Level set-based topology optimization of thermal fluid-structure systems

LI HAO
2022

KYOTO UNIVERSITY

DOCTORAL THESIS

Level Set-based Topology Optimization of Thermal Fluid–structure Systems

Author:
Hao Li

Advisor:
Professor Shinji Nishiwaki
Co-advisor:
Professor Kazuhiro Izui

Committee member:
Professor Tomoko Hirayama (chair)
Professor Hiroshi Iwai
Professor Atsushi Matsubara

A thesis submitted in fulfillment of the requirements for the Doctor of Philosophy

of the

Department of Mechanical Engineering and Science
Manufacturing System Engineering Lab
Graduate School of Engineering

July 1, 2022

Declaration of Authorship

I, Hao Li, declare that this thesis titled, "Level Set-based Topology Optimization of Thermal Fluid–structure Systems" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly 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:

"We are not now that strength which in old days
Moved earth and heaven, that which we are, we are;
One equal temper of heroic hearts,
Made weak by time and fate, but strong in will
To strive, to seek, to find, and not to yield."

Alfred, Lord Tennyson

Acknowledgements

I would like to thank the following people and many others, without whom I would not have been able to complete this research project, and without whom I would not have made it through my doctoral degree.

First of all, I would like to sincerely thank my supervisors Professor Shinji Nishiwaki, Professor Kazuhiro Izui, and Professor Takayuki Yamada for their dedicated support and guidance. They continuously provide me encouragement and are always willing and enthusiastic to assist in any way they could throughout my research. I have benefited greatly from their wealth of knowledge and meticulous editing. I am extremely grateful that Professor Nishiwaki took me on as a student and continue to have faith in me over the years.

I would like to express my sincere gratitude and appreciation for Dr. Tsuguo Kondoh. His support, guidance and overall insights in the fields of fluid mechanics and optimization have made this an unusual and inspiring experience for me. I would also like to thank Dr. Kozo Furuta whose guidance, support and encouragement has been invaluable throughout my study and daily life. Big thanks to all the members in the Manufacturing Systems Engineering Laboratory (MSEL) at Kyoto University who support me in ways that gives me a stable and sustainable environment for advancing our research work.

I would like to say a special thank you to Dr. Pierre Jolivet at Institut de Recherche en Informatique de Toulouse for guiding me to implement large-scale finite element analysis and for his great contribution to our joint publications. The meetings and conversations were vital in inspiring me to think outside the box, from multiple perspectives to form a comprehensive and objective critique.

Some parts of this thesis are inspired by the pioneering works by Professor Joe Alexandersen at University of Southern Denmark who firstly solved the natural convection topology optimization problems, and Professor Renato Picelli at University of São Paulo who has intensively investigated the fluid–structure interaction topology optimization problems. And some of my ideas were developed based on Joe and Renato's comments and suggestions. Immense gratitude as always to them for their patience and support.

I also would like to express my gratitude to co-researchers for their invaluable advise, stimulating discussions, and giving me the opportunity to expose me to the state-of-the-art studies during collaborative research. I received support on compliant mechanism design from Professor Benliang Zhu in the School of Mechanical Engineering at South China University of Technology. I got thoughtful advice on thermal fluid topology optimization from Professor Kentaro Yaji, who was in our laboratory and is in the Department of Mechanical Engineering at Osaka University. Fruitful discussions with Professor Atsushi Suzuki in Cybermedia Center at Osaka University are gratefully acknowledged. A special thanks to my former master degree's advisor Professor Xiaohong Ding, my former co-advisor Professor Dalei Jing, my former colleagues Dr. Heng Zhang and Dr. Min Xiong in the School of Mechanical Engineering at University of Shanghai for Science and Technology, for their continuous support even after my graduation. I am thankful to Dr. Minghao Yu with whom I had spend several months studying together in their lab in the Department of

Engineering Mechanics at Dalian University of Technology. I could not have accomplished my doctoral research without their kind help.

Also, I am grateful to my colleagues and friends in academia and industrial fields including Dr. Mojtaba Barzegaria in the Department of Mechanical Engineering, at KU Leuven for his helpful advice and discussions about the biodegradable material designs, Mr. Sheng Pan in the Department of Engineering Mechanics at Dalian University of Technology for stimulating discussions about thermal fluid optimum designs, Dr. Yulin Xiong in the Centre for Innovative Structure and Materials at RMIT University for his helpful advice about lattice infill designs. A special thanks to Mr. Tiannan Hu and Mr. Kousei Wano in MSEL for their technical support on the lattice infill design. I am always impressed by their talent and skills. Another special thanks to Dr. Ryoto Misawa who was a postdoc researcher in our laboratory and is an assistant professor at Saitama University, for his kind help in constructing our computing platform. A special thanks also to Dr. Mario Petrovic in Sumitomo Precision Products Co., Ltd. and Dr. Atsushi Koguchi in Cybernet Systems Co., Ltd. who provided me many helpful comments and suggestions which made my coding experience more productive and enjoyable.

I am also thankful to the developers of Mmg platform, especially to Dr. Algiane Froehly for all their considerable efforts to develop a powerful and an easy-to-use open-source numerical tool for the mesh adaptation.

I am indebted also to our English teacher, Mr. Steve Wever, for his continued help in editing my papers in English and for improving my academic writing skills. This is reflected by the concise but clear writing style of our papers, which is something I hope to carry forward throughout my future academic career.

I also sincerely appreciate the financial support from

1. Grant-in-Aid for the Japan Society for the Promotion of Science Research Fellow (Grant Number JP21J13418);

2. Mr. Hiroshi Ogawa and his colleagues at DENSO Co., Ltd. for their financial support and their advice with regards to the HPC on Rescale cluster (www.rescale.com).

Last but not the least, I cannot forget to thank my parents and my family for their unconditional support in these very intense academic years.

Hao Li
Kyoto, July, 2022

KYOTO UNIVERSITY

Abstract

Manufacturing System Engineering Lab
Graduate School of Engineering

Doctor of Philosophy

Level Set-based Topology Optimization of Thermal Fluid–structure Systems

by Hao Li

This thesis focuses on topology optimization (TO) of thermal fluid–structure systems motivated by aeronautic and thermal management industrial applications. Topological design sensitivity of arbitrary cost functionals are derived for a weakly coupled thermal fluid-structure model. A reaction–diffusion equation-based level-set method is then developed for solving generic constrained topology optimization problems with which one can design from scratch, that is, new holes (resp. islands) can be nucleated during topological evolution. Motivated by the need for real-world applications, two key ingredients are introduced into this workflow. The first is the physics-tailored multigrid preconditioner for distributed finite element analysis. This ensures that the physical computation part of the TO framework can be highly scalable with respect to the problem size. The second is the two different unstructured mesh adaptation techniques. More specifically, body-fitted meshes, as one of the surface capturing techniques, allows the disjoint-reunion of a global mesh that involves several (fluid/solid) subdomains. Anisotropic meshes fit high aspect ratio elements (highly stretched elements) along the regions of rapid variation of the solutions like interior or boundary layers. All these ingredients allowed us to solve a variety of two- and three-dimensional multiphysics test cases, from single physics problems in 2D to coupled physics in large-scale 3D setting including minimal mean compliance, minimal power dissipation, design dependent and design independent-fluid-structure interaction (FSI), natural/-forced convection, lift–drag problems. The final opening chapter sheds a light on the lattice designing. Motivated by the need for porous structure in the design of biodegradable implants, a variational method (PDE-filter) is used in order to simplify the numerical evaluation of geometric constraints: it enables to compute “local averaged” characteristic function on an unstructured mesh by solving this PDE without requiring the neighborhood element spatial information.

Keywords: topology optimization, level-set method, mesh adaptation, distributed finite elements, power dissipation, lift–drag, fluid–structure interaction, natural convection, forced convection, lattice designing, FreeFEM-PETSc-Mmg

Contents

List of Figures

List of Tables

List of Abbreviations

AD	**A**utomatic **D**ifferential
AM	**A**dditive **M**anufacturing
BC	**B**oundary **C**ondition
BESO	**B**i-directional **E**volutionary **S**tructural **O**ptimization
B-Reps	**B**oundary **R**epresentation
cf.	**C**ompare
CR	**C**oncentration **R**atio
CSRBF	**C**ompactly **S**upported **R**adiul **B**asis **F**unction
DSL	**D**omain-**S**pecific **L**anguage
ESO	**E**volutionary **S**tructural **O**ptimization
FDM	**F**inite **D**ifference **M**ethod
FEA	**F**inite **E**lement **A**nalysis
FEM	**F**inite **E**lement **M**ethod
FSI	**F**luid–**S**tructure **I**nteraction
FVM	**F**inite **V**olume **M**ethod
GMG	**G**alerkin-**P**rojection **G**eometric
GPU	**G**raphics **P**rocessing **U**nit
HCPV	**H**igh-**C**oncentration **P**hotovoltaic
hex.	**h**exahedral
HJE	**H**amilton-**J**acobian **E**quation
HS	**H**eat **S**ink
ISPH	**I**ncompressible **S**moothed **P**article **H**ydrodynamics
LBM	**L**attice **B**oltzmann **M**ethod
LED	**L**ight-**E**mitting **D**iode
LSM	**L**evel-**S**et **M**ethod
KKT	**K**arush-**K**uhn-**T**ucker
mAL	**m**odified **A**ugmented **L**agrangian
MMA	**M**ethod of **M**oving **A**sympototes
MMC	**M**oving **M**orphable **C**omponent
MMV	**M**oving **M**orphable **V**oid
MPI	**M**essage **P**assing **I**nterface
MPS	**M**oving **P**article **S**emi-implicit
MESL	**M**anufacturing **S**ystems **E**ngineering **L**aboratory
NS	**N**avier–**S**tokes
OC	**O**ptimality **C**riteria
PCD	**P**ressure **C**onvection **D**iffusion
RAMP	**R**ational **A**pproximation of **M**aterial **P**roperty
RANS	**R**eynolds-**A**veraged **N**avier–**S**tokes
ROI	**R**egion-**o**f-**I**nterest
PETSc	**P**ortable and **E**xtendable **T**oolkit

RDE	**R**eaction **D**iffusion **E**quation
resp.	**R**espectively
SIMP	**S**olid **I**sotropic **M**aterial with **P**enalization
STL	**S**tereolithography
tetra.	**t**etrahedral
TO	**T**opology **O**ptimization
TOBS	**T**opology **O**timization of **B**inary **S**tructures
XFEM	**E**xtended **F**inite **E**lement **M**ethod

List of Symbols

$\boldsymbol{e}$	unit vector
$\boldsymbol{g}$	surface traction
$\boldsymbol{n}$	outward normal vector of the boundary
$\boldsymbol{u}$	displacement
$\boldsymbol{v}$	velocity
$\boldsymbol{x}$	coordinate
Δt	fictitious time step for level-set evolution equation
δ	increment
$\mathbb{A}$	material property tensor
$\mathbb{C}$	elasticity tensor
$\mathbf{f}$	force vector
$\mathbf{K}$	Stiffness matrix
$\mathbf{Tr}\,(\cdot)$	Trace operator
$\mathbf{u}$	displacement vector
$\mathbf{x}$	design variable
$\mathcal{F}'$	design sensitivity
$\mathcal{G}'$	sensitivity associated to constraint functional
$\mathcal{L}$	Lagrangian
$\mathcal{T}$	mesh
I	identity matrix
∇	gradient operator
div	divergence operator
Gr	Grashoff number
Pr	Prandtl number
Ra	Rayleigh number
Re	Reynolds number
$\tilde{C}$	normalizer for the design sensitivity
C_p	specific heat
d_tF	topological derivative
E	Young's modulus
e	strain tensor
G	constraint functional
H	Heaviside function
H^1	Hilbert space
h_{grad}	mesh size gradation
J	objective functional
L	characteristic length
L	lower limit of the moving asymptotes
l_e	mesh edge size
m	moving limit
p	pressure
Q	heat source
q	penalty factor
r	PDE filter radius
T	temperature
t	fictitious time
U	upper limit of the moving asymptotes
V	characteristic velocity
$V_{\max}$	maximum allowed volume fraction

Subscripts

max	maximum

min	minimum
f	fluid
ref	reference
s	solid
A	adjoint variable
e	element
it	iterative step
p	MPI process

Greek symbols

α	inverse permeability
α_i	sensitivity number in BESO method
$\bar{\chi}$	averaged characteristic function
χ	characteristic function
ϵ	radius
η	numerical damping coefficient
Γ	boundary
γ	pseudo density
κ	thermal conductivity
λ	Lagrange multiplier
μ	Lamé coefficient, or dynamic viscosity [$\mathrm{kg\,m^{-1}\,s^{-1}}$]
ν	Poisson's ratio
Ω	computational domain
ω	heat source region
Φ	power dissipation
ϕ	level-set function
ρ	density [$\mathrm{kg\,m^{-3}}$]
σ	stress tensor
τ	regularization parameter
$\tilde{\gamma}$	filtered pseudo density
ε	tolerence, or perturbation
ξ	dimensionless height

Superscripts

$*$	dimensional variable
T	transpose
vm	Von Mises
~	test function
D	Dirichlet boundary
N	Neumann boundary
p	$p-$norm

Dedicated to my parents and my grandmother

To my friends

Chapter 1

General Introduction

CONTENTS

1.1 Introduction

Structural optimization has been widely used in many real-world applications, such as automotive industries, aviation industries, due to its great potential for shortening the product development time. Generally, the structural optimization can be classified into three categories: sizing, shape, and topology optimization. Fig. 1.1 shows the schematic diagram of these three structural optimization methods. Sizing optimization, as the oldest approach, was proposed in the 1960's (Schmit, 1960). Sizing optimization is necessary if one is seeking for an accurate design by means of fine-tune a few CAD parameters. Shape optimization was proposed in 1970's (Gallagher and Zienkiewicz, 1973), the key idea is to optimize a given functional performance by means of moving structural boundaries. Unlike the sizing or shape optimization methods, topology optimization (TO) is well-known for its high flexibility to attain any shape within a given design space. Therefore, TO methods are a suitable mathematical tool at the conceptual design stage which can be used to develop free-of-form generative designs. Since the first work by Bendsøe and Kikuchi (1988), TO has undergone a tremendous develop in versatile directions, and it has been applied in a variety of problems, including multi-physics problems, such as thermal-fluidic device

design (Alexandersen and Andreasen, 2020; Fawaz, Hua, Le Corre, Fan, and Luo, 2022), acoustic metamaterial design (Noguchi, Yamada, Otomori, Izui, and Nishiwaki, 2015), biodegradable implant design (Zhang, Takezawa, Ding, Zhang, Xu, Li, Nozawa, and Nishiwaki, 2022; Zhang, Takezawa, Ding, Xu, Duan, Li, and Guo, 2021), and so on. In recent decades, there are more and more commercial software that can perform TO techniques (XIE-Engineering-Technologies., 2021; COMSOL, 2018). This thesis builds upon the recent advancement of TO techniques to construct a level-set based topology optimization of the large-scale multiphysics systesm. More specifically, this thesis focuses on the optimum design of thermal fluid–structure problems.

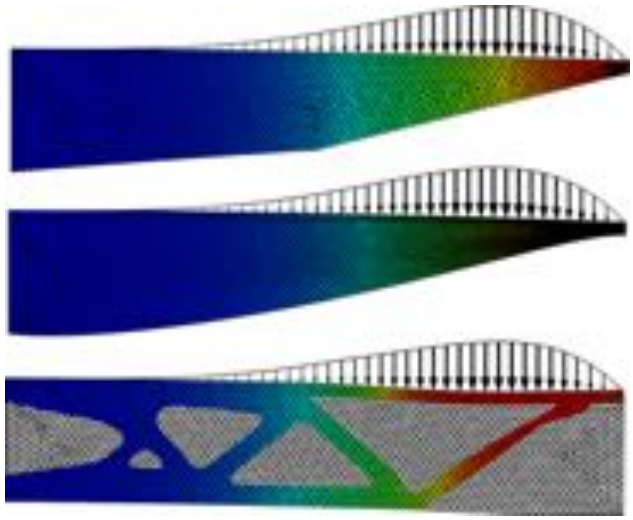


FIGURE 1.1: Schematic of the size (top), shape (middle), and topology (bottom) optimization (www.comsol.com).

The remainder of this chapter is organized as follows. In Section 1.2, we overview the basic concept of the topology optimization and we introduce the mathematical frames of the three popular TO methods. in Section 1.3, we illustrate the mathematical background of the sensitivity analysis and illustrate three different methods to compute the sensitivity. In Section 1.4, we describe the motivation and our objective of this thesis. Lastly, we describe in Section 1.5 the structure of this thesis.

1.2 Shape and topology optimization methods

In this section, we overview the basic concept of the topology optimization. Then we introduce the basic idea of three TO methods: density approach, evolutionary approach, and level-set method (LSM). For each method, we illustrate the updating scheme for the design variable and summarize its features (i.e., merits and demerits).

Please note that there are some other TO methods including the bionic method such as the adaptive growth method (Ding and Yamazaki, 2005) and the feature driven method such as the

moving morphable component method (MMC) approach (Guo, Zhang, and Zhong, 2014), moving morphable void (MMV) approach (Zhang, Chen, Zhu, Zhou, Xue, Lei, and Guo, 2017) and moving wide bezier components with constrained ends (Zhu, Wang, Wang, Li, Zhang, and Nishiwaki, 2021). However, these are not the focus of this thesis, and will not be further discussed here.

Now we discuss the formulation of a topology optimization problem. Let us take the classical mean compliance problem as an example, a TO problem is defined to include both solid and void subdomains. Let the computational domain be denoted as $D \subset \mathbb{R}^d$, where $d = 2$ or 3. A solid subdomain is defined as $\Omega \subseteq D$ and a void domain is represented as the complementary domain $D \backslash \Omega$. Then we use the characteristic function $\chi(\mathbf{x})$ to represent the material distribution such that

$$\chi(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega \\ 0 & \text{if } \mathbf{x} \in D \backslash \Omega, \end{cases} \tag{1.1}$$

where $\mathbf{x}$ represents the position in the design domain.

We intend to maximize or minimize a functional performance, (i.e., minimal mean compliance) under the given constraints (i.e., maximum allowed volume fraction). Therefore, the general formulation of optimization mathematical model can be define as

$$\begin{aligned} \min_{\mathbf{x} \in \mathcal{X}} \quad & f(\mathbf{x}) \quad \mathbf{x} = [x_1, x_2, \ldots, x_n]^{\mathrm{T}} \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq \bar{g}_i \qquad\qquad i \in [1, m] \\ & x_j \in \left[x_{j_{\min}}, x_{j_{\max}}\right] \qquad j \in [1, n], \end{aligned} \tag{1.2}$$

where $f(\mathbf{x})$ is the objective functional, $g_i(\mathbf{x})$ is the constraint. $x_{j_{\min}}$ and $x_{j_{\max}}$ are the lower and upper bound of the design variable value.

1.2.1 Density approach

Density approach is one of the most popular TO methods. In this approach, the characteristic function $\chi(\mathbf{x})$ in Eq. (1.1) is replaced by a continuous function $\gamma(\mathbf{x})$, named as the pseudo density.

Using the solid isotropic material with penalization (SIMP) (Bendsøe, 1989), the density field can be expressed as

$$\gamma(\mathbf{x}) = \mathbf{x}^p, \tag{1.3}$$

or using the rational approximation of material properties (RAMP) (Stolpe and Svanberg, 2001) as

$$\gamma(\mathbf{x}) = \frac{\mathbf{x}}{1 + q(1 - \mathbf{x})}, \tag{1.4}$$

where $p > 1$, $q > 1$ are the penalty factors for the SIMP and RAMP schemes, respectively.

The key idea is to use this continuous density field to represent the existence or absence of a material phase. Therefore, a material property tensor $\mathbb{A}$, i.e., elasticity tensor in elasticity, inverse permeability in fluid flows, can be expanded as

$$\mathbb{A} = \gamma(\mathbf{x})\,\mathbb{A}_0, \tag{1.5}$$

where $\mathbb{A}_0$ is the material property tensor of a given material in the domain Ω.

The design variable $\mathbf{x}$ can be updated using different approaches such as the optimality criteria (OC) method, sequential programming method, or the method of moving asymptotes (MMA).

In the early works on density method (Sigmund, 2001), an optimality criteria (OC) method was adopted. The basic idea is to construct an system of equations to update the design variable based on the Karush-Kuhn-Tucker (KKT) conditions, also known as the Kuhn–Tucker conditions. One heuristic updating scheme (Bendsøe, 1995) can be formulated as

$$x_i^{it+1} = \begin{cases} \max(x_{\min}, x_i - m) & \text{if} \quad x_i B_i^{\eta} \leq \max(x_{\min}, x_i - m) \\ x_i B_i^{\eta} & \text{if} \quad \max(x_{\min}, x_i - m) < x_i B_i^{\eta} < \min(1, x_i + m) \\ \min(1, x_i + m) & \text{if} \quad \min(1, x_i + m) \leq x_i B_i^{\eta}, \end{cases} \tag{1.6}$$

where m is a positive moving limit, η is a numerical damping coefficient and B_e is derived based on the optimality criteria as

$$B_i = \frac{-\dfrac{\partial f}{\partial x_i}}{\lambda \dfrac{\partial g}{\partial x_i}}, \tag{1.7}$$

where λ is a Lagrange multiplier that can be calculated by a bi-sectioning algorithm. The OC method can have some drawbacks especially when treating multiple constraints TO problems. An educational code written in FreeFEM language can be found in Appendix B.3. For detailed explanations, readers are referred to in our recent educational paper (Zhu, Zhang, Li, Liang, Wang, Li, and Nishiwaki, 2021).

Another widely used method is the MMA method proposed by Svanberg (1987). The idea behind is to solve a series of approximate sub-problems rather than the original non-linear problem. The approximations should be convex in order to obtain a numerically tractable optimization problem. Since the approximation is only valid in the vicinity of the expansion point, the process of generating and solving a sub-problem must be repeated until a specified convergence criteria is fulfilled. In MMA, the approximations are based on convex functions of the type as

$$g_i^{it}(\mathbf{x}) = \sum_{j=1}^{n} \left(\frac{p_{ij}^{it}}{U_j^{it} - x_j} + \frac{q_{ij}^{it}}{x_j - L_j^{it}} \right) + r_i^{it}, \tag{1.8}$$

where

$$p_{ij}^{it} = \begin{cases} \left(U_j^{it} - x_j^{it}\right)^2 \partial g_i / \partial x_j & \text{if } \partial g_i / \partial x_j > 0 \\ 0 & \text{if } \partial g_i / \partial x_j \leq 0, \end{cases} \tag{1.9}$$

$$q_{ij}^{it} = \begin{cases} 0 & \text{if } \partial g_i / \partial x_j \geq 0 \\ -\left(x_j^{it} - L_j^{it}\right)^2 \partial g_i / \partial x_j & \text{if } \partial g_i / \partial x_j < 0, \end{cases} \tag{1.10}$$

$$r_i^{it} = g_i\left(\mathbf{x}^{it}\right) - \sum_{j=1}^{n} \left(\frac{p_{ij}^{it}}{U_j^{it} - x_j^{\text{it}}} + \frac{q_{ij}^{it}}{x_j^{it} - L_j^{it}} \right), \tag{1.11}$$

where $\partial g_i / \partial x_j$ is the sensitivity of the i-th constraint function with respect to the j-th design variable at the point $\mathbf{x} = \mathbf{x}^{it}$. U and L are the upper and lower bound of the moving asymptotes. See details of a parallel generation and solution of the MMA sub-problem in Aage and Lazarov (2013).

1.2.2 Evolutionary method & binary structure approach

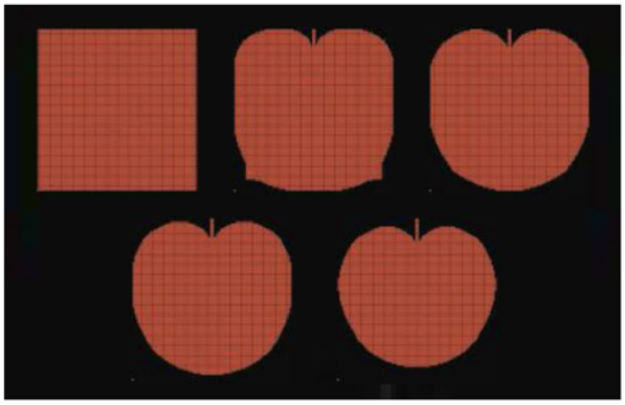


FIGURE 1.2: ESO solutions of an object hanging in the air under its own weight (Xie and Steven, 1997).

ESO and BESO method

The evolutionary structural optimization (ESO) method was originally proposed by Xie and Steven (1993). Inspired by the evolution of naturally occurring structures. For example, Fig. 1.2 shows an object hanging in the air under its own weight. the original idea of the ESO method was to use the stress level as an indicator for the gradual removal of inefficient material for a structure.

In ESO method, the material property tensor is expanded in a similar way as in Eq. (1.5). However, unlike the density approach, ESO method imposes the design variable to be 1/0 binary structure which can avoid the misinterpretation of the optimal solution. Therefore, the optimization

can be formulated as

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{X}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq \bar{g}_i \quad i \in [1,m] \\ & x_j \in \{0,1\} \quad j \in [1,n]\,. \end{aligned} \tag{1.12}$$

The early ESO method was limited by allowing only material removal from the structure. Later in Querin, Steven, and Xie (1998), they combined the removal and addition mechanism and proposed a bi-direction ESO (BESO) method. The element can be removed or added using a heuristic-based criteria as

$$\begin{cases} \sigma_e^{\text{vm}} < c_{\text{rr}} \cdot \sigma_{\max}^{\text{vm}} \rightarrow \text{ element-removal} \\ \tilde{\sigma}_e^{\text{vm}} > c_{\text{ir}} \cdot \sigma_{\max}^{\text{vm}} \rightarrow \text{ element-addition,} \end{cases} \tag{1.13}$$

where $\sigma_{\max}^{\text{vm}}$ is a given value used to evaluate the efficiency of an element inside the design domain.

In Chu, Xie, Hira, and Steven (1996), a so-called sensitivity number α_i was introduced as the variation of element compliance due to the removal or addition of the element as

$$\alpha_i = \frac{\partial f}{\partial x_i}, x_i \in \{0,1\} \quad i \in [1,n]\,. \tag{1.14}$$

This can be used the indicator for the ranking of elemental sensitivities. For more comprehensive review on ESO method, the readers are referred to in Xia, Xia, Huang, and Xie (2018).

TOBS method

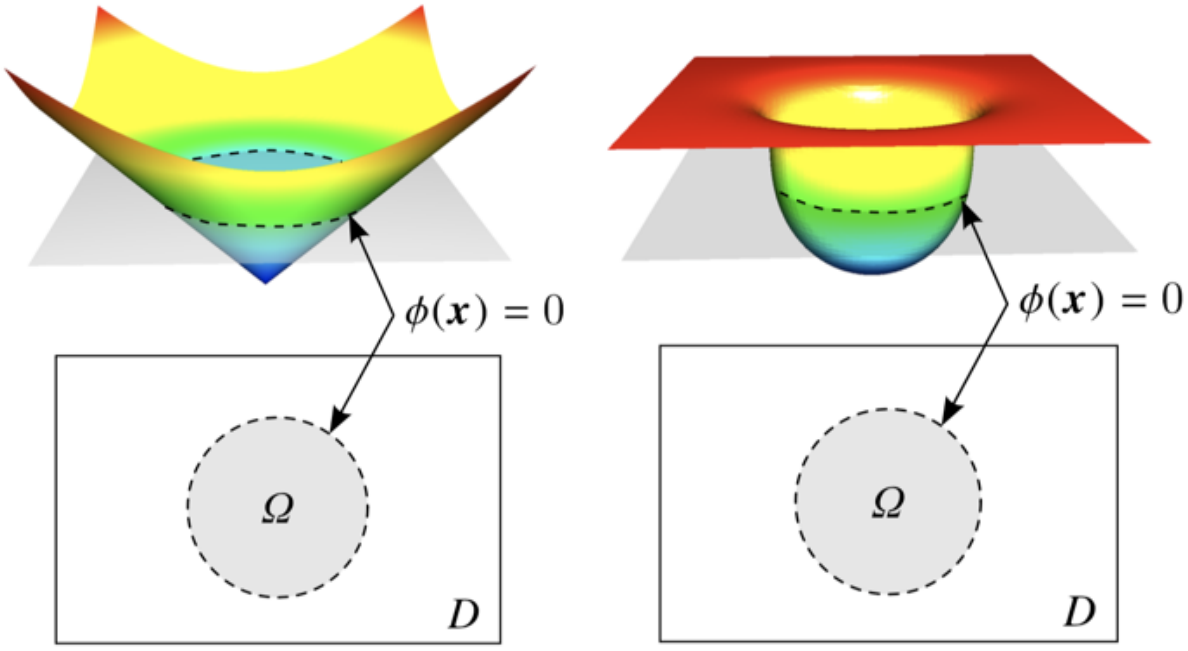
Sivapuram and Picelli (2018) proposed an improved method for gradient-based topology optimization in a discrete setting of design variables–Topology Optimization of Binary Structure (TOBS) method. The method combines the features of BESO and the discrete topology optimization method improve the effectiveness of binary variable optimization.

The objective and constraint functions given in Eq. (1.12) are sequentially linearized using Taylor's first order approximation as

$$\begin{aligned} \min_{\Delta\mathbf{x}_{it}} \quad & \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}_{it}} \Delta\mathbf{x}_{it} \\ \text{s.t.} \quad & \left.\frac{\partial g_i}{\partial \mathbf{x}}\right|_{\mathbf{x}_{it}} \Delta\mathbf{x}_{it} \leq \bar{g}_i - g_{i_{it}} \quad i \in [1,m] \\ & \Delta x_j \in \left\{-x_j, 1-x_j\right\} \quad j \in [1,n]\,. \end{aligned} \tag{1.15}$$

After each iteration, the design variables are updated as

$$\mathbf{x}_{it+1} = \mathbf{x}_{it} + \Delta\mathbf{x}_{it} \tag{1.16}$$



(A) $\phi(x)$ based on the signed distance function. (B) $\phi(x)$ based on the piecewise constant function.

FIGURE 1.3: Schematic figure of fixed design domain D and the level set function $\phi(x)$ based on the (a) signed distance function and (b) piecewise constant function. (Yaji, 2016).

The binary linear optimization problem defined in Eq. (1.15) can be solved using the integer linear programming (ILP). And ILP problem can be considered as a linear programming with an additional restriction that the feasible space of solutions is the integer space. Branch and bound algorithm is a famous technique used to solve an ILP (Williams, 2009).

1.2.3 Level-set method (LSM)

The level-set method (LSM) was proposed by Wang, Wang, and Guo (2003) and Allaire, Jouve, and Toader (2004), independently. The key idea is to use an iso-surface of a scalar function to explicitly express the boundary $\partial\Omega$ between different material phases, e.g., solid/void phases, fluid/solid phases, etc. This implicit function is the level-set function defined as

$$\begin{cases} \phi(x) < 0 & \text{for } x \in \Omega \\ \phi(x) = 0 & \text{for } x \in \partial\Omega \\ \phi(x) > 0 & \text{for } x \in D \backslash \bar{\Omega}. \end{cases} \tag{1.17}$$

Then the material property tensors can be expanded using the ersatz material approach as

$$\mathbb{A}_{H_\phi} = H_\phi \mathbb{A}, \tag{1.18}$$

where H_ϕ is a unit step Heaviside function.

To evolve the level-set function from one iteration to another, a commonly used way is to solve a time evolution equation. Here, we briefly discuss two types of level-set evolution equation: (1) Hamilton-Jacobi equation (HJE) and (2) reaction–diffusion equation (RDE).

Note that the mathematical programming can be also used to update the level-set function, see Luo, Wang, Wang, and Wei (2008) for a parameterized LSM and Wang and Kang (2018) for a velocity field LSM.

Hamilton-Jacobi equation (HJE)

In the original works by Wang, Wang, and Guo (2003) and Allaire, Jouve, and Toader (2004), the level set function ϕ is defined as a function of the signed distance function as

$$\begin{cases} \phi(\boldsymbol{x}) = -\operatorname{dist}(\boldsymbol{x}, \partial\Omega) & \text{for } \boldsymbol{x} \in \Omega \\ \phi(\boldsymbol{x}) = 0 & \text{for } \boldsymbol{x} \in \partial\Omega \\ \phi(\boldsymbol{x}) = \operatorname{dist}(\boldsymbol{x}, \partial\Omega) & \text{for } \boldsymbol{x} \in D \backslash \bar{\Omega}. \end{cases} \tag{1.19}$$

where $\operatorname{dist}(\cdot, \partial\Omega)$ is the Euclideean distance function to the boundary $\partial\Omega$. See the schematic of the level-set function shown in Fig. 1.3a.

A fictitious time $t \in [0, \infty)$ is introduced, the level-set is then varied using the following advection equation:

$$\frac{\partial \phi}{\partial t} + \boldsymbol{V} \cdot \nabla \phi = 0, \tag{1.20}$$

where $\boldsymbol{V}$ is the convection velocity which is defined as $\boldsymbol{V} = -v\boldsymbol{n}$, where v is a scalar function defined as the shape derivative. The unit normal vector is defined as $\boldsymbol{n} = \dfrac{\nabla\phi}{|\nabla\phi|}$. Therefore, the advection equation can be rewritten as a Hamilton-Jacobi equation as

$$\frac{\partial \phi}{\partial t} - v\,|\nabla \phi| = 0. \tag{1.21}$$

For the sake of the numerical stability, the level-set function should be reinitialized periodically. That is because the property of the signed distance function, i.e., $|\nabla\phi| = 1$ cannot be retained during the optimization process. However, without such properties, we cannot ensure that the smoothness of the level-set function. With this in mind, the reinitialization scheme can be an unsteady Eikonal equation:

$$\begin{cases} \dfrac{\partial \phi}{\partial t} + \operatorname{sign}(\phi_0)\,(|\nabla\phi| - 1) = 0 & \text{in } D \times \mathbb{R}^+ \\ \phi(t = 0, \boldsymbol{x}) = \phi_0(\boldsymbol{x}) & \text{in } D, \end{cases} \tag{1.22}$$

Reaction–diffusion equation (RDE)

In this thesis, we follow the lead of Yamada, Izui, Nishiwaki, and Takezawa (2010) by updating the level-set function using a reaction–diffusion equation. Unlike the HJE-based LSM, the level-set function here has a piecewise constant profile (see Fig. 1.3b) as

$$\begin{cases} -1 \leq \phi(\boldsymbol{x}) < 0 & \text{for } \boldsymbol{x} \in \Omega \\ \phi(\boldsymbol{x}) = 0 & \text{for } \boldsymbol{x} \in \partial\Omega \\ 0 < \phi(\boldsymbol{x}) \leq 1 & \text{for } \boldsymbol{x} \in D \backslash \bar{\Omega}. \end{cases} \tag{1.23}$$

The RDE is defined as

$$\frac{\partial \phi}{\partial t} = -K \left(\mathcal{F}' - \tau \nabla^2 \phi \right), \tag{1.24}$$

where K is a proportionality, $\mathcal{F}'$ is the topological design sensitivity, τ is the regularization parameter. The use of $\mathcal{F}'$ allows us to nucleate new holes during optimization process. In addition, the regularization term ensures the regularity of the level-set function, thus it does not require the reinitialization. The details of the mathematical background and the features of the proposed method will be provided in Chapter 2.

1.3 Sensitivity analysis

As has been discussed above, the gradient-based optimization solvers require the sensitivity, that is, the gradient of an objective function with respect to design variables. Then, the design variables can be updated to an optimum point based on the optimization methods as illustrated in Section 1.2. In this section, we discuss three different types of sensitivity analysis and their mathematical essence. They are: (1) finite difference method (FDM), (2) direct method, and (3) adjoint method.

Hereinafter, let us consider an optimization problem defined as follows:

$$\begin{aligned} \min_{\mathbf{x} \in \mathcal{X}} \quad & F\left(\mathbf{x}, \mathbf{u}\left(\mathbf{x}\right)\right) \quad \mathbf{x} = \left[x_1, x_2, \ldots, x_n\right]^{\mathrm{T}} \\ \text{s.t.} \quad & \mathbf{K}\mathbf{u} = \mathbf{f}, \end{aligned} \tag{1.25}$$

where F is the objective functional, $\mathbf{x}$ is the design variable, $\mathbf{u}$ is the state variable, $\mathbf{K}$ is the global stiffness matrix, $\mathbf{u}$ is the displacement vector (in the context of lamé systems), and $\mathbf{f}$ is the force vector. The equilibrium $\mathbf{K}\mathbf{u} = \mathbf{f}$ represent the system of the state equations.

The state variable $\mathbf{u}\left(\mathbf{x}\right)$ implicitly depends on the design variable $\mathbf{x}$, and it can be obtained by solving the state equations which is generally second-order partial differential equation. Then the sensitivity analysis is to calculate $\dfrac{\mathrm{d}F}{\mathrm{d}x_i}$ which is the functional derivative of the objective functional with respect to the design variable.

1.3.1 Finite difference method (FDM)

Finite difference method (FDM), as its name indicate, approximates the derivative by finite difference. That is, let $\varepsilon > 0$ denote the degree of the perturbation, using the second-order Taylor expansion, we get

$$F(\mathbf{x}+\varepsilon) = F(\mathbf{x}) + dF(\mathbf{x})\varepsilon + \frac{1}{2}d^2F(\mathbf{x})\varepsilon^2 + O\left(\varepsilon^3\right) \tag{1.26a}$$

$$F(\mathbf{x}-\varepsilon) = F(\mathbf{x}) - dF(\mathbf{x})\varepsilon + \frac{1}{2}d^2F(\mathbf{x})\varepsilon^2 + O\left(\varepsilon^3\right), \tag{1.26b}$$

where $d^2F(\mathbf{x})$ represents the second derivative. Then, applying the forward Euler scheme to Eq. (1.26a), we get

$$dF(\mathbf{x}) \approx \frac{F(\mathbf{x}+\varepsilon) - F(\mathbf{x})}{\varepsilon}, \tag{1.27}$$

and applying the backward Euler scheme to Eq. (1.26b), we get

$$dF(\mathbf{x}) \approx \frac{F(\mathbf{x}) - F(\mathbf{x}-\varepsilon)}{\varepsilon}. \tag{1.28}$$

These schemes have the first-order accuracy. Next, we subtract Eq. (1.26a) from Eq. (1.26a), we obtain the centered scheme as follows:

$$\frac{\mathrm{d}F(\mathbf{x})}{\mathrm{d}x_i} = \frac{F\left(x_1,\ldots,x_i+\varepsilon,\ldots,x_n\right) - F\left(x_1,\ldots,x_i-\varepsilon,\ldots,x_n\right)}{2\varepsilon}. \tag{1.29}$$

The merit of the FDM is its simplicity. However, it has the following drawback: (1) the value of ε should be small enough to guarantee the numerical accuracy. If its value is too big, the truncation error becomes larger. (2) To compute the sensitivity for all the design variables $x_i, i = [1, n]$, it requires us to solve n times of state equations. Therefore, it leads to an expensive computational cost.

1.3.2 Direct difference method

The second method is the direct difference method. Applying the Chain rule, $\dfrac{\mathrm{d}F}{\mathrm{d}\mathbf{x}}$ can be expressed as

$$\frac{\mathrm{d}F}{\mathrm{d}\mathbf{x}} = \frac{\partial F}{\partial \mathbf{x}} + \frac{\partial F}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \tag{1.30}$$

where $\dfrac{\partial F}{\partial \mathbf{x}}$ and $\dfrac{\partial F}{\partial \mathbf{u}}$ can be obtained explicitly.

Then the unknown is $\dfrac{\partial \mathbf{u}}{\partial \mathbf{x}}$. Taking the partial derivative, we get

$$\frac{\partial(\mathbf{K}\mathbf{u})}{\partial x_i} = \frac{\partial \mathbf{f}}{\partial x_i} \Rightarrow \frac{\partial \mathbf{K}}{\partial x_i}\mathbf{u} + \mathbf{K}\frac{\partial \mathbf{u}}{\partial x_i} = \frac{\partial \mathbf{f}}{\partial x_i}, \tag{1.31}$$

and Eq. (1.31) can be rewritten as

$$\mathbf{K}\frac{\partial \mathbf{u}}{\partial x_i} = \frac{\partial \mathbf{f}}{\partial x_i} - \frac{\partial \mathbf{K}}{\partial x_i}\mathbf{u}. \tag{1.32}$$

By solving Eq. (1.32), $\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$ can be calculated. To compute the sensitivity for all the design variables $x_i, i = [1, n]$, it requires us to solve n times of Eq. (1.32). Therefore, it leads to an expensive computational cost.

1.3.3 Adjoint method

The third method is the adjoint method. It can be further classified into discrete and continuous adjoint method. As for the discrete adjoint method, the state equation is discretized at first followed by calculating the derivatives, while for the continuous adjoint method, the derivatives are derived manually at first followed by the discretization of the state and adjoint equations.

The key idea of the discrete adjoint method is described here. First, the state equations and adjoint variables (or Lagrange multipliers) $\boldsymbol{\lambda}$ are discretized. Next, the Lagrangian $\mathcal{L}$ is constructed as

$$\mathcal{L} = F + \boldsymbol{\lambda}^{\mathrm{T}}(\mathbf{K}\mathbf{u} - \mathbf{f}). \tag{1.33}$$

If the equilibrium $\mathbf{K}\mathbf{u} = \mathbf{f}$ is satisfied, it indicates that the constructed Lagrangian is equivalent to the objective function, that is, $\mathcal{L} = F$. Therefore, we get

$$\frac{\mathrm{d}F}{\mathrm{d}x_i} \equiv \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}x_i} = \frac{\partial F}{\partial x_i} + \frac{\partial F}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial x_i} + \boldsymbol{\lambda}^{\mathrm{T}}\left(\frac{\partial \mathbf{K}}{\partial x_i}\mathbf{u} + \mathbf{K}\frac{\partial \mathbf{u}}{\partial x_i} - \frac{\partial \mathbf{f}}{\partial x_i}\right), \tag{1.34}$$

and Eq. (1.34) can be rewritten as

$$\frac{\mathrm{d}F}{\mathrm{d}x_i} = \frac{\partial F}{\partial x_i} + \boldsymbol{\lambda}^{\mathrm{T}}\left(\frac{\partial \mathbf{K}}{\partial x_i}\mathbf{u} - \frac{\partial \mathbf{f}}{\partial x_i}\right) + \left(\frac{\partial F}{\partial \mathbf{u}} + \boldsymbol{\lambda}^{T}\mathbf{K}\right)\frac{\partial \mathbf{u}}{\partial x_i}. \tag{1.35}$$

We define the adjoint equation as follows:

$$\frac{\partial F}{\partial \mathbf{u}} + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{K} = 0, \tag{1.36}$$

and the sensitivity finally yields:

$$\frac{\mathrm{d}F}{\mathrm{d}x_i} = \frac{\partial F}{\partial x_i} + \boldsymbol{\lambda}^{\mathrm{T}}\left(\frac{\partial \mathbf{K}}{\partial x_i}\mathbf{u} - \frac{\partial \mathbf{f}}{\partial x_i}\right). \tag{1.37}$$

Unlike the above-mentioned finite difference method and direct difference method, the adjoint method requires much less computational cost. For example, as for a optimization problem subject to m constraints, it requires us to solve m times of adjoint equations Eq. (1.36). This means that the sensitivity analysis does not depend on the number of design variables, thus making it more

efficient in the context of topology optimization considering that fact that in most of TO problems, the number of design variables are much higher than that of the constraints. From Eq. (1.37), we note that if the global stiffness matrix $\mathbf{F}$ has been assembled when the state equation is solved, then $\frac{\partial \mathbf{K}}{\partial x_i}$ can be computed straightforwardly. However, this is not always the case. For example, Yu, Ruan, Gu, Ren, Li, Wang, and Shen (2020) adopted a Semi-Implicit Method for Pressure Linked Equation (SIMPLE) which is a segregated solver. In their solver, the global stiffness matrix is not assembled during the iterative computation. This brings difficulties to use discrete adjoint method. To overcome this problem, one can use the automatic differential (AD) toolbox.

As has been mentioned above, unlike the discrete adjoint method, the continuous adjoint method does not require any specific scheme of the solver (i.e., coupled or segregated solver). The key idea is to first multiplier the state equations by adjoint variables and construct the Lagrangian. Then by applying the integration by parts, one can derive the adjoint equations and adjoint boundary conditions in a continuous form. After that, the state and adjoint equations are discretized for the numerical implementation. In this thesis, we adopt the continuous adjoint method to derive the sensitivity for the optimum design problems. The adjoint equations and sensitivities will be given in detail in the later chapters.

1.4 Motivation and objective

This thesis focuses on the topology optimization of thermal fluid–structure systems. As has been mentioned above, various shape and topology optimization methods have been proposed and notable achievements have been made during past decades. However, existing TO methods and frameworks have several limitations that impede the industrial application of these techniques. We discuss the challenging issues here.

The first is that of the *large-scale three-dimensional* topology optimization, summarized as follows:

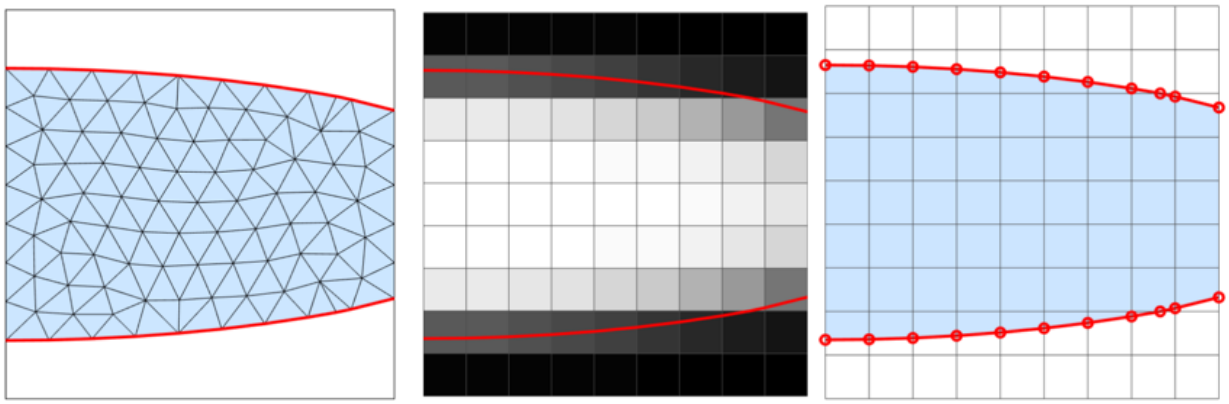
- In fixed mesh-based full-scale 3D TO algorithms, high resolution designs can be achieved using very fine meshes, which in turn, requires very high computational cost, i.e., CPU cores and memory usage.
- Adaptive mesh refinement is a powerful tool to achieve a high quality of boundaries. However, to date, only a few research works have extended the mesh adaption technique from 2D to 3D cases.
- In the traditional level set-based TO methods, some holes need to be distributed in the initial guess. This cause difficulties when the design domain features complex geometry.

The second is about the *fluid flow modeling* in fluid TO, summarized as follows:

- Geometric representation, cf. Fig. 1.4. As for the density-based approach, the generative design needs to be extracted from the obtained density field based on a threshold value. However, it is not easy to find a general guideline from the literature to decide the particular value of this threshold. In most of the cases, it is suggested to conduct the performance validation for different threshold values by trial-and-error. The LSM is capable of describing the geometric interface accurately. However, if this accurate boundary description cannot be converted to the model for the post-processing, simulation, and manufacturing, the ability of this accurate description makes less sense than it should.

- Flow modeling and computational cost. As has been studied in the previous contributions, some conclusions can be drawn that the full Navier–Stokes equations are more accurate for capturing the physics phenomenon of the fluid flow motion, compared to that of simplified models, i.e., Newton's law of cooling or Darcy's law for porous medium. However, solving the full NS equations usually implies a high computational cost. Most of the-state-of-the-art works were performed on multi-node clusters. Therefore, it still remains a challenge to construct a TO framework which is more affordable to most of the users, but at the same time, accurate enough to capture the behavior of the fluid flow motion.

- Parameter robustness. For a density-based approach, a continuation scheme is often needed for the material interpolation parameters, so as to penalize the density field to the binary structure progressively Alexandersen and Andreasen, 2020. To obtain a robust and efficient continuation scheme for a particular problem being solved, extensive numerical experiments are required. Therefore, there still remains some improvement space to reduce the overall algorithmic efforts.



(A) Explicit boundary expression.

(B) Density/ersatz material approach.

(C) Level set-based XFEM/cutFEM representation.

FIGURE 1.4: Fluid nozzle illustrating the basic differences among design representations in topology optimization: (a) explicit boundary representation (body fitted mesh); (b) density/ersatz material based representation; (c) level set based X-FEM/cutFEM representation (Alexandersen and Andreasen, 2020).

The third is about the *geometric constraint* in topology optimization. More specifically, to mimic the porous structures observed in natural systems, one can impose a maximum allowable volume fraction to the locally averaged value of the characteristic function. The main challenging issues are summarized as follows:

- Most of the existing works adopted density approach. There are huge number of gray elements in the optimal solution. Therefore, it can cause the misinterpretation of the optimal structure.
- Existing approach requires the measurement of the percentage of solid voxels overall voxels in a prescribed neighborhood. Therefore, it can not be easily used in the unstructured mesh and/or adaptive mesh-based TO framework. In other words, there are some specific restrictions on the numerical techniques.

This overall objective of this thesis is to construct a level set-based topology optimization for the optimum design of large-scale thermal fluid–structure systems.

1.5 Thesis structure

This thesis consists of 7 chapters. The main novelty and scientific contributions are summarized here.

In Chapter 2, the basic concept of the level-set method is introduced. The mathematical and physical frame of the reaction–diffusion equation (RDE) based topology optimization (TO) method is provided and the derivation of the topological derivative is given in details. Building upon this cornerstone, we propose a parallel distributed and open-source framework for full-scale three-dimensional structural topology optimization.

In Chapter 3, we extend the proposed TO framework to solve the two- and three-dimensional weakly-coupled fluid–structure problems. From the numerical point of view, two key ingredients are highlighted: (i) the body-fitted adaptive mesh strategy allows the disjoint-reunion of a global mesh that contains several (fluid/solid) subdomains, whose interfaces can be described by an implicitly defined surface (zero level-set); (ii) physics tailored multigrid preconditioner techniques are utilized to solve the large-scale finite element systems. More specifically, a modified augmented Lagrangian (mAL) preconditioner is adopted to solve the linearized flow equations. From the engineering stand point, we propose a complete product development workflow including the pre-processing, TO, B-Rep conversion, and the numerical experiment.

In Chapter 4, we shed a light on the passive heat sinks cooled by natural convection which are known to be reliable, compact, and low-noise. They are widely used in telecommunication devices, LEDs, etc. This chapter builds upon the recent advancements in fluid topology optimization

to present a case study of two- and three-dimensional optimum design and thermal modeling for natural convection problems using the proposed design methodology. To this end, first, a high-fidelity thermal–fluid model is constructed where the full Navier–Stokes equations are strongly coupled with the energy equation through the Boussinesq approximation. We benchmark our simulation solver against experimental analysis and other numerical analysis methods such as the ISHP method and the FVM. Next, we carefully investigate the flow behavior under different Grashof numbers using a fully transient simulation solver. Then we run our algorithm to deliver interesting 2D and 3D generative designs.

In Chapter 5, we focus on an important technical issue encountered in the large-scale fluid relevant TO. That is to improve the computational efficiency and to save the overall computational cost. To this end, we introduce the distributed unstructured mesh adaptation into the fluid-related topology optimization which is a first step in that direction. We incorporate three different remeshing techniques (isotropic, anisotropic, or body-fitted adaptive mesh refinement) into the reaction–diffusion equation-based (RDE) fluid topology optimization method. It requires a fully-distributed framework (including scalable domain decomposition, matrix assembly, parallel interpolation, linear solver) that very few general purpose libraries offer. In addition, this is the first attempt to conduct a comparative study by showcasing two different flow modeling strategies with their advantages and disadvantages. More specifically, the "separate" modeling, relying on the surface capturing technique, i.e., body-fitted mesh, allows the disjoint-reunion of a global mesh that contains several (fluid/solid) subdomains. The no-slip boundary conditions can be applied on the moving fluid-solid interface. The "hybrid" modeling, on the other hand, relying on the monolithic formulation, can be incorporated with iso-/anisotropic meshes. For comparison and for accessing the constructed framework, a lift–drag optimization problem is formulated. Various 2D/3D, small-/large-scale test cases are presented to validate the computational efficiency of this framework. A large-scale 3D test case (using fixed mesh) with a total tetrahedra number of $3.12 \cdot 10^6$, having approximately $1.32 \cdot 10^7$ degrees of freedom for the linearized fluid system, is solved using a cluster mounted with 352 MPI processes.

In Chapter 6, we focus on the biomimetic design of porous structures observed in natural systems. The basic idea is to impose a maximum allowable volume fraction tot he locally averaged value of the characteristic function. To compute this local averaged value, a PDE-filter is introduced. The proposed method has the following good features: (1) it can be used in the unstructured mesh and/or adaptive mesh-based TO framework. There is no specific restriction on the numerical techniques. (2) The use of this PDE-filter makes the distributed computing quite straightforward. (3) This geometrical constraint can be applied to any physics governed by different constitutive laws. To formulate the optimization model, we use the $p-$norm function to approximate the maximum value of the local averaged value thus makes it differentiable. Various large-scale three-dimensional test cases are presented. We push the tetrahedral element number to the level of 20 million and perform the computation on cloud-based clusters.

In Chapters 2 to 6, for each chapter, we first provide an in-depth literature review. We then point out the main challenging issues in the existed research works and the main contributions and novelty of our work. Next, we formulate the optimum design problem and describe the mathematical and physical frames. After that, we illustrate the implementation details and provide necessary sample code for the ease of reproductivity. Then, we run our algorithm to solve various two- and three-dimensional benchmark problems. Finally, at the end of each chapter, we summarize the new findings, limitations of the proposed method, and future works.

Chapter 2

Reaction Diffusion Equation-based Topology Optimization Method[†]

CONTENTS

2.1 Introduction

As it is well known, topology optimization (TO) methods have a high degree of flexibility to find an optimal structure within a given design domain. Therefore, TO methods have been widely used in structural design, for example, lightweight designs in aerospace industry (Zhu, Zhang, and Xia, 2016), aesthetics in civil engineering (Huang and Xie, 2010), high heat flux dissipation in thermal management (Li, Ding, Meng, Jing, and Xiong, 2019; Li, Ding, Jing, Xiong, and Meng,

[†]The work in this chapter has been published in *Finite Elements in Analysis and Design* DOI: 10.1016/j.finel.2021.103561.

2019; Yu, Ruan, Gu, Ren, Li, Wang, and Shen, 2020; Yu, Ruan, Wang, Li, and Shen, 2019), compliant mechanism design (Zhu, Zhang, Zhang, Liang, Zang, Li, and Wang, 2020; Zhu, Zhang, and Wang, 2013), etc. In addition, the rapid development of advanced manufacturing technology allows us to build complex high-performance products based on the TO approach in an easier way. During the past several decades, a number of typical shape/topology optimization methods have been proposed, such as the homogenization based approach (Bendsøe and Kikuchi, 1988), the density-based approach (Zhou and Rozvany, 1991; Rozvany, Zhou, and Birker, 1992; Bendsøe and Sigmund, 1999), the level-set-based approach (Wang, Wang, and Guo, 2003; Allaire, Jouve, and Toader, 2004), the evolutionary structural optimization approach (ESO) (Xie and Steven, 1993; Yang, Xie, Steven, and Querin, 1999), and the moving morphable components (MMC) approach (Guo, Zhang, and Zhong, 2014).

Among these methods, the level-set-based method is able to track boundary movements accurately. The basic idea is to incorporate an implicit representation of boundaries, and the level-set function is updated by solving the level-set equation, commonly known as a Hamilton–Jacobi equation (Osher, Fedkiw, and Piechor, 2004; Osher and Sethian, 1988). The idea of applying the level-set method to structural optimization was proposed in the early work of Wang, Wang, and Guo (2003) and Allaire, Jouve, and Toader (2004). Soon after that, several level-set updating strategies had been proposed to overcome the shortcomings in the early level-set methods due to the inability of nucleating new holes during the iterations. Allaire, De Gournay, Jouve, and Toader (2005) adopted the bubble method (Eschenauer, Kobelev, and Schumacher, 1994) to the level-set-based shape optimization using topological derivatives. In this work, new holes can be generated during the optimization process. Wang, Lim, Khoo, and Wang (2007) presented an extended level-set method based on one of their previous works (Wang, Wang, and Guo, 2003). In this extended method (Wang, Lim, Khoo, and Wang, 2007), an extended velocity which has a non-zero value in the material domain is introduced and there is no need to reinitialize the level-set function to maintain the property of a signed distance function, hence, allowing the introduction of new holes in a material domain. Yamada, Izui, Nishiwaki, and Takezawa (2010) proposed a level-set-based TO method where the level-set is updated using the RDE. This method allows not only shape but also topological changes, and allows the geometrical complexity of the optimal configuration to be specified.

In addition, large-scale (or high resolution) shape/topology optimization has attracted an increasing scientific interest because it can discover novel designs that cannot be achieved by using coarse meshes (Aage, Andreassen, Lazarov, and Sigmund, 2017). For example, for a minimal mean compliance problem, the optimal structure is a plate- or shell-like structure with various thickness rather than a truss-like structure. In other words, a coarse grid will limit the solution space of the optimization problem (Sigmund, Aage, and Andreassen, 2016). Such effect due to the use of a coarse grid is named as artificial length-scale effect. To overcome such a limitation, a high resolution structural TO formulation should be considered. To this end, parallel computing should be used to handle large computations.

Early parallel computing research in TO was mainly found in the density-based method. Borrvall and Petersson (2001) proposed a CPU parallel strategy using the regularized intermediate density control method and the method of moving asymptotes (MMA) is used as the update scheme. Aage, Andreassen, and Lazarov (2015) proposed a fully distributed TO framework using the Portable and Extendable Toolkit for Scientific Computing (PETSc). In their consequential work, a full-scale aircraft wing with more than one billion 3D elements was optimized on a cluster with 8,000 processors (Aage, Andreassen, Lazarov, and Sigmund, 2017). In this work, MMA is adopted as the update scheme. Liu, Hu, Zhu, Matusik, and Sifakis (2018) presented a narrow-band TO framework on a sparsely populated grid. Their technique can accommodate computational domains with over one billion grid voxels on a single shared-memory multiprocessors platform, allowing to generate optimal structures with both feature-rich shapes and exceptional mechanical performance. Their aircraft wing design also resemble the one from Aage, Andreassen, Lazarov, and Sigmund (2017). Wu, Dick, and Westermann (2015) presented a scalable algorithm running on a graphics processing unit (GPU) which can efficiently handle models with several millions of elements. They presented a variety of optimized structures with different shapes such as interior structures within closed surfaces, exposed support structures, and surface models. They then printed out the design structures. Martínez-Frutos, Martínez-Castejón, and Herrero-Pérez (2017) also accelerated TO computation using GPU, and the maximum number of mesh elements is up to $2.1 \cdot 10^6$ elements.

