner [147, 148, 149] and Othmer [110].

6.1. Augmented Objective Functional

The optimisation problem subject to the incompressible Navier–Stokes equations can be turned into an unconstrained problem via the Lagrangian calculus as shown in Section 2.3. The objective functional is extended by the domain integral of the Navier–Stokes constraints weighted by the so-called Lagrange multipliers or adjoint variables (\hat{U}_i, \hat{p}) :

$$L = J + \int d\Omega \left[\hat{U}_i R_i + \hat{p} Q \right]. \quad (6.1)$$

The adjoint variables are field variables like their primal counterparts. Given that the Navier–Stokes equations are satisfied for the investigated design state or reference case, expression (6.1) exactly meets the value of the objective functional J . The first variation of the extended objective functional reads

$$\delta_{LCG} L = \delta_{LCG} J + \int_{\Omega} d\Omega \left[\hat{U}_i \delta R_i + \hat{p} \delta Q \right]. \quad (6.2)$$

Given that the variation of the Navier–Stokes equations (5.25) is satisfied, $0 = \delta Q = \delta R_i$, the variation of the Lagrangian $\delta_{LCG} L$ equals the variation of the objective functional

$\delta_{LCG}J$. The subsequent analysis is confined to the local variation of the Lagrangian, $\delta_L L$ in order to derive the adjoint PDE. The remaining contributions, $\delta_{CG}L = \delta_{CG}J$, have been discussed in Section 5.2. Expansion of the local variation terms of Eqn. (6.2) yields:

$$\delta_L L = \int_{\Gamma_o} \delta_L j_\Gamma \, d\Gamma + \int_{\Omega_o} \delta_L j_\Omega \, d\Omega \quad (6.3a)$$

$$+ \int d\Omega \left[\hat{U}_i \left(\rho \delta U_j \frac{\partial U_i}{\partial x_j} + \rho U_j \frac{\partial \delta U_i}{\partial x_j} \right) \right] \quad (6.3b)$$

$$+ \frac{\partial}{\partial x_j} \left[\delta_{ij} \delta p - \mu_{\text{eff}} \left(\frac{\partial \delta U_i}{\partial x_j} + \frac{\partial \delta U_j}{\partial x_i} \right) \right] - \hat{p} \left(\frac{\partial \delta U_i}{\partial x_i} \right). \quad (6.3c)$$

The basic idea is to choose the adjoint multipliers (\hat{U}_i, \hat{p}) such that any contribution to Eqn. (6.3) depending on any admissible local flow variation $(\delta U_i, \delta p)$ is eliminated. This particular choice (\hat{U}_i, \hat{p}) requires to satisfy the adjoint Navier–Stokes equations, which are specific to the objective functional. After solving the adjoint problem associated with the objective functional of interest, its derivative is evaluated from the remaining terms of Eqn. (6.3) in the so-called *adjoint post-processing* described in Section 6.3.

6.2. Adjoint Navier–Stokes Equations

The subsequent expression is obtained from Eqn. (6.3) via integration by parts, cf. Appendix A.2. It constitutes the adjoint field equations and boundary conditions:

$$\delta_L L = \oint d\Gamma_j \left[\left(\rho \delta U_i U_j + \delta p \delta_{ij} - \mu_{\text{eff}} \left[\frac{\partial \delta U_i}{\partial x_j} + \frac{\partial \delta U_j}{\partial x_i} \right] \right) \hat{U}_i \right] \quad (6.4a)$$

$$+ \left(-\hat{p} \delta_{ij} + 2 \mu_{\text{eff}} \hat{S}_{ji} \right) \delta U_i \quad (6.4b)$$

$$+ \int d\Omega \left[\left(\rho \hat{U}_j \frac{\partial U_j}{\partial x_i} - \rho U_j \frac{\partial \hat{U}_i}{\partial x_j} - \frac{\partial}{\partial x_j} \left[2 \mu_{\text{eff}} \hat{S}_{ij} - \hat{p} \delta_{ij} \right] \right) \delta U_i - \frac{\partial \hat{U}_j}{\partial x_j} \delta p \right] \quad (6.4c)$$

$$+ \int_{\Omega_o} \delta_L j_\Omega \, d\Omega + \int_{\Gamma_o} \delta_L j_\Gamma \, d\Gamma \quad (6.4d)$$

with $\hat{\underline{S}} = (\underline{\nabla} \hat{\underline{U}} + [\underline{\nabla} \hat{\underline{U}}]^T)/2$. Note that the leading term in (6.4c), called “advection” in this work, has not been integrated by parts. This advection treatment is referred to as ADV1 and is used for the subsequent analysis. It is also followed by Hinze [59], Carnarius et al. [8, 9]. An alternative formulation, ADV2, is obtained via integration by parts as described in Appendix B. This strategy was, among others pursued by Soto and Löhner [149], Othmer [110] and Zymaris et al. [170, 171]. Soto and Löhner [149] also suggest to neglect the advection term (ADV0) so that the adjoint PDE are similar to the primal ones, which facilitates the numerical solution.

Adjoint Field Equations

The adjoint field equations eliminate the volume contributions to (6.4c) and (6.4d) for arbitrary, admissible choices $(\delta U_i, \delta p)$. The adjoint equations of momentum and continuity are

$$\begin{cases} -\rho U_j \frac{\partial \hat{U}_i}{\partial x_j} = \frac{\partial}{\partial x_j} \left(2\mu_{\text{eff}} \hat{S}_{ij} - \hat{p} \delta_{ij} \right) - \rho \hat{U}_j \frac{\partial U_j}{\partial x_i} - \frac{\partial j_\Omega}{\partial U_i} & \text{in } \Omega_O \\ -\rho U_j \frac{\partial \hat{U}_i}{\partial x_j} = \frac{\partial}{\partial x_j} \left(2\mu_{\text{eff}} \hat{S}_{ij} - \hat{p} \delta_{ij} \right) - \rho \hat{U}_j \frac{\partial U_j}{\partial x_i} & \text{in } \Omega \setminus \Omega_O \end{cases} \quad (6.5)$$

and

$$\begin{cases} \frac{\partial \hat{U}_i}{\partial x_i} = \frac{\partial j_\Omega}{\partial p} & \text{in } \Omega_O \\ \frac{\partial \hat{U}_i}{\partial x_i} = 0 & \text{in } \Omega \setminus \Omega_O. \end{cases} \quad (6.6)$$

The source terms in Ω_O originate from the linear variation of the integrand of the considered objective functional

$$\delta_L j_\Omega = \delta p \frac{\partial j_\Omega}{\partial p} + \delta U_i \frac{\partial j_\Omega}{\partial U_i}. \quad (6.7)$$

Formulation (6.7) is only valid for algebraic terms and requires a reformulation if the volume-declared integrand of the objective functional contains differential expressions.

Adjoint Boundary Conditions

In order to derive the adjoint boundary conditions, the boundary types used in the primal code have to be redivided depending on their contribution to the considered objective functional. The design surfaces Γ_D , which may undergo spatial perturbations, are confined to no-slip wall boundaries in this work. The remaining boundaries, $\Gamma \setminus \Gamma_D$, are not affected by the shape control, i.e. they keep their original position and experience local flow variations alone.

Inflow boundaries and **no-slip walls** on $\Gamma \setminus \Gamma_D$ have a Dirichlet-type boundary condition for the velocity

$$\mathcal{D}_i = U_i \quad \text{on } \Gamma_{\text{in}} \text{ and } \Gamma_{\text{wall}} \not\subset \Gamma_D. \quad (6.8)$$

The local variation is zero by definition

$$0 = \delta U_i \quad \text{on } \Gamma_{\text{in}} \text{ and } \Gamma_{\text{wall}} \not\subset \Gamma_D. \quad (6.9)$$

The pressure is extrapolated to the boundaries. If no contributions to the objective functional are declared over wall or inlet boundaries, the remaining terms of the boundary integral (6.4a/6.4b)

$$0 = -\mu_{\text{eff}} \left(\frac{\partial \delta U_i}{\partial n} + \frac{\partial \delta U^n}{\partial x_i} \right) \hat{U}_i + \delta p \hat{U}^n \quad \text{on } \Gamma_{\text{in}} \not\subset \Gamma_O \text{ and } \Gamma_{\text{wall}} \not\subset (\Gamma_D \cup \Gamma_O) \quad (6.10)$$

define the adjoint boundary condition

$$0 = \hat{U}_i \quad \text{on } \Gamma_{\text{in}} \not\subset \Gamma_O \text{ and } \Gamma_{\text{wall}} \not\subset (\Gamma_D \cup \Gamma_O). \quad (6.11)$$

In case the considered boundary wall carries an objective functional, such as the force criterion (4.34/5.14) [e.g. 147, 148, 8], the non-zero boundary terms of (6.4a/6.4b/6.4d) depending on the local variation of the flow ($\delta U_i, \delta p$) are

$$\left[\delta p \delta_{ij} - \mu_{\text{eff}} \left(\frac{\partial \delta U_i}{\partial x_j} + \frac{\partial \delta U_j}{\partial x_i} \right) \right] (\hat{U}_i + d_i^*) = 0 \quad \text{on } \Gamma_{\text{wall}} \subset (\Gamma_O \setminus \Gamma_D), \quad (6.12)$$

requiring

$$\hat{U}_i = -d_i^* \quad \text{on } \Gamma_{\text{wall}} \subset (\Gamma_O \setminus \Gamma_D). \quad (6.13)$$

If the power-loss criterion (4.38/5.12) is declared on the inlet [e.g. 110, 170, 171], the adjoint boundary condition is deduced from the boundary terms

$$n_j \left[\delta p \delta_{ij} - \mu_{\text{eff}} \left(\frac{\partial \delta U_i}{\partial x_j} + \frac{\partial \delta U_j}{\partial x_i} \right) \right] \hat{U}_i - n_j U_j \delta p = 0 \quad \text{on } \Gamma_{\text{in}} \subset \Gamma_O \quad (6.14)$$

which, in conjunction with (a) $0 = T_{ij} U_i|_{\text{in}}$ and (b) $0 = \partial \delta U^n / \partial n|_{\text{in}}$, implies

$$\hat{U}_i = U_i \quad \text{on } \Gamma_{\text{in}} \subset \Gamma_O. \quad (6.15)$$

Condition (a) is satisfied for the internal flow problems considered in this study; (b) is feasible given that the inlet flow is developed. It is interesting to note that (a) and (b) can be abandoned when the underlined term in the power-loss criterion (4.37) is taken into account.

The adjoint **symmetry boundary condition** is derived from the boundary terms (6.4a/6.4b) in conjunction with

$$0 = \frac{\partial U^t}{\partial n} = \frac{\partial U^s}{\partial n} = \frac{\partial \delta U^t}{\partial n} = \frac{\partial \delta U^s}{\partial n} \quad \text{on } \Gamma_{\text{sym}}$$

and

$$0 = U^n = \delta U^n \quad \text{on } \Gamma_{\text{sym}}. \quad (6.16)$$

It leads to

$$\begin{aligned} 0 &= \delta p \hat{U}^n - \mu_{\text{eff}} \hat{U}_i \left(\frac{\partial \delta U_i}{\partial n} + \frac{\partial \delta U^n}{\partial x_i} \right) + \mu_{\text{eff}} \delta U_i \left(\frac{\partial \hat{U}_i}{\partial n} + \frac{\partial \hat{U}^n}{\partial x_i} \right) \\ &= \left(\delta p - 2\mu_{\text{eff}} \frac{\partial \delta U^n}{\partial n} \right) \hat{U}^n + \mu_{\text{eff}} (\delta U^t \underline{\hat{t}}^\delta + \delta U^s \underline{\hat{s}}^\delta) \cdot \left(\frac{\partial \hat{U}^t}{\partial n} \underline{\hat{t}} + \frac{\partial \hat{U}^s}{\partial n} \underline{\hat{s}} \right) \quad \text{on } \Gamma_{\text{sym}} \end{aligned} \quad (6.17)$$

with the boundary unit vectors on Γ_{sym}

$$\begin{aligned} \underline{\hat{t}} &= \underline{U}/|\underline{U}|; & \underline{\hat{s}} &= \underline{n} \times \underline{\hat{t}}; \\ \underline{\hat{t}} &= \underline{\hat{U}}/|\underline{\hat{U}}|; & \underline{\hat{s}} &= \underline{n} \times \underline{\hat{t}}. \\ \underline{\hat{t}}^\delta &= \delta \underline{U}/|\delta \underline{U}|; & \underline{\hat{s}}^\delta &= \underline{n} \times \underline{\hat{t}}^\delta. \end{aligned} \quad (6.18)$$

It implies

$$0 = \hat{U}^n = \frac{\partial \hat{U}^t}{\partial n} = \frac{\partial \hat{U}^s}{\partial n} \quad \text{on } \Gamma_{\text{sym}}. \quad (6.19)$$

A **pressure outlet** boundary condition states

$$\mathcal{D} = p \quad \rightsquigarrow \quad 0 = \delta p \quad \text{on } \Gamma_{\text{out}}. \quad (6.20)$$

The velocity is extrapolated to the outlet with a zero-gradient, viz.

$$0 = \frac{\partial U_i}{\partial n} = \frac{\partial \delta U_i}{\partial n} \quad \text{on } \Gamma_{\text{out}}. \quad (6.21)$$

The remaining boundary terms in the extended objective functional (6.4a/6.4b) are

$$0 = \delta U_i \left[\rho U^n \hat{U}_i + \mu_{\text{eff}} \left(\frac{\partial \hat{U}_i}{\partial n} + \frac{\partial \hat{U}^n}{\partial x_i} \right) - \hat{p} n_i \right] - \mu_{\text{eff}} \frac{\partial \delta U^n}{\partial x_i} \hat{U}_i \quad \text{on } \Gamma_{\text{out}} \not\subset \Gamma_O. \quad (6.22)$$

Assuming $\underline{0} = \underline{\nabla} \nu_{\text{eff}}$ in the vicinity of the outlet, e.g. Othmer [110], leads to

$$0 = \delta U_i \left[\rho U^n \hat{U}_i + \mu_{\text{eff}} \frac{\partial \hat{U}_i}{\partial n} - \hat{p} n_i \right] \quad \text{on } \Gamma_{\text{out}} \not\subset \Gamma_O. \quad (6.23)$$

For arbitrary $\delta U_i|_{\text{out}}$, Eqn. (6.23) implies

$$\hat{p} n_i = \rho U^n \hat{U}_i + \mu_{\text{eff}} \frac{\partial \hat{U}_i}{\partial n} \quad \text{on } \Gamma_{\text{out}} \not\subset \Gamma_O. \quad (6.24)$$

In a local system of boundary coordinates (6.18), the following boundary conditions are obtained on the pressure outlet

$$\begin{aligned} \hat{p} &= \rho U^n \hat{U}^n + \mu_{\text{eff}} \frac{\partial \hat{U}^n}{\partial n}, \\ 0 &= \rho U^n \hat{U}^t + \mu_{\text{eff}} \frac{\partial \hat{U}^t}{\partial n}, \\ 0 &= \rho U^n \hat{U}^s + \mu_{\text{eff}} \frac{\partial \hat{U}^s}{\partial n} \quad \text{on } \Gamma_{\text{out}} \not\subset \Gamma_O. \end{aligned} \quad (6.25)$$

In case the power-loss criterion (4.38/5.12) is declared on the outlet with $p|_{\text{out}} = \delta p|_{\text{out}} = 0$ [e.g. 110, 170, 171], the adjoint boundary condition is defined by the non-zero boundary terms of (6.4a/6.4b/6.4d)

$$0 = \delta U_i \left[\rho U^n \hat{U}_i + \mu_{\text{eff}} \frac{\partial \hat{U}_i}{\partial n} - \hat{p} n_i - \frac{\rho}{2} n_i U_j^2 - \rho U^n U_i \right] \quad \text{on } \Gamma_{\text{out}} \subset \Gamma_O, \quad (6.26)$$

i.e.

$$\hat{p} n_i = \rho U^n \hat{U}_i + \mu_{\text{eff}} \frac{\partial \hat{U}_i}{\partial n} - \frac{\rho}{2} n_i U_j^2 - \rho U^n U_i \quad \text{on } \Gamma_{\text{out}} \subset \Gamma_O. \quad (6.27)$$

The adjoint boundary conditions for **wall boundaries subject to shape variations**, Γ_D , need a closer look. The no-slip type Dirichlet boundary condition implies

$$0 = U_i \quad \text{on} \quad \Gamma_{\text{wall}}. \quad (6.28)$$

Due to the position shift $\delta n = \underline{n} \cdot \delta \underline{x}$ both local and convective variations are present. According to Eqn. (5.28), the Dirichlet boundary condition can be linearly developed to the new position, i.e.

$$\delta U_i = -\delta n \frac{\partial U_i}{\partial n} \quad \text{on} \quad \Gamma_{\text{wall}} \subset \Gamma_D. \quad (6.29)$$

When a force criterion (4.34/5.14) is considered on $\Gamma_D \cap \Gamma_O$, the specific contributions (6.4d) must be taken into account together with the non-zero boundary terms of (6.4a/6.4b)

$$n_j \left[\left(-\hat{p} \delta_{ij} + 2 \mu_{\text{eff}} \hat{S}_{ij} \right) \delta U_i \right] \quad (6.30a)$$

$$+ \left(\rho \delta U_i U_j + \delta p \delta_{ij} - \mu_{\text{eff}} \left[\frac{\partial \delta U_i}{\partial x_j} + \frac{\partial \delta U_j}{\partial x_i} \right] \right) \hat{U}_i \Big|_{\Gamma_{\text{wall}} \subset \Gamma_D} \quad (6.30b)$$

$$+ n_j \left[\left(\delta p \delta_{ij} - \mu_{\text{eff}} \left[\frac{\partial \delta U_i}{\partial x_j} + \frac{\partial \delta U_j}{\partial x_i} \right] \right) d_i^* \right] \Big|_{\Gamma_{\text{wall}} \subset (\Gamma_D \cap \Gamma_O)}. \quad (6.30c)$$

By virtue of Eqn. (6.29), the terms (6.30a) can be rewritten in terms of the control δn . These contributions are considered in the adjoint gradient equation in Section 6.3. The remaining boundary terms (6.30b) and (6.30c) are to be eliminated by the adjoint boundary conditions. With the primal no-slip Dirichlet boundary condition (6.28), the only remaining boundary terms, depending on local flow variations ($\delta U_i, \delta p$), are

$$\left[\delta p \delta_{ij} - \mu_{\text{eff}} \left(\frac{\partial \delta U_i}{\partial x_j} + \frac{\partial \delta U_j}{\partial x_i} \right) \right] \hat{U}_i = 0 \quad \text{on} \quad \Gamma_{\text{wall}} \subset (\Gamma_D \setminus \Gamma_O) \quad (6.31)$$

and

$$\left[\delta p \delta_{ij} - \mu_{\text{eff}} \left(\frac{\partial \delta U_i}{\partial x_j} + \frac{\partial \delta U_j}{\partial x_i} \right) \right] (\hat{U}_i + d_i^*) = 0 \quad \text{on} \quad \Gamma_{\text{wall}} \subset (\Gamma_D \cap \Gamma_O). \quad (6.32)$$

These terms are eliminated by the adjoint boundary conditions

$$\hat{U}_i = 0 \quad \text{on} \quad \Gamma_{\text{wall}} \subset (\Gamma_D \setminus \Gamma_O) \quad (6.33)$$

and

$$\hat{U}_i = -d_i^* \quad \text{on} \quad \Gamma_{\text{wall}} \subset (\Gamma_D \cap \Gamma_O). \quad (6.34)$$

The adjoint boundary conditions derived above are compiled in Table 6.1.

Table 6.1.: Boundary conditions for the adjoint Navier–Stokes equations (ADV1)

boundary	objective	\hat{U}	\hat{p}
no-slip wall $\not\subset \Gamma_O$	–	$\hat{U} = \underline{0}$	$\hat{p}^n = 0$
no-slip wall $\subset \Gamma_O$	force	$\hat{U} = -\underline{d}^*$	$\hat{p}^n = 0$
inflow $\not\subset \Gamma_O$	–	$\hat{U} = \underline{0}$	$\hat{p}^n = 0$
inflow $\subset \Gamma_O$	power loss	$\hat{U} = \underline{U}$	$\hat{p}^n = 0$
symmetry	–	$\hat{U}^n = \hat{U}^{t,n} = 0$	$\hat{p}^n = 0$
pressure outflow $\not\subset \Gamma_O$	–	$\rho U^n \hat{U}_i + \mu_{\text{eff}} \frac{\partial \hat{U}_i}{\partial n} = \hat{p} n_i$	
pressure outflow $\subset \Gamma_O$	power loss	$\rho U^n \hat{U}_i + \mu_{\text{eff}} \frac{\partial \hat{U}_i}{\partial n} - \frac{\rho}{2} U_j^2 n_i - \rho U^n U_i = \hat{p} n_i$	

6.3. Adjoint Gradient Equation

Having eliminated the local flow variations in the augmented objective functional by satisfying the adjoint Navier–Stokes equations and boundary conditions, the remaining terms of Eqn. (6.4) state the local portion of the objective functional derivative [e.g. 149, 110, 170, 171]:

$$\delta_L L = \delta_L J = \int_{\Gamma_D} d\Gamma_j \left[\left(-\hat{p} \delta_{ij} + 2 \mu_{\text{eff}} \hat{S}_{ij} \right) \delta U_i \right]. \quad (6.35)$$

These are the contributions (6.30a), which have been transferred to the sensitivity equation. Substituting the linearised boundary condition on Γ_D , Eqn. (6.29), into Eqn. (6.35) yields:

$$\begin{aligned} \delta_L L &= \int_{\Gamma_D} d\Gamma_j \left[\left(\hat{p} \delta_{ij} - 2 \mu_{\text{eff}} \hat{S}_{ij} \right) \delta n \frac{\partial U_i}{\partial n} \right] \\ &= \int_{\Gamma_D} d\Gamma \left[\delta n \left(\hat{p} \frac{\partial U^n}{\partial n} - 2 \mu_{\text{eff}} \hat{S}_{ij} \frac{\partial U_i}{\partial n} \right) \right] \\ &\approx \int_{\Gamma_D} d\Gamma \left[-\delta n \mu_{\text{eff}} \frac{\partial \hat{U}^t}{\partial n} \frac{\partial U^t}{\partial n} \hat{t}_i t_i \right]. \end{aligned} \quad (6.36)$$

The local wall coordinate systems for flow $(\underline{n}, \underline{t}, \underline{s})$ and adjoints $(\underline{n}, \underline{\hat{t}}, \underline{\hat{s}})$ are defined through the outward boundary unit vector \underline{n} and the in-plane components

$$\begin{aligned} \underline{t} &= \underline{n} \cdot \underline{S} \cdot \underline{T} / |\underline{n} \cdot \underline{S} \cdot \underline{T}|; & \underline{s} &= \underline{n} \times \underline{t}; \\ \underline{\hat{t}} &= \underline{n} \cdot \underline{\hat{S}} \cdot \underline{T} / |\underline{n} \cdot \underline{\hat{S}} \cdot \underline{T}|; & \underline{\hat{s}} &= \underline{n} \times \underline{\hat{t}}. \end{aligned} \quad (6.37)$$

The final expression of Eqn. (6.36) is obtained with $0 = \partial U^n / \partial n|_{\text{wall}}$ according to Eqn. (4.28). Note that the integration (6.36) is restricted to the design surface Γ_D , which is subject to the shape control δn . Eqn. (6.36) can be rewritten as

$$\delta_L L = \int_{\Gamma_D} \delta n G_L d\Gamma, \quad (6.38)$$

with the distribution of the local component of the sensitivity derivative

$$G_L = -\mu_{\text{eff}} \frac{\partial \hat{U}^t}{\partial n} \frac{\partial U^t}{\partial n} \hat{t}_i t_i \quad \text{on } \Gamma_D. \quad (6.39)$$

Note that the approximation of G_L is of first-order accuracy, $\mathcal{O}(\delta n)$, since the underlying linear development of the wall boundary condition (5.28) is second-order accurate, $\mathcal{O}(\delta n^2)$.

6.4. Complete Gradient Equation

The complete variation of the objective functional can be expressed as

$$\begin{aligned} \delta J &= (\delta_L + \delta_C + \delta_G) J \\ &= \int_{\Gamma_D} \delta n G_L d\Gamma + \int_{\Gamma_O \cap \Gamma_D} \delta n (G_C + G_G) d\Gamma. \end{aligned} \quad (6.40)$$

Accordingly, the distribution of the *complete gradient* G , or $G_{LCG} = G_L + G_C + G_G$, can be decomposed into the following portions:

- The *local gradient* contribution accounts for local variations of the flow

$$G_L = -\mu_{\text{eff}} \frac{\partial U^t}{\partial n} \frac{\partial \hat{U}^t}{\partial n} t_i \hat{t}_i \quad \text{on } \Gamma_D. \quad (6.41)$$

For an optimisation problem constrained by elliptic PDE, it is non-zero over the entire design surface Γ_D . It is efficiently obtained from the adjoint calculus (6.39).

- The *convective gradient* contribution obtained from a truncated Taylor series expansion is associated with a convective variation of the flow

$$G_C = \frac{\partial j_\Gamma}{\partial n} \quad \text{on } \Gamma_O \cap \Gamma_D. \quad (6.42)$$

This contribution is practically zero for the force objective functional (4.34), cf. Eqns. (5.15) through (5.18).

- The *geometric gradient* contribution originates from changes of the boundary orientation and surface area. According to Eqn. (5.19), the distribution of the geometric gradient of the force objective functional (4.34) reads

$$G_G = \frac{\partial}{\partial x_j^\Gamma} (\pi_{ij} d_i^*) \quad \text{on } \Gamma_O \cap \Gamma_D. \quad (6.43)$$

Convective and geometric gradient contributions, G_C and G_G , only exist if the boundary surface is subject to the control, i.e. $\Gamma_O \cap \Gamma_D = \emptyset$. Otherwise G_C and G_G are not declared. As suggested by Söding [140] in the context of a discrete-adjoint potential flow method,

the surface gradient can be interpreted as the sensitivity of the objective functional with respect to a local volume change:

$$\delta J = \int_{\Gamma_D} \delta n G \, d\Gamma \quad \text{or} \quad G = \frac{\delta J}{\delta n \, d\Gamma} = \frac{\delta J}{\delta V} \quad \text{on} \quad \Gamma_D. \quad (6.44)$$

A positive gradient G implies that a local increase in fluid (solid) volume $\delta V \approx \delta n \, d\Gamma$ leads to a positive (negative) objective functional increment δJ .

6.5. Example: Interior vs. Exterior Evaluation of Forces

A simple model illustrates how two different formulations of the same objective functional lead to two different adjoint setups. The objective functional is the component of the force \underline{F}^* acting on the surface Γ_D of a submerged body into the direction of \underline{d}^* , viz.

$$J = d_i^* F_i^*. \quad (6.45)$$

It can be evaluated in two different ways, referred to as *interior* (i) and *exterior* (ii) strategy:

- (i) According to Figure 6.1 (left), the integration is carried out over the body of interest (dashed line= Γ_O):

$$J = d_i^* F_i^* = -d_i^* \int_{\Gamma_O} \pi_{ij} \, d\Gamma_j. \quad (6.46)$$

The complete body surface is subject to the shape design, so that Γ_O and Γ_D are equivalent. The corresponding boundary conditions obtained for the adjoint problem are (cf. Table 6.2, ADV2):

$$\hat{U} = -\underline{d}^* \quad \text{on} \quad \text{body surf.} \quad (\Gamma_D = \Gamma_O).$$

The external boundary conditions on inlet and wall faces, where the primal velocity is prescribed, are:

$$\hat{U} = \underline{0} \quad \text{on} \quad \text{ext. walls, inlet} \quad (\not\subset \Gamma_O).$$

Note that, for $\Gamma_O \cap \Gamma_D \neq \emptyset$, the partial variation of the objective functional with respect to the shape is non-zero, $\delta_{CG} J \neq 0$.

- (ii) According to Figure 6.1 (right), the force acting on the body is evaluated by integration over the exterior boundaries of the flow domain (dashed line= Γ_O):

$$J = d_i^* F_i^* = -d_i^* \oint_{\Gamma_O} [\rho U_j U_i - \pi_{ij}] \, d\Gamma_j. \quad (6.47)$$

The objective surface is not subject to shape perturbations $\Gamma_O \cap \Gamma_D = \emptyset$, so that the partial variation of the objective functional with respect to the shape is zero, $\delta_{CG} J = 0$. The variation of Eqn. (6.47) is

$$\delta_L J = d_i^* \delta_L F_i^* = -d_i^* \oint_{\Gamma_O} [\rho \delta (U_j U_i) - \delta \pi_{ij}] \, d\Gamma_j. \quad (6.48)$$

The adjoint boundary conditions for this declaration of the objective functional in conjunction with the advection treatment ADV2 are compiled in Table 6.3:

$$\hat{\underline{U}} = \underline{0} \quad \text{on} \quad \text{body surf. } (\Gamma_D = \Gamma \setminus \Gamma_O).$$

The adjoint velocities imposed on the exterior inlet and wall boundaries are

$$\hat{\underline{U}} = \underline{d}^* \quad \text{on} \quad \text{ext. walls, inlet } (\subset \Gamma_O).$$

A comparison of (i) and (ii) reveals two interesting aspects:

- (a) An adjoint transpiration boundary condition is obtained in configuration (i) on the body surface (Γ_O). On the exterior boundaries, where the primal velocity is prescribed (inlet, walls), the adjoint velocity sticks to the boundary surface. The situation is inverse for configuration (ii). The adjoint velocity sticks to the body surface and flows through the outer domain boundaries (inlet, walls, outlet).

The boundary conditions of configuration (ii) remind of the Navier–Stokes equations formulated in an arbitrary Lagrangian Eulerian (ALE) frame of reference for moving grids. For incompressible flow, the convection velocity is replaced by the relative velocities $\underline{U}^r = \underline{U} - \underline{U}^g$ observed on a grid moving with the velocity \underline{U}^g :

$$\begin{aligned} \rho U_j^r \frac{\partial U_i}{\partial x_j} &= \frac{\partial \pi_{ij}}{\partial x_j} + f_i, \\ -\frac{\partial U_j^r}{\partial x_j} &= 0 \quad \text{in } \Omega. \end{aligned} \quad (6.49)$$

With the free-stream velocity \underline{U}_0 , two standard ALE configurations are:

- (a.i) With $\underline{U}^g = -\underline{U}_0$, the frame of reference moves with the body. The corresponding inlet velocity is $\underline{U} = \underline{0}$ in conjunction with $\underline{U} = -\underline{U}_0$ applied to the body surface.
- (a.ii) The standard Eulerian description is recovered for $\underline{U}^g = \underline{0}$, along with $\underline{U} = \underline{U}_0$ on the inlet and $\underline{U} = \underline{0}$ on the body surface.

Both descriptions (a.i/ii) are fully equivalent and lead to the same effective fluxes. The adjoint formulation (i) obviously resembles the moving grid method (a.i), whereas (ii) is similar to the Eulerian description (a.ii).

- (b) Formulation (i) requires to evaluate the complete gradient, $G = G_L + G_C + G_G$, including local, convective and geometric contributions as described in Section 6.4. In formulation (ii), the variation of the objective functional consists of local variations alone, so that the gradient can be evaluated from Eqn. (6.38). This feature is very welcome as it significantly simplifies the gradient evaluation. It is remarkable that the adjoint systems (i/ii) are equivalent from a quasi-ALE point of view (a). A difference is observed in the outlet boundary condition for the following reason: In configuration (ii) the flux of adjoint momentum convected through the inlet re-enters at the outlet. On the contrary, an adjoint “do-nothing” or no-flux outlet boundary condition is found in configuration (i), so that no adjoint momentum is convected through the exterior boundaries of the domain.

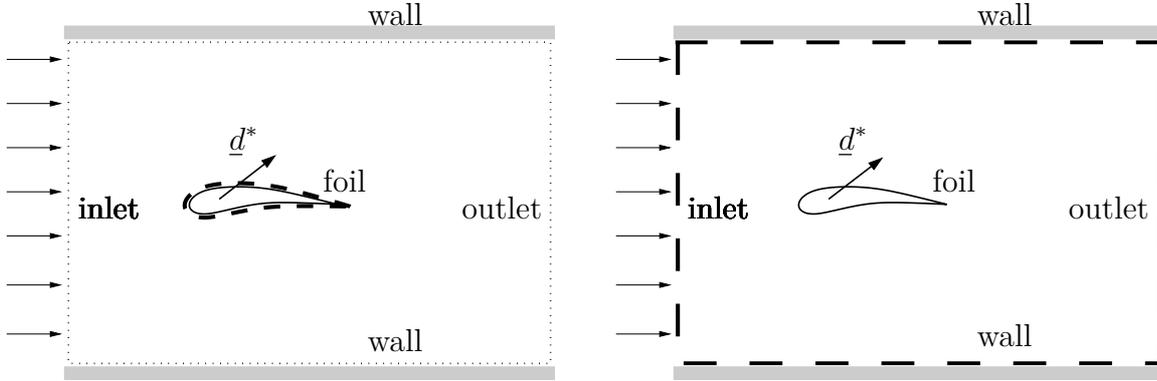


Figure 6.1.: Force integration. Left: integration of surface stress over dashed line ($\Gamma_O = \Gamma_D = \text{foil}$). Right: external flux integration over dashed line ($\Gamma_O = \text{hold-all}$)

Table 6.2.: Adjoint boundary conditions for “interior” force evaluation (ADV2)

boundary	\hat{U}	\hat{p}
wing surf. $\subset \Gamma_O$	$\hat{U} = -\underline{d}^*$	$\hat{p}^n = 0$
ext. walls $\not\subset \Gamma_O$	$\hat{U} = \underline{0}$	$\hat{p}^n = 0$
inflow $\not\subset \Gamma_O$	$\hat{U} = \underline{0}$	$\hat{p}^n = 0$
pressure outflow $\not\subset \Gamma_O$	$\hat{p} n_i \approx \rho U^n \hat{U}_i + \rho U_j \hat{U}_j n_i + \mu_{\text{eff}} \hat{U}_i^n$	

Table 6.3.: Adjoint boundary conditions for “exterior” force evaluation (ADV2)

boundary	\hat{U}	\hat{p}
wing surf. $\not\subset \Gamma_O$	$\hat{U} = \underline{0}$	$\hat{p}^n = 0$
ext. walls $\subset \Gamma_O$	$\hat{U} = \underline{d}^*$	$\hat{p}^n = 0$
inflow $\subset \Gamma_O$	$\hat{U} = \underline{d}^*$	$\hat{p}^n = 0$
pressure outflow $\subset \Gamma_O$	$\hat{p} n_i \approx \mu_{\text{eff}} \hat{U}_i^n$	

7. Finite-Volume Discretisation

The discretisation schemes used to solve the non-linear and the linearised Navier–Stokes equations are introduced in this chapter. These standard discretisation schemes for incompressible flow, which are described below for completeness, are the basis for the derivation of the adjoint discretisation schemes presented in Section 8. Elementary discretisation techniques are described in terms of a scalar transport equation first (Section 7.1) before the method is extended to the requirements of the non-linear Navier–Stokes equations (Sections 7.2 to 7.6) and their linearised form (Section 7.7). Special attention is given to the discretisation features that are of interest for the derivation of the adjoint discretisation considered in the next chapter. A detailed description of the underlying primal numerics can be found in Ferziger and Peric [30] or Rung [132, 133].

7.1. Scalar Transport Equation

A finite-volume (FV) discretisation of second-order accuracy is used to approximate the RANS equations. The computational mesh subdivides the fluid domain completely into convex polyhedra called control volumes (CVs). The mesh is organised in a fully unstructured manner. The CVs consist of a number of cell faces, which are either internal or lie on the boundary of the fluid domain. The faces, in turn, are closed polygons connecting the associated grid vertices. The flow variables are stored in the CV centres in a collocated arrangement. The presentation is based on the following conventions: The internal faces separating CVs P and N are denoted by PN , cf. Figure 7.1. The associated face unit-vectors, \underline{n}_{PN} , are oriented from P to N . Looking from N to P , the face normal and the face fluxes are negative, i.e. $\underline{n}_{NP} = -\underline{n}_{PN}$ and $F_{NP} = -F_{PN}$. In a face-loop over all internal faces PN of the computational mesh each face is visited only once. The flux contributions F_{PN} and F_{NP} are added to the balances of CVs P and N , respectively. Another face-loop runs over the boundary faces PB . According to Figure 7.2 the outward boundary unit-vector \underline{n}_{PB} points from the centre of the boundary-adjacent CV P to the boundary node B . The corresponding boundary flux F_{PB} is added to the balance of CV P . The set of immediate internal neighbours surrounding CV P is denoted $N(P)$. The corresponding set of surrounding boundary faces is $B(P)$.

A steady-state advection-diffusion equation is considered first with a prescribed, solenoidal velocity field \underline{U} :

$$R^{(\phi)} = \rho U_i \frac{\partial \phi}{\partial x_i} - \gamma_\phi \frac{\partial^2 \phi}{\partial x_i^2} - s = 0 \quad \text{in } \Omega. \quad (7.1)$$

γ_ϕ and s denote the diffusion coefficient distributed homogeneously and the source term, respectively. Eqn. (7.1) is transferred to a weak, integral form in order to be solved on

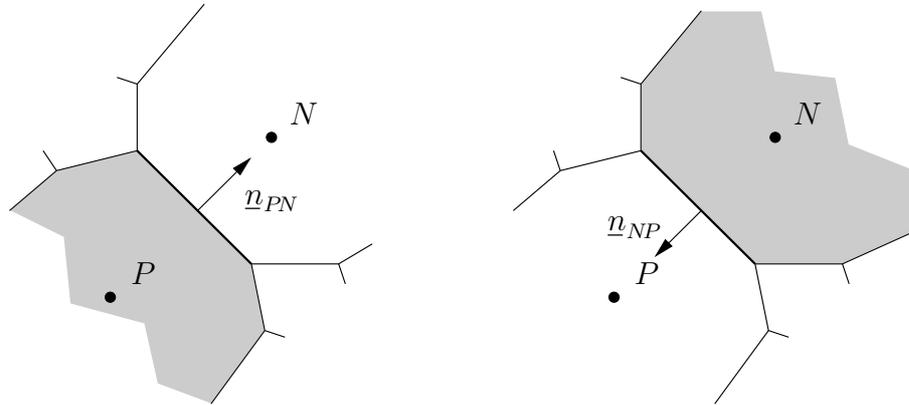


Figure 7.1.: Cell face PN (NP) separating CVs P and N on the left-hand (right-hand) side

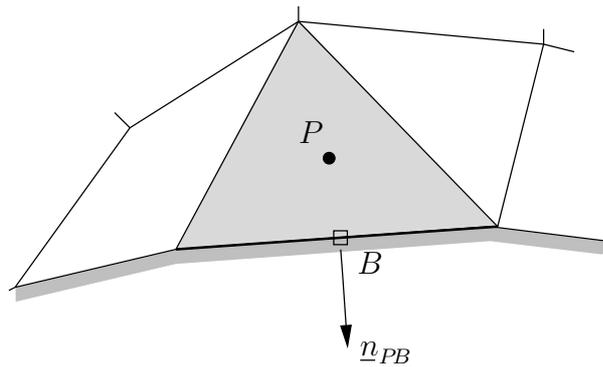


Figure 7.2.: Boundary cell-face PB and boundary unit-vector \underline{n}_{PB}

the cell-level:

$$\begin{aligned}
0 &= \sum_{N(P)} \int_{\Delta\Gamma_{PN}} \left[\rho U_i \phi - \gamma_\phi \frac{\partial \phi}{\partial x_i} \right] d\Gamma_i + \sum_{B(P)} \int_{\Delta\Gamma_{PB}} \left[\rho U_i \phi - \gamma_\phi \frac{\partial \phi}{\partial x_i} \right] d\Gamma_i - \int_{\Delta\Omega_P} s d\Omega \\
&= \sum_{N(P)} F_{PN} + \sum_{B(P)} F_{PB} - S_P \quad \text{for each CV } P.
\end{aligned} \tag{7.2}$$

The face fluxes F can be subdivided into convective and diffusive portions denoted F^c and F^d , respectively. The source term in the balance of CV P is S_P . For the sake of brevity, the following presentation is confined to internal faces PN ; the cell-face indices PN and PB are omitted unless required for clarity.

Face and Cell Integrals

According to Figure 7.3, the centre of the considered cell is P , the cell centre of its neighbour is N , and the cell-face centre is denoted by F . The corresponding position vectors are \underline{x}_P , \underline{x}_N and \underline{x}_F , respectively. The face-adjacent CV centres are connected by

$$\underline{d} = \underline{x}_N - \underline{x}_P. \tag{7.3}$$

The flow variables ϕ are computed and stored in the centre of each CV. In order to evaluate the flux balance for every cell, it is necessary to

- interpolate the face values from the values in the adjacent CV centre, and subsequently
- integrate them over the CV faces.

Unless declared differently, the CV-face values are obtained by linear interpolation from the neighbouring cell centres

$$\phi_F \approx (1 - \lambda)\phi_P + \lambda\phi_N + (\underline{x}_F - \underline{x}_{F'}) \cdot \underline{\nabla}\phi|_{F'}, \tag{7.4}$$

with the interpolation factor

$$\lambda = \frac{(\underline{x}_F - \underline{x}_P) \cdot \underline{d}}{\underline{d} \cdot \underline{d}}. \tag{7.5}$$

According to Figure 7.3, the auxiliary point F' is located in

$$\underline{x}_{F'} = \underline{x}_P + \lambda \underline{d}. \tag{7.6}$$

In general, $\underline{x}_{F'}$ does not coincide with the position of the cell-face centre \underline{x}_F . The correction term in Eqn. (7.4) is treated explicitly using the gradient of ϕ interpolated to the cell faces.

Volume and face integrals are calculated to second-order accuracy from the midpoint rule:

$$\int_{\Delta\Omega} \phi d\Omega \approx \phi_P \Delta\Omega \quad \text{and} \quad \int_{\Delta\Gamma} \phi d\Gamma \approx \phi_F \Delta\Gamma. \tag{7.7}$$

A conservative approximation to the gradient in the internal cell P is obtained through Gauss' method

$$\underline{\nabla}\phi|_P \approx \frac{1}{\Delta\Omega} \sum_{N(P)} (\phi_F \Delta\underline{\Gamma}), \quad (7.8)$$

approximating the face value ϕ_F according to Eqn. (7.4).

Convective Fluxes

The development of convection schemes has been a major field of research in CFD over the past decades [e.g. 160, 161, 49, 157, 129, 84, 85]. If the transport equations are dominated by convection, accuracy and stability of the discretisation scheme are decisive for the quality of the prediction and the robustness of the algorithm. Face integration is carried out according to the midpoint rule (7.7)

$$F^c = \int_{\Delta\Gamma} \rho U_j \phi \, d\Gamma_j \approx [\rho \Delta\Gamma_j U_j]_F \phi_F = \dot{m} \phi_F. \quad (7.9)$$

Standard techniques are used to approximate the face value ϕ_F :

Upstream discretisation scheme (UDS):

$$F^c \approx \dot{m} \phi_F \approx \dot{m} \phi_U = \max[\dot{m}, 0] \phi_P + \min[\dot{m}, 0] \phi_N \quad (7.10)$$

Only the UDS contribution is treated implicitly, whereas the subsequent higher-order schemes are evaluated explicitly in a deferred-correction approach based on the values of the previous iteration.

Downstream discretisation scheme (DDS):

$$F^c \approx \dot{m} \phi_F \approx \dot{m} \phi_D = \max[\dot{m}, 0] \phi_N + \min[\dot{m}, 0] \phi_P \quad (7.11)$$

Up- and downstream cell-face interpolations are first-order accurate.

Central differencing scheme (CDS): ϕ_F is obtained by linear cell-face interpolation according to Eqn. (7.4). UDS and CDS approximations can be blended in a deferred correction approach by $0 < \beta_\phi < 1$:

$$\phi_F = [\phi^{\text{UDS}}]^{\text{impl}} + \beta_\phi [\phi^{\text{CDS}} - \phi^{\text{UDS}}]^{\text{expl}}. \quad (7.12)$$

Like UDS and DDS, CDS leads to a compact stencil that is confined to the nearest cell neighbours. The CDS approximation is second-order accurate. However, it is not limited and prone to produce oscillating solutions.

Unified high-order schemes (MUSCL): Several higher-order schemes can be stated in a unified formulation. Higher-order polynomials used to approximate the cell-face value

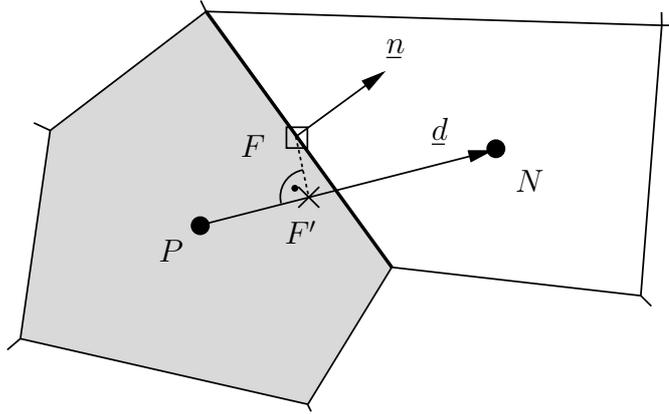


Figure 7.3.: Vicinity of CV face: \underline{d} connecting CV centres P and N ; face unit normal \underline{n}

ϕ_F tend to produce oscillating solutions, particularly in conjunction with steep gradients or discontinuities in the flow field. Such schemes can be limited [160, 161, 49, 157, 129, 85] to be monotonicity preserving or total variation diminishing (TVD), so that new extrema are avoided and existing extrema are not amplified. Through a damping function ψ the spatial derivatives are limited dynamically by locally decreasing the approximation order. The cell-face value ϕ_F is interpolated according to Leer's MUSCL scheme [161] and can be written as a combination of first-order upstream and higher-order approximations characterised by the parameter κ (see Table 7.1)

$$\begin{aligned}\phi_F &\approx \phi_U + \frac{1}{4} [(1 + \kappa) (\phi_D - \phi_U) \psi(r) + (1 - \kappa) (\phi_U - \phi_{UU}) \psi(r^{-1})] \\ &\approx \phi_U + \frac{1}{2} (\phi_D - \phi_U) \varphi(r)\end{aligned}\quad (7.13)$$

with

$$\varphi(r) = \left[\frac{1 + \kappa}{2} \psi(r) + \frac{1 - \kappa}{2} r\psi(r^{-1}) \right]. \quad (7.14)$$

A potential non-monotonicity in the interval $[UU-D]$ is detected by negative values of the sensor

$$r = \frac{\phi_U - \phi_{UU}}{\phi_D - \phi_U}. \quad (7.15)$$

The scheme is nonlinear since ψ depends on ϕ . Several interpolation schemes are obtained depending on the choice of κ as listed in Table 7.1 for the unlimited case with $\psi = 1$ [e.g. 84, 136, 133]. The corresponding unlimited distributions $\varphi(r)$ are plotted in the Sweby diagram 7.4 [157]. Over the region shaded in light grey, the TVD property is satisfied. Second-order TVD schemes are limited to the area plotted in dark grey.

An extension of the scheme by Lien and Leschziner [84], e.g. documented by Schatz [136], was used in this study to limit the unified description (7.13) to the shaded area

$$\psi(r) = \min\text{mod}[1, \omega_2 r]; \quad r\psi(r^{-1}) = \min\text{mod}[r, \omega_1]. \quad (7.16)$$

The minmod-operator is defined as

$$\text{minmod}(A, \omega B) = \text{sgn}(A) \max[0, \min(|A|, \omega B \text{sgn}[A])] \quad (7.17)$$

with the argument values

$$\omega_1 = \max\left(1, 0.9 \frac{3 - \kappa}{1 - \kappa}\right); \quad \omega_2 = \max\left(1, 0.9 \frac{3 + \kappa}{1 + \kappa}\right). \quad (7.18)$$

On unstructured grids, the variable value ϕ_{UU} of the remote node UU is not directly available (Figure 7.5). Instead, the difference $(\phi_U - \phi_{UU})$ is reconstructed according to

$$\phi_U - \phi_{UU} = \phi_U - \phi_D + \phi_D - \phi_{UU} \approx \phi_U - \phi_D + 2 \underline{d}_F \cdot \underline{\nabla} \phi|_U. \quad (7.19)$$

The substitute $\phi_D - \phi_{UU} \approx 2 \underline{d}_F \cdot \underline{\nabla} \phi|_U$ is second-order accurate with an effective mesh-size of $2|\underline{d}|$. The gradient $\underline{\nabla} \phi|_U$ is determined explicitly from the solution of the previous iteration.

Mind that the approximation of face integrals is limited to second-order accuracy since the mid-point rule is used for integration (7.7).

Diffusive Fluxes

The diffusive face-flux is approximated as

$$F^d = - \int_{\Delta\Gamma} \gamma_\phi \frac{\partial\phi}{\partial n} d\Gamma \approx - \left[\gamma_\phi \frac{\partial\phi}{\partial n} \right]_F \Delta\Gamma. \quad (7.20)$$

When the grid is non-orthogonal, i.e. the face normal \underline{n} is not aligned with \underline{d} as shown in Figure 7.6, the face-normal derivative, $\partial\phi/\partial n = \underline{n} \cdot \underline{\nabla} \phi$, is decomposed as follows

$$\begin{aligned} \frac{\partial\phi}{\partial n} \Big|_F &= \left(\frac{\partial\phi}{\partial x_j} d_j \right)_F \frac{d_i n_i}{d_l^2} + \left(\frac{\partial\phi}{\partial x_j} \right)_F \left(n_j - d_j \frac{d_i n_i}{d_l^2} \right) \\ &\approx (\phi_N - \phi_P) \frac{d_i n_i}{d_j^2} + \left(\frac{\partial\phi}{\partial x_j} \right)_F \left(n_j - d_j \frac{d_i n_i}{d_l^2} \right). \end{aligned} \quad (7.21)$$

A relaxation exponent α is introduced to control the implicit contribution

$$\frac{\partial\phi}{\partial n} \Big|_F \approx \frac{(\phi_N - \phi_P)}{\sqrt{d_m^2}} \left(\frac{d_i n_i}{\sqrt{d_l^2}} \right)^\alpha + \left(\frac{\partial\phi}{\partial x_j} \right)_F \left[n_j - \frac{d_j}{\sqrt{d_m^2}} \left(\frac{d_i n_i}{\sqrt{d_l^2}} \right)^\alpha \right]. \quad (7.22)$$

Table 7.1.: Unified formulation for unlimited convection schemes [84, 136, 133]

scheme	κ	deferred correction	$\varphi(r)$
CDS	1	$(\phi_D - \phi_U) / 2$	1
LUDS	-1	$(\phi_U - \phi_{UU}) / 2$	r
QUICK	1/2	$(3\phi_D - 2\phi_U - \phi_{UU}) / 8$	$3/4 + r/4$
CUI	1/3	$(2\phi_D - \phi_U - \phi_{UU}) / 6$	$2/3r + r/3$

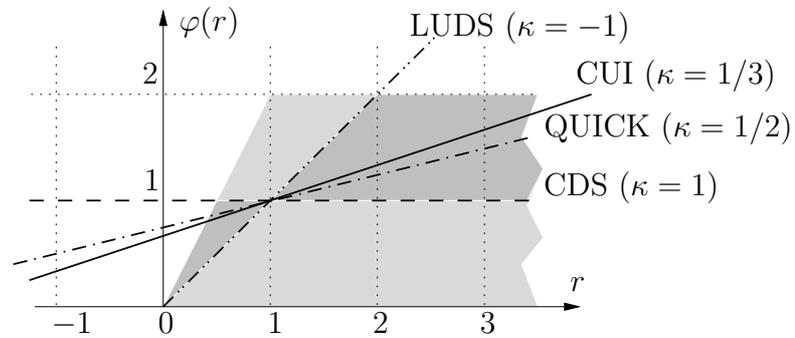


Figure 7.4.: Unified formulation of higher-order convection schemes; the TVD property is satisfied over the area shaded in light grey. Second-order TVD schemes are restricted to the dark grey zone (Sweby [157]).

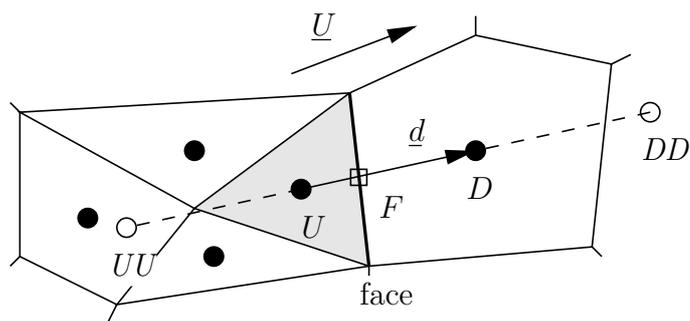


Figure 7.5.: Reconstruction of remote virtual nodes DD and UU on unstructured grids

Eqn. (7.21) is reproduced for $\alpha = 1$. The leading right-hand side term is treated implicitly, whereas the second is evaluated explicitly based on the interpolated cell-face gradient.

Algebraic Equation System

In the previous sections, the PDE (7.1) has been discretised in terms of the variable values stored in the CV centres. The number of cell faces varies in a fully unstructured grid topology. A compact discretisation leads to the algebraic equation for CV P

$$A_P \phi_P + \sum_{N(P)} A_N \phi_N = S. \quad (7.23)$$

The corresponding system of equations is

$$A_{ii} \phi_i + \sum_{j \in N(i)} A_{ij} \phi_j = \sum_j A_{ij} \phi_j = S_i, \quad i = 1, \dots, n \quad (7.24)$$

or

$$\underline{\underline{A}} \cdot \underline{\phi} = \underline{S}. \quad (7.25)$$

According to Eqn. (7.24), $\underline{\underline{A}}$ [$n \times n$] is a sparse coefficient matrix. On unstructured grids, the non-zero entries are distributed in an irregular pattern. $\underline{\phi}$ [n] and \underline{S} [n] represent the vector of unknowns and the right-hand side vector, respectively.

