Level Set Topology Optimization with Meshfree Methods for Design-Dependent Multiphysics Problems

Andreas Neofytou

School of Engineering

Cardiff University

Supervised by
Prof. Hyunsun Alicia Kim

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy (Ph. D.)

July 2021

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. H. Alicia Kim for giving me the amazing opportunity for this PhD project. Her valuable guidance and inspirational advice, constant support, and the strong encouragement to seize every opportunity helped me advance both as a researcher and a person. The knowledge and skills I acquired under her supervision will accompany me for the rest of my life.

To my Cardiff-based colleagues and friends Renato Picelli, Scott Townsend, Xiaoyi Zhou and Sheng Chu with whom I had the pleasure to share an office during this project. For their help and support, and for all the great times we shared both inside and outside the office.

To my San Diego (UCSD)-based colleagues and friends Carolina Jarequie, Douglas de Aquino Castro, Zongliang Du, Sandilya Kambampati, Lei Li, Hayoung Chung, Jaeyub Hyun , Eliana Bortot, Hesaneh Kazemi. For the endless support and for their unbelievable hospitality and all the fun times we shared during my six-month visit to UCSD, California.

Many thanks also to Prof. Jiun-Shyan Chen at UCSD for providing his great expertise, help and valuable advice during this project.

A great thank you to my friend Dr. Tsung-Hui Huang for his solid contributions to this project and for the creative and inspiring discussions we had during our coffee breaks and lunches at UCSD.

I would also like to thank Prof. Lucy T. Zhang and her amazing team for the opportunity to spent two weeks with them at the Rensselaer Polytechnic Institute (RPI) in New York and for their great help to my subsequent work.

I would also like to gratefully acknowledge the support of the Engineering and Physical Sciences Research Council for funding this research.

Last but not least, my deepest gratitude and love to my family to whom I owe everything.

Abstract

This thesis aims to develop computational tools for multiphysics problems in topology optimization with particular focus on design-dependent surface physics. This is a challenging class of problems governed by interface conditions and loadings. Fluid-structure interaction (FSI) is a typical example. Level set topology optimization (LSTO) possesses an advantage over traditional density-based methods since it provides clearly defined boundaries. However, maintaining this crisp boundary representation onto the computational model for the analysis is not straightforward especially with fixed grid methodologies. On the other hand, remeshing the structure at each iteration directly addresses the problem, it poses additional difficulties because of the need to ensure good quality meshes at each iteration. In this thesis a meshfree level set topology optimization methodology based on the reproducing kernel particle method (RKPM) is developed to ensure the well-defined geometrical representation of the structural boundary is transferred onto the computational domain by placing RKPM particles along the boundary. In this way, the difficulties associated with fixed grid LSTO methods and remeshing-based approaches are avoided.

The methodology is first validated for purely hydrostatic pressure, which is a very simple case of design-dependent physics. The obtained results are validated through comparison with the literature. Different integration schemes, particle distributions and continuity orders are also explored to pin down the best balance between accuracy and efficiency.

The development of the LSTO-RKPM methodology is later extended to fluid-structure interactions. To accomplish this, the LSTO-RKPM methodology is further combined with the modified immersed finite element method (mIFEM). The coupling of the different methods is illustrated through the analysis of transient FSI examples. For the optimization, the simplified case of steady-state FSI is assumed. The applicability of the methodology is illustrated through examples and compared with the literature. For the sensitivity analysis, a particle-based discrete adjoint methodology for the level set topology optimization method is presented. Additionally, an algorithm for identifying and removing free-floating volumes of solid material into the fluid domain is explained.

List of Publications and Conferences

Journal Papers

- A. Neofytou, T-H Huang, S. Kambampati, R. Picelli, J-S Chen and H. A. Kim (2021), Level set topology optimization with nodally integrated reproducing kernel particle method, Computer Methods in Applied Mechan-ics and Engineering (CMAME), accepted for publication.
- A. Neofytou, R. Picelli, T-H Huang, J-S Chen and H. A. Kim (2020), Level set topology optimization for design-dependent pressure loads using the reproducing kernel particle method, Structural and Multidisciplinary Optimization (SAMO), 61: 1805-1820.
- R. Picelli, A. Neofytou and H. A. Kim (2019), Topology optimization for design-dependent hydrostatic pressure loading via the level set method. Structural and Multidisciplinary Optimization (SAMO), 60: 1313-1326.

In preparation

- A. Neofytou, F. Yu, L. T. Zhang and H. A. Kim, Level set topology optimization for fluid-structure interactions.

Conference Presentations

- A. Neofytou, R. Picelli and H. A. Kim, Level set topology optimization for fluid-structure interaction, ECCM - ECFD, Glasgow, UK, 2018.
- A. Neofytou, R. Picelli, J-S Chen and H. A. Kim, Level Set Topology Optimization for Design Dependent Pressure Loads: A Comparison Between FEM and RKPM, AIAA Aviation, Dallas, Texas, 2019.
- A. Neofytou, R. Picelli, J-S Chen and H. A. Kim, Level set topology optimization for design-dependent hydrostatic loading using the reproducing kernel particle method, ASME-IDETC, Anaheim, CA, USA, 2019.

- A. Neofytou, F. Yu, L. T. Zhang and H. A. Kim, Level set topology optimization for fluid-structure interactions, AIAA SciTech, Nashville, Tennessee (virtual event), 2021.

- A. Neofytou, F. Yu, L. T. Zhang and H. A. Kim, Level set topology optimization for fluid-structure interactions, WCSMO-14 (virtual event), 2021.

- A. Neofytou, F. Yu, L. T. Zhang and H. A. Kim, Level set topology optimization for fluid-structure interactions, ICTAM, Milan, Italy (virtual event), 2021.

- A. Neofytou, F. Yu, L. T. Zhang and H. A. Kim, Invited Keynote Lecture: Level set topology optimization for fluid-structure interactions, USNCCM16 (virtual event), 2021.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview of structural optimization

Structural optimization is a procedure that seeks for the best, or else the optimum, structure in terms of a performance measure by changing the structure's parame-ters. Some examples of performance measures include the weight, stress or stiffness of a structure, the crashworthiness of a vehicle, the aerodynamic efficiency of an aircraft etc. whereas design parameters, also known as design variables, are the variables that affect the performance and can be altered during the design procedure. The dimensions of a structure such as the thickness and cross-sectional area, or the geometric shape of a structure are some examples of design variables. Material properties such as Young's modulus can also be design variables. The development of computer-aided design (CAD) and computer-aided engineering (CAE) allowed for the emergence of simulation-based, gradient-based design optimization. This consists of the following steps: structural modelling, design parameterization, structural analysis, problem definition, sensitivity analysis and optimization:

- Structural Modelling simplifies the physical engineering problem into a mathematical problem of a desired level of accuracy. Design parameterization is then carried out to define the geometric parameters of the structural model and define a set of geometric parameters as design variables. Design parameterization is essential for obtaining a good optimum design since the design variables define the design space within which powerful mathematical tools known as optimization algorithms will search for optimum solutions.

- Structural analysis in simulation-based optimization is typically carried out by employing numerical approaches which are able to solve partial differential equations (PDE) describing complex engineering problems. A variety of numerical methods are used in structural analysis including the well-known finite element analysis (FEA), boundary element analysis (BEA), computational fluid dynamics (CFD) and meshfree analysis. The analysis selected must be able to handle all possible designs in the defined design space both in terms of accuracy and efficiency and also the required performance measures (objectives and constraints).

- To define an optimization problem, the cost function also called the objective function, and constraints need to be defined. The objective function representing the performance measure of choice, is minimized or maximized during optimization. Constraint functions are the criteria that the system has to satisfy for a design to be feasible. In other words, they indicate design requirements or restrictions. This means that among all possible designs, only those that satisfy the constraints are candidates for the optimum design. For example, it may be desirable to design a structure with minimum weight while the maximum stress is less than the yield stress. In this situation the weight is the objective function whereas stress acts as a constraint with the yield stress as the limiting constraint value. Mathematically, an optimization problem is formulated as:

$$\begin{aligned} \underset{\mathbf{x}}{\textbf{minimize}} \quad & f(\mathbf{x}) \\ \textbf{subject to} \quad & \mathbf{g}(\mathbf{x}) \leq 0 \\ & \mathbf{h}(\mathbf{x}) = 0 \end{aligned} \tag{1.1}$$

  where $f(\mathbf{x})$ is the objective function, $\mathbf{g}(\mathbf{x}) = [g_1(\mathbf{x}), g_2(\mathbf{x}), ..., g_n(\mathbf{x})]^T$ and $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), h_2(\mathbf{x}), ..., h_m(\mathbf{x})]^T$ are functions representing inequality and equality constraints, respectively, and $\mathbf{x} = [x_1, x_2, ..., x_p]^T$ is the vector of design variables.

- Typically, gradient-based optimization algorithms require the function values and gradient information at given design variables. For a given set of design variables that define a structural model, the structural analysis provides the values of the objective and constraint functions whereas design sensitivity analysis is used to compute the gradients, or else the sensitivities, of the objective function and constraints with respect to the design variables.

- Once the sensitivity is computed, a gradient-based optimization algorithm can be used to find an optimum design that achieves the best cost function while satisfying all the constraints. This is done in an iterative manner as shown in Fig. 1.1.

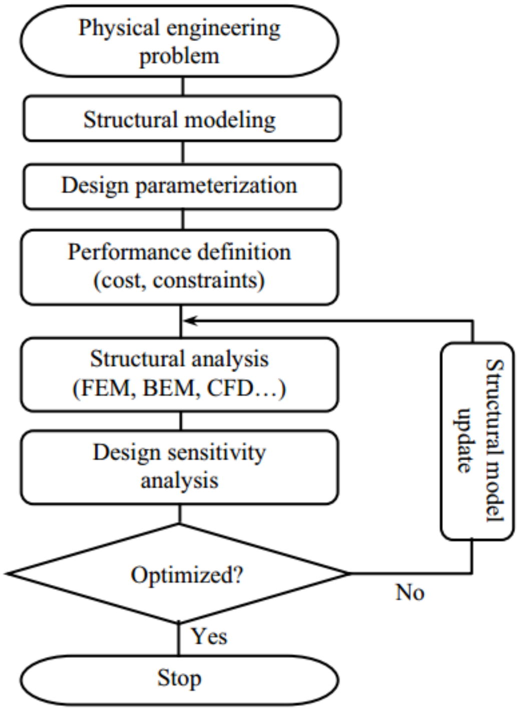

Physical engineering problem
Structural modeling
Design parameterization
Performance definition (cost, constraints)
Structural analysis (FEM, BEM, CFD…)
Design sensitivity analysis
Optimized?
No
Yes
Stop
Structural model update

Figure 1.1: Structural design process. (Choi and Kim, 2005)

Structural optimization is a powerful tool that can assist engineers to find innovative designs, especially in problems in which engineering intuition and experience alone are not sufficient to provide the best performing structures. A good example is problems governed by multiple physics, and in which different objectives may contradict each other. For instance, consider a system governed by heat transfer and fluid flow such as heat dissipating devices including cooling channels, heat exchangers and heat sinks. Such devices are commonly used to dissipate heat generated from engines, batteries or other devices by passing a liquid through the structure. When it comes to designing these structures, common performance measures include the amount of heat exchanged and the pressure gradient required to pump the fluid. These are objectives that oppose each other since a higher pressure drop results in higher fluid velocity and conse-quently higher heat dissipation. In such complex design situations structural optimization can be useful especially at the initial stages of the design process where innovative solutions are sought.

Depending on the design parameterization and the set of design variables, structural optimization can be categorized into *sizing*, *shape* and *topology* optimization (Choi and Kim, 2005).

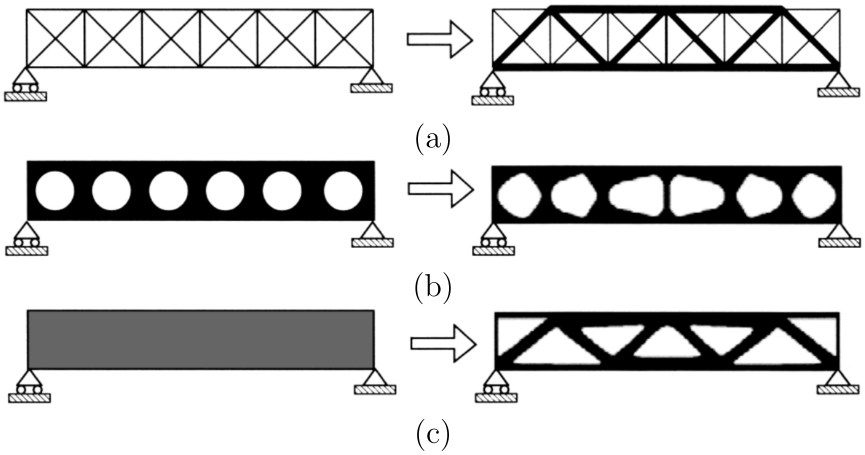


Figure 1.2: Three types of structural optimization: (a) Sizing, (b) Shape and (c) Topology optimization (Bendse and Sigmund, 2004)

Sizing design variables are related to the geometric parameter of the structure, and these may include for example the thickness or length of the structure. Thus, in sizing optimization the shape and topology of the structure do not change, and as shown in Fig. 1.2 (a) the global geometry of the final structure is the same as the initial one.

Shape optimization on the other hand offers more design freedom since the shape design variable is related to the structure's geometry. Different structural shapes can be obtained by different values of the shape design variables which is illustrated in Fig. 1.2 (b) where the shape of holes in the final design is altered in comparison to the initial design. Note however that new holes cannot spontaneously appear in the design domain.

This leads to the third type of structural optimization, i.e., topology optimization, which provides the most design freedom in comparison with size and shape optimization. Topology optimization determines the structure's layout, and it allows for new holes and structural members to be created with no assumption on the initial design as illustrated in Fig. 1.2 (c).

1.2 Topology optimization methodologies

The main idea of topology optimization is to define the optimum structure by distributing the material inside the design domain. In other words, it answers the question "where to place the holes inside the material to achieve the best performing structure.". Over the years different approaches have been developed depending on the type of design variables and material model used. The most established methods can be categorized as

- Density-based methods such as the Solid Isotropic Material with Penalization (SIMP) and the Homogenization method.

- Discrete methods, including the Evolutionary Structural Optimization (ESO) and the Bi-directional Evolutionary Structural Optimization (BESO).
- Level-set topology optimization methods (LSTO), based on boundary variations.

The specific problems studied in this thesis pose certain challenges and one of the main objectives of the thesis is to address those challenges by developing alternative implementations in the context of level set topology optimization. To explain why LSTO is the preferred choice in this thesis, it is worth providing an overview of each topology optimization approach in this section.

1.2.1 Density-based methods

Typically, density-based topology optimization employs finite element analysis with the material distribution described by a continuous density variable ρ. Once the domain is discretized into a mesh of elements, the material properties such as Young's modulus within elements can be controlled. The density variable within each element indicates whether the element should be void ($\rho = 0 \rightarrow$ material property $= 0$) or whether it should be filled with material ($\rho > 0 \rightarrow$ material property $\neq 0$). The general formulation of this problem can be written as (Sigmund and Maute, 2013).

$$\begin{array}{ll} \underset{\boldsymbol{\rho}}{\textbf{minimize}} & F(\boldsymbol{\rho}, \mathbf{u}(\boldsymbol{\rho})) \\ \textbf{subject to} & \mathbf{g}(\boldsymbol{\rho}, \mathbf{u}(\boldsymbol{\rho})) \leq 0 \\ & \mathbf{h}(\boldsymbol{\rho}, \mathbf{u}(\boldsymbol{\rho})) = 0 \\ & \rho_{min} \leq \rho_i \leq 1, \quad i = 1, ..., N \end{array} \tag{1.2}$$

where $F(\boldsymbol{\rho})$ is the objective function depending on the vector of design variables $\boldsymbol{\rho}$, and the vector of state variables $\mathbf{u}(\boldsymbol{\rho})$ that represents the solution to the governing equations of the problem. For example, $\mathbf{u}(\boldsymbol{\rho})$ could be the displacement solution to the linear elasticity equation governing a structure for which we are trying to maximize stiffness. As in Eq. 1.1, $\mathbf{g}$ and $\mathbf{h}$ are functions representing inequality and equality constraints, respectively. The element densities are restricted to a lower bound ρ_{min} to avoid singular structural problems. A problem often encountered in the early days of topology optimization was the checkerboard problem with patches of alternating black and white element. Checkerboards can be avoided by use of higher order elements or they are taken care of by restriction methods that ensure mesh-independence (Sigmund and Maute, 2013). Ideally a solution that converges to well defined 0-1 solutions is sought, i.e. eliminating the presence of elements with intermediate density variables as much as possible. To this end, the widely used SIMP (Solid Isotropic Material with Penalization) approach was developed (Bendse, 1989; Bendse and Sigmund, 2004). In the SIMP approach the relation between the density design variable and the material property is given by the power-law, which through the example of Young's modulus as the material property looks like

$$E(\rho_i) = \rho_i^p E_0 \tag{1.3}$$

where p is the penalization parameter and E_0 is the Young's modulus of the solid material. Other material interpolation schemes exist and essentially all aim to provide continuous interpolation between solid and void with a penalization of intermediate density values (Sigmund and Maute, 2013). Although penalization drives the design towards a 0-1 solution, often some elements with intermediate densities remain close to the boundary of the structure. As it will be discussed later, this makes certain classes of problems difficult to solve with density-based methods.

1.2.2 ESO and BESO methods

Evolutionary structural optimization (Xie and Steven, 1993) applies a discrete design update scheme employing discrete design variables. This means that if the design variable is one within a finite element then the element carries material, whereas if the design variable is equal to zero then the element is void. So, in contrast to density-based methods that have gray elements close to the boundary, in evolutionary methods the boundaries are well defined by the edges of the finite elements.

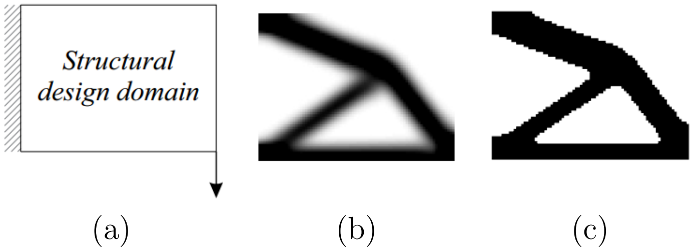


Figure 1.3: Different solutions obtained for the design problem defined in (a). Solutions are shown using: (b) The SIMP method (Sigmund, 2001a) and (c) The BESO method (Sanches, 2015)

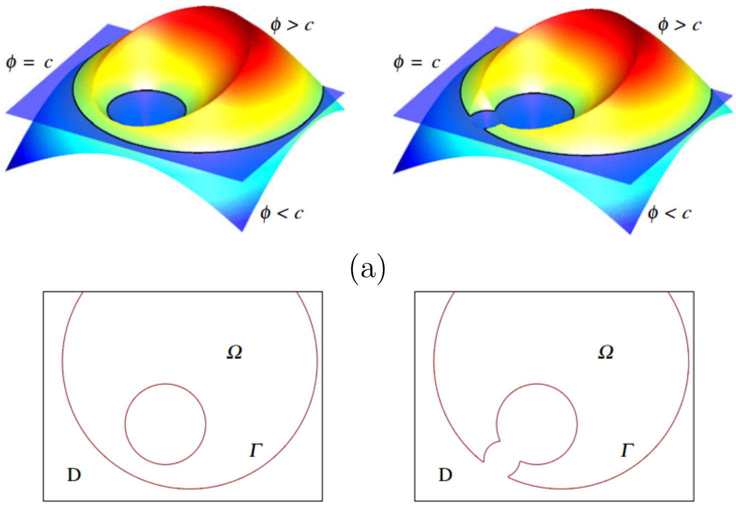
Evolutionary methods are similar to density-based methods in the sense that they both employ the finite element analysis, filtering and adjoint sensitivity analysis. However, the way the design variables are updated is different. In the ESO method, inefficient material is gradually removed from a structure. Through this process the resulting structure will evolve towards its optimal shape and topology. A later version of the ESO method is the bi-directional evolutionary topology optimization method (BESO) (Huang and Xie, 2007) which allows material to be removed and added simultaneously and uses standard adjoint gradient analysis and filtering techniques, similar to those used in the density-based approaches (Sigmund and Maute, 2013).

1.2.3 Level set methods

Level set topology optimization methods define a structure implicitly by iso-contours of a level set function (Wang et al., 2003; Allaire et al., 2004). The level set function defines the structure in the following way (van Dijk et al., 2013)

$$\begin{cases} \phi(\mathbf{x}) \geq c & \mathbf{x} \in \text{material} \\ \phi(\mathbf{x}) = c & \mathbf{x} \in \text{interface} \\ \phi(\mathbf{x}) < c & \mathbf{x} \in \text{void} \end{cases} \tag{1.4}$$

where c is a constant (conventionally $c = 0$) and $\mathbf{x}$ is a point in the design domain. Based on this representation, the topology of the structure can be altered by changing the level set function. Figure 1.4 shows a visual representation of this approach.



(a)

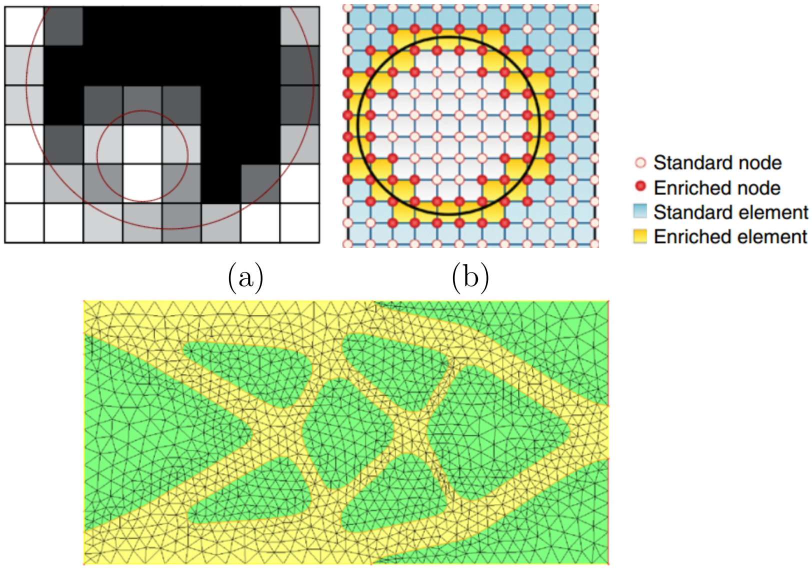
(b)

Figure 1.4: Illustration of how a change in the level set function ϕ in (a) translates into change in the topology of the solid domain Ω in (b). The symbols Γ and D represent the interface where $\phi = c$ and the non-material domain, respectively (van Dijk et al., 2013)

Once a geometry is defined by the level set method, a numerical approach such as the FEA is used to discretize the model and solve the governing equation for the structure. Depending on the level set function parameterization, different LSTO methods have emerged in the literature.

The classical LSTO introduced by Wang et al. (2003) and Allaire et al. (2004) employ a Hamilton-Jacobi equation which is solved to update the location of the boundary. This is also the method used in this thesis. Level set methods that use different strategies also exist in the literature such as the parametric level set method (Pingen et al., 2010), the topological derivative method (Norato et al., 2007) and the reaction-diffusion equation method (Yamada et al., 2010). An comprehensive review can be found in van Dijk et al. (2013).

In the context of the classical level set method, different approaches have appeared based on the mapping used to transfer the geometry from the level set representation onto the computational domain. Conventionally, Eulerian (fixed) grid FEA is employed to avoid having to remesh as a structure changes. This FEA mesh may or may not coincide with the level set mesh on which the geometry is constructed. However, this results in the appearance of finite elements that are cut by the structural boundary. These require special treatment since they belong both in the solid and void domains. The two most common ways is the Ersatz material approximation (Dunning et al., 2011) and the extended finite element method (XFEM) (Jenkins and Maute, 2016) shown in Figs. 1.5 (a) and (b), respectively. Ersatz approximation averages the material property within a cut element by an element density defined as the area of the solid within the cut element divided by the total element area. The appearance of a density can be seen as similar to density-based methods however the gray elements are limited to just one layer of elements along the boundary. Nevertheless, the crisp boundary representation of the level set method is not maintained onto the computation mesh in this way. The XFEM on the other hand is essentially an immersed boundary method which models the boundary by adding enrichment functions and additional degrees of freedom to nodes around the boundary discontinuity (Wein et al., 2020). Boundary conditions can be enforced directly along the interface, which provides a crisp description of the boundary on the structural model. Elements that are cut by the discontinuity still need to be integrated. This is generally done by locally remeshing the element, i.e., partitioning the domain into elemental shapes such triangles as shown in Fig. 1.6 and then using Gaussian quadrature over the solid part of these sub-elements (Bosma, 2013).



(a) (b)

(c)

Figure 1.5: Different methods for mapping the level set geometry onto the computational domain: (a) Ersatz material approximation (van Dijk et al., 2013), (b) Extended finite element method (XFEM) (Khoei, 2014) and (c) Remeshing for conforming mesh (Allaire et al., 2014)

An alternative to fixed grid LSTO approaches is to remesh the structure at each iteration such that nodal degrees of freedom are aligned with the level set boundary (Allaire et al., 2014). This is shown in Fig. 1.5 (c). This is the most direct way of ensuring that the crisp boundary representation is maintained onto the computational domain. However, remeshing can be challenging since in order to maintain stability and good accuracy of the analysis the mesh needs to undergo several operations to ensure good element qualities near the boundary. This can be computationally intensive especially for when large topological changes occur.

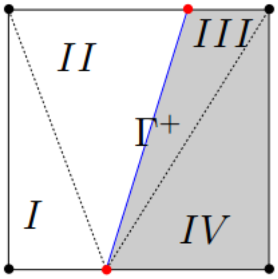


Figure 1.6: Gauss integration for cut elements in XFEM: Partitioning of the element into triangular sub-elements and integration over the solid areas I and II (Bosma, 2013).

1.3 Main focus of the thesis: The design-dependent problem

Despite the advancements in topology optimization, there are still problems that are considered challenging for all methods. The class of problems known as "design-dependent" is a good example. These are problems in which the applied loads depend on the design itself. As the structure undergoes topological changes throughout the optimization process, the direction, location and magnitude of the loads change accordingly as shown in Fig. 1.7. Such design-dependent problems can be classified into two categories based on the type of loads considered, namely, volumetric and surface loads. Volumetric loads, such as thermal expansion and self-weight loads, depend on the volume of the material in the structure. Surface design-dependent loads on the other hand act on the boundary of the structure and they depend on the surface configuration as the shape changes. Hydrostatic pressure, convective heat transfer, acoustics and fluid-structure interactions are characteristic examples. Design-dependent surface loads are the focus of this thesis. The main challenge lies in tracking the interface at each optimization iteration to correctly apply loads and coupling conditions.

The research aim of this thesis is to develop methodologies for design-dependent multiphysics topology optimization in which the physical phenomena at the boundaries is what drives the optimization procedure. Starting from simple cases such as hydrostatic pressure loads, the effectiveness of the proposed methodologies and numerical ingredients can be validated. Then, once an approach is established, more challenging design dependent physics are considered. Specifically, fluid-structure interaction is the main interest, although there is no theoretical limita-tions in applying the same methodologies for other design-dependent surface physics. At this point it is worth giving a brief overview of what topology optimization approaches currently exist for addressing the challenge of the design-dependency and what their limitations are. Note that only an overview of the different methods is given here, whereas specific citations are discussed later in the literature review in Ch. 2.

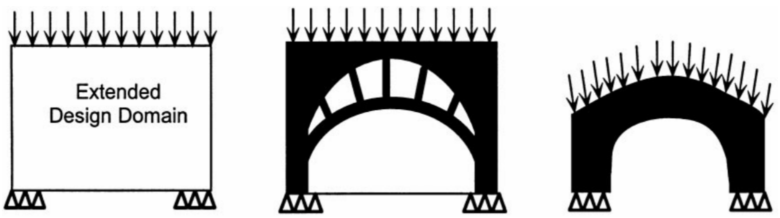
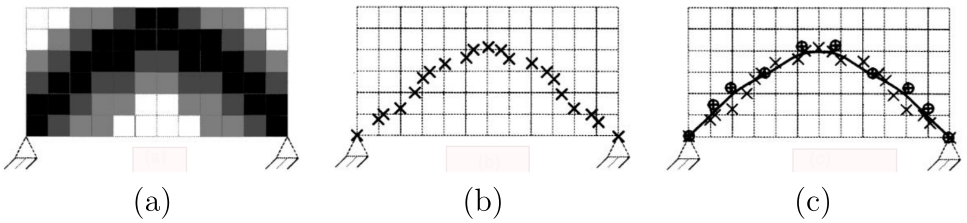


Figure 1.7: Difference in optimum designs when fixed and design dependent loads are considered: (a) Design domain, (b) Possible optimum topology for fixed loads case in which loading surface is fixed throughout optimization and (b) Possible optimum topology for design-dependent loads case in which the loading surface changes throughout optimization and thus the loads change in direction and location (Chen and Kikuchi, 2001)

Starting from density-based methods such as SIMP, the presence of elements with intermediate densities makes it difficult to identify a clear structural boundary as shown in Fig. 1.8. Thus, alternative ways of mimicking a boundary need to be implemented. One example of such an approach is shown in Fig. 1.8 (a) and (b), where at each iteration points of equal density are first identified and then used to fit spline curves such that they create an approximate loading surface. A main issue with this approach is that the iso-density value needs to be carefully selected every time to avoid ill-defined loading curves. Alternative density-based implementations will be discussed in more detail in Ch. 2 where the literature review is presented.

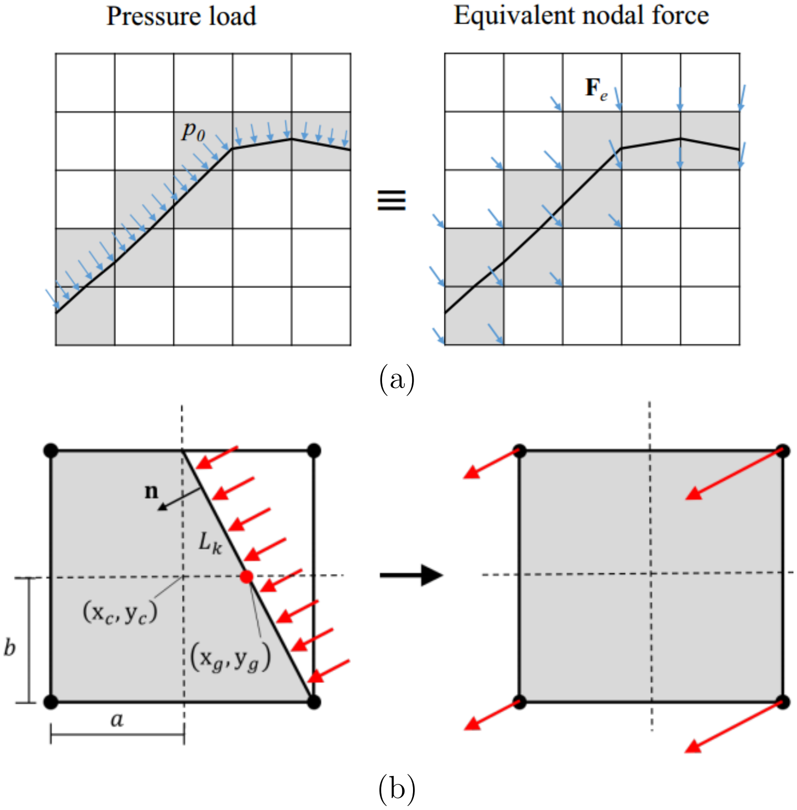


(a) (b) (c)

Figure 1.8: (a) Lack of a clear interface due to the presence of "gray" elements with intermediate densities, (b) Points of equal density and (c) Fitted spline curves to construct a loading surface. (Hammer and Olhoff, 2000)

BESO methods on the other hand do not employ the continuous density variable, thus avoiding some of the challenges in SIMP. However, the piecewise constant discrete nature of BESO often yields boundaries represented by the finite elements' jagged edges. In more complex multiphyiscs problems such as fluid-structure interactions, this can cause problems especially for coarser meshes leading to flow instabilities near the interface and thus a poor description of the interface physics (Alexandersen and Andreasen, 2020).

Level set methods have an advantage when it comes to boundary definition as the boundary is implicitly represented by a level set function. Therefore, solid and void regions are well defined. Despite this, when fixed finite element grids and the Ersatz material approximation is used for the analysis, this crisp boundary representation of the level set mesh is not maintained on to the computational domain due to the presence of elements cut by the boundary. Typically, load transformation techniques are employed to transfer the load from the interface onto the nodes of the element as work-equivalent nodal loads. The examples in Fig. 1.9 illustrate such a transformation.



(a)

(b)

Figure 1.9: Work-equivalent nodal loads in LSTO: (a) By Emmendoerfer et al. (2018), (b) By Picelli et al. (2019). Where L_k is the kth segment defined by cutting the element with the level set boundary, $\mathbf{n}$ is the inward normal vector, (x_c, y_c) are the coordinates of the element centroid and (x_g, y_g) are the coordinates of the integration point for Gauss quadrature along the segement to compute the equivalent nodal load.

XFEM on the other hand offers a crisp boundary representation, however several difficulties can occur. For example, noise and ill-conditioning of the discretization may occur in case of small intersections of finite elements (i.e., cut elements with small solid fractions as reported by Miegroet and Duysinx (2007) which is an issue that Ersatz approximation also shares. Another potential source of error is the situation of non-physical coupling when the gap between solid components is less than the size of an element (Wein et al., 2020), as shown in Fig.

1.10.

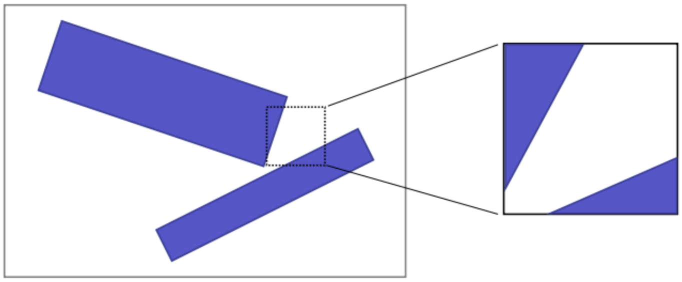


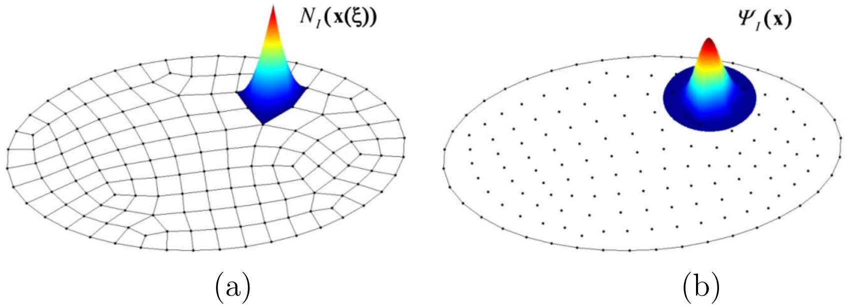
Figure 1.10: Non-physical coupling that can potentially happen in XFEM (Wein et al., 2020)

Remeshing the structure at each iteration such that the FEA mesh conforms the level set boundary is another alternative presented by Allaire et al. (2014), however, a series of operations is required to ensure good element qualities which depending on user expertise may not be straightforward since certain parameters are user defined.

1.4 Addressing the interface identification challenge by employing the meshfree reproducing kernel particle method (RKPM)

As can be concluded from Sec. 1.3, depending on the methodology used, different numerical schemes are needed to address the design-dependency challenge. Typically, this requires either for the loads to be transformed into equivalent nodal loads or for an artificial interface to constructed. Even methods that directly address the problem such as XFEM or remeshing have additional difficulties to deal with as discussed. One thing that becomes apparent, is that since the level set method provides a clear interface at least geometrically, it can naturally be more advantageous in design-dependent surface problems. Thus, the LSTO is the optimization technique used in this thesis. The remaining question is what could be an appropriate numerical method to combine with the LSTO to solve design-dependent problems in a straightforward way, without load interpolations, which at the same time avoids the numerical difficulties arising with conventionally used methods.

A new class of analysis methods has emerged in the last two decades, in which no mesh is needed and shape functions are constructed from a scattered set of particles as shown in Fig. 1.11. These are known as meshfree methods and they are designed to inherit the useful characteristics of the FEM, such as compact supports of shape functions and good approximation properties, and at the same time overcome the main disadvantages caused by the mesh-dependence. Over the years, a wide variety of meshfree methods have been studied and they can generally be categorized into collocation and Galerkin meshfree methods. Collocation methods are based on the strong form of PDEs whereas Galerkin meshfree methods are based on the weak form of PDEs. The element free Galerkin method (EFG) (Belytschko et al., 1994) and the reproducing kernel particle method (RKPM) (Liu et al., 1995; Chen et al., 1996) are typical Galerkin-based examples. A recent extensive review can be found in Chen et al. (2017b). Meshfree methods have been used to study a wide variety of problems in engi-neering as an alternative to the FEM. Some examples include fracture mechanics (Belytschko et al., 1994), large deformation problems (Chen et al., 1997) and contact mechanics (Wang et al., 2014).



(a)

(b)

Figure 1.11: Shape function construction in (a) Finite element method based on a mesh, (b) Meshfree methods based on a scattered set of points (Chen et al., 2017b)

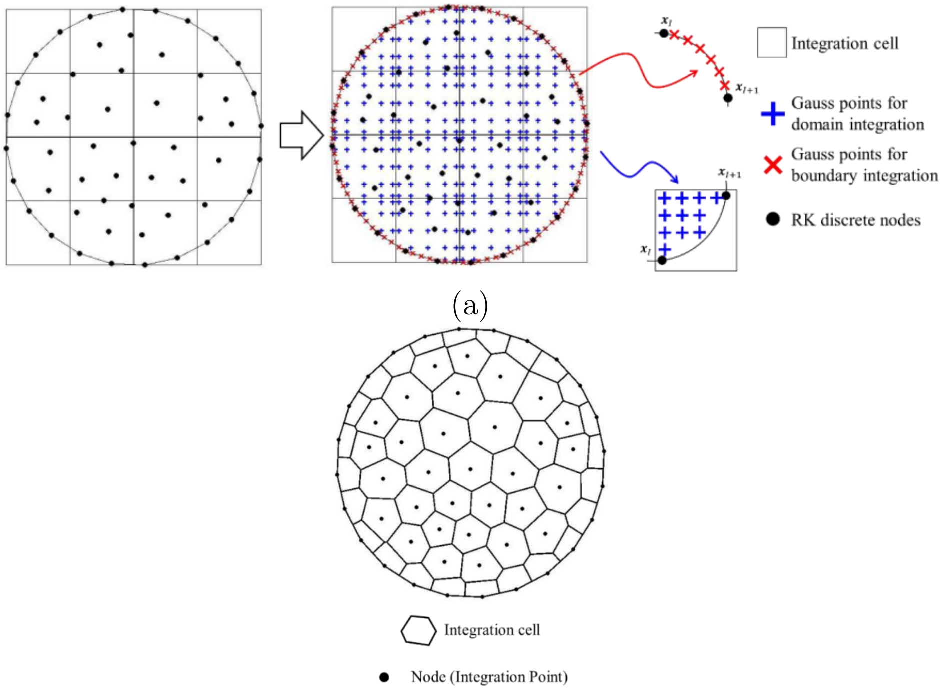
In this thesis we employ the meshfree reproducing kernel particle method (RKPM) for performing the analysis and computing sensitivities. As RKPM offers the ability to place particles anywhere in the design-domain, and furthermore the analysis is not sensitive to the locations of these particles, this allows for the development of an LSTO-RKPM methodology that can maintain the crisp boundary representation from the level set onto the computational domain. This can straightforwardly be done by placing RKPM particles along the structural boundary, which enables the direct application of the loads on the boundary without any load interpolation or transformation schemes. Furthermore, since RKPM does not rely on a mesh, the mesh-based difficulties appearing in conven-tional approaches are avoided.

1.5 LSTO-RKPM methodologies

The combination of LSTO with RKPM for specifically addressing design-dependent problems in topology optimization is one of the main contributions of this thesis. Apart from a straightforward way of dealing with design-dependent problems, RKPM provides additional useful features in the context of LSTO which are further investigated in this thesis. For example, different particle distributions that may enhance efficiency while maintaining accuracy. Moreover, RKPM allows for a controllable order of continuity and completeness of the shape functions without additional complexity, and independent from one another. This enables effective solutions of partial differential equations (PDEs) involving high-order smoothness or discontinuities. In the process of solving the design-dependent problem, different implementations have been developed and tested that illustrate these useful characteristics, which can be useful not only in the design dependent problem but also in a range of problems where improved smoothness and accuracy is required.

Imposition of the essential boundary conditions is also an area that requires special attention. More specifically, it is known that meshfree shape functions lack the Kronecker delta property, similarly to XFEM. In the finite element method (FEM) due to this property, shape functions corresponding to a node have a value equal to one at that node whereas their values at all other nodes are zero. This is not true in the case of meshfree methods where the shape functions of a node do not disappear completely in other nodes. In other words, meshfree shape functions do not pass through the data. This is because the meshfree shape functions are not interpolation functions, rather, they are approximation functions (Chen and Wang, 2000). This complicates the imposition of essential (Dirichlet) boundary conditions and creates the need for special techniques. Nevertheless, this topic has reached a level of maturity now and a variety of methods exist in the literature for imposition of boundary conditions (Fernandez-Mendez and Huerta, 2004; Chen et al., 2017b, 2001).

Domain integration is also a point of consideration. Although no mesh is required for constructing the shape functions, domain integration is still needed in Galerkin meshfree methods to evaluate the integrals in the weak form. This is generally done by either Gauss integration on a background mesh (Belytschko et al., 1994) or by using nodal integration over representative domains. Such domains could be for example the polygon cells created by a Voronoi diagram (Chen et al., 2001). The two methods are illustrated in Fig. 1.12. Both techniques are implemented and tested in this thesis in the context of LSTO and their differences are discussed in detail with respect to accuracy and efficiency.



(a)

(b)

Figure 1.12: Domain integration in Galerkin meshfree methods: (a) Gauss integration on a background mesh and (b) Nodal integration (Huang et al., 2019)

1.6 Extension to fluid-structure interactions

In the last part of the thesis, the LSTO-RKPM implementations developed throughout the previous chapters are extended to the more complex design-dependent case of fluid-structure interactions. Having developed a methodology for tracking the interface throughout the optimization procedure, allows for exten-sion to such more complex design-dependent physics.

Instead of pressure, the structure is now in contact with a viscous flow field governed by the Navier-Stokes equation. The literature on fluid-structure interac-tion topology optimization is very limited which is an indication of the challenging nature of these problems. This has been recognized by a recent, extensive review on topology optimization for fluids by Alexandersen and Andreasen (2020). The existing works considered steady-state laminar flows with low Reynolds numbers, and small structural deformations. In this thesis one of the main aims is to develop a methodology that is capable of solving high fidelity fluid-structure interaction topology optimization problems. In order to do so, the LSTO-RKPM methodology is coupled with a popular method for modelling fluid-structure interactions efficiently, namely, the modified immersed finite element method (mIFEM) discussed next in section 1.6.1.

1.6.1 Modified immersed finite element method (mIFEM)

Modelling fluid structure interactions (FSI) is difficult because of the complicated motions and deformations of the fluid-structure interface. Methods requiring conforming meshes between fluid and solid such as the Arbitrary Lagrange Eulerian (ALE) method (Liu et al., 1988; Hu et al., 2001) can handle complicated fluid-solid interfaces. However, they require expensive mesh-updating and remeshing. In practice, remeshing becomes increasingly challenging with large deformation of the fluid-structure interface. The situation complicates further if in addition topology optimization is to be considered, where the topological changes and the structural deformations due to the action of the fluid occur simultaneously within each optimization iteration.

