TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Numerische Mechanik

An Efficient Partitioned Coupling Scheme for Tire Hydroplaning Analysis

Julien Gillard

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Prof. dr. ir. Daniel J. Rixen

Prüfer der Dissertation:

1. Prof. Dr.-Ing. Wolfgang A. Wall
2. Prof. Dr. rer. nat. habil. Hans-Joachim Bungartz

Die Dissertation wurde am 23.05.2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 30.01.2019 angenommen.

Abstract

The tire hydroplaning phenomenon represents one of the main safety hazards for road users driving under wet conditions. The improvement of the wet performance is therefore of prime importance for the automotive industry. In the development loop of new tire designs, the cost and the limited number of results provided by experimental measurements call for the use of numerical simulation. In this context, Goodyear as one of the leading tire manufacturers initiated the present research, which aims at improving the simulation of the tire-water interaction to enlarge the understanding of the underlying physics and optimize the hydroplaning performance of tires.

The current application is very challenging in the field of computational fluid-structure interaction (FSI) due to the complexity of each sub-domain and their coupling. A partitioned procedure is used in this work to couple two single-field black-box solvers specifically chosen for their abilities to compute accurately and efficiently their respective physical sub-domain. Such a partitioned coupling procedures is particularly demanding in the case of the tire hydroplaning problem as it suffers from the so-called artificial added-mass effect that can introduce strong instabilities. More specifically, a loosely-coupled approach has been chosen to fulfill the time efficiency and robustness requirements imposed by the industrial character of this research. Although strongly-coupled methods are commonly employed for dealing with large added-mass effects, it is shown in this work that a loosely-coupled procedure can successfully compute the hydroplaning problem for real full-scale tire designs.

Replacing a basic coupling algorithm embedded in one of the single-field solvers, a new coupling shell is developed as a standalone flexible development platform, that is used as a workbench to evaluate alternative advanced coupling strategies. It essentially allows to manage the execution of the solid and fluid solvers and to extract coupling data required for developing the proposed improvements.

Depending on the application, several approaches are proposed to obtain the most time efficient yet accurate solution path. The proposed methods all rely on a rigorous control of the energy artificially produced at the interface by the partitioning process. On the one hand, an interface energy-based error is dynamically evaluated and used to choose the largest coupling time step size depending on the accuracy requirements in each time station. To the best of the author's knowledge, it is the first time that such an interface energy-based criterion is used to take correcting actions for improving the coupling procedure, in particular optimize the coupling time step size along the simulation. On the other hand, a predictor on the solid displacements at the interface is employed to reduce the partitioning error, which in turn allows for the use of larger FSI time steps.

The novel coupling strategy and proposed developments are validated with use of a numerical model considered as a standard benchmark for FSI problems. The hydroplaning application is then computed for a Grosch wheel, a small size simplified tire model, as well as for a real tire construction. It is shown that for weakly-coupled problems such as the present Grosch wheel example, the use of a solid predictor is beneficial as it reduces the overall simulation time by up to 50 %. As for the real tire design prone to large added-mass instabilities, the application of the adaptive time stepping procedure substantially reduces the simulation time leading to a gain of more than 30 %. Ultimately, a robust loosely-coupled procedure is used successfully for efficiently solving the hydroplaning problem for real tire designs.

Zusammenfassung

Das Reifen-Aquaplaning-Phänomen stellt eine der größten Sicherheitsrisiken für Verkehrsteilnehmer bei nassen Bedingungen dar. Die Verbesserung der Leistung bei Nässe ist daher für die Automobilindustrie von größter Bedeutung. Die hohen Kosten und begrenzte Anzahl von Experimenten erfordern den Einsatz numerischer Simulationen in der Entwicklung neuer Reifenkonstruktionen. In diesem Zusammenhang hat Goodyear, als einer der führenden Reifenhersteller, das vorliegende Forschungsprojekt mit dem Ziel die Simulation der Reifen-Wasser-Interaktion zu verbessern initiiert, um das Verständnis der zugrunde liegenden Physik zu erweitern und die Aquaplaning-Leistung von Reifen zu optimieren.

Die vorliegende Anwendung im Feld der numerischen Fluid-Struktur Interaktion (FSI) ist aufgrund der Komplexität jedes Teilgebiets sowie deren Kopplung sehr anspruchsvoll. Zur Lösung dieses numerischen Problems wird in dieser Arbeit ein partitioniertes Verfahren verwendet, um zwei Einzelfeld-Blackbox-Solver zu koppeln, die speziell aufgrund ihrer Eignung das jeweilige physikalische Teilgebiet genau und effizient zu berechnen ausgewählt wurden. Derartige partitionierte Lösungsverfahren sind bei Reifen-Aquaplaning-Problemen sehr anspruchsvoll, da sie am sogenannten “artificial added-mass”-Effekt leiden, der zu starken Instabilitäten in dem gekoppelten System führen kann. Durch den industriellen Charakter der Problemstellung wird ein schwach gekoppeltes (loosely-coupled) Verfahren gewählt, um die Effizienz- und Robustheitsanforderungen zu erfüllen. Obwohl stark gekoppelte (strongly-coupled) Methoden häufig für den Umgang mit großen “artificial added-mass”-Effekten verwendet werden, wird in dieser Arbeit gezeigt, dass das Aquaplaning-Problem für reale Vollskala Reifenkonstruktionen mit schwach gekoppelten Einzelfeldlösern erfolgreich berechnet werden kann.

Eine neue eigenständige und flexible Entwicklungsumgebung (coupling shell) ersetzt den bestehenden Kopplungsalgorithmus, der in einen der Löser eingebaut ist. Diese ermöglicht die Ausführung der Solid- und Fluidlöser zu verwalten sowie die Kopplungsdaten zu extrahieren, die für die Entwicklung und Bewertung neuartiger Kopplungsstrategien benötigt werden.

Zum Erhalt des effizientesten und genauesten Lösungspfades werden mehrere problemspezifische Ansätze entwickelt. Die vorgeschlagenen Methoden beruhen auf einer strikten Kontrolle der Energie, die durch den Partitionierungsprozess künstlich am Kopplungsrand erzeugt wird. Zum einen wird ein dynamisch ausgewertetes Fehlermaß der Interfaceenergie verwendet, um abhängig von den Genauigkeitsanforderungen größtmögliche Zeitschritte wählen zu können. Nach bestem Wissen ist es das erste Mal, dass ein solches Interfaceenergie Kriterium verwendet wird, um korrigierend in den Kopplungsvorgang einzugreifen, insbesondere um die Kopplungszeitschrittgröße während der Simulation zu optimieren. Zum anderen wird ein Prädiktor für die Strukturverschiebungen am Kopplungsrand verwendet, der den Partitionierungsfehler reduziert und wiederum größere FSI-Zeitschritte zu ermöglichen.

Die neue Kopplungsstrategie und die vorgeschlagenen Entwicklungen werden mit Hilfe eines Standardbenchmarks für FSI-Probleme validiert. Als Aquaplaninganwendung werden sowohl ein sogenanntes Grosch-Rad, ein kleines vereinfachtes Reifenmodell, als auch eine reale Reifenkonstruktion berechnet. Es zeigt sich, dass für schwach gekoppelte Probleme, wie das vorliegende Grosch-Rad Beispiel, die Verwendung des Solid-Prädiktors vorteilhaft ist und zu einer

Reduktion der gesamten Simulationszeit von bis zu 50 % führt. Für die Simulation der realen Reifenkonstruktion, die zu “added-mass”-Instabilitäten neigt, reduziert die Anwendung des dynamischen adaptiven FSI-Zeitschrittverfahrens die Simulationszeit um 30 %. Zusammenfassend lässt sich sagen, dass ein schwach gekoppeltes Verfahren erfolgreich zur effizienten Berechnung des Aquaplaningverhaltens realer Reifenkonstruktionen unter Einhaltung eines benutzerdefinierten Genauigkeitsniveaus eingesetzt werden kann.

Acknowledgments

First and foremost, I would like to thank Goodyear for giving me the opportunity to carry out this research. In particular, I would like to thank my industrial supervisor, Dr. Vincent Decouvreur, who has guided me during these last few years. Without you, this adventure would not have ended as happily as it did. A big thank you to Dr. Didier Quoirin leading the group I was part of at Goodyear for his continuous support and good advice.

I would also like to thank Prof. Wolfgang A. Wall for accepting to take this project on and be my academic supervisor. I will keep a very fond memory of my time at the Institute for Computational Mechanics at the Technical University of Munich, both from a professional and personal standpoint. I am proud to say that many of my colleagues have become friends that I will keep for many years to come. To name just a few: Fabi, Jonas, Charles, Maggie, Pfallus, Anna, Alex and Keijo. I would also like to thank Romain Pennec and GSDM for all their help towards the end of the project.

I am lucky to have a family who has always believed in me and allowed me to pursue any ideas I came up with, including the crazy one of doing a Ph.D.! I cannot thank you enough for being there in the good and hard moments. Papa, maman, Catherine and Manu, thank you so much for everything.

Last but not least, I would like to thank Marwi, Max, and Pit for being there for me over the last decade(s), even though it was mostly on the other end of a phone. Then, of course Kirsten. You were there to support me in the last moments of this journey. Thank you for the great English classes, and the close review of the content of my presentation.

This project was supported by Fonds National de la Recherche (FNR) in Luxembourg under Grant 4845982.

Munich, May 2019 Julien Gillard

Contents

Nomenclature

Representation of scalars, tensors & discrete quantities

x, X	Scalar variable
$\boldsymbol{x}$	Vector
$\boldsymbol{X}$	Second-order tensor
$\mathcal{X}$	Higher-order tensor
x, X	Discrete scalar
$\boldsymbol{\mathsf{x}}$	Discrete vector
$\boldsymbol{\mathsf{X}}$	Discrete matrix

Subscripts & superscripts

$(\cdot)_{\mathrm{A}}$	Acceptor quantity (fluid domain discretization)
$(\cdot)_{\mathrm{D}}$	Donor quantity (fluid domain discretization)
$(\cdot)_f$	Facet of a moving boundary or interface (index)
$(\cdot)_h$	Spatially discretized value
$(\cdot)_i$	Node (index)
$(\cdot)_{i,j}$	Node (index) designated by its position in a 2D grid
$(\cdot)_k$	Face of a fluid finite volume (index)
$(\cdot)_n$	Time step (index)
$(\cdot)_{\mathrm{n}}$	Normal component
$(\cdot)_{\mathrm{t}}$	Tangential component
$(\cdot)_{\Gamma}$	Quantity on the fluid-structure interface
$(\cdot)^c$	Fluid cell (index)
$(\cdot)^e$	Solid element (index)
$(\cdot)^{\mathcal{F}}$	Fluid field related quantity
$(\cdot)^{\mathrm{MB}}$	Moving boundary related quantity (fluid domain discretization)
$(\cdot)^{\mathcal{S}}$	Solid field related quantity
$(\cdot)^{\star}$	Predicted quantity
$\bar{(\cdot)}$	Prescribed quantity
$\hat{(\cdot)}$	Averaged quantity
$\widetilde{(\cdot)}$	Intermediate quantity (fluid time discretization)

Operators & symbols

$(\cdot)^T$	Transpose of a tensor
$(\cdot)^{-1}$	Inverse of a tensor
$(\cdot)^{-T}$	Inverse of the transpose of a tensor
$\dot{(\cdot)}, \ddot{(\cdot)}$	First and second time derivative
det	Determinant
tr	Trace
$\frac{\mathrm{D}\cdot}{\mathrm{D}t}$	Material derivative
$\boldsymbol{\nabla}$	Gradient operator
$\boldsymbol{\nabla}_0$	Material gradient operator
$\boldsymbol{\nabla}_H(\cdot)$	High-order gradient operator (fluid time discretization)
$\boldsymbol{\nabla}_{\boldsymbol{n}}$	Directional derivative
$\boldsymbol{\nabla}\cdot$	Divergence operator
$\boldsymbol{\nabla}_0\cdot$	Material divergence operator
$\otimes$	Dyadic product
$\boldsymbol{I}$	Identity tensor
$\delta(\cdot)$	Variation of a quantity
δ_{ij}	Kronecker's delta

Domains & boundaries

$\Omega_0^{\mathcal{S}}, \Omega^{\mathcal{S}}$	Solid domain in the reference and current configuration
$\Omega^{\mathcal{F}}$	Fluid domain
$\partial\Omega_0^{\mathcal{S}}, \partial\Omega^{\mathcal{S}}$	Solid boundary in the reference and current configuration
$\partial\Omega^{\mathcal{F}}$	Fluid boundary
Γ	Fluid-structure interface
$\Gamma_{\mathrm{D}}^{\mathcal{S}}, \Gamma_{\mathrm{N}}^{\mathcal{S}}$	Solid Dirichlet and Neumann boundary
$\Gamma_{\mathrm{D}}^{\mathcal{F}}, \Gamma_{\mathrm{N}}^{\mathcal{F}}$	Fluid Dirichlet and Neumann boundary
Γ^{MB}	Moving boundary within the fluid domain

Sets & spaces

$\mathbb{N}$	Set of natural numbers
$\mathbb{N}_0$	Set of natural numbers (including 0)
$\mathbb{R}$	Set of real numbers
$\mathcal{D}$	Space of solution functions (PVW in solid mechanics)
$\mathcal{S}^1$	Sobolev space of square-integrable functions equipped with first derivatives
$\mathcal{T}$	Space of test functions (PVW in solid mechanics)

Solid mechanics

$\boldsymbol{b}^{\mathcal{S}}$	External body force (per unit volume)
$\boldsymbol{d}^{\mathcal{S}}, \dot{\boldsymbol{d}}^{\mathcal{S}}, \ddot{\boldsymbol{d}}^{\mathcal{S}}$	Displacement, velocity and acceleration
$\mathsf{d}_n^{\mathcal{S}}, \dot{\mathsf{d}}_n^{\mathcal{S}}, \ddot{\mathsf{d}}_n^{\mathcal{S}}$	Discrete nodal displacement, velocity and acceleration at time t_n
c_n^d	Dilatational wave speed of the material in the solid system at time t_n
$\boldsymbol{C}$	Right Cauchy-Green tensor
$\boldsymbol{\mathcal{C}}$	Elasticity fourth-order tensor
$E^{\mathcal{S}}$	Young's modulus
$\boldsymbol{E}$	Green-Lagrange strain tensor
f_{ext}	Vector of external forces
f_{int}	Vector of internal forces
$\boldsymbol{F}$	Deformation gradient
g_{n}	Normal gap (contact)
$G^{\mathcal{S}}$	Shear modulus
k	Spring stiffness
k_{n}	Penalty stiffness parameter (contact)
$K^{\mathcal{S}}$	Bulk modulus
J	Jacobian of the deformation gradient $\boldsymbol{F}$
L_n^e	Characteristic length of element e at time t_n
m	Elementary mass
$\mathsf{M}^{\mathcal{S}}$	Mass matrix
$\mathsf{M}_{\text{L}}^{\mathcal{S}}$	Lumped mass matrix
$\boldsymbol{n}$	Outward unit normal vector
$\boldsymbol{n}_{\text{c}}$	Unit normal contact vector
N_i^e	Shape function of node i in element e
$\mathrm{N}_{\text{nodes}}^e$	Number of nodes in element e
p_{c}	Contact pressure
$\boldsymbol{P}$	First Piola-Kirchhoff stress tensor
$\boldsymbol{R}$	Rotation tensor
S	Integration surface
$S_{\Delta t}^{\mathcal{S}}$	Safety factor (time discretization)
$\boldsymbol{S}$	Second Piola-Kirchhoff stress tensor
t	Time
$t_n^{\mathcal{S}}$	n^{th} discrete time station
$\boldsymbol{t}^{\mathcal{S}}$	Traction vector (per unit area)
$\boldsymbol{t}_{\text{c}}$	Contact traction vector
$\boldsymbol{t}_{\text{t}}$	Tangential contact traction
T	Total simulation time
$\Delta t_n^{\mathcal{S}}$	n^{th} solid time step (size)
$\Delta t_n^{\mathcal{S},\text{cr}}$	n^{th} critical time step (size)
$\boldsymbol{U}$	Right stretch tensor
$\boldsymbol{v}_{\text{t, rel}}$	Relative tangential velocity vector (contact)

V	Integration volume
$\boldsymbol{x}^{\mathcal{S}}$	Position in the current configuration
$\boldsymbol{X}^{\mathcal{S}}$	Position in the reference configuration

α	Scalar parameter (contact)
$\boldsymbol{\varepsilon}^{\mathcal{S}}$	Cauchy strain tensor (small deformations)
$\lambda^{\mathcal{S}}, \mu^{\mathcal{S}}$	Lamé constants
μ_{c}	Friction coefficient
$\nu^{\mathcal{S}}$	Poisson's ratio
$\boldsymbol{\xi}$	Position in parametric space
$\rho^{\mathcal{S}}$	Solid mass density
$\boldsymbol{\sigma}^{\mathcal{S}}$	Cauchy stress tensor
$\boldsymbol{\Psi}$	Strain energy function (hyperelastic material models)
$\omega, \omega_{\mathrm{max}}$	Eigenfrequency and maximum eigenfrequency of the solid system

Fluid mechanics (if not defined previously)

$\boldsymbol{b}^{\mathcal{F}}$	External body force (per unit volume)
C	CFL number (time discretization)
C^{S}	Surface CFL number (time discretization)
f	Fluid quantity (space discretization)
F	Flux of a fluid quantity f (space discretization)
L_n^c	Characteristic length of cell c at time t_n
$\mathrm{N}_{\mathrm{cells}}^{\mathcal{F}}$	Number of cells in the computational mesh
$\mathrm{N}_{\mathrm{faces}}^{\mathcal{F}}$	Number of faces in a cell
$\mathrm{N}_{\mathrm{facets}}^{\mathrm{MB}}$	Number of plane facets composing a discrete moving boundary
p	Fluid (thermodynamic) pressure
$\bar{p}^{\infty}$	Prescribed pressure of reference (far-field value)
$\bar{p}_{\mathrm{out}}$	Prescribed pressure at an outlet boundary
q	Fluid property associated to Q
Q	VOF variable
S^c	External surface of a control volume (of a fluid cell c)
S_f	Surface of a plane facet composing a discrete moving boundary
$S_{\Delta t}^{\mathcal{F}}$	Safety factor (time discretization)
$\boldsymbol{t}^{\mathcal{F}}$	Traction vector (per unit area)
$\Delta t_n^{\mathcal{F}}$	n^{th} fluid time step (size)
$\Delta t_n^{\mathcal{F},\mathrm{cr}}$	n^{th} critical time step (size)
$\Delta t_n^{\mathrm{MB,cr}}$	n^{th} surface critical time step (size)
$\boldsymbol{u}^{\mathcal{F}}, \dot{\boldsymbol{u}}^{\mathcal{F}}$	Velocity and acceleration
$\boldsymbol{u}^{\mathrm{MB}}$	Velocity of a moving boundary within the fluid domain
$\bar{\boldsymbol{u}}_{\mathrm{wall}}$	Prescribed velocity at a wall boundary
$\bar{U}_{\mathrm{in}}$	Prescribed velocity magnitude at an inlet boundary

V^c	Control volume (volume of a fluid cell c)
$\boldsymbol{x}^{\mathcal{F}}$	Position in the current configuration
$\Delta\mathsf{x}^{\mathcal{F}}, \Delta\mathsf{y}^{\mathcal{F}}$	Discrete grid size in direction x and y
β	Time discretization parameter
$\boldsymbol{\varepsilon}^{\mathcal{F}}$	Strain-rate tensor
λ	Dimensionless distance (space discretization)
$\mu^{\mathcal{F}}$	Dynamic viscosity
$\nu^{\mathcal{F}}$	Kinematic viscosity
$\rho^{\mathcal{F}}$	Fluid mass density
$\boldsymbol{\sigma}^{\mathcal{F}}$	Cauchy stress tensor
$\boldsymbol{\tau}^{\mathcal{F}}$	Shear stress tensor

Fluid-structure interaction (if not defined previously)

$\boldsymbol{d}_{\Gamma}^{\mathcal{S}\star}$	Predicted solid displacement at the fluid-structure interface
D, D_d, D_k	Scalar parameters (energy-accuracy analysis)
$\Delta E_{\Gamma,n}$	Interface energy during n^{th} coupling time step
$\Delta \boldsymbol{E}_{\Gamma,n}^{\mathrm{l}}$	Vector of local (nodal) energy imbalance during n^{th} coupling time step
$\Delta E_{\Gamma,n}^{\mathcal{F}}$	Fluid contribution to the interface energy during n^{th} coupling time step
$\Delta E_{\Gamma,n}^{\mathcal{S}}$	Solid contribution to the interface energy during n^{th} coupling time step
δE	Error estimation of the interface energy
$\boldsymbol{f}_{\Gamma}^{\mathcal{F}}$	Fluid forces on the fluid-structure interface
$\boldsymbol{f}_{\Gamma}^{\mathcal{S}}$	Flow-induced solid forces on the fluid-structure interface
h	Adimensional parameter of the coupling time step (energy-accuracy analysis)
M_{up}	Number of steps averaged for a coupling time step size decrease
M_{up}	Number of steps averaged for a coupling time step size increase
$\mathrm{N}_{\Gamma,\text{faces}}^{\mathcal{F},f}$	Number of fluid faces created on a single solid face f of the interface
$\mathrm{N}_{\text{nodes}}^{\Gamma}$	Number of shared nodes on the FSI interface
$R_n^{\mathcal{F}}$	Fluid sub-cycling ratio at n^{th} coupling time step
$R_n^{\mathcal{S}}$	Solid sub-cycling ratio at n^{th} coupling time step
$S_{fk}^{\mathcal{F}}$	Surface of a fluid face k associated with a solid face f of the interface
Δt_n	n^{th} coupling time step (size)
Δt_{min}	Minimum coupling time step size
Δt_{max}	Maximum coupling time step size
T_{c}	Characteristic time (energy-accuracy analysis)

α_0, α_1	Solid predictor parameters
ϵ	Tolerance (FSI time step adaptation)
$\kappa_n^{\circledcirc}$	Optimized scaling factor (FSI time step adaptation)
κ_{min}	Minimum scaling factor (FSI time step adaptation)
κ_{max}	Maximum scaling factor (FSI time step adaptation)
κ_{s}	Safety factor (FSI time step adaptation)

Numerical examples

$\bar{F}_z^{\mathrm{GW}}, \bar{F}_z^{\mathrm{tire}}$	Prescribed vertical load for Grosch wheel and tire
$\boldsymbol{g}$	Gravitational acceleration
$H^{\mathrm{GW}}, H^{\mathrm{tire}}$	Height of the tire side wall for Grosch wheel and tire
m_{car}	Mass of the car
p_{infl}	Tire inflation pressure
R^{GW}	Radius of the non-rotating Grosch wheel
$R_{\mathrm{dyn}}^{\mathrm{GW}}$	Dynamic radius of the Grosch wheel
R^{rim}	Radius of the rim
$\bar{T}_y^{\mathrm{tire}}$	Prescribed torque around the rotating axis of the tire
$\boldsymbol{u}_{\mathrm{rel}}^{\mathcal{F}}$	Fluid relative velocity
$\bar{U}_x^{\mathrm{road}}, \bar{U}_x^{\mathrm{car}}$	Prescribed velocity of the road and the car in the moving direction
$W^{\mathrm{GW}}, W^{\mathrm{tire}}$	Width of Grosch wheel and tire
$\bar{\boldsymbol{\Omega}}_y^{\mathrm{GW}}$	Prescribed angular velocity around the rotating axis of the Grosch wheel
$\mathit{\Omega}_y^{\mathrm{dry}}, \mathit{\Omega}_y^{\mathrm{wet}}$	Angular velocity around the rotating axis of the tire in dry and wet conditions

Abbreviations

ALE	Arbitrary Lagrangian-Eulerian
API	Application programming interface
AMR	Adaptive mesh refinement
CEL	Coupled Eulerian-Lagrangian
CFD	Computational fluid dynamics
CSS	Conventional serial staggered
DEM	Discrete element method
EOS	Equations of state
FD(M)	Finite difference (method)
FE(M)	Finite element (method)
FSI	Fluid-structure interaction
FV(M)	Finite volume (method)
GCL	Geometric conservation law
GSS	Generalized conventional serial staggered

HHT	Hilbert-Hughes-Taylor (time integration scheme)
IB(M)	Immersed boundary (method)
IBVP	Initial boundary value problem
ISS	Improved serial staggered
LB(M)	Lattice Boltzmann (method)
MD	Molecular dynamics
RCA	Remaining contact area
PVW	Principle of virtual work
SPH	Smooth particle hydrodynamics
SR	Slip rate
SUV	Sport utility vehicle
VOF	Volume of fluid

1 Introduction

1.1 Motivation

Tire hydroplaning, also known as tire aquaplaning, is familiar to all road users driving under wet conditions. This physical phenomenon occurs when a water layer builds between the road surface and the wheel of the vehicle, leading to a loss of contact between the tire and the road. As a consequence the driver can lose control of its vehicle, which can lead to dramatic consequences. To tackle this safety hazard the tire and automotive industry have been actively exploring ways to improve the performance of tires under wet conditions and in various driving configurations such as braking, accelerating or cornering. This tire performance hence dominates the overall driving safety in certain critical situations. As a matter of fact, the EU commission regulation on tire labeling effective since 2012 has retained wet grip as one of the three main tire characteristics together with the tire rolling resistance and noise [43].

The phenomenon of tire hydroplaning counts to one of many applications in the research area of *fluid-structure interaction* (FSI), in which the behavior of fluid and structure is closely interdependent. The deformation of a solid depends on a fluid load applied onto its external surface and, at the same time, the behavior of the fluid flow depends on the shape and motion of the solid body. This class of coupled problems requires a simultaneous solution of the governing equations of both fluid and solid single-fields as they interact and are influenced by each other. It is therefore very difficult, if not impossible, to solve these equations analytically so that a numerical approach is required in order to compute the inherently nonlinear and time-dependent FSI problems. One speaks then of *computational fluid-structure interaction*. In particular, the case of tire-water interaction can be raised among the most challenging computational FSI problems due to a large number of constraints. They include, but are not limited to, the nature of the materials at stake, the high rotating speed of the wheel, the complex geometry of the tire tread, its large deformations near the road and the contact occurring at the so-called *triple region* where the tire meets the road and the water. These constraints are an additional source of nonlinearity and instability in the coupled system and need to be treated with care.

Typical experimental investigation of hydroplaning performance of tires are expensive and highly time consuming as an industrial process. Not only are there many different tire designs to be tested but each of them also comes in a large variety of sizes and constitutive materials. In addition to that, and more importantly, the results of experimental testing are quite limited with respect to the local resolution of relevant physical effects and are therefore not sufficient to understand the underlying physics. These reasons also call for the use of numerical simulation while dealing with the tire hydroplaning problem.

Thereupon the aim of this thesis is to improve the numerical simulation of tire hydroplaning as a computational FSI problem. While the numerical modeling of fluids and solids has advanced tremendously in the last few decades, the combination of both physics remains a very challenging subject. This is addressed in this work, while focusing on the coupling techniques that are most suitable for tire applications. Further investigation in the field is needed as very limited solutions, usually neglecting very important aspects of the problem, are proposed today to predict properly the hydroplaning phenomenon of tires.

Aiming at expanding the general understanding of the tire hydroplaning problematic and improving the wet performance of new tire designs, Goodyear[1] as one of the world's leading tire manufacturer, initiated the present research. As a consequence, the proposed findings are also expected to fit in an industrial framework. Adding to the inherent complexity of the problem, not only the implemented solution should be robust, it is also required to run in a time efficient manner.

Based on an academic-private partnership, this project could benefit from Goodyear's many years of experience in numerical modeling of tires as well as from the renown expertise in computational methods of the Institute for Computational Mechanics[2] of the Technical University of Munich. As a very challenging application of computational FSI, the generated research presents therein a great interest for the scientific community. Supporting industries and research conducted in Luxembourg where Goodyear is implanted, the FNR[3] co-funded the present project.

1.2 Tire hydroplaning

The physics of tire hydroplaning is further described in this section. Causes and factors contributing to the phenomenon are given, which opens the door to solutions aiming at tackling this particular problem. These solutions involve the experimental testing of tires on wet road surfaces in order to evaluate their wet performance and, in turn, improve the construction and design of future tires. Experimental techniques used for that purpose are shortly presented. Limitations of experimental measurements are highlighted, which shows the need of complementary numerical results in order to properly predict the tire hydroplaning problem. Computational methods employed to that effect are reviewed in section 1.3.3.

[1] *The Goodyear Tire & Rubber Co.*, `www.goodyear.com`.

[2] *Lehrstuhl für Numerische Mechanik*, `www.lnm.mw.tum.de`.

[3] *Fonds National de la Recherche*, `www.fnr.lu`.

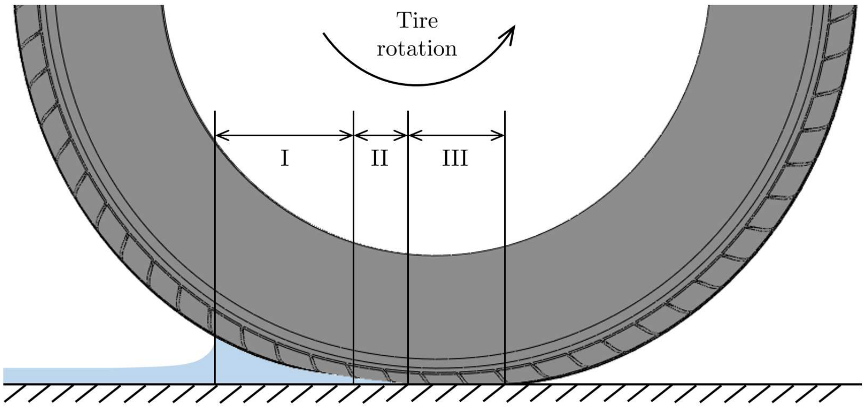

Figure 1.1: Three zones in the tire hydroplaning process according to Hays and Browne [82]: hydrodynamic (I), viscous (II) and full contact (III) regions.

1.2.1 Physical process

The hydroplaning phenomenon takes place when a tire loses grip to the road due to the presence of a water film between the two bodies. A good tire design will ensure that the tire remains in contact with the road in any foreseeable weather conditions. In wet conditions, the contact between the tire and the road can be described by the three-zone concept outlined e.g. in [82] as illustrated in Figure 1.1. In the hydrodynamic region (I), inertial effects are predominant and no contact between the tire and the road exists. The tire leading edge collides with the water layer, which leads to a rise in the fluid hydrodynamic pressure as a water wedge forms in front of the tire footprint. The lift force resulting from the fluid pressure increase deforms the tire, which detaches from the road surface. The viscous region (II) is a transition zone dominated by viscous effects of water where some rubber-road contact exists. The contact pressure is large enough to counteract the fluid pressure, which is not sufficient to lift the tire off the road. As for the third zone, it is the full contact region (III). From a wet grip point of view, the perfect tire design aims at evacuating as fast as possible water away from the contact patch area by channeling water through the groove network of the tire. This limits in turn the rise in fluid pressure and generates the largest dry contact region (zone III) in the back of the footprint. Ultimately, if p_c is the contact pressure between the tire and the road and p the hydrodynamic pressure, hydroplaning will occur if

$$p > p_\mathrm{c}. \tag{1.1}$$

By integrating p over the wetted surface area and p_c over the contact area between the road and the tire, the condition can also be expressed as follows: tire hydroplaning occurs if the fluid lift force (resulting reaction force from the fluid onto the tire) is greater than the vertical load applied on the tire (weight of the vehicle if dynamic effects are neglected).

The factors influencing tire hydroplaning are numerous and stem from different origins. Already in the early 60's, Horne and Dreher published in [88] some observations about the phenomenon. They were used as a starting point in order to establish a list of important parameters contributing to the hydroplaning phenomenon, namely:

- Fluid parameters: fluid layer thickness and density. Surprisingly, the fluid viscosity was not listed as one of the impacting factors in [88] even though it clearly influences the tire wet performance, as the fluid viscosity affects the ability of the tire to squeeze through the water film.
- Tire parameters: inflation pressure (the higher the pressure, the better), tread design (footprint shape, groove network), tire construction, tire width (the narrower, the better).
- Vehicle parameters: weight and speed.
- Road surface parameters: pavement shape and texture.

To that, the driver's behavior plays obviously an enormous role but cannot easily be taken into account. Altogether, only the tire parameters can be influenced by the tire designer in order to prevent and predict tire hydroplaning. However, as external environment factors have a huge impact on the wet performance of tires, they must be taken into account while creating new tire designs. To that effect, the tire industry commonly uses experimental testing and numerical simulations which, as shown in the following, are both essential for evaluating properly a tire performance and for understanding the phenomenon itself.

1.2.2 Experimental testing

Experimental testing is of prime importance as it allows to quantify the hydroplaning performance of actual tire designs. The test results obtained in real conditions are very valuable to the designers during the construction iteration of new tire models. Among the tests commonly carried out by the tire industry, the most relevant in the present context are:

- Glass plate hydroplaning test (see Figure 1.2 (left)): a straight free-rolling (no acceleration nor braking) tire is driven over a glass plate flooded with up to 3 mm of water. The shape as well as the remaining contact area of the tire footprint can be measured by post-processing pictures taken from below the glass plate by a high speed camera. This test allows to assess the ability of a tire to evacuate water away from the footprint region.
- Deep straight aquaplaning test (see Figure 1.2 (right)): the tire is tested with a thicker fluid layer on actual asphalt. This test is a full-throttle acceleration test, for which the slip rate of the tire on the road is measured by comparing its angular velocity in wet and dry conditions. In this case, the footprint shape and remaining contact area cannot be measured.

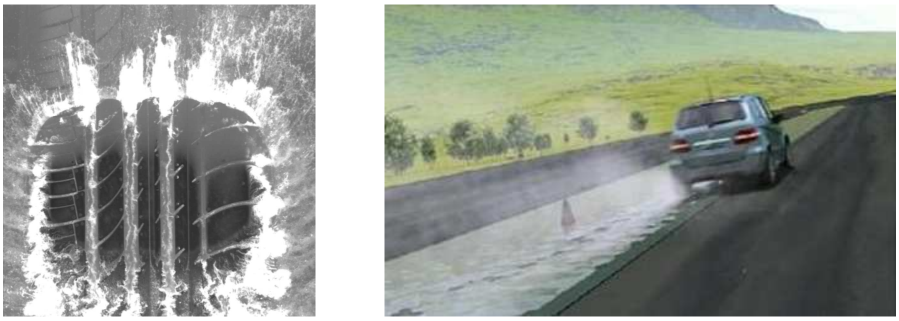

Figure 1.2: Snapshot of the footprint in a glass plate hydroplaning test (left) and setting of the deep straight aquaplaning test in which a vehicle drives over a flooded asphalt surface (right) [171].

- Curved aquaplaning test: same as the straight aquaplaning test except that the tire is driven in a curve. Here, the experimental indicator is the lateral acceleration the tire is able to undergo in a curve.

Not only these experimental tests are very expensive to carry out, especially considering the fact that each idea for a tire design leads to the construction of the actual tire in multiple sizes, they are also not sufficient in terms of produced output data due to the complexity of the testing framework. Very interesting data such as the pressure distribution in the footprint or a precise measurement of the tire deformation in the contact patch are not provided by experimental measurements under wet conditions. While assessing the dry performance of tires, captors built into the surface on which the tire rolls allow to measure the contact pressure as well as the shape of the footprint. It is not possible in the wet case due to the presence of water, which disrupts the measurements. In order to evaluate such quantities for full-scale tire designs, gain a broader understanding of the phenomenon and limit the cost of experiments, numerical simulation of the tire hydroplaning problem is achieved. It consists of a particular example of computational FSI as described in the following section.

1.3 Computational methods for fluid-structure interaction

Before introducing the techniques used specifically for the numerical investigation of tire hydroplaning, the research area of computational FSI to which this application belongs is presented in this section.

As suggested by its name, the field of computational FSI focuses on modeling and simulating numerically the interaction of a movable rigid or deformable solid with a fluid flow. It counts to one of the most challenging multi-field class of problems and represents therefore a research area of continuously growing interest. The applications of FSI span a broad range of domains in engineering and natural science where experimental results are limited or even nonexistent. The analysis of aircraft wing flutter [51, 147, 177] or wing flapping [100, 185], the deployment of parachutes [168, 170] or airbags [176], the response of civil engineering structures to aerodynamic forces [144, 191], the lung breathing process [182, 193] or the blood flow in vessels [17, 133], and many more, are all topics of past and on-going research. Each one of these applications has its own particularity with specific challenges and requirements. The problematic of tire hydroplaning at hand counts to one of the most complex FSI applications due to the running conditions as well as the materials and geometries involved. A detailed description of the computational framework for tire hydroplaning is given in chapter 5, while numerical results for a full-scale tire model are presented.

The solution procedures for computational FSI problems can be classified into different categories. They are distinguished here by the formulations and the discretization approaches for each single-field, and by the coupling scheme.

1.3.1 FSI problem formulations and space discretization

The problem formulation takes place at the continuous level before any kind of numerical discretization. More complicated than single-field problems, FSI problems require the sets of differential equations and boundary conditions associated with each physical sub-domain to be satisfied simultaneously at the shared *fluid-structure interface* along which both physics are coupled. A correct FSI problem statement consists then of the description of the solid and fluid single-fields with their boundary conditions and appropriate *coupling conditions* enforcing displacement and load continuity at the shared interface.

As the solid moves through space, the shape of the fluid sub-domain must change to conform to every new position of the interface over time. The motion of the fluid sub-domain is therefore also function of the unknown solid displacement. This makes FSI a three-field problem where the unknowns are not only the solid and fluid fields but also the motion of the fluid sub-domain. There are different formulations available to describe the material motion in each participating single-field: a *Lagrangian* also called *material description*, and an *Eulerian* or *spatial description*. In a Lagrangian formulation, a material particle is followed and its properties are monitored as it moves through space and time. In an Eulerian formulation, it is the properties of all particles passing through a fixed location in space that are monitored. These modeling choices made at a continuous level have direct implications on the space discretization of each single-field. Indeed, in the discrete setting, a Lagrangian formulation translates into a mesh following the material motion where the unknowns are the position of each mesh node. Whereas for an Eulerian-based discretization, the unknowns are evaluated at fixed points in space within a motionless mesh and the continuum moves with respect to the grid, which corresponds to a so-called *fixed-grid* approach as opposed to a *moving-grid* approach. The modeling of FSI systems under large displacements usually involves a combination of several formulations and discretization techniques.

Solid formulation

The solid field is treated most of the time by a Lagrangian formulation, benefiting from the ability to track easily free-surfaces and interfaces between different materials. However it deals poorly with large distortions of the computational mesh without frequent remeshing of the solid subdomain. Most of solid dynamics code use this approach in combination with the *finite element method* (FEM) for discretizing the domain in space.

Fluid formulation

The description of the fluid motion in a deforming domain is based in general on two formulations, either on an Eulerian formulation or on the so-called *Arbitrary Lagrangian Eulerian* (ALE) formulation that combines both Lagrangian and Eulerian descriptions.

Moving-grid approach In the first case of an ALE-based formulation [40, 41], the nodes of the computational fluid dynamics (CFD) mesh can move as in a Lagrangian approach, remain fixed in an Eulerian manner or move arbitrarily to optimize the quality of the elements. The fluid field is then discretized and solved on a deforming mesh, which follows the movement of the fluid-structure interface [50, 126, 166]. As a result, the solid and fluid mesh can match exactly at the fluid-structure interface. However, if both solvers use different meshes or discretizations, a mapping between fluid and solid quantities at the interface is required. A review of such coupling methods is done in [30] and recent developments based on a *dual mortar* formulation dealing with non-conforming meshes at the interface can be found e.g. in [94]. Although ALE methods are perfectly suited for particular FSI problems, their use becomes restricted when the wet surface of the solid undergoes a large displacement as the fluid mesh distorts excessively. A robust mesh moving technique that also allows periodically regenerating the fluid mesh is then necessary to preserve the quality of the grid and the accuracy of the computation, which can be computationally expensive and even fail in some extreme cases of mesh deformation. It is worth noting here that an analog strategy based on a moving computational domain called *space-time finite element* formulation has been proposed by Tezduyar et al. [169], see e.g. [170] for recent developments.

Fixed-grid approach In the second case of a pure Eulerian description, the fluid field is discretized on a fixed-grid. Avoiding the aforementioned drawbacks associated with large displacement of the fluid-structure interface by immerging the wet boundary surface of the solid in a fixed fluid mesh, this approach also handles well contacts [116, 150] of moving bodies or topology changes [53, 186] inside the fluid sub-domain. Simplifying greatly the computational mesh generation, Eulerian-based methods require however computational geometry procedures to capture the evolution of the position, shape and topology of the dynamic surface representing the fluid-structure interface [111]. Techniques aiming at reconstructing the discretized interface within the fixed fluid background mesh and impose the coupling condition are known as *immersed boundary*, *embedded boundary*, *fictitious domain* or *Cartesian methods*. For simplicity, all of these non-body-conforming methods will be referred to here as immersed boundary (IB) methods. Originally proposed by Peskin more than 40 years ago in [141, 142], IB methods were first developed for incompressible flows in conjunction with a *finite difference method* (FDM). Since then their applications have been expanded with success to different discretization techniques such as *finite volume methods* (FVM) [28, 129, 184], FE approaches [7, 91, 175, 194] or with a *extended finite element method* (XFEM) [74, 116, 165]. For more information on IB methods, the interested reader is referred to the review articles [89, 124]. Particular attention must be paid to viscous FSI problems where these methods generally do not allow to capture boundary layers around the moving wet interface, requiring in turn a higher CFD mesh resolution in large parts of the fluid sub-domain. Not feasible for large FSI problems due to obvious computational efficiency reasons, the resolution of the boundary layer during the numerical simulation can be maintained by using an *adaptive mesh refinement* (AMR) method as proposed first by Berger et al. in [15, 16] and recently in e.g. [110]. Recent developments by Shahmiri et al. in [159] and Schott et al. in [156] also include the use of an XFEM-based formulation to embed arbitrary mesh patches into a fixed background grid. These mesh patches follow the motion of the solid, allowing in turn to accurately resolve the flow in the boundary layer region. An analogous approach proposed by Farhat et al. in [44, 100] and based on a so-called *ALE-embedded* computational framework translates and rotates the underlying fixed-grid in order to effectively capture the position of the boundary layer. Although easy to implement, this technique is in general not as computationally efficient as AMR methods, which are widely used in commercial and research CFD codes [42].

Consequently, Eulerian fixed-grid approaches together with IB methods are particularly attractive for dynamic FSI problems characterized by large motions and deformation of the solid as well as with contacts, topology changes or other geometrical complications, for which ALE methods are often not feasible. Together with an appropriate treatment of moving boundary layers with a procedure such as e.g. AMR, viscous fluids can be treated accurately in a coupled problem, while the solid single-field is in general solved by a FEM on a Lagrangian computational mesh.

1.3.2 Coupling schemes

In FSI problems both solid and fluid are interdependent such that the equilibrium conditions between the two sub-domains at the wet interface must be satisfied at all times, making the numerical simulation of such strong coupled problems very challenging. The *coupling scheme* is the procedure used to couple both sub-systems in time, which can also be seen as the time discretization of the coupling conditions at the interface. They are commonly sorted into two distinct categories: *monolithic*, and *partitioned* or *staggered* procedures.

Monolithic coupling procedure

In the monolithic approach the solid and fluid single-fields are solved simultaneously within a unique coupled system of nonlinear equations. The mutual influence of both physics is taken directly into account preventing any unbalance of the continuity conditions at the wet interface. Monolithic procedures have been successfully applied to various problems such as channels with flexible walls [12, 83], thin-walled structures in the human blood system [69, 99] or the mechanical behavior of human red blood cells [93]. A recent approach allowing free and independent choice of time integration schemes in the solid and fluid fields has been proposed by Mayr et al. in [118]. While this ultimate form of strong coupling procedure can be optimized for a specific problem, it does not recognize the differences between the mathematical properties of the fluid and the solid sub-systems and requires most of the time a code developed for each particular combination of physical problems. This can result in an ill-conditioned coupled system for which existing numerical solution methods are mostly not adapted [147]. Furthermore, the size of the system and the lack of software modularity in monolithic schemes are proved to be a major obstacles in practice when large and complex industrial applications are considered, which usually directly disqualifies this approach for such problems.

Partitioned coupling procedure

In partitioned coupling strategies solid and fluid equations are solved separately, requiring an appropriate coupling procedure to establish the interaction and to determine the solution of the coupled problem. Advantages of such coupling procedures are numerous. In particular, the computational complexity per time step is reduced, *sub-cycling*[4] within each single-field problem is facilitated and software modularity achieved, allowing the solid and fluid flow equations to be solved by different and possibly more efficient techniques developed specifically for each sub-system. Within partitioned schemes, two sub-categories of coupling strategies exist: *loosely-coupled* or *strongly-coupled* partitioned procedures.

Loosely-coupled procedure In so-called loosely or *weakly-coupled* strategies, also known as *sequentially staggered methods*, a violation of the continuity conditions at the interface is allowed, which results in a coupling that is only weakly enforced between the two sub-domains.

[4] Sub-cycling refers to a technique allowing each sub-system in a partitioned scheme to be solved with an independent time step that is smaller than the synchronizing FSI time step.

In these coupling procedures, the solid and fluid are solved only once within each time step and a coupling algorithm establishing the interaction between the fluid flow and the solid motion is required. Clearly, a sequential treatment of the coupling introduces an explicit flavor into the solution method even when the individual sub-systems are solved by implicit time marching schemes. For this reason, loosely-coupled methods are also often designated by *explicit* coupling procedures. Therefore, restrictions on the time step have to be expected. While too large time steps lead to a partitioning error corresponding to a too large time lag between the two subsystems, observations show that decreasing the time step size causes an increase in instability when incompressible flows are considered. The problem known as *artificial added-mass effect* can be interpreted as though major parts of the fluid act as a supplementary mass on the solid degrees of freedom at the fluid-structure interface. It was already described by Mok et al. in [127] and Le Tallec et al. in [104], and a mathematical background was provided by Causin et al. in [25] by means of a reduced model problem. They showed that this disruptive effect yields unstable simulations and is reinforced in the case of a strong interaction between solid and fluid due to high fluid/solid density ratios and for problems with slender shapes. Consequently, the time step size is required to lie in a range that can be rather small or in some cases even empty, giving rise to unconditional numerical instability [66]. Nevertheless, loosely-coupled procedures can be constructed even in configurations subjected to high added-mass effects as further discussed in section 3.3.5.2. If the application allows it, loosely-coupled strategies can hence be recommended due to their computational efficiency, their simple implementation and their compatibility with black-box solvers.

Strongly-coupled procedure Strongly-coupled or *fully-coupled* solution strategies were introduced to overcome the deficiencies of loosely-coupled approaches, see e.g. [104, 114, 125]. In this case, the interface coupling conditions are enforced by performing *sub-iterations* between the fields before advancing to the next time station. This *implicit* coupling procedure, in contrast to the previously described explicit approach, is generally more robust and stable than loosely-coupled schemes but yields a highly nonlinear coupled system to be solved at each time step. As a matter of fact, as stated by Fernàndez in [60], ”in addition to the common nonlinearities of the fluid and solid equations, implicit coupling induces geometrical nonlinearities within the fluid equations [...]”, increasing substantially the complexity of the coupled system to solve as well as the computational cost every coupling time step. Several approaches to resolve the coupling in an iterative fashion have been proposed and are reviewed in the following.

A common method is the so-called *Dirichlet-Neumann* partitioning or *fixed-point iteration* formulation that uses a *block Gauss-Seidel* scheme as follows: the fluid flow is solved for a given position of the shared interface then followed by the computation of the deformed solid subdomian due to the fluid load on the wet interface and iterations are performed between both single-fields until sufficient convergence is obtained. Due to the artificial added-mass effect for flexible solids and incompressible flows, the convergence rate of this scheme is rather slow. The instability can even result in a diverging solution leading to the unusual situation where the number of coupling iterations increases when the time step decreases [32, 66]. The convergence can however be accelerated by the use of *Aitken's dynamic underrelaxtion* [97], *transpiration techniques* [34], *vector extrapolation* [98] or *reduced-order models* [179]. In the recent years, a new family of partitioned procedures based on *Robin transmission conditions*, i.e. a linear combination of Dirichlet and Neumann boundary conditions on the shared interface, were introduced. While all combinations of boundary conditions do not necessarily make sense, a *Robin-Neumann* coupling procedure, that solves the fluid with a Robin boundary condition at the interface and the solid with a Neumann boundary condition at the interface, shows much better convergence without relaxation and guarantees added-mass free stability [8, 61, 132].

The coupled problem can also be implicitly solved with methods generically known as *Newton methods*. First issued from monolithic applications [83, 154], they are used to solve the solid and flow equations for the variables in the entire solid and fluid sub-domain with a Newton-Raphson solution procedure. These methods require knowledge of the exact Jacobian of both solid and fluid sub-systems to solve the linear system within the Newton-Raphson iteration [36, 59], which can be time consuming and not practical for the use of black-box solvers [32]. In an attempt to reduce these limitations, *quasi* or *inexact-Newton* methods using only approximate Jacobians confined only to the interface degrees of freedom were proposed by Gerbeau et al. in [72, 73] and by van Brummelen et al. in [122, 123]. While altogether more efficient and less restrictive, they may however lead to inefficient Newton iterations [72]. Also known as *interface-quasi Newton* methods [33, 121], these strategies show superior convergence of the coupling iterations compared to relaxation methods and have the advantage of being compatible with the use of black-box solvers.

In order to reduce the computational complexity, *semi-implicit* coupling schemes involving a simplified coupled problem, where part of it is treated explicitly, were developed e.g. in [6, 60, 152]. It is worth noting that these schemes are in essence strongly coupled as coupling conditions are still treated implicitly. Although they improve the computational efficiency of implicit coupling and are less sensitive to the added-mass effect, the stability is still sensitive to changes in the fluid/solid density ratio. In so-called *stabilized explicit* coupling schemes, a stability independent of the added-mass effect was achieved, however at the expense of a loss of accuracy compared to implicit counterparts. They were first proposed in [24] for thin structures and more recently extended to the case of the coupling with thick-walled structures in [61]. It is important to stress here that these schemes are explicit in the sense that they only make one iteration within each time step but, in preparation of that iteration, they still necessitate an implicit treatment of some interface terms. They are hence not directly comparable to loosely-coupled procedures, which represent truly explicit coupling schemes.

Some of the main implicit, semi-implicit and stabilized explicit strategies briefly introduced here are very well reviewed in [58], including comparative studies for stability and performance. The interested reader is also referred to Küttler [96] and Mok [125] for a more general presentation of partitioned coupling procedures.

The intent pursued in this section is merely to give a short historic and a basic overview of the methods that are currently developed by the scientific community to tackle the coupling of FSI problems, thus completeness is not claimed in any way. Especially regarding strongly-coupled procedures on which most of academic research is currently focusing, with the aim of improving computational efficiency or minimizing the number of sub-iterations to reach stability while keeping a high level of accuracy. But, as explained in section 1.4.1, the coupling strategy preferred in this work for the numerical simulation of tire hydroplaning is based on a loosely-coupled partitioned procedure. A thorough presentation of strongly-coupled schemes would therefore be out of scope of this thesis. The coupling strategy followed in this work will be presented in detail in chapters 3 and 4 and will be put in perspective with past and current developments in the field by means of an exhaustive literature review.

In the following section, an assessment of numerical studies addressing the problem of tire hydroplaning with computational FSI by both academia and industry is given.

1.3.3 Computational FSI for tire hydroplaning

As mentioned in section 1.2.2, experimental investigations of the tire hydroplaning phenomenon deliver limited results, which can be completed by numerical simulations. To that end, a computational FSI framework is built, in which the tire is the solid sub-domain and the water the fluid sub-domain. The tire hydroplaning problem is a challenging application in the field of computational FSI due to, among others, the size of the problem, the complexity of the solid model, the materials at hand, the free-surface flow as well as the running conditions. More details on the challenges that need to be considered are given in section 1.4.1.

The creation of the computational model and the solution procedure for computing this particular application are therefore not straightforward. As of today, there is only limited literature that covers that subject and the proposed methods usually consider very simplified tire models. However, a few of these are of interest and are reviewed in the following.

In the late 90's, Grogger and Weiss proposed in [76, 77] a simplified hydroplaning model for non-rotating tires. The fluid method solves a free-surface flow on a rather coarse mesh whereas a very simplified model is used for the tire. These early works compare numerical results to the experiment and highlight the importance of using deformable tires in order to obtain conclusive results in terms of pressure distribution in the footprint. In [157], Seta et al. use a computational framework based on the FVM and the FEM to solve respectively the fluid and solid sub-domains, which are coupled with the commercial package *MSC.Dytran*[5]. The latter is employed as a black-box tool, in which an explicit coupling procedure simply transferring coupling data through the

[5] `http://www.mscsoftware.com/product/dytran`.

interface is used. First results were obtained for a rotating tire with a partial tread pattern, whose effect on hydroplaning performance is shortly discussed. Okano and Koishi compare in [137] several tire models with simplified full tread patterns, for which the hydroplaning performance is for the first time quantified. Both physics are solved using the FVM-FEM combination and are also coupled with MSC.Dytran. As a further simplification, the water is considered in [137] as compressible, which renders the procedure independent of the artificial added-mass effect introduced in section 1.3.2. The same coupling tool is used by Cho et al. in [26], where the water flow is considered this time as incompressible and inviscid. A real tire design with a full tread pattern is computed successfully for the first time and compared in terms of hydroplaning performance to a simplified tire with three longitudinal grooves. Another approach consisting in using only one computational package provided by *SIMULIA*[6] to solve the complete FSI problem has been considered by Donatellis et al. in [38] and by Kumar et al. in [95]. The *Abaqus/Explicit* solid solver is employed together with the so-called *Coupled Eulerian-Lagrangian* (CEL) technique directly included in the solver. By doing so, the tire hydroplaning problem is computed using the FE formulation for solving both sub-domains and in a unique software environment, which simplifies greatly the computational framework as the difficulties of coupling properly external solvers, especially black-box solvers, are removed. The CEL approach is not based on the resolution of the Navier-Stokes governing equations to model the fluid flow. Instead, the material behavior of water is described using an equations of state (EOS) model [162], for wich mechanical properties of the fluid are to be given. In [180], Vincent et al. focus on the development of a CFD solver capable of dealing more accurately with unsteady multi-phase flows interacting with obstacles such as a tire. In that particular case, the effect of the flow on the tire deformation is not taken into account so that one cannot really speak of a two-way coupled FSI simulation. This introduces a substantial limitation to the approach since the tire deformation due to the presence of water is non-negligible and needs to be considered for computing accurately tire hydroplaning problems.

The research community and industry have only had limited success at tackling the hydroplaning problem due to the complexity of the computational model and is therefore actively looking at improving it. In the previous review, many simplifications have been emphasized and are summarized in Table 1.1 with the main features of each approach to facilitate the comparison to the present research. In particular, some simplifications are not acceptable when the wet performance of real tires is to be numerically computed, such as by considering non-rotating, non-deformable tires or unrealistic tread designs. Furthermore, several approaches consider unique commercial packages to solve the FSI problem. This prevents of making any further developments in either the single-field solvers or the coupling algorithm. While the use of specifically designed solvers can be advantageous, relying completely on one black-box computational package does not allow to choose the best possible fluid, solid and coupling computational tools available on the market. As discussed in the next section, the present work considers two separate commercial solvers, which were specifically chosen for the current application, while a coupling algorithm

[6] `https://www.3ds.com/products-services/simulia/`.

Literature	Year	Solid & fluid formulation	Coupling scheme	Simplifications limitations
Grogger and Weiss [76, 77]	1996-97	FEM-FVM	Implicit coupling	Simplified stationary tire model and very coarse mesh
Seta et al. [157]	2000	FEM-FVM	Explicit coupling (MSC.Dytran)	Partial tread pattern and black-box coupling shell
Okano and Koishi [137]	2001	FEM-FVM	Explicit coupling (MSC.Dytran)	Compressible fluid and black-box coupling shell
Cho et al. [26]	2006	FEM-FVM	Explicit coupling (MSC.Dytran)	Black-box coupling shell
Donatellis et al. [38] and Kumar et al. [95]	2009-12	FEM-FEM	Explicit coupling (CEL)	Closed computational package and EOS fluid model
Vincent et al. [180]	2011	Analytic-FVM	-	No tire deformation and no coupling
Present work	2018	FEM-FVM	Flexible explicit coupling	Unique combination of single-field solvers (can be replaced)

Table 1.1: Review of computational FSI approaches for tire hydroplaning.

has been implemented to couple both black-box solvers at hand. In doing so, a flexible approach is proposed, which implements further developments using single-field outputs in order to improve the coupling procedure, while controlling the error introduced by the partitioning process.

1.4 Thesis scope and objectives

Based on the previous presentation of the tire hydroplaning application, the scope of the thesis is more closely identified. Relying on the requirements of the problem at hand and on the constraints of the computational framework available, a new coupling scheme used for improving the resolution of the FSI problem is proposed.

1.4.1 Scope

Aiming at improving the FSI computational framework to simulate the tire hydroplaning application, the present research focuses on the coupling of a solid and a fluid solver. The two solvers have been chosen for their abilities at tackling the problem at hand as further detailed in the following. A coupling scheme capable of controlling both single-field solvers and of using relevant coupling output data is implemented in this work in order to improve the preexisting coupling strategy already in place at Goodyear. To that end, a partitioned approach is chosen, which allows to profit at most from the capabilities offered by each solver specifically designed to address the problem in question. The inherent modularity of partitioned approaches also allows to easily introduce new numerical schemes and models while keeping everything else the same [183]. The newly implemented procedure is not only required to deliver accurate results, it is also expected to fit in an industrial framework. Therefore, it must be robust and efficient[7] enough to include the tire FSI computations in the development loop of real tire designs, optimizing the wet performance in parallel with other tire characteristics such as durability, dry grip, noise or rolling resistance. The applicability of the research findings in a real industrial environment is therefore of prime importance.

The tire hydroplaning application is a complex computational FSI problem due to various reasons, which are listed in the following:

- First and foremost, the complexity of the solid tire model. The tire is a composite solid system consisting of multiple constituents with different rubber compounds, steel and synthetic reinforcements. Most of these materials exhibit a nonlinear behavior and must be described with nonlinear constitutive material models. Moreover, a full-scale tire design has a complicated geometry and includes different scales so that the solid model is of very large size.
- The solid problem includes external loading conditions (rim mounting, inflation pressure) and a transient fluid loading, which lead to a large deformation in the contact patch region where the tire meets the road and the water. Geometric nonlinearities must thus be taken into account while modeling the solid field.
- The contact between the tire tread and the road as well as with the rim add to the complexity of the overall solid computation.

[7] Note that throughout this thesis, the term *efficiency* denotes the efficiency in time, that is time-to-solution, unless specified otherwise.

- The large size of the fluid sub-domain influences substantially the overall computational time. The detailed geometry of the tire tread in the contact patch region requires a fine fluid mesh in order to accurately resolve the flow in each groove of the tire. As a result, the fluid computational system is also of significant size and expensive to solve.
- The fluid sub-domain includes a free-surface flow with large fluid velocity. Not only the free-surface of the flow needs a special treatment, the interface reconstruction between the fluid and solid sub-domains has to be properly taken care of. To do so, a fixed-grid approach together with an immersed boundary method is chosen in the fluid solver, which is particularly attractive for solving FSI problems with large interface motion as explained in section 1.3.1. A deforming fluid mesh would be impractical since it could not follow the motion of the interface as the tire rotates without constant and expensive remeshing. The situation is even worse in the contact region where the tire squeezes the water layer at high speed to eventually come in contact with the road.
- While reconciling both sub-domains, FSI problems with incompressible flows and high fluid/solid density ratios such as for the tire hydroplaning application are subjected to the artificial added-mass effect. As described in section 1.3.2, the latter limits substantially the numerical stability of the coupling procedure. For efficiency reasons required by the industrial aspect of this research and in order to limit the complexity per coupling time step, a loosely-coupled solution procedure is chosen. It is shown that, even though the present application is prone to the artificial added-mass instability, a robust loosely-coupled strategy can be used and deliver accurate results. Note that the chosen solver package used for solving the tire hydroplaning application does not allow for a strongly-coupled strategy. Indeed, both solvers in their present configuration do not include the possibility of redoing the same time step, which is needed per definition for achieving an implicit coupling.
- The fast running conditions with a high rotating velocity lead to a rapidly changing free-surface flow and interface geometry. These impact the size of the time step in both single-fields as well as the coupling time step size.
- Another limiting factor is that both single-field solvers are black-box software packages. Their coupling represents a challenging task, which needs proper attention as demonstrated in this work.

To compute the coupled system, a FE solid solver based on an explicit time integration scheme is used. The chosen explicit solver runs faster than an implicit one in the case of a rotating tire, which is characterized by a transient response. An implicit scheme is however used in the event of a static loading of the tire. The present solid solver is chosen for its ability to compute such a complex system in a time efficient manner and for its flexibility in defining material laws with data obtained from the lab. As for the fluid, an implicit solver and a FVM are used to compute the Navier-Stokes equations, while the resolution of the phase transfer for the free-surface flow is achieved by an explicit method. The overall fluid approach can hence be described as explicit.

One of the main advantages of the fluid solver is its ability to reconstruct very efficiently the interface within the fluid fixed-grid, in particular for complex interface geometries.

As already mentioned, both chosen single-field solvers were already used internally at Goodyear to compute the tire hydroplaning simulation. In that configuration, the coupling algorithm simply transfers interface data between both physical sub-domains, while being embedded within one of the black-box solvers. As a result, no access to the coupling scheme is provided, which prevents any further developments to improve the coupling procedure. To by-bass this limitation, a new coupling shell is implemented, which grants more flexibility to optimize the coupling of both solvers for the present industrial application, as outlined in the next section.

