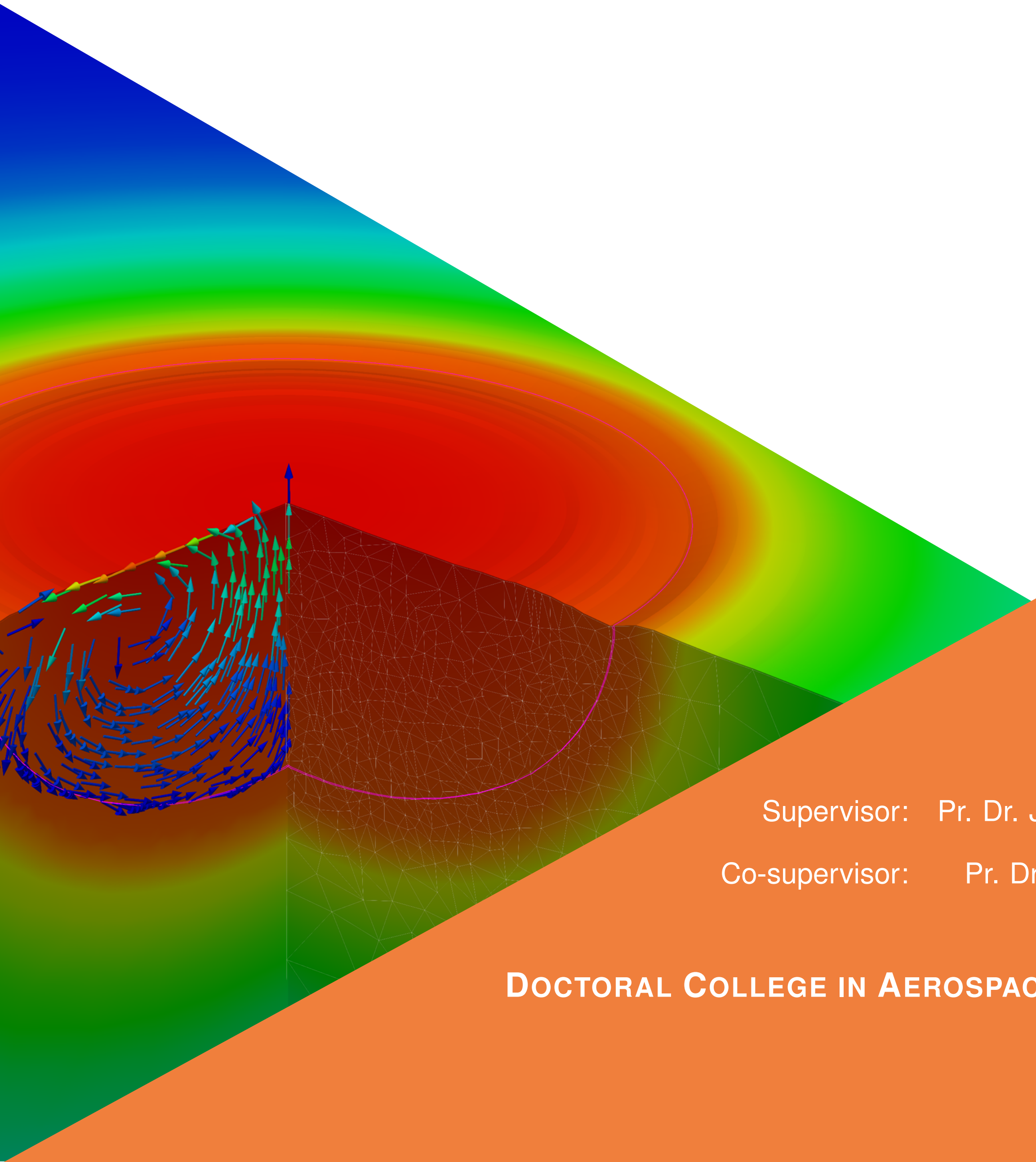
LIÈGE université
Sciences Appliquées

Towards a unified formulation for the simulation of thermo-fluid-solid problems with phase change

A thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy (PhD) in Engineering Science

by

Billy-Joe BOBACH



Supervisor: Pr. Dr. Jean-Philippe PONTHOT

Co-supervisor: Pr. Dr. Vincent E. TERRAPON

DOCTORAL COLLEGE IN AEROSPACE AND MECHANICS

OCTOBER 2023

Declaration of Authorship

I, Billy-Joe Bobach, declare that this thesis titled, “Towards a unified formulation for the simulation of thermo-fluid-solid problems with phase change” and the work presented in it are my own. I confirm that:

- This work was done wholly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

“*A smooth sea never made a skilled sailor.*”

Franklin D. Roosevelt

UNIVERSITÉ DE LIÈGE

Abstract

Faculty of Applied Science
Department of Aerospace & Mechanical Engineering

Doctor of Philosophy (PhD) in Engineering Science

Towards a unified formulation for the simulation of thermo-fluid-solid problems with phase change

by Billy-Joe Bobach

Industrial processes such as welding or additive manufacturing (AM) are driven by a concentrated heat source and involve phase change. Simulating these processes at the mesoscale presents a dilemma: It is challenging for classic coupled fluid-solid simulation strategies, due to the evolving melting front. Solid mechanics approaches may be insufficient, because it is well established in the literature that such processes can be sensitive to the convective flow in the liquid melt pool. Fluid dynamics approaches may be unable to reproduce the residual stresses that can cause warping and other defects.

This work presents a simulation technique that is able to capture fluids and solids in the same framework with one single solver, i.e. without coupling fluid and solid solvers. The technique is based on the Lagrangian Particle Finite Element Method (PFEM), which has been shown to be able to simulate fluid dynamics and solid mechanics problems in the literature. The key development in this work is the unified formulation for fluids and elastic solids: A single set of governing equations is used to describe a material that can locally be in its solid or fluid state. The thermal solution step governs the heat transfer and the phase change. Everything combined, this simulation technique is able to capture phase change, the convective flow in the melt pool (driven by buoyancy and the Marangoni effect) and the evolution of stresses in the elastic solid due to non-uniform thermal expansion.

This work outlines the mathematical formulation and algorithm of the simulation technique, then presents a series of verification test cases to finally demonstrate the capabilities claimed above. The demonstration test cases include a bird strike Fluid-Structure interaction (FSI) example, followed by two spot welding applications taken from the literature.

While work remains to be done for the accurate simulation of welding or AM processes, the method is successfully proven to be able to capture the flow in the fluid and the residual stresses in the solid, the fluid-solid interaction and the phase change correctly.

Acknowledgements

I would like to express my sincere gratitude and appreciation to the following individuals and institutions who have played a crucial role in the completion of my PhD thesis.

First and foremost, I am grateful to my primary advisor, Jean-Philippe Ponthot. I want to highlight his patience during long meetings and for my occasional missing of deadlines. I also thank him for keeping my mind off of administrative worries by securing my funding, when I needed more time. Both were of utmost importance to the completion of my doctorate. I am equally thankful towards my secondary advisor, Vincent E. Terrapon, who acted as a mentor and motivator, throughout the entire time. His invaluable advice regarding engaging and effective scientific communication was especially appreciated. Most importantly, I want to say thanks to both for their guidance, expertise, and support throughout this journey. Their invaluable feedback and insightful discussions have significantly shaped the direction and quality of my research.

I am indebted to the numerous colleagues, fellow researchers, and collaborators who have contributed to my work. Their collaborative spirit, intellectual discussions, and willingness to share their expertise have enriched my research experience. Special thanks goes to Romain Boman and Luc Papeleux for their insightful contributions to all technical aspects of my work, from providing reliable infrastructure to advice on good programming and of course their expertise in the implementation of numerical simulation codes. I want to particularly commend their immediate availability and readiness to answer any question and solve any problem, day and night.

I would like to acknowledge the financial support provided by the FNRS. Their generous funding through the FRIA grant has been instrumental in enabling me to pursue my research goals. I also thank the university of Liège for their preparation and extension grants that allowed me to acquire the FRIA grant and to complete my work, respectively.

I thank the jury for their time and willingness to read this work and to travel to Liège for the defense. At the time of writing, I am looking forward to a lively discussion and to learn from their outside perspective on the matter discussed in this thesis.

I would like to express my deepest gratitude to my friends and family for their unwavering support, understanding, and encouragement during this challenging endeavor. Their belief in my abilities and their presence in my life have been a constant source of motivation. First and foremost I thank my wife Vivian, who kept my back free and sacrificed much of her own time and energy to allow me to complete my doctorate. I thank my daughter for allowing me to spend time away from my work and enjoy memorable moments without a bit of guilt. I thank Mama for her emotional support, career guidance and strategic advice. I

also thank the rest of my family all over the world for their support and interest in my work.

Likewise, I extend my great appreciation to all the colleagues and friends at ULiège who discussed any matter with me, both research-related and otherwise, and for the pleasant times we shared during and after work. I especially enjoyed the nights out in the Carré, summer schools, hiking trips, board game nights and other adventures.

In conclusion, I acknowledge with deep appreciation the contributions and support of all those mentioned above and many others who have played a part in my academic and personal growth. Without their involvement, this thesis would not have been possible.

Contents

List of Figures

List of Tables

List of Abbreviations

AM	Additive manufacturing
BC	Boundary condition
CFD	Computational Fluid Dynamics
CPU	Central processing unit
CST	Constant strain triangle
DED	Direct energy deposition
DEM	Discrete element method
DFT	Discrete Fourier Transform
EBM	Electron beam melting
EVP	Elasto-visco-plastic (material)
FDM	Finite difference method
FEM	Finite element method
FFT	Fast Fourier Transform
FSI	Fluid-structure interaction
FVM	Finite volume method
HAZ	Heat affected zone
LBB	Ladyzhenskaya–Babuška–Brezzi (condition)
LBM	Lattice-Boltzmann method
L-PBF	Laser powder bed fusion
LS	Level set (method)
NR	Newton-Raphson (method)
NSE	Navier-Stokes equations
PBF	Powder bed fusion
PFEM	Particle finite element method
ppm	Parts per million ($= 1 \times 10^{-6}$)
PSPG	Pressure stabilizing Petrov-Galerkin (stabilization)
SEM	Scanning electron microscope
SLM	Selective laser melting
SPH	Smooth particle hydrodynamics
TIG	Tungsten Inert Gas (welding)
VoF	Volume of Fluid (method)
VP	Visco-plastic (material)
WAAM	Wire and Arc Additive Manufacturing

List of Symbols

Roman

a_r	Relaxation factor	-
$\boldsymbol{a}$	Thermal expansion vector	
A	Area	m^2
$\mathbf{A}$	Left hand side matrix of the system of equations	
b	Thickness	m
$\boldsymbol{b}$	Body force vector	$\mathrm{N\,m^{-3}}$
$\mathbf{B}$	Strain-displacement matrix	
c	Polynomial coefficient	
c_p	Specific isobaric heat capacity	$\mathrm{J\,kg^{-1}\,K^{-1}}$
C_{CK}	First coefficient of the Carman-Kozeny equation	$\mathrm{kg\,m^{-3}\,s^{-1}}$
$\mathbf{C}$	Heat capacity matrix	
d	Distance or displacement	m
$\mathbf{D}$	Discrete divergence operator	
e	Specific energy	$\mathrm{J\,kg^{-1}}$
E	Young's modulus	Pa
E_k	Kinetic energy	J
E_i	Internal energy	J
f_1	First natural frequency	$\mathrm{s^{-1}}$
f_{enh}	Enhancement factor	-
f_l	Liquid fraction (between solid and liquid state)	-
$\bar{\boldsymbol{f}}$	External distributed load	$\mathrm{N\,m^{-2}}$
F	Force	N
g	Gravitational acceleration, 9.81	$\mathrm{m\,s^{-2}}$
G	Shear modulus	Pa
h	Height or thickness	m
h_{conv}	Convective heat transfer coefficient	$\mathrm{W\,m^{-2}\,K^{-1}}$
H^1	Hilbert space of square integrable functions with square integrable first derivative	
$\mathbb{H}$	Stress-strain tensor (4th order)	Pa
$\mathbf{H}$	Stress-strain operator in Voigt notation (2nd order matrix)	Pa
I	Moment of inertia	m^4
$\mathbf{J}$	Jacobian matrix (transformation from current physical space to parent space)	

$\mathbf{J}_0$	Jacobian matrix (transformation from reference physical space to parent space)	
k	Heat conductivity	$\mathrm{W\,m^{-1}\,K^{-1}}$
$\boldsymbol{k}_{dev}$	Reference deviatoric stress vector	
K	Bulk modulus	Pa
$\mathbf{K}$	Stiffness matrix	
l	Length	m
L_m	Latent heat of fusion	$\mathrm{J\,kg^{-1}}$
L^2	Space of square integrable functions	
L^*	Target mesh edge length	m
m	Mass	kg
M	Moment of force	N m
$\boldsymbol{M}$	(Linear) momentum vector	N m
$\mathbf{M}$	Mass matrix	
n	Number	-
$\boldsymbol{n}$	Normal vector	N m
N	Shape function	-
$\boldsymbol{N}$	Shape function vector	-
$\nabla\boldsymbol{N}$	Shape function gradient vector	$\mathrm{m^{-1}}$
p	Pressure	Pa
P	Power	W
$\boldsymbol{P}$	Power vector	W
q	Heat flux	$\mathrm{W\,m^{-2}}$
$\tilde{q}$	Pressure test function	Pa
$\boldsymbol{q}$	Nodal solution vector	
Q	Heat	J
Q	Moment of Area	$\mathrm{m^3}$
$\mathcal{Q}$	Pressure trial and test function space	
r	Radius, Radial coordinate (Cylindrical coordinate system)	m
$\tilde{r}$	Temperature test function	K
$\boldsymbol{r}$	Right hand side vector of system of equations	
R	Volumetric heat source (power density)	$\mathrm{W\,m^{-3}}$
$\mathcal{R}$	Temperature trial function space	
$\mathcal{R}_0$	Temperature test function space	
s	Local variable along a curve, arc	m
$\boldsymbol{s}$	Nodal heat source vector ($\boldsymbol{s}_L$, $\boldsymbol{s}_R$ or $\boldsymbol{s}_q$)	
$\boldsymbol{s}$	Deviatoric stress tensor (first order column matrix in Voigt notation)	Pa
S	Surface	$\mathrm{m^2}$
$\boldsymbol{S}$	Momentum sink term (Carman-Kozeny equation)	
$\mathcal{S}$	Velocity vector trial function space	
$\mathcal{S}_0$	Velocity vector test function space	

t	Time	s
$\boldsymbol{t}$	Surface traction vector	s
T	Temperature	K
$\mathbf{T}$	Tangent stiffness matrix	
u	Displacement	m
v	Specific volume	$\mathrm{m^3\,kg^{-1}}$
$\boldsymbol{v}$	Velocity vector	$\mathrm{m\,s^{-1}}$
V	Volume	$\mathrm{m^3}$
$\dot{\boldsymbol{V}}$	Volume rate of change vector	$\mathrm{m^3\,s^{-1}}$
w	Width	m
w_i	Weight factor of Gauss point i	-
$\tilde{\boldsymbol{w}}$	Velocity vector test function	$\mathrm{m\,s^{-1}}$
W	Work	J
$\boldsymbol{x}$	Position vector	m
z	Axial coordinate (Cylindrical coordinate system)	m

Greek

α	Angle	rad
$\alpha = \frac{\lambda}{\rho c_p}$	Thermal diffusivity	$\mathrm{m^2\,s^{-1}}$
α_{crit}	Parameter of α-shape method	-
α_l	(Laser) absorptivity	-
α_V	(Volumetric) thermal expansion coefficient	$\mathrm{K^{-1}}$
γ	Contour, curve (e.g. projection of surface Γ onto a plane)	
γ	Surface tension (coefficient)	$\mathrm{N\,m^{-1}}$
$\frac{\partial\gamma}{\partial T}$	Marangoni coefficient	$\mathrm{N\,m^{-1}\,K^{-1}}$
γ_{crit}	Remeshing parameter (governs minimum nodal distance before deletion)	-
γ_s	Shear rate	$\mathrm{s^{-1}}$
Γ	Boundary	
δ	Logarithmic decrement	-
δ_{ij}	Kronecker delta	-
Δ	Difference	-
ϵ	Tolerance or residual	
ϵ_{rad}	Emissivity	-
ϵ_{CK}	Second coefficient of the Carman-Kozeny equation	-
$\boldsymbol{\varepsilon}$	Strain tensor (first order column matrix in Voigt notation)	-
$\dot{\boldsymbol{\varepsilon}}$	Strain rate tensor (first order column matrix in Voigt notation)	-
η	y-coordinate in isoparametric parent space	-
θ	Circumferential coordinate (Cylindrical coordinate system)	rad
θ	Slope	-

$\kappa = \frac{1}{K}$	Compressibility	Pa^{-1}
$\kappa = \frac{1}{r}$	Curvature	m^{-1}
κ_f	Bulk viscosity	Pa s
λ	Lagrange multiplier	
λ	Second Lamé parameter	Pa
λ_f	Second viscosity	Pa s
μ	(Dynamic) viscosity	Pa s
μ^*	Viscosity-like parameter	Pa s
ν	Poisson's ratio	-
ν_ϕ	Normalization factor for the residual ϵ_ϕ	-
ξ	x-coordinate in isoparametric parent space	-
ρ	Density	$\mathrm{kg\,m^{-3}}$
σ_{SB}	Stefan-Boltzmann constant, $5.670\,373 \times 10^{-8}$	$\mathrm{W\,m^{-2}\,K^{-4}}$
σ_y	Yield stress	Pa
$\boldsymbol{\sigma}$	Cauchy stress tensor (first order column matrix in Voigt notation)	Pa
$\bar{\sigma}_{VM}$	Equivalent von Mises stress	Pa
τ_{PSPG}	PSPG stabilization parameter	
ϕ	Placeholder for any given field variable	
φ_{LS}	Level set field variable	
ω_{crit}	Remeshing parameter (governs maximum nodal distance before splitting)	-
Ω	Domain	-

Subscripts (for a given variable ϕ)

ϕ_0	Reference configuration (i.e. beginning of time step/beginning of simulation)
ϕ_{abs}	Absolute
ϕ_{axi}	Axisymmetric
ϕ_b	Bending or bottom or bird
ϕ_c	Characteristic or circumcircle or cold side
ϕ_e	Element or edge
ϕ_{end}	End of simulation
ϕ_{ext}	External
ϕ_f	Fluid (phase/material/region)
ϕ_{cond}	Conductive heat transfer
ϕ_{conv}	Convective heat transfer
ϕ_{crit}	Critical
ϕ_{cut}	Cut-off (radius)
ϕ_{GP}	Gauss point
ϕ_h	Hot side
ϕ_{ine}	Inertial
ϕ_{int}	Internal

ϕ_l	Laser or end of laser heating phase or left side
ϕ_{lp}	Laser, projected
ϕ_L	Latent heat
ϕ_{liq}	Liquidus (temperature)
ϕ_m	Melting or melt pool or middle
ϕ_{max}	Maximum
ϕ_{mech}	Mechanical
ϕ_n	Normal or node
ϕ_{nf}	Neutral fiber
ϕ_q	Heat flux related, Neumann type boundary
ϕ_r	Right side
ϕ_{rect}	For a rectangle
ϕ_R	Heat source related
ϕ_{rad}	Radiative heat transfer
ϕ_{ref}	Reference
ϕ_{reg}	Regularization
ϕ_{rel}	Relative
ϕ_s	Solid (phase/material/region) or shear
ϕ_{sol}	Solidus (temperature)
ϕ_{ST}	Surface tension
ϕ_{std}	Standard deviation (radius)
ϕ_t	Tangential or top or traction
ϕ_{th}	Thermal
ϕ_λ	Lagrange multiplier related
ϕ_∞	Spatially far away or after a long time span

Other symbols

∇	Gradient operator
∇_S	Gradient operator tangential to a surface
$\bar{\phi}$	Imposed, equivalent or normalized quantity
$\hat{\phi}$	Non-dimensionalized quantity
$\tilde{\phi}$	Test function
$\frac{d\phi}{dt}$	Material derivative
$\lVert\phi\rVert$	Euclidean norm

Dimensionless numbers

C	Courant-Friedrichs-Levi (CFL) number	$\frac{v_c t_c}{l_c}$
Fo	Fourier number	$\frac{k t_c}{\rho\, c_p\, l_c^2}$
Ma	Marangoni number	$\frac{\rho\, c_p\, \partial\gamma/\partial T\, \Delta T_c\, l_c}{\mu\, k}$
Nu	Nusselt number	$\frac{h_{conv} l_c}{k}$
Pr	Prandtl number	$\frac{c_p\, \mu}{k}$
Ra	Rayleigh number	$\frac{\rho^2\, c_p\, \alpha_V\, \Delta T_c\, l_c^3\, g}{\mu\, k}$

Chapter 1

Introduction

1.1 Problem description

Computer simulations play an important role in engineering, as they attempt to virtually recreate a process of the real world. Since the real world is endlessly complex, simulations require models of reality and assumptions to simplify aspects of the process. On the one hand, a useful simulation must take into account all relevant aspects of the process, depending on the insight the engineer is hoping to gain from the process. On the other hand, all irrelevant aspects should be excluded from a simulation to reduce its complexity and cost. The difficulty for the engineer lies in finding the ideal balance between remaining sufficiently true to reality and cost-saving simplification.

For many common branches of engineering, individual simulation methods have established themselves as the go-to approach for solving their typical problems. With trusted simulation methods available and a body of experience using a given method, engineers can speed up development processes. This is because either the need for expensive and time-consuming experiments can be reduced or because experiments are complemented by simulations to give a more complete picture. This second point is especially important, when even state of the art experiments cannot (efficiently) give the insight that the engineer requires.

One of many examples of this shortfall of experiments is the type of processes studied in this doctoral thesis: phase change problems at a very small time and length scale. If we consider for example laser welding, we will find that experiments can only deliver very limited information. Because the heat source is very concentrated, the melted portion of the material being welded is extremely small, with a length scale between several micrometers and few millimeters. However, many physical phenomena take place at this small length scale in and around the melt pool. The individual behavior of these phenomena and the interaction between them can have a profound effect on the macroscopic result of the process. The problem begins with experiments not being able to resolve some of these phenomena both due to the small length scale and due to the difficult accessibility during the process. Using only experiments, engineers cannot easily investigate such a process systematically in-situ. Instead the engineer may sometimes only investigate the finished product and hypothesize about the physical phenomena that took place during the welding process. When the engineer wants to improve such a product or process, he or she is left with two (not mutually exclusive) pathways:

1. Repeat the process many times while systematically varying the process parameters, studying the result and finding patterns or
2. Falling back on simulations that can give insight into the microscopic physical behavior and gain a better understanding of how process parameters affect the microscopic physics and the how the microscopic physics affect the finished product.

This work aims at providing a foundation for engineers to take the second path in the context of phase change processes at small scales.

State of the art simulation methods for phase change at small length scales can already well resolve the fluid dynamics of the liquid melt pool and how it affects the morphology of the surface and the temperature evolution of the material. Such approaches, however, lack the ability to capture the solid mechanics effects around the melt pool, such as residual stresses, plasticity and cracking for example. The coupling of classic solid mechanics and fluid dynamics simulation methods, albeit a common approach in many other applications, is difficult for phase change problems. This is due to the unknown and constantly evolving fluid-solid interface.

This work aims to provide the bridge between fluid dynamics in the melt pool and solid mechanics in the surrounding substrate, without any such coupling. The key development is the combination of a unified material model for fluid and solid, paired with phase change capabilities. This allows to simulate the material as one that can change state within a single solver. The strong interaction between the solid and fluid can be captured and a better understanding of small scale phase change problems can be obtained in the future.

1.2 Industrial processes with phase change

Phase change plays a role in many areas of science and engineering, but even industrial processes that involve phase change are numerous. Casting, welding, soldering, injection molding and additive manufacturing are some of the most widespread and well-known examples.

While any engineering application involving phase change could benefit from the developments in this work, we will only consider a small selection of industrial processes to be used as reference: Laser beam welding and Laser Powder Bed Fusion (L-PBF), the latter being an Additive Manufacturing (AM) process. The focus lies on these examples, because of a favorable combination of conditions:

- The research interest in numerical methods for these processes is large, especially for L-PBF, due to a lack of experience with this new technology.
- Lasers as a heat source can be comparably straight-forward to model, depending on the material. In their review, Cook and Murphy [13] observe that even a very simplistic model for the laser-material interaction may deliver good results in additive manufacturing simulations. For a laser, there is no physical contact between heat source and material and the

heat source does not emit an electric or magnetic field that interacts with the material, reducing some of the complexity.

- In these processes the phase change occurs at such a small time and length scale (few millimeters to several tens of micrometers and over a duration of the order of microseconds) that experiments are challenging and the need for simulations is augmented.

In this section, these processes and their relevant physics are briefly revisited.

1.2.1 Length scales in welding and additive manufacturing simulations

One major obstacle in simulating such processes is the multi-scale nature of the problem [14]. For these applications, we define three distinct length scales: micro-, meso- and macro-scale. Since the physics at all scales are coupled, the long term aim is to tie together the different length scales for a complete understanding of such processes [13].

Macro-scale

Also known as the part scale, since it concerns the entire part or a major section of a part. At this scale, heat conduction through the part and heat convection and radiation at its surface play an important role. The accumulation of heat can be studied, when process parameters such as heat source intensity, scanning speed, scanning pattern etc. are varied. If the residual stresses can be obtained at the part scale, then the warping of the finished part after cooling can be computed. The Macro-scale is not considered any further in this work.

Meso-scale

At the intermediate scale, the meso-scale, both the melt pool and the heat source (i.e. the laser beam) are resolved. That means that the convective flow in the melt pool is captured, as the laser is heating it, together with the evolution of the melting front and the heat transfer in the solid and the surrounding atmosphere. Since the evolution of the phase front around the melt pool directly depends on the flow of heat, the thermal problem becomes non-linear. The topology of the re-solidified melt track can also be captured at this scale. Note that this work is entirely focused on the meso-scale and the following sections are written with this in mind.

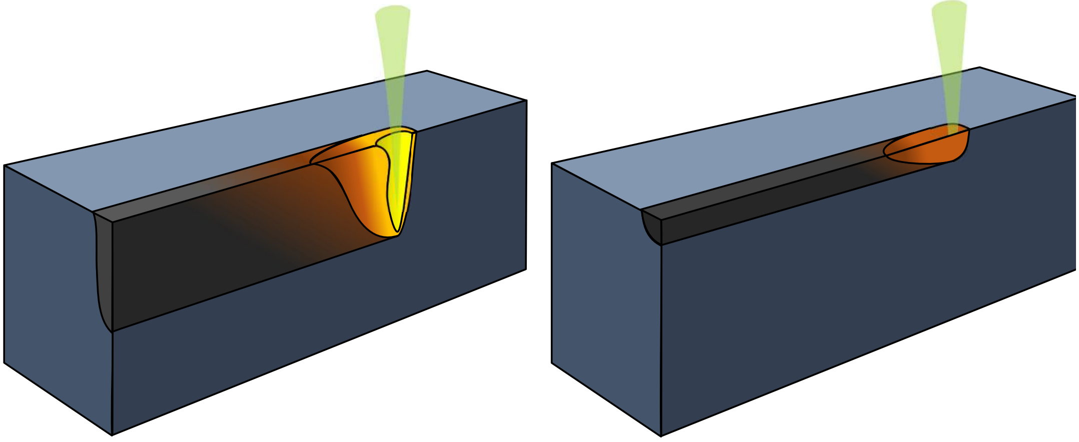
Micro-scale

At the microscopic scale the micro-structure evolution and grain growth are studied. In the context of laser beam welding and L-PBF, the grain growth is of particular interest, because the extreme temperature gradients at the melt pool lead to large columnar grains, which cause anisotropic material properties [15]. While these micro-scale effects could be included in this present simulation method in the future, it is not part of this work.

1.2.2 Laser beam welding

Welding processes use a heat source to locally melt a portion of a material, usually with the purpose of joining two parts together. The locally melted material is referred to as the melt pool. Welding is usually done with metals or polymers, rarely with ceramics (see e.g. Penilla et al. [16]). Welding is foremost characterized by the heat source, which is often included in the name of any given welding process (e.g. arc welding, laser welding, friction welding, resistance welding, gas welding etc.). The choice of the heat source comes with many implications as to what materials and geometries can be welded, the speed, quality and reliability of the process. Weman's *Welding process handbook* [17] provides a good overview over the range and characteristics of most welding techniques.

Laser beam welding is particularly interesting due to its high power density, leading to deep penetration, high process speed and minimal distortion, compared to other common welding techniques [17]. The deep penetration is the result of the typical keyhole formation during keyhole mode welding, one of the two modes, together with conduction mode welding.



(a) Keyhole mode.

(b) Conduction mode.

FIGURE 1.1: Two welding modes for laser welding. Laser (traversing left to right) green, liquid melt pool orange, keyhole yellow, substrate dark blue, weld track dark grey (adapted from www.ionix.fi/en/technologies/laser-processing/laser-welding, 15/05/2023).

For **keyhole mode welding** the power density of the heat source is high and the temperature increases around the heat source to such an extend that the material evaporates and a so-called keyhole forms. A keyhole is a narrow and deep cavity in the liquid melt pool, as illustrated in fig. 1.1a. For this cavity to form and sustain itself against collapsing, the recoil pressure of the evaporating material must be very high, which means that mass must evaporate at a large rate. The result is a much deeper weld, but the process can be unstable.

Conduction mode welding is characterized by a heat source power density low enough that little or no vaporization occurs. A shallow melt pool and therefore a shallow weld profile are obtained, as in fig. 1.1b.

More details on how to achieve one or the other welding mode and their respective characteristics can be found in Quintino et al. [18]. The keyhole mode laser welding is not investigated in this work due to the additional complexity compared to conduction mode laser welding.

Conduction mode laser welding can be further simplified under the following conditions:

- no added filler material,
- no gap between materials to be joined,
- both materials identical,
- no relative movement between laser heat source and substrate (i.e. spot welding).

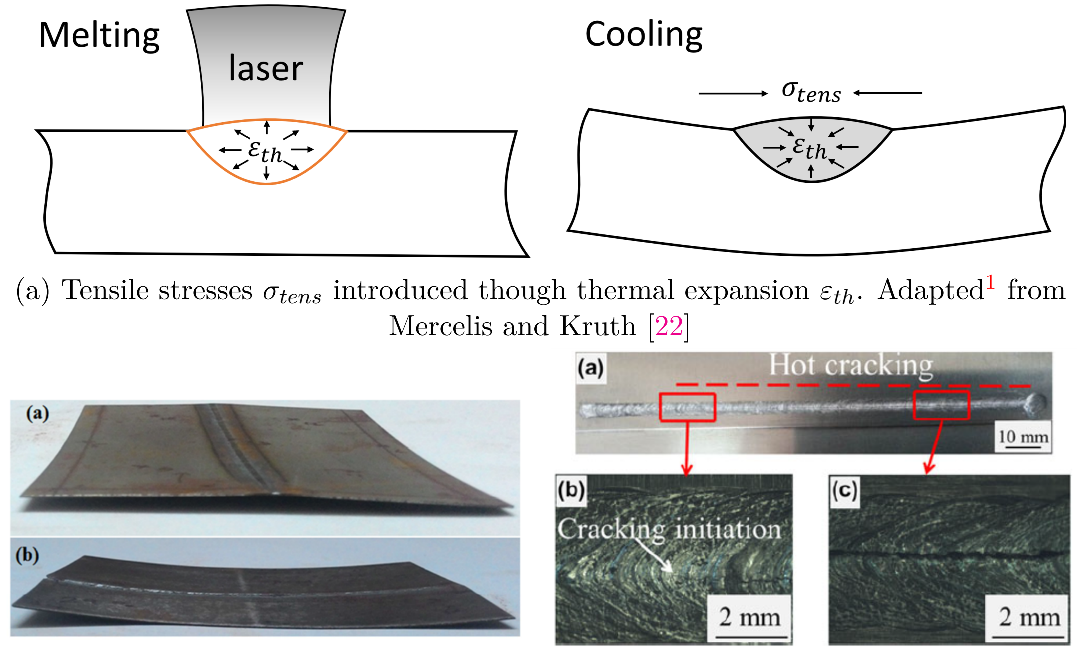
The method developed in this work is therefore demonstrated on laser spot welding without gaps or dissimilar materials. The method lays the groundwork for investigating the physics of welding in general and may be useful in better understanding the following issues that are common in welding processes:

- residual stresses and distortion,
- penetration depth and width,
- splatter,
- porosity.

Residual stresses are introduced into the part, because the stresses are low in the melt pool and therefore also in the newly solidified material. Since the newly solidified material has a temperature near the melting point or solidus temperature, it contracts as it cools down to room temperature. Because the local contraction is constrained by the surrounding solid material, a tensile stress occurs (see fig. 1.2a). As this introduction of residual stresses is local to where the melting occurred, a non-uniform distribution of residual stresses in the part can be obtained after cooling down. Residual stresses can lead to an undesirable change in shape of the part, referred to as distortion or warping [19]. Residual stresses can also lead to cracking (see fig. 1.2c, in the context of welding referred to as hot cracking, thermal cracking or contraction cracking [17]). Process conditions that lead to hot cracking are investigated in Zhang, X. et al. [20] and Zhang Y. et al. [21] among others. Note that cracking is not modeled in this work.

The **penetration** is the width and depth of the melt pool. Welds with a high aspect ratio (depth/width) are more prone to hot cracking [17]. An incomplete penetration or under-penetration, where the melt pool does not reach the root

[1] Mercelis and Kruth [22] point out that such a distortion is achieved even without melting. For this to occur, the thermal expansion must cause plastic deformation (not included in this work) in the surrounding material.



(a) Tensile stresses σ_{tens} introduced though thermal expansion ε_{th}. Adapted[1] from Mercelis and Kruth [22]

(b) Distortion of welded plates. Taken from Baruah and Bag [19].

(c) Hot cracking. Taken from Zhang et al. [21].

FIGURE 1.2: Residual stresses and distortion after welding.

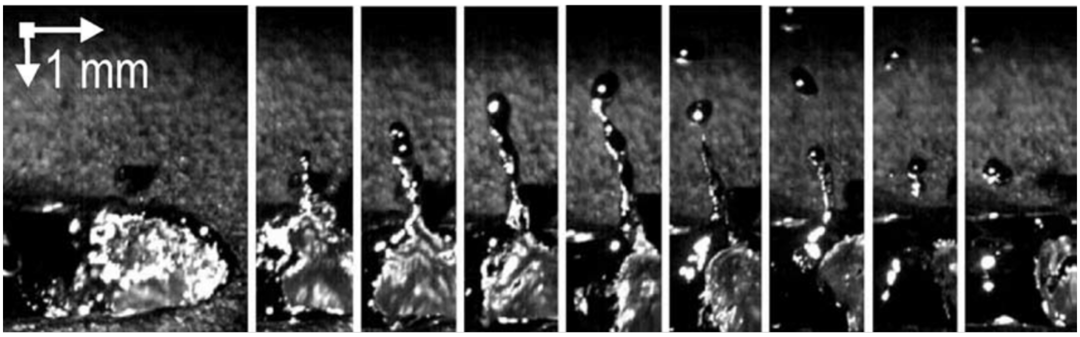


FIGURE 1.3: Spatter during laser welding: ejection of droplets from the melt pool. Taken from Kaplan and Powell [23].

of the weld joint is a common weld defect that can be avoided by choosing the right process parameters.

Spatter occurs when melted material is projected from the melt pool (see fig. 1.3) onto another surface of the part. While this may just affect the surface quality, spatter my also cause defects inside the finished part, when a spatter particle becomes included in a following weld bead. It may then create an internal weak spot of brittle material surrounded by freshly added material, which can be a crack initiation site or create a local bonding fault. A categorization of spatter mechanisms can be found in [23].

Porosity in the bulk of the material can be caused by several mechanisms. For laser beam welding, the keyhole instability is the main mechanism, where the keyhole collapses momentarily, trapping gas at the root of the melt pool [25] (see fig. 1.4).

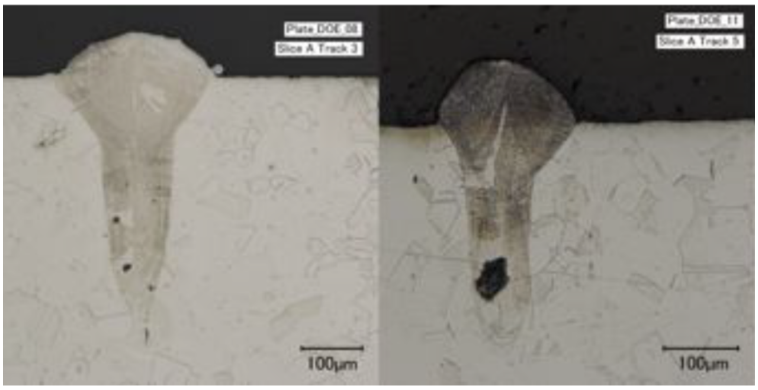


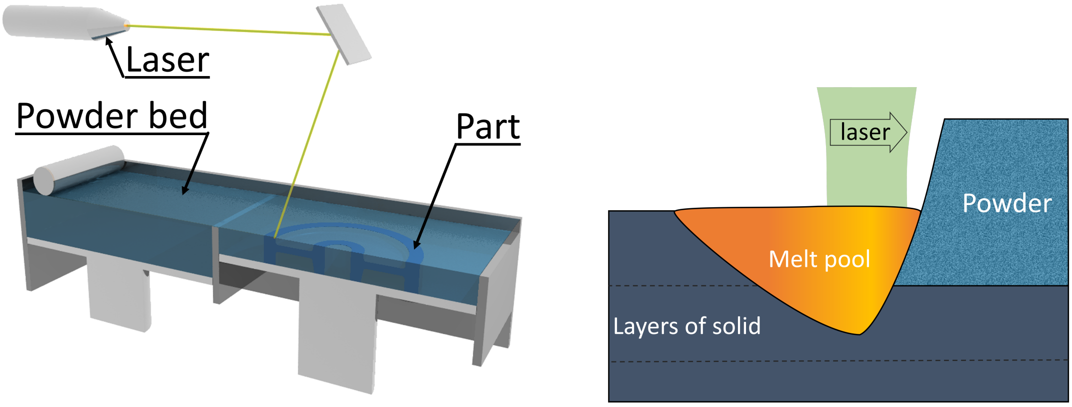
FIGURE 1.4: Porosity at the root of the weld track, typically caused by a keyhole instability trapping some gas. Taken from King et al. [24].

1.2.3 Laser Powder Bed Fusion (L-PBF)

Most Additive Manufacturing (AM) processes for metals and ceramics fall under one of these categories:

1. Directed Energy Deposition (DED)
2. Powder Bed Fusion (PBF)
   (a) Laser Powder Bed Fusion (L-PBF) or Selective Laser Melting (SLM)
   (b) Electron Beam Melting (EBM)

Many other processes exist that do not utilize phase change (e.g. sintering, curing, glueing), so they are not within the scope of this work.



(a) Schematic of a L-PBF machine.

(b) Schematic of L-PBF at the meso scale (adapted from [26]).

FIGURE 1.5: The L-PBF process.

In this work, literature regarding the Laser Powder Bed Fusion (L-PBF) processes is taken into consideration due to the process' similarity to laser welding. In L-PBF processes, a powder layer is spread on the substrate, which is then selectively melted by a laser beam, to form one layer of new material. Once one layer is complete, the next layer of powder is applied and a new layer is fused to the existing one. In this fashion, a part is built layer-by-layer. Fig. 1.5a. shows a schematic of a machine used for this process.

Some recent developments in the field of L-PBF are useful in the context of this work, where the most complex tests show laser welding. Due to the lack of a powder layer, laser welding is less complex than L-PBF, but the the main characteristics remain the same. Therefore, works on the modeling of L-PBF are included in this chapter.

Additional to the aspects of interest with regard to welding, L-PBF processes can have other characteristics related to the presence of the powder and the layer-by-layer building. Further development of the method presented here may help investigate these characteristics:

- surface quality,
- powder denudation,
- lack of fusion defects,
- additional sources of porosity, and
- balling.

A bad **surface quality** is a common problem in many AM processes [27]. A bad surface quality may require extra manufacturing steps (e.g. sanding, milling, shot peening etc.), where the surface is improved to meet the requirements. Each additional manufacturing step increases the cost and should therefore be avoided by achieving the desired surface quality directly from the AM process. For dynamically loaded parts, the surface quality may be of particular importance, to reduce the number of potential crack initiation sites [28].

Powder denudation and **spattering** can be two manifestations of the same phenomenon, where powder is blown away by some form of gas flow over the powder surface (see fig. 1.6). Denudation describes the deficiency of powder left behind and spattering refers to the uncontrolled projection of the powder itself. The entraining gas may be surrounding shielding gas or evaporated substrate material, where the latter has been found to be the main cause for powder denudation and powder spattering (see e.g. Mayi et al. [29]). This phenomenon causes an undesired local redistribution of powder that can then cause some of the following defects.

Lack of fusion defects describe an incomplete bonding between two layers or two hatches of melt tracks in AM processes. It presents a weak spot and a potential crack initiation site. While lack of fusion defects can also occur in welding, more literature is available in the context of L-PBF. There, the use of powder combined with the large number of overlapping welding beads makes a lack of fusion defect likely [31]. A local lack of powder or a locally insufficient melting of the powder can cause this defect. Spatter can, for example, disrupt the application of the next powder layer (fig. 1.7a), causing a deficiency of material that becomes a lack of fusion defect (fig. 1.7b).

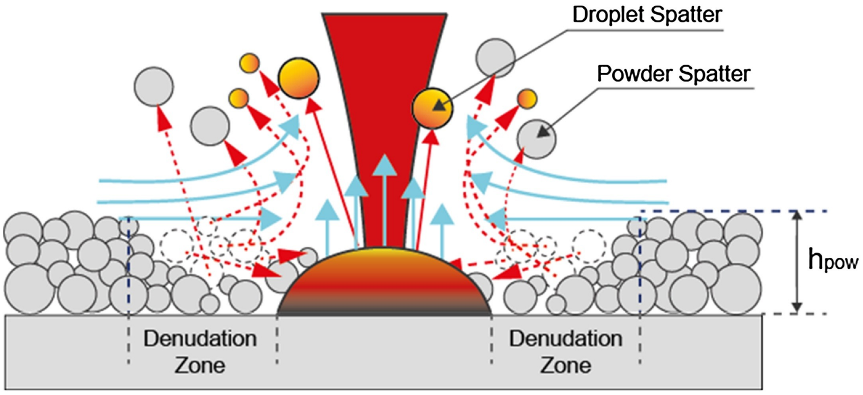
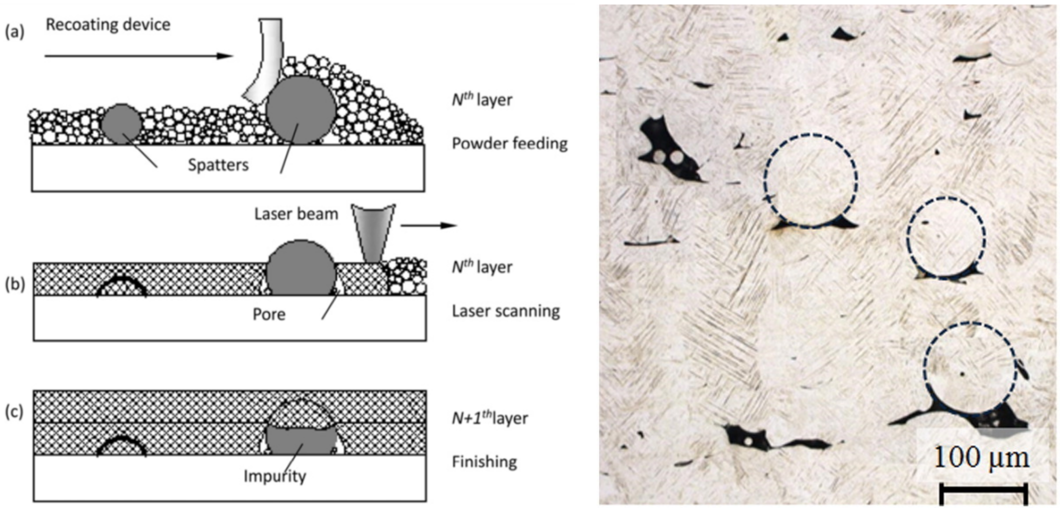


FIGURE 1.6: Spatter can be caused by liquid droplets ejected from the melt pool or by powder particles entrained by the atmopheric gas flow (blue arrows). Powder denudation can result around the melt track. Taken from Wischeropp et al. [30].



(a) Schematic of how spatter particles can lead to defects. Taken from Wang et al. [33].

(b) Porosity caused by spatter (dashed circles are spattered particles). Taken from Kasperovich et al. [34].

FIGURE 1.7: Defects in laser powder bed fusion (L-PBF) additive manufacturing.

There are three main mechanisms that lead to **porosity** in L-PBF processes according to DebRoy et al. [32]. The first one is the aforementioned keyhole instability that occurs also in laser welding. Second, the powder used in L-PBF processes may unintentionally contain gasses from the powder production process. The entrapped gas may not escape from the melt pool before solidification. Third, lack of fusion defects may create voids in the finished part.

Balling occurs, when the surface tension contracts the melt pool into a line of spheres before solidification, referred to as Plateau-Rayleigh instability. This is caused by badly chosen process parameters. In the example in fig. 1.8, the scanning speed is increased and balling becomes more severe [32]. Balling leads to lack of fusion defects, porosity and bad surface quality.

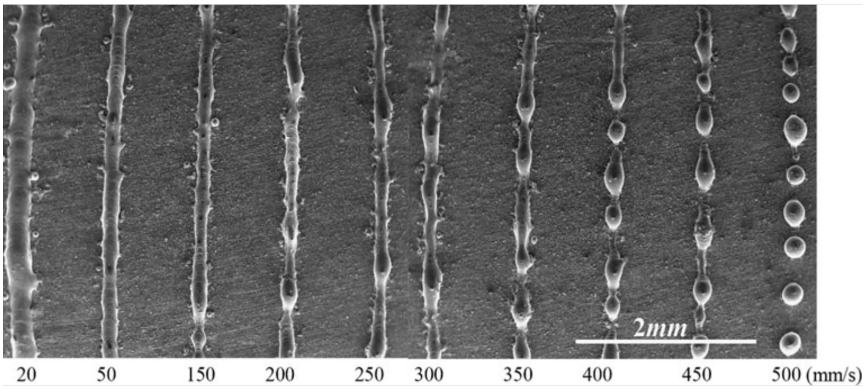


FIGURE 1.8: Balling occurring at too high scanning speed. Taken from Li et al. [35].

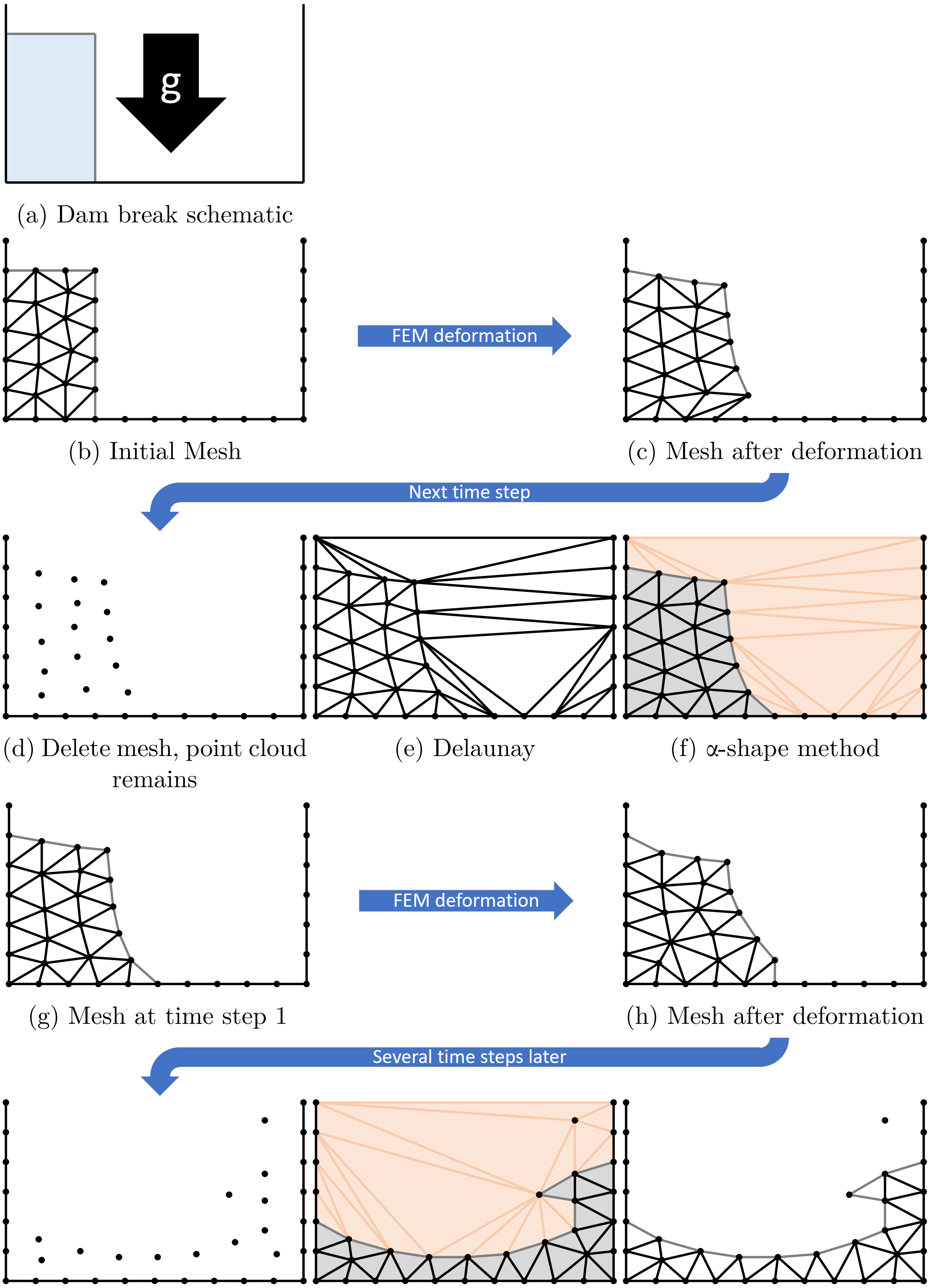
1.3 Proposed solution

Simulation methods for problems with concentrated heat sources and localized phase change at the meso-scale need to model both solid and liquid behavior simultaneously, which is what this work is attempting to accomplish.

The difficulty lies in the a priori unknown contour of the evolving melt pool. As a consequence, solid region and liquid region cannot be well-defined a priori. This makes coupling of classic fluid mechanics solvers and solid mechanics solvers difficult. In many examples in the literature coupling is avoided and only a fluid solver is used, while the solid behavior is only approximated. Such approaches lack the capability to model the residual stresses that occur as a direct consequence of the solidification and the temperature evolution around the melt pool. On the contrary, classic solid mechanics approaches neglect or approximate the convective flows in the melt pool. While residual stresses can be determined with such a method, the temperature history and the melt pool evolution may not be correctly captured, calling into question the accuracy of said residual stresses.

In this work, we propose a unified material model combined with a simulation method that can simulate both fluids and solids, called the Particle Finite Element Method (PFEM). With such a simulation method, the complex interactions of fluid, solid and heat can be captured, allowing to gain a better understanding of the process. With a better understanding of the process, the numerous process parameters can be better adjusted to avoid the issues described in sections 1.2.2 and 1.2.3.

The Particle Finite Element Method (original publication by Idelsohn et al. [37]) is based on the robust and well-known Lagrangian Finite Element Method (FEM). The nodes of the FEM mesh act as particles in the PFEM, because they are free to move over large distances. While this would normally cause the FEM mesh to distort, in the PFEM a frequent and automatic remeshing is employed that renews the mesh connectivity, restoring a good mesh quality (fig. 1.9). All solution variables (e.g. velocity, pressure, temperature, ...) are stored at the nodes and are therefore unaffected by the remeshing. The dam



(a) Dam break schematic

(b) Initial Mesh

(c) Mesh after deformation

(d) Delete mesh, point cloud remains

(e) Delaunay

(f) α-shape method

(g) Mesh at time step 1

(h) Mesh after deformation

(i) Point cloud

(j) Delaunay + α-shape

(k) Mesh at a later time with closing wave and splashing

FIGURE 1.9: Example schematic of PFEM for several time steps of a dam break test case. One time step: 1. remeshing (d - f), 2. solving the equations using the FEM (g - h). Remeshing: 1. delete mesh, 2. Delaunay triangulation, 3. α-shape technique ([36]). The Delaunay triangulation produces a convex hull with many unwanted triangles (e). These triangles (salmon) are detected and deleted by α-shape (f), leaving the discretized domain (light grey) behind with the free surface (dark grey lines). Note the changing connectivity, while the mesh quality is maintained.

break problem is used to exemplify the principle of the PFEM, as described above. The details of the remeshing will be discussed in section 2.7.

In essence, the PFEM combines the robustness of the FEM and the flexibility of a particle method [38]. This allows the PFEM to be used for fluid dynamics (see e.g. [39–42]) and solid mechanics (see e.g. [43–45]).

Several publications even demonstrate a unified fluid-solid formulation implemented in PFEM. **Idelsohn** et al. [46] proposed a similar approach as the one in this work. The main difference is the large deformation formulation for the solid that they included and some finer points (e.g. a different stabilization, no handling of pressure at fluid-solid interface described). To the best of the authors knowledge no publications followed that show any further developments of this approach by the original authors.

Franci et al. [47] implemented a very similar approach as Idelsohn et al. [46], that only appears to differ in some details (e.g. yet another stabilization, handling of pressure at fluid-solid interface, but different than in this work). They added energy conservation and even simple, but effective phase change capabilities. Franci et al. [48] and Franci's doctoral thesis [49] then demonstrate this powerful simulation method on a nuclear core melt accident test case. The publications show the hot and liquid corium melting its way through its steel confinement.

1.4 State of the art

The proposed approach is put in contrast with existing methods for simulating welding or additive manufacturing.

1.4.1 Overview of relevant computational methods

To put into perspective the method used in this work, the particle finite element method (PFEM), a short summary of other well known or similar methods is given below. To help categorize the different simulation methods, some important characteristics of simulation methods are:

- The use of a mesh or not,
- based on particles or not,
- the use of the Eulerian, Lagrangian or Arbitrary Lagrangian Eulerian (ALE) description of motion,
- the choice of solution variables.

A **mesh** or grid is a collection of points in space that are interconnected with their respective neighbors. In numerical methods it is commonly used to systematically subdivide a large, complex geometry into a set of simple geometric entities like polygons (2D) or polyhedra (3D), on which the solution can be calculated more efficiently by using some sort of systematic pattern. **Meshless methods**, as defined by Idelsohn et al. [50], use local approximation functions that only depend on the nodes and where the connectivity between nodes may be short-lived.

There appears to be no precise definition of **particle methods** in the literature, but the idea is intuitive. The material in the simulation is represented by independent particles that interact. In many cases this is because the material in question is comprised of particles in reality, but also continuous media can be approximated by particles. Note that particle methods are not necessarily meshless and vice-versa.

The **Eulerian** description of motion views physical quantities (e.g. velocity, pressure, temperature) from a stationary observer, as these quantities change over time in space. The space itself is fixed and movement is described as a change in position within this space. This has important implications in the description of some physical quantities that are related to acceleration or that change as a function of space. The mathematical manifestation of this is the appearance of a convective term in the equations of conservation of a physical quantity. In the context of comparing numerical simulation methods, Eulerian methods that use a computational grid will have a stationary grid and the material moves through the grid from cell to cell. Another implication is that any moving boundary or interface within the computation domain (as opposed to one on the boundary of the domain) will need to be tracked in some way.

The **Lagrangian** description of motion views physical quantities from the point of view of the moving particle itself. Space around the particle is considered fixed to the particle itself. Within this space the acceleration of a particle becomes simpler, as it is only the rate of change of the particles velocity. Conservation equations are generally defined for any given point on the body and are therefore also expressed in the Lagrangian frame of reference. Therefore, no convective term is needed. In the context of simulation methods, a computational grid deforms as the meshed body deforms. Any point on the body remains in the same position in the same cell at all times. As opposed to boundaries and interfaces in Eulerian methods, the ones in Lagrangian methods therefore remain in the same position relative to the moving mesh and no tracking is required. A significant drawback of the deforming mesh is that an overly deformed or distorted mesh becomes inaccurate to the point of breaking the simulation [51].

The **Arbitrary Lagrangian Eulerian (ALE)** description of motion combines both Lagrangian and Eulerian approach in an attempt to combine the advantages of both [52]. The key idea is that a third frame of reference is introduced and it is not required to coincide with either the Eulerian or the Lagrangian frame of reference. The grid is defined on this reference domain, which means that the particles move relative to the grid, but the grid itself moves relative to the fixed global frame of reference. With some clever choice of grid displacement and deformation, the mesh distortions can be minimized.

Finally, the **choice of nodal solution variables** is an interesting characteristic of a numerical method. Typical choices for the momentum conservation equation are a displacement or velocity formulation, sometimes a mixed velocity-pressure formulation. For the energy conservation the usual choice is either the temperature as the solution variable or the enthalpy.

These characteristics help get a quick overview, but the critical aspects are sometimes found in the fine details. Each method comes with its own strengths and limitations and typical fields of application. The following methods are selected to be briefly summarized for the reader to better understand the justification for the choice of the method in this work.

Finite Difference Method (FDM)

The Finite Difference Method utilizes a regular grid of nodes to discretize spatially and approximate the solution to a Partial Differential Equation (PDE) in its differential form. The derivatives of the PDE are approximated at the points by finite differences, which then yields a system of algebraic equations that can be solved numerically.

The most severe limitation of the FDM is the requirement for a regular grid, which makes the method difficult to use for complex geometries, such as the ones that appear in welding and AM.

Finite Volume Method (FVM)

The Finite Volume Method is typically an Eulerian method utilized in fluid dynamics. It also uses a grid, but the method is focused on the cells, which act as control volumes. In each control volume the conservation equations can be written in their integral form directly and equal the integral of the surface fluxes at the cells boundaries. The FVM utilizes cell-averaged function values instead of the function values at the nodes to represent a field of a physical quantity [53].

The main drawback of the Eulerian FVM in this context is that with a fixed mesh, the evolving boundaries and interfaces need to be tracked, which adds another layer of complexity. According to the review by Cardiff and Demirdžić [54], implementing solid mechanics in the FVM is possible, albeit not very common.

Finite Element Method (FEM)

The Finite Element Method is typically a Lagrangian method and a common choice for solid mechanics, while many of the examples shown below indeed use an Eulerian FEM for fluid dynamics. The subdomains in the mesh or grid are referred to as finite elements and they are again of simple shape, polygons (2D) or polyhedra (3D). The FEM approximates the solution by a trial function, which is usually a linear combination of polynomials as basis functions (better known as shape functions) defined on each element and each associated with one node. The coefficients of these polynomials are the unknowns to be solved for, which are the nodal solution variables. Since nodes are shared between all adjacent elements, the solution can be continuous, depending on the choice of trial functions and element types. The FEM uses a weighted residual method, in many cases the so-called Bubnov-Galerkin method is chosen to obtain a system of equations that is then solved by a solution scheme. As part of the process, the governing equations, given in their local form (or strong form) are converted to the weak form to reduce the order of the derivatives that appear in the equations.

The main advantage of the FEM is its ability to handle complex material laws and complex geometries. But the Lagrangian framework also means that meshes can get distorted, if the deformation is too large.

Discontinuous Galerkin Method (DG)

The Discontinuous Galerkin method is similar to the classic FEM described above and it is especially well suited for discontinuities in the solution like shocks or cracks. The main difference is the discontinuity between elements, as each element's degrees of freedom (i.e. nodal solution variables) are unique. Elements are not directly connected with their neighbors, but indirectly though a numerical flux that can enforce a weak continuity. The numerical flux at the interface between two elements is computed from both elements' individual, internal fluxes. These fluxes themselves depend on the solution, so they appear in the final system of equations and are solved simultaneously with all the other contributions.

Since DG requires no global continuity at the nodes, higher order elements are easily used. Similarly, since DG requires no global continuity of the trial functions and because elements are locally conservative [55], the DG method allows easily for higher order shape functions, which can be very useful or even necessary. The order of both element and shape function can be freely adapted from element to element to meet the user's needs.

Discrete Element Method (DEM)

The Discrete Element Method models entire particles of any shape and their interactions through contact forces. While DEM is used to model granular flows or soil deformation, it can also approximate the flow of a continuous fluid, if the particles' length scale is much smaller than that of the flow features. In the context of additive manufacturing, DEM appears to be a common choice for composing a realistic powder bed for simulations of L-PBF that resolve the powder in a second step with another method (e.g. FEM, FVM), according to Cook and Murphy [13].

PFEM-2

The Particle Finite Element Method of the second generation utilizes particles that move freely though a fixed grid. Despite what its name suggests, it is quite different from the regular PFEM. In the Lagrangian PFEM, particles velocity is determined using FEM and then the particles are moved along their velocity vector for the duration of a time step. Sklar et al. [56] summarize

PFEM-2 as calculating the pressure and velocity field using the FEM on the mesh, determining the streamlines from the velocity field and moving the free particles along the streamlines instead of in a straight line along the velocity vector (as in other Lagrangian methods). After the convection of the particles, their properties (velocity, pressure, temperature) are projected back onto the fixed mesh nodes by interpolation and weighted averaging.

Smoothed Particle Hydrodynamics (SPH)

The Smoothed Particle Hydrodynamics is a Lagrangian meshless particle method normally used for fluid dynamics [57]. Individual particles with their own properties, interact with one another, when in close proximity. Particles are characterized by a compact support (kernel functions) with a radius of influence in which they interact with other particles or boundaries.

The main advantages are a simple implementation, great flexibility to model different material behaviors and that large deformations of free surfaces are efficiently treated. SPH is not limited by complex geometries and discontinuities (e.g. phase change) are no problem. The method is one of the oldest meshfree particle methods and is reaching maturity [57]. The major drawbacks are the difficulty enforcing some boundary conditions, several types of instabilities and the relatively high CPU cost.

Lattice Boltzmann Method (LBM)

The Lattice Boltzmann Method is a computational fluid dynamics technique that discretizes the domain into a regular grid (or lattice) and assumes that the fluid is represented by particles located at the grid nodes. The fluid's properties like viscosity, density and temperature are reflected in the way that all particles at one lattice node interact with one-another. These interactions are governed by so-called collision rules. A time step consists of two steps: the collision step and the streaming step. The collision step changes the particles' velocity vectors, following the collision rules that describe the fluid's behavior. During the streaming step, the particles are re-distributed to neighboring lattice nodes according to their velocity vector after the collisions.

The main advantage of the LBM is its computation speed and excellent parallelization properties. It is suitable for complex geometries and boundary conditions and for porous media [58]. According to Cook and Murphy [13], the only real disadvantage of this method is its relative immaturity, along with some other minor difficulties.

1.4.2 Previous work simulating welding and additive manufacturing at the meso scale

The literature contains many works that aim at providing a better understanding of welding and additive manufacturing processes at the meso-scale. The majority falls in one of two categories: focused on fluid dynamics or focused on solid mechanics. In most cases, the typical choice for a fluid-focused approach is to use an Eulerian FVM-based method and for a solid-focused approach the typical choice is a Lagrangian FEM-based method. However, this is not always the case and a few relevant examples are given in this section.

A technique that is often used in the modeling of a phase change problem, where the simulation code is specialized on fluids is the use of a **flow resistance term** to approximate the solid's inability to flow. Such a flow resistance term in the momentum equation will inhibit the movement in the solid region to such a degree that the velocities there approach zero. In the mushy zone or semi-solid zone, the resistance grows slowly when $T \approx T_l$ and then very rapidly, when $T \to T_s$, where T_l and T_s are the liquidus and solidus temperatures, respectively. This flow resistance term is calculated using the **Carman-Kozeny equation** (eq. 2.49), which is normally used to model flow through a porous medium. Many authors therefore refer to the source term as **Darcy term** and to the methodology as **Porosity approach** (coined by Brent et al. [59]). We shall refer to this methodology throughout this work by mentioning the Carman-Kozeny equation. The equations, parameters and some commentary can be found in section 2.3.3. Crucially, this approach does not allow to predict residual stresses, because the medium remains a viscous fluid without elastic properties required for residual stresses to appear.

To track interfaces in an Eulerian method, where the mesh cannot deform to remain conformal with the free surface or interface contour, an interface tracking method must be employed. There are two commonly used options: Level Sets (LS) and Volume of Fluid (VoF).

The **Level Set** approach (Osher and Sethian [60]) defines an additional scalar field variable φ_{LS} that takes the value $\varphi_{LS} = 0$ at the interface Γ that is to be tracked. An individual φ_{LS} is defined for each independent tracked interface or boundary. The value of φ_{LS} throughout the domain represents a signed distance to the interface. In the context here, the material-gas interface, for example, is governed by the velocity field of the fluid near the surface and therefore, the LS is convected with the flow velocity. After convecting the entire field of φ_{LS}, a correction (so-called re-initialization) must take place to restore the Eikonal-condition, which states that $\|\nabla\varphi_{LS}\| = 1$.

The **Volume of Fluid** approach (Hirt and Nichols [61]) also defines a scalar field C which is bounded $0 \leq C \leq 1$. Assuming only two distinct phases, like gas and material, then $C = 1$ refers to a cell that contain only one of the phases and $C = 0$ refers to a cell that contains only the other phase. A value $0 < C < 1$ signifies a cell that contains a mix of both and therefore contains the interface. This scalar field is also convected in the domain, just as the level set. As opposed to the sharp interface that can be obtained by the level set, the VoF interface is smeared over at least the length scale of the cells at the interface.

Determining the position of the fluid-solid interface with the above methods is one aspect, while the application of the interface conditions (e.g. a pressure and stress jump) is another. Interfaces, just like boundaries, can be either mesh-conforming or use an immersed interface approach.

For **mesh-conforming interfaces** are simpler to implement but the interface can end up being either inaccurate or have a bad local convergence

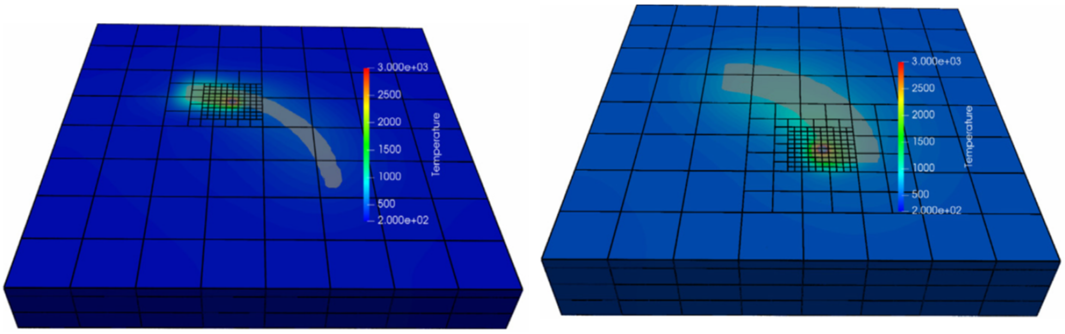
[62]. Lagrangian methods lend themselves to mesh-conforming boundaries and interfaces. In the context of the industrial applications mentioned above, a mesh-conforming interface could be easily established at the interface between dense material and surrounding gas, since there is no mixing and no transition. For the fluid-solid interface during phase change, the interface cannot generally remain mesh-conforming and either an approximation of the interface (smearing) or a local mesh modification to re-establish mesh-conformity (see e.g. X-MESH approach, Moës et al. [63]) are required.

The non-conforming methods, known as **immersed interface methods** (IIM), can achieve a high order accuracy in the vicinity of the interface, but are more complex to implement. A notable example is the work by F. Kummer's group at the TU Darmstadt (e.g. Rieckmann et al. [64]) that combine a DG method with level set and a cut-cell approach, one of several IIM. The advantage is that the sharp stress and pressure jump at the immersed interface can be achieved at the cut cells using the numerical fluxes already used by the DG method. The downside is the relive complexity of this approach and some new numerical issues that require special treatment (e.g. small cut cells).

Purely thermal approach using FEM

Kollmannsberger et al. [1] developed a FEM-based purely thermal simulation method aimed at analyzing the heat input by a laser heat source, with L-PBF applications in mind. The article focuses, along with some validation cases, on the meshing and solution strategy, where the thermal problem is solved on an efficiently refined mesh. The phase change is modeled according to the method described by Celentano et al. [65], where an enthalpy formulation is used and the isothermal phase transition is smoothened with a regularization, albeit a slightly different one than in [65]. An example from this work can be seen in fig 1.10. Note that this method covers the part scale, but is resolved all the way to the meso scale, where the laser intensity distribution is well-resolved. This multi-scale capability is a unique advantage of this approach.

Later extension by Kollmannsberger et al. [66] added more boundary conditions, more complex material models and a more realistic laser heat source model. They report that the melt pool shape, which is compared against experimental results, is not satisfactory at first. An anisotropic heat conductivity in the liquid melt pool was found to increase the accuracy to produce valid results.



(a) at time step 220

(b) at time step 1000

FIGURE 1.10: Purely thermal FEM simulation by Kollmannsberger et al. [1].

Thermo-fluid approach using FEM

Bayat et al. [2] use the commercial code *COMSOL Multiphysics*, an Eulerian FEM based approach, to investigate the laser-material interaction and predict the dendrite growth direction during solidification in a single track L-PBF example. The fixed computational domain is a simple cuboid containing a thick layer of solid at the bottom and a thin powder layer at the top. No gas is modeled and the free surface does not deform. A Newtonian fluid behavior is used to model all phases: liquid melt pool, solid and powder. The solid behavior is then approximated with the above mentioned Carman-Kozeny equation and the powder uses effective properties obtained with mixing laws (mixing metal and air properties). They do not explain if the powder layer contracts upon melting and what fills the empty space. Since only thermal properties are given for the powder, it appears as if the powder layer is essentially nothing but a solid with modified thermal properties, according to the mixing laws.

The metallurgical sub-model to predict the dendrite growth orientation is very simple. It determines the dendrite growth direction from the orientation of the temperature gradient in the mushy zone during the solidification process. The expected grain morphology (e.g. equiaxed, columnar etc.) is also determined using a very simple criterion.

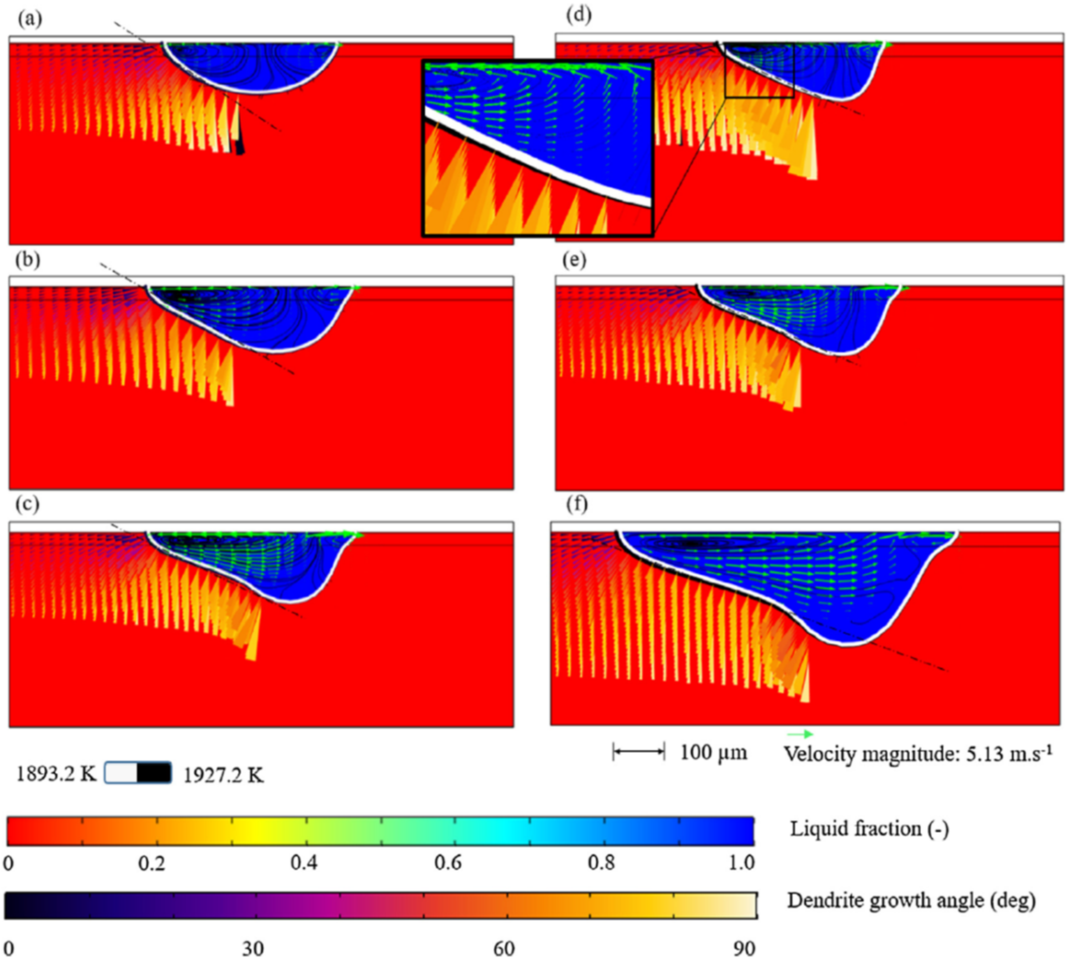


FIGURE 1.11: Plot of liquid fraction, velocity vectors and dendrite growth direction vectors by Bayat et al. [2].

The volumetric laser heat source is interesting, because it models the attenuation of the laser intensity as it penetrates deeper into the material, but the attenuation is not following the usual Beer-Lambert law, that is commonly used in this context (see Cook and Murphy [13]). Instead it appears that the authors have developed their own model which is tuned to meet expectations regarding width to depth ratio of the melt pool, based on observations published in the literature that the authors cite.

An interesting finding is the importance of the convective heat transfer in the melt pool. A first simplified simulation only considered heat conduction, which led to 20% higher peak temperatures and a larger melt pool by a factor 3 to 4. With the convection allowed, the authors compare different parameters for the laser heat source and their effect on melt pool size and grain orientation and morphology, where they find a good agreement with the literature. An example of the simulation results obtained is given in fig. 1.11.

Khairallah et al. [3] use the *ALE3D* code to resolve the powder particles and study the effects of recoil pressure, the Marangoni effect, evaporative and radiative cooling on the melt track. Phenomena like splatter and denudation are captured and the particular importance of the recoil pressure in these phenomena is highlighted. The powder layer is generated using the Discrete Element Method (DEM). Unfortunately, there is no detailed information regarding the ALE3D code other than the use of an ALE framework.

Khairallah et al. provide very impressive simulation results of a L-PBF process in keyhole mode (fig. 1.12). They explain that they observed some of the typical L-PBF phenomena, like pores due to a collapsing keyhole, spatter and powder denudation and provide some suggestions how to avoid these issues.

As an example of more recent work with this simulation method, the same authors are later involved in a publication by Khairallah et al. [67], where very specific scenarios of spatter and pore formation are investigated. They use the aforementioned simulation method and compare with detailed in-situ experimental observations of these highly dynamic phenomena. From the combined experimental and simulation data, they derive guidelines for suppressing some spatter and pore formation mechanisms in PBF processes, such as specific scanning patterns or power variations (so-called power mapping) of the heat source, depending on the powder grain size and alloy used.

Mayi et al. [29] use the commercial code *COMSOL Multiphysics* to investigate the metal vapor plume during keyhole mode laser welding. The FEM code uses an ALE framework to maintain a conformal meshing, as the liquid-gas interface deforms. Both liquid melt pool and the surrounding gas are modeled as an incompressible Newtonian fluid with a velocity-pressure formulation, while the solid is approximated using the Carman-Kozeny equation. The heat equation is given in the temperature formulation and the latent heat of melting is taken into account through an equivalent heat capacity. The evaporative cooling is computed according to standard practice as outlined by Cook and Murphy [13]. From the evaporation mass flux, the recoil pressure is also obtained. The modeling of the mixed gas, comprised of atmosphere and metal vapor, is discussed in detail.

The approach is validated with some experiments and then used to analyze the powder denudation and spattering in a L-PBF test case. Most relevant to this work is the modeling of the melt pool forming a deep cavity (not a keyhole yet) under the recoil pressure during laser spot welding. Numerical and experimental data for melt pool dimensions are given for different laser intensities, as well as temperature distributions in the melt pool.

Thermo-fluid-solid approach using FEM

From the team of Prof. Bellet (CEMEF/2MS) at MINES Paris - PSL, **Chen** et al. [26, 68] and Chen's doctoral thesis [4] present a staggered fluid-solid approach based on the FEM for the purpose of simulating L-PBF. It includes an elasto-plastic solid phase, a Newtonian fluid phase and visco-plastic mushy zone and a powder phase. The powder is modeled as a continuum, based on the assumption that the ceramic powder used in the validation example is very

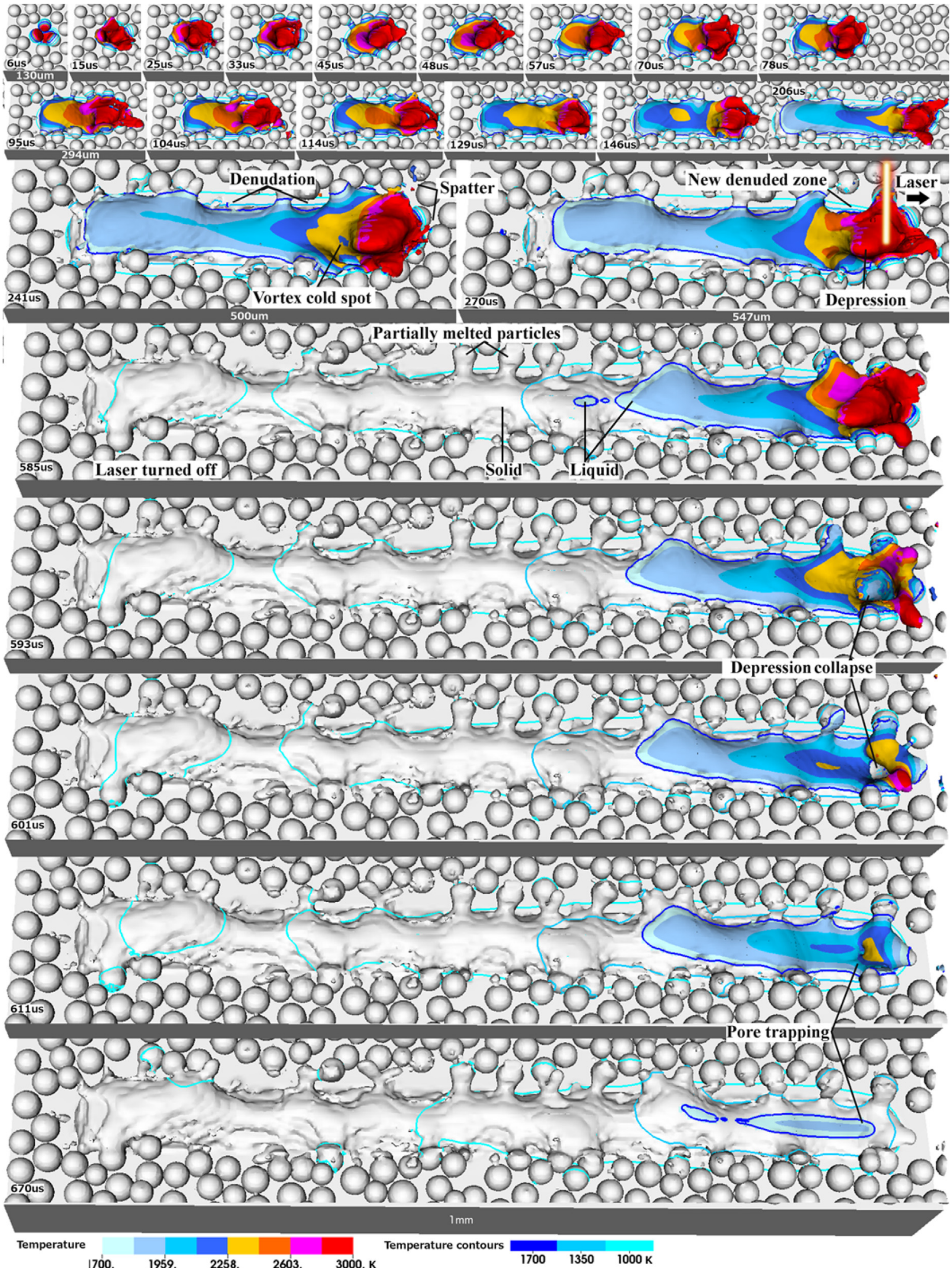


FIGURE 1.12: Plot of melt pool with keyhole and powder with temperature contours by Khairallah et al. [3].

fine. The mechanical problem uses a mixed velocity-pressure formulation and the thermal problem uses the enthalpy as solution variable.

Within a time step, the thermal problem is solved first, the fluid second and the solid third in a staggered scheme. All steps utilize the exact same domain and mesh. These steps are followed by the update of the nodal positions and a remeshing with adaptive mesh refinement, before moving on to the next time step. The interface between gas and material, as well as the interface between solid and non-solid are tracked using the Level Set (LS) method.

The key to the method lies in the special treatment of the fluid and the solid during the fluid and solid solving steps: during the fluid step the entire domain behaves like a Newtonian fluid and the solid region is given a very high viscosity, making it virtually stationary. During the solid step, the entire domain is elasto-visco-plastic and the fluid region is given a zero yield strength, making it very compliant. In this fashion, the fluid step obtains an accurate result for the fluid region and the solid step obtains an accurate result for the solid region. One interesting detail is that the thermal and fluid problems are solved using an Eulerian description of motion, while the solid uses a Lagrangian description of motion, but using the same mesh, which is convected and adaptively refined at each time step.

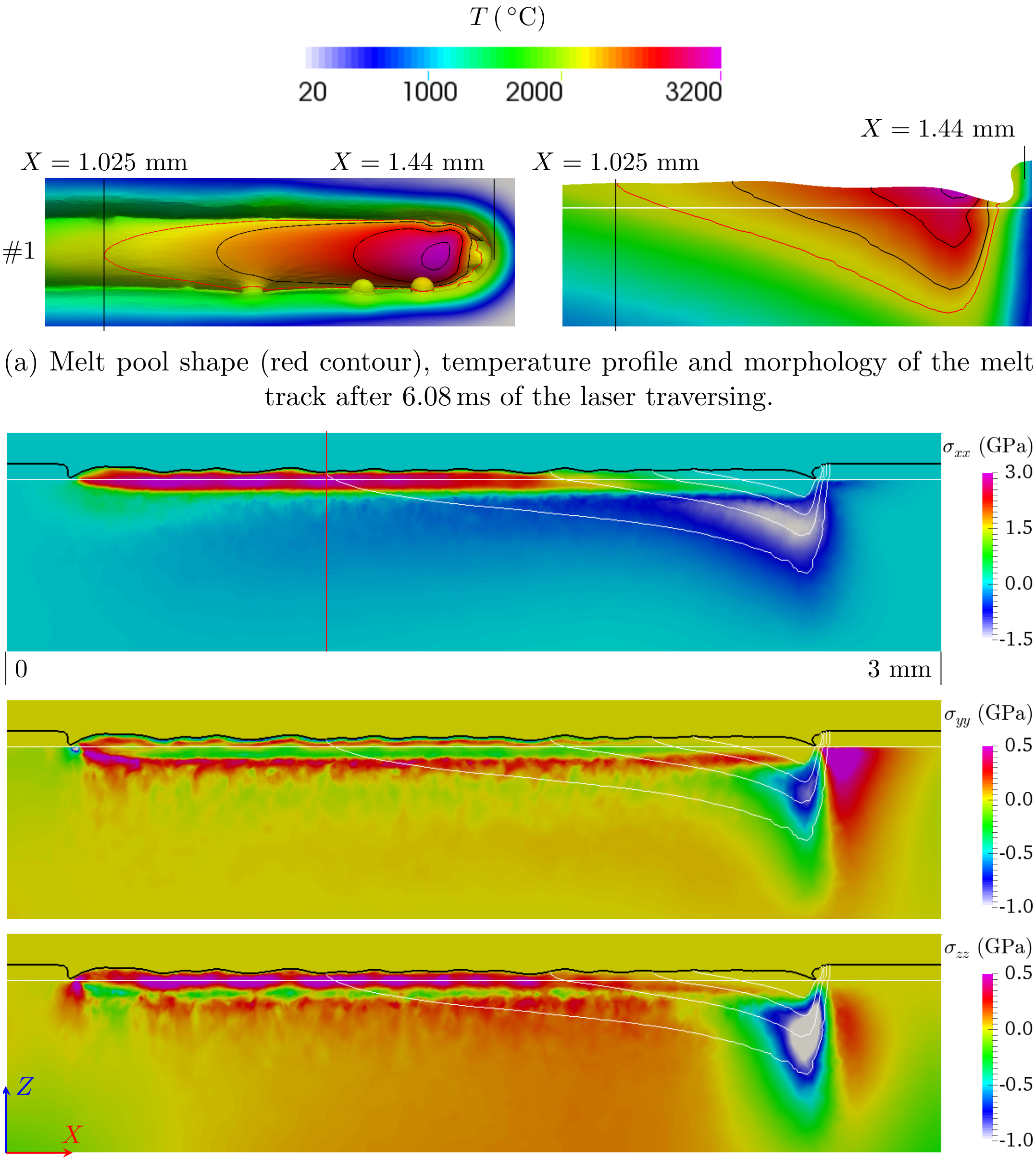
The interface tracking using the LS approach is used to determine the material behavior. Depending on the location of the interfaces relative to a given element, the element decides individually whether it exhibits gas, fluid, mushy or solid behavior or a mix thereof. For any element that is occupied by fractions of different materials, a Representative Volume Element (RVE) is essentially used where the material properties are combined in a weighted average. The same treatment is used throughout the powder phase, since a powder is never densely packed and is hence a fraction air and a fraction actual powder material.

For the equal order velocity-pressure formulation for the incompressible fluid, he employs the so-called Variational Multi-Scale (VMS) stabilization. For the elasto-visco-plastic solid a bubble function is introduced into the velocity field.

A particularly interesting feature is the modeling of the powder condensation upon melting, because the change in volume is both very large and nearly instantaneous, which leads to large velocities that would destabilize the solution process. Chen developed a special treatment to avoid the excessive velocities due to the shrinkage by allowing the powder to change phase more gradually.

Chen also describes an innovative algorithm to model the laser-material interaction. The laser penetrates deep into the material due to the use of ceramics, that is less absorbing than metal. Likewise, powder that is assumed to be loosely packed allows the beam to travel deeper into the powder layer, as it is only gradually absorbed. To model the deep penetration and gradual absorption, the well-known Beer-Lambert law is implemented. Since the interfaces and surfaces are evolving under the laser beam, the absorption (i.e. heat input into the material) is not a function of the z-direction (with its fixed origin), but of the depth into the material as measured from the deforming surface. Chen uses a local coordinate system attached to the laser in which a regular mesh is defined. This mesh overlaps with the FEM mesh and it detects where the FEM nodes and elements are located. This allows to apply the heat source from top to bottom on the regular grid, where the regular grid intersects the material represented by the FEM grid. The portion of the laser power not absorbed at the first layer of elements passes deeper to the next layer, where it is partially absorbed again and so on. This approach is very intuitive and presumably very close to the realistic behavior of a laser penetrating a material. Much care is taken to keep the cost of this algorithm minimal and to integrate it into the parallelization of the rest of the FEM code.

In accordance with other sources, the parameter studies by Chen et al. [68] reveal that the Marangoni effect plays the central role in the melt track morphology of L-PBF processes. Plots of the melt track morphology, melt pool shape, temperature distribution and stresses for many different laser settings are obtained. An example is given in fig. 1.13.



(a) Melt pool shape (red contour), temperature profile and morphology of the melt track after 6.08 ms of the laser traversing.

(b) Stresses obtained after 8.00 ms of the laser traversing. The white contour enclosing the largest area marks the melt pool.

FIGURE 1.13: Fully coupled fluid-solid-thermal simulation results by Chen [4].

From the same group, **Grange** et al. [69] simulate L-PBF to investigate laser-matter interaction and powder denudation and splatter using the method developed by Chen [4]. The aim is to obtain an empirical rule to tune the laser absorptivity to give results that agree well with experiments. This calibration rule should eliminate the need for expensive ray-tracing, when the surface is very uneven an the laser-matter interaction becomes very complex.

Grange et al. [69] cite an earlier work of theirs [70] in which they find that large melt pools tend to accumulate higher residual stresses due to different thermal expansion between melt pool and Heat-Affected Zone (HAZ). The remedy to large residual stresses is found to be narrow melt pools with strong overlap between hatches.

Zhang et al. [71] also continue the development of the method, but apply it to casting processes. A notable further development compared to [4] is the more refined mushy zone behavior. A critical temperature (found in physical experiments) is introduced which lies inside the solidification interval. The mushy material behaves like a weakened metal below this temperature and then begins transitioning smoothly to a Newtonian behavior above the critical temperature. Likewise, the mechanical behavior of the mushy zone is model more realistically during the fluid resolution step using the Carman-Kozeny equation with the inter-dendritic distance used to define the mushy zone's porosity. Additive manufacturing simulation approaches could benefit from a more realistic thermal and mechanical behavior of the mushy zone, which appears to be a topic that is glossed over oftentimes.

Thermo-fluid approach using FVM

Bayat et al. [5], who previously used *COMSOL Multiphysics* (see above), later used a FVM code called *Flow-3D* with a DEM-generated powder layer to investigate the formation of a keyhole with ray tracing. The Carman-Kozeny equation governs the kinematics of the solidification and a Volume of Fluid (VoF) approach is used to capture the phase transition front. They include surface tension, the Marangoni effect and recoil pressure.

A ray-tracing model for the laser-material interaction is implemented, where individual rays of laser beams are resolved and are only partially absorbed, every time a ray is reflected from a surface. The Fresnel absorption (absorptivity as a function of incidence angle) is taken into account.

The pore formation in the keyhole is investigated and compared to their own x-ray images of experimental L-PBF builds. The results presented are very detailed and show the evolution of a keyhole over time (e.g. fig. 1.14). Among other findings, they determine that cold spots inside the keyhole play an important role in the partial collapse of the keyhole due to a local lack of recoil pressure. The keyhole instability is clearly demonstrated and the inclusion of pores can be easily understood from the illustrations provided. The comparison with the ex-situ experimental data obtained from X-CT scans shows that pore size, morphology and distribution agrees well with the numerical results.

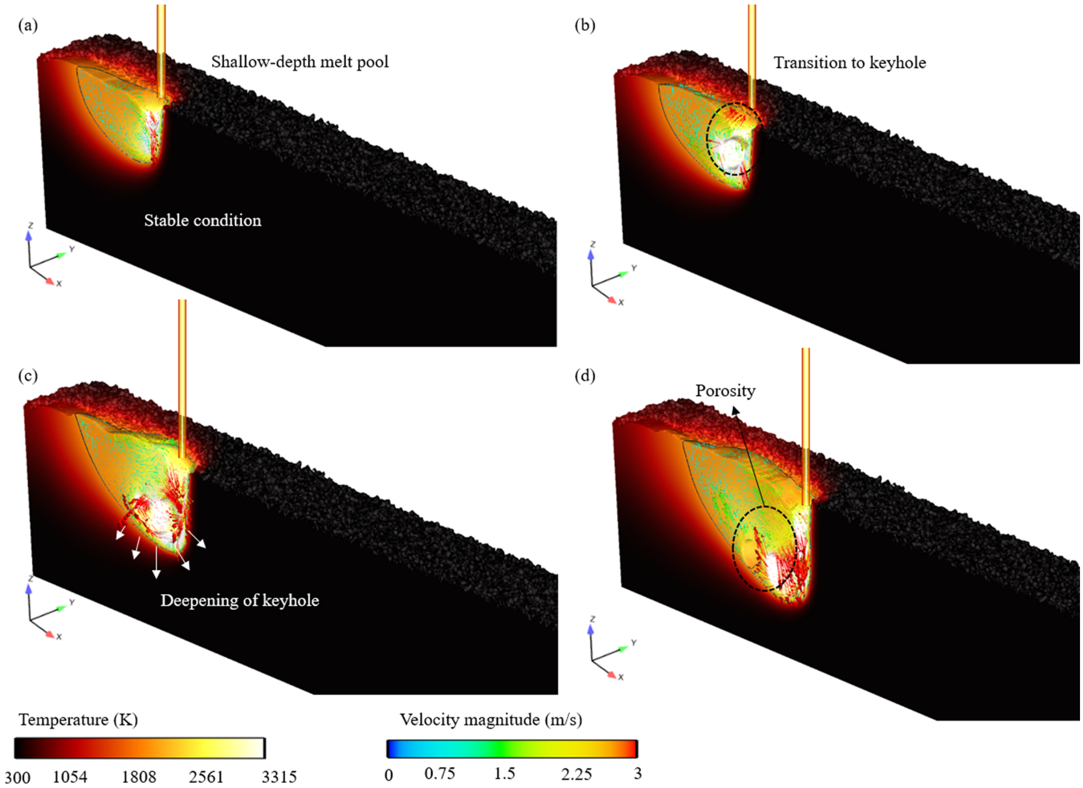


FIGURE 1.14: Plots showing the emergence of a keyhole and the occurence of keyhole porosity by Bayat et al. [5].

Pang et al. [6] study the time evolution of a keyhole during a laser welding process using an Eulerian Finite Volume approach with a level set to track the solid, liquid and gas phases. The modeled physics include recoil pressure, surface tension and Marangoni effect. Particular focus lies on the jet of evaporating material, the vapor plume, and its transient behavior.

The vapor plume is found to be very unstable, which appears to be related to the keyhole's own instability, where the walls of the keyhole keep evolving in a chaotic fashion (see fig. 1.15). The simulations of the keyhole and vapor plume evolution is compared with their own experiments, which help validate the numerical results.

They later turned their attention to Direct Energy Deposition (DED) AM processes, such as laser powered Wire Arc Additive Manufacturing (WAAM) [72], which is outside of the scope of this work and what follows after.

Panwisawas et al. [7, 73] developed an Eulerian FVM method in the open source package *OpenFOAM*. A solid, liquid and gas phase are modeled by an incompressible Newtonian fluid model, where the solid behavior is approximated with the Carman-Kozeny equation. The interface is tracked with the VoF method. Surface tension, including the Marangoni effect, recoil pressure are modeled at the melt pool surface, as well as radiation, convection and evaporative cooling. The laser is included as a volumetric heat source with a Gaussian distribution over the beam radius and a linear distribution along the depth into

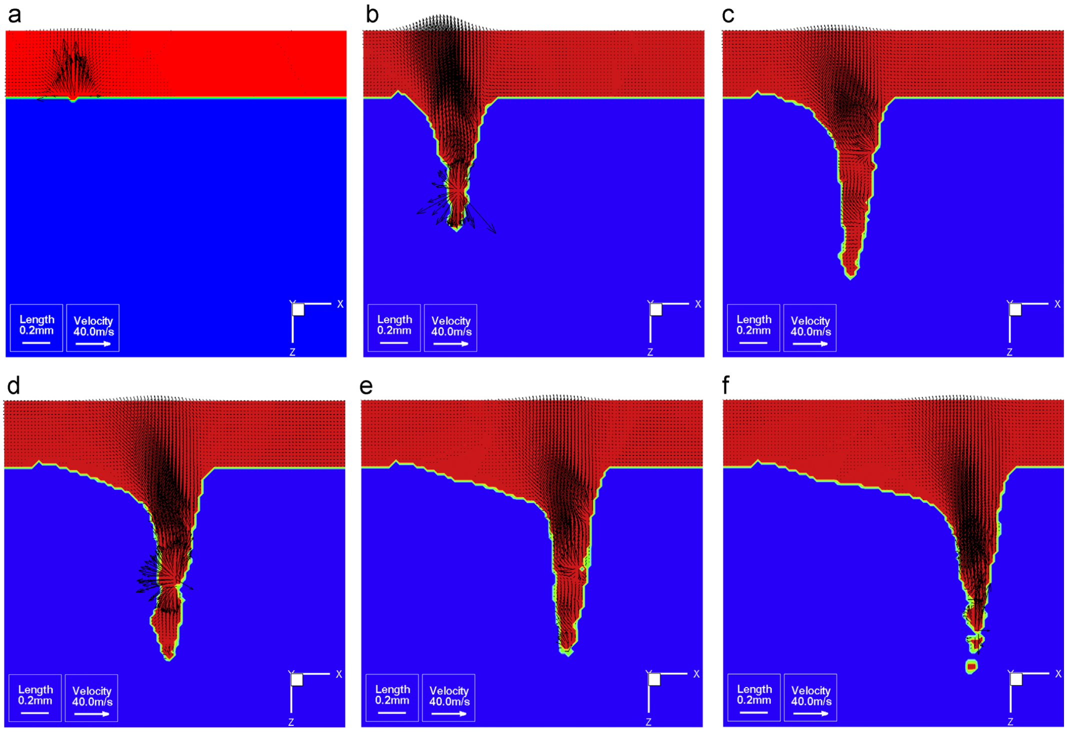


FIGURE 1.15: Plots of keyhole formation with metal vapor plume velocity vector field and keyhole porosity forming by Pang et al. [6].

the material.

The temperature history and melt pool evolution was furthermore exported to a 3D microstruture code called *Cellular Automaton – Finite Elements (CAFE)* to make a prediction of the microstructure. Experimental builds were analyzed using a Scanning Electron Microscope (SEM) that took pictures of isolated melt tracks, with individual metal particles visible, along with cut, polished and etched cross sections for the melt pool dimensions, porosity and grain structure.

Simulation results show single melt tracks with a detailed topology next to SEM images of actual melt tracks obtained with the same process parameters. The simulation results clearly reproduce the real world experiments and capture the dependence of the final result on the different settings, such as powder layer thickness, laser power, scanning speed etc. The authors come to the conclusion that simulations must reproduce the powder layer realistically in order to obtain predictive results, due to the strong dependence of the laser-material interaction on the powder layer morphology (thickness, grain size distribution, packing density). One porosity mechanism that the simulations captured well (porosity from lack of fusion) is analyzed and the likelihood of pores occurring is increased by a thicker powder layer and a higher scanning speed due to a more unstable flow in the melt pool. The authors also analyze the grain structure and compare the simulation results with experimental data, which agree well.

