Shape Optimization Directly from CAD: an Isogeometric Boundary Element Approach

Haojie Lian

Supervisors: Professor Stéphane Bordas
Dr. Pierre Kerfriden

A thesis submitted to the graduate school in fulfilment of the requirement for the degree of Doctor of Philosophy

February, 10, 2015

Advanced Materials and Computational Mechanics Group
The School of Engineering, Cardiff University
Cardiff, Wales, UK

天之道，其犹张弓欤？高者抑之，下者举之，有余者损之，不足者补之。

Does not the Tao of heaven resemble the bending of a bow? Pressing down the high, lifting up the low, reducing the excessive, compensating the deficient.

—Lao Tzu, *Tao Te Ching*, Chapter 77, 516 B.C.

Summary

The present thesis addresses shape sensitivity analysis and optimization in linear elasticity with the isogeometric boundary element method (IGABEM), where the basis functions used for constructing geometric models in computer-aided design (CAD) are also employed to discretize the boundary integral equation (BIE) for structural analysis, and to discretize the material differentiation form of the BIE for shape sensitivity analysis. To guarantee water-tight and locally-refined geometries, we use non-uniform rational B-splines (NURBS) and T-splines for two-dimensional and three dimensional problems, respectively. In addition, we take advantage of the regularized form of BIE instead of the singular form, to bypass the difficulties caused by the evaluation of strongly singular integrals and jump terms. The main advantages of the present work arise from the ability of the IGABEM to seamlessly integrate CAD and numerical analysis, since they share the same boundary representation of geometric models. Therefore, throughout the whole shape optimization, it does not need a costly meshing/remeshing procedure. Moreover, the control points can be naturally chosen as the design variables, and the optimal solution can be directly returned to the CAD system without any smoothing procedure.

Declaration

DECLARATION

This work has not previously been accepted in substance for any degree and is not concurrently submitted in candidature for any degree.

Signed........................... (Candidate) Date.........................

STATEMENT 1

This thesis is being submitted in partial fulfilment of the requirements for the degree of PhD.

Signed........................... (Candidate) Date.........................

STATEMENT 2

This thesis is the result of my own independent work/investigation, except where otherwise stated. Other sources are acknowledged by explicit references.

Signed........................... (Candidate) Date.........................

STATEMENT 3

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed........................... (Candidate) Date......................

Acknowledgements

I would like firstly to express my deep gratitude to my supervisor Prof. Stéphane Bordas, for his guidance, support and encouragement during the past four years. He opened a door for me to a fascinating academic world. With the wide vision on the science and the deep insight into the numerical analysis, he always can find an exciting research direction and guide me go through any difficulty to reach the goal. Talking with him was always enjoyable and also inspiring, not only in the academics but also the life.

I cannot thank Dr. Robert Simpson enough for his endless help in my study. His assistance on the theory and programming of boundary element methods and isogeometric analysis was essential to my research. The present work would not be possible without his help. His attitude and enthusiasm on academics also set a goal for me to pursue.

I wish to acknowledge Dr. Pierre Kerfriden, whose broad knowledge on mathematics and nonlinear mechanics was enlightening to me. Even a short discussion with him could give me important clues to my research questions.

I would like to give some special thanks to my colleagues and friends. Dr Sundararajan Natarajan offered considerable help to me, which was especially precious at the beginning stage of my research when I lacked experience extremely. Dr Ahmad Akabari taught me a large amount of knowledge in meshfree methods and Matlab programming. Olivier Goury, an excellent mathematician, was always the first one I resorted to when encountering mathematical difficulties. His clear explanations enabled any challenging mathematics concept to become easily understood. Chang-Kye Lee is experienced in structural optimization and nonlinear mechanics,

who helped me clarify numerous concepts in these areas. Daniel Alves Paladimis is my first tutor in C++ languages and I learned countless theories and skills in programming from him. Thanks also goes to Xuan Peng, whose research area was close to mine, and my research project benefited directly from the discussion with him. I also learned a lot from Danas Sutula, not only the knowledge on XFEM and fracture mechanics, but also the spirit of the full dedication to the research. In addition, I wish acknowledge Dr. Octavio Andrés González-Estrada, Dr. Claire Heaney, Dr. Iulia Mihai, Dr. Nguyen Vinh Phu, Dr. Chi Hoang and Dr. Elena Atroshchenko, who are all experienced researchers and gave me numerous precious advices on my research.

My words are not enough to thank my girlfriend Xiao. Without her love, care and devotion, I could not have focused on the research. It is her faith in me that renders me courage to go through the difficulties.

Last but not least, I would like to express my gratitude to my parents for everything. I owe everything to my parents for what I have become.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Shape optimization is a process to find the optimal shape of a component or structure under given constraints and objectives. Shape optimization involves the variation of boundaries, and the design space is the parameters which construct the geometry. It is worth noting that shape optimization is different from topology optimization [14–16, 91, 92] in that the latter requires nucleation of holes and determination of the hole numbers and locations. The design space of topology optimization is material parameters and an optimal material layout is searched. See Fig. 1.1 for the comparison. In the engineering design process, shape optimization can be carried out independently of topology optimization, or performed after the topology optimization to "tune" the shape. In a long period, shape optimization was performed manually in industry, relying on designers' experiences and intuitive, as shown in Fig. 1.2. This procedure is typically computationally expensive and cannot guarantee a reliable optimal result. To achieve automated and fast shape optimization, numerous methods were proposed, whose advantages and limitations will be reviewed in this chapter.

1.1 The formulation of shape optimization

Shape optimization can be conducted through a gradient-less or gradient-based method. Gradient-less shape optimization does not require the evaluation of the

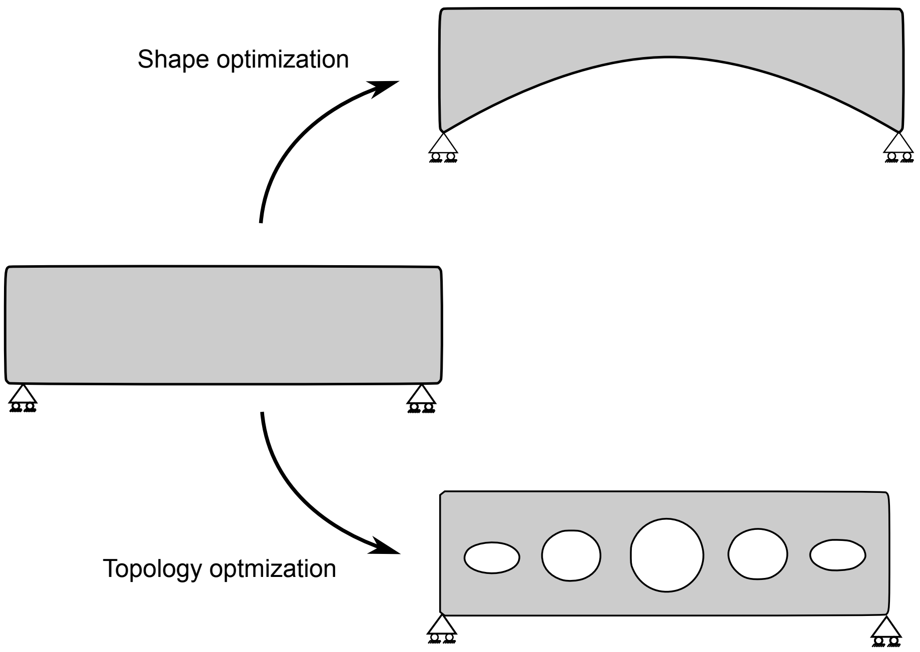


Figure 1.1: Shape optimization and topology optimization

shape derivatives, but can be prohibitively time-consuming for realistic problems and is not supported by a mathematical theory. So, gradient-based methods are normally preferred and thus are also used in the present work. Gradient-based shape optimization has a well-grounded mathematical foundations rooted in optimal-control theory. A shape optimization problem can be formulated as minimizing an objective function

$$f : \mathbb{R}^n \to \mathbb{R}, \tag{1.1.1}$$

$$f : \mathbf{t} \mapsto f\left(\mathbf{t}\right), \tag{1.1.2}$$

subject to the constraints

$$g_i\left(\mathbf{t}\right) \leqslant 0 \qquad \text{for } i = 1, \ldots, m, \tag{1.1.3}$$

$$t_i^l \leqslant t_i \leqslant t_i^u, \tag{1.1.4}$$

where $\mathbf{t}$ is a vector of parameters which controls geometrical configurations, also called design variables. f is the objective function, g_i the constraint functions, i the constraint function index, m the number of constraints. Eq. (1.1.4) gives side constraints to limit the search region of the design variables, where t_i^l and t_i^u are

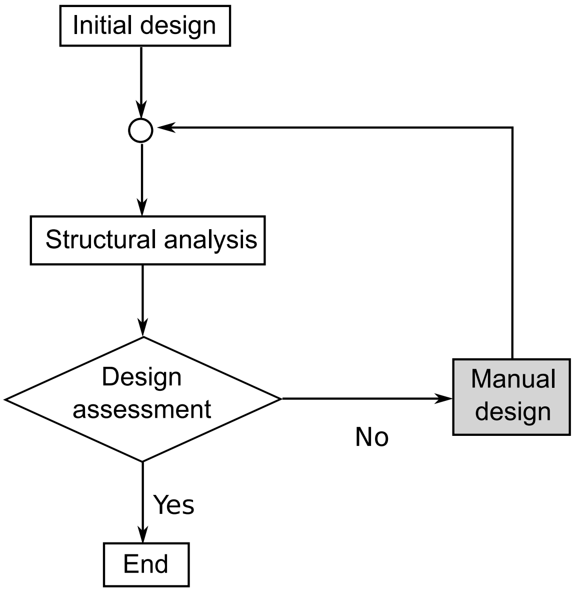


Figure 1.2: Manual design process

lower and upper bounds of the design variables, respectively. A design is called feasible if all constraints are satisfied.

To find the parameters associated with the minimum value of the objective function, numerical optimization algorithms employ the gradient of the objective and constraint functions within an iterative algorithm:

$$\left(f_k, g_i^k, \frac{\mathrm{d}}{\mathrm{d}\mathbf{t}} f_k, \frac{\mathrm{d}}{\mathrm{d}\mathbf{t}} g_i^k \right) \to (f_{k+1}, g_i^{k+1}), \tag{1.1.5}$$

where k denotes the kth iteration step, $\frac{\mathrm{d}}{\mathrm{d}\mathbf{t}} f_k$ and $\frac{\mathrm{d}}{\mathrm{d}\mathbf{t}} \mathbf{g}_k$ are the shape derivatives or sensitivities. A numerical shape optimization procedure is divided into the following steps:

1. Define the objective function and constraints.
2. Parameterize the boundary and choose the design variables.
3. Evaluate the objective functions and the constraint functions.
4. Evaluate the shape derivatives of the objective and constraint functions.
5. Check whether the convergence criteria are satisfied. Calculate the next set of design variables if the criteria are satisfied, or stop the iterations otherwise.

1.1.1 Objectives and constraints in shape optimization

The commonly used objectives or constraints in shape optimization in elasticity include

- The volume of the structure

  $$f\left(\mathbf{t}\right)=\int_{\Omega}\mathrm{d}\Omega, \tag{1.1.6}$$

  where Ω is the domain occupied by the materials. This volume is widely used as objective combined with stress constraints, or used as constraints for conserved energy or displacement minimization.

- The weight of the structure

  $$f\left(\mathbf{t}\right)=\int_{\Omega}\rho\mathrm{d}\Omega, \tag{1.1.7}$$

  where ρ is the density of the material. If ρ is a constant over the domain, the weight optimization of the structure is reduced to the volume optimization.

- Displacement. The displacement $\mathbf{u}$ should not exceed a prescribed value $\hat{\mathbf{u}}$.

  $$\mathbf{u}\left(\mathbf{t}\right)-\hat{\mathbf{u}}\leqslant 0 \qquad \mathrm{for} i=1,\ldots,m. \tag{1.1.8}$$

- Average stress over part of the structure Ω_c,

  $$f\left(\mathbf{t}\right)=\int_{\Omega_c}\left(\frac{\sigma-\bar{\sigma}}{\bar{\sigma}}\right)^2\mathrm{d}\Omega. \tag{1.1.9}$$

  where σ is the equivalent stress, and $\bar{\sigma}$ is the average stress to be achieved.

- Conserved energy

  $$\int_{\Omega}\sigma_{ij}\epsilon_{ij}\mathrm{d}\Omega, \tag{1.1.10}$$

  where σ_{ij} and ϵ_{ij} are the stress and strain tensor.

- Maximum equivalent stress. The equivalent stress σ over a structure should not exceed the allowable stress $\hat{\sigma}$.

  $$\sigma_i\left(\mathbf{t}\right)-\hat{\sigma}_i\leqslant 0 \qquad \text{for } i=1,\ldots,m, \tag{1.1.11}$$

where i is the index of the stress monitor points. The equivalent stress can be chosen as principal stress, von Mises stress, tangential stress, *etc.* Such function is usually used for optimizing the stress concentration or for the stress constrained volume minimization.

- Eigenvalue. The eigenvalue problem is widely seen in free vibration and buckling problems. For free vibration problems, a commonly used objective is to maximize the minimum eigenvalue, i.e.

$$\max : \min_{i=1,2,3,...} \lambda_i \tag{1.1.12}$$

For buckling problems, a useful objective is to maximize the minimum critical load, i.e.

$$\max : \{P_{critical} = \min_{i=1,2,3,...} P_i\} \tag{1.1.13}$$

1.1.2 Boundary representation

Parameter-free approach

Parameter-free approach employs finite element nodes (coordinates or movement) as the design variables directly, which was also the first approach used in shape optimization [116]. The method possesses the following prominent advantages: 1) It does not need a parameterization procedure. Once the mesh is generated, the shape optimization iterative procedure can be performed independent of CAD. However, a postprocessing is still needed to recover CAD models, which is cumbersome and contaminates the optimized solution. 2) It can provide a large design space. 3) It is easier to integrate with topology optimization [30, 31]. However, it also faces the following main challenges: 1) A large number of design variables are used, although the associated time cost can be alleviated by the adjoint variable method, if the number of constraints is small. 2) It is easier to yield jagged boundaries [48] compared to parameterized approach. Inspired by the pioneering work in topology optimization [16], the filter was proposed in shape optimization to reach smooth geometries. [5, 17, 44] adopted the filters to regularize the shape sensitivities and thus the resulting geometry will become smooth consequently. This method lacks a strict mathematical basis, and thus the optimality cannot be guaranteed and verified. A recent advance in the area is [57], which proposed a scheme to consistently filter the design variables, i.e. the influence of the filter has been taken into account in the shape sensitivity evaluation, which precludes the oscillations by allowing only the smoothed mesh to be used for analysis.

Parameterised approach

The approach parameterizes the boundary before the optimization is performed. A preferred boundary representation should have the capability of constructing a sufficient large space where the optimal solution is searched, but without many redundant design variables.

- **Level set** Level set is an implicit representation of the geometry. Normally the level set function can be discretized using the domain nodes. The advantage of level set function is that it can construct smooth geometry and track the surface in a fixed grid without needing to parameterize the surface. The shortcomings are that a care must be taken for constructing the velocity field and the geometry advancing needs to solve a differential equation. Furthermore, although it does not need to conform to the geometry, the mesh has to be refined sufficiently to reduce the geometrical and numerical errors. However, even a fine mesh is used, the edges and sharp corners are still difficult to be captured by level set. A contribution in this direction can be seen in [72] where a simple adaptive mesh refinement strategy was proposed.

- **Spline representation** Splines are composed of low-order polynomials, which can achieve high order smoothness with a small number of nodes. The coefficients in the spline expression or the nodes interpolated by the splines can be chosen as the design variables.

- **Free-form representation** Free-form representation also employs splines, such as Bézier curves, B-splines and NURBS. However, it possesses the advantage that control points, normally taken as design variables, have a very clear geometrical interpretation, and thus is superior for interactive design and shape optimization.

1.2 Numerical methods in shape optimization

A characteristic of shape optimization is that objective or constraint functions are not given explicitly, which necessitates solving a system of equations to evaluate the functions and the shape derivatives. As a most widely used method in computational solid mechanics, the finite element method (FEM) [116] was firstly applied in shape optimization [115]. However, a mesh must be created in FEM to approximate the geometry and discretize the governing partial differential equation (PDE) to allow analysis to be performed. Shape optimization is an iterative procedure and geometries vary at each step, thus leading to cumbersome remeshing procedures, which occupy around 80% of the total problem solving time for linear elastic problems at each iterative step and accumulate to an unaffordable computational burden. Moreover, it may happen in industrial practices that the geometry is so complex that available mesh generators fail, or require significant human intervention, which precludes the automated algorithm. So the mesh burden has become the bottleneck to achieving an automated and fast shape optimization. To solve this problem, numerous works are proposed from various perspectives, which will be reviewed in next section.

1.2.1 Meshfree/Meshless methods

Meshfree (meshless) methods [74], refer to a broad collection of numerical methods, including the smoothed particle hydrodynamics method (SPH), the element-free Galerkin method (EFG) [13], the reproducing kernel particle Method (RKPM) [63], the meshless local Petrov-Galerkin method (MLPG) [3] , the *hp*-cloud method [42], the partition of unity finite element method (PUFEM) [70], *etc.* Although different in the way of formulating shape functions, they share the same characteristic of

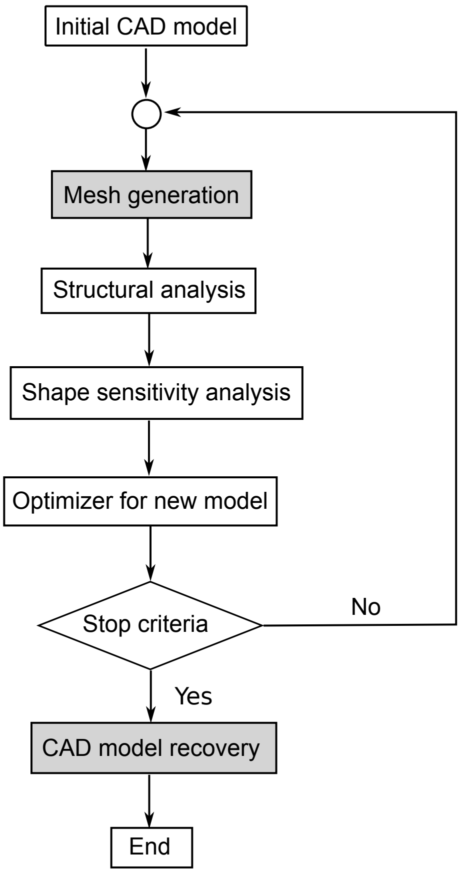


Figure 1.3: Shape optimization with FEM

lifting the strict connectivity requirements posed by the FEM. In contrast to FEM, meshfree methods do not employ elements in the construction of the approximation. Instead, a set of nodes associated with a domain of influence are sufficient (Fig. 1.4). The connectivity between the nodes determined by the overlapping of these domains of influence can be defined more flexibly than in the FEM. The application of meshfree methods in shape optimization can be found in [18, 19, 114]. However, the arbitrariness in the node placement is relative since the quality of the approximation is known to be dependent on the geometrical location of the nodes and on the domain of influence of each node. For recent progress in this direction, the interested readers can refer to the work in [84], where a variational adaptivity approach was proposed to optimize the support domain size of meshfree shape functions. Since the most widely employed shape functions in meshfree methods are rational functions, Gauss quadrature is not sufficient to achieve an exact inte-

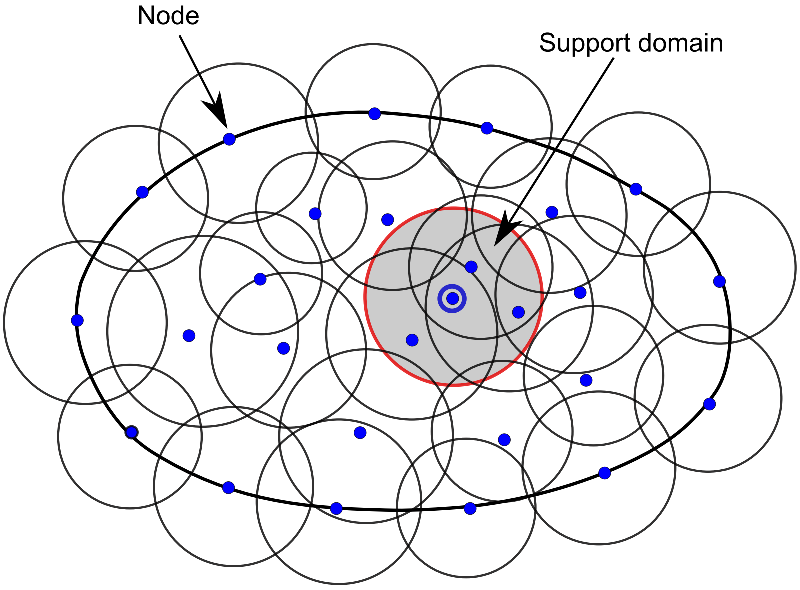


Figure 1.4: Meshfree methods

gration. Background meshes [40] or other advanced techniques must thus be used, such as direct nodal integration [8, 11, 12], stabilized nodal integration [25], stress point integration [10, 82], support-based integration [24, 43], which complicate the approach and somewhat detract from the "meshfree" concept.

1.2.2 Implicit boundary methods

The main difficulty in mesh generation emanates from the requirement of the mesh to conform to the (usually arbitrarily complex) geometry of the domain. To separate the FEM mesh and the geometry representation, implicit boundary methods were proposed (Fig. 1.5), including immersed boundary methods [71], fictitious domain methods [45], embedded boundary methods [51], virtual boundary methods [85], Cartesian grid methods [113]. The extended finite element method (XFEM) [9, 21, 22, 40] also falls into this category and its application in shape optimization can be read in [100]. The advantage of implicit boundary methods over IGA is that it allows a certain flexibility in the choice of basis functions, which may be different for the field variables and the geometry of the domain. However, due to the separation of the geometry and the analysis mesh, the capture of the geometry boundary for domain integration is not a trivial task.

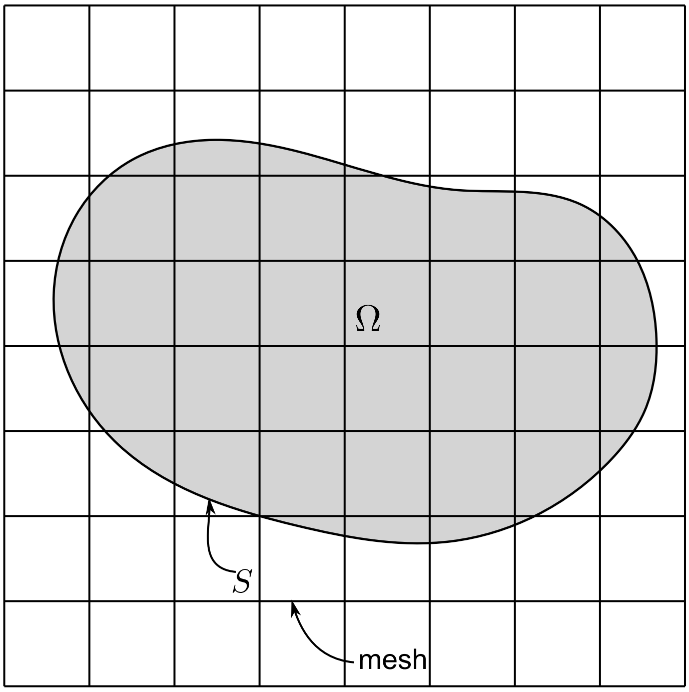


Figure 1.5: XFEM mesh

1.2.3 Boundary element methods

Boundary element methods (BEM) [37,38,50,83,96] take the advantage of boundary integral equations to decrease the dimension of the problem by one, *i.e.* only line integrals are needed for two-dimensional problems, and surface integrals for three-dimensional problems (Fig. 1.6). The main benefit of using BEM in shape optimization is that it alleviates the mesh generation burden because surface mesh generation is much easier and faster than domain mesh generation. The application of BEM in shape optimization can be read in [105, 111, 112]. However, the surface meshing is still not affordable for shape optimization in large scale problems due to the repeated remeshing procedure. Moreover, BEM's advantages are mostly visible in cases where Green's functions are available.

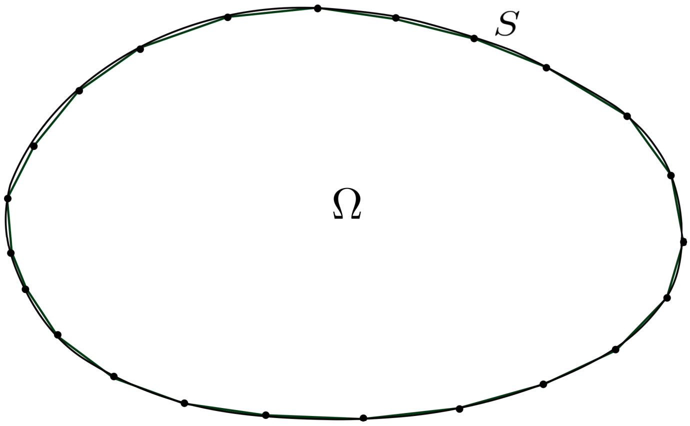


Figure 1.6: BEM mesh

1.2.4 CAD analysis integration

Isogeometric analysis

A promising way to overcome the mesh burden is the integration between CAD and analysis. To the author's knowledge, the first work of integrating CAD and engineering analysis can be traced back to the work by Kagan *et al.* [52], where B-splines were used as basis functions to represent both the geometry and the unknown fields. Following this idea, Cirak *et al.* [32–34] proposed a paradigm for thin-shell analysis, but used subdivision surfaces instead of B-splines. These ideas were formalized and generalized by Hughes *et al.* in isogeometric analysis (IGA), which was initially based on NURBS [49], and then extended to T-splines [7] and PHT-splines [75, 76, 102]. The key idea of IGA is to use the data provided by CAD models *directly* rather than converting it through a preprocessing routine into a form suitable for analysis. Hence, the meshing procedure is bypassed because an existing CAD geometry is used directly for analysis, meanwhile keeping the exact geometry. The application of IGA in shape optimization can be found in [26, 47, 69, 81, 101]. However, IGA enfronts the following challenges:

- The main bottleneck of IGA is that CAD is based on boundary representation, conflicting with a domain-discretization based analysis model. A general and efficient algorithm needs to be developed to construct the parameterization of the interior of the domain, which is far from a trivial task. The recent progress in this direction can be read in [62, 107–110].

- The basis functions of IGA must be the same as that used to describe the geometry, thus losing the flexibility to take advantage of other types of basis functions, which has particular use in some cases, for example, in the spectral element method.

To remedy the aforementioned difficulties of IGA, numerous methods are proposed and will be reviewed as follows.

NURBS-enhanced finite element methods

NURBS-enhanced finite element methods (NEFEM) [89, 90] employ NURBS for the geometric description of the boundary, while keeping the flexibility of FEM by using polynomial interpolation. In NEFEM, only the elements having an edge or face in contact with the NURBS boundary are treated using specific interpolation and integration strategy, whereas interior elements not affected by the NURBS boundary can be defined as standard finite elements. See Fig. 1.7. NEFEM possess the advantage of accurate representation of geometry and alleviate the difficulty of generating interior isogeometric elements. However, NEFEM does not reduce the complexity in mesh generation compared to FEM.