Very recently, the utilization of parallel computation in level-set-based shape/topology optimization algorithms is in rapid development. Liu, Tian, Zong, Ma, Wang, and Zhang (2019) suggested a fully parallel parameterized level-set method with compactly supported radial basis functions (CSRBFs) based on both uniform and non-uniform structured meshes. In this work, a numerical example with at most 7 million elements was conducted. They also reached the same conclusion as Sigmund, Aage, and Andreassen (2016) with regard to the artificial length-scale effect. That is, for a compliance minimization problem, the optimal structure is a plate- or shell-like structure with various thickness rather than a truss-like structure. Kambampati, Jauregui, Museth, and Kim (2020) presented a large-scale level-set TO method, in which the spatially adaptive and temporally dynamic volumetric dynamic B+ (VDB) tree data structure is adopted to address the shortcomings arising from the challenges in both slow convergence and high memory consumption. In this work, a numerical example with at most 34 million elements was computed. Several research efforts have been made to achieve large-scale 3D TO in different physical problems such as laminar incompressible flow (Villanueva and Maute, 2017; Yaji, Yamada, Yoshino, Matsumoto, Izui, and Nishiwaki, 2014), unsteady thermal-fluid problem (Yaji, Ogino, Chen, and Fujita, 2018), etc.

On the other hand, the recent development of mesh adaption techniques offers shape/topology optimization a possibility to achieve a better quality of boundaries by imposing mesh refinement around the solid/void material interface while coarser mesh elements remain in the rest of the domains. Furthermore, mesh adaption can be used to reduce the computational cost during the

iterative process, especially when the maximum allowed volume fraction is relatively small, because after several iterations, a huge percentage of the computational domain becomes void. As a result, although the void domains contribute little to the overall functional performance, it still costs a huge percentage of the overall computational resources which can be regarded as a waste. Therefore, mesh adaption enjoys the advantage not only of high resolution of boundary, but also of cheaper computational cost.

The idea of adaptive grid design combining numerical grid-generation methods and adaptive finite element methods (FEM) was proposed in the early work of Kikuchi, Chung, Torigaki, and Taylor (1986). Since this pioneering work, several successful implementations of adaptive mesh-based shape/topology optimization have been reported. Persson used mesh adaption (or mesh evolution) in between shape optimization iterations and showcased this in his thesis (Persson, 2004). Micheletti, Perotto, and Soli (2019) proposed a density-based TO driven by anisotropic mesh adaption based on a recovery-based posteriori error estimator, which allows elimination of the checkerboard phenomenon often seen in density-based TO method. Yaji et al. (Yaji, Otomori, Yamada, Izui, Nishiwaki, and Pironneau, 2016) constructed a shape optimization method based on the convected level-set method, in which the mesh adaption is applied to the level-set to concentrate fine meshes in an area near the level-set interface. Kim, Mingook, Takayuki, Shinji, and Jeonghoon (2020) presented a simple FreeFEM code to represent high-resolution boundaries of the optimal shape using the RDE-based TO method and adaptive mesh refinement, in which a fixed mesh is used to get a rough topology at first, and an adapted mesh is used to reach the final design with refined boundaries.

The benefits of mesh adaption become more evident in full-scale 3D cases. Nana, Cuillière, and Francois (2016) implemented a fully-automated adaptation strategy for density-based TO method on 3D unstructured tetrahedral meshes, in which the density gradient is used to locate the solid–void interface and h-adaptation is applied for a better definition of this interface. Baiges, Martínez-Frutos, Herrero-Pérez, Otero, and Ferrer (2019) proposed a fully parallelized level-set-based TO framework using a sparse grid stochastic collocation method, to calculate the statistical metrics of the topology optimization under uncertainty formulation, and a parallel adaptive mesh refinement method available in RefficientLib, to efficiently solve each of the stochastic collocation nodes. From the results, we can find that the optimal solution obtained with a deterministic design sometimes cannot converge to a symmetric design due to the lack of symmetry of the initial mesh.

Unlike mesh adaption strategies presented in above-mentioned works, Allaire, Dapogny, and Frey (2013) independently proposed a mesh evolution algorithm based on the level-set method for shape/topology optimization, in which the level-set function is moving on an unstructured mesh, and the surface corresponding to the zero level-set is remeshed. Feppon et al. applied this mesh revolution strategy in several multi-physics optimization problems in his thesis (Feppon, 2019) such as Lamé system, thermal conduction problem, and thermal fluid–structure coupling

problem (Feppon, Allaire, Bordeu, Cortial, and Dapogny, 2019; Feppon, Allaire, Dapogny, and Jolivet, 2020b) for both 2D and 3D cases. In these works, a proposed initial guess needs to be discretized by means of a mesh which is deformed to a better shape. And the design results sometimes may vary with the initial guess.

As mentioned above, notable achievements have been made by applying those TO methodologies. However, challenging issues still exist: (1) in fixed mesh-based full-scale 3D TO algorithms, high resolution designs can be achieved using very fine meshes, which in turn, requires high computational cost, i.e., CPU cores and memory usage. (2) Adaptive mesh refinement is a powerful tool to achieve a high quality of boundaries. However, to date, only a few research works have extended the mesh adaption technique from 2D to 3D cases. (3) In the traditional level-set-based TO methods, the optimal solution usually depends on the initial guess. Additionally, mesh dependency issues are often observed in TO designs. To date, only a few research work reported effective strategies to mitigate initial guess/mesh dependency simultaneously. (4) Though the generative designs have been introduced in the real world industry for several years, pre/post-processing always requires a considerable manual effort which is very time consuming. More importantly, the process of threshold and smoothing inevitably introduces the loss of accuracy.

In this chapter, we address the above mentioned challenges that impede an efficient full-scale 3D TO design. The main contributions of this chapter can be summed up as follows:

1. the RDE-based level-set TO method is formulated to solve the compliance minimization problem. Topological derivative acts as the source term in the RDE. Together with mesh refinement around the zero-level-set, the present method can mitigate the dependency to initial guess/mesh resolution to some extent. This feature is in contrast to what is generally seen in most of the traditional level-set-based shape optimization methods;

2. by properly combining parallel computing and mesh adaption in the proposed TO method, one can efficiently achieve a high-resolution and clear boundary by solving a smaller problem than with a fixed-mesh framework. This feature is in contrast to what is generally required by most large-scale 3D TO frameworks;

3. the boundary representation (B-Reps) is sampled with the volumetric representation of body-fitted tetrahedral meshes based on the level-set. The 3D printing data is directly converted from obtained generative meshes. We are free of most of the post-processing steps, i.e., thresholding and smoothing.

The remainder of this chapter is organized as follows. Section 2.2 formulates the mathematical model of the level-set-based TO where the level-set function is updated using the RDE. Section 2.3 illustrates our parallel strategy, mesh adaption technique and optimization flow chart. In Section 2.4, several typical numerical examples are conducted and the corresponding runtime analyses are provided to validate the effectiveness of the proposed TO method and of the parallel

framework/mesh adaption. Section 2.5 presents several 3D-printed prototypes to further support these remarkable features. Lastly, the conclusion is documented in Section 2.6.

2.2 Formulation

2.2.1 Governing equations

Considering a domain $\Omega \subset \mathbb{R}^d (d = 2 \text{ or } 3)$ under a surface traction, we intend to find the optimal topological configuration which can minimize the compliance (or maximize the stiffness) under a volume constraint. We assume the surface traction $\bar{g}$ to be applied on a portion of boundary $\partial\Omega$. The governing equation underlying the displacement field u can be formulated by the following linear elasticity equation:

$$\begin{cases} -\operatorname{div}\sigma(u) = 0 & \text{in } \Omega \\ u = 0 & \text{on } \Gamma_D \\ \sigma(u)\cdot n = \bar{g} & \text{on } \Gamma_N, \end{cases} \tag{2.1}$$

where $\sigma(u) = 2\mu e(u) + \lambda \mathrm{Tr}(e(u))\mathrm{Id}$ is the stress tensor, in which $e(u) = \frac{1}{2}\left(\nabla u + (\nabla u)^T\right)$ is the strain tensor, $\mathrm{Tr}(\cdot)$ is the trace operator, Id is the identity tensor, n is the outward normal vector to $\partial\Omega$, Γ_D is the Dirichlet boundary where the structure is fixed, and Γ_N is the Neumann boundary where the structure is loaded. The Lamé coefficients μ and λ is defined as

$$\mu = \frac{E}{2(1+\nu)}, \ \lambda = \frac{E\nu}{(1-2\nu)(1+\nu)}, \tag{2.2}$$

where E and ν denote the Young's modulus and the Poisson ratio, respectively. Now, we introduce the fourth-order elasticity tensor C as:

$$C_{ijkl} := \lambda\delta_{ij}\delta_{kl} + \mu\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right), \tag{2.3}$$

where δ_{ij} is the Kronecker delta function. The governing equations in Eq. (2.1) can be rewritten as:

$$\begin{cases} -\operatorname{div}(C : e(u)) = 0 & \text{in } \Omega \\ u = 0 & \text{on } \Gamma_{\mathrm{D}} \\ (C : e(u))\cdot n = \bar{g} & \text{on } \Gamma_{\mathrm{N}}. \end{cases} \tag{2.4}$$

Using the finite element analysis (FEA), the governing equation Eq. (2.4) is expressed in weak form as follows:

$$-\int_\Omega (e(u) : C) : e(\tilde{u}) d\Omega + \int_{\Gamma_N} \bar{g}\cdot\tilde{u} d\Gamma = 0 \ \forall \tilde{u} \in \mathscr{U}, \tag{2.5}$$

where $\tilde{u}$ in Eq. (2.5) denotes the test function for the state variable u.

We define the classical Hilbert space for the displacement u as

$$\mathscr{U} := \left\{ \tilde{u} \in H^1\left(\Omega, \mathbb{R}^d\right) \mid \tilde{u} = 0 \text{ on } \partial\Omega_s^D \right\}. \tag{2.6}$$

The elasticity tensor defined in Eq. (2.3): $C_{ijkl} := \lambda\delta_{ij}\delta_{kl} + \mu\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right)$, can be written into the vector form as follows:

$$C_{ijkl} = \begin{bmatrix} C_{1111} & C_{1122} & C_{1133} & C_{1123} & C_{1113} & C_{1112} \\ & C_{2222} & C_{2233} & C_{2223} & C_{2213} & C_{2212} \\ & & C_{3333} & C_{3323} & C_{3313} & C_{3312} \\ & & & C_{2323} & C_{2313} & C_{2312} \\ & \text{symm} & & & C_{1313} & C_{1312} \\ & & & & & C_{1212} \end{bmatrix} = \begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix}. \tag{2.7}$$

Therefore, the tensor defined in Eq. (2.5), $(e(u) : C) : e(\tilde{u})$, can be expanded as:

$$(e_{ij}(u) : C_{ijkl}) : e_{kl}(\tilde{u}) = \begin{bmatrix} \frac{\partial u_1}{\partial x_1} \\ \frac{\partial u_2}{\partial x_2} \\ \frac{\partial u_3}{\partial x_3} \\ \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} \\ \frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2} \\ \frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \begin{bmatrix} \frac{\partial \tilde{u}_1}{\partial x_1} \\ \frac{\partial \tilde{u}_2}{\partial x_2} \\ \frac{\partial \tilde{u}_3}{\partial x_3} \\ \frac{\partial \tilde{u}_1}{\partial x_2} + \frac{\partial \tilde{u}_2}{\partial x_1} \\ \frac{\partial \tilde{u}_2}{\partial x_3} + \frac{\partial \tilde{u}_3}{\partial x_2} \\ \frac{\partial \tilde{u}_1}{\partial x_3} + \frac{\partial \tilde{u}_3}{\partial x_1} \end{bmatrix}. \tag{2.8}$$

2.2.2 Level-set boundary expression

We briefly summarize the basic concept of the level-set-based TO method. A TO problem for the structural optimization problem is defined to include both void and solid domains. Let the design domain be denoted as $D \subset \mathbb{R}^d$. A solid domain is then defined as $\Omega \subseteq D$ and a void domain is represented as the complementary domain $D\backslash\Omega$, the scalar function, namely, the level-set function ϕ, as follows:

$$\begin{cases} 0 < \phi(\mathbf{x}) \leq 1 & \text{for } \mathbf{x} \in \Omega \\ \phi(\mathbf{x}) = 0 & \text{for } \mathbf{x} \in \partial\Omega \\ -1 \leq \phi(\mathbf{x}) < 0 & \text{for } \mathbf{x} \in D\backslash\Omega. \end{cases} \tag{2.9}$$

Note that the level-set function ϕ has a piecewise constant profile imposed in the above formulas, that is, $-1 \leq \phi(\mathbf{x}) \leq 1$.

2.2.3 Interpolation of material property

Here, we discuss how an elastic material field can be modeled using the level-set function ϕ in the design domain D. First, the characteristic function χ_ϕ is given by:

$$\chi_\phi := \begin{cases} 1 & \text{for } \phi(\boldsymbol{x}) \geq 0 \\ 0 & \text{for } \phi(\boldsymbol{x}) < 0. \end{cases} \tag{2.10}$$

Next, the Young's modulus E in the design domain is given by $E(\chi_\phi) = E\chi_\phi$. Thus, E and ν in Eq. (2.2) can be interpolated as:

$$\mu = \frac{1}{2(1+\nu)} E(\chi_\phi),\ \lambda = \frac{\nu}{(1-2\nu)(1+\nu)} E(\chi_\phi). \tag{2.11}$$

Hence, the elasticity tensor C can be expanded as:

$$C_{\chi_\phi} = \chi_\phi C. \tag{2.12}$$

As shown in Fig. 2.1, the non-negative value of the level-set function represents the solid domain $(\chi_\phi = 1)$ while the negative value represents the void domain $(\chi_\phi = 0)$.

2.2.4 Optimum design for the minimization of compliance

We now formulate the optimum design problem considering the minimization of compliance as the objective function. The volume constraint is applied to control the upper limit of the admissible volume fraction $V_{\max} \in (0, 1)$. Thus, the optimization mathematical model can be formulated as follows:

$$\inf_{\chi_\phi \in \mathcal{X}} J(\Omega) = \int_{\Gamma_N} \bar{g} \cdot u \, d\Gamma \tag{2.13}$$

$$\text{s.t.} \begin{cases} G_1 = \dfrac{\int_D \chi_\phi d\Omega}{\int_D d\Omega} - V_{\max} \leq 0 & \\ -\operatorname{div}(C : e(u)) = 0 & \text{in } \Omega \\ u = 0 & \text{on } \Gamma_D \\ (C : e(u)) \cdot n = \bar{g} & \text{on } \Gamma_N. \end{cases} \tag{2.14}$$

In the present work, the design variables are updated using the RDE which will be illustrated in the next section. Here, we introduce the topological derivative which is derived as the sensitivity information. As shown in Fig. 2.2, the topological derivative indicates the influence when a small

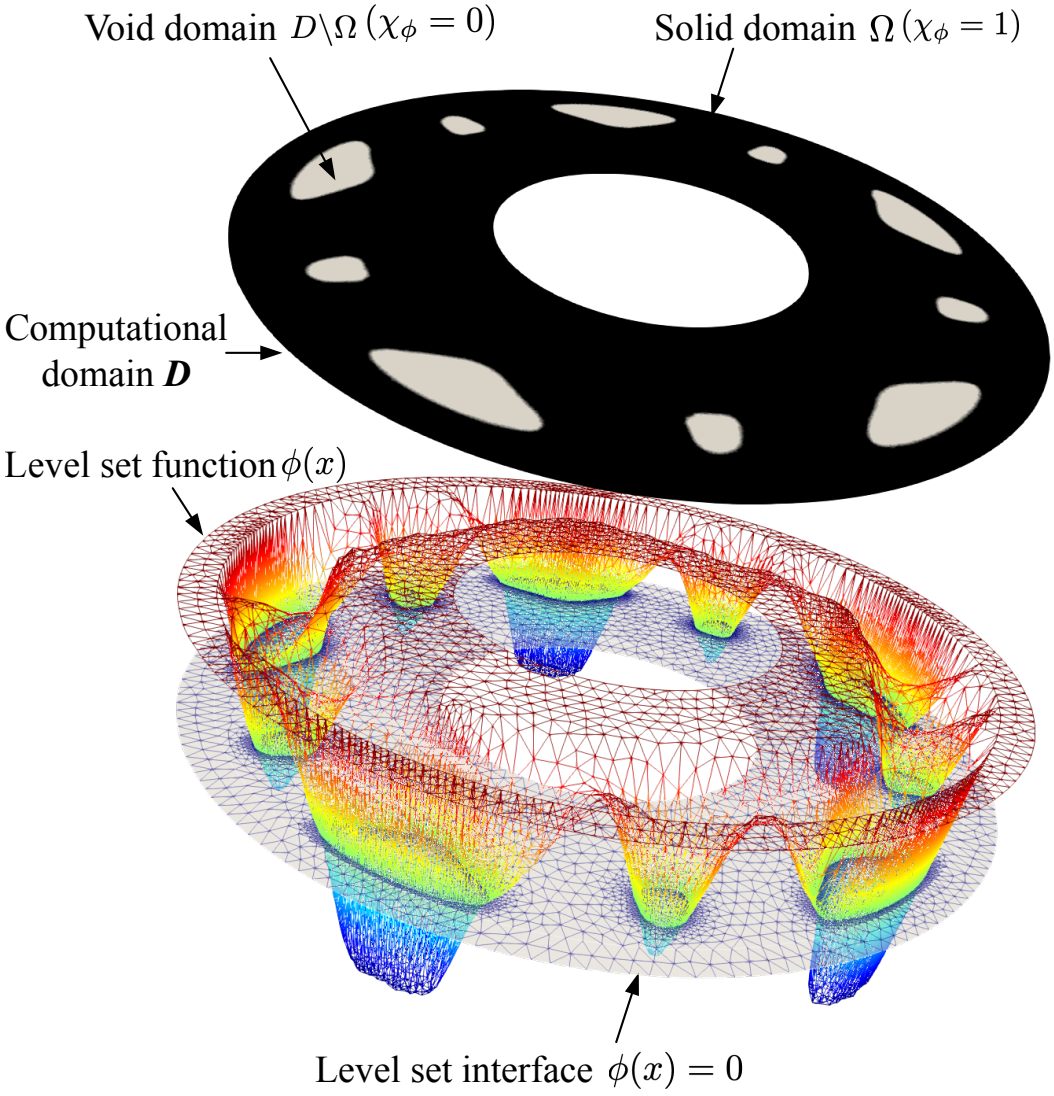


FIGURE 2.1: Schematic of the level-set function $\phi(x)$ and the characteristic function χ_ϕ.

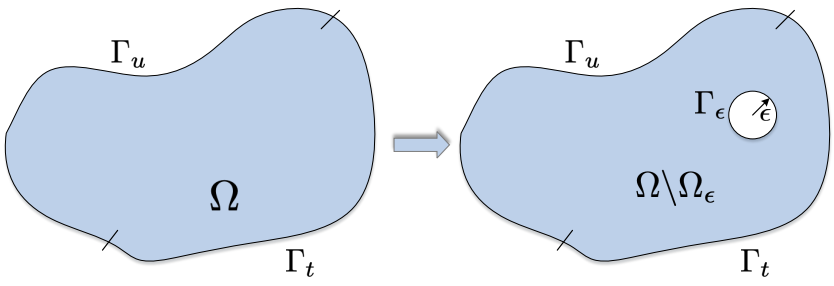


FIGURE 2.2: Schematic of topological derivative.

hole Ω_ϵ with a radius of $\epsilon \to 0$ is generated at a certain point in the domain Ω. For the compliance minimization problem, the topological derivative $d_t\bar{F}$ has been derived in Otomori, Yamada, Izui, and Nishiwaki (2015), as follows:

$$d_t\bar{F} = \lim_{\epsilon \to 0} \frac{\delta \bar{F}}{\frac{4\pi\epsilon^3}{3}} = -(e(v) : A) : e(u) - \lambda_1. \tag{2.15}$$

where $\delta\bar{F}$ is the variation of the Lagrangian. λ_1 is the Lagrange multiplier associated to the volume constraint. The adjoint variable v is defined to satisfy the adjoint equations as follows:

$$\begin{cases} \mathrm{div}(C : e(v)) = 0 & \text{in } \Omega \\ v = 0 & \text{on } \Gamma_\mathrm{D} \\ -(C : e(v)) \cdot n = \bar{g} & \text{on } \Gamma_\mathrm{N}. \end{cases} \tag{2.16}$$

Compared with the governing equation and boundary conditions of Eq. (2.4), it can be easily found that such problem is self-adjoint. Taking $v = -u$, we rewrite the topological derivative in Eq. (2.15), as follows:

$$d_t\bar{F} = (e(u) : A) : e(u) - \lambda_1, \tag{2.17}$$

where the fourth-order tensor A is defined as follows:

$$A_{ijkl} := A_1 \delta_{ij}\delta_{kl} + A_2 \left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right), \tag{2.18}$$

and it can be rewritten in the vector form as

$$\begin{aligned} &(e_{ij}(u) : A_{ijkl}) : e_{kl}(u) = \\ &\begin{bmatrix} \frac{\partial u_1}{\partial x_1} \\ \frac{\partial u_2}{\partial x_2} \\ \frac{\partial u_3}{\partial x_3} \\ \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} \\ \frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2} \\ \frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1} \end{bmatrix}^\mathrm{T} \begin{bmatrix} A_1 + 2A_2 & A_1 & A_1 & 0 & 0 & 0 \\ A_1 & A_1 + 2A_2 & A_1 & 0 & 0 & 0 \\ A_1 & A_1 & A_1 + 2A_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & A_2 \end{bmatrix} \begin{bmatrix} \frac{\partial u_1}{\partial x_1} \\ \frac{\partial u_2}{\partial x_2} \\ \frac{\partial u_3}{\partial x_3} \\ \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} \\ \frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2} \\ \frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1} \end{bmatrix}, \end{aligned} \tag{2.19}$$

where A_1 and A_2 are derived given as follows:

$$\begin{cases} A_1 = -\dfrac{3(1-\nu)\left(1-14\nu+15\nu^2\right)}{2(1+\nu)(7-5\nu)(1-2\nu)^2} E(\chi_\phi) \\ A_2 = \dfrac{15(1-\nu)}{2(1+\nu)(7-5\nu)} E(\chi_\phi). \end{cases} \tag{2.20}$$

The derivation of the adjoint equations and the topological derivative are summarized in Appendix A.1.

2.2.5 Level-set evolution equation

We introduce a fictitious time t and assume that the level-set function ϕ is an implicit function of t. To represent the presence or absence of material in the computational domain D over time using the sign of the level-set function, we assume that variations of the level-set function are proportional to the design sensitivity. Thus, the level-set function can be updated using the reaction equation as follows:

$$\begin{cases} \dfrac{\partial \phi}{\partial t} = -K\bar{F}' & \text{in } D \\ \nabla\phi \cdot \boldsymbol{n} = 0 & \text{on } \partial\Omega \backslash \partial\Omega_m \\ \phi = 1 & \text{on } \partial\Omega_m, \end{cases} \tag{2.21}$$

where $K > 0$ is the coefficient of proportionality, and $\bar{F}'$ is the design sensitivity. $\partial\Omega_m$ is the portion of the boundaries in between the design and non-design domains.

With the above assumption, it can be derived that with a descent step $\Delta t > 0$, the Lagrangian $\mathscr{L}$ monotonically decreases with the time evolution.

The level set function ϕ is the design variable, and a characteristic function χ_ϕ is monotonically increase with ϕ, as follows:

$$\begin{aligned} &\phi = \phi(x) \quad \text{design variable} \\ &\chi_\phi = \chi(\phi(x)), \quad \frac{\mathrm{d}\chi_\phi}{\mathrm{d}\phi} \geq 0 \quad \text{characteristic function.} \end{aligned} \tag{2.22}$$

Then, assume that there is an optimization problem in which the Lagrangian is defined as $\mathscr{L}(\chi_\phi)$:

$$\inf_{\chi_\phi \in \mathcal{X}} \mathscr{L} = \mathscr{L}(\chi_\phi). \tag{2.23}$$

The sensitivity with respect to the characteristic function χ_ϕ is given, namely, the design sensitivity $\bar{F}'$

$$\bar{F}' = \frac{\delta \mathscr{L}(\chi_\phi)}{\delta \chi_\phi}. \tag{2.24}$$

Then the functional derivative of the Lagrangian with respect to the design variable (level set function ϕ), can be expressed as follows:

$$\frac{\delta \mathscr{L}(\chi_\phi)}{\delta \phi} = \frac{\delta \mathscr{L}(\chi_\phi)}{\delta \chi_\phi} \frac{\mathrm{d}\chi_\phi}{\mathrm{d}\phi} = \bar{F}' \frac{\mathrm{d}\chi_\phi}{\mathrm{d}\phi}. \tag{2.25}$$

To solve the optimization problem expressed by Eq. (2.23), the design variable ϕ is evolved in time using the following reaction equation:

$$\frac{\partial \phi}{\partial t} = -K\bar{F}' \quad \text{in } D, \tag{2.26}$$

where $K > 0$ is the proportionality. The design sensitivity in domain $\bar{F}'$ acts as a reaction term of the evolution equation.

Then, choose a time step $\Delta t > 0$, the variation of ϕ with the time step can be expressed as follows:

$$\delta\phi(t) = \phi(t+\Delta t) - \phi(t) \approx \frac{\partial \phi(t)}{\partial t}\Delta t = -K\bar{F}'\Delta t \quad \text{in } D. \tag{2.27}$$

Therefore, the variation of χ_ϕ with the time step can be expanded as follows:

$$\delta\chi_\phi(t) = \frac{\mathrm{d}\chi_\phi(t)}{\mathrm{d}\phi(t)}\delta\phi(t) = -K\bar{F}'\frac{\mathrm{d}\chi_\phi(t)}{\mathrm{d}\phi(t)}\Delta t. \tag{2.28}$$

Then the variation of Lagrangian with the time step can be rewritten as:

$$\begin{aligned} &\delta\mathscr{L}(\chi_\phi(t)) \equiv \mathscr{L}(\chi_\phi(t+\Delta t)) - \mathscr{L}(\chi_\phi(t)) = \mathscr{L}\left(\chi_\phi(t) + \delta\chi_\phi(t)\right) - \mathscr{L}(\chi_\phi(t)) \\ &= \int_D \bar{F}'\delta\chi_\phi(t)dD. \end{aligned} \tag{2.29}$$

Finally, substituting Eq. (2.28) into Eq. (2.29), we can get:

$$\delta\mathscr{L}(\chi_\phi(t)) \approx -K\Delta t\int_D \left(\bar{F}'\right)^2 \frac{\mathrm{d}\chi_\phi(t)}{\mathrm{d}\phi(t)}dD \leq 0. \tag{2.30}$$

Therefore, the Lagrangian monotonically decreases with the time evolution.

It should be noticed that the sign of $\bar{F}'$ is opposite to the topological derivative $d_t\bar{F}$ defined in Eq. (2.15) due to the relationship between the definitions of topological derivative and the sign of the level-set function. Thus, the design sensitivity $\bar{F}'$ is given as:

$$\bar{F}' = -\left(e(u) : A\right) : e(u) + \lambda_1. \tag{2.31}$$

The optimization problem described by Eq. (2.21) is an ill-posed problem. Thus, it is regularized by a diffusive term $\nabla^2\phi$, so as to ensure the smoothness of the level-set function. Hence, the time-evolution scheme is rewritten as:

$$\begin{cases} \dfrac{\partial \phi}{\partial t} = -K\left(\bar{F}' - \tau\nabla^2\phi\right) & \text{in } D \\ \nabla\phi \cdot \boldsymbol{n} = 0 & \text{on } \partial\Omega\backslash\partial\Omega_m \\ \phi = 1 & \text{on } \partial\Omega_m, \end{cases} \tag{2.32}$$

The above scheme is named the reaction–diffusion equation (RDE) based time-evolution scheme

(Yamada, Izui, Nishiwaki, and Takezawa, 2010). It allows to simultaneously update the design variables and alleviate the overestimation of optimal shapes such as checkerboard patterns. Furthermore, the value of the regularization parameter τ controls the degree of this diffusivity, thus a larger value of τ can increase the diffusivity for the level-set function, which in turn, results in a more complex topological configuration.

Next, the sensitivity $\bar{F}'$ is normalized by adding a parameter $\tilde{C}$ so that the value of τ can be selected regardless of the particular problem being solved. We rewrite Eq. (2.32) to be dimensionless equations, as follows:

$$\begin{cases} \dfrac{\partial \phi}{\partial t} = -\left(\tilde{C}\bar{F}' - \tau\nabla^2\phi\right) & \text{in } D \\ \nabla\phi \cdot \boldsymbol{n} = 0 & \text{on } \partial\Omega\backslash\partial\Omega_m \\ \phi = 1 & \text{on } \partial\Omega_m, \end{cases} \tag{2.33}$$

The normalizer $\tilde{C}$ is defined as follows:

$$\tilde{C} = \frac{\int_D d\Omega}{\int_D |\bar{F}'|\, d\Omega}. \tag{2.34}$$

Using the finite difference method, Eq. (2.33) can be semi-discretized in time as follows:

$$\begin{cases} \dfrac{\phi(t+\Delta t)}{\Delta t} - \tau\nabla^2\phi(t+\Delta t) = -\tilde{C}\bar{F}' + \dfrac{\phi(t)}{\Delta t} & \text{in } D \\ \nabla\phi \cdot \boldsymbol{n} = 0 & \text{on } \partial\Omega\backslash\partial\Omega_m \\ \phi = 1 & \text{on } \partial\Omega_m, \end{cases} \tag{2.35}$$

where Δt is the step for fictitious time t. For the ease of numerical implementation using the FEA, the above equations are expressed in weak form as follows:

$$\int_\Omega \frac{1}{\Delta t}\phi_{n+1}\tilde{\phi} + \nabla^{\mathrm{T}}\phi_{n+1}\left(\tau\nabla\tilde{\phi}\right) + \left(\tilde{C}\bar{F}' - \frac{1}{\Delta t}\phi_n\right)\tilde{\phi}\, d\Omega = 0 \qquad \forall\tilde{\phi} \in \mathscr{P}, \phi \in \mathscr{P}. \tag{2.36}$$

where $\tilde{\phi}$ is the test function for ϕ. The design sensitivity $\bar{F}'$ will be given in the next subsection.

We set the classical Hilbert spaces for level-set function ϕ as

$$\mathscr{P} := \left\{\tilde{p} \in H^1(\Omega) \mid \tilde{p} = 0 \text{ on } \partial\Omega_p^D\right\}. \tag{2.37}$$

2.2.6 Augmented Lagrangian for inequality constrained problem

Now, we present the Lagrangian multiplier for the volume constraint λ_1. The augmented Lagrangian method is adopted (Rockafellar, 1973; Hestenes, 1969; Powell, 1969). The updating

scheme is given as:

$$\lambda_1^{i+1} = \lambda_1^i + c_1 G_1^i + c_2 \left(G_1^{i-1} - G_1^i \right), \tag{2.38}$$

where c_1 and c_2 are the constant coefficients. i is the current iteration number, and G is the constraint. In the present work, the volume constraint G_1 is given by:

$$G_1 = \frac{\int_D \chi_\phi d\Omega}{\int_D d\Omega} - V_{\max} \leq 0. \tag{2.39}$$

To stabilize convergence, the original volume constraint G_1 is relaxed and written as $\tilde{G}_1$:

$$\begin{aligned} \tilde{G}_1 = & \frac{\int_D \chi_\phi d\Omega}{\int_D d\Omega} \\ & - \underbrace{\left(V_{\max} + (V_0 - V_{\max}) \max \left(0, 1 - \frac{i}{n_{\text{vol}}} \right) \right)}_{\tilde{V}_{\max}} \leq 0. \end{aligned} \tag{2.40}$$

We note that the maximum admissible volume constraint $\tilde{V}_{\max}$ is gradually tightened until iteration $i = n_{\text{vol}}$. For $i = n_{\text{vol}} + 1, ..., n_{\max}$, the relaxed volume constraint $\tilde{G}_1$ yields the original G_1. Thus, the updating scheme can be rewritten as:

$$\lambda_1^{i+1} = \lambda_1^i + c \frac{\tilde{G}_1}{\tilde{V}_{\max}}, \tag{2.41}$$

with the initial λ_1^i for $i = 1$ given as:

$$\lambda_1 = \frac{\int_D d_t \bar{F} d\Omega}{\int_D d\Omega}. \tag{2.42}$$

When the volume constraint is not sufficiently satisfied in the current iteration, namely $\tilde{G}_1 \geq 0$, the λ_1 value will increase in the next iteration. In this way, the sensitivity of the volume constraint becomes larger compared to the objective functional. Vice versa, when the volume constraint is satisfied, namely $\tilde{G}_1 < 0$, the λ_1 will decrease. In that case, the sensitivity will be dominated by the objective functional.

2.3 Implementation details

2.3.1 Distributed computing

As already briefly explained, in this work, FreeFEM (Hecht, 2012; Jolivet, Dolean, Hecht, Nataf, Prud'homme, and Spillane, 2012) is used for the discretization of PDEs while PETSc (Abhyankar, Brown, Constantinescu, Ghosh, Smith, and Zhang, 2018; Balay, Gropp, McInnes, and Smith, 1997; Balay, Abhyankar, Adams, Benson, Brown, Brune, Buschelman, Constantinescu, Dalcin, Dener,

Eijkhout, Gropp, Hapla, Isaac, Jolivet, Karpeev, Kaushik, Knepley, Kong, Kruger, May, McInnes, Mills, Mitchell, Munson, Roman, Rupp, Sanan, Sarich, Smith, Zampini, Zhang, Zhang, and Zhang, 2021; Balay, Abhyankar, Adams, Brown, Brune, Buschelman, Dalcin, Dener, Eijkhout, Gropp, Karpeyev, Kaushik, Knepley, May, McInnes, Mills, Munson, Rupp, Sanan, Smith, Zampini, Zhang, and Zhang, 2019) is used for the linear algebra backend.

FreeFEM is a domain-specific language (DSL), internally written in C++. It makes extensive use of operator overloading and templates, something known as generic programming. Some other numerical libraries are integrated inside the DSL, such as PETSc and Mmg. They can be called from within FreeFEM. For parallelism, it relies on the distributed-memory paradigm, and more specifically, the message passing interface (MPI).

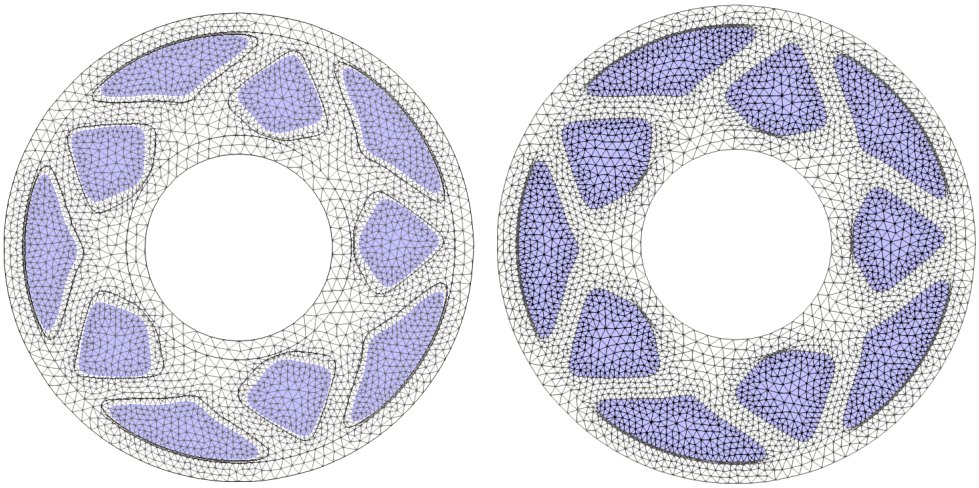
Parsing regular expression, i.e., x*(1+x) requires techniques that are at the core of compilation. Algorithms and data structures are well explained by Wirth (1985). The syntax of FreeFEM scripts obeys the rule of LL(1) computer languages. They can be parsed in one pass, generating a stack of instructions that are dispatched to low-level compiled code at execution.

More precisely, after decomposing an initial mesh using a graph partitioner package like METIS (Karypis and Kumar, 1998), the distributed assembly of the weak forms of Eqs. (2.5) and (2.36) is performed by FreeFEM. The resulting discreate linear systems are then passed over to PETSc. They are solved using multigrid preconditioners, GAMG (Adams, Bayraktar, Keaveny, and Papadopoulos, 2004) for Eq. (2.5) and *hypre* (Falgout and Yang, 2002) for Eq. (2.36). This ensures that this part of the TO method is scalable with respect to the number of processes and the mesh size of the models.

2.3.2 Body-fitted mesh adaptation technique

In the present work, we adopt the level-set-based mesh adaption method proposed by Allaire, Dapogny, and Frey (2013). The basic idea is to combine both numerical representations of the domain $\Omega \subset D$ seen as a subset of a computational domain D, as shown in Fig. 2.5. On the one hand, a computational mesh $\mathcal{T}$ of D is available, in which Ω is explicitly discretized as a meshed subdomain. On the other hand, Ω is implicitly described, using the level-set method. For example, it can be seen as the positive subdomain of a function $\phi : D \to \mathbb{R}$. The finite element resolutions are carried out using the meshed description, while the motion of the phase field is predicted using the level-set method. Therefore, the consistent alternation between both descriptions of the domain Ω can be achieved.

To implement this mesh adaption, we use the open-source library Mmg (Dapogny, Dobrzynski, and Frey, 2014). When written or read from disk, meshes are stored in the Inria Medit format, which is natively parsed by FreeFEM, Mmg, or Gmsh. The workflow of the *body-fitted mesh adaption* can be briefly summarized as follows: first, we calculate the topological derivative $d_t\bar{F}$ using Eq. (2.31) as the reaction source term and choose a time evolution step $\Delta t > 0$, on the current step



(A) $\mathcal{T}_i$ mesh. (B) $\mathcal{T}_{i+1}$ mesh.

FIGURE 2.3: Schematic of the body-fitted mesh evolution.

computational mesh $\mathcal{T}_i$, we solve the Reaction-diffusion equation Eq. (2.36). Secondly, we initialize the Mmg data structure, parse the mesh file and store the mesh $\mathcal{T}_i$. Thirdly, Mmg discretizes the newly updated Ω_{i+1} into $\mathcal{T}_i$ based on the obtained zero-level-set of ϕ_{i+1} (the black outline shown in Fig. 2.3a). Finally, Mmg iteratively improve the local mesh quality and end up with a body-fitted mesh as shown in Fig. 2.3b.

Here, we use a 2D cantilever beam example (shown in Fig. 2.4) to demonstrate the features of the proposed methodology. The left end is fixed on the wall ($\mathbf{u} = [0, 0]^T$) and a surface traction is applied on the bottom surface at the right-hand side ($\mathbf{g} = [0, -1]^T$).

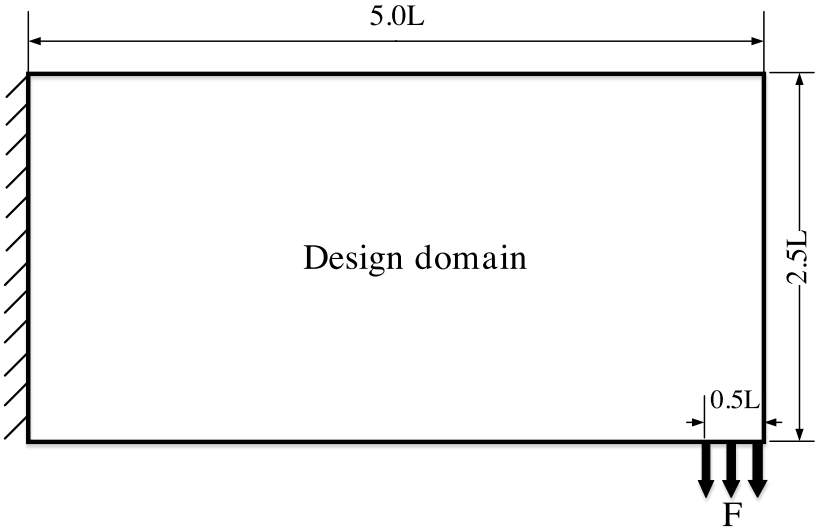


FIGURE 2.4: Design model of a 2D cantilever beam.

First, we use different Mmg adaptive mesh parameters and different values of regularization

parameter τ to study mesh dependency. The test cases are summarized in Table 2.1. $l_{e_{\text{global, max}}}$ and $l_{e_{\text{global, min}}}$ are the maximum and minimum edge sizes in the whole computational domain. h_{grad} allows the setting of the gradation value which controls the ratio between two adjacent edges. With a gradation of h_{grad}, two adjacent edges e_1 and e_2 must satisfy: $\frac{1}{h_{\text{grad}}} \leq \frac{l_{e_1}}{l_{e_2}} \leq h_{\text{grad}}$. $l_{e_{\text{local, max}}}$ and $l_{e_{\text{local, min}}}$ denote the maximum and minimum edge size alongside the zero-level-set interface. From the results shown in Fig. 2.5, we can observe the following features.

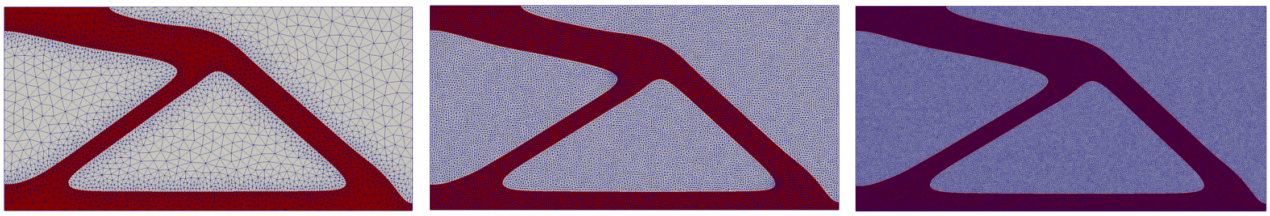
TABLE 2.1: The Mmg mesh adaption input parameters and regularization parameter τ for the 2D cantilever example shown in Fig. 2.5. n_{element} is the element number at the final iteration.

Case	$l_{e_{\text{global, max}}}$	$l_{e_{\text{global, min}}}$	h_{grad}	$l_{e_{\text{local, max}}}$	$l_{e_{\text{local, min}}}$	τ	n_{element}
#1	1.5×10^{-2}	1.0×10^{-3}	1.6	3.0×10^{-3}	3.0×10^{-3}	1.0×10^{-4}	6,686
#2	3.0×10^{-3}	1.0×10^{-3}	1.1	3.0×10^{-3}	3.0×10^{-3}	1.0×10^{-4}	30,573
#3	1.5×10^{-3}	1.0×10^{-3}	1.1	1.5×10^{-3}	1.5×10^{-3}	1.0×10^{-4}	121,105
#4	1.5×10^{-2}	1.0×10^{-3}	1.6	3.0×10^{-3}	3.0×10^{-3}	1.0×10^{-5}	8,257
#5	3.0×10^{-3}	1.0×10^{-3}	1.1	3.0×10^{-3}	3.0×10^{-3}	1.0×10^{-5}	31,452
#6	1.5×10^{-3}	1.0×10^{-3}	1.1	1.5×10^{-3}	1.5×10^{-3}	1.0×10^{-5}	121,894
#7	1.5×10^{-2}	1.0×10^{-3}	1.6	3.0×10^{-3}	3.0×10^{-3}	2.0×10^{-6}	17,604
#8	3.0×10^{-3}	1.0×10^{-3}	1.1	3.0×10^{-3}	3.0×10^{-3}	2.0×10^{-6}	35,955
#9	1.5×10^{-3}	1.0×10^{-3}	1.1	1.5×10^{-3}	1.5×10^{-3}	2.0×10^{-6}	122,620

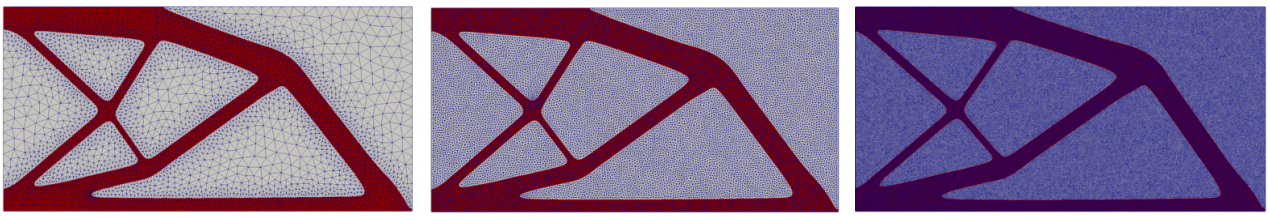
Feature 1. We use different combinations of the adaptive mesh parameters ($h_{\text{grad}}, l_{e_{\text{max}}}, l_{e_{\text{min}}}$, etc.). If the same value of the regularization parameter τ is used, we can observe very similar optimal solutions, as shown in Figs. 2.5a–i. For example, the "extremely fine mesh" case shown in Fig. 2.5c has 121,105 while the "coarse mesh" case shown in Fig. 2.5a has only 6,686 elements. This feature is contributed by the diffusive term in the Reaction-diffusion-equation together with the body-fitted mesh refinement at the solid-void boundary.

Feature 2. We use the same adaptive mesh parameters ($h_{\text{grad}}, l_{e_{\text{max}}}, l_{e_{\text{min}}}$, etc.) and use different values of regularization parameter τ. The optimal configuration becomes more complex with the decrease of the value of τ, as shown in Figs. 2.5a–i. Compare the element number for case #1 and case #7: $\tau = 2 \times 10^{-6} \rightarrow n_{element} = 17,604$; $\tau = 1 \times 10^{-4} \rightarrow n_{element} = 6,686$; This is because we impose the same local mesh size along the zero-level-sets, and a more complex geometry has a longer distance of zero-level-sets.

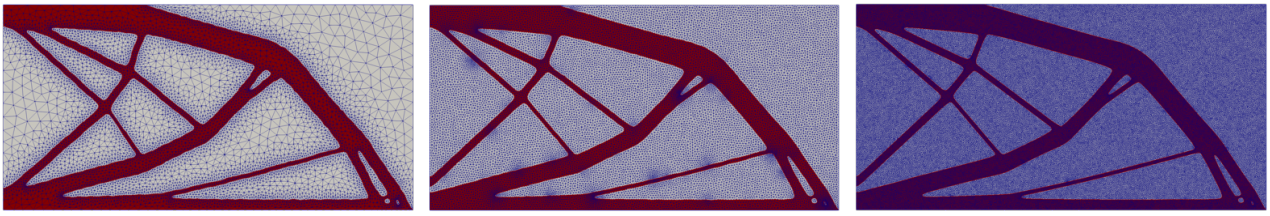
Feature 3. Next, we examine the dependency to initial guess. Another good feature can be observed. We use the same mesh parameters ($h_{\text{grad}}, l_{e_{\text{max}}}, l_{e_{\text{min}}}$, etc.) and the same value of the regularization parameter $\tau = 1 \times 10^{-4}$. Then initialize the level-set field with six different initial guesses, as shown in Fig. 2.6. The proposed method can mitigate the dependency to initial guess to some extent. This feature is contributed by the use of topological derivative. Thus, holes can be generated, combined, removed and deformed freely during iteration.



(A) 2D case #1. (B) 2D case #2. (C) 2D case #3.

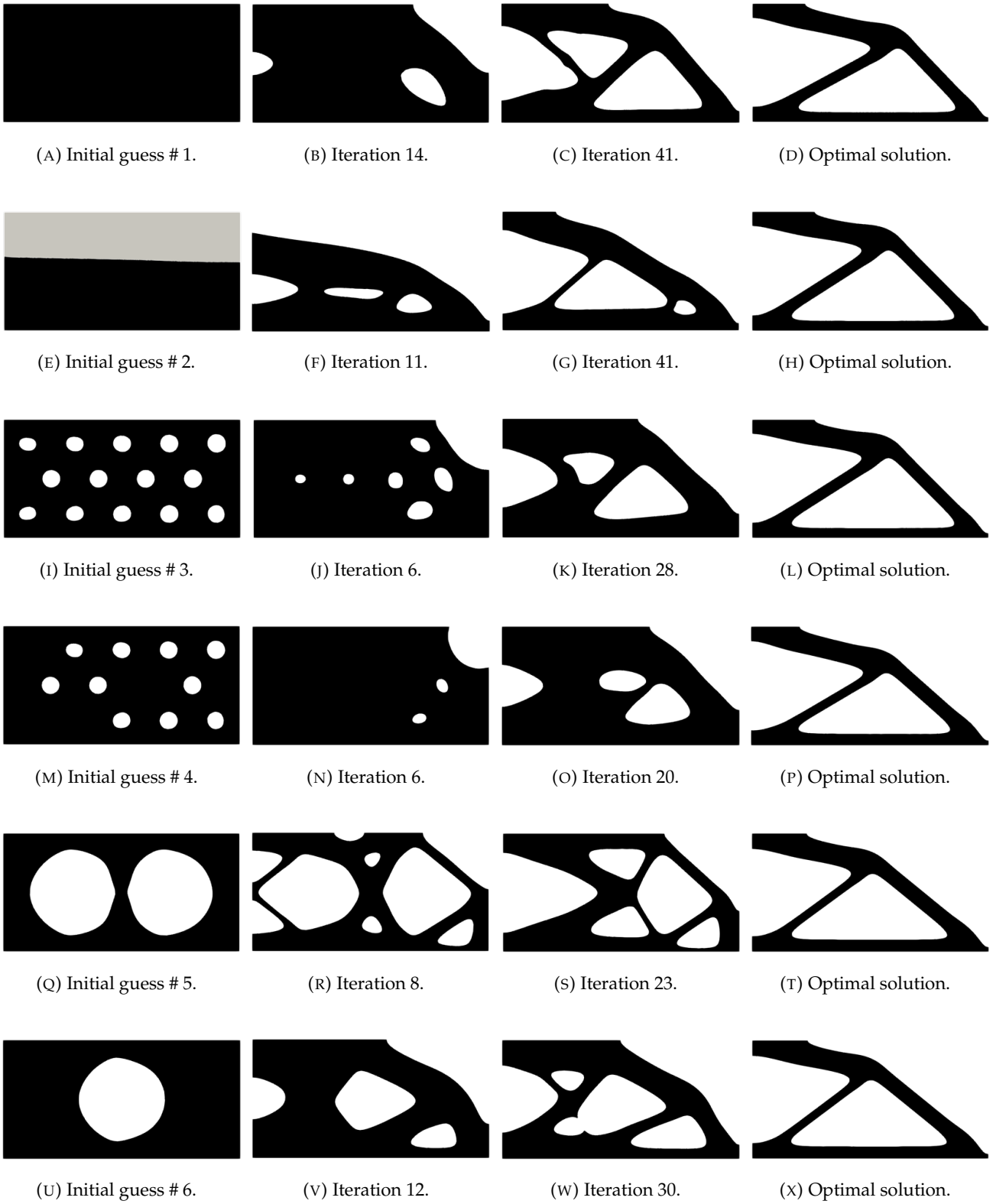


(D) 2D case #4. (E) 2D case #5. (F) 2D case #6.



(G) 2D case #7. (H) 2D case #8. (I) 2D case #9.

FIGURE 2.5: Design results for the case of 2D cantilever beam plotted in Fig. 2.4 with different mesh parameter settings and different values of τ documented in Table 2.1. Meshed subdomain $\Omega \in D$ (in red) of a computational domain D.



(A) Initial guess # 1. (B) Iteration 14. (C) Iteration 41. (D) Optimal solution.

(E) Initial guess # 2. (F) Iteration 11. (G) Iteration 41. (H) Optimal solution.

(I) Initial guess # 3. (J) Iteration 6. (K) Iteration 28. (L) Optimal solution.

(M) Initial guess # 4. (N) Iteration 6. (O) Iteration 20. (P) Optimal solution.

(Q) Initial guess # 5. (R) Iteration 8. (S) Iteration 23. (T) Optimal solution.

(U) Initial guess # 6. (V) Iteration 12. (W) Iteration 30. (X) Optimal solution.

FIGURE 2.6: Iterative history and the optimal solutions initialized with different guesses for the case of 2D cantilever.

From the above-mentioned three features, it can be concluded that: to manage the trade-off between high resolutions and low computational costs, Mmg input parameters should be selected carefully, especially in 3D cases. For example, meshes can be progressively refined from the mesh used in the previous iterations. This allows skipping a lot of unneeded computations, especially during the early initial iterations when the configuration is still far beyond the optimal solution. Such progressively adaptive mesh strategy is not only reasonable in the physical sense but also cost-efficient in the computational sense.

Furthermore, it should be pointed out that the low-dependency to initial guess/mesh resolution is observed only in the minimum compliance problem. There is no guarantee that it is also true in other physics problems.

2.3.3 Optimization flow chart

The flow chart of our optimization algorithm is shown in Fig. 2.7. In the optimization loop, all finite element steps and algebraic solves are performed in parallel using FreeFEM and PETSc, respectively. They both rely on MPI. Mesh adaptation is currently performed sequentially on a single process. It needs (resp. generates) a global input (resp. output) mesh and a global input level-set function. Thus, once the level-set function is updated by solving the RDE in parallel, on a decomposed global mesh, we generate a global level-set function by performing an MPI reduction on process 0. We then have the global input mesh from the current iteration, as well as the global level-set function, which are passed to Mmg to generate the new adapted mesh used in the next iteration.

The closed-loop optimization algorithm is described as follows: the computational domain is built and the initial global mesh is backed up. Secondly, the computational domain has to be decomposed using a standard mesh partitioner such as METIS. Thirdly, the initial level-set function (initial guess) is set. In the fourth step, the governing equations are solved using the FEM. Then, the objective functional is computed. If the objective functional is converged, the optimization ends, otherwise the sensitivities are computed. In the fifth step, the level-set function is updated by solving the RDE, after which ϕ is mapped to satisfy $\phi(x) \in [-1, 1]$ followed by updating the characteristic function χ_ϕ with the step function. Then, we generate a global level-set function by performing an MPI reduction on process 0. After that, the global mesh which is gathered on process #0 can be updated based on the level-set function using Mmg which is a sequential package. Eventually, the new global mesh is backed up, and it is decomposed. The workload is then distributed again, using MPI, until the end of the iteration where the previous workflow for generating a sequential global mesh and level-set function needed by Mmg is performed.

Considering the bottleneck caused by the sequential mesh adaption, in the present work, we limit the problem scale to no more than 3 million tetrahedra. In the near future, once the parallel

version of Mmg will be able to handle surface mesh adaptation, it will be possible to switch to a fully distributed framework.

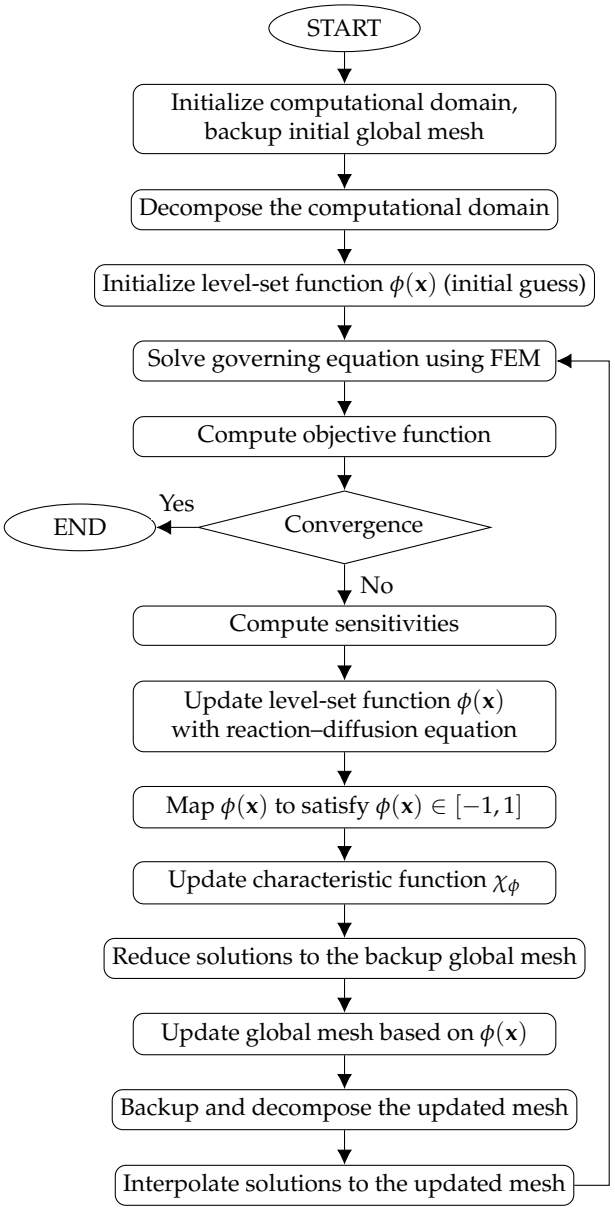


FIGURE 2.7: Optimization flow chart.

2.4 Numerical investigations

In this section, five design examples are studied to examine the effectiveness, stability and extensibility of the proposed TO algorithm. All numerical experiments are performed on the same workstation with at most 80 processes, and with a memory size of 256GB. The CPU is an Intel(R) Xeon(R) E5-2698 V4, using the GNU compiler, OpenMPI, and the same linear solver parameters.

As already explained in **Feature 2**, the the optimal solution has a more complex configuration with a smaller value of τ in Eq. (2.33). In this work, to avoid an overly complex optimal design, τ is set to $\geq 10^{-5}$. Also note that all quantities used in the optimization in this work are dimensionless. Thus, we use L to denote the unit length. The Young's Modulus E is 1 for the solid material ($\chi_\phi = 1$) and the Poisson's ratio ν is 0.3. In all the following examples, 4-node linear tetrahedron elements are used. The maximum admissible volume fraction is set to 30% except the final bridge case (15%). In this chapter, all the numerical examples are computed with full model, that is, no symmetric boundary condition is used.

2.4.1 Cantilever beam

In this subsection, we report a successful implementation of full-scale 3D TO performed on general workstations by combining parallel computing with body-fitted mesh adaption. The scalability analysis is conducted using the fixed-mesh test cases. Then the total runtime analysis reveal that mesh adaptivity allows us to save a huge amount of computational cost while keeping the high resolution of boundary.

The design model of a cantilever bean is shown in Fig. 2.8 with "length $\times$ height $\times$ width" = $5L \times 2.5L \times 2.5L$ and the surface traction $\mathbf{g} = [0, 0, -1]^T$ applied on the area of the bottom surface highlighted in purple, and the left end fixed on the wall ($\mathbf{u} = [0, 0, 0]^T$). The ϕ field is initiated with a uniform value of 1.

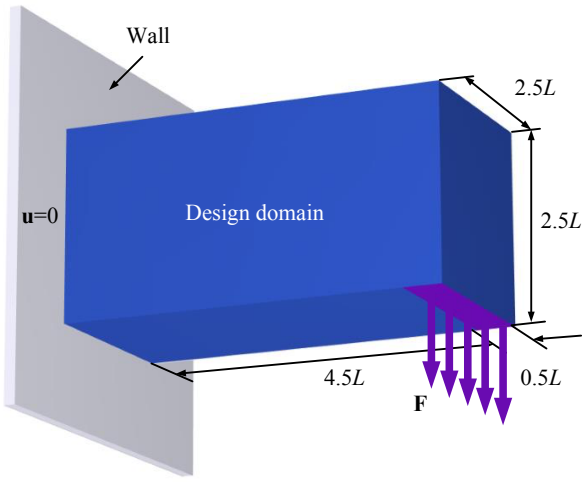


FIGURE 2.8: Design model for the cantilever beam.