They continue to improve the method and publish articles that investigate the grain growth in more complex scenarios. The most recent example [74] is the

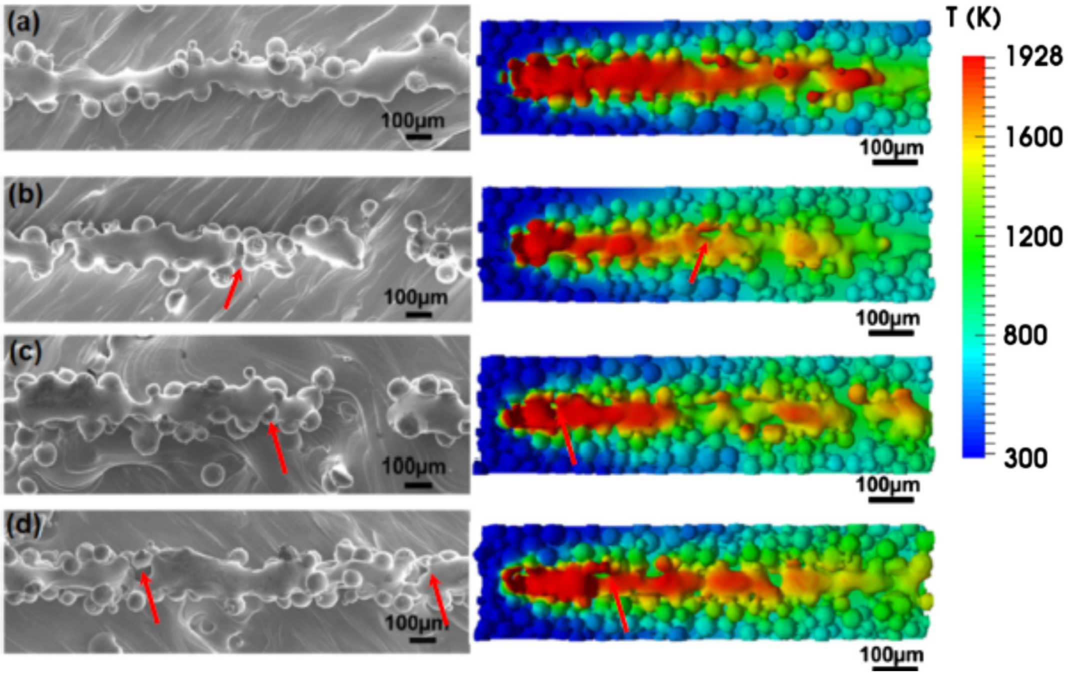


FIGURE 1.16: Melt track morphology for an increasing powder layer thickness (top to bottom), comparison of simulations and experiments by Panwisawas et al. [7].

simulation of a L-PBF process with multiple layers that re-melt and re-solidify parts of the material several times, while also mixing powders of different metals, forming an alloy in the process. They list suggestions how to better control the grain structure in the real world L-PBF process.

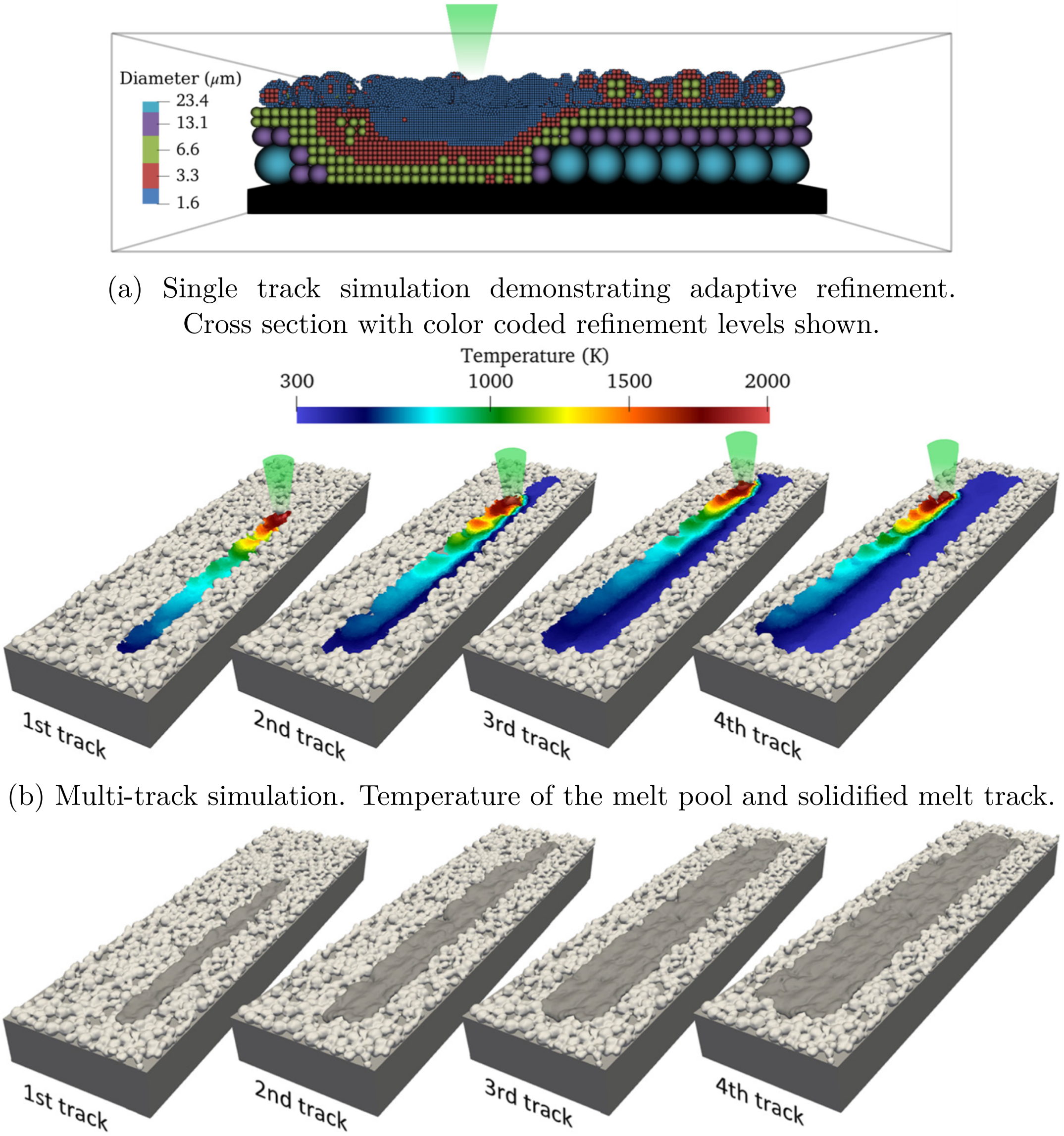
Other approaches

Körner et al. [58] use a 2D Lattice-Boltzmann Method (LBM) approach to simulate L-PBF processes. A powder bed is created by an algorithm (so-called rain model) that randomly deposits particles of a given mean size. A Gaussian distribution volumetric laser heat source is applied. Radiation and convection at the liquid surface are neglected and no evaporation occurs either. The gas-material interface is detected using a method similar to VoF.

In their analysis they investigate some process parameters and find that the packing density of the powder strongly affects the melt pool characteristics. The laser power over scanning speed (so-called line energy) is the next most important factor. At low power and large speed, balling occurs for example, while higher power and lower speeds lead to a more continuous melt track.

Lüthi et al. [8] have published a method for simulating the melt pool of a L-PBF process at the mesoscale using the Smoothed Particle Hydrodynamics (SPH). Their novel development uses an adaptive resolution scheme, where particles can be split into smaller particles or merged into bigger ones, if needed. This capability greatly improves the efficiency to resolve the powder layer and the flow in the melt pool, which is otherwise costly with SPH. The melt pool is modeled with surface tension, Marangoni effect and recoil pressure taken into account. The powder bed is generated using the DEM, where the physical powder particles are later represented by clusters of numerical particles. The solid region is modeled by a viscosity increased by a factor of ten.

After several verification and validation tests, they first present a single track L-PBF simulation, comparing the cost depending on the number of refinement levels allowed, demonstrating the necessity of the adaptive refinement. Fig. 1.17a shows a cross section of the track in the middle of the process, where the different refinement levels are color coded. They then go on to simulate a long multi-track L-PBF process with a high resolution, shown in figs. 1.17b and 1.17c.



(a) Single track simulation demonstrating adaptive refinement. Cross section with color coded refinement levels shown.

(b) Multi-track simulation. Temperature of the melt pool and solidified melt track.

(c) Multi-track simulation. Surface morphology of the melt tracks.

FIGURE 1.17: Melt pool simulation using SPH by Lüthi et al. [8].

At a conference (ESAFORM 2021, Liège) Arbaoui et al. [75] presented a laser spot welding test case using a Discontinuous Galerkin (DG) approach in 2D.

Much like this present work, the test case serves as a demonstration of the potential of the approach, while being an intermediate step towards the simulation of L-PBF. The location of the gas-material interface is determined by the Level Set (LS) method and then enforced using an immersed boundary with cells cut along the interface. This approach produces a sharp interface, as opposed to the simpler approach with a diffuse interface. Some focus in [75] lies on the modeling of the recoil pressure due to vaporization of the metal under the laser beam, a feature that this present work is lacking. The momentum sink term governs the solid. Unfortunately, there have been no newer publications to the author's best knowledge that show the progress in the development of this simulation approach.

1.5 Aim of this work

A lack of numerical simulation capabilities has been identified in the literature: a numerical simulation method that can solve the phase transition from fluid to solid or vice-versa in a single domain with a single solver. The fluid dynamics and solid mechanics must be correctly captured, as well as the heat transfer including the absorption and release of latent heat. A simulation method with such capabilities would be able to simulate certain industrial applications with phase change, where the solid mechanics, fluid dynamics and the heat transfer are strongly coupled. The prime example for such an application is additive manufacturing, where engineers are eager to gain better insight into the complex physics that take place at a very small time- and length-scale.

Additive manufacturing processes are very complex being 3-dimensional, spanning several length- and time-scales, with potentially complicated interaction of heat source and material, involving powders or filler material and undergoing large deformations and plasticity, to name some of the challenges. The starting point for this present work is the PFEM fluid dynamics code developed by Cerquaglia [39], leaving most of the above mentioned capabilities to be implemented, verified and validated. Since reaching the long term goal of simulating additive manufacturing with all its intricacies is barely possible within one doctorate, the focus of this work lies on the essence of the required capabilies: a unified thermo-fluid-solid simulation approach with phase change. With such a simulation method, a sensible industrial application to aim for as an intermediate step is spot welding. The final goal in this work is therefore to verify the unified fluid-solid approach and then demonstrate its capabilities on the example of spot welding.

This work is therefore split into four chapters after this introduction chapter. Chapter 2 describes in detail all the equations to be solved and the algorithms implemented in the simulation code. In chapter 3, the verification of all new developments is carried out on simple test cases, assuming that the pure fluid dynamics part is fully verified by Cerquaglia [39]. Since many different physics are combined in this work, great care was taken to take small steps of increasing complexity. Chapter 4 finally demonstrates the code's capabilities on more complex test cases. An isothermal Fluid-Structure Interaction (FSI) simulation comes first, followed by two different spot welding examples, where the full thermo-fluid-solid capabilities with phase change are utilized. In chapter 5, a summary of the findings and a critical analysis of the simulation method with recommended future developments are given.

Chapter 2

The method

The simulation method used in this work is based on the Particle Finite Element Method (PFEM), which essentially combines three concepts:

1. The Lagrangian Finite Element Method (FEM).
2. Frequent remeshing to avoid mesh distortion.
3. The solution stored at the nodes.

This combination produces a particle method classically utilized to simulate fluid flow with complex free surface deformation. The classic PFEM is then extended to have phase change capabilities and to model fluid and solid materials alike. To solve for fluids and solids simultaneously, a unified formulation is derived, such that the difference between fluid and solid is reduced to different material parameters. This allows to model Fluid-Structure Interaction (FSI) problems and even phase change problems in a single solver.

The conservation equations on which the method is based are described in the following section 2.2 "Conservation Equations". Models of the material behavior relevant for the simulation of processes like welding are then explained in section 2.3 "Material properties". The equations to solve are then completed by the unified formulation for fluids and solids in section 2.4 "Derivation of the unified formulation". These equations are finally discretized in time and space in section 2.5 "Finite element formulation". The numerical solution procedure, as well as the characteristic remeshing and data management that turns the classic FEM into the particle method PFEM are described in sections 2.6 "Solution process" and 2.7 "Mesh management".

2.1 Kinematic description

The motion of a material can be described from the perspective of a fixed outside observer or from the perspective of a particle of that material. These two distinct concepts are referred to as the Eulerian and Lagrangian description of motion, respectively. In this work, a Lagrangian description of the kinematics is chosen, as is the usual choice in the classic FEM for solid mechanics problems. Some authors use the FEM with an Eulerian description for similar applications (e.g. Chen [26]). To justify our choice, the most notable advantages of the Lagrangian description are:

- Keeping track of the deformation history of a particle is simpler, which becomes especially important when introducing plasticity in the future.
- Free surface and interface deformation requires no tracking, as it is automatically captured by the deforming domain.
- The non-linear convective term vanishes in the momentum conservation equation.

In the context of this work, the most notable downside of using the Lagrangian description over an Eulerian description is that mesh deformation occurs. If the mesh deformation is severe, the mesh quality degrades and with it the accuracy and stability of the method. This issue is circumvented here by the frequent remeshing mentioned above, which is aimed at restoring a good mesh quality (see section 2.7).

2.2 Conservation equations

This introduction is a short summary of the more detailed steps found in section 1.4 "Finite Element Methods for Flow Problems" by Donea and Huerta [52]. The interested reader will find a more detailed step-by-step explanation of the equations there. The method for the simulation of phase change problems in this work is based on the following three conservation equations:

1. The mass conservation equation.
2. The momentum conservation equation.
3. The energy conservation.

In each conservation equation, there is a quantity ϕ whose integral over a material volume Ω is conserved. In a Lagrangian framework, Ω crucially contains always the same set of particles and Ω moves with the particles and thus can deform. The material time derivative $\dot{\phi}$ of a quantity $\phi = \phi(x_i, t)$ defined in the Eulerian space is

$$\dot{\phi} = \frac{d\phi}{dt} = \frac{\partial \phi}{\partial t} + v_i \, \frac{\partial \phi}{\partial x_i} \tag{2.1}$$

where t is the time, v_i and x_i are the velocity and position in the i-th dimension. When the material time derivative of ϕ is integrated over volume Ω and the time derivative of that integral is taken, it must obey the well-known Reynolds transport theorem

$$\frac{d}{dt} \int\limits_{\Omega(t)} \phi \, dV = \int\limits_{\Omega(t)} \left[\frac{\partial \phi}{\partial t} + \phi \frac{\partial v_i}{\partial x_i} \right] dV \tag{2.2}$$

In the case of ϕ being a scalar quantity, the divergence theorem can be applied subsequently to obtain a useful variation of the Reynolds transport theorem:

$$\frac{d}{dt}\int\limits_{\Omega(t)} \phi\, dV = \int\limits_{\Omega(t)} \frac{\partial \phi}{\partial t}\, dV + \int\limits_{\Gamma(t)} \phi v_i n_i\, dS \tag{2.3}$$

which states that the change of the integral of ϕ is comprised of the accumulation of ϕ in Ω over time and the flux of ϕ through the surface Γ that encloses the volume Ω, where n_i is the i-component of the unit outward normal on Γ.

2.2.1 Mass conservation equation

The mass conservation equation (in fluid dynamics often referred to as continuity equation) states that the mass m of a closed control volume V, containing the same particles, remains constant and the change of mass dm/dt remains 0:

$$\frac{dm}{dt} = \frac{d}{dt}\underbrace{\int\limits_{\Omega(t)} \rho\, dV}_{m} = 0 \tag{2.4}$$

where m is the total mass of the volume and ρ is the density. Applying the Reynolds transport theorem eq. 2.2 to the mass conservation eq. 2.4, then injecting the material derivative 2.1 of the density and using the divergence theorem on the surface integral, the integral form of the local mass conservation is obtained

$$\int\limits_{\Omega(t)} \left(\dot{\rho} + \rho\frac{\partial v_i}{\partial x_i}\right) dV = 0 \tag{2.5}$$

which must hold for any choice of control volume V and therefore the integrand itself must be equal to zero:

$$\dot{\rho} + \rho\frac{\partial v_i}{\partial x_i} = 0 \tag{2.6}$$

Assuming that $\rho = \rho(p, T)$, where p is the pressure and T is the temperature, we can expand the time derivative of the density $\dot{\rho}$.

$$\dot{\rho} = \frac{\partial \rho}{\partial p}\dot{p} + \frac{\partial \rho}{\partial T}\dot{T} \tag{2.7}$$

The continuity equation obtains the form that is useful in this work by injecting eq. 2.7 into eq. 2.6 and dividing by ρ:

$$\frac{1}{\rho}\frac{\partial \rho}{\partial p}\dot{p} + \frac{1}{\rho}\frac{\partial \rho}{\partial T}\dot{T} + \frac{\partial v_i}{\partial x_i} = 0 \tag{2.8}$$

Expressions for $\partial\rho/\partial p$ and $\partial\rho/\partial T$ will come from the mechanical compressibility and the thermal expansion respectively, which are discussed in section 2.3.

2.2.2 Momentum conservation equation

The momentum equation is also known as momentum balance equation or equation of motion and it describes the linear momentum $\boldsymbol{M}$ of a volume of material. The equation's derivation begins with the rate of change of $\boldsymbol{M}$:

$$\frac{d(M_i)}{dt} = \frac{d}{dt}\int\limits_{\Omega(t)} \rho v_i \, dV \tag{2.9}$$

After using eqs. 2.1, 2.2 and 2.6 the expression becomes

$$\frac{d}{dt}\int\limits_{\Omega(t)} \rho v_i \, dV = \int\limits_{\Omega(t)} \rho \dot{v}_i \, dV \tag{2.10}$$

The momentum is conserved, when obeying Newton's first law, where the momentum changes at the rate of the sum of the forces acting on the material volume. Following Cauchy's postulate, we assume these forces are comprised of body forces $\boldsymbol{b}$ in $\mathrm{N\,kg^{-1}}$ and surface forces $\boldsymbol{t}$ in $\mathrm{N\,m^{-2}}$:

$$\int\limits_{\Omega(t)} \rho \dot{v}_i \, dV = \int\limits_{\Omega(t)} \rho b_i \, dV + \int\limits_{\Gamma(t)} t_i \, dS \tag{2.11}$$

where we assume the existence of the Cauchy stress tensor $\boldsymbol{\sigma}$, such that $t_i = \sigma_{ij} n_j$, with $\boldsymbol{n}$ being the unit outward normal on Γ. The momentum equation can then be written as

$$\int\limits_{\Omega(t)} \rho \dot{v}_i \, dV = \int\limits_{\Omega(t)} \rho b_i \, dV + \int\limits_{\Gamma(t)} \sigma_{ij} n_j \, dS \tag{2.12}$$

After applying the divergence theorem and rearranging, the momentum conservation for volume Ω becomes

$$\int\limits_{\Omega(t)} \left(\rho \dot{v}_i - \frac{\partial \sigma_{ij}}{\partial x_j} - \rho b_i \right) dV = 0 \tag{2.13}$$

As was the case before, the equation is independent of the choice of V and the integrand must therefore fulfill

$$\rho \dot{v}_i - \frac{\partial \sigma_{ij}}{\partial x_j} - \rho b_i = 0 \tag{2.14}$$

The Dirichlet and Neumann boundary conditions for the momentum equation read

$$v_i(x_i, t) = \bar{v}_i(x_i, t) \text{ on } \Gamma_v(t) \tag{2.15}$$

$$\sigma_{ij}(x_i, t) n_j = \bar{t}_i(x_i, t) \text{ on } \Gamma_t(t) \tag{2.16}$$

where $\bar{v}_i$ is the imposed velocity component and $\bar{t}_i$ the imposed surface traction component in i-direction. $\Gamma_v(t)$ and $\Gamma_t(t)$ are parts of the (evolving) boundary $\Gamma(t)$, where either velocity or surface traction are imposed, respectively.

Splitting up the imposed surface traction $\bar{t}_i$ in 2.16 into normal and tangential components is straightforward in 2D:

$$\bar{t}_i = \bar{t}_n n_i + \bar{t}_t t_i \tag{2.17}$$

n_i and t_i are the normal and tangent vector component in i-direction on the boundary. $\bar{t}_n$ and $\bar{t}_t$ are the surface traction components normal and tangential to the boundary. These are given by surface tension and surrounding pressure, as is discussed in section 2.3.

2.2.3 Energy conservation equation

Energy can take many forms, but for the simulation of solid mechanics and fluid dynamics, two forms of energy are useful to take into account: internal energy E_i and kinetic energy E_k. Their respective rates of change are

$$\frac{dE_i}{dt} = \frac{d}{dt}\int\limits_{\Omega(t)} \rho e \, dV \tag{2.18}$$

$$\frac{dE_k}{dt} = \frac{d}{dt}\int\limits_{\Omega(t)} \frac{1}{2}\rho v_i v_i \, dV \tag{2.19}$$

where e is the internal energy density. As before, applying eqs. 2.1, 2.2 and 2.6 we obtain the sum of the rates of change in a more convenient form:

$$\frac{d}{dt}(E_i + E_k) = \frac{d}{dt}\int\limits_{\Omega(t)} \rho\left(e + \frac{1}{2}v_i v_i\right) dV = \int\limits_{\Omega(t)} \rho\left(\dot{e} + \frac{1}{2}\dot{v}_i v_i\right) dV \tag{2.20}$$

According to the first law of thermodynamics, the change in energy in a closed volume must be equal to the heat Q and work W added to the volume.

$$\int\limits_{\Omega(t)} \rho\left(\dot{e} + \frac{1}{2}\dot{v}_i v_i\right) dV = \underbrace{\int\limits_{\Omega(t)} \rho b_i v_i \, dV + \int\limits_{\Gamma(t)} t_i v_i \, dS}_{\dot{W}} + \underbrace{\int\limits_{\Omega(t)} R \, dV - \int\limits_{\Gamma(t)} q_i n_i \, dS}_{\dot{Q}} \tag{2.21}$$

where $\boldsymbol{q}$ is the heat flux vector with dimensions $\mathrm{W\,m^{-2}}$, R is the power density of a volumetric heat source in $\mathrm{W\,m^{-3}}$ and $\boldsymbol{n}$ is the unit outward normal on Γ. After applying the divergence theorem to the surface terms and rearranging we obtain

$$\int\limits_{\Omega(t)} \left[\rho\left(\dot{e} + \frac{1}{2}\dot{v}_i v_i\right) - \rho b_i v_i - \frac{\partial}{\partial x_j}(\sigma_{ij} v_i) - R + \frac{\partial q_i}{\partial x_i}\right] dV = 0 \tag{2.22}$$

By splitting the derivative in the external force power term and inserting the momentum conservation eq. 2.14, the energy conservation simplifies to

$$\int\limits_{\Omega(t)} \left(\rho \dot{e} - \sigma_{ij} \frac{\partial v_i}{\partial x_j} + \frac{\partial q_i}{\partial x_i} - R \right) dV = 0 \tag{2.23}$$

The stress power term is neglected at this stage because the solid does not undergo large deformations. Lastly, replacing $\dot{e} = c_p \dot{T}$, where c_p is the (isobaric) specific heat capacity[1] in $\mathrm{J\,kg^{-1}\,K^{-1}}$ and T is the absolute temperature in K, the so-called heat equation is obtained

$$\int\limits_{\Omega(t)} \left(\rho c_p \dot{T} + \frac{\partial q_i}{\partial x_i} - R \right) dV = 0 \tag{2.24}$$

Since it must be valid for any choice of the volume Ω, the integrand itself must satisfy

$$\rho c_p \dot{T} + \frac{\partial q_i}{\partial x_i} - R = 0 \tag{2.25}$$

The Dirichlet and Neumann boundary conditions for the heat equation are

$$T(x_i, t) = \bar{T}(x_i, t) \text{ on } \Gamma_T(t) \tag{2.26}$$

$$q_i(x_i, t) n_i = \bar{q}(x_i, t) \text{ on } \Gamma_q(t) \tag{2.27}$$

where $\bar{T}$ is the imposed temperature and $\bar{q}$ the imposed surface heat flux. $\Gamma_T(t)$ and $\Gamma_q(t)$ are parts of the (evolving) boundary $\Gamma(t)$, where either temperature or surface heat flux are imposed, respectively.

2.3 Material properties, boundary conditions and sources

Material properties to be included in the model are selected based on the requirements of the application to phase change at the meso-scale. Only physics that are strictly necessary are included, so that the model is kept as simple as possible. This section explains the choices that were made, along with a justification.

2.3.1 Thermal

Welding or additive manufacturing are characterized by a concentrated heat source that causes steep temperature gradients and localized melting. The heat input is in fact the driving force in this system that leads to a wide spectrum of effects. Heat transfer therefore plays a central role in such a system and is governed by the following mechanisms included in the model:

- conduction in the fluid and solid,

[1] Only gases have significantly different isobaric and isochoric heat capacities (c_p and c_V, respectively). For solids and fluids, as used in this work, they can be considered equal.

- convection in the fluid,
- surface radiation at the interface of dense material with the atmosphere,
- convective heat transfer at the interface of dense material with the atmosphere,
- heat input into the dense material by the laser heat source.

Heat transfer mechanisms that are ignored in this work are

- convection and conduction in the atmosphere,
- temperature changes in the gaseous atmosphere due to compression/shocks or expansion,
- radiative interaction between surfaces,
- evaporative cooling (latent heat absorption upon evaporation),
- convection due to evaporation (hot mass leaving the dense material at the interface),
- viscous dissipation as a heat source.

Laser heat source

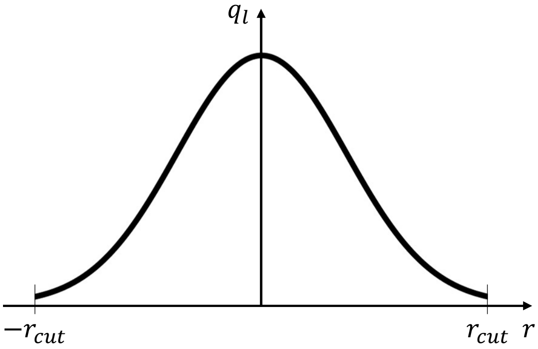


FIGURE 2.1: The Gaussian distribution of the laser intensity profile over radius r with a cut-off at r_{cut}.

The laser heat source is most commonly modeled either as surface heat flux (for optically opaque materials like metals) or as volume heat source (for optically transparent materials like some ceramics) with an intensity distribution in the depth direction. We only consider the surface heat flux option in this work.

First, a local cylindrical coordinate system e_z, e_r is introduced, where z is the coordinate along the laser center axis and r is the radial coordinate. In the radial direction, a common choice is a Gaussian distribution of the power density q_l [13]. The Gaussian distribution can be expressed by the following equation in the plane:

$$q_l = \frac{\alpha_l P_l}{\sqrt{2\pi} r_{std}} \exp\left(-\frac{1}{2}\frac{r^2}{r_{std}^2}\right) \qquad \forall\ r \text{ on } \Gamma_q \tag{2.28}$$

where $0 \leq \alpha_l \leq 1$ is the absorptivity of the targeted material for the incident laser, P_l is the nominal laser power in W. The parameter r_{std} is the radius related to a standard deviation, which governs the spread of the distribution. For an axisymmetric case the definition for the axisymmetric Gaussian distribution of the heat flux from [2] is used, as given in eq. 2.29:

$$q_l = \frac{\alpha_l P_l}{2\pi r_{std}^2} \exp\left(-\frac{1}{2}\frac{r^2}{r_{std}^2}\right) \qquad \forall\ r \text{ on } \Gamma_q \tag{2.29}$$

Such Gaussian distributions are sometimes used with a cut-off radius r_{cut}, such that $q_l = 0\ \forall\ r > r_{cut}$, as schematized in fig. 2.1. The cutoff radius should be significantly larger than r_{std} (for example $r_{cut} = 3r_{std}$) so that the nominal power P_l is mostly recovered.

In either case, when the laser central axis does not coincide with the surface normal, the laser's intensity profile must be projected onto the surface. This preserves the total heat input P_l, when integrating q_l over a curved surface. We define a deviation α between laser axis and unit outward surface normal as in figure 2.2. The intensity q_l is scaled down from full intensity at 180° deviation to 0 intensity at ±90° and we arrive at the following expression for the projected laser heat flux intensity q_{lp}

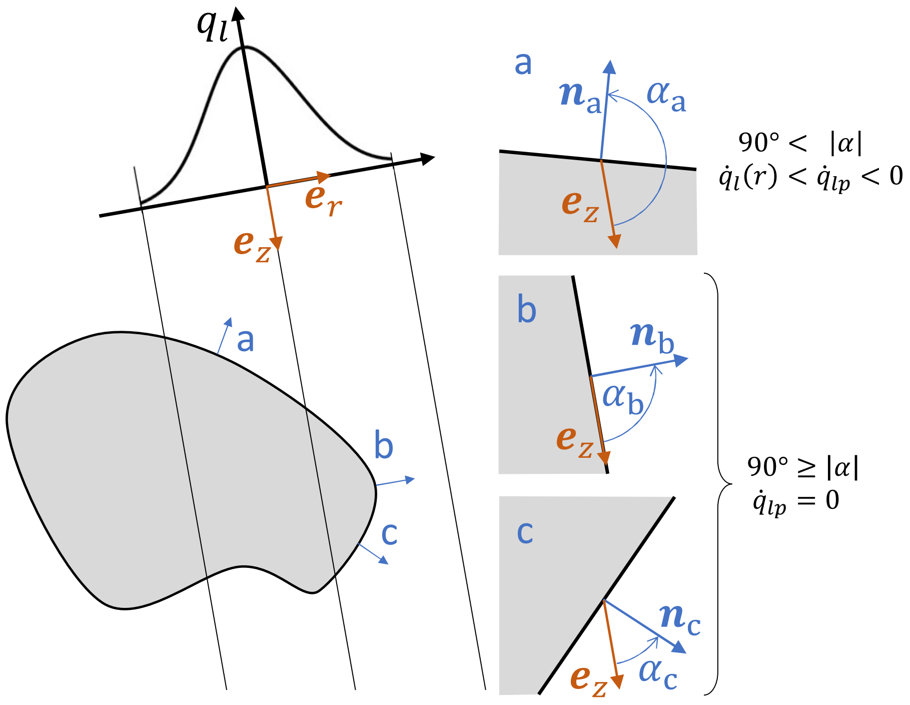


FIGURE 2.2: Laser power density for different incidence angles between laser axis $\boldsymbol{e}_z$ and unit outward surface normal $\boldsymbol{n}$.

$$q_{lp}(x, y, t) = \begin{cases} -\cos(\alpha)\, q_l(x, y, t) & \text{for } 90^\circ \leq \alpha \leq 270^\circ \\ 0 & \text{for } \alpha < 90^\circ \text{ and } \alpha > 270^\circ \end{cases} \tag{2.30}$$

No ray tracing is used, since we limit ourselves to conduction mode welding, where no keyhole appears and we expect no splashing, powder projection, powder denudation or any other challenging surface deformation that can influence the radiative interaction between the surfaces involved.

Radiation

The surface radiation heat flux $\boldsymbol{q}_{rad}$ to the atmosphere is modeled using the following equation

$$q_{rad_i} = \epsilon_{rad}\sigma_{SB}(T^4(x, y, t) - T_\infty^4)n_i \tag{2.31}$$

where ϵ_{rad} is the material-specific emissivity, $\sigma_{SB} = 5.670\,373 \times 10^{-8}\,\mathrm{W\,m^{-2}\,K^{-4}}$ is the Stefan-Boltzmann constant, T is the local temperature of the surface, T_∞ is the surrounding temperature. As for the laser heat input, the surface radiation is idealized and simplistic. No interaction between different radiating surfaces is expected and therefore not accounted for in the model. Radiation is a Robin type boundary condition of the heat equation because it involves both the local temperature and the local temperature gradient (i.e. the heat flux).

Convection

The convective heat transfer at the surface is modeled with a phenomenological model as well, Newton's law of cooling:

$$q_{conv_i} = h_{conv}(T(x, y, t) - T_\infty)n_i \tag{2.32}$$

where h_{conv} is the convective heat transfer coefficient in $\mathrm{W\,m^{-2}\,K^{-1}}$ specific to the scenario (material-atmosphere pairing, surface roughness, relative velocity, turbulence, etc.), T is the local temperature of the surface, T_∞ is the surrounding temperature, which does not need to be the same as in eq. 2.31, but is chosen to be identical in this work. The surface convection is also a Robin type boundary condition. Note that this boundary condition is not used in any of the test cases described in this work.

Conduction

Conduction in all materials is modeled using the well-known Fourier's law

$$q_i = -k\frac{\partial T}{\partial x_i} \tag{2.33}$$

where k is the material's heat conductivity in $\mathrm{W\,m^{-1}\,K^{-1}}$, which is here assumed to be isotropic. Fourier's law (eq. 2.33) can now be inserted into the heat equation (eq. 2.25), which now becomes

$$\rho c_p \dot{T} - \frac{\partial}{\partial x_i}\left(k\frac{\partial T}{\partial x_i}\right) - R = 0 \tag{2.34}$$

Liquid fraction

Phase change is governed by the liquid fraction f_l (sometimes called phase fraction), which can be function of many state variables. We define it in our work only as a function of temperature T:

$$f_l = \begin{cases} 1 & \text{for } T \geq T_{liq} \\ \frac{T - T_{liq}}{T_{liq} - T_{sol}} & \text{for } T_{liq} > T > T_{sol} \\ 0 & \text{for } T \leq T_{sol} \end{cases} \tag{2.35}$$

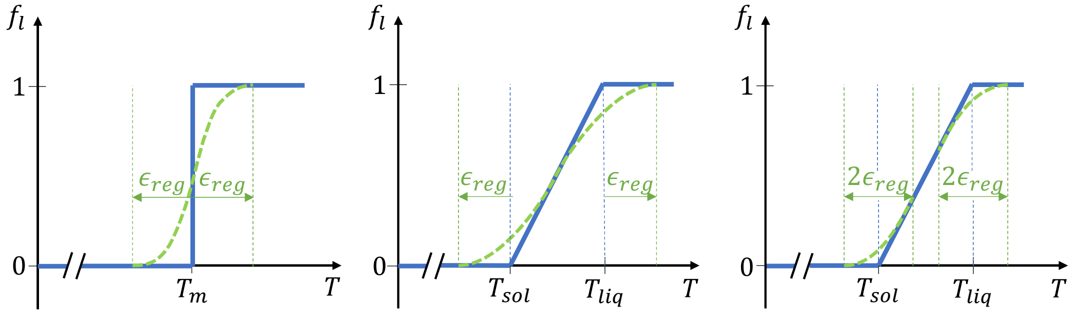
where T_{liq} is the liquidus temperature, T_{sol} is the solidus temperature and $f_l = 1$ marks a fully liquefied material state, $f_l = 0$ marks a fully solidified material state and $0 < f_l < 1$ marks a semi-solidified or mushy material state. Having a non-zero phase change interval $\Delta T = T_{liq} - T_{sol}$ is the general case and shall be referred to as **non-isothermal phase change**. Conversely, the particular case where the interval collapses into one melting temperature $T_m = T_{sol} = T_{liq}$ is referred to as **isothermal phase change**. Note that the linear transition of f_l within the phase change interval is a modeling choice. More realistic transition functions (e.g. Scheil equation as applied in Koeune and Pothot [76]) can be chosen in the future instead, to take cooling rates into account.

Latent heat

Latent heat is absorbed from the (sensible) heat upon melting and released back to it upon solidification. The latent heat is therefore treated as a source term to the heat equation. The phase change is governed by the liquid fraction (eq. 2.35) defined above and so is the release of latent heat. The power density of the latent heat release R_L is

$$R_L = -\rho \frac{\partial f_l}{\partial T} \dot{T} L_m \tag{2.36}$$

where L_m is the latent heat of fusion, in J kg^{-1}. Examining the definition of the liquid fraction f_l in eq. 2.35, we see that $\partial f_l / \partial T = 1/\Delta T$ inside the transition interval $T_{liq} > T > T_{sol}$ and $\partial f_l / \partial T = 0$ outside of it. It becomes apparent that isothermal phase change lets $\partial f_l / \partial T \to \infty$. Even when ΔT is finite but small, numerical problems can be encountered. A popular solution (e.g. [65, 77, 78]) to dealing with this large gradient is the use of a regularization. There, the sharp transition of eq. 2.35 is replaced by a smooth function. A cubic function is used in a temperature interval of $\pm\epsilon_{reg}$ around a sharp step, ramp or edge. A cubic function is one of the simplest functions that can generate a smooth transition (C^1 continuous) between two points on a function and is easily parameterized. At each end of the interval, the cubic function matches the value and gradient of the original function, as shown in fig. 2.3.



(a) At the melting point with a tolerance $\pm\epsilon_{reg}$.

(b) Across the entire solidificaton interval with a tolerance $\pm\epsilon_{reg}$.

(c) At each edge individually with a tolerance $\pm\epsilon_{reg}$.

FIGURE 2.3: Regularization of sharp liquid fraction f_l to avoid problems with latent heat term. Original function in solid blue, smooth regularization function in dashed green.

Omitted heat transfer mechanisms

The atmospheric gas and its flow are not included in the model. Not including the atmosphere is made possible by the Lagrangian approach, while allowing the free surface to deform. Including the atmosphere is also not necessary, as long as the interaction between dense material and atmosphere is well enough approximated by the surface convection model (eq. 2.32). This is common practice, if the simulation approach allows it (see e.g. Saadlaoui et al. [79], Khairallah et al. [3]) This model would break down, when the surface shape becomes too complex (i.e. keyhole, splashing, etc., see e.g. Courtois et al. [80]) or the flow field, temperature or composition of the atmospheric gas is too non-uniform or not well known a priori (i.e. plume of evaporating substrate material, impinging cooling gas, etc., see e.g. Mayi et al. [29]). In such scenarios, modeling of the gaseous atmosphere becomes necessary.

The cooling through evaporation occurs mainly at the hottest region on the surface, where the laser impinges. In that region we can expect the current simplistic laser model to have a larger error than what the omission of the evaporative cooling would contribute, so we deem it safe to neglect it until the laser model is improved.

While evaporative cooling can have a significant contribution towards the distribution and flow of heat, the heat lost in the material due to mass lost upon evaporation is assumed to be small (see e.g. Khairallah et al. [3]).

While viscosity plays an important role in the fluid dynamics of the melt pool, the dissipated viscous power that would be added to the heat balance can be neglected. Its contribution is insignificant compared to the laser heat input and the convective heat transfer (see e.g. Körner et al. [58]).

2.3.2 Mass and density

Mass is to be preserved and according to eq. 2.7 any change in density must come with a corresponding change in volume. In eq. 2.8, changes in density or volume were defined to be achieved through changes in temperature or pressure. In this work two material properties are included that relate density to pressure and temperature: (mechanical) compressibility and thermal expansion.

Thermal expansion

Thermal expansion is the change in volume or density of matter due to a change in temperature. At constant pressure it is defined as

$$\left.\frac{\partial \rho}{\partial T}\right|_p = -\alpha_V \rho \tag{2.37}$$

where α_V is the coefficient of volumetric thermal expansion in K^{-1}. If a coefficient of linear thermal expansion α_L is given, the volumetric one is obtained by

$$\alpha_V = 3\,\alpha_L \tag{2.38}$$

After inserting eq. 2.37, the mass conservation equation (eq. 2.8) with thermal expansion becomes

$$\frac{1}{\rho}\frac{\partial \rho}{\partial p}\dot{p} - \alpha_V \dot{T} + \frac{\partial v_i}{\partial x_i} = 0 \tag{2.39}$$

Note that no two-way coupling with the heat equation is implemented. A temperature-related term appears in the mass conservation equation, but no term related to deformation appears in the heat equation (see explanation in section 2.2.3).

Compressibility

In this work, (mechanical) compressibility refers to the effect of changing the density or volume through pressure and vice-versa. A compressible material has a compressibility $\kappa \neq 0$ in Pa^{-1} or a bulk modulus $K = \kappa^{-1} < \infty$ in Pa, whereas an incompressible medium is idealized to have $\kappa = 0$ or $K = \infty$. Some authors in the field of fluid dynamics also use the term "incompressible flow" when referring to a velocity field that is free of divergence, independent of the material used. In this work, (in)compressibilty will only refer to a material behavior, whereas a divergence-free flow will be called isochoric.

We will write our equations using the bulk modulus K, as is common in solid mechanics so that the compressibility is governed by the following relation

$$\left.\frac{\partial \rho}{\partial p}\right|_T = \frac{\rho}{K} \tag{2.40}$$

where K is the bulk modulus. To recover an incompressible material behavior, the user may set $K \to \infty$. The equation of mass conservation eq. 2.39 can now be completed using eq. 2.40 and we obtain

$$\frac{1}{K}\dot{p} + \alpha_V \dot{T} + \frac{\partial v_i}{\partial x_i} = 0 \tag{2.41}$$

In this work all fluids are considered mechanically incompressible ($\frac{\partial \rho}{\partial p} = 0$), unless a bulk modulus is explicitly given.

2.3.3 External and body forces

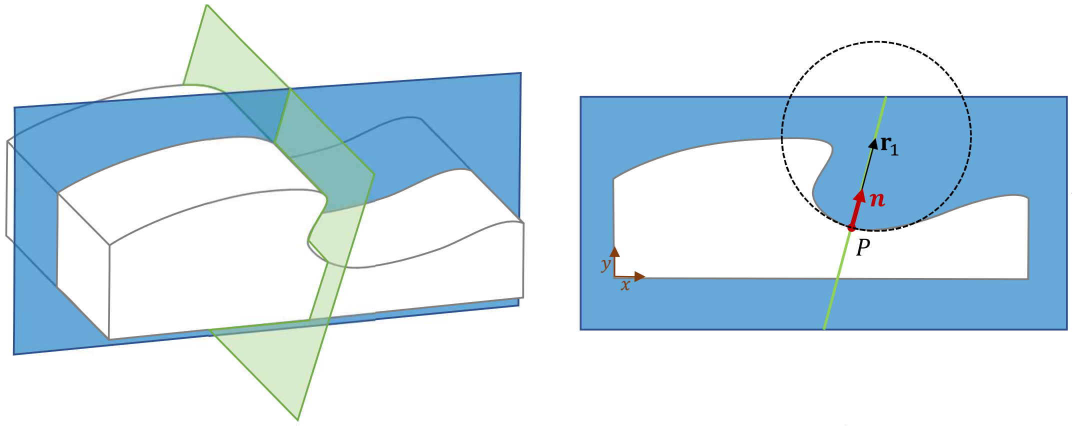
Surface tension

Surface tension occurs at the interface between two media. The local magnitude of the surface tension γ in $\mathrm{N\,m^{-1}}$ depends mainly on the material pairing, temperature and solute concentration. The latter two can vary along the interface, which means that the tension can vary along the interface. While microscopically, the tension can be explained by a disparity of cohesive and adhesive forces of the two media in contact, two distinct phenomena can be observed macroscopically: a normal traction and a tangential traction.

The **normal traction** is the net pressure on a curved surface, which effectively minimizes the surface area. The magnitude of the surface tension normal traction $\bar{t}_{n_{ST}}$ follows

$$\|\bar{t}_{n_{ST}}\| = \gamma \kappa_s \tag{2.42}$$

where the overbar indicates that the quantity is an applied surface traction at the boundary, the subscript $_n$ indicates the normal component and the subscript $_{ST}$ indicates that that surface tension is causing it. γ is the surface tension (coefficient) in $\mathrm{N\,m^{-1}}$, κ_s is the local curvature of the surface or interface in $\mathrm{m^{-1}}$.

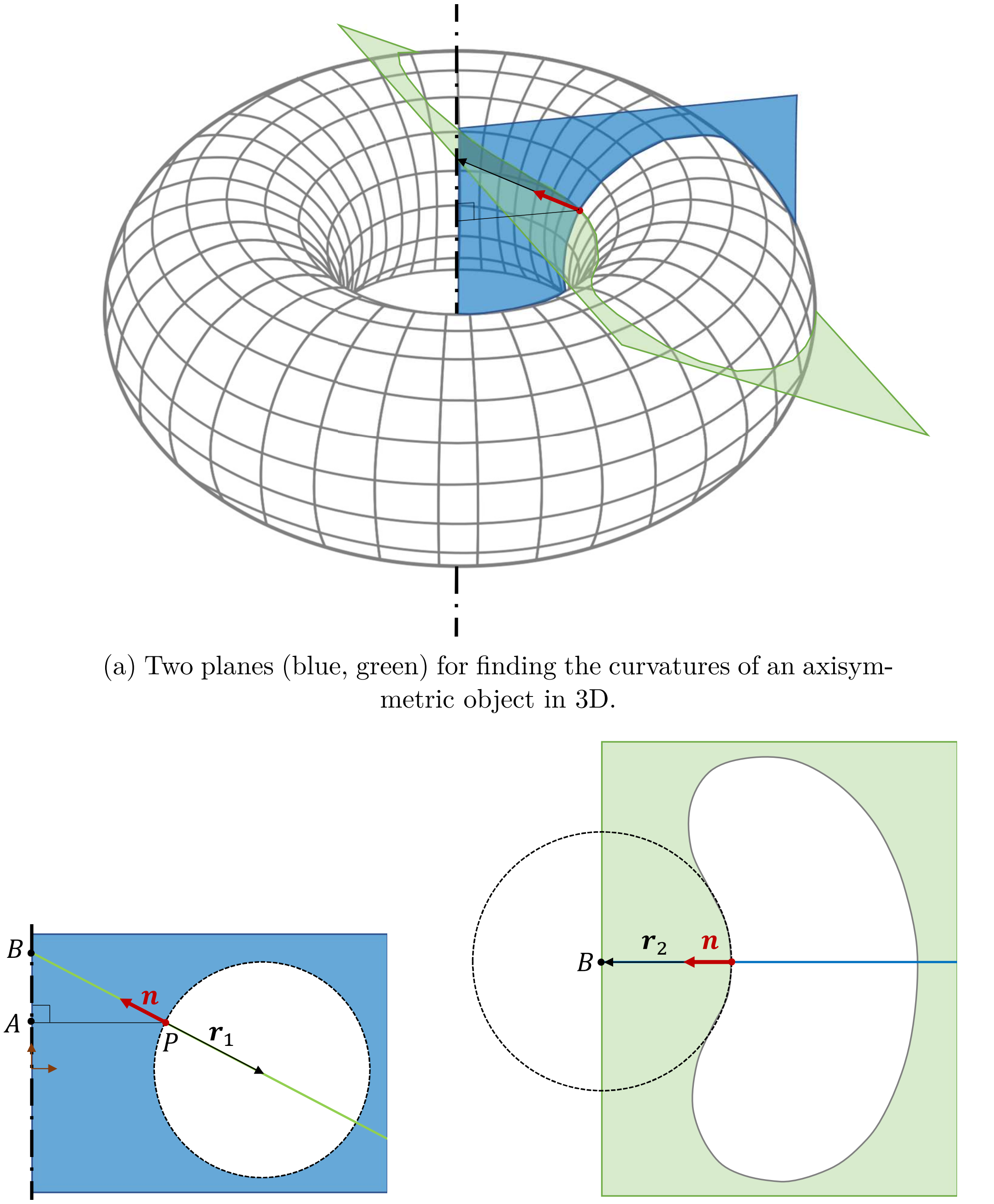


(a) Two planes (blue, green) for finding the curvatures of an object (white) in 3D.

(b) Projection of white object onto the blue plane (i.e. the x, y-plane). Curvature $\kappa_1 = r_1^{-1}$ at point P.

FIGURE 2.4: Curvature in 2D plane strain. Two planes (blue, green) are orthogonal to one another and orthogonal to the tangent plane (not shown) that is defined by the outward normal vector (red) at point P. The curvature is obtained from the projection of the object on each of the planes.

To find the curvature at a given point P on a surface in 3D, one finds the tangent plane orthogonal to the unit outward normal vector $\boldsymbol{n}$ at that point. Then we define two planes orthogonal to the tangent plane and orthogonal to



(a) Two planes (blue, green) for finding the curvatures of an axisymmetric object in 3D.

(b) Blue plane (i.e. x, y plane) for finding the curvature $\kappa_1 = r_1^{-1}$ at point P.

(c) Green plane for finding the curvature $\kappa_2 = r_2^{-1}$ at point P. r_2 is the distance between point P and B.

FIGURE 2.5: Curvature in 2D axisymmetric. Two planes (blue, green) are orthogonal to one another and orthogonal to the tangent plane (not shown) that is defined by the outward normal vector (red) at point P. The curvature is obtained from the projection of the object on each of the planes.

each other, as depicted in figs 2.4a and 2.5a. The intersection of the surface with each of the planes gives a curve each, as in figs 2.4b, 2.5b and 2.5c. The curvatures κ_1 and κ_2 on each of these curves at P can then be obtained. The resulting curvature κ_s at the point P on a surface in 3D is calculated as

$$\kappa_s = \kappa_1 + \kappa_2 = \frac{1}{r_1} + \frac{1}{r_2} \tag{2.43}$$

which is twice the mean curvature. r_1 and r_2 are the signed radii of curvature for κ_1 and κ_2 respectively. For r_1 the sign depends on whether or not the radius vector $\boldsymbol{r}_1$ points inward or outward, hence

$$r_1 = \frac{\|\boldsymbol{r}_1\|}{\boldsymbol{n} \cdot \boldsymbol{r}_1} \tag{2.44}$$

In the 2D representation used in this work, one of the planes is the x, y plane (blue plane in figs. 2.4,2.5) and the curve that we see (black dotted circle) is the intersection of x, y with the 3D surface. The radius of curvature r_1 (black arrow) and therefore κ_1 can be obtained from this curve (see figs. 2.4b, 2.5b). The other plane (green) is the one orthogonal to the tangent plane and the x, y plane. Under the plane strain assumption, the intersection with the second plane is a straight line and $\kappa_2 = 0$ (fig. 2.4a). Under the axisymmetry assumption, the second radius of curvature $r_2 = \overline{BP}$ is the distance of point P from the symmetry axis along the outward normal $\boldsymbol{n}$ at P. It can be obtained from the more readily available radial distance $\boldsymbol{r}_p = \overline{AP}$ of point P from the symmetry axis using

$$r_2 = \frac{\|\boldsymbol{r}_p\|}{\boldsymbol{n} \cdot \boldsymbol{r}_p} \tag{2.45}$$

Note that the dot product is used to preserve the sign of r_2, which is important. Using eq. 2.42, the resulting surface traction of the normal component $\bar{\boldsymbol{t}}_{n_{ST}}$ has a direction that is determined by the direction of the vector $\boldsymbol{r}_1$, such that

$$\bar{\boldsymbol{t}}_{n_{ST}} = \gamma \kappa_s \frac{\boldsymbol{r}_1}{\|\boldsymbol{r}_1\|} \tag{2.46}$$

and in the case that $\|\boldsymbol{r}_2\| < \|\boldsymbol{r}_1\|$ and $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ point in opposite directions, the sign will be reversed (according to eq. 2.43).

The **Marangoni effect** is the surface tension's net tangential traction and it originates from a gradient in surface tension along the surface or interface. Its magnitude, $\bar{t}_{t_{ST}}$ is given by

$$\bar{t}_{t_{ST}} = \frac{\partial \gamma}{\partial T} \nabla_S T = \frac{\partial \gamma}{\partial T} \left[\nabla T - (\boldsymbol{n} \cdot \nabla T)\, \boldsymbol{n} \right] \tag{2.47}$$

where $\partial\gamma/\partial T$ is the Marangoni coefficient in $\mathrm{N\,m^{-1}\,K^{-1}}$ and $\nabla_S \phi$ is the operator for the gradient of any variable ϕ along the surface and $\boldsymbol{n}$ being the unit outward normal vector. It is assumed here that γ is only a function of temperature. Note that γ and $\partial\gamma/\partial T$ could be defined independently, which is the case in all test cases in the present work.

Body force and buoyancy

The body force term in the momentum equation eq. 2.14 is usually used to include the effect of gravity. In this work the body force is caused by a gravitational acceleration of $g = 9.81\,\text{meter}/\text{s}^2$, unless otherwise stated.

The effect of buoyancy is incorporated into the momentum equation by altering the local value of the density ρ. The density is updated using eqs. 2.37 and 2.40 using the following equation:

$$\rho = \rho_{ref} \exp\left(\int_{p_{ref}}^{p} \frac{1}{K}\,dp - \int_{T_{ref}}^{T} \alpha_V\,dT\right) \tag{2.48}$$

where the subscript $_{ref}$ refers to a variable at some known reference value, K is the bulk modulus and α_V is the thermal expansion coefficient (see section 2.3.2). Under gravitation, differences in density between particles will lead to a difference in the body force term, which then leads to buoyancy. In our code, the user can choose between

1. applying the changed density only to the body force term (= **Boussinesq approximation**)
2. applying the changed density to all terms that contain the density

Independently of the choice above, the user can include the actual expansion in space due to temperature changes as outlined in section 2.3.2). In most applications it is sufficient to model the change in density only in the body force term, to reproduce natural convection sufficiently accurately. For a mechanically incompressible medium (i.e. $K \to \infty$, see section 2.3.2) in a closed system it may even be necessary to avoid the spatial expansion, since the incompressible medium cannot expand anywhere. This situation is encountered for example in the natural convection test case in section 3.2.2. Conversely, if volume expansion is modeled, the density changes should always be modeled in all terms or else an error in mass conservation and energy conservation is introduced.

Solid region flow resistance (Carman-Kozeny equation)

When a physically accurate behavior of the solid is not required, the solid behavior can be approximated by fluid particles that are unable to move. This can be achieved by a source term $\boldsymbol{S}$ in the momentum equation. According to Cook and Murphy [13], this approach is commonly chosen when fluid dynamics codes are used to simulate phase change problems. There, the commonly used equation for this source term is based on the **Carman-Kozeny equation**, usually used to describe flow resistance in porous media. The adapted source term $\boldsymbol{S}$ is

$$S_i = -C_{CK} \frac{1 - f_l^2}{f_l^3 + \epsilon_{CK}} v_i \tag{2.49}$$

where subscript $_i$ refers to the component in the i-direction and C_{CK} and ϵ_{CK} are user-chosen parameter. Cook and Murphy [13] explain that there is no physical meaning to either of the parameters in the context of modeling a solid. The first coefficient C_{CK} was originally related to the porosity of the medium, but for this model it is simply chosen to be large (e.g. $C_{CK} = 1 \times 10^5\,\mathrm{kg\,m^{-3}\,s^{-1}}$). The second coefficient exists to avoid division by zero when $f_l = 0$ and must be small (e.g. $\epsilon_{CK} = 1 \times 10^{-5}$) to not introduce a large error in the fluid. The source term becomes very large and negative when the liquid fraction $f_l \to 0$, which prevents the movement of the particle, making the particle appear solidified.

If this source term is used, it is included in the momentum conservation equation eq. 2.14, which then becomes

$$\rho \dot{v}_i - \frac{\partial \sigma_{ij}}{\partial x_j} - \rho b_i - S_i = 0 \tag{2.50}$$

Omitted forces

Recoil pressure is a normal force on a surface from which a considerable amount of mass is vaporized. The volume of a given amount of material increases rapidly as it evaporates. Being confined by the surface of the remaining liquid and solid material, the vapor can only expand away from the surface. This effectively accelerates the vaporized material and leads to a reaction force on the surface, which is referred to as recoil pressure. This effect is particularly important in the context of welding and additive manufacturing, when switching from (low power density) conduction mode to (high power density) keyhole mode (see section 1.2.2). The recoil pressure is the main force that opens up and keeps open the thin long hole under the heat source that is referred to as a keyhole [18]. Liquid splatter from the melt pool can also be attributed to the recoil pressure. We omit this effect at this stage, limiting ourselves to conduction mode applications, where the recoil pressure plays a less important role. A very complete summary of the implementation of recoil pressure in the context of additive manufacturing can be found in [13].

Turbulence in the fluid is not considered because the small-scale turbulent motion cannot be resolved. Reynolds numbers ($Re = \rho v l_c / \mu$, with l_c being a characteristic length scale) are low in the type of problems considered in this work (e.g. $Re \approx 20$ for a L-PBF process with ceramics in [4], $Re \approx 2000$ for laser spot welding of steel in [12]). To compensate for this, the physical effect of increased thermal and viscous diffusivity can be increased artificially. Some authors (e.g. [12, 81]) refer to this as "enhancement" of the diffusivity and introduce an enhancement factor f_{enh} by which these quantities are increased. In this work, no enhancement of such sort is used unless specifically mentioned.

2.4 Derivation of the unified solid/fluid material behavior

We show that fluid and solid treatments are not fundamentally different and we use the similarities to cast fluid and solid material behaviors in the same form using an incremental approach.

As a first step, the stress is split into volumetric stress $\mathrm{vol}(\boldsymbol{\sigma})$ and deviatoric stress $\mathrm{dev}(\boldsymbol{\sigma}) = \boldsymbol{s}$:

$$\sigma_{ij} = \underbrace{-p\delta_{ij}}_{\mathrm{vol}(\boldsymbol{\sigma})} + \underbrace{s_{ij}}_{\mathrm{dev}(\boldsymbol{\sigma})} \tag{2.51}$$

where δ_{ij} is the Kronecker delta. The pressure p here is the mechanical pressure that is defined as

$$p = -\frac{1}{3}\,\mathrm{tr}(\boldsymbol{\sigma}) \tag{2.52}$$

The negative sign in front of the pressure p is added because it is more intuitive to use the convention from fluid dynamics in a fluid dynamics approach such as PFEM. The convention is that a positive pressure indicates a compressive stress state and vice-versa. Now, the momentum equation (eq. 2.14) can be updated by inserting eq. 2.51:

$$\rho\dot{v}_i + \frac{\partial p}{\partial x_i} - \frac{\partial s_{ij}}{\partial x_j} - \rho b_i = 0 \tag{2.53}$$

To solve this form of the momentum conservation, we still require an expression for the deviatoric stress tensor s_{ij}. The pressure on the other hand is determined using the continuity equation (eq. 2.41), as is explained in detail further down. First, the deviatoric stress tensor s_{ij} is rewritten in an incremental form over a given time increment $\Delta t = t - t_0$, where t is the current time and t_0 is the reference time at the beginning of the time interval.

$$s_{ij}(t) = \int_{t_0}^{t} \dot{s}_{ij}\, dt \; + \; s_{ij}^{t_0} \; = \Delta s_{ij} \; + \; s_{ij}^{t_0} \tag{2.54}$$

We denote $\Delta\boldsymbol{s}$ as the deviatoric stress increment over a given time interval and $\boldsymbol{s}^{t_0}$ as the reference deviatoric stress at the beginning of the time interval. What is inserted for either of these variables depends on the type of material. We rewrite the momentum equation (eq. 2.53), taking eq. 2.54 into account:

$$\rho\dot{v}_i + \frac{\partial p}{\partial x_i} - \frac{\partial}{\partial x_j}\left(\Delta s_{ij} + s_{ij}^{t_0}\right) - \rho b_i = 0 \tag{2.55}$$

This equation is valid for any continuum, as long as the expression for Δs is exact. Eq. 2.55 can be used for any continuum, solid and fluid, as shown in the following sections.

2.4.1 Linear elastic solid

In the solid part, **small displacements are assumed** in the context of welding and additive manufacturing. The solid material does build up residual stresses during the process, but the resulting deformation of the finished part is usually a mild warping at the part-scale, which may even be hardly noticeable at the meso-scale. For industrial use, the warping is a significant problem, but in the numerical simulation context this strong assumption will not be a limitation.

Let us first define the strain and strain rate tensor, which play an important role in continuum mechanics. The strain tensor ε_{kl} and strain rate tensor $\dot{\varepsilon}_{kl}$ are defined as

$$\varepsilon_{kl} = \frac{1}{2}\left(\frac{\partial u_k}{\partial x_l} + \frac{\partial u_l}{\partial x_k}\right) \tag{2.56}$$

$$\tag{2.57}$$

$$\dot{\varepsilon}_{kl} = \frac{1}{2}\left(\frac{\partial v_k}{\partial x_l} + \frac{\partial v_l}{\partial x_k}\right) \tag{2.58}$$

where $\boldsymbol{u}$ is the displacement. With this in mind, the linear elastic solid constitutive relation is written as

$$\sigma_{ij} = \mathbb{H}_{ijkl}\varepsilon_{kl} \tag{2.59}$$

where the 4th order tensor $\mathbb{H}_{ijkl}$ for isotropic materials is a function of Young's modulus E in Pa and Poisson's ratio ν. We use instead the shear modulus G[2], the bulk modulus K and the second Lamé parameter λ, all of which are in Pa.

$$G = \frac{E}{2(1+\nu)} \tag{2.60}$$

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \tag{2.61}$$

$$K = \frac{E}{3(1-2\nu)} \qquad = \lambda + \frac{2}{3}G \tag{2.62}$$

For isotropic materials we rewrite eq. 2.59, replacing $\mathbb{H}_{ijkl}$ by a parametric expression that uses the above parameters λ and G:

$$\sigma_{ij} = \lambda\delta_{ij}\varepsilon_{kk} + G(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})\varepsilon_{kl} \tag{2.63}$$

With eq. 2.62 we can split the stresses into deviatoric and volumetric contributions:

$$\sigma_{ij} = \underbrace{K\delta_{ij}\varepsilon_{kk}}_{\mathrm{vol}(\boldsymbol{\sigma})} + \underbrace{G\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk} - \frac{2}{3}\delta_{ij}\delta_{kl}\right)\varepsilon_{kl}}_{\mathrm{dev}(\boldsymbol{\sigma})} \tag{2.64}$$

The volumetric part is simply replaced by the (negative) pressure p on the diagonal of $\mathrm{vol}(\sigma_{ij})$:

[2] The shear modulus G is also known as first Lamé parameter μ_s, not to be confused with the fluid dynamic viscosity also denoted by the symbol μ_f.

$$\begin{aligned}\mathrm{vol}(\sigma_{ij}) &= K\delta_{ij}\varepsilon_{kk} \\ &= -p\delta_{ij} \qquad (2.65)\\ \mathrm{dev}(\sigma_{ij}) &= G\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk} - \frac{2}{3}\delta_{ij}\delta_{kl}\right)\varepsilon_{kl} \\ &= 2G\ \mathrm{dev}\left(\varepsilon_{ij}\right) = s_{ij} \qquad (2.66)\end{aligned}$$

Recall that the negative sign in eq. 2.65 appears due to the fluid dynamics convention for the pressure. As stresses increase linearly with increasing strains and assuming G to be constant over the time interval, the time derivative of eq. 2.66 yields

$$\dot{s}_{ij} = 2G\ \mathrm{dev}\left(\dot{\varepsilon}_{ij}\right) \tag{2.67}$$

Integrating eq. 2.67 over the time interval $\Delta t = t - t_0$ and assuming that $\dot{\boldsymbol{\varepsilon}}$ is constant during Δt, the deviatoric stress increment Δs_{ij} is obtained as.

$$\Delta s_{ij} = 2G\Delta t\ \mathrm{dev}\left(\dot{\varepsilon}_{ij}\right) \tag{2.68}$$

Note that once the small displacement assumption is abandoned in the future, the stress deviator rate of change $\dot{\boldsymbol{s}}$ needs to become frame-indifferent (unaffected by rotations) by using an objective measure of the stress rate (see e.g. Koeune and Ponthot [76] or Franci et al. [47]).

2.4.2 Newtonian fluid

The Newtonian fluid law is

$$\sigma_{ij} = -p_{th}\delta_{ij} + \lambda_f\dot{\varepsilon}_{kk} + 2\mu_f\dot{\varepsilon}_{ij} \tag{2.69}$$

where p_{th} is a component of the pressure p, sometimes referred to as "thermodynamic pressure" (e.g. [76, 82, 83]), although there is no rigorously used convention for this. As will become apparent in the next paragraph, $p_{th} = p$ in most cases, which is why the distinction is usually not made. λ_f is the second viscosity (not to be confused with the second Lamé parameter λ) and μ_f the dynamic shear viscosity. Similar to eq. 2.62, we introduce the bulk viscosity κ_f to replace λ_f in eq. 2.69:

$$\kappa_f = \lambda_f + \frac{2}{3}\mu_f \tag{2.70}$$

This allows us to then split the stress tensor into volumetric and deviatoric components.

$$\sigma_{ij} = \underbrace{-p_{th}\delta_{ij} + \kappa_f\delta_{ij}\dot{\varepsilon}_{kk}}_{\mathrm{vol}(\boldsymbol{\sigma}) = -p\delta_{ij}} + \underbrace{\mu_f\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk} - \frac{2}{3}\delta_{ij}\delta_{kl}\right)\dot{\varepsilon}_{kl}}_{\mathrm{dev}(\boldsymbol{\sigma})} \tag{2.71}$$

The term that contains κ_f contributes towards the pressure, while the medium is being compressed or expanded at a non-zero rate. However, in many applications Stokes' hypothesis ($\kappa_f = 0$) is applied for several practical reasons:

- Neither λ_f nor κ_f are easy to measure and data is therefore seldomly available.
- When a medium is assumed incompressible ($\dot{\varepsilon}_{kk} = 0$), this term automatically vanishes.
- The term containing κ_f is often small and its inclusion does not alter the pressure significantly, unless the rate of compression or expansion is very large, for example when shocks are encountered.

Contrary to the last point, it has been noted that κ itself is indeed oftentimes not much smaller or even larger than μ (e.g. for water $2.1 < \kappa_f/\mu_f < 2.5$ according to Dukhin and Goetz [84] or Rosenhead [85], $\kappa_f/\mu_f \approx 1.7$ for molten bismuth and $\kappa_f/\mu_f \approx 4$ for molten lead according to Flinn et al. [86]). It is however argued (see e.g. [82]) that the last point ($|\kappa_f \dot{\varepsilon}_{kk}| << |p_{th}|$) is what really matters. Due to the absence of shocks, Stokes' hypothesis is an acceptable assumption in this work, so the second term in eq. 2.71 will be omitted. Finally, we arrive at the following equations for the Newtonian fluid with Stokes' hypothesis:

$$\mathrm{vol}(\sigma_{ij}) = -p_{th}\delta_{ij} = -p\delta_{ij} \tag{2.72}$$

$$\mathrm{dev}(\sigma_{ij}) = \mu_f \left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk} - \frac{2}{3}\delta_{ij}\delta_{kl}\right)\dot{\varepsilon}_{kl} \tag{2.73}$$

$$= 2\mu_f \ \mathrm{dev}\left(\dot{\varepsilon}_{kl}\right) = s_{ij} \tag{2.74}$$

The volumetric stress is now identical to that of the solid material and the focus can now remain on the deviatoric stress. To rewrite the deviatoric stress s_{ij} in incremental form, we first acknowledge that a Newtonian fluid "forgets" any viscous stress[3], effectively resetting the reference deviatoric stress at the beginning of the time interval:

$$\boldsymbol{s}^{t_0} = 0 \tag{2.75}$$

and as a consequence of eq. 2.75 and 2.54, we can write

$$\boldsymbol{s} = \Delta\boldsymbol{s} \tag{2.76}$$

The deviatoric stress increment for a Newtonian fluid in this framework therefore becomes

$$\Delta s_{ij} = 2\mu_f \ \mathrm{dev}\left(\dot{\varepsilon}_{ij}\right) \tag{2.77}$$

Note that the fluid can be made compressible by setting a finite value for K in eq. 2.41. If the user prefers to assume the fluid to be incompressible, then $K \to \infty$ and the pressure acts as a Lagrange multiplier that prescribes

[3] Viscous stresses are volumetric and deviatoric stresses caused by a viscosity and a strain rate. Here, this is only the viscous shear stress, but if Stoke's hypothesis were not used, the viscous stresses would also include the viscous volumetric stress related to κ_f.

the relative volume change. In the absence of thermal expansion, this relative volume change is forced to be zero. The pressure p then only consists of the hydrostatic pressure in the fluid. This choice brings with it new challenges, which will be discussed further down (section 2.5.4 "Stabilization").

2.4.3 Unified formulation

A single set of equations can be derived that describes both linear elastic solid and compressible Newtonian fluid, such that only material parameters make up the difference between the two.

First, we recall that both solid and fluid obey mass conservation eq. 2.41, which governs the compressibility as well. On the other hand, comparing the deviatoric stress increments in eqs. 2.68 and 2.77, we find two different material parameters, the shear modulus G in the solid and the shear viscosity μ_f. They each describe unrelated physical properties, tend to differ in several orders of magnitude and use different units ([Pa] and [Pa s] respectively). For the sake of unification, we introduce a viscosity-like parameter μ^* of the solid:

$$G\Delta t = \mu_s^* \tag{2.78}$$

With the new parameter from eq. 2.78 for the solid, we unify eqs. 2.68 and 2.77 and get a single expression for the deviatoric stress increment Δs_{ij} that is valid in the solid and fluid

$$\Delta s_{ij} = 2\mu \ \operatorname{dev}\left(\dot{\varepsilon}_{ij}\right) \tag{2.79}$$

where μ is defined as

$$\mu = \begin{cases} \mu_s^* \text{ if } f_l = 0 \\ \mu_f \text{ otherwise} \end{cases} \tag{2.80}$$

This means that elastic stresses are only present in a material that is *fully* solid ($f_l = 0$), not just semi-solid ($0 < f_l < 1$). This assumption is made because different materials may behave fundamentally differently in the semi-solid state and there is no assumption that can be made that is valid for all materials. Instead, the semi-solid will be treated like a Newtonian fluid, but with increasing flow resistance as it crosses from liquid ($f_l = 1$) to solid ($f_l = 0$). This can be achieved either by artificially increasing the viscosity, or by using the momentum sink term in eq. 2.49. In the future, more realistic, material-specific models for the semi-solid phase can be introduced (to give one example: Koeune and Ponthot [76] present a thixotropic material model for semi-solid aluminium using a viscoplastic material law with isotropic hardening that is a function of liquid fraction.).

The above eq. 2.79 can be inserted into eq. 2.54, along with the reference deviatoric stress $\boldsymbol{s}^{t_0}$ that is known during the current time interval. This yields the current deviatoric stress $\boldsymbol{s}$ at time t, which can then be inserted back into the momentum equation 2.53.

2.4.4 Voigt notation

The deviatoric stress increment in eq. 2.79 is currently a function of the strain rate tensor $\dot{\varepsilon}_{ij}$. This section aims at rewriting this expression into a function of velocity. Recall how the stress tensor σ_{ij} is composed in this work, according to eqs. 2.51 and 2.54:

$$\sigma_{ij} = \underbrace{\Delta s_{ij} + s_{ij}^{t_0}}_{s_{ij}} - p\delta_{ij} \tag{2.81}$$

Analogous to the fourth-order stress-strain tensor in eq. 2.59, we introduce a fourth-order tensor $\mathbb{H}_{ijkl}^{dev}$ that governs the relation between the second-order deviatoric stress increment Δs_{ij} and strain rate $\dot{\varepsilon}_{kl}$ tensors:

$$\Delta s_{ij} = \mathbb{H}_{ijkl}^{dev}\, \dot{\varepsilon}_{kl} \tag{2.82}$$

According to eq. 2.79 its composition is

$$\mathbb{H}_{ijkl}^{dev} = \mu \left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk} - \frac{2}{3}\delta_{ij}\delta_{kl} \right) \tag{2.83}$$

The appearance of a fourth-order tensor makes the whole expression difficult to handle efficiently in a computer code. The first step is to use Voigt notation to take advantage of the symmetry of both the stress and strain rate tensors. The 6 independent components of the symmetric second-order tensors σ_{ij} and $\dot{\varepsilon}_{ij}$ are respectively mapped to the (first-order) column matrix σ_i and $\dot{\varepsilon}_i$ as follows

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ & \sigma_{yy} & \sigma_{yz} \\ & & \sigma_{zz} \end{bmatrix} \rightarrow \boldsymbol{\sigma} = [\sigma_{xx}\ \sigma_{yy}\ \sigma_{zz}\ \sigma_{yz}\ \sigma_{xz}\ \sigma_{xy}]^T \tag{2.84}$$

$$\dot{\boldsymbol{\varepsilon}} = \begin{bmatrix} \dot{\varepsilon}_{xx} & \dot{\varepsilon}_{xy} & \dot{\varepsilon}_{xz} \\ & \dot{\varepsilon}_{yy} & \dot{\varepsilon}_{yz} \\ & & \dot{\varepsilon}_{zz} \end{bmatrix} \rightarrow \dot{\boldsymbol{\varepsilon}} = [\dot{\varepsilon}_{xx}\ \dot{\varepsilon}_{yy}\ \dot{\varepsilon}_{zz}\ 2\dot{\varepsilon}_{yz}\ 2\dot{\varepsilon}_{xz}\ 2\dot{\varepsilon}_{xy}]^T \tag{2.85}$$

and analogously to the stress tensor, the mapping into Voigt notation is done also for the deviatoric stress increment tensor Δs_{ij}

$$\Delta \boldsymbol{s} = \begin{bmatrix} \Delta s_{xx} & \Delta s_{xy} & \Delta s_{xz} \\ & \Delta s_{yy} & \Delta s_{yz} \\ & & \Delta s_{zz} \end{bmatrix} \rightarrow \Delta \boldsymbol{s} = [\Delta s_{xx}\ \Delta s_{yy}\ \Delta s_{zz}\ \Delta s_{yz}\ \Delta s_{xz}\ \Delta s_{xy}]^T \tag{2.86}$$

Since only 2 dimensions are modeled in our current method, we must make an assumption about the third dimension. The two options featured in this work are the plane strain and axisymmetric assumptions that allow us to simplify the equations.

Plane Strain

In plane strain there are no out-of-plane strains ($\dot{\varepsilon}_{zz} = \dot{\varepsilon}_{xz} = \dot{\varepsilon}_{yz} = 0$), but there is the out-of-plane normal stress σ_{zz} (and likewise Δs_{zz}), which is usually not stored in the stress vector, as it can be reconstructed from known quantities. For consistency (especially in the code), both $\dot{\varepsilon}_{zz}$ and Δs_{zz} are listed.

$$\Delta \boldsymbol{s} = \begin{bmatrix} \Delta s_{xx} & \Delta s_{xy} & 0 \\ & \Delta s_{yy} & 0 \\ & & \Delta s_{zz} \end{bmatrix} \rightarrow \Delta \boldsymbol{s} = [\Delta s_{xx}\ \Delta s_{yy}\ \Delta s_{zz}\ \Delta s_{xy}]^T \tag{2.87}$$

$$\dot{\boldsymbol{\varepsilon}} = \begin{bmatrix} \dot{\varepsilon}_{xx} & \dot{\varepsilon}_{xy} & 0 \\ & \dot{\varepsilon}_{yy} & 0 \\ & & 0 \end{bmatrix} \rightarrow \dot{\boldsymbol{\varepsilon}} = [\dot{\varepsilon}_{xx}\ \dot{\varepsilon}_{yy}\ \dot{\varepsilon}_{zz}\ 2\dot{\varepsilon}_{xy}]^T \tag{2.88}$$

To preserve the stress-strain relation in eq. 2.83 in Voigt notation we get the corresponding second-order matrix H_{ij}^{dev}

$$\mathbf{H}^{dev} = \mu \begin{bmatrix} 4/3 & -2/3 & -2/3 & 0 \\ -2/3 & 4/3 & -2/3 & 0 \\ -2/3 & -2/3 & 4/3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.89}$$

The missing piece is the relation between strain rate $\dot{\varepsilon}_j$ and velocity field v_k, which is

$$[\dot{\varepsilon}_{xx}\ \dot{\varepsilon}_{yy}\ \dot{\varepsilon}_{zz}\ 2\dot{\varepsilon}_{xy}]^T = \left[\frac{\partial v_x}{\partial x} \quad \frac{\partial v_y}{\partial y} \quad 0 \quad \left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right)\right]^T \tag{2.90}$$

The strain rate-velocity matrix $\mathbf{B}$ is introduced (in a displacement formulation $\mathbf{B}$ is also known as strain-displacement matrix), such that

$$\dot{\varepsilon}_j = B_{jk} v_k \tag{2.91}$$

and therefore

$$B_{jk} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ 0 & 0 \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \tag{2.92}$$

Finally, combining eq. 2.83 with eqs. 2.87, 2.88 (the mapping to Voigt notation) and inserting eqs. 2.89, 2.91 and 2.92, we get an expression for the deviatoric stress increment Δs_i as a function of velocity v_k

$$\Delta s_i = H_{ij} B_{jk} v_k \tag{2.93}$$

Note that to obtain the stress in Voigt notation σ_i, the pressure p must be added to the normal components of the deviatoric stress s_i as follows:

$$\boldsymbol{\sigma} = \boldsymbol{s} + p[1\ 1\ 1\ 0]^T \tag{2.94}$$

Axisymmetry

To assume axisymmetry, cylindrical coordinates are used according to fig. 2.6. The previous x and y axes become r and z respectively, with the z-axis being the axis of symmetry.

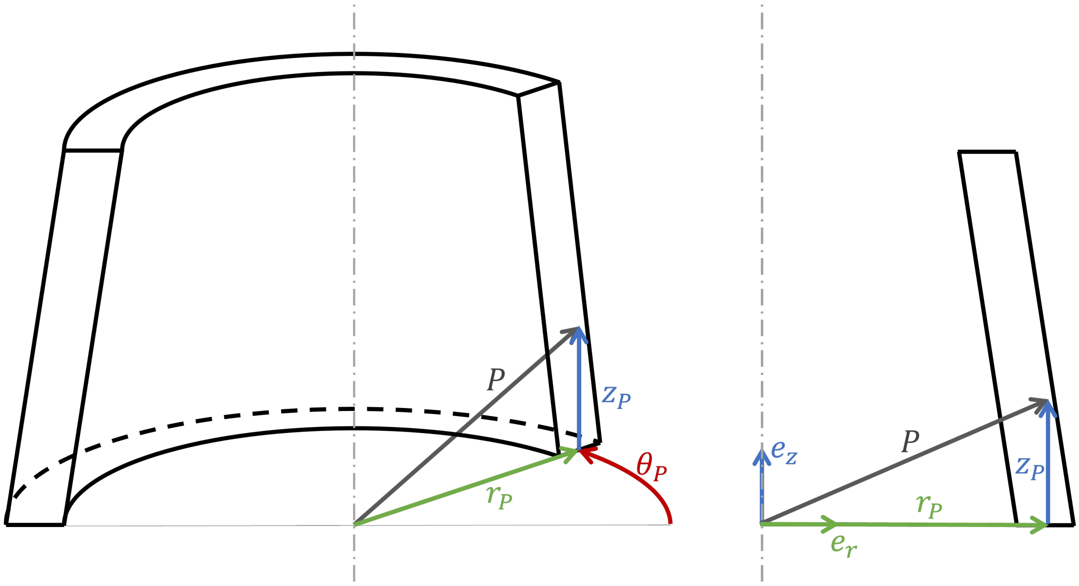


FIGURE 2.6: Cylindrical coordinate system. Left: axisymmetric object in 3D view. Right: Projection of the object onto the r-z-plane. Note that the axis of symmetry is always at $r = 0$.

Then, the stress and strain components can be renamed in accordance with the change in coordinate system

$$\Delta \boldsymbol{s}_{pl.strain} = [\Delta s_{xx}\ \Delta s_{yy}\ \Delta s_{zz}\ \Delta s_{xy}]^T \rightarrow \Delta \boldsymbol{s}_{axisymm} = [\Delta s_{rr}\ \Delta s_{zz}\ \Delta s_{\theta\theta}\ \Delta s_{rz}]^T \tag{2.95}$$

$$\dot{\boldsymbol{\varepsilon}}_{pl.strain} = [\dot{\varepsilon}_{xx}\ \dot{\varepsilon}_{yy}\ \dot{\varepsilon}_{zz}\ 2\dot{\varepsilon}_{xy}]^T \rightarrow \dot{\boldsymbol{\varepsilon}}_{axisymm} = [\dot{\varepsilon}_{rr}\ \dot{\varepsilon}_{zz}\ \dot{\varepsilon}_{\theta\theta}\ 2\dot{\varepsilon}_{rz}]^T \tag{2.96}$$

Note that both the hoop strain rate $\dot{\varepsilon}_{\theta\theta}$ and the hoop stress $\sigma_{\theta\theta}$ can be non-zero despite the absence of all out-of-plane displacement as $v_\theta = 0$. The stress-strain relation in eq. 2.89 remains the same within the axisymmetric framework. The most important difference lies in the strain rate-velocity relation, where the hoop strains are induced by the radial velocity v_r:

$$[\dot{\varepsilon}_{rr}\ \dot{\varepsilon}_{zz}\ \dot{\varepsilon}_{\theta\theta}\ 2\dot{\varepsilon}_{rz}]^T = \left[\frac{\partial v_r}{\partial r} \quad \frac{\partial v_z}{\partial z} \quad \frac{v_r}{r} \quad \left(\frac{\partial v_r}{\partial z} + \frac{\partial v_z}{\partial r}\right)\right]^T \tag{2.97}$$

The corresponding strain rate-velocity matrix $\mathbf{B}$ becomes

$$B_{jk} = \begin{bmatrix} \frac{\partial}{\partial r} & 0 \\ 0 & \frac{\partial}{\partial z} \\ \frac{1}{r} & 0 \\ \frac{\partial}{\partial z} & \frac{\partial}{\partial r} \end{bmatrix} \tag{2.98}$$

All other equations remain the same, when compared to the plane strain formulation.

2.5 Finite element formulation

We solve for the solution variables velocity $v_i(\boldsymbol{x}, t)$, pressure $p(\boldsymbol{x}, t)$ and temperature $T(\boldsymbol{x}, t)$ using the governing equations, i.e. the conservation of mass, momentum and energy (eqs. 2.41, 2.55, 2.34) and the constitutive equation 2.79 written in Voigt notation according to section 2.4.4. v_i, p and T will be discretized as primary unknowns and the nodal values for those quantities are obtained with the finite element method (FEM) for the solid, semi-solid and liquid state simultaneously.

2.5.1 Weak form

The conservation equations are given in the strong form and the first step is to convert them into the weak form. This is done by multiplying each equation with an appropriate trial function and integrating over the entire domain Ω. We define the spaces for trial and test functions:

$$\boldsymbol{v} \in \mathcal{S}, \qquad \mathcal{S} = \left\{ \boldsymbol{v} \in \boldsymbol{H}^1(\Omega) \mid \boldsymbol{v} = \bar{\boldsymbol{v}} \text{ on } \Gamma_v \right\} \tag{2.99}$$

$$\tilde{\boldsymbol{w}} \in \mathcal{S}_0, \qquad \mathcal{S}_0 = \left\{ \tilde{\boldsymbol{w}} \in \boldsymbol{H}^1(\Omega) \mid \tilde{\boldsymbol{w}} = 0 \text{ on } \Gamma_v \right\} \tag{2.100}$$

$$p \in \mathcal{Q}, \qquad \mathcal{Q} = \left\{ \tilde{q} \in L^2(\Omega) \right\}^{4} \tag{2.101}$$

$$T \in \mathcal{R}, \qquad \mathcal{R} = \left\{ T \in H^1(\Omega) \mid T = \bar{T} \text{ on } \Gamma_T \right\} \tag{2.102}$$

$$\tilde{r} \in \mathcal{R}_0, \quad \mathcal{R}_0 = \left\{ \tilde{r} \in H^1(\Omega) \mid \tilde{r} = 0 \text{ on } \Gamma_T \right\} \tag{2.103}$$

where $\mathcal{S}$ is the space of vector trial functions for the velocity $\boldsymbol{v}$, $\mathcal{S}_0$ is the space of vector test functions for the velocity $\tilde{\boldsymbol{w}}$, $\mathcal{Q}$ is the space of scalar trial and test functions for the pressure p, $\tilde{q}$ is a scalar test function for the pressure, $\mathcal{R}$ is the space of scalar trial functions for the temperature, $\tilde{r}$ is a scalar test function for the temperature, $\boldsymbol{H}^1(\Omega)$ and $H^1(\Omega)$ denote the space of square integrable functions with square integrable first derivative for vectors and scalars respectively and $L^2(\Omega)$ denotes the space of square integrable functions over the domain Ω.

[4] One may find that there is an asymmetry in the definition of the function spaces, since the function space for the pressure trial and test functions $\mathcal{Q}$ is not restricted to respect any kind of essential boundary condition for the pressure. A paragraph is section 2.5.5 will comment on this.

The multiplication of the conservation equations (eqs. 2.41, 2.14, 2.34) with each respective test function and subsequent integration over the current domain yields the weak form

$$\int\limits_{\Omega} \tilde{q}\left(\frac{1}{K}\dot{p} + \alpha_V \dot{T} + \frac{\partial v_i}{\partial x_i}\right) dV = 0 \qquad \forall \tilde{q} \in \mathcal{Q} \tag{2.104}$$

$$\int\limits_{\Omega} \tilde{w}_i\left(\rho \dot{v}_i - \frac{\partial \sigma_{ij}}{\partial x_j} - \rho b_i\right) dV = 0 \qquad \forall \tilde{w}_i \in \mathcal{S}_0 \tag{2.105}$$

$$\int\limits_{\Omega} \tilde{r}\left(\rho c_p \dot{T} - \frac{\partial q_i}{\partial x_i} - R\right) dV = 0 \qquad \forall \tilde{r} \in \mathcal{R} \tag{2.106}$$

Notice that we keep the original momentum equation (eq. 2.14) here and not perform the operations that lead to eqs. 2.53 and 2.55 yet. The next step is to perform an integration by parts on the stress term in eq. 2.105 and the heat flux term in eq. 2.106.

$$\int\limits_{\Omega} \left(\rho \tilde{w}_i \dot{v}_i + \frac{\partial \tilde{w}_i}{\partial x_j}\sigma_{ij} - \rho \tilde{w}_i b_i\right) dV - \int\limits_{\Gamma} \tilde{w}_i \sigma_{ij} n_j \, dS = 0 \qquad \forall \tilde{w}_i \in \mathcal{S}_0 \tag{2.107}$$

$$\int\limits_{\Omega} \left(\tilde{r} \rho c_p \dot{T} + \frac{\partial \tilde{r}}{\partial x_i} q_i - \tilde{r}\, R\right) dV - \int\limits_{\Gamma} \tilde{r} q_i n_i \, dS = 0 \qquad \forall \tilde{r} \in \mathcal{R} \tag{2.108}$$

where S is the surface of the boundary and the surface integral in eqs. 2.107 and 2.108 is where the Neumann boundary condition is to be inserted. Because these terms are zero where $\Gamma \cap \Gamma_v$ and $\Gamma \cup \Gamma_T$, we can limit the surface integration to the portion of the surface that has a Neumann boundary condition and set $\Gamma = \Gamma_t$ and $\Gamma = \Gamma_q$ respectively. Furthermore, we can now replace the stress σ_{ij} by $-p\delta_{ij} + \Delta s_{ij} + s_{0_{ij}}$ in the volume integral of the momentum equation, while leaving σ_{ij} in the surface integral. This is possible because $\sigma_{ij} n_j = \bar{t}_i$ in the integral over Γ is a known quantity, being the Neumann boundary condition. The same is true for the second term in eq. 2.108, where $q_i n_i = \bar{q}$. Therefore we can write

$$\int\limits_{\Omega} \left(\rho \tilde{w}_i \dot{v}_i + \frac{\partial \tilde{w}_i}{\partial x_i} p + \frac{\partial \tilde{w}_i}{\partial x_j}\left(\Delta s_{ij} + s_{0_{ij}}\right) - \rho \tilde{w}_i b_i\right) dV - \int\limits_{\Gamma_t} \tilde{w}_i \bar{t} \, dS = 0 \qquad \forall \tilde{w}_i \in \mathcal{S}_0 \tag{2.109}$$

$$\int\limits_{\Omega} \left(\tilde{r} \rho c_p \dot{T} + \frac{\partial \tilde{r}}{\partial x_i} q_i - \tilde{r}\, R\right) dV - \int\limits_{\Gamma_q} \tilde{r} \bar{q} \, dS = 0 \qquad \forall \tilde{r} \in \mathcal{R} \tag{2.110}$$

The weak form is mathematically equivalent to the strong form, provided that σ_{ij} and q_i are continuous functions. The weak form has a reduced order of the spatial derivatives of σ_{ij} and q_i, making it much easier to solve. The trade-off is that we now only work with domain-averaged stresses instead of a detailed local stress field.

2.5.2 Discretization in time

The evolution of the velocity, pressure and temperature fields in the domain over time is to be simulated. In order to simplify integrals and derivatives over time, we progress through time in discrete time steps of length $\Delta t = t_n - t_{n-1}$, where n refers to the current time step and $n-1$ refers to the previous time step. This allows us to replace time derivatives by an approximation. We choose the backward Euler implicit scheme:

$$\dot{\phi}(t) \approx \frac{\phi^n - \phi^{n-1}}{\Delta t} \tag{2.111}$$

where ϕ is any variable. Here, the current configuration is explicitly marked by the superscript n, but for better readability, the current configuration may be without any indication of the time step, while the reference condition will always be marked by a superscript $n-1$. The reference condition refers to the state at the end of the previous time step, which is also the state at the beginning of the current time step.

We use the relation in eq. 2.111 to replace the time derivatives in eqs. 2.104, 2.109 and 2.110.

$$\int\limits_{\Omega} \left(\tilde{q} \frac{1}{K} \frac{p^n - p^{n-1}}{\Delta t} + \tilde{q}\, \alpha \frac{T^n - T^{n-1}}{\Delta t} + \tilde{q} \frac{\partial v_i^n}{\partial x_i} \right) dV = 0 \qquad \forall \tilde{q} \in \mathcal{Q} \tag{2.112}$$

$$\int\limits_{\Omega} \left(\rho \tilde{w}_i \frac{v_i^n - v_i^{n-1}}{\Delta t} - \frac{\partial \tilde{w}_i}{\partial x_i} p^n + \frac{\partial \tilde{w}_i}{\partial x_j} \left(\Delta s_{ij}^n + s_{ij}^{n-1} \right) - \rho \tilde{w}_i b_i^n \right) dV - \int\limits_{\Gamma_t} \tilde{w}_i \bar{t}\, dS = 0 \qquad \forall \tilde{w}_i \in \mathcal{S}_0 \tag{2.113}$$

$$\int\limits_{\Omega} \left(\tilde{r} \rho c_p \frac{T^n - T^{n-1}}{\Delta t} + \frac{\partial \tilde{r}}{\partial x_i} q_i^n - \tilde{r}\, R^n \right) dV - \int\limits_{\Gamma_q} \tilde{r} \bar{q}^n \, dS = 0 \qquad \forall \tilde{r} \in \mathcal{R} \tag{2.114}$$

To integrate ϕ over time we rearrange eq. 2.111:

$$\phi^n \approx \dot{\phi}^n \, \Delta t + \phi^{n-1} \tag{2.115}$$

The most prominent use of eq. 2.115 is to determine the current position of the nodes as

$$x_i^n = v_i^n \Delta t + x_i^{n-1} \tag{2.116}$$

Deviatoric stress increments (in Voigt notation) Δs_i are the rate of change integrated over an arbitrary time interval $\Delta t = t - t_0$, as described by eq. 2.68.

What was previously referred to as an "arbitrary time interval" now becomes a time step and the reference deviatoric stress s_i^{n-1} is updated to represent the state at the beginning of that time step. It is initialized across the entire domain as $s_i^0 = 0$ and at the end of each time step, the current deviatoric stress s_i^n is:

$$s_i^n = s_i^{n-1} + \Delta s_i \tag{2.117}$$

and at the beginning of a time step we set

$$s_i^{n-1} = \begin{cases} s_i^{n-1} & \text{if fully solid } (f_l = 0.0) \\ 0 & \text{otherwise} \end{cases} \tag{2.118}$$

From section 2.54, we recall that the incremental form is equivalent to the time continuous form, if $\Delta \boldsymbol{s}$ is equivalent to the time integral of $\boldsymbol{s}$. This is would be the case, if the strain rate $\dot{\varepsilon}$ is constant during each time step, which is rarely the case. A good approximation is however achieved if the time step is chosen small enough.

2.5.3 Discretization in space

To avoid solving the complex equations over a complex domain shape, the domain is discretized into elements of simple shape, triangles in this case.

Shape functions

The continuous velocity, pressure and temperature fields are replaced by an approximation using linear shape functions and nodal velocities, pressures and temperatures:

$$v_i(\boldsymbol{x}, t) \approx N_I(\boldsymbol{x}) v_{i,I}(t^n) \tag{2.119}$$

$$p(\boldsymbol{x}, t) \approx N_I(\boldsymbol{x}) p_I(t^n) \tag{2.120}$$

$$T(\boldsymbol{x}, t) \approx N_I(\boldsymbol{x}) T_I(t^n) \tag{2.121}$$

where the subscript $_I$ refers to a value at node I or a function associated with node I. $\boldsymbol{N}$ is the vector of shape functions. It contains a linear shape function N_I for each node I such that $N_I(\boldsymbol{x}_J) = \delta_{IJ}$ and a linear distribution from 1 to 0 between node I and its direct neighboring nodes that are part of the same element. The vector $\boldsymbol{N}$ for a linear triangle (i.e. with 3 nodes) is

$$\boldsymbol{N} = [N_1 \; N_2 \; N_3]^T \tag{2.122}$$

Isoparametric elements

To avoid adapting interpolation and integration operations to each element's shape and orientation, each triangle is mapped from the current physical space

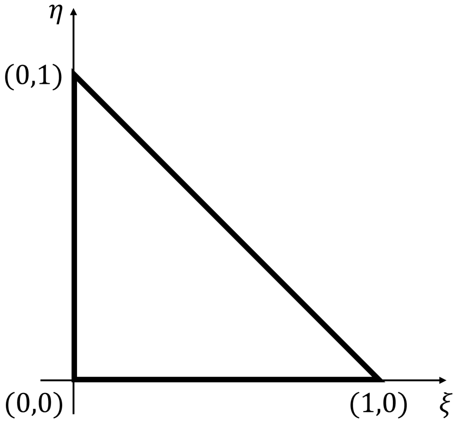


FIGURE 2.7: Parent triangle.

(x, y) back into a standardized parent triangle in the parent space ξ, η. Interpolation and integration operations can be executed uniformly on this standardized triangle. The transformation matrix is referred to as Jacobian $\mathbf{J}$

$$\mathbf{J} = \begin{bmatrix} \frac{\partial N_I}{\partial \xi} x_I & \frac{\partial N_I}{\partial \xi} y_I \\ \frac{\partial N_I}{\partial \eta} x_I & \frac{\partial N_I}{\partial \eta} y_I \end{bmatrix} \tag{2.123}$$

The integration of any variable ϕ over a linear triangle element then becomes

$$\iint\limits_A \phi(x, y) dy dx = \int\limits_0^1 \int\limits_0^{1-\xi} \phi(\xi, \eta) \det(\mathbf{J}) \, d\eta \, d\xi \tag{2.124}$$

To perform the integration over the element, we use the Gauss-Legendre method for isoparametric triangles.

$$\int\limits_0^1 \int\limits_0^{1-\xi} \phi(\xi, \eta) \det(\mathbf{J}) \, d\eta \, d\xi \approx \sum_{i=1}^{n_{GP}} w_i \phi(\xi_i, \eta_i) \det(\mathbf{J}) \tag{2.125}$$

where n_{GP} is the number of Gauss points, w_i is the weighting factor of Gauss point i. For a given number of Gauss points, the weights w_i and positions (ξ_i, η_i) of each Gauss point can be found in the literature (e.g. [87]). The deciding advantage of combining the Gauss-Legendre method with the isoparametric elements is that the Gauss points are always located in the same ξ, η location and the values of the shape functions and the shape function derivatives are the same for all elements. This makes the computer code very efficient, as integrating in space over any element becomes very simple. For the sake of keeping the equations less convoluted, we continue to write the integral over x, y of the domain Ω, even though the Gauss-Legendre method and the transformation into the parent space is always actually used.

For an integration in the **reference configuration**, the triangle is instead mapped back from the reference physical space (x_0, y_0) to the parent space ξ, η.

The corresponding Jacobian matrix is symbolized as $\mathbf{J}_0$ and the integration in the reference configuration is written as

$$\iint\limits_{A_0} \phi(x_0, y_0) dy_0 dx_0 = \int\limits_0^1 \int\limits_0^{1-\xi} \phi(\xi, \eta) \det(\mathbf{J}_0) d\eta \, d\xi \approx \sum_{i=1}^{n_{GP}} w_i \phi(\xi_i, \eta_i) \det(\mathbf{J}_0) \tag{2.126}$$

where A_0 is the area of the element in its reference configuration. How the reference configuration is defined, depends on what needs to be achieved. In this work, the Updated Lagrangian framework (i.e. solving the equations in the current configuration) is used for the non-solid elements, which means that the integration is performed on the current configuration. It is equally valid to use an older state as the reference configuration in a Lagrangian framework. If this older state is the initial state, the framework is referred to as Total Lagrangian, but this is not used in this work. Instead, because the solid elements are assumed to obey the small displacement assumption, it is valid to perform the integration in the reference configuration. The reference configuration of a solid element is the configuration at which it first became solid. Solid elements do not need to be updated anymore as a consequence. This is discussed further in section 2.6.1.

For line elements of any length and orientation in the current configuration or reference configuration, a similar mapping is performed. The isoparametric line in the parent space is one-dimensional and has only one spatial variable ξ which runs from -1 to 1. This means that the length l of a line in the current physical space $l = \det(\mathbf{J})/2$ and $l = \det(\mathbf{J}_0)/2$ in the reference physical space. Otherwise, the integration and the mapping are analogous to the triangle.

Integration in the third direction (plane strain & axisymmetric assumptions)

The above integrals are over the 2D area of an element. The third dimension is taken into account depending on the assumption made about the third dimension. In the 2D plane strain framework, a constant thickness $h = 1$ is assumed which makes

$$\int\limits_{\Omega} \phi \, dV = h \int\limits_x \int\limits_y \phi \, dy \, dx \tag{2.127}$$

$$\int\limits_{\Gamma} \phi \, dS = h \int\limits_{\gamma} \phi \, ds \tag{2.128}$$

where γ is the boundary contour in the 2D plane and s is the local coordinate along γ. h can be outside the integral because the thickness does not change in plane strain, which is not the case in the plane stress and axisymmetric frameworks. In the latter case the integrals become

$$\int\limits_{\Omega} \phi \, dV = 2\pi \int\limits_{z} \int\limits_{r} r\phi \, dr \, dz \tag{2.129}$$

$$\int\limits_{\Gamma} \phi \, dS = 2\pi \int\limits_{\gamma} r\phi \, ds \tag{2.130}$$

where r is the radius. Of course the integrals over an area or a curve are then evaluated using the Gauss-Legendre method described above.

Discrete system of equations

Now we can discretize the conservation equations. All field variables v_i, p and T as well as their respective test function fields $\tilde{w}_i$, $\tilde{q}$ and $\tilde{r}$ are replaced by shape functions and nodal values. The conservation equations are written in the discrete form as:

$$\mathbf{A}_{mech} \cdot \boldsymbol{q}_{mech} = \boldsymbol{r}_{mech} \tag{2.131}$$

and

$$\mathbf{A}_{th} \cdot \boldsymbol{q}_{th} = \boldsymbol{r}_{th} \tag{2.132}$$

where subscripts $_{mech}$ and $_{th}$ refer to the two systems of equation for the mechanical part (mass conservation and momentum conservation) and the thermal part (heat equation). $\mathbf{A}$ is the system matrix, $\boldsymbol{q}$ is the solution vector of nodal values and $\boldsymbol{r}$ is the right hand side. For the monolithic system of momentum and continuity equations $\boldsymbol{q}_{mech}$ and $\boldsymbol{r}_{mech}$ are composed as follows:

$$\boldsymbol{q}_{mech} = [v_{x,1} \cdots v_{x,I} \cdots v_{x,n_n} \quad v_{y,1} \cdots v_{y,I} \cdots v_{y,n_n} \quad p_1 \ldots p_I \cdots p_{n_n}]^T \tag{2.133}$$

$$\boldsymbol{r}_{mech} = [r_{x,1} \cdots r_{x,I} \cdots r_{x,n_n} \quad r_{y,1} \cdots r_{y,I} \cdots r_{y,n_n} \quad r_{p,1} \cdots r_{p,I} \cdots r_{p,n_n}]^T \tag{2.134}$$

where n_n is the number of nodes, r_x, r_y and r_p are contributions to x and y momentum equations and mass conservation equation that are not functions of the current solution variables v_x, v_y, p. For example, r_x and r_y contain external forces, body forces and source terms acting in x and y direction respectively, while r_p contains contributions to the volume expansion rate or the reference pressure.