To avoid such conforming processes, the immersed methods such as the immersed finite element method (Zhang et al., 2004; Zhang and Gay, 2007) have emerged as non-conforming techniques to model FSI problems. In these methods, the fluid and solid domains co-exist so that non-conforming meshes or discretizations can be used. The main idea is that an Eulerian background fluid Ω^f is coupled with a Lagrangian solid domain Ω^s that "floats" on top of the fluid as shown in Fig. 1.13 and is free to deform. The fluid region overlapping with the solid domain is called the artificial fluid domain $\bar{\Omega}$. The entire fluid domain consisting of the real fluid and the artificial fluid domains is governed by the Navier-Stokes equations. The impact of the solid is reflected as a body FSI force and no-slip boundary condition in the artificial fluid region whereas the fluid exerts a traction force on the solid boundary.

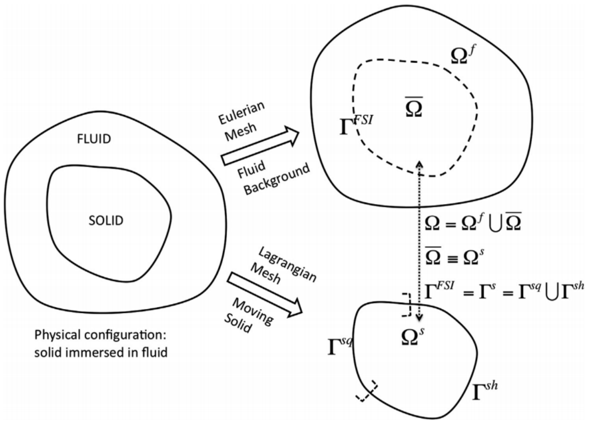


Figure 1.13: Immersed finite element method scheme (Wang and Zhang, 2013). Γ^{sq} represents the Dirichlet boundary and Γ^{sh} is the Neumann boundary.

In conventional immersed methods such as the immersed finite element method, the solid displacement is imposed from the fluid velocity, rather than being solved from its own governing equations. Essentially the solid velocity and displacement are computed by interpolation of the overlapping fluid velocity field. As the fluid and solid dynamics are different from one another, a slight mismatch usually results at each time step, with the solid solution being slightly overestimated. The accumulation of this effect over time may lead to severe mesh distortion for the solid especially for high Reynolds number flows. To avoid these issues, Wang and Zhang (2013) developed the modified immersed finite element method (mIFEM) which changes the formulation to reverse the imposition of the velocity: The solid dynamics is first solved using its own governing equation. The obtained solid velocity is then imposed onto its overlapping fluid domain. As a result, the solid dynamics is preserved, therefore resulting in more accurate and realistic coupled solutions. Moreover, since the fluid velocity is not imposed onto the solid, the incompressibility constraint on the solid when the background fluid is incompressible is being removed.

1.6.2 mIFEM-LSTO-RKPM

Considering its aforementioned advantages, mIFEM is an ideal choice for combining with the LSTO-RKPM scheme for the following reasons:

1. First and foremost it allows for modularity in the solvers. Thus, it makes it possible to link the RKPM solver developed in this thesis with any fluid solver through the mIFEM.

2. Efficiency is another important advantage of mIFEM. By employing an Eulerian grid for the fluid, the fluid equations can be efficiently solved without having to remesh for the fluid which would be a tedious task.

3. The mIFEM allows for the solid to deform freely on top of the fluid mesh as it undergoes topological changes throughout the optimization procedure. Since the RKPM can handle straightforwardly such topological changes, this combination makes it possible to have a crisp solid boundary represen-ta-tion. The idea is that the structure is geometrically described by LSTO and then using RKPM the crisp level set boundary representation is maintained onto the computational model by placing RKPM particles along the boundary and inside the structure.

1.7 Thesis objectives

The overall goal of this thesis is to develop meshfree level set topology optimization methodologies for design-dependent surface physics. The main challenges to be addressed were discussed in this first Chapter. To conclude this introduction, the main objectives and key elements are summarized below.

1. To develop a meshfree level set topology optimization methodology based on Galerkin meshfree methods, specifically the reproducing kernel particle

method. Explore the different features of the methodology such as different particle distributions, different orders of continuity, different methods for imposition of essential boundary conditions and different domain integration techniques to test what combination provides the most robust methodology.

2. To employ the LSTO-RKPM methodology to solve the design-dependent problems with hydrostatic pressure loads, the simplest case of design-dependent physics. Validate for this simple case of design dependent physics and compare with the literature.

3. To extend the methodology to the more complex case of design-dependent fluid structure interaction by coupling the LSTO-RKPM methodology with the mIFEM.

4. To investigate fluid-structure interaction topology optimization problems and compare with the literature for verification.

1.8 Layout of the thesis

The thesis is composed by seven chapters as follows,

1. In this first introductory chapter the basic terminology for structural optimization has been explained, leading to a discussion on topology optimization and the different approaches currently used. The main challenges and objectives that this thesis aims to address were also outlined. Specifically, design-dependent surface physics are of interest, from the simplest case of hydrostatic pressure to the more complex FSI problems. A methodology that can handle any type of design-dependent surface physics is the main objective. For this purpose, a meshfree LSTO-RKPM approach is developed that is capable of maintaining the crisp level set boundary onto the computational domain.

2. In Ch. 2 the state of the art in the main areas of interest is provided with a detailed description of specific works. The chapter is divided into three parts to categorize works for design-dependent topology optimization, meshfree-based topology optimization and FSI topology optimization. At the end of each category a summary of the main conclusions drawn from the existing literature is discussed to emphasize the main challenges the thesis aims to address.

3. Chapter 3 goes into the mathematical formulation and specific details of the LSTO and RKPM methods separately. Based on the different ingredients of RKPM, different implementations can be developed as illustrated through the examples in the subsequent chapters.

4. An LSTO-RKPM implementation with Gaussian integration on a fixed background mesh is presented in Ch. 4 to solve design-dependent problems with hydrostatic pressure. The obtained results are compared with the literature and also against an FEA implementation to illustrate the benefits of the methodology.

5. Before extending the methodology to FSI, an alternate LSTO-RKPM implementation employing nodal integration is developed. The purpose for this is to address several challenges with the Gauss integration approach that were observed in Ch. 4 especially in terms of computational efficiency. As explained in the chapter, the nodally integrated-based RKPM achieves much higher efficiency while maintaining the same (or even better) levels of accuracy. The robustness of the approach is tested against stress-based and design-dependent examples. Through the examples, numerical experimentations with different particle distributions and continuity orders are also illustrated.

6. With a robust LSTO-RKPM methodology at hand, extension to design-dependent FSI is presented in Ch. 6. The solid LSTO-RKPM solver is coupled with the modified immersed finite element method through the OpenIFEM opensource platform to perform the FSI analysis. Several challenges specific to FSI topology optimization are addressed in the chapter such as the identification and removal of free-floating volumes of solid material into the fluid domain and the separation of the loading portion of the boundary from the load-free part. Furthermore, a particle based discrete adjoint in the context of LSTO is explained to compute the FSI sensitivities. To validate the mIFEM-LSTO-RKPM approach examples from the literature are solved and compared.

7. In the last chapter, the main conclusions drawn from the thesis are discussed and possible future directions of the current work are provided.

Chapter 2

Literature Review

2.1 Meshfree methods in topology optimization

The vast majority of topology optimization methods are based on the finite element method (FEM). In recent years, authors started exploring meshfree methods in topology optimization and taking advantage of their useful features such as mesh independency, higher order approximation and smoothness.

A significant part of the literature on meshfree topology optimization combines meshfree methods with density-based approaches such as the SIMP method. Especially the use of Galerkin based methods has been shown to alleviate the need for sensitivity filtering due to the higher order smoothness of the meshfree shape functions and avoid mesh-dependence phenomena (Gong et al., 2010; Luo et al., 2013). Commonly, density-based meshfree methods are based on pointwise density interpolation schemes, with either the densities at the Gauss points considered directly as design variables or with the nodal densities defined as design variables and then used to interpolate the density field at the computational points based on the meshfree shape functions as explained in the next paragraph.

Du et al. (2009) used the EFG method integrated into topology optimization with the SIMP method for the design of thermomechanical compliant mechanisms with geometrical nonlinearities. To eliminate the appearance of discontinuous scattered points in the topology, sensitivity filtering is used to smear out the numerical instability. For the integration of the weak form Gaussian quadrature on rectangular background cells is used, and the artificial densities at the Gauss points are considered as the design variables. The EFG method with Gaussian integration on a background mesh in combination with the SIMP method was subsequently used by other authors to study a variety of problems. Gong et al. (2010) presented a study with multiple loading conditions and compliance minimization under stress constraints. The densities at the particles are chosen as the design variables and are used to interpolate the densities at the computational/ Gauss points using the meshfree shape functions. This exploits the smoothness of the meshfree shape functions to improve the smoothness of the density field and eliminate the checkerboard problem. Zheng et al. (2012) studied free vibrating continuum structures with the objective of maximizing the fundamental eigenvalue and considering the nodal densities as design variables and Gong et al. (2012) applied the EFG-SIMP method to the modal topology optimization method for maximizing the first-order natural frequency. Luo et al. (2013) proposed an approach in which the Shepard function is applied to construct a physically meaningful dual-level density approximant, due to its non-negative and range-restricted properties. The density at any computational point is interpolated by the Shephard function and the density values of nodes inside the influence domain of the current point. This density approximant acts as a heuristic filtering scheme to enhance the smoothness of the density field. Geometrically nonlinear structures were considered by He et al. (2014). The approach is based on a similar point-wise material density field and Shepard interpolation used in Luo et al. (2013). The difference in this work is that the density design variable points are freely positioned independently of the displacement nodes used in the displacement analysis. However, for ensuring well-posedness of the optimization problem, the density points should be distributed in a reasonable manner according to the displacement node arrangement. Gong et al. (2018) proposed a SIMP based topology optimization-EFG method based on moving particles rather than "soft delete" or "hard delete". The moving of particles is controlled based on the density values change in each optimization step, by moving particles with densities lower than a threshold value to positions with higher density values and keeping high density particles still. Zhang et al. (2020) used the EFG method combined with the rational approximation of material properties (RAMP) model for topology optimization of heat conduction in both isotropic and anisotropic materials. The densities of the EFG nodes are chosen as design variables and the densities at the Gauss points are computed by the interpolation of the nodal densities with the moving least squares (MLS) shape function in the domain of influence.

There are also implementations based on different schemes in terms of the meshfree method and integration scheme. Cho and Kwak (2006) used the reproducing kernel particle method in combination with a density-based approach for topology optimization of geometrically non-linear structures. The bulk density at the integration points of the background mesh is interpolated using the nodal densities with support domains covering each integration point using the reproducing kernel (RK) shape function. Cui et al. (2017) used the EFG method for multi-material topology optimization based on the alternating active-phase algorithm within the SIMP framework. The nodal relative density which is taken as the design variable is obtained through the Shepard interpolation in combination with the moving least square (MLS) shape function. In contrast to most meshfree topology optimization methods presented in the literature, this work employs nodal integra-tion instead of Gauss quadrature which increases the computational efficiency. Due to the order smoothness of the MLS shape functions sensitivity filtering is not used. A direct coupling between the FEM and EFG was presented by Zhang et al. (2018a) to reduce the computational cost of the EFG method. A constraint centroidal Voronoi tessellation (CCVT) algorithm linked with the density variable of SIMP is used to generate the point set for discretizing the EFG domain. A dual-level density approximant is adopted to formulate a continuous material distribution. The density field at the Gauss points is interpolated by the density field at the FE and EFG nodes in the influence domain using two different Shepard functions. The domain integration is performed using the Gauss quadrature on the background cells whereas multilevel Gauss quadrature points are used to further reduce the computational cost. The authors later extended the approach to topology optimization of hyperelastic structures (Zhang et al., 2018b). Zhou and Zou (2008) presented a meshless topology optimization method based on the implicit topology description and the reproducing kernel particle method using nodal design variables and the smoothed Heaviside function with a regular background mesh for the integration of the weak form in the domain.

Some works exist in the literature employing the smoothed particle hydrodynamics method which is based on the strong form of the governing equations in contrast with the Galerkin meshfree methods. Lin et al. (2016) introduced the corrective smoothed particle method and total Lagrangian formulation in order to eliminate intrinsic problems in the smoothed particle hydrodynamics (SPH) method such as inconsistency and instability. Each SHP particle is assigned a density design variable and topology optimization is performed using a modified SIMP approach. In contrast to the Galerkin meshfree topology optimization methods, this approach requires a sensitivity filter to avoid the formation of checker-board patterns. Minimization of compliance was considered in the examples. The corrective smoothed particle method (CSPM) was also used by Li et al. (2020) for optimizing linear structures under multiple load cases. Particle's density is chosen as the design variable and it is approximated using the Shepard interpolation scheme to make it smooth and range-restricted in [0,1]. The modified SIMP method is applied to determine the Young's modulus, which is related to the particle densities. A sensitivity filtering technique is also introduced to enhance the numerical stability and prevent the checker-board patterns.

In the context of level set methods, Luo et al. (2012a) proposed a meshless Galerkin parametric level set method based on compactly supported basis functions (CSRBFs). CSRBFs are used to both parameterize the level set function and to construct the shape functions for the meshfree approximation. For the domain integration a fixed background mesh is employed with 4×4 Gaussian quadrature. The particles were placed at the nodal positions of the background mesh and their positions remain unchanged throughout the optimization process. To account for the boundary discontinuity, the CSRBFs were used to interpolate the level set function at quadrature points based on the discrete level set function nodal values. Quadrature points with a positive or zero level set value were considered as solid whereas a weak material was assigned to points with negative level set values. Compliance minimization for linear elastic structures under constant point loads was solved in this work and the authors later extended the methodology to optimization of compliant multiphysics actuators (Luo et al., 2012b). Subsequent level set optimization based on meshless methods also used interpolation schemes and Gaussian integrations on fixed background meshes. For example, the parametric level set method based on CSRBFs was also used by Ai and Gao (2019) to optimize two-dimensional microarchitected periodic metamaterials for the cases of single and multiple materials. Similarly to Luo et al. (2012a), the material discontinuity is treated by an interpolation scheme in which the level set function value at a Gaussian quadrature point is determined through the interpolation of the field nodes in the support domain using the meshfree shape functions. Khan et al. (2019) combined the EFG method with level set topology optimization using the Hamilton-Jacobi equation with a topological derivative term. Two-dimensional linear elastic problems under single and multiple point loads were examined considering compliance minimization as the objective function. The particles were at fixed locations throughout the whole optimization process.

The bi-directional evolutionary structural optimization (BESO) method was also used in combination with the EFG method by Shobeiri (2016) employing uniformly distributed nodes and Gaussian quadrature in the examples for minimum compliance of linear elastic structures. Zhao (2014) incorporated an improved meshless density variable approximation into the BESO method and used the Shepard function to create a physically meaningful dual-level density approximation based on the previous work by Luo et al. (2012a). The shape functions of the meshless Galerkin method are constructed using CSRBFs and domain integration is performed on a regular background cell structure using 4×4 Gauss quadrature.

2.1.1 Conclusions from the literature and identification of challenges

As seen from the literature, previous works are based on interpolation schemes based on nodal densities. Even level set methods use such interpolation schemes thus not transferring the clear boundary representation generated by the level set method on the computational domain. Conventionally in topology optimization, fixed grids and interpolation schemes are used to avoid the cumbersome task of remeshing the structure at each iteration. Recently Allaire et al. (2014) presented such a remeshing approach and demonstrated its applicability in a range of different physics. Nevertheless, a series of operations has to be performed at each optimization step to ensure good mesh quality. One of the main benefits of the meshfree methods is the ability to freely place particles in the design domain without relying on a mesh and without the problems caused by mesh dependency. This can allow for an exact representation of shapes without remeshing and the requirement of element quality operations. However, meshfree topology optimization approaches that use interpolation schemes do not fully exploit the advantages of the meshfree methods and lose the potential of an exact representation of the structure. Furthermore, almost all Galerkin meshfree methods used in the literature use Gauss integration on a background mesh with a higher order quadrature. Higher quadrature rules are necessary to ensure accuracy. This was identified by Belytschko et al. (1994) as a requirement, due to misalignment of the supports of the shape functions with the background cells.

One of the main objectives of this thesis is to incorporate different meshfree implementations into level set topology optimization to exploit their benefits in the context of design-dependent physics. RKPM implementations using both Gaussian integration on a fixed background mesh and nodal integration have been tested and compared in the examples presented in the thesis to highlight and address the conclusions made in this section based on the literature.

2.2 Hydrostatic pressure loads

Since hydrostatic-pressure is the simplest case of design-dependent physics, a major part of the literature is focused on pressure loads. A few works also exist on acoustics, heat transfer, and fluid structure interactions. The primary challenge which these works aim to address is to identify a clear interface and illustrate the applicability of their methodologies through this simple case of design-dependent loads.

In density-based methods, the presence of elements with intermediate densities near the boundaries makes it difficult to identify the specific surface for the moving loads to be applied. Hammer and Olhoff (2000) proposed the use of iso-density nodal points and Bezier spline curves. The volumetric density of material is used to define the load surface which is represented by spline functions with control points that depend on the design variables. The pressure acting on this surface is then transformed into nodal forces in the FEA model. A challenge with this methodology however, is the appearance of ill-defined loading curves if a cut-off value of the material density is not appropriately selected. As a consequence, invalid load surfaces may appear. To overcome this, Du and Olhoff (2004), improved this method by determining the loading surface in a given step based on the isoline of material density not only in the current step but also in the previous one. Furthermore, the loading surface is represented directly by straight segments instead of a spline curve, thus making the calculation of nodal forces simpler. The same framework for tracking the moving load surfaces was later adopted by Lee and Martins (2012).

Another way of tracking pressure load surfaces in SIMP, was introduced by Chen and Kikuchi (2001) and also used by Bourdin and Chambolle (2003). Instead of constructing a parameterized surface for the pressure to act on, the design dependent loads are simulated by fictitious thermal loads. A fluid domain is introduced along with the solid and void domains. This results in a three-phase material distribution problem within the design domain in which the solid, void, and hydrostatic fluid phases are optimally distributed. A hydrostatic pressure force exists at the interface between the solid and fluid regions and it is simulated by the thermal load due to the mismatch of thermal expansion coefficients of the two materials. The thermal stress tensor of the non-fluid area is set to be constant regardless of the density distribution to ensure that pressure does not act between the solid and fluid regions. Later, Sigmund and Clausen (2007) introduced a way to solve the pressure load problem based on a mixed displacement-pressure FE formulation. Pressure is included as a separate variable and is used to define the void phase as hydrostatic incompressible fluid. The pressure load is transferred in the domain through the incompressible hydrostatic fluid.

Bidirectional Evolutionary Structural Optimization (BESO) does not employ the continuous density variable thus, avoids some of the challenges in SIMP. Picelli et al. (2015a) proposed a BESO approach where binary solid-void design variables are used along with the process of fluid flooding which allows the fluid and structure to be modeled during optimization with separate domains. Sivapuram and Picelli (2017) recently created the Topology Optimization of Binary Structures (TOBS) method and applied a similar approach to solve for a design-dependent fluid pressure problem. However, the piecewise constant discrete nature of BESO often yield the boundaries to be represented by finite elements' jagged edges.

In contrast, the level set topology optimization method has an advantage that the boundary can be clearly represented as the structure is implicitly described by a level set function. Therefore, solid and void regions are well defined. However, without remeshing, there are no nodes along the pressure boundaries to apply the loading. In the seminal paper by Allaire et al. (2004) a Dirac delta function was used together with the Ersatz material approximation to replace the surface loads by equivalent volume forces thus avoiding the need for a pressure surface identification. Using a similar approach, Xia et al. (2015) employed two level set functions to represent the free and pressure boundaries. Shu et al. (2014) also used a Dirac delta function to minimize sound pressure in an acoustic-structural system. More recently, Emmendoerfer et al. (2018) proposed a level set topology optimization in which the surface loads acting on the moving boundary are transformed into work equivalent nodal forces to solve pressure loading problems with level set topology optimization and a fixed grid. In this work, the Ersatz material was also used for the elements cut by the boundary and a virtual fluid flooding process was used to track the pressure surface. A similar approach has recently been employed by Picelli et al. (2019) where a discretized fluid domain was used to explore benchmarking examples with purely hydrostatic pressure loading.

Alternative to the ersatz material approximation, other approaches have been used in combination with the level set topology optimization method such as the extended finite element method (XFEM) or remeshing the geometry at every iteration. To model the structural boundary using XFEM, the state variable interpolation is enriched to account for discontinuities of the variables within an element that is cut by an interface. Local remeshing is then performed within the intersected elements for the integration of the weak form. Integration problems may arise when integrating small areas such as cut elements with a small solid volume. This issue also appears in the Ersatz approximation scheme. Jenkins and Maute (2016) combined a level set method XFEM to track fluid-structure interface in a fluid structure interaction problem. A remeshing algorithm was used by Isakari et al. (2017) to track the acoustic-interfaces in a FE analysis coupled with the boundary element (BE) method. Recently, Allaire et al. (2014) presented a remeshing approach and demonstrated its applicability in a range of different physics. Nevertheless, a series of numerical operations have to be performed at each optimization step to ensure mesh quality.

2.2.1 Conclusions from the literature and identification of challenges

What can be observed from the literature on design-dependent pressure loads, is that most works employ interpolation schemes for the loads to transform them into equivalent nodal loads. The challenge of applying the loads in a direct way without the need of such interpolation and approximation schemes remains. The difficulty can rise either due to the topology optimization approach used or the analysis method employed for computing sensitivities. For example, in the case of density-based methods the optimization methodology makes the direct application of the loads difficult due to the presence of gray elements. On the other hand, with level set methods the optimization method allows for a crisp boundary representation that could help the direct application of the loads. However, the use of fixed FEA grids for the analysis again adds to the challenge. The more direct remeshing LSTO approach on the other hand requires additional mesh operations at each iteration to maintain the quality of the mesh. The direct application of loads on the structural boundary without interpolation schemes or remeshing is thus one the main challenges that this thesis aims to address. It is clear that the level set method with the implicit boundary representation has the most potential for offering an exact loading surface. However, alternatives to the fixed grid and remeshing FEA need to be explored in order to maintain this crisp boundary representation onto the computational domain. To this end, the meshfree reproducing kernel particle method is employed in this thesis in combination with level set topology optimization.

Inserting new boundary particles for the pressure loading, which can be done straightforwardly by RKPM, is an advantage of using this method. This eliminates the need to transform the loads or to remesh the structure. The clear boundary representation provided by the level set method in combination with the ability to freely place RKPM particles wherever needed in the design domain, allows for a clear identification of the pressure boundary. The particles on the boundary points, can also be useful in sensitivity computation. Sensitivities at the boundary points are required by the level set method used in this work to evolve the level set function and move the boundary as explained later in Sec. 3.1. In previous works employing the level set method described in section Sec. 3.1 with Ersatz material approximation, the computation of sensitivities at the boundary points (which exist at the finite element edges), was performed using a least squares interpolation from the sensitivity vales at the Gauss points of the elements (Picelli et al., 2018). The presence of particles on the boundary points allows to compute sensitivities at the boundary points directly without the need for stress recovery techniques.

2.3 FSI literature

One of the main challenges in topology optimization for FSI is to track the interface as the structure undergoes topological changes.

The initial work on topology optimization for FSI was done by Yoon (2010) using a density-based approach. The formulation of distinct regions for solid and fluid i.e distinct sets of governing equations is difficult in the context of density-based methods since the appearance of gray elements results in blurred boundaries thus lacking a crisp representation of the fluid-solid interface. This makes the imposition of coupling boundary conditions between separated fluid and solid domains challenging. To circumvent this challenge, typically the entire domain is assumed to be filled with a porous material resulting in a unified domain rather that separated solid and fluid domains. The system of equations is thus solved simultaneously in the whole domain. Different volume fractions of fluid and solid exist in each element in the design domain and the material properties are interpolated with respect to these volume fractions (design variables). Cases of fluids governed by the steady-state incompressible Navier-Stokes equations interacting with linear elastic structures were considered in this work. Only small deformations of the solid domain were assumed and thus two-way coupling was not actively applied. Yoon later extended the framework (Yoon, 2014) for stress-based FSI, considering mass minimization under stress constraints. Again in this work the steady state incompressible Navier-Stokes equations are considered along with linear elastic structures with small displacements. Andreasen and Sigmund (2013) used a similar approach for topology optimization of fluid-struc-ture interaction problems in saturated poroelastic media. In their model they used a macroscopic Darcy-type flow law to couple the Stokes flow in the pores of the structure with the deformation of the elastic skeleton. Lundgaard et al. (2018) revisited the unified density formulation by Yoon (2010), presenting a modified formulation which ensures length-scale-controlled designs and makes the optimization results less sensitive to the choice of interpolation function parameters and penalization strategies. Multiple objective functions were con-sidered under the assumptions of steady state conditions and small deformations of linear elastic solids.

Density methods require a fine mesh resolution to avoid blurred (intermediate density) elements as much as possible. When considering multiple physics, with an increased number of material properties, the complexity of choosing the correct form of interpolation increases substantially. Further, the fluid-structure interface is smeared over multiple elements. This lack of a clear interface is challenging when considering more complicated flow phenomena, due to the inaccurate enforce-ment of the coupling conditions and the difficulty of performing local mesh refinement around a smeared interface.

The Bi-directional Evolutionary Structural Optimization (BESO) approach for FSI presented by Picelli et al. (2015a) uses discrete rather than continuous density variables, thus avoiding some of the challenges encountered in density-based methods. Recently, topology optimization of binary structures (TOBS) was also developed by Picelli et al. (2020) for steady-state FSI problems in contact with linear elastic structures. In this method the BESO method is combined with the COMSOL Multiphysics software. The discrete interface provided by BESO is remeshed with triangular at each iteration using COMSOL to solve the fluid-structure interaction equations. Although the BESO method provides clearly defined boundaries, the piecewise constant discrete nature of BESO often yields jagged boundaries represented by the finite element edges. For coarser meshes, this may lead to flow instabilities near the interface and thus a poor description of the boundary layer (Alexandersen and Andreasen, 2020).

In contrast to density based methods, level set topology optimization provides clear structural boundaries through implicit representation. However, when the conventional Ersatz approximation is considered the crisp boundary representation is not transferred onto the finite element model, causing the same difficulties as with the density-based methods when considering FSI problems. Alternatively Jenkins and Maute (2016) used the level set method in combination with the extended finite element method (XFEM) and deformed fluid mesh. The interface is defined by the zero level set function and XFEM enriches the state variable interpolation to account for discontinuities of the variables within an element cut by the interface. This work also considered steady state FSI with the solid being linear elastic with the examples presented in the paper limited to small deformations. As mentioned in the paper, the approach is limited to moderate structural displacements. More recently, Feppon et al. (2019) employed adaptive remeshing in combination with level set topology optimization in the 2D domain. At each iteration, the fluid-solid interface is evolved by the level set method and subsequently remeshed in order to keep track of a clear interface and distinct fluid and solid domains. A series of operations is used in order to ensure good mesh quality at each step. This work considers the coupling of the steady state Navier-Stokes equations for the fluid domain, the convection diffusion equation for the whole domain and linear elasticity for the solid domain, with all equations considered in steady state. The three physics are coupled weakly meaning that the equations are solved consecutively rather than solving a monolithic coupled system.

2.3.1 Conclusions from the literature and identification of challenges

The current literature on topology optimization for FSI problems is limited and mainly considers small deformations and steady state FSI problems with only an emphasis on developing methodologies.

Furthermore, only low to moderate Reynolds numbers have been considered. These were also recognized by a recent, extensive review on fluid-based topology optimization by Alexandersen and Andreasen (2020), which showed that only 15 papers considered FSI out of 186 papers, all with such simplifications assumed. A classification of the most relevant of these works according to the methodology used and the equations considered is shown in Fig. 6.13.

There is, thus, a significant potential in topology optimization for FSI problems provided a robust methodology is developed. This leads to the main goals of this thesis, which is to develop a methodology that can handle FSI problems straightforwardly. Once a methodology has been established and validated against the existing literature, more complex problems can be considered such as transient FSI with large structural deformations. This latest challenge is not addressed in this thesis, however, the methodology developed here provides all the necessary ingredients for such an extension.

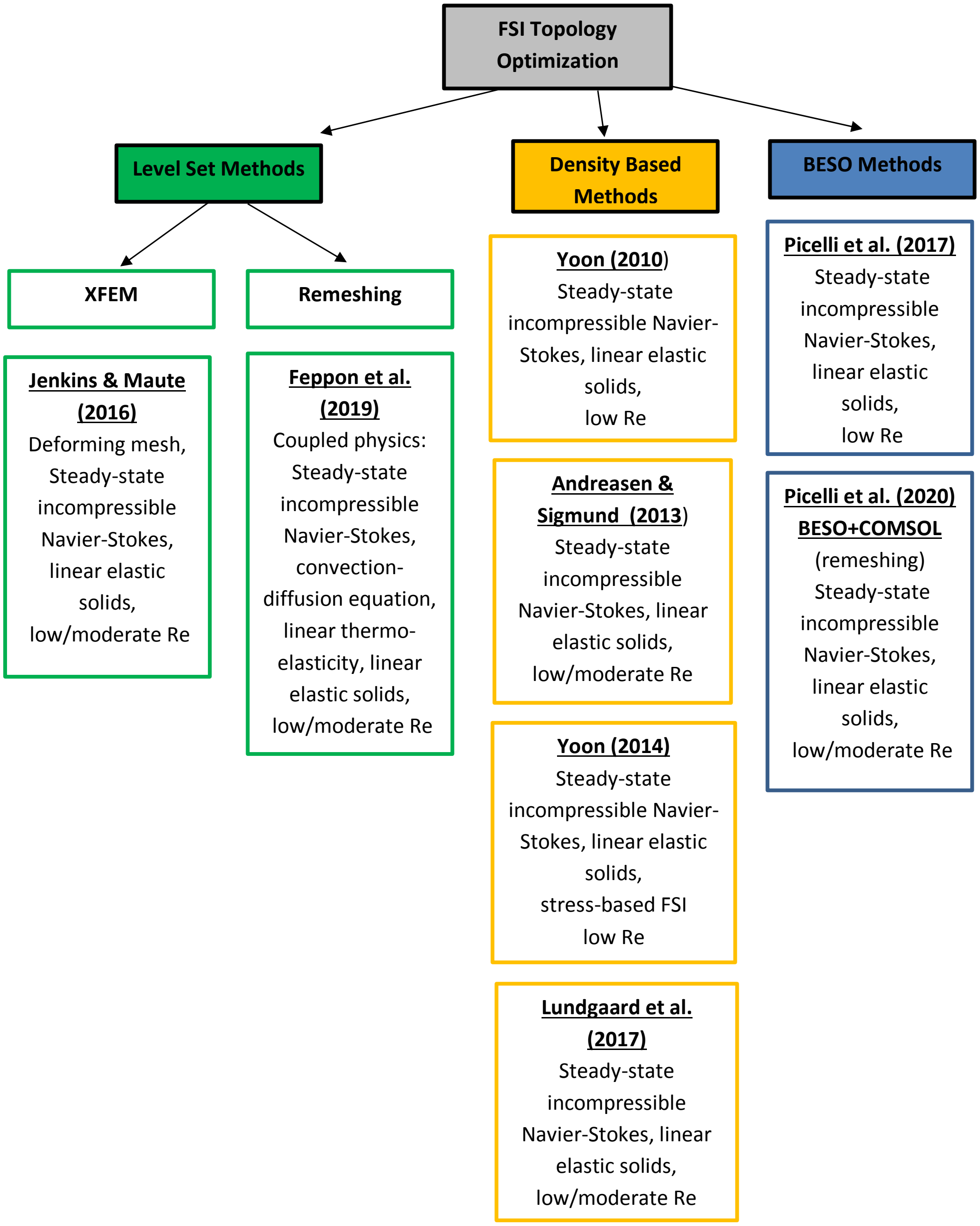

FSI Topology Optimization
Level Set Methods
Density Based Methods
BESO Methods
XFEM
Remeshing
Jenkins & Maute (2016)
Deforming mesh, Steady-state incompressible Navier-Stokes, linear elastic solids, low/moderate Re
Feppon et al. (2019)
Coupled physics: Steady-state incompressible Navier-Stokes, convection-diffusion equation, linear thermo-elasticity, linear elastic solids, low/moderate Re
Yoon (2010)
Steady-state incompressible Navier-Stokes, linear elastic solids, low Re
Andreasen & Sigmund (2013)
Steady-state incompressible Navier-Stokes, linear elastic solids, low/moderate Re
Yoon (2014)
Steady-state incompressible Navier-Stokes, linear elastic solids, stress-based FSI low Re
Lundgaard et al. (2017)
Steady-state incompressible Navier-Stokes, linear elastic solids, low/moderate Re
Picelli et al. (2017)
Steady-state incompressible Navier-Stokes, linear elastic solids, low Re
Picelli et al. (2020)
BESO+COMSOL
(remeshing) Steady-state incompressible Navier-Stokes, linear elastic solids, low/moderate Re

Figure 2.1: FSI topology optimization literature

Chapter 3

LSTO - RKPM Methodology

3.1 Level set topology optimization method

This section briefly summarizes the level set topology optimization method used in this study. More details of the method can be found in Hedges et al. (2017) and Picelli et al. (2018). In the level set topology optimization, the structural boundary is defined as the zero level set of an implicit function:

$$\begin{cases} \phi(\mathbf{x}) \geq 0 & \mathbf{x} \in \mathit{\Omega} \\ \phi(\mathbf{x}) = 0 & \mathbf{x} \in \mathit{\Gamma} \\ \phi(\mathbf{x}) < 0 & \mathbf{x} \notin \mathit{\Omega} \end{cases} \tag{3.1}$$

where ϕ is the level set function, Ω is the structural domain and Γ is the structural boundary. Commonly, the implicit function is initialized as a signed distance function (Wang et al., 2003; Allaire et al., 2004).

The structural boundary is optimized by iteratively solving the following Hamilton-Jacobi equation

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} + |\nabla \phi(\mathbf{x})| \, V_n(\mathbf{x}) = 0 \tag{3.2}$$

where t is a fictitious time domain for the level set evolution and V_n is the normal velocity.

The level set function at each node is updated by solving the following discretized Hamilton-Jacobi equation using an up-wind differential scheme,

$$\phi_r^{k+1} = \phi_r^k - \Delta t \left| \nabla \phi_r^k \right| V_{nr} \tag{3.3}$$

where r is a discrete node in the design domain, V_{nr} is the normal velocity at node r, k is the iteration number and $\left|\nabla\phi_r^k\right|$ is computed for each node using the Hamilton-Jacobi weighted essentially non-oscillatory method (HJ-WENO). To improve the computational efficiency, the level set update is restricted to nodes within a narrow band close to the boundary. This means that ϕ_r is given by a signed distance to the boundary only within the narrow band. To correct this effect, ϕ_r is periodically reinitialized to a signed distance function. For the reinitialization and velocity extension the fast-marching method is used (Sethian, 1996). The velocities required for the level set update are obtained by solving the linearized optimization problem,

$$\begin{aligned} &\underset{\mathit{\Omega}}{\textbf{minimize}} \quad && \tfrac{\partial f}{\partial \mathit{\Omega}^k}\Delta\mathit{\Omega}^k \\ &\textbf{subject to} \quad && \tfrac{\partial \mathrm{g}_m}{\partial \mathit{\Omega}^k}\Delta\mathit{\Omega}^k \leq -\mathrm{g}_m^k \end{aligned} \tag{3.4}$$

where f is the objective function, g_m is the m^{th} inequality constraint function, $\Delta\mathit{\Omega}^k$ is the update for the design domain Ω and g_m^k is the change in the m^{th} constraint. Shape derivatives that provide information about how the objective and constraint functions change with respect to the boundary movement, typically take the form of boundary integrals (Allaire et al., 2004):

$$\frac{\partial f}{\partial \mathit{\Omega}}\Delta\mathit{\Omega} = \Delta t \int_{\mathit{\Gamma}} s_f V_n d\mathit{\Gamma} \tag{3.5}$$

$$\frac{\partial \mathrm{g}_m}{\partial \mathit{\Omega}}\Delta\mathit{\Omega} = \Delta t \int_{\mathit{\Gamma}} s_{\mathrm{g}m} V_n d\mathit{\Gamma} \tag{3.6}$$

where s_f and $s_{\mathrm{g}m}$ are the shape sensitivity functions for the objective and the m^{th} constraint, respectively. The integrals in Eqs. (3.5) and (3.6) can be estimated as,

$$\frac{\partial f}{\partial \mathit{\Omega}}\Delta\mathit{\Omega} \approx \sum_{j=1}^{nb} \Delta t V_{nj} s_{f,j} l_j = \mathbf{C}_f \cdot \mathbf{V}_n \Delta t \tag{3.7}$$

$$\frac{\partial \mathrm{g}_m}{\partial \mathit{\Omega}}\Delta\mathit{\Omega} \approx \sum_{j=1}^{nb} \Delta t V_{nj} s_{\mathrm{g}m,j} l_j = \mathbf{C}_{\mathrm{g}_m} \cdot \mathbf{V}_n \Delta t \tag{3.8}$$

where j is a discrete boundary point, V_{nj}, $s_{f,j}$ and $s_{\mathrm{g}m,j}$ are the normal velocity and boundary point sensitivities for the objective and m^{th} constraint functions, respectively, at point j. l_j is the length of the local boundary around the boundary point j, $\mathbf{C}_f$ and $\mathbf{C}_{\mathrm{g}_m}$ are vectors containing the product of boundary lengths and shape sensitivities and $\mathbf{V}_n$ is the vector of normal velocities.

The linearized optimization problem is solved using the Simplex method (Arora, 2004) implemented via the GLPK library, version 4.65

(gnu.org/software/glpk/). The obtained optimum boundary point velocities are then used in Eq. (3.3) to update the level set function and this process is repeated until convergence is obtained. This method is implemented in the object oriented C++ code (Kambampati et al., 2018) and is available as opensource at http://m2do.ucsd.edu/software/.

3.2 Reproducing kernel particle method

This section outlines the reproducing kernel particle method (RKPM). For more details the readers are referred to Liu et al. (1995) and Chen et al. (1996).

3.2.1 Reproducing kernel approximation

To construct the RK approximation for a finite dimensional solution of the PDEs, the domain Ω is discretized by a set of nodes $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{NP}\}$, where $\mathbf{x}_I$ is the position vector of node I, and NP is the total number of particles, then the RK approximation of a function $u(\mathbf{x})$, denoted by $u^h(\mathbf{x})$, is expressed as,

$$u(\mathbf{x}) \approx u^h\left(\mathbf{x}\right) = \sum_{I=1}^{NP} \Psi_I\left(\mathbf{x}\right) u_I \tag{3.9}$$

where $\Psi_I(\mathbf{x})$ is the RK shape function at node I, and u_I is the corresponding nodal coefficients to be determined. Then the RK shape function can be expressed as

$$\Psi_I\left(\mathbf{x}\right) = C\left(\mathbf{x}; \mathbf{x} - \mathbf{x}_I\right) \Phi_a(\mathbf{x} - \mathbf{x}_I) \tag{3.10}$$

where $\Phi_a(\mathbf{x} - \mathbf{x}_I)$ is the kernel function centered at $\mathbf{x}_I$ with compact support size a. The kernel function controls the smoothness (continuity) and locality of the approximation function. The order of continuity in this approximation can be introduced without complexity. For example, the box function gives C^{-1} continuity, the hat function leads to C^0 continuity, the quadratic spline function results in C^1 continuity, the cubic B-spline function yields C^2 continuity and the quartic and quintic functions give C^3 and C^4 continuities, respectively. This is illustrated in fig. 3.1 where the C^0 linear B-spline (tent) kernel is compared with the C^2 cubic B-spline kernel function:

Linear B-spline (tent),

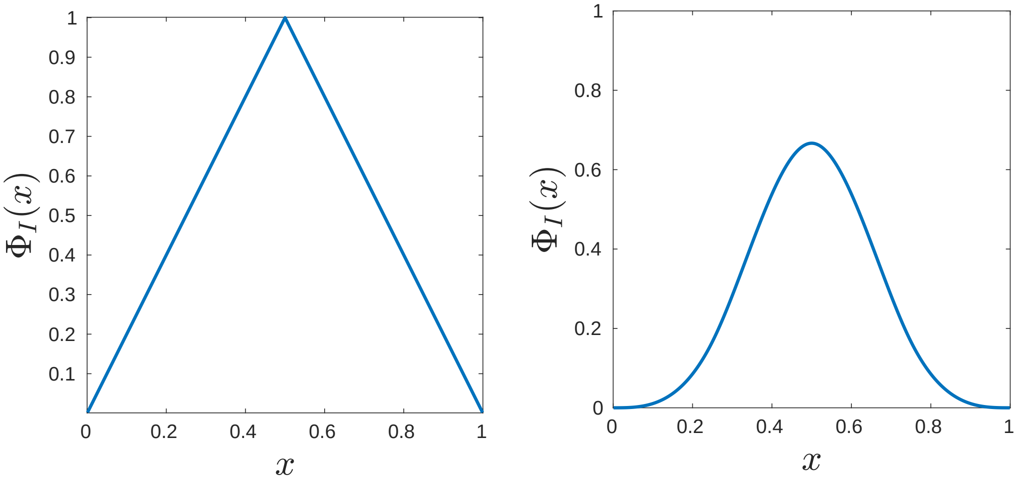
$$\Phi_a(\mathbf{x} - \mathbf{x}_I) = \begin{cases} 1 - z_I & for \quad z_I \leq \frac{1}{2} \\ \\ 0 & for \quad z_I > 1 \end{cases} \tag{3.11}$$

Cubic B-spline,

$$\Phi_a(\mathbf{x}-\mathbf{x}_I) = \begin{cases} \frac{2}{3} - 4z_I^2 + 4z_I^3 & for \quad z_I \leq \frac{1}{2} \\ \frac{4}{3} - 4z_I + 4z_I^2 - \frac{4}{3}{z_I}^3 & for \quad \frac{1}{2} < z_I \leq 1 \\ 0 & for \quad z_I > 1 \end{cases} \tag{3.12}$$

where z_I is defined as:

$$z_I = \|\mathbf{x}-\mathbf{x}_I\| / a_I \tag{3.13}$$



(a) (b)

Figure 3.1: Different kernel functions: (a) Linear B-Spline (C_0 continuity) and (b) Cubic B-Spline (C_2 continuity)

The higher-order smoothness achieved by the RK shape functions using different kernels has been demonstrated in many examples in the literature of RKPM. For example, it has been shown that a smooth transition in material properties can be achieved by the smooth RK approximation in the modelling of biomaterials (Chen et al., 2017a). Other examples can be found in the context of contact mechanics (Wang et al., 2014) and large deformation analysis of nonlinear structures (Chen et al., 1996). Numerical comparison of using different kernel functions with different levels of smoothness and locality can be found in Huang et al. (2019) and Belytschko et al. (1994).