7.2. Momentum Equations

The techniques introduced in Section 7.1 are tailored to the non-linear, vector-valued Navier–Stokes equations (4.9) and (4.10). The linearised system matrix for the momentum–continuity problem is

$$\begin{pmatrix} \underline{\underline{A}} & \underline{\underline{B}} \\ \underline{\underline{C}} & \underline{\underline{Q}} \end{pmatrix} \cdot \begin{pmatrix} \underline{U} \\ \underline{p} \end{pmatrix} = \begin{pmatrix} \underline{S} \\ \underline{Q} \end{pmatrix}, \quad (7.26)$$

wherein the first line represents the discrete momentum equations, and the discrete continuity constraint is treated in the second line. An incomplete Picard linearisation is used,

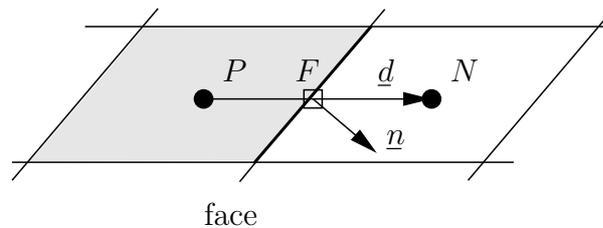


Figure 7.6.: Face-based definition of mesh non-orthogonality

so that the Cartesian components of the momentum equations decouple. The resulting sub-matrices $\underline{\underline{A}}_{ij}$ [$n \times n$] and $\underline{\underline{B}}_i$ [$n \times n$] are

$$\underline{\underline{A}} = \begin{pmatrix} \underline{\underline{A}}_{11} & \underline{\underline{0}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{A}}_{22} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{0}} & \underline{\underline{A}}_{33} \end{pmatrix}; \quad \underline{\underline{B}} = \begin{pmatrix} \underline{\underline{B}}_1 \\ \underline{\underline{B}}_2 \\ \underline{\underline{B}}_3 \end{pmatrix}. \quad (7.27)$$

The corresponding variable and right-hand side vectors are

$$\underline{U} = \begin{pmatrix} \underline{U}_1 \\ \underline{U}_2 \\ \underline{U}_3 \end{pmatrix}; \quad \underline{S} = \begin{pmatrix} \underline{S}_1 \\ \underline{S}_2 \\ \underline{S}_3 \end{pmatrix}.$$

The discrete continuity constraint is described by the sub-matrices $\underline{\underline{C}}_i$ [$n \times n$]

$$\underline{\underline{C}} = \begin{pmatrix} \underline{\underline{C}}_1 & \underline{\underline{C}}_2 & \underline{\underline{C}}_3 \end{pmatrix}. \quad (7.28)$$

For solenoidal velocity fields, the right-hand side vector \underline{Q} is zero. Instead of solving the coupled system (7.26), the sub-systems are solved sequentially in a segregated approach. The underlying FV discretisation for the momentum equations is presented below, followed by a description of the SIMPLE-type pressure-correction scheme that is used to compute the pressure.

The balance of momentum for each CV P is

$$\sum_{N(P)} F_i|_{PN} + \sum_{B(P)} F_i|_{PB} = S_i|_P. \quad (7.29)$$

The face-fluxes contain convective, diffusive and pressure contributions. Discarding the cell and face indices, these are:

$$F_i = F_i^c + F_i^d + F_i^p. \quad (7.30)$$

The source term \underline{S} hosts any source of momentum, or flux terms discretised in a non-conservative manner.

The **convective flux** of momentum through a cell face is non-linear in the velocity \underline{U} . Using a Picard iteration, the mass fluxes \dot{m} are explicitly evaluated from the cell-centre velocities of the previous iteration by linear cell-face interpolation (7.4):

$$F_i^c = \int_{\Delta\Gamma} \rho U_j U_i d\Gamma_j \approx [\rho \Delta\Gamma_j U_j]_F U_i|_F = \dot{m} U_i|_F. \quad (7.31)$$

The cell-face velocity \underline{U}_F is calculated from the convection schemes described in Eqns. (7.9) to (7.19).

The **viscous diffusion flux** through a CV face is

$$F_i^d = - \int_{\Delta\Gamma} 2 \mu_{\text{eff}} S_{ij} d\Gamma_j = - \int_{\Delta\Gamma} \mu_{\text{eff}} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) d\Gamma_j. \quad (7.32)$$

The first term, $\underline{n} \cdot \nabla U$, equivalent to the corresponding term of the scalar transport equation is treated implicitly. The additional contribution, $\nabla U \cdot \underline{n}$, is calculated explicitly. The gradient calculated in the CV centre is interpolated to the face centre according to

$$[\nabla U]_F \approx \lambda [\nabla U]_N + (1 - \lambda) [\nabla U]_P. \quad (7.33)$$

The pressure term is calculated in a conservative manner by summation over the surrounding faces for each internal CV P

$$\int_{\Delta\Omega} \frac{\partial p}{\partial x_i} d\Omega \approx \Delta\Omega \left[\frac{\partial p}{\partial x_i} \right]_P = \sum_{N(P)} F_i^p|_{PN}, \quad (7.34)$$

wherein the pressure face flux is

$$F_i^p = \int_{\Delta\Gamma} p d\Gamma_i \approx p_F \Delta\Gamma_i. \quad (7.35)$$

The face value p_F is linearly interpolated between the neighbouring cell-centres according to Eqn. (7.4).

7.3. Pressure-Correction Scheme

In an incompressible flow regime, the treatment of the pressure is not as obvious as the calculation of the velocity. The equation of continuity, which is left to determine the pressure, does not contain the pressure itself. The problem is overcome through a pressure-Poisson equation that is obtained by taking the divergence of the momentum equation:

$$\frac{\partial R_i}{\partial x_i} = \frac{\partial}{\partial x_i} \left[\rho U_j \frac{\partial U_i}{\partial x_j} - \frac{\partial \pi_{ij}}{\partial x_j} - f_i \right] = 0 \quad \text{in } \Omega. \quad (7.36)$$

Assuming that viscosity and density are distributed homogeneously, $\underline{0} \approx \nabla \rho \approx \nabla \mu_{\text{eff}}$, and \underline{f} is solenoidal, Eqn. (7.36) reduces to

$$\frac{1}{\rho} \frac{\partial^2 p}{\partial x_i^2} = - \frac{\partial U_j}{\partial x_i} \frac{\partial U_i}{\partial x_j} \quad \text{in } \Omega. \quad (7.37)$$

Following the notation of Ferziger and Peric [30], the SIMPLE algorithm by Caretto et al. [7] is devised from the semi-discrete equations of momentum and continuity. It leads to a consistent, discrete pressure-correction equation corresponding to the analytical form (7.37).

The starting point for the iterative scheme is the divergence-free velocity field $U_i|^{m-1}$ and the compatible pressure field $p|^{m-1}$ of the previous iteration ($m-1$). In the first step, a velocity prediction $U_i^*|_P^m$ is calculated from the semi-discrete, Picard-linearised equations of momentum

$$A_P U_i^*|_P^m + \sum_{N(P)} A_N U_i^*|_N^m = -\Delta\Omega \frac{\partial p}{\partial x_i} \Big|_P^{m-1} + S_i|_P^{m-1}. \quad (7.38)$$

The discrete source term of the momentum equations, S_i , is considered to be independent of U_i and p , so that the iteration counter is omitted in the following. The velocity prediction U_i^* obtained from Eqn. (7.38) is not necessarily solenoidal. The corrections U'_i and p' applied to the velocity field and the corresponding pressure field

$$U_i|_P^m = U_i^*|_P^m + U'_i|_P^m \quad \text{and} \quad p|_P^m = p|_P^{m-1} + p'|_P^m, \quad (7.39)$$

lead to a divergence-free velocity field $U_i|_P^m$. An equation for the velocity correction U'_i is obtained subtracting Eqn. (7.38) from the momentum equation for the corrected flow

$$A_P U_i|_P^m + \sum_{N(P)} A_N U_i|_N^m = -\Delta\Omega \left. \frac{\partial p}{\partial x_i} \right|_P^m + S_i|_P, \quad (7.40)$$

viz.

$$A_P \underline{U'_i|_P^m} + \sum_{N(P)} A_N U'_i|_N^m = -\Delta\Omega \left. \frac{\partial p'}{\partial x_i} \right|_P^m. \quad (7.41)$$

The underlined term is neglected in the following, cf. for example Patankar [114], resulting in the pressure-velocity correlation

$$U'_i|_P \approx -\frac{\Delta\Omega}{A_P} \left[\frac{\partial p'}{\partial x_i} \right]_P. \quad (7.42)$$

Postulating continuity of mass for the corrected velocity field $U_i|_P^m$

$$0 = \frac{\partial}{\partial x_i} (\rho U_i|_P^m) = \frac{\partial}{\partial x_i} (\rho U_i^*|_P^m) + \frac{\partial}{\partial x_i} (\rho U'_i|_P^m), \quad (7.43)$$

in conjunction with Eqn. (7.42) leads to a Poisson-type pressure-correction equation

$$\frac{\partial}{\partial x_i} \left[\rho \frac{\Delta\Omega}{A_P} \left(\frac{\partial p'}{\partial x_i} \right)_P \right] = \frac{\partial}{\partial x_i} (\rho U_i^*|_P). \quad (7.44)$$

The iteration counter has been dropped for brevity. A FV approximation reads

$$\sum_{N(P)} \int_{\Delta\Gamma_{PN}} \overline{\left(\frac{\rho \Delta\Omega}{A_P} \right)} \frac{\partial p'}{\partial x_i} d\Gamma_i = \sum_{N(P)} \int_{\Delta\Gamma_{PN}} \rho U_i^* d\Gamma_i \quad \text{for each CV } P. \quad (7.45)$$

The fluxes need to be evaluated at the cell faces, so that the overlined term including the central coefficient of the momentum equation A_P has to be interpolated at the face centres F . The volume of the auxiliary CV, indicated by the dashed line in Figure 7.7, is determined via

$$\overline{\Delta\Omega}|_F \approx d_j \Delta\Gamma_j. \quad (7.46)$$

The pressure-correction p' obtained from Eqn. (7.45) is used to correct the pressure field according to (7.39). The corresponding velocity correction for the volume or mass flux update is obtained from (7.42)

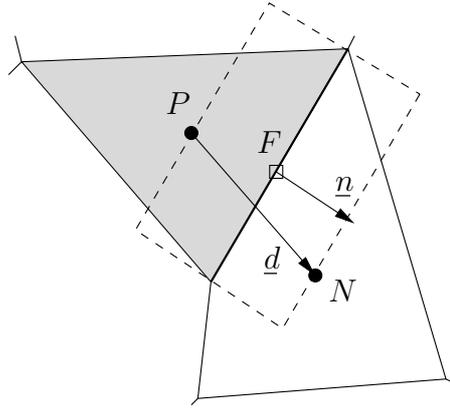


Figure 7.7.: Auxiliary CV $\overline{\Delta\Omega}|_F$ outlined by dashed line

Rhie and Chow Interpolation

On collocated, cell-centred grids, the discretisation of first derivatives contained in gradient and divergence operators is prone to produce oscillations. A straight-forward approximation of the first derivative in the CV centres, based on $2\Delta x$ -differences, does not resolve the oscillation shown in Figure 7.8. The implications for the momentum-continuity problem of the Navier–Stokes equations are:

- The pressure gradient in the momentum equation does not detect potential pressure oscillations.
- The divergence operator on the right-hand side of Eqn. (7.44) does not resolve oscillations in the velocity field, so that they are not considered in the calculated pressure correction.

To obtain smooth solutions, the interpolated cell-face velocity in Eqn. (7.45) is augmented by a third derivative of the pressure as suggested by Rhie and Chow [127]:

$$U_i^*|_F = \overline{(U_i^*)}_F + [\Delta\Gamma_j d_j]_F \left(\overline{\left(\frac{1}{A_P} \right)}_F \left[\left(\frac{\partial p}{\partial x_i} \right)_F - \overline{\left(\frac{\partial p}{\partial x_i} \right)}_F \right] \right). \quad (7.47)$$

The over-lined terms are linearly interpolated from the neighbouring cell-centres. The leading pressure gradient is a CDS approximation based on the adjacent cell-centre values.

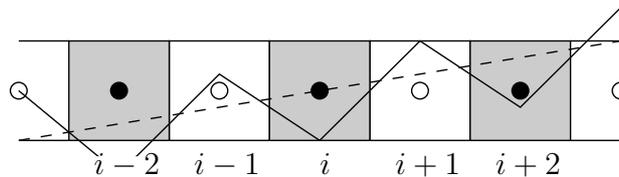


Figure 7.8.: Chequer-board distribution on collocated grids; oscillating distribution (solid line) and its slope resolved by a $2\Delta x$ derivative approximation (dashed line)

Adding a third derivative to the face fluxes can be interpreted as a fourth-order damping term [e.g. 115] applied to the pressure-Poisson equation. The associated error is reduced quadratically when the mesh is refined. If the pressure oscillates, the damping term becomes large and inhibits the oscillation.

Matrix-Vector Interpretation

A matrix-vector interpretation of the SIMPLE algorithm according to Eqn. (7.26) is very instructive for the discrete-adjoint analysis presented in Chapter 8. The inverse of the system matrix $\underline{\underline{A}}^{-1}$ is approximated by $\tilde{\underline{\underline{A}}}^{-1}$ with

$$\tilde{A}_{ij} = \begin{cases} A_{ij}, & \text{if } i = j \\ 0, & \text{else.} \end{cases} \quad (7.48)$$

According to Eqn. (7.38), a velocity prediction \underline{U}^* is calculated from the Picard-linearised momentum equations, which are solved for each velocity component individually:

$$\underline{\underline{A}} \cdot \underline{U}^* = -\underline{\underline{B}} \cdot \underline{p}^{m-1} + \underline{S}. \quad (7.49)$$

Subsequently, a pressure-correction \underline{p}' is determined from the matrix-vector equivalent to Eqn. (7.45):

$$\left(\underline{\underline{C}} \cdot \tilde{\underline{\underline{A}}}^{-1} \cdot \underline{\underline{B}} \right) \cdot \underline{p}' = \underline{\underline{C}} \cdot \underline{U}^*. \quad (7.50)$$

Eqn. (7.50) represents a discrete continuity constraint on the corrected velocity field in combination with $\underline{U}' = -\tilde{\underline{\underline{A}}}^{-1} \cdot \underline{\underline{B}} \cdot \underline{p}'$, cf. Eqn. (7.42). Finally, velocity and pressure fields are corrected according to Eqns. (7.39), i.e.

$$\underline{p}^m = \underline{p}^{m-1} + \alpha_p \underline{p}' \quad \text{and} \quad \underline{U}^m = \underline{U}^* - \tilde{\underline{\underline{A}}}^{-1} \cdot \underline{\underline{B}} \cdot \underline{p}'. \quad (7.51)$$

An under-relaxation factor $0 < \alpha_p < 1$ is applied in the pressure update and the system is iterated until convergence. For interior CVs, the gradient operator $\underline{\underline{B}}$ applied to the pressure is the transposed of the negative divergence operator $\underline{\underline{C}}$ for the velocity. This verifies the symmetry property of the approximation to the Schur complement or the Laplace operator $(\underline{\underline{C}} \cdot \tilde{\underline{\underline{A}}}^{-1} \cdot \underline{\underline{B}})$.

7.4. Boundary Conditions

A local system of boundary coordinates (n, t, s) is used for the formulation of boundary conditions:

$$\underline{U} = U^n \underline{n} + U^t \underline{t} + U^s \underline{s}. \quad (7.52)$$

The in-plane unit vector \underline{t} is computed from

$$\underline{t} = \underline{n} \cdot \underline{S} \cdot \underline{T} / |\underline{n} \cdot \underline{S} \cdot \underline{T}| \approx \underline{T} \cdot \Delta \underline{U} / |\underline{T} \cdot \Delta \underline{U}| \quad (7.53)$$

with

$$\Delta \underline{U} \approx \underline{U}_B - \underline{U}_P \quad \text{and} \quad \underline{T} = \underline{I} - \underline{n}\underline{n}. \quad (7.54)$$

According to Figure 7.9, \underline{U}_B and \underline{U}_P represent the velocity vectors in the centres B of the boundary face and the adjacent interior CV P , respectively. \underline{I} $[3 \times 3]$ denotes the unit matrix. The local coordinate system is completed by the second in-plane unit vector $\underline{s} = \underline{t} \times \underline{n}$.

As shown in Section 4.2, a **no-slip wall** boundary condition, $\underline{U} = \underline{U}_{\text{wall}}$, implies

$$0 = U^n = \frac{\partial U^n}{\partial n}, \quad (7.55)$$

so that the convective flux through a wall face, $\underline{F}^c|_{\text{wall}}$, is null. In a strong conservation formulation, the force acting on a wall surface is equal to the flux of momentum through the boundary:

$$\underline{F}|_{\text{wall}} = [\underline{F}^d + \underline{F}^p]_{\text{wall}} = - \int_{\Delta \Gamma} (2\mu_{\text{eff}} \underline{S} - p \underline{I}) \cdot d\underline{\Gamma}. \quad (7.56)$$

The viscous diffusive flux of momentum $\underline{F}^d|_{\text{wall}}$ is applied to the balance of the wall-adjacent CV. The discrete integration of the wall shear stress should avoid to generate a normal stress component τ^{nn} , which does not exist over no-slip wall boundaries due to $0 = U^{t,t} = U^{s,s} = U^{n,n}$. This is considered in the following for both low- and high-Re boundary conditions by means of the projection matrix $\underline{T} = \underline{I} - \underline{n}\underline{n}$.

When a **low-Re strategy** is employed, the governing flow equations are integrated numerically right up to the wall. A fine grid resolution is required in the normal direction to resolve the wall boundary layer numerically. The wall distance of the first cell centre, $y_P = \underline{n} \cdot (\underline{x}_B - \underline{x}_P)$, is non-dimensionalised according to

$$y_P^+ = y_P U^\tau / \nu \quad \text{with} \quad U^\tau = \sqrt{\tau^w} / \rho. \quad (7.57)$$

The first interior node should be placed in the viscous sublayer, i.e. y_P^+ should be of the order of one. The diffusive flux of momentum through a no-slip wall face contained in

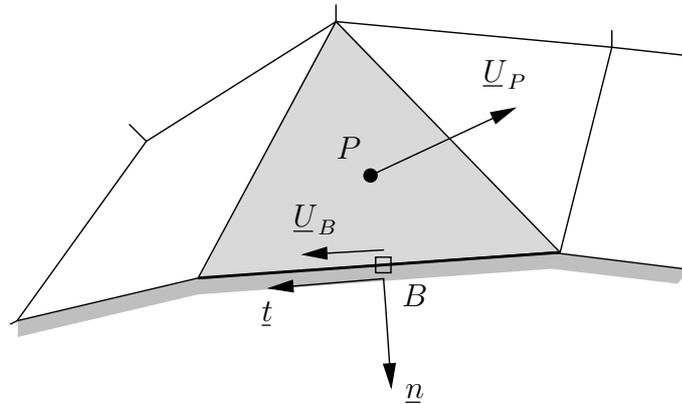


Figure 7.9.: Grid topology next to the wall

Eqn. (7.56) is calculated via

$$\underline{F}^d|_{\text{wall}} = - \int_{\Delta\Gamma} 2\mu \underline{S} \cdot d\underline{\Gamma} \approx - \frac{\mu \Delta\Gamma}{y_P} \underline{T} \cdot \Delta\underline{U}. \quad (7.58)$$

The boundary condition for the k - ε and k - ω type turbulence models were taken from Wilcox [166, Sections 4.7 and 4.9]. On the wall, $y = 0$, the turbulent kinetic energy k is zero which is asymptotically consistent with

$$0 = \frac{\partial k}{\partial n} \quad \text{on the wall surface, i.e.} \quad k_B \approx k_P. \quad (7.59)$$

A consistent formulation for the dissipation rate ε reads

$$\varepsilon \approx \nu \frac{\partial^2 k}{\partial n^2} \approx 2\nu \left(\frac{\partial \sqrt{k}}{\partial n} \right)^2. \quad (7.60)$$

A pseudo boundary-condition is implemented defining the dissipation rate in the first cell layer

$$\varepsilon_P \approx 2\nu \frac{k_P}{y_P^2}. \quad (7.61)$$

The specific dissipation rate ω , which is required in conjunction with k - ω type turbulence models, has a singularity at the wall boundary since k asymptotically tends to zero for $y \rightarrow 0$:

$$\omega = \frac{\varepsilon}{C_\mu k}. \quad (7.62)$$

The problem is resolved setting $\omega_P = \omega(y_P)$ in the first cell layer to its analytical near-wall solution in the absence of convection and turbulent viscosity:

$$\omega_P = 80 \nu / y_P^2. \quad (7.63)$$

Alternatively, a **high-Re implementation** based on logarithmic wall functions can be used to model the velocity distribution in the boundary layer. Over the logarithmic region, $20 < y^+ < 300$, the velocity profile is well described by the velocity profile sketched in Figure 7.10:

$$U^+ = -\frac{U^\tau}{U^\tau} = \frac{1}{\kappa} \ln(Ey^+) \quad \text{with} \quad \kappa = 0.41; \quad E = 8.432. \quad (7.64)$$

The negative sign in Eqn. (7.64) is a consequence of the definition of the wall coordinate system (7.52) and (7.53). The pseudo boundary conditions for U^t and the turbulence variables k, ε, ω in the logarithmic layer are:

$$U^t = -\frac{U^\tau}{\kappa} \ln(Ey^+); \quad \frac{\partial k}{\partial n} = 0; \quad \varepsilon_P = \left[\frac{C_\mu^{3/4} k^{3/2}}{\kappa y} \right]_P; \quad \omega_P = \left[\frac{\sqrt{k}}{C_\mu^{1/4} \kappa y} \right]_P. \quad (7.65)$$

Based on the assumption of the logarithmic law of the wall (7.64), these quantities can be expressed in terms of the shear velocity U^τ :

$$U^t = -\frac{U^\tau}{\kappa} \ln(Ey^+); \quad k = \frac{(U^\tau)^2}{C_\mu^{1/2}}; \quad \varepsilon_P = \frac{(U^\tau)^3}{\kappa y_P}; \quad \omega_P = \frac{U^\tau}{C_\mu^{1/2} \kappa y_P}. \quad (7.66)$$

Instead of prescribing the boundary values explicitly, the diffusive flux of momentum through the wall (shear forces applied to the left-hand side of the momentum equations) is applied to the momentum equations in a conservative cut-free strategy:

$$F_i^d|_{\text{wall}} = - \int_{\Delta\Gamma} 2\mu_{\text{eff}} S_{ij} d\Gamma_j = - \int_{\Delta\Gamma} \tau^w t_i d\Gamma = - \int_{\Delta\Gamma} \rho(U^\tau)^2 t_i d\Gamma. \quad (7.67)$$

When a high-Re approach is pursued, the diffusive momentum flux through the boundary is modelled in terms of the shear velocity $\tau^w = \rho(U^\tau)^2$. A close coupling between the equations of momentum and turbulence is achieved expressing the boundary stress in terms of both,

- the velocity next to the wall described through the logarithmic law of the wall (7.64), $U^\tau = -\kappa U^t(y_P) / \ln(Ey_P^+)$, and
- the turbulent kinetic energy $U^\tau = C_\mu^{1/4} k_P^{1/2}$.

Accordingly, the flux of momentum through the considered wall face can be calculated in a formulation that is analogue to the low-Re approach (7.58),

$$\underline{F}^d|_{\text{wall}} = - \int_{\Delta\Gamma} 2\mu_{\text{eff}} \underline{S} \cdot d\underline{\Gamma} \approx - \frac{\mu_{\text{log}} \Delta\Gamma}{y_P} \underline{T} \cdot \Delta\underline{U}, \quad (7.68)$$

based on the auxiliary definition of the viscosity in terms of the logarithmic law of the wall

$$\mu_{\text{log}} = \frac{\rho \kappa C_\mu^{1/4} k_P^{1/2}}{\ln(Ey_P^+)} y_P. \quad (7.69)$$

A Neumann boundary condition is imposed on the pressure for both low- and high-Re formulations. A first-order approximation reads

$$0 = \frac{\partial p}{\partial n} \approx (p_B - p_P) / y_P \quad \text{or} \quad p_B \approx p_P. \quad (7.70)$$

With the wall pressure flux approximated as $\underline{F}^p \approx \Delta\underline{\Gamma} p_B$, the total flux of momentum through a wall face is calculated according to Eqn. (7.56) from $\underline{F}|_{\text{wall}} = \underline{F}^d + \underline{F}^p$.

Under the assumption of zero curvature, the subsequent boundary condition is obtained for **symmetry** planes or slip-walls

$$0 = U^n = \frac{\partial U^t}{\partial n} = \frac{\partial U^s}{\partial n}. \quad (7.71)$$

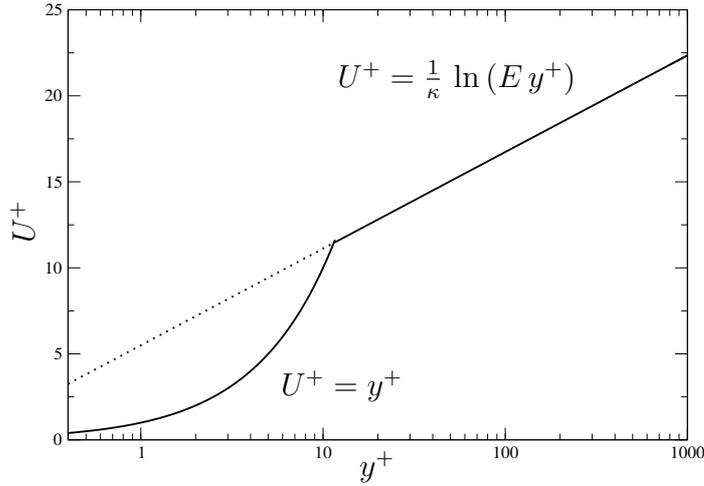


Figure 7.10.: Logarithmic law of the wall

The corresponding diffusive momentum flux through a symmetry face is

$$\underline{F}^d|_{\text{sym}} = - \int_{\Delta\Gamma} 2\mu_{\text{eff}} \underline{S} \cdot d\underline{\Gamma} \approx -2 \frac{\mu_{\text{eff}} \Delta\Gamma}{y_P} \underline{P} \cdot \Delta\underline{U}, \quad (7.72)$$

wherein the boundary-normal portion of the velocity difference, $\Delta\underline{U} = \underline{U}_B - \underline{U}_P$, is calculated through scalar multiplication by $\underline{P} = \underline{n}\underline{n}$. A zero-gradient Neumann boundary condition is applied to the pressure and further variables for turbulence, heat, etc.

$$0 = \frac{\partial\phi}{\partial n} \approx \frac{\phi_B - \phi_P}{y_P} \quad \text{or} \quad \phi_B \approx \phi_P. \quad (7.73)$$

The convective flux \underline{F}^c through a symmetry face is zero. With $\underline{F}^p \approx \Delta\Gamma p_B$, the boundary flux of momentum is computed from

$$\underline{F}|_{\text{sym}} = \underline{F}^d + \underline{F}^p. \quad (7.74)$$

At the **inlet**, a Dirichlet boundary condition is defined for the velocity components, the turbulence variables or additional scalar variables

$$\mathcal{D} = \phi_B, \quad (7.75)$$

together with a Neumann condition for the pressure (7.70), i.e.

$$\begin{aligned} \underline{F}|_{\text{in}} &= \underline{F}^c + \underline{F}^d + \underline{F}^p \\ &\approx \dot{m} \underline{U}_B - \mu_{\text{eff}} \Delta\Gamma \cdot [\underline{\nabla}\underline{U} + (\underline{\nabla}\underline{U})^T]_B + \Delta\Gamma p_B. \end{aligned} \quad (7.76)$$

A prescribed pressure value at the outlet is referred to as **pressure outlet** boundary condition, viz.

$$\mathcal{D} = p_B. \quad (7.77)$$

The normal derivatives of the velocity, turbulence and scalar variables are set to zero at the pressure outlet according to Eqn. (7.73), viz.

$$\underline{F}|_{\text{out}} \approx \underline{F}^c + \underline{F}^p \approx \dot{m} \underline{U}_B + \Delta \underline{\Gamma} p_B. \quad (7.78)$$

The boundary condition for the **pressure-correction equation** is $\partial p' / \partial n = 0$ if the mass flux for the considered boundary face is prescribed, e.g. on inlet, wall or symmetry faces. In that case, no change of the flux is allowed. A prescribed pressure value on the pressure outlet, i.e. a zero pressure correction p' , leads to a velocity correction on the boundary face according to Eqn. (7.42).

The discrete treatment of turbulence variables is in accordance with Ferziger and Peric [30] and Rung [131]. Since it is not in the focus of the following analysis of the adjoint discretisation, it is omitted for the sake of brevity.

7.5. Solution Algorithm

The state equations are solved sequentially in a segregated manner for

- (i) the velocity components U_i determined from the momentum equations;
- (ii) the pressure correction p' calculated from the pressure-correction equation;
- (iii) the turbulence variables k, ε or ω , optionally augmented by
- (iv) the free water-surface in a volume of fluid (VOF) or level set approach; or
- (v) further active scalar variables ϕ .

The individual equations associated with the steps (i) to (v) are solved iteratively in so-called *inner iterations*. The Picard-linearised momentum equations (7.38) are based on the mass fluxes of the previous iteration. The velocity prediction from the momentum equations does not necessarily satisfy continuity. Based on the imbalance of mass, a pressure correction is predicted from Eqn. (7.45) in step (ii) which is subsequently used to correct the pressure (7.39), the velocities (7.42) and the mass fluxes consistently with (7.47) to achieve continuity of the velocity field. The corrected state is used to compute the variables in the steps (iii) to (v). The steps (i) to (v) are repeated in a so-called *outer iteration* loop until a convergence criterion is met for the coupled system (Section 7.5). Further “passive variables”, which do not influence the other field variables, can be addressed after solving the coupled system in the segregated manner. Iterative methods are adopted to solve the sparse linear equation systems as described in Section 7.5. A sufficiently improved solution of the inner equation system can often be achieved in just a few inner iterations per outer iteration.

The algorithm is implemented in parallel using the MPI protocol [31] in a distributed memory concept. The domain decomposition is carried out with the aid of the ParMETIS [72] library package.

Under-Relaxation

To obtain a converged solution of the coupled system of equations in a segregated pressure-correction procedure, the rate of change of the individual variables ϕ is reduced by under-relaxation. Only a fraction $0 < \alpha_\phi < 1$ of the predicted variable change, $\Delta\phi = \phi^{m*} - \phi^{m-1}$, is applied in the current iteration loop m :

$$\phi^m = \phi^{m-1} + \alpha_\phi (\phi^{m*} - \phi^{m-1}) . \quad (7.79)$$

Instead of using ϕ^{m*} , the algorithm proceeds with the under-relaxed, or damped value ϕ^m . Eqn. (7.79) is used implicitly, substituting

$$\phi^{m*} = \phi^{m-1} + (\phi^m - \phi^{m-1}) / \alpha_\phi \quad (7.80)$$

for ϕ^m into Eqn. (7.23). Discarding the iteration counter yields

$$A_{ii}^* \phi_i + \sum_{j \in N(i)} A_{ij} \phi_j = S_i^* , \quad i = 1, \dots, n . \quad (7.81)$$

The under-relaxed central coefficient and source term are

$$A_{ii}^* = A_{ii} / \alpha_\phi \quad \text{and} \quad S_i^* = S_i - \left(1 - \frac{1}{\alpha_\phi}\right) A_{ii} \phi_i^{m-1} . \quad (7.82)$$

The neighbour coefficients are not affected, so that the diagonal dominance of the relaxed equation system is increased when the under-relaxation factor α_ϕ is reduced. The stabilisation is achieved by reducing the rate of convergence. The under-relaxation does not influence the converged solution since $\Delta\phi$ tends to zero eventually.

Solution of Linear Equation Systems

The set of algebraic equations for momentum, pressure correction, turbulence, etc. is iteratively driven to convergence as described in Section 7.5. The algebraic, integral residual \bar{R} for the individual scalar fields ϕ , associated with the steps (i) to (v), is calculated according to the L_1 -norm

$$\bar{R}^k = \frac{1}{\bar{R}^1} \sum_{i=1}^n R_i^k \quad \text{with} \quad R_i^k = A_{ii}^k \phi_i^{k-1} + \sum_{j \in N(i)} A_{ij}^k \phi_j^{k-1} - S_i^k . \quad (7.83)$$

The integral residual of iteration k is normalised with respect to the integral residual of the first iteration \bar{R}^1 . A_{ij}^k and S_i^k denote the components of the sparse system matrix and the right-hand side vector of iteration k . Alternatively, the L_2 - or root mean-square norm can be used. By default, the maximum value of the individual residuals for U_i , p , k , ε or ω is used to evaluate the convergence process.

The PETSc toolkit [4] was used to precondition and solve the linearised equation systems. In the parallel computations, the Jacobi preconditioner was used. Different iterative

solution procedures were employed depending on the properties of the system matrix. The symmetric equation system of the pressure-correction equation was mainly solved by the conjugate gradient (CG) method. For the non-symmetric matrices associated with the equations of momentum, heat, turbulence, etc., the generalised minimal residual (GMRES) and biconjugate gradient methods (BiCG or BiCGSTAB) were used.

7.6. Objective Functionals

For steady-state problems, the objective functional value is evaluated at the end of the primal simulation. The objective functionals are approximated consistently with the discretisation schemes used for the flow equations, which is particularly important for the following adjoint analysis. The force criterion (4.34), for example, is evaluated from the momentum flux through a wall face, $\underline{F} = \underline{F}^p + \underline{F}^d$, consistently with the discretisation of the boundary conditions in the momentum equations (7.58) or (7.68):

$$J_\Gamma = -d_i^* \int_{\Gamma_O} \pi_{ij} d\Gamma_j \approx d_i^* \sum_{PB \in \Gamma_O} [F_i^d + F_i^p]. \quad (7.84)$$

$PB \in \Gamma_O$ denotes the set of boundary patches on Γ_O carrying the objective functional. The implementation is in line with the method of sections, so that integral conservation of momentum is ensured.

7.7. Navier–Stokes Variation

In general, the variation of the Navier–Stokes equations is discretised and solved like the non-linear Navier–Stokes or RANS equations. Three specific features are discussed below, regarding

- the full linearisation of the non-linear convection term;
- the variation of the boundary conditions on the design surfaces Γ_D ; and
- the evaluation of the variation of the objective functional on Γ_O .

Convection and Advection Terms

It is assumed that the shape variations according to Eqn. (5.24) are confined to the design surfaces Γ_D . The variation of the convection term is derived from the chain rule of differentiation, viz.

$$\delta \int_{\Delta\Omega} \rho U_j \frac{\partial U_i}{\partial x_j} d\Omega = \rho \int_{\Delta\Omega} \left(U_j \frac{\partial \delta U_i}{\partial x_j} + \delta U_j \frac{\partial U_i}{\partial x_j} \right) d\Omega. \quad (7.85)$$

Two terms arise from the linearisation of the non-linear convective term in the momentum equations, denoted linear *convection* and *advection*¹, being the first and the second right-

¹The naming convection “advection” is arbitrary. It has no deeper meaning or bearing on particular conventions used elsewhere.

hand side terms of Eqn. (7.85), respectively. The linearised convection is equivalent to the Picard-linearised convection term in the non-linear momentum equations². Hence, the discretisation schemes described in Section 7.1 are applicable. An explicit, volume-based approximation of the advection term is considered on the right-hand side of the linearised momentum equations:

$$\delta S_i = - \int_{\Delta\Omega} \rho \delta U_j \frac{\partial U_i}{\partial x_j} d\Omega \approx - \left[\rho \delta U_j \frac{\partial U_i}{\partial x_j} \right]_P \Delta\Omega. \quad (7.86)$$

Alternatively, a face-based approximation can be derived for the advection term from the discretisation of the non-linear system. A linearisation of the discrete convective flux of momentum through a CV face gives

$$\begin{aligned} \delta F_i^c &\approx \rho \Delta\Gamma_j (U_j \delta U_i + \delta U_j U_i)_F \\ &\approx \dot{m} \delta U_i|_F + \delta \dot{m} U_i|_F \end{aligned} \quad (7.87)$$

with

$$\delta \dot{m} \approx \rho \Delta\Gamma_j [\lambda \delta U_j|_N + (1 - \lambda) \delta U_j|_P]. \quad (7.88)$$

The linearised convection term, which is the first right-hand side term of Eqn. (7.87), is treated as the convection term of the non-linear problem. The second right-hand side contribution, being the advection term, represents the variation of the mass fluxes, which is neglected in the Picard iteration scheme. The face value \underline{U}_F is interpolated consistently with the primal convection scheme (UDS, CDS or MUSCL-based according to Section 7.1). Note that the off-diagonal sub-matrices, $\underline{A}_{ij}|_{i \neq j}$, of the variation of the matrix-vector problem (7.26) are non-zero unless the advection terms are neglected.

Boundary Conditions on Γ_D

Whereas the boundary conditions on Γ_D are linearly developed to the new position in the continuous approach, cf. Eqn. (6.29), the discrete, primal boundary fluxes through no-slip walls on Γ_D are derived here with respect to both flow and position. As illustrated in Figure 7.11, it is assumed that only the position of the surface node B undergoes a variation ($\underline{x}_{B'} = \underline{x}_B + \delta \underline{x}_B$) and P is fixed ($\underline{x}_P = \underline{x}_{P'}$ or $\delta \underline{x}_P = \underline{0}$). In that way, the discrete approach corresponds to the boundary-based gradient evaluation (6.39) of the continuous adjoint analysis. Hence, a movement of the boundary implies a change of the wall distance of the cell centre next to the design surface Γ_D :

$$\delta y_P = y_{P'} - y_{B'} - (y_P - y_B) = -y_{B'} = -\delta y_B = \underline{n} \cdot \delta \underline{x}_B = \delta n_B. \quad (7.89)$$

A **low-Re formulation** is considered first, which also holds for laminar flow when the effective viscosity μ_{eff} is substituted by the molecular viscosity μ . Since the boundary value

²An incomplete linearisation of the Navier–Stokes convection term yields the so-called Oseen equations.

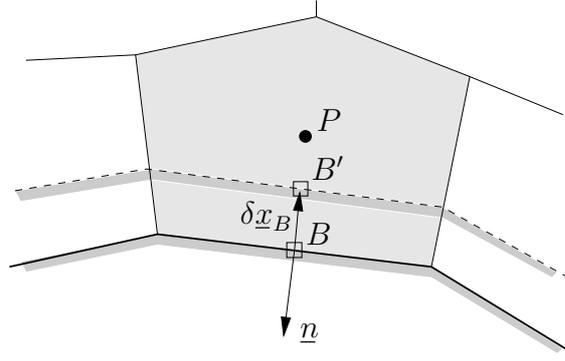


Figure 7.11.: Perturbation of boundary mesh by $\delta n_B = \underline{n} \cdot \delta \underline{x}_B$

for the velocity \underline{U}_B is prescribed, the corresponding variation $\delta \underline{U}_B$ is zero. A zero pressure gradient, $0 = \underline{n} \cdot \underline{\nabla} p|_{\text{wall}}$, is assumed, i.e. $p_P \approx p_B$. The boundary flux of momentum

$$\begin{aligned} F_i|_{\text{wall}} &\approx p|_P \Delta \Gamma_i - \tau_i^w \Delta \Gamma \\ &\approx p|_P \Delta \Gamma_i - \mu_{\text{eff}} T_{ij} y_P^{-1} (U_j|_B - U_j|_P) \Delta \Gamma \end{aligned} \quad (7.90)$$

experiences a variation with respect to y , $p|_P$ and $U_i|_P$:

$$\begin{aligned} \delta F_i|_{\text{wall}} &\approx \delta p|_P \Delta \Gamma_i - \delta \tau_i^w \Delta \Gamma \\ &\approx \delta p|_P \Delta \Gamma_i - \mu_{\text{eff}} T_{ij} y_P^{-1} [\delta y_B y_P^{-1} (U_j|_B - U_j|_P) - \delta U_j|_P] \Delta \Gamma. \end{aligned} \quad (7.91)$$

Viscous normal stresses are avoided in the discretisation (7.91) enforcing $\partial U^n / \partial n|_{\text{wall}} = 0$ by virtue of the projection tensor $T_{ij} = \delta_{ij} - n_i n_j$.

Alternatively, the boundary layer flow can be modelled according to the **logarithmic law of the wall (high-Re)** (7.64). The high-Re analysis is carried out on the discrete level alone since it is difficult to argue in a continuous approach if not boundary values but the values in the adjacent CV-centre are considered. The total variation of the velocity profile (7.64) with respect to the velocity $U_P^t = U^t(y_P)$, the wall distance $y_P = -n_P$ and the shear velocity U^τ is

$$-\delta U^t|_P = \delta U^\tau \frac{1}{\kappa} [\ln(E y_P^+) + 1] + \delta y_P \left[\frac{U^\tau}{\kappa y_P} \right]. \quad (7.92)$$

The negative sign on the left-hand side of Eqn. (7.92) is in line with Eqn. (7.64). The summand “1” in the first bracketed term of (7.92) results from a change of y_P^+ with respect to the shear velocity. The variation of the viscous flux of momentum through the boundary is determined from

$$\delta \tau^w = 2\rho U^\tau \delta U^\tau = -\frac{2\rho \kappa U^\tau}{\ln(E y_P^+) + 1} \left[\delta U_P^t + \delta y_P \frac{U^\tau}{\kappa y_P} \right]. \quad (7.93)$$

Eqn. (7.93) is rewritten in tensor notation to be used in combination with the Navier–Stokes equations. For the sake of brevity the following conventions are declared

$$U_i^\tau = U^\tau t_i \quad \text{and} \quad \tau_i^w = \tau^w t_i. \quad (7.94)$$

The vector-valued equivalent of Eqn. (7.92) is

$$-T_{ij} \delta U_j|_P = \delta U_j^\tau \frac{1}{\kappa} [\delta_{ij} \ln(Ey_P^+) + t_i t_j] + \delta y_P \left[\frac{U_i^\tau}{\kappa y_P} \right]. \quad (7.95)$$

Let \underline{t}^δ denote the surface unit vector associated with the direction of the variation of the surface shear stress: $\underline{t}^\delta = \underline{T} \cdot \Delta \delta \underline{U} / |\underline{T} \cdot \Delta \delta \underline{U}|$ with $\Delta \delta \underline{U} = \delta \underline{U}_B - \delta \underline{U}_P$. The multiplier of the variation of the shear velocity $\delta \underline{U}^\tau$ in Eqn. (7.95)

$$t_j^\delta [\delta_{ij} \ln(Ey_P^+) + t_i t_j] = t_i^\delta \ln(Ey_P^+) + t_i t_j^\delta t_j \quad (7.96)$$

is dominated by the logarithmic expression

$$\frac{\ln(Ey_P^+)}{t_j t_j^\delta} > 5.7 \quad \text{for } y_P^+ > 30. \quad (7.97)$$

Thus, a reduced version of (7.95) is used, wherein the variation of y_P^+ with respect to the shear velocity is neglected

$$-T_{ij} \delta U_j|_P \approx \delta U_i^\tau \left[\frac{\ln(Ey_P^+)}{\kappa} \right] + \delta y_P \left[\frac{U_i^\tau}{\kappa y_P} \right]. \quad (7.98)$$

With the nomenclature illustrated in Figure 7.12 and $\cos \gamma = \underline{t} \cdot \underline{t}^\delta$, the simplification is defensible looking at the distributions of the angle error

$$\varepsilon_\gamma(\gamma) = \begin{cases} \arctan \left(\frac{\cos \gamma \sin \gamma}{\ln(Ey^+) + \cos^2 \gamma} \right), & -\frac{\pi}{2} < \gamma < \frac{\pi}{2} \\ \arctan \left(\frac{\cos(\pi - \gamma) \sin(\pi - \gamma)}{\ln(Ey^+) - \cos^2(\pi - \gamma)} \right), & \frac{\pi}{2} < \gamma < \frac{3}{2}\pi \end{cases} \quad (7.99)$$

and the associated relative error in the vector norm (amplitude error)

$$\varepsilon_A(\gamma) = \begin{cases} 1 - \ln(Ey^+) \left([\ln(Ey^+) + \cos^2 \gamma]^2 + [\sin \gamma \cos \gamma]^2 \right)^{-\frac{1}{2}}, & -\frac{\pi}{2} < \gamma < \frac{\pi}{2} \\ 1 - \ln(Ey^+) \left([\ln(Ey^+) - \cos^2(\pi - \gamma)]^2 + [\sin(\pi - \gamma) \cos(\pi - \gamma)]^2 \right)^{-\frac{1}{2}}, & \frac{\pi}{2} < \gamma < \frac{3}{2}\pi. \end{cases} \quad (7.100)$$

The corresponding plots of the angle error ε_γ and the relative error in the vector norm ε_A over γ are provided in Figure 7.13. The errors are highest for small y^+ -values. For $y^+ = 30$, ε_A is below 22 per cent with the maximum amplitude at $\gamma = \pi$. ε_γ does not exceed 0.031π . Note that ε_A and ε_γ are not in phase.

Substituting the reduced formulation (7.98) into

$$\delta \tau_i^w = \rho \delta(U^\tau U_i^\tau) = \rho \delta U_j^\tau U^\tau (\delta_{ij} + t_i t_j), \quad (7.101)$$

the tensor-equivalent to Eqn. (7.93) is obtained:

$$\delta \tau_i^w \approx -\frac{\rho U^\tau \kappa}{\ln(Ey_P^+)} \left[(T_{ij} + t_i t_j) \delta U_j|_P + 2 \delta y_P \frac{U_i^\tau}{\kappa y_P} \right]. \quad (7.102)$$

Using the reduced expression (7.98) instead of Eqn. (7.95), a matrix inversion of $(\underline{I} \ln(Ey_P^+) + \underline{t} \underline{t})$ is avoided. Accordingly, the variation of the momentum flux through the boundary is

$$\begin{aligned} \delta F_i|_{\text{wall}} &\approx \delta p|_P \Delta \Gamma_i - \delta \tau_i^w \Delta \Gamma \\ &\approx \delta p|_P \Delta \Gamma_i + \frac{\rho U^\tau \kappa}{\ln(Ey_P^+)} \left[(T_{ij} + t_i t_j) \delta U_j(y_P) + 2 \delta y_P \frac{U_i^\tau}{\kappa y_P} \right] \Delta \Gamma. \end{aligned} \quad (7.103)$$

Objective Functional on $\Gamma_O \cap \Gamma_D$

When a force objective is declared on $\Gamma_O \cap \Gamma_D$, the description presented in the previous section is consistently applied to the variation of the objective functional, viz.

$$\delta J = d_i^* \sum_{PB \in \Gamma_O} \delta F_i \approx d_i^* \sum_{PB \in \Gamma_O} [\delta p_P \Delta \Gamma_i - \delta \tau_i^w \Delta \Gamma]. \quad (7.104)$$

The boundary-flux variation $\delta \underline{F}_{PB}$ is summed up over the boundary wall faces in the set PB lying on Γ_O . Depending on the wall treatment, either the low-Re formulation (7.91) or the high-Re description (7.103) are used to calculate the variation of the viscous portion of the force objective.

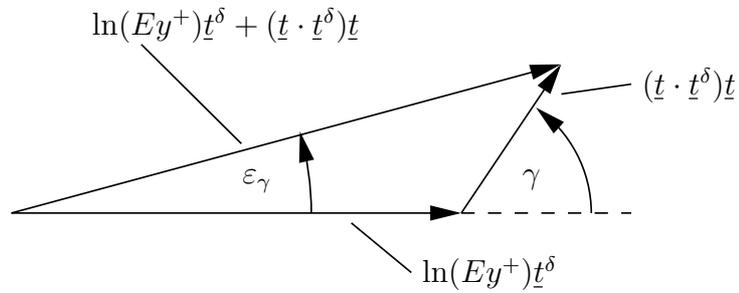


Figure 7.12.: Definition of vectors on design surface Γ_D

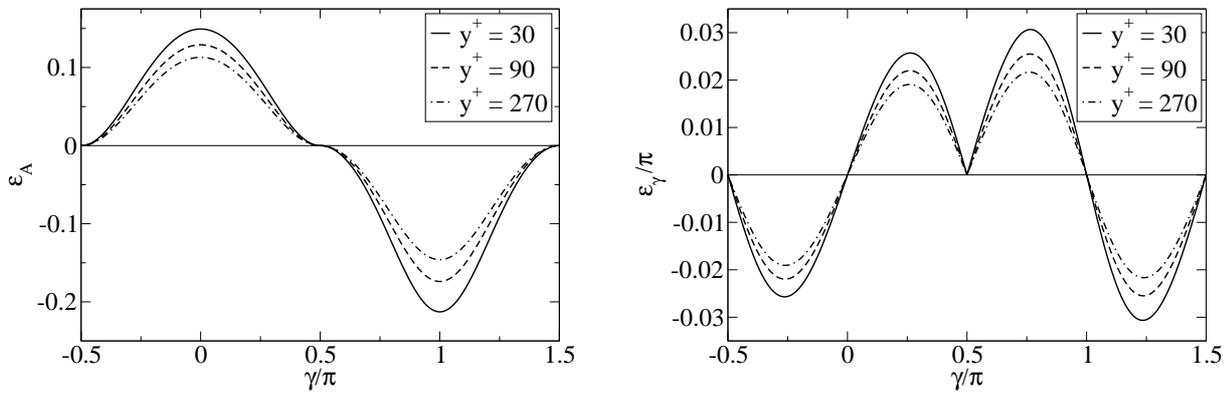


Figure 7.13.: Amplitude error ε_A (left) and angle error ε_γ (right) at $y^+ = 30, 90$ and 270 if the variation of y^+ is neglected

8. Adjoint Discretisation

“The discrete and continuous formulations of the adjoint problem clearly represent two very different ways of doing essentially the same thing [...]” Peter and Dwight [116]

The adjoint Navier–Stokes module was planned and set up as an extension to the existing general-purpose RANS code FreSCo⁺. Due to the similarity of the primal and the adjoint PDE derived in Chapters 5 and 6, many features of the flow solver concerning discretisation and implementation are like-wise apt for the adjoint module. However, some aspects have to be interpreted in a special way. The subsequent properties are considered particularly important for the development of the adjoint RANS module:

- (i) accuracy and adjoint consistency;
- (ii) numerical robustness and stability;
- (iii) maintainability, flexibility and user-friendliness;
- (iv) numerical efficiency and performance.

These issues are linked to each other in various ways as discussed in the following.

The adjoint PDE devised in Chapter 6 are the starting point for the analysis. In conjunction with the continuous adjoint method, adjoint consistency (i) is not a matter of course but must be ensured with due caution. A so-called hybrid adjoint strategy was pursued in this work to derive consistent discretisation schemes for the individual adjoint PDE terms from the corresponding primal schemes via summation by parts (cf. Reuther et al. [126], Nadarajah [101]). Based on the primal discretisation, a unified, discrete description of the adjoint boundary conditions and the sensitivity equation is presented that supports both low- and high-Reynolds number boundary formulations. The analysis incorporates consistent definitions of the primal boundary conditions, the primal objective functional and the discrete shape variation.

The issue of adjoint consistency is also linked to the numerical robustness of the adjoint solver (ii). The discretisation schemes for the primal system have been a subject of research over the past decades. Since no comparable experience is available for the adjoint problem, it is obvious to derive the adjoint schemes from the primal ones, so that the adjoint discretisation is a valid approximation to the adjoint PDE (i). Ideally, both the discrete linearised and adjoint systems have the same matrix properties in terms of diagonal dominance, sparseness, etc (ii). When so-called Picard iterations are used [e.g. 30], the complete linearisation is not available in the non-linear implementation and must be supplemented in the adjoint code.

A lean adjoint implementation is achieved by re-using large parts of the primal FV discretisation presented in the previous chapter. Code-maintenance (iii) is facilitated by avoiding redundant adjoint subroutines. Accordingly, the self-adjoint operators of the nonlinear flow solver are shared by the adjoint code. The underlying data-structure is re-employed by the adjoint as, for example, regards the unstructured book-keeping and the concept of parallelisation. A user-coding strategy was installed to provide a high level of flexibility (iii) so that the adjoint boundary conditions and source terms can be imposed for the objective functional of interest. This is necessary to use one adjoint body for different optimisation problems.

The flexibility of the continuous adjoint method is traded for the accuracy of the discrete adjoint method (Sections 2.3 and 2.4). The flexibility was mainly exploited for the implementation of the adjoint pressure-correction scheme, which is an adapted SIMPLE algorithm. It was accepted that the succession of operations in the forward pressure-correction algorithms (linear and non-linear) is not exactly reversed in the adjoint implementation, so that only converged solutions can be compared. The pressure-velocity coupling is expected to have a high influence on the numerical efficiency (iv) of the adjoint module. The treatment of the additional cross-coupling of the adjoint equations (Chapter 6) is an inherent weak point of a segregated adjoint solution scheme. However, a huge portion of the analytical pressure equation is already neglected in the SIMPLE algorithm, so that one could hope that the incomplete scheme can also cope with the densely-coupled adjoint problem.

This chapter aims to highlight the central features of the adjoint discretisation. The elementary adjoint schemes are derived in Section 8.1 for a scalar transport equation. The adjoint schemes are subsequently extended to the Navier–Stokes equations, Sections 8.2 et seqq., based on a term-by-term analysis of the primal discretisation schemes. The adjoint pressure-correction scheme is described in Section 8.3. It is followed by the analysis of the adjoint boundary terms, Section 8.4, and the gradient equation in Section 8.5 for both low- and high-Re wall boundary conditions. This part closes with Section 8.6 where the adjoint algorithm is outlined and selected implementation features are discussed.