For the heat equation $\boldsymbol{q}_{th}$ and $\boldsymbol{r}_{th}$ contain

$$\boldsymbol{q}_{th} = [T_1 \cdots T_I \cdots T_{n_n}]^T \tag{2.135}$$

$$\boldsymbol{r}_{th} = [r_{q,1} \cdots r_{q,I} \cdots r_{q,n_n}]^T \tag{2.136}$$

where r_q contains external heat fluxes and external heat sources and any term that is not a function of T. Both solution vectors in eqs. 2.133 and 2.135 have their test function counterpart that is structured in an analogous way, containing $\tilde{w}_i$, $\tilde{q}$ and $\tilde{r}$. The eqs. 2.112, 2.113 and 2.114 must be true for any choice of test function. Therefore, the vectors of nodal values of the test functions can be eliminated from each respective equation. The result is the discrete system of equations. The global coupled velocity-pressure system becomes

$$\frac{1}{\Delta t}\mathbf{M}\cdot\boldsymbol{v}+\mathbf{K}_{dev}\cdot\boldsymbol{v}+\mathbf{D}^{T}\cdot\boldsymbol{p}=\underbrace{\boldsymbol{b}+\boldsymbol{t}+\frac{1}{\Delta t}\mathbf{M}\cdot\boldsymbol{v}^{n-1}+\boldsymbol{k}_{dev}}_{[\boldsymbol{r}_x\,\boldsymbol{r}_y]^T} \tag{2.137}$$

$$\mathbf{D}\cdot\boldsymbol{v}+\frac{1}{\Delta t}\mathbf{K}_{vol}\cdot\boldsymbol{p}=\underbrace{\boldsymbol{a}+\frac{1}{\Delta t}\mathbf{K}_{vol}\cdot\boldsymbol{p}^{n-1}}_{\boldsymbol{r}_p} \tag{2.138}$$

With the terms from eqs. 2.137 and 2.138, the system of equations of the mechanical problem (eq. 2.131) can also be written in a more graphic representation as

$$\underbrace{\begin{bmatrix}\begin{bmatrix}\frac{1}{\Delta t}\mathbf{M}+\mathbf{K}_{dev}\end{bmatrix} & \begin{bmatrix}\mathbf{D}^{T}\end{bmatrix}\\ \begin{bmatrix}\mathbf{D}\end{bmatrix} & \begin{bmatrix}\frac{1}{\Delta t}\mathbf{K}_{vol}\end{bmatrix}\end{bmatrix}}_{\mathbf{A}_{mech}}\cdot\underbrace{\begin{bmatrix}\begin{bmatrix}\boldsymbol{v}_x\end{bmatrix}\\ \begin{bmatrix}\boldsymbol{v}_y\end{bmatrix}\\ \begin{bmatrix}\boldsymbol{p}\end{bmatrix}\end{bmatrix}}_{\boldsymbol{q}_{mech}}=\underbrace{\begin{bmatrix}\begin{bmatrix}\boldsymbol{r}_x\end{bmatrix}\\ \begin{bmatrix}\boldsymbol{r}_y\end{bmatrix}\\ \begin{bmatrix}\boldsymbol{r}_p\end{bmatrix}\end{bmatrix}}_{\boldsymbol{r}_{mech}} \tag{2.139}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{K}_{dev}$ is the stiffness matrix related to deviatoric stresses, $\mathbf{K}_{vol}$ is the stiffness matrix related to volumetric stresses, $\mathbf{D}$ is the discrete divergence operator, $\boldsymbol{b}$ is the body force vector, $\boldsymbol{t}$ is the vector resulting from the surface traction at the Neumann boundaries, $\boldsymbol{k}_{dev}$ is the reference deviatoric stress vector and $\mathbf{a}$ is the thermal expansion vector. All these structural (or global) matrices and vectors are assembled from elemental matrices and vectors (subscript $_e$), which are defined as follows

$$\mathbf{M}_{i_e} = \int_\Omega \rho \, N_I N_J \, dV \tag{2.140}$$

$$\mathbf{M}_e = \begin{bmatrix} \mathbf{M}_{i_e} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{i_e} \end{bmatrix} \tag{2.141}$$

$$\mathbf{K}_{dev_e} = \int_\Omega \mathbf{B}^T \cdot \mathbf{H_{dev}} \cdot \mathbf{B} \, dV \tag{2.142}$$

$$\mathbf{K}_{vol_e} = \int_\Omega \frac{1}{K} \, N_I N_J \, dV \tag{2.143}$$

$$\boldsymbol{b}_e = \int_\Omega \rho \boldsymbol{g} \, [N_I N_I]^T \, dV \tag{2.144}$$

$$\boldsymbol{t}_e = \int_{\Gamma_t} \bar{t} \, [N_I N_I]^T \, dS \tag{2.145}$$

$$\boldsymbol{a}_e = \int_\Omega \alpha_V \frac{T^n - T^{n-1}}{\Delta t} \, N_I N_J \, dV \tag{2.146}$$

$$\boldsymbol{k}_{dev_e} = \int_\Omega \mathbf{B}^T \cdot \boldsymbol{s}^{n-1} \, dV \tag{2.147}$$

where $N_I = [N_1 \; N_2 \; N_3]^T$. Furthermore, we have the discrete element strain rate-velocity matrix $\mathbf{B}_e$ and the discrete element divergence operator $\mathbf{D}_e$ to be used **in plane strain**:

$$\mathbf{B}_e = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial y} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} \end{bmatrix} \tag{2.148}$$

$$\mathbf{D}_e = \int_\Omega N_I \nabla N_J \, dV \tag{2.149}$$

with an element gradient vector

$$\nabla \boldsymbol{N} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} & \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial y} \end{bmatrix} \tag{2.150}$$

and $\mathbf{B}_e$ and $\mathbf{D}_e$ to be used **in axisymmetry**

$$\mathbf{B}_e = \begin{bmatrix} \frac{\partial N_1}{\partial r} & \frac{\partial N_2}{\partial r} & \frac{\partial N_3}{\partial r} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial N_1}{\partial z} & \frac{\partial N_2}{\partial z} & \frac{\partial N_3}{\partial z} \\ \frac{N_1}{r} & \frac{N_2}{r} & \frac{N_3}{r} & 0 & 0 & 0 \\ \frac{\partial N_1}{\partial z} & \frac{\partial N_2}{\partial z} & \frac{\partial N_3}{\partial z} & \frac{\partial N_1}{\partial r} & \frac{\partial N_2}{\partial r} & \frac{\partial N_3}{\partial r} \end{bmatrix} \tag{2.151}$$

$$\mathbf{D}_e = \int\limits_{\Omega} N_{axi,IK} \nabla N_{axi,KJ} \, dV \tag{2.152}$$

with an element gradient matrix $\nabla \mathbf{N}_{axi}$

$$\nabla \mathbf{N}_{axi} = \begin{bmatrix} \frac{\partial N_1}{\partial r} & \frac{\partial N_2}{\partial r} & \frac{\partial N_3}{\partial r} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial N_1}{\partial z} & \frac{\partial N_2}{\partial z} & \frac{\partial N_3}{\partial z} \\ \frac{N_1}{r} & \frac{N_2}{r} & \frac{N_3}{r} & 0 & 0 & 0 \end{bmatrix} \tag{2.153}$$

and a shape function matrix $\mathbf{N}_{axi}$

$$\mathbf{N}_{axi} = \begin{bmatrix} N_1 & N_1 & N_1 \\ N_2 & N_2 & N_2 \\ N_3 & N_3 & N_3 \end{bmatrix} \tag{2.154}$$

Note that at the end of a time step, the stress (in Voigt notation) is updated according to eqs. 2.118 and 2.94 at each Gauss point and kept in memory. $\boldsymbol{s}^{n-1}$ in eq. 2.147 can then be evaluated at the beginning of the next time step, using eq. 2.117.

In the current configuration, the shape function gradients $\partial N_I / \partial x_i$ are calculated according to

$$\frac{\partial N_I}{\partial x_i} = \frac{\partial N_I}{\partial \xi} \frac{\partial \xi}{\partial x_i} + \frac{\partial N_I}{\partial \eta} \frac{\partial \eta}{\partial x_i} \tag{2.155}$$

whereas in the reference configuration, all x_i in eqs. 2.148, 2.150, 2.151, 2.153 and 2.155 are replaced by x_{0_i}. As explained in section 2.5.3, this approach is used for solid elements and their reference configuration is the configuration at the end of the time step in which they become solid.

For the discrete heat equation, which will be solved separately (see description of staggered scheme in section 2.6), the linear system is

$$\frac{1}{\Delta t} \mathbf{C} \cdot \boldsymbol{T} - \mathbf{K}_{cond} \cdot \boldsymbol{T} = \underbrace{\boldsymbol{s}_R + \boldsymbol{s}_L + \boldsymbol{s}_q + \frac{1}{\Delta t} \mathbf{C} \cdot \boldsymbol{T}_0}_{r_q} \tag{2.156}$$

where $\mathbf{C}$ is the heat capacity matrix, $\mathbf{K}_{cond}$ is the conductivity matrix, $\boldsymbol{s}_R$ is the heat source vector, $\boldsymbol{s}_L$ is the latent heat absorption vector and $\boldsymbol{s}_q$ is the vector of surface heat fluxes at the Neumann boundaries.
With the individual terms from eq. 2.156, the system of equations of the thermal problem (eq. 2.131) can be written in a more graphic representation as

$$\underbrace{\begin{bmatrix} \\ \frac{1}{\Delta t}\mathbf{C} + \mathbf{K}_{cond} \\ \\ \end{bmatrix}}_{\mathbf{A}_{th}} \cdot \underbrace{\begin{bmatrix} \\ \boldsymbol{T} \\ \\ \end{bmatrix}}_{\boldsymbol{q}_{th}} = \underbrace{\begin{bmatrix} \\ \boldsymbol{r}_q \\ \\ \end{bmatrix}}_{\boldsymbol{r}_{th}} \tag{2.157}$$

Again, the structural matrices and vectors are assembled from the following elemental matrices and vectors:

$$\mathbf{C} = \int\limits_{\Omega} \rho c_p \, N_I N_J \, dV \tag{2.158}$$

$$\mathbf{K}_{cond} = \int\limits_{\Omega} \lambda \, \nabla N_I \nabla N_J \, dV \tag{2.159}$$

$$\boldsymbol{s}_R = \int\limits_{\Omega} R \, N_I \, dV \tag{2.160}$$

$$\boldsymbol{s}_L = -\int\limits_{\Omega} \rho \frac{\Delta f_l}{\Delta t} L_m \, N_I \, dV \tag{2.161}$$

$$\boldsymbol{s}_q = \int\limits_{\Gamma_q} \bar{q} \, N_I \, dS \tag{2.162}$$

where the boundary heat flux $\bar{q}$ is composed of the contributions from eqs. 2.30, 2.31 and 2.32:

$$\bar{q} = q_{lp} + q_{conv} + q_{rad} \tag{2.163}$$

Choosing the number of Gauss points

Triangles normally use three Gauss points (2nd order), unless the user chooses to lump the mass matrix **M**, the stiffness matrix related to deviatoric stresses $\mathbf{K}_{vol}$ and the heat capacity matrix **C**. These matrices are bilinear in their consistent form and require 2nd order triangles for their correct integration. When lumped, the requirement reduces to one Gauss point (1st order) in plane strain and allowing a reduction of the cost at the expense of possibly some accuracy. Under the axisymmetry assumption, 2nd order elements are still required, since some terms in the stiffness matrix related to deviatoric stresses $\mathbf{K}_{dev}$ become very large and cause numerical problems according to Cerquaglia [39].

Computation of the surface curvature

As discussed in section 2.3.3, the normal component of the surface tension exerts a force (more precisely the Laplace pressure) onto the surface as a function of the curvature and the material pairing. To determine the curvature on the discretized surface in 2D, only a few simple steps suffice:

1. for each surface node I, find both neighbor nodes $J = 1, 2$

2. construct a circle with radius vector $\boldsymbol{r}_1$ from the 3 nodes
3. calculate curvature according to eq. 2.43

Note that the outward surface normal vector $\boldsymbol{n}_I$ is the radius vector $\boldsymbol{r}_1$ after normalization and possible inversion, if pointing inwards.

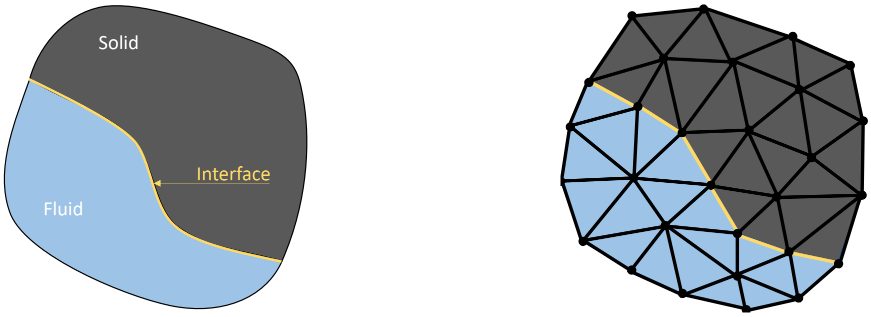
2.5.4 Stabilization for incompressible media

To prevent the violation of the Ladyzhenskaya–Babuška–Brezzi (LBB) condition [88], a stabilization is required, when the medium is (quasi-) incompressible ($K \to \infty$). In our implementation, we use the Pressure-Stabilizing Petrov-Galerkin (PSPG) method developed by Tezduyar et al. [89] and introduced in our code by Cerquaglia [90]. The equations that the stabilization introduces are explained in detail and recommendations regarding some user-chosen parameters are given therein. The stabilization is deactivated for all elements that are mechanically compressible, which can be both solid and fluid elements. A short summary of the main steps is given in appendix A.

Note that many other stabilization methods exist and have been used in PFEM. A summary thereof and respective references can be found in section 3.2 of [38].

2.5.5 Fluid-solid interface

The unified fluid-solid formulation is the central development of this work. Its purpose is to allow fluid and solid materials to coexist and interact in a single solver. Without this unified formulation, two specialized solvers would need to be coupled. The latter type of coupling is a well established practice for fluid-structure interaction (FSI) problems and many examples can be found in the literature. For example, this PFEM code for fluids coupled with the FEM solver *Metafor* [91] for solids is one of the main subjects in Cerquaglia's doctoral thesis [9]. However, such a coupled approach is difficult to use for phase change problems due to the unknown evolution of the fluid-solid interface. An interface is therefore introduced to allow fluid and solid to coexists and interact in the PFEM framework.



(a) A continuum may have a sharp interface between solid and fluid.

(b) The discretized domain should reproduce the sharp interface.

FIGURE 2.8: Schematic of a fluid-solid interface.

The interface is the surface (or curve in 2D) of contact between the solid and fluid materials (yellow line in fig. 2.8a). After discretization, the interface will be represented by a series of edges that lie between fluid and solid elements (fig. 2.8b). Since edges are defined by their nodes, the interface can also be considered located along a series of nodes.

At the interface a no-slip condition must be enforced and the contact surface traction between fluid and solid must be equal and opposite. Due to the different role that the pressure plays in fluids and solids, special attention is brought to the treatment of the pressure at the interface in the following. This section then continues to explain how the aforementioned conditions are enforced in this work.

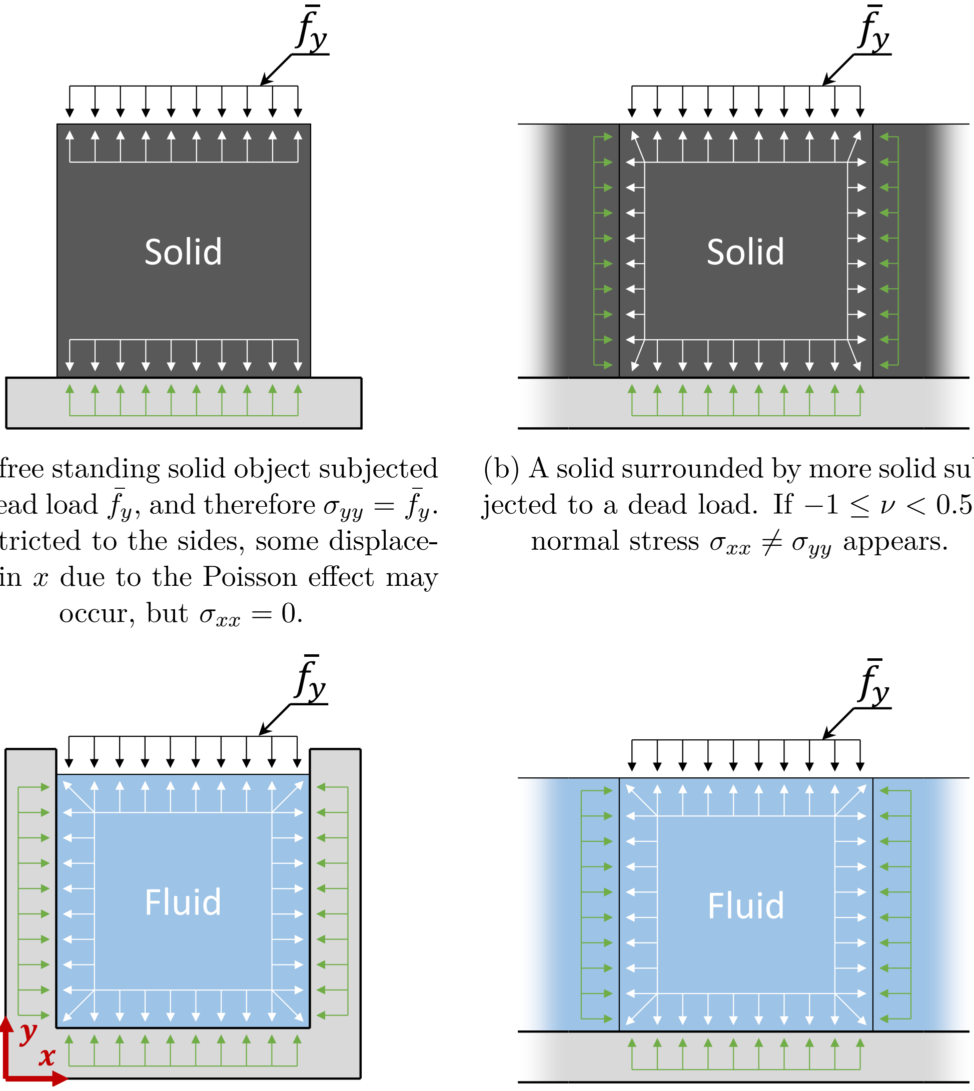
The role of pressure in solids and fluids

To better understand the requirement for an interface and how it can be implemented, we may first examine how the pressure appears differently in Newtonian fluids and elastic solids.

Stress is a quantification of the internal resistance force in a material, which arises as a response to external forces applied to the material. As discussed in sections 2.4, an **elastic solid** is characterized by a shear modulus G and a bulk modulus K, which both relate the stress to a strain. Specifically, G relates the deviatoric stress tensor $\mathrm{dev}(\boldsymbol{\sigma})$ to the deviatoric strain tensor $\mathrm{dev}(\boldsymbol{\varepsilon})$ and K relates the volumetric stress tensor $\mathrm{vol}(\boldsymbol{\sigma})$ to a volumetric strain tensor $\mathrm{vol}(\boldsymbol{\varepsilon})$. A solid object under a load therefore experiences a combination of deviatoric and volumetric stresses. From eq. 2.51 we recall that only the volumetric stress tensor is linked to the pressure ($\mathrm{vol}(\boldsymbol{\sigma}) = p\boldsymbol{I}$). Since $\mathrm{dev}(\boldsymbol{\sigma})$ is not linked to the pressure, it can be concluded that the more a given load can be borne by deviatoric stresses, the less must be borne by volumetric stresses and therefore a smaller change in pressure is observed in such a case. The change in pressure in an elastic solid therefore depends on how the load is distributed between deviatoric and volumetric stresses.

A **Newtonian fluid**, on the other hand, is characterized by a bulk modulus K ($K = \infty$ if incompressible) and a viscosity μ_f, which relates the deviatoric stress tensor $\mathrm{dev}(\boldsymbol{\sigma})$ to a deviatoric strain *rate* tensor $\mathrm{dev}(\dot{\boldsymbol{\varepsilon}})$. More importantly, a Newtonian fluid does not have a shear modulus. The effect is that for a Newtonian fluid at rest ($\dot{\boldsymbol{\varepsilon}} = 0$), any given stress as a response to external forces, must be entirely a volumetric stress. In other words, a Newtonian fluid at rest can only support an external load by increasing or decreasing its pressure and no deviatoric stresses can be observed. It also implies that all three normal stresses are equal ($\sigma_{xx} = \sigma_{yy} = \sigma_{zz} = p$). If the fluid is not at rest, some viscous (deviatoric) stresses appear, but they tend to be several orders of magnitude lower than p in the cases studied here, so the previous conclusion are nearly true also for fluids in motion.

The illustration in fig. 2.9 highlights these concepts, where internal forces (white arrows) occur due to external loading (black) and reaction forces (green). Comparing figs. 2.9a and 2.9b, we see how the solid reacts to two different loading cases and how the internal response is not required to be isotropic. This is



(a) A free standing solid object subjected to a dead load $\bar{f}_y$, and therefore $\sigma_{yy} = \bar{f}_y$. Unrestricted to the sides, some displacement in x due to the Poisson effect may occur, but $\sigma_{xx} = 0$.

(b) A solid surrounded by more solid subjected to a dead load. If $-1 \leq \nu < 0.5$ a normal stress $\sigma_{xx} \neq \sigma_{yy}$ appears.

(c) A fluid in a container subjected to a dead load. Internal forces act equally in all directions ($\sigma_{xx} = \sigma_{yy} = \sigma_{zz} = p$) and the container provides the reaction force. Without the container (as in (a), the fluid would flow away sideways due to the absence of a shear modulus.

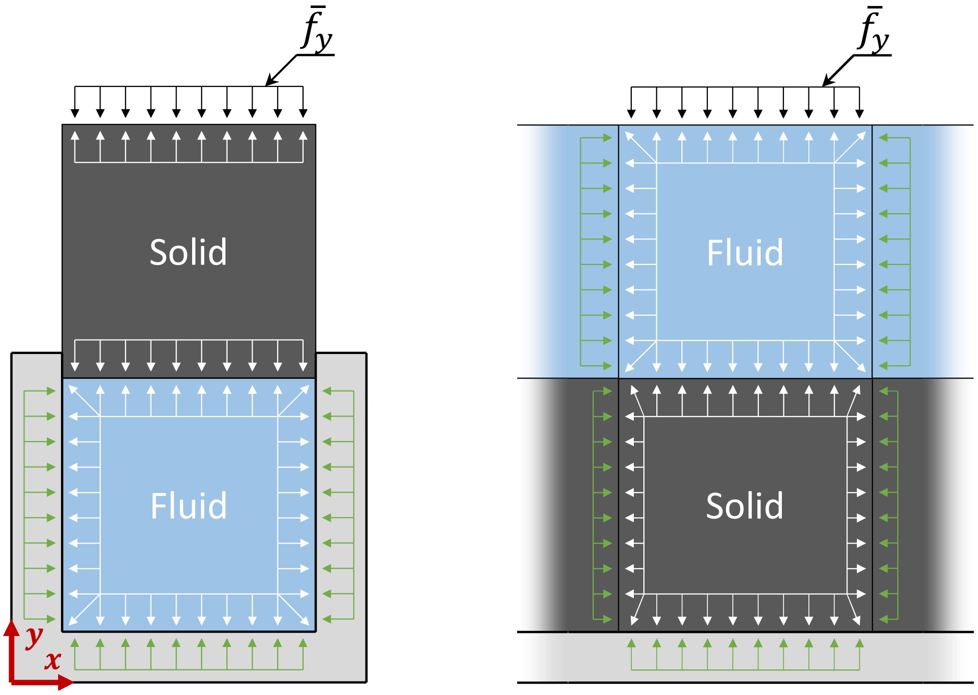
(d) A fluid surrounded by more fluid subjected to a dead load. Virtually the same situation as in (c).

FIGURE 2.9: Comparison of internal forces between fluids and solids for different boundaries and loading situations at equilibrium. Rigid boundaries are light grey, external forces are in black, internal forces in white, and reaction forces in green. Note that gravity is neglected.

because the solid responds with a combination of deviatoric and volumetric stresses. A fluid at rest, as the one in figs. 2.9c and 2.9d, will always exhibit isotropic internal forces because there are only volumetric stresses present. To illustrate further, any material exhibits the following stresses and pressure under the dead load in y-direction, $\bar{f}_y$, while constrained in x and z directions (i.e. as in figs. 2.9b,d:

$$\left.\begin{aligned} \sigma_{yy} &= \bar{f}_y \\ \sigma_{xx} = \sigma_{zz} &= \tfrac{\nu}{1-\nu}\sigma_{yy} \\ p = \tfrac{1}{3}\sigma_{ii} &= \tfrac{1}{3}\left(1 + \tfrac{2\nu}{1-\nu}\right)\bar{f}_y \end{aligned}\right\} \textbf{for fluid and solid} \tag{2.164}$$

where Newtonian fluids always have $\nu = 0.5$, while solids can have $\nu = [-1, 0.5]$. As a result of eq. 2.164, the same loading leads to different pressures in fluids and solids (unless the solid is incompressible $\nu = 0.5$). With this in mind, consider combining a fluid and a solid and loading them, as illustrated in fig. 2.10.



(a) A solid pressing down on a fluid like a piston.

(b) A fluid resting on a solid, like a melt pool on a substrate during welding.

FIGURE 2.10: Fluids and solids loaded while in contact. A difference in internal forces (white arrows) is observed leading to different pressure, despite being subjected to the same loading.

The surface traction normal to the interface must be equal and opposite, as indicated by the rows of opposing white arrows at the interface in fig. 2.10. This also implies that stresses (here: σ_{yy} and σ_{xy}) acting in the normal direction of the interface (here: y-direction) are continuous across the interface. Conversely, a discontinuity of stresses acting in the tangential direction of the interface (here: σ_{xx} and σ_{zz}) and of the pressure may occur, depending on ν in the solid. Note that eq. 2.164 is still valid in the example in fig. 2.10b.

In conclusion, to correctly simulate the fluid and solid behavior at the interface, this discontinuity of the pressure and the continuity of the surface traction across the interface must be respected.

Pressure at the fluid-solid interface

To understand how to correctly manage the pressure at an interface, let us first recall how the pressure appears in the discrete system of equations. The difference between PFEM and FEM with regard to pressure shall highlight the issue.

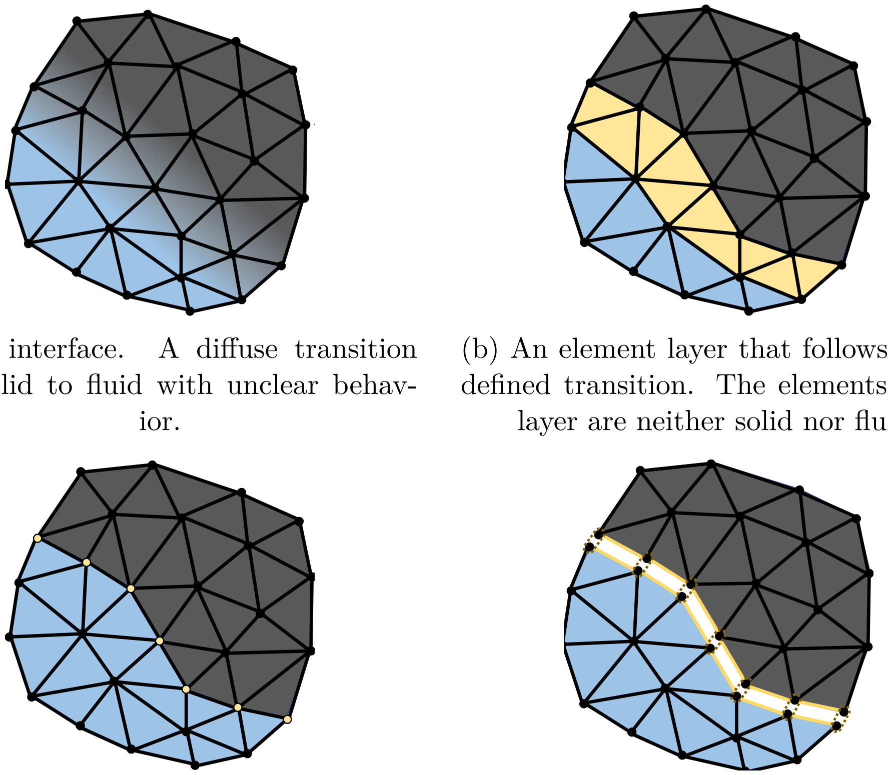
In the classic FEM with a displacement formulation (e.g. *Metafor* [91]), the pressure is not a solution variable and it is not defined at the nodes. The pressure is implicitly taken into account in the momentum conservation equation as part of the stress tensor. Since the stress tensor is evaluated at each Gauss point, the pressure is also known at the Gauss point and is therefore closely related to the element rather than the node.

Using PFEM with a velocity-pressure formulation, the pressure is a nodal solution variable. The nodal pressure (or more precisely, its gradient) is taken into account at the Gauss point through the discrete divergence operator $\mathbf{D}_e^T$ (eq. 2.149), which appears as a separate term in the discrete momentum conservation equation (eq. 2.137 after assembly of the element contributions), next to the term containing the deviatoric stresses.

This fundamental difference of the pressure being defined at the node instead of at the Gauss point leads to a problem with regard to the interface. When a Newtonian fluid and an elastic solid are in contact, they may both experience a different pressure internally due to the explanation given above. Between a fluid element and a solid element, a discontinuity of the pressure must be possible. At the interface, fluid and solid cannot use the same nodal pressure variable to compute the volumetric stress contribution. There are several options to solve this issue:

1. Ignore the need for a discontinuity of the pressure and accept an error of unknown magnitude at the interface (fig. 2.11a).
2. Approximate the sharp interface using a layer of special transition elements (fig. 2.11b).
3. Introduce a second nodal pressure variable for nodes at the interface
   (a) by modifying the interface nodes to accommodate two pressures (fig. 2.11c).
   (b) by doubling the node, but keep them dependent on one another (fig. 2.11d).

Option 1 does not require any further modification of the equations. The entire system of equations is built as described in chapter 3, without regard for the pressure discontinuity. After some preliminary testing it has been found that such an approach can yield results that are sufficiently accurate, except for the area surrounding the interface. There, on the solid side, the pressure underwent some non-physical spatial oscillation. Two or three rows of elements away form the interface, the correct stresses (including the pressure) are recovered. Since it is difficult to judge based on the preliminary testing, whether this option could have its uses, it is not pursued any further in this work and no detailed results are included here. A publication describing a nearly identical unified formulation implemented in a PFEM code by Idelsohn et al. [46] does not elaborate on the interface condition and it is possible that simply no special interface treatment is done. The authors claim to be satisfied with their results (no details were given), which may indicate that this simple approach might be sufficient in some well-behaved cases (e.g. fully submerged solids in 3D or 2D plane stress assumption, where the solid pressure nearly equals the fluid pressure).



(a) No interface. A diffuse transition from solid to fluid with unclear behavior.

(b) An element layer that follows a user defined transition. The elements of the layer are neither solid nor fluid.

(c) A sharp interface that uses special nodes with two pressure degrees of freedom to be used by the adjacent fluid and solid elements respectively.

(d) A sharp interface that uses duplicated (regular) nodes at the interface, but pairs of nodes are interdependent. Note that the gap between the subdomains is only for a clearer visualization. The pairs of nodes and edges are at all times collocated.

FIGURE 2.11: Schematic of the different implementation options for an interface. Solid in dark grey, fluid in blue, interface in yellow.

For Option 2 a layer of elements connects the fluid and solid side, but these elements exhibit neither solid nor fluid behavior. Instead, these elements merely enforce the interface conditions. A similar approach is sometimes used for solid-solid contact problems in PFEM (see e.g. Cerquaglia [39], Oñate [92]), where a layer of elements with special properties, so-called contact elements, is used. In these examples from the literature, a penalty method was implemented for this purpose. Such an approach for the fluid-solid interface has not been tried in this work, but may be more promising than option 1 in the author's opinion, as it can profit from existing experience with such a method.

Option 3a and 3b are very similar, as they both allow for a sharp discontinuity in pressure between the solid and the fluid. The difference lies in the implementation in the code, which will not be discussed here in detail. Option 3a has notably been used by Franci and is described in detail in his doctoral thesis [49]. Such an interface is shown to work as desired through many example problems and validation cases therein.

Finally, option 3b is chosen for this work. An equally good performance is expected as option 3a, but due to the structure of our PFEM code, option 3b appears to be less intrusive and therefore faster to implement. However, it might be the costliest to run and the other options should be explored in the future. The added cost comes from the bloating of the system of equations with additional equations to solve, as will become apparent in the following section. The author is not aware of any examples in the literature that use this approach in a unified formulation. How and when the nodes are duplicated and where the operations related to the interface are executed will be discussed at the beginning of section 2.6 and in section 2.7.5.

Implementation of the interface

The interface is set up in two steps. The first step is the above mentioned splitting of the domain, with the key aspect being the duplication of the nodes that lie on the interface (see fig 2.11d). The two sub-domains are completely separate at this stage. The system of equations of the mechanical problem (eq. 2.131) is built as if the fluid and solid sub-domain are independent and can displace and deform without any interaction with one another. This first step therefore adds three new degrees of freedom per node duplicated: two additional nodal velocities $v_{x,I}, v_{y,I}$ and a nodal pressure p_I. Since only the additional nodal pressure is wanted, the additional velocities must be constrained, specifically, the nodal velocities of the duplicate nodes must be equal to those of its original node. The second step is therefore to impose the same velocities for both original and duplicate node through the Lagrange multiplier technique.

The Lagrange multiplier technique is a well-known technique to enforce constraints of degree of freedom. The constraint is enforced by introducing an additional degree of freedom λ, but also an additional equation.

As a reminder, the original system of equations for the mechanical problem (eq. 2.131) is

$$\mathbf{A}_{mech} \cdot \boldsymbol{q}_{mech} = \boldsymbol{r}_{mech}$$

The Lagrange multiplier technique enlarges the system to become

$$\begin{bmatrix} \mathbf{A}_{mech} & \mathbf{A}_{\lambda}^{T} \\ \mathbf{A}_{\lambda} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{q}_{mech} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_{mech} \\ \mathbf{0} \end{bmatrix} \tag{2.165}$$

where $\mathbf{A}_{\lambda}$ is a sparse matrix of size $2n_{in} \times 3n_n$, n_{in} is the number of nodes on the interface, n_n is the total number of nodes, $\boldsymbol{\lambda}$ is a vector of new unknowns of size $2n_{in}$. $\mathbf{A}_\lambda$ is populated with only zeros, a single 1 and a single -1 in each row, such that for each pair of nodes along the interface the following equations are fulfilled:

$$v_{x,f} - v_{x,s} = 0 \tag{2.166}$$

$$v_{y,f} - v_{y,s} = 0 \tag{2.167}$$

where the subscripts $_s$ and $_f$ refer to the original and duplicate nodes on the solid and fluid side respectively. The above equations obviously enforce that each pair of nodes moves in unison so that the two sub-domains remain in bilateral contact at all times. The vector $\boldsymbol{\lambda}$, which is appended to the solution vector $\boldsymbol{q}_{mech}$, contains the Lagrange multipliers $\lambda_{x,I}$ and $\lambda_{y,I}$ for each pair of nodes I, which represent the x and y-components of a unknown nodal force at the interface. This force must be equal and opposite on each side of the interface, acting on the pair of nodes. Through $\mathbf{A}_\lambda^T \cdot \boldsymbol{\lambda}$, these unknown forces are added to the momentum conservation of the two nodes in each pair I. The original node experiences the additional nodal force components $+\lambda_{x,I}$ and $+\lambda_{y,I}$, while the duplicate node experiences $-\lambda_{x,I}$ and $-\lambda_{y,I}$. This way, equal and opposite nodal forces are imposed at the interface. Note that these forces must be taken into account when calculating the residual of the momentum equation (see section 2.6.1) and are treated as additional external forces.

As mentioned in the previous section, the major drawback of this approach is the increase in size of the system of equations by an extra $2n_{in}$ equations. The advantage is the simple and elegant implementation. Lagrange multipliers furthermore allow to impose a free-slip condition in the future, by only constraining the velocity normal to the interface.

A side note regarding the pressure in the fluid

The mass conservation equation, which governs the pressure, was defined in section 2.2.1, but no boundary condition was given that allows to impose the pressure. This is because the pressure is not normally directly imposed.

However, when solving the Navier-Stokes equations in a purely incompressible fluid domain, a unique field of the pressure *gradient* exists for any given velocity field. Conversely, the actual pressure *level* itself can take any value. In other words, the entire pressure field of a valid solution can be offset by an arbitrary constant pressure difference and still be a valid result that fulfills the Navier-Stokes equations. This is somewhat like a rigid body mode of the pressure field. To fix the pressure field to one pressure level and to have a unique solution, the pressure must be imposed in at least one point. Practically, when there is a free surface, we impose the pressure in the corners, where the free surface meets either a side wall or the solid region (i.e. the interface). If the fluid's free surface does not come into contact with a rigid side wall or a solid, then the pressure is imposed at the node with a number chosen by the user. Because this is an ad-hoc solution, the author decided to not define this as a proper Dirichlet type boundary condition of the mass conservation equation.

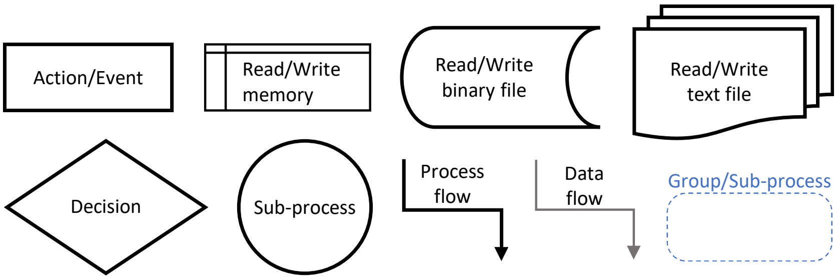


FIGURE 2.12: Flowchart legend.

In a domain that contains a compressible medium, be it fluid or solid, this problem is not encountered. This is due to the coupling between deformation and pressure, which does not exist in the incompressible fluid. Because the compressible material deforms, whenever the pressure changes, it implies that there is a unique deformation for any pressure level too. The domain is initialized with a given pressure distribution and a given shape and that is considered a valid solution. Then, for any pressure change thereafter, a deformation can occur leading to another valid solution with a unique pressure field.

2.6 Solution process

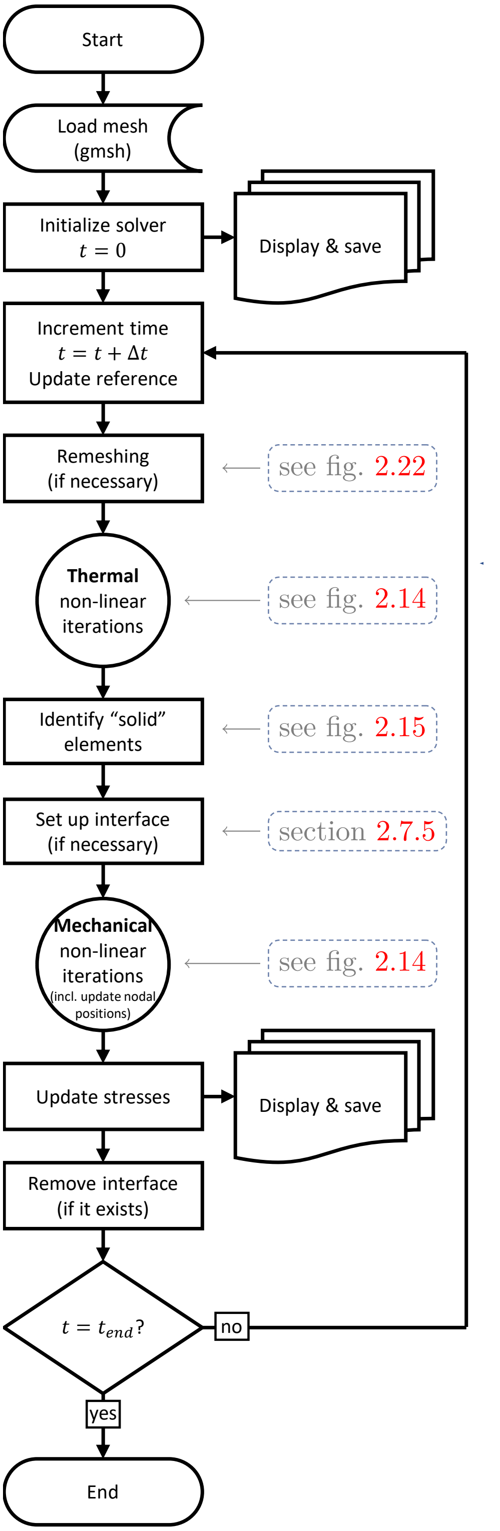
In this section several flow charts are used to illustrate the algorithms. The following legend in fig. 2.12 shall help better identify the nature of the steps. Beginning from the most top-level algorithm, several layers of sub-processes exist and cross-references are used to help guide the reader.

The overall solution process is illustrated in the flow chart in fig. 2.13. The first noteworthy aspect of this algorithm is the **staggered solution scheme**, where, at each time step, the thermal problem is solved first and the mechanical problem is solved consecutively. This staggered approach is chosen because it is easier to implement than a monolithic scheme and a strong coupling is not required. The choice of solving the thermal problem first is due to the applications, welding and additive manufacturing. The phase change and the convective flow in the liquid melt pool are ultimately driven by the heat input and not by external forces. Therefore it seems natural to update the temperature field before the flow field.

The second noteworthy aspect is the optional **fluid-solid interface** that can be established to allow a discontinuity of the pressure, as described in section 2.5.5. The detailed description of the algorithms that are related to the interface are found in section 2.7.5

2.6.1 Non-linear iterative solving

The non-linear solver iteratively solves the non-linear equations at each time step, as illustrated in fig. 2.14. How the system to be solved is assembled depends on the chosen non-linear algorithm (see next sections). In most cases we use the simpler Picard iterations without the line search algorithm (see section


Start
Load mesh (gmsh)
Initialize solver $t = 0$
Display & save
Increment time $t = t + \Delta t$ Update reference
Remeshing (if necessary)
see fig. 2.22
Thermal non-linear iterations
see fig. 2.14
Identify "solid" elements
see fig. 2.15
Set up interface (if necessary)
section 2.7.5
Mechanical non-linear iterations (incl. update nodal positions)
see fig. 2.14
Update stresses
Display & save
Remove interface (if it exists)
$t = t_{end}$?
no
yes
End

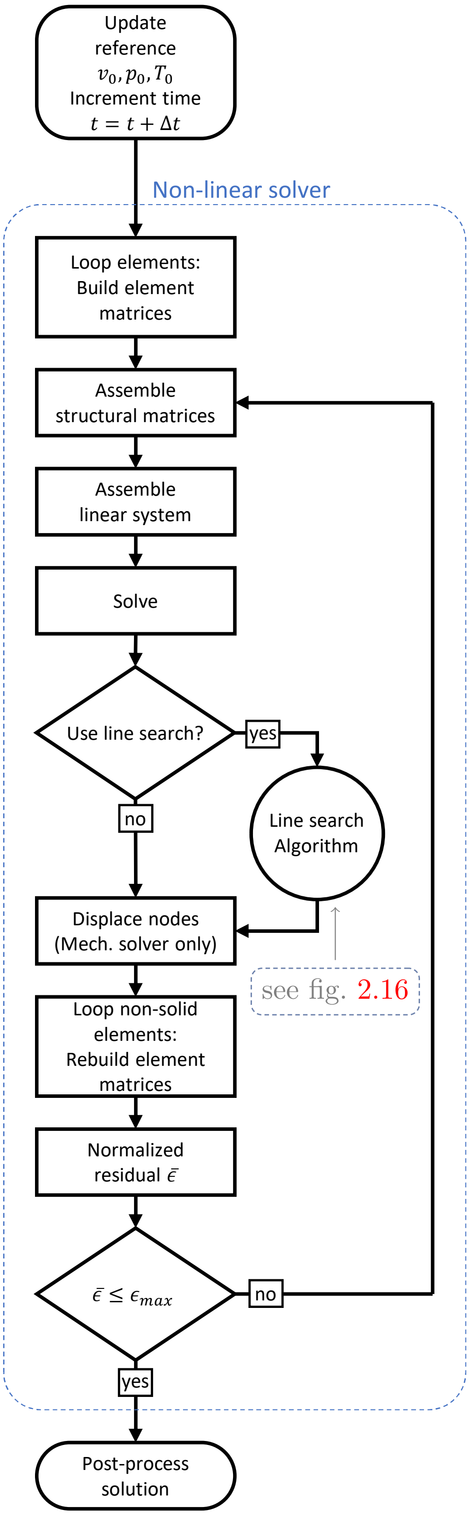

Update reference v_0, p_0, T_0 Increment time $t = t + \Delta t$
Non-linear solver
Loop elements: Build element matrices
Assemble structural matrices
Assemble linear system
Solve
Use line search?
yes
no
Line search Algorithm
see fig. 2.16
Displace nodes (Mech. solver only)
Loop non-solid elements: Rebuild element matrices
Normalized residual $\bar{\epsilon}$
$\bar{\epsilon} \leq \epsilon_{max}$
no
yes
Post-process solution

FIGURE 2.13: Solution process with time marching loop.

FIGURE 2.14: A time step in the non-linear solver

2.6.2) for both the thermal and the mechanical problem, unless the highly non-linear latent heat term is included. In that case the thermal problem is solved using the Newton-Raphson (NR) method with the line search algorithm. The NR method is not used for the mechanical problem because no (cost efficient) analytical computation of the tangent stiffness matrix is currently available due to the complexity of the Navier-Stokes equations with all the different source terms. Currently, a tangent stiffness matrix can be obtained by perturbation, but the drawbacks (costly computation) tends to outweigh the benefits (fewer iterations) and no significant time savings can be achieved.

Identification of "solid" elements

When phase change occurs in a discretized domain, deciding which elements are considered solid is not straight forward. This is because phase change is governed by the liquid fraction f_l (eq. 2.35), which is a continuous function of temperature T. Each triangle has 3 nodes and a given number of Gauss points, each with its own temperature and, by extension, its own liquid fraction. Furthermore, the phase change may not occur at a sharp melting point T_m, but over a melting interval $\Delta T = T_{liq} - T_{sol}$. Either way, some entities associated with a triangle may be liquid, others may be solid and some may be semi-solid. Fig. 2.15 illustrates how a mesh entity arrives at being considered "solid" in this work. In short, if all Gauss points are below the solidus temperature, the element is declared "solid" and with it all its edges and nodes. Partially solid elements, where none or only some Gauss points are below the solidus temperature, are counted towards the non-solid.

Rebuilding of solid element matrices and vectors

As explained in section 2.4.1, solid materials are assumed to undergo only small displacements. This means that the elemental matrices and vectors (eqs. 2.141 - 2.155) do not undergo any changes and do not need to be rebuilt. Elemental matrices and vectors from solid elements remain unchanged from the moment an element is considered solid and these elements are excluded from the matrix rebuilding step. Since these matrices and vectors remain unchanged, they effectively refer back to the configuration at which they were last built. This is referred to as the reference configuration[5]. In other words, the reference configuration is the configuration at which an element first became solid. In Fluid-Structure interaction (FSI) applications, the solid region is solid from the beginning and the reference configuration is the initial configuration. For phase change applications on the other hand, an element may exists as a non-solid element at first and then turn into a solid at any point throughout the simulation, which is where the reference configuration for that element is set.

[5] The reference configuration here is not to be confused with the one typically mentioned in the context of the Total Lagrangian approach. For non-solids, the Updated Lagrangian approach is always used in this work and for solids, there is a reference configuration, but it would be misleading to refer to this as a Total Lagrangian approach.

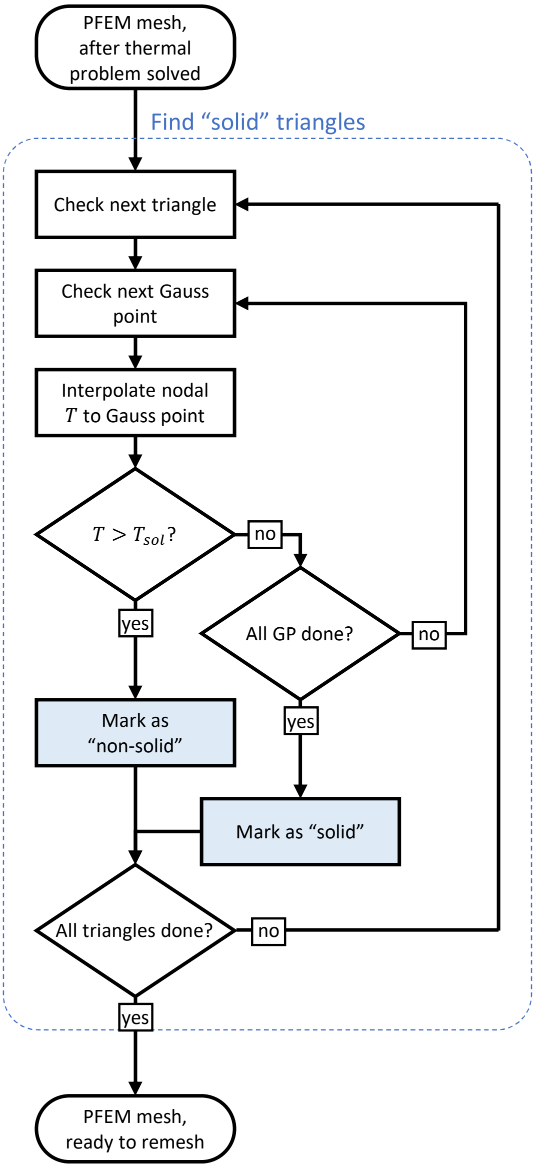


FIGURE 2.15: Determining "solid" triangles.

If for any reason (e.g. changing mesh connectivity) a solid element matrix still needs to be rebuilt, it is rebuilt in that reference configuration (see section 2.5.3).

Note that for consistency, an external loading at a solid boundary must also be integrated in the reference configuration to remain consistent with the stresses in the element that are effectively calculated in the reference configuration.

Residual

The residual is a measure of how well the (non-linear) equations are satisfied. It is calculated by injecting the solution vector $\boldsymbol{q}_{mech}$ or $\boldsymbol{q}_{th}$ back into the system of equations eq. 2.131 or 2.132, rearranging and introducing the vector of nodal residuals $\boldsymbol{\epsilon}_{mech}$ or $\boldsymbol{\epsilon}_{th}$, such that

$$\boldsymbol{\epsilon}_{mech} = \mathbf{A}_{mech} \cdot \boldsymbol{q}_{mech} - \boldsymbol{r}_{mech} \tag{2.168}$$
$$\boldsymbol{\epsilon}_{th} = \mathbf{A}_{th} \quad \cdot \boldsymbol{q}_{th} \quad - \boldsymbol{r}_{th} \tag{2.169}$$

where the individual components of $\boldsymbol{\epsilon}_{mech}$ and $\boldsymbol{\epsilon}_{th}$ are arranged as follows:

$$\boldsymbol{\epsilon}_{mech} = \left[\epsilon_{x,1} \cdots \epsilon_{x,I} \cdots \epsilon_{x,n_n} \qquad \epsilon_{y,1} \cdots \epsilon_{y,I} \cdots \epsilon_{y,n_n} \qquad \epsilon_{p,1} \cdots \epsilon_{p,I} \cdots \epsilon_{p,n_n}\right]^T \tag{2.170}$$
$$\boldsymbol{\epsilon}_{th} = \left[\epsilon_{q,1} \cdots \epsilon_{q,I} \cdots \epsilon_{q,n_n}\right]^T \tag{2.171}$$

where $\epsilon_{x,I}$ and $\epsilon_{y,I}$ are the residuals of the x- and y-momentum conservation equation contributions, $\epsilon_{p,I}$ is the mass conservation equation contribution and $\epsilon_{q,I}$ is the energy conservation equation contribution to node I. Each nodal residual ideally becomes zero and has the units of the respective conservation equation.

All three conservation equations can be split up into their individual contributions and then grouped into internal, external and inertial terms.

$$\boldsymbol{F}_{int} = \mathbf{K}_{dev} \cdot \boldsymbol{v}^n + \mathbf{D} \cdot \boldsymbol{p}^n - \boldsymbol{k}_{dev} \tag{2.172}$$
$$\boldsymbol{F}_{ext} = \boldsymbol{b} + \boldsymbol{t} + \boldsymbol{\lambda} \tag{2.173}$$
$$\boldsymbol{F}_{ine} = \frac{1}{\Delta t}\mathbf{M} \cdot \left(\boldsymbol{v}^n - \boldsymbol{v}^{n-1}\right) \tag{2.174}$$

$$\dot{\boldsymbol{V}}_{int} = \mathbf{K}_{vol} \cdot \left(\boldsymbol{p}^n - \boldsymbol{p}^{n-1}\right) + \mathbf{D} \cdot \boldsymbol{v}^n \tag{2.175}$$
$$\dot{\boldsymbol{V}}_{ext} = \boldsymbol{a} \tag{2.176}$$
$$\dot{\boldsymbol{V}}_{ine} = 0 \tag{2.177}$$

$$\boldsymbol{P}_{int} = \mathbf{K}_{cond} \cdot \boldsymbol{T}^n \tag{2.178}$$
$$\boldsymbol{P}_{ext} = \boldsymbol{q}^n + \boldsymbol{s}_L^n + \boldsymbol{s}_R^n \tag{2.179}$$
$$\boldsymbol{P}_{ine} = \frac{1}{\Delta t}\mathbf{C} \cdot \left(\boldsymbol{T}^n - \boldsymbol{T}^{n-1}\right) \tag{2.180}$$

Note that eqs. 2.175 - 2.177 may also contain stabilization terms that are omitted here for clarity. For easier interpretation, the residuals are normalized with the help of the terms in eqs. 2.172 - 2.180. In this work, we chose the following normalization to obtain the normalized residuals $\bar{\epsilon}_\phi$

$$\bar{\epsilon}_\phi = \frac{\|\epsilon_\phi\|}{\nu_\phi} \tag{2.181}$$

where ν_ϕ is the normalization factor for the equation specified by the subscript $_\phi$ (either $_{mech}$ or $_{th}$). Unless otherwise stated, the following definition of ν_ϕ is used for each conservation equation:

$$\nu_{mom} = \|\boldsymbol{F}_{int}\| + \|\boldsymbol{F}_{ext}\| + \|\boldsymbol{F}_{ine}\| \tag{2.182}$$

$$\nu_{mass} = \|\dot{\boldsymbol{V}}_{int}\| + \|\dot{\boldsymbol{V}}_{ext}\| + \|\dot{\boldsymbol{V}}_{ine}\| \tag{2.183}$$

$$\nu_{heat} = \|\boldsymbol{P}_{int}\| + \|\boldsymbol{P}_{ext}\| + \|\boldsymbol{P}_{ine}\| \tag{2.184}$$

where ν_{mom}, ν_{mass} and ν_{heat} are the normalization factors for the residual of the momentum, mass and energy conservation equations, respectively. Unless otherwise stated, the residual target ϵ_{max} is set to

$$\bar{\epsilon}_{\phi} < \epsilon_{max} = 10^{-6} \tag{2.185}$$

where for the mechanical problem, only $\bar{\epsilon}_{mom} < \epsilon_{max}$ is tested, if not stated otherwise.

Picard iterations

Picard iterations are also referred to as fixed-point iterations and are a simple iterative solution scheme for non-linear partial differential equations (PDE), such as the ones to be solved here. In the literature there are few examples of the Picard method being used in PFEM. Aubry et al. [93] state that it is the most memory-efficient choice for their specific application. The complete algorithm based on the Picard method used in this work can be summarized as follows.

1. (Re)build element matrices and vectors
2. Assemble left-hand-side matrix and right-hand-side vector as in eq. 2.131 or 2.132
3. Invert left-hand-side matrix and pre-multiply both sides with inverse
4. Solution vector $\boldsymbol{q}_{mech}$ or $\boldsymbol{q}_{th}$ is obtained
5. Update nodal unknowns (velocities $\boldsymbol{v}_I$ and pressure p_I or temperature T_I)
6. Update nodal position using eq. 2.116 (mechanical problem only)
7. Compute and check residual (eq. 2.185)
8. Repeat, if necessary

The Picard iterations are characterized by steps 3 and 4, as opposed to a more complicated algorithm in the following section. Whenever the physics are actually linear in nature (e.g. no mesh deformation in the mechanical problem and constant source terms), the correct solution is reached in one iteration. Otherwise the time step is chosen such that two or three iterations are needed.

We use the *Gmm++* library (`www.getfem.org/gmm/`) to perform all linear algebra operations, including the inversion of the left-hand-side matrix. The inversion is internally performed by a sparse matrix LU factorization and this step typically accumulates to 30% to 70% of the total CPU time of a simulation run, with no parallelization used for any part of the algorithm.

Newton-Raphson method

The Newton-Raphson (NR) method allows to reach the solution with fewer iterations, by approaching the correct solution more systematically. Instead of solving directly for the nodal solution variables in the solution vectors $\boldsymbol{q}_{mech}$ or $\boldsymbol{q}_{th}$, NR solves for a change in the solution variables $\Delta\boldsymbol{q}_{mech}$ or $\Delta\boldsymbol{q}_{th}$ that is to be added to the previous iteration's solution.

$$\boldsymbol{q}_{\phi}^{k} = \boldsymbol{q}_{\phi}^{k-1} + \Delta\boldsymbol{q}_{\phi} \tag{2.186}$$

where k refers to the current iteration and the subscript $_{\phi}$ is a placeholder for either system of equations ($_{mech}$ or $_{th}$). Note that at the beginning of a time step n, when $k = 1$, we set $\boldsymbol{q}_{\phi}^{k-1}$ in the above equation to be

$$\boldsymbol{q}_{\phi}^{k-1} = \boldsymbol{q}_{\phi}^{n-1} \tag{2.187}$$

Now, $\Delta\boldsymbol{q}_{\phi}$ in eq. 2.186 must be determined such that it eliminates the residuals of the current iteration $\boldsymbol{\epsilon}_{\phi}^{k}$.

$$\boldsymbol{\epsilon}_{\phi}^{k} = \boldsymbol{\epsilon}_{\phi}^{k-1} + \Delta\boldsymbol{\epsilon}_{\phi} = 0 \tag{2.188}$$

where $\Delta\boldsymbol{\epsilon}_{\phi}$ is the change in the residual between the current and the previous iteration, with subscript $_{\phi}$ being the placeholder referencing the residual associated with the respective system of equations ($_{mech}$ or $_{th}$). To solve eq. 2.188, the slope of the residual with respect to the solution vector $\partial\boldsymbol{\epsilon}_{\phi}/\partial\boldsymbol{q}_{\phi}$ is needed.

$$\Delta\boldsymbol{\epsilon}_{\phi} = \frac{\partial\boldsymbol{\epsilon}_{\phi}}{\partial\boldsymbol{q}_{\phi}} \cdot \Delta\boldsymbol{q}_{\phi} = \mathbf{T} \cdot \Delta\boldsymbol{q}_{\phi} \tag{2.189}$$

where $\mathbf{T}$ is commonly referred to as tangent stiffness matrix. There are two common methods to obtain $\mathbf{T}$:

1. by perturbation
2. analytically

The first option is very simple to implement, no matter how complex the system of equations actually is, but it can be very costly to obtain at runtime. More details are explained in Cerquaglia's thesis [39]. The second option requires determining the partial derivatives of each term in the equation with respect to each solution variable. This may require manual upfront work and a more intricate implementation, but the algorithm will provide $\mathbf{T}$ much more quickly. For the mechanical problem, we have implemented the derivation of $\mathbf{T}$ by perturbation, but it is rarely used (usually Picard instead). For the thermal problem we have implemented the analytical derivation of $\mathbf{T}$ and it is used by default.

Once $\mathbf{T}$ is obtained, it is inverted and both sides of the equation are premultiplied with it to obtain

$$\Delta \boldsymbol{q}_\phi = \mathbf{T}_\phi^{-1} \cdot \Delta \boldsymbol{\epsilon}_\phi \tag{2.190}$$

This is injected back into eq. 2.186, which now delivers the updated solution vector of the current iteration $\boldsymbol{q}_\phi^k$.

$$\boldsymbol{q}_\phi^k = \boldsymbol{q}_\phi^{k-1} - \mathbf{T}_\phi^{-1} \cdot \boldsymbol{\epsilon}_\phi^{k-1} \tag{2.191}$$

For comparison with the simpler Picard algorithm, the implemented NR algorithm follows the following steps:

1. (Re)build
   (a) element matrices and vectors and
   (b) derivatives of element matrices and vectors
2. Assemble tangent stiffness matrix and residual vector
3. Invert and pre-multiply tangent stiffness matrix
4. Solution increment vector $\Delta \boldsymbol{q}_{mech}$ or $\Delta \boldsymbol{q}_{th}$ is obtained
5. Update solution vector $\boldsymbol{q}_{mech}$ or $\boldsymbol{q}_{th}$ by adding increment vector
6. Update nodal unknowns (velocities v_I and pressure p_I or temperature T_I)
7. Update nodal position using eq. 2.116 (mechanical problem only)
8. Compute and check residual (eq. 2.185)
9. Repeat, if necessary

Compared with the Picard algorithm, the NR algorithm is characterized by the additional step 1 (b) and the steps 3, 4 and 5. Since $\mathbf{A}_\phi$ is of the same size as $\mathbf{T}_\phi$, the inversion and pre-multiplication take about the same time. Step 1 is very simple to parallelize[6] and the added CPU time is insignificant. In case of an analytically derived $\mathbf{T}$, no additional CPU is required during step 2. In case of a $\mathbf{T}$ derived by perturbation, the steps 1 and 2 need to be repeated 2 times for each independent variable (e.g. 6 repetitions for the mechanical problem with 3 independent variables v_x, v_y and p), which can quickly add up. The additional step 5 barely adds any overhead. From our experience, the use of NR with the analytically derived $\mathbf{T}$ is nearly as fast per iteration as the simpler Picard algorithm, but requires fewer iterations when non-linearities are present. The NR algorithm with a $\mathbf{T}$ derived by perturbation is clearly slower per iteration and can rarely make up for the added cost by requiring sufficiently fewer iterations. A discussion of CPU time is omitted, since it is not the scope of this work.

[6] Parallelization of the classic PFEM is not trivial [38][94] and in our code only this step 1 in both Picard and NR algorithms are well-parallelized using the *oneTBB* library (`www.github.com/oneapi-src/oneTBB`). This means that individual element matrices and vectors are built in parallel, before being assembled into their structural counterparts

2.6.2 Line search algorithm

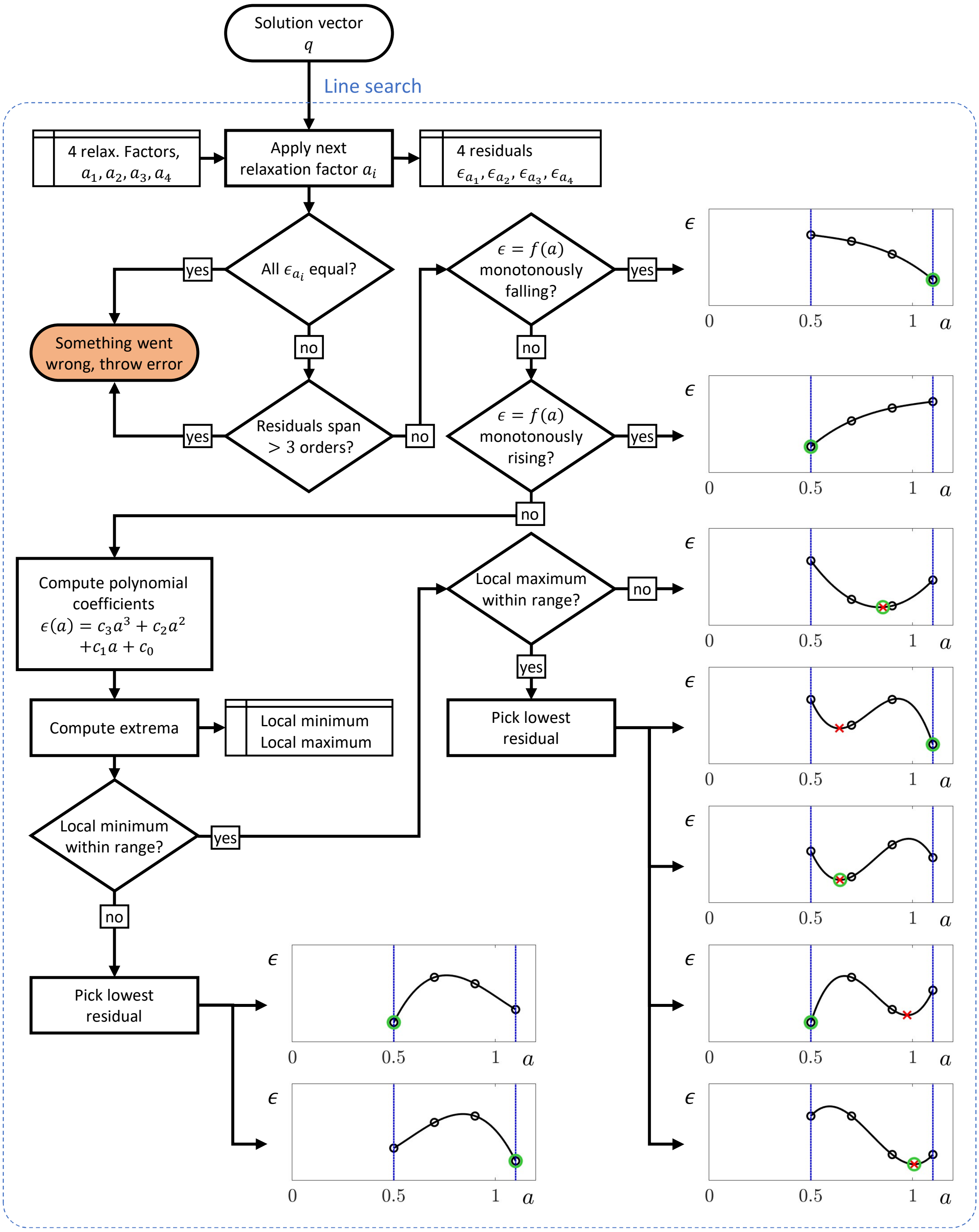

Solution vector
q
Line search
4 relax. Factors, a_1, a_2, a_3, a_4
Apply next relaxation factor a_i
4 residuals $\epsilon_{a_1}, \epsilon_{a_2}, \epsilon_{a_3}, \epsilon_{a_4}$
All ϵ_{a_i} equal?
yes
no
Something went wrong, throw error
Residuals span > 3 orders?
yes
no
$\epsilon = f(a)$ monotonously falling?
yes
no
$\epsilon = f(a)$ monotonously rising?
yes
no
Compute polynomial coefficients $\epsilon(a) = c_3a^3 + c_2a^2 + c_1a + c_0$
Compute extrema
Local minimum
Local maximum
Local minimum within range?
yes
no
Pick lowest residual
Local maximum within range?
no
yes
Pick lowest residual
ϵ
0
0.5
1
a

FIGURE 2.16: Schematic of the line search algorithm. This example uses as relaxation factors $a = \{0.5, 0.7, 0.9, 1.1\}$ resulting in some constellation of 4 values of the residual (black circles). From the 4 value pairs, a cubic function can be constructed. A cubic function always has a global maximum; either at $+\infty$ or at $-\infty$. If a local minimum is within the range of relaxation factors (delimited by blue lines), it is marked with a red cross. Following the flow chart, the optimal value for a is selected (green circle).

Relaxation is the process of scaling the change of the solution vector $\Delta\boldsymbol{q}_{mech}$ or $\Delta\boldsymbol{q}_{th}$ by a given factor, the relaxation factor a_r. Typically, $a_r < 1$, which is referred to as under-relaxation. The aim is to suppress overshooting the target, when iteratively approaching the balanced solution. The trade-off for a more stable convergence is the need for more iterations, as the algorithm approaches the final solution more slowly.

The line search algorithm is used to optimize convergence of the residual during the non-linear iterations. It essentially finds the optimal relaxation factor that is applied to the raw solution after solving. There exist many approaches to find such an optimal relaxation factor. In this work, we consecutively apply four different relaxation factors $a_1 \cdots a_4$ that the user defines[7] and four respective values for the residual $\epsilon_{a_1} \cdots \epsilon_{a_4}$ are obtained. Using the four value pairs, a cubic polynomial $\epsilon = f(a)$ is constructed. From the polynomial, the optimal relaxation factor can be obtained, which results in the lowest residual. The algorithm used in this work is illustrated in fig. 2.16

2.6.3 Higher-order spatial integration

Spatial integration is accomplished with the Gauss-Legendre method. This method is exact for any integrand that is a polynomial of order up to $2n_{GP} - 1$ in space, n_{GP} being the number of Gauss points. For any other integrand, an error is introduced and more Gauss points result in a more accurate integration. This is particularly interesting for the latent heat term, eq. 2.36. In the extreme case of isothermal phase change without a regularization, this term tends to infinity as the time step size approaches 0. Even with a regularization and a finite time step size, this term is still very large and highly non-linear. More Gauss points can better capture the non-linearity, but using more Gauss points leads to higher CPU cost. A good compromise is to use extra Gauss points only for elements that are undergoing phase transition in the current time step and only use the extra Gauss points for the latent heat term. Our implementation is capable of detecting elements undergoing phase transition and subsequently increasing the number of Gauss points locally in these elements and only for the integration of the latent heat term.

2.7 Mesh management

This simulation method is based on the classic particle finite element method (PFEM). While PFEM has particle character, a mesh is used as in the classic finite element method FEM. The mesh is the geometric representation of the discretization of a complexly-shaped domain divided into many small elements of simple shape. As with all methods that use a mesh, the mesh quality is of great importance for the stability and accuracy of the method. The information

[7] From our experience, the range of the 4 factors should span from well below unity to just above unity. This means that when the system is unstable or very non-linear, the line search algorithm can select a strong under-relaxation and when it is stable and linear, it can select $a = 1$ (no relaxation) or even a slight over-relaxation ($a > 1$).

that a mesh contains is the list of mesh entities and their interconnection. Mesh entities are

- volume elements (i.e. Polyhedra, usually hexahedra or tetrahedra)
- surface elements (i.e. Polygons, usually quads or triangles)
- curve elements (usually polynomial-like lines or parabolas)
- edges
- point entities (i.e. nodes)

where edges are purely geometric in contrast to the line element, which performs mathematical operation such as integration. Every line has a coinciding edge, but not every edge has a line (e.g. internal edges do not need to perform integrations). The nodes play a special role because they are defined by their coordinates, while the other entities are defined by their nodes. For example, an edge or line is defined by two nodes, a triangle is defined by 3 nodes and 3 edges and so on.

A key characteristic of PFEM is the frequent remeshing. Based on the initially provided mesh, the nodes are determined. Being a Lagrangian method, as the simulation progresses and marches through time, a gradual deformation of the domain can occur. This means that the user-provided initial mesh deforms along with its elements. This may reach a point, where elements that were of a good shape become highly distorted. A bad shape can cause inaccuracies that can even destabilize the simulation. Since the classic PFEM is supposed to model large deformations of the domain (for example fluid flow with waves breaking), such excessive mesh deformations, so-called mesh distortions, occur frequently. To avoid these problems, a remeshing can restore a good quality mesh. For this, only the nodes are kept, while all other mesh entities are deleted. An automatic remeshing creates a new set of these mesh entities, but they may have a different connectivity as before the remeshing. This is why nodes play a special role in the PFEM: they are the only mesh entities that survive a remeshing, so it is wise to store the solution variables at the nodes. The other mesh entities merely become a temporary tool that allows obtaining the solution for the current time step. Note that this is different for a solid, where a deformation history is associated with the (deformed) element instead of with the nodes. This problem that goes beyond the classic PFEM is discussed further down in section 2.7.3.

Since the mesh and the remeshing procedure play such an important role in the PFEM, this section explains in detail how the algorithm works. In this work, there is only a 2D representation of the domain (see explanation in section 2.5.3) and the elements are linear, meaning that only lines and triangles are used. Triangles are used instead of the often preferred quads because a well-established, fast and reliable automatic remeshing algorithm is needed. The most suitable one provides only triangles and shall be introduced in the following section.

2.7.1 Delaunay triangulation

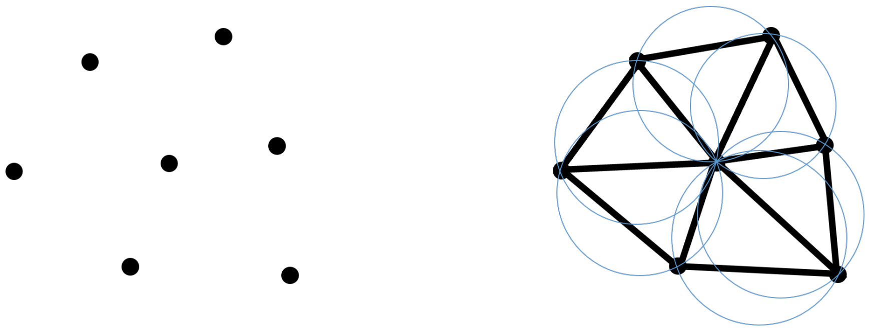
Given a set of nodes, the Delaunay triangulation in 2D produces a mesh of triangles that maximizes the minimum angle of each triangle (see fig. 2.17). From a FEM perspective, a good quality triangle is ideally equilateral, which means that a Delaunay triangulation will produce triangles that are as ideal as possible, based on the set of nodes provided.

This is achieved by enforcing the condition that each triangle's circumcircle is guaranteed to contain no other node (see fig. 2.17b). This property has the remarkable effect that a Delaunay triangulation is unique, which means every remeshing of the same set of nodes produces the same outcome (with only very few exceptions to this rule). This is very useful in the context of PFEM because frequent changing of the connectivity would introduce more diffusion. Another important effect of this property is that if, in a given mesh, a node is moved such that it enters another triangle's circumcircle, the mesh is not a true Delaunay triangulation anymore. A very simple operation that only involves the 3 nodes on the original circumcircle and the "intruding" node can immediately restore a correct Delaunay triangulation: the flip. Flipping will become important further down, so fig. 2.18 shall demonstrate it.

In this work, the Delaunay triangulation is performed by *Triangle* (`https://www.cs.cmu.edu/~quake/triangle.html`), which is limited to 2D, but fast and reliable. The triangulation is performed by writing the list of nodes and their position to a file and giving it as input to *Triangle*. *Triangle* then returns files that contain lists of lines and triangles with their respective node numbers. This information is sufficient for the algorithm to rebuild the complete mesh. The details of *Triangle* are described in Shewchuk's publication [95].

2.7.2 Classic α-shape technique

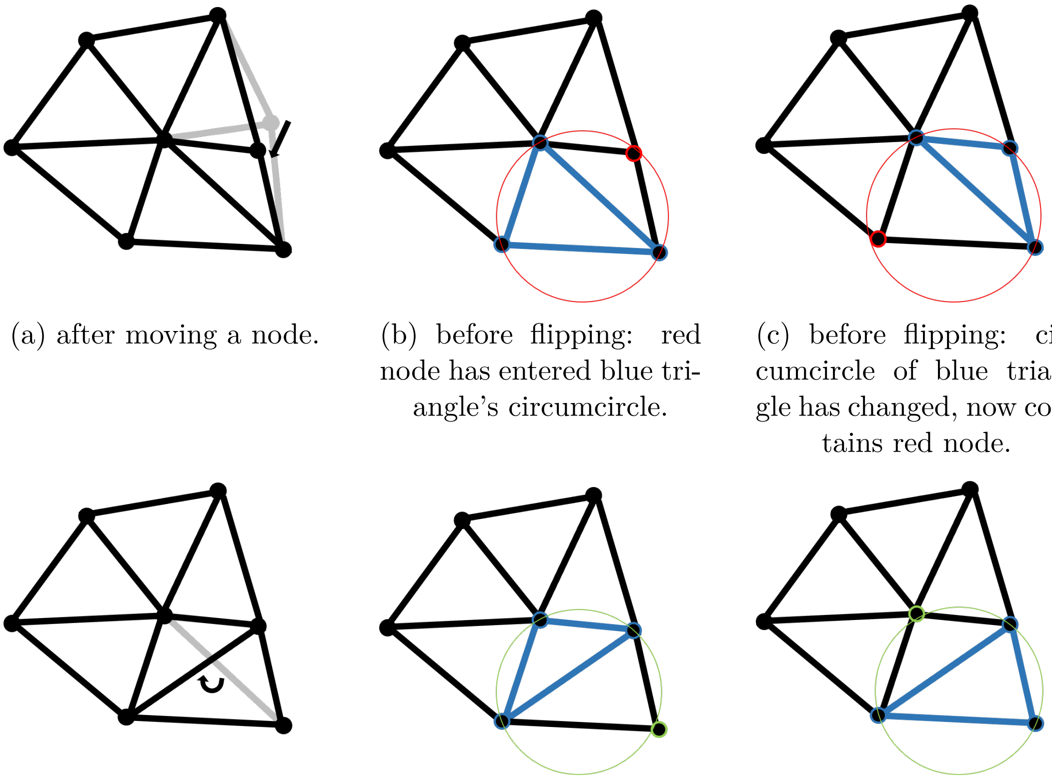
In the early publications of PFEM (e.g. [37, 93]), the use of the α-shape (or Alpha shape) technique proposed by Edelsbrunner und Mücke [36] is another key component of the PFEM. It allows to find the contour of a triangulated



(a) A point cloud of nodes.

(b) Delaunay triangulation: Circumcircles each contain no other node.

FIGURE 2.17: A Delaunay triangulation on a small patch.



(a) after moving a node.

(b) before flipping: red node has entered blue triangle's circumcircle.

(c) before flipping: circumcircle of blue triangle has changed, now contains red node.

(d) flip.

(e) after flipping: flip has restored empty circumcircle of upper triangle.

(f) after flipping: flip has restored empty circumcircle of bottom triangle.

FIGURE 2.18: Flipping in a Delaunay triangulation.

point cloud (fig. 2.19a) that is more detailed than that of the convex hull from the Delaunay triangulation (fig. 2.19b). The α-shape technique allows to remove the unwanted triangles using a simple geometric criterion, which is

$$\alpha_e = \frac{r_c}{\bar{l}_e} < \alpha_{crit} \ \forall \text{ valid triangles} \tag{2.192}$$

where α_e is the triangles's alpha value, r_c is the current radius of the triangle's circumcircle, $\bar{l}_e$ is a characteristic edge length of a triangle. Here we assume a uniform mesh at first (non-uniform see further down), so that the mean edge length of the initial mesh $\bar{l}_{e0}$ can be used:

$$\bar{l}_e = \bar{l}_{e0} \tag{2.193}$$

Lastly, α_{crit} is a global parameter chosen by the user. According to Falla et al. [96] $\alpha_{crit} = 1.2$ is a good choice, which is used in this work unless otherwise stated. After removing all triangles that violate this criterion, a certain shape is obtained. The shape that emerges depends on the choice of α_{crit}.

The larger α_{crit} is chosen, the larger a triangle's circumcircle can be before the triangle is deleted. With a larger allowed circumcircle, flatter or larger elements survive and finer features of the contour are not captured anymore (figs. 2.19c,d). On the other hand, when decreasing α_{crit} too much, the free surface deteriorates and the body may be fragmented (figs. 2.19g,h).

This technique together with the Lagrangian nature of the PFEM, allows to capture the free surface and interface deformation without requiring an interface tracking algorithm, which may introduce new challenges. However, as PFEM is more and more developed, later publications (e.g. [49]) do not rely on the α-shape technique alone anymore. It is too simple to give the desired result in a broad range of situations. For example, the α-shape technique will detect and eliminate one type of sliver elements in 2D, where there is one long edge and the third node being close to it (a flat shape), because this constellation results in a large r_c (fig. 2.20d). The α-shape technique will not reliably detect another type of sliver, where there are two long edges and one extremely short one (a needle shape) because it may not result in a large value for α_e (fig. 2.20c). The common approach in the PFEM literature is to add further mesh manipulation techniques, but it is rarely described in detail. The additional mesh manipulations are summarized in the next sections.

While the classic α-shape technique is very useful, providing a simple solution to a complex problem, it also suffers from some drawbacks:

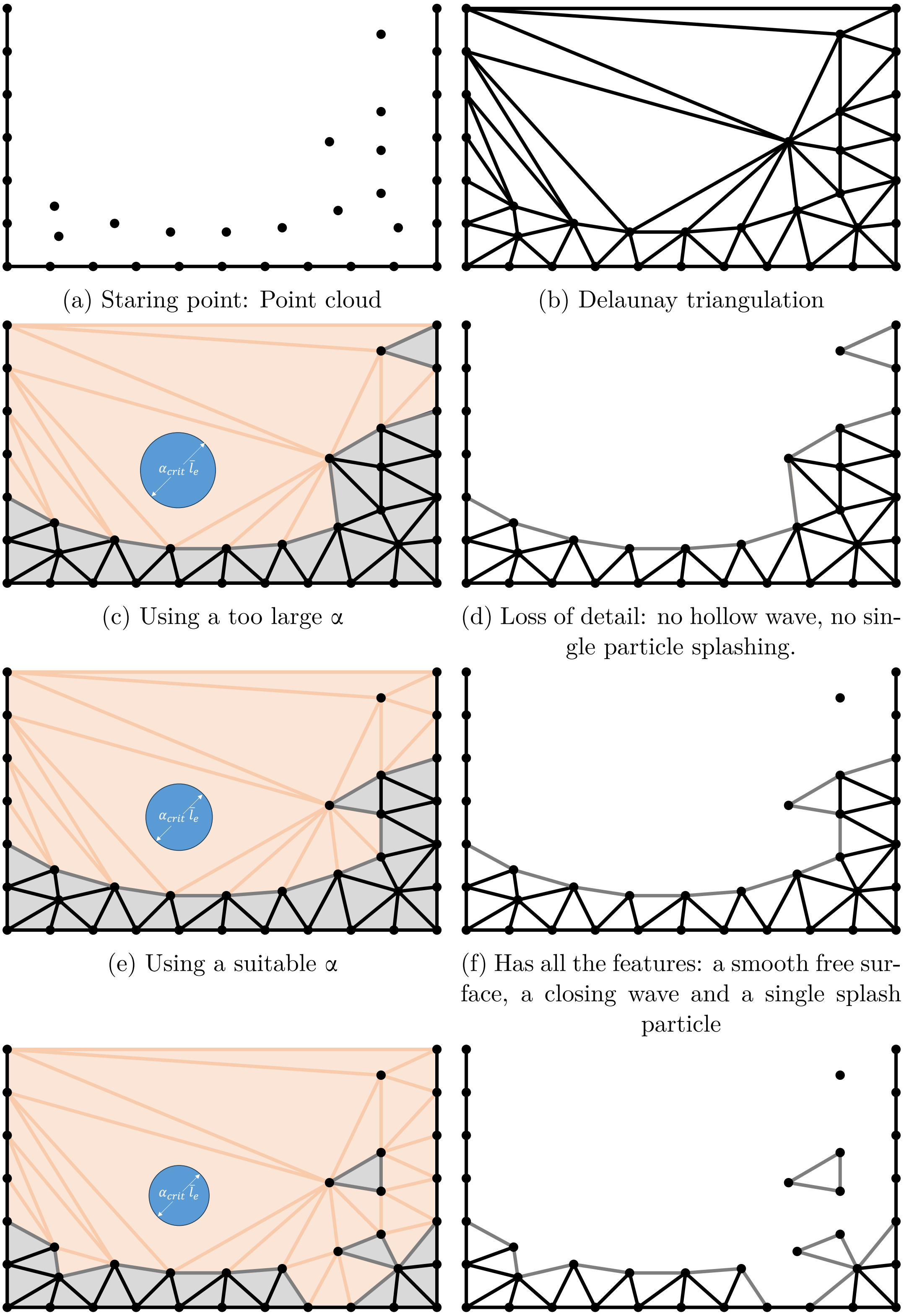
- The solution may depend somewhat on the choice of α_{crit}
- The classic α-shape technique is not suitable for non-uniform meshes
- The α-shape technique is known to cause mass conservation issues at the free surface, at interfaces and at walls

For **non-uniform meshes**, the global parameter $\bar{l}_e$ in eq. 2.192 can simply be replaced by a local parameter that we call a characteristic element length l_e

$$\bar{l}_e = l_e \tag{2.194}$$

where l_e is simply the shortest edge of triangle element e. This criterion now does not delete large elements anymore, as was the case before. This is because l_e and r_c scale linearly as an elements changes in size, provided it keeps the same shape (fig. 2.21b). This criterion therefore can only detect elements of bad shape (i.e. needle-shaped type fig. 2.21c and flat type fig. d) in non-uniform meshes. This shows that the α-shape criterion alone cannot reliably preserve a good quality mesh.

Regarding the **mass conservation issue**, the free surface recognition is indirectly controlled by the α-shape technique and depending on the settings, there can be regions, where elements are systematically falsely deleted and other regions where elements are systematically falsely added. This is a removal or addition of mass that is not physical, but purely an artifact of the method. To the best of the authors knowledge, these error mechanisms cannot be entirely avoided, but only minimized. The interested reader can find a short summary of how the α-shape technique contributes to mass conservation issues in the PFEM review article by Cremonesi et al. [38]. A more detailed explanation


$\alpha_{crit}\,\bar{l}_e$
$\alpha_{crit}\,\bar{l}_e$
$\alpha_{crit}\,\bar{l}_e$

(a) Staring point: Point cloud

(b) Delaunay triangulation

(c) Using a too large α

(d) Loss of detail: no hollow wave, no single particle splashing.

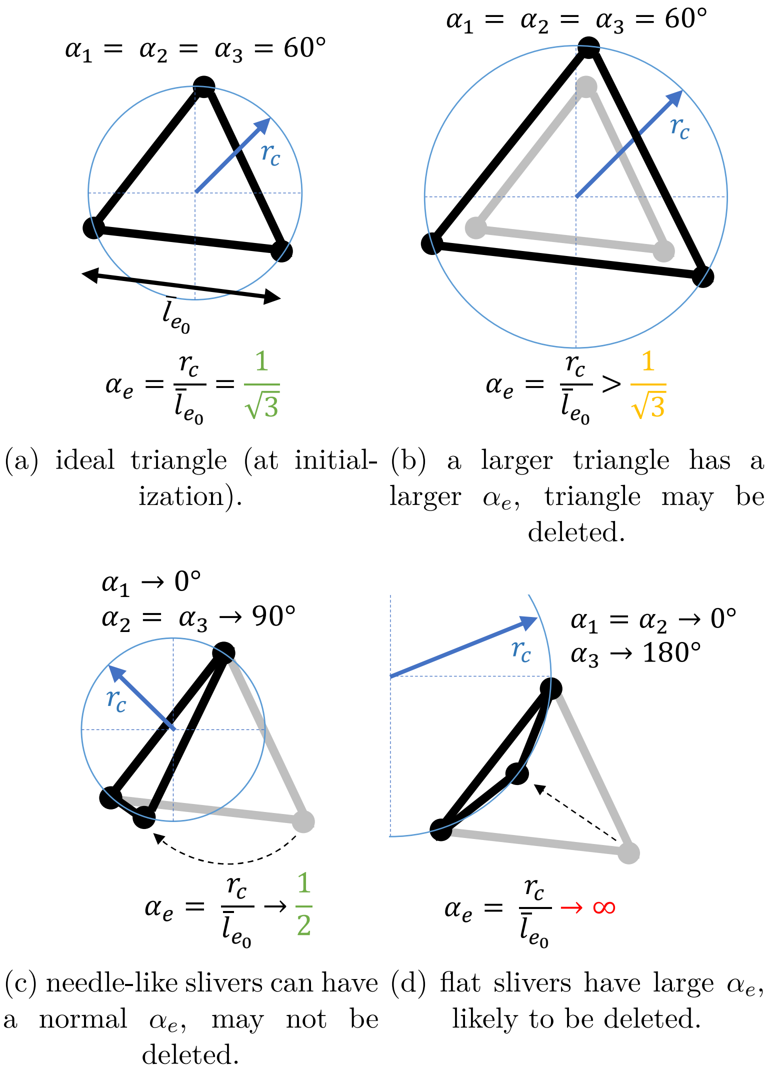
(e) Using a suitable α

(f) Has all the features: a smooth free surface, a closing wave and a single splash particle

(g) Using a too small α

(h) Free surface is degraded and unwanted fragmentation occurs

Figure 2.19: Example of a Delaunay triangulation and subsequent α-shape, illustrating the effect of the choice of the parameter α_{crit}. Red triangles and lines are deleted, light grey triangles are kept. Light grey lines are identified as free surface. The blue circle shows the maximum allowed triangle circumcircle.


$\alpha_1 = \alpha_2 = \alpha_3 = 60°$
r_c
l_{e_0}
$\alpha_e = \frac{r_c}{\bar{l}_{e_0}} = \frac{1}{\sqrt{3}}$
$\alpha_1 = \alpha_2 = \alpha_3 = 60°$
r_c
$\alpha_e = \frac{r_c}{\bar{l}_{e_0}} > \frac{1}{\sqrt{3}}$
$\alpha_1 \rightarrow 0°$
$\alpha_2 = \alpha_3 \rightarrow 90°$
r_c
$\alpha_e = \frac{r_c}{\bar{l}_{e_0}} \rightarrow \frac{1}{2}$
r_c
$\alpha_1 = \alpha_2 \rightarrow 0°$
$\alpha_3 \rightarrow 180°$
$\alpha_e = \frac{r_c}{\bar{l}_{e_0}} \rightarrow \infty$

(a) ideal triangle (at initialization).

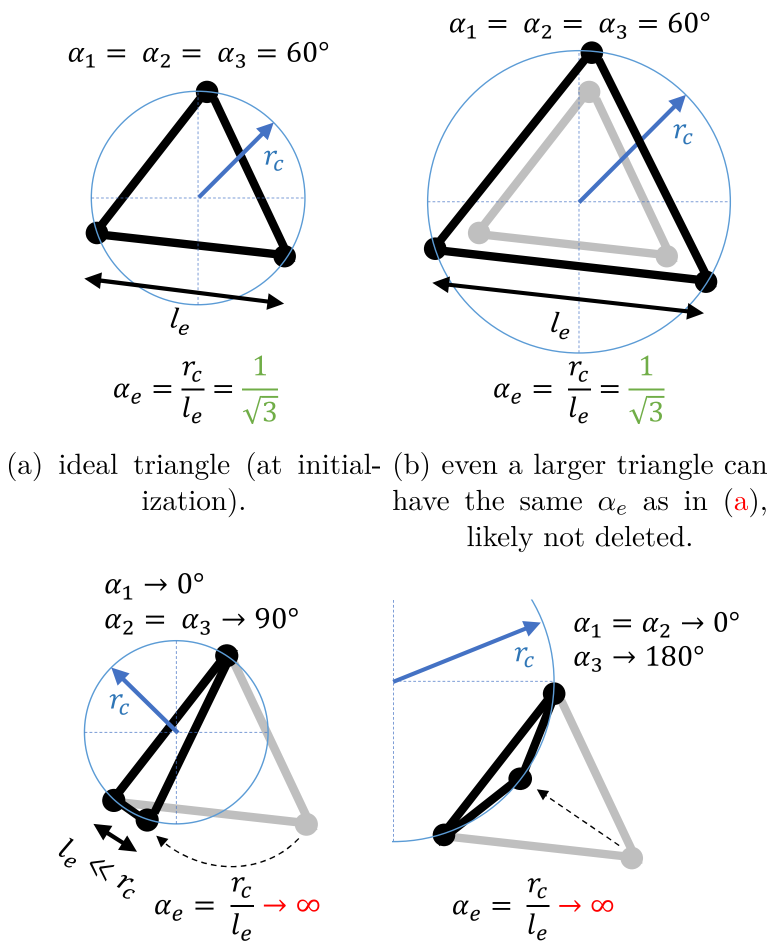
(b) a larger triangle has a larger α_e, triangle may be deleted.

(c) needle-like slivers can have a normal α_e, may not be deleted.

(d) flat slivers have large α_e, likely to be deleted.

FIGURE 2.20: Effect of triangle shape and size on α_e for a uniform mesh ($\bar{l}_e = \bar{l}_{e0}$).

and an in-depth analysis can be found in the articles of Falla et al. [96] and Franci and Cremonesi [42]. Recommendations regarding the minimization of the error are given therein. In this work, the recommendations are followed as much as possible.



(a) ideal triangle (at initialization).

(b) even a larger triangle can have the same α_e as in (a), likely not deleted.

(c) needle-like slivers have a large α_e, likely to be deleted.

(d) flat slivers have large α_e, likely to be deleted.

FIGURE 2.21: Effect of triangle shape and size on α_e for a non-uniform mesh ($\bar{l}_e = l_e$).

2.7.3 PFEM remeshing algorithm

The basic remeshing in PFEM is achieved by combining a Delaunay triangulation with the α-shape method. The latter is used to discard triangles from the triangulation that are not needed. As explained in section 1.3, this combination allows to simulate free surface flows without a boundary tracking algorithm.

There are many approaches to storing, accessing and modifying mesh information. In our implementation, the mesh can be thought of as lists of the mesh entities: nodes, edges and triangles. Each entity has information on other entities that are connected to it: each node knows all its connected edges and triangles, each triangle knows its 3 edges and nodes, each edge knows its 2 nodes and the one or two triangles on either side. This arrangement is very heavy on memory, but it allows to quickly and easily access most of the information relevant to any of the mesh entities. Keeping them in a global list allows for easy looping of any of the mesh entities over the entire mesh.

For the treatment of phase change in a welding or additive manufacturing

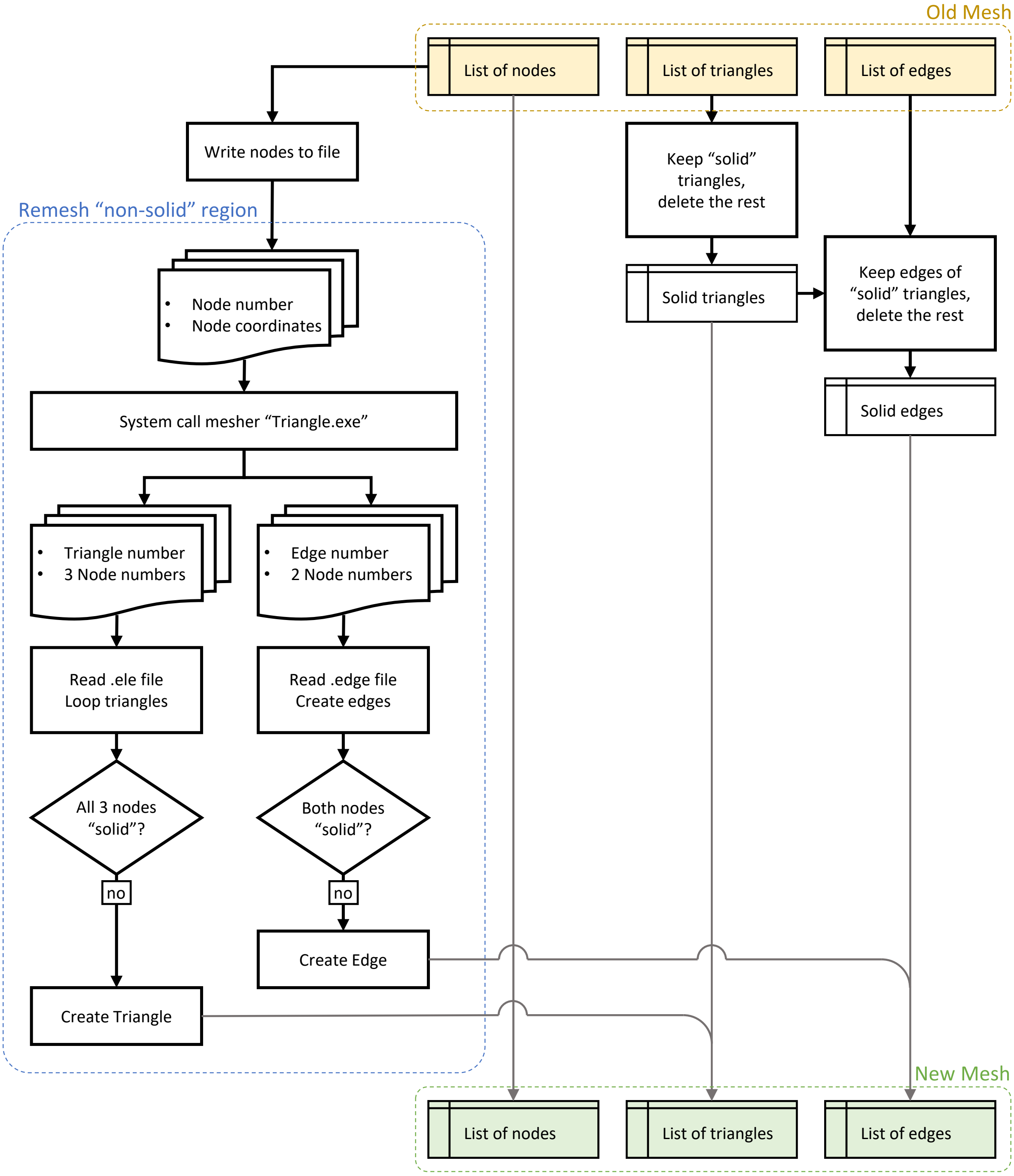


FIGURE 2.22: Remeshing algorithm that preserves solid regions.

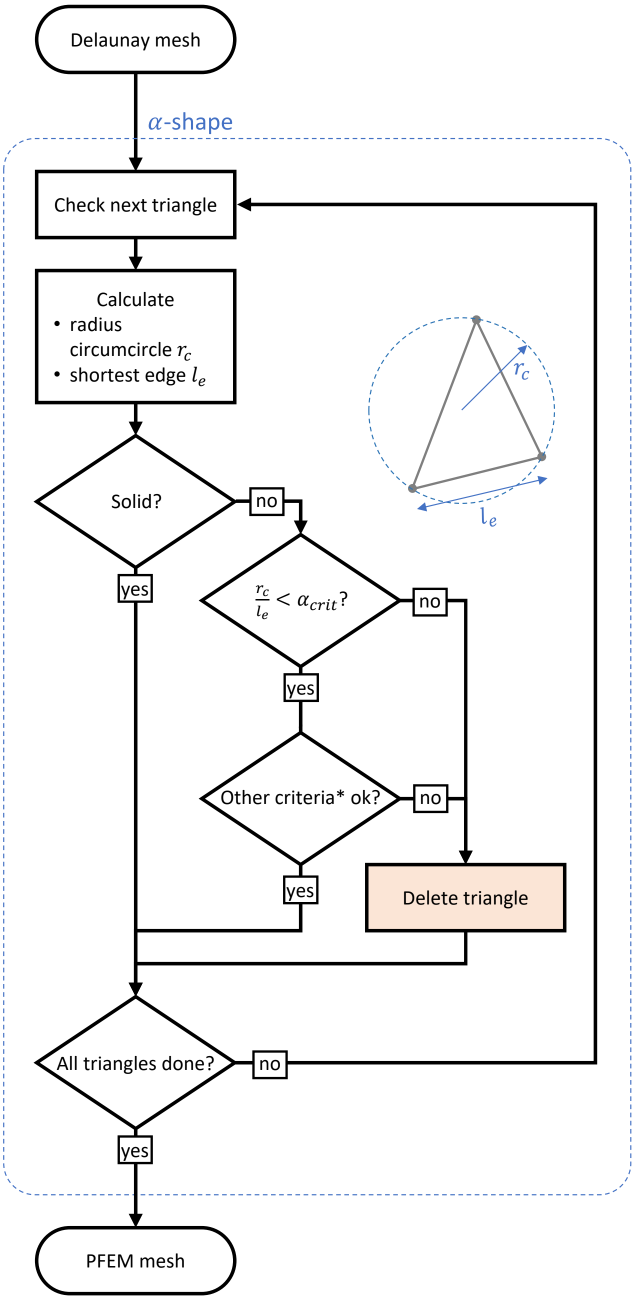


FIGURE 2.23: The α-shape method. *The "other criteria" refer to additional techniques to improve the mesh quality that are not part of the traditional α-shape technique.

context, we assume that the solid region does not undergo large displacements and that the solid domain can not be split (fracture) or come into (self-)contact. Under these assumptions, it is possible to exempt the regions of the mesh that are considered "solid" from the remeshing process. Not deleting these triangles at every remeshing allows to keep the solid deformation history in memory without additional effort. How a triangle is marked as "solid" is described further down. The Delaunay triangulation that is extended to preserve the solid region of the mesh is illustrated in fig. 2.22.