The correction function $C(\mathbf{x};\mathbf{x}-\mathbf{x}_I)$ is introduced to ensure the completeness of the RK approximation by enforcing the following n-th order reproducing conditions

$$C(\mathbf{x};\mathbf{x}-\mathbf{x}_I)=\sum_{|\beta|=0}^{n}(\mathbf{x}-\mathbf{x}_I)^{\beta}b_{\beta}(\mathbf{x})=\mathbf{H}^{T}(\mathbf{x}-\mathbf{x}_I)\mathbf{b}(\mathbf{x}) \tag{3.14}$$

where n is the specified order of completeness, which is related to the order of consistency in solving PDEs, $\beta=(\beta_1,\beta_2,\beta_3)$ is the three-dimensional index and $|\beta|\equiv\sum_{i=1}^{3}\beta_i$.
$b_\beta(\mathbf{x})$ is the corresponding coefficient of the monomials $(\mathbf{x}-\mathbf{x}_I)^{\beta}$, where $(\mathbf{x}-\mathbf{x}_I)^{\beta}\equiv(x_1-x_{1I})^{\beta_1}(x_2-x_{2I})^{\beta_2}(x_3-x_{3I})^{\beta_3}$, and $\mathbf{H}^{\mathrm{T}}(\mathbf{x}-\mathbf{x}_I)$ can be expressed as

$$\mathbf{H}^{\mathrm{T}}(\mathbf{x}-\mathbf{x}_I)=[1,\quad x-x_I,\quad y-y_I,\quad z-z_I,\quad (x-x_I)^2,\quad \ldots,\quad (z-z_I)^n] \tag{3.15}$$

and $\mathbf{b}^{T}(\mathbf{x})$ is the unknown coefficient vector which can be computed by the n-th order reproducing conditions

$$\sum_{I=1}^{NP}\Psi_I(\mathbf{x})\mathbf{x}_I^{\beta}=\mathbf{x}^{\beta},\quad |\beta|\le n \tag{3.16}$$

and by substituting Eq. (3.10) and Eq. (3.14) into Eq. (3.16). The coefficients $\mathbf{b}(\mathbf{x})$ can be determined as

$$\mathbf{b}(\mathbf{x})=\mathbf{M}^{-1}(\mathbf{x})\mathbf{H}(\mathbf{0}) \tag{3.17}$$

where $\mathbf{M}(\mathbf{x})$ is the moment matrix. The support size a_I is typically defined as:

$$a_I=ch_I \tag{3.18}$$

where c is the normalized support size, chosen between 1.5 and 2.0 in practice, and h_I is the nodal spacing associated with point $\mathbf{x_I}$ (Huang et al. 2019). This choice for the support has been shown to be stable in the literature (Belytschko et al., 1994; Chen et al., 1996; Liu et al., 1997). The substitution of Eq. (3.17) into Eq. (3.14) results in

$$\mathbf{M}(\mathbf{x})=\sum_{I=1}^{NP}\mathbf{H}(\mathbf{x}-\mathbf{x}_I)\mathbf{H}^{\mathrm{T}}(\mathbf{x}-\mathbf{x}_I)\mathbf{\Phi}_a(\mathbf{x}-\mathbf{x}_I) \tag{3.19}$$

where $\mathbf{H}^{\mathrm{T}}(\mathbf{x}-\mathbf{x}_I)$ is the vector of monomial basis functions. Through the combination of Eqs. (3.17), (3.14), and (3.10), the RK shape function in Eq. (3.10) can then be obtained:

$$\Psi_I(\mathbf{x})=\mathbf{H}^{\mathrm{T}}(\mathbf{0})\mathbf{M}^{-1}(\mathbf{x})\mathbf{H}(\mathbf{x}-\mathbf{x}_I)\mathbf{\Phi}_a(\mathbf{x}-\mathbf{x}_I) \tag{3.20}$$

By construction, the RK shape function in Eq. (3.20) satisfies the following n-th order reproducing conditions,

$$\sum_{I}^{NP} \Psi_I(\mathbf{x}) x_I^i y_I^j z_I^k, \quad 0 \leq i + j + k \leq n \tag{3.21}$$

where n is the specified order of completeness, which determines the order of consistency in the solution of PDEs. In this work, linear basis is employed for which both the zero-th and first-order reproducing conditions are satisfied.

An illustration of the RK discretization and shape functions with circular supports is shown in Fig. 3.2. It should also be noted the support domains need not to be circular rather different shapes may be used.

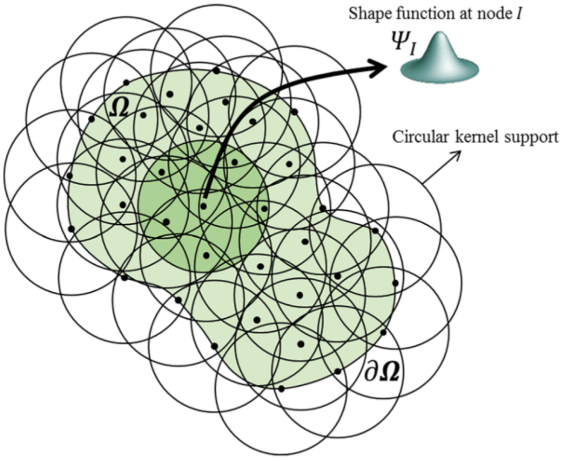


Figure 3.2: RK discretization for 2D domain illustrating the particles and shape functions with circular supports overlapping over the domain (Huang et al., 2019)

3.2.2 Galerkin approximation and formulation

In this section the Galerkin framework for RKPM is outlined. Enforcement of essential boundary conditions is discussed first since the formulations of the weak form discussed later depend on the method selected for the imposition of boundary conditions.

3.2.3 Imposing boundary conditions

The RK shape functions lack the Kronecker delta property. For this reason applying the essential boundary conditions is not straightforward. Several ap-proaches have been proposed for this purpose. In general these can be classified into two categories: Either strong enforcement at the nodes on the essential boundary or weak enforcement of conditions along the essential boundary (Chen et al., 2017b). Strong enforcement methods such as the mixed transformation method and the boundary singular kernel method (Chen and Wang, 2000) modify the standard meshfree shape functions whereas weak methods such as the Lagrange multiplier method (Belytschko et al., 1994), the penalty method (Zhu and Atluri, 1998) and Nitsche's method (Griebel and Schweitzer, 2002) are based on a modifi-cation of the weak form. The Lagrange multiplier and Nitsche's method are implemented in this work. Both methods have their benefits and drawbacks. The Lagrange method is one of the most commonly used because of it is general and can be straightforwardly implemented for a wide range of problems. In fact, there is no need to even know the weak form with this method. It essentially introduces a new unknown function, namely, the Lagrange multiplier for which the physical meaning corresponds to the flux (traction in a mechanical problem) along the essential boundary. However, with this method the dimension of the resulting system of equations is increased. Furthermore, the interpolation spaces for the Lagrange multiplier and the principal unknown must satisfy an inf-sup, Babuska-Brezzi stability condition to ensure convergence of the approximation. Thus, the choice of an appropriate interpolation for the Lagrange multiplier requires a careful treatment in certain situations (Fernandez-Mendez and Huerta, 2004). The Nitsche's method on the other hand, can be obtained by combining the Lagrange multiplier and penalty methods, and only requires the choice of a scalar parameter similarly to the penalty method. However, Nitsche's method does not suffer from ill-conditioning in contrast to the penalty method in which large values of this parameter can lead to ill-conditioned systems of equations. Furthermore, it was shown in the work by Fernandez-Mendez and Huerta (2004), that the Nitsche's method yields optimal convergence. The difficulty with the Nitsche's method is that the modification of the weak form is different for each particular problem. This means that generalization to any problem is not as straightforward as for the Lagrange multiplier method. Nevertheless, the deriva-tion under the Nitsche's method for well-known problems can be found in the literature. For example for the Navier-Stokes problem (Becker, 2002), the Stokes problem (Freud and Stenberg, 1995), and elasticity problems (Hansbo and Larson, 2002).

Galerkin formulation under Nitsche's method

In this work we consider the case for linear elasticity

$$\begin{aligned} &\sigma_{ij,j} + b_i = 0 && \text{on} \quad \Omega \\ &\sigma_{ij} n_j = h_i && \text{on} \quad \Gamma^{h_i} \\ &u_i = g_i && \text{on} \quad \Gamma^{g_i} \end{aligned} \tag{3.22}$$

where u_i is the displacement, σ_{ij} is the Cauchy stress, n_j is the surface normal,

b_i is the body force and h_i and g_i are the prescribed traction and displacement on the Neumann and Dirichlet boundaries, Γ^{h_i} and Γ^{g_i}, respectively. Nitsche's method results in the following weak form of Eq. 3.22 (Chen and Wang, 2000)

$$\begin{aligned}\int_{\Omega} \delta\varepsilon_{ij} C_{ijkl}\varepsilon_{kl}\mathrm{d}\Omega = \int_{\Omega}\delta u_i b_i \mathrm{d}\Omega + \int_{\Gamma^{h_i}} \delta u_i h_i \mathrm{d}\Gamma + \int_{\Gamma^{g_i}} \delta u_i \lambda_i \mathrm{d}\Gamma \\ + \int_{\Gamma^{g_i}} \delta\lambda_i (u_i - g_i)\mathrm{d}\Gamma + \beta \int_{\Gamma^{g_i}} \delta u_i (u_i - g_i)\mathrm{d}\Gamma\end{aligned} \tag{3.23}$$

where C_{ijkl} is the elasticity tensor, ε_{ij} is the strain, λ_i is the Lagrange multiplier which for elasticity problems in Nitsche's method is taken as the surface traction, i.e., $\lambda_i = \sigma_{ij} n_j$, and β is the penalty taken as $\beta = \beta_{nor}\frac{E}{\bar{h}}$ with β_{nor} the normalized penalty parameter, E the Young's modulus and $\bar{h}$ the average of nodal spacing.

Introducing the RK approximation for u_i in Eq. 3.9 into Eq. 3.23 results in the following matrix equation

$$\delta \mathbf{u}^T(\mathbf{K}\mathbf{u} - \mathbf{f}) = 0 \tag{3.24}$$

where

$$\begin{aligned}\mathbf{K}_{IJ} = \int_{\Omega} \mathbf{B}_I^T \mathbf{C} \mathbf{B}_J \mathrm{d}\Omega + \beta \int_{\Gamma^{g_i}} \mathbf{\Psi}_I^T \mathbf{S} \mathbf{\Psi}_J \mathrm{d}\Gamma \\ - \int_{\Gamma^{g_i}} \mathbf{B}_I^T \mathbf{C} \boldsymbol{n} \mathbf{S} \mathbf{\Psi}_J \mathrm{d}\Gamma - \int_{\Gamma^{g_i}} \mathbf{\Psi}_I^T \mathbf{S}^T \boldsymbol{n}^T \mathbf{C} \mathbf{B}_J \mathrm{d}\Gamma\end{aligned} \tag{3.25}$$

$$\mathbf{f}_I = \int_{\Omega} \mathbf{\Psi}_I^T \mathbf{b} \mathrm{d}\Omega + \int_{\Gamma^{h_i}} \mathbf{\Psi}_I^T \mathbf{h} \mathrm{d}\Gamma + \beta \int_{\Gamma^{g_i}} \mathbf{\Psi}_I^T \mathbf{S}\mathbf{g} \mathrm{d}\Gamma - \int_{\Gamma^{g_i}} \mathbf{B}_I^T \mathbf{C} \boldsymbol{n} \mathbf{S} \mathbf{g} \mathrm{d}\Gamma \tag{3.26}$$

where $\mathbf{K}$ is the stiffness matrix, $\mathbf{\Psi}$ represents the matrix of the RK shape functions, $\mathbf{B}$ is the shape function gradient matrix, $\mathbf{C}$ is the elasticity tensor, $\mathbf{u}$ is the vector of structural displacements, $\mathbf{f}$ represents the external load due to the body force, $\mathbf{b}$ and surface load, $\mathbf{h}$. Γ^{h_i} is the Neumann boundary and Γ^{g_i} is the Dirichlet boundary with a prescribed displacement, $\mathbf{g}$, to be enforced. The term $\mathbf{n}$ is the surface unit normal on the essential boundary, β is the penalty taken as $\beta = \beta_{nor}\frac{E}{\bar{h}}$ with β_{nor} the normalized penalty parameter, E the Young's modulus and $\mathbf{S}$ is used to impose each component of the boundary diplacement by setting $s_i = 0$ or 1.

The discretization of the Nitsches weak form leads to a system of equations with the same size as $\mathbf{K}$, and whose matrix is symmetric and positive defi- nite. This is not the case for the Lagrange multiplier method as shown in the next section. However, with Nitsches method the deduction of the weak form and generalization of the implementation for other problems is not as straightforward as for the method of Lagrange multipliers (Fernandez-Mendez and Huerta, 2004).

Galerkin formulation under Lagrange multipliers method

Using the Lagrange multiplier method to impose the essential boundary conditions, the discrete equations corresponding to the weak formulation of 2-dimensional linear elasticity can be expressed by

$$\begin{bmatrix} \mathbf{K} & \mathbf{G^T} \\ \mathbf{G} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{q} \end{bmatrix} \tag{3.27}$$

where

$$\mathbf{K}_{IJ} = \int_{\Omega} \mathbf{B}_I^T \mathbf{D} \mathbf{B}_J d\Omega \tag{3.28}$$

$$\mathbf{G}_{IJ} = -\int_{\Gamma^{g_i}} \boldsymbol{\Psi}_I \mathbf{N}_J d\Gamma \tag{3.29}$$

$$\mathbf{f}_I = \int_{\Omega} \boldsymbol{\Psi}_I \mathbf{b} d\Omega + \int_{\Gamma^{h_i}} \boldsymbol{\Psi}_I \mathbf{h} d\Gamma \tag{3.30}$$

$$\mathbf{q}_I = -\int_{\Gamma^{g_i}} \mathbf{N}_I^T \mathbf{g} d\Gamma \tag{3.31}$$

$$\boldsymbol{\Psi}_I = \begin{bmatrix} \Psi_I & 0 \\ 0 & \Psi_I \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_x \\ b_y \end{bmatrix} \quad \mathbf{h} = \begin{bmatrix} h_x \\ h_y \end{bmatrix} \quad \mathbf{g} = \begin{bmatrix} g_x \\ g_y \end{bmatrix} \tag{3.32}$$

$$\mathbf{B}_I = \begin{bmatrix} \Psi_{I,x} & 0 \\ 0 & \Psi_{I,y} \\ \Psi_{I,y} & \Psi_{I,x} \end{bmatrix} \quad \mathbf{N}_I = \begin{bmatrix} N_I & 0 \\ 0 & N_I \end{bmatrix} \quad \boldsymbol{n} = \begin{bmatrix} n_x & 0 \\ 0 & n_y \\ n_y & n_x \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \tag{3.33}$$

$$\mathbf{D} = \frac{\mathrm{E}}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix} \tag{3.34}$$

In the above equations, **G** is a matrix to enforce boundary conditions using the Lagrange multipliers method. Matrix **D** is the matrix of material elastic constants with E, the Young's modulus and Poisson's ratio ν. $\mathbf{N}_I$ in Eq. (3.31) and Eq. (3.33) is the standard Lagrangian interpolant along the boundary to be enforced.

Chapter 4

Design-Dependent Hydrostatic Pressure

In this chapter the RKPM based LSTO is employed to solve for moving hydrostatic pressure. Several necessary ingredients are first implemented. These include a technique for numerical integration of the integrals in the weak form of the Galerkin approximation. An algorithm for separating the loading portion of the boundary from the load-free portion of the interior holes is also presented. The examples solved here, include both steady and variable pressure fields and the optimization problem considered is compliance minimization under a volume constraint.

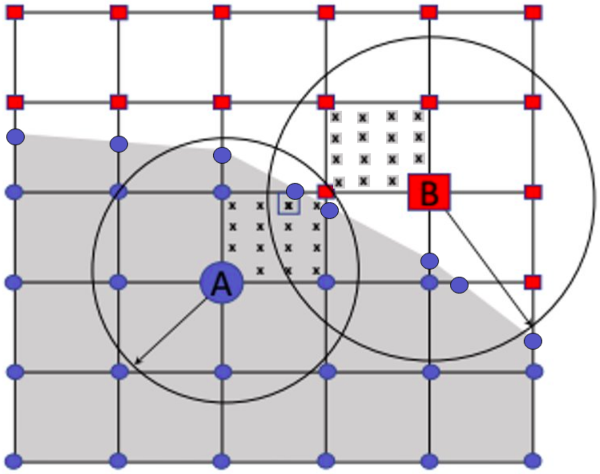
4.1 Domain integration using a background mesh approach

Since a Galerkin meshless method is used, domain integration is required. In this chapter, Gauss quadrature on a fixed rectangular background mesh is employed. As has been shown by Dolbow and Belytschko (1999), when the Gauss quadrature is used for the domain integration significant errors may arise when the background mesh does not coincide with the support domains. This becomes particularly difficult when circular supports are being used. As has been shown by Dolbow and Belytschko (1999), this can be improved by using sufficiently high quadrature rules. Following this work, each background cell has 4×4 quadrature points in our implementation to ensure accuracy. Integration schemes based on nodal integration rather than Gauss integration have also been proposed in the literature such as the stabilized conforming nodal integration (SCNI) by Chen et al. (2001). Nodal integration is explained and implemented for LSTO in Ch. 5. It is important to note that RKPM is not limited to a regular background mesh. Furthermore, the positions of the particles within the design domain are indepen-dent from the background mesh. This very attractive feature is one of the main advantages exploited in this work.

4.1.1 Modeling the Interface

There are two different ways to perform Gauss integration on a background grid. The first is to create cells that are conforming to the structure and integrate only the solid region. To avoid this process here, the alternative of a fixed background grid over the entire domain is used. In this scheme, integration points and particles also exist in the void region throughout optimization as shown in Fig. 4.1. The void region is modeled as a weak material with a very low Young's modulus value.

It is important to clarify at this point, that with the interpolation scheme proposed here, the crispiness of the boundary is not maintained onto the computational domain. The crisp boundary represented by the level set method is smeared out on the computational domain by interpolating the Young's modulus using the RK shape functions as the interpolation functions. This is in contrast to methods such as XFEM which can capture the sharp interface on fixed-grids. Essentially the scheme described here is similar to using the Ersatz approximation to smear the interface on the computational domain.



B Particle with level set function value $\phi < 0$ and Young's modulus $10^{-4} \times$ E

A Particle with level set function value $\phi \geq 0$ and Young's modulus E

x Computational point (Gauss point) covered by support domains of A and B.

Figure 4.1: Interpolating Young's Modulus at Computational Point (Gauss point)

Figure 4.1 illustrates how the moving boundary discontinuity is represented. The level set function defines whether a point lies inside or outside the structure, and the Young's modulus values for solid and weak materials are also known. Further, an interpolation of the Young's modulus at quadrature points is performed using the RK shape functions to achieve a smooth Young's modulus distribution. This in turn leads to a smooth boundary sensitivity field for a better convergence. The Gauss points that lie near a structural boundary, are covered by the support domains of the particles in both the solid and void regions as shown in

Fig. 4.1. At these points, Young's modulus E is computed by RK approximation using the level set function values at the particles whose RK support domains cover the Gauss point. Particles with $\phi \geq 0$ (For example particle A in Fig. 4.1) have a Young's modulus value E, whereas particles with $\phi < 0$ (For example particle B in Fig. 4.1), have a Young's modulus value equal to 10^{-4}E. The particles at the boundary points are placed there in order to apply the design-dependent loads directly at the boundary. So their main purpose is to address the design-dependency of the load. They also assist in boundary point sensitivity computation which is required by the level set method used in this work. Previous implementations of this level set method with the Ersatz approximation required a least squares interpolation from the Gauss points of the elements (Picelli et al., 2018) to compute boundary sensitivities whereas this can be avoided with the boundary particles. These boundary particles have the same effect as the solid particles on the Young's modulus of a Gauss point in their viscinity. They are assigned a Young's modulus value equal to E as shown in Fig. 4.1, and they participate in the Young's modulus interpolation through the RK shape functions just like the interior solid particles in the support domain of each Gauss point.

$$\mathrm{E \text{ at node } I :} \begin{cases} \mathrm{E}(\mathbf{x}_\mathrm{I}) = \mathrm{E} & \text{if} \quad \phi \geq 0 \\ \\ \mathrm{E}(\mathbf{x}_\mathrm{I}) = 10^{-4} \times \mathrm{E} & \text{if} \quad \phi < 0 \end{cases} \tag{4.1}$$

where $\mathrm{E}(\mathbf{x}_\mathrm{I})$ is the Young's modulus associated with the I^{th} node and E at Gauss point $\mathbf{x}_{\mathrm{gp}}$ is computed as

$$\mathrm{E}(\mathbf{x}_{\mathrm{gp}}) = \sum_{\mathrm{I} \in \mathrm{G}_{\mathrm{gp}}} \Psi_\mathrm{I}(\mathbf{x}_{\mathrm{gp}}) \mathrm{E}_\mathrm{I} \tag{4.2}$$

where G_{gp} is the node set containing all nodes with their support covering the evaluation point $\mathbf{x}_{\mathrm{gp}}$ and Ψ_I represents the RK shape function. The obtained E is then used to compute the **D** matrix in Eq. 3.34.

To illustrate the resulting Young's modulus distribution obtained from the interpolation in Eq. 4.2, a 1D example is shown in Fig. 4.2. For this example, 500 particles are placed uniformly with a dimensionless spacing between them equal to 0.01. For this 1D case the boundary is placed at x=3 as indicated by the red dashed line in the figure. For all particles on the left of the boundary, Young's modulus is equal to 1 whereas for particles on the right of the boundary Young's modulus is equal to 0.0001. As shown in the figure, the interpolated Young's modulus has a smooth transition from the maximum to the minimum value near the boundary region.

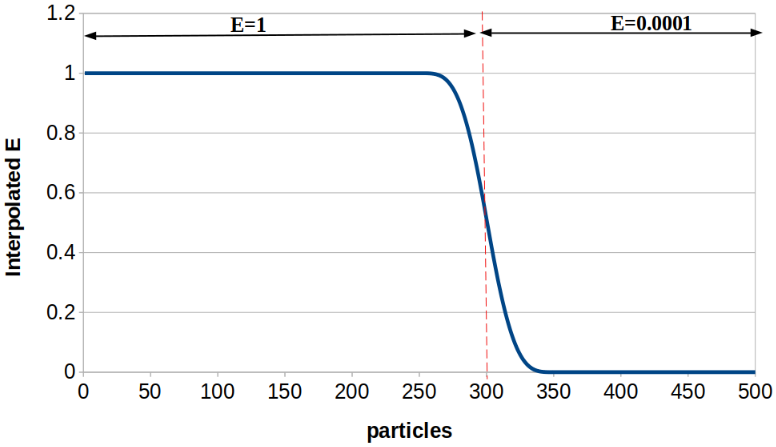


Figure 4.2: 1D illustration of interpolated Young's modulus

Later in Ch. 5, a different methodology is adopted for integration which possesses several advantages over the implementation presented here especially in terms of computational efficiency, as it will be discussed. Nevertheless, the current implementation has been shown to be robust and gave good results. Moreover, the implementation of this methodology allowed for comparison with the different implementation discussed later, which was valuable for a deeper understanding of the pros and cons of different LSTO-RKPM implementations.

4.1.2 Methodology

At every iteration, particles are added on the boundary at the intersection points between the background mesh and the boundary. As explained in Sec. 4.1.1, these boundary particles are placed to allow for a direct application of the design-dependent loads at the interface. They also assist in boundary point sensitivity computation which is required by the level set method used in this work. Previous implementations of this level set method with the Ersatz approximation required a least squares interpolation from the Gauss points of the elements (Picelli et al., 2018) to compute boundary sensitivities whereas this can be avoided with the boundary particles. Boundary particles from the previous iteration are removed. This means that particles do not move to different locations between iterations but instead they are generated. It is thus only required to compute the intersection points and not track the particles. The process of generating new particles on the boundary is shown in Fig. 4.3. Initially, domain particles are placed at the nodal positions of the background mesh. These particles are shown with blue circles in Fig. 4.3 (a). At every iteration during the optimization process, these domain particles are generated first. As the boundary moves, domain particles are separated into particles inside the structure shown in blue circles in Fig. 4.3 (b) and particles outside the structure shown in red circles. Particles outside the structure are kept to be used in the domain integration scheme with a fixed background mesh described in section 4.1. At every iteration particles are added to the structural boundary. In Fig. 4.3(b) these boundary particles are indicated by blue circles at the points where the updated boundary crosses the edges of the background mesh. Finally, if it happens that a domain particle is near a boundary particle, the domain particle is removed as shown by the yellow ”x” symbols. In the numerical tests, the removal does not influence the final topology but it produces a faster convergence. In fact, even if these particles were not removed, the final topologies do not change. The minimum distance between particles was set to 0.05 times the length of a background element.

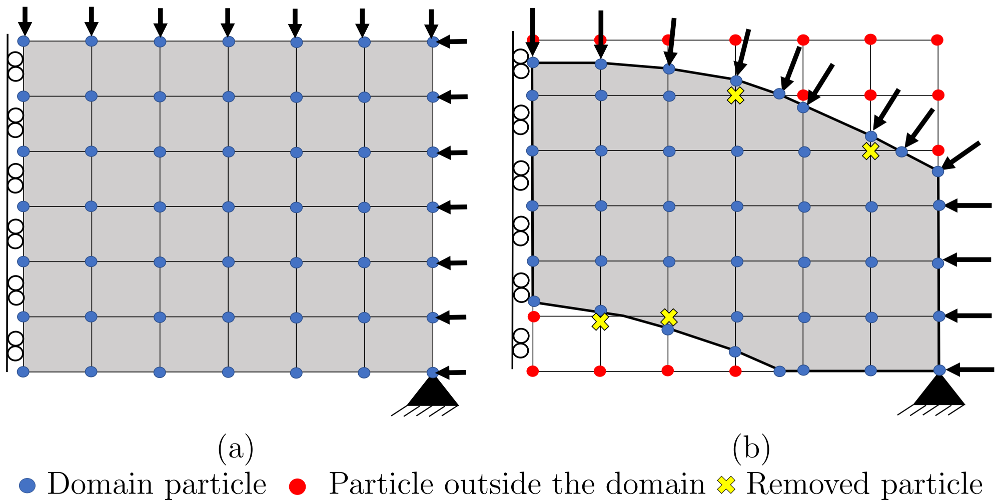
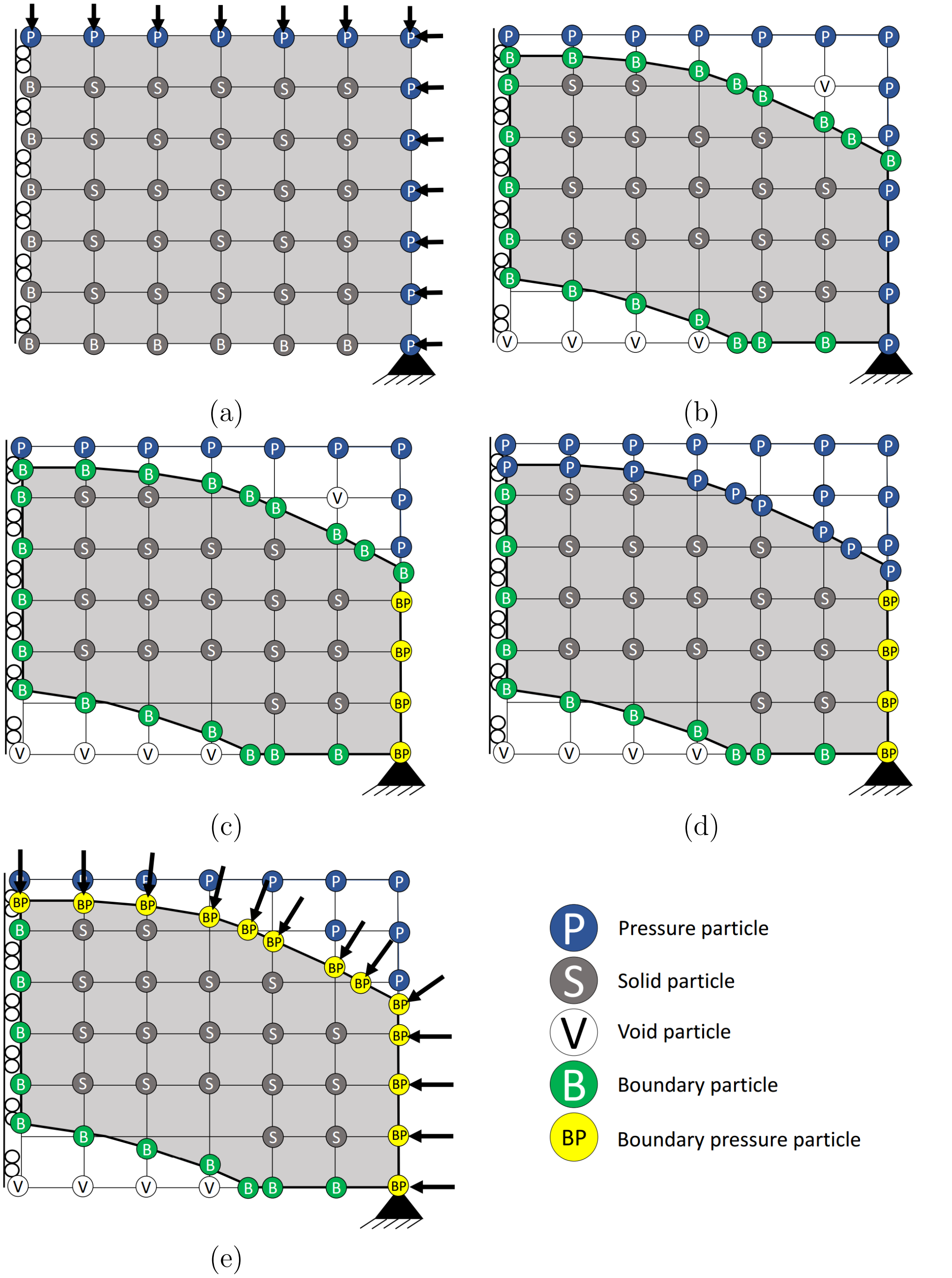


Figure 4.3: (a) Initial particle distribution and (b) Particle distribution after an optimization iteration

To identify the part of the boundary on which the pressure load is to be applied a process similar to the fluid flooding proposed by Chen and Kikuchi (2001) is used here. The difference in this work is that instead of elements we use the particles to identify the loading surface. Before optimization starts, the particles on the boundary segments carrying the initial pressure loads are marked as ”pressure” particles. Depending on the signed distance value at the particles and the position of the pressure boundary, the following particle types emerge in addition to pressure particles:

- solid particles that lie inside the structure,
- void particles that lie outside the structure,
- boundary particles that lie on the structural boundary
- boundary pressure particles that indicate where a pressure load should be applied

Figure 4.4 illustrates the algorithm of moving boundaries. Fig. 4.4(a) shows the initial arrangement. The particles in blue with symbol “P” indicate “pressure” particles, and the gray particles with “S” indicate solid particles. When the boundaries are updated, Fig. 4.4(b), the signed distance, ϕ, at the particles is calculated to classify them into solids(“S”) with $\phi > 0$, voids(“V”) with $\phi < 0$ and boundary(“B”) particles with $\phi = 0$. The initial pressure particles remain unchanged throughout optimization except for those initial pressure particles that also lie on the boundary. These become boundary pressure particles in subsequent iterations (“BP”) as shown in Fig. 4.4(c). The reason for changing these boundary particles into “boundary pressure” instead of “pressure” is to stop the pressure load from advancing to the pressure-free portion of the boundary as will become apparent shortly. Also note that boundary pressure particles only appear after the first iteration, this is why they do not yet exist in Fig. 4.4(a). In Fig. 4.4(d), void or boundary particles that have pressure neighbors are transformed into pressure particles and this advances the pressure region. Neighboring between grid nodes and boundary particles is known by their connec-tivity on the level set mesh. The process is continued until the pressure region comes into contact with the boundary. Once the “P” type encounters a boundary particle, the particle is turned into “BP” type as shown in Fig. 4.4(e). Had these particles remained as “pressure” particles, then all the “B” particles would eventually turn into “P” thus transferring the load also in the portion of the boundary that must remain pressure-free (left and bottom boundary particles). Thus, the separate “BP” type is used to stop further advancement of the pressure region on the load-free part of the boundary. Finally, the pressure load is applied to the part of the boundary on which “BP” particles lie.



(a)

(b)

(c)

(d)

(e)

Figure 4.4: (a) Initial particle arrangement, (b) boundary evolves and particle type assigned based on particle signed distance value, (c) initial boundary pressure particles are identified, (d) pressure region advances and (e) boundary pressure particles are identified and pressure load is applied on boundary segments including these particles.

An important point is that in this work, the same mesh used for discretizing the level set function is also used as the background mesh for the domain integration in RKPM. For the particular case where an element is cut twice by a boundary, the following two configurations may arise:

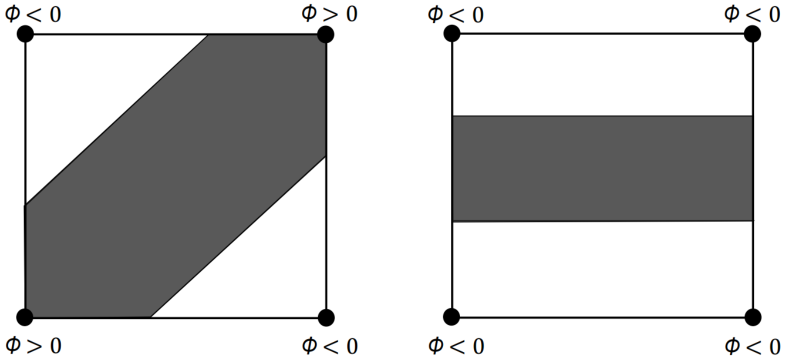


Figure 4.5: Elements cut twice by the boundary: (a) Possible configuration and (b) Not possible configuration

The configuration shown in Fig. 4.5(b) is not possible due to the linear interpolation of the level set function. The configuration shown in Fig. 4.5(a) is possible however, and the scheme for identifying the pressure boundary based on particle types, is able to correctly apply the pressure loads for this instance because the load is transferred from point to point rather than from element to element. Thus, the load transfer stops once the pressure region encounters the first of the two segments within the element and does not proceed to the second segment.

We define in this problem that the outside pressure cannot be transferred to holes inside the structure. However, there is nothing fundamentally limiting in the methodology to consider problems with inner pressure like pressure vessels.

4.2 Hydrostatic pressure examples

Three benchmark examples for design-dependent pressure loading problems are investigated and compared with the literature. For all examples, a plane stress condition is assumed. For the solid material Young's modulus is equal to 1 whereas for the void region Young's modulus is set to be 10^{-4}. A Poisson's ratio of 0.3 is used. For the domain integration, 4×4 Gauss points are used in each background cell. A constant normalized support size of 1.5 times the size of a background cell was used for all particles as this was found to be the minimum support size that yields good convergence rate. This constant support size worked well even with the addition of new boundary particles at every iteration. For support sizes beyond the selected size and up to 3.5 times the size of a background cell there was no change in the final solution. For each example a number of initial hole configurations were investigated. Consistent solutions were obtained from all cases. The convergence of the objective is checked over 5 consecutive iterations and the absolute tolerance is 0.001.

4.2.1 Compliance minimization formulation

Topology optimization in this study considers the well known compliance minimization problem

$$\begin{aligned} \mathbf{min} \quad & J = l(u) \\ \mathbf{s.t.} \quad & a(u,v) = l(v), \quad \forall \ v \in U \\ & V_s(\Omega) - \bar{V} \leq 0 \end{aligned} \tag{4.3}$$

where the energy bilinear functional $a(\mathbf{u}, \boldsymbol{v})$ and the load linear form $l(\boldsymbol{v})$ are defined as:

$$a(\mathbf{u}, \boldsymbol{v}) = \int_{\Omega} \{\boldsymbol{\epsilon}(\mathbf{u})\}^T : \mathbf{D} : \{\boldsymbol{\epsilon}(\boldsymbol{v})\} \, d\Omega \tag{4.4}$$

$$l(\boldsymbol{v}) = \int_{\Gamma_N} \boldsymbol{p} \cdot \boldsymbol{v} d\Gamma \tag{4.5}$$

Here, U is the space of kinematically admissible displacement fields. $V_s(\Omega)$ is the volume fraction of the structure with respect to the design domain, $\bar{V}$ is the maximum allowed volume fraction, $\boldsymbol{\epsilon}$ is the strain tensor, $\boldsymbol{v}$ is the virtual displacement, Γ_N is the Neumann boundary on which the pressure load is applied and p is the pressure load. The pressure load is assumed to be constant although the method can be generalized for varying pressure.

$$\boldsymbol{p} = -p_0 \boldsymbol{n} \tag{4.6}$$

where p_0 is the constant magnitude of the pressure load, and $\boldsymbol{n}$ is the surface normal.

4.2.2 Boundary sensitivities

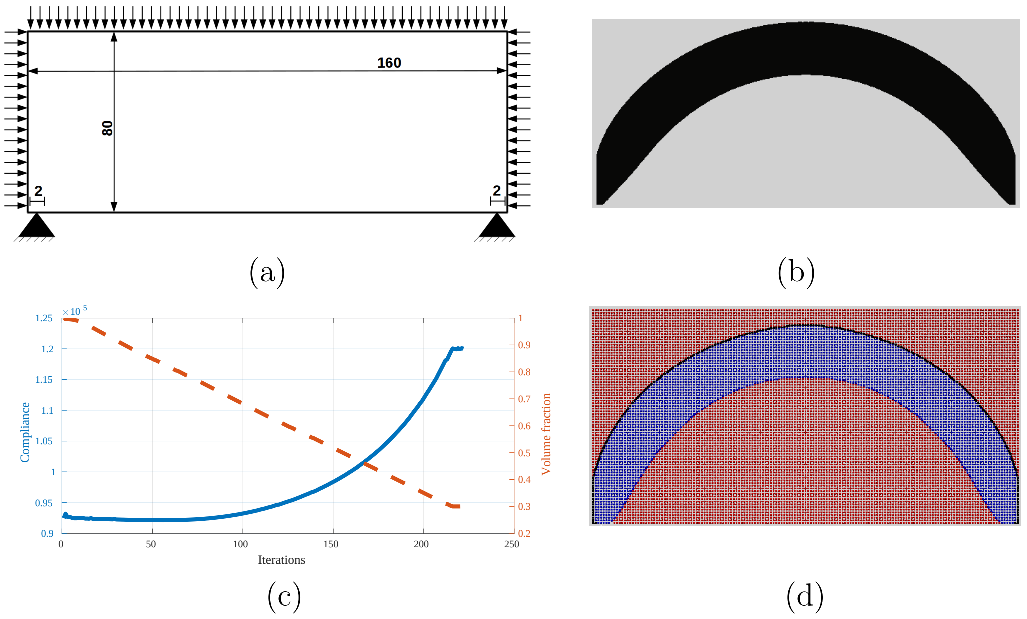
Shape sensitivities for the structural compliance function when the surface load is a pressure load oriented in the direction of the normal vector were derived by Allaire et al. (2004) as,

$$\frac{\partial f}{\partial \mathrm{x}} = \int\limits_{\Gamma_s} \left[-2\mathrm{div}\left(p_0 \boldsymbol{u}\right) - \boldsymbol{\sigma}(\boldsymbol{u}) \cdot \boldsymbol{\epsilon}\left(\boldsymbol{u}\right)\right] V_n d\Gamma_s, \tag{4.7}$$

where p_0 is the pressure load, $\boldsymbol{u}$ is the displacement, $\boldsymbol{\sigma}(\boldsymbol{u}) = \mathbf{D} : \boldsymbol{\epsilon}(\boldsymbol{u})$ is the stress tensor, $\boldsymbol{\epsilon}(\boldsymbol{u})$ is the strain tensor and Γ_s represents the structural boundary. For the boundary points not on the pressure surface the divergence term becomes zero. As points on the boundary are covered by the support domains of particles from both the solid and void regions, Young's modulus is interpolated using Eq. (4.1) and Eq. (4.2) given in Section 4.1. As can be seen in Eq. (4.7), the shape sensitivity includes stress and strain terms. The higher order continuity in the approximation of RKPM is another advantage over the finite element method with linear elements where stress is discontinuous across the element edges. The RK approximation has the advantage of employing higher order smoothness with arbitrary order consistency, which avoids stress discontinuity in FEM with linear elements. Thus, smooth stresses can be obtained directly at the boundary points without any additional treatment such as stress recovery.

4.2.3 Arch structure

The arch example is a popular example used to validate topology optimization with design dependent pressure loads (Sigmund and Clausen, 2007; Xia et al., 2015; Picelli et al., 2019; Emmendoerfer et al., 2018). The model considered here is shown in figure 4.6(a). A background mesh consisting of 160×80 rectangular cells is used. The final volume fraction is set to 30%. The structure is subjected to a constant pressure load $p = 1$ on the top, left and right edges. The arch-like optimum solution in Fig. 4.6(b) agrees well with those obtained by previous works that considered this example (Sigmund and Clausen, 2007; Picelli et al., 2015a, 2019; Xia et al., 2015; Emmendoerfer et al., 2018). Figure 4.7 shows snapshots of the optimization history for a different initial configuration with holes. As can be seen from these results the consistent solution shown in Fig. 4.6(b) is obtained with and without holes in the initial design. The solutions from the literature are shown in Fig. 4.8 for comparison. This solution is also what expected intuitively since spherical shapes are theoretically the ideal structures for pressure vessels. The convergence and volume fraction plots are given in Fig.4.6(c). The figure indicates a smooth convergence. The behavior of the compliance curve depends on the initial solution of the problem. The final topology and convergence history shown in Fig. 4.6(b) and (c) respectively, resulted from an initial configuration without any holes, i.e., infeasible solution. In these cases, the convergence is expected to move up because as volume is reduced the structure becomes more compliant (less stiff). Similar behavior can be seen in the literature where authors have solved this example with different methods (see for example Picelli et al. (2015a)). Fig. 4.6(d) illustrates the particle distribution in the solid (blue particles) and void (red particles) regions, with the pressure interface indicated by the black particles.



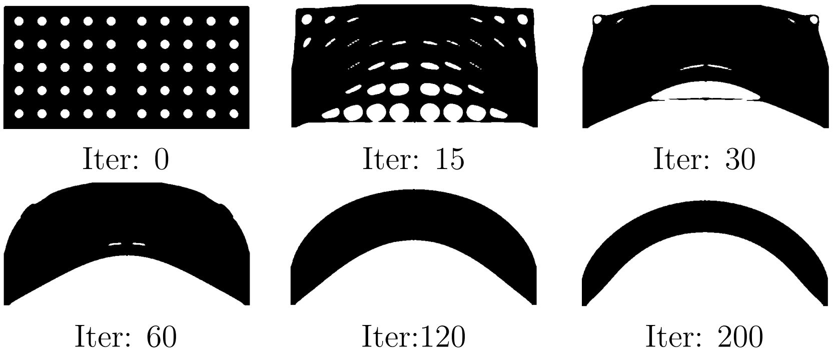
(a)

(b)

(c)

(d)

Figure 4.6: Arch structure: (a) problem definition (initial domain with no holes), (b) optimum solution, (c) convergence history and (d) particle distribution: void particles in red, solid particles in blue, pressure boundary particles in black



Iter: 0

Iter: 15

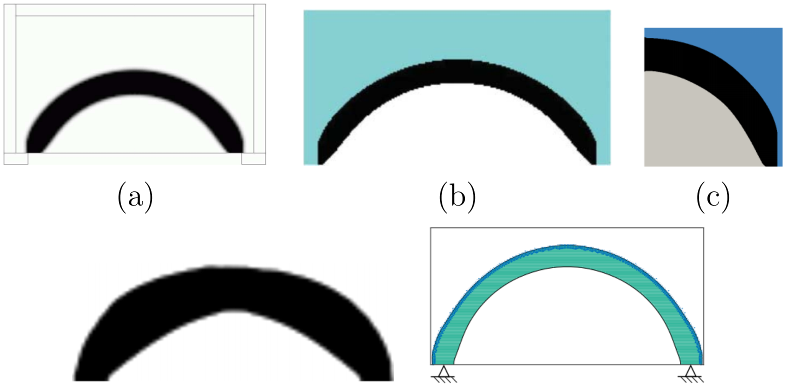
Iter: 30

Iter: 60

Iter:120

Iter: 200

Figure 4.7: Snapshots of optimization history in the arch example.