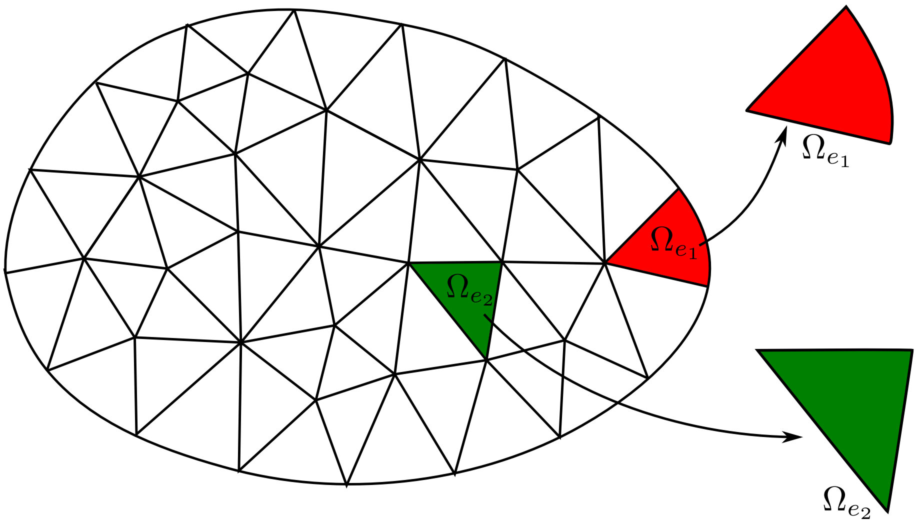


Figure 1.7: NEFEM mesh

Implicit boundary method with NURBS

Based on the work of [72], Moumnassi *et al.* [73] proposed a scheme to obtain an implicit domain definition from arbitrary parametric surfaces provided from CAD data, through multiple level set and boolean operations. The methods can exactly capture geometries, including the corners and sharp edges. Moreover, the method does not require local refinement of the finite element mesh in regions of high curvature, enhancing the independence of the domain's geometry on the mesh. Another approach to combine the advantages of implicit boundary methods and exact geometry representation is NURBS-enhanced XFEM [58]. Inspired by NEFEM, NURBS-enhanced mapping is employed for the subelements in the elements intersected by the interface, instead of level sets commonly used in conventional XFEM.

Both approaches mentioned above can use the geometry data provided by CAD directly in analysis and preserving geometric exactness. However, the process for utilizing the CAD data is still time consuming and thus not practical in large scale shape optimization.

Geometry independent field approximation

In the geometry-independent field approximation (GIFT) [106], different spline spaces for the geometry and the field variables can be chosen and adapted independently while keeping the exact CAD description and tight CAD integration. Compared to IGA, GIFT provides the flexibility to choose a spline space more suitable to analysis than that used in CAD. In addition, refinement operations by knot insertion and order elevation for analysis can be performed directly in the solution field, independently of the spline space of the geometry. Similar to IGA, however, GIFT still requires a volume parameterization which is an open problem as mentioned above.

1.3 The thesis organization

The limitation of the methods discussed above motivated the present work, where the isogeometric boundary element method (IGABEM) will be applied to shape optimization in linear elasticity. Based on the same boundary representation as CAD, IGABEM can achieve a seamless CAD and analysis integration, which renders IGABEM immediate advantages in the application of shape optimization: 1) the mesh generation/regeneration can be avoided, 2) the automation of the optimization process is more realistic because no human intervention is introduced for constructing analysis-suitable model and meshing, 3) the free-form representation for shape optimization can be naturally reached, and 4) the optimal geometrical model output can be used directly in CAD without any postprocessing or “smoothing” procedure.

Compared to the work [60] in 2011 which applied IGABEM with NURBS for shape optimization in linear elasticity, the thesis has the following differences: 1) T-splines [87, 88] are used in the IGABEM shape optimization for surface representation, so a water-tight geometry can be guaranteed, which can avoid the time cost in geometry repair and more importantly, guarantee the automation of the shape optimization. A recent work by [56] in 2015 has also used T-spline based IGABEM for shape optimization, which is for wave-resistance problem. 2) The thesis incorporates the technique to separate the NURBS geometric control mesh and sensitivity analysis mesh, which was firstly proposed by [81] in IGAFEM. 3) The regularized form of the Boundary Integral Equation is discretized for the analysis of the unknown fields and the sensitivities, avoiding the necessity to evaluate jump terms and strongly singular integrals, which were the main obstacle in IGABEM implementation for arbitrary geometries. The remainder of the thesis is organized as follows.

Chapter 2 reviews the concept and formulations of two important geometric modelling techniques, NURBS and T-splines, which play key roles both in CAD and IGABEM. Their element structures and Bézier extraction techniques are explained, which facilitate the integration of IGA with existing numerical analysis code.

Chapter 3 details the implementation of IGABEM. We start with the derivation of fundamental solutions and boundary integral equations (BIE). Then we compare the differences between the implementations of the singular BIE based BEM and IGABEM. In this Chapter, our contribution is to employ a regularized boundary integral equation to form the IGABEM formulations, which does not need to evaluate jump terms or strongly singular integrals.

Chapter 4 applies IGABEM to shape sensitivity analysis. Shape sensitivity analysis is a critical step in gradient-based shape optimization algorithms. We adopt the implicit differentiation method which was widely used in BEM. The present work is distinct from the previous work in that 1) the shape derivatives of field variables are discretized using NURBS or T-spline, and 2) the material differentiation is performed on a regularized form of BIE. The numerical examples are presented for the verification of IGABEM sensitivity analysis, where the geometries in two-dimensions are modelled by NURBS and that in three-dimensions by T-splines.

Chapter 5 presents the application of IGABEM in shape optimization in linear elasticity. We firstly reviewed the algorithm to transit shape derivatives from coarse mesh to refined mesh in NURBS, allowing structural analysis to be performed in a fine mesh while restricting the shape derivatives to be performed with respect to the design variables (control points) in a relatively coarse mesh. To investigate the performance of the IGABEM in shape optimization, the numerical examples are given, involving displacement minimization problems, stress constraint optimization problems, and conserved energy minimization problems. The two-dimensional geometries employ NURBS and three-dimensional models adopt T-splines. Therefore, a water-tight geometry can be always guaranteed.

Chapter 6 contains the conclusions of the present thesis, and the remarks on the future work.

Chapter 2

NURBS and T-splines

NURBS and T-splines are two important geometric modelling techniques in computer-aided design (CAD). Due to the need of integrating CAD and analysis, they also play important roles in the isogeometric analysis (IGA) and the isogeometric boundary element method. This chapter will review the formulations of NURBS and T-splines. In addition, the Bézier extraction technique is also explained, which improves the efficiency of shape function evaluation and enables the existing finite element or boundary element codes to incorporate NURBS and T-spline easily.

2.1 B-splines

2.1.1 Knot vector

A knot vector is a set of non-decreasing real numbers in the parametric space:

$$\{\xi_1, \xi_{2,} \cdots, \xi_{n+p+1}\} \qquad \xi_A \in \mathbb{R},$$

where A denotes the knot index, p the curve order, and n the number of basis functions or control points. Each real number ξ_A is called a knot. The number of knots in a valid knot vector is always $n + p + 1$. The half open interval $[\xi_i, \xi_{i+1})$ is called a knot span. See Fig. 2.1.

Within the knot vector, knots can be repeated. For example, $\{0, 0, 0, 1, 1, 2, 2, 3, 3, 3\}$ is a valid knot vector. The knots with different values can be viewed as differ-

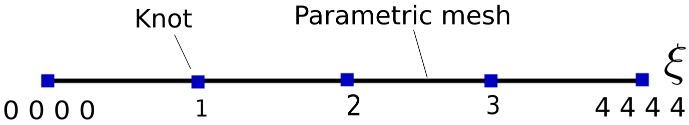


Figure 2.1: Knot vector

ent break points which divide the one-dimensional parametric space into different elements. Hence, the physical interpretation of the knots can be explained as the parametric coordinates of the element edges, while the "knot span" between two knots with different values can be viewed as the definition of elements in the parametric space. The insertion of a new knot will split an element, much like h-refinement in FEM. However, the repetition of existing knots will not increase the number of elements, but can be used to decrease the order of the basis functions. For example, the knot vector $\{0,0,0,1,1,2,2,3,3,3\}$ has 10 knot values and 9 knot spans, $[0,0),[0,0),[0,1),[1,1),[1,2),[2,2),[2,3),[3,3),[3,3)$, but only 3 elements, $[0,1],[1,2],[2,3]$.

It is called open knot vector if its first and last knot values are repeated $p+1$ times, such as $\{0,0,0,1,2,3,4,4,4\}$ for the NURBS with $p=2$. The open knot vector is the standard in CAD, so all the examples in the present work use open knot vectors. The knot vector values can be normalized without affecting the resulting B-splines. Therefore $\{0,0,0,1,2,3,4,4,4\}$ is equivalent to $\{0,0,0,1/4,2/4,3/4,1,1,1\}$. It is called a uniform knot vector if the knots are uniformly spaced, for example, $\{0,0,0,1,2,3,4,5,5,5\}$.

2.1.2 B-spline basis functions

With the concept of a knot vector, we can now define B-spline basis functions using the **Cox-de Boor recursion formula** [36, 39]

$$N_{A,0}(\xi) = \begin{cases} 1 & \text{if } \xi_A \leqslant \xi < \xi_{A+1}, \\ 0 & \text{otherwise,} \end{cases} \tag{2.1.1}$$

$$N_{A,p}(\xi) = \frac{\xi - \xi_A}{\xi_{A+p} - \xi_A} N_{A,p-1}(\xi) + \frac{\xi_{A+p+1} - \xi}{\xi_{A+p+1} - \xi_{A+1}} N_{A+1,p-1}(\xi). \tag{2.1.2}$$

In essence a B-spline basis function is a piecewise polynomial function. The function are C^∞ within elements and C^{p-m} on element boundaries, where m is the number of knot repetitions. B-spline basis functions possess the following properties (Fig. 2.2):

- Local support. The B-spline basis function $N_{A,p}$ is always non-negative in the knot span of $[\xi_A, \xi_{A+p+1})$. This is significant for interactive design: the change of one control point only affects the local part of the curve, providing flexibility in curve modifications.
- Partition of unity. $\sum_{A=1}^{n} N_{A,p}(\xi) = 1$.
- Pointwise non-negativity.
- Weak Kronecker delta property. A weak *Kronecker delta* property means $N_A(\mathbf{x}) = 0$ but $N_A(\mathbf{x}_A) \neq 1$, which is useful for enforcing boundary conditions in engineering analysis, because only the control points corresponding to boundaries need to be considered.
- Linear independence. This property is essential to construct the approximation space for numerical analysis.

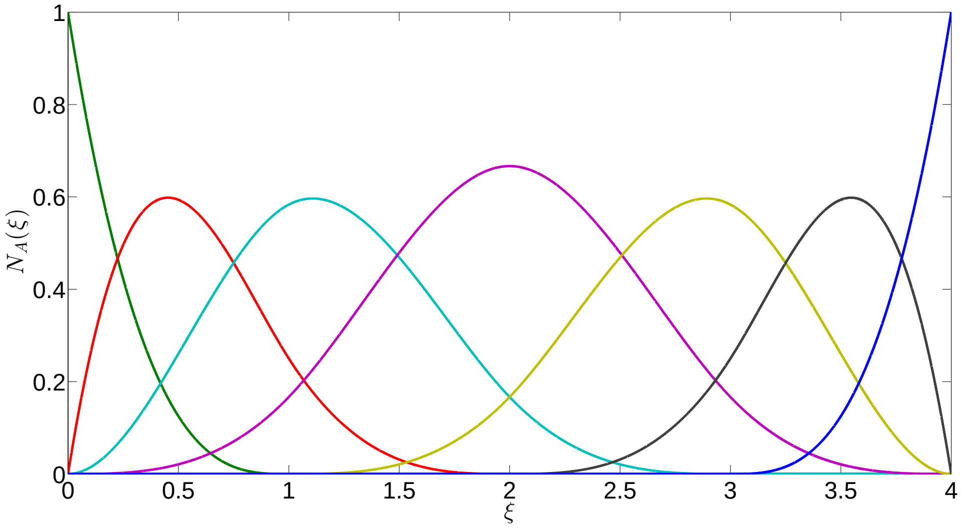


Figure 2.2: B-spline basis functions ($p = 3$) for knot vector $\{0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4\}$

The first order derivative of the B-spline basis function is

$$\frac{\mathrm{d}}{\mathrm{d}\xi}N_{A,p}(\xi) = \frac{p}{\xi_{A+p} - \xi_A}N_{A,p-1}(\xi) - \frac{p}{\xi_{A+p+1} - \xi_{A+1}}N_{A+1,p-1}(\xi). \tag{2.1.3}$$

The kth order derivatives of the B-spline basis function is given by

$$\begin{aligned}\frac{\mathrm{d}^k}{\mathrm{d}^k\xi}N_{A,p}(\xi) = &\frac{p}{\xi_{A+p} - \xi_A}\left(\frac{\mathrm{d}^{k-1}}{\mathrm{d}^{k-1}\xi}N_{A,p-1}(\xi)\right)\\ &-\frac{p}{\xi_{A+p+1} - \xi_{A+1}}\left(\frac{\mathrm{d}^{k-1}}{\mathrm{d}^{k-1}\xi}N_{A+1,p-1}(\xi)\right).\end{aligned} \tag{2.1.4}$$

In the implementation, an iterative algorithm exists to expand $\frac{\mathrm{d}^k}{\mathrm{d}^k\xi}N_{A,p}(\xi)$ in terms of low order basis functions as follows

$$\frac{\mathrm{d}^k}{\mathrm{d}^k\xi}N_{A,p}(\xi) = \frac{p!}{(p-k)!}\sum_{j=0}^{k}\alpha_{k,j}N_{A+j,p-k}(\xi), \tag{2.1.5}$$

with

$$\begin{aligned}\alpha_{0,0} &= 1,\\ \alpha_{k,0} &= \frac{\alpha_{k-1,0}}{\xi_{A+p-k+1} - \xi_A},\\ \alpha_{k,j} &= \frac{\alpha_{k-1,j} - \alpha_{k-1,j-1}}{\xi_{A+p+j-k+1} - \xi_{A+j}}, \quad j = 1, \ldots, k-1,\\ \alpha_{k,k} &= \frac{-\alpha_{k-1,k-1}}{\xi_{A+p+1} - \xi_{A+k}}.\end{aligned} \tag{2.1.6}$$

B-spline geometries

A B-spline geometry is a mapping from parametric space to physical space through a linear combination of B-spline basis functions, which are defined in parametric space, and the corresponding coefficients are called control points because their physical meaning is a series of points scattered in physical space. A B-spline curve can be expressed as

$$\mathbf{x}(\xi) = \sum_{A}^{n} N_{A,p}(\xi)\mathbf{P}_A, \tag{2.1.7}$$

where $\mathbf{x}(\xi)$ denotes the physical curve of interest, ξ is the spatial coordinate in parametric space, $\mathbf{P}_A$ the control points, $N_{A,p}$ the B-spline basis functions of order p. See Fig. 2.3.

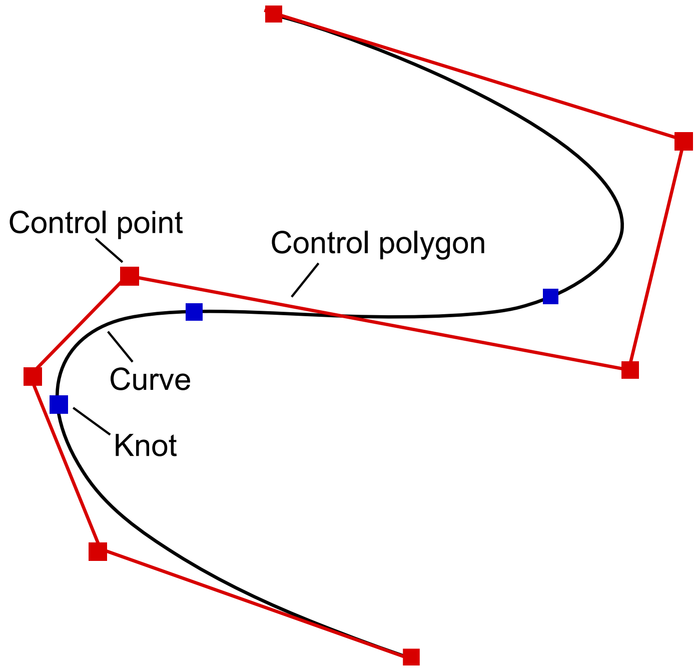


Figure 2.3: B-spline curve

To construct B-spline surfaces and solids, the basis functions can be obtained from the tensor product. B-spline surface basis function is given by

$$N_A(\boldsymbol{\xi}|\boldsymbol{\Xi}_A) \equiv \prod_{i=1}^{d_p} N_A^i(\xi_A^i|\Xi_A^i). \tag{2.1.8}$$

where i denotes the direction index, d_p is the number of dimensions.

So B-spline surface basis function is

$$N_{A,B}^{p,q}(\xi,\eta) = \sum_{A=1}^{n}\sum_{B=1}^{m} N_{A,p}(\xi)M_{B,q}(\eta), \tag{2.1.9}$$

and solid basis function is

$$N_{A,B,C}^{p,q,r}(\xi,\eta,\zeta) = \sum_{A=1}^{n}\sum_{B=1}^{m}\sum_{C=1}^{l} N_{A,p}(\xi)M_{B,q}(\eta)L_{C,r}(\zeta). \tag{2.1.10}$$

The continuity and differentiability of a B-spline curve is inherited directly from its basis functions and the continuity of a B-spline curve is at least C^{p-m}.

B-spline geometries possess the following properties:

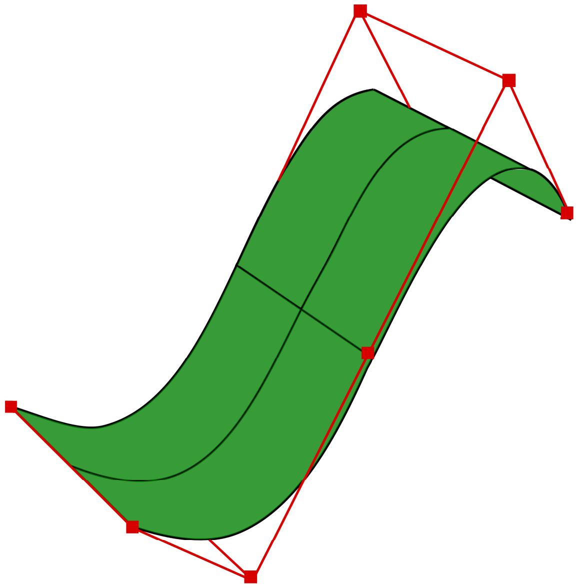


Figure 2.4: B-spline surface

- The convex hull property. The B-splines geometry is contained in the convex hull constructed by the control grid, which is a mesh interpolated by control points. See Fig. (2.4).

- The variation diminishing property. No plane has more intersections with the curve than it has with the control grid. This property renders B-splines less oscillatory than Lagrangian polynomials.

- The transformation invariance property. An affine transformation of a B-splines curve can be achieved by applying an affine transformation to the control points.

- Non-interpolatory. The B-spline geometry does not interpolate the control points except at the starting point of the curve, the end point of the curve and any point whose knot value is repeated p times.

2.1.3 Knot insertion in B-splines

Knot insertion algorithm is used to enrich the basis function space without changing the geometry. For a given knot vector $\Xi = \{\xi_1, \xi_2, \cdots, \xi_{n+p+1}\}$, a new knot

$\bar{\xi} \in [\xi_k, \xi_{k+1}]$ can be inserted, leading to a modification of control points as

$$\bar{\mathbf{P}}_A = \begin{cases} \mathbf{P}_1 & A = 1, \\ \alpha_A \mathbf{P}_A + (1 - \alpha_A)\mathbf{P}_{A-1} & 1 < A < m, \\ \mathbf{P}_n & A = m, \end{cases} \tag{2.1.11}$$

with

$$\alpha_A = \begin{cases} 1 & 1 \leqslant A \leqslant k - p, \\ \frac{\bar{\xi}_A - \xi_A}{\xi_{A+p} - \xi_A} & k - p + 1 \leqslant A \leqslant k, \\ 0 & A \geqslant k + 1, \end{cases} \tag{2.1.12}$$

where $\mathbf{P}$ denotes the control points corresponding to the initial knot vector, and $\bar{\mathbf{P}}$ the added control point. The existing knot values can also be repeated in this algorithm, thereby decreasing the the basis continuity. However, continuity of the geometry is preserved by choosing the control points using Eq. (2.1.11).

2.2 NURBS

2.2.1 NURBS basis functions

Non-uniform rational B-splines (NURBS) [79] are developed from B-splines but can offer significant advantages due to their ability to represent a wide variety of geometric entities such as conic sections. NURBS are an important geometric modelling technique in CAD and are seen as the industry standard with implementation in several commercial software packages. Therefore, all the geometries in the two-dimensional case in the present work are represented by NURBS. The expression defining NURBS curve is very similar to that of B-splines,

$$\mathbf{x}(\xi) = \sum_{A=1}^{n} R_{A,p}(\xi)\mathbf{P}_A. \tag{2.2.13}$$

Here, $\mathbf{P}_A$ is the set of control point coordinates, $R_{A,p}$ are NURBS basis functions, defined as

$$R_{A,p}(\xi) = \frac{N_{A,p}(\xi)w_A}{W(\xi)}, \tag{2.2.14}$$

with

$$W(\xi) = \sum_{A=1}^{n} w_A N_{A,p}(\xi), \tag{2.2.15}$$

where w_A denotes a weight associated to each basis function or control point. It can influence the distance between the associated control point and the NURBS geometry, with higher values drawing the curve closer to that point (Fig. 2.5). When all of the weights are equal to 1, the NURBS curve reduces to a B-spline curve.

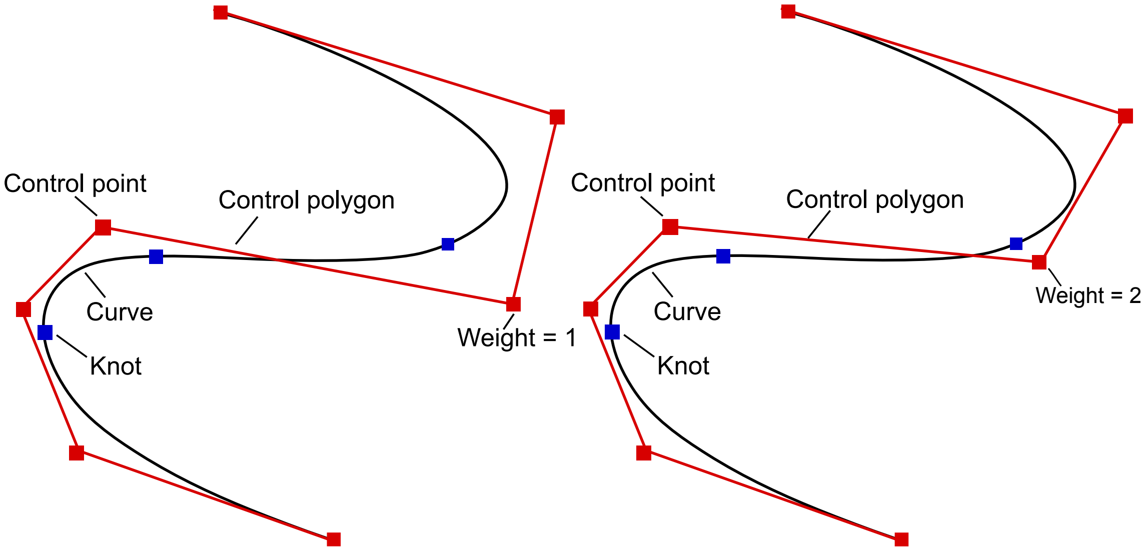


Figure 2.5: The comparison between NURBS curve with different weights

The derivatives of a NURBS basis function are expressed by

$$\frac{\mathrm{d}}{\mathrm{d}\xi}R_{A,p}(\xi) = w_A \frac{W(\xi)\frac{\mathrm{d}}{\mathrm{d}\xi}N_{A,p}(\xi) - \frac{\mathrm{d}}{\mathrm{d}\xi}W(\xi)N_{A,p}(\xi)}{(W(\xi))^2}, \tag{2.2.16}$$

and

$$\frac{\mathrm{d}}{\mathrm{d}\xi}W(\xi) = \sum_{A=1}^{n} \frac{\mathrm{d}}{\mathrm{d}\xi}N_{A,p}(\xi)w_A. \tag{2.2.17}$$

An important interpretation of NURBS geometries from a different perspective is a linear combination of standard B-spline basis functions and weighted control points

$$\mathbf{x}(\xi) = \sum_{A=1}^{n} N_{A,p}(\xi)\tilde{\mathbf{P}}_A, \tag{2.2.18}$$

where $\tilde{\mathbf{P}}_A = \{w_A\mathbf{P}_A, w_A\}^{\mathrm{T}}$ are the weighted control points in projective space. $N_{A,p}$ is the standard B-spline basis function.

2.2.2 The property of NURBS geometries

A NURBS basis in multi-dimensions can be obtained using tensor product as

$$R_A(\boldsymbol{\xi}|\Xi_A) \equiv \prod_{i=1}^{d_p} R_A^i(\xi_A^i|\Xi_A^i), \tag{2.2.19}$$

where i denotes the direction index and d_p is the dimension number. Hence NURBS basis functions in two-dimensions and three-dimensions are written as

$$R_{A,B}^{p,q}(\xi,\eta) = \frac{N_{A,p}(\xi)M_{B,q}(\eta)w_{A,B}}{\sum_{\hat{A}=1}^{n}\sum_{\hat{B}=1}^{m} N_{\hat{A},p}(\xi)M_{\hat{B},q}(\eta)w_{\hat{A},\hat{B}}}, \tag{2.2.20}$$

$$R_{A,B,C}^{p,q,r}(\xi,\eta,\zeta) = \frac{N_{A,p}(\xi)M_{B,q}(\eta)L_{C,r}(\zeta)w_{A,B,C}}{\sum_{\hat{A}=1}^{n}\sum_{\hat{B}=1}^{m}\sum_{\hat{C}=1}^{l} N_{\hat{A},p}(\xi)M_{\hat{B},q}(\eta)L_{\hat{C},r}(\zeta)w_{\hat{A},\hat{B},\hat{C}}}. \tag{2.2.21}$$

NURBS inherit the aforementioned properties of B-splines, but still have some drawbacks:

Rational functions As NURBS are not polynomial functions, integrating them cannot be done exactly using Gauß quadrature.

Tensor product The parametric space and control points rely on a structured grid due to the tensor product property of NURBS and thus does not allow local refinement, which increases the redundancy of the degrees of freedom (Fig. 2.7).

Continuity For complex geometry, NURBS normally need multiple patches, each patch associated with a parametric space (Fig. 2.6). NURBS usually achieve only C^0 continuity between the patches.

Geometry repair From a computational geometry point of view, a NURBS based geometry always requires some level of repair due to gaps or overlaps of the various patches making up the geometry.