8.1. Scalar Transport Equation

The scalar transport equation (7.2) is considered first. The flux contributions to the variation of the Lagrangian are computed by summing over all internal and boundary face-flux variations multiplied by the adjoint variables. In a compact discretisation scheme each internal face PN contributes to the balances of the neighbouring CVs P and N , so that the internal face-flux variations are weighted by the adjoint multipliers $\hat{\phi}_P$ and $\hat{\phi}_N$, respectively. The boundary-faces PB only contribute to the balance of the adjacent CV P , viz.

$$\delta L = \dots + \sum_{PN} \delta F_{PN} \cdot (\hat{\phi}_P - \hat{\phi}_N) + \sum_{PB} \delta F_{PB} \cdot \hat{\phi}_P + \dots \quad (8.1)$$

In the following, the adjoint pseudo-fluxes \hat{F} are obtained via summation by parts of the individual contributions to Eqn. (8.1) to give the adjoint balance for each CV P

$$\sum_{N(P)} \hat{F}_{PN} + \sum_{B(P)} \hat{F}_{PB} = \hat{S}, \quad (8.2)$$

wherein \hat{S} is the adjoint source term, which for example hosts contributions from the objective functional. To keep the presentation concise, the subsequent analysis is confined to internal cell faces PN .

Convection

A first-order **upstream discretisation** (UDS)

$$\delta F_{PN}^c = \dot{m}_{PN} \delta \phi_F \approx \max[\dot{m}_{PN}, 0] \delta \phi_P + \min[\dot{m}_{PN}, 0] \delta \phi_N \quad (8.3)$$

makes the following internal-face contributions to the variation of the Lagrangian

$$\begin{aligned} \delta L &= \dots + \sum_{PN} (\max[\dot{m}_{PN}, 0] \delta \phi_P + \min[\dot{m}_{PN}, 0] \delta \phi_N) \cdot (\hat{\phi}_P - \hat{\phi}_N) + \dots \\ &= \dots + \sum_{PN} \left(-\min[\dot{m}_{PN}, 0] \hat{\phi}_P - \max[\dot{m}_{PN}, 0] \hat{\phi}_N \right) \cdot (\delta \phi_P - \delta \phi_N) + \dots \end{aligned} \quad (8.4)$$

by virtue of the continuity-balance for the internal CV P

$$0 = \sum_{N(P)} (\max[\dot{m}_{PN}, 0] + \min[\dot{m}_{PN}, 0]) . \quad (8.5)$$

The adjoint pseudo-flux for face PN is identified from Eqn. (8.4):

$$\hat{F}_{PN}^c = -\min[\dot{m}_{PN}, 0] \hat{\phi}_P - \max[\dot{m}_{PN}, 0] \hat{\phi}_N . \quad (8.6)$$

This is a first-order downstream (DDS) approximation to the negative convective fluxes in the adjoint equations (6.5). Since the negative DDS-operator is the transposed of the primal UD scheme, the discrete adjoint identity (2.42) is satisfied and an adjoint-consistent scheme is obtained.

For the **central differencing scheme** (CDS)

$$\delta F_{PN}^c = \dot{m}_{PN} \delta \phi_F \approx \dot{m}_{PN} ([1 - \lambda_{PN}] \delta \phi_P + \lambda_{PN} \delta \phi_N) , \quad (8.7)$$

the flux contributions to the Lagrangian read

$$\begin{aligned} \delta L &= \dots + \sum_{PN} \dot{m}_{PN} ([1 - \lambda_{PN}] \delta \phi_P + \lambda_{PN} \delta \phi_N) \cdot (\hat{\phi}_P - \hat{\phi}_N) + \dots \\ &= \dots - \sum_{PN} \dot{m}_{PN} \left(\lambda_{PN} \hat{\phi}_P + [1 - \lambda_{PN}] \hat{\phi}_N \right) \cdot (\delta \phi_P - \delta \phi_N) + \dots \end{aligned} \quad (8.8)$$

According to Eqn. (8.8), the corresponding dual fluxes are

$$\begin{aligned}\hat{F}_{PN}^c &= -\dot{m}_{PN} \left(\lambda_{PN} \hat{\phi}_P + [1 - \lambda_{PN}] \hat{\phi}_N \right) \\ &= -\dot{m}_{PN} \left([1 - \hat{\lambda}_{PN}] \hat{\phi}_P + \hat{\lambda}_{PN} \hat{\phi}_N \right).\end{aligned}\quad (8.9)$$

Note that the primal and the (negative) dual face-interpolation schemes are only equivalent on equidistant grids, i.e. $\lambda_{PN} = \hat{\lambda}_{PN} = 0.5$. Like in the flow solver the popular **UDS-CDS blending** with $0 < \beta_\phi < 1$

$$F_{PN}^c = \beta_\phi [F_{PN}^c]^{\text{CDS}} + (1 - \beta_\phi) [F_{PN}^c]^{\text{UDS}}, \quad (8.10)$$

can also be applied to the adjoint discretisation:

$$\hat{F}_{PN}^c = \beta_\phi [\hat{F}_{PN}^c]^{\text{CDS}} + (1 - \beta_\phi) [\hat{F}_{PN}^c]^{\text{DDS}}. \quad (8.11)$$

The dual of the **MUSCL-scheme** (7.13) is used to calculate the adjoint face-flux $\hat{F}_{PN} = [-\dot{m} \hat{\phi}_F]_{PN}$. According to the notation illustrated in Figure 8.1, the adjoint MUSCL-scheme for face interpolation of $\hat{\phi}_F$ is downstream biased and obtained via summation by parts

$$\hat{\phi}_F \approx \hat{\phi}_D + \frac{1}{4} \left[(1 + \kappa) \left(\hat{\phi}_U - \hat{\phi}_D \right) \psi(\hat{r}) + (1 - \kappa) \left(\hat{\phi}_D - \hat{\phi}_{DD} \right) \psi(\hat{r}^{-1}) \right], \quad (8.12)$$

with

$$\hat{r} = \frac{\hat{\phi}_D - \hat{\phi}_{DD}}{\hat{\phi}_U - \hat{\phi}_D} \quad (8.13)$$

and the primal definitions (7.16) through (7.18). The resulting adjoint schemes depend on the choice of κ . These are compiled in Table 8.1 with “U” (“UU”) for (remote) upstream replaced by “D” (“DD”) for (remote) downstream. For simplicity, the non-linear limiting scheme is directly applied to the adjoint problem instead of a transposition of the completely-linearised primal scheme. The equidistant description of the proposed “reverse” formulation (8.12) can directly be applied to the unstructured, pseudo-equidistant formulation described in Section 7.1, wherein the variable values in the remote nodes *UU* or *DD* are explicitly reconstructed based on the gradient of ϕ or $\hat{\phi}$.

Diffusion

With the diffusive flux through face *PN*

$$\delta F_{PN}^d \approx -\frac{\gamma_\phi \Delta \Gamma}{|\underline{x}_N - \underline{x}_P|} (\delta \phi_N - \delta \phi_P), \quad (8.14)$$

the dual discretisation scheme for the adjoint diffusion operator is obtained from the corresponding flux-contributions to the variation of the Lagrangian

$$\delta L = \dots - \sum_{PN} \frac{\gamma_\phi \Delta \Gamma}{|\underline{x}_N - \underline{x}_P|} (\delta \phi_N - \delta \phi_P) \cdot (\hat{\phi}_P - \hat{\phi}_N) + \dots \quad (8.15)$$

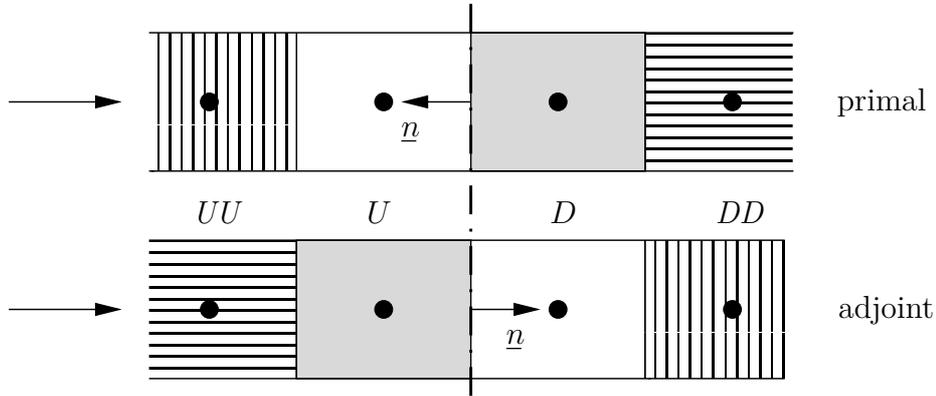


Figure 8.1.: 1D illustration of primal and adjoint convection schemes: transposition or “mirroring” leads to the adjoint downstream scheme in the balance of the CV shaded in the bottom row, which is the counterpart of the corresponding primal upstream scheme in the balance of the CV shaded in the top row.

Table 8.1.: Comparison of primal and adjoint convection schemes in a unified formulation

scheme	κ	primal deferred correction	adjoint deferred correction
CDS	1	$(\phi_D - \phi_U)/2$	$(\hat{\phi}_U - \hat{\phi}_D)/2$
LU(D)DS	-1	$(\phi_U - \phi_{UU})/2$	$(\hat{\phi}_D - \hat{\phi}_{DD})/2$
QU(D)ICK	1/2	$(3\phi_D - 2\phi_U - \phi_{UU})/8$	$(3\hat{\phi}_U - 2\hat{\phi}_D - \hat{\phi}_{DD})/8$
CU(D)I	1/3	$(2\phi_D - \phi_U - \phi_{UU})/6$	$(2\hat{\phi}_U - \hat{\phi}_D - \hat{\phi}_{DD})/6$

The corresponding adjoint pseudo-fluxes through face PN are:

$$\hat{F}_{PN}^d = -\frac{\gamma_\phi \Delta\Gamma}{|\underline{x}_N - \underline{x}_P|} \left(\hat{\phi}_N - \hat{\phi}_P \right). \quad (8.16)$$

Eqn. (8.16) verifies the well-known symmetry property of the diffusive flux operator, i.e. the discrete diffusion operator is self-adjoint. The primal non-orthogonality correction (7.22) is directly reapplied in the adjoint algorithm. The implementation is not strictly symmetric on non-equidistant, unstructured grids due to the evaluation of the gradient $\underline{\nabla}\phi$ and its CV-face interpolation.

8.2. Momentum Equations

The subsequent description based on the Navier–Stokes equations is face-oriented and supports unstructured polyhedral grids. The complete variation of the discrete Lagrangian with respect to the discrete state and control variables is

$$\delta L = (\delta_{\underline{U}} + \delta_p + \delta_y) L. \quad (8.17)$$

The partial variation with respect to the flow, $(\delta_{\underline{U}} + \delta_p)(\cdot)$, is considered in the discrete analysis of the adjoint momentum equations and the pressure-projection scheme. Subsequently, the discretisation schemes for the adjoint boundary conditions on Γ_D and the boundary-based gradient expressions are derived from the partial variation with respect to the geometry $\delta_y(\cdot)$.

The transposed system matrix of the adjoint momentum-continuity problem is obtained from the primal system (7.26) via summation by parts

$$\begin{pmatrix} \underline{\underline{A}}^T & \underline{\underline{C}}^T \\ \underline{\underline{B}}^T & \underline{\underline{0}} \end{pmatrix} \cdot \begin{pmatrix} \underline{\hat{U}} \\ \underline{\hat{p}} \end{pmatrix} = \begin{pmatrix} \underline{\hat{S}} \\ \underline{\hat{Q}} \end{pmatrix}, \quad (8.18)$$

with the adjoint variable and right-hand side vectors

$$\underline{\hat{U}} = \begin{pmatrix} \underline{\hat{U}}_1 \\ \underline{\hat{U}}_2 \\ \underline{\hat{U}}_3 \end{pmatrix}; \quad \underline{\hat{S}} = \begin{pmatrix} \underline{\hat{S}}_1 \\ \underline{\hat{S}}_2 \\ \underline{\hat{S}}_3 \end{pmatrix}.$$

Note that the off-diagonal sub-matrices, $\underline{\underline{A}}_{ij}|_{i \neq j}$, of the full variation of the matrix-vector problem (7.26) are zero since the advection terms are treated explicitly in $\underline{\hat{S}}_i$.

The discretisation schemes for the convection, diffusion and pressure operators of the adjoint momentum equations are derived from the primal schemes in the following in a term-by-term analysis.

Convection

Two terms arise from the linearisation of the non-linear convective term in the momentum equations, denoted convection and advection. The contributions to the variation of the Lagrangian are

$$\begin{aligned}\delta L &= \dots + \sum_P \rho \int_{\Delta\Omega_P} \hat{U}_i \delta \left(U_j \frac{\partial U_i}{\partial x_j} \right) d\Omega + \dots \\ &\approx \dots + \sum_P \left[\rho \hat{U}_i \left(U_j \frac{\partial \delta U_i}{\partial x_j} + \delta U_j \frac{\partial U_i}{\partial x_j} \right) \Delta\Omega \right]_P + \dots\end{aligned}\quad (8.19)$$

The linearised convection term being the first inner summand in Eqn. (8.19) is equivalent to the convection term of the scalar transport problem. Within the FV approach, the corresponding convective flux of adjoint momentum is

$$\hat{F}^c \approx -\dot{m} \hat{U}_F, \quad (8.20)$$

using the “transposed” convection schemes described in Section 8.1 for cell-face interpolation of \hat{U}_F .

Advection

The advection term, being the second inner summand in Eqn. (8.19), does not have a counterpart in the primal discretisation if an incomplete Picard linearisation is applied.

A straight-forward, **volume-based approximation (ADV1)** obtained from the continuous adjoint calculus leads to the underlined expression, which is treated explicitly in the adjoint momentum equations:

$$\begin{aligned}\delta L &= \dots + \sum_P \int_{\Delta\Omega_P} \hat{U}_i \rho \delta U_j \frac{\partial U_i}{\partial x_j} d\Omega + \dots \\ &\approx \dots + \sum_P \left[\delta U_i \rho \hat{U}_j \frac{\partial U_j}{\partial x_i} \Delta\Omega \right]_P + \dots\end{aligned}\quad (8.21)$$

An **alternative approximation (ADV2)** underlined below is obtained from the continuous adjoint analysis via integration by parts (see A.3):

$$\begin{aligned}\delta L &= \dots - \sum_P \int_{\Delta\Omega_P} \delta U_i \rho U_j \frac{\partial \hat{U}_j}{\partial x_i} d\Omega + \sum_{PB} \int_{\Gamma_{PB}} \delta U_i \rho U_j \hat{U}_j d\Gamma_i + \dots \\ &\approx \dots - \sum_P \left[\delta U_i \rho U_j \frac{\partial \hat{U}_j}{\partial x_i} \Delta\Omega \right]_P + \sum_{PB} \left[\delta U_i \rho U_j \hat{U}_j \Delta\Gamma_i \right] + \dots\end{aligned}\quad (8.22)$$

A third, **face-based approximation (ADV3)** can be derived from the primal discretisation. A linearisation of the discrete convective flux of momentum through a cell face, Eqn. (7.31), gives

$$\delta F_i^c|_{PN} \approx \dot{m}_{PN} \delta U_i|_F + \delta \dot{m}_{PN} U_i|_F \quad (8.23)$$

with

$$\delta m_{PN} \approx \rho \Delta \Gamma_j|_{PN} [\lambda_{PN} \delta U_j|_N + (1 - \lambda_{PN}) \delta U_j|_P]. \quad (8.24)$$

The first right-hand side term of Eqn. (8.23) leads to the adjoint convection term discussed above. The second right-hand side term makes the inner-face contributions to the variation of the discrete Lagrangian

$$\delta L \approx \dots + \sum_{PN} \left[\rho (\lambda_{PN} \delta U_j|_N + [1 - \lambda_{PN}] \delta U_j|_P) n_j|_{PN} U_i|_F \Delta \Gamma \left(\hat{U}_i|_P - \hat{U}_i|_N \right) \right] + \dots \quad (8.25)$$

Expression (8.25) is rearranged

$$\delta L \approx \dots + \sum_{PN} \left(\delta U_i|_P \left[\rho (1 - \lambda_{PN}) n_i|_{PN} U_j|_F \Delta \Gamma \left(\hat{U}_j|_P - \hat{U}_j|_N \right) \right] \right) \quad (8.26a)$$

$$+ \delta U_i|_N \left[\rho (1 - \lambda_{NP}) n_i|_{NP} U_j|_F \Delta \Gamma \left(\hat{U}_j|_N - \hat{U}_j|_P \right) \right] + \dots \quad (8.26b)$$

Note the negative orientation of the face unit-vector in CV N , i.e. $\underline{n}_{NP} = -\underline{n}_{PN}$. With $\hat{\lambda}_{PN} = 1 - \lambda_{PN}$, the advective face flux is obtained from

$$\hat{F}_{PN}^a \approx -\rho \hat{\lambda}_{PN} \underline{U}_F \cdot \left(\hat{\underline{U}}_N - \hat{\underline{U}}_P \right) \Delta \Gamma_{PN}. \quad (8.27)$$

A consistent implementation is achieved if the face value \underline{U}_F is interpolated by the primal convection scheme (UDS, CDS, MUSCL, etc.) using the primal transport direction. The face under consideration contributes the bracketed term in (8.26a) to the balance of adjoint momentum of CV P and the bracketed term in (8.26b) to the balance of CV N . Due to the analogy of integration and summation by parts, the advection formulations ADV2 and ADV3 are equivalent on a Cartesian, equidistant grid in conjunction with a CDS convection scheme as shown in Appendix B.3. The equivalence is not generally true on unstructured grids. When the transport equations are solved one by one in a segregated pressure-projection scheme, only one out of three summands of the inner product $[\underline{U} \cdot \hat{\underline{U}}]_F$ can be considered implicitly per adjoint momentum equation. The advection term was treated explicitly in this study. Boundary faces featuring a zero-variation of the convective flux $\delta \underline{F}^c = \underline{0}$, such as wall-, inlet- or symmetry-faces, neither have convective nor advective fluxes in the adjoint discretisation. To calculate the advective flux through the outflow, the primal velocity is extrapolated using the upstream velocity value:

$$\hat{\underline{F}}^a|_{\text{out}} \approx \rho \Delta \Gamma_{PB} \hat{\underline{U}}_P \cdot \underline{U}_P. \quad (8.28)$$

Formulation ADV3 is consistent with the strong conservation form of the primal discretisation it is derived from. It is directly compatible with high-Re wall boundary conditions. On the contrary, formulation ADV1 (8.21) is not necessarily consistent with the primal high-Re discretisation and requires an exceptional treatment for $[\underline{\nabla} U]_P$ in the first cell layer: A simplistic numerical evaluation according to Eqn. (7.8) usually overpredicts $[\underline{n} \cdot \underline{\nabla} U]_P$, and causes a strong source of adjoint momentum normal to the wall; pressure and velocity oscillations were observed in the adjoint solution. A work-around for ADV1

in combination with high-Re wall boundary conditions is to calculate $[\underline{n} \cdot \underline{\nabla} U]_P$ according to the logarithmic law of the wall instead:

$$[\underline{n} \cdot \underline{\nabla} U]_P = \frac{U^\tau}{\kappa y_P} \underline{t}. \quad (8.29)$$

ADV3 does not require such an exceptional treatment in combination with high-Re wall boundary conditions since the advective fluxes are directly determined from Eqn. (8.27), based on the primal convective face fluxes. Compared to ADV1/2, it alleviates the strength of the momentum source in the wall-normal direction.

Viscous Diffusion

The viscous diffusion operator in the momentum equations is self-adjoint. The corresponding adjoint face fluxes are

$$\hat{F}_i^d = - \int_{\Delta\Gamma} 2 \mu_{\text{eff}} \hat{S}_{ij} d\Gamma_j = - \int_{\Delta\Gamma} \mu_{\text{eff}} \left(\frac{\partial \hat{U}_i}{\partial x_j} + \frac{\partial \hat{U}_j}{\partial x_i} \right) d\Gamma_j. \quad (8.30)$$

The first term on the left-hand side resembles the diffusion term in the scalar transport equation. It has symmetric matrix coefficients and is treated implicitly according to Section 8.1. The second term is considered explicitly re-using the primal scheme, see Eqns. (7.32) and (7.33).

Pressure Term

According to Eqns. (7.26) and (8.18), a consistent approximation of the adjoint pressure term in the adjoint momentum equation can be derived from the variation of the discretised equation of continuity. In the primal pressure-correction scheme outlined in Section 7.3, the latter contributes the right-hand side terms to the pressure-Poisson equation. An example is given for a CV adjacent to a symmetry boundary patch as shown in Figure 8.2. The variation of the mass flux through the symmetry boundary patch is zero. The corresponding contributions to the variation of the Lagrangian are

$$\delta L \approx \dots - \hat{p}_P (\delta \dot{m}_{PN_1} + \delta \dot{m}_{PN_2}) - \hat{p}_{N_1} (\dots) - \hat{p}_{N_2} (\dots) + \dots \quad (8.31a)$$

$$\begin{aligned} \approx \dots - \hat{p}_P \left[\Delta\Gamma_i|_{PN_1} (\lambda_{PN_1} \delta U_i|_{N_1} + [1 - \lambda_{PN_1}] \delta U_i|_P) \right. \\ \left. + \Delta\Gamma_i|_{PN_2} (\lambda_{PN_2} \delta U_i|_{N_2} + [1 - \lambda_{PN_2}] \delta U_i|_P) \right] \\ - \hat{p}_{N_1} [\dots] - \hat{p}_{N_2} [\dots] + \dots \end{aligned} \quad (8.31b)$$

$$\begin{aligned} \approx \dots - \delta U_i|_P \left[\Delta\Gamma_i|_{PN_1} (1 - \lambda_{PN_1}) (\hat{p}_P - \hat{p}_{N_1}) \right. \\ \left. + \Delta\Gamma_i|_{PN_2} (1 - \lambda_{PN_2}) (\hat{p}_P - \hat{p}_{N_2}) \right] \\ - \delta U_i|_{N_1} [\dots] - \delta U_i|_{N_2} [\dots] + \dots \end{aligned} \quad (8.31c)$$

By virtue of

$$0 = \Delta\Gamma_i|_{PN_1} + \Delta\Gamma_i|_{PN_2} + \Delta\Gamma_i|_{PB}, \quad (8.32)$$

Eqn. (8.31c) can be rewritten as

$$\begin{aligned} \delta L \approx \dots + \delta U_i|_P & \left[\Delta\Gamma_i|_{PN_1} ([1 - \lambda_{PN_1}] \hat{p}_{N_1} + \lambda_{PN_1} \hat{p}_P) \right. \\ & + \Delta\Gamma_i|_{PN_2} ([1 - \lambda_{PN_2}] \hat{p}_{N_2} + \lambda_{PN_2} \hat{p}_P) \\ & \left. + \Delta\Gamma_i|_{PB} \hat{p}_P \right] + \delta U_i|_{N_1} [\dots] + \delta U_i|_{N_2} [\dots] + \dots \end{aligned} \quad (8.33)$$

Note that an adjoint-consistent first-order approximation, $\hat{p}_B = \hat{p}_P$, is obtained on the symmetry face. A conservative description of the adjoint pressure contribution to CV P is identified from Eqn. (8.33)

$$\begin{aligned} \Delta\Omega \frac{\partial \hat{p}}{\partial x_i} \Big|_P & \approx \left[\hat{\lambda}|_{PN_1} \hat{p}_{N_1} + (1 - \hat{\lambda}|_{PN_1}) \hat{p}_P \right] \Delta\Gamma_i|_{PN_1} \\ & + \left[\hat{\lambda}|_{PN_2} \hat{p}_{N_2} + (1 - \hat{\lambda}|_{PN_2}) \hat{p}_P \right] \Delta\Gamma_i|_{PN_2} + \hat{p}_P \Delta\Gamma_i|_{PB}. \end{aligned} \quad (8.34)$$

As observed for the adjoint CDS convection scheme, Eqn. (8.9), the interpolation weights $\hat{\lambda}_{PN} = 1 - \lambda_{PN}$ are inverse to the primal ones. Similarly, the primal, CDS-based pressure scheme leads to the adjoint pressure flux approximation

$$\hat{F}_i^p|_{PN} \approx \left[\hat{\lambda}_{PN} \hat{p}_N + (1 - \hat{\lambda}_{PN}) \hat{p}_P \right] \Delta\Gamma_i|_{PN}. \quad (8.35)$$

The adjoint pressure gradient is obtained from the discrete velocity divergence via summation by parts. Like integration by parts, summation by parts implies a sign change between Eqns. (8.31b) and (8.33).

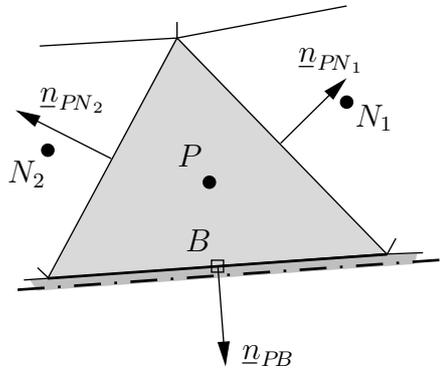


Figure 8.2.: CV next to a symmetry surface patch

8.3. Pressure-Correction Scheme

Continuity Equation

The adjoint continuity equation is derived from the primal pressure term in the discrete momentum equation, cf. Eqn. (8.18). With $\delta p_B \approx \delta p_P$, the contributions to the CV depicted in Figure 8.2 are

$$\delta L \approx \dots + \hat{U}_i|_P (\delta F_i^p|_{PN_1} + \delta F_i^p|_{PN_2} + \delta F_i^p|_{PB}) + \hat{U}_i|_{N_1} (\dots) + \hat{U}_i|_{N_2} (\dots) + \dots \quad (8.36a)$$

$$\begin{aligned} \approx \dots + \hat{U}_i|_P \left[\Delta\Gamma_i|_{PN_1} (\lambda_{PN_1} \delta p_{N_1} + [1 - \lambda_{PN_1}] \delta p_P) \right. \\ \left. + \Delta\Gamma_i|_{PN_2} (\lambda_{PN_2} \delta p_{N_2} + [1 - \lambda_{PN_2}] \delta p_P) \right. \\ \left. + \Delta\Gamma_i|_{PB} \delta p_P \right] + \hat{U}_i|_{N_1} [\dots] + \hat{U}_i|_{N_2} [\dots] + \dots \end{aligned} \quad (8.36b)$$

$$\begin{aligned} \approx \dots + \delta p_P \left[\Delta\Gamma_i|_{PN_1} (1 - \lambda_{PN_1}) (\hat{U}_i|_P - \hat{U}_i|_{N_1}) \right. \\ \left. + \Delta\Gamma_i|_{PN_2} (1 - \lambda_{PN_2}) (\hat{U}_i|_P - \hat{U}_i|_{N_2}) \right. \\ \left. + \Delta\Gamma_i|_{PB} \hat{U}_i|_P \right] + \delta p_{N_1} [\dots] + \delta p_{N_2} [\dots] + \dots \end{aligned} \quad (8.36c)$$

With Eqn. (8.32), Eqn. (8.36c) can be restated as

$$\begin{aligned} \delta L \approx \dots - \delta p_P \left[\Delta\Gamma_i|_{PN_1} \left([1 - \lambda_{PN_1}] \hat{U}_i|_{N_1} + \lambda_{PN_1} \hat{U}_i|_P \right) \right. \\ \left. + \Delta\Gamma_i|_{PN_2} \left([1 - \lambda_{PN_2}] \hat{U}_i|_{N_2} + \lambda_{PN_2} \hat{U}_i|_P \right) \right] \\ - \delta p_{N_1} [\dots] - \delta p_{N_2} [\dots] + \dots \end{aligned} \quad (8.37)$$

According to Eqn. (8.37), the flux through the symmetry patch is found to be zero. The general FV discretisation of the adjoint continuity equation for CV P reads

$$\begin{aligned} 0 = - \left[\hat{\lambda}|_{PN_1} \hat{U}_i|_{N_1} + (1 - \hat{\lambda}|_{PN_1}) \hat{U}_i|_P \right] \Delta\Gamma_i|_{PN_1} \\ - \left[\hat{\lambda}|_{PN_2} \hat{U}_i|_{N_2} + (1 - \hat{\lambda}|_{PN_2}) \hat{U}_i|_P \right] \Delta\Gamma_i|_{PN_2}. \end{aligned} \quad (8.38)$$

Note that the operator for the adjoint velocity divergence is obtained from the discrete pressure gradient via summation by parts. If carried out once, summation by parts leads to a change of sign as observed in the previous section.

Pressure-Correction Scheme

A pressure equation can be derived for the adjoint problem like for the primal. When the continuity operator is applied to the adjoint momentum balance (ADV1), the following

continuous expression is obtained

$$\frac{1}{\rho} \frac{\partial^2 \hat{p}}{\partial x_j^2} = \frac{\partial U_j}{\partial x_i} \frac{\partial \hat{U}_i}{\partial x_j} + \frac{\partial}{\partial x_i} \left(U_j \frac{\partial \hat{U}_j}{\partial x_i} \right) \quad (8.39a)$$

$$= 2 S_{ij} \hat{S}_{ij} + U_j \frac{\partial^2 \hat{U}_j}{\partial x_i^2} \quad \text{in } \Omega. \quad (8.39b)$$

An alternative formulation (ADV2) is obtained via integration by parts of the advection term according to Eqn. (A.3)

$$\frac{1}{\rho} \frac{\partial^2 \hat{p}}{\partial x_j^2} = \frac{\partial U_j}{\partial x_i} \frac{\partial \hat{U}_i}{\partial x_j} - \frac{\partial}{\partial x_i} \left(\hat{U}_j \frac{\partial U_j}{\partial x_i} \right) \quad (8.40a)$$

$$= 2 W_{ij} \hat{W}_{ij} - \hat{U}_j \frac{\partial^2 U_j}{\partial x_i^2} \quad \text{in } \Omega. \quad (8.40b)$$

The adjoint RANS equations are solved by means of a SIMPLE-based algorithm, which is presented in a semi-discrete formulation here. A consistent pressure-correction equation for the adjoint system is obtained from the *discrete* representation of the adjoint momentum-continuity problem. The source term \hat{S}_i of the adjoint momentum equations contains contributions from the advection term ADV1/2/3 treated explicitly. Additional contributions to the adjoint source term may result from the objective functional. The semi-discrete momentum equations are formulated for both the predictor stage, carrying the asterisk, and the corrected stage at the end of the adjoint pressure-correction cycle m . Explicit treatment of variables is referred to by superscript $(m-1)$. The velocity prediction, $\hat{U}_i^*|_P^m$, is obtained from the adjoint momentum equations

$$\hat{A}_P \hat{U}_i^*|_P^m + \sum_{N(P)} \hat{A}_N \hat{U}_i^*|_N^m = -\Delta\Omega \left[\frac{\partial \hat{p}}{\partial x_i} \right]_P^{m-1} - \rho \Delta\Omega \frac{\partial U_j}{\partial x_i} \hat{U}_j|_P^{m-1}. \quad (8.41)$$

The adjoint momentum balance can be expressed in terms of the corrected adjoint pressure and velocity components

$$\hat{U}_i|_P^m = \hat{U}_i^*|_P^m + \hat{U}_i'|_P^m \quad \text{and} \quad \hat{p}|_P^m = \hat{p}|_P^{m-1} + \hat{p}'|_P^m, \quad (8.42)$$

which leads to

$$\begin{aligned} \hat{A}_P \left(\hat{U}_i^* + \hat{U}_i' \right) |_P^m + \sum_{N(P)} \hat{A}_N \left(\hat{U}_i^* + \hat{U}_i' \right) |_N^m = \\ - \Delta\Omega \left[\frac{\partial}{\partial x_i} \left(\hat{p}^{m-1} + \hat{p}' \right) \right]_P - \rho \Delta\Omega \frac{\partial U_j}{\partial x_i} \hat{U}_j|_P^m. \end{aligned} \quad (8.43)$$

Note that the iteration counter for the advection term has increased from $(m-1)$ to m . The difference between Eqn. (8.41) and (8.43) is

$$\begin{aligned} \hat{A}_P \hat{U}_i'|_P^m + \sum_{N(P)} \overline{\hat{A}_N \hat{U}_i'|_N^m} = \\ - \Delta\Omega \left[\frac{\partial \hat{p}'}{\partial x_i} \right]_P - \rho \Delta\Omega \frac{\partial U_j}{\partial x_i} \left(\overline{\hat{U}_j^*|_P^m} + \hat{U}_j'|_P^m - \overline{\hat{U}_j|_P^{m-1}} \right). \end{aligned} \quad (8.44)$$

The terms overlined above are neglected in the following. With regard to the neighbour contributions this treatment is in accordance with the original SIMPLE-algorithm. Both the iteration counter m and the subscripts P and N are left out for the sake of brevity, viz.

$$\hat{U}'_i = -\frac{\Delta\Omega}{\hat{A}_P} \left[\frac{\partial \hat{p}'}{\partial x_i} + \rho \frac{\partial U_j}{\partial x_i} \hat{U}'_j \right]. \quad (8.45)$$

The underlined term in Eqn. (8.45) depending on \hat{U}'_j is neglected in the standard implementation. For $\partial j_\Omega / \partial p = 0$, the corrected adjoint velocity field is forced to be solenoidal at the end of the pressure-correction cycle, viz.

$$0 = \frac{\partial}{\partial x_i} \left[\hat{U}_i^* + \hat{U}'_i \right]. \quad (8.46)$$

The reduced expression for Eqn. (8.45),

$$\hat{U}'_i = -\frac{\Delta\Omega}{\hat{A}_P} \left[\frac{\partial \hat{p}'}{\partial x_i} \right], \quad (8.47)$$

is substituted into Eqn. (8.46) to arrive at the adjoint pressure-correction equation

$$\frac{\partial}{\partial x_i} \left[\frac{\Delta\Omega}{\hat{A}_P} \left(\frac{\partial \hat{p}'}{\partial x_i} \right) \right] = \frac{\partial \hat{U}_i^*}{\partial x_i}. \quad (8.48)$$

The semi-discrete expression (8.48) is discretised by the FV method like its primal counterpart, cf. (7.45). The adjoint pressure-correction obtained from Eqn. (8.48) is used to correct the adjoint pressure field according to (8.42). The corresponding velocity correction obtained from (8.47) is also used to update the ‘‘adjoint mass fluxes’’. An extended, two-fold pressure-correction approach is outlined in Appendix D. It iteratively accounts for the advection contribution to the momentum equations.

The self-adjoint ‘‘pressure diffusion’’ on the left-hand side of the adjoint pressure-correction equation (8.48) is approximated according to the diffusion treatment outlined in Section 7.1. The adjoint pressure gradient inside the divergence operator originates from the adjoint momentum balance.

Matrix-Vector Interpretation

The adjoint SIMPLE algorithm outlined above may also be viewed from the matrix-vector point of view (8.18). Since a complete linearisation underlies the adjoint system, \hat{S} also hosts the additional advection terms ADV1/2/3. The adjoint matrix-vector problem reveals that the discrete-adjoint pressure-gradient operator of the adjoint momentum equation, $\underline{\underline{C}}^T$, is found by transposition of the corresponding primal divergence operator of the primal continuity equation. In the same way, the adjoint continuity operator $\underline{\underline{B}}^T$ is found by transposition of the primal pressure-gradient operator. Moreover, the coupled view is very instructive for the derivation of the discrete-adjoint boundary conditions.

The problem is solved iteratively as its primal counterpart:

$$\underline{\underline{A}}^T \cdot \hat{\underline{U}}^* = -\underline{\underline{C}}^T \cdot \hat{\underline{p}}^{m-1} + \hat{\underline{S}}. \quad (8.49)$$

Then, the adjoint pressure-correction $\hat{\underline{p}}'$ is determined from

$$\left(\underline{\underline{B}}^T \cdot \underline{\underline{A}}^{-1} \cdot \underline{\underline{C}}^T \right) \cdot \hat{\underline{p}}' = \underline{\underline{B}}^T \cdot \hat{\underline{U}}^* - \hat{\underline{Q}}. \quad (8.50)$$

Mind that, in the adjoint system, the underlying assumption $\hat{\underline{U}}' = -(\underline{\underline{A}}^{-1} \cdot \underline{\underline{C}}^T) \cdot \hat{\underline{p}}'$ is more restrictive than in the primal problem since more terms are neglected. Finally, the adjoint velocity and pressure fields are corrected via

$$\hat{\underline{p}}^m = \hat{\underline{p}}^{m-1} + \alpha_{\hat{\underline{p}}} \hat{\underline{p}}' \quad \text{and} \quad \hat{\underline{U}}^m = \hat{\underline{U}}^* - (\underline{\underline{A}}^{-1} \cdot \underline{\underline{C}}^T) \cdot \hat{\underline{p}}'. \quad (8.51)$$

The adjoint Schur-complement is the transposed of the primal due to

$$\underline{\underline{C}} \cdot \underline{\underline{A}}^{-1} \cdot \underline{\underline{B}} = (\underline{\underline{B}}^T \cdot \underline{\underline{A}}^{-1} \cdot \underline{\underline{C}}^T)^T. \quad (8.52)$$

The Rhie and Chow correction (7.47) of the primal system is reapplied to obtain smooth adjoint solutions.

8.4. Boundary Conditions

Since the approximation order of the boundary fluxes is often lower than of the interior fluxes—typically first-order gradient approximations are used on the boundaries—duality violations on the discrete level can lead to major inconsistencies in objective functional values and derivatives. Boundary conditions to the Navier–Stokes equations and the objective functionals declared on Γ_O should be discretised in a consistent, strictly conservative manner. It allows to devise conform, adjoint-consistent discretisations for both the adjoint boundary conditions and the adjoint gradient expression.

In the adjoint calculus, a conform boundary treatment involves

- (a) boundary terms stemming from the definition of the objective functional defined on Γ_O ;
- (b) primal boundary conditions;
- (c) adjoint boundary conditions;
- (d) the adjoint gradient equation on Γ_D .

The boundary stress acting on solid and fluid, respectively, is considered according to the method of sections. In a momentum-conservative FV discretisation, the net sum of the fluid force on the boundary and the boundary force on the fluid should be zero. In the discrete Lagrangian, the boundary stress acting on the fluid (b) is weighted by the adjoint multipliers $\hat{\underline{U}}_P$ in the balance for the wall-adjacent CV P . If the contribution (a)

from the force objective acting on Γ_O into the direction \underline{d}^* is discretised consistently, the contribution to the discrete Lagrangian can be written as

$$L \approx \dots + \left(d_i^* + \hat{U}_i|_P \right) F_i|_{\text{wall}} + \dots$$

Since the analysis is confined to the contribution of the boundary face PB to the Lagrangian, the boundary-face index PB is omitted here. The presentation also holds if no force objective functional is declared on the considered boundary wall face; in that case the force projection vector is the null vector, $\underline{d}^* = \underline{0}$.

A deformation of the design surface Γ_D implies a perturbation of the position vectors of the boundary nodes, \underline{x}_B , as described in Section 7.7. Based on the full variation of the Lagrangian, $\delta L = (\delta \underline{u} + \delta_p + \delta_y)L$, a unified formulation covering low- and high-Re descriptions is derived to obtain the discretisation schemes for the adjoint wall boundary condition and the sensitivity equation.

A **low-Re wall boundary formulation** is considered first. The description also covers no-slip wall faces that are part of the objective surface Γ_O and may be part of the design surface Γ_D as well. The eddy viscosity μ_T is zero on the wall boundary and negligible in the viscous sublayer, i.e. the effective viscosity μ_{eff} equals the molecular viscosity μ . In fully laminar flow, μ_{eff} is to be replaced by μ . With $\delta F_i|_{\text{wall}}$ substituted according to Eqn. (7.91), the following expression is obtained

$$\begin{aligned} \delta L &\approx \dots + \left(d_i^* + \hat{U}_i|_P \right) \delta F_i|_{\text{wall}} \\ &\approx \dots + \left(d_i^* + \hat{U}_i|_P \right) \left(n_i \delta p_P \right. \\ &\quad \left. + \mu_{\text{eff}} T_{ij} y_P^{-1} \left[\delta y_P y_P^{-1} (U_j|_B - U_j|_P) + \delta U_j|_P \right] \right) \Delta \Gamma + \dots \end{aligned} \quad (8.53)$$

Expression (8.53) is reordered via summation by parts in order to obtain a discretisation for both the adjoint boundary condition (c) and the adjoint gradient equation (d):

$$\delta L \approx \dots + \delta p_P \left(d_i^* + \hat{U}_i|_P \right) n_i \Delta \Gamma \quad (8.54a)$$

$$+ \delta U_i|_P \left(\frac{\mu_{\text{eff}} T_{ij}}{y_P} \left[d_j^* + \hat{U}_j|_P \right] \right) \Delta \Gamma \quad (8.54b)$$

$$+ \delta y_P \left(\frac{\mu_{\text{eff}} T_{ij}}{y_P^2} \left[d_j^* + \hat{U}_j|_P \right] [U_i|_B - U_i|_P] \right) \Delta \Gamma + \dots \quad (8.54c)$$

With $\hat{U}_j|_B = -d_j^*$, the adjoint boundary condition (6.11) of the continuous formulation is recovered (c). It reappears as a discrete source term in the CV P next to the wall. The dual discretisation for the viscous boundary flux in the adjoint system is calculated through

$$\hat{F}_i^d|_{\text{wall}} = -\frac{\mu_{\text{eff}} T_{ij}}{y_P} \left(\hat{U}_j|_B - \hat{U}_j|_P \right) \Delta \Gamma. \quad (8.55)$$

It is interesting to note that, like in the flow problem, the adjoint strain vector is projected onto the boundary via scalar multiplication by \underline{T} .

Alternatively, the boundary-layer flow can be modelled with **high-Re wall boundary conditions** based on the logarithmic law of the wall (7.64). The variation of the Lagrangian, including contributions from a force objective (a) according to Section 7.6, is

$$\begin{aligned}\delta L &\approx \dots + \left(d_i^* + \hat{U}_i|_P\right) \delta F_i|_{\text{wall}} + \dots \\ &\approx \dots + \left(d_i^* + \hat{U}_i|_P\right) (n_i \delta p_P - \delta \tau_i^w) \Delta \Gamma + \dots\end{aligned}\quad (8.56)$$

In the high-Re formulation, $\delta \tau_i^w$ is substituted according to Eqn. (7.102). With the auxiliary definition of the viscosity in terms of the logarithmic law of the wall

$$\mu_{\log} = \frac{\rho U^\tau \kappa}{\ln(E y_P^+)} y_P = \frac{\rho \kappa C_\mu^{1/4} k^{1/2}}{\ln(E y_P^+)} y_P, \quad (8.57)$$

Eqn. (8.56) can be rearranged via summation by parts:

$$\delta L \approx \dots + \delta p_P \quad n_i \left(d_i^* + \hat{U}_i|_P\right) \Delta \Gamma \quad (8.58a)$$

$$+ \delta U_i|_P \quad \frac{\mu_{\log}}{y_P} (T_{ij} + t_i t_j) \left(d_j^* + \hat{U}_j|_P\right) \Delta \Gamma \quad (8.58b)$$

$$+ \delta y_P \quad 2 \frac{\mu_{\log}}{y_P} \frac{U^\tau}{\kappa y_P} t_j \left(d_j^* + \hat{U}_j|_P\right) \Delta \Gamma + \dots \quad (8.58c)$$

Once again, the adjoint boundary condition $\hat{U}_i|_B = -d_i^*$ is recovered. With the auxiliary definition of the viscosity (8.57), a FV implementation analogue to the low-Re description (8.55) is found:

$$\hat{F}_i^d|_{\text{wall}} \approx -\frac{\mu_{\log}}{y_P} (T_{ij} + t_i t_j) \left(-d_j^* - \hat{U}_j|_P\right) \Delta \Gamma. \quad (8.59)$$

Adjoint pressure and volume fluxes through no-slip wall faces are discretised as shown for symmetry boundary faces. The adjoint pressure flux through a wall face is identified from Eqn. (8.33), which yields a first-order approximation $\hat{p}_B = \hat{p}_P$, which is a valid, adjoint-consistent approximation to the boundary condition of the continuous adjoint problem. The adjoint volume flux, $-d^* \cdot \Delta \Gamma$, through a wall boundary face that contributes to the objective functional is found from (8.54a) or (8.58a).

The *continuous* form of the **adjoint pressure boundary condition** (6.23) can be considered as a “do-nothing” or no-flux condition

$$n_j \left[\rho U_j \hat{U}_i + \mu_{\text{eff}} \frac{\partial \hat{U}_i}{\partial x_j} - \hat{p} \delta_{ij} \right] = 0 \quad \text{on} \quad \Gamma_{\text{out}} \quad (8.60)$$

or

$$\hat{F}|_{\text{out}} = \hat{F}^c + \hat{F}^d + \hat{F}^p = \mathbf{0}. \quad (8.61)$$

Accordingly, the adjoint pressure takes the value

$$\hat{p} = \rho U^n \hat{U}^n + \mu_{\text{eff}} \frac{\partial \hat{U}^n}{\partial n} \quad \text{on} \quad \Gamma_{\text{out}}. \quad (8.62)$$

The viscous normal stress is neglected in the vicinity of the outflow, which corresponds to the velocity boundary condition, $\underline{n} \cdot \underline{\nabla} U = \underline{0}$, used in the primal discretisation. It leads to

$$\hat{p}^m = \rho U^n \hat{U}^n \quad \text{on } \Gamma_{\text{out}}. \quad (8.63)$$

From the pressure-correction point of view, Eqn. (8.63) determines a pseudo-Dirichlet boundary value for the adjoint pressure. Thus, \hat{p}' at the boundary is

$$\hat{p}' = \hat{p}^m - \hat{p}^{m-1} \quad \text{on } \Gamma_{\text{out}}, \quad (8.64)$$

so that the corrected pressure at the end of the SIMPLE cycle assumes the value demanded by (8.63). Eqn. (8.64) is a pseudo-Dirichlet condition which evolves during the SIMPLE-iteration. Since the value of \hat{p}' is prescribed at the boundary, the adjoint velocity has to be corrected via:

$$(\hat{U}^n)' = \frac{\Delta \Omega}{\hat{A}_P} \frac{\partial \hat{p}'}{\partial n} \quad \text{on } \Gamma_{\text{out}}. \quad (8.65)$$

A one-sided approximation is used for the adjoint pressure gradient, which is of first-order accuracy. The boundary value for \hat{p}' required in (8.65) is prescribed by Eqn. (8.64). The boundary value for \hat{A}_P is extrapolated from the adjacent CV.

The numerical treatment of the outlet face proposed above is confirmed by a *discrete* adjoint analysis for CV P next to the boundary sketched in Figure 8.3. A first-order description of the convective contribution of CV P to the variation of the Lagrangian is

$$\begin{aligned} \delta L \approx & \dots + (\max[\dot{m}_{PN_1}, 0] \delta U_i|_P + \min[\dot{m}_{PN_1}, 0] \delta U_i|_{N_1}) (\hat{U}_i|_P - \hat{U}_i|_{N_1}) \\ & + (\max[\dot{m}_{PN_2}, 0] \delta U_i|_P + \min[\dot{m}_{PN_2}, 0] \delta U_i|_{N_2}) (\hat{U}_i|_P - \hat{U}_i|_{N_2}) \\ & + (\max[\dot{m}_{PB}, 0] + \min[\dot{m}_{PB}, 0]) \delta U_i|_P \hat{U}_i|_P + \dots \end{aligned} \quad (8.66)$$

By virtue of the CV continuity balance, $\sum \min(\dot{m}, 0) = -\sum \max(\dot{m}, 0)$, Eqn. (8.66) can be restated as

$$\begin{aligned} \delta L \approx & \dots - \delta U_i|_P \left[\min(\dot{m}_{PN_1}, 0) \hat{U}_i|_P + \max(\dot{m}_{PN_1}, 0) \hat{U}_i|_{N_1} \right. \\ & \left. + \min(\dot{m}_{PN_2}, 0) \hat{U}_i|_P + \max(\dot{m}_{PN_2}, 0) \hat{U}_i|_{N_2} \right] \\ & - \delta U_i|_{N_1} [\dots] - \delta U_i|_{N_2} [\dots] + \dots \end{aligned} \quad (8.67)$$

The negative adjoint downstream scheme is recovered for the internal faces PN_1 and PN_2 . Obviously, the resulting adjoint convective flux through the outlet patch, $\hat{F}^c|_{\text{out}}$, is zero. The adjoint diffusion, pressure and volume fluxes are identified from Eqn. (8.18). Assuming $\delta U_i|_B \approx \delta U_i|_P$ leads to $\underline{0} = \hat{F}^p|_{\text{out}} = \hat{F}^d|_{\text{out}}$. The findings of the discrete adjoint analysis are consistent with the weak boundary condition (8.61) of the continuous adjoint problem.

The numerical treatment of **inlet**, **wall** and **symmetry patches** on $\Gamma \setminus \Gamma_O$ has been derived in the same way and is equivalent to the primal implementation described in Section 7.4.

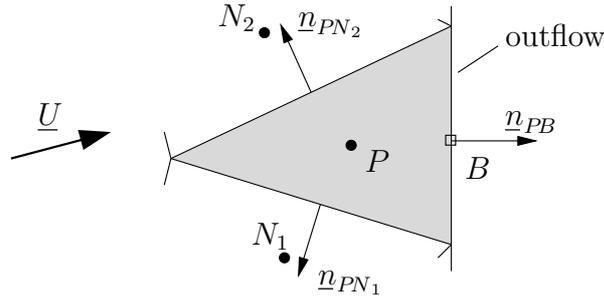


Figure 8.3.: CV next to a pressure outflow boundary patch

8.5. Gradient Equation

After solving the discretised adjoint equations for the considered objective functional, its sensitivity derivative is evaluated in the *adjoint post-processing*. According to Section 6.3, the local portion of the gradient distribution, G_L , is evaluated over Γ_D . The formulation is specific to the type of control and independent of the considered objective functional.

When boundary layers are resolved numerically in a **low-Re** strategy, a consistent discretisation scheme for the gradient equation (d) directly follows from (8.54c). It is the partial derivative of the discrete Lagrangian with respect to the shape control δy divided by the patch area $\Delta\Gamma$, viz.

$$G_L \approx -\frac{\mu_{\text{eff}}}{y_P^2} (U_i|_B - U_i|_P) T_{ij} \left(-d_j^* - \hat{U}_j|_P \right) \quad \text{on } \Gamma_D. \quad (8.68)$$

Note that the description is an adjoint-consistent approximation to the continuous adjoint formulation (6.39).

When a **high-Re** wall boundary condition is employed, the discrete gradient equation is identified from (8.58c)

$$G_L \approx -2 \frac{\mu_{\text{log}}}{y_P} \frac{U^\tau}{\kappa y_P} t_j \left(-d_j^* - \hat{U}_j|_P \right) \quad \text{on } \Gamma_D, \quad (8.69)$$

with the auxiliary viscosity μ_{log} declared according to Eqn. (8.57). Eqn. (8.69) is an approximation to the continuous adjoint formulation (6.39) expressed in terms of the logarithmic law of the wall. The velocity gradient based on the logarithmic law of the wall is found again in Eqn. (8.69):

$$\left. \frac{\partial U^t}{\partial n} \right|_P = \frac{U^\tau}{\kappa y_P}.$$

In either case, the one-sided first-order approximation used for the boundary-normal velocity derivative is recovered in the gradient equation. The formulations are analogue to the approximations of boundary-normal derivatives applied in the primal discretisation—force objective (a) and boundary condition (b)—which are of first-order accuracy. Mind that the influence of the boundary curvature is neglected here. This simplification is common practice in industrial CFD codes and is consistent with the primal discretisation.

8.6. Solution Algorithm

The process chains for both the direct-differentiation method and the adjoint method are illustrated in Figure 8.4. In the direct-differentiation method the linearised Navier–Stokes problem is solved in forward direction, whereas the adjoint code calculates the sensitivity derivative in reverse mode. This is reflected in the top-down (linearised) and the bottom-up (adjoint) strategy, respectively. The adjoint module is integrated into the main code. It is possible to

- run the adjoint directly after the primal solver; or
- execute the adjoint alone restoring the converged primal solution from a restart-file.

According to Eqns. (6.5) and (6.6), the adjoint problem is driven by the adjoint boundary conditions on Γ_O and the volume source terms in Ω_O , both of which are specific to the objective functional. These are defined in the so-called *adjoint pre-processing*, which is executed prior to solving the adjoint PDE, cf. Figure 8.4. This step corresponds to the evaluation of the objective functional variation in the post-processing of the linearised RANS problem, which is carried out in the same stage. The adjoint code must have the flexibility to be tailored to different types of user-defined objective functionals. The following strategy is pursued in the adjoint module:

- A collection of standard objectives, such as fluid forces or energy dissipation, are implemented within the main program.
- Further, case-specific definitions can be provided through a user-coding interface. User-routines are compiled individually and linked to the main program. This strategy permits to keep the program clearly arranged and, at the same time, ensures a high level of versatility.