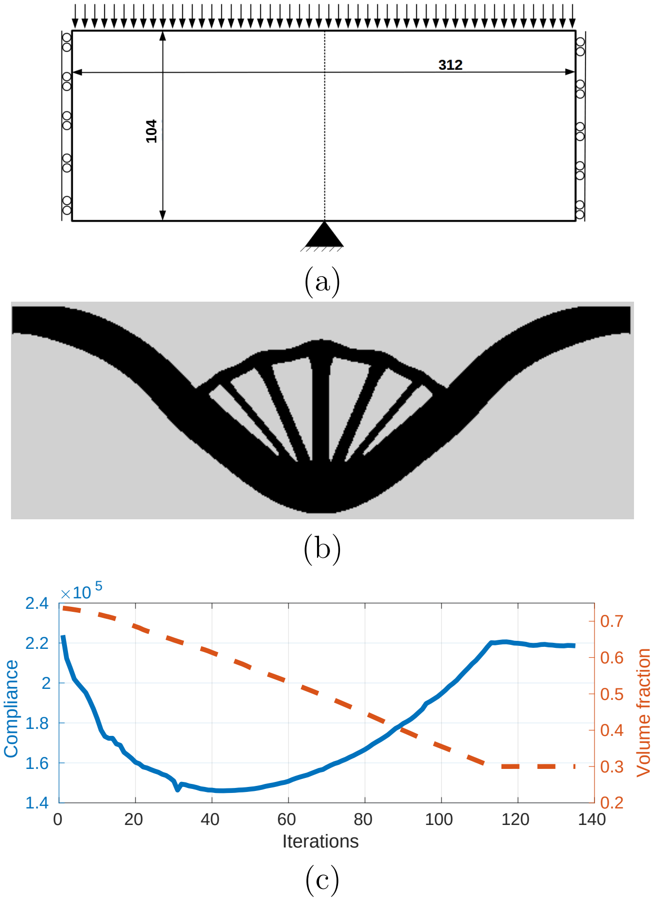
(a) (b) (c)

(d) (e)

Figure 4.8: Arch solutions from the literature: (a) Sigmund and Clausen (2007)(SIMP), (b) Picelli et al. (2015a)(BESO), (c) Picelli et al. (2019)(LSM), (d)Xia et al. (2015)(LSM) and (e) Emmendoerfer et al. (2018)(LSM)

4.2.4 Piston head model

Another commonly solved example in the literature is the piston head shown in Fig. 4.9(a). Authors have considered this example using SIMP by Sigmund and Clausen (2007) and Lee and Martins (2012), BESO by Picelli et al. (2015a) and the level set method by Xia et al. (2015), Emmendoerfer et al. (2018) and Picelli et al. (2019). The roller boundary condition is applied on the sides and the center point is fixed. Due to symmetry only the right half of the model is solved here using 156×104 background cells. A volume constraint of 30% is applied. The mirrored optimum topology is shown in Fig. 4.9(b), and Fig. 4.9(c) shows the convergence history and volume fraction plots. Figure 4.10 illustrates the iterative process for a particular initial hole distribution. Figure 4.11 shows solutions obtained for the piston head example by other authors using different methodologies. Since different methodologies were used for each of these solutions with different specifications for the analysis such as different mesh sizes, the numbers are not directly comparable. We thus compare the solutions qualitively based on the features they have in common. As can be seen, the arch-like curves near the lateral left and right walls and the elongated triangular holes in the center section are features similar to those obtained by other authors who previously solved this example. The solution obtained by RKPM in Fig. 4.9(b) is similar to the one obtained by Picelli et al. (2019) in which the level set method with equivalent nodal loads is used, and Picelli et al. (2015a) using the BESO method. The main difference between the different approaches is in the number of internal structural members appearing. There is also a noticeable difference compared with the other level set methods by Emmendoerfer et al. (2018) and Xia et al. (2015). Although the overall looks are similar the shapes of the holes are different in these two works and also the position of the top structural members is higher in Emmendoerfer et al. (2018).

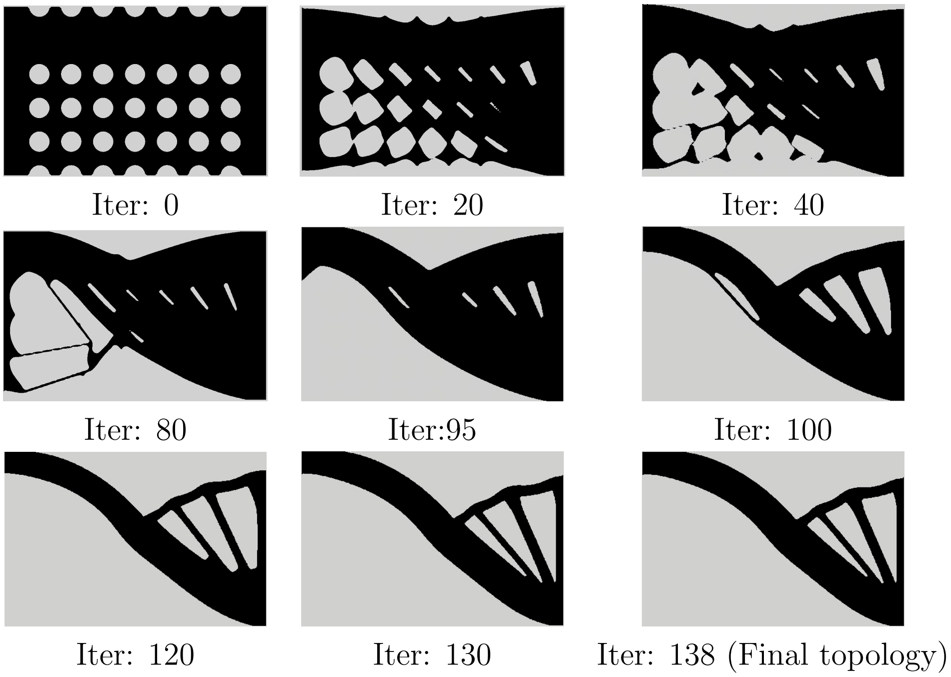


(a)

(b)

(c)

Figure 4.9: (a) Piston-head structure problem definition, (b) optimum solution, (c) convergence history


Iter: 0

Iter: 20

Iter: 40

Iter: 80

Iter:95

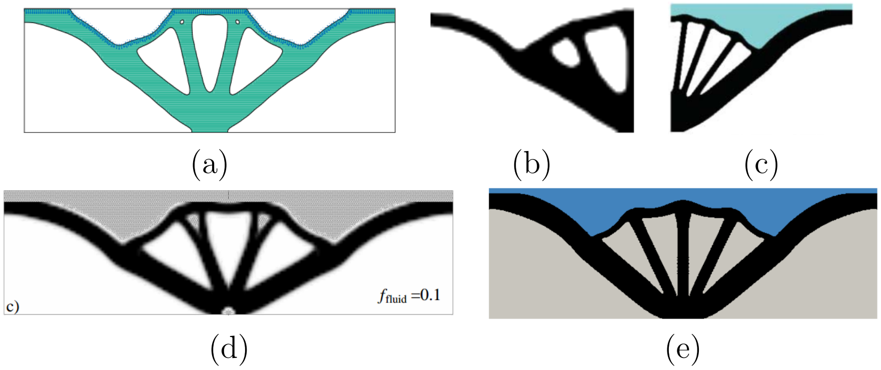
Iter: 100

Iter: 120

Iter: 130

Iter: 138 (Final topology)

Figure 4.10: Snapshots of optimization history in the piston example.


(a)

(b)

(c)

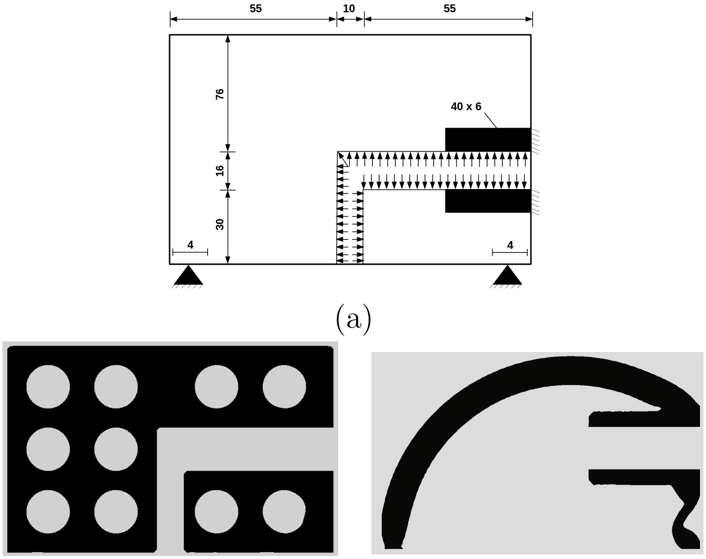
(d)

(e)

Figure 4.11: Piston solutions from the literature: (a) Emmendoerfer et al. (2018)(LSM),(b) Xia et al. (2015)(LSM), (c) Picelli et al. (2015a)(BESO), (d) Sigmund and Clausen (2007)(SIMP) and (e) Picelli et al. (2019)(LSM)

4.2.5 Pressure chamber model

The pressurized chamber example was first proposed by Hammer and Olhoff (2000) and Chen and Kikuchi (2001). The only subsequent works that considered this problem were Zhang et al. (2008) using the SIMP method and Picelli et al. (2015a) using the BESO method. The problem definition as shown in Fig. 4.12(a) is solved with 120×76 rectangular background cells. The two 40×60 flanges shown in black in Fig. 4.12 (a) are considered fixed and are excluded from the design domain. The applied pressure $p = 1$ is indicated by the arrows. Starting from an initial design with no holes the final solution in Fig. 4.12(c) is obtained. The same solution is also obtained from a design with a different initial configuration such as the one shown in Fig. 4.12 (b), which shows the independency of the results on the initial design. The optimized structure agrees well with the ones obtained in the previous literature as shown in Fig. 4.13, especially the one obtained by Picelli et al. (2015a). An interesting point here is that in this work well defined structural boundaries can be obtained with a relatively small grid. For example, the similar solution to the present work obtained by Picelli et al. (2015a) in Fig. 4.13(d) used 57000 finite elements whereas here we only use 9120 level set elements. The SIMP results obtained by Chen and Kikuchi (2001) and Hammer and Olhoff (2000) show differences with the RKPM solution in Fig. 4.12(c). The top structural member appears to be thicker with solid covering all the area on top of horizontal fluid region. The support at the right-hand side corner is also thinner in these examples. These differences are possibly due to the slightly different shape of the pressure region used in these examples as shown in 4.13(b). Similar differences can also be seen compared with the optimum structure by Picelli et al. (2019). In this work the level set method was used with Ersatz material approximation and the loads were transformed into work equivalent nodal loads.



(a)

(b)

(c)

Figure 4.12: Pressure chamber: (a) Problem definition, (b)initial design with holes and (c) optimum solution

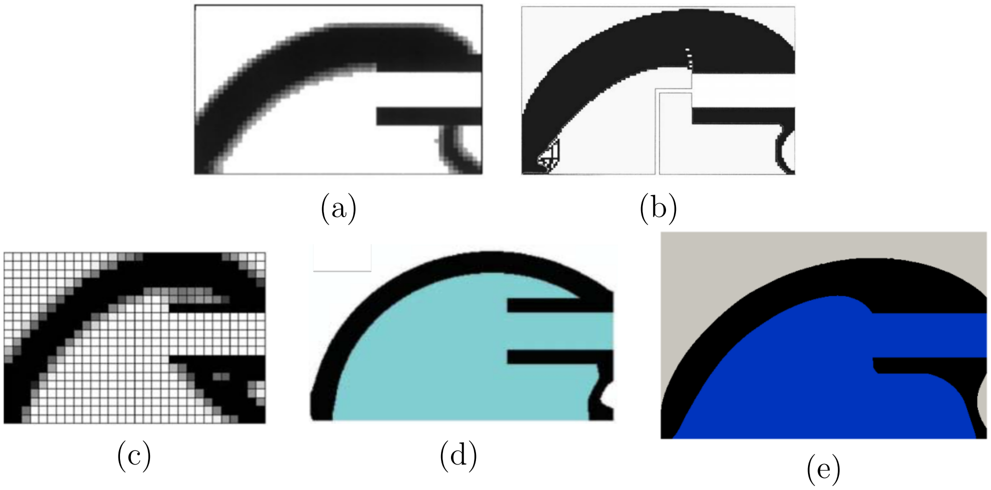
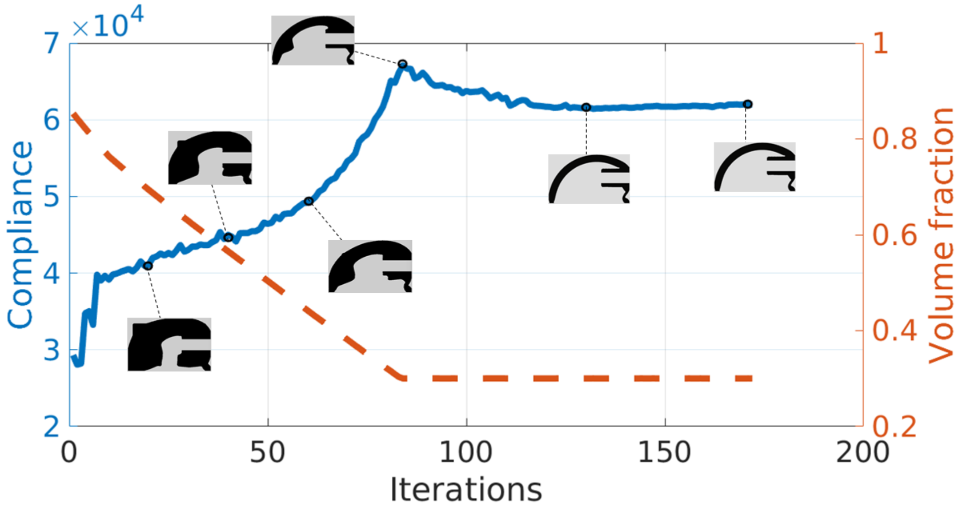
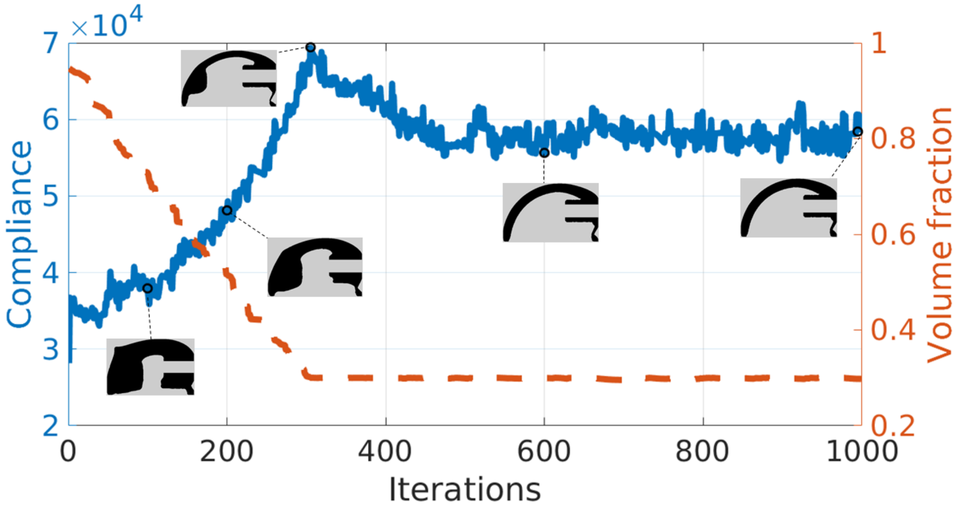


Figure 4.13: Chamber solutions from the literature: (a) Chen and Kikuchi (2001)(SIMP), (b) Hammer and Olhoff (2000)(SIMP), (c) Zhang et al. (2008)(SIMP), (d) Picelli et al. (2015a)(BESO) and (e) Picelli et al. (2015a)(LSM)

The geometry and changing loading direction make this example more challenging as compared to the arch and piston examples. Based on the literature one of the most common ways to deal with design dependent loads using a fixed grid is to transform them into equivalent nodal loads (for example in Hammer and Olhoff (2000), Du and Olhoff (2004) and Lee and Martins (2012) using SIMP, and Emmendoerfer et al. (2018) and Picelli et al. (2019) using the level set method). Thus, we compare the performance of the RKPM approach with an equivalent finite element method with work equivalent nodal loads for this example. The implementation for the specific fixed grid approach we used here can be found in Neofytou et al. (2019). The same level set algorithm is used for the two approaches whereas for the analysis we simply replaced RKPM with the fixed grid FEA for a fair comparison. Also, no regularization has been used in either approach. The convergence history and volume fraction plots for the two methodologies are shown in Fig. 4.14. As can be seen, although the final topologies are very similar, RKPM has a much smoother and more stable behavior compared to the FEA approach. This results in a much faster convergence, with RKPM converging at 171 iterations whereas the fixed grid FEA struggles to converge even after 1000 iterations have passed. It is also worth noting that the solution of this particular RKPM implementation is about 20 times slower per iteration compared to FEA. This is because of the large number of integration points required by the background cells. This is one of the main drawbacks of the implementation described in this chapter. Later in Ch. 5, this issue is addressed by employing a much more efficient nodal integration technique which significantly speeds up the computation.



(a)



(b)

Figure 4.14: Convergence history comparison between: (a) RKPM and (b) fixed grid FEA with work equivalent nodal loads for the chamber example.

4.3 Variable pressure field

To illustrate the applicability of the methodology to variable pressure fields, in this section the steady pressure load is replaced by a pressure field governed by the Laplace equation as

$$\nabla^2 P_f = 0 \quad \text{in} \quad \Omega_p \tag{4.8}$$

where P_f is the pressure and ∇^2 is the Laplacian operator. Boundary conditions include an imposed pressure P_0 on a portion Γ_p of the pressure boundary

$$P_f = P_0 \quad \text{on} \quad \Gamma_p \tag{4.9}$$

and the hard wall condition is given by

$$\mathbf{n} \cdot \nabla P_f = 0 \quad \text{on} \quad \Gamma_w \tag{4.10}$$

Here $\mathbf{n}$ is the outward unit normal vector to the fluid. For the imposition of boundary conditions in the pressure field the Lagrange multiplier method described in Ch. 3. The Galerkin discretization of Eq. 4.8 - Eq. 4.10 then yields

$$\begin{bmatrix} \mathbf{K}_p & \mathbf{G}_p^T \\ \mathbf{G}_p & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{P} \\ \boldsymbol{\lambda}_p \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{q}_p \end{bmatrix} \tag{4.11}$$

where $\mathbf{P}$ is the pressure vector to be computed and $\mathbf{q}_p$ is the equivalent load to enforce the Dirichlet boundary from Eq. 4.9. The global stiffness matrix $\mathbf{K}_p$ is computed as

$$\mathbf{K}_p = \int_{\Omega_p} (\nabla \boldsymbol{\Psi}_p^T) \nabla \boldsymbol{\Psi}_p^T d\Omega_p \tag{4.12}$$

where $\boldsymbol{\Psi}_p$ is the vector with RK shape functions of the fluid particles. Matrix $\mathbf{G}_p$ and vector $\mathbf{q}_p$ are used to enforce pressure boundary conditions using the Lagrange multiplier method and they are defined as

$$\mathbf{G}_p = -\int_{\Gamma_p} \boldsymbol{\Psi}_p^T \mathbf{N} d\Gamma \tag{4.13}$$

$$\mathbf{q}_p = -\int_{\Gamma_p} \mathbf{N}^T P_0 d\Gamma \tag{4.14}$$

$$\mathbf{N} = \begin{bmatrix} N_I & 0 \\ 0 & N_I \end{bmatrix} \tag{4.15}$$

where N_I is the standard Lagrangian interpolant along the Dirichlet boundary. The solution of Eq. 4.11 provides the pressure field acting on the structural loading surface. The discretized equation for the solid domain then becomes

$$\begin{bmatrix} \mathbf{K}_s & \mathbf{G}_s^T \\ \mathbf{G}_s & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \boldsymbol{\lambda}_s \end{bmatrix} = \begin{bmatrix} \mathbf{f}_s \\ \mathbf{q}_s \end{bmatrix} \tag{4.16}$$

where the subscript s denotes solid quantities and the load $\mathbf{f}_s$ is given by the pressure field obtained from Eq. 4.11 as

$$\mathbf{f}_s = \int_{\Gamma_h} \boldsymbol{\Psi}_s^T \mathbf{P} d\Gamma \tag{4.17}$$

The domain integration for the pressure field is performed similarly to the procedure described in Sec. 4.1. The pressure stiffness at matrix $\mathbf{K}_p$ at each Gauss point is multiplied by an artificial Young's modulus E_p which for "pressure" type particles with $\phi \leq 0$ takes the value 1 and for "void" type particles with $\phi < 0$ or "solid" particles with $\phi > 0$ takes the value 10^{-4}. This is expressed as

$$\text{E at node I} : \begin{cases} \mathrm{E}_p(\mathbf{x}_\mathrm{I}) = 1 & \text{if} \quad \phi \leq 0 \text{ and "pressure}'' \text{ type} \\ \\ \mathrm{E}_p(\mathbf{x}_\mathrm{I}) = 10^{-4} \times \mathrm{E} & \text{if} \quad \phi < 0 \text{ and "void}'' \text{ type or } \phi > 0 \end{cases} \tag{4.18}$$

Then at each Gauss point, E_p^{GP} is computed as

$$\mathrm{E}_p^{GP}(\mathbf{x}_{\mathrm{gp}}) = \sum_{\mathrm{I} \in \mathrm{G}_{\mathrm{gp}}} \Psi_\mathrm{I}(\mathbf{x}_{\mathrm{gp}}) \mathrm{E}_{p\mathrm{I}} \tag{4.19}$$

where np is the number of particles within the Gauss point's support domain, and Ψ_I and $\mathrm{E}_p\mathrm{I}$ are the RK shape function and Young's modulus associated with the I^{th} node in the Gauss point's support domain, respectively.

4.4 Pressure chamber subjected to variable pressure field

The pressurized chamber example described in Sec. 4.2.5 is solved here using the variable pressure field. In the current case the pressure load is computed by solving the Laplace equation for the pressure region highlighted in blue in Fig. 4.15. An inlet pressure P_{in} is imposed on the right edge of the pressure domain and an outlet pressure P_{out} is imposed on the bottom edge of the pressure domain. The objective is again compliance minimization under the same volume constraint, however, now the pressure field is provided by Eq. 4.8. The solution of Eq. 4.11 provides the surface pressure to be applied on the structure as shown in Fig. 4.16.

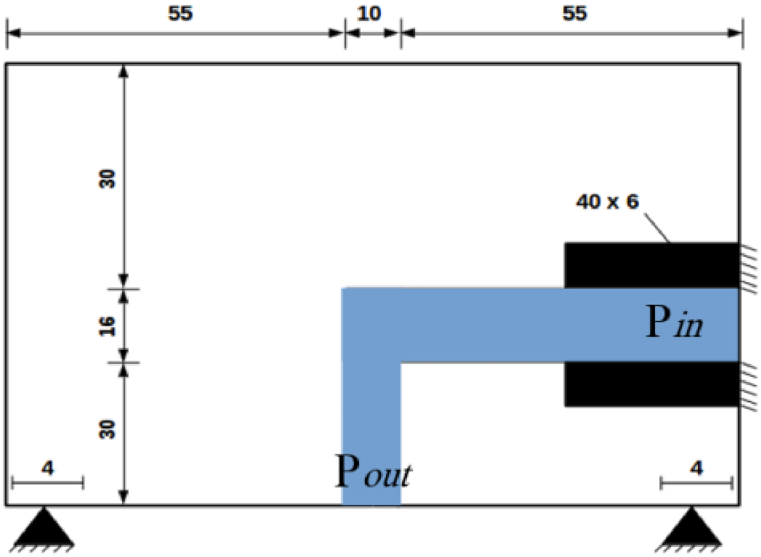
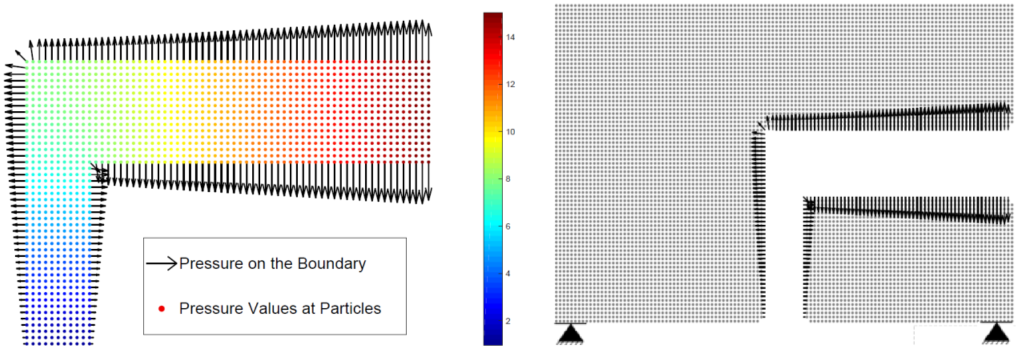


Figure 4.15: Schematic for pressure chamber example under variable pressure field. Same setup as Sec.4.2.5 only now the pressure load is computed by solving the Laplace equation for the pressure region highlighted in blue.



(a) (b)

Figure 4.16: Surface load obtained from solution of Eq. 4.11: (a) Pressure region with linear pressure field solution and (b) Resulting surface load on the structure.

The optimum solutions for different combinations of inlet (P_{in}) and outlet (P_{out}) pressure are presented in Fig. 4.17. As expected when $P_{in} = P_{out}$ the same solution as in Sec. 4.2.5 is obtained as shown in Fig. 4.17(a). In Fig. 4.17 (b), the pressure difference increases which results in the appearance of thin structural supports to support the higher pressure gradient. As the pressure difference increases even more in Fig. 4.17 (c) additional members appear. In Fig. 4.17 (d) the pressure difference at the inlet and outlet is about the same as in Fig. 4.17 (b) despite the absolute values at the inlet and outlet being different. This indicates that the pressure difference is what drives the design rather than the absolute pressure values. Figure 4.18 illustrates snapshots of the particle distributions at different iterations for the case $P_{in} = 15.0\ \text{Pa}, P_{out} = 1.0\ \text{Pa}$.

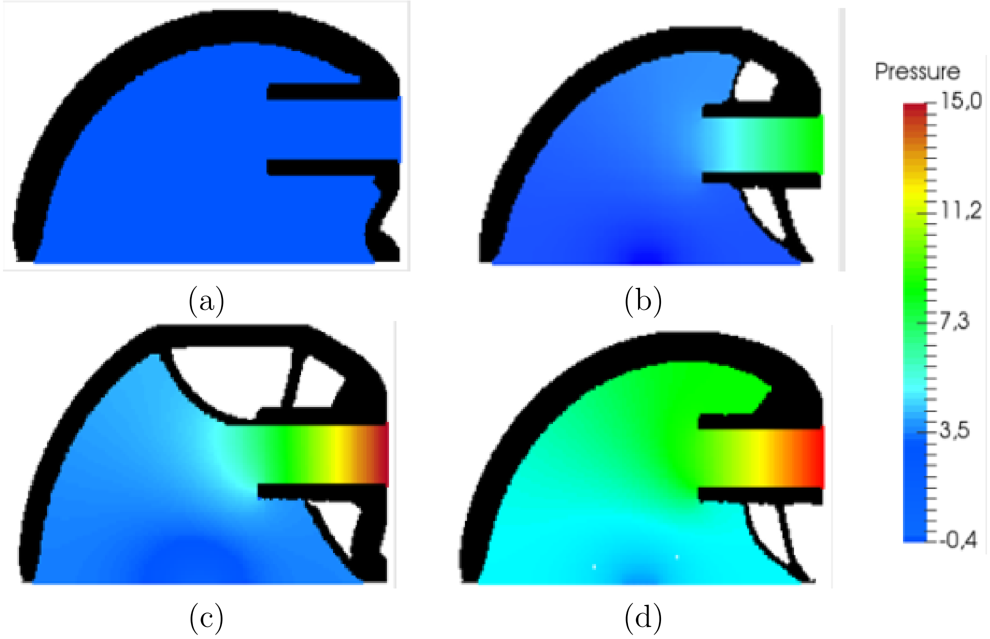
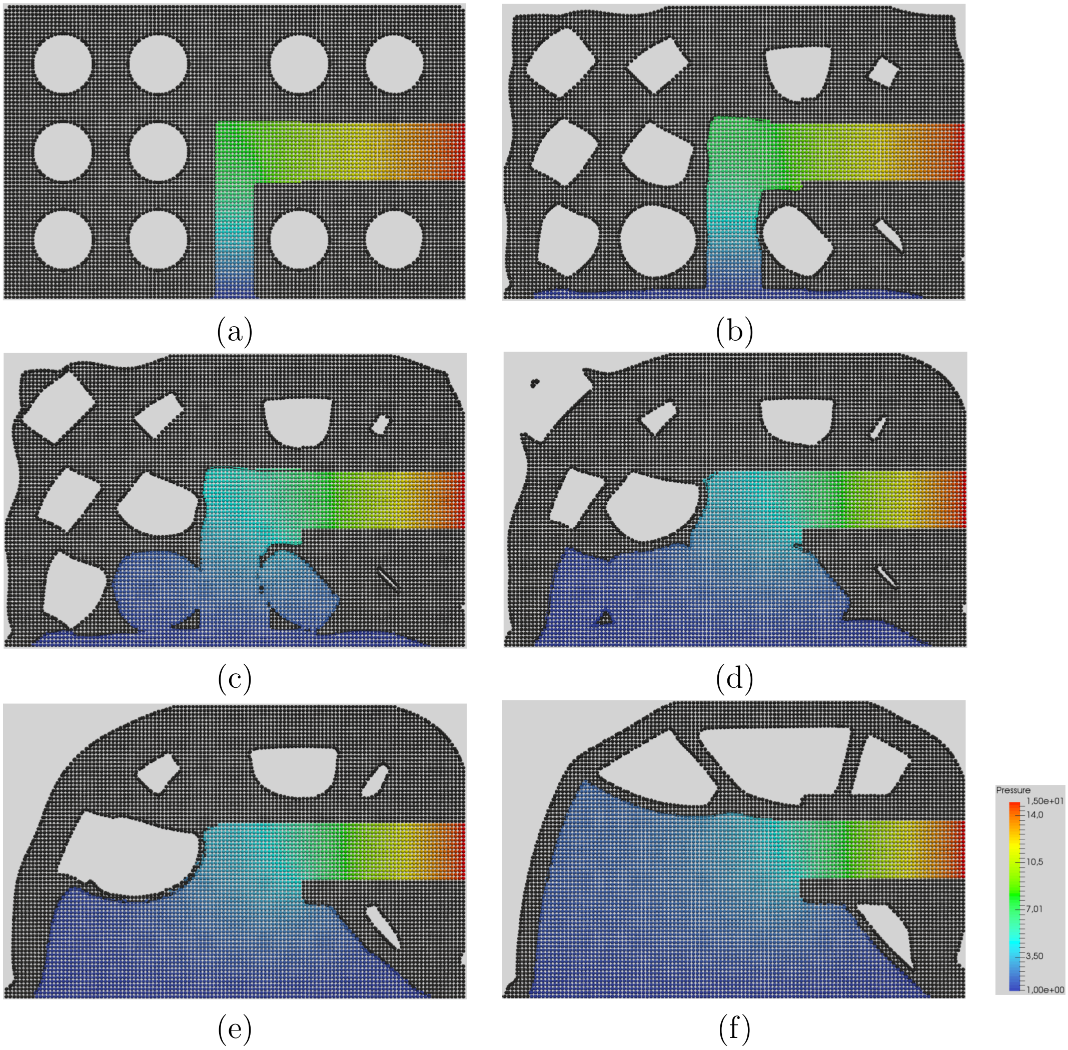


Figure 4.17: Optimum solution for pressure chamber for different pressure values at the inlet and outlet: (a) $P_{in} = 1.0\ \text{Pa}, P_{out} = 1.0\ \text{Pa}$, (b) $P_{in} = 10.0\ \text{Pa}, P_{out} = 1.0\ \text{Pa}$, (c) $P_{in} = 15.0\ \text{Pa}, P_{out} = 1.0\ \text{Pa}$ and (d) $P_{in} = 15.0\ \text{Pa}, P_{out} = 4.0\ \text{Pa}$



(a) (b) (c) (d) (e) (f)

Figure 4.18: Snapsots of the particle distribution for the pressurized chamber example for $P_{in} = 15P_{out}$ after 200 iterations

4.5 Conclusions

This chapter presented a level set topology optimization method for compliance minimization of structures subjected to design-dependent pressure loads. To address the challenge of identifying the surface for the loads to act on, the reproducing kernel particle method was used along with the LSTO. The LSTO provides the clear boundary representation and the RKPM offers the freedom to place particles on the new boundaries to apply the pressure loads. As shown through the numerical examples, the proposed methodology gives results that are in good agreement with the results obtained by different methods in the literature. The meshfree method can straightforwardly handle the design-dependent loading by directly applying the pressure loads on the relevant structural boundaries without the need of any special load treatments or remeshing. This demonstrates an advantage of this methodology. The explicitly defined boundaries can be advantageous when using separate governing equations, as will be further discussed in Ch. 6 for fluid-structure interactions. In addition, the RKPM offers higher order approximation of the equations. The RK "smooth approximation" allows for Young's modulus interpolation in this chapter to yield a smooth Young's modulus distribution for better conditioned boundary sensitivities. This leads to a faster convergence rate of the optimization procedure. Before tackling more complex problems such as fluid-structure interactions, herein we show that our method addresses correctly the hydrostatic problem, which is the base for any design-dependent loading problem.

Chapter 5

Nodally Integrated RKPM for Stress-Based and Design-Dependent Problems

In the previous chapter, design-dependent problems with hydrostatic loads were solved with RKPM and Gauss quadrature. The method has been shown to perform well and effectively handle the design-dependency. However, the use of a background mesh with a large number of quadrature points results in low computational efficiency. Moreover, since the background mesh is fixed, an interpolation scheme had to be used for the integration.

In this chapter, we aim to solve the problem of efficiency by switching to a nodally integrated RKPM, where quantities are sampled at the particles themselves. As will be explained in this chapter, nodal integration methods have been exten-sively studied and improved since the early days to provide both efficiency and accuracy. As will be shown in the Chapter, certain types of nodal integration such as the stabilized conforming nodal integration (SCNI) and the naturally stabilized nodal integration (NSNI), not only maintain the efficiency of direct nodal integration but even present higher accuracy than Gauss integration. This is due to the fact that they achieve linear exactness by satisfying the so-called integration constraints whereas Gauss integration does not.

5.1 Stabilized conforming nodal integration

Galerkin meshfree methods such as RKPM are based on the weak form of PDEs. Thus, domain integration is required to evaluate the integrals in the weak form as explained in Ch. 4. This can be done either by performing Gaussian integration with background cells as the quadrature domains as implemented in Ch. 4, or by using nodal integration which employs nodal representative domains. As has been shown by Dolbow and Belytschko (1999), when Gauss quadrature is used for the domain integration significant errors may arise when the background mesh does not coincide with the support domains. This becomes particularly difficult when circular supports are being used. This can be improved by using sufficiently high quadrature rules. However, this results in a high computational cost since a large number of computational points has to be used.

Nodal integration on the other hand is very efficient since it does not require as many computational points as Gaussian integration. However, direct nodal integration where shape functions and their derivatives are evaluated directly at the nodes is known to be prone to non-convergent numerical solutions and instability due to rank deficiency. Evaluating shape function gradients directly at the nodes yields instability in the solution due to severe underestimation of the strain energy of short-wavelength modes (Beissel and Belytschko, 1996).

To achieve accuracy, stability and optimal convergence properties while maintaining efficiency, Chen et al. (2001) proposed the stabilized conforming nodal integration (SCNI) method. In this method, shape function gradients are smoothed over conforming nodal representative domains to perform the integration at each particle. SCNI ensures accuracy by satisfying the so-called integration constraints as necessary conditions for achieving linear exactness, *i.e.* passing the linear patch test. In fact, Gaussian integration for the meshfree weak form does not satisfy the integration constraints, therefore the first order accuracy is not guaranteed even if the approximation of test and trial functions is linearly complete. The generalization of SCNI for arbitrary order variational consistency is presented in Chen et al. (2013). Moreover, strain smoothing avoids evaluating derivatives of meshfree shape functions at nodes and thus eliminates the rank instability.

The smoothed gradient $\tilde{\Psi}_{I,i}$ of the shape functions at each nodal point is computed using the divergence theorem as follows

$$\tilde{\Psi}_{I,i}(\mathbf{x}_N) = \frac{1}{A_N}\int_{\Omega_N} \Psi_{I,i}(\mathbf{x})\mathrm{d}\Omega = \frac{1}{A_N}\int_{\Gamma_N} \Psi_I(\mathbf{x})n_i\mathrm{d}\Gamma \tag{5.1}$$

where A_N is the area of the nodal representative domain (i.e voronoi cell) Ω_N associated with node N, Γ_N is the boundary of the nodal representative domain, and n_i denotes the i-th component of the outward unit normal vector to the representative domain. For the evaluation of $\tilde{\Psi}_{I,i}$ at the nodal points using Eq. 5.1, boundary integration of the nodal representative domain is needed. This is done in this work using a one-point Gauss integration rule

$$\tilde{\Psi}_{I,i}(\mathbf{x}_N) \approx \frac{1}{A_N}\sum_{k=1}^{S_N} \Psi_I(\tilde{\mathbf{x}}_N^k)n_i(\tilde{\mathbf{x}}_N^k)L_k \tag{5.2}$$

where S_N is the number of midpoints for each boundary segment of the Voronoi cell associated with node $\mathbf{x}_N$ and L_k is the length of the k_{th} segment. The nodal representative domains can be constructed by Delaunay triangulation or Voronoi diagram as shown in Fig. 5.1.

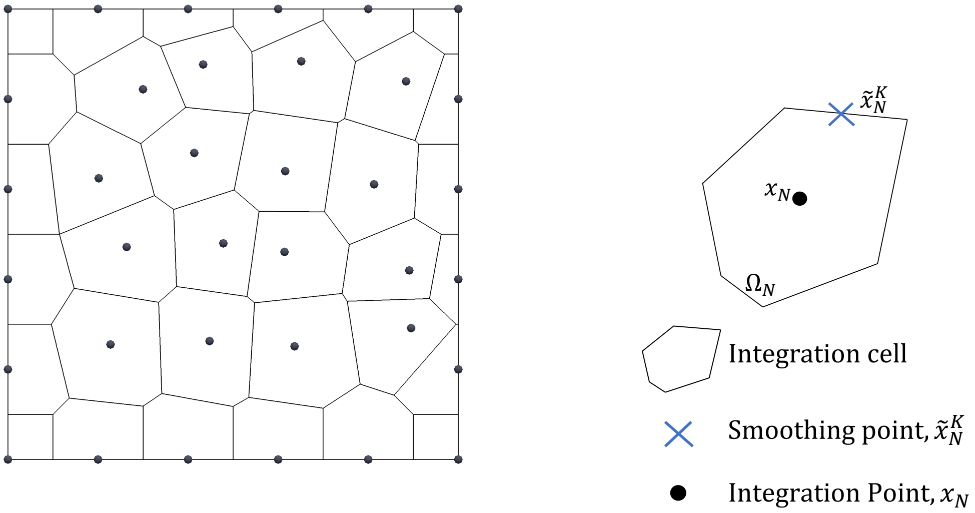


Figure 5.1: Voronoi diagram and SCNI integration scheme

Finally the smoothed gradient matrix $\tilde{\mathbf{B}}_I$ for the two-dimensional case can be expressed as

$$\tilde{\mathbf{B}}_I = \begin{bmatrix} \tilde{\Psi}_{I,x}(\mathbf{x}_N) & 0 \\ 0 & \tilde{\Psi}_{I,y}(\mathbf{x}_N) \\ \tilde{\Psi}_{I,y}(\mathbf{x}_N) & \tilde{\Psi}_{I,x}(\mathbf{x}_N) \end{bmatrix} \tag{5.3}$$

The smoothed gradient matrix $\tilde{\mathbf{B}}$ can now replace matrix $\mathbf{B}$ in Eqs. 3.25 and 3.26. It is important to emphasize that although the strain smoothing involves evaluation of shape functions at the edges of the representative nodal domain, it is only needed for shape functions. The stiffness is integrated entirely at the particles. Also, since the smoothed gradient matrices are consistent with the weights in the nodal integration, the distribution of weights, *i.e* the areas of the cells, have marginal effects on the solution accuracy.

Naturally stabilized nodal integration

SCNI avoids rank deficiency in direct nodal integration, however spurious low-energy modes may still appear in the solution as illustrated in Hillman and Chen (2016). Therefore, an additional stabiliza-tion technique is needed to eliminate these low-energy modes. In this work we employ the naturally stabilized nodal integration (NSNI) technique proposed in Hillman and Chen (2016), that overcomes the instabilities by introducing an implicit gradient expansion of the strain field, equivalent to the first order Taylor expansion as

$$\boldsymbol{\varepsilon}\left(\boldsymbol{u}^h(\mathbf{x})\right) \approx \boldsymbol{\varepsilon}\left(\boldsymbol{u}^h(\mathbf{x}_N)\right) + \sum_{i=1}^{d}(x_i - x_{Ii})\boldsymbol{\varepsilon}\left(\hat{\mathbf{u}}_i^h(\mathbf{x}_N)\right) \tag{5.4}$$

where for a 2-dimensional problem, i = x, y and the implicit gradient of the displacement, $\hat{\mathbf{u}}_i^h(\mathbf{x}_N)$, is given by

$$\hat{\mathbf{u}}_i^h(\mathbf{x}_N) = \sum_{i=1}^{NP} \Psi_{Ii}^{\nabla} \mathbf{u}_I \tag{5.5}$$

and Ψ_{Ii}^{∇} is the implicit gradient of the shape function (Chen et al., 2004)

$$\Psi_{Ii}^{\nabla} = \mathbf{H}_i^T \mathbf{M}^{-1}(\mathbf{x}) \mathbf{H}(\mathbf{x} - \mathbf{x}_I) \Phi_a(\mathbf{x} - \mathbf{x}_I) \tag{5.6}$$

where for a linear basis the vector $\mathbf{H}_i$ takes the following values

$$\begin{aligned} \mathbf{H}_{\mathbf{x}} &= [0, -1, 0]^T \\ \mathbf{H}_{\mathbf{y}} &= [0, 0, -1]^T \end{aligned} \tag{5.7}$$

Putting Eq. 5.4 into the variational equations, the stiffness matrix is derived as

$$\mathbf{K}_{IJ}^c = \sum_{N=1}^{NP} \left(\underbrace{\tilde{\mathbf{B}}_I^T(\mathbf{x}_N) \mathbf{C} \tilde{\mathbf{B}}_J^T(\mathbf{x}_N) A_N}_{SCNI} + \underbrace{\mathbf{B}_{Ix}^{\nabla}(\mathbf{x}_N) \mathbf{C} \mathbf{B}_{Jx}^{\nabla} \mathbf{M}_x(\mathbf{x}_N) + \mathbf{B}_{Iy}^{\nabla}(\mathbf{x}_N) \mathbf{C} \mathbf{B}_{Jy}^{\nabla} \mathbf{M}_y(\mathbf{x}_N)}_{Stabilization} \right) \tag{5.8}$$

where $\mathbf{B}_{I1}^{\nabla}(\mathbf{x}_N)$ and $\mathbf{B}_{I2}^{\nabla}(\mathbf{x}_N)$ are defined as

$$\begin{aligned} \mathbf{B}_{Ix}^{\nabla}(\mathbf{x}_N) &= \begin{bmatrix} \Psi_{Ix,x}^{\nabla}(\mathbf{x}_N) & 0 \\ 0 & \Psi_{Ix,y}^{\nabla}(\mathbf{x}_N) \\ \Psi_{Ix,y}^{\nabla}(\mathbf{x}_N) & \Psi_{Ix,y}^{\nabla}(\mathbf{x}_N) \end{bmatrix} \\ \mathbf{B}_{Iy}^{\nabla}(\mathbf{x}_N) &= \begin{bmatrix} \Psi_{Iy,x}^{\nabla}(\mathbf{x}_N) & 0 \\ 0 & \Psi_{Iy,y}^{\nabla}(\mathbf{x}_N) \\ \Psi_{Iy,y}^{\nabla}(\mathbf{x}_N) & \Psi_{Iy,x}^{\nabla}(\mathbf{x}_N) \end{bmatrix} \end{aligned} \tag{5.9}$$

with $\Psi_{Ii,j}^{\nabla}(\mathbf{x}_N)$ obtained by direct differentiation of Ψ_{Ii}^{∇} in Eq. 5.6 with respect to x_j Hillman and Chen (2016).