2.2.3 Knot insertion in NURBS

As in B-splines, knot insertion is used to enrich NURBS basis function space. Let $\Xi = \{\xi_1, \xi_2, \cdots, \xi_{n+p+1}\}$ be a knot vector, $\tilde{\mathbf{P}}$ the corresponding weighted control

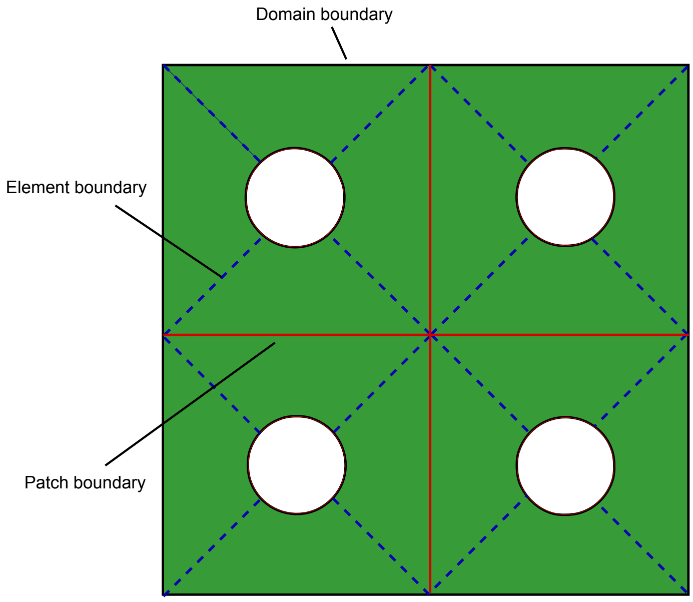


Figure 2.6: NURBS plane with multiple patches

points. If we insert a new knot $\bar{\xi} \in [\xi_k, \xi_{k+1}]$, the added control point $\bar{\tilde{\mathbf{P}}}$ can be obtained as follows without changing the geometry,

$$\bar{\tilde{\mathbf{P}}}_A = \begin{cases} \tilde{\mathbf{P}}_1 & A = 1, \\ \alpha_A \tilde{\mathbf{P}}_A + (1 - \alpha_A)\tilde{\mathbf{P}}_{A-1} & 1 < A < m, \\ \tilde{\mathbf{P}}_n & A = m, \end{cases} \tag{2.2.22}$$

with

$$\alpha_A = \begin{cases} 1 & 1 \leqslant A \leqslant k - p, \\ \frac{\bar{\xi}_A - \xi_A}{\xi_{A+p} - \xi_A} & k - p + 1 \leqslant A \leqslant k, \\ 0 & A \geqslant k + 1. \end{cases} \tag{2.2.23}$$

2.2.4 Element structure of NURBS

Knot vectors used to define NURBS basis functions provide natural element structures which are very useful for numerical analysis using FEM or BEM. We can view a non-zero knot interval as an element in each dimension. To employ the Gauss-Legendre quadrature rule, we can transfer the space Ω_e defined on each knot interval into a standard $[-1, +1]$ space $\hat{\Omega}$, and $\mathrm{d}\Omega_e = J_e \mathrm{d}\hat{\Omega}_e$, with

$$J_e = \frac{\xi_b - \xi_a}{2}, \tag{2.2.24}$$

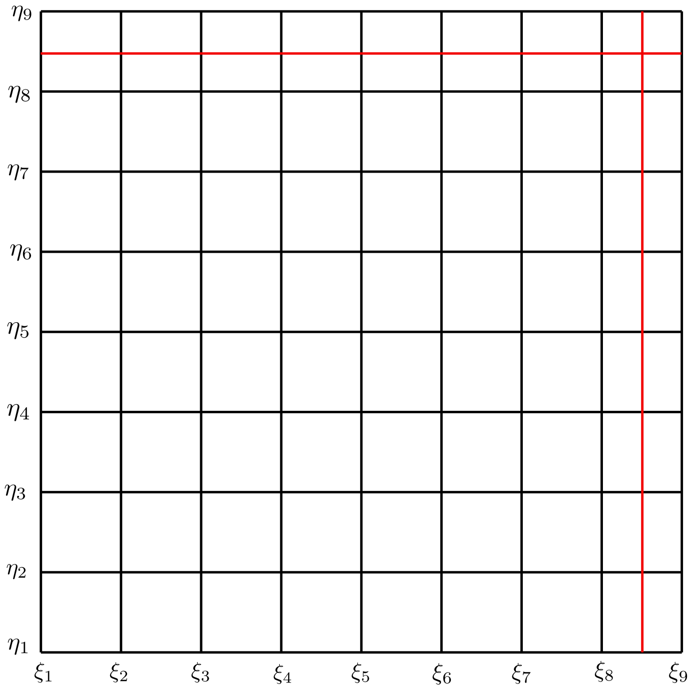


Figure 2.7: NURBS refinement

where ξ_a and ξ_b are the parametric coordinates of the starting knot and end knot of the element, respectively. J_e is the Jacobian which maps parent elements to parametric elements.

2.3 T-splines

T-splines were proposed by Sederberg *et al.* [88] to overcome the drawbacks of NURBS. The main advantage of T-splines is that it can construct water-tight geometry and allows local refinement. The application of T-splines to the structural analysis can be seen in the work of Bazilevs *et al.* [7]. A posteriori error estimation for local h-refinement with T-splines is given by Dorfel *et al.* [41].

2.3.1 T-mesh

A T-spline control mesh, or called T-mesh, is distinct from a NURBS control mesh in that it has T-junctions, which are similar to the concept of "hanging nodes" and oct/quad-tree meshes in the FEM. See Fig. 2.8. If a T-mesh is simply a rectangular grid with no T-junctions, T-splines reduce to B-splines. A T-junction with a number of edges $N \neq 4$ is called an extraordinary point. For simplicity, herein we only illustrate T-mesh without extraordinary point. The interested reader is referred to [87] for more details.

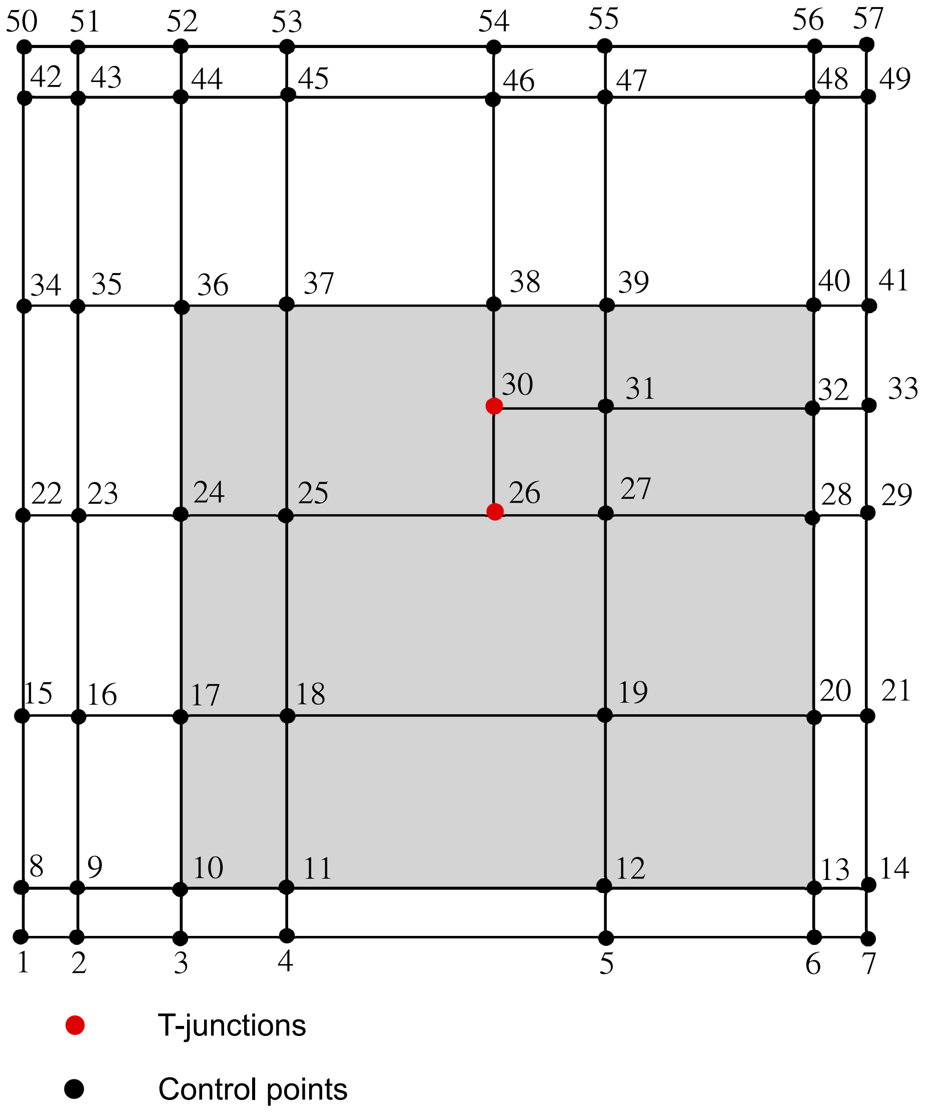


Figure 2.8: T-mesh and T-junctions

2.3.2 Local knot interval vectors and T-spline basis

Instead of sharing a global knot vector like NURBS, each control point in T-mesh is associated with a local knot vector in each dimension

$$\Xi_A = \{\xi_1, \xi_2, \cdots, \xi_{p+1}\}, \tag{2.3.25}$$

where A is the global index of the control point in T-mesh, ξ the knot in local knot vector, and p the order of T-splines. The local knot vector in multiple directions can be collected as

$$\boldsymbol{\Xi}_A = \{\Xi_A\}_{i=1}^{d_p}, \tag{2.3.26}$$

where i denotes the direction index, and d_p the dimension of the geometry.

Now we define a knot interval vector as

$$\Delta\Xi = \{\Delta\xi_1, \Delta\xi_2, \cdots, \Delta\xi_{p+1}\}, \tag{2.3.27}$$

and its vector form in multiple directions is

$$\boldsymbol{\Delta\Xi}_A = \{\Delta\Xi_A^i\}_{i=1}^{d_p}. \tag{2.3.28}$$

Remark that basis functions can be determined by knot intervals vectors. The local knot interval vectors are derived from a T-mesh, where a knot interval configuration has been predefined (Fig. 2.9). A valid knot interval configuration requires that the knot intervals on opposite sides of every element sum to the same value.

To infer the local knot vectors of vertex A, we start at the vertex A and march through the T-mesh in each direction, until $p-1$ vertices or perpendicular edges are intersected. The knot intervals assigned to the transversed edges are added to the local knot interval vector. If a T-mesh boundary is crossed before $p-1$ knot intervals are encountered, knot intervals of zero are appended to complete the knot interval vector. Fig. 2.10 illustrates the inference of knot interval vector associated with $\mathbf{P}_{26}$ in the T-mesh shown in Fig. 2.8. The knot interval vectors of the control point $\mathbf{P}_{26}$ are $\Delta\Xi_{26}^1 = \{0.5,\ 1, 0.5,\ 1\}$, $\Delta\Xi_{26}^2 = \{1,\ 1,\ 0.5,\ 0.5\}$, or collected together as

$$\boldsymbol{\Delta\Xi}_{26} = \begin{bmatrix} 0.5,\ 1, 0.5,\ 1 \\ 1,\ 1,\ 0.5,\ 0.5 \end{bmatrix}. \tag{2.3.29}$$

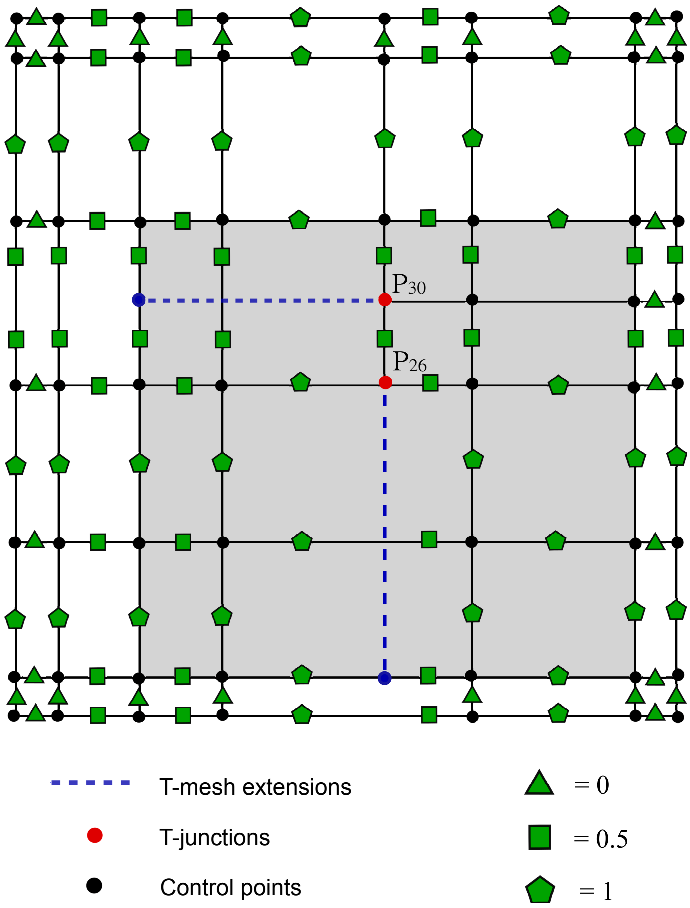


Figure 2.9: T-mesh with knot interval configuration. The triangles denote knot intervals of 0, the squares knot denote intervals of $\frac{1}{2}$, and the pentagons denote knot intervals of 1.

By setting the origin of the local knot interval vector to be zero, the local knot vector can obtained readily. Then, a local coordinative system is constructed. The local basis function domain in each dimension is

$$\hat{\Omega}^i_A = [0, \xi_{p+1}]. \tag{2.3.30}$$

And the local basis function domain in multi-dimensional cases are

$$\hat{\Omega}_A = \bigotimes_{i=1}^{d_p} \hat{\Omega}^i_A. \tag{2.3.31}$$

The shaded area in Fig. 2.10 shows the the local basis function domain associated to the T-junction $\mathbf{P}_{26}$. The local parametric space and the local coordinate system are shown in Figs. 2.11a and 2.11b.

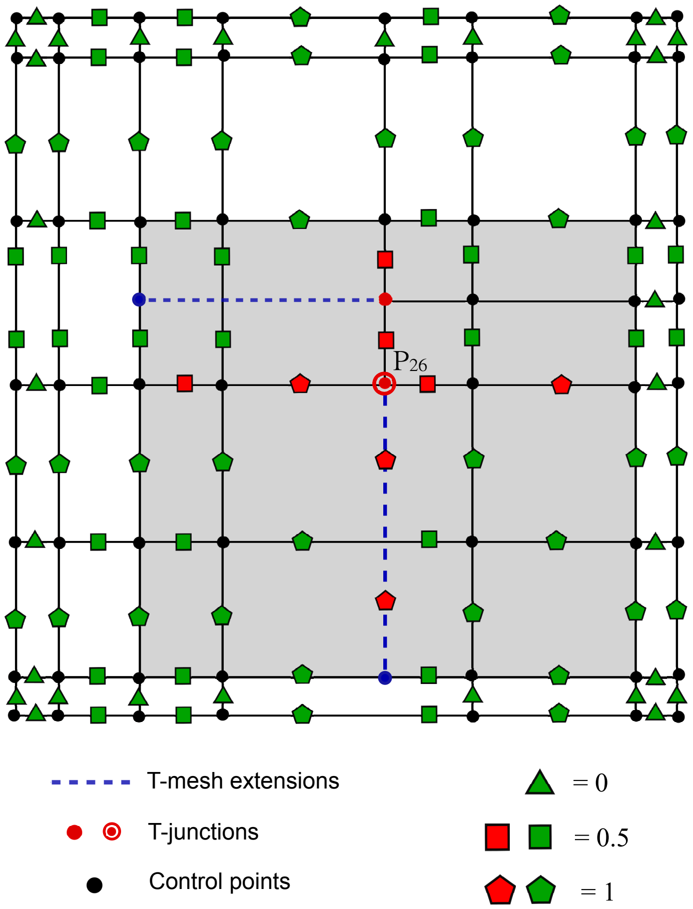


Figure 2.10: Knot interval vector inference. The triangles denote knot intervals of 0, the squares denote knot intervals of $\frac{1}{2}$, and the pentagons denote knot intervals of 1.

Based on the local parametric space, T-spline basis functions on the ith dimension can be formulated using an iterative formula similar to B-splines

$$N_A^i = \begin{cases} 1 & \text{if}\, \xi_{A,1}^i \leqslant \xi_A^i < \xi_{A,2}^i \\ 0 & \text{otherwise}, \end{cases} \tag{2.3.32}$$

$$N_A^i = \frac{\xi_A^i - \xi_{A,1}^i}{\xi_{A,p+1}^i - \xi_{A,1}^i} N_A + \frac{\xi_{A,p+2}^i - \xi_A^i}{\xi_{A,p+2}^i - \xi_{A,2}^i} N_A. \tag{2.3.33}$$

The multivariate T-spline basis function can be obtained using tensor product

$$N_A(\boldsymbol{\xi}|\Xi_A) \equiv \prod_{i=1}^{d_p} N_A^i(\xi_A^i|\Xi_A^i). \tag{2.3.34}$$

To obtain a rational form for T-spline basis functions, a weight can be assigned to each control point, in the same way as when extending B-splines to NURBS.

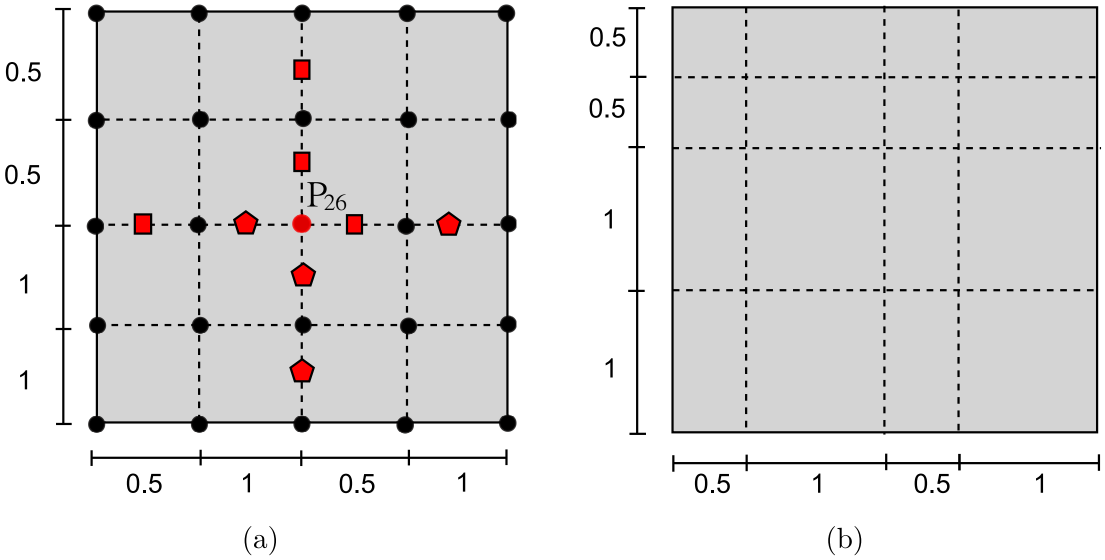


Figure 2.11: (a) T-spline local basis function mesh, and (b) T-spline local coordinate system

2.3.3 Element structure of T-splines

We further explore the element structure for the facility of numerical analysis. T-spline elements are formed by adding T-spline extensions to the T-mesh (Fig. 2.12).

The coordinate $\boldsymbol{\xi}^e$ in the local system of each element in the extended T-mesh can be mapped from a parent element using $\tilde{\Phi}^e : \tilde{\Omega} \to \hat{\Omega}^e$ as

$$\boldsymbol{\xi}^e = \tilde{\Phi}^e(\tilde{\boldsymbol{\xi}}). \tag{2.3.35}$$

Then a mapping $\hat{\Phi}^e_a : \hat{\Omega}^e \to \hat{\Omega}_A$ can map local coordinates from the element domain to the local basis function domain as

$$\boldsymbol{\xi}_A = \hat{\Phi}^e_a(\boldsymbol{\xi}^e). \tag{2.3.36}$$

Hence, the definition of T-splines basis function is localized to parent elements

$$N_A(\boldsymbol{\xi}_A)|_e = N_A(\hat{\Phi}^e_a(\boldsymbol{\xi}^e))|_e = N_A(\hat{\Phi}^e_a(\tilde{\Phi}^e(\tilde{\boldsymbol{\xi}})))|_e = N^e_a(\tilde{\boldsymbol{\xi}}), \tag{2.3.37}$$

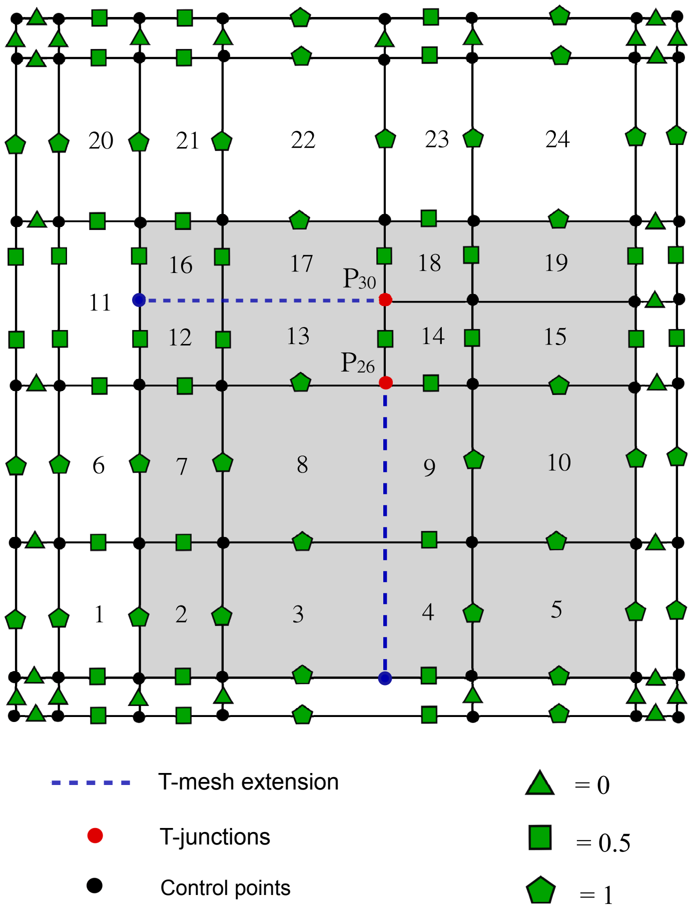


Figure 2.12: T-spline elements. The triangles denote knot intervals of 0, the squares denote knot intervals of $\frac{1}{2}$, and the pentagons denote knot intervals of 1.

T-splines inherit most of merits of NURBS, but T-spline basis functions cannot always guarantee the linear independences, which is an indispensable requirement for subsequent numerical analysis. Analysis-suitable T-splines [61], a large subset of T-splines, satisfy this requirement. For T-meshes without extraordinary points, an analysis-suitable T-spline is defined to be one whose T-mesh has no intersecting extension. In the present work, all the three-dimensional models utilize analysis-suitable T-splines.

2.4 Bézier extraction

Although NURBS and T-splines have intrinsic element structures, the set of the basis functions supported by each element are different. To further integrate

IGA with existing FEM codes, the Bézier extraction technique was introduced, first for NURBS in [23] and then T-splines in [86]. The idea of Bézier extraction is that localized NURBS or T-spline basis functions can be represented by a linear combination of Bernstein polynomials. Bézier extraction provides an element data structure suitable for analysis. That is, similar to Lagrangian polynomial elements in traditional FEM, Bernstein bases do no change from element to element.

2.4.1 Bézier extraction of B-splines

The mechanism underlying Bézier extraction is to replicate the existing knots using the knot insertion algorithm until their multiplicity is equal to the order p, thus subdividing the geometry into Bézier elements. Assuming the knot vector of the initial geometry is $\Xi = \{\xi_1, \xi_2, \cdots, \xi_{n+p+1}\}$ with control points, the added control points $\bar{\mathbf{P}}$ after knot insertion are given by

$$\bar{\mathbf{P}} = (\mathbf{C}^1)\mathbf{P}, \tag{2.4.38}$$

where $\mathbf{C}^1$ is a matrix form representing the linear map defined in Eq. (2.1.11) while keeping the geometry the same. If the knots are replicated multiple times m, then

$$\bar{\mathbf{P}}^{m+1} = \mathbf{C}^{\mathrm{T}}\mathbf{P}, \tag{2.4.39}$$

where $\mathbf{C}^{\mathrm{T}} = (\mathbf{C}^m)^{\mathrm{T}}(\mathbf{C}^{m-1})^{\mathrm{T}} \cdots (\mathbf{C}^1)^{\mathrm{T}}$. If $m = p$, a Bézier curve is obtained, and $\mathbf{C}$ is called the Bézier extraction operator. Hence, the geometry can be expressed by the Bézier basis functions $\mathbf{B}(\xi)$

$$\begin{aligned}\mathbf{x}(\xi) &= (\bar{\mathbf{P}}^{m+1})^{\mathrm{T}}\mathbf{B}(\xi)\\ &= (\mathbf{C}^{\mathrm{T}}\mathbf{P})^{\mathrm{T}}\mathbf{B}(\xi)\\ &= \mathbf{P}^{\mathrm{T}}\mathbf{C}\mathbf{B}(\xi)\end{aligned} \tag{2.4.40}$$

After the rearrangement above, the B-spline basis functions can be represented by the linear combination of the Bézier basis $\mathbf{B}(\xi)$ and Bézier extraction operator $\mathbf{C}$,

$$\mathbf{N}(\xi) = \mathbf{C}\mathbf{B}(\xi). \tag{2.4.41}$$

The Bézier basis $\mathbf{B}(\xi)$ is also called Bernstein basis, defined as

$$B_{i,p}^{k}(\xi^{k}) = \frac{1}{2^{p}} \left(\begin{array}{c} p \\ i-1 \end{array} \right) (1-\xi^{k})^{p-(i-1)}(1+\xi^{k})^{i-1}, \tag{2.4.42}$$

where

$$\left(\begin{array}{c} p \\ i-1 \end{array} \right) = \frac{p!}{(i-1)!(p+1-i)!}, \quad 1 \leqslant i \leqslant p+1. \tag{2.4.43}$$

And Bernstein basis satisfies the following properties

- Partition of unity

$$\sum_{i=1}^{p+1} B_{i,p}^{k}(\xi^{k}) = 1. \tag{2.4.44}$$

- Pointwise non-negativity

$$B_{i,p}^{k}(\xi^{k}) \geqslant 0. \tag{2.4.45}$$

- Endpoint interpolation

$$B_{1,p}^{k}(-1) = B_{p+1,p}^{k}(1) = 1. \tag{2.4.46}$$

- Symmetry

$$B_{i,p}^{k}(\xi^{k}) = B_{p+1-i,p}^{k}(-\xi^{k}). \tag{2.4.47}$$

- Linear independence

It is noteworthy that Bézier extraction operator is only determined by the knot vector, independent on the positions of control points.

2.4.2 Bézier extraction of NURBS

The Eq. (2.4.41) can be used to extend Bézier extraction from B-splines to NURBS. Now writing Eqs. (2.2.14) and (2.2.15) in matrix form as

$$\mathbf{R}(\xi) = \frac{1}{\mathbf{w}^{\mathrm{T}}\mathbf{N}(\xi)}\mathbf{W}\mathbf{N}(\xi), \tag{2.4.48}$$

where $\mathbf{W}$ is the diagonal matrix of weights, and $\mathbf{N}$ is the matrix form of B-spline basis. Now we can write NURBS in terms of Bernstein basis as

$$\mathbf{x}(\xi) = \mathbf{P}^{\mathrm{T}}\mathbf{R}(\xi)$$

$$\begin{aligned} &= \frac{1}{\mathbf{w}^{\mathrm{T}}\mathbf{N}(\xi)}\mathbf{P}^{\mathrm{T}}\mathbf{W}\mathbf{N}(\xi) \\ &= \frac{1}{\mathbf{w}^{\mathrm{T}}\mathbf{C}\mathbf{B}(\xi)}\mathbf{P}^{\mathrm{T}}\mathbf{W}\mathbf{C}\mathbf{B}(\xi) \\ &= \frac{1}{\mathbf{w}^{\mathrm{T}}\mathbf{C}\mathbf{B}(\xi)}(\mathbf{C}^{\mathrm{T}}\mathbf{W}\mathbf{P})^{\mathrm{T}}\mathbf{B}(\xi). \end{aligned} \tag{2.4.49}$$

The above equation has represented NURBS with Bézier elements.

2.4.3 Bézier extraction of T-splines

The T-spline extraction is similar to B-splines, but because each control point has a local knot vector, the extraction needs to be performed in the parent element which is shared by the corresponding T-spline basis functions, *i.e.*

$$\mathbf{N}^e(\tilde{\xi}) = \mathbf{C}^e\mathbf{B}(\tilde{\xi}). \tag{2.4.50}$$

And the rational form is

$$\begin{aligned} \mathbf{R}^e(\xi) &= \frac{1}{(\mathbf{w}^e)^{\mathrm{T}}\mathbf{N}^e(\tilde{\xi})}\mathbf{W}^e\mathbf{N}^e(\tilde{\xi}) \\ &= \frac{1}{(\mathbf{w}^e)^{\mathrm{T}}\mathbf{C}^e\mathbf{B}^e(\tilde{\xi})}\mathbf{W}^e\mathbf{C}^e\mathbf{B}(\tilde{\xi}). \end{aligned} \tag{2.4.51}$$

A T-spline geometry can be written in terms of Bernstein basis function as follows

$$\begin{aligned} \mathbf{x}(\xi) &= (\mathbf{P}^e)^{\mathrm{T}}\mathbf{R}^e(\xi) \\ &= \frac{1}{(\mathbf{w}^e)^{\mathrm{T}}\mathbf{N}^e(\tilde{\xi})}(\mathbf{P}^e)^{\mathrm{T}}\mathbf{W}^e\mathbf{N}^e(\tilde{\xi}) \\ &= \frac{1}{(\mathbf{w}^e)^{\mathrm{T}}\mathbf{C}^e\mathbf{B}^e(\tilde{\xi})}(\mathbf{P}^e)^{\mathrm{T}}\mathbf{W}^e\mathbf{C}^e\mathbf{B}(\tilde{\xi}), \end{aligned} \tag{2.4.52}$$

where $\mathbf{N}$ is a vector of T-spline basis functions which are supported by element e, and $\mathbf{B}$ denotes Bernstein polynomial basis functions associated with Bézier element e.