The α-shape method that is extended to better preserve a good mesh quality and to ignore "solid" elements is illustrated in fig. 2.23. The only user input is the desired value for α_{crit}, which from experience should be set to $\alpha_{crit} = 1.2$. In the flow chart, there are "other criteria" mentioned, next to the α-shape criterion. This is a placeholder for the multitude of small individual tweaks that aim to alleviate problems with the mesh that the α-shape cannot prevent. An example for such a tweak would be to prevent the α-shape to delete sliver elements inside the bulk, even if they violate the α-shape criterion. The remeshing algorithm would then get another chance to delete ill-placed nodes in the bulk to solve this problem or to add nodes in strategic places. Deleting a badly shaped element would otherwise create a hole in the bulk, which leads to many new problems. There, it makes more sense to tackle the problem at the root and improve the element than to blindly apply the α-shape algorithm. Many more of such additional mesh management techniques used in this work are presented by Falla et al. [96]. The majority there aims to preserve a good quality at the free surface boundary, where the α-shape algorithm is not sufficient.

2.7.4 Adaptive remeshing

The adaptive remeshing allows to control the local mesh density based on some given criteria. In our work, there are two types of criteria: geometric and physical. A geometric criterion defines a geometric region (e.g. a circle, box, line etc.) in which a certain mesh density must be achieved and also a transition of the mesh density from the specified region to the rest of the mesh. A physical criterion is a solution-based criterion, where a given physical quantity (e.g. T, $\nabla \boldsymbol{v}$ etc.) is used to control the mesh density. For the free surface boundary and for slip boundaries, there are a number of other mechanisms that modify the mesh that all aim to preserve a good quality of these boundaries. A mesh refinement is achieved by simply adding more nodes, where the mesh density must be increased. Coarsening, conversely, is achieved by removing nodes from a region that requires a lower mesh density. The overall process is illustrated in fig. 2.24

It is obvious that the adaptive remeshing, which uses 3 individual Delaunay triangulations and subsequent runs of the α-shape technique is computationally costly. However, it keeps the algorithm fairly simple because refinement (i.e. adding nodes) and coarsening (i.e. removing nodes) remain independent of one another and new mesh manipulation algorithms can be integrated without much effort.

The algorithms for adding nodes and deleting nodes are given in figs. 2.25 and 2.29. These two algorithms are used to adjust the mesh density by adding and removing nodes such that a given target edge length L^* is achieved everywhere. In the absence of any mesh refinement criteria, a uniform L^* and a **uniform mesh** is obtained. In this case, the target mesh size is simply evaluated from the initial mesh that the user provides and remains constant throughout the entire simulation run:

$$L^* = \left[\frac{1}{n_e} \sum_{e=1}^{n_e} l_e \right]_{t0} \tag{2.195}$$

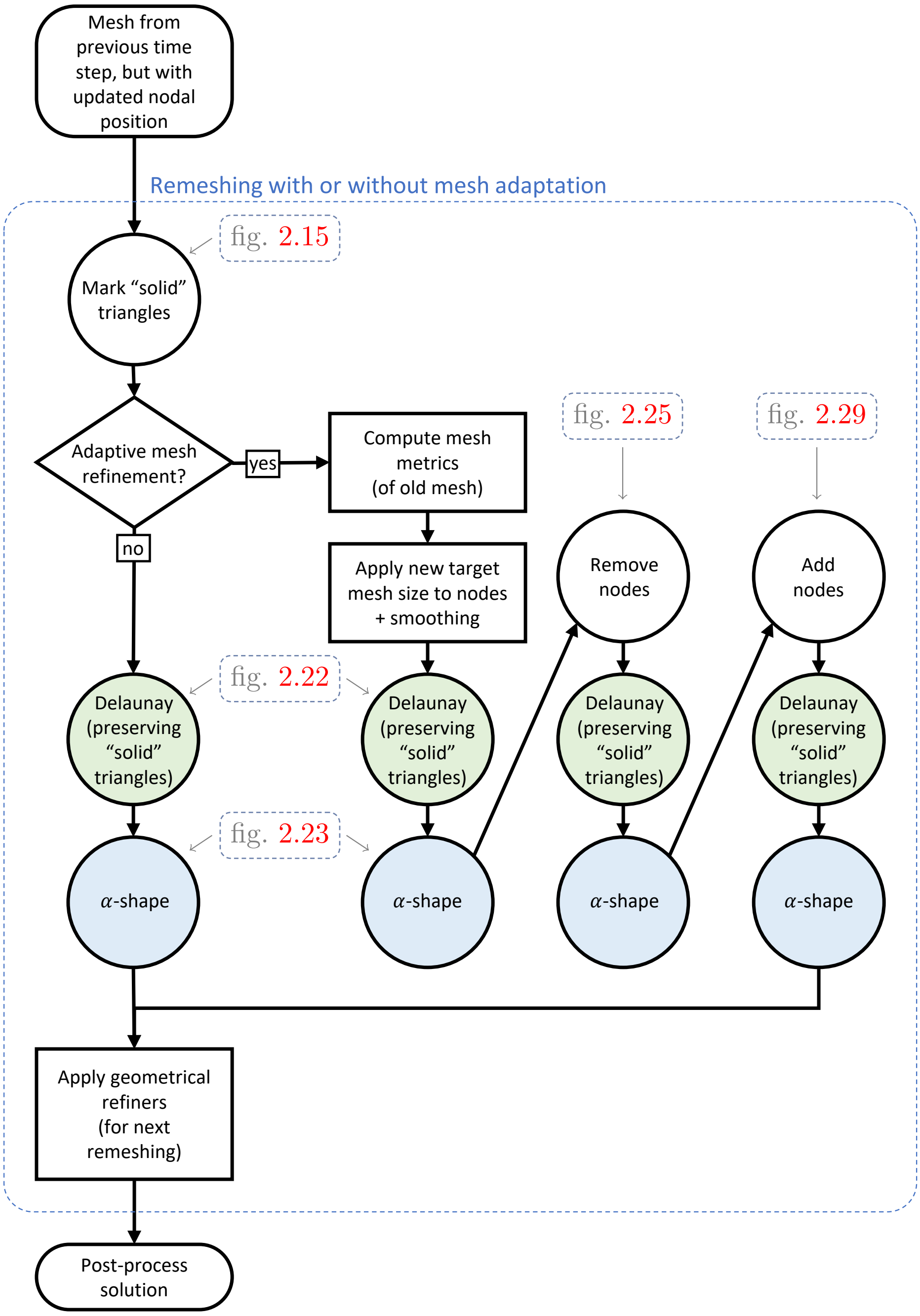

Mesh from previous time step, but with updated nodal position
Remeshing with or without mesh adaptation
fig. 2.15
Mark "solid" triangles
Adaptive mesh refinement?
yes
no
Compute mesh metrics (of old mesh)
Apply new target mesh size to nodes + smoothing
fig. 2.25
fig. 2.29
Remove nodes
Add nodes
fig. 2.22
Delaunay (preserving "solid" triangles)
fig. 2.23
α-shape
Apply geometrical refiners (for next remeshing)
Post-process solution

FIGURE 2.24: Adaptive remeshing algorithm.

where n_e is the number of edges, l_e is the length of an edge e. Nodes are therefore added and removed to preserve a good mesh quality, even in the absence of any mesh refinement. If at least one refiner is defined by the user, a **non-uniform mesh** is obtained and the target mesh size L_I^* at each node I is

$$L_I^* = \min(L_{I_r}^*) \tag{2.196}$$

where $L_{I_r}^*$ is the individual target mesh size at node I from each refiner r. The mesh density is dictated by the refiner with the lowest target to be conservative. Depending on the user's choice, each independent refiner can produce a target mesh size as a function of position $\boldsymbol{x_I}$, magnitude of velocity $\|\boldsymbol{v_I}\|$ or magnitude of its gradient $\|\nabla \boldsymbol{v_I}\|$, temperature T_I or magnitude of its gradient $\|\nabla T\|$ or surface curvature κ_s. Any number of refiners can be defined. Each refiner has limiters L^*_{min} and L^*_{max} that restrict $L^*_{I_r}$ from going below or above a defined interval respectively. Refiners may also be limited to act within a specific region only. Each refiner should be defined such that a smooth transition from fine to coarse is imposed on the mesh. More details on available refiners and recommendations regarding their parameters and how to combine them can be found in the recent article by Falla et al. [96]. In the present work, the used refiners are pointed out on a case-by-case basis.

The **deletion of nodes** is accomplished by looping over all nodes and deciding if the current node I shall be deleted or not.

Firstly, nodes currently marked as "solid" are not deleted because the mesh of the solid region must be preserved in its current state. This status can, however, change before the next remeshing and is by no means permanent. Some other nodes have a permanent "protected" status and are never deleted. There are 2 relevant types of protected nodes:

1. Nodes that are specifically selected to sample and output data, using a so-called nodal extractor. This type of extractor is rarely used.
2. The node on the corner where a free surface and a free-slip boundary meet, as deleting such a node causes an instability that deteriorates the solution.

A node on a Dirichlet boundary of the mechanical problem Γ_v is only deleted when it is too close to another node on the same boundary. If it is too close to a node in the bulk, the node in the bulk will always be deleted, never the boundary node.

Next, if the node is allowed to be deleted, two very inexpensive checks are performed. Firstly, if the node is outside a user defined bounding box, it is deleted (fig. 2.26a). The deletion of nodes outside the bounding box reduces the computational cost at the very least, but may also prevent certain serious problems that destabilize the solution. Secondly, the user can chose to remove isolated nodes, i.e. nodes without any edge or triangle associated (fig. 2.26b). Again, this may just reduce computational cost or may prevent excessive unphysical mass creation.

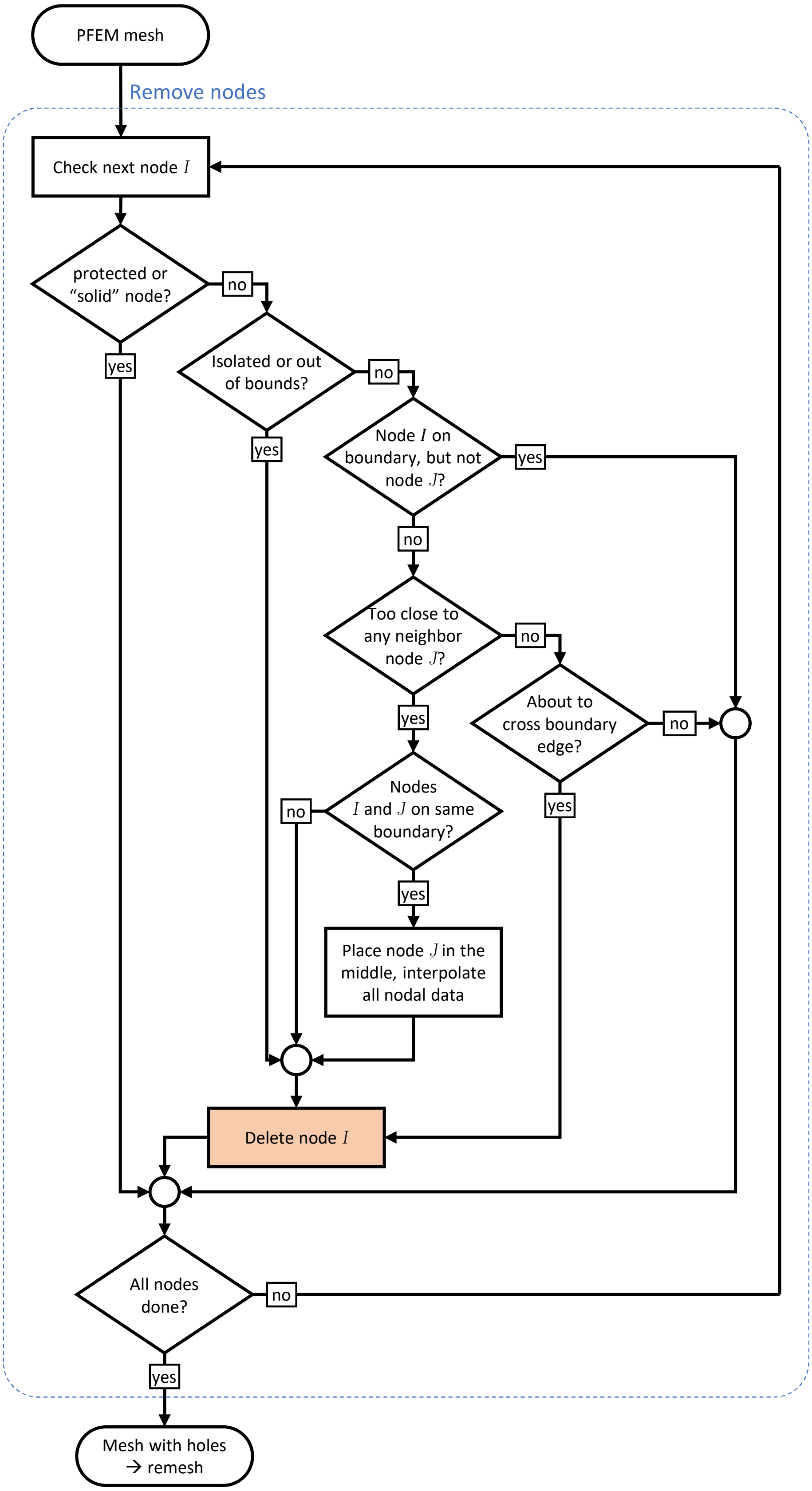
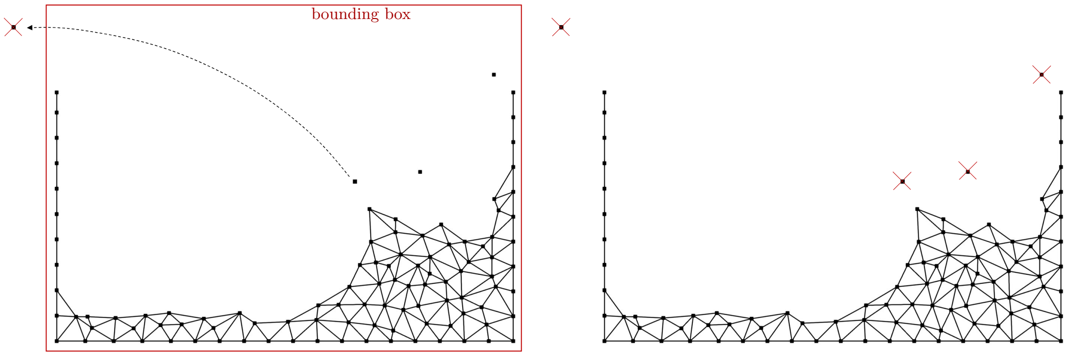

PFEM mesh
Remove nodes
Check next node I
protected or "solid" node?
no
yes
Isolated or out of bounds?
no
yes
Node I on boundary, but not node J?
yes
no
Too close to any neighbor node J?
no
yes
About to cross boundary edge?
no
yes
Nodes I and J on same boundary?
no
yes
Place node J in the middle, interpolate all nodal data
Delete node I
All nodes done?
no
yes
Mesh with holes → remesh

FIGURE 2.25: Algorithm for removing nodes.



(a) Option: delete nodes outside bounding box.

(b) Option: delete separated nodes.

FIGURE 2.26: First set of (optional) node deletion criteria.

Node I that survives these checks is now assessed, whether it needs to be removed to preserve the local target mesh size L_I^*.

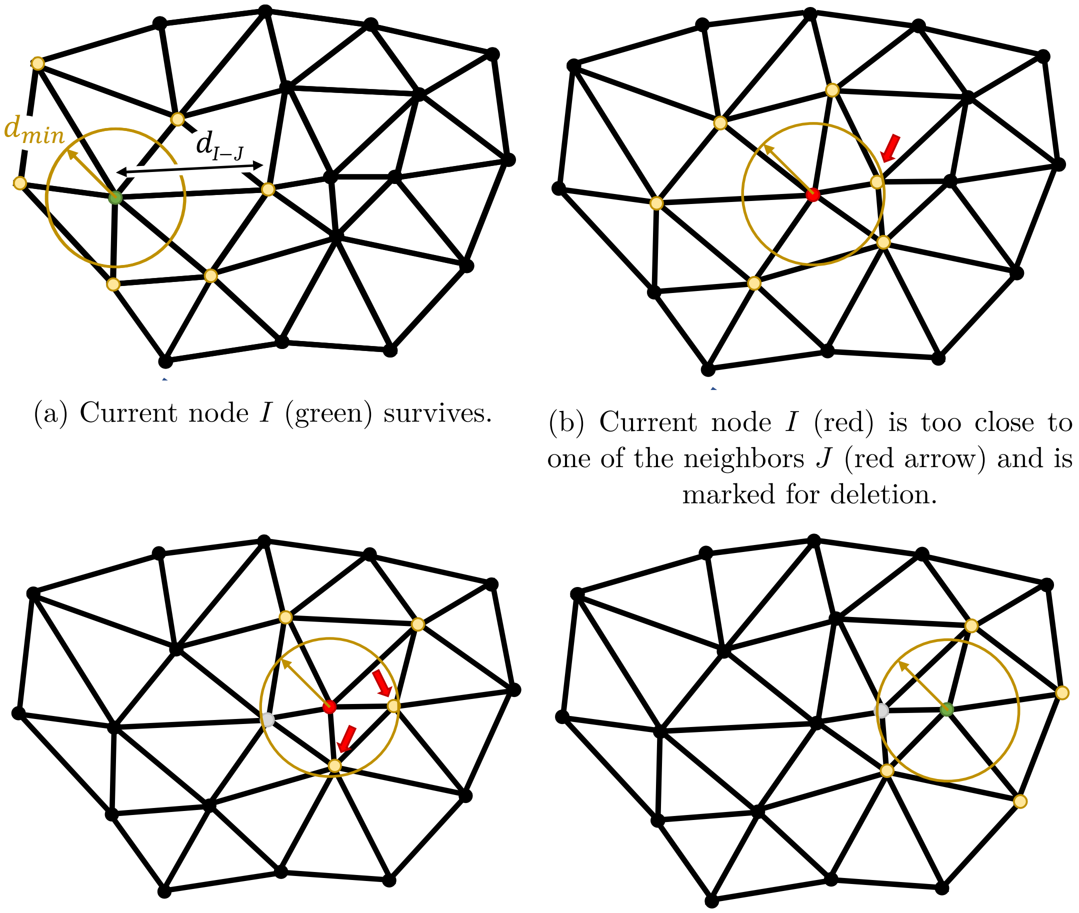
$$d_{I-J} \geq d_{min} = \gamma_{crit}\, L_I^* \quad \forall \text{ valid nodes } I \tag{2.197}$$

where d_{I-J} is the distance between node I and each of its neighbors J, d_{min} is a minimal admissible distance and γ_{crit} is a user-chosen parameter that quantifies, how much denser a mesh can be than the target mesh density. By default $\gamma_{crit} = 0.7$. If eq. 2.197 is violated, only node I is marked for deletion. Previously marked nodes are not taken into account when they are now the neighbor J to another node I, as illustrated in fig. 2.27.

The user can also choose to modify γ_{crit} to a slightly larger value (i.e. a stricter adherence to the target mesh density), when I and J lie on a free-slip or free surface boundary. There, nodes can get too close to another more easily than in the bulk, resulting in long, thin sliver elements. Choosing $\gamma = 1.0$ on these boundaries, for example, would result in less narrowly shaped triangles, which preserves the quality of the boundary. This is especially useful in this work, where the Marangoni effect causes a localized acceleration of free surface boundary nodes that tends to accumulate surface nodes too densely.

Likewise, a better quality of the boundary triangles is also achieved, if after deletion of node I, the other node J is moved to the middle of the connecting edge. This also results in better shaped boundary elements. All nodal data of both I and J is interpolated and written to node J, to minimize the error caused by moving the node. This appears to add less unwanted numerical diffusion than simply deleting node I and leaving node J unchanged.

If node I has not been deleted, the final check is whether it may be approaching a boundary. Normally, if a node gets close to a boundary, it would eventually get too close to one of the boundary nodes and the previous mechanism would eliminate the approaching node (figs. 2.28a,b). However, nodes can approach a boundary edge straight down the middle, equidistant from the two nodes that bound the edge, but with sufficient distance, such that it is not deleted (figs. 2.28c - e). Therefore, a second criterion is introduced that utilizes the distance of node I to the boundary edge, instead of a neighboring node J. The



(a) Current node I (green) survives.

(b) Current node I (red) is too close to one of the neighbors J (red arrow) and is marked for deletion.

(c) Current node I (red) is too close to two neighbors J (red arrows) and is marked for deletion. Previously marked node (grey) is not considered.

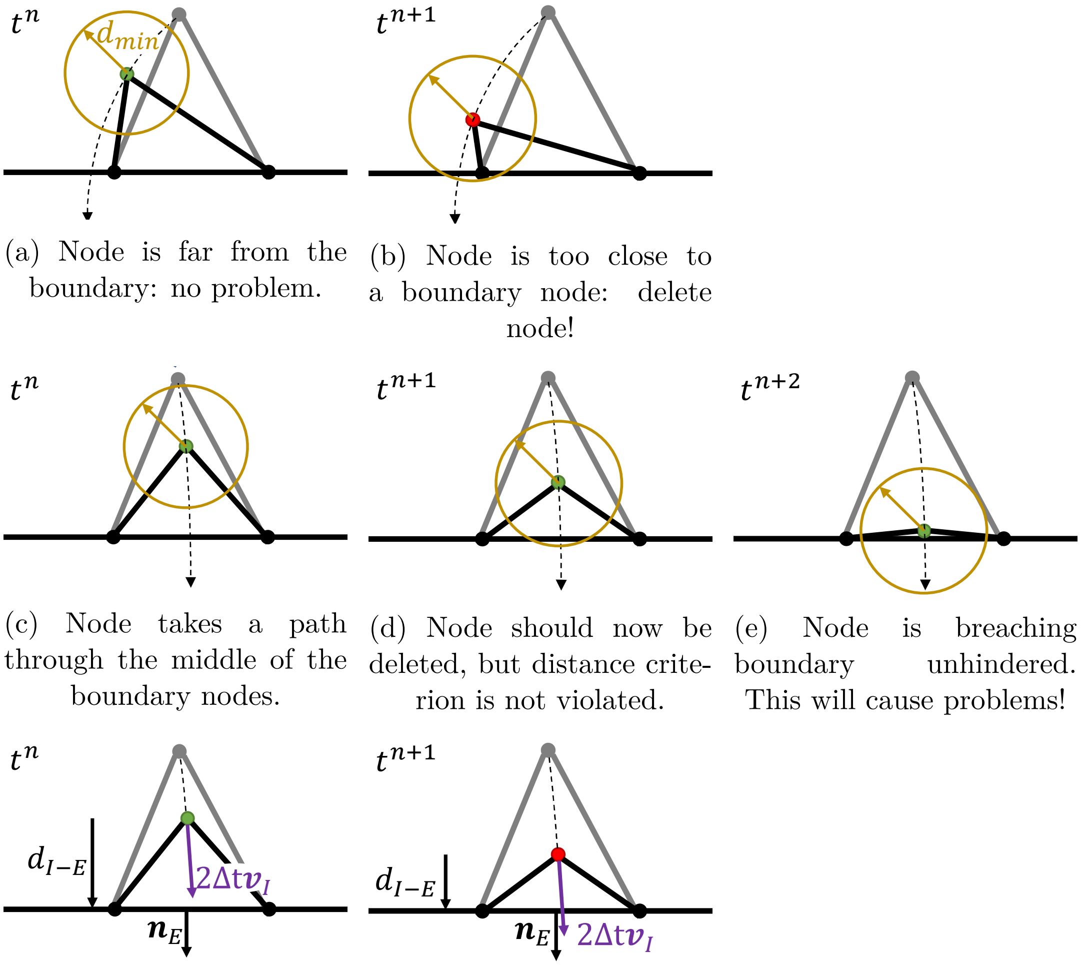
(d) Current node I (green) survives because previously marked node (grey) is not considered.

FIGURE 2.27: Deletion of too close nodes. Checking each node I (red or green), if it is too close to any neighbor J (yellow). If not, node survives (green) and if so it is marked for deletion (red) and becomes excluded from following comparisons (grey). Note that allowed distance d_{min} is shown as constant in the example, while it can change depending on the refiners used.

first step is to loop over all triangles that connect to node I. In each triangle, we then find the opposite edge and check, if this edge is labeled as a boundary edge. If this is the case, the distance d_{I-E} between node I and boundary edge E is determined, as well as the node's velocity vector $\boldsymbol{v}^I$ and the edge's unit outward normal $\boldsymbol{n}^E$. We can then predict, if the node is likely to penetrate the boundary within 2 time steps Δt and eliminate it (figs. 2.28f,g). The deletion criterion is

$$d_{I-E} \geq 2\Delta t \boldsymbol{v}_I \cdot \boldsymbol{n}_E \; \forall \text{ valid nodes} \tag{2.198}$$

Two time steps are needed because the remeshing takes place at the beginning of a time step, so that a node that survives this check in the current time step can safely move once more towards the boundary before the mesh is too distorted.



(a) Node is far from the boundary: no problem.

(b) Node is too close to a boundary node: delete node!

(c) Node takes a path through the middle of the boundary nodes.

(d) Node should now be deleted, but distance criterion is not violated.

(e) Node is breaching boundary unhindered. This will cause problems!

(f) Node will not reach boundary within two timesteps: no problem.

(g) Node will breach boundary within two time steps: delete node!

FIGURE 2.28: Deletion of nodes breaching a boundary: a node travels from its initial position (grey) towards a boundary. Depending on the path taken, the distance criterion, eq. 2.197, may fail (middle row) and the criterion taking the velocity towards the boundary (eq. 2.198) is needed (bottom row). Green node survives, red node is deleted.

It is implicitly assumed that the current nodal velocity $\boldsymbol{v}_I$ is maintained, which may not be the case. If the velocity proves to be faster, the node may actually breach the boundary in the current time step, even if the deletion criterion was not met. Furthermore, a node that almost penetrates a boundary right in the middle between two existing boundary nodes will form an excessively flat triangle, which is to be avoided as well. Hence, 2 time steps appear to be a reasonably conservative approach to mitigate this well-known issue of PFEM. Note that this detection algorithm relies on the existence of a triangle between the approaching node I and the boundary edge E. It is a common occurrence in cases with a free surface and splashing that single nodes fly at high speed through the air, over the bulk of fluid towards a rigid boundary (e.g. a container wall). Such nodes do not form a triangle element with the boundary 2 time steps in advance, maybe not even in the current time step. One solution is to let the fast node breach the boundary and be eliminated by the bounding box. Another solution is to reduce the time step size to essentially increase the sampling rate at which nodes get a chance to form a triangle with the boundary edge so that this mechanism can detect this node.

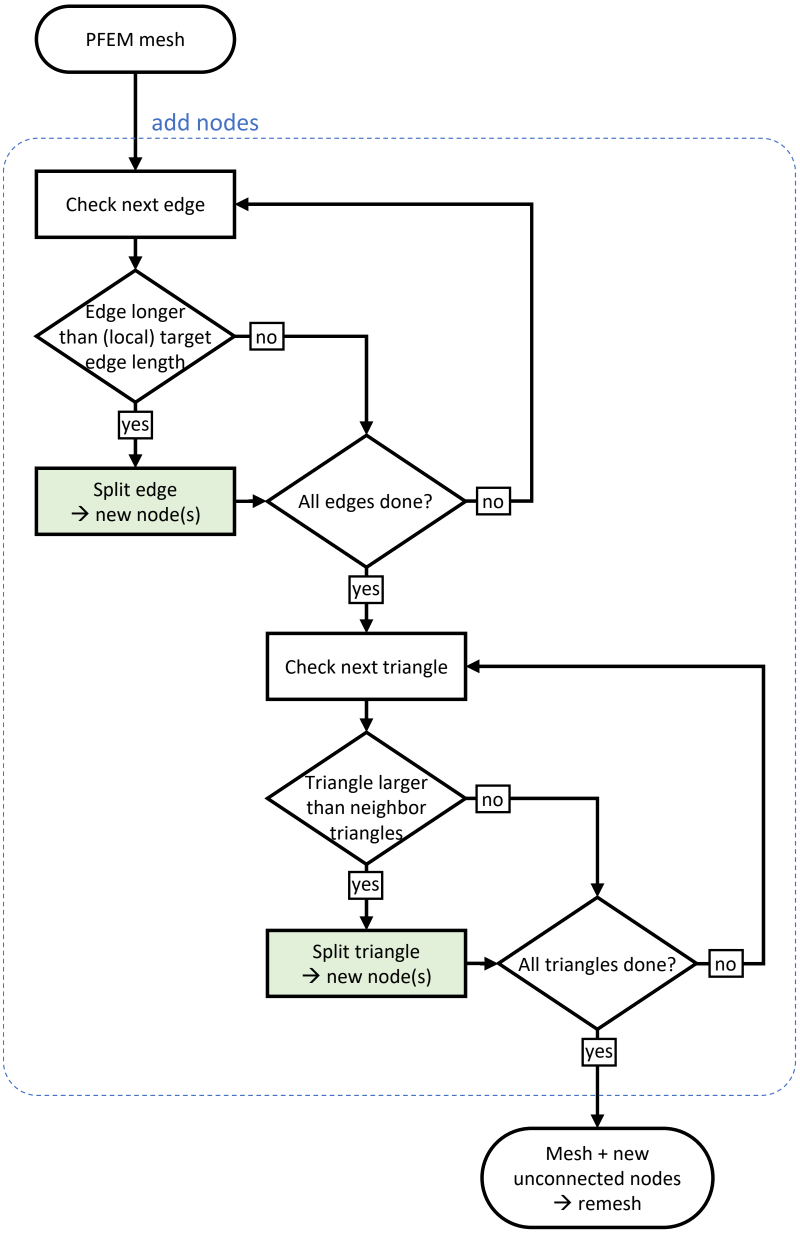

PFEM mesh
add nodes
Check next edge
Edge longer than (local) target edge length
no
yes
Split edge → new node(s)
All edges done?
no
yes
Check next triangle
Triangle larger than neighbor triangles
no
yes
Split triangle → new node(s)
All triangles done?
no
yes
Mesh + new unconnected nodes → remesh

FIGURE 2.29: Algorithm for adding nodes.

The **addition of nodes** is simpler, as evident from the flow chart in fig. 2.29. A first loop identifies edges that are too long, according to the criterion

$$l_E \leq l_{max} = \omega_{crit} \sum_{I=1}^{2} L_I^* \quad \forall \text{ valid edges } E \tag{2.199}$$

where l_E is the length of an edge E, and l_{max} is a allowed maximum length. This criterion is similar to the node removal criterion (eq. 2.197), except that it is indeed checked once for each edge instead of being checked for each combination of node I and neighbor node J. This is simply because the node deletion must be related to a given node, while the node addition must be related to some sort of empty space in which to insert an additional node. The empty space here is a long edge (further down it will be a large triangle). Similar to the node deletion, the allowed distance d_{max} depends on a user-chosen parameter ω_{crit} and the mean target mesh size of both nodes I on edge E. In this work $\omega_{crit} = 0.7$ is the default value.

This edge-refining criterion is especially important at free surface boundaries, where the boundary is stretched out. Stretching of the surface creates flat triangles that are deleted by the α-shape technique. This degrades the free surface and the addition of a node onto a stretched edge can prevent this degradation of the free surface.

A similar mechanism exists for excessively large triangles, where one node is inserted in the centroid of the triangle to increase the mesh density.

$$A_T \leq A_{max} = 2\omega_{crit} \left(\frac{1}{3} \sum_{I=1}^{3} L_I^* \right)^2 \quad \forall \text{ valid triangles } T \tag{2.200}$$

where A_T is the area of a triangle element T and A_{max} is the maximum allowed area based on the mean nodal target mesh size L_I^* of the triangle's nodes I. More specialized criteria are also implemented (see Falla et al. [96]), but not used in this work, because they are not relevant to any of the test cases presented further down.

Newly added nodes are initialized with all nodal data (e.g. $\boldsymbol{x}_I, \boldsymbol{v}_I, p_I, T_I, L_I^*$) linearly interpolated from the original edge or triangle nodes.

2.7.5 Managing the fluid-solid interface

If an interface is needed, it is created after the remeshing and after the thermal solving process (see flow chart in fig. 2.13) because both these steps may alter the location of the interface.

At this point, the interface is identified by checking for each edge in the domain, whether it has a fluid and a solid element on either side (see identification of solid elements in section 2.6.1). If yes, the edge belongs to the interface and the edges nodes and the edge itself are duplicated. The connectivity of elements, edges, lines and nodes is organized such that the original domain is split along the interface and two separate sub-domains are established. Fig. 2.30 illustrates the interface creation process. The removal of the interface is the exact inverse order of operations.

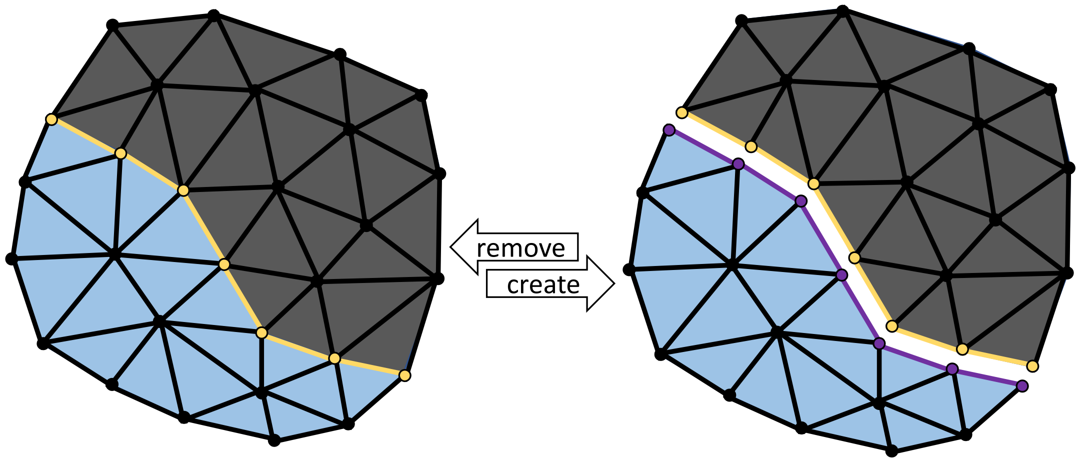


Figure 2.30: Domain splitting and node duplication operation and the interface. Yellow edges and nodes are identified as the interface and are duplicated (purple). After re-connecting nodes, edges and triangles, two separate sub-domains are created. The splitting is later reversed exactly.

Chapter 3

Verification

This chapter aims to provide compelling evidence that our method can deliver accurate results in very complex multiphysics simulations, such as laser welding (see chapter 4). For this purpose, we will verify the implementation of some of the physics that are less common in a PFEM context. The verification is conducted for each of the physical phenomena in isolation and in combination, where possible. Previous simulations from the literature, as well as experimental and analytical results, wherever available, are taken into consideration for this.

This work builds on previous developments in our group by Cerquaglia, whose doctoral thesis [39] lays the groundwork for this work, both scientifically and from a code implementation point of view. The PFEM code he wrote was entirely limited to isothermal fluid dynamics, specialized on complex free surface deformation, such as sloshing and dam break (see Cerquaglia et al. [97]) or the impact of a fluid on a wall (see Cerquaglia et al. [9]). His research headed in 2 main directions: firstly the solving of the Navier-Stokes equations through fractional step schemes instead of a monolithic scheme. This can be computationally cheaper and avoids the need for stabilization. Secondly, the coupling of the present PFEM code for fluid dynamics with other solid mechanics codes to simulate fluid structure interaction (FSI) problems (see [9, 90] and chapter 3 of the thesis [39]). This leaves us with the starting point of the verification of this work, a Lagrangian PFEM code with the following characteristics and capabilities:

- 2D plane strain or axisymmetric,
- Newtonian fluids only,
- Incompressible only,
- Free surface detection using the α-shape method,
- Surface tension normal component,
- Uniform mesh only (i.e. no mesh refinements).

This includes the framework for handling time marching, non-linear iteration, remeshing, data input and output etc. All these implementations can be considered verified by Cerquaglia.

New implementations therefore include the addition of the temperature field, the solid behavior and several relevant physical effects in the form of boundary conditions, source terms, external forces:

- Surface tension tangential component (Marangoni effect),
- Convection and radiation boundary conditions,
- Heat source to model laser interaction,
- Phase change and latent heat,
- Model to approximate solid behavior (fix nodes in space using a Carman-Kozeny equation),
- Isotropic linear elastic solid behavior (bulk modulus and shear modulus).

First, we introduce thermal capabilities and test them in a static domain, then the flow is added along with thermal expansion, before introducing phase change. Second, the linear elastic solid and finally the unified fluid-solid formulation with an interface is verified. In this fashion, the complexity is slowly increased until all the capabilities are combined for the demostration test cases in chapter 4.

3.1 Static thermal

Purely thermal problems are solved here to verify and validate the implementation. There is no movement of any of the nodes and no remeshing, which means the PFEM code is performing classic FEM simulations. For these types of problems, it is useful to define the thermal diffusivity α

$$\alpha = \frac{k}{\rho c_p} \tag{3.1}$$

which characterizes the transient heat transfer property of a material and the Fourier number Fo

$$Fo = \frac{\alpha t_c}{l_c^2} = \frac{k t_c}{\rho \, c_p \, l_c^2} \tag{3.2}$$

where l_c is a characteristic distance through a material or thickness of a material over which heat is conducted and t_c is a characteristic time scale. Fo is the ratio between conductive heat transfer rate and rate of heat storage and therefore describes transient heat transfer problems. Fo can also be used as a dimensionless time and a large value describes a situation where a material reaches thermal equilibrium.

3.1.1 Thermal boundary condition verification

There are 4 types of thermal boundary conditions (BC) used in this work:

- Imposed temperature (Dirichlet BC),

- Imposed surface heat flux (Neumann BC),
- Surface convection (Robin BC) and
- Surface radiation (Robin BC).

Not imposing any BC automatically implies an adiabatic boundary (Neumann BC with $\bar{q} = 0$). All boundary conditions are verified in several different scenarios:

- Over entire boundary or over a section of a boundary,
- In 2D plane strain or in 2D axisymmetric,
- On a regular mesh, irregular mesh or with mesh refinement/coarsening,
- Constant magnitude from start to finish or as a function of time and
- Each BC isolated or several in combination.

Combinations of thermal boundary conditions are for example an imposed heat flux at one wall and a radiation heat flux at another, where the resulting equilibrium temperature can be easily determined. Such tests were performed on numerous academic examples, but none of these are shown here, as all these boundary conditions are used also in other more complex test cases that the reader might find more interesting.

3.1.2 1D transient heat conduction

A long, thin rectangle of width $w = 5\,\mathrm{m}$ and height $h = 1\,\mathrm{m}$ is initialized with a constant low temperature T_0 and a hot wall on the left of constant temperature $\bar{T}$ and all other walls being adiabatic $\bar{q} = 0$, as schematized in fig 3.1. The material properties are given in table 3.1.

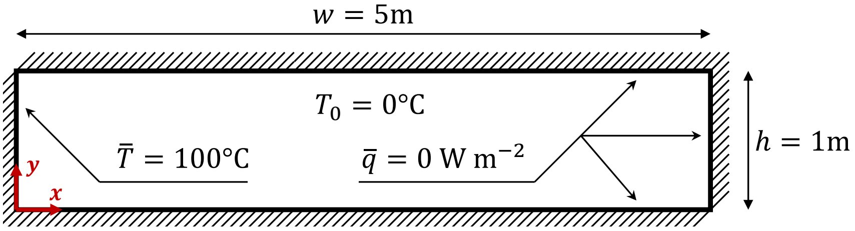


FIGURE 3.1: Schematic of the static 1D heat conduction test case.

The heat flows into the system, gradually raising the temperature following this analytical 1D solution for a semi-infinite domain easily found in literature (see e.g. [98], p. 34 - 36:

$$T(x,t) = \left(T_0 - \bar{T}\right) \operatorname{erf}\left(\frac{x}{\sqrt{4t\alpha}}\right) + \bar{T} \tag{3.3}$$

where erf is the well-known error function. The process is simulated using PFEM until a total time $t_{end} = 1.0 \times 10^5\,\mathrm{s}$ with a constant time step $\Delta t =$

TABLE 3.1: Material properties.

Property	**Symbol**	**Value**	**Unit**
Heat capacity	c_p	500.0	$\mathrm{J\,kg^{-1}\,K^{-1}}$
Conductivity	k	16.3	$\mathrm{W\,m^{-1}\,K^{-1}}$
Density	ρ	8000.0	$\mathrm{kg\,m^{-3}}$
Thermal diffusivity	α	4.075×10^{-6}	$\mathrm{m^2\,s^{-1}}$

1.0×10^3 s for a good resolution at the beginning of the simulation. An irregular mesh with a mean edge length $\bar{l}_e = 0.25$ m and 129 nodes is used. The resulting temperature field is depicted below in fig. 3.2.

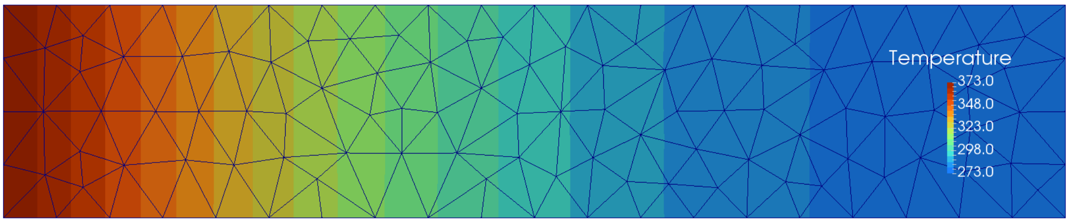


FIGURE 3.2: Final Temperature field ($t = t_{end}$.

The time evolution of the average temperature along $x = 0.5$ m and $x = 1.0$ m is compared against the analytical solution and our group's non-linear thermo-mechanical FEM code *Metafor* [91]. The comparison is depicted in fig. 3.3 and 3.4. An excellent agreement between the different results is achieved, even on this coarse mesh and with a large temperature gradient at the beginning of the simulation.

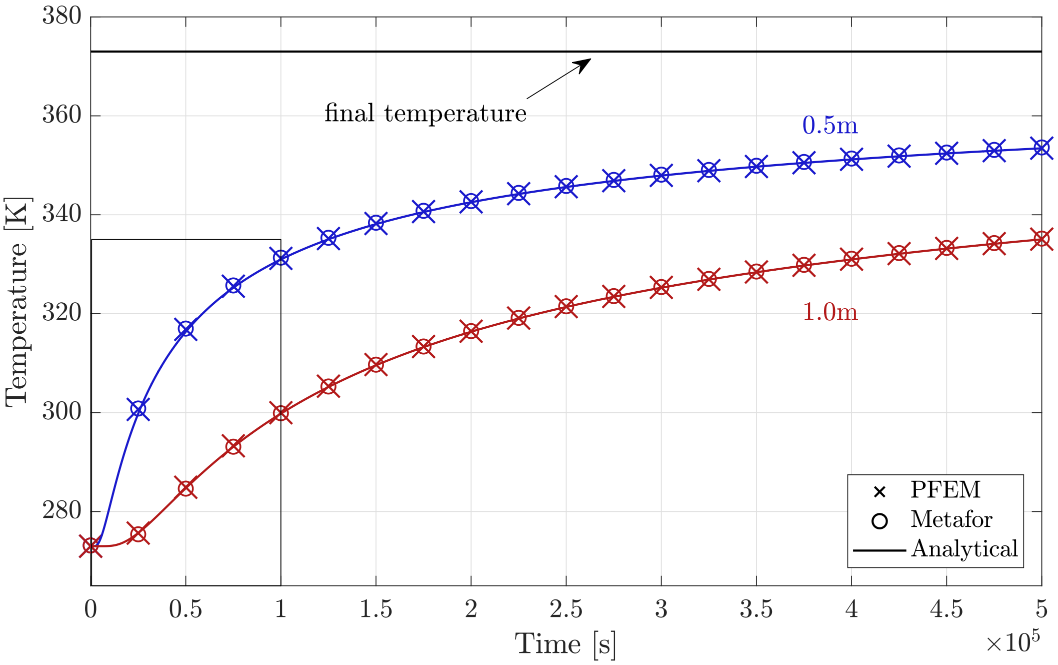


FIGURE 3.3: Temperature over time at two distances $x = 0.5\,\mathrm{m}$ and $x = 1.0\,\mathrm{m}$. Comparison of this work, commercial FEM code Metafor and the analytical solution.

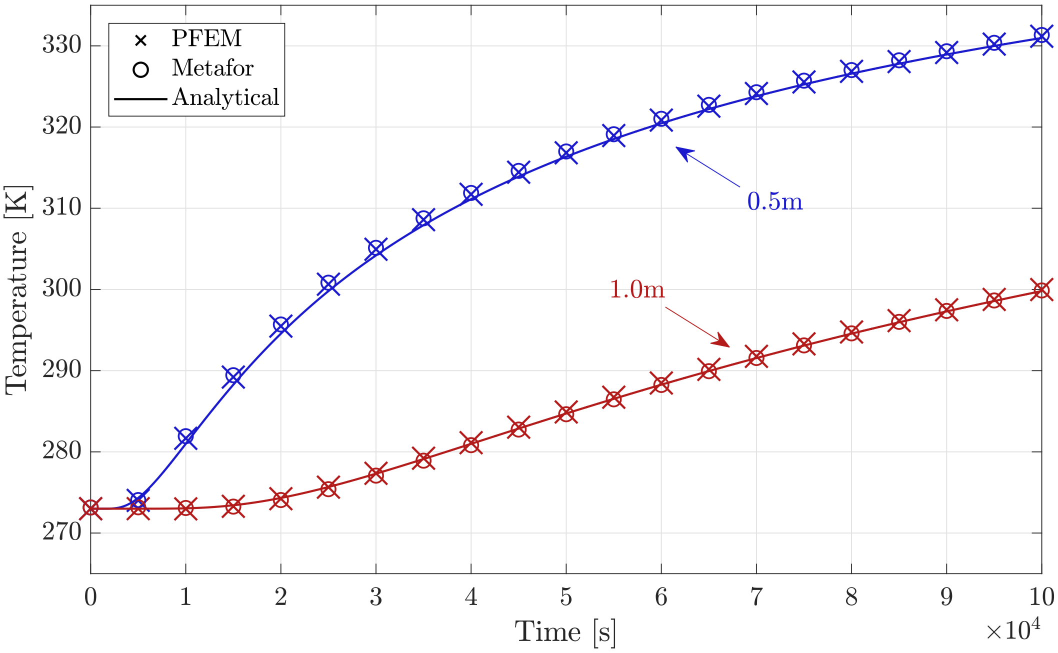


FIGURE 3.4: Detailed view of highly transient start-up phase in fig. 3.3.

3.2 Thermo-fluid

In a thermo-fluid, temperature and flow interact in some form and at each time step, both the Navier-Stokes equations and the heat equation are solved (see section 2.6). This does not just allow for convective heat transfer to be modeled, but also allows some new physical effects to be added. The relevant ones for welding and additive manufacturing are buoyancy and the Marangoni effect, both of which act as driving forces of the convective flow. Buoyancy is a result of density differences in the same fluid due to thermal expansion. Thermal expansion not only affects the density (in the sense of a material property), but also the volume the material occupies (in the sense of a kinematic effect). Both effects can be modeled separately (see section 2.3.3), which allows us to gradually increase the complexity of the problems by including more and more of these physical effects.

Important dimensionless numbers are the Prandtl number Pr, the Rayleigh number Ra, the Nusselt number Nu and the Marangoni number Ma (Mg is also used occasionally to avoid confusion with the Mach number). Pr is the ratio between momentum diffusivity and thermal diffusivity and is defined as

$$Pr = \frac{\mu}{\rho\,\alpha} = \frac{c_p\,\mu}{k} \tag{3.4}$$

where μ is the dynamic viscosity, ρ the density and α the thermal diffusivity (eq. 3.1). Pr is particularly important for characterizing convective heat transfer and comparing thermal and momentum boundary layers. It purely depends on material properties: Liquid metals tend to have a low Pr in the order of 10^{-2} to 10^{-3}, air has $Pr = 0.71$ and water has $Pr \approx 7.6$. Ra is the ratio of convective flux to conductive flux and characterizes the flow regime in the boundary layer of buoyancy driven flows (i.e. natural convection) and is defined as

$$Ra = \frac{\rho\,\alpha_V\,\Delta T_c\,l_c^3\,g}{\mu\,\alpha} = \frac{\rho^2\,c_p\,\alpha_V\,\Delta T_c\,l_c^3\,g}{\mu\,k} \tag{3.5}$$

where α_V is the volumetric thermal expansion coefficient, ΔT_c is a characteristic temperature difference, l_c is characteristic length, g is gravity. The Nusselt number Nu is closely related to the Rayleigh number and correlations between the two exist for specific geometries and operating conditions. Nu characterizes the heat transfer across a boundary layer of the flow along a wall. Nu is the natural convection heat transfer normalized with the fluid's heat conduction.

$$Nu = \frac{h_{conv} l_c}{k} \tag{3.6}$$

where h_{conv} is the convective heat transfer coefficient. Locally, the convective heat transfer coefficient is

$$h_{conv} = \frac{\boldsymbol{q}\cdot\boldsymbol{n}}{\bar{T} - T_\infty} \tag{3.7}$$

where $\boldsymbol{q}\cdot\boldsymbol{n}$ is the heat flux achieved across a boundary, $\bar{T}$ is the (imposed) temperature of the boundary and T_∞ is the free stream temperature. Lastly, Ma is the ratio of heat convection due to the Marangoni effect and the diffusivity. It is defined as

$$Ma = \frac{\partial\gamma/\partial T\,\Delta T_c\,l_c}{\mu\,\alpha} = \frac{\rho\,c_p\,\partial\gamma/\partial T\,\Delta T_c\,l_c}{\mu\,k} \tag{3.8}$$

where $\partial\gamma/\partial T$ is the derivative of surface tension coefficient with respect to temperature, also known as the Marangoni coefficient, l_c and ΔT_c are again a characteristic length and temperature difference, but they are not necessarily the same as for other dimensionless numbers. The Marangoni effect appears at the surface, so these characteristic quantities need to be related to the surface as well.

3.2.1 Quasi-1D Natural convection

A fluid is located between two infinitely tall parallel plates that are close together. A temperature difference between both plates causes the density of the fluid to change locally, leading to natural convection. This test case verifies the correct implementation of the temperature dependent buoyancy, without a dependence on the fluid flow because the fluid flow occurs orthogonal to the temperature gradient.

In this example we approximate the fluid between the infinitely tall plates in 2D by a rectangular cavity of high aspect ratio $h/w = 40$ with a plane strain assumption. When the cavity is much taller than it is wide, as in this example, the problem becomes quasi-1D when sufficiently far from the top and bottom edge. Fig. 3.5 shows the cavity with a width $w = 0.1\,\text{m}$ and height $h = 4.0\,\text{m}$ and imposed temperature on the left and right walls, $\bar{T}_l = 300\,\text{K}, \bar{T}_r = 320\,\text{K}$ respectively and thus with a temperature difference $\Delta T = 20\,\text{K}$. The average temperature is used as the reference temperature $T_{ref} = 1/2(\bar{T}_l + \bar{T}_r)$. The gravity is set to $g = 10\,\text{m}\,\text{s}^{-2}$. The exact material properties of the fluid and dimensionless numbers using characteristic length $l_c = w$ are given in table 3.2

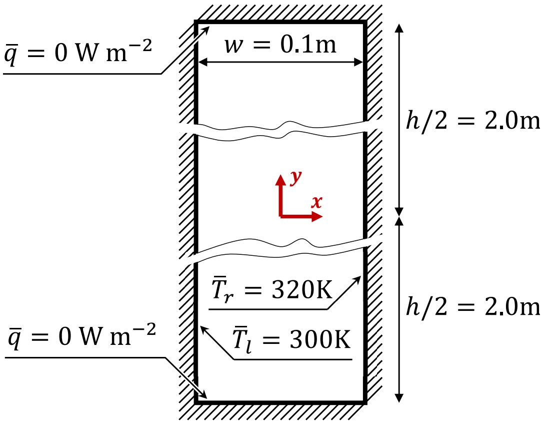

$\bar{q} = 0\ \text{W}\,\text{m}^{-2}$
$w = 0.1\text{m}$
$h/2 = 2.0\text{m}$
y
x
$\bar{T}_r = 320\text{K}$
$\bar{T}_l = 300\text{K}$
$h/2 = 2.0\text{m}$
$\bar{q} = 0\ \text{W}\,\text{m}^{-2}$

FIGURE 3.5: Schematic of the steady state 1D natural convection test case.

An analytical solution for the velocity profile of the natural convection between the infinitely tall walls can be derived for the steady state ($\partial v/\partial t = 0$). Assuming $v_x = 0$ and $\partial v_y/\partial y = 0$ due to the infinite height, we can use the conservation of momentum in the y-direction (the y element of the vector function eq. 2.14), which now reads

TABLE 3.2: Material properties.

Property	**Symbol**	**Value**	**Unit**
Viscosity	μ	5.0×10^{-3}	Pa s
Heat capacity	c_p	1.0	$\mathrm{J\,kg^{-1}\,K^{-1}}$
Conductivity	k	1.0	$\mathrm{W\,m^{-1}\,K^{-1}}$
Density	$\rho_{ref} = \rho(T_{ref})$	1.0×10^{2}	$\mathrm{kg\,m^{-3}}$
Thermal expansion coeff.	α_V	2.0×10^{-4}	$\mathrm{K^{-1}}$
Thermal diffusivity	α	0.01	$\mathrm{m^2\,s^{-1}}$
Prandtl number	Pr	5.0×10^{-3}	-
Rayleigh number	Ra	80.0	-

$$-\frac{\partial \sigma_{xy}}{\partial x} - \frac{\partial \sigma_{yy}}{\partial y} - \rho(T) b_y = 0 \tag{3.9}$$

and insert the constitutive equation for a Newtonian fluid, while keeping the fluid mechanics convention for the sign of the pressure in mind. With $b_y = -g$, this yields

$$-2\mu \frac{\partial \dot{\varepsilon}_{xy}}{\partial x} + \frac{\partial p}{\partial y} + \rho(T) g = 0 \tag{3.10}$$

Because the fluid is mechanically incompressible and the temperature does not vary in the y-direction ($\partial T/\partial y = 0$), the pressure term becomes

$$\frac{\partial p}{\partial y} = -g\rho_{ref} \tag{3.11}$$

For the shear term, we can write

$$\dot{\varepsilon}_{xy} = \frac{1}{2}\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right) = \frac{\partial v_y}{\partial x} \tag{3.12}$$

Due to the high thermal conductivity k and the lack of mass transfer in the x-direction ($v_x = 0$), the temperature profile becomes a linear distribution between the two imposed temperatures at the walls:

$$T(x) = \frac{\left(\bar{T}_l + \bar{T}_r\right)}{2} + \frac{\bar{T}_r - \bar{T}_l}{w} x = T_{ref} + \frac{\Delta T}{w} x \tag{3.13}$$

With the temperature distribution, we can derive a density distribution for this problem from eq 2.48

$$\rho(x) = \rho_{ref}\left(\alpha_V \left(T(x) - T_{ref}\right) + 1\right) \tag{3.14}$$

This density is then applied to the body force term. Inserting eqs. 3.11, 3.12, 3.13 and 3.14 into 3.10 and rearranging the terms, a differential equation is obtained:

$$\frac{d^2 v_y}{dx^2} = -\frac{\rho_{ref} g \alpha_V \Delta T}{\mu w} x \tag{3.15}$$

Expecting a cubic function for the v_y-distribution, we substitute

$$v_y = c_3 x^3 + c_2 x^2 + c_1 x + c_0 \tag{3.16}$$

$$\frac{d^2 v_y}{dx^2} = 6c_3 x + 2c_2 \tag{3.17}$$

With boundary conditions $v_y(x = -w/2) = v_y(x = w/2) = 0$ due to the no-slip condition at the walls, the y-velocity profile is found to be

$$v_y(x) = \frac{\rho_{ref} g \alpha_V \Delta T}{6\mu w} \left(-x^3 + \frac{w^2}{4} x \right) \tag{3.18}$$

The expected maximum and minimum velocities are included in tables 3.3 and 3.4.

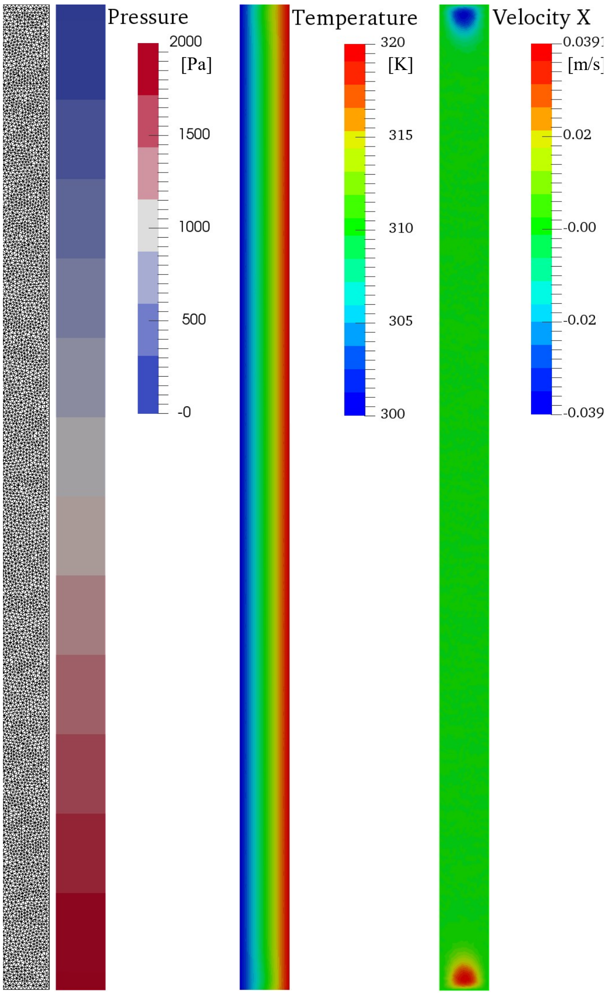


FIGURE 3.6: Initial mesh, pressure, temperature and x-velocity.

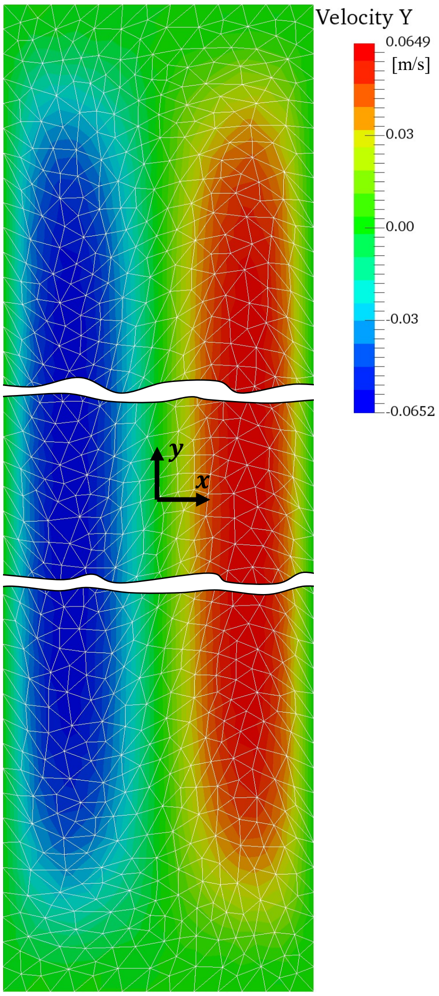


FIGURE 3.7: Detailed view of the y-velocity.

For the simulation with PFEM, a mesh of a mean edge length $l = 8.2 \times 10^{-3}\,\mathrm{m}$ and 8338 nodes is used. A time step $\Delta t = 0.1\,\mathrm{s}$ is used. For this choice of material properties, the steady state condition is reached well before the end time $t_{end} = 30.0\,\mathrm{s}$ or expressed in a non-dimensionalized way when $Fo = \alpha t_{end}/l_c^2 = 30.0$. The user can make several variations:

- use the Boussinesq approximation or not
- solve directly for the steady state or march in time until the steady state is reached
- use a free surface at the top or a fixed no-slip wall

In case of a free surface, the time step should be smaller at the beginning due to a possible build up of nodes at the free surface. We choose to use a constant $\Delta t = 0.025\,\mathrm{s}$ during the entire simulation, although a larger time step could be used after the first second, when the build-up has leveled out again. With a closed domain, it is possible to neglect the inertia and heat capacity terms and reach the steady state instantaneously. The Boussinesq approximation can be used to avoid modeling the actual thermal expansion and only take the effect on the body force into account (see section 2.3.3). We compare a free surface and a closed domain with a wall at the top. For the closed domain we run the direct steady state simulation and the time marching simulation. For each of these three setups we compare the results with the Boussinesq approximation and the regular thermal expansion. The temperature, pressure and velocity fields in figs. 3.6 and 3.7 are obtained. The results shown are from closed domain case with Boussinesq approximation, but visually, all six tests deliver the same contour plots.

The approximation of an infinitely tall pair of walls is best met in the midsection, far away from the top or bottom, where the fluid turns around. All following evaluations are therefore made at $y = 0$. A linear temperature distribution over x is reached due to the high thermal diffusivity acting over a long enough time. The y-velocity is positive near the hot wall on the right due to the fluid expanding, while sinking near the cold wall. Both walls have $v = 0$ due to the no-slip condition and the center ($x = 0$) has $v = 0$ due to the effects of hot and cold wall being balanced in the middle. The velocity profile over x at $y = 0$ is extracted and compared to the analytical solution in figs. 3.8 and 3.9.

The data is interpolated from the obtained nodal values onto the midsection ($y = 0$). No averaging over time is performed. The maximum and minimum velocities $\mathsf{max}(v_y)$ and $\mathsf{min}(v_y)$ and their respective x-position on the midsection $x_{\mathsf{max}(v_y)}$ and $x_{\mathsf{min}(v_y)}$ are given in tables 3.3 and 3.4.

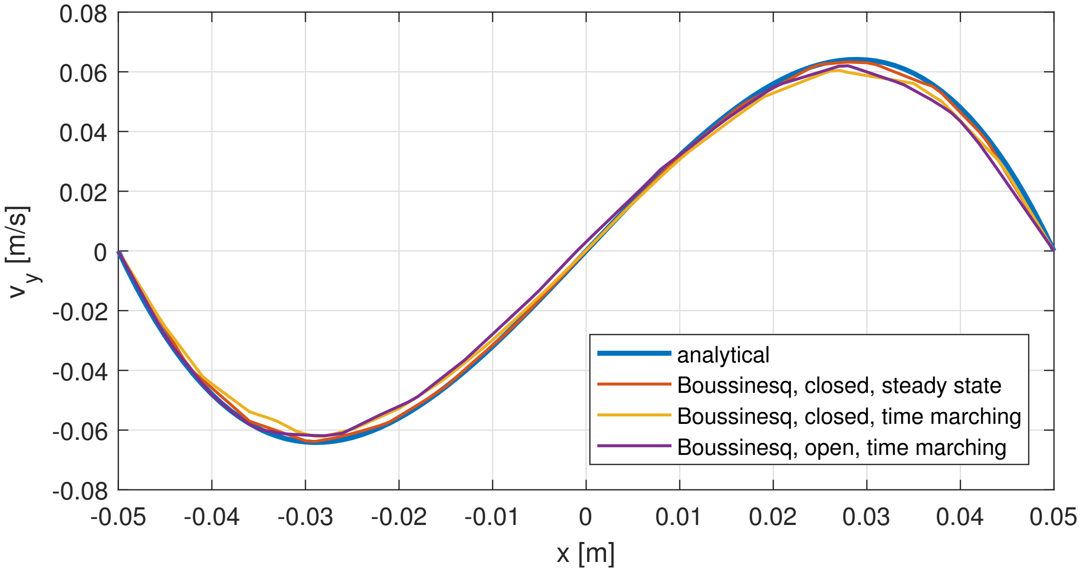


FIGURE 3.8: Comparison of y-velocity across the midsection ($y = 0$) of the 1D natural convection test case (with Boussinesq approximation).

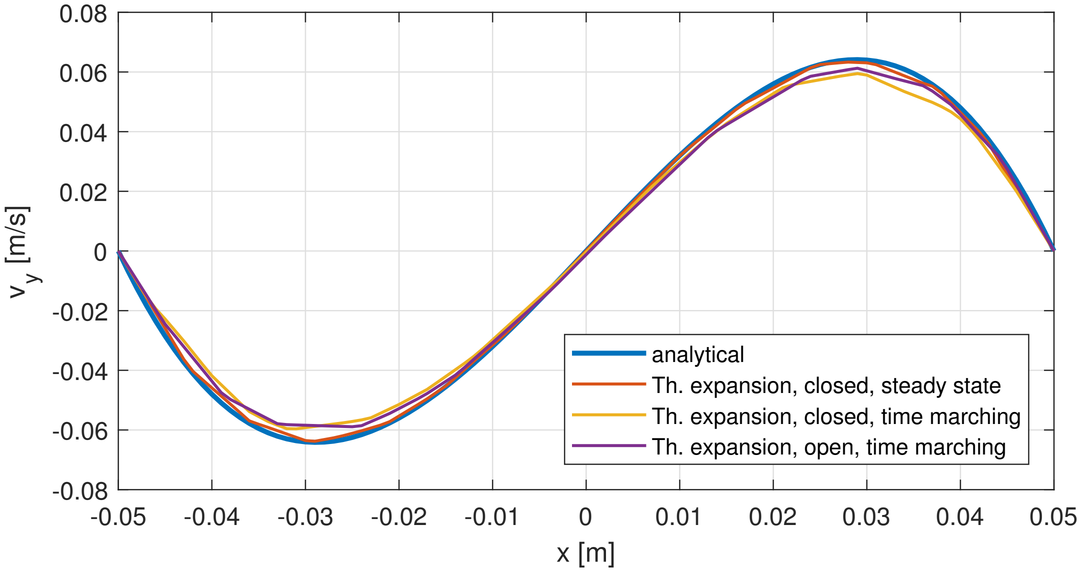


FIGURE 3.9: Comparison of y-velocity across the midsection ($y = 0$) of the 1D natural convection test case (with thermal expansion).

Overall, the expected physical behavior is well captured and the largest errors for the maximum and minimum y-velocities at $y = 0$ are 7.3% and 8.3%, respectively. The direct steady state solution is among the best, for both models for buoyancy, which might indicate that dynamic effects play a role. Further mesh refinement, smaller time steps, longer total simulation times or lower residual targets do not appear to improve the result.

Table 3.3: Maximum and minimum velocities of 1D natural convection problem (with Boussinesq approximation), relative error given in parentheses.

Quantity	analytical	Boussinesq, closed, steady st.	Boussinesq, closed	Boussinesq, open	Unit
$\mathsf{max}(v_y)$	0.0642	0.0633(1.4%)	0.0605(5.8%)	0.0620(3.4%)	$[\mathrm{m\,s^{-1}}]$
$\mathsf{min}(v_y)$	−0.0642	−0.0638(0.6%)	−0.0620(3.4%)	−0.0619(3.6%)	$[\mathrm{m\,s^{-1}}]$
$x_{\mathsf{max}(v_y)}$	0.0289	0.028(3.1%)	0.027(6.6%)	0.028(3.1%)	[m]
$x_{\mathsf{min}(v_y)}$	−0.0289	−0.029(0.3%)	−0.029(0.3%)	−0.028(3.1%)	[m]

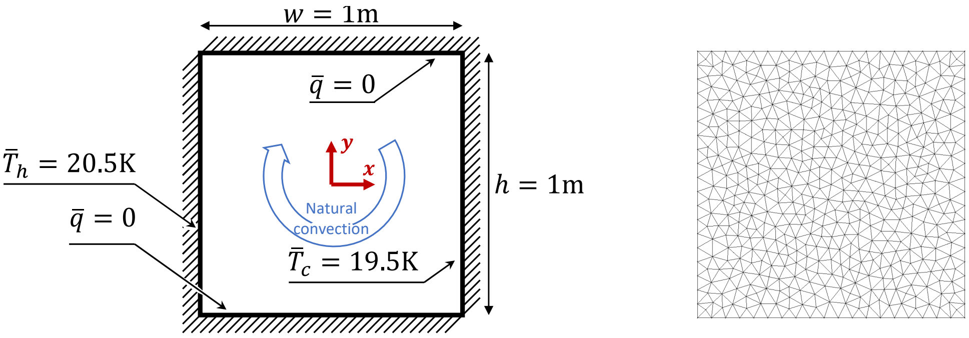
Table 3.4: Maximum and minimum velocities of 1D natural convection problem (with thermal expansion), relative error given in parentheses.

Quantity	analytical	Expanding, closed	Expanding, open	Unit
$\mathsf{max}(v_y)$	0.0642	0.0595(7.3%)	0.0613(4.5%)	$[\mathrm{m\,s^{-1}}]$
$\mathsf{min}(v_y)$	−0.0642	−0.0596(7.2%)	−0.0589(8.3%)	$[\mathrm{m\,s^{-1}}]$
$x_{\mathsf{max}(v_y)}$	0.0289	0.029(3.1%)	0.029(3.1%)	[m]
$x_{\mathsf{min}(v_y)}$	−0.0289	−0.031(7.3%)	−0.031(7.3%)	[m]

3.2.2 2D Natural convection

A temperature gradient is applied over a square cavity causing natural convection to occur. As opposed to the previous example, here the flow of the fluid influences the transport of heat, which makes this a non-linear problem.

The setup is according to Aubry et al. [93]. A square cavity of edge length $w = h = 1\,\mathrm{m}$, where all walls are no-slip walls, is filled with a viscous fluid initialized with a constant temperature $T_0 = 20.0\,\mathrm{K}$. The hot wall on the left has a constant temperature $\bar{T}_h = 20.5\,\mathrm{K}$, the cold wall on the right has a constant temperature of $\bar{T}_c = 19.5\,\mathrm{K}$ and all other walls are adiabatic $\bar{q} = 0$, as schematized in fig 3.10a. The gravity is set to $g = 1\,\mathrm{m\,s^{-2}}$. The material properties are given in table 3.5.



(a) Schematic.

(b) Mesh used: average element edge length $\bar{l}_e = 0.05\,\text{m}$, 515 nodes, 952 elements initially.

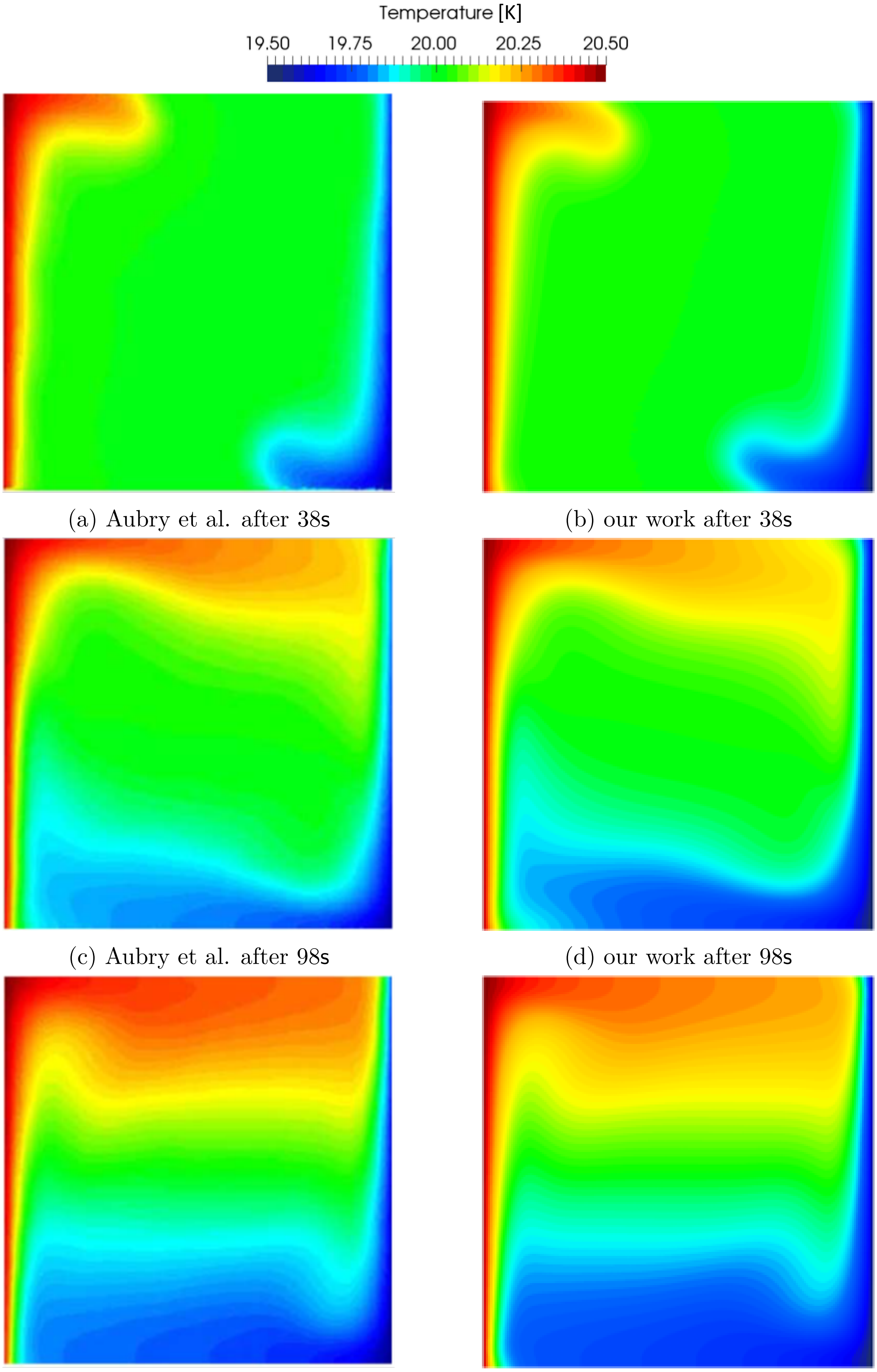
FIGURE 3.10: Setup of the 2D natural convection problem.

TABLE 3.5: Material properties.

Property	**Symbol**	**Value**	**Unit**
Viscosity	μ	1.0×10^{-4}	Pa s
Heat capacity	c_p	1.0	$\text{J kg}^{-1}\,\text{K}^{-1}$
Conductivity	k	1.0×10^{-4}	$\text{W m}^{-1}\,\text{K}^{-1}$
Density	ρ	1.0	kg m^{-3}
Thermal expansion coeff.	α_V	1.0×10^{-2}	K^{-1}
Thermal diffusivity	α	1.0×10^{-4}	$\text{m}^2\,\text{s}^{-1}$
Prandtl number	Pr	1.0	-
Rayleigh number	Ra	1.0×10^{6}	-

The process is simulated until a total time $t_{end} = 398.0\,\text{s}$ with a constant time step $\Delta t = 0.5\,\text{s}$. The mesh is irregular and uniform, with an average element edge length $\bar{l}_e = 0.05\,\text{m}$, which results in 952 triangles and 515 nodes (fig. 3.10b).

The results are compared with Aubry et al. [93], who use their own PFEM implementation. The authors only provide the geometry, a Pr and a Ra, which is sufficient to describe the final steady state result of the test case. The time scale at which the flow evolves before reaching steady state has not been defined, but through trial-and-error the α in table 3.5 was found to reproduce the transient evolution of the flow and temperature as depicted in [93]. The agreement then is excellent, despite the coarse mesh.



(a) Aubry et al. after 38s

(b) our work after 38s

(c) Aubry et al. after 98s

(d) our work after 98s

(e) Aubry et al. after 398s

(f) our work after 398s

FIGURE 3.11: Temperature contours over time.

While Aubry et al. [93] provide the temperature contour plots of the transient problem, a more recent publication by Marti and Ryzhakov [40], also using PFEM, provides complementing quantified results. They present more detailed velocity data at different values for Ra, but only for the steady state result. Furthermore, they compare to previously published results from the literature that use methods other than PFEM: de Vahl Davis [99] uses the Finite Difference Method (FDM) and Corzo et al. [100] use the Finite Volume Method (FVM), both based on an Eulerian description. Sklar et al. [56] use the PFEM-2, a method that, despite the name, is very different from the classic PFEM (see section 1.4.1). In [40] the material properties are clearly described for the three test cases that are compared. The Rayleigh number is changed, while everything else remains the same. In this work, we only show the highest and lowest, as given in table 3.6.

TABLE 3.6: Material properties.

Property	**Symbol**	**Value**	**Unit**
Viscosity	μ	1.0×10^{-3}	$\mathrm{Pa\,s}$
Heat capacity	c_p	1.0	$\mathrm{J\,kg^{-1}\,K^{-1}}$
Conductivity	k	1.0×10^{-3}	$\mathrm{W\,m^{-1}\,K^{-1}}$
Density	ρ	1.0	$\mathrm{kg\,m^{-3}}$
Thermal expansion coeff.	α_V	$1.0 \times 10^{-1}/1.0 \times 10^{-3}$	$\mathrm{K^{-1}}$
Thermal diffusivity	α	1.0×10^{-3}	$\mathrm{m^2\,s^{-1}}$
Prandtl number	Pr	1.0	-
Rayleigh number	Ra	$1.0 \times 10^4/1.0 \times 10^6$	-

The gravity is set to $g = 10\,\mathrm{m\,s^{-2}}$ and the temperature difference is $\Delta t = 1\,\mathrm{K}$. In the square of edge length $w = h = 1.0\,\mathrm{m}$ and the origin in the center, horizontal and vertical center lines are introduced that coincide with the x and y-axes. On the horizontal center line the maximum vertical velocity $\mathsf{max}(v_y)$ and the x-position at which it occurs $x_{\mathsf{max}(v_y)}$ are compared. On the vertical center line the maximum horizontal velocity $\mathsf{max}(v_x)$ and the respective y-position $y_{\mathsf{max}(v_x)}$ are recorded. This yields four values per test case that describe the velocity field well in a simple manner. It is implicitly assumed that the velocity fields are sufficiently symmetric, since only the positive maximum velocities are compared. In an ideal case, the minimum velocities should be negative, but of equal magnitude as their maximum counterpart and their position should be the same distance from the origin, but in the opposite direction.

For an easier comparison, a non-dimensionalization proposed by [100] is used. All distances become the non-dimensional distance $\hat{x}_i = x_i/w$, the time becomes a non-dimensional time $\hat{t} = tw^2/\alpha$ and velocities become a non-dimensional velocity $\hat{v}_i = v_i w/\alpha$. The results are given in tables 3.7 and 3.8. A uniform mesh with an average element edge length $\bar{l}_e = 0.05\,\mathrm{m}$ is used (same as in previous test, fig. 3.10b).

TABLE 3.7: Maximum velocities and position of maxima at $Ra = 10^6$.

Data ($Ra = 10^6$)	**This work**	**Marti & Ryzhakov**	**de Vahl Davis**	**Corzo et al.**	**Sklar et al.**
$\mathsf{max}(\hat{v}_x)$	65.115	65.45	65.81	64.558	64.483
$\hat{y}_{\mathsf{max}(v_x)}$	0.854	0.850	0.8520	0.851	0.845
$\mathsf{max}(\hat{v}_y)$	225.913	213.070	214.640	221.572	218.054
$\hat{x}_{\mathsf{max}(v_y)}$	0.0404	0.0474	0.0396	0.0670	0.0370

TABLE 3.8: Maximum velocities and position of maxima at $Ra = 10^4$.

Data ($Ra = 10^4$)	**This work**	**Marti & Ryzhakov**	**de Vahl Davis**	**Corzo et al.**	**Sklar et al.**
$\mathsf{max}(\hat{v}_x)$	16.154	16.250	16.182	16.282	15.982
$\hat{y}_{\mathsf{max}(v_x)}$	0.820	0.821	0.823	0.822	0.824
$\mathsf{max}(\hat{v}_y)$	20.032	19.541	19.509	19.547	19.378
$\hat{x}_{\mathsf{max}(v_y)}$	0.120	0.115	0.120	0.123	0.116

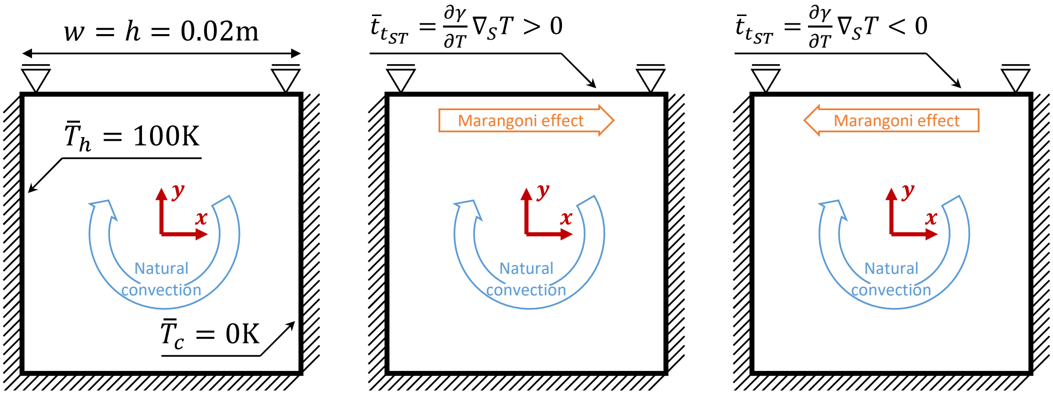
A good agreement with the literature is achieved, where the maximum relative difference of the non-dimensional velocities is 5.7% for the comparison of $\mathsf{max}(\hat{v}_y)$ with Marti and Ryzhakov [40] at $Ra = 10^6$.

3.2.3 Natural convection with Marangoni effect

A natural convection test case, as the one described in section 3.2.2 is repeated, but with the tangential component of the surface tension (the Marangoni effect) applied to the top boundary.

The test is taken from Saldi's doctoral thesis [12] who uses an Eulerian Finite Volume Method (FVM) to investigate numerous tests involving the Marangoni effect. In his work, comparisons are made with earlier simulations by Bergman and Keller [101], who also employ an Eulerian FVM approach that directly solves for the steady state solution. The material properties of molten aluminium are listed in table 3.9. In a square of $w = h = 0.02\,\mathrm{m}$, a temperature difference $\Delta T = \bar{T}_h - \bar{T}_c = 100\,\mathrm{K}$ is imposed between the side walls. A free-slip boundary condition is used for the top boundary, as the Eulerian methods in the literature do not feature a free surface deformation.

Three cases are investigated that each impose a different surface traction caused by the Marangoni effect. The three different Marangoni coefficients are listed in table 3.9. Figure 3.12 illustrates the effect that the coefficient has in each case. Most notably, the Marangoni effect can either reinforce or oppose the circulation caused by pure natural convection.


$w = h = 0.02$m
$\bar{T}_h = 100$K
$\bar{T}_c = 0$K
y
x
Natural convection
$\bar{t}_{tST} = \frac{\partial\gamma}{\partial T}\nabla_S T > 0$
Marangoni effect
$\bar{t}_{tST} = \frac{\partial\gamma}{\partial T}\nabla_S T < 0$
Marangoni effect

(a) Case 1: Natural convection only.

(b) Case 2: With a negative Marangoni coefficient leading to a surface traction in positive x-direction.

(c) Case 3: With a positive Marangoni coefficient leading to a surface traction in negative x-direction.

FIGURE 3.12: Schematic of the 2D natural convection test case with the Marangoni effect. Depending on the sign of the Marangoni coefficient $\partial\gamma/\partial T$, the Marangoni convection coincides with (b) or opposes (c) the natural convection.

TABLE 3.9: Material properties (with three different Marangoni coefficients).

Property	**Symbol**	**Value**	**Unit**
Viscosity	μ	1.3×10^{-3}	Pa s
Heat capacity	c_p	1080.0	$\mathrm{J\,kg^{-1}\,K^{-1}}$
Conductivity	k	94.03	$\mathrm{W\,m^{-1}\,K^{-1}}$
Density	ρ	2385.0	$\mathrm{kg\,m^{-3}}$
Thermal exp. coeff.	α_V	1.17×10^{-4}	$\mathrm{K^{-1}}$
Marangoni coeff.	$\partial\gamma/\partial T$	$0/-3.5 \times 10^{-4}/2.0 \times 10^{-4}$	$\mathrm{N\,m^{-1}\,K^{-1}}$
Thermal diffusivity	α	3.65×10^{-5}	$\mathrm{m^2\,s^{-1}}$
Prandtl number	Pr	0.0149	-
Rayleigh number	Ra	4.6×10^4	-
Marangoni number	Ma	$0/-1.47 \times 10^4/8.4 \times 10^3$	-

A total simulation time of $t_{end} = 10$ s with a constant time step $\Delta t = 1 \times 10^{-4}$ s is set. The mesh is irregular and uniform, with 40 elements per edge, which results in a number of triangle elements $n_e = 4024$ and a number of nodes $n_n = 2091$.

Streamline plots

Streamlines are an intuitive visualization of the flow. They represent the flow direction a massless particle would follow at a given point in the flow field at the current moment. For a steady state flow, these streamlines are closed loops and a particle will arrive back at its starting position. In this test case, streamlines show the circulation caused by natural convection and the Marangoni effect.

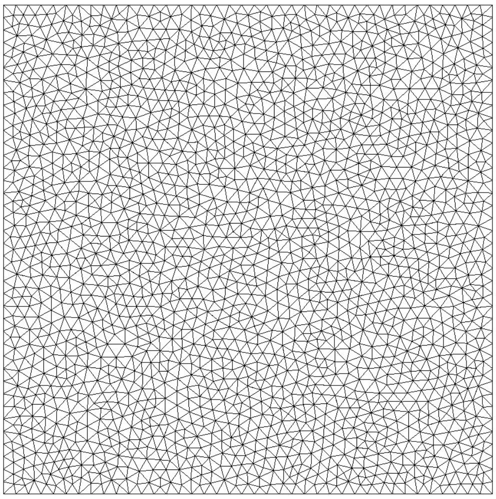
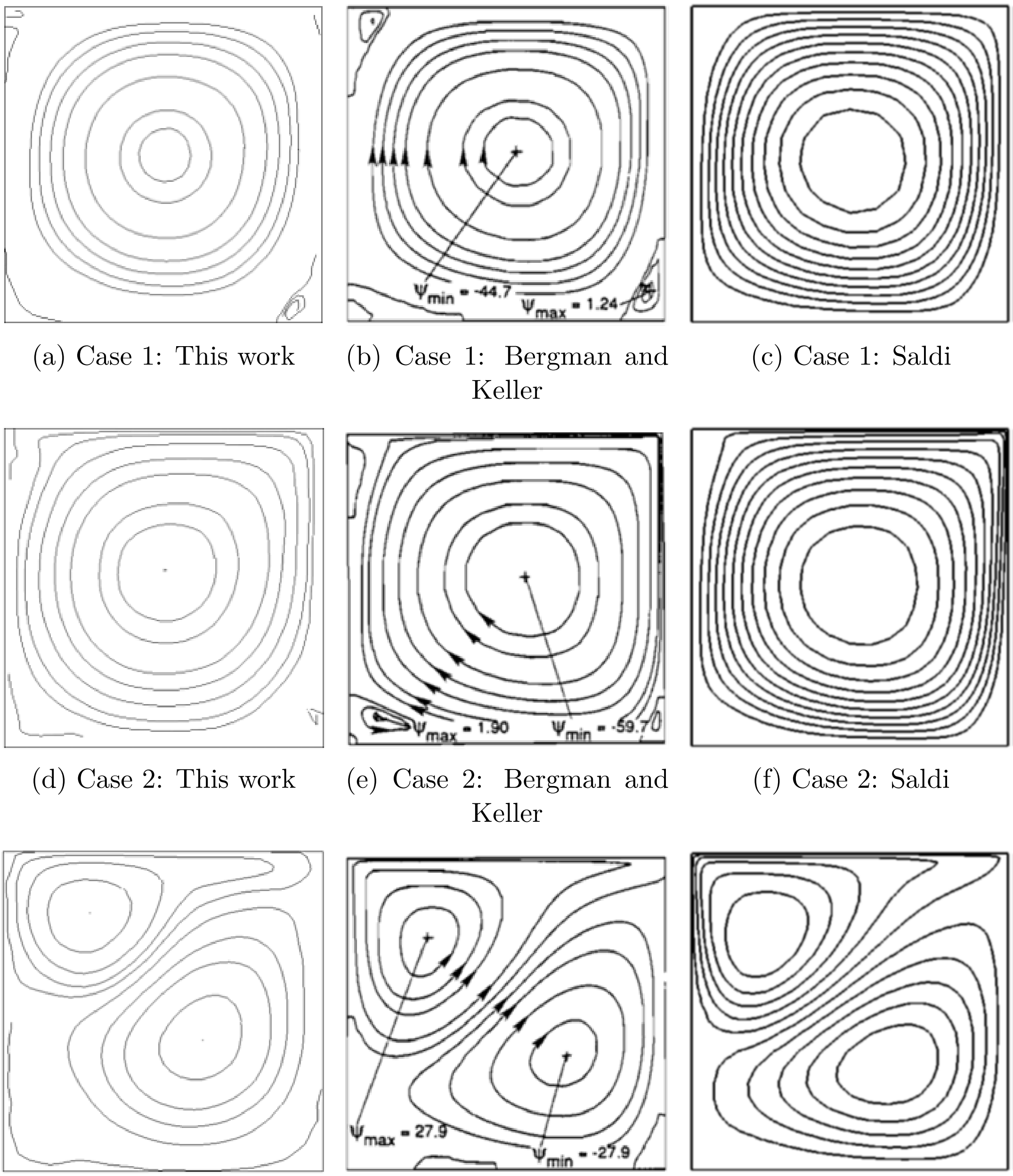


FIGURE 3.13: Mesh used: average element edge length $\bar{l}_e = 0.5\,\mathrm{mm}$, 2091 nodes, 4024 elements initially.

Fig. 3.14 compares the streamlines for the three cases between this work and the literature. A good agreement is observed in all cases. The differences are the largest in case 3, where all three authors find slightly different vortex size and location. This is not surprising, since in this case a delicate balance between the two driving forces is maintained, which leads to the appearance of the two main vortices. Small recirculation bubbles are found in the corners for all three cases, but are not well captured. This is due to their small size and unsteady nature, which requires a fine spatial resolution, a long sampling time and time averaging. We were not able to obtain consistent results regarding the recirculation bubbles, which is a flaw that we can accept, given the good agreement of the larger flow features.

Isotherm contour plots

Isotherms reveal the temperature levels and indirectly the temperature gradients. The agreement in case 1 is found to be very good among all works compared. For cases 2 and 3, noticeable differences between all 3 works are found. The results obtained in this work appear to deviate slightly from the literature. A possible but unlikely explanation is the issue with the recirculation bubbles described above. It is possible that our method is less diffusive, which can be viewed as beneficial. Bergman and Keller use a very coarse mesh and an upwind differencing scheme (UDS), which both add numerical diffusion. There is however, no reason to assume that Saldi's method is particularly diffusive. Experimental results could best confirm this hypothesis, but none exist in the literature to be best knowledge of the author. It must also be noted that small differences in temperature can dramatically change the distribution of the isotherms, which may not be easily visible in a contour plot. In other words, this visualization method is very sensitive and clearly shows even small discrepancies.



(a) Case 1: This work

(b) Case 1: Bergman and Keller

(c) Case 1: Saldi

(d) Case 2: This work

(e) Case 2: Bergman and Keller

(f) Case 2: Saldi

(g) Case 3: This work

(h) Case 3: Bergman and Keller

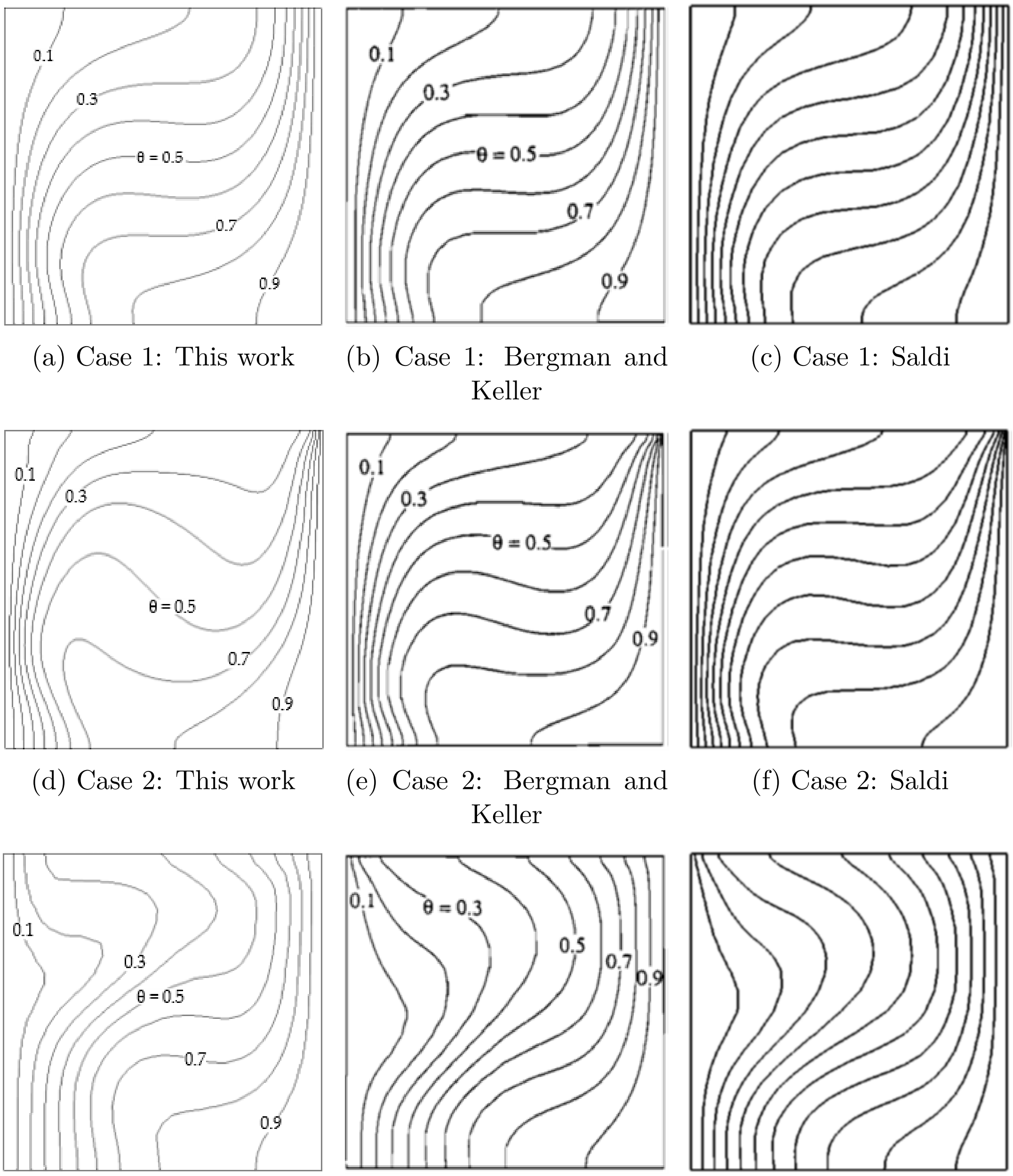
(i) Case 3: Saldi

FIGURE 3.14: Comparison of streamlines. This work (left), Bergman and Keller [101] (middle), Saldi [12] (right). Cases 1, 2 and 3 from top to bottom.

Local Nusselt number plots

The Nusselt number Nu can be used to quantify the heat transfer by a flow passing a wall. Using the definition of Nu in eq. 3.6, together with eq. 3.7. Calculating the local $Nu = Nu(y)$ and plotting it against y, reveals where the heat is transferred most effectively from the hot wall to the fluid. Fig. 3.16 compares this dimensionless heat transfer profile of the 3 cases with the literature results. The equivalent graph for the cold wall is found in fig. 3.17.

A good agreement of all three cases with the literature is found. The most



(a) Case 1: This work

(b) Case 1: Bergman and Keller

(c) Case 1: Saldi

(d) Case 2: This work

(e) Case 2: Bergman and Keller

(f) Case 2: Saldi

(g) Case 3: This work

(h) Case 3: Bergman and Keller

(i) Case 3: Saldi

Figure 3.15: Comparison of isotherms. This work (left), Bergman and Keller [101] (middle), Saldi [12] (right). Cases 1, 2 and 3 from top to bottom. Following Bergman and Keller, a non-dimensionalized temperature $\theta = (\bar{T}_h - T)/\Delta T$ is used.

obvious deviation of this work from the examples in the literature is for case 2 at the hot wall, where the local maximum of Nu is shifted up and slightly smaller in magnitude. This is clearly an effect of the slightly different temperature distributions highlighted by the isotherms. Comparing fig. 3.15d with figs. 3.15e, f reveals that the flow of cold fluid impinges on the hot wall closer to the center, which is then reflected in the shifted peak of the non-dimensional heat transfer in fig. 3.16.

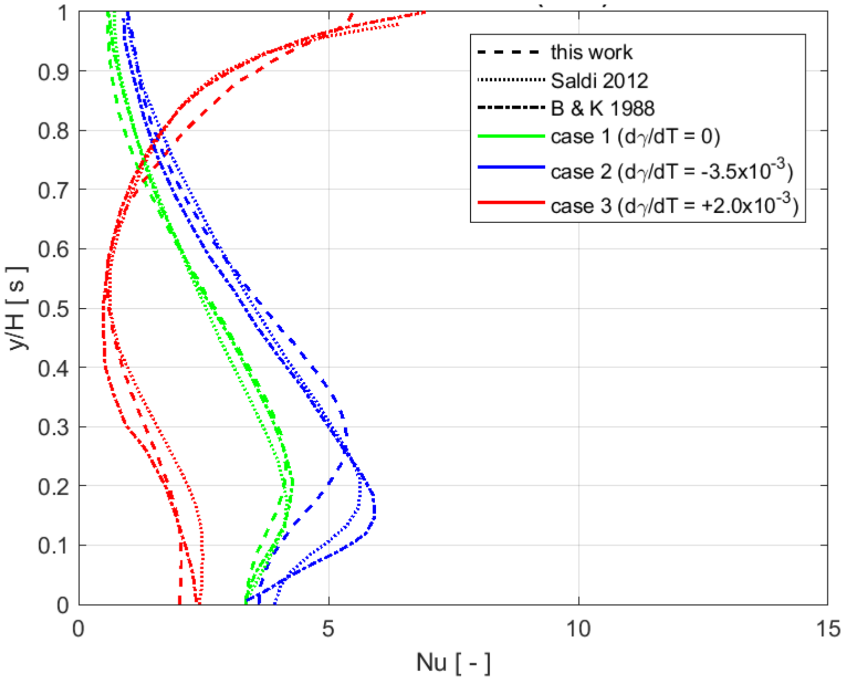


FIGURE 3.16: Nusselt number on the hot wall. Comparison with Bergman and Keller [101] and Saldi [12].