The adjoint PDE are iteratively solved by the adapted SIMPLE algorithm sketched in Section 8.3. The adjoint momentum equations (8.41) are solved for the adjoint velocity components followed by the adjoint pressure equation (8.48). In the subsequent correction step, the adjoint pressure and velocity components are updated according to (8.42) and (8.47). The adjoint routines are similar to the primal ones. The same PETSc-solvers and pre-conditioners that are used to solve the primal linear equation systems (Section 7.5) are reused for the corresponding adjoint systems. The adjoint pressure-correction system is symmetric as its primal counterpart, so that the same equation solvers could be used. As the adjoint system is linear, the individual adjoint system matrices can be stored and reused during the iteration. In spite of the linearity of the adjoint problem, the right-hand side contributions—stemming from the adjoint advection terms ADV1–3, explicit deferred corrections for second-order accurate reconstructions, the definition of the objective functional or further adjoint transport equations—need to be updated continuously. The dense coupling of the adjoint equations increases the number of right-hand side terms in the segregated approach. The set of adjoint equations for momentum and pressure-projection is iterated using the adjoint SIMPLE scheme. The adjoint system

is under-relaxed as the primal Navier–Stokes system (7.5). Like in the primal solver (7.83), the adjoint residual for $\hat{\phi}$ in iteration k is calculated according to the L_1 -norm

$$\overline{\hat{R}}^k = \frac{1}{\overline{\hat{R}}^1} \sum_{i=1}^n \hat{R}_i^k \quad \text{with} \quad \hat{R}_i^k = \hat{A}_{ii}^k \hat{\phi}_i^{k-1} + \sum_{j \in N(i)} \hat{A}_{ij}^k \hat{\phi}_j^{k-1} - \hat{S}_i^k. \quad (8.70)$$

When the adjoint residual value has fallen below a given threshold, or a prescribed number of adjoint iterations is exceeded, the gradient is evaluated in the so-called *adjoint post-processing*. According to Figure 8.4, it is executed on the same level like the boundary perturbation step of the forward linearisation. The gradient evaluation does not have to be modified or adapted by the user unless the type of control changes. Therefore, the boundary-based gradient Eqns. (8.68) and (8.69) are included within the main program. When a different sort of control is considered, e.g.

- a volume-based evaluation of the sensitivity derivative either with respect to the shape [e.g. 61, 126, 99, 108] or with respect to a porosity defined inside the fluid domain in the context of topology optimisation [6, 35, 111, 110],
- active flow control via blowing or suction [10, 8, 9], or
- grid parameters or points in conjunction with error analysis [169, 51, 52],

the adjoint post-processing can be adapted through user-coding interfaces as implemented for the adjoint pre-processing. It is interesting to note that the adjoint equations are the same for all these problems unless extra control terms do explicitly depend on the state or flow variables.

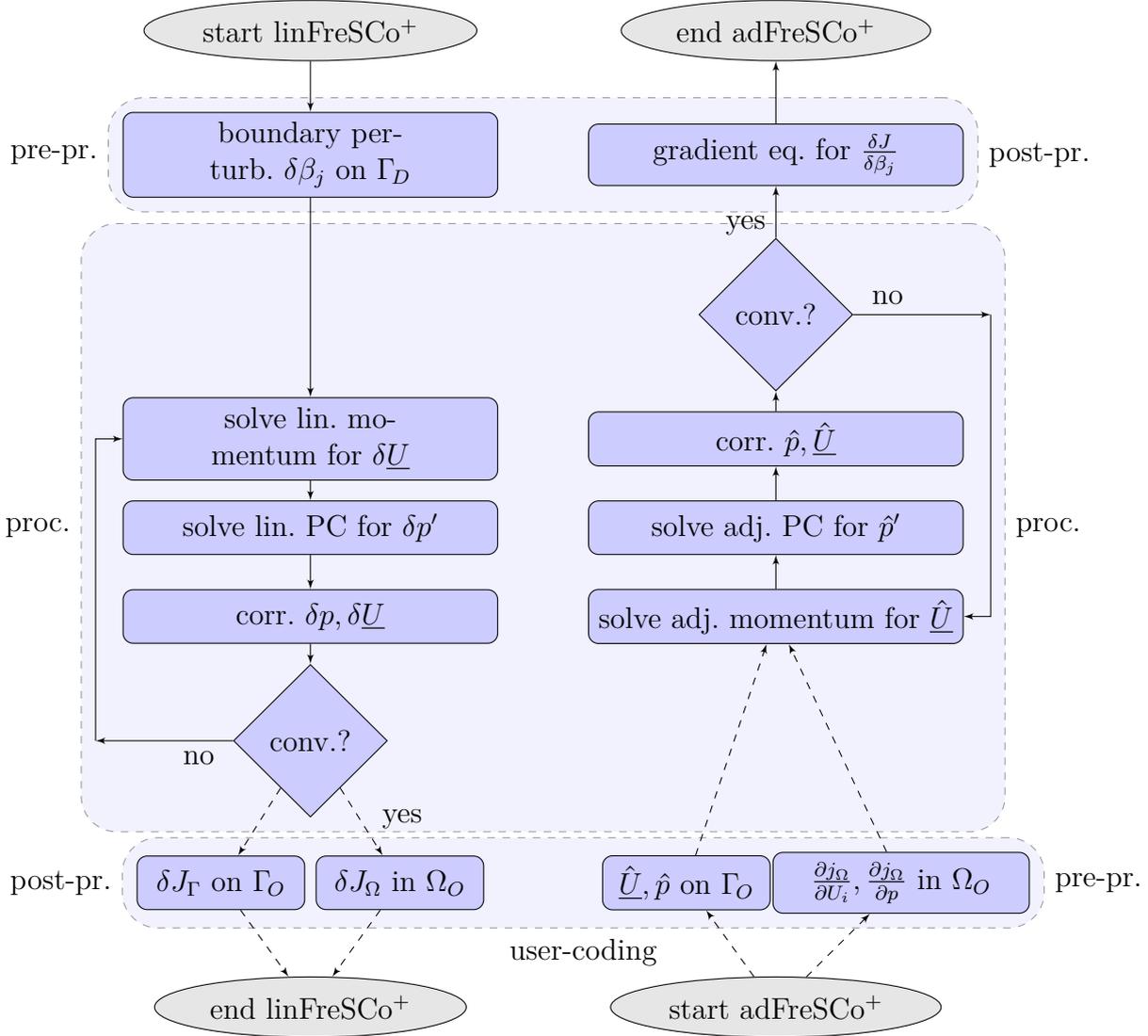


Figure 8.4.: Solution schemes for the linearised and the adjoint FreSCo⁺ solvers

9. Verification and Validation Studies

The origin of the adjoint Navier–Stokes equations is mathematical rather than physical. Since a direct comparison against experimental data is hardly possible, there is a need for alternative validation and verification strategies for the adjoint-based results. In a numerical experiment, the adjoint solution can be verified against sensitivity derivatives with respect to control parameters of *local* or *integral* influence. If, for example, a finite differentiation is carried out with respect to a parameter of global influence, the corresponding adjoint-based sensitivity derivative is obtained via an integral expression over the area of influence, Γ_D or Ω_C . Potential local deviations resulting from insufficiencies in the adjoint numerics may accumulate or cancel each other out in the sensitivity integral. When control parameters of local influence are used, many calculations have to be carried out to verify the sensitivity derivative over a large design surface Γ_D or control volume Ω_C .

Two concepts are pursued in this chapter which allow to locally assess the sensitivity derivative; these are based on

- adjoint field solutions derived analytically, or
- gradient distributions reconstructed via the direct-differentiation method.

9.1. Analytic Solution for Axis-Symmetric Couette Flow

An analytic adjoint solution was tailored in order to validate the adjoint solver. The axis-symmetric Couette problem sketched in Figure 9.1 was considered in combination with incompressible Navier–Stokes flow of homogeneous viscosity. The absence of in- and outflow boundaries simplifies the validation of the numerics. The axis-symmetric Couette problem represents an engineering shear flow which, when discretised in Cartesian coordinates, also allows to assess the adjoint discretisation schemes for convection and advection.

Analytic Solution

A description in physical cylinder coordinates (r, φ, z) enables an analytic solution. The velocity can be expressed for an arbitrary point:

$$\underline{U} = U^r \underline{\tilde{e}}^r + U^\varphi \underline{\tilde{e}}^\varphi + U^z \underline{\tilde{e}}^z. \quad (9.1)$$

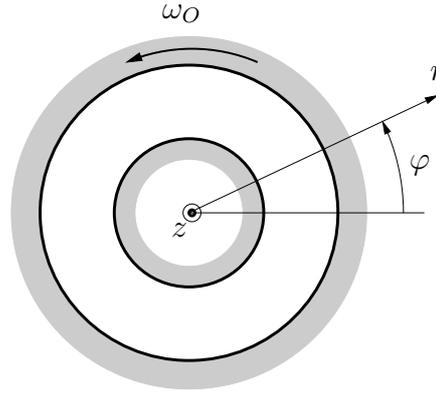


Figure 9.1.: Sketch of axis-symmetric Couette flow in physical cylinder coordinates (r, φ, z) ; outer radius r_O , inner radius r_I , angular velocity of outer ring ω_O

\tilde{e}^r , \tilde{e}^φ and \tilde{e}^z represent the unit vectors of the r , φ and z components, respectively. The corresponding representation of the gradient operator is

$$\nabla(\cdot) = \frac{\partial(\cdot)}{\partial r} \tilde{e}^r + \frac{1}{r} \frac{\partial(\cdot)}{\partial \varphi} \tilde{e}^\varphi + \frac{\partial(\cdot)}{\partial z} \tilde{e}^z. \quad (9.2)$$

The Navier–Stokes equations transformed into physical cylinder coordinates read:

Continuity equation:

$$\frac{\partial U^r}{\partial r} + \frac{U^r}{r} + \frac{1}{r} \frac{\partial U^\varphi}{\partial \varphi} + \frac{\partial U^z}{\partial z} = 0 \quad (9.3)$$

Radial momentum equation (\tilde{e}^r direction):

$$\rho \left(U^r \frac{\partial U^r}{\partial r} + \frac{U^\varphi}{r} \frac{\partial U^r}{\partial \varphi} + U^z \frac{\partial U^r}{\partial z} - \frac{(U^\varphi)^2}{r} \right) = -\frac{\partial p}{\partial r} + \mu \left(\frac{\partial^2 U^r}{\partial r^2} + \frac{1}{r} \frac{\partial U^r}{\partial r} - \frac{U^r}{r^2} + \frac{1}{r^2} \frac{\partial^2 U^r}{\partial \varphi^2} + \frac{\partial^2 U^r}{\partial z^2} - \frac{2}{r^2} \frac{\partial U^\varphi}{\partial \varphi} \right) \quad (9.4)$$

Circumferential momentum equation (\tilde{e}^φ direction):

$$\rho \left(U^r \frac{\partial U^\varphi}{\partial r} + \frac{U^\varphi}{r} \frac{\partial U^\varphi}{\partial \varphi} + U^z \frac{\partial U^\varphi}{\partial z} + \frac{U^r U^\varphi}{r} \right) = -\frac{\partial p}{\partial \varphi} + \mu \left(\frac{\partial^2 U^\varphi}{\partial r^2} + \frac{1}{r} \frac{\partial U^\varphi}{\partial r} - \frac{U^\varphi}{r^2} + \frac{1}{r^2} \frac{\partial^2 U^\varphi}{\partial \varphi^2} + \frac{\partial^2 U^\varphi}{\partial z^2} + \frac{2}{r^2} \frac{\partial U^r}{\partial \varphi} \right) \quad (9.5)$$

Axial momentum equation (\tilde{e}^z direction):

$$\rho \left(U^r \frac{\partial U^z}{\partial r} + \frac{U^\varphi}{r} \frac{\partial U^z}{\partial \varphi} + U^z \frac{\partial U^z}{\partial z} \right) = -\frac{\partial p}{\partial z} + \mu \left(\frac{\partial^2 U^z}{\partial r^2} + \frac{1}{r} \frac{\partial U^z}{\partial r} + \frac{1}{r^2} \frac{\partial^2 U^z}{\partial \varphi^2} + \frac{\partial^2 U^z}{\partial z^2} \right) \quad (9.6)$$

For symmetry reasons, $\frac{\partial}{\partial \varphi}(\cdot) = \frac{\partial}{\partial z}(\cdot) = 0$, the set of equations above reduces to

$$\frac{\partial p}{\partial r} = \rho \frac{(U^\varphi)^2}{r}, \quad (9.7)$$

$$0 = r \frac{\partial^2 U^\varphi}{\partial r^2} + \frac{\partial U^\varphi}{\partial r} - \frac{U^\varphi}{r}. \quad (9.8)$$

A no-slip wall boundary condition is imposed on both inner and outer cylinder walls. The outer wall, radius r_O , rotates with the angular velocity ω_O , whereas the inner wall, radius r_I , is fixed. This axis-symmetric Couette problem has the following analytical solution

$$U^r(r) = 0, \quad U^\varphi(r) = \alpha \left(r - \frac{r_I^2}{r} \right), \quad (9.9)$$

$$p(r) = p(r_I) + \rho \alpha^2 \left[\frac{r^2}{2} + 2r_I^2 \ln \left(\frac{r_I}{r} \right) - \frac{r_I^4}{2r^2} \right], \quad (9.10)$$

with

$$\alpha = \frac{\omega_O}{1 - (r_I/r_O)^2}. \quad (9.11)$$

The objective functional for the manufactured adjoint problem is the torque acting on the inner cylinder wall, $r = r_I$. With the position vector \underline{x}_0 denoting a point on the cylinder axis, the objective functional reads

$$J_\Gamma = \tilde{\underline{e}}^z \cdot \int_{\Gamma_O} (\underline{x} - \underline{x}_0) \times (\underline{\pi} \cdot d\underline{\Gamma}), \quad (9.12)$$

which can be rewritten as

$$J_\Gamma = \int_{\Gamma_O} j_\Gamma d\Gamma \quad \text{with} \quad j_\Gamma = -r_I \mu \frac{\partial U^\varphi}{\partial r}. \quad (9.13)$$

Accordingly, the adjoint Navier–Stokes equations can be stated in physical cylinder coordinates:

Adjoint continuity equation:

$$\frac{\partial \hat{U}^r}{\partial r} + \frac{\hat{U}^r}{r} + \frac{1}{r} \frac{\partial \hat{U}^\varphi}{\partial \varphi} + \frac{\partial \hat{U}^z}{\partial z} = 0 \quad (9.14)$$

Radial adjoint momentum equation ($\tilde{\underline{e}}^r$ direction):

$$\begin{aligned} & -\rho \left(U^r \frac{\partial \hat{U}^r}{\partial r} + \frac{U^\varphi}{r} \frac{\partial \hat{U}^r}{\partial \varphi} + U^z \frac{\partial \hat{U}^r}{\partial z} - \frac{U^\varphi \hat{U}^\varphi}{r} \right) \\ & + \rho \left(\hat{U}^r \frac{\partial U^r}{\partial r} + \hat{U}^\varphi \frac{\partial U^\varphi}{\partial r} + \hat{U}^z \frac{\partial U^z}{\partial r} \right) = \\ & -\frac{\partial \hat{p}}{\partial r} + \mu \left(\frac{\partial^2 \hat{U}^r}{\partial r^2} + \frac{1}{r} \frac{\partial \hat{U}^r}{\partial r} - \frac{\hat{U}^r}{r^2} + \frac{1}{r^2} \frac{\partial^2 \hat{U}^r}{\partial \varphi^2} + \frac{\partial^2 \hat{U}^r}{\partial z^2} - \frac{2}{r^2} \frac{\partial \hat{U}^\varphi}{\partial \varphi} \right) \end{aligned} \quad (9.15)$$

Circumferential adjoint momentum equation (\tilde{e}^φ direction):

$$\begin{aligned}
& -\rho \left(U^r \frac{\partial \hat{U}^\varphi}{\partial r} + \frac{U^\varphi}{r} \frac{\partial \hat{U}^\varphi}{\partial \varphi} + U^z \frac{\partial \hat{U}^\varphi}{\partial z} + \frac{\hat{U}^r U^\varphi}{r} \right) \\
& + \frac{\rho}{r} \left(\hat{U}^r \frac{\partial U^r}{\partial \varphi} + \hat{U}^\varphi \frac{\partial U^\varphi}{\partial \varphi} + \hat{U}^z \frac{\partial U^z}{\partial \varphi} + \hat{U}^\varphi U^r - U^\varphi \hat{U}^r \right) = \\
& -\frac{\partial \hat{p}}{\partial \varphi} + \mu \left(\frac{\partial^2 \hat{U}^\varphi}{\partial r^2} + \frac{1}{r} \frac{\partial \hat{U}^\varphi}{\partial r} - \frac{\hat{U}^\varphi}{r^2} + \frac{1}{r^2} \frac{\partial^2 \hat{U}^\varphi}{\partial \varphi^2} + \frac{\partial^2 \hat{U}^\varphi}{\partial z^2} + \frac{2}{r^2} \frac{\partial \hat{U}^r}{\partial \varphi} \right) \quad (9.16)
\end{aligned}$$

Axial adjoint momentum equation (\tilde{e}^z direction):

$$\begin{aligned}
& -\rho \left(U^r \frac{\partial \hat{U}^z}{\partial r} + \frac{U^\varphi}{r} \frac{\partial \hat{U}^z}{\partial \varphi} + U^z \frac{\partial \hat{U}^z}{\partial z} \right) + \rho \left(\hat{U}^r \frac{\partial U^r}{\partial z} + \hat{U}^\varphi \frac{\partial U^\varphi}{\partial z} + \hat{U}^z \frac{\partial U^z}{\partial z} \right) = \\
& -\frac{\partial \hat{p}}{\partial z} + \mu \left(\frac{\partial^2 \hat{U}^z}{\partial r^2} + \frac{1}{r} \frac{\partial \hat{U}^z}{\partial r} + \frac{1}{r^2} \frac{\partial^2 \hat{U}^z}{\partial \varphi^2} + \frac{\partial^2 \hat{U}^z}{\partial z^2} \right) \quad (9.17)
\end{aligned}$$

The left-hand side terms in Eqns. (9.15) through (9.17) consist of the negative convective contributions. These are followed by the advection terms according to formulation ADV1 transformed into physical cylinder coordinates. For completeness, the alternative formulation ADV2 described in physical cylinder coordinates is provided in Appendix B.2. The right-hand side contributions, being the adjoint diffusion and pressure terms, correspond to the primal Navier–Stokes equations.

The adjoint boundary conditions are specific to the definition of the objective functional (9.12). The adjoint boundary conditions on the outer cylinder ring are

$$\hat{U}_i = \hat{U}^r = \hat{U}^\varphi = \hat{U}^z = 0 \quad \text{for } r = r_O. \quad (9.18)$$

For the inner ring carrying the objective functional, $\Gamma \subset \Gamma_O$, the boundary conditions are

$$\hat{U}_i = r_I \tilde{e}_i^\varphi \quad \text{or} \quad \hat{U}^\varphi = r_I; \quad \hat{U}^r = \hat{U}^z = 0 \quad \text{for } r = r_I. \quad (9.19)$$

The adjoint continuity equation (9.14) and the z -component of the adjoint momentum equation (9.17) are irrelevant for symmetry reasons, $\frac{\partial}{\partial \varphi}(\cdot) = \frac{\partial}{\partial z}(\cdot) = 0$. Eqns. (9.15) and (9.16) reduce to

$$\frac{\partial \hat{p}}{\partial r} = -\rho \hat{U}^\varphi \left(\frac{U^\varphi}{r} + \frac{\partial U^\varphi}{\partial r} \right), \quad (9.20)$$

$$0 = r \frac{\partial^2 \hat{U}^\varphi}{\partial r^2} + \frac{\partial \hat{U}^\varphi}{\partial r} - \frac{\hat{U}^\varphi}{r}. \quad (9.21)$$

The term underlined in Eqn. (9.20) corresponds to the standard formulation ADV1. The corresponding solution for the alternative formulation ADV2 is presented in Appendix

B.2. The analytical solution for the adjoint axis-symmetric Couette problem is obtained by integration

$$\hat{U}^\varphi(r) = \hat{\alpha} \left(\frac{r}{r_O^2} - \frac{1}{r} \right) \quad (9.22)$$

$$\hat{p}(r) = \hat{p}(r_I) + \rho \alpha \hat{\alpha} \left[\frac{r_I^2 - r^2}{r_O^2} + 2 \ln \left(\frac{r}{r_I} \right) \right], \quad (9.23)$$

with

$$\hat{\alpha} = \frac{r_I}{(r_I/r_O)^2 - 1}. \quad (9.24)$$

Results

The laminar Couette flow sketched in Figure 9.1 was solved numerically on a computational grid of 100×50 cells in circumferential and radial directions that is depicted in Figure 9.2(a). The Reynolds number based on the width $\Delta = r_O - r_I$ and the velocity of the outer ring U_O^φ was $Re = 100$. The primal solutions obtained with the LUDS scheme for the circumferential velocity component $U^\varphi(r)$ and the pressure $p(r)$ are plotted over the radius in Figure 9.3 along with the analytic solutions. Velocity and pressure distributions meet the analytical results. The adjoint problem was solved on the same grid with the LDDS convection scheme and different advection treatments ADV0–3. The nondimensional distributions of the adjoint velocity $\hat{U}^\varphi(r)$ and the adjoint pressure $\hat{p}(r)$ are shown in Figure 9.4 on top of the corresponding analytic solutions. It is interesting to note that the momentum equations for the radial and the circumferential components decouple for both primal and adjoint problem. Hence, different treatments of the advection term in the adjoint equation do not affect the circumferential component of the adjoint velocity: As shown in Figure 9.4(a), advection treatments ADV0–3 result in the same nondimensional $\hat{U}^\varphi(r)$ distribution. Only the radial equation is affected by the advection treatment: According to Figure 9.4(b), very different $\hat{p}(r)$ distributions were obtained for neglected advection (ADV0), advection formulations ADV1 and ADV2/3 derived from the primal discretisation. The numerical adjoint results coincide with the corresponding analytic solutions. Note that ADV2 and ADV3 result in the same adjoint pressure distribution. This is obvious since ADV2 and ADV3 are closely related as shown in Appendix B.3.

In more complex configurations the adjoint momentum equations can hardly be expected to decouple. On the contrary, the advective term intensifies the cross-coupling between the adjoint system of equations. Compared to the primal Navier–Stokes equations solved through a Picard iteration, the interaction between the adjoint momentum equations will probably increase.

Since the numerical solution of both flow and adjoint problem is based on Cartesian coordinates, (adjoint) momentum is redistributed between the Cartesian components by convection and advection terms during the numerical solution. In physical cylinder coordinates, the convection and advection terms do not explicitly appear in the circumferential adjoint momentum equation. It allows to derive an analytic solution to assess the numerical convection scheme. The coarse mesh of 15×60 cells in radial and circumferential

directions depicted in Figure 9.2(b) was set up in order to bring out the differences. Figure 9.5 shows the nondimensional $\hat{U}^\varphi(r)$ -distribution obtained via DDS-CDS blending and the LDDS convection-schemes. For increased DDS-CDS blending factors the numerical solution approaches the analytic result. The LDDS scheme shows the best agreement. Though the expected trend is found in the curves, one has to bear in mind that the primal solution underlying all adjoint cases is computed using the LUDS convection scheme. All adjoints were linearised about the same primal solution to compare the effect of the adjoint convection scheme. However, there is a numerical inconsistency associated with the combination of primal LUDS and dual DDS-CDS blending.

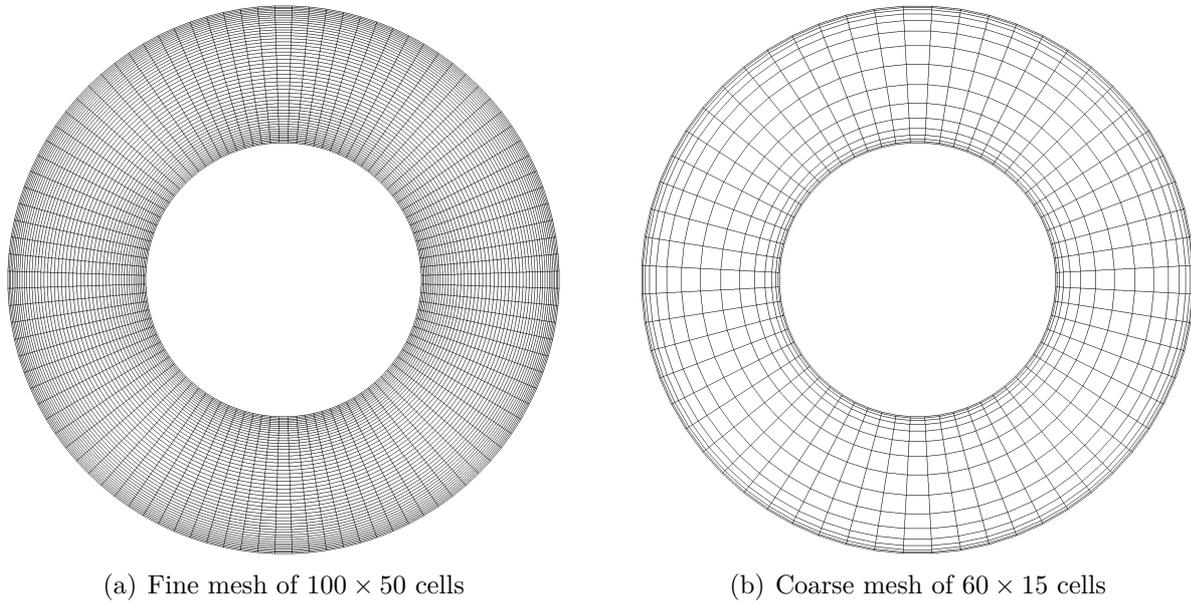


Figure 9.2.: Computational grids for axis-symmetric Couette flow

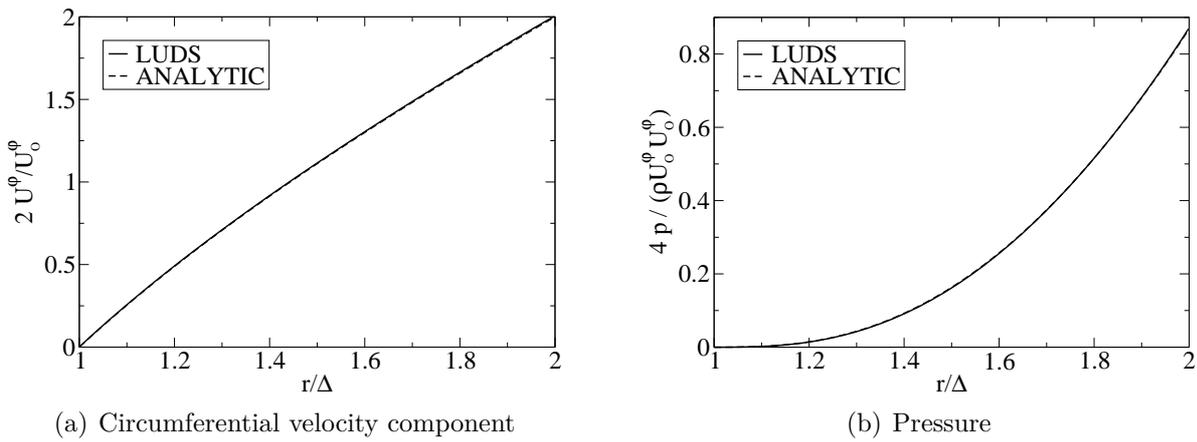


Figure 9.3.: Axis-symmetric Couette flow; nondimensional primal solution obtained on fine mesh compared against analytic solution

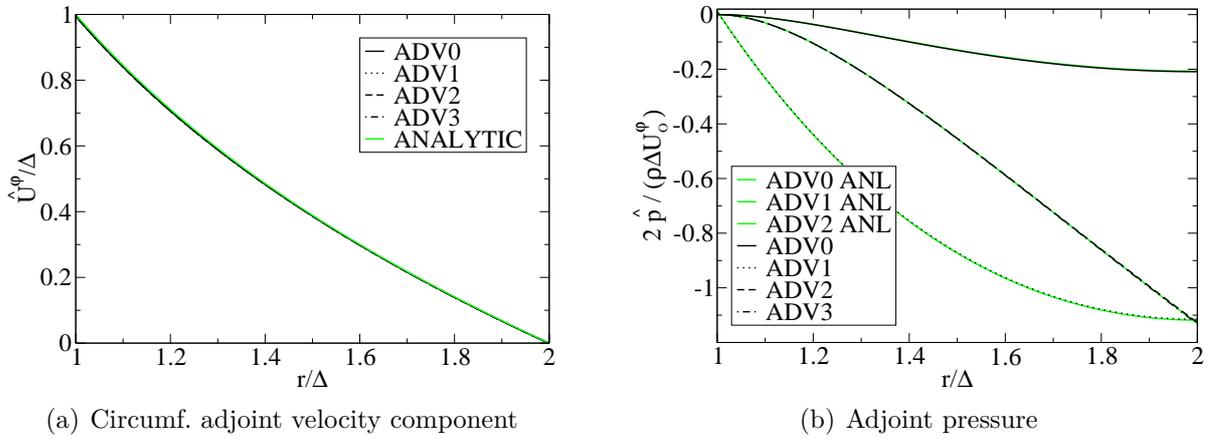


Figure 9.4.: Axis-symmetric Couette flow; nondimensional distributions of $\hat{U}^\varphi(r)$ and $\hat{p}(r)$ for advection schemes ADV0–3 compared against analytic solution

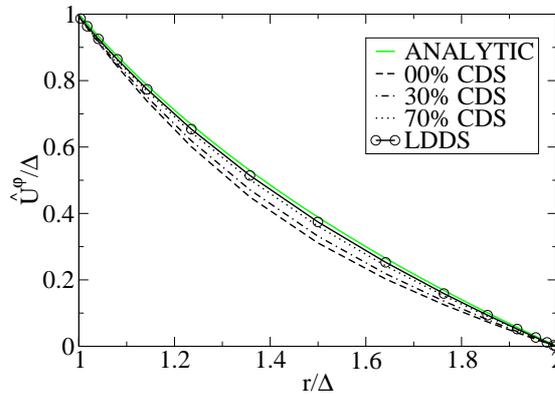


Figure 9.5.: Axis-symmetric Couette flow; nondimensional distribution of $\hat{U}^\varphi(r)$ obtained from different convection schemes (DDS, 30% CDS, 70% CDS, LDDS) compared against analytic solution; radial mesh resolution of coarse grid indicated by circles

9.2. Direct-Differentiation Method

Simple internal and external flow cases are considered to verify the adjoint method against the direct-differentiation method, wherein the linearised Navier–Stokes equations are solved once per degree of freedom. The internal flow studies are based on the power loss criterion (4.38) evaluated for a bent duct and a diffuser geometry. The verification study for external flow is concerned with a quarter cylinder in free-stream conditions. A drag force objective is evaluated either on the interior or the exterior domain boundaries as described in Section 6.5.

Bent Duct in Laminar Flow

The bent duct sketched in Figure 9.6 was investigated first for 2D, laminar flow. It resembles the S-shaped duct configurations investigated by Zymaris et al. [170, 171]. The geometry used in this study is composed of circular ring segments and straight lines alone, so that it can be reproduced easily. The computational mesh depicted in Figure 9.7 consisted of 50×90 cells. A uniform inflow velocity was prescribed resulting in a bulk Reynolds number $Re = U_{\text{in}} d/\nu = 500$. At the outflow, the pressure was set to zero. According to Eqn. (4.37), the power loss between inlet and outlet, $\Gamma_O = \Gamma_{\text{in}} \cup \Gamma_{\text{out}}$, is reduced by maximising the power objective functional:

$$J_{\Gamma} = \int_{\Gamma_O} \left(p + \frac{\rho}{2} U_i^2 \right) U_j d\Gamma_j. \quad (9.25)$$

Primal and adjoint convective fluxes of momentum were approximated according to the LU(D)DS scheme (7.17). The distribution of the primal pressure coefficient, $C_p = 2p/(\rho U_{\text{in}}^2)$, is plotted in Figure 9.8 together with the primal and the adjoint velocity fields. The flow field features a recirculation zone on the lower wall. Note that the derivatives are zero at the point of separation. In Figure 9.9, the distribution of the sensitivity derivative is plotted over the channel walls for advection treatment ADV1. In order to verify the adjoint code, the derivative obtained from the adjoint method was compared against its counterpart calculated via the direct-differentiation method. The linearised Navier–Stokes equations were solved for boundary conditions perturbed patch-wise over the tunnel walls Γ_D ; that is, a total of 158 calculations were necessary to compute the full sensitivity derivative over upper and lower walls of the duct in the range of $-1 < x/d < 4.5$.

Figure 9.10(a) shows the distributions of the nondimensional sensitivity derivative

$$G^* = G \frac{d}{\rho U_{\text{in}}^3},$$

calculated from different advection treatments ADV0–3. If the advection term is neglected (ADV0), the maximum amplitudes on upper and lower walls are almost equivalent. The distribution of the sensitivity derivative has a different character when the advection term is taken into account through formulations ADV1–3. The curves for ADV1–3, Figure 9.10(a), show the main differences in the peak prediction and on the upper wall for

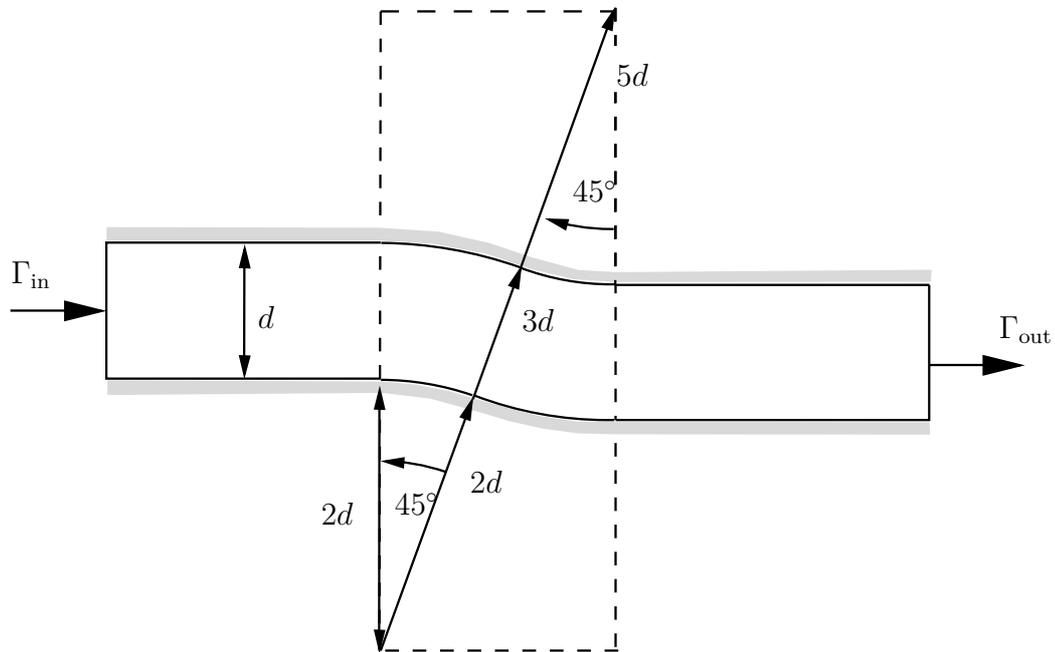


Figure 9.6.: Setup and dimensions for a bent duct configuration

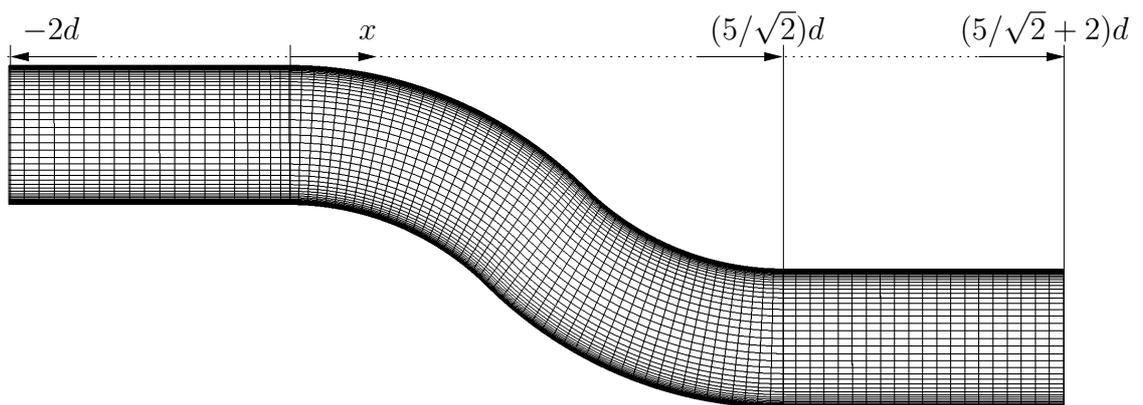


Figure 9.7.: Computational grid of 50×90 cells for a bent duct configuration

$2d < x < 2.5d$. The adjoint-based sensitivity derivative is compared against its counterpart predicted by the direct-differentiation method using consistent approximations for convection and advection terms in Figures 9.10 (b–e). In the direct-differentiation method, the volume-based approximation (7.86) of the advection term is referred to as ADV1. The adjoint-based ADV1/2 results are compared against the ADV1 results of the direct-differentiation method, since an ADV2 description obtained via integration by parts does not exist in the direct-differentiation method. According to Figure 9.10(c), ADV1 shows the best agreement. ADV2 (d) and ADV3 (e), which are closely related to each other as demonstrated in Appendix B.3, mainly deviate in the peak predictions and upstream, particularly on the upper wall surface.

Bent Duct in Turbulent Flow with Low-Re Wall Boundary Condition

For the configuration introduced above, the bulk Reynolds number was increased to $Re = 20,000$. The Wilcox $k-\omega$ turbulence model was used in conjunction with low-Re wall boundary conditions. Figure 9.13(b) shows the y^+ distribution over upper and lower tunnel walls. A uniform velocity was prescribed at the inlet together with the turbulence quantities

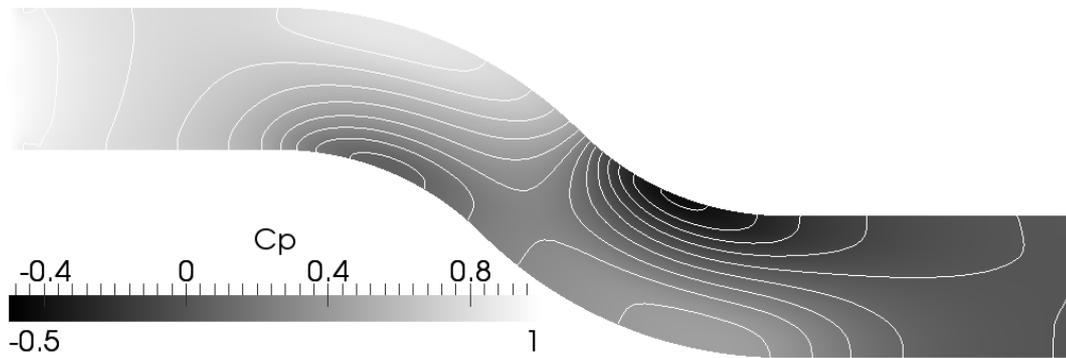
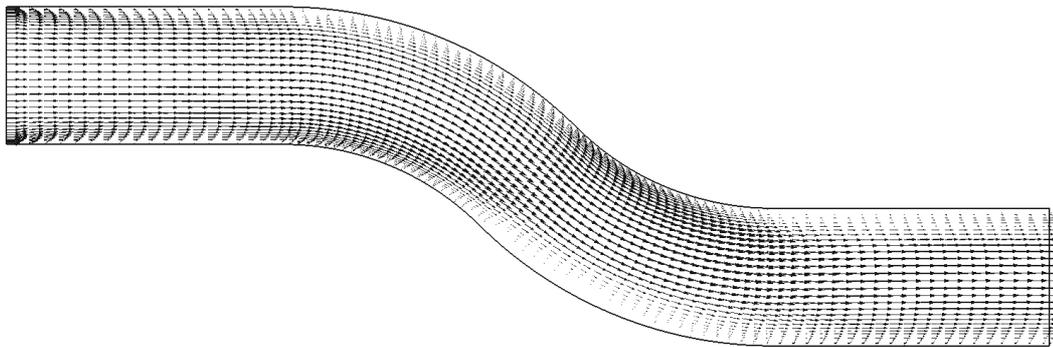
$$k = \frac{3}{2} Tu^2 U_{\text{in}}^2 \quad \text{and} \quad \omega = \sqrt{\frac{3}{2}} \frac{Tu U_{\text{in}}}{C_\mu^{1/4} L_T},$$

determined from empirical relationships for pipe flow

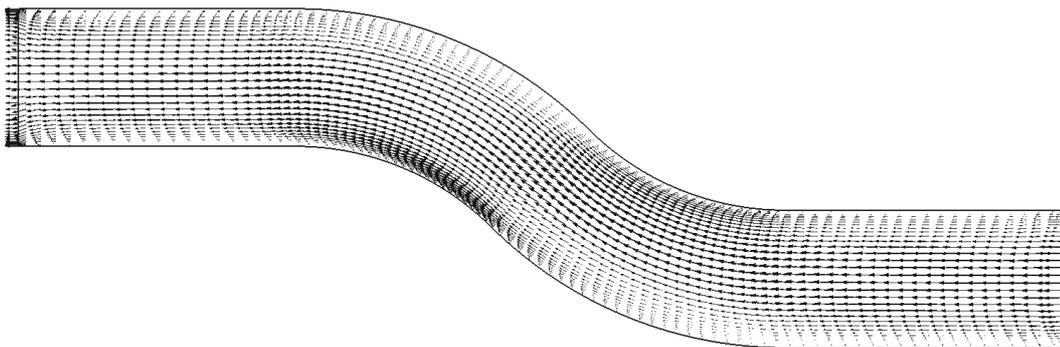
$$Tu = 0.16 Re^{-1/8} \quad \text{and} \quad L_T = 0.07 d. \quad (9.26)$$

The frozen-turbulence assumption was consistently applied for both the adjoint and the direct-differentiation method. The convective momentum fluxes in primal, linearised and adjoint PDE were discretised according to the LU(D)DS scheme. Figure 9.11 shows the adjoint-based distribution of the sensitivity derivative of the power objective (9.25) using ADV3 and how it correlates with the separation bubble ranging from x_S to x_R . For the turbulent case the sensitivity distribution clearly reflects the onset of separation, suggesting a wider duct shape upstream of the separation point, a zero change in the separation point and a reduced duct width downstream, which will intuitively delay the separation.

Figure 9.13(c) depicts the adjoint-based sensitivity derivative obtained by ADV0–3. For ADV0 the amplitudes on upper and lower walls are very similar. The curves for ADV1–3 have a different character. The differences between the sensitivity derivatives for ADV0 and ADV1 can in parts be explained by the shapes of the corresponding adjoint boundary-layer profiles plotted in Figure 9.12. Whereas for ADV0 the adjoint boundary layer resembles the primal one, ADV1 causes a velocity peak in the boundary layer. In the recirculation zone, the primal velocity boundary layer has non-zero derivatives $U^{t,t} = -U^{n,n}$ and $U^{n,t}$. In this area, the adjoint boundary-layer flow is essentially driven by the source of momentum acting in the wall-tangential direction, $\rho \hat{U}_j U_{j,i} t_i = \rho \hat{U}^t U^{t,t} + \rho \hat{U}^n U^{n,t}$ (ADV1). The influence is convected in the upstream direction. Accordingly, the adjoint wall shear rates for ADV0 and ADV1 are different, leading to different sensitivity

(a) Primal pressure coefficient C_p 

(b) Primal velocity field



(c) Adjoint velocity field

Figure 9.8.: Laminar flow through a bent duct at $Re = 500$

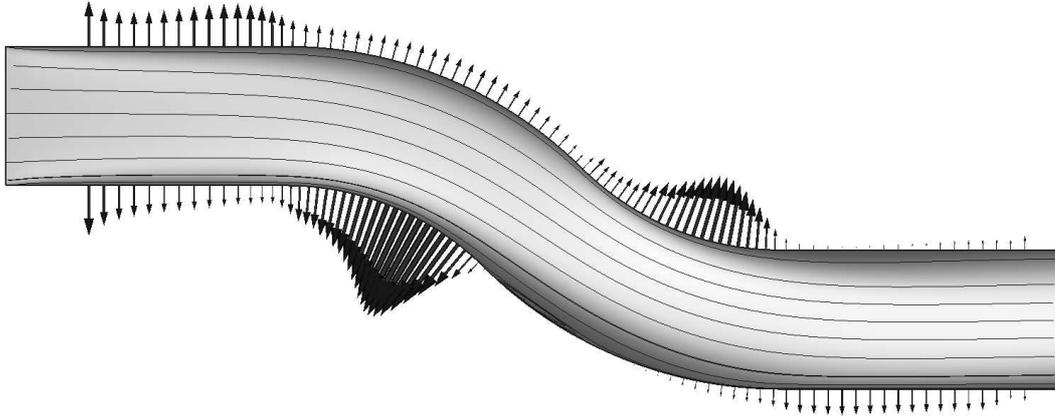


Figure 9.9.: Laminar flow through a bent duct at $Re = 500$; derivative G with velocity streamlines and magnitude

derivatives calculated from the adjoint gradient equation (6.39) or its discrete form (8.68). The adjoint method is verified against the direct-differentiation method in Figures 9.13(d) through 9.13(g). The sensitivity derivatives obtained from both methods oscillate at the scale of the boundary patch length. The oscillations are more pronounced in the adjoint case. This is expected to be caused by the adjoint gradient formulation (6.39), which is a differential expression including the product of primal and adjoint wall shear rates, both of which are not smooth in the low-Re calculation. The rough distribution of the primal wall shear rate is reflected in the y^+ plot in Figure 9.13(b). In the direct-differentiation method, the sensitivity derivative was obtained by integration over Γ_O , which led to smoother distributions though the boundary conditions on the boundary patches of Γ_D were individually perturbed. The wider length scales predicted by the direct-differentiation method and the adjoint method are in fair agreement. ADV0 and ADV1 show the best agreement. The main differences between the ADV2 and ADV3 curves are found in the upper and lower wall peaks and upstream. A similar trend was observed in the laminar case. The local extrema predicted by the direct-differentiation method are more distinct than the adjoint-based extrema.

The numerical experiment was repeated with a refined low-Re mesh of 50×240 cells depicted in Figure 9.14(a). The numerical resolution of the boundary curvature was improved by increasing the number of mesh points over the bent section from 50 to 200; in the transverse direction the mesh remained unchanged. Compared to the coarser grid, the primal y^+ distribution plotted in Figure 9.14(b) is significantly smoother. Also the oscillations in the adjoint-based sensitivity derivatives based on ADV0–3, cf. Figure 9.14(c), were considerably reduced. According to Figures 9.14(d) to 9.14(g), the sensitivity derivatives based on the direct-differentiation method and the adjoint method are in good agreement.