In this subsection, runtime analysis is carried out to investigate the parallel efficiency and the mesh adaption behavior: (1) different numbers of processes and mesh elements are used to test the runtime consumption with fixed-mesh cases (3D cases #1 to #10). (2) The adaptive mesh case is computed for comparison (3D case #11). The test cases are summarized in Table 2.2 in which n_{element} and n_P denote the number of elements and processes, respectively.

TABLE 2.2: Test cases used to conduct runtime analysis for 3D cantilever beam case: number of mesh elements and number of processes.

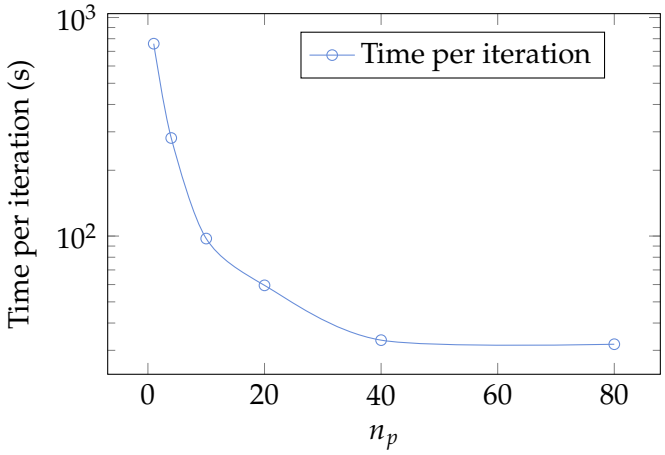
Case No.	n_{element}	n_P
3D case #1	864,000	1
3D case #2	864,000	4
3D case #3	864,000	10
3D case #4	864,000	20
3D case #5	864,000	40
3D case #6	864,000	80
3D case #7	108,000	80
3D case #8	364,500	80
3D case #9	864,000	80
3D case #10	2,916,000	80
3D case #11	13,500 $\rightarrow$ 874,835	20

First, the parallel efficiency is analyzed based on the fixed-mesh cases with $8.64 \cdot 10^5$ elements. Fig. 2.9a plots the runtime per iteration for the cases performed with different n_P (3D cases #1 to #6). It can be found that runtime gradually reduces with increasing number of processes. The CPU speed-up does not increase linearly above 40 processes and the strong scaling maximum efficiency is then reached.

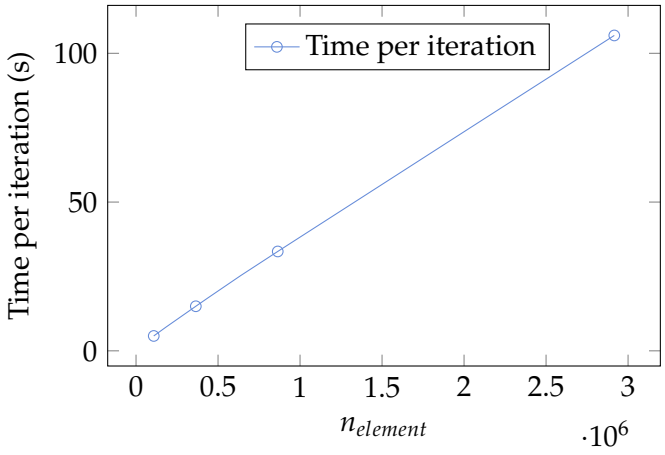
Secondly, the runtime is measured for the fixed-mesh cases with different element numbers ranging from as low as 108,000 to as high as $2.92 \cdot 10^6$. All computations for 3D cases #7 to #10 are performed using 80 processes. Fig. 2.9b shows the runtime per iteration for the cases with different n_{element}. It can be found that: (1) the runtime shows an almost linear increase with n_{element}. In the present work, GAMG is used for solving elasticity systems, and *hypre* for RDE. These multigrid preconditioners are state-of-the-art and scale almost linearly with problem size. We can expect the same memory scaling between GAMG and *hypre* since they are both algebraic multigrid solvers with low operator complexities for these discretized PDEs. (2) The largest case in the present work is case #10 which has $2.92 \cdot 10^6$ elements. There, the time per iteration is 106.3 s.

Now, we integrate the body-fitted adaptive mesh into the framework, as illustrated in Section 2.3.3. The Mmg input parameters as a function of the iteration number n_{it} are summarized in Table 2.3. The element size is progressively decreasing throughout the iterations. The element corresponding to the level-set interface is imposed to be finer than the others. In addition, to save computation cost, the h_{grad} is set to 1.8. The beam is initially discretized into 13,500 elements and ends up with 874,835 elements at iteration 150. The calculation is performed using 20 processes.

The design result is plotted in Fig. 2.10, and from left to right are the isometric, top , side, and back views, respectively. On the one hand, by comparing the design results from both the full-scale 3D design (3D case #11, see Fig. 2.10) and the 2D design (2D cases #1 to #3, see Fig. 2.5), From the back view it can be observed that a huge percentage of material ($\chi_\phi = 1$) is removed from the



(A) 3D case #1–#6.



(B) 3D case #7–#10.

FIGURE 2.9: Time per iteration of 3D cantilever example using different (a) number of processes and (b) number of elements.

TABLE 2.3: Mmg input parameters used for 3D case #11.

Mmg input parameters	Value
$l_{e_{\text{global, max}}}$	$1.5 \times 10^{-2} \times (1 - n_{\text{it}}/300)$
$l_{e_{\text{global, min}}}$	0.3×10^{-2}
h_{grad}	1.8
$l_{e_{\text{local, max}}}$	$1.0 \times 10^{-2} \times (1 - n_{\text{it}}/300)$
$l_{e_{\text{local, min}}}$	$0.8 \times 10^{-2} \times (1 - n_{\text{it}}/300)$

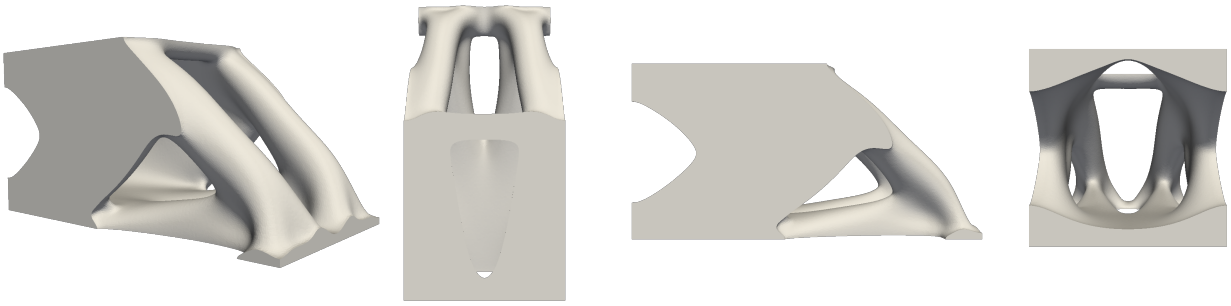


FIGURE 2.10: Design result for the 3D cantilever beam using adaptive meshes, from left to right: isometric, top, side, and back views.

central region of the design domain, which means that the material can contribute more efficiently to the overall functional performance when it is concentrated on the side walls.

Moreover, the iteration histories of the objective value and the volume fraction for 3D case #11 is depicted in Fig. 2.11, in which the optimization iterations converge smoothly in less than 150 steps. It should be mentioned that after iteration #100, the objective value calculated based on the FEA remains nearly unchanged with the increase of the mesh number.

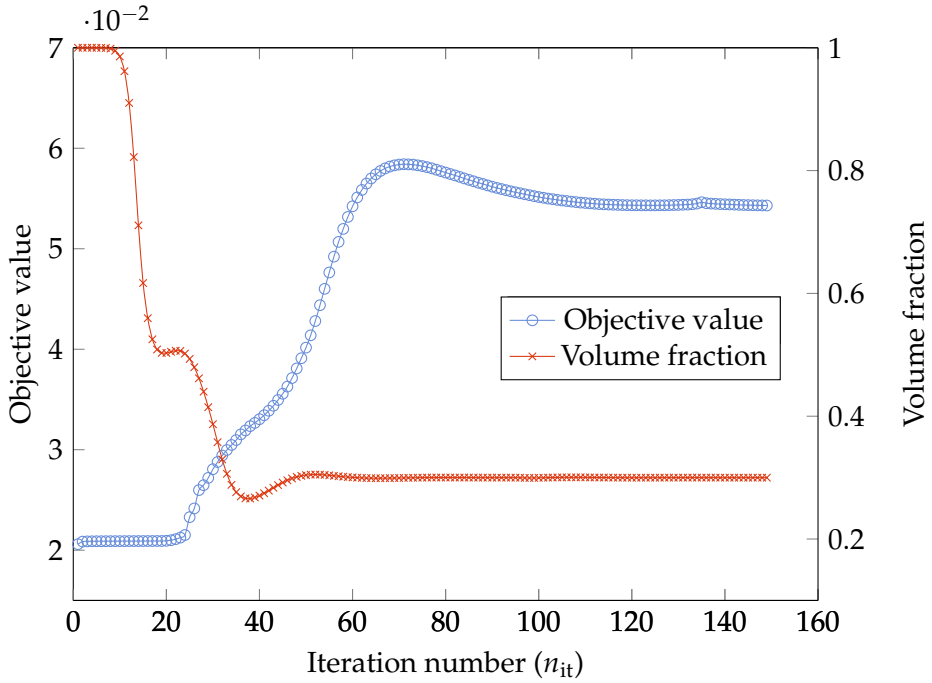
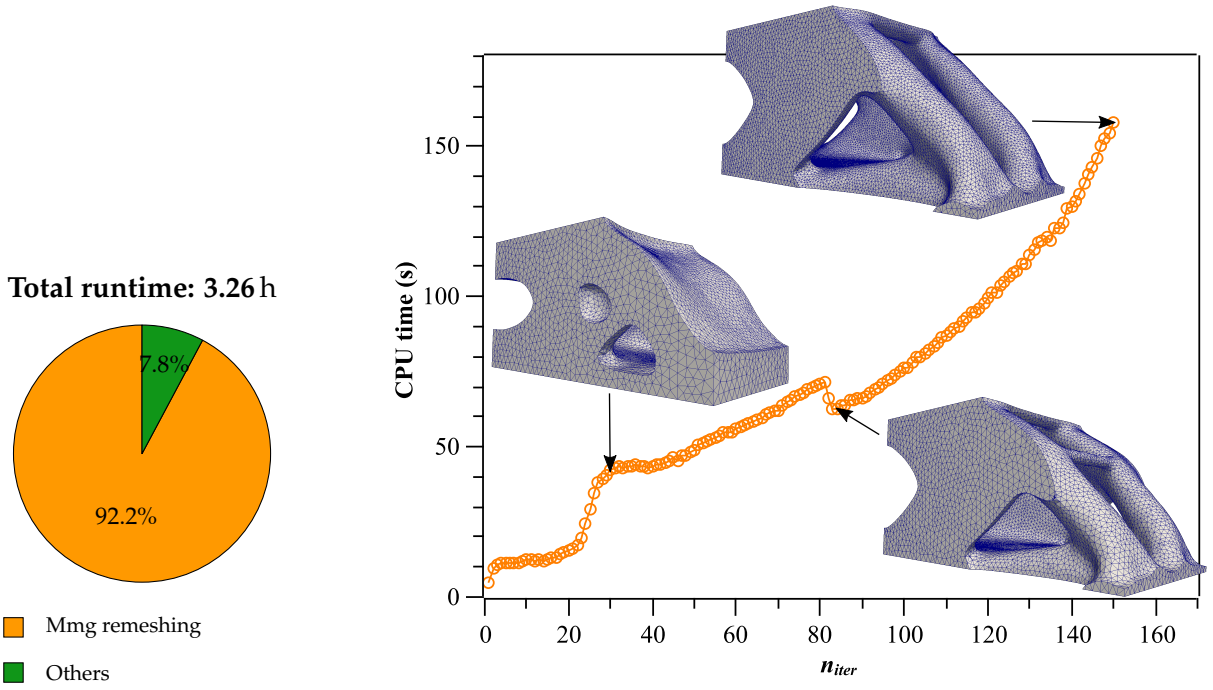


FIGURE 2.11: Histories of the objective value and the volume fraction for the 3D cantilever beam design case (3D case #11).

Next, the runtime analysis is carried out. The total runtime is 3.26 h which comprises two parts: one is mesh updating, the other one is the solving process (including FEA, decompose domain, etc.). The former takes 92.2% of the total runtime while the latter takes 7.8%, as presented in the pie chart in Fig. 2.12a. These two parts will be analyzed respectively, as follows.

Fig. 2.12b plots the Mmg updating time for each iteration, and three intermediate steps of the topological configurations are presented above the curve. It can be seen that the mesh along the level-set surface is gradually refined iteration by iteration. The topology forms roughly at around iteration #30. After iteration #80, the configuration does not change dramatically until convergence. A stable topology evolution and the convergence is obtained. Furthermore, the mesh updating process takes 92.9% of the total runtime because Mmg is running on a single MPI process. Therefore, with the increasing number of elements, the bottleneck caused by the sequential Mmg package becomes more and more evident.



(A) Total runtime

(B) Mmg mesh updating time

FIGURE 2.12: Runtime analysis for the 3D cantilever beam example using adaptive mesh (3D case #11): (a) total runtime consumed in solving and Mmg mesh updating, (b) Mmg mesh updating time for each iteration.

In Fig. 2.13, the total runtime for both fixed mesh-based (3D case #10) and adaptive mesh-based (3D case #11) test cases are broken down into each step: mesh updating, solving RDE, computing the Lagrangian multiplier and the normalizer, computing the objective value and the constraint value, exporting visualization data, and solving the governing equation. The comparative analysis results can be concluded as follows: (1) the adaptive mesh-based case is consistently faster than the fixed mesh-based one (except the mesh updating). This is because the numbers of elements in the adaptive mesh case are much less than those in the fixed-mesh case. (2) Besides the mesh adaption, the most time consuming step is to solve the RDE, followed by the governing equation, computing normalizer. (3) The total runtime for the adaptive mesh case (3.26 h) is 36.4% shorter than that for fixed mesh (5.08 h). More importantly, the number of processes used in the adaptive

mesh case (20 processes) is only one fourth of that used in the fixed-mesh case (80 processes).

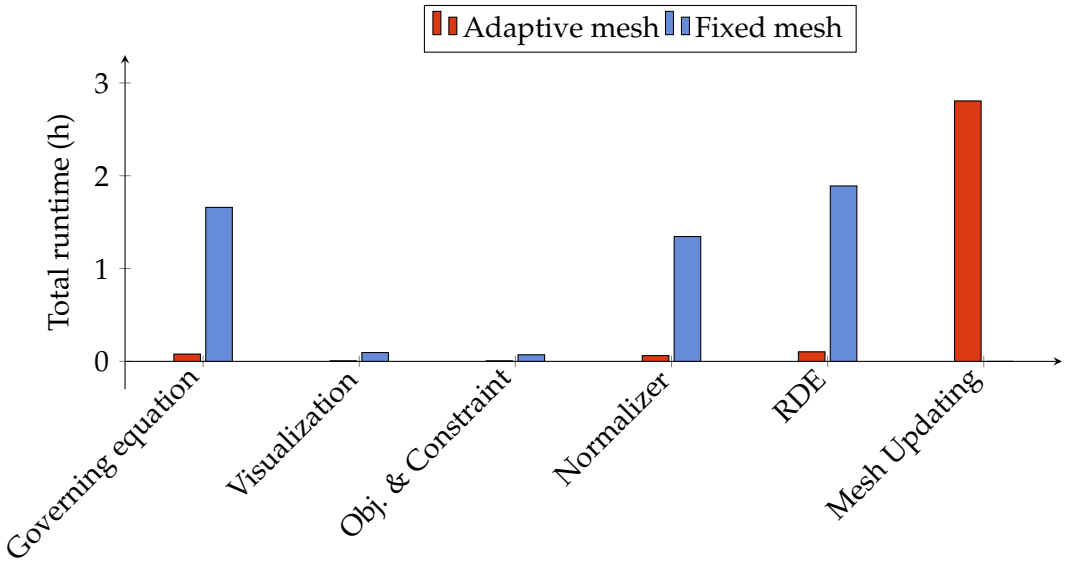


FIGURE 2.13: Time breakdown of the total runtime (150 iteration) for the 3D cantilever examples using adaptive meshes (3D case #11) and a fixed mesh (3D case #10).

2.4.2 Wheel-like structure

The second example is a three-dimensional wheel-like structure as shown in Fig. 2.14, where the inner and outer gray regions are the non-design domains and the middle blue region is the design domain. The innermost surface of the wheel is fixed. There are five uniform counter-clockwise torsion loads imposed tangent to the outer surface of the wheel-like structure on the five lines highlighted in purple.

The majority of light vehicles use 5 (or less frequently 6, and hardly seen 3 or 4) wheel studs for each wheel. One reason is the die cast often has an odd number of spokes because directly opposed spokes cause problems with residual stress distribution as the casting cools and shrinks. Therefore, 5 studs tends to be a more convenient number. There is also the issue of redundancy. If one of the five studs fails, the wheel still has a reasonable distribution of functional nuts which cover more than 180° of the wheel hub. Conversely, if only three of four are present, then most probably the wheel cannot sit flat on the hub. Considering the practical engineering preference, five torsion loads are employed in this example.

Fig. 2.15 shows the iterative history of the surface mesh at the level-set interface. It should be mentioned that the mesh elements at the interface between design and non-design domains remain unchanged during the iterations while only those inside the gray design domain are adapted based on the level-set. Thanks to Mmg built-in options, such mesh adaption features can be realized by the following steps:

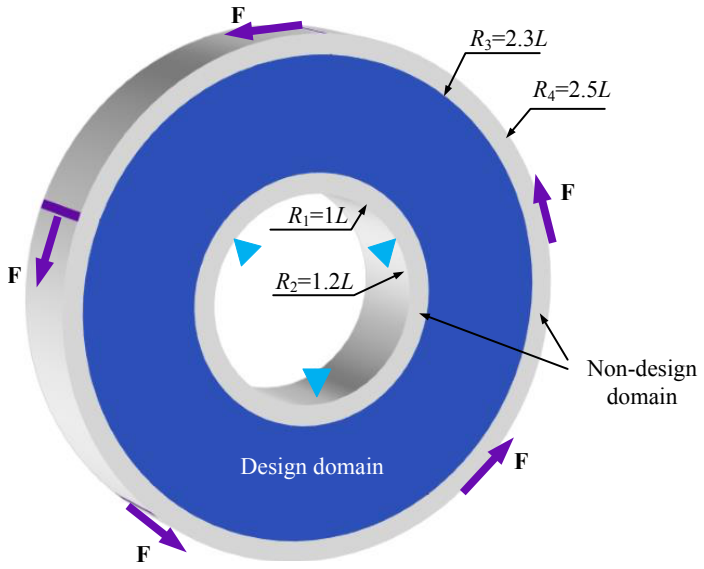


FIGURE 2.14: Design model for the wheel-like structure.

Step 1 parse the mesh file and pass it to Mmg;

Step 2 get the number of triangles of the mesh and store it in **nt**;

Step 3 get the reference of each triangle of the **nt** triangles. Mmg uses an internal counter, i.e., if one calls Mmg and gets a triangle function for the first time, the triangle number is 1, if one calls it for the fifth time, the triangle number is 5, etc.;

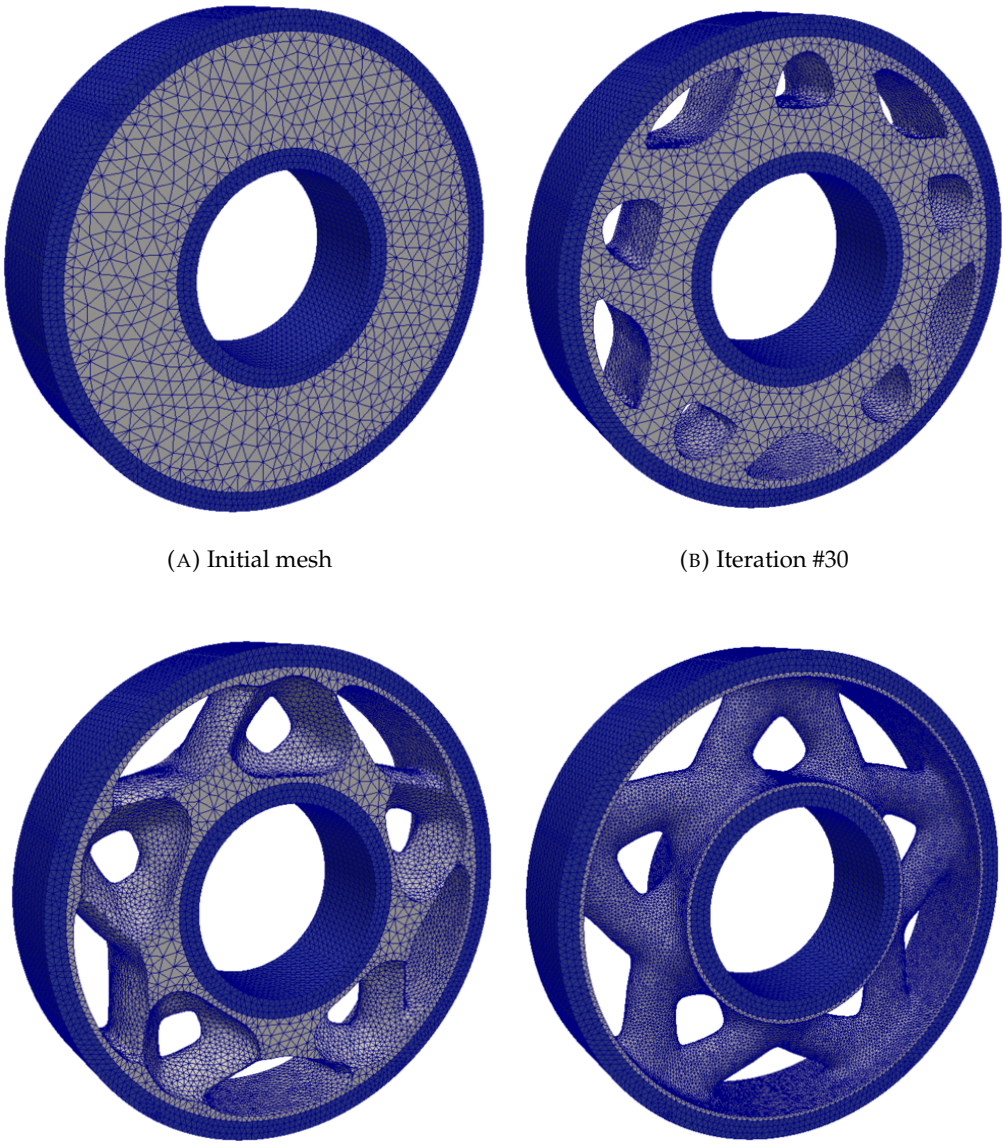
Step 4 test this reference and set the triangle as required if its reference is 1;

Step 5 save the mesh and free the Mmg data structures.

2.4.3 Bracket-like structure

The third example is a bracket-like structure as shown in Fig. 2.16, where the two gray rings are the non-design domains and the middle blue region is the design domain. The top-right ring is fixed and the surface traction is applied on the inner surface of the bottom-left ring. Figs. 2.17a and b show the full-scale 3D design result and Fig. 2.17c shows the 2D design result.

Now, we further verify the effectiveness of the 3D design. To this end, the mechanical performance is compared between the full-scale 3D design and the 2D extrusion design with the same volume fraction. The established models are exported from FreeFEM into a commercial FEA software COMSOL Multiphysics 5.4 COMSOL, 2018. In the following numerical simulations, we assume the bracket is made of aluminum, the Young's modulus is 70 GPa, the Poisson's ratio is 0.33, and the density is 2700 kg m^{-3}. The surface traction is $\mathbf{g} = \left[0, 0, -1.0 \times 10^5\right]^T$ N^2 m^{-2}. The



(A) Initial mesh (B) Iteration #30

(C) Iteration #40 (D) Iteration #150

FIGURE 2.15: Iterative history of the topological configuration for the wheel-like structure example.

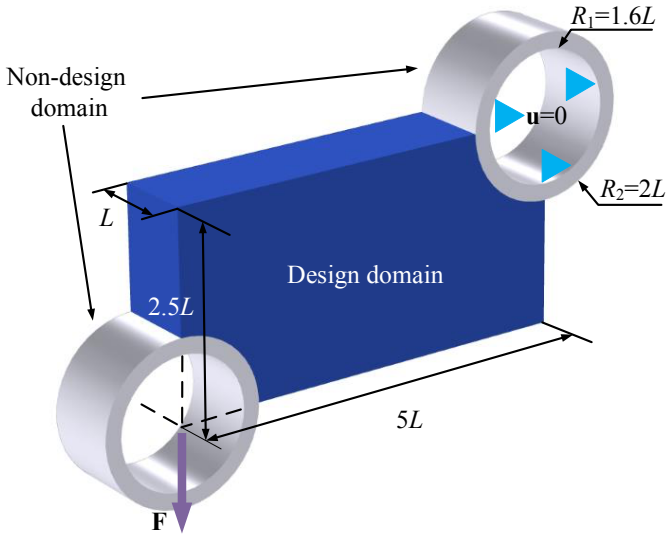


FIGURE 2.16: Design model for the bracket-like structure.

thickness of the bracket is 20 mm. Note that the volume difference between the 3D/2D-extrusion designs is less than 0.1%.

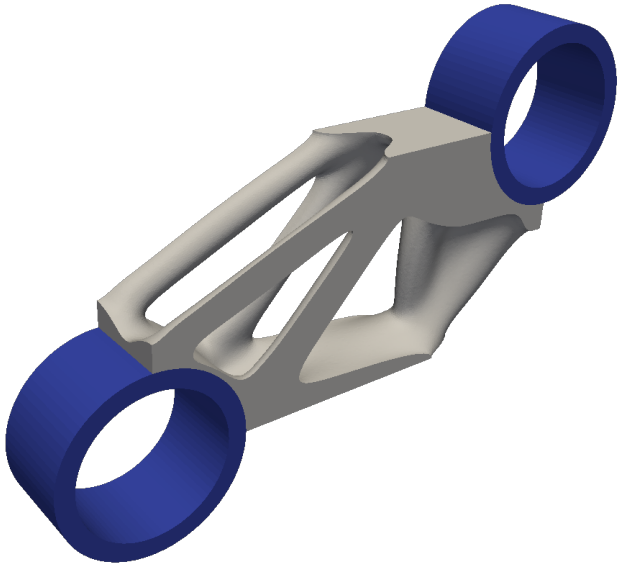
Before conducting any simulation, the mesh-independency analysis should be carried out to ensure the FEA accuracy. In this numerical case, the solutions (maximum displacement and maximum von Mises stress) become mesh independent when the element numbers reach $1.15 \cdot 10^5$ (full-scale 3D design) and 51,980 (2D extrusion design), respectively. The displacement distribution of both designs are plotted in Fig. 2.18. The simulation results show that the maximum displacement of the full-scale 3D design is 3.7% smaller than that of the 2D extrusion design. The comparison results indicate that the full-scale 3D model includes more abundant design space information so that it predicts the design sensitivity more accurately than its 2D-extrusion counterpart.

2.4.4 Engineering cases. Airplane bearing bracket and Bridge

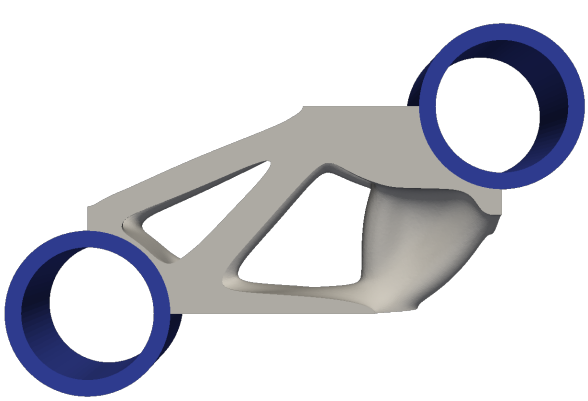
Now, we move towards two engineering cases: (1) aircraft components, (2) bridge.

Airplane bearing bracket. Flight cost per 100 kg varies from several hundreds of dollars to thousands of dollars depending on the flight state (steady flight, climbing or descending) and fuel cost takes about 35% of the total cost of a flight. In this section, we will design a weight-saving and fuel-economy bearing bracket structure of airplanes.

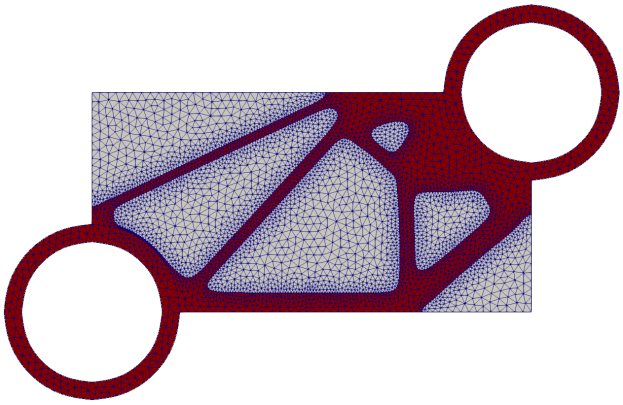
The design model is plotted in Fig. 2.19. The bottom of the bearing bracket is fixed by means of four high strength bolts. To simplify the design model, we set $\mathbf{u} = [0, 0, 0]^T$ on the inner walls of



(A) Isometric view



(B) Front view



(C) 2D design

FIGURE 2.17: Design result for the bracket-like structure. (a) and (b): full-scale 3D design, (c): 2D design.

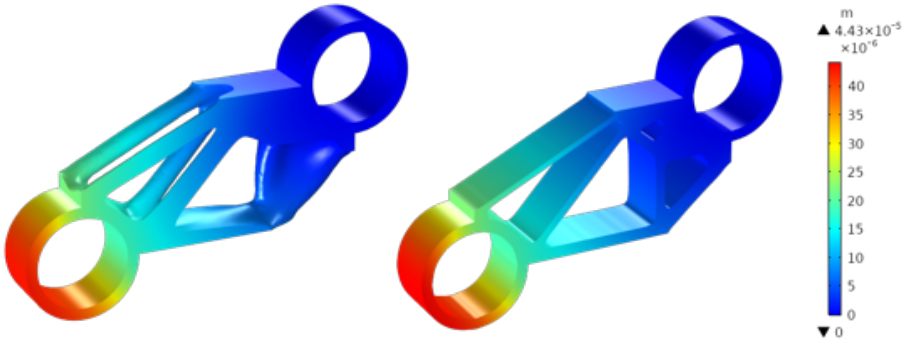


FIGURE 2.18: Displacement distribution of the full-scale 3D design (left) and the 2D extrusion design (right).

the four internal threads. Regarding the load conditions, a surface traction with a 45° inclination with respect to the horizontal axis is applied on the inner wall of the ring, as shown in Fig. 2.19.

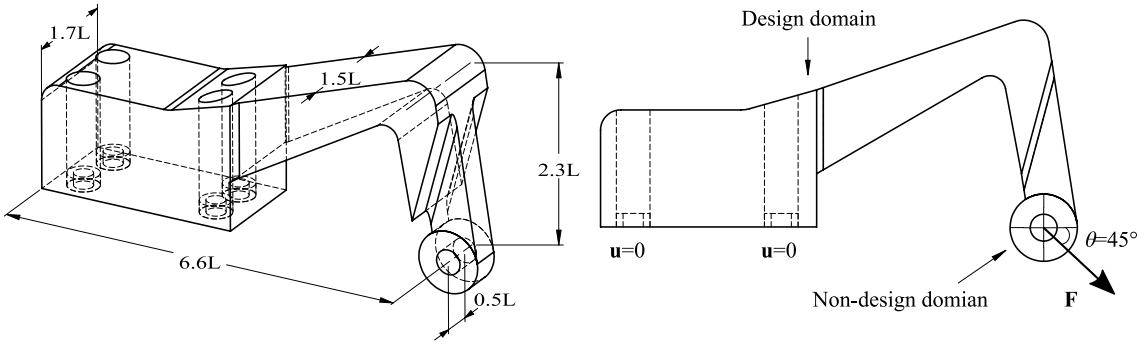
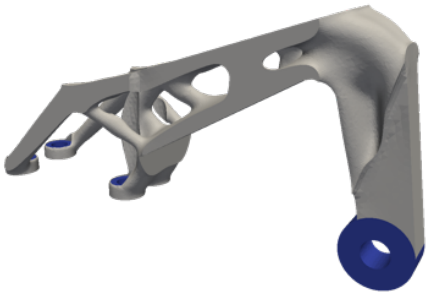


FIGURE 2.19: Design model for the airplane bearing bracket.

To model the bearing bracket, the 3D model is formed in a CAD software and saved as a standard CAD format of STEP AP203. Next, the CAD data is imported into the open-source tool Gmsh (Geuzaine and Remacle, 2009), and the 3D mesh for the first iteration can be generated using the built-in Delaunay 3D mesh algorithm. The maximum element size is set to 10^{-2}. Then the grid data (.mesh format) including region and boundary information can be read by FreeFEM to start the optimization.

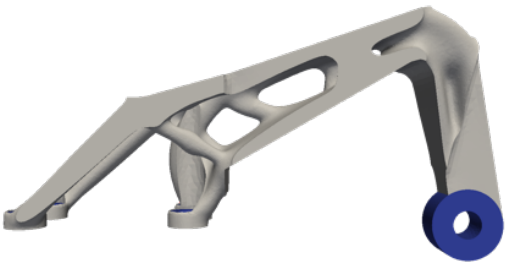
The design result shows a free-form geometry as presented in Fig. 2.20. It should be mentioned that the configuration is symmetric even though the initial mesh is not distributed in a perfectly symmetric way and there is no symmetric boundary condition. The computation is performed using 20 processes and the total runtime is 4.56 h for 150 iterations with a final number of elements of 1.1 million.

Bridge. Adaptive Mesh v.s. Fixed Mesh. The surface traction is applied downward on the top surface. The left and right ends of the bottom surface are fixed. The design domain is $L \times 3L \times 10L$. The maximum volume fraction is set to 15%. Fig. 2.21 shows the optimal solution. In this bridge example, to obtain a more feature-rich shape, we set τ to 10^{-5} and we push the element number to 3.1 million for the adaptive mesh case. To validate our body-fitted adaptive mesh scheme, we run our TO algorithm by using the fixed mesh, as shown in Fig. 2.21b. The number of elements for the fixed-mesh case is approximately 6.01 million. The evolution of the objectives are recorded in Fig. 2.22. The compliance curves for the two schemes closely match each other. From Fig. 2.21a and Fig. 2.21b, it can be observed that the optimal structures obtained by both schemes also look similar.



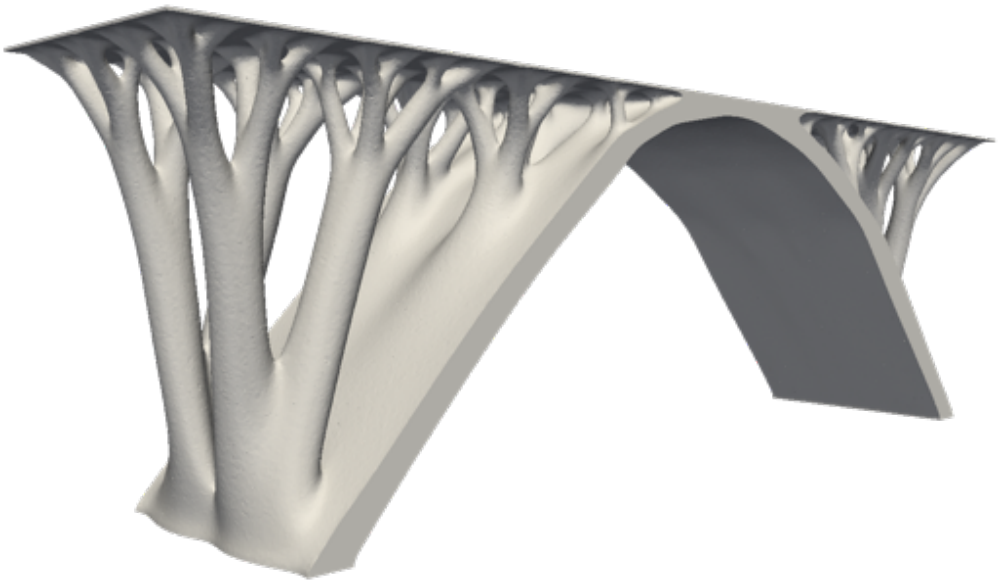
(A) Isometric view

(B) Top view

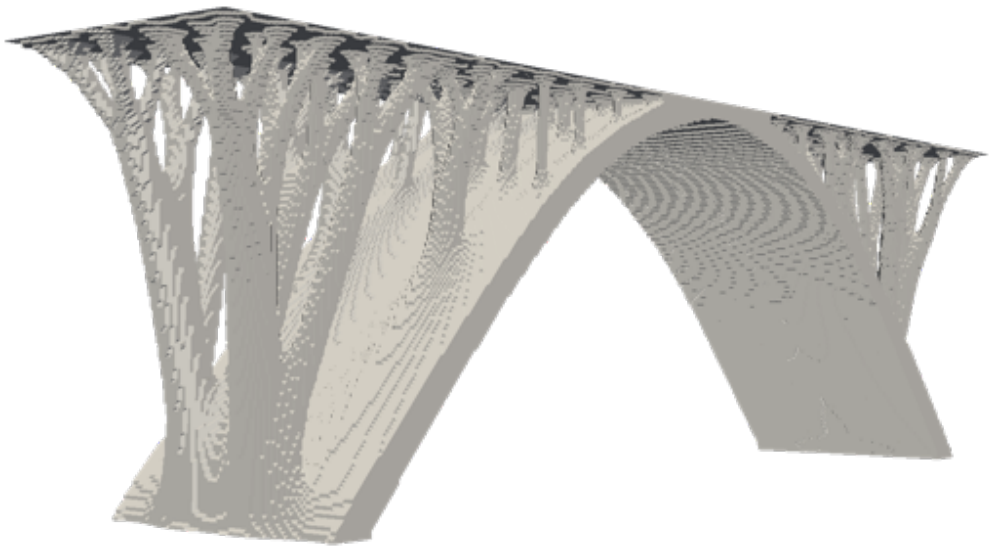


(C) Front view

FIGURE 2.20: Design result for the airplane bearing bracket.



(A) Body-fitted adaptive mesh, $3.1 \cdot 10^6$ elements.



(B) Fixed mesh, $6.01 \cdot 10^6$ elements.

FIGURE 2.21: Design result for the bridge example.

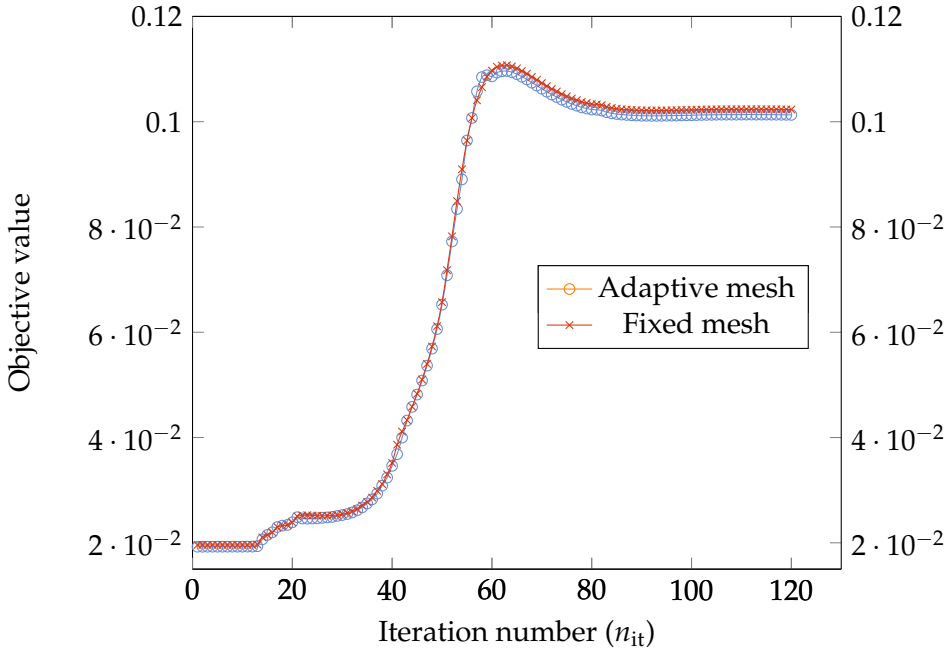


FIGURE 2.22: Comparison of the compliance evolution for bridge example using the adaptive and fixed mesh, c.f. Fig. 2.21.

2.4.5 Limitation of the framework

One of the main limitations of the present framework is that the mesh adaptation step is done in sequential, meaning that a global mesh has to be centralized. This can be quite memory demanding. For example, the bridge case (Fig. 2.21a) has an input mesh with 3.1 million elements at the last iterative step. It takes over 26.3 min for one step of mesh adaption, in which the "mesh quality improvement" step takes more than 2/3 of the total mesh evolution time. This step is the most tedious of the whole process which includes the mesh manipulation such as split the edges that are "too long", merge two endpoints of and edge, swap connectivities, etc., see details in Feppon, Allaire, Bordeu, Cortial, and Dapogny (2019), Dapogny, Dobrzynski, and Frey (2014), and Bacuta (2006).

We further examine the above-mentioned limitation by running the proposed algorithm on the two high-resolution TO problems presented in two state-of-the-art works.

The first test case is a "hand" example inspired by Wu, Dick, and Westermann (2015). In their work, GPU was used for the acceleration of their TO algorithm. Fig. 2.23a shows the design model. The surface traction is applied on the index, middle, and ring fingers and the right end is fixed. The optimal solution is plotted in Fig. 2.23b. The number of tetrahedra for the last iteration is $5.24 \cdot 10^5$ and the total runtime is 87 min for 100 iterations. As for comparison, Wu, Dick, and Westermann (2015) used $4.39 \cdot 10^6$ voxels and the total runtime was 421 s for 31 iterations. We can clearly see that the sequential part of Mmg makes it not possible to run this in 30 min or less.

The second test case is a "bird beak" example inspired by Liu, Hu, Zhu, Matusik, and Sifakis (2018). Fig. 2.24 shows the slice view of the optimal solution. The surface traction is applied on the normal of the outer surface and the bottom end is fixed. There are around $2.93 \cdot 10^6$ tetrahedra for the last iteration and the total runtime is around 25.4 h for 120 iterations. However, the resulting geometry is still much less sparse compared with a real bird beak bone structure. In the work of Liu, Hu, Zhu, Matusik, and Sifakis (2018), their algorithm was able to obtain a more thinner and spares features by using over billions of voxels. It is worth noting that they used the *sparsely populated grid data structure*, which allows for the dynamic allocation of the degrees of freedom within a narrow band only around the structure. In this sense, our mesh adaptation strategy shares a similar idea with theirs to reduce the overall computational cost. Hence, it may be deduced that we need to increase the element number by at least an order of magnitude to pursue an ultra-high resolution result. Indeed, the current framework (integrated with the sequential version of Mmg) is still not scalable to billions of voxels, but is targeted in our future work.

Once the parallel version of Mmg will be able to handle body-fitted mesh adaption, all steps will be fully parallel, so that the memory consumption per MPI process will be well under control, since we are using multigrid solvers which have a rather low memory-footprint (compared to more expensive solvers such as exact *LU* factorizations). In that case, we will be able to solve large problems, e.g., up to 10 million elements or even higher. Hence, we can expect to obtain a more feature-rich shape.

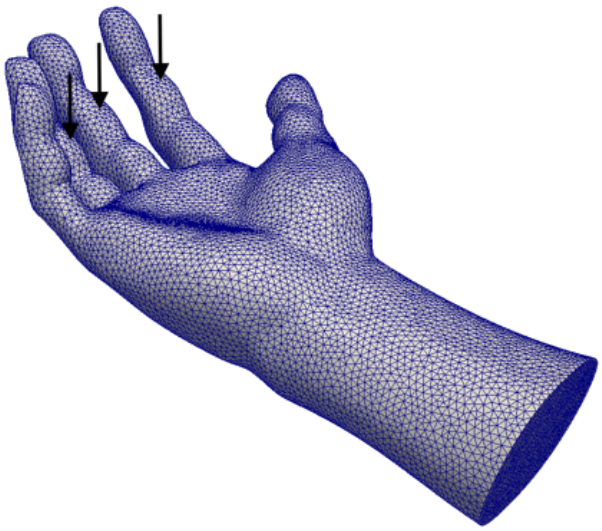
2.5 Manufacture of the prototype

Now, we translate the design results obtained in Section 2.4 to prototypes using 3D printing technology. The benefits of the body-fitted adaptive mesh can be further demonstrated in this section.

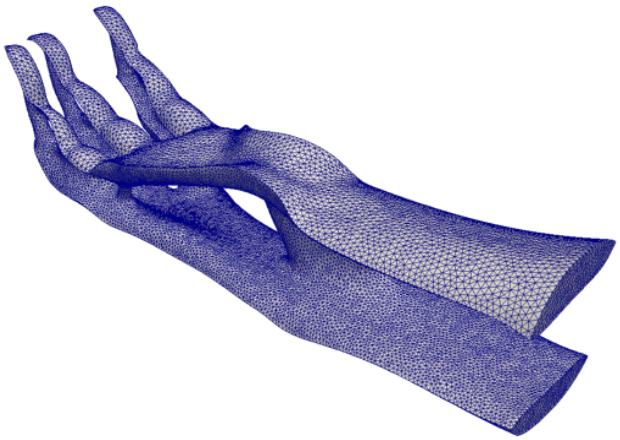
2.5.1 Preparation for printing data

First, the established models file needs to be converted into the stereolithography data (STL) written by binary code or ASCII before it can be imported into the slice software for code compilation. Thanks to Mmg, the solid and void domains can be marked by different region labels, hence the solid domain can be directly extracted and converted to STL files in the open-source data analysis and visualization platform ParaView. Then, the STL data can be compiled by the slice software and finally imported into the 3D printer for printing.

It should be mentioned that we do not use any density threshold or mesh reconstruction to determine the solid–void boundary because the boundary is described by the level-set interface and Mmg has inserted nodes at the intersection between the level-set and the edges, as discussed in Section 2.3.2.



(A) Initial shape.



(B) Optimal solution.

FIGURE 2.23: The hand example inspired by Wu, Dick, and Westermann (2015).

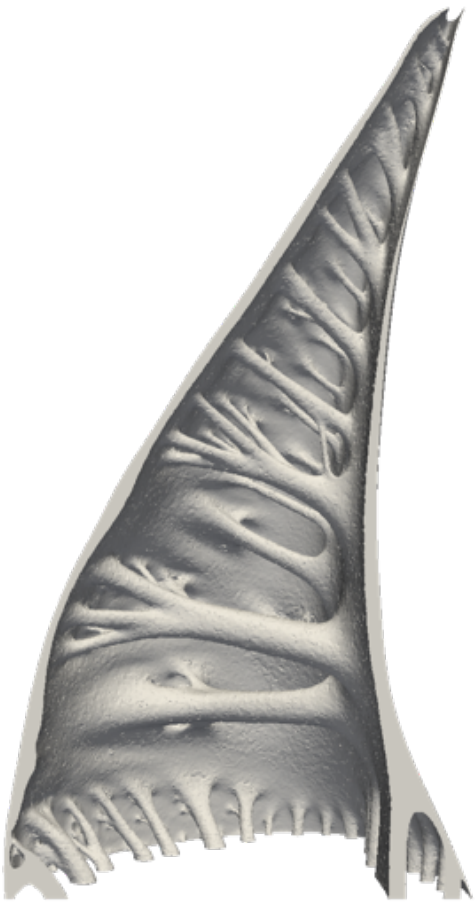


FIGURE 2.24: The bird beak example inspired by Liu, Hu, Zhu, Matusik, and Sifakis (2018).

2.5.2 3D printed prototypes

The topology optimized models are printed by the Stratasys Objet^{TM} Eden260VS at *Manufacturing Systems Engineering Laboratory* (MSEL), Kyoto University. The resolution of the 3D printer is 600 DPI in the x-direction, 600 DPI in y-axis, and 1600 DPI in z-axis, with the minimum layer thickness of 16 microns. The printing accuracy is 20–85 microns for features below 50 mm and up to 200 microns for full model size. TheVeroGray-FullCure 850 is used as the model material. FullCure support material, used in combination with the FullCure model material, can be easily removed with a water-jet or by hand after printing.

The design results as shown in Figs. 2.10, 2.15, 2.17 and 2.20 are printed out. The photographs of the 3D printed prototypes are presented in Fig. 2.25. It can be clearly observed from the zoom views that the surface finish of the prototypes featured a high resolution and smoothness. It should be emphasized that to get such fine surface finishes, we "only" need around 1 million tetrahedral elements during the optimization phase.

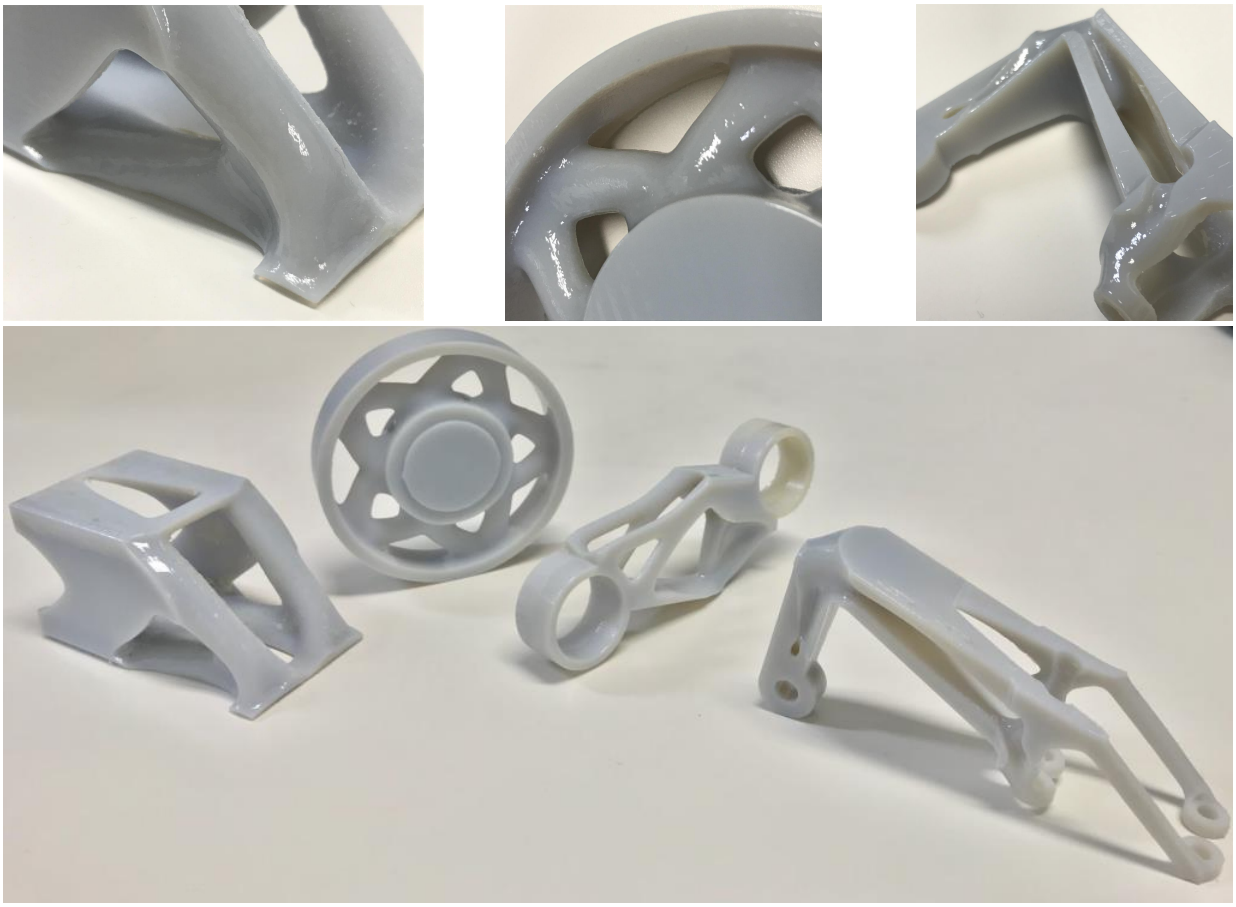


FIGURE 2.25: Photographs of the prototypes. Top figures: zoom views of the prototypes; Bottom figure, from left to right: cantilever, wheel-like structure, bracket-like structure and airplane bearing bracket.

2.5.3 Discussion of the post-processing

As mentioned above, in this work, the boundary representation (B-Reps) is sampled with the volumetric representation of body-fitted tetrahedral meshes based on a level-set. The 3D printing data is directly converted from obtained generative meshes. However, Marinov, Amagliani, Barback, Flower, Barley, Furuta, Charrot, Henley, Santhanam, Finnigan, et al. (2019) pointed out that this conversion inevitably introduces a loss of surface precision due to the limitations of the target representation, while this loss is negligible in the topology optimization computation. Therefore, this conversion is more convenient and efficient for the physical simulations, but when integrating the optimized part in the whole assembly, the loss of the surface accuracy (or surface residual) may lead to manufacturing deficiency such as part interference.

As a matter of fact, reviewing the details of converting the input mesh (optimized result) to the B-Reps is out of the scope of the present work, here we refer to several related research which may provide the reader some potential solutions. In the CAD industry, the approaches that approximates the entire mesh with a global quad parameterization algorithm seems to be one of the solutions (Myles, Pietroni, and Zorin, 2014; Campen, Bommes, and Kobbelt, 2015). The idea is to apply Boolean operations to remove the interferences and add the missing volume. Some commercial CAD software include a built-in function that can reconstruct a B-Reps model using an input mesh as a geometry reference (Autodesk, n.d.; Siemens, n.d.). This may require manual efforts to smoothen surfaces with the processed mesh geometry. Very recently, Marinov, Amagliani, Barback, Flower, Barley, Furuta, Charrot, Henley, Santhanam, Finnigan, et al. (2019) proposed a fully automatic method for the conversion of a general generative design to a watertight B-Rep composed of the exact residual geometry of the input solids and editable T-NURCC surfaces. Their method requires only the rudimentary data from the solver, and a boundary mesh of the optimized design, which appears to be applicable to our problem.

2.6 Summary

In this chapter, we have presented an RDE-based level-set TO algorithm, in which we integrate mesh adaption in a parallel computing framework to solve full-scale 3D problems. The main idea is to utilize the level-set-based mesh evolution to achieve an essentially clear and high resolution solid–void material boundary, and at the same time to pursue a cheaper computational cost by coarsening the mesh distributed in the void domain. To this end, first, the compliance minimization TO problem has been formulated. Secondly, the parallel strategy adopted in FEA has been presented in detail. Thirdly, the mesh refinement algorithm which discretizes and optimizes an implicitly defined surface has been illustrated. After that, several testing examples have been presented to verify the correctness and the effectiveness of the proposed method in both physically

reasonable sense and computationally cost-efficient sense. Finally, the prototypes are printed out. The testing results can be concluded as follows:

1. The suggested TO method can mitigate the dependence to initial guess to some extent for the minimum compliance problem due to the use of topological derivative. Furthermore, the dependence to mesh resolution can be restrained because of the diffusive term in the RDE together with adaptive mesh refinement along the zero-level-set interface. The regularization parameter τ can be used to control the complexity of the optimal design according to engineering preferences.

2. The computational code efficiency is investigated using fixed meshes and body-fitted adaptive meshes for 3D cantilever examples. The scalability in terms of runtime is tested on a standard workstation. Results show that the total runtime for the adaptive mesh ($\leq$ 1 million elements, 3.26 h) is 36.4% shorter than that for the fixed mesh ($\leq$ 3 million elements, 5.08 h) while the CPU usage of the adaptive mesh-based case is only 1/4 of that used in the fixed-mesh case. More importantly, the boundary obtained with body-fitted adaptive meshes is essentially of high-resolution and clear even though the maximum number of elements is only 1/3 of that with fixed meshes; In the present work, the maximum problem scale is limited to no more than 3 million element due to the sequential adaptive mesh package Mmg.

3. Full-scale 3D designs can discover more feature-rich shapes compared to the extrusion of 2D designs. In the bracket-like structure, the simulation shows that the maximum displacement of the 3D design is 3.7% smaller than that of the extrusion of the 2D design;

4. Two engineering cases (airplane bearing bracket and bridge) are presented to highlight the applicability of the proposed TO methodology to industrial product design. The B-Reps is sampled with the volumetric representation of body-fitted tetrahedral meshes based on level-sets, which can be efficiently treated during post-processing stage.

In the later chapters, we will conduct future research to apply the proposed TO algorithm to the multiphysics, i.e., 3D fluid–structure interaction problems. Furthermore, mesh adaptation is currently performed sequentially on a single MPI process. So far, we present the “moderately" large scale problem. In near future, once the parallel version of Mmg will be able to handle surface mesh adaptation, it will be possible to switch to a fully distributed framework which can solve a larger scale problem.

Chapter 3

Topology Optimization of Fluid–Structure Systems[†]

CONTENTS

3.1 Introduction

Topology optimization (TO) technique has become mature in the field of structural optimization and it has been widely used in real-world industry. One of the typical application scenarios is to design innovative architectural structures. For example, Beghini, Beghini, Katz, Baker, and Paulino (2014) focused on the close collaboration between the aesthetics of a structure and its

[†]The work in this chapter has been published in *Applied Mathematical Modelling*, DOI: 10.1016/j.apm.2021.08.021.

stability. They presented an integrated design idea by means of a tailored structural TO framework based on the density approach. Dapogny, Faure, Michailidis, Allaire, Couvelas, and Estevez (2017) formulated three novel constraints on the geometry of shapes in order to generate a conceptual architectural design which satisfies the user's personal aesthetic taste and handles several fabrication issues. Their work is based on the level-set method. He, Cai, Zhao, and Xie (2020) presented a stochastic approach to generate diverse and competitive structural designs based on the bi-directional evolutionary structural optimization (BESO) technique. Their proposed algorithm is recently available in the commercial TO software package Ameba (Zhou, Shen, Wang, Zhou, and Xie, 2018; XIE-Engineering-Technologies., 2021). It can be said that notable achievements have been made by applying the TO techniques to solve the classical structural optimization problem. However, new challenges keep attracting considerable scientific interests. For example, the rapid development of the large-scale computation technique allows one to obtain an optimal solution which has an ultra-high resolution structure (Liu, Tian, Zong, Ma, Wang, and Zhang, 2019; Träff, Sigmund, and Aage, n.d.; Aage, Andreassen, Lazarov, and Sigmund, 2017). Though this is not the focus of the present work, it is targeted in the future plan.

As discussed above, the TO method has its origin in the structural optimization problem, but was lately extended to fluid-based optimization. Borrvall and Petersson (2003) are known as the very first who introduced the TO concept into fluid optimization. In their pioneering work, the design parametrization is based on the lubrication theory. They presented an in-depth mathematical basis for TO of Stokes flow. This idea was extended to Navier–Stokes flow by Gersborg-Hansen, Sigmund, and Haber (2005). In both of the works, they pointed out the similarity of the obtained equations with that of the flows observed in the porous medium, which can be modelled by the Brinkman's law or the Darcy's law. Soon after that, Olesen, Okkels, and Bruus (2006) used a high-level programming language, Femlab, to implement the TO for the steady-state Navier–Stokes flow using the fictitious porous media approach. These early works established the foundation for the fluid-based topology optimization. Since then, a large number of contributions have expounded the TO method to a variety of fluidic problems.