2.5 Conclusions

Two important CAD modelling techniques, NURBS and T-splines, were reviewed. NURBS have been widely used and become the industry standard. On the other hand, as a recent advance in CAD, T-splines are capable of constructing watertight and locally-refined surfaces. We explained their element structures and Bézier extraction techniques for the purpose of incorporating NURBS and T-splines into the existing numerical analysis code.

Chapter 3

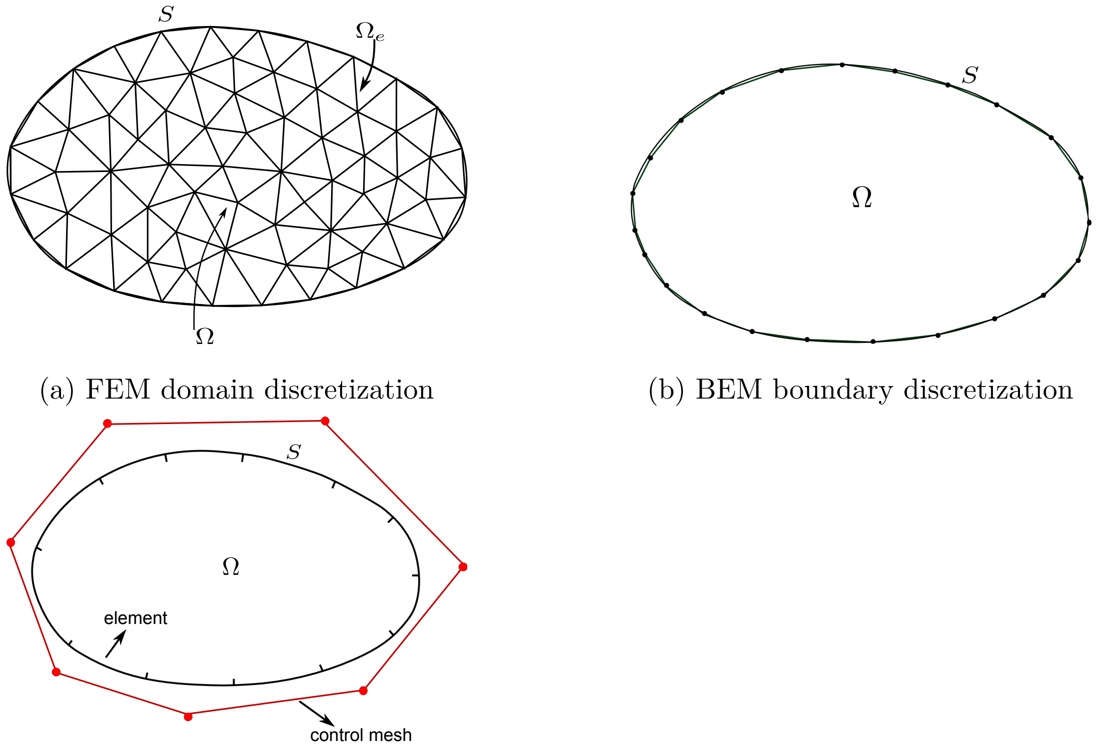
Isogeometric Boundary Element Methods

As mentioned previously, the main bottleneck of IGA is the conflict between boundary representations of geometric models in CAD and domain representations required by finite element analysis. An isogeometric approach using the framework of the boundary element method - coined the isogeometric boundary element method (IGABEM) - was proposed to solve this problem. The idea relies on the fact that both CAD models and boundary element methods rely on quantities defined entirely on the boundary. In the IGABEM, NURBS or T-spline basis functions are used to discretize boundary integral equations (BIE) for the surface geometry of the computational domain and boundary displacement and traction fields. The first implementations of this concept were published by Simpson *et al.* [93] and Politis *et al.* [80] for two dimensional linear elastostatic problems and exterior potential flow problems, respectively. Recently, IGABEM was extended to include T-splines for 3D linear elastostatic analysis in [87] and acoustic analysis [94] on complex geometries. In addition to tight integration of CAD and analysis, IGABEM inherits the benefits of the conventional boundary element method (BEM) [37, 38, 50, 83, 96] as follows:

Only the surface is discretized An obvious merit is that the dimension of the problem is decreased by one (see Fig. 3.1). This benefit is more manifest in simulating crack propagation, because only the new portion of the crack surface front needs to be meshed in each step of the crack propagation. It is also worth comparing the (IGA)BEM and the extended finite element method (XFEM) [9, 40] in the application of fracture mechanics. By introducing the asymptotic fields around the crack tip and the step functions across the crack body as the enrichment functions into the numerical solution, the XFEM allows for a mesh not conforming to the crack, thus avoiding the costly remeshing procedure during the crack propagation. To improve the stress accuracy around the crack tip, the high order terms of the asymptotic solutions were incorporated in the enrichment functions in the work of [54, 64, 103], where the stress intensity factors were also evaluated directly by constraining the enriched nodes. To further predict the coefficients associated with the high order terms accurately, a hybrid crack element was used for the region around the crack tip to couple with the XFEM (HCE-XFEM) [104]. The HCE-XFEM possesses the benefits of the XFEM, and also shares some similarities with BEM, that is, the integral is performed along the boundary of the crack element around the crack tip, which overcomes the singular integral difficulty in the classical XFEM.

More accurate stress Applications such as fracture mechanics and contact problems require significant stress accuracy, which cannot always be satisfied fully by FEM. (IGA)BEM is normally more accurate in stress evaluation because it is a semi-analytical method and tractions are, as displacements, primal unknowns in the equation.

Problems on infinite domains such as acoustics and electromagnetics In (IGA)FEM, the open domain around structures still needs to be meshed and boundary conditions at infinity are exerted approximately. In contrast, IGABEM requires no mesh generation since the control points provided by CAD can be directly used. BEM does require the generation of the boundary mesh for the structure, but no mesh is required outside of the structure, which, for infinite problems, is the majority of the domain. The boundary conditions can be exactly satisfied due to the nature of fundamental solutions.



(a) FEM domain discretization

(b) BEM boundary discretization

(c) IGABEM bounary discretization

Figure 3.1: The geometry discretizations of FEM, BEM and IGABEM

Analogously to the conventional BEM, however, the IGABEM has the following limitations:

- Non-linear problems. The (IGA)BEM needs a mesh for nonlinear regions thus losing the advantage of a boundary only approach. However, it is still desirable to apply BEM for the problems with “small” non-linear regions, for example, the small plasticity domains around crack tips and contact areas [35, 59].

- Heterogeneous materials. There is no existing available fundamental solution in this application.

- Dense matrix. Assembling and solving the dense matrix requires a huge amount of memory and slows down the solution process considerably.

- Unsymmetric matrix. This property increases the memory burden even further and prohibits certain iterative solvers. Symmetric Galerkin BEM [2, 20] can avoid this deficiency but with the expense of time cost.

In this chapter, the formulation of IGABEM will be explained. We will start with the BIEs and fundamental solutions in linear elasticity, followed by the formulation of classical BEM with isoparametric elements. Then we will detail the implementation of IGABEM, placing special emphasis on the differences between IGABEM and BEM. Finally, we will propose a regularized form for IGABEM, which is one of the contributions of the present thesis, to bypass the difficulty in the evaluation of jump terms and strongly singular integrals. In addition, we will also propose to use a nodal parameter extraction method to impose boundary conditions for its simplicity of implementation and convenience of combining with existing BEM codes.

3.1 Boundary integral equations

3.1.1 Fundamental solutions

In the linear elastic problem as shown in Fig. 3.2, the governing equations are expressed by

$$\sigma_{ij,j} + b_i = 0, \tag{3.1.1}$$

$$\sigma_{ij} = \lambda \delta_{ij} \epsilon_{kk} + 2\mu \epsilon_{ij}, \tag{3.1.2}$$

$$\epsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}), \tag{3.1.3}$$

where σ_{ij} are the components of the Cauchy stress tensor, ϵ_{ij} the strain tensor, b_i the body force, u_i the displacement field, and i, j are the indices running from 1 to 3 in three dimensions and from 1 to 2 in two dimensions. The comma implies a differentiation. $\lambda = 2\nu\mu/(1-2\nu)$ is a Lamé constant, and $\mu = E/2(1+\nu)$ is shear modulus of elasticity, with E denoting Young's modulus, and ν Poisson's ratio. δ_{ij} is the *Kronecker delta* symbol with the property

$$\delta_{ij} = \begin{cases} 0 & i \neq j, \\ 1 & i = j. \end{cases} \tag{3.1.4}$$

The boundary conditions are

$$u_i = \bar{u}_i \quad \text{on } S_u \subseteq S, \tag{3.1.5}$$

$$\sigma_{ij} n_j = \bar{t}_i \quad \text{on } S_t \subseteq S, \tag{3.1.6}$$

where $\bar{u}_i$ and $\bar{t}_i$ are the prescribed displacements and tractions, respectively.

Substituting Eq. (3.1.3) into Eq. (3.1.2) leads to the relationship between stress and displacement gradients

$$\sigma_{ij} = \frac{2\mu\nu}{1-2\nu} \delta_{ij} u_{m,m} + \mu(u_{i,j} + u_{j,i}). \tag{3.1.7}$$

Then substituting Eq. (3.1.7) into Eq. (3.1.1) yields the governing equation represented using displacement, or called Navier's equation

$$\mu u_{i,jj} - \frac{\mu}{2(1-\nu)} u_{j,ji} + b_i = 0. \tag{3.1.8}$$

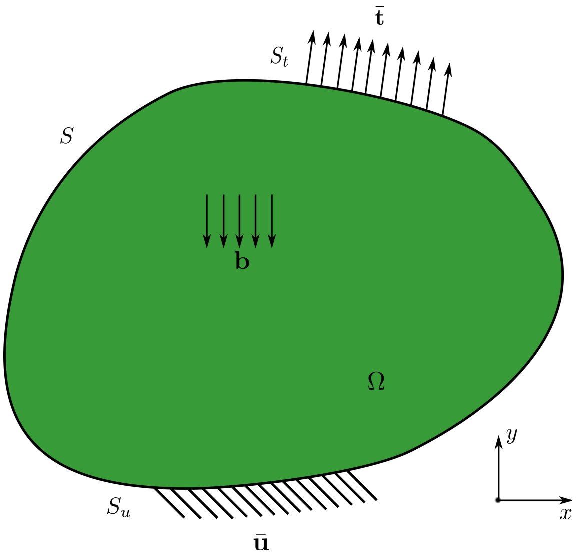


Figure 3.2: Two-dimensional elastic problem

For a body subject to a unit point force on $\mathbf{S}$, Eq. (3.1.8) can be rewritten as

$$\mu u_{i,jj} - \frac{\mu}{2(1-\nu)} u_{j,ji} + \delta(\mathbf{x}-\mathbf{S}) e_i = 0, \tag{3.1.9}$$

where e_i is a unit vector and $\delta(\mathbf{x}-\mathbf{S})e_i$ represents a unit point force exerted on $\mathbf{S}$ as

$$\delta(\mathbf{x}-\mathbf{S}) = \begin{cases} +\infty & \mathbf{x}=\mathbf{S}, \\ 0 & \mathbf{x}\neq\mathbf{S}. \end{cases} \tag{3.1.10}$$

Assuming the analytical displacement and traction solutions for the above equation are denoted by u_i^* and t_i^*, respectively. Then the fundamental solutions are defined as the tensors associated with them by

$$u_i^* = U_{ij}(\mathbf{S},\mathbf{x}) e_j, \tag{3.1.11}$$

$$t_i^* = T_{ij}(\mathbf{S},\mathbf{x}) e_j, \tag{3.1.12}$$

where U_{ij} is the displacement fundamental solution and T_{ij} the traction fundamental solution. To derive the form of fundamental solutions, we express the analytical displacement u_i^* using Galerkin vector as

$$u_i^* = G_{i,kk} - \frac{1}{2(1-\nu)} G_{k,ik}. \tag{3.1.13}$$

Substituting Eq. (3.1.13) into Eq. (3.1.9) leads to

$$\mu G_{i,kkjj} + \delta\left(\mathbf{x} - \mathbf{S}\right) e_i = 0. \tag{3.1.14}$$

For three dimensional problems, the Galerkin vector can be derived as

$$G_i = \frac{1}{8\pi\mu} r e_i, \tag{3.1.15}$$

where $r = r(\mathbf{x}, \mathbf{S}) = ||\mathbf{x} - \mathbf{S}||$ is the distance between the source point and field point, as shown in Fig. 3.3. Thus the solution of Eq. (3.1.9) is obtained by substituting Eq. (3.1.15) to (3.1.9) as

$$u_i^* = \frac{1}{8\pi\mu}\left[r_{,kk} e_i - \frac{1}{2(1-\nu)} r_{,ik} e_k\right]. \tag{3.1.16}$$

Then the displacement fundamental solutions can be written as

$$U_{ij}(\mathbf{S}, \mathbf{x}) = \frac{1}{16\pi\mu(1-\nu)r}[(3-4\nu)\delta_{ij} + r_{,i} r_{,j}]. \tag{3.1.17}$$

The analytical traction solution is obtained through substituting Eq. (3.1.7) and then using the relationship between stress and traction,

$$\begin{aligned} t_i^* = -\frac{1}{8\pi(1-\nu)r^2}\Big\{&\frac{\partial r}{\partial n}[(1-2\nu)\delta_{ij} + 3r_{,i}r_{,j}] \\ &+(1-2\nu)(n_i r_{,j} - n_j r_{,i})\Big\} e_j. \end{aligned} \tag{3.1.18}$$

Hence, the traction fundamental solution is expressed by

$$\begin{aligned} T_{ij}(\mathbf{S}, \mathbf{x}) = -\frac{1}{8\pi(1-\nu)r^2}\Big\{&\frac{\partial r}{\partial n}[(1-2\nu)\delta_{ij} + 3r_{,i}r_{,j}] \\ &+(1-2\nu)(n_i r_{,j} - n_j r_{,i})\Big\}. \end{aligned} \tag{3.1.19}$$

For two-dimensional elasticity, the Galerkin vector is written as

$$G_i = -\frac{1}{8\pi\mu} r^2 \ln(r) e_i. \tag{3.1.20}$$

Analogously to the derivation in three dimensional problems, the displacement and traction fundamental solutions in two dimensions are given by

$$U_{ij}(\mathbf{S}, \mathbf{x}) = \frac{1}{8\pi\mu(1-\nu)}\left[(3-4\nu)\ln\left(\frac{1}{r}\right)\delta_{ij} + r_{,i} r_{,j}\right], \tag{3.1.21}$$

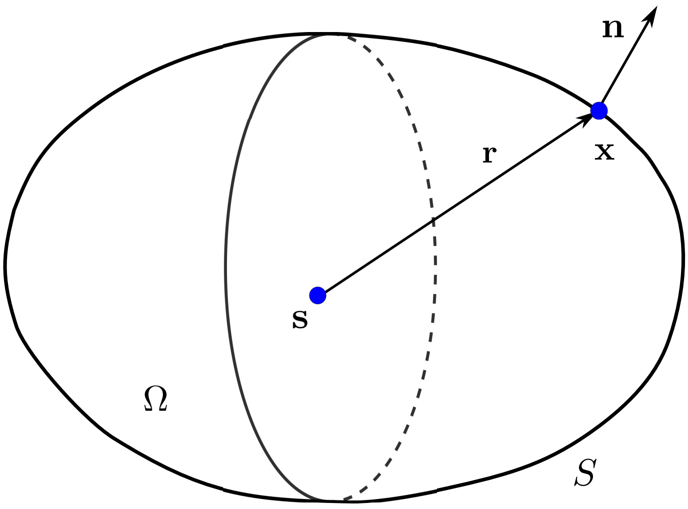


Figure 3.3: The distance between source point and field point

$$T_{ij}(\mathbf{S},\mathbf{x}) = -\frac{1}{4\pi(1-\nu)r}\Big\{\frac{\partial r}{\partial n}[(1-2\nu)\delta_{ij} + 2r_{,i}r_{,j}] - (1-2\nu)(r_{,i}n_j - r_{,j}n_i)\Big\}. \tag{3.1.22}$$

Taking the derivatives of fundamental solutions with respect to source points and using Hooke's law in Eq. (3.1.2), the hypersingular fundamental solutions can be obtained in two dimensions as

$$D_{kij} = \frac{1}{4\pi(1-\nu)r}[(1-2\nu)(r_{,i}\delta_{jk} + r_{,j}\delta_{ki} - r_{,k}\delta_{ij}) + 2r_{,i}r_{,j}r_{,k}], \tag{3.1.23}$$

$$\begin{aligned} S_{kij} = &\frac{\mu}{2\pi(1-\nu)r^2}\left\{2\frac{\partial r}{\partial n}[(1-2\nu)\delta_{ij}r_{,k} + \nu(r_{,j}\delta_{ik} + r_{,i}\delta_{jk}) - 4r_{,i}r_{,j}r_{,k}]\right\} \\ &+\frac{\mu}{2\pi(1-\nu)r^2}\{2\nu(n_ir_{,j}r_{,k} + n_jr_{,i}r_{,k})\} \\ &+\frac{\mu}{2\pi(1-\nu)r^2}\{(1-2\nu)(2n_kr_{,i}r_{,j} + n_j\delta_{ik} + n_i\delta_{jk}) - (1-4\nu)n_k\delta_{ij}\}, \end{aligned} \tag{3.1.24}$$

and in three dimensions as

$$D_{kij} = \frac{1}{8\pi(1-\nu)r^2}[(1-2\nu)(\delta_{ik}r_{,j} + \delta_{jk}r_{,i} - \delta_{ij}r_{,k}) + 3r_{,i}r_{,j}r_{,k}], \tag{3.1.25}$$

$$\begin{aligned} S_{kij} = &\frac{\mu}{4\pi(1-\nu)r^3}\left\{3\frac{\partial r}{\partial n}[(1-2\nu)\delta_{ij}r_{,k} + \nu(r_{,j}\delta_{ik} + r_{,i}\delta_{jk}) - 5r_{,i}r_{,j}r_{,k}]\right\} \\ &+\frac{\mu}{4\pi(1-\nu)r^3}\{3\nu(n_ir_{,j}r_{,k} + n_jr_{,i}r_{,k})\} \\ &+\frac{\mu}{4\pi(1-\nu)r^3}\{(1-2\nu)(3n_kr_{,i}r_{,j} + n_j\delta_{ik} + n_i\delta_{jk}) - (1-4\nu)n_k\delta_{ij}\}. \end{aligned} \tag{3.1.26}$$

An important property of fundamental solutions is the singularity at the source point. The singularity orders are summarized in Tab. 3.1.

Kernel	Dimension	Order	Singularity type
U_{ij}	2D	$O(\ln(1/r))$	weakly singular
T_{ij}	2D	$O(1/r)$	strongly singular
U_{ij}	3D	$O(1/r)$	weakly singular
T_{ij}	3D	$O(1/r^2)$	strongly singular
D_{kij}	2D	$O(1/r)$	strongly singular
S_{kij}	2D	$O(1/r^2)$	hypersingular
D_{kij}	3D	$O(1/r^2)$	strongly singular
S_{kij}	3D	$O(1/r^3)$	hypersingular

Table 3.1: The singularity of kernel functions

3.1.2 Boundary integral equations

From fundamental solutions, a boundary integral equation (BIE) can be obtained. The BIE can be classified into two categories:

- **Indirect Boundary Integral Equations**

  The unknowns in the equation are not with an obvious physical interpretation, so the field quantities of interest need to be recovered after solving the equations.

- **Direct Boundary Integral Equations**

  Direct boundary integral equation is obtained directly from Somigliana's identities through a limiting approach. The unknowns in the equation are the quantities of interest in mechanics, such as displacement and traction around the boundary for linear elasticity. Hence, the direct boundary integral equation is used in the present thesis and will be reviewed in this section.

Betti's reciprocal theorem

Suppose $(u_i^{adj}, \epsilon_{ij}^{adj}, \sigma_{ij}^{adj})$ denotes an adjoint state which satisfies elastic governing equations. We take u_i^{adj} as a test function and multiply it with Eq. (3.1.1), yielding the following integral form

$$\int_\Omega (\sigma_{ij,j} + b_i) u_i^{adj} \mathrm{d}\Omega = 0. \tag{3.1.27}$$

Eq. (3.1.27) can be integrated by parts and yields

$$\int_\Omega \sigma_{ij,j} u_i^{adj} \mathrm{d}\Omega = \int_\Omega (\sigma_{ij} u_i^{adj})_{,j} \mathrm{d}\Omega - \int_\Omega \sigma_{ij} u_{i,j}^{adj} \mathrm{d}\Omega. \tag{3.1.28}$$

As the strain tensor is the symmetric gradient of displacement, the following equation holds:

$$\sigma_{ij} u_{i,j}^{adj} = \sigma_{ij} \epsilon_{ij}^{adj}. \tag{3.1.29}$$

From the divergence theorem, the second domain integral of Eq. (3.1.28) becomes

$$\int_\Omega (\sigma_{ij} u_i^{adj})_{,j} \mathrm{d}\Omega = \int_S \sigma_{ij} n_j u_i^{adj} \mathrm{d}\Omega. \tag{3.1.30}$$

By writing $\sigma_{ij} n_j = t_i$, the above equation becomes

$$\int_\Omega (\sigma_{ij} u_i^{adj})_{,j} \mathrm{d}\Omega = \int_S t_i u_i^{adj} \mathrm{d}S. \tag{3.1.31}$$

Substituting Eq. (3.1.29) and Eq. (3.1.31) to Eq. (3.1.28) yields

$$\int_\Omega \sigma_{ij,j} u_i^{adj} \mathrm{d}\Omega = \int_S t_i u_i^{adj} \mathrm{d}S - \int_\Omega \sigma_{ij} \epsilon_{ij}^{adj} \mathrm{d}\Omega. \tag{3.1.32}$$

Substituting Eq. (3.1.1) into the above equation, we obtain the following expression

$$\int_\Omega b_i u_i^{adj} \mathrm{d}\Omega = \int_S t_i u_i^{adj} \mathrm{d}S - \int_\Omega \sigma_{ij} \epsilon_{ij}^{adj} \mathrm{d}\Omega, \tag{3.1.33}$$

or

$$\int_\Omega \sigma_{ij} \epsilon_{ij}^{adj} \mathrm{d}\Omega = \int_S t_i u_i^{adj} \mathrm{d}S + \int_\Omega b_i u_i^{adj} \mathrm{d}\Omega. \tag{3.1.34}$$

Through a similar procedure, we can get

$$\int_\Omega \sigma_{ij}^{adj} \epsilon_{ij} \mathrm{d}\Omega = \int_S t_i^{adj} u_i \mathrm{d}S + \int_\Omega b_i^{adj} u_i \mathrm{d}\Omega. \tag{3.1.35}$$

It is noticed that the left hand sides of Eqs. (3.1.34) and (3.1.35) are equal, *i.e.*,

$$\int_\Omega \sigma_{ij}\epsilon_{ij}^{adj}\mathrm{d}\Omega = \int_\Omega \sigma_{ij}^{adj}\epsilon_{ij}\mathrm{d}\Omega. \tag{3.1.36}$$

The above identity can be proved as follows:

$$\begin{aligned}\int_\Omega \sigma_{ij}\epsilon_{ij}^{adj}\mathrm{d}\Omega &= \int_\Omega [\lambda\delta_{ij}\epsilon_{ij}^{adj}\epsilon_{kk} + 2\mu\epsilon_{ij}\epsilon_{ij}^{adj}]\mathrm{d}\Omega \\ &= \int_\Omega [\lambda\delta_{ij}\epsilon_{ij}^{adj}\epsilon_{kk} + 2\mu\epsilon_{ij}\epsilon_{ij}^{adj}]\mathrm{d}\Omega \\ &= \int_\Omega [\lambda\delta_{ij}\epsilon_{mm}^{adj} + 2\mu\epsilon_{ij}^{adj}]\mathrm{d}\Omega \\ &= \int_\Omega \sigma_{ij}^{adj}\epsilon_{ij}\mathrm{d}\Omega. \end{aligned}\tag{3.1.37}$$

From Eqs. (3.1.34, 3.1.35, 3.1.36), we get the following identity, also known as Betti's reciprocal work theorem,

$$\int_S t_i u_i^{adj}\mathrm{d}S + \int_\Omega b_i u_i^{adj}\mathrm{d}\Omega = \int_S t_i^{adj}u_i\mathrm{d}S + \int_\Omega b_i^{adj}u_i\mathrm{d}\Omega. \tag{3.1.38}$$

Somigliana's identity

The boundary integral equation for elasticity can be derived from Betti's reciprocal work theorem by taking the adjoint state as the solutions corresponding to a concentrated body force in Eq. (3.1.9). Substituting Eqs. (3.1.11,3.1.12) into Eq. (3.1.38) leads to

$$\begin{aligned}u_i(\mathbf{S}) = &\int_S U_{ij}(\mathbf{S},\mathbf{x})t_j(\mathbf{x})\mathrm{d}S(\mathbf{x}) - \int_S T_{ij}(\mathbf{S},\mathbf{x})u_j(\mathbf{x})\mathrm{d}S(\mathbf{x}) \\ &+ \int_\Omega U_{ij}(\mathbf{S},\mathbf{x})b_j(\mathbf{x})\mathrm{d}\Omega(\mathbf{x}),\end{aligned}\tag{3.1.39}$$

by remembering the sifting property of *Dirac delta* distribution

$$\int_\Omega \delta(\mathbf{x}-\mathbf{S})e_i u_i \mathrm{d}\Omega = u_i(\mathbf{S})\,e_i. \tag{3.1.40}$$

Eq. (3.1.39) is known as Somigliana's identity for displacements. It relates the displacements at interior points to the displacements and tractions on the boundary.

Finally, Somigliana's identity for stresses can be obtained by substituting Eq. (3.1.39) into Eq. (3.1.7) as

$$\begin{aligned}\sigma_{ij}(\mathbf{S}) = &\int_S D_{kij}(\mathbf{S},\mathbf{x})t_k(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) - \int_S S_{kij}(\mathbf{S},\mathbf{x})u_k(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) \\ &+ \int_\Omega D_{kij}(\mathbf{S},\mathbf{x})b_k(\mathbf{x})\,\mathrm{d}\Omega(\mathbf{x}).\end{aligned}\tag{3.1.41}$$

Displacement boundary integral equations

In Eqs. (3.1.39) and (3.1.41), the source point is still located inside the domain. To get a boundary integral equation for a source point on the boundary, it is necessary to consider the limit process as the source point approaches the boundary, *i.e.* $\mathbf{S} \rightarrow \mathbf{s}$.

Consider a computational domain augmented around the boundary source point by a semi-circular region for two-dimensional problems and a hemispherical region for three-dimensional problems, with radius ε as illustrated in Figs 3.4 and 3.5. The augmented boundary is now expressed by

$$(S - S_\varepsilon) + S_\varepsilon^+, \tag{3.1.42}$$

where S_ε is the portion of the original geometry that has been removed, and S_ε^+ the portion of the semi-circular arc or hemispherical surface.