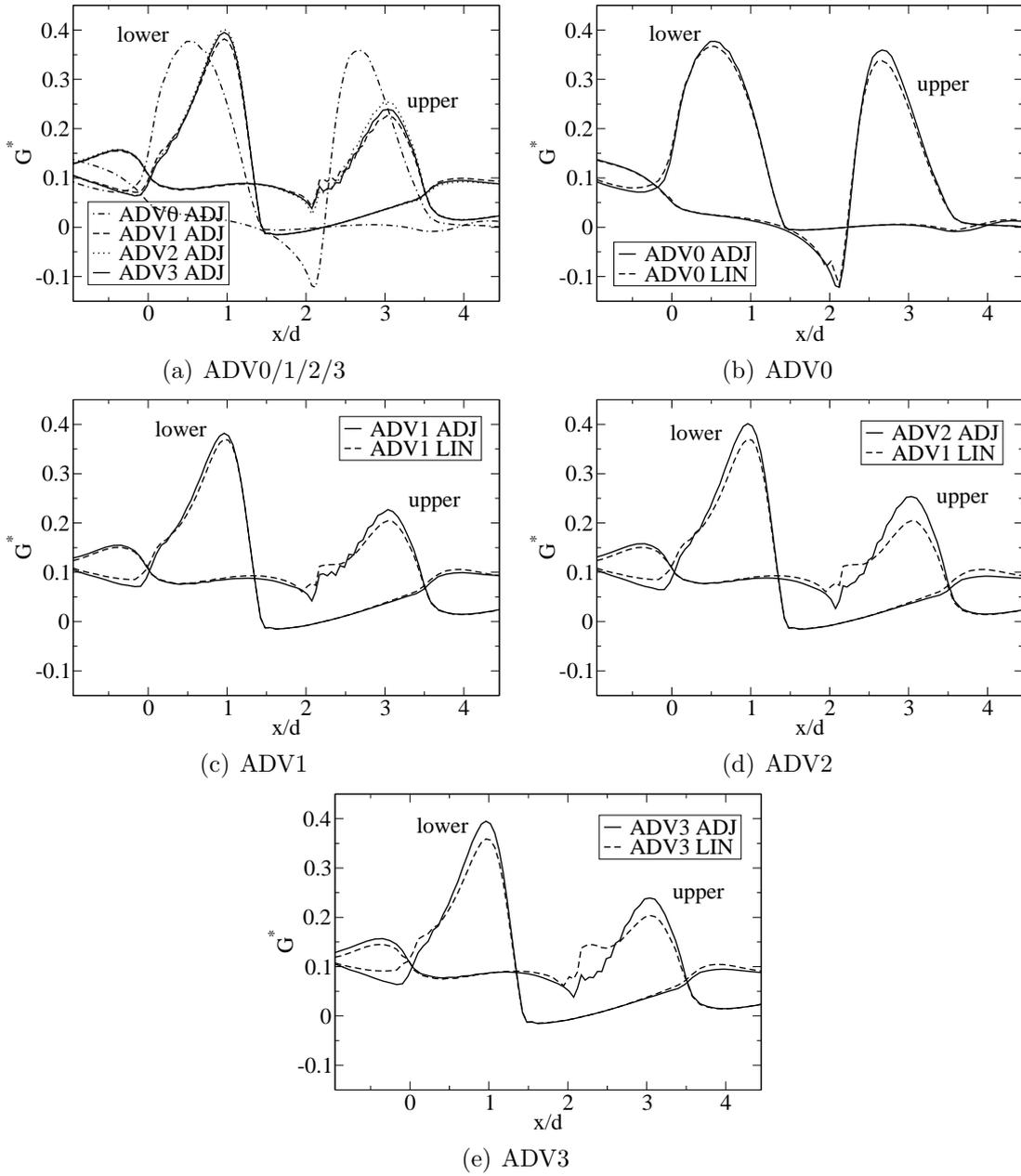


Figure 9.10.: Laminar flow through a bent duct at $Re = 500$; nondimensional derivative G^* according to adjoint (ADJ) and direct-differentiation method (LIN) using advection schemes ADV0/1/2/3

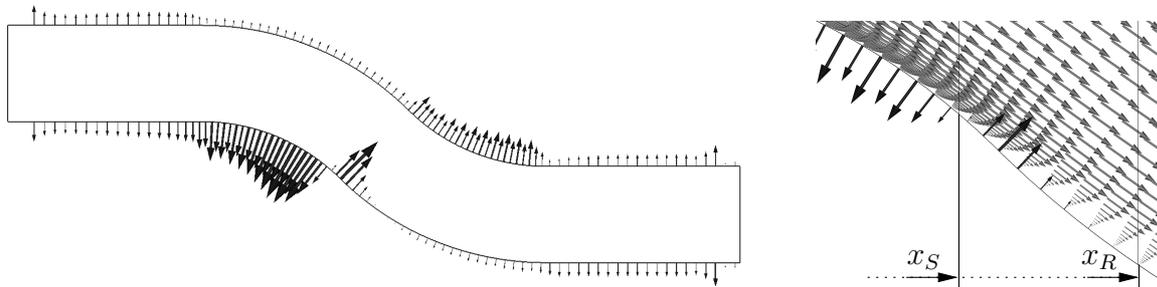


Figure 9.11.: Bent duct at $Re = 20,000$ with low-Re wall boundary condition; derivative G (left); correlation with flow separation between $x_S = 1.31d$ and $x_R = 1.85d$ (right)

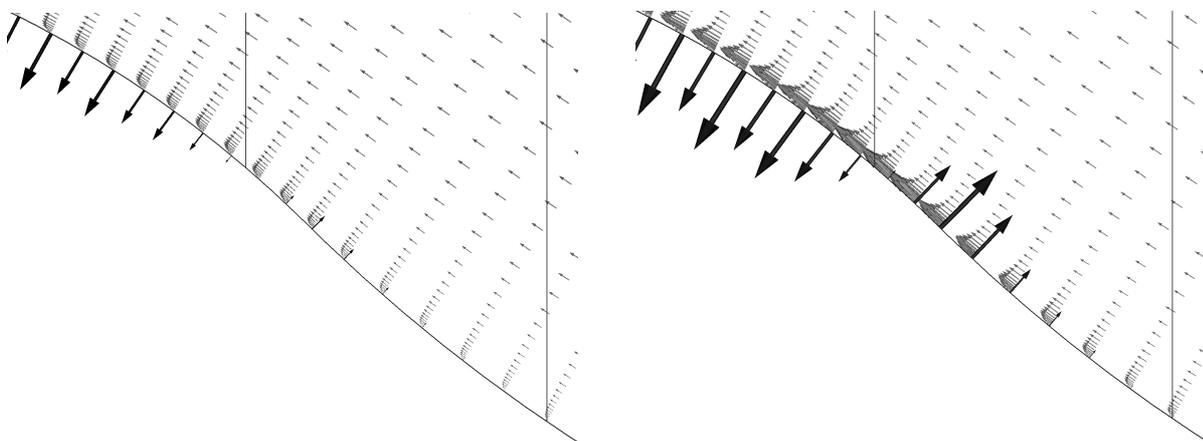


Figure 9.12.: Bent duct at $Re = 20,000$ with low-Re wall boundary condition; adjoint velocity field \hat{U} and derivative G according to ADV0 (left) and ADV1 (right); separation bubble outlined by vertical lines

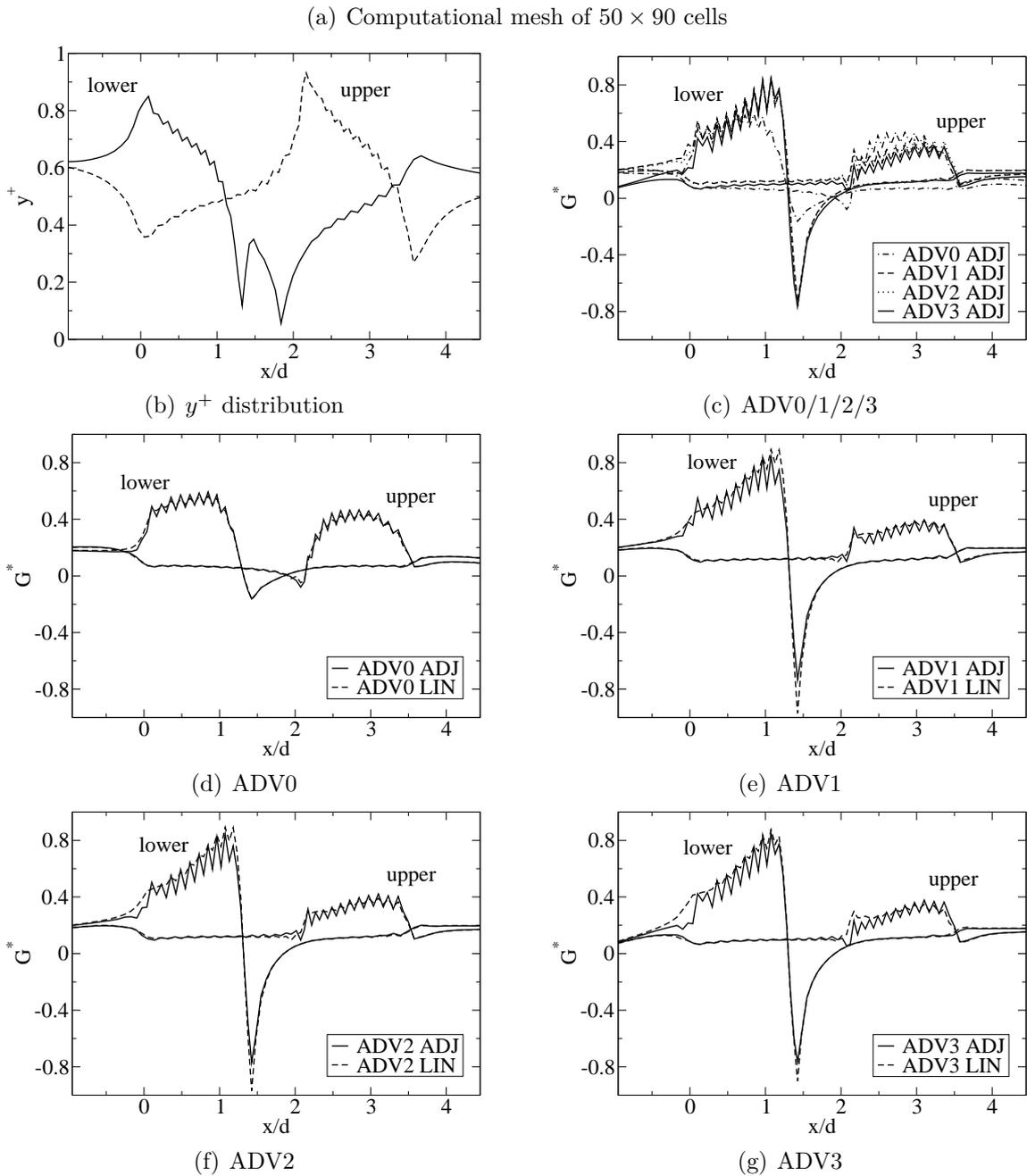
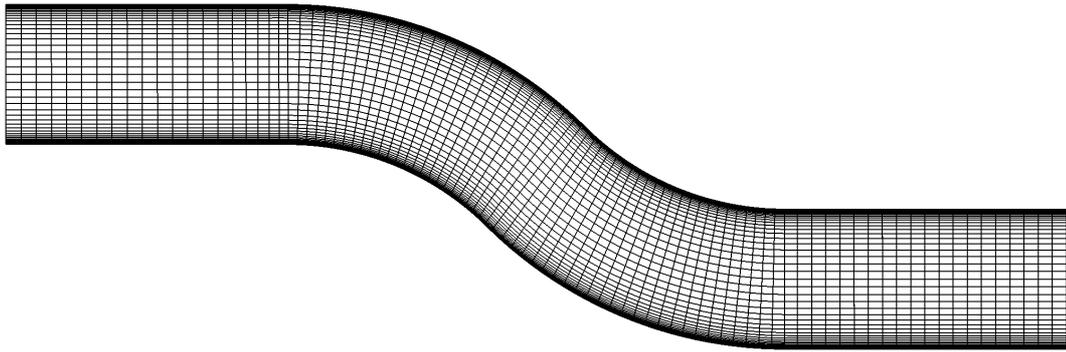
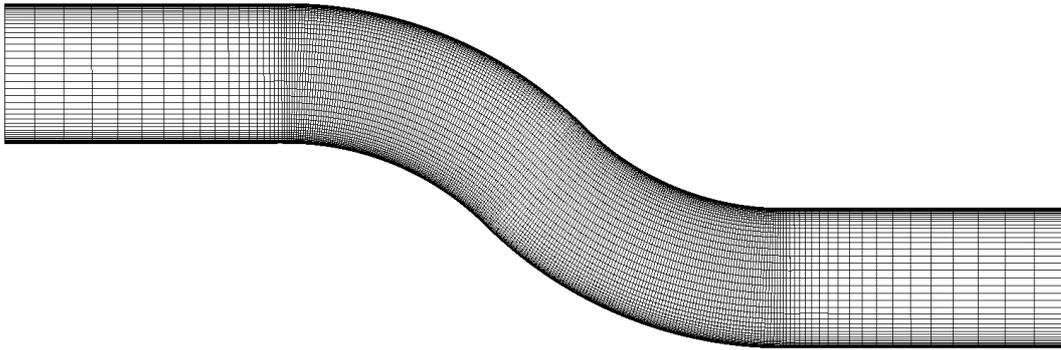
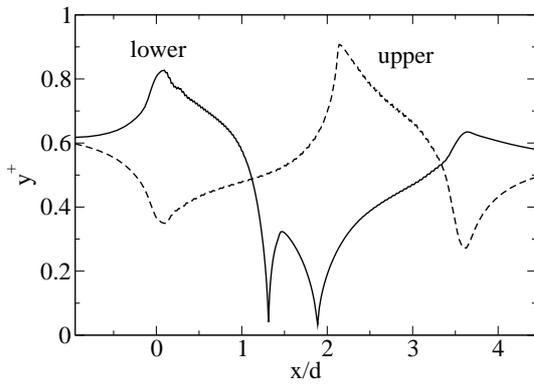


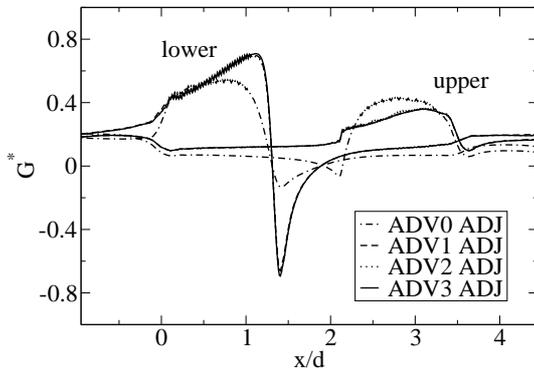
Figure 9.13.: Bent duct at $Re = 20,000$ with low- Re wall boundary condition; nondimensional derivative G^* according to adjoint (ADJ) and direct-differentiation (LIN) method using advection schemes ADV0/1/2/3



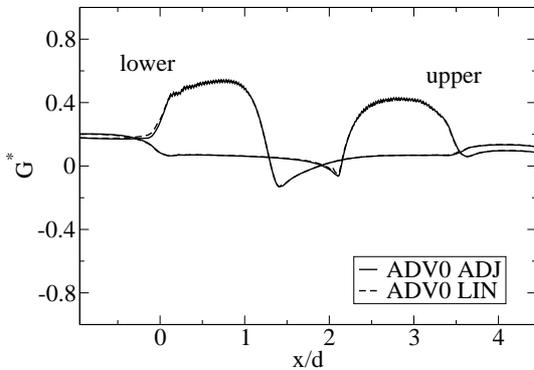
(a) Computational mesh refined to 50×240 cells



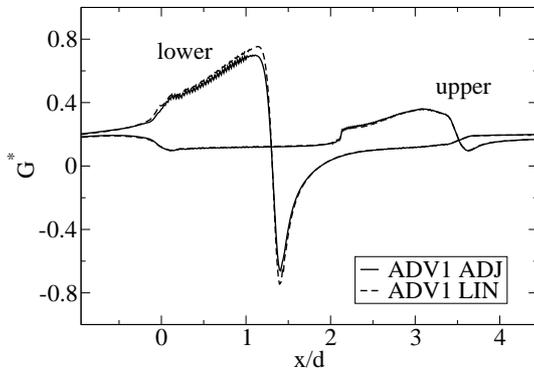
(b) y^+ distribution



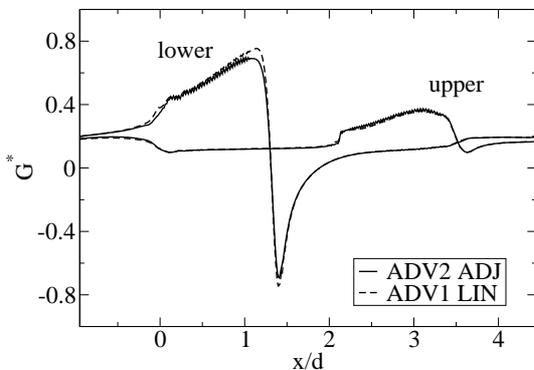
(c) ADV0/1/2/3



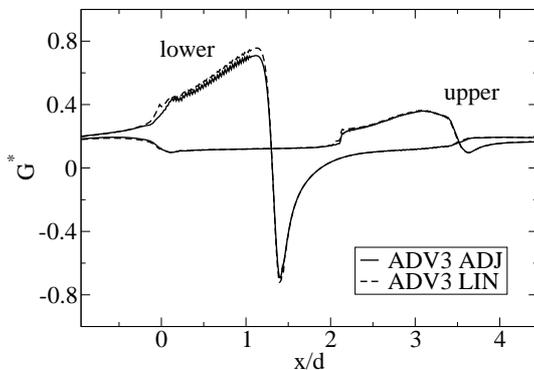
(d) ADV0



(e) ADV1



(f) ADV2



(g) ADV3

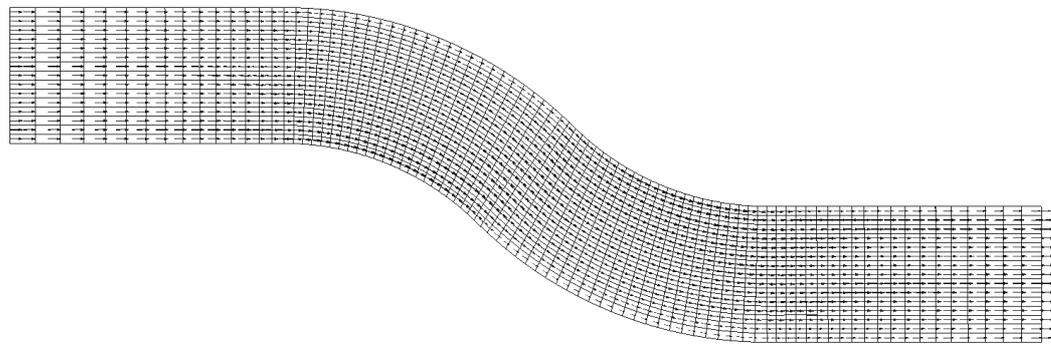
Figure 9.14.: Bent duct at $Re = 20,000$ with refined mesh and low- Re wall boundary condition; nondimensional derivative G^* according to adjoint (ADJ) and direct-differentiation (LIN) method using advection schemes ADV0/1/2/3

Bent Duct in Turbulent Flow with High-Re Wall Boundary Condition

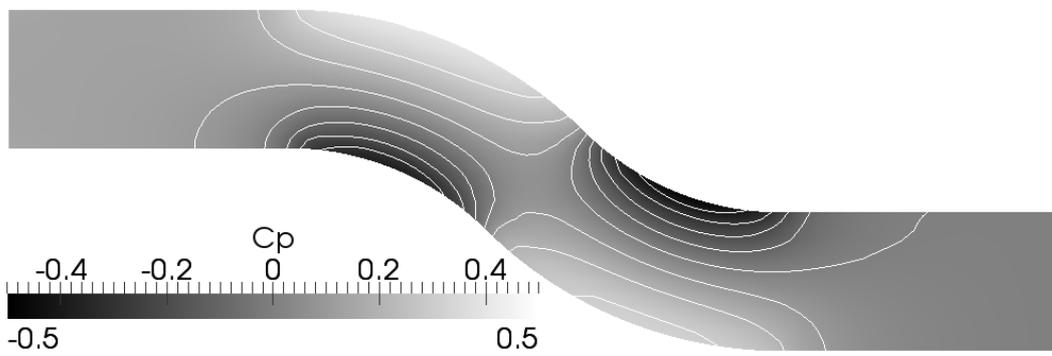
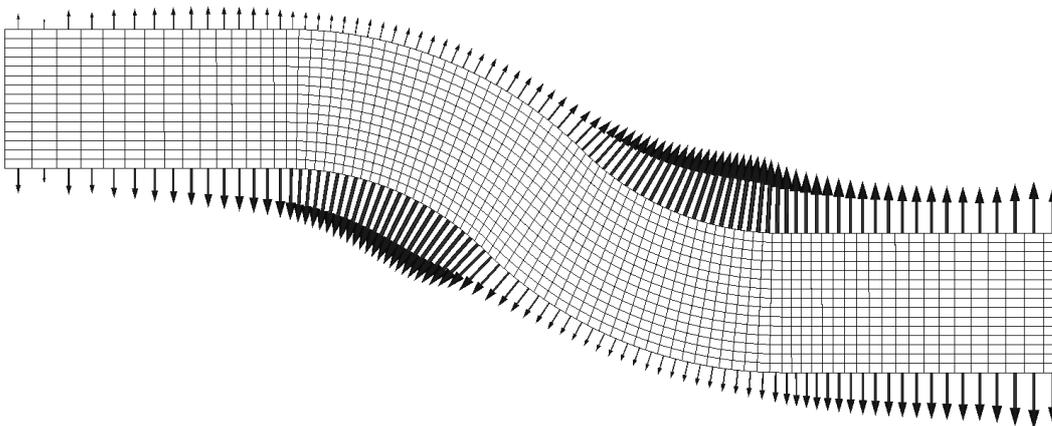
The same test case was used to verify the implementation of the high-Re wall boundary condition. The primal flow field calculated on the high-Re mesh of 15×85 cells is depicted in Figures 9.15(a) and 9.15(b). Figure 9.15(c) shows the sensitivity distribution (ADV3) obtained in the high-Re case, which is smooth compared to the low-Re cases above. Also the underlying y^+ distribution plotted in Figure 9.16(a) is smooth. The adjoint-based sensitivity plots for ADV0 and ADV3 are compared in Figure 9.16(b). Formulations ADV1 and ADV2 caused strong oscillations of the wall-normal component of the adjoint velocity next to the wall so that they are not presented here. The differences between ADV0 and ADV3 are smaller than in the low-Re case. This is expected to be a consequence of not resolving the wall boundary layers which are responsible for the main difference between ADV0 and ADV1–3 in the low-Re case. The adjoint-based derivative according to Eqn. (8.69) is verified against the result of the direct-differentiation method in Figures 9.16(c) and 9.16(d). In the direct-differentiation method, source terms were successively applied to the linearised momentum equations in the wall-adjacent CVs in the range $-1 < x/d < 4.5$, compare Eqn. (7.103).

Diffuser in Turbulent Flow

A 2D diffuser configuration was considered next to verify the adjoint approach. The diffuser sketched in Figure 9.17 is composed of straight lines and circular arcs alone, which can be modelled easily. It resembles the configuration investigated by Zymaris et al. [171], for which an explicit geometry definition was not available. Inlet and outlet areas are different so that the kinetic energy contribution to the objective functional (9.25) plays an important role in the budget of energy. This case features a symmetry boundary condition which has not been investigated in the previous cases. The bulk Reynolds number based on the uniform inlet velocity was $Re = U_{\text{in}} d/\nu = 20,000$. At the outflow the pressure was set to zero. The computational mesh shown in Figure 9.18(a) consisted of 30×60 cells. The Wilcox $k-\omega$ turbulence model was used in conjunction with low-Re wall boundary conditions. The convective fluxes of (adjoint) momentum were discretised by means of the LU(D)DS scheme. The primal velocity field depicted in Figure 9.18(b) shows a huge recirculation zone. The pressure coefficient, $C_p = 2p/(\rho U_{\text{in}}^2)$, is plotted in Figure 9.18(c). Figure 9.19 illustrates the primal and the adjoint (ADV3) velocity fields in the recirculation zone together with the sensitivity derivative plotted over the boundary walls. The sensitivity distributions for ADV0/1/3 are compared in Figure 9.20(a). When the advection term is neglected (ADV0), the peaks are strongly underestimated. Figures 9.20(b), 9.20(c) and 9.20(d) verify the adjoint-based derivatives against the direct-differentiation method. The over-all agreement is good—the major deviations are found upstream of the separation, particularly for advection scheme ADV3.



(a) Primal velocity field

(b) Primal pressure coefficient C_p 

(c) Sensitivity derivative (ADV3)

Figure 9.15.: Bent duct at $Re = 20,000$ with high-Re wall boundary condition

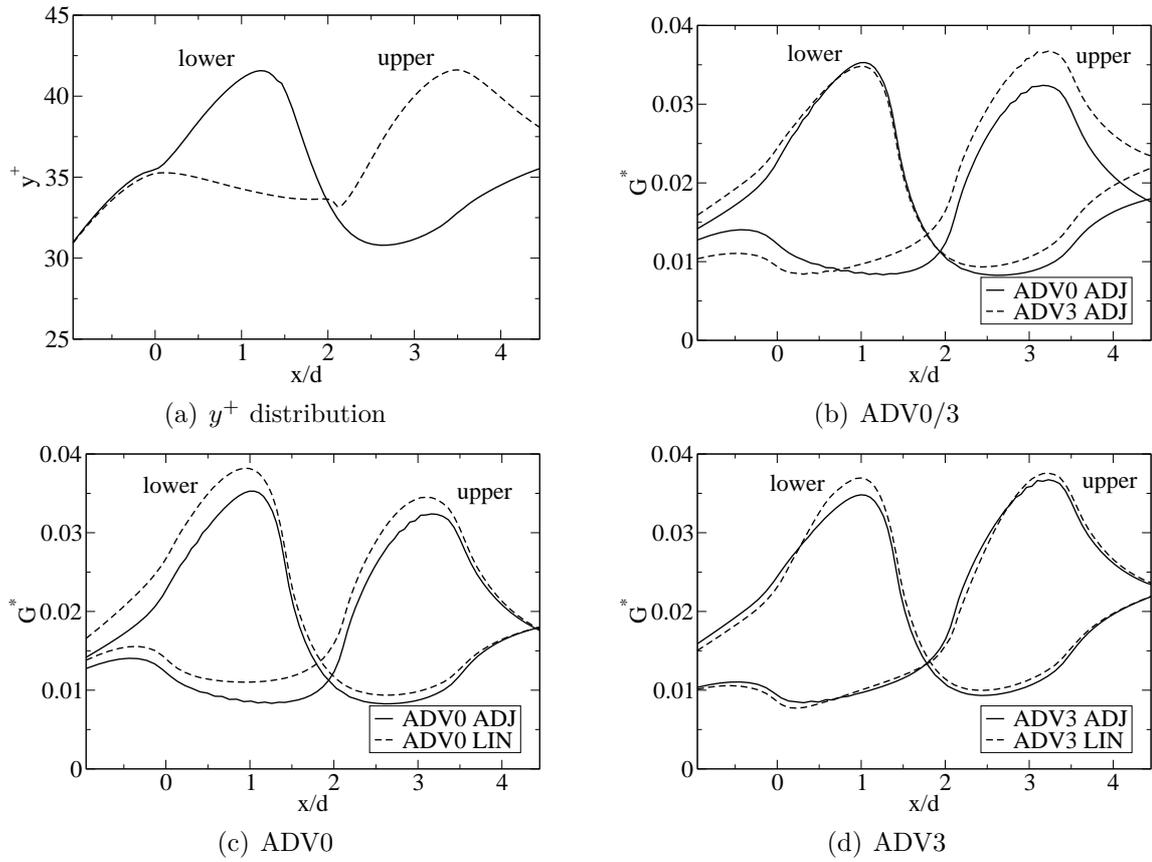


Figure 9.16.: Bent duct at $Re = 20,000$ with high-Re wall boundary condition; nondimensional derivative G^* according to adjoint (ADJ) and direct-differentiation (LIN) method using advection schemes ADV0/3

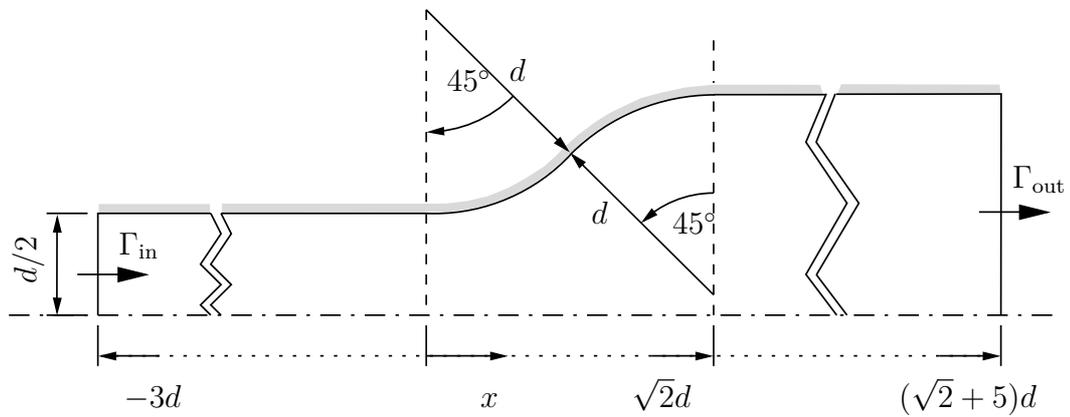


Figure 9.17.: Setup and dimensions of diffuser configuration

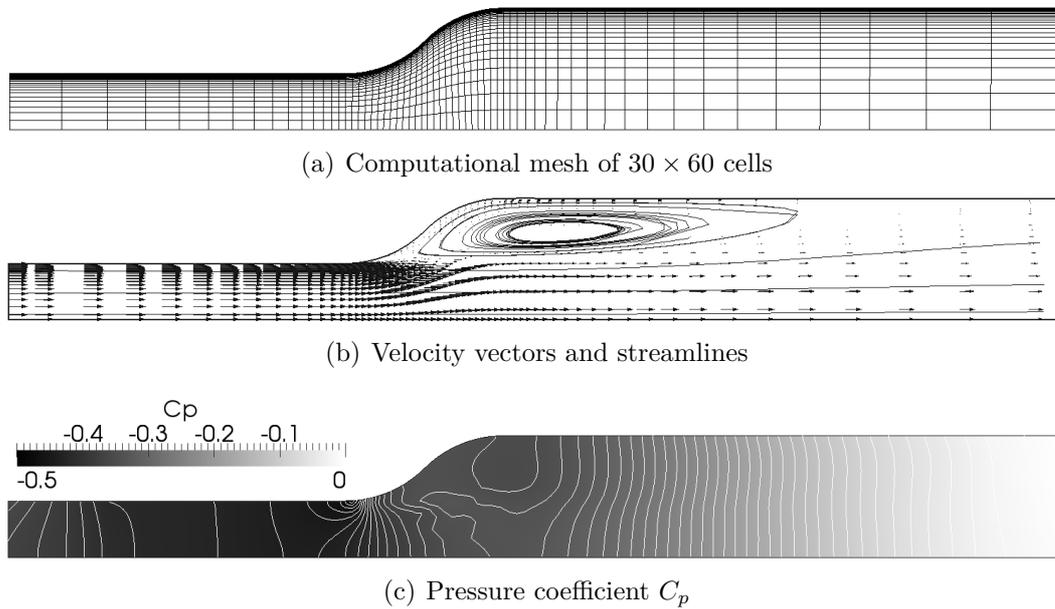


Figure 9.18.: Diffuser at $Re = 20,000$ with low-Re wall boundary condition

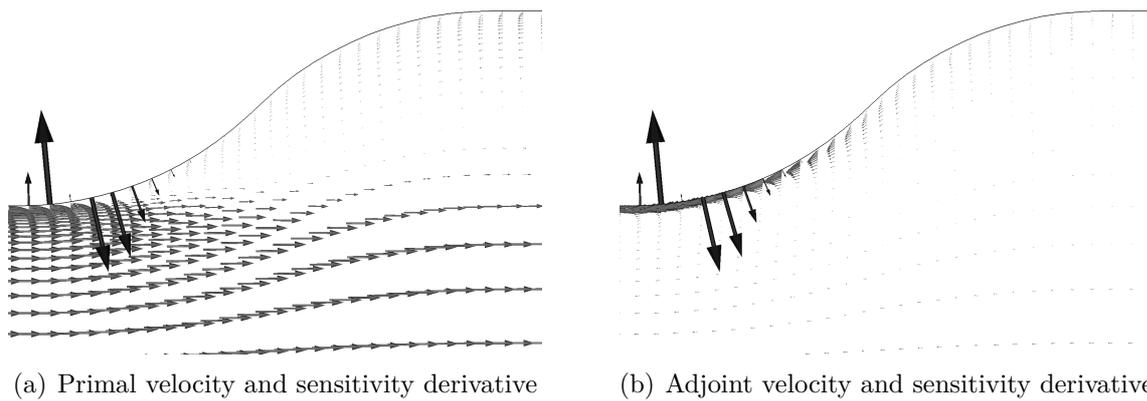


Figure 9.19.: Diffuser at $Re = 20,000$ with low-Re wall boundary condition; recirculation zone

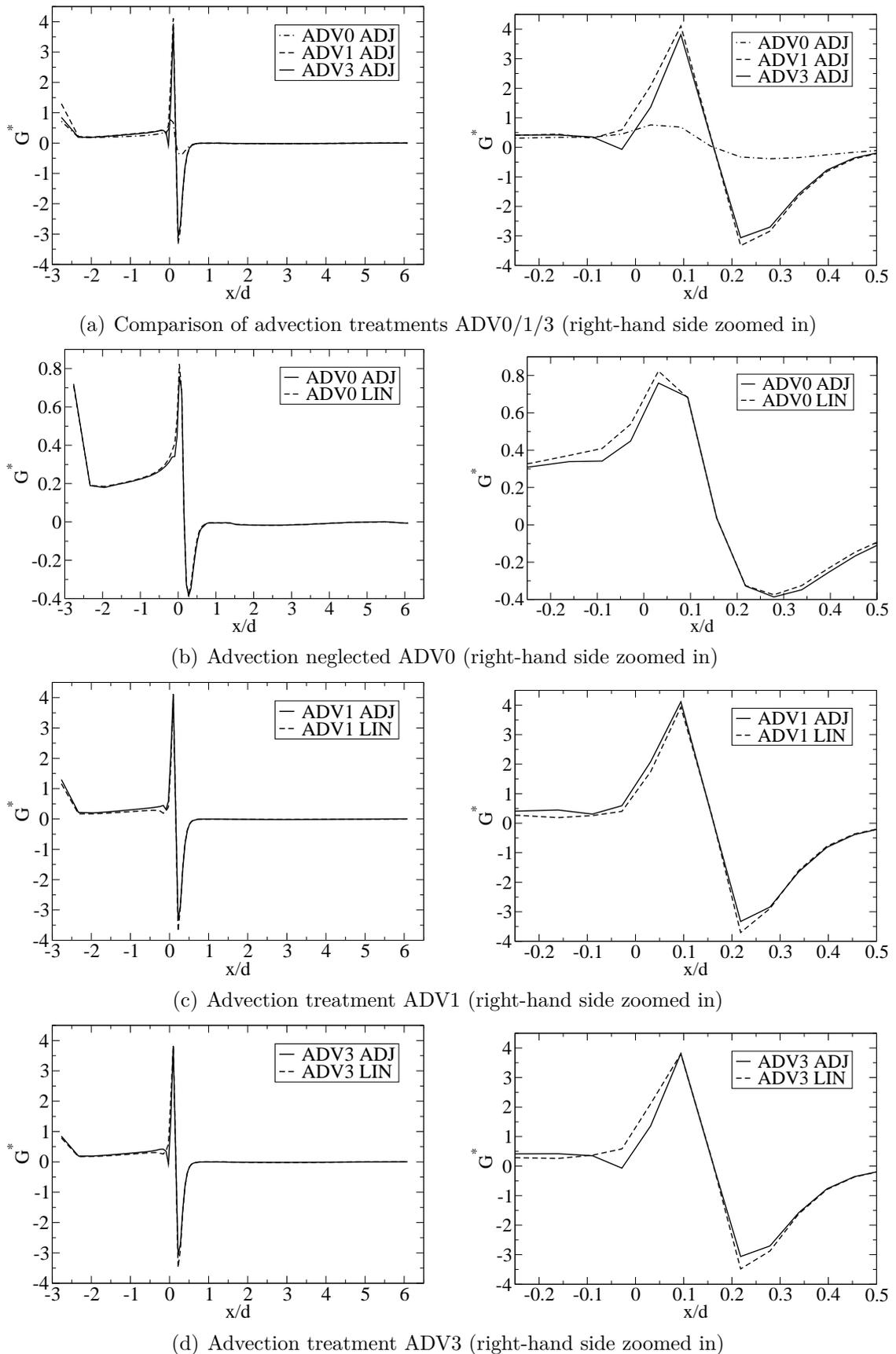


Figure 9.20.: Diffuser at $Re = 20,000$ with low- Re wall boundary condition; nondimensional derivative G^* according to adjoint (ADJ) and direct-differentiation (LIN) method using advection schemes ADV0/1/3

Quarter Cylinder in Free-Stream

The last verification sample is a 2D quarter cylinder in turbulent free-stream conditions. The blunt-body configuration featuring a distinct separation point was selected in order to verify the approach for external flow problems. The Wilcox $k\text{-}\omega$ turbulence model was used with a low-Re wall boundary condition imposed on the cylinder walls. The computational grid is shown in Figure 9.21 with a system of physical cylinder coordinates (r, θ) aligned with the cylinder axis. The diameter of the full cylinder is denoted by d . The flow domain extended $15d$ upstream and to the side and $20d$ in the downstream direction. A symmetry plane was defined on the centreline. The Reynolds number based on the inlet velocity and the diameter of the full cylinder was $Re = U_{\text{in}}d/\nu = 10^6$. At the outlet, the pressure value was set to zero. The LU(D)DS scheme was used to evaluate the convective face fluxes of (adjoint) momentum. Figure 9.22 illustrates the primal flow by velocity streamlines and contours of the pressure coefficient $C_p = 2(p - p_{\text{out}})/(\rho U_{\text{in}}^2)$. The objective functional was the drag force acting on the cylinder surface,

- either evaluated by integration over the interior (INT) boundaries, i.e. over the cylinder surface (Γ_O)

$$J = \int_{\Gamma_O} j_{\Gamma} d\Gamma \quad \text{with} \quad j_{\Gamma} = -n_j \pi_{ij} \delta_{1i}, \quad (9.27)$$

- or by integration over the exterior (EXT) boundaries (Γ_O) of the solution domain

$$J = \int_{\Gamma_O} j_{\Gamma} d\Gamma \quad \text{with} \quad j_{\Gamma} = n_j (-\rho U_i U_j + \pi_{ij}) \delta_{1i}. \quad (9.28)$$

The adjoint pressure and velocity fields obtained by INT and ADV1 are depicted in Figure 9.23. Extreme values of both the adjoint velocity and the adjoint pressure are observed in the point of separation ($\theta = 0, r = d/2$). Next to the cylinder surface the adjoint pressure distribution is dominated by the advection term ADV1. The distributions of the nondimensional sensitivity derivatives

$$G^* = G \frac{d}{\rho U_{\text{in}}^2} \quad \text{on} \quad \Gamma_D \quad (9.29)$$

over the circular front, $0 < \theta < \pi/2$ and $r = d/2$, based on interior and exterior force declarations and advection schemes ADV0/1/3 are plotted in Figures 9.24(b) to 9.24(e). All sensitivity curves obtained with low-Re wall boundary conditions oscillate. The ADV0 distributions clearly differ from the corresponding ADV1/3 curves. Figures 9.24(c) through 9.24(e) compare the nondimensional derivative G^* obtained via the adjoint method (INT/EXT) against the direct-differentiation method (INT). Once again, the distribution predicted by the direct-differentiation method is much smoother than the adjoint-based results. For ADV0 the adjoint-based results (INT/EXT) are in good agreement with the direct-differentiation method (INT). The agreement of the adjoint and the direct-differentiation method is fair for advection treatments ADV1/3 consistently applied.

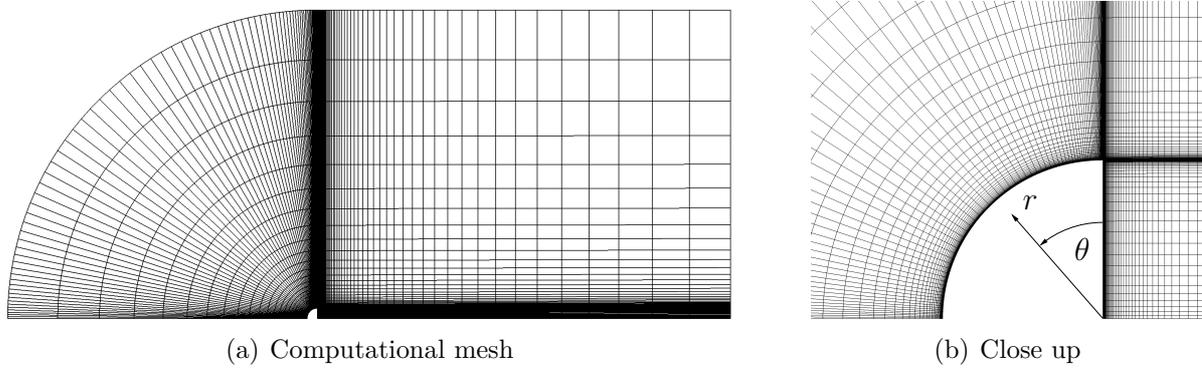


Figure 9.21.: Quarter circle in free flow at $Re = 10^6$; computational grid of 9,600 cells for low-Re wall boundary condition

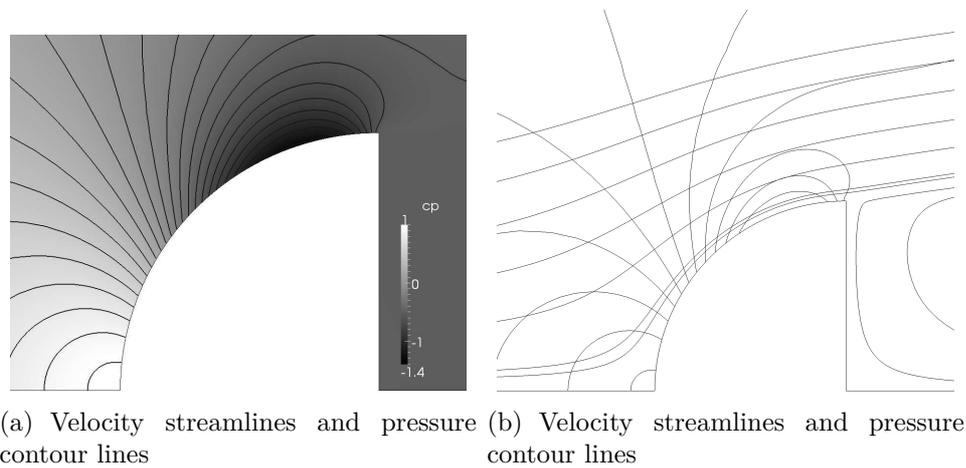


Figure 9.22.: Quarter cylinder in free flow at $Re = 10^6$ with low-Re wall boundary condition; primal flow

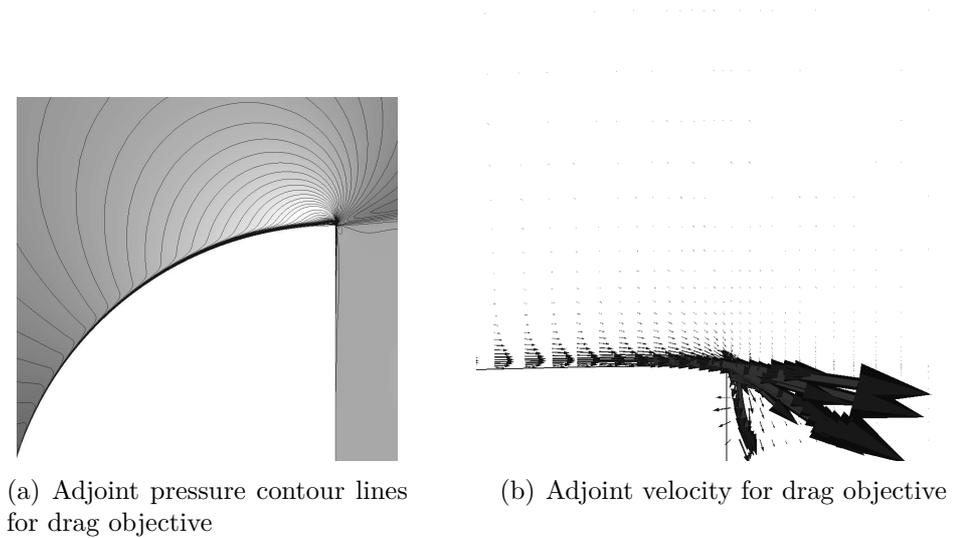


Figure 9.23.: Quarter cylinder in free flow at $Re = 10^6$ with low-Re wall boundary condition; adjoint solution using advection scheme ADV1 and interior (INT) definition of objective functional

Concluding Remarks

The adjoint solution was validated against analytic solutions found for an axis-symmetric Couette problem. The radial distributions of the adjoint velocity and pressure distributions coincide with the corresponding analytical solutions for different adjoint convection and advection treatments.

The adjoint method was also verified against the direct-differentiation method to address more realistic configurations. This allows to verify the implementations of (i) the adjoint boundary conditions declared in the adjoint pre-processing for the investigated objective functional, (ii) the adjoint field equations, and (iii) the evaluation of the sensitivity derivative in the adjoint post-processing. Internal and external flow cases were considered in laminar and turbulent flow using both low- and high-Re wall boundary conditions. The general agreement of the sensitivity derivatives based on the adjoint method and the direct-differentiation method is good, depending on the application, the mesh and the numerical settings, in particular the advection treatment. The deviations observed in conjunction with high-Re wall boundary conditions may be a consequence of the rather coarse high-Re grids, which may have affected the accuracy of both the direct-differentiation method and the adjoint method.

The sensitivity derivatives obtained for interior (INT) and exterior (EXT) definitions of force criteria according to Section 6.5 are in fair agreement. Since the objective surface is not subject to shape variations in the EXT configuration (i.e. $\Gamma_O \cap \Gamma_D = \emptyset$ and $G_{CG} = 0$), I conclude that G_C and G_G formally existing in the INT configuration are of minor influence for the investigated case.

In conjunction with turbulent low-Re wall boundary conditions, the primal solver tends

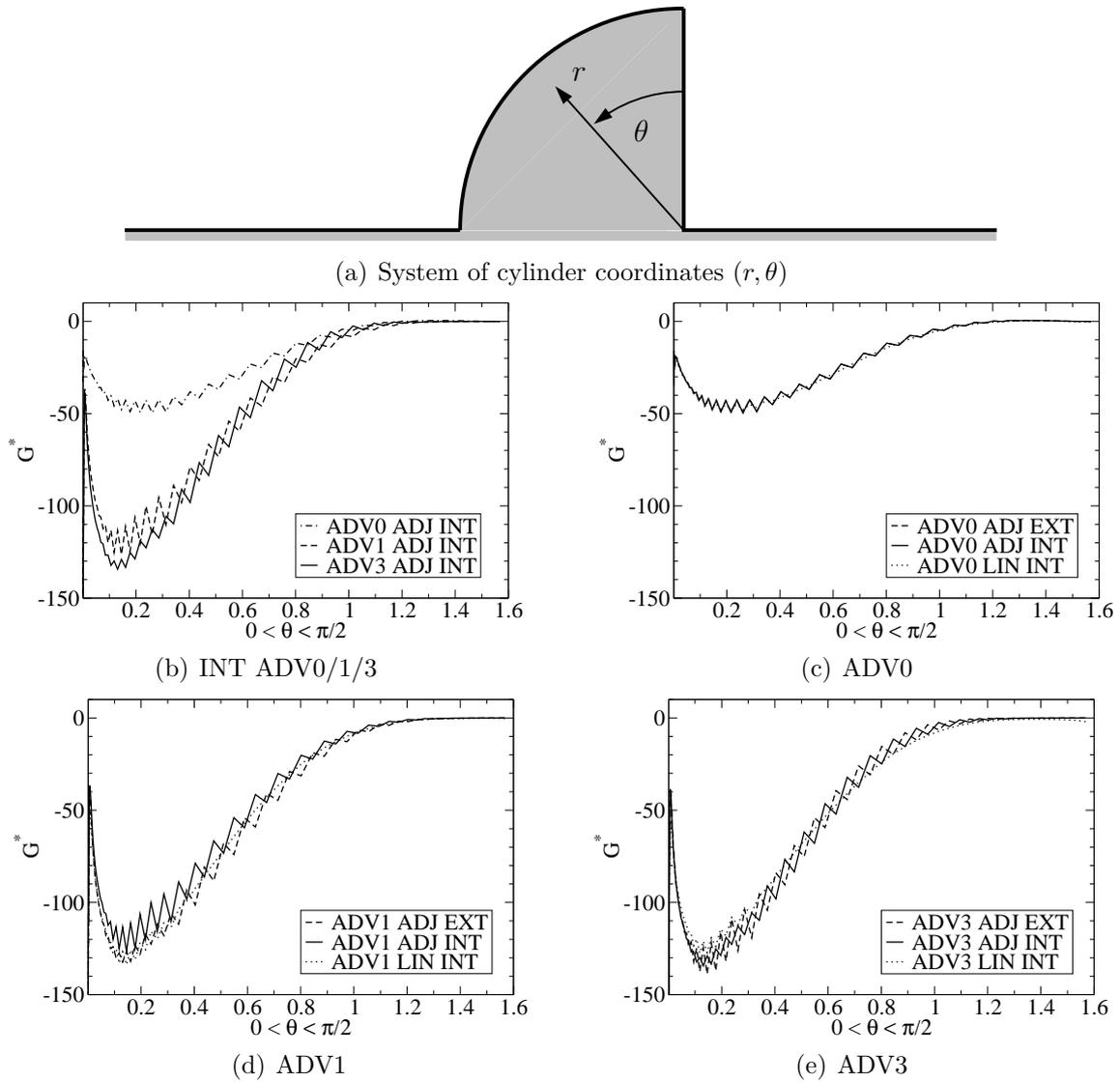


Figure 9.24.: Quarter cylinder in free stream at $Re = 10^6$ with low-Re wall boundary condition; nondimensional derivative G^* of drag force according to adjoint (ADJ) and direct-differentiation (LIN) method using advection schemes ADV0/1/3; interior (INT) vs. exterior (EXT) declaration of objective functional

to produce moderately oscillating distributions of the wall shear rate if the ratio of the grid spacing in the wall-tangential direction and the local curvature radius is large. Potential reasons for the rough distributions of the wall shear rate next to highly stretched boundary-layer cells are: the metric schemes for the computation of the wall distance y_P and the boundary vector $\Delta\underline{\Gamma}$ for polygonal boundary patches that are not necessarily planar; the formulation of the primal pseudo boundary condition for ω_P (7.63) which is $\mathcal{O}(y_P^{-2})$; and the neglected boundary curvature in the formulation of the wall boundary condition. The adjoint wall shear rate is comparably rough, so that the adjoint-based sensitivity derivative, calculated from a differential expression based on the product of primal and adjoint shear rates (8.68/8.69), is quadratically affected by the oscillations. The amplitudes of the oscillations decreased when the curvature resolution was increased. To process the rough sensitivity derivative in an automated optimisation program, a filtering-based preconditioning of the sensitivity derivative suggested by Stück and Rung [152] is described in Section 10.3. Other authors [e.g. 62, 2, 78, 99, 151, 109, 108] evaluate the sensitivity derivative (2.41) from integral, volume-based expressions—cf. Section 3.3—so that potential irregularities in the adjoint solution are averaged over several cells and the huge content of information available from adjoint methods is reduced to a pre-selection of (CAD) parameters.

10. CAD-free Geometry Concept for Shape Optimisation

In a CAD-free optimisation strategy, the geometry has to be evaluated and manipulated on the basis of the computational mesh alone. The basic calculus presented in the following is applicable in the framework of fully-unstructured finite-volume grids. It consists of basic, CAD-free concepts for the evaluation of metric quantities (Section 10.1), a handling of simple metric constraints (Section 10.2), a filter-based preconditioning of surface derivatives (Section 10.3, also see Stück and Rung [152]), and a mesh adaptation approach (Section 10.4).

10.1. Evaluation of Metric Terms

The volume enclosed by a number of surfaces can be calculated from the divergence theorem (e.g. Soto et al. [151])

$$V = \frac{1}{3} \oint x_i d\Gamma_i. \quad (10.1)$$

The formulation may also be used in conjunction with symmetry planes cutting the body of interest. In that case a closed surface representation is not available. With the mirror plane defined by a position vector \underline{x}^s located on the plane, formulation (10.1) can be recast as

$$V^s = \frac{1}{3} \int (x_j - x_j^s) d\Gamma_j. \quad (10.2)$$

Accordingly, the coordinates of the barycentre can be obtained using the first geometric moment

$$x_i|_B = \frac{1}{4V} \oint x_i x_j d\Gamma_j, \quad (10.3)$$

or, in the presence of a symmetry plane

$$x_i^s|_B = \frac{1}{4V^s} \int (x_j - x_j^s) (x_i - x_i^s) d\Gamma_j. \quad (10.4)$$

The descriptions above are also valid for concave volumes.

10.2. Handling of Constraints

A *penalisation of the objective functional* is a very popular technique to account for (metric) constraints. The objective functional can, for instance, be augmented by a volume

constraint

$$C_V = \frac{1}{2}(V - V_0)^2 = 0, \quad (10.5)$$

so that deviations from the desired volume V_0 are penalised. The integral constraint (10.5) can be linearised with respect to a change of the control and considered in the sensitivity analysis.

Alternatively, a *gradient projection* method can be pursued to eliminate the components of the gradient that violate the constraint(s). A gradient projection \tilde{G} , compatible with the volume constraint (10.5), can be obtained by subtracting the average as—for instance—proposed by Huan and Modi [60] and followed by Stück and Rung [152]

$$\tilde{G} = G - \frac{\int_{\Gamma_D} G \, d\Gamma}{\int_{\Gamma_D} d\Gamma}. \quad (10.6)$$

A potential change in the surface area is neglected in the calculation of the average volume per step size, $\delta V/\alpha \approx \int_{\Gamma_D} G \, d\Gamma$, “swept” by the deformation. This approximation is satisfactory as long as the resolution of the boundary curvature is sufficient. In contrast to the penalisation method, the constraint is satisfied during the whole optimisation process. Given the violation is severe, the orthogonal projection of the constraint can be a considerable offset applied of the unconstrained sensitivity derivative.

10.3. Filtering of Derivatives

Small-scale oscillations (noise) are usually undesired in practical shape design for several reasons. Among these are hydrodynamic, structural, manufacturing, operational or economic considerations. In a CAD-free approach the same mesh is used to discretise both flow and shape. This strongly reduces the number of tools and data-interfaces involved in the optimisation. The resolution of the computational mesh is usually chosen in accordance with the RANS requirements. Thus, in some areas the grid resolution can be very high in order to capture the important flow features (e.g. boundary layers, stagnation or separation and re-attachment zones). The resolution requirements of the shape description can be different, mainly driven by the local curvature of the boundary so that the characteristic length scales of the description of geometry and flow can be different. Very fine mesh resolutions are prone to non-smooth derivatives with respect to the shape, dominated by fine-scale structures. Such structures usually slow down the local optimisation or lead to undesired shapes.

This motivates the use of filters that eliminate short length-scales from the distribution of the derivative with respect to the shape. Linear convolution filters with a uniform filter kernel $K(r)$ are applied in this study. The filtered distribution of the derivative is given by the convolution

$$\bar{G}(\xi) = H G(\xi) = \int_{\Gamma_D} K(r) G(\xi - r) \, dr, \quad \forall \xi \in \Gamma_D, \quad (10.7)$$

with r, H and ξ representing the local filter radius, the filtering operator and the curved boundary coordinate, respectively. The extension of the one-dimensional description to two-dimensional surfaces with a local system of orthogonal surface coordinates, $\xi_i, i = 1, 2$, is straight-forward. The filter is assumed to be a bounded operator, to have a normalised filter function

$$\int K(r) dr = 1, \quad \forall \xi \in \Gamma_D \quad (10.8)$$

and to be conservative, so that it does not alter the integral of the derivative with respect to the shape over the design surface Γ_D (global conservation)

$$\int \bar{G}(\xi) d\xi = \int H G(\xi) d\xi = \int G(\xi) d\xi. \quad (10.9)$$

Gaussian Filters

Many formulations for low-pass filters are available in literature, such as box-filters, Gaussian, spectral or Pao filters. Motivated by scale-space theory [86, 88, 15, 29, 87], Gaussian filters are used in this study. These conform with Lindeberg [87], who postulates in the context of image data processing:

“Coarse-scale representations should constitute simplifications of corresponding structures at finer scales—they should not be accidental phenomena created by the smoothing method intended to suppress fine scale structures.”

Among others, Gaussian kernels have the subsequent scale-space properties likewise apt in the context of shape optimisation: linearity, shift invariance, non-enhancement of local extrema, invariance of scale and rotation—cf. for example Lindeberg and ter Haar Romeny [88] and [87]. The Gaussian filter is local in both physical and wave-number space. It is interesting to note that Gaussian kernels are a Green’s function or fundamental solution to the unsteady diffusion-equation

$$\frac{\partial G}{\partial t} = \frac{\gamma}{2} \frac{\partial^2 G}{\partial \xi^2} \quad \text{on } \Gamma_D, \quad (10.10)$$

with $\gamma/2$ being the diffusion coefficient. The corresponding Gaussian kernel is

$$K(t) = (2\pi\gamma t)^{-D/2} \exp\left(-\frac{r^2}{2\gamma t}\right), \quad \forall \xi \in \Gamma_D, \quad (10.11)$$

wherein $\sigma^2 = \gamma t$ [m²] is the variance of the filter and D denotes the dimensionality of the boundary that is subject to the filter ($D = 1$ for lines; $D = 2$ for surfaces). Details present in the “raw” distribution of the derivative with respect to the shape, which are significantly smaller than the standard deviation or length scale of the filter kernel, $\sigma = \sqrt{\gamma t}$, are eliminated by the filter. For $t = 0$ the original distribution is retained.

An implicit first-order approximation to Eqn. (10.10)

$$G(t) - \frac{\gamma t}{2} \frac{\partial^2 G}{\partial \xi^2}(t) = G(0) + \mathcal{O}(t) \quad \text{over } \Gamma_D \quad (10.12)$$

allows for a direct comparison against the implicit smoothing technique based on a ‘‘Sobolev-Gradient’’, which is well-established in adjoint shape optimisation [65, 69, 138]. There, a second-order PDE is solved over the design surface in order to obtain a smoothed derivative, or gradient \bar{G} , from the raw derivatives G , viz.

$$\bar{G} - \frac{\partial}{\partial \xi} \left(\varepsilon \frac{\partial \bar{G}}{\partial \xi} \right) = G \quad \text{over } \Gamma_D. \quad (10.13)$$

A step in the negative direction of \bar{G} with the stride α guarantees a negative (desired) variation of the objective functional for arbitrary choices of a uniform value ε [65]:

$$\delta J = - \int_{\Gamma_D} [\alpha \bar{G} G] \, d\Gamma = -\alpha \int_{\Gamma_D} \left[\bar{G}^2 + \varepsilon \left(\frac{\partial \bar{G}}{\partial \xi} \right)^2 \right] \, d\Gamma. \quad (10.14)$$

The equivalence above is obtained using (10.13) and integration by parts in conjunction with zero values of \bar{G} along the bounding lines of Γ_D . It is interesting to note that Eqns. (10.12) and (10.13) are first-order equivalent. A direct comparison reveals

$$\varepsilon \approx \frac{\gamma t}{2}, \quad (10.15)$$

so that the smoothing intensity ε involved in Eqn. (10.13) can be interpreted as half the variance, $\sigma^2 = \gamma t$, of the Gaussian filter kernel. The optimal choice of ε is case dependent. Gherman and Schulz [36] and Schmidt et al. [138] suggested to identify an appropriate (optimal) value for ε by analysis of the corresponding Hessian, Kim et al. [79] used automatic procedures.

As outlined by Schmidt et al. [138] and Eppler et al. [26], the preconditioned steepest-descent can be considered as an approximate Newton method. The close relation to the Sobolev-smoothing puts the suggested filtering technique on a firm rational basis. In contrast to the implicit smoothing technique, Eqn. (10.13), the filtering operation is fully explicit. It can easily be applied in the context of unstructured grids, where the grid connectivity is often not available for the surface patches.