1.4.2 Proposal for a dynamic coupling scheme applied to tire hydroplaning

The proposed research aims at improving the coupling of a specifically chosen pair of solvers in order to compute the tire hydroplaning application. By optimizing the resolution of the overall coupled system, the intent is to deliver the most time efficient solution path for a given level of accuracy. To do so, this thesis proposes a novel approach for computing FSI problems using black-box solvers based on the following contributions:

1. Implement a coupling shell that synchronizes the actions of both single-field solvers and allows to manipulate coupling data flowing from one sub-domain to the other.
2. Within the new coupling shell, monitor the level of partitioning error in the coupled system for the computational package at hand.
3. Control the accuracy of the solution to the coupled problem based on the evaluation of the energy artificially produced at the interface in each time station during the simulation.
4. Optimize the time efficiency of the solution procedure by using a predictor-corrector approach and a dynamic process for adapting the FSI time step size.
5. Validate the new coupling procedure with an academic benchmark example and apply it to a simplified tire hydroplaning model.
6. Improve the hydroplaning application for a real full-scale tire construction using the proposed approach.

The first five above-listed points are also presented by Gillard et al. in [75] as a general method applicable for any chosen pair of solvers.

In summary, by using the new coupling shell, the energy imbalance is assessed at the interface in each coupling time step. This quantity is artificially introduced in the coupled system by the partitioned procedure, which per definition never conserves the interface energy [147]. The latter therefore represents a very meaningful quantity in order to evaluate the level of error within the coupled system. Based on that knowledge, the error can be bounded, which in turn allows to control the accuracy of the solution procedure. In doing so, two approaches are proposed. On the one hand, a predicted value for the solid displacement at the interface in the next coupling time station that allows to reduce the time shift between both sub-domains is used. On the other hand, a dynamic adaptation of the FSI time stepping procedure is proposed, which considers the level of the interface energy-based error to increase or decrease the coupling time step size. In the event of an increasing error, the FSI time step size is decreased in order to reach the required level of numerical stability. At the same time, if the level of error is under a certain threshold, the coupling time step size can be increased, which in turn allows to improve the time-to-solution of the approach. Ultimately, it is shown that the interface energy consists in a very meaningful criterion in order to build a robust loosely-coupled procedure to compute the problem at hand in a time efficient manner. Validated on an academic FSI example, a flexible flag in a flow, the new coupling procedure is then applied to the resolution of a simplified tire hydroplaning model known as the Grosch wheel example. The new approach is then used to improve the hydroplaning application for a real sport utility vehicle (SUV) tire construction.

Not only this research allows to quantify the quality of the numerical solution for the tire hydroplaning problem, it also leads to the reduction of the overall simulation time, which is a severe bottleneck of the current application. To the best of the author's knowledge, it is the first time that such a dynamic coupling scheme would be applied to the challenging FSI problem of tire hydroplaning.

1.5 Outline

The current research is presented in this manuscript and organized as follows in the next chapters. In chapter 2, the physical models of the solid and fluid single-fields as well as the corresponding solution procedure chosen to compute each physics are presented. The coupling of both sub-domains defining the FSI coupled problem is considered in chapter 3. Partitioned coupling procedures and in particular loosely-coupled approaches are further detailed as several coupling schemes are presented based on an exhaustive literature review. At that occasion, the notion of solid displacement predictor and the concept of artificial interface energy in an explicit coupling procedure are introduced. It is emphasized that the energy artificially introduced at the interface by the staggering process can be used to improve the robustness of the solution procedure. The partitioning error, which can be related to the interface energy imbalance, is presented for the current choice of single-field solvers. The artificial added-mass effect and the factors impacting this instability are also further described. In chapter 4, the coupling strategy used for solving the coupled system is introduced. Details are given on the implementation of the coupling shell and how it can interact with both solvers, while the coupling scheme used in this work is presented. The dynamic procedure employed for adapting the FSI time step size based on the level of interface energy-based error is introduced. The developments implemented in the new coupling scheme are then applied to two benchmark examples. First, a flexible flag in a flow, which is considered as a standard benchmark for FSI in the literature, is used to validate the coupling strategy. Second, a simplified hydroplaning simulation based on the so-called Grosch wheel example that includes most of the constraints to be considered while computing real scale tires is performed. Based on the level of error delivered by the evaluation of the interface energy, the time-to-solution of the Grosch wheel example is optimized while controlling the level of accuracy. Chapter 5 is devoted to the presentation of results for a full-scale tire model. After a brief description of the computational model used for tire hydroplaning simulations, numerical results typically employed for assessing the wet performance of tires are exposed for an SUV tire rolling at high speed in a thick water film. The optimization of the tire hydroplaning simulation is then presented. Finally, chapter 6 gathers the conclusions and outlook of the present work as main results and achievements are summarized and some ideas for possible improvements are given.

2 Single-field governing equations and computational models

The numerical simulation of mechanical problems requires the construction of appropriate models that faithfully describe the physics of interest. In the present multi-field framework, the physical model and the corresponding sets of governing equations must be established for both solid and fluid sub-domains. They are presented individually for each single-field starting with the equations of continuum mechanics before describing their numerical discretization in space and time. Due to the running conditions and the materials at hand in the present FSI problem, the systems to solve consist of the equations of nonlinear solid dynamics and the *Navier-Stokes equations* for an incompressible flow of a viscous fluid. Specifics regarding the numerical modeling of the tire hydroplaning problem are also given for both sub-domains.

It is worth noting at this point that the mathematical and physical characteristics of each single-field presented in this chapter have a direct impact on the stability and convergence properties of the FSI coupled problem. The computational models, including the discretization techniques and their implementation in appropriate software, must therefore be chosen carefully as they affect the construction of the coupling scheme presented in chapter 4. The subject matter discussed in the present chapter is essentially intended to demonstrate the capabilities of both single-field solvers for the tire hydroplaning application and provide an appropriate theoretical background for the following chapters.

2.1 Solid mechanics

After a brief introduction on the choice of computational model employed for the application at hand, the equations governing the motion of a solid body are presented then discretized in space and time with appropriate approaches. Specific features of the solid solver used in the context of tire simulations are then given, including the modeling of rubber compounds and the contact definition of the tire with the road and the rim.

2.1.1 Choice of computational model

The FSI applications in this work involve large solid deformations so that geometric nonlinearities must be taken into account while modeling the solid field. Moreover, some of the materials that are considered exhibit a nonlinear behavior and have to be described with use of nonlinear constitutive material models. An overview of nonlinear solid dynamics covering these topics is presented in this section. The intent is to describe the modeling of the solid field and, in particular, show how the coupling variables are computed on the solid side of the interface. For a more exhaustive and thorough presentation of solid and structural mechanics, the interested reader is referred to the literature and the textbooks by e.g. Bonet and Wood [20], Holzapfel [87] or Marsden and Hughes [113].

The solid motion is described by a Lagrangian formulation corresponding, after discretization in space, to a moving mesh. In combination with the finite element method, this choice is very common practice and is made by almost all computational solid dynamics code developers. While many time integration schemes are available and frequently used to discretize the solid equations, the application of tire hydroplaning of interest limits this choice to an explicit integration operator, as further discussed in section 2.1.4. The corresponding implementation in the FE solver employed in this work is presented in the following.

2.1.2 Governing equations in solid mechanics

2.1.2.1 Kinematics

An homogeneous solid body defined by all its material points in a three-dimensional *Euclidean space* is considered. These material points initially located at some positions $\boldsymbol{X}^{\mathcal{S}}$ in space form the *reference configuration* $\Omega_0^{\mathcal{S}} \subset \mathbb{R}^3$ of the solid domain and move to a new position $\boldsymbol{x}^{\mathcal{S}}$, which corresponds to a *current* or *deformed configuration* $\Omega^{\mathcal{S}} \subset \mathbb{R}^3$ at the current time t. Assuming that no material can appear or disappear, the history of the current position of material points can always be written as a bijective mapping between $\boldsymbol{x}^{\mathcal{S}}$ and $\boldsymbol{X}^{\mathcal{S}}$ such that

$$\boldsymbol{x}^{\mathcal{S}} = \boldsymbol{x}^{\mathcal{S}}(\boldsymbol{X}^{\mathcal{S}}, t). \tag{2.1}$$

The displacement field of the material points can then be expressed by

$$\boldsymbol{d}^{\mathcal{S}}(\boldsymbol{X}^{\mathcal{S}}, t) = \boldsymbol{x}^{\mathcal{S}}(\boldsymbol{X}^{\mathcal{S}}, t) - \boldsymbol{X}^{\mathcal{S}}, \tag{2.2}$$

as illustrated in Figure 2.1 for a material point designated by P. Since in a Lagrangian description all kinematic quantities are described with respect to material coordinates, the reference positions $\boldsymbol{X}^{\mathcal{S}}$ represent the independent variables in the problem formulation and the solid displacements $\boldsymbol{d}^{\mathcal{S}}(\boldsymbol{X}^{\mathcal{S}}, t)$ the primary field of unknowns to be solved.

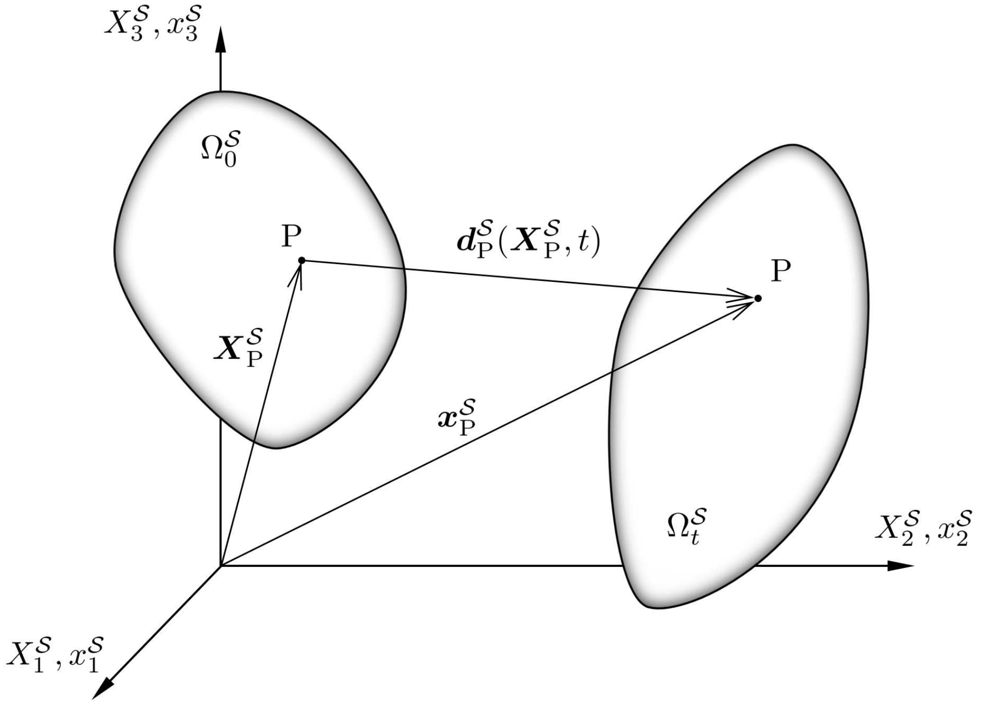
X_3^S, x_3^S
Ω_0^S
P
$\boldsymbol{d}_{\mathrm{P}}^S(\boldsymbol{X}_{\mathrm{P}}^S, t)$
P
$\boldsymbol{X}_{\mathrm{P}}^S$
$\boldsymbol{x}_{\mathrm{P}}^S$
Ω_t^S
X_2^S, x_2^S
X_1^S, x_1^S

Figure 2.1: Reference configuration Ω_0^S and deformed configuration Ω^S in a Lagrangian description of the solid motion.

The velocities $\dot{\boldsymbol{d}}^S(\boldsymbol{X}^S, t)$ and accelerations $\ddot{\boldsymbol{d}}^S(\boldsymbol{X}^S, t)$ are then defined in the reference configuration by

$$\dot{\boldsymbol{d}}^S(\boldsymbol{X}^S, t) = \frac{\mathrm{D}\boldsymbol{d}^S}{\mathrm{D}t} = \left.\frac{\partial \boldsymbol{d}^S}{\partial t}\right|_{\boldsymbol{X}^S} \tag{2.3}$$

$$\ddot{\boldsymbol{d}}^S(\boldsymbol{X}^S, t) = \frac{\mathrm{D}\dot{\boldsymbol{d}}^S}{\mathrm{D}t} = \left.\frac{\partial \dot{\boldsymbol{d}}^S}{\partial t}\right|_{\boldsymbol{X}^S} = \left.\frac{\partial^2 \boldsymbol{d}^S}{\partial t^2}\right|_{\boldsymbol{X}^S} \tag{2.4}$$

where, in a Lagrangian approach, the material derivative denoted here by $(\mathrm{D}\,\cdot\,/\,\mathrm{D}t)$ is simply the partial derivative with respect to time holding the material coordinate $\boldsymbol{X}^S$ fixed.

Considering now two neighboring material points located in $\boldsymbol{X}^S$ and $\boldsymbol{X}^S + \mathrm{d}\boldsymbol{X}^S$ in the reference configuration, the mapping (2.1) can be used to obtain in the deformed configuration

$$\mathrm{d}\boldsymbol{x}^S = \frac{\partial \boldsymbol{x}^S}{\partial \boldsymbol{X}^S} \cdot \mathrm{d}\boldsymbol{X}^S, \tag{2.5}$$

in which the tensor

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}^S}{\partial \boldsymbol{X}^S} \tag{2.6}$$

is known as the *deformation gradient*. Rewriting equations (2.5) as

$$\mathrm{d}\boldsymbol{x}^{\mathcal{S}} = \boldsymbol{F} \cdot \mathrm{d}\boldsymbol{X}^{\mathcal{S}}, \tag{2.7}$$

shows that the deformation gradient $\boldsymbol{F}$ can be interpreted as a mapping between an infinitesimal line element $\mathrm{d}\boldsymbol{X}^{\mathcal{S}}$ in the reference configuration and its image $\mathrm{d}\boldsymbol{x}^{\mathcal{S}}$ in the deformed configuration. It is the fundamental measure of deformation in continuum mechanics and is at the basis of several kinematic quantities as it allows to connect both configurations. For instance, its determinant $J = \det \boldsymbol{F}$, commonly known as the *Jacobian* of the deformation gradient, relates an infinitesimal volume element $\mathrm{d}V$ in the deformed configuration to its counterpart in the reference configuration $\mathrm{d}V_0$ as

$$\mathrm{d}V = J\,\mathrm{d}V_0. \tag{2.8}$$

The tensor $\boldsymbol{F}$ can also be expressed as the product of an orthogonal *rotation tensor* $\boldsymbol{R}$ and a so-called symmetric positive definite *right stretch tensor* $\boldsymbol{U}$ to define a *polar decomposition*, see e.g. [20] for more details, as

$$\boldsymbol{F} = \boldsymbol{R} \cdot \boldsymbol{U}, \tag{2.9}$$

which is used to introduce the *right Cauchy-Green* tensor $\boldsymbol{C}$ defined by

$$\boldsymbol{C} = \boldsymbol{F}^T \cdot \boldsymbol{F} = \boldsymbol{U}^T \cdot \boldsymbol{R}^T \cdot \boldsymbol{R} \cdot \boldsymbol{U} = \boldsymbol{U}^T \cdot \boldsymbol{U} = \boldsymbol{U}^2, \tag{2.10}$$

by exploiting the orthogonality of $\boldsymbol{R}$. While $\boldsymbol{F}$ is non-symmetric, the tensor $\boldsymbol{C}$ is and does not contain any rotation contribution of the deformation. Thus it can be used to formulate suitable strain measures if it is accounted for that $\boldsymbol{C} = \boldsymbol{I}$ for rigid body translations, where $\boldsymbol{I}$ is the second-order identity tensor. In that context, a proper strain measure is obtained with the *Green-Lagrange strain tensor* $\boldsymbol{E}$ defined, in the reference configuration, as

$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{F}^T \cdot \boldsymbol{F} - \boldsymbol{I}) = \frac{1}{2}(\boldsymbol{C} - \boldsymbol{I}), \tag{2.11}$$

which is also consistent in the sense that it guarantees zero strain in the undeformed configuration.

2.1.2.2 Balance equations

Solid dynamics problems can be described by balance equations of mass, momentum and energy, which advise on the conservation of these quantities within the mechanical system. The objective pursued here is to establish a mathematical formulation of the solid dynamics problem, which can be obtained from the balance equations of linear momentum, as shown in the following.

Balance of linear momentum

Exploiting *Newton's second law* [131], the balance of linear momentum expresses the dynamic equilibrium of all forces present in the system, i.e. the forces of inertia, the internal forces and the external forces acting on the solid body. The corresponding nonlinear *elastodynamic equations* written in the deformed configuration Ω^S and in their global form read

$$\int_{\Omega^S} \rho^S \,\ddot{\boldsymbol{d}}^S \,\mathrm{d}V = \int_{\Omega^S} \left(\boldsymbol{\nabla} \cdot \boldsymbol{\sigma}^S + \bar{\boldsymbol{b}}^S\right) \,\mathrm{d}V, \tag{2.12}$$

where ρ^S is the *solid density* of an infinitesimal volume element $\mathrm{d}V$, $\bar{\boldsymbol{b}}^S$ an external *body force*[1] acting on a unit volume element of the solid body in the deformed configuration and $\boldsymbol{\sigma}^S$ the *Cauchy stress tensor*. Also called “true” stress tensor, $\boldsymbol{\sigma}^S$ is a second-order symmetric[2] tensor and characterizes the true internal stress state within a solid body. According to the *Cauchy's stress theorem*, it can be related to the *traction vector* $\boldsymbol{t}^S$ representing a force (per unit area in the deformed configuration) acting at any point within the body associated with a plane of normal unit vector $\boldsymbol{n}$ as

$$\boldsymbol{t}^S = \boldsymbol{\sigma}^S \cdot \boldsymbol{n}. \tag{2.13}$$

Starting from the Cauchy stress tensor, other stresses can be defined. The *first Piola-Kirchhoff stress tensor* $\boldsymbol{P}$ relates to $\boldsymbol{\sigma}^S$ as

$$\boldsymbol{P} = J\,\boldsymbol{\sigma}^S \cdot \boldsymbol{F}^{-T}, \tag{2.14}$$

by recalling the definition of the Jacobian J from (2.8), and can be seen as a mapping from a surface element in the deformed configuration onto a force in the reference configuration. This interpretation suggests the possibility to build a stress tensor not based on variables in the deformed configuration as for $\boldsymbol{\sigma}^S$ but in the reference configuration instead. To do so, and also to avoid the non-symmetry of $\boldsymbol{P}$, the symmetric *second Piola-Kirchhoff stress tensor* $\boldsymbol{S}$ defined in the reference configuration by

$$\boldsymbol{S} = \boldsymbol{F}^{-1} \cdot \boldsymbol{P} = J\,\boldsymbol{F}^{-1} \cdot \boldsymbol{\sigma}^S \cdot \boldsymbol{F}^{-T} \tag{2.15}$$

is introduced. Unlike the Cauchy stress tensor $\boldsymbol{\sigma}^S$, $\boldsymbol{S}$ cannot be measured experimentally and does not allow for a clear interpretation. However, it plays an important role in the construction of material models as discussed in the next section. Furthermore, although several definitions of stress and strain tensors exist, they cannot all be combined arbitrarily. It is however the case for the second Piola-Kirchhoff stress tensor $\boldsymbol{S}$ and the Green-Lagrange strain tensor $\boldsymbol{E}$ defined in the reference configuration by respectively (2.15) and (2.11) as they form a *work conjugate*

[1] The notation $\bar{(\cdot)}$ denotes prescribed values.

[2] The symmetry of the Cauchy tensor can be deduced from the conservation of angular momentum.

pair (see e.g. [20]) and are therefore compatible to establish the FE formulation, which will be presented in section 2.1.3.

The balance equations (2.12) can now be reformulated with respect to the reference configuration as

$$\int_{\Omega_0^S} \rho_0^S \, \ddot{\boldsymbol{d}}^S \, \mathrm{d}V_0 = \int_{\Omega_0^S} \boldsymbol{\nabla}_0 \cdot (\boldsymbol{F} \cdot \boldsymbol{S}) + \bar{\boldsymbol{b}}_0^S \, \mathrm{d}V_0, \tag{2.16}$$

where ρ_0^S is the reference solid density[3] of the infinitesimal undeformed volume element $\mathrm{d}V_0$, $\bar{\boldsymbol{b}}_0^S$ the external body force per unit undeformed volume element and $(\boldsymbol{\nabla}_0 \,\cdot)$ the divergence operator with respect to the initial coordinate system or material divergence operator. Given that (2.16) also holds for any material point within the solid domain, the local form of the balance equations of linear momentum in the reference configuration Ω_0^S and $\forall t \in [0, T]$ then read

$$\rho_0^S \, \ddot{\boldsymbol{d}}^S = \boldsymbol{\nabla}_0 \cdot (\boldsymbol{F} \cdot \boldsymbol{S}) + \bar{\boldsymbol{b}}_0^S \quad \text{in } \Omega_0^S \times [0, T]. \tag{2.17}$$

2.1.2.3 Material laws

As shown before, the tensors $\boldsymbol{S}$ and $\boldsymbol{E}$ are commonly used to describe the stress and strain state of nonlinear solid dynamics problems. The *constitutive laws* or *material models* that describe the material response and relate stress and strain such as $\boldsymbol{S}(\boldsymbol{E})$ still need to be established. Only homogeneous solid bodies undergoing elastic deformation without internal dissipation and in an isotherm configuration are considered here. In particular, the solid models of interest in this thesis are composed of elastic and hyperelastic materials, for which the constitutive relations are shortly presented in the following.

Isotropic linear elasticity

If the deformation of the considered solid body is much smaller than its dimensions, it can be assumed that the geometry and constitutive properties at each material point are not modified after deformation. Thus deformed and undeformed configurations are presumed identical making the Cauchy stress tensor $\boldsymbol{\sigma}^S$ linearly dependent on the so-called *Cauchy strain tensor* $\boldsymbol{\varepsilon}^S$. This is expressed by the famous *Hooke's law*, which reads[4]

$$\boldsymbol{\sigma}^S = \boldsymbol{\mathcal{C}} : \boldsymbol{\varepsilon}^S \tag{2.18}$$

[3] The reference solid density can be written in terms of the solid density in the reference configuration as $\rho_0^S = J \, \rho^S$ by considering the conservation of mass in the solid domain.

[4] As opposed to the theory of large deformations presented in this section, the theory of small deformations only includes one definition of stress and strain as both reference and current configurations coincide, namely the Cauchy stress $\boldsymbol{\sigma}^S$ and the Cauchy strain $\boldsymbol{\varepsilon}^S$.

with

$$\boldsymbol{\varepsilon}^{\mathcal{S}} = \frac{1}{2}\left(\boldsymbol{\nabla d}^{\mathcal{S}} + (\boldsymbol{\nabla d}^{\mathcal{S}})^T\right) \tag{2.19}$$

where $\boldsymbol{\mathcal{C}}$ is the *elasticity tensor*. It is a fourth-order tensor that, for an isotropic material, is only function of two constants, as conveniently shown using the index notation by

$$\mathcal{C}_{ijkl} = \lambda^{\mathcal{S}}\delta_{ij}\delta_{kl} + \mu^{\mathcal{S}}\,\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right) \tag{2.20}$$

where δ_{ij} is the *Kronecker's delta*. The two *Lamé constants* $\lambda^{\mathcal{S}}$ and $\mu^{\mathcal{S}}$ can be related to the *Young's modulus* $E^{\mathcal{S}}$ and the *Poisson's ratio* $\nu^{\mathcal{S}}$ respectively as

$$\lambda^{\mathcal{S}} = \frac{E^{\mathcal{S}}\,\nu^{\mathcal{S}}}{(1+\nu^{\mathcal{S}})(1-2\,\nu^{\mathcal{S}})}, \;\; \mu^{\mathcal{S}} = \frac{E^{\mathcal{S}}}{2\,(1+\nu^{\mathcal{S}})}. \tag{2.21}$$

Isotropic hyperelasticity

Hyperelastic materials are still elastic but also allow for a nonlinear behavior if larger deformations are involved. In this case, the constitutive equations are based on a *strain energy function* or *elastic potential* Ψ to derive the stresses, which defines the strain energy stored in the material per unit of undeformed volume as a function of the strain at that material point [20]. It must remain invariant when the current configuration undergoes a rigid body rotation, which implies that Ψ cannot depend on the rotation tensor $\boldsymbol{R}$ but can only be expressed in terms of the right stretch tensor $\boldsymbol{U}$ and derived measures such as the right Cauchy-Green tensor $\boldsymbol{C}$ or the Green-Lagrange strain tensor $\boldsymbol{E}$. Stresses and strains are commonly connected in the reference configuration using the formulation

$$\boldsymbol{S} = 2\,\frac{\partial\Psi}{\partial\boldsymbol{C}} = \frac{\partial\Psi}{\partial\boldsymbol{E}}, \tag{2.22}$$

which is in general nonlinear. The elasticity tensor $\boldsymbol{\mathcal{C}}$ introduced in (2.18) for small deformations analysis can be reformulated in the context of hyperelasticity as

$$\boldsymbol{\mathcal{C}} = \frac{\partial\boldsymbol{S}}{\partial\boldsymbol{E}} = \frac{\partial^2\Psi}{\partial\boldsymbol{E}\partial\boldsymbol{E}}. \tag{2.23}$$

As a matter of fact, the *St.-Venant-Kirchhoff* hyperelastic and isotropic material model defined by the quadratic strain energy function

$$\Psi_{\text{lin}} = \frac{\lambda^{\mathcal{S}}}{2}\,(\text{tr}\,\boldsymbol{E})^2 + \mu^{\mathcal{S}}\boldsymbol{E}:\boldsymbol{E}, \tag{2.24}$$

where λ^S and μ^S are the Lamé constants given in (2.21), leads to a linear relationship between $\boldsymbol{S}$ and $\boldsymbol{E}$ such that

$$\boldsymbol{S} = \boldsymbol{\mathcal{C}} : \boldsymbol{E}, \tag{2.25}$$

with $\boldsymbol{\mathcal{C}}$ as defined in equation (2.20). The St.-Venant-Kirchhoff material model can therefore be interpreted as a generalization of Hooke's law given by (2.18) to mechanical problems with large deformations. Another model very commonly used to describe the behavior of hyperelastic materials is the *compressible Neo-Hookean* model. It is defined by the strain energy function

$$\Psi_{\mathrm{NH}} = \frac{\mu^S}{2}\,(\operatorname{tr}\boldsymbol{C} - 3) - \mu^S \ln J + \frac{\lambda^S}{2}\,(\ln J)^2 \tag{2.26}$$

and yields a nonlinear stress-strain relationship between $\boldsymbol{S}$ and $\boldsymbol{E}$. Simple to use, it delivers a good approximation for relatively small strains. There exists a multitude of hyperelastic material models, each one defined by a particular strain energy function that tries to fit at best the material response by also including thermal expansion effects or hysteresis behaviors. Some of these functions are described by an algebraic expression such as Ψ_{lin} or Ψ_{NH}, while others are directly constructed using experimental test data. Among the most well-known hyperelastic constitutive laws, the *Mooney-Rivin*, *Ogden* or *Yeoh* material models can be cited. As a thorough presentation of material models used in nonlinear solid dynamics would be out of scope of this work, the interested reader is referred to the classical textbooks by e.g. Holzapfel [87], Ogden [136], Simo and Hughes [161] or de Souza Nito et al. [31] for further details on hyperelasticity but also viscoelasticity, elastoplasticity or viscoplasticity. Note that any material model can be used within the coupling framework proposed in this thesis, as long as its implementation is possible in the chosen solid solver. Material models used for tire applications are briefly discussed in section 2.1.5.1.

Along with the kinematic equations and the balance equations of linear momentum, the material laws complete the mathematical formulation of the mechanical system and allow to form the so-called *initial boundary value problem* (IBVP), which will be presented in next section.

2.1.2.4 Initial boundary value problem

The expressions presented in the previous sections are gathered to construct the initial boundary value problem for nonlinear solid mechanics in the reference configuration[5]. The IBVP is based on a set of coupled second-order partial differential equations which govern the evolution of the solid displacement field $\boldsymbol{d}^S$, the stresses and strains in a time interval $t \in [0, T]$, which are further required to satisfy a set of given *initial* and *boundary conditions*. To define the boundary conditions, the solid boundary $\partial\Omega_0^S$ is partitioned into two complementary surfaces, not considering

[5] The IBVP can equivalently be described in the deformed configuration but the reference configuration is chosen here to simplify the derivation of the FE formulation as shown in section 2.1.3.

at this point the fluid-structure interface: a *Dirichlet boundary* Γ_{D}^{S} where displacements $\bar{\boldsymbol{d}}^{S}$ are prescribed and a *Neumann boundary* Γ_{N}^{S} where surface traction $\bar{\boldsymbol{t}}_{0}^{S}$ are imposed such that

$$\partial\Omega_0^S = \Gamma_{\mathrm{D}}^S \cup \Gamma_{\mathrm{N}}^S, \quad \Gamma_{\mathrm{D}}^S \cap \Gamma_{\mathrm{N}}^S = \emptyset. \tag{2.27}$$

Based on equations (2.17), the IBVP in the reference configuration then reads

$$\begin{aligned}
\rho_0^S \ddot{\boldsymbol{d}}^S &= \boldsymbol{\nabla}_0 \cdot (\boldsymbol{F} \cdot \boldsymbol{S}) + \bar{\boldsymbol{b}}_0^S && \text{in } \Omega_0^S \times [0,T], && (2.28a)\\
\boldsymbol{d}^S &= \bar{\boldsymbol{d}}^S && \text{on } \Gamma_{\mathrm{D}}^S \times [0,T], && (2.28b)\\
\boldsymbol{P} \cdot \boldsymbol{n}_0^S &= \bar{\boldsymbol{t}}_0^S && \text{on } \Gamma_{\mathrm{N}}^S \times [0,T], && (2.28c)
\end{aligned}$$

where $\boldsymbol{n}_0^S$ describes the outward normal unit vector in the reference configuration as illustrated in Figure 2.2. Furthermore, the system of partial differential equations being of second-order, two suitable initial conditions specifying the initial displacement $\bar{\boldsymbol{d}}_0^S(\boldsymbol{X}^S)$ and velocity $\bar{\dot{\boldsymbol{d}}}_0^S(\boldsymbol{X}^S)$ fields must be provided and are given by

$$\begin{aligned}
\boldsymbol{d}_0^S(\boldsymbol{X}^S, 0) &= \bar{\boldsymbol{d}}_0^S(\boldsymbol{X}^S) && \forall \boldsymbol{X}^S \in \Omega_0^S, && (2.29a)\\
\dot{\boldsymbol{d}}_0^S(\boldsymbol{X}^S, 0) &= \bar{\dot{\boldsymbol{d}}}_0^S(\boldsymbol{X}^S) && \forall \boldsymbol{X}^S \in \Omega_0^S. && (2.29b)
\end{aligned}$$

Finally, choosing an appropriate material model defined by e.g. (2.25) completes the IBVP for large deformations in the reference configuration. It is also referred to as *strong* or *local* form of nonlinear solid mechanics, as the system of equations (2.28-2.29) holds for all material points within the solid domain Ω_0^S. However, as shown in next section, a *weak* formulation is used to derive the finite element approximation while discretizing the solid domain in space.

2.1.3 Space discretization and finite element method

A general method to discretize the continuum mechanics problem in space consists in using the finite element method. By doing so, the governing equations describing the nonlinear behavior of the solid previously derived in the IBVP (2.28-2.29) are recast in a *weak integral form* using in this case the *principle of virtual work* (PVW). In a weak formulation, the balance equations and the boundary conditions are not required to be satisfied in every material point of the solid domain but only in an average (or integral) sense over an arbitrary finite number of elements. This will lead to a finite set of nonlinear algebraic equations in the primary field of unknowns $\boldsymbol{d}^S$, which is numerically computable. The FEM for solid mechanics is outlined in the following. For a thorough and detailed presentation of the method applied to various problems as well as its fundamental theory and implementation, the interested reader can consult, among others, the textbooks by Bathe [11], Hughes [90], Wriggers [190], Belytschko et al. [14], Larson [101] or Zienkiewicz et al. [195] .

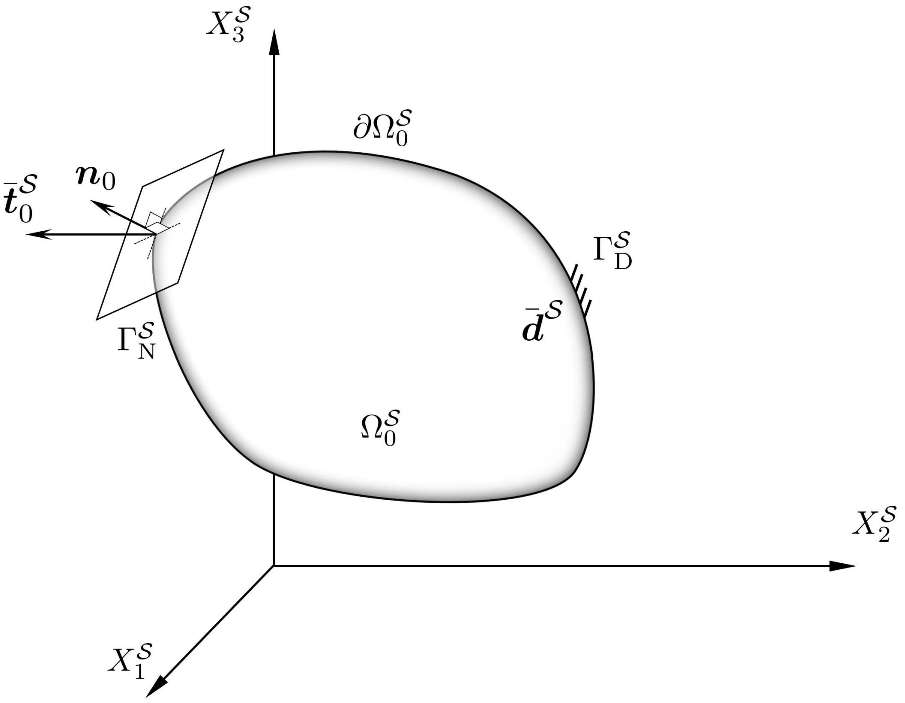
$X_3^{\mathcal{S}}$
$\partial\Omega_0^{\mathcal{S}}$
$\bar{\boldsymbol{t}}_0^{\mathcal{S}}$
$\boldsymbol{n}_0$
$\Gamma_{\mathrm{D}}^{\mathcal{S}}$
$\bar{\boldsymbol{d}}^{\mathcal{S}}$
$\Gamma_{\mathrm{N}}^{\mathcal{S}}$
$\Omega_0^{\mathcal{S}}$
$X_2^{\mathcal{S}}$
$X_1^{\mathcal{S}}$

Figure 2.2: Solid body in the reference configuration $\Omega_0^{\mathcal{S}}$ with the boundary surface $\partial\Omega_0^{\mathcal{S}}$ partitioned into a Dirichlet boundary $\Gamma_{\mathrm{D}}^{\mathcal{S}}$ and a Neumann boundary $\Gamma_{\mathrm{N}}^{\mathcal{S}}$, on which Dirichlet (2.28b) and Neumann (2.28c) boundary conditions are applied respectively.

Weak form of the balance equations

In order to construct the weak form of the solid mechanics problem, the balance equations (2.28a) and the Neumann boundary conditions (2.28c) are first rewritten as variational equations. Applying a *method of weighted residuals*, the *variational formulation* is obtained by multiplying (2.28a) and (2.28c) by an arbitrary *variation* or *test functions* $\delta\boldsymbol{d}^{\mathcal{S}}$. Following an integration over the reference solid volume, they read

$$\int_{\Omega_0^s} \left(\rho_0^{\mathcal{S}}\ddot{\boldsymbol{d}}^{\mathcal{S}} - \boldsymbol{\nabla}_0 \cdot (\boldsymbol{F}\cdot\boldsymbol{S}) - \bar{\boldsymbol{b}}_0^{\mathcal{S}}\right)\cdot\delta\boldsymbol{d}^{\mathcal{S}}\,\mathrm{d}V_0 + \int_{\partial\Omega_0^s}\left(\boldsymbol{P}\cdot\boldsymbol{n}_0^{\mathcal{S}} - \bar{\boldsymbol{t}}_0^{\mathcal{S}}\right)\cdot\delta\boldsymbol{d}^{\mathcal{S}}\,\mathrm{d}S_0 = 0, \tag{2.30}$$

where $\delta\boldsymbol{d}^{\mathcal{S}}$ can be interpreted as a *virtual displacement* field. The latter is required to be *kinematically admissible* [196], i.e. $\delta\boldsymbol{d}^{\mathcal{S}}$ must have sufficient continuity and cancel out where displacements are imposed:

$$\delta\boldsymbol{d}^{\mathcal{S}} = \boldsymbol{0} \qquad \text{on } \mathit{\Gamma}_{\mathrm{D}}^{\mathcal{S}} \times [0,T]. \tag{2.31}$$

Integrating by parts the second term of the first integral in (2.30), inserting (2.15) and applying (2.31) then yields

$$\int_{\Omega_0^s} \left(\rho_0^S \ddot{\boldsymbol{d}}^S - \bar{\boldsymbol{b}}_0^S\right) \cdot \delta \boldsymbol{d}^S \,\mathrm{d}V_0 + \int_{\Omega_0^s} \left(\boldsymbol{\nabla}_0 \delta \boldsymbol{d}^S\right)^T : \left(\boldsymbol{F} \cdot \boldsymbol{S}\right) \,\mathrm{d}V_0 - \int_{\Gamma_\mathrm{N}^s} \bar{\boldsymbol{t}}_0^S \cdot \delta \boldsymbol{d} \,\mathrm{d}S_0 = 0. \tag{2.32}$$

By expressing the variation of the Green-Lagrange strains due to a virtual displacement $\delta\boldsymbol{d}^S$ as, see e.g. [20],

$$\delta \boldsymbol{E} = \frac{1}{2}\left(\left(\boldsymbol{\nabla}_0 \delta \boldsymbol{d}^S\right)^T \cdot \boldsymbol{F} + \boldsymbol{F}^T \cdot \boldsymbol{\nabla}_0 \delta \boldsymbol{d}^S\right) \tag{2.33}$$

and, since $\boldsymbol{S}$ is symmetric, the variational formulation (2.32) can simplify into

$$\underbrace{\int_{\Omega_0^s} \rho_0^S \ddot{\boldsymbol{d}}^S \cdot \delta \boldsymbol{d}^S \,\mathrm{d}V_0}_{\delta W_\text{inert}} + \underbrace{\int_{\Omega_0^s} \boldsymbol{S} : \delta \boldsymbol{E} \,\mathrm{d}V_0}_{\delta W_\text{int}} = \underbrace{\int_{\Omega_0^s} \bar{\boldsymbol{b}}_0^S \cdot \delta \boldsymbol{d}^S \,\mathrm{d}V_0 + \int_{\Gamma_\mathrm{N}^s} \bar{\boldsymbol{t}}_0^S \cdot \delta \boldsymbol{d} \,\mathrm{d}S_0}_{\delta W_\text{ext}}, \tag{2.34}$$

which corresponds to the expression in the continuous form of the principle of virtual work in the reference configuration. It shows that the virtual work done by the external forces subjected to an arbitrary virtual displacement (δW_ext) is balanced by the virtual work done by the same virtual displacement field operating on inertial forces (δW_inert) and by strain variations (δW_int). The PVW (2.34) is the weak form of the IBVP and is used as a basic equilibrium statement to formulate an approximate solution using the finite element method. It is a very general concept in continuum mechanics as equations (2.34) are independent of any constitutive relations, making them valid for any material including solid, liquid or gas. Furthermore, equivalency between both forms can be shown, see e.g. [90], such that all solutions to the strong form satisfy the weak form. The opposite is however not true. As a result, the weak form proposes an equivalent representation of the partial differential equations (2.28) with weaker differentiability requirements and can be reformulated as follows.

Find $\boldsymbol{d}^S \in \mathcal{D}$ such that for all test functions $\delta\boldsymbol{d}^S \in \mathcal{T}$, equations (2.34) hold, where the *solution* and *test function spaces* are respectively defined by

$$\mathcal{D} = \left\{ \boldsymbol{d}^S \in \left[\mathcal{S}^1(\Omega^S)\right]^3 \mid \boldsymbol{d}^S(\boldsymbol{X}^S, t) = \bar{\boldsymbol{d}}^S(\boldsymbol{X}^S, t) \text{ on } \Gamma_\mathrm{D}^S \right\}, \tag{2.35a}$$

$$\mathcal{T} = \left\{ \delta\boldsymbol{d}^S \in \left[\mathcal{S}^1(\Omega^S)\right]^3 \mid \delta\boldsymbol{d}^S(\boldsymbol{X}^S) = \boldsymbol{0} \text{ on } \Gamma_\mathrm{D}^S \right\}, \tag{2.35b}$$

and $\mathcal{S}^1(\Omega^S)$ denotes the *Sobolev space* of square-integrable functions equipped with first derivatives [90].

Finite element formulation

To be able to solve numerically these integral equations, they must first be converted into a system of algebraic equations. To do so, the solid continuum is spatially discretized into a finite number of non-overlapping *elements*, which are connected at discrete points called *nodes*. The finite element method is now introduced to find a numerical solution to the unknown displacement field at each node of the partitioned solid domain, introducing for the first time an approximation of the solid governing equations.

The displacement field $\boldsymbol{d}^{\mathcal{S},e}$ within an element e is approximated by interpolating discrete nodal values $\mathbf{d}_i^{\mathcal{S},e}$ using so-called *interpolation* or *shape functions* $N_i^e(\boldsymbol{X}^{\mathcal{S}})$ for each node i in the element e as[6,7]

$$\boldsymbol{d}^{\mathcal{S},e}(\boldsymbol{X}^{\mathcal{S}},t) \approx \boldsymbol{d}_h^{\mathcal{S},e}(\boldsymbol{X}^{\mathcal{S}},t) = \sum_{i=1}^{\mathrm{N}^e_{\mathrm{nodes}}} N_i^e(\boldsymbol{X}^{\mathcal{S}})\ \mathbf{d}_i^{\mathcal{S},e}(t), \tag{2.36}$$

where $\mathrm{N}^e_{\mathrm{nodes}}$ denotes the number of nodes in the element e. The shape functions are usually low-order polynomials that are chosen to meet the differentiability requirements of the weak formulation. In doing so, the solution and test function spaces defined by (2.35) in the continuous setting are restricted to discrete sub-spaces $\mathcal{D}_h \subset \mathcal{D}$ and $\mathcal{T}_h \subset \mathcal{T}$, which respectively contain a finite number of solution and test functions chosen as C^0-continuous piecewise polynomials. Moreover, the *isoparametric* concept is used. It consists in employing the same shape functions to interpolate both the element geometry and displacements allowing to construct a mapping between the element physical coordinates $\boldsymbol{X}^{\mathcal{S},e} \in \mathbb{R}^3$ and parametric coordinates $\boldsymbol{\xi} = (\xi, \eta, \zeta) \in \mathbb{R}^3$, which describe a reference element. The finite element solutions over a physical element are eventually given by

$$\boldsymbol{d}_h^{\mathcal{S},e}(\boldsymbol{X}^{\mathcal{S},e},t) = \sum_{i=1}^{\mathrm{N}^e_{\mathrm{nodes}}} N_i^e(\boldsymbol{\xi}(\boldsymbol{X}^{\mathcal{S},e}))\ \mathbf{d}_i^{\mathcal{S},e}(t), \tag{2.37}$$

where the parametric shape functions $N_i^e(\boldsymbol{\xi}(\boldsymbol{X}^{\mathcal{S}}))$ satisfy the properties of *partition of unity* and of interpolation, which correspond respectively to

$$\sum_{i=1}^{\mathrm{N}^e_{\mathrm{nodes}}} N_i^e\left(\boldsymbol{\xi}\right) = 1 \quad \text{and} \quad N_a^e(\boldsymbol{\xi}_b) = \delta_{ab}, \tag{2.38}$$

and where δ_{ab} designates once again the Kronecker's delta. Furthermore, a *Bubnov-Galerkin* approach is adopted meaning that time derivatives of the displacement field and the test functions $\delta\boldsymbol{d}^{\mathcal{S}}$ are also approximated using the same shape functions. Depending on the choice of interpolation functions and geometric shape of the element, a multitude of finite elements can be created.

[6] The subscript $(\cdot)_h$ denotes spatially discretized values.

[7] The notations $e \in \mathbb{N}$ and $i \in \mathbb{N}$ refer to the element and node indexes, respectively.

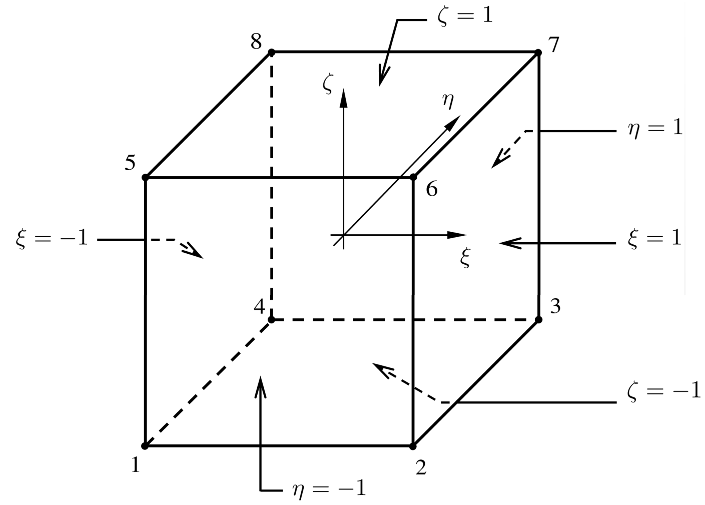

Figure 2.3: Solid isoparametric hexahedral 8-node element with 1st order interpolation shape functions defined in the parametric coordinates $\boldsymbol{\xi} = (\xi, \eta, \zeta)$.

For instance, for a 3D 8-node hexahedral solid element, the parametric domain consists of a bi-unit cube centered at the origin of the parametric coordinate system such as $(\xi, \eta, \zeta) \in [-1, 1]^3$ as shown in Figure 2.3. The isoparametric 1st order shape functions of this tri-linear hexahedron element are given by

$$\begin{aligned} N_1(\boldsymbol{\xi}) &= \frac{(1-\xi)(1-\eta)(1-\zeta)}{8}, & N_5(\boldsymbol{\xi}) &= \frac{(1-\xi)(1-\eta)(1+\zeta)}{8}, \\ N_2(\boldsymbol{\xi}) &= \frac{(1+\xi)(1-\eta)(1-\zeta)}{8}, & N_6(\boldsymbol{\xi}) &= \frac{(1+\xi)(1-\eta)(1+\zeta)}{8}, \\ N_3(\boldsymbol{\xi}) &= \frac{(1+\xi)(1+\eta)(1-\zeta)}{8}, & N_7(\boldsymbol{\xi}) &= \frac{(1+\xi)(1+\eta)(1+\zeta)}{8}, \\ N_4(\boldsymbol{\xi}) &= \frac{(1-\xi)(1+\eta)(1-\zeta)}{8}, & N_8(\boldsymbol{\xi}) &= \frac{(1-\xi)(1+\eta)(1+\zeta)}{8}, \end{aligned} \tag{2.39}$$

which indeed comply with the properties given in (2.38). Some remarks about the elements used for tire applications are given in section 2.1.5.1.

Finally, the discretized weak form is integrated over each element using a *Gaussian quadrature*, by which the integrands are evaluated at well-chosen integration points called *Gauss points*, see e.g. [90]. All element contributions are then assembled to form a global discretized formulation of the problem. Inserting the discrete approximation of the displacement field (2.37) and other quantities into the weak formulation (2.34) yields

$$\left(\mathbf{M}^{\mathcal{S}}\ddot{\mathbf{d}}^{\mathcal{S}} + \mathbf{f}^{\mathcal{S}}_{\text{int}}(\mathbf{d}^{\mathcal{S}}) - \mathbf{f}^{\mathcal{S}}_{\text{ext}}\right) \cdot \delta\mathbf{d}^{\mathcal{S}} = 0 \qquad \forall t \in [0, T] \tag{2.40}$$

or, since $\delta\mathbf{d}^{\mathcal{S}}$ is arbitrary,

$$\mathbf{M}^{\mathcal{S}}\ddot{\mathbf{d}}^{\mathcal{S}} + \mathbf{f}^{\mathcal{S}}_{\text{int}}(\mathbf{d}^{\mathcal{S}}) = \mathbf{f}^{\mathcal{S}}_{\text{ext}} \qquad \forall t \in [0, T], \tag{2.41}$$

where $\mathbf{M}^{\mathcal{S}}$, $\mathbf{f}^{\mathcal{S}}_{\text{ext}}$ and $\mathbf{f}^{\mathcal{S}}_{\text{int}}$ are global quantities which correspond respectively to the *solid mass matrix*, the *vector of external forces* and the nonlinear *vector of internal forces*. The nonlinear equations (2.41) commonly known as the *semi-discretized form of the PVW* or the *(global) semi-discretized equations of motion* are discretized in space but are still continuous in time. Their time discretization is presented in the following section.

2.1.4 Time discretization

The time discretization of the semi-discrete equations of motion is performed by approximating the time derivatives in (2.41) with use of discrete differentials. In doing so, the continuous time interval $t \in [0, T]$ considered so far is subdivided into a finite number of *time steps*. The time step size after $n \in \mathbb{N}_0$ time steps can be given by

$$\Delta t^{\mathcal{S}}_{n+1} = t^{\mathcal{S}}_{n+1} - t^{\mathcal{S}}_{n}, \tag{2.42}$$

where $t^{\mathcal{S}}_{n}$ and $t^{\mathcal{S}}_{n+1}$ denote the time stations at the beginning and at the end of the $(n+1)^{\text{th}}$ time step respectively.

The integration over a time step $\Delta t^{\mathcal{S}}_{n+1}$ is carried out by *time integration schemes* or *time integration operators* that can be characterized as *explicit* or *implicit*. Explicit operators evaluate the unknown solutions $\mathbf{d}^{\mathcal{S}}_{n+1} = \mathbf{d}^{\mathcal{S}}(t^{\mathcal{S}}_{n+1})$ at a time instant $t^{\mathcal{S}}_{n+1}$ based entirely on already computed quantities. This approach is easy to implement and very efficient as the solution to each time step is very cheap to compute. It is however at best conditionally stable, which introduces an upper limit for the choice of time step size requiring usually a very large number of time steps. To the contrary, implicit schemes are in general unconditionally stable and allow for larger time steps while solving a fully coupled system of equations. The evaluation of the unknowns $\mathbf{d}^{\mathcal{S}}_{n+1}$ depends not only on current and previous states of the system but also on variables in $t^{\mathcal{S}}_{n+1}$. Consequently, an iterative process is required to advance the solid field making the implementation of implicit schemes more complex and computationally more expensive in each time step than explicit approaches. Additionally, the method involves linearizing the entire finite element formulation, which can lead to overall higher simulation time compared to explicit approaches even though larger time steps are allowed. Ultimately it is the application and running conditions of interest that determine which time integration scheme should be used.

In the case of a rolling tire, neglecting at this point the interaction with the fluid, the solid model of a full-treaded tire is of very large size and involves many complex contact occurrences. Furthermore, the pressure loading in the contact patch region constantly varies over time (even more with additional fluid pressure in tire hydroplaning simulations) leading to a computation that can be seen as transient where no steady state solution exists. Due to these reasons, and to avoid convergence issues or even complete divergence of iterative procedures, an explicit operator is adopted in this thesis. Note that before being able to simulate a rolling tire, preliminary computations are needed to build the actual 3D computational tire model, refer to section 5.1 for further details. These preparation steps are carried out using an implicit scheme, for which a detailed description is however beyond the scope of this work. Further details on both explicit and implicit time integration in nonlinear FEM can be found in the literature, including the textbooks by Belytschko et al. [14] and Hughes [90], or in the volumes by Hairer et al. [80, 81] specifically dedicated to numerical integration methods for ordinary differential equations such as (2.41).

The explicit procedure employed in this thesis is based on the so-called *central difference integration rule* given e.g. in [90], where the equations of motion are satisfied at the beginning of the time step, i.e. in $t_n^{\mathcal{S}}$. The acceleration field computed at that time instant is used to advance the velocities to the time $t_{n+\frac{1}{2}}^{\mathcal{S}} = t_n^{\mathcal{S}} + \frac{\Delta t_{n+1}^{\mathcal{S}}}{2}$ and the displacements to $t_{n+1}^{\mathcal{S}}$ as

$$\dot{\mathbf{d}}_{n+\frac{1}{2}}^{\mathcal{S}} = \dot{\mathbf{d}}_{n-\frac{1}{2}}^{\mathcal{S}} + \frac{\Delta t_{n+1}^{\mathcal{S}} + \Delta t_n^{\mathcal{S}}}{2}\,\ddot{\mathbf{d}}_n^{\mathcal{S}}, \tag{2.43a}$$

$$\mathbf{d}_{n+1}^{\mathcal{S}} = \mathbf{d}_n^{\mathcal{S}} + \Delta t_{n+1}^{\mathcal{S}}\,\dot{\mathbf{d}}_{n+\frac{1}{2}}^{\mathcal{S}}, \tag{2.43b}$$

with

$$\ddot{\mathbf{d}}_n^{\mathcal{S}} = (\mathbf{M}_{\mathrm{L}}^{\mathcal{S}})^{-1}\left(\mathbf{f}_{\mathrm{ext},n}^{\mathcal{S}} - \mathbf{f}_{\mathrm{int},n}^{\mathcal{S}}\right), \tag{2.43c}$$

where $\mathbf{M}_{\mathrm{L}}^{\mathcal{S}}(\mathbf{M}^{\mathcal{S}})$ is a so-called *diagonal lumped mass matrix* and is constructed based on the solid mass matrix introduced in (2.41)[8]. The latter is in general not diagonal and can be difficult to invert. Therefore a lumped mass matrix $\mathbf{M}_{\mathrm{L}}^{\mathcal{S}}$ that, being diagonal, can be easily inverted and that needs much less computational storage than $\mathbf{M}^{\mathcal{S}}$ was introduced. Mass matrix "lumping" and diagonalization is discussed for the solution of transient problems with explicit analysis in e.g. Zienkiewicz et al. [195].

The attentive reader will already have noticed that the velocity field cannot be computed by (2.43a) at the first time step $\Delta t_1^{\mathcal{S}}$ $(n = 0)$ of the computation as mean velocities $\dot{\mathbf{d}}_{-\frac{1}{2}}^{\mathcal{S}}$ are not defined. At the initialization of the time stepping procedure, (2.43a) is therefore replaced by

$$\dot{\mathbf{d}}_{\frac{1}{2}}^{\mathcal{S}} = \dot{\mathbf{d}}_0^{\mathcal{S}} + \frac{\Delta t_1^{\mathcal{S}}}{2}\,\ddot{\mathbf{d}}_0^{\mathcal{S}}. \tag{2.44}$$

Overall, the central difference is 2nd order accurate and conditionally stable as further discussed in the following.

[8] Note that the lumped mass matrices are actually constructed at a local element level based on element mass matrices and are then assembled to form the global mass matrix $\mathbf{M}_{\mathrm{L}}^{\mathcal{S}}$. For the sake of clarity, only global matrices are denoted here.

Stability of the explicit operator

While the explicit time integration scheme is fairly simple to implement, it is only conditionally stable and requires a constraining upper limit on the time step size, which can be related to a *critical time step size* $\Delta t^{\mathcal{S},\mathrm{cr}}$ such that

$$\Delta t^{\mathcal{S}} \leq \Delta t^{\mathcal{S},\mathrm{cr}} = \frac{2}{\omega_{\mathrm{max}}}, \tag{2.45}$$

where ω_{max} denotes the maximum eigenfrequency of the entire solid dynamics system, see e.g. [90]. In practice, the stable critical time step size is computed for each element of the mesh individually and the minimum value is kept as upper limit. Taking that into account, the critical time step size can also be rewritten, see e.g. [11], as

$$\Delta t^{\mathcal{S},\mathrm{cr}} = \min_{e} \left\{ \frac{L^e}{c^d} \right\}, \tag{2.46}$$

where L^e is a characteristic element dimension and c^d the *current dilatational wave speed of the material*, which both depend on the used element. In nonlinear analysis, i.e. with geometric and/or material nonlinearities, these wave speeds represented in the finite element system change during its response so that their values need to be updated in the course of the simulation as the stability limit continually changes. Therefore the limit on the time step size must be evaluated for each time step n, which eventually reads

$$\Delta t^{\mathcal{S}}_{n} \leq S^{\mathcal{S}}_{\Delta t}\, \Delta t^{\mathcal{S},\mathrm{cr}}_{n} = S^{\mathcal{S}}_{\Delta t}\, \min_{e} \left\{ \frac{L^e_n}{c^d_n} \right\}, \tag{2.47}$$

where a safety factor $S^{\mathcal{S}}_{\Delta t} \in [0, 1]$ was introduced to guarantee a conservative estimation of the critical time step size. From expression (2.47) it can be deduced that the size of elements in the computational mesh have a very large influence on the critical time step size. This leads to a severe limiting factor for solid models that are composed of very small elements as, for instance, the tread region in a rolling tire simulation. A stability limit in the form of (2.47) is also commonly called *CFL condition* named after Courant, Friedrichs and Lewy [27] and is widely used in the analysis of mechanical systems.

Acceleration technique

As previously discussed, the stability limit for explicit analyses can be quite restrictive and a large number of time steps is in general necessary. In order to reduce computational cost and make the time integration operator more efficient, various techniques exist to loosen the constraint given in (2.47). The one employed in this work is a so-called *local mass scaling factor* and consists in scaling the mass of specific elements in the solid domain. These elements are either very small (as previously explained) or very stiff compared to the rest of the elements in the system and thus control on their own the overall stable time step size. The mass scaling must however be applied with care as an accurate representation of the physical mass and inertial effects in the model is required to accurately capture the transient response of the system. In general, as the mass of only a few elements is scaled, the impact on the system is negligible while the time step size can be increased significantly.

In order to briefly illustrate the influence of the mass scaling factor, a 1D mass-spring system with a concentrated mass m and a spring of stiffness k is considered. The value of its unique eigenfrequency is given by $\omega = \sqrt{k/m}$, see e.g. [71]. It is obvious that increasing the mass will reduce the eigenfrequency, while reducing at the same time the effect of a large stiffness. Comparing that observation with the expression (2.45) conceptually shows the positive effect of a local mass increase on the critical time step size. Different forms of mass scaling techniques for an explicit central difference procedure can also be found e.g. in [138].

The resolution of the solid mechanics system considered in this thesis has been presented. The semi-discrete equations of motion (2.41) with the explicit time integration operator given by (2.43) eventually allow to compute the solid displacements $\mathbf{d}^{\mathcal{S}}$, including the coupling variables describing the motion of the solid body within the coupled multi-field system. Before introducing the solution procedure used to compute the fluid sub-domain, specificities proper to the simulation of tire problems are proposed in the following.

2.1.5 Tire-specific modeling aspects

This section is intended to present specific features included in the solid computational model that are used to simulate the problem of tire hydroplaning or more precisely of a rolling tire, as no coupling with the fluid model is considered at this point. Since a detailed presentation of the solid model does not represent the core topic of this thesis, only an overview of the main specificities are introduced here. These include a description of the finite elements and material models employed for rubber compounds as well as contact definitions of the tire with the road and the rim. Complementary information concerning the composition of tire models including reinforcement materials as well as the generation of the full-scale 3D tire model will be given in section 5.1, while presenting associated results.

2.1.5.1 Incompressibility and finite elements for rubber compounds

Rubber materials used in tires exhibit a nearly incompressible behavior in the sense that their compressibility is very small compared to their shear flexibility. This particular property of a material known as *relative compressibility* can be assessed by the ratio of its *bulk modulus* $K^{\mathcal{S}}$ to its *shear modulus* $G^{\mathcal{S}}$, which characterize the ability to resist volume change and shear stress

	Steel	Cured rubber
$K^{\mathcal{S}}$ [GPa]	175	2
$G^{\mathcal{S}}$ [GPa]	80	0.002
$\nu^{\mathcal{S}}$ [-]	~0.3	~0.4995

Table 2.1: Steel and cured rubber material properties.

respectively. Typical values for these quantities are given in Table 2.1 for steel and cured rubber[9], which shows that only considering the compressibility of a material with $K^{\mathcal{S}}$, i.e. its resistance to volume change, does not allow to asses completely its incompressible behavior: the bulk modulus for steel is almost two orders of magnitude larger than that of rubber even though steel is definitely not considered as incompressible. However, the relative compressibility, which can be expressed by the Poisson's ratio $\nu^{\mathcal{S}}$ as

$$\nu^{\mathcal{S}} = \frac{3\,K^{\mathcal{S}}/G^{\mathcal{S}} - 2}{6\,K^{\mathcal{S}}/G^{\mathcal{S}} + 2} \tag{2.48}$$

also listed in Table 2.1 is the right indicator with $\nu^{\mathcal{S}} \to 0.5$ for incompressible materials.