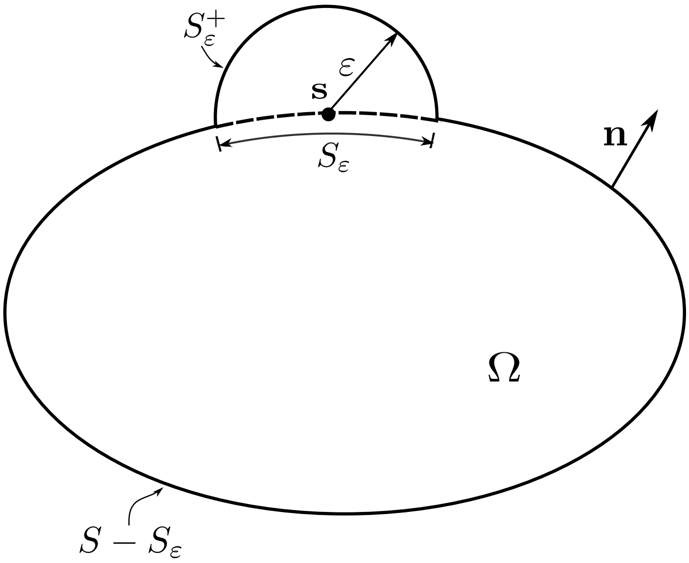


Figure 3.4: Semi-circular arc around source point in two dimensions

Consider the first boundary integral on the right-hand side of Eq. (3.1.39). In the limit when $\varepsilon \rightarrow 0$, it can be split into

$$\begin{aligned} \int_S U_{ij}(\mathbf{s},\mathbf{x}) t_j(\mathbf{x}) \mathrm{d}S(\mathbf{x}) = \lim_{\varepsilon \rightarrow 0} \int_{S-S_\varepsilon} & U_{ij}(\mathbf{s},\mathbf{x}) t_j(\mathbf{x}) \mathrm{d}S(\mathbf{x}) \\ + \lim_{\varepsilon \rightarrow 0} \int_{S_\varepsilon^+} & U_{ij}(\mathbf{s},\mathbf{x}) t_j(\mathbf{x}) \mathrm{d}S(\mathbf{x}). \end{aligned} \tag{3.1.43}$$

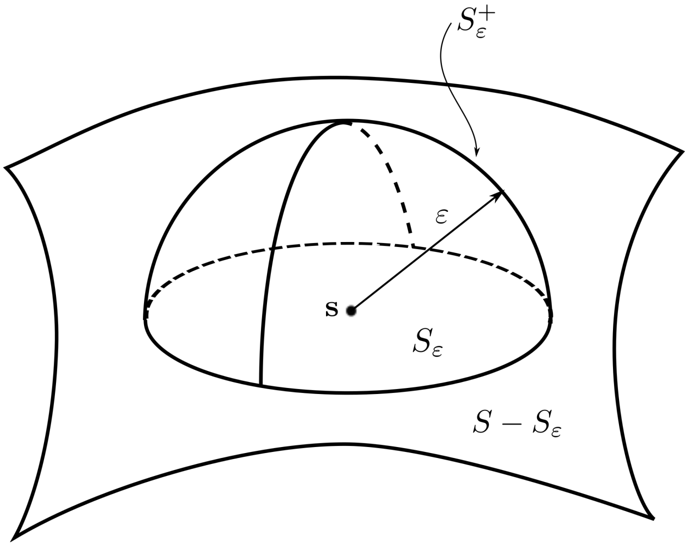


Figure 3.5: Hemisphere around source point in three dimensions

The integrand is of order $O(\ln(1/r))$ in two dimensions and $O(1/r)$ in three dimensions. The first term of the right-hand side of the above equation is an improper integral and should be evaluated using special techniques. The second term of the right-hand side vanishes when $\varepsilon \to 0$.

Now consider the second boundary integral of the right-hand side of Eq. (3.1.39), which can be written as

$$\begin{aligned}\int_S T_{ij}(\mathbf{s},\mathbf{x})u_j(\mathbf{x})\mathrm{d}S(\mathbf{x}) = &\lim_{\varepsilon\to 0}\int_{S-S_\varepsilon} T_{ij}(\mathbf{s},\mathbf{x})u_j(\mathbf{x})\mathrm{d}S(\mathbf{x}) \\ &+\lim_{\varepsilon\to 0}\int_{S_\varepsilon^+} T_{ij}(\mathbf{s},\mathbf{x})u_j(\mathbf{x})\mathrm{d}S(\mathbf{x}). \qquad (3.1.44)\end{aligned}$$

The integrand is of order $O(1/r)$ in two dimensions and $O(1/r^2)$ in three dimensions. The first term of the right-hand side of the above equation is only integrable in Cauchy principal value sense, and the second integral can be rewritten by subtracting and adding the first term of the Taylor series expansion of the displacement at the source point,

$$\begin{aligned}\lim_{\varepsilon\to 0}\int_{S_\varepsilon^+} T_{ij}(\mathbf{s},\mathbf{x})u_j(\mathbf{x})\mathrm{d}S(\mathbf{x}) = &\lim_{\varepsilon\to 0}\int_{S_\varepsilon^+} T_{ij}(\mathbf{s},\mathbf{x})\left[u_j(\mathbf{x}) - u_j(\mathbf{s})\right]\mathrm{d}S(\mathbf{x}) \\ &+u_j(\mathbf{s})\lim_{\varepsilon\to 0}\int_{S_\varepsilon^+} T_{ij}(\mathbf{s},\mathbf{x})\mathrm{d}S(\mathbf{x}). \qquad (3.1.45)\end{aligned}$$

The second term on the right-hand side of the above equation can be written in the following form

$$u_j\left(\mathbf{s}\right)\lim_{\varepsilon\to 0}\int_{S_\varepsilon^+} T_{ij}\left(\mathbf{s},\ \mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right)=\alpha_{ij}\left(\mathbf{s}\right)u_j\left(\mathbf{s}\right), \tag{3.1.46}$$

where α_{ij} are the coefficients depending only on the local geometry around the source point. The value is $-\frac{1}{2}\delta_{ij}$ for a smooth geometry.

Finally, Eq. (3.1.39) is transferred to the displacement boundary integral equation in the limit of $\varepsilon \to 0$ by ignoring domain integrals

$$C_{ij}(\mathbf{s})u_j(\mathbf{s})+\fint_S T_{ij}(\mathbf{s},\mathbf{x})u_j(\mathbf{x})\mathrm{d}S(\mathbf{x})=\int_S U_{ij}(\mathbf{s},\mathbf{x})t_j(\mathbf{x})\mathrm{d}S(\mathbf{x}). \tag{3.1.47}$$

Stress boundary integral equations

Now consider the limit of Eq. (3.1.41) as $\varepsilon \to 0$. Its first integral can be split into

$$\begin{aligned}\int_S D_{kij}(\mathbf{s},\mathbf{x})t_k(\mathbf{x})\mathrm{d}S(\mathbf{x})=&\lim_{\varepsilon\to 0}\int_{S-S_\varepsilon} D_{kij}(\mathbf{s},\mathbf{x})t_k(\mathbf{x})\mathrm{d}S(\mathbf{x})\\&+\lim_{\varepsilon\to 0}\int_{S_\varepsilon^+} D_{kij}(\mathbf{s},\mathbf{x})t_k(\mathbf{x})\mathrm{d}S(\mathbf{x}).\end{aligned} \tag{3.1.48}$$

The second term of Eq. (3.1.48) is

$$\begin{aligned}\lim_{\varepsilon\to 0}\int_{S_\varepsilon^+} D_{kij}(\mathbf{s},\mathbf{x})t_k(\mathbf{x})\mathrm{d}S(\mathbf{x})=&\lim_{\varepsilon\to 0}\int_{S_\varepsilon^+} D_{kij}(\mathbf{s},\mathbf{x})\left[t_k(\mathbf{x})-t_k(\mathbf{s})\right]\mathrm{d}S(\mathbf{x})\\&+t_k(\mathbf{s})\lim_{\varepsilon\to 0}\int_{S_\varepsilon^+} D_{kij}(\mathbf{s},\mathbf{x})\mathrm{d}S(\mathbf{x}).\end{aligned} \tag{3.1.49}$$

The first integral in the right-hand side of the above equation vanishes in the limiting process. The second integral leads to a jump term on the tractions, given by

$$t_k\left(\mathbf{s}\right)\lim_{\varepsilon\to 0}\int_{S_\varepsilon^+} D_{kij}\left(\mathbf{s},\ \mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right)=\gamma_{ij}\left(\mathbf{s}\right). \tag{3.1.50}$$

Thus Eq. (3.1.48) can be written as

$$\int_S D_{kij}\left(\mathbf{s},\ \mathbf{x}\right)t_k\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right)=\fint_S D_{kij}\left(\mathbf{s},\ \mathbf{x}\right)t_k\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right)+\gamma_{ij}\left(\mathbf{s}\right). \tag{3.1.51}$$

Now consider the second integral of Eq. (3.1.41) and split it into two terms as

$$\begin{aligned}&\int_S S_{kij}\left(\mathbf{s},\ \mathbf{x}\right)u_k\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right)\\&=\lim_{\varepsilon\to 0}\int_{S-S_\varepsilon^+} S_{kij}\left(\mathbf{s},\ \mathbf{x}\right)u_k\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right)+\lim_{\varepsilon\to 0}\int_{S_\epsilon^+} S_{kij}\left(\mathbf{s},\ \mathbf{x}\right)u_k\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right).\end{aligned} \tag{3.1.52}$$

The second integral on the right-hand side of Eq. (3.1.52) is

$$\begin{aligned}
&\int_{S_\varepsilon^+} S_{kij}(\mathbf{s},\mathbf{x})u_k(\mathbf{x})\mathrm{d}S(\mathbf{x}) \\
&= \lim_{\varepsilon\to 0}\left\{\int_{S_\varepsilon^+} S_{kij}(\mathbf{s},\mathbf{x})[u_k(\mathbf{x})-u_k(\mathbf{s})-u_{k,m}(x_m-s_m)]\mathrm{d}S(\mathbf{x})\right. \\
&\left.\quad +u_k(\mathbf{s})\int_{S_\varepsilon^+} S_{kij}(\mathbf{s},\mathbf{x})\mathrm{d}S(\mathbf{x}) + u_{k,m}(\mathbf{s})\int_{S_\varepsilon^+} S_{kij}(\mathbf{s},\mathbf{x})(x_m-s_m)\mathrm{d}S(\mathbf{x})\right\}. \qquad (3.1.53)
\end{aligned}$$

The first integral of the right-hand side of Eq. (3.1.53) vanishes in the limiting process and the second integral is unbounded if considered in isolation,

$$u_k\left(\mathbf{s}\right)\int_{S_\varepsilon^+} S_{kij}\left(\mathbf{s},\ \mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right) = u_k\left(\mathbf{s}\right)\frac{b_{kij}}{\varepsilon}. \qquad (3.1.54)$$

The last integral of the right-hand side of Eq. (3.1.53) leads to jump terms of displacements,

$$\lim_{\varepsilon\to 0} u_{k,m}\left(\mathbf{s}\right)\int_{S_\varepsilon^+} S_{kij}\left(\mathbf{s},\mathbf{x}\right)\left(x_m-s_m\right)\mathrm{d}S\left(\mathbf{x}\right) = \beta_{ij}\left(\mathbf{s}\right). \qquad (3.1.55)$$

The sum of the first integral of the right-hand side of Eq. (3.1.52) and the second integral of Eq. (3.1.53) leads to an integral in Hadamard principal Value sense

$$\mathop{=\!\!\!\!\!\!\int}_S S_{kij}\left(\mathbf{s},\ \mathbf{x}\right)u_k\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right) = \lim_{\epsilon\to 0}\left\{\int_{S-S_\varepsilon^+} S_{kij}\left(\mathbf{s},\ \mathbf{x}\right)u_k\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right) + u_j\left(\mathbf{s}\right)\frac{b_{kij}\left(\mathbf{s}\right)}{\varepsilon}\right\}. \qquad (3.1.56)$$

Finally the integral of Eq. (3.1.41) can be written as

$$\begin{aligned}
\sigma_{ij}\left(\mathbf{s}\right) - \left(\beta_{ij}\left(\mathbf{s}\right) - \gamma_{ij}\left(\mathbf{s}\right)\right) = &\fint_S D_{kij}\left(\mathbf{s},\ \mathbf{x}\right)t_k\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right) - \mathop{=\!\!\!\!\!\!\int}_S S_{kij}\left(\mathbf{s},\ \mathbf{x}\right)u_k\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right) \\
&+ \int_\Omega D_{kij}\left(\mathbf{s},\ \mathbf{x}\right)b_k\left(\mathbf{x}\right)\mathrm{d}\Omega\left(\mathbf{x}\right), \qquad (3.1.57)
\end{aligned}$$

and the stress boundary integral equation is obtained by ignoring domain integrals

$$C_{ij}\left(\mathbf{s}\right)\sigma_{ij}\left(\mathbf{s}\right) + \mathop{=\!\!\!\!\!\!\int}_S S_{kij}\left(\mathbf{s},\ \mathbf{x}\right)u_k\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right) = \fint_S D_{kij}\left(\mathbf{s},\ \mathbf{x}\right)t_k\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right). \qquad (3.1.58)$$

3.2 Boundary element methods

Given the boundary integral equation, the BEM formulations can be obtained by the following two steps:

- Discretize displacement and traction fields in the BIE using a set of locally supported basis functions.

- Construct a series of discrete functions by the collocation method or the Galerkin method.

According to the choice of weighting functions, we obtain the collocation BEM and the symmetric Galerkin BEM (SGBEM) [2, 20]. Collocation BEM can be viewed as taking *Dirac delta* functions as weighting functions, *i.e.* enforcing boundary integral equations satisfied on a series of discrete points. The SGBEM employs the shape functions discretizing the field as weighting functions, satisfying the equation in an "average" sense. The following is a comparison between the SGBEM and collocation BEM:

- As in collocation FEM, the collocation BEM is not supported by a strong mathematical theory, although numerous experiments have produced satisfactory results. In comparison, SGBEM has a well-founded mathematical formulation, and the existence and convergence of the solution can be proved for many practical problems.

- Collocation BEM normally leads to a non-symmetric matrix, whereas SGBEM produces a symmetric matrix. A symmetric matrix has a better condition and requires less storage. Moreover, it has the ability of taking advantage of iterative solvers. The coupling with FEM is also easier with a symmetric matrix.

- SGBEM deals with edges and corners more easily, as well as the hypersingular boundary integrals equations which are essential in fracture mechanics, acoustics problems and electromagnetic problems. This is due to the fact that source points are always located inside elements so that high order continuity requirement of the BIE at the source point is satisfied.

- The main advantage of collocation BEM over SGBEM is computational efficiency, because SGBEM requires a dual boundary integral which is very time-consuming. In the present thesis, the collocation scheme is used and will be presented in the following.

3.2.1 Boundary element method formulations

In classical BEM, the boundary can be divided into a set of non-overlapping elements as

$$S = \bigcup_{e=1}^{n_e} S_e, \qquad S_i \cap S_j = \emptyset, \qquad i \neq j, \tag{3.2.59}$$

where n_e is the number of elements and e is the element index. The field can then be approximated as piecewise by the polynomial shape functions defined on each element. A widely used element type in BEM and FEM is isoparametric elements, where the geometry is approximated using the same shape functions as that discretizing the field. It is noteworthy that it should be advantageous in some cases to have independent field and geometry approximations [106]. Following are some commonly used types of isoparametric elements and shape functions.

- Quadratic elements (Fig. 3.6):

$$\begin{aligned} N_1(\tilde{\xi}) &= \frac{1}{2}\tilde{\xi}(\tilde{\xi}-1), \\ N_2(\tilde{\xi}) &= 1-\tilde{\xi}^2, \\ N_3(\tilde{\xi}) &= \frac{1}{2}\tilde{\xi}(\tilde{\xi}+1). \end{aligned} \tag{3.2.60}$$

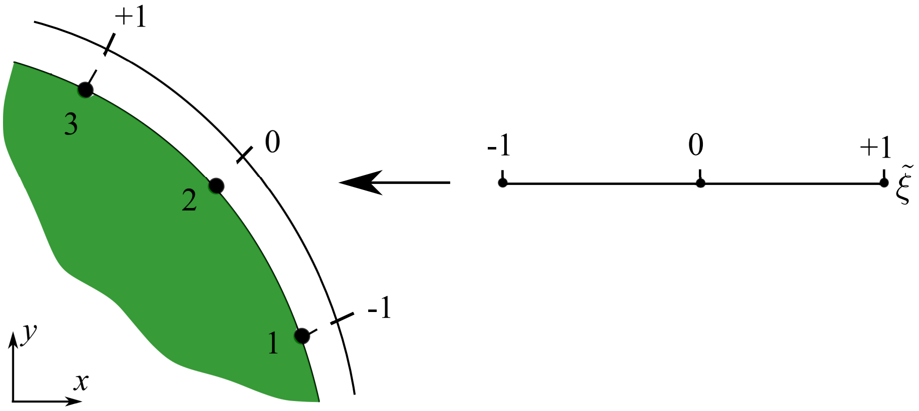


Figure 3.6: Quadratic element for two-dimension problems

- Quadrilateral elements (Fig. 3.7):

$$
\begin{aligned}
N_1(\tilde{\xi}_1, \tilde{\xi}_2) &= \frac{1}{4}(\tilde{\xi}_1 + 1)(\tilde{\xi}_2 + 1), \\
N_2(\tilde{\xi}_1, \tilde{\xi}_2) &= \frac{1}{4}(1 - \tilde{\xi}_1)(\tilde{\xi}_2 + 1), \\
N_3(\tilde{\xi}_1, \tilde{\xi}_2) &= \frac{1}{4}(\tilde{\xi}_1 - 1)(\tilde{\xi}_2 - 1), \\
N_4(\tilde{\xi}_1, \tilde{\xi}_2) &= \frac{1}{4}(\tilde{\xi}_1 + 1)(1 - \tilde{\xi}_2). \qquad (3.2.61)
\end{aligned}
$$

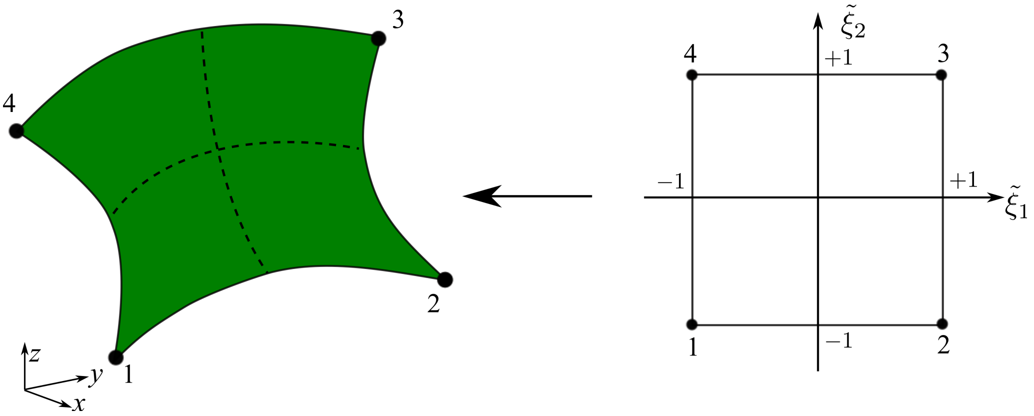


Figure 3.7: Bilinear element for three-dimension problems

An important property of isoparametric elements is that the shape functions are polynomials and possess the *Kronecker delta* property, as shown in Fig. 3.8 for quadratic elements.

To formulate the BEM equations, the displacements and tractions around the boundary are discretized with isoparametric elements as

$$u_j^e(\tilde{\boldsymbol{\xi}}) = \sum_{a=1}^{n_a} N_{ea}(\tilde{\boldsymbol{\xi}}) u_j^{ea}, \qquad (3.2.62)$$

$$t_j^e(\tilde{\boldsymbol{\xi}}) = \sum_{a=1}^{n_a} N_{ea}(\tilde{\boldsymbol{\xi}}) t_j^{ea}, \qquad (3.2.63)$$

where e is the element index and a the local index of the node in element e, $\tilde{\boldsymbol{\xi}}$ the intrinsic coordinates in parent elements. In one dimensional parent elements, $\tilde{\boldsymbol{\xi}}$ reduces to a scalar $\tilde{\xi}$, and in two dimensions it is a vector of $(\tilde{\xi}_1, \tilde{\xi}_2)$. u_j and t_j are

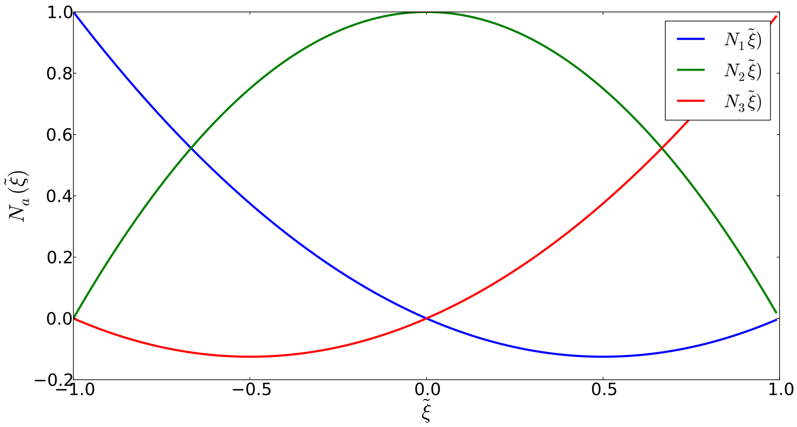


Figure 3.8: Quadratic basis functions

nodal displacements and tractions. The geometry is then approximated using the same polynomial basis functions as Eqs. (3.2.62) and (3.2.63),

$$\mathbf{x}_e(\tilde{\boldsymbol{\xi}}) = \sum_{a=1}^{n_a} N_{ea}(\tilde{\boldsymbol{\xi}})\mathbf{x}_{ea}. \tag{3.2.64}$$

Substituting Eqs. (3.2.62) and (3.2.63) into Eq. (3.1.47) yields

$$\begin{aligned} C_{ij}\left(\mathbf{s}_c\right) u_j^{e_0 a_0} + \sum_{e=1}^{n_e}\sum_{a=1}^{n_a} \fint_{\tilde{S}} T_{ij}(\mathbf{s}_c, \tilde{\boldsymbol{\xi}}) N_{ea}(\tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}) u_j^{ea} \\ = \sum_{e=1}^{n_e}\sum_{a=1}^{n_a} \int_{\tilde{S}} U_{ij}(\mathbf{s}_c, \tilde{\boldsymbol{\xi}}) N_{ea}(\tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}) t_j^{ea}, \end{aligned} \tag{3.2.65}$$

where c indicates the collocation point index, e_0 the element in which the collocation point is located, and a_0 the local index of the collocation point in element e_0.

We write the terms in Eq. (3.2.65) with the following shorthand symbols,

$$\hat{H}_{ij}^{ce_0a_0} = C_{ij}(\mathbf{s}_c) u_j^{ce_0a_0}, \tag{3.2.66}$$

$$\bar{H}_{ij}^{cea} = \fint_{\tilde{S}} T_{ij}(\mathbf{s}_c, \tilde{\boldsymbol{\xi}}) N_{ea}(\tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}), \tag{3.2.67}$$

$$G_{ij}^{cea} = \int_{\tilde{S}} U_{ij}(\mathbf{s}_c, \tilde{\boldsymbol{\xi}}) N_{ea}(\tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}). \tag{3.2.68}$$

In classical BEM, a natural and convenient choice is to choose the nodes as the collocation points. Using the mapping from the element index space and from the local node index space to the global index space,

$$(e, a) \mapsto A, \qquad (e_0, a_0) \mapsto c, \tag{3.2.69}$$

and substituting matrix entries of Eqs. (3.2.66, 3.2.67, 3.2.68) to Eq. (3.2.65) yields

$$H_{ij}^{cA} u_j^A = G_{ij}^{cA} t_j^A, \tag{3.2.70}$$

with

$$H_{ij}^{cA} = \hat{H}_{ij}^{cc} \delta_{cA} + \bar{H}_{ij}^{cA}, \tag{3.2.71}$$

$$G_{ij}^{cA} = G_{ij}^{cA}. \tag{3.2.72}$$

Eq. (3.2.70) can be written in matrix form

$$\mathbf{Hu} = \mathbf{Gt}, \tag{3.2.73}$$

where matrix $\mathbf{H}$ collects the entries of H_{ij}^{cA} and $\mathbf{G}$ of G_{ij}^{cA}. The column vector $\mathbf{u}$ contains the nodal displacements, and $\mathbf{t}$ the nodal tractions. Both $\mathbf{u}$ and $\mathbf{t}$ include unknowns and the values prescribed by boundary conditions. By swapping the unknowns and the related coefficients of both sides, we obtain

$$\mathbf{Az} = \mathbf{By}. \tag{3.2.74}$$

The column vector $\mathbf{z}$ contains all the displacement and traction unknowns, $\mathbf{y}$ contains all the nodal parameters given by boundary conditions, $\mathbf{A}$ is a coefficient matrix which is usually non-symmetric and densely populated, and $\mathbf{B}$ is a matrix which contains the coefficients corresponding to the prescribed boundary conditions. The product of $\mathbf{B}$ and $\mathbf{y}$ yields the right-hand side vector $\mathbf{f}$, *i.e.*

$$\mathbf{Az} = \mathbf{f}. \tag{3.2.75}$$

The above equation is a linear system which can be solved to obtain the values of the unknown displacement and traction parameters.

3.2.2 Evaluation of integrals

For elements which do not contain the collocation point, the element integral is regular and the Gauss-Legendre quadrature can be used, *i.e.*

$$\int_{-1}^{+1} f(\tilde{\xi}) \mathrm{d}\xi \cong \sum_{g=1}^{n_g} f(\tilde{\xi}_g) w_g, \tag{3.2.76}$$

for line integrals, and

$$\int_{-1}^{+1}\int_{-1}^{+1} f(\tilde{\xi}_1,\tilde{\xi}_2)\mathrm{d}\tilde{\xi}_1\mathrm{d}\tilde{\xi}_2 \cong \sum_{g=1}^{n_g}\sum_{g=1}^{n_g} f(\tilde{\xi}_{1,g},\tilde{\xi}_{2,g})w_g, \tag{3.2.77}$$

for surface integrals, where g is the index of the Gauss point, w_g is the weight, and n_g is the number of Gauss points in each dimension.

Evaluation of weakly singular integrals

When a collocation point is located in the element, the right-hand side of Eq. (3.2.65) is a weakly singular integral, *i.e.* $O(\ln(1/r))$ in two-dimensions, and $O(1/r)$ in three-dimensions, which can be treated by introducing a mapping such that its Jacobian cancels the singularity.

In two-dimensional problems, Telles transformation [99] is used,

$$\tilde{\xi} = (1-\bar{\xi}^2)\frac{\bar{\zeta}}{2} + \bar{\xi}, \tag{3.2.78}$$

where $\bar{\xi}$ and $\bar{\zeta}$ are the coordinates of the field point and collocation points in the transformed system, respectively. The Jacobian for this mapping is

$$\bar{J} = (1-\bar{\xi}\bar{\zeta}), \tag{3.2.79}$$

which approaches zero at the collocation point, so the singularity vanishes and a Gauss-Legendre quadrature scheme can be used as for regular integrals.