In general, there are two different strategies to solve the fluid optimization problem. The first option is that of the density-based approach. The idea is that the design domain is penalized by the fictitious porous media with very low (resp. high) permeability to represent the solid (resp. fluid) (Aage, Poulsen, Gersborg-Hansen, and Sigmund, 2008). However, with this option, it is hard to impose the corresponding boundary conditions on the fluid–solid boundaries due to the existence of the gray elements. The second option is that of the surface-capturing level-set methods. Duan, Ma, and Zhang (2008) are known as the first to apply the variational level-set method to fluid topology optimization obtaining similar designs to those previously studied using the density-based approach. In the work by Zhou and Li (2008), they presented a framework excluding elements outside the fluid domain to improve the accuracy of the no-slip boundary condition, but increasing book-keeping through updating the fluid mesh at each iterative step. The later

contributions on LSM, where cut elements are incorporated using a special scheme such as the extended finite element method (XFEM) (Kreissl and Maute, 2012) or globally remeshing techniques such as body-fitted adaptive meshes (Feppon, 2019). This option requires the advection of the level-set function. In recent years, some attempts have been made to take good advantage of both their merits. For example, Behrou, Ranjan, and Guest (2019) introduced the Brinkman model to formulate the incompressible laminar flow problem with mass flow constraint. They integrated an adaptive no-slip boundary condition scheme within the context of this TO framework in order to accurately capture flow characteristics and to reduce computation cost. Duan, Li, and Qin (2016) used the Ersatz material-based level-set method to solve a fluid flow optimization problem. In their work, they used the level-set function to describe the material phase interface, which is naturally easier for identifying the fluid–solid boundary compared with the density-based approach. At the same time, they introduced the Brinkman model in the flow modelling, and the topological gradient was used to nucleate new holes during the topology evolution. Here, we mention that the Ersatz material method essentially shares the same idea with the density-based approach in the sense of the material interpolation. Note that in the present work, our level-set method is also one of the Ersatz material approaches but based on the reaction–diffusion equation (RDE) proposed by Yamada, Izui, Nishiwaki, and Takezawa (2010).

Regarding the numerical analysis method being incorporated in the fluid optimization, the finite element analysis (FEA) is the most widely used method in the literature (Sato, Yaji, Izui, Yamada, and Nishiwaki, 2018; Yaji, Yamada, Kubo, Izui, and Nishiwaki, 2015; Li, Ding, Meng, Jing, and Xiong, 2019). Besides, several other options can be also adopted. For example, Yu, Ruan, Gu, Ren, Li, Wang, and Shen (2020) presented a density-based TO framework for a coupled thermal-fluid–structure problem. Their solver is based on the finite volume method (FVM), relying on the open-source FVM software OpenFOAM. Sasaki, Sato, Yamada, Izui, and Nishiwaki (2019) introduced the moving particle semi-implicit (MPS) method in fluid optimization to avoid surface tracking by a mesh or a scalar quantity. Yaji, Yamada, Yoshino, Matsumoto, Izui, and Nishiwaki (2014) and Yaji, Yamada, Yoshino, Matsumoto, Izui, and Nishiwaki (2016) formulated a level-set-based topology optimization using the lattice Boltzmann method (LBM) and the continuous adjoint sensitivity analysis was based on the Boltzmann equation. Other related fluid-based optimization research work can be referred to in a comprehensive review paper by Alexandersen and Andreasen (2020).

After dealing with the TO problem for the fluid flow and the solid structure separately, the next step was to combine these two physical problems together as the fluid–structure interaction (FSI) problem. In the context of FSI optimization, the problem setup can be classified into two categories according to the way that the fluid–solid interfaces are treated (Maute and Allen, 2004).

The first category of the FSI optimization is known as design-independent (or "dry" FSI), implying that only the infill structure can be optimized. Jenkins and Maute (2015) combine an explicit level-set method and the extended finite element method (XFEM-LSM) to solve both two- and

three-dimensional "dry" FSI problems considering a strongly-coupled condition.

The second category of the FSI optimization is known as design-dependent (or "wet" FSI), implying that not only the infill structure but also the fluid–solid boundary can be optimized simultaneously. This "wet" boundary problem can be modeled in three different approaches. Yoon (2010b) pioneered the idea of using a monolithic approach to interpolate both structural and fluid equations based on the density model. Following their work, Lundgaard, Alexandersen, Zhou, Andreasen, and Sigmund (2018) revisited the density-based method for an FSI problem and provided further comparison. In both of the works, the fixed mesh was used. The second strategy to solve "wet" FSI is to use the level-set-based method. For example, Jenkins and Maute extended the XFEM-LSM reported in their previous work (Jenkins and Maute, 2015) to the "wet" FSI problem (Jenkins and Maute, 2016). In their work, the local mesh of the fluid–structure boundaries was remeshed. Feppon, Allaire, Bordeu, Cortial, and Dapogny (2019) and Feppon, Allaire, Dapogny, and Jolivet (2020b) integrated a body-fitted mesh evolution method (Allaire, Dapogny, and Frey, 2013) into their LSM which relies on the boundary variation of Hadamard for the description of the sensitivity with respect to the domain. In their work, a full-scale three-dimensional FSI test case was presented. In their following work, they extended this methodology to optimize a two-flow heat exchanger (Feppon, Allaire, Dapogny, and Jolivet, 2021). Picelli, Neofytou, and Kim (2019) adopted LSM to optimize structures under hydrostatic pressure loads using the Ersatz material approach and a least-squares interpolation of the fluid–structure responses at the boundaries. Note that in the aforementioned LSM based works, the shape sensitivity was used, indicating that the new holes cannot be generated during the topology evolution. The third strategy to solve "wet" FSI is to use the binary changes between the fluid and structure. Picelli, Vicente, and Pavanello (2017) applied the bi-directional evolutionary structural optimization (BESO) method to disjoint the fluid–structure system into fluid and solid subdomains, and the FSI framework can be modeled with separate governing equations. In their following work (Picelli, Ranjbarzadeh, Sivapuram, Gioria, and Silva, 2020), they applied a recently proposed topology optimization of binary structures (TOBS) method (Sivapuram and Picelli, 2018) to solve an FSI problem and showed its numerical implementation using COMSOL Multiphysics software. In this work, a concise "specification" of the above-mentioned three different strategies was summarized in a selection sheet.

In the present work, we propose a new topology optimization framework for the weakly-coupled fluid–structure systems. To the best of our knowledge, this is the first attempt to use the reaction–diffusion equation-based level-set method to solve the FSI optimization problem. By using the Ersatz material approach for the material interpolation, and the topological sensitivity as the reaction source term in the RDE, the proposed method enables the nucleation of new holes during the fluid–structure topology evolution. Meanwhile, for the RDE-based TO method, this is the first attempt to use the body-fitted adaptive mesh to solve the fluid-based optimization problems. This enables the disjoint reunion of the subdomains according to the zero level-set interface. With this remeshing technique, the fluid–solid boundary can be described explicitly. This allows us to model the fluid–solid system with separate governing equations, which means that the fictitious

body force term is only to aid the topological sensitivity, and can be excluded from the forward problem analysis. FreeFEM is used for the finite element analysis, and PETSc (Portable, Extensible Toolkit for Scientific Computation) for scalable (parallel) solution of scientific applications modeled by partial differential equations. The moderately large-scale finite element systems are solved by using efficient preconditioners. By calling *hypre* to solve the RDE, the level-set updating part of the solver is scalable with respect to the problem size. From an engineering point of view, we construct a complete product development workflow: pre-processing, topology optimization, post-processing (B-Rep conversion), and numerical experiment. We test our design methodology by solving three classical topology optimization problems: compliance, power dissipation and the FSI problem (including both "wet" and "dry" FSI). For each optimum design problem, we benchmark our designs against state-of-the-art works followed by showcasing a variety of more practical engineering design examples.

The remainder of this chapter is organized as follows. Section 3.2 formulates the governing equation, the RDE-based TO method, and the mathematical optimization model for the three classical TO design problems. Section 3.3 illustrates the implementation details including the parallel computing strategy, the body-fitted adaptive mesh evolution method, and the post-processing strategy. In Section 3.4, a variety of two- and three-dimensional benchmarks and several engineering design cases are presented to demonstrate the effectiveness of the proposed design methodology. Lastly, the conclusion and prospective works are documented in Section 3.5.

3.2 Formulation

In this section, the mathematical and physical frames are described. First, we present in Section 3.2.1 the governing equation considering two physics models coupling fluid flow and elasticity. In Section 3.2.2, we revisit the reaction–diffusion equation-based level-set topology optimization method. We then formulate in Section 3.2.3 the optimization mathematical model considering three optimum design problems in this context.

3.2.1 Governing equation

In the present work, $\Omega = \Omega_s \cup \Omega_f$ denotes the computational domain in $\mathbb{R}^d$, where $d = 2$ or 3. It is the disjoint reunion of a fluid domain Ω_f and a solid domain Ω_s, divided by the fluid–solid interface $\Gamma_{s,f}$.

The weakly-coupled physics models are solved in Ω:

(1) the basic motion of a fluid particle within the fluid domain Ω_f is characterized by the velocity $\boldsymbol{v} : \Omega_f \to \mathbb{R}^d$, and pressure $p : \Omega_f \to \mathbb{R}$;

(2) the solid region Ω_s is described by the displacement $\boldsymbol{u} : \Omega_s \to \mathbb{R}^d$.

Fluid flow

The governing equation of the fluid flow is formulated under the assumptions of: (1) incompressible steady-state fluid flow; (2) the fluid–solid interface satisfies no-slip velocity boundary condition; and (3) the gravity effect is neglected. With these assumptions, the fluid flow is governed by the following continuity and Navier–Stokes equations in the non-dimensional forms, which are characterized by the dimensionless velocity $\boldsymbol{v}$ and pressure p, with the Reynolds number Re and dimensionless gradient operator ∇.

$$\nabla = L\nabla^*, \quad \boldsymbol{v} = \frac{\boldsymbol{v}^*}{V}, \quad p = \frac{p^*}{\rho V^2}, \quad Re = \frac{\rho V L}{\nu}, \tag{3.1}$$

where $\boldsymbol{v}^*$, p^*, ∇^* denote the dimensional velocity, pressure, and gradient operator, respectively. U and L are the characteristic velocity and characteristic length, and ρ and ν are the fluid density and dynamic viscosity, respectively.

The non-dimensional continuity and Navier–Stokes equations, without body force, are then given as:

$$\begin{cases} -\operatorname{div}\left(\sigma_f\left(\boldsymbol{v}, p\right)\right) + \left(\boldsymbol{v} \cdot \nabla\right)\boldsymbol{v} = 0 & \text{in } \Omega_f \\ -\operatorname{div}\left(\boldsymbol{v}\right) = 0 & \text{in } \Omega_f \\ \boldsymbol{v} = \boldsymbol{v}_0 & \text{on } \partial\Omega_f^D \\ \sigma_f\left(\boldsymbol{v}, p\right) \cdot \boldsymbol{n}_f = 0 & \text{on } \partial\Omega_f^N \\ \boldsymbol{v} = 0 & \text{on } \Gamma_{s,f}, \end{cases} \tag{3.2}$$

where $\boldsymbol{n}_f$ is the unit normal vector which is oriented from Ω_f to Ω_s. The fluid stress tensor $\sigma_f\left(\boldsymbol{v}, p\right)$ is defined related to the strain tensor $e\left(\boldsymbol{v}\right) := \frac{1}{2}\left(\nabla\boldsymbol{v} + \nabla\boldsymbol{v}^T\right)$ as follows:

$$\sigma_f\left(\boldsymbol{v}, p\right) := 2\frac{1}{Re}e\left(\boldsymbol{v}\right) - p\mathbf{I}, \tag{3.3}$$

where $\mathbf{I}$ is the identity matrix. In this context, the boundary conditions of the fluid flow as the disjoint reunion of $\Omega_f = \partial\Omega_f^D \cup \partial\Omega_f^N \cup \Gamma_{s,f}$ feature:

(1) on the inlet Ω_f^D, a Dirichlet boundary condition is imposed with a given velocity profile;

(2) on the outlet Ω_f^N, a zero normal stress is given;

(3) on the fluid–structure interface $\Gamma_{s,f}$, a no-slip boundary condition is observed.

For the Navier–Stokes equation defined in Eq. (3.2), using the finite element analysis (FEA), the weak forms of the Jacobian and the residual at a given linearization point $(\boldsymbol{v}, p)$, for an increment

$(\delta \boldsymbol{v}, \delta p)$, are as follows:

$$\begin{aligned} F(\boldsymbol{v}, p) = \int_{\Omega_f} \left((\boldsymbol{v} \cdot \nabla) \boldsymbol{v} \right) \cdot \tilde{\boldsymbol{v}} + \frac{1}{Re} \left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T \right) : \nabla \tilde{\boldsymbol{v}} \\ - p \nabla \cdot \tilde{\boldsymbol{v}} - \tilde{p} \nabla \cdot \boldsymbol{v} d\Omega \quad \forall (\tilde{\boldsymbol{v}}, \tilde{p}) \in \tilde{\boldsymbol{V}} \times \tilde{P} \end{aligned} \tag{3.4}$$

$$\begin{aligned} DF(\boldsymbol{v}, p)(\delta \boldsymbol{v}, \delta p) = \int_{\Omega_f} \left((\delta \boldsymbol{v} \cdot \nabla) \boldsymbol{v} \right) \cdot \tilde{\boldsymbol{v}} + \left((\boldsymbol{v} \cdot \nabla) \delta \boldsymbol{v} \right) \cdot \tilde{\boldsymbol{v}} \\ + \frac{1}{Re} \left(\nabla \tilde{\boldsymbol{v}} + \nabla \tilde{\boldsymbol{v}}^T \right) : \nabla \delta \boldsymbol{v} \\ - \delta p \nabla \cdot \tilde{\boldsymbol{v}} - \tilde{p} \nabla \cdot \delta \boldsymbol{v} d\Omega \quad \forall (\tilde{\boldsymbol{v}}, \tilde{p}) \in \tilde{\boldsymbol{V}} \times \tilde{P}, \end{aligned} \tag{3.5}$$

where $\tilde{\boldsymbol{v}}$ and $\tilde{p}$ are the test functions for the increments $\delta \boldsymbol{v}$ and δp, respectively. Then, the nonlinear problem is solved using a Newton method where the following linearized equations are solved in sequence:

$$DF(\boldsymbol{v}, p)(\delta \boldsymbol{v}, \delta p) = F(\boldsymbol{v}, p). \tag{3.6}$$

Lamé system

Here, we formulate the governing equation for the structure occupying a solid domain Ω_s, under the assumptions that: (1) small displacements and deformation are observed (linear elasticity); (2) the body force and gravity effect is neglected; and (3) when the solid is interacting with the viscous fluid flow load, the deformations of the fluid–structure boundaries are sufficiently small, indicating that the change on the fluid flow domain is negligible. Thus, the weakly-coupled condition is used in the FSI computation.

With the aforementioned assumptions, the solid can be governed by the following Lamé equation, which is characterized by the dimensionless displacement $\boldsymbol{u}$, as follows:

$$\begin{cases} -\operatorname{div} \sigma_s(\boldsymbol{u}) = 0 & \text{in } \Omega_s \\ \boldsymbol{u} = \boldsymbol{u}_0 & \text{on } \partial\Omega_s^D \\ \sigma_s(\boldsymbol{u}) \cdot \boldsymbol{n}_s = \boldsymbol{g} & \text{on } \partial\Omega_s^N \\ \sigma_s(\boldsymbol{u}) \cdot \boldsymbol{n}_s = -\dfrac{1}{E} \sigma_f(\boldsymbol{v}, \boldsymbol{p}) \cdot \boldsymbol{n}_f & \text{on } \Gamma_{s,f}. \end{cases} \tag{3.7}$$

The resulting stress tensor is non-dimensionalized by the Young's modulus E. The solid stress tensor $\boldsymbol{\sigma}_s(\boldsymbol{u})$ is defined related to the strain tensor $e(\boldsymbol{u}) := \dfrac{1}{2} \left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T \right)$ as follows:

$$\boldsymbol{\sigma}_s(\boldsymbol{u}) = 2\mu e(\boldsymbol{u}) + \lambda \mathbf{Tr}(e(\boldsymbol{u})) \mathbf{I}, \tag{3.8}$$

where $\mathbf{Tr}(\cdot)$ is the trace operator. μ and λ are the Lamé coefficients. Now, we introduce the fourth-order elasticity tensor $\boldsymbol{C}$ as:

$$\boldsymbol{C}_{ijkl} := \lambda \delta_{ij} \delta_{kl} + \mu \left(\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk} \right), \tag{3.9}$$

where δ_{ij} is the Kronecker delta function. The governing equations in Eq. (3.7) can be rewritten as:

$$\begin{cases} -\operatorname{div}(\boldsymbol{C} : e(\boldsymbol{u})) = 0 & \text{in } \Omega_s \\ \boldsymbol{u} = \boldsymbol{u}_0 & \text{on } \partial\Omega_s^D \\ (\boldsymbol{C} : e(\boldsymbol{u})) \cdot \boldsymbol{n}_s = \boldsymbol{g} & \text{on } \partial\Omega_s^N \\ \sigma_s(\boldsymbol{u}) \cdot \boldsymbol{n}_s = -\dfrac{1}{E}\sigma_f(\boldsymbol{v}, \boldsymbol{p}) \cdot \boldsymbol{n}_f & \text{on } \Gamma_{s,f}, \end{cases} \tag{3.10}$$

In this context, the boundary conditions of the solid domain Ω_s feature:

(1) on the fixed wall $\partial\Omega_s^D$, the displacement is prescribed as $\boldsymbol{u} = 0$;

(2) a surface traction $\boldsymbol{g}$ is applied on the $\partial\Omega_s^N$;

(3) on the fluid–structure boundary $\Gamma_{s,f}$, the solid undergoes the pressure load by the fluid. An equality holds between the normal components of the fluid and solid stress tensor, $\sigma_s(\boldsymbol{u}) \cdot \boldsymbol{n}_s = -\dfrac{1}{E}\sigma_f(\boldsymbol{v}, \boldsymbol{p}) \cdot \boldsymbol{n}_f$.

Using the FEA, the elasticity governing equation Eq. (3.10) is expressed as follows:

$$\begin{aligned} &-\int_{\Omega_s} (e(\boldsymbol{u}) : \boldsymbol{C}) : e(\tilde{\boldsymbol{u}}) d\Omega + \int_{\partial\Omega_s^N \cup \Gamma_{s,f}} (\sigma_s(\boldsymbol{u}) \cdot \boldsymbol{n}_s \cdot) \tilde{\boldsymbol{u}} d\Gamma \\ &= 0 \quad \forall \tilde{\boldsymbol{u}} \in \tilde{\boldsymbol{U}}, \end{aligned} \tag{3.11}$$

where $\tilde{\boldsymbol{u}}$ denotes the test function for the displacement $\boldsymbol{u}$.

3.2.2 Level-set method in fluid-based TO problems

Here, we overview the reaction–diffusion equation-based level-set topology optimization method (Yamada, Izui, Nishiwaki, and Takezawa, 2010). A TO problem for the fluid–structure system is defined to include three different phases: fluid, solid, and internal hole (or void). Let the design domain be denoted as $D \subset \mathbb{R}^d$. A solid domain is then defined as $\Omega_s \subseteq D$ and a fluid domain Ω_f and a void domain Ω_h are represented as the complementary domain. In the present framework, we use one level-set function $\phi(\mathbf{x})$ to identify different phases. The zero level-set interface includes two sets of boundaries: fluid–solid interface $\Gamma_{s,f}$, and the solid–void interface $\Gamma_{s,h}$.

As for the dry-FSI problem (cf. Section 3.4.3), the fluid–solid interface $\Gamma_{s,f}$ does not move during the optimization. The Navier–Stokes equations are solved in Ω_f and the no-slip boundary condition is imposed on $\Gamma_{s,f}$. The flow equations are not solved in Ω_h. The three phases are expressed by the level-set function (see Fig. 3.1), as follows:

$$\begin{cases} 0 < \phi(\mathbf{x}) \leq 1 & \text{for } \mathbf{x} \in \Omega_{s\cup f} \\ \phi(\mathbf{x}) = 0 & \text{for } \mathbf{x} \in \Gamma_{s,h} \\ -1 \leq \phi(\mathbf{x}) < 0 & \text{for } \mathbf{x} \in \Omega_h. \end{cases} \tag{3.12}$$

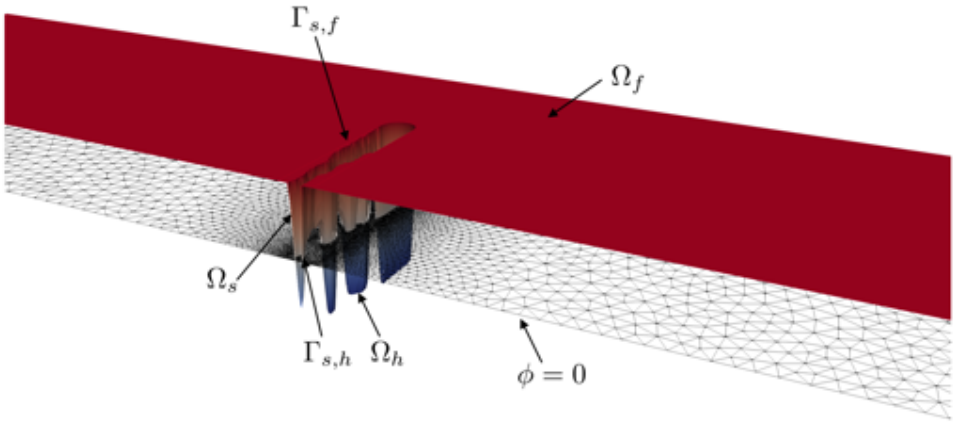


FIGURE 3.1: Level-set expression for the dry-FSI problem, cf. Section 3.4.3.

As for the wet-FSI problem (cf. Section 3.4.4), the fluid–solid interface $\Gamma_{s,f}$ is moving during the optimization. The three phases are expressed by the level-set function (see Fig. 3.2), as follows:

$$\begin{cases} 0 < \phi(\mathbf{x}) \leq 1 & \text{for } \mathbf{x} \in \Omega_s \\ \phi(\mathbf{x}) = 0 & \text{for } \mathbf{x} \in \Gamma_{s,f} \cup \Gamma_{s,h} \\ -1 \leq \phi(\mathbf{x}) < 0 & \text{for } \mathbf{x} \in \Omega_{f\cup h}. \end{cases} \tag{3.13}$$

The Navier–Stokes equations are solved in $\Omega_{f\cup h}$ and the no-slip boundary condition is imposed on all the zero level-set interface, namely, $\Gamma_{s,f} \cup \Gamma_{s,h}$. Therefore, the velocity and pressure in Ω_h will be equal to zero. The treatment of the internal holes will be further demonstrated in Sections 3.4.3 and 3.4.4.

Note that our level-set-based method uses the Ersatz material approach. The material field can be modeled using the characteristic function χ_ϕ given by:

$$\chi_\phi = \begin{cases} 1 & \text{if } \phi \geq 0 \\ 0 & \text{if } \phi < 0. \end{cases} \tag{3.14}$$

Then, the interpolation of the material properties can be done inside the design domain D. We will illustrate the expansion of governing equations with respect to the material interpolation in Section 3.2.3.

To update the level-set function iteratively, the reaction–diffusion equation is introduced, see details in Section 2.2.5. Note that the optimization problem described by the reaction equation is ill-posed. It is regularized by a diffusive term, which is used to guarantee the smoothness of the level-set function. This regularization can be traced back to the early works on shape and topology

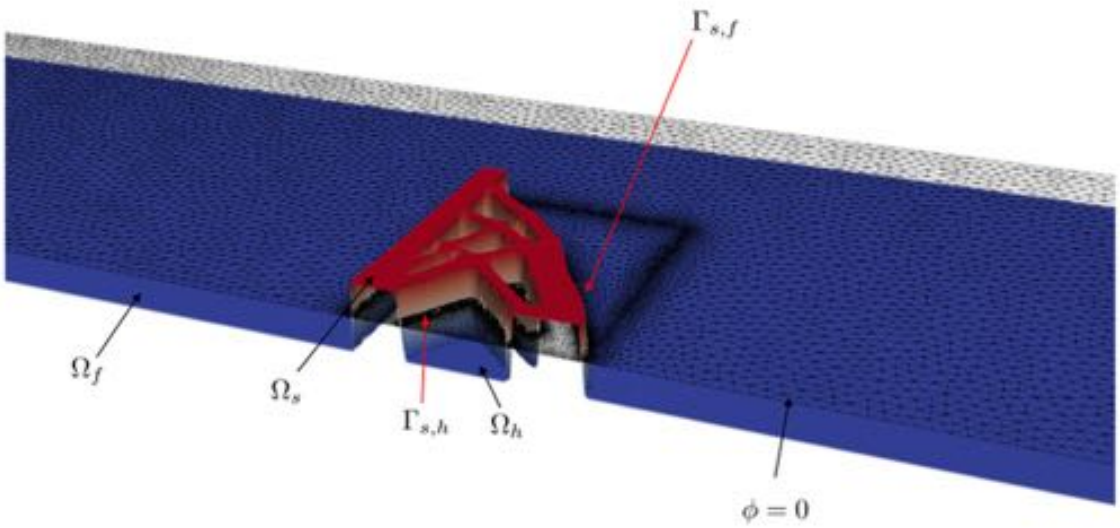


FIGURE 3.2: Level-set expression for the wet-FSI problem, cf. Section 3.4.4

optimization (Burger, 2003; Allaire, Jouve, and Toader, 2004), and it was re-discovered in Lazarov and Sigmund (2011). The values of the level-set function $\phi(\mathbf{x})$ are bounded in the range of $[-1, 1]$. This constraint works to equalize the effect of regularization. In other words, this constraint is necessary to ensure that the effect of the regularization term is approximately constant around the boundaries. As a result, the geometric complexity of the optimal structure can be controlled by adjusting the regularization factor τ. We also note here that the a PDE-based filter is implicitly included (see Eq. (3.15)).

$$-r^2\nabla^2\tilde{\gamma} + \tilde{\gamma} = \gamma, \tag{3.15}$$

where $\tilde{\gamma}$ is the value after filtering, while γ is the value before filter, r is the filtering radius. This filtering scheme can be used to mitigate the dependency of the optimal solution on mesh resolution to some extent (Lazarov and Sigmund, 2011).

3.2.3 Topology optimization problem

Now we present three optimum design problems to be solved in this paper. For each design problem, we first clarify the Ersatz material property and how it is interpolated by the characteristic function χ_ϕ. Then, we formulate the optimization mathematical model, and present the design sensitivity $\overline{F}'$ which acts as the reaction source term to compute the level-set field using Eq. (2.36).

Minimization of compliance

The first context of interest is that of topology optimization in solid mechanics. Considering a domain Ω_s under a surface traction, we intend to find the optimal structure which can minimize

the mean compliance (or maximize the stiffness) under volume constraint, as illustrated in Fig. 3.3.

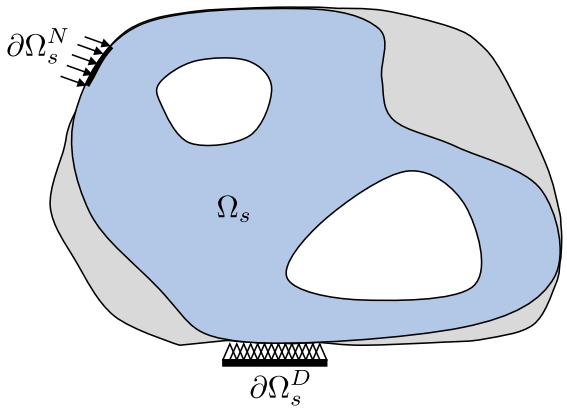


FIGURE 3.3: Schematic of the minimum compliance optimization problem.

In the structural problem, the elasticity tensor $\boldsymbol{C}$ is expanded as:

$$\boldsymbol{C}_{\chi_\phi} = \chi_\phi \boldsymbol{C}. \tag{3.16}$$

We then formulate the optimization mathematical model as follows:

$$\inf_{\chi_\phi \in \mathcal{X}} J(\Omega) = \int_{\partial\Omega_s^N} \boldsymbol{g} \cdot \boldsymbol{u}\, d\Gamma \tag{3.17}$$

$$\text{s.t.} \begin{cases} G_1 = \dfrac{\int_D \chi_\phi d\Omega}{\int_D d\Omega} - V_{\max} \leq 0 & \\ -\operatorname{div}(\boldsymbol{C} : e(\boldsymbol{u})) = 0 & \text{in } \Omega_s \\ \boldsymbol{u} = \boldsymbol{u}_0 & \text{on } \partial\Omega_s^D \\ (\boldsymbol{C} : e(\boldsymbol{u})) \cdot \boldsymbol{n}_s = \boldsymbol{g} & \text{on } \partial\Omega_s^N, \end{cases} \tag{3.18}$$

where $V_{\max}$ is the maximum allowable volume usage.

It is known that the mean compliance optimization problem governed by the equations of linear elasticity is self-adjoint, thus the design sensitivity $\overline{F}'$ can be expressed as follows:

$$\overline{F}' = -\left(e(\boldsymbol{u}) : \boldsymbol{C}_{\chi_\phi}\right) : e(\boldsymbol{u}) + \lambda_1, \tag{3.19}$$

where λ_1 is the Lagrange multiplier associated to the volume constraint. In this paper, the Lagrange multiplier λ_1 is updated by means of the Lagrangian multiplier method (Rockafellar, 1973; Hestenes, 1969; Powell, 1969), though the details are skipped here.

Minimization of power dissipation

The second optimum design problem focuses only on fluid mechanics. Considering a domain Ω_f, we intend to find the optimal channel layout which can minimize the total pressure drop, representing the energy loss in the flow channel, as illustrated in Fig. 3.4.

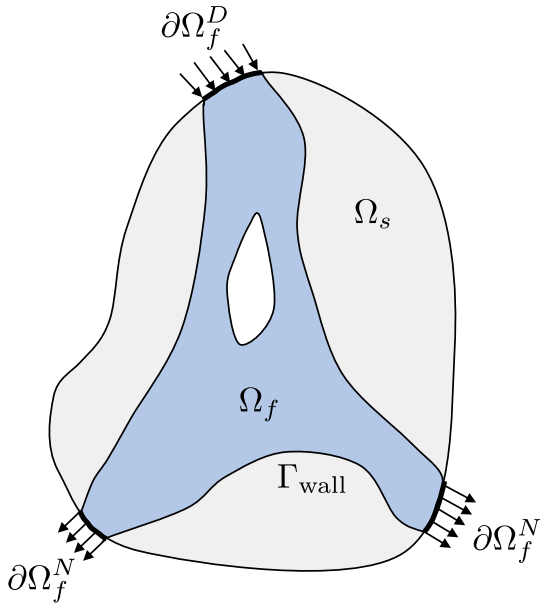


FIGURE 3.4: Schematic of the minimum power dissipation optimization problem.

Since the fluid flow distribution is the target to be optimized, we introduce a body force term to the Navier–Stokes equations. This body force represents the fictitious porous medium in the fluid flow. According to Darcy's law, this friction force is proportional to the velocity $\boldsymbol{v}$ with the proportionality α. Hence, it can be expressed as $-\alpha\boldsymbol{v}$, where α is the inverse permeability of a porous medium.

We then formulate the optimization mathematical model as follows:

$$\inf_{\chi_\phi \in \mathcal{X}} J(\Omega) = \int_{\partial\Omega_f^D \cup \partial\Omega_f^N} \left(p + \frac{1}{2} |\boldsymbol{v}|^2 \right) \left(-\boldsymbol{v} \cdot \boldsymbol{n}_f \right) d\Gamma \tag{3.20}$$

$$\text{s.t.} \begin{cases} G_1 = \dfrac{\int_D \chi_\phi d\Omega}{\int_D d\Omega} - V_{\max} \leq 0 \\ (\boldsymbol{v} \cdot \nabla)\, \boldsymbol{v} - \dfrac{1}{Re} \nabla \cdot \left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T \right) + \nabla p + \alpha \boldsymbol{v} = 0 \\ -\nabla \cdot \boldsymbol{v} = 0 \\ \boldsymbol{v} = \boldsymbol{v}_0 \qquad\qquad\qquad\qquad\qquad\qquad \text{on } \partial\Omega_f^D \\ \left(-p\mathbf{I} + \dfrac{1}{Re} \nabla \cdot \left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T \right) \right) \cdot \boldsymbol{n}_f = 0 \quad \text{on } \partial\Omega_f^N. \end{cases} \tag{3.21}$$

A simple interpolation scheme can be the linear interpolation as follows:

$$\alpha\left(\chi_\phi\right) = \alpha_{\max}\left(1 - \chi_\phi\right), \tag{3.22}$$

where $\alpha_{\max}$ is the upper bound of the effective inverse impermeability in the solid domains. Using this scheme, the design sensitivity can be derived as:

$$\overline{\boldsymbol{F}}' = -\alpha_{\max} \boldsymbol{v} \cdot \boldsymbol{v}_A + \lambda_1. \tag{3.23}$$

An alternative scheme is the nonlinear RAMP (Rational Approximation of Material Properties) scheme, given below:

$$\alpha\left(\chi_\phi\right) = \alpha_{\max} \frac{q_\alpha \left(1 - \chi_\phi\right)}{q_\alpha + \chi_\phi}, \tag{3.24}$$

where q_α is the penalty factor. Using the RAMP scheme, the design sensitivity can be derived as follows:

$$\overline{\boldsymbol{F}}' = -\frac{q\alpha_{\max}(q+1)}{\left(q + \chi_\phi\right)^2} \boldsymbol{v} \cdot \boldsymbol{v}_A + \lambda_1, \tag{3.25}$$

where $\boldsymbol{v}_A$ is the adjoint velocity which can be obtained by solving the following adjoint equation:

$$\begin{cases} -\boldsymbol{v} \cdot \nabla \boldsymbol{v}_A + \boldsymbol{v}_A \cdot \nabla \boldsymbol{v}^T - \dfrac{1}{Re} \nabla \cdot \left(\nabla \boldsymbol{v}_A + \nabla \boldsymbol{v}_A^T \right) \\ + \alpha \boldsymbol{v}_A + \nabla p_A = 0 \\ -\nabla \cdot \boldsymbol{v}_A = 0, \end{cases} \tag{3.26}$$

with the following adjoint boundary condition:

$$\begin{cases} -\left(\dfrac{1}{2}(\boldsymbol{v} \cdot \boldsymbol{v}) + p \right) \boldsymbol{n}_f - \boldsymbol{v}(\boldsymbol{v} \cdot \boldsymbol{n}_f) + \boldsymbol{v}_A(\boldsymbol{v} \cdot \boldsymbol{n}_f) \\ + \dfrac{1}{Re} \left(\nabla \boldsymbol{v}_A + \nabla \boldsymbol{v}_A^T \right) \cdot \boldsymbol{n}_f - p_A \boldsymbol{n}_f = 0 \qquad \text{on } \partial\Omega_f^N \\ \boldsymbol{v}_A = \boldsymbol{v}_0 \qquad\qquad\qquad\qquad\qquad\qquad\quad \text{on } \partial\Omega_f^D. \end{cases} \tag{3.27}$$

Using the FEA, the adjoint equation is expressed in the weak form as follows:

$$\begin{aligned}
&\int_{\Omega_f} -\tilde{\boldsymbol{v}}_A \cdot (\boldsymbol{v} \cdot \nabla \boldsymbol{v}_A) + \tilde{\boldsymbol{v}}_A \cdot \left(\boldsymbol{v}_A \cdot \nabla \boldsymbol{v}^T\right) \\
&+ \frac{1}{Re} \nabla \tilde{\boldsymbol{v}}_A : \nabla \boldsymbol{v}_A + \tilde{\boldsymbol{v}}_A \cdot \alpha \boldsymbol{v}_A - p_A \nabla \cdot \tilde{\boldsymbol{v}}_A d\Omega \\
&+ \int_{\partial\Omega_f^N} - \left(\frac{1}{2}(\boldsymbol{v} \cdot \boldsymbol{v}) + p\right)(\tilde{\boldsymbol{v}}_A \cdot \boldsymbol{n}_f) \\
&- (\tilde{\boldsymbol{v}}_A \cdot \boldsymbol{v})(\boldsymbol{v} \cdot \boldsymbol{n}_f) + (\tilde{\boldsymbol{v}}_A \cdot \boldsymbol{v}_A)(\boldsymbol{v} \cdot \boldsymbol{n}_f) d\Gamma \\
&+ \int_{\Omega_f} -\tilde{p}_A \nabla \cdot \boldsymbol{v}_A d\Omega = 0,
\end{aligned} \tag{3.28}$$

where $\tilde{\boldsymbol{v}}_A$ and $\tilde{p}_A$ are the test functions for the adjoint velocity and adjoint pressure, respectively.

Here, we clarify the choice of the interpolation scheme. Since the binary structure is used (Eq. (3.14)), the resulting inverse permeability does not depend on the interpolation scheme (linear or RAMP), and finally yields to:

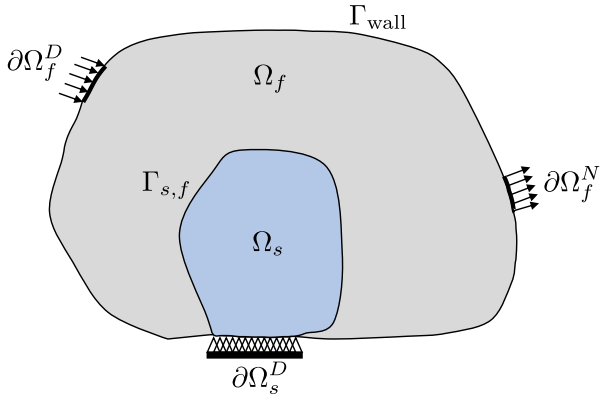
$$\alpha\left(\chi_\phi\right) = \begin{cases} 0, & \mathbf{x} \in \Omega_f \\ \alpha_{\max}, & \mathbf{x} \in \Omega_s. \end{cases} \tag{3.29}$$

This implies that the choice of the interpolation scheme does not affect the results of the forward analysis. On the other hand, from Eqs. (3.23) and (3.25), it can be found that the deduced sensitivities are different. In the authors experience, for the pure fluid optimization problems (i.e., minimal power dissipation problem), reasonable optimal solutions can be achieved using either the linear or RAMP scheme. However, for the thermal-fluid optimization problem, there are multiple material properties (i.e., inverse permeability, thermal conductivity, etc.) that need to be interpolated. The use of the linear scheme may cause the topology to converge towards a low quality local optimum rapidly at the very early stage of the optimization process. The nonlinear scheme will be necessary. This will be investigated in a future contribution, and will not be furthered in the present work.

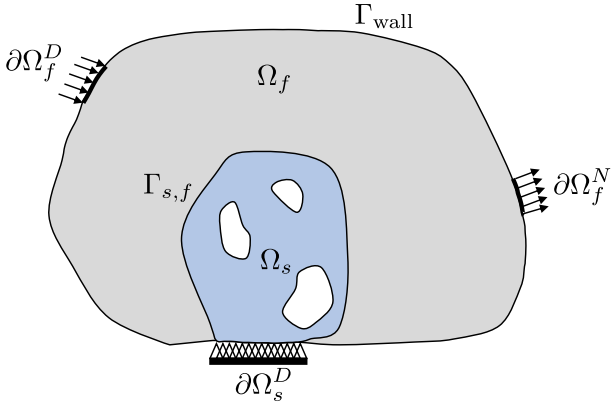
Fluid–structure interaction optimization

Our third optimum design problem targets the fluid–structure interaction (FSI) optimization. Fig. 3.5 illustrates two fundamental types of FSI optimization problems, that is, "dry" and "wet" FSI. The concept of "dry" and "wet" FSI was firstly introduced by Jenkins and Maute (2015). The former only allows the optimization of the internal structure (or infill structure) while the latter allows the manipulation of the geometry of the fluid–structure interface. This paper is the first attempt to solve both types of FSI optimization via the RDE-based TO method.

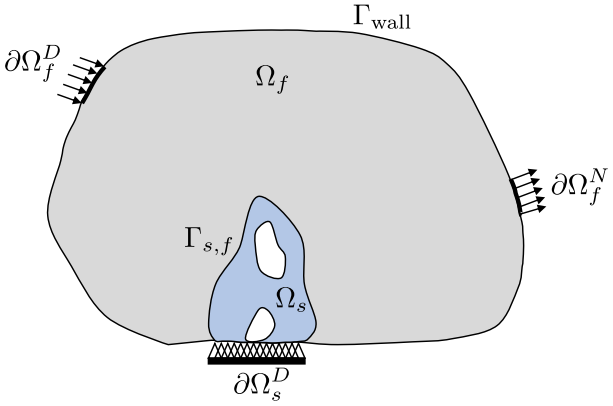
Considering a disjoint reunion domain $\Omega = \Omega_f \cup \Omega_s$, we intend to find an optimal supporting



(A) Initial shape.



(B) "Dry" FSI design.



(C) "Wet" FSI design.

FIGURE 3.5: Schematic of the fluid–structure interaction (FSI) optimization problem.

structure (internal supports for "dry" FSI, both internal and externally immersed supports for "wet" FSI).

Herein, the solid structure is the target to be optimized. Thus, similar to Section 3.2.3, in the solid domain Ω_s, the elasticity tensor $\boldsymbol{C}$ is expanded using Eq. (3.16). The compliance of the solid domain Ω_s is minimized under the flow load by Ω_f. Then, the optimization mathematical model is formulated as follows:

$$\begin{aligned} &\inf_{\chi_\phi \in \mathcal{X}} J(\Omega) = \int_{\Gamma_{s,f}} \left(\sigma_s\left(\boldsymbol{u}\right) \cdot \boldsymbol{n}_s\right) \cdot \boldsymbol{u}\, d\Gamma. \\ &\text{s.t.} \quad G_1 = \frac{\int_D \chi_\phi d\Omega}{\int_D d\Omega} - V_{\max} \leq 0. \end{aligned} \tag{3.30}$$

As discussed in Section 3.2.1, the FSI problem considered in the present work is weakly coupled (or one-way-coupled), which implies that only the fluid load is applied on structures but not vice versa. Thus, the fluid–structure system can be evaluated by solving the weak form of the Navier–Stokes equations *without body force* using Eqs. (3.4) to (3.6), followed by solving the Lamé using Eq. (3.11).

Next, we elaborate the design sensitivity. For those problems with high pressure and shear stress (i.e., turbulent flow), it may be not easy to obtain a physically reasonable optimal solution without considering the pressure sensitivity at the fluid–solid interface. In that case, it is necessary to interpolate the fluid stress tensor ($\sigma_s\left(\boldsymbol{u}\right) \cdot \boldsymbol{n}_s = -\chi_\phi \sigma_f\left(\boldsymbol{v}, \boldsymbol{p}\right) \cdot \boldsymbol{n}_f$). However, for the potential flow problem (see Picelli, Neofytou, and Kim (2019) or Kumar, Frouws, and Langelaar (2020)), or for the low Reynolds incompressible laminar flow (i.e., $Re \leq 100$ in the present work), the effect of this pressure sensitivity at the interface is quite trivial. Therefore, we do not interpolate the fluid stress tensor ($\sigma_s\left(\boldsymbol{u}\right) \cdot \boldsymbol{n}_s = -\sigma_f\left(\boldsymbol{v}, \boldsymbol{p}\right) \cdot \boldsymbol{n}_f$), and treat this weakly coupled problem to be a self-adjoint optimization problem. A high pressure and shear stress FSI problem is targeted in a future contribution. Then the design sensitivity is the same as for the compliance problem Eq. (3.19).

3.3 Implementation details

In this section, we discuss the numerical implementation of the complete product development workflow, as shown in Fig. 3.6. First, we use the commercial CAD software Rhinoceros (McNeel, 2010) to build the design model. Then, an open-source software Gmsh (Geuzaine and Remacle, 2009) is employed to generate the mesh for the initial step of the topology optimization. After pre-processing, we start the topology optimization. We use FreeFEM (Hecht, 2012; Jolivet, Dolean, Hecht, Nataf, Prud'homme, and Spillane, 2012) for finite element analysis (FEA), PETSc (Abhyankar, Brown, Constantinescu, Ghosh, Smith, and Zhang, 2018; Balay, Gropp, McInnes, and Smith, 1997; Balay, Abhyankar, Adams, Benson, Brown, Brune, Buschelman, Constantinescu, Dalcin, Dener, Eijkhout, Gropp, Hapla, Isaac, Jolivet, Karpeev, Kaushik, Knepley, Kong, Kruger, May, McInnes, Mills, Mitchell, Munson, Roman, Rupp, Sanan, Sarich, Smith, Zampini, Zhang, Zhang,

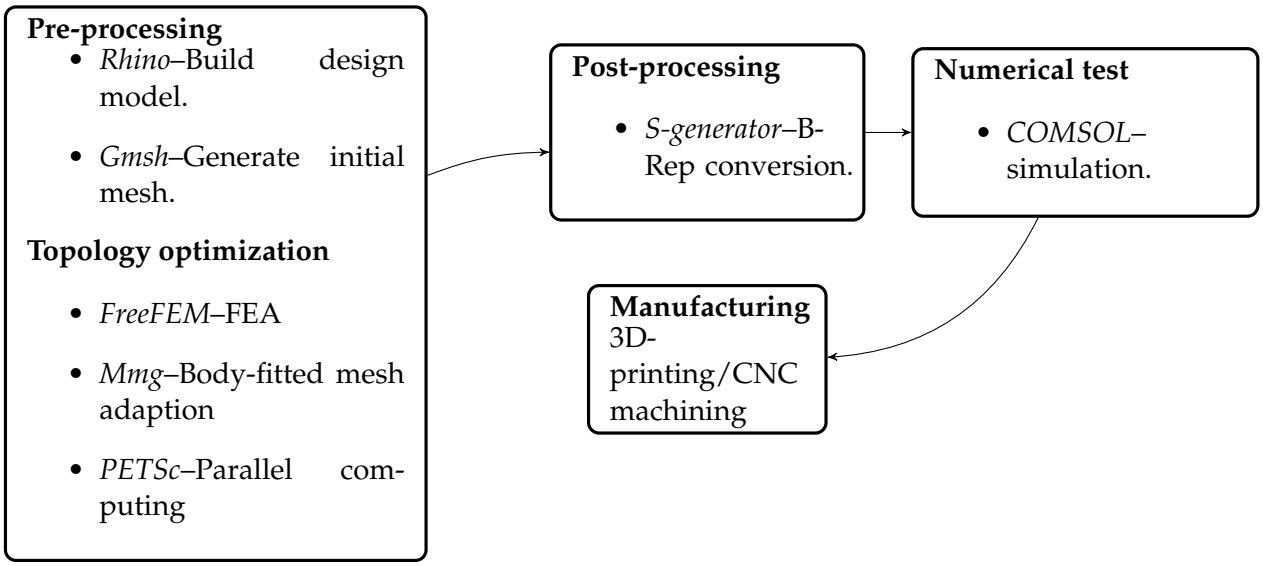


FIGURE 3.6: The complete workflow of the product development based on topology optimization.

and Zhang, 2021; Balay, Abhyankar, Adams, Brown, Brune, Buschelman, Dalcin, Dener, Eijkhout, Gropp, Karpeyev, Kaushik, Knepley, May, McInnes, Mills, Munson, Rupp, Sanan, Smith, Zampini, Zhang, and Zhang, 2019) for distributed linear algebra, and Mmg (Dapogny, Dobrzynski, and Frey, 2014) for body-fitted mesh adaption. After obtaining the generative design, we move to the post-processing stage, and we convert the mesh representation to CAD data (B-Rep conversion) using the commercial software package S-generator (QUINT, 2021). Finally, the numerical test can be conducted using the commercial FEA software COMSOL Multiphysics (COMSOL, 2018).

3.3.1 Parallel computing

As has been explained in Section 2.3.1, FreeFEM (Hecht, 2012; Jolivet, Dolean, Hecht, Nataf, Prud'homme, and Spillane, 2012) is used for the discretization of PDEs while PETSc (Abhyankar, Brown, Constantinescu, Ghosh, Smith, and Zhang, 2018; Balay, Gropp, McInnes, and Smith, 1997; Balay, Abhyankar, Adams, Benson, Brown, Brune, Buschelman, Constantinescu, Dalcin, Dener, Eijkhout, Gropp, Hapla, Isaac, Jolivet, Karpeev, Kaushik, Knepley, Kong, Kruger, May, McInnes, Mills, Mitchell, Munson, Roman, Rupp, Sanan, Sarich, Smith, Zampini, Zhang, Zhang, and Zhang, 2021; Balay, Abhyankar, Adams, Brown, Brune, Buschelman, Dalcin, Dener, Eijkhout, Gropp, Karpeyev, Kaushik, Knepley, May, McInnes, Mills, Munson, Rupp, Sanan, Smith, Zampini, Zhang, and Zhang, 2019) is used for the linear algebra backend.

To solve the Navier-Stokes equations, a modified augmented Lagrangian preconditioner proposed by Moulin, Jolivet, and Marquet (2019) is adopted to solve the linearized fluid flow governed by Eqs. (3.4) to (3.6). The aforementioned preconditioners ensure that the physical computation part of our TO framework is scalable with respect to the number of processes and the

problem size.

In this chapter, all the two-dimensional cases are performed on a laptop equipped with an Intel Core i7 with a clock frequency of 2.8 GHz having four cores and 16 GB of memory. The three-dimensional cases are performed on a Linux workstation with dual Intel(R) Xeon(R) Gold 6246R with a clock frequency of 3.40 GHz, having 32 cores in total and 128 GB of memory. We use GNU compilers and OpenMPI.

3.3.2 Body-fitted mesh adaption technique

In this chapter, we use the body-fitted mesh adaption method proposed by Allaire, Dapogny, and Frey (2013). Our numerical implementation of this mesh evolution algorithm relies on an open-source library Mmg (Dapogny, Dobrzynski, and Frey, 2014), see the details of this workflow in Section 2.3.2.

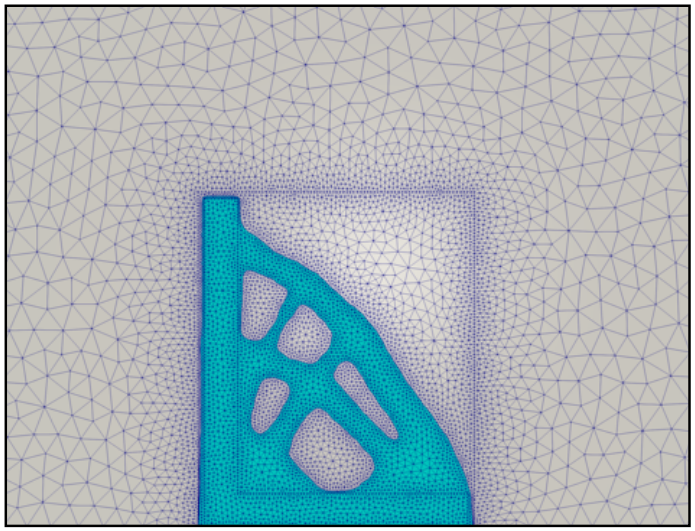


FIGURE 3.7: A snapshot of the body-fitted adaptive mesh during the "wet" FSI optimization iteration. (For interpretation of the references to color in the figure legend in the present work, the reader is referred to the web version of this paper.)

Using the above-mentioned body-fitted mesh adaption algorithm, the Mmg package allows us to control the following mesh parameters: (1) the maximum and minimum edge sizes in the whole computational domain, $l_{e_{\text{global, max}}}$ and $l_{e_{\text{global, min}}}$; and (2) h_{grad} allows the setting of the gradation value which controls the ratio between two adjacent edges. With a gradation of h_{grad}, two adjacent edges e_1 and e_2 must satisfy: (1) $\frac{1}{h_{\text{grad}}} \leq \frac{l_{e_1}}{l_{e_2}} \leq h_{\text{grad}}$; and (2) the maximum and minimum edge size alongside the zero level-set interface $l_{e_{\text{local, max}}}$ and $l_{e_{\text{local, min}}}$, respectively. Note that the local mesh parameters overwrite the global parameters. For example, Fig. 3.7 shows a snapshot of the mesh view during the "wet" FSI optimization iteration. It can be clearly observed that: (1) the

mesh is refined around the fluid–solid boundary, which results in a high-resolution boundary; (2) relatively coarser mesh is distributed at the far field, which can save the overall computational cost.

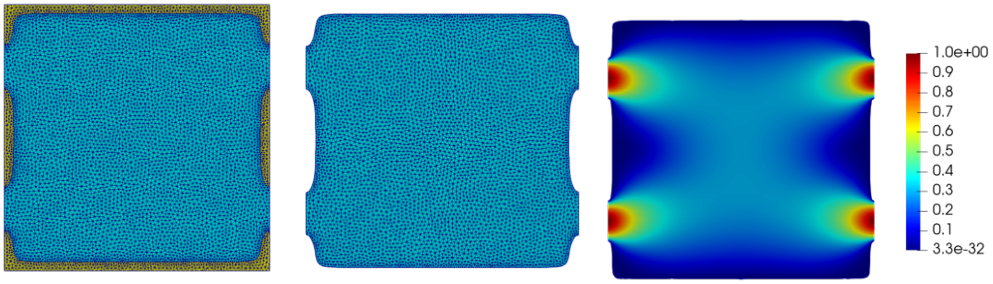
Moreover, the use of body-fitted mesh allows the extraction of the fluid domain and solid domain directly using the function *trunc* which is built in FreeFEM. This allows us to apply the "separate modelling" strategy. We take the minimal power dissipation problem as an example. The workflow is illustrated in Fig. 3.8. First, the forward problem and the adjoint problem (if any) are solved on $\mathcal{T}_{\text{fluid},it}$, as shown in Fig. 3.8c. Next, the topological sensitivity $\overline{F}'$ is computed on $\mathcal{T}_{it}$, as shown in Fig. 3.8d. The level-set is then updated on $\mathcal{T}_{it}$, as shown in Fig. 3.8e. Due to the effect of the Lagrange multiplier associated with the volume constraint, the new islands (solid phase) can be nucleated. After that, the mesh is evolved based on the newly updated level-set field and is body-fitted to the zero level-set interface, as shown in Fig. 3.8f. Finally, $\mathcal{T}_{\text{fluid},it+1}$ is extracted for the forward analysis for the next iteration, as shown in Fig. 3.8g.

In short, using the separate modelling strategy, the interpolation model is only to aid the sensitivity analysis, and can be excluded from the forward analysis. Therefore, the values of $\alpha_{\max}$ and the penalty factor q_α can be chosen quite freely[1]. For example, in this work, we set $\alpha_{\max} = 1.0$, $q_\alpha = 1.0$.

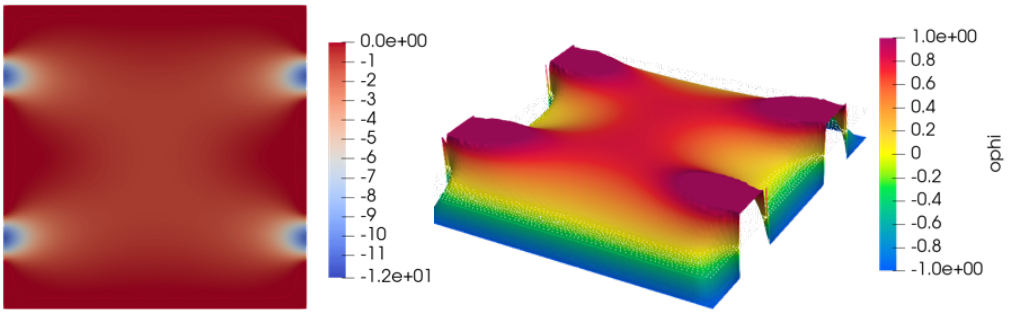
It should be noted that the traditional "hybrid modelling" strategy[2] could be an alternative option in the proposed framework. This can be realized by actuating the fictitious body force term in the governing equations and the adjoint equations (if any). In that case, the overall algorithm implementation can be simplified to a great extent, and it might be more preferred in the thermal-fluid coupled optimization problem because the unknowns in both solid and fluid phases can be computed using the same system of equations. In addition, we can use not only the body-fitted mesh but also the anisotropic/isotropic mesh for the mesh refinement. It will be worthy making comparison between these two strategies in the fluid–solid phase changing modelling. On the other hand, when using the "hybrid" modelling strategy, the values of $\alpha_{\max}$ and q (if RAMP scheme is employed) need to be selected carefully. $\alpha_{\max}$ should be large enough to prevent the fluid flow from "penetrating" the fluid–solid boundary, but should not be too large to cause the numerical difficulties during the physical computations. The particular value of the $\alpha_{\max}$ does highly depend on reference length scale. Besides, the value of the penalty factor q may influence the optimal solution to a great extent in the thermal-fluid optimization problem. It is not easy to find a general guideline from the literature. Thus it would be necessary to perform extensive numerical experiments before we can decide these values. However, it is always the case when using the "hybrid modelling" strategy, which deserves further studies and will not be here, see

[1] An extensive numerical experiment has been performed to test different values of the parameters: $\alpha_{\max} = \{10^0, 10^1, 10^3, 10^5, 10^7\}$, $q_\alpha = \{10^2, 10^1, 10^0, 10^{-1}, 10^{-2}\}$. The test results show that the optimal solutions are practically identical.

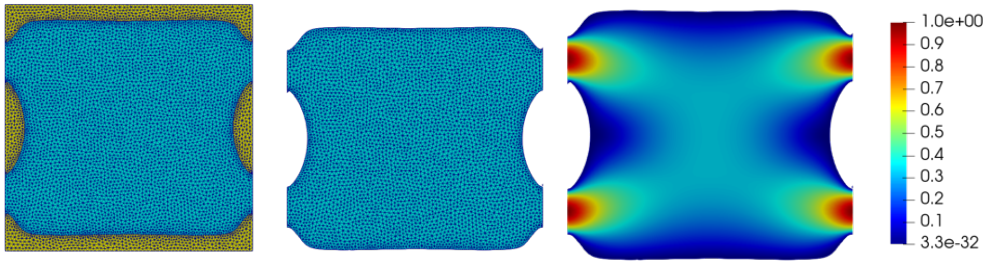
[2] Hybrid modelling: The fluid equations are solved in the whole computational domain. The fictitious body force term needs to be included in the analysis to mimic the presence or absence of the solid phase (resp. fluid phase).



(A) $\mathcal{T}_{it}$. (B) $\mathcal{T}_{\text{fluid},it}$. (C) FEA on $\mathcal{T}_{\text{fluid},it}$.



(D) Compute $\overline{\boldsymbol{F}}'$ on $\mathcal{T}_{it}$. (E) Update Level-set on $\mathcal{T}_{it}$.



(F) $\mathcal{T}_{it+1}$. (G) $\mathcal{T}_{\text{fluid},it+1}$. (H) FEA on $\mathcal{T}_{\text{fluid},it+1}$.

FIGURE 3.8: Schematic of the separate modelling strategy.

Chapter 5.

3.3.3 Post-processing strategy

As discussed in Section 3.3.2, in this work, the optimal solution (or generative design) is sampled with the volumetric representation of body-fitted tetrahedral meshes based on a zero level-set. In real-world application, to manufacture the generative design, the mesh representation should be converted to the editable spline representation. This post-processing stage is called B-Rep conversion, which is an important increment to Chapter 2.