It is common knowledge that conventional finite element formulations often exhibit too stiff behaviors in their mesh with a (nearly) incompressibility material response. This phenomenon known as *volumetric mesh locking* must be treated with care and is often handled by imposing an incompressibility constraint to the FE system (2.41) with an approach such as the *Lagrange multiplier method*. A mixed finite element method that enforces the incompressibility conditions with the Lagrange multiplier method for a full-scale tire model is presented in [149]. However, in the explicit procedure presented in section 2.1.4, a strict incompressibility constraint cannot be imposed in each node of the solid system and some compressibility must be provided. To describe the volumetric locking mentioned previously, a 3D mesh composed of the hexahedral solid element introduced in 2.1.3 is used. On average over the whole mesh, one can consider that there are three degrees of freedom per element. Considering *fully-integrated elements*, i.e. all terms in the equations of motion (2.41) are integrated in 8 Gauss points per element, an incompressibility condition would impose that the volume at each of these points must remain fixed, while only three degrees of freedom are available. This results in an over-constrained mesh, which eventually locks and causes a too stiff behavior for deformations that should not lead to any volume changes. Analogously to the volumetric locking, *shear locking* appears as fully-integrated 1st order elements cannot provide a pure bending solution and create parasitic shear stresses at the Gauss points, which in turn leads to artificial stiffening and eventually locks the mesh.

A remedy to these instabilities consists in using *reduced-integration finite elements*, which do not suffer from the same over-constraint as fully-integrated elements and can even eliminate

[9] Cured rubber designates a rubber material in its vulcanized state as opposed to its natural state. After completion of their construction process, tires are composed of different cured rubber compounds.

locking. Reduced integration means here that a lower-order integration is carried out to form the element stiffness [63]. In other words, the vector of internal forces $\mathbf{f}_{\text{int}}^{\mathcal{S}}$ in (2.41) is integrated in a unique integration point to obtain the strains at the center of the element, while the mass matrix $\mathbf{M}^{\mathcal{S}}$ and external loads $\mathbf{f}_{\text{ext}}^{\mathcal{S}}$ still use a full 8-point integration. However, due to that unique integration point, it is possible for the element to distort in such a way that the strains calculated at the Gauss point remain all zero. This instability called *zero-energy* or *hourglass mode*, named after the shape the distorted element can have in that particular configuration, leads to nodal displacements, which neither produce any strain energy nor define a rigid body motion. Since elements do not have any stiffness in these modes, they are unable to resist this type of deformation. These nonphysical response modes can then propagate through the mesh unless they are controlled, which is done in this case by adding small hourglass stiffness using the *artificial damping method* given in [63].

For these reasons, the rubber parts of the tire are discretized with 3D 8-node reduced-order hexahedral elements in tire rolling and tire hydroplaning applications. Furthermore, 1st order elements with appropriate hourglass control are preferred to 2nd order elements as they are more efficient in terms of computational time and perform well in problems with complex contact definitions, as their nodal forces are directionally consistent with the stresses in the element. First order elements also directly use a lumped mass matrix so that the lumping of each element mass matrix mentioned for (2.43c) is not needed. Finally the neo-Hookean model defined in (2.26), with an appropriate value of the Poisson's ratio to allow for little compressibility, is employed to describe the material response of cured rubber in tires, which corresponds to an approximately incompressible model that is compatible with the explicit integration scheme given by (2.43).

Note that no viscoelastic material models are used for tire hydroplaning simulations as viscous and dissipative effects are not predominant while evaluating the resulting vertical reaction force or the remaining footprint contact area, both introduced in section 5.2. In other fields of tire mechanics such as the ones focusing on rolling resistance, viscous and thermal effects can be taken into account by considering more complex material models. Nevertheless, note that a precise characterization of the material properties of rubber compounds used in tires remains to this day a quite challenging task, even at a experimental level in the lab. For futher details on material models used for rubber materials the interested reader is referred to e.g. [130].

2.1.5.2 Tire-road and tire-rim contact definitions

The tire model includes various contacts, including the contact of the tire with the road and with the rim. In both these cases, the tire is in contact with a rigid surface and the contact constraints are enforced using a so-called *penalty method* briefly described in the following. For a more extensive review of penalty methods and computational contact mechanics in general, the interested reader is referred to the textbook e.g. by Laursen [102] and Wriggers [189]. A review of existing contact formulations is also given by Popp in [150], see also references therein.

The contact enforcement presents an additional constraint that must be incorporated into the solid mechanics formulation. In contrast to more complex approaches such as the previously mentioned Lagrange multiplier method that adds constraint equations to the FE system, the penalty method simply introduces contact constraints that can be interpreted as a supplementary set of boundary conditions. In doing so, the penalty contact algorithm generates contact forces to resist node-into-rigid surface contact penetrations, while guaranteeing conservation of momentum between the contacting bodies. The nodes on which the constraints are applied form a so-called *slave surface*, which belongs in this case to the tire, while the interacting surface is referred to as the *master surface*, i.e. the road or the rim. The classical set of conditions for contact enforcement can be stated as

$$g_\text{n} \geq 0, \tag{2.49a}$$

$$p_\text{c} \leq 0, \tag{2.49b}$$

$$p_\text{c}\, g_\text{n} = 0, \tag{2.49c}$$

$$p_\text{c}\, \dot{g}_\text{n} = 0, \tag{2.49d}$$

where g_n denotes the *normal gap* between a pair of contact entities and p_c the normal component of the *contact traction vector* $\boldsymbol{t}_\text{c}$ also known as *contact pressure* such that $p_\text{c} = \boldsymbol{t}_\text{c} \cdot \boldsymbol{n}_\text{c}$, with $\boldsymbol{n}_\text{c}$ representing the unit normal contact vector (normal to the slave surface). These conditions can easily be interpreted as (2.49a) corresponds to the constraint of non-penetration of the two bodies, whereas (2.49b) avoids adhesive forces on the contact interface, (2.49c) sets the contact pressure to zero when the gap is open and keeps the gap closed when contact occurs, and (2.49d) is a complement to condition (2.49c) allowing it to persist in time.

When bodies are in contact, not only normal traction but also tangential traction commonly known as *frictional forces* are transmitted across the contact interface. In the case of tire-road contact, an *isotropic Coulomb friction model*, see e.g. [189], is used to relate the maximum allowable frictional forces to the contact pressure between the two bodies. Both contacting surfaces can carry frictional forces across their interface up to a certain magnitude known as *frictional capacity* before they start sliding relatively to one another, which corresponds to the following additional conditions to be fulfilled in each point of the contact interface:

$$\|\boldsymbol{t}_\text{t}\| - \mu_\text{c}\, |p_\text{c}| \leq 0, \tag{2.50a}$$

$$\boldsymbol{v}_\text{t, rel} + \alpha\, \boldsymbol{t}_\text{t} = \boldsymbol{0}, \tag{2.50b}$$

$$(\|\boldsymbol{t}_\text{t}\| - \mu_\text{c}\, |p_\text{c}|)\ \alpha = 0, \tag{2.50c}$$

where $\mu_\text{c} \geq 0$ is the *friction coefficient* between the tire and the road, $\alpha \geq 0$ a scalar parameter and $\boldsymbol{t}_\text{t} = (\boldsymbol{I} - \boldsymbol{n}_\text{c} \otimes \boldsymbol{n}_\text{c}) \cdot \boldsymbol{t}_\text{c}$ the tangential contact traction. These constraints can also be physically interpreted starting with the so-called *slip condition* given by (2.50a), which states that the two contacting surfaces will slide as soon as the magnitude of the frictional stress is at least equal to a fraction of the contact pressure controlled by the friction coefficient, i.e. the frictional capacity

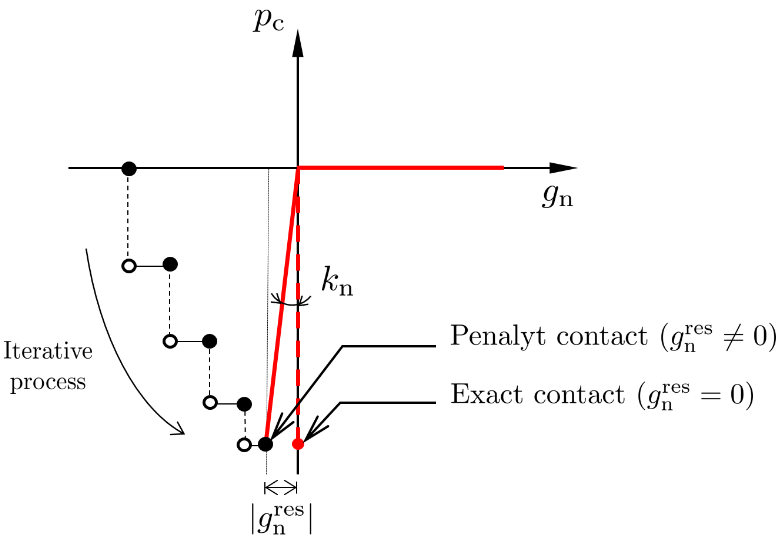
p_c
g_n
k_n
Iterative process
Penalyt contact ($g_n^{res} \neq 0$)
Exact contact ($g_n^{res} = 0$)
$|g_n^{res}|$

Figure 2.4: Schematic representation of the iterative process enforcing the contact constraint in a penalty contact method: a normal contact pressure p_c is applied on slave nodes as a function of the penalty stiffness parameter k_n and the normal gap g_n in order to close the latter, leaving a residual penetration gap g_n^{res} after a finite number of increments. In an exact contact enforcement, no residual penetration remains between slave and master surfaces.

$\mu_c\,|p_c|$. The second condition (2.50b) allows to introduce the *stick condition*: when $\alpha = 0$, both surfaces stick together so that relative tangential velocity between the two bodies given by $\boldsymbol{v}_{t,\,rel}$ is set to zero. However, when $\alpha > 0$, the two bodies slide on each other at a relative tangential velocity proportional to the frictional stress $\boldsymbol{t}_t$. Finally, the last equality (2.50c) simply allows to separate the solution of the stick and the slip conditions. Note that more advanced tire-road friction models can be used to better reproduce available laboratory data characterizing a given rubber compound. As for the tire-rim contact, the tire is not allowed to slide on the rim, so that the two bodies are enforced to stick together during the entire computation, i.e. $\alpha = 0$ at all times.

In practice, the penalty method works in two steps. In a fist step, a search algorithm is used to look for slave nodes that penetrate the rigid surface. In a second step, a normal contact pressure p_c is applied to oppose penetration to the found slave node, while respecting the conditions (2.49) and guaranteeing equal and opposite forces on the rigid surface at the penetration point. At the same time, an eventual frictional slip distance g_t is added in the direction of the relative tangential velocity vector, if permitted by the conditions given by (2.50). Simply speaking, since the contact pressure p_c depends on the gap distance g_n, a relation in the form of $p_c = -k_n\, g_n$ can be written for each slave node. Consequently the parameter k_n, that has the dimensions of a stiffness, must be properly chosen in order to position the slave node on the master surface.

The choice of this *penalty stiffness* parameter represents the largest drawback of the penalty methods as no universal rule can be established for its determination. In the current approach, a penalty stiffness based on a representative stiffness of the underlying elements forming the slave surface is used to approximate the contact enforcement. As an exact value of the penalty stiffness cannot be predicted, an iterative process is used to enforce the required constraints and obtain a satisfactory distance between the node and the surface as illustrated in Figure 2.4. Note that this procedure introduces an additional stiffness behavior into the model, such that the resulting "spring" stiffness of the concerned elements can influence the stable time increment, as discussed in section 2.1.4.

2.2 Fluid mechanics

In contrast to the field of solid mechanics where the vast majority uses exclusively the finite element method for the analysis of deformable bodies, various approaches are commonly employed to solve numerically the fluid flow. The preferred approaches in the field of *computational fluid dynamics* (CFD) used to spatially discretize the fluid field can be separated into two categories: continuum-based approaches that use a mesh to discretize the fluid domain in space, and mesh-free or particle approaches for which the approximation of the governing equations is constructed entirely in terms of nodes without relying on a mesh. The most common continuum methods include the *finite volume* (FVM), the *finite difference* (FDM) and the *finite element method* (FEM), whereas the *smooth particle hydrodynamics* (SPH) and the *molecular dynamics* (MD) methods are among the most popular mesh-free techniques. The current approach is based on the FVM and will be described in this section.

The literature offers a great amount of material that presents fluid flow problems and their numerical resolution. Considering first continuum-based approaches, the interested reader is for instance referred to the textbook by Patankar [140], who is a pioneer of modern CFD and in particular of the FVM, being among others the co-developer of the famous *SIMPLE* algorithm also described in the book. Blazek [18] and Versteeg [178] present more recent advancements for the FVM, while Ferziger and Perić propose a detailed introduction to the FDM and the FVM in [62]. The FEM is also a popular choice among the CFD community especially for solving FSI problems as both fluid and solid sub-domains are discretized using the same formulation, which allows for instance to construct a monolithic solver as proposed e.g. in the thesis by Mayr [117]. An extensive analysis of the FEM for fluid flow problems can be found in Donea and Huerta [39], Zienkiewicz et al. [197] or in the theses of Förster [64] and Wall [181]. Although these mesh-based methods are dominant in numerical simulations, some difficulties resulting from the use of a mesh can limit their application, such as in problems with free-surface, extremely large deformations or crack propagations [109]. In such cases, mesh-free methods can provide an attractive alternative to solve the fluid system. One of the most widespread approach is the SPH method in which the system is described by a set of particles, each one representing a finite volume in continuum scale and interacting with one another within a range controlled by a weight function [13]. These particles carry the fluid characteristic properties such as position, mass, velocity, etc., and since they are explicitly associated with different materials, the interface between phases is naturally tracked. However, mesh-free methods need a large number of particles to produce equivalent resolution compared to continuum-based techniques which, associated to a large computational cost, represents one of their main drawbacks [158]. Another problem emerges as particles cluster in some region of the flow, which in turn leads to an insufficient particle resolution in some other region so that mesh-free methods can suffer from an inaccurate representation of the interface comparing to grid-based Lagrangian methods [158]. An overview of the SPH method and recent advances in the field are proposed by e.g. Liu and Liu in [109] and Monaghan in [128], see also references therein. In the MD method, the particles are considered at a nano-scale level and represent atoms or molecules, see e.g. [68]. The MD method is applied in the fields of chemistry, chemical physics or biology, where the dynamics of each molecule or atom is to be resolved. The interested reader is referred to the textbook by e.g. Li and Liu [108] for more details on mesh-free particle methods such as SPH and MD. The SPH method can also be used for solving FSI problems as shown e.g. by Stasch et al. in [163], which have developed a SPH-FEM approach to compute the coupled system. For the sake of completeness, another CFD approach that has gained interest in the last few decades should be mentioned: the *Lattice Boltzmann method* (LBM), see e.g. the textbooks by Succi [164] or Wolf-Gladrow [188] for a complete introduction. Also in this case, the LBM does not rely on the Navier-Stokes equations to solve the fluid field but rather describes it by means of fictitious particles moving and colliding on a regular mesh. Based on the *kinetic gas theory*, this approach can also be employed for solving FSI problems, as in [70] where it is coupled to a FE solver to compute the solid sub-domain. Its use is however to this day quite limited compared to the other methods.

The choice of computational model used to handle the tire hydroplaning problem is first described in this section. The equations governing the fluid flow are then presented together with appropriate discretization techniques. Finally, a specificity of the fluid solver used for tire hydroplaning applications, which treats multi-phase flows and the corresponding free-surface is briefly introduced.

2.2.1 Choice of computational model and hypotheses

The fluid of interest in the current FSI application is liquid water that flows with a relative velocity with respect to the tire ranging from 0 up to 350 km/h, while considering the extreme case of the wet performance of racing tires. In these conditions, the fluid flow remains immutably incompressible and is thus governed by the incompressible Navier-Stokes equations. Furthermore, the fluid is considered as *Newtonian*, viscous and no heat exchanges are taken into account. These hypotheses are kept in the rest of the thesis.

As already stated in section 1.4.1, Eulerian fixed-grid approaches are attractive for discretizing the fluid sub-domain in FSI applications with large motion of the interface, as in tire hydroplaning simulations. A FVM combined with appropriate techniques to capture the motion of any obstacles and moving boundaries within the fluid field, including a fluid-structure interface, are therefore adopted in this work. The computational mesh is based on a *co-located structured Cartesian grid* with dynamic adaptive mesh refinement (AMR) allowing to deal with the motion of boundary layers and improve the quality of the mesh where needed. The treatment of the free-surface is based on a *volume of fluid* (VOF) method to evaluate the fraction of liquid water in each cell intersected by the free-surface. Its reconstruction is performed using the same technique as for the moving boundary. An implicit time integration scheme is implemented in the fluid solver to solve the Navier-Stokes equations whereas an explicit operator is used for the VOF method introducing therefore a critical time step size and making the overall fluid solution procedure eventually explicit.

An overview of the corresponding implementation in the fluid solver employed in this thesis is given in the following. Particular attention is paid to the computation of fluid quantities along a moving boundary immersed within the fluid domain, which can be the fluid-structure interface.

2.2.2 Governing equations in fluid mechanics

2.2.2.1 Kinematics

Contrary to solid mechanics where the body motion is described with respect to a reference configuration in a Lagrangian formulation, the Eulerian approach adopted for the fluid field describes kinematic quantities based on a unique current configuration $\Omega^{\mathcal{F}} \subset \mathbb{R}^3$. In this case, the primary kinematic unknowns correspond to the fluid velocity field $\boldsymbol{u}^{\mathcal{F}}(\boldsymbol{x}^{\mathcal{F}}, t)$. Considering that in an Eulerian formulation velocities are given in terms of the current coordinates $\boldsymbol{x}^{\mathcal{F}}$, the accelerations are defined using the material derivative as

$$\dot{\boldsymbol{u}}^{\mathcal{F}}(\boldsymbol{x}^{\mathcal{F}}, t) = \frac{\mathrm{D}\boldsymbol{u}^{\mathcal{F}}}{\mathrm{D}t} = \left.\frac{\partial \boldsymbol{u}^{\mathcal{F}}}{\partial t}\right|_{\boldsymbol{x}^{\mathcal{F}}} + \boldsymbol{u}^{\mathcal{F}} \cdot \boldsymbol{\nabla} \boldsymbol{u}^{\mathcal{F}}, \tag{2.51}$$

which, unlike accelerations (2.4) in the solid domain, contain a nonlinear convective term. The *strain-rate tensor* is also directly given as the symmetric gradient of $\boldsymbol{u}^{\mathcal{F}}$ and reads

$$\boldsymbol{\varepsilon}^{\mathcal{F}}(\boldsymbol{u}^{\mathcal{F}}) = \frac{1}{2}\left(\boldsymbol{\nabla}\boldsymbol{u}^{\mathcal{F}} + \left(\boldsymbol{\nabla}\boldsymbol{u}^{\mathcal{F}}\right)^T\right). \tag{2.52}$$

2.2.2.2 Balance equations

Similarly to the solid domain, fluid dynamics problems can be described with use of balance equations. In order to derive a mathematical formulation describing completely the fluid flow, i.e. the Navier-Stokes equations, not only the balance of linear momentum but also the mass conservation must be established, as shown in the following. The conservation of energy as well as multiple forms of the balance equations of momentum can be found e.g. in Ferziger and Perić [62].

Conservation of mass

Considering an arbitrary fluid volume V surrounded by a boundary surface ∂V of outward unit normal $\boldsymbol{n}$, the rate of mass change within this fixed volume is equal to the rate of inflow through the boundary surface (without any mass source or sink), which reads

$$\frac{\partial}{\partial t}\int_V \rho^{\mathcal{F}}\,\mathrm{d}V = -\int_{\partial V} \rho^{\mathcal{F}}\,\boldsymbol{u}^{\mathcal{F}}\cdot\boldsymbol{n}\,\mathrm{d}S = -\int_V \boldsymbol{\nabla}\cdot(\rho^{\mathcal{F}}\boldsymbol{u}^{\mathcal{F}})\,\mathrm{d}V, \tag{2.53}$$

where $\rho^{\mathcal{F}}$ is the *fluid mass density* and the *divergence theorem* has been used to write the last equality. Since V is arbitrary, the *local form of the mass conservation* valid everywhere in the fluid domain $\Omega^{\mathcal{F}}$ and $\forall t \in [0,T]$ can be written as

$$\frac{\partial\rho^{\mathcal{F}}}{\partial t} + \boldsymbol{\nabla}\cdot(\rho^{\mathcal{F}}\boldsymbol{u}^{\mathcal{F}}) = 0 \quad \text{in } \Omega^{\mathcal{F}}\times[0,T], \tag{2.54}$$

and, after neglecting compressibility effects, further simplifies into

$$\boldsymbol{\nabla}\cdot\boldsymbol{u}^{\mathcal{F}} = 0 \quad \text{in } \Omega^{\mathcal{F}}\times[0,T], \tag{2.55}$$

which is also commonly referred to as the *continuity equation*.

Balance of linear momentum

The balance of linear momentum accounting for the dynamic equilibrium of all forces in the fluid domain is given, in its *local convective form*, by

$$\rho^{\mathcal{F}}\frac{\mathrm{D}\boldsymbol{u}^{\mathcal{F}}}{\mathrm{D}t} = \boldsymbol{\nabla}\cdot\boldsymbol{\sigma}^{\mathcal{F}} + \bar{\boldsymbol{b}}^{\mathcal{F}} \text{ in } \Omega^{\mathcal{F}}\times[0,T], \tag{2.56}$$

which, by inserting (2.51), yields

$$\rho^{\mathcal{F}}\left(\frac{\partial\boldsymbol{u}^{\mathcal{F}}}{\partial t} + \boldsymbol{u}^{\mathcal{F}}\cdot\boldsymbol{\nabla}\boldsymbol{u}^{\mathcal{F}}\right) = \boldsymbol{\nabla}\cdot\boldsymbol{\sigma}^{\mathcal{F}} + \bar{\boldsymbol{b}}^{\mathcal{F}} \text{ in } \Omega^{\mathcal{F}}\times[0,T], \tag{2.57}$$

where $\bar{\boldsymbol{b}}^f$ are prescribed external body forces (per unit volume).

The *Cauchy stress tensor* denoted by $\boldsymbol{\sigma}^{\mathcal{F}}$ in (2.57) can be decomposed into a normal and shear component as

$$\boldsymbol{\sigma}^{\mathcal{F}}(\boldsymbol{u}^{\mathcal{F}}, p) = -p\,\boldsymbol{I} + \boldsymbol{\tau}^{\mathcal{F}}, \tag{2.58}$$

where p denotes the fluid *thermodynamic pressure* and $\boldsymbol{\tau}^{\mathcal{F}}$ the *shear stress tensor*. The latter is function of the strain-rate tensor and can be expressed in the case of a Newtonian fluid as the linear expression

$$\boldsymbol{\tau}^{\mathcal{F}} = 2\,\mu^{\mathcal{F}}\,\boldsymbol{\varepsilon}^{\mathcal{F}}(\boldsymbol{u}^{\mathcal{F}}), \tag{2.59}$$

where $\mu^{\mathcal{F}}$ represents the *fluid dynamic viscosity*.

Inserting (2.58) into (2.57) then yields

$$\rho^{\mathcal{F}}\left(\frac{\partial \boldsymbol{u}^{\mathcal{F}}}{\partial t} + \boldsymbol{u}^{\mathcal{F}} \cdot \boldsymbol{\nabla}\boldsymbol{u}^{\mathcal{F}}\right) = -\boldsymbol{\nabla}p + \boldsymbol{\nabla}\cdot\boldsymbol{\tau}^{\mathcal{F}} + \bar{\boldsymbol{b}}^f \;\text{ in }\; \Omega^{\mathcal{F}} \times [0,T]. \tag{2.60}$$

The transient Navier-Stokes equations of an incompressible flow can finally be expressed using the continuity equation (2.55) and the balance of linear momentum (2.60) as

$$\boldsymbol{\nabla}\cdot\boldsymbol{u}^{\mathcal{F}} = 0 \quad \text{in } \Omega^{\mathcal{F}} \times [0,T], \tag{2.61a}$$

$$\rho^{\mathcal{F}}\left(\frac{\partial \boldsymbol{u}^{\mathcal{F}}}{\partial t} + \boldsymbol{u}^{\mathcal{F}} \cdot \boldsymbol{\nabla}\boldsymbol{u}^{\mathcal{F}}\right) = -\boldsymbol{\nabla}p + \mu^{\mathcal{F}}\,\boldsymbol{\nabla}\cdot\left(\boldsymbol{\nabla}\boldsymbol{u}^{\mathcal{F}}\right) + \bar{\boldsymbol{b}}^{\mathcal{F}} \quad \text{in } \Omega^{\mathcal{F}} \times [0,T], \tag{2.61b}$$

where constant viscosity was assumed and equations (2.59) with (2.52) were used. It forms a nonlinear system of partial differential equations for which the primary unknowns are the fields of fluid velocity $\boldsymbol{u}^{\mathcal{F}}$ and of thermodynamic pressure p. For the sake of simplicity, the latter will simply be referred to as *fluid pressure* in the rest of the thesis.

2.2.2.3 IBVP for Navier-Stokes equations of incompressible flow

In the same way as for the solid field, the fluid initial boundary value problem is built based on the Navier-Stokes equations given by (2.61) and appropriate initial and boundary conditions in a time interval $t \in [0,T]$.

The initial conditions at time $t = 0$ are given by a divergence free initial velocity field $\bar{\boldsymbol{u}}_0^{\mathcal{F}}(\boldsymbol{x}^{\mathcal{F}})$ as

$$\boldsymbol{u}^{\mathcal{F}}(\boldsymbol{x}^{\mathcal{F}}, 0) = \bar{\boldsymbol{u}}_0^{\mathcal{F}}(\boldsymbol{x}^{\mathcal{F}}) \quad \text{with} \quad \boldsymbol{\nabla}\cdot\bar{\boldsymbol{u}}_0^{\mathcal{F}} = 0 \qquad \forall \boldsymbol{x}^{\mathcal{F}} \in \Omega^{\mathcal{F}}. \tag{2.62}$$

For incompressible flows, no initial conditions must be set as such on the pressure field. However, since pressure appears as a gradient $\boldsymbol{\nabla} p$ in equations (2.61), it is determined up to a constant. This pressure constant is imposed either as a mean pressure value or at a specific point in the fluid domain or even implicitly by fixing a pressure value at a boundary of the fluid domain [197]. In this work, the first of these three options is employed to impose a far-field pressure value of reference $\bar{p}^{\infty}$ for the whole duration of the simulation.

In order to define boundary conditions on the fluid domain $\Omega^{\mathcal{F}}$, the fluid boundary surface $\partial\Omega^{\mathcal{F}}$ is partitioned into non-overlapping *Dirichlet* and a *Neumann boundaries* denoted respectively by $\Gamma_{\mathrm{D}}^{\mathcal{F}}$ and $\Gamma_{\mathrm{N}}^{\mathcal{F}}$ such that

$$\partial\Omega^{\mathcal{F}} = \Gamma_{\mathrm{D}}^{\mathcal{F}} \cup \Gamma_{\mathrm{N}}^{\mathcal{F}}, \quad \Gamma_{\mathrm{D}}^{\mathcal{F}} \cap \Gamma_{\mathrm{N}}^{\mathcal{F}} = \emptyset, \tag{2.63}$$

away from the fluid-structure interface. Multiple types of *Dirichlet* and *Neumann boundary conditions* exist in order to model the fluid flow at its boundaries. Some of them, which are used in this work, are described in the following:

– *No-slip wall boundary conditions* are chosen when viscosity is taken into account in the Navier-Stokes equations. All fluid particles at this boundary are assumed to stick to the wall and eventually follow its motion as

$$\boldsymbol{u}^{\mathcal{F}} = \begin{cases} \boldsymbol{0} & \text{fixed wall} \\ \bar{\boldsymbol{u}}_{\text{wall}} & \text{moving wall} \end{cases} \quad \text{on } \Gamma_{\mathrm{D}}^{\mathcal{F}} \times [0,T], \tag{2.64}$$

where both the normal and tangential velocity components are prescribed. It typically causes a so-called *boundary layer* in a region close to the wall, where the flow features large velocity gradients and is dominated by internal friction. A fine computational mesh is required to properly resolve the physics of the flow in this thin region from where phenomena such as *separation* or *laminar-turbulent transition* originate. More information on the boundary layer, its theory and application to fluid mechanics can be found e.g. in the reference textbook by Schlichting and Gersten [155].

– *Slip wall* or *symmetry boundary conditions* only prescribe the normal velocity component as

$$\boldsymbol{u}^{\mathcal{F}} \cdot \boldsymbol{n} = \begin{cases} 0 & \text{fixed wall} \\ \bar{\boldsymbol{u}}_{\text{wall}} \cdot \boldsymbol{n} & \text{moving wall} \end{cases} \quad \text{on } \Gamma_{\mathrm{D}}^{\mathcal{F}} \times [0,T]. \tag{2.65}$$

This boundary can be interpreted as a wall with no friction and on which fluid particles can bounce without loosing momentum in the process.

– *Inlet* and*free-outlet boundary conditions* are the most challenging since they do not represent actual physical boundary conditions. The conceptual domain is truncated to a bounded one by introducing artificial boundaries on which appropriate boundary conditions corresponding to far-field values of velocity or pressure must be chosen. These artificial boundaries must be set sufficiently far away from eventual obstacles located or moving within the fluid domain in order to avoid any disturbance of the flow. The ones employed in this work prescribe a *normal mass velocity* at the inlet and a pressure at the outlet by imposing respectively

$$\rho^{\mathcal{F}}\,\boldsymbol{u}^{\mathcal{F}}\cdot\boldsymbol{n} = \rho^{\mathcal{F}}\,\bar{U}^{\mathcal{F}}_{\text{in}}(t) \qquad \text{on } \Gamma^{\mathcal{S}}_{\text{D}}\times[0,T], \tag{2.66}$$

$$p = \bar{p}_{\text{out}} \qquad \text{on } \Gamma^{\mathcal{S}}_{\text{N}}\times[0,T], \tag{2.67}$$

where $\bar{U}^{\mathcal{F}}_{\text{in}}(t)$ is a velocity magnitude not necessarily constant in time prescribed at the inlet and $\bar{p}_{\text{out}}$ an imposed pressure at the outlet. Note that the latter should be chosen in order to avoid creating a pressure gradient over the computational domain. In practice, a pressure value equal to the reference pressure $\bar{p}^{\infty}$ previously introduced is used. These types of artificial boundary conditions and their mathematical formulation are further discussed e.g. in the paper by Heywood et al. [84].

For further information on boundary conditions and additional definitions such as *rough wall* or *periodic boundary conditions*, the interested reader is referred to e.g. Versteeg [178].

The Navier-Stokes equations (2.61) with the initial and boundary conditions (2.62-2.67) complete the IBVP and form the *strong form of the Navier-Stokes equations of an incompressible flow* valid at all times and for all fluid particles within the fluid domain $\Omega^{\mathcal{F}}$. It is used as a starting point to derive the finite volume approximation of the fluid system.

2.2.3 Discretization of the Navier-Stokes equations with moving boundaries

In order to be solved numerically, the fluid IBVP defined at a continuous level must be transformed into a set of algebraic equations. As previously stated, multiple methods are commonly used to discretize the governing equations (2.61). In this work, a solver based on the FVM is employed, which also corresponds to the main trend followed nowadays by the scientific and industrial software community.

The formulation of the fluid flow problem with the FVM approximates the continuous Navier-Stokes equations on the fluid domain $\Omega^{\mathcal{F}}$ by integrating them over a finite number of so-called *control volumes* V^c constituting the entire fluid computational mesh as

$$\Omega^{\mathcal{F}} \approx \Omega^{\mathcal{F}}_h = \bigcup_{c=1}^{\text{N}^{\mathcal{F}}_{\text{cells}}} V^c, \qquad \bigcap_{c=1}^{\text{N}^{\mathcal{F}}_{\text{cells}}} V^c = \emptyset, \tag{2.68}$$

where $c \in \mathbb{N}$ is the index related to a fluid *cell* and $\mathrm{N}^{\mathcal{F}}_{\text{cells}}$ represents the number of fluid cells in the discretized fluid domain $\Omega^{\mathcal{F}}_{h}$. When *moving boundaries* are considered within the fluid domain, the motion of these boundaries, corresponding e.g. to a fluid-structure interface or a moving rigid body[10], must be treated appropriately with non-body-conforming approaches such as *immersed boundary methods* already reviewed in section 1.3.1. It is achieved by considering the fluid volume change during one time step $\Delta t^{\mathcal{F}}_{n+1} = t^{\mathcal{F}}_{n+1} - t^{\mathcal{F}}_{n}$ in the Navier-Stokes equations for each cell intersected by a moving body. In doing so, the discretization (2.68) is used to integrate equations (2.61) over each control volume V^c of external surface S^c during $\Delta t^{\mathcal{F}}_{n+1}$ as

$$\frac{1}{\Delta t^{\mathcal{F}}_{n+1}} \left(\int_{V^c_{n+1}} \rho^{\mathcal{F}} \,\mathrm{d}V - \int_{V^c_{n}} \rho^{\mathcal{F}} \,\mathrm{d}V \right) + \oint_{S^c_{n+1}} \rho^{f} \boldsymbol{u}^{\mathcal{F}} \cdot \boldsymbol{n} \,\mathrm{d}S = 0, \tag{2.69a}$$

$$\begin{aligned} \frac{1}{\Delta t^{\mathcal{F}}_{n+1}} \left(\int_{V^c_{n+1}} \rho^{\mathcal{F}} \boldsymbol{u}^{\mathcal{F}} \,\mathrm{d}V - \int_{V^c_{n}} \rho^{\mathcal{F}} \boldsymbol{u}^{\mathcal{F}} \,\mathrm{d}V \right) + \oint_{S^c_{n+1}} \rho^{\mathcal{F}} \boldsymbol{u}^{\mathcal{F}}\, \boldsymbol{u}^{\mathcal{F}} \cdot \boldsymbol{n} \,\mathrm{d}S \\ = - \int_{V^c_{n+1}} \boldsymbol{\nabla} p \,\mathrm{d}V + \oint_{S^c_{n+1}} \mu^{\mathcal{F}}\, \boldsymbol{\nabla}_{\boldsymbol{n}} \boldsymbol{u}\, \boldsymbol{n} \,\mathrm{d}S + \int_{V^c_{n+1}} \bar{\boldsymbol{b}}^{\mathcal{F}} \,\mathrm{d}V, \end{aligned} \tag{2.69b}$$

where the divergence theorem was used and the operator $\boldsymbol{\nabla}_{\boldsymbol{n}}(\cdot)$ denoting *directional derivative* defined as $\boldsymbol{\nabla}_{\boldsymbol{n}}(\cdot) = \begin{pmatrix} n_x\, \partial(\cdot)/\partial x \\ n_y\, \partial(\cdot)/\partial y \\ n_z\, \partial(\cdot)/\partial z \end{pmatrix}$ was introduced.

Compared to the strong form (2.61), equations (2.69) lead to the *weak form of the Navier-Stokes equations of an incompressible flow with moving boundaries* for which the balance equations and the boundary conditions are required to be satisfied only in an average (or integral) sense over an arbitrary number of control volumes. This weak form represents the basis of the FVM formulation introduced in section 2.2.4.

The two volumes V^c_n to V^c_{n+1} in (2.69) are used to describe the fluid volume change in a fluid cell c intersected by a moving boundary during $\Delta t^{\mathcal{F}}_{n+1}$ and can be further described as follows:

- V^c_n is referred to as *data volume* and corresponds to the volume of a fluid cell at the beginning of the time step $(t = t^{\mathcal{F}}_{n})$. At this moment, the volume as well as the distribution of all fluid quantities (data) are known on the whole fluid mesh.
- V^c_{n+1} is referred to as *geometrical volume* and corresponds to the volume of a fluid cell intersected by a moving boundary at the end of the time step $(t = t^{\mathcal{F}}_{n+1})$. At this point the volume is still an unknown, which must be estimated by an appropriate method before being able to advance the fluid field to the next time station $t^{\mathcal{F}}_{n+1}$.

[10] Since the fluid is considered in this section as a single-field problem, all moving boundaries are assumed to belong to a rigid undeformable body. As a matter of fact, in a partitioned coupled FSI problem, the fluid-structure interface is considered by the fluid solver as a rigid body during each fluid time step.

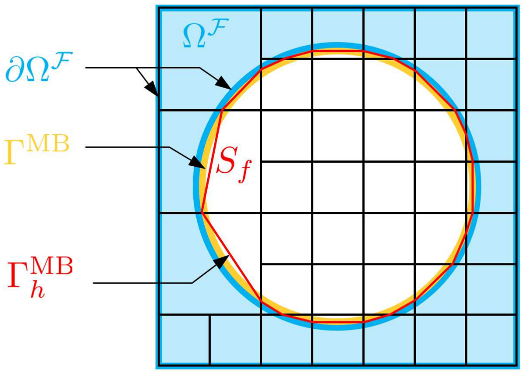

Figure 2.5: Schematic representation of a 2D fluid domain $\Omega^{\mathcal{F}}$ intersected by a moving boundary $\Gamma^{\mathrm{MB}} \subset \partial\Omega^{\mathcal{F}}$ associated with an immersed rigid body and approximated by a set Γ_h^{MB} of line edges (plane facets of surface S_{f} in 3D) according to (2.70). A better approximation of the external shape of the boundary obtained with use of a finer mesh is also highlighted.

To be able to evaluate the geometrical volume V_{n+1}^c in each intersected cell, the moving boundary associated with the moving body must be reconstructed within the fluid domain. The related reconstruction process is covered in the next section.

Moving boundary reconstruction

A moving boundary $\Gamma^{\mathrm{MB}} \subset \partial\Omega^{\mathcal{F}}$ associated with an immersed rigid body of arbitrarily complex shape is considered, as shown in Figure 2.5 for a 2D example. After the body motion, the moving boundary must be reconstructed within the fluid domain, which is achieved by approximating the curvilinear external surface of the considered body by a set of *plane facets* of surface S_f. The moving boundary can then be discretized as

$$\Gamma^{\mathrm{MB}} \approx \Gamma_h^{\mathrm{MB}} = \bigcup_{f=1}^{\mathrm{N}_{\text{facets}}^{\mathrm{MB}}} S_f, \qquad \bigcap_{f=1}^{\mathrm{N}_{\text{facets}}^{\mathrm{MB}}} S_f = \emptyset, \tag{2.70}$$

where $f \in \mathbb{N}$ is the facet index and $\mathrm{N}_{\text{facets}}^{\mathrm{MB}}$ represents the number of plane facets composing the discrete moving boundary Γ_h^{MB}. The number of facets must be large enough in order to accurately describe a curvinilear surface: the finer the mesh, the more accurate the approximation of the boundary shape will be (see also Figure 2.5). It is important to state here that in the case the moving boundary is the fluid-structure interface, its shape is already discretized as a set of plane facets, which correspond to the outputs of the solid FE computation. Hence no further approximation or geometrical error is introduced in that case by the fluid solver.

An appropriate IB method is then used to fit the fluid mesh to these facets and describe the updated shape of the fluid cells as well as their volume after the motion of the moving boundary. The corresponding algorithm is based on the following four steps [4]:

1. The fluid cells intersected by the facets of the moving boundary are found by an efficient search algorithm.
2. Among these cells, initial so-called "*parent*" rectangular cells are cut off and the resulting "*children*" cells intersected by the moving boundary are detached in some volumes of arbitrary complex shape bounded by these facets and fluid grid cells, as illustrated in Figure 2.6.
3. All geometrical information necessary for approximating the governing equations is determined, i.e. the geometrical volume V_{n+1}^c of all new children cells, the surface of their faces as well as the distance from the boundaries to the center of mass of the cells and their neighbors. These quantities are further used in section 2.2.4.2 while approximating the surface and volume integrals in equations (2.69).
4. Small children cells referred to as *donor* cells, i.e. cells whose volume is lower than a threshold percentage of their parent cells' initial volume, are removed and their boundaries are transferred to the neighboring cells called *acceptor* cells. Acceptor cells receive the donors' geometrical information and related fluid data. In this joining process, the fluid velocity is mass averaged whereas the pressure is volume averaged as[11]

$$\boldsymbol{u}_{\text{A, joined}}^{\mathcal{F}} = \frac{\boldsymbol{u}_{\text{A}}^{\mathcal{F}}\, V_{\text{A}}^c\, \rho_{\text{A}}^{\mathcal{F}} + \boldsymbol{u}_{\text{D}}\, V_{\text{D}}^c\, \rho_{\text{D}}^{\mathcal{F}}}{V_{\text{A}}^c\, \rho_{\text{A}}^{\mathcal{F}} + V_{\text{D}}^c\, \rho_{\text{D}}^{\mathcal{F}}}, \tag{2.71a}$$

$$p_{\text{A, joined}} = \frac{p_{\text{A}}\, V_{\text{A}}^c + p_{\text{D}}\, V_{\text{D}}^c}{V_{\text{A}}^c + V_{\text{D}}^c}, \tag{2.71b}$$

   where the subscripts $(\cdot)_{\text{A}}$ and $(\cdot)_{\text{D}}$ designate acceptor and donor quantities respectively. In doing so, the resulting cells are not geometrically simplified and stored in the form of complex polyhedrons, where all information needed to solve the Navier-Stokes equations are available.

During the cell cutting process following a boundary motion, five situations can occur in each cell of the fluid grid. Each one of these five situations illustrated for a 2D example in Figure 2.7a is treated separately and can be summarized as follows:

1. The cell is far from the moving boundary and its volume does not change during $\Delta t_{n+1}^{\mathcal{F}}$ (blue cells in Figure 2.7a) : $V_{n+1}^c = V_n^c$. The classic Navier-Stokes equations (with a constant cell volume) can be solved.

[11] The time index $(n+1)$ is omitted here for the sake of clarity.

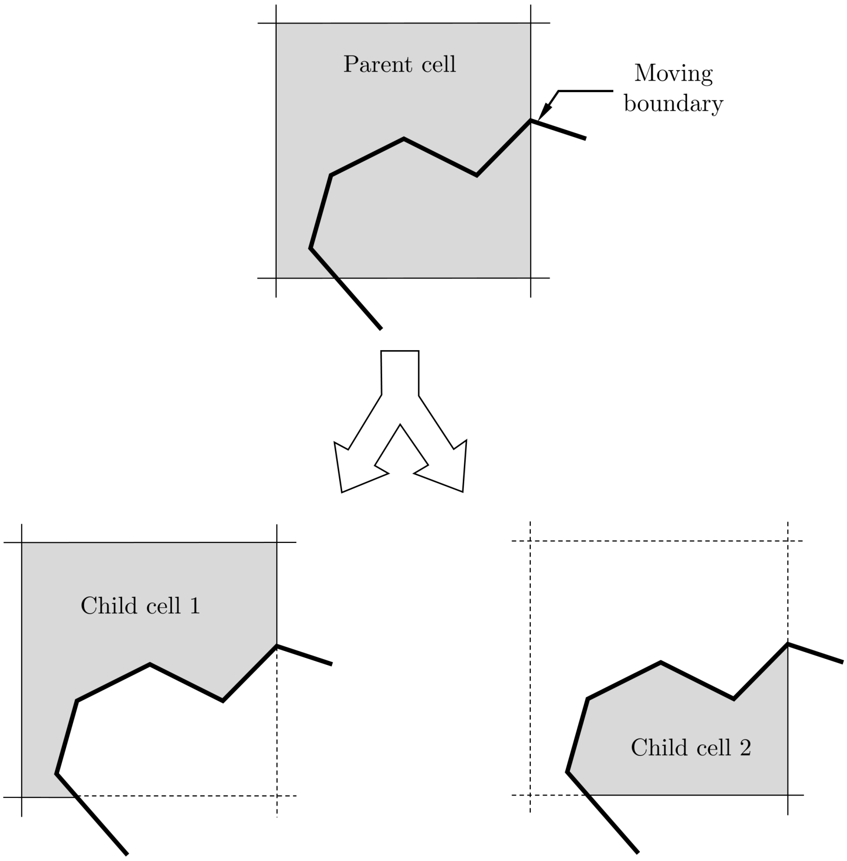

Figure 2.6: Cell cutting process inspired by [4] where a “parent” initial cell (top) is cut off by the facets of the moving boundary into “children” cells (bottom) on the underlying fixed-grid of finite volumes.

2. The boundary moves inside the cell and its fluid volume changes during $\Delta t^{\mathcal{F}}_{n+1}$ (yellow cells in Figure 2.7a) : $V^c_{n+1} > V^c_n$ or $V^c_{n+1} < V^c_n$ (see Figure 2.7b). Both volumes and related data must be considered to solve the fluid dynamics system of equations.

3. A new fluid cell is generated during $\Delta t^{\mathcal{F}}_{n+1}$ (green cells in Figure 2.7a) : $V^c_n = 0$ and $V^c_{n+1} \neq 0$. These cells are treated as small or donor cells as defined previously. The donor’s geometrical volume is shared between the acceptors, which see their geometrical volume increased by the donor cell’s contribution. The data volume of acceptor cells remains in this case unchanged.

4. The fluid cell disappears during $\Delta t^{\mathcal{F}}_{n+1}$ (red cells in Figure 2.7a) : $V^c_n \neq 0$ and $V^c_{n+1} = 0$. This kind of cell is treated in the same way as in the previous case by considering a donor and acceptor cells. The solid body sweeps away all the fluid contained in the considered cell, whose data must be transported to acceptor cells. The velocity of the moving boundary $\boldsymbol{u}^{\mathrm{MB}}_{n+1}$ is used to determine the direction in which the swept fluid data needs to be transferred and which cells will be chosen as acceptors. In this case, the donor’s

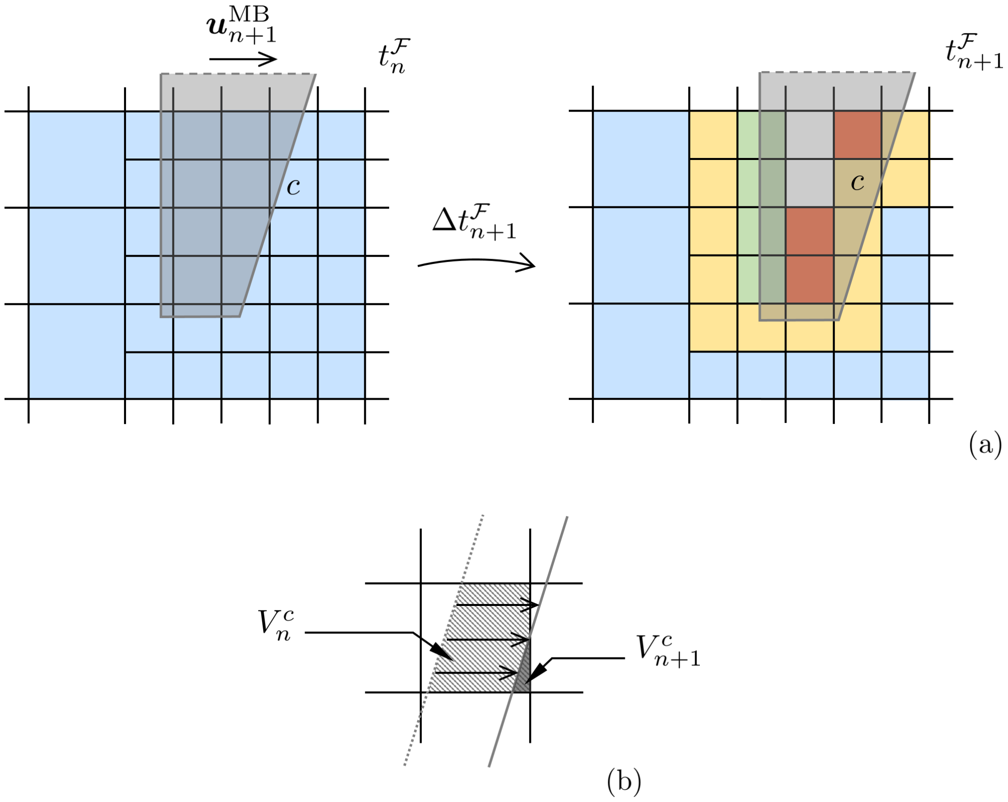

Figure 2.7: (a) A boundary immersed within a fluid fixed-grid and moving with a velocity $\boldsymbol{u}^{\mathrm{MB}}_{n+1}$ during a time step $\Delta t^{\mathcal{F}}_{n+1} = t^{\mathcal{F}}_{n+1} - t^{\mathcal{F}}_{n}$ can generate up to five types of fluid cells: cells far away from the moving boundary keep a constant fluid volume (blue), cells intersected by the boundary have a modified fluid volume (yellow), new cells are generated (green), old cells disappear (red) and cells remain completely overlapped by the solid (white). (b) Fluid volume changing from V^c_n (data volume) to V^c_{n+1} (geometrical volume) in a cell c during $\Delta t^{\mathcal{F}}_{n+1}$ after the motion of the boundary.

data volume is shared between the acceptor cells while the acceptors' geometrical volume receives no contribution of the donor cell.

5. The cell remains overlapped by the solid body during $\Delta t^{\mathcal{F}}_{n+1}$ (white cells in Figure 2.7a) : $V^c_n = V^c_{n+1} = 0$. These cells do not need any treatment.

Now that not only the data volume V^c_n but also the geometrical volume V^c_{n+1} and related geometrical information are known in each cell c of the computational mesh, the not yet discretized Navier-Stokes equations (2.69) can be computed to evaluate the fluid unknowns at the next time station $t^{\mathcal{F}}_{n+1}$. Their spatial discretization using the FVM and results from this section is presented next.

2.2.4 Space discretization

With an appropriate treatment of the moving boundaries, the Navier-Stokes equations can be discretized in space to form a system of numerically computable algebraic equations. As already mentioned, a structured Cartesian mesh is employed to divide the fluid continuum into a finite number of control volumes due to numerous advantages. Compared to unstructured or curvilinear computational meshes, Cartesian grids are easy to construct for any kind of fluid domain geometry, facilitate the reconstruction of moving fluid boundaries and perform well both in terms of numerical and hardware efficiency as reported e.g. by Mehl et al. in [120]. A fluid mesh can also be further described as a so-called *co-located* or *non-staggered* grid. In that case, both the primary fields of unknowns, the velocity $\boldsymbol{u}^{\mathcal{F}}$ and the pressure p, are stored and evaluated at the center of the cells, see e.g. Versteeg [178] for further details. Another approach consists in using *staggered* grids where the scalar quantities such as pressure and density are stored in cell centers whereas the velocity are located at the cell faces. Since a co-located grid is more efficient and more convenient for body fitted and moving boundaries, it is also adopted here. However, this grid arrangement requires splitting the evaluation of the pressure and the velocity as well as interpolating the pressure from the nodes onto the cell faces, which can introduce non-physical solutions for the pressure and corresponds to the well known "*checker-board*" problem. These spurious oscillations can eventually be suppressed by an appropriate treatment in the fluid procedure, which will be described in section 2.2.5 while presenting the successive steps for solving the fluid domain.

Before detailing the space discretization of equations (2.69), a few words are given about the grid adaptation procedure employed in the fluid solver.

2.2.4.1 Dynamic adaptive mesh refinement

The dynamic adaptivity of the structured Cartesian mesh implemented in the fluid solver is based on an AMR approach already reviewed in section 1.3.1. The entire mesh is composed of a collection of rectangular grids of different levels, where each cell can be subdivided into eight (in 3D) new cells in a lower refinement level. The resulting grid being stored as an octree, the method is very efficient while accessing computer memory. Compared to non-structured or structured curvilinear grids, the present method is faster and produces a mesh of good quality in the sense that all cells have a constant aspect ratio. Furthermore, the adaptive refinement approach is described as *dynamic* because it can automatically follow an immersed body within the fluid domain along its moving boundary or even refine the mesh in challenging regions of the flow, for instance characterized by a large pressure gradient. The fluid computational mesh and its adaptive refinement procedure are illustrated in Figure 2.8 for a 2D grid.

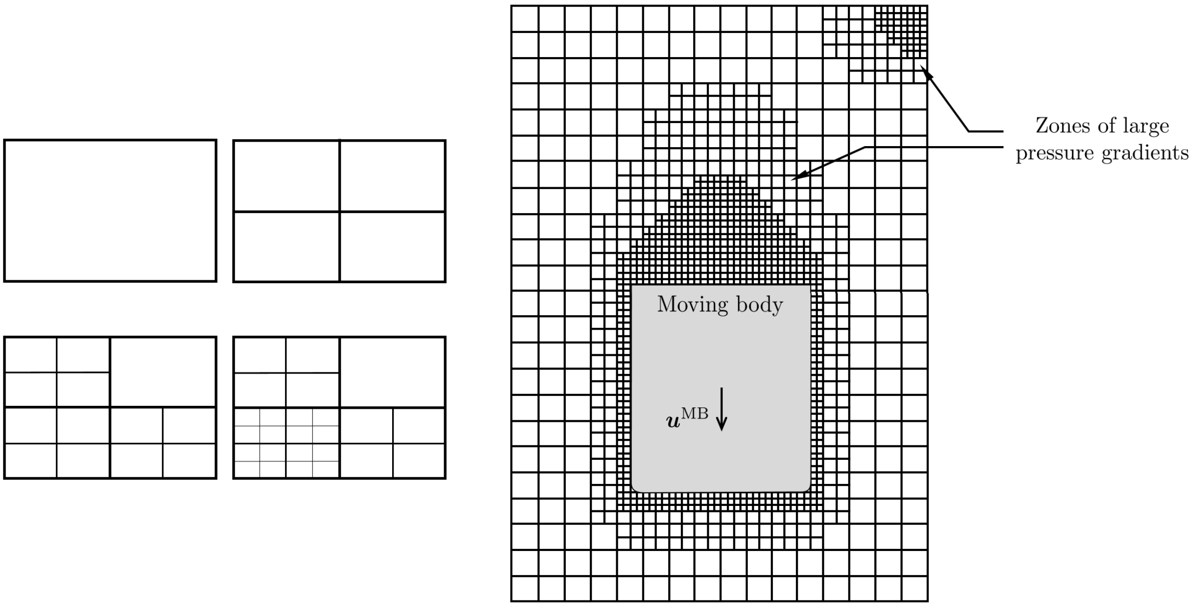

Figure 2.8: Refinement levels from 0 to 3 in the fluid computational mesh (left). Dynamic adaptive mesh refinement procedure following the moving boundary associated with a body traveling with a velocity $\boldsymbol{u}^{\mathrm{MB}}$ as well as in regions of the flow featuring large pressure gradients as in the wake of the body (right).

2.2.4.2 Finite volume formulation

The finite volume approximation of the Navier-Stokes equations is presented in the following by treating the surface and volume integrals in (2.69) separately, which eventually comes down to evaluating fluxes through cell faces in the case a co-located mesh is used.

Surface integrals approximation

Starting with the surface integrals, a flux F of an arbitrary scalar quantity f (e.g. velocity component) through a control volume V^c is considered. It can also be expressed as the flux of this same quantity through the closed surface S^c bounding the finite volume V^c. Accounting for this, the integrals over the faces of a finite volume can be approximated as follows[12]

$$\oint_{S^c} F \, \mathrm{d}S = \sum_{k=1}^{\mathrm{N}^{\mathcal{F}}_{\text{faces}}} \int_{S^c_k} F \, \mathrm{d}S \approx \sum_{k=1}^{\mathrm{N}^{\mathcal{F}}_{\text{faces}}} \mathsf{F}_k \, \mathsf{S}^c_k, \tag{2.72}$$

where S^c_k is the surface area of the k^{th} face, F_k the discrete averaged value of the flux F over the cell's face and $\mathrm{N}^{\mathcal{F}}_{\text{faces}} = 6$ in 3D (if no boundary intersects the cell).

[12] The notation $k \in \mathbb{N}$ refers to the face index within a fluid cell.

Two types of fluxes, a *convective flux* F^{conv} and a *diffusive flux* F^{diff} of a quantity f through the cell faces, corresponding respectively to a convective and diffusive transfer in equations (2.69), need to be approximated, which can be formulated as

$$\mathsf{F}^{\text{conv}} = \mathsf{f}\,\mathbf{u}^{\mathcal{F}} \cdot \mathbf{n}, \qquad\qquad \mathsf{F}^{\text{diff}} = \boldsymbol{\nabla}\mathsf{f} \cdot \mathbf{n}, \tag{2.73}$$

where discretized quantities were introduced. These two fluxes and in particular the approximation of the discrete scalar quantity f in these two cases are presented separately in the following.

Convective flux F^{conv}

To be able to approximate the convective flux F^{conv} during a time step Δt_{n+1}, the profile of the quantity f must first be reconstructed inside each computational cell[13]. To do so, a polynomial approximation based on averaged values at the center of the considered cell and its neighbors is carried out. More specifically, the solution of the variable f is reconstructed within the cell using a *monotone upwind scheme*, for which monotony is achieved by building a bi-linear profile of f in each spatial direction. The reconstruction procedure is illustrated in Figure 2.9a for a 1D example with a regular grid built along the x-axis as

$$\mathsf{x}_{i+1/2} = i\,\Delta\mathsf{x}, \tag{2.74}$$

where the index $i \in \mathbb{N}$ denotes nodal values at the center of a cell bounded by faces (points in 1D) of coordinate $\mathsf{x}_{i-1/2}$ and $\mathsf{x}_{i+1/2}$.

The bi-linear profile $f(x)$ used to reconstruct the solution inside the cell is defined as follows[14]:

$$f(x) = \begin{cases} \mathsf{f}_{i,\mathrm{r}} + \partial_{\mathsf{x}}\mathsf{f}_{i,\mathrm{r}}\,(\mathsf{x}_{i+1/2} - x) & \text{if} \quad x \geq \mathsf{x}_{i-1/2} + \lambda\,\Delta\mathsf{x}, \\ \mathsf{f}_{i,\mathrm{l}} + \partial_{\mathsf{x}}\mathsf{f}_{i,\mathrm{l}}\,(x - \mathsf{x}_{i-1/2}) & \text{if} \quad x < \mathsf{x}_{i-1/2} + \lambda\,\Delta\mathsf{x}, \end{cases} \tag{2.75}$$

where λ is a dimensionless distance to the junction point of the bi-linear profile defined by $\lambda = (\mathsf{f}_{i,\mathrm{r}} - \mathsf{f}_i)/(\mathsf{f}_{i,\mathrm{r}} - \mathsf{f}_{i,\mathrm{l}})$, $\mathsf{f}_{i,\mathrm{r}}$ and $\mathsf{f}_{i,\mathrm{l}}$ values reconstructed on the right and left faces of the i^{th} cell, and $\partial_{\mathsf{x}}\mathsf{f}_{i,\mathrm{r}}$ and $\partial_{\mathsf{x}}\mathsf{f}_{i,\mathrm{l}}$ the discrete derivations of these quantities given respectively by

$$\begin{aligned} \mathsf{f}_{i,\mathrm{r}} &= \frac{1}{2}(\mathsf{f}_{i+1} + \mathsf{f}_i) - \frac{1}{6}(\mathsf{f}_{i-1} - 2\,\mathsf{f}_i + \mathsf{f}_{i+1}), \\ \mathsf{f}_{i,\mathrm{l}} &= \frac{1}{2}(\mathsf{f}_i + \mathsf{f}_{i-1}) - \frac{1}{6}(\mathsf{f}_{i-1} - 2\,\mathsf{f}_i + \mathsf{f}_{i+1}), \end{aligned} \tag{2.76}$$

[13] For the sake of clarity, the superscript $(\cdot)^{\mathcal{F}}$ denoting fluid quantities is left out in the rest of this section.

[14] It is assumed that the profile $f(x)$ does not vary while it is advected by the flow during Δt_{n+1}.

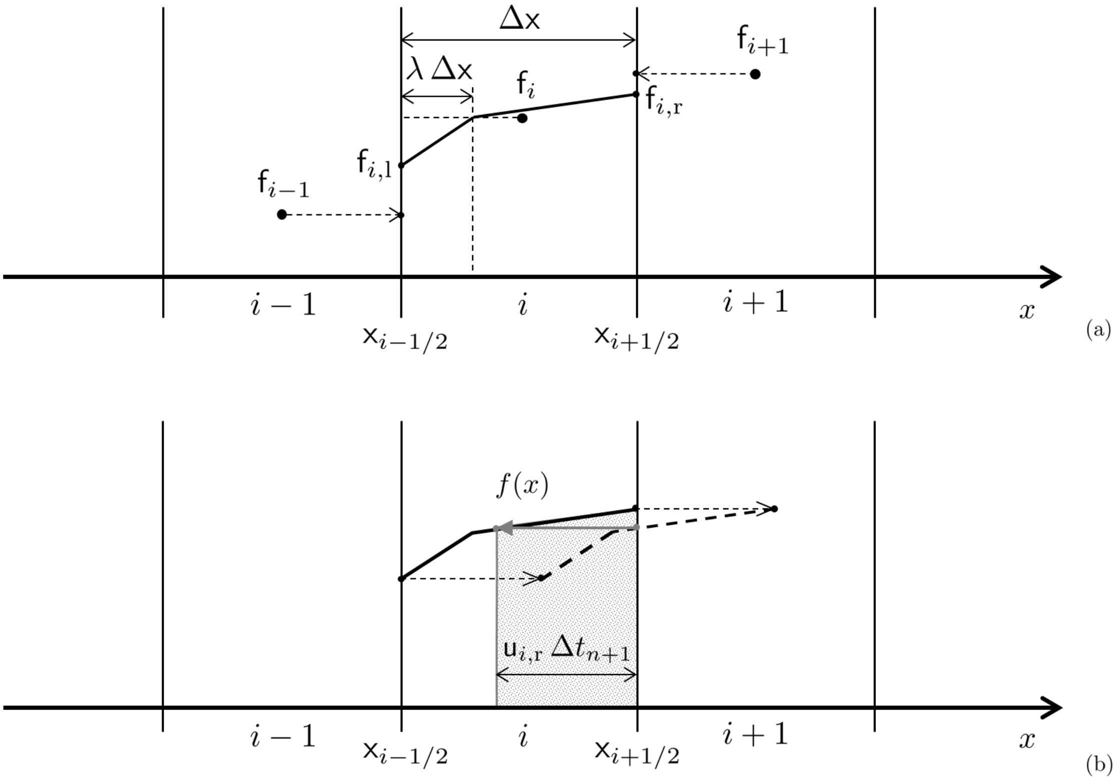

Figure 2.9: (a) Representation of the 1D monotone upwind (bi-linear) reconstruction of a quantity f inside a cell denoted by its center i (2nd order approximation), inspired by [3]. (b) 1D approximation of the convective flux $F^{\text{conv}}_{i,\text{r}}$ through the right face of cell i during a time step Δt_{n+1} achieved by integrating the reconstructed function $f(x)$ over the backward characteristic of length $u_x\,\Delta t_{n+1}$.

and

$$\begin{aligned}\partial_x f_{i,\text{r}} &= \frac{f_{i,\text{r}} - f_i}{(1-\lambda)\,\Delta x},\\ \partial_x f_{i,\text{l}} &= \frac{f_i - f_{i,\text{l}}}{\lambda\,\Delta x}.\end{aligned} \tag{2.77}$$

This reconstruction scheme provides a 2nd order approximation in space $\mathcal{O}(\Delta x^2)$ and has monotonous and small artificial dissipating properties [3]. The reconstruction algorithm switches to a 1st order scheme such that $f(x) = f_i$ in the case the following conditions are met: $\lambda < 0.01$, $\lambda > 0.99$, $|f_{i,\text{r}} - f_{i,\text{l}}| \leq 10^{-5}(|f_{i,\text{l}}| + |f_i| + |f_{i,\text{r}}|)$ or $(f_{i,\text{r}} - f_i)(f_i - f_{i,\text{l}}) \leq 10^{-10}(|f_{i,\text{l}}| + |f_i| + |f_{i,\text{r}}|)^2$. The values reconstructed on the cell faces are then evaluated as follows:

$$\begin{aligned} f_{i,\text{l}} &= f_{i-1}, & f_{i,\text{r}} &= f_i & &\text{if} \quad u_x > 0,\\ f_{i,\text{l}} &= f_i, & f_{i,\text{r}} &= f_{i+1} & &\text{if} \quad u_x < 0,\end{aligned} \tag{2.78}$$

where u_x is the velocity component normal to the cell face.