For three dimensional problems, a polar integration scheme can also introduce a Jacobian cancelling singularities of order $O(1/r)$. As shown in Fig. 3.9, the parent element is subdivided into subelements sharing the collocation point as a common vertex, then a Gauss-Legendre integral can be performed for each subelement in the polar coordinate system (θ-ρ),

$$\begin{aligned}&\int_{-1}^{+1}\int_{-1}^{+1} f(\tilde{\xi}_1,\tilde{\xi}_2)\mathrm{d}\tilde{\xi}_1\mathrm{d}\tilde{\xi}_2\\ =&\int_\rho\int_\theta f(\tilde{\xi}_1(\theta,\rho),\tilde{\xi}_2(\theta,\rho))J(\rho,\theta)\mathrm{d}\rho\mathrm{d}\theta.\end{aligned} \tag{3.2.80}$$

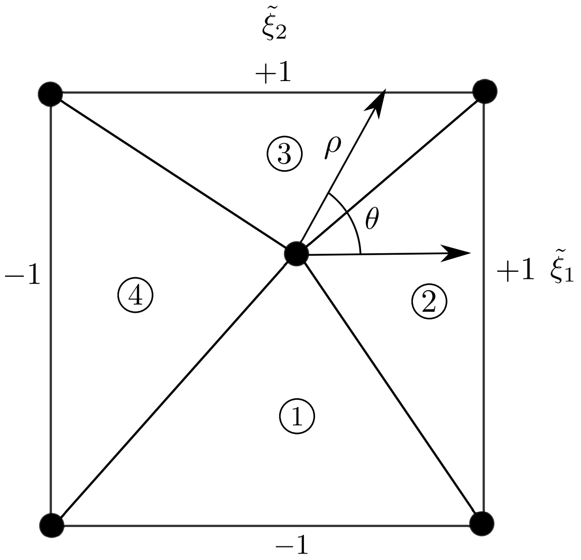


Figure 3.9: Polar integration

Evaluation of strongly singular integrals

When the collocation point is located in the element, the integrals of the left-hand side in Eq. (3.2.65) is strongly singular. Recall the form of the strongly singular integral as follows,

$$\fint_{\tilde{S}} T_{ij}\left(\mathbf{s}_c,\ \tilde{\boldsymbol{\xi}}\right) N_A(\tilde{\boldsymbol{\xi}}) J(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}). \tag{3.2.81}$$

In isoparametric elements, when $c \neq A$, *i.e.* the collocation point does not coincide with the node A, the value of shape function N_A vanishes on the collocation point due to the *Kronecker delta* property. Moreover, it approaches zero as $O(r)$, which can cancel the strong singularity of kernel function $T_{ij}(\mathbf{s}_c, \mathbf{x}(\tilde{\boldsymbol{\xi}}))$. When $c = A$, *i.e.*, collocation point is located on node A, strongly singular integrals must be evaluated. By using a special technique, so-called rigid body motion method, strongly singular integrals together with jump terms do not need to be evaluated explicitly but obtained from the known values in the matrix $\mathbf{H}$ in Eq. (3.2.73). By noticing that the jump terms and strongly singular integrals are always arranged along the diagonal line of matrix $\mathbf{H}$, rigid body motion methods rely on a physical interpretation of BEM: if the traction is zero, the equation should recover a rigid body motion. For example, Eq. (3.2.82) is reduced from Eq. (3.2.73) on a two-dimensional traction free state, where the black dots in matrix $\mathbf{H}$ denote the positions of jump terms and singular integrals, and the right-hand side is a zero-vector due to the zero traction. Then the displacement vector represents a unit rigid body motion in the x-direction, and an entry on the diagonal line of the matrix is calculated from the sum of the remaining terms on the same row, which are regular integrals and can be obtained readily.

$$\begin{bmatrix} \bullet & \bullet & \circ & \circ & \circ & \circ & \circ & \circ & \dots \\ \bullet & \bullet & \circ & \circ & \circ & \circ & \circ & \circ & \dots \\ \circ & \circ & \bullet & \bullet & \circ & \circ & \circ & \circ & \dots \\ \circ & \circ & \bullet & \bullet & \circ & \circ & \circ & \circ & \dots \\ \circ & \circ & \circ & \circ & \bullet & \bullet & \circ & \circ & \dots \\ \circ & \circ & \circ & \circ & \bullet & \bullet & \circ & \circ & \dots \\ \circ & \circ & \circ & \circ & \circ & \circ & \bullet & \bullet & \dots \\ \circ & \circ & \circ & \circ & \circ & \circ & \bullet & \bullet & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{Bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ \vdots \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{Bmatrix}. \tag{3.2.82}$$

The procedure can be expressed as

$$H_{ij}^{cc} = - \sum_{A=1, c \neq A} H_{ij}^{cA}, \tag{3.2.83}$$

where c is the collocation point index, and A is the global node index.

3.2.3 Treatment of corners

For a collocation BEM, corners need to be treated carefully, because the traction field is not continuous on the corner. Taking two-dimensional problems as an example, the situations and the corresponding methods depend on the particular cases at hand:

1. Both sides around the corner are prescribed traction boundary conditions. See Fig. 3.10. In this case, the displacement is without needing special treatment, but the traction nodal values on the corner are set to be discontinuous and have different values for the elements touching the corner. So traction degrees of freedom are more than displacement, but the total number of unknowns is still equal to the number of available equations.

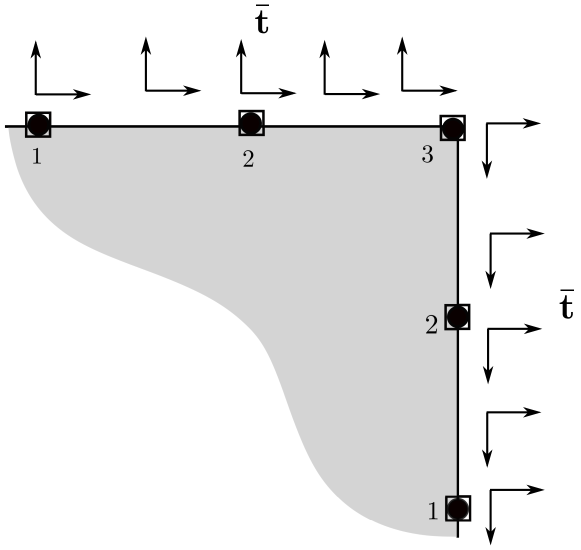


Figure 3.10: Traction-traction boundary conditions around corners

2. One side around the edge is prescribed tractions and the other side displacements. See Fig. (3.11). The case is treated in the same way as above.

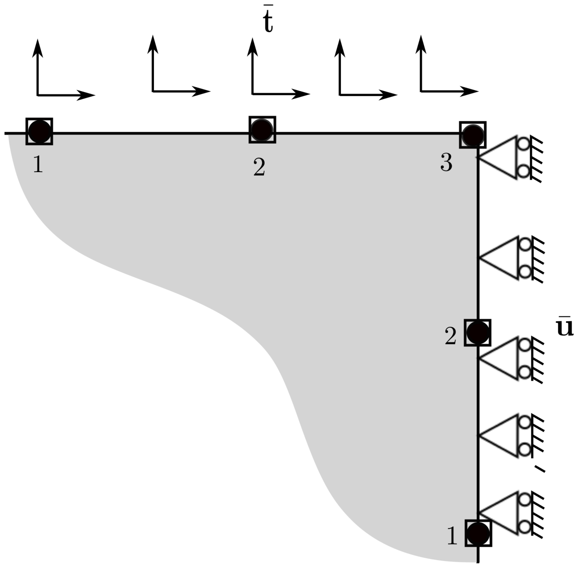


Figure 3.11: Traction-displacement boundary conditions around corners

3. Both sides around the corner are prescribed displacement boundary conditions. In this case, the number of unknowns is larger than the number of equations. An approach to increase the number of equations is to shift the nodes from the corner into the elements. See Fig. 3.12.

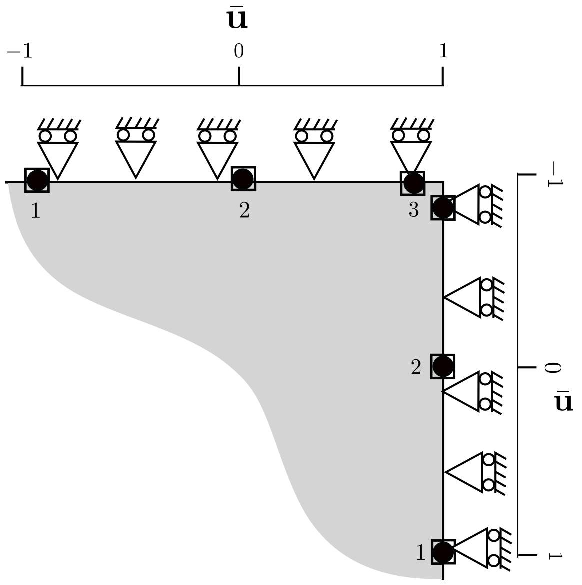


Figure 3.12: Displacement-displacement boundary condition around corners

3.2.4 Postprocessing

Evaluate stresses and displacements at interior points

After computing the displacement and traction of the boundary by solving Eq. (3.2.73), we can evaluate the displacement or stress in the domain if necessary. The displacement and the stress at an interior point results from a straightforward use of Somigliana's identities. For the interior displacement field, the expression is

$$\begin{aligned} u_i\left(\mathbf{S}\right) = & \sum_{e=1}^{n_e}\sum_{a=1}^{n_a}\int_{\tilde{S}} U_{ij}(\mathbf{S},\tilde{\boldsymbol{\xi}})N_{ea}(\tilde{\boldsymbol{\xi}})t_j^{ea}J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}) \\ & -\sum_{e=1}^{n_e}\sum_{a=1}^{n_a}\int_{\tilde{S}} T_{ij}\left(\mathbf{S},\ \tilde{\boldsymbol{\xi}}\right)N_{ea}(\tilde{\boldsymbol{\xi}})u_j^{ea}J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}), \end{aligned} \tag{3.2.84}$$

and for the interior stress field, the formulation is

$$\begin{aligned} \sigma_{ij}\left(\mathbf{S}\right) = & \sum_{e=1}^{n_e}\sum_{a=1}^{n_a}\int_{\tilde{S}} D_{kij}\left(\mathbf{S},\ \tilde{\boldsymbol{\xi}}\right)N_{ea}(\tilde{\boldsymbol{\xi}})t_k^{ea}J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}) \\ & -\sum_{e=1}^{n_e}\sum_{a=1}^{n_a}\int_{\tilde{S}} S_{kij}(\mathbf{S},\tilde{\boldsymbol{\xi}})N_{ea}(\tilde{\boldsymbol{\xi}})u_k^{ea}J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}). \end{aligned} \tag{3.2.85}$$

Evaluate stresses at boundary points

The evaluation of the stress at boundary points can be done using Somigliana's equations, but a singular integral needs to be computed. Furthermore, an integral over the surface is time-consuming. So a simple and efficient way consists in recovering the stress by Hooke's law and Cauchy's formula from the displacement, the displacement gradient and the traction fields:

$$\mathbf{u}_e(\tilde{\xi}) = \sum_{a=1}^{n_a} N_{ea}(\tilde{\xi})\mathbf{u}^{ea}, \tag{3.2.86}$$

$$\frac{\mathrm{d}\mathbf{u}_e(\tilde{\xi})}{\mathrm{d}\tilde{\xi}} = \sum_{a=1}^{n_a} \frac{\mathrm{d}N_{ea}(\tilde{\xi})}{\mathrm{d}\tilde{\xi}}\mathbf{u}^{ea}, \tag{3.2.87}$$

$$\mathbf{t}_e(\tilde{\xi}) = \sum_{a=1}^{n_a} N_{ea}(\tilde{\xi})\mathbf{t}^{ea}. \tag{3.2.88}$$

For two-dimensional problems, define a local coordinate system such that $\hat{\mathbf{e}}_1$ is the unit vector in the normal direction and $\hat{\mathbf{e}}_2$ is the unit vector in the tangential direction (Fig. 3.13), and the vectors in this system can be represented as,

$$\hat{\mathbf{x}} = \hat{x}_1\hat{\mathbf{e}}_1 + \hat{x}_2\hat{\mathbf{e}}_2. \tag{3.2.89}$$

The local unit tangential vector can be obtained by

$$\hat{\mathbf{e}}_1 = \mathbf{n}, \tag{3.2.90}$$

$$\hat{\mathbf{e}}_2 = \frac{\mathbf{m}}{|\mathbf{m}|}, \tag{3.2.91}$$

where $\mathbf{n}$ is the normal, and $\mathbf{m}$ is the tangential vector,

$$\mathbf{m} = \frac{\mathrm{d}\mathbf{x}(\tilde{\xi})}{\mathrm{d}\tilde{\xi}}. \tag{3.2.92}$$

The transformation matrix for the quantities from the global coordinate system to the local tangential system is

$$A = \begin{bmatrix} \hat{\mathbf{e}}_1 \\ \hat{\mathbf{e}}_2 \end{bmatrix} = \begin{bmatrix} n_1 & n_2 \\ -n_2 & n_1 \end{bmatrix}. \tag{3.2.93}$$

Defining displacements, tractions, strains, and stresses in the local coordinates as $\hat{u}_j$, $\hat{t}_j$, $\hat{\epsilon}_{ij}$, and $\hat{\sigma}_{ij}$ respectively, $\hat{\epsilon}_{22}$ can be evaluated through the displacement gradient in the global coordinates,

$$\hat{\epsilon}_{22}(\tilde{\xi}) = \hat{u}_{2,2}(\tilde{\xi}) = \frac{\partial \hat{u}_2}{\partial \tilde{\xi}}\frac{\partial \tilde{\xi}}{\partial \hat{x}_2} = A_{2j}\frac{\partial u_j}{\partial \tilde{\xi}}\frac{\partial \tilde{\xi}}{\partial \hat{x}_2}, \tag{3.2.94}$$

with

$$\frac{\partial\tilde{\xi}}{\partial\hat{x}_2} = \frac{1}{|\mathbf{m}|}. \tag{3.2.95}$$

The stress tensor in the local coordinate system is

$$\hat{\sigma}_{11} = \hat{t}_1, \tag{3.2.96}$$

$$\hat{\sigma}_{12} = \hat{t}_2, \tag{3.2.97}$$

$$\hat{\sigma}_{22} = \left(\frac{E}{1-\nu^2}\right)\hat{\epsilon}_{22} + \left(\frac{\nu}{1-\nu}\right)\hat{t}_1. \tag{3.2.98}$$

Finally, the stress in the global Cartesian coordinate system can be obtained as:

$$\sigma_{ij} = A_{ki}A_{nj}\hat{\sigma}_{kn}. \tag{3.2.99}$$

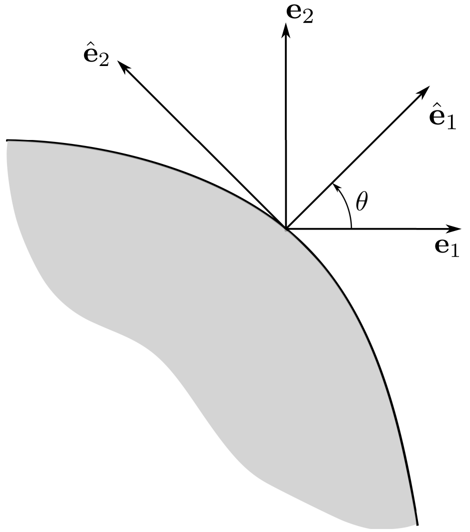


Figure 3.13: Local coordinate system on curve

For three dimensional problems, the procedure is similar to two-dimensional problems. The first step is to construct a tangential coordinate system as

$$\mathbf{m}_1(\tilde{\xi}_1,\tilde{\xi}_2) = \frac{\partial\mathbf{x}}{\partial\xi_1}(\tilde{\xi}_1,\tilde{\xi}_2), \tag{3.2.100}$$

$$\mathbf{m}_2(\tilde{\xi}_1,\tilde{\xi}_2) = \frac{\partial\mathbf{x}}{\partial\xi_2}(\tilde{\xi}_1,\tilde{\xi}_2), \tag{3.2.101}$$

$$\mathbf{n}(\tilde{\xi}_1,\tilde{\xi}_2) = \mathbf{m}_1(\tilde{\xi}_1,\tilde{\xi}_2) \times \mathbf{m}_2(\tilde{\xi}_1,\tilde{\xi}_2), \tag{3.2.102}$$

where $\mathbf{m}_1$ and $\mathbf{m}_2$ are the two tangential vectors, and $\mathbf{n}$ the normal vector to the surface. The tangential coordinate system is neither orthogonal nor unit generally, which is a subtle difference from two-dimensional problems. So we need to establish a unit orthogonal coordinate system based on the the tangential coordinate system (Fig. 3.14), and its three base vectors are given by

$$\hat{\mathbf{e}}_1 = \frac{\mathbf{m}_1}{|\mathbf{m}_1|}, \tag{3.2.103}$$

$$\hat{\mathbf{e}}_3 = \frac{\mathbf{n}}{|\mathbf{n}|}, \tag{3.2.104}$$

$$\hat{\mathbf{e}}_2 = \hat{\mathbf{e}}_1 \times \hat{\mathbf{e}}_3. \tag{3.2.105}$$

And the rotation tensor A_{ij} for the coordinate system transition can be written as the following matrix form

$$A = \begin{bmatrix} \hat{\mathbf{e}}_1 \\ \hat{\mathbf{e}}_2 \\ \hat{\mathbf{e}}_3 \end{bmatrix}. \tag{3.2.106}$$

We also can get the derivatives of intrinsic coordinates of parent element with respect to that of the local orthogonal system

$$\frac{\partial \tilde{\xi}_1}{\partial \hat{x}_1} = \frac{1}{|\mathbf{m}_1|}, \qquad \frac{\partial \tilde{\xi}_1}{\partial \hat{x}_2} = \frac{-\cos\theta}{|\mathbf{m}_1| \sin\theta}, \tag{3.2.107}$$

$$\frac{\partial \tilde{\xi}_2}{\partial \hat{x}_1} = 0, \qquad \frac{\partial \tilde{\xi}_2}{\partial \hat{x}_2} = \frac{1}{|\mathbf{m}_2|} \sin\theta, \tag{3.2.108}$$

where $\hat{x}_1$, $\hat{x}_2$ and $\hat{x}_3$ denote the local orthogonal coordinates.

The strain components in the $\hat{\mathbf{e}}_1$-$\hat{\mathbf{e}}_2$ of local orthogonal system are

$$\hat{\epsilon}_{ij} = \frac{\partial \hat{u}_i}{\partial \hat{x}_j} = \frac{\partial \hat{u}_i}{\partial \tilde{\xi}_k} \frac{\partial \tilde{\xi}_k}{\partial \hat{x}_j} \qquad i, j, k = 1, 2 \tag{3.2.109}$$

with

$$\frac{\partial \hat{u}_i}{\partial \tilde{\xi}_k} = A_{il} \frac{\partial u_l}{\partial \tilde{\xi}_k} \qquad k = 1, 2 \text{ and } i, l = 1, 2, 3. \tag{3.2.110}$$

From the constitutive equations and the relationships between stress and traction, we can get the stress in the local orthogonal system as

$$\hat{\sigma}_{11} = \frac{E}{1-\nu^2}(\hat{\epsilon}_{11} + \nu\hat{\epsilon}_{22}) + \frac{\nu}{1-\nu}\hat{t}_3, \tag{3.2.111}$$

$$\hat{\sigma}_{12} = \frac{E}{1+\nu}\hat{\epsilon}_{12}, \tag{3.2.112}$$

$$\hat{\sigma}_{22} = \frac{E}{1-\nu^2}(\hat{\epsilon}_{22} + \nu\hat{\epsilon}_{11}) + \frac{\nu}{1-\nu}\hat{t}_3, \tag{3.2.113}$$

$$\hat{\sigma}_{33} = \hat{t}_3, \tag{3.2.114}$$

$$\hat{\sigma}_{23} = \hat{t}_2, \tag{3.2.115}$$

$$\hat{\sigma}_{13} = \hat{t}_1. \tag{3.2.116}$$

Then we transfer the stress from local orthogonal system to the global Cartesian system

$$\sigma_{ij} = A_{ki} A_{nj} \hat{\sigma}_{kn}. \tag{3.2.117}$$

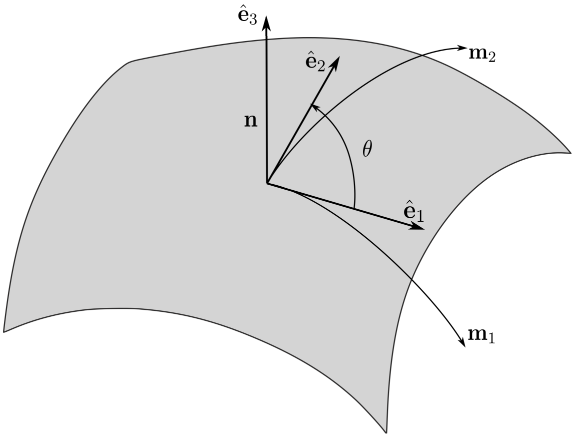


Figure 3.14: Local coordinate system on surface

3.3 Isogeometric boundary element methods

The implementation of isogeometric boundary element methods (IGABEM) is similar to classical BEM with isoparametric elements in that they discretize both the BIE and the geometry using the same shape functions. In isoparametric elements, the shape functions for representing geometries are borrowed from the field discretization, and thus can only approximate the geometry. In contrast, the element structures and basis functions in IGABEM are initially used for constructing CAD models, then utilized by analysis for discretizing the field around the boundary. The basis functions in IGABEM are rational and can guarantee an exact geometry. However, the lack in *Kronecker delta* property induces an inconvenience in the evaluation of strongly singular integrals and jump terms.

3.3.1 Isogeometric boundary element method formulations

The geometry provided by CAD is usually formulated using NURBS or T-spline basis functions

$$\mathbf{x}_e(\tilde{\boldsymbol{\xi}}) = \sum_{a=1}^{n_a} R_{ea}(\tilde{\boldsymbol{\xi}})\mathbf{P}_{ea}, \tag{3.3.118}$$

where R_{ea} are the NURBS or T-spline basis functions, with e denoting the parent element index and a the local index of the basis function in element e. $\mathbf{P}$ are the control points, and $\tilde{\boldsymbol{\xi}}$ the intrinsic coordinates of the field points in parent elements.

The displacement and traction fields around the boundary are also discretized using NURBS or T-splines, which is the main difference from the traditional boundary element method,

$$u_j^e(\tilde{\boldsymbol{\xi}}) = \sum_{a=1}^{n_a} R_{ea}(\tilde{\boldsymbol{\xi}})\tilde{u}_j^{ea}, \tag{3.3.119}$$

$$t_j^e(\tilde{\boldsymbol{\xi}}) = \sum_{a=1}^{n_a} R_{ea}(\tilde{\boldsymbol{\xi}})\tilde{t}_j^{ea}, \tag{3.3.120}$$

where $\tilde{u}_j^{ea}$ and $\tilde{t}_j^{ea}$ are the nodal displacement and traction parameters associated with control points, and $\tilde{\boldsymbol{\xi}}$ the intrinsic coordinates of the field points in the parent element. It is noted that $\tilde{u}_j^{ea}$ and $\tilde{t}_j^{ea}$ are not the nodal displacements or tractions because the basis functions lack the *Kronecker delta* property. Consequently, the boundary conditions cannot be enforced in the same way as BEM. An available approach is to semi-discretize the BIE [93], which rearranges the BIE (3.1.47) by separating the integrals into two sides

$$\begin{aligned} & C_{ij}(\mathbf{s})u_j\left(\mathbf{s}\right)\Big|_{S_t} + \frac{}{}\!\!\!\!\!\!\int_{S_t} T_{ij}\left(\mathbf{s},\mathbf{x}\right)u_j\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right) - \int_{S_u} U_{ij}\left(\mathbf{s},\mathbf{x}\right)t_j\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right) \\ = & -C_{ij}\left(\mathbf{s}\right)\bar{u}_j\left(\mathbf{s}\right)\Big|_{S_u} - \frac{}{}\!\!\!\!\!\!\int_{S_u} T_{ij}\left(\mathbf{s},\ \mathbf{x}\right)\bar{u}_j\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right) + \int_{S_t} U_{ij}\left(\mathbf{s},\ \mathbf{x}\right)\bar{t}_j\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right), \end{aligned} \tag{3.3.121}$$

where S_u denotes the portion of the boundary prescribed displacement boundary conditions, and S_t traction boundary conditions. Now u_j and t_j on the left-hand side of Eq. (3.3.121) are unknowns, and $\bar{u}_j$ and $\bar{t}_j$ on the right-hand side are the displacement and traction values given by boundary conditions.

We only discretize the left-hand side of Eq. (3.3.121) with Eqs. (3.3.119) and

(3.3.120), then a discretized form of BIE is obtained as

$$\begin{aligned}
&\sum_{a_0=1}^{n_{a_0}} C_{ij}(\tilde{\boldsymbol{\zeta}}_c)R_{e_0a_0}(\tilde{\boldsymbol{\zeta}}_c)\tilde{u}_j^{e_0a_0} + \sum_{e=1}^{n_e}\sum_{a=1}^{n_a} \fint_{\tilde{S}_t} T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_{ea}(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\tilde{u}_j^{ea} \\
&-\sum_{e=1}^{n_e}\sum_{a=1}^{n_a} \int_{\tilde{S}_u} U_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_{ea}(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\tilde{t}_j^{ea} \\
=&-\sum_{a_0=1}^{n_{a_0}} C_{ij}(\tilde{\boldsymbol{\zeta}}_c)\bar{u}_j^{e_0a_0}(\tilde{\boldsymbol{\zeta}}_c) - \sum_{e=1}^{n_e}\sum_{a=1}^{n_a} \fint_{\tilde{S}_u} T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})\bar{u}_j^{ea}(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}) \\
&+\sum_{e=1}^{n_e}\sum_{a=1}^{n_a} \int_{\tilde{S}_t} U_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})\bar{t}_j^{ea}(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}),
\end{aligned} \tag{3.3.122}$$

where c indicates the collocation point index, $\tilde{\boldsymbol{\zeta}}_c$ the intrinsic coordinate of the collocation point, e_0 the element in which the collocation point is located, and a_0 the local index of the collocation point in element e_0. In classical BEM, the collocation points are chosen on the nodes straightforwardly. This approach is not available, however, for IGABEM because the control points are not located on the boundary. A simple but effective approach for NURBS is using Greville abscissae, which compute the coordinates of collocation points in parametric space as follows

$$\zeta_A = \frac{\xi_{A+1} + \xi_{A+2} + \cdots + \xi_{A+p}}{p}. \tag{3.3.123}$$

This method is initially used in the isogeometric collocation method [4], then adopted in [93] for NURBS-based IGABEM in 2D linear elastic problems, and further extended to 3D IGABEM analysis with T-splines [87].

For convenience, we define some shorthand symbols to represent the terms in the left-hand side of Eq. (3.3.122) as

$$\hat{H}_{ij}^{ce_0a_0} = C_{ij}(\tilde{\boldsymbol{\zeta}}_c)R_{e_0a_0}(\tilde{\boldsymbol{\zeta}}_c), \tag{3.3.124}$$

$$\bar{H}_{ij}^{cea} = \fint_{\tilde{S}} T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_{ea}(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}), \tag{3.3.125}$$

$$G_{ij}^{cea} = \int_{\tilde{S}} U_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_{ea}(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}), \tag{3.3.126}$$

and the terms in the right-hand side of Eq. (3.3.122) as

$$\hat{h}_i^{ce_0a_0} = C_{ij}(\tilde{\boldsymbol{\zeta}}_c)\bar{u}_j^{e_0a_0}(\tilde{\boldsymbol{\zeta}}_c), \tag{3.3.127}$$

$$\bar{h}_i^{cea} = \fint_{\tilde{S}} T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})\bar{u}_j^{ea}(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}), \tag{3.3.128}$$

$$g_i^{cea} = \int_{\tilde{S}} U_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) \bar{t}_j^{ea}(\tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}). \tag{3.3.129}$$

With the above symbols, Eq. (3.3.122) can be expressed by

$$\begin{aligned} &\sum_{a_0=1}^{n_{a_0}} \hat{H}_{ij}^{ce_0a_0} \tilde{u}_j^{e_0a_0} + \sum_{e=1}^{n_e}\sum_{a=1}^{n_a} \bar{H}_{ij}^{cea} \tilde{u}_j^{ea} + \sum_{e=1}^{n_e}\sum_{a=1}^{n_a} G_{ij}^{cea} \tilde{t}_j^{ea} \\ =& \sum_{a_0=1}^{n_{a_0}} \hat{h}_i^{ce_0a_0} + \sum_{e=1}^{n_e}\sum_{a=1}^{n_a} \bar{h}_i^{cea} + \sum_{e=1}^{n_e}\sum_{a=1}^{n_a} g_i^{cea}. \end{aligned} \tag{3.3.130}$$

Eq. (3.3.130) can be further assembled into a matrix form as

$$\mathbf{A}\mathbf{z} = \mathbf{f}. \tag{3.3.131}$$

Matrix **A** contains the entries of the left-hand side of Eq. (3.3.130), **f** is a column vector from the right-hand side of Eq. (3.3.130), and **z** includes all the unknowns of displacements and tractions. The above linear discrete system is directly obtained from the integrals, and the boundary conditions have been taken into account when assembling the system matrix.

3.3.2 Evaluation of strongly singular integral

In BEM, jump terms and strongly singular integrals are not evaluated explicitly but bypassed through the rigid body motion method. However, the rigid body motion method is available only if the following requirements are satisfied:

- The terms containing jump terms should be arranged along the diagonal line of the matrices.

- The strongly singular integral terms should be arranged along the diagonal line of matrices.

Unfortunately, IGABEM impinges on both of the two requirements. Considering the terms containing the product of jump terms and shape functions

$$\hat{H}_{ij}^{cA} = C_{ij}(\tilde{\boldsymbol{\zeta}}_c) R_A(\tilde{\boldsymbol{\zeta}}_c), \tag{3.3.132}$$

where c is the collocation point index and determines the row index of the term of $\hat{H}_{ij}^{cA}$ in matrix **H**, and A is the shape function index and determines the column number. Unlike in BEM where only jump terms are present in $\mathbf{H}$, in IGABEM $\mathbf{H}$ contains the product of jump terms with the values of shape functions R_A on collocation points, so the distribution of $\hat{H}_{ij}$ in matrix $\mathbf{H}$ depends on the element supporting R_A. Due to the lack of the *Kronecker delta* property, the values of R_A normally do not vanish on the edges of the neighbouring elements, so the distribution of $\hat{H}_{ij}$ will not be restricted to the diagonal line only.