The terms $M_1(\mathbf{x}_N)$ and $M_2(\mathbf{x}_N)$ are the second moments of inertia in each nodal integration domain given by

$$\begin{aligned} M_1(\mathbf{x}_N) &= \int_{\Omega_N} (x - x_N)^2 \mathrm{d}\Omega \\ M_2(\mathbf{x}_N) &= \int_{\Omega_N} (y - y_N)^2 \mathrm{d}\Omega \end{aligned} \tag{5.10}$$

Stabilization with Eq. 5.8 is termed naturally stabilized nodal integration (NSNI) since the constants $M_1(\mathbf{x}_N)$ and $M_2(\mathbf{x}_N)$ associated with the additional terms occur completely naturally, and thus no tuning of any parameters is required, which is in contrast to other stabilized methods (Beissel and Belytschko, 1996; Puso et al., 2008).

5.2 Level set method and nodally integrated RKPM

The nodally integrated RKPM is used in this work within the context of level set topology optimization to achieve an exact description of structures without remeshing. The level set boundary and signed distance values are used for the particle distribution and the construction of the Voronoi diagram of the nodally integrated RKPM method as explained in this section. In turn, the LSTO benefits from the useful properties of the RKPM method such as placing particles on the discretized boundary points to compute sensitivities, and the higher order shape functions to improve accuracy. This section describes several ingredients of the methodology which are discussed in more detail through the examples in Sec. 5.3.

5.2.1 Particle positions and boundary sensitivity

As explained in Sec. 3.1, in order to obtain the optimum velocities required for the update of the level set function and thus the topology, shape sensitivities need to be computed at the discretized boundary points i.e. the points at which the boundary intersects the elements of the level set mesh. The positioning of particles at these boundary points as shown in Fig. 5.2, allows for direct computation of sensitivities. Thus, at each optimization iteration, particles are always placed at the boundary points first. These boundary particles are stored at the top of the list of all particles to identify them from the particles in the interior of the structure. The particles in the interior of the structure can be freely positioned. For example in Fig. 5.3 (a) the particles are regularly positioned at the nodal positions of the level set mesh with the respective Voronoi diagram for the NSNI shown in Fig. 5.3. In Fig. 5.3 (c) the particles are randomly positioned and the resulting Voronoi diagram is shown in Fig. 5.3 (d). Based on the above discussion on boundary sensitivity, accuracy is more critical close to the boundary. In Sec. 5.3.1 a particle scheme where more particles are strategically placed within a region close to the boundary is discussed and illustrated through a stress minimization example.

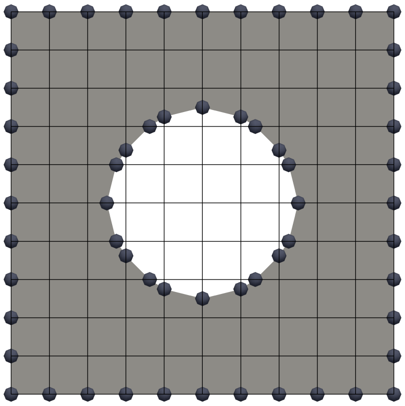


Figure 5.2: Particles are first placed at the discretized boundary points

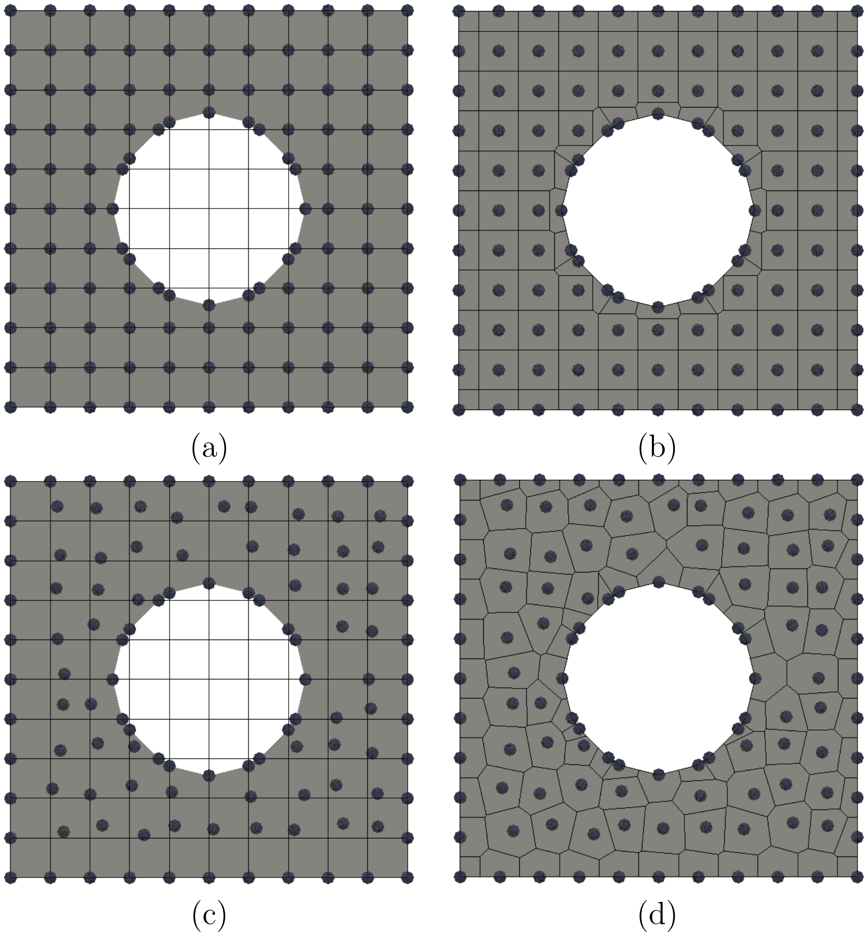


Figure 5.3: Different particle distributions: (a) Regularly placed particles, (b) Voronoi diagram for regularly placed particles, (c) irregularly placed particles and (d) Voronoi diagram for irregularly placed particles

5.2.2 Construction of Voronoi diagram

For the construction of the nodal representative domains used in the NSNI, the Voronoi diagram is employed here using the open source software library voro++ (Rycroft, 2009). For the construction of the diagram, the particle coordinates are given as an input which results in an unclipped Voronoi diagram. An example is shown in Fig. 5.4. Figure 5.4 (a) shows the unclipped diagram for a rectangular plate with a hole, with the cells along the boundary extending outside of the structure and get intersected by the boundary segments. In order to make the diagram conforming to the structure, which is employed for the NSNI method, the intersected cells need to be clipped as shown in Fig. 5.4 (b). For such clipping process, the signed distance values at the vertices of each intersected cell are employed as described in Fig. 5.5. The intersections of the level set boundary segments with the edges of the Voronoi cells are first computed. For the most part, the cells of boundary particles will be intersected by the boundary segments, however it can happen that interior cells are intersected as well. If an intersected cell is also a boundary cell, the intersections along with the particle to which the cell belongs to split the cell into two polygons as shown in Fig. 5.5 (a). When interior cells are intersected, only the intersection points split the cell into polygons as shown in Fig. 5.5 (b). The polygon with the biggest sum of the signed distance values at its vertices is kept as the new clipped cell, while the vertices of the remaining polygon are removed. This is illustrated in Figs. 5.5 (a) and (b) for boundary and interior cells, respectively. The remaining polygon after clipping, for example polygon A in Figs. 5.5(a) and (b), is then used as the integral domain to calculate the stiffness matrix. Since this clipping operation only has to be done for the cells along the boundary and maybe a few neighboring interior cells, the process is not computationally expensive.

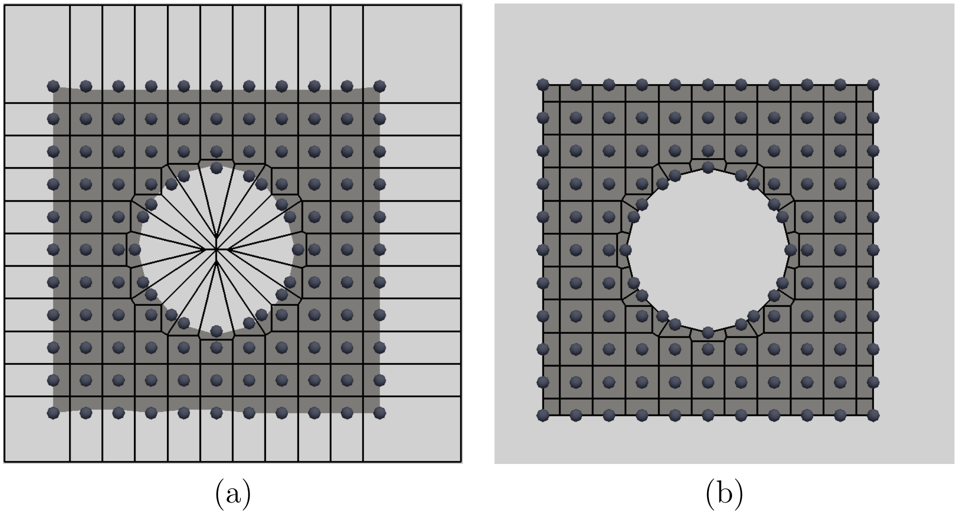
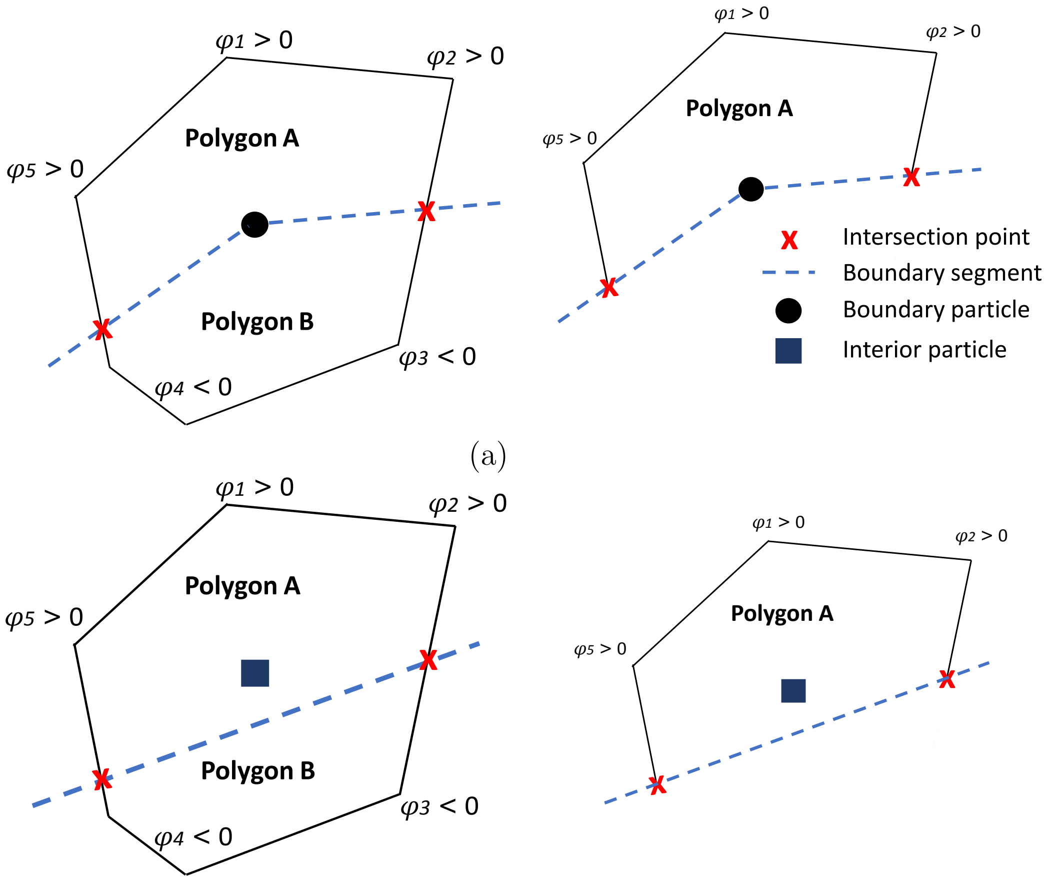


Figure 5.4: Construction of Voronoi diagram: (a) Unclipped diagram, (b) Clipped diagram



(a)

(b)

Figure 5.5: Clipping process for intersected cells for: (a) Boundary intersected cells and (b) interior intersected cells. First, an intersected cell is split into two polygons based on intersection points with the boundary segments. The next step is clipping the cell based on signed distance values at the vertices. The sum $\phi_1 + \phi_2 + \phi_5$ is larger than $\phi_3 + \phi_4$ and thus polygon B is removed

Size of support domain

The Voronoi diagram information can also be used to compute the support domain size for each particle in a way that it is consistent with the density of the particle distribution around it, as explained in Lu and Chen (2003). The neighbors of each particle are identified as those that share an edge with the particle's cell. For example in Fig. 5.6 particles 2-11 except particle 7 are neighbors of particle 1. The support size a_I of particle 1 is then defined as

$$a = \tilde{c} \cdot d_{max} \tag{5.11}$$

where d_{max} is the distance to the furthest neighbor as shown in Fig. 5.6 in which the distance between particles 1 and 11 is the maximum amongst all neighbors. The normalized support size $\tilde{c}$ is chosen as 2.0, which is twice the order of the linear basis used in this work. As can be seen in the figure, in areas with denser particle distribution support sizes are smaller than areas with less dense distribution.

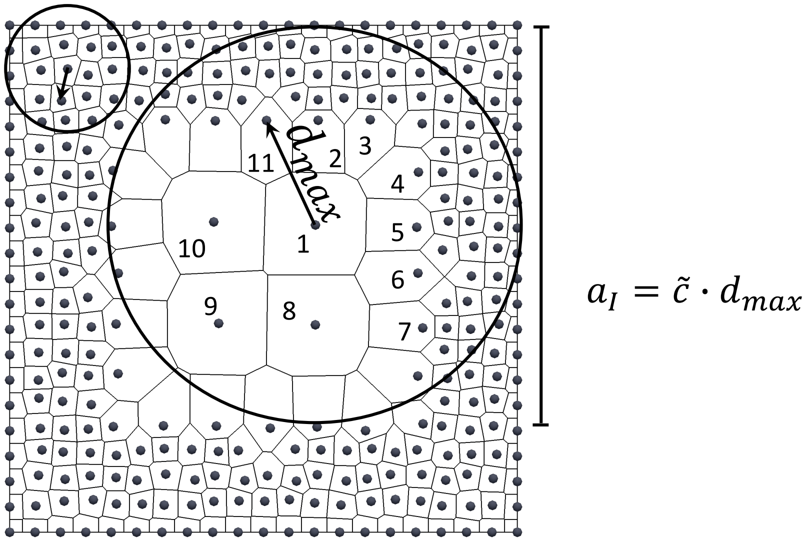
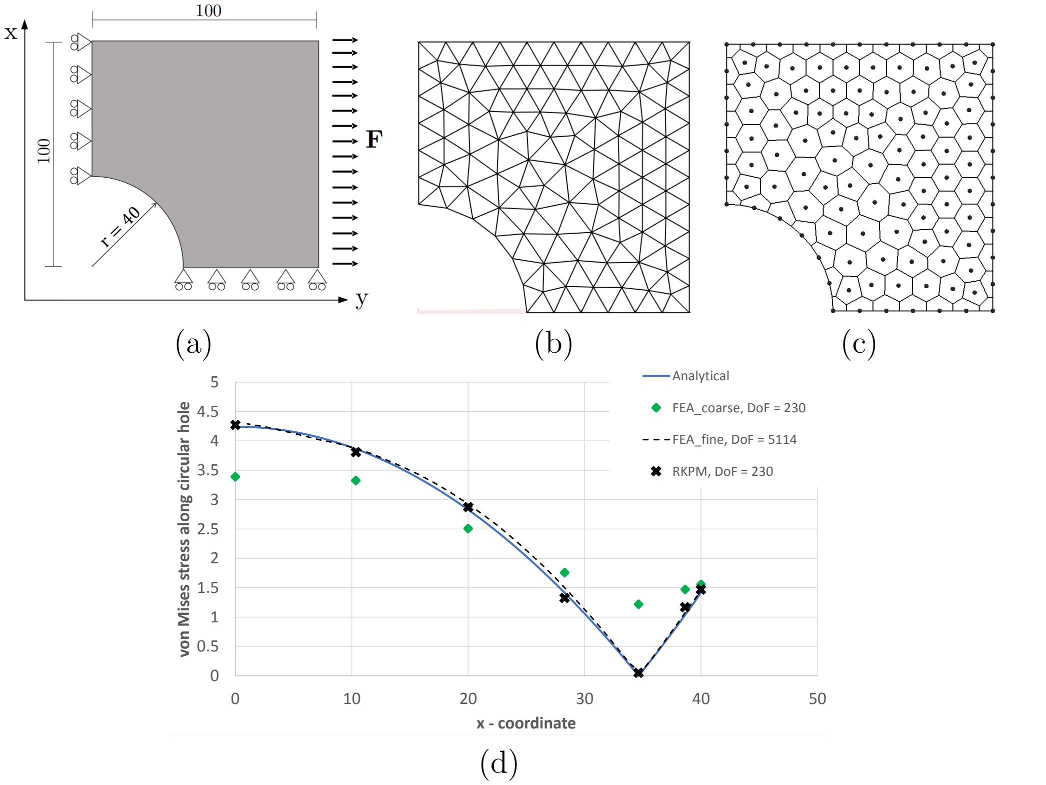


Figure 5.6: Support domain size definition based on Voronoi diagram information

5.3 Examples

In the literature of Galerkin meshfree methods, stress-based computations have been used to validate against the FEM in terms of accuracy and convergence properties (Belytschko et al., 1994; Lacroix and Bouillard, 2003; Tanojo and Pudjisuryadi, 2006) for linear elastic structures. As a simple illustration, the example of a linear elastic plate with a hole is presented here as shown in Fig. 5.7 (a). Specifically, the FEA is carried out via the partial differential equation toolbox in MATLAB, using linear, triangular elements. The FEA mesh is shown in Fig. 5.7 (b). For the RKPM analysis, the mesh generated in MATLAB is used to place the particles at the nodal positions of the elements in order to create a comparable discretization. The particles along with the Voronoi diagram are shown in Fig. 5.7 (c). Symmetry conditions are applied at the left and bottom edges. The plate is loaded uniaxially and analyzed under plane stress conditions. The values used for Young's modulus and Poisson's ratio are 1 and 0.3, respectively. The von Mises stress is computed by the two methods at the nodes on the circular boundary to compare the accuracy of each approach against the analytical solution Belytschko et al. (1994). As can be seen in Fig. 5.7 (d), the solution obtained by RKPM is in good agreement with the analytical solution for this specific discretization, which consists of 230 degrees of freedom (DOF). Of course, when a finer FEA mesh is used as shown in Fig. 5.7 (d) for 5114 DoF, FEA also provides an accurate solution as expected.



(a) (b) (c)

(d)

Figure 5.7: Plate with a hole example: (a) Problem setup, (b) FEA mesh, (c) RKPM particles and Voronoi diagram and (d) von Mises stress computation at the nodes along the circular hole by FEA and RKPM, and comparison with the analytical solution

In the context of topology optimization, stress-based problems are challenging and remain an active area of research (Picelli et al., 2018; Zhang et al., 2017; Sharma et al., 2018; Conlan-Smith and James, 2019). For the aforementioned reasons, stress-based problems are thus chosen here to illustrate the effectiveness of the proposed LSTO-RKPM methodology and to investigate several interesting features such as different particle distributions, kernel functions and efficiency. A design-dependent problem with hydrostatic pressure loads is also used to show how the crisp representation of the level set boundary in combination with particle placement at boundary points can straightforwardly handle the design dependency without interpolation for either the loads or the domain integration. For the examples Young's modulus value is set to 1, Poisson's ratio is 0.3, the dimensionless support size $\tilde{c}$ is 2.0 and the normalized penalty parameter β_{nor} for Nitsche's method is chosen as 1000. Convergence of the objective is checked over 5 consecutive iterations and the absolute tolerance is 0.001.

5.3.1 Stress-based examples

The p-norm function is used here which approximates the maximum stress in the structure (Picelli et al., 2018). This is a global stress measure and can be used either as an objective for stress minimization, or as an inequality constraint. The p-norm functional defined in the domain Ω can be written as follows

$$G(\Omega) = \boldsymbol{\sigma}_{PN} = \left(\int_{\Omega} \boldsymbol{\sigma}_{vm}^{p} d\Omega \right)^{\frac{1}{p}} \tag{5.12}$$

where p is the p-norm parameter, σ_{PN} denotes the p-norm stress and σ_{vm} is the von Mises stress.

For the solution of the stress-based problems the shape sensitivity, $G'(\Omega)$, of the p-norm functional $\boldsymbol{G}(\Omega)$ is used, the derivation of which can be found in Picelli et al. (2018),

$$G'(\Omega) = \frac{G(\Omega)^{1-p}}{p} \int_{\Gamma} (\boldsymbol{\sigma}_{vm}^{p} + C_{ijkl}\varepsilon_{ij}(\boldsymbol{u}) \cdot \varepsilon_{kl}(\boldsymbol{\lambda}) \tag{5.13}$$

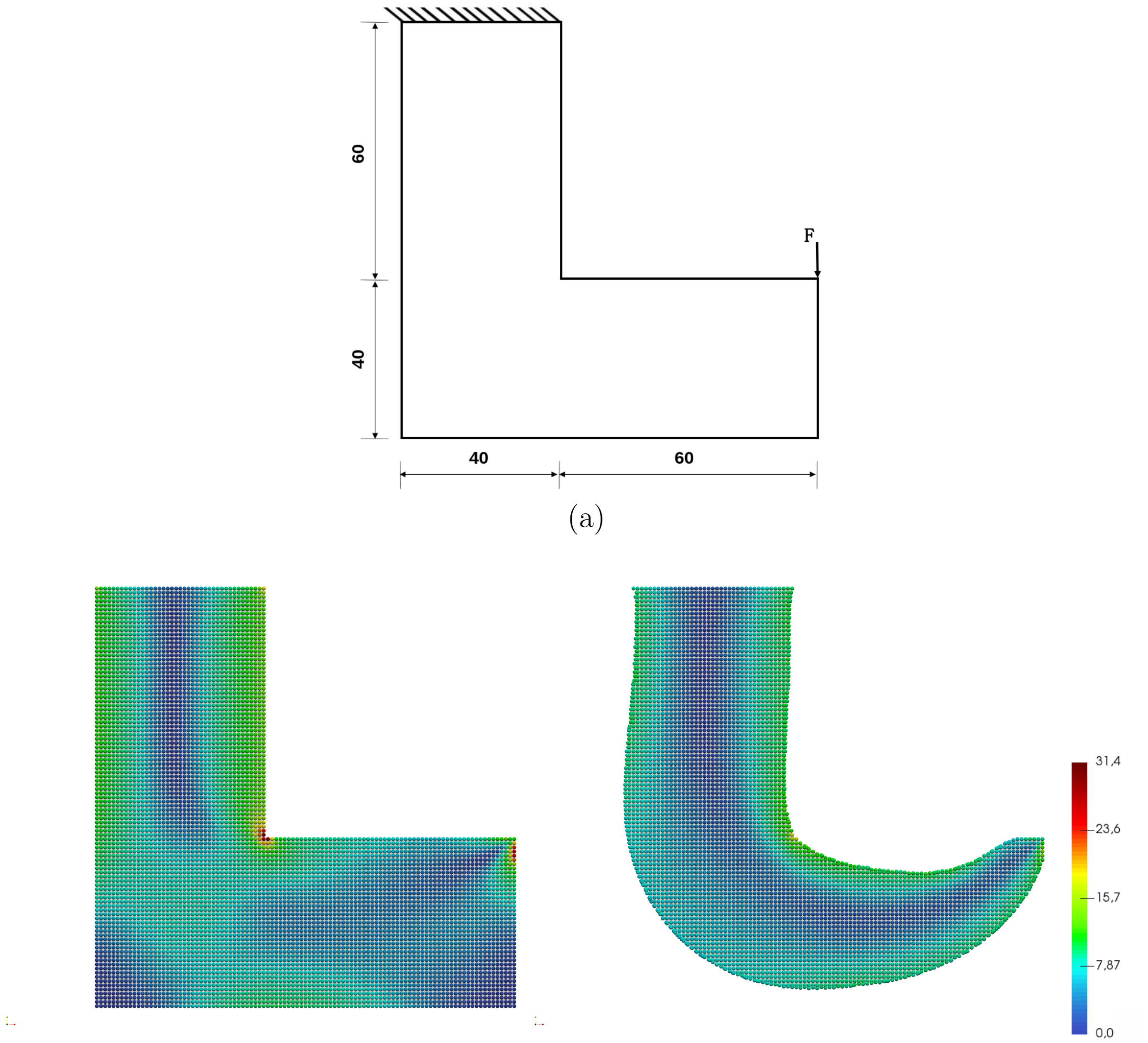
where C_{ijkl} is the elasticity tensor,
varepsilon represents the mechanical strain and $\boldsymbol{u}$ and $\boldsymbol{\lambda}$ denote the state and adjoint variable vectors respectively. Eq. 5.13 indicates how the p-norm stress functional changes when a structural boundary point moves in its normal direction. The shape sensitivity terms are computed directly at the boundary points where RKPM particles are placed.

Example 1: Stress Minimization

Stress minimization under a volume constraint for the L-bracket shown in Fig. 5.8 (a) is considered as the first example. The optimization problem is expressed as follows

$$\begin{aligned} &\text{minimize} \quad \boldsymbol{\sigma}_{PN} \\ &\text{subject to} \quad \textstyle\int_{\Omega} \mathrm{d}\Omega \leq \bar{V} \end{aligned} \tag{5.14}$$

where $\bar{V}$ is the limit for the volume constraint set as 70% and the $p-$norm value used is 6. Starting from a design without any holes as shown in Fig. 5.8 (b), the optimum hook-like structure of Fig. 5.8 (c) is obtained as expected. For this result the cubic kernel spline was used for the construction of RK shape functions. In the following paragraphs we make use of this example to investigate the effects of different features such as particle distribution and kernel function choice.



(a)

Figure 5.8: L-bracket example: (a) problem definition, (b) von Mises stress field for the initial design and (c) von Mises stress field for the optimum solution

Regular particle distribution

Having a regular particle distribution is useful when trying to reduce the computa-tional cost during optimization. This is because with such a distribution, new particles are only inserted along the boundary at each iteration. The supports of these newly inserted particles cover the particles in the region around them and thus, shape functions of the particles only in that region need to change accordingly. At each iteration, particles further away from the current location of the boundary, *i.e.*, a certain value of the signed distance function, will have the exact same neighbors as in the initial configuration before optimization begun. Thus, their information can be stored before optimization starts and then reused throughout the optimization process. This is illustrated in Fig. 5.9. In Fig. 5.9 (a) the information for particle "P" including the support domain and the stiffness matrix is computed on the regular grid. In fig. 5.9 (b) after new particles have been inserted on the boundary created by the circular hole, the information for particles within the highlighted region has to be computed at this step, whereas since particle "P" is unaffected, its information can be read from the previous step. Support sizes vary depending on neighbor distances. However, since particles are irregular only up to the neighbors of boundary points, this variation is limited compared to support sizes of the regularly distributed particles in the interior region. This means that the size of the zone within which the shape functions need to be recomputed does not have to be much larger than the size of the average support size. A size of twice the size of a support of an interior particle is sufficiently large to make sure that prestored information is only used for particles that are not covered by new particles thus, reducing the errors in the computation. The scheme described above is shown for the stress minimization of the L-Bracket example in Fig. 5.10. The dark highlighted zone indicates particles for which the information has to be recomputed. The light highlighted regions indicate particles for which the prestored information can be used. In terms of computational efficiency, we compare the overall time for the whole optimization process with and without the prestore scheme for the same number of iterations. The prestore scheme turns out to be 1.4 times faster than the normal computation ($\frac{without \quad prestore}{with \quad prestore} \approx 1.4$). The overall assembly time for the stiffness matrix for the entire optimization process is two times faster for the prestore scheme. It is of course expected that this difference will be more significant as the number of particles increases.

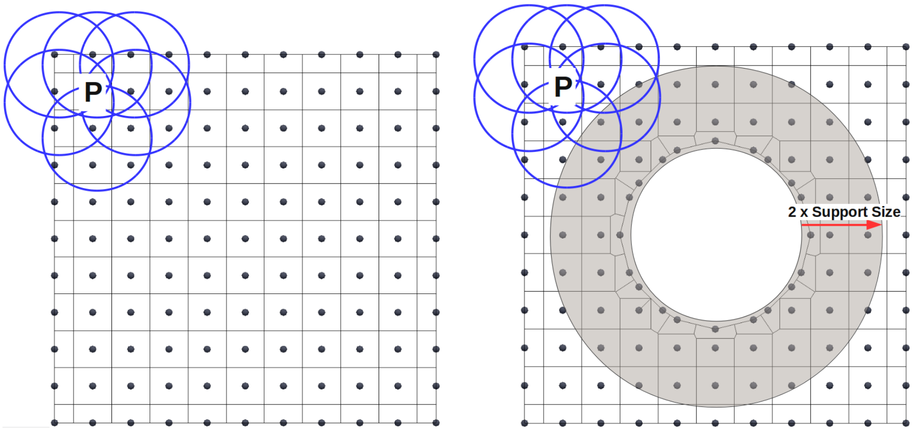
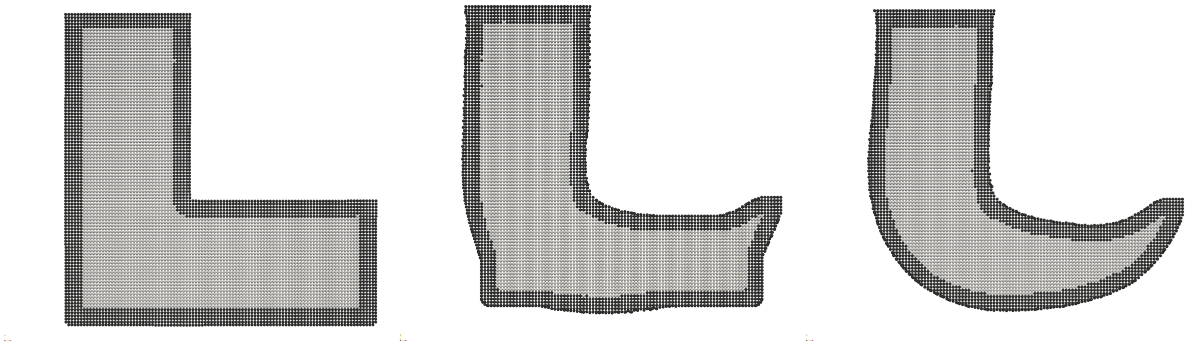


Figure 5.9: Prestore scheme: (a) Supports and local stiffness matrices are computed and stored for a regular particle arrangement before optimization starts and (b) Once new boundary particles are created, information only has to be computed for particles in the highlighted zone. Particles such as particle P do not see any change and thus the information can be used from the initial regular distribution.



Iter. 0 Iter. 60 Iter. 140

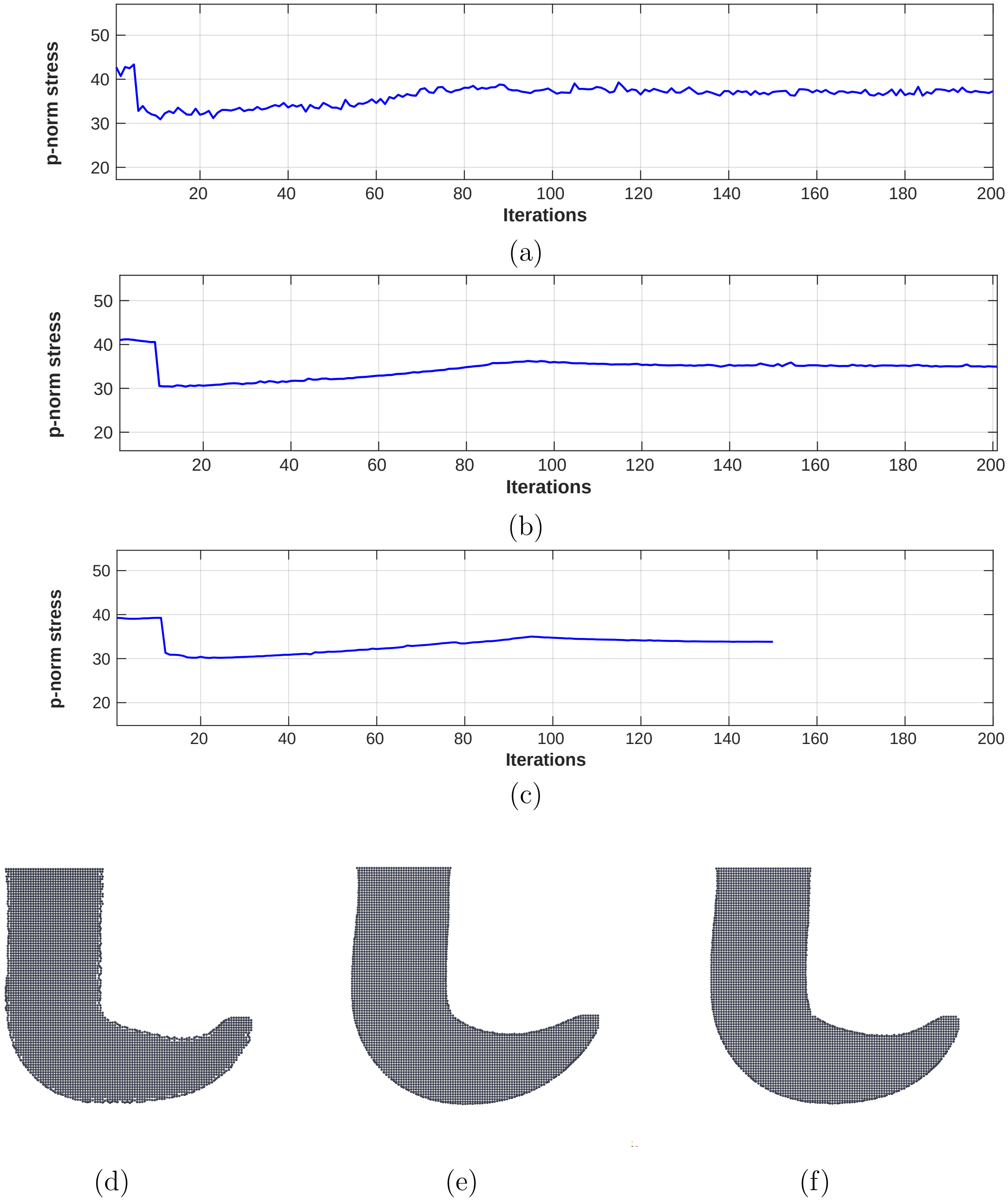
Figure 5.10: Prestore scheme for the L-Bracket example: The darkly highlighted zone indicates particles for which the information has to be recomputed. Lightly highlighted regions indicate particles for which prestored information can be used.

Different choices of kernel functions

RKPM provides controllable orders of continuity and completeness independent from one another, without additional complexity just by choosing the kernel function Φ_a and the basis $\mathbf{H}^T(\mathbf{x})$, respectively, in Eq. 3.20. We investigate here with three different kernel functions to see the effect on the convergence behavior. Specifically we solve the L-Bracket example with the linear (C^0 continuity) and cubic (C^2 continuity) kernel functions given in Sec. 3.2.1 and the quintic kernel (C^4 continuity) given as,

$$\Phi_a(\mathbf{x}-\mathbf{x}_I) = \begin{cases} 1 - \frac{90}{11}z_I^2 + \frac{405}{11}z_I^4 - \frac{405}{11}z_I^5 & for \quad 0 \leq z_I \leq \frac{1}{3} \\ \frac{17}{22} + \frac{75}{22}z_I - \frac{315}{11}z_I^2 + \frac{674}{11}z_I^3 - \frac{1215}{22}z_I^4 + \frac{405}{22}z_I^5 & for \quad \frac{1}{3} \leq z_I \leq \frac{2}{3} \\ \frac{81}{22} - \frac{405}{22}z_I + \frac{405}{11}z_I^2 - \frac{405}{11}z_I^3 + \frac{405}{22}z_I^4 - \frac{81}{22}z_I^5 & for \quad \frac{2}{3} \leq z_I \leq 1 \\ 0 & for \quad z_I > 1 \end{cases} \tag{5.15}$$

The optimization is left to run for 200 iterations. Figures 5.11 (a), (b) and (c) show the history of $p-$norm stress for the linear, cubic and quintic kernel functions respectively. Figures 5.11 (d), (e) and (f) present the final topologies for the three cases. For the 1st and the 3rd order kernels although the final structure is reached around iteration 140, the 0.001 tolerance is not achieved before 200 iterations whereas with the 5th order kernel the tolerance is achieved at 148 iterations as shown in Fig. 5.11. The final hook-like optimum solutions are obtained in all three cases, although boundary oscillations are visible in the case of the linear kernel function as shown in Fig. 5.11 (d). As the order of the kernel function increases, the continuity/smoothness of the RK shape function increases accordingly. As can be seen from these results increasing the order of the kernel can make the optimization process smoother and speed up the convergence. For higher order kernel functions the number of particles remains the same. Comparing Eq. 5.15 and Eq. 3.12 in Sec. 3.2.1, the only difference is that the higher order kernel has a few additional terms of higher exponents of z. The difference in computational time turns out to be insignificant (≈ 2 % difference).



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.11: Convergence history of the minimum stress L-bracket design for different kernel functions: (a) Linear B-spline (C^0 continuity), (b) Cubic B-spline (C^2 continuity) and (c) Quintic B-spline (C^4 continuity), (d) optimum solution for linear kernel function, (e) optimum solution for cubic kernel function and (f) optimum solution for quintic kernel function

Quadtree particle distribution

As explained in Sec. 3.1, the level set function is updated based on shape sensitivities computed at the boundary points. Good accuracy is thus more important close to the boundary than in the interior of the structure. The flexibility that RKPM offers in adaptivity can be used to create a denser particle distribution within a narrow band close to the boundary whereas keeping the distribution coarser away from the boundary at each iteration. Such a scheme is shown in Fig. 5.12 for the L-Bracket example where a quadtree structure is used to create the particle distribution. Specifically the particles away from the boundary are kept regular and coarse, while within a narrow band close to the boundary the distribution gets denser according to the quadtree data structure. As can be seen in the figure, the same end result is obtained as with the fully dense particle arrangement in Fig. 5.8, only here much less particles are used. In the case of the fully dense particle distribution, the optimization starts with 6601 particles for the first iteration and ends with 5090 particles for the last iteration, whereas the quadtree distribution starts with 1838 particles and ends with 1542 particles. Regarding computational efficiency, the overall time for the whole optimization process is 1.5 times faster for the quadtree distribution compared to the regular dense distribution (i.e. $\frac{regular \quad distribution \quad time}{quadtree \quad distribution \quad time} \approx 1.5$) which again will scale up as the mesh size increases. The true benefits of this scheme are expected to be more significant for larger scale and 3-dimensional structures and this will be the aim of our future work. Moreover, for such larger scale problems the quadtree distribution can be combined with the prestore scheme for the regularly spaced interior particles for even better efficiency.

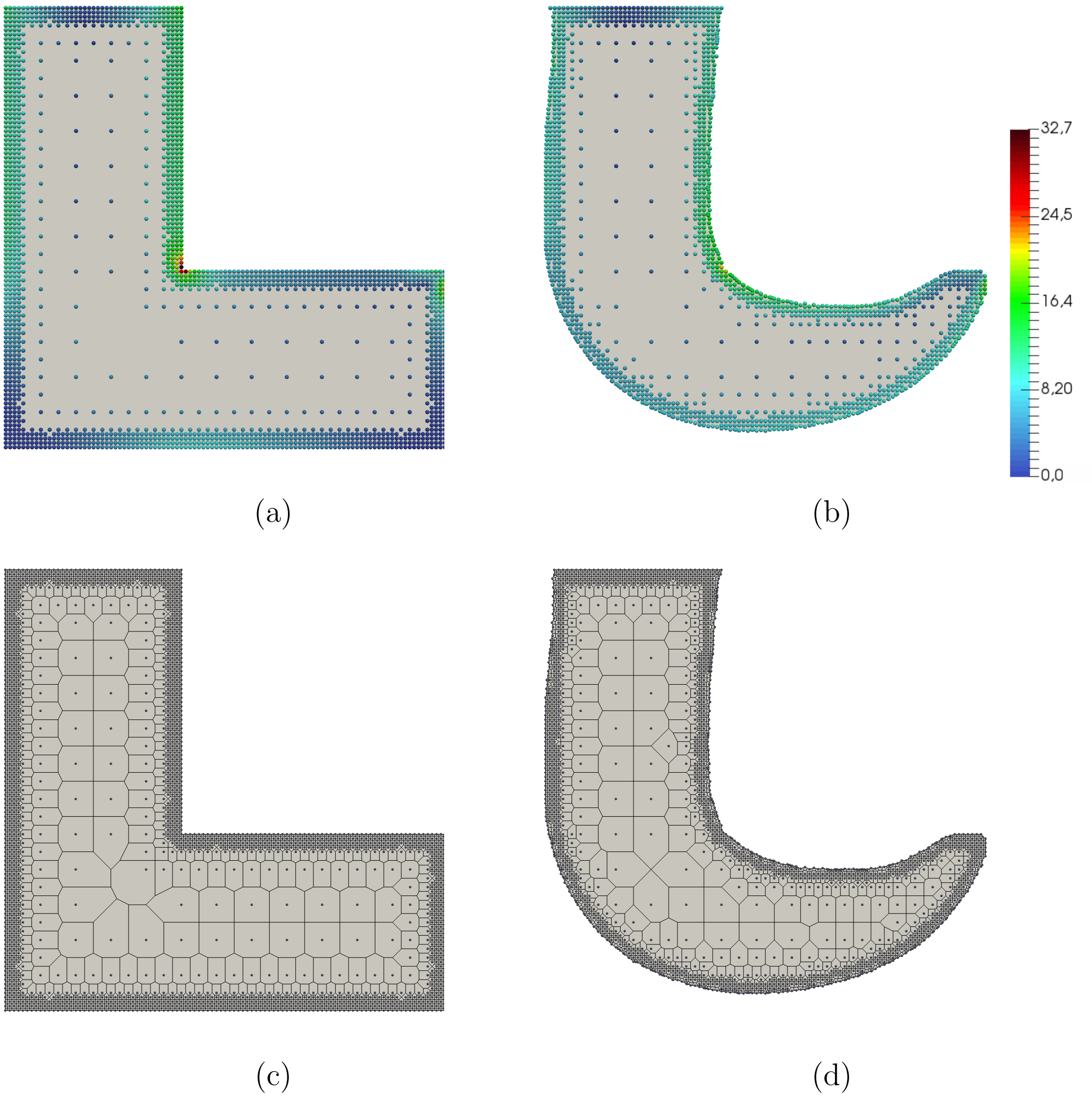
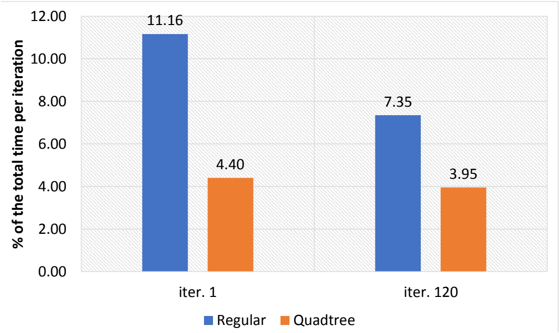


Figure 5.12: Stress minimization for the L-bracket example with quadtree particle distribution: (a) Initial design with von Mises stress field, (b) optimum solution with von Mises stress field, (c) Voronoi diagram for initial design and (b) Voronoi diagram for optimum solution

The time spent for the construction of the Voronoi diagram per iteration is shown in Fig. 5.13 for both regular and quadtree particle distributions. The time comparison is done first at the initial iteration when the number of particles is maximum and thus the process will take the most time. The comparison is repeated for iteration 120 at which point the number of particles is decreased due to the removal of material. As can be seen, the construction of the Voronoi diagram takes about 11% of the total time per iteration for the regular distribution at the first iteration. For the quadtree distribution the process is faster due to the smaller number of particles, taking about 4.5% of the total time. As expected, the process becomes faster for less particles at iteration 120 in both cases. Alternative implementations for the construction of the Voronoi diagram can of course further improve the computational efficiency. For example, since the positions of the particles in the interior of the domain do not change as the topology evolves, only partial reconstruction of the Voronoi diagram close to the boundary is required. Such an efficient implementation will be further examined in the future. It is important however, to note that the main difference between the Voronoi diagram used here and remeshing, is that the RKPM with NSNI is not sensitive to the shape of the polygons, as long as the polygons are conforming. Once the diagram is constructed, no additional operations are required to modify the shape of the polygons, whereas for FEA with remeshing additional operations are performed to improve the mesh quality.