The convective flux of f through the right face of the i^{th} cell (Figure 2.9b) during Δt_{n+1} can now be written as

$$\mathsf{F}^{\text{conv}}_{i,\text{r}} = \int_{t_n}^{t_{n+1}} f(x)\ \mathsf{u}_{i,\text{r}}\ \mathrm{d}t. \tag{2.79}$$

By operating a variable change using the backward characteristics length $\mathsf{u}_{i,\text{r}}\,\Delta t_{n+1}$, it becomes

$$\mathsf{F}^{\text{conv}}_{i,\text{r}} = \int_{\mathsf{x}_{i+1/2}-\mathsf{u}_{i,\text{r}}\,\Delta t_{n+1}}^{\mathsf{x}_{i+1/2}} f(x)\ \mathrm{d}x \quad \text{with} \quad \mathsf{u}_{i,\text{r}} > 0. \tag{2.80}$$

In the context of the Navier-Stokes equations, the reconstructed function corresponds to the fluid velocity such that $f(x) = u(x)$. The expression of the convective flux through the k^{th} face of cell i during Δt_{n+1} thus reads

$$\mathsf{F}^{\text{conv}}_{i,k}\,(\mathsf{u}_{i,k},\,\mathsf{u}_i) = \int_{\mathsf{x}_{i+1/2}-\mathsf{u}_{i,k}\,\Delta t_{n+1}}^{\mathsf{x}_{i+1/2}} u(x)\ \mathrm{d}x \quad \text{with} \quad \mathsf{u}_{i,k} > 0, \tag{2.81}$$

where $\mathsf{u}_{i,k}$ is the velocity component normal to the k^{th} face and $u(x)$ the velocity function reconstructed inside the cell based on neighbor cells' center values as in (2.75). The 3D reconstruction scheme is simply built by superposing three 1D reconstruction schemes presented above in each spatial direction, which keeps the 2nd order approximation in space if the bi-linear reconstruction is used.

Diffusive flux F^{diff}

In order to approximate the diffusive flux of a variable f, the value of its gradient $\boldsymbol{\nabla}\mathsf{f}$ must be evaluated on the cell faces. The approximation of the gradient is presented in the following for an *unadapted* as well as a *non-uniform adapted grid* intersected by a moving boundary. A 2D grid defined similarly as in (2.74) in two spatial directions $\boldsymbol{x} = (x,\,y)$ is considered for clarity reasons, the generalization to the 3D form being straightforward.

In the case of an unadapted Cartesian grid as illustrated in Figure 2.10a, the discrete derivatives are evaluated by a *two point central difference scheme*. The gradient of f on the right and top faces of a cell designated by the index couple i,j is then given respectively by

$$\begin{aligned} \boldsymbol{\nabla}\mathsf{f}_{i,j}\big|_{\text{r}} &= \frac{\mathsf{f}_{i+1,j} - \mathsf{f}_{i,j}}{\Delta\mathsf{x}}\ \mathbf{n}_{\text{r}}, \\ \boldsymbol{\nabla}\mathsf{f}_{i,j}\big|_{\text{t}} &= \frac{\mathsf{f}_{i,j+1} - \mathsf{f}_{i,j}}{\Delta\mathsf{y}}\ \mathbf{n}_{\text{t}}, \end{aligned} \tag{2.82}$$

whereas the gradient on any other faces of the computational mesh is approximated in the same way.

For an adapted non-uniform Cartesian grid intersected by a solid boundary, the evaluation of the gradient corresponding to the diffusive flux is slightly more complex. The grid illustrated in Figure 2.10b is considered, in which capital letters denote positions with known values (at the center and boundaries of a cell) whereas small letters designate locations where the fluid quantities must be approximated. To do so, a linear interpolation is performed along a line crossing the face center and perpendicular to the cell face. Moreover, to remain general, some cells have been adapted and subdivided into four cells (in 2D) of equal size such that $\Delta x_{I} = 2\,\Delta x_{II}$ and $\Delta y_{I} = 2\,\Delta y_{II}$. The approximation of the gradient on two faces can then be given by:

- Right face of cell B:

$$\nabla f_{B_r} = \frac{1}{2}\left(\nabla f_d + \nabla f_b\right), \tag{2.83}$$

with

$$\nabla f_d = \frac{f_D - f_c}{x_D - x_c}\,\mathbf{n}_d = \frac{f_D - f_c}{\frac{\Delta x_I}{2} + \frac{\Delta x_{II}}{2}}\,\mathbf{n}_d, \qquad \nabla f_b = \frac{f_e - f_a}{x_e - x_a}\,\mathbf{n}_b,$$

where values f_c, f_e and f_a are approximated by a linear interpolation of center values. For example, the expression of f_c is given by

$$f_c = \frac{f_C\,(y_c - y_B) + f_B\,(y_C - y_c)}{y_C - y_B} = \frac{f_C\left(\frac{\Delta y_{II}}{2}\right) + f_B\left(\frac{\Delta y_I}{2} + \frac{\Delta y_{II}}{2}\right)}{\Delta y_I}.$$

- Lower face of the wall intersecting cell E, whose center is designated by W and where f_W corresponds to a known boundary condition:

$$\nabla f_{E_{W,1}} = \frac{f_f + f_W}{\overline{fW}}\,\mathbf{n}_{W,1}, \tag{2.84}$$

where $\overline{fW}$ is the distance between points f and W, and $\mathbf{n}_{W,1}$ the outward normal to the face.

When cells are deformed due to the presence of a moving boundary (e.g. cells E and F), the center of mass of the cell moves and the lines along which the linear interpolation is performed are not perpendicular to the initial cell faces anymore. In that case, intermediate values must be interpolated in additional locations on the grid (e.g. in e and f). These locations are defined by the intersection between lines joining cell centers (e.g. line segment EF) and lines perpendicular to cell faces crossing their center (e.g. line segments be or fW).

A generalization to a 3D approximation of the diffusive fluxes in the Navier-Stokes equations is carried out similarly by considering an additional spatial direction.

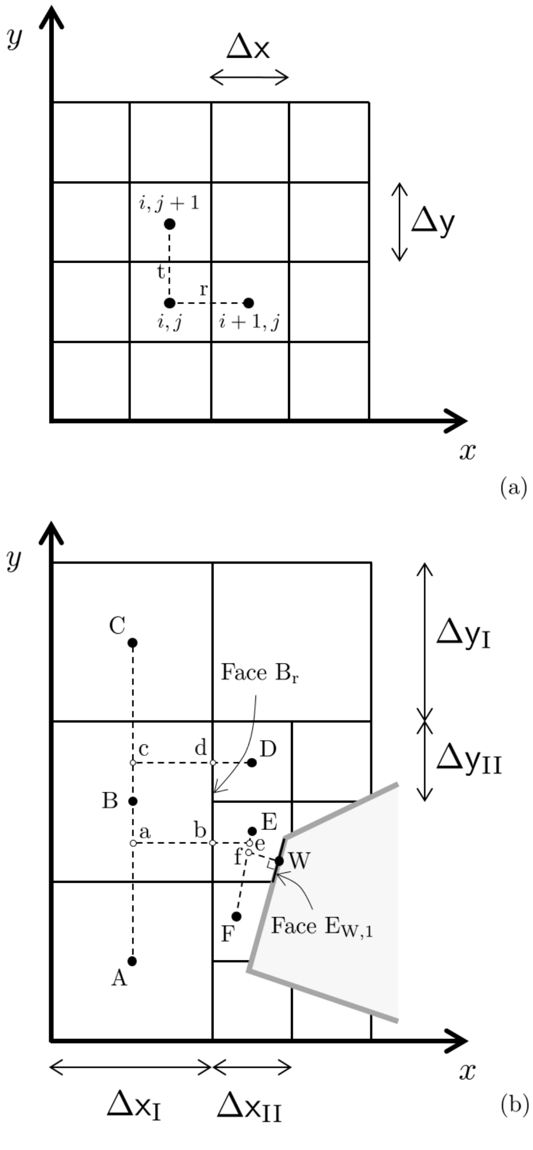

Figure 2.10: (a) Approximation of the gradient on the right and top faces of cell i, j in a 2D unadapted grid. (b) Approximation of the gradient on the right face of cell B (B_r) and on the lower face of the wall intersecting cell E ($E_{W,1}$) in a 2D non-uniform and adapted grid.

Volume integrals approximation

The volume integrals for a fluid cell c are approximated as follows

$$\int_{V^c} f \; \mathrm{d}V = \hat{f}\, V^c \approx \mathsf{f}_i\, \mathsf{V}^c, \tag{2.85}$$

where $\hat{f}$ is the averaged value of quantity f over the volume V^c and f_i is the discretized quantity at the center of that cell of finite volume V^c. The evaluation of the volume integrals is straightforward since it only involves quantities at cell centers and does not require any interpolation. The approximation is exact if f is constant or varies linearly within the control volume V^c and achieves 2nd order accuracy.

With the approximation of the surface and volume integrals presented in this section, the Navier-Stokes equations (2.69) are now fully discretized in space. The final discretized form is given in the next section while introducing the time discretization of the fluid system.

2.2.5 Time discretization

The time integration scheme implemented in the fluid solver is based on an *implicit pressure-velocity split* approach. Considering that the fluid variables at time $t = t^{\mathcal{F}}_n$ as well as both volumes V^c_n and V^c_{n+1} are known in each cell of the computational grid, the fluid field can now be advanced to the next time station $t^{\mathcal{F}}_{n+1} = t^{\mathcal{F}}_n + \Delta t^{\mathcal{F}}_{n+1}$. Introducing the spatial discretization from the previous section, the Navier-Stokes equations of an incompressible flow with moving boundaries given by (2.69) can be expressed for each cell c of the computational mesh as

$$\frac{\mathsf{V}^c_{n+1} - \mathsf{V}^c_n}{\Delta t^{\mathcal{F}}_{n+1}} + \sum_{k=1}^{\mathrm{N}^{\mathcal{F}}_{\text{faces}}} \mathbf{u}^{\mathcal{F}}_{n+1,k} \cdot \mathbf{n}_{n+1,k}\, \mathsf{S}^c_{n+1,k} = 0, \tag{2.86a}$$

$$\begin{aligned} &\frac{\mathbf{u}^{\mathcal{F}}_{n+1}\, \mathsf{V}^c_{n+1} - \mathbf{u}^{\mathcal{F}}_n\, \mathsf{V}^c_n}{\Delta t^{\mathcal{F}}_{n+1}} + \sum_{k=1}^{\mathrm{N}^{\mathcal{F}}_{\text{faces}}} \mathbf{F}^{\text{conv}}_k \left(\mathbf{u}^{\mathcal{F}}_{n,k}, \mathbf{u}^{\mathcal{F}}_{n+1}\right) \mathsf{S}^c_{n+1,k} \\ &\qquad = -\frac{\boldsymbol{\nabla} \mathsf{p}_{n+1}}{\rho^{\mathcal{F}}}\, \mathsf{V}^c_{n+1} + \nu^{\mathcal{F}} \sum_{k=1}^{\mathrm{N}^{\mathcal{F}}_{\text{faces}}} \boldsymbol{\nabla}_{\boldsymbol{n}_k} \mathbf{u}^{\mathcal{F}}_{n+1,k}\, \mathbf{n}_{n+1,k}\, \mathsf{S}^c_{n+1,k} + \frac{\bar{\mathbf{b}}^{\mathcal{F}}_n}{\rho^{\mathcal{F}}}\, \mathsf{V}^c_{n+1}, \end{aligned} \tag{2.86b}$$

where $\nu^{\mathcal{F}} = \mu^{\mathcal{F}} / \rho^{\mathcal{F}}$ is the *kinematic viscosity* and quantities with a subscript k correspond to values evaluated on cell faces. The reconstruction of the velocity gradients on the cell faces in the diffusive flux is performed using the 3D form of the operators given in (2.82-2.84) whereas the convective flux $\mathbf{F}^{\text{conv}}_k \left(\mathbf{u}^{\mathcal{F}}_{n,k}, \mathbf{u}^{\mathcal{F}}_{n+1}\right)$ is approximated as in (2.81).

In order to solve this fully discretized system of equations, classical pressure-velocity split methods assume a sequential solution of the momentum and pressure equations. By introducing an *intermediate velocity* field denoted by $\widetilde{\mathbf{u}}^{\mathcal{F}}_{n+1}$, equations (2.86b) can be rewritten as

$$\begin{aligned}\frac{\mathbf{u}^{\mathcal{F}}_{n+1}-\widetilde{\mathbf{u}}^{\mathcal{F}}_{n+1}}{\Delta t^{\mathcal{F}}_{n+1}}\,\mathsf{V}^{c}_{n+1}+\frac{\widetilde{\mathbf{u}}^{\mathcal{F}}_{n+1}\,\mathsf{V}^{c}_{n+1}-\mathbf{u}^{\mathcal{F}}_{n}\,\mathsf{V}^{c}_{n}}{\Delta t^{\mathcal{F}}_{n+1}}+\sum_{k=1}^{\mathrm{N}^{\mathcal{F}}_{\text{faces}}}\mathbf{F}^{\text{conv}}_{k}\left(\mathbf{u}^{\mathcal{F}}_{n,k},\mathbf{u}^{\mathcal{F}}_{n+1}\right)\mathsf{S}^{c}_{n+1,k}&\\ =-\frac{1}{\rho^{f}}\left(\boldsymbol{\nabla}\mathsf{p}_{n+1}+\boldsymbol{\nabla}\mathsf{p}_{n}-\boldsymbol{\nabla}\mathsf{p}_{n}\right)\mathsf{V}^{c}_{n+1}&\\ +\nu^{\mathcal{F}}\sum_{k=1}^{\mathrm{N}^{\mathcal{F}}_{\text{faces}}}\boldsymbol{\nabla}_{\boldsymbol{n}_k}\mathbf{u}^{\mathcal{F}}_{n+1,k}\,\mathbf{n}_{n+1,k}\,\mathsf{S}^{c}_{n+1,k}+\frac{\bar{\mathbf{b}}^{\mathcal{F}}_{n}}{\rho^{\mathcal{F}}}\,\mathsf{V}^{c}_{n+1},&\end{aligned}\tag{2.87}$$

which is then split into

$$\begin{aligned}\frac{\widetilde{\mathbf{u}}^{\mathcal{F}}_{n+1}\,\mathsf{V}^{c}_{n+1}-\mathbf{u}^{\mathcal{F}}_{n}\,\mathsf{V}^{c}_{n}}{\Delta t^{\mathcal{F}}_{n+1}}+\sum_{k=1}^{\mathrm{N}^{\mathcal{F}}_{\text{faces}}}\mathbf{F}^{\text{conv}}_{k}\left(\mathbf{u}^{\mathcal{F}}_{n,k},\mathbf{u}^{\mathcal{F}}_{n+1}\right)\mathsf{S}^{c}_{n+1,k}&\\ =-\frac{\boldsymbol{\nabla}\mathsf{p}_{n}}{\rho^{\mathcal{F}}}\,\mathsf{V}^{c}_{n+1}+\nu^{\mathcal{F}}\sum_{k=1}^{\mathrm{N}^{\mathcal{F}}_{\text{faces}}}\boldsymbol{\nabla}_{\boldsymbol{n}_k}\mathbf{u}^{\mathcal{F}}_{n+1,k}\,\mathbf{n}_{n+1,k}\,\mathsf{S}^{c}_{n+1,k}+\frac{\bar{\mathbf{b}}^{\mathcal{F}}_{n}}{\rho^{\mathcal{F}}}\,\mathsf{V}^{c}_{n+1}&\end{aligned}\tag{2.88}$$

and

$$\frac{\mathbf{u}^{\mathcal{F}}_{n+1}-\widetilde{\mathbf{u}}^{\mathcal{F}}_{n+1}}{\Delta t^{\mathcal{F}}_{n+1}}\,\mathsf{V}^{c}_{n+1}=-\frac{1}{\rho^{\mathcal{F}}}\left(\boldsymbol{\nabla}\mathsf{p}_{n+1}-\boldsymbol{\nabla}\mathsf{p}_{n}\right)\mathsf{V}^{c}_{n+1}.\tag{2.89}$$

In general, the intermediate velocity field in the momentum equation (2.88) does not satisfy the continuity equation (2.86a). To rectify this, the velocity field is corrected at the cell center and faces in order to be conservative. Using equations (2.89), it can be done with

$$\mathbf{u}^{\mathcal{F}}_{n+1}=\widetilde{\mathbf{u}}^{\mathcal{F}}_{n+1}-\frac{\Delta t^{\mathcal{F}}_{n+1}}{\rho^{\mathcal{F}}}\left(\boldsymbol{\nabla}\mathsf{p}_{n+1}-\boldsymbol{\nabla}\mathsf{p}_{n}\right),\tag{2.90a}$$

$$\mathbf{u}^{\mathcal{F}}_{n+1,k}=\widetilde{\mathbf{u}}^{\mathcal{F}}_{n+1,k}-\frac{\Delta t^{\mathcal{F}}_{n+1}}{\rho^{\mathcal{F}}}\left(\boldsymbol{\nabla}\mathsf{p}_{n+1,k}-\boldsymbol{\nabla}\mathsf{p}_{n,k}\right).\tag{2.90b}$$

The *Poisson's equation for pressure*, used to evaluate the pressure at the next time station, is then obtained by substituting equations (2.90b) into (2.86a), which reads

$$\begin{aligned}\frac{\mathsf{V}^{c}_{n+1}-\mathsf{V}^{c}_{n}}{\Delta t^{\mathcal{F}}_{n+1}}+\sum_{k=1}^{\mathrm{N}^{\mathcal{F}}_{\text{faces}}}\widetilde{\mathbf{u}}^{\mathcal{F}}_{n+1,k}\cdot\mathbf{n}_{n+1,k}\,\mathsf{S}^{c}_{n+1,k}&\\ =\frac{\Delta t^{\mathcal{F}}_{n+1}}{\rho^{\mathcal{F}}}\sum_{k=1}^{\mathrm{N}^{\mathcal{F}}_{\text{faces}}}\left(\boldsymbol{\nabla}\mathsf{p}_{n+1,k}-\boldsymbol{\nabla}\mathsf{p}_{n,k}\right)\cdot\mathbf{n}_{n+1,k}\,\mathsf{S}^{c}_{n+1}.&\end{aligned}\tag{2.91}$$

Following this approach, the momentum equation is solved with use of the pressure field and the conservative velocity at the cell faces from the previous time station. When solid bodies move in the fluid sub-domain with respect to the fixed grid, the mesh changes during the new time step and the velocities on the cell faces are not valid anymore for solving the momentum equations. In the current method, the pressure equation is computed first, which allows having the conservative velocity field on cell faces at the current time station for solving the momentum equations. As a consequence, the intermediate velocity field in equations (2.90) and (2.91) is replaced by the velocity from the previous time station, which is reconstructed on the cell faces by a linear interpolation of cell center values.

As mentioned before, with co-located grid arrangements, the velocity and pressure fields are evaluated at the cell center while the discrete equations for velocity and pressure are decoupled, which leads to pressure oscillations on the computational grid and is commonly referred to as checker-board problem. These non-physical pressures can be dealt with by using a *higher-order pressure gradient* operator in equations (2.91) as proposed by Rhie and Chow in [153], which eventually allows to damp the spurious oscillations. In the present context, it implies that the pressure gradient at the time station $t_n^{\mathcal{F}}$ in (2.90b) be approximated with a higher order of accuracy. Including the changes mentioned previously as well as a high-order operator $\boldsymbol{\nabla}_H\left(\cdot\right)$ that approximates the pressure gradient, (2.90b) becomes[15]

$$\mathbf{u}_{n+1,k}^{\mathcal{F}} = \mathbf{u}_{n,k^+}^{\mathcal{F}} - \frac{\Delta t_{n+1}^{\mathcal{F}}}{\rho^{\mathcal{F}}}\left(\boldsymbol{\nabla}\mathsf{p}_{n+1,k} - \boldsymbol{\nabla}_H\,\mathsf{p}_{n,k^+}\right). \tag{2.92}$$

However, applying this correction makes even a steady-state solution dependent on the time step. This dependency can be eliminated by rather using the following expression for the fluid velocity at the cell faces:

$$\mathbf{u}_{n+1,k}^{\mathcal{F}} = \mathbf{u}_{n,k^+}^{\mathcal{F}} - \frac{\Delta t_{n+1}^{\mathcal{F}}}{\rho^{\mathcal{F}}}\left(\boldsymbol{\nabla}\mathsf{p}_{n+1,k} - (1-\beta)\,\boldsymbol{\nabla}\mathsf{p}_{n,k^+} - \beta\,\boldsymbol{\nabla}_H\,\mathsf{p}_{n,k^+}\right), \tag{2.93}$$

where the parameter $\beta = \min\left\{1, \Delta t_{n+1}^{\mathcal{F},\mathrm{cr}}/\Delta t_{n+1}^{\mathcal{F}}\right\}$ and

$$\Delta t_{n+1}^{\mathcal{F},\mathrm{cr}} = \min_c \left\{\frac{L_{n+1}^c}{|\mathbf{u}_{n+1}^{\mathcal{F},c}|}\right\} \quad \forall c \in \Omega_h^{\mathcal{F}} \tag{2.94}$$

is a *critical time step size* with L_{n+1}^c the characteristic dimension of cell c and $|\mathsf{u}_{\mathsf{x},i}^{\mathcal{F},c}|$ the velocity magnitude in that same cell. This parameter also introduces a stability condition to the simulation, which is further discussed in section 2.2.5.1. As for the pressure equation (2.91) it

[15] The notation $(\cdot)_{n,k^+}$ is used to designate variables reconstructed on a cell face k at time $t_{n+1}^{\mathcal{F}}$ with a linear interpolation from neighbor cells' center values at time $t_n^{\mathcal{F}}$.

becomes

$$\frac{V^c_{n+1} - V^c_n}{\Delta t^{\mathcal{F}}_{n+1}} + \sum_{k=1}^{\mathrm{N}^{\mathcal{F}}_{\text{faces}}} \mathbf{u}^{\mathcal{F}}_{n,k^+} \cdot \mathbf{n}_{n+1,k}\, \mathsf{S}^c_{n+1,k}$$
$$= \frac{\Delta t^{\mathcal{F}}_{n+1}}{\rho^{\mathcal{F}}} \sum_{k=1}^{\mathrm{N}^{\mathcal{F}}_{\text{faces}}} \left(\boldsymbol{\nabla} \mathsf{p}_{n+1,k} - (1-\beta)\, \boldsymbol{\nabla} \mathsf{p}_{n,k^+} - \beta\, \boldsymbol{\nabla}_H\, \mathsf{p}_{n,k^+} \right) \cdot \mathbf{n}_{n+1,k}\, \mathsf{S}^c_{n+1,k}. \quad (2.95)$$

Solution algorithm

The solution procedure to solve the Navier-Stokes equations (2.69) at time $t^{\mathcal{F}}_{n+1}$ is based on the previous developments and is finally given by the following algorithm:

1. Mesh update: following a body motion, the new position of the interface is captured and volumes V^c_{n+1} are calculated in each fluid cell using the procedure presented in section 2.2.3.

2. In the case of a free-surface flow, the phase transfer equations are solved using a VOF approach and the position of the free-surface is updated. Free-surface flows and their resolution are briefly discussed in section 2.2.6.1.

3. The pressure field p_{n+1} is found by solving the Poisson's equation for pressure (2.95).

4. The conservative (satisfying the continuity equation) velocity at cell faces and the mass flow rates $\rho^{\mathcal{F}}\, \mathbf{u}^{\mathcal{F}}_{n+1,k}$ through these faces are computed with (2.93).

5. Using the conservative velocity face values computed at step 4., the conservative velocity field $\mathbf{u}^{\mathcal{F}}_{n+1}$ at cell centers is finally obtained by solving the momentum equations:

$$\frac{\mathbf{u}^{\mathcal{F}}_{n+1}\, \mathsf{V}^c_{n+1} - \mathbf{u}^{\mathcal{F}}_n\, \mathsf{V}^c_n}{\Delta t^{\mathcal{F}}_{n+1}} + \sum_{k=1}^{\mathrm{N}^{\mathcal{F}}_{\text{faces}}} \mathbf{F}^{\text{conv}}_k \left(\mathbf{u}^{\mathcal{F}}_{n+1,k}, \mathbf{u}^{\mathcal{F}}_{n+1}\right) \mathsf{S}^c_{n+1,k}$$
$$= -\frac{1}{\rho^{\mathcal{F}}} \left(2\, \boldsymbol{\nabla} \mathsf{p}_{n+1} + \boldsymbol{\nabla} \mathsf{p}_n\right) \mathsf{V}^c_{n+1} + \nu^{\mathcal{F}} \sum_{k=1}^{\mathrm{N}^{\mathcal{F}}_{\text{faces}}} \boldsymbol{\nabla}_{\boldsymbol{n}_k} \mathbf{u}^{\mathcal{F}}_{n+1,k}\, \mathbf{n}_{n+1,k}\, \mathsf{S}^c_{n+1,k} + \frac{\bar{\boldsymbol{b}}^{\mathcal{F}}_n}{\rho^{\mathcal{F}}} \mathsf{V}^c_{n+1}. \quad (2.96)$$

6. Fluid values requested as output anywhere within the fluid sub-domain, including on the fluid moving boundaries or on the fluid-structure interface, are approximated using a linear interpolation of cell center values as shown in Figure 2.10b.

2.2.5.1 Stability of the time integration scheme and CFL number

Reconsidering the CFL condition introduced in section 2.1.4 for the solid field that sets a stability limit on the time step size based on a critical time step, an analogous condition can be introduced for the computation of the fluid flow. The inverse of the parameter β in equations (2.93) can be related to a so-called *CFL number* C that acts as a stability limit for the simulation. The time step size used in practice can then be expressed as

$$\Delta t_n^{\mathcal{F}} = C \; S_{\Delta t}^{\mathcal{F}} \; \Delta t_n^{\mathcal{F},\mathrm{cr}} \tag{2.97}$$

where $S_{\Delta t}^{\mathcal{F}} = 0.5$ is a *safety factor* that guarantees a conservative estimation of the critical time step size $\Delta t_n^{\mathcal{F},\mathrm{cr}}$ defined in (2.94). This critical time step relates the grid size to a characteristic velocity, which characterizes the time a perturbation in the flow passes through the mesh. It plays an important role especially in unsteady flows such as the ones considered in the thesis to set a stable time step size. As can be seen in (2.94), the smaller the grid size and the larger the fluid velocity, the more restrictive the critical time step will be. The CFL number allows to control the impact of the stability limit given by the critical time step and is chosen in practice by making a compromise between stability and simulation time such that a CFL number $C > 1$ can in general be considered for implicit schemes.

When the fluid domain includes moving boundaries and the overall simulation is driven by the movement of the moving boundary, as it is the case in FSI problems, another condition based on a *surface CFL number* C^{S} and a *surface critical time step* $\Delta t_n^{\mathrm{MB,cr}}$ can be used. The latter can be expressed as

$$\Delta t_n^{\mathrm{MB,cr}} = \min_c \left\{ \frac{L_n^c}{|\mathbf{u}_n^{\mathrm{MB},c}|} \right\} \quad \forall c \in \Gamma^{\mathrm{MB}} \tag{2.98}$$

where the velocity magnitude of the moving boundary $|\mathbf{u}_n^{\mathrm{MB},c}|$ is considered in each cell. The corresponding time step is given in this case by

$$\Delta t_n^{\mathrm{MB}} = C^{\mathrm{S}} \; S_{\Delta t}^{\mathcal{F}} \; \Delta t_n^{\mathrm{MB,cr}} \tag{2.99}$$

with a less restrictive safety factor $S_{\Delta t}^{\mathcal{F}} = 1$ and allows accounting for the effect of the boundary motion on the stability limit.

This completes the presentation of the fluid mechanics system employed in this thesis for solving the FSI applications of interest. A detailed description of the space and time discretization of the fluid domain has shown how the primary fields of unknowns $\mathbf{u}^{\mathcal{F}}$ and p are computed on the fluid computational grid, which are in turn used to evaluate the fluid loads applied onto the fluid side of the fluid-structure interface. This will be presented in the next chapter while the solid and fluid sub-domains are merged to formulate the FSI problem. Before that, the free-surface included in the fluid domain and the corresponding multi-phase flow present in tire hydroplaning simulations are briefly introduced in the next section.

2.2.6 Hydroplaning-specific modeling aspects

2.2.6.1 Free-surface flow and phase transfer with VOF method

In many engineering applications [167, 180, 192] such as the one of interest in this thesis, the fluid domain can be split into two or more phases defined by multiple fluids and a so-called *free-surface* at their interface. In the case of tire hydroplaning, in addition to liquid water, the fluid domain is also filled with air in its gas phase. However, the computation of the air flow around a tire is of no interest while evaluating its hydroplaning performance. Since solving both phases is computationally very expensive, the two-phase fluid model is simplified in that the rest of the fluid domain not occupied by water is simply considered as void. In doing so, the additional physical flow does not need to be computed and the free-surface can still be defined at the interface between water and the void phase.

The position of this free-surface and its evolution over time must be described appropriately. Various methods consider this problem, for which a common solution is to use a so-called *surface tracking* approach. In that case, the mesh moves to “track” the interface and fit the free-surface as in *front tracking approaches* described e.g. in [160, 173] or in [35] where an ALE description is used to account for the motion of internal and external (i.e. on the free-surface) nodes. The main disadvantage of these approaches is that the grid generation at each time step is computationally expensive and that the simulation of very large deformations of the interface is in general not feasible. An alternative solution consists in using *surface capturing* approaches, where the position of the interface needs to be “captured” within a fixed-grid. These methods include *level set* techniques [1, 2, 139] or the *volume of fluid* (VOF) approach [86, 112, 187], which do not require any mesh regeneration and can therefore be used with any deformation of the free-surface. A typical drawback of these approaches is that the interface is captured with the resolution of the underlying fixed-grid. Following a large displacement of the free-surface, the latter could find itself in a region of the fluid domain where the mesh is too coarse to describe appropriately the fluid flow. However, an AMR technique such as the one introduced in section 2.2.4.1 allows to circumvent this issue by refining automatically the cells close the free-surface and following its motion.

The method implemented in the fluid solver employed in this work is based on an extension of the general VOF approach [86] for which the governing equations are solved in all fluid cells, i.e. not only in the cells completely occupied by one fluid phase but also the ones intersected by the free-surface. The reconstruction of the free-surface inside the fluid cells in question is achieved through the use of a *VOF variable* denoted here by Q, which allows to characterize the phase transfer within the fluid domain. In each cell of the computation grid, Q takes values from 0 in pure gas (or void) to 1 in pure fluid, so if $0 < Q < 1$ the cell contains a free-surface and needs further treatment. The VOF variable Q is transferred through the grid by means of a *transport equation* given by

$$\frac{\partial Q}{\partial t} + \boldsymbol{u}^{\mathcal{F}} \cdot \boldsymbol{\nabla} Q = 0 \tag{2.100}$$

and fluid properties such as $\mu^{\mathcal{F}}$ or $\rho^{\mathcal{F}}$ denoted here by q are computed in each cell of interest as

$$q = Q\, q_{\text{liq}} + (1 - Q)\, q_{\text{gas}}, \tag{2.101}$$

where q_{liq} and q_{gas} are fluid properties associated to a liquid and a gas phase respectively.

An explicit procedure is used to solve the transport equation (2.100) and compute the transfer of Q between cells intersected by a free-surface and their neighbors. Based on the value of Q, the free-surface is then reconstructed within the cells by a set of connected facets as for the moving boundary presented in the previous section. While precisely capturing the position of the free-surface is important to accurately describe the fluid-structure interface in an FSI application such as in the tire hydroplaning problem, a detailed description of the VOF method is out of scope of this work. More details can however be found e.g. in [5].

The explicit treatment of the phase transfer introduces an additional critical time step in the time integration procedure, which can be prohibitive in cells occupied by droplets due to their potential small size, as in the case of tire hydroplaning. In order to avoid a severe restriction on the time step size for the entire fluid computation, a surface CFL condition of the type of (2.99) is used, for which the critical time step is calculated based on the velocity magnitude of the free-surface instead of the moving boundary in each cell. In doing so, the stability limit is evaluated for cells intersected by the main free-surface and discards the effect of small droplets on the time step size, for which the solution is of no concern in the current application anyway.

3 Multi-field problem and coupling procedures

The numerical modeling of the solid and fluid sub-domains presented in the previous chapter can be merged together to build a so-called multi-field problem, which in this case is the coupled FSI problem. To do so, an important prerequisite is the compatibility of the chosen solvers both in terms of implementation and mathematical formulation of each single-field. The solid model presented in section 2.1 based on a Lagrangian formulation and the fluid model described in section 2.2 relying on a Eulerian fixed-grid approach with appropriate moving boundary reconstruction and mesh refinement techniques is suitable for computing FSI problems involving large interface motion and complex physics. To be able to couple the two physical models, a coupling procedure enforcing *coupling conditions* at the interface shared by both sub-domains must be constructed. Among the approaches available and reviewed in section 1.3.2, a *partitioned coupling strategy* has been adopted in this work. In such a case, the coupled problem is split into two independent single-field systems, more easily solved separately, which then need to be coupled and synchronized with use of an appropriate *coupling scheme* that enforces the above mentioned coupling conditions.

As discussed in section 1.4.1, partitioned coupling strategies are best suited for solving problems where both physics are very complex, which require specifically designed single-field solvers. In particular, loosely-coupled partitioned procedures, chosen for their efficiency and compatibility with the black-box solvers at hand, are applied to the tire hydroplaning problem. The design of such coupling procedures and related requirements are presented in this chapter along with an exhaustive review of past and current research in the field. The pursued intent is to provide sufficient theoretical background on partitioned coupling procedures, and in particular loosely-coupled schemes, before addressing in chapter 4 the coupling strategy proposed in this thesis for tire applications.

3.1 Problem formulation and coupling conditions

Considering a generic mechanical system composed of a deformable solid domain $\Omega^{\mathcal{S}}$ interacting with a fluid under motion $\Omega^{\mathcal{F}}$, both evolving in time and finding themselves in their current

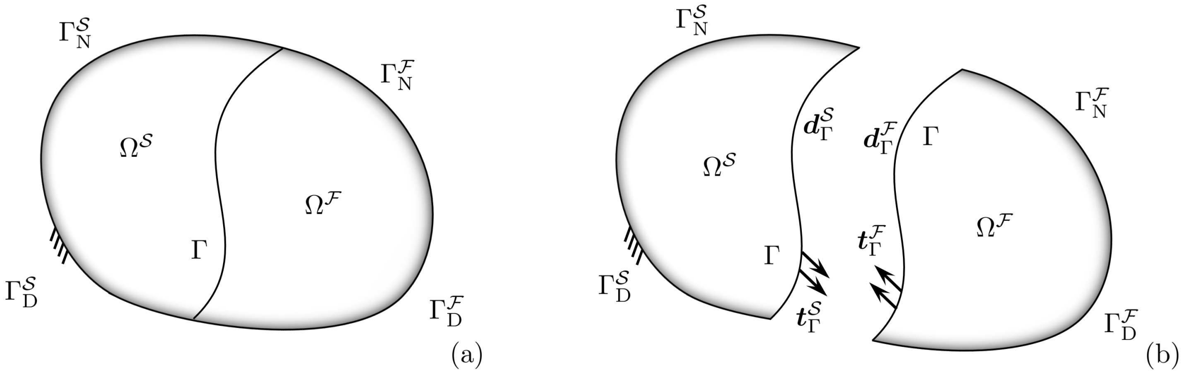

Figure 3.1: (a) The solid $\Omega^{\mathcal{S}}$ and fluid $\Omega^{\mathcal{F}}$ sub-domains are bounded by Dirichlet boundaries $\Gamma_D^{\mathcal{S}}$ and $\Gamma_D^{\mathcal{F}}$, Neumann boundaries $\Gamma_N^{\mathcal{S}}$ and $\Gamma_N^{\mathcal{F}}$ and a common fluid-structure interface Γ. (b) Kinematic and dynamic continuity conditions are applied at the fluid-structure interface Γ according to (3.3) and (3.4) respectively.

configuration, the *fluid-structure interface* can be defined as the intersection of the solid and fluid boundaries as

$$\Gamma = \partial\Omega^{\mathcal{S}} \cap \partial\Omega^{\mathcal{F}} \tag{3.1}$$

and corresponds to a shared surface (in 3D) across which both sub-domains interact. In an FSI problem statement, the moving boundary Γ^{MB} immersed within the fluid sub-domain described in section 2.2.3 for a rigid body, can be replaced by the fluid-structure interface Γ. The discretization and reconstruction technique of the moving boundary presented at that occasion can be directly transposed to the fluid-structure interface without any further assumption or simplification. The coupled problem is eventually composed of both sub-domains $\Omega^{\mathcal{S}}$ and $\Omega^{\mathcal{F}}$, their Dirichlet and Neumann boundaries $\Gamma_D^{\mathcal{S}}$, $\Gamma_D^{\mathcal{F}}$ and $\Gamma_N^{\mathcal{S}}$, $\Gamma_N^{\mathcal{F}}$ as well as their shared fluid-structure interface Γ as illustrated in Figure 3.1a.

Both single-fields are entirely described by their governing equations together with initial and boundary conditions presented in chapter 2. To complete the mathematical formulation of the coupled system, an additional set of physically meaningful conditions must be enforced on the fluid-structure interface Γ. These coupling conditions, also commonly referred to as *transmission conditions* or *interface continuity conditions*, guarantee the compatibility and continuity of kinematic and dynamic quantities between the two sub-domains.

The *kinematic continuity condition* imposes a no-slip condition at the interface, which prevents any fluid flow across Γ and any relative motion between the solid and fluid at Γ, reads

$$\dot{\boldsymbol{d}}_\Gamma^{\mathcal{S}} = \boldsymbol{u}_\Gamma^{\mathcal{F}} \qquad \text{on } \Gamma \times [0,T]. \tag{3.2}$$

Integrating expression (3.2) in time then yields

$$\boldsymbol{d}_\Gamma^{\mathcal{S}} = \boldsymbol{d}_\Gamma^{\mathcal{F}} \qquad \text{on } \Gamma \times [0, T], \tag{3.3}$$

where $\boldsymbol{d}_\Gamma^{\mathcal{F}}$ denotes the displacement of fluid particles at the interface and corresponds, in this particular setting, to the displacement of the fluid moving boundary within the fluid domain. The reconstruction of that boundary is performed in the present fluid solver based on the exact displacement field of the fluid-structure interface. This intrinsic feature of the flow solver guarantees that the kinematic condition (3.3) is automatically fulfilled, at least right after the reconstruction process, as further discussed in section 3.3.5.1.

The *dynamic continuity condition* stems from *Newton's third law* [131], the *action-reaction* principle, which requires the traction field at the interface to be balanced at all times during the simulation. Applying the *method of sections* by separating the two sub-domains at the shared interface as illustrated in Figure 3.1b, the dynamic condition then reads

$$\boldsymbol{t}_\Gamma^{\mathcal{S}} = -\boldsymbol{t}_\Gamma^{\mathcal{F}} \qquad \text{on } \Gamma \times [0, T], \tag{3.4}$$

where $\boldsymbol{t}_\Gamma^{\mathcal{S}}$ is the traction vector (2.13) on the solid side of the interface and $\boldsymbol{t}_\Gamma^{\mathcal{F}}$ the *fluid traction vector* at the interface defined by

$$\boldsymbol{t}_\Gamma^{\mathcal{F}} = \boldsymbol{\sigma}^{\mathcal{F}} \cdot \boldsymbol{n}_\Gamma^{\mathcal{F}}, \tag{3.5}$$

in which $\boldsymbol{\sigma}^{\mathcal{F}}$ is the fluid Cauchy stress tensor given in (2.58) and $\boldsymbol{n}_\Gamma^{\mathcal{F}}$ the outward unit normal to the fluid side of the interface. The surface traction vector $\boldsymbol{t}^{\mathcal{F}}$ contains the loads resulting from the fluid pressure and viscous stresses at the fluid-structure interface and its evaluation by the fluid solver used in this work is further detailed in section 3.1.1. The dynamic condition can also be written in terms of *interface force vectors* by multiplying both sides of (3.4) by the surface area of the corresponding side of the interface as

$$\boldsymbol{f}_\Gamma^{\mathcal{S}} = -\boldsymbol{f}_\Gamma^{\mathcal{F}} \qquad \text{on } \Gamma \times [0, T], \tag{3.6}$$

where $\boldsymbol{f}_\Gamma^{\mathcal{S}}$ and $\boldsymbol{f}_\Gamma^{\mathcal{F}}$ denote the force vectors applied on the solid and fluid side of the interface respectively. Note that, in the discrete setting, the transition from expression (3.5) to (3.6) only holds in the case of a matching mesh at the fluid-structure interface, in particular when the surface area is equivalent on both fluid and solid sides of the shared interface.

The expressions (3.3) and (3.4) already conclude the mathematical formulation of the coupled problem, which enforce the continuity of kinematic and dynamic quantities across the fluid-structure interface and, as shown in section 3.2.2, are sufficient to guarantee the conservation of mass, momentum and energy at the interface. While these conditions can be respected per construction in monolithic approaches, partitioned strategies manage at best to fulfill them up to a certain tolerance in implicit coupling strategies. As for explicit partitioned schemes, they are unable to enforce both continuity conditions but still allow to construct robust loosely-coupled schemes as further discussed in the remaining of this chapter.

3.1.1 Fluid loads on the fluid-structure interface

Once the fluid state is fully computed using the procedure presented in section 2.2.5, the reaction force from the fluid onto the moving boundary, or in this case the fluid-structure interface, can be computed. In order to fulfill the dynamic continuity condition (3.4), the traction field at the fluid-structure interface must be balanced and thus a traction vector $\boldsymbol{t}_\Gamma^\mathcal{S} = -\boldsymbol{t}_\Gamma^\mathcal{F}$ applied onto the solid external surface at the interface. In practice, the surface traction vector is multiplied by the surface area of the corresponding interface to obtain the sought reaction force, which is then transferred to the solid solver.

As discussed in section 2.2.3, the geometry of a moving body is approximated by a collection of plane facets, which are used to reconstruct the moving boundary within the fluid sub-domain. When this immersed body corresponds to the solid field of an FSI problem, its external surface is already represented as a set of plane facets that coincide with the outputs of the solid FE solver. The situation at the end of a fluid time step can be illustrated for a 2D example as in Figure 3.2, where the surface traction $\boldsymbol{t}_{\Gamma,fk}^\mathcal{F}$ reconstructed on each fluid face k are multiplied by the associated surface area to obtain the resulting fluid force on the solid facet f as

$$\boldsymbol{f}_{\Gamma,f}^\mathcal{F} = \sum_{k=1}^{\mathrm{N}_{\Gamma,\text{faces}}^{\mathcal{F},f}} \boldsymbol{t}_{\Gamma,fk}^\mathcal{F}\, S_{fk}^\mathcal{F} \tag{3.7}$$

where $\mathrm{N}_{\Gamma,\text{faces}}^{\mathcal{F},f}$ denotes the number of fluid faces created at the solid facet f of the interface, whereas $S_{fk}^\mathcal{F}$ is the surface area of each fluid face k. The fluid reaction force for all solid facets belonging to the fluid-structure interface are computed in the same way before being sent to the solid solver. The synchronization of the data exchange between two solvers and the application of the fluid load onto the fluid-structure interface in partitioned coupling schemes are further discussed in section 3.3.

3.2 Partitioned coupling procedures

Thanks to their flexibility and modularity, partitioned procedures are very often used to solve multi-field problems, including FSI problems. First introduced in the late 70's by Felippa and Park [55, 56], partitioned strategies allow to use specifically designed solvers to compute each single-field separately, which then need to be coupled together in order to merge the resulting partitioned problem at its interface. A coupling scheme is employed to that effect by enforcing the continuity conditions (3.3) and (3.4) for the whole duration of the simulation. General

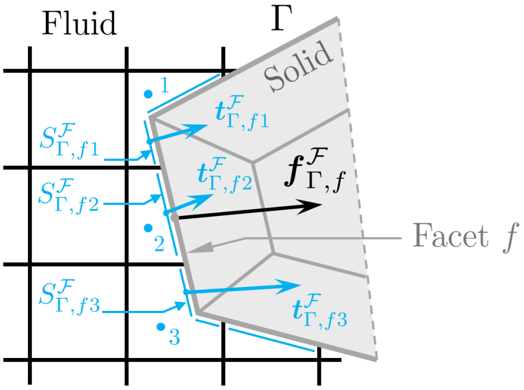
Fluid
Γ
Solid
1
2
3
$S^{\mathcal{F}}_{\Gamma,f1}$
$S^{\mathcal{F}}_{\Gamma,f2}$
$S^{\mathcal{F}}_{\Gamma,f3}$
$\boldsymbol{t}^{\mathcal{F}}_{\Gamma,f1}$
$\boldsymbol{t}^{\mathcal{F}}_{\Gamma,f2}$
$\boldsymbol{t}^{\mathcal{F}}_{\Gamma,f3}$
$\boldsymbol{f}^{\mathcal{F}}_{\Gamma,f}$
Facet f

Figure 3.2: Evaluation of the fluid force $\boldsymbol{f}^{\mathcal{F}}_{\Gamma,f}$ on the solid facet f of the fluid-structure interface Γ based on the surface traction $\boldsymbol{t}^{\mathcal{F}}_{\Gamma,fk}$ and the corresponding surface areas $S^{\mathcal{F}}_{fk}$ computed by the fluid solver for each fluid face k (here $k = 1, \ldots, 3$) according to (3.7).

properties and key features of partitioned procedures are described with more detail in the following.

3.2.1 Generalities and properties

Two categories of such partitioned coupling procedures already introduced in section 1.3.2 exist: a strongly-coupled or implicit coupling strategy, and a loosely-coupled or explicit coupling strategy. Implicit approaches aim at enforcing the continuity at the interface by *sub-iterating* between the two sub-domains until sufficient convergence of one of the coupling conditions is achieved, approaching in that way the ultimate strongly coupled solution offered by monolithic schemes. Various methods reviewed in section 1.3.2 are given in the literature to solve the partitioned system but in general at the cost of complex implementation and expensive computations for guaranteeing numerical stability and reducing the error made on the continuity conditions. Furthermore, strongly-coupled approaches require in general a minimum of access to the single-field solvers in order to control their synchronization and reach data used during the iterative process. For some black-box solvers, such as the ones employed in this work, these requirements are not met and such a strategy cannot be implemented. In an explicit coupling strategy, no sub-iterations are performed so that the solution obtained at the end of an FSI time step is directly transferred to the other solver without keeping account of the continuity conditions. As a matter of fact, explicit coupling schemes always violate the continuity conditions and introduce a *partitioning error* in the coupled system as discussed in detail in section 3.3.5.1. Nevertheless, robust and efficient explicit coupling schemes have been developed over the past recent years and successfully applied to FSI problems as shown in section 3.3, while reviewing loosely-coupled strategies proposed in the literature. Strongly-coupled and loosely-coupled procedures

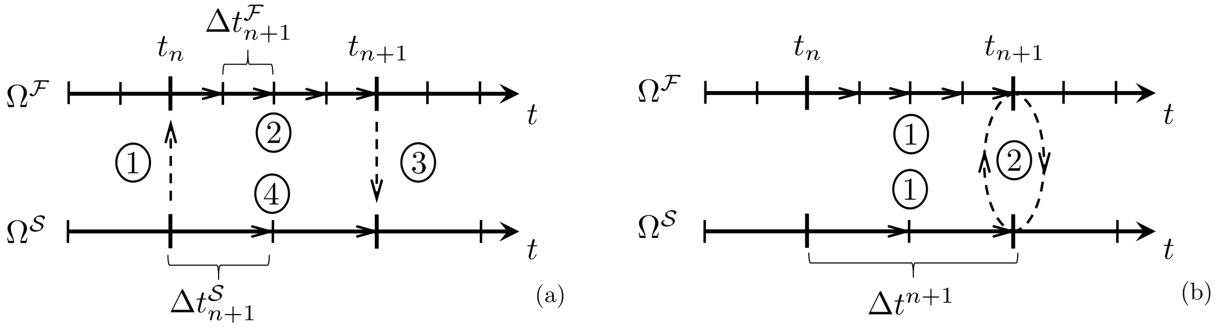

Figure 3.3: Schematic representation of most basic explicit (a) and implicit (b) coupling schemes. Each single-field is sub-cycled such that $R^{\mathcal{S}}_{n+1} = 2$ and $R^{\mathcal{F}}_{n+1} = 4$.

are schematically illustrated in their simplest version in Figure 3.3, where each single-field is computed separately during an *FSI time step*, which can be defined as

$$\Delta t_{n+1} = t_{n+1} - t_n. \tag{3.8}$$

The solution to each single-field must be computed at least at each time instant t_n and t_{n+1}, which correspond respectively to the beginning and the end of the $(n+1)^{\text{th}}$ FSI time step, since these correspond to synchronization points in time when both sub-domains are reconciled. The FSI time step is thus also commonly referred to as *coupling* or *synchronizing time step*. In the explicit variant, coupling quantities are exchanged only once per time step (step 1 and 3 in Figure 3.3a) whereas, in the implicit approach, both single-fields are advanced (step 1 in Figure 3.3b) before being sub-iterated until sufficient convergence is achieved (step 2 in Figure 3.3b).

An extra advantage of partitioned algorithms resides in the possibility for each single-field to use its own time step $\Delta t^{\mathcal{F}}_{n+1}$ or $\Delta t^{\mathcal{S}}_{n+1}$, smaller or equal to the FSI time step Δt_{n+1}, in order to fulfill the CFL conditions (2.47), and (2.97) or (2.99) for the solid and fluid sub-domains respectively. In that situation, also illustrated in Figure 3.3, the solid and/or fluid sub-domains are said to be *sub-cycled* based on a so-called *sub-cycling ratio*

$$R^{\mathcal{S}}_{n+1} = \frac{\Delta t_{n+1}}{\Delta t^{\mathcal{S}}_{n+1}} \in \mathbb{N} \quad \text{and} \quad R^{\mathcal{F}}_{n+1} = \frac{\Delta t_{n+1}}{\Delta t^{\mathcal{F}}_{n+1}} \in \mathbb{N}, \tag{3.9}$$

for the solid and fluid fields respectively. These ratios can either be constant for the whole duration of the simulation or an additional variable in each FSI time step.

In the rest of this section, coupled problems are further characterized with help of their balance equations. It will allow to emphasize specific requirements and criteria to be met while constructing staggered solution procedures as well as corresponding errors arising from the partition of the coupled problem. A thorough review of partitioned strategies, including an historic of successive developments, can also be found e.g. in the thesis by Mok [125] or in the review article by Felippa et al. [57].

3.2.2 Balance equations at the fluid-structure interface

Similarly to the solid and fluid single-fields, a coupled problem can be described by its balance equations of mass, momentum and energy. The connection between these conservation equations and the coupling conditions introduced in section 3.1 are emphasized and will allow to introduce a criterion for robustness and accuracy of the coupled numerical procedure.

Following a similar development as the one proposed in the thesis by Mok [125], the conservation equations are written at the fluid-structure interface during one FSI time step Δt_{n+1} in order to highlight possible creation or destruction of artificial quantities resulting from the time discretization of the coupled problem.

3.2.2.1 Mass conservation

Assuming that no material can appear or disappear and that both sub-domains can neither overlap nor separate from one another during one FSI time step, the conservation of mass of the coupled system is guaranteed if the continuity of displacement at the interface given by (3.3) is respected and that both single-field solvers are mass conserving. Since both solvers employed in this work fulfill the latter condition, mass conservation is achieved by simply enforcing the kinematic continuity condition in each time step.

3.2.2.2 Momentum conservation

A coupled problem formed by the two sub-systems $\Omega^{\mathcal{S}}$ and $\Omega^{\mathcal{F}}$ with the interface traction vectors $\boldsymbol{t}_{\Gamma}^{\mathcal{S}}$ and $\boldsymbol{t}_{\Gamma}^{\mathcal{F}}$ previously defined is considered. The conservation of momentum requires the change of momentum in the coupled system due to the loads at the interface to correspond exactly to the sum of contributions from each partitioned sub-system during one time step Δt_{n+1}, thus avoiding any artificial momentum that might result from the partition in time of the coupled system. Imposing the momentum conservation then yields

$$\int_{t_n}^{t_{n+1}} \boldsymbol{t}_{\Gamma}^{\mathcal{S}} + \boldsymbol{t}_{\Gamma}^{\mathcal{F}} \,\mathrm{d}t = \mathbf{0}, \tag{3.10}$$

which accounts for the dynamic equilibrium in the coupled domain. The expression (3.10) hence simply requires the dynamic continuity condition (3.4) to be imposed in each coupling time step of the FSI simulation. Note that, since the geometry of the interface is exactly matching on the solid and fluid sides, i.e. the reconstruction of the interface within the fluid sub-domain is achieved with the exact geometry resulting from the solid solver, interface traction or interface force vectors can be equivalently used in the equilibrium expression (3.10). As both solvers use interface forces as primary outputs, the latter are rather used to describe balance formulations in the following.

3.2.2.3 Energy conservation

An additional expression describing the balance of energy at the fluid-structure interface, which proves to be very useful for characterizing coupled systems, can be given by following the same logic. In this case, the coupling procedure should not create or dissipate any energy at the interface compared to the solution resulting from the two partitioned sub-systems. The amount of energy possibly produced at the interface during one FSI time step Δt_{n+1} can be evaluated as

$$\Delta E_{\Gamma,n+1} = \Delta E^{\mathcal{S}}_{\Gamma,n+1} + \Delta E^{\mathcal{F}}_{\Gamma,n+1}, \tag{3.11}$$

where $\Delta E^{\mathcal{S}}_{\Gamma,n+1}$ and $\Delta E^{\mathcal{F}}_{\Gamma,n+1}$ denote respectively the energy contribution from the solid and fluid fields at the interface, which should be balanced such that $\Delta E_{\Gamma,n+1} = 0$ in each time step. These energy contributions can be related to the work performed by the interface solid and fluid loads along the motion of the interface [126, 147] respectively given by

$$\Delta E^{\mathcal{S}}_{\Gamma,n+1} = \int_{t_n}^{t_{n+1}} \boldsymbol{f}^{\mathcal{S}}_{\Gamma} \cdot \dot{\boldsymbol{d}}^{\mathcal{S}}_{\Gamma} \, \mathrm{d}t, \tag{3.12}$$

$$\Delta E^{\mathcal{F}}_{\Gamma,n+1} = \int_{t_n}^{t_{n+1}} \boldsymbol{f}^{\mathcal{F}}_{\Gamma} \cdot \boldsymbol{u}^{\mathcal{F}}_{\Gamma} \, \mathrm{d}t. \tag{3.13}$$

In other words, it means that the work performed by the fluid loads, i.e. fluid pressure and viscous stresses, at the interface must be balanced by the work of the interface solid force field along the motion of the interface and for each FSI time step. Introducing (3.12) and (3.13) into (3.11), the energy balance then reads

$$\Delta E_{\Gamma,n+1} = \int_{t_n}^{t_{n+1}} \boldsymbol{f}^{\mathcal{S}}_{\Gamma} \, \dot{\boldsymbol{d}}^{\mathcal{S}}_{\Gamma} + \boldsymbol{f}^{\mathcal{F}}_{\Gamma} \, \boldsymbol{u}^{\mathcal{F}}_{\Gamma} \, \mathrm{d}t = 0. \tag{3.14}$$

If the dynamic condition (3.6) is fulfilled, the balance expression (3.14) can be rewritten as

$$\Delta E_{\Gamma,n+1} = \int_{t_n}^{t_{n+1}} \boldsymbol{f}^{\mathcal{S}}_{\Gamma} \left(\dot{\boldsymbol{d}}^{\mathcal{S}}_{\Gamma} - \boldsymbol{u}^{\mathcal{F}}_{\Gamma} \right) \, \mathrm{d}t = 0, \tag{3.15}$$

which can only hold if the kinematic condition (3.2) is simultaneously enforced. Therefore, no artificial energy is produced during an FSI time step if both kinematic and dynamic continuity conditions are respected.

Since both continuity conditions appear in (3.14), the interface energy conservation directly guarantees mass and momentum conservation at the interface and consists therefore in a very appropriate tool to evaluate partitioned coupling procedures. As a matter of fact, the interface energy can be used as a rigorous criterion to assess the accuracy of the coupled solution, by introducing the time discretization operator employed for each single-field solver and for the coupled problem into (3.14) [46], as further discussed in the following.

Many research groups have successfully used the interface energy to evaluate partitioned solution procedures. These include, among others, the work by Mok [126], by Piperno [143] also with Farhat [145, 146], by Le Tallec with Mouro [104] and Hauret [103], by Bloom [19] or by Van Brummelen [174], where the last two analyze the interface energy conservation for monolithic approaches as compared to partitioned methods. The suitability of a given partitioned procedure can thus be assessed by such a criterion that measures the artificial energy that is numerically created at the shared interface by the staggering process. Some of these coupling strategies, which are relevant in the context of this thesis, are described in more detail in the next section, while reviewing loosely-coupled partitioned procedures proposed in the literature.

3.3 Loosely-coupled partitioned strategies

As already stated in section 1.4.1, the coupling strategy adopted in this work is based on a loosely-coupled partitioned procedure. While it is the only option that can be considered with the black-box solvers at hand, this coupling strategy also offers the highest degree of flexibility and modularity in terms of software coupling for solving FSI problems. Together with the fact that explicit coupling procedures are the most efficient, avoiding any kind of sub-iterations, they represent the preferred choice for developing coupling algorithms for large scale problems in an industrial environment. However, it comes at the price of numerical stability and even accuracy of the entire coupling scheme if special care is not taken.

3.3.1 CSS algorithm and terminologies

The simplest yet very widespread coupling scheme is the famous *conventional serial staggered* (CSS) strategy illustrated in Figure 3.4 in its original form as proposed by Piperno and Farhat in [47, 148]. The CSS procedure is also commonly known as *Z-scheme* due to the Z-shaped pattern formed by the arrows describing the data transfer. The steps undertaken for advancing the coupled system from t_n to t_{n+1} in this classic explicit coupling strategy are given in algorithm 3.1. It is noteworthy that there is no restriction whatsoever on the time integration scheme to be used in both single-fields. The latter are only required to be reconciled at the end of each time step, when the data transfer takes place. This confers the high flexibility attached to partitioned

Algorithm 3.1 CSS

```
Require: Initial time $t = T_0$, final time $t = T$, time step index $n = 0$, corresponding time step size $\Delta t_1$ and initial solid displacement field $\boldsymbol{d}^{\mathcal{S}}_0$.
while $t < T$ do
    The last known solid displacement field of the interface $\boldsymbol{d}^{\mathcal{S}}_{\Gamma,n}$ is transferred to the fluid solver.
    The fluid moving boundary is reconstructed within $\Omega^{\mathcal{F}}$ and the fluid field advanced to $t_{n+1}$: $(\boldsymbol{u}^{\mathcal{F}}, p)_n \rightarrow (\boldsymbol{u}^{\mathcal{F}}, p)_{n+1}$.
    The fluid loads $\boldsymbol{f}^{\mathcal{F}}_{\Gamma,n+1}$ are computed at the interface as in (3.7) then transferred to the solid solver.
    The load is applied on the solid side of the interface and the solid field advanced to $t_{n+1}$: $\boldsymbol{d}^{\mathcal{S}}_n \rightarrow \boldsymbol{d}^{\mathcal{S}}_{n+1}$.
    Next time step: $n = n + 1$ and $t_{n+1} = t_n + \Delta t_{n+1}$.
end while
```

procedures, which allows in turn both single-field solvers to be separately sub-cycled as needed by their respective stability limits.