Now consider the strongly singular integrals written with the global node index,

$$\bar{H}_{ij}^{ca} = \fint_{\tilde{S}} T_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) R_A(\tilde{\boldsymbol{\xi}}) J^e(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}). \tag{3.3.133}$$

For BEM, the above integral is singular only when $c = A$. If $c \neq A$, polynomial shape functions vanish at collocation points and thus cancel the singularity of the kernels. In IGABEM, however, it cannot be guaranteed that $R_A(\tilde{\boldsymbol{\zeta}}_c) = 0$ when $c \neq A$, so the singularity of the kernel cannot be cancelled and the singular integrals must be evaluated explicitly. A classical technique to evaluate strongly singular integrals is the singularity subtraction technique (SST) [1,46,96], which will be illustrated here for two-dimensional cases. We denote the integrand in Eq. (3.3.125) as

$$F(\tilde{\zeta}, \tilde{\xi}) = \frac{f(\tilde{\xi})}{\tilde{\xi} - \tilde{\zeta}} = T_{ij}(\tilde{\zeta}, \tilde{\xi}) R_{ea}(\tilde{\xi}) J_e(\tilde{\xi}), \tag{3.3.134}$$

which can be expanded at the source point in power series as

$$F(\tilde{\zeta}, \tilde{\xi}) = \frac{f(\tilde{\xi})}{\tilde{\xi} - \tilde{\zeta}} = \frac{f(\tilde{\zeta})}{\tilde{\xi} - \tilde{\zeta}} + \mathcal{O}(1), \tag{3.3.135}$$

where $\tilde{\zeta}$ is the coordinate of the collocation point in parent elements. In our work, we only take the first term of the series, which is sufficient for cancelling the strong singularity. Higher order terms can be used for evaluating hypersingular integrals if necessary. Denoting the integral of Eq. (3.3.125) by I, it can be written by subtracting the first term of the Taylor series and adding it back as follows

$$I = \sum_{e=1}^{n_e} \int_{-1}^{1} \left[F_e(\tilde{\zeta}, \tilde{\xi}) - \frac{f_e(\tilde{\zeta})}{\tilde{\xi} - \tilde{\zeta}} + \frac{f_e(\tilde{\zeta})}{\tilde{\xi} - \tilde{\zeta}} \right] \mathrm{d}\tilde{\xi}. \tag{3.3.136}$$

The integral I consists of two parts,

$$I = I_0 + I_{-1}, \tag{3.3.137}$$

where I_0 is

$$I_0 = \sum_{e=1}^{n_e} \int_{-1}^{1} \left[F_e(\tilde{\zeta}, \tilde{\xi}) - \frac{f_e(\tilde{\zeta})}{\xi - \tilde{\zeta}} \right] \mathrm{d}\tilde{\xi}, \tag{3.3.138}$$

and I_{-1} is

$$I_{-1} = \sum_{e=1}^{n_e} \int_{-1}^{1} \frac{f_e(\tilde{\zeta})}{\tilde{\xi} - \tilde{\zeta}} \mathrm{d}\tilde{\xi}. \tag{3.3.139}$$

Note that the integrand of I_0 is now regular. And I_{-1} can be integrated analytically as illustrated in the following.

Supposing the collocation point is located in the middle of the parent elements Γ_1 and Γ_2 (Fig. 3.15), we only focus on the integrals over the two elements where singularity arises.

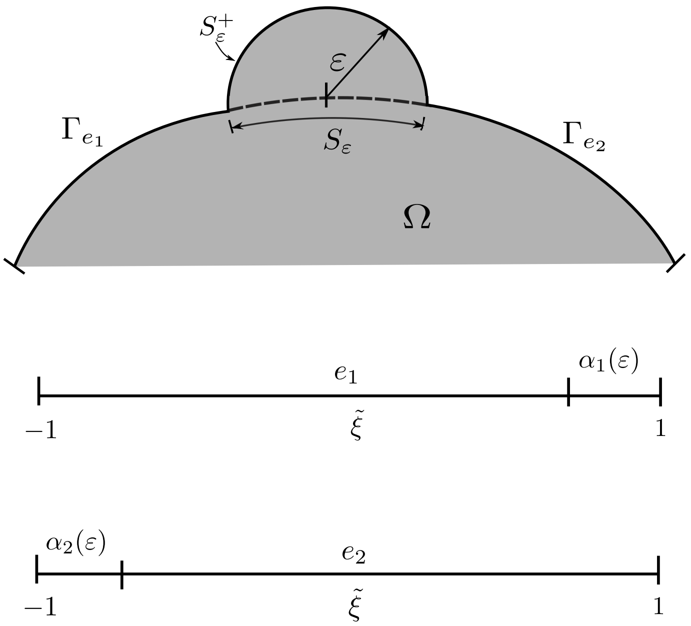


Figure 3.15: Limiting process in SST

By noticing that $\tilde{\zeta} = 1$ for element 1, $\tilde{\zeta} =$ -1 for element 2, and

$$\alpha(\varepsilon) = \varepsilon \beta_e(\tilde{\xi}) + \mathcal{O}(1). \tag{3.3.140}$$

we can obtain the expression for I_{-1} on the two elements

$$I_{-1} = \lim_{\varepsilon \to 0} \left(\int_{-1}^{1-\alpha_1(\varepsilon)} \frac{f_1(1)}{\tilde{\xi} - 1} \mathrm{d}\tilde{\xi} + \int_{-1+\alpha_2(\varepsilon)}^{1} \frac{f_2(-1)}{\tilde{\xi} + 1} \mathrm{d}\tilde{\xi} \right)$$

$$
\begin{aligned}
&= \lim_{\varepsilon\to 0}\left(f_1(1)\int_{-1}^{1-\alpha_1(\varepsilon)}\frac{1}{\tilde{\xi}-1}\mathrm{d}\tilde{\xi} + f_2(-1)\int_{-1+\alpha_2(\varepsilon)}^{1}\frac{1}{\tilde{\xi}+1}\mathrm{d}\tilde{\xi}\right)\\
&= \lim_{\varepsilon\to 0}\left(f_1(1)\ln\left|\frac{-\alpha_1(\varepsilon)}{-2}\right| + f_2(-1)\ln\left|\frac{2}{\alpha_2(\varepsilon)}\right|\right)\\
&= f_1(1)\lim_{\varepsilon\to 0}\left(\ln\left|\frac{\varepsilon\beta_1}{2}\right| + \ln\left|\frac{2}{\varepsilon\beta_2}\right|\right)\\
&= f_1(1)\lim_{\varepsilon\to 0}(\ln|\varepsilon| + \ln|\beta_1| - \ln|2| + \ln|2| - \ln|\varepsilon| - \ln|\beta_2|)\\
&= f_1(1)(\ln|\beta_1| - \ln|\beta_2|), \qquad (3.3.141)
\end{aligned}
$$

where $\beta_e = 1/J_e(\xi)$, only depends on the local geometry of the source point.

In the case where the collocation point is located inside of the element, the Cauchy integral can be calculated in a similar way by dividing the element into two subelements

$$
\begin{aligned}
I_{-1} &= \lim_{\varepsilon\to 0}\left(\int_{-1}^{\tilde{\zeta}-\alpha_1(\varepsilon)}\frac{f_1(1)}{\tilde{\xi}-\tilde{\zeta}}\mathrm{d}\tilde{\xi} + \int_{\tilde{\zeta}+\alpha_2(\varepsilon)}^{1}\frac{f_2(-1)}{\tilde{\xi}-\tilde{\zeta}}d\tilde{\xi}\right)\\
&= \lim_{\varepsilon\to 0}\left(f_1(1)\int_{-1}^{\tilde{\zeta}-\alpha_1(\varepsilon)}\frac{1}{\tilde{\xi}-\tilde{\zeta}}\mathrm{d}\tilde{\xi} + f_2(-1)\int_{\tilde{\zeta}+\alpha_2(\varepsilon)}^{1}\frac{1}{\tilde{\xi}-\tilde{\zeta}}d\tilde{\xi}\right)\\
&= \lim_{\varepsilon\to 0}\left(f_1(1)\ln\left|\frac{-\alpha_1(\varepsilon)}{-1-\tilde{\zeta}}\right| + f_2(-1)\ln\left|\frac{1-\tilde{\zeta}}{\alpha_2(\varepsilon)}\right|\right)\\
&= f_1(1)\lim_{\varepsilon\to 0}\left(\ln\left|\frac{\varepsilon\beta}{1+\tilde{\zeta}}\right| + \ln\left|\frac{1-\tilde{\zeta}}{\varepsilon\beta}\right|\right)\\
&= f_1(1)\lim_{\varepsilon\to 0}(\ln|\varepsilon\beta| - \ln|1+\tilde{\zeta}| + \ln|1-\tilde{\zeta}| - \ln|\varepsilon\beta|)\\
&= f_1(1)\left(\ln\left|\frac{1-\tilde{\zeta}}{1+\tilde{\zeta}}\right|\right). \qquad (3.3.142)
\end{aligned}
$$

The idea is the same for three-dimensional problems, although the formulations are different.

3.3.3 Evaluation of jump terms

Recall that jump terms arising from the limiting process

$$C_{ij}(\mathbf{s}) = \delta_{ij}(\mathbf{s}) + \alpha_{ij}(\mathbf{s}), \qquad (3.3.143)$$

with

$$\alpha_{ij} = \lim_{\varepsilon\to 0}\int_{S_\varepsilon^+} T_{ij}(\mathbf{s},\ \mathbf{x})\mathrm{d}S(\mathbf{x}), \qquad (3.3.144)$$

where S_ε^+ is a semi-circular arc added to the initial boundary S, and ε is the radius of the arc. See Fig. 3.16. Transferring the above to a polar system centered at source point $\mathbf{s}$, then we have

$$r = \varepsilon \cos\theta + \varepsilon \sin\theta, \tag{3.3.145}$$

$$dS_\varepsilon^+ = \varepsilon \mathrm{d}\theta, \tag{3.3.146}$$

$$\mathbf{r}_{,1} = \cos\theta, \tag{3.3.147}$$

$$\mathbf{r}_{,2} = \sin\theta. \tag{3.3.148}$$

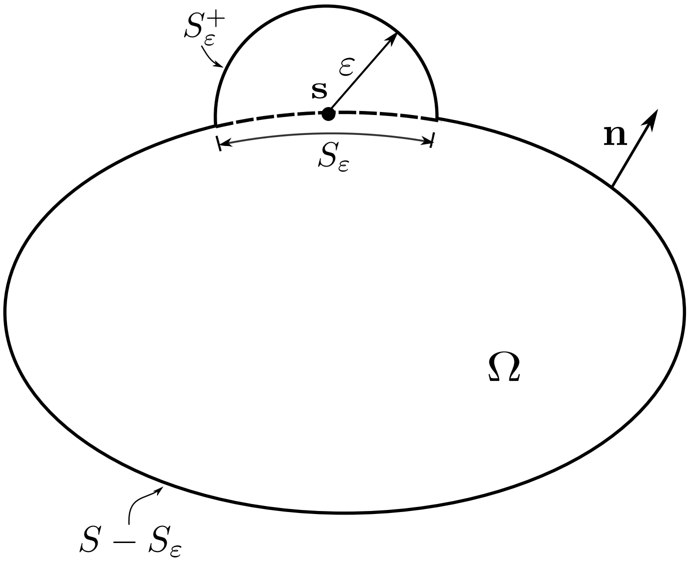


Figure 3.16: Jump terms in limiting process when $\varepsilon \to 0$

Using the above relationships, the constant α_{ij} can be evaluated as follows (Fig. 3.17),

$$\begin{aligned}\alpha_{11} &= \frac{-1}{4\pi(1-\nu)} \lim_{\varepsilon\to 0} \int_{S_\varepsilon} \frac{1}{\varepsilon}[(1-2\nu) + 2r_{,1}r_{,2}]\varepsilon \mathrm{d}\theta \\ &= \frac{-1}{4\pi(1-\nu)} \int_0^\pi [(1-2\nu) + 2\cos^2\theta]\mathrm{d}\theta \\ &= \frac{1}{8\pi(1-\bar{\nu})}[4(1-\nu)(\theta_1 - \theta_2) + (\sin 2\theta_1 - \sin 2\theta_2)].\end{aligned} \tag{3.3.149}$$

Following a similar procedure, we get

$$\alpha_{12} = \frac{-1}{4\pi(1-\nu)} \int_0^\pi 2\sin\theta\cos\theta \mathrm{d}\theta = \cos 2\theta_2 - \cos 2\theta_1, \tag{3.3.150}$$

$$\alpha_{21} = \frac{-1}{4\pi(1-\nu)} \int_0^\pi 2\sin\theta\cos\theta \mathrm{d}\theta = \frac{-1}{4\pi(1-\nu)}\cos 2\theta, \tag{3.3.151}$$

$$\begin{aligned}\alpha_{22} &= \frac{-1}{4\pi(1-\nu)} \int_0^\pi [(1-2\nu)+2\sin^2\theta]\mathrm{d}\theta \\ &= \frac{1}{8\pi(1-\nu)}[4(1-\bar{\nu})(\theta_1-\theta_2)-(\sin 2\theta_1 - \sin 2\theta_2)].\end{aligned} \tag{3.3.152}$$

So the matrix form of the jump term is

$$\mathbf{C} = \frac{1}{8\pi(1-\nu)} \begin{bmatrix} 4(1-\nu)(\theta_1-\theta_2) & \cos 2\theta_2 - \cos 2\theta_1 \\ +(\sin 2\theta_1 - \sin 2\theta_2), & \\ \cos 2\theta_2 - \cos 2\theta_1, & 4(1-\nu)(\theta_1-\theta_2) \\ & -(\sin 2\theta_1 - \sin 2\theta_2) \end{bmatrix}. \tag{3.3.153}$$

For three-dimensional problems the procedure is similar, but the integral is over a hemisphere S_ε.

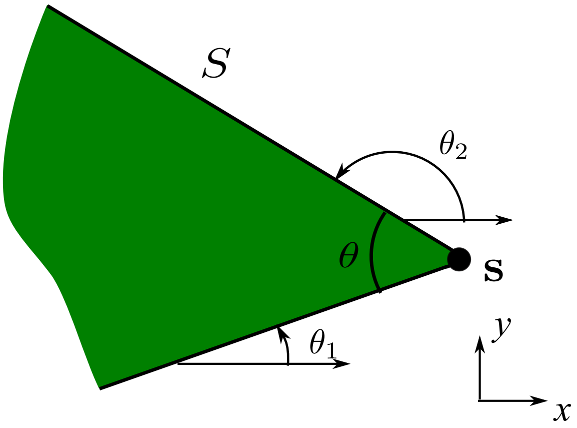


Figure 3.17: Evaluation of jump terms

3.3.4 Treatment of corners

Similar to the BEM for treatment of edges and corners, the collocation points of the IGABEM can be shifted from the edges and corners. However, the control points given from CAD are typically located on the corners. Hence, we can split the control points on the corners into several overlapped nodes. See Fig. 3.18.

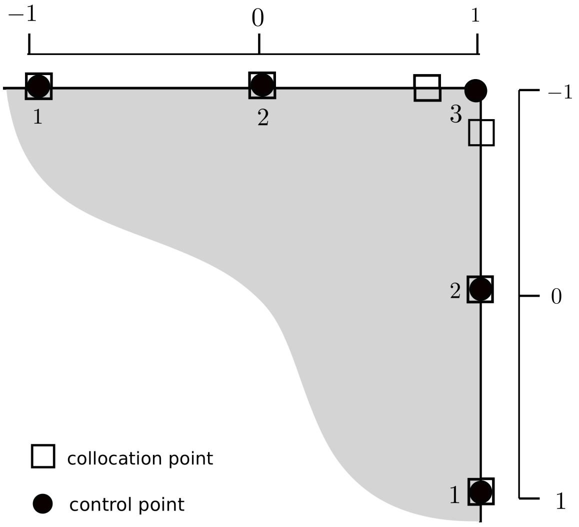


Figure 3.18: Nodes and collocation points around corners

3.4 Regularized isogeometric boundary element methods

3.4.1 Regularized form of boundary integral equations

Fig. 3.19 illustrates the implementation procedure of the IGABEM which is based on the singular form of the BIE. Through the above, the inconvenience of the evaluation of jump terms and strongly singular integral in IGABEM is obvious. To facilitate the implementation, we propose to adopt a regularized boundary integral equation given in [65–67] in the context of IGABEM, where there is no jump terms or strong singularity. The regularized form of the displacement boundary integral equation is written as

$$\int_S T_{ij}\left(\mathbf{s},\ \mathbf{x}\right)\left[u_j\left(\mathbf{x}\right)-u_j\left(\mathbf{s}\right)\right]\mathrm{d}S\left(\mathbf{x}\right)=\int_S U_{ij}\left(\mathbf{s},\ \mathbf{x}\right)t_j\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right). \tag{3.4.154}$$

The order of the integrand in the first integral is

$$T_{ij}\left(\mathbf{s},\ \mathbf{x}\right)\left[u_j\left(\mathbf{x}\right)-u_j\left(\mathbf{s}\right)\right]\sim\begin{cases}\mathcal{O}\left(\frac{1}{r}\right)\mathcal{O}(r)=\mathcal{O}(1) & \text{in 2D,}\\ \mathcal{O}\left(\frac{1}{r^2}\right)\mathcal{O}(r)=\mathcal{O}\left(\frac{1}{r}\right) & \text{in 3D.}\end{cases} \tag{3.4.155}$$

Consequently, in two-dimension problems, the regularized form can completely remove the singularity of the integrals containing traction kernel. In three-dimensions,

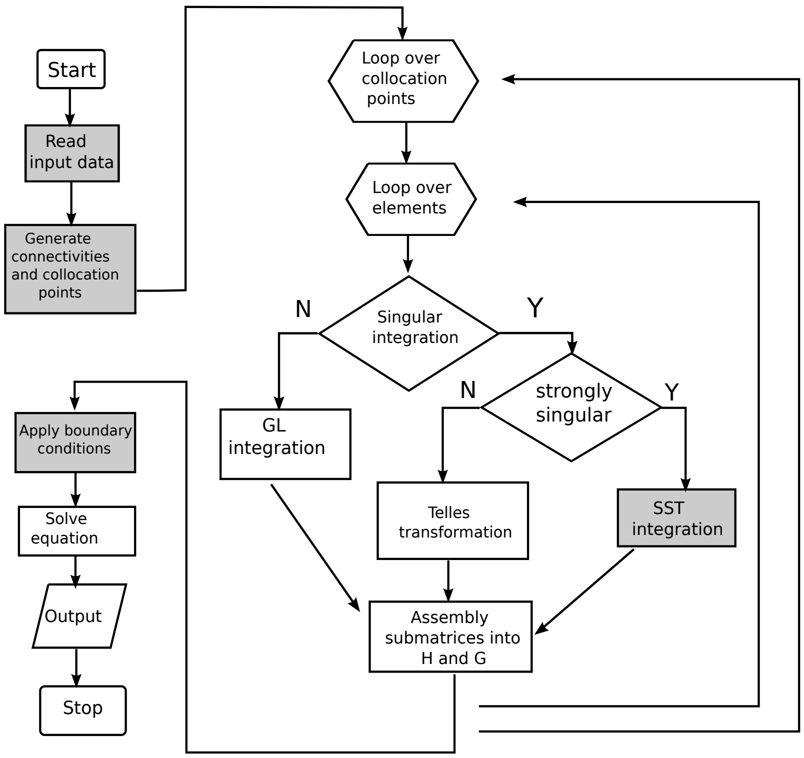


Figure 3.19: Singular BIE based IGABEM analysis flowchart

the integral becomes weakly singular, which can be treated easily using polar integration.

3.4.2 Regularized IGABEM formulations

Using the same discretization scheme as Eqs. (3.3.119 - 3.3.120), the regularized BIE can be discretized as

$$\sum_{e=1}^{n_e} \int_{\tilde{S}} T_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) \left[\sum_{a=1}^{n_a} R_{ea}(\tilde{\boldsymbol{\xi}}) \tilde{u}_j^{ea} - \sum_{a_0=1}^{n_{a_0}} R_{e_0 a_0}(\tilde{\boldsymbol{\zeta}}_c) \tilde{u}_j^{e_0 a_0} \right] J_e(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})$$

$$= \sum_{e=1}^{n_e} \int_{\tilde{S}} U_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) \sum_{a=1}^{n_a} R_{ea}(\tilde{\boldsymbol{\xi}}) \tilde{t}_j^{ea} J_e(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}). \qquad (3.4.156)$$

Because the integrals are evaluated numerically using Gauss-Legendre quadrature rule which is a summation form, we can rewrite Eq. (3.4.156) by splitting its left-hand side as,

$$\sum_{e=1}^{n_e} \sum_{a=1}^{n_a} \left\{ \int_{\tilde{S}} T_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) R_{ea}(\tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}) \right\} \tilde{u}_j^{ea}$$

$$
\begin{aligned}
&-\sum_{e=1}^{n_e}\sum_{a_0=1}^{n_{a_0}}\left\{\int_{\tilde{S}} T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_{e_0a_0}(\tilde{\boldsymbol{\zeta}}_c)J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\right\}\tilde{u}_j^{e_0a_0}\\
&=\sum_{e=1}^{n_e}\sum_{a=1}^{n_a}\left\{\int_{\tilde{S}} U_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_{ea}(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}}))\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\right\}\tilde{t}_j^{ea}. \qquad (3.4.157)
\end{aligned}
$$

With the following definitions,

$$\bar{H}_{ij}^{cea} = \int_{\tilde{S}} T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_{ea}(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}, \qquad (3.4.158)$$

$$\hat{H}_{ij}^{ce_0a_0} = \int_{\tilde{S}} T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_{e_0a_0}(\tilde{\boldsymbol{\zeta}}_c)J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}, \qquad (3.4.159)$$

$$G_{ij}^{cea} = \int_{\tilde{S}} U_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_{ea}(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}, \qquad (3.4.160)$$

Eq. (3.4.157) can be written as

$$\sum_{a_0=1}^{n_{a_0}}\hat{H}_{ij}^{ce_0a_0}\tilde{u}_j^{e_0a_0}+\sum_{e=1}^{n_e}\sum_{a=1}^{n_a}\bar{H}_{ij}^{cea}\tilde{u}_j^{ea}=\sum_{e=1}^{n_e}\sum_{a=1}^{n_a}G_{ij}^{cea}\tilde{t}_j^{ea}. \qquad (3.4.161)$$

Using the mapping from the local node index space to the global index space, a matrix form is generated as

$$\mathbf{Hu} = \mathbf{Gt}. \qquad (3.4.162)$$

By swapping the unknowns of the two sides of the above matrix equation, we can get

$$\mathbf{Az} = \mathbf{f}. \qquad (3.4.163)$$

Remark that the equation above is identical to Eq. (3.3.131) generated from singular form of BIE, but the matrix assembly procedure is simplified greatly. Moreover, the regularized form has almost no difference between 2D and 3D, whereas the formula of SST is different depending on the number of spatial dimensions and the evaluation of jump terms in 3D is not trivial.

3.4.3 Imposition of boundary conditions

As mentioned above, the basis functions in IGABEM lack the *Kronecker delta* property, so the nodal parameters do not possess a clear physical interpretation. Hence, the values prescribed by boundary conditions cannot be substituted directly into the governing matrix equation. In the isogeometric finite element method, imposing boundary conditions can be done through Lagrange multiplier methods, penalty methods, or Nitsche methods [77]. However, these methods are not available in collocation IGABEM, because it is not based on a variational equality. Hence, we herein propose a nodal parameter extraction technique, which can be conducted by collocation methods or Galerkin methods.

Collocation method

The collocation method enforces boundary conditions at a series of discrete points. To construct the equations, we collocate at a series of points on the boundary portion prescribed boundary conditions, and evaluate the field values,

$$\mathbf{u}(\tilde{\boldsymbol{\xi}}_c) = \bar{\mathbf{u}}(\tilde{\boldsymbol{\xi}}_c) \qquad \text{on } S_u, \tag{3.4.164}$$

$$\mathbf{t}(\tilde{\boldsymbol{\xi}}_c) = \bar{\mathbf{t}}(\tilde{\boldsymbol{\xi}}_c) \qquad \text{on } S_t, \tag{3.4.165}$$

where S_u is the portion of the boundary with displacement boundary conditions, and S_t with traction boundary conditions. $\tilde{\boldsymbol{\xi}}_c$ denotes the collocation point with index c, which can be identical to that used for constructing IGABEM equations.

Substituting Eqs.(3.3.119) and (3.3.120) into the above equations and using a matrix form produces

$$\mathbf{R}(\tilde{\boldsymbol{\xi}}_c)\tilde{\mathbf{u}} = \bar{\mathbf{u}}(\tilde{\boldsymbol{\xi}}_c) \qquad \text{on } S_u, \tag{3.4.166}$$

$$\mathbf{R}(\tilde{\boldsymbol{\xi}}_c)\tilde{\mathbf{t}} = \bar{\mathbf{t}}(\tilde{\boldsymbol{\xi}}_c) \qquad \text{on } S_t, \tag{3.4.167}$$

where $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{t}}$ are the column vectors collecting the components of boundary nodal parameters. $\mathbf{R} = R\mathbf{I}$ is the shape function matrix with R denoting the shape function and $\mathbf{I}$ the identity matrix.

After obtaining the $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{t}}$ by solving Eqs. (3.4.166, 3.4.167), we can substitute them into the governing equations for analysis.

Galerkin method

The Galerkin method enforces boundary conditions in an “average” sense, *i.e.*

$$\int_{S_u} \mathbf{R}^{\mathrm{T}}\mathbf{u}\mathrm{d}S = \int_{S_u} \mathbf{R}^{\mathrm{T}}\bar{\mathbf{u}}\mathrm{d}S \qquad \text{on } S_u, \tag{3.4.168}$$

$$\int_{S_t} \mathbf{R}^{\mathrm{T}}\mathbf{t}\mathrm{d}S = \int_{S_t} \mathbf{R}^{\mathrm{T}}\bar{\mathbf{t}}\mathrm{d}S \qquad \text{on } S_t, \tag{3.4.169}$$

where the shape functions $\mathbf{R}$ are used as the weighting functions. Substituting Eqs. (3.3.119) and (3.3.120) into the above equations leads to

$$\int_{S_u} \mathbf{R}^{\mathrm{T}}\mathbf{R}\tilde{\mathbf{u}}\mathrm{d}S = \int_{S_u} \mathbf{R}^{\mathrm{T}}\bar{\mathbf{u}}\mathrm{d}S \qquad \text{on } S_u, \tag{3.4.170}$$

$$\int_{S_t} \mathbf{R}^{\mathrm{T}}\mathbf{R}\tilde{\mathbf{t}}\mathrm{d}S = \int_{S_t} \mathbf{R}^{\mathrm{T}}\bar{\mathbf{t}}\mathrm{d}S \qquad \text{on } S_t. \tag{3.4.171}$$

Hence, $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{t}}$ can be obtained by solving the following matrix equations

$$\mathbf{A}_1\tilde{\mathbf{u}} = \mathbf{z}_1 \qquad \text{on } S_u, \tag{3.4.172}$$

$$\mathbf{A}_2\tilde{\mathbf{t}} = \mathbf{z}_2 \qquad \text{on } S_t, \tag{3.4.173}$$

where

$$\mathbf{A}_1 = \int_{S_u} \mathbf{R}^{\mathrm{T}}\mathbf{R}\mathrm{d}S \qquad \text{on } S_u, \tag{3.4.174}$$

$$\mathbf{A}_2 = \int_{S_t} \mathbf{R}^{\mathrm{T}}\mathbf{R}\mathrm{d}S \qquad \text{on } S_t, \tag{3.4.175}$$

and

$$\mathbf{z}_1 = \int_{S_u} \mathbf{R}^{\mathrm{T}}\bar{\mathbf{u}}\mathrm{d}S \qquad \text{on } S_u, \tag{3.4.176}$$

$$\mathbf{z}_2 = \int_{S_t} \mathbf{R}^{\mathrm{T}}\bar{\mathbf{t}}\mathrm{d}S \qquad \text{on } S_t. \tag{3.4.177}$$

Requiring no integration, the collocation method is more efficient than the Galerkin method. However, the Galerkin method is more elegant in the case of geometries with corners where special care must be taken for choosing collocation points. Moreover, collocation methods may lead to instabilities along interfaces between Dirichlet and Neumann boundary conditions since the equations are satisfied only poinwise.