(a)

Figure 5.13: Average time per iteration spent for Voronoi diagram construction for (a) regular particle distribution and (b) Quadtree particle distribution.

Example 2: Volume minimization under Stress Constraint

In this example the volume minimization under a stress constraint is considered. As explained in Picelli et al. (2018) the p-norm stress is always greater than the actual maximum. Thus, an adaptive scaling scheme is used,

$$\boldsymbol{\sigma}_{max} \approx c^k \boldsymbol{\sigma}_{PN} \leq \bar{\boldsymbol{\sigma}} \tag{5.16}$$

where $\bar{\boldsymbol{\sigma}}$ is the stress constraint limit and c is defined as

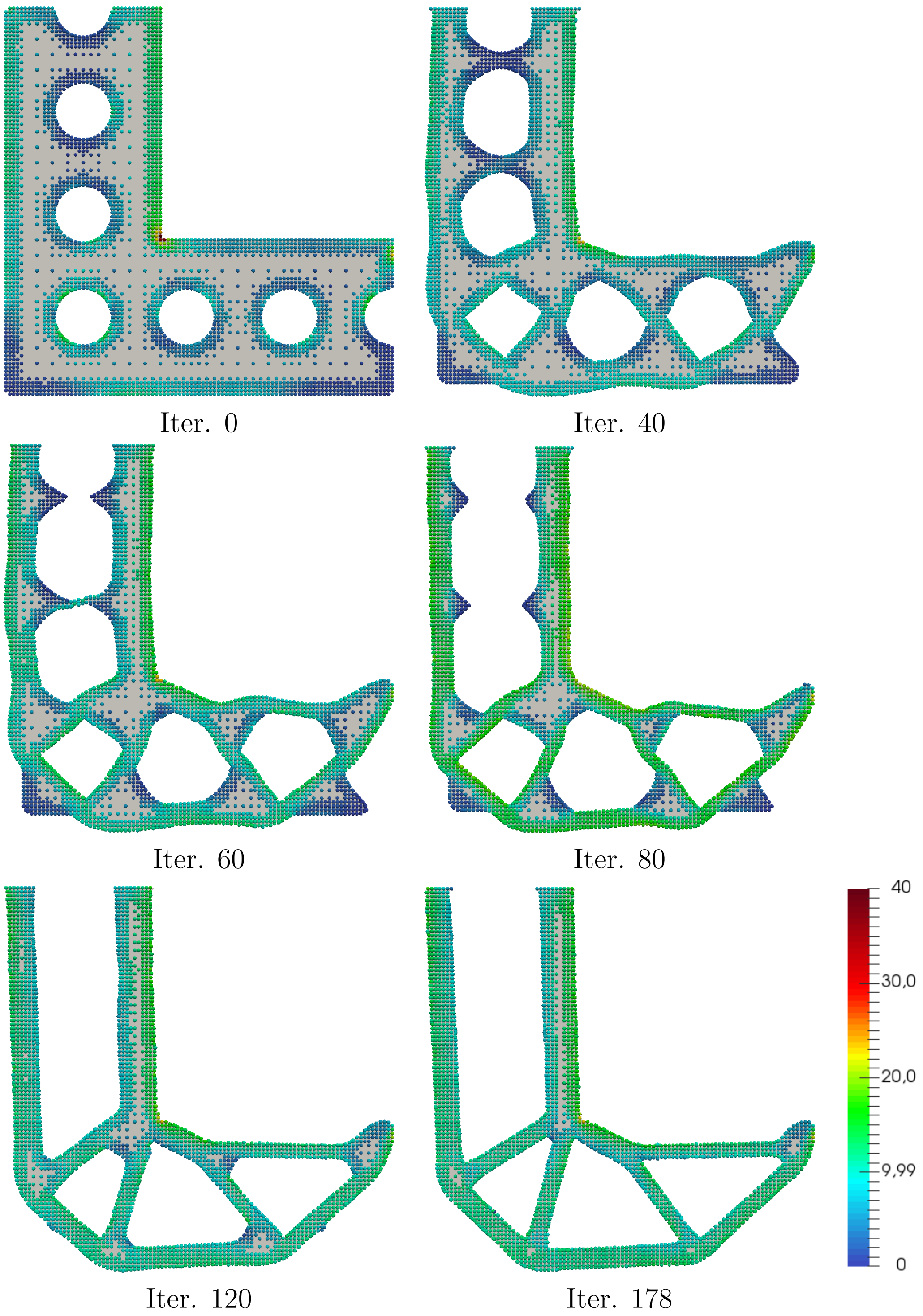
$$c^k = \eta^k \frac{\boldsymbol{\sigma}_{max}^{k-1}}{\boldsymbol{\sigma}_{NP}^{k-1}} + (1 - \eta^k) c^{k-1} \tag{5.17}$$

where $\eta \in (0, 1]$ controls the variations between c^k and c^{k-1}. The scaling uses information from the previous iteration to normalize the constraint as $c^k \boldsymbol{\sigma}_{PN}$ so it approximates the actual maximum stress $\boldsymbol{\sigma}_{max}$ as the design converges.

The optimization problem is formulated as,

$$
\begin{aligned}
&\text{minimize} \quad \int_\Omega \mathrm{d}\Omega \\
&\text{subject to} \quad c^k \boldsymbol{\sigma}_{PN} \leq \bar{\boldsymbol{\sigma}}
\end{aligned}
\tag{5.18}
$$

The cubic kernel spline is used in this example and the evolution of the structure throughout optimization is shown in Fig. 5.14. Starting from the initial design with holes and for a stress constraint $c^k \boldsymbol{\sigma}_{PN} \leq 20$, optimization converges at 178 iterations. The convergence history for σ_{PN}, σ_{max} and the volume is shown in Fig. 5.15. This example also illustrates the robustness of the methodology in the presence of holes and significant topological changes.



Iter. 0

Iter. 40

Iter. 60

Iter. 80

Iter. 120

Iter. 178

Figure 5.14: Volume minimization with a stress constraint for the L-bracket example: Evolution of the structure with the von Mises stress field

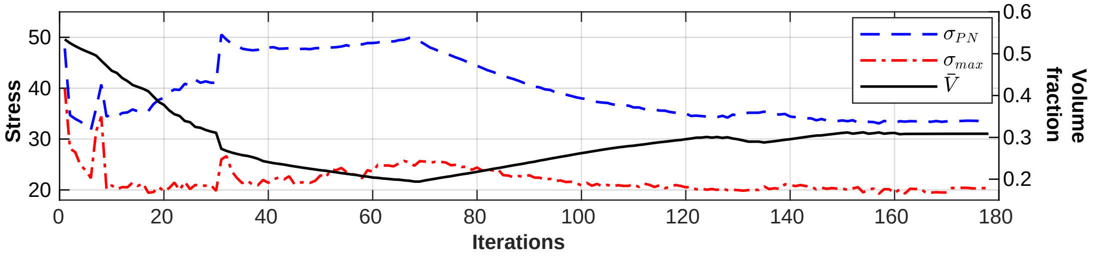


Figure 5.15: Convergence history of the minimum volume, stress constraint L-bracket design.

5.3.2 Example 3: A design-dependent problem: Piston-head example

The example in Fig. 5.16 is used to illustrate two useful features of the proposed methodology. The first, is the ability to handle problems with design-dependent surface loads where the loads can change in magnitude and direction during optimization. Fluid-structure interactions and hydrostatic pressure problems fall under this category. Here we consider hydrostatic pressure loads of constant magnitude. In such problems a clear boundary identification on the computational model is essential to correctly apply the loads at each iteration. Due to the absence of an interpolation scheme and the conforming description of the structure, the LSTO-RKPM methodology proposed here preserves the clear boundary from the level set method on the computational domain. Thus, it can straightforwardly handle such problems by directly applying the loads on the boundary without any interpolation schemes.

The second feature is the computational efficiency of this methodology compared to the Gaussian integration for the domain. This specific example was solved in Ch. 4 using the Gaussian quadrature on a fixed background mesh and an interpolation scheme based on nodal signed distance values using the RK shape functions. The two methodologies are com-pared here. The aim is to minimize the compliance under a volume constraint and the problem is formulated as,

$$\begin{aligned} \min \quad & J = l(u) \\ \text{subject to} \quad & a(u, v) = l(v) \\ & V_s(\Omega) - \bar{V} \leq 0 \end{aligned} \tag{5.19}$$

where the energy bilinear functional $a(\mathbf{u}, \boldsymbol{v})$ and the load linear form $l(\boldsymbol{v})$ are defined as:

$$a(\mathbf{u}, \boldsymbol{v}) = \int_\Omega \{\boldsymbol{\varepsilon}(\mathbf{u})\}^T : \mathbf{D} : \{\boldsymbol{\varepsilon}(\boldsymbol{v})\} \, d\Omega \tag{5.20}$$

$$l(\boldsymbol{v}) = \int_{\Gamma_N} \boldsymbol{p} \cdot \boldsymbol{v} d\Gamma \tag{5.21}$$

Here $V_s(\Omega)$ is the volume fraction of the structure with respect to the design domain, $\bar{V}$ is the maximum allowed volume fraction, $\boldsymbol{\varepsilon}$ is the strain tensor, $\boldsymbol{v}$ is the virtual displacement, Γ_N is the Neumann boundary on which the pressure load is applied and p is the pressure load. The pressure load is assumed to be constant although the method can be generalized for varying pressure.

$$\boldsymbol{p} = -p_0\boldsymbol{n} \tag{5.22}$$

where p_0 is the constant magnitude of the pressure load, and $\boldsymbol{n}$ is the surface normal.

Shape sensitivity for the structural compliance function when the surface load is a pressure load was derived by Allaire et al. (2004) as,

$$\frac{\partial f}{\partial \mathrm{x}} = \int\limits_{\Gamma_N} \left[-2\mathrm{div}\left(p_0\boldsymbol{u}\right) - \boldsymbol{\sigma}(\boldsymbol{u}) \cdot \boldsymbol{\varepsilon}\left(\boldsymbol{u}\right)\right] V_n \mathrm{d}\Gamma, \tag{5.23}$$

where p_0 is the pressure load and V_n is the normal velocity on the boundary.

The result obtained here as shown in Fig. 5.17 is the same as the one in Neofytou et al. (2020) and the solution is achieved in about the same number of iterations. However now the process is about 20 times faster. This is because in the case of the Gaussian integration approach almost $3{\times}10^5$ Gauss points were used to achieve the sufficient accuracy (16 Gauss points per cell), whereas the nodally integrated RKPM achieves the same level of accuracy with only 1800 particles. Fig. 5.17 shows the evolution of the structure. As can be seen the methodology is effective in managing the large topological changes such as holes merging and islands disappearing.

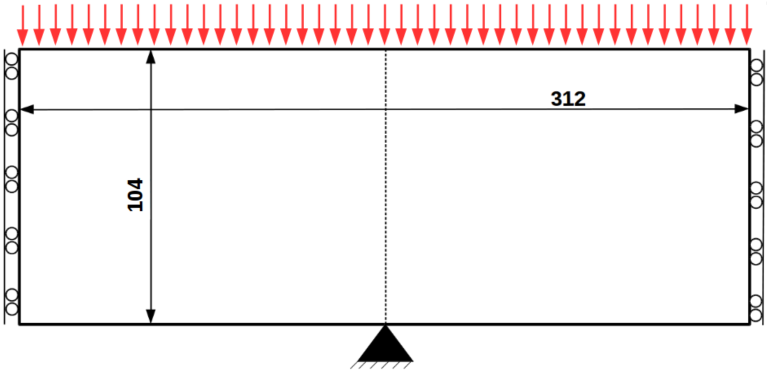
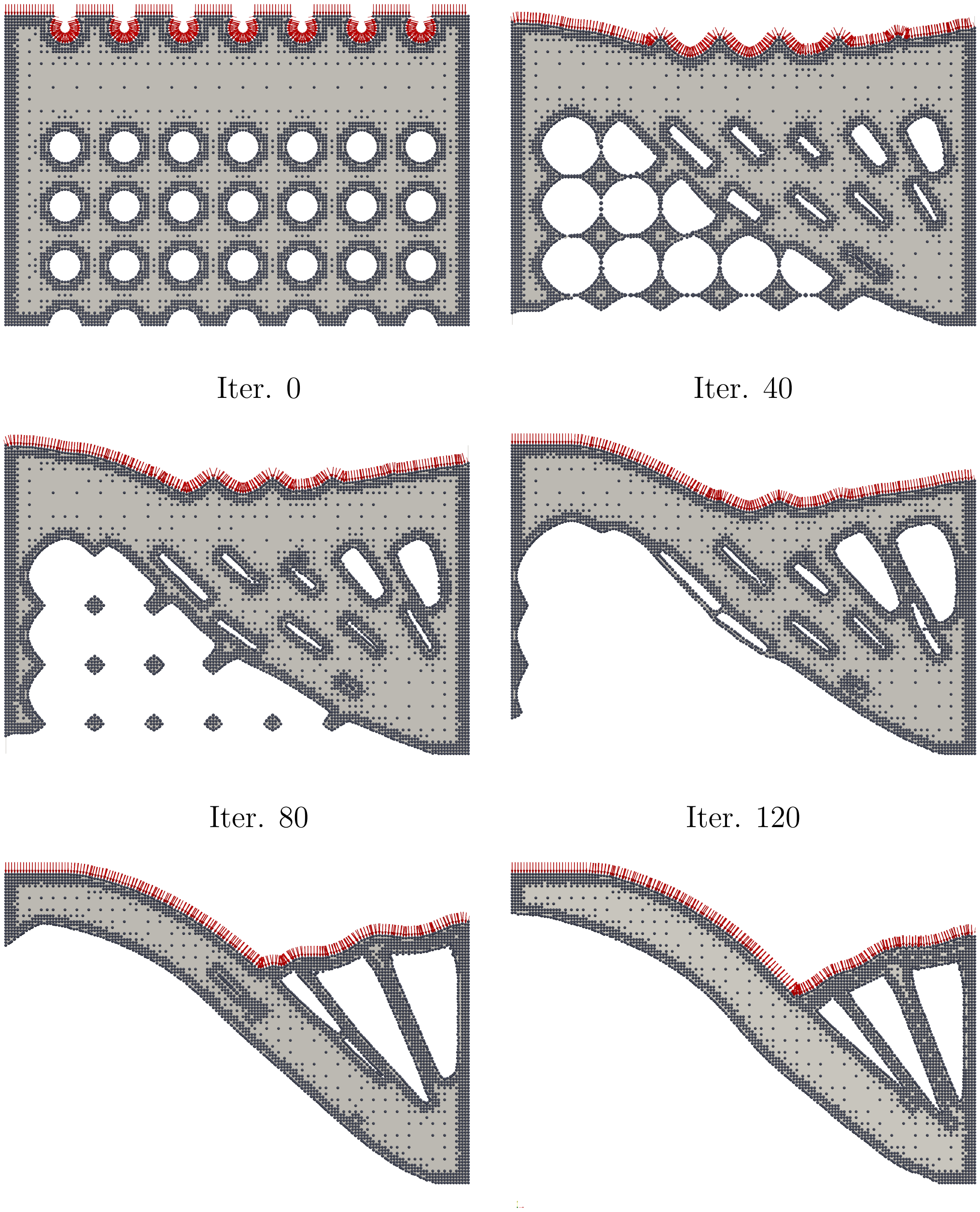


Figure 5.16: Piston-head structure problem definition.



Iter. 0

Iter. 40

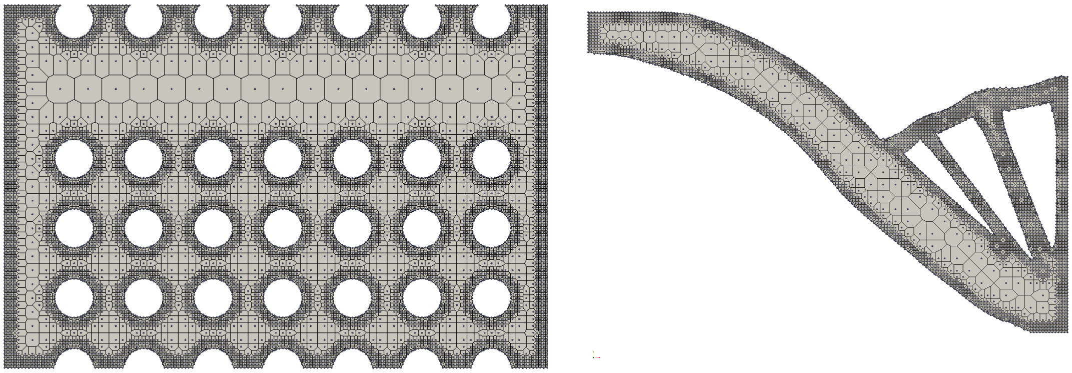
Iter. 80

Iter. 120

Iter.140

Iter. 160

Figure 5.17: Snapshots of the piston-head solution with pressure loads



(a) (b)

Figure 5.18: Voronoi diagram for the piston-head example: (a) Initial diagram and (b) Diagram at the optimum solution

5.4 Conclusion

In this chapter a combination of the nodally integrated RKPM with the LSTO method has been proposed for an exact geometry description of structures during optimization without any interpolation schemes and without remeshing. The effectiveness and robustness of the methodology has been illustrated through stress-based examples and the piston-head design-dependent problem with hydrostatic pressure loads examined in Ch. 4. As shown, the method presents useful features such as the ability to change the order of continuity simply by changing the continuity order of the kernel function without adding computational complexity. Increasing the order of continuity in the RK shape function has been shown to speed up the convergence without an increase in the computational time. Moreover, the method is efficient in handling different particle distributions. This can be used to increase efficiency by increasing the particle density only around the boundaries where the accuracy for sensitivity computation is more crucial. As illustrated through the piston-head example, this approach is significantly more efficient than the Gauss integration procedure of Ch. 4 while providing the same results. Furthermore, large topological changes such as holes merging and disappearing can be handled naturally.

Chapter 6

Level Set Topology Optimization for FSI

In chapter 4 a methodology was described for solving design-dependent problems using LSTO-RKPM. By replacing the background mesh integration with nodally integrated RKPM it was shown in Ch. 5 that superior computational efficiency can be achieved while obtaining the same results for hydrostatic loads as in Ch. 4. Moreover, the robustness of the nodally integrated RKPM was illustrated through well-known challenging stress-based problems. Several useful features such as efficient particle schemes and employment of higher order RK shape functions were also examined. Furthermore, the exact description of the structure provided by the LSTO-nodally integrated RKPM results in a clear boundary description on the computational domain.

With these ingredients at hand, the work on design-dependent hydrostatic loads can now be extended to a more challenging class of design-dependent physics, i.e., fluid-structure interactions, where a viscous fluid governed by the Navier-Stokes equations, interacts with linear or non-linear solids. For this purpose, the immersed finite element method is used for the coupling, which allows for the solid and fluid domains to be analyzed with different discretization methods. Specifically, the solid domain is analyzed and optimized via the nodally integrated RKPM and level set topology optimization, whereas for the fluid, the finite element analysis is performed.

A transient framework is developed that can potentially be used to solve transient FSI problems. However, for validation purposes the optimization examples presented are solved for steady-state conditions and then compared with the results from the literature. There are currently no works on transient FSI topology optimization in the literature. This is of course not a surprise since high fidelity transient FSI problems pose great challenges especially in the context of topology optimization. Such challenges include sensitivity computation and computational cost. Nevertheless, the approach developed here provides a solid base for an extension to transient FSI problems where fluids in contact with non-linear solids may be considered.

6.1 Modified immersed finite element method

For the FSI analysis, the fluid equations are solved using an Eulerian background mesh whereas the solid is geometrically described by the level set method and the solid equations are solved with the RKPM method. The full details of the mIFEM can be found in Wang and Zhang (2013) and its opensource implementation named OpenIFEM in Cheng et al. (2019). The main equations are provided in this chapter along with the implementation details for the FSI level set topology optimization.

6.1.1 Solid domain

The dynamic solid equation is given by

$$\rho^s a_i^s = \sigma_{ij,j}^s \quad \text{in} \quad \Omega^s \tag{6.1}$$

where $\mathbf{a}^s$ is the solid acceleration and $\boldsymbol{\sigma}^s$ is the solid stress defined as

$$\sigma_{kl}^s = C_{ijkl}\varepsilon_{ij}^s + \eta_{ijkl}\varepsilon_{ij}^s \quad \text{in} \quad \Omega^s \tag{6.2}$$

where $\boldsymbol{\varepsilon}^s$ is the solid strain tensor defined as $\varepsilon_{ij}^s = \frac{1}{2}(u_{i,j}^s + u_{j,i}^s)$, $\mathbf{u}^s$ is the solid displacement, and different combinations of C_{ijkl} and η_{ijkl} result in different material constitutive laws such as linear elastic and hyperelastic materials. The Dirichlet and Neumann boundary conditions on the boundaries Γ^{sq} and Γ^{sh}, respectively, are given by

$$u_i^s = q_i \quad \text{on} \quad \Gamma^{sq} \tag{6.3}$$

$$\sigma_{ij}^s n_j = h_i \quad \text{on} \quad \Gamma^{sh} \tag{6.4}$$

where $\Gamma^s = \Gamma^{sq} \cup \Gamma^{sh}$ and $\Gamma^{sq} \neq \emptyset$ to ensure the existence and uniqueness of the solid solution.

The interface conditions on the solid boundary in contact with the fluid can be applied as either a Dirichlet boundary condition on Γ^{sq} (i.e., the solid displacement can be prescribed as the fluid displacement in a selected portion of the boundary) and/or as a Neumann boundary condition on Γ^{sh} (i.e. as a traction on the solid boundary due to the action of the fluid). These two boundary conditions can co-exist, but cannot overlap. They are computed using the fluid velocity and pressure fields obtained from the fluid solution at the previous time step as

$$q_i = \left[\int_\Omega \upsilon_i^f \phi(\mathbf{x}^f - \mathbf{x}^s) d\Omega \right] \Delta t \quad \text{on} \quad \Gamma^{sq} \tag{6.5}$$

$$h_i = \left[\int_\Omega \sigma_{ij}^f \phi(\mathbf{x}^f - \mathbf{x}^s) d\Omega \right] n_j \quad \text{on} \quad \Gamma^{sh} \tag{6.6}$$

where q_i is the solid displacement interpolated from the fluid region onto the solid interface. The fluid traction acting on the solid interface is represented by h_i. In this work the interface condition is applied as a Neumann boundary condition on the solid boundary and thus Eq. 6.5 is not used. Function ϕ is an interpolation function which is a function of the distance between a fluid grid point $\mathbf{x}^f$ and a solid point $\mathbf{x}^s$.

The solid velocities are then computed by solving the solid governing equation, Eq. 6.1. The initial displacement and velocity conditions for the solid are also needed to solve the equations. If the solid starts from a resting position, these are initially set to zero.

The set of the dynamic solid equations is discretized using the RKPM methodology described in Ch. 5 and solved numerically using the α-method (Hughes, 2000; Chen et al., 1996), which is an implicit and unconditionally stable scheme, meaning that there is no limitation to the time step size.

6.1.2 Artificial fluid domain

In the mIFEM, part of the fluid called the artificial fluid, overlaps with the solid domain. The idea of the artificial fluid is that it physically does not exist, rather, it should mimic the solid as much as possible. Consequently, the artificial fluid should produce the same velocity as the solid

$$\mathbf{v}^f = \mathbf{v}^s \quad \text{in} \quad \bar{\Omega} \tag{6.7}$$

where $\bar{\Omega}$ denotes the artificial fluid domain, $\mathbf{v}^f$ is the velocity vector in the artificial region and $\mathbf{v}^s$ is the solid velocity vector.

The continuity equation for the artificial fluid should allow compressibility for two reasons: First, to ensure that the artificial fluid velocity is the same as the solid velocity and second when any solid volume change needs to be considered. Thus, the continuity equation can be written as

$$\frac{1}{\kappa^s} \frac{\partial p}{\partial t} + \upsilon_{i,j}^f = 0 \quad \text{in} \quad \bar{\Omega} \tag{6.8}$$

where κ^s is the bulk modulus of the solid, since we want the compressibility of the artificial fluid to be the same as the solid to ensure the velocity in the artificial fluid domain is the same as the solid velocity.

The momentum Navier-Stokes equation is

$$\rho^s \frac{\partial v_i^f}{\partial t} + \rho^s v_i^f v_{i,j}^f = \sigma_{ij,j}^f + f_i^{FSI,f} \quad \text{in} \quad \bar{\Omega} \tag{6.9}$$

where ρ^s is the solid density, again to ensure that the artificial fluid behaves as the solid. The interaction force is used to further enforce $\mathbf{v}^f = \mathbf{v}^s$ in $\bar{\Omega}$. It is first evaluated in the solid domain and then distributed onto the fluid domain through the interpolation function as

$$\mathbf{f}^{FSI,f} = \int_{\Omega_s} \mathbf{f}^{FSI,s} \phi(\mathbf{x} - \mathbf{x}^s) d\Omega^s \tag{6.10}$$

The fluid-structure interaction force per unit volume represents the viscous effects due to the existence of the solid in the fluid domain and is defined as

$$f_i^{FSI,s} = \sigma_{ij,j}^s - \sigma_{ij,j}^f \quad \text{in} \quad \Omega^s \tag{6.11}$$

in which $\boldsymbol{\sigma}^f$ is the fluid stress and $\boldsymbol{\sigma}^s$ is the solid stress.

6.1.3 Fluid domain

The fluid domain including both the real fluid and the artificial fluid region is governed by the Navier-Stokes equations. The real fluid can be either compressible or incompressible but in this work the incompressible case is considered. The main equations are given below.

The continuity equation expressing the mass conservation is written as

$$\begin{cases} v_{i,i}^f = 0, & \text{in} \quad \Omega^f \\ \frac{1}{k^s}\frac{\partial p^f}{\partial t} + v_{i,i}^f = 0, & \text{in} \quad \bar{\Omega} \end{cases} \tag{6.12}$$

and the momentum conservation Navier-Stokes equations are

$$\begin{cases} \rho^f \frac{\partial v_i^f}{\partial t} + \rho^f v_j^f v_{i,j}^f = \sigma_{ij,j}^f, & \text{in} \quad \Omega^f \\ \rho^s \frac{\partial v_i^f}{\partial t} + \rho^s v_j^f v_{i,j}^f = \sigma_{ij,j}^f + f_i^{FSI,f} & \text{in} \quad \bar{\Omega} \end{cases} \tag{6.13}$$

The fluid stress is given by

$$\sigma_{ij}^f = -p^f \delta_{ij} + \tau_{ij}^f \tag{6.14}$$

where τ_{ij}^f is the viscous stress

$$\tau_{ij}^f = \mu(v_{i,j}^f + v_{j,i}^f) \tag{6.15}$$

The real fluid and artificial fluid domains are combined together and solved numerically through the use of an indicator delta function $\mathbf{I}(\mathbf{x}^f)$ defined as

$$\mathbf{I}(\mathbf{x}^f)\begin{cases} 0, & \text{in} \quad \Omega^f \\ \\ 1, & \text{in} \quad \bar{\Omega} \end{cases} \tag{6.16}$$

where $\mathbf{x}^f$ are the fluid element coordinates. In this work, the solid velocity is interpolated at the fluid quadrature points of the artificial fluid region using the RK shape functions. The artificial fluid region based on this indicator field does not extend beyond the solid boundary. The requirement for a fluid element to be considered "artificial" fluid, is that all the nodes of the fluid element must be within the solid boundary. Thus, there is no smearing of the solid properties beyond this region on the fluid side.

The continuity and momentum equations from Eq. 6.12 and Eq. 6.13 can then be expressed in terms of the indicator function as shown in Eq. 6.17 and Eq. 6.18 below, respectively.

$$\frac{1}{k^s}\frac{\partial p^f}{\partial t}\mathbf{I}(\mathbf{x}^f) + v^f_{i,i} = 0 \quad \text{in} \quad \Omega \tag{6.17}$$

$$\bar{\rho}\frac{\partial v^f_i}{\partial t} + \bar{\rho}v^f_j vi, j^f = \sigma^f_{ij,j} + f^{FSI,f}_i \quad \text{in} \quad \Omega \tag{6.18}$$

where $\bar{\rho}$ is defined as

$$\bar{\rho} = \rho^f + (\rho^s - \rho^f)\mathbf{I}(\mathbf{x}^f) \quad \text{in} \quad \Omega \tag{6.19}$$

In the opensource software of mIFEM (Cheng et al., 2019) the governing equations in the entire fluid domain are solved by the finite element method with the Matrix-free Newton Krylov method.

6.2 Transient FSI analysis

In this thesis, the solid domain is analyzed using the nodally integrated RKPM methodology described in Ch. 5. To illustrate the applicability of this combination, transient FSI analysis is performed here for a commonly used problem in the literature, known as the leaflet example.

6.2.1 Transient RKPM solver validation

Before performing the FSI analysis, the transient RKPM solver is validated for a 2D cantilever beam as shown in Fig. 6.1, which is made of Neo-Hookean material. This example is provided in the OpenIFEM opensource software (Cheng et al., 2019) where it is also validated against ABAQUS (Smith, 2014). Here, the analysis is compared against the OpenIFEM code. As mentioned in Sec. 6.1.1, the $\alpha-$method (Hughes, 2000; Chen et al., 1996) is used to solve the dynamic solid equations.
To begin, the discrete equation of motion can be expressed as

$$\mathbf{M}\mathbf{a}^s + \mathbf{C}\mathbf{v}^s + \mathbf{K}\mathbf{u}^s = \mathbf{F} \tag{6.20}$$

where $\mathbf{M}$ is the mass matrix formed as $M_{IJ} = \int_\Omega \rho^s \mathbf{\Psi}_I \mathbf{\Psi}_J \mathbf{I} d\Omega$, $\mathbf{C}$ is the viscous damping matrix, $\mathbf{K}$ is the stiffness matrix, $\mathbf{F}$ is the vector of applied forces and vectors $\mathbf{u}_n^s, \mathbf{v}_n^s$ and $\mathbf{a}_n^s$ represent displacement, velocity, and acceleration vectors for the solid, respectively. Solution of Eq. 6.20 using the Newmark method consists of the following equations

$$\mathbf{M}\ddot{\mathbf{d}}_{n+1}^s + \mathbf{C}\dot{\mathbf{d}}_{n+1}^s + \mathbf{K}\mathbf{d}_{n+1}^s = \mathbf{F}_{n+1} \tag{6.21}$$

$$\mathbf{d}_{n+1}^s = \mathbf{d}_n^s + \Delta t \dot{\mathbf{d}}_n^s + \frac{\Delta t^2}{2}[(1-2\beta)\ddot{\mathbf{d}}_n^s + 2\beta \ddot{\mathbf{d}}_{n+1}^s] \tag{6.22}$$

$$\dot{\mathbf{d}}_{n+1}^s = \dot{\mathbf{d}}_n^s + \Delta t[(1-\gamma)\ddot{\mathbf{d}}_n^s + \gamma \ddot{\mathbf{d}}_{n+1}^s] \tag{6.23}$$

where the subscript n denotes the previous time step, $\ddot{\mathbf{d}}^s$, $\dot{\mathbf{d}}^s$ and $\mathbf{d}^s$ are the approximations of $\mathbf{a}_n^s$, $\mathbf{v}_n^s$ and $\mathbf{u}_n^s$, respectively. The parameters β and γ determine the stability and accuracy of the algorithm under consideration. The $\alpha-$method then defines *predictors* in terms of the information from the previous time step n as:

$$\tilde{\mathbf{d}}_{n+1} = \mathbf{d}_n + \Delta t \dot{\mathbf{d}}_n + \frac{\Delta t^2}{2}(1-2\beta)\ddot{\mathbf{d}}_n \tag{6.24}$$

$$\tilde{\dot{\mathbf{d}}}_{n+1} = \dot{\mathbf{d}}_n + (1-\gamma)\Delta t \ddot{\mathbf{d}}_n \tag{6.25}$$

Eqs. 6.22 and 6.23 can be re-written using Eqs. 6.26 and 6.28 as

$$\mathbf{d}_{n+1} = \tilde{\mathbf{d}}_{n+1} + \beta \Delta t^2 \ddot{\mathbf{a}}_{n+1} \tag{6.26}$$

$$\dot{\mathbf{d}}_{n+1} = \tilde{\dot{\mathbf{d}}}_{n+1} + \gamma \Delta t \ddot{\mathbf{a}}_{n+1} \tag{6.27}$$

The overall process can be summarized as

1. Form the matrices $\mathbf{M}$, $\mathbf{K}$ and $\mathbf{F}$.

2. Compute initial nodal acceleration $\ddot{\mathbf{d}}^0$ using the initial nodal displacement $\mathbf{d}^0$ as
$$\mathbf{M}\ddot{\mathbf{d}}^0 = \mathbf{F} - \mathbf{C}\dot{\mathbf{d}}^0 - \mathbf{K}\mathbf{d}^0 \tag{6.28}$$

3. At the current time t = n+1 compute predictors according to Eqs. 6.26 and 6.28.

4. Solve for $\ddot{\mathbf{a}}_{n+1}$ using
$$(\mathbf{M} + \gamma\Delta t\mathbf{C} + \beta\Delta t\mathbf{K})\ddot{\mathbf{a}}_{n+1} = \mathbf{F}_{n+1} - \mathbf{C}\tilde{\dot{\mathbf{d}}}_{n+1} - \mathbf{K}\tilde{\mathbf{d}}_{n+1} \tag{6.29}$$

Note that in the case of nonlinear hyperelastic material, such as the Neo-Hookean model, the Newton's method is used to solve Eq. 6.29 iteratively within each time step.

The material properties used for the analysis are: shear modulus $G = 59.55$ kPa, bulk modulus $\kappa = 100$ kPa, density $\rho = 1.1 \times 10^6$ kg/m^3. The uniformly distributed traction on the top edge has a value of 2.5 Pa, and the time step for the analysis is $0.1s$. For the FEA analysis in OpenIFEM, 512 uniform first-order quadrilateral elements are used whereas 585 uniformly distributed particles are used for the RKPM analysis. Furthermore, it is assumed that damping is zero which means that all the terms involving $\mathbf{C}$ in Eq. 6.29 vanish. The vertical displacements obtained by the two methods at point P in Fig. 6.1 are compared. Under the load, the beam initially bends downward reaching a maximum deflection to the point at which the external work equals the elastic energy. Inertia then takes the beam back to its original position, and in the absence of damping the beam continues to bounce up and down with the same magnitude for the entire simulation time. The results shown in Fig. 6.2 indicate a good agreement between the two methods, and thus by extension, with the ABAQUS commercial software against which OpenIFEM was validated.

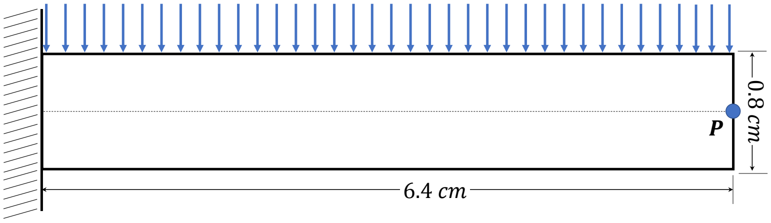


Figure 6.1: Cantilever beam˙test case

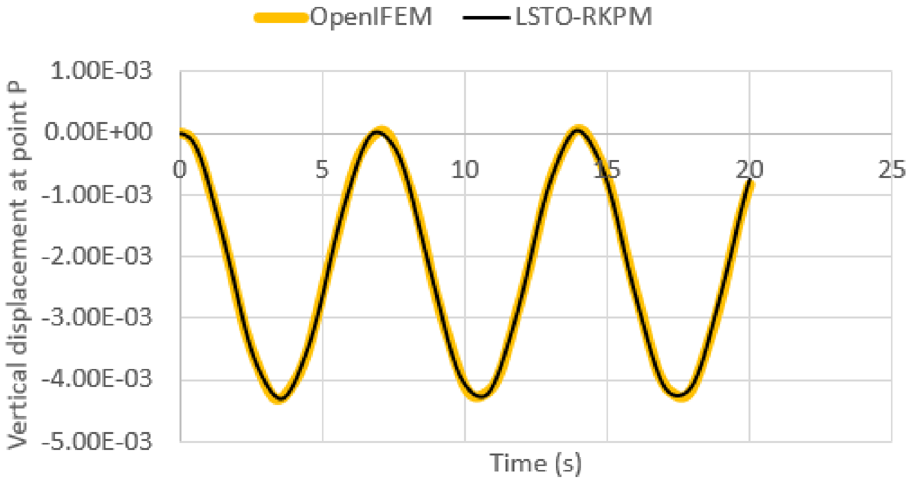
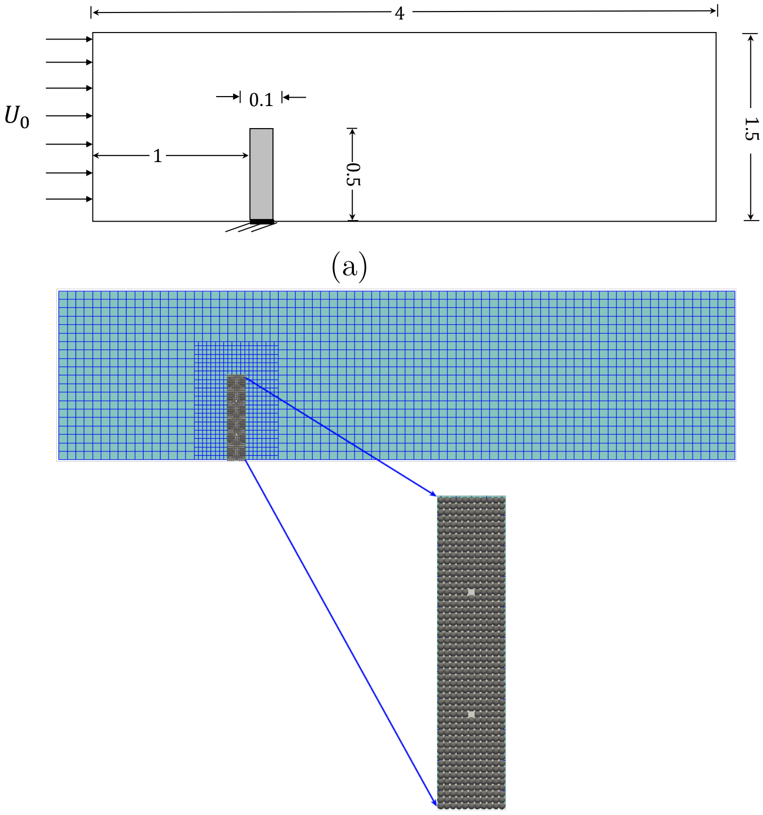


Figure 6.2: Validation of RKPM transient solver against OpenIFEM (Cheng et al., 2019)

6.2.2 Transient FSI analysis for the leaflet example

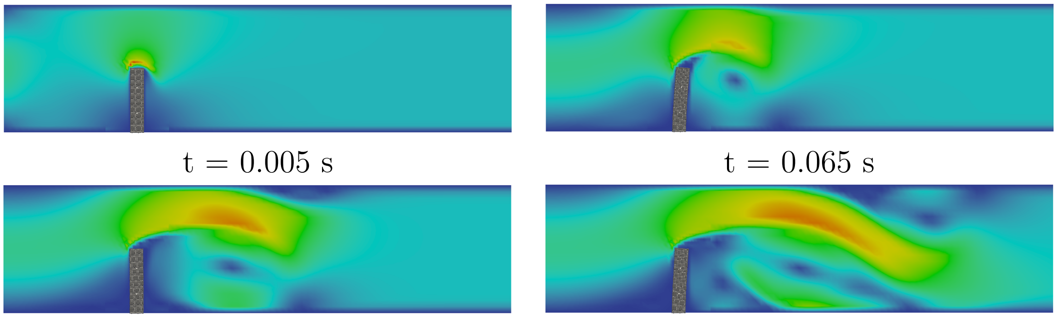
Following the validation of the transient solid solver, the transient FSI analysis is shown here for the leaflet example shown in Fig. 6.3 (a). The inlet velocity of the fluid has a value of $U_0 = 0.1\ m/s$. The fluid density is set to $\rho_f = 1000\ kg/m^3$ and the dynamic viscosity is $\mu = 0.01\ kg/(m \cdot s)$. The material is modeled as nearly incompressible Neo-Hookean with a shear modulus of $G = 7\ kPa$ and a bulk modulus of $\kappa = 0.86\ MPa$, which correspond to the initial Youngs modulus of 19 kPa and Poissons ratio of 0.49. The solid density is set to $\rho_s = 6000\ kg/m^3$. The bottom boundary is modeled as no-slip wall, while the upper boundary has no-penetration condition, and the right boundary is outflow. The fluid mesh is shown in Fig. 6.3 (b) with a refinement around the solid region. The beam is analyzed using 561 uniformly distributed RKPM particles. The time step is set to $\Delta t = 5 \times 10^{-3} s$ and the simulation is run until $t = 0.2s$. The velocity field and beam deformation at four different times are shown in Fig. 6.4. The beam undergoes large displacements when the flow begins. As the flow develops towards steady state, the amplitude of the beam displacement decreases until finally, the oscillations stop.