However, it is known that B-Rep conversion inevitably introduces a loss of surface precision due to the limitations of the target representation, as shown in Fig. 3.9. It should be noted that this loss is negligible in the physical computation of the topology optimization. However, when integrating the optimized part in the whole assembly, the loss of the surface accuracy (or surface residual) may lead to manufacturing deficiency such as part interference (Marinov, Amagliani, Barback, Flower, Barley, Furuta, Charrot, Henley, Santhanam, Finnigan, et al., 2019).

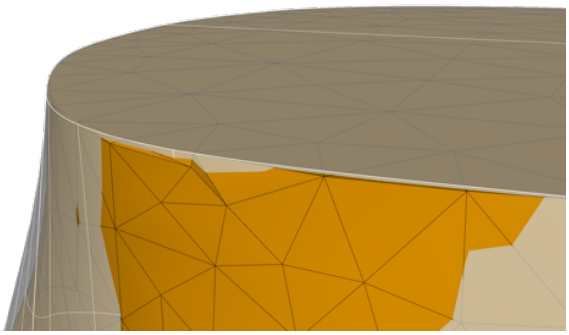
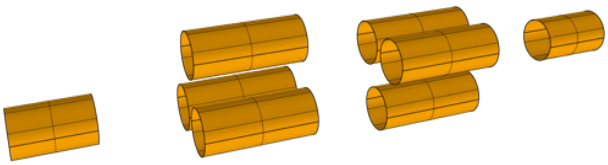


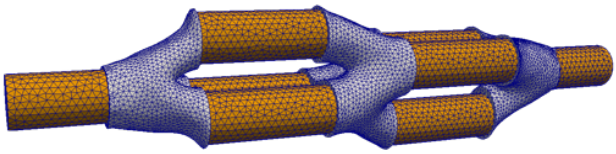
FIGURE 3.9: A fragment of a mesh extracted from our generative design.

Frankly, reviewing the details of the B-Rep conversion algorithm is beyond the scope of this paper. Herein, we implement the conversion using the commercial software S-generator (QUINT, 2021). Here, we use a pipe connector design as a test case for showing our generative design conversion in brief. From Fig. 3.10, the yellow pipes are the fixed design domain, as shown in Fig. 3.10a. Then, we perform our topology optimization algorithm to obtain the generative design, as shown in Fig. 3.10b. Next, we reconstruct a B-Rep model using an input mesh as a geometry reference. In this test case, the yellow pipes should be used as a reference so that the output pipes will features the same geometrical parameters as the input CAD data. After that, we perform the B-Rep conversion to smoothen surfaces with the processed mesh geometry, and end up with the output B-Rep composed from the input and organic parts shown in Fig. 3.10d. Finally, we extract the editable CAD data of the optimized pipe connector which can be used for manufacturing.

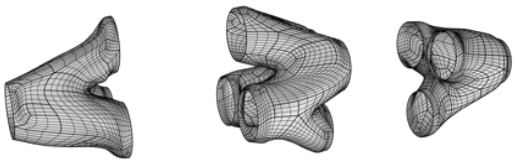
In short, the present B-Rep conversion strategy allows us to achieve the following:



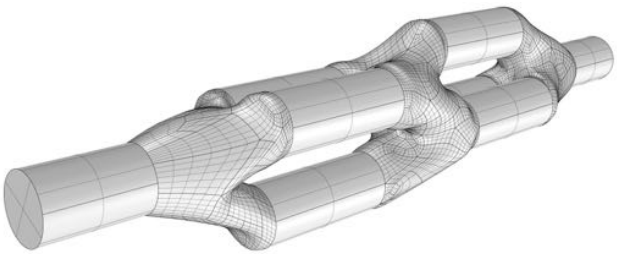
(A) Input CAD data of pipes.



(B) Body-fitted mesh representation of the optimal solution.



(C) Editable surfaces constructed for the organic part.



(D) Output CAD data for numerical experiment.

FIGURE 3.10: Our generative design conversion using S-generator (QUINT, 2021).

1. assemble the exact input solid boundary surfaces where the organic part (optimized structure) coincides with the input;
2. smoothen everywhere else the boundary with editable surfaces;
3. join the surfaces together to output the CAD data for manufacturing and numerical experiment.

Note that in our post-processing stage, it may still require some manual efforts to smoothen surfaces with the processed mesh geometry. As a matter of fact, it is not easy to quantify the B-Reps conversion timing since all the actions are conducted via a GUI environment. The step that required the most manual efforts was to detect the unclosed surface (if any) and to stitch them up. The action timing depends on the geometrical complexity of the generative designs. In the authors experience, for an experienced user, the "detection–stitching" process usually takes a few minutes for each 3D generative design presented in this paper. In our future work, we intend to employ a fully automatic method for the conversion of our generative design to a watertight B-Rep composed of the exact residual geometry of the input solids and editable surfaces.

3.3.4 Optimization flow chart

As discussed above, we treat three different optimum design problems. Here, we illustrate the optimization algorithm for a weakly-coupled "wet" FSI optimization problem, which is the most complicated one in the present work, as summarized in Algorithm 1.

Algorithm 1 "Wet" FSI topology optimization.

Input: global mesh $\mathcal{T}$
initialization:
decompose the computational domain for parallel computing
initialize the level-set field ϕ
for $it = 0; it < Maxloop; it = it + 1$ **do**
solve Navier–Stokes equations on $\mathcal{T}_{\text{fluid}}$ using the Newton method, cf. Eqs. (3.4) to (3.6)
calculate fluid stress tensor σ_f, cf. Eq. (3.3)
solve Lamé equations on $\mathcal{T}_{\text{solid}}$, cf. Eq. (3.11)
calculate objective functional
if $\|w_i\| < \varepsilon$ **then**
break
else
calculate sensitivity and Lagrange multipliers
update the level-set function ϕ_{it} by solving RDE on $\mathcal{T}$, cf. Eq. (2.36)
update the characteristic function $\chi_{\phi_{it}}$
reduce solution of ϕ_{it} to a single process
call Mmg to obtain $\mathcal{T}_{it+1}$ based on ϕ_{it}
truncate $\mathcal{T}_{it+1} \to \mathcal{T}_{\text{fluid},it+1} + \mathcal{T}_{\text{solid},it+1}$
broadcast to all processes
interpolate ϕ_{it} to $\mathcal{T}_{it+1}$; $\chi_{\phi_{it}}$ to $\mathcal{T}_{\text{solid},it+1}$
end if
end for

The initialization is described as follows: the input mesh $\mathcal{T}$ is generated by Gmsh and stored in the Inria Medit format. The fluid and solid subdomains are labeled with different numbering. By using the built-in function *trunc* in FreeFEM, the global mesh can be divided into two separate computational domains $\mathcal{T}_{\text{fluid}}$ and $\mathcal{T}_{\text{solid}}$. After that, the computational domains $\mathcal{T}$, $\mathcal{T}_{\text{fluid}}$, and $\mathcal{T}_{\text{solid}}$ are decomposed using a standard mesh partitioner such as METIS. Next, the initial level-set field (initial guess) is set.

Then, the optimization loop begins. First, we solve Stokes flow on $\mathcal{T}_{\text{fluid}}$. The obtained velocity and pressure fields act as the initial guess for the Navier–Stokes equations. After solving the Navier–Stokes equations, the fluid stress tensor σ_f is computed and the Lamé equation can be solved on $\mathcal{T}_{\text{solid}}$. Then, the objective functional is computed. If the objective functional is converged, the optimization ends, otherwise the sensitivity and the Lagrange multipliers are computed. After that the level-set function ϕ_{it} is updated by solving the RDE on $\mathcal{T}$, after which ϕ is bounded to satisfy $\phi(x) \in [-1, 1]$, followed by updating the characteristic function $\chi_{\phi_{it}}$ with the step function. Next, we generate a global level-set function by performing a standard MPI reduction on process 0. Then, the global mesh gathered on process 0 can be updated sequentially based on the level-set function by calling the Mmg package. Now we obtain a new mesh $\mathcal{T}_{it+1}$ which includes the new fluid and solid subdomains divided by the zero level-set interface. After being truncated to $\mathcal{T}_{\text{fluid},it+1}$ and $\mathcal{T}_{\text{solid},it+1}$, the work load is decomposed again. Finally, the current step solutions are interpolated to the updated meshes accordingly. The workload is then distributed

again, using MPI, until the end of the iteration where either of the following criteria are satisfied: (1) the relative difference between the objective values in current and previous steps is lower than 10^{-4} for 4 steps in a sequence; or (2) the maximum iteration number is reached[3].

3.4 Design examples

In this section, we present several numerical examples to the design problems illustrated in Section 3.2.3. The first two problems treat only the single physics, which are used to demonstrate the workflow shown in Fig. 3.6 before passing to more complicated multi-physics problems. To validate our methodology, for each design problem, we provide a benchmark example from the literature.

In Section 3.4.1, we consider the optimization of an architectural structure subject to a surface traction load governed by the Lamé equation. The three-dimensional design model is inspired by the test case of Ameba–a topology optimization tool based on the BESO method (Zhou, Shen, Wang, Zhou, and Xie, 2018). The physics at play is the least complicated to solve in this paper since it only involves linear elasticity. However, it clearly shows the efficiency of our level-set-based TO method integrated with the body-fitted adaptive mesh strategy.

Section 3.4.2 examines the validity of our methodology to an optimal problem in pure fluid mechanics. We consider a very classical pipe design problem to minimize the power dissipation in the fluid flow. In Section 3.4.2, we show a two-dimensional benchmark double channel example in many previous contributions (Sá, Amigo, Novotny, and Silva, 2016; Yaji, Yamada, Yoshino, Matsumoto, Izui, and Nishiwaki, 2014; Pingen and Maute, 2010; Borrvall and Petersson, 2003; Challis and Guest, 2009; Souza, Yamabe, Sá, Ranjbarzadeh, Picelli, and Silva, 2021), after that a more practical three-dimensional pipe connector example is shown in Section 3.4.2. As pointed out by Feppon et al. Feppon, Allaire, Dapogny, and Jolivet, 2020b, industrial contexts often feature very high Reynolds numbers in the turbulent regime, and even a slight change in the shape can result in a dramatic change in the flow performance. Thus, the shape to be optimized is often parameterized by a number of parameters. By contrast, in this work, we present the application of our RDE-based TO method, which allows finding the optimal solution of the fluid mechanics problem governed by the Navier–Stokes equation considering moderate Reynolds numbers in the laminar regime, without any predefined hole in the initial guess.

Section 3.4.3 involves a coupled physics problem–"dry" FSI optimization. In Section 3.4.3, we reassemble the two-dimensional beam example from Jenkins and Maute (2015). Note that in their work, a strongly-coupled condition is taken into account and the numerical analysis is based

[3]In this work, for most of the test cases, the topological configurations can reach stable and remain unchanged within 150 iterative steps. The maximum iteration number, for most of them, is therefore set to less than or equal to 150. For each test case, the reader is referred to the iterative histories of the objective value and the constraints plotted in Section 3.4.

on the extended finite element method (XFEM). By contrast, in the present work, we assume a weakly-coupled condition. The standard finite element analysis is adopted for the numerical computation. In Section 3.4.3, we further extend to a three-dimensional problem. A sphere-like structure is standing under the fluid flow. The infill support is optimized to resist the deformation.

Section 3.4.4 shows the test cases for the "wet" FSI optimization as described in Algorithm 1. As illustrated above, this problem is to find the supporting structure immersed in the fluid flow in order to make the whole structure as stiff as possible. In Section 3.4.4, we reassemble a two-dimensional wall example from Picelli, Ranjbarzadeh, Sivapuram, Gioria, and Silva (2020). Finally, a three-dimensional "solar plate" support design can be found in Section 3.4.4.

It should be noted that many of the results in this work are still within the academic scope, since there remains a huge improvement space to meet the industrial standards. (1) the body-fitted mesh adaptation is performed sequentially on a single MPI process. It is still not possible to handle with an ultra-high resolution TO problem. (2) For some of the numerical examples, the convergence cannot be fully reached at the last iteration. We can observe the oscillation of the objective values, though the topology does not change anymore. This behavior is due to the fact that the mesh adaptation allows us to generate some small-scale changes during the mesh revolution which can make a definite impact on the objective value. This effect can be more significant when dealing with three-dimensional nonlinear problems. In any case, it is an important issue which deserves further investigations. (3) Those problems with high pressure and shear stress (i.e., turbulent flows) and strongly-coupled conditions are also worth investigating in future work.

3.4.1 Minimization of compliance

In this subsection, we start by reassembling an interesting architectural design found in the test cases of Ameba (XIE-Engineering-Technologies., 2021). The optimization mathematical model is given in Section 3.2.3. The design model is created in Rhinoceros, as shown in Fig. 3.11. The computational domain has a dimension of $\approx 0.5L \times L \times L$ with a curved top surface. Inside the domain, there is a non-design domain with the shape of capital letter "A" (in dark blue). The bottom surface of this letter "A" is fixed on the ground, $\boldsymbol{u} = [0,0,0]^T$. The uniformly distributed surface traction is applied downward on the top surface, $\boldsymbol{g} = [0,0,-1]^T$. The regularization parameter τ is set to 10^{-4} and the maximum allowed volume fraction is 10%.

The iterative history is shown in Fig. 3.12. These plots clearly show the body-fitted adaptive mesh evolution used in our topology optimization. First, the solid material is largely removed from the bottom half of the design domain. Then, the supporting structure tends to bifurcate into multiple levels of branches. With the removal of the solid material, the non-design domain–capital letter "A" is gradually unfolded to us. The iterative histories of the objective value and volume fraction are plotted in Fig. 3.13.

As for this test case, the number of tetrahedra for the last iteration is $2.58 \cdot 10^6$. This includes $4.57 \cdot 10^5$ nodes, thus approximately 1.3 million degrees of freedom for the linearized elasticity system.

Next, the optimal solution sampled with tetrahedral meshes (see Fig. 3.12d) is converted to the CAD data (see Fig. 3.14a). As discussed above, this conversion inevitably introduces a loss of surface precision. However, the use of body-fitted adaptive mesh can reduce such loss to a lower level compared with the fixed-mesh-based generative design. Then, the CAD data is imported into COMSOL Multiphysics and re-meshed using the software built-in mesh tool, as shown in Fig. 3.14b. The Von Mises stress distribution is depicted in Fig. 3.14c, in which we can observe the stress mainly concentrated at the connecting part between the end of branches and the top roof as well as around the bottom surfaces. The objective values obtained by the FreeFEM optimization solver and the COMSOL simulation solver are summarized in Table 3.1. Note that in the COMSOL solver, the strain energy is computed using the built-in function "solid.Ws".

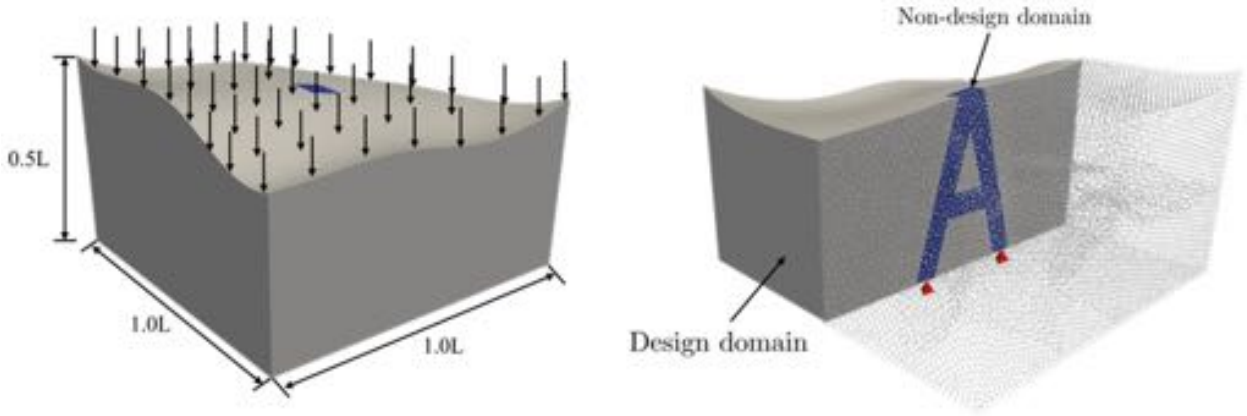


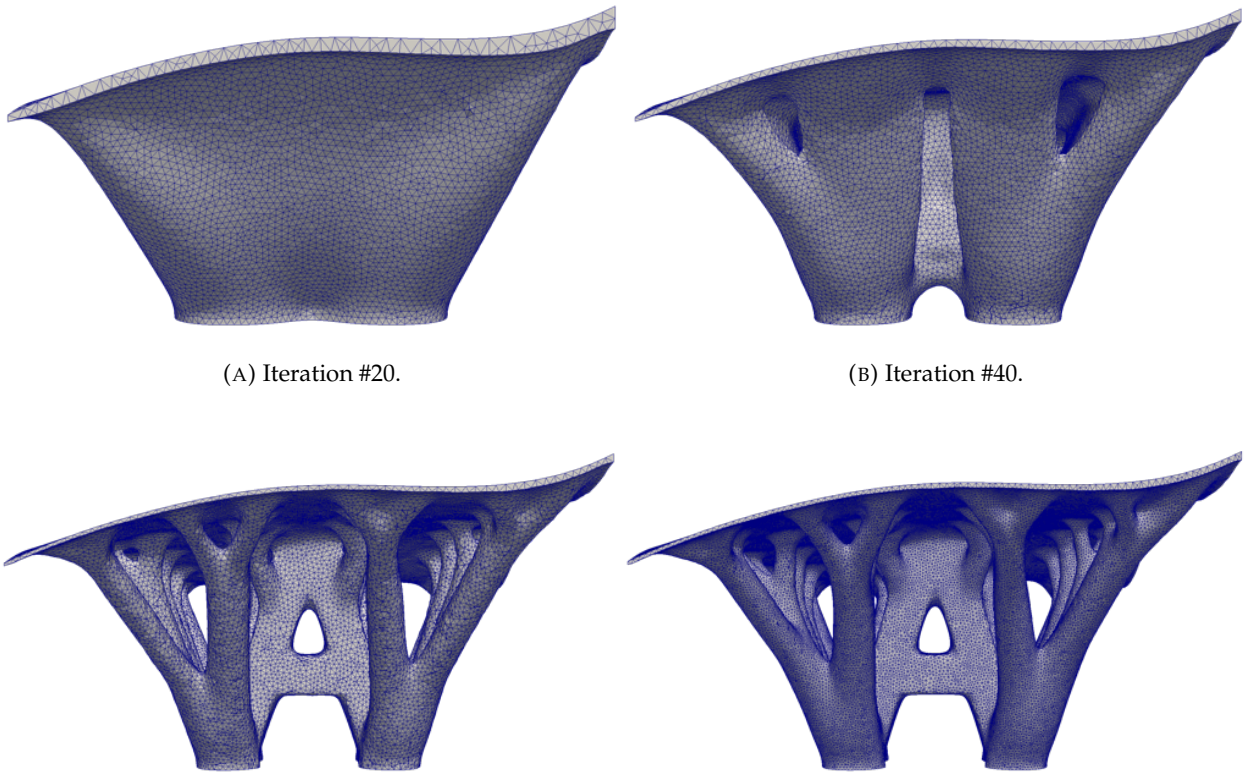
FIGURE 3.11: Design model of the 3D roof support.

TABLE 3.1: Comparison of the objective value of the 3D roof support design, cf. Section 3.4.1.

Case	FreeFEM optimization solver	COMSOL simulation solver
Initial design	0.25	0.25
Generative design	1.15	1.13

3.4.2 Minimization of fluid power dissipation

This subsection examines the optimum design problem for the power dissipation in the field of fluid mechanics. The power dissipation is the energy loss when the fluid flows through the channel, formulated as the inlet and outlet pressure drop. The optimization mathematical model is given in Section 3.2.3.



(A) Iteration #20.

(B) Iteration #40.

(C) Iteration #70.

(D) Iteration #120.

FIGURE 3.12: Iterative history of the roof support design, cf. Section 3.4.1.

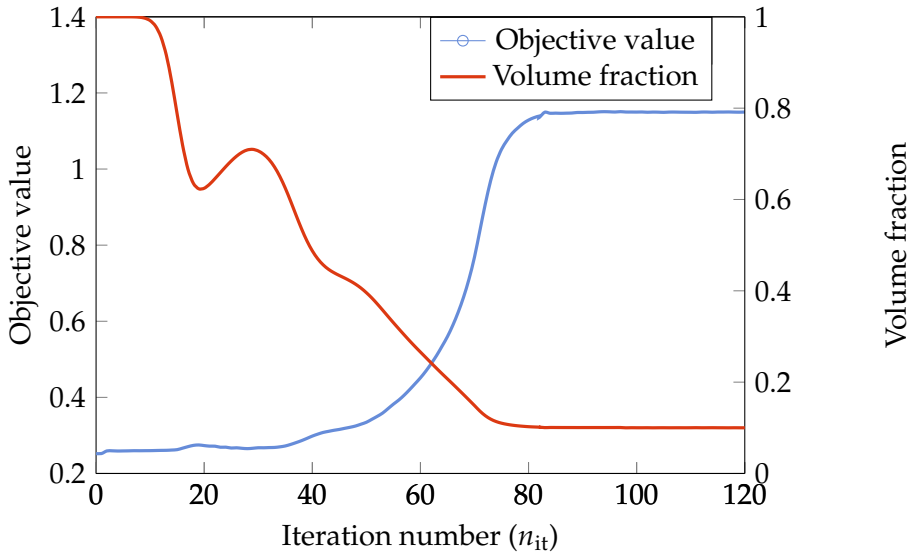
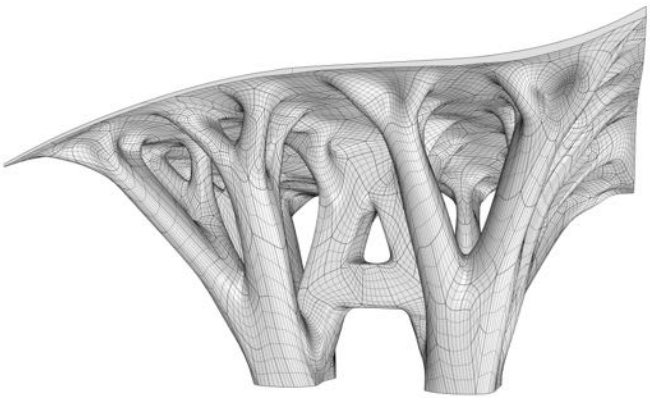
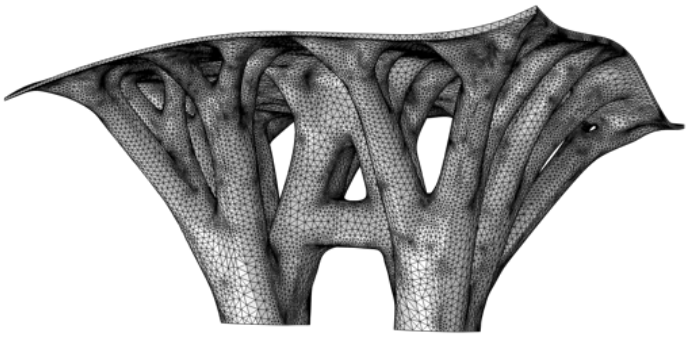


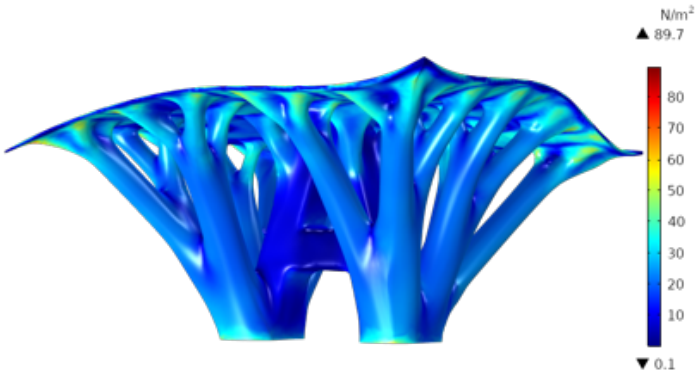
FIGURE 3.13: Histories of the objective value and the volume fraction for the 3D roof support design, cf. Fig. 3.12.



(A) B-Rep conversion.



(B) Remeshing in COMSOL.



(C) Von Mises stress distribution.

FIGURE 3.14: Post-processing and simulation of the generative design for the roof supporting structure.

2D benchmark problem

Now we test our design methodology by reproducing a classical 2D double channel design problem which has been used as a benchmark in many previous contributions (Sá, Amigo, Novotny, and Silva, 2016; Yaji, Yamada, Yoshino, Matsumoto, Izui, and Nishiwaki, 2014; Pingen and Maute, 2010; Borrvall and Petersson, 2003; Challis and Guest, 2009; Souza, Yamabe, Sá, Ranjbarzadeh, Picelli, and Silva, 2021). The design model is shown in Fig. 3.15. The height of the computational domain is set to be the reference length $1.0L$, and the width is set to $W = \{1.0L, 1.5L\}$. The fluid enters the domain from two inlets on the left-hand side with a parabolic profile $\boldsymbol{v}_x = -144(y - \frac{1}{12}L)(y + \frac{1}{12}L)$ and flows out via the two outlets on the right-hand side with a zero normal stress boundary condition $\boldsymbol{\sigma}_f \cdot \boldsymbol{n}_f = 0$. The aim of this benchmark is to examine the adequacy of the proposed framework by comparing the obtained results with those in the literature. The value of τ is set to 10^{-3} and the maximum allowed volume fraction is 33% which is equal to one-third of the total volume of the design domain.

First, we run our algorithm to solve the classical Stokes flow problem. The optimal solutions are shown in Fig. 3.16. For the case where $W = 1.0L$, the pipe remains almost straight between the in-/outlets, while for the case where $W = 1.5L$, we can observe a confluence of the inflows. The snapshots of the topology evolution are shown in Fig. 3.17. The iteration histories of the objective value and volume fraction are plotted in Fig. 3.18. Next, we compare the obtained results with those from the literature (Borrvall and Petersson, 2003; Challis and Guest, 2009; Souza, Yamabe, Sá, Ranjbarzadeh, Picelli, and Silva, 2021). Let us recall the formulation of the objective functional. The boundary integral given in Eq. (3.20) can be rewritten as the domain integral, as follows:

$$\inf_{\chi_\phi \in \mathcal{X}} J(\Omega) = \int_{\partial\Omega_f^D \cup \partial\Omega_f^N} \left(p + \frac{1}{2}|\boldsymbol{v}|^2\right)(-\boldsymbol{v} \cdot \boldsymbol{n}_f)\, d\Gamma = \int_D \mu \nabla \boldsymbol{v} : \left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T\right) d\Omega. \tag{3.31}$$

Note that the objective functional used by Borrvall and Petersson (2003) and Challis and Guest (2009) is given as follows:

$$\inf_{\chi_\phi \in \mathcal{X}} J(\Omega) = \frac{\mu}{2} \int_D \nabla \boldsymbol{v} : \nabla \boldsymbol{v} d\Omega. \tag{3.32}$$

Here, we show that the boundary and domain integrals are equivalent. The Navier–Stokes equations are

$$\int_\Omega \left((\boldsymbol{v} \cdot \nabla)\, \boldsymbol{v}\right) \cdot \boldsymbol{v} + \nabla p \cdot \boldsymbol{v} - \frac{1}{\mathrm{Re}} \left(\nabla \cdot \left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^{\mathrm{T}}\right)\right) \cdot \boldsymbol{v} + \alpha \boldsymbol{v} \cdot \boldsymbol{v} d\Omega = 0. \tag{3.33}$$

The first term in Eq. (3.33), $\int_\Omega \left((\boldsymbol{v} \cdot \nabla)\, \boldsymbol{v}\right) \cdot \boldsymbol{v} d\Omega$ can be rewritten as:

$$\int_\Omega \left((\boldsymbol{v} \cdot \nabla)\boldsymbol{v}\right) \cdot \boldsymbol{v} d\Omega = \int_\Gamma \frac{1}{2}(\boldsymbol{v} \cdot \boldsymbol{v})\boldsymbol{v} \cdot \boldsymbol{n} d\Gamma. \tag{3.34}$$

Eq. (3.34) can be proved as follows:

$$\begin{aligned}
&\nabla \cdot \left[(\boldsymbol{v} \otimes \boldsymbol{v})^{\mathrm{T}} \cdot \boldsymbol{v} \right] \\
=&\frac{\partial}{\partial x_k} \left[\left(v_i v_j \boldsymbol{e}_i \boldsymbol{e}_j \right) \cdot v_m \boldsymbol{e}_m \right] \boldsymbol{e}_k \\
=&\frac{\partial}{\partial x_k} v_i v_j v_m \delta_{im} \delta_{jk} \\
=&\frac{\partial \left(v_i v_j v_i \right)}{\partial x_j} \\
=&v_i v_j \frac{\partial v_i}{\partial x_j} + v_i \cancelto{0}{v_i \frac{\partial v_j}{\partial x_j}} + v_j v_i \frac{\partial v_i}{\partial x_j} \\
=&2\boldsymbol{v} \cdot (\boldsymbol{v} \cdot \nabla \boldsymbol{v}).
\end{aligned} \tag{3.35}$$

Note that $v_i v_j \dfrac{\partial v_i}{\partial x_j} = v_i \boldsymbol{e}_i \cdot \left[\left(v_j \boldsymbol{e}_j \cdot \dfrac{\partial v_i}{\partial x_j} \boldsymbol{e}_j \right) \boldsymbol{e}_i \right] = \boldsymbol{v} \cdot (\boldsymbol{v} \cdot \nabla \boldsymbol{v})$. Similarly, $v_j v_i \dfrac{\partial v_i}{\partial x_j} = \boldsymbol{v} \cdot (\boldsymbol{v} \cdot \nabla \boldsymbol{v})$. By using the conservation of mass, $-\nabla \cdot \boldsymbol{v} = 0$, we get $v_i v_i \dfrac{\partial v_j}{\partial x_j} = 0$.

$$\begin{aligned}
&(\boldsymbol{v} \otimes \boldsymbol{v})^{\mathrm{T}} \cdot \boldsymbol{v} \\
=& \left(v_i v_j \boldsymbol{e}_i \boldsymbol{e}_j \right) \cdot v_m \boldsymbol{e}_m \\
=&v_i v_j v_i \boldsymbol{e}_j \\
=& \left(v_i \boldsymbol{e}_i \cdot v_i \boldsymbol{e}_i \right) v_j \boldsymbol{e}_j \\
=&(\boldsymbol{v} \cdot \boldsymbol{v}) \boldsymbol{v}.
\end{aligned} \tag{3.36}$$

Therefore,

$$\int_\Omega \nabla \cdot \left[(\boldsymbol{v} \otimes \boldsymbol{v})^{\mathrm{T}} \cdot \boldsymbol{v} \right] d\Omega = \int_\Gamma \left[(\boldsymbol{v} \otimes \boldsymbol{v})^{\mathrm{T}} \cdot \boldsymbol{v} \right] \cdot \boldsymbol{n} d\Gamma. \tag{3.37}$$

Substituting Eqs. (3.35) and (3.36) into Eq. (3.37), Eq. (3.34) can be obtained.

Then, we apply the integration by parts formula to Eq. (3.33), together we use the conservation of mass, we can get:

$$\begin{aligned}
&\int_\Gamma \frac{1}{2} (\boldsymbol{v} \cdot \boldsymbol{v}) \boldsymbol{v} \cdot \boldsymbol{n} d\Gamma + \int_\Gamma p \boldsymbol{v} \cdot \boldsymbol{n} d\Gamma - \int_\Gamma \frac{1}{\mathrm{Re}} \boldsymbol{n} \cdot \left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^{\mathrm{T}} \right) \cdot \boldsymbol{v} d\Gamma \\
+&\int_\Omega \frac{1}{\mathrm{Re}} \nabla \boldsymbol{v} : \left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^{\mathrm{T}} \right) d\Omega + \int_\Omega \alpha \boldsymbol{v} \cdot \boldsymbol{v} d\Omega = 0,
\end{aligned} \tag{3.38}$$

where $\Gamma = \Gamma_{\mathrm{in}} \cup \Gamma_{\mathrm{out}} \cup \Gamma_{\mathrm{wall}}$, and $\boldsymbol{v} = 0$ on Γ_{wall}. Therefore,

$$\begin{aligned}
&\int_{\Gamma_{\mathrm{in}} \cup \Gamma \mathrm{out}} \left(p + \frac{1}{2} |\boldsymbol{v}|^2 \right) (\boldsymbol{v} \cdot \boldsymbol{n}) d\Gamma - \int_{\Gamma_{\mathrm{in}} \cup \Gamma \mathrm{out}} \frac{1}{\mathrm{Re}} \boldsymbol{n} \cdot \left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^{\mathrm{T}} \right) \cdot \boldsymbol{v} d\Gamma \\
+&\int_\Omega \frac{1}{\mathrm{Re}} \nabla \boldsymbol{v} : \left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^{\mathrm{T}} \right) d\Omega + \int_\Omega \alpha \boldsymbol{v} \cdot \boldsymbol{v} d\Omega = 0.
\end{aligned} \tag{3.39}$$

Then, the second term in Eq. (3.39) becomes zero when the flow on $\Gamma = \Gamma_{\text{in}} \cup \Gamma_{\text{out}}$ is fully developed, i.e., when $\boldsymbol{v} = U\boldsymbol{n}$, yielding the following:

$$\int_{\Gamma_{\text{in}} \cup \Gamma \text{out}} \frac{1}{\text{Re}} U\boldsymbol{n} \cdot \left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^{\text{T}} \right) \cdot \boldsymbol{n} d\Gamma = \int_{\Gamma_{\text{in}} \cup \Gamma \text{out}} \frac{1}{\text{Re}} U\boldsymbol{n} \otimes \boldsymbol{n} : \left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^{\text{T}} \right) d\Gamma = 0. \tag{3.40}$$

Furthermore, the third term in Eq. (3.39) can be rewritten as:

$$\int_{\Omega} \frac{1}{\text{Re}} \nabla \boldsymbol{v} : \left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^{\text{T}} \right) d\Omega = \int_{\Omega} \frac{1}{2\,\text{Re}} \left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^{\text{T}} \right) : \left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^{\text{T}} \right) d\Omega. \tag{3.41}$$

Finally, we obtain the following equation:

$$\int_{\Omega} \left(\frac{1}{2\,\text{Re}} \left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^{\text{T}} \right) : \left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^{\text{T}} \right) + \alpha \boldsymbol{v} \cdot \boldsymbol{v} \right) d\Omega = \int_{\Gamma_{\text{in}} \cup \Gamma \text{out}} \left(p + \frac{1}{2} |\boldsymbol{v}|^2 \right) (-\boldsymbol{v} \cdot \boldsymbol{n}) d\Gamma, \tag{3.42}$$

where the left hand side represents the total potential energy while the right hand side represents the total pressure drop.

In Table 3.2, we summarize the TO methodology, modelling strategy of these studies and compare the objective values of Eq. (3.32).

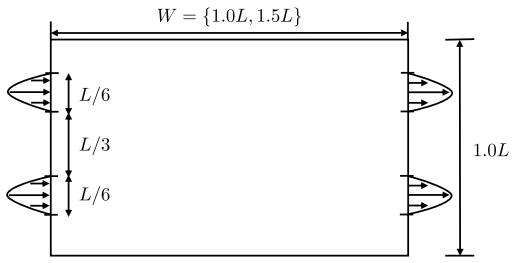
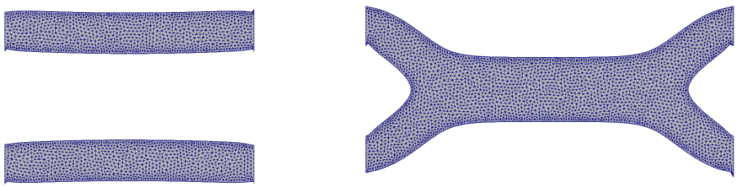


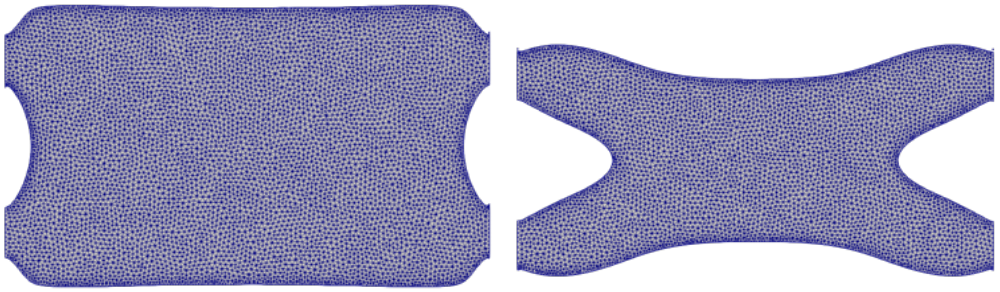
FIGURE 3.15: Design model of the 2D minimum power dissipation benchmark problem.



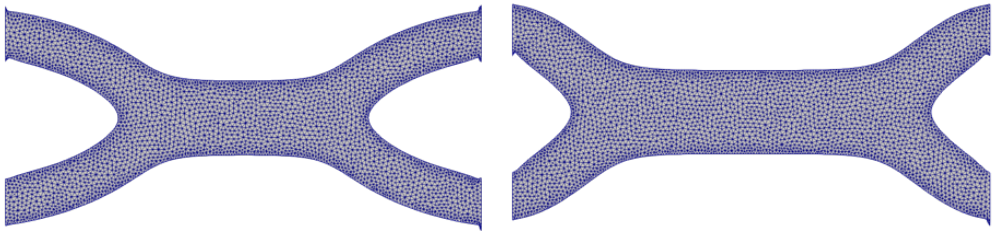
(A) $W = 1.0L$ (B) $W = 1.5L$

FIGURE 3.16: Optimal solution of the 2D Stokes flow minimal power dissipation problem, cf. Section 3.4.2.

Next, we solve the Navier–Stokes flow problem with a higher Reynolds number ($Re = 200$) for the case where $W = 1.5L$. The topology evolution can be observed in Fig. 3.19. Unlike the



(A) Iteration #3. (B) Iteration #7.



(C) Iteration #37. (D) Iteration #150.

FIGURE 3.17: Snapshots of the channel layout during optimization of 2D Stokes flow minimal power dissipation problem for the case where $W = 1.5L$.

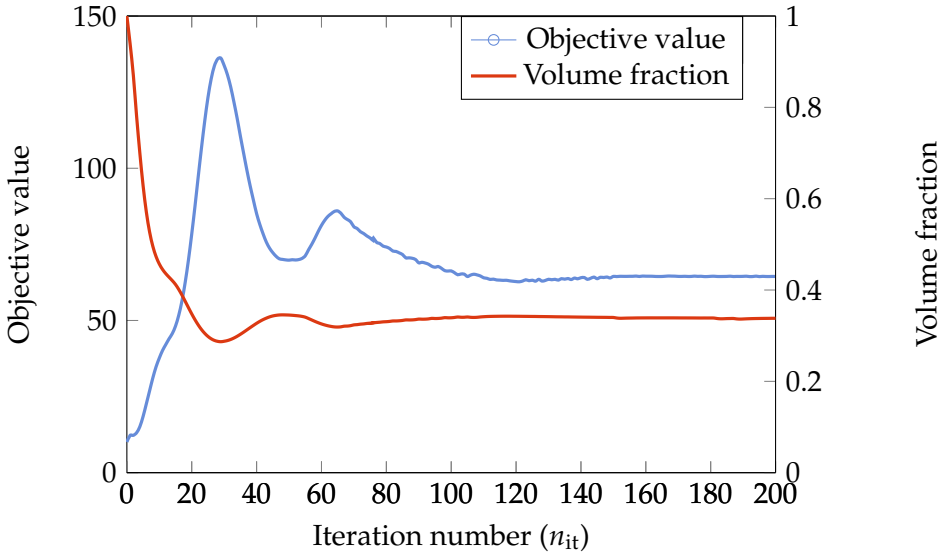
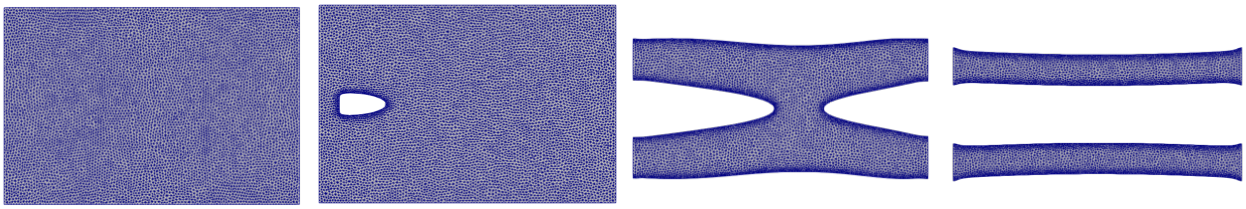


FIGURE 3.18: Histories of the objective value and the volume fraction for the 2D Stokes flow minimal power dissipation problem, cf. Fig. 3.17.

TABLE 3.2: Comparison of the TO methodology, modelling strategy, and the objective value of the 2D Stokes flow minimal power dissipation problem, cf. Fig. 3.16b.

Case	TO method	Modelling strategy	$W = 1.0L$
Analytical solution	-	-	32.0
Borrvall and Petersson (2003)	Density	Hybrid	25.67
Challis and Guest (2009)	LSM	Hybrid	31.68
Souza, Yamabe, Sá, Ranjbarzadeh, Picelli, and Silva (2021)	TOBS	Hybrid	31.28
This work	LSM	Separate	31.16

Stokes flow result as shown in Fig. 3.16b, the double channel remains almost straight between the in-/outlets. The iterative history of the objective value and volume fraction is plotted in Fig. 3.20.



(A) Iteration #1. (B) Iteration #8. (C) Iteration #10. (D) Iteration #150.

FIGURE 3.19: Snapshots of the channel layout of 2D Navier–Stokes flow minimal power dissipation problem for the case where $W = 1.5L$ under Reynolds of 200.

3D pipe connector design

Next, by applying the methodology presented, a three-dimensional pipe connector is optimized. The design model is shown in Fig. 3.21. The fluid enters the domain via the inlet on the left-hand side with a parabolic velocity profile. The diameter of the inlet and outlet are equal to the characteristic length L. There are six pipes in the middle which are the non-design domains (in dark blue). Three pipe connectors (in white) are considered as the design domains. In this example, the Reynolds number is set to $Re = 300$, τ is 10^{-3}, and the maximum allowed volume fraction is 40%.

The topology evolution is shown in Fig. 3.22. The iterative histories of the objective value and volume fraction are plotted in Fig. 3.23. As can be observed in Fig. 3.22, the most essential features of the final topology have been formed after about 20 iterations, the surface reaches stable after about 60 iterations and there are no more topological changes with the increase of the iterations. However, it should be noted that the convergence is not fully reached at the last iterative step due to the generation of the small-scale features of the adaptive meshes.

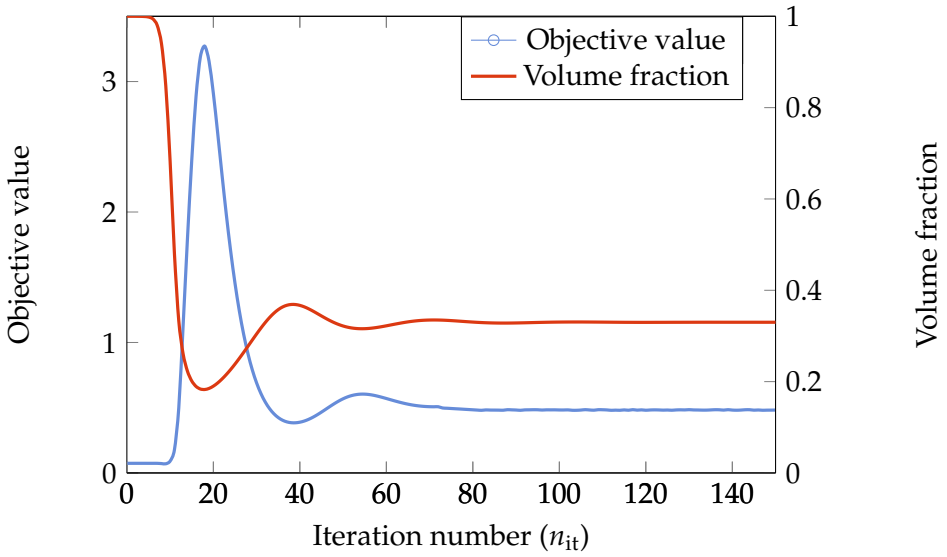


FIGURE 3.20: Histories of the objective value and the volume fraction for the 2D Navier–Stokes flow minimal power dissipation problem, cf. Fig. 3.19.

Then, we validate the performance using COMSOL Multiphysics. The generative pipe connector design is converted to the CAD B-Rep as shown in Fig. 3.24a. The eight pipes (including the inlet and outlet) are prefixed as the cylinder shape during the conversion process. After that, we use the built-in mesh tool in COMSOL Multiphysics to generate the boundary layer mesh as shown in Fig. 3.24b. The objective values obtained by the FreeFEM optimization solver and COMSOL simulation solver are summarized in Table 3.3. The numerical error could be explained by the facts that: (1) the B-Reps. conversion can inevitably introduce some loss in surface accuracy; and (2) the boundary layered mesh is used in the COMSOL solver while it is not used in FreeFEM solver. The different flow streamlines distribution before and after the pipe connectors are optimized can be clearly seen in Figs. 3.24c and d. The required pump power can be calculated as Li, Ding, Jing, Xiong, and Meng (2019):

$$P_{pump} = \Delta p \cdot \dot{V}. \tag{3.43}$$

where Δp is the pressure drop between the inlet and outlet, and $\dot{V}$ is the volumetric flow rate. The required pump power for the optimized design is 22.9% less than the initial design.

TABLE 3.3: Comparison of the objective value of the 3D pipe connector design, cf. Section 3.4.2.

Case	FreeFEM optimization solver	COMSOL simulation solver
Initial design	1.27	1.23
Generative design	1.03	0.994

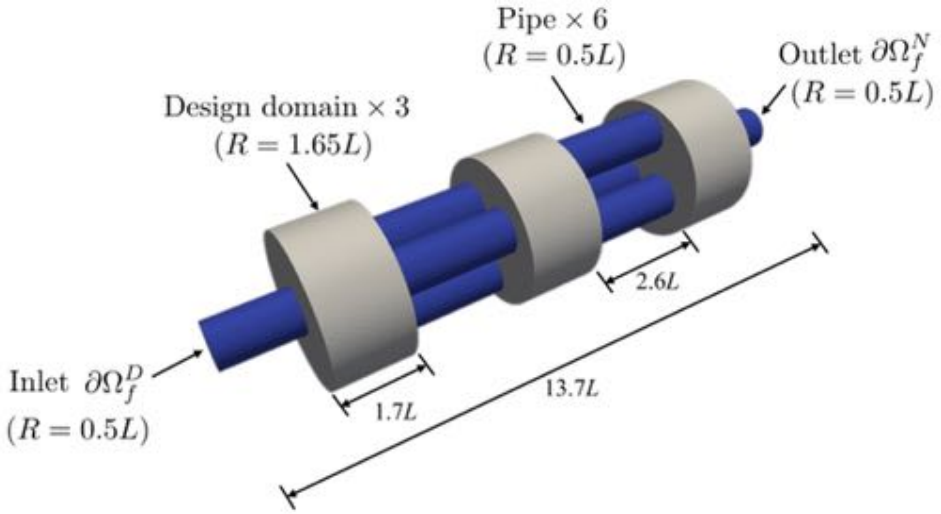
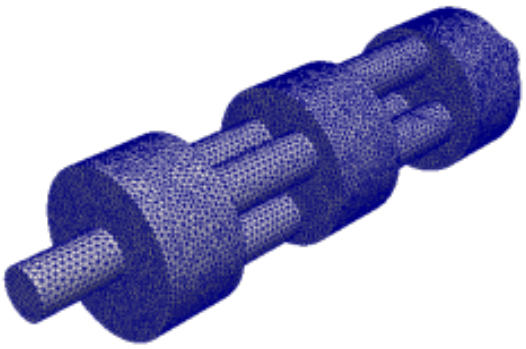
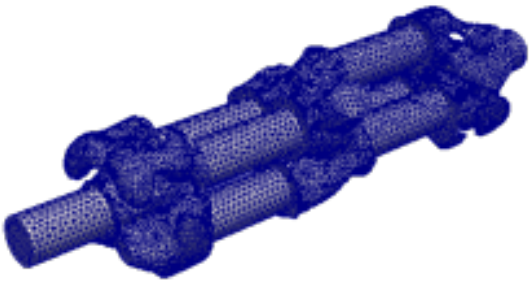


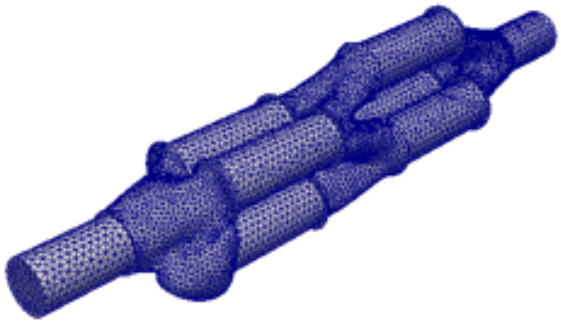
FIGURE 3.21: Design model of the pipe connector.



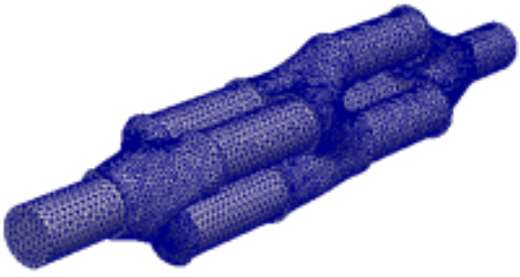
(A) Iteration #1.



(B) Iteration #16.



(C) Iteration #23.



(D) Iteration #120.

FIGURE 3.22: Snapshots of the pipe connector layout during the optimization, cf. Section 3.4.2.

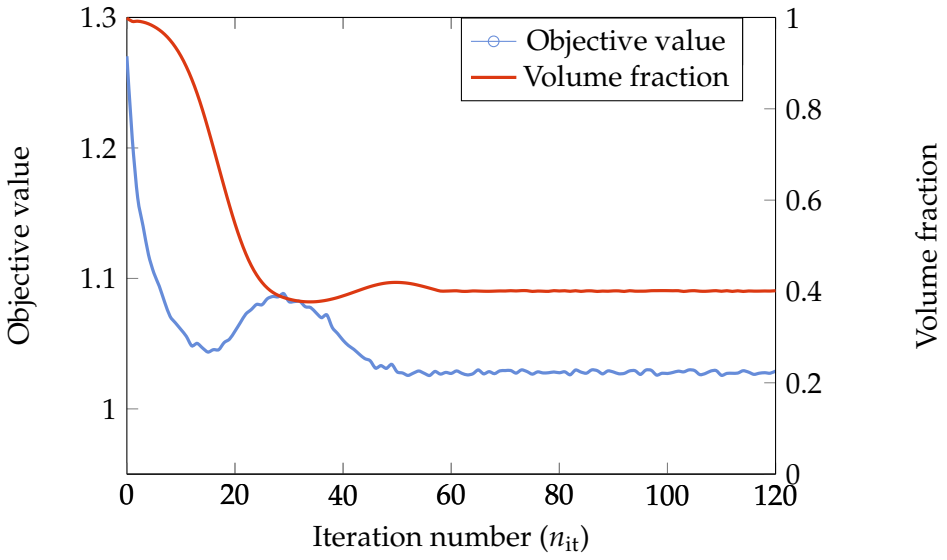
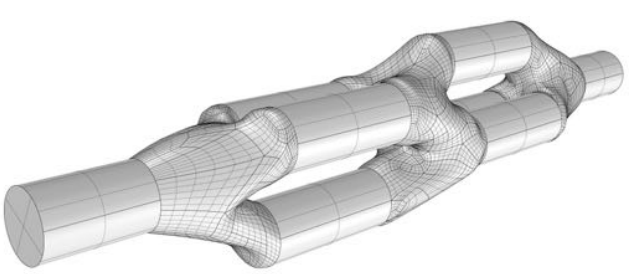
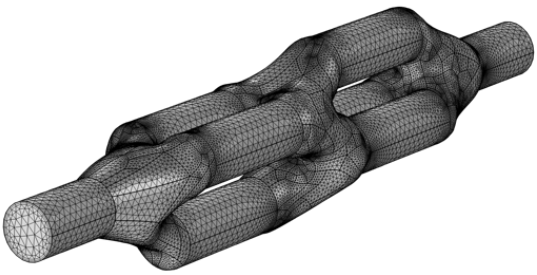


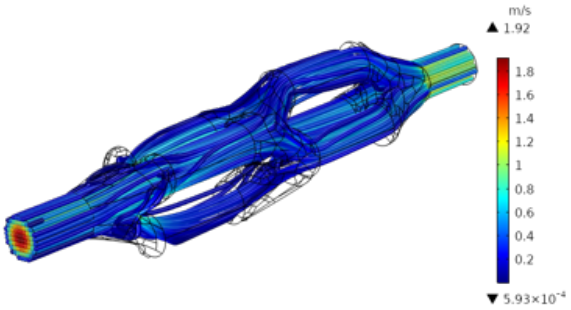
FIGURE 3.23: Histories of the objective value and the volume fraction for the 3D Navier–Stokes flow minimal power dissipation problem, cf. Fig. 3.22.



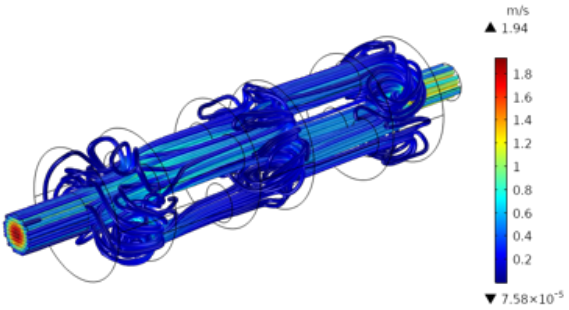
(A) B-Rep conversion.



(B) Boundary layer mesh used for CFD simulation in COMSOL.



(C) Velocity streamline in the optimized design.



(D) Velocity streamline in the initial design.

FIGURE 3.24: Post-processing and simulation of the generative design for the pipe connector, cf. Section 3.4.2.

3.4.3 "Dry" FSI optimum design

After solving the compliance problem and the power dissipation problem separately, we now focus on our first multi-physics test case–"dry" FSI optimum design. The optimization mathematical model is illustrated in Section 3.2.3.

2D benchmark problem

First, we reassemble a 2D benchmark problem given by Jenkins and Maute (2015) which is an infill structure design as shown in Fig. 3.25. The design domain is the beam (in blue) which is immersed in a flow channel. The fluid enters the domain from inlet on the left-hand side with a parabolic profile $v_x = -6(y - \frac{1}{2}L)(y + \frac{1}{2}L)$ and flows out via the outlet on the right-hand side with a zero normal stress boundary condition $\boldsymbol{\sigma}_f \cdot \boldsymbol{n}_f = 0$. The diameter of the inlet and outlet are equal to the characteristic length L. The Reynolds number of the flow is set to 10. The top and bottom walls are no-slip walls. The beam is pinned on the bottom wall, $\boldsymbol{u} = [0, 0]^T$. In this example, the maximum allowed volume fraction is 60%.

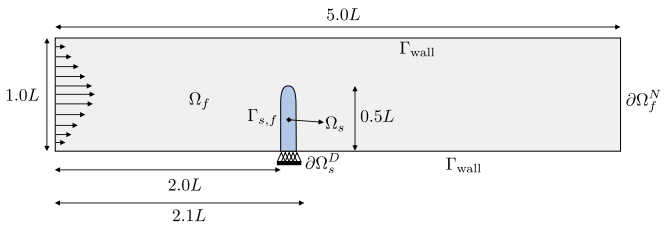
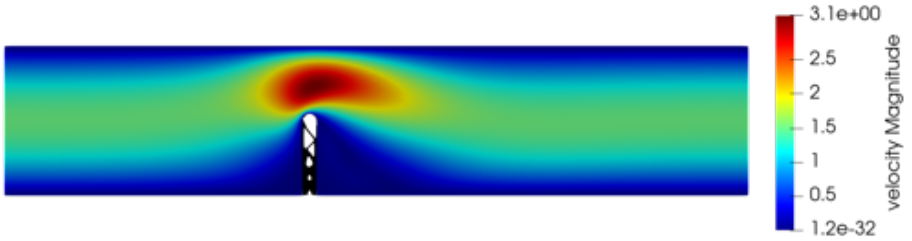


FIGURE 3.25: Design model of the 2D "dry" FSI benchmark problem.

Fig. 3.26 shows the velocity and pressure distribution of the optimized structure. The topology evolution history is plotted in Fig. 3.27. The iterative histories of the objective value and volume fraction are plotted in Fig. 3.28. In Fig. 3.27, the blue domain denotes the solid material while the yellow domain denotes the void. Not surprisingly, more solid material is placed along the upstream side (left side) to stiffen the skin layer against the deformation. More solid material is distributed at the root of the beam to reduce the overall bending. Very interestingly, although the design details are slightly different from those given in Jenkins and Maute (2015), one can say that the general configurations are quite similar. In our intuition, the weakly-/strongly-coupled condition has a certain degree of influence on the optimal solution, but such influence is quite insignificant. Indeed, making comparison between these two coupling conditions is beyond the scope of this paper, but is targeted for future studies, especially when the flow performance needs to be taken into account.

Furthermore, we demonstrate the effect of the regularization parameter τ on the optimal solutions, as shown in Fig. 3.29. The objective values are summarized in Table 3.4. It can be observed

that an increasing geometrical complexity can provide a better structural performance, but requires a finer mesh. The manipulation of the adaptive mesh parameters has been extensively discussed in our previous contribution Li, Yamada, Jolivet, Furuta, Kondoh, Izui, and Nishiwaki (2021) and will thus not be further detailed here.



(A) Velocity profile.

(B) Pressure field.

FIGURE 3.26: Velocity and pressure fields for the optimal solution of 2D dry-FSI problem.

TABLE 3.4: Comparison of the objective value of the 2D dry-FSI problem obtained by different values of τ, cf. Fig. 3.29.

τ	$J(\Omega)$
1×10^{-4}	4.01×10^{-6} cf. Fig. 3.29a.
1×10^{-5}	3.37×10^{-6} cf. Fig. 3.29b.
1×10^{-6}	3.34×10^{-6} cf. Fig. 3.29c.