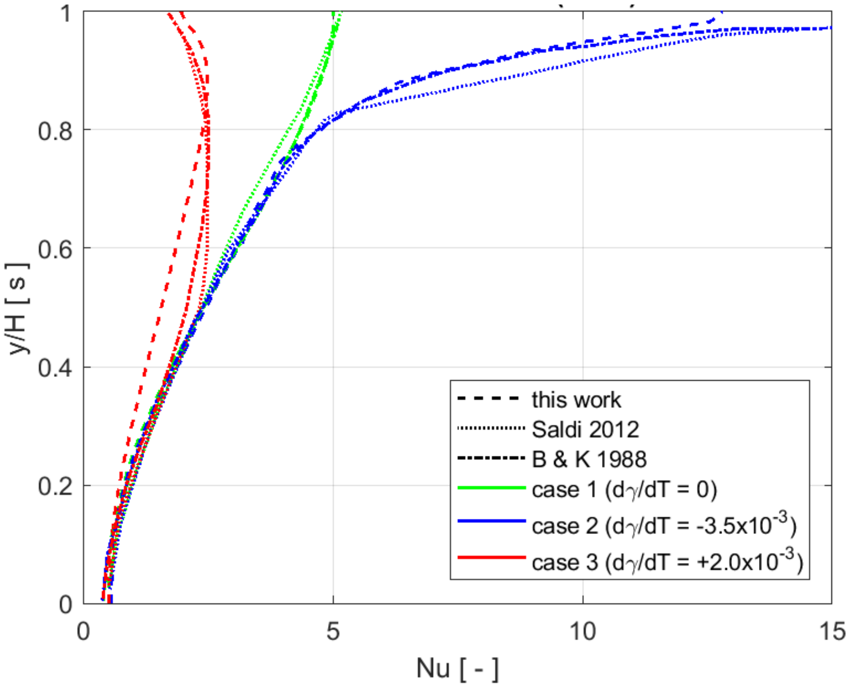
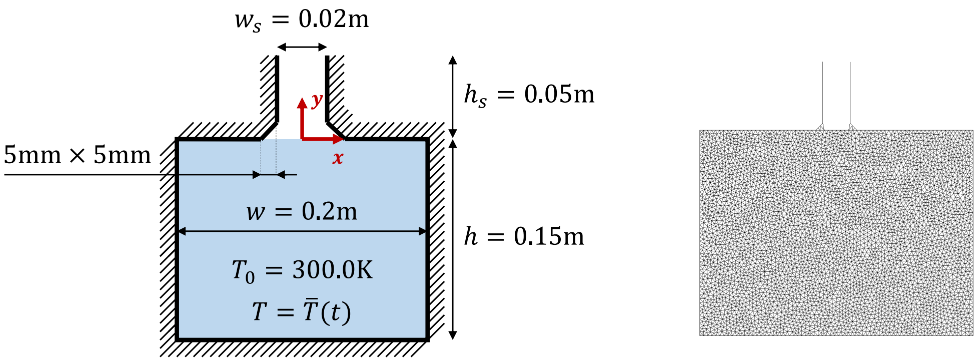


FIGURE 3.17: Nusselt number on the cold wall. Comparison with Bergman and Keller [101] and Saldi [12].

3.2.4 Fluid displacement by thermal expansion

A closed container with an opening at the top is filled with a fluid. The temperature of the fluid is imposed throughout the domain (no solving of the heat equation) and increases linearly over time. The fluid expands and pushes through the opening. This test is supposed to demonstrate mass conservation during thermal expansion with a free surface. The fluid is assumed to be mechanically incompressible ($\frac{\partial\rho}{\partial p} = 0$), while volume and density change due the change in temperature ($\frac{\partial\rho}{\partial T} \neq 0$). A schematic of the geometry is given in fig. 3.18a and the material properties are given in table 3.10. The free surface is exposed to a pressure of $\bar{p} = 0\,\mathrm{Pa}$, all other boundaries are no-slip walls. The temperature is imposed on the entire domain to rise from $T_0 = 300\,\mathrm{K}$ to $T(t = t_{end}) = 500\,\mathrm{K}$, where $t_{end} = 0.3\,\mathrm{s}$ is the total simulation time. A time step $\Delta t = 0.001\,\mathrm{s}$ and a mean edge length $\bar{l}_e = 0.002\,\mathrm{m}$ of the mesh elements is found to be sufficient (fig. 3.18b).



(a) Schematic of the thermal expansion in an open container problem.

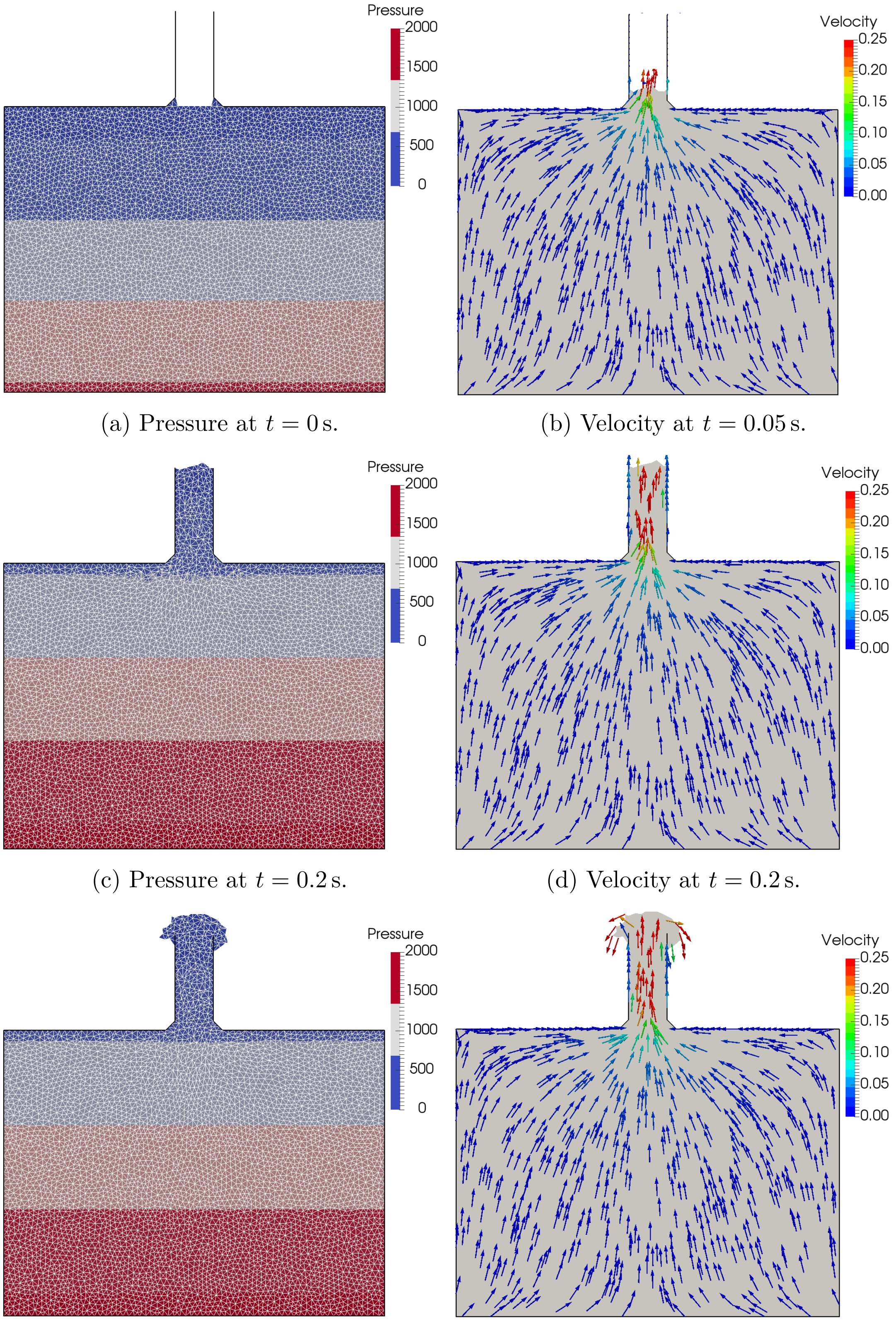
(b) Mesh used: average element edge length $\bar{l}_e = 0.002\,\mathrm{m}$, 2666 nodes, 5302 elements initially.

FIGURE 3.18: Setup of the thermal expansion in an open container problem.

TABLE 3.10: Material properties.

Property	Symbol	Value	Unit
Viscosity	μ	5.0×10^{-3}	Pa s
Density at T_0	ρ	1.0×10^{3}	kg m^{-3}
Thermal expansion coeff.	α_V	2.0×10^{-4}	K^{-1}

Fig. 3.19 shows the time evolution of the problem. As the fluid's temperature increases at a steady rate, the density decreases at the same time as the volume increases and the product of the two, the mass, should remain constant. The differential equation that relates density to temperature, eq. 2.37, can be solved by assuming an exponential function for ρ and integrating over the temperature interval. The following function for ρ is obtained:



(a) Pressure at $t = 0\,\mathrm{s}$.

(b) Velocity at $t = 0.05\,\mathrm{s}$.

(c) Pressure at $t = 0.2\,\mathrm{s}$.

(d) Velocity at $t = 0.2\,\mathrm{s}$.

(e) Pressure at $t = 0.3\,\mathrm{s}$.

(f) Velocity at $t = 0.3\,\mathrm{s}$.

FIGURE 3.19: Snap shots of pressure (left) and velocity (right) over time. The fluid begins overflowing at $t \approx 0.2\,\mathrm{s}$.

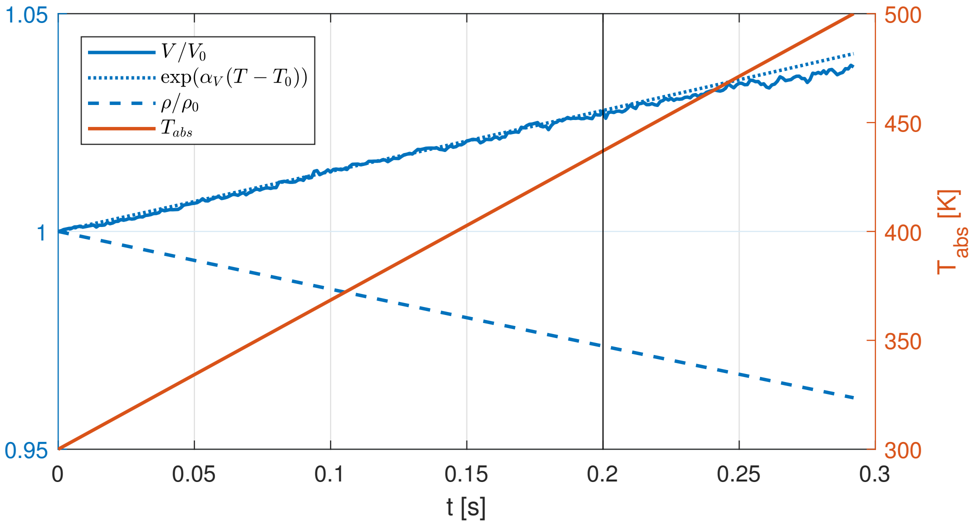


FIGURE 3.20: Time evolution of volume and density. Overflow begins at the thin black line.

$$\frac{\rho(T)}{\rho_0} = \exp(-\alpha_V(T - T_0)) \tag{3.19}$$

where ρ_0 is the reference density at reference temperature T_{ref}. Since the specific volume $v = \rho^{-1}$ and the reference specific volume $v_0 = \rho_0^{-1}$, eq. 3.19 can easily be transformed into an equation describing the evolution of the specific volume. In this particular example, where the material and the temperature field are uniform and therefore also v, we can even write eq. 3.19 for the volume of the fluid V:

$$\frac{V(T)}{V_0} = \exp(\alpha_V(T - T_0)) \tag{3.20}$$

where V_0 is the volume of the fluid at reference condition. Naturally, the reference condition here is the initial condition, where ρ_0, T_0 and V_0 are known. Given the imposed increase in temperature we can plot the evolution of these quantities over time, as in fig. 3.20.

With the temperature (red line) increasing over time, the normalized density (blue dashed line) must decrease (eq. 3.19). If the mass were perfectly conserved, the normalized volume (blue solid line) would need to follow exactly eq. 3.20 (blue dotted line). But since the volume in PFEM is also affected by the free surface, small fluctuations are observed due to elements appearing and disappearing at the free surface. Once the free surface reaches the top of the opening at around $t = 0.2\,\mathrm{s}$ (compare fig. 3.19c,d), marked by a thin vertical black line, the overflow appears to consistently remove mass at a low rate. This is likely due to the free surface elements being stretched, which is a known mechanism in PFEM that can cause a volume deficiency (see e.g. Falla et al. [96]).

The mass m (thick black line in fig. 3.21) is nearly constant (±0.1% error),

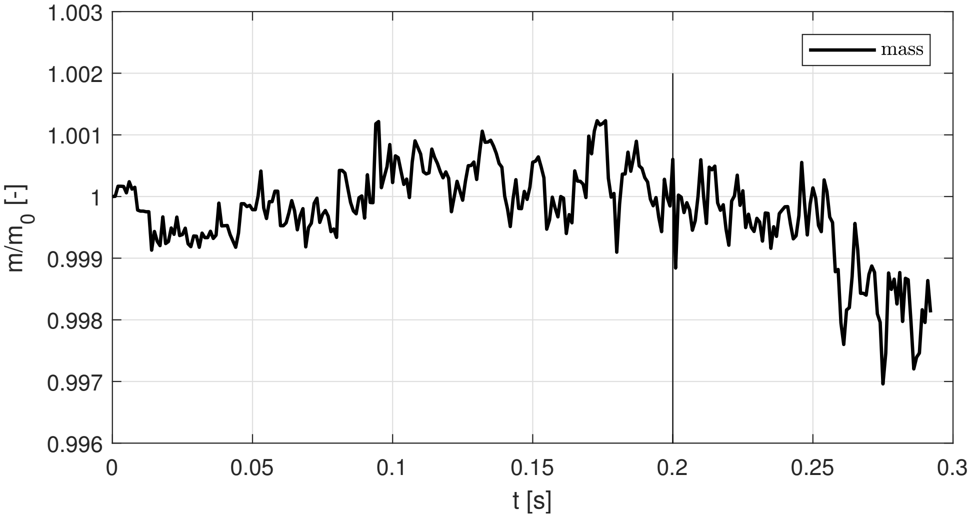


FIGURE 3.21: Time evolution of mass.

until the overflow begins (thin black line), where mass is lost, as volume is lost. Nevertheless, the expanding fluid in the reservoir causes a flow through the opening and a realistic behavior is captured with minimal errors until the overflow.

3.3 Thermo-fluid with phase change

Assuming that the thermo-fluid with thermal expansion, free surface deformation, surface tension and Marangoni effect are now sufficiently validated, the next step is to introduce phase change. Two academic tests and one slightly more complex test are considered.

3.3.1 1D Stefan Problem - non-isothermal

The first step is a static thermal test case with non-isothermal phase change in a quasi-1D setup. This type of test case, referred to as Stefan problem, features a phase transition front that moves across the initially liquid domain, whose initial temperature $T_0 > T_{liq}$, where T_{liq} is the liquidus temperature. The phase transition is driven by a cold wall with an imposed temperature $\bar{T}_c < T_{sol}$, where T_{sol} is the solidus temperature. The domain must be long enough such that no significant temperature increase occurs far from the wall with the imposed temperature, making the domain quasi-infinite in the direction facing away from that wall.

The specific test setup here follows that of Celentano et al. [65], who use a 1D FEM approach with a regular mesh of 32 linear line elements. The phase change is implemented with a temperature-based heat equation (as in this work, see section 2.2.3) and a regularization of the discontinuity at the phase front (as in this work, see section 2.3.1) in case of isothermal phase change.

They also compare their results with Rolph and Bathe [102], who also use a 1D FEM approach with a mesh of 32 equally spaced line elements. They employ an enthalpy formulation for the energy conservation equation where the latent heat appears as a source term. To the best knowledge of the author, no analytical solution exists for a non-isothermal ($T_{sol} \neq T_{liq}$) Stefan problem.

As illustrated in fig. 3.22, a domain of length $l = 4\,\mathrm{m}$ and an arbitrary height h (we choose $h = 0.25\,\mathrm{m}$) has its cold wall with the imposed temperature $\bar{T}_c = 228.15\,\mathrm{K}$ on the left. The initial temperature is $T_0 = T_{liq} + 0.1\,\mathrm{K} = 273.15\,\mathrm{K}$. The material properties are given in table 3.11.

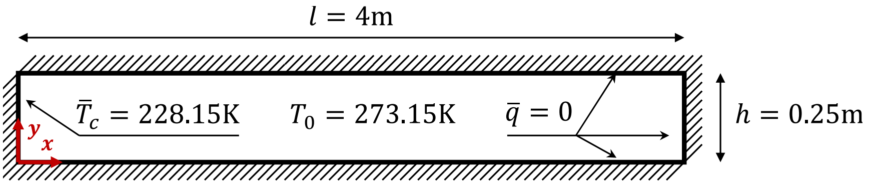


FIGURE 3.22: Schematic of quasi-1D Stefan problem.

TABLE 3.11: Material properties.

Property	Symbol	Value	Unit
Density	ρ	1.0	$\mathrm{kg\,m^{-3}}$
Heat capacity	c_p	1.0	$\mathrm{J\,kg^{-1}\,K}$
Conductivity	k	1.08	$\mathrm{W\,m^{-1}\,K}$
Latent heat	L_m	70.26	$\mathrm{J\,kg^{-1}}$
Liquidus temp.	T_{liq}	273.05	K
Solidus temp.	T_{sol}	263.05	K

Following the references in the literature [65, 102], the simulation is carried out on a regular mesh with 33 nodes along the length and 3 across the height, leading to a number of elements $n_e = 128$. The simulation time is $t_{end} = 4\,\mathrm{s}$ with a time step $\Delta t = 0.2\,\mathrm{s}$, in accordance with the literature. The heat equation is solved using the Newton-Raphson algorithm, but no line search is necessary. No regularization is used either. The temperature at $x = 1\,\mathrm{m}$ is recorded over time and compared with the literature results.

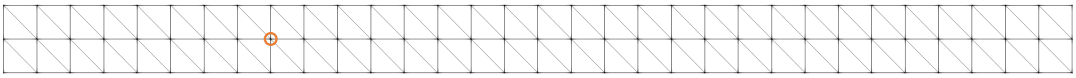


FIGURE 3.23: Mesh with 33x3 nodes and 128 elements. The temperature is extracted at the marked node at $x = 1\,\mathrm{m}$ and $y = 0.125\,\mathrm{m}$.

As shown in fig. 3.24, in the first half-second the temperature at $x = 1\,\mathrm{m}$ decreases slowly, as the phase change front is supplying heat to the cold wall. Only after the $x = 1\,\mathrm{m}$ mark is well inside the transition interval, the temperature can drop faster as heat flows out of the domain through the cold wall.

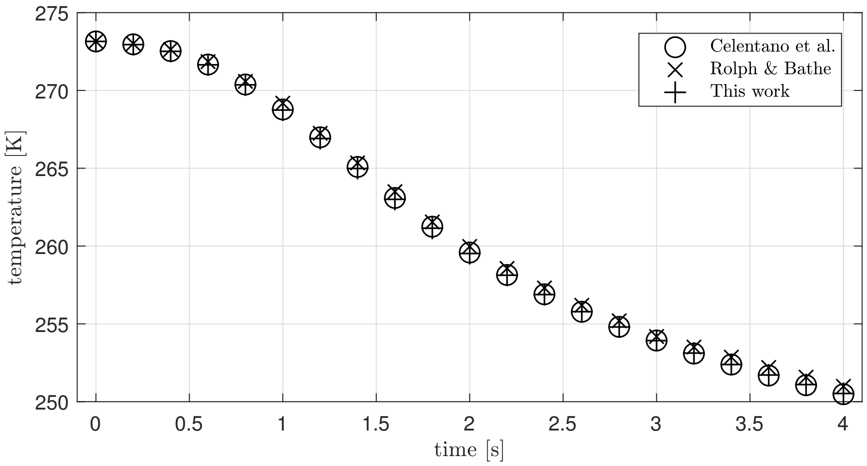


FIGURE 3.24: Temperature evolution at $x = 1\,\mathrm{m}$. Comparing with Celentano et al. [65] and Rolph and Bathe [102].

All methods reproduce this behavior well, despite the coarse meshes used in all methods. The PFEM results match more closely with Celentano et al., which is likely due to the methods being more similar to one another than to that of Rolph and Bathe.

3.3.2 1D Stefan Problem - isothermal

The previous test is repeated with isothermal phase change. Comparisons are made with the work of Celentano et al. [65], Rolph and Bathe [102], Morgan et al. [103] and Budhia and Kreith [104]. The methods of the first two publications are described in section 3.3.1. Morgan et al. [103] use a 2D FEM approach with the so-called enthalpy method that introduces a modified heat capacity that takes the latent heat into account. A regularization is also used in [103] to better handle the discontinuity. A more detailed summary of the individual methods can be found in Celentano et al. [65]. In their publication, an analytical solution to the isothermal Stefan problem is presented based on the work of Budhia and Kreith [104].

The material properties of the previous test case are kept, except for the melting interval of solidus and liquidus temperatures (T_{sol}, T_{liq}) being replaced by a sharp melting point $T_m = 272.15\,\mathrm{K}$. The mesh and the time step are kept the same. The major difference is the use of additional methods that help converge despite the discontinuous relation between temperature and internal energy:

- A regularization with $\epsilon_{reg} = 0.05\,\mathrm{K}$ at the step of the liquid fraction function (see section 2.3.1).
- The Newton-Raphson method is enhanced with a line search algorithm (see section 2.6.2), which has been found to be an absolute necessity for such coarse meshes (less critical for fine meshes that better resolve the transition).

- The integration of the latent heat term uses 6 Gauss points instead of 1 in elements transitioning to better capture the discontinuity inside each element (see section 2.6.3).

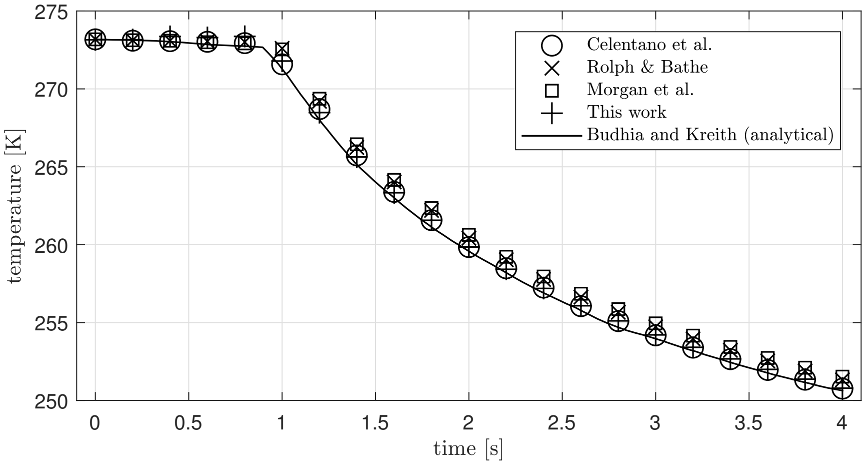


FIGURE 3.25: Temperature evolution at $x = 1\,\mathrm{m}$. Comparing with Celentano et al. [65] and Rolph and Bathe [102], Morgan et al. [103], and an analytical solution by Budhia and Kreith [104].

As shown in fig. 3.25, the behavior at $x = 1\,\mathrm{m}$ described in the previous section is found again here, but with a much sharper transition between the plateau and the decline of the temperature. The phase front appears to pass the $x = 1\,\mathrm{m}$ mark after $t = 0.89\,\mathrm{s}$, which is confirmed in a separate simulation with higher temporal and spatial resolution (not shown). Once again, the agreement of PFEM results is best with those of Celentano et al. [65], but other results in the literature, including the analytical solution, agree well with one another.

3.3.3 Gallium melting with natural convection

The next step is a test with a flowing fluid and phase change, where the solid is modeled by the Carman-Kozeny equation, eq. 2.49 ($C_{CK} = 1 \times 10^6\,\mathrm{kg\,m^{-3}\,s^{-1}}$ and $\epsilon_{CK} = 1 \times 10^{-3}$). The heat was previously transported to the phase front by conduction only. When the fluid can flow, convective heat transfer is added. Not just can the fluid flow influence the advancement of the phase front through convective heat transfer, but also vice-versa. The advancement of the phase front determines where the fluid is able to flow. In other words, there is a two-way coupling between the mechanical and thermal problem.

An interesting example of this has been published in Saldi's doctoral thesis [12], where a rectangular block of solid Gallium is melted by a hot wall on one side (see schematic in fig. 3.26). The phase front is initially straight and heat arrives through conduction. When a certain amount of fluid is available, natural convection circulation can occur. As a result, the phase front begins to become curved due to the locally increased heat supply from the impinging hot fluid transported by convection cells (blue arrow) from the hot wall to the cold phase

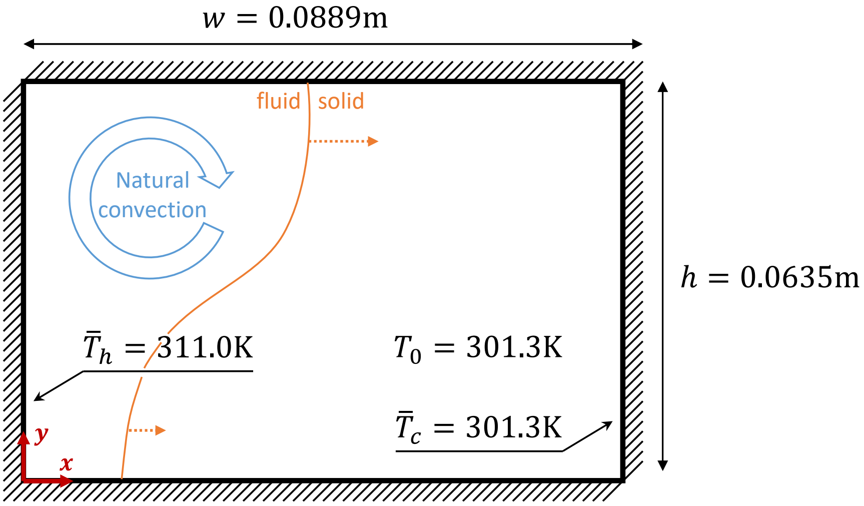


FIGURE 3.26: Schematic of gallium melting problem. Natural convection flow indicated (blue arrow) and curved phase front (orange) schematically in the middle of the simulation with non-uniform advancement (orange arrows) due to convective heat transfer to the phase front.

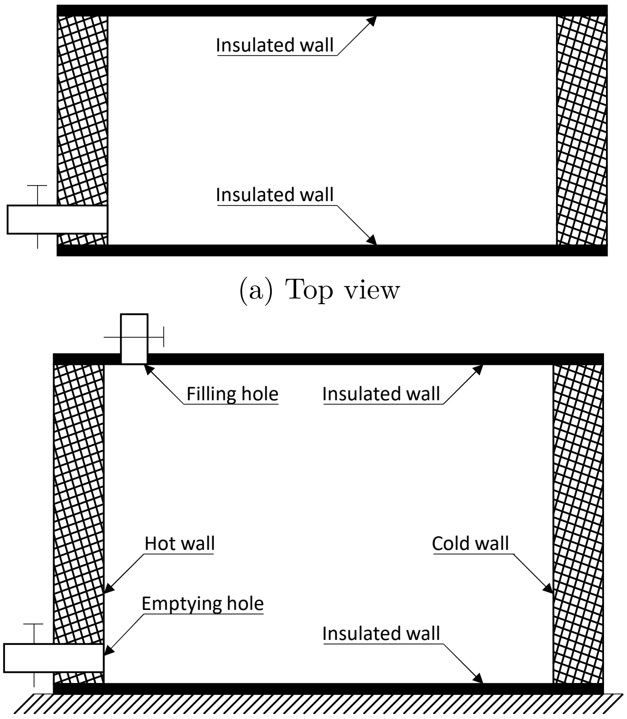
TABLE 3.12: Material properties.

Property	**Symbol**	**Value**	**Unit**
Density at T_{ref}	ρ_{ref}	6093.0	$\mathrm{kg\,m^{-3}}$
Viscosity	μ	1.81×10^{-3}	$\mathrm{kg\,m^{-3}}$
Heat capacity	c_p	381.5	$\mathrm{J\,kg^{-1}\,K}$
Conductivity	k	32.0	$\mathrm{W\,m^{-1}\,K}$
Latent heat	L_m	80160	$\mathrm{J\,kg^{-1}}$
Melting point	T_m	302.78	K
Thermal exp. coeff.	α_V	1.2×10^{-4}	$\mathrm{K^{-1}}$
Thermal diffusivity	α	1.38×10^{-5}	$\mathrm{m^2\,s^{-1}}$
Prandtl number	Pr	0.0216	-

front. This test case has been analyzed in more detail in Bobach et al. [105].

Saldi [12] uses an Eulerian 2D FVM based approach, implemented in OpenFOAM. The entire domain is a Newtonian fluid, but the solid subdomain experiences an increased flow resistance from an additional source term of the momentum equation, based on the Carman-Kozeny equation (see section 2.3.3). For the latent heat term Saldi employs an iterative method that determines the liquid fraction f_l, but it is not clear how it works in an isothermal phase change case such as this one, as some of the terms used in the iterative scheme are not defined for an isothermal phase change. The material properties are given in table 3.12. The results presented were obtained on a very fine mesh (number of cells $n_e = 1120 \times 800$), but the time resolution was not disclosed.

Saldi [12] compares his results with experimental data by Gau and Viskanta



(a) Top view

(b) Side view

FIGURE 3.27: Schematic of the gallium melting experimental setup by Gau and Viskanta [106].

[106]. They use a cavity (fig. 3.27) that is initially filled with liquid gallium. The gallium then cools, solidifies and reaches thermal equilibrium before the actual experiment begins. At that moment the hot and cold walls are brought to their set temperature and the process of melting begins. To obtain data for the advancement of the melt front at a given time, the experiment is aborted and the currently liquid gallium is rapidly drained from the cavity, leaving the remaining solid behind. The surface that was the melting front is photographed and traced to acquire the data for that time instance. This process is then repeated from the beginning for each of the time instances: 2, 3, 6, 8, 10, 12.5, 15, 17 and 19 minutes.

Saldi [12] also compares his results to 2D FVM simulations by Brent et al. [59] who also use the flow resistance approach for modeling the solid. The main difference is the use of an enthalpy formulation for the energy conservation equation with a modified heat capacity that takes the latent heat absorption into account. Brent et al. [59] coin the term enthalpy-porosity approach for this combination of models (see section 1.4.2). Brent et al. [59] use a very coarse mesh ($n_e = 42 \times 32$) and a large time step of $\Delta t = 5\,\mathrm{s}$ due to the limited CPU at their time. Saldi himself expresses doubt in the validity of the results of Brent et al. [59] due to the coarse mesh. Brent also provides stream lines and isotherms at some of the time marks, which are not discussed here (see Bobach et al. [107]).

A more recent work by Tiari et al. [108] uses the commercial software *ANSYS Fluent 17.0*, an Eulerian FVM code. The problem is modeled in 2D, with the enthalpy-porosity approach for the phase change phenomena. Their mesh has a resolution of $n_e = 250 \times 100$ elements and a time step of $\Delta t = 1\,\mathrm{s}$.

Similarly, the recent work by Sharma et al. [109] uses the commercial solver *COMSOL Multiphysics* based on the FEM in an Eulerian framework. The velocity-pressure system is solved using linear triangular elements and the Galerkin least-square (GLS) method. Again, the enthalpy-porosity approach is implemented to account for the phase change effects. Interestingly, for the thermal problem they use second order triangular elements, which *COMSOL Multiphysics* provides readily. Sharma et al. [109] do not disclose any details regarding the mesh and time step used, nor do they discuss the results in detail, as it is only one of many validation cases in [109].

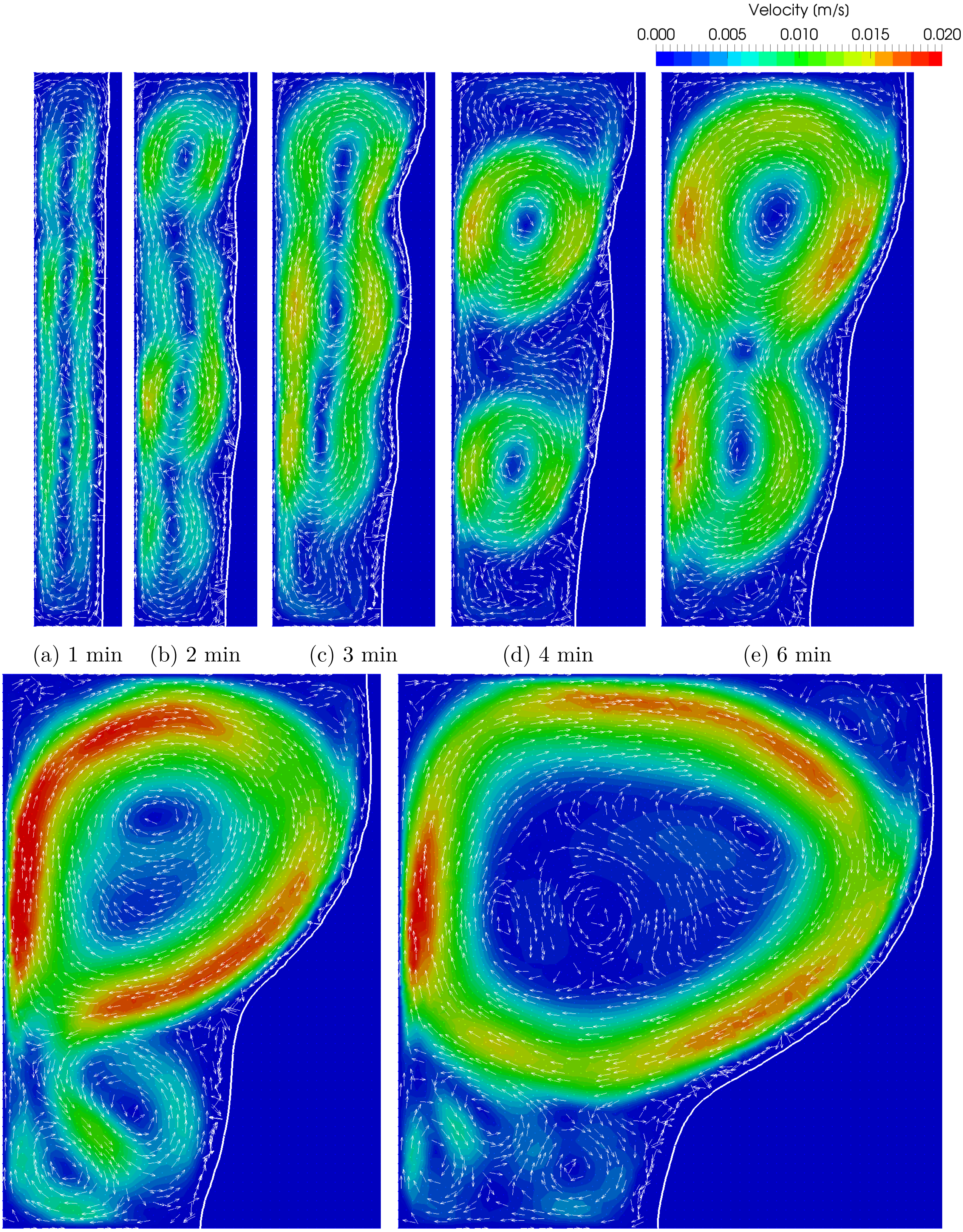
The results obtained in this work use an initially regular mesh with a number of nodes $n_n = 85 \times 61$, leading to a number of elements $n_e = 11280$. This initial edge length $\bar{l}_e \approx 0.001\,\mathrm{m}$ remains the smallest target mesh size throughout the simulation, but larger elements are allowed in regions with small velocity and temperature gradients to reduce the CPU cost. The total simulation time $t_{end} = 19 \times 60\,\mathrm{s}$ is reached in fixed time steps of $\Delta t = 0.05\,\mathrm{s}$. Following Brent et al., all boundaries assume a no-slip condition. As in the previous isothermal phase change test case, the following methods improve the stability of the simulation:

- A regularization with $\epsilon_{reg} = 0.4\,\mathrm{K}$ at the discontinuity of the liquid fraction function (see section 2.3.1).
- The Newton-Raphson method is enhanced with a line search algorithm (see section 2.6.2).
- The integration of the latent heat term uses 6 Gauss points instead of 1 in transitioning elements (see section 2.6.3).

The velocity field at several time instances is given in fig 3.28. One can observe the chaotic formation and collapse of different vortex systems, as the fluid region expands. The exact evolution of the vortex patterns is mesh-dependent, but the advancement of the melting front is not influenced significantly by the vortex patterns in the first few minutes. The pattern tends to stabilize between 4 and 5 minutes, independent of the mesh. Towards the end, a single large vortex and a small recirculation region at the bottom form at remain relatively stable until the end of the simulation.

The phase front location is determined using an isoline at $T = T_m$ and writing the coordinates of points on the line to file. This output is generated for each of the time instances given by Gau and Viskanta [106]. Since not all authors provide data for all the time instances, the results presented below only show comparisons, where data is available.

The first set of results in fig. 3.29 include data from Gau and Viskanata [106], Brent et al. [59] and Saldi [12], the latter only providing phase front data for



(a) 1 min (b) 2 min (c) 3 min (d) 4 min (e) 6 min

(f) 10 min (g) 17 min

FIGURE 3.28: Velocity field evolution.

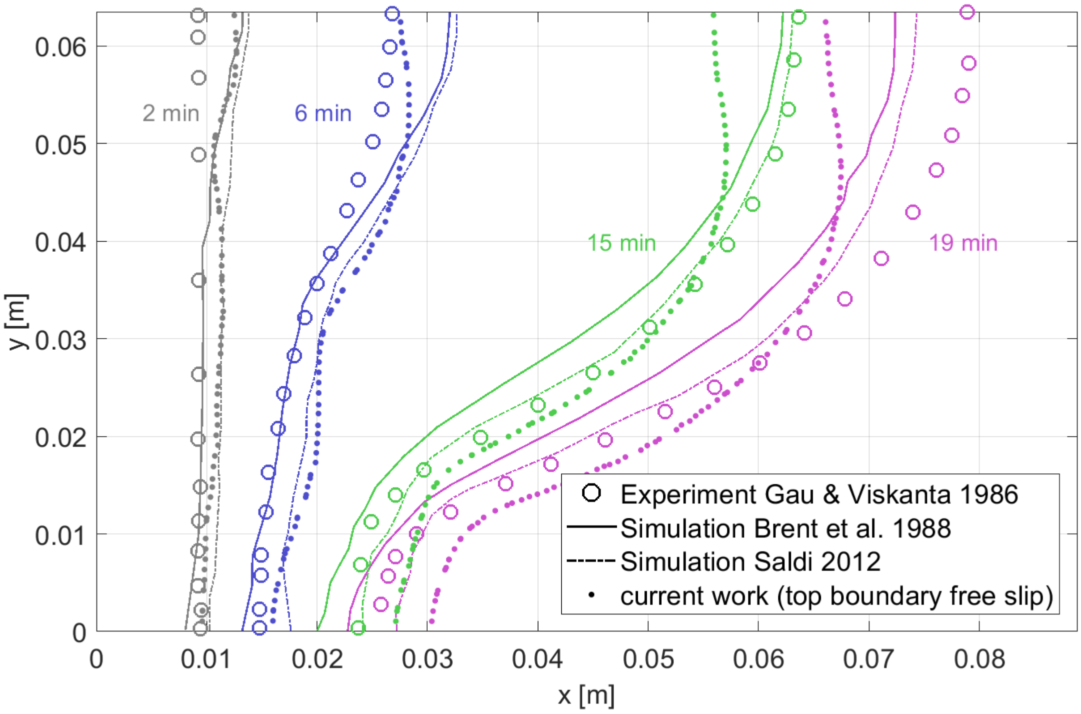


FIGURE 3.29: Comparison of results of gallium melting problem with experiments by Gau and Viskanta [106] and simulations by Brent et al. [59] and Saldi [12].

the 2, 6, 15 and 19 minute time instances. Overall, some agreement with the simulations by Saldi [12] and Brent et al. [59] is achieved. Notable differences exist for the top portion at the 6-minute, 15-minute and the 19-minute marks, as well as the bottom portion at the 15-minute and 19-minute marks. Regarding the latter, the results especially deviate from Brent's results, whose curve appears not to have the remarkable bend, where the heat transfer switches from convection dominated to conduction dominated. The smoother curve in Brent et al. [59] may be explained by the overall more smeared out results due to the lower mesh resolution. All simulation results seems to not match perfectly with the experimental results. At the 2-minute and 6-minute marks, the phase front in the simulations is ahead of the experimental one, while at the 15-minute and 19-minute marks, this is reversed. There are many possible explanations where either there is a systematic error in the experiment itself (e.g. imposed wall temperature not steady, adiabatic walls not well isolated, data acquisition not accurate, etc.) or where the simulations systematically do not represent the reality well (e.g. 3D effects, increased diffusivity through turbulence, air gap between fluid and top wall etc.). All of These points aret discussed in far greater detail in Bobach et al. [107].

The more recent publications by Tiari et al. [108] and Sharma et al. [109] only provide data for the 2, 6, 10, 15 minute instances and to avoid overloading fig. 3.29, these are shown separately in fig. 3.30. Brent et al. [59] and Gau and Viskanta [106] are included again, since they provide data for all the time instances. Both Tiari et al. [108] and Sharma et al. [109] share the following characteristics: the 2-minute and 6-minute marks show a particular waviness

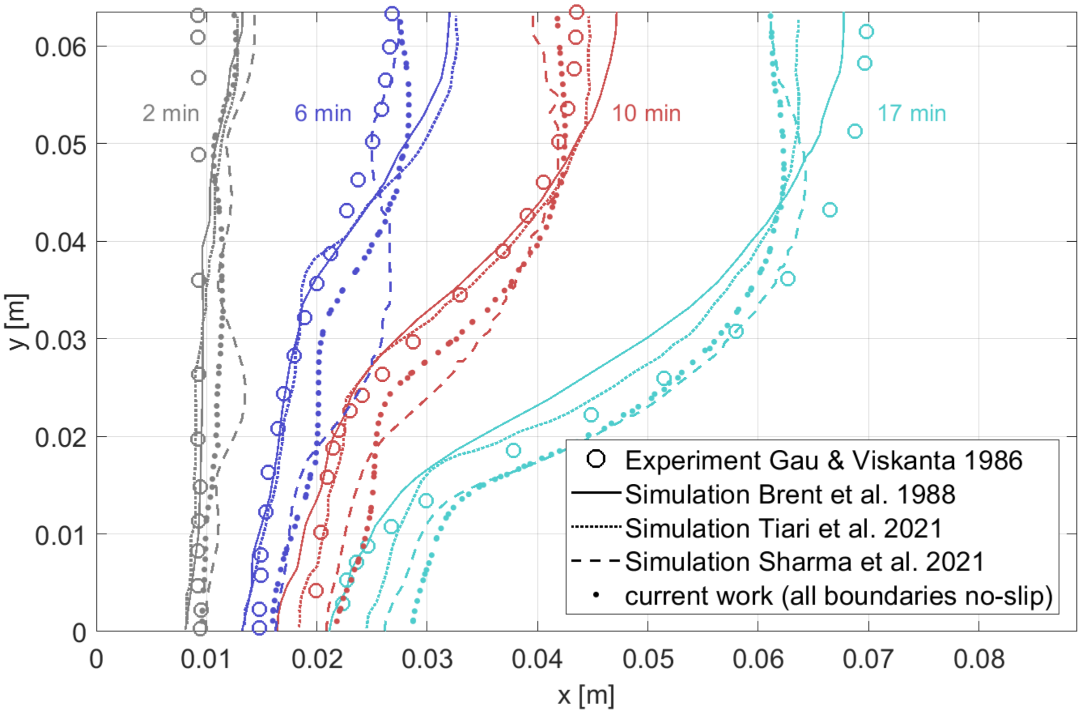


FIGURE 3.30: Comparison of results of gallium melting problem with experiments by Gau and Viskanta [106] and simulations by Brent et al. [59], Tiari et al. [108] and Sharma [109].

and the later 10-minute and 15-minute marks show a distinct bend at the bottom third and a more convex shape in the top half. Qualitatively these features are also found in the results of the present work, albeit the differences are quantitatively still large. The early waviness is caused by several small convection cells that carve into the phase front. The large convex shape later on is caused by one massive convection cell that appears to carve out the middle of the domain more than the top (as is the case for Brent et al. [59] and Saldi [12]). This is not supported by the experimental data, which seem to support a flow that causes the fastest advancement of the phase front at the top, as found by Brent et al. [59] and Saldi [12]. Once again, a more detailed discussion and an in-depth analysis of the flow field is laid out in Bobach et al. [107].

In summary, all simulation results agree well with one another and with the experimental data, although many differences can be found in the details. The method in this work appears to be on par with the other methods presented and the complex physics are captured sufficiently accurately.

3.4 Elastic solid

With the introduction of an elastic material model, the stress components are now of particular interest. First, an equivalent stress and a smoothing technique is introduced, before the verification test cases are presented.

The stress state in complex test cases is more easily understood by using an equivalent stress that combines the different stress components into one scalar quantity. Since the solids in the following test cases are metals, one of the most commonly used equivalent stress is the equivalent von Mises stress $\bar{\sigma}_{VM}$. It is usually used to compare the complex stress state with a single yield stress to determine if the stress state causes plasticity. Although plasticity is not yet implemented in the present method, $\bar{\sigma}_{VM}$ is still useful as it is well known and it gives a good idea about stress concentrations and the order of magnitude of the stress field. It is defined as follows:

$$\bar{\sigma}_{VM} = \sqrt{\frac{3}{2}\, s_{ij}\, s_{ij}} \tag{3.21}$$

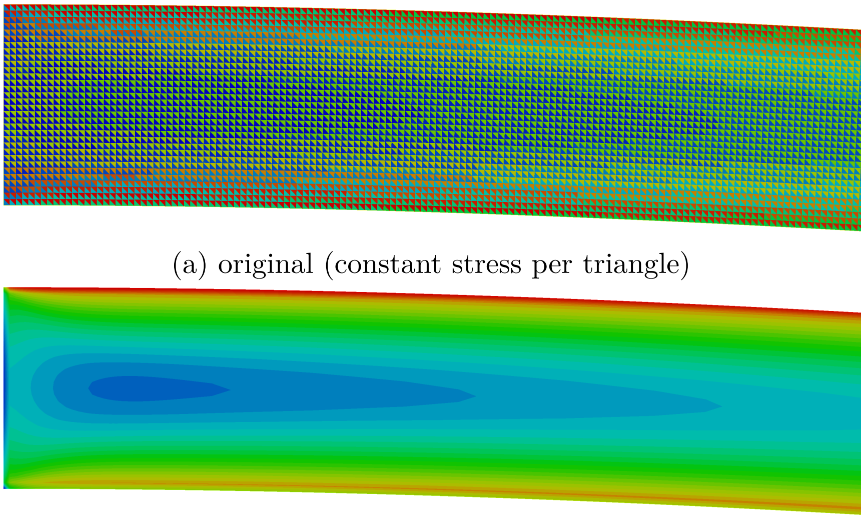
where $\boldsymbol{s}$ is the current deviatoric stress tensor.

Stresses can be smoothened when the element stress is too discontinuous. Due to the use of constant strain triangles (CST), the stress is constant within an element. This stress represents some sort of average stress over the area of the element. To obtain a smoother stress distribution that is easier to interpret two simple steps are required:

1. The stress components are each interpolated back from the Gauss points to the nodes using the shape functions.

2. The contributions from all adjacent elements of a given node are added and divided by the number of elements (i.e. arithmetic mean)

The result is a smooth stress field, as depicted in fig. 3.31.



(a) original (constant stress per triangle)

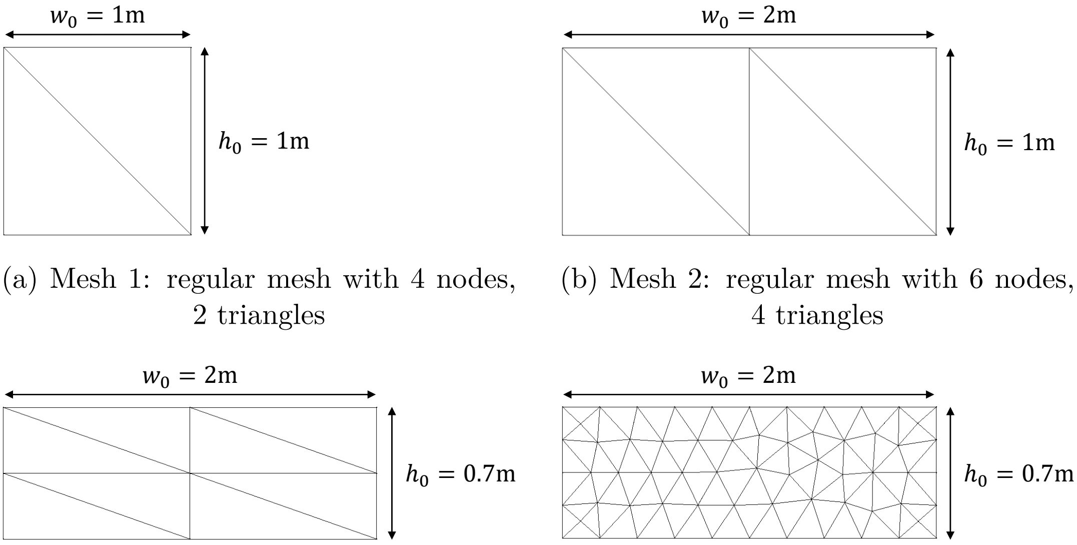
(b) smoothened (taken from fig. 3.42c)

Figure 3.31: Example of a smoothened stress component.

While the PFEM code used in this work was originally designed to simulate Newtonian fluids, a unified fluid and solid formulation is now implemented, as described in section 2.4.3. Before verifying the unified formulation on an example that combines the Newtonian fluid and the linear elastic solid, the elastic solid shall be verified in isolation first. Since a linear elastic solid with no dependence on temperature implemented in the finite element method is comparably simple, some simple verification test cases shall suffice to prove the correct implementation.

A variety of simple shapes paired with different materials is exposed to a variety of simple external loads and the resulting stresses and strains are compared to expected results obtained from basic continuum mechanics.

Four meshes are introduced that contain different number of elements and nodes. The first three are regular with increasing mesh size and the last one is irregular. See figure 3.32 for the dimensions, which vary between the meshes.



(a) Mesh 1: regular mesh with 4 nodes, 2 triangles

(b) Mesh 2: regular mesh with 6 nodes, 4 triangles

(c) Mesh 3: regular mesh with 9 nodes, 8 triangles

(d) Mesh 4: non-regular mesh with 64 nodes, 98 triangles

FIGURE 3.32: Meshes used for verification of solid material behavior.

In this section, the meshes are referred to by their numbers from 1 to 4. Likewise, 3 different materials are used throughout this section. The first material (steel) is chosen due to its relevance to the intended applications, such as welding and additive manufacturing. Steel is relatively stiff and dense and has a Poisson's ratio typical for many engineering materials. The second material (cork) on the other hand, is relatively compliant and light and has a Poisson's ratio close to 0, which shall be idealized to become exactly zero. This is interesting for the purpose of verification because no Poisson effect will occur. Likewise, the third material (rubber) is rather compliant and light, but nearly incompressible with a Poisson's ratio close to 0.5. All materials are assumed to have no viscosity.

TABLE 3.13: Material properties.

Property	Symbol	Steel	Cork	Rubber	Unit
Viscosity	μ	0.0	0.0	0.0	Pa s
Density	ρ	7000.0	150.0	1000.0	$\mathrm{kg\,m^{-3}}$
Poisson's ratio	ν	0.2766	0.0	0.4990	
Young's modulus	E	2.145×10^{11}	2.8×10^{7}	1.0×10^{7}	Pa
Shear modulus	G	8.4×10^{10}	1.4×10^{7}	3.336×10^{6}	Pa
Bulk modulus	K	1.6×10^{11}	9.333×10^{6}	1.666×10^{9}	Pa

The shear modulus G and bulk modulus K can be derived from Poisson's ratio ν and Young's modulus E using

$$G = \frac{E}{2(1+\nu)} \tag{3.22}$$

$$K = \frac{E}{3(1-2\nu)} \tag{3.23}$$

Likewise, one can convert the quantities back using

$$E = \frac{9KG}{3K+G} \tag{3.24}$$

$$\nu = \frac{3K-2G}{6K+2G} \tag{3.25}$$

Gravity will be idealized to be $10.0\,\mathrm{m\,s^{-2}}$, unless otherwise stated.

3.4.1 Rectangle compressed under a surface traction (dead load)

A rectangle is fixed at its bottom boundary and loaded at its top boundary with a downward surface traction. the surface traction's magnitude and direction are constant, which is also referred to as a dead load. Both the left and right walls are restricted from expanding or contracting horizontally, but are allowed to move vertically as the rectangle is compressed.

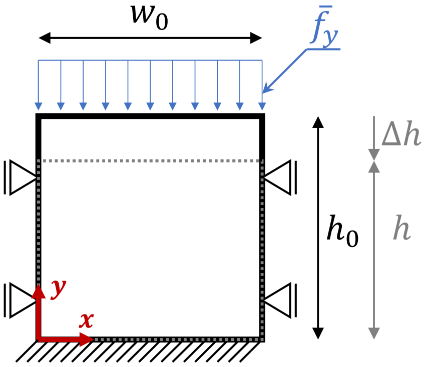


FIGURE 3.33: Schematic of a rectangle of initial width w_0 and initial height h_0 under a (compressive) surface traction. Bottom boundary fixed and side boundaries free-slip. Top boundary subjected to a surface traction in the y-direction $\bar{f}_y$. Final configuration indicated by grey dotted lines.

Considering a quasi-static loading, expected stresses, strains and displacements are easily obtained. In this setup, the expected normal stress in the y-direction σ_{yy} must be

$$\sigma_{yy} = \bar{f}_y \tag{3.26}$$

where $\bar{f}_y < 0$, if it is pointing downward, as depicted in fig 3.33. In this setup, where the lateral movement is restricted, the normal strain in the x-direction $\varepsilon_{xx} = 0$ and as a result, the normal stresses in the x-direction and z-direction depend on Poisson's ratio ν, following

$$\sigma_{xx} = \sigma_{zz} = \sigma_{yy}\frac{\nu}{1-\nu} \tag{3.27}$$

With the shear stress $\sigma_{xy} = 0$, all the stress components are known and the normal strain in the y-direction ε_{yy} can be computed.

$$\varepsilon_{yy} = \sigma_{yy}\frac{1}{E}\frac{(1+\nu)(1-2\nu)}{1-\nu} \tag{3.28}$$

The change in height Δh can be computed from ε_{yy}, assuming small displacements, as

$$\Delta h = \varepsilon\, h_0 \tag{3.29}$$

As established in eq. 2.52, the pressure is a third of the trace of the stress tensor. Now, the loading is chosen to be $\bar{f}_y = -1.0 \times 10^9\,\mathrm{Pa}$ for steel and $\bar{f}_y = -1.0 \times 10^5\,\mathrm{Pa}$ for cork and rubber. No gravity is acting and all fields are initialized with a value of 0. The inertial term is removed from the momentum equation. A time step $\Delta t = 0.01\,\mathrm{s}$ over a total simulation time $t_{end} = 0.1\,\mathrm{s}$ is chosen. All 3 materials listed in table 3.13 are used. Tables 3.14 - 3.16 show the results on mesh 4 compared with the analytical values for all 3 materials. In these results, the relative errors are of the order of 10^{-8} or lower on the largest mesh. The other meshes produce equal or lower relative errors.

Table 3.14: Results for dead load with constrained side wall on mesh 4 and with steel. Expected results according to eqs. 3.26 - 3.29.

Quantity	**Symbol**	**Obtained**	**Expected**	**Unit**
Mass	m	9800.0	9800.0	kg
Height	h	0.69743	0.69743	m
Width	w	2.0	2.0	m
Pressure	p	5.8824×10^{8}	5.8824×10^{8}	Pa
Normal Stress in the x-direction	σ_{xx}	-3.8235×10^{8}	-3.8235×10^{8}	Pa
Normal Stress in the y-direction	σ_{yy}	-1.0×10^{9}	-1.0×10^{9}	Pa
Shear Stress	σ_{xy}	-2.235×10^{-8}	0.0	Pa

Table 3.15: Results for dead load with constrained side wall on mesh 4 and with cork ($\nu = 0.0$). Expected results according to eqs. 3.26 - 3.29.

Quantity	**Symbol**	**Obtained**	**Expected**	**Unit**
Mass	m	210.0	210.0	kg
Height	h	0.69750	0.69750	m
Width	w	2.0	2.0	m
Pressure	p	3.3333×10^{4}	3.3333×10^{4}	Pa
Normal Stress in the x-direction	σ_{xx}	-1.8610×10^{-9}	0.0	Pa
Normal Stress in the y-direction	σ_{yy}	-1.0×10^{5}	-1.0×10^{5}	Pa
Shear Stress	σ_{xy}	1.477×10^{-9}	0.0	Pa

Alternatively, the wall on the right can be free to move instead of only allowing a displacement in the y-direction. Such a setup is depicted in fig. 3.34.

TABLE 3.16: Results for dead load with constrained side wall on mesh 4 and with rubber ($\nu = 0.4990$). Expected results according to eqs. 3.26 - 3.29.

Quantity	Symbol	Obtained	Expected	Unit
Mass	m	1400.0	1400.0	kg
Height	h	0.69996	0.69996	m
Width	w	2.0	2.0	m
Pressure	p	9.9734×10^4	9.9734×10^4	Pa
Normal Stress in the x-direction	σ_{xx}	-9.9600×10^4	-9.9600×10^4	Pa
Normal Stress in the y-direction	σ_{yy}	-1.0×10^5	-1.0×10^5	Pa
Shear Stress	σ_{xy}	1.5049×10^{-8}	0.0	Pa

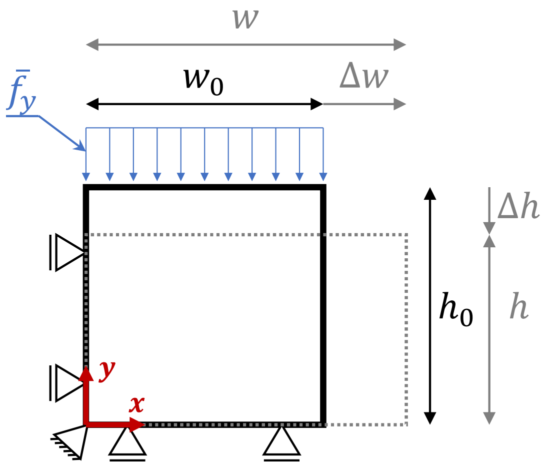


FIGURE 3.34: Schematic of a rectangle of initial width w_0 and initial height h_0 under surface traction. Bottom boundary fixed, the left side boundary free-slip and right side boundary free. Top boundary subjected to a surface traction in the y-direction $\bar{f}_y$. Final configuration indicated by grey dotted lines.

Instead of $\varepsilon_{xx} = 0$ and $\sigma_{xx} \neq 0$, as in the previous setup, this setup will produce $\varepsilon_{xx} \neq 0$ and $\sigma_{xx} = 0$. With $\sigma_{yy} = \bar{f}_y$, as before, the normal stress in z-direction σ_{zz} can be determined first:

$$\sigma_{zz} = \nu\sigma_{yy} \tag{3.30}$$

Now, with all normal stresses known and the shear stress $\sigma_{xy} = 0$, the strain in x and y-directions ε_{xx} and ε_{yy} can be determined respectively as

$$\varepsilon_{xx} = \frac{1}{E}(\quad \sigma_{xx} - \nu\sigma_{yy} - \nu\sigma_{zz}) \tag{3.31}$$

$$\varepsilon_{yy} = \frac{1}{E}(-\nu\sigma_{xx} + \quad \sigma_{yy} - \nu\sigma_{zz}) \tag{3.32}$$

As before, the displacements Δh and Δw are calculated from the strains under the small strain assumption, such that

$$\Delta h = h_0 \varepsilon_{xx} \tag{3.33}$$

$$\Delta w = w_0 \varepsilon_{yy} \tag{3.34}$$

Finally, we define the load $\bar{f}_y$ for each material to be the same as in the previous test. The integration of the surface traction over the line element that represents the surface, is performed on the reference configuration. The reference configuration for a solid element is the initial configuration in this test. This is done to remain consistent with the computation of the stresses, which is also performed on the reference configuration, as described in section 2.6.1. The results for mesh 4 are listed in tables 3.17 - 3.19.

Table 3.17: Results for dead load with deforming side wall on mesh 4 and with steel. Expected results according to eqs. 3.30 - 3.34.

Quantity	**Symbol**	**Obtained**	**Expected**	**Unit**
Mass	m	9800.0	9800.0	kg
Height	h	0.69699	0.69699	m
Width	w	2.0033	2.0033	m
Pressure	p	4.2553×10^8	4.2553×10^8	Pa
Normal Stress in the x-direction	σ_{xx}	1.3867×10^{-7}	0.0	Pa
Normal Stress in the y-direction	σ_{yy}	-1.0×10^9	-1.0×10^9	Pa
Shear Stress	σ_{xy}	-6.546×10^{-7}	0.0	Pa

Table 3.18: Results for dead load with deforming side wall on mesh 4 and with cork ($\nu = 0.0$). Expected results according to eqs. 3.30 - 3.34.

Quantity	**Symbol**	**Obtained**	**Expected**	**Unit**
Mass	m	210.0	210.0	kg
Height	h	0.69750	0.69750	m
Width	w	2.0	2.0	m
Pressure	p	3.3333×10^4	3.3333×10^4	Pa
Normal Stress in the x-direction	σ_{xx}	-3.672×10^{-7}	0.0	Pa
Normal Stress in the y-direction	σ_{yy}	-1.0×10^5	-1.0×10^5	Pa
Shear Stress	σ_{xy}	6.706×10^{-9}	0.0	Pa

Some small errors are encountered in some of the tests. The largest error is observed in the last example, the rubber material under a dead load with an unrestricted side wall. There, σ_{xx} has an absolute error of $\approx 3 \times 10^{-2}$, which is a relative error of $\approx 3 \times 10^{-7}$ if the loading $\bar{f}_y = \sigma_{yy}$ is used as reference. In

Table 3.19: Results for dead load with deforming side wall on mesh 4 and with rubber ($\nu = 0.4990$). Expected results according to eqs. 3.30 - 3.34.

Quantity	Symbol	Obtained	Expected	Unit
Mass	m	1399.8	1400.0	kg
Height	h	0.69474	0.69474	m
Width	w	2.01496	2.014960	m
Pressure	p	4.9967×10^4	4.9967×10^4	Pa
Normal Stress in the x-direction	σ_{xx}	-3.1166×10^{-2}	0.0	Pa
Normal Stress in the y-direction	σ_{yy}	-1.0×10^5	-1.0×10^5	Pa
Shear Stress	σ_{xy}	-1.8309×10^{-4}	0.0	Pa

fact, most relative errors for the pressure and the stresses are of the order of 10^{-7} and some are lower. Such an error is acceptable for a quasi-incompressible solid and these tests are considered successful.

The tests have also been repeated with a time dependent load $\bar{f}_{y,sin}(t)$ that follows a sine wave to return to its initial configuration:

$$\bar{f}_{y,sin}(t) = \bar{f}_y \sin\left(\frac{t\pi}{t_{end}}\right) \tag{3.35}$$

where $\bar{f}_y$ is the magnitude of the sine, which is kept the same as described above. The total simulation time is $t_{end} = 2.0\,\mathrm{s}$ with a fine time step $\Delta t = 0.01\,\mathrm{s}$ to resolve the sine function. Upon returning to the initial configuration at $t = t_{end}$, all stresses and displacements should have disappeared. This is indeed the case and the errors are found to be of the same order of magnitude or lower as in the constant loading above (not shown).

3.4.2 Uniaxial elongation by an imposed displacement

Contrary to the test above, the deformation is now caused by an imposed velocity instead of an external force. There are again two distinct ways to set up such a test: with the transversal contraction due to the Poisson effect allowed or prohibited. In the following, both setups are tested and compared with the easily derived analytical solution. In both cases, there is a rectangle of initial width w_0 and initial height h_0, being elongated in the x-direction by an imposed constant x-velocity $\bar{v}_x$.

With the contraction being prohibited, a setup may look like the one depicted in fig. 3.35, where the top wall has a y-velocity imposed ($\bar{v}_y = 0$), but is free to slide in the x-direction (i.e. a roller boundary condition) as the rectangle is stretched. Note that in this case, none of the corner points have any degrees of freedom. This means that for a single rectangle or two triangle elements, all nodal velocities are imposed, which makes such a setup a good verification test case for the implementation of the constitutive model.

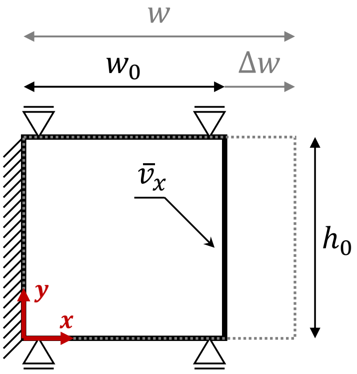


FIGURE 3.35: Schematic rectangle with imposed x-velocity on the right wall, top wall constrained (no deformation due to Poisson effect). Final configuration illustrated in dotted grey lines.

We expect $\sigma_{xx} \neq 0$ due to the stretching and $\sigma_{yy} \neq 0$ due to the restricted contraction of the Poisson effect. The normal strain rate in the x-direction $\dot{\varepsilon}_{xx}$ for small strains is

$$\dot{\varepsilon}_{xx} = \bar{v}_x / w_0 \tag{3.36}$$

and the total normal strain in the x-direction ε_{xx} over the entire duration of the simulation t_{end} is

$$\varepsilon_{xx} = \dot{\varepsilon}_{xx} t_{end} \tag{3.37}$$

The resulting normal stresses in the x-direction σ_{xx} is

$$\sigma_{xx} = \varepsilon_{xx} \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \tag{3.38}$$

and the normal stress in the y-direction σ_{yy} is

$$\sigma_{yy} = \varepsilon_{xx} \frac{E\,\nu}{(1+\nu)(1-2\nu)} \tag{3.39}$$

The imposed velocity is chosen to be $\bar{v}_x = 1.0\,\mathrm{m\,s^{-1}}$ for all materials over a total simulation time of $t_{end} = 0.1\,\mathrm{s}$, in order to induce a noticeable deformation, without violating the small displacement assumption. No gravity is acting and all fields are initialized with a value of 0. A small time step $\Delta t = 0.001\,\mathrm{s}$ is used. As before, results are only shown for mesh 4 in tables 3.20 - 3.22 for the 3 materials. Once again, an excellent agreement with the theoretical solution is obtained, where the largest errors are of the order of 10^{-7} for the steel on mesh 4 and lower for all other material-mesh combinations.

Table 3.20: Results for imposed elongation with constrained top wall on mesh 4 and with steel. Expected results according to eqs. 3.36 - 3.39.

Quantity	**Symbol**	**Obtained**	**Expected**	**Unit**
Mass	m	9800.0	9800.0	kg
Height	h	0.7	0.7	m
Width	w	2.1	2.1	m
Pressure	p	-8.000×10^{9}	-8.000×10^{9}	Pa
Normal Stress in the x-direction	σ_{xx}	1.360×10^{10}	1.360×10^{10}	Pa
Normal Stress in the y-direction	σ_{yy}	5.200×10^{9}	5.200×10^{9}	Pa
Shear Stress	σ_{xy}	-8.921×10^{-7}	0.0	Pa

Table 3.21: Results for imposed elongation with constrained top wall on mesh 4 and with cork ($\nu = 0.0$). Expected results according to eqs. 3.36 - 3.39.

Quantity	**Symbol**	**Obtained**	**Expected**	**Unit**
Mass	m	210.0	210.0	kg
Height	h	0.7	0.7	m
Width	w	2.1	2.1	m
Pressure	p	-4.667×10^{5}	-4.667×10^{5}	Pa
Normal Stress in the x-direction	σ_{xx}	1.400×10^{6}	1.400×10^{6}	Pa
Normal Stress in the y-direction	σ_{yy}	2.328×10^{-10}	0.0	Pa
Shear Stress	σ_{xy}	6.101×10^{-11}	0.0	Pa

Table 3.22: Results for imposed elongation with constrained top wall on mesh 4 and with rubber ($\nu = 0.4990$). Expected results according to eqs. 3.36 - 3.39.

Quantity	**Symbol**	**Obtained**	**Expected**	**Unit**
Mass	m	1400.0	1400.0	kg
Height	h	0.7	0.7	m
Width	w	2.1	2.1	m
Pressure	p	-8.333×10^{7}	-8.333×10^{7}	Pa
Normal Stress in the x-direction	σ_{xx}	8.356×10^{7}	8.356×10^{7}	Pa
Normal Stress in the y-direction	σ_{yy}	8.322×10^{7}	8.322×10^{7}	Pa
Shear Stress	σ_{xy}	3.206×10^{-10}	0.0	Pa

3.4.3 Biaxial elongation by an imposed displacement

The same test as above is repeated, but with the top surface free to move and the rectangle free to contract in the transversal direction due to the Poisson effect. The schematic setup of this test is depicted in fig. 3.36.

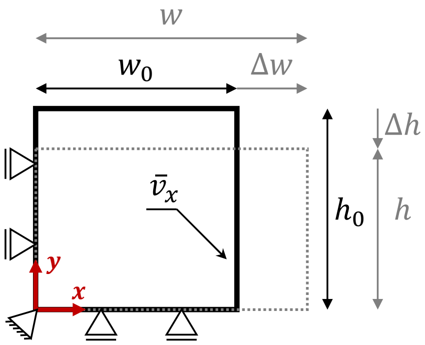


FIGURE 3.36: Schematic rectangle with imposed x-velocity on the right wall, top wall free to contract in the y-direction. Final configuration indicated by dotted grey lines.

With an imposed velocity in the x-direction as before, eqs. 3.36 and 3.37 remain valid in this case. Given that the contraction is allowed, the stress in the y-direction σ_{yy} is under plane strain assumption

$$\sigma_{yy} = 0 = \frac{E}{(1+\nu)(1-2\nu)} \left(\varepsilon_{xx}\nu + \varepsilon_{yy}(1-\nu)\right) \tag{3.40}$$

which results in a normal strain in the y-direction ε_{yy}

$$\varepsilon_{yy} = -\varepsilon_{xx}\frac{\nu}{1-\nu} \tag{3.41}$$

Because of this non-zero strain in the y-direction, the normal stress in the x-direction σ_{xx} now becomes

$$\sigma_{xx} = \frac{E}{(1+\nu)(1-2\nu)} \left(\varepsilon_{xx}(1-\nu) + \varepsilon_{yy}\nu\right) \tag{3.42}$$

Simulations are run with the same imposed x-velocity $\bar{v}_x = 1.0\,\mathrm{m\,s^{-1}}$ over a total simulation time $t_{end} = 0.1\,\mathrm{s}$ with a time step $\Delta t = 0.001\,\mathrm{s}$. Results in tables 3.23 - 3.25 are once again shown for all three materials, but only for mesh 4. As before, the agreement is excellent and the largest errors are in the order of 10^{-7} for the steel on mesh 4 and lower for all other material-mesh combinations.

Table 3.23: Results for imposed elongation with free top wall on mesh 4 and with steel. Expected results according to eqs. 3.40 - 3.42.

Quantity	**Symbol**	**Obtained**	**Expected**	**Unit**
Mass	m	9800.0	9800.0	kg
Height	h	0.68662	0.68662	m
Width	w	2.1	2.1	m
Pressure	p	-4.941×10^{9}	-4.941×10^{9}	Pa
Normal Stress in the x-direction	σ_{xx}	1.161×10^{10}	1.161×10^{10}	Pa
Normal Stress in the y-direction	σ_{yy}	-7.629×10^{9}	0.0	Pa
Shear Stress	σ_{xy}	-3.581×10^{-7}	0.0	Pa

Table 3.24: Results for imposed elongation with free top wall on mesh 4 and with cork ($\nu = 0.0$). Expected results according to eqs. 3.40 - 3.42.

Quantity	**Symbol**	**Obtained**	**Expected**	**Unit**
Mass	m	210.0	210.0	kg
Height	h	0.7000	0.7000	m
Width	w	2.1	2.1	m
Pressure	p	-4.667×10^{5}	-4.667×10^{5}	Pa
Normal Stress in the x-direction	σ_{xx}	1.400×10^{6}	1.400×10^{6}	Pa
Normal Stress in the y-direction	σ_{yy}	2.910×10^{-10}	0.0	Pa
Shear Stress	σ_{xy}	9.803×10^{-12}	0.0	Pa

Table 3.25: Results for imposed elongation with free top wall on mesh 4 and with rubber ($\nu = 0.4990$). Expected results according to eqs. 3.40 - 3.42.

Quantity	**Symbol**	**Obtained**	**Expected**	**Unit**
Mass	m	1400.0	1400.0	kg
Height	h	0.66514	0.66514	m
Width	w	2.1	2.1	m
Pressure	p	-3.327×10^{5}	-3.327×10^{5}	Pa
Normal Stress in the x-direction	σ_{xx}	6.656×10^{5}	6.656×10^{5}	Pa
Normal Stress in the y-direction	σ_{yy}	-9.895×10^{-10}	0.0	Pa
Shear Stress	σ_{xy}	1.164×10^{-11}	0.0	Pa

3.4.4 Simple shear by an imposed displacement

A rectangle of initial width w_0 and height h_0 has its bottom wall fixed, the top wall with an imposed velocity in the x-direction $\bar{v}_x$, therefore experiencing shear. The left and right walls also have an imposed velocity in the x-direction, such that there is a linear distribution (blue arrows in fig. 3.37) between the bottom that is at rest and the top that moves at $\bar{v}_x$. If, additionally, all the walls are not allowed to move in the y-direction, then this is referred to as

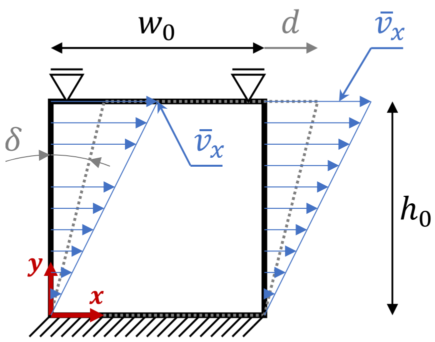


FIGURE 3.37: Schematic of a rectangle under simple shear, imposed velocity $\bar{v}_x$ and $\bar{v}_y = 0$ on the top wall, side walls imposed velocity distribution from $\bar{v}_x$ at the top to 0 at the bottom (blue arrows), bottom wall fixed. Final configuration indicated by dotted grey lines. The displacement of the top right corner is d and the angle at which the left and right walls deflect is δ.

simple shear. In this case, all boundaries have all their velocities imposed and no degrees of freedom are left on the boundaries. Nodes internal to the mesh on the other hand are free to move. Using a single rectangle or two triangle elements (mesh 1) without any internal nodes means that all nodal movement is prescribed, making this another good verification test case.

As long as the deformation is small, simple shear has no normal stresses and strains:

$$\sigma_{xx} = \sigma_{yy} = \sigma_{zz} = \varepsilon_{xx} = \varepsilon_{yy} = \varepsilon_{zz} = 0 \tag{3.43}$$

Given the imposed x-velocity $\bar{v}_x$, the shear rate γ_s and the shear strain rate $\dot{\varepsilon}_{xy}$ are

$$\dot{\varepsilon}_{xy} = \frac{1}{2}\gamma_s = \frac{\bar{v}_x(y = h_0)}{2h_0} \tag{3.44}$$

where $\bar{v}_x(y = h_0)$ is the imposed velocity at the top surface. With $\dot{\varepsilon}_{xy}$ known, the total shear strain ε_{xy} over the total duration of the simulation t_{end} is therefore

$$\varepsilon_{xy} = \dot{\varepsilon}_{xy} t_{end} \tag{3.45}$$

The resulting shear stress σ_{xy} is expected to be

$$\sigma_{xy} = \frac{E}{1+\nu}\varepsilon_{xy} = 2G\,\varepsilon_{xy} \tag{3.46}$$

The simulations are run until a total time of $t_{end} = 0.1\,\text{s}$ with a time step of $\Delta t = 0.001\,\text{s}$. Unlike in the tension test case, we only use mesh 1 and mesh 2 with the three different materials. Only results on mesh 2 are shown here. The agreement is excellent with the largest error of the order of 10^{-8} encountered for steel.

Table 3.26: Results for simple shear with an imposed velocity on mesh 2 and with steel. Expected results according to eqs. 3.43 - 3.46.

Quantity	**Symbol**	**Obtained**	**Expected**	**Unit**
Mass	m	14000.0	14000.0	kg
Pressure	p	1.563×10^{-8}	0.0	Pa
Normal Stress in the x-direction	σ_{xx}	1.563×10^{-8}	0.0	Pa
Normal Stress in the y-direction	σ_{yy}	1.563×10^{-8}	0.0	Pa
Shear Stress	σ_{xy}	8.400×10^{9}	8.400×10^{9}	Pa

Table 3.27: Results for simple shear with an imposed velocity on mesh 2 and with cork ($\nu = 0.0$). Expected results according to eqs. 3.43 - 3.46.

Quantity	**Symbol**	**Obtained**	**Expected**	**Unit**
Mass	m	300.0	300.0	kg
Pressure	p	1.415×10^{-13}	0.0	Pa
Normal Stress in the x-direction	σ_{xx}	1.415×10^{-13}	0.0	Pa
Normal Stress in the y-direction	σ_{yy}	1.415×10^{-13}	0.0	Pa
Shear Stress	σ_{xy}	1.400×10^{5}	1.400×10^{5}	Pa

Table 3.28: Results for simple shear with an imposed velocity on mesh 2 and with rubber ($\nu = 0.4990$). Expected results according to eqs. 3.43 - 3.46.

Quantity	**Symbol**	**Obtained**	**Expected**	**Unit**
Mass	m	2000.0	2000.0	kg
Pressure	p	1.427×10^{-9}	0.0	Pa
Normal Stress in the x-direction	σ_{xx}	1.427×10^{-9}	0.0	Pa
Normal Stress in the y-direction	σ_{yy}	1.427×10^{-9}	0.0	Pa
Shear Stress	σ_{xy}	3.336×10^{5}	3.336×10^{5}	Pa

3.4.5 Cantilever beam under distributed load - quasi-static case

A thin long beam of length l and height h and out-of-plane thickness b is fixed at the left boundary ($x = 0$) and is free at all other boundaries. On the top boundary, a constant distributed load $\bar{f}_y$ is applied. A simple analytical solution from beam theory (Euler-Bernoulli beam) for the equilibrium configuration is used and compared with simulations.
The bending moment M_b at any cross section x is

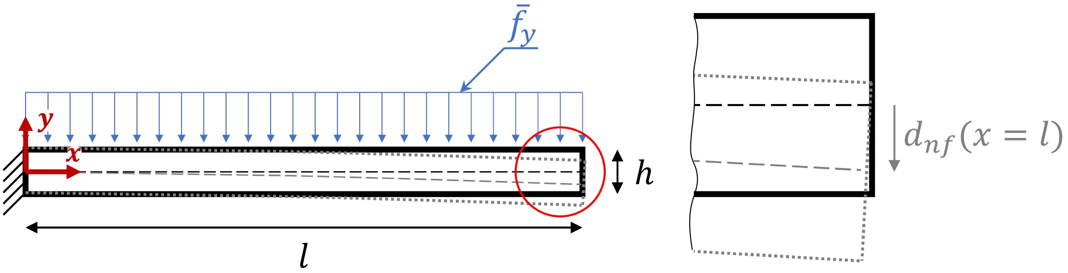


FIGURE 3.38: Schematic of a cantilever beam under a line load of magnitude $\bar{f}_y$. Final configuration illustrated indicated by dotted grey lines, neutral fiber in thin dashed lines with the deflection of the neutral fiber in the y-direction d_{nf} at the tip $x = l$ as a grey arrow.

$$M_b = \frac{\bar{f}_y}{2}\left(x^2 - 2xl + l^2\right) \tag{3.47}$$

With the moment of inertia for a rectangular cross section $I_{y_{rect}}$ defined as

$$I_{y_{rect}} = \frac{bh^3}{12} \tag{3.48}$$

where b is the thickness (here constant in plane strain with $b = 1\,\mathrm{m}$) and h is the height, the curvature κ_b can be evaluated using

$$\kappa_b = \frac{M_b}{EI_{y_{rect}}} \tag{3.49}$$

where E is Young's modulus. Integrating the curvature κ_b over x once yields the slope of the neutral fiber θ_{nf} and integrating again over x yields the y-displacement of the neutral fiber d_{nf}

$$\theta_{nf} = \frac{\bar{f}_y x}{8EI_{y_{rect}}}\left(x^2 - 3xl + 3l^2\right) \tag{3.50}$$

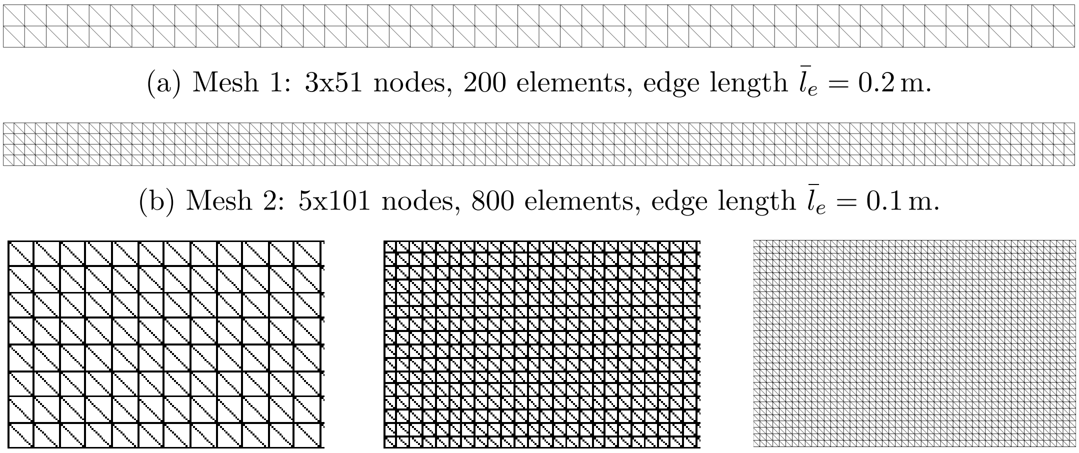
$$d_{nf} = \frac{\bar{f}_y x^2}{24EI_{y_{rect}}}\left(x^2 - 4xl + 6l^2\right) \tag{3.51}$$

These equations describe the deformed beam in its final, static configuration. The material used resembles the steel from the previous tests, but with Poisson's ratio set to $\nu = 0$ to simplify the problem, as given in table 3.29.

The beam has a length $l = 10.0\,\mathrm{m}$ and height $h = 0.4\,\mathrm{m}$. The surface traction $\bar{f}_y = -2.0 \times 10^6\,\mathrm{Pa}$ is applied immediately and constantly and with constant direction in $-y$ for the duration of the simulation $t_{end} = 1.0\,\mathrm{s}$. A time step of $\Delta t = 0.1\,\mathrm{s}$ is chosen with the inertia term deactivated in the momentum equation. We compare several meshes (fig. 3.39) of increasingly higher density to show mesh convergence (fig. 3.40) of the results. The simplest metric for convergence is the maximum deflection of the neutral fiber $\max(d_{nf})$. Mesh convergence is reached with mesh 5 (fig. 3.39e), but mesh 4 (fig. 3.39d) can also be used for faster simulation runs at only slightly reduced accuracy. The shape

Table 3.29: Material properties.

Property	**Symbol**	**Value**	**Unit**
Viscosity	μ	0.0	Pa s
Density	ρ	7000.0	$\mathrm{kg\,m^{-3}}$
Poisson's ratio	ν	0.0	
Young's modulus	E	4.8×10^{11}	Pa
Bulk modulus	G	2.4×10^{11}	Pa
Shear modulus	K	1.6×10^{11}	Pa



(a) Mesh 1: 3x51 nodes, 200 elements, edge length $\bar{l}_e = 0.2\,\mathrm{m}$.

(b) Mesh 2: 5x101 nodes, 800 elements, edge length $\bar{l}_e = 0.1\,\mathrm{m}$.

(c) Mesh 3: 9x201 nodes, 3200 triangles, edge length $\bar{l}_e = 0.05\,\mathrm{m}$ (cut view).

(d) Mesh 4: 17x401 nodes, 12800 triangles, edge length $\bar{l}_e = 0.025\,\mathrm{m}$ (cut view).

(e) Mesh 5: 33x801 nodes, 51200 triangles, edge length $\bar{l}_e = 0.0125\,\mathrm{m}$ (cut view).

Figure 3.39: Meshes used.

of the whole deflected neutral fiber is depicted in fig. 3.41 and compared with the one obtained with eq. 3.51. Both are virtually indistinguishable.

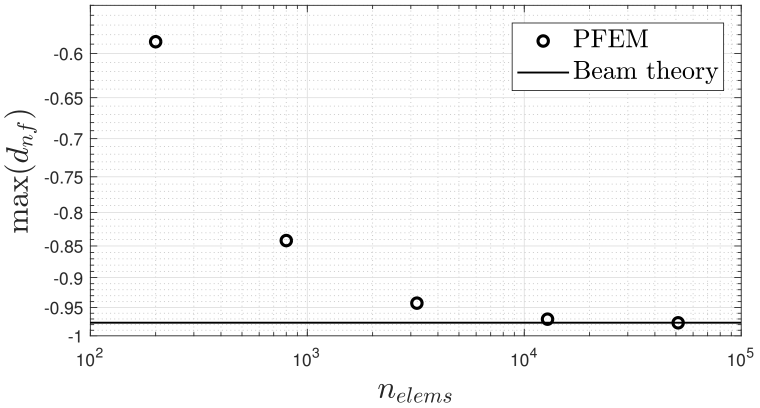


Figure 3.40: Mesh convergence of neutral fiber deflection.

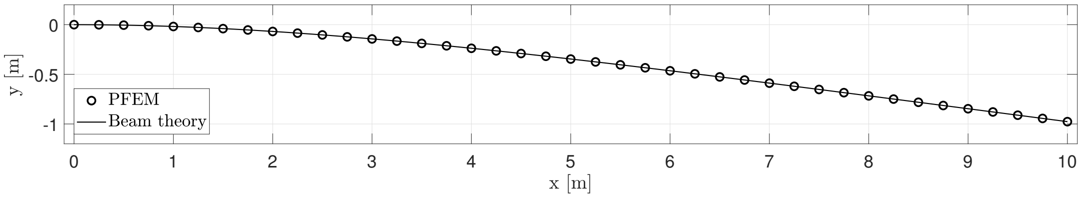


FIGURE 3.41: Comparison of the deflected neutral fiber of a cantilever beam using mesh 5 with beam theory

To further verify the implementation of the elastic solid model, the stresses in the beam can be compared. The downward bending of the beam causes tensile and compressive stresses that are largest in the fibers at the top ($y = h/2$) and bottom ($y = -h/2$) boundaries respectively.

$$\sigma_{xx}(x, y) = \frac{M(x)\, y}{I_{yrect}} \tag{3.52}$$

The maximum tension $\max(\sigma_{xx})$ and compression $\min(\sigma_{xx})$ are evaluated as

$$\max(\sigma_{xx}) = \sigma_{xx}(x = 0, y = h/2) \quad = \frac{\bar{f}_y h l^2}{4I_{yrect}} \tag{3.53}$$

$$\min(\sigma_{xx}) = \sigma_{xx}(x = 0, y = -h/2) = -\frac{\bar{f}_y h l^2}{4I_{yrect}} \tag{3.54}$$

The obtained shear stress can be also compared with an analytical solution. First, the shear force F_{shear} is calculated for this setup as

$$F_{shear} = \frac{dM_b}{dx} = \bar{f}_y(x - l) \tag{3.55}$$

with the first moment of Area for a rectangle Q_{rect} as a function of y being defined as

$$Q_{rect} = \frac{b}{2}\left(\frac{h^2}{4} - y^2\right) \tag{3.56}$$

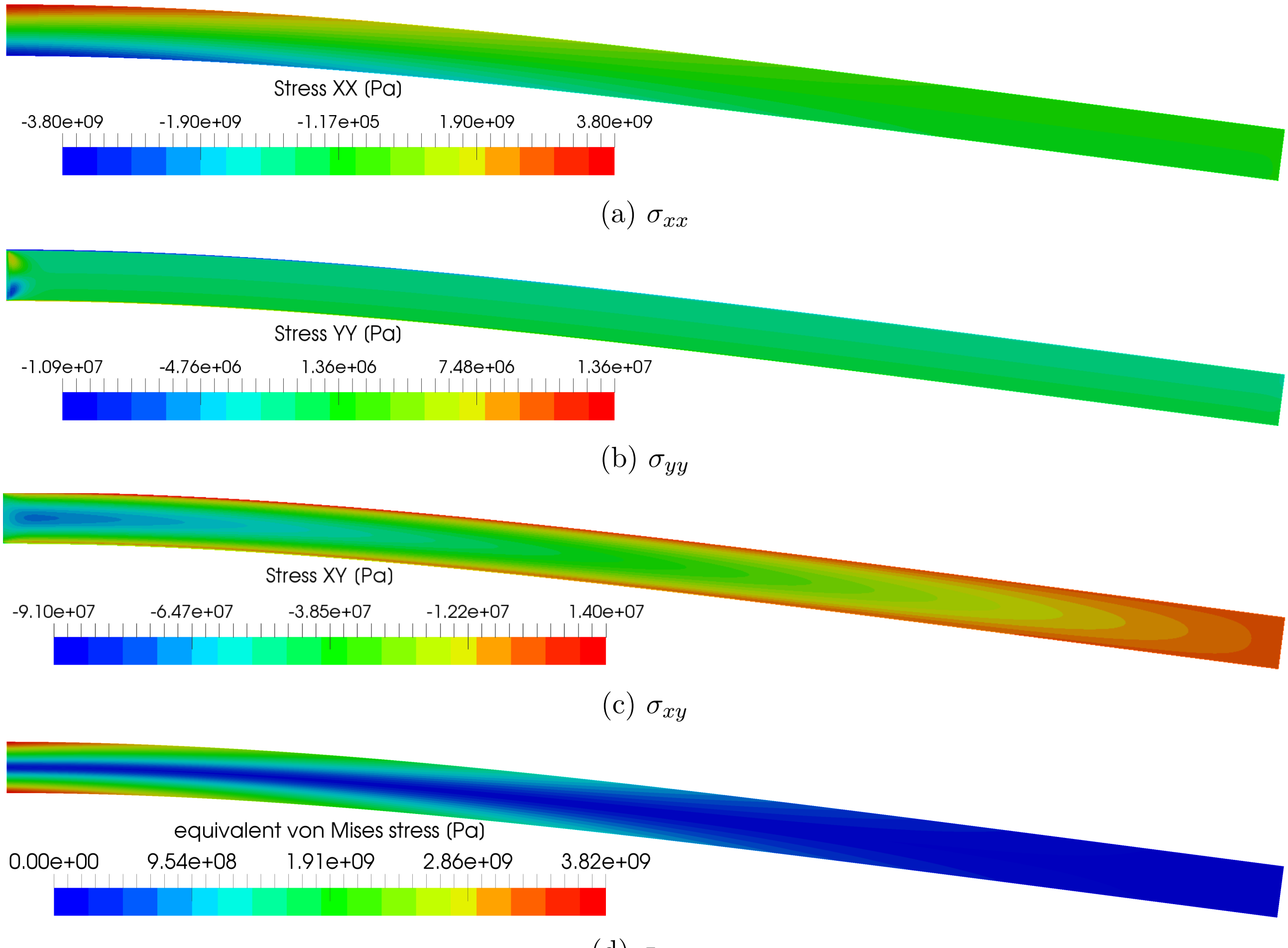
the shear stress σ_{xy} can be obtained as

$$\sigma_{xy} = \frac{F_{shear} Q_{rect}}{I_{yrect} b} \tag{3.57}$$

Due to the downward load, the highest magnitude shear stress is negative i.e. $\min(\sigma_{xy})$. It is evaluated as

$$\min(\sigma_{xy}) = \sigma_{xy}(x = 0, y = 0) = \frac{F_{shear} Q_{rect}}{I_{yrect} b} \tag{3.58}$$

The resulting distribution of σ_{xx}, σ_{yy} and σ_{xy} of the finest mesh is illustrated in fig. 3.42. For the finest mesh, the maximum and minimum values for σ_{xx}, σ_{yy} and σ_{xy} obtained are listed in table 3.30. The results show a good match



(a) σ_{xx}

(b) σ_{yy}

(c) σ_{xy}

(d) σ_{VM}

FIGURE 3.42: Stresses, smoothened. Note that deformation is exaggerated for a better visualization.

with the analytical solution based on beam theory, although stresses tend to be slightly too high.

TABLE 3.30: Results for static cantilever beam on mesh 5. Expected results according to eqs. 3.47 - 3.58.

Quantity	**Symbol**	**Obtained**	**Expected**	**Unit**
Mass	m	28000.0	28000.0	kg
Deflection neutral fiber	$\max(d_{nf})$	-0.97674	-0.97656	m
Normal Stress in x	$\min(\sigma_{xx})$	-3.800×10^9	-3.750×10^9	Pa
	$\max(\sigma_{xx})$	3.800×10^9	3.750×10^9	Pa
Normal Stress in y	$\min(\sigma_{yy})$	-3.873×10^7	–	Pa
	$\max(\sigma_{yy})$	3.968×10^7	–	Pa
Shear Stress	$\min(\sigma_{xy})$	-7.667×10^7	-7.500×10^7	Pa

3.4.6 Cantilever beam under distributed load - transient case

The same beam as before under the same load is examined for its transient behavior. The distributed load is applied instantaneously at full magnitude to the resting, straight beam (gravity is neglected). The beam deflects over time, but surpassing the equilibrium configuration from the previous section due to inertia. It then swings back and forth over time at a certain frequency and with the amplitude diminishing due to damping.

The same material properties and dimensions as in the previous case are used, except that the time step is varied ($\Delta t \in \{1, 2, 4, 8, 16, 32, 64\}$ ms) and the same total simulation time $t_{end} = 8.0$ s. A long simulation time is needed to have many sample points to get a sufficient frequency resolution from the Discrete Fourier Transform (DFT) used to determine the resulting frequency. The DFT is able to detect frequencies up to the maximum frequency $\max(f_{DFT}) = 1/(2\Delta t)$ with a frequency resolution $\Delta f_{DFT} = 1/t_{end}$. There should be no damping observed for a beam that is elastic, since there is no physical dissipation of kinetic energy. Plotting the y-displacement d_{nf} of the free end of the neutral fiber over time, one can clearly observe a dominant frequency, but also numerical damping. The numerical damping is most likely caused by the first order accurate backward Euler time integration scheme. The example in fig. 3.43 shows the second finest mesh with a small time step ($\Delta t = 0.001$ s) over a total duration $t_{end} = 8.0$ s.

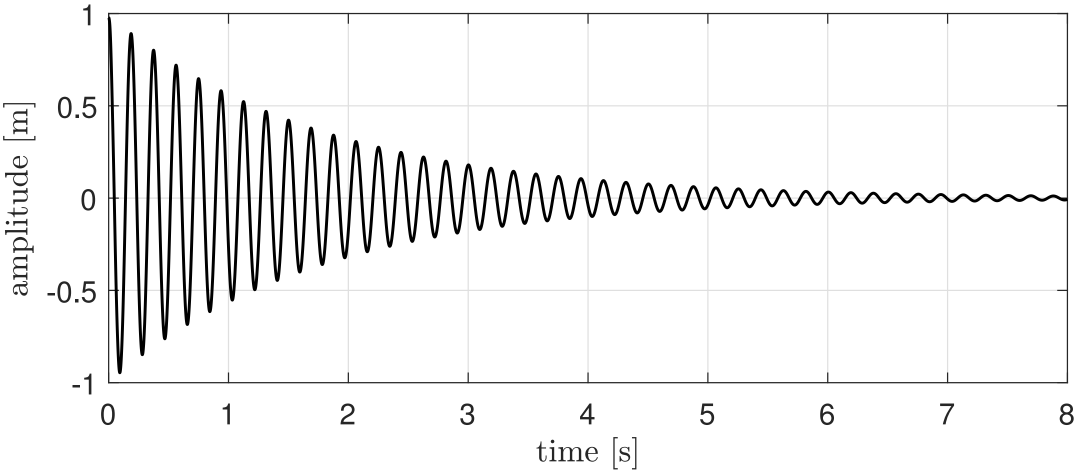


FIGURE 3.43: Oscillation of a cantilever beam after instantaneous loading. This result is obtained with mesh 4 (fig. 3.39d) and a small time step ($\Delta t = 0.001$ s).

An analytical solution for all the natural frequencies of a thin beam exists, but this load case will only excite the first natural frequency f_1. It is computed as

$$f_1 = \frac{1.875^2}{2\pi L^2}\sqrt{\frac{EI_{rect}}{\rho A}} = 5.3501\,\text{Hz} \tag{3.59}$$

where $A = bh$ is the cross section of the beam. The factor 1.875 is obtained from the solution of the Fourier decomposed Dynamic Beam Equation (DBE). The

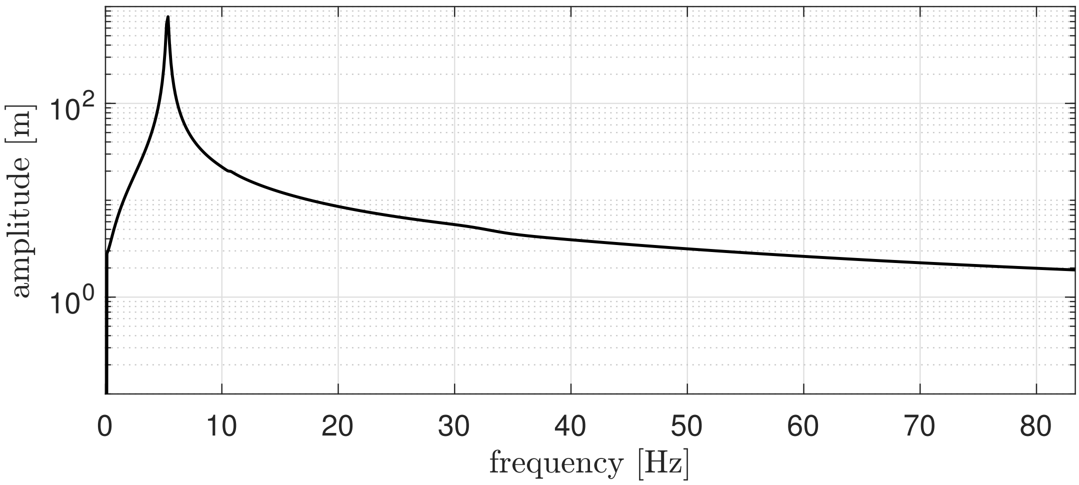


FIGURE 3.44: Frequency spectrum of a cantilever beam after sudden loading. This graph is derived from the time signal in fig. 3.43. The peak is located at $f_{PFEM} = 5.3743\,\text{Hz}$.

result of such a DFT performed on the time signal in fig. 3.43 using *Matlab*'s FTT algorithm can be found in fig. 3.44. The resulting frequency of vibration for the different meshes is summarized in fig. 3.45.