A time shift between the evaluation of the primary unknowns in the two single-fields is directly apparent in the CSS procedure. The solid displacements on the interface used to advance the fluid field from t_n to t_{n+1} remains $\boldsymbol{d}^{\mathcal{S}}_{\Gamma,n}$ during the whole time step Δt_{n+1}, thus introducing a time lag in the scheme and a *temporal partitioning error* that can be formulated as

$$\boldsymbol{f}^{\mathcal{F}}_{\Gamma,n+1}(\boldsymbol{d}^{\mathcal{S}}_{\Gamma,n}) \neq \boldsymbol{f}^{\mathcal{F}}_{\Gamma,n+1}(\boldsymbol{d}^{\mathcal{S}}_{\Gamma,n+1}). \tag{3.16}$$

As a consequence, the time-accuracy of the CSS algorithm is in general at least one order lower than that of its underlying single-field solvers, and the numerical stability limit of the coupled problem can be much more restrictive than that of the flow and/or solid solvers [147]. The limited stability and accuracy in time of the CSS algorithm is described more in detail in the following, while more evolved explicit coupling procedures are presented.

Before that, some terminologies commonly employed in the analysis of loosely-coupled partitioned procedures are briefly introduced. The term *collocated* or *synchronous* is used to designate algorithms such as the CSS in which the fluid and solid sub-systems are evaluated at the same time stations, as opposed to *non-collocated* or *asynchronous* schemes where both single-fields are reconciled at different time instants. These schemes are also classified as *serial* or *sequential* in the sense that the solid and the fluid are advanced successively and not simultaneously in time. In the latter case, the algorithm would be referred to as *parallel*. Various versions of these explicit schemes have been analyzed in the literature and are described in the following.

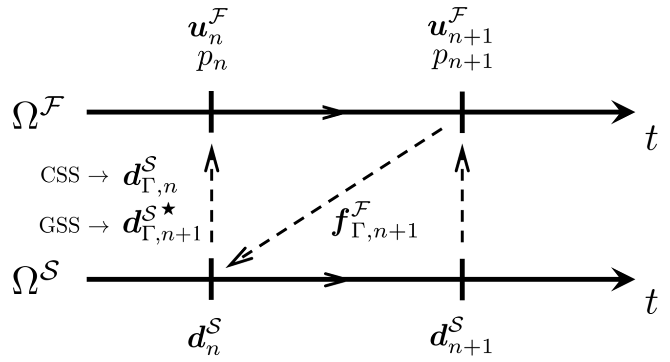
$\boldsymbol{u}_n^{\mathcal{F}}$
p_n
$\boldsymbol{u}_{n+1}^{\mathcal{F}}$
p_{n+1}
$\Omega^{\mathcal{F}}$
t
CSS → $\boldsymbol{d}_{\Gamma,n}^{\mathcal{S}}$
GSS → $\boldsymbol{d}_{\Gamma,n+1}^{\mathcal{S}\star}$
$\boldsymbol{f}_{\Gamma,n+1}^{\mathcal{F}}$
$\Omega^{\mathcal{S}}$
t
$\boldsymbol{d}_n^{\mathcal{S}}$
$\boldsymbol{d}_{n+1}^{\mathcal{S}}$

Figure 3.4: Schematic representation inspired by [47] of the conventional serial staggered (CSS) and generalized conventional serial staggered (GSS) coupling procedures for which the predicted displacement $\boldsymbol{d}_{\Gamma,n+1}^{\mathcal{S}\star}$ is given by (3.17).

3.3.2 GSS algorithm and artificial interface energy

In a first attempt to limit the above mentioned temporal error, Piperno and Farhat [46, 145] introduced a generalization of the CSS scheme by adding a solid predictor in the so-called *generalized conventional serial staggered* (GSS) algorithm. In that case, the solid displacement field at the interface transferred to the fluid solver in the step 2 of algorithm 3.1 is replaced by a predicted interface displacement field given by

$$\boldsymbol{d}_{\Gamma,n+1}^{\mathcal{S}\star} = \boldsymbol{d}_{\Gamma,n}^{\mathcal{S}} + \alpha_0\, \dot{\boldsymbol{d}}_{\Gamma,n}^{\mathcal{S}}\, \Delta t_{n+1} + \alpha_1 \left(\dot{\boldsymbol{d}}_{\Gamma,n}^{\mathcal{S}} - \dot{\boldsymbol{d}}_{\Gamma,n-1}^{\mathcal{S}} \right) \Delta t_{n+1} \tag{3.17}$$

where $\alpha_0 \in \mathbb{R}$ and $\alpha_1 \in \mathbb{R}$ are two parameters characterizing the order of time accuracy of the predicted displacement field, and the superscript $(\cdot)^\star$ designates a predicted quantity. The prediction (3.17) is 1st order time-accurate if $\alpha_0 = 1$ and $\alpha_1 = 0$ and 2nd order time-accurate if $\alpha_0 = 1$ and $\alpha_1 = 1/2$ [145]. The step 4 of algorithm 3.1 can then be interpreted as a correction of the predicted interface displacement, making the GSS a *predictor-corrector* approach. Although the introduction of a predictor allows to reduce the temporal partitioning error during each time step Δt_{n+1}, the interface kinematic continuity condition is still not fulfilled such that

$$\boldsymbol{d}_{\Gamma,n+1}^{\mathcal{S}} \neq \boldsymbol{d}_{\Gamma,n+1}^{\mathcal{S}\star}. \tag{3.18}$$

and results in the imbalance of the interface energy, as previously seen in section 3.2.2.3. This imbalance can also be interpreted as though an artificial energy were introduced by the partitioning process in the coupled system.

Artificial interface energy analysis

Piperno and Farhat analyzed in [147] the production of artificial energy for different loosely-coupled coupling schemes by comparing the work performed by the fluid pressure and stresses to that of the solid forces at the fluid-structure interface as a function of the coupling time step. Based on a similar idea developed in [48] for the data exchange through a non-matching interface, this approach enables to build a criterion that qualifies any kind of partitioned coupling scheme in terms of accuracy. It allows at the same time to measure the conservation of the work done by forces transmitted by the fluid to the solid sub-domain through the shared interface and, in that, corresponds to the energy criterion evoked in section 3.2.2.3. The monitoring of the artificial energy represents therefore a key component in the development of coupling schemes. The latter should aim at minimizing the artificial interface energy as it can deteriorate the accuracy of the coupled solution compared to the order of accuracy expected from both single-field solvers. Furthermore, such a spurious interface energy leads to unstable coupling schemes which, even if both single-field solvers are unconditionally stable, can be conditionally stable or even unconditionally unstable [52, 66]. Therefore, both the numerical stability and the accuracy of the coupled system are governed by the creation of interface energy, which can be controlled by the coupling scheme as shown in the following for different configurations.

Using equation (3.13), the energy transfer during Δt_{n+1} from the fluid to the solid field through the fluid-structure interface Γ, as viewed by the fluid, can be evaluated in the case of the GSS scheme as

$$\Delta E^{\mathcal{F}}_{\Gamma,n+1} = \boldsymbol{f}^{\mathcal{F}}_{\Gamma,(n+1)} \cdot \left(\boldsymbol{d}^{\mathcal{F}}_{\Gamma,n+1} - \boldsymbol{d}^{\mathcal{F}}_{\Gamma,n} \right), \tag{3.19}$$

which, by introducing the predictor (3.17) yields

$$\Delta E^{\mathcal{F}}_{\Gamma,n+1} = \boldsymbol{f}^{\mathcal{F}}_{\Gamma,(n+1)} \cdot \left(\boldsymbol{d}^{\mathcal{S}\star}_{\Gamma,n+1} - \boldsymbol{d}^{\mathcal{S}\star}_{\Gamma,n} \right). \tag{3.20}$$

The fluid force term[1] $\boldsymbol{f}^{\mathcal{F}}_{\Gamma,(n+1)}$ depends on the choice of time discretization scheme for the fluid solver and does not necessarily correspond to the fluid loads computed at the interface at time t_{n+1}. More precisely, it depends on the value of the fluid load that is used by the solver to compute the fluxes across the interface Γ, when it is advanced from time station t_n to t_{n+1} [147]. Different fluid time integration schemes are considered in [147] to compare loosely-coupled procedures, for which the fluid force term is given as follows:

- 1st order *forward Euler* explicit scheme:

$$\boldsymbol{f}^{\mathcal{F}}_{\Gamma,(n+1)} = \boldsymbol{f}^{\mathcal{F}}_{\Gamma,n} \tag{3.21a}$$

[1] The notations with parentheses around the time step index as in $\boldsymbol{f}^{\mathcal{F}}_{\Gamma,(n+1)}$ designate quantities used in the expression of the interface energy transfer during the time step Δt_{n+1} but are not necessarily computed at the time station t_{n+1}. All choices considered here for $\boldsymbol{f}^{\mathcal{F}}_{\Gamma,(n+1)}$ are given by (3.21).

– 1st order *backward Euler* implicit scheme:

$$\boldsymbol{f}^{\mathcal{F}}_{\Gamma,(n+1)} = \boldsymbol{f}^{\mathcal{F}}_{\Gamma,n+1} \tag{3.21b}$$

– 2nd order scheme: a good approximation consists in using

$$\boldsymbol{f}^{\mathcal{F}}_{\Gamma,(n+1)} \approx \frac{\boldsymbol{f}^{\mathcal{F}}_{\Gamma,n} + \boldsymbol{f}^{\mathcal{F}}_{\Gamma,n+1}}{2}, \tag{3.21c}$$

– In the event of fluid sub-cycling, the fluid load term can be approximated as

$$\boldsymbol{f}^{\mathcal{F}}_{\Gamma,(n+1)} \approx \frac{1}{\Delta t_{n+1}} \int_{t_n}^{t^{n+1}} \boldsymbol{f}^{\mathcal{F}}_{\Gamma}(t)\, \mathrm{d}t, \tag{3.21d}$$

assuming that the fluid interface velocity is constant over Δt_{n+1}.

The fluid procedure presented in section 2.2.5 and used in this thesis is 1st order time-accurate and computes the fluxes across the interface with the most recent fluid pressure and stresses such that the fluid force vector at the interface is given by (3.21b). In that case, equation (3.20) then reads

$$\Delta E^{\mathcal{F}}_{\Gamma,n+1} = \boldsymbol{f}^{\mathcal{F}}_{\Gamma,n+1} \cdot \left(\boldsymbol{d}^{\mathcal{S}\star}_{\Gamma,n+1} - \boldsymbol{d}^{\mathcal{S}\star}_{\Gamma,n} \right). \tag{3.22}$$

Moving on with the solid side, the explicit central difference 2nd order time-accurate scheme used in this work and presented in section 2.1.4 is considered. In that case, the variation of energy during Δt_{n+1} due to fluid forces on the shared interface as viewed by the solid can be written using equation (3.12) as

$$\Delta E^{\mathcal{S}}_{\Gamma,n+1} = -\frac{\boldsymbol{f}^{\mathcal{S}}_{\Gamma,(n)} + \boldsymbol{f}^{\mathcal{S}}_{\Gamma,(n+1)}}{2} \cdot \left(\boldsymbol{d}^{\mathcal{S}}_{\Gamma,n+1} - \boldsymbol{d}^{\mathcal{S}}_{\Gamma,n} \right), \tag{3.23}$$

where the terms $\boldsymbol{f}^{\mathcal{S}}_{\Gamma,(n)}$ and $\boldsymbol{f}^{\mathcal{S}}_{\Gamma,(n+1)}$ represent the *flow-induced solid load* applied by the solid solver on Γ at time instants t_n and t_{n+1} respectively. They are constructed based on the fluid pressure and stresses computed by the flow solver at the interface and represent in (3.23) the load applied by the 2nd order time-accurate solid solver during Δt_{n+1}. The way the solid load $\boldsymbol{f}^{\mathcal{S}}_{\Gamma,(n+1)}$ (or $\boldsymbol{f}^{\mathcal{S}}_{\Gamma,(n)}$ at the previous coupling time step) is built can hence define different instances of loosely-coupled procedures and has a large influence on the numerical stability and the interface energy conservation of the coupled FSI system. In that context, the solid load $\boldsymbol{f}^{\mathcal{S}}_{\Gamma,(n+1)}$ applied at the interface does not necessarily correspond to the most recent computed fluid load $\boldsymbol{f}^{\mathcal{F}}_{\Gamma,n+1}$. It can rather be interpreted as the time discretization of the load transfer from the fluid to the solid during Δt_{n+1} [146]. In [52], the solid load is even depicted as a *load corrector* that is supposed to rectify the displacement prediction given by (3.17). This terminology is used in the following to describe the solid force term, which can be constructed using the following choices proposed in [147]:

$$\boldsymbol{f}^{\mathcal{S}}_{\Gamma,(n+1)} = \boldsymbol{f}^{\mathcal{F}}_{\Gamma,n}(\boldsymbol{d}^{\mathcal{S}\star}_{\Gamma,n}), \tag{3.24a}$$

$$\boldsymbol{f}^{\mathcal{S}}_{\Gamma,(n+1)} = \boldsymbol{f}^{\mathcal{F}}_{\Gamma,n+1}(\boldsymbol{d}^{\mathcal{S}\star}_{\Gamma,n+1}), \tag{3.24b}$$

$$\boldsymbol{f}^{\mathcal{S}}_{\Gamma,(n+1)} = \frac{1}{2}\left(\boldsymbol{f}^{\mathcal{F}}_{\Gamma,n}(\boldsymbol{d}^{\mathcal{S}\star}_{\Gamma,n}) + \boldsymbol{f}^{\mathcal{F}}_{\Gamma,n+1}(\boldsymbol{d}^{\mathcal{S}\star}_{\Gamma,n+1})\right), \tag{3.24c}$$

$$\boldsymbol{f}^{\mathcal{S}}_{\Gamma,(n+1)} = 2\,\boldsymbol{f}^{\mathcal{F}}_{\Gamma,n+1}(\boldsymbol{d}^{\mathcal{S}\star}_{\Gamma,n+1}) - \boldsymbol{f}^{\mathcal{S}}_{\Gamma,n}, \tag{3.24d}$$

$$\boldsymbol{f}^{\mathcal{S}}_{\Gamma,(n+1)} = \frac{2}{\Delta t_{n+1}} \int_{t_n}^{t_{n+1}} \boldsymbol{f}^{\mathcal{F}}_{\Gamma}(t)\;\mathrm{d}t - \boldsymbol{f}^{\mathcal{S}}_{\Gamma,n+1}, \tag{3.24e}$$

where (3.24b) represents the natural choice for the classic GSS procedure and (3.24a) would allow the coupling scheme to work in parallel by solving both sub-problems concurrently as proposed e.g. in [45, 46]. This last option often leads to large errors at the interface and does not reduce substantially the simulation time if one solver needs much more time than the other to advance its system from t_n to t_{n+1}. In the framework of tire hydroplaning simulations, the solid solver is considerably faster than the flow solver to compute one coupling time step making such parallel coupling schemes irrelevant. They will therefore not be further discussed in this thesis.

Energy-accuracy of GSS procedures

Based on a load $\boldsymbol{f}^{\mathcal{F}}_{\Gamma,(n+1)}$ given by (3.21) that depends on the fluid solver at hand, various coupling schemes can be built with the aim of reaching the least energy imbalance at the interface. To do so, multiple combinations of predictor $\boldsymbol{d}^{\mathcal{S}\star}_{\Gamma,n+1}$ (3.17) and solid load corrector $\boldsymbol{f}^{\mathcal{S}}_{\Gamma,(n+1)}$ (3.24) can be chosen. Piperno and Farhat did so in [147] by estimating the energy error produced at the interface in each time step for all aforementioned choices by means of a parameter δE. In that context, a specific procedure is said to be p^{th} order *energy-accurate* as long as the coupling time steps Δt_{n+1} are assumed sufficiently small and if

$$\delta E \approx D\, h^p \qquad \text{with} \qquad h = \frac{\Delta t_{n+1}}{T_\mathrm{c}}, \qquad \text{when} \qquad h \to 0, \tag{3.25}$$

where D is a constant and T_c is a characteristic time associated with the FSI problem at hand. Applied on each prediction case given by (3.17), this analysis allows to define the following level of energy-accuracy:

- *Trivial predictor* $(\alpha_0 = \alpha_1 = 0)$, which corresponds to the CSS procedure: irrespective of the flow solver, the coupling scheme will always be 1st order energy-accurate except if the fluid sub-domain is integrated by an explicit 1st order scheme, i.e. (3.21a), and that the solid force term is given by (3.24b). In that particular case, the CSS algorithm is at least 4th order energy-accurate. However, this scheme does not conserve the momentum (3.10) at the interface and degrades accuracy in the event of fluid sub-cycling.

- *1st order prediction* $(\alpha_0 = 1$ and $\alpha_1 = 0)$: the coupling scheme is proved to be 2nd order energy-accurate for most of the single-field time discretization schemes. In particular, the combination of a fluid solver that computes the fluid forces as (3.21c) and a solid corrector chosen as (3.24e) provides the minimum error on the interface energy such that

  $$\delta E = -D_d \, \frac{5}{12} \, h^2 + \mathcal{O}(h^3), \tag{3.26}$$

  where D_d is a scalar parameter depending on added damping effects that result from the coupling of the treated FSI problem.

- *2nd order prediction* $(\alpha_0 = 1$ and $\alpha_1 = 1/2)$: 3rd order energy-accuracy can be achieved with a fluid solver that computes the loads at the interface as (3.21b) and a solid corrector (3.24d), in which case the error on the interface energy is given by

  $$\delta E = -D_k \, \frac{1}{3} \, h^3 + \mathcal{O}(h^4), \tag{3.27}$$

  where D_k is a scalar parameter that depends on added mass and stiffness that result from the coupling of the treated FSI problem. Note that this configuration enforces the conservation of momentum at the interface.

The mathematical framework based on the estimation of artificial energy created at the interface provides a very good tool to evaluate partitioned coupling procedures as well as their numerical stability and accuracy. As a matter of fact, it is shown in [147] for 2D and 3D examples that the higher the order of energy-accuracy, the more time-accurate and the more stable a coupling procedure is, in the sense that a higher time step can be employed for a given level of accuracy.

GSS procedures with current computational framework

Based on the present computational framework, a GSS coupling procedure complying with the analysis of energy-accuracy can be constructed. The non-conservation of the interface energy transfer during a time step Δt_{n+1} leads to the violation of equation (3.11) which, by considering the solid and fluid solvers presented in section 2 and expressions (3.22) and (3.23), can be written as

$$\begin{aligned} \Delta E_{\Gamma,n+1} &= \boldsymbol{f}^{\mathcal{F}}_{\Gamma,n+1} \cdot \left(\boldsymbol{d}^{\mathcal{S}\star}_{\Gamma,n+1} - \boldsymbol{d}^{\mathcal{S}\star}_{\Gamma,n} \right) - \frac{\boldsymbol{f}^{\mathcal{S}}_{\Gamma,(n)} + \boldsymbol{f}^{\mathcal{S}}_{\Gamma,(n+1)}}{2} \cdot \left(\boldsymbol{d}^{\mathcal{S}}_{\Gamma,n+1} - \boldsymbol{d}^{\mathcal{S}}_{\Gamma,n} \right) \\ &\neq 0. \end{aligned} \tag{3.28}$$

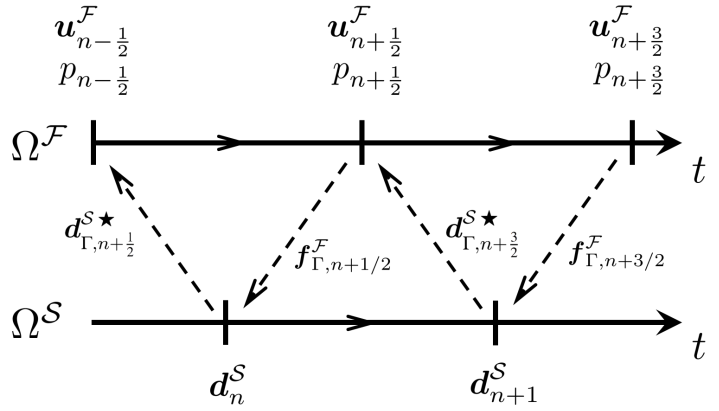
$\boldsymbol{u}^{\mathcal{F}}_{n-\frac{1}{2}}$
$p_{n-\frac{1}{2}}$
$\boldsymbol{u}^{\mathcal{F}}_{n+\frac{1}{2}}$
$p_{n+\frac{1}{2}}$
$\boldsymbol{u}^{\mathcal{F}}_{n+\frac{3}{2}}$
$p_{n+\frac{3}{2}}$
$\Omega^{\mathcal{F}}$
t
$\boldsymbol{d}^{\mathcal{S}\star}_{\Gamma,n+\frac{1}{2}}$
$\boldsymbol{f}^{\mathcal{F}}_{\Gamma,n+1/2}$
$\boldsymbol{d}^{\mathcal{S}\star}_{\Gamma,n+\frac{3}{2}}$
$\boldsymbol{f}^{\mathcal{F}}_{\Gamma,n+3/2}$
$\Omega^{\mathcal{S}}$
t
$\boldsymbol{d}^{\mathcal{S}}_{n}$
$\boldsymbol{d}^{\mathcal{S}}_{n+1}$

Figure 3.5: Schematic representation of the improved serial staggered (ISS) coupling procedure inspired by [46].

3.3.3 ISS algorithm and geometric conservation law

In many cases, FSI problems are tackled with moving mesh approaches such as the ALE formulation briefly presented in section 1.3.1. In order to be consistent, the computational modeling of the fluid sub-system must be able to predict exactly a uniform flow on a moving grid, which is achieved by computing the *mesh velocity* as[2]

$$\boldsymbol{u}^{\mathcal{F}}_{\Gamma,n+1} = \frac{\boldsymbol{d}^{\mathcal{F}}_{\Gamma,n+1} - \boldsymbol{d}^{\mathcal{F}}_{\Gamma,n}}{\Delta t_{\Gamma,n+1}}. \tag{3.29}$$

The expression (3.29) is commonly referred to as *geometric conservation law* (GCL) and was first introduced by Lesoinne and Farhat in [105]. It was shown in [50] and [79] that the violation of the GCL substantially deteriorates the numerical stability of the coupling procedure in the case of moving mesh approaches. In [105], Lesoinne and Farhat also showed that the kinematic continuity condition in terms of displacement (3.3) and velocity (3.2) cannot be simultaneously satisfied on the interface, while conserving the GCL. Given that both these kinematic conditions are desirable in order to avoid any discontinuities at the shared interface and that a violation of the GCL would severely limit the size of the coupling time step, Lesoinne and Farhat first introduced in [106] a non-collocated scheme named *improved serial staggered* (ISS). In this improved algorithm, the fluid sub-system is always computed at half discrete time stations ($t_{n-1/2}$, $t_{n+1/2}$), while the solid is still evaluated at instants (t_{n-1}, t_n, t_{n+1}) as illustrated in Figure 3.5. The successive steps of the ISS can be summarized as in algorithm 3.2. In [46, 106] the ISS procedure is proved to satisfy both continuity equations (3.2-3.3) without violating the GCL.

[2] The velocity and displacement fields in equation (3.29) are written here at the interface Γ to emphasize that they correspond to a mesh motion as, in this framework, $\boldsymbol{u}^{\mathcal{F}}_{\Gamma,n+1}$ and $\boldsymbol{d}^{\mathcal{F}}_{\Gamma,n+1}$ indeed describe the motion of the moving interface. In ALE approaches, not only the mesh motion at the interface but also in the rest of the fluid moving domain must fulfill expression (3.29).

Algorithm 3.2 ISS

```
Require: Initial time $t = T_0$, final time $t = T$, time step index $n = 0$, corresponding time step
    size $\Delta t_1$ and initial solid displacements $\boldsymbol{d}_0^{\mathcal{S}}$ and velocities $\dot{\boldsymbol{d}}_0^{\mathcal{S}}$.
while $t < T$ do
    A predicted interface displacement field computed as
        $$\boldsymbol{d}_{\Gamma,n+\frac{1}{2}}^{\mathcal{S}\star} = \boldsymbol{d}_n^{\mathcal{S}} + \frac{\Delta t_{n+1}}{2}\,\dot{\boldsymbol{d}}_{\Gamma,n}^{\mathcal{S}} \qquad (3.30)$$
    is transferred to the fluid solver.
    The fluid moving boundary is reconstructed within $\Omega^{\mathcal{F}}$ to the position is would have if
        the solid were advanced by $\boldsymbol{d}_{\Gamma,n+\frac{1}{2}}^{\mathcal{S}\star}$ and the fluid field is advanced from $t_{n-\frac{1}{2}}$ to $t_{n+\frac{1}{2}} =$
        $t_{n-\frac{1}{2}} + \Delta t_{n+1}$: $(\boldsymbol{u}^{\mathcal{F}}, p)_{n-\frac{1}{2}} \rightarrow (\boldsymbol{u}^{\mathcal{F}}, p)_{n+\frac{1}{2}}$.
    The fluid loads $\boldsymbol{f}_{\Gamma,n+\frac{1}{2}}^{\mathcal{F}}$ are computed at the interface as in (3.7) then transferred to the solid
        solver.
    The load is applied on the solid side of the interface and the solid field advanced to $t_{n+1}$:
        $\boldsymbol{d}_n^{\mathcal{S}} \rightarrow \boldsymbol{d}_{n+1}^{\mathcal{S}}$.
    Next time step: $n = n + 1$ and $t_{n+1} = t_n + \Delta t_{n+1}$.
end while
```

Energy-accuracy of ISS procedures

Analogously to the interface energy analysis for the CSS and GSS procedures, the energy-accuracy for different instances of the ISS algorithm can be assessed. In this case, the fluid quantities are computed (at least) at half time stations, which requires adapting the expression (3.20) for the energy transfer during a coupling time step Δt_{n+1} as viewed by the fluid with [147]

$$\Delta E_{\Gamma,n+1}^{\mathcal{F}} = \frac{1}{2}\,\boldsymbol{f}_{\Gamma,\left(n+\frac{1}{2}\right)}^{\mathcal{F}} \cdot \left(\boldsymbol{d}_{\Gamma,n+\frac{1}{2}}^{\mathcal{S}\star} - \boldsymbol{d}_{\Gamma,n-\frac{1}{2}}^{\mathcal{S}\star}\right) + \frac{1}{2}\,\boldsymbol{f}_{\Gamma,\left(n+\frac{3}{2}\right)}^{\mathcal{F}} \cdot \left(\boldsymbol{d}_{\Gamma,n+\frac{3}{2}}^{\mathcal{S}\star} - \boldsymbol{d}_{\Gamma,n+\frac{1}{2}}^{\mathcal{S}\star}\right). \tag{3.31}$$

The energy transfer, as viewed by the solid, remains unchanged as in (3.23). In this case, the interface energy imbalance for each time step Δt_{n+1} then reads

$$\begin{aligned}\Delta E_{\Gamma,n+1} = &\ \frac{1}{2}\,\boldsymbol{f}_{\Gamma,n+\frac{1}{2}}^{\mathcal{F}} \cdot \left(\boldsymbol{d}_{\Gamma,n+\frac{1}{2}}^{\mathcal{S}\star} - \boldsymbol{d}_{\Gamma,n-\frac{1}{2}}^{\mathcal{S}\star}\right) + \frac{1}{2}\,\boldsymbol{f}_{\Gamma,n+\frac{3}{2}}^{\mathcal{F}} \cdot \left(\boldsymbol{d}_{\Gamma,n+\frac{3}{2}}^{\mathcal{S}\star} - \boldsymbol{d}_{\Gamma,n+\frac{1}{2}}^{\mathcal{S}\star}\right) \\ &- \frac{\boldsymbol{f}_{\Gamma,(n)}^{\mathcal{S}} + \boldsymbol{f}_{\Gamma,(n+1)}^{\mathcal{S}}}{2} \cdot \left(\boldsymbol{d}_{\Gamma,n+1}^{\mathcal{S}} - \boldsymbol{d}_{\Gamma,n}^{\mathcal{S}}\right), \\ \neq &\ 0\end{aligned} \tag{3.32}$$

which also shows that the ISS procedure, same as the GSS, creates artificial energy at the interface.

Piperno and Farhat have shown in [147] that the ISS algorithm can be 2nd or 3rd order energy-accurate in the following cases:

- *2nd order energy-accuracy* is achieved by using a fluid time integration operator where the fluid load is computed as $\boldsymbol{f}^{\mathcal{F}}_{\Gamma,\left(n+\frac{1}{2}\right)} = \boldsymbol{f}^{\mathcal{F}}_{\Gamma,n-\frac{1}{2}}$ or $\boldsymbol{f}^{\mathcal{F}}_{\Gamma,\left(n+\frac{1}{2}\right)} = \frac{1}{\Delta t_{n+1}} \int_{t_{n-\frac{1}{2}}}^{t_{n+\frac{1}{2}}} \boldsymbol{f}^{\mathcal{F}}_{\Gamma}(t)\ \mathrm{d}t$ and by applying a solid force corrector at the interface as

$$\boldsymbol{f}^{\mathcal{S}}_{\Gamma,(n+1)} = 2\,\boldsymbol{f}^{\mathcal{F}}_{\Gamma,n+\frac{1}{2}}(\boldsymbol{d}^{\mathcal{S}\star}_{\Gamma,n+\frac{1}{2}}) - \boldsymbol{f}^{\mathcal{S}}_{\Gamma,(n)}. \tag{3.33}$$

- *3rd order energy-accuracy* is achieved by using a second-order fluid time integration scheme such that $\boldsymbol{f}^{\mathcal{F}}_{\Gamma,\left(n+\frac{1}{2}\right)} = \frac{\boldsymbol{f}^{\mathcal{F}}_{\Gamma,n-\frac{1}{2}}+\boldsymbol{f}^{\mathcal{F}}_{\Gamma,n+\frac{1}{2}}}{2}$ in combination with the same solid corrector as in (3.33).

By applying the same energy analysis as defined in (3.25), the order of energy-accuracy delivered by the ISS scheme equipped with the current fluid solver can only be of 1st order, irrespective of the choice of solid corrector $\boldsymbol{f}^{\mathcal{S}}_{\Gamma,(n+1)}$. This points out a limitation of the current approach if used with an ISS procedure. Yet, as further discussed in chapter 4, the coupling procedure of both solvers at hand does not allow to implement non-collocated coupling schemes as the ISS procedure, which makes this approach obsolete in this work anyway.

In general, ISS coupling procedures are definitely desirable for moving mesh approaches due to their potential high-order of energy-accuracy and the fulfillment of the GCL. However, as long as fixed-grid methods are considered for the resolution of the fluid sub-domain, GSS coupling schemes can achieve the same order of energy-accuracy as ISS schemes by appropriately choosing the prediction (3.17) and the solid corrector (3.24). As far as the order of energy-accuracy is concerned, no clear advantage of the ISS over the GSS can be deduced. Concerning the CSS procedure, its use proved to be quite restrictive as it can only be 1st order energy-accurate. It is noteworthy to mention that these conclusions were drawn for aeroelastic problems, where the solid to fluid mass density ratio is large or where the fluid flow is compressible. If it is not the case, other effects commonly known as artificial added-mass effects must be taken into account in order to construct explicit coupling schemes, as further discussed in section 3.3.5.2.

3.3.4 Recent advances and time-accuracy of coupling procedures

The interface energy has proved to be a very meaningful criterion in order to build partitioned coupling schemes and monitor their partitioning error as the simulation advances. Based on the classic staggered strategies discussed in the previous section, that is CSS, GSS and ISS coupling schemes, further research has been conducted in order to propose more advanced loosely-coupled procedures. Some of the most relevant works are reviewed in the following.

Time-accuracy of explicit partitioned procedures

Even though the order of time-accuracy can be related to the order of energy-accuracy, see [147], the time-accuracy of particular explicit coupling scheme had not yet been proven. It was eventually achieved by considering particular instances of single-field solvers chosen for their time integration schemes which, when coupled in a specific manner, could lead to a coupling scheme providing a provably order of time-accuracy. Contrary to Piperno and Farhat in [147], who proposed a generic way of analyzing coupling procedures for any combination of single-field solvers, these approaches are restricted to the use of specific single-field solvers. As a consequence, these coupling procedures are less easily transferable to the resolution of FSI problems in an industrial environment, which are in general based on black-box or even commercial single-field solvers with a given time integration scheme, as it is the case in this thesis. Nevertheless, for the sake of completeness, some of these coupling procedures proposed in the literature are briefly reviewed in the following.

Following up on their work in [147], Farhat et al. prove for the first time in [52] that 2nd order time-accurate and yet loosely-coupled solution algorithms can be designed to tackle aeroelastic FSI problems. In particular, it is shown that, as far as time-accuracy is concerned, loosely-coupled strategies can be successfully constructed for solving compressible flows and aeroelastic problems avoiding the complex implementation and computational cost of strongly-coupled procedures. Based on an implicit/implicit coupling approach (a three-point backward difference scheme for the flow and a trapezoidal rule for the solid), a GSS and an ISS algorithm are proved to achieve 2nd order time-accuracy by choosing a 2nd order predictor for the solid displacement and a 2nd order solid load corrector. The CSS procedure is also shown to be only 1st order time-accurate, which could be expected from the order of energy-accuracy of such algorithms presented in section 3.3.2. Applied successfully to the simulation of the aeroelastic response of a full 3D *F-16* fighter aircraft, these two coupling procedures also deliver good numerical stability, even though the latter is not formally proved in the study.

In [54], Farhat et al. extend the applicability of loosely-coupled procedures to explicit/explicit and explicit/implicit time integration schemes used for solving compressible FSI problems. The two proposed coupling schemes are proved to be 2nd order time-accurate and exhibit good numerical stability properties. The requirements on the single-field solvers are however higher than in [52] to achieve the wanted order of accuracy of the coupling algorithm, which is furthermore required to implement a 3rd order solid displacement predictor.

A coupling procedure with higher-order time-accuracy is proposed in [177] by van Zuijlen et al. with the *IMEX* (implicit/explicit) scheme. In this particular case, the fluid and solid sub-domains are integrated with an *implicit multi-stage Runge-Kutta* scheme, while only the solid load applied on the fluid-structure interface is computed with an *explicit Runge-Kutta* scheme. Applied to the flutter problem of a 3D wing model, a 3rd order accuracy was achieved by IMEX. Although very attractive for problems requiring a high temporal accuracy, e.g. to discriminate between physical and numerical divergence, this specific combination of single-field solvers is in practice not feasible with the use of independent black-box solvers.

All previously mentioned studies consider either compressible flows or large solid to fluid mass density ratios, such as in aeroelastic applications. For many, these are the only configurations in which an explicit coupling procedure can be used without leading to restrictive numerical instabilities and inaccuracies. In [37], Dettmer and Perić however proposed an unconditionally stable and 2$^{\text{nd}}$ order loosely-coupled procedure with use of implicit/implicit solvers, both based on a 2$^{\text{nd}}$ order *generalized-*α time integration scheme. Contrary to the GSS or ISS strategies, the solid is here evaluated before the fluid in each FSI time step. This particular coupling scheme is based on an *interface load predictor* that is corrected with the fluid forces computed by the flow solver and a user-defined scalar parameter. The latter is compared by the authors to the relaxation factor used in strongly-coupled Dirichlet-Neumann partitioned procedures introduced in section 1.3.2, which allows to control the added-mass effect. With help of a convergence study, it is shown that with an appropriate choice of the control parameter, this explicit coupling procedure can provide stable solutions in the presence of added-mass effects, while being computationally much more efficient than implicit solution strategies. A coupling algorithm with the same sequence of steps was proposed by Monasse et al. in [129] for computing 2D compressible flows and rigid solids with explicit/explicit approaches. In this case, pre-computed fluid fluxes evaluated together with predicted fluid loads before the solid steps are corrected in such a way that fluid mass conservation and balance of momentum and energy at the interface are enforced in each time step. The results were extended with success to the 3D case in [151].

The goal pursued in the works presented in this section consists in creating the most suitable coupling scheme for an FSI problem using any kind of single-field solvers. In this thesis that logic is inverted in the sense that the coupling scheme that is most suitable with the single-field solvers available must be implemented for solving numerically the tire hydroplaning problem. As already stated, these coupling schemes achieving 2$^{\text{nd}}$ or higher order of time-accuracy hence cannot be directly employed as they are in this work. Nevertheless, they provide a broader understanding of loosely-coupled partitioned procedures, which have been proposed so far in the literature and could represent a basis for future research.

3.3.5 Present computational framework

Before introducing in the next chapter the coupling procedure proposed in this thesis for solving industrial FSI problems, the previous review of loosely-coupled procedures is put into perspective with the current computational framework. This will allow to emphasize the requirements the coupling scheme should fulfill in order to couple in the best possible way the single-field solvers at hand. At this point, many observations were made that can be useful for constructing an appropriate coupling scheme. They are summarized in the following, while presenting more in detail the different sources of errors that originate from the partitioning of the coupled FSI problem. Important remarks on the artificial added-mass effect that affects FSI problems with an incompressible flow and small solid to fluid mass density ratios such as in the tire hydroplaning problem are then given to conclude this chapter.

3.3.5.1 Partitioning errors

While looking at creating partitioned coupling procedures, two main types of errors must be considered and minimized by the coupling scheme. These *partitioning errors* can either be geometric or temporal.

Geometric error

A geometric partitioning error is made through the approximation of the geometry of the fluid-structure interface. In the current approach, the external solid surface of the interface is approximated by the solid solver by a set of plane facets resulting from the use of 1^{st} order elements, as discussed in sections 2.1.3 and 2.1.5.1. In the fluid sub-domain, an approximation is in general made due to non-matching meshes at the interface, which require the interpolation of interface values from one sub-domain onto the other. As already stated in section 2.2.3, the geometry of an immersed boundary, which corresponds in this case to the shared interface, is exactly reconstructed within the fluid fixed-grid without any further approximation. This in turn allows the fluid solver to compute the fluid forces on each facet of the interface without any further interpolation, i.e. approximation. Consequently, the fluid approach, due to its reconstruction technique, creates a matching grid at the interface thus minimizing the overall geometrical error and enforcing automatically the kinematic coupling condition (3.3).

Since the only approximation in space is made exclusively at the level of the solid single-field solver and is in fact independent of the coupling procedure, the geometrical partitioning error can be considered nonexistent in the present approach.

Temporal error

As continuously discussed throughout the present chapter, loosely-coupled partitioned procedures introduce a temporal partitioning error due to the time shift between the two sub-domains which, as they cannot be simultaneously computed, leads to a violation of the continuity conditions (3.3) and (3.6). This inherent characteristic of explicit coupling schemes is the principal source of error that needs to be tackled in this thesis, while aiming at minimizing it.

Although this temporal error cannot be avoided without sub-iterations, it was shown that explicit coupling procedures can be constructed in order to achieve a level of accuracy that matches or even sometimes surpasses strongly-coupled strategies. Many of the approaches previously reviewed cannot be transferred to this work due to the specific combination of black-box solvers employed here. Nevertheless, it was shown that the numerical stability and the accuracy of the coupling procedure can be directly linked to its order of energy-accuracy, which hence consists of a rigorous criterion to control the temporal error due to the partitioning process. Ultimately, the goal consists in constructing a coupling scheme producing the least energy imbalance at the interface with both solvers at hand, while at the same time fulfilling efficiency requirements. Based on the quantitative estimation of that artificial quantity introduced by the staggering process into the coupled system, active actions can be taken in order to reduce the energy imbalance in subsequent time steps or speed up the simulation. The approaches used to address these topics, which represent the core subject of this thesis, are described in the next chapter where the coupling procedure proposed for the analysis of tire hydroplaning problems in an industrial framework is presented.

3.3.5.2 Artificial added-mass effect

Fluid-structure interaction problems with close solid and fluid mass densities ($\rho^{\mathcal{S}}/\rho^{\mathcal{F}}$ getting closer to 1) and involving incompressible flows, such as in the current application, suffer from the so-called *artificial added-mass effect*. It was already briefly introduced in section 1.3.2, while reviewing solutions proposed in the literature to address this limiting factor with help of strongly-coupled partitioned procedures. More recently, explicit approaches dealing successfully with problems affected by added-mass effects were introduced and are briefly reviewed in this section. Its impact on the current computational model is then discussed, which finally concludes the characterization of loosely-coupled procedures in this chapter.

The artificial added-mass effect is a numerical instability that was already noticed by Felippa et al. [56] in the early days of partitioned algorithms and was further described by Mok et al. in [127] and Le Tallec et al. in [104]. A mathematical foundation of the instability was provided by Causin et al. in [25], which was studied in a more general manner by Förster in [65, 66]. Its name originates from the fact that major parts of the fluid acts as an extra mass moving with the solid. In [56], a limit on the time step size was assumed to depend, among other things, on the speed of sound in the fluid medium. It is clear that, in the case of incompressible flows where an infinite wave speed exists, this observation predicts the instability irrespective of the chosen coupling time step. In [54], Farhat et al. clarified that the added-mass instability is definitely limited to FSI problems with incompressible flows, arguing that no difficulty associated to the instability is encountered while considering compressible flows.

Main factors affecting the added-mass effect have been established and the most relevant for the current work can be listed as follows:

- *Solid to fluid mass density ratio*: the smaller the ratio $\rho^{\mathcal{S}}/\rho^{\mathcal{F}}$, the earlier the instability occurs.
- *Coupling time step size*: the smaller Δt, the earlier the instability occurs.

- *Solid displacement predictor order*: the higher the order of $\boldsymbol{d}_{\Gamma}^{\mathcal{S}\star}$ given in (3.17), the earlier the instability occurs.
- *Non-collocated coupling strategies*: non-collocated coupling algorithms such as the ISS presented in section 3.3.3 are more subjected to the instability.
- *Fluid velocity*: higher $\boldsymbol{u}^{\mathcal{F}}$ have shown to increase the instability.
- *Solid stiffness*: stiff solid materials have shown to deliver better numerical stability.

While some parameters, such as the first and the last two items listed above, depend on the considered FSI problem and cannot be avoided, the coupling procedure can however have an influence on the numerical stability properties of the coupled system. In particular, the rather unusual situation that smaller time step sizes deteriorate the numerical stability, already warns about the use of too small time steps. A range for the time step size in which the added-mass instability is considered acceptable is thus expected. Furthermore, it can also be seen that the use of a solid predictor may be disadvantageous in terms of stability for problems facing the added-mass effect. These aspects will be investigated by means of numerical examples in the following chapters.

Many have argued in the past that loosely-coupled partitioned procedures applied to incompressible FSI problems are not feasible due to their lack of numerical stability. A first counter example already presented in section 3.3.4 is the explicit coupling algorithm by Dettmer and Perić in [37] that achieves unconditional numerical stability. Other groups working mainly on FSI problems with blood flows, characterized by high added-mass effects, have proposed explicit coupling procedures based on special interface conditions that confer improved numerical stability by modifying the single-field solvers. Among these, a stable explicit scheme for incompressible flows coupled to thin structures was first proposed by Guidoboni et al. in [78], where a *kinematically coupled scheme* using an operator splitting of the kinematic interface condition is employed. In this approach, the solid equations are split into a fluid load part, which is used as a boundary condition in the fluid sub-problem, and an elastodynamic part that is solved separately. This scheme was further developed e.g. in [23], which proved to be comparable to monolithic schemes in terms of accuracy while keeping the advantages of loosely-coupled schemes. Another noteworthy stable explicit coupling procedure is the so-called *added-mass partition* algorithm by Banks et al. proposed in two parts for elastic solid and shells elements in [9, 10]. This approach based on the construction of particular *Robin interface conditions* is shown to be stable without sub-iterations and to reach 2$^{\text{nd}}$ or even higher order of accuracy in time. In [107], the scheme is extended to the case of large interface motion for more general geometries. Although these coupling schemes are explicit in the sense that they only require one iteration per coupling time step, a full access to the single-field solvers is needed, which have to be, if not completely written from scratch, substantially modified. They are therefore not conceivable in an industrial framework and with the use of black-box solvers but once again highlight the fact that powerful explicit coupling schemes can be constructed even with high artificial added-mass effects.

To conclude, it can be asserted that the limited access to the pre-established single-field solvers represents a further challenge, while comparing the current computational model to many research studies reviewed herein. However, some ingredients to face the challenges and fulfill the requirements enumerated in this chapter have been exposed in order to obtain the most robust and yet time efficient coupling strategy for solving an FSI industrial application such as the tire hydroplaning problem. In particular, being aware of the added-mass instability and of the contributing factors listed above, an explicit coupling procedure fitted for the current application will be proposed in the next chapter.

4 A coupling scheme using black-box solvers for industrial applications

The coupling strategy implemented for solving the FSI problem of tire hydroplaning is presented in this chapter. A loosely-coupled procedure is used to couple both single-field solvers with help from a separate program specifically dedicated to manage the data exchange between the two black-box solvers for which only limited output data is available. This coupling platform is then used for developing new features intended to improve the coupled solution procedure based on the solver package at hand. Ultimately, the goal pursued is to construct a robust coupling scheme, which will produce the least energy imbalance at the interface, while remaining time efficient. Furthermore, it is shown here that the assessment of the produced temporal error through the estimation of the interface energy-based error provides a meaningful tool to monitor the level of inaccuracy resulting from the partitioning of the coupled system. This interface energy is also used as a criterion in order to take correcting actions in subsequent time steps by adapting the coupling time step size. These topics represent the core subject of this thesis and are described in this chapter.

After a presentation of the coupling scheme and corresponding implementation, the interface energy criterion used to control the accuracy and improve the efficiency of the solution procedure is described. These developments are then applied on two benchmark examples, which are used to validate the proposed coupling strategy and assess the feasibility of the proposed improvements to the coupling procedure. An academic problem commonly referred to in the literature and a simplified tire model rolling on a water layer are employed to that effect. The problem of hydroplaning with a full-scale real tire model will eventually be discussed in the next chapter.

4.1 From simple data exchange algorithm to flexible coupling scheme

Initially, in order to simulate the tire hydroplaning problem, the simplest strategy was used to exchange coupling data between the two single-field solvers presented in chapter 2. Similarly to the CSS procedure introduced in section 3.3.1, a set of subroutines embedded in the fluid solver was responsible for synchronizing the data exchange between the two black-box solvers by performing a single exchange per coupling time step, corresponding ultimately to the most basic two-way loosely-coupled partitioned procedure. These embedded coupling sub-routines are first to be replaced by a separate program that will take complete control over the coupling procedure between the solid and fluid simulation tools, allowing in turn to reach the overall objective of this thesis: implementing coupling strategies aiming at improving the simulation of the coupled problem of tire hydroplaning.

Fundamentally, the task of a coupling procedure is to transfer appropriate single-field outputs between both solvers at the right time during the FSI simulation. Even though it appears straightforward at first sight, the complexity can quickly increase in some particular situations. A first difficulty appears when the mesh at the fluid-structure interface is non-matching. In that case a so-called *transfer algorithm* [29] is used to map/interpolate coupling data onto both meshes at the interface prior to the actual data transfer. As a matching mesh is guaranteed at all times at the fluid-structure interface by the present computational framework, this particular situation is not further discussed here. The interested reader is however referred to section 1.3.1 where such transfer techniques are briefly reviewed for fluid moving-grid approaches. A second challenging situation can emerge while using black-box solvers for computing the solid and fluid sub-domains, due to the fact that they provide at best a limited access to the codes and even to their outputs. The synchronization of the data exchange requiring full-control to some of the most basic operations performed by the solvers can thus become difficult or even impossible.

The coupling of black-box solvers therefore remains a challenging task and requires the whole computational package to fulfill some requirements in order to solve multi-field problems. General requirements and recommendations valid for any solver with restricted access are given in the following. On this basis, the coupling shell implemented in the framework of this thesis to replace the initial coupling procedure is then presented.

4.1.1 Requirements for coupling black-box solvers

One of the main advantages in partitioned coupling approaches resides in the possibility of reusing existing solvers that have been developed to meet specific requirements. It is of particular interest when both physics are very complex to compute such as in the tire hydroplaning application that involves, naming just a few, nonlinear constitutive materials, a transient loading leading to large solid deformation in the contact patch region where the tire meets the road and the water, a rapidly changing free-surface flow due to fast running conditions or complex contact occurrences between the tire, the road and the rim. The solvers are chosen for their capabilities to handle the requirements and constraints from the problem at hand. While implementing a new solver for each application would represent the best scenario, it is in general not conceivable in practice, especially in an industrial environment. The favored alternative generally consists in carefully choosing the most appropriate solver among the large number of commercial tools available today on the market. In doing so, the single-field solvers presented in chapter 2 were chosen for computing the tire hydroplaning application. In particular, what ruled in favor of the solid solver is its ability to deal efficiently with large computational models featuring complex contact definitions and constitutive models with material properties obtained e.g. from lab measurements. Whereas the choice of fluid solver was primarily guided by its capabilities in terms of interface reconstruction with arbitrarily complex geometries and for its treatment of free-surface flows.

Not only must the solvers meet single-field requirements, they also need to be compatible for coupled computations. While the use of black-box solvers can represent a great advantage (if not the only possible choice) when complex physics are involved, an appropriate coupling of the solver package can be a real challenge. The communication and coordination of actions between the two solvers must therefore be properly handled, which is achieved by a so-called *coupling shell*. Most of the coupling shells are embedded subroutines designed to couple two specific solvers, see e.g. [49, 97], which often prevents the replacement of one solver by another. This is particularly true in the case of commercial black-box solvers for which the coupling shell is specifically designed to couple a well-defined combination of software. The other approach consists in using a separate program as coupling shell, which takes control of both solvers to synchronize their execution and manage the data exchange between them. The most prominent is the *Mesh-based parallel Code Coupling Interface* (MpCCI) software [67], developed at the Fraunhofer Institute for Algorithms and Scientific Computing SCAI and that allows to compute various multi-physics applications by coupling different simulation tools, including a limited number of commercial solvers. Even though it provides many advanced features that are continuously improved over the years, MpCCI remains a commercial product for which the source code is inaccessible. It is therefore not suitable for research and the development of new coupling techniques for a particular choice of black-box solvers. In the last few years, open-source coupling shells primarily used for research purposes have emerged, which provide a platform to implement new coupling strategies while taking care of the communication between the single-field solvers. One of these coupling shells is the so-called *FSI coupling environment* (FSI✻ce) [21, 22] developed at the TU Munich. Proposing several standard partitioned coupling schemes, it solely focuses on the coupling between Cartesian meshes. Another example is the *FLExible Coupling Shell* (FLECS) [134, 135] from TU Delft, which can be described as a generic and flexible coupling shell in the sense that it can be used to couple almost any kind of solvers written in different programming languages and essentially allows the user to only focus on the implementation of coupling schemes and, if needed, transfer algorithms. Due to its flexibility, the FLECS has been used in this work for replacing the coupling shell initially embedded in the fluid solver. Its operation and utilization in the framework of this thesis is further discussed in the next section.

It now remains that the chosen solvers must be compatible with the use of a separate coupling shell. As far as black-box solvers are concerned, the coupling shell is required to take control of the execution of both programs. The operations that the coupling shell must be able to perform in a loosely-coupled partitioned procedure can be summarized as follows:

- Starting the execution of the solvers with appropriate conditions to initiate their coupling.
- Forcing the solvers to hold until further notice before restarting for a subsequent simulation time step: while solver A is computing the solution on its sub-domain, solver B must be put on hold. Once the solution step carried out by solver A is finished, the execution of solver B must be restarted with updated coupling data resulting from solver A, which is in turn put on hold, and so on until the end of the coupled simulation.
- Extracting outputs from each single-field solver that take part in the data exchange of the coupled simulation, making them incidentally available in the coupling shell for further use.
- Transferring previously extracted outputs as well as eventual further coupling variables (e.g. coupling time step) to the solvers. Note that before being transferred, these data can be modified by the coupling shell.
- Terminating the execution of the solvers in a clean way.

By using the FLECS, the sequencing of these operations are carried out by simply adding a few subroutine calls to the code of each single-field solver. Although easy to apply, this approach is in general not practicable when truly black-box solvers are used, i.e. solvers for which the code is not at all accessible such as most of the commercial simulation tools[1]. The only remaining possibility for a separate coupling shell to interact with such black-box solvers is if the latter provide a communication channel to which the coupling shell can connect. This kind of communication channel generally takes the form of a *socket connection* that must be established between the solver and the coupling shell, and through which data is exchanged in a well-defined manner according to a predefined protocol. The solvers employed in this work are coupled using a socket connection, which will be further described in the next section.

If none of the aforementioned requirements are met for a chosen combination of black-box solvers, their connection through a separate coupling shell cannot be achieved. It is also important to add here that the coupling of commercial solvers through socket connection is not a straightforward task. As a matter of fact, it usually requires assistance from the code developers since the realization of the data exchange depends on a protocol that is in general not made available to the user. Furthermore, following a predefined protocol inexorably introduces limitations as to what kind of coupling variables can be exchanged and to what extent the execution of the solvers can be controlled. These aspects must be carefully considered before choosing the solvers best suited for computing a given multi-field problem. To close the subject, it can also be asserted that the creation of a universal coupling shell capable of connecting any simulation

[1] Nowadays, several software provide a so-called *application programming interface* (API) allowing the user to implement personalized applications by using a predefined set of subroutines and protocols defined within the code itself. Always more black-box codes provide an API, which could eventually be used to implement a coupling algorithm and communicate with an external solver. Note that, at this point, no API is provided in the solvers used in this work.

tool remains unfortunately an empty wish, at least as long as all commercial software developers do not allow for a minimum of access to their code or follow a common norm regulating the connection/communication between codes.

4.1.2 Coupling shell implementation

The separate coupling shell meant to replace the coupling procedure embedded in the fluid solver has been implemented based on the previously introduced platform called FLECS. Due to the black-box character of both single-field solvers, many FLECS operations cannot be used as intended, as they require adding subroutines calls directly to the codes. However, as shown here, the low-end communication features provided by FLECS can be used to build a separate coupling shell and synchronize the execution of both employed solvers.

FLECS is based on a client-server architecture in which a provider named *server* produces services for requesters also known as *clients*. In that context, the general design of FLECS can be decomposed into a *client library* that is called by each client, i.e. the single-field solvers, and a *coupling server* that coordinates the executions of the solvers and synchronizes the exchange of coupling data between them [134]. On the one side, the client library provides functions allowing for each client to connect to the coupling server as well as to exchange data with it. The latter is achieved by two standard operations: the first operation is responsible for sending data to the coupling server whereas the second receives data back from it. These two operations play a key role in controlling the sequence in which both solvers interact since the receive operation is blocking. In other words, the execution of client A is put on hold once it reaches a point where the receive operation is executed. It will not be restarted until client B completes the send operation and transfers data to client A. On the other side, the coupling server is responsible for setting up the connection with the clients, performing the actual data exchange and managing the structure of exchanged data. In the traditional way of using FLECS, functions provided by the client library to interact with the coupling server are directly included in the code of the single-field solvers. Since these are inaccessible with the computational package at hand, the client library can only been used within two separate programs, one for each solver, which act as intermediaries between the coupling server and the black-box solvers. The resulting coupling shell is then finally composed of these two programs, called here `clientSolid` and `clientFluid`, and of a coupling server, `server`, as illustrated in Figure 4.1. Both client programs are responsible for establishing a connection with their corresponding solver, exchanging data with it and controlling its execution (represented by white arrows in Figure 4.1). By using communication and coordination operations from FLECS, they can also interact with the coupling server (black arrows in Figure 4.1) in order to manage the execution sequence between both solvers. The coupling strategy employed within the coupling shell and the connection of the client programs with the black-box solvers is detailed separately in the next two sections.

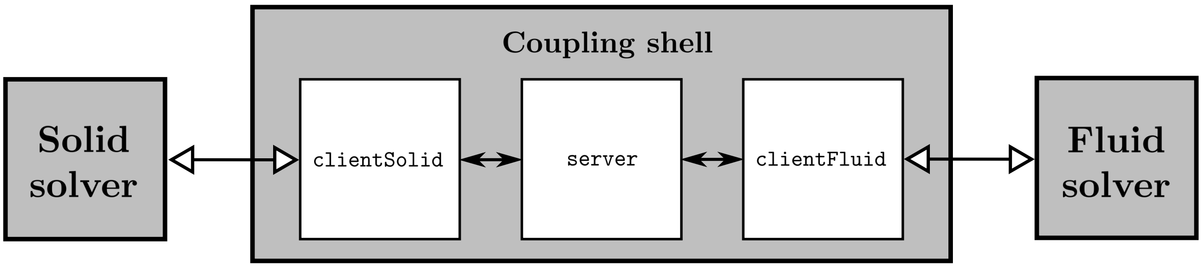

Figure 4.1: General architecture of the coupling shell coordinating the execution of the solid and fluid black-box solvers. Black arrows represent the communication channel within the coupling shell using FLECS operations. White arrows depict the connection between the coupling shell and separate solvers, which handles the data transfer and controls both simulation tools.

The presentation of FLECS given here is obviously far from complete and in no way representative of all its capabilities. The intent was merely to give a broad overview since essentially basic communication procedures provided by FLECS has in fact been used to build the proposed coupling framework. More details can however be found in e.g. [29, 135] as well as in the user-guide [172].

4.1.2.1 Coupling algorithm within the coupling shell

The general coupling algorithm implemented in the coupling shell for solving the FSI coupled system is based on three main phases as shown in Figure 4.2, where each box corresponds to a task performed either by the server or the clients. Boxes with dashed borders represent tasks involving FLECS operations, whereas double-border boxes require a communication between the coupling shell and the separate black-box solvers.

The first phase, INITIALIZATION, corresponds to the connection of the clients with the server and to the preparation of the FSI simulation. Once both clients are connected to the server using FLECS functions, a preliminary data exchange between both clients is executed with information that is needed on both sides of the interface. They include the initial coupling time step size Δt_0, which each solver will have to use for their first coupling iteration, the initial time of the FSI simulation T_0, the FSI simulation duration T as well as any information that must be shared by both solvers, for instance the path to the project files or specific solver options. Both solvers are then started on the cluster with the number of CPU's required by the user. If needed, an initial computation can be carried out for one or both sub-domains. In the case of the tire hydroplaning simulation, the solid tire model is built in multiple steps requiring preliminary computations before being able to solve the coupled problem where both tire and water interact. Each step of the solid model generation for a full-scale tire model is further discussed in section 5.1.

The actual FSI simulation is carried out in the second phase, during the so-called COUPLING LOOP, for each coupling time station t_n as defined in (3.8). The simulation will only stop once the simulation time T is reached or if an error occurred along the way. In each coupling iteration, the simulation time in both solvers is checked in order to avoid any time shift between both sub-domains at the beginning of a new coupling time step. The fluid sub-domain is then first computed for Δt_n, which can vary during the FSI simulation. Note that, in this configuration, both the solid and fluid fields can be freely sub-cycled according to (3.9) depending on their CFL requirements, guaranteeing already a certain level of numerical stability in both sub-domains. At that point, `clientSolid` is blocked on a FLECS receive operation and waits for the fluid solver to send updated coupling data. As soon as the fluid solver has finished solving the n^{th} coupling time step, coupling data (updated fluid forces at the interface) as well as any eventual outputs at the interface further required by the coupling shell for subsequent operations are extracted from the fluid solver and the updated coupling data is eventually sent to the other client. At this point, it is now `clientFluid` that blocks on the receive operation. The coupling data resulting from the fluid computation is then imported into the solid solver, which carries on with solving the n^{th} coupling time step. Same as on the fluid side, needed outputs are extracted from the solid solver and coupling data (updated position of the interface) transferred to the other client, which can then restart. Before importing updated solid coupling data into the fluid solver and starting the next coupling time step, eventual modifications are performed on coupling data before jumping to the next coupling time step. These modifications, which are made possible by the proposed coupling approach, include for example the use of a predictor-corrector technique as in the GSS algorithm presented in section 3.3.2 or a varying coupling time step size as discussed in section 4.2.

The last phase, denoted by END in Figure 4.2, consists in terminating the FSI simulation if the end time is reached or if an error occurred in one of the two single-field solvers. In that event, the coupling server informs both clients to kill their respective solver (if still running) in a clean way and then disconnects from both clients before shutting itself down.

4.1.2.2 Connection and data exchange with black-box solvers

Now that a solution has been found to coordinate the execution of both solvers through a separate coupling shell, a communication channel must still be established between the black-box solvers and the intermediary client programs `clientSolid` and `clientFluid` (white arrows in Figure 4.1). To do so, two approaches were used, each one with their advantages and disadvantages. The first approach achieves the communication between the client programs and the solvers through a *flat file exchange*. The second technique is based on a *socket connection* established by the coupling shell between both solvers. While the first approach is much easier to implement and provides more flexibility in controlling both solvers, the second is much faster and represents the only option that can realistically be implemented in an industrial environment. Both approaches are described in the following.