3D sphere infill design

Next, we show a moderately large-scale three-dimensional test case. The design domain is a "sphere" which is mounted at the bottom of the flow channel. The channel has a dimension of $5L \times 0.5L \times L$. The fluid enters the domain from the inlet on the left-hand side and flows out via the outlet on the right-hand side with a zero normal stress. The length of the inlet and outlet are equal to the characteristic length L. The Reynolds number of the flow is set to 100. The top, bottom, and side walls are no-slip walls. To save computational cost, a symmetric boundary condition is

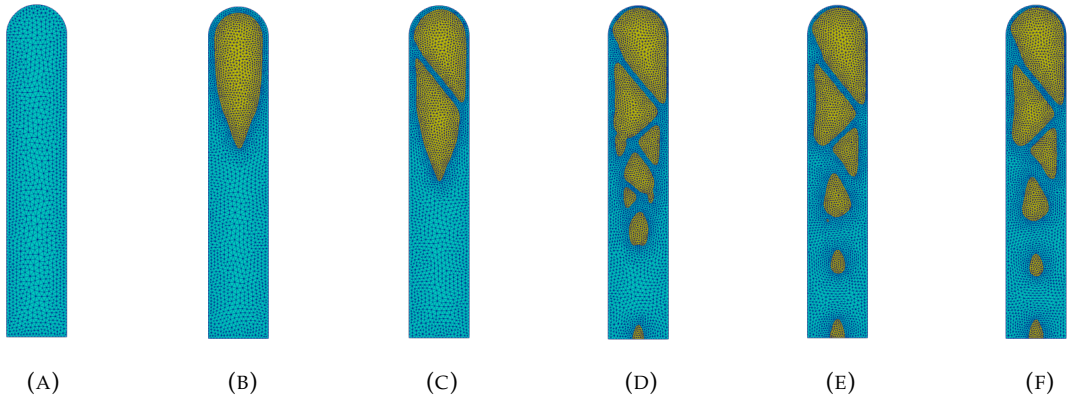


FIGURE 3.27: Snapshots of the infill structure of 2D dry-FSI problem for the case where $\tau = 1 \times 10^{-5}$ under Reynolds of 10. From left to right: iterations 1, 10, 37, 62, 120 and 150.

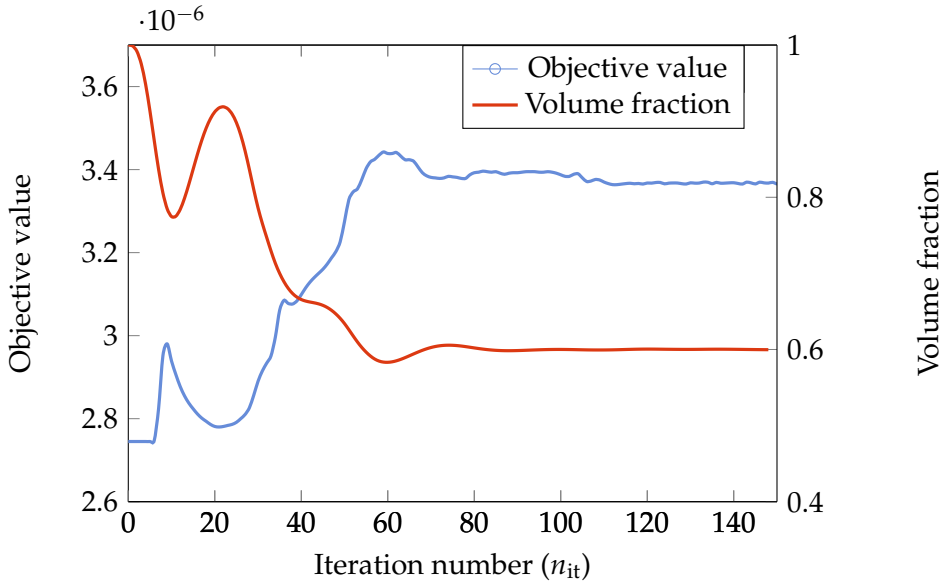


FIGURE 3.28: Histories of the objective value and the volume fraction for the 2D-dry FSI problem for the case where $\tau = 1 \times 10^{-5}$ under Reynolds of 10, cf. Fig. 3.27.

(A) $\tau = 1 \times 10^{-4}$. (B) $\tau = 1 \times 10^{-5}$. (C) $\tau = 1 \times 10^{-6}$.

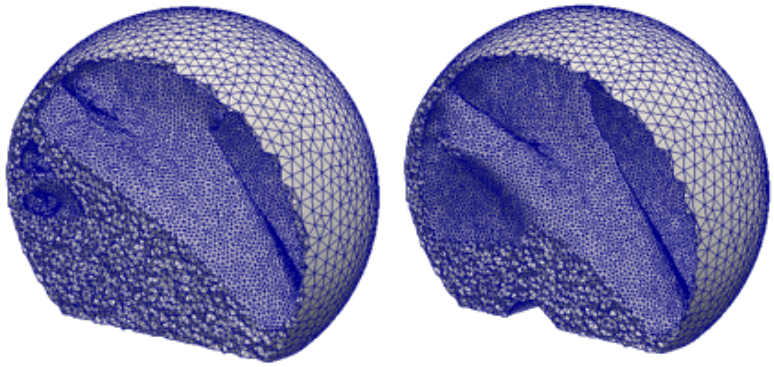
FIGURE 3.29: Optimal solutions for the 2D dry FSI problem obtained by different values of regularization parameter τ.

imposed in the XZ plane ($v_y = 0, \partial v / \partial y = 0$). The bottom surface of the "sphere" is pinned on the wall. In this example, the value of τ is set to 10^{-4} and the maximum allowed volume fraction is 25%. The topology evolution is shown in Fig. 3.30. The iterative histories of the objective value and volume fraction are plotted in Fig. 3.31.

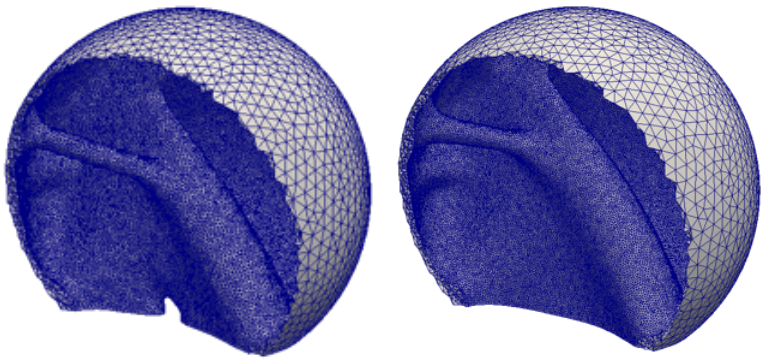
Figs. 3.32a and b show the velocity and pressure distributions, respectively, with a slice view of the optimized infill structure. The mesh representation of the optimal solution is depicted in Fig. 3.32c. The linear elastic deformation of the optimal solution is shown in Fig. 3.32d. It can be observed that the main supporting structure is placed between the top left and the bottom right corners, and several small branch-like structures are placed in the middle left to resist the deformation. Note that the deformed structure is simply amplified for visualization. Finally, the B-Rep conversion is conducted and the CAD data is obtained as shown in Fig. 3.32e. The objective values obtained by the FreeFEM optimization solver and COMSOL simulation solvers are summarized in Table 3.5.

TABLE 3.5: Comparison of the objective value of the 3D dry-FSI problem, cf. Section 3.4.3.

Case	FreeFEM optimization solver	COMSOL simulation solver
Initial design	$4.13 \cdot 10^{-10}$	$4.26 \cdot 10^{-10}$
Generative design	$1.58 \cdot 10^{-9}$	$1.62 \cdot 10^{-9}$



(A) Iteration #50. (B) Iteration #60.



(C) Iteration #80. (D) Iteration #150.

FIGURE 3.30: Snapshots of the sphere internal supporting structure during the optimization, cf. Section 3.4.3.

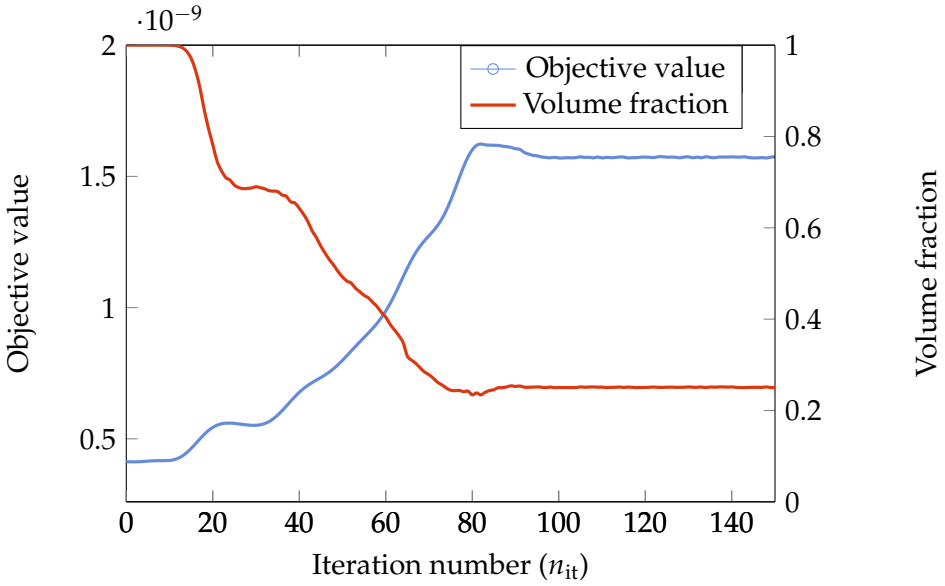
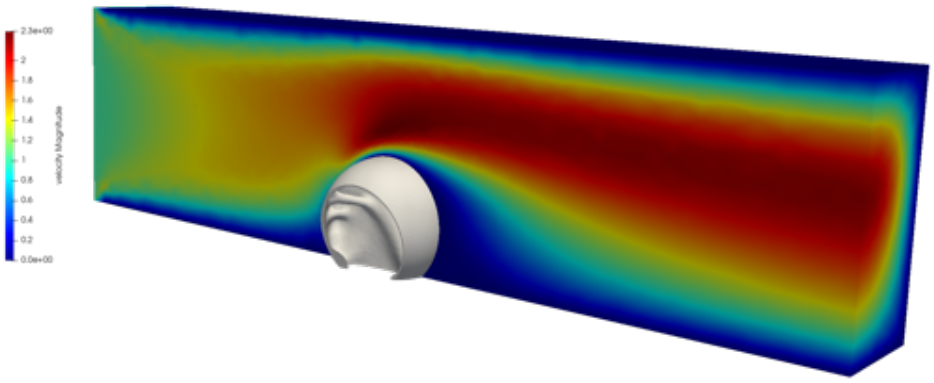
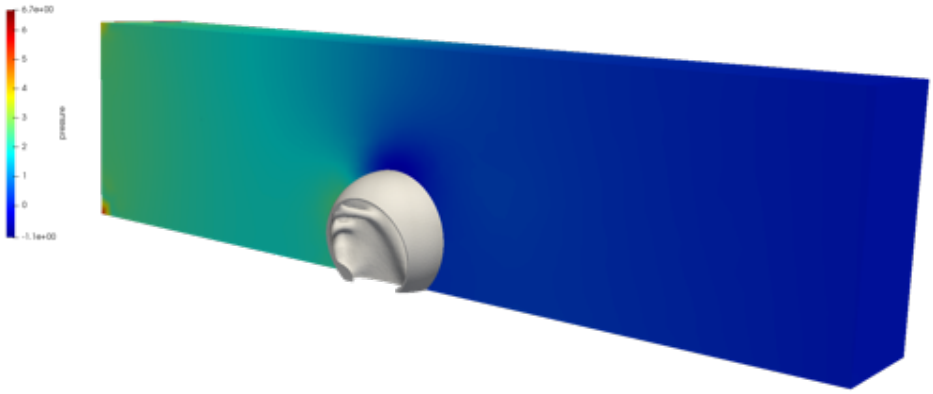


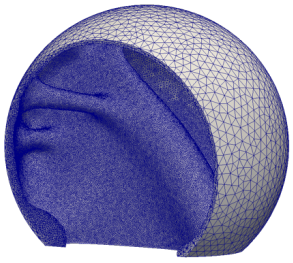
FIGURE 3.31: Histories of the objective value and the volume fraction for the 3D dry FSI problem, cf. Fig. 3.30.



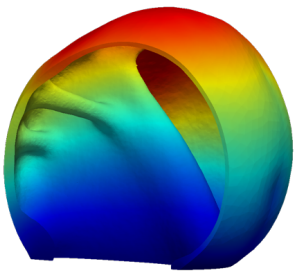
(A) Velocity distribution.



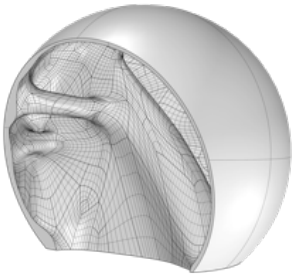
(B) Pressure distribution.



(C) Optimal solution.



(D) Deformation representation under the load force.



(E) B-Rep conversion.

FIGURE 3.32: Generative design for the sphere infill structure under fluid flow ($Re = 100$), cf. Section 3.4.3.

3.4.4 "Wet" FSI optimum design

Our second and last multi-physics case focuses on the "wet" FSI optimum design. The optimization mathematical model is illustrated in Section 3.2.3. In this section, we will demonstrate our optimization Algorithm 1 by showcasing 2D and 3D test cases.

2D benchmark problem

First, we reproduce a 2D benchmark problem given by Picelli, Ranjbarzadeh, Sivapuram, Gioria, and Silva (2020) which is a "wall support" design as shown in Fig. 3.33. An L-shaped wall Ω_s is the non-design domain, which is fixed on the bottom of the channel. The supporting structure behind the wall should be designed inside the blue domain D. The fluid enters the domain from the inlet on the left-hand side with a parabolic profile and flows out via the outlet on the right-hand side with a zero normal stress boundary condition $\boldsymbol{\sigma}_f \cdot \boldsymbol{n}_f = 0$. The diameter of the inlet and outlet are equal to the characteristic length L. The Reynolds numbers of the flow are set to 0.01, 10, and 100. The top and bottom walls are considered to be no-slip walls. In this example, the value of τ is set to 5×10^{-5} and the maximum allowed volume fraction is 40%. Fig. 3.34 shows the snapshots of the velocity profile during optimization for the case where $Re = 100$. The iterative histories of the objective value and volume fraction are plotted in Fig. 3.35.

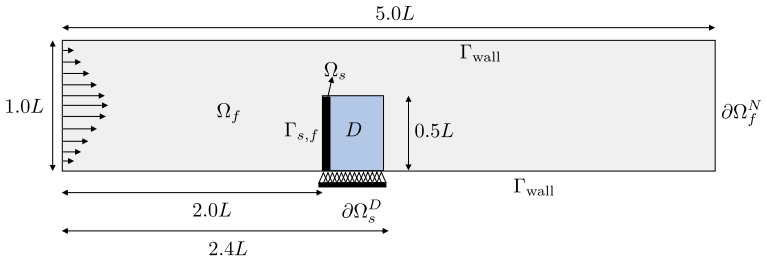
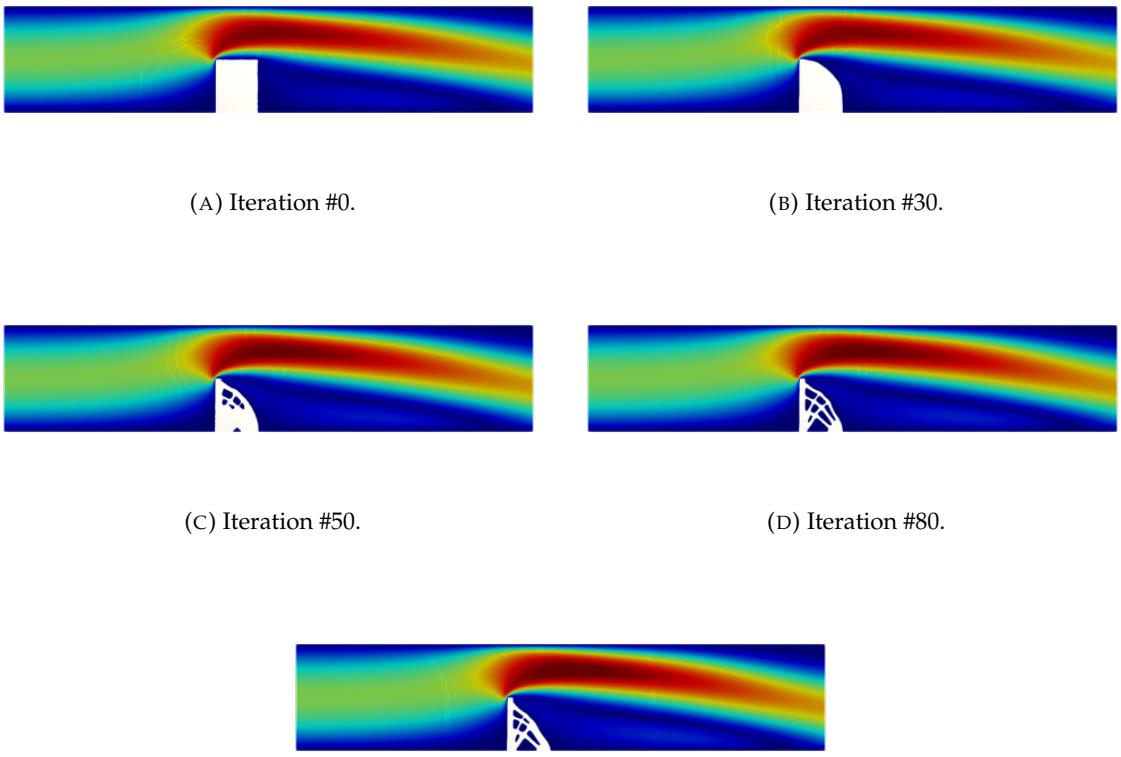


FIGURE 3.33: Design model of the 2D "wet" FSI benchmark problem.

Figs. 3.36a–c plot the optimal solution under Reynolds numbers of 0.01, 10, and 100, respectively. The corresponding velocity profiles and pressure fields are shown in Fig. 3.37. From the results, it can be observed that: (1) for a lower Reynolds number ($Re = 0.01$), the supporting structure at the downstream side shows a more concave shape compared with that for a higher Reynolds number ($Re = 100$); (2) a more complex internal supporting structure can be obtained under a higher Reynolds number; and (3) the top position of the support is placed lower for the case of a higher Reynolds number. These behaviors are quite similar to those given by Picelli, Ranjbarzadeh, Sivapuram, Gioria, and Silva (2020), Jenkins and Maute (2016), and Lundgaard, Alexandersen, Zhou, Andreasen, and Sigmund (2018). It should be noticed that since we use the body-fitted mesh adaption technique, the optimal solution of $\mathcal{T}_{\text{solid}}$ can be directly exported to the



(A) Iteration #0.

(B) Iteration #30.

(C) Iteration #50.

(D) Iteration #80.

(E) Iteration #150.

FIGURE 3.34: Snapshots of the velocity profile during optimization of 2D wet-FSI benchmark problem ($Re = 100$), cf. Section 3.4.4.

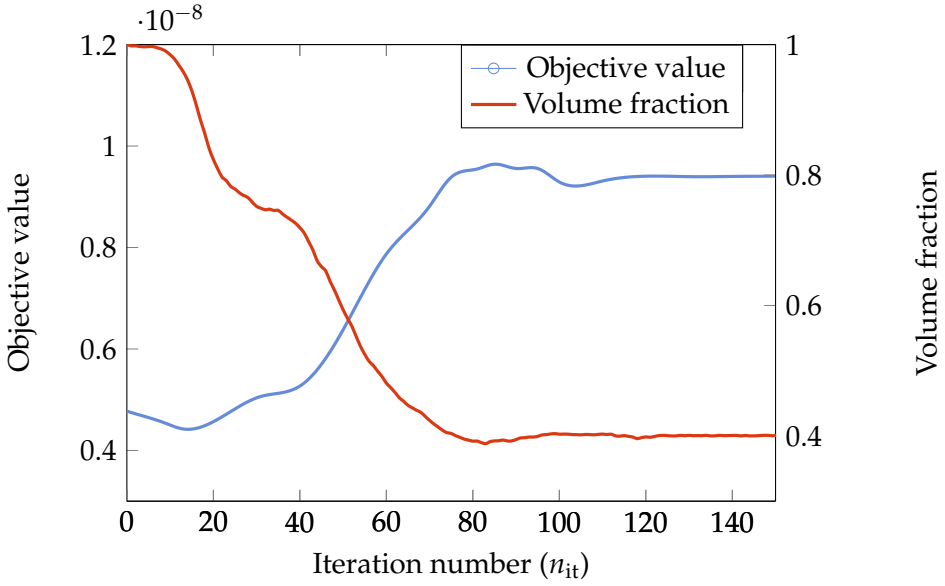
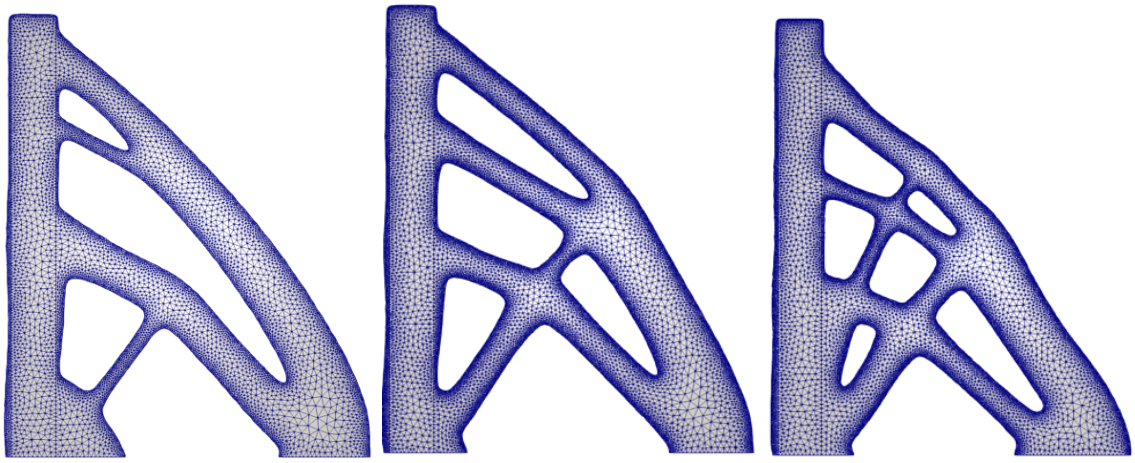
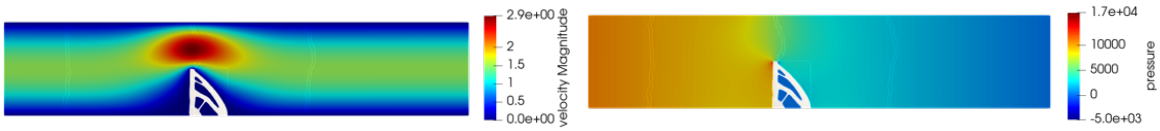


FIGURE 3.35: Histories of the objective value and the volume fraction for the 2D wet-FSI problem, cf. Fig. 3.34.

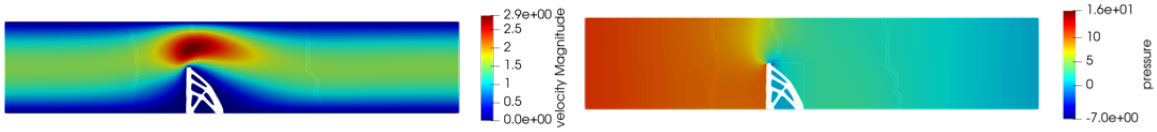


(A) $Re = 0.01$. (B) $Re = 10$. (C) $Re = 100$.

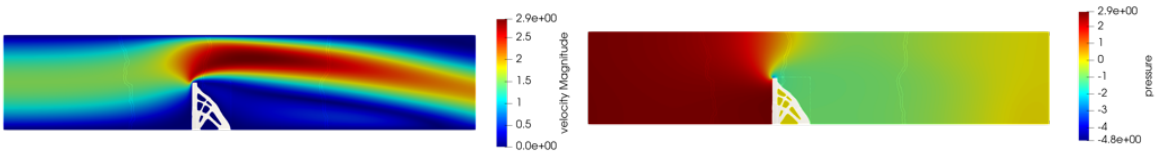
FIGURE 3.36: Optimal solutions of 2D wet-FSI problem under different Reynolds numbers, cf. Section 3.4.4.



(A) $Re = 0.01$. (B) $Re = 0.01$.



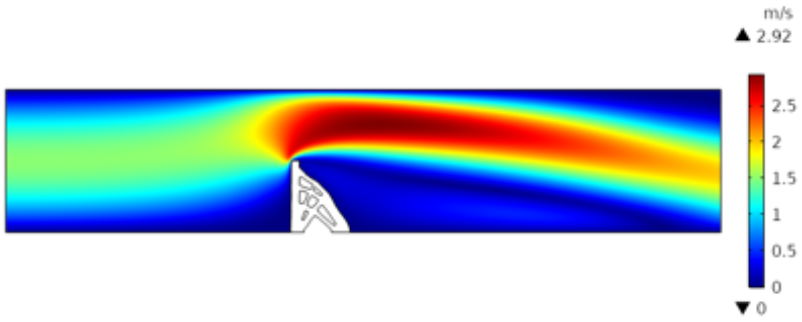
(C) $Re = 10$. (D) $Re = 10$.



(E) $Re = 100$. (F) $Re = 100$.

FIGURE 3.37: Velocity profile (left column); pressure field (right column), cf. Fig. 3.36.

COMSOL simulation solver for the performance validation. The simulation results by the COMSOL solver are shown in Fig. 3.38. After that, we compare the objective values (see Table 3.6). Note that in the COMSOL simulation solver, the fluid equations are not solved in Ω_h. This further validates the treatment of the internal holes (cf. Fig. 3.2).



(A) Velocity field (COMSOL solver).

(B) Pressure field (COMSOL solver).

FIGURE 3.38: COMSOL solver simulation results for the optimal solution under Reynolds of 100, cf. Fig. 3.36c.

TABLE 3.6: Comparison of the objective value of the 2D wet-FSI problem for Reynolds number of $Re = 100$, cf. Section 3.4.4.

Case	FreeFEM optimization solver	COMSOL simulation solver
Initial design	$4.77 \cdot 10^{-9}$	$4.7 \cdot 10^{-9}$
Generative design	$9.53 \cdot 10^{-9}$	$9.44 \cdot 10^{-9}$

Next, we perform the crosscheck, that is, the generative designs are analyzed under the different Reynolds number and the design optimized for a certain Reynolds numbers should preferably perform better than the others for its particular Reynolds number. As can be seen from Table 3.7, the designs perform as they should.

Note that in the present work, the identification of the internal holes has not been taken into consideration. In our future work, we intend to consider the additive manufacturing oriented

TABLE 3.7: Crosscheck of the objective values for the 2D wet-FSI problem, cf. Fig. 3.36.

	Optimization		
Simulation	$Re = 0.01$	$Re = 10$	$Re = 100$
$Re = 0.01$	$\mathbf{6.71 \cdot 10^{-2}}$	$6.83 \cdot 10^{-2}$	$7.3 \cdot 10^{-2}$
$Re = 10$	$9.89 \cdot 10^{-8}$	$\mathbf{9.26 \cdot 10^{-8}}$	$9.96 \cdot 10^{-8}$
$Re = 100$	$1.47 \cdot 10^{-8}$	$1.03 \cdot 10^{-8}$	$\mathbf{9.44 \cdot 10^{-9}}$

topology optimization of structures with self-supported enclosed voids (or internal holes). In that case, it will be necessary to identify the internal holes. This can be achieved by the introduction of a fictitious physics model. For example, Luo, Sigmund, Li, and Liu (2020) introduced a fictitious heat conduction model. The void domains are assumed to be filled with a temperature-dependent heating material with a higher heat conductivity while the solid domains have a lower heat conductivity. By applying a non-linear heat source term, the resulting temperature field can clearly imply the positions of the internal holes.

3D solar plate support design

Finally, we present a moderately large-scale three-dimensional "wet" FSI test case. Here, we want to improve the design of the supporting structure for a "solar plate" which is installed in an open area. The solar plate is standing under a wind load. The design model is shown in Fig. 3.39. The black domain Ω_s, including the "plate" and the "base", is the non-design domain. The yellow domain D is the design domain for the supporting structure. The computational domain has a dimension of $17.5L \times 0.65L \times 4.5L$. The height of the solar plate is equal to the characteristic length L. The flow enters the domain from the left-hand side with a velocity profile which is approximated to the Blasius solution of the laminar boundary layer equation using a smoothed Heaviside function $\mathrm{H}(x, \delta_x)$, see Eq. (3.44), and the x component of the inlet velocity profile is thus calculated as $v_x = 2\mathrm{H}(4.91z/5.5, 1) - 1$. The flow exits the domain from the right-hand side with a zero normal stress. The Reynolds number of the flow is set to 100. On the bottom wall, a no-slip boundary condition is imposed. The top wall bears a free-slip boundary condition $\boldsymbol{v} \cdot \boldsymbol{n} = 0$. A symmetric boundary condition is imposed in the XZ plane ($v_y = 0, \partial v/\partial y = 0$). The "base" is fixed on the ground, $\boldsymbol{u} = [0, 0, 0]^T$. In this example, the value of τ is set to 10^{-4} and the maximum allowed volume fraction is 30%.

$$\mathrm{H}(x, \delta_x) = \begin{cases} \frac{1}{2} + \frac{15}{16}\frac{x}{\delta_x} - \frac{5}{8}\left(\frac{x}{\delta_x}\right)^3 + \frac{3}{16}\left(\frac{x}{\delta_x}\right)^5 & 0 \le x \le \delta_x \\ 1 & \delta_x < x. \end{cases} \tag{3.44}$$

The velocity and pressure distribution for the initial guess and the optimal solution are plotted in Figs. 3.40a–d, respectively. Several intermediate snapshots of the topology are depicted

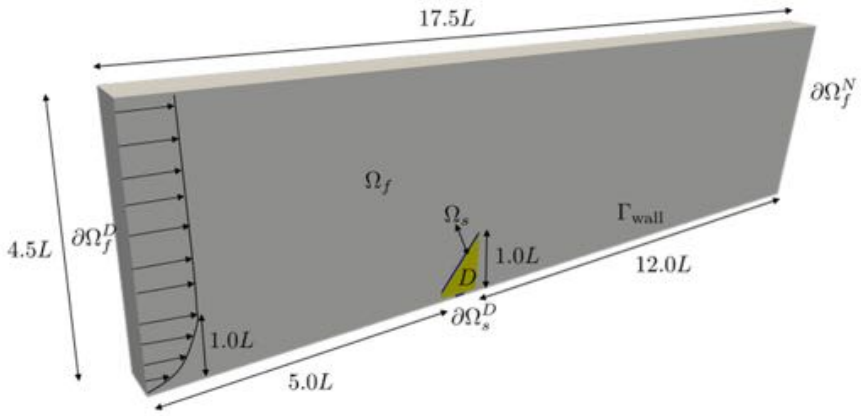


FIGURE 3.39: Design model of the solar plate support.

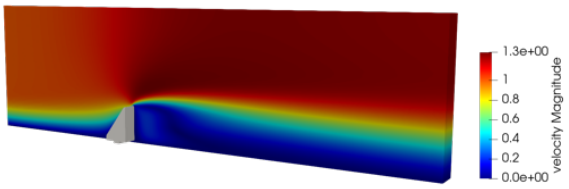
in Figs. 3.40e–j. The iterative histories of the objective value and volume fraction are plotted in Fig. 3.41. The optimal solution is quite physically reasonable because we can observe two beams that are supporting the upper part of the plate in order to reduce the stress applied by the wind load. The rest of the supports are placed at the lower part to reduce the overall bending.

For this test case, the number of tetrahedra for the first iteration is $5.35 \cdot 10^5$, where $2.65 \cdot 10^5$ of them are in the fluid domain. For the last iteration, there are $8.17 \cdot 10^5$ tetrahedra, $5.32 \cdot 10^5$ of which are in the fluid domain, having approximately $2.41 \cdot 10^6$ degrees of freedom for the linearized fluid system. The time breakdown of the finite element actions performed at the first and last iterations are summarized in Table 3.8. The total runtime for 140 iterations is approximately 20.07 h. The mentioned actions include the mesh decomposition, finite element matrix assemblies, linearized solver for Stokes problem, Newton solver for the Navier–Stokes (the Newton tolerance is set to $1 \cdot 10^{-3}$), linear solver for the lamé system, etc.

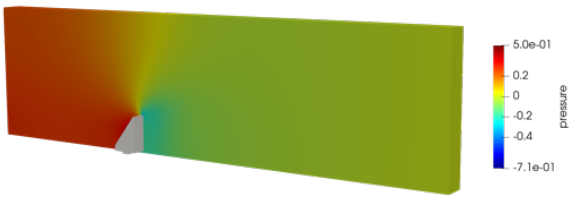
TABLE 3.8: Time breakdown of the finite element actions performed at the first and last iterations of the 3D wet-FSI problem, cf. Fig. 3.41.

Action.	Iteration #0 (s)	Iteration #140 (s)
Solve state equations	304.6	776.2
Compute topological sensitivity	29.0	56.2
Solve RDE	1.25	1.62
Output '.vtu' data for visualization	0.296	0.414
Mmg body-fitted mesh adaptation (sequential)	63.2	86.7
Partition updated mesh	8.72	12.3
Interpolate solution to the updated mesh	0.55	0.96
Total	407.6	934.4

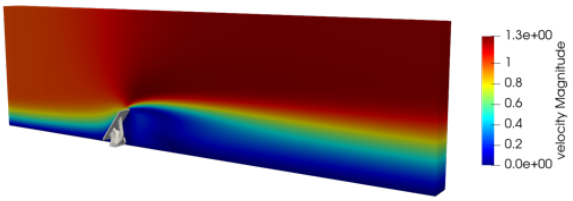
From the time breakdown, it can be observed that the most time consuming task is the solution of the state equations. The second is the body-fitted mesh adaptation. Since this part of the solver is performed sequentially on a single processor, it is indeed one of the bottlenecks in the current framework. This issue will be addressed once the Parmmg (a parallel version of Mmg recently



(A) Velocity profile for the initial guess.

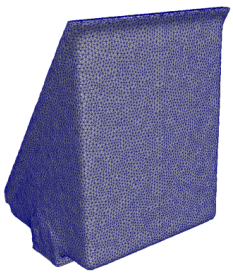


(B) Pressure field for the initial guess.

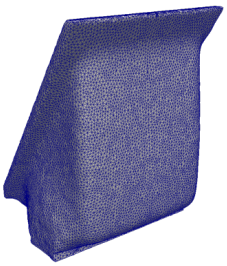


(C) Velocity profile for the optimal solution.

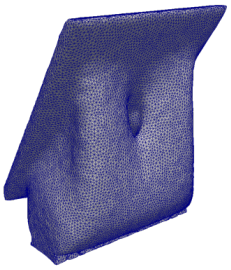
(D) Pressure field for the optimal solution.



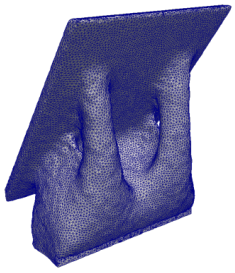
(E) Iteration #0.



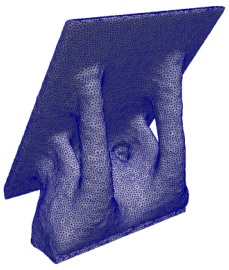
(F) Iteration #15.



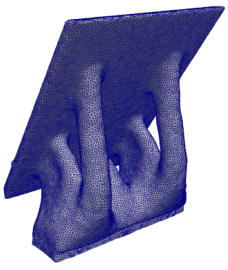
(G) Iteration #30.



(H) Iteration #50.



(I) Iteration #70.



(J) Iteration #110.

FIGURE 3.40: Generative design for the solar plate support structure under wind flow ($Re = 100$), cf. Section 3.4.4.

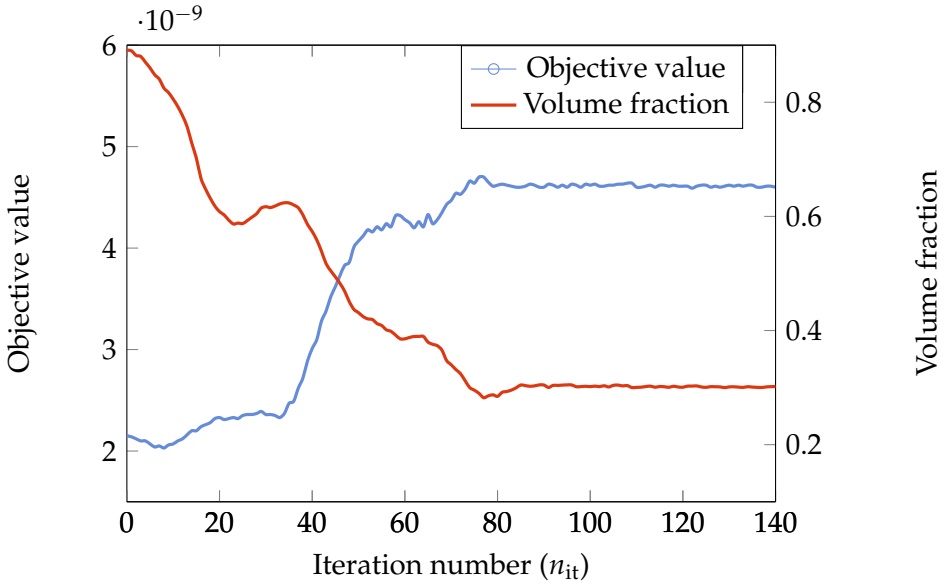
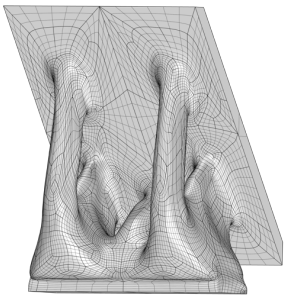


FIGURE 3.41: Histories of the objective value and the volume fraction for the 3D wet FSI problem, cf. Fig. 3.40.

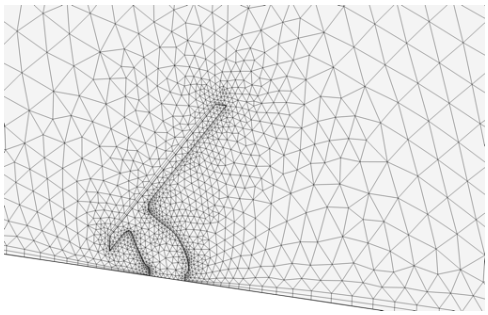
developed by Cirrottola and Froehly (2021)) will be able to handle the implicit surface mesh. An extensive discussion on this bottleneck can be found in Chapter 2.

Next, we perform the B-Rep conversion. The obtained CAD data (see Fig. 3.42a) is imported into COMSOL Multiphysics and a high-quality boundary layer mesh can be generated as shown in Fig. 3.42b. Finally, the numerical test is carried out. We simulate an array of "solar plates" standing in the open area. The wind flow enters from the left-hand side of the computational domain. The simulation result is shown in Fig. 3.42c. The computed objective values are summarized in Table 3.9.

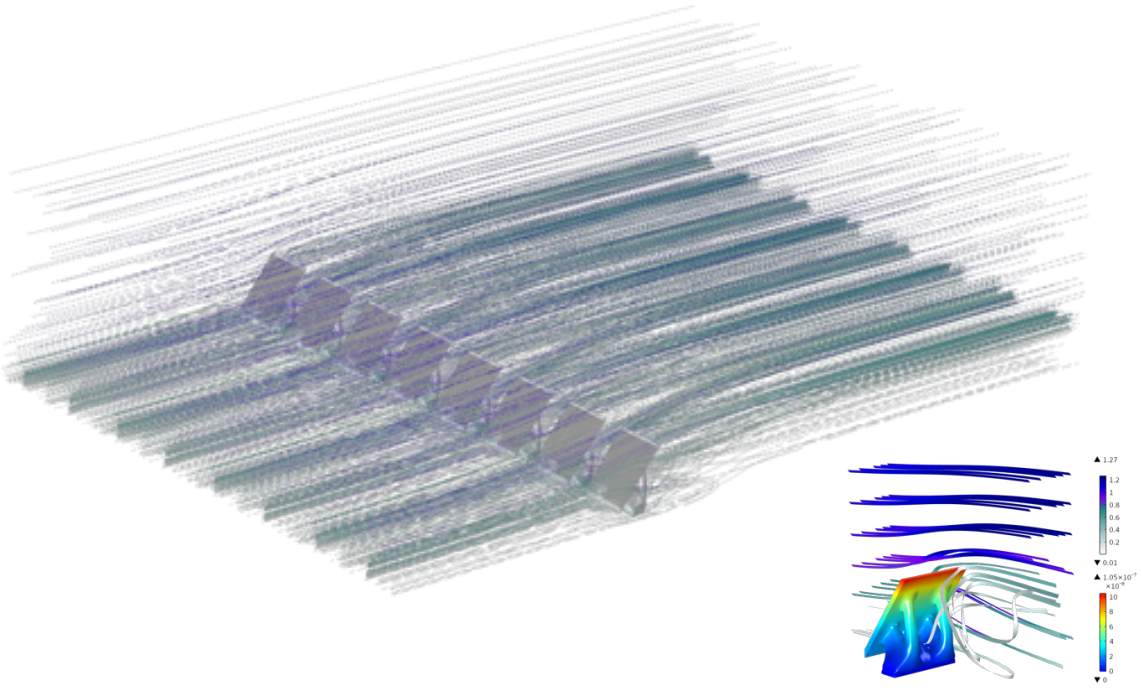
It should be mentioned that in reality the wind flow features very high Reynolds numbers. The turbulence can lead to an unsteady flow condition. Furthermore, in solar power generation, the angle of the plate facing the sun and sunlight is a key ingredient determining the efficiency of solar power generation. To maximize the efficiency, the plate needs to be rotated according to the sunlight during day time. From the engineering point of view, one should consider the multiple loading case design. By contrast, we assume the steady-state laminar flow condition, and one loading case is taken into account in the current work. The main purpose of this test case is to demonstrate the application of our design methodology, relying on the RDE-based TO method, allowing us to optimize a full-scale three-dimensional fluid–structure system, for moderate Reynolds numbers, without resorting to an initial guess containing any predefined configuration. Moreover, we demonstrate our complete product development cycle, that is, from pre-processing, topology optimization, and post-processing, to the numerical test.



(A) B-rep conversion.



(B) Boundary layer mesh.



(C) Simulation result of the solar plate support design.

FIGURE 3.42: Generative design for the solar plate support structure under wind flow ($Re = 100$), cf. Section 3.4.4.

TABLE 3.9: Comparison of the objective value of the 3D solar plate support design example, cf. Section 3.4.4.

Case	FreeFEM optimization solver	COMSOL simulation solver
Initial design	$2.15 \cdot 10^{-9}$	$2.32 \cdot 10^{-9}$
Generative design	$4.61 \cdot 10^{-9}$	$4.95 \cdot 10^{-9}$

3.5 Summary

This paper presents a new framework for two- and three-dimensional topology optimization of weakly-coupled fluid–structure systems. The main idea is to use the reaction–diffusion equation to update the level-set function, and the topological design sensitivity is used as the reaction source term. To achieve a high-resolution boundary as well as to disjoint reunion different computation domains (i.e., $\mathcal{T}_{\text{fluid}}$ for Navier–Stokes, $\mathcal{T}_{\text{solid}}$ for Lamé , $\mathcal{T}$ for RDE), we integrate a body-fitted mesh adaption technique into the suggested TO framework. The generative designs represented by the volumetric mesh can be efficiently converted to the spline B-Rep CAD data for manufacturing and numerical experiment.

To demonstrate the complete workflow of our design methodology, first, we formulate three classical TO problems: a compliance problem, power dissipation problem, and "dry" plus "wet" FSI problems. The weak forms of the governing equations, the design sensitivities, and the adjoint problems are presented. Secondly, we revisit the RDE-based level-set TO method which is the cornerstone of our work. Thirdly, we discuss the parallel computing strategy and the adaptive mesh refinement details. Next, we demonstrate the B-Rep conversion process for the post-processing. Finally, we validate the proposed workflow by solving a variety of two- and three-dimensional test cases. From the design results, the main findings of this work can be concluded as follows:

1. the suggested RDE-based level-set method allows the nucleation of new holes during the topology evolution. The boundary of the solid–void (resp. fluid–solid) is clearly described by the zero level-set interface. The body-fitted adaptive mesh technique allows the disjoint reunion of these subdomains. Therefore, by combining this remeshing technique into the suggested TO method, we are able to solve the "wet" (design-dependent) FSI problem with no predefined holes in the initial guess;

2. the open-source FEA software FreeFEM enables us to implement the suggested TO framework with a relatively compact coding. Efficient multigrid preconditioners are adopted to ensure the physical computation part to be scalable. Several full-scale three-dimensional test cases verify the parallel efficiency of the suggested solver;

3. several engineering cases (e.g., roof support design, pipe connector design, solar plate support design) are presented to highlight the potential of the proposed design workflow to real-world applications. The B-Rep conversion allows the organic part (design domain) to be integrated into (resp. extracted from) the assembly. The converted CAD data features an editable and smoothened boundary. This allows the generation of a high quality boundary layer mesh for the CFD simulation. In addition, it makes for ease of manufacturing.

We hope to conduct future research to apply the proposed design methodology to the thermo-fluidic optimization problem, i.e., a passive heat-sink cooled by natural convection. Additionally,

the body-fitted mesh adaption is so far performed sequentially on a single MPI process. This limits us for solving larger industrial-level problems. In the near future, once the parallel version of Mmg will be able to handle the implicit domain mesh adaption, it will allow us to switch to a fully parallel distributed framework. Furthermore, the post-processing still requires some manual efforts before we can meaningfully align the smooth surfaces. In the near future, we intend to introduce a fully automatic B-Rep conversion strategy which does not rely on any user guidance.

Chapter 4

Topology Optimization of Thermal–Fluid Systems[†]

CONTENTS

4.1 Introduction

With the growing trend of compactness and power density of electronic components, efficient cooling techniques have received extensive scientific attention. The cooling techniques can be broadly classified into two categories: (1) two-phase cooling; and (2) single-phase cooling. Two-phase flow is used in situations where the high heat flux is of major concern while a low pumping power is required. For instance in the solar-tracking high-concentration photovoltaic (HCPV)

[†]The work in this chapter has been published in *International Journal for Numerical Methods in Engineering*, DOI: 10.1002/nme.6923..

thermal hybrid system, a concentration ratio (CR) can go up to 100–3000 suns (1 sun = 1000 W m^{-2}) (Chemisana, 2011). For example, Hong, Zhang, Dang, and Hihara (2020) conducted outdoor real-time solar-tracking high-concentration photovoltaic (HCPV) tests under a concentration ration (CR) of 1070 suns (1 sun = 1000 W m^{-2}). With the two-phase flow microchannel heat sink (HS), the highest surface temperature on the cell surface can be controlled below 110 °C. However, phase change suffers from pressure fluctuation and flow instability which degrades the thermal performance of heat sinks (Balasubramanian, Lee, Teo, and Chou, 2013). Single-phase cooling, in contrast, is simple in structure, and high in efficiency and reliability. For example, Tuckerman and Pease (1981) first demonstrated in their experiments that the heat flux up to 790 W cm^{-2} could be dissipated with a pressure drop of 214 kPa. In a physical sense, the single-phase cooling can be further classified into: (a) active-HS cooled by forced convection; and (b) passive-HS cooled by natural convection. The former is a promising choice to achieve maximum heat dissipation and heat transfer. The latter, relying on buoyancy to drive surrounding air motion, does not require a pump or a fan to drive flow. Therefore, it is a good candidate for those low-noise and compact electronic devices such as telecommunication devices, LEDs, and semiconductors (Sidik, Muhamad, Japar, and Rasid, 2017).

The pin-fin heat sink has a good capability of heat dissipation per unit mass. Over the past few decades, a great number of efforts have been devoted to design an optimal pin-fin layout which can provide maximum thermal performance under the given constraints. The early works mainly focused on the uniformly-populated pin-fin arrangement, which means that a unit cell of pin-fin repeats itself throughout the entire design domain (Sparrow and Vemuri, 1985; Bejan, Fowler, and Stanescu, 1995). Later, it was found that the non-uniform pin-fin arrangement can enhance the heat exchange. For example, Baldry, Timchenko, and Menictas (2019) demonstrated in their experiments that the removal of central pins can facilitate the use of natural convection. Compared with the uniformly-populated pin-fin array, their design was able to reduce the temperature on the base plate by 11.7 °C, with 51% less surface area. Lee and Kim (2021) presented pin-fin heat sinks with variable fin density. In their work, the flow is modeled by a fluid-saturated porous medium, which is governed by the Brinkman–Forchheimer-extended Darcy equation. The pin-fin density was optimized using the Kriging method, and then the cylindrical pin-fins were constructed based on the obtained density field. A similar strategy was used in the design of forced-convection pin-fin heat sinks by Takezawa, Zhang, Kato, and Kitamura (2019).

The aforementioned studies delivered interesting heat sink designs, whereas these designs are restricted to the parametric models with few design variables, i.e., pin-fin shape, height, diameter, spacing, etc. To obtain a more free-form optimal configuration, this work uses the topology optimization (TO) technique.

TO technique has its origin in structural optimization (Kikuchi, Chung, Torigaki, and Taylor,

1986), but was later extended to fluid mechanics by Borrvall and Petersson (2003). Since the papers by Dede (2009) and Yoon (2010a), TO techniques have been rapidly developed in forced-convection heat sink design (Matsumori, Kondoh, Kawamoto, and Nomura, 2013; Koga, Lopes, Nova, De Lima, and Silva, 2013; Sato, Yaji, Izui, Yamada, and Nishiwaki, 2018; Yu, Ruan, Wang, Li, and Shen, 2019). Recently, the full-scale three-dimensional problems have attracted increasing scientific interest (Yaji, Ogino, Chen, and Fujita, 2018; Dilgen, Dilgen, Fuhrman, Sigmund, and Lazarov, 2018b; Kambampati and Kim, 2020; Sun, Liebersbach, and Qian, 2020; Feppon, Allaire, Dapogny, and Jolivet, 2020a; Yu, Ruan, Gu, Ren, Li, Wang, and Shen, 2020). Unlike 2D cases, the implementation in the large-scale 3D case, relying on various numerical techniques such as domain decomposition and preconditioners, requires substantial algorithmic effort. Despite a considerable increase in the numerical studies, the experimental verification has been rarely performed. But there are still a few in the literature (Li, Ding, Jing, Xiong, and Meng, 2019; Zeng, Kanargi, and Lee, 2018; Zeng and Lee, 2019; Li, Xie, Yin, Lu, Ma, Liu, and Hong, 2021). Other related fluid-based optimization research work can be referred to in a comprehensive review paper by Alexandersen and Andreasen (2020).

In contrast to the forced-convection TO problems, to date, natural convection TO problems have received much less scientific attention. Among the literature, there are three different modeling strategies for the natural convection TO problems: (1) high-fidelity model (Alexandersen, Aage, Andreasen, and Sigmund, 2014; Alexandersen, Sigmund, and Aage, 2016; Li, Lan, Zhang, and Cao, 2021; Coffin and Maute, 2016a); (2) low-fidelity model (or "Poor man's" approach) (Asmussen, Alexandersen, Sigmund, and Andreasen, 2019; Pollini, Sigmund, Andreasen, and Alexandersen, 2020); and (3) simplified convection model (Bruns, 2007; Iga, Nishiwaki, Izui, and Yoshimura, 2009; Zhou, Alexandersen, Sigmund, and Pedersen, 2016; Coffin and Maute, 2016b; Joo, Lee, and Kim, 2017; Yan, Liang, and Cheng, 2021). The first two strategies take into account the fluid motion through fluid equations while the third one does not require the solution of the flow field. Instead, an effective convection coefficient is predefined at the "fluid–solid" interface, and the convective boundary conditions are formulated based on Newton's law of cooling. Hereinafter we will mainly discuss the first two modeling strategies considering the fact that fluid motion plays an important role in the natural convection phenomenon.

The first option is that of the high-fidelity modeling, which means that the full Navier–Stokes equations are strongly coupled to the convection–diffusion equation through the Boussinesq approximation. Alexandersen, Aage, Andreasen, and Sigmund (2014) is known as the first work on topology optimization of natural convection problems, adopting the density-based approach for the design of 2D passive heat sinks and micro-pumps. Continuing their work, they proposed a PETSc-based parallel framework to demonstrate the extensibility of their methodology to the large-scale 3D design problems, by solving a problem with up to 330 million state degrees of freedom on a cluster with 1,280 processes (Alexandersen, Sigmund, and Aage, 2016). In their later

contributions, they designed passive coolers for the light-emitting diode (LED) lamps (Alexandersen, Sigmund, Meyer, and Lazarov, 2018), followed by conducting in-depth numerical and experimental comparisons to the reference lattice and pin-fin designs, where the metal-based prototypes are fabricated using additive manufacturing (AM) (Lazarov, Sigmund, Meyer, and Alexandersen, 2018; Lei, Alexandersen, Lazarov, Wang, Haertel, De Angelis, Sanna, Sigmund, and Engelbrecht, 2018). In the recent work by Li, Lan, Zhang, and Cao (2021), they reassembled 2D design results from Alexandersen, Aage, Andreasen, and Sigmund (2014) using the COMSOL in-house optimizer. They presented extensive numerical comparisons between different heating powers, heat source sizes, allowed volumes and material thermal conductivities. However, this solver appears to be only applicable to smaller scale problems. In addition to the aforementioned works on the density-based approach, Coffin and Maute (2016a) presented one of the few contributions using the level-set method (LSM). In their work, they introduced an explicit LSM incorporated with the extended finite element method (XFEM) to solve both the steady- and unsteady-state problems.

Regarding the numerical analysis methods being used in the natural convection modeling based on the full Navier-Stokes equations, besides the traditional finite element analysis (FEA) (Roy and Basak, 2005) and finite volume method (FVM) (Kim, Lee, Ha, and Yoon, 2008), a meshfree method "ISPH method" (Incompressible Smoothed Particle Hydrodynamics) is an alternative option. The key idea is to discretize the fluid medium by interaction between particles rather than mesh cells, which was firstly introduced by Lucy (1977), and Gingold and Monaghan (1977). Recently, Aly (2020) modified this method to simulate the mixing processes between the solid/fluid particles governed by the natural convection flow in an H-shaped cavity. See also their recent works (Aly and Raizah, 2020; Raizah, Ahmed, and Aly, 2020) for more case studies.

In effect, one of the bottlenecks for popularizing the high-fidelity modeling strategy is its high computational cost. To reduce the computational cost, one option is to consider the low-fidelity model (or "Poor man's" approach). This strategy was originated in the turbulent forced convection cooling channel designs (Zhao, Zhou, Sigmund, and Andreasen, 2018; Kambampati and Kim, 2020), but recently extended to the natural convection problems (Asmussen, Alexandersen, Sigmund, and Andreasen, 2019; Pollini, Sigmund, Andreasen, and Alexandersen, 2020). In the work by Zhao, Zhou, Sigmund, and Andreasen (2018), they suggested a way to approximate the thin boundary layers for turbulent flow by using a Darcy flow model, where the inertia is not considered in this model. Soon after that, Asmussen, Alexandersen, Sigmund, and Andreasen (2019) extended this idea to 2D natural convection problems. A potential flow model is formulated to "reduce" the Navier–Stokes equations under the assumption that the buoyancy term is of dominance. Note that for the low-fidelity modeling strategy, the particular values of the fictitious permeability in the fluid domains should be carefully calibrated in order to achieve well-performed optimal solutions that coincide with those obtained by the high-fidelity model. Consequently, in any case, it is necessary to validate the functional performances using a high-fidelity model. To utilize the cost efficiency of the low-fidelity model, as well as to reduce the accuracy loss due to the

negligence of the terms in the high-fidelity model, the multi-fidelity design seems to be a compromise. Pollini, Sigmund, Andreasen, and Alexandersen (2020) proposed an optimization sequence using the high-fidelity model at the beginning stage to evolve the topological configuration in a correct direction, and then switched to a low-fidelity model to accelerate the optimization process. Their testing results showed that for large-scale three-dimensional problems, an overall runtime reduction of 80%–95% can be achieved. Conversely, Yaji, Yamasaki, and Fujita (2020) generated initially various patterns of topology-optimized candidates based on the low-fidelity model (incompressible laminar flow), and then performed the high-fidelity evaluation (RANS model) to obtain a promising solution.

As mentioned above, notable achievements have been made by applying different TO methodologies to the thermal–fluid system design. However, challenging issues still exist, which arise from the following aspects.

(1) Geometric representation. As for the density-based approach, the generative design needs to be extracted from the obtained density field based on a threshold value. However, it is not easy to find a general guideline from the literature to decide the particular value of this threshold. In most of the cases, it is suggested to conduct the performance validation for different threshold values by trial-and-error. The LSM is capable of describing the geometric interface accurately. However, if this accurate boundary description cannot be converted to the model for the post-processing, simulation, and manufacturing, the ability of this accurate description makes less sense than it should. Therefore, the surface capturing techniques are highly recommended such as X-FEM (Coffin and Maute, 2016a), and body-fitted adaptive mesh (Feppon, Allaire, Dapogny, and Jolivet, 2021).

(2) Flow modeling and computational cost. As has been studied in the previous contributions, some conclusions can be drawn that the full Navier–Stokes equations are more accurate for capturing the physics phenomenon of the fluid flow motion, compared to that of simplified models, i.e., Newton's law of cooling or Darcy's law for porous medium. However, solving the full NS equations usually implies a high computational cost. We selected several state-of-the-art works on the three-dimensional thermal–fluid topology optimization and we summarize the computational techniques in Table 4.2. As can be seen in these selected works, many of them were performed on multi-node clusters. Therefore, it still remains a challenge to construct a TO framework which is more affordable to most of the users, but at the same time, accurate enough to capture the behavior of the fluid flow motion.

(3) Parameter robustness. For a density-based approach, a continuation scheme is often needed for the material interpolation parameters, so as to penalize the density field to the binary structure progressively (Alexandersen and Andreasen, 2020). To obtain a robust and efficient continuation scheme for a particular problem being solved, extensive numerical experiments are required. Therefore, there still remains some improvement space to reduce the overall algorithmic efforts.

Aiming at these challenging issues, this chapter builds upon the recent advancements in fluid topology optimization to present a case study of two- and three-dimensional TO for the natural convection problems. In this chapter, the high-fidelity modeling strategy is used. We benchmark our simulation solver against the experimental analysis and other numerical analysis methods such as the ISHP method and the FVM. And we carefully investigate the flow behavior under different Grashof numbers using a fully transient simulation solver and make the steady-state assumption for the optimum design problem. Then, the reaction–diffusion equation-based level-set TO method is formulated. The main findings reveal that the proposed method is free of the continuation approach to push the design variables to the binary structure. By introducing the body-fitted mesh adaptation technique, the fluid–solid boundaries are captured by the zero-level-set interface. The generative designs can be converted into simulation models straightforwardly. We demonstrate the parallel efficiency of the proposed framework by solving a moderately large-scale TO problem with $3.56 \cdot 10^6$ DOFs on a standard multi-process platform. Our framework is coupled with open-source mesh adaptation software Mmg and with the parallel algorithms based on PETSc, implemented in FreeFEM, to deliver both two- and three-dimensional designs.

The remainder of this chapter is organized as follows. Section 4.2 formulates the governing equations, the mathematical optimization model, and the RDE-based TO methodology for the natural convection problem. Section 4.3 illustrates the numerical implementation details including the finite element modeling, parallel computing strategy, adaptive mesh evolution method, and the optimization algorithm. In Section 4.4, we first clarify the steady-state assumption, then present a variety of 2D and 3D design examples, next validate their thermal performances, and finally examine the computational cost. Lastly, the conclusion and prospective works are documented in Section 4.5.

4.2 Formulation

In this section, we present the governing equations for the natural convection problem. Next, we describe the level-set boundary expression and discuss the material interpolation schemes. Then, we formulate the optimum design problem. Finally, we recall the reaction–diffusion equation-based topology optimization method and highlight its features.