(a)

(b)

Figure 6.3: (a) Setup for the leaflet example (dimensions in *cm*) and (b) Fluid and solid discretizations and initial particle distribution



t = 0.005 s

t = 0.065 s

t = 0.12 s

t = 0.2 s

Figure 6.4: Transient FSI analysis result: Velocity field and particle distribution at different time steps

6.3 Discrete adjoint sensitivity analysis

One of the main challenges in topology optimization with complex physics is the computation of the gradients, or else sensitivities, of the objective function and constraints with respect to the design variables. The accuracy in the computation of these gradients is essential in the convergence of the gradient-based optimization algorithms. In topology optimization typically the number of design variables is usually much larger than the number of the objective and constraint functions. Because of this, adjoint methods (Choi and Kim, 2005) are preferred since they are able to provide accurate gradients with a cost that is independent of the number of design variables.

Adjoint methods are separated into *continuous* and *discrete* (Choi and Kim, 2005). In the continuous adjoint methods, the continuous governing equations are first differentiated in order to derive the adjoint equations in the continuous space. Once the adjoint system is obtained, it is discretized along with the governing field equations using a numerical method to compute sensitivities. In the discrete adjoint method on the other hand, the governing equations are first discretized and then differentiated to derive the adjoint equations in the discrete space. A schematic illustration of the difference between continuous and discrete adjoints for linear elasticity is shown in Fig. 6.5.

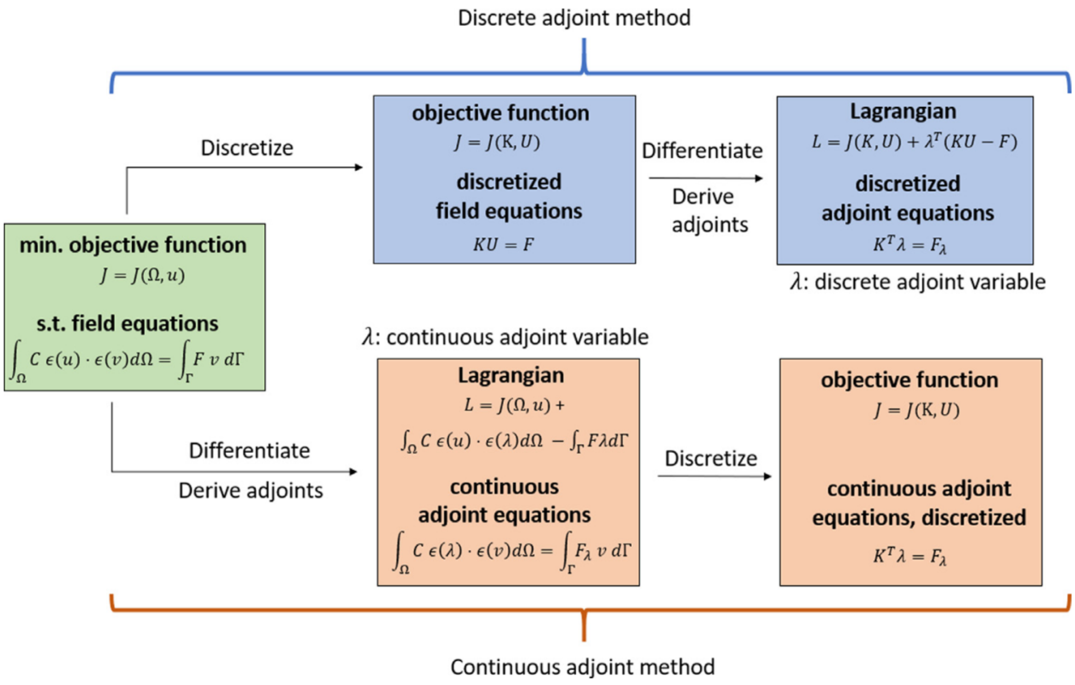

Discrete adjoint method
Discretize
objective function
$J = J(\mathrm{K}, U)$
discretized field equations
$KU = F$
Differentiate
Derive adjoints
Lagrangian
$L = J(K,U) + \lambda^T(KU - F)$
discretized adjoint equations
$K^T\lambda = F_\lambda$
λ: discrete adjoint variable
min. objective function
$J = J(\Omega, u)$
s.t. field equations
$\int_\Omega C\,\epsilon(u)\cdot\epsilon(v)d\Omega = \int_\Gamma F\,v\,d\Gamma$
λ: continuous adjoint variable
Differentiate
Derive adjoints
Lagrangian
$L = J(\Omega, u) +$
$\int_\Omega C\,\epsilon(u)\cdot\epsilon(\lambda)d\Omega - \int_\Gamma F\lambda d\Gamma$
continuous adjoint equations
$\int_\Omega C\,\epsilon(\lambda)\cdot\epsilon(v)d\Omega = \int_\Gamma F_\lambda\,v\,d\Gamma$
Discretize
objective function
$J = J(\mathrm{K}, U)$
continuous adjoint equations, discretized
$K^T\lambda = F_\lambda$
Continuous adjoint method

Figure 6.5: A schematic of the continuous and discrete adjoint methods. Linear elasticity equations example. (Kambampati et al., 2020)

.

In the classical level set topology optimization method used in this thesis, in which the boundary is updated by solving the Hamilton-Jacobi equation, the continuous adjoint is typically used. Indeed, up to this point in the thesis the continuous adjoint approach has been used which required a derivation of the shape derivative. However, as the physics become more complex, the derivation of such shape derivatives can become more challenging. Moreover, when employing numerical methods such as FEM or RKPM, what is actually being solved is the discretized governing equations. Thus, the exact cost function is the discretized form. Consequently, from the point of view of the discrete problem, the continuous adjoint sensitivities are only an approximation to the discrete gradients whereas the discrete adjoint provides the exact gradients. To quote Nadarajah (2003): "the continuous adjoint approach provides the inexact gradient to the exact cost function (discrete objective function)". Since the gradient is derived from the continuous equations and then computed numerically, it is not necessarily consistent with the discrete cost function. The discrete adjoint on the other hand provides the exact gradient to discrete cost function. Thus, the advantage of the discrete adjoint method is that the resulting gradient is exactly consistent with the discrete cost function, meaning that the optimizer correctly "sees" what we are solving for. If the discrete gradient is driven to zero, then a local optimum of the discrete cost is obtained whereas with the continuous adjoint even if the gradient is driven to zero the discrete objective function may not have converged to the discrete optimum. Of course, in the limit as the mesh size is reduced, both approaches yield the exact gradient of the continuous cost function. In addition, the discrete adjoint has the benefit that it can facilitate the employment of automatic differentiation (Mader and Martins, 2008).

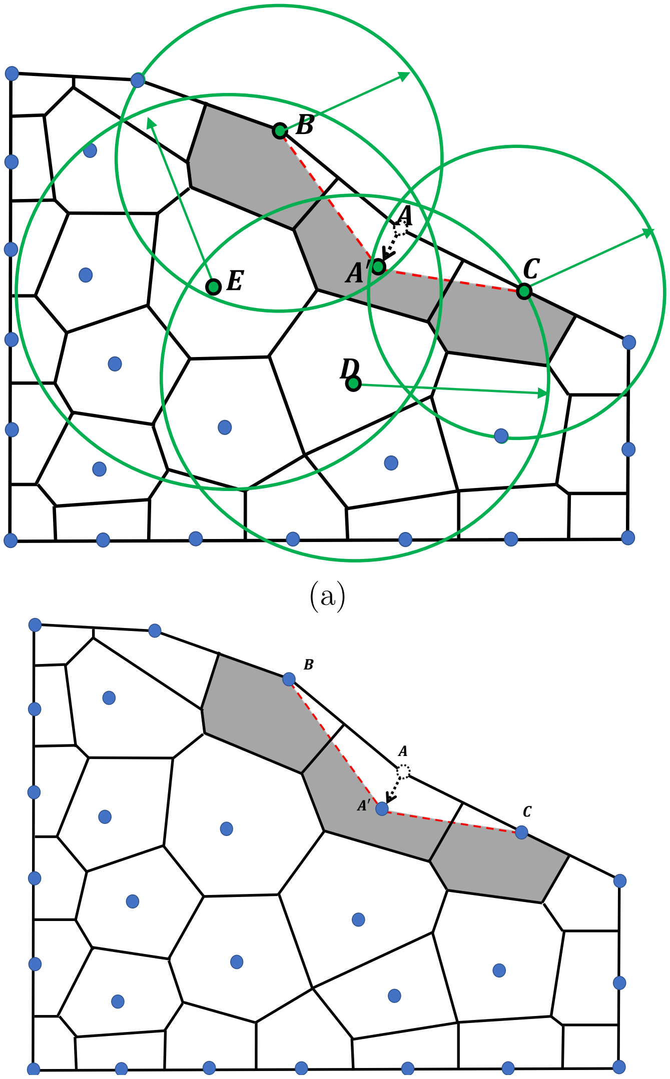
Since the discretized equations are solved for the entire domain, the discrete adjoint formulation results in volumetric sensitivities. For the discrete adjoint to be used in the classical level set method, the main challenge lies in connecting the volume sensitivities resulting from the discrete adjoint formulation to the structural shape change as a boundary moves. In the case of the fixed grid approach this is more straightforward since the mesh does not change due to boundary point movement. Rather, the value of the level set function changes. In their work based on XFEM-LSTO, Coffin and Maute (2016) used the chain rule to link the derivatives of the objective and constraints with respect to the nodal level set values to the derivatives of the nodal level set values with respect to the boundary movements. For the computation of the gradients, they employed the discrete adjoint in combination with finite difference sensitivity. More recently, Kambampati et al. (2020) presented an approach for coupling the elemental sensitivities to the level set boundary movement in fixed grid FEA-based LSTO with the Ersatz material approximation. In this case, the chain rule is also used to link the derivatives of the objectives with respect to the area fractions of the elements to the derivatives of the area fractions with respect to the boundary movement. A perturbation method is then used to evaluate the derivatives of the area fractions with respect to the boundary movement via finite difference.

However, in the case of a moving mesh scheme this becomes more complicated. This is because the mesh sensitivity that describes how the mesh, i.e., the coordi-nates of the interior nodes, deforms with respect to the boundary movement, needs to be computed. This problem is commonly encountered in shape optimiza-tion in computational fluid dynamics (CFD) (Nadarajah, 2003; Mader and Martins, 2008; Engels-Putzka and Frey, 2021; Secco and Martins, 2019; Patil et al., 2015; Nielsen and Park, 2006). The mesh sensitivity term is not straightforward to compute since grid regeneration, either locally or globally, is typically required for every movement of the surface. To the best of our knowledge, there are currently no LSTO approaches employing the discrete adjoint approach for unstructured meshes or particle-based analysis.

The particle-based approach described in Ch. 5, essentially is similar to the moving mesh scenario. However, since it does not employ a mesh to construct the approximation, the movement of the boundary has no effect on the position of the interior particles and so the mesh sensitivity term does not need to be computed. Taking advantage of this, a discrete adjoint approach in the context of the particle-based level set topology optimization described in Ch. 5 is presented in the next section.

6.3.1 Discrete adjoint for particle-based LSTO

In the level set topology optimization method the design variables are the boundary point movements. The sensitivity analysis thus computes the gradients of the objective and constraint functions with respect to these boundary point movements. In the LSTO-RKPM method with NSNI as described in Ch. 5, a mesh does not exist and when a boundary particle moves, only the coordinates of this specific particle change. The coordinates of all other particles are independent of this movement and do not change. Thus, differentiating the objective function by the paritcle's coordinates is equivalent to differentiating with respect to the boundary point movement. This means that changes happen only locally around the neighborhood of the boundary particle. This eliminates the need of computing the mesh sensitivity term that usually appears in FEA based shape optimization problems. Instead, in the context of the particle-based level set topology optimiza-tion, a deformation of the surface affects the shape functions of the particles present in the local region around the deformation. The RKPM shape function at a particle location is constructed based on the global coordinates of both the particle under consideration and the particles whose supports cover the particle. This is illustrated in Fig. 6.6 (a) in which as boundary particle $\boldsymbol{A}$ moves, the supports of particles $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$ and $\boldsymbol{E}$ that cover particle $\boldsymbol{A}$ need to change based on the new coordinates of $\boldsymbol{A}$. Moreover, in the NSNI-RKPM scheme described in section 5, the area, edge length, and edge normals of each Voronoi cell are used to compute the smoothed gradient matrix $\tilde{\mathbf{B}}$. The boundary point movement will change the Voronoi diagram locally as shown in Fig. 6.6 (b). As the boundary point $\boldsymbol{A}$ moves, the segments $\boldsymbol{AB}$ and $\boldsymbol{AC}$ change to $\boldsymbol{A'B}$ and $\boldsymbol{A'C}$ respectively. In turn, the Voronoi cells for particle $\boldsymbol{A}$ and its direct boundary neighbors $\boldsymbol{B}$ and $\boldsymbol{C}$, need to change. This is done based on the construction of the Voronoi diagram described in Sec. 5, where the boundary cells are cut based on their intersections with the boundary segments. Consequently, this also has an effect on the RKPM shape functions and the smoothed gradient matrix $\tilde{\mathbf{B}}$. To summarize, when a boundary particle moves, this will affect the shape functions around it due to the change in its coordinates and the change in the Voronoi diagram locally, which in turn will affect the objective function and constraints.



(a)

(b)

Figure 6.6: (a) Support domains affected due to boundary movement and (b) Local changes in Voronoi diagram due to boundary movement

To illustrate this idea, the simple case of compliance minimization in a linearly elastic structure is considered as an example. In the discrete adjoint method, the linear elasticity equation is first discretized as

$$\mathbf{Ku} = \mathbf{f} => \mathbf{R} = \mathbf{f} - \mathbf{Ku} = 0 \tag{6.30}$$

where $\mathbf{u}$ is the displacement vector and $\mathbf{f}$ is the force vector, $\mathbf{R}$ represents the residual and $\mathbf{K}$ is the stiffness matrix assembled using particle stiffness matrices as

$$\mathbf{K} = \sum_{i=1}^{NP} \mathbf{K}_i \tag{6.31}$$

where NP is the total number of particles. The discretized compliance function can the be formulated as

$$C = \mathbf{f}^{\mathbf{T}}\mathbf{u} \tag{6.32}$$

where C denotes the structural compliance. The total derivative of the objective function with respect to the change in coordinates of boundary point j can be expressed as

$$\frac{dC}{d\mathbf{x}_j} = \frac{\partial C}{\partial \mathbf{x}_j} + \frac{\partial C}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}_j} \tag{6.33}$$

where $\mathbf{x}_j$ denotes the particle coordinates. To treat the unknown implicit term $\frac{\partial \mathbf{u}}{\partial \mathbf{x}_j}$, the governing equation, i.e., Eq. 6.30 is also differentiated with respect to the particle coordinates as

$$\begin{aligned} \frac{d\mathbf{R}}{d\mathbf{x}_j} &= \frac{\partial \mathbf{R}}{\partial \mathbf{x}_j} + \frac{\partial \mathbf{R}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}_j} = 0 \\ => \frac{\partial \mathbf{u}}{\partial \mathbf{x}_j} &= -\left[\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\right]^{-1}\frac{\partial \mathbf{R}}{\partial \mathbf{x}_j} \end{aligned} \tag{6.34}$$

Substituting the result from Eq. 6.34 back into Eq. 6.33 the total sensitivity equation becomes

$$\frac{dC}{d\mathbf{x}_j} = \frac{\partial C}{\partial \mathbf{x}_j} - \frac{\partial C}{\partial \mathbf{u}}\left[\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\right]^{-1}\frac{\partial \mathbf{R}}{\partial \mathbf{x}_j} \tag{6.35}$$

From Eq. 6.35 one can either use the direct method which solves the system generated by the last two terms, or alternatively, the adjoint method can be used which solves the system of the second and third terms as

$$\boldsymbol{\lambda} = -\frac{\partial C}{\partial \mathbf{u}}\left[\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\right]^{-1} \tag{6.36}$$

or else

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\right]^{T}\boldsymbol{\lambda} = -\frac{\partial C}{\partial \mathbf{u}} \tag{6.37}$$

where $\boldsymbol{\lambda}$ is the vector of adjoint variables. The direct method requires solving one linear system for each design variable whereas with the adjoint method a linear system needs to be solved for each objective and constraint function. Thus for cases with large numbers of design variables, as is the case for topology optimization, the adjoint method is the preferred option.

For compliance, Eq. 6.37 takes the form

$$\left[\frac{\partial(\mathbf{f} - \mathbf{K}\mathbf{u})}{\partial \mathbf{u}}\right]^{T}\boldsymbol{\lambda} = -\frac{\partial(\mathbf{f}^{T}\mathbf{u})}{\partial \mathbf{u}} => \mathbf{K}\boldsymbol{\lambda} = \mathbf{f} \tag{6.38}$$

and thus for compliance $\boldsymbol{\lambda} = \mathbf{u}$. Using this result, Eq. 6.33 becomes

$$\begin{aligned}\frac{dC}{d\mathbf{x}_j} &= \frac{\partial(\mathbf{f}^{T}\mathbf{u})}{\partial \mathbf{x}_j} + \boldsymbol{\lambda}^{T}\frac{\partial(\mathbf{f} - \mathbf{K}\mathbf{u})}{\partial \mathbf{x}_j} \\ &= \mathbf{f}^{T}\frac{\partial \mathbf{u}}{\partial \mathbf{x}_j} - \boldsymbol{\lambda}^{T}\frac{\partial \mathbf{K}}{\partial \mathbf{x}_j}\mathbf{u} - \boldsymbol{\lambda}^{T}\mathbf{K}\frac{\partial \mathbf{u}}{\partial \mathbf{x}_j} \\ &= (\mathbf{f}^{T} - \boldsymbol{\lambda}^{T}\mathbf{K})\frac{\partial \mathbf{u}}{\partial \mathbf{x}_j} - \boldsymbol{\lambda}^{T}\frac{\partial \mathbf{K}}{\partial \mathbf{x}_j}\mathbf{u} \end{aligned} \tag{6.39}$$

if $\boldsymbol{\lambda}$ satisfies the adjoint Eq. 6.38 the the first term disappears and finally

$$\frac{dC}{d\mathbf{x}_j} = -\mathbf{u}^{T}\frac{\partial \mathbf{K}}{\partial \mathbf{x}_j}\mathbf{u} \tag{6.40}$$

Recall from Ch. 5 that the $\mathbf{K}$ matrix is defined as $\mathbf{K} = \tilde{\mathbf{B}}^{T}\mathbf{D}\tilde{\mathbf{B}}$ and so Eq. 6.41 can be written as

$$\begin{aligned}\frac{dC}{d\mathbf{x}_j} &= -\mathbf{u}^{T}\frac{\partial(\tilde{\mathbf{B}}^{T}\mathbf{D}\tilde{\mathbf{B}})}{\partial \mathbf{x}_j}\mathbf{u} \\ &= -\mathbf{u}^{T}2\tilde{\mathbf{B}}^{T}\mathbf{D}\frac{\partial \tilde{\mathbf{B}}}{\partial \mathbf{x}_j}\mathbf{u}\end{aligned} \tag{6.41}$$

Once again it is important to note, that when a boundary particle j moves, only the coordinates of this boundary particle change. As a result of this, only the particles whose supports cover the boundary particle will be affected, and thus Eq. 6.41 only needs to be computed for this small number of particles when computing the sensitivity value for boundary point j. In the example of Fig. 6.6 (a) these would be the particles $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$ and $\boldsymbol{E}$.

Computation of $\frac{\partial \tilde{\mathbf{B}}_i}{\partial \partial x_j}$

The term $\frac{\partial \tilde{\mathbf{B}}_i}{\partial \partial x_j}$ is computed by finite differencing. For a given boundary particle j of interest, a small perturbation δ is applied in the boundary normal direction as shown in Fig. 6.6 for particle $\boldsymbol{A}$. The perturbation magnitude is of the order of $\delta \sim 10^{-4} \times d_{min}$, where d_{min} is the distance to the closest neighbor. The term $\frac{\partial \tilde{\mathbf{B}}_i}{\partial \partial x_j}$ then may be approximated as follows

$$\frac{\partial \tilde{\mathbf{B}}_i}{\partial x_j} \approx \frac{\tilde{\mathbf{B}}_i - \tilde{\mathbf{B}}_i^{\delta}}{\delta} \tag{6.42}$$

where $\tilde{\mathbf{B}}_i^{\delta}$ is the smoothed gradient matrix of the ith particle after the perturbation of boundary particle j.

6.4 Free floating volumes

One of the challenges associated with topology optimization for fluid structure interaction problems is the emergence of free-floating volumes of solid material. These non-physical free-floating volumes that usually appear during the optimization process as material is being removed and rearranged. They are usually eliminated by optimization naturally and do not present challenges in structural optimization. This was also the case for the pressure loading problems considered in Chapters 4 and 5, since they tend to disappear quickly in the optimization procedure. However in the case of FSI problems, as the free-floating volumes are not connected to an anchor point, they undergo rigid-body motion within the flow field. Figure 6.7 illustrates the formation of such free floating volumes.

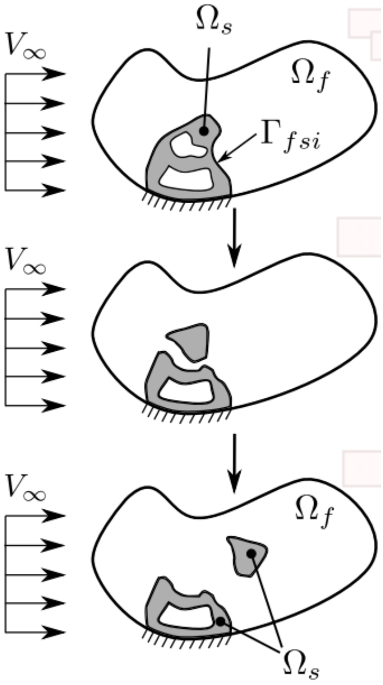


Figure 6.7: Appearance of free-floating volumes of solid material in FSI topology optimization (Jenkins and Maute, 2016)

.

Some works on FSI topology optimization do not report observing such features (Yoon, 2010; Picelli et al., 2015a) in their examples. On the other hand, several works in the literature have proposed methods for addressing the free-floating volumes problem. Jenkins and Maute (2016) proposed a scheme in which free-floating volumes are tracked prior to the FSI analysis, by computing an indicator field governed by the linear, isotropic diffusion equation. Solid material identified as free-floating solid is considered to be fluid in the subsequent FSI analysis. In a later work Lundgaard et al. (2018) presented an alternative to the indicator method for removing free floating islands of solid elements (FFIOSE) as called by the authors. In this work the islands are removed by combining different objective functions with different features and weights in a weighted multi-objective formu-lation. This approach requires a tuning of an additional parameter which is important to obtain an adequate ratio between the influence of the multiple objective functions.

For the examples considered in this thesis, free floating volumes naturally disappear without specific treatment. However, it was observed that if not treated they may take several iterations to disappear, which can slow down the convergence. For this reason, and to avoid potential problems when different objectives and constraint formulations are considered in the future, an approach is presented next for removing free-floating volumes or “islands” as referred to hereafter.

6.4.1 Algorithm for removing islands

The process of removing islands is based on a particle type list which is a list of integer indices to classify particles into different types according to their location, i.e., on the Dirichlet boundary, the main chain, inside the main chain or outside the main chain. The algorithm is divided into two parts. First, the boundary portion connected to the Dirichlet boundary, called the ”Main Chain”, is identified. Subsequently, all particles that are enclosed by the main chain are given a separate particle type. Any particles outside the main chain are identified as islands and are being removed in the next optimization step.

Main Chain

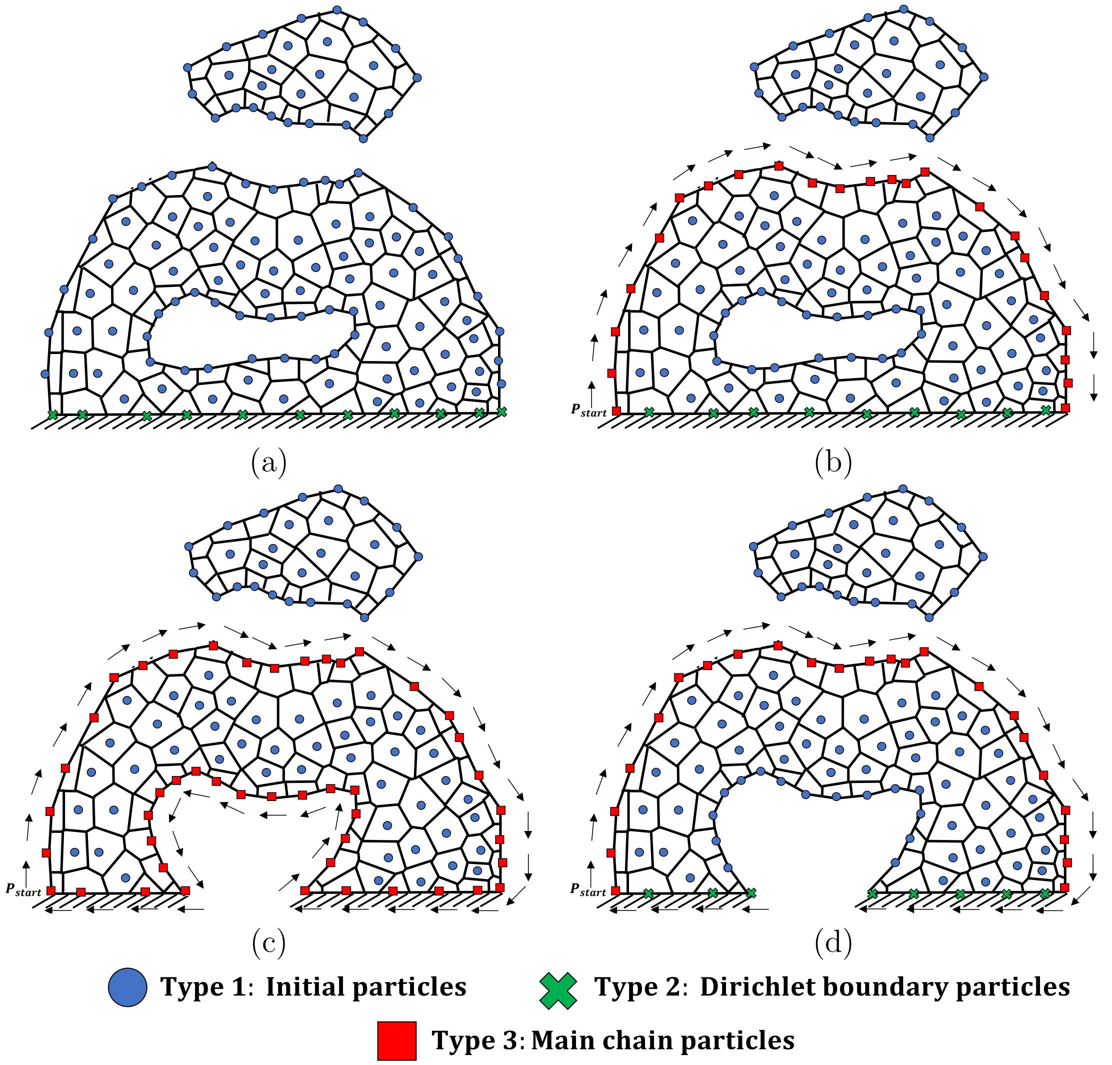
The process of identifying the main chain is illustrated in Fig. 6.8. Initially, all particles are assigned “type 1” except particles on the Dirichlet boundary which are assigned “type 2”, as shown by the blue circles and the green “x” symbols, respectively, in Fig. 6.8 (a). Starting from the first point on the Dirichlet boundary indicated as P_{start} in Fig. 6.8 (b), the main chain is created by moving from one boundary point neighbor with “type 1” to the next and turning them into “type 3”, as indicated by the arrows in the figure. The process continues until finally a Dirichlet boundary particle with “type 2” is encountered, in which case the main chain stops as shown by the end of the arrow line in Fig. 6.8 (b). Figure 6.8 (c), shows what would happen if the Dirichlet boundary particles were not assigned a separate type. In this case, the main chain extends into the interior hole. As a result, an interior island, i.e., an island within the void region, would be removed. However, only islands in contact with the fluid are problematic because they undergo rigid body motion in the fluid domain. Thus, the Drichlet particles are assigned a separate type to achieve the configuration in Fig. 6.8 (d), in which the main chain does not extend to the interior holes. The boundary neighbors are stored during the level set function computation and this makes it possible to move from neighbor to neighbor without having to loop through all particles each time, which would be computationally expensive. Thus, to create the main chain it is only required to loop through as many particles as the number of particles on the main chain. Since this is only a small number compared to the total number of particles, the algorithm is not computationally demanding.

Loading Surface

It is also important to note, that the main chain acts as the loading surface on which the FSI loads are applied. This way FSI load is not transferred to the load-free portion of the boundary such as the Dirichlet boundary or the interior holes of the structures.

Particles Enclosed by the Main Chain

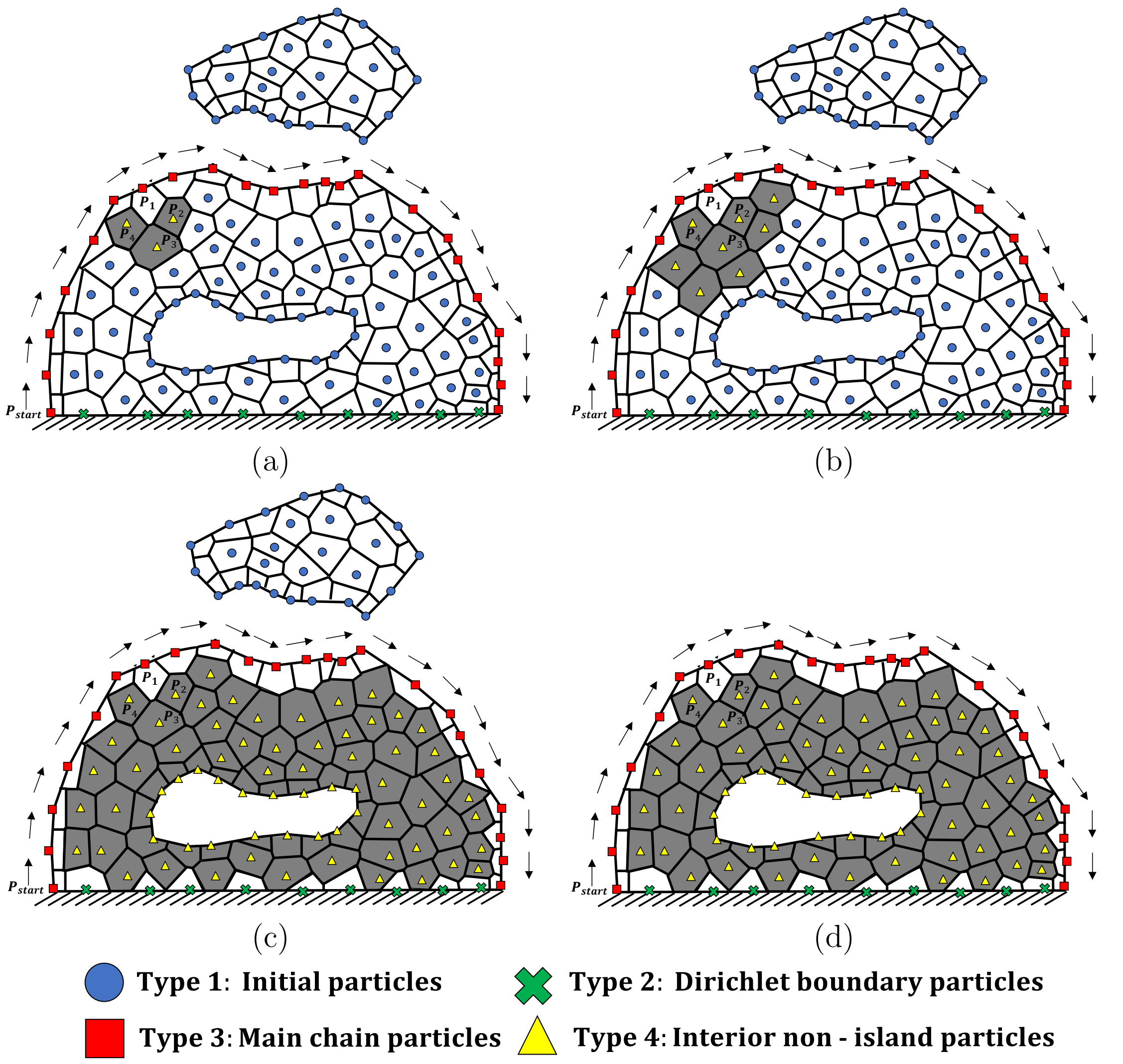
Once the main chain has been identified, the second part of the island identification process begins. At this stage, interior (i.e. non-boundary) particles that are enclosed by the main chain are assigned a separate “type 4”, to distinguish them from island particles. This process is described in Fig. 6.9. Starting from any particle on the main chain such as P_1, its interior neighbors are identified and turned into “type 4”. This is shown for particles P_2, P_3 and P_4 with the yellow triangles in Fig. 6.9 (a). As explained in Chapter 5, neighbors for each particle are stored during the construction of the Voronoi diagram and this information is directly accessible at each iteration. Next, the direct neighbors of P_2, P_3 and P_4 are also turned into “type 4” as shown in Fig. 6.9 (b). Moving from neighbor to neighbor eventually all particles inside the main chain are turned into “type 4”, whereas for islands both boundary and interior particles remain “type 1”, as Fig. 6.9 (c) illustrates. In the end, as can be seen in Fig. 6.9 (d), all “type 1” particles are identified as islands and are removed completely from the next optimization iteration.



P_{start}
P_{start}
P_{start}
● Type 1: Initial particles
✖ Type 2: Dirichlet boundary particles
🟥 Type 3: Main chain particles

(a) (b) (c) (d)

Figure 6.8: Main chain identification: (a) Setting initial type 0 for all particles and setting Dirichlet boundary particles as type 2, (b) Creating main chain by moving from one boundary neighbor to the next, (c) Illustration of problematic situation if Dirichlet particles were not given a separate type, (d) Illustration of how a separate type for Dirichlet particles avoids situation in (c) (Bendse and Sigmund, 2004)


P_1
P_2
P_3
P_4
P_{start}
Type 1: Initial particles
Type 2: Dirichlet boundary particles
Type 3: Main chain particles
Type 4: Interior non - island particles

(a)

(b)

(c)

(d)

Figure 6.9: Island identification: (a) Starting from a random point, P_1 on the main chain and turning its neighbors P_2, P_3, P_4 into "type 4", (b) Continuing the process by turning the neighbors of P_2, P_3, P_4 into "type 4", (c) Repeating the process until all particles within the main chain are turned into "type 4" and (d) In the end of the process all remaining "type 1" particles are identified as islands and removed from the computation

6.5 Compliance minimization for steady-state FSI

In this section the compliance minimization of linear elastic structures subjected to steady-state FSI is considered. This one way coupling is a good starting point to compare with the literature and validate the approach.

6.5.1 Particle discrete adjoint for steady state fluid-structure interactions

Section 6.1 illustrated the capability of the LSTO-RKPM-mIFEM method in analyzing transient FSI problems. Although there is no theoretical limitation in extending this scheme to transient FSI topology optimization problems, there are no known solutions or previous literature to compare with. And so, a simplified case of steady-state FSI problems is considered in this thesis to validate our approach by comparing with results from the literature.

To consider steady-state conditions, the time dependent terms in the equations of Sec. 6.1 are disregarded. Under this assumption, the fluid domain is discretized with the finite element method and the finite element matrices for the incompressible steady-state Navier-Stokes equations are given by (Sanches, 2015; Zienkiewicz et al., 2014).

$$\begin{bmatrix} \mathbf{K}_f + \mathbf{K}_T(\mathbf{v}_f) & -\mathbf{Q} \\ -\mathbf{Q} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{v}_f \\ \mathbf{P}_f \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \end{Bmatrix} \tag{6.43}$$

where,

$$\mathbf{K}_f = \mu \int_{\Omega_{fe}} \mathbf{B}_{fe}^T \mathbf{I_0} \mathbf{B}_{fe} d\Omega_{fe} \tag{6.44}$$

$$\mathbf{K}_T(\mathbf{v}_f) = \rho_f \int_{\Omega_{fe}} \mathbf{N}_\upsilon^T \mathbf{N}_\upsilon (\mathbf{v}_x \nabla \mathbf{N}_\upsilon^x + \mathbf{v}_y \nabla \mathbf{N}_\upsilon^y) d\Omega_{fe} \tag{6.45}$$

$$\mathbf{Q} = \int_{\Omega_{fe}} \nabla \mathbf{N}_\nu^T \mathbf{N}_P d\Omega_{fe} \tag{6.46}$$

where the matrices $\mathbf{N}$ contain the shape functions for velocity and pressure as indicated by the subscripts ν and P subscripts, respectively. The matrix $\mathbf{B}_{fe}$ contains the gradients of the finite element shape functions and Ω_{fe} represents the area of the finite element. The matrix $\mathbf{I_0}$ is the identity matrix. Matrices $\mathbf{K}_f$ and $\mathbf{K}_T(\mathbf{v}_f)$ include the diffusive and inertia terms, respectively.

Compliance Minimization under FSI loads

For the problems considered in this chapter, compliance minimization of linear elastic structures under FSI loads is the main objective and this can be formulated in discretized form as

$$\begin{aligned} \textbf{min} \quad & C(\mathbf{x}_j) = \mathbf{u}^T\mathbf{K}\mathbf{u} \\ \textbf{s.t.} \quad & \mathbf{K}\mathbf{U} = \mathbf{f}_{fsi} \\ & V(\mathbf{x}_j)/V_0 = \bar{V} \end{aligned} \tag{6.47}$$

whereas $\mathbf{u}$ and $\mathbf{K}$ are the solid displacement and solid stiffness matrix in turn, $\bar{V}$ is the volume constraint limit, and the right hand side $\mathbf{f}_{fsi}$ is the fluid traction applied on the solid at the FSI interface given by

$$\mathbf{f}_{fsi,I} = \int_{\Gamma_{fsi}} \boldsymbol{\Psi}_I[\mathbf{n}(\mathbf{Q}\mathbf{P}_f - (\mathbf{K}_f + \mathbf{K}_T(\mathbf{v}_f)]\mathbf{v}_f)_i) \tag{6.48}$$

where Γ_{fsi} is the FSI interface and the index i denotes the i_{th} fluid element intersected by the FSI interface and from which the fluid quantities are interpolated onto the interface. Similar to Eq. 6.39 in Sec. 6.3.1, the total derivative of the compliance with respect to coordinates is given by

$$\frac{dC}{d\mathbf{x}_j} = \frac{\partial(\mathbf{f}_{fsi}^T\mathbf{u})}{\partial\mathbf{x}_j} + \boldsymbol{\lambda}^T\frac{\partial(\mathbf{f}_{fsi} - \mathbf{K}\mathbf{u})}{\partial\mathbf{x}_j} \tag{6.49}$$

whereas as before $\boldsymbol{\lambda} = \mathbf{u}$. The difference now is that the load is the fluid traction resulting from the FSI analysis as shown in Eq. 6.47. Then, Eq. 6.49 can be written as

$$\begin{aligned} \frac{dC}{d\mathbf{x}_j} &= \frac{\partial(\mathbf{f}_{fsi}^T\mathbf{u})}{\partial\mathbf{x}_j} + \boldsymbol{\lambda}^T\frac{\partial}{\partial\mathbf{x}_j}\left\{\boldsymbol{\Psi}[\mathbf{n}((\mathbf{Q}\mathbf{P}_f - (\mathbf{K}_f + \mathbf{K}_T(\mathbf{v}_f))\mathbf{v}_f) - \mathbf{K}\mathbf{u}]\right\} \\ &= \mathbf{f}_{fsi}^T\frac{\partial\mathbf{u}}{\partial\mathbf{x}_j} + \boldsymbol{\lambda}^T\frac{\partial(\boldsymbol{\Psi}[\mathbf{n}\mathbf{Q}\mathbf{P}_f])}{\partial\mathbf{x}_j} - \boldsymbol{\lambda}^T\frac{\partial\mathbf{K}}{\partial\mathbf{x}_j}\mathbf{u} - \boldsymbol{\lambda}^T\mathbf{K}\frac{\partial\mathbf{u}}{\partial\mathbf{x}_j} \\ &= (\mathbf{f}_{fsi}^T - \boldsymbol{\lambda}^T\mathbf{K})\frac{\partial\mathbf{u}}{\partial\mathbf{x}_j} + \boldsymbol{\lambda}^T\frac{\partial(\boldsymbol{\Psi}[\mathbf{n}\mathbf{Q}\mathbf{P}_f])}{\partial\mathbf{x}_j} - \boldsymbol{\lambda}^T\frac{\partial\mathbf{K}}{\partial\mathbf{x}_j}\mathbf{u} \end{aligned} \tag{6.50}$$

Note that all the terms multiplied by $\mathbf{v}_f$ vanish since $\mathbf{v}_f$ is zero within the solid and on the FSI interface. If $\boldsymbol{\lambda}$ satisfies the adjoint equation 6.38 then the first term disappears and Eq. 6.50 results in

$$\frac{dC}{d\mathbf{x}_j} = -\mathbf{u}^T 2\tilde{\mathbf{B}}^T\mathbf{D}\frac{\partial\tilde{\mathbf{B}}}{\partial\mathbf{x}_j}\mathbf{u} + \mathbf{u}^T\frac{\partial(\boldsymbol{\Psi}[\mathbf{n}\mathbf{Q}\mathbf{P}_f])}{\partial\mathbf{x}_j} \tag{6.51}$$

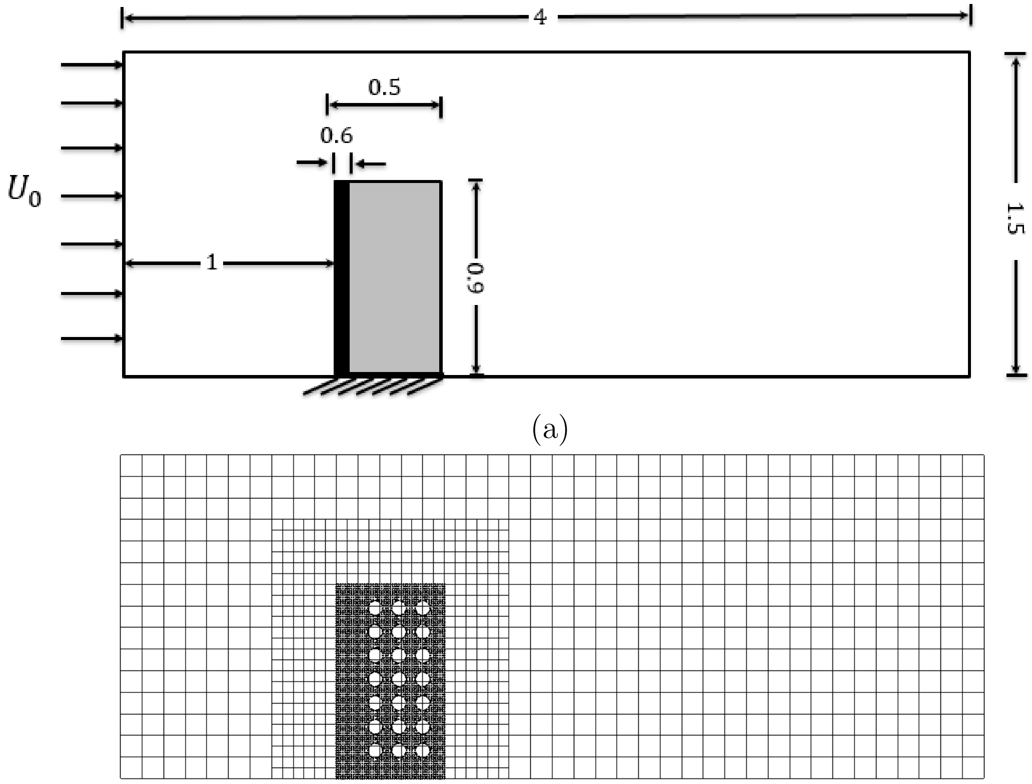
where the term $\frac{\partial(\boldsymbol{\Psi}[\mathbf{n}\mathbf{Q}\mathbf{P}_f])}{\partial\mathbf{x}_j}$ expresses the variation of the pressure load applied at the FSI interface due to the boundary movement and it is computed via finite differencing similar to $\frac{\partial\tilde{\mathbf{B}}_i}{\partial x_j}$.