This study is confined to uniform Gaussian filter kernels, which are globally conservative, cf. Eqn. (10.9). In practice, a part of the kernel is cut off close to the bounding lines of Γ_D , since it has an unbounded support. However, the filter operation acts locally as for $r/\sqrt{(\gamma t)} = 2$, approximately 95 per cent of the kernel integral is included. The kernel is renormalised numerically to enforce Eqn. (10.8). The filter operation becomes particularly simple when the influence of the local curvature of the design surface is ignored within the kernel width. The approach is defensible due to the dense support of the kernel as long as $\sqrt{(\gamma t)}$ is kept small compared to the local radius of curvature. The latter is usually satisfied and an undesired filtering across edges can be suppressed by screening the changes of the face normals.

Numerical Implementation

The suggested filtering of the derivative is performed in a fully explicit manner. In many cases, the domain decomposition, which is optimised to compute the flow in parallel based

on the MPI protocol, turns out to be suboptimal for surface operations. To avoid an extra decomposition, the boundary patches for every separate surface area to be filtered are gathered on a single processor to avoid MPI communication overheads. The resulting memory consumption of approximately 10^5 or less variables per processor is considered tolerable. Boundary conditions are not considered explicitly, instead the part of the filter that exceeds the considered surface is simply cut off. The filtering can be considered as a weighted local average over the kernel radius. Particularly on unstructured grids, the implementation of the filtering approach is less involved than the implementation of the smoothing algorithm based on ‘‘Sobolev-gradients’’: Often the data-structures existing to solve the governing PDE in the domain do not have a counterpart on the domain boundaries. Beyond the necessary connectivity table for the boundary faces, a management of boundary conditions is needed to declare the boundary conditions for the surface PDE over the confining surface lines.

10.4. Mesh Adaptation

In the face-centred FV approach the gradient is available in the centres of the boundary faces. The boundary-normal perturbation of the optimisation cycle i , $\delta n|^i$, was calculated from the filtered gradient \overline{G}^i and step size α , viz.

$$\delta n|^i = -\alpha \overline{G}^i \quad \text{on } \Gamma_D. \quad (10.16)$$

The boundary displacements $\underline{d}|^i = [\underline{n} \delta n]^i$ were passed into the interior domain by solving a Laplace field equation

$$\underline{\nabla} \cdot (\gamma_d \underline{\nabla} \underline{d}|^i) = \underline{0} \quad \text{in } \Omega, \quad (10.17)$$

with the boundary displacements $\underline{d}|^i$ imposed as boundary conditions over the design surface(s) Γ_D . Zero displacements were declared on the outer boundaries. The deformation was extrapolated to symmetry boundaries attached to the design surfaces. Undesirable out-of-plane displacements of the symmetry boundaries, $\underline{P} \cdot \delta \underline{n}$, were subtracted explicitly. The implementation allows to use inhomogeneous distributions of γ_d to control or optimise the field deformation, e.g. Wick [165]. In the current study γ_d was set to unity. The displacement vector of the mesh nodes was subsequently obtained by interpolation from the centres of the surrounding CVs or boundary faces. After a succession of n optimisation cycles, the final displacement of a mesh node is the sum over its trajectory

$$\underline{x}|^n - \underline{x}|^0 = \sum_{i=1}^n \underline{d}|^i. \quad (10.18)$$

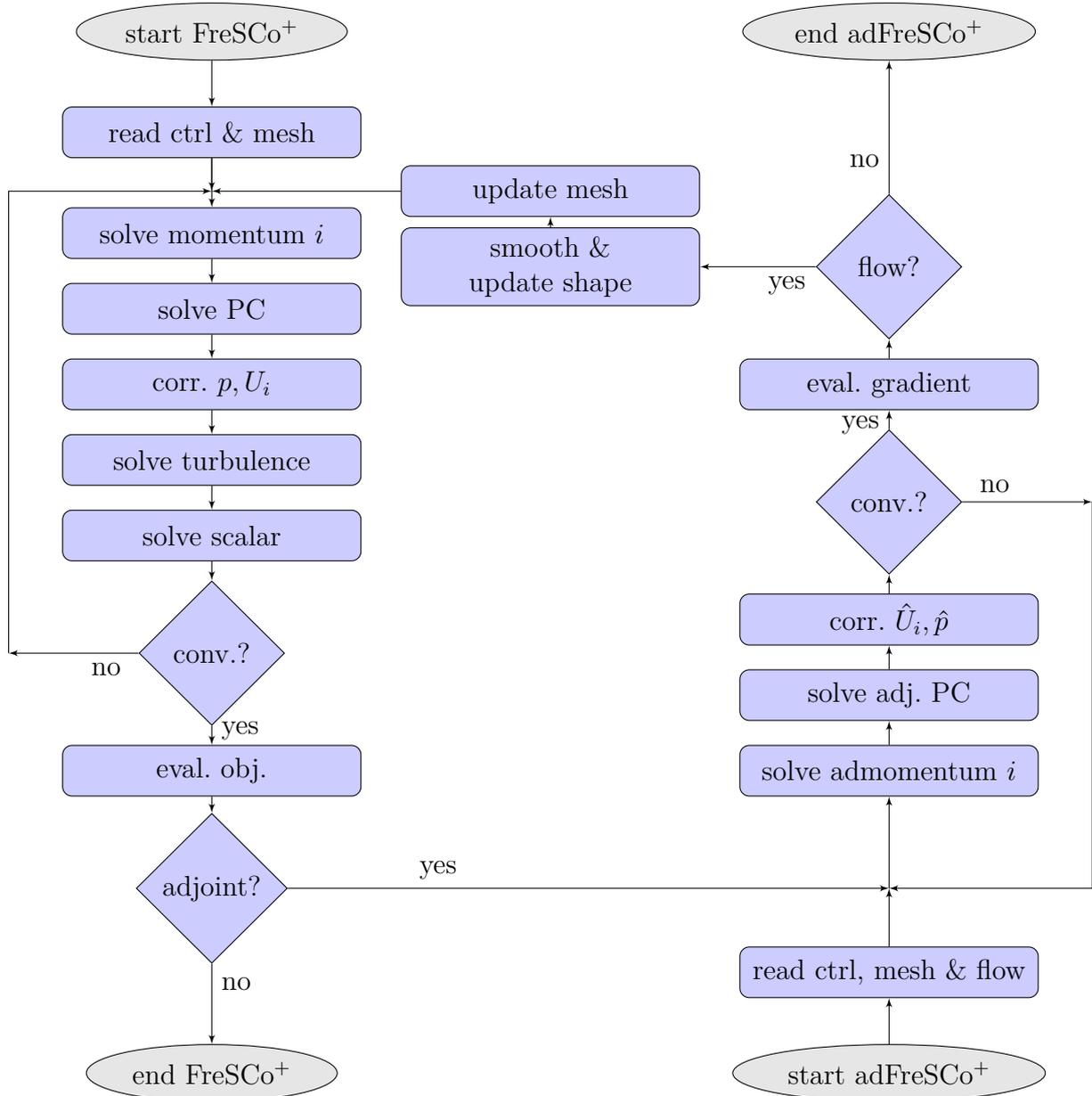


Figure 10.1.: Iterative optimisation scheme coupling flow solver and adjoint solver

11. Applications

The adjoint Navier–Stokes method presented in this work can be used to support both manual and automatic shape optimisations. In Section 11.1, the adjoint method is introduced to wake design and tailored to specific wake objective functionals that quantify the quality of a ship’s velocity wake field. An adjoint-based sensitivity analysis was carried out to guide a manual aftship redesign (cf. Stück et al. [155, 156]). Moreover, automatic shape optimisation runs were performed for a 2D T-junction in Section 11.2 to reduce the power loss. Subsequently, a 3D double-bent pipe was optimised in Section 11.3.

11.1. Sensitivity Analysis for Manual Wake Optimisation

The propeller operates in the wake field of the ship. To improve the propeller inflow conditions, the nominal wake field is investigated in the absence of the propeller. The flow is evaluated in the propeller disk Ω_O representing the volume swept by the propeller blades. According to Figure 11.1, a system of physical cylinder coordinates (r, θ, a) is located in the centre of the propeller hub

$$x_i e_i = r \tilde{e}^r + \theta \tilde{e}^\theta + a \tilde{e}^a. \quad (11.1)$$

\tilde{e}^a is aligned with the propeller axis pointing from fore to aft. The upright position is described by $\theta = 0$. The corresponding velocity coordinates are U^r , U^θ and U^a . The propeller disk extends over the volume $R_I < r < R_O$, $0 < \theta < 2\pi$ and $a_{\text{fore}} < a < a_{\text{aft}}$, where the inner radius R_I is the propeller hub radius and the outer radius R_O is slightly beyond the propeller tip radius. In the axial direction the disk reaches from a_{fore} to a_{aft} .

It is common practice to evaluate the quality of the wake field in terms of the circumferential variations of the (axial) velocity wake over the propeller disk. For a single-screw ship, the 12 o’clock position is strongly influenced by the presence of the ship hull, whereas the lower region is usually less affected. Depending on the hull shape, bilge vortices may disturb the flow field in the propeller disk. Local deviations in the axial velocity U^a and the circumferential velocity component U^θ lead to a local variation of the propeller blade load, due to changes in the blade angle of attack and the magnitude of the inflow velocity. The propeller blades pass wake inhomogeneities at the blade frequency which is the number of blades times the number of revolutions. The pressure fields of the passing blades excite the ship structure at the blade frequency and its multiples causing vibrations and noise. This effect is accompanied by cavitation, particularly in the area of highest blade loads and low hydrostatic pressure above the propeller. Bubble growth and collapse lead to pressure pulses at high frequencies. Beyond noise and vibration, cavitation damages may occur on the propeller and the rudder. These devices are particularly sensitive in the context of ship safety and security.

The ship's nominal wake field has to be taken into account during the design of the propeller blade sections. The radial variation of the velocity wake can be compensated by choosing an appropriate pitch distribution over the radius. The circumferential variation cannot be met by the propeller design. An appropriate hull shape that leads to more homogeneous inflow conditions over the perimeter facilitates the propeller design and improves the propeller-hull interaction.

Wake Objective Functionals

The wake design problem was turned into an optimisation problem: Find the hull form parameters β_i that lead to an increased homogeneity of the wake field. The design is evaluated in terms of integral hydrodynamic wake objective functionals J , quantifying the quality of the wake field. The objective functional is obtained by integration over the propeller disk volume or objective volume Ω_O :

$$J = \int_{\Omega_O} j \, d\Omega. \quad (11.2)$$

Among several wake objective functionals used in maritime industry is the formulation proposed by the SVA Potsdam. The *SVA criterion* is based on the deviation of the axial velocity component U^a from its mean value $\bar{U}^a(r)$ that is obtained by averaging over the circumference via

$$\bar{U}^a(r) = \frac{1}{2\pi} \int_0^{2\pi} U^a(r) \, d\theta \quad \text{for } R_I < r < R_O. \quad (11.3)$$

With U_S denoting the ship velocity, the integral variation for each radius

$$\mathcal{D}_I(r) = \frac{1}{2\pi U_S} \int_0^{2\pi} |U^a - \bar{U}^a| \, d\theta, \quad R_I < r < R_O \quad (11.4)$$

is weighted by the relative bandwidth experienced on that radius

$$\mathcal{D}_{II}(r) = \frac{1}{U_S} [\max(U^a) - \min(U^a)], \quad R_I < r < R_O. \quad (11.5)$$

The optimisation goal is to find a hull shape that maximises the wake objective functional

$$J_A = 1 - \frac{1}{R_O - R_I} \int_{R_I}^{R_O} \sqrt{2\mathcal{D}_I(r) \mathcal{D}_{II}(r)} \, dr. \quad (11.6)$$

In combination with unstructured grids, in which the nodes are not aligned with the propeller coordinate system, the wake objective functional requires to be re-stated, e.g.

$$J_A = 1 - C_A \int_{\Omega_O} \frac{d\Omega}{r} \sqrt{U_S \mathcal{D}_{II}(r) |U^a - \bar{U}^a(r)|}, \quad (11.7)$$

with $\mathcal{D}_{II}(r)$ according to Eqn. (11.5) and

$$C_A = \left[U_S \int_{\Omega_O} \frac{d\Omega}{r} \right]^{-1}. \quad (11.8)$$

The integration is carried out over the computational cells whose centres are located inside the propeller disk volume Ω_O .

A simplified wake objective functional J_B , measures the integral quadratic deviation in U^a from its radial average, i.e. maximise

$$J_B = 1 - C_B \int_{\Omega_O} \frac{d\Omega}{2r} [U^a - \bar{U}^a(r)]^2 \quad \text{with} \quad C_B = \left[U_S^2 \int_{\Omega_O} \frac{d\Omega}{r} \right]^{-1}. \quad (11.9)$$

For a constant ship velocity U_S and \mathcal{D}_{II} frozen, the variation of the SVA criterion (11.7) with respect to the flow reads

$$\delta J_A = \int_{\Omega_O} \delta U_i \frac{\partial j_A}{\partial U_i} d\Omega \quad \text{with} \quad \frac{\partial j_A}{\partial U_i} = -\tilde{e}_i^a C_A \frac{U^a - \bar{U}^a(r)}{2r} \sqrt{\frac{U_S \mathcal{D}_{II}(r)}{\max(|U^a - \bar{U}^a(r)|^3, \epsilon)}}. \quad (11.10)$$

A small number ϵ is used in Eqn. (11.10) to suppress the singularity of $\partial j_A / \partial U_i$ where $U^a = \bar{U}^a(r)$.

The variation of the quadratic deviation norm (11.9) is given by

$$\delta J_B = \int_{\Omega_O} \delta U_i \frac{\partial j_B}{\partial U_i} d\Omega \quad \text{with} \quad \frac{\partial j_B}{\partial U_i} = -\tilde{e}_i^a C_B \frac{U^a - \bar{U}^a(r)}{r}. \quad (11.11)$$

Eqns. (11.10) and (11.11) can be considered as different weights applied to the deviation $[U^a - \bar{U}^a(r)]$. Compared to J_B , J_A has two formal deficiencies in the context of the differential analysis: (i) $\partial j_A / \partial U_i$ has a singularity in $U^a = \bar{U}^a(r)$, which is numerically attenuated by a small number ϵ in Eqn. (11.10); (ii) it is difficult to differentiate the max-statement in \mathcal{D}_{II} with respect to the flow over Ω_O , thus \mathcal{D}_{II} is frozen in the adjoint analysis.

Manual Hull Redesign

A generic PanMax container vessel was investigated at a model Reynolds number of

$$Re = \frac{U_0 L}{\nu} = \frac{1 \text{ m/s} \times 200 \text{ m}}{4/3 \times 10^{-5} \text{ m}^2/\text{s}} = 1.5 \cdot 10^7. \quad (11.12)$$

The main dimensions of the CAD-parametrised ship hull are listed in Table 11.1. The flow domain extended one ship length ahead, to the side and below the ship hull. Behind the ship two ship lengths were resolved numerically. For symmetry reasons only half of the ship hull was modelled. Another symmetry boundary condition was declared at the still-water surface. The velocity was prescribed at the inlet, at the outlet the pressure

was set. Over the ship hull the boundary layer was resolved numerically ($y^+ < 1$). The computational grid, cf. Figure 11.2, consisted of approximately 1.1 million cells. It is locally refined in the vicinity of the hull and in the wake region. The diameter of the propeller disk, wherein the wake objective functional was evaluated, was set to 105% of the propeller diameter. The inner diameter and the axial extension of the disk were 17.5% and 16% of the propeller diameter, respectively. The Wilcox k - ω turbulence model was used.

Figure 11.3 depicts the shear stress acting on the aftship hull in conjunction with streamlines tracing the major wake field disturbances upstream. The axial velocity wake in the propeller disk was normalised with the ship velocity U_S . Due to the formal deficits of J_A in the context of differentiation, the adjoint sensitivity analysis was based on formulation J_B . Nonetheless, J_A was evaluated during the redesign in addition to J_B to quantify the wake field improvements. The source terms in the adjoint momentum equations, $-\partial j_B / \partial U_i$, drive the adjoint solution inducing an adjoint velocity wake shown in Figure 11.4. As expected, the strongest response of the adjoint velocity can be observed in the area of the highest velocity deficit of the primal wake. Having solved the adjoint equations, the distribution of the sensitivity derivative over the hull surface, Figure 11.5, is obtained from Eqn. (8.68): A dark (light) shading indicates that the wake objective functional is improved by a local increase (decrease) of the displacement. A contour line represents the zero level in the gradient distributions (left). Within a manual redesign study, the sensitivity analysis was carried out for three design stages 0 (initial), 1 and 2. Flow field, wake objective functional and sensitivity derivative were updated for each design stage. The cycle was carried out twice. The design evolution is visualised in Figure 11.6 by waterlines, buttock-lines and frame sections. The design steps 0, 1 and 2 are coloured in black, dark and light grey. A smoother heel-shape of the stern gondola (cf. buttock line B0) in combination with a more circular frame-character of the stern gondola (buttock lines B1 to B3, frame sections F2 to F5) reduced the disturbances in the 6 and 12 o'clock positions of the propeller disk. The corresponding velocity wake fields are depicted in Figure 11.5 (right). The white circles outline the circular rings which were used to evaluate the wake

Table 11.1.: Main dimensions and characteristics of the generic PanMax container vessel

length between perpendiculars L	200.0 m
length waterline	204.0 m
beam	32.2 m
draft	11.5 m
propeller hub vert. pos. above basis	4.5 m
propeller diameter	8.0 m
block coefficient	0.68
midship section coefficient	0.98
Reynolds number at model scale	1.5×10^7

objective functional over the perimeter. The circumferential inhomogeneities in the wake velocity field were alleviated. Accordingly, the hook-shaped “signature” dominating the 6 o’clock position in step 0 decreased in steps 1 and 2. It is interesting to note that the improved shapes result in smoother distributions of the derivative with respect to the shape. As illustrated in Figure 11.7, the value of the wake objective functional increased from $J_A = 0.716$ to 0.787 and $J_B = 0.970$ to 0.984 after two iterations. Trim and sinkage were frozen in the analysis, resulting in a relative increase of the displacement of 0.05%. The longitudinal centre of buoyancy moved astern by 0.025% L .

Concluding Remarks

Wake objective functionals used in practical hull design were introduced into the adjoint calculus. A sensitivity analysis was carried out to guide the redesign of a generic PanMax container vessel. The method provides a continuous sensitivity distribution over the hull surface and yields considerable insight into the design problem from the objective point of view. The wake objective functional J_A increased by approximately 10 per cent. Several improvements are required to prepare the adjoint technique for practical use, including

- improved wake objective functionals, accounting for variations of the propeller blade load due to changes in the local blade angle of attack and the incoming flow velocity. It must be kept in mind that the practical success of the adjoint-based sensitivity analysis all depends on the validity of the underlying objective functional.
- tailored data interfaces to several CAD frameworks;
- a mathematical connection of the continuous sensitivity derivative to the shape parametrisation according to the chain-rule of differentiation;
- enhanced robustness of the adjoint numerics for simulations at full scale; and
- to account for the influence of propulsion within the aftship optimisation.

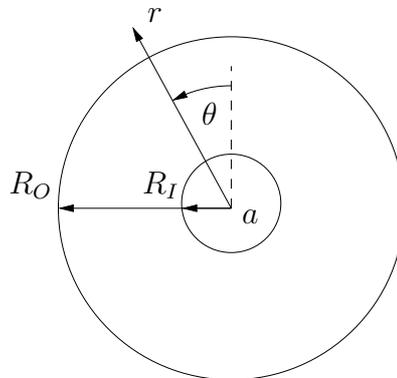


Figure 11.1.: System of physical cylinder coordinates used for the propeller disk (back view); r , θ , and a denote the radial, circumferential and axial coordinates, respectively. The inner and outer radii of the propeller disk are referred to as R_I and R_O .

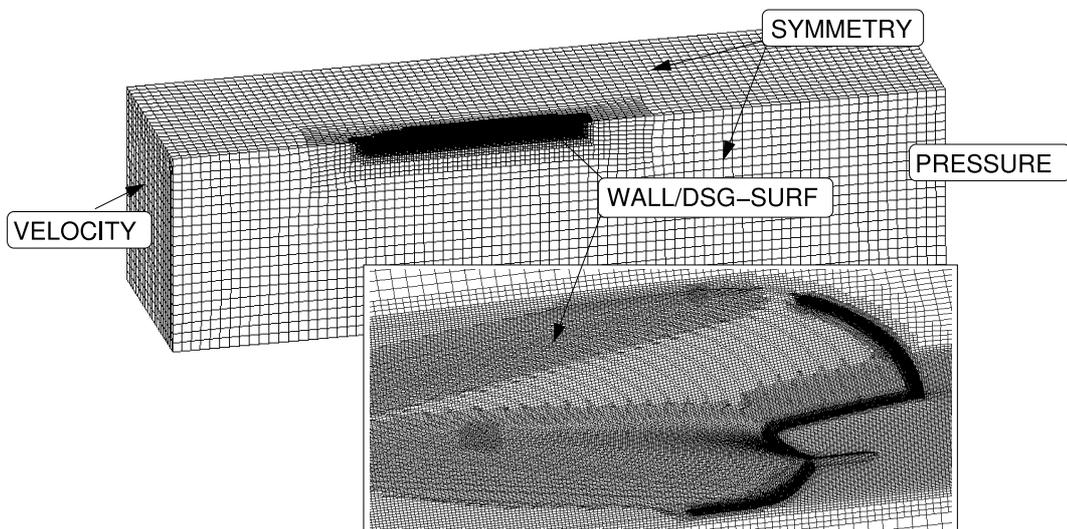


Figure 11.2.: Computational grid with boundary conditions

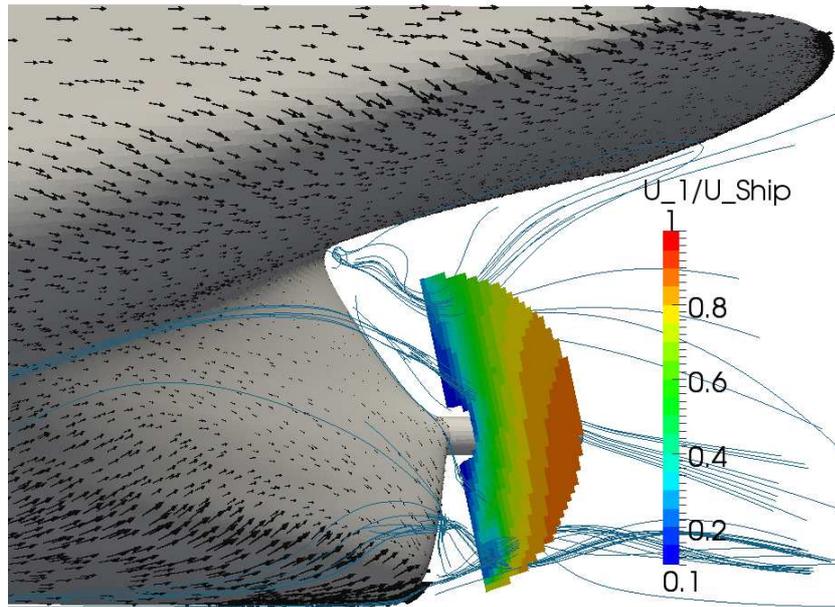


Figure 11.3.: Inhomogeneities in the propeller disk are traced upstream by velocity streamlines for the reference geometry (step 0). The wall shear stress is visualised through vectors on the wetted hull surface.

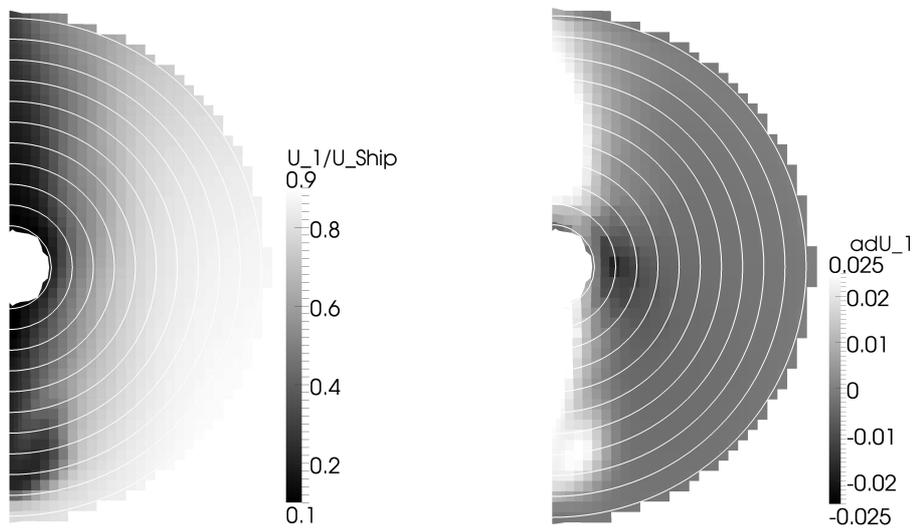


Figure 11.4.: Normalised axial velocity wake (left) and corresponding axial component of the adjoint velocity wake, \hat{U}_1 [1/N] (right) of J_B for the reference geometry (step 0)

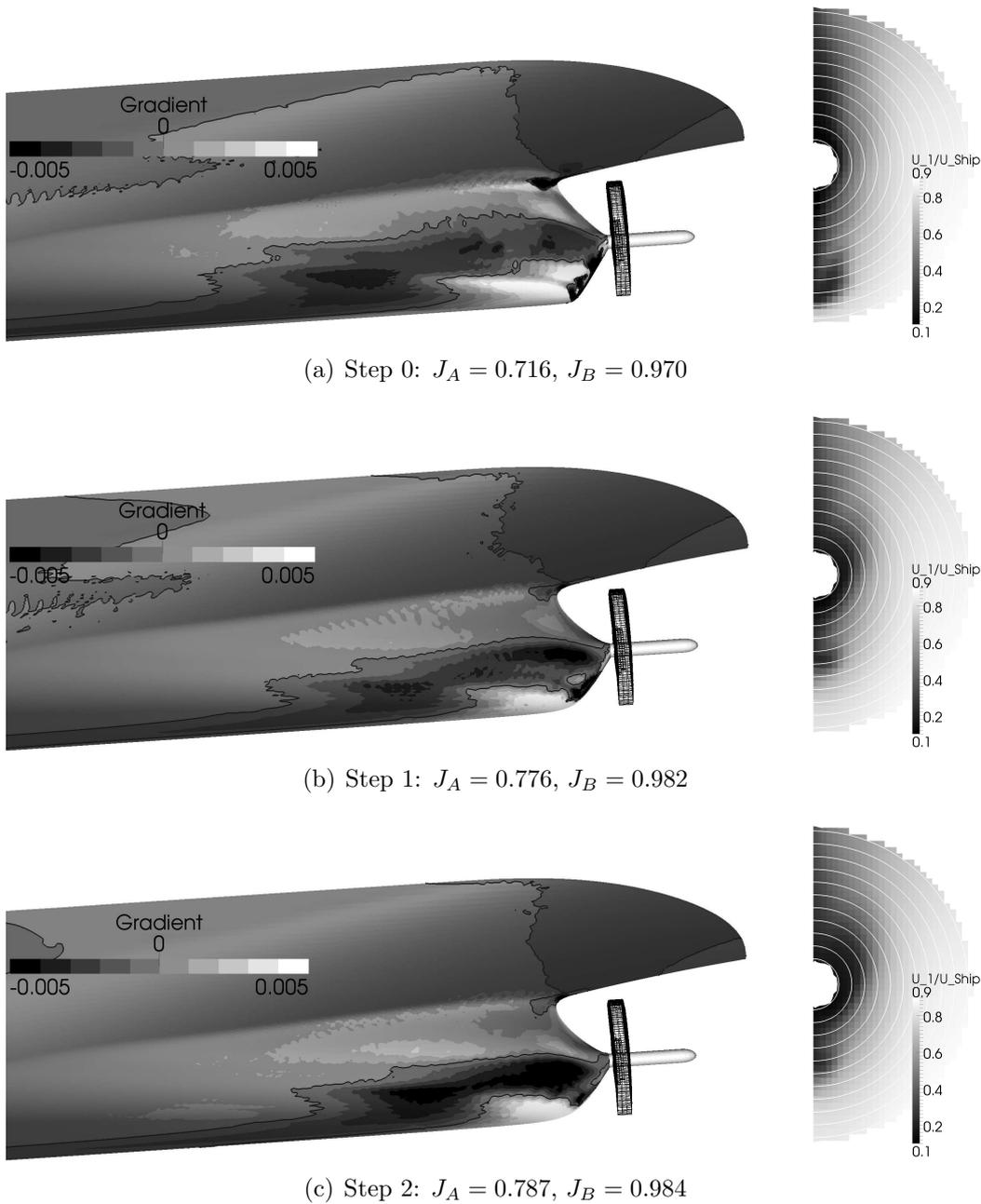


Figure 11.5.: Gradient distribution G [$1/m^3$] of J_B (left) and normalised axial velocity wake (right) for design steps 0, 1 and 2. A dark (light) shading indicates an increase of the wake objective functional by a local increase (decrease) of the hull displacement.

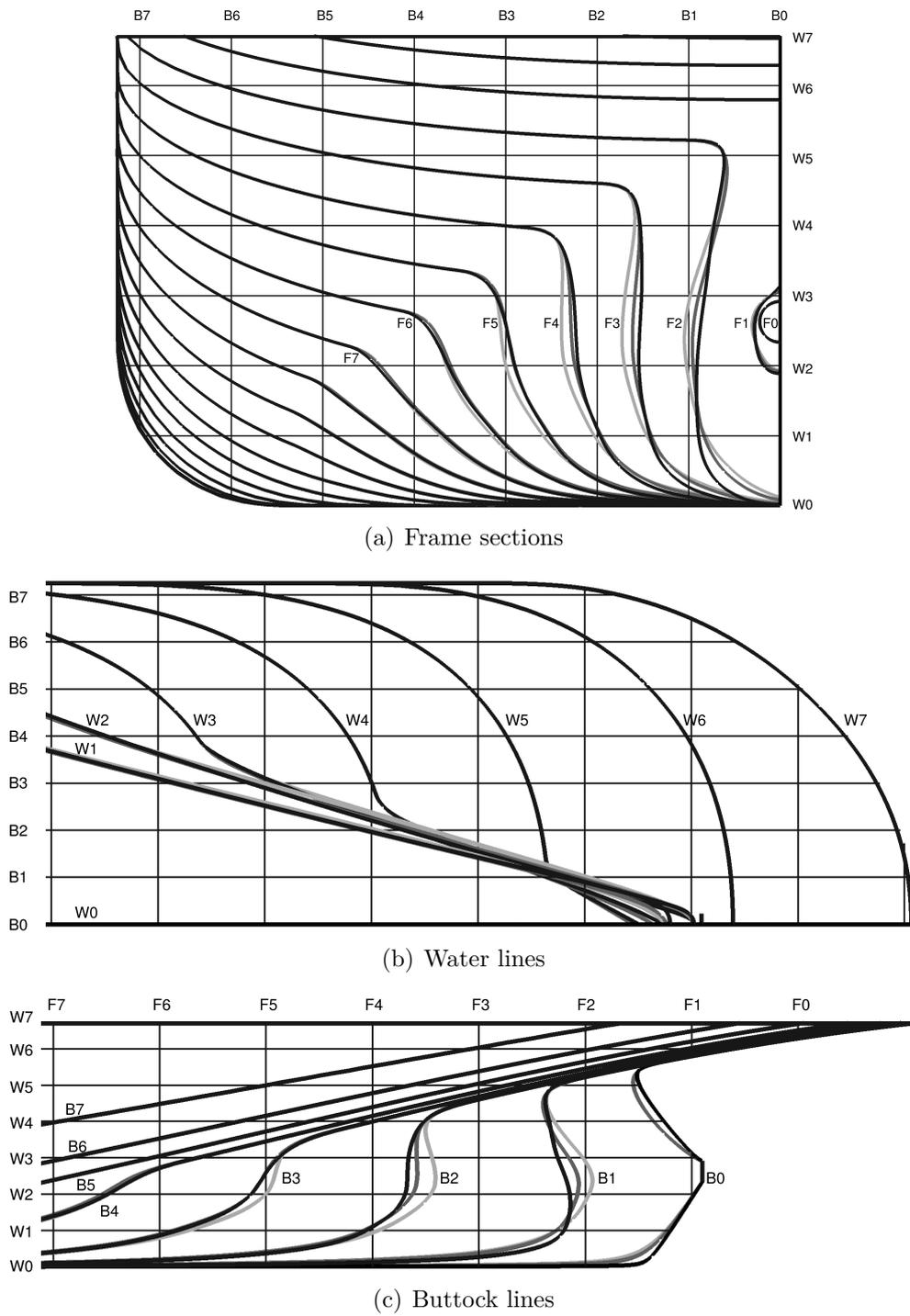
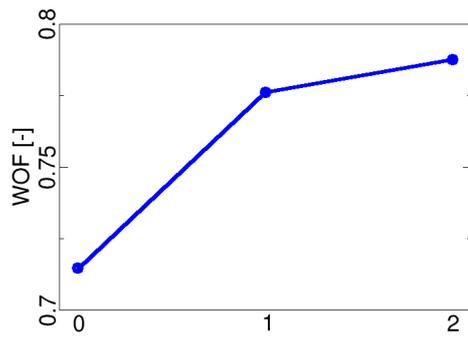
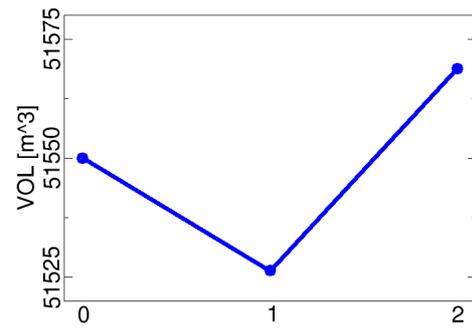


Figure 11.6.: Design steps 0 (black), 1 (dark grey) and 2 (light grey)

(a) Wake objective functional J_A (WOF)

(b) Hull displacement (VOL)

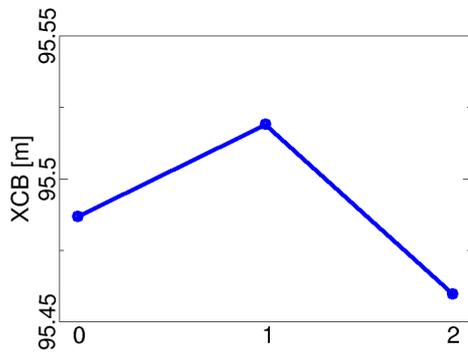
(c) Longitudinal centre of buoyancy (XCB),
positive forward

Figure 11.7.: Design evolution over design steps 0, 1 and 2

11.2. Optimisation of a 2D T-Junction

A 2D T-junction was investigated at a bulk Reynolds number $Re = 50,000$ based on the mean inlet velocity U_{in} and the width D of the inflow section. The shape was optimised to reduce the dissipated power between in- and outlet (Γ_O):

$$\text{minimise } J = - \int_{\Gamma_O} \left(\frac{\rho}{2} U_j^2 + p \right) U_i d\Gamma_i. \quad (11.13)$$

Figure 11.8(a) depicts the initial configuration with the main dimensions, $L_1 = L_2 = 3D$. A comparable setup was optimised by Schmidt [137, Section 6.2.4.] for laminar flow at a Reynolds number of $Re = 100$. Since the design surface Γ_D was confined to no-slip wall boundaries, $\Gamma_O \cap \Gamma_D = \emptyset$, the sensitivity derivative consists of local contributions G_L alone. The low-Re mesh consisted of 92×60 cells in streamline and perpendicular direction, respectively. Half of the system was modelled numerically assuming a symmetry boundary condition on the centre-line. A homogeneous velocity \underline{U}_{in} was imposed on the inlet with the turbulence variables estimated from Eqn. (9.26). A zero pressure value was defined at the outflow.

Low-Re Wall Boundary Condition

A low-Re wall boundary condition was declared on the channel walls. The nondimensional wall distance y^+ was of the order of unity. The turbulence field calculated according to the Wilcox $k-\omega$ turbulence model was frozen in the adjoint analysis. Figure 11.8(b) shows streamlines of the primal velocity field. A recirculation zone in the upper right corner is responsible for a large portion of the power loss. The adjoint velocity field obtained for the objective functional (11.13) using the LU(D)DS convection and the ADV3 advection scheme is shown in Figure 11.8(c). The corresponding sensitivity derivative clearly responds to the recirculation zone as can be observed in Figure 11.8(d). The gradient \overline{G} obtained by filtering the sensitivity derivative G was imposed on the geometry with a step size α . Next to inlet and outlet boundaries, which are not part of the design surface, the gradient \overline{G} was faded to zero by multiplication with a linear ramp function in the intervals $-2 < x_1/D < -1$ and $-2 < x_2/D < -1$.

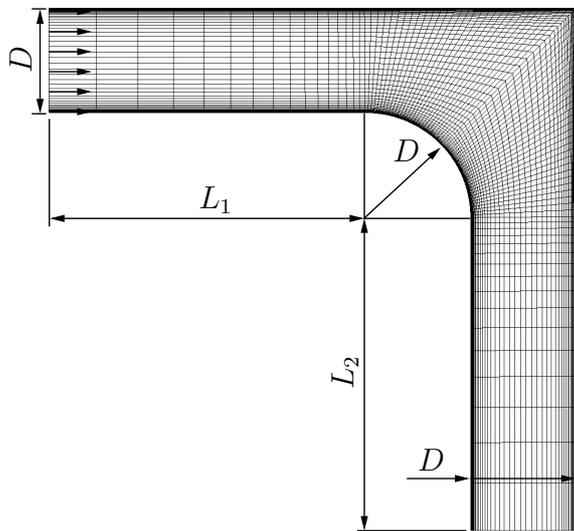
Figure 11.9 illustrates the evolution of the nondimensional objective functional

$$J^* = \frac{J}{\rho U_{in}^3 D}, \quad (11.14)$$

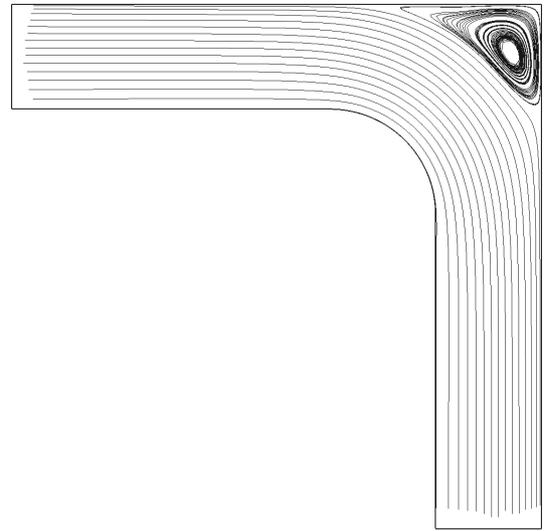
depending on the nondimensional values of the step size and the filter width,

$$\alpha^* = \alpha \frac{\rho U_{in}^3}{D^2} \quad \text{and} \quad \sigma^* = \sigma/D. \quad (11.15)$$

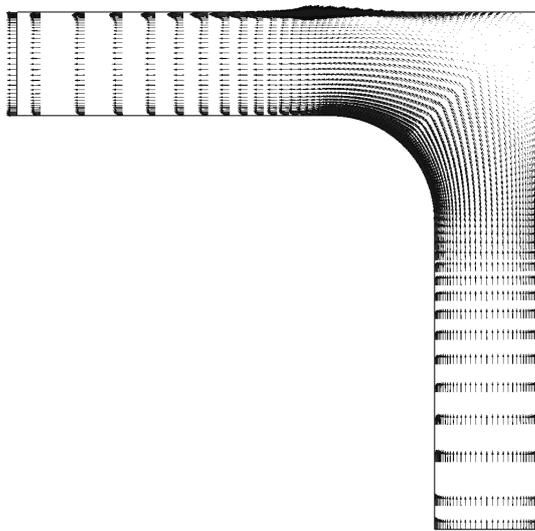
Obviously, the optimisation stopped before the objective functional converged. This was due to the mesh deformation algorithm (10.17) which, in conjunction with RANS grids of very high aspect ratios, could not cope with large deformations. For the small step size



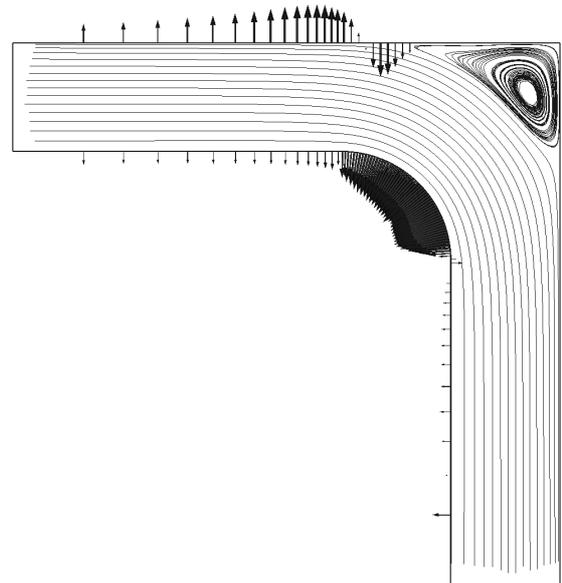
(a) Low-Re mesh with main dimensions



(b) Velocity streamlines



(c) Adjoint velocity field



(d) Sensitivity derivative with velocity streamlines

Figure 11.8.: Initial configuration of T-junction configuration at $Re = 50,000$ with low-Re wall boundary conditions

$\alpha^* = 0.0025$, the optimisation was slightly retarded by increasing the filter width as can be observed in Figure 11.9(a). This behaviour was expected since the filtering operation reduces the maximum amplitudes of the sensitivity derivatives. For the medium step size $\alpha^* = 0.005$ shown in Figure 11.9(b), the small filter width $\sigma^* = 0.025$ seems to be too narrow so that it performs slightly worse than $\sigma^* = 0.05$. For the largest step size $\alpha^* = 0.01$, Figure 11.9(c), the small filter width $\sigma^* = 0.025$ is too narrow so that the optimisation got stuck after two cycles. According to Figure 11.9(d), increasing the step size α^* inversely proportionally reduces the number of iterations required to achieve the same reduction in J^* . It is interpreted as an indication that the step size could still be increased without a notable influence on the optimisation path. The objective functional was reduced by approximately 15 per cent. The optimisation stopped before the objective functional converged. One must bear in mind that due to the mesh deformation also the approximation error has probably changed during the optimisation process.

The shape evolutions for $(\alpha^* = 0.0025, \sigma^* = 0.025)$ and $(\alpha^* = 0.01, \sigma^* = 0.1)$ are shown in Figures 11.10(a) and 11.10(b), respectively. The full outline is complemented by a zoom in on the upper wall and the inner radius. A deflection spoiler developed on the upper wall which successively moved in the x_1 direction. For $(\alpha^* = 0.0025, \sigma^* = 0.025)$, the spoiler contour is sharper than for the wider filter kernel $(\alpha^* = 0.01, \sigma^* = 0.1)$, which led to a smoother shape.

The effect of different filter widths σ^* at a constant step size $\alpha^* = 0.0025$ is shown in Figure 11.11(a). For the wider filter kernels the evolution of the upper wall contour slightly lags behind the narrow filter curves as observed in the history of the objective functional, Figure 11.9(a). A rounder spoiler is obtained by the wider filter kernels. In Figure 11.11(b) the filter width σ^* is increased with the step size α^* showing the same trends. The distribution of the derivative on the inner bend is smooth enough so that the filter width is of minor influence there.

High-Re Wall Boundary Condition

The same geometry was considered in conjunction with a high-Re boundary condition imposed on the no-slip walls. The computational mesh consisted of 110×30 cells in streamline and perpendicular direction, cf. Figure 11.12(a). The numerical settings were inherited from the low-Re simulation. Figure 11.12(b) shows velocity streamlines together with the sensitivity derivative of the power loss criterion (11.13). Whereas the low-Re case shows a sign change in the sensitivity map at the end of the inner bend owing to a small recirculation zone, Figure 11.8(d), the high-Re sensitivities are directed outwards over the entire inner bend. The response to the recirculation area on the upper wall is less distinct than in the low-Re case.

Several optimisation runs were conducted varying step size α^* and filter width σ^* . The optimisation runs were stopped when the objective functional J^* could not be reduced any more. For $(\alpha^* = 0.05, 0.0 < \sigma^* < 0.1)$, cf. Figure 11.13(a), the choice of σ^* has little influence on the evolution of the power loss between inlet and outlet. This is attributed to the smooth distributions of the sensitivity derivatives obtained from the high-Re approach. According to Figure 11.13(b), the number of optimisation cycles required to converge the

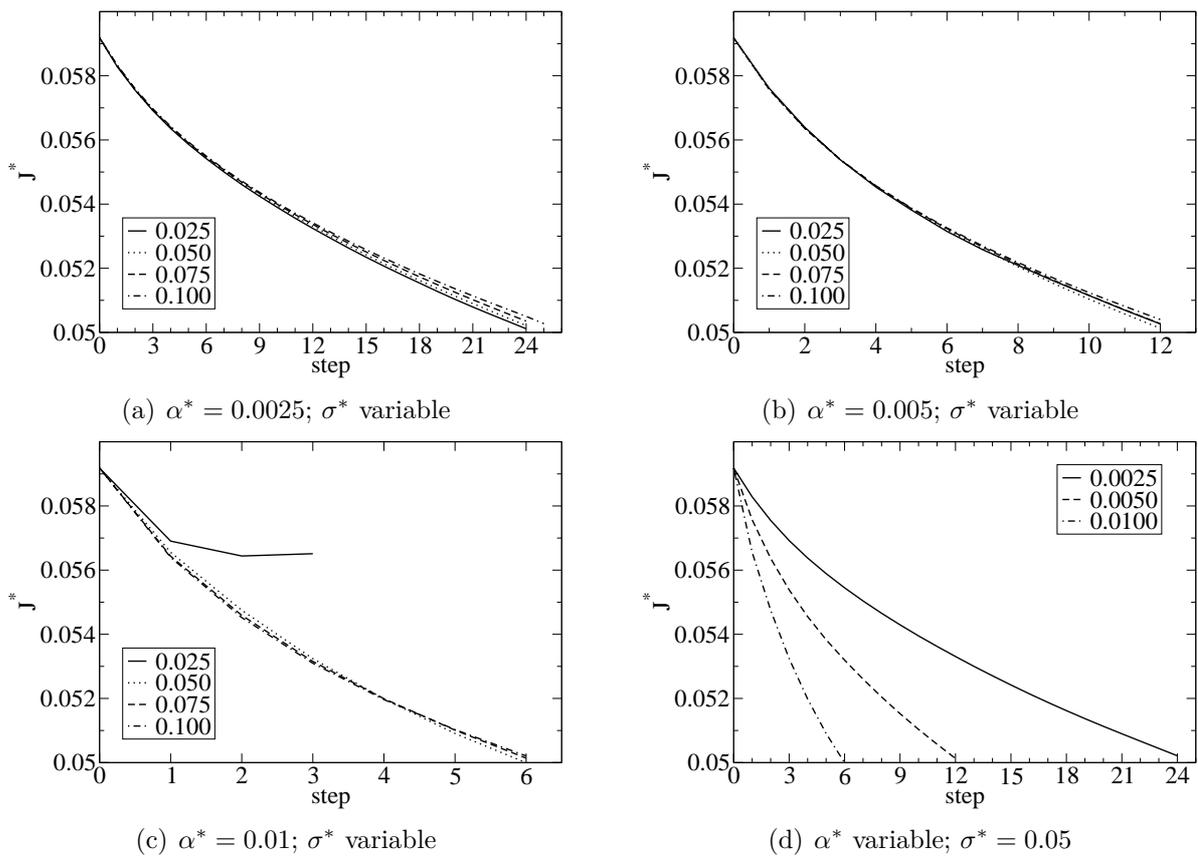


Figure 11.9.: T-junction at $Re = 50,000$ with low-Re wall boundary condition; evolution of power loss J^*

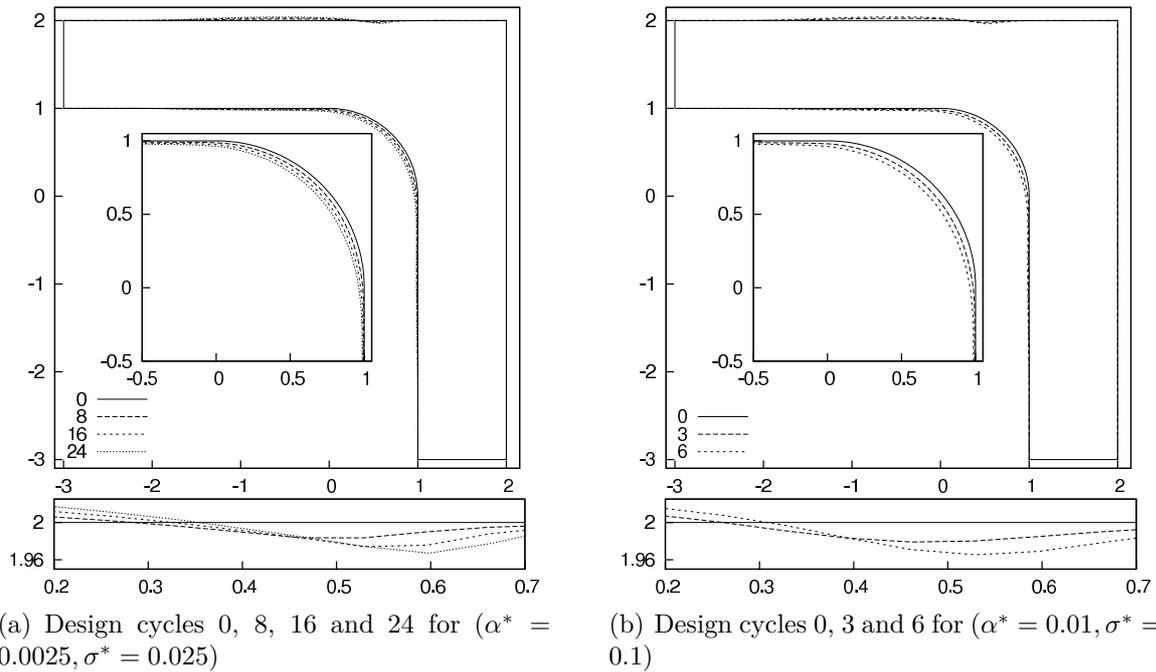


Figure 11.10.: T-junction at $Re = 50,000$ with low-Re wall boundary condition; shape evolution depending on step size α^* and filter width σ^*

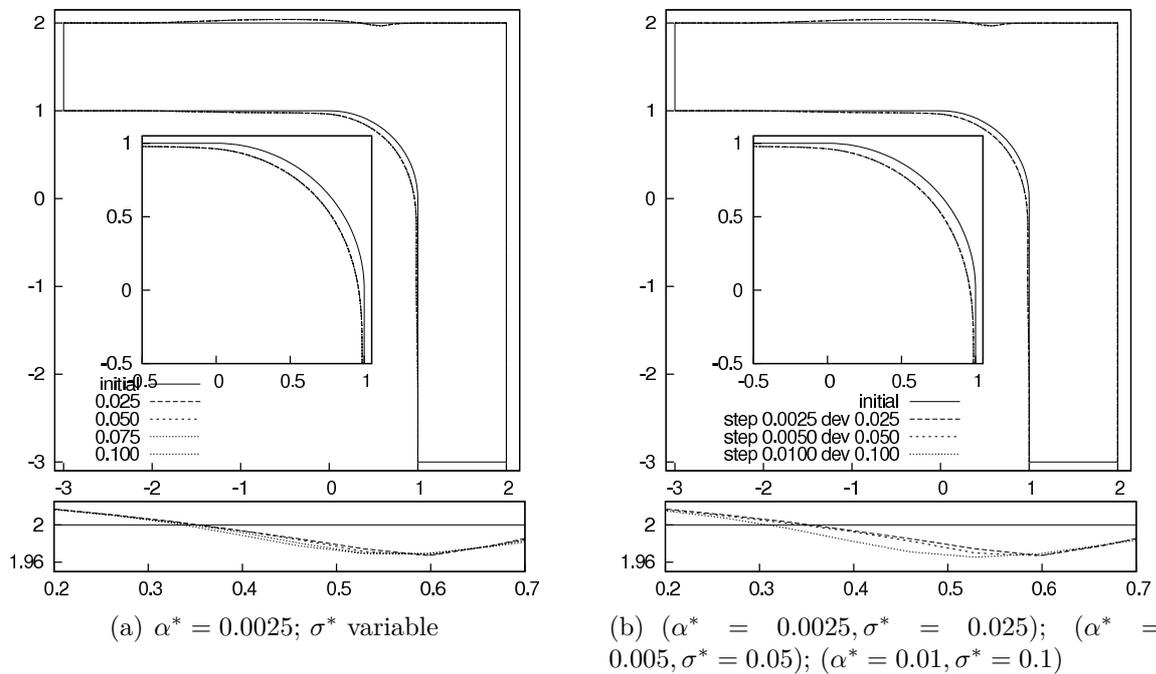


Figure 11.11.: T-junction at $Re = 50,000$ with low-Re wall boundary condition; final shapes depending on step size α^* and filter width σ^*

objective functional was inversely proportionally linked to the step size $0.025 < \alpha^* < 0.2$. The objective functional was reduced by approximately 28 per cent.

The final configurations for $(\alpha^* = 0.1; 0 < \sigma^* < 0.1)$ almost coincide, Figure 11.14(a), as was previously indicated by the optimisation histories of different J^* for variable σ^* , Figure 11.13(a). The shape evolution is illustrated in Figure 11.14(b) for $(\alpha^* = \sigma^* = 0)$. The shape modifications achieved during design cycles 1 to 10 are disproportionately high, for example on the upper wall, due to larger amplitudes of the sensitivity derivative. According to Figure 11.14(c), the final shapes obtained at a reduced number of pressure-correction iterations per design cycle are almost identical.