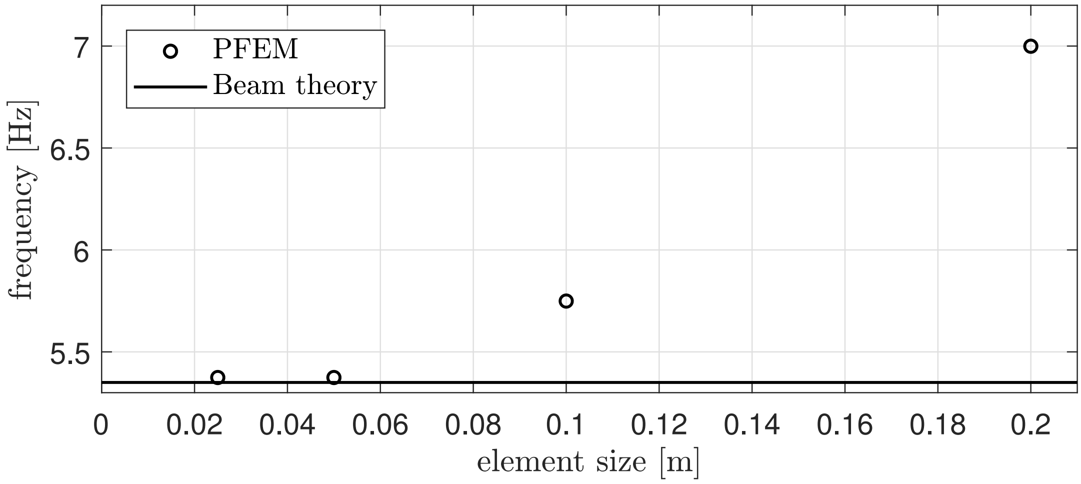


FIGURE 3.45: Frequency of oscillation for different mesh sizes with a time step of $\Delta t = 0.001\,\text{s}$.

With a sufficiently refined mesh, the correct frequency is recovered, with only a small, acceptable difference. Using the intermediate mesh 3 with an edge length $\bar{l}_e = 0.05\,\text{m}$, the time step is varied and the frequencies in fig. 3.46 are obtained. The correct frequency according to eq. 3.59 is sufficiently accurately recovered for time steps $\Delta t \leq 8\,\text{ms}$.

The damping can be quantified in several ways. One of the most intuitive options is the use of the logarithmic decrement δ. It directly represents the

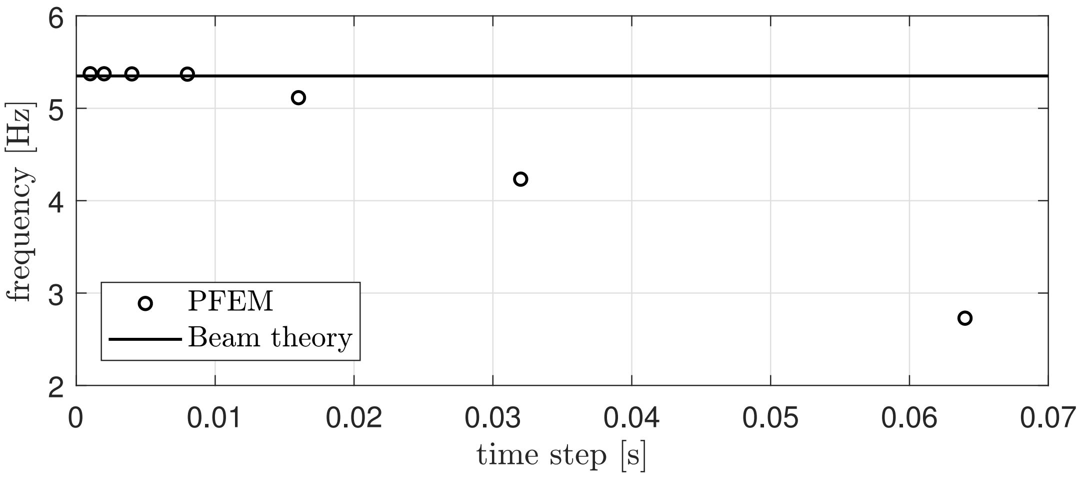


FIGURE 3.46: Frequency of oscillation for different time step sizes on mesh 3.

logarithmic decay of the amplitude that is evident in the plots such as the one above (fig. 3.43). It is calculated with eq. 3.60

$$\delta = \ln\left(\frac{d_{max_i}}{d_{max_{i-1}}}\right) \tag{3.60}$$

where $d_{max_{i-1}}$ and d_{max_i} are two consecutive maxima in the plot of the beam's tip displacement $d_{nf}(x = l)$ over time t. Two consecutive minima can be used equally well. We use the mean of δ of all the pairs of maxima and minima, since there is a slight variance. The damping is found to strongly depend on the time step size, but not much on the element size, as illustrated in figs. 3.47 and 3.48, respectively. This further confirms the suspicion that the time integration is introducing numerical damping. Either way, the elastic solid is shown to behave as expected and its verification is successful.

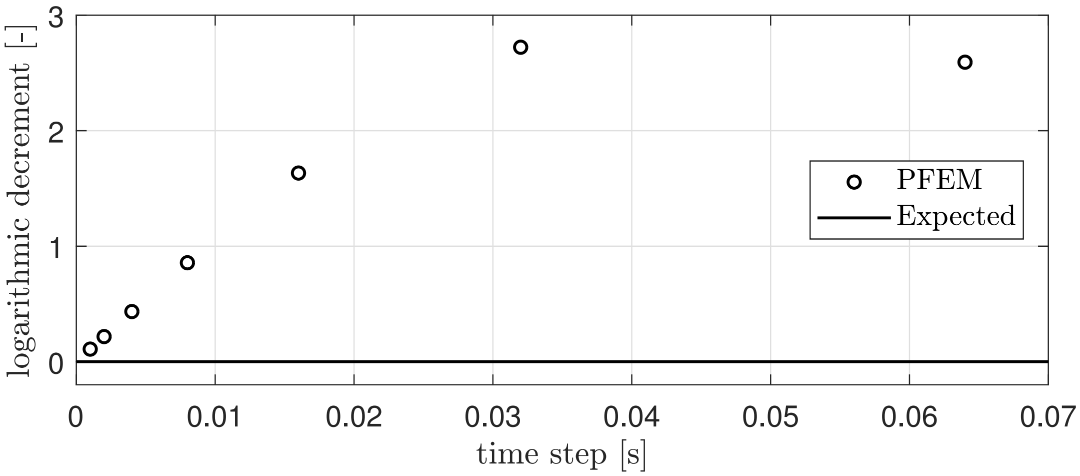


FIGURE 3.47: Damping (logarithmic decrement) of oscillation for different time step sizes on mesh 3.

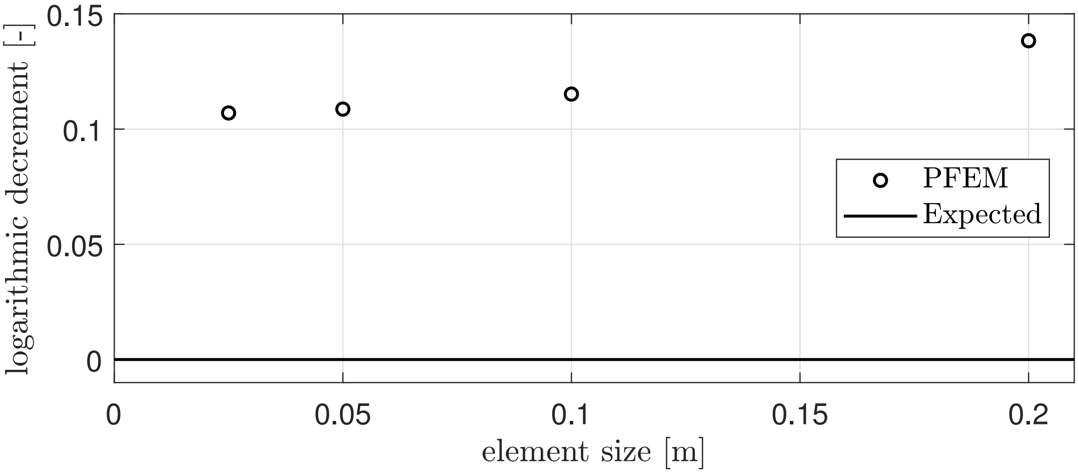


FIGURE 3.48: Damping (logarithmic decrement) of oscillation for different mesh sizes with a time step of $\Delta t = 0.001\,\mathrm{s}$.

3.5 Fluid-solid

Simulations where fluids and solids interact are commonly referred to as Fluid-Structure interaction (FSI) problems. It is well-known that in many cases, such simulations are carried out by coupling a dedicated solid solver with a dedicated fluid solver, each of which may solve the same equations using different formulations. In this work, a method is presented that does not require a coupling. Instead, one solver is used where the simulation domain contains fluids and solids simultaneously thanks to the unified fluid-solid formulation for the material's constitutive law.

When fluids and solids coexist in the same domain, an interface can be introduced at the contact surface between the two phases to allow for contact forces to act between the two without smearing out the sharp discontinuity of usually greatly different material properties. The location of this interface is detected automatically and set up as described in section 2.7.5. It is also possible to run a multiphase simulation without an interface, which means that the behavior at the contact surface is an undefined mix of fluid and solid.

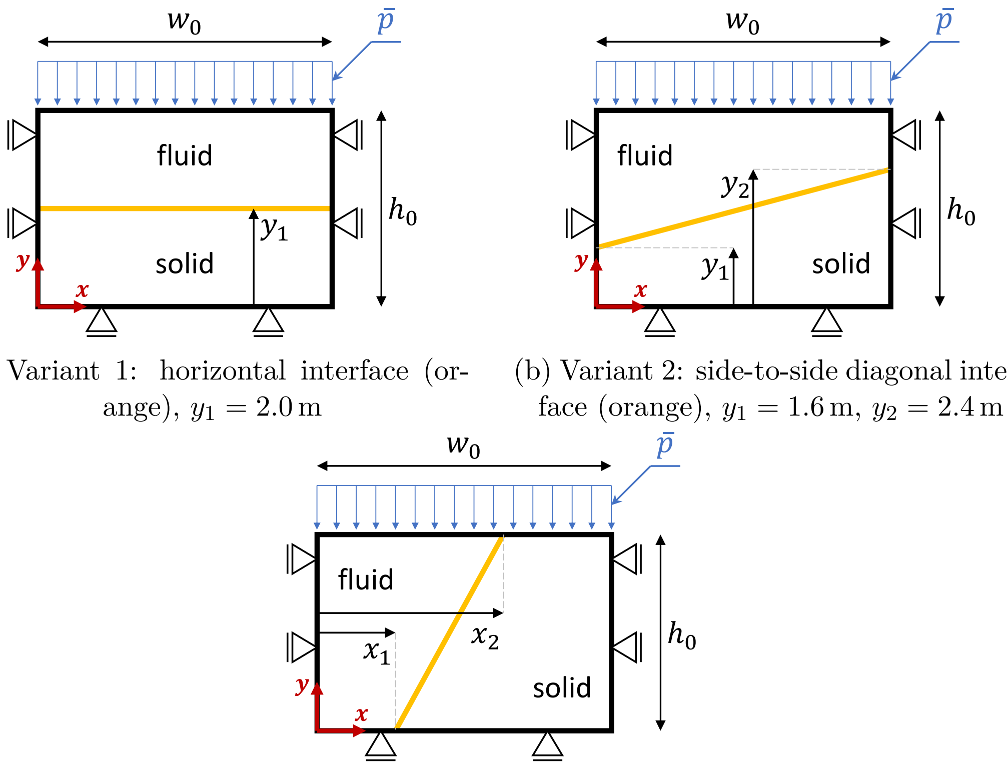
3.5.1 Fluid and solid under external pressure

A rectangular domain contains an elastic solid and a Newtonian fluid region and is subjected to an external pressure $\bar{p}$ acting on the top surface, but no gravity. The side walls are assumed to be free-slip (or sliding) in the y-direction and the bottom wall is free-slip (or sliding) in the x-direction. This test is supposed to demonstrate consistent material behavior with an interface present. For this reason, the split of the domain into fluid and solid is done in several different variations:

1. horizontal split
2. side-to-side diagonal split

3. top-to-bottom diagonal split

The variants are visualized schematically in fig. 3.49.



(a) Variant 1: horizontal interface (orange), $y_1 = 2.0\,\mathrm{m}$

(b) Variant 2: side-to-side diagonal interface (orange), $y_1 = 1.6\,\mathrm{m}$, $y_2 = 2.4\,\mathrm{m}$

(c) Variant 3: top-to-bottom diagonal interface (orange), $x_1 = 2.5\,\mathrm{m}$, $x_2 = 3.5\,\mathrm{m}$

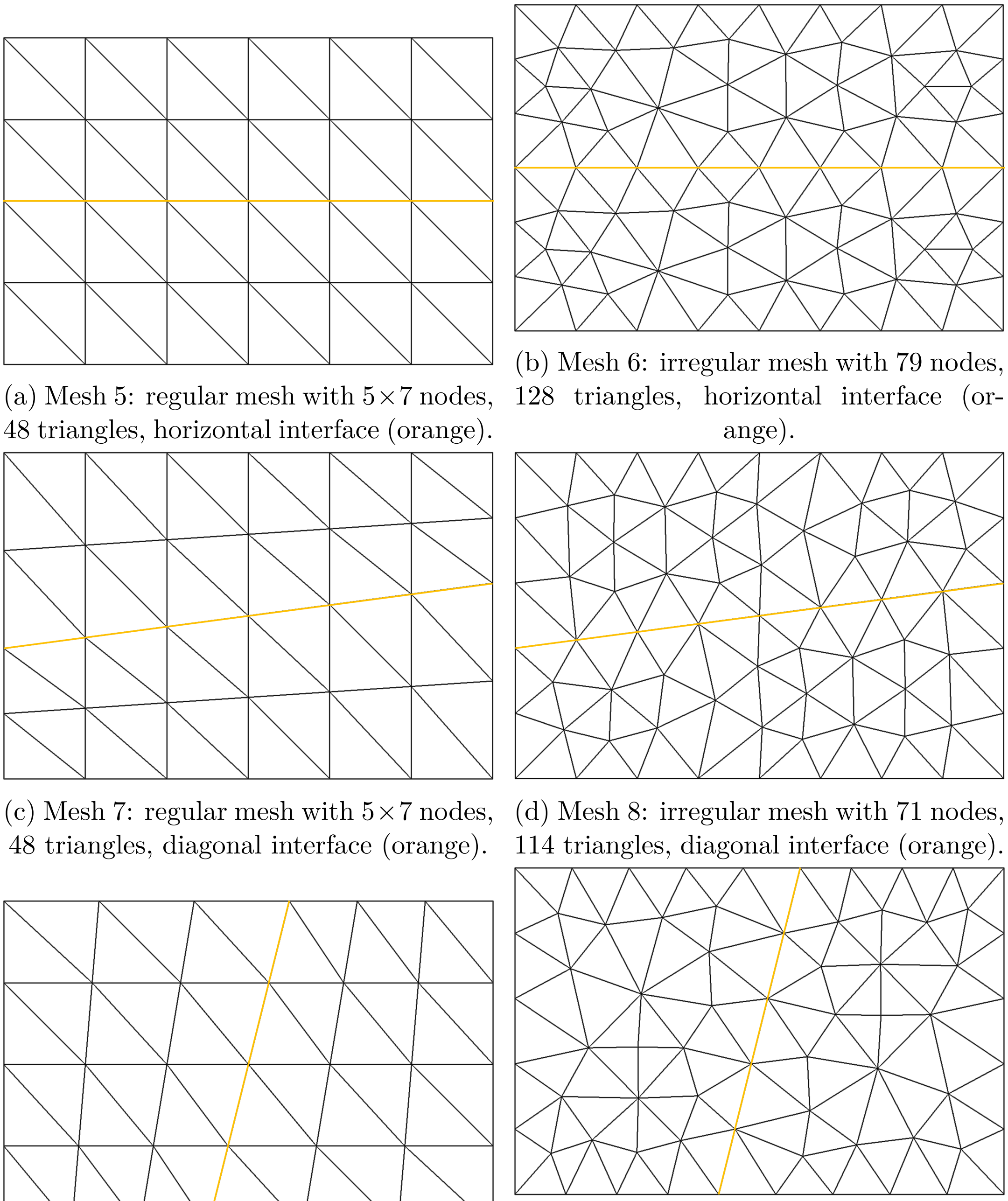
FIGURE 3.49: Schematic of the 3 variants, $h_0 = 4.0\,\mathrm{m}$, $w_0 = 6.0\,\mathrm{m}$ in all cases.

In the first variant, the principal stresses that occur due to the external load are aligned with the interface, which is not the case for the other two variants. The third variant features the loaded boundary intersected by the interface. This demonstrates that the interface works as desired in situations that occur commonly in the advanced tests of the following chapter.

The six meshes that are used to reflect these variants are given in fig. 3.50. The numbering is chosen such that it avoids any confusion with the four meshes in fig. 3.32 of section 3.4.

The fluid is assumed to be incompressible and its material properties are given in table 3.31. The solid's material properties are given in table 3.32.

The external pressure of magnitude $\bar{p} = 100.0\,\mathrm{Pa}$ is applied as a normal surface traction $\bar{\boldsymbol{t}}_n = -\bar{p}\boldsymbol{n}$ on the free surface at the top, where $\boldsymbol{n}$ is the unit outward normal vector on the boundary and using the fluid dynamics convention for the sign of the pressure. The magnitude is low enough to not cause any visible displacement in the elastic solid and no flow in the fluid. Inertia shall be neglected for simplicity.



(a) Mesh 5: regular mesh with 5×7 nodes, 48 triangles, horizontal interface (orange).

(b) Mesh 6: irregular mesh with 79 nodes, 128 triangles, horizontal interface (orange).

(c) Mesh 7: regular mesh with 5×7 nodes, 48 triangles, diagonal interface (orange).

(d) Mesh 8: irregular mesh with 71 nodes, 114 triangles, diagonal interface (orange).

(e) Mesh 9: regular mesh with 5×7 nodes, 48 triangles, diagonal interface (orange).

(f) Mesh 10: irregular mesh with 68 nodes, 106 triangles, diagonal interface (orange).

Figure 3.50: Meshes used for FSI problems.

For all variants, the incompressible fluid at rest exhibits the following strains

$$\varepsilon_{xx} = \varepsilon_{yy} = \varepsilon_{zz} = \varepsilon_{xy} = 0 \tag{3.61}$$

and therefore the stresses under the external pressure $\bar{p}$:

$$\left.\begin{aligned} \sigma_{xx} = \sigma_{yy} = \sigma_{zz} &= -\bar{p} \\ \sigma_{xy} &= 0 \end{aligned}\right\} \quad \forall \boldsymbol{x} \text{ in the fluid} \tag{3.62}$$

TABLE 3.31: Material properties of the fluid.

Property	Symbol	Value	Unit
Viscosity	μ	0.006	Pa s
Density	ρ	1000.0	kg m^{-3}
Bulk modulus	K	$\to \infty$	

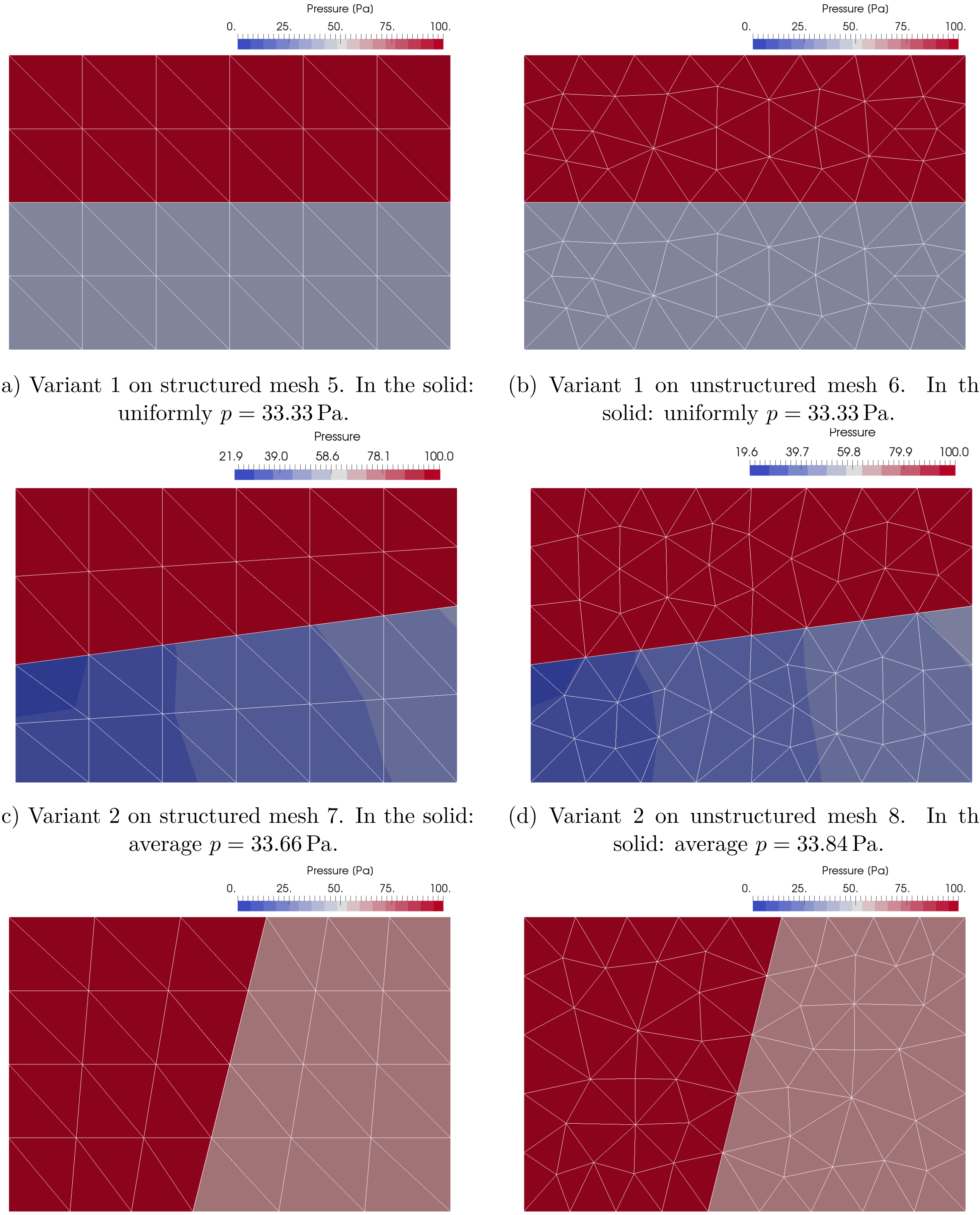
TABLE 3.32: Material properties of the solid.

Property	Symbol	Value	Unit
Density	ρ	150.0	kg m^{-3}
Poisson's ratio	ν	0.0	
Young's modulus	E	6.0×10^{10}	Pa
Shear modulus	G	3.0×10^{10}	Pa
Bulk modulus	K	2.0×10^{10}	Pa

The stresses and strains of the solid, on the other hand, differ between the variants and vary in space and are thus difficult to predict exactly. A better way to make a quantified and exact prediction about the material behavior is to examine the forces at the boundaries and interfaces, which are shown further down. Regarding the stresses, it is expected that the solid exhibits a normal stress in the y-direction $\sigma_{yy} \approx -\bar{p}$, to support the external load for all variants. In the absence of a Poisson effect, the normal stress in the x-direction $\sigma_{xx} = 0$ in variant 1. For variant 2, however, $0 < \sigma_{xx} < -\bar{p}$ due to the angle of the interface. For variant 3 $\sigma_{xx} = -p$ because the fluid is exerting a pressure on the solid region across the entire height h_0. The in-plane shear stress $\sigma_{xy} \approx 0$ in all cases. Assuming plane strain and no Poisson effect, we expect $\sigma_{zz} = 0$ for all cases. Based on the expected stresses, a prediction for the pressure can be made with $p = -1/3\,\mathrm{tr}(\boldsymbol{\sigma})$ (eq. 2.52).

The simulations are carried out for all meshes 5 to 10 using a single time step of $\Delta t = 1.0\,\mathrm{s}$. The interface can be fixed by the user before the simulation run or it can be detected automatically during run-time; the first option is chosen here. Following the explanation in section 2.5.5, the nodal pressure is imposed where the fluid's free surface meets the side wall and where it meets the interface, while no nodal pressure is imposed on the solid. The distributed load by the external pressure is applied to the entire free surface, i.e. solid and fluid portion. The stabilization of section 2.5.4 is applied on the incompressible fluid, but not on the compressible solid.

The contour plots of the pressure are given in fig. 3.51 and of the stresses σ_{xx}, σ_{yy}, σ_{zz} and σ_{xy} are given in figs. 3.52 - 3.54.



(a) Variant 1 on structured mesh 5. In the solid: uniformly $p = 33.33\,\text{Pa}$.

(b) Variant 1 on unstructured mesh 6. In the solid: uniformly $p = 33.33\,\text{Pa}$.

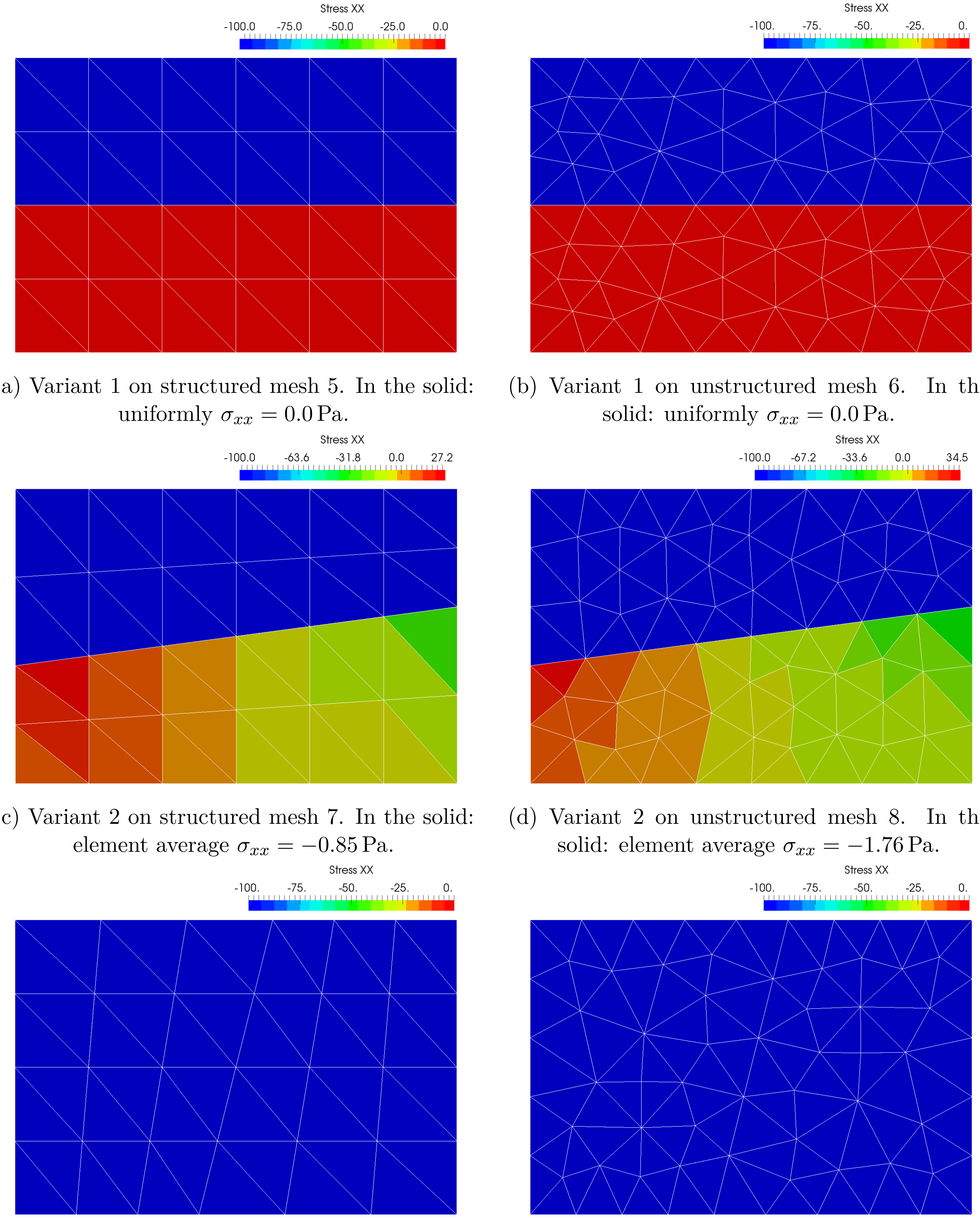
(c) Variant 2 on structured mesh 7. In the solid: average $p = 33.66\,\text{Pa}$.

(d) Variant 2 on unstructured mesh 8. In the solid: average $p = 33.84\,\text{Pa}$.

(e) Variant 3 on structured mesh 9. In the solid: uniformly $p = 66.67\,\text{Pa}$.

(f) Variant 3 on unstructured mesh 10. In the solid: uniformly $p = 66.67\,\text{Pa}$.

FIGURE 3.51: Pressure distribution. In the fluid: uniformly $p = 100\,\text{Pa}$ in all cases.



(a) Variant 1 on structured mesh 5. In the solid: uniformly $\sigma_{xx} = 0.0\,\text{Pa}$.

(b) Variant 1 on unstructured mesh 6. In the solid: uniformly $\sigma_{xx} = 0.0\,\text{Pa}$.

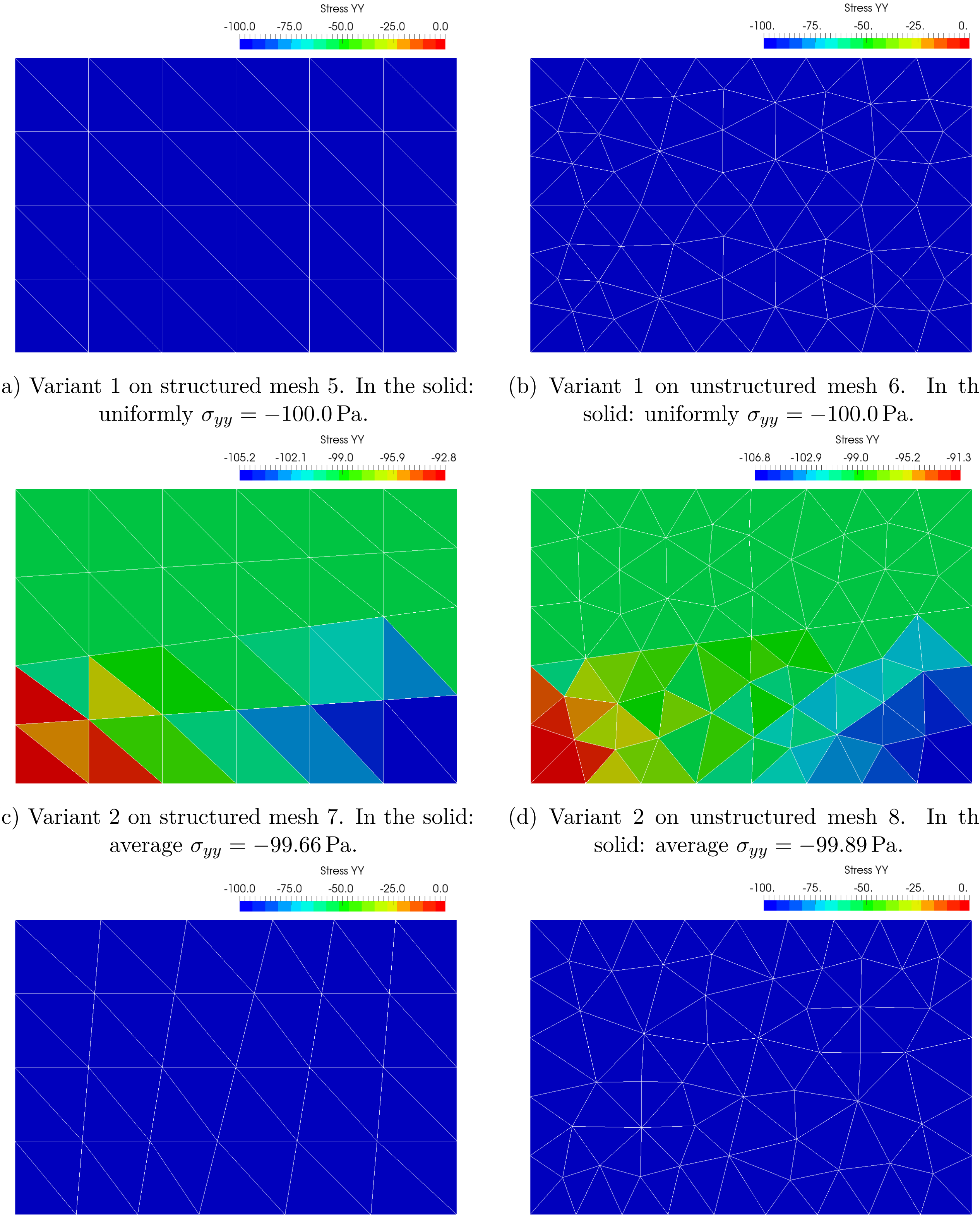
(c) Variant 2 on structured mesh 7. In the solid: element average $\sigma_{xx} = -0.85\,\text{Pa}$.

(d) Variant 2 on unstructured mesh 8. In the solid: element average $\sigma_{xx} = -1.76\,\text{Pa}$.

(e) Variant 3 on structured mesh 9. In the solid: uniformly $\sigma_{xx} = -100.0\,\text{Pa}$.

(f) Variant 3 on unstructured mesh 10. In the solid: uniformly $\sigma_{xx} = -100.0\,\text{Pa}$.

FIGURE 3.52: Distribution of σ_{xx}. In the fluid: uniformly $\sigma_{xx} = 100\,\text{Pa}$ in all cases.



(a) Variant 1 on structured mesh 5. In the solid: uniformly $\sigma_{yy} = -100.0\,\text{Pa}$.

(b) Variant 1 on unstructured mesh 6. In the solid: uniformly $\sigma_{yy} = -100.0\,\text{Pa}$.

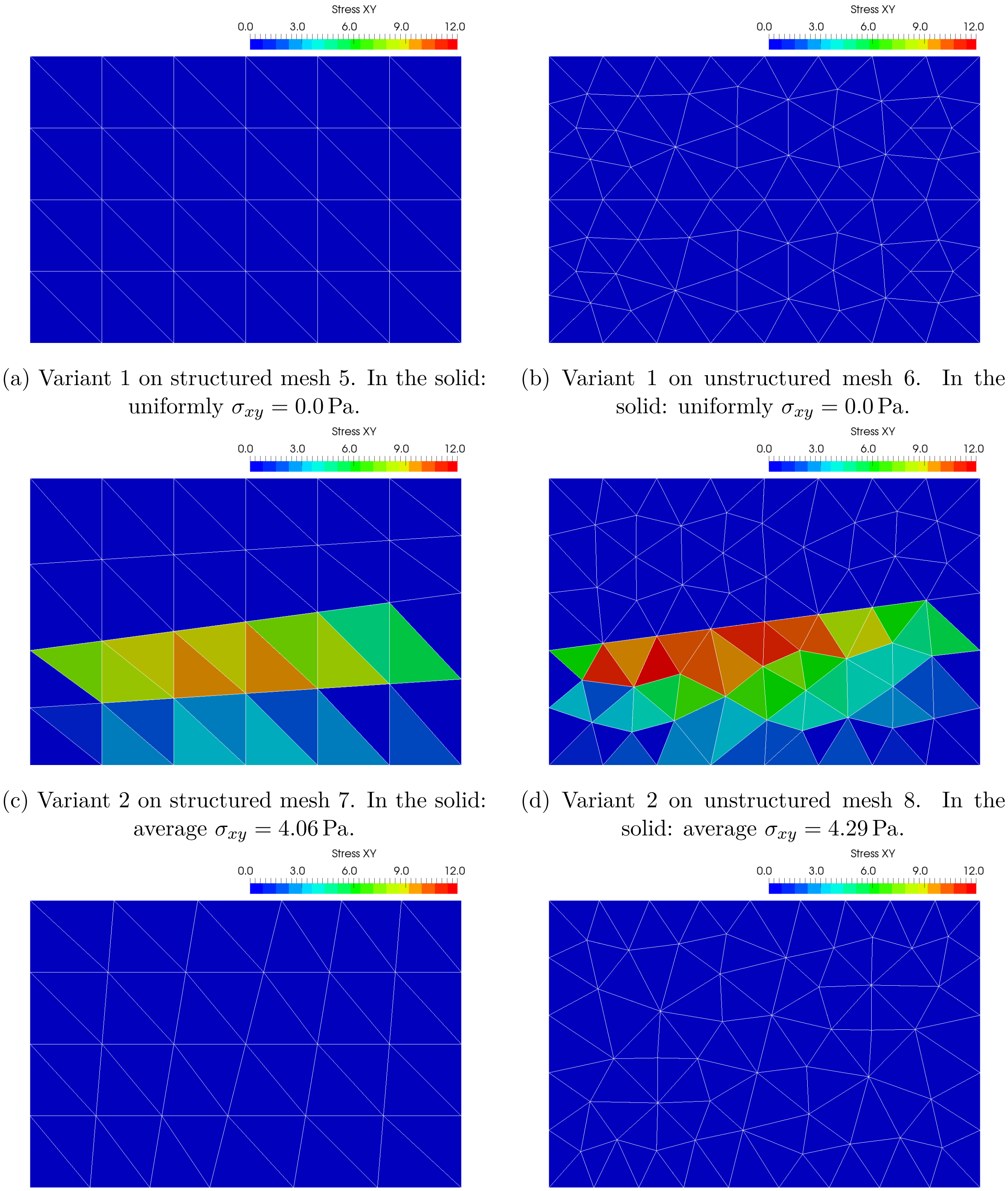
(c) Variant 2 on structured mesh 7. In the solid: average $\sigma_{yy} = -99.66\,\text{Pa}$.

(d) Variant 2 on unstructured mesh 8. In the solid: average $\sigma_{yy} = -99.89\,\text{Pa}$.

(e) Variant 3 on structured mesh 9. In the solid: uniformly $\sigma_{yy} = -100.0\,\text{Pa}$.

(f) Variant 3 on unstructured mesh 10. In the solid: uniformly $\sigma_{yy} = -100.0\,\text{Pa}$.

FIGURE 3.53: Distribution of σ_{yy}. In the fluid: uniformly $\sigma_{yy} = 100\,\text{Pa}$ in all cases.



(a) Variant 1 on structured mesh 5. In the solid: uniformly $\sigma_{xy} = 0.0\,\mathrm{Pa}$.

(b) Variant 1 on unstructured mesh 6. In the solid: uniformly $\sigma_{xy} = 0.0\,\mathrm{Pa}$.

(c) Variant 2 on structured mesh 7. In the solid: average $\sigma_{xy} = 4.06\,\mathrm{Pa}$.

(d) Variant 2 on unstructured mesh 8. In the solid: average $\sigma_{xy} = 4.29\,\mathrm{Pa}$.

(e) Variant 3 on structured mesh 9. In the solid: uniformly $\sigma_{xy} = 0.0\,\mathrm{Pa}$.

(f) Variant 3 on unstructured mesh 10. In the solid: uniformly $\sigma_{xy} = 0.0\,\mathrm{Pa}$.

FIGURE 3.54: Distribution of σ_{xy}. In the fluid: uniformly $\sigma_{xy} = 0.0\,\mathrm{Pa}$ in all cases.

Overall, the expected behavior is observed:

- The incompressible fluid has a uniform pressure.
- All normal stresses in the fluid equal the externally applied pressure with the reversed sign due to the sign convention for the pressure.
- The pressure and stress components in the solid are not generally the same as in the fluid.
- The change in pressure and stress components is sharp when crossing the interface.
- Since the external pressure is acting in the y-direction only, the materials exhibit a normal stress in the y-direction $\sigma_{yy} \approx -p$, no matter how fluid and solid are arranged.
- The solid may exhibit a normal stress in the x-direction $\sigma_{xx} \neq 0$, if the interface is not orthogonal to the y-axis.
- The stresses in the solid may be non-uniform. This depends on the choice of boundary conditions and the orientation of the interface, assuming that the fluid has a uniform pressure.

For variant 3 a uniform $\sigma_{xx} = -p$ is observed because the pressurized fluid is acting on the solid over the entire height h_0. Variant 2 shows a non-uniform stress distribution because the fluid is pushing the solid in the positive x-direction, while the solid is "attached" to the left and right boundaries that restrict the movement in the x-direction. Since the solid is not equally free to deform in the x-direction, the non-uniform stresses are observed. The same effect induces the small but non-zero shear stresses in the solid near the interface because the interface nodes are moving in the positive x-direction more than the nodes at the bottom boundary.

In any static case like this one, an external load must be supported by the boundaries through the material. This means that the sum of all external and boundary forces must add up to 0. This can be expressed in the integral form as

$$\underbrace{\int\limits_{\Gamma_t} \bar{\boldsymbol{t}}_n \, dA}_{F_{\Gamma_t}} + \underbrace{\int\limits_{\Gamma_v} \boldsymbol{\sigma} \cdot \boldsymbol{n} \, dA}_{F_{\Gamma_v}} + \underbrace{\int\limits_{\Gamma_\lambda} \boldsymbol{\lambda} \, dA}_{F_{\Gamma_\lambda}} = \boldsymbol{0} \tag{3.63}$$

where Γ_t is the boundary at which the traction is applied (top surface), Γ_v is the boundary at which the velocities are imposed, Γ_λ is the interface, $\boldsymbol{\lambda}$ is the force at the interface and $\boldsymbol{n}$ is the unit outward normal vector on a boundary. Note that the interface should be viewed as two coincident surfaces, each attached to either solid or fluid region and with their respective outward normal vectors pointing in opposite directions. Viewing the entire domain, the equal and opposite interface forces cancel out, but must be taken into account when regarding the fluid or solid region individually. The terms of the equation shall be called F_{Γ_t}, F_{Γ_v} and F_{Γ_λ} for the sum of Neumann boundary forces, sum of Dirichlet boundary forces and sum of interface forces, respectively. In the discretized system of equations, all terms of the equations can be expressed as the sums of nodal forces over the respective boundaries:

$$\underbrace{\sum_{I=1}^{n_{\Gamma_t}} \boldsymbol{F}_{ext_I}}_{F_{\Gamma_t}} + \underbrace{\sum_{J=1}^{n_{\Gamma_v}} \boldsymbol{F}_{int_J}}_{F_{\Gamma_v}} + \underbrace{\sum_{K=1}^{n_{\Gamma_\lambda}} \boldsymbol{\lambda}_K}_{F_{\Gamma_\lambda}} = \mathbf{0} \tag{3.64}$$

where $\boldsymbol{F}_{ext_I}$ is the external nodal force vector at node I located on the boundary Γ_t, which has a total of n_{Γ_t} nodes. The components of $\boldsymbol{F}_{ext_I}$ are found in $\boldsymbol{F}_{ext}$ of eq. 2.173, where no body forces $\boldsymbol{b}$ are imposed in this present case. Analogously, $\boldsymbol{F}_{int_J}$ is the internal nodal force vector of node J located on boundary Γ_v, which has a total of n_{Γ_v} nodes. The components of $\boldsymbol{F}_{int_J}$ are found in $\boldsymbol{F}_{int}$ of eq. 2.172. $\boldsymbol{\lambda_K}$ is the interface nodal force vector of node K located on boundary Γ_λ, which has a total of n_{Γ_λ} nodes. The components of $\boldsymbol{\lambda}_K$ are found in $\boldsymbol{\lambda}$ of eq. 2.165. These nodal forces are readily available at no additional cost.

The sum of external forces at the Neumann boundaries and the interface are given in tables 3.33 and 3.34 and are compared to the sum of internal forces at the boundaries in the x and y-direction respectively. The table only shows the solid region's forces.

Table 3.33: Sum of external forces of the Neumann boundaries and the interface compared to the sum of internal forces in the x-direction in N in the solid region.

Variant (Mesh)	**Neumann boundary contribution**	**Interface contribution**	**Sum** F_{ext_x}	**Sum** F_{int_x}
1(5)	0.0	0.0	0.0	0.0
1(6)	0.0	0.0	0.0	0.0
2(7)	0.0	80.0	80.0	80.0
2(8)	0.0	80.0	80.0	80.0
3(9)	0.0	400.0	400.0	400.0
3(10)	0.0	400.0	400.0	400.0

For all values that are supposed to be equal to zero, an error of the order of 10^{-14} was found. In all cases, the external forces (including interface force) match the internal forces exactly, which means that the interface applies the correct forces onto the solid, no matter the orientation of the interface. This confirms that the stresses and pressure obtained with this method are accurate for different configurations of solid and liquid.

TABLE 3.34: Sum of external forces of the Neumann boundaries and the interface compared to the sum of internal forces in the y-direction in N in the solid region.

Variant (Mesh)	Neumann boundary contribution	Interface contribution	Sum F_{ext_y}	Sum F_{int_y}
1(5)	0.0	600.0	600.0	600.0
1(6)	0.0	600.0	600.0	600.0
2(7)	0.0	600.0	600.0	600.0
2(8)	0.0	600.0	600.0	600.0
3(9)	250.0	100.0	350.0	350.0
3(10)	250.0	100.0	350.0	350.0

3.5.2 Shearing of a fluid on a solid

A layer of an elastic solid is at the bottom of an initially rectangular domain and its bottom boundary is fixed. A layer of a Newtonian fluid is ontop of the solid and a constant shearing velocity $\bar{v}_x$ is applied to the fluid's top surface, which causes a non-zero shear strain rate and therefore a non-zero shear stress. This shear stress acts on the interface and begins deforming the solid, shearing it until both the solid's and fluid's shear stress are equal and the system reaches equilibrium. At equilibrium, the solid shear strain and the fluid's shear strain rate become constant and a steady state solution is obtained. No pressure is imposed and no gravity is acting.

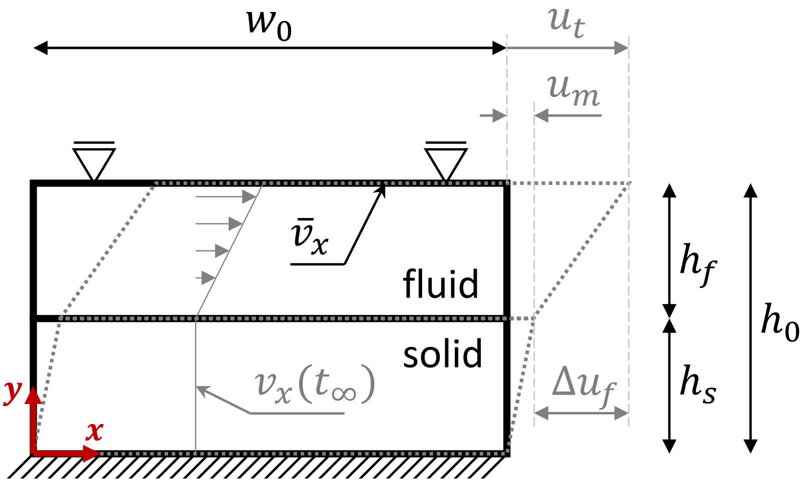


FIGURE 3.55: Schematic of a fluid layer (top) on a solid layer (bottom) being sheared by an imposed velocity $\bar{v}_x$ at the top surface. Equilibrium configuration illustrated in grey dotted lines with the velocity profile at equilibrium $v_x(t_\infty)$, where the fluid keeps shearing at a constant rate at equilibrium, while the solid is sheared but at rest.

On the geometry in fig. 3.55 of height $h_f = h_s = 0.35\,\mathrm{m}$ and $w_0 = 2.0\,\mathrm{m}$, an imposed velocity in the x-direction $\bar{v}_x = 1.0\,\mathrm{m\,s^{-1}}$ is applied at the top boundary and an imposed velocity in the y-direction $\bar{v}_y = 0.0\,\mathrm{m\,s^{-1}}$ is imposed on all nodes[1]. This makes this test similar to the simple shear test of section 3.4.4. The materials used for the fluid and the solid are the ones from the previous section (tables 3.31 and 3.32), except that the fluid's viscosity has been increased by a factor of 10^9 (new value: $\mu_f = 6.0 \times 10^6\,\mathrm{Pa\,s}$) to provoke a more noticeable shear in the solid.

At any point in time, the shear strain rate $\dot{\varepsilon}_{xy}$ of the fluid layer can be computed based on eq. 2.58:

$$\dot{\varepsilon}_{xy} = \frac{1}{2}\left(\underbrace{\frac{\partial v_y}{\partial x}}_{=0} + \frac{\partial v_x}{\partial y}\right) = \frac{(v_{t_x} - v_{m_x})}{2h_f} \quad \forall \boldsymbol{x} \text{ in the fluid} \tag{3.65}$$

where $v_{t_x} = \bar{v}_x$ is the imposed x-velocity of the top boundary of the fluid, v_{m_x} is the x-velocity of the interface in the middle and h_f is the height of the fluid layer on top at any point in time t. Assuming that the solid has reached the constant sheared configuration under the constant flow of the fluid on top, the top surface of the solid (i.e. the interface) is at rest. This means that the fluid's shear strain rate becomes

$$\dot{\varepsilon}_{xy} = \frac{\bar{v}_x}{2h_f} \quad \forall \boldsymbol{x} \text{ in the fluid} \tag{3.66}$$

As a consequence a shear stress σ_{xy} occurs in the fluid according to eq. 2.69:

$$\sigma_{xy} = 2\mu_f \dot{\varepsilon}_{xy} = \mu_f \frac{\bar{v}_x}{h_f} \quad \forall \boldsymbol{x} \text{ in the fluid} \tag{3.67}$$

where μ_f is the fluid's viscosity. All other strain rate and stress components are assumed to be zero in the fluid. At the interface between fluid and solid, σ_{xy} must be equal due to a no-slip condition between the materials in the absence of any other stresses acting on the interface from either side. This means that when equilibrium is reached we have

$$\sigma_{xy} = 2\mu_f \frac{\bar{v}_x}{h_t} \quad \forall \boldsymbol{x} \tag{3.68}$$

In the solid layer, the shear stress has caused a deformation after reaching the steady state, which obeys eq. 2.66, which can be expanded further using eq. 2.56:

$$\sigma_{xy} = 2G\varepsilon_{xy} = G\left(\underbrace{\frac{\partial u_y}{\partial x}}_{=0} + \frac{\partial u_x}{\partial y}\right) \quad \forall \boldsymbol{x} \text{ in the solid} \tag{3.69}$$

where G is the shear modulus and $\boldsymbol{u}$ is the displacement. Since we already imposed that the displacement in the y-direction is zero and since the bottom boundary is fixed, we can rewrite eq. 3.69.

[1] Restricting the movement of nodes in the y-direction is necessary to get results that agree exactly with the analytical solution.

$$\sigma_{xy} = G\frac{u_{m_x}}{h_s} \quad \forall \boldsymbol{x} \text{ in the solid} \tag{3.70}$$

where h_s is the height of the solid bottom layer at any point in time t and u_{m_x} is the x-displacement of the interface. Inserting eq. 3.68 into eq. 3.70 and rearranging yields an expression for the x-displacement of the interface

$$u_{m_x} = \bar{v}_x \frac{\mu_f}{G}\frac{h_s}{h_f} \tag{3.71}$$

The simulation is carried out on the mesh in fig. 3.56 with a total duration $t_{tot} = 0.05\,\mathrm{s}$ and a time step $\Delta t = 0.1\,\mathrm{ms}$. Inertia is neglected and no external pressure and no gravity are applied.

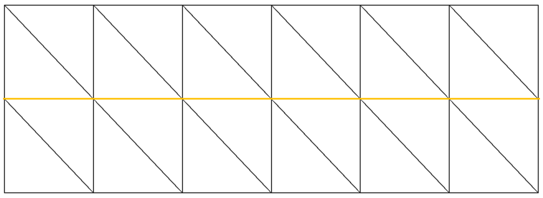


FIGURE 3.56: Mesh used with interface in orange.

First, fig. 3.57 shows the time evolution of the velocity and the displacement in the x-direction of the interface, u_{m_x} and v_{m_x}, respectively. The velocity v_{m_x} approaches zero at a logarithmic rate until reaching machine accuracy. The displacement u_{m_x} reaches the expected value and then remains constant.

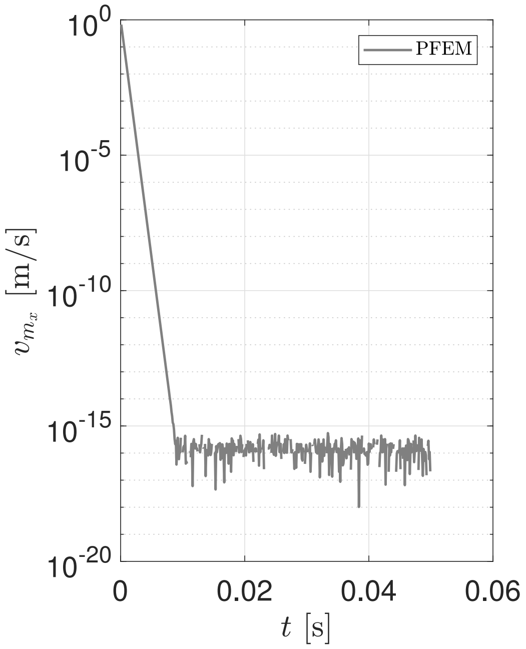
The shear stress σ_{xy} at the final time is plotted in fig. 3.58 with a narrow range of the magnitude around the expected value. Even in this narrow range, there appears to be a perfectly uniform shear stress throughout the fluid and the solid.

The final values are summarized in table 3.35. The comparison of expected results based on the analytical solution (eqs. 3.67 - 3.71) with the values obtained from the simulation shows that the agreement is excellent and the interface appears to handle shearing well.

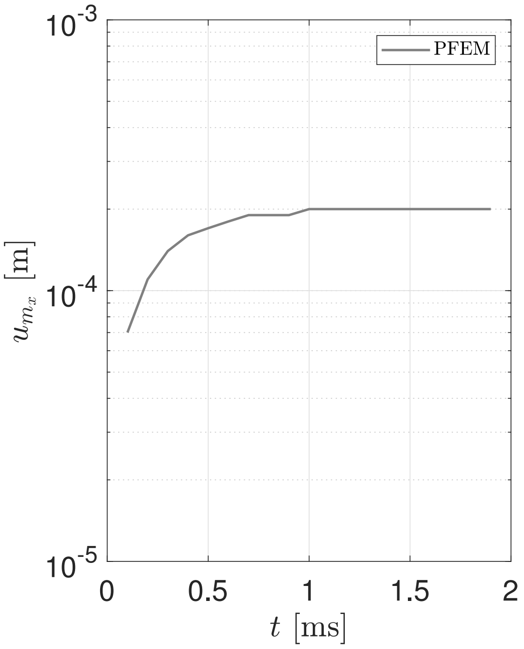
TABLE 3.35: Results after reaching equilibrium with relative or absolute errors indicated.

Symbol	Obtained value	Expected value	Error	Unit
v_{m_x}	$< 10^{-16}$	0.0	$< 10^{-15}$ (abs)	$\mathrm{m\,s^{-1}}$
u_{m_x}	0.2	0.2	$< 10^{-12}$ (rel)	m
σ_{xy} (fluid)	17.143×10^6	17.143×10^6	$< 10^{-14}$ (rel)	Pa
σ_{xy} (solid)	17.143×10^6	17.143×10^6	$< 10^{-12}$ (rel)	Pa

To summarize this chapter, many verification steps with new physics have been shown, where a good agreement with the expected result is obtained. The



(a) Mean velocity in the x-direction of the interface v_{m_x} over time (logarithmic y-axis).



(b) Mean displacement in the x-direction of the interface u_{m_x} over time (only first 20 time steps shown, logarithmic y-axis).

FIGURE 3.57: Interface reaching equilibrium under shear flow.

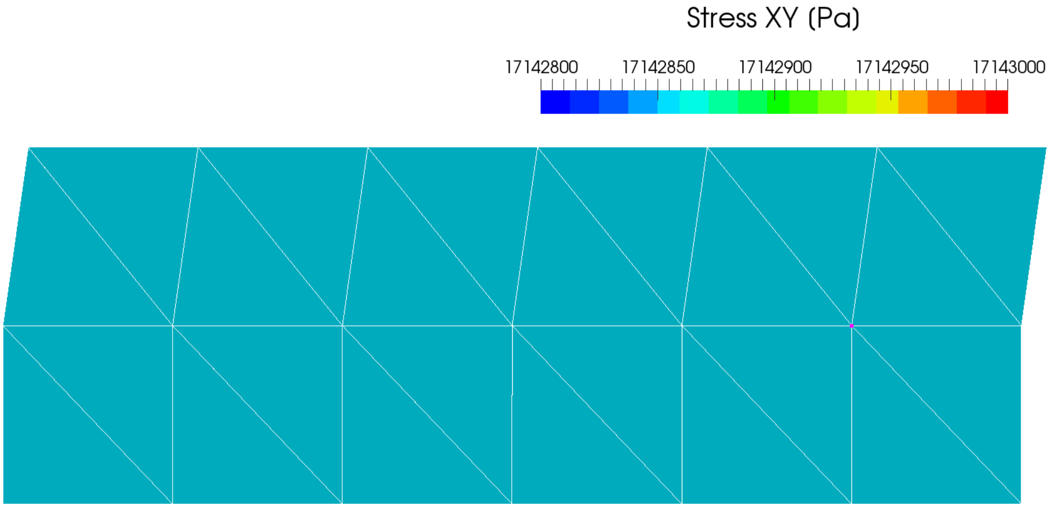


FIGURE 3.58: Shear stress $\sigma_{xy} = 17.143 \times 10^6$ Pa.

verification of the current capabilities, especially the unified formulation, can therefore be considered successful.

Chapter 4

Demonstration test cases for unified fluid-solid formulation

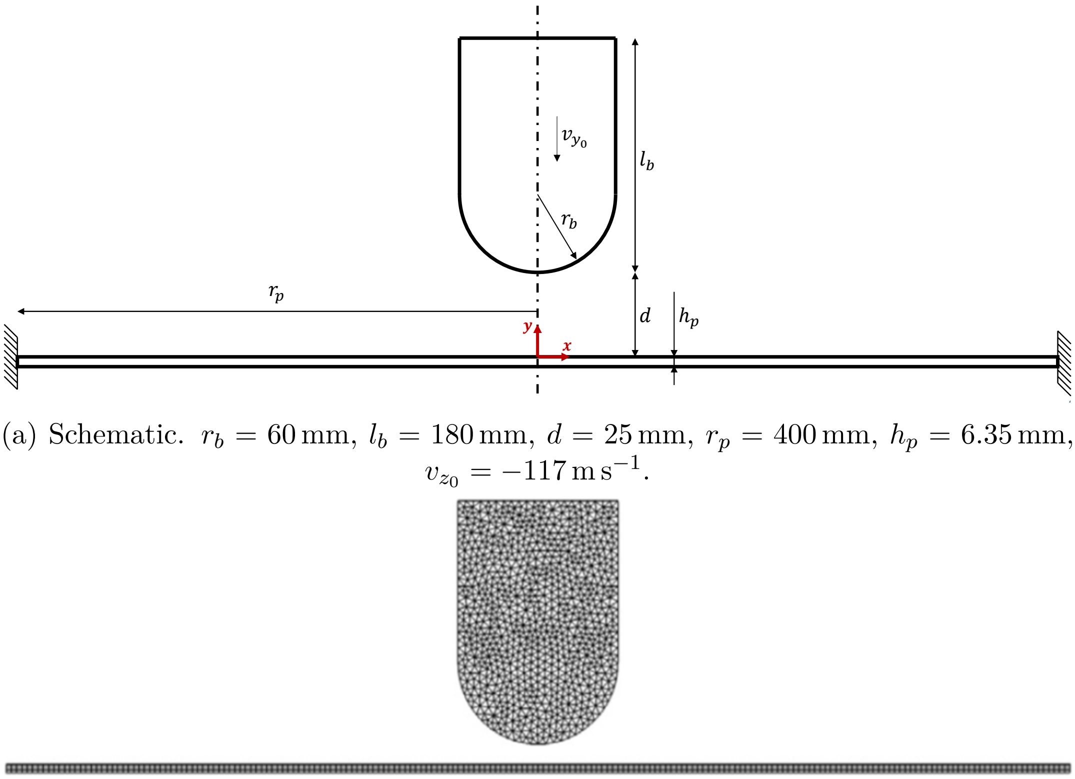
These test cases make use of the proposed unified fluid-solid formulation, which allows them to simulate the fluid and the solid simultaneously. As a result, the fluid flow can be correctly captured, as well as the stresses in the elastic solid. This would normally require a coupling algorithm that allows a specialized fluid solver and a specialized solid solver to communicate. The unified formulation allows to avoid the coupling whenever this is convenient or even necessary.

4.1 Bird strike

This test case is adapted from Cerquaglia et al. [9], who demonstrated the FSI capabilities of the coupling of this PFEM code for Newtonian fluids and the non-linear FEM code *Metafor* [91] for solid mechanics. It features a bullet-shaped volume of fluid (approximating a bird) impacting an initially flat axisymmetric metallic panel that is clamped around the outside diameter. This is one of many scenarios that investigate the damage a bird causes to different parts of an airplane's wings upon impact. The bird approaches the center of the circular panel at a speed of $117\,\mathrm{m\,s^{-1}}$, a typical air speed of a jetliner shortly after takeoff. Axisymmetry is assumed in this test case. Its schematic and mesh used in the original publication are depicted in fig. 4.1.

In [9], the bird has a radius of $r_b = 60\,\mathrm{mm}$ with a length of $l_b = 3r_b$ and is modeled as a volume of incompressible water with a density $\rho_b = 1000.0\,\mathrm{kg\,m^{-3}}$ and a dynamic viscosity $\mu_b = 1.0 \times 10^{-3}\,\mathrm{Pa\,s}$. The metallic panel is relatively thin with a thickness of $h_p = 6.35\,\mathrm{mm}$, while the radius is $r_p = 0.4\,\mathrm{m}$. Its material is steel with the material properties given in [9].

Fig. 4.2 shows the results of the bird impact in [9]. Since a large deformation FEM code was used for the solid mechanics, their test case included large deformation of the solid, which cannot be reproduced with the small displacement solid code presented in this work. Thus, the original thin large panel is replaced by a small thick one. The new dimensions of the solid part are $h_p = 0.03\,\mathrm{m}$ and $r_p = 0.15\,\mathrm{m}$, as shown in fig. 4.3 with the properties in table 4.1. These settings are chosen to obtain a very stiff plate that barely deforms under the impact to not violate the small displacement assumption. The bird's properties



(a) Schematic. $r_b = 60\,\mathrm{mm}$, $l_b = 180\,\mathrm{mm}$, $d = 25\,\mathrm{mm}$, $r_p = 400\,\mathrm{mm}$, $h_p = 6.35\,\mathrm{mm}$, $v_{z_0} = -117\,\mathrm{m\,s^{-1}}$.

(b) Mesh used, showing symmetric image for better visualization

FIGURE 4.1: The original bird strike test case published by Cerquaglia et al. [9].

and dimensions remain unchanged.

TABLE 4.1: Material properties of the steel panel adapted for this work.

Property	**Symbol**	**Value**	**Unit**
Density	ρ	7000.0	$\mathrm{kg\,m^{-3}}$
Poisson's ratio	ν	0.3	
Young's modulus	E	1.0×10^{10}	Pa
Bulk modulus	K	8.333×10^{9}	Pa
Shear modulus	G	3.846×10^{9}	Pa

The coupled simulation is repeated as a reference with a setup that the present PFEM code is also able to replicate. The coupled method employs the nonlinear finite element code *Metafor* for the solid problem, where now a linear elastic material model is selected instead of the original elasto-plastic one. The fluid part is simulated using the same PFEM code as in the rest of this work. The coupling is accomplished by the coupling tool *CUPyDO* developed by Thomas et al. [110] in the same lab as *Metafor* and the present PFEM code, although it is not limited to these simulation codes (see work by Thomas et al. [110]

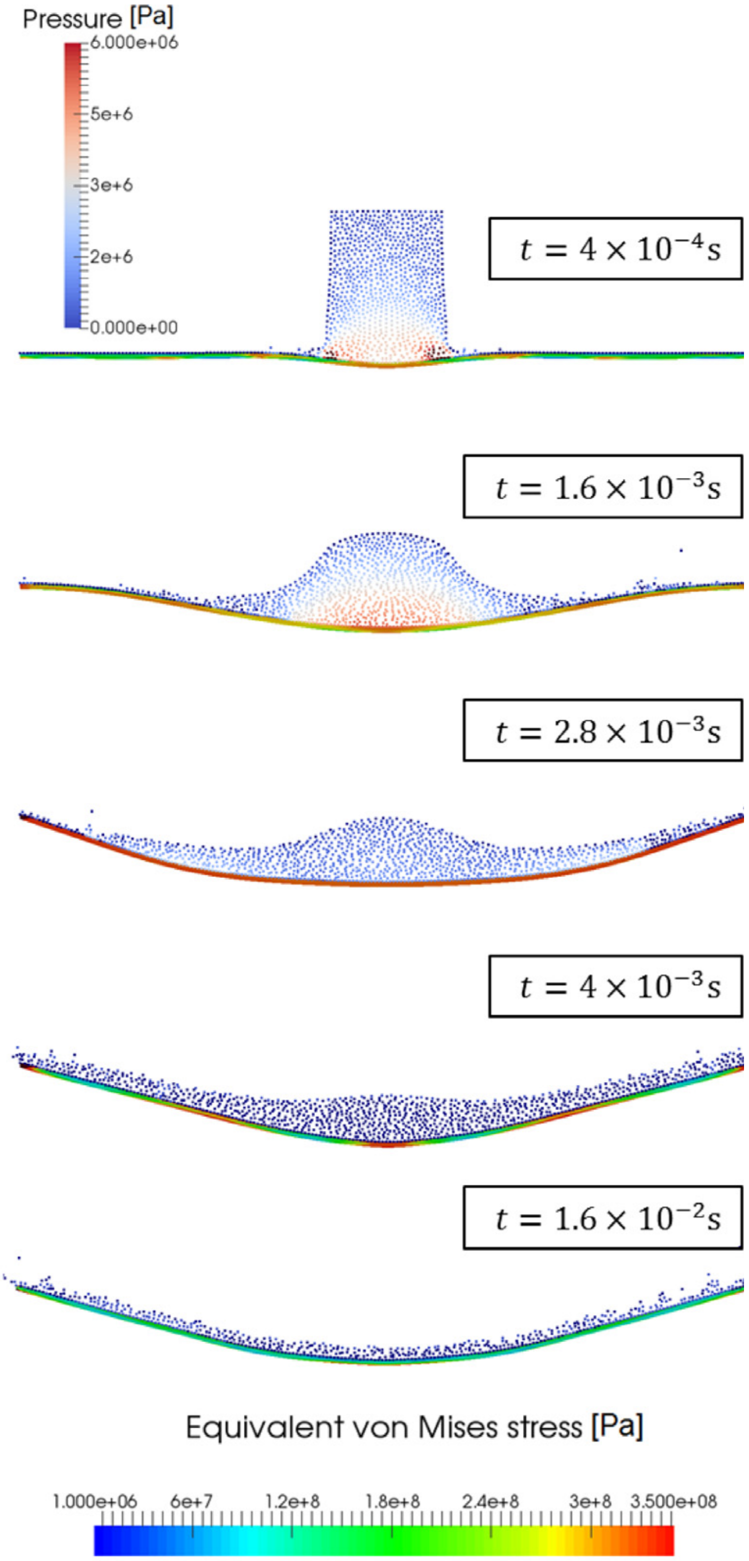
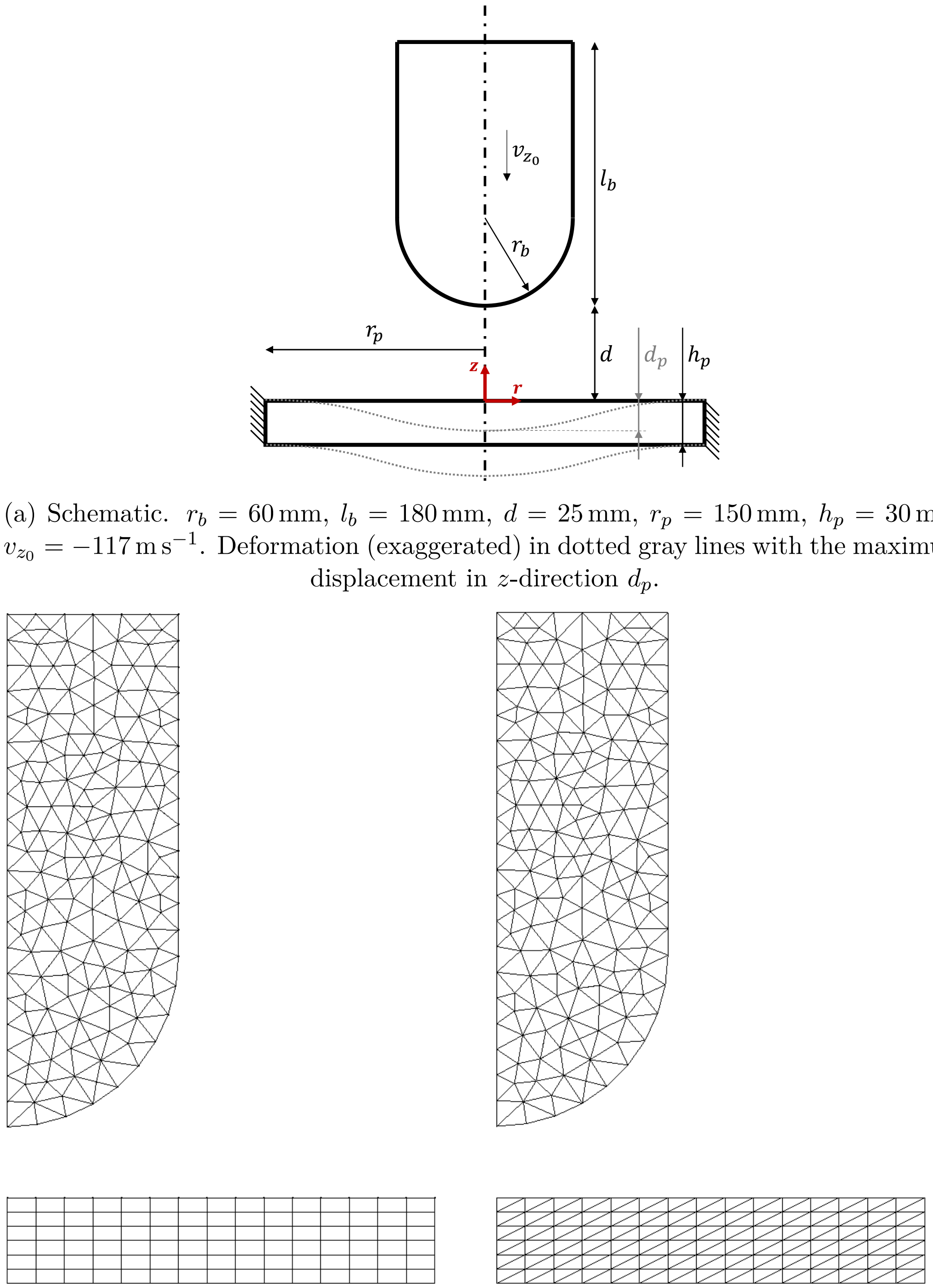


FIGURE 4.2: Results obtained by Cerquaglia et al. [9] for an axisymmetric bird strike FSI simulation. Pressure distribution in the fluid volume (bird) and von Mises equivalent stress $\bar{\sigma}_{VM}$ in the elasto-plastic steel panel shown.



(a) Schematic. $r_b = 60\,\mathrm{mm}$, $l_b = 180\,\mathrm{mm}$, $d = 25\,\mathrm{mm}$, $r_p = 150\,\mathrm{mm}$, $h_p = 30\,\mathrm{mm}$, $v_{z_0} = -117\,\mathrm{m\,s^{-1}}$. Deformation (exaggerated) in dotted gray lines with the maximum displacement in z-direction d_p.

(b) Mesh used by the coupled approach (Solid: *Metafor* [91], Fluid: PFEM [39], Coupling: *CUPyDO* [110]

(c) Mesh used with this present method: PFEM with a unified fluid-solid formulation

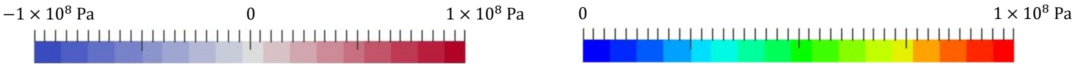
FIGURE 4.3: The new bird strike test case similar to Cerquaglia's [9]. Note that PFEM does not support quadrangle elements, which were kept in the coupled method.

and Crovato et al. [111, 112]). The coupling is accomplished by iterating between fluid and solid code during each time step. The fluid code computes the fluid's forces applied on the current boundary of the solid. These forces are then imposed as Neumann boundary conditions on the solid's boundaries in the solid code. The solid code then computes the solid's deformation and returns the updated solid boundary configuration to be given to the fluid code before repeating the whole process in the next iteration. The fluid boundary's displacement (or velocity) is hence imposed as a Dirichlet boundary condition. This scheme is referred to as "Dirichlet-Neumann paradigm" by Cerquaglia [39, 90] and many others before him. A more detailed summary of *CUPyDO* is provided in [39, 90].

The mesh of the solid panel used quadrangular elements because these were used in the original publication and are commonly known to be superior to triangular elements in bending. PFEM relies on the use of triangles, so the quadrangular elements of the coupled approach (4.3b) are replaced by pairs of triangles that form the same grid (4.3c). The other important difference between *Metafor* and the solid formulation implemented in PFEM is that the former solves for displacements (as is typical for purely solid mechanics codes), while the latter solves for pressure and velocity (as is typical for purely fluid mechanics codes). Together with the different mesh elements used, these are the two main differences of the methods used for the solid mechanics problem. The methods for the fluid mechanics problem can be considered nearly identical.

Both coupled and unified approaches use the same time step $\Delta t = 1.0\,\mu\text{s}$ over a total time of $t_{tot} = 2\,\text{ms}$ for this test case. The residual target for the momentum equation is $\bar{\epsilon} < \epsilon_{max} = 10^{-6}$ for all simulation codes. There is no gravity, no surface tension and no external pressure applied. There is no friction between dense matter and air and there is no slip between fluid and solid when they are in contact.

For the unified formulation, two simulation runs are presented: one with the interface between fluid and solid (interface nodes duplicated, see section 2.5.5) and one without any interface. In the latter case, the pressure at the shared nodes is unique and thus it is the same for the fluid and solid, which is unphysical. The comparison will highlight the necessity for an interface for accurate results. The color bars for the pressure and the equivalent von Mises stress are given in fig 4.4 and are used throughout the entire section for all results shown.

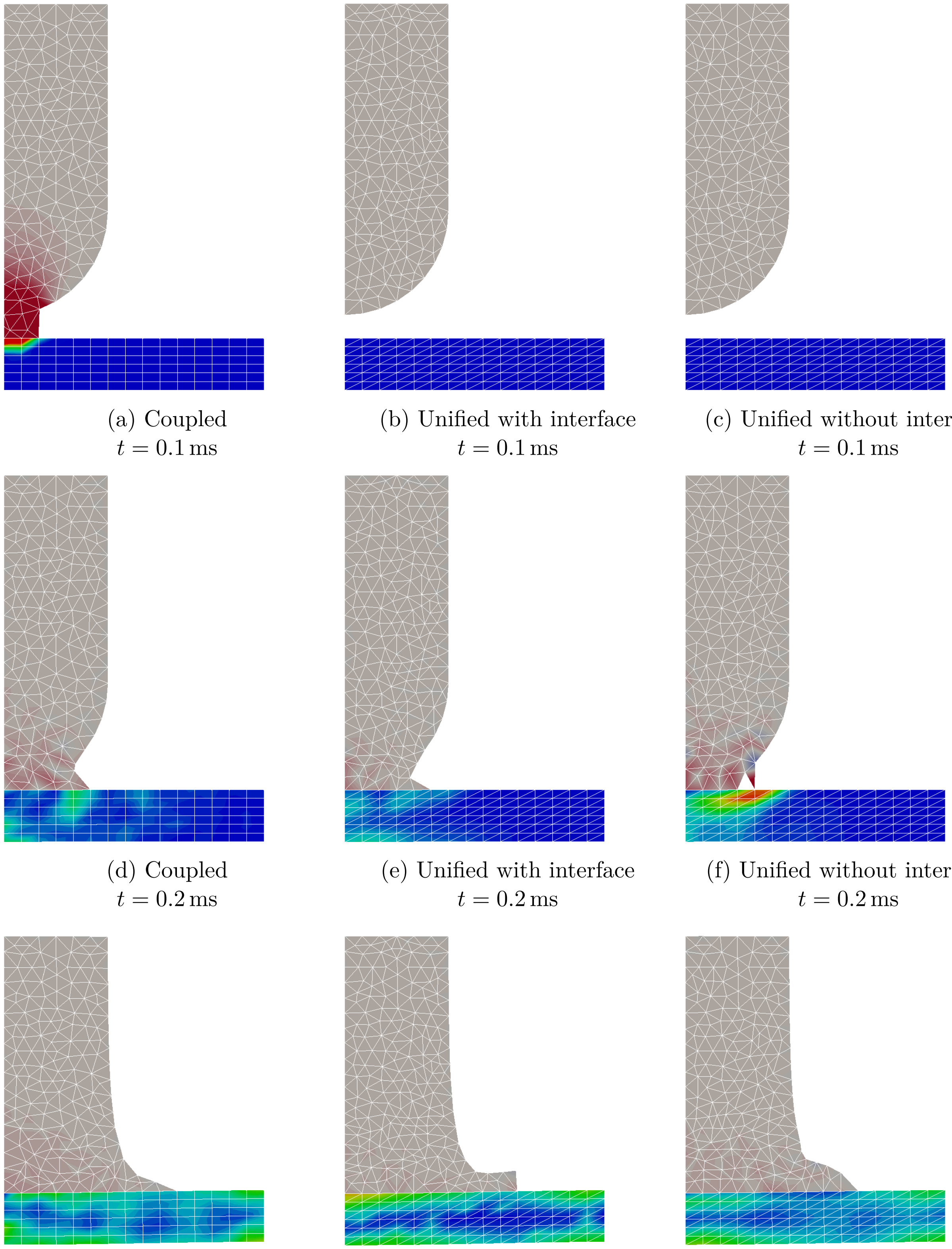


(a) Pressure p

(b) Equivalent von Mises stress $\bar{\sigma}_{VM}$

FIGURE 4.4: Color scales used for the bird strike problem.

In figs. 4.5 and 4.6, the evolution of the bird colliding with the panel can be observed. This collision is highly dynamic due to the large initial velocity v_{z_0} of the bird. The bird is making contact with the panel at around $t = 0.1\,\text{ms}$ in the first row in fig. 4.5. The most striking observation is that some elements have



(a) Coupled
$t = 0.1\,\mathrm{ms}$

(b) Unified with interface
$t = 0.1\,\mathrm{ms}$

(c) Unified without interface
$t = 0.1\,\mathrm{ms}$

(d) Coupled
$t = 0.2\,\mathrm{ms}$

(e) Unified with interface
$t = 0.2\,\mathrm{ms}$

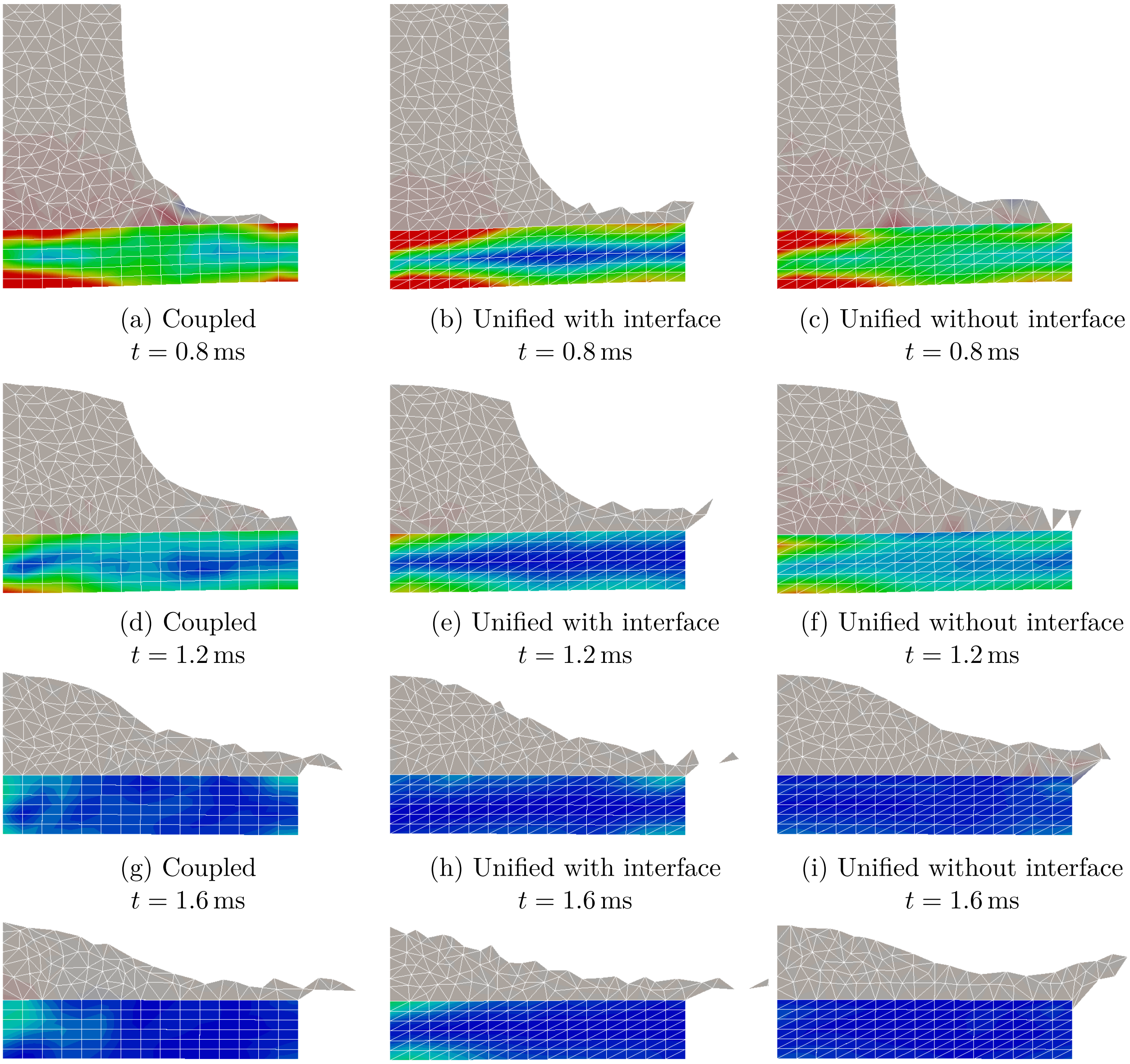
(f) Unified without interface
$t = 0.2\,\mathrm{ms}$

(g) Coupled
$t = 0.5\,\mathrm{ms}$

(h) Unified with interface
$t = 0.5\,\mathrm{ms}$

(i) Unified without interface
$t = 0.5\,\mathrm{ms}$

FIGURE 4.5: Bird strike results over time (top to bottom): Pressure p in the fluid, equivalent von Mises stress $\bar{\sigma}_{VM}$ in the solid. Comparison of coupled FEM-PFEM approach (left) and unified PFEM approach with (middle) and without an interface (right).



(a) Coupled
$t = 0.8\,\mathrm{ms}$

(b) Unified with interface
$t = 0.8\,\mathrm{ms}$

(c) Unified without interface
$t = 0.8\,\mathrm{ms}$

(d) Coupled
$t = 1.2\,\mathrm{ms}$

(e) Unified with interface
$t = 1.2\,\mathrm{ms}$

(f) Unified without interface
$t = 1.2\,\mathrm{ms}$

(g) Coupled
$t = 1.6\,\mathrm{ms}$

(h) Unified with interface
$t = 1.6\,\mathrm{ms}$

(i) Unified without interface
$t = 1.6\,\mathrm{ms}$

(j) Coupled
$t = 2.0\,\mathrm{ms}$

(k) Unified with interface
$t = 2.0\,\mathrm{ms}$

(l) Unified without interface
$t = 2.0\,\mathrm{ms}$

FIGURE 4.6: (contd.) Bird strike results over time (top to bottom): Pressure p in the fluid, equivalent von Mises stress $\bar{\sigma}_{VM}$ in the solid. Comparison of coupled FEM-PFEM approach (left) and unified PFEM approach with (middle) and without an interface (right).

already appeared in the gap between the panel and the bird in fig. 4.5a. This is normal behavior in PFEM, where such a gap is filled as soon as the α-shape criterion (eq. 2.192) is fulfilled. This is essentially a spurious addition of mass, a well-known problem of the classic PFEM (see e.g. Falla [96] for details). Due to a small incompatibility with *CUPyDO*, an older version of PFEM is used in the coupled method. Here, the only relevant effect of this is that the remeshing is slightly different in the coupled and the unified approach. This effect manifests itself most notably in figs. 4.5b and c, where no such elements have appeared (first element in the gap at $t = 0.107\,\mathrm{ms}$, not shown). This means that in those two figures we still see the bird traveling at the initial speed, undisturbed.

The panel is in its initial configuration as well, resting. The bird in fig. 4.5a, on the other hand, is experiencing a very high pressure in the contact region. Likewise, we see a large equivalent stress in the panel at the contact site, where the material is suddenly compressed.

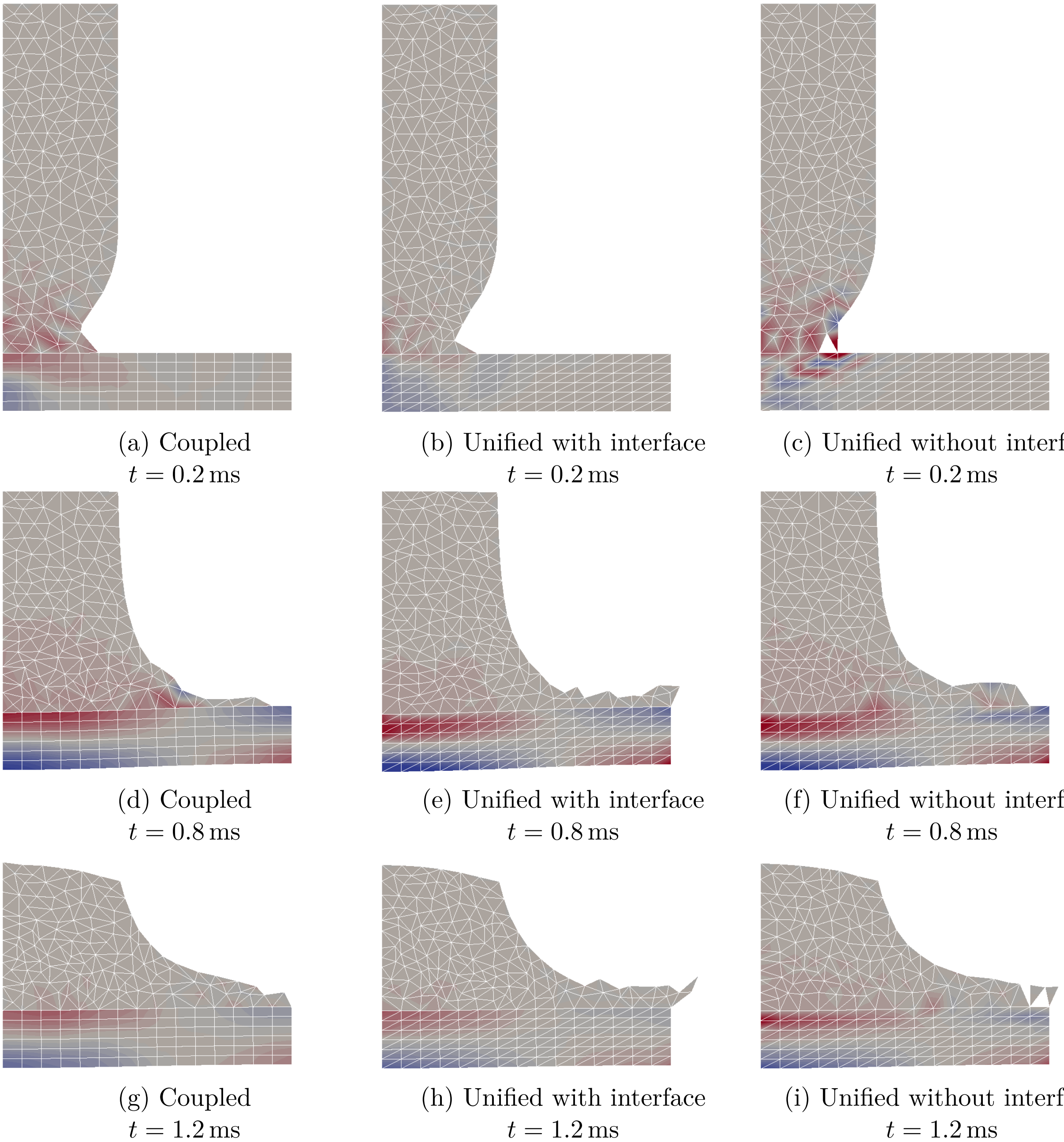
A bit later at $t = 0.2\,\mathrm{ms}$ in the second row of fig. 4.5, the contact area and the deformation of the tip of the bird is similar for all three methods. The pressure distribution in the bird and the stresses in the panel are distributed unevenly. For both, we see a snapshot of waves of pressure and stress propagating through the material. Upon close inspection, the bending of the panel begins. In fig. 4.5f, a new element has just been added on the far right. This element induces a pressure concentration in the fluid and a stress concentration in the solid. This is the same effect that is observed in fig. 4.5a. In both cases, contact has been recently established leading to this sudden and localized increase in stresses, which then reduce to a moderate level shortly after. This effect is typical for PFEM, where the α-shape algorithm decides when an element is created in the gap and the sudden appearance of such an element leads to an unphysical localized spike of contact forces.

At $t = 0.5\,\mathrm{ms}$ in the third row of fig. 4.5, the bird is squished against the relatively rigid panel and is beginning to spread outward. The pressure is highest at the axis and of similar magnitude for all three methods. In the solid, the highest stressed fibers are the ones furthest away from the neutral fiber. Superposed to the macroscopic stress distribution, one can notice the uneven pattern of higher and lower stress areas that propagate through the solid as a result of the highly dynamic impact. Another subtle difference between the unified PFEM approach with and without an interface can be found when examining the contact area in the solid. There, one finds that without the interface, the highest stress is not found in the fiber furthest from the neutral fiber (i.e. directly at the contact surface), but rather one element height away from the contact surface. This is an indirect effect of the nodal pressure at the contact surface, which is shared between fluid and solid, even though the pressure does not directly contribute to the equivalent von Mises stress $\bar{\sigma}_{VM}$ in fig. 4.5. Since the system must be at equilibrium, the forces normal to the contact surface must be equal and opposite, with or without the interface. So when the pressure is falsely smoothened on either the fluid or the solid side or both sides due to the absence of an interface, the deviatoric stresses are also affected, which is what is seen in the plot of $\bar{\sigma}_{VM}$.

At around $t = 0.8\,\mathrm{ms}$, the maximum displacement of the panel center is observed. Figs. 4.6a - c show a large equivalent stress magnitude in the panel. As before, the stress is concentrated at the top-most and bottom-most fibers in the regions where the curvature is largest. All methods show qualitatively similar stress distributions in the panel.

During the following time marks in figs. 4.6d - l, the panel swings back up and the stresses are consequently reducing to much lower levels. The bird has lost its shape completely and is continuously spreading out to a flat disk. The most interesting feature to be observed in these later stages is the splashing, i.e. separation of bird particles from the bulk in the typical PFEM fashion. Any element that does not violate the α-shape criterion continues to exist, even when completely separated from the bulk. Single particles are deleted in this simulation.

To complement the plots of the equivalent von Mises stresses in the solid, fig. 4.7 provides the pressure in the solid for some of the time marks. The most important observation is once again that the existence of an interface allows for a sharp discontinuity of the pressure between fluid and solid.



(a) Coupled
$t = 0.2\,\mathrm{ms}$

(b) Unified with interface
$t = 0.2\,\mathrm{ms}$

(c) Unified without interface
$t = 0.2\,\mathrm{ms}$

(d) Coupled
$t = 0.8\,\mathrm{ms}$

(e) Unified with interface
$t = 0.8\,\mathrm{ms}$

(f) Unified without interface
$t = 0.8\,\mathrm{ms}$

(g) Coupled
$t = 1.2\,\mathrm{ms}$

(h) Unified with interface
$t = 1.2\,\mathrm{ms}$

(i) Unified without interface
$t = 1.2\,\mathrm{ms}$

FIGURE 4.7: Bird strike results over time (top to bottom): Pressure p in the fluid and the solid. Comparison of coupled FEM-PFEM approach (left) and unified PFEM approach with (middle) and without an interface (right).

The highly dynamic impact leads to a deflection of the panel over time, quantified by the displacement of the top center-point of the solid disk d_p, as depicted in fig. 4.8. The coupled approach and the unified approach agree well despite the coarse discretization and the slight difference in the appearance of the first contact (see figs. 4.5a - c).

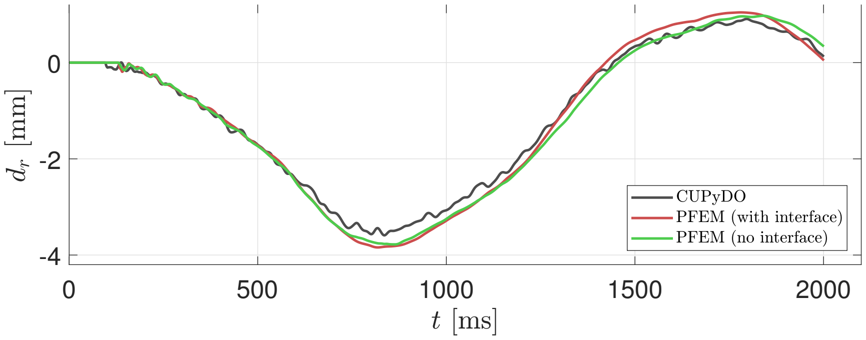


FIGURE 4.8: Displacement in z-direction of the top center point of the panel d_p. Coupled FEM-PFEM approach compared with unified PFEM approach, with and without an interface. Good overall agreement, but high-frequency oscillation in the solid barely captured by the unified PFEM approach.

In summary it can be noted that the unified fluid-solid formulation in PFEM is capable of replacing a traditional coupled approach in a highly dynamic FSI test case. The overall behavior of the bird and the panel agree between all methods. It is observed that the use of an interface does allow for the pressure to be discontinuous between fluid and solid at the contact surface (see fig. 4.7), which is the expected realistic behavior. However, fig. 4.8 demonstrates that the overall behavior is recovered even without an interface. It is unclear whether this conclusion can be reached in general or only for this specific test case.

For the investigation of an actual bird strike, both the coupled FEM-PFEM method and the PFEM with a unified formulation can be improved to be more realistic.

The appearance of elements in the gap before the bird is actually making contact with the panel should be avoided. In the classic PFEM, a finer mesh is the simplest solution, which leads to a smaller gap that can be achieved before the α-shape algorithm creates the unwanted elements. These unwanted elements would therefore be smaller, reducing the addition of mass. To completely remove this mass addition, the α-shape algorithm needs to be fundamentally improved or replaced by a different approach altogether. Methods that use a constrained Delaunay triangulation, for example, could preserve the real boundary of the bird in a smart way. Only when the boundary of the bird and the panel intersect, contact would be established. Then, a contact algorithm like the well-known penalty method could be used.

When bird and panel are already in contact, a free-slip condition with friction might provide a better representation of the real-world boundary interaction in the tangential direction.

The solid material model in the unified formulation of the PFEM is limited to small displacements, which is not sufficient for the simulation of the collision of birds with actual airplane wings, where large deformations are to be expected. In the same sense, the solid model in the unified formulation is currently limited to linear elasticity, which is not sufficient to model a real bird strike, where along with large deformations, even plasticity and even fracture can occur.

It may also be worth noting that the coupled approach took only 476 s to complete on a single processor, while the unified approach with an interface required 796 s (67% longer) and the unified approach without interface required 685 s (44% longer) on the same machine. The coupled approach is therefore more efficient for a classic FSI problem. Maybe equally interesting is the 14% CPU time reduction from the removal of the interface in the unified approach. This is likely caused by the increased size of the system of equations due to the use of Lagrange multipliers. Perhaps the interface option 3a (fig. 2.11c used by Franci [49]) should be compared in future work, since it does not add Lagrange multipliers.

4.2 TIG Spot Welding by Chen and Desmaison

A simulation method developed in the team of Prof. Bellet (CEMEF/2MS) at MINES Paris - PSL is presented in Chen's doctoral thesis [4] and in several articles by Chen et al. [26, 68]. The method is capable of simulating Laser powder bed fusion (L-PBF) at the meso-scale. Their work on L-PBF has been summarized in section 1.4.2, as it was the inspiration for the present work. Their method is based on a FEM code, originally designed for welding applications and later extended by Chen for L-PBF. In section 4.4 of his thesis [4], a spot welding test case is presented as part of the validation and verification process. This test case was originally published in the doctoral thesis [10] of Desmaison of the same research group. This test case is of interest because the results include the stress distribution. A brief comparison of the published results and the results from the present method follows.

The test case models a stationary Tungsten Inert Gas (TIG) welding heat source impacting the center of a cylindrical disk. Unlike Chen and Desmaison who use a 3D model (see mesh fig. 4.9), we make use of the axisymmetry assumption. The 2D domain is therefore a rectangle, with the axis of symmetry on the left at $r = 0$. The top surface at $z = 0$ is subjected to the heat source (electric arc) around the axis of symmetry. A Gaussian distribution of the heat source power density is used as described in section 2.30, with the parameters given in table 4.2. The nominal arc radius r_{std} (see eq. 2.29) given by the authors and no cut-off radius r_{cut} is used. The heat source is constant throughout the entire simulation and no cooling phase is included. No radiation and convection boundary conditions are applied to the top surface and all other surfaces are adiabatic. The right and bottom walls are allowed to slide and the top wall is

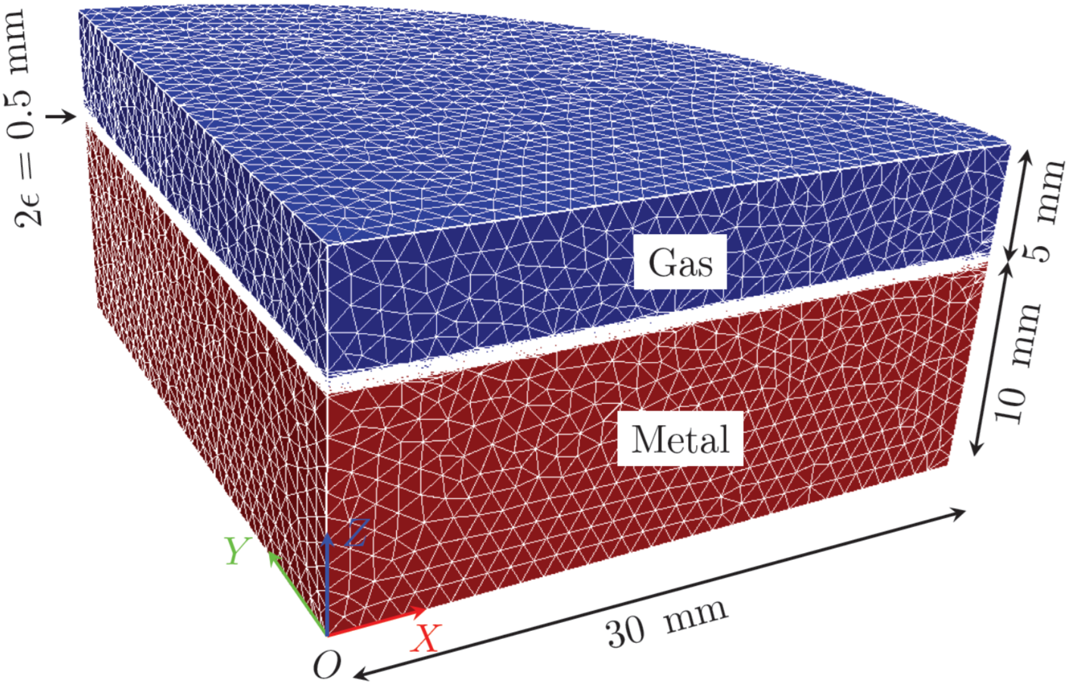


FIGURE 4.9: Geometry and mesh for spot welding by Desmaison [10] and later by Chen [4].

a free surface (see fig. 4.10).