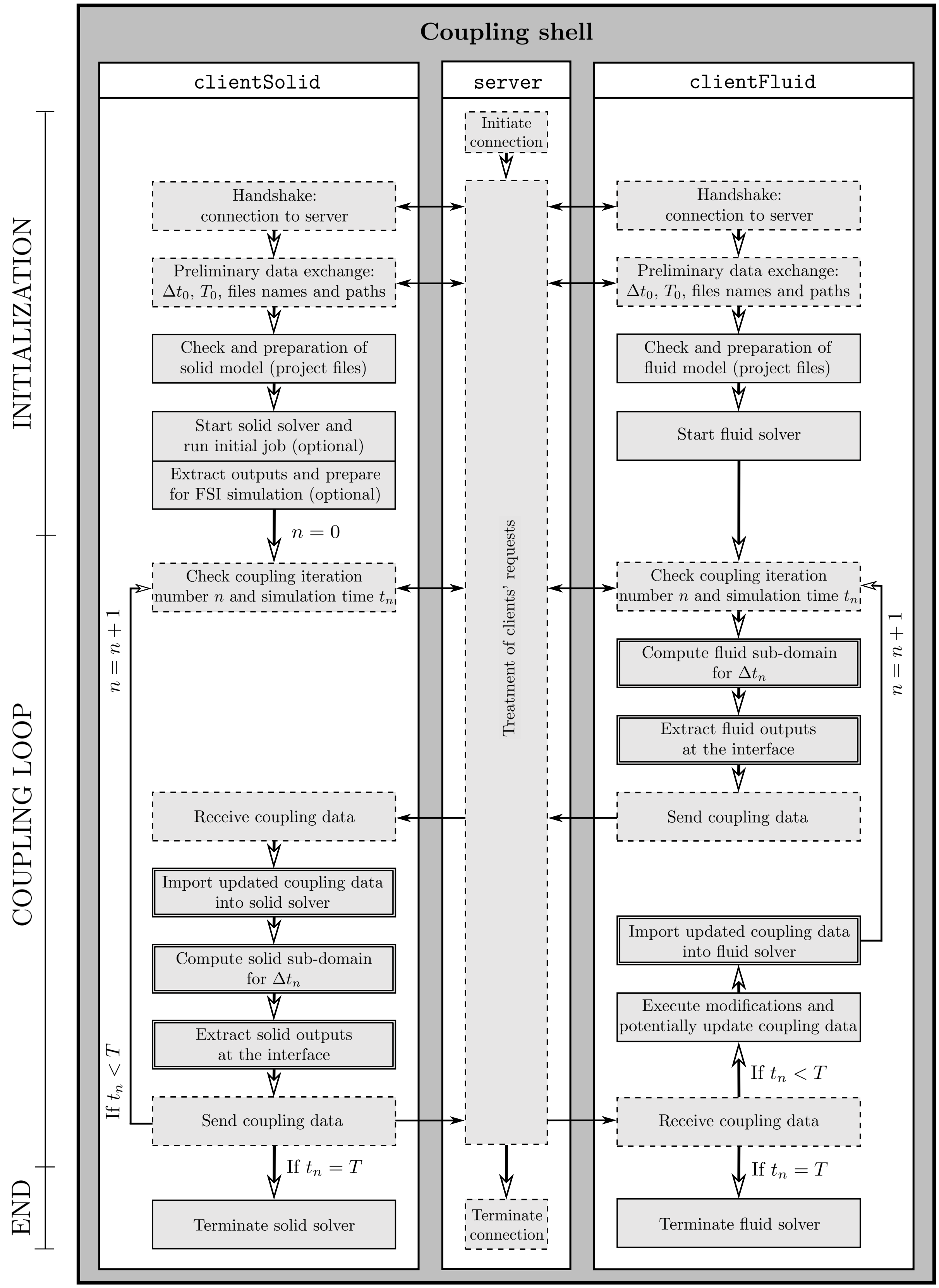
Coupling shell
clientSolid
server
clientFluid
INITIALIZATION
COUPLING LOOP
END
Initiate connection
Handshake: connection to server
Handshake: connection to server
Preliminary data exchange: Δt_0, T_0, files names and paths
Preliminary data exchange: Δt_0, T_0, files names and paths
Check and preparation of solid model (project files)
Check and preparation of fluid model (project files)
Start solid solver and run initial job (optional)
Extract outputs and prepare for FSI simulation (optional)
Start fluid solver
$n = 0$
Treatment of clients' requests
Check coupling iteration number n and simulation time t_n
Check coupling iteration number n and simulation time t_n
$n = n + 1$
$n = n + 1$
Compute fluid sub-domain for Δt_n
Extract fluid outputs at the interface
Receive coupling data
Send coupling data
Import updated coupling data into solid solver
Compute solid sub-domain for Δt_n
Extract solid outputs at the interface
Import updated coupling data into fluid solver
Execute modifications and potentially update coupling data
If $t_n < T$
If $t_n < T$
Send coupling data
Receive coupling data
If $t_n = T$
If $t_n = T$
Terminate solid solver
Terminate connection
Terminate fluid solver

Figure 4.2: Schematic representation of the coupling strategy within the coupling shell, where boxes with dashed borders represent tasks involving FLECS operations and double-border boxes require a communication between the coupling shell and the separate black-box solvers.

Flat file exchange

The first way of controlling the black-box solvers from the separate coupling shell and of transferring data is achieved by a flat file exchange. In this case, the solvers are completely controlled by the coupling shell by being shut down and restarted for each coupling time step. There is indeed no other way of proceeding if the black-box software do not allow the user to introduce subroutines within its code. At the end of each coupling time step, the outputs on the interface are written by one of the solvers in a flat file, which can then be imported into the other solver and restart for the next step with updated data. That way, if any modification is to be performed on the coupling data, it can simply be achieved by properly replacing values in the appropriate flat files. The coupling shell might also need to adapt the format of the flat file in order to be readable by the solver into which it is imported. This method is very easy to implement and was first used in order to check the feasibility of the project. Since both solvers must be shut down and restarted in each step of the coupling loop, all solver parameters can be modified by the coupling shell providing the highest level of flexibility for controlling black-box solvers. In that scenario, an implicit coupling is even conceivable, in which case the solver would simply need to rerun for the same coupling time step instead of advancing to the next FSI time station. The largest issue with this approach is that the process of restarting any kind of software can be very long since the entire project must be reloaded and the solvers are usually not designed to do it in an efficient manner. As a result, this first approach was considered insufficient and not appropriate for coupling both solvers.

Socket connection

The second approach, which is eventually adopted in this work, consists in using an existing communication channel that initially operated the data exchange between the two single-field solvers and was controlled by a program embedded in the fluid solver. This communication channel takes the form of a socket connection, which is very commonly used by programs to communicate with one another on a network. A socket is the endpoint of such a two-way communication channel, which is also based on a client-server model: a socket server listens for requests made by clients on the other end of the socket connection. Each single-field solver is associated to one of these sockets at one node of the network, which is identified by a unique combination of IP address and port number. In the present case, the solid solver provides the socket server, while the fluid solver that controls the connection acts as the client as illustrated in Figure 4.3(a). Once the connection is established between the socket server and the client, both solvers can communicate by writing to or reading from their sockets. In order to guarantee a proper data exchange, both solvers are bound to follow a shared protocol that dictates in which sequence each transfer operation must be performed on the socket. In short, after a handshake between the two solvers, the protocol is built such that in each iteration of the coupling loop the following operations are performed: first the coupling time step size is exchanged between both solvers; then, as soon as the fluid step is completed, the fluid forces at the interface are

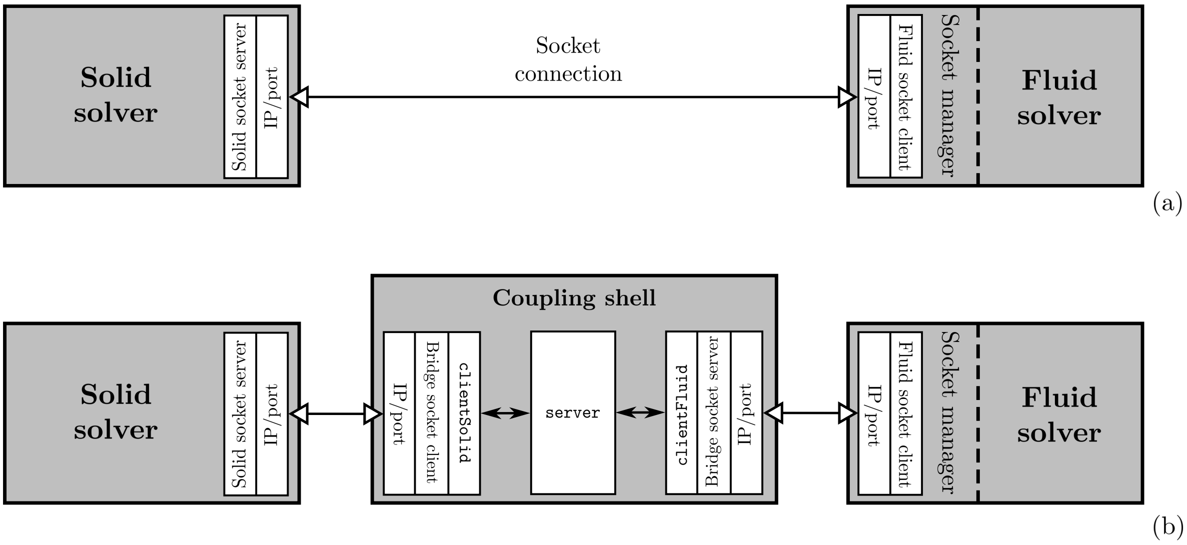

Figure 4.3: (a) Initial coupling procedure based on a socket connection to exchange data between both single-field solvers and managed by the fluid solver. (b) New coupling procedure achieved by inserting the separate coupling shell on the communication channel between both solvers, which provides more flexibility in handling the data transfer and controlling both solvers.

sent on the socket in the form of a data array to the solid solver; finally the solid solver sends the coordinates of the new position of the interface once the solid run is finished; and so on for each step of the coupling loop until the simulation ends. In this initial version of the coupling procedure, there is unfortunately no possibility for an external user to access the data traveling along the socket connection, which makes this approach, as it is, not suitable for implementing new developments.

To be able to control the solvers and modify coupling data, the proposed idea consists in inserting the coupling shell introduced in 4.1.2.1 between the two solvers by connecting it to sockets made available by the two single-field solvers. It has been realized in practice by implementing a code named *bridge* that provides a socket server (bridge socket server) and a client (bridge socket client), and which is placed on the communication channel between both solvers as represented in Figure 4.3(b). While both the solid and fluid solvers think they communicate with one another, they actually exchange data with the bridge, which is now able to modify the data transferred on the sockets. The synchronization of each exchange operation is achieved within the coupling shell by reproducing exactly the aforementioned protocol. In doing so, the coupling shell can have full control on the data exchange, while being only limited by the sequence foreseen in the protocol. This limitation for example excludes the use of an ISS algorithm described in 3.3.3 or any kind of implicit coupling procedures.

Ultimately, a generic coupling framework for computing FSI problems with the current combination of black-box solvers has been built. Even though many limitations persist in terms of coupling scheme that can be implemented, the proposed coupling shell allows to fully operate a loosely-coupled partitioned procedure. Among the new features offered by the new coupling shell are the possibility of using a solid predictor as in (3.17), a fluid corrector as in (3.24) as well as a dynamic adaptation of the coupling time step size Δt_n. It is important to stress that a socket connection can only be realistically considered if the employed software provide the capability. Furthermore, the implementation of the socket connection is not an easy task and necessitates information usually not made available to every commercial code users. For instance, the sequence of actions included in the protocol had to be reproduced within the bridge, which could only be achieved by knowing exactly how the protocol was implemented. Thanks to a great collaboration with the fluid solver developers' team, the connection through the bridge could be achieved successfully. It should also be mentioned here that a large support has been provided by Romain Pennec, Goodyear researcher also doing a PhD at the Technical University of Munich, in order to achieve the implementation of the coupling procedure based on socket connections.

4.2 An interface energy-based criterion for the accuracy and efficiency of coupling procedures

Using the coupling shell introduced in the previous section, the most appropriate coupling scheme for solving industrial FSI applications, and in particular the tire hydroplaning problem, is to be implemented with the solver package at hand. Although limited to a loosely-coupled partitioned procedure, the new coupling framework allows to use and modify the coupling data in order to improve the preexisting coupling scheme and in turn the solution to the coupled problem. In doing so, a quantity advising on the level of error artificially introduced by the coupling scheme in the coupled system is used as a criterion to control the level of accuracy and improve the robustness of the coupled solution procedure. This quantity is based on the concept of interface energy already introduced in section 3.3.2. As shown in the rest of this chapter, such an interface energy criterion has a twofold advantage: it can be used to guarantee a certain level of accuracy but also to improve the time efficiency of the solution procedure. As the proposed work is supposed to be applied in an industrial framework, where the simulation time for long lasting jobs is of significant importance, the objective will ultimately consist in finding the most time efficient approach while keeping a sufficient level of accuracy. In the remaining of this section, the formulation of the interface energy for the current combination of solvers is described. It is then followed by the presentation of the criterion employed to optimize the time efficiency of the procedure, which is achieved by dynamically adapting the coupling time step size.

4.2.1 Formulation of energy imbalance at the fluid-structure interface

As discussed in chapter 3, the evaluation of the energy artificially introduced at the interface by the coupling procedure represents a rigorous criterion to assess the accuracy and numerical stability of the employed approach and in turn allows to control the related temporal error. As exposed in section 3.3.5.1, the only partitioning error that is of concern in this thesis is the temporal error, whereas the geometric error is minimized by the present computational framework. Among the loosely-coupled procedures presented at that occasion, it was already made clear that the ISS algorithm as well as all techniques requiring an access to the solvers' code could not be used with the current coupling strategy and combination of solvers. The remaining approach that can be considered is the CSS procedure (algorithm 3.1). It can be generalized to the GSS procedure described in 3.3.2 by considering a predictor on the solid displacement at the interface and a corrector on the fluid force transmitted to the solid. The pursued goal eventually consists in constructing a GSS coupling scheme, which will produce the least energy imbalance at the interface or at least keep it within a certain range before the simulation diverges.

In that context, the energy imbalance corresponding to the non-conservation of the interface energy transfer during a time step Δt_{n+1} is reminded here as given in equation (3.28) for the pair of solvers at hand:

$$\Delta E_{\Gamma,n+1} = \boldsymbol{f}^{\mathcal{F}}_{\Gamma,n+1} \cdot \left(\boldsymbol{d}^{\mathcal{S}\star}_{\Gamma,n+1} - \boldsymbol{d}^{\mathcal{S}\star}_{\Gamma,n} \right) - \frac{\boldsymbol{f}^{\mathcal{S}}_{\Gamma,(n)} + \boldsymbol{f}^{\mathcal{S}}_{\Gamma,(n+1)}}{2} \cdot \left(\boldsymbol{d}^{\mathcal{S}}_{\Gamma,n+1} - \boldsymbol{d}^{\mathcal{S}}_{\Gamma,n} \right) , \tag{4.1}$$

for which different possibilities can be considered for the solid predictor (3.17) and the fluid corrector given by (3.24). If not specified otherwise, a 0th order predictor given by (3.17) with $\alpha_0 = \alpha_1 = 0$ and the corrector (3.24b) are used. With $\Delta E_{\Gamma,n+1} \neq 0$ for any loosely-coupled procedure, an estimator-based indicator is obtained that instructs on the level of error introduced in the system by the coupling scheme. Since the contribution of both sub-domains is included in the formulation of the energy imbalance (4.1), the latter represents indeed a very meaningful tool to characterize the overall coupled problem and to quantitatively measure the produced temporal error.

Equation (4.1) delivers a measure of the energy artificially introduced along the entire interface in each time step n. A local version of the energy imbalance advising on the level of error produced at each node i of the interface at the time station t_{n+1} can also be written, which then reads

$$\Delta E^{\mathrm{l}}_{\Gamma,i,n+1} = f^{\mathcal{F}}_{\Gamma,i,n+1} \cdot \left(d^{\mathcal{S}\star}_{\Gamma,i,n+1} - d^{\mathcal{S}\star}_{\Gamma,i,n} \right) - \frac{f^{\mathcal{S}}_{\Gamma,i,(n)} + f^{\mathcal{S}}_{\Gamma,i,(n+1)}}{2} \cdot \left(d^{\mathcal{S}}_{\Gamma,i,n+1} - d^{\mathcal{S}}_{\Gamma,i,n} \right) \tag{4.2}$$

and can be stored into a vector as

$$\Delta \boldsymbol{E}^{\mathrm{l}}_{\Gamma,n+1} = \begin{pmatrix} \Delta E^{\mathrm{l}}_{\Gamma,1,n+1} \\ \Delta E^{\mathrm{l}}_{\Gamma,2,n+1} \\ \vdots \\ \Delta E^{\mathrm{l}}_{\Gamma,\mathrm{N}^{\Gamma}_{\mathrm{nodes}},n+1} \end{pmatrix} , \tag{4.3}$$

where the superscript $(\cdot)^{\mathrm{l}}$ denotes the local form and $\mathrm{N}^{\Gamma}_{\mathrm{nodes}}$ is the number of nodes shared by both sub-domains on the fluid-structure interface. As discussed in the following, this quantity will be used in this work to characterize the error made by the coupling scheme.

Not only can the energy at the interface be used to control that error, it can also be taken advantage of in order to take corrective actions in subsequent FSI time steps. On the one hand, a high energy imbalance highlights a deficit in accuracy and numerical stability at a certain time instant in the coupling procedure. It can be corrected by advancing more cautiously in time by reducing the FSI time step size. On the other hand, a low energy imbalance indicates that the simulation is currently evolving within a safe stability window, which can be taken advantage of by increasing the FSI time step size and in turn speed up the computation. An automated procedure to reach an optimized efficiency level by dynamically adapting the FSI time step size while controlling the error level has been implemented in the coupling shell described in 4.1.2 and is presented in the next section.

4.2.2 Dynamic adaptation of the time stepping procedure

The assessment of the interface energy in each coupling time step from equations (4.1) or (4.3) provides a quantitative measurement of the temporal error in the coupled system at a given time during the simulation. It can therefore be appropriately used to control the course of the simulation and, in particular, optimize the FSI time step size. The procedure presented in the following is said to be *dynamic* in the sense that the time step size will be automatically modified over the course of the simulation.

4.2.2.1 An error-based approach

For computations with a transient response, as all applications considered in this thesis, the choice of the FSI time step size Δt_{n+1} used to discretize in time the coupled problem is not an easy task. If the coupling time step size is chosen too large, it can be detrimental in terms of accuracy and numerical stability of the solution procedure. If chosen too small, not only will that lead to evident efficiency problems but it can also deteriorate the numerical stability of the coupling scheme due to an increased artificial added-mass effect as discussed in section 3.3.5.2. A compromise must therefore be found to obtain the optimum FSI time step size. The latter can be directly linked to an error, which describes the deviation of the numerical solution from the exact solution. For problems with a transient response, that level of error can fluctuate over the course of the simulation such that a variable size of the coupling time step Δt_{n+1} can be desirable.

The still unanswered question is how some error estimation can be correlated to the time step size in each iteration of the coupling loop. To do so, a similar idea as the one proposed by Mayr in [119] for monolithic solvers is followed but in a different context. In [119], the estimation of the error is carried out for the solid and the fluid fields separately, depending on their respective accuracy requirements. The optimum time step size retained for advancing the system to the next time station is chosen as the smallest value obtained between both single-fields, as sub-cycling is not permitted by monolithic approaches. In the present case, the error estimation is based on the assessment of the interface energy, which is a quantity measured for the whole coupled problem and eventually leads to a unique coupling time step size. While Mayr's approach considers the requirements for each single-field separately, the current procedure considers the partitioning error introduced by the coupling scheme to find an optimum time step size for each coupling iteration. Even though the adaptive time stepping procedure was initially designed for a monolithic scheme, the general concept can be transferred to the current loosely-coupled partitioned approach and is outlined in the following. The intent is not in any way to cover the very broad subject of error estimation. For that subject matter, the interested reader is encouraged to consult the thesis by Mayr [117] and references therein.

4.2.2.2 Optimum time step size

When the temporal error is estimated for each single-field as in [119], the expression of the sought-for optimum time step size depends on the employed time integration scheme and, in particular, on the order of convergence of that scheme. In the present case, since the error estimation is based on some quantity that merges both sub-domains, such a criterion cannot strictly be used. Moreover, while Mayr's adaptive time stepping algorithm can repeat the current FSI time step if the error estimate is not fulfilled, the present scheme checks the level of error at the end of every time step and modifies the subsequent coupling time step size if needed. The adaptation procedure allows to take preventive actions based on the current level of error in the coupled system, thus improving the robustness of the solution procedure. The optimum coupling time step size to advance to the next time station can be approximated by

$$\Delta t_{n+1} \approx \kappa_n^{\circledcirc} \, \Delta t_n, \tag{4.4}$$

where the superscript $(\cdot)^{\circledcirc}$ designates optimized values and $\kappa_n^{\circledcirc}$ is an *optimized scaling factor* that is used to increase or decrease the time step size. The latter can then be linked to the interface energy imbalance $\Delta\boldsymbol{E}^{\mathrm{I}}_{\Gamma,n}$ evaluated in each time step as in (4.3) as

$$\kappa_n^{\circledcirc} \approx \kappa_{\mathrm{s}} \sqrt{\frac{\epsilon}{||\Delta\boldsymbol{E}^{\mathrm{I}}_{\Gamma,n}||_{\mathrm{L}_2}}}, \tag{4.5}$$

where ϵ is a user-defined *tolerance* with energy dimensions representing the wanted level of accuracy and κ_{s} a *safety factor* keeping the error away from the tolerance limit. The denominator in (4.5) represents the estimate on the temporal error made by the coupling scheme. This *interface energy-based error* is built around a *scaled discrete L_2 norm* of the vector (4.3), which is defined as

$$||\Delta\boldsymbol{E}^{\mathrm{I}}_{\Gamma,n}||_{\mathrm{L}_2} = \sqrt{\frac{\sum_{i=1}^{\mathrm{N}^{\Gamma}_{\mathrm{nodes}}} \left(\Delta E^{\mathrm{I}}_{\Gamma,i,n}\right)^2}{\mathrm{N}^{\Gamma}_{\mathrm{nodes}}}} \tag{4.6}$$

for which the scaling by the number of nodes shared by both sub-domains on the FSI interface $\mathrm{N}^{\Gamma}_{\mathrm{nodes}}$ provides a quantity independent of the mesh resolution. It is commonly used in computer programs for evaluating errors and is employed here for measuring the energy imbalance.

In practice, a well-established strategy is employed to find the optimum time step size for the subsequent coupling iteration Δt_{n+1}, which is kept within a user-defined range such that it is eventually computed as

$$\Delta t_{n+1} = \min\{\, \Delta t_{\max},\, \max\{\, \min\{\, \kappa_{\max},\, \max\{\, \kappa_{\min},\, \kappa_n^{\circledcirc} \,\}\}\, \Delta t_n,\, \Delta t_{\min} \,\}\}. \tag{4.7}$$

This expression simply provides boundaries for the FSI time step size, which can be described as follows. The optimized scaling factor $\kappa_n^{\circledcirc}$ modifying the time step size is bounded by minimum and maximum values such that $\kappa_{\min} \leq \kappa_n^{\circledcirc} < \kappa_{\max}$, whereas the resulting time step size is further limited by a user-defined lower and upper bounds, $\Delta t_{\min}$ and $\Delta t_{\max}$ respectively.

Smoothing of time step size adaptation

In order to avoid an erratic change of the time step size over the course of the simulation, which can deteriorate the numerical stability, a filter that allows to smooth the evolution of the time step size is applied. A simple filter consisting of an averaging procedure is used here: the next optimum time step size is not directly computed as in (4.7) but a finite number of lastly computed values are averaged. In the case of a time step size increase, the last $\mathrm{M}_{\mathrm{up}} \in \mathbb{N}$ time steps are taken into account whereas, when it decreases, the last $\mathrm{M}_{\mathrm{down}} \in \mathbb{N}$ time steps are considered such that

$$\Delta t_{n+1} = \begin{cases} \frac{1}{\mathrm{M_{up}}} \sum\limits_{m=0}^{\mathrm{M_{up}}} \Delta t_{n+1-m} & \text{if } \Delta t_{n+1} \geq \Delta t_n, \\ \frac{1}{\mathrm{M_{down}}} \sum\limits_{m=0}^{\mathrm{M_{down}}} \Delta t_{n+1-m} & \text{if } \Delta t_{n+1} < \Delta t_n. \end{cases} \tag{4.8}$$

As a general rule, $\mathrm{M_{up}}$ can be chosen as high as needed by the coupled problem of interest in order to smooth the process of time step size increase. However, while decreasing the time step size, $\mathrm{M_{down}}$ must be chosen small and close to 1. Indeed, the time step size decrease is provoked by a violation of the wanted level of accuracy and should therefore be applied without too much delay.

Adaptive time stepping procedure and algorithm parameters

To summarize, the adaptive time stepping procedure is ultimately included in the coupling loop presented in Figure 4.2 in the end of each coupling iteration n, when all needed data is available to evaluate the error corresponding to the energy imbalance $\Delta \boldsymbol{E}^{1}_{\Gamma,n}$. The optimum time step size applied in the next coupling iteration $(n+1)$ is then found by computing (4.7) using expression (4.5) and the smoothing (4.8), which requires establishing the following set of algorithm parameters:

$$\{\, \Delta t_{\mathrm{min}},\, \Delta t_{\mathrm{max}},\, \epsilon,\, \kappa_{\mathrm{s}},\, \kappa_{\mathrm{min}},\, \kappa_{\mathrm{max}},\, \mathrm{M_{up}},\, \mathrm{M_{down}} \,\}. \tag{4.9}$$

In the first coupling iteration, the time step size is chosen as the minimum such that $\Delta t_1 = \Delta t_{\mathrm{min}}$, which must be chosen in order to fulfill accuracy requirements. The time step size is then expected to increase in the subsequent coupling iterations and eventually reach an optimum. The choice of parameters is illustrated in the next section, where numerical examples are presented.

4.3 Numerical examples

The developments previously introduced to improve the preexisting coupling strategy with the coupling shell presented in section 4.1 and the solver package described in chapter 2 are illustrated by means of two benchmark examples. In the first example, the coupling strategy is validated based on a numerical model that is commonly used in the literature and can be considered as a standard benchmark for FSI problems. The second example consists in a simplified wheel rolling on a wetted rigid surface. It includes most of the constraints that need to be considered in order to compute full-scale tires, while requiring limited computational resources and time.

4.3.1 Flexible flag in a flow

4.3.1.1 Problem statement

The first numerical example is a semi-2D problem in which a thin flexible body attached to a rigid square is immersed in an incompressible fluid flow as described in Wall's thesis [181], t. The so-called *flag* example has been widely used to validate FSI procedures as e.g. in [36, 92, 115] and remains a classic in the literature. It exemplifies very well the coupling of an instationary flow with a solid that undergoes large deformation and represents in that an appropriate numerical model in order to assess the quality of the present coupling scheme.

The problem statement of the flag is illustrated in Figure 4.4. The solid body is positioned far enough from the inlet of the fluid domain, which guarantees a uniform inflow whose velocity is prescribed in the x-direction at $\bar{U}_{\text{in}}^{\mathcal{F}} = 51.3$ m/s. The fluid properties and geometry given in Figure 4.4 lead to a Reynolds number[2] of $Re = 330$, at which the flow exhibits a vortex street. The material properties of the solid domain and the geometry of the tail also given in Figure 4.4 were chosen such that the first eigenfrequency of the body approaches the vortex frequency. As a result, the flexible tail oscillates at a frequency close to its natural frequency equal to 3.03 Hz.

The fluid domain is discretized in space into $144 \times 77 \times 1$ (1 cell in the z-direction) cells ensuring a sufficient mesh refinement level in the wake of the rigid square of the body. The time integration scheme presented in section 2.2.5 to solve the discretized Navier-Stokes equations uses a time step size following the FSI time step size such that $\Delta t_n^{\mathcal{F}} = \Delta t_n$ for each time step n. The solid body is discretized with 20 twenty-node reduced-order hexahedral solid elements along the x-direction (1 element in the y-direction) and a hyperelastic neo-Hookean material model as defined in (2.26) is used to deal with the large deformation of the tail. In this example, and only in this one, an implicit solver is employed to solve the solid equations of motion (2.41). Instead of using the procedure given in (2.43), a *Hilbert-Hughes-Taylor* (HHT) time integration scheme [85] is employed, which can be summarized as follows:

$$\mathbf{d}_{n+1}^{\mathcal{S}} = \mathbf{d}_n^{\mathcal{S}} + \Delta t_{n+1}^{\mathcal{S}}\, \dot{\mathbf{d}}_n^{\mathcal{S}} + (0.5 - \beta^{\text{HHT}})\, \Delta t_{n+1}^2\, \ddot{\mathbf{d}}_n^{\mathcal{S}} + \beta^{\text{HHT}}\, \Delta t^2\, \ddot{\mathbf{d}}_{n+1}^{\mathcal{S}}, \tag{4.11a}$$

$$\dot{\mathbf{d}}_{n+1}^{\mathcal{S}} = \dot{\mathbf{d}}_n^{\mathcal{S}} + (1 - \gamma^{\text{HHT}})\, \Delta t_{n+1}^{\mathcal{S}}\, \ddot{\mathbf{d}}_n^{\mathcal{S}} + \gamma^{\text{HHT}}\, \Delta t_{n+1}^{\mathcal{S}}\, \ddot{\mathbf{d}}_{n+1}^{\mathcal{S}}, \tag{4.11b}$$

$$\ddot{\mathbf{d}}_{n+1}^{\mathcal{S}} = (\mathbf{M}^{\mathcal{S}})^{-1} \left(\mathbf{f}_{\text{ext},n}^{\mathcal{S}} - \mathbf{f}_{\text{int},n}^{\mathcal{S}}(\mathbf{d}_{n+1-\alpha^{\text{HHT}}}^{\mathcal{S}})\right), \tag{4.11c}$$

[2] The dynamic behavior of a flow in a particular configuration can be predicted by a dimensionless quantity called the *Reynolds number* Re. It is the ratio between inertial forces to viscous forces within a fluid, which can be formulated as

$$Re = \frac{L\, U^{\mathcal{F}}}{\nu^{\mathcal{F}}}, \tag{4.10}$$

where L is characteristic length, $U^{\mathcal{F}}$ the flow velocity with respect to the obstacle and $\nu^{\mathcal{F}}$ the fluid kinematic viscosity. For a certain range of Reynolds numbers, the flow around a bluff body separates leading to the vortex shedding also known as *von Kármán vortex street*.

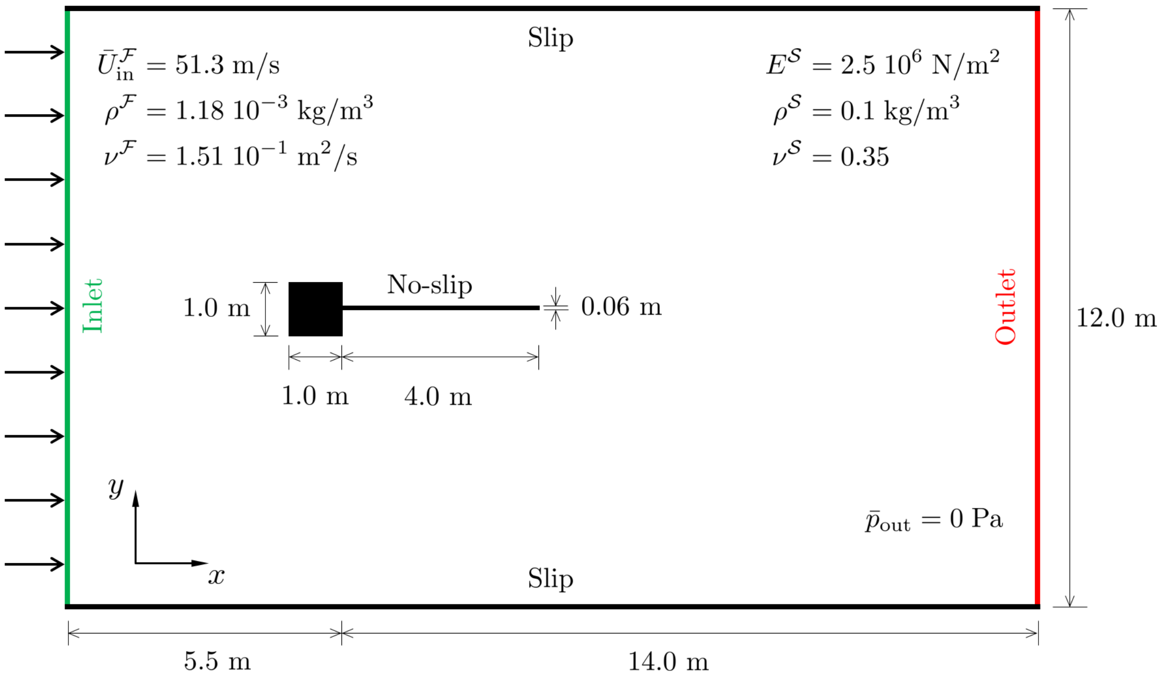
Slip
$\bar{U}_{\text{in}}^{\mathcal{F}} = 51.3$ m/s
$\rho^{\mathcal{F}} = 1.18\ 10^{-3}$ kg/m^3
$\nu^{\mathcal{F}} = 1.51\ 10^{-1}$ m^2/s
$E^{\mathcal{S}} = 2.5\ 10^6$ N/m^2
$\rho^{\mathcal{S}} = 0.1$ kg/m^3
$\nu^{\mathcal{S}} = 0.35$
Inlet
1.0 m
No-slip
0.06 m
1.0 m
4.0 m
Outlet
12.0 m
y
x
$\bar{p}_{\text{out}} = 0$ Pa
Slip
5.5 m
14.0 m

Figure 4.4: Problem statement of a flexible flag in a flow.

with

$$\mathbf{d}^{\mathcal{S}}_{n+1-\alpha^{\text{HHT}}} = (1 - \alpha^{\text{HHT}})\, \mathbf{d}^{\mathcal{S}}_{n} + \alpha^{\text{HHT}}\, \mathbf{d}^{\mathcal{S}}_{n+1}, \tag{4.11d}$$

where the algorithm parameters for the HHT time integrator are chosen as $\alpha^{\text{HHT}} = -0.05$, $\beta^{\text{HHT}} = 0.28$ and $\gamma^{\text{HHT}} = 0.55$ in this particular instance. The nonlinear dynamic equilibrium equations resulting from the implicit operator are then solved by a Newton's method. The HHT procedure was chosen due to the choice of elements available in the implicit version of the solid single-field solver, which are more suitable for computing the deformation of a beam-like solid body such as the tail of the flag.

4.3.1.2 Validation of the FSI computational framework

The results obtained with different combinations of algorithm parameters for the flexible flag are discussed in this section. All test cases are associated to an ID and the corresponding parameters are summarized in Table 4.1.

The typical flow induced vibration of the flexible flag is illustrated in Figure 4.5 for the F0 test case with a constant FSI time step size of $\Delta t = 1.0\,10^{-3}$ s. The fluid velocity magnitude and the pressure fields at several time stations show the development of vortices around the bluff body that trigger the vibration of the flexible body. The vertical tip displacement $d^{\mathcal{S}}_{\text{tip},y}$ of the flag

Flag ID	Δt_{min} [s]	Δt_{max} [s]	ϵ [J]	M_{up} [-]
F0	$1.0\,10^{-3}$ (cst)	-	-	-
F01	$5.0\,10^{-3}$ (cst)	-	-	-
F02	$1.0\,10^{-2}$ (cst)	-	-	-
F1	$1.0\,10^{-3}$	$5.0\,10^{-3}$	$1.0\,10^{-7}$	3
F2	$1.0\,10^{-3}$	$5.0\,10^{-3}$	$5.0\,10^{-7}$	3
F3	$1.0\,10^{-3}$	$5.0\,10^{-3}$	$1.0\,10^{-6}$	3
F4	$1.0\,10^{-3}$	$5.0\,10^{-3}$	$5.0\,10^{-6}$	3
F5	$1.0\,10^{-3}$	$1.0\,10^{-2}$	$1.0\,10^{-6}$	3
F6	$1.0\,10^{-3}$	$1.0\,10^{-2}$	$1.0\,10^{-6}$	5

Table 4.1: Flexible flag: test case ID with corresponding algorithm parameters. Cases named F0* are run with a constant FSI time step size, while an adaptive time stepping procedure is applied to all other cases.

is plotted in Figure 4.6 (black dotted line), where an harmonic response dominated by the first eigenmode of the solid body is apparent after approximately 6 s.

To validate the current coupling approach, the tip displacement is compared to the maximum amplitude obtained for the same setup by various authors in the literature. In [92], Kassiotis et al. also consider finite volumes for discretizing the fluid domain and an implicit couling procedure. It can be seen in Figure 4.6 that the extrema of the amplitude obtained by Kassiotis et al. is very close to the ones achieved with the test case F0. The flag response is also compared to the results published by Matthies & Steindorf [115], Dettmer & Perić [36] and Wall [181], which all use a FE approach to discretize the fluid domain. Although these works use various coupling schemes as well as an ALE formulation of the Navier-Stokes equations, a good agreement is found with the results obtained by the current approach, which is based on an explicit coupling procedure and a fluid fixed-grid combined with an interface reconstruction technique to solve the fluid sub-domain.

4.3.1.3 Coupling time step size optimization

The FSI time step size and its adaptation using the procedure introduced in section 4.2.2 is now considered. In Figure 4.7, the results obtained for the reference test case F0 with $\Delta t = 1.0\,10^{-3}$ s is compared to two other cases, namely F01 with $\Delta t = 5.0\,10^{-3}$ s and F02 with $\Delta t = 1.0\,10^{-2}$ s. The plot of the vertical tip displacement $d^{S}_{\text{tip},y}$ shows that both coupling time step sizes chosen higher than $1.0\,10^{-3}$ s lead to an inaccurate response. In the case F02, the coupled simulation even crashes after $t = 5.51$ s as highlighted by the green cross in Figure 4.7. As expected, the interface energy-based error $||\Delta \boldsymbol{E}^{1}_{\Gamma,n}||_{\text{L}_2}$ given by (4.6) also shows that the smallest error is obtained for the F0 test case. The error increases as soon as the body starts moving around $t \approx 0.7$ s. It then oscillates following the movement of the flag and evolves in an harmonic manner

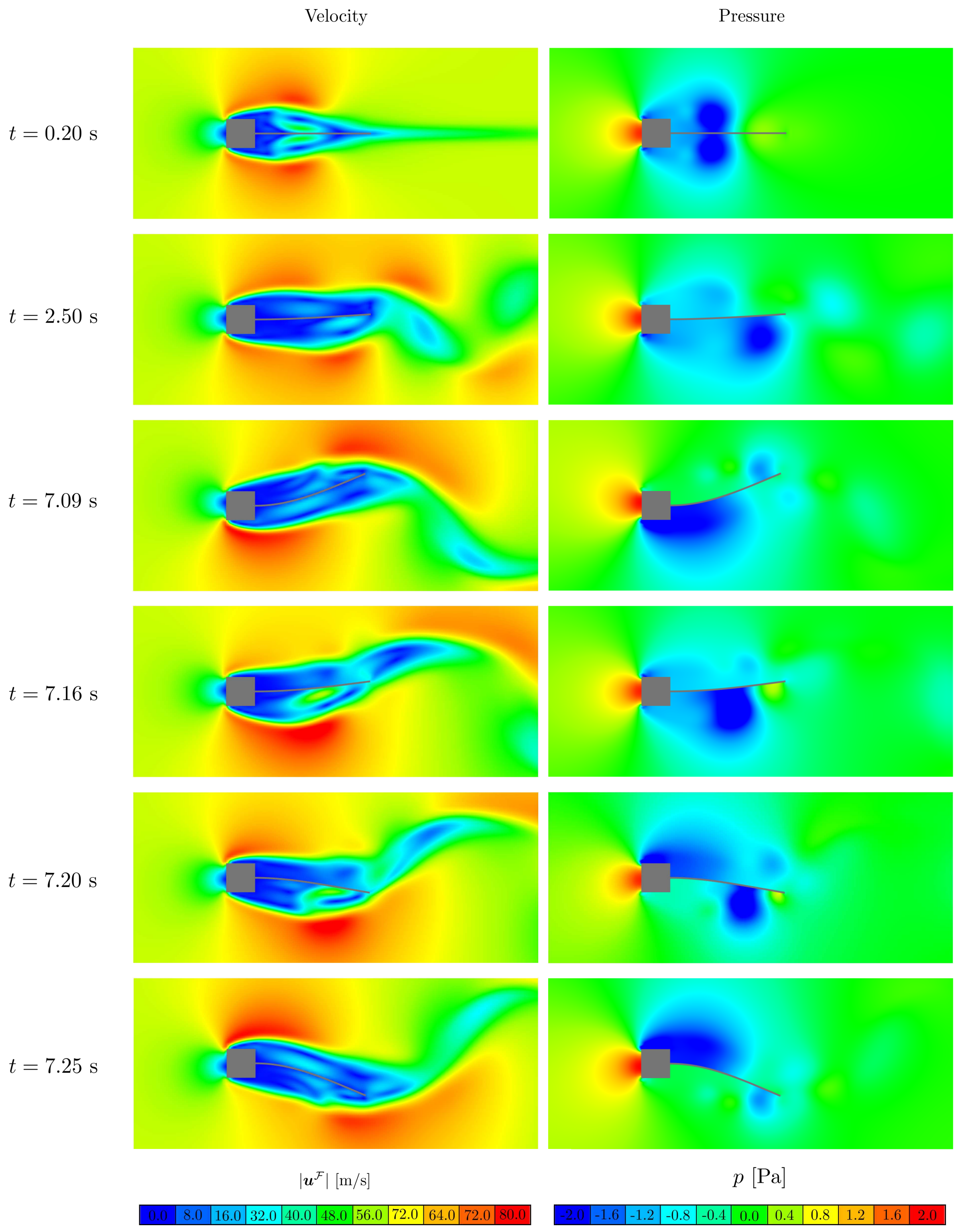

Figure 4.5: Flexible flag: snapshots of fluid velocity magnitude $|\boldsymbol{u}^{\mathcal{F}}|$ and pressure p fields at several time instants (flag test case F0).

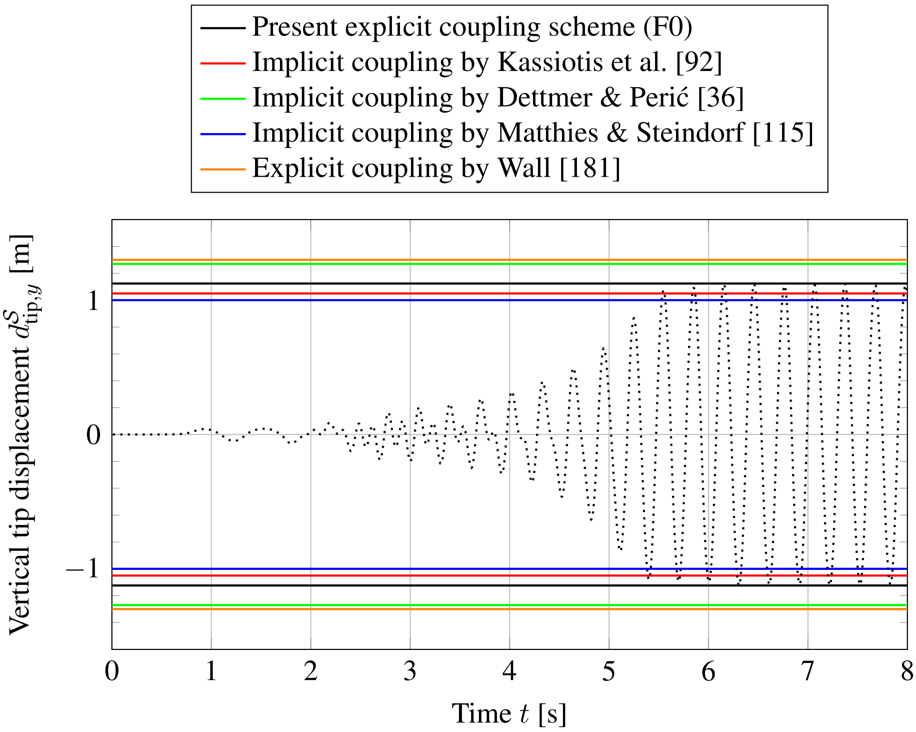

Figure 4.6: Flexible flag: comparison of the maximum amplitude of the vertical tip displacement $d^S_{\text{tip},y}$ from the current explicit coupling approach (flag test case F0) with standard results from the literature using various coupling schemes. The black dotted line shows the evolution over time of the response obtained with the current approach for the example F0.

for F0. Peaks are however observed in the two other cases, which correspond to an increased temporal error in the couple system. The larger the time step size, the higher the error, which complies with the previous observation on the accuracy of the response. The interface energy already shows to be an appropriate criterion to foresee inaccuracies in the coupled solution and to quantify the temporal error resulting from the coupling scheme.

So far, an upper limit on the time step size has been established for the flag example when Δt is considered as a constant. Due to the transient character of the response and the variation of the amplitude of $d^S_{\text{tip},y}$ over the simulation time (the amplitude remains small at the beginning of the simulation and gradually increases before oscillating harmonically), an adaptive coupling time step can be envisaged as advantageous. The simplicity of the current example provides a good opportunity to test the adaptive time stepping procedure and to identify appropriate values for the algorithm parameters given in (4.9). It became quickly clear that some parameters had a larger impact on the change of FSI time step size. The lower and upper limits on the scaling factors κ_{min} and κ_{max} do not influence much the computed time step size in (4.7) and were chosen in order to allow for a maximum variation of the optimized scaling factor $\kappa_n^{\circledcirc}$ of 25%, that is $\kappa_{\text{min}} = 0.75$ and $\kappa_{\text{max}} = 1.25$. The safety factor has been chosen as $\kappa_{\text{s}} = 0.95$ in order to keep the error away

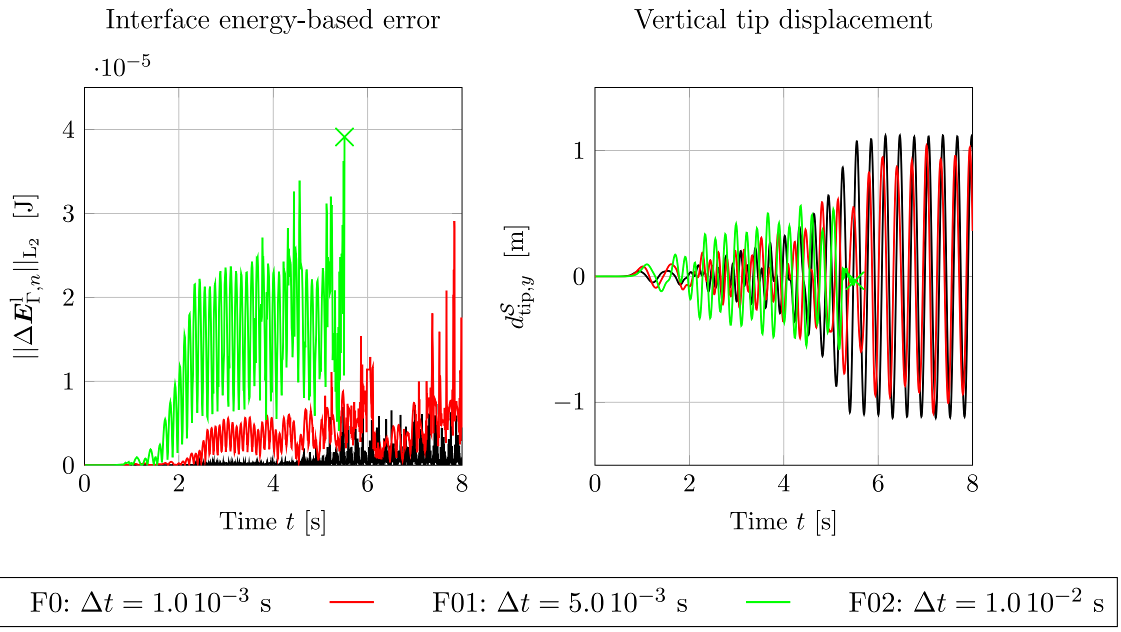

Figure 4.7: Flexible flag: interface energy-based error $||\Delta \boldsymbol{E}^{1}_{\Gamma,n}||_{\mathrm{L}_2}$ and vertical tip displacement $d^{S}_{\mathrm{tip},y}$ for three different FSI time step sizes. The cross indicates the point at which the simulation crashes.

from the tolerance limit and the smoothing parameter $\mathrm{M}_{\mathrm{down}} = 2$ so that a decrease is quickly reflected in the time step size. Larger values for $\mathrm{M}_{\mathrm{down}}$ lead to an early crash of the simulation due to the fact that the accuracy requirements dictated by the interface energy-based error are not met quickly enough.

The other parameters proved to influence far more the evaluation of the optimum time step size. In particular, the tolerance factor ϵ has the largest effect as shown in Figure 4.8 for the test cases F0-F4[3]. An increase of ϵ allows for a larger time step size, which in turn leads to an increased error. At the beginning of the simulation, as the flag remains still, the time step size increases rapidly regardless of ϵ. The increase is only limited by the parameters κ_{max} and M_{up} before the time step reaches Δt_{max}. A small tolerance, as for the F1 test case, leads to a rapid decrease of the time step size, which then sticks to the minimum value of Δt_{min}. For a larger ϵ, the time step size decreases more gradually and can still reach larger values until the end of the simulation, which in turn leads to a higher time efficiency of the solution procedure. The best compromise between time-to-solution and accuracy is achieved here for the example F3 as shown in Table 4.2, where a decrease of the total simulation time of approximately 48% is observed for a difference of only 0.3% in the maximum amplitude compared to the case F0. The limit is reached here for the test case F4 and a tolerance of $\epsilon = 5.0\,10^{-6}$ J, for which the time step size never leaves its maximum

[3] A backward moving average with 100 intervals has been applied to the interface energy-based error plots presented in the Figures 4.8-4.9 to be able to compare more easily the results from the different flag examples. As already apparent in Figure 4.7, this particular quantity has the tendency to show large oscillations so that the curves corresponding to raw data are usually hard to compare.

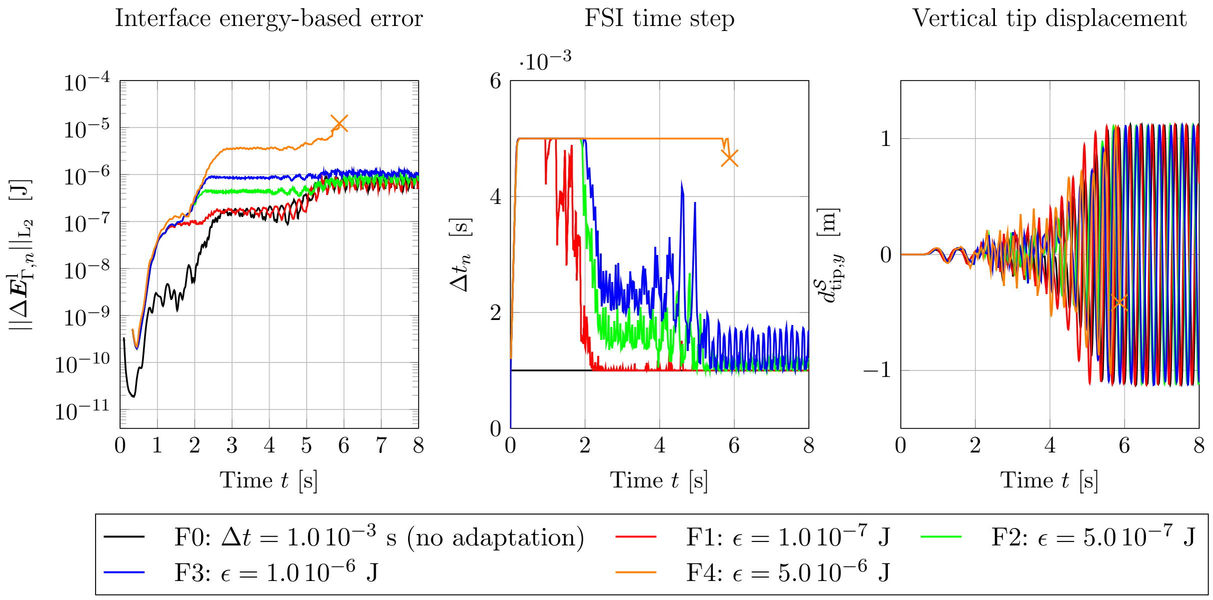

Figure 4.8: Flexible flag: interface energy-based error $||\Delta \boldsymbol{E}^{\mathrm{I}}_{\Gamma,n}||_{\mathrm{L}_2}$ (moving average with 100 intervals), FSI time step size Δt_n and vertical tip displacement $d^S_{\mathrm{tip},y}$ for different values of the tolerance parameter ϵ in the adaptive time stepping procedure. The cross indicates the point at which the simulation crashes.

Δ vs. F0 [%]	F1	F2	F3	F5	F6
$\max\{\,|d^S_{\mathrm{tip},y}|\,\}$	0.2	0.3	0.3	1.6	0.4
Simulation time	- 23	- 40	- 48	- 54	- 52

Table 4.2: Flexible flag: difference between the test cases F0 versus F1-F3, F5 and F6 in terms of the maximum amplitude of the vertical tip displacement and the overall simulation time.

allowed value until the error suddenly increases and leads to a simulation crash after $t \approx 5.87$ s, i.e. at a time where the vertical tip displacement starts reaching its maximum amplitude.

In Figure 4.9, the case F3 is kept as a reference for evaluating other algorithm parameters, namely the maximum time step size Δt_{max} and the smoothing upper limit M_{up}. In a further attempt to improve the time efficiency of the simulation, Δt_{max} has been set to $1.0\,10^{-2}$ s in the case F5. The resulting time step size increase is only taken advantage of at the beginning of the simulation when the flag remains still and falls down at the same level as for F3 as soon as the body starts moving. The drop of the time step size around $t = 2$ s is actually faster for F5 than for F3, which can be explained by the higher level of error that the F5 case suffers from as the time step size decreases. In order to delay that drop of Δt, the case F6 was constructed with $\mathrm{M}_{\mathrm{up}} = 5$ in order to lower the level of error while approaching $t = 2$ s, which also resulted in a more gradual time step size increase at the simulation start. If the level of accuracy is acceptable for a lower value of M_{up} as for F5, the corresponding solution procedure delivers the best time efficiency. If not,

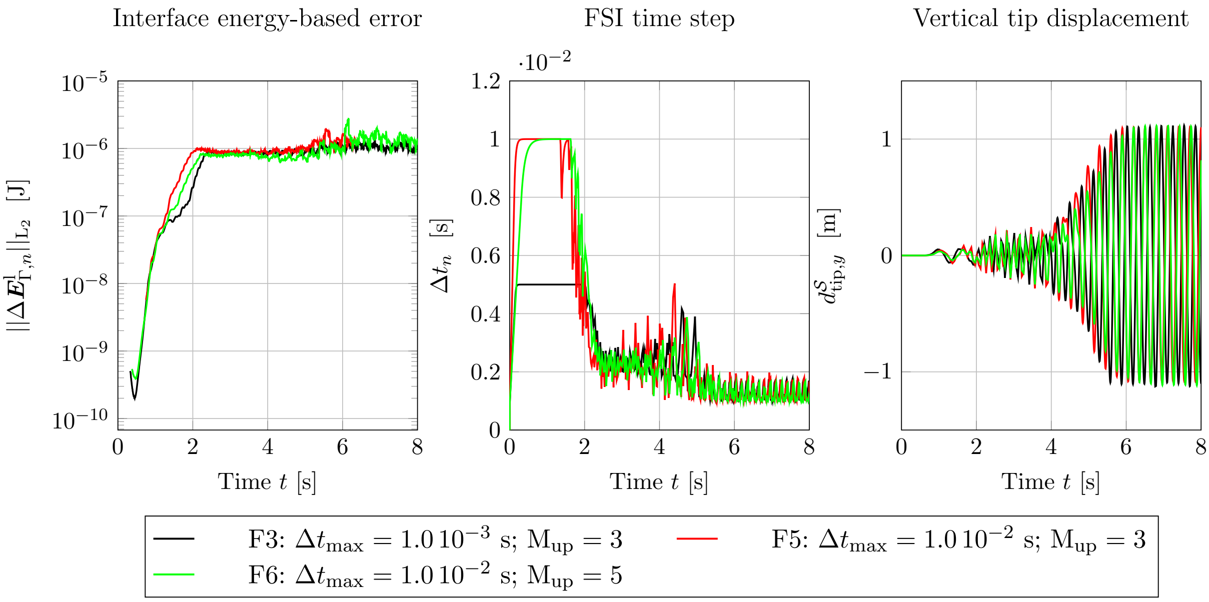

Figure 4.9: Flexible flag: interface energy-based error $||\Delta \boldsymbol{E}^{\mathrm{I}}_{\Gamma,n}||_{\mathrm{L}_2}$ (moving average with 100 intervals), FSI time step size Δt_n and vertical tip displacement $d^S_{\mathrm{tip},y}$ for different values of the algorithm parameters $\Delta t_{\max}$ and $\mathrm{M_{up}}$ optimizing the adaptive time stepping procedure ($\epsilon = 1.0\,10^{-6}$).

the upper limit for the smoothing parameter can be increased, which in turn limits the error while producing an acceptable overall simulation time as shown in Table 4.2.

The flag example has been used to evaluate the influence of the algorithm parameters. It appeared that the tolerance ϵ has the largest effect on the time efficiency, while $\Delta t_{\max}$ and $\mathrm{M_{up}}$ can be used for a finer tuning. It is important to note that the flag test case is particular as the body motion comprises multiple phases for which the FSI time step size requirements are very different. As a result, a decrease of up to 54% of the overall simulation time could be obtained, while the first results from Figure 4.7 indicated that a constant FSI time step size larger than $\Delta t = 1.0\,10^{-3}$ s was not applicable. The situation is different in the case of the Grosch wheel example in which the body motion includes only one regular phase as discussed in the next section.

4.3.2 Grosch wheel rolling on a thick water layer

4.3.2.1 Problem statement

The Grosch wheel example is a small scale wheel composed of a unique rubber compound rolling on a wetted rigid surface. Contrary to full scale tire models, no reinforcement nor cavity are included in the model so that the wheel is not inflated by air. This simplified tire model is much easier to run and can be simulated one order of magnitude faster than a tire hydroplaning simulation. Since the Grosch wheel hydroplaning simulation includes the main computational requirements as for a real tire, it represents a suitable example to test the coupling procedure and package used in this work.

The hydroplaning performance of the Grosch wheel is computed here for a straight free-rolling test, reproducing the test conditions of the glass plate experiment described in section 1.2.2. As can be seen in Figure 4.10 (a), the tire is mounted on a rigid rim and rotates around an axis in the y-direction, which is only free to move along the z-direction. The road is translated in the x-direction with a prescribed constant velocity of $\bar{U}_x^{\text{road}} = 48.3$ km/h. As a result, the wheel remains in a bounded spatial domain and does not actually travel down the road, which avoids having a fluid domain of large size. The wheel rotates with an angular velocity of

$$\bar{\Omega}_y^{\text{GW}} = \frac{\bar{U}_x^{\text{road}}}{R_{\text{dyn}}^{\text{GW}}}, \tag{4.12}$$

where $R_{\text{dyn}}^{\text{GW}}$ is the wheel's *dynamic radius* that is equal to $R^{\text{GW}} = 0.04$ m at the beginning of the simulation and is subjected to change as the tire spins and deforms. Furthermore, a vertical load $\bar{F}_z^{\text{GW}} = -500$ N is prescribed at the center of the rim and the contact between the wheel and the road is modeled by an isotropic Coulomb friction model as described in section 2.1.5.2 with a coefficient of friction $\mu_{\text{c}} = 0.9$. The material properties as well as the dimensions of the wheel are given in Figure 4.10 (a), where W^{GW} and H^{GW} denote the width of the wheel (in the y-direction) and the height of the tire side wall respectively. The fluid domain is split into two parts as illustrated in Figure 4.10 (b): a fluid layer of 3 mm covering the road and a region filled with void on top describe a free-surface flow. A fluid velocity $\bar{U}_{\text{in}}^{\mathcal{F}} = \bar{U}_x^{\text{road}}$ is prescribed at the inlet in both the fluid and void regions. As initial condition, the VOF variable Q is set to 1 in the fluid layer and 0 in the void region. The fluid properties and the boundary conditions used for the fluid model are also given in Figure 4.10 (b).