The design iteration can be regarded as a succession of small boundary shifts, $\delta n = -\alpha \bar{G}$, applied to the design surface. Primal and dual solutions were re-initialised with the results of the previous step, which reduces the necessary number of pressure-correction iterations per design cycle. Instead of accurately resolving the optimisation history of flow and objective functional, flow and shape can be iterated in the optimisation loop. In the limit of one outer iteration per design cycle, this approach is known as one-shot or piggy-back method. The evolution of the objective functional (11.13) obtained with a limited number of pressure-correction iterations per design cycle is shown in Figure 11.15(a) and Figure 11.15(b) for $(\alpha^* = 0.05, \sigma^* = 0)$ and $(\alpha^* = 0.1, \sigma^* = 0)$, respectively. In either case a maximum of 100 cycles per design step results in similar optimisation histories. A further reduction led to non-smooth curves, however, the final values J^* are very similar independent of the number of pressure-correction cycles. For $(\alpha^* = 0.05, \sigma^* = 0)$, the number of pressure-correction cycles could be reduced down to 30, whereas approximately 40 cycles were necessary to achieve convergence for $(\alpha^* = 0.1, \sigma^* = 0)$. Given that domain boundary and flow do not change topologically, I assume that it is possible to reduce the number of pressure-correction cycles further if the step size α^* is reduced. All optimisation runs started from a converged primal and dual solution since, in parts, the sensitivity derivatives calculated in the early pressure-correction iterations considerably differed in amplitude and direction.

Concluding Remarks

It was observed that the rough sensitivity distributions obtained in conjunction with low-Re wall boundary conditions require a preconditioning before they are applied to the reference shape. The roughness is expected to be caused by the schemes for the calculation of surface metrics and the implementation of low-Re wall boundary conditions; it becomes apparent on curved boundaries with highly-stretched grid cells, cf. Section 9.2. The filtering technique outlined in Section 10.3 was applied here using various combinations of filter width and step size. Due to the deficits of the simple mesh-deformation approach in conjunction with highly stretched cells in the boundary layer, the low-Re optimisation runs did not reach a local optimum. However, the objective functional value monotonously decreased until the mesh deformation failed. The optimisation runs conducted with the high-Re boundary condition allowed for larger deformations, which have reached a point of zero-slope of the objective functional. The raw sensitivity derivative was found to be smooth enough to be directly applied to the reference shape. Advection treatments

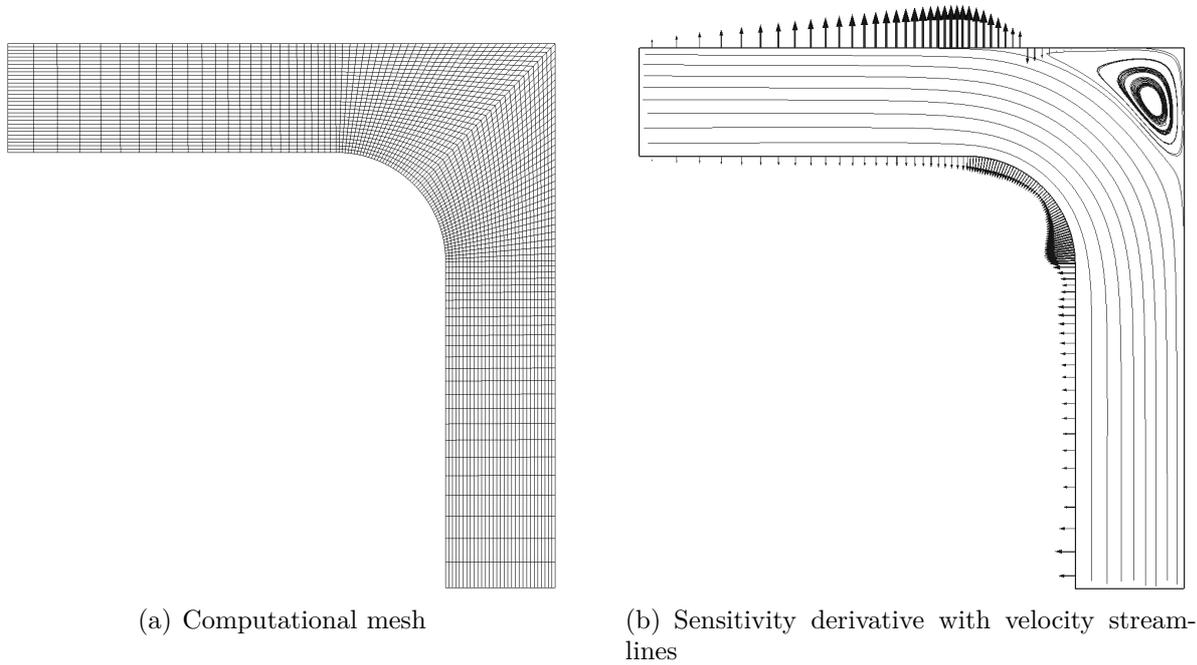


Figure 11.12.: T-junction at $Re = 50,000$ with high-Re wall boundary condition; initial configuration

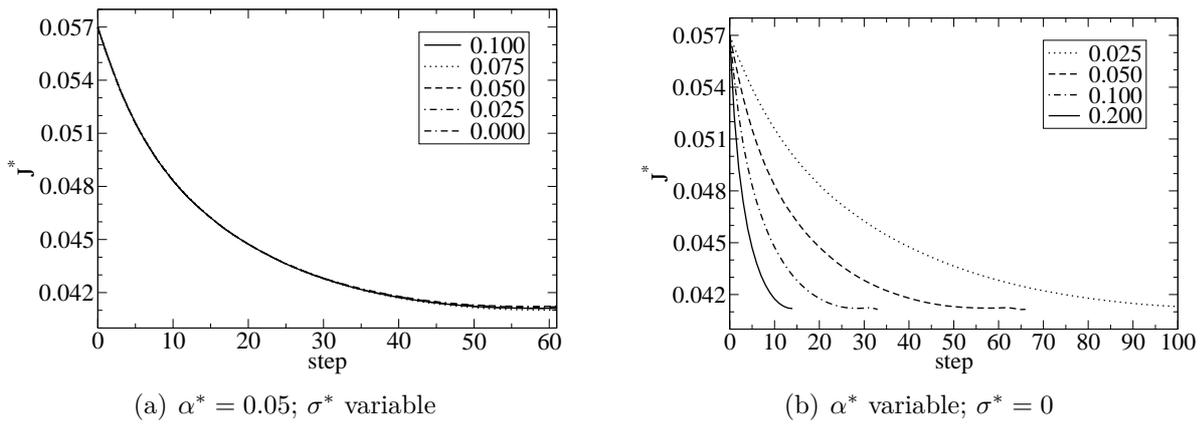


Figure 11.13.: T-junction at $Re = 50,000$ with high-Re wall boundary condition; evolution of power loss J^*

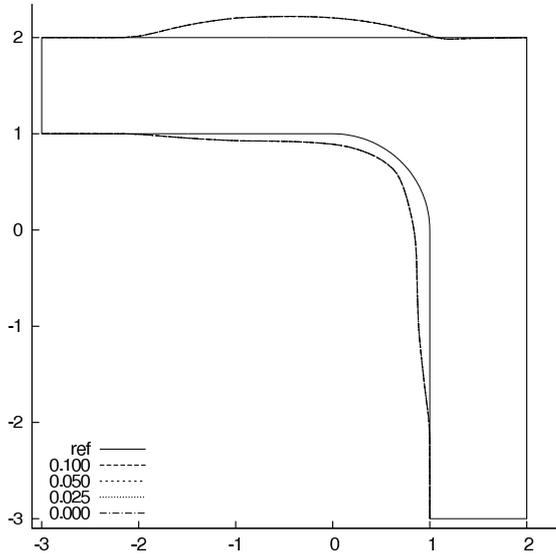
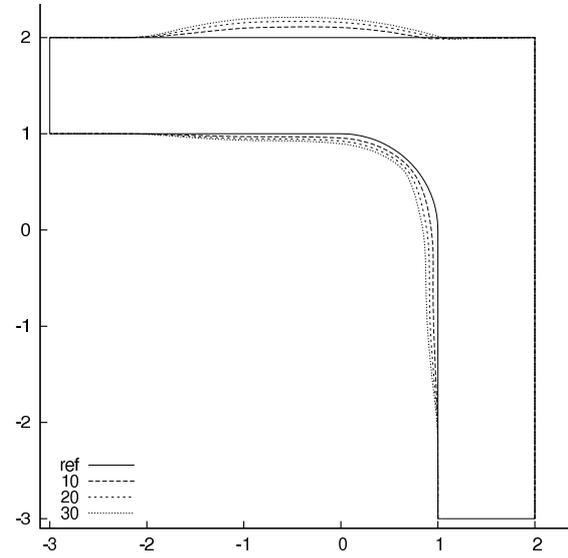
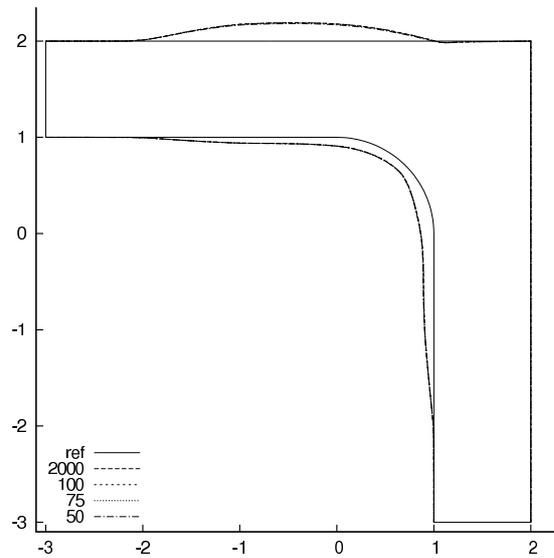
(a) Final shapes for $\alpha^* = 0.1$; σ^* variable(b) Intermediate shapes after n cycles for $\alpha^* = 0.1$; $\sigma^* = 0$ (c) Final shapes obtained with variable number of outer iterations per design cycle; $\alpha^* = 0.1$; $\sigma^* = 0$

Figure 11.14.: T-junction at $Re = 50,000$ with high- Re wall boundary condition; shape evolution and final shapes

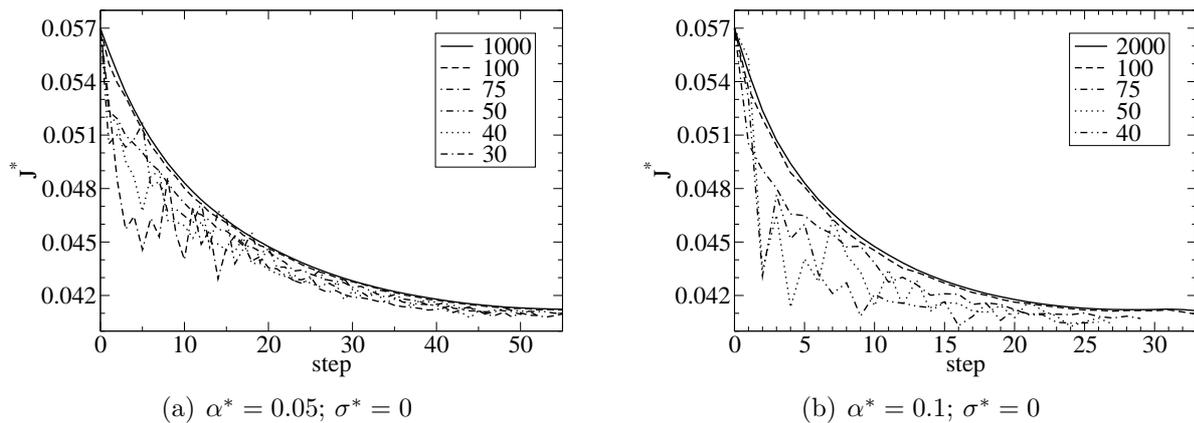


Figure 11.15.: T-junction at $Re = 50,000$ with high-Re wall boundary condition; evolution of power loss J^* using a limited number of pressure-correction iterations per design cycle

ADV1/2 overestimate the advection term in the first cell layer in conjunction with high-Re wall boundary conditions (cf. Subsection “Advection term”, page 95). This is overcome by the flux-based scheme ADV3. However, when the Reynolds number and the relative height of the first cell layer are increased to a higher level, the current implementation requires a stabilisation to bear the wall-normal source of adjoint momentum induced by the advection term in the first cell layer in combination with high-Re boundary conditions.

Using a reinitialisation based on the previous optimisation step allowed to limit the number of pressure-correction cycles per optimisation step, which has accelerated the optimisation process. The maximum number of pressure-correction steps per design cycle can be reduced with the step size α . Mind that the flow fields estimated by low- and high-Re wall boundary conditions differ, which is reflected in the sensitivity derivatives based on a linearisation about the corresponding flow solution. This particularly affects areas of flow separation or stagnation, where the high-Re assumption is generally invalid.

11.3. Optimisation of a 3D Double-Bent Pipe

The geometry of a 3D, double-bent pipe section is shown in Figure 11.16 with its main dimensions. The boundary shape was optimised to reduce the nondimensional power loss (11.13) between in- and outflow (Γ_O):

$$J^* = \frac{J}{\rho U_{\text{in}}^3 D^2}. \quad (11.16)$$

D and U_{in} denote the pipe diameter and the homogeneous inlet velocity, respectively. As for the T-junction case, design and objective surface are separate, $\Gamma_O \cap \Gamma_D = \emptyset$, i.e. only local contributions G_L to the sensitivity derivative were considered. The bulk Reynolds number based on D and U_{in} was $Re = 20,000$. The turbulence quantities at the inlet were defined according to Eqn. (9.26). The LU(D)DS convection scheme was used for primal and adjoint computations. The adjoint advection term was discretised according to ADV3. The sensitivity derivative G was filtered and subsequently applied to the initial boundary shape. A linear ramp function was multiplied to the gradient \bar{G} , $2D$ downstream of the inlet over $1.5D$ in the downstream direction and $4D$ upstream of the outlet over $2D$ in the upstream direction.

Low-Re Wall Boundary Condition

The quasi-structured grid used for the low-Re calculations is shown in Figure 11.16. The pipe's cross-section was subdivided into 1625 faces, 70 subdivisions were defined in the longitudinal direction. The Lien-Leschziner k - ε turbulence model [83] was used for the low-Re study, freezing the primal turbulence field in the sensitivity analysis. According to Figure 11.17, the nondimensional wall distance y^+ was of the order of unity. Areas of high shear stress can be identified in the y^+ map since the first cell layer had a constant height. Recirculation or stagnation zones were observed behind the first and the second bend.

A comparison of different σ^* along with $\alpha^* = 0.001$, Figure 11.18(a), indicates that the narrow filter kernels led to a steeper descent in the early steps but got stuck earlier than the wider filter kernels. Wide filter kernels tend to reduce the amplitudes of the derivative which may retard the optimisation. Local trends in the small length scales may be eliminated by a wide filter kernel. However, I assume that more distant and more attractive optima can only be reached if the small length scales in the derivative are filtered out. Figure 11.18(b) shows the history of the nondimensional power loss J^* for different step sizes α^* using a constant filter width $\sigma^* = 0.2$. Primal and adjoint computations were reinitialised with the solution of the previous design cycle and the number of pressure-correction cycles per design step was limited to 250. The required number of optimisation cycles inversely proportionally scales with the step size α^* .

The grey boundary shape, Figure 11.19, outlines the final geometry obtained for ($\alpha^* = 0.002, \sigma^* = 0.2$) together with the initial geometry plotted in red. Closeups of the first and the second bend are shown in Figure 11.20, slicing the pipe of optimisation cycles 0, 10 and 40 in black, red and green, respectively. The pipe has widened in large part.

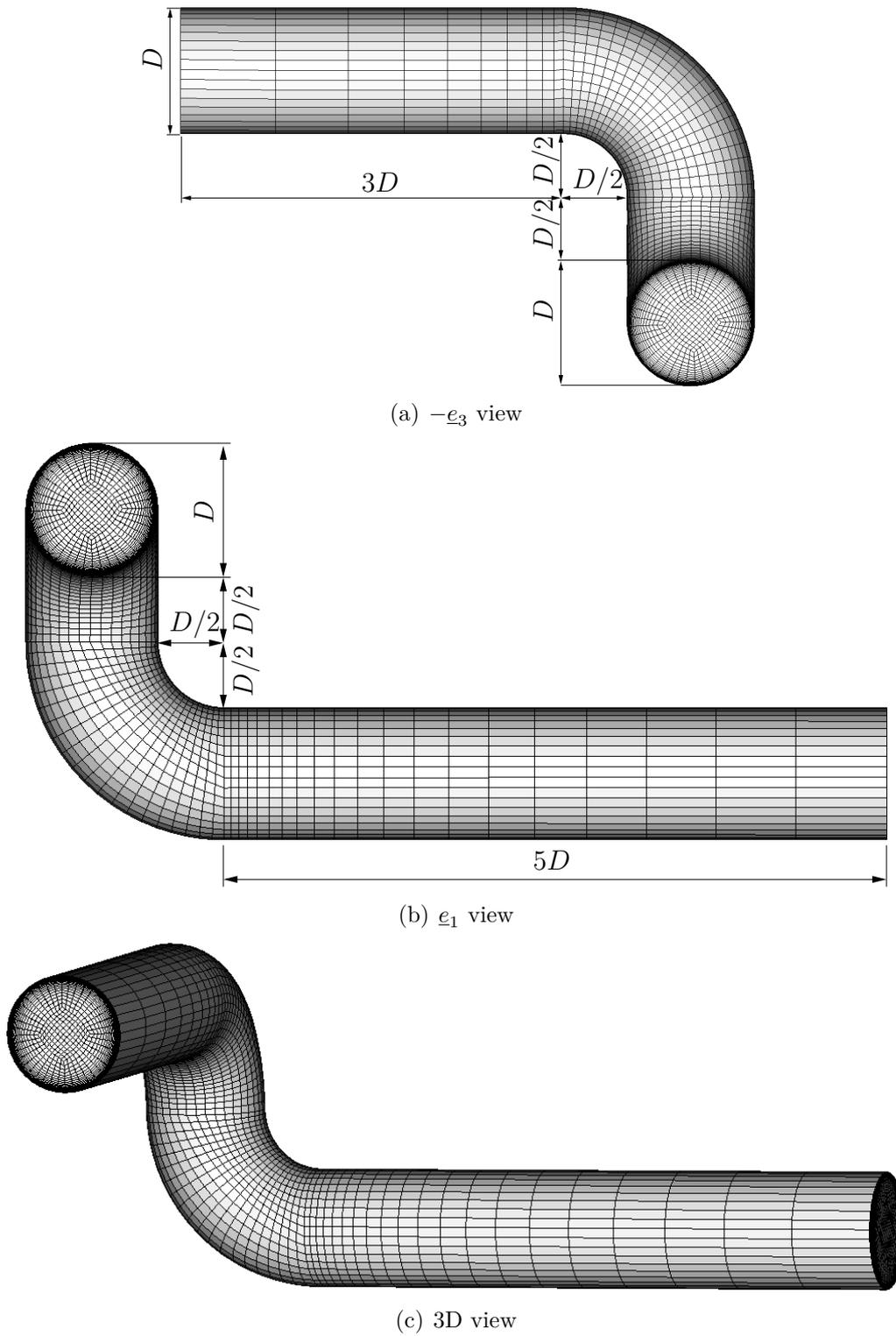


Figure 11.16.: Double-bent pipe at $Re = 20,000$ with low-Re wall boundary condition; surface mesh and dimensions for reference shape

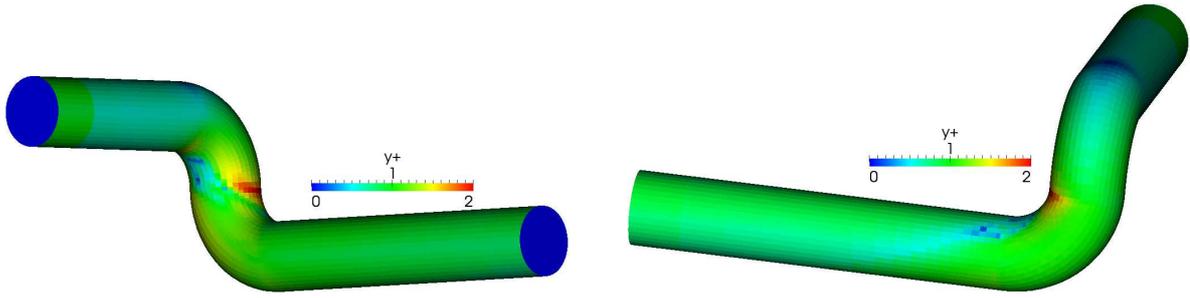


Figure 11.17.: y^+ distribution for double-bent pipe at $Re = 20,000$ with low-Re wall boundary condition

On the low- x_1 boundary between the elbows the pipe boundary has gradually contracted in order to reduce the recirculation area. The power loss between inlet and outlet was reduced by approximately 20 per cent.

High-Re Wall Boundary Condition

The same case was investigated in conjunction with a high-Re boundary condition imposed on the no-slip walls. The standard $k-\varepsilon$ turbulence model [70] was applied freezing the turbulence field in the adjoint calculation. The cross-section was divided into 300 patches. The subdivision in the longitudinal direction was taken from the low-Re mesh. The y^+ distribution over the pipe wall, Figure 11.21, qualitatively differs from its low-Re counterpart, Figure 11.17, due to different flow topologies. The high-Re model was applied to the entire wall surface, including recirculation or stagnation zones where $y^+ < 30$. The assumption of the logarithmic law-of-the-wall (7.64) underlying the high-Re boundary model is generally invalid in such areas. The sensitivity derivative obtained for the initial configuration is shown in Figure 11.22.

Several optimisation runs were performed with $\alpha = 0.2, 0.4$ and 0.8 , along with $\sigma^* = 0, 0.025, 0.05, 0.1$ and 0.2 . According to Figure 11.23, the optimisation runs based on the unfiltered sensitivity performed best independently of the step size α^* . A comparison of the optimisation histories for different α^* shows that the required number of optimisation cycles is roughly inversely proportional to α^* . The final shape obtained for $(\alpha^* = 0.8, \sigma^* = 0)$ is depicted in Figure 11.24. The deformation is much larger than in the low-Re case. The boundary shapes of the bends have become smoother, i.e. the local radii have virtually increased. A pronounced bulge has developed upstream of the first bend, cf. Figure 11.24(c) top left. The effect of the ramp function used to fade out the amplitudes between $2D$ and $D/2$ upstream of the first bend can be seen in Figure 11.24(b). A kink has developed at the intersection of the elbows, which is attributed to the high rate of change in the distribution of the sensitivity derivative G , cf. Figure 11.22(b). After the optimisation run, the predicted power loss was reduced by approximately 44 per cent.

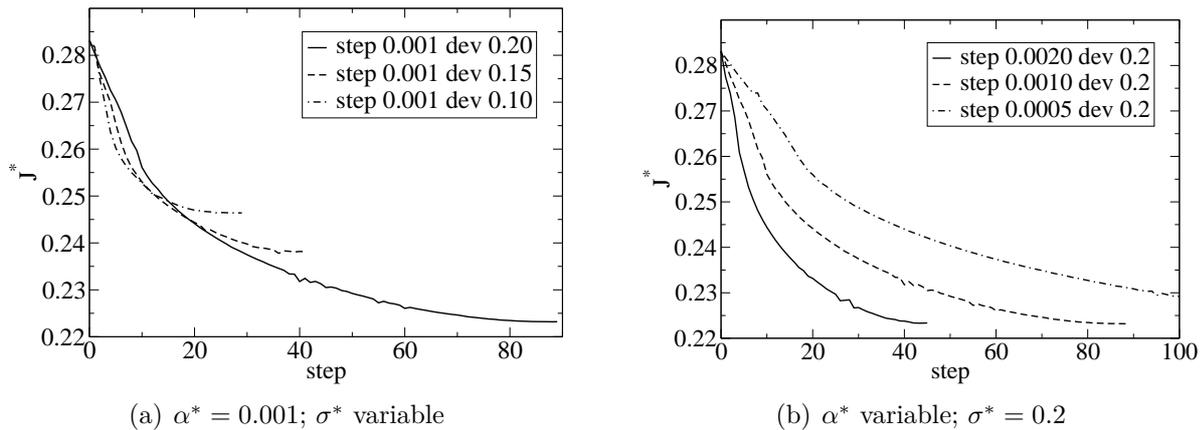


Figure 11.18.: Double-bent pipe at $Re = 20,000$ with low-Re wall boundary condition; evolution of power loss J^* depending on filter width σ^* and step size α^*

Concluding Remarks

Several optimisation runs were conducted for the turbulent 3D pipe case using low- and high-Re wall boundary conditions. The general trends observed in the 2D cases were confirmed: Whereas the high-Re derivative is sufficiently smooth, it is necessary to filter the low-Re-based derivative before it is applied to the geometry (cf. concluding remarks on page 158 and Section 9.2). Compared to the high-Re case, the low-Re optimisation runs have led to shapes lying closer to the reference configuration. This has already been observed in the 2D cases in Section 9.2. Different models predict different flow fields which, in turn, show different sensitivity derivatives. A discrepancy in the primal separation prediction, for example, leads to very different distributions of the sensitivity derivative. The convergence (or stagnation) of the objective functional achieved in some of the optimisation runs may be an indication that a local optimum has been found. However, the resulting shapes are considered as improved designs rather than strict local optima. Among the open questions are: (a) It is unclear whether the evolution of irregularities, such as kinks, avoids a further reduction of the objective functional. Do further local shape improvements cancel with developing shape deficits such as steepening irregularities? (b) The filtering is merely an approximation to the so-called ‘‘Sobolev gradient’’ and the differences have to be studied systematically. (c) The linear ramping applied to the derivative next to inlet and outlet is no gradient-projection method. Hence, a shape convergence next to inlet and outlet is not expected. A more elaborate, mathematical handling of constraints should be pursued in future studies to enforce that inlet and outlet boundaries remain untouched by the shape optimisation. (d) How does the approximation error change during the shape evolution and how trustworthy are the RANS models in combination with high- and low-Re boundary conditions for the current shape?

These aspects are beyond the scope of this study and need to receive particular attention. However, the shape evolution takes a reasonable direction, reduces the objective functional mostly monotonously and gives a clear indication how to modify the shape.

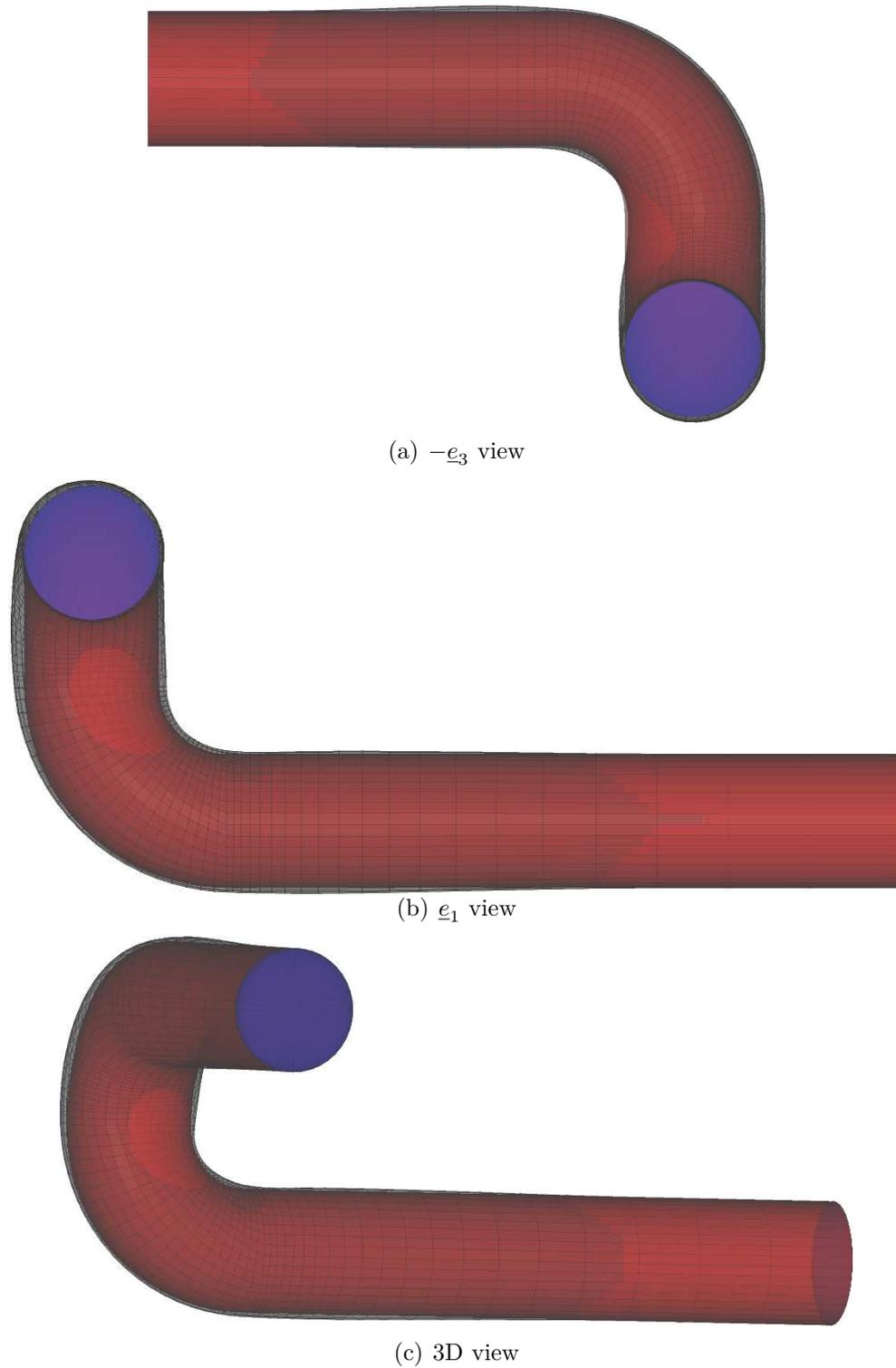


Figure 11.19.: Double-bent pipe at $Re = 20,000$ with low-Re wall boundary condition; final shape (grey with surface mesh) obtained with $\alpha^* = 0.002$, $\sigma^* = 0.2$ together with inial shape (red)

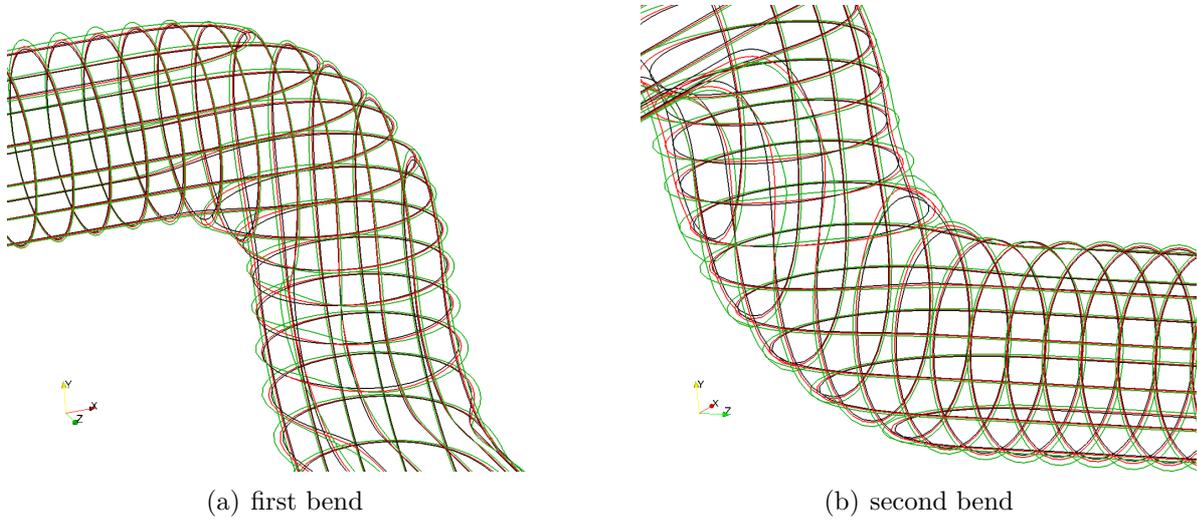


Figure 11.20.: Double-bent pipe at $Re = 20,000$ with low-Re wall boundary condition; shape evolution for $\alpha^* = 0.002$, $\sigma^* = 0.2$: black – initial, red – cycle 10, green – cycle 40

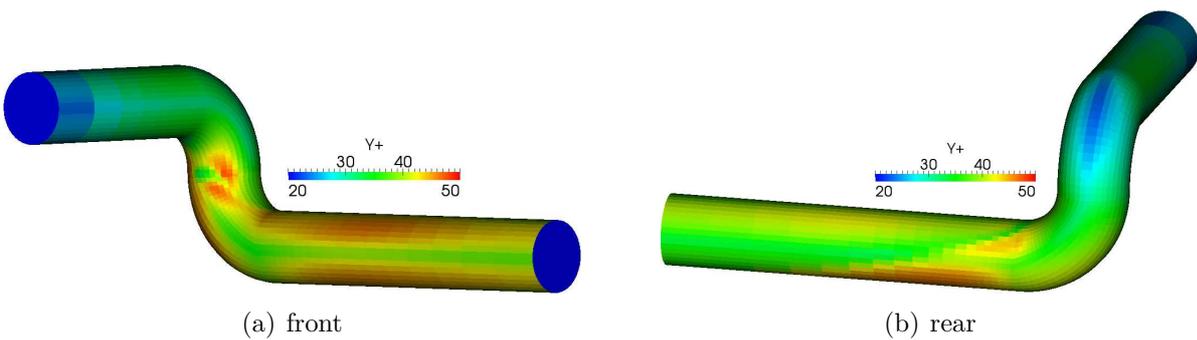


Figure 11.21.: y^+ distribution for double-bent pipe at $Re = 20,000$ with high-Re wall boundary condition

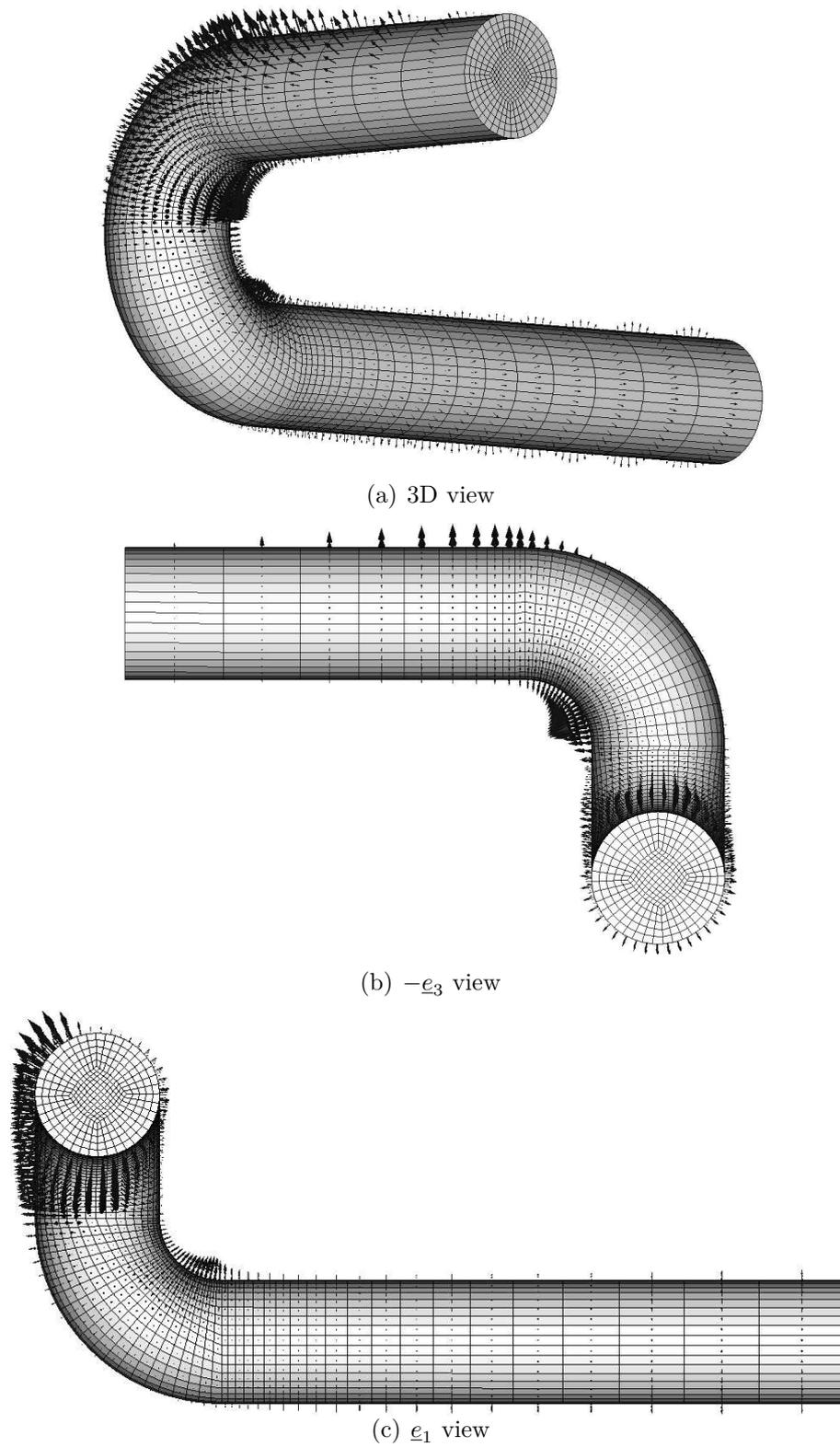


Figure 11.22.: Double-bent pipe at $Re = 20,000$ with high- Re wall boundary condition; sensitivity derivative for initial configuration

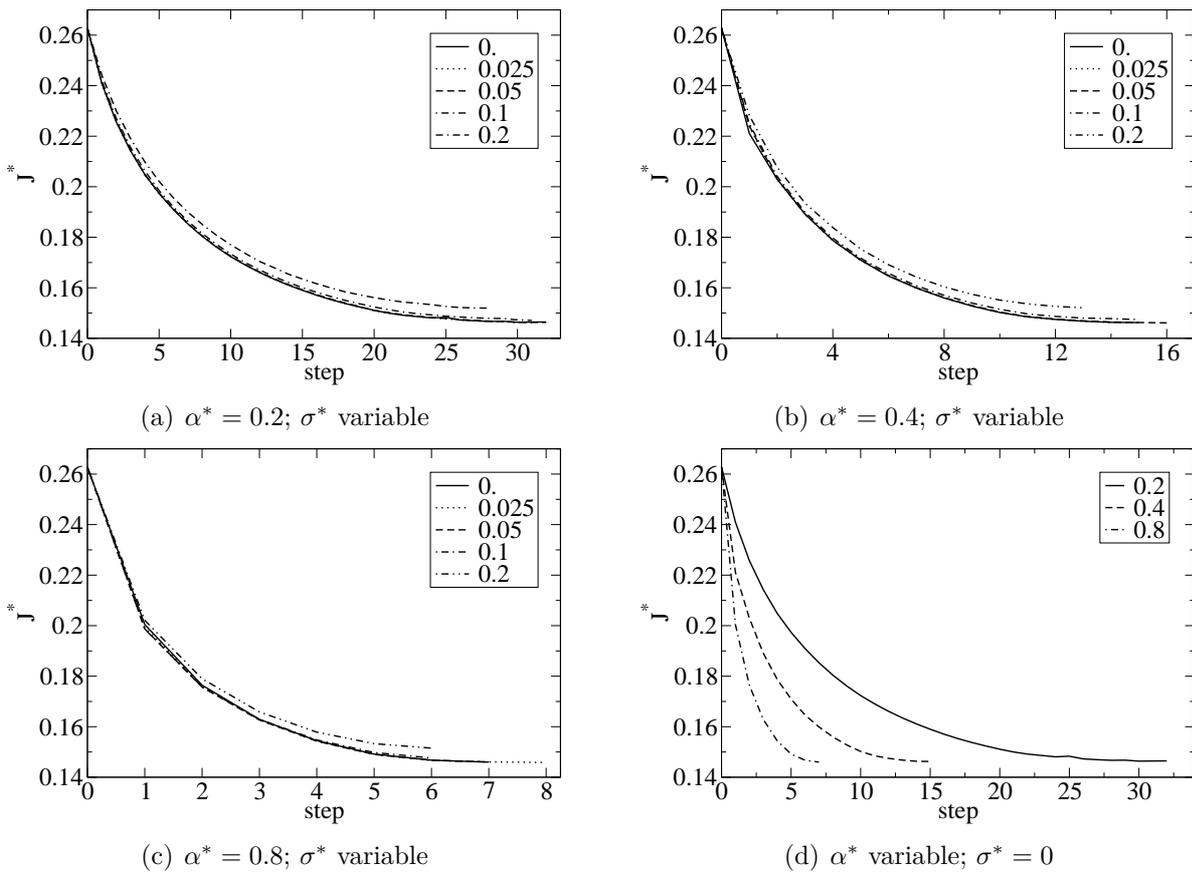
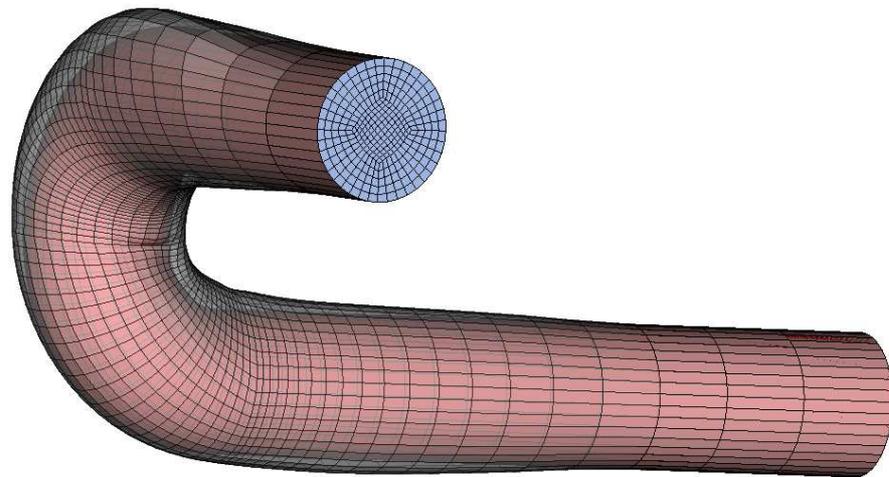


Figure 11.23.: Double-bent pipe at $Re = 20,000$ with high-Re wall boundary condition; evolution of power loss J^* depending on step size α^* and filter width σ^*



(a) 3D view

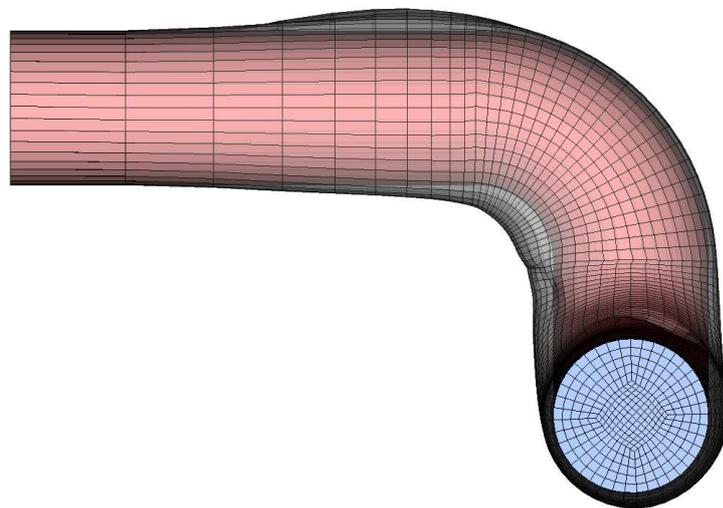
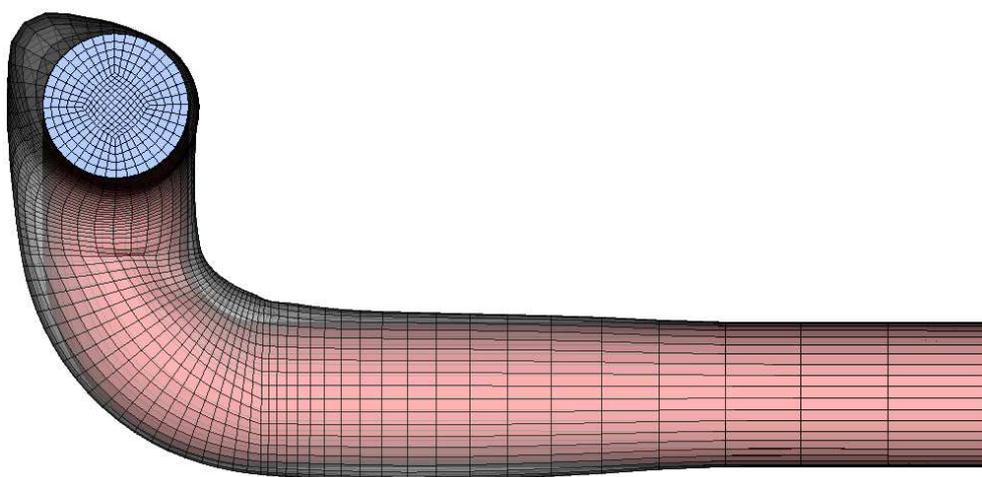
(b) $-\mathbf{e}_3$ view(c) \mathbf{e}_1 view

Figure 11.24.: Double-bent pipe at $Re = 20,000$ with high-Re wall boundary condition; final shape obtained with $\alpha^* = 0.8$, $\sigma^* = 0$

12. Outlook and Conclusions

Adjoint Navier–Stokes concepts that allow for an efficient shape optimisation in the face of many shape parameters were derived and implemented in this study. Whereas adjoint methods for compressible flow and their discretisation have been a subject of exhaustive research over the past twenty years, the adjoint discretisation and solution schemes for viscous, incompressible flow are less represented and described in the literature.

Starting from a continuous adjoint analysis of the incompressible Navier–Stokes problem, consistent adjoint discretisation schemes for convection (upstream, central, MUSCL-based QUICK, LUDS and CUI schemes) and diffusion terms were derived from the unstructured, cell-centred FV discretisation. The Picard iteration used in the primal solver neglects the advection term which is the second contribution arising from a complete linearisation of the convection term of the momentum equation. In the adjoint approach, a complete linearisation is necessary and the advection term needs to be discretised. Different schemes, ADV0–3, were proposed, discretised and investigated. Based on the primal discretisation a unified formulation for adjoint high- and low-Re wall boundary models was developed. This includes the discrete schemes for (a) the adjoint wall boundary condition and (b) the reduced, boundary-based sensitivity equation both of which are a valid, adjoint-consistent approximation to the continuous adjoint PDE problem.

Adjoining a semi-implicit, segregated pressure-projection scheme requires to construct a sequence of adjoint or transposed solutions corresponding to the primal, segregated approach. The primal, SIMPLE-based pressure-projection scheme was adapted to solve the adjoint problem, wherein the primal iteration was not exactly traversed in reverse. The method was implemented in parallel in the framework of the unstructured, state-of-the-art FV method FreSCo⁺. A user-coding interface allows for a flexible, user-defined handling of contributions to the adjoint equations that stem from the considered objective functional. It keeps the adjoint code both versatile and short.

The method was validated against analytical adjoint solutions derived for a Couette flow problem. It was solved numerically in Cartesian coordinates so that the adjoint convection terms are non-zero and the corresponding discretisation schemes could be assessed. The higher-order advection schemes showed the best agreement. The direct-differentiation method was used to verify the adjoint solver for more complex 2D configurations. Internal and external problems of turbulent flow were investigated which, depending on the application, show a fair or good agreement of the sensitivity derivatives obtained from the adjoint and the direct-differentiation method. The proposed discretisation schemes for the advection term, ADV0–3, were verified individually in conjunction with low- and high-Re wall descriptions for the adjoint boundary conditions and the sensitivity equation. The concepts proposed for interior and exterior declarations of force objectives in the adjoint method could be verified. The study indicates that the local sensitivity derivative pre-

dominates over geometric and convective contributions, which are of minor influence for the investigated configuration.

The adjoint method was introduced to ship wake design. To guide a manual hull re-design towards an improved wake field, a wake objective functional was derived and applied to the right-hand side of the adjoint problem. More elaborate wake objective functionals allowing for a well-founded quantification of the wake quality should be considered in future studies since the success of the adjoint method depends on the validity of the considered objective functional. Additionally, the effect of the propulsion should be considered in the optimisation. The sensitivity distributions, obtained from a differential, boundary-based expression and low-Re wall boundary conditions for turbulent flow, were too rough to be directly applied to the reference shape. These oscillations are attributed to the schemes used for the computation of surface metrics and to the formulation of the primal low-Re wall boundary condition in combination with curved boundaries next to highly-stretched boundary-layer cells. An explicit, filtering-based gradient preconditioning method was presented and applied in this work. It is first-order equivalent to the well-known “Sobolev-smoothing”, easy to implement and intuitive to use. Supplemented by a straight-forward grid deformation approach, a selection of 2D and 3D optimisation runs were carried out for ducted flows following the direction of the “filtered steepest descent”. Guided by the adjoint-based sensitivity derivative, the initial shapes were successfully improved in a CAD-free environment. The design space is not a-priori limited and the number of external software tools requiring tailored data interfaces is small. The shapes obtained indicate how to modify a design or the underlying parametrisation. However, the handling of geometric design constraints such as “hard points” or “feature lines” based on the CFD mesh is not straight-forward and requires some further refinement. The elementary mesh-deformation approach based on a diffusion equation limited the maximum deformation that could be achieved—particularly in combination with low-Re RANS grids of high cell aspect-ratios in the boundary layer. In the context of industrial CFD, the improved shape has to be passed back to the CAD/CAE system to be processed further.

In future studies the frozen-turbulence approach should be extended by a complete differentiation including the turbulence variables and the associated PDE. Several terms arise from a complete differentiation leading to a strong cross-coupling of the adjoint equations of momentum and turbulence. Many CFD codes solve the turbulence equations separately so that a complete linearisation of the coupled system is usually unavailable in the flow solver. Discretisation schemes have to be found for the additional terms and the cross-coupling must be treated with due care in a segregated approach. To obtain an efficient and robust adjoint method that calculates the full derivative for turbulent flow problems and complex geometries, it may be necessary to develop (partially) coupled solution algorithms. Some optimisation problems require to extend the adjoint system by additional PDE constraints, for example the VOF or level-set equations for free-surface flow or the energy equation. Unsteady simulations require to solve the associated adjoint problem backwards in time. The existing approach must be complemented by an adjoint time-stepping scheme. Such schemes are very memory-intensive since the primal solution must be provided in every adjoint time step. Check-pointing schemes [37, 46, 163, 164] can

be used to trade memory storage for CPU time. A bundle of efficient measures is proposed by Stück et al. [154]. Simplified approaches proposed by Nadarajah [101] and Nadarajah et al. [100] may lead to sufficiently accurate results depending on the application at hand.

To exploit the potential of the adjoint sensitivity approach in practical engineering, it has to be integrated into the CAD/E-based design chain. Tailored data interfaces, currently being developed in the framework of the FormPro project (Kröger, Stück, and Rung [80]) are necessary to link the sensitivity information to the underlying shape parametrisation, e.g. Soto et al. [151].

The adjoint method implemented during this work can also be used for other optimisation tasks involving a large number of parameters. Among them are topology optimisation, active or passive flow control, goal-oriented error analysis, mesh adaptation and convergence improvement.

A. Derivation of Adjoint Navier–Stokes Equations

A.1. Variation of the Navier–Stokes Equations

Assuming that the fluid density ρ , the viscosity $\mu_{\text{eff}} = \mu + \mu_T$ and the body force density \underline{f} are not subject to variations, the local variation of the residual of the incompressible momentum equations (4.9) reads

$$\delta R_i = \rho \delta U_j \frac{\partial U_i}{\partial x_j} + \rho U_j \frac{\partial \delta U_i}{\partial x_j} - \frac{\partial \delta \pi_{ij}}{\partial x_j} = 0 \quad \text{in } \Omega \quad (\text{A.1})$$

with

$$\delta \pi_{ij} = -\delta p \delta_{ij} + 2 \mu_{\text{eff}} \delta S_{ij} \quad \text{and} \quad \delta S_{ij} = \frac{1}{2} \left(\frac{\partial \delta U_i}{\partial x_j} + \frac{\partial \delta U_j}{\partial x_i} \right).$$

The linearised form of the equation of continuity (4.10) is

$$\delta Q = -\frac{\partial \delta U_i}{\partial x_i} = 0 \quad \text{in } \Omega. \quad (\text{A.2})$$

A.2. Integration by Parts

In order to devise the adjoint equations, the derivatives are transferred from the local variations $(\delta U_i, \delta p)$ to the adjoint multipliers (\hat{U}_i, \hat{p}) through integration by parts:

$$\begin{aligned} \int d\Omega \left[\hat{U}_i \delta U_j \frac{\partial U_i}{\partial x_j} \right] &= \oint d\Gamma_j \left[\hat{U}_i \delta U_j U_i \right] - \int d\Omega \left[\frac{\partial}{\partial x_j} \left(\hat{U}_i \delta U_j \right) U_i \right] \\ &= \oint d\Gamma_i \left[\delta U_i U_j \hat{U}_j \right] - \int d\Omega \left[\delta U_i U_j \frac{\partial \hat{U}_j}{\partial x_i} \right], \end{aligned} \quad (\text{A.3})$$

$$\begin{aligned} \int d\Omega \left[\hat{U}_i U_j \frac{\partial \delta U_i}{\partial x_j} \right] &= \oint d\Gamma_j \left[\hat{U}_i U_j \delta U_i \right] - \int d\Omega \left[\frac{\partial}{\partial x_j} \left(\hat{U}_i U_j \right) \delta U_i \right] \\ &= \oint d\Gamma_j \left[\delta U_i U_j \hat{U}_i \right] - \int d\Omega \left[\delta U_i U_j \frac{\partial \hat{U}_i}{\partial x_j} \right], \end{aligned} \quad (\text{A.4})$$

$$\begin{aligned} \int d\Omega \left[\hat{U}_i \frac{\partial}{\partial x_j} (2 \mu_{\text{eff}} S_{ij}) \right] &= \oint d\Gamma_j \left[\hat{U}_i 2 \mu_{\text{eff}} \delta S_{ij} - \delta U_i 2 \mu_{\text{eff}} \hat{S}_{ij} \right] \\ &\quad + \int d\Omega \left[\frac{\partial}{\partial x_j} \left(2 \mu_{\text{eff}} \hat{S}_{ij} \delta U_i \right) \right], \end{aligned} \quad (\text{A.5})$$

$$\int d\Omega \left[\hat{U}_i \frac{\partial \delta p}{\partial x_i} \right] = \oint d\Gamma_j [\hat{U}_j \delta p] - \int d\Omega \left[\delta p \frac{\partial \hat{U}_j}{\partial x_j} \right], \quad (\text{A.6})$$

$$\int d\Omega \left[\hat{p} \frac{\partial \delta U_j}{\partial x_j} \right] = \oint d\Gamma_j [\hat{p} \delta U_j] - \int d\Omega \left[\delta U_j \frac{\partial \hat{p}}{\partial x_j} \right], \quad (\text{A.7})$$

with

$$\hat{S}_{ij} = \frac{1}{2} \left(\frac{\partial \hat{U}_i}{\partial x_j} + \frac{\partial \hat{U}_j}{\partial x_i} \right).$$

B. Advection Formulations ADV2/3

An alternative continuous formulation of the advection term called ADV2 is obtained via integration by parts. The Cartesian description given in Appendix B.1 is rewritten in a system of physical cylinder coordinates in Appendix B.2. Formulation ADV2 is compared against the face-based representation ADV3 derived from the primal discretisation in Appendix B.3.