4.2.1 Governing equation

We consider a fluid–solid system which is placed in the computational domain $\Omega = \Omega_s \cup \Omega_f$ in $\mathbb{R}^d$, where $d = 2$ or 3, as shown in Fig. 4.1. This computational domain Ω consists of a fluid domain Ω_f and a solid domain Ω_s, where Ω_f and Ω_s are divided by the fluid–solid interface $\Gamma_{s,f}$. The body heat source is applied in $\omega \subset \Omega_s$.

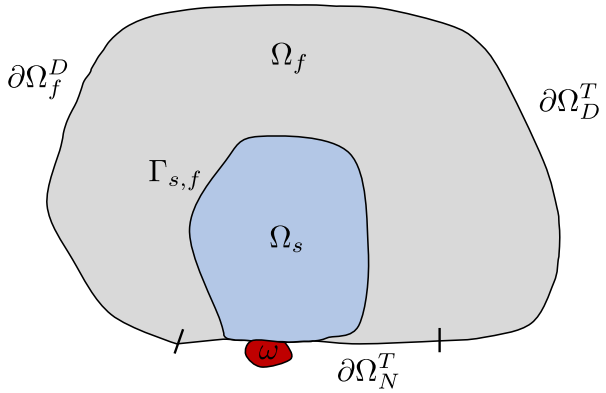


FIGURE 4.1: Schematic of the natural convection problem.

The strongly-coupled physics system to be solved is governed by the incompressible steady-state Navier–Stokes (NS) equations, with linear Boussinesq approximation of buoyancy effects, and the energy equation for the heat transportation. Thus, the dimensionless governing equations are formulated as follows:

$$\begin{cases} (\boldsymbol{v}\cdot\nabla)\,\boldsymbol{v} - \mathrm{Pr}\,\nabla\cdot\left(\nabla\boldsymbol{v}+\nabla\boldsymbol{v}^{\mathrm{T}}\right) + \nabla p - \boldsymbol{f}_{\mathrm{Da}}(\boldsymbol{v}) - f_B(T)\,\boldsymbol{e}_3 = 0 \\ -\nabla\cdot\boldsymbol{v} = 0 \\ (\boldsymbol{v}\cdot\nabla T) - \nabla\cdot(\kappa\nabla T) - Q(\mathbf{x}) = 0, \end{cases} \tag{4.1}$$

where $Q(\mathbf{x})$ is the body heat source depending on the space $\mathbf{x}$:

$$Q(\mathbf{x}) = \begin{cases} Q_0 & \mathbf{x}\in\omega \\ 0 & \mathbf{x}\in\Omega\backslash\omega. \end{cases} \tag{4.2}$$

Eq. (4.1) is endowed with appropriate boundary conditions (BC) as follows: as for the fluid field, we prescribe the velocity components on all the boundary conditions. For the temperature field, we prescribe Dirichlet BC at the outer boundaries Γ, and define the Neumann ("do-nothing") BC to the adiabatic boundaries $\partial\Omega\backslash\Gamma$,

$$\begin{cases} \boldsymbol{v} = 0 & \text{on } \partial\Omega \\ T = 0 & \text{on } \Gamma \\ \nabla T\cdot\boldsymbol{n} = 0 & \text{on } \partial\Omega\backslash\Gamma. \end{cases} \tag{4.3}$$

The dimensionless space $\boldsymbol{x}$, gradient operator ∇, velocity $\boldsymbol{v}$, pressure p, and temperature T in Eq. (4.1) are calculated by applying the following scaling to the dimensional forms, as below:

$$\boldsymbol{x}\to\frac{\boldsymbol{x}}{L}, \nabla\to\frac{\nabla}{1/L}, \quad \boldsymbol{v}\to\frac{\boldsymbol{v}}{V_{\mathrm{ref}}}, p\to\frac{p-p_0}{\delta p_0}, T\to\frac{T-T_0}{\delta T_0}=\frac{T-T_{\mathrm{ref}}}{\delta T_0}, \tag{4.4}$$

where L is the characteristic length which is usually set to the height of the cavity if the computational domain is rectangular (Sadaka, Rakotondrandisa, Tournier, Luddens, Lothodé, and Danaila, 2020). V_{ref} and T_{ref} are the reference velocity and temperature, and the temperature difference δT is used to define a temperature scale. For example, T_{ref} is the room temperature. Consequently, the dimensionless reference temperature is set to $T_0 = 0$.

The Boussinesq buoyancy force $f_B(T)$, the Grashof number Gr, and the Prandtl number Pr are given as below:

$$f_B(T) = \text{Gr}\,\text{Pr}^2\,T, \quad \text{Gr} = \frac{g\beta\delta T_0 L^3}{\nu^2}, \quad \text{Pr} = \frac{\nu}{a} = \frac{\mu C_p}{\kappa_f}, \tag{4.5}$$

where g is the gravitational acceleration, β is the thermal expansion coefficient, ν is the kinematic viscosity, $a = \dfrac{\kappa}{\rho C_p}$ is the thermal diffusivity with κ the thermal conductivity, ρ the density, and C_p the specific heat. In this work, we set $\text{Pr} = 1$.

The aformentioned dimensionless parameters (Gr and Pr) are pertinent only in the fluid phase. The following dimensionless parameters should be defined in both phases:

(1) the dimensionless thermal conductivity $\kappa(\mathbf{x})$ is defined as:

$$\kappa(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \Omega_f \\ \frac{1}{C_k} & \mathbf{x} \in \Omega_s, \end{cases} \tag{4.6}$$

where $c_k = \dfrac{\kappa_s}{\kappa_f}$, is the ratio of the thermal conductivities of the solid and fluid phases. In the present work, we set $c_\kappa = 100$;

(2) the dimensionless fictitious body force term $\boldsymbol{f}_{\text{Da}}(\boldsymbol{v}) = -\alpha\boldsymbol{v}$ is introduced in the momentum equation, where α is the inverse permeability of the fictitious porous medium given as below:

$$\alpha(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \in \Omega_f \\ \gg 1, & \mathbf{x} \in \Omega_s \end{cases}, \boldsymbol{f}_{\text{Da}}(\boldsymbol{v}) = \begin{cases} 0, & \mathbf{x} \in \Omega_f \\ \gg 1, & \mathbf{x} \in \Omega_s. \end{cases} \tag{4.7}$$

4.2.2 Level-set boundary expression and binary structure

Here, we recall the basic concept of the level-set-based TO method proposed by Yamada, Izui, Nishiwaki, and Takezawa (2010). A TO problem for the fluid optimization problem is defined to include both fluid and solid domains. Let the computational domain be denoted as $D \subset \mathbb{R}^d$. A fluid domain is then defined as $\Omega \subseteq D$ and a solid domain is represented as the complementary domain $D \backslash \Omega$. Therefore, the level-set function ϕ can be defined to have a piecewise constant profile as follows:

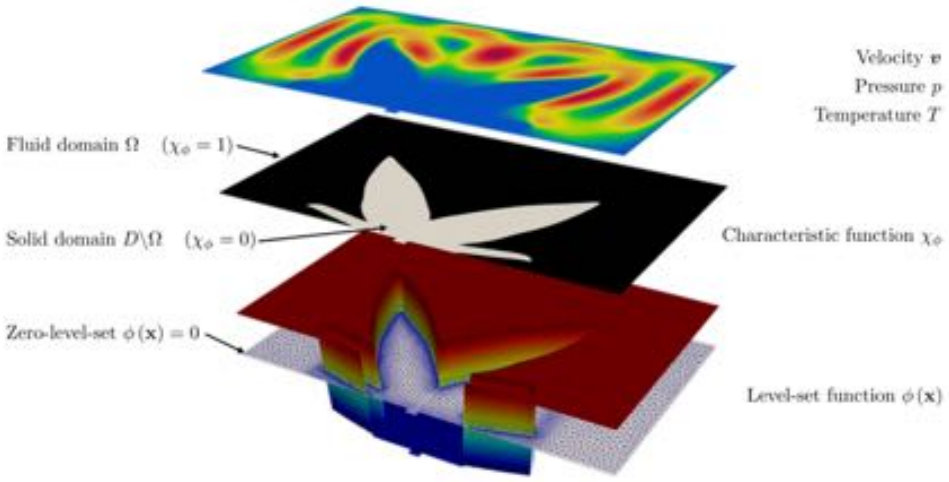


FIGURE 4.2: Schematic of the level-set function $\phi(x)$ and the characteristic function χ_ϕ. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\begin{cases} 0 < \phi(\mathbf{x}) \leq 1 & \text{for } \mathbf{x} \in \Omega \\ \phi(\mathbf{x}) = 0 & \text{for } \mathbf{x} \in \partial\Omega \\ -1 \leq \phi(\mathbf{x}) < 0 & \text{for } \mathbf{x} \in D\backslash\Omega. \end{cases} \tag{4.8}$$

The fluid–solid interface is clearly represented by the iso-surface where $\phi(\mathbf{x}) = 0$. Such clear boundary representation allows us to use the so-called "body-fitted mesh evolution strategy" which has been detailed in Section 2.3.2.

Next, a fluid–solid system can be modeled using a "1/0 binary structure" by mapping the level-set function $\phi(\mathbf{x})$ to the characteristic function χ_ϕ, as follows:

$$\chi_\phi = \begin{cases} 1 & \text{if } \phi \geq 0 \\ 0 & \text{if } \phi < 0. \end{cases} \tag{4.9}$$

This means that there is no "porous medium" throughout the topology optimization process. Then, we use χ_ϕ to interpolate the material properties (i.e., κ and α). The details of the interpolation scheme will be given in Section 4.2.3. Finally, the forward problem described in Section 4.2.1 can be solved. The schematic of the level-set boundary expression and the characteristic function mapping are illustrated in Fig. 4.2.

4.2.3 Interpolation of the material property

As we shall show below, our methodology differs from those level-set methods which rely on the boundary variation of Hadamard, namely the shape derivative, for describing the sensitivity of functions with respect to the domains. In contrary, our method uses the Ersatz material approach, where the material properties are interpolated by the characteristic function χ_ϕ, which is a function of level-set function ϕ. ϕ is evolved by solving a time evolution equation which has be explained in Section 2.2.5. Now, we interpolate the material properties (κ and α) using the characteristic function χ_ϕ. A simple scheme is linear interpolation as follows:

$$\begin{cases} \kappa\left(\chi_\phi\right) = \chi_\phi\left(1 - c_\kappa\right) + c_\kappa \\ \alpha\left(\chi_\phi\right) = \alpha_{\max}\left(1 - \chi_\phi\right), \end{cases} \tag{4.10}$$

where $\alpha_{\max}$ is the upper bound of the effective inverse impermeability in the solid domains. Linear interpolation was used in our previous works where the forced convection problem is considered (Yaji, Yamada, Kubo, Izui, and Nishiwaki, 2015; Sato, Yaji, Izui, Yamada, and Nishiwaki, 2018) . However, in these two works, it was assumed that the thermal conductivity of the fluid and solid phases are the same ($c_k = \kappa_s/\kappa_f = 1$). In our experience, a higher c_k, e.g., $c_k = 100$, causes bad scaling of initial sensitivity, which makes the topology rapidly converges towards a low-quality local optima. This is addressed in this chapter by using the RAMP (Rational Approximation of Material Properties) scheme (Stolpe and Svanberg, 2001):

$$\begin{cases} \kappa\left(\chi_\phi\right) = 1 + \left(c_k - 1\right)\dfrac{q_\kappa\left(1 - \chi_\phi\right)}{q_\kappa + \chi_\phi} \\ \alpha\left(\chi_\phi\right) = \alpha_{\max}\dfrac{q_\alpha\left(1 - \chi_\phi\right)}{q_\alpha + \chi_\phi}, \end{cases} \tag{4.11}$$

where q_κ and q_α are the penalty factors (or convexity factors) for κ and α, respectively.

The RAMP scheme is often used in the density-based approach when solving thermal–fluid problems (Alexandersen, Sigmund, and Aage, 2016; Yu, Ruan, Gu, Ren, Li, Wang, and Shen, 2020; Sun, Liebersbach, and Qian, 2020; Pollini, Sigmund, Andreasen, and Alexandersen, 2020; Dilgen, Dilgen, Fuhrman, Sigmund, and Lazarov, 2018b). The continuation scheme should be used in order to ensure the stability of the optimization process and to prevent bad-quality local optima. For example, a continuation sequence is performed on various parameters: q_κ, q_α, and $\alpha_{\max}$. The particular value of these parameters are chosen based on extensive numerical experiments.

Unlike the density-based approach, in the present work, since the binary structure is used, it can be easily found that the resulting $\kappa(\chi_\phi)$ and $\alpha(\chi_\phi)$ are independent to the penalty parameters, and will finally yield to either fluid or solid material property Eqs. (4.6) and (4.7). Therefore, the present framework needs *no continuation approach* for those penalty parameters. We can simply use a set of fixed values throughout the entire optimization process, which saves the manual efforts

for adjusting them.

In this chapter, $\alpha_{\max}$ is set to 10^7. In the authors experience, a lower order of magnitude of $\alpha_{\max}$ may cause an unstable fluid–solid surface evolution, especially for the case of higher Grashof number. Thus, its value should be large enough to prevent the fluid flow from "penetrating" the solid phase. The accuracy of the boundary effect is validated by comparing the results computed by the COMSOL Multiphysics (COMSOL, 2018) solver based on the separate modeling[1] using the boundary layered mesh, see details in Section 4.4.4. In addition, q_α is set to 0.01. q_κ can be used to control the complexity of the optimal solution and it will be discussed in Section 4.4.3.

4.2.4 Optimum design problem for the passive heat sink

Now we formulate the optimum design problem for the steady state natural convection. Considering a domain $\Omega = \Omega_s \cup \Omega_f$, we intend to find the optimal passive heat sink layout which can minimize the thermal compliance. The optimization mathematical model can be defined as follows:

$$\begin{aligned} &\inf_{\chi_\phi \in \mathcal{X}} J(\Omega) = \int_\omega QT d\Omega \\ &\text{s.t.} \quad G_1 = \frac{\int_D 1 - \chi_\phi d\Omega}{\int_D d\Omega} - V_{\max} \leq 0 \\ &\qquad Eqs.\ (4.1) to\ (4.3), \end{aligned} \tag{4.12}$$

where $V_{\max}$ is the maximum allowed volume fraction for the solid phase in the design domain.

The design sensitivity $\overline{\boldsymbol{F}}'$ can be derived as follows:

$$\begin{aligned} \overline{\boldsymbol{F}}' = \int_\Omega &- \frac{q_\alpha \alpha_{\max} \left(q_\alpha + 1\right)}{\left(q_\alpha + \chi_\phi\right)^2} \boldsymbol{v} \cdot \boldsymbol{v}_A \\ &- \nabla T_A \cdot \nabla T \frac{q_\kappa \left(c_k - 1\right) \left(q_\kappa + 1\right)}{\left(q_\kappa + \chi_\phi\right)^2} - \lambda_1 d\Omega, \end{aligned} \tag{4.13}$$

where λ_1 is the Lagrange multiplier associated to the volume constraint. In this chapter, the λ_1 is updated by means of the augmented Lagrangian multiplier method (Rockafellar, 1973; Hestenes, 1969; Powell, 1969). The details of the updating scheme can be found in Section 2.2.6 and is skipped here for brevity. $\boldsymbol{v}_A$ and T_A are the adjoint velocity and adjoint temperature, respectively,

[1] Separate modeling: the Navier–Stokes equations are computed only in the fluid domain without the fictitious body force term $f_{\mathrm{Da}}(\boldsymbol{v})$. The energy equation is separately solved in the fluid and solid domains: $(\boldsymbol{v} \cdot \nabla T) = \nabla \cdot \left(\kappa_f \nabla T\right)$ (in fluid domain); $\nabla \cdot (\kappa_s \nabla T) + Q(\mathbf{x}) = 0$ (in solid domain).

which can be computed by solving the following adjoint equations:

$$\begin{cases} -\boldsymbol{v}\cdot\nabla\boldsymbol{v}_A+\boldsymbol{v}_A\cdot\nabla\boldsymbol{v}^{\mathrm{T}}-\mathrm{Pr}\,\nabla\cdot\left(\nabla\boldsymbol{v}_A+\nabla\boldsymbol{v}_A^{\mathrm{T}}\right) \\ +\alpha\boldsymbol{v}_A+\nabla p_A+T_A\nabla T=0 \\ -\nabla\cdot\boldsymbol{v}_A=0 \\ Q-\mathrm{Gr}\,\mathrm{Pr}^2\,\boldsymbol{v}_A\cdot\boldsymbol{e}_3-\nabla T_A\cdot\boldsymbol{v}-\nabla\cdot(\kappa\nabla T_A)=0, \end{cases} \tag{4.14}$$

with the following adjoint boundary conditions:

$$\begin{cases} \boldsymbol{v}_A=0 & \text{on } \partial\Omega \\ T_A=0 & \text{on } \Gamma \\ (T_A\boldsymbol{v}+\kappa\nabla T_A)\cdot\boldsymbol{n}=0 & \text{on } \partial\Omega\backslash\Gamma. \end{cases} \tag{4.15}$$

The derivation of the adjoint problem and the sensitivity can be found in Appendix A.2.

4.3 Implementation details

4.3.1 Finite element modeling

We use a classical Galerkin finite element method to solve the systems of equations formulated in Section 4.2.

We set the classical Hilbert spaces for the velocity $\boldsymbol{v}$, pressure p, and temperature T as follows:

$$\begin{aligned} &\tilde{\boldsymbol{V}}=\tilde{V}\times\tilde{V}\times\tilde{V},\tilde{V}=H_0^1(\Omega) \\ &\tilde{P}=\left\{\tilde{p}\in L^2(\Omega)\mid\int_\Omega\tilde{p}=0\right\}. \end{aligned} \tag{4.16}$$

First, for solving the governing equations defined in Eqs. (4.1) to (4.3), the weak formulation of the Jacobian and the residual at a given linearization point $(\boldsymbol{v},p,T)$, for an increment $(\delta\boldsymbol{v},\delta p,\delta T)$, are as follows:

$$\begin{aligned} F(\boldsymbol{v},p,T)=&\int_\Omega\left((\boldsymbol{v}\cdot\nabla)\,\boldsymbol{v}\right)\cdot\tilde{\boldsymbol{v}}+\mathrm{Pr}\left(\nabla\boldsymbol{v}+\nabla\boldsymbol{v}^{\mathrm{T}}\right):\nabla\tilde{\boldsymbol{v}} \\ &-p\nabla\cdot\tilde{\boldsymbol{v}}+\alpha\boldsymbol{v}\cdot\tilde{\boldsymbol{v}}-\mathrm{Gr}\,\mathrm{Pr}^2\,T\boldsymbol{e}_3\cdot\tilde{\boldsymbol{v}}-\tilde{p}\nabla\cdot\boldsymbol{v} \\ &+\tilde{T}\,(\boldsymbol{v}\cdot\nabla T)+\left(\kappa\nabla\tilde{T}\right)\cdot\nabla T d\Omega+\int_\omega Q\tilde{T}d\Omega\quad\forall\left(\tilde{\boldsymbol{v}},\tilde{p},\tilde{T}\right)\in\tilde{\boldsymbol{V}}\times\tilde{P}\times\tilde{V} \end{aligned} \tag{4.17}$$

$$
\begin{aligned}
DF\left(\boldsymbol{v}, p, T\right)\left(\delta\boldsymbol{v}, \delta p, \delta T\right) = \int_{\Omega} & \left(\left(\delta\boldsymbol{v}\cdot\nabla\right)\boldsymbol{v}\right)\cdot\tilde{\boldsymbol{v}} + \left(\left(\boldsymbol{v}\cdot\nabla\right)\delta\boldsymbol{v}\right)\cdot\tilde{\boldsymbol{v}} + \mathrm{Pr}\left(\nabla\tilde{\boldsymbol{v}} + \nabla\tilde{\boldsymbol{v}}^{T}\right) : \nabla\delta\boldsymbol{v} \\
& - \delta p\nabla\cdot\tilde{\boldsymbol{v}} + \alpha\delta\boldsymbol{v}\cdot\tilde{\boldsymbol{v}} - \mathrm{Gr}\,\mathrm{Pr}^{2}\,\delta T\boldsymbol{e}_{3}\cdot\tilde{\boldsymbol{v}} - \tilde{p}\nabla\cdot\delta\boldsymbol{v} \\
& + \tilde{T}\left(\delta\boldsymbol{v}\cdot\nabla T\right) + \tilde{T}\left(\boldsymbol{v}\cdot\nabla\delta T\right) + \nabla\tilde{T}\cdot\left(\kappa\nabla\delta T\right) d\Omega \\
& \forall\left(\tilde{\boldsymbol{v}}, \tilde{p}, \tilde{T}\right) \in \tilde{\boldsymbol{V}}\times\tilde{P}\times\tilde{V},
\end{aligned}
\tag{4.18}
$$

where $\tilde{\boldsymbol{v}}$, $\tilde{p}$, and $\tilde{T}$ are the test functions for the increments $\delta\boldsymbol{v}$, δp, and δT, respectively. Then, the nonlinear problem is solved using a Newton method where the following linearized equations are solved in sequence:

$$
DF\left(\boldsymbol{v}, p, T\right)\left(\delta\boldsymbol{v}, \delta p, \delta T\right) = F\left(\boldsymbol{v}, p, T\right). \tag{4.19}
$$

Secondly, for the adjoint equations defined in Eqs. (4.14) and (4.15), the system of equations is expressed in the weak form as follows:

$$
\begin{aligned}
& \int_{\Omega} -\tilde{\boldsymbol{v}}_{A}\cdot\left(\boldsymbol{v}\cdot\nabla\boldsymbol{v}_{A}\right) + \tilde{\boldsymbol{v}}_{A}\cdot\left(\boldsymbol{v}_{A}\cdot\nabla\boldsymbol{v}^{\mathrm{T}}\right) + \mathrm{Pr}\,\nabla\tilde{\boldsymbol{v}}_{A} : \left(\nabla\boldsymbol{v}_{A} + \boldsymbol{v}_{A}^{\mathrm{T}}\right) \\
& + \tilde{\boldsymbol{v}}_{A}\cdot\alpha\boldsymbol{v}_{A} - p_{A}\nabla\cdot\tilde{\boldsymbol{v}}_{A} + \tilde{\boldsymbol{v}}_{A}\cdot\left(T_{A}\nabla T\right) - \tilde{p}_{A}\nabla\cdot\boldsymbol{v}_{A} d\Omega \\
& + \int_{\omega}\tilde{T}_{A}Q d\Omega - \int_{\Omega}\tilde{T}_{A}\,\mathrm{Gr}\,\mathrm{Pr}^{2}\,\boldsymbol{v}_{A}\cdot\boldsymbol{e}_{3} + \tilde{T}_{A}\nabla T_{A}\cdot\boldsymbol{v} - \kappa\nabla\tilde{T}_{A}\cdot\nabla T_{A} d\Omega \\
& + \int_{\partial\Omega\setminus\Gamma}\tilde{T}_{A}T_{A}\boldsymbol{v}\cdot\boldsymbol{n} d\Gamma = 0 \quad \forall(\tilde{\boldsymbol{v}}_{A}, \tilde{p}_{A}, \tilde{T}_{A}) \in \tilde{\boldsymbol{V}}\times\tilde{P}\times\tilde{V},
\end{aligned}
\tag{4.20}
$$

where $\tilde{\boldsymbol{v}}_{A}$, $\tilde{p}_{A}$, and $\tilde{T}_{A}$ are the test functions for the adjoint velocity, adjoint pressure, and adjoint temperature, respectively.

Thirdly, the reaction–diffusion equation defined in Eq. (2.35) is written in the weak form as follows:

$$
\begin{cases}
\displaystyle\int_{D}\frac{1}{\Delta t}\phi_{n+1}\tilde{\phi} dD + \int_{D}\nabla^{T}\phi_{n+1}(\tau\nabla\tilde{\phi}) dD = \int_{D}\left(-\tilde{C}\bar{F}' + \frac{1}{\Delta t}\phi_{n}\right)\tilde{\phi} dD \quad \forall\tilde{\phi}\in\tilde{V} \\
\phi = 0 \qquad\qquad \text{on } \partial D,
\end{cases}
\tag{4.21}
$$

where $\tilde{\phi}$ is the test function for ϕ.

The values of the level-set function are bounded in the range of $[-1, 1]$. This constraint is to ensure that the effect of the regularization term is approximately constant around the boundaries. Hence, the geometric complexity of the topological configuration can be controlled by adjusting the regularization factor τ. Furthermore, we can use an anisotropic variation of τ to realize the extrusion constraint. For example, by increasing the regularization factor value τ_{n} along the extrusion direction (at least one order of magnitude higher than the values of τ_{t1} and τ_{t2}), the output field can be kept nearly constant along the extrusion direction. We will present a test case later in Section 4.4.3.

For the space discretization of the above-mentioned systems, we use the mini-element[2] introduced by Fortin and Brezzi (1991) and Arnold, Brezzi, and Fortin (1984), and approximate v and $\tilde{v}$ with piecewise P_1-bubble finite elements, and p, T, and ϕ with piecewise linear P_1 finite elements:

$$\mathbb{P}^1_{b|h} = \left\{ v \in H^1(\Omega) \mid \forall K \in \mathcal{T}_h, v_{|K} \in P_1 \oplus \text{Span}\left\{ \lambda_i^K \right\} \right\}, \tag{4.22}$$

where K is an element of the triangle (2D) or tetrahedron (3D) element $\mathcal{T}_h$, with the characteristic mesh size h, and λ_i^K, $i = 0, ..., d$ are the $d+1$ barycentric coordinate functions of the element K. In the very recent work by Sadaka, Rakotondrandisa, Tournier, Luddens, Lothodé, and Danaila (2020), it was reported that the mini-element has remarkable accuracy when compared to the classical Taylor–Hood discretization element, while the runtime can be reduced significantly. They also pointed out that a careful assessment of this type of discretization in computing more complicated geometry cases should be conducted. We examine it and further validate the accuracy in our work (see Section 4.4.4).

As has been discussed in Chapter 2, an open-source finite element software FreeFEM (Hecht, 2012; Jolivet, Dolean, Hecht, Nataf, Prud'homme, and Spillane, 2012) is used for the discretization of PDEs. FreeFEM is a high-level programming language whose syntax is close to the mathematical formulations, which thus makes the implementation quite straightforward. It has been used to implement different shape and topology optimization methods, (see the educational papers for LSM (Allaire and Pantz, 2006), phase field method (Kim, Mingook, Takayuki, Shinji, and Jeonghoon, 2020), and density-based approach (Zhu, Zhang, Li, Liang, Wang, Li, and Nishiwaki, 2021)). To show the user-friendly syntax, the FreeFEM code implementation for the adjoint equations defined in Eq. (4.20) is briefly illustrated in Appendix B.1.1.

4.3.2 Parallel computing

As discussed above, the finite element modeling method outlined in Section 4.3.1 can be coded in FreeFEM. For parallelism, it utilizes a distributed-memory paradigm through the Message Passing Interface (MPI).

First, the global mesh $\mathcal{T}$ is partitioned into N non overlapping meshes $\{\mathcal{T}_i\}_{1 \le i \le N}$ using a graph partitioner package like METIS (Karypis and Kumar, 1998), as shown in Figs. 4.8a and d. This step can be realized by calling the macro `buildDmesh`. Then, each MPI process holds the local mesh corresponding to each subdomain. After that, the distributed assembly of the weak formulations are performed by FreeFEM. The resulting discrete linear systems are passed over to PETSc (Abhyankar, Brown, Constantinescu, Ghosh, Smith, and Zhang, 2018; Balay, Gropp, McInnes, and Smith, 1997; Balay, Abhyankar, Adams, Benson, Brown, Brune, Buschelman, Constantinescu, Dalcin, Dener, Eijkhout, Gropp, Hapla, Isaac, Jolivet, Karpeev, Kaushik, Knepley,

[2] The mini-element is the simplest element for the Stokes-type problem which provides an inf-sup stability and global linear convergence (Fortin and Brezzi, 1991; Arnold, Brezzi, and Fortin, 1984).

Kong, Kruger, May, McInnes, Mills, Mitchell, Munson, Roman, Rupp, Sanan, Sarich, Smith, Zampini, Zhang, Zhang, and Zhang, 2021; Balay, Abhyankar, Adams, Brown, Brune, Buschelman, Dalcin, Dener, Eijkhout, Gropp, Karpeyev, Kaushik, Knepley, May, McInnes, Mills, Munson, Rupp, Sanan, Smith, Zampini, Zhang, and Zhang, 2019) for distributed linear algebra. The next step is to solve these linear systems using physics-tailored preconditioners. To solve the three-dimensional natural convection problem governed by Eqs. (4.17) to (4.19), a modified augmented Lagrangian preconditioner (mAL) is used. The mathematical background of the mAL preconditioner can be found in the original work by Moulin, Jolivet, and Marquet (2019), and will not be recalled in detail here. The performance of the mAL preconditioner in a large-scale three-dimensional problem with 75 million unknowns has been reported in their work: an 80% parallel efficiency on 256 up to 2,048 processes is obtained. Thanks to the free availability of their numerical implementation in FreeFEM and PETSc, this part of the solver can be integrated in our framework rather straightforwardly. In addition, we call *hypre* (Falgout and Yang, 2002) to solve the RDE Eq. (4.21). This makes the level-set updating part become very efficient, see the table of runtime breakdown in Section 4.4.5. The aforementioned preconditioners ensure that the physical computational part of our TO framework is scalable with respect to the number of processes and the problem size.

In this chapter, all the two-dimensional tests are performed on a laptop equipped with an Intel Core i7 processor with a clock frequency of 2.8 GHz having 8 cores and 32 GB of memory. The three-dimensional tests are performed on a Linux workstation with dual Intel Xeon Gold 6246R processors with a clock frequency of 3.40 GHz, having 32 cores in total and 128 GB of memory. We use the GNU compilers and OpenMPI.

4.3.3 Mesh adaptation

Here, we introduce the third key ingredient: unstructured adaptive mesh refinement. In this work, we adopt two different types of mesh adaptation strategies: (1) body-fitted mesh, and (2) anisotropic mesh. The former is used in the topology optimization solver while the latter is integrated in the fully-transient simulation solver.

Body-fitted mesh for topology optimization

In the topology optimization solver, we use the body-fitted mesh adaptation method proposed by Allaire, Dapogny, and Frey (2013). The body-fitted adaptive mesh can accurately capture the moving fluid–solid interface during topological evolution. Furthermore, the metrics control used by FreeFEM makes it possible to simultaneously refine the mesh in the region near the explicit boundary and coarsen the mesh at far field. The dynamic sparsened grid strategy helps to save a huge amount of computational cost compared with the fixed mesh-based TO framework (see

Table 4.2). Our numerical implementation of this mesh evolution algorithm relies on an open-source library Mmg (Dapogny, Dobrzynski, and Frey, 2014), which can be called from within FreeFEM. This part of the FreeFEM code implementation can be found in Appendix B.1.2.

Here, we will not discuss further these detailed matters, since they have been extensively investigated in Chapter 2. Note that the Mmg package is performed sequentially on a single MPI process, which is still a limitation of the present TO framework (see the runtime breakdown in Table 4.3).

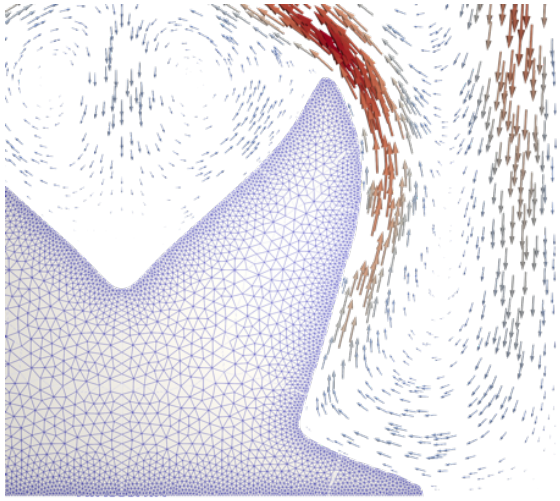


FIGURE 4.3: A snapshot of the body-fitted mesh.

Fig. 4.3 shows a snapshot of the mesh during the optimization. It can be clearly observed that the fluid–solid interface is explicitly described by the body-fitted mesh, and the elements are refined near the interface.

Anisotropic mesh for transient simulation

In the fully transient simulation solver, we use the anisotropic adaptive mesh method. Anisotropic meshes are those that introduce narrow mesh elements in the regions where the solutions are expected to change rapidly. The basic idea of this mesh evolution strategy (see Borouchaki, Castro-Diaz, George, Hecht, and Mohammadi (1996), Castro-Diaz, Hecht, and Mohammadi (2000), Hecht and Mohammadi (1997), George and Borouchaki (1998), and Mohammadi and Pironneau (2010)) is to use the Delaunay algorithm (George and Borouchaki, 1998) to generate a new triangular (2D) or tetrahedral (3D) mesh whose edges are close to the unit length in the metric $\mathcal{M} = \frac{|\mathcal{H}|}{\varepsilon}$, where $|\mathcal{H}|$ is the Hessian of the variable (i.e., velocity components v_i in this work) at point x and ε the interpolation error for the variables. The algorithm details can be found in Frey and George (1999) and are not recalled here due to the lack of space. See also the very recent work by Rakotondrandisa, Sadaka, and Danaila (2020) and Sadaka, Rakotondrandisa, Tournier, Luddens, Lothodé, and Danaila (2020) in which they reported the mesh adaptivity capability of the method

in the fluid–solid phase-change simulation solver. This part of the FreeFEM code implementation can be found in Appendix B.1.3.

4.3.4 Optimization flow chart

Now, we describe the algorithm of the proposed topology optimization framework (see Algorithm 2). First, the input mesh $\mathcal{T}$ is generated and stored in the Inria Medit format which is natively parsed by FreeFEM and Mmg. Next, the computational domain is decomposed by a standard mesh partitioner such as METIS. After that, the initial level-set field is given[3].

Then, the optimization loop begins. The forward problem is computed using the Newton method. Next, the objective functional is calculated. If it is converged, the optimization ends, otherwise, the adjoint problem is solved and the sensitivity is computed. After that, the level-set function is updated by solving the RDE, and the updated field of ϕ is bounded to satisfy $\phi(x) \in [-1, 1]$, and the characteristic function χ_ϕ can be mapped based on the bounded ϕ. Then, we perform a standard MPI reduction on process #0 to obtain a global level-set. Next, we call the sequential Mmg package to update the global mesh $\mathcal{T} \rightarrow \mathcal{T}_{it}$. Finally, the current step solutions are interpolated to the updated mesh $\mathcal{T}_{it}$. The workload is then distributed again, using MPI, until the end of the iteration where either of the following criteria are satisfied: (1) the relative difference between the objective values in current and previous steps is lower than 10^{-4} for 4 steps in a sequence; or (2) the maximum iteration number is reached (300 in this work).

Finally, we summarize in Fig. 4.4 the numerical implementation workflow. First, the commercial CAD software Rhinoceros (McNeel, 2010) is used to build the design model. Next, we use an open-source software Gmsh (Geuzaine and Remacle, 2009) to generate an initial mesh and store it in the Intira Medit format. Then, we start the topology optimization (see Algorithm 2). After obtaining the optimal solution, we convert the mesh representation to CAD data using the commercial software package S-generate (QUINT, 2021). Next, we perform the numerical test in COMSOL Multiphysics (COMSOL, 2018). Finally, the prototypes are built using a 3D printer.

4.4 Numerical investigation

In this section, we first benchmark our FEA solver against the experimental analysis and other numerical solvers. Then we clarify the rationality of the design problem setting. Next, we run our algorithm to solve two- and three-dimensional benchmarks. Then, we examine the accuracy

[3] In this chapter, for all test cases, we initialize the design domain with a uniform level-set field $\phi = -1$, indicating the solid phase. It could be said that for the thermal–fluid TO problem, the optimal solution highly depends on the initial guess. One can only reach a local optima, see more discussions in our previous work Li, Ding, Meng, Jing, and Xiong, 2019.

Algorithm 2 Natural convection topology optimization.

Input: global mesh $\mathcal{T}_0$
initialization:
decompose the computational domain for parallel computing
initialize the level-set field ϕ

for $it = 0; it < 300; it = it + 1$ **do**
  solve the forward problem on $\mathcal{T}$ using Newton method, cf. Eqs. (4.17) to (4.19)
  solve the adjoint problem on $\mathcal{T}$, cf. Eq. (4.20)
  calculate objective functional, cf. Eq. (4.12)
  if $\|J_{it+1} - J_{it}\| < \varepsilon$ **then**
    break
  else
    calculate sensitivity, cf. Eq. (4.13), and Lagrange multipliers
    update the level-set function ϕ by solving RDE on $\mathcal{T}$, cf. Eq. (4.21)
    update the characteristic function χ_ϕ, cf. Eq. (4.9)
    reduce solution of ϕ_{it} on a single process
    call Mmg to obtain $\mathcal{T}_{it+1}$ based on ϕ
    broadcast to all processes
    interpolate ϕ, χ_ϕ, v, p, and T on $\mathcal{T}_{it+1}$
  end if
end for

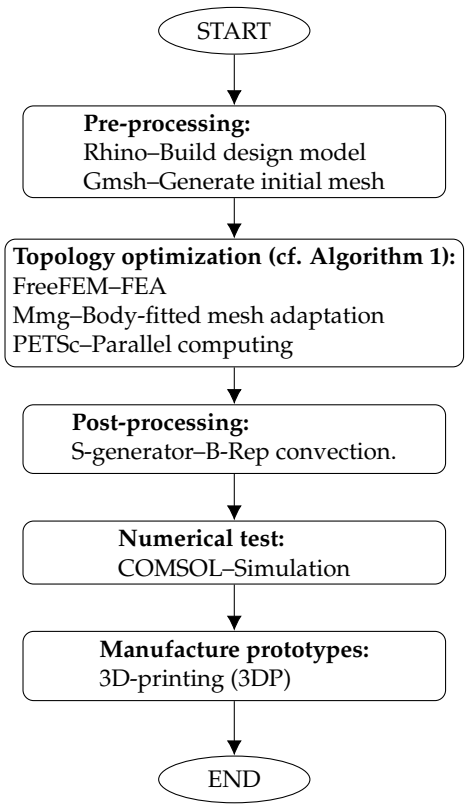


FIGURE 4.4: Workflow of the numerical implementation.

of the boundary effect and crosscheck the thermal performance of the obtained optimal solutions. Finally, we discuss the computational cost of the proposed TO framework.

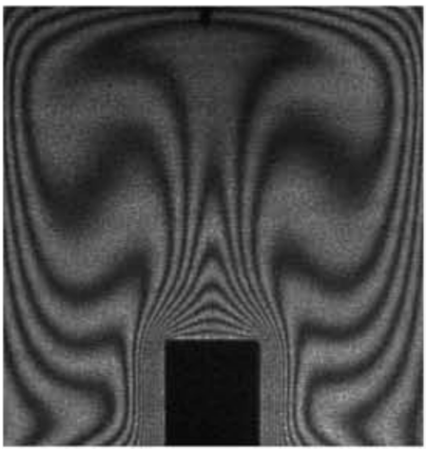
4.4.1 Verification of the FEA solver

Before we present the design problem setting, the accuracy of the FEA solver needs to be verified. To this end, we benchmark our solver against to the experimental results of Paroncini and Corvaro (2009). The test case is shown in Fig. 4.5. The heat source has a dimensionless height of $\xi = H_{\text{HS}}/H_{\text{Cavity}}$, where H_{HS} and H_{Cavity} are the heights of the heat source and the cavity, respectively. The boundary conditions are summarized as follows: (1) for the fluid, all the walls have no-slip boundary conditions; (2) for the temperature, the lateral vertical walls are set to cold wall. The three boundaries of the heated square are set to hot wall. The top and bottom walls are adiabatic wall. However, as has been also pointed out by Paroncini and Corvaro (2009), since it is not feasible in the real-world experimental analysis, the top and bottom surfaces are implemented with a convective heat flux $-\boldsymbol{n} \cdot (\kappa \nabla T) = h\,(T_{\text{ext}} - T)$, where h is the convective heat flux. Fig. 4.5 shows the isothermal contour at $\text{Ra} = 2.11 \cdot 10^5$ & $\xi = 0.25$, and $\text{Ra} = 2.25 \cdot 10^5$ & $\xi = 0.5$, where $\text{Ra} = \text{Gr}\,\text{Pr}$ is the Rayleigh number. The well agreement between the experimental and numerical results verifies the accuracy of our solver.

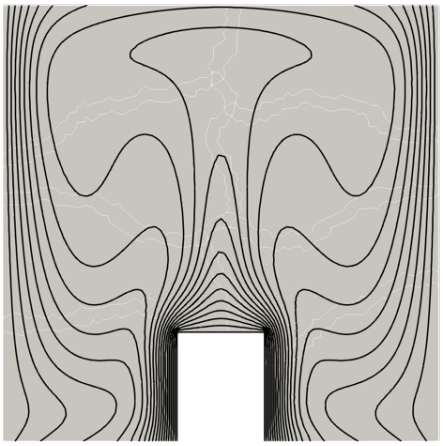
In addition, we run our FEA solver to reproduce a benchmark presented in Aly, Raizah, and Al-Hanaya (2021), and in Kim, Lee, Ha, and Yoon (2008). As shown in Fig. 4.6, this system simulates the natural convection from an inner circular cylinder with a radius $R = 0.2$ located in the center of the square cavity with the length of $L = 1.0$. The boundary conditions are summarized as follows: (1) for the fluids, all the walls have no-slip boundary conditions; (2) for the temperature, the square enclosure is a cold wall, whereas the cylinder enclosure is a hot wall. Fig. 4.6 shows the isothermal contour at different Rayleigh numbers of $\text{Ra} = \left\{10^3, 10^4, 10^5, 10^6\right\}$. Our FEA results show well agreement with those computed using the ISPH method and the FVM.

4.4.2 Design problem setting

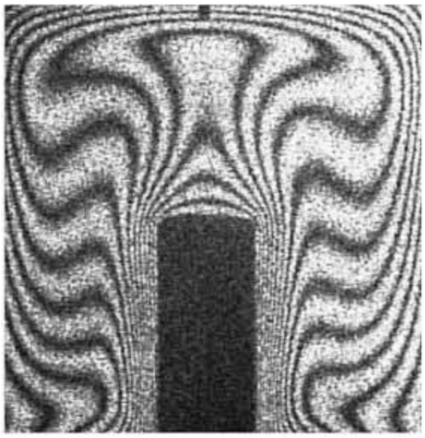
Here, we recall the definition of Grashof number Gr, see Eq. (4.5). It is the ratio between the buoyancy force and viscous force, which describes to what extent the heat transport is dominated by conductive effect or convective effect. A small value of Gr indicates that the fluid motion contributes rather insignificantly to the heat transport, and the heat transport is dominated by the conductive effect in the solid phase. With an increasing value of Gr, the buoyancy force becomes large enough to drive the surrounding fluid, which means that the convective effect starts to dominate the heat transport. In this interval, the fluid flow can converge to a steady state and still remains laminar. If we keep increasing Gr up to a critical value, the natural convection becomes essentially an unsteady-state phenomenon.



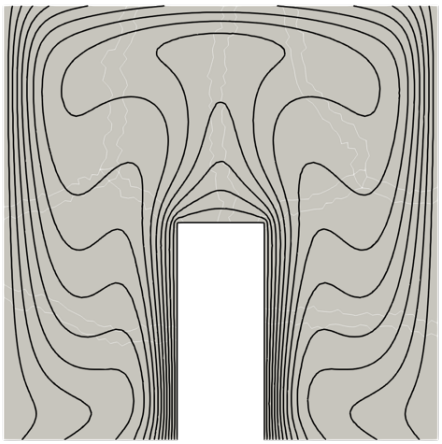
(A) Experimental (Paroncini and Corvaro, 2009).



(B) Numerical simulation.

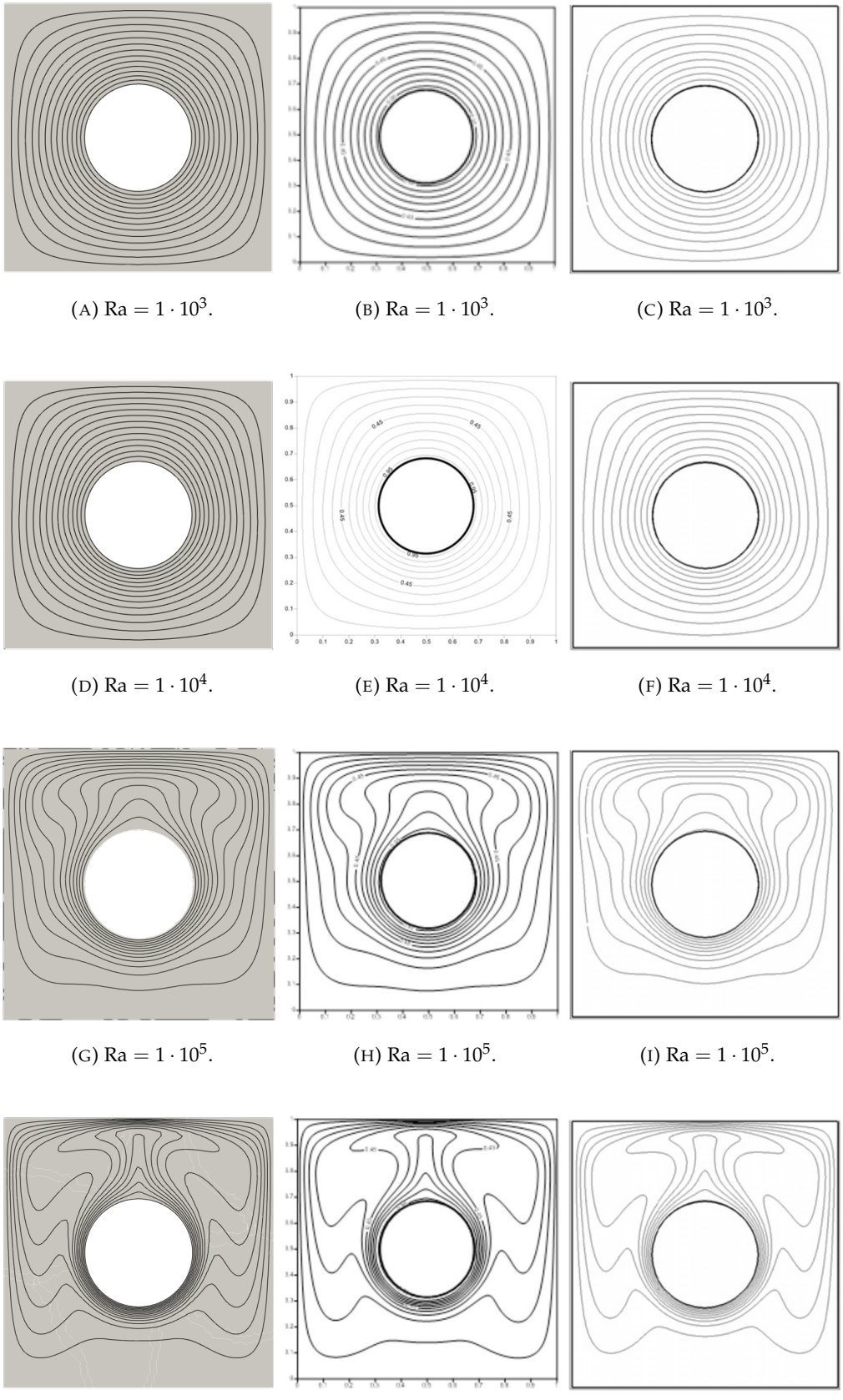


(C) Experimental (Paroncini and Corvaro, 2009).



(D) Numerical simulation.

FIGURE 4.5: Comparison amongst the numerical (right) and the experimental (left) results of Paroncini and Corvaro (2009) for the isothermal contour at Ra $= 2.11 \cdot 10^5$ & $\xi = 0.25$, and Ra $= 2.25 \cdot 10^5$ & $\xi = 0.5$. [(a) and (c) reprinted from Paroncini and Corvaro (2009) with permission from Elsevier.]



(A) $\mathrm{Ra} = 1 \cdot 10^3$. (B) $\mathrm{Ra} = 1 \cdot 10^3$. (C) $\mathrm{Ra} = 1 \cdot 10^3$.

(D) $\mathrm{Ra} = 1 \cdot 10^4$. (E) $\mathrm{Ra} = 1 \cdot 10^4$. (F) $\mathrm{Ra} = 1 \cdot 10^4$.

(G) $\mathrm{Ra} = 1 \cdot 10^5$. (H) $\mathrm{Ra} = 1 \cdot 10^5$. (I) $\mathrm{Ra} = 1 \cdot 10^5$.

(J) $\mathrm{Ra} = 1 \cdot 10^6$. (K) $\mathrm{Ra} = 1 \cdot 10^6$. (L) $\mathrm{Ra} = 1 \cdot 10^6$.

FIGURE 4.6: Comparison amongst the FEA solver (left), the ISPH method of Aly, Raizah, and Al-Hanaya (2021) (middle) and the FVM solver of Kim, Lee, Ha, and Yoon (2008) (right) for the isothermal contour at $\mathrm{Ra} = \{10^3, 10^4, 10^5, 10^6\}$. [figures in the middle and right columns are reprinted from Aly, Raizah, and Al-Hanaya (2021) and Kim, Lee, Ha, and Yoon (2008) with permission from Elsevier.]

However, the critical value of Gr highly depends on the problem at hand, for example, the shape of the cavity and heat sink, boundary conditions, heat generation rate Q_0, reference length scale L, etc. Therefore, it is not easy to predict the critical value of Gr for a particular problem without making the fully transient simulation. With this in mind, before we start to optimize a heat sink layout, the flow behavior under different values of Gr should be carefully examined.

The target problem is an academic benchmark as illustrated in Fig. 4.7. Considering a cavity whose dimension is 1.0×1.0, a uniform body heat source $Q_0 = 10^4$ is applied in the red region ω whose dimension is 0.1×0.05. The blue region is the solid phase Ω_s whose dimension is 0.7×0.5, while the grey region is the fluid phase Ω_f. The boundary conditions are summarized as follows: (1) for the fluid, all walls have no-slip boundary conditions, $\boldsymbol{v} = 0$; (2) for the temperature, the top and side walls are set to cold wall, $T = 0$, while the bottom wall is adiabatic, $\nabla T \cdot \boldsymbol{n} = 0$.

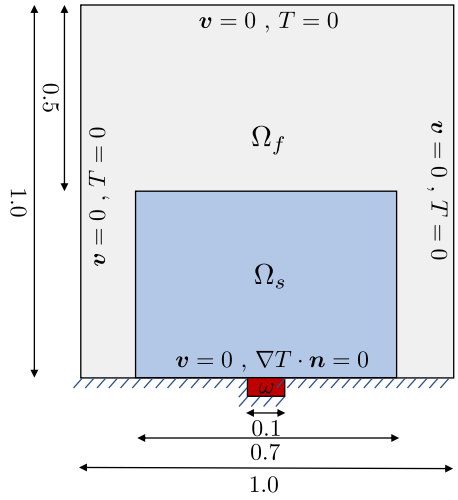


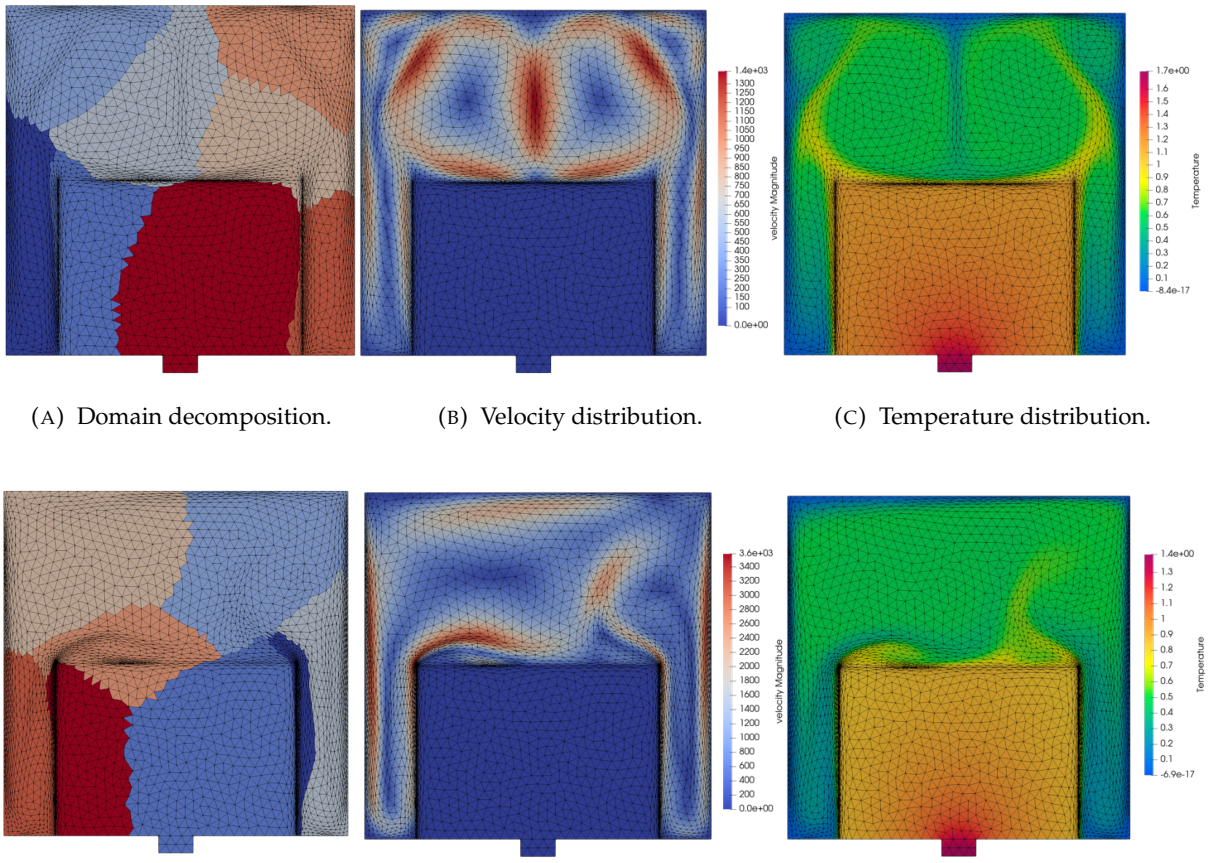
FIGURE 4.7: Schematic of the design problem.

Next, we perform the fully transient simulation. We use the second-order backward difference scheme (BDF2) for time discretization. The finite element modeling and the nested-loop algorithm is described here.

We introduce a dimensionless time by scaling the real time such that $t \rightarrow \frac{V_{\text{ref}}}{L} t$. Then, the unsteady-state natural convection problem can be formulated as follows:

$$\begin{cases} \dfrac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \nabla)\, \boldsymbol{v} - \mathrm{Pr}\, \nabla \cdot \left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^{\mathrm{T}} \right) + \nabla p + \alpha \boldsymbol{v} - \mathrm{Gr}\, \mathrm{Pr}^2\, T \boldsymbol{e}_3 = 0 \\ -\nabla \cdot \boldsymbol{v} = 0 \\ \dfrac{\partial T}{\partial t} + (\boldsymbol{v} \cdot \nabla T) - \nabla \cdot (\kappa \nabla T) - Q = 0. \end{cases} \tag{4.23}$$

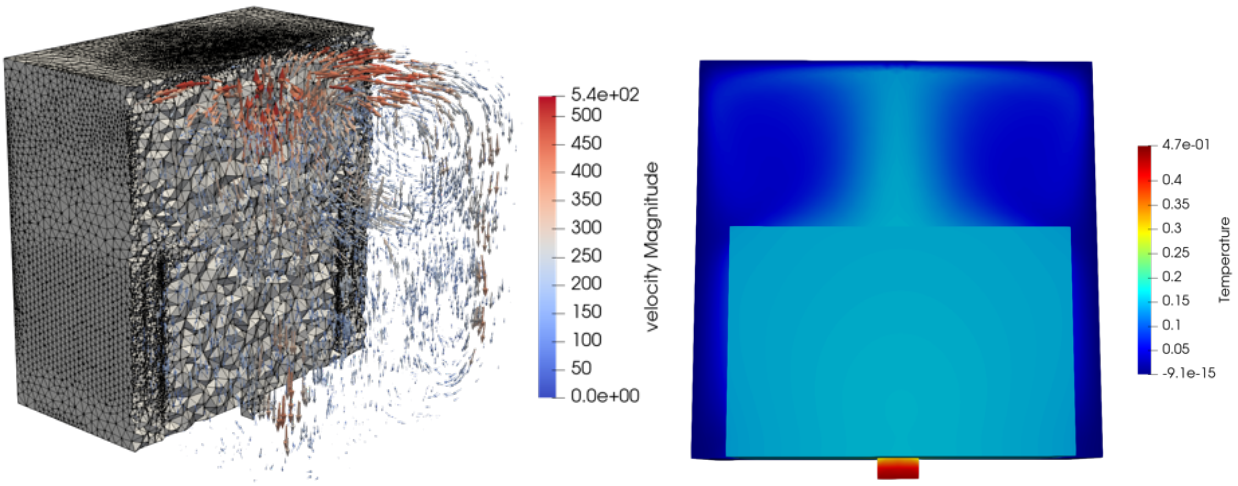
For the time integration, we use a second-order (BDF2) finite difference scheme to advance in time the velocity v and temperature T. Then, the time-dependent terms can be semi-implicitly



(A) Domain decomposition. (B) Velocity distribution. (C) Temperature distribution.

(D) Domain decomposition. (E) Velocity distribution. (F) Temperature distribution.

FIGURE 4.8: 2D transient simulation result for different Grashof numbers: (a)–(c) $\mathrm{Gr} = 10^7$ at time $t = 0.025$ and (d)–(f) $\mathrm{Gr} = 10^8$ at time $t = 0.0994$. Anisotropic finite element mesh and subdomains for the domain decomposition and parallel computing using 8 MPI processes (left column). Velocity distribution (central column). Temperature distribution (right column).



(A) Velocity vectors and cross section view of mesh adaptivity.

(B) Temperature distribution.