6.5.2 Leaflet example

The most encountered example in the literature of FSI topology optimization is the leaflet example discussed in Sec. 6.2. Indeed, this example is considered by every FSI topology optimization work for low to moderate Reynolds numbers: Yoon (2010) for Re = 5, Picelli et al. (2015a) for Re = 100, Re = 1000, Jenkins and Maute (2016) for Re = 10, Lundgaard et al. (2018) for Re = 100, Feppon et al. (2019) for Re = 60, Picelli et al. (2020) for Re = 0.01, Re = 100. It is thus a good test case for comparison. As shown in Fig. 6.10 (a), this consists of a rectangular beam placed inside a viscous fluid field, with fluid flowing from left to right with a prescribed velocity, which here is set to $U_0 = 0.1m/s$. The fluid density is set to $\rho^f = 1000kg/m^3$. The Reynolds number is computed from the height of the beam as $Re = \frac{\rho^f \cdot h \cdot U_0}{\mu}$, where h is the height of the beam and μ is the dynamic viscosity. Two cases for dynamic viscosity are tested, one with a value of $\mu = 60(kg/m \cdot s)$ which gives a Reynolds number $Re = 0.01$ and one with $\mu = 0.0006(kg/m \cdot s)$ for which $Re = 1000$. The results of the two different Reynolds number cases are compared to investigate the effect of the flow behavior on the optimum topologies. The top and bottom boundaries are modeled as no-slip wall. The beam is fixed at the bottom edge, and it is assumed to be linear elastic with a Young's modulus value equal to 10 kPa and Poisson's ratio equal to 0.3. The black rectangular region on the left side of the beam is considered as a non-design domain. The initial fluid and solid discretizations are shown in Fig. 6.10 (b). As can be seen, the fluid mesh is refined near the solid while the initial particle distribution with holes sits on top of the fluid. The objective is to minimize the compliance under a volume constraint of 40%.

The optimization results are provided in Fig. 6.11 along with the velocity fields for the two Reynolds number cases. The obtained solutions appear to be different for the different Reynolds number cases. Specifically, the outer FSI interface on the right side of the structure is more rounded, or convex, in the case of the low Reynolds number flow, whereas for the high Reynolds number the interface becomes flatter, nearly concave. This difference can be explained by examining the pressure fields produced in Fig. 6.12. In Fig. 6.12 (a), the pressure acting on the exterior FSI surface is positive, meaning that it points towards the interior of the structure. This compressive load forces the interface to become more rounded, since as discussed in Ch. 4 such shapes are optimum in carrying positive pressure. On the other hand, pressure is negative in that region for the high Reynolds number case. This means that the interface is being pulled outwards and as a result it becomes concave to resist this load. A similar observation was reported by Picelli et al. (2020), which is the only FSI work that considered such comparison with different Reynolds numbers.

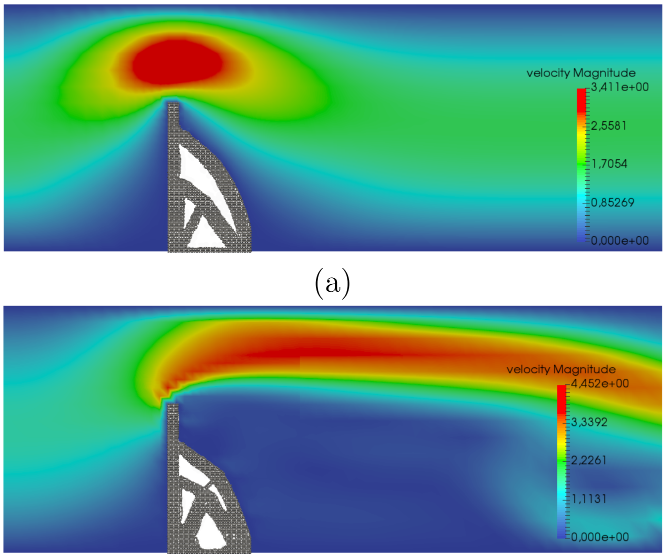
Figure 6.13 provides the results obtained by other authors in the literature for the same example for different Reynolds numbers. It seems that the result obtained by LSTO-RKPM-mIFEM bears similarities with these previous solutions. Especially the solution for Re = 1000 is closer to the results obtained by Jenkins and Maute (2016) using LSTO and XFEM and considering $Re = 10$ and the results obtained by Picelli et al. (2020) using the TOBS method with COMSOL considering $Re = 0.01$ and $Re = 100$.



(a)

(b)

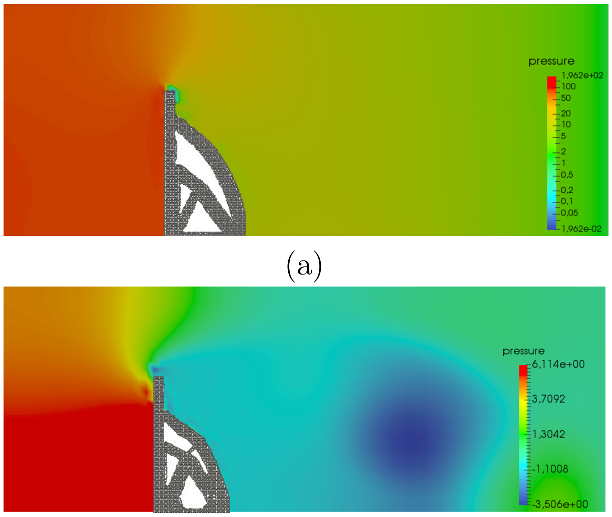
Figure 6.10: Topology optimization of the leaflet example: (a) Setup (dimensions are in *cm*) and (b) Initial particle distribution and fluid mesh with refinement near the solid



(a)

(b)

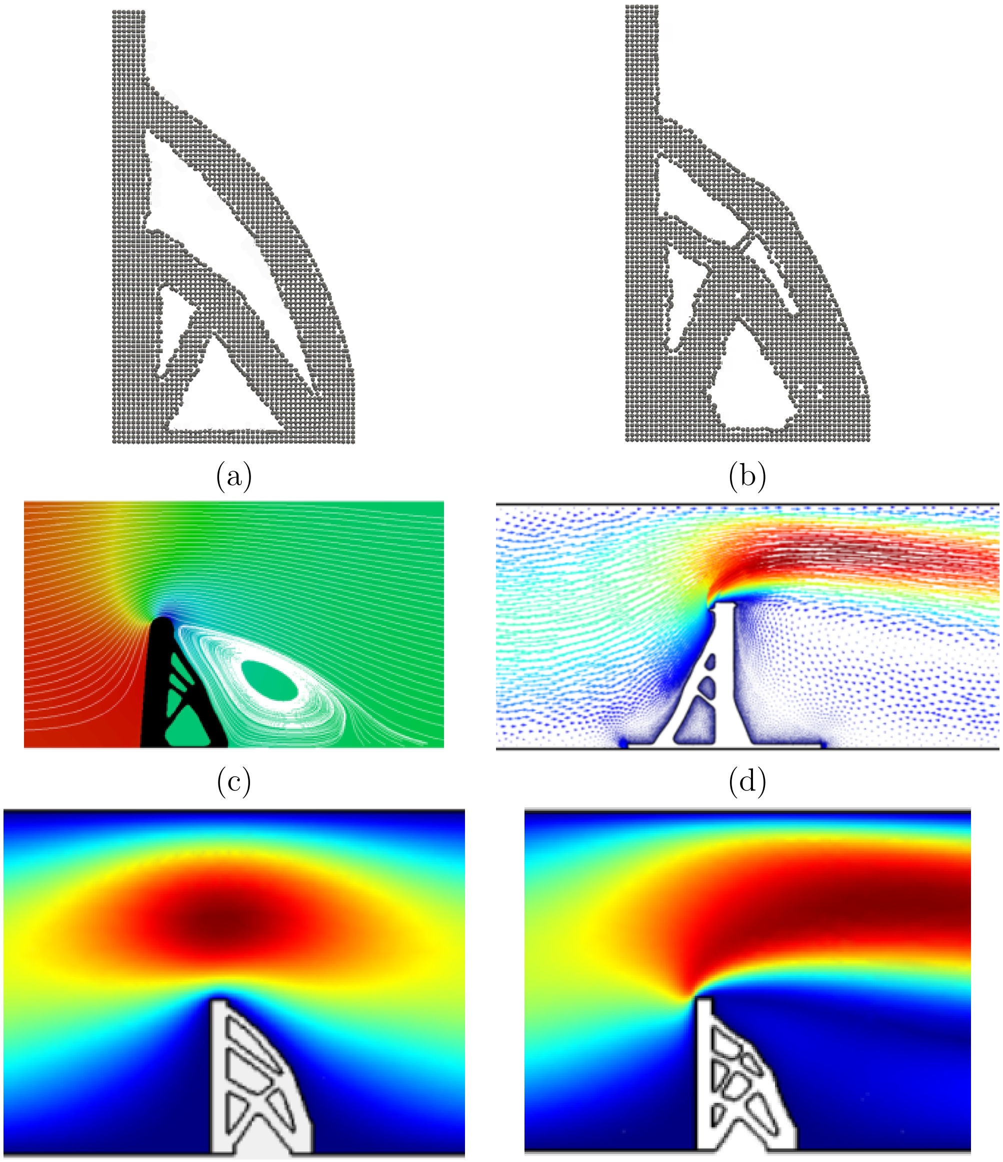
Figure 6.11: Optimum solutions and velocity fields for: (a) Re = 0.01 and (b) Re = 1000



(a)

(b)

Figure 6.12: Optimum solutions and pressure fields for: (a) Re = 0.01 and (b) Re = 1000



(a) (b)

(c) (d)

(e) (f)

Figure 6.13: Comparison with the literature: (a) LSTO-RKPM solution for Re = 0.01, (b) LSTO-RKPM solution for Re = 1000, (c) Jenkins and Maute (2016) solution for Re = 10 using LSTO and XFEM, (d) Feppon et al. (2019) solution for Re = 60 using LSTO and remeshing, (e) Picelli et al. (2020) solution for Re = 0.01 and (f) Picelli et al. (2020) solution for Re = 100 using TOBS and COMSOL

6.6 Conclusions

In this chapter the LSTO-RKPM methodology was extended to FSI problems. The modified immersed finite element method is used to couple the separate solid and fluid governing equations. This LSTO-RKPM-mIFEM combination has certain advantages. First, it allows for modularity in the solvers and even different methods of discretization for the solid and the fluid domains. This was illustrated in this chapter by employing the RKPM for the solid and the FEM for the fluid. Moreover, the use of a fixed fluid grid allows for efficiently solving the coupled equations without the need for remeshing, and as was demonstrated through the example adaptive refinement can also be used. The LSTO-RKPM combination for the solid can handle straightforwardly the topological changes while providing a well-defined structural boundary. For the sensitivity analysis, an approach was presented that allows for the discrete adjoint method to be used in the context of LSTO-RKPM based on a boundary particle perturbation scheme. Additionally, an algorithm was developed to identify and remove islands of solid material, which is a well-known challenge in FSI topology optimization. For validation through comparison with the literature, the well-known leaflet example was solved for compliance minimization under a volume constraint, assuming steady state conditions. It can be seen from the results, that they are in good agreement with the results obtained by other authors in the literature. The assumption of steady state simplifies the physics and leads to essentially a one-way coupling. Since this is currently the state of the art in FSI topology optimization it is a good starting point to compare against benchmarking problems in the existing literature and validate the approach. Nevertheless, as it was shown in this chapter, the methodology can be readily extended to time dependent problems. Specifically, this was illustrated through the validation of the transient RKPM solver and the transient FSI analysis for the leaflet example in section 6.2.

Chapter 7

Conclusions

In this final chapter some concluding remarks are discussed. The main contributions of this thesis are outlined along with the methodologies developed for the different problems. Suggestions for future research are also provided.

7.1 Summary and main contributions

This thesis is focused on a specific class of challenging problems in topology optimization known as design-dependent surface problems. This includes multi-physics problems that are governed by the interfaces. The main challenge lies in tracking the interface for applying coupling or loading conditions. To develop a general methodology for solving such problems irrespectively of the type of physics, the simple case of purely hydrostatic pressure was considered first. This is the situation when a structure is submerged in a static fluid. When using topology optimization to design such a structure, throughout optimization the pressure surface changes and thus the position and direction of the loads is altered to follow this loading surface.

To tackle the problem, a meshfree level set topology optimization approach was developed in Ch. 4 using the reproducing kernel particle method and Gauss integration to solve the governing equations and compute sensitivities. The attractiveness of this combination lies in the straightforward way it can handle the design dependency without any loading interpolation schemes. The crisp boundary representation provided by the level set method, is maintained onto the computational domain by placing RKPM particles along the structural boundary without requiring remeshing. This allows for direct application of the loads and it also helps to compute the sensitivities at the boundary points without requiring additional approximations such as stress recovery techniques as in fixed grid FEA. The results obtained by the LSTO-RKPM methodology are in good agreement with the results presented by other authors for the same examples, using load interpolation schemes or remeshing. Furthermore, the convergence rates have been shown to be superior to a fixed-grid FEA LSTO approach with a load transformation scheme.

Validating the LSTO-RKPM methodology for hydrostatic pressure opened up the road for an extension to more complex design-dependent physics such as FSI, which is the main interest in this thesis. Before moving on to FSI, an alternative LSTO-RKPM implementation based on nodal integration was developed in Ch. 5 with the aim of improving certain limitations observed for the LSTO-RKPM with Gauss integration in Ch. 4. Specifically, improving the computational efficiency and exploring different particle distributions were the main objectives. As shown, not only is the nodally integrated approach much more efficient, it is also capable of achieving higher accuracy than the Gauss integration. Furthermore, a scheme was presented for storing information such that computations between iterations are only required for particles in a narrow band around the boundary. This significantly improved efficiency for a regular distribution of particles. Different particle distributions were also explored, illustrating another useful feature of the methodology. Specifically, since in level set topology optimization the structure evolves based on sensitivities computed at the boundary points, accuracy is more important near the boundary. Thus, it makes sense to have a denser particle distribution in a narrow band around the boundary while keeping the distribution sparse in the interior. The effectiveness of such a particle placement scheme was illustrated through stress-based examples, in which good convergence rates were achieved. Moreover, the additional benefit of a higher order approximation in the RK shape functions was explored. As the order of continuity can be controlled without additional complexity, simply by choosing the appropriate kernel function, different continuity orders were investigated. It was shown that as the order of continuity increases, the convergence rate for stress-based problems also improves. Lastly, a design-dependent example previously solved with the Gauss integration approach was considered again for comparison, showing that the nodally integrated approach gave the same result only 20 times faster. One last thing to note here is that although the construction of a Voronoi diagram may seem like remeshing, unlike remeshing in FEA, RPKM is not sensitive to the shape of the Voronoi cells. It should be emphasized that no effort was needed in regularizing the shape of the cells in any way. The only requirement for the accuracy is that the cells conform such that they satisfy the integration constraints as explained in Ch. 5.

In Ch. 6, the nodally integrated LSTO-RKPM methodology was used to tackle the more complex design-dependent FSI topology optimization problems. For this purpose, the modified immersed finite element method was combined with LSTO-RKPM to solve the FSI problems in an effective and efficient way. The LSTO-RKPM-mIFEM combination has several benefits starting from the capability of modularity in the solvers. This means that different solvers can be used for the solid and fluid domains, and even different discretizations. This was illustrated by employing FEM to solve the fluid equations and RKPM to solve the solid governing equations. Furthermore, the geometrical description of the solid using the level set method and the maintenance of the crisp boundary representation on the computational model by RKPM, provides a clearly identified FSI interface. The Eulerian fluid mesh allows for efficiently solving the coupled equations using the artificial fluid region, thus avoiding tedious remeshing for the fluid. A transient analysis for a fluid in contact with a Neo-Hookean solid was performed to illustrate the capability of the method to analyze transient FSI problems. However, for topology optimization FSI, the simplified case of steady-state problems was considered for validation and comparison with the literature, since currently there are no other works considering transient FSI topology optimization. Transient problems are however the natural next step to this research, and this is one of the main suggestions for future work. For sensitivity computation in a multiphysics environment, a particle-based discrete adjoint approach was presented which bridges the gap between volumetric discrete sensitivities and shape deformation. Specifically, boundary particle perturbations were applied to compute the effect of the boundary movement on the volume sensitivities using the finite difference method. The derivations for the compliance and steady-state FSI were provided and the effectiveness of this approach was illustrated through the examples. Another useful implementation presented was the methodology for identifying and eliminating free floating volumes of solid materials, which is a challenge specific to topology optimization of FSI problems. This approach was developed based on neighbor information provided from the Voronoi diagram which was used to perform the nodal integration.

Overall, the main contributions of the thesis can be summarized in accordance with Sec. 1.7 as follows:

1. Development of meshfree LSTO-RKPM methodologies. Implementations with different integration schemes and boundary conditions. Illustration of the useful features of nodally integrated LSTO-RKPM such as the controllable order of continuity of the RK shape functions, different particle distributions and efficiency through stress-based examples.

2. LSTO-RKPM implementation for addressing design-dependent problems in topology optimization. This has been illustrated through benchmarking hydrostatic pressure examples. The results were validated by comparison with the literature.

3. The methodology was extended to the more complex case of FSI by coupling the LSTO-RKPM methodology with mIFEM. The LSTO-RKPM-mIFEM methodology for addressing FSI topology optimization problems is the most significant contribution of the thesis. This is because it brings together all the ingredients developed throughout the thesis to solve this complex class of problems. To address FSI topology optimization problems, several additional ingredients have been developed which may also be considered as contributions:

    - A particle-based discrete adjoint sensitivity methodology for meshfree LSTO.

  - A new neighbor-based algorithm for identifying and isolating free-floating volumes of solid material into the fluid domain.

4. A benchmarking steady state FSI topology optimization problem was investigated and the results were compared with the literature for validation.

Overall, the objectives defined in section 1.7 are achieved. Objective 4 in Sec. 1.7 is not entirely met. This is because in the end, more complex transient FSI problems were not solved in the thesis. Nevertheless, the methodology presented in Ch. 6 is directly extensible to such problems. It comes down to identifying a meaningful transient FSI problem and performing the difficult task of computing the sensitivities for the transient problem. Further discussion and suggestions on how to extend this work to transient FSI is provided in the next section.

7.2 Suggestions for further research

Based on the conclusions and the main contributions of the thesis, several suggestions for future extensions of this research are provided below.

- Large Scale Problems.

  The particle scheme employed in Ch. 5 where a denser particle distribution is used in a narrow band close to the boundary while keeping the distribution sparce in the interior based on a quadtree particle distribution, inclines towards consideration of large scale problems. Indeed, through large scale problems the true benefit of such a scheme would be illustrated. Due to the higher order approximation and domain overlap of the RK shape functions, the method can afford to keep a sparse particle distribution in the interior while providing sufficient accuracy as illustrated through the stress-based problems in Ch. 5. This means that the number of particles in large sale problems can be greatly reduced, thus potentially improving computational efficiency significantly compared to a denser finite element distribution.

  Particularly, the combination of LSTO-RKPM with the spatially adaptive and temporally dynamic Volumetric Dynamic B+ (VDB) tree data structure (Museth, 2017), open sourced as OpenVDB, would be very interesting. OpenVDB, is tailored to minimize the computational cost and memory footprint on three-dimensional grids by avoid carrying high fidelity data outside the narrow band. In fact, the level set used in this thesis has already been combined with OpenVDB by Kambampati et al. (2019) for large scale elasticity and heat conduction problems using the Ersatz material approximation. As shown in that study, although VDB significantly im-proves the efficiency of the level set operations, the FEA remains the computational bottleneck taking 90-95% of the total time. The particle-based approach on the other hand could more naturally follow the data structures in VDB to reap its benefits also for the analysis part of the computation.

- Adaptive particle refinement

  Instead of placing particles based on a quadtree particle distribution, more sophisticated schemes can be explored in future works. For example, an adaptive refinement based on error estimators can be used to create particle distributions strategically. A question to consider is whether it would be more beneficial to perform several steps of refinement within each optimization iteration or to refine the particles in the next iteration based on error estimations form the previous iteration.

- Transient FSI

  A natural extension of this work is the consideration of high fidelity transient FSI topology optimization problems. This implies that the fluid is in contact with highly deformable solids. In this case, not only the topology changes throughout optimization, but the current topology also needs to undergo FSI deformations in a number of time steps within each optimization iteration. Such complex problem not only drastically increases the computa-tional cost but also adds challenges to the computation of sensitivities. Thus, future research directions can go two ways, one in developing more efficient implementations and the other in formulating meaningful transient FSI topology optimization problems that justify the need of such a highly complex model. The identification of existing challenges in applications governed by transient FSI, may assist in formulating interesting and useful optimization problems.

- Other design-dependent multiphysics

  In this thesis, a general methodology for addressing design-dependent physics was developed. The proposed LSTO-RKPM approach can be further extended to other types of design-dependent multiphysics problems such as conjugate heat transfer in which convection or radiation conditions need to be applied at the interface or acoustic-structure interactions and electromag-netics. In general, any type of multiphysics that requires imposition of interface conditions or loading, can be considered using the methodologies developed here.

- Nonlinearities and large deformations

  The LSTO-RKPM implementation has been shown in this thesis to possess benefits that go beyond the design-dependent problems. As explained in the introduction and literature review sections, RKPM has been developed to tackle complex problems in which conventional FEA struggles. Thus, the application of the LSTO-RKPM developed in this thesis to tackle problems such as material and geometric nonlinearities including contact and crashworthiness problems, would be an interesting extension of this research. In this context, alternative integration techniques that do not require conforming cells such as the stabilized non-conforming nodal integra-tion (SNNI) (Chen et al., 2017b) may be explored for use in topology optimization, to study problems with severe deformations or even fragmen-tations.

- Automatic differentiation (AD) and complex step method

  In the particle-based discrete adjoint scheme described in Ch. 6, the finite difference method was used to compute the partial derivatives of the shape functions with respect to the boundary point coordinates. For the simplified problems considered in Ch. 6 this approach provided good results. However, the sensitivity of the finite difference method on the step size, can cause problems with accuracy. An alternative method for computing partial derivatives is the complex step method that computes derivatives of real functions using complex variables (Martins et al., 2003). Like the finite-difference method, the complex step derivative approximation is derived from the Taylor series expansion. However, in the complex step method the only source of numerical error is the truncation error, and by decreasing the step size to a small enough value, one can ensure that the truncation error is of the same order as the numerical precision of the evaluation of the function at hand. Thus, the complex step method is superior to the finite difference method in terms of accuracy.

  Furthermore, in more complex multiphysics problems in which an analytical derivation of the partial derivatives in the adjoint equations is difficult or impossible, automatic differentiation (AD) can be useful. The idea is to use a hybrid approach similar to the one developed by Mader and Martins (2008), where the discrete adjoint formulation is combined with AD to take advantage of the benefits of each approach. Namely, to exploit the efficiency of the adjoint formulation for computing the total derivatives of the objectives with respect to the design variables and the accuracy of AD in computing the partial derivatives.

Overall, the discussion and suggestions above aim to illustrate that the work presented in this thesis provides a good basis for extension to various research directions.

Bibliography

Ai L., Gao X. L. (2019) Topology optimization of 2-D mechanical metamaterials using a parametric level set method combined with a meshfree algorithm. Composite Structures 229:111,318

Alexandersen J., Andreasen C. (2020) A review of topology optimisation for fluid-based problems. Fluids 5(1):29

Allaire G., Jouve F., Toader A. M. (2004) Structural optimization using sensitivity analysis and a level-set method. Journal of Computational Physics 194:363–393

Allaire G., Dapogny C., Frey P. (2014) Shape optimization with a level set based mesh evolution method. Computer Methods in Applied Mechanics and Engineering 282:22–53

Andreasen C. S., Sigmund O. (2013) Topology optimization of fluidstructure-interaction problems in poroelasticity. Computer Methods in Applied Mechanics and Engineering 258:55–62

Arora J. (2004) Introduction to optimum design. Elsevier

Becker R. (2002) Mesh adaptation for Dirichlet flow control via Nitsches method. Communications in Numerical Methods in Engineering 18(9):669–680

Beissel S., Belytschko T. (1996) Nodal integration of the element-free Galerkin method. Computer methods in Applied Mechanics and Engineering 139:49–74

Belytschko T., Lu Y. Y., Gu L. (1994) Element-free Galerkin methods. International Journal for Numerical Methods in Engineering 37:229–256

Bendse M. (1989) Optimal shape design as a material distribution problem. Structural Optimization 1:193–202

Bendse M. P., Sigmund O. (2004) Topology Optimization: Theory, Methods and Applications. Berlin, Springer

Bosma T. (2013) Levelset based fluid-structure interaction modeling with the eXtended Finite Element Method. Master's thesis, TUDelft

Bourdin B., Chambolle A. (2003) Design-dependent loads in topology optimization. Control, Optimisation and Calculus of Variations 9:19–48

Chen B. C., Kikuchi N. (2001) Topology optimization with design-dependent loads. Finite Elements in Analysis and Design 37:57–70

Chen J. S., Wang H.-P. (2000) New boundary condition treatments in meshfree computation of contact problems. Computer Methods in Applied Mechanics and Engineering 187:441–468

Chen J. S., Pan C., Wu C. T., Liu W. K. (1996) Reproducing kernel particle methods for large deformation analysis of non-linear structures. Computer Methods in Applied Mechanics and Engineering 139(1-4):195–227

Chen J. S., Pan C., Wu C. T. (1997) Large deformation analysis of rubber based on a reproducing kernel particle method. Computational Mechanics 19(3):211–227

Chen J. S., Wux C. T., Yoon S., Youk Y. (2001) A stabilized conforming nodal integration for Galerkin mesh-free methods. International Journal for Numerical Methods in Engineering 50:435–466

Chen J. S., Zhang X., Belytschko T. (2004) An implicit gradient model by a reproducing kernel strain regularization in strain localization problems. Computer Methods in Applied Mechanics and Engineering 193(27-29):28272844

Chen J. S., Hillman M., Rter M. (2013) An arbitrary order variationally consistent integration for Galerkin meshfree methods. International Journal for Numerical Methods in Engineering 95:387–418

Chen J. S., Basava R. R., Zhang Y., Csapo R., Malis V., Sinha U., Hodgson J., Sinha S. (2017a) Pixel-based meshfree modelling of skeletal muscles. Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization 4(2):73–85

Chen J. S., Hillman M., Chi S. W. (2017b) Meshfree methods: Progress made after 20 years. Journal of Engineering Mechanics 143(4)(04017001)

Cheng J., Yu F., Zhang L. T. (2019) OpenIFEM: A high performance modular open-source software of the immersed Finite Element Method for fluid-structure interactions. Computer Modeling in Engineering and Sciences 119:91–124

Cho S., Kwak J. (2006) Topology design optimization of geometrically non-linear structures using meshfree method. Computer Methods in Applied Mechanics and Engineering 195:5909–5925

Choi K. K., Kim N. H. (2005) Structural Sensitivity Analysis and Optimization 1: Linear Systems. Mechanical Engineering, Springer

Coffin P., Maute K. (2016) A level-set method for steady-state and transient natural convection problems. Structural and Multidisciplinary Optimization 53:1047–1067

Conlan-Smith C., James K. A. (2019) A stress-based topology optimization method for heterogeneous structures. Structural and Multidisciplinary Optimization 60:167–183

Cui M., Chen H., Zhou J., Wang F. (2017) A meshless method for multi-material topology optimization based on the alternating active-phase algorithm. Engineering with Computers 33:871–884

van Dijk N. P., Maute K., Langelaar M., van Keulen F. (2013) Level-set methods for structural topology optimization: a review. Structural and Multidisciplinary Optimization 48:437–472

Dolbow J., Belytschko T. (1999) Numerical integration of the Galerkin weak form in meshfree methods. Computational Mechanics 23:219–230

Du J., Olhoff N. (2004) Topological optimization of continuum structures with design-dependent surface loading - part i: new computational approach for 2D problems. Structural and Multidisciplinary Optimization 27:151–165

Du Y.-X., Chen L.-P., Tian Q.-H., Wu Z.-J. (2009) Topology synthesis of thermomechanical compliant mechanisms with geometrical nonlinearities using meshless method. Advances in Engineering Software 40:315–322

Dunning P. D., Kim H. A., Mullineux G. (2011) Investigation and improvement of sensitivity computation using the area-fraction weighted fixed grid FEM and structural optimization. Finite Elements in Analysis and Design 47(8):933–941

Emmendoerfer H., Fancello E. A., Silva E. C. N. (2018) Level set topology optimization for design-dependent pressure load problems. International Journal for Numerical Methods in Engineering online 115(7):825–848

Engels-Putzka A., Frey C. (2021) Adjoint mesh deformation and adjoint-based sensitivities with respect to boundary values. European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS)

Feppon F., Allaire G., Bordeu F., Cortial J., Dapogny C. (2019) Shape optimization of a coupled thermal fluid-structure problem in a level set mesh evolution framework. SeMA 76:413–458

Fernandez-Mendez S., Huerta A. (2004) Imposing essential boundary conditions in mesh-free methods. Computer Methods in Applied Mechanics and Engineering 193:1257–1275

Freud J., Stenberg R. (1995) On weakly imposed boundary conditions for second order problems. Proceedings of the International Conference on Finite Elements in Fluids - New trends and applications, Venezia pp. 327–336

Gong S., Chen M., Zhang J., He R. (2012) Study on modal topology optimization method of continuum structure based on EFG method. International Journal of Computational Methods 9(1):1240,005

Gong S. G., Du J. X., Liu X., Xie G. L., Zhang J. P. (2010) Study on topology optimization under multiple loading conditions and stress constraints based on EFG method. International Journal for Computational Methods in Engineering Science and Mechanics 11(6):328–336

Gong S. G., Wei Y. B., lan Xie G., Zhang J. P. (2018) Study on topology optimization method of particle moving based on element-free Galerkin method. International Journal for Computational Methods in Engineering Science and Mechanics 19(5):305–313

Griebel M., Schweitzer M. A. (2002) A particle-partition of unity method. Part V: Boundary conditions. Springer, Berlin

Hammer V. B., Olhoff N. (2000) Topology optimization of continuum structures subjected to pressure loading. Structural and Multidisciplinary Optimization 19:85–92

Hansbo P., Larson M. G. (2002) Discontinuous Galerkin methods for incompressible and nearly incompressible elasticity by Nitsches method. Computer Methods in Applied Mechanics and Engineering 191(17-18):1895–1908

He Q., Kang Z., Wang Y. (2014) A topology optimization method for geometrically nonlinear structures with meshless analysis and independent density field interpolation. Computational Mechanics 54:629–644

Hedges L. O., Kim H. A., Jack R. L. (2017) Stochastic level-set method for shape optimisation. Journal of Computational Physics 348:82–107

Hillman M., Chen J. S. (2016) An accelerated, convergent, and stable nodal integration in Galerkin meshfree methods for linear and nonlinear mechanics. International Journal for Numerical Methods in Engineering 107(7):603–630

Hu H. H., Patankar N. A., Zhu M. Y. (2001) Direct numerical simulations of fluidsolid systems using the arbitrary LagrangianEulerian technique. Journal of Computational Physics 169:427–462

Huang T. H., Wei H., Chen J. S., Hillman M. (2019) RKPM2D: An open-source implementation of nodally integrated reproducing kernel particle method for solving partial differential equations. Computational Particle Mechanics 7:393–433

Huang X., Xie Y. M. (2007) Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method. Finite Elements in Analysis and Design 43:1039–1049

Hughes T. J. R. (2000) The Finite Element Method: Linear Static and Dynamic Finite Element Analysis. Dover Publications, Inc.

Isakari H., Kondo T., Takahashi T., Matsumoto T. (2017) A level-set-based topology optimisation for acoustic-elastic coupled problems with a fast BEM-FEM solver. Computer Methods in Applied Mechanics and Engineering 315:501–521

Jenkins N., Maute K. (2016) An immersed boundary approach for shape and topology optimization of stationary fluid-structure interaction problems. Structural and Multidisciplinary Optimization 54:1191–1208

Kambampati S., Du Z., Chung H., Kim H. A., Jauregui C., Townsend S., Picelli R., Zhou X. Y., Hedges L. (2018) OpenLSTO: Open-source software for level set topology optimization. In: Multidisciplinary Analysis and Optimization Conference, p. 3882

Kambampati S., Jauregui . C., Museth K., Kim H. A. (2019) Large-scale level set topology optimization for elasticity and heat conduction. Structural and Multidisciplinary Optimization 61:19–38

Kambampati S., Chung H., Kim H. A. (2020) A discrete adjoint based level set topology optimization method for stress constraints. Computer Methods in Applied Mechanics and Engineering 377:113,563

Khan W., Islam S., Ullah B. (2019) Structural optimization based on meshless element free Galerkin and level set methods. Computer Methods in Applied Mechanics and Engineering 344:144–163

Khoei A. R. (2014) Extended Finite Element Method: Theory and Applications. Wiley Series in Computational Mechanics

Lacroix D., Bouillard P. (2003) Improved sensitivity analysis by a coupled FEEFG method. Computers and Structures 81:2431–2439

Lee E., Martins J. R. R. A. (2012) Structural topology optimization with design-dependent pressure loads. Computer Methods in Applied Mechanics and Engineering 233-236:40–48

Li J., Guan Y., Wang G., Wang G., Zhang H., Lin J. (2020) A meshless method for topology optimization of structures under multiple load cases. Structures 25:173–179

Lin J., Guan Y., Zhao G., Naceur H., Lu P. (2016) Topology optimization of plane structures using smoothed particle hydrodynamics method. International Journal for Numerical Methods in Engineering 110:726–744

Liu W. K., Chang H., Chen J. S., Belytschko T. (1988) Arbitrary LagrangianEulerian PetrovGalerkin finite elements for nonlinear continua. Computer Methods in Applied Mechanics and Engineering 68:259–310

Liu W. K., Jun S., Zhang Y. F. (1995) Reproducing kernel particle methods. International Journal for Numerical Methods in Fluids 20:1081 – 1106

Liu W. K., Jun S., Sihling D. T., Chen Y., Hao W. (1997) Multiresolution reproducing kernel particle method for computational fluid dynamics. International journal for numerical methods in fluids 24(12):1391–1415

Lu H., Chen J. S. (2003) Adaptive Galerkin particle method. Meshfree Methods for Partial Differential Equations, Lecture Notes in Computational Science and Engineering 26:251–265

Lundgaard C., Alexandersen J., Zhou M., Andreasen C. S., Sigmund O. (2018) Revisiting density-based topology optimization for fluid-structure-interaction problems. Structural and Multidisciplinary Optimization 58:969–995

Luo Z., Zhang N., Gao W., Ma H. (2012a) Structural shape and topology optimization using a meshless Galerkin level set method. International Journal for Numerical Methods in Engineering 90:369–389

Luo Z., Zhang N., Ji J., Wu T. (2012b) A meshfree level-set method for topological shape optimization of compliant multiphysics actuators. Computer Methods in Applied Mechanics and Engineering 223-223:133–152

Luo Z., Zhang N., Wang Y., Gao W. (2013) Topology optimization of structures using meshless density variable approximants. International Journal for Numerical Methods in Engineering 93:443–464

Mader C., Martins J. R. R. A. (2008) Adjoint: An approach for rapid development of discrete adjoint solvers. AIAA Journal 46(4):863–873

Martins J., Sturdza P., Alonso J. (2003) The complex-step derivative approximation. ACM Transactions on Mathematical Software, Association for Computing Machinery 29:245–262

Miegroet V. L., Duysinx P. (2007) Stress concentration minimization of 2D filets using X-FEM and level set description. Structural and Multidisciplinary Optimization 33(4):425–438

Museth K. (2017) Novel algorithm for sparse and parallel fast sweeping: efficient computation of sparse signed distance fields. ACM SIGGRAPH 2017 Talks (74):1–2

Nadarajah S. K. (2003) The discrete adjoint approach to aerodynamic shape optimization. PhD thesis, Stanford University

Neofytou A., Picelli R., Chen J. S., Kim H. A. (2019) Level set topology optimization for design dependent pressure loads: A comparison between FEM and RKPM. AIAA 2019-3559 https://doi.org/10.2514/6.2019-3559

Neofytou A., Picelli R., Huang T. H., Chen J. S., Kim H. A. (2020) Level set topology optimization for design-dependent pressure loads using the reproducing kernel particle method. Structural and Multidisciplinary Optimization 61:1805–1820

Nielsen E. J., Park M. A. (2006) Using an adjoint approach to eliminate mesh sensitivities in computational design. AIAA Journal 44(5):948–953

Norato J. A., Bendse M. P., Haber R. B., Tortorelli D. A. (2007) A topological derivative method for topology optimization. Structural and Multidisciplinary Optimization 33(4-5):375–386

Patil M. J., Kulkarni M. D., Canfield R. (2015) Discrete adjoint formulation for continuum sensitivity analysis. AIAA Journal 54(2):758–766

Picelli R., Vicente W. M., Pavanello R. (2015a) Bi-directional evolutionary structural optimization for design-dependent fluid pressure loading problems. Engineering Optimization 47(10):1324–1342

Picelli R., Townsend S., Brampton C., Norato J., Kim H. (2018) Stress-based shape and topology optimization with the level set method. Computer Methods in Applied Mechanics and Engineering 329:1–23

Picelli R., Neofytou A., Kim H. A. (2019) Topology optimization for design-dependent hydrostatic pressure loading via the level set method. Structural and Multidisciplinary Optimization 60:1313–1326

Picelli R., Ranjbarzadeh S., Sivapuram R., Gioria R. S., Silva E. C. N. (2020) Topology optimization of binary structures under design-dependent fluid-structure interaction loads. Structural and Multidisciplinary Optimization 62:2101–2116

Pingen G., Waidmann M., Evgrafov A., Maute K. (2010) A parametric level-set approach for topology optimization of flow domains. Structural and Multidisciplinary Optimization 41(1):117–131

Puso M., Chen J. S., Zywicz E., Elmer W. (2008) Meshfree and finite element nodal integration methods. International Journal for Numerical Methods in Engineering 74:416–446

Rycroft C. H. (2009) A three-dimensional Voronoi cell library in c++. Chaos 19:041,111

Sanches R. P. (2015) Evolutionary topology optimization of fluid-structure interaction problems. PhD thesis, University of Campinas

Secco N. R., Martins J. R. R. A. (2019) Rans-based aerodynamic shape optimization of a strut-braced wing with overset meshes. Journal of Aircraft 56(1):217–227

Sethian J. (1996) A fast marching level set method for monotonically advancing fronts. Proceedings of the National Academy of Sciences 93(4):1591–1595

Sharma R., Zhang J., Langelaar M., van Keulen F., Aragn A. M. (2018) An improved stress recovery technique for loworder 3D finite elements. International Journal for Numerical Methods in Engineering 114(1):88–103

Shobeiri V. (2016) Topology optimization using bi-directional evolutionary structural optimization based on the element-free Galerkin method. Engineering Optimization 48(3):380–396

Shu L., Wang M. Y., Ma Z. (2014) Level set based topology optimization of vibrating structures for coupled acoustic-structural dynamics. Computers and Structures 132:34–42

Sigmund O. (2001a) A 99 line topology optimization code written in matlab. Structural and Multidisciplinary Optimization 21:120–127

Sigmund O., Clausen P. M. (2007) Topology optimization using a mixed formulation: An alternative way to solve pressure load problems. Computer Methods in Applied Mechanics and Engineering 196:1874–1889

Sigmund O., Maute K. (2013) Topology optimization approaches: A comparative review. Structural and Multidisciplinary Optimization 48:1031–1055

Sivapuram R., Picelli R. (2017) Topology optimization of binary structures using Integer Linear Programming. Finite Elements in Analysis and Design 139:49–61

Smith M. (2014) Abaqus/standard users manual Version 6.14

Tanojo E., Pudjisuryadi P. (2006) Meshless methods: Alternatives for solving 2D elasticity problems. International Civil Engineering Conference ”Towards Sustainable Civil Engineering Practice”

Wang H. P., Wu C. T., Chen J. S. (2014) A reproducing kernel smooth contact formulation for metal forming simulations. Computational Mechanics 54(1):151–169

Wang M. Y., Wang X., Guo D. (2003) A level set method for structural topology optimization. Computer Methods in Applied Mechanics and Engineering 192:227–246

Wang X., Zhang L. T. (2013) Modified immersed finite element method for fully-coupled fluidstructure interactions. Computer Methods in Applied Mechanics and Engineering 267:150–169

Wein F., Dunning . P. D., Norato J. A. (2020) A review on feature-mapping methods for structural optimization. Structural and Multidisciplinary Optimization 62:1597–1638

Xia Q., Wang M. Y., Shi T. (2015) Topology optimization with pressure load through a level set method. Computer Methods in Applied Mechanics and Engineering 283:177–195

Xie Y. M., Steven G. P. (1993) A simple evolutionary procedure for structural optimization. Computer Structures 49:885–896

Yamada T., Izui K., Nishiwaki S., Takezawa A. (2010) A topology optimization method based on the level set method incorporating a fictitious interface energy. Computer Methods in Applied Mechanics and Engineering 199(45-48):28762891

Yoon G. H. (2010) Topology optimization for stationary fluidstructure interaction problems using a new monolithic formulation. International Journal for Numerical Methods in Engineering 82:591–616

Yoon G. H. (2014) Stress-based topology optimization method for steady-state fluidstructure interaction problems. Computer Methods in Applied Mechanics and Engineering 278:499–523

Zhang H., Zhang X., Liu S. (2008) A new boundary search scheme for topology optimization of continuum structures with design-dependent loads. Structural and Multidisciplinary Optimization 37:121–129

Zhang J., Wang S., Zhou G., Gong S., Yin S. (2020) Topology optimization of thermal structure for isotropic and anisotropic materials using the element-free Galerkin method. Engineering Optimization 52(7):1097–1118

Zhang L. T., Gay M. (2007) Immersed finite element method for fluidstructure interactions. Journal of Fluids and Structures 23:839857

Zhang L. T., Gerstenberger A., Wang X., Liu W. K. (2004) Immersed finite element method. Computer Methods in Applied Mechanics and Engineering 193:20512067

Zhang S., Gain A. L., Norato J. A. (2017) Stress-based topology optimization with discrete geometric components. Computer Methods in Applied Mechanics and Engineering 325:1–21

Zhang Y., Ge W., Zhang Y., Zhao Z. (2018a) Topology optimization method with direct coupled finite elementelement free Galerkin method. Advances in Engineering Software 115:217–229

Zhang Y., Ge W., Zhang Y., Zhao Z., Zhang J. (2018b) Topology optimization of hyperelastic structure based on a directly coupled finite element and element-free Galerkin method. Advances in Engineering Software 123:25–37

Zhao F. (2014) Topology optimization with meshless density variable approximations and BESO method. Computer-Aided Design 56:1–10

Zheng J., Long S., Li G. (2012) Topology optimization of free vibrating continuum structures based on the element free Galerkin method. Structural and Multidisciplinary Optimization 45:119–127

Zhou J. X., Zou W. (2008) Meshless approximation combined with implicit topology description for optimization of continua. Structural and Multidisciplinary Optimization 36:347–353

Zhu T., Atluri S. N. (1998) A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free Galerkin method. Computational Mechanics 21(3):211–222

Zienkiewicz O., Taylor R., Nithiarasu P. (2014) The Finite Element Method for Fluid Dynamics, 7th edn. Elsevier