In the original publications [4, 10], the material uses an elasto-visco-plastic (EVP) behavior below the solidus temperature $T_{sol} = 1691.15\,\text{K}$, a visco-plastic (VP) behavior in the mushy zone and a Newtonian behavior above the liquidus temperature $T_{liq} = 1781.15\,\text{K}$. The equations used for the material behavior can be found in Desmaison's thesis [10]. One important detail that Desmaison mentions is that the yield stress $\sigma_y = 0$ in this test case for the elasto-visco-plastic part, which essentially makes it a visco-plastic material as well and no elasticity occurs. No gravity and no external pressure act on the material, the only loading stems from the constrained thermal expansion.

The liquid melt pool becomes Newtonian, but without a Marangoni effect, without natural convection (no gravity) and without recoil pressure (evaporation), no force whatsoever acts on the fluid. The fluid therefore does not flow, it is only displaced upwards as the solid material below expands under the increasing temperature.

The simulation method employed by [4, 10] is based on an Eulerian FEM for the staggered thermal and fluid solution and then switches to a Lagrangian description for the solid solution. The material-atmosphere interface is tracked using a level set approach. A more detailed summary of the underlying method and its solution scheme is given in section 1.4.2 on page 21. Note that the mesh used by [4, 10] (fig. 4.9) is relatively coarse (edge length $\bar{l}_e = 0.5\,\text{mm}$), while the interface has small elements (edge length $\bar{l}_e = 0.05\,\text{mm}$) in a layer of thickness 0.5 mm.

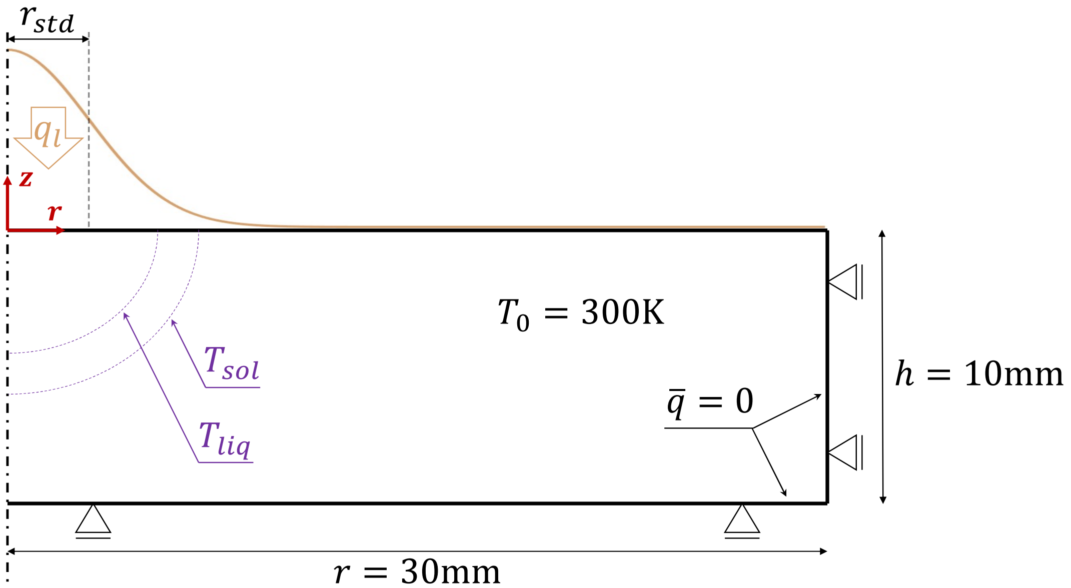


Figure 4.10: Schematic of TIG spot welding test case by Chen [4] and Desmaison [10]. A Gaussian distribution is used for the heat flux boundary condition that models the electric arc.

The material properties are found in Annex 2 of the doctoral thesis of Costes [11]. Most of the material properties vary as a function of temperature, so table 4.2 only provides the ranges of each property to give a rough idea. Note that the EVP and VP material laws use parameters that are not listed here, but that can be found in Costes' work [11]. Costes also provides the elastic properties

Table 4.2: Material properties used.

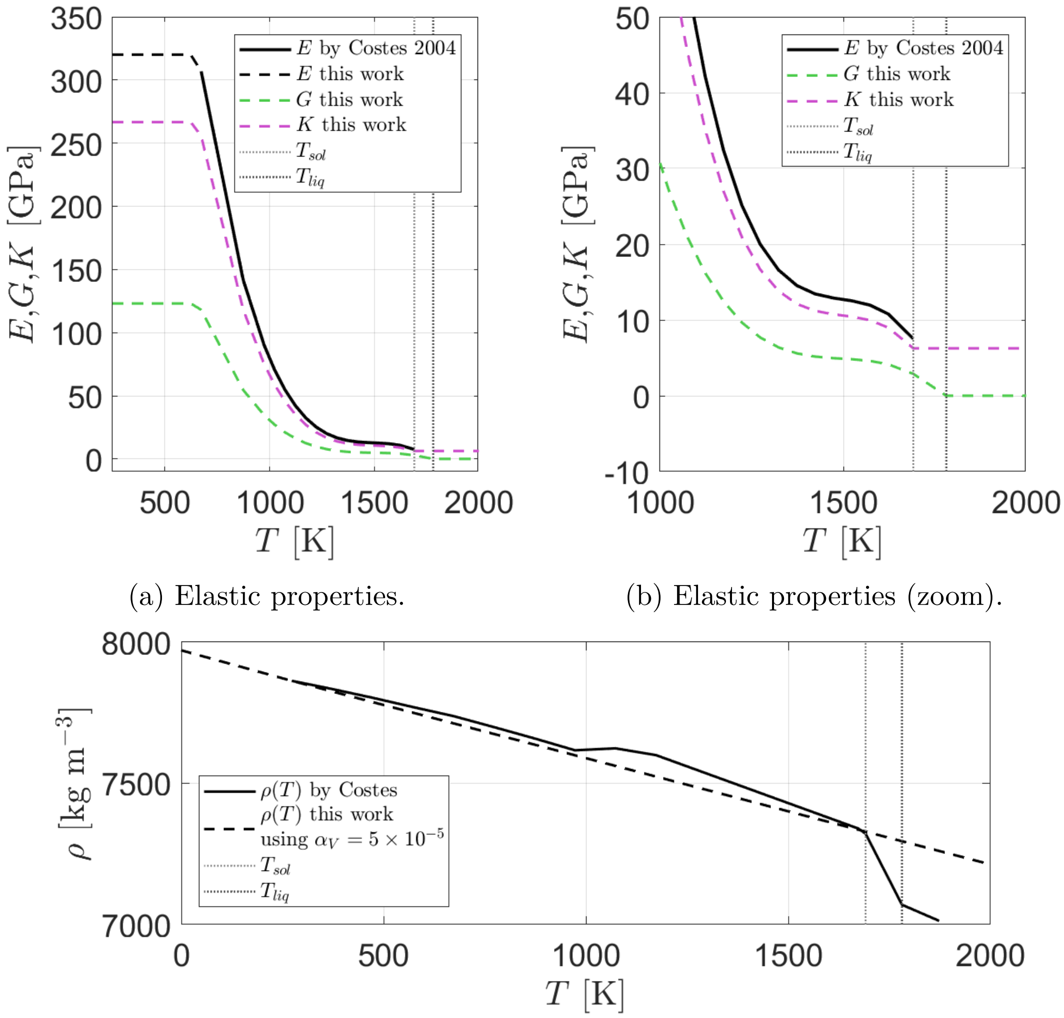
Property	Symbol	Value	Unit
Gravity	g	0.0	$\text{m}\,\text{s}^{-2}$
Viscosity	μ_f	0.001	$\text{Pa}\,\text{s}$
Heat capacity	c_p	$[469, 1431]$	$\text{J}\,\text{kg}^{-1}\,\text{K}^{-1}$
Conductivity	k	$[25.9, 51.8]$	$\text{W}\,\text{m}^{-1}\,\text{K}^{-1}$
Density	ρ	$[7013, 7863]$	$\text{kg}\,\text{m}^{-3}$
Thermal expansion coeff.	α_V	$\approx 5.0 \times 10^{-5}$	K^{-1}
Latent heat	L_m	260000	$\text{J}\,\text{kg}^{-1}$
Solidus temperature	T_{sol}	1691.15	K
Liquidus temperature	T_{liq}	1781.15	K
Nominal heating power	P_l	1200	W
Heat source radius	r_{std}	3.0×10^{-3}	m
Poisson's ratio	ν	0.3	
Young's modulus	E	$[7.511, 306.848] \times 10^9$	Pa

(E and ν), which are not used by Chen [4] or Desmaison [10] due to the yield strength $\sigma_y = 0$. For the simulations with the present method, only the elastic law is used for the solid with the E and ν provided, not the original EVP law. Figs. 4.11a,b show $E = E(T)$, where a significant decrease in stiffness is observed at elevated temperatures. The temperature range of E provided (solid black line) is insufficient, so the function was extended (dotted black line). Since the input in our code uses the bulk and shear moduli K and G, they were derived from E using eqs. 3.23 and 3.22 respectively, with the given Poisson's ratio $\nu = 0.3$. The colored dotted lines in figs. 4.11a,b represent these parameters, which we extended for $T > T_{sol}$. Upon melting we choose that the bulk modulus K remains constant $K(T \geq T_{sol}) = K(T_{sol})$, while the shear modulus decreases linearly from $G(T_{sol})$ to $G(T \geq T_{liq}) = 0$. Costes [11] provides no viscosity for the fluid, so the one in table 4.2 was chosen. Costes [11] also does not provide a thermal expansion coefficient α_V, but rather a density $\rho = \rho(T)$. The constant value of α_V given in table 4.2 is approximated from $\rho(T)$ using eq. 2.37, which happens to be a reasonable fit, as depicted in fig.4.11c.

For the semi-solid phase, the viscous effect is mixed with the effect of the decreasing shear modulus, but no residual stresses are retained (see eq. 2.118). The mixing law used is known as the Kelvin-Voigt visco-elastic model and it adds the fluid viscosity μ_f and the viscosity-like parameter μ^* of the elastic solid to yield an equivalent viscosity $\mu_{eq} = \mu_f + \mu^*$. The equivalent viscosity μ_{eq} is then used in place of μ in eq. 2.79. This ensures a smooth transition from solid to fluid, where the semi-solid acts like a fluid with very high viscosity when near T_{sol}, which decreases linearly until it reaches the nominal viscosity of the fluid at T_{liq}.

The simulation is carried out with a constant time step $\Delta t = 1\,\text{ms}$ over the total time $t_{end} = 10\,\text{s}$. The mechanical problem is solved using the Picard algorithm, while the thermal problem is solved using the Newton-Raphson algorithm with a line search algorithm to speed up the non-linear iterations. Mesh refinements are set for the temperatures $T \leq T_{sol}$ to maintain the mesh density distribution of the initial mesh. The residual target for the momentum equation and the heat equation is set to $\bar{\epsilon} < \epsilon_{max} = 10^{-4}$.

Desmaison [10] provides the temperature distribution obtained after 10 s of heating, including the isolines of T_{sol} and T_{liq} in fig. 4.12. The image published in [10] appears to have a different width-to-height ratio of the domain. Whether the image has been scaled down in the horizontal direction or whether it has been trimmed back at the left or right side is unclear. Note that the temperature range of the color bar is the same, but that the color representation may not match exactly. Desmaison does however state that the melt pool has a width $w_m = 3\,\text{mm}$ and a depth $h_m = 1\,\text{mm}$, but since no decimal places were given, it is unclear, whether this is coincidence or whether some uncertainty prompted them to not be more precise. The present PFEM method obtains a width $w_m = 3.48\,\text{mm}$ and a depth $h_m = 1.26\,\text{mm}$ with the above described settings and with the melt pool boundary assumed at $T_m = 1/2(T_{sol} + T_{liq})$. The variance with different settings (different mesh, time step, residual target, incompressible fluid etc.) is relatively low. Both the comparison of the contour



(a) Elastic properties.

(b) Elastic properties (zoom).

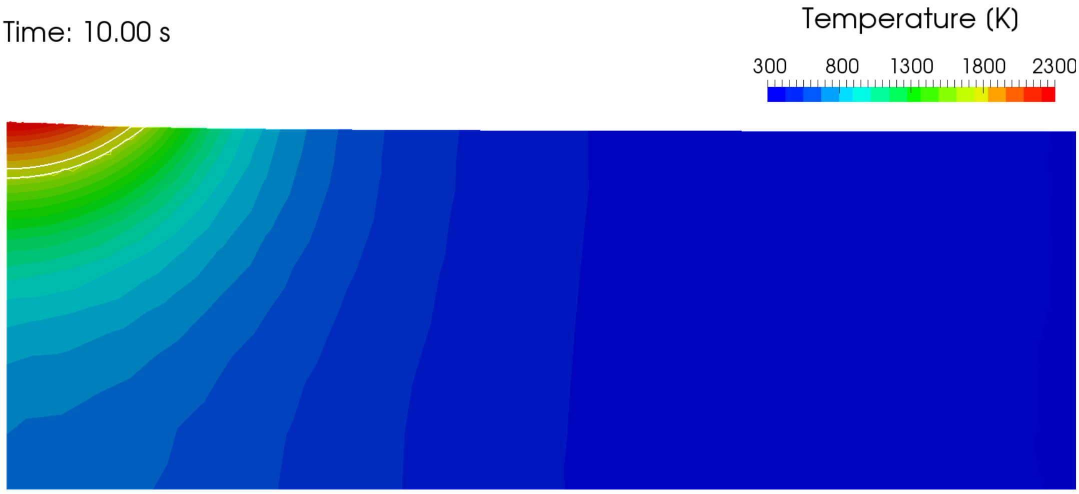
(c) Density.

FIGURE 4.11: Some material properties as given by Costes [11] compared with the ones used in this work.

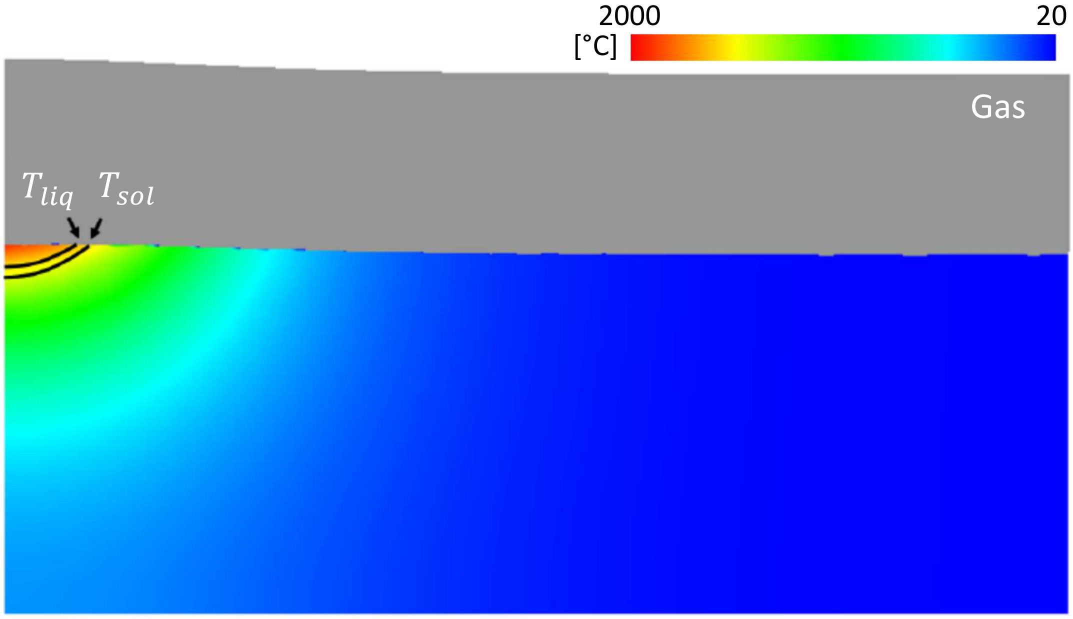
plots and the numerical values indicate that the present method produces a larger melt pool. The reason for this is unknown.

Comparing the distribution of the equivalent von Mises stress $\bar{\sigma}_{VM}$ is not expected to produce a good match due to the different material models used in the solid (non-linear visco-plastic solid in [4, 10] against a linear elastic in the present PFEM code). We shall show the two contour plots side-by-side anyway, since it may be interesting to see where there are similarities and where the results differ strongly. Furthermore, future work aiming to replicate such simulations may want to begin with a simple linear elastic solid and later move on to a non-linear (elasto-)visco-plastic one and the results in appendix B may be useful for that.

First, note that the $\bar{\sigma}_{VM}$ distribution in fig. 4.13b is the one published in Chen's



(a) The present method.



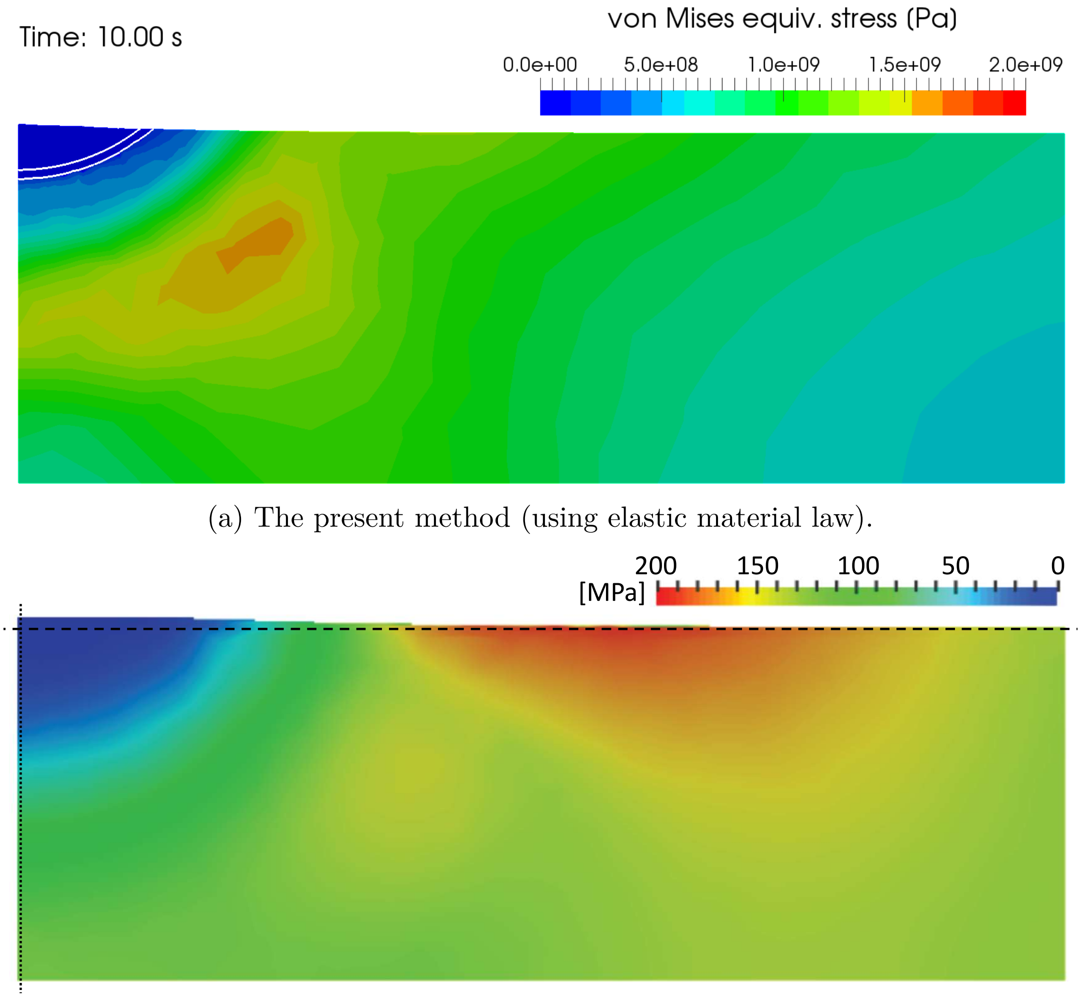
(b) Desmaison.

FIGURE 4.12: Comparison of temperature field by Desmaison [10] with this work.

doctoral thesis [4]. Desmaison [10] published results for several settings regarding the free surface tracking, but the one that corresponds to those used by Chen appears to have an identical stress distribution and is not shown here. Second, note that the range of the magnitude of $\bar{\sigma}_{VM}$ is 10 times larger in the plot obtained with the present method (fig. 4.13a) than in the reference (fig. 4.13b).

With that in mind, the most important observation is that the linear elastic material in fig. 4.13a produces much higher stresses ($\max(\bar{\sigma}_{VM}) = 1.62{\times}10^9$ Pa). Furthermore, the area of highest $\bar{\sigma}_{VM}$ is in a different location and covers a different sized area of the domain.

The only similarity, aside from the melt pool having $\bar{\sigma}_{VM} = 0$, is the band of relatively low $\bar{\sigma}_{VM}$ around the melt pool. For the elastic material used in the present method (fig. 4.13a), this is clearly to be attributed to the decrease



(a) The present method (using elastic material law).

(b) Chen [4] (using visco-plastic material law). Note that a similar plot is published by Desmaison [10] that is virtually indistinguishable from Chen's.

FIGURE 4.13: Comparison of equivalent von Mises stress field.

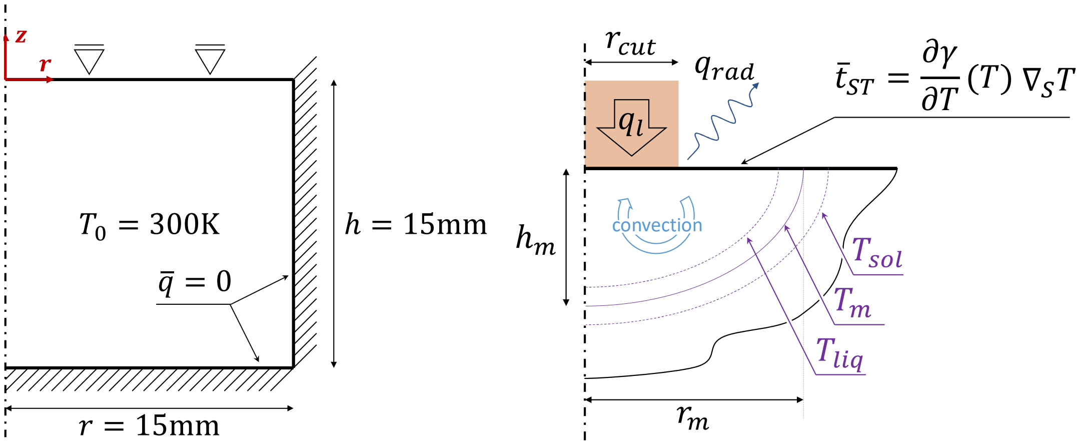
in Young's modulus E over the temperature T (fig. 4.11a). This band is of very high temperature, making the material very compliant and reducing the stresses. Based on the material parameters given by Costes [11], it is possible for the non-linear visco-plastic material to also become more compliant at high temperature, which could explain the band of low $\bar{\sigma}_{VM}$ in fig. 4.13b, but it is impossible to be certain.

To summarize the observations, it is hardly surprising that the use of different material laws produces a different stress distribution. Sadly, the present method does not yet have an elasto-visco-plastic material law implemented, which would be required for a fair comparison. Nevertheless, the present method is capable of producing plausible results for a spot welding test case such as this one.

4.3 Laser Spot Welding by Saldi

4.3.1 Problem description and simulation setup

Saldi's doctoral thesis [12] features simulations of laser spot welding of a cylindrical disk of steel, schematized in fig. 4.14. He examines the influence of sulfur concentration in the steel on the melt pool shape, temperature profile and velocity field. Saldi is comparing his results to those of Pitscheneder et al. [81], who conducted experiments and simulations to investigate the influence of sulfur concentration during laser spot welding of steel themselves. The sulfur content is assumed to only affect the Marangoni coefficient $\partial\gamma/\partial T$, which is known to heavily influence the above mentioned characteristics of a laser spot welding process. They compare two concentrations: 20 ppm and 150 ppm of sulfur. The Marangoni coefficient is temperature dependent and even changes sign as the melt pool heats up, as shown in fig. 4.15. The mathematical expression can be found in [12] and [81]. In this work we are not interested in the effect of sulfur concentration, so we pick only the 20 ppm variant and compare the results for the purpose of validation.



(a) Whole domain (depicting the flat surface variant).

(b) Details top left corner.

FIGURE 4.14: Schematic of laser beam welding by Saldi [12].

Except for $\partial\gamma/\partial T$, only the heat capacity c_p changes between the fluid and solid phase. All material properties are listed in table 4.3.

Saldi only takes the tangential component of the surface tension into account, not the normal component. The laser heat source is modeled as a surface heat flux with a top-hat profile:

[1] The Marangoni number Ma varies over time and depends strongly on the evolution of the melt pool. To give the reader an idea of the order of magnitude, the value in table 4.3 is calculated using the maximum temperature difference encountered in Saldi's results ($\Delta T_c \approx$ 2200 K), an estimated representative radius of the melt pool as the characteristic length $l_c \approx 2.5\,\text{mm}$, the largest magnitude of the Marangoni coefficient $(|\partial\gamma/\partial T|) = 4.945 \times 10^{-4}$ according to fig. 4.15 for a 20 ppm concentration of sulfur.

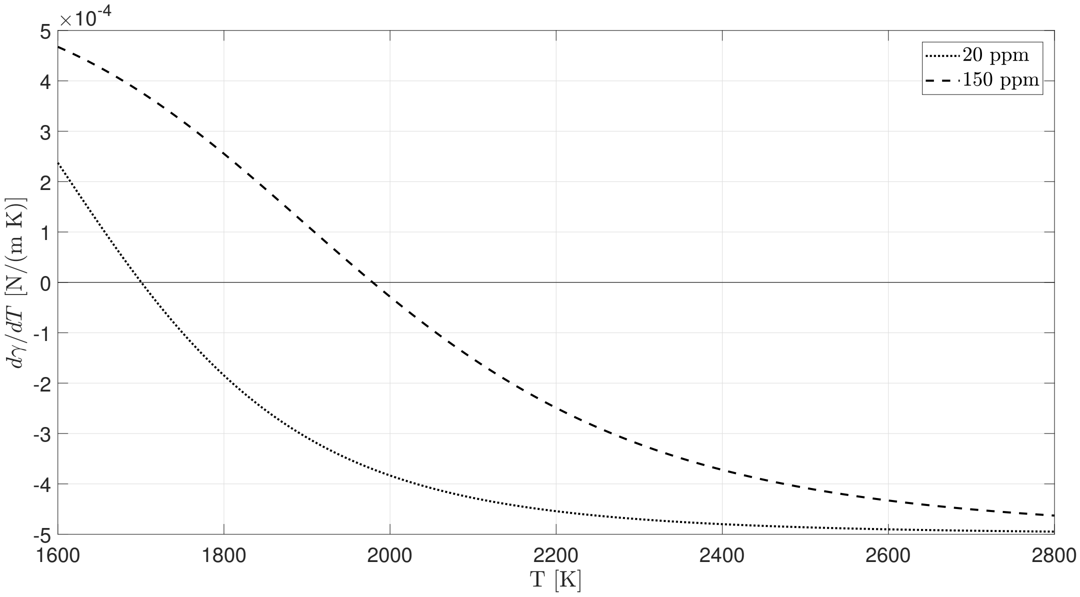


FIGURE 4.15: Marangoni coefficient as function of temperature for two concentrations of sulfur in steel, according to Saldi [12]. Note that only the 20 ppm curve is used in this work. The sign change of the 20 ppm curve occurs at $T = T_{crit} = 1700.5\,\mathrm{K}$.

$$\bar{q}_l(r) = \begin{cases} \frac{\alpha_l P_l}{\pi r_{cut}^2} & \text{for } r \leq r_{cut} \\ 0 & \text{for } r > r_{cut} \end{cases} \tag{4.1}$$

where $\alpha_l = 0.13$ is the absorptivity, $r_{cut} = 1.4\,\mathrm{mm}$ is the nominal radius of the laser beam (and acting as cut-off radius) and P_l is the nominal laser power as given by Saldi. He varies the laser power between three options $P_l \in \{1900,\ 3850,\ 5200\}\mathrm{W}$, but we will only consider the intermediate $P_l = 3850\,\mathrm{W}$ in this work.

Saldi elaborates on the use of a so-called enhancement factor f_{enh} that artificially increases thermal and momentum diffusion in the fluid:

$$\mu_f = \begin{cases} \mu_{f_0} & \text{where} f_l = 0 \\ \mu_{f_0} f_{enh} & \text{otherwise} \end{cases} \tag{4.2}$$

$$k = \begin{cases} k_0 & \text{where} f_l = 0 \\ k_0 f_{enh} & \text{otherwise} \end{cases} \tag{4.3}$$

where μ_{f_0} and k_0 are the nominal viscosity and conductivity. This factor presumably accounts for small-scale effects such as turbulence, but it is not clear. Saldi lists several other authors that use an enhancement factor $f_{enh} = 1 \cdots 100$ in similar test cases and describes that f_{enh} is used by these authors to tune their numerical results to fit the experimental ones. He then dedicates several pages on comparing the influence of the choice of enhancement factor on the melt pool shape and dimensions for the flat surface variant only. For the different laser powers and sulfur concentrations, it appears that the best match with

TABLE 4.3: Material properties

Property	**Symbol**	**Value**	**Unit**
Density at T_{ref}	ρ_{ref}	8100.0	$\mathrm{kg\,m^{-3}}$
(Nominal) Viscosity	μ_{f0}	6.0×10^{-3}	$\mathrm{Pa\,s}$
Heat capacity (fluid)	$c_{p,f}$	723.14	$\mathrm{J\,kg^{-1}\,K}$
Heat capacity (solid)	$c_{p,s}$	627.00	$\mathrm{J\,kg^{-1}\,K}$
(Nominal) Conductivity	k_0	22.9	$\mathrm{W\,m^{-1}\,K}$
Latent heat	L_m	25080	$\mathrm{J\,kg^{-1}}$
Melting point	T_m	1620	K
Thermal exp. coeff.	α_V	0	$\mathrm{K^{-1}}$
Thermal diffusivity (liquid)	α_l	3.91×10^{-6}	$\mathrm{m^2\,s^{-1}}$
Thermal diffusivity (solid)	α_s	4.51×10^{-6}	$\mathrm{m^2\,s^{-1}}$
Prandtl number (liquid)	Pr	0.189	-
Rayleigh number (liquid)	Ra	0	-
Marangoni number[1]	Ma	$< 1.16 \times 10^{-5}$	-

Pitscheneder's experimental results is achieved with $f_{enh} = 4 \cdots 8$. This can be seen in the plots that Saldi provides, but Saldi does not explicitly make such a statement. For the rest of the simulations, Saldi uses $f_{enh} = 1$ or $f_{enh} = 2$ for the flat surface variants and $f_{enh} = 1$ (i.e. no enhancement) for the free surface simulations. Thermal expansion is also neglected in Saldi's simulations. The solid material is governed by the Carman-Kozeny equation (see section 2.3.3) with $C_{CK} = 1 \times 10^6\,\mathrm{kg\,m^{-3}\,s^{-1}}$ and $\epsilon_{CK} = 1 \times 10^{-3}$.

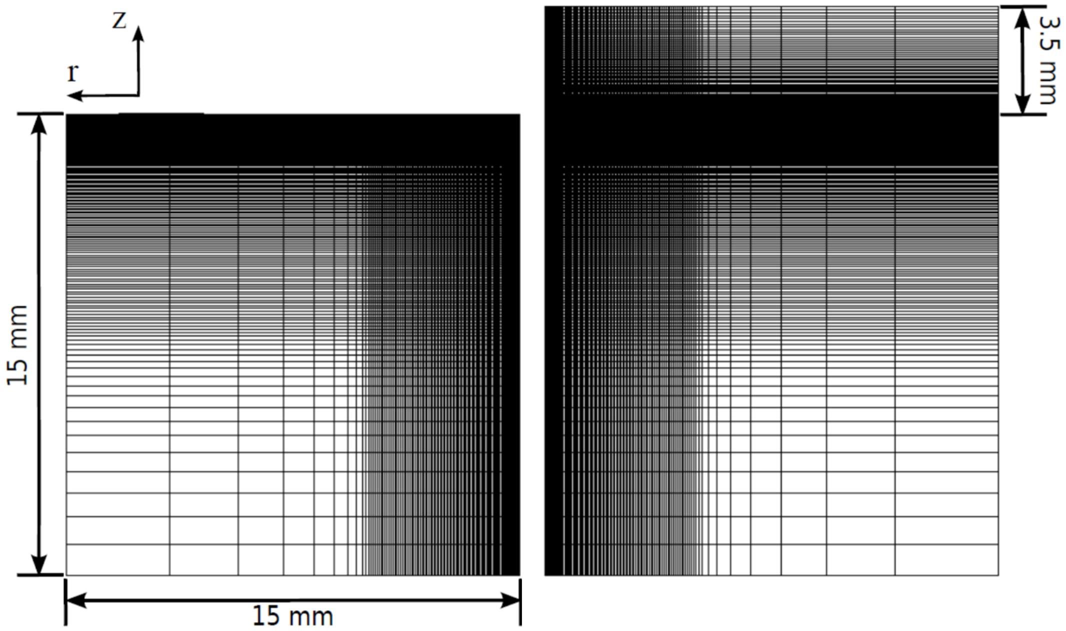
Saldi is modeling a cylindrical disk of radius $r = 15\,\mathrm{mm}$ and height $h = 15\,\mathrm{mm}$ in an axisymmetric representation. The left side wall acts as the axis of symmetry, while the right side wall and the bottom wall are fixed in Saldi's test case. Due to the Eulerian description of motion used in his method, he presents two approaches to model the free surface:

1. Either assume that the top surface remains flat and impose a free-slip boundary there or

2. include the gaseous atmosphere in the model and allow the free surface to deform with the aid of an interface tracking algorithm.

Saldi pursued both options and made comparisons of the results. The meshes used by Saldi are depicted in fig. 4.16.

The laser irradiation time is $t_l = 5.0\,\mathrm{s}$ after which the domain is left to cool until the end time of the simulation $t_{end} = 5.5\,\mathrm{s}$. A typical time step $\Delta t = 1.0\,\mathrm{ms}$ is used by Saldi such that the Courant-Friedrichs-Levi (CFL) number $C = v\Delta t / l_e < 2$, where v is the velocity magnitude and l_e is the cell size (representative element edge length) at any location.



(a) Assuming a flat top surface. 110 cells radially, 160 cells axially with a smallest cell edge length $\min(l_e) = 0.02\,\mathrm{mm}$

(b) With atmosphere included to allow for a deforming top surface. 110 cells radially, 230 cells axially with a smallest cell edge length $\min(l_e) = 0.02\,\mathrm{mm}$

FIGURE 4.16: Meshes used by Saldi [12].

The simulations with the present PFEM formulation shall be split into 3 parts:

1. Purely fluid simulation with the Carman-Kozeny equation to govern the solid region and with a flat or free surface.
2. Simulation with the unified fluid-solid approach
   (a) with a flat surface
   (b) with a free surface.
3. Simulation with the unified fluid-solid approach and with a free surface and with an additional thermal expansion not included in Saldi's test cases

The first variant is to verify good agreement with the simulation results obtained by Saldi [12], as the settings are most similar to his approach. Comparisons with experimental results by Pitscheneder [81] are also possible, but the data is very limited. The second variant acts as an intermediate step to ensure that the unified formulation continues to deliver the results as published by Saldi. Comparison with experimental results is not possible, since a flat surface does not represent real-world physical behavior of a melt pool. The third variant is meant to demonstrate the potential of the unified approach by introducing non-uniform stresses into the solid that evolve over time due to thermal expansion. The cost for such simulations are very high and a very limited data set was generated with the unified method. The third variant is mainly attempting to show how the present method can go beyond the more conventional method, such as Saldi's one.

For the simulation setup using the present PFEM method, some small modifications must be made. This is mainly due to time constraints, as Saldi's mesh size and time step cannot be used without incurring excessive CPU cost using the PFEM. In short, the mesh size was increased by a factor of > 5 and only the high enhancement factors $f_{enh} \in \{8, 16\}$ are used, allowing a larger time step as well. The reasons for this are explained in more detail in section 4.3.6.

Some settings are shared between all three variants, such as the material properties, as given in table 4.3. Solid properties are given when the unified fluid-solid approach is introduced. The mechanical problem is again solved using the Picard algorithm, while the thermal problem is solved using the Newton-Raphson algorithm with a line search algorithm to speed up the iterations. In all variants, the initial temperature is $T_0 = 300\,\mathrm{K}$. The sharp melting point $T_m = 1620\,\mathrm{K}$ is replaced by a melting interval of solidus and liquidus temperatures: $T_{sol} = 1610\,\mathrm{K}$, $T_{liq} = 1630\,\mathrm{K}$. A linear function for the liquid fraction f_l is assumed within the interval. Furthermore, a regularization (see section 2.3.1) of the transition between constant and linear f_l with $\epsilon_{reg} = \pm 2\,\mathrm{K}$ is chosen to further smoothen the transition and improve convergence. The residual target for the momentum equation and the heat equation is set to $\bar{\epsilon} < \epsilon_{max} = 10^{-4}$.

All three variants share the same mesh refiners, which are set to refine areas of high temperature gradient, high velocity gradient and for temperatures near the melting point. The remeshing parameters are set to $\alpha_{crit} = 1.4$ (section 2.7.3), $\gamma_{crit} = 0.4$, $\omega_{crit} = 0.4$ (section 2.7.4).

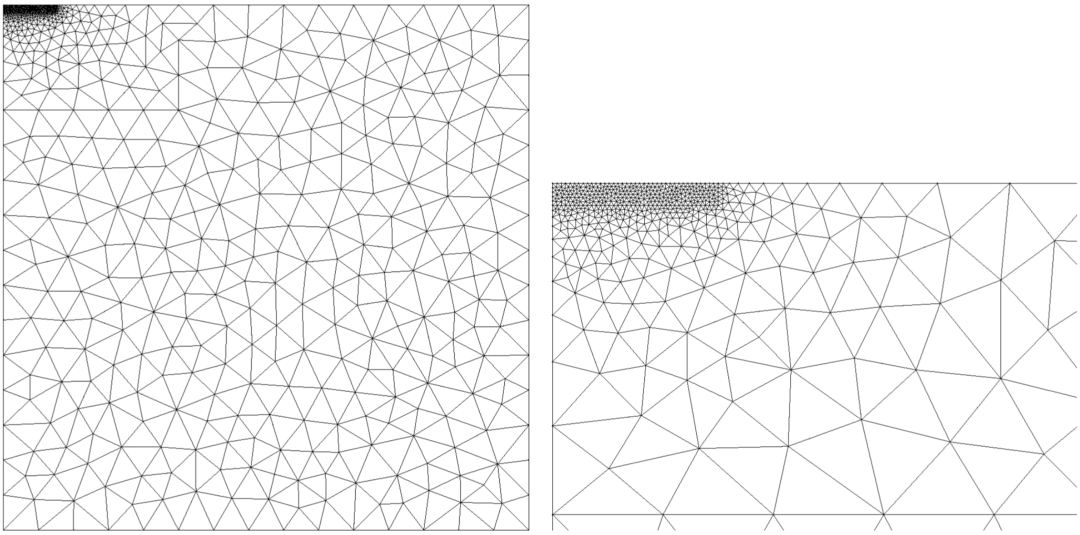
In many cases, only the top left corner of the domain is shown, where the melt pool exists and where the laser impacts. Most often, nothing of interest is to be seen in the rest of the domain.

4.3.2 Simulation using a fluid with the Carman-Kozeny equation and a free surface

This approach uses a Newtonian fluid to represent the material in the entire domain. Wherever the material is below liquidus temperature T_{liq}, the well-known Carman-Kozeny equation (see section 2.3.3) is used. The values for the parameters $C_{CK} = 1 \times 10^6\,\mathrm{kg\,m^{-3}\,s^{-1}}$ and $\epsilon_{CK} = 1 \times 10^{-3}$ are given, but it was found that instead using $\epsilon_{CK} = 1 \times 10^{-5}$ yields more realistic results (the solid is significantly more rigid) and reduces a mass conservation issue described in section 4.3.6. The latter value is therefore used inresults described in this section.

Using the mesh in fig. 4.17, where only a minimal area is initially refined, and a constant time step of $\Delta t = 2.0\,\mathrm{ms}$ with $f_{enh} = 16$ the following results are obtained.

Fig. 4.18 shows the free surface of the melt pool and the melting front after 5 seconds, comparing the PFEM results with $f_{enh} = 16$ with the results obtained by Saldi for a flat surface. Unfortunately, the only free surface results obtained



(a) Whole mesh. Largest edge length $\max(l_e) = 1\,\mathrm{mm}$.

(b) Zoom on refined area. Smallest edge length $\min(l_e) = 0.12\,\mathrm{mm}$.

FIGURE 4.17: Mesh used for purely fluid approach. Only the area where laser impacts is refined initially.

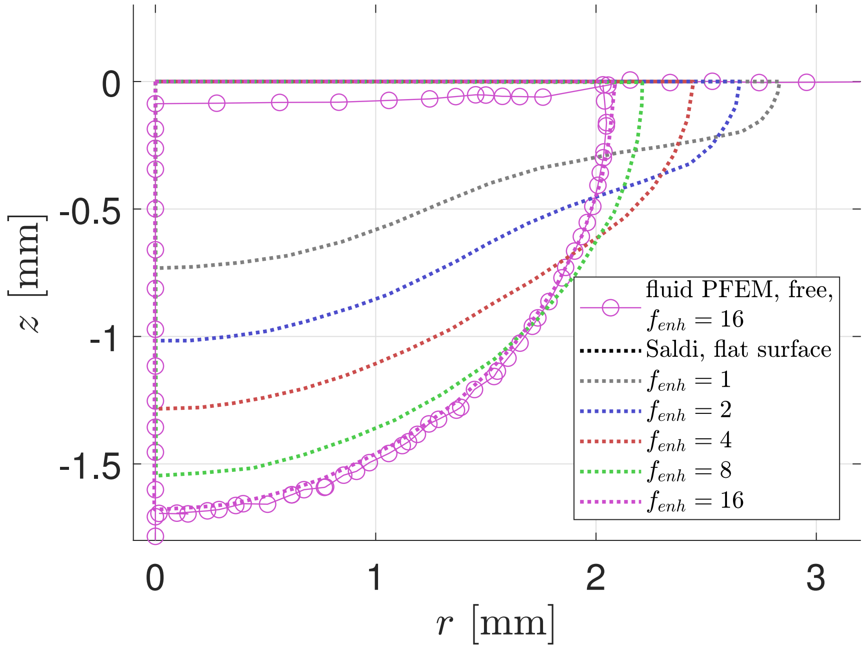
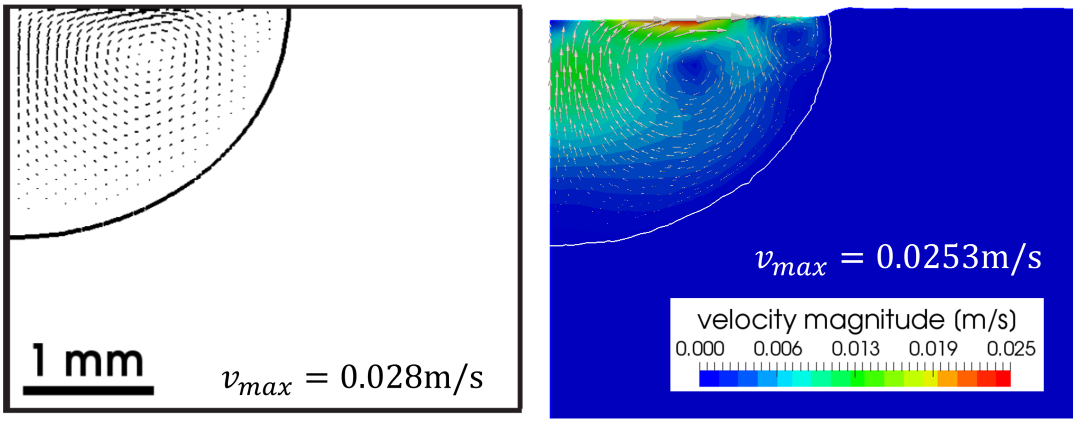


FIGURE 4.18: Melt pool dimensions after $t = 5\,\mathrm{s}$ using purely fluid PFEM with the Carman-Kozeny equation. Using an enhancement factor $f_{enh} = 16$.

by Saldi are for $f_{enh} = 1$ and $f_{enh} = 2$, which cannot be replicated here due to the issues described in section 4.3.6.

One observes that the free surface of the PFEM results has dropped below $y = 0$, meaning that mass has disappeared. The likely cause for this is also described in section 4.3.6. The melt pool size and shape on the other hand agree well with the results of Saldi for the same f_{enh}. Note that the fluid-solid interface in this graph is represented by an isoline at $T = T_m = 1620\,\mathrm{K}$, even though the PFEM simulations use the above mentioned melting interval.

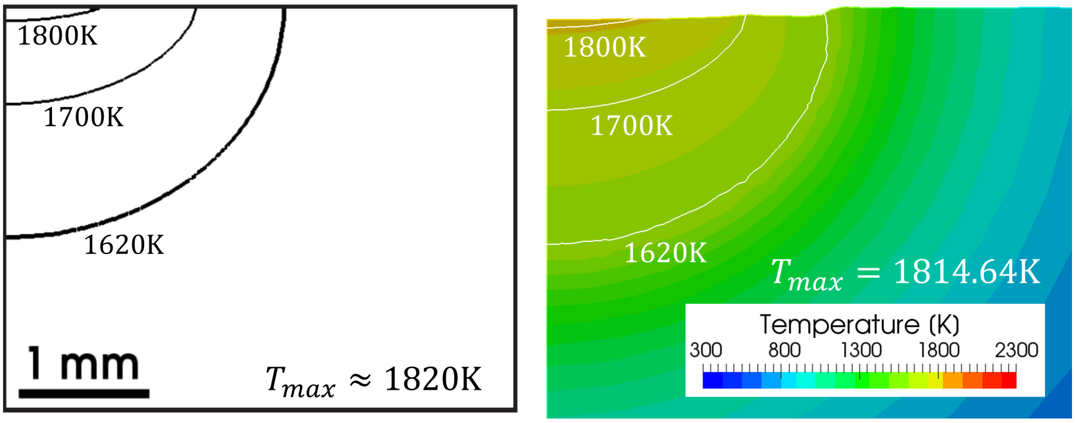


(a) Adapted from Saldi [12]

(b) PFEM

FIGURE 4.19: Velocity vector graph after $t = 5$ s using purely fluid PFEM with the Carman-Kozeny equation. Using an enhancement factor $f_{enh} = 16$.

Fig. 4.19 compares the velocity vector field obtained by Saldi with the one obtained using the present PFEM method. The vector field appears very similar and the maximum velocities occur at the same location and with nearly the same magnitude.



(a) Adapted from Saldi [12]

(b) PFEM

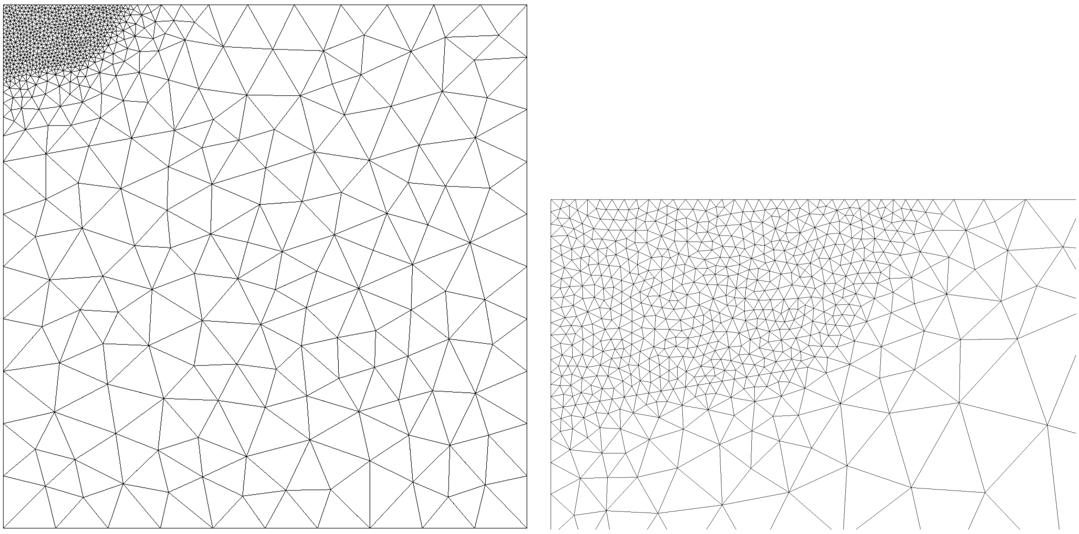
FIGURE 4.20: Temperature isolines after $t = 5$ s using purely fluid PFEM with the Carman-Kozeny equation. Using an enhancement factor $f_{enh} = 16$.

The temperature isolines are compared in fig 4.20 and a very good agreement is observed. Unlike for the maximum velocity, Saldi does not provide the value of the maximum temperature. Based on the distances of the isolines, the temperature at the origin is estimated (see fig. 4.20a) and agrees very well with the one obtained using the PFEM based method.

Melt pool dimensions, velocities and temperatures show an excellent agreement, despite the slight loss of mass.

4.3.3 Unified fluid-solid simulation with a flat surface

When using the unified formulation, the whole domain is initially solid. This means that no mesh modification can be made anywhere and any area that needs to be refined (i.e. where the melting occurs) must be refined a priori. For this reason, the mesh in fig. 4.21 is used instead of the much lighter mesh in fig. 4.17. With a surface that is remaining flat (i.e. with a sliding boundary), there are no problems related to the free surface deformation and a lower f_{enh} can be reached. For $f_{enh} = 8$, a time step $\Delta t = 0.5\,\mathrm{ms}$ is used and for $f_{enh} = 16$, a time step $\Delta t = 2.0\,\mathrm{ms}$ is used. While the fluid behavior remains unchanged (see table 4.3), the solid is chosen to have the properties listed in table 4.4. Thermally, the fluid and solid in the unified approach here should behave identical to the material in the purely fluid approach, while mechanically the modeling of the solid behavior is obviously very different. The observable effect of the choice of solid material modeling should nonetheless remain minimal. For the transition from solid to liquid, it is assumed that the bulk modulus K used for the solid remains constant throughout the mushy zone and the fluid. The shear modulus G is assumed to linearly decrease with the liquid fraction f_l, such that $G = 0$ when a Gauss point is fully liquid. As usual, the mushy zone is not treated as a "solid" and therefore does not build up any residual stresses. These assumptions are made due to the absence of the required data and because the same approach resulted in a smooth transition in section 4.2 (TIG welding).



(a) Whole mesh. Largest edge length $\max(l_e) = 1.5\,\mathrm{mm}$.

(b) Zoom on refined area. Smallest edge length $\min(l_e) = 0.16\,\mathrm{mm}$.

FIGURE 4.21: Mesh used for unified fluid-solid approach. The entire area that may melt must be refined a priori so that the mesh is fine around the phase transition front, since no refinement is allowed in the solid.

The resulting melt pool shape and dimensions are shown in fig. 4.22 and an excellent agreement with the literature is obtained. Fig. 4.23 shows the temperature distribution around the melt pool for two different enhancement factors.

TABLE 4.4: Solid material properties used.

Property	Symbol	Value	Unit
Viscosity	μ	0	
Heat capacity	c_p	627.00	$\mathrm{J\,kg^{-1}\,K^{-1}}$
Conductivity	k	22.9	$\mathrm{W\,m^{-1}\,K^{-1}}$
Density	ρ	8100	$\mathrm{kg\,m^{-3}}$
Thermal expansion coeff.	α_V	0	$\mathrm{K^{-1}}$
Thermal diffusivity	α	4.51×10^{-6}	$\mathrm{m^2\,s^{-1}}$
Poisson's ratio	ν	0.2857	
Young's modulus	E	2.057×10^{10}	Pa
Bulk modulus	K	1.6×10^{10}	Pa
Shear modulus	G	8.0×10^{9}	Pa

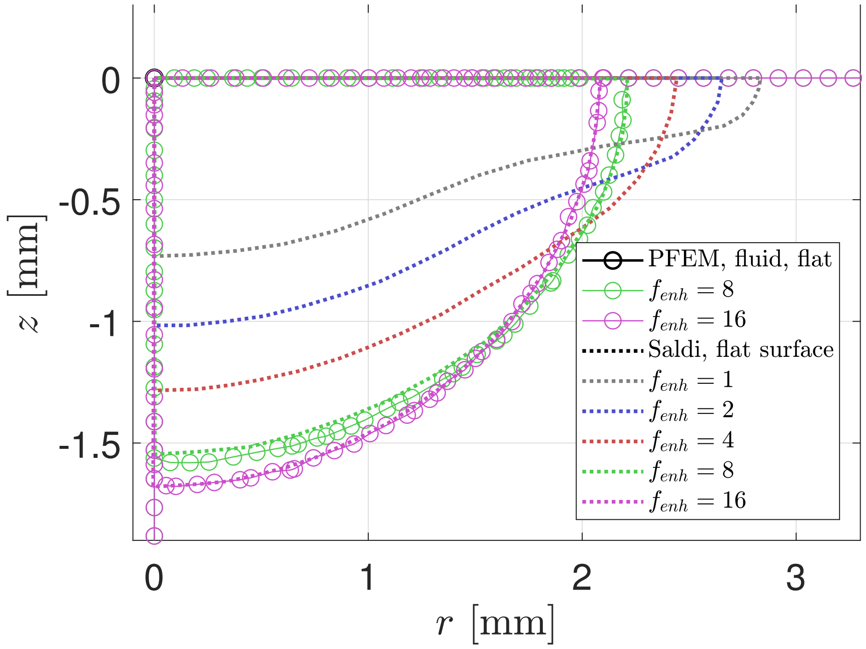
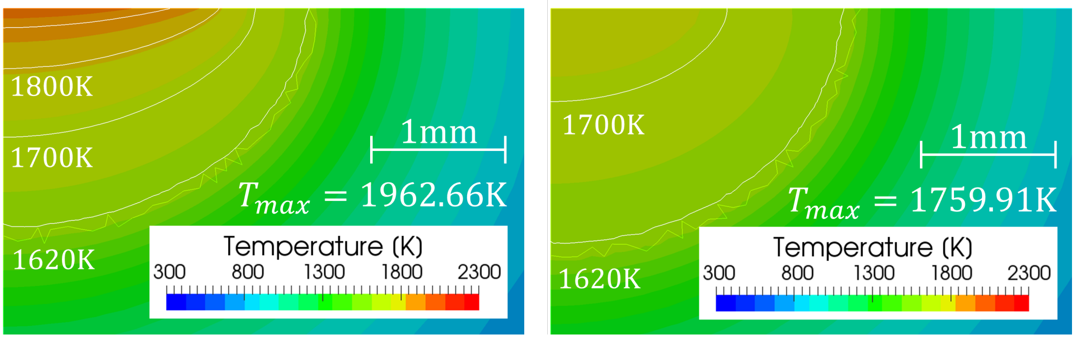


FIGURE 4.22: Melt pool dimensions after $t = 5\,\mathrm{s}$ using the unified fluid-solid formulation of the PFEM with a flat surface. Using an enhancement factors $f_{enh} \in [8, 16]$.

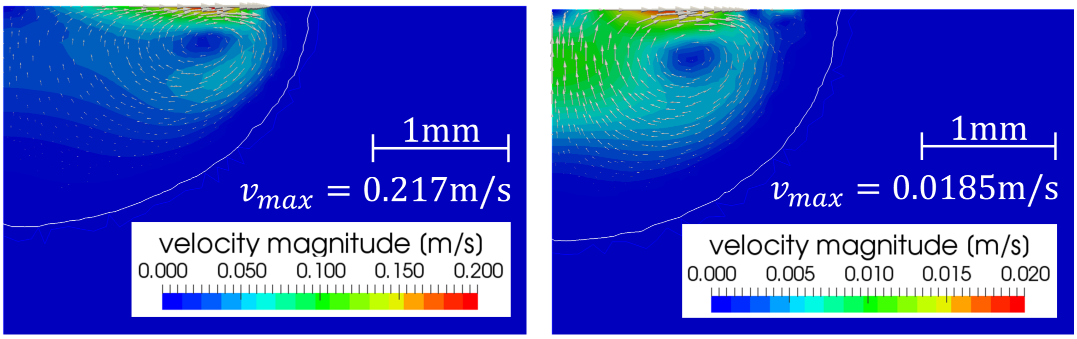
First, comparing the maximum temperature obtained with the unified PFEM approach for the two enhancement factors $f_{enh} = 8$ and $f_{enh} = 16$ (figs. 4.23a and b respectively), a decrease in temperature is observed when f_{enh} is doubled. This is due to the (artificially) increased thermal diffusivity, which allows the heat to flow away from the laser heat source at a higher rate.

Second, comparing the temperature reached for $f_{enh} = 16$ using the unified PFEM approach (fig. 4.23b) with the purely fluid PFEM approach (fig. 4.20b), some slight differences are observed. The unified approach produces a maximum temperature that is lower by 3%. This relative difference sounds very small, but the effect can be quite significant in a purely temperature-driven test case like this one. The reason for this difference is currently unknown and requires further investigation.



(a) $f_{enh} = 8$ (b) $f_{enh} = 16$

FIGURE 4.23: Temperature isolines obtained after $t = 5\,\mathrm{s}$ with the unified fluid-solid PFEM approach.



(a) $f_{enh} = 8$ (b) $f_{enh} = 16$

FIGURE 4.24: Velocity vector field obtained after $t = 5\,\mathrm{s}$ with the unified fluid-solid PFEM approach. Note the different scaling in a and b.

Again, comparing different enhancement factors using the same simulation method, the most noticeable effect of the different temperatures (and therefore also the temperature gradients) is the effect on the flow velocity. Comparing figs. 4.24a and 4.24b shows a qualitatively similar velocity field, but with the magnitude decreased by one order of magnitude when doubling f_{enh}. It is also interesting to note that the location of the maximum velocity and in fact the entire main vortex shift towards the center, as f_{enh} is increased. This is, again, due to the lower temperature, where in fig. 4.24b the counter-rotating secondary vortex is larger, pushing the main vortex to the left. This shift is due to the temperature of the change of the sign of the Marangoni effect (see fig. 4.15) being located further towards the center of the melt pool.

Comparing the velocity field for $f_{enh} = 16$ obtained with the unified fluid-solid formulation (fig. 4.24b) with the one obtained with the purely fluid approach, a significant decrease in the velocity magnitude is observed, around 27% for maximum velocity. This is the aforementioned significant effect that a reduction in peak temperature by 3% can cause. It should now be clear that the reason for the reduced temperature using the unified formulation must be found. Other than that, it is evident that the unified formulation produces good results, when compared to the reference.

4.3.4 Unified fluid-solid simulation with a free surface

This test case is exactly as the previous one, with the same mesh, same time step and same material parameters. The only difference is the free surface that can deform instead of remaining flat. Other than that, an external pressure $p_{ext} = 1 \times 10^5\,\mathrm{Pa}$ is applied on the free surface, which also serves as the reference pressure $p_{ref} = p_{ext}$, such that the density $\rho_{ref} = \rho(p_{ref}, T_0) = 8100\,\mathrm{kg\,m^{-3}}$. The initial pressure $p_0 = p_{ext} - \rho_{ref}\, g\, y$ is applied in the whole domain. Furthermore, normal component of the surface tension is introduced, contrary to Saldi, who only takes the tangential component into account. With a relatively low magnitude of $\gamma = 0.005\,\mathrm{N\,m^{-1}}$ (Pitschneder et al. [81] cite for the same steel a value of $\gamma \approx 1.5\,\mathrm{N\,m^{-1}}$), the overall behavior is not expected to be significantly altered, but the stabilizing effect is observed in the present work. It was, however, noted that a higher value of γ does indeed cause oscillations, where a better resolution in space (at the surface) and time might become necessary.

For $f_{enh} = 16$ the melt pool dimensions in fig. 4.25 are obtained. Once again, the free surface approach incurs a slight mass loss, but the melt pool dimensions agree very well with the expected ones.

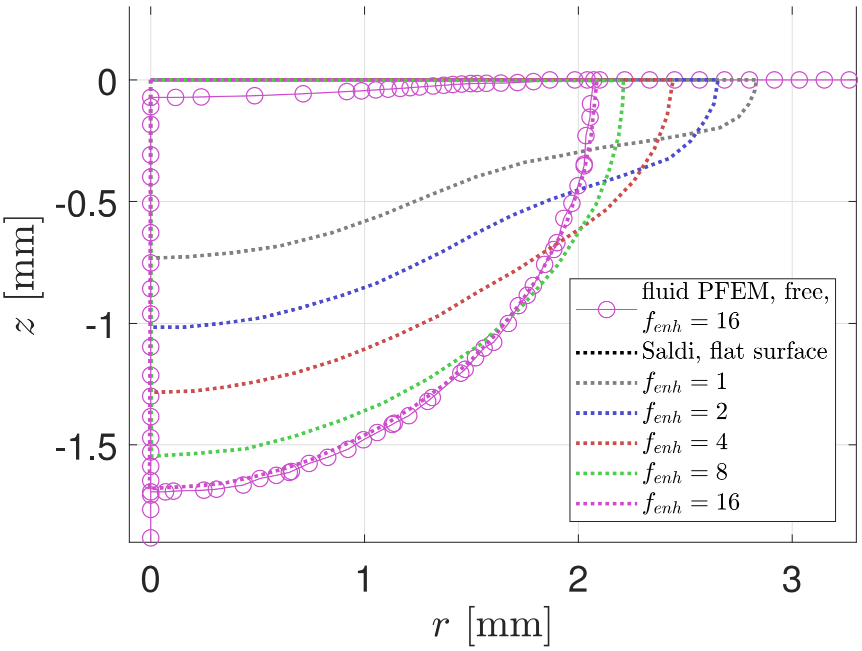
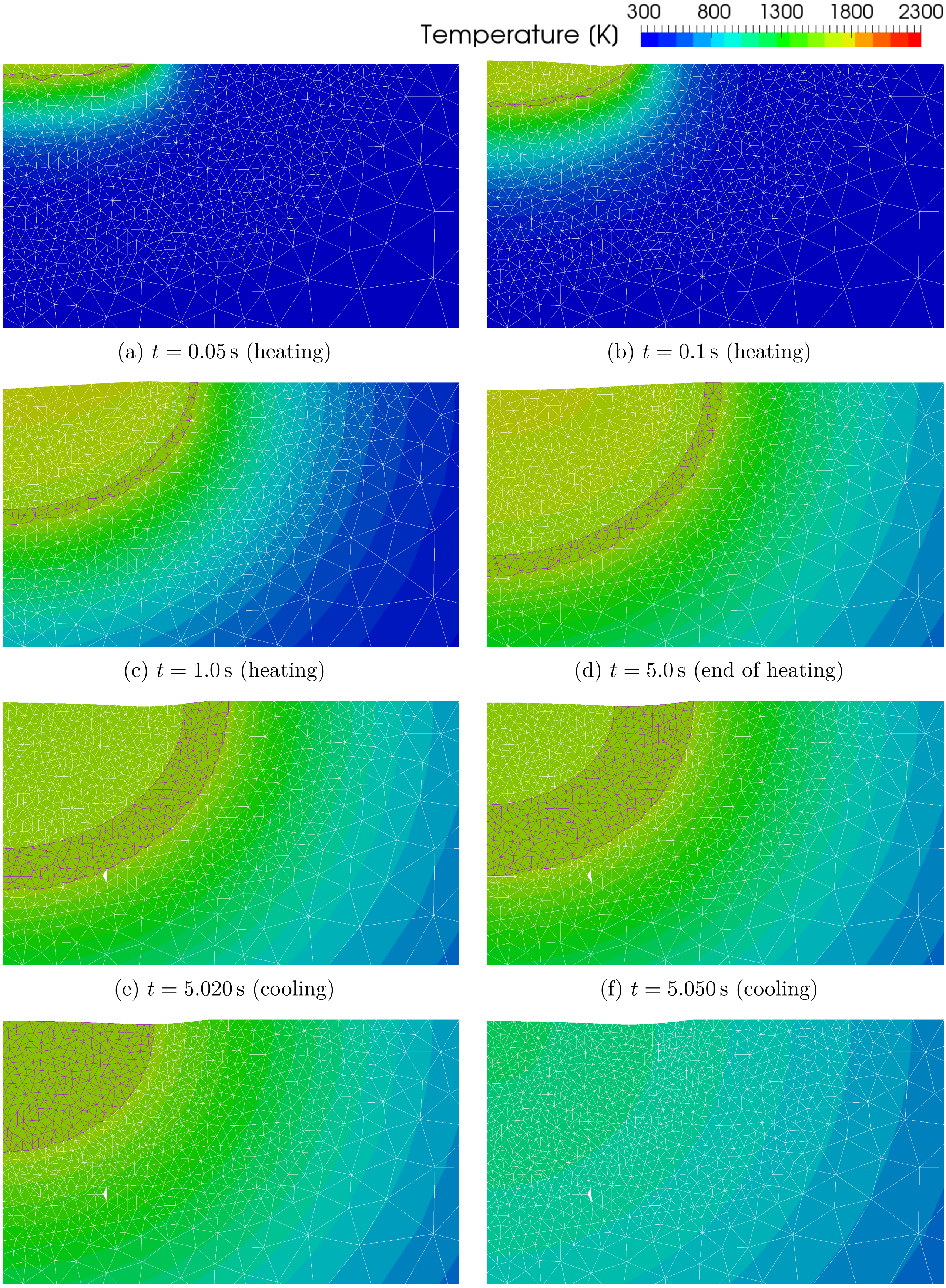


FIGURE 4.25: Melt pool dimensions after $t = 5\,\mathrm{s}$ using the unified fluid-solid formulation of the PFEM with a free surface. Using an enhancement factors $f_{enh} \in [8, 16]$.

Finally, fig. 4.26 shows the temperature evolution for $f_{enh} = 16$ during the heating and cooling period. The melting front (approximated as a melting interval, see above) is slowly advancing into the solid material, expanding the melt pool. When the heating is switched off, the melting front retreats quickly, as the heat flows from the hot area and the melt pool freezes. At the end of the cooling phase as given by Saldi, the area previously heated by the laser still remains at a very high temperature $(\max(T(t_{end})) = 1102.77K)$.

Fig. 4.27 shows the resulting equivalent von Mises stress $\bar{\sigma}_{VM}$ evolving over time. Two observations can be made immediately. Firstly, the melt pool has $\bar{\sigma}_{VM} \approx 0$ because only the viscous shear stresses contribute towards $\bar{\sigma}_{VM}$ and these are comparably small. Secondly, the entire solid region shows the same $\bar{\sigma}_{VM} \approx 6 \times 10^4\,\mathrm{Pa}$, which is simply the effect of the external pressure p_{ext} being applied. Since there is no thermal expansion and no other significant forces acting on the fluid, $\bar{\sigma}_{VM}$ is not particularly interesting.



(a) $t = 0.05\,\mathrm{s}$ (heating)

(b) $t = 0.1\,\mathrm{s}$ (heating)

(c) $t = 1.0\,\mathrm{s}$ (heating)

(d) $t = 5.0\,\mathrm{s}$ (end of heating)

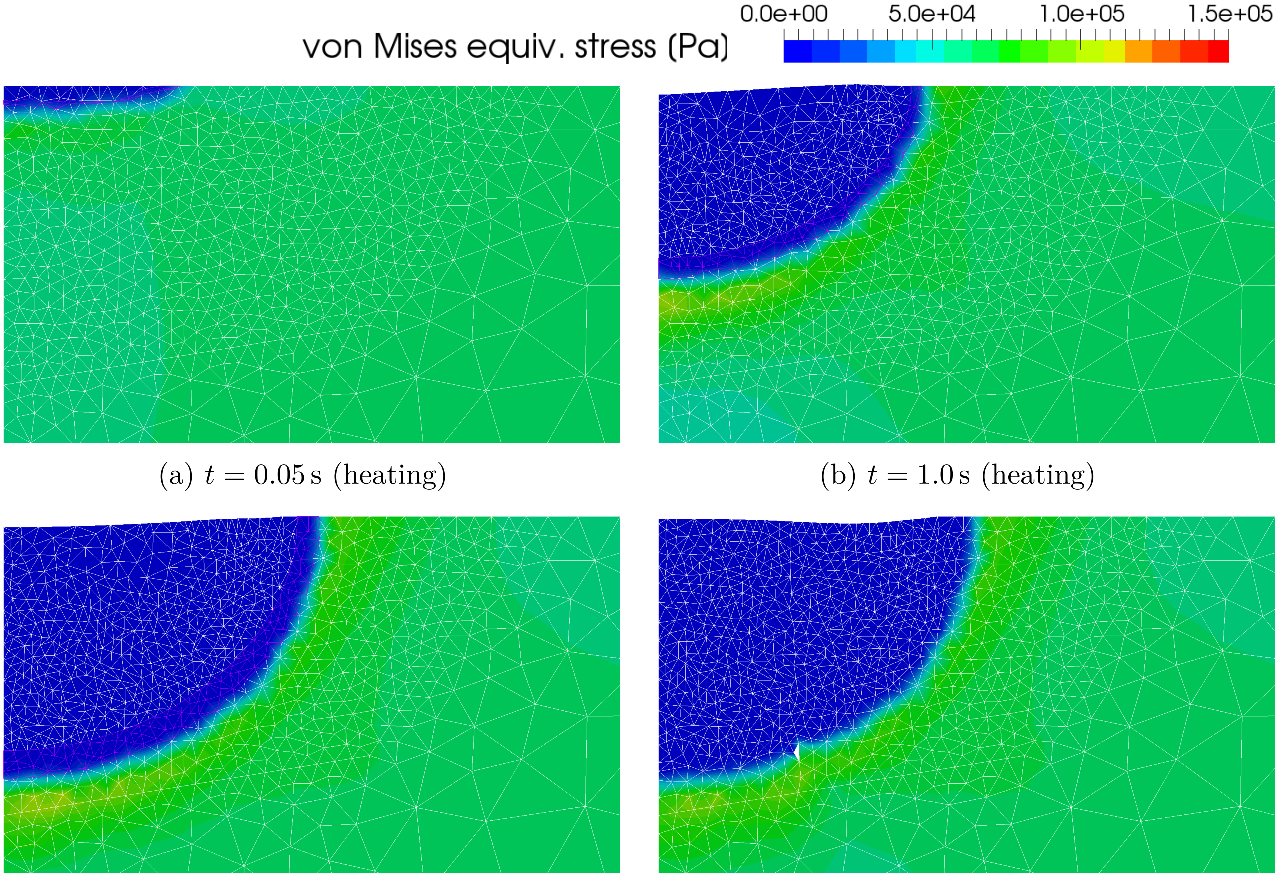
(e) $t = 5.020\,\mathrm{s}$ (cooling)

(f) $t = 5.050\,\mathrm{s}$ (cooling)

(g) $t = 5.120\,\mathrm{s}$ (cooling)

(h) $t = 5.5\,\mathrm{s}$ (end of simulation)

FIGURE 4.26: Temperature evolution around the melt pool over time for $f_{enh} = 16$. Phase change interval shown with dark purple mesh lines. Note that one element has been falsely deleted in the solid region, as the cooling begins (e).



(a) $t = 0.05\,\mathrm{s}$ (heating)

(b) $t = 1.0\,\mathrm{s}$ (heating)

(c) $t = 5.0\,\mathrm{s}$ (end of heating)

(d) $t = 5.5\,\mathrm{s}$ (end of simulation)

FIGURE 4.27: Residual stress (showing the equivalent von Mises stress $\bar{\sigma}_{VM}$) build-up around the melt pool over time for $f_{enh} = 16$. Phase change interval shown with dark purple mesh lines.

4.3.5 Unified fluid-solid simulation with a free surface and thermal expansion

The previous test case for $f_{enh} = 16$ is repeated for a shorter duration with a laser heating period $t_l = 1.0\,\mathrm{s}$ and a cooling period that lasts until $t_{end} = 1.5\,\mathrm{s}$. The length of the cooling time $t_{end} - t_l = 0.5\,\mathrm{s}$ is chosen to agree with that of Saldi's original test cases. The same material as before is reused, but with an added thermal expansion coefficient $\alpha_V = 1 \times 10^{-5}\,\mathrm{K}^{-1}$, which is of a realistic order of magnitude for steel. All other settings remain the same.

The temperature evolution is found in fig. 4.28 and it behaves similar to the previous test case: The melting front advances slowly, as the laser heats and then retreats rapidly once the laser is deactivated. During the heating period, the temperatures and temperature gradients are the highest around the melt pool, within the solid region. Inside the melt pool, the temperature is more homogeneous due to the convective heat transfer and due to the artificially increased ($f_{enh} = 16$) heat conduction.

At the end of the heating period at $t = t_l$ the highest temperature is recorded $\max(T(t = t_l)) = 1795.08\,\mathrm{K}$ at the center of the melt pool surface. The highest temperature at the end of the cooling period $\max(T(t = t_{end})) = 783.91\,\mathrm{K}$ is recorded at the same point.

During the first milliseconds of the cooling period, the temperature around the remaining melt pool remains relatively high in the solid because the freezing of the melt pool supplies the latent heat, that was formerly absorbed, back into the material. Only when the melt pool has frozen entirely, the temperature in the solid region begins to drop significantly. While radiation and convection at the free surface remove some of the heat ($< 7\%$), the majority of the heat from the hot region (yellow to green) flows into the much colder bulk (blue) of the material ($> 93\%$).

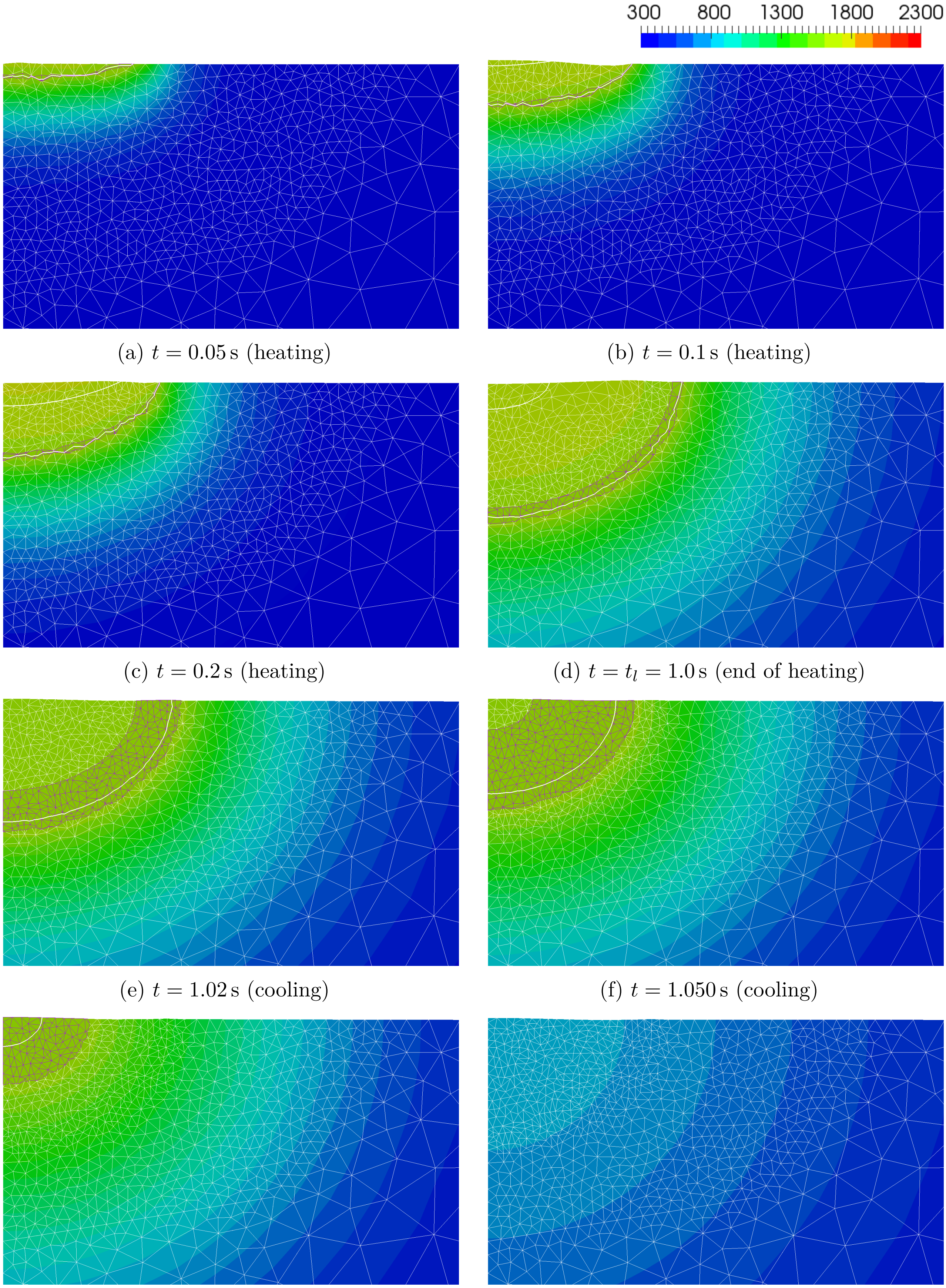
Fig. 4.28 also shows the evolution of the free surface deformation, although it is very mild with the high enhancement ($f_{enh} = 16$). The free surface morphology at the melt pool has been isolated, magnified and exaggerated in fig. 4.29 for better visibility. The highest elevation in the melt pool appears to move from the melt pool center (left) towards the edge of the melt pool (right) over time during the heating, which is the effect of the sign change of the Marangoni effect at $T_{crit} = 1700.5\,\mathrm{K}$ (see fig.4.15). Effectively, the fluid is drawn to the point at the surface where $T = T_{crit}$. The highest point has risen by $\Delta z(t = t_l) = 26.4\,\mathrm{\mu m}$ and is located at $r = 1.30\,\mathrm{mm}$, where the temperature is $T = 1659.77\,\mathrm{K}$, just below T_{crit}.

However, when the heating period ends, the point where $T = T_{crit}$ moves back towards the center, while the melt pool rapidly freezes from the edge towards the center. The final configuration after freezing shows the highest point near the center ($r = 0.065\,\mathrm{mm}$) with an elevation of $\Delta z = 19.3\,\mathrm{\mu m}$.

The evolution of the stresses $\sigma_{rr}, \sigma_{zz}, \sigma_{rz}, \bar{\sigma}_{VM}$ is depicted in figs. 4.30 - 4.33. In general, stresses increase in magnitude during the heating period, especially around the melt pool, where the temperature and temperature gradients are highest. During the cooling period, there is no common tendency for the stress components. The liquid melt pool has uniformly $\sigma_{rr} = \sigma_{zz} = -p_{ext} = -1 \times 10^5\,\mathrm{Pa}$ at all times.

Fig. 4.30 shows that the normal stress in radial direction σ_{rr} is very small and slightly negative (light red to dark red) throughout most of the solid region because the thermal expansion causes a compression. For $y > -h_m$, where h_m is the depth of the melt pool, σ_{rr} remains comparably small because the melt pool allows some radial expansion, which leads to the material being less constrained there. Conversely, below the melt pool ($y < -h_m$) a distinct region of negative stress with a much higher magnitude (blue) is observed. The temperature is very high there and the material is constrained and cannot expand in the radial direction. This leads to this region of intense compression caused by the more constrained thermal expansion ($\min(\sigma_{rr}(t_l)) = -1.107 \times 10^8\,\mathrm{Pa}$).

During the cooling phase, the magnitude of this hot-spot of tensile stress reduces to almost half its magnitude ($\min(\sigma_{rr}(t_{end})) = -6.994 \times 10^7\,\mathrm{Pa}$). Interestingly, the outline of the hemispherical maximum extent of the melt pool can still be seen in the stress distribution after cool down.



(a) $t = 0.05\,\mathrm{s}$ (heating)

(b) $t = 0.1\,\mathrm{s}$ (heating)

(c) $t = 0.2\,\mathrm{s}$ (heating)

(d) $t = t_l = 1.0\,\mathrm{s}$ (end of heating)

(e) $t = 1.02\,\mathrm{s}$ (cooling)

(f) $t = 1.050\,\mathrm{s}$ (cooling)

(g) $t = 1.1\,\mathrm{s}$ (cooling)

(h) $t = t_{end} = 1.5\,\mathrm{s}$ (end of simulation)

FIGURE 4.28: Temperature evolution around the melt pool over time for $f_{enh} = 16$ with thermal expansion. Phase change interval shown with dark purple mesh lines. White lines are isotherms at $T = 1620\,\mathrm{K}$ and at $T = 1700\,\mathrm{K}$.

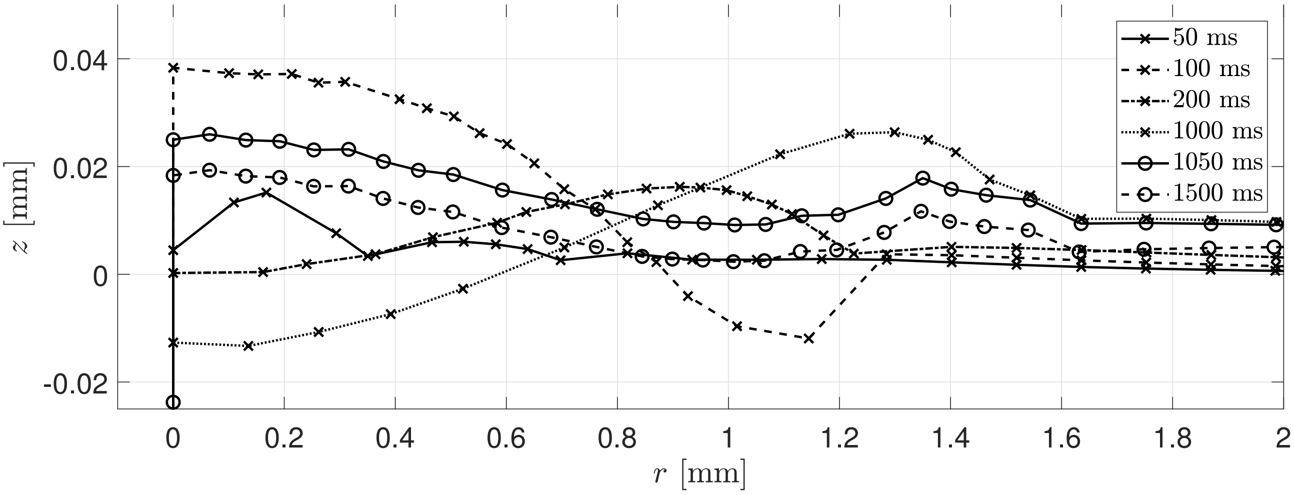


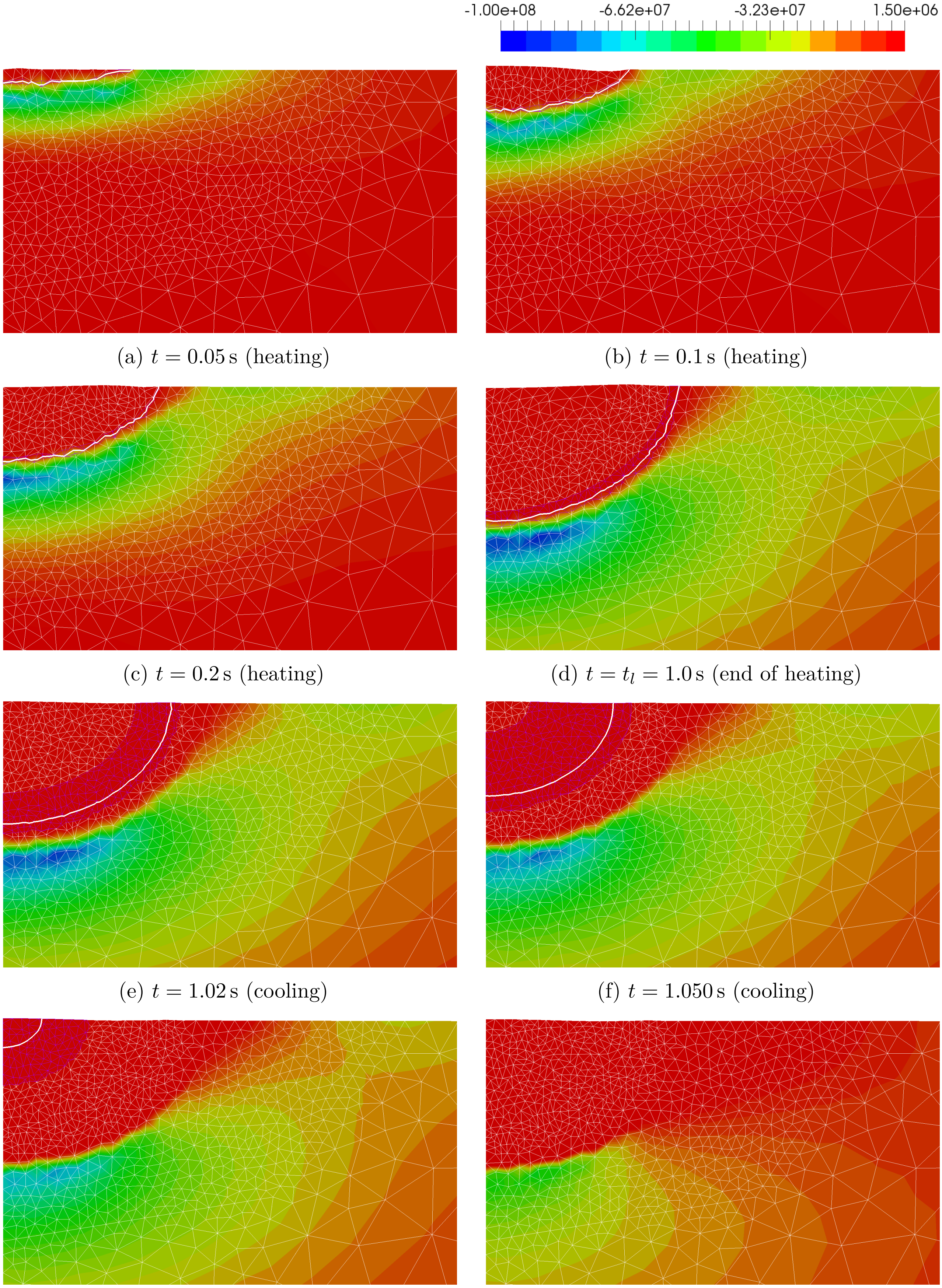
FIGURE 4.29: Evolution of the free surface morphology, zoomed on the melt pool.

The most notable observation is that the stresses increase as the temperature increases due to thermal expansion, and that they reduce again when cooling down.

The evolution of the normal stress component in the axial direction σ_{zz} over time is shown in fig. 4.31. The the same general observation as before can be made that most of the solid region has a low magnitude (light olive green), while a hot-spot of very high (negative) magnitude is found around the melt pool (dark blue). σ_{zz} is negative there, indicating a compressive stress in the z-direction. Surrounding this area of large compressive stress is a diffuse area of low magnitude tensile stress (orange). This can be explained by the highest temperature near the melt pool leading to a more intense thermal expansion of the material. However, the higher thermal expansion near the melt pool is constricted (thus under compression) by the bulk of the solid material around it (which is consequently under tension). Near the free surface of the solid region, $\sigma_{zz} \to 0$ because the material is free to expand there, only restricted by the external pressure p_{ext} applied there. The highest magnitude compressive stress at the end of the heating period is found to be $\min(\sigma_{zz}(t = t_l)) = -2.567 \times 10^7$ Pa, while the highest tensile stress is only $\max(\sigma_{zz}(t = t_l)) = 3.815 \times 10^6$ Pa.

More interestingly, a distinct build-up of tensile stress (dark red) is observed at the root of the former melt pool during cool down. There, the solid material wants to contract, but is once again restricted by the surrounding solid material, leading to a positive offset of the stress in that region as the material cools. The highest magnitude of tensile stress at the end of the cooling period is found to be $\max(\sigma_{zz}(t = t_{end})) = 1.273 \times 10^7$ Pa, while the large compressive stress has reduced to $\min(\sigma_{zz}(t = t_{end})) = -1.969 \times 10^7$ Pa.

Once again, the outline of the maximum extend of the melt pool can be clearly seen in the distribution of σ_{zz}. A new and unexpected observation is made when examining the area around the phase front in figs. 4.31a - c, where localized hot-spots of large positive stress are found, which each last only for one or two time steps. Currently, no obvious cause could be determined and it is likely a numerical artifact related to the creation or removal of the interface.



(a) $t = 0.05\,\mathrm{s}$ (heating)

(b) $t = 0.1\,\mathrm{s}$ (heating)

(c) $t = 0.2\,\mathrm{s}$ (heating)

(d) $t = t_l = 1.0\,\mathrm{s}$ (end of heating)

(e) $t = 1.02\,\mathrm{s}$ (cooling)

(f) $t = 1.050\,\mathrm{s}$ (cooling)

(g) $t = 1.1\,\mathrm{s}$ (cooling)

(h) $t = t_{end} = 1.5\,\mathrm{s}$ (end of simulation)

FIGURE 4.30: Evolution of σ_{rr} around the melt pool over time for $f_{enh} = 16$ with thermal expansion. Phase change interval shown with dark purple mesh lines. Magnitude is given in Pascal. The white line is an isotherm at $T = 1620\,\mathrm{K}$.

More work is needed to understand the origin of these hot-spots.

The evolution of the in-plane shear stress component σ_{rz} is given in fig. 4.32. Most of the solid domain remains at $\sigma_{rz} \approx 0$ (light olive green) throughout the heating and cooling phase. In the vicinity of the melt pool on the other hand, a region of high (negative) shear stress is observed (dark blue). The magnitude reduces near the axis ($r = 0$) and at the free surface ($z = 0$) and has its highest value somewhere between. It is noteworthy that the magnitude reached by the shear stress is of similar magnitude as those reached by any of the normal stresses. The largest (negative) magnitude is $\min(\sigma_{rz}(t = t_l)) = -2.906 \times 10^7\,\text{Pa}$. A larger region of lower magnitude positive shear stress is found further away from the melt pool below the region of negative shear stress. The maximum (positive) magnitude there is $\max(\sigma_{rz}(t = t_l)) = 4.568 \times 10^6\,\text{Pa}$.

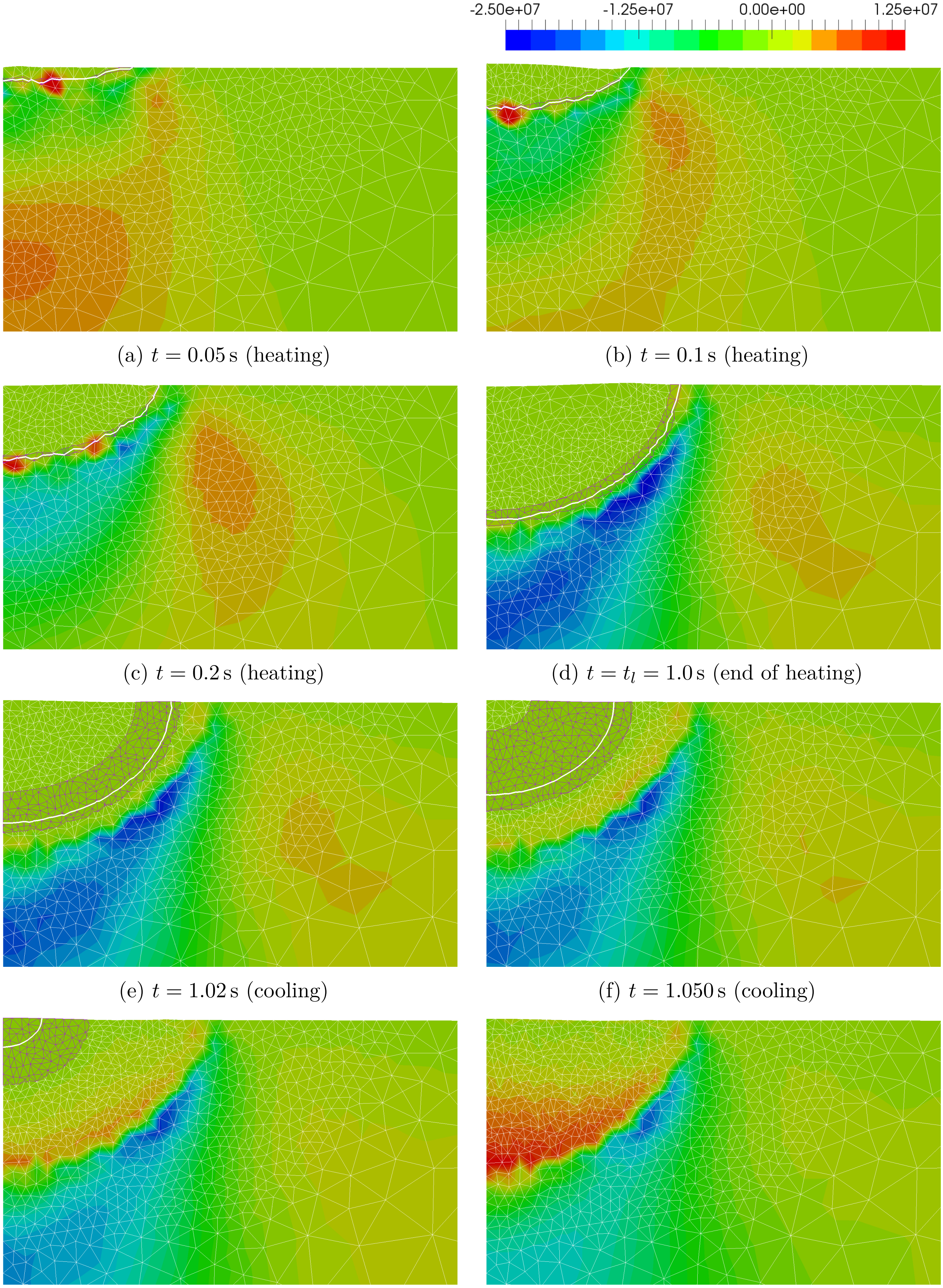
In the liquid melt pool $\sigma_{rz} \approx 0$ because viscous shear stresses are much smaller than the elastic shear stresses in the solid.

When cooling down, the area around melt pool that is not at the axis or at the free surface is positively offset by about $2 \times 10^7\,\text{Pa}$. At the end, the former melt pool edge exhibits a maximum shear stress $\max(\sigma_{rz}(t = t_{end})) = 7.890 \times 10^6\,\text{Pa}$ and the part where previously the minimum stress was found has reduced to $\min(\sigma_{rz}(t = t_{end})) = -2.494 \times 10^7\,\text{Pa}$.

Finally, the equivalent von Mises stress $\bar{\sigma}_{VM}$ is shown in fig. 4.33 evolving over time. First, the liquid melt pool has $\bar{\sigma}_{VM} \approx 0$ because the viscous stresses are low and the pressure does not factor into $\bar{\sigma}_{VM}$. As some sort of combination of the normal and shear stresses, $\bar{\sigma}_{VM}$ is particularly high in the solid around the melt pool, during the heating period. The highest value reached is $\max(\bar{\sigma}_{VM}(t = t_l)) = 8.428 \times 10^7\,\text{Pa}$ below the root of the melt pool, i.e. at $r = 0$. To put this into perspective, assuming that steel typically has a yield strength $\sigma_y > 2 \times 10^8\,\text{Pa}$, plasticity would not be reached in this scenario, with the chosen thermal expansion coefficient.

As the melt pool freezes and the solid cools down, the stress distribution smoothens over the entire domain. Within the former melt pool, the equivalent stress increases from near-zero to a range of about $2 \times 10^7\,\text{Pa}$ to $4 \times 10^7\,\text{Pa}$. The area of the previously highest equivalent stress experiences a reduction in magnitude, leaving behind a hollow hemisphere (i.e. a ring in the 2D representation) with an equivalent stress magnitude of $\max(\bar{\sigma}_{VM}(t = t_{end})) = 5.785 \times 10^7\,\text{Pa}$.