FIGURE 4.9: 3D transient simulation result for Grashof number $\mathrm{Gr} = 10^7$ at time $t = 0.05$.

discretized as follows:

$$\frac{d\boldsymbol{v}}{dt} \simeq \frac{3\boldsymbol{v}_{n+1} - 4\boldsymbol{v}_n + \boldsymbol{v}_{n-1}}{2\delta t}, \quad \frac{dT}{dt} \simeq \frac{3T_{n+1} - 4T_n + T_{n-1}}{2\delta t}, \tag{4.24}$$

where δt is the time step. To advance the solution from time t_n to t_{n+1}, we give an initial guess $(\boldsymbol{v}_n, p_n, T_n)$. Then, the weak formulation of the Jacobian and the residual at a given linearization point $(\boldsymbol{v}_{n+1}, p_{n+1}, T_{n+1})$, for an increment $(\delta\boldsymbol{v}_{n+1}, \delta p_{n+1}, \delta T_{n+1})$, are as follows:

$$\begin{aligned} F\left(\boldsymbol{v}_{n+1}, p_{n+1}, T_{n+1}\right) = & \int_\Omega \left(\frac{3\boldsymbol{v}_{n+1} - 4\boldsymbol{v}_n + \boldsymbol{v}_{n-1}}{2\delta t}\right) \cdot \tilde{\boldsymbol{v}} + \left(\left(\boldsymbol{v}_{n+1} \cdot \nabla\right) \boldsymbol{v}_{n+1}\right) \cdot \tilde{\boldsymbol{v}} \\ & + \mathrm{Pr}\left(\nabla\boldsymbol{v}_{n+1} + \nabla\boldsymbol{v}_{n+1}^{\mathrm{T}}\right) : \nabla\tilde{\boldsymbol{v}} \\ & - p_{n+1}\nabla \cdot \tilde{\boldsymbol{v}} + \alpha\boldsymbol{v}_{n+1} \cdot \tilde{\boldsymbol{v}} - \mathrm{Gr}\,\mathrm{Pr}^2\, T_{n+1}\boldsymbol{e}_3 \cdot \tilde{\boldsymbol{v}} - \tilde{p}\nabla \cdot \boldsymbol{v}_{n+1} \\ & + \left(\frac{3T_{n+1} - 4T_n + T_{n-1}}{2\delta t}\right) \tilde{T} + \tilde{T}\left(\boldsymbol{v}_{n+1} \cdot \nabla T_{n+1}\right) \\ & + \left(\kappa\nabla\tilde{T}\right) \cdot \nabla T_{n+1} d\Omega + \int_\omega Q\tilde{T} d\Omega \\ & \forall \left(\tilde{\boldsymbol{v}}, \tilde{p}, \tilde{T}\right) \in \tilde{\boldsymbol{V}} \times \tilde{P} \times \tilde{V} \end{aligned} \tag{4.25}$$

$$
\begin{aligned}
&DF\left(\boldsymbol{v}_{n+1}, p_{n+1}, T_{n+1}\right)\left(\delta\boldsymbol{v}_{n+1}, \delta p_{n+1}, \delta T_{n+1}\right) \\
&= \int_{\Omega_f} \frac{3}{2\delta t}\delta\boldsymbol{v}_{n+1}\cdot\tilde{\boldsymbol{v}} + \left(\left(\delta\boldsymbol{v}_{n+1}\cdot\nabla\right)\boldsymbol{v}_{n+1}\right)\cdot\tilde{\boldsymbol{v}} + \left(\left(\boldsymbol{v}_{n+1}\cdot\nabla\right)\delta\boldsymbol{v}_{n+1}\right)\cdot\tilde{\boldsymbol{v}} \\
&+ \mathrm{Pr}\left(\nabla\tilde{\boldsymbol{v}} + \nabla\tilde{\boldsymbol{v}}^T\right) : \nabla\delta\boldsymbol{v}_{n+1} \\
&- \delta p_{n+1}\nabla\cdot\tilde{\boldsymbol{v}} + \alpha\delta\boldsymbol{v}_{n+1}\cdot\tilde{\boldsymbol{v}} - \mathrm{Gr}\,\mathrm{Pr}^2\,\delta T_{n+1}\boldsymbol{e}_3\cdot\tilde{\boldsymbol{v}} - \tilde{p}\nabla\cdot\delta\boldsymbol{v}_{n+1} \\
&+ \frac{3}{2\delta t}\delta T_{n+1}\tilde{T} + \tilde{T}\left(\delta\boldsymbol{v}_{n+1}\cdot\nabla T\right) + \tilde{T}\left(\boldsymbol{v}_{n+1}\cdot\nabla\delta T_{n+1}\right) + \nabla\tilde{T}\cdot\left(\kappa\nabla\delta T_{n+1}\right) d\Omega \\
&\quad \forall\left(\tilde{\boldsymbol{v}}, \tilde{p}, \tilde{T}\right) \in \tilde{\boldsymbol{V}}\times\tilde{P}\times\tilde{V}.
\end{aligned}
\tag{4.26}
$$

Then, we construct the Newton sequence and solve Eq. (4.27) iteratively:

$$
DF\left(\boldsymbol{v}_{n+1}, p_{n+1}, T_{n+1}\right)\left(\delta\boldsymbol{v}_{n+1}, \delta p_{n+1}, \delta T_{n+1}\right) = F\left(\boldsymbol{v}_{n+1}, p_{n+1}, T_{n+1}\right). \tag{4.27}
$$

At each time step δt (*n-loop*), the governing equations are solved repeatedly within the Newton loop (*k-loop*) until the convergence condition is satisfied (see Algorithm 3).

The anisotropic adaptive mesh refinement introduced in Section 4.3.3 is performed every dimensionless time step. We run two test cases for different Grashof numbers. Fig. 4.8 shows in the left column the adaptive mesh and the 8 subdomains of the domain decomposition used for parallel computing. The middle column shows the velocity distribution and the right column shows the temperature distribution. Figs. 4.8a–c show the testing results for $\mathrm{Gr} = 10^7$ at time $t = 0.025$. The flow converges to a steady state. The velocity and temperature distribution are symmetric with respect to the middle line. Figs. 4.8d–f show the testing results for $\mathrm{Gr} = 10^8$ at time $t = 0.0994$. The results exhibit an asymmetric distribution of the velocity and temperature fields. The flow cannot reach a steady state. Therefore, we may deduce that the critical value of Gr may be between the interval of 10^7 to 10^8. We further test the 3D counterpart cases, and the flow converges to a steady-state solution up until $\mathrm{Gr} = 10^7$, see Fig. 4.9.

Thus, we set $\mathrm{Gr} \leq 10^6$ in all the optimization design cases to ensure that the steady-state assumption can be satisfied to the greatest possible extent. It should be noted that there is still a possibility to encounter the unsteady-state effect during the topological evolution, especially before the fluid–solid surface reaches stability. However, it is always the case when making an assumption.

Besides, the expensive computational cost becomes a critical limit when the fluid flow comes to unsteady state. The time increment δt should be small enough to ease the convergence of nested Newton iterations, see Algorithm 3. The convergence difficulty becomes especially prevalent for higher Grashof numbers, which makes the total runtime of the forward problem untractable for the present TO framework. This requires additional profound investigations and it is out of the scope of this chapter.

Algorithm 3 Nested loops to solve the unsteady-state natural convection problem.

set initial guess $\boldsymbol{w}_0 = (\boldsymbol{v}_n, p_n, T_n)$
  for each *time step* (n-loop) **do**
    for each *Newton iterative step* (k-loop) **do**
      solve Eq. (4.27) to get $(\delta\boldsymbol{v}_k, \delta p_k, \delta T_k)$
      $(\boldsymbol{v}_{k+1}, p_{k+1}, T_{k+1}) = (\boldsymbol{v}_k, p_k, T_k) - (\delta\boldsymbol{v}_k, \delta p_k, \delta T_k)$
      break $\|\boldsymbol{w}_{k+1} - \boldsymbol{w}_k\| < \varepsilon$
    end for
    $(\boldsymbol{v}_{n+1}, p_{n+1}, T_{n+1}) = (\boldsymbol{v}_{k+1}, p_{k+1}, T_{k+1})$
    $(\boldsymbol{v}_{n-1}, p_{n-1}, T_{n-1}) = (\boldsymbol{v}_n, p_n, T_n)$
    $(\boldsymbol{v}_n, p_n, T_n) = (\boldsymbol{v}_{n+1}, p_{n+1}, T_{n+1})$
  end for

TABLE 4.1: The Grashof number Gr, penalty factor q_κ, regularization parameter τ, and volume fraction V_f for 2D and 3D design cases.

Case No.	Gr	q_κ	τ_x, τ_y , τ_z	V_f
2D case #1	1.0×10^1	1/30	$\tau_x = \tau_y = 5.0 \times 10^{-5}$	50%
2D case #2	5.0×10^3	1/30	$\tau_x = \tau_y = 5.0 \times 10^{-5}$	50%
2D case #3	1.0×10^6	1/30	$\tau_x = \tau_y = 5.0 \times 10^{-5}$	50%
2D case #4	1.0×10^1	1/30	$\tau_x = \tau_y = 2.0 \times 10^{-5}$	50%
2D case #5	1.0×10^1	1/30	$\tau_x = \tau_y = 1.0 \times 10^{-4}$	50%
2D case #6	1.0×10^1	1/40	$\tau_x = \tau_y = 2.0 \times 10^{-5}$	50%
2D case #7	1.0×10^1	1/50	$\tau_x = \tau_y = 2.0 \times 10^{-5}$	50%
3D case #1	1.0×10^3	1/30	$\tau_x = \tau_y = \tau_z = 1.0 \times 10^{-4}$	30%
3D case #2	1.0×10^3	1/10	$\tau_x = \tau_y = \tau_z = 1.0 \times 10^{-4}$	30%
3D case #3	1.0×10^3	1/10	$\tau_x = \tau_z = 1.0 \times 10^{-4}, \tau_y = 1.0 \times 10^{-3}$	30%
3D case #4	1.0×10^6	1/30	$\tau_x = \tau_y = \tau_z = 1.0 \times 10^{-4}$	30%
3D case #5	1.0×10^6	1/10	$\tau_x = \tau_y = \tau_z = 1.0 \times 10^{-4}$	30%
3D case #6	1.0×10^6	1/20	$\tau_x = \tau_y = \tau_z = 5.0 \times 10^{-5}$	30%

4.4.3 Design examples

Taking into account all the reasons mentioned in Section 4.4.2, the steady-state assumption is clarified. Therefore, as for the design problem shown in Fig. 4.7, we can expect a symmetric distribution of the velocity and temperature fields with respect to the center line (in 2D) or center plane (in 3D). Hence, we consider a half model (in 2D) or a quarter model (in 3D) with symmetric boundary conditions in the design model. We run our algorithm to solve both two- and three-dimensional benchmarks. In all test cases, we use the same body heat generation $Q_0 = 10^4$, Prandtl number $\mathrm{Pr} = 1.0$. The main purpose of the present study is to examine how the natural convection phenomena will change with the Grashof number Gr. Note that it could be equivalent to vary the heat generation or reference length scale while keeping the Grashof number constant. Another purpose for the numerical study is to examine the effect of penalty factor q_κ and regularization parameter τ on the optimal solutions.

2D examples

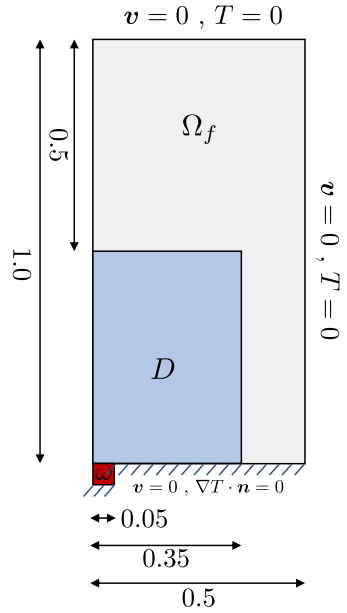


FIGURE 4.10: Design model for 2D test cases.

Fig. 4.10 shows the design model of the 2D test cases. First, we use the same combinations of q_κ and τ, and examine the effects of Grashof number $\mathrm{Gr} = \{10, 5 \times 10^3, 10^6\}$ on the optimal solutions, cf. 2D cases #1–#3. The design results are shown in Fig. 4.11 where the upper row shows the temperature distribution inside the cavity, and the lower row shows the flow velocity. For an easier comparison, the maximum of the temperature scale (upper row) is set to the maximum temperature of these three designs (case #1). To clearly visualize the fluid flow motion, the velocity magnitude (lower row) is scaled to the maximum value of each design case itself.

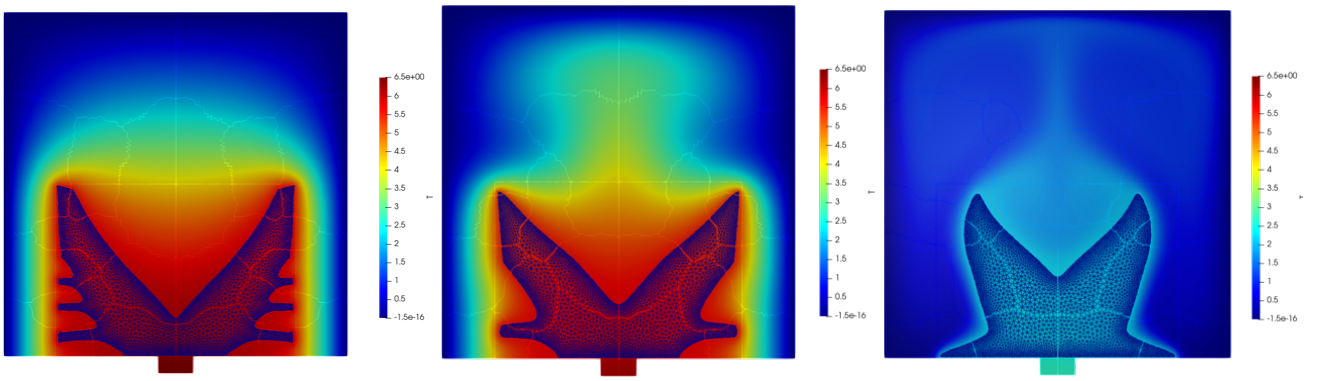
From Fig. 4.11, the following can be observed. (1) For the case of the lowest Grashof Gr $= 10$, the conductive effect plays a dominant role in the heat transportation. The main branches are connected from the heat source to the far end of the design domain, and several small branches are formed towards the side boundaries of the design domain. However, there is no small branch growing toward the top boundary of the design domain. This is quite physically reasonable, because the solid material is placed in the form such that the branches can conduct the heat energy from the heat source towards the nearest cold boundaries. The left and right cold walls are much closer than the top cold boundary. Therefore, the branches tend to grow at both sides. If the height of the cavity is smaller, we can expect the small branches to be formed toward the top boundary. (2) For the case of a higher Grashof Gr $= 5 \times 10^3$, the convective effect starts to play a role but it is still not strong enough to dominate the heat transport. There is only one small branch remaining at each side, and the main branch seems to be thicker. (3) If we keep increasing Grashof number to Gr $= 10^6$, the optimal solution becomes quite different from the former two cases. The branches at both sides disappear, and the main branch becomes curved and more compact. The fluid motion shows two large convection circulations at the side and two small circulations in the middle. Unlike low Grashof number cases, the branches seem not to grow towards the cold walls. Instead, they are shaped to fasten the surrounding flow motion so that the convection can be enhanced.

Next, we test different combinations of regularization parameter τ and the penalty factor q_κ, and examine how they affect the optimal solution. To this end, we set the same Grashof number Gr $= 10$. Fig. 4.12 shows the design results obtained with different values of τ and Fig. 4.13 shows the results with different q_κ. It is clearly reflected in the results that, either a smaller value of τ or a higher value of q_κ can generate a more complicated optimal configuration. A more complicated structure has a better heat conducting performance. Interestingly, the optimal solution seems to favour a needle-like structure as shown in Fig. 4.12a for the conduction-dominated problem. This finding seems to agree with the conclusion by Yan, Wang, and Sigmund (2018). In their work, a volume-to-point pure heat conduction problem was considered, and they proved that the needle-like structure, rather than the tree-like structure, is the optimum for that case.

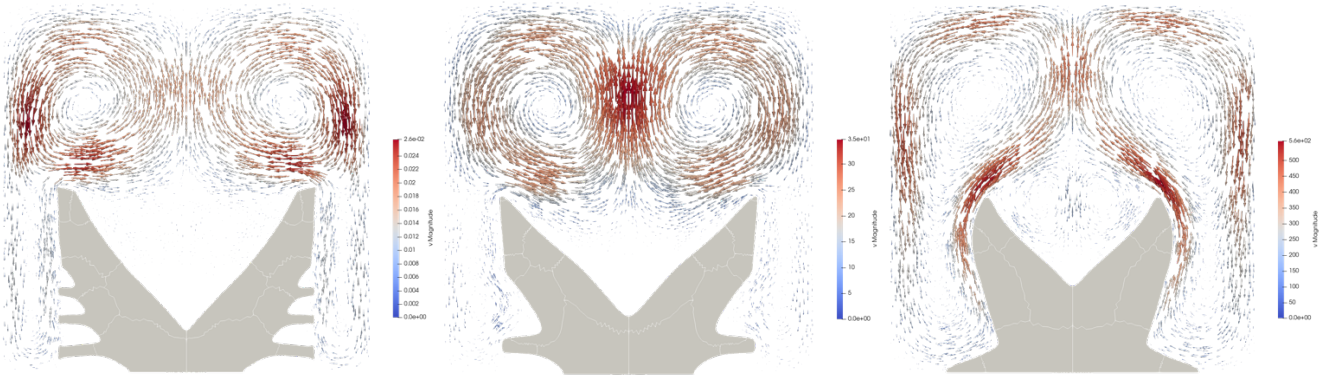
3D examples

Now, we run our algorithm to solve the full-scale 3D benchmark inspired from Alexandersen, Sigmund, and Aage (2016) and Pollini, Sigmund, Andreasen, and Alexandersen (2020). The design model for 3D cases #1–#5 is shown in Fig. 4.14.

Fig. 4.15 shows the topological evolution history of 3D case #2. It can be clearly reflected from the figure that: (1) the fluid and solid phases are natively separated due to the binary structure; (2) the fluid–solid boundary is outlined by the explicit body-fitted mesh associated with an implicit zero-level-set surface; and (3) the entire design domain is initialized with the solid phase, and the



(A) 2D case #1. (B) 2D case #2. (C) 2D case #3.



(D) 2D case #1. (E) 2D case #2. (F) 2D case #3.

FIGURE 4.11: Design results for the 2D cases # 1–#3 under different Grashof numbers. From left to right: Gr $= \{10, 5 \times 10^3, 10^6\}$.Temperature distribution (upper row) and velocity vectors (lower row).

(A) 2D case #4 ($\boldsymbol{J = 156.4}$). (B) 2D case #1 ($J = 161.9$). (C) 2D case #5 ($J = 164.4$).

FIGURE 4.12: 2D design results for the conduction-dominated cases with different values of τ: (a) $\tau = 2 \times 10^{-5}$, (b) $\tau = 5 \times 10^{-5}$, (c) $\tau = 1 \times 10^{-4}$.

(A) 2D case #4 ($\boldsymbol{J = 156.4}$).

(B) 2D case #6 ($J = 160.7$).

(C) 2D case #7 ($J = 162.5$).

FIGURE 4.13: 2D design results for the conduction-dominated cases with different values of q_κ: (a) $q_\kappa = 1/30$, (b) $q_\kappa = 1/40$, (c) $q_\kappa = 1/50$.

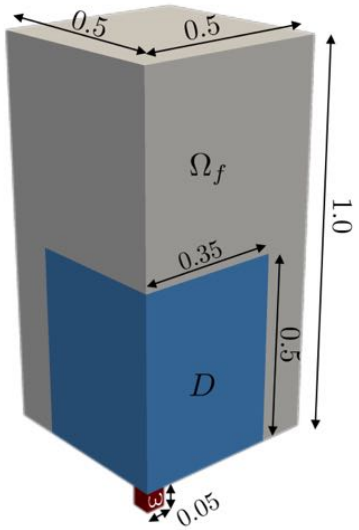


FIGURE 4.14: Design model for 3D cases #1–#5.

inefficient solid material is iteratively removed, in other words, the fluid flow occupies the newly generated voids. These features make clear physical sense considering the fact that to solve a fluid optimization problem is essentially to answer the question of "where the fluid–solid boundaries should be placed".

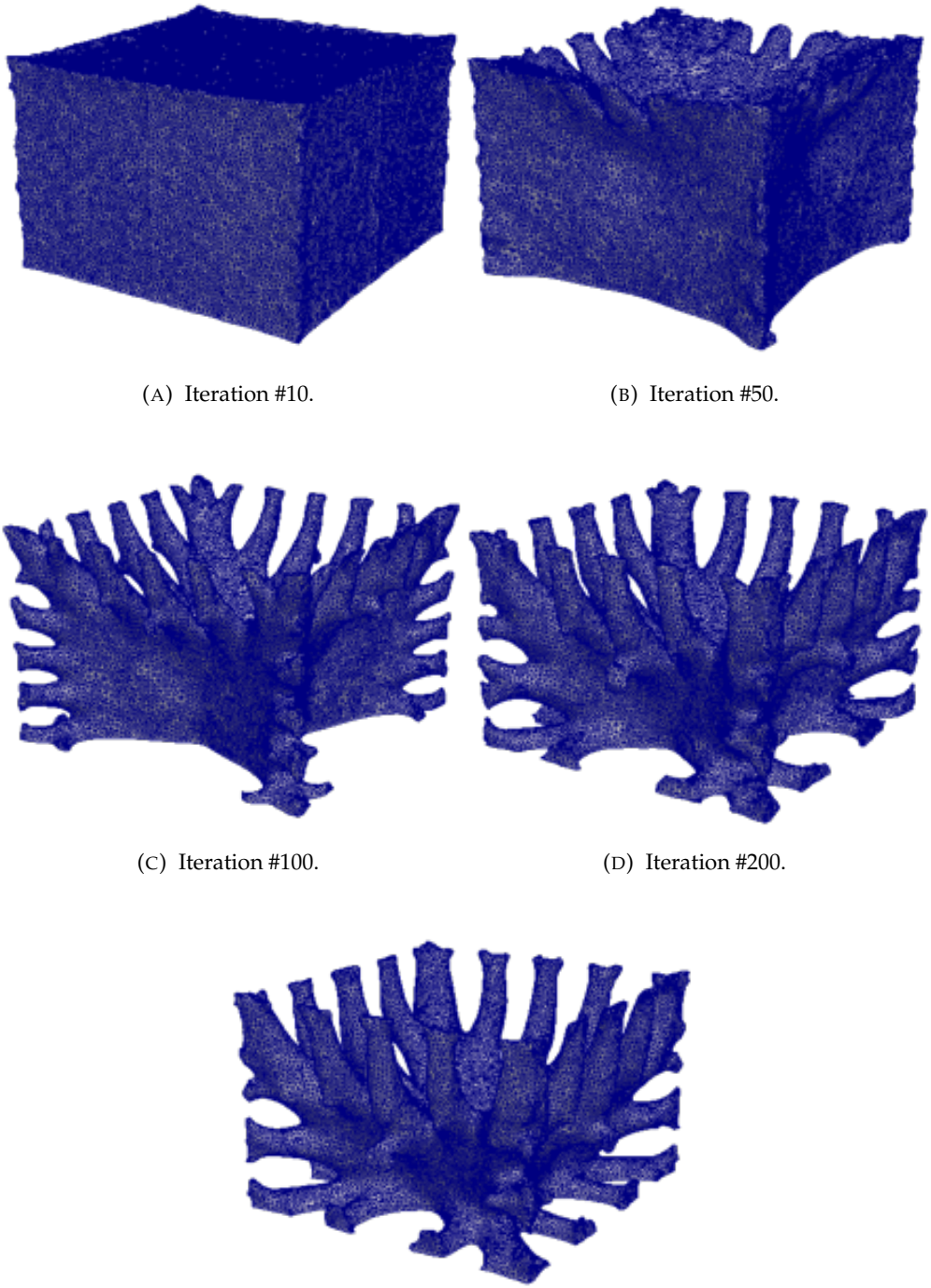
Fig. 4.16 shows the design results for different combinations of Grashof number Gr, penalty factor q_κ, and regularization parameter τ (see 3D cases #1–#5 in Table 4.1). The temperature legend is scaled according to the maximum and minimum temperature values of these five cases. At first glance, the general shapes of all designs are "thermal flowers". However, as for cases #1 and #2 (Figs. 4.16a and b), they look like "blooming flowers" while for cases #4 and #5 (Figs. 4.16d and e), they look like "budding flowers". Such characteristics of the 3D design results are similar to their 2D counterparts. This seems to be physically reasonable, and may be explained intuitively as follows: one the one hand, for the conduction-dominated cases #1 and #2 ($\mathrm{Gr} = 10^3$), the more access to the near side cold walls, the more heat energy can be removed from the heat source. On the other hand, as for the convection-dominated cases #4 and #5 ($\mathrm{Gr} = 10^3$), the overextended branches may disturb the fluid circulation. Instead, we can observe that more compact and thicker primary branches are placed above the heat source.

In addition, Fig. 4.16c shows the design result for 3D case #3 which is obtained by an anisotropic implementation of the regularization parameter τ. We set $\tau_x = \tau_z = 1.0 \times 10^{-4}$ in the x- and z-direction, and $\tau_y = 1.0 \times 10^{-3}$ in the y-direction as the extrusion direction. From the results, we confirmed that the semi-implicit discretization scheme, see Eq. (4.21), can effectively control the geometrical complexity in a specific direction (y-direction in this example).

4.4.4 Thermal performance validation

Here, we will validate the thermal performance of the generative designs. The first step is to convert the generative design to the editable spline representation (B-Rep conversion). As stressed above, the optimal solutions are sampled with the volumetric representation of body-fitted meshes. This makes the B-Rep conversion rather straightforward. We can easily translate the design results from "what we see" to "what we get". Herein, we implement the conversion using a commercial software S-generator (QUINT, 2021). The converted CAD models of 3D cases #2 and #5 are shown in Figs. 4.17a and b. For comparison, we introduce a reference design as shown in Fig. 4.17c which has the same pin-fin volume as 3D cases #2 and #5.

The next step is to verify the results obtained by FreeFEM using an alternative solver. To this end, we model 3D case #5 using the commercial FEA software COMSOL Multiphysics (COMSOL, 2018). This verification procedure should be done prior to the thermal performance crosscheck for the following two reasons. (1) In the TO solver, the fluid and solid phases are modeled in the same system of equation using a fictitious body force term. In the COMSOL solver, we use the separate modeling. It means that the solid domain only participates in the heat transfer computation and is



(A) Iteration #10.

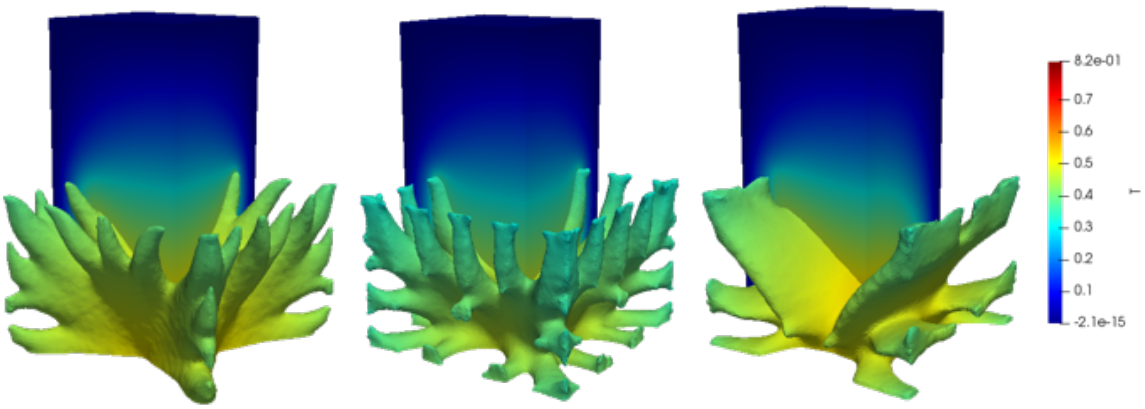
(B) Iteration #50.

(C) Iteration #100.

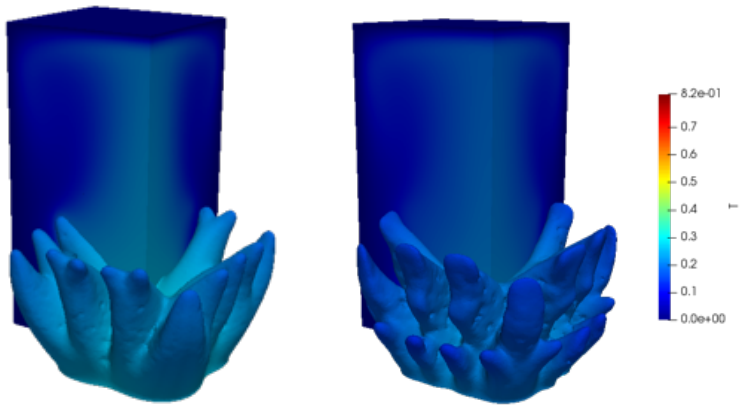
(D) Iteration #200.

(E) Iteration #300.

FIGURE 4.15: Iterative history of the 3D case #2.



(A) 3D case #1. (B) 3D case #2. (C) 3D case #3.



(D) 3D case #4. (E) 3D case #5.

FIGURE 4.16: 3D design results for different combinations of Grashof number Gr, penalty factor q_κ, and regularization parameter τ. (a)–(c): $\mathrm{Gr} = 10^3$, (d) and (e): $\mathrm{Gr} = 10^6$. For an easier comparison, the maximum of the temperature legend scale is set to the maximum temperature of these five designs.

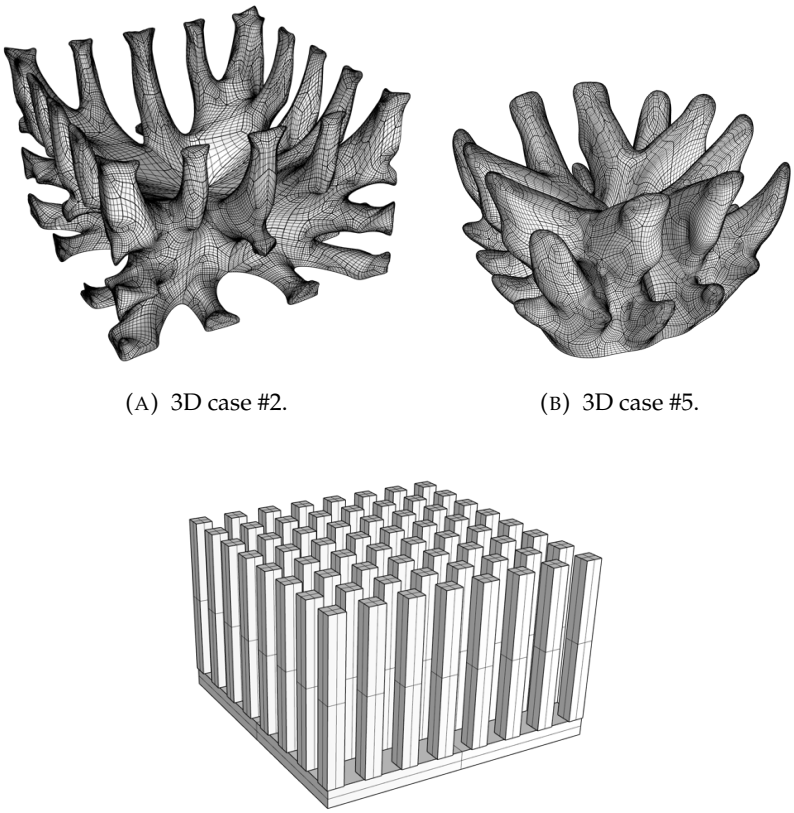
coupled to the surrounding fluid domain though the continuous temperature and flux conditions. (2) As mentioned in Section 4.3.1, the mini-element is used for the space discretization to reduce the overall computational cost. The accuracy of this type of discretization needs to be confirmed in computing the complicated geometry cases.

Fig. 4.18 shows the verification model constructed in COMSOL. The boundary layered mesh is used as shown in the zoomed view of Fig. 4.18b. Fig. 4.19 plots the velocity magnitude and temperature as functions of the arc length of the cut line as shown in Fig. 4.18a. The coordinates of the two vertices are $(0, 0, 0)$ and $(0.5, 0.5, 1.0)$. This line connects between the center point of the heat source and the outer-most vertex of the computational domain. It passes through both fluid and solid phases. The result shows that the FreeFEM solver (using mini-element and fictitious body-force term) is remarkably accurate when compared to those obtained using the COMSOL solver (Taylor–Hood discretization and separate modeling). As for FreeFEM, the velocity magnitude in the solid phase are nearly equal to zero, implying that $\alpha_{\max}$ is sufficient to prevent the fluid penetration. The verification results give us clear evidence that the binary structure expression together with the body-fitted mesh adaptation mimics the fluid–solid boundary effect quite well.

Finally, we perform the crosscheck, that is, the generative designs are analyzed under the different Grashof numbers and the design optimized for a certain Grashof number should preferably perform better than the others for its particular Grashof number. As can be seen from Fig. 4.20, the designs perform as they should. Moreover, the generative designs provide better thermal performance compared with the reference design. This finding seems to agree with the conclusion by Baldry, Timchenko, and Menictas (2019) that the removal of central pins can facilitate the use of natural convection.

4.4.5 Computational cost

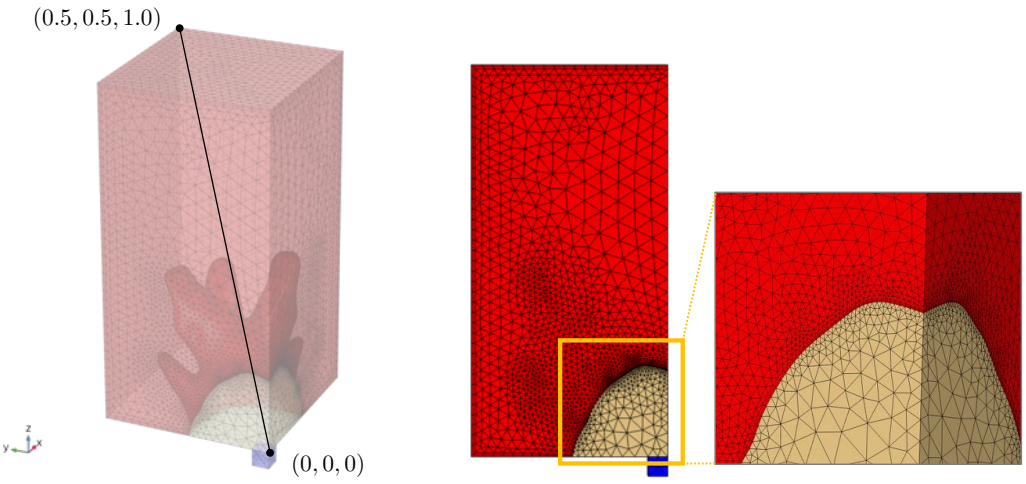
The last example in this chapter is a suspended heated cylinder, which is inspired by Pollini, Sigmund, Andreasen, and Alexandersen (2020). The aim of this subsection is to demonstrate the parallel efficiency and the required computational resource of the proposed framework. The design model is shown in Fig. 4.21. Considering a cavity whose dimension is $1.0 \times 1.0 \times 1.0$, a uniform body heat source $Q_0 = 10^3$ is applied in the suspended cylindrical subdomain ω (in red). The blue region is the design domain and the grey region is the fluid phase Ω_f which is a non-design domain. The boundary conditions are summarized as follows: (1) for the fluid, all walls have no-slip boundary conditions, $\boldsymbol{v} = 0$; (2) for the temperature, the top and side walls are assumed as cold wall, $T = 0$, while the bottom wall is adiabatic, $\nabla T \cdot \boldsymbol{n} = 0$. A quarter of the full model is used and symmetric boundary conditions are imposed. In this test case, the Grashof number is set to $\mathrm{Gr} = 10^6$. The design domain is initialized with a solid-only phase.



(A) 3D case #2. (B) 3D case #5.

(C) Reference design.

FIGURE 4.17: B-Rep conversion of the 3D heat sink.



(A) Cut line. (B) Boundary layered mesh in COMSOL solver.

FIGURE 4.18: Verification model for 3D case #5.

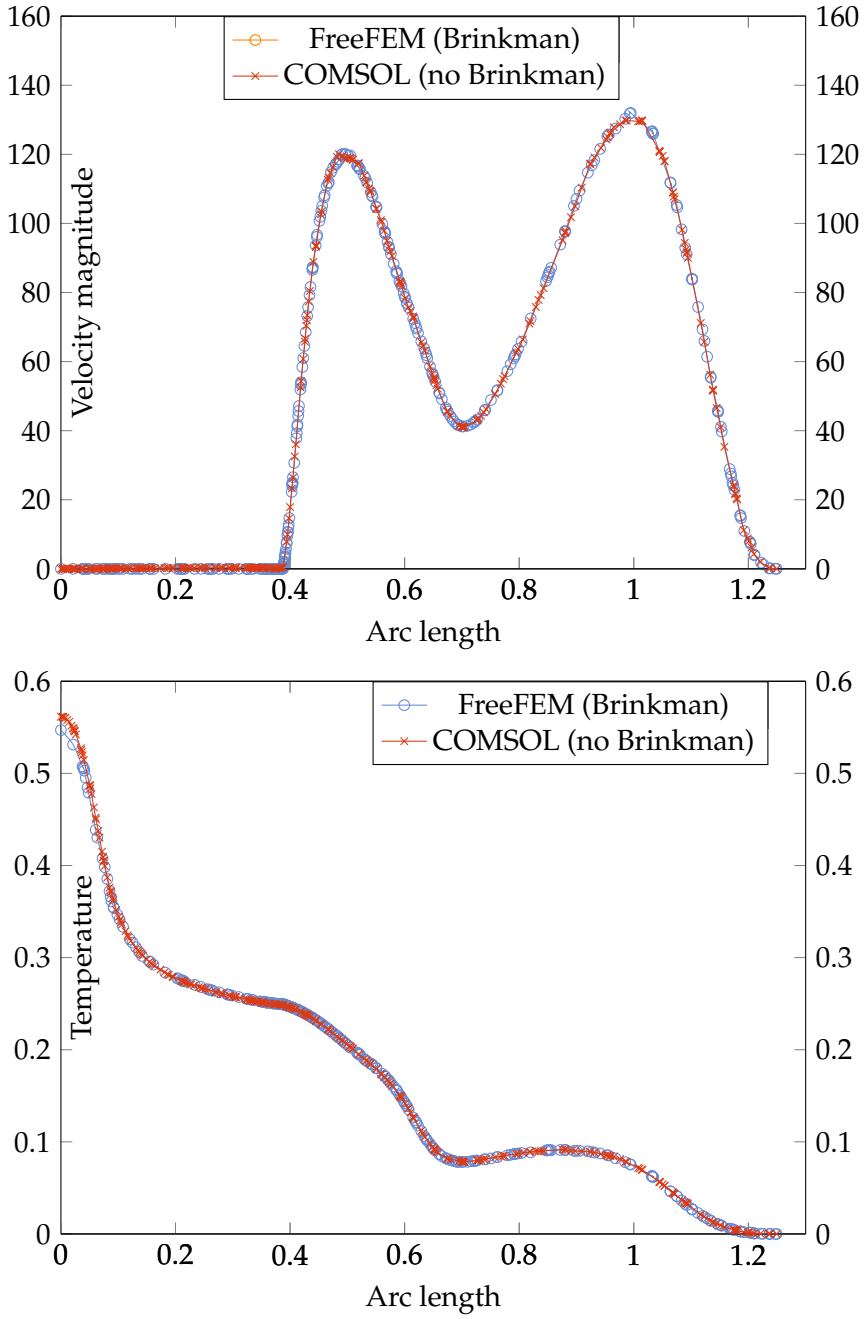
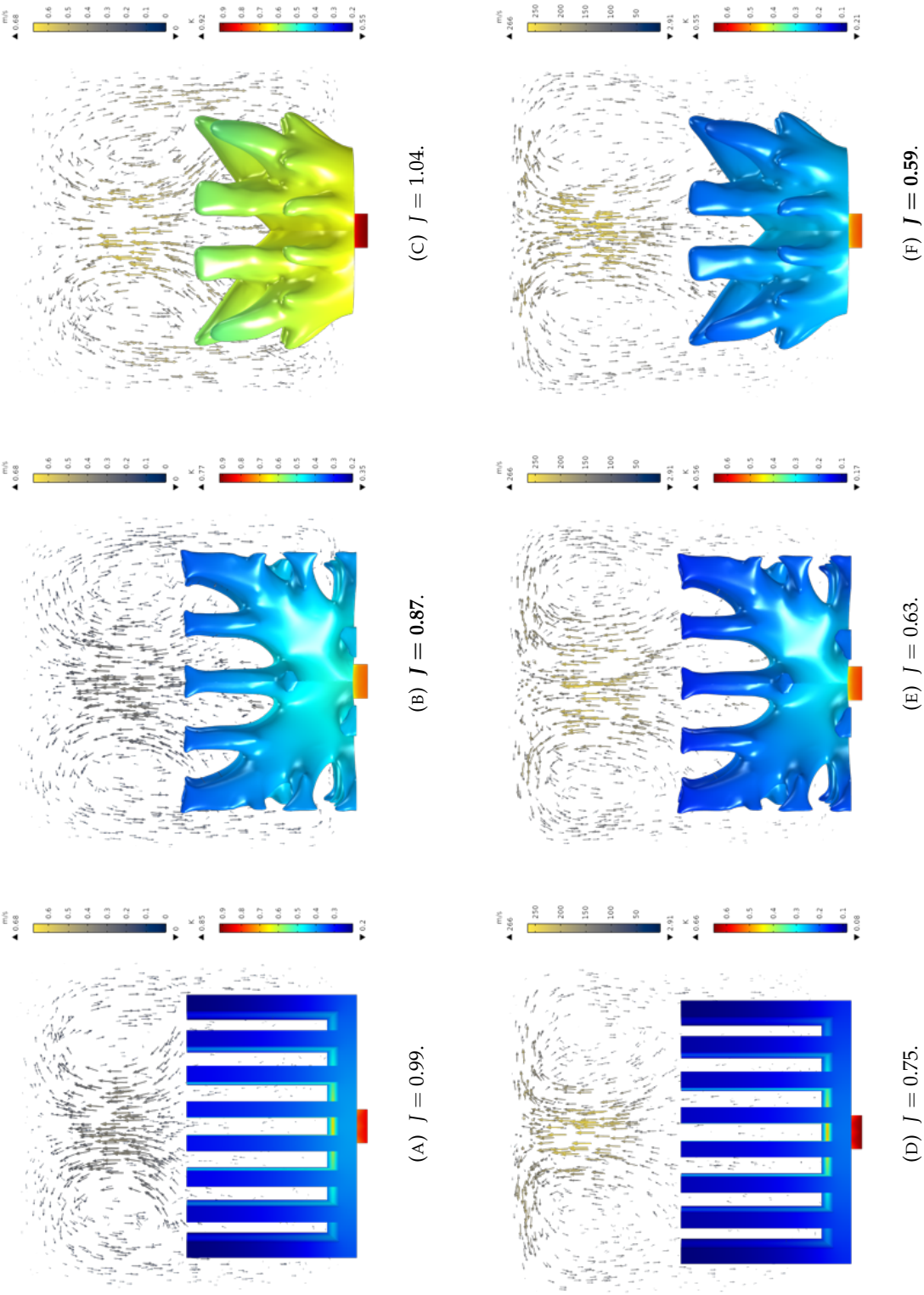


FIGURE 4.19: Velocity magnitude and temperature as a function of arc length of the cut line shown in Fig. 4.18a.



(A) $J = 0.99$. (B) $\boldsymbol{J = 0.87}$. (C) $J = 1.04$.

(D) $J = 0.75$. (E) $J = 0.63$. (F) $\boldsymbol{J = 0.59}$.

FIGURE 4.20: Crosscheck for the thermal performance.

Figs. 4.22a–c show different views of the optimal solution. The temperature field and the velocity profile are depicted in Figs. 4.22d and e, respectively. From the design result, it can be observed that the conductive (solid) branches are split into upper and lower parts. The optimal configuration is seen to have several short branches distributed around the middle section. Although the design details are different from those given in Pollini, Sigmund, Andreasen, and Alexandersen (2020), one can say that the general configurations are similar. As has been discussed in the previous subsections, we can expect different optimal shapes if we use different combinations of penalty factor q_κ and regularization parameter τ. Therefore, these issues will not be furthered here.

For this test case, the number of tetrahedra used at the last iteration is $9.16 \cdot 10^5$, having approximately $3.56 \cdot 10^6$ degrees of freedom for the linearized fluid system. The time breakdown of the finite element actions performed at the last iteration are summarized in Table 4.3. The total runtime for 300 iterations is approximately 41 h 23 min. It is worth noting that the nonlinear Newton solver starts from a good initial solution (solution from the previous iterative step). Usually, no more than three nonlinear iterations are required for most of the iterative steps to reach the Newton tolerance.

From the time breakdown, it can be observed that the most time consuming task is the solution of the governing equations. The second most time consuming action is that of the body-fitted mesh evolution. Since this part of the solver is performed sequentially on a single MPI process, it is indeed one of the bottlenecks in the current framework. This issue will be addressed once ParMmg (recently developed by Cirrottola and Froehly (2021)), is able to handle implicit surface remeshing.

Here, we clarify the choice between "separate" and "hybrid" modeling strategies. The hybrid modeling strategy, relying on the fictitious body force term to mimic an immersed solid phase in an integrated domain, has its drawbacks, especially for those problems where the pressure diffusion inside the solid phase is problematic. For example, for the fluid–structure interaction (FSI) topology optimization problem, the optimal shape highly depends on the pressure difference between the enclosed voids and the external fluid flow. In other words, FSI problems should be essentially modeled by separate governing equations. However, using the separate modeling strategy, we initialized the flow field by the solution of the Stokes flow, then we repeatedly solve the Newton iteration for a number of Reynolds numbers until the target Reynolds. see our recent work on FSI optimization problems (Li, Kondoh, Jolivet, Furuta, Yamada, Zhu, Izui, and Nishiwaki, 2022). Except in these specific cases, in general, the introduction of the fictitious body-force term is not a big issue for other fluid-based optimization problems. Considering all the aforementioned reasons, the hybrid modeling strategy is used in the present work.

Finally, we compare the computational techniques to those used in the state-of-the-art works on the topology optimization of three-dimensional thermal–fluid problems. From Table 4.2, some findings are summarized as follows:

1. FEA is the most widely used numerical analysis method with 6 works or 60% (Yaji, Yamada, Kubo, Izui, and Nishiwaki, 2015; Alexandersen, Sigmund, and Aage, 2016; Pollini, Sigmund, Andreasen, and Alexandersen, 2020; Kambampati and Kim, 2020; Sun, Liebersbach, and Qian, 2020; Feppon, Allaire, Dapogny, and Jolivet, 2020a), owing to the maturity of the stabilized FEM. On the other hand, we can see increasing efforts on FVM-based framework considering the fact that FVM is more preferred in the field of computational fluid dynamics (CFD) (Dilgen, Dilgen, Fuhrman, Sigmund, and Lazarov, 2018b; Yu, Ruan, Gu, Ren, Li, Wang, and Shen, 2020; Sun, Ghandhi, and Qian, 2020).

2. Regarding the choice of numerical tool, recent works tend to use open-source software packages with 4 works or 40% (Sun, Liebersbach, and Qian, 2020; Sun, Ghandhi, and Qian, 2020; Feppon, Allaire, Dapogny, and Jolivet, 2020a; Yu, Ruan, Gu, Ren, Li, Wang, and Shen, 2020). Unlike the underlying coding languages (Alexandersen, Sigmund, and Aage, 2016; Yaji, Ogino, Chen, and Fujita, 2018; Dilgen, Dilgen, Fuhrman, Sigmund, and Lazarov, 2018b; Pollini, Sigmund, Andreasen, and Alexandersen, 2020; Kambampati and Kim, 2020), open-source software allow one to easily implement multi-physics models using a high-level language. Strong meshing and parallel capabilities make it possible to solve large-scale three-dimensional problems.

3. The density-based framework outnumbers the LSM-based one, with 7 works or 70% (Alexandersen, Sigmund, and Aage, 2016; Yaji, Ogino, Chen, and Fujita, 2018; Dilgen, Dilgen, Fuhrman, Sigmund, and Lazarov, 2018b; Pollini, Sigmund, Andreasen, and Alexandersen, 2020; Sun, Liebersbach, and Qian, 2020; Sun, Ghandhi, and Qian, 2020; Yu, Ruan, Gu, Ren, Li, Wang, and Shen, 2020). The other 3 works use LSM (Yaji, Yamada, Kubo, Izui, and Nishiwaki, 2015; Kambampati and Kim, 2020; Feppon, Allaire, Dapogny, and Jolivet, 2020a). Of these 3 works, one uses adapted surface capturing technique through body-fitted meshes (Feppon, Allaire, Dapogny, and Jolivet, 2020a).

4. Seven works use multi-node clusters to run their algorithms. Three works perform computations using single multiprocessor workstations (Yaji, Yamada, Kubo, Izui, and Nishiwaki, 2015; Yu, Ruan, Gu, Ren, Li, Wang, and Shen, 2020; Feppon, Allaire, Dapogny, and Jolivet, 2020a).

5. Regarding sensitivity analysis, the continuous adjoint method outnumbers the discrete adjoint method with 6 works or 60% (Yaji, Yamada, Kubo, Izui, and Nishiwaki, 2015; Yaji, Ogino, Chen, and Fujita, 2018; Sun, Liebersbach, and Qian, 2020; Sun, Ghandhi, and Qian, 2020; Yu, Ruan, Gu, Ren, Li, Wang, and Shen, 2020; Feppon, Allaire, Dapogny, and Jolivet, 2020a). On the one hand, for laminar flows, by solving the adjoint equations which are linear PDEs (in most of the cases), the sensitivity analysis can be more efficient than the discrete adjoint method. On the other hand, for more complicated two–equation turbulent flow models, e.g., $k-\epsilon$ or $k-\omega$ RANS model, automatic differentiation will be required.

TABLE 4.2: Computational techniques in the state-of-the-art works on the topology optimization of three-dimensional thermal fluidic problem.

Ref.	Problem-at-hand	Flow[a]	Model	Numer. Analy.	TO	Adjoint	Element Num.	Precond.[b]	Mesh Type[c]	Process Num.	Numer. tool	Total runtime[d]
Yaji, Yamada, Kubo, Izui, and Nishiwaki (2015)	FC HS[e]	I/S/L	NS[f]	FEM	RDE-LSM[g]	continuous	n/a	n/a	fixed tetra.	n/a	COMSOL	n/a
Alexandersen, Sigmund, and Aage (2016)	NC HS[h]	I/S/L	NS	FEM	density	discrete	$8.19 \cdot 10^6$	GMG[i]	fixed hex.	1,280	PETSc	10 h
Yaji, Ogino, Chen, and Fujita (2018)	FC HS	I/U/L	NS	LBM[j]	density	continuous	$8 \cdot 10^5$	n/a	fixed grid	128	n/a	13.3 h
Dilgen, Dilgen, Fuhrman, Sigmund, and Lazarov (2018b)	FC HS	I/S/T	RANS[k]	FVM	density	discrete	$2.16 \cdot 10^5$	n/a	fixed hex.	100	PETSc	10 h
Pollini, Sigmund, Andreasen, and Alexandersen (2020)	NC HS	I/S/L	NS–Darcy	FEM	density	discrete	$8.19 \cdot 10^6$	GMG	fixed hex.	500	PETSc	8 h 39 min
Kambampati and Kim (2020)	FC HS	I/S/T	Darcy	FEM	HJE-LSM[l]	n/a	$1.02 \cdot 10^6$	n/a	fixed hex.	n/a	n/a	n/a
Sun, Liebersbach, and Qian (2020)	FC HS	I/S/L	NS	FEM	density	continuous	$6 \cdot 10^6$	PCD[m]	fixed tetra.	96	FeniCS	14 d
Feppon, Allaire, Dapogny, and Jolivet (2020a)	FC HS	I/S/L	NS	FEM	HJE-LSM	continuous	$6.17 \cdot 10^5$	mAL[n]	adapt. tetra.	24	FreeFEM	3 d
Yu, Ruan, Gu, Ren, Li, Wang, and Shen (2020)	FC HS	I/S/L	NS	FVM	density	continuous	$5.16 \cdot 10^5$	n/a	fixed hex.	20	OpenFOAM	3.5 h
Sun, Ghandhi, and Qian (2020)	mixer	I/S/L	NS	FVM	density	continuous	$4.5 \cdot 10^6$	n/a	fixed hex.	1,024	OpenFOAM	15 h
This work	NC HS	I/S/L	NS	FEM	RDE-LSM	continuous	$9.16 \cdot 10^5$	mAL	adapt. tetra.	32	FreeFEM	41 h 23 min

[a] I: incompressible; S: steady-state; U: unsteady-state; L: laminar; T: turbulent. E.g., I/S/L: incompressible steady-state laminar flow.
[b] Precond.: preconditioner
[c] Mesh type: fixed (resp. adaptive) tetraheral (resp. hexahedral) elements.
[d] We select the benchmark from each work which are most comparable to the 3D test case #6 in our work.
[e] FC HS: forced convection heat sink.
[f] NS: full Navier–Stokes.
[g] RDE-LSM: reaction–diffusion equation based level-set method.
[h] NC HS: natural convection heat sink.
[i] GMG: Galerkin-projection geometric multigrid preconditioner.
[j] LBM: lattice Boltzmann method.
[k] RANS: Reynolds-averaged Navier–Stokes.
[l] HJE-LSM: Hamilton–Jacobi equation-based level-set method.
[m] PCD: pressure convection diffusion preconditioner.
[n] mAL: modified augmented Lagrangian preconditioner.

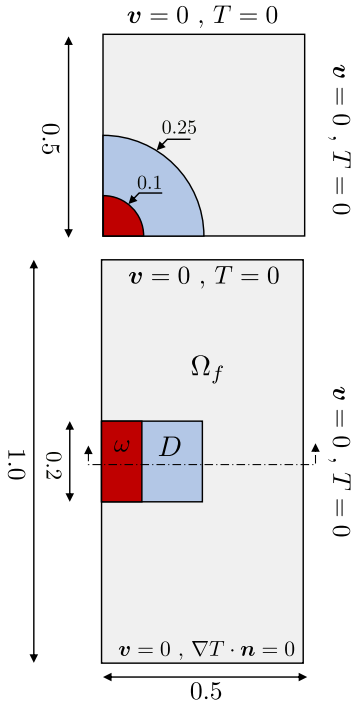


FIGURE 4.21: Design model for 3D cases #6.

4.5 Summary

This chapter presents a novel framework for two- and three-dimensional topology optimization of natural convection problems. The main idea is to incorporate body-fitted mesh adaptation into the RDE-based level-set method. The fluid and solid phases are described using a binary structure. Together with the surface-capturing schemes, this accurate binary description can be transferred to the CAD model for simulation and manufacturing, in a straightforward manner. This feature further exploits the potential of LSM in the thermal–fluid topology optimization.

To demonstrate the complete workflow of the proposed methodology, first, we recall the high-fidelity thermal–fluid modeling where the full Navier–Stokes equations are strongly coupled with the convection–diffusion equation. Secondly, the optimum design problem is formulated and the adjoint equations are derived. Thirdly, we present the numerical implementation details for the finite element modeling, parallel computing, and the two types of adaptive mesh strategies. Then, we carefully examine the flow behavior under different Grashof number, and clarify the steady-state assumption. Finally, we run our algorithm to solve both 2D and 3D test cases. The main findings of this work can be concluded as follows.

1. The suggested RDE-based LSM uses the Ersatz material approach. The nonlinear RAMP scheme is employed for material interpolation to prevent bad quality local optima. The binary structure expression allows us to avoid the use of the continuation approach. The

penalty parameters can be used to control the geometrical complexity of the optimal configuration.

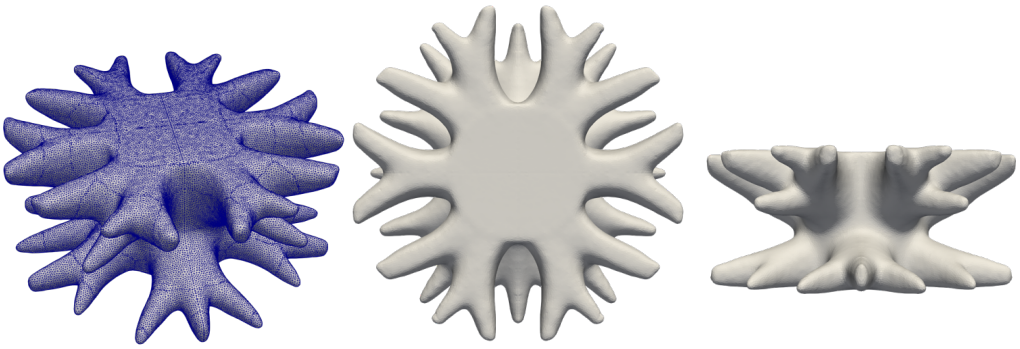
2. The generative designs represented by the body-fitted mesh are converted into editable CAD models, and they are imported to the COMSOL simulation solver for performance validation. The simulation results agree well with the optimization solver, confirming the accuracy of the hybrid modeling strategy. The crosscheck results show that the design optimized for a certain Grashof number preferably performs better than the others for its particular Grashof number. Moreover, the TO designs perform better than the reference design, which further validates the effectiveness of the suggested design methodology.

3. The proposed TO framework is used with the mesh adaptation software Mmg and with PETSc, suitably interfaced in FreeFEM, to deliver full-scale three-dimensional designs. A moderately large-scale 3D problem, having approximately $3.56 \cdot 10^6$ degrees of freedom for the linearized fluid system, is computed with 32 MPI processes.

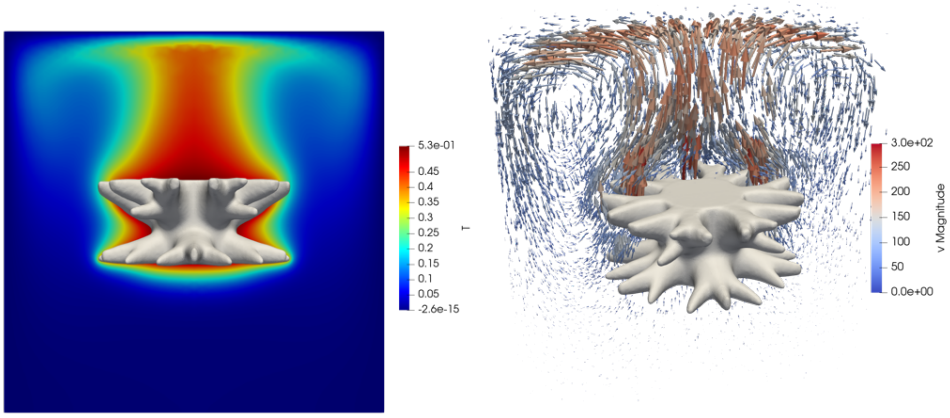
We intend to conduct future research from the following aspects. (1) Fig. 4.23 shows a resin-based 3D printed prototype based on 3D case #2. We will fabricate the metal-based prototype (using 3D printing or five-axis machine tool) and further validate the thermal performances through experimental investigations. (2) The problems with a higher Grashof number are targeted in our future contribution, but they require a fully-transient optimization process. (3) It is worth further investigating on which mesh adaptation strategy (body-fitted or anisotropic mesh) is more preferred for the proposed TO framework in terms of the computational efficiency and accuracy. A pure fluid optimization problem such as lift–drag problems could be used as a benchmark for the further discussion. (4) The present topology optimization framework can be also adapted to solve internal flow problems such as micro-fluid mixing, forced convection heat sink, fluid-to-fluid heat exchanger, etc.

TABLE 4.3: Time breakdown of the finite element actions performed at the last iteration of the 3D case #6, cf. Fig. 4.22.

Action.	Runtime (s)
Solve governing equations	392.9
Solve adjoint equations	46.5
Compute Lagrange multiplier and sensitivity	0.45
Solve RDE	1.04
Visualization output	0.97
Centralize solution to process #0	0.34
Mmg body-fitted mesh evolution (sequential)	87.3
Partition updated mesh	6.75
Interpolate solution to updated mesh	0.76
Total	537



(A) Isometric view. (B) Top view. (C) Front view.



(D) Temperature distribution. (E) Velocity distribution.

FIGURE 4.22: Design result of the suspended cylinder case #6.

FIGURE 4.23: Resin-based 3D printed prototype, cf. 3D case #2.