The Grosch wheel solid model is discretized with ~29000 3D 8-node reduced-order hexahedral elements as described in section 2.1.5.1. The hyperelastic neo-Hookean material model is used for describing the behavior of the rubber compound, whereas the contact definition between the road and the tire as well as between the rim and the tire are dealt with the penalty method described in section 2.1.5.2. The explicit time integration scheme presented in section 2.1.4 is employed to solve the solid system (2.41). The fluid domain is discretized in space into ~95000 cells which are the smallest at the level of road and along the wetted surface of the tire, where the highest mesh density is needed. The cell size gradually increases as one moves away from the road and from the tire. The fluid method presented in section 2.2 is used to solve the fluid system with a fluid time step size following the FSI time step such that $\Delta t_n^{\mathcal{F}} = \Delta t_n$ for each time step n. In this case, both single-fields have different requirements in terms of time step size so that the solid sub-domain is sub-cycled to reach an acceptable level of numerical stability as described in section 2.1.4. If the FSI time step remains constant during the simulation, a sub-cycling ratio as defined by (3.9) with $R_{n+1}^{\mathcal{S}} = 46$ for all n is adopted. In the event of an adaptive FSI time step,

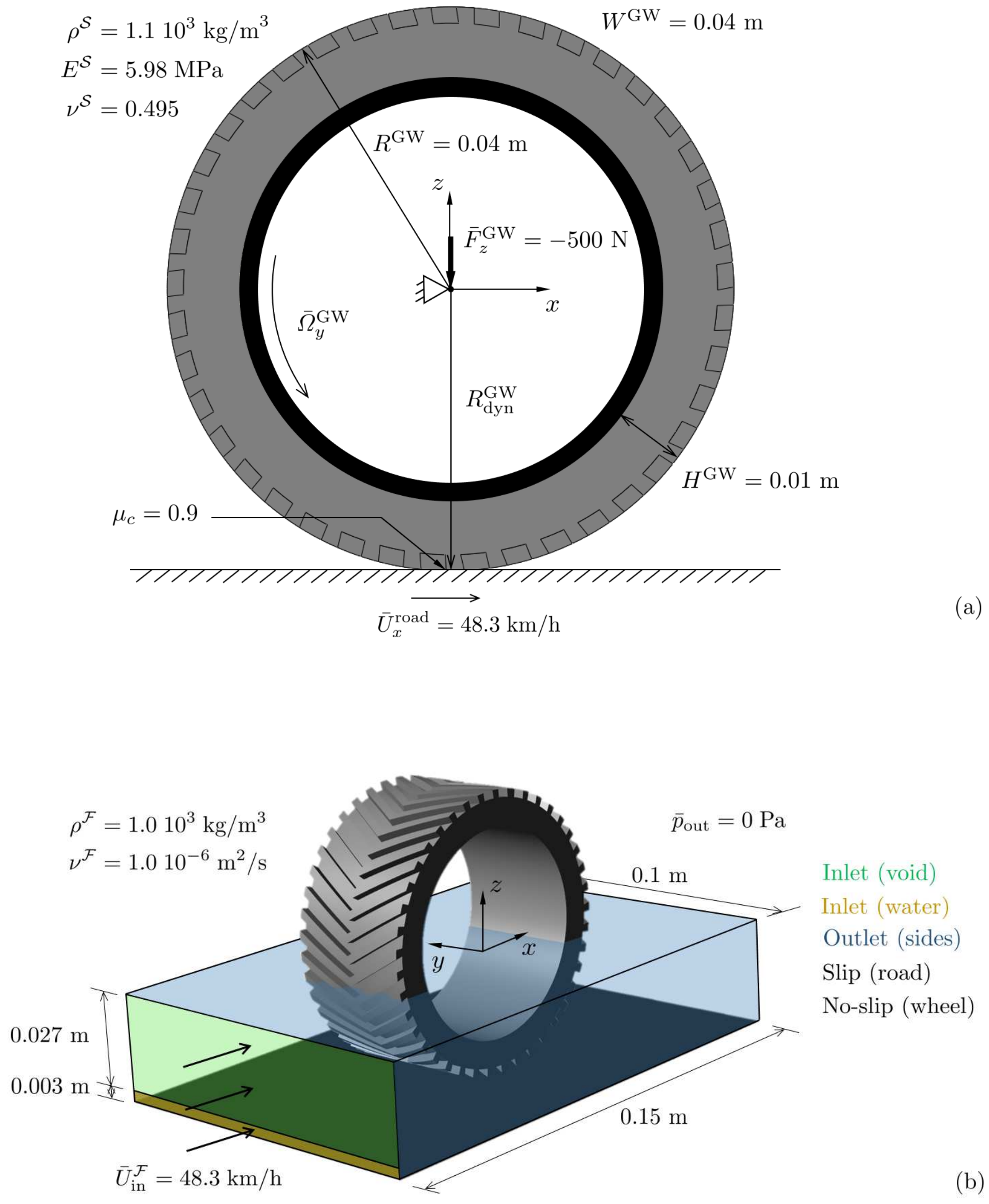
$\rho^{\mathcal{S}} = 1.1\ 10^3\ \mathrm{kg/m^3}$
$E^{\mathcal{S}} = 5.98\ \mathrm{MPa}$
$\nu^{\mathcal{S}} = 0.495$
$W^{\mathrm{GW}} = 0.04\ \mathrm{m}$
$R^{\mathrm{GW}} = 0.04\ \mathrm{m}$
z
$\bar{F}_z^{\mathrm{GW}} = -500\ \mathrm{N}$
x
$\bar{\Omega}_y^{\mathrm{GW}}$
$R_{\mathrm{dyn}}^{\mathrm{GW}}$
$H^{\mathrm{GW}} = 0.01\ \mathrm{m}$
$\mu_c = 0.9$
$\bar{U}_x^{\mathrm{road}} = 48.3\ \mathrm{km/h}$
(a)
$\rho^{\mathcal{F}} = 1.0\ 10^3\ \mathrm{kg/m^3}$
$\nu^{\mathcal{F}} = 1.0\ 10^{-6}\ \mathrm{m^2/s}$
$\bar{p}_{\mathrm{out}} = 0\ \mathrm{Pa}$
0.1 m
Inlet (void)
Inlet (water)
Outlet (sides)
Slip (road)
No-slip (wheel)
z
y
x
0.027 m
0.003 m
0.15 m
$\bar{U}_{\mathrm{in}}^{\mathcal{F}} = 48.3\ \mathrm{km/h}$
(b)

Figure 4.10: Problem statement of the Grosch wheel benchmark: solid (a) and fluid (b) model description.

Grosch wheel ID	Δt_{min} [s]	Δt_{max} [s]	ϵ [J]	M_{up} [-]	Order of predictor (3.17)
GW01	$3.0\,10^{-5}$ (cst)	-	-	-	0
GW02	$1.5\,10^{-5}$ (cst)	-	-	-	0
GW03	$7.5\,10^{-6}$ (cst)	-	-	-	0
GW04	$5.0\,10^{-6}$ (cst)	-	-	-	0
GW02P1	$1.5\,10^{-5}$ (cst)	-	-	-	1
GW02P2	$1.5\,10^{-5}$ (cst)	-	-	-	2
GW1P1	$1.0\,10^{-6}$	$1.0\,10^{-2}$	$1.0\,10^{-2}$	10	1
GW1P2	$1.0\,10^{-6}$	$1.0\,10^{-2}$	$1.0\,10^{-2}$	10	2
GW2P2	$1.5\,10^{-5}$	$1.0\,10^{-2}$	$1.0\,10^{-2}$	5	2
GW3P2	$1.5\,10^{-5}$	$1.0\,10^{-2}$	$5.0\,10^{-3}$	5	2

Table 4.3: Grosch wheel: test case ID with corresponding algorithm parameters. Cases named GW0* are run with a constant FSI time step size, while an adaptive time stepping procedure is applied to all other cases. The P indicates that a predictor (1st or 2nd order) on the solid displacement has been applied.

the ratio at the beginning of the simulation is also given by $R_1^S = 46$ ($n = 0$) and is then free to change as Δt_n and Δt_n^S fluctuate over time.

In the end, the Grosch wheel example represents a complex free-surface flow problem due to the complicated solid geometry, the contact definition between the tire and the road as well as the relatively large fluid and angular velocity compared to the size of the wheel. Not only does this example allow to test the new coupling approach with a free-surface flow, it also gathers some of the main challenging features of a tire hydroplaning simulation and consists, in that, in a very meaningful test case.

Multiple instances of the Grosch wheel example are considered in this section depending on the algorithm parameters and the order of the solid predictor (3.17) employed. Each example is associated to an ID, for which the parameters are summarized in Table 4.3.

4.3.2.2 Typical results for tires wet performance

Prior to the actual FSI computation, a so-called *dry rolling* run is performed for the Grosch wheel. In this first step, no water is present in the computational model and the wheel rotates at an angular velocity $\bar{\Omega}_y^{\text{GW}}$ with a vertical load $\bar{F}_z^{\text{GW}}$ applied to the center of the rim. Once the tire has reached a stabilized deformed state the solution is saved and then used as an initial step for the FSI computation. Only the results obtained during this FSI simulation, which consists of one revolution of the wheel and lasts $T \approx 0.0183$ s, are discussed in the following.

Some of the typical results obtained for evaluating the wet performance of a tire are presented here for the Grosch wheel. A constant FSI time step size $\Delta t = 7.5\,10^{-6}$ s corresponding to the example GW03 has been used, which showed to deliver stabilized results as discussed in the following. Figure 4.11 (a) shows the fluid relative velocity $\boldsymbol{u}^{\mathcal{F}}_{\text{rel}}$ at the end of the simulation $(t = T)$, which is defined as the difference between the fluid velocity vector $\boldsymbol{u}^{\mathcal{F}}$ and the road velocity vector given by $\begin{pmatrix}\bar{U}^{\text{road}}_x & 0 & 0\end{pmatrix}^T$. By subtracting the road velocity component, this relative velocity quantity allows to visualize the “true” fluid velocity field, which would be obtained if the wheel instead of the road was moving in the x-direction. It can be observed that an important water splash occurs in front of the tire, which ejects the water away from the contact patch region. Moreover, the water trapped within the grooves is swept away as the wheel spins and eventually rejected in the wake of the tire. The pressure field p is shown in Figure 4.11 (b) highlighting that, at $t = T$, most of the fluid in the wake of the tire has been ejected away from the wheel’s trajectory for which no fluid data is available. A high pressure region is observed in front of the tire, which is responsible for ejecting the water away form the contact patch region. Figure 4.11 (c) exhibits the free-surface of the flow at the end of the simulation by showing the distribution of the VOF variable Q within the fluid sub-domain, which highlights once again the strength of the water splash. It is also clearly visible here that the mesh resolution quickly reduces once one moves away from the road, leading to an uneven free-surface on the top of the fluid sub-domain. While having a fine mesh in the entire fluid sub-domain would lead to much better results visualization, the high number of cells that would result from that would be impractical for such simulations. Especially considering the fact that there is absolutely no interest from a hydroplaning performance point of view of having precise results in that region of the flow where water particles only exit the fluid sub-domain. In Figure 4.11 (d), the tire footprint on the road is compared at the beginning and at the end of the simulation. It is clearly visible that, in this configuration, the tire does not lose contact with the road and does not enter in hydroplaning. Due to the size of the grooves that can easily penetrate the water layer and that the tire deformation is almost negligible for a non-inflated tire, the shape of the footprint is almost not altered by the presence of water. These observations are corroborated in the following, while looking at the lift force resulting from the action of the water on the rotating tire.

The lift force that acts on the tire is one of the few results available that allows to quantitatively estimate the hydroplaning performance of a tire. Therefore it also represents a meaningful quantity for testing various configurations of the coupling scheme. In Figure 4.12, the lift force F_z is plotted for different FSI time step sizes, which have been kept constant during the whole duration of the simulation. While $\Delta t > 7.5\,10^{-6}$ s does not lead to accurate results, a time step size of $\Delta t = 7.5\,10^{-6}$ s corresponding to the case GW03 is small enough for capturing the so-called *pitch*, that is each peak in the plot of the lift force that coincides with the entry of a lug of the tire tread into the water. Due to a relatively large noise signal in the lift force response, the results proposed in Figure 4.12 are averaged using a backward moving average with 30 intervals, which showed to deliver the clearest results while keeping the pitch apparent. The Grosch wheel case GW03 is therefore chosen as a reference for the rest of this section. Note that with a resulting

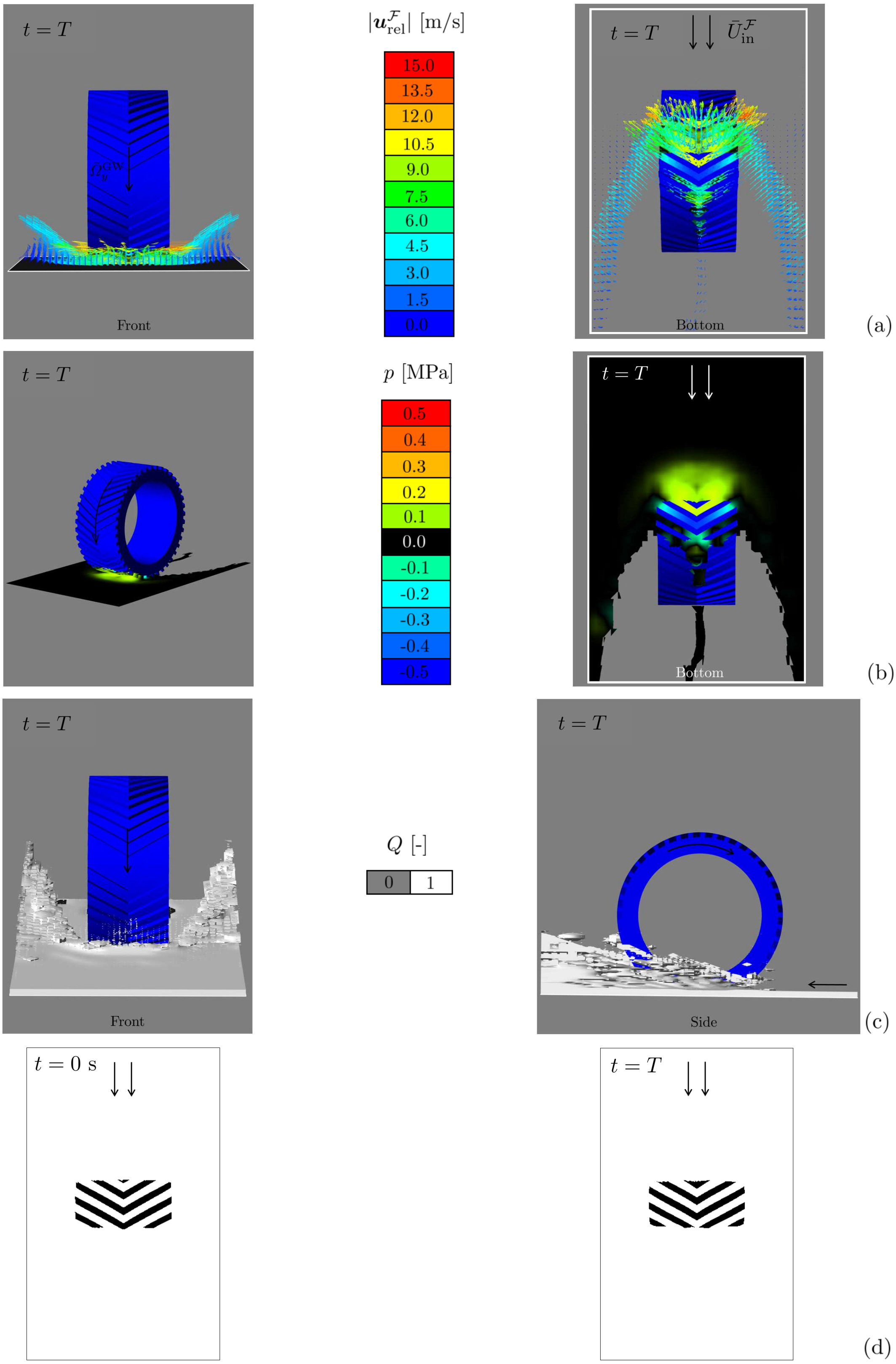

Figure 4.11: Grosch wheel: snapshots of fluid relative velocity $\boldsymbol{u}_{\mathrm{rel}}^{\mathcal{F}}$ (the color of the arrows indicates the fluid relative velocity magnitude $|\boldsymbol{u}_{\mathrm{rel}}^{\mathcal{F}}|$) (a), pressure p (b), VOF variable Q (c) at the end of the simulation ($t = T$) as well as the tire footprint on the road (d) at the start and end of the simulation (Grosch wheel GW03).

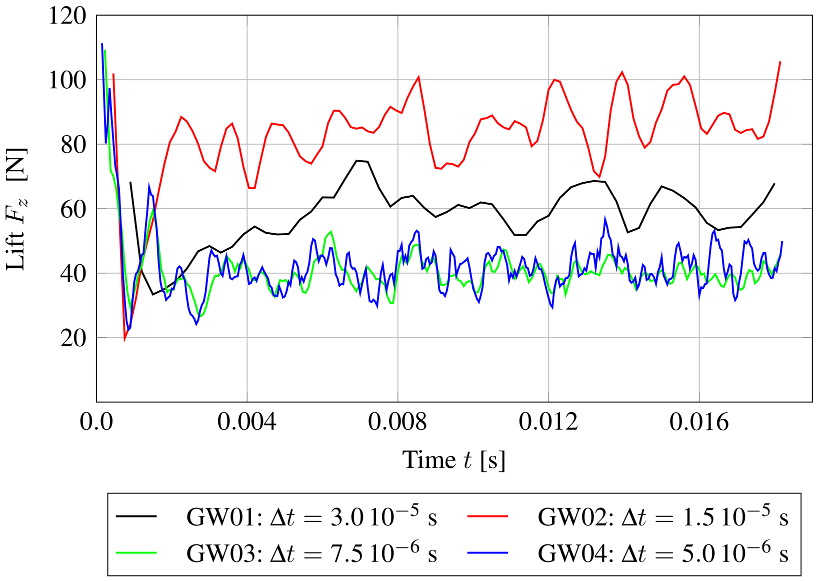

Figure 4.12: Grosch wheel: lift force F_z (moving average with 30 intervals) for different FSI time step sizes.

lift force oscillating around 40 N, the wheel is far from entering in hydroplaning as the vertical force $\bar{F}_z^{\mathrm{GW}}$ pushing the wheel against the road is more than one order of magnitude higher. This also explains the small difference between the shape of the footprint at the beginning and the end of the simulation in Figure 4.11 (d). As will be discussed in the next chapter, the situation is way different in the case of a real tire model, where the tire is inflated with air and the size of the grooves is much smaller compared to the size of the wheel.

4.3.2.3 Effect of solid predictor

The solid predictor introduced in section 3.3.2 that replaces the displacement field at the interface computed by the solid solver by expression (3.17) is considered in this section. A 1st order ($\alpha_0 = 1$ and $\alpha_1 = 0$ in (3.17)) and a 2nd order ($\alpha_0 = 1$ and $\alpha_1 = 1/2$ in (3.17)) time-accurate predictor are used and applied to the Grosch wheel example GW02, resulting in the cases GW02P1 and GW02P2 respectively. The lift force F_z and the corresponding temporal error given by the interface energy-based error $||\Delta \boldsymbol{E}^{\mathrm{I}}_{\Gamma,n}||_{\mathrm{L}_2}$ given by (4.6) are plotted in Figure 4.13. Note that, in the case of a free-surface flow, the total number of nodes on the FSI interface $\mathrm{N}^{\Gamma}_{\mathrm{nodes}}$ in expression (4.6) is replaced by the number of *active* nodes on the interface, that is the number of nodes shared by the solid sub-domain (tire) and the fluid sub-domain where the VOF variable $Q = 1$ (water). In doing so, the interface energy-based error is scaled appropriately with the number of nodes taking part to the coupling data exchange.

As already mentioned, the example GW02 produced an inaccurate response with an FSI time step size of $\Delta t = 1.5\,10^{-5}$ s that was too large to converge towards the lift force obtained for

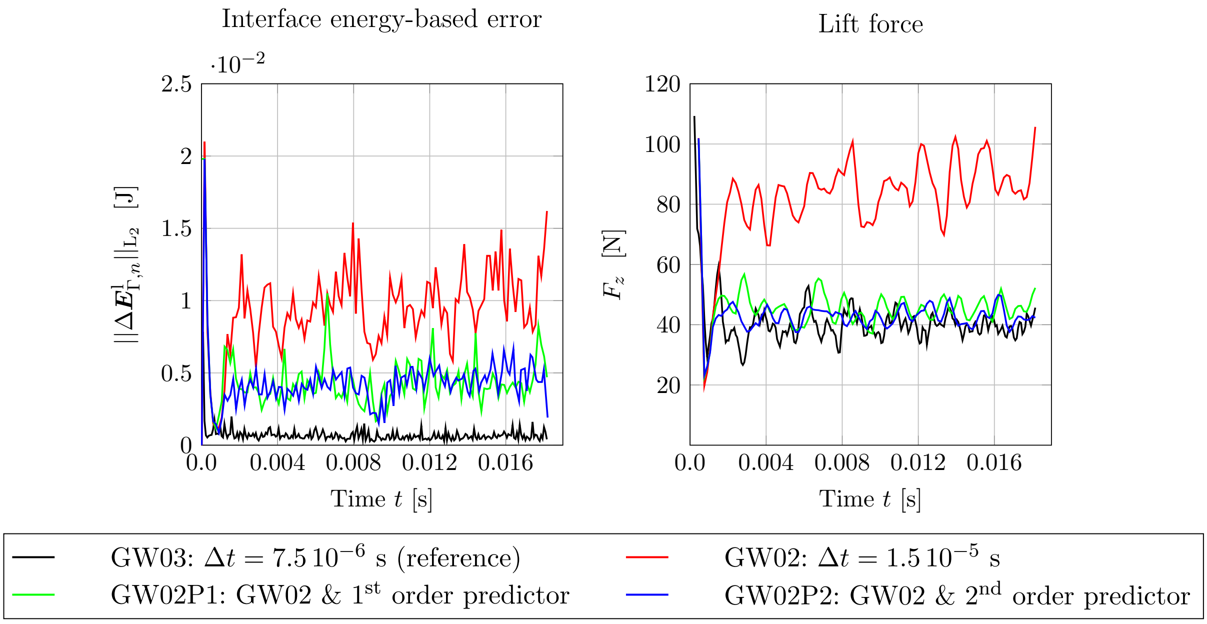

Figure 4.13: Grosch wheel: interface energy-based error $||\Delta \boldsymbol{E}^{\mathrm{I}}_{\Gamma,n}||_{\mathrm{L}_2}$ and lift force F_z (moving average with 30 intervals) showing the beneficial impact of the 1st and 2nd order solid predictors applied on the test case GW02 compared to the reference test case GW03.

the reference case GW03. The application of a 1st and 2nd order solid predictor for GW02P1 and GW02P2 shows to bring the lift force at the same level as for GW03, while using an FSI time step size twice as large. Looking now at the produced error in Figure 4.13, both cases with a predictor improve the level of accuracy of the response but still cannot reach the level of accuracy delivered by the GW03 reference example. At this point, no clear difference can be identified between both predictors in terms of produced error or resulting lift force.

These examples show that the use of a solid predictor in the coupling scheme tends to improve the level of accuracy of the coupled procedure, which corroborates the observations made by Piperno and Farhat in [147] as discussed in section 3.3.2. From an time efficiency point of view, the use of the predictor allows to use a larger FSI time step size in order to compute the coupled problem while keeping a sufficient level of accuracy. Both GW02P1 and GW02P2 jobs reduced the overall simulation time by ~48 % compared to the GW03 reference test case that ran with an FSI time step size twice as large. In order to keep a good compromise between time-to-solution and accuracy, the application of a solid predictor either of 1st and 2nd order is therefore highly recommended in this case.

4.3.2.4 Coupling time step size optimization

In an attempt to optimize further the time efficiency of the solution procedure, the previously discussed test cases are compared to examples run with the adaptive coupling time stepping procedure introduced in section 4.2.2. The test case GW1P1 was built with the algorithm parameters given in Table 4.3 and a 1st order solid predictor. Moreover, the following values for the less significant algorithm parameters as discussed in section 4.3.1.3 are considered: $\kappa_{\mathrm{s}} = 0.95$, $\mathrm{M}_{\mathrm{down}} = 2$, while $\kappa_{\mathrm{min}} = 0.5$ and $\kappa_{\mathrm{max}} = 2$ allowing for a variation of the optimized scaling factor $\kappa_n^{\circledcirc}$ by a factor 2. Note that these parameters will be kept as such for all the following cases discussed in this section.

As shown in Figure 4.14, the FSI time step size of the GW1P1 case converges rapidly to a value oscillating around $1.5\,10^{-5}$ s, which corresponds to the constant FSI time step size delivering an efficient and accurate response with a 1st order solid predictor in the example GW02P1. It shows that the dynamic adaptive procedure employed for optimizing the FSI time step size, when used with proper algorithm parameters, can automatically converge to an appropriate Δt_n. This can be quite useful especially knowing how challenging the choice of the FSI time step size can be for problems with a transient response. Also apparent in Figure 4.14, the introduction of an adaptive time stepping process leads to a degradation of the solution as shown by the evolution of the interface energy-based error for GW1P1 compared to the analogous Grosch wheel test case using a constant FSI time step size and the same predictor order, namely GW02P1. The variation of the FSI time step size introduces a slight instability, which results in this temporal error increase. The resulting lift force is accordingly impacted by that effect and converges at a value further away from the reference solution delivered by GW03, while comparing the green and orange curves with the black curve. As a matter of fact, the overall simulation time obtained with the adaptive procedure in GW1P1 is less than 1 % faster compared to the case GW02P1 with a fixed FSI time step. Similar observations can be made for the cases with a 2nd order predictor, that is by comparing the example GW1P2 with GW02P2. In this case however, the resulting lift force converges closer to the reference solution so that a 2nd order predictor is kept in the following in order to find appropriate algorithm parameters and improve the overall simulation time.

The results obtained for the case GW1P2 implied that the minimum value of the FSI time step size Δt_{min} was too conservative as the FSI time step size increases rapidly to $1.5\,10^{-5}$ s, value around which it oscillates during the rest of the simulation. This led to the construction of the test case GW2P2 for which Δt_{min} is set directly to $1.5\,10^{-5}$ s and with a lower value of M_{up} guaranteeing a smoother evolution of the FSI time step size over the course of the simulation. As can be seen in Figure 4.15, the ramp up of the FSI time step size at the beginning of the simulation is much smaller, which accelerates the simulation. This is at the cost of accuracy as can be seen by the higher level of error and the peak at the beginning of the curve of the interface energy-based error. As a result, the case GW2P2 runs $\sim$ 13 % faster than GW1P2 but with a lift force that converges further apart from the reference solution. In order to reduce the level of error,

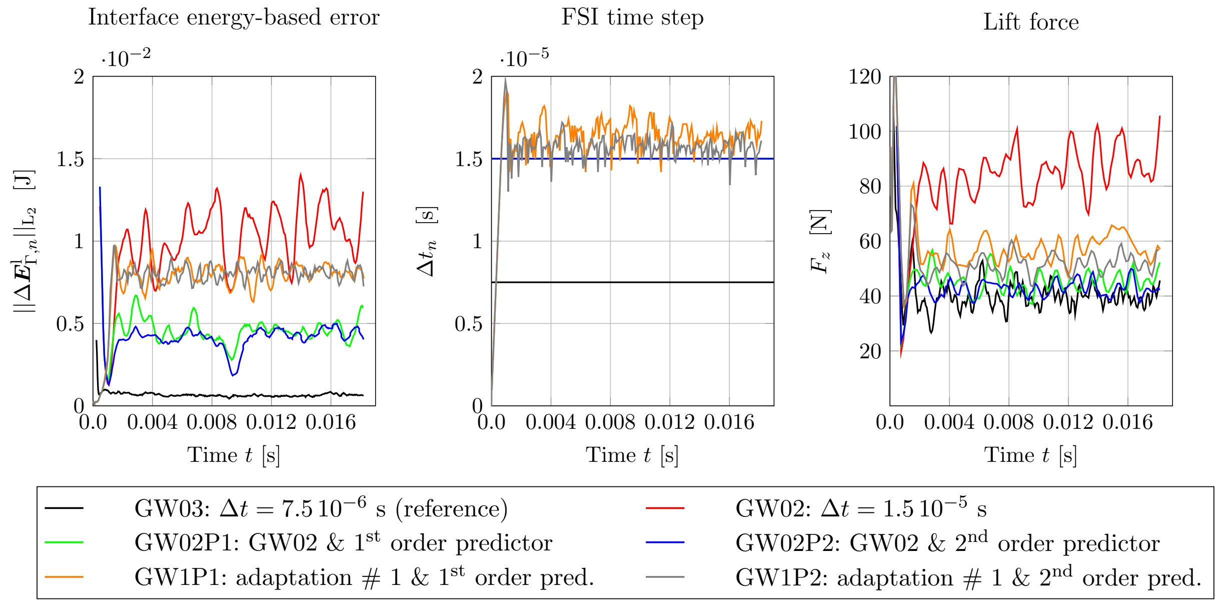

Figure 4.14: Grosch wheel: interface energy-based error $||\Delta \boldsymbol{E}^{\mathrm{I}}_{\Gamma,n}||_{\mathrm{L}_2}$ (moving average with 30 intervals), FSI time step size Δt_n and lift force F_z (moving average with 30 intervals) for two examples considered with an adaptive FSI time stepping procedure and solid predictors (GW1P1 and GW1P2) compared to test cases with a constant FSI time step with and without solid predictors.

the test case GW3P2 using half the value of the tolerance ϵ has been built. While the resulting lift force is more accurate, the simulation with an adaptive time stepping procedure is not more efficient than the example with a constant time step size fixed at $1.5\,10^{-5}$ s (GW02P2).

To summarize, it can be stated that the only way of improving the overall simulation time of the Grosch wheel example is at the cost of a non negligible accuracy loss. Any attempt to reduce the temporal error with an adaptive time stepping procedure leads to a simulation time that is of the same order as when a constant FSI time step size is used. Due to the loss of numerical stability introduced by the FSI time step adaptation, the use of a constant FSI time step is preferred in this case.

4.3.3 Concluding remarks

The coupling shell introduced in section 4.1 for coupling the single-field solvers described in chapter 2 have been used successfully to compute FSI problems as sophisticated as the Grosch wheel example featuring a complex free-surface flow problem and contact occurrences.

The dynamic adaptation procedure of the FSI time step introduced in section 4.2.2 showed to be very time efficient in the case of the flag example where the total simulation time could be reduced by more than 50 % while keeping a totally satisfying level of accuracy. For the

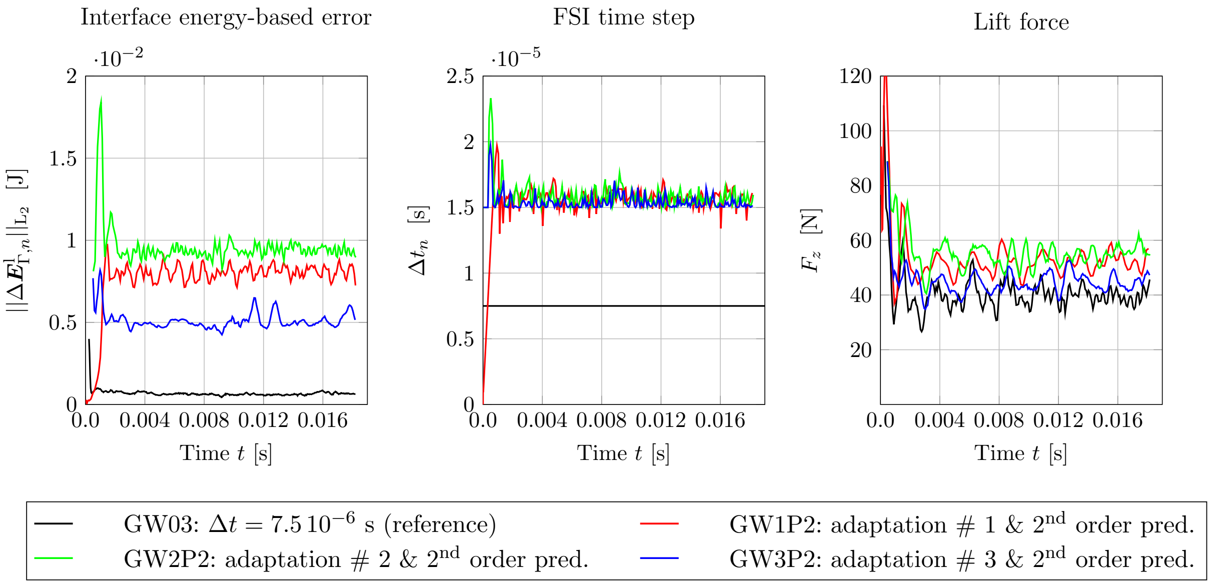

Figure 4.15: Grosch wheel: interface energy-based error $||\Delta \boldsymbol{E}^{\mathrm{I}}_{\Gamma,n}||_{\mathrm{L}_2}$ (moving average with 30 intervals), FSI time step size Δt_n and lift force F_z (moving average with 30 intervals) obtained for different adaptive FSI time stepping procedures and a 2^{nd} order solid predictor compared to the reference test case GW03.

Grosch wheel, the adaptation of the FSI time step size did not prove to speed up the computation without affecting the accuracy of the solution. As a result, a constant FSI time step size showed to perform better while leading to an overall simulation time similar to the adaptive time step counterpart. The reason for that can be explained as follows. Since the prescribed vertical load is high compared to the resulting lift force $\left(F_z/\bar{F}_z^{\mathrm{GW}} \approx 10\%\right)$ and since the tire of the Grosch wheel is not inflated, it does not deform much under the action of water. Contrary to the flag example, the computation of the solid sub-domain is therefore not much affected by the presence of water in the system and the problem can be considered as weakly-coupled, where only the fluid sub-domain is impacted by the solid sub-domain. This means that if an appropriate FSI time step size is found that meets the stability requirements for solving the fluid sub-system, the temporal error in the coupled system will not fluctuate much leaving in turn only very little room for optimizing further the FSI time step size. However, the adaptive time stepping procedure proved to be quite useful by allowing to automatically find the optimum time step size associated to the Grosch wheel problem as shown in section 4.3.2.4 for a chosen level of accuracy.

Furthermore, the application of a solid predictor to the Grosch wheel example led to a substantial reduction of the interface energy-based error, which in turn allowed using a larger coupling time step size. Note that the trivial fluid corrector (3.24b) has been used here. Other instances of (3.24) were also tested and delivered an unstable pressure distribution in the tire footprint, which led to a rapid increase of numerical instabilities in the coupled system and a early crash of the computation. The application of other forms of the fluid corrector was therefore deemed unpractical for the current application.

In the end, an explicit coupling procedure has been implemented which allows to successfully compute various FSI problems in an time efficient manner, while controlling the level of temporal error in the coupled system. In the next chapter, the coupling strategy will be used for computing the tire hydroplaning problem, which is achieved without any further modifications to the proposed coupling shell.

5 Numerical results for full-scale tire model

In the previous chapter, the computational package including the two single-field solvers presented in chapter 2 and the newly proposed coupling shell has been successfully applied to FSI computations ranging from an academic example to a simplified tire model rolling on a wet road surface. The overall solution procedure is now used to compute the hydroplaning performance of a full-scale tire.

The computational model of a real tire is first presented. In doing so, the complexity of a full-scale tire construction is highlighted and the simplifications necessary to obtain the computational model starting from the physical model are exposed. The preliminary steps carried out to build the 3D solid model used in the subsequent FSI computation is presented and the modeling of the many constituents that form a real tire is briefly commented. Numerical results obtained for a sport utility vehicle (SUV) tire construction are then presented. The areas for improvement for the computation of tire hydroplaning applications based on the preexisting coupling scheme are emphasized, while the limitations of the loosely-coupled approach for solving such FSI examples are discussed.

5.1 Tire computational model

Tires are complex and composite solid systems that are made of multiple constituent materials including rubber compounds, steel and synthetic reinforcements. Figure 5.1 shows the main components, which fulfill various roles such as:

- Tread: the tire part in direct contact with the road and made of rubber. Its design impacts the traction and the handling in dry and wet conditions as well as the noise and the rolling resistance of a tire.
- Plies: usually made of nylon, the plies hold the shape of the tire due to the inflation pressure.
- Bead: steel wire that carries the ply loads and locks the tire onto the rim.

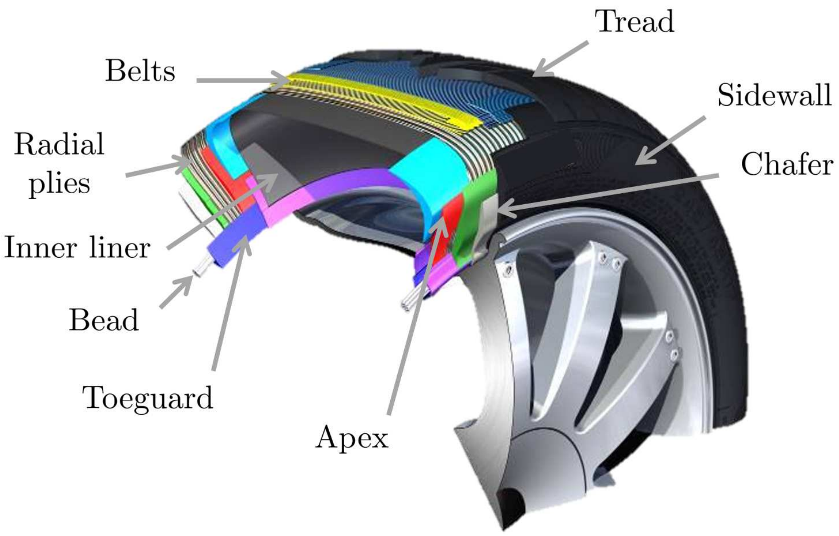

Figure 5.1: Principal constituents of a tire construction. A tire is a composite solid system that includes multiple rubber compounds as well as synthetic and metallic materials.

- Belts: also made of steel, belts stiffen the footprint area and guarantee lateral and torsional stability as well as puncture resistance.
- Inner line: another rubber part responsible for ensuring air impermeability.
- Sidewall: made of rubber, the outer flank of the tire protects from outside agents and influences the damping characteristics of the tire, which can have an impact on handling and noise.
- Apex: rubber parts responsible for stiffening the sidewall.
- Toeguard: protects from tire damages during mounting/dismounting, also made of rubber.
- Chafer: rubber part guaranteeing fitment between the tire and the rim.

The material properties of all constituents vary much, in particular the reinforcement components such as the belts, plies and beads are much stiffer than the surrounding rubber compounds. The broad variation in the stiffness of the constitutive materials add to the difficulties arising from the nearly-incompressibility of rubber compounds and the multiple contact definitions discussed in section 2.1.5 that need to be considered while computing a rolling tire. As a detailed description of a full-scale tire model generation is out of scope of this thesis, a few aspects of the 3D computational model are shortly tackled in the following.

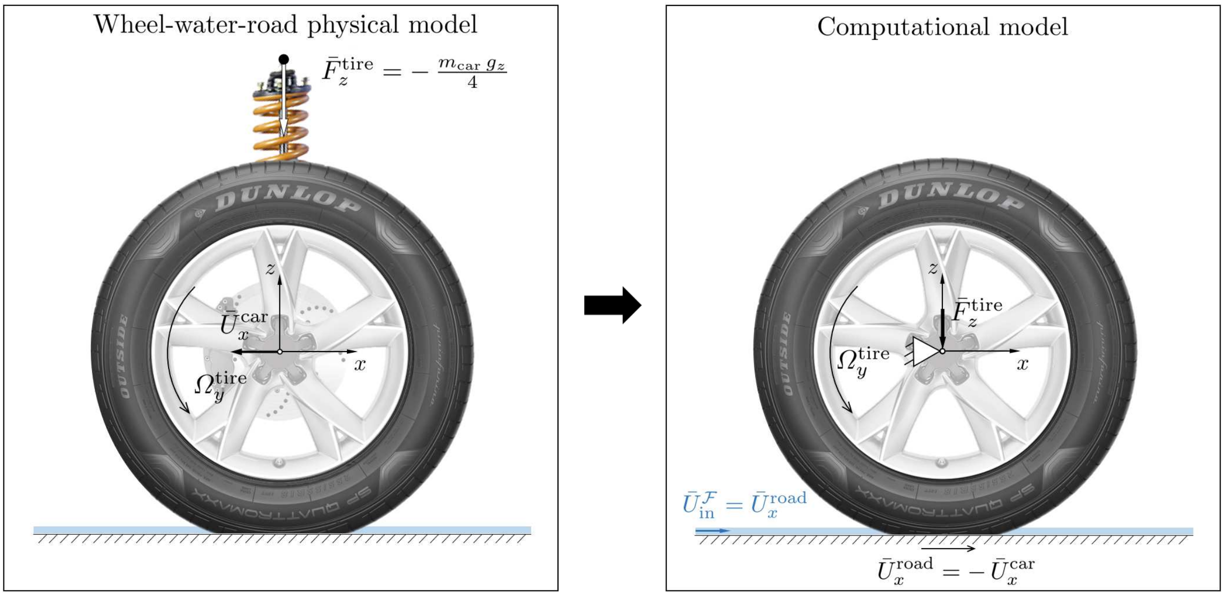
Wheel-water-road physical model
$\bar{F}_z^{\text{tire}} = -\frac{m_{\text{car}}\, g_z}{4}$
DUNLOP
z
$\bar{U}_x^{\text{car}}$
x
Ω_y^{tire}
OUTSIDE
SP QUATTROMAXX
Computational model
DUNLOP
z
$\bar{F}_z^{\text{tire}}$
Ω_y^{tire}
x
OUTSIDE
SP QUATTROMAXX
$\bar{U}_{\text{in}}^{\mathcal{F}} = \bar{U}_x^{\text{road}}$
$\bar{U}_x^{\text{road}} = -\bar{U}_x^{\text{car}}$

Figure 5.2: Simplifications from the physical to the computational wheel model, where the brakes and the suspensions are removed. Instead of translating the rotating wheel in the x-direction with the velocity magnitude $\bar{U}_x^{\text{car}}$ as in the physical model, a prescribed velocity of the same amplitude and opposite direction is applied to the road and the water in the computational model.

Physical to computational tire model

To be able to compute the deformation of a rolling tire, the corresponding computational model is simplified compared to its physical counterpart, as schematically represented in Figure 5.2. In the approximated solid sub-domain, the car suspension as well as the braking systems attached to the rim are removed, which results in a fixed hub only free to move in the z-direction. Moreover, the weight of the moving vehicle applied on each wheel is replaced by a static vertical load $\bar{F}_z^{tire} = -\frac{m_{\text{car}}\, g_z}{4}$, where g_z is the vertical component of the gravitational acceleration vector and m_{car} the mass of the considered vehicle. In order to keep the rotating wheel in a bounded fluid sub-domain, the car translation motion is split into a water-road translation with the same uniform prescribed velocity such that $\bar{U}_{\text{in}}^{\mathcal{F}} = \bar{U}_x^{\text{road}} = -\bar{U}_x^{\text{car}}$. Similarly to the Grosch wheel example described in section 4.3.2.1, the fluid sub-domain consists of a water layer and another region filled with void, which describes a free-surface flow. A more detailed description of the coupled problem statement for the tire hydroplaning application will be given in section 5.2.1.

3D tire model generation

Only considering the solid sub-domain at this point, the numerical analysis of a rolling tire based on the previously introduced computational model involves first constructing the 3D tire model. It is not a straightforward task due to the multiple contact occurrences and the large number of

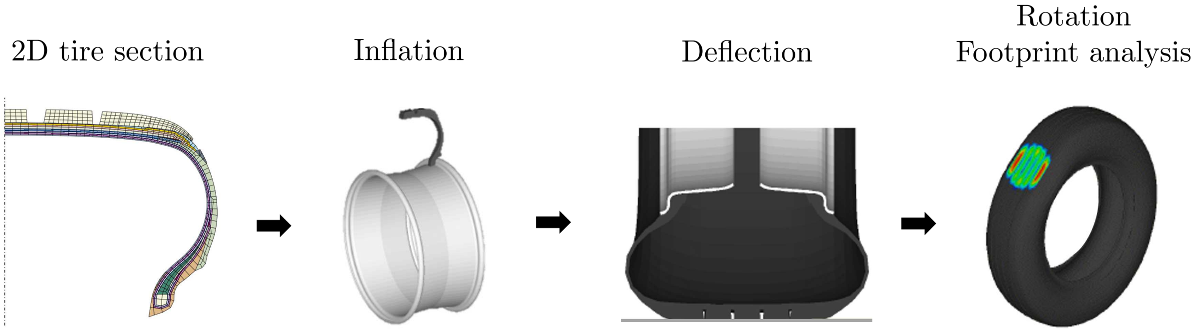

Figure 5.3: Stages of a 3D tire model generation: a 2D tire section is mounted on a rim (rigid body) and an inflation pressure is applied on the inside of the section; the resulting deformed tire wedge is revolved around the rim to build the 3D tire model; external loads are applied as the tire comes into contact with the road; the tire is rotated on the road to perform a footprint analysis.

constituents involved. The latter have very different material properties and shapes, which can be very complex to model. This includes the reinforcement components that need to be included in the rubber matrix composed of the tire tread and the rest of the rubber components.

As shown in Figure 5.3, to be able to perform a tire analysis, the tire must first be inflated and mounted on the rim, then deflected on the road and finally rotated. During inflation, a uniformly distributed *inflation pressure* p_{infl} is applied on the inner line of the 2D tire section so that the tire comes into contact with the rigid rim. The rim mounting results thus in enforcing the contact of the tire with the rim, which involves a frictional sliding of the chafer over the rim. Once the 2D tire section has reached a stable deformed state, the resulting wedge is revolved by 360° around the rim and mirrored along the longitudinal direction to obtain the 3D tire model[1]. After inflation, the tire is loaded with the vertical force $\bar{F}_z^{tire}$ and the contact of the tire with the road is enforced. At this point and during the rest of the analysis, the inflation pressure p_{infl} remains engaged. Finally, the tire is rotated on the road around the y-direction and a camber angle (rotation around the x-direction) can be added if required. In the end, the tire is rotated and free to move along the z-direction to perform the footprint analysis. These steps represent the preliminary stage prior to the actual FSI computation, for which the deformed tire resulting from the dry rolling computation is used as an initial condition.

Tire reinforcement modeling

As previously mentioned, the tire computational model is subdivided into multiple parts, each one having different material properties and composed of different finite elements. It goes from

[1] Note that in the event of an non-axisymmetric tire tread design, the latter is glued onto the tire carcass after it has been revolved around the rim.

2D membrane elements with anisotropic material properties for the tire belts and plies to 3D solid elements for the tread. While the tire tread and all rubber compounds are modeled with solid isoparametric hexahedral 8-node elements as described in section 2.1.5.1, the belts and plies are modeled with *rebar layers* in membrane elements, which are embedded in the rubber solid elements.

Materials such as nylon cords used for the tire plies are embedded in a rubber matrix and do not possess compressible stiffness. In order to model these types of components, membrane elements including rebar layers are embedded in the 3D solid elements employed for the rubber constituents. These rebar layers have their stiffness defined in an uniaxial direction, which depends on the cords orientation within the tire and confers the anisotropic behavior to these constituents. An elastic material model is used to characterize the behavior of these reinforcements. As for the bead, 1D beam elements providing axial reinforcement are employed.

The resulting solid model of a tire is therefore very complex with a large number of components and elaborate shapes. A considerable number of degrees of freedom is thus required to satisfy the accuracy requirements and simulate the rolling tire, as presented in the next section.

5.2 Numerical results for SUV tire *Dunlop SP Quattromaxx* 225/55 R19

In order to illustrate the tire hydroplaning problem, a SUV tire is employed, namely the *Dunlop SP Quattromaxx* 225/50 R19. Due to its relatively large size, this tire is expected to move a large amount of water and is therefore well indicated for demonstrating the wet performance of tires using the proposed solver package. The developments proposed in the previous chapter in order to improve the preexisting coupling scheme are applied to the present industrial example. The effect of the solid predictor as well as the adaptation of the FSI time step size for optimizing the accuracy and time efficiency of the solution procedure are tested for the this tire model.

5.2.1 Problem statement

Similarly to the Grosch wheel example presented in section 4.3.2.1, the computational setup consists in a tire rolling on a rigid surface covered by a water film. In this case, the tire accelerates in a straight line with a constant torque $\bar{T}_y^{\text{tire}} = -500$ Nm as shown in Figure 5.4 (a) and is subjected to a vertical load of $\bar{F}_z^{\text{tire}} = -5000$ N, representing the quarter of the weight of a typical SUV. The inflation pressure is set at $p_{\text{infl}} = 2.2$ bar. The road as well as the water are translated along the moving direction with a constant velocity of $\bar{U}_{\text{in}}^{\mathcal{F}} = \bar{U}_x^{\text{road}} = 80$ km/h. The contact between the road and the tire tread is described by the isotropic Coulomb friction model presented in section 2.1.5.2. The fluid sub-domain is split into a water and a void part, describing

Full-scale tire ID	Δt_{min} [s]	Δt_{max} [s]	ϵ [J]	M$_{\text{up}}$ [-]	Order of predictor (3.17)
T01	$2.0\,10^{-5}$ (cst)	-	-	-	0
T01P1	$2.0\,10^{-5}$ (cst)	-	-	-	1
T01P2	$2.0\,10^{-5}$ (cst)	-	-	-	2
T01P1/2	$2.0\,10^{-5}$ (cst)	-	-	-	1/2
T1	$2.0\,10^{-5}$	$2.0\,10^{-4}$	$5.0\,10^{-3}$	10	0
T2	$2.0\,10^{-5}$	$2.0\,10^{-4}$	$20.0\,10^{-3}$	10	0
T3	$2.0\,10^{-5}$	$2.0\,10^{-4}$	$25.0\,10^{-3}$	10	0

Table 5.1: Full-scale tire: test case ID with corresponding algorithm parameters. Cases named T0* are run with a constant FSI time step size, while an adaptive time stepping procedure is applied to all other cases. The P indicates that a predictor on the solid displacement has been applied.

a free-surface flow with an initial water layer thickness of 5 mm. The boundary conditions and other dimensions for both the solid and fluid sub-domains are given in Figure 5.4.

The solid tire model is discretized altogether with $\sim$ 485000 elements. All rubber compounds are modeled with the hyperelastic neo-Hookean law and the contact occurrences are dealt with the penalty method, both described in section 2.1.5. The explicit time integration from section 2.1.4 is used to solve the solid sub-system. As for the fluid sub-domain, it is discretized into $\sim$ 2.5 10^6 fluid cells, in which the fluid sub-system is computed by the method presented in section 2.2. In the same way as for the Grosch wheel example, the fluid time step size follows the FSI coupling time step $(\Delta t_n^{\mathcal{F}} = \Delta t_n)$. Due to the different numerical stability requirements from both sub-domains, the solid system is sub-cycled such that the sub-cycling ratio (3.9) is given by $R_{n+1}^{\mathcal{S}} = 100$ in the event of a constant FSI time step or in the beginning of the simulation $(n = 0)$ if an adaptive time stepping procedure is chosen.

All tire hydroplaning test cases are summarized in Table 5.1, which gathers the algorithm parameters considered in this chapter.

5.2.2 Typical results for tires wet performance

Once the preliminary dry rolling simulation described in section 5.1 is over, the coupled simulation starts and runs for $T \approx 0.085$ s, which corresponds to a 300° tire rotation. Due to the size and complexity of the problem, the overall simulation time needed for computing the coupled system is substantial. Therefore only a 300° rotation is computed, which is however sufficient for reaching a stabilized state and converged results, as shown in the following.

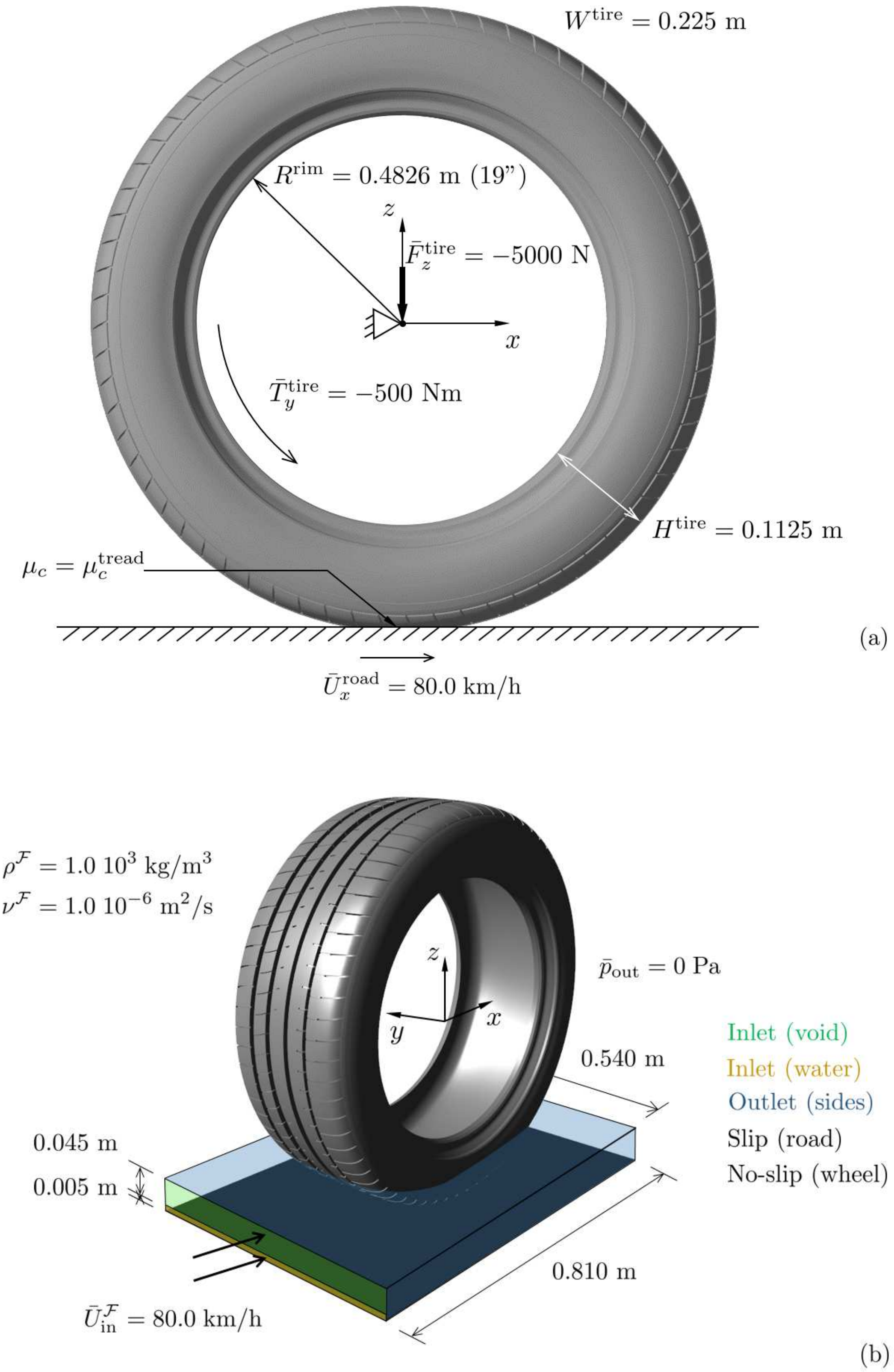
$W^{\text{tire}} = 0.225$ m
$R^{\text{rim}} = 0.4826$ m (19")
z
$\bar{F}_z^{\text{tire}} = -5000$ N
x
$\bar{T}_y^{\text{tire}} = -500$ Nm
$H^{\text{tire}} = 0.1125$ m
$\mu_c = \mu_c^{\text{tread}}$
(a)
$\bar{U}_x^{\text{road}} = 80.0$ km/h
$\rho^{\mathcal{F}} = 1.0\ 10^3$ kg/m^3
$\nu^{\mathcal{F}} = 1.0\ 10^{-6}$ m^2/s
z
y
x
$\bar{p}_{\text{out}} = 0$ Pa
0.540 m
Inlet (void)
Inlet (water)
Outlet (sides)
Slip (road)
No-slip (wheel)
0.045 m
0.005 m
0.810 m
$\bar{U}_{\text{in}}^{\mathcal{F}} = 80.0$ km/h
(b)

Figure 5.4: Problem statement of the tire *Dunlop SP Quattromaxx* 225/50 R19: solid (a) and fluid (b) model description.

A constant FSI time step size of $\Delta t = 2.0\,10^{-5}$ s is used in the test case T01, which is considered as the reference example in the remaining of this chapter. The distribution of the VOF variable Q at different time instants is illustrated in Figure 5.5 (a). It shows that a water splash builds up over the duration of the simulation and reaches a large size at $t = T$, even though the water film thickness is relatively small compared to the tire dimensions. The fluid relative velocity $\boldsymbol{u}_{\text{rel}}^{\mathcal{F}}$ already introduced in section 4.3.2.2 also highlights in Figure 5.5 (b) the strength of the splash, which ejects the water away from the tire trajectory and in particular in front of the tire where a water wedge is created. The impact of the tread with the water leads to a rise in fluid pressure in the contact patch region, which causes the tire footprint to lift off the road surface. This fluid pressure rise can be observed in the front of the contact patch in Figure 5.5 (c). A good tire design aims at evacuating as much water as possible away from the front wedge area in order to keep that fluid pressure as low as possible and generate a dry contact in the back of the footprint as mentioned in section 1.2.1. Figure 5.5 (c) also highlights that there is almost no water left in the wake of the tire at $t = T$, while the tire remains however in contact with the road. This is apparent in Figure 5.6, where the evolution of the contact pressure p_{c}, the fluid pressure p and the VOF variable Q in the tire footprint is shown at different time instants. The snapshots of the contact pressure distribution in the footprint show that the remaining contact area between the tire tread and the road rapidly decreases as the tire rotates but does not disappear at $t = T$, i.e. no hydroplaning occurs. The distribution of the fluid pressure is relatively uniform in the front of the contact patch region and does not vary much over the course of the simulation. Integrating this fluid pressure over the wetted surface area delivers the hydrodynamic force that must be counteracted by the tire vertical stress in order to avoid hydroplaning, which proves to be the case in the current example.

Typical quantitative results for assessing the hydroplaning performance of a tire in an accelerated test consist in the lift force F_z, the *remaining contact area* RCA and the *slip rate* SR, both defined in the following. They are all plotted in Figure 5.7 for the reference test case T01. The lift force reaches a converged value around 4200 N after about $t \approx$ 0.05 s, which is indeed below the vertical load $\bar{F}_z^{\text{tire}}$ applied to the tire so that there is no hydroplaning. This is in accordance with the above mentioned observations made in terms of pressure distribution in the tire footprint. With the RCA, the remaining contact area between the tire and the road is compared to that at the beginning of the simulation. It tends also to a constant value of RCA $\approx$ 25 % after around $t \approx$ 0.05 s, which means that the tire loses about 3/4 of its contact area at 80 km/h due to the presence of the 5 mm water layer. One of the most important quantities for assessing the wet performance of a tire in an accelerated wet traction test is the *slip rate* SR, which compares the angular velocity of the tire in wet and dry conditions such as

$$\text{SR} = \frac{\mathit{\Omega}_y^{\text{wet}} - \mathit{\Omega}_y^{\text{dry}}}{\mathit{\Omega}_y^{\text{dry}}}. \tag{5.1}$$

The SR is an important quantity as it can be easily compared to experimental data (the deep straight aquaplaning test described in section 1.2.2) and is very sensitive to the tire design and

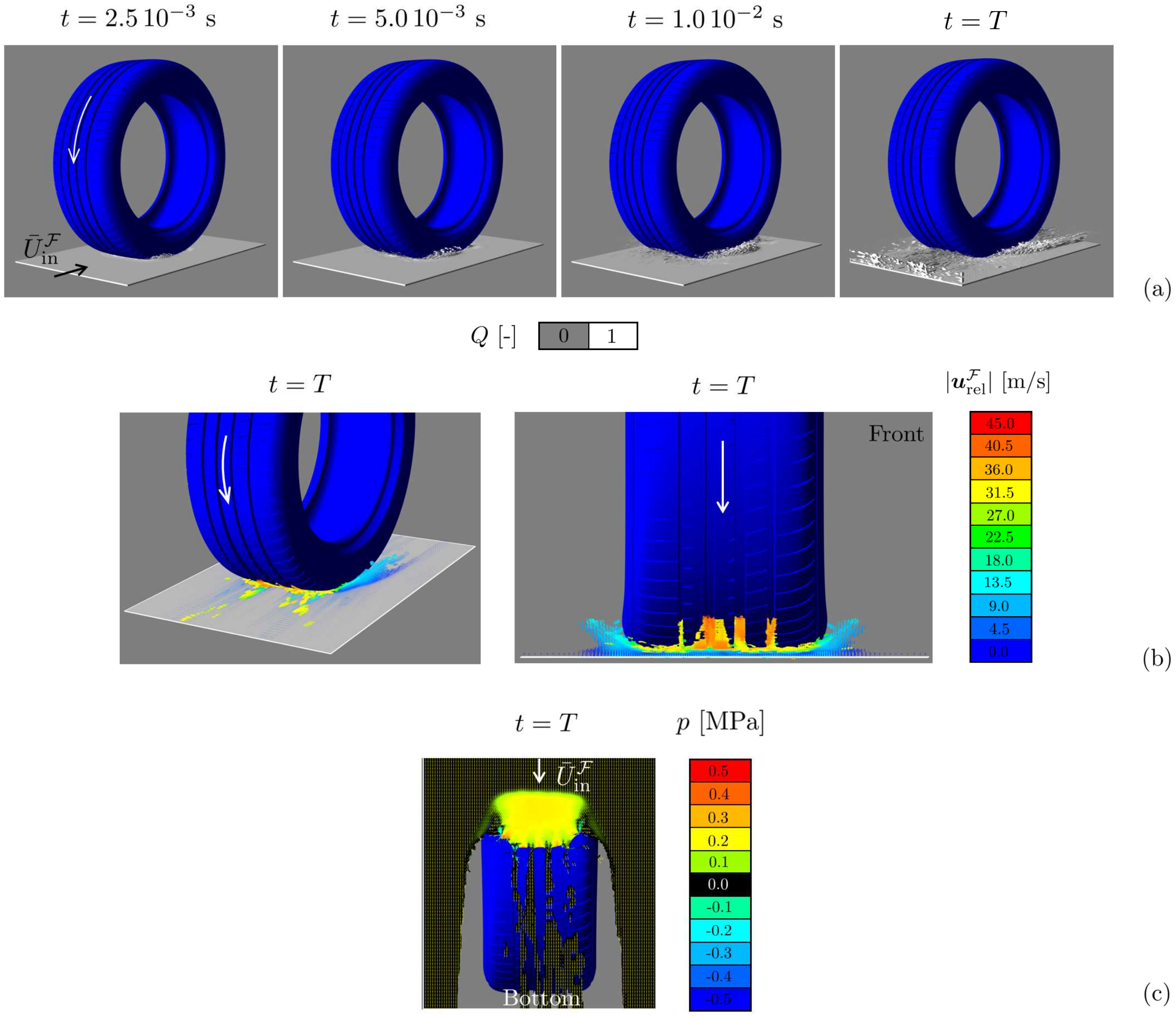

Figure 5.5: Snapshots of fluid results: VOF variable Q at different time instants (a) as well as fluid relative velocity $\boldsymbol{u}^{\mathcal{F}}_{\text{rel}}$ (b) and fluid pressure p on the road (arrows indicate the direction of fluid particles) (c) at the end of the simulation at $t = T$ (tire T01).

the computational method used for solving the problem at hand. In the present example, the SR reaches ~ 11 % after $t = T$.

Based on the reference model T01, various test cases of the *Dunlop SP Quattromaxx* tire were constructed in order to improve the overall coupled simulation with the proposed coupling scheme. In the same manner as for the Grosch wheel example, the effect of the solid predictor as well as the optimization of the FSI time step size based on the control of the interface energy criterion are considered for the full-scale tire model in the remaining of this chapter.