B.1. Cartesian Coordinates

After integration by parts of the advection term, the local variation of the Lagrangian reads

$$\delta_L L = \oint d\Gamma_j \left[\left(\rho \left[\underline{U_i} \delta U_j + \delta U_i U_j \right] + \delta p \delta_{ij} - \mu_{\text{eff}} \left[\frac{\partial \delta U_i}{\partial x_j} + \frac{\partial \delta U_j}{\partial x_i} \right] \right) \hat{U}_i \right. \quad (\text{B.1a})$$

$$\left. + \left(-\hat{p} \delta_{ij} + 2 \mu_{\text{eff}} \hat{S}_{ij} \right) \delta U_i \right] \quad (\text{B.1b})$$

$$+ \int d\Omega \left[\left(-\rho U_j \left(\frac{\partial \hat{U}_j}{\partial x_i} + \frac{\partial \hat{U}_i}{\partial x_j} \right) - \frac{\partial}{\partial x_j} \left[2 \mu_{\text{eff}} \hat{S}_{ij} - \hat{p} \delta_{ij} \right] \right) \delta U_i - \frac{\partial \hat{U}_j}{\partial x_j} \delta p \right] \quad (\text{B.1c})$$

$$+ \int_{\Omega_O} \delta_L j_\Omega d\Omega + \int_{\Gamma_O} \delta_L j_\Gamma d\Gamma. \quad (\text{B.1d})$$

Compared to the formulation ADV1, the underlined terms are new (B.1a) or different (B.1c). The adjoint field equations in combination with the ADV2 formulation [e.g. 149, 110, 170, 171] are

$$-2 \rho U_j \hat{S}_{ij} = \frac{\partial}{\partial x_j} \left(2 \mu_{\text{eff}} \hat{S}_{ij} - \hat{p} \delta_{ij} \right) \quad \text{in } \Omega \setminus \Omega_O. \quad (\text{B.2})$$

The additional boundary term in (B.1a), $\rho \delta U_j U_i n_j$, arising from integration by parts of the advection term has to be considered in the derivation of the adjoint boundary conditions for the ADV2 formulation. The corresponding boundary conditions are listed in Table B.1. The adjoint gradient equation (6.39) is independent of the advection treatment and holds for ADV0–3.

B.2. Physical Cylinder Coordinates

Integration by parts of the advection term leads to the left-hand side contribution, $-\rho \underline{\nabla} \hat{U} \cdot \underline{U}$, to the adjoint momentum equations. The corresponding formulations for constant density and viscosity transformed into physical cylinder coordinates are:

Table B.1.: Summary of adjoint boundary conditions for advection scheme ADV2

boundary	objective	\hat{U}	\hat{p}
no-slip wall $\not\subset \Gamma_O$	–	$\hat{U} = \underline{0}$	$\hat{p}^n = 0$
no-slip wall $\subset \Gamma_O$	force	$\hat{U} = -\underline{d}^*$	$\hat{p}^n = 0$
inflow $\not\subset \Gamma_O$	–	$\hat{U} = \underline{0}$	$\hat{p}^n = 0$
inflow $\subset \Gamma_O$	power loss	$\hat{U} = \underline{U}$	$\hat{p}^n = 0$
symmetry	–	$\hat{U}^n = \hat{U}^{t,n} = 0$	$\hat{p}^n = 0$
prs. outflow $\not\subset \Gamma_O$	–	$\rho U^n \hat{U}_i + \rho U_j \hat{U}_j n_i + \mu_{\text{eff}} \frac{\partial \hat{U}_i}{\partial n} = \hat{p} n_i$	
prs. outflow $\subset \Gamma_O$	power loss	$\rho U^n \hat{U}_i + \rho U_j \hat{U}_j n_i + \mu_{\text{eff}} \frac{\partial \hat{U}_i}{\partial n} - \frac{\rho}{2} U_j^2 n_i - \rho U^n U_i = \hat{p} n_i$	

Radial Adjoint Momentum Equation (\tilde{e}^r direction):

$$\begin{aligned}
& -\rho \left(U^r \frac{\partial \hat{U}^r}{\partial r} + \frac{U^\varphi}{r} \frac{\partial \hat{U}^r}{\partial \varphi} + U^z \frac{\partial \hat{U}^r}{\partial z} - \frac{U^\varphi \hat{U}^\varphi}{r} \right) \\
& \quad - \rho \left(U^r \frac{\partial \hat{U}^r}{\partial r} + U^\varphi \frac{\partial \hat{U}^\varphi}{\partial r} + U^z \frac{\partial \hat{U}^z}{\partial r} \right) = \\
& \quad - \frac{\partial \hat{p}}{\partial r} + \mu \left(\frac{\partial^2 \hat{U}^r}{\partial r^2} + \frac{1}{r} \frac{\partial \hat{U}^r}{\partial r} - \frac{\hat{U}^r}{r^2} + \frac{1}{r^2} \frac{\partial^2 \hat{U}^r}{\partial \varphi^2} + \frac{\partial^2 \hat{U}^r}{\partial z^2} - \frac{2}{r^2} \frac{\partial \hat{U}^\varphi}{\partial \varphi} \right) \tag{B.3}
\end{aligned}$$

Circumferential Adjoint Momentum Equation (\tilde{e}^φ direction):

$$\begin{aligned}
& -\rho \left(U^r \frac{\partial \hat{U}^\varphi}{\partial r} + \frac{U^\varphi}{r} \frac{\partial \hat{U}^\varphi}{\partial \varphi} + U^z \frac{\partial \hat{U}^\varphi}{\partial z} + \frac{\hat{U}^r U^\varphi}{r} \right) \\
& \quad - \frac{\rho}{r} \left(U^r \frac{\partial \hat{U}^r}{\partial \varphi} + U^\varphi \frac{\partial \hat{U}^\varphi}{\partial \varphi} + U^z \frac{\partial \hat{U}^z}{\partial \varphi} + U^\varphi \hat{U}^r - \hat{U}^\varphi U^r \right) = \\
& \quad - \frac{\partial \hat{p}}{\partial \varphi} + \mu \left(\frac{\partial^2 \hat{U}^\varphi}{\partial r^2} + \frac{1}{r} \frac{\partial \hat{U}^\varphi}{\partial r} - \frac{\hat{U}^\varphi}{r^2} + \frac{1}{r^2} \frac{\partial^2 \hat{U}^\varphi}{\partial \varphi^2} + \frac{\partial^2 \hat{U}^\varphi}{\partial z^2} + \frac{2}{r^2} \frac{\partial \hat{U}^r}{\partial \varphi} \right) \tag{B.4}
\end{aligned}$$

Axial Adjoint Momentum Equation (\tilde{e}^z direction):

$$\begin{aligned}
& -\rho \left(U^r \frac{\partial \hat{U}^z}{\partial r} + \frac{U^\varphi}{r} \frac{\partial \hat{U}^z}{\partial \varphi} + U^z \frac{\partial \hat{U}^z}{\partial z} \right) - \rho \left(U^r \frac{\partial \hat{U}^r}{\partial z} + U^\varphi \frac{\partial \hat{U}^\varphi}{\partial z} + U^z \frac{\partial \hat{U}^z}{\partial z} \right) = \\
& \quad - \frac{\partial \hat{p}}{\partial z} + \mu \left(\frac{\partial^2 \hat{U}^z}{\partial r^2} + \frac{1}{r} \frac{\partial \hat{U}^z}{\partial r} + \frac{1}{r^2} \frac{\partial^2 \hat{U}^z}{\partial \varphi^2} + \frac{\partial^2 \hat{U}^z}{\partial z^2} \right) \tag{B.5}
\end{aligned}$$

The advection terms are underlined in the above equations. For the problem sketched in Figure 9.1, Eqns. (B.3) through (B.5) reduce to

$$\frac{\partial \hat{p}}{\partial r} = \rho U^\varphi \left(-\frac{\hat{U}^\varphi}{r} + \frac{\partial \hat{U}^\varphi}{\partial r} \right), \quad (\text{B.6})$$

$$0 = \frac{\partial^2 \hat{U}^\varphi}{\partial r^2} + \frac{1}{r} \frac{\partial \hat{U}^\varphi}{\partial r} - \frac{\hat{U}^\varphi}{r^2}. \quad (\text{B.7})$$

For the adjoint boundary conditions (9.18) and (9.19), the analytic solution of the problem (B.6) and (B.7) is obtained by integration:

$$\hat{U}^\varphi(r) = \hat{\alpha} \left(\frac{r}{r_O^2} - \frac{1}{r} \right) \quad (\text{B.8})$$

$$\hat{p}(r) = \hat{p}(r_I) + \rho \alpha \hat{\alpha} \left[2 \ln \left(\frac{r}{r_I} \right) + \left(\frac{r_I}{r} \right)^2 - 1 \right], \quad (\text{B.9})$$

with

$$\hat{\alpha} = \frac{r_I}{(r_I/r_O)^2 - 1}. \quad (\text{B.10})$$

Neglecting the advection terms (ADV0) leads to the solution

$$\hat{U}^\varphi(r) = \hat{\alpha} \left(\frac{r}{r_O^2} - \frac{1}{r} \right) \quad (\text{B.11})$$

$$\begin{aligned} \hat{p}(r) = \hat{p}(r_I) - \rho \alpha \hat{\alpha} \left[\frac{1}{2} \frac{r^2 - r_I^2}{r_O^2} - \left(\left[\frac{r_I}{r_O} \right]^2 + 1 \right) \ln \left(\frac{r}{r_I} \right) \right. \\ \left. - \frac{r_I^2}{2} \left(\frac{1}{r^2} - \frac{1}{r_I^2} \right) \right]. \end{aligned} \quad (\text{B.12})$$

Note that only the radial equation is affected by the advection treatment, leading to a different adjoint pressure distribution in radial direction, compare Eqns. (9.23), (B.9) and (B.12).

B.3. Comparison of Advection Schemes

Advection formulation ADV3 is a face-based scheme derived from the primal discretisation of the convective flux. According to Eqn. (8.27), the adjoint face fluxes are

$$\hat{F}^a \approx -\rho \hat{\lambda} \underline{U}_F \cdot \left(\hat{U}_N - \hat{U}_P \right) \Delta \underline{\Gamma}. \quad (\text{B.13})$$

For the Cartesian grid with equidistant cell spacings $(\Delta x, \Delta y)$ depicted in Figure B.3, the advective net flux based on advection formulation ADV3 and CDS face interpolation

reads:

$$\begin{aligned}
\frac{1}{\Delta\Omega} \sum_{e,w,n,s} \hat{F}^a &\approx \frac{\rho}{\Delta x \Delta y} \left[\left(\hat{F}_y^a|_n + \hat{F}_y^a|_s \right) \underline{e}_y + \left(\hat{F}_x^a|_e + \hat{F}_x^a|_w \right) \underline{e}_x \right] \\
&\approx \frac{\rho}{4\Delta y} \left[-(\underline{U}_N + \underline{U}_P) \cdot (\hat{\underline{U}}_N - \hat{\underline{U}}_P) + (\underline{U}_S + \underline{U}_P) \cdot (\hat{\underline{U}}_S - \hat{\underline{U}}_P) \right] \underline{e}_y + \\
&\quad \frac{\rho}{4\Delta x} \left[-(\underline{U}_E + \underline{U}_P) \cdot (\hat{\underline{U}}_E - \hat{\underline{U}}_P) + (\underline{U}_W + \underline{U}_P) \cdot (\hat{\underline{U}}_W - \hat{\underline{U}}_P) \right] \underline{e}_x \\
&\approx \frac{\rho}{2} \left[-\frac{\underline{U}_N \cdot \hat{\underline{U}}_N - \underline{U}_S \cdot \hat{\underline{U}}_S}{2\Delta y} + \frac{\underline{U}_N - \underline{U}_S}{2\Delta y} \cdot \hat{\underline{U}}_P - \frac{\hat{\underline{U}}_N - \hat{\underline{U}}_S}{2\Delta y} \cdot \underline{U}_P \right] \underline{e}_y + \\
&\quad \frac{\rho}{2} \left[-\frac{\underline{U}_E \cdot \hat{\underline{U}}_E - \underline{U}_W \cdot \hat{\underline{U}}_W}{2\Delta x} + \frac{\underline{U}_E - \underline{U}_W}{2\Delta x} \cdot \hat{\underline{U}}_P - \frac{\hat{\underline{U}}_E - \hat{\underline{U}}_W}{2\Delta x} \cdot \underline{U}_P \right] \underline{e}_x \\
&\approx \frac{\rho}{2} \left[-\nabla \left(\underline{U} \cdot \hat{\underline{U}} \right) + \nabla \underline{U} \cdot \hat{\underline{U}} - \nabla \hat{\underline{U}} \cdot \underline{U} \right] \\
&\approx -\rho \nabla \hat{\underline{U}} \cdot \underline{U}. \tag{B.14}
\end{aligned}$$

The final description, Eqn. (B.14), is equivalent to the formulation ADV2, cf. Eqn. (8.22). That is, the formulations ADV2 and ADV3 are equivalent for a Cartesian, equidistant grid. The equivalence (B.14) is not generally true on unstructured grids; however, the analysis reveals that ADV2 and ADV3 are closely related.

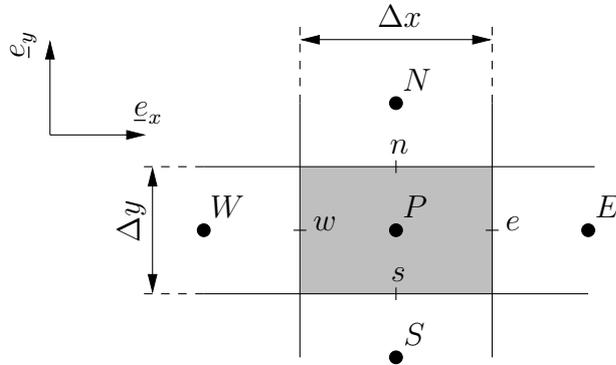


Figure B.1.: Cartesian grid with compass notation

C. Differential Geometry

It is assumed that a tangent surface perpendicular to the outer normal \underline{n} exists in every point \underline{x} on Γ . The tangent surface can be extended into the normal direction to form a narrow band enwrapping the boundary Γ , cf. Figure C.1. This object is often referred to as “tubular neighbourhood” of Γ . The band width has to be such that the boundary normals do not intersect. Let f and \underline{V} denote a function and a vector field existing in the tubular neighbourhood of Γ . Given that both f and \underline{V} are C^1 -steady, the following operations are the orthogonal projections of their Cartesian counterparts onto Γ [c.f. 57, 18, 145, 144, 137]

- the **tangential gradient**

$$\underline{\nabla}_\Gamma f = \underline{\underline{T}} \cdot \underline{\nabla} f \quad \text{or} \quad \frac{\partial f}{\partial x_i^\Gamma} = T_{ij} \frac{\partial f}{\partial x_j}; \quad (\text{C.1})$$

- the **tangential divergence**

$$\underline{\nabla}_\Gamma \cdot \underline{V} = \underline{\underline{T}} : \underline{\nabla} \underline{V} \quad \text{or} \quad \frac{\partial V_j}{\partial x_j^\Gamma} = T_{ij} \frac{\partial V_i}{\partial x_j}; \quad (\text{C.2})$$

- the **Laplace-Beltrami operator** obtained by successive application of (C.1) and (C.2) to a C^2 -continuous function f :

$$\underline{\nabla}_\Gamma^2 f = \underline{\nabla}_\Gamma \cdot (\underline{\nabla}_\Gamma f) \quad \text{or} \quad \frac{\partial^2 f}{\partial x_j^{\Gamma^2}} = \frac{\partial}{\partial x_j^\Gamma} \left(\frac{\partial f}{\partial x_j^\Gamma} \right); \quad (\text{C.3})$$

with the projection tensor

$$\underline{\underline{T}} = \underline{\underline{I}} - \underline{n}\underline{n} \quad \text{or} \quad T_{ij} = \delta_{ij} - n_i n_j. \quad (\text{C.4})$$

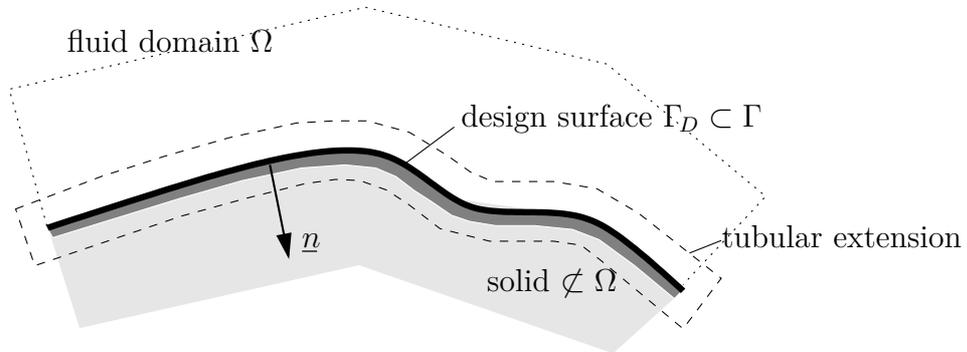


Figure C.1.: Tubular neighbourhood (dashed) of Γ

Mind that the projections are declared in 3D Cartesian coordinates. Based on the operations above, the local curvature of the boundary is defined as the surface divergence of the unit normal

$$\kappa = \underline{\nabla}_\Gamma \cdot \underline{n} \quad \text{on } \Gamma. \quad (\text{C.5})$$

When a perturbation $\underline{V} = \delta \underline{x}$ is applied to Γ_D and its neighbourhood, the corresponding variation of the unit normal lies in the tangential plane and can be calculated from

$$\delta(\underline{n}) = -(\underline{\nabla}_\Gamma \delta \underline{x}) \cdot \underline{n} \quad \text{or} \quad \delta(n_j) = -\frac{\partial}{\partial x_j^\Gamma} (\delta x_i) n_i \quad \text{on } \Gamma_D. \quad (\text{C.6})$$

D. Two-Step Adjoint Pressure-Correction Scheme

A two-fold pressure-correction scheme for the adjoint system is suggested that accounts for the advection term (ADV1) in the adjoint momentum equations. To set up the adjoint pressure-correction cycle, Eqn. (8.45) is artificially extended:

$$\hat{U}'_i \leftarrow \hat{U}'_i + \hat{U}''_i \quad \text{and} \quad \hat{p}' \leftarrow \hat{p}' + \hat{p}'' . \quad (\text{D.1})$$

Substituting (D.1) into Eqn. (8.45) immediately leads to

$$\overline{\hat{U}'_i} + \overline{\hat{U}''_i} = -\frac{\Delta\Omega}{\hat{A}_P} \left[\frac{\partial}{\partial x_i} \left(\overline{\hat{p}'} + \overline{\hat{p}''} \right) + \rho \frac{\partial U_j}{\partial x_i} \left(\overline{\hat{U}'_j} + \overline{\hat{U}''_j} \right) \right] , \quad (\text{D.2})$$

wherein the terms overlined once (twice) are treated in the first (second) pressure-correction cycle. Continuity is postulated for the adjoint velocity field corrected twice

$$0 = \frac{\partial}{\partial x_i} \left[\overline{\hat{U}'_i} + \overline{\hat{U}''_i} + \overline{\overline{\hat{U}'_i}} \right] . \quad (\text{D.3})$$

In the **first sweep** the adjoint advection terms depending on \hat{U}'_i are neglected by solving

$$0 = \frac{\partial}{\partial x_i} \left[\overline{\hat{U}'_i} + \overline{\hat{U}''_i} \right] \quad (\text{D.4})$$

with

$$\hat{U}'_i = -\frac{\Delta\Omega}{\hat{A}_P} \left[\frac{\partial \hat{p}'}{\partial x_i} \right] . \quad (\text{D.5})$$

It leads to the adjoint counter-part to the primal SIMPLE algorithm:

$$\frac{\partial}{\partial x_i} \left[\frac{\Delta\Omega}{\hat{A}_P} \left(\frac{\partial \hat{p}'}{\partial x_i} \right) \right] = \frac{\partial \overline{\hat{U}'_i}}{\partial x_i} . \quad (\text{D.6})$$

Subsequently the velocity and the fluxes are updated using (8.47).

The **second sweep** equates the remaining terms of Eqns. (D.2) and (D.3) overlined twice

$$0 = \frac{\partial}{\partial x_i} \left(\overline{\overline{\hat{U}'_i}} \right) . \quad (\text{D.7})$$

This time the advection contributions depending on \hat{U}''_i are neglected, treating the terms in Eqn. (8.45) from sweep 1 (\hat{U}'_i) in an explicit manner

$$\frac{\partial}{\partial x_i} \left[\frac{\Delta\Omega}{\hat{A}_P} \frac{\partial \hat{p}''}{\partial x_i} \right] = -\frac{\partial}{\partial x_i} \left[\rho \frac{\Delta\Omega}{\hat{A}_P} \frac{\partial U_j}{\partial x_i} \hat{U}'_j \right] . \quad (\text{D.8})$$

Accordingly, the velocity correction in the second loop is

$$\hat{U}_i'' = -\frac{\Delta\Omega}{\hat{A}_P} \left[\frac{\partial \hat{p}''}{\partial x_i} + \rho \hat{U}_j' \frac{\partial U_j}{\partial x_i} \right]. \quad (\text{D.9})$$

A comparison of expression (D.9) against the adjoint pressure equation (8.40a) reveals that the influence of the advection term is considered in sweep 2 of the modified SIMPLE algorithm.

Bibliography

- [1] W. K. Anderson and D. L. Bonhaus. Aerodynamic design on unstructured grids for turbulent flows. NASA Technical Memorandum 112867, 1997.
- [2] W. K. Anderson and D. L. Bonhaus. Airfoil design on unstructured grids for turbulent flows. *AIAA J.*, 37(2):185–191, 1999.
- [3] Autodiff.org. <http://www.autodiff.org>.
- [4] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc web page, 2009. <http://www.mcs.anl.gov/petsc>.
- [5] J. T. Batina. Unsteady Euler airfoil solutions using unstructured dynamic meshes. *AIAA J.*, 28(8):1381–1288, 1990.
- [6] T. Borrvall and J. Petersson. Topology optimization of fluids in Stokes flow. *Int. J. Numer. Meth. Fluids*, 41:77–107, 2003.
- [7] L. S. Caretto, A. D. Gosman, S. V. Patankar, and D. B. Spalding. Two calculation procedures for steady, three-dimensional flows with recirculation. In *Proceedings of the 3rd International Conference on Numerical Methods in Fluid Dynamics*, pages 60–68, 1973.
- [8] A. Carnarius, B. Günther, F. Thiele, D. Wachsmuth, F. Tröltzsch, and J. C. de los Reyes. Numerical study of the optimization of separation control. *AIAA-2007-58*, 2007.
- [9] A. Carnarius, F. Thiele, E. Özkaya, and N. R. Gauger. Adjoint approaches for optimal flow control. *AIAA-2010-5088*, 2010.
- [10] Y. Chang and S. Collis. Active control of turbulent channel flows based on large eddy simulation. *3rd ASME/JSME Joint Fluids Engineering Conference*, San Francisco, California, July 18–23, 1999.
- [11] M. Chevalier. *Feedback and Adjoint Based Control of Boundary Layer Flows*. PhD thesis, Department of Mechanics, Royal Institute of Technology, Stockholm, Sweden, 2004.
- [12] B. Christianson. Reverse accumulation of functions containing gradients. Numerical Optimisation Centre Technical Report 278, Theory Institute Argonne National Laboratory Illinois, School of Information Sciences, University of Hertfordshire, Hatfield, Herts, UK, 1993.

- [13] S. S. Collis and Y. Chang. Computer simulation of active control in complex turbulent flows. *Modeling and Simulation Based Engineering*, Volume I, p. 851–856, 1998.
- [14] G. Corliss, C. Faure, A. Griewank, L. Hascoët, and U. Naumann. *Automatic Differentiation of Algorithms – From Simulation to Optimization*. Springer, 2002.
- [15] A. Cunha, R. Teixeira, and L. Velho. Discrete scale spaces via heat equation. *SIBGRAPI 2001*, Florianópolis, Brazil, October 15–18, 2001.
- [16] P. Cusdin and J.-D. Müller. Deriving linear and adjoint codes for CFD using automatic differentiation. Draft submitted to *AIAA J.*, Nov. 2003.
- [17] K. Deb. *Multi Objective Optimization Using Evolutionary Algorithms*. Wiley-Interscience Series in Systems and Optimization. Wiley, 2001.
- [18] K. Dems and Z. Mróz. Variational approach by means of adjoint systems to structural optimization and sensitivity analysis, part II: Structure shape variation. *Int. J. Solids Struct.*, 20:527–552, 1984.
- [19] J. J. Dreyer and L. Martinelli. Hydrodynamic shape optimization of propulsor configurations using continuous adjoint approach. *AIAA-2001-2580*, 2001.
- [20] R. Duvigneau. Adaptive parameterization using free-form deformation for aerodynamic shape optimization. Technical Report RR-5949, Institut National de Recherche en Informatique et en Automatique, 2006.
- [21] R. P. Dwight. *Efficiency Improvements of RANS-based Analysis and Optimization using Implicit and Adjoint Methods on Unstructured Grids*. PhD thesis, Faculty of Engineering and Physical Sciences, University of Manchester, 2006.
- [22] R. P. Dwight and J. Brezillon. Effect of approximations of the discrete adjoint on gradient-based optimization. *AIAA J.*, 44(12):3022–3071, 2006.
- [23] R. P. Dwight and J. Brezillon. Efficient and robust algorithms for solution of the adjoint compressible Navier–Stokes equations with applications. *Int. J. Numer. Meth. Fluids*, 60(4):365–389, 2008.
- [24] R. P. Dwight and Z.-H. Han. Efficient uncertainty quantification using gradient-enhanced kriging. *AIAA-2009-2276*, 2009.
- [25] O. Enoksson and P. Weinerfeld. Numerical methods for aerodynamic optimization. *8th Int. Symp. on CFD*, Bremen, Germany, September 5–10, 1999.
- [26] K. Eppler, S. Schmidt, V. Schulz, and C. Ilic. Preconditioning the pressure tracking in fluid dynamics by shape Hessian information. *J. Optimiz. Theory App.*, 141(3): 513–531, June 2009.

- [27] C. Farhat, C. Degand, B. Koobus, and M. Lesoinne. Torsional spring for two-dimensional dynamic unstructured fluid meshes. *Comput. Method Appl. M.*, 163: 231–245, 1998.
- [28] J. Farmer, L. Martinelli, and A. Jameson. A fast multigrid method for solving incompressible hydrodynamic problems with free surfaces. *AIAA-1993-0767*, 1993.
- [29] M. Felsberg and G. Sommer. Scale adaptive filtering derived from the Laplace equation. In *Pattern Recognition*, pages 124–131. Springer, 2001.
- [30] J. H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics*. Springer, 3rd edition, 2002.
- [31] M. P. I. Forum. MPI: A message-passing interface standard, version 2.1. High Performance Computing Center Stuttgart (HLRS), 2008.
- [32] J. B. Freund. Adjoint-based optimization for understanding and suppressing jet noise. *IUTAM Symp. on Computational Aero-Acoustics for Aircraft Noise Prediction*, University of Southampton, UK, 29-31 March, 2010.
- [33] T. Gerhold and J. Neumann. The parallel mesh deformation of the DLR TAU-code. In C. Tropea, S. Jakirlic, H.-J. Heinemann, R. Henke, and H. Hönliger, editors, *New Res. in Num. and Exp. Fluid Mech. VI, NNFM 96*, pp. 162–169. 2007.
- [34] A. Gersborg-Hansen. *Topology Optimization of Flow Problems*. PhD thesis, Department of Mathematics, Technical University of Denmark, Apr. 2007.
- [35] A. Gersborg-Hansen, O. Sigmund, and R. B. Haber. Topology optimization of channel flow problems. *Struct. Multidisc. Optim*, 30:181–192, 2005.
- [36] I. Gherman and V. Schulz. Preconditioning of one-shot pseudo-timestepping methods for shape optimization. *PAMM Proc. Appl. Math. Mech.*, 5(1):741–742, 2005.
- [37] R. Giering and T. Kaminski. Recipes for adjoint code construction. *ACM T. Math. Software*, 24(4):437–474, 1998.
- [38] M. Giles. Adjoint equations in CFD: Duality, boundary conditions and solution behaviour. *AIAA-1997-1850*, 1997.
- [39] M. B. Giles and N. A. Pierce. An introduction to the adjoint approach to design. *Flow Turbul. Combust.*, 65:393–415, 2001.
- [40] M. B. Giles and N. A. Pierce. Analytic adjoint solutions for the quasi-one-dimensional Euler equations. *J. Fluid Mech.*, 426:327–345, 2001.
- [41] M. B. Giles, J.-D. Müller, and M. C. Duta. Adjoint code developments using the exact discrete approach. *AIAA-2001-2596*, 2001.

- [42] M. B. Giles, M. C. Duta, J.-D. Müller, and N. A. Pierce. Algorithm developments for discrete adjoint methods. *AIAA J.*, 41(2):198–205, 2003.
- [43] M. B. Giles, D. Ghate, and M. C. Duta. Using automatic differentiation for adjoint CFD code development. In *Post SAROD Workshop, Bangalore, India, December 12-13*, 2005.
- [44] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.
- [45] A. Griewank. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Number 19 in Frontiers in Appl. Math. SIAM, 2000.
- [46] A. Griewank and A. Walther. Revolve: An implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Trans. Math. Software*, 26:200, 2000.
- [47] J. K. Guest and J. H. Prévost. Topology optimization of creeping fluid flows a Darcy–Stokes finite element. *Int. J. Numer. Meth. Eng.*, 66:461–484, 2006.
- [48] B. Günther, A. Carnarius, and F. Thiele. Numerical investigation of active flow control applied to an airfoil with a camber flap. In R. King, editor, *Active Flow Control II*, volume 108 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 45–61. Springer, 2010.
- [49] A. Harten. High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.*, 49(3):357–393, 1983.
- [50] R. Hartmann. Derivation of an adjoint consistent discontinuous Galerkin discretization of the compressible Euler equations. *BAIL 2006*, Göttingen, Germany, July 24–28, 2006.
- [51] R. Hartmann and P. Houston. Error estimation and adaptive mesh refinement for aerodynamic flows. *36th CFD/ADIGMA course on hp-adaptive and hp-multigrid methods*, Rhode Saint Genese, Belgium, October 26–30, 2009.
- [52] R. Hartmann, J. Held, and T. Leicht. Adjoint-based error estimation and adaptive mesh refinement for the RANS and k- ω turbulence model equations. *J. Comput. Phys.*, 230(11):4268–4284, 2010.
- [53] E. J. Haug, K. K. Choi, and V. Komkov. *Design Sensitivity Analysis of Structural Systems*, volume 177 of *Mathematics in Science and Engineering*. Academic Press, 1986.
- [54] S. B. Hazra and A. Jameson. One-shot pseudo-time method for aerodynamic shape optimization using the Navier–Stokes equations. *AIAA-2007-1470*, 2007.

- [55] D. S. Henningson and A. Hanifi. The application of optimal control to boundary layer flow. *IUTAM Symp. on One Hundred Years of Boundary Layer Research*, Göttingen, Germany, August 12–14, 2004.
- [56] R. M. Hicks and P. A. Henne. Wing design by numerical optimization. *J. Aircraft*, 15(7):407–412, 1978.
- [57] R. Hill. Aspects of invariance in solid mechanics. *Adv. Appl. Mech.*, 18:1–75, 1979.
- [58] T. Hino. Shape optimization of practical ship hull forms using Navier–Stokes analysis. *7th Conference on Numerical Ship Hydrodynamics*, Nantes, France, July 19–22, 1999.
- [59] M. Hinze. Optimal and instantaneous control of the instationary Navier–Stokes equations. Habilitationsschrift, July 2002.
- [60] J. Huan and V. Modi. Optimum design of minimum drag bodies in incompressible laminar flow using a control theory approach. *J. Inverse Problems in Sci. and Eng.*, 1(1):1–25, 1994.
- [61] A. Jameson. Aerodynamic design via control theory. *J. Sci. Comput.*, 3(3):233–260, Sept. 1988.
- [62] A. Jameson. Optimum aerodynamic design using CFD and control theory. *AIAA-1995-1729-CP*, 1995.
- [63] A. Jameson. Advances in aerodynamic shape optimization. In C. Groth and D. W. Zingg, editors, *Proceedings of the Third International Conference on Computational Fluid Dynamics, ICCFD3, Toronto, 12–16 July 2004*, 2004.
- [64] A. Jameson and S. Kim. Reduction of the adjoint gradient formula in the continuous limit. *AIAA-2003-0040*, 2003.
- [65] A. Jameson and J. C. Vassberg. Studies of alternative numerical optimization methods applied to the Brachistochrone problem. *Int. J. Comput. Fluid Dyn.*, 9(3):281–296, 2000.
- [66] A. Jameson, L. Martinelli, and N. A. Pierce. Optimum aerodynamic design using the Navier–Stokes equations. *Theor. Comput. Fluid Dyn.*, 10:213–237, 1998.
- [67] A. Jameson, Sriram, and L. Martinelli. A continuous adjoint method for unstructured grids. *AIAA-2003-3955*, 2003.
- [68] S. Jameson, A. Jameson, and M. G. Gerritsen. Numerical analysis and design of upwind sails. *AIAA-2003-3531*, 2003.
- [69] A. Jaworski and J.-D. Müller. *Advances in Automatic Differentiation*, volume 64 of *Lecture Notes in Computational Science and Engineering*, chapter Toward Modular Multigrid Design Optimisation, pages 281–291. Springer, 2008.

- [70] W. P. Jones and B. E. Launder. The prediction of laminarization with a two-equation model of turbulence. *Int. J. Heat Mass Transfer*, 15:301–315, 1972.
- [71] K. A. D. Jong. *Evolutionary Computation*. MIT Press, 2006.
- [72] G. Karypis. ParMETIS web page. <http://glaros.dtc.umn.edu/gkhome/views/metis>.
- [73] C. S. Kim, C. Kim, and O. H. Rho. Sensitivity analysis for the Navier–Stokes equations with two-equation turbulence models. *AIAA J.*, 39(5):838–845, 2001.
- [74] H.-J. Kim, C. Kim, O.-H. Rho, and K. D. Lee. Aerodynamic sensitivity analysis for Navier–Stokes equations. *J. KSIAM*, 3(2):161–171, 1999.
- [75] J. Kim, D. J. Bodony, and J. B. Freund. A high-order, overset-mesh algorithm for adjoint-based optimization for aeroacoustics control. *AIAA-2010-3818*, 2010.
- [76] J.-H. Kim and K. Nakahashi. Unstructured adjoint method for Navier–Stokes equations. *JSME Int. J. Ser. B*, 48(2):202–207, 2005.
- [77] S. Kim and H.-S. Chung. Multiobjective optimization using adjoint gradient enhanced approximation models for genetic algorithms. *ICCSA 2006*, Glasgow, UK, May 8–11, 2006.
- [78] S. Kim, J. J. Alonso, and A. Jameson. Two-dimensional high-lift aerodynamic optimization using the continuous adjoint method. *AIAA-2000-4741*, 2000.
- [79] S. Kim, K. Hosseini†, K. Leoviriyakit, and A. Jameson. Enhancement of adjoint design methods via optimization of adjoint parameters. *AIAA-2005-0448*, 2005.
- [80] J. Kröger, A. Stück, and T. Rung. Adjoint aftship re-design for wake optimisation under the influence of active propulsion. *MARINE 2011*, Lisbon, Portugal, 28–30 September, 2011.
- [81] G. Kuruvila, S. Taasan, and M. D. Salas. Airfoil optimization by the one-shot method. *AIAA-92-0025*, 1992.
- [82] E. Laporte and P. L. Tallec. *Numerical Methods in Sensitivity Analysis and Shape Optimization*. Birkhäuser, 2003.
- [83] F. S. Lien and M. A. Leschziner. Computational modelling of 3d turbulent flow in s-diffuser and transition ducts. In *Engineering Turbulence Modelling and Experiments 2*, 1993.
- [84] F. S. Lien and M. A. Leschziner. A general non-orthogonal collocated finite volume algorithm for turbulent flow at all speeds incorporating second-moment turbulence-transport closure, part 1: Computational implementation. *Comput. Meth. Appl. Mech. Eng.*, 114:123–148, 1994.

- [85] F. S. Lien and M. A. Leschziner. Upstream monotonic interpolation for scalar transport with application to complex turbulent flows. *Int. J. Num. Meth. Fluids*, 19:527–548, 1994.
- [86] T. Lindeberg. *Discrete Scale-Space Theory and the Scale-Space Primal Sketch*. PhD thesis, Computational Vision and Active Perception Laboratory (CVAP), Royal Institute of Technology, Stockholm, Sweden, 1991.
- [87] T. Lindeberg. Scale-space. In B. Wah, editor, *Encyclopedia of Computer Science and Engineering*, volume IV, pages 2495–2504. Wiley, 2008.
- [88] T. Lindeberg and B. ter Haar Romeny. *Geometry-Driven Diffusion in Computer Vision*, volume 1 of *Computational Imaging and Vision*, chapter Linear Scale-Space 1 & 2, pages 1–77. Kluwer, 1994.
- [89] R. Löhner, O. Soto, and C. Yang. An adjoint-based design methodology for CFD optimization problems. *AIAA-2003-0299*, 2003.
- [90] L. Martinelli and G. Cowles. An adjoint method for the incompressible Reynolds averaged Navier–Stokes using artificial compressibility. In *Aerospace Conference Proceedings*, volume 2, pages 89–110, 2000.
- [91] L. Martinelli and A. Jameson. An adjoint method for design optimization of ship hulls. *9th International Conference on Numerical Ship Hydrodynamics*, Ann Arbor, Michigan, USA, August 5–8, 2007.
- [92] J. R. R. A. Martins, P. Sturdza, and J. J. Alonso. The complex-step derivative approximation. *ACM T. Math. Software*, 29(3):245–262, 2003.
- [93] B. Mohammadi. A new optimal shape design procedure for inviscid and viscous turbulent flows. *Int. J. Numer. Meth. Fluids*, 25(2):183–203, 1997.
- [94] B. Mohammadi. Mesh adaption and automatic differentiation for optimal shape design. *Int. J. Comput. Fluid Dyn.*, 10(3):199–211, 1998.
- [95] B. Mohammadi and O. Pironneau. Mesh adaption and automatic differentiation in a CAD-free framework for optimal shape design. *Int. J. Numer. Meth. Fluids*, 30:127–136, 1999.
- [96] A. L. Moigne. *A discrete Adjoint Approach to Aerodynamic Shape Optimization of Blended Wing-Body Configurations*. PhD thesis, Cranfield University, December 2002.
- [97] J.-D. Müller and P. Cusdin. On the performance of discrete adjoint CFD codes using automatic differentiation. *Int. J. Numer. Meth. Fluids*, 47(8–9):939–945, 2005.
- [98] M. Murayama, K. Nakahashi, and K. Matsushima. Unstructured dynamic mesh for large movement and deformation. *AIAA-2002-0122*, 2002.

- [99] S. Nadarajah and A. Jameson. Studies of the continuous and discrete adjoint approaches to viscous automatic aerodynamic shape optimization. *AIAA-2001-2530*, 2001.
- [100] S. Nadarajah, M. S. McMullen, and A. Jameson. Non-linear frequency domain based optimum shape design for unsteady three-dimensional flows. *AIAA-2006-1052*, 2006.
- [101] S. K. Nadarajah. *The Discrete Adjoint Approach to Aerodynamic Shape Optimization*. PhD thesis, Stanford University, Jan. 2003.
- [102] S. K. Nadarajah and A. Jameson. A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. *AIAA-2000-0667*, 2000.
- [103] M. Nemec and D. W. Zingg. Towards efficient aerodynamic shape optimization based on the Navier–Stokes equations. *AIAA-2001-2532*, 2001.
- [104] J. C. Newman III, A. C. Taylor III, R. W. Barnwell, P. A. Newman, and G. J.-W. Hou. Overview of sensitivity analysis and shape optimization for complex aerodynamic configurations. *J. Aircraft*, 36(1):87–96, 1999.
- [105] J. C. Newman III, R. Pankajakshan, D. L. Whitfield, and L. K. Taylor. Computational design optimization using RANS. *24th Symp. on Naval Hydrodynamics*, Fukuoka, Japan, 8–13 July, 2002.
- [106] E. J. Nielsen. *Aerodynamic Design Sensitivities on an Unstructured Mesh using the Navier–Stokes Equations and a Discrete Adjoint Formulation*. PhD thesis, Faculty of Virginia Polytechnic Institute and State University, Blacksburg, Virginia, Dec. 1998.
- [107] E. J. Nielsen and W. K. Anderson. Recent improvements in aerodynamic design optimization on unstructured meshes. *AIAA-2001-0596*, 2001.
- [108] E. J. Nielsen and M. A. Park. Using an adjoint approach to eliminate mesh sensitivities. *AIAA-2005-0491*, 2005.
- [109] E. J. Nielsen, L. B. James, M. A. Park, and D. L. Darmofal. An implicit, exact dual adjoint solution method for turbulent flows on unstructured grids. *Computers & Fluids*, 33:1131–1155, 2004.
- [110] C. Othmer. A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows. *Int. J. Numer. Meth. Fluids*, 58(8):861–877, 2008.
- [111] C. Othmer, T. Kaminski, and R. Giering. Computation of topological sensitivities in fluid dynamics: Cost function versatility. *ECCOMAS CFD 2006*, Egmond aan Zee, The Netherlands, September 5–8, 2006.

- [112] E. Özkaya and N. R. Gauger. Automatic transition from simulation to one-shot shape optimization with Navier–Stokes equations. *GAMM Reports*, 33(2):133–147, 2010.
- [113] D. I. Papadimitriou, A. S. Zymaris, and K. C. Giannakoglou. Discrete and continuous adjoint formulations for turbomachinery applications. *EUROGEN 2005*, Munich, September 12–14, 2005.
- [114] S. V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Taylor and Francis, 1980.
- [115] J. Peraire, J. Peiro, and K. Morgan. A 3d finite-element multigrid solver for the Euler equations. *AIAA-92-0449*, 1992.
- [116] J. E. V. Peter and R. P. Dwight. Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches. *Computers & Fluids*, 39(3):373–391, 2010.
- [117] C.-T. Pham. *Linéarisation du flux visqueux des équations de Navier–Stokes et de modèles de turbulence pour l’optimisation aérodynamique en turbomachines*. PhD thesis, École Nationale Supérieure d’Arts et Métiers, Centre de Paris, 2006.
- [118] O. Pironneau. On optimum profiles in Stokes flow. *J. Fluid Mech.*, 59:117–128, 1973.
- [119] O. Pironneau. On optimum design in fluid mechanics. *J. Fluid Mech.*, 64(1):97–110, 1974.
- [120] O. Pironneau. *Optimal shape design for elliptic systems*. Springer Series in Computational Physics. Springer, 1984.
- [121] S. A. Ragab. An adjoint formulation for shape optimization in free-surface potential flow. *J. Ship Res.*, 45(4):269–278, December 2001.
- [122] S. A. Ragab. Shape optimization in free surface potential flow using an adjoint formulation. Technical report, Naval Surface Warfare Center, Carderock Division West Bethesda, Hydrodynamics Directorate, Aug. 2003.
- [123] H. C. Raven and M. Hoekstra. A practical system for hydrodynamic optimization of ship hull forms. *VNSI Innovatiedag*, Wageningen, The Netherlands, 2003.
- [124] F. Renac. Aerodynamic sensitivity analysis of the RANS equations via the recursive projection method. *AIAA-2010-4364*, 2010.
- [125] F. Renac, C.-T. Pham, and J. Peter. Sensitivity analysis for the RANS equations coupled with linearized turbulence model. *AIAA-2007-3948*, 2007.
- [126] J. Reuther, A. Jameson, J. Farmer, L. Martinelli, and D. Saunders. Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation. *AIAA-96-0094*, 1996.

- [127] C. M. Rhie and W. L. Chow. A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation. *AIAA J.*, 21:1525–1532, 1983.
- [128] T. T. Robinson, C. G. Armstrong, H. S. Chua, C. Othmer, and T. Grahls. Sensitivity-based optimization of parameterised CAD geometries. *8th World Congress on Structural and Multidisciplinary Optimization*, Lisbon, Portugal, June 2009.
- [129] P. L. Roe. Characteristic-based schemes for the Euler equations. *Ann. Rev. Fluid Mech.*, 18:337–365, 1986.
- [130] A. Ronzheimer. Shape parametrization using freeform deformation. In *MEGAFLOW - Numerical Flow Simulation for Aircraft Design, Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, number 89, pages 211–222. 2005.
- [131] T. Rung. Vorlesungsmanuskript: Statistische Turbulenzmodellierung. Hermann-Föttinger-Institut für Strömungsmechanik, Technische Universität Berlin, 2001.
- [132] T. Rung. Vorlesungsmanuskript CFD II, Kapitel 9–11. Institut für Fluidodynamik und Schiffstheorie, Technische Universität Hamburg-Harburg, 2008.
- [133] T. Rung. Vorlesungsmanuskript CFD II, Kapitel 5. Institut für Fluidodynamik und Schiffstheorie, Technische Universität Hamburg-Harburg, 2008.
- [134] M. Sakamoto and M. Kawahara. Shape optimization of a body located in incompressible flow using adjoint method and partial control algorithm. *Int. J. Numer. Meth. Fluids*, 66:n/a. 2011. October 2010 (published online).
- [135] A. Saltelli, S. Tarantola, F. Campolongo, and M. Ratto. *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*. Wiley, 2004.
- [136] M. Schatz. Konvektionsschemata in FAN/ELAN2/ELAN3, 2003. URL <http://www.cfd.tu-berlin.de/~schatz/PUBLICATIONS/tvd.pdf>.
- [137] S. Schmidt. *Efficient Large Scale Aerodynamic Design Based on Shape Calculus*. PhD thesis, Fachbereich IV der Universität Trier, 2010.
- [138] S. Schmidt, C. Ilic, N. Gauger, and V. Schulz. Shape gradients and their smoothness for practical aerodynamic design optimization. Preprint-Number SPP1253-10-03, 2008. URL <http://www.am.uni-erlangen.de/home/spp1253/wiki/index.php/Preprints>.
- [139] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. *SIGGRAPH 1986*, Dallas, Texas, USA, August 18–22, 1986.
- [140] H. Söding. Practical hull shape design for reduced resistance. *HYDRONAV 2001*, Szczecin, Poland, September 27-29, 2001.
- [141] H. Söding. Hull shape design for reduced resistance. *Ship Technol. Res.*, 48:135–144, 2001.

- [142] H. Söding. Resistance decrease by computer-aided hull shape improvements. *HYPER 2001*, Hamburg, May 2–5, 2001.
- [143] B. Soemarwoto. The variational method for aerodynamic optimization using the Navier–Stokes equations. Report 97-71, ICASE, December 1997.
- [144] B. I. Soemarwoto. *Multi-Point Aerodynamic Design by Optimization*. PhD thesis, Delft University of Technology, 1996.
- [145] J. Sokolowski and J.-P. Zolésio. *Introduction to Shape Optimization: Shape Sensitivity Analysis*. Springer, 1992.
- [146] O. Soto and R. Löhner. CFD optimization using an incomplete-gradient adjoint approach. *AIAA-2000-0666*, 2000.
- [147] O. Soto and R. Löhner. General methodologies for incompressible flow design problems. *AIAA-2001-1061*, 2001.
- [148] O. Soto and R. Löhner. A mixed adjoint formulation for incompressible RANS problems. *AIAA-2002-0451*, 2002.
- [149] O. Soto and R. Löhner. On the computation of flow sensitivities from boundary integrals. *AIAA-2004-0112*, 2004.
- [150] O. Soto, R. Löhner, and C. Yang. A stabilized pseudo-shell approach for surface parametrization in CFD design problems. *Commun. Numer. Meth. Engng*, 18:251–258, 2002.
- [151] O. Soto, R. Löhner, and C. Yang. An adjoint-based design methodology for CFD problems. *Int. J. Numer. Methods Heat Fluid Flow*, 14(6):734–759, 2004.
- [152] A. Stück and T. Rung. Adjoint RANS with filtered shape derivatives for hydrodynamic optimisation. *Computers & Fluids*, 47(1):22–32, 2011.
- [153] A. Stück, F. F. Camelli, and R. Löhner. Adjoint-based design of shock mitigation devices. *Int. J. Numer. Meth. Fluids*, 64:443–472, 2010.
- [154] A. Stück, F. F. Camelli, and R. Löhner. Adjoint-based design of passive and active shock mitigation devices. *AIAA-2010-1430*, 2010.
- [155] A. Stück, J. Kröger, and T. Rung. Adjoint RANS for aft-ship design. *ECCOMAS CFD 2010*, Lisbon, Portugal, June 14–17, 2010.
- [156] A. Stück, J. Kröger, and T. Rung. Adjoint-based hull design for wake optimisation. *Ship Technol. Res.*, 58(1):34–44, 2011.
- [157] P. K. Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J. Numer. Anal.*, 21(5):995–1011, 1984.

- [158] Y. Tahara, F. Stern, and Y. Himeno. Computational fluid dynamics-based optimization of a surface combatant. *J. Ship Res.*, 28(4):273–287, 2004.
- [159] J. A. Templeton, M. Wang, and P. Moin. A predictive wall model for large-eddy simulation based on optimal control techniques. *Phys. Fluids*, 20, 2008.
- [160] B. van Leer. Towards the ultimate conservative difference scheme II. monotonicity and conservation combined in a second order scheme. *J. Comput. Phys.*, 14:361–370., 1974.
- [161] B. van Leer. Towards the ultimate conservative difference scheme. V – a second-order sequel to Godunov’s method. *J. Comput. Phys.*, 32:101–136, 1979.
- [162] V. Venkatakrishnan and D. J. Mavriplis. Implicit method for the computation of unsteady flows on unstructured grids. *J. Comput. Phys.*, 127:380–397, 1996.
- [163] A. Walther and A. Griewank. Advantages of binomial checkpointing for memory-reduced adjoint calculations. Preprint, DFG-Forschungszentrum Matheon, 2004.
- [164] Q. Wang and P. Moin. Minimal repetition dynamic checkpointing algorithm for unsteady adjoint calculation. Center for Turbulence Research, Annual Research Briefs 2008-55, 2008.
- [165] A. Wick. *Ein numerisches Verfahren zur Strömungssimulation in zeitveränderlichen Gebieten mit integriertem Modul zur Gitternachführung*. PhD thesis, Technische Universität Berlin, 2003.
- [166] D. C. Wilcox. *Turbulence Modeling for CFD*. DCW industries, Inc., 2nd edition, Dec. 2002.
- [167] J. A. S. Witteveen and H. Bijl. Explicit mesh deformation using inverse distance weighting interpolation. *AIAA-2009-3996*, 2009.
- [168] K. S. Won and T. Ray. Performance of kriging and cokriging based surrogate models within the unified framework for surrogate assisted optimization. *CEC 2004*, Portland, Oregon, US, June 19–23, 2004.
- [169] T. Zervogiannis, P. I. K. D. Liakopoulos, D. I. Papadimitriou, and K. C. Giannakoglou. A grid enrichment and movement strategy for a posteriori error analysis in viscous flows. *7th European Turbomachinery Conference*, Athens, Greece, March 5–8, 2007.
- [170] A. S. Zymaris, D. I. Papadimitriou, K. C. Giannakoglou, and C. Othmer. Continuous adjoint approach to the Spalart–Allmaras turbulence model for incompressible flows. *Computers & Fluids*, 38(8):1528–1538, Sept. 2009.
- [171] A. S. Zymaris, D. I. Papadimitriou, K. C. Giannakoglou, and C. Othmer. Adjoint wall functions: A new concept for use in aerodynamic shape optimization. *J. Comput. Phys.*, 229(13):5228–5245, 2010.