In this test case, it was demonstrated that the unified fluid-solid formulation is able to reproduce residual stresses after a laser spot welding process with a cooling period. The residual stress components appear to have a plausible magnitude and distribution around the weld spot, but no stress data for the underlying welding experiment is available to verify the results. Furthermore, a realistic simulation of such a welding process would require plasticity in the solid to be taken into account. Nonetheless, the main goal of showing the unique capabilities of such a unified formulation has been fully met: the initially solid material melts continuously under the localized heat source, introducing stresses due to thermal expansion. Then the molten material re-solidifies as



(a) $t = 0.05\,\mathrm{s}$ (heating)

(b) $t = 0.1\,\mathrm{s}$ (heating)

(c) $t = 0.2\,\mathrm{s}$ (heating)

(d) $t = t_l = 1.0\,\mathrm{s}$ (end of heating)

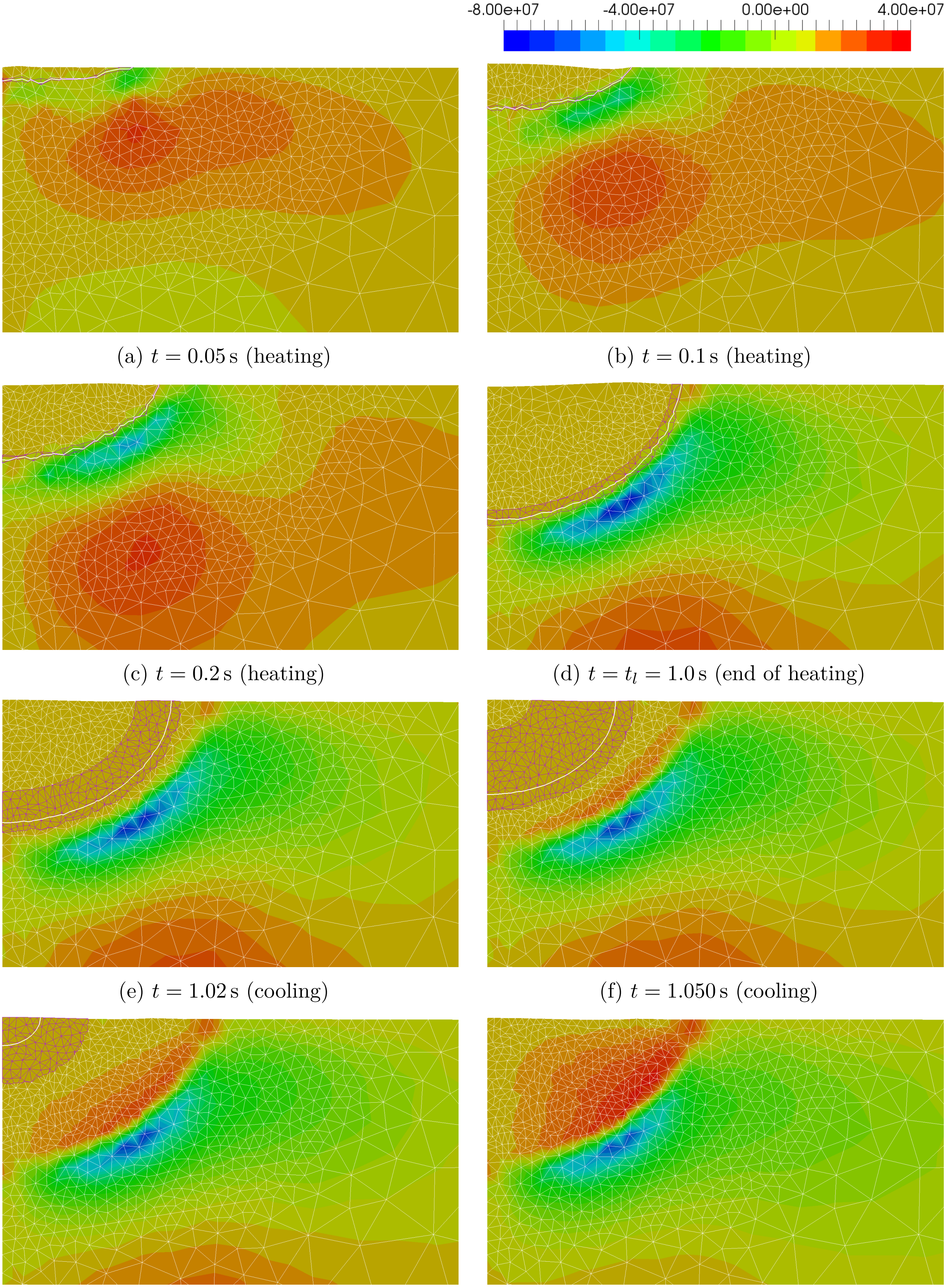
(e) $t = 1.02\,\mathrm{s}$ (cooling)

(f) $t = 1.050\,\mathrm{s}$ (cooling)

(g) $t = 1.1\,\mathrm{s}$ (cooling)

(h) $t = t_{end} = 1.5\,\mathrm{s}$ (end of simulation)

FIGURE 4.31: Evolution of σ_{zz} around the melt pool over time for $f_{enh} = 16$ with thermal expansion. Phase change interval shown with dark purple mesh lines. Magnitude is given in Pascal. The white line is an isotherm at $T = 1620\,\mathrm{K}$.



(a) $t = 0.05\,\text{s}$ (heating)

(b) $t = 0.1\,\text{s}$ (heating)

(c) $t = 0.2\,\text{s}$ (heating)

(d) $t = t_l = 1.0\,\text{s}$ (end of heating)

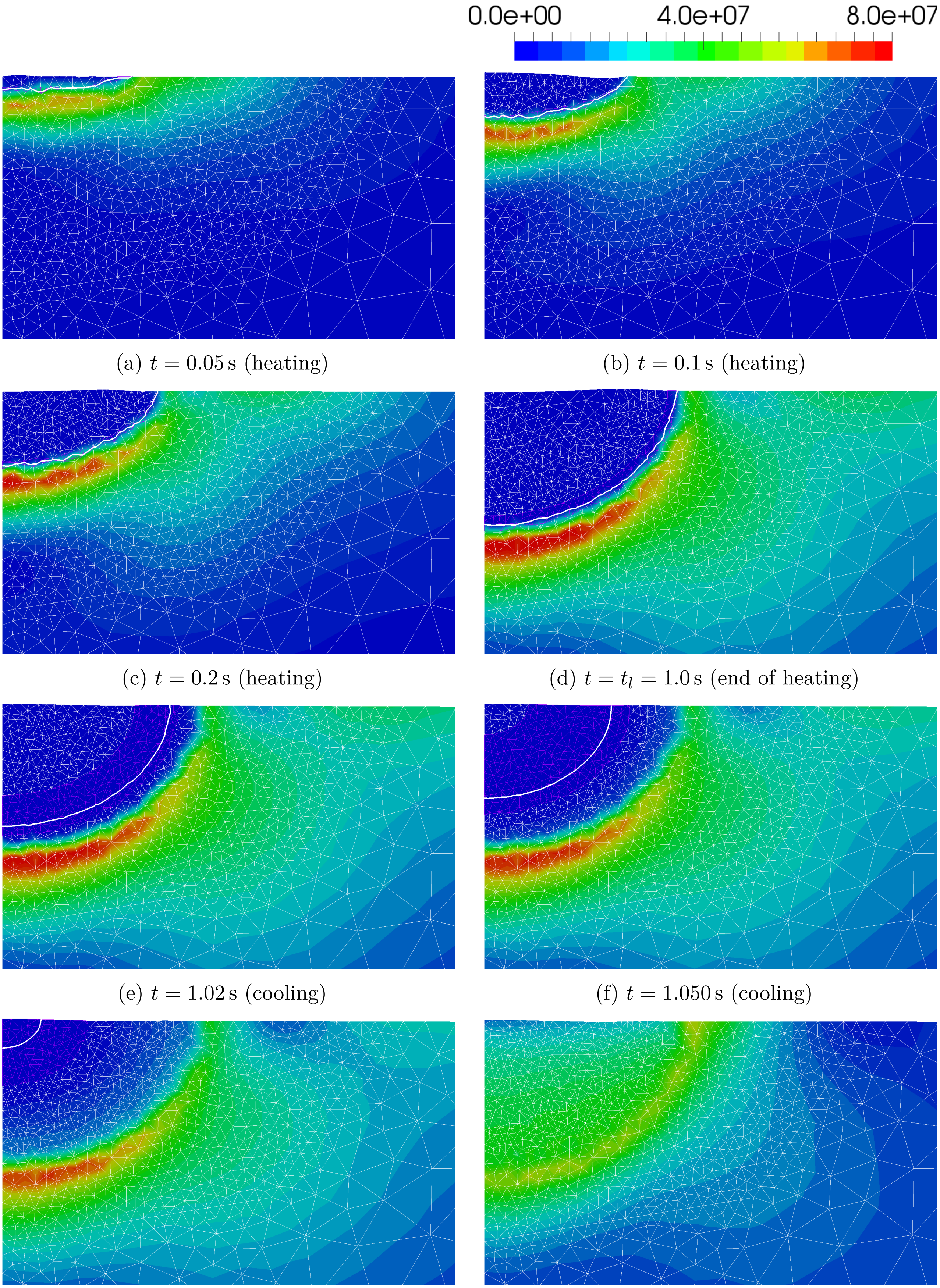
(e) $t = 1.02\,\text{s}$ (cooling)

(f) $t = 1.050\,\text{s}$ (cooling)

(g) $t = 1.1\,\text{s}$ (cooling)

(h) $t = t_{end} = 1.5\,\text{s}$ (end of simulation)

FIGURE 4.32: Evolution of σ_{rz} around the melt pool over time for $f_{enh} = 16$ with thermal expansion. Phase change interval shown with dark purple mesh lines. Magnitude is given in Pascal. The white line is an isotherm at $T = 1620\,\text{K}$.



(a) $t = 0.05\,\mathrm{s}$ (heating)

(b) $t = 0.1\,\mathrm{s}$ (heating)

(c) $t = 0.2\,\mathrm{s}$ (heating)

(d) $t = t_l = 1.0\,\mathrm{s}$ (end of heating)

(e) $t = 1.02\,\mathrm{s}$ (cooling)

(f) $t = 1.050\,\mathrm{s}$ (cooling)

(g) $t = 1.1\,\mathrm{s}$ (cooling)

(h) $t = t_{end} = 1.5\,\mathrm{s}$ (end of simulation)

FIGURE 4.33: Evolution of $\bar{\sigma}_{VM}$ around the melt pool over time for $f_{enh} = 16$ with thermal expansion. Phase change interval shown with dark purple mesh lines. Magnitude is given in Pascal. The white line is an isotherm at $T = 1620\,\mathrm{K}$.

the heat source is switched off and the previously introduced stresses change, leaving behind a fully solid part with a non-zero and non-uniform residual stress distribution, which is a well-known issue in real world welding processes.

4.3.6 Issues encountered in the simulation

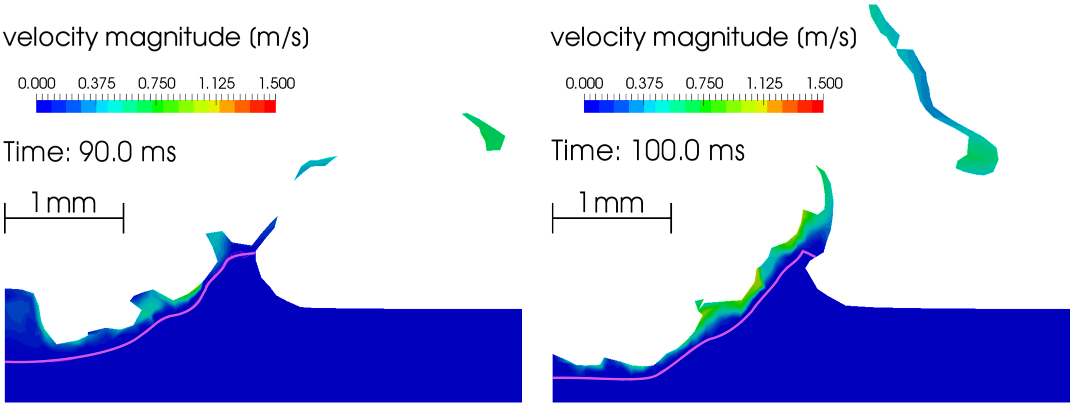
Two problems are encountered that greatly reduce the range of variations of this test case: The cost and the robustness.

Due to the relativly high cost of the PFEM, simulation run times would be very long (days or weeks), if Saldi's spatial and temporal resolution were exactly followed. While this alone does not present a serious issue, the high likelihood of crashes exponentially increases the time taken to investigate this test case. Already expensive simulations need to be restarted many times until a simulation run is completed.

The robustness isuee in this laser spot welding test stems from the Marangoni effect that leads to a strong tangential forcing on the fluid's surface. This causes several problems:

- A higher velocity (even in only a small part of the domain) requires a lower time step, increasing the cost.
- The tangential force leads to a localized acceleration of surface elements. Accelerated elements are stretched and decelerated ones are squished (e.g. when reaching the melt pool edge). This can cause:
  - mesh quality issues, if left untreated;
  - mass conservation errors accumulating due to a large number of mesh operations that attempt to maintain a good mesh quality.
- A constant (and not chaotic) shear flow at the free surface keeps adding the same small mass error over a long time, leading to a large accumulated mass error (see previous point). Normally, small mass errors cancel out in more chaotic free surface deformations in the classic PFEM (see examples in Franci and Cremonesi [42]). It is not clear, how this type of mass error can be avoided.
- With a coarse mesh, the first layer of fluid elements upon melting is pushed away before a circulation sets in (which requires at least 3 layers of fluid elements), see fig. 4.34. A finer resolution in space and time should reduce this issue.

Since the time step requirements reduce when the mesh elements are larger, a slightly coarser mesh than Saldi's is used to reduce the cost. The element size varies between the three variants (enumeration on page 197), but remains $\bar{l}_e > 0.1\,\mathrm{mm}$, therefore at least 5 times coarser than Saldi's mesh. To increase the time step further, the velocities could be decreased. To achieve this, the tests conducted are limited to those with a high enhancement factor $f_{enh} \in \{8, 16\}$. A larger enhancement factor artificially increases the heat conduction and viscosity in the fluid. A more effective conduction reduces the temperature gradient



(a) $t = 90\,\text{ms}$: As each freshly liquefied layer of elements is immediately blown away, no circulation can establish itself.

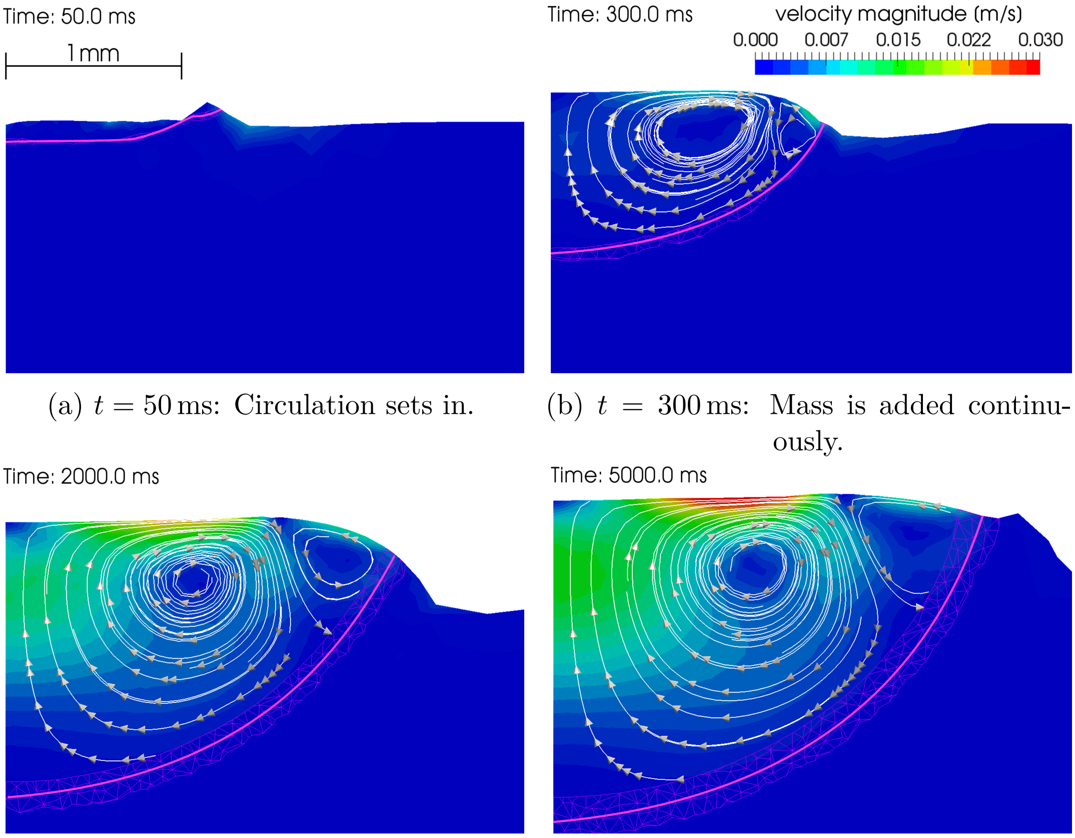
(b) $t = 100\,\text{ms}$: The small amount of newly melted fluid experiences the entire force of the Marangoni effect, which is strong enough to project the fluid from the melt pool into the air.

FIGURE 4.34: The strong Marangoni effects projects the fluid into the air. These results are obtained with $f_{enh} = 2$ and a coarse mesh $\bar{l}_e = 0.1\,\text{mm}$. With both snapshots only a few milliseconds apart, the melt pool is observed to be continually "blown away".

at the surface, which reduces the magnitude of the Marangoni effect. With a reduced Marangoni effect and an artificially increased viscosity the velocities achieved at the surface are significantly smaller, allowing for a larger time step. The reduced velocity also leads to a "calmer" free surface, which in some instances was agitated to the point where droplet are ejected from the melt pool and violent splashing was observed, which is numerically challenging, even for the PFEM (see fig. 4.34).

It was also hypothesized that the severe under-resolution of the free surface could lead to an inaccurate computation of the curvature, especially near the outer edge of the melt pool. This could lead to unrealistic local surface tension normal forces or oscillations. It was, however, observed that early versions of this test did not use a surface tension normal component at all and still the same issue persists. On the contrary, the introduction of the surface tension normal component, which was not featured in the original test by Saldi [12], had a stabilizing effect (as described in section 4.3.4). The surface became calmer, smoother and more tolerant towards the strong accelerations to some degree, but not so much so that it suppressed the spatter in fig. 4.34.

In fig. 4.35 a significant spurious mass gain is observed, and in fig. 4.36, the opposite effect is observed. Both effects are caused by a constant flow (note that the flow pattern is roughly the same in both figures) over a long time, where elements at the free surface are stretched and squished continually. As the algorithms attempt to maintain a good mesh quality, elements are therefore split or recombined, when an element becomes too oblong or narrow respectively. Fig. 4.37 demonstrates where these mesh operations occur. And since the flow



(a) $t = 50\,\mathrm{ms}$: Circulation sets in.

(b) $t = 300\,\mathrm{ms}$: Mass is added continuously.

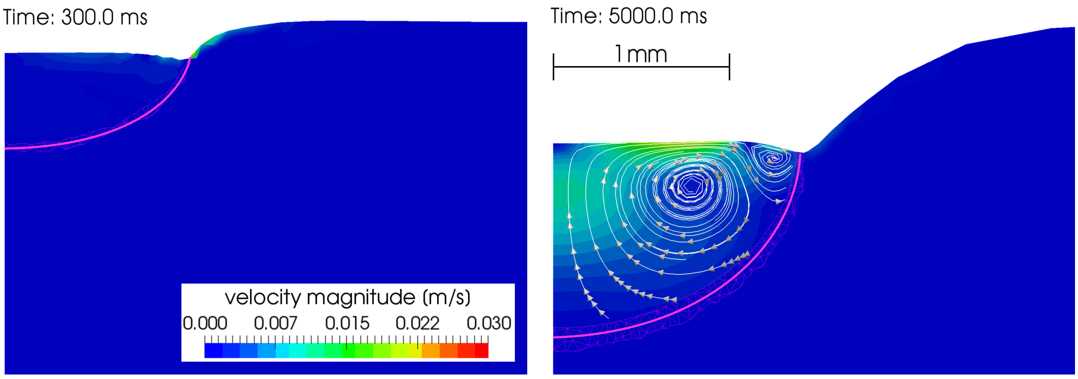
(c) $t = 2000\,\mathrm{ms}$: Accumulated mass has grown.

(d) $t = 5000\,\mathrm{ms}$: By the end a mass gain roughly equal to the mass of the melt pool can be reached.

FIGURE 4.35: The continuous flow causes small mass errors to accumulate and cause considerable mass gain.

does not change qualitatively for thousands of time steps, the same mesh operations are performed again and again at the same location. Whether such a mesh operation adds or removes some mass depends on the circumstances (mainly surface curvature), but also these circumstances, whatever they may be, do not change locally for thousands of time steps. This means that the same error (mass addition or removal) is committed over and over. In classic PFEM applications, the surface deformation is not continuous (i.e. chaotic splashing or sloshing) and mass addition and subtraction are more likely to cancel out, but even there, this problem has been observed (see e.g. Falla et al. [96]).

Since many parameters are involved and since each single error is small, it is difficult to determine how to tune numerical or remeshing parameters such that the mass remains largely constant. In this work, no consistent rule could be established as to what parameter influences this phenomenon in which way. The only tendency that has been observed is that reducing time steps sometimes help, but surprisingly not the increase of the mesh density. Remeshing parameters such as α_{crit} (section 2.7.3), γ_{crit}, ω_{crit} (section 2.7.4) do not deliver consistent trends. Other than the time step, a too large parameter ϵ_{CK} for the



(a) t = 300 ms: With conituous circulation some mass has been lost.

(b) t = 5000 ms: By the end, a mass loss much larger than the mass of the melt pool can be reached.

Figure 4.36: The continuous flow causes small mass errors to accumulate and cause considerable mass loss.

Carman-Kozeny equation, as the one given by Saldi ($\epsilon_{CK} = 1 \times 10^{-3}$) can aggravate the issue. The extreme examples in figs. 4.35 and 4.36 are obtained with that value, while the ones in figs. 4.19 - 4.33 are obtained with ($\epsilon_{CK} = 1 \times 10^{-5}$)

A more permanent solution to this mass conservation issue than to tune remeshing parameters might once again lie in a more sophisticated handling of the free surface. The α-shape algorithm is simple and fast and well suited to handle splashing and closing waves, but not so much for a smooth, but fast moving free surface such as the one encountered in this test case.

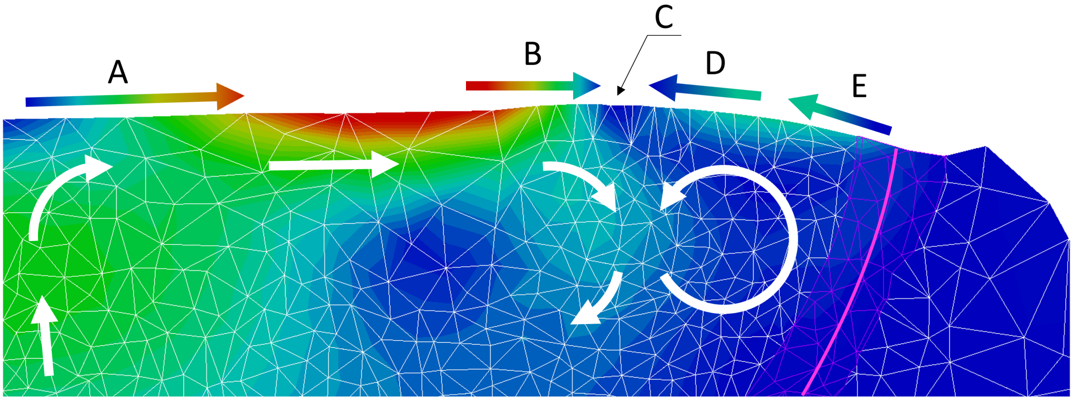


Figure 4.37: Example velocity magnitude contour plot. Nodal acceleration and deceleration at the surface can lead to mass error. White arrows show general flow direction in the bulk. Around A, nodes are accelerated and elements stretched. Around B nodes are decelerated and elements narrowed. At C, the Marangoni effect has a zero magnitude (sign change in fig. 4.15. Around D elements narrow and at E elements stretch again. Elongated elements are eventually split and narrow elements are recombined, depending on meshing parameters α_{crit} (section 2.7.3), γ_{crit}, ω_{crit} (section 2.7.4). Each splitting or merging can cause a small mass error that accumulates.

4.3.7 Summary and conclusion

This section shows a laser spot welding test case, where a cylinder of steel is subjected to a concentrated laser heat source, leading to the partial melting and subsequent re-solidification of the material. The simulation of such a process can be executed with different levels of fidelity, which was done in this case. The aim of this progression from a simpler to a more complex simulation approach is to begin with a verification and step-by-step transition into a demonstration test case. This is to ensure a good confidence in the final results for which no verification is possible.

In the first step, the laser spot welding test case is simulated using a very similar approach to that of Saldi, who published this test case in his doctoral thesis [12] at TU Delft in the group of Prof. Kleijn (Chemical engineering Dept./TP). While Saldi uses a combination of the Eulerian Finite Volume Method (FVM), an interface tracking with the level set method and a modeling of the solid with the Carman-Kozeny equation (see section 2.3.3), our approach uses a combination of the Lagrangian Particle Finite Element Method (PFEM) and the Carman-Kozeny equation for the solid.

Despite some limitations related to the strong Marangoni effect (see section 4.3.6) at the free surface, some of Saldi's results could be reproduced with satisfactory accuracy. This proves that the modeling approach using a Lagragian method works for these types of simulations. It also proves that the modeling of the fluid flow and the heat transfer are correctly handled by the formulation.

In a second step, the newly developed unified fluid-solid formulation replaces the combination of purely fluid formulation with the Carman-Kozeny equation.

This second step is split into two parts, where at first, a flat surface is imposed. While it is a strong assumption that the melt pool surface remains flat throughout a welding process, it allows to make simplifications, especially when working with an Eulerian method, such as Saldi's one. In our case, the flat surface is included because it bypasses the issues related to the free surface with the strong Marangoni effect. This assumption is then removed in the second part of the second step. Both sub-steps maximize the number of results that can be compared with the literature, therefore maximizing the confidence in the verification.

Again, comparisons with the literature yield a good agreement. This proves that the unified fluid-solid formulation produces equally accurate results compared to the classic approach with the fluid solver and the Carman-Kozeny equation.

The third step now deviates from the test case that is described in the literature because thermal expansion is added. Thermal expansion is necessary to produce non-uniform stresses throughout the confined solid material and together with the melting and re-solidification, residual stresses can be generated.

The results presented clearly show plausible stresses that are introduced into the material upon heating by the laser, which do not vanish upon subsequent cooling.

These three steps in combination attempt to prove that the unified fluid-solid formulation integrated into the PFEM allows to model laser spot welding, where phase change, convective flow in the melt pool and residual stresses play an important role. All these coupled physical effects can be taken into account in a single unified solver.

Chapter 5

Summary and conclusions

5.1 Summary

This work presents a novel simulation technique that is able to simulate fluids and solids in the same computational domain with one single solver and is used for phase change problems. This method is based on the Particle Finite Element Method (PFEM). The key ingredient is a unified formulation for fluids and solids.

This work outlines the mathematical formulation of the simulation technique in detail, followed by a verification and finally some test cases that demonstrate the potential of this method. At the end, the strengths and weaknesses of the method are assessed and recommendations for future work are given.

5.1.1 Method

The starting point for this work is the Particle Finite Element Method (PFEM) as presented by Cerquaglia [39]. This is a Lagrangian simulation method for incompressible, isothermal fluid flow, where the fluid is represented by particles and the equations are solved by the Finite Element Method (FEM). This combination of robust FEM algorithms with the flexibility of particles is what makes the PFEM so powerful. It is in fact the reason why PFEM is suitable for both fluids and solids. The classic PFEM is particularly well suited for complex free surface phenomena such as sloshing, breaking waves and many other. This original PFEM code in [39] uses a pressure-velocity formulation, as is often done in Computational Fluid Dynamics (CFD). The code was verified for incompressible, isothermal fluid flow with complex free surface deformation. The main additional capabilities that have been implemented and described in detail in chapter 2 are listed here and then briefly summarized:

- The unified formulation for fluid and solid.
- Extension to thermal and thermo-mechanical problems including phase change.
- An optional interface between fluid and solid, with automatic detection in case of an unknown location of the interface.
- The adaptation of the code to manage fluid and solid status based on the liquid fraction at the Gauss points of the finite elements.

In the existing PFEM code for fluids, the viscous term has been replaced by a **unified formulation** that can represent both elastic solids and fluids. First we recall that viscous stresses of the fluid are a function of strain rate, while elastic stresses of the solid are a function of strain. To unify these two different material behaviors, we first assume that the strain needed for the elastic material can be expressed in terms of strain rate. This can be achieved by splitting the current strain into a current strain increment and an existing reference strain. In a time marching scheme, we can keep the reference strain in memory and add a strain increment to it at the end of each time step, therefore continuously updating the current strain.

We then assume that the strain increment can be expressed as the strain rate integrated over the current time step. Now, the elastic solid has become a function of strain rate, with a reference stress in memory.

With both fluid and solid behaviors expressed as a function of strain rate, one single set of equations can govern both phases. The price to pay is the assumption that the strain rate for any given solid element is constant during a time step, which should require moderate time steps to minimize the error committed due to this assumption, but in practice no problems were observed.

The other important development is **thermal capabilities with latent heat**. Implementing the heat equation with Fourier's law for conduction is straight forward. The advective term is not needed due to the Lagrangian description of motion. The thermal boundary conditions (imposed temperature, imposed heat flux, convection, radiation) are easily implemented.

Difficulty arises where heat equation and Navier-Stokes equations are coupled. Physical phenomena such as thermal expansion and the Marangoni effect have been modeled, which led to a strong dependence of the flow on the temperature distribution. Conversely, the heat sources or surface heat flux that are supposed to model a laser, for example, lead to a strong dependence of the heat flow on the morphology of the surface. It becomes obvious, that the thermal and mechanical problems can be strongly coupled, when such physics are included in a simulation.

The difficulty is further increased with the inclusion of the latent heat term: it is very non-linear and acts very localized, which can cause numerical difficulties. Several techniques (other than a fine mesh) are presented that reduce the difficulty caused by the non-linearity:

- Increase the number of Gauss points temporarily and locally, where phase change occurs.
- Use of a regularization to smoothen discontinuities.
- Iteratively solve non-linear equations with a Newton-Raphson method.
- Use of a line search algorithm to solve non-linear equations faster and suppress oscillations.

The last noteworthy ingredient is the optional **interface**. In a first step, it detects the location of the fluid-solid contact surface using the liquid fraction parameter f_l. The interface is located between elements that are fully solid ($f_l = 0$) and elements that are not fully solid. Secondly, it splits the domain into two disconnected subdomains and then finally re-connects them using Lagrange multipliers. Because the remeshing and the solving of the thermal problem may shift the surface between fluid and solid elements within the original domain, the interface shall be established after the remeshing and the thermal solution.

The purpose of the interface is to produce a pressure discontinuity and therefore reproduce correct stresses in the solid and the fluid. The interface is useful for pure Fluid Structure Interaction (FSI) simulations, where fluid and solid are two distinct materials (e.g. steel and water). For phase change problems with a single material transitioning smoothly from fluid to solid and vice versa, the interface may not be necessary, but more research is needed to confirm this.

Futhermore, an overview over the automatic remeshing and free surface detection of the classic PFEM and some of its extensions are described in chapter 2 as well. The algorithms that are implemented in the code that govern all of the above capabilities are described in detail with many flow charts and illustrations.

5.1.2 Verification

A verification confirms the correct implementation of the equations and algorithms. It tells the user whether the model works as expected. A validation confirms that the simplifying model represents the complex reality sufficiently accurately. It tells the user, if the model is correctly selected to fulfill a specific purpose.

Chapter 3 contains verification test cases, some of which also have a slight validation character for laser welding of L-PBF. However, this work does not aim to validate the modeling of any specific real world process. Chapter 3 should instead be viewed as a basic verification for a completely new modeling method for FSI problems and phase change problems in general.

Due to the large number of physical phenomena involved (see above), the verification is performed step by step, continuously increasing the complexity of the verification tests.

Thermo-fluid material with phase change capabilities

First, the static thermal capabilities are verified on a quasi-1D transient heat conduction test case for which an analytical solution exists. The results agree well, so we assume that the heat equation is solved correctly.

This is followed by several dynamic thermo-fluid tests being performed, where the fluid flow and the heat transfer are coupled: natural convection between two vertical plates and natural convection in a square cavity. The natural convection due to thermal expansion is verified, as a very good agreement with expected results is obtained. Thermal expansion is shown in a third test case, where an open container is filled with a fluid and the rising of a fluid column of a heated fluid is shown. The mass is well preserved as the fluid's density decreases and volume increases as expected.

Then, the Marangoni effect is added to the natural convection test case in the square cavity. This means that two competing forces now drive the flow and interesting vortex patterns are obtained, depending on which effect dominates. These vortices, along with the temperature distribution agree well with the literature.

The handling of the latent heat is verified in a static quasi-1D Stefan problem. The above mentioned combination of techniques to ease the difficulty of the non-linearity of the latent heat term is shown to be able to handle isothermal and non-isothermal phase change accurately.

The next step is to combine the verified thermo-fluid with the verified latent heat capability. This step marks the most important component in the verification of the thermo-fluid with phase change. A large 2D test case is presented where a solid block of gallium is melted by a hot wall on one side, attempting to replicate some experiments. Natural convection in the already melted material causes a flow that leads to a non-uniform heat transfer from the hot wall to the colder phase change front, which begins to curve more and more over time. The front advances faster, where the hot fluid impinges on the front. The most important result to be obtained is the evolution of the phase change front over time, among other aspects. The comparison with the literature is not simple because the results vary significantly between publications. The overall trend obtained by all simulations in the literature and the results obtained with this present simulation method agree well with the experiments, given the variance of the results in the literature.

Elastic material

Secondly, the implementation of the elastic solid is verified using a large number of small quasi-static test cases that include uniaxial tension and simple shear on rectangles for example. Expected results are derived from basic solid mechanics. Different combinations of rectangle dimensions, mesh resolution, boundary conditions and loadings are tested and the exact result is recovered in all cases, with errors that correspond to machine accuracy.

Further more, a quasi-static and dynamic bending of a beam under a distributed load is tested, where expected results can be obtained with beam theory. Both the quasi-static loading and the dynamic loading deliver good results, given that the observed unphysical damping of the oscillating beam is to be expected when using the first order accurate backward Euler time integration.

Thermo-fluid-solid material

The last verification step for the unified formulation puts fluids and solids in the same domain and applies external loads to check, if the materials react as desired and if the interface functions correctly. These verification tests bear some resemblance to the purely elastic tests described above, as they model very simple load cases on small domains with varying geometry, mesh density etc. The simulation results are then compared to analytical solutions obtained from basic solid mechanics. All test cases produce a good agreement with the expected results.

The verification spans a large number of individual test cases because many new features are introduced in this work. While there are certainly more possible scenarios that were not tested, the results that were obtained so far allow to conclude that the unfied formulation and the energy conservation with latent heat are implemented correctly and that useful results can be obtained with this novel method.

5.1.3 Demonstration test cases

Chapter 4 contains three demonstration test cases that resemble some industrial applications. In these test cases, most capabilities of the present PFEM code are combined leading to a very high complexity. While in the future, these test cases could grow into actual validation test cases, this is out of the focus of this doctoral thesis, where the emphasis lies on the novelty of the method and on highlighting its potential, rather than its capability to model a specific industrial process accurately. For this purpose these three test cases are selected to shine the focus on a specific aspect:

1. A bird strike to show an FSI application. The reference in the literature uses a coupled approach.
2. A Tungsten Inert Gas (TIG) spot welding process to show thermal stresses with phase change. The reference in the literature uses a coupling between an Eulerian FEM code for fluid dynamics and Lagrangian FEM code for solid mechanics.
3. A laser spot welding process to show free surface and melt pool evolution with phase change. The reference in the literature uses an Eulerian FVM code for fluid dynamics.

The **bird strike** test case was derived from a test case published by Cerquaglia et al. [9] of Prof. Ponthot's group (LTAS/MN2L) at ULiège. A bullet shaped object made of water (i.e. the bird) impacts a thick steel disk at a speed of $117\,\mathrm{m\,s^{-1}}$. The bird is squished against the relatively rigid steel disk and splashes radially outward. The steel disk does deform slightly under the load of the impact and the stresses in the disk can be evaluated. The results are nearly identical to the ones obtained with the coupled method used in [9] and the test case highlight well the potential of the unified approach for FSI problems.

An interesting comparison can be made, when running the simulation with the unified PFEM approach and with or without the interface. With the interface, a sharp discontinuity of the pressure is observed between the fluid and the solid. This is the expected behavior and also the behavior observed in the coupled method. Without the interface, the pressure at the fluid-solid contact is continuous, meaning that either the fluid or the solid or both have a wrong pressure at the contact surface. Surprisingly, the overall behavior of the system bird-and-disk is nearly unchanged. This may hint at the possibility that the interface may not even be needed in FSI problems, where the fluid and solid regions are distinct.

The **TIG spot welding** test case was originally published by Desmaison [10] and later repeated by Chen [4], both from the team of Prof. Bellet (CEMEF/2MS) at MINES Paris - PSL. In this test, a cylindrical steel disk is subjected to a stationary and constant heat source. The steel eventually begins to melt under the heat source, forming the so-called melt pool. As a result of thermal expansion, the volume of the steel increases locally which leads to stresses in the solid material and to a deformation of the free surface.

The original aim for this test case was to compare the stresses obtained by the method published in the literature and the present approach. Unfortunately, this comparison is not yet possible, since the more complex solid material behavior could not be implemented in the present simulation code in time. There are nonetheless some interesting observations to be made: due to a decrease in stiffness at high temperatures, a complex stress distribution is obtained, both in the literature and with the present simulation method. The stresses are locally high enough that they could surpass the yield stress, leading to plastic flow. In a real world application, this would be interesting to study in order to predict the strength of the weld accurately. Including the convective flow in the melt pool caused by buoyancy, Marangoni effect, Lorentz force and evaporation recoil pressure, would increase the realism of the prediction of the heat flow and therefore the prediction of the thermal expansion and the resulting residual stresses. Desmaison did not include these effects, presumably to keep this test case simple, despite their simulation code being able to include such effects.

The **laser spot welding** test case originally published by Saldi [12] from the group of Prof. Kleijn (Chemical engineering Dept./TP) at the TU Delft features a spot welding test case similar to the one above. The main feature of the test case is the significant convective flow in the melt pool caused by the Marangoni effect. This test case was originally conducted to study the shape of the melt pool as a function of process parameters such as laser power, material composition and heating time. Saldi provides a large body of data (mainly melt pool dimensions, sometimes temperatures and flow velocities), for many different combinations of these parameters. Saldi uses an Eulerian FVM approach for Newtonian fluids and uses a fine mesh and very small time steps to resolve the rapid fluid flow.

With the present method, the cost of running simulations with a similar mesh size and time step size is excessive. From the large number of variations of the test case, a selection is made that allows for larger time steps and coarser meshes. Unfortunately, this is very limiting such that only some comparisons can be made. However, being a demonstration case, rather than a validation case, this is not a big problem. Three levels of increasing difficulty are presented for this purpose:

1. A purely Newtonian fluid approach with the well-known Carman-Kozeny equation and a free surface.
2. Simulation with the unified formulation
   (a) with a flat surface.
   (b) with a free surface.
3. Simulation with the unified formulation and with a free surface and an additional thermal expansion with a heating phase and a cooling phase.

The first two variants serve as a verification of the new modeling approach because they can be easily compared to the literature. The third variant is the demonstration for the real advantage of the unified formulation: modeling a fluid and an elastic solid in a single solver to obtain accurate fluid flow in the melt pool and also residual stresses in the solid, both of which are coupled. This is a capability that goes beyond that of Saldi's method and therefore no comparison of results is possible.

From the validation steps 1 and 2 it is found that an acceptable overall agreement with the reference can be observed. The free surface deformation under the Marangoni effect is observed in the cases that feature a free surface, as well as the temperature evolution and the flow in the melt pool. Some discrepancies are found with the temperatures and velocity, likely due to the coarser mesh that is used. The melt pool shape agrees well, which appears to be the most important feature in Saldi's work.

The demonstration test case in step 3 shows the non-uniform thermal expansion in the constrained solid material and subsequently induced non-uniform stresses, which are analyzed in detail. Particular attention is given to the stress evolution throughout the cooling phase, where the residual stresses are generated that later lead to distortion of the finished part in a real world welding process. Stresses in the solid cannot be obtained with Saldi's simulation method, so no comparison is made.

In general, it appears that the stresses that are observed are plausible and the demonstration of this unique capability of the unified fluid-solid formulation is successful.

In all demonstration cases in chapter 4, the novelty lies in the simulation using the unified fluid-solid formulation, which allows to simulate the viscous fluid flow and the elastic response to external forces *simultaneously*. While in all the test cases, the references in the literature do not utilize a unified formulation, we show, that some advantages can be gained from the use of this method:

- Compared with coupled approach relying on two pieces of software (bird strike): the unified approach removes the complexity of the coupling algorithm, while strong coupling is ensured naturally.
- Compared with the coupled FEM for thermo-solids and thermo-fluids (TIG spot welding): a unified approach avoids the complexity of the coupling, while it may be able to produce the same results (which needs to be confirmed in the future, see section 5.2.3).

- Compared with FVM for thermo-fluids (laser spot welding): the unified approach allows modeling+ residual stresses in the solid, which can have a significant effect in a real world design problem (welding is particularly well known for residual stresses being a problem). This is not possible with the commonly used fluid approach in the context of welding and AM.

5.2 Conclusion

There is a lack of unified fluid-solid simulation methods that are able to model certain Fluid-Structure Interaction (FSI) problems or phase change problems, where coupled methods may not be able to take all relevant physics into account. This work presents such a unified simulation method. As outlined above, the verification of the unified formulation, where the solid is limited to linear elasticity, was successful. This marks the important first step towards a new class of simulation methods that can simulate phase change in a unified fashion, where only one material exists that can switch between its solid and fluid state, which is difficult to achieve with conventional coupling methods.

The demonstration test cases summarized above aim to provide an idea about the capabilities of such a simulation method. It becomes clear that such a method can play a key role in many applications, where phase change could not be simulated easily at this meso-scale. Likewise, it also becomes apparent that these abilities do come at a cost and that this method already has some known limitations. These capabilities and limitations shall be briefly discussed in this section, followed by some promising future development paths.

5.2.1 Capabilities of the method

The results demonstrate that the unified formulation can accomplish the following:

- Model melting and re-solidification.
- Model the convective flow in the melt pool.
- With the convection, model the heat transfer in the fluid and the solid correctly.
- Model the the deformation of the free surface.
- Model the stresses introduced during heating, due to thermal expansion.
- Model the final residual stresses after cooling and contraction.

While there are many simulation methods that can easily and efficiently reproduce *most* of these effects, only few approaches can include *all* of them at once. This is the main advantage of the unified fluid-solid formulation.

The underlying simulation method that employs the unified formulation is the Particle Finite Element Method (PFEM), which can be described as a particle method based on the classic Lagrangian Finite Element Method (FEM). Its strength lies in the combination of the robustness and versatility of the FEM with the flexibility of the particle approach. In particular, this means that the particle approach allows to model fluid flows with unlimited deformation and the versatility of the FEM allows for complex material laws, like a unified formulation to be implemented.

5.2.2 Limitations of the method

The presented PFEM code underwent rapid development, leaving many parts of the implemented code crude and inefficient. However, even with an optimized code, PFEM is known to be **relatively expensive**. This is due to several factors:

- Frequent remeshing can increase the cost (both memory and CPU). This is not a detrimental problem yet because the present PFEM code only operates in 2D. In 3D this cost could skyrocket
  1. because the mesh manipulations and the remeshing become more costly and
  2. because the mesh used to simulate a given process becomes much larger (more nodes, elements etc.)
- Parallelization by domain partitioning is difficult because of a constantly changing connectivity.

Furthermore, the unified formulation incurs extra cost that a specialized simulation method would not have. This is due to keeping more data in memory (e.g. reference deviatoric stress at the Gauss points) and additional computations (e.g. updating the current stress). Lastly, the use of an interface adds additional cost due to the complex algorithm that detects, establishes and removes the interface. Likewise, the Lagrange multipliers increase the size of the system of equations, therefore increasing the cost further.

The remeshing and mesh refinement require **tuning of parameters** that can be difficult to determine for the user. Wrongly chosen parameters often lead to illegal mesh operations or excessive mesh distortion that can crash a simulation without warning. The remeshing algorithm in the present PFEM code is very complex, with many small details that are not explained in this work. In some cases, the mass conservation can depend on the choice of remeshing parameters, which reduces the predictive capabilities of the PFEM.

The **interface** is complicated and is prone to cause crashes, when mesh manipulations occur in the vicinity of the interface. This is, however, not a systematic issue, but rather a problem with the implementation. The PFEM code was written with the idea in mind that nodes can be added or deleted anywhere and the automatic remeshing will take care of the rest. This means that the remeshing becomes a black box to some extent. This is in conflict with the way the interface detection is designed, where a precise control over the mesh is required (e.g. neighbor elements or nodes are known). A well designed simulation code would feature a remeshing algorithm that uses local mesh manipulations, which facilitates keeping track of all changes in the mesh and having precise control over the mesh. On the other hand, there is some evidence that suggests that the interface is not needed in the first place, which would eliminate both the added cost and the complexity that leads to errors in the algorithm.

The classic PFEM **free surface handling** (Delaunay + α-shape) is well suited for random and chaotic free surface deformation (splashing, closing waves, sloshing), but struggles with the continuous tangential force of the Marangoni effect over long period of time. Remeshing errors can occur or the mass may not be well preserved.

5.2.3 Future work

Besides the application of this simulation method to other processes than the ones found in this work, there are two directions in which future work should head:

1. Improvement of the existing method to eradicate the above mentioned shortcomings.
2. Further development and validation of the method to simulate laser welding and L-PBF more realistically.

Improvements of the method

To improve the **free surface handling** under a strong tangential forcing (i.e. the Marangoni effect) over a long period of time, a smarter handling of the free surface would be beneficial. There are two interesting options: replacing the unconstrained Delaunay tessellation by a constrained Delaunay tessellation or replacing the α-shape technique by a technique based on level sets (LS). The first option is mentioned by Cremonesi [38], who recommends its use for solid mechanics applications only, due to the difficulty encountered when splashing, droplets or closing waves occur. The second option is currently being explored by Fernández et al. [113], who successfully demonstrates the improvements regarding the above-mentioned issues when using an LS-based PFEM approach.

Along with a better handling of the free surface, the **surface tension** normal (i.e. an effect of the surface curvature) and tangential (i.e. the Marangoni effect) components should be modeled more directly. Currently, both components are treated separately and calculated in different parts of the algorithm, while they are obviously intimately related. Instead of deriving models that include functions of curvature and temperature gradients, an actual tension along the surface could be modeled. Leblond et al. [114] propose such an approach, where membrane elements are introduced that exert a tension on the surface.

This would directly cover both the curvature effect and the Marangoni effect, whenever there is a curvature or a temperature gradient.

The **remeshing algorithm** itself needs to be simplified and ideally the complete remeshing should be replaced by local mesh manipulations. This simplifies further development and maintenance of the code, probably reduces the cost (especially in 3D) and makes the code more robust. In the same sense, the algorithm for the interface detection is relatively complex because it has to correct errors of the remeshing algorithm to work correctly. With the simplified remeshing, the interface management becomes simpler as well and less costly.

Further development of the method for laser welding and L-PBF simulations

Maybe the most important future development to make this simulation method a more general one is the **extension from 2D to 3D**. In the context of welding it would allow to simulate processes that are not spot welding. For the use in an additive manufacturing context, which is the long term goal, a 2D plane strain or axisymmetric formulation is not useful and a 3D representation is even required. The extension to 3D is simple in principle, since all of the equations and methods are already written in a general way. The difficulty lies in the details. Besides the increased cost, remeshing in 3D using a Delaunay tessellation produces new challenges. Some well-known examples of these challenges include the non-uniqueness of the Delaunay tessellation in 3D or the appearance of sliver elements. Another well-known problematic detail is the computation of the curvature on a discretized free surface in 3D, which is required for the surface tension modeling. Besides the well-known difficulties, there may be many more that cannot be predicted. One example of this is the currently complex algorithm for the detection of the interface, which may become even more complex in 3D.

The current unified fluid-solid formulation is capable of modeling a simple linear elastic solid material. While this can already be useful for some applications, an extension of the solid model to an **elasto-visco-plastic material** model as in [4, 10, 76] would open up many possibilities to simulate more complex processes. One example was given in section 4.2 and some others in section 1.4.2, but there are endless other applications, where plasticity occurs in the solid.

Along with the plasticity, the small displacement formulation needs to be extended to **large deformations**, so that there is no limitation to the deformation of the solid. Although there may not be any large strains that typically occur in welding applications, some rotations may already be encountered (e.g. a slender structure warping after welding and cooling). Furthermore, this simulation method shall not be limited to welding, but rather become a very general tool for any phase change applications, where large strains and rotations occur.

The treatment of the **mushy zone** at the meso-scale is not always clear in the literature, presumably due to the complexity of modeling the solidification. Most works cited in section 1.4.2 do not delve into this topic and oftentimes assume a linear distribution of the liquid fraction f_l between solidus and liquidus, which is then commonly inserted into the Carman-Kozeny equation (see eq. 2.49, section 2.3.3), as was done in the present work. More literature needs to be reviewed to determine an appropriate modeling strategy and potentially, a more elaborate material law needs to be implemented. Good examples include the visco-elastic behavior for a ceramic, as done by Chen [4] or a visco-elastic model for the deviatoric stress, as done by Saadlaoui et al. [79] or the elasto-visco-plastic behavior for thixotropic aluminium, as done by Koeune and Ponthot [76]. In the long term, the solidification could be modeled more realistically by employing a micro-scale model, as was done by Bayat et al. [2]

For the long term aim of simulating additive manufacturing processes, specifically Powder Bed Fusion (PBF), a model for the **powder** must also be included. While a fine powder could be modeled as a continuum, the difficulty lies in the rapid densification, as the porous powder becomes a dense liquid nearly instantly. The literature suggests solutions to deal with this challenge that should be implemented and tested in this present simulation method (e.g. Chen [4]).

Laser welding and L-PBF processes can also be operated in keyhole mode (see section 1.2.2), where the heat input is so intense that evaporation of the material causes significant **recoil pressure**. This recoil pressure is essential in the formation of the keyhole and is therefore included in many simulations mentioned in section 1.4.2. The equations needed are well-known (see e.g. Cook and Murphy [13]), but the implementations has not been attempted in favor of a more complete verification of the existing simulation capabilities. With keyhole mode welding included, many more process scenarios could be investigated.

Usually with keyhole mode welding, the simple laser models (e.g. a radially Gaussian distribution for a heat flux) may not capture the realistic laser-material interaction anymore. With a complex surface under the laser beam, as is the case with keyhole, **ray tracing** may be the most appropriate approach. Some authors' works presented in section 1.4.2 (e.g. Bayat et al. [5]) include ray tracing and achieve realistic result, but with increased cost due to the more complex algorithm.

Lastly, an interesting question that still needs an answer is whether or not an **interface** is actually needed or in which cases the interface makes a (positive) difference. Being aware of the unrealistic smooth transition of the pressure between fluid and solid may be sufficient for some users to avoid the added complexity and the added overhead cost of managing the interface.

Appendix A

Pressure-Stabilizing Petrov-Galerkin (PSPG)

PSPG is a residual method, which introduces a perturbation in the mass conservation equation that vanishes, as the solution converges towards a balanced state. The perturbation of the mass conservation equation is derived from the momentum conservation equation. When the momentum equation is solved and its residual vanishes, then the perturbation of the mass conservation equation also vanishes. In this way this approach provides stability to the unconverged solution, which allows the non-linear iterations to converge in a stable manner, to finally end up with a (nearly) unperturbed and balanced solution. This is achieved by using a modified velocity test function $\tilde{\tilde{\boldsymbol{w}}}$

$$\tilde{\boldsymbol{w}} = \boldsymbol{w} + \tau_{PSPG} \nabla q \tag{A.1}$$

where $\tilde{\boldsymbol{w}}$ is the original velocity test function, q is the pressure test function and τ_{PSPG} is a stabilization factor that will be defined shortly. We introduce eq. A.1 into the time-discretized (but not spatially discretized) weak form of the momentum equation (eq. 2.113) and then apply all the steps of the spatial discretization described thereafter. Due to the modified $\tilde{\boldsymbol{w}}$, the new additional terms of the momentum equation appear, but they all add up to zero. Because the sum of these terms is zero, they can be subtracted from the momentum equation and added to the mass conservation equation. After this rearrangement of the terms, the momentum equation returns to its original form, but a new, stabilized mass conservation equation is obtained:

$$\frac{1}{\Delta t}\mathbf{M}_\tau \cdot \boldsymbol{v} + \mathbf{D} \cdot \boldsymbol{v} + \frac{1}{\Delta t}\mathbf{K}_{vol} \cdot \boldsymbol{p} + \mathbf{L}_\tau \cdot \boldsymbol{p} = \underbrace{\boldsymbol{a} + \frac{1}{\Delta t}\mathbf{K}_{vol} \cdot \boldsymbol{p}^{n-1} + \boldsymbol{h}_\tau + \frac{1}{\Delta t}\mathbf{M}_\tau \cdot \boldsymbol{v}^{n-1}}_{\boldsymbol{r}_p} \tag{A.2}$$

$$\frac{1}{\Delta t}\mathbf{M}\cdot\boldsymbol{v}+\mathbf{K}_{dev}\cdot\boldsymbol{v}+\mathbf{D}^{T}\cdot\boldsymbol{p}=\underbrace{\boldsymbol{b}+\boldsymbol{f}+\boldsymbol{t}+\frac{1}{\Delta t}\mathbf{M}\cdot\boldsymbol{v}_0+\boldsymbol{k}_{dev}}_{[\boldsymbol{r}_x\,\boldsymbol{r}_y]^T} \tag{A.3}$$

$$\frac{1}{\Delta t}\mathbf{M}_{\tau}\cdot\boldsymbol{v}+\mathbf{D}\cdot\boldsymbol{v}+\frac{1}{\Delta t}\mathbf{K}_{vol}\cdot\boldsymbol{p}+\mathbf{L}_{\tau}\cdot\boldsymbol{p}=\underbrace{\boldsymbol{a}+\frac{1}{\Delta t}\mathbf{K}_{vol}\cdot\boldsymbol{p}_0+\boldsymbol{h}_{\tau}+\frac{1}{\Delta t}\mathbf{M}_{\tau}\cdot\boldsymbol{v}_0}_{\boldsymbol{r}_p} \tag{A.4}$$

The details of these steps are outlined in Tezduyar et al. [89]. There, two definition for the stabilization parameter τ_{PSPG} can also be found. One uses a local velocity scale and one uses a global velocity scale. In all simulations in this work, the second option was chosen, as it reliably seemed to suppress any pressure oscillation. First, a Reynolds number Re_u is defined:

$$Re_u=\frac{u_c\,h_e\,\rho}{2\mu} \tag{A.5}$$

where u_c is the characteristic velocity set by the user and h_e is the element size, defined as the diameter of a circle with an area A_e equal to that of the element:

$$h_e=2\sqrt{\frac{A_e}{\pi}} \tag{A.6}$$

Then, τ_{PSPG} is computed depending on the magnitude of the Reynolds number Re_u:

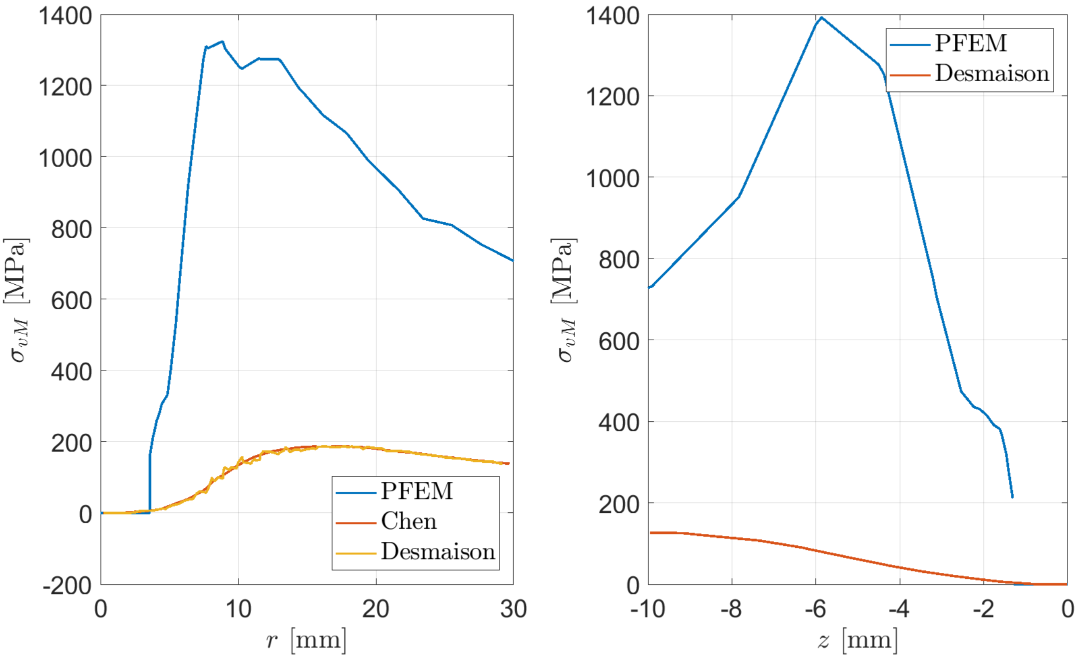
$$\tau_{PSPG}=\begin{cases}\dfrac{h_e}{2u_c}\,\dfrac{Re_u}{3}=\dfrac{A_e\,\mu}{3\pi^2\,\rho} & \text{for } 0\leq Re_u\leq 3\\[2ex] Re_u=\dfrac{\sqrt{A_e}}{\pi\,u_c} & \text{for } 3>Re_u\end{cases} \tag{A.7}$$

Appendix B

Additional results for TIG spot welding test

Following the test case by Chen [4] and Desmaison [10] the profiles of the equivalent von Mises stress $\bar{\sigma}_{VM}$ are plotted. These shall serve as a future reference.

A comparison of $\bar{\sigma}_{VM}$ along at $y = 0$ and the at $x = 0$ (dashed horizontal line and dotted vertical line in fig. 4.13b) reveals more clearly how different the stresses are. While the horizontal distribution in fig. B.1a shows a similar trend, but with a magnitude about 7 times larger than the reference, the vertical distribution in fig. B.1b has a completely different trend. In the latter, the highest magnitude is achieved at the bottom of the domain ($z = -10\,\mathrm{mm}$) for Desmaison [10] and at mid-height ($z \approx -6\,\mathrm{mm}$) using the present method with the elastic material.



(a) Distribution of equivalent von Mises stress by Chen [4], Desmaison [10] and the present work. Data extracted along horizontal dashed line

(b) Distribution of equivalent von Mises stress by Desmaison [10] and the present work. Data extracted along vertical dotted line

FIGURE B.1: Comparison of equivalent von Mises stress profiles extracted at dashed lines in fig. 4.13b

Bibliography

[1] S. Kollmannsberger, A. Özcan, M. Carraturo, N. Zander, and E. Rank, "A hierarchical computational model for moving thermal loads and phase changes with applications to selective laser melting," *Computers and Mathematics with Applications*, vol. 75, no. 5, pp. 1483–1497, Mar. 2018, ISSN: 08981221. DOI: `10.1016/j.camwa.2017.11.014`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S0898122117307289`.

[2] M. Bayat, S. Mohanty, and J. H. Hattel, "A systematic investigation of the effects of process parameters on heat and fluid flow and metallurgical conditions during laser-based powder bed fusion of Ti6Al4V alloy," *International Journal of Heat and Mass Transfer*, vol. 139, pp. 213–230, 2019, ISSN: 00179310. DOI: `10.1016/j.ijheatmasstransfer.2019.05.017`. [Online]. Available: `https://doi.org/10.1016/j.ijheatmasstransfer.2019.05.017`.

[3] S. A. Khairallah, A. T. Anderson, A. Rubenchik, and W. E. King, "Laser powder-bed fusion additive manufacturing: Physics of complex melt flow and formation mechanisms of pores, spatter, and denudation zones," *Acta Materialia*, vol. 108, pp. 36–45, 2016, ISSN: 13596454. DOI: `10.1016/j.actamat.2016.02.014`. [Online]. Available: `http://dx.doi.org/10.1016/j.actamat.2016.02.014`.

[4] Q. Chen, "Thermomechanical numerical modelling of additive manufacturing by selective laser melting of powder bed : Application to ceramic materials," PhD, MINES ParisTech, Apr. 2018. [Online]. Available: `https://pastel.archives-ouvertes.fr/tel-01814209`.

[5] M. Bayat, A. Thanki, S. Mohanty, A. Witvrouw, S. Yang, J. Thorborg, N. S. Tiedje, and J. H. Hattel, "Keyhole-induced porosities in Laser-based Powder Bed Fusion (L-PBF) of Ti6Al4V: High-fidelity modelling and experimental validation," *Additive Manufacturing*, vol. 30, p. 100 835, Dec. 2019, ISSN: 22148604. DOI: `10.1016/j.addma.2019.100835`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S2214860419303914`.

[6] S. Pang, X. Chen, X. Shao, S. Gong, and J. Xiao, "Dynamics of vapor plume in transient keyhole during laser welding of stainless steel: Local evaporation, plume swing and gas entrapment into porosity," *Optics and Lasers in Engineering*, vol. 82, pp. 28–40, Jul. 2016, ISSN: 01438166. DOI: `10.1016/j.optlaseng.2016.01.019`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S014381661600035X`.

[7] C. Panwisawas, C. Qiu, M. J. Anderson, Y. Sovani, R. P. Turner, M. M. Attallah, J. W. Brooks, and H. C. Basoalto, "Mesoscale modelling of selective laser melting: Thermal fluid dynamics and microstructural evolution," *Computational Materials Science*, vol. 126, pp. 479–490, 2017, ISSN: 09270256. DOI: 10.1016/j.commatsci.2016.10.011. [Online]. Available: http://dx.doi.org/10.1016/j.commatsci.2016.10.011.

[8] C. Lüthi, M. Afrasiabi, and M. Bambach, "An adaptive smoothed particle hydrodynamics (SPH) scheme for efficient melt pool simulations in additive manufacturing," *Computers and Mathematics with Applications*, vol. 139, no. March, pp. 7–27, Jun. 2023, ISSN: 08981221. DOI: 10.1016/j.camwa.2023.03.003. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0898122123000925.

[9] M. Cerquaglia, G. Deliége, R. Boman, L. Papeleux, and J. Ponthot, "The particle finite element method for the numerical simulation of bird strike," *International Journal of Impact Engineering*, vol. 109, pp. 1–13, Nov. 2017, ISSN: 0734743X. DOI: 10.1016/j.ijimpeng.2017.05.014. [Online]. Available: http://dx.doi.org/10.1016/j.ijimpeng.2017.05.014%20https://linkinghub.elsevier.com/retrieve/pii/S0734743X16309022.

[10] O. Desmaison, "Modélisation numérique d ' un procédé de soudage hybride arc / laser en approche level set : application au soudage multi-passes de tôles d ' acier de forte épaisseur," PhD, Ecole Nationale Supérieure des Mines de Paris, 2014. [Online]. Available: https://pastel.archives-ouvertes.fr/pastel-00982377/.

[11] F. Costes, "Modélisation thermomécanique tridimensionnelle par éléments finis de la coulée continue d'aciers," Ph.D. dissertation, École Nationale Supérieure des Mines de Paris, 2004. [Online]. Available: https://pastel.archives-ouvertes.fr/pastel-00001334.

[12] Z. S. Saldi, "Marangoni driven free surface flows in liquid weld pools," PhD, Technische Universiteit Delft, 2012, p. 160, ISBN: 9789461909800. DOI: 8401374b-9e9c-4d25-86b7-fc445ec73d27.

[13] P. S. Cook and A. B. Murphy, "Simulation of melt pool behaviour during additive manufacturing: Underlying physics and progress," *Additive Manufacturing*, vol. 31, no. October 2019, 2020, ISSN: 22148604. DOI: 10.1016/j.addma.2019.100909.

[14] W. Yan, S. Lin, O. L. Kafka, Y. Lian, C. Yu, Z. Liu, J. Yan, S. Wolff, H. Wu, E. Ndip-Agbor, M. Mozaffar, K. Ehmann, J. Cao, G. J. Wagner, and W. K. Liu, "Data-driven multi-scale multi-physics models to derive process–structure–property relationships for additive manufacturing," *Computational Mechanics*, vol. 61, no. 5, pp. 521–541, 2018, ISSN: 01787675. DOI: 10.1007/s00466-018-1539-z.

[15] A. Antonysamy, J. Meyer, and P. Prangnell, "Effect of build geometry on the β-grain structure and texture in additive manufacture of Ti6Al4V by selective electron beam melting," *Materials Characterization*, vol. 84, pp. 153–168, Oct. 2013, ISSN: 10445803. DOI: `10.1016/j.matchar.2013.07.012`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S1044580313002131`.

[16] E. H. Penilla, L. F. Devia-Cruz, A. T. Wieg, P. Martinez-Torres, N. Cuando-Espitia, P. Sellappan, Y. Kodera, G. Aguilar, and J. E. Garay, "Ultrafast laser welding of ceramics," *Science*, vol. 365, no. 6455, pp. 803–808, Aug. 2019, ISSN: 0036-8075. DOI: `10.1126/science.aaw6699`. [Online]. Available: `https://www.science.org/doi/10.1126/science.aaw6699`.

[17] K. Weman, *Welding Processes Handbook*. CRC Press, Aug. 2003, ISBN: 978-0-8493-1773-6. DOI: `10.1201/9780203499764`. [Online]. Available: `http://www.crcnetbase.com/doi/book/10.1201/9780203499764`.

[18] L. Quintino and E. Assunção, "Conduction laser welding," in *Handbook of Laser Welding Technologies*, Elsevier, 2013, pp. 139–162. DOI: `10.1533/9780857098771.1.139`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/B9780857092649500064`.

[19] M. Baruah and S. Bag, "Influence of heat input in microwelding of titanium alloy by micro plasma arc," *Journal of Materials Processing Technology*, vol. 231, no. November 2017, pp. 100–112, 2016, ISSN: 09240136. DOI: `10.1016/j.jmatprotec.2015.12.014`.

[20] X. Zhang, Z. Cao, and P. Zhao, "Investigation on solidification cracks in pulsed laser spot welding of an AZ31 magnesium alloy," *Optics and Laser Technology*, vol. 126, no. August 2019, 2020, ISSN: 00303992. DOI: `10.1016/j.optlastec.2020.106132`.

[21] Y. Zhang, F. Lu, H.-P. Wang, X. Wang, H. Cui, and X. Tang, "Reduced hot cracking susceptibility by controlling the fusion ratio in laser welding of dissimilar Al alloys joints," *Journal of Materials Research*, vol. 30, no. 7, pp. 993–1001, Apr. 2015, ISSN: 0884-2914. DOI: `10.1557/jmr.2015.64`. [Online]. Available: `http://link.springer.com/10.1557/jmr.2015.64`.

[22] P. Mercelis and J.-P. Kruth, "Residual stresses in selective laser sintering and selective laser melting," *Rapid Prototyping Journal*, vol. 12, no. 5, pp. 254–265, Oct. 2006, ISSN: 1355-2546. DOI: `10.1108/13552540610707013`. [Online]. Available: `https://www.emerald.com/insight/content/doi/10.1108/13552540610707013/full/html`.

[23] A. F. H. Kaplan and J. Powell, "Spatter in laser welding," *Journal of Laser Applications*, vol. 23, no. 3, p. 032 005, Aug. 2011, ISSN: 1042-346X. DOI: `10.2351/1.3597830`. [Online]. Available: `http://lia.scitation.org/doi/10.2351/1.3597830`.

[24] W. E. King, H. D. Barth, V. M. Castillo, G. F. Gallegos, J. W. Gibbs, D. E. Hahn, C. Kamath, and A. M. Rubenchik, "Observation of keyhole-mode laser melting in laser powder-bed fusion additive manufacturing," *Journal of Materials Processing Technology*, vol. 214, no. 12, pp. 2915–2925, Dec. 2014, ISSN: 09240136. DOI: `10.1016/j.jmatprotec.2014.06.005`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S0924013614002283`.

[25] L. Huang, X. Hua, D. Wu, and F. Li, "Numerical study of keyhole instability and porosity formation mechanism in laser welding of aluminum alloy and steel," *Journal of Materials Processing Technology*, vol. 252, no. October 2017, pp. 421–431, 2018, ISSN: 09240136. DOI: `10.1016/j.jmatprotec.2017.10.011`. [Online]. Available: `http://dx.doi.org/10.1016/j.jmatprotec.2017.10.011`.

[26] Q. Chen, G. Guillemot, C. A. Gandin, and M. Bellet, "Three-dimensional finite element thermomechanical modeling of additive manufacturing by selective laser melting for ceramic materials," *Additive Manufacturing*, vol. 16, pp. 124–137, 2017, ISSN: 22148604. DOI: `10.1016/j.addma.2017.02.005`. [Online]. Available: `http://dx.doi.org/10.1016/j.addma.2017.02.005`.

[27] M. K. Thompson, G. Moroni, T. Vaneker, G. Fadel, R. I. Campbell, I. Gibson, A. Bernard, J. Schulz, P. Graf, B. Ahuja, and F. Martina, "Design for Additive Manufacturing: Trends, opportunities, considerations, and constraints," *CIRP Annals - Manufacturing Technology*, vol. 65, no. 2, pp. 737–760, 2016, ISSN: 17260604. DOI: `10.1016/j.cirp.2016.05.004`. arXiv: `arXiv:1011.1669v3`. [Online]. Available: `http://dx.doi.org/10.1016/j.cirp.2016.05.004`.

[28] I. Koutiri, E. Pessard, P. Peyre, O. Amlou, and T. De Terris, "Influence of SLM process parameters on the surface finish, porosity rate and fatigue behavior of as-built Inconel 625 parts," *Journal of Materials Processing Technology*, vol. 255, no. June 2017, pp. 536–546, 2018, ISSN: 09240136. DOI: `10.1016/j.jmatprotec.2017.12.043`. [Online]. Available: `https://doi.org/10.1016/j.jmatprotec.2017.12.043`.

[29] Y. A. Mayi, M. Dal, P. Peyre, M. Bellet, C. Metton, C. Moriconi, and R. Fabbro, "Laser-induced plume investigated by finite element modelling and scaling of particle entrainment in laser powder bed fusion," *Journal of Physics D: Applied Physics*, vol. 53, no. 7, p. 075 306, Feb. 2020, ISSN: 0022-3727. DOI: `10.1088/1361-6463/ab5900`. [Online]. Available: `https://iopscience.iop.org/article/10.1088/1361-6463/ab5900`.

[30] T. M. Wischeropp, C. Emmelmann, M. Brandt, and A. Pateras, "Measurement of actual powder layer height and packing density in a single layer in selective laser melting," *Additive Manufacturing*, vol. 28, no. August 2018, pp. 176–183, Aug. 2019, ISSN: 22148604. DOI: `10.1016/j.addma.2019.04.019`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S2214860418306444`.

[31] T. Mukherjee and T. DebRoy, "Mitigation of lack of fusion defects in powder bed fusion additive manufacturing," *Journal of Manufacturing Processes*, vol. 36, no. April, pp. 442–449, 2018, ISSN: 15266125. DOI: 10.1016/j.jmapro.2018.10.028. [Online]. Available: https://doi.org/10.1016/j.jmapro.2018.10.028.

[32] T. DebRoy, H. L. Wei, J. S. Zuback, T. Mukherjee, J. W. Elmer, J. O. Milewski, A. M. Beese, A. Wilson-Heid, A. De, and W. Zhang, "Additive manufacturing of metallic components – Process, structure and properties," *Progress in Materials Science*, vol. 92, pp. 112–224, 2018, ISSN: 00796425. DOI: 10.1016/j.pmatsci.2017.10.001.

[33] D. Wang, S. Wu, F. Fu, S. Mai, Y. Yang, Y. Liu, and C. Song, "Mechanisms and characteristics of spatter generation in SLM processing and its effect on the properties," *Materials & Design*, vol. 137, pp. 33–37, Jan. 2018, ISSN: 02641275. DOI: 10.1016/j.matdes.2017.09.058. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0264127517309139.

[34] G. Kasperovich, J. Haubrich, J. Gussone, and G. Requena, "Corrigendum to "Correlation between porosity and processing parameters in TiAl6V4 produced by selective laser melting" [Materials and Design 105 (2016) 160–170]," *Materials & Design*, vol. 112, pp. 160–161, Dec. 2016, ISSN: 02641275. DOI: 10.1016/j.matdes.2016.09.040. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S026412751631214X.

[35] R. Li, J. Liu, Y. Shi, L. Wang, and W. Jiang, "Balling behavior of stainless steel and nickel powder during selective laser melting process," *International Journal of Advanced Manufacturing Technology*, vol. 59, no. 9-12, pp. 1025–1035, 2012, ISSN: 02683768. DOI: 10.1007/s00170-011-3566-1.

[36] H. Edelsbrunner and E. P. Mücke, "Three-dimensional alpha shapes," *ACM Transactions on Graphics*, vol. 13, no. 1, pp. 43–72, Jan. 1994, ISSN: 0730-0301. DOI: 10.1145/174462.156635. [Online]. Available: https://dl.acm.org/doi/10.1145/174462.156635.

[37] S. R. Idelsohn, E. Onate, and F. Del Pin, "The particle finite element method: A powerful tool to solve incompressible flows with free-surfaces and breaking waves," *International Journal for Numerical Methods in Engineering*, vol. 61, no. 7, pp. 964–989, 2004, ISSN: 00295981. DOI: 10.1002/nme.1096.

[38] M. Cremonesi, A. Franci, S. Idelsohn, and E. Oñate, "A State of the Art Review of the Particle Finite Element Method (PFEM)," *Archives of Computational Methods in Engineering*, vol. 27, no. 5, pp. 1709–1735, 2020, ISSN: 18861784. DOI: 10.1007/s11831-020-09468-4.

[39] M. L. Cerquaglia, "Development of a fully-partitioned PFEM-FEM approach for fluid-structure interaction problems characterized by free surfaces , large solid deformations , and strong added-mass effects," Ph.D. dissertation, Université de Liège, 2019.

[40] J. Marti and P. Ryzhakov, “An explicit/implicit Runge–Kutta-based PFEM model for the simulation of thermally coupled incompressible flows,” *Computational Particle Mechanics*, vol. 7, no. 1, pp. 57–69, 2020, ISSN: 21964386. DOI: 10.1007/s40571-019-00229-0.

[41] M. Cremonesi and A. Frangi, “A Lagrangian finite element method for 3D compressible flow applications,” *Computer Methods in Applied Mechanics and Engineering*, vol. 311, pp. 374–392, 2016, ISSN: 00457825. DOI: 10.1016/j.cma.2016.08.005. [Online]. Available: http://dx.doi.org/10.1016/j.cma.2016.08.005.

[42] A. Franci and M. Cremonesi, “On the effect of standard PFEM remeshing on volume conservation in free-surface fluid flow problems,” *Computational Particle Mechanics*, vol. 4, no. 3, pp. 331–343, Jul. 2017, ISSN: 2196-4378. DOI: 10.1007/s40571-016-0124-5. [Online]. Available: http://link.springer.com/10.1007/s40571-016-0124-5.

[43] J. M. Carbonell, E. Oñate, and B. Suárez, “Modelling of tunnelling processes and rock cutting tool wear with the particle finite element method,” *Computational Mechanics*, vol. 52, no. 3, pp. 607–629, 2013, ISSN: 01787675. DOI: 10.1007/s00466-013-0835-x.

[44] E. Oñate, A. Franci, and J. M. Carbonell, “A particle finite element method for analysis of industrial forming processes,” *Computational Mechanics*, vol. 54, no. 1, pp. 85–107, 2014, ISSN: 01787675. DOI: 10.1007/s00466-014-1016-2.

[45] J. M. Rodríguez, J. M. Carbonell, J. C. Cante, and J. Oliver, “Continuous chip formation in metal cutting processes using the Particle Finite Element Method (PFEM),” *International Journal of Solids and Structures*, vol. 120, pp. 81–102, 2017, ISSN: 00207683. DOI: 10.1016/j.ijsolstr.2017.04.030.

[46] S. R. Idelsohn, J. Marti, A. Limache, and E. Oñate, “Unified Lagrangian formulation for elastic solids and incompressible fluids: Application to fluid-structure interaction problems via the PFEM,” *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 19-20, pp. 1762–1776, 2008, ISSN: 00457825. DOI: 10.1016/j.cma.2007.06.004.

[47] A. Franci, E. Oñate, and J. M. Carbonell, “Unified Lagrangian formulation for solid and fluid mechanics and FSI problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 298, pp. 520–547, 2016, ISSN: 00457825. DOI: 10.1016/j.cma.2015.09.023.

[48] A. Franci, E. Oñate, J. M. Carbonell, and M. Chiumenti, “PFEM formulation for thermo-coupled FSI analysis. Application to nuclear core melt accident,” *Computer Methods in Applied Mechanics and Engineering*, vol. 325, pp. 711–732, 2017, ISSN: 00457825. DOI: 10.1016/j.cma.2017.07.028. [Online]. Available: http://dx.doi.org/10.1016/j.cma.2017.07.028.

[49] A. Franci, *Unified Lagrangian Formulation for Fluid and Solid Mechanics, Fluid-Structure Interaction and Coupled Thermal Problems Using the PFEM* (Springer Theses April 2015). Cham: Springer International Publishing, 2017, ISBN: 978-3-319-45661-4. DOI: 10.1007/978-3-319-45662-1. arXiv: arXiv:1106.3562. [Online]. Available: http://link.springer.com/10.1007/978-3-319-45662-1.

[50] S. R. Idelsohn, E. Oñate, N. Calvo, and F. Del Pin, "The meshless finite element method," *International Journal for Numerical Methods in Engineering*, vol. 58, no. 6, pp. 893–912, Oct. 2003, ISSN: 00295981. DOI: 10.1002/nme.798. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/nme.798.

[51] S. Li and W. K. Liu, "Meshfree and particle methods and their applications," *Applied Mechanics Reviews*, vol. 55, no. 1, pp. 1–34, Jan. 2002, ISSN: 0003-6900. DOI: 10.1115/1.1431547. [Online]. Available: https://asmedigitalcollection.asme.org/appliedmechanicsreviews/article/55/1/1/458966/Meshfree-and-particle-methods-and-their.

[52] J. Donea and A. Huerta, *Finite Element Methods for Flow Problems.* Wiley, Apr. 2003, pp. 28–29, ISBN: 9780471496663. DOI: 10.1002/0470013826. [Online]. Available: https://onlinelibrary.wiley.com/doi/book/10.1002/0470013826.

[53] C. Hirsch, "Finite Volume Method and Conservative Discretization with an Introduction to Finite Element Method," *Numerical Computation of Internal and External Flows*, vol. M, pp. 203–248, 2007. DOI: 10.1016/b978-075066594-0/50046-1.

[54] P. Cardiff and I. Demirdžić, "Thirty Years of the Finite Volume Method for Solid Mechanics," *Archives of Computational Methods in Engineering*, vol. 28, no. 5, pp. 3721–3780, Aug. 2021, ISSN: 1134-3060. DOI: 10.1007/s11831-020-09523-0. arXiv: 1810.02105. [Online]. Available: https://doi.org/10.1007/s11831-020-09523-0%20https://link.springer.com/10.1007/s11831-020-09523-0.

[55] B. Cockburn, "Discontinuous Galerkin methods," *ZAMM*, vol. 83, no. 11, pp. 731–754, Nov. 2003, ISSN: 0044-2267. DOI: 10.1002/zamm.200310088. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/zamm.200310088.

[56] D. M. Sklar, J. M. Gimenez, N. M. Nigro, and S. Idelsohn, "Thermal coupling in Particle Finite Element Method - second generation," *Mecánica Computacional*, vol. XXXI, pp. 13–16, 2012.

[57] G. R. Liu and M. B. Liu, *Smoothed Particle Hydrodynamics: A Meshfree Particle Method.* World Scientific, 2003, p. 449, ISBN: 9789812564405. [Online]. Available: https://books.google.be/books?id=%5C_cwFMmEQvZQC.

[58] C. Körner, E. Attar, and P. Heinl, "Mesoscopic simulation of selective beam melting processes," *Journal of Materials Processing Technology*, vol. 211, no. 6, pp. 978–987, 2011, ISSN: 09240136. DOI: 10.1016/j.jmatprotec.2010.12.016.

[59] A. D. Brent, V. R. Voller, and K. J. Reid, "Enthalpy-Porosity technique for modeling convection-diffusion phase change: Application to the melting of a pure metal," *Numerical Heat Transfer*, vol. 13, no. 3, pp. 297–318, Apr. 1988, ISSN: 0149-5720. DOI: 10.1080/10407788808913615. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/10407788808913615.

[60] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, Nov. 1988, ISSN: 00219991. DOI: 10.1016/0021-9991(88)90002-2. arXiv: 1011.1669. [Online]. Available: http://arxiv.org/abs/1011.1669%20http://dx.doi.org/10.1088/1751-8113/44/8/085201%20https://linkinghub.elsevier.com/retrieve/pii/0021999188900022.

[61] C. Hirt and B. Nichols, "Volume of fluid (VOF) method for the dynamics of free boundaries," *Journal of Computational Physics*, vol. 39, no. 1, pp. 201–225, Jan. 1981, ISSN: 00219991. DOI: 10.1016/0021-9991(81)90145-5. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/0021999181901455.

[62] E. M. Kolahdouz, A. P. S. Bhalla, B. A. Craven, and B. E. Griffith, "An immersed interface method for discrete surfaces," *Journal of Computational Physics*, vol. 400, p. 108 854, Jan. 2020, ISSN: 00219991. DOI: 10.1016/j.jcp.2019.07.052. arXiv: 1812.06840. [Online]. Available: https://doi.org/10.1016/j.jcp.2019.07.052%20https://linkinghub.elsevier.com/retrieve/pii/S0021999119305388.

[63] N. Moës, J.-F. Remacle, J. Lambrechts, B. Lé, and N. Chevaugeon, "The eXtreme Mesh deformation approach (X-MESH) for the Stefan phase change model," *Journal of Computational Physics*, vol. 477, p. 111 878, Mar. 2023, ISSN: 00219991. DOI: 10.1016/j.jcp.2022.111878. arXiv: 2111.04179. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S002199912200941X.

[64] M. Rieckmann, M. Smuda, P. Stephan, and F. Kummer, "The extended Discontinuous Galerkin method for two-phase flows with evaporation [submitted for publication]," *Journal of Computational Physics*, 2023. [Online]. Available: https://ssrn.com/abstract=4471968.

[65] D. Celentano, E. Oñate, and S. Oller, "A temperature-based formulation for finite element analysis of generalized phase-change problems," *International Journal for Numerical Methods in Engineering*, vol. 37, no. 20, pp. 3441–3465, Oct. 1994, ISSN: 0029-5981. DOI: 10.1002/nme.1620372004. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/nme.1620372004.

[66] S. Kollmannsberger, M. Carraturo, A. Reali, and F. Auricchio, "Accurate Prediction of Melt Pool Shapes in Laser Powder Bed Fusion by the Non-Linear Temperature Equation Including Phase Changes," *Integrating Materials and Manufacturing Innovation*, vol. 8, no. 2, pp. 167–177, Jun. 2019, ISSN: 2193-9764. DOI: `10.1007/s40192-019-00132-9`. [Online]. Available: `http://link.springer.com/10.1007/s40192-019-00132-9`.

[67] S. A. Khairallah, A. A. Martin, J. R. I. Lee, G. Guss, N. P. Calta, J. A. Hammons, M. H. Nielsen, K. Chaput, E. Schwalbach, M. N. Shah, M. G. Chapman, T. M. Willey, A. M. Rubenchik, A. T. Anderson, Y. M. Wang, M. J. Matthews, and W. E. King, "Controlling interdependent meso-nanosecond dynamics and defect generation in metal 3D printing," *Science*, vol. 368, no. 6491, pp. 660–665, May 2020, ISSN: 0036-8075. DOI: `10.1126/science.aay7830`. [Online]. Available: `https://www.science.org/doi/10.1126/science.aay7830`.

[68] Q. Chen, G. Guillemot, C. A. Gandin, and M. Bellet, "Numerical modelling of the impact of energy distribution and Marangoni surface tension on track shape in selective laser melting of ceramic material," *Additive Manufacturing*, vol. 21, no. November 2017, pp. 713–723, 2018, ISSN: 22148604. DOI: `10.1016/j.addma.2018.03.003`. [Online]. Available: `https://doi.org/10.1016/j.addma.2018.03.003`.

[69] D. Grange, A. Queva, G. Guillemot, M. Bellet, J. D. Bartout, and C. Colin, "Effect of processing parameters during the laser beam melting of Inconel 738: Comparison between simulated and experimental melt pool shape," *Journal of Materials Processing Technology*, vol. 289, no. May 2020, 2021, ISSN: 09240136. DOI: `10.1016/j.jmatprotec.2020.116897`.

[70] D. Grange, J. Bartout, B. Macquaire, and C. Colin, "Processing a non-weldable nickel-base superalloy by Selective Laser Melting: role of the shape and size of the melt pools on solidification cracking," *Materialia*, vol. 12, p. 100 686, Aug. 2020, ISSN: 25891529. DOI: `10.1016/j.mtla.2020.100686`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S2589152920301034`.

[71] S. Zhang, G. Guillemot, C.-A. Gandin, and M. Bellet, "A partitioned two-step solution algorithm for concurrent fluid flow and stress–strain numerical simulation in solidification processes," *Computer Methods in Applied Mechanics and Engineering*, vol. 356, pp. 294–324, Nov. 2019, ISSN: 00457825. DOI: `10.1016/j.cma.2019.07.006`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S0045782519304025`.

[72] J. Wang, J. Zhang, L. Liang, A. Huang, G. Yang, and S. Pang, "A line-based flash heating method for numerical modeling and prediction of directed energy deposition manufacturing process," *Journal of Manufacturing Processes*, vol. 73, no. September 2021, pp. 822–838, 2022, ISSN: 15266125. DOI: `10.1016/j.jmapro.2021.11.041`.

[73] C. Panwisawas, Y. Sovani, R. P. Turner, J. W. Brooks, H. C. Basoalto, and I. Choquet, "Modelling of thermal fluid dynamics for fusion welding," *Journal of Materials Processing Technology*, vol. 252, no. April 2017, pp. 176–182, 2018, ISSN: 09240136. DOI: 10.1016/j.jmatprotec.2017.09.019.

[74] J. Shinjo, A. Kutsukake, A. Arote, Y. T. Tang, D. G. McCartney, R. C. Reed, and C. Panwisawas, "Physics-based thermal-chemical-fluid-microstructure modelling of in-situ alloying using additive manufacturing: Composition-microstructure control," *Additive Manufacturing*, vol. 64, no. August 2022, p. 103 428, 2023, ISSN: 22148604. DOI: 10.1016/j.addma.2023.103428.

[75] L. Arbaoui, P. Schrooyen, N. Poletz, and K. Hillewaert, "Powder bed fusion modelling based on Discontinuous Galerkin formulation," in *Proceedings of the 24th International Conference on Material Forming ESAFORM 2021*, vol. 13, Liège, Apr. 2021, pp. 1–11. DOI: 10.25518/esaform21.2441. [Online]. Available: https://popups.uliege.be/esaform21/index.php?id=2441.

[76] R. Koeune and J. P. Ponthot, "A one phase thermomechanical model for the numerical simulation of semi-solid material behavior. Application to thixoforming," *International Journal of Plasticity*, vol. 58, pp. 120–153, 2014, ISSN: 07496419. DOI: 10.1016/j.ijplas.2014.01.004. [Online]. Available: http://dx.doi.org/10.1016/j.ijplas.2014.01.004.

[77] M. Storti, L. A. Crivelli, and S. R. Idelsohn, "Numerical implementation of a discontinuous finite element algorithm for phase-change problems," *Advances in Engineering Software (1978)*, vol. 9, no. 2, pp. 66–73, 1987, ISSN: 01411195. DOI: 10.1016/0141-1195(87)90026-X.

[78] S. R. Idelsohn, M. A. Storti, and L. A. Crivelli, "Archives of Computational Methods in Engineering Numerical methods in phase-change problems," vol. 1, no. May, pp. 49–74, 1994.

[79] Y. Saadlaoui, A. Delache, E. Feulvarch, J. B. Leblond, and J. M. Bergheau, "New strategy of solid/fluid coupling during numerical simulation of thermo-mechanical processes," *Journal of Fluids and Structures*, vol. 99, p. 103 161, 2020, ISSN: 10958622. DOI: 10.1016/j.jfluidstructs.2020.103161. [Online]. Available: https://doi.org/10.1016/j.jfluidstructs.2020.103161.

[80] M. Courtois, M. Carin, P. L. Masson, S. Gaied, and M. Balabane, "A new approach to compute multi-reflections of laser beam in a keyhole for heat transfer and fluid flow modelling in laser welding," *Journal of Physics D: Applied Physics*, vol. 46, no. 50, p. 505 305, Dec. 2013, ISSN: 0022-3727. DOI: 10.1088/0022-3727/46/50/505305. [Online]. Available: https://iopscience.iop.org/article/10.1088/0022-3727/46/50/505305.

[81] W. Pitscheneder, T. DebRoy, K. Mundra, and R. Ebner, "Role of sulfur and processing variables on the temporal evolution of weld pool geometry during multikilowatt laser beam welding of steels," *Welding Journal (Miami, Fla)*, vol. 75, no. 3, 1996, ISSN: 00432296.

[82] G. Buresti, "A note on Stokes' hypothesis," *Acta Mechanica*, vol. 226, no. 10, pp. 3555–3559, 2015, ISSN: 00015970. DOI: 10.1007/s00707-015-1380-9.

[83] J. Valdés, "Nonlinear analysis of orthotropic membrane and shell structures including fluid-structure interaction," Ph.D. dissertation, Universitat Politècnica de Catalunya, 2007, p. 225, ISBN: 9788496736375. [Online]. Available: http://hdl.handle.net/2117/94177.

[84] A. S. Dukhin and P. J. Goetz, "Bulk viscosity and compressibility measurement using acoustic spectroscopy," *The Journal of Chemical Physics*, vol. 130, no. 12, p. 124 519, Mar. 2009, ISSN: 0021-9606. DOI: 10.1063/1.3095471. [Online]. Available: http://aip.scitation.org/doi/10.1063/1.3095471.

[85] L. Rosenhead, "Introduction - The second coefficient of viscosity: a brief review of fundamentals," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 226, no. 1164, pp. 1–6, Oct. 1954, ISSN: 0080-4630. DOI: 10.1098/rspa.1954.0224. [Online]. Available: https://royalsocietypublishing.org/doi/10.1098/rspa.1954.0224.

[86] J. M. Flinn, J. Jarzynski, and T. A. Litovitz, "Mechanism of Volume Viscosity in Molten Bismuth and Lead," *The Journal of Chemical Physics*, vol. 54, no. 10, pp. 4331–4340, May 1971, ISSN: 0021-9606. DOI: 10.1063/1.1674681. [Online]. Available: http://aip.scitation.org/doi/10.1063/1.1674681.

[87] D. A. Dunavant, "High degree efficient symmetrical Gaussian quadrature rules for the triangle," *International Journal for Numerical Methods in Engineering*, vol. 21, no. 6, pp. 1129–1148, Jun. 1985, ISSN: 0029-5981. DOI: 10.1002/nme.1620210612. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/nme.1620210612.

[88] R. S. Falk, *Mixed and Hybrid Finite Element Methods (Franco Brezzi and Michel Fortin)*. 1993, vol. 35, pp. 514–517, ISBN: 9783642365188. DOI: 10.1137/1035113.

[89] T. E. Tezduyar, S. Mittal, S. E. Ray, and R. Shih, "Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements," *Computer Methods in Applied Mechanics and Engineering*, vol. 95, no. 2, pp. 221–242, Mar. 1992, ISSN: 00457825. DOI: 10.1016/0045-7825(92)90141-6. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/0045782592901416.

[90] M. Cerquaglia, D. Thomas, R. Boman, V. Terrapon, and J.-P. Ponthot, "A fully partitioned Lagrangian framework for FSI problems characterized by free surfaces, large solid deformations and displacements, and strong added-mass effects," eng, *Computer Methods in Applied Mechanics and Engineering*, vol. 348, pp. 409–442, May 2019, ISSN: 00457825. DOI: 10.1016/j.cma.2019.01.021. [Online]. Available: https://doi.org/10.1016/j.cma.2019.01.021%20https://linkinghub.elsevier.com/retrieve/pii/S0045782519300428.

[91] J.-P. Ponthot, R. Boman, and L. Papeleux, *Metafor, A nonlinear finite element code.* [Online]. Available: `http://metafor.ltas.ulg.ac.be`.

[92] E. Oñate, M. A. Celigueta, and S. R. Idelsohn, "Modeling bed erosion in free surface flows by the particle finite element method," *Acta Geotechnica*, vol. 1, no. 4, pp. 237–252, 2006, ISSN: 18611133. DOI: `10.1007/s11440-006-0019-3`.

[93] R. Aubry, S. R. Idelsohn, and E. Oñate, "Particle finite element method in fluid-mechanics including thermal convection-diffusion," *Computers and Structures*, vol. 83, no. 17-18, pp. 1459–1475, 2005, ISSN: 00457949. DOI: `10.1016/j.compstruc.2004.10.021`.

[94] J. M. Gimenez, N. M. Nigro, and S. R. Idelsohn, "Evaluating the performance of the particle finite element method in parallel architectures," *Computational Particle Mechanics*, vol. 1, no. 1, pp. 103–116, May 2014, ISSN: 2196-4378. DOI: `10.1007/s40571-014-0009-4`. [Online]. Available: `http://link.springer.com/10.1007/s40571-014-0009-4`.

[95] J. R. Shewchuk, "Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1148, 1996, pp. 203–222, ISBN: 354061785X. DOI: `10.1007/BFb0014497`. [Online]. Available: `http://link.springer.com/10.1007/BFb0014497`.

[96] R. Falla, B.-J. Bobach, R. Boman, J.-P. Ponthot, and V. E. Terrapon, "Mesh adaption for two-dimensional bounded and free-surface flows with the particle finite element method," *Computational Particle Mechanics*, Jan. 2023, ISSN: 2196-4378. DOI: `10.1007/s40571-022-00541-2`. [Online]. Available: `https://doi.org/10.1007/s40571-022-00541-2%20https://link.springer.com/10.1007/s40571-022-00541-2`.

[97] M. L. Cerquaglia, G. Deliége, R. Boman, V. Terrapon, and J.-P. Ponthot, "Free-slip boundary conditions for simulating free-surface incompressible flows through the particle finite element method," *International Journal for Numerical Methods in Engineering*, vol. 110, no. 10, pp. 921–946, Jun. 2017, ISSN: 00295981. DOI: `10.1002/nme.5439`. [Online]. Available: `https://onlinelibrary.wiley.com/doi/10.1002/nme.5439`.

[98] D. Annaratone, *Engineering Heat Transfer.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, ISBN: 978-3-642-03931-7. DOI: `10.1007/978-3-642-03932-4`. [Online]. Available: `http://link.springer.com/10.1007/978-3-642-03932-4`.

[99] G. D. E. V. Davis and I. P. Jones, "Natural convection in a Square," *International Journal for Numerical Methods in Fluids*, vol. 3, no. July 1982, pp. 227–248, 1983, ISSN: 15383644. [Online]. Available: `http://onlinelibrary.wiley.com/doi/10.1002/fld.1650030304/abstract`.

[100] S. F. Corzo, S. M. Damián, D. Ramajo, and M. N. Norberto, "Numerical Simulation of Natural Convection Phenomena," *Mecánica Computacional*, vol. XXX, pp. 277–296, 2011.

[101] T. L. Bergman and J. R. Keller, "Combined buoyancy, surface tension flow in liquid metals," *Numerical Heat Transfer*, vol. 13, no. 1, pp. 49–63, Jan. 1988, ISSN: 0149-5720. DOI: 10.1080/10407788808913603. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/10407788808913603.

[102] W. D. Rolph and K.-J. Bathe, "An efficient algorithm for analysis of nonlinear heat transfer with phase changes," *International Journal for Numerical Methods in Engineering*, vol. 18, no. 1, pp. 119–134, Jan. 1982, ISSN: 0029-5981. DOI: 10.1002/nme.1620180111. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/nme.1620180111.

[103] K. Morgan, R. W. Lewis, and O. C. Zienkiewicz, "An improved algrorithm for heat conduction problems with phase change," *International Journal for Numerical Methods in Engineering*, vol. 12, no. 7, pp. 1191–1195, 1978, ISSN: 0029-5981. DOI: 10.1002/nme.1620120710. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/nme.1620120710.

[104] H. Budhia and F. Kreith, "Heat transfer with melting or freezing in a wedge," *International Journal of Heat and Mass Transfer*, vol. 16, no. 1, pp. 195–211, Jan. 1973, ISSN: 00179310. DOI: 10.1016/0017-9310(73)90262-7. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/0148906274920828.

[105] B.-J. Bobach, R. Falla, R. Boman, V. Terrapon, and J.-P. Ponthot, "Phase change driven adaptive mesh refinement in PFEM," *ESAFORM 2021*, vol. 13, pp. 2–7, Apr. 2021. DOI: 10.25518/esaform21.3861. [Online]. Available: https://popups.uliege.be/esaform21/index.php?id=3861.

[106] C. Gau and R. Viskanta, "Melting and solidification of a pure metal on a vertical wall," *Journal of Heat Transfer*, vol. 108, no. 1, pp. 174–181, 1986, ISSN: 15288943. DOI: 10.1115/1.3246884.

[107] B.-J. Bobach, R. Boman, D. Celentano, V. E. Terrapon, and J.-P. Ponthot, "Simulation of the Marangoni Effect and Phase Change Using the Particle Finite Element Method," *Applied Sciences*, vol. 11, no. 24, p. 11 893, Dec. 2021, ISSN: 2076-3417. DOI: 10.3390/app112411893. [Online]. Available: https://www.mdpi.com/2076-3417/11/24/11893.

[108] S. Tiari, A. Hockins, and M. Mahdavi, "Numerical study of a latent heat thermal energy storage system enhanced by varying fin configurations," *Case Studies in Thermal Engineering*, vol. 25, no. November 2020, p. 100 999, Jun. 2021, ISSN: 2214157X. DOI: 10.1016/j.csite.2021.100999. [Online]. Available: https://doi.org/10.1016/j.csite.2021.100999%20https://linkinghub.elsevier.com/retrieve/pii/S2214157X21001623.

[109] A. Sharma, M. Trivedi, K. Agarwal, and N. Nirmalkar, "Thermal energy storage in a confined cylindrical heat source filled with phase change materials," *International Journal of Heat and Mass Transfer*, vol. 178, p. 121 603, Oct. 2021, ISSN: 00179310. DOI: 10.1016/j.ijheatmasstransfer.2021.121603. [Online]. Available: https://doi.org/10.1016/j.ijheatmasstransfer.2021.121603%20https://linkinghub.elsevier.com/retrieve/pii/S0017931021007067.

[110] D. Thomas, M. Cerquaglia, R. Boman, T. Economon, J. Alonso, G. Dimitriadis, and V. Terrapon, "CUPyDO - An integrated Python environment for coupled fluid-structure simulations," *Advances in Engineering Software*, vol. 128, no. May 2018, pp. 69–85, Feb. 2019, ISSN: 09659978. DOI: 10.1016/j.advengsoft.2018.05.007. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0965997817310906.

[111] A. Crovato, R. Boman, H. Guner, V. E. Terrapon, H. S. Almeida, A. P. Prado, C. Breviglieri, P. H. Cabral, and G. H. Silva, "A full potential static aeroelastic solver for preliminary aircraft design," in *Proceedings of the 18th International Forum on Aeroelasticity and Structural Dynamics (IFASD2019)*, Savannah, Georgia, USA, 2019, pp. 1–14.

[112] A. Crovato, H. S. Almeida, G. Vio, G. H. Silva, A. P. Prado, C. Breviglieri, H. Guner, P. H. Cabral, R. Boman, V. E. Terrapon, and G. Dimitriadis, "Effect of Levels of Fidelity on Steady Aerodynamic and Static Aeroelastic Computations," *Aerospace*, vol. 7, no. 4, p. 42, Apr. 2020, ISSN: 2226-4310. DOI: 10.3390/aerospace7040042. [Online]. Available: https://www.mdpi.com/2226-4310/7/4/42.

[113] E. Fernández, S. Février, M. Lacroix, R. Boman, L. Papeleux, and J. P. Ponthot, "A particle finite element method based on Level–Set functions," *Journal of Computational Physics*, vol. 487, 2023, ISSN: 10902716. DOI: 10.1016/j.jcp.2023.112187.

[114] J. B. Leblond, H. A. El Sayed, and J. M. Bergheau, "On the incorporation of surface tension in finite-element calculations," *Comptes Rendus - Mecanique*, vol. 341, no. 11-12, pp. 770–775, 2013, ISSN: 16310721. DOI: 10.1016/j.crme.2013.10.004. [Online]. Available: http://dx.doi.org/10.1016/j.crme.2013.10.004.