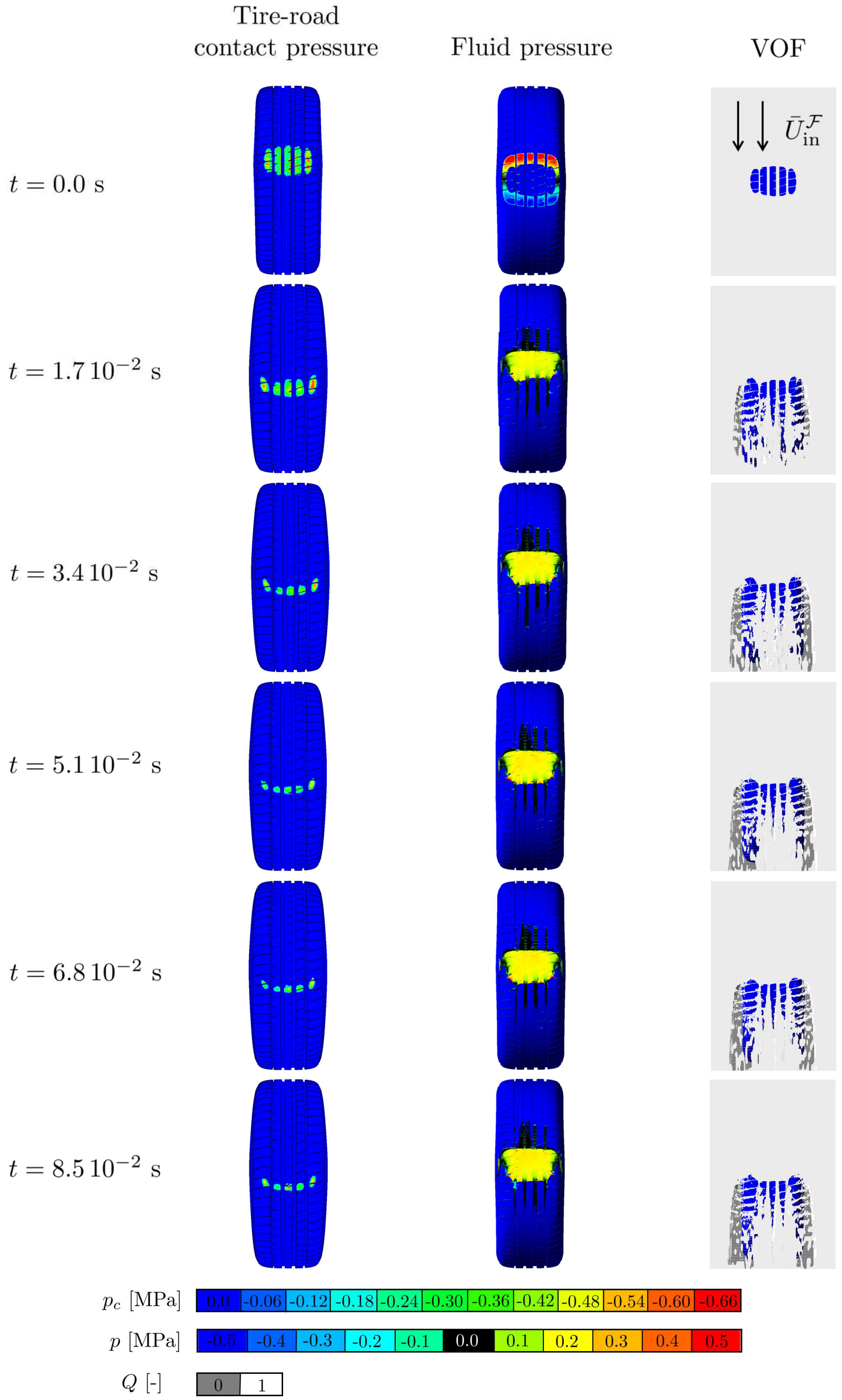

Figure 5.6: Snapshots of the contact pressure p_c, the fluid pressure p and the VOF variable Q in the tire footprint at different time stations (tire T01).

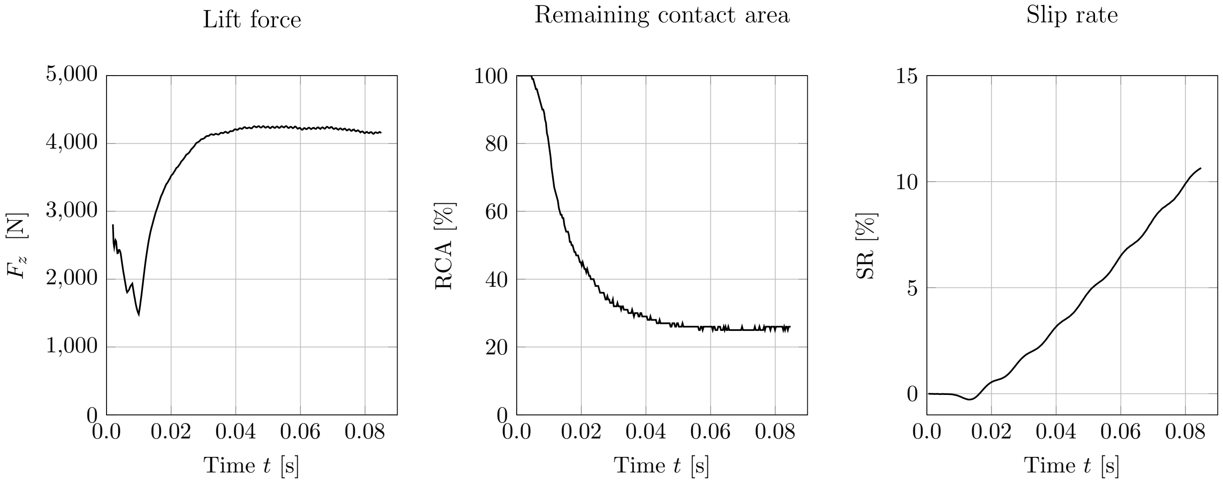

Figure 5.7: Full-scale tire: lift force F_z (moving average with 100 intervals), remaining contact area RCA and slip rate SR for the reference configuration T01.

5.2.3 Effect of solid predictor

The predictor given by expression (3.17) is applied at the FSI interface of the solid sub-domain of the tire example. Three test cases based on the reference example T01 with an FSI time step size $\Delta t = 2.0\,10^{-5}$ s are considered, namely T01P1, T01P2 and T01P1/2. T01P1 and T01P2 correspond to a 1st order ($\alpha_0 = 1$ and $\alpha_1 = 0$) and a 2nd order ($\alpha_0 = 1$ and $\alpha_1 = 1/2$) time-accurate predictor, whereas T01P1/2 is constructed with $\alpha_0 = 1/2$ and $\alpha_1 = 0$ to determine the predicted value $\boldsymbol{d}^{S\star}_{\Gamma,n+1}$ in equation (3.17). The interface energy-based error $||\Delta \boldsymbol{E}^{1}_{\Gamma,n}||_{\mathrm{L}_2}$ defined in (4.6) advising on the level of temporal error in the coupled system is plotted in Figure 5.8 together with the resulting lift force F_z exerted by the fluid on the tire for these three cases. Contrary to the Grosch wheel example, it can be observed that the higher the predictor order the higher the error, which leads for both T01P1 and T01P2 cases to an overestimated lift force compared to the reference example T01 and does not even converge after $t = T$. In an attempt to find an appropriate predictor the example T01P1/2 was built. In this case, the level of error is lower than for the 1st and 2nd order examples but remains higher than for the reference test case. Although the resulting lift force tends to converge, it is also overestimated compared to T01.

The situation is much different compared to that of the Grosch wheel example presented in section 4.3.2.3, where the use of a solid predictor improved the accuracy of the solution and allowed to use a larger coupling time step size. The discrepancy between these two models could be explained by the effect of the artificial added-mass effect instability summarized in section 3.3.5.2 that affects FSI problems with incompressible flows, small solid to fluid mass density ratios and which are solved with a loosely-coupled partitioned procedure. Although the tire and Grosch wheel models both possess these characteristics, the current Grosch wheel example can be considered as a weakly-coupled problem where the solid sub-domain is not impacted by the fluid flow, i.e. the deformation of the Grosch wheel due to the water layer is negligible. It is not

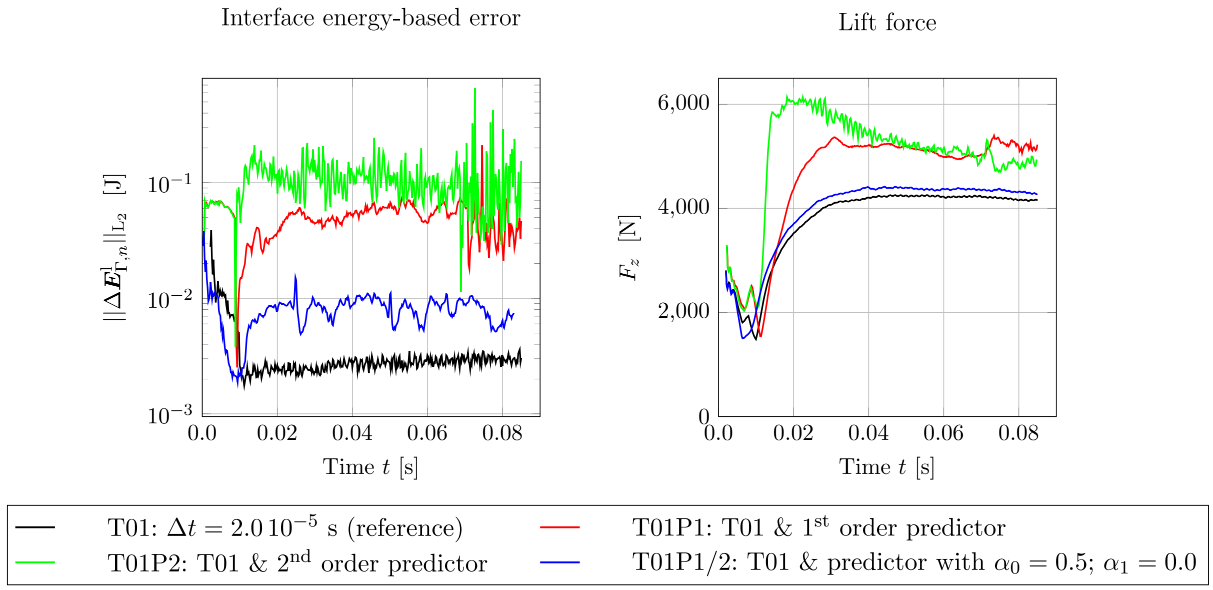

Figure 5.8: Full-scale tire: interface energy-based error $||\Delta \boldsymbol{E}^{\mathrm{I}}_{\Gamma,n}||_{\mathrm{L}_2}$ and lift force F_z (moving average with 100 intervals) obtained for various solid predictors and a constant FSI time step size $\Delta t = 2.0\,10^{-5}$ s compared to the reference test cases T01.

the case of the inflated tire, which is therefore much more sensible to the artificial added-mass effect instability. As a matter of fact, the tire hydroplaning problem is a very good example of FSI applications prone to the added-mass effect instability, which not only limits the choice of FSI time step size but is also triggered by the use of a solid displacement predictor as stated in section 3.3.5.2. The higher the predictor order, the earlier the instability occurs, which is in accordance with the above mentioned observations.

Ultimately, it can be stated that the use of a solid predictor within the present explicit coupling scheme is not indicated for computing the tire hydroplaning problem, due to its sensitivity to the artificial added-mass instability. Instead of improving the coupled solution procedure the solid displacement predictor rather deteriorates the overall solution.

5.2.4 Coupling time step size optimization

In order to improve the time efficiency of the tire hydroplaning problem while guaranteeing a given level of accuracy, the adaptive time stepping procedure introduced in section 4.2.2 is applied to the reference example T01. Since the solid predictor did not prove to be appropriate for the problem at hand, it is not further used in the test cases considered in this section, namely T1, T2 and T3. These three jobs were constructed with the algorithm parameters given in Table 5.1, which are completed with the following values to obtain the full set (4.9): $\kappa_{\mathrm{s}} = 0.95, \mathrm{M}_{\mathrm{down}} = 2$, $\kappa_{\mathrm{min}} = 0.9$ and $\kappa_{\mathrm{max}} = 2$. The optimized scaling factor $\kappa^{\circledcirc}_n$ is thus allowed to decrease by maximum 10 % and to double in size at most.

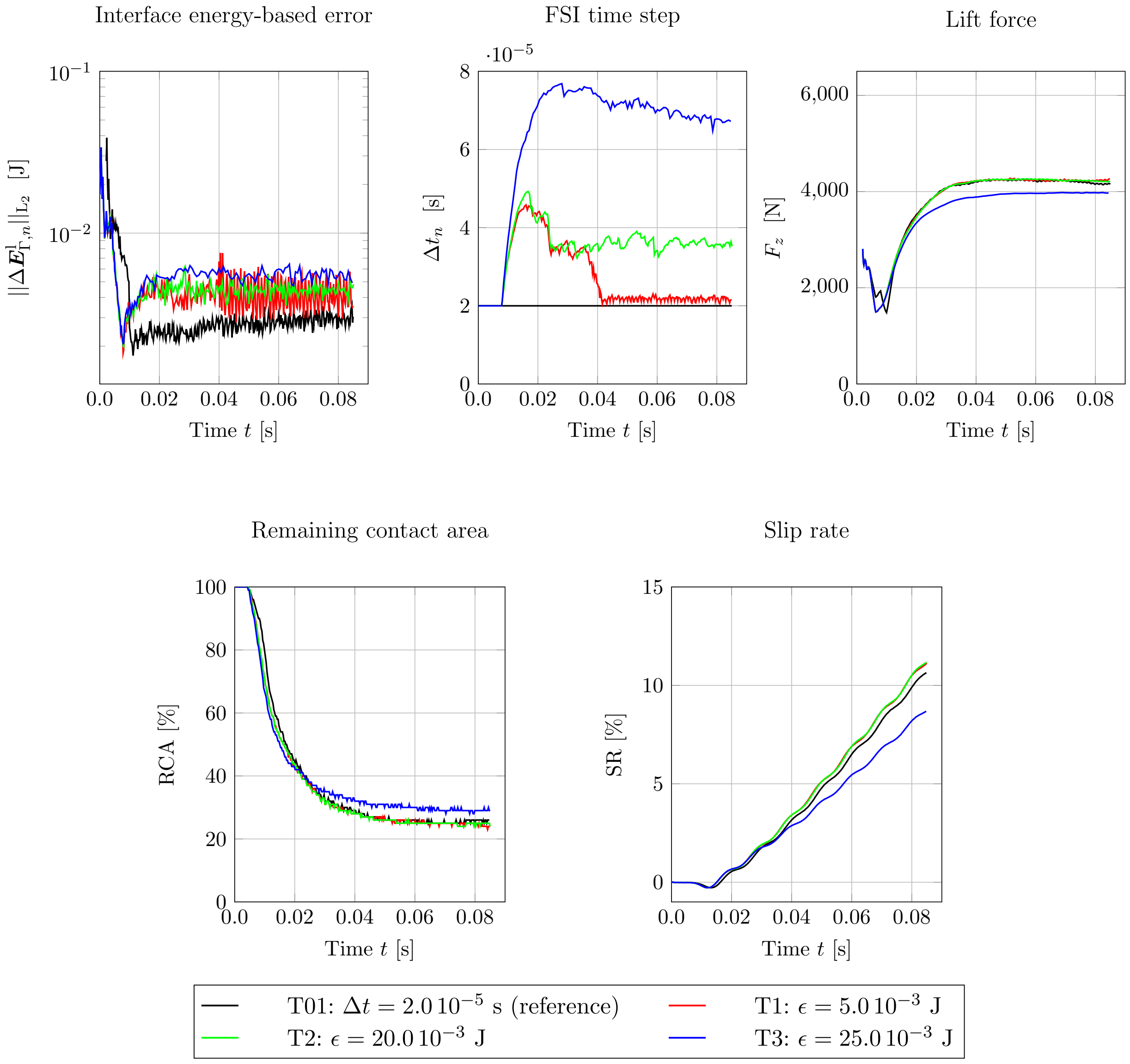

Figure 5.9: Full-scale tire: interface energy-based error $||\Delta \boldsymbol{E}^{\mathrm{l}}_{\Gamma,n}||_{\mathrm{L}_2}$, FSI time step size Δt_n, lift force F_z (moving average with 100 intervals), remaining contact area RCA and slip rate SR obtained for different adaptive FSI time stepping procedures compared to the reference test cases T01.

The temporal error resulting from the partitioning of the coupled problem is plotted along the obtained FSI time step size over the course of the simulation in Figure 5.9. All three cases using an adaptive coupling time step lead to a specific level of error, which is evidently higher for a decreasing level of expected accuracy (corresponding to an increasing value of the tolerance factor ϵ). The meaningful output results for computing the tire wet performance are also plotted in Figure 5.9 and show that both T1 and T2 lead to a level of accuracy that is similar to the reference test case. Indeed, the converged lift force, the RCA and the SR vary of only ~ 1 %, ~ 2 %, ~ 4 % respectively, compared to the reference T01. The overall simulation time is accordingly reduced by ~ 13 % in T1 and by ~ 33 % in T2. The level of accuracy wished for the test case T1 is too restrictive in comparison to T2 and leads to a coupling time step size that eventually oscillates around that of the reference test T01 ($\Delta t = 2.0\,10^{-5}$ s). It can be added that the error for T1 undergoes larger variations, which leads to an unstable time step size after $t \approx 0.04$ s. This instability also participates in keeping the time step size low. As for the case T3, the tolerance ϵ is too loose and leads to larger deviations for the outputs compared to the reference model. In T3, the overall simulation time is decreased by about 50 % but the model fails in delivering a sufficient level of accuracy and is therefore rejected. As was already observed in section 4.3 for simplified examples, these results once again show that the current procedure allows to automatically derive an optimum FSI time step size for a given level of wanted accuracy. In the current case, the optimum is hence obtained by T2, which reduces the simulation time of about 1/3 compared to the reference example T01, while keeping a level of accuracy that is totally acceptable.

In doing so, the objective of optimizing the time efficiency of the tire hydroplaning simulation by controlling the level of accuracy through a rigorous quantification of the partitioning error within the system has been reached for the current industrial application. Furthermore, the proposed method only requires little effort in terms of implementation and is not intrusive in that no modifications are needed in the single-field solvers. It is therefore also expected to be easily transferable to other FSI applications in an industrial environment. The only requirement consists in having a suitable platform where the proposed method can be implemented, which was achieved by the elaboration of the coupling shell presented in section 4.1.2. It can also be stated that an example of a loosely-coupled solution procedure has been presented in this thesis that can be used to solve FSI problems, which are per design strongly-coupled such as the tire hydroplaning problem. Even though the artificial added-mass effect introduces serious limitations one should remain aware of, the proposed method based on the interface energy criterion has proved to work successfully for solving such problems.

6 Conclusions and outlook

In this thesis, the tire hydroplaning problem is numerically simulated in order to assess the wet performance of real tire designs in an industrial environment. Due to the cost and the limited number of data that can be retrieved from experimental measurements, computational results are needed in order to better understand the underlying physical phenomenon and evaluate the wet performance for new tire designs. The simulation of the tire hydroplaning problem is a complicated application in the field of computational fluid-structure interaction due to the complexity of each sub-domain and their coupling. Therefore, particular attention is required to create the computational model and build an appropriate solution procedure. To do so, a loosely-coupled partitioned approach is used to couple both the solid and fluid sub-domains, which are computed by a pair of solvers specifically chosen to tackle the tire hydroplaning problem at hand. The goal of this thesis relies on optimizing the coupling strategy based on the software package available in order to improve the solution to the overall FSI problem in terms of robustness and time efficiency, while monitoring the accuracy in the coupled system. The choice for the solid solver is guided by its ability to accurately compute a composite solid system with large deformations, which includes a large variety of materials, complex contact occurrences and transient loading conditions. The discretization in space of the solid sub-domain is achieved by the FEM and an explicit procedure is used to advance the system in time. The fluid solver is based on a fixed-grid approach to compute the flow using the FVM together with the VOF method to treat the free-surface. The efficiency of the fluid-structure interface reconstruction within the fluid mesh, especially in the case of complex interface geometries as it is the case here, has played in favor of the fluid solver employed in this work.

The starting point is a coupling algorithm embedded in one of the single-field software used as a black-box for coupling this particular pair of solvers. Limiting strongly the possibilities for improving the coupling strategy by preventing any access to it, the preexisting coupling platform is replaced by a new coupling shell. The latter has been implemented to grant flexibility in order to control the execution of both solvers, extract single-field outputs and use coupling data for further developments. In particular, interface variables can now be manipulated, which allows for the use of a predictor for the solid displacement and a corrector for the fluid load at the interface, while the coupling time step size can also be freely adapted. However, due to the black-box character of each single-field solver, a full control of the coupling procedure cannot be achieved. The largest limitation stems from the fact that, in their current configuration, the solvers can only be advanced in time without redoing an already computed time step. Even though the proposed coupling framework would allow it, this restriction excludes any possibility of implementing an implicit coupling approach. In the end, a flexible platform is implemented that is generic and non-intrusive in the sense that it does not require any modification to the solvers. In that, it is appropriate for industrial applications usually using trusted commercial and black-box tools specifically designed for computing the considered problem.

By focusing on improving the coupling scheme, the current approach uses the energy artificially introduced at the fluid-structure interface by the staggering process in order to control the level of accuracy the coupled solution. An interface energy criterion is successfully employed to improve the robustness and optimize the time efficiency of the solution procedure, while monitoring the level of accuracy. To do so, a dynamic adaptation of the FSI time stepping procedure is introduced, which uses an interface energy-based error to modify the FSI time step size depending on the accuracy requirements in each time station of the simulation.

First, the coupling strategy within the newly implemented coupling shell is validated with help from a numerical model commonly used in the literature and considered as a standard benchmark for FSI problems, namely the flexible flag example. The same level of accuracy is reached by the current explicit coupling approach compared to implicit coupling procedures applied to the same problem in the literature. Furthermore, the application of the adaptive time stepping procedure has proved to reduce the overall simulation time by more than a factor 2. Second, the Grosch wheel example consisting in a simplified tire model rolling on a wet road surface is considered. It is shown that the application of a solid predictor for the interface displacements is advantageous in that it leads to a substantial reduction of the interface energy-based error, which in turn allows for the use of a coupling time step size as much as twice as large to solve the problem. Only the trivial corrector for the fluid loads at the interface can be employed, while other forms prove to introduce instabilities in the pressure distribution in the tire footprint making it unpractical for the current application. In this case, the adaptation of the FSI time step size does not speed-up the computation without affecting the accuracy of the solution. The reason for that relies on the fact that the deformation of the tire due the presence of fluid is negligible. As a consequence, the Grosch wheel problem can be seen as weakly-coupled, which limits the room for improving further the FSI time step size if an appropriate constant time step size can be found. The adaptive time stepping procedure can however be used to automatically find the optimum coupling time step size associated to the problem at hand. Third, the hydroplaning problem for a full-scale SUV tire is successfully computed with the proposed method. In this case, the solid predictor introduces instabilities, which can be associated to the artificial added-mass effect, and is therefore deemed unpractical. In accordance with observations made in the literature, this behavior is witnessed for the tire-water coupled problem, which by construction fulfills the conditions for a strong added-mass instability. The dynamic adaptive time stepping procedure applied to the real tire model proves to speed-up the overall simulation time by more than 30 %, while keeping a comparable level of accuracy.

Ultimately, depending on the application, various solutions are proposed to improve the robustness and time efficiency of the coupled simulation for a given level of accuracy. For weakly-coupled problems, a solid predictor shows to be beneficial, while the adaptive time stepping procedure proves to dramatically reduce the simulation time even for problems prone to the added-mass instability. As a matter of fact, it is shown that a loosely-coupled procedure, which represents the preferred solution from an industrial point of view for clear efficiency reasons, can be used even in the case of strong added-mass effect. Additionally, the proposed coupling shell, being non-intrusive and independent of any single-field solver, can be used for any coupled problem and transferred to other industrial applications. These include the evaluation of the noise (vibro-acoustic simulation) or of the aerodynamic performance of a tire. The latter aims at minimizing the air drag produced by the tire and only requires modifying the setup for the fluid sub-system. Also consisting in a weakly-coupled problem with a large density ratio $\rho^{\mathcal{S}}/\rho^{\mathcal{F}}$, some of the solutions proposed here can be directly used for this application. Nevertheless, the present coupling strategy can be further improved. In that context, an outlook on possible research directions for the future is given in the following.

The present coupling shell is flexible in the sense that multiple correcting features, as previously exposed, can be activated to improve the coupling procedure. Another feature, which is foreseen in the coupling shell but cannot be applied with the pair of solvers in their current configuration, is the use of implicit coupling schemes. In the latter, sub-iterations are performed between both fields in order to reduce the artificial added-mass effect. Some of these approaches, which could be implemented for the tire hydroplaning application are reviewed in section 1.3.2. In order to keep an efficient procedure, the idea consists in optimizing the coupling time step by switching automatically from a loose to a strong coupling strategy only at times of large interface error during the simulation. A limited number of sub-iterations should thus allow reducing locally the interface energy-based error and lead to the most stable yet efficient solution path. In the case of the tire, it could for example allow for the use of a solid predictor, which showed to deliver very promising results in the case of the Grosch wheel example. Having more control on the energy creation at the interface might also allow for the use of other forms of the fluid corrector, which showed inconclusive for all examples tested in the present work. Another possible improvement consists in implementing the non-collocated ISS coupling strategy, which shifts the resolution of both sub-domains by half an FSI time step size and tends to reduce the creation of artificial energy at the interface as explained in section 3.3.3. Its application is prevented by the coupling protocol followed by both solvers. More access to both black-box codes would allow exchanging interface data at time instants different for each sub-domain, which is required by the ISS strategy. The latter is however more subjected to the added-mass effect and must therefore be considered with care. In general, more flexibility in the control of both solvers and in the coupling protocol would permit implementing different coupling strategies such as some of the promising approaches reviewed in section 3.3.4.

Bibliography

[1] J. E. Akin, T. E. Tezduyar, and M. Ungor. Computation of flow problems with the mixed interface-tracking/interface-capturing technique (MITICT). *Computers & Fluids*, 36:2–11, 2007.

[2] I. Akkerman, Y. Bazilevs, C. Kees, and M. Farthing. Isogeometric analysis of free-surface flow. *Journal of Computational Physics*, 230:4137–4152, 2011.

[3] A. Aksenov, A. Dyadkin, and A. Gudzovsky. Numerical simulation of car tire aquaplaning. In *ECCOMAS Conference on Computational Fluid Dynamics*, pages 815–820, Paris, France, 1996.

[4] A. Aksenov, A. Dyadkin, and V. Pokhilko. Overcoming of barrier between CAD and CFD by modified finite volume method. In *ASME Pressure Vessels and Piping Division Conference*, San Diego, USA, 1998.

[5] A. Aksenov, V. Pokhilko, and A. Dyadkin. Numerical simulation of water flow around ship with screw propeller. In *ASME Pressure Vessels and Piping Division Conference*, Atlanta, USA, 2001.

[6] M. Astorino, F. Chouly, and M. A. Fernández. Robin based semi-implicit coupling in fluid-structure interaction: stability analysis and numerics. *SIAM Journal on Scientific Computing*, 31:4041–4065, 2010.

[7] F. P. T. Baaijens. A fictitious domain/mortar element method for fluid-structure interaction. *International Journal for Numerical Methods in Fluids*, 35:743–761, 2001.

[8] S. Badia, F. Nobile, and C. Vergara. Fluid-structure partitioned procedures based on Robin transmission conditions. *Journal of Computational Physics*, 227:7027–7051, 2008.

[9] J. W. Banks, W. D. Henshaw, and D. W. Schwendeman. An analysis of a new stable partitioned algorithm for FSI problems. Part II: Incompressible flow and structural shells. *Journal of Computational Physics*, 268:399–416, 2014.

[10] J. W. Banks, W. D. Henshaw, and D. W. Schwendeman. An analysis of a new stable partitioned algorithm for FSI problems. Part I: Incompressible flow and elastic solids. *Journal of Computational Physics*, 269:108–137, 2014.

[11] K.-J. Bathe. *Finite element procedures*. Prentice-Hall, 1996.

[12] K.-J. Bathe and H. Zhang. Finite element developments for general fluid flows with structural interactions. *International Journal for Numerical Methods in Engineering*, 60: 213–232, 2004.

[13] T. Belytschko, T. Rabczuk, A. Huerta, and S. Fernández-Méndez. Meshfree Methods. In *Encyclopedia of Computational Mechanics*. John Wiley & Sons, 2004.

[14] T. Belytschko, W. K. Liu, B. Moran, and K. Elkhodary. *Nonlinear finite elements for continua and structures*. Wiley, 2013.

[15] M. J. Berger and P. Colella. Local adaptative mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, 84:64–84, 1989.

[16] M. J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53:484–512, 1984.

[17] C. Bertoglio, P. Moireau, and J.-F. Gerbeau. Sequential parameter estimation for fluid-structure problems: Application to hemodynamics. *International Journal for Numerical Methods in Biomedical Engineering*, 28:434–455, 2012.

[18] J. Blazek. *Computational fluid dynamics: principles and applications*. Butterworth-Heinemann, 3rd edition, 2015.

[19] F. J. Blom. A monolithical fluid-structure interaction algorithm applied to the piston problem. *Computer Methods in Applied Mechanics and Engineering*, 167:369–391, 1998.

[20] J. Bonet and R. D. Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge University Press, 1997.

[21] M. Brenk, H.-J. Bungartz, M. Mehl, R.-P. Mundani, D. Scholz, A. Düster, and D. Scholz. Efficient interface treatment for fluid-structure interaction on Cartesian grids. In *ECCOMAS Coupled Problems*, Barcelona, 2005. CIMNE.

[22] M. Brenk, H.-J. Bungartz, M. Mehl, and T. Neckel. Fluid-structure interaction on Cartesian grids: flow simulation and coupling environment. In *Fluid-Structure Interaction*, pages 233–269. Springer Berlin Heidelberg, 2006.

[23] M. Bukač, S. Čanić, R. Glowinski, J. Tambača, and A. Quaini. Fluid-structure interaction in blood flow capturing non-zero longitudinal structure displacement. *Journal of Computational Physics*, 235:515–541, 2013.

[24] E. Burman and M. A. Fernández. Stabilization of explicit coupling in fluid-structure interaction involving fluid incompressibility. *Computer Methods in Applied Mechanics and Engineering*, 198:766–784, 2009.

[25] P. Causin, J.-F. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Computer Methods in Applied Mechanics and Engineering*, 194:4506–4527, 2005.

[26] J. Cho, H. Lee, J. Sohn, G. Kim, and J. Woo. Numerical investigation of hydroplaning characteristics of three-dimensional patterned tire. *European Journal of Mechanics - A/Solids*, 25:914–926, 2006.

[27] R. Courant, K. Friedrichs, and H. Lewy. On the partial difference equations of mathematical physics. *IBM Journal of Research and Development*, 11:215–234, 1967.

[28] S. De, S. Das, J. Kuipers, E. Peters, and J. Padding. A coupled finite volume immersed boundary method for simulating 3D viscoelastic flows in complex geometries. *Journal of Non-Newtonian Fluid Mechanics*, 232:67–76, 2016.

[29] A. de Boer. *Computational fluid-structure interaction. Spatial coupling, coupling shell and mesh deformation.* Phd thesis, Delft University of Technology, 2008.

[30] A. de Boer, A. H. van Zuijlen, and H. Bijl. Review of coupling methods for non-matching meshes. *Computer Methods in Applied Mechanics and Engineering*, 196:1515–1525, 2007.

[31] E. A. de Souza Neto, D. Perić, and D. R. J. Owen. *Computational methods for plasticity: theory and applications*. Wiley, 2008.

[32] J. Degroote, P. Bruggeman, R. Haelterman, and J. Vierendeels. Stability of a coupling technique for partitioned solvers in FSI applications. *Computers & Structures*, 86:2224–2234, 2008.

[33] J. Degroote, K.-J. Bathe, and J. Vierendeels. Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction. *Computers & Structures*, 87: 793–801, 2009.

[34] S. Deparis, M. A. Fernández, and L. Formaggia. Acceleration of a fixed point algorithm for fluid-structure interaction using transpiration conditions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37:601–616, 2003.

[35] W. G. Dettmer and D. Perić. A computational framework for free surface fluid flows accounting for surface tension. *Computer Methods in Applied Mechanics and Engineering*, 195:3038–3071, 2006.

[36] W. G. Dettmer and D. Perić. A fully implicit computational strategy for strongly coupled fluid-solid interaction. *Archives of Computational Methods in Engineering*, 14:205–247, 2007.

[37] W. G. Dettmer and D. Perić. A new staggered scheme for fluid-structure interaction. *International Journal for Numerical Methods in Engineering*, 93:1–22, 2013.

[38] M. Donatellis, E. Gelosa, and R. Sangalli. Virtual treaded tire simulation as a design predictive tool : Application to tire hydroplaning. In *Simulia Customer Conference*, London, 2009.

[39] J. Donea and A. Huerta. *Finite element methods for flow problems*. Wiley, 2003.

[40] J. Donea, S. Giuliani, and J. Halleux. An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 33:689–723, 1982.

[41] J. Donea, A. Huerta, J.-P. Ponthot, and A. Rodríguez-Ferran. Arbitrary Lagrangian-Eulerian methods. In *Encyclopedia of Computational Mechanics*, chapter 14. Wiley, 2004.

[42] A. Dubey, A. Almgren, J. Bell, M. Berzins, S. Brandt, G. Bryan, P. Colella, D. Graves, M. Lijewski, F. Löffler, B. O'Shea, E. Schnetter, B. Van Straalen, and K. Weide. A survey of high level frameworks in block-structured adaptive mesh refinement packages. *Journal of Parallel and Distributed Computing*, 74:3217–3227, 2014.

[43] European Commission. Commission regulation (EU) No 1235/2011 amending Regulation (EC) No 1222/2009 of the European Parliament and of the Council. *Official Journal of the European Union*, L 317:17–23, 2011.

[44] C. Farhat and V. K. Lakshminarayan. An ALE formulation of embedded boundary methods for tracking boundary layers in turbulent fluid-structure interaction problems. *Journal of Computational Physics*, 263:53–70, 2014.

[45] C. Farhat and M. Lesoinne. On the accuracy, stability, and performance of the solution of three-dimensional nonlinear transient aeroelastic problems by partitioned procedures. *37th Structural Dynamics Meeting*, pages 629–641, 1996.

[46] C. Farhat and M. Lesoinne. Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems. *Computer Methods in Applied Mechanics and Engineering*, 182:499–515, 2000.

[47] C. Farhat, M. Lesoinne, and N. Maman. Mixed explicit/implicit time integration of coupled aeroelastic problems: three-field formulation, geometric conservation and distributed solution. *International Journal for Numerical Methods in Fluids*, 21:807–835, 1995.

[48] C. Farhat, M. Lesoinne, and P. Le Tallec. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 157:95–114, 1998.

[49] C. Farhat, C. Harris, and D. Rixen. Expanding a flutter envelope using accelerated flight data - Application to an F16 fighter configuration. In *41st Structures, Structural Dynamics, and Materials Conference and Exhibit*, Reston, Virigina, 2000. AIAA 2000-1702.

[50] C. Farhat, P. Geuzaine, and C. Grandmont. The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. *Journal of Computational Physics*, 174:669–694, 2001.

[51] C. Farhat, P. Geuzaine, and G. Brown. Application of a three-field nonlinear fluid-structure formulation to the prediction of the aeroelastic parameters of an F-16 fighter. *Computers and Fluids*, 32:3–29, 2003.

[52] C. Farhat, K. G. van der Zee, and P. Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 195:1973–2001, 2006.

[53] C. Farhat, K. Maute, B. Argrow, and M. Nikbay. Shape optimization methodology for reducing the sonic boom initial pressure rise. *AIAA Journal*, 45:1007–1018, 2007.

[54] C. Farhat, A. Rallu, K. Wang, and T. Belytschko. Robust and provably second-order explicit-explicit and implicit-explicit staggered time-integrators for highly non-linear compressible fluid-structure interaction problems. *International Journal for Numerical Methods in Engineering*, 84:73–107, 2010.

[55] C. A. Felippa and K. C. Park. Staggered transient analysis procedures for coupled mechanical systems: formulation. *Computer Methods in Applied Mechanics and Engineering*, 24:61–111, 1980.

[56] C. A. Felippa, K. C. Park, and J. A. DeRuntz. Stabilization of staggered solution procedures for fluid-structure interaction analysis. *Computational methods for fluid-structure interaction problems*, 26:95–124, 1977.

[57] C. A. Felippa, K. C. Park, and C. Farhat. Partitioned analysis of coupled mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 190:3247–3270, 2001.

[58] M. A. Fernández. Coupling schemes for incompressible fluid-structure interaction: implicit, semi-implicit and explicit. *SeMA Journal*, 55:59–108, 2011.

[59] M. A. Fernández and M. Moubachir. A Newton method using exact jacobians for solving fluid-structure coupling. *Computers & Structures*, 83:127–142, 2005.

[60] M. A. Fernández, J.-F. Gerbeau, and C. Grandmont. A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid. *International Journal for Numerical Methods in Engineering*, 69:794–821, 2007.

[61] M. A. Fernández, J. Mullaert, and M. Vidrascu. Generalized Robin-Neumann explicit coupling schemes for incompressible interaction: stability analysis and numerics. *International Journal for Numerical Methods in Engineering*, 101:199–229, 2015.

[62] J. H. Ferziger and M. Perić. *Computational methods for fluid dynamics*. Springer Berlin Heidelberg, 2002.

[63] D. P. Flanagan and T. Belytschko. A uniform strain hexahedron and quadrilateral with orthogonal hourglass control. *International Journal for Numerical Methods in Engineering*, 17:679–706, 1981.

[64] C. Förster. *Robust methods for fluid-structure interaction with stabilised finite elements*. Phd thesis, Universität Stuttgart, 2007.

[65] C. Förster, W. A. Wall, and E. Ramm. On the geometric conservation law in transient flow calculations on deforming domains. *International Journal for Numerical Methods in Fluids*, 50:1369–1379, 2006.

[66] C. Förster, W. A. Wall, and E. Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 196:1278–1293, 2007.

[67] Fraunhofer Institut for Algorithms and Scientific Computing SCAI. MpCCI 4.5 Coupling environment documentation, 2017.

[68] D. Frenkel and B. Smit. *Understanding molecular simulation: from algorithms to applications.* Academic Press, New York, 2nd edition, 2011.

[69] M. W. Gee, U. Küttler, and W. A. Wall. Truly monolithic algebraic multigrid for fluid-structure interaction. *International Journal for Numerical Methods in Engineering*, 85: 987–1016, 2011.

[70] S. Geller, C. Janssen, and M. Krafczyk. A Lattice Boltzmann approach for distributed three-dimensional fluid-structure interaction. In *Progress in Computational Physics: Novel Trends in Lattice-Boltzmann Methods*, volume 3, chapter 8, pages 199–216. Bentham Books, 2013.

[71] M. Geradin and D. J. Rixen. *Mechanical vibrations: theory and application to structural dynamics*. Wiley, 3rd edition, 2015.

[72] J.-F. Gerbeau and M. Vidrascu. A quasi-Newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37:631–647, 2003.

[73] J.-F. Gerbeau, M. Vidrascu, and P. Frey. Fluid-structure interaction in blood flows on geometries based on medical imaging. *Computers & Structures*, 83:155–165, 2005.

[74] A. Gerstenberger. *An XFEM based fixed-grid approach to fluid-structure interaction*. Phd thesis, Technische Universität München, 2010.

[75] J. Gillard, V. Decouvreur, and W. Wall. An explicit coupling scheme using an interface energy-based criterion for industrial fluid-structure interaction applications including tire hydroplaning. In preparation.

[76] H. Grogger and M. Weiss. Calculation of the three-dimensional free surface flow around an automobile tire. *Tire Science and Technology*, 24:39–49, 1996.

[77] H. Grogger and M. Weiss. Calculation of the hydroplaning of a deformable smooth shaped and longitudinally grooved tire. *Tire Science and Technology*, 25:265–287, 1997.

[78] G. Guidoboni, R. Glowinski, N. Cavallini, and S. Canic. Stable loosely-coupled-type algorithm for fluid-structure interaction in blood flow. *Journal of Computational Physics*, 228:6916–6937, 2009.

[79] H. Guillard and C. Farhat. On the significance of the geometric conservation law for flow computations on moving meshes. *Computer Methods in Applied Mechanics and Engineering*, 190:1467–1482, 2000.

[80] E. Hairer and G. Wanner. *Solving ordinary differential equations II: stiff and differential-algebraic problems*. Springer Berlin Heidelberg, 2nd edition, 1996.

[81] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving ordinary differential equations I: nonstiff problems*. Springer Berlin Heidelberg, 2nd edition, 1993.

[82] D. F. Hays and A. L. Browne. *The physics of tire traction - theory and experiment*. Plenum Press, 1974.

[83] M. Heil. An efficient solver for the fully coupled solution of large-displacement fluid-structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 193:1–23, 2004.

[84] J. G. Heywood, R. Rannacher, and S. Turek. Artificial boundaries and flux and pressure conditions for the incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 22:325–352, 1996.

[85] H. M. Hilber, T. J. R. Hughes, and R. L. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering & Structural Dynamics*, 5:283–292, 1977.

[86] C. W. Hirt and B. D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39:201–225, 1981.

[87] G. A. Holzapfel. *Nonlinear solid mechanics: a continuum approach for engineering*. Wiley, 2000.

[88] W. B. Horne and R. C. Dreher. Phenomena of pneumatic tire hydroplaning. Technical report, Nasa TN D-2056, 1963.

[89] G. Hou, J. Wang, and A. Layton. Numerical methods for fluid-structure interaction - A review. *Communications in Computational Physics*, 12:337–377, 2012.

[90] T. J. R. Hughes. *The finite element method: linear static and dynamic finite element analysis*. Dover, 2000.

[91] C. Kadapa, W. G. Dettmer, and D. Perić. A stabilised immersed boundary method on hierarchical b-spline grids for fluid-rigid body interaction with solid-solid contact. *Computer Methods in Applied Mechanics and Engineering*, 318:242–269, 2017.

[92] C. Kassiotis, A. Ibrahimbegovic, R. Niekamp, and H. G. Matthies. Nonlinear fluid-structure interaction problem. Part I: Implicit partitioned algorithm, nonlinear stability proof and validation examples. *Computational Mechanics*, 47:305–323, 2011.

[93] T. Klöppel and W. A. Wall. A novel two-layer, coupled finite element approach for modeling the nonlinear elastic and viscoelastic behavior of human erythrocytes. *Biomechanics and Modeling in Mechanobiology*, 10:445–459, 2011.

[94] T. Klöppel, A. Popp, U. Küttler, and W. A. Wall. Fluid-structure interaction for non-conforming interfaces based on a dual mortar formulation. *Computer Methods in Applied Mechanics and Engineering*, 200:3111–3126, 2011.

[95] S. S. Kumar, K. Anupam, T. Scarpas, and C. Kasbergen. Study of hydroplaning risk on rolling and sliding passenger car. In *Procedia - Social and Behavioral Sciences*, volume 53, pages 1020–1028. Elsevier Ltd., 2012.

[96] U. Küttler. *Effiziente Lösungsverfahren für Fluid-Struktur-Interaktions-Probleme*. Phd thesis, Technische Universität München, 2009.

[97] U. Küttler and W. A. Wall. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43:61–72, 2008.

[98] U. Küttler and W. A. Wall. Vector extrapolation for strong coupling fluid-structure interaction solvers. *Journal of Applied Mechanics*, 76:021205–1–021205–7, 2009.

[99] U. Küttler, M. W. Gee, C. Förster, A. Comerford, and W. A. Wall. Coupling strategies for biomedical fluid-structure interaction problems. *International Journal for Numerical Methods in Biomedical Engineering*, 26:305–321, 2010.

[100] V. K. Lakshminarayan and C. Farhat. Nonlinear aeroelastic analysis of highly flexible flapping wings using an ALE formulation of embedded boundary method. In *52nd Aerospace Sciences Meeting*, National Harbor, Maryland, 2014. AIAA.

[101] M. G. Larson and F. Bengzon. *The finite element method: theory, implementation, and applications*. Springer Berlin Heidelberg, 2013.

[102] T. A. Laursen. *Computational contact and impact mechanics: fundamentals of modeling interfacial phenomena in nonlinear finite element analysis*. Springer Berlin Heidelberg, 2003.

[103] P. Le Tallec and P. Hauret. Energy conservation in fluid-structure interactions. In *Numerical methods for scientific computing. Variational problems and applications*, Barcelona, Spain, 2003. CIMNE.

[104] P. Le Tallec and J. Mouro. Fluid structure interaction with large structural displacements. *Computer Methods in Applied Mechanics and Engineering*, 190:3039–3067, 2001.

[105] M. Lesoinne and C. Farhat. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Computer Methods in Applied Mechanics and Engineering*, 134:71–90, 1996.

[106] M. Lesoinne and C. Farhat. Higher-order subiteration-free staggered algorithm for nonlinear transient aeroelastic problems. *AIAA Journal*, 36:1754–1757, 1998.

[107] L. Li, W. D. Henshaw, J. W. Banks, D. W. Schwendeman, and A. Main. A stable partitioned FSI algorithm for incompressible flow and deforming beams. *Journal of Computational Physics*, 312:272–306, 2016.

[108] S. Li and W. K. Liu. *Meshfree particle methods*. Springer Berlin Heidelberg, 2004.

[109] M. B. Liu and G. R. Liu. Smoothed particle hydrodynamics (SPH): an overview and recent developments. *Archives of Computational Methods in Engineering*, 17:25–76, 2010.

[110] F. Löffler, Z. Cao, S. R. Brandt, and Z. Du. A new parallelization scheme for adaptive mesh refinement. *Journal of Computational Science*, 16:79–88, 2016.

[111] R. Löhner. *Applied computational fluid dynamics techniques: an introduction based on finite element methods*. Wiley, 2nd edition, 2008.

[112] R. Löhner, C. Yang, and E. Oñate. Simulation of flows with violent free surface motion and moving objects using unstructured grids. *International Journal for Numerical Methods in Fluids*, 53:1315–1338, 2007.

[113] J. E. Marsden and T. J. R. Hughes. *Mathematical foundations of elasticity*. Dover, 1994.

[114] H. G. Matthies and J. Steindorf. Partitioned but strongly coupled iteration schemes for nonlinear fluid-structure interaction. *Computers and Structures*, 80:1991–1999, 2002.

[115] H. G. Matthies and J. Steindorf. Partitioned strong coupling algorithms for fluid-structure interaction. *Computers & Structures*, 81:805–812, 2003.

[116] U. M. Mayer, A. Popp, A. Gerstenberger, and W. A. Wall. 3D fluid-structure-contact interaction based on a combined XFEM FSI and dual mortar contact approach. *Computational Mechanics*, 46:53–67, 2010.

[117] M. Mayr. *A monolithic solver for fluid-structure interaction with adaptive time stepping and a hybrid preconditioner*. Phd thesis, Technische Universität München, 2016.

[118] M. Mayr, T. Klöppel, W. A. Wall, and M. W. Gee. A temporal consistent monolithic approach to fluid-structure interaction enabling single field predictors. *SIAM Journal on Scientific Computing*, 37:B30–B59, 2015.

[119] M. Mayr, W. Wall, and M. Gee. Adaptive time stepping for fluid-structure interaction solvers. *Finite Elements in Analysis and Design*, 141:55–69, 2018.

[120] M. Mehl, M. Brenk, I. L. Muntean, T. Neckel, and T. Weinzierl. Benefits of structured cartesian grids for the simulation of fluid-structure interactions. In *Third Asian-Pacific Congress on Computational Mechanics*, Kyoto, 2007.

[121] M. Mehl, B. Uekermann, H. Bijl, D. Blom, B. Gatzhammer, and A. H. van Zuijlen. Parallel coupling numerics for partitioned fluid-structure interaction simulations. *Computers & Mathematics with Applications*, 71:869–891, 2016.

[122] C. Michler, E. H. van Brummelen, and R. de Borst. An interface Newton-Krylov solver for fluid-structure interaction. *International Journal for Numerical Methods in Fluids*, 47: 1189–1195, 2005.

[123] C. Michler, E. H. van Brummelen, and R. de Borst. Error-amplification analysis of subiteration-preconditioned GMRES for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 195:2124–2148, 2006.

[124] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37:239–261, 2005.

[125] D. P. Mok. *Partitionierte Lösungsansätze in der Strukturdynamik und der Fluid-Struktur-Interaktion*. Phd thesis, Universität Stuttgart, 2001.

[126] D. P. Mok and W. A. Wall. Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures. In *Trends in Computational Structural Mechanics*, Barcelona, 2001. CIMNE.

[127] D. P. Mok, W. A. Wall, and E. Ramm. Accelerated iterative substructuring schemes for instationary fluid-structure interaction. *Computational Fluid and Solid Mechanics*, 2: 1325–1328, 2001.

[128] J. J. Monaghan. Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68: 1703–1759, 2005.

[129] L. Monasse, V. Daru, C. Mariotti, S. Piperno, and C. Tenaud. A conservative coupling algorithm between a compressible flow and a rigid body using an embedded boundary method. *Journal of Computational Physics*, 231:2977–2994, 2012.

[130] A. H. Muhr. Modeling the stress-strain behavior of rubber. *Rubber Chemistry and Technology*, 78:391–425, 2005.

[131] I. Newton. *Philosophiæ naturalis principia mathematica*. Londini : Jussu Societatis Regiae ac Typis Josephi Streater. Prostat apud plures bibliopolas, 1687.

[132] F. Nobile and C. Vergara. An effective fluid-structure interaction formulation for vascular dynamics by generalized Robin conditions. *SIAM Journal on Scientific Computing*, 30: 731–763, 2008.

[133] F. Nobile and C. Vergara. Partitioned algorithms for fluid-structure interaction problems in haemodynamics. *Milan Journal of Mathematics*, 80:443–467, 2012.

[134] M. Nool, E. J. Lingen, A. de Boer, and H. Bijl. Flecs, a flexible coupling shell application to fluid-structure interaction. In *Applied Parallel Computing. State of the Art in Scientific Computing.*, volume 4699, pages 1016–1025. Springer Berlin Heidelberg, 2007.

[135] M. Nool, E. J. Lingen, S. van Zuijlen, M. Stroeven, and H. Bijl. Flecs, a flexible coupling shell - Parallel application to fluid-structure interaction. In *International Conference of Computational Methods in Sciences and Engineering*, Rhodes, Greece, 2012. AIP conference proceedings 1504.

[136] R. Ogden. *Non-linear elastic deformations*. Dover, 1997.

[137] T. Okano and M. Koishi. A new computational procedure to predict transient hydroplaning performance of a tire. *Tire Science and Technology*, 29:2–22, 2001.

[138] L. Olovsson, K. Simonsson, and M. Unosson. Selective mass scaling for explicit finite element analyses. *International Journal for Numerical Methods in Engineering*, pages 1436–1445, 2005.

[139] S. Osher and F. Ronald. *Level set methods and dynamic implicit surfaces*, volume 153. Springer New York, 2003.

[140] S. V. Patankar. *Numerical heat transfer and fluid flow*. McGraw-Hill New York, 3rd edition, 1980.

[141] C. S. Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10:252–271, 1972.

[142] C. S. Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25:220–252, 1977.

[143] S. Piperno. Explicit/implicit fluid/structure staggered procedures with a structural predictor and fluid subcycling for 2D inviscid aeroelastic simulations. *International Journal for Numerical Methods in Fluids*, 25:1207–1226, 1997.

[144] S. Piperno and P.-E. Bournet. Numerical simulations of wind effects on flexible civil engineering structures. *Revue Européenne des Éléments Finis*, 8:659–687, 1999.

[145] S. Piperno and C. Farhat. Design of efficient partitioned procedures for the transient solution of aeroelastic problems. *Revue Européenne des Éléments Finis*, 9:655–680, 2000.

[146] S. Piperno and C. Farhat. Energy based design and analysis of staggered solvers for nonlinear transient aeroelastic problems. In *41st Structures, Structural Dynamics, and Materials Conference and Exhibit*, Reston, Virigina, 2000. AIAA.

[147] S. Piperno and C. Farhat. Partitioned procedures for the transient solution of coupled aeroelastic problems – Part II: energy transfer analysis and three-dimensional applications. *Computer Methods in Applied Mechanics and Engineering*, 190:3147–3170, 2001.

[148] S. Piperno, C. Farhat, and B. Larrouturou. Partitioned procedures for the transient solution of coupled aroelastic problems Part I: Model problem, theory and two-dimensional application. *Computer Methods in Applied Mechanics and Engineering*, 124:79–112, 1995.

[149] M. J. Poldneff and M. W. Heinstein. Computational mechanics of rubber and tires. In *Modeling and Simulation in Polymers*, chapter 8, pages 385–403. Wiley, 2010.

[150] A. Popp. *Mortar methods for computational contact mechanics and general interface problems*. Phd thesis, Technische Universität München, 2012.

[151] M. A. Puscas and L. Monasse. A three-dimensional conservative coupling method between an inviscid compressible flow and a moving rigid solid. 2014.

[152] A. Quaini and A. Quarteroni. A semi-implicit approach for fluid-structure interaction based on an algebraic fractional step method. *Mathematical Models and Methods in Applied Sciences*, 17:957–983, 2007.

[153] C. M. Rhie and W. L. Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal*, 21:1525–1532, 1983.

[154] S. Rugonyi and K.-J. Bathe. On finite element analysis of fluid flows fully coupled with structural interactions. *Computer Modelling in Engineering & Science*, 2:195–212, 2001.

[155] H. Schlichting and G. Klaus. *Boundary layer theroy*. Springer Berlin Heidelberg, 8th edition, 2000.

[156] B. Schott, S. Shahmiri, R. Kruse, and W. A. Wall. A stabilized Nitsche-type extended embedding mesh approach for 3D low- and high-Reynolds-number flows. *International Journal for Numerical Methods in Fluids*, 82:289–315, 2016.

[157] E. Seta, Y. Nakajima, T. Kamegawa, and H. Ogawa. Hydroplaning analysis by FEM and FVM: effect of tire rolling and tire pattern on hydroplaning. *Tire Science and Technology*, 28:140–156, 2000.

[158] M. Shadloo, G. Oger, and D. Le Touzé. Smoothed particle hydrodynamics method for fluid flows, towards industrial applications: Motivations, current state, and challenges. *Computers & Fluids*, 136:11–34, 2016.

[159] S. Shahmiri, A. Gerstenberger, and W. A. Wall. An XFEM-based embedding mesh technique for incompressible viscous flows. *International Journal for Numerical Methods in Fluids*, 65:166–190, 2011.

[160] S. Shin and D. Juric. Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity. *Journal of Computational Physics*, 180:427–470, 2002.

[161] J. C. Simo and T. J. R. Hughes. *Computational inelasticity*. Springer New York, 1998.

[162] SIMULIA. Abaqus, v6.12.13, User Guide. 2012.

[163] J. Stasch, B. Avci, and P. Wriggers. Numerical simulation of fluid-structure interaction problems by a coupled SPH-FEM approach. *Proceedings in Applied Mathematics and Mechanics*, 16:491–492, 2016.

[164] S. Succi. *The Lattice Boltzmann equation: for fluid dynamics and beyond*. Oxford University Press, 2001.

[165] Y. Sudhakar, J. Moitinho de Almeida, and W. A. Wall. An accurate, robust, and easy-to-implement method for integration over arbitrary polyhedra: Application to embedded interface methods. *Journal of Computational Physics*, 273:393–415, 2014.

[166] N. Takashi and T. J. R. Hughes. An arbitrary lagrangian-eulerian finite element method for interaction of fluid and a rigid body. *Computer Methods in Applied Mechanics and Engineering*, 95:115–138, 1992.

[167] K. Takizawa, K. Tanizawa, T. Yabe, and T. E. Tezduyar. Ship hydrodynamics computations with the CIP method based on adaptive Soroban grids. *International Journal for Numerical Methods in Fluids*, 54:1011–1019, 2007.

[168] K. Takizawa, T. E. Tezduyar, and R. Kolesar. FSI modeling of the Orion spacecraft drogue parachutes. *Computational Mechanics*, 55:1167–1179, 2015.

[169] T. E. Tezduyar, M. Behr, and J. Liou. A new strategy for finite element computations involving moving boundaries and interfaces. The deforming-spatial-domain/space-time procedure: I. The concept and the preliminary numerical tests. *Computer Methods in Applied Mechanics and Engineering*, 94:339–351, 1992.

[170] T. E. Tezduyar, K. Takizawa, C. Moorman, S. Wright, and J. Christopher. Space-time finite element computation of complex fluid-structure interactions. *International Journal for Numerical Methods in Fluids*, 64:1201–1218, 2010.

[171] The Goodyear Tire & Rubber Company. Internal documentation.

[172] TU Delft. *The Flexible Coupling Shell.* User guide, 2009.

[173] S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100:25–37, 1992.

[174] E. H. van Brummelen, S. J. Hulshoff, and R. de Borst. Energy conservation under incompatibility for fluid-structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 192:2727–2748, 2003.

[175] R. van Loon, P. D. Anderson, and F. N. van de Vosse. A fluid-structure interaction method with solid-rigid contact for heart valve dynamics. *Journal of Computational Physics*, 217: 806–823, 2006.

[176] T. M. van Opstal and E. H. van Brummelen. A finite-element/boundary-element method for large-displacement fluid–structure interaction with potential flow. *Computer Methods in Applied Mechanics and Engineering*, 266:57–69, 2013.

[177] A. H. van Zuijlen, A. de Boer, and H. Bijl. Higher-order time integration through smooth mesh deformation for 3D fluid–structure interaction simulations. *Journal of Computational Physics*, 224:414–430, 2007.

[178] H. K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics: the finite volume method.* Pearson Education Limited, 2nd edition, 2007.

[179] J. Vierendeels, L. Lanoye, J. Degroote, and P. Verdonck. Implicit coupling of partitioned fluid-structure interaction problems with reduced order models. *Computers & Structures*, 85:970–976, 2007.

[180] S. Vincent, A. Sarthou, J.-P. Caltagirone, F. Sonilhac, P. Février, C. Mignot, and G. Pianet. Augmented Lagrangian and penalty methods for the simulation of two-phase flows interacting with moving solids. Application to hydroplaning flows interacting with real tire tread patterns. *Journal of Computational Physics*, 230:956–983, 2011.

[181] W. A. Wall. *Fluid-Struktur-Interaktion mit stabilisierten Finiten Elementen*. Phd thesis, Universität Stuttgart, 1999.

[182] W. A. Wall and T. Rabczuk. Fluid-structure interaction in lower airways of CT-based lung geometries. *International Journal for Numerical Methods in Fluids*, 57:653–675, 2008.

[183] W. A. Wall, S. Genkinger, and E. Ramm. A strong coupling partitioned approach for fluid-structure interaction with free surfaces. *Computers and Fluids*, 36:169–183, 2007.

[184] K. Wang, A. Rallu, J.-F. Gerbeau, and C. Farhat. Algorithms for interface treatment and load computation in embedded boundary methods for fluid and fluid-structure interaction problems. *International Journal for Numerical Methods in Fluids*, 67:1175–1206, 2011.

[185] K. Wang, J. Grétarsson, A. Main, and C. Farhat. Computational algorithms for tracking dynamic fluid-structure interfaces in embedded boundary methods. *International Journal for Numerical Methods in Fluids*, 70:515–535, 2012.

[186] K. G. Wang, P. Lea, and C. Farhat. A computational framework for the simulation of high-speed multi-material fluid-structure interaction problems with dynamic fracture. *International Journal for Numerical Methods in Engineering*, 104:585–623, 2015.

[187] S. W. Welch and J. Wilson. A volume of fluid based method for fluid flows with phase change. *Journal of Computational Physics*, 160:662–682, 2000.

[188] D. A. Wolf-Gladrow. *Lattice gas cellular automata and Lattice Boltzmann models*. Springer Berlin Heidelberg, 2000.

[189] P. Wriggers. *Computational contact mechanics*. Springer Berlin Heidelberg, 2006.

[190] P. Wriggers. *Nonlinear finite element methods*. Springer Berlin Heidelberg, 2008.

[191] R. Wüchner, A. Kupzok, and K.-U. Bletzinger. A framework for stabilized partitioned analysis of thin membrane-wind interaction. *International Journal for Numerical Methods in Fluids*, 54:945–963, 2007.

[192] J. Yan, A. Korobenko, X. Deng, and Y. Bazilevs. Computational free-surface fluid-structure interaction with application to floating offshore wind turbines. *Computers & Fluids*, 141:155–174, 2016.

[193] L. Yoshihara, C. J. Roth, and W. A. Wall. Fluid-structure interaction including volumetric coupling with homogenised subdomains for modeling respiratory mechanics. *International Journal for Numerical Methods in Biomedical Engineering*, 26:807–827, 2016.

[194] L. Zhang, A. Gerstenberger, X. Wang, and W. K. Liu. Immersed finite element method. *Computer Methods in Applied Mechanics and Engineering*, 193:2051–2067, 2004.

[195] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The finite element method: its basis and fundamentals*. Butterworth-Heinemann, 7th edition, 2013.

[196] O. C. Zienkiewicz, R. L. Taylor, and D. Fox. *The finite element method for solid and structural mechanics*. Butterworth-Heinemann, 7th edition, 2014.

[197] O. C. Zienkiewicz, R. L. Taylor, and P. Nithiarasu. *The finite element method for fluid dynamics*. Butterworth-Heinemann, 7th edition, 2014.