Nodal displacement and traction parameters associated with the control points can be extracted before the IGABEM analysis. This is a separate process conducted before the system matrix assembly. However, once the nodal parameters are extracted, they can be substituted into the matrix equation directly to enforce boundary conditions, as the conventional boundary element methods.

3.5 Conclusions

A formulation of the isogeometric boundary element method was presented. The singular form of the BIE was widely used in the BEM with isoparametric elements, but based on this form, the explicit evaluation of jump terms and strongly singular integrals is needed in IGABEM. So an IGABEM formulation based on the regularized BIE is proposed, which can remove jump terms and strongly singular integrals from the equation. Moreover, the nodal parameter extraction method is introduced to further facilitate the imposition of boundary conditions.

Chapter 4

Shape Sensitivity Analysis with IGABEM

Shape sensitivity analysis refers to the evaluation of the derivatives of quantities of interest with respect to design variables. This is a critical step for gradient-based shape optimization, although its application is not limited to it. In the context of boundary integral equations, three methods are available to conduct sensitivity analysis, 1) finite difference methods, 2) adjoint variable methods [29, 68], and 3) implicit differentiation methods [6,27,53]. Finite difference methods are very easy to implement but in accuracy is limited. The adjoint variable methods use an adjoint state to obtain sensitivity expression for each design variable and are particularly useful for the optimization problem with a large number of design variables but a small number of constraints. However, adjoint variables normally correspond to a concentrated point force, which is not consistent with distributed tractions used in BEM. The concentrated force is thus approximated using a traction exerted on a small area, which decreases the accuracy and robustness of the algorithm. Implicit differentiation methods rely on a direct differentiation of the BIE with respect to the design variables, and generate analytical forms of the BIE sensitivities. Due to its accuracy and convenience for BIE, the present work will employ the implicit differentiation method, and use regularized BIE to generate the material differentiation form.

In this chapter, we will firstly review the concept of material derivatives. Then we introduce the formulation of implicit differentiation method in BEM for sensitivity analysis. After that, the formulation of the IGABEM in shape sensitivity analysis is detailed, followed by the numerical examples in two-dimensional and three-dimensional problems.

4.1 Material derivatives

In shape sensitivity analysis and shape optimization, material coordinates will be used because they do not change with the geometry deformation. The design variable τ is a time-like quantity, determining the current configuration of the geometry. The physical coordinates $\mathbf{x}$ are dependent on both material coordinates and design variables through a mapping

$$T : \Omega_0 \to \Omega_\tau, \tag{4.1.1}$$

$$\mathbf{x}_\tau \equiv T\left(\mathbf{x}_0, \tau\right). \tag{4.1.2}$$

The operator T denotes the mapping from the reference configuration to the deformed configuration, Ω_0 the reference configuration with material coordinates $\mathbf{x}_0$, Ω_τ the current configuration with physical coordinates $\mathbf{x}_\tau$. See Fig. 4.1.

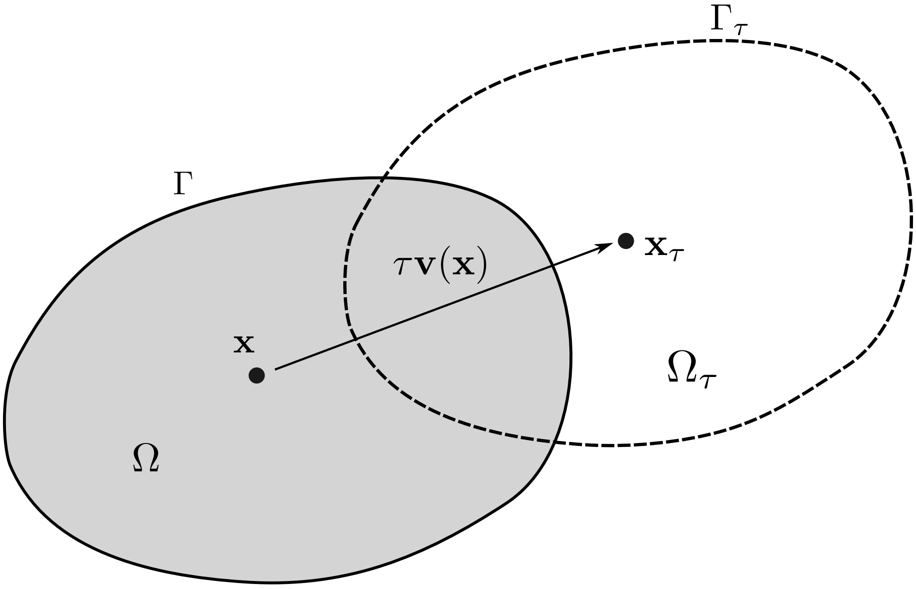


Figure 4.1: Shape perturbation by design parameter with design “velocity” $\mathbf{v}(\mathbf{x})$

A design velocity field associated with the mapping T can be defined as

$$v\left(\mathbf{x}_{\tau},\tau\right)\equiv\frac{d\mathbf{x}_{\tau}}{d\tau}=\frac{\partial T\left(\mathbf{x},\tau\right)}{\partial\tau}. \tag{4.1.3}$$

If the mapping $T\left(\mathbf{x},\tau\right)$ is assumed to be regular enough in the neighbourhood of $\tau=0$, then it can be expanded using the Taylor series as

$$T\left(\mathbf{x},\tau\right)=T\left(\mathbf{x},0\right)+\tau\frac{\partial T}{\partial\tau}\left(\mathbf{x}_0,0\right)+\cdots=\mathbf{x}+\tau v\left(\mathbf{x}_0,0\right)+\cdots. \tag{4.1.4}$$

By ignoring higher order terms, the linear mapping relation is obtained as

$$T\left(\mathbf{x},\tau\right)=\mathbf{x}+\tau\mathbf{v}\left(\mathbf{x},0\right). \tag{4.1.5}$$

Then, the shape sensitivity of $\mathbf{z}$ is the material derivatives of $\mathbf{z}$ as

$$\dot{\mathbf{z}}=\frac{d}{d\tau}\mathbf{z}_{\tau}\left(\mathbf{x}+\tau v\left(\mathbf{x}\right)\right)|_{\tau=0}=\lim_{\tau\to 0}\left[\frac{\mathbf{z}_{\tau}\left(\mathbf{x}+\tau v\left(\mathbf{x}\right)\right)-\mathbf{z}\left(\mathbf{x}\right)}{\tau}\right]. \tag{4.1.6}$$

Then, Eq. (4.1.6) can be separated into two terms as

$$\begin{aligned}\dot{\mathbf{z}}\left(\mathbf{x}\right)&=\lim_{\tau\to 0}\left[\frac{\mathbf{z}_{\tau}\left(\mathbf{x}+\tau v\left(\mathbf{x}\right)\right)-\mathbf{z}\left(\mathbf{x}\right)}{\tau}\right]\\&=\lim_{\tau\to 0}\left[\frac{\mathbf{z}_{\tau}\left(\mathbf{x}\right)-\mathbf{z}\left(\mathbf{x}\right)}{\tau}\right]+\lim_{\tau\to 0}\left[\frac{\mathbf{z}_{\tau}\left(\mathbf{x}+\tau v\left(\mathbf{x}\right)\right)-\mathbf{z}\left(\mathbf{x}\right)}{\tau}\right]\\&=\mathbf{z}'\left(\mathbf{x}\right)+\nabla\mathbf{z}v\left(\mathbf{x}\right),\end{aligned} \tag{4.1.7}$$

where $\mathbf{z}'\left(\mathbf{x}\right)$ is the gradient with respect to the space coordinates and $\nabla\mathbf{z}v\left(\mathbf{x}\right)$ is the convective term.

According to the design velocity theory [28,55], the design velocity should depend linearly on the variation of the shape design variables. Hence, the movement of the material points can be specified by the optimization solver user provided that the above condition is satisfied, for example, through Laplacian smoothing, solving a virtual mechanical problem [5], etc. However, in (IGA)BEM the material points in the domain does not need to be taken into account, which is also an advantage over (IGA)FEM. Although the numerical examples in the chapter illustrated the sensitivity field in the domain, they are purely for verifying the sensitivity result and actually not needed in the optimization.

4.2 Implicit differentiation method in IGABEM

Now we introduce the implementation of implicit differentiation method which has been widely used in BEM for shape sensitivity analysis. The procedure is taking shape derivatives of the BIE

$$C_{ij}\left(\mathbf{s}\right)u_j\left(\mathbf{s}\right)+\fint_S T_{ij}\left(\mathbf{s},\ \mathbf{x}\right)u_j\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right)=\int_S U_{ij}\left(\mathbf{s},\ \mathbf{x}\right)t_j\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right). \tag{4.2.8}$$

to get the following form

$$\begin{aligned}
&\dot{C}_{ij}\left(\mathbf{s}\right)u_j\left(\mathbf{s}\right)+C_{ij}\left(\mathbf{s}\right)\dot{u}_j\left(\mathbf{s}\right)\\
&+\fint_S\left[\dot{T}_{ij}\left(\mathbf{s},\mathbf{x}\right)u_j\left(\mathbf{x}\right)+T_{ij}\left(\mathbf{s},\mathbf{x}\right)\dot{u}_j\left(\mathbf{x}\right)\right]\mathrm{d}S\left(\mathbf{x}\right)+\fint_S T_{ij}\left(\mathbf{s},\mathbf{x}\right)u_j\left(\mathbf{x}\right)[\mathrm{d}S^{\dot{}}(\mathbf{x})]\\
&=\int_S\left[\dot{U}_{ij}\left(\mathbf{s},\mathbf{x}\right)t_j\left(\mathbf{x}\right)+U_{ij}\left(\mathbf{s},\mathbf{x}\right)\dot{t}_j\left(\mathbf{x}\right)\right]\mathrm{d}S\left(\mathbf{x}\right)\\
&\quad+\int_S U_{ij}\left(\mathbf{s},\mathbf{x}\right)t_j\left(\mathbf{x}\right)[\mathrm{d}S^{\dot{}}(\mathbf{x})],
\end{aligned} \tag{4.2.9}$$

where superimposed dot $(\dot{\cdot})$ denotes the shape derivative with respect to the given design variable. The implicit differentiation method leads to an analytical expression, in contrast to the semi-analytical approach in FEM where the sensitivity is based on a discretized weak form.

The present work will employ implicit differentiation method combined with IGABEM, but has two main differences:

- The sensitivities of the displacement and traction fields are discretized by the NURBS or T-splines.

- The material differentiation is performed on a regularized BIE.

We have taken the regularized IGABEM for structural analysis as

$$\int_S T_{ij}\left(\mathbf{s},\ \mathbf{x}\right)\left[u_j\left(\mathbf{x}\right)-u_j\left(\mathbf{s}\right)\right]\mathrm{d}S\left(\mathbf{x}\right)=\int_S U_{ij}\left(\mathbf{s},\ \mathbf{x}\right)t_j\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right). \tag{4.2.10}$$

Now we take shape derivatives for both sides of the regularized BIE and obtain the following expression

$$\int_S\left\{\dot{T}_{ij}\left(\mathbf{s},\mathbf{x}\right)\left[u_j\left(\mathbf{x}\right)-u_j\left(\mathbf{s}\right)\right]+T_{ij}\left(\mathbf{s},\mathbf{x}\right)\left[\dot{u}_j\left(\mathbf{x}\right)-\dot{u}_j\left(\mathbf{s}\right)\right\}\mathrm{d}S\left(\mathbf{x}\right)$$

$$
\begin{aligned}
&+\int_S T_{ij}(\mathbf{s},\mathbf{x})\left[u_j(\mathbf{x})-u_j(\mathbf{s})\right][\mathrm{d}\dot{S}(\mathbf{x})] \\
&=\int_S \left[\dot{U}_{ij}(\mathbf{s},\mathbf{x})\,t_j(\mathbf{x})+U_{ij}(\mathbf{s},\mathbf{x})\,\dot{t}_j(\mathbf{x})\right]\mathrm{d}S(\mathbf{x}) \\
&+\int_S U_{ij}(\mathbf{s},\mathbf{x})\,t_j(\mathbf{x})\,[\mathrm{d}\dot{S}(\mathbf{x})]. \qquad (4.2.11)
\end{aligned}
$$

We remark that $\dot{T}_{ij}$ and $\dot{U}_{ij}$ share the same singularity order with T_{ij} and U_{ij} respectively. Hence, the equation is still without strong singularity.

We set the intrinsic coordinates in parent elements as the material coordinates, which are independent of the design variables. Thus the shape derivatives of the field points are

$$\dot{\mathbf{x}}_e(\tilde{\boldsymbol{\xi}})=\sum_{a=1}^{n_a}R_{ea}(\tilde{\boldsymbol{\xi}})\dot{\mathbf{P}}_{ea}, \qquad (4.2.12)$$

where R_{ea} denotes the basis functions of CAD in constructing geometric models, such as NURBS or T-splines.

We discretize the displacement field and traction field around the boundary using NURBS or T-splines

$$u_j^e(\tilde{\boldsymbol{\xi}})=\sum_{a=1}^{n_a}R_{ea}(\tilde{\boldsymbol{\xi}})\tilde{u}_j^{ea}, \qquad (4.2.13)$$

$$t_j^e(\tilde{\boldsymbol{\xi}})=\sum_{a=1}^{n_a}R_{ea}(\tilde{\boldsymbol{\xi}})\tilde{t}_j^{ea}, \qquad (4.2.14)$$

and we also discretize the shape derivatives of the boundary displacement and traction field using NURBS or T-splines as

$$\dot{u}_j^e(\tilde{\boldsymbol{\xi}})=\sum_{a=1}^{n_a}R_{ea}(\tilde{\boldsymbol{\xi}})\dot{\tilde{u}}_j^{ea}, \qquad (4.2.15)$$

$$\dot{t}_j^e(\tilde{\boldsymbol{\xi}})=\sum_{a=1}^{n_a}R_{ea}(\tilde{\boldsymbol{\xi}})\dot{\tilde{t}}_j^{ea}. \qquad (4.2.16)$$

By noticing

$$[\mathrm{d}S_e(\mathbf{x})]=J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}), \qquad (4.2.17)$$

$$[\mathrm{d}\dot{S}_e(\mathbf{x})]=\dot{J}_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}), \qquad (4.2.18)$$

we substitute Eqs. (4.2.13-4.2.18) to Eq. (4.2.11) and gain

$$\sum_{e=1}^{n_e}\int_{\tilde{S}}\left\{\dot{T}_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})\left[\sum_{a=1}^{n_a}R_{ea}(\tilde{\boldsymbol{\xi}})\tilde{u}_j^{ea}-\sum_{a_0=1}^{n_{a_0}}R_{e_0a_0}(\tilde{\boldsymbol{\zeta}}_c)\tilde{u}_j^{e_0a_0}\right]J_e(\tilde{\boldsymbol{\xi}})\right.$$

$$
\begin{aligned}
&+T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})\left[\sum_{a=1}^{n_a}R_{ea}(\tilde{\boldsymbol{\xi}})\tilde{u}_j^{ea}-\sum_{a_0=1}^{n_{a_0}}R_{e_0a_0}(\tilde{\boldsymbol{\zeta}}_c)\tilde{u}_j^{e_0a_0}\right]\dot{J}_e(\tilde{\boldsymbol{\xi}})\Bigg\}\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\\
&+\sum_{e=1}^{n_e}\int_{\tilde{S}}T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})\left[\sum_{a=1}^{n_a}R_{ea}(\tilde{\boldsymbol{\xi}})\dot{\tilde{u}}_j^{ea}-\sum_{a_0=1}^{n_{a_0}}R_{e_0a_0}(\tilde{\boldsymbol{\zeta}}_c)\dot{\tilde{u}}_j^{e_0a_0}\right]J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\\
=&\sum_{e=1}^{n_e}\sum_{a=1}^{n_a}\left\{\int_{\tilde{S}}\left[\dot{U}_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_{ea}(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})+U_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_{ea}(\tilde{\boldsymbol{\xi}})\dot{J}_e(\tilde{\boldsymbol{\xi}})\right]\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\right\}\tilde{t}_j^{ea}\\
&+\sum_{e=1}^{n_e}\sum_{a=1}^{n_a}\left\{\int_{\tilde{S}}U_{ij}(\tilde{\boldsymbol{\zeta}}_c,\boldsymbol{\xi})R_{ea}(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\right\}\dot{\tilde{t}}_j^{ea}. \qquad (4.2.19)
\end{aligned}
$$

Because the integrals will be evaluated numerically, the above equation can be rearranged as

$$
\begin{aligned}
&\sum_{e=1}^{n_e}\sum_{a=1}^{n_a}\int_{\tilde{S}}\left[\dot{T}_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_{ea}(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})+T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_{ea}(\tilde{\boldsymbol{\xi}})\dot{J}_e(\tilde{\boldsymbol{\xi}})\right]\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\tilde{u}_j^{ea}\\
&-\sum_{e=1}^{n_e}\sum_{a_0=1}^{n_{a_0}}\int_{\tilde{S}}\left[\dot{T}_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})+T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})\dot{J}_e(\tilde{\boldsymbol{\xi}})\right]\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})R_{e_0a_0}(\tilde{\boldsymbol{\zeta}}_c)\tilde{u}_j^{e_0a_0}\\
&+\sum_{e=1}^{n_e}\sum_{a=1}^{n_a}\int_{\tilde{S}}T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_{ea}(\tilde{\boldsymbol{\xi}})\dot{J}_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\dot{\tilde{u}}_j^{ea}\\
&-\sum_{e=1}^{n_e}\sum_{a_0=1}^{n_{a_0}}\int_{\tilde{S}}T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})R_{e_0a_0}(\tilde{\boldsymbol{\zeta}}_c)\dot{\tilde{u}}_j^{e_0a_0}\\
=&\sum_{e=1}^{n_e}\sum_{a=1}^{n_a}\left\{\int_{\tilde{S}}\left[\dot{U}_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_{ea}(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})+U_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_{ea}(\tilde{\boldsymbol{\xi}})\dot{J}_e(\tilde{\boldsymbol{\xi}})\right]\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\right\}\tilde{t}_j^{ea}\\
&+\sum_{e=1}^{n_e}\sum_{a=1}^{n_a}\left\{\int_{\tilde{S}}U_{ij}(\tilde{\boldsymbol{\zeta}}_c,\boldsymbol{\xi})R_{ea}(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\right\}\dot{\tilde{t}}_j^{ea}. \qquad (4.2.20)
\end{aligned}
$$

The above equation can be assembled to a matrix form in the same way as structural analysis, yielding the following form

$$\dot{\mathbf{H}}\mathbf{u}+\mathbf{H}\dot{\mathbf{u}}=\dot{\mathbf{G}}\mathbf{t}+\mathbf{G}\dot{\mathbf{t}}. \qquad (4.2.21)$$

where the displacement $\mathbf{u}$ and $\mathbf{t}$ are vectors containing the displacement and traction nodal parameters, and $\mathbf{H}$ and $\mathbf{G}$ are the corresponding coefficient matrices. These values can be obtained from the IGABEM structural analysis result. $\dot{\mathbf{H}}$ and $\dot{\mathbf{G}}$ are the coefficient matrices associated with the unknown field sensitivities $\dot{\mathbf{u}}$ and $\dot{\mathbf{t}}$. Matrix $\dot{\mathbf{H}}$ is assembled from $\dot{\hat{\mathbf{H}}}$ and $\dot{\bar{\mathbf{H}}}$, whose entries are given as follows:

$$\dot{\hat{H}}_{ij}^{ce_0a_0}=-\int_{\tilde{S}}[\dot{T}_{ij}(\tilde{\boldsymbol{\zeta}}_c,\boldsymbol{\xi})J_e(\boldsymbol{\xi})+T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\boldsymbol{\xi})\dot{J}_e(\boldsymbol{\xi})]\mathrm{d}\tilde{S}(\boldsymbol{\xi})R_{e_0a_0}(\tilde{\boldsymbol{\zeta}}_c), \qquad (4.2.22)$$

$$\dot{\bar{H}}_{ij}^{cea} = \int_{\tilde{S}} [\dot{T}_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) R_{ea}(\tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}}) + T_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) R_{ea}(\tilde{\boldsymbol{\xi}}) \dot{J}_e(\tilde{\boldsymbol{\xi}})] \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}). \quad (4.2.23)$$

And the entries in matrix $\dot{\mathbf{G}}$ are

$$\dot{G}_{ij}^{cea} = \int_{\tilde{S}} \left[\dot{U}_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) R_{ea}(\tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}}) + \int_{\tilde{S}} U_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) R_{ea}(\tilde{\boldsymbol{\xi}}) \dot{J}_e(\tilde{\boldsymbol{\xi}}) \right] \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}). \quad (4.2.24)$$

The boundary conditions of sensitivity analysis can be found from the material differentiation of the boundary conditions prescribed for structural analysis,

$$\begin{aligned} \dot{u}_j(\mathbf{x}) &= \dot{\bar{u}}_j(\mathbf{x}) \qquad \text{on } S_u, & (4.2.25) \\ \dot{t}_j(\mathbf{x}) &= \dot{\bar{t}}_j(\mathbf{x}) \qquad \text{on } S_t, & (4.2.26) \end{aligned}$$

where $\dot{\bar{u}}_j$ and $\dot{\bar{t}}_j$ are the displacement and traction sensitivity boundary conditions, respectively.

By swapping the unknowns in Eq. (4.2.21), a final matrix form is obtained as

$$\mathbf{A}\dot{\mathbf{z}} = \dot{\mathbf{f}} + \dot{\mathbf{A}}\mathbf{z}, \quad (4.2.27)$$

where the matrix $\mathbf{A}$ and column vector $\mathbf{z}$ are identical to that in IGABEM structural analysis, and $\dot{\mathbf{f}}$ is formed by imposing sensitivity boundary conditions.

Similar to structural analysis with IGABEM, the imposition of boundary conditions in shape sensitivity analysis also requires a special treatment. The nodal parameter extraction methods are still available but for the sensitivities of boundary conditions.

4.3 The sensitivities of fundamental solutions

The shape derivatives of fundamental solutions play a key role in shape sensitivity analysis in IGABEM. The analytical form can be obtained by taking shape derivatives on fundamental solutions.

4.3.1 Two-dimensional problems

The sensitivity of displacement and traction fundamental solutions $\dot{U}_{ij}$ and $\dot{T}_{ij}$ are

$$\dot{U}_{ij}(\mathbf{s},\ \mathbf{x}) = \frac{1}{8\pi\mu(1-\nu)} \left[(3-4\nu) \left(\dot{\overline{\ln \frac{1}{r}}} \right) \delta_{ij} + (\dot{r_{,i}}) r_{,j} + r_{,i}(\dot{r_{,j}}) \right], \quad (4.3.28)$$

$$\begin{aligned}\dot{T}_{ij}\left(\mathbf{s},\ \mathbf{x}\right) = & -\frac{1}{4\pi(1-\nu)}\dot{\left(\frac{1}{r}\right)}\left\{\frac{\partial r}{\partial n}[(1-2\nu)\delta_{ij}+2r_{,i}r_{,j}]\right\} \\ & -\frac{1}{4\pi(1-\nu)}\dot{\left(\frac{1}{r}\right)}[-(1-2\nu)(r_{,i}n_j-r_{,j}n_i)] \\ & -\frac{1}{4\pi(1-\nu)r}\left\{\dot{\left(\frac{\partial r}{\partial n}\right)}[(1-2\nu)\delta_{ij}+2r_{,i}r_{,j}]\right\} \\ & +\frac{1}{4\pi(1-\nu)r}\left\{-2\frac{\partial r}{\partial n}[\dot{(r_{,i})}r_{,j}+r_{,i}\dot{(r_{,j})}]\right\} \\ & +\frac{1}{4\pi(1-\nu)r}\{-(1-2\nu)[\dot{(r_{,i})}n_j+r_{,i}\dot{n}_j-\dot{(r_{,j})}n_i-r_{,j}\dot{n}_i]\}.\end{aligned} \tag{4.3.29}$$

where superimposed dot $\dot{(\cdot)}$ denotes the derivative with respect to the given design variable, and

$$\dot{\left(\frac{\partial r}{\partial n}\right)} = \dot{(r_{,i}n_i)} = \dot{(r_{,i})}n_i + r_{,i}\dot{n}_i, \quad \dot{\left(\frac{1}{r}\right)} = -\frac{\dot{r}}{r^2}, \tag{4.3.30}$$

$$\dot{(r_{,i})} = \dot{\left(\frac{x_i-s_i}{r}\right)} = \frac{(\dot{x}_i-\dot{s}_i)r-(x_i-s_i)\dot{r}}{r^2}, \quad \dot{\left(\ln\frac{1}{r}\right)} = -\frac{\dot{r}}{r}. \tag{4.3.31}$$

The Jacobian is

$$J(\tilde{\xi}) \;=\; \sqrt{J_i(\tilde{\xi})J_i(\tilde{\xi})}, \tag{4.3.32}$$

with

$$J_i(\tilde{\xi}) = \frac{\mathrm{d}x_i}{\mathrm{d}\tilde{\xi}}. \tag{4.3.33}$$

The shape derivative of the Jacobian is given by

$$\dot{J}(\tilde{\xi}) = \frac{\dot{J}_i(\tilde{\xi})J_i(\tilde{\xi})}{J(\tilde{\xi})}. \tag{4.3.34}$$

Now the sensitivity of the unit outward normal n_i on the boundary can be derived from that of the Jacobian as

$$\dot{n}_i = \frac{\dot{J}_i(\tilde{\xi})J(\tilde{\xi})-J_i(\tilde{\xi})\dot{J}(\tilde{\xi})}{J^2(\tilde{\xi})}. \tag{4.3.35}$$

The shape derivatives of hypersingular fundamental solutions are

$$\begin{aligned}\dot{D}_{kij}\left(\mathbf{s},\ \mathbf{x}\right) = & \frac{1}{4\pi(1-\nu)}\dot{\left(\frac{1}{r}\right)}[(1-2\nu)(r_{,i}\delta_{jk}+r_{,j}\delta_{ki}-r_{,k}\delta_{ij})+2r_{,i}r_{,j}r_{,k}] \\ & +\frac{1}{4\pi(1-\nu)r}[(1-2\nu)(\dot{(r_{,i})}\delta_{jk}+\dot{(r_{,j})}\delta_{ki}-\dot{(r_{,k})}\delta_{ij})]\end{aligned}$$

$$+\frac{1}{4\pi(1-\nu)r}[2((\dot{r}_{,i})r_{,j}r_{,k}+r_{,i}(\dot{r}_{,j})r_{,k}+r_{,i}r_{,j}(\dot{r}_{,k}))], \tag{4.3.36}$$

$$\dot{S}_{kij}\left(\mathbf{s},\ \mathbf{x}\right)=\dot{S}^{1}_{kij}\left(\mathbf{s},\ \mathbf{x}\right)+\dot{S}^{2}_{kij}\left(\mathbf{s},\ \mathbf{x}\right)+\dot{S}^{3}_{kij}\left(\mathbf{s},\ \mathbf{x}\right), \tag{4.3.37}$$

with

$$\begin{aligned}
\dot{S}^{1}_{kij}\left(\mathbf{s},\ \mathbf{x}\right)=&\frac{\mu}{2\pi(1-\nu)}\left(\dot{\frac{1}{r^2}}\right)\left\{2\frac{\partial r}{\partial n}[(1-2\nu)\delta_{ij}r_{,k}+\nu(r_{,j}\delta_{ik}+r_{,i}\delta_{jk})]\right\}\\
&+\frac{\mu}{2\pi(1-\nu)}\left(\dot{\frac{1}{r^2}}\right)\left\{2\frac{\partial r}{\partial n}(-4r_{,i}r_{,j}r_{,k})+2\nu(n_ir_{,j}r_{,k}+n_jr_{,i}r_{,k})\right\}\\
&+\frac{\mu}{2\pi(1-\nu)}\left(\dot{\frac{1}{r^2}}\right)\{(1-2\nu)(2n_kr_{,i}r_{,j}+n_j\delta_{ik}+n_i\delta_{jk})\},\\
&+\frac{\mu}{2\pi(1-\nu)}\left(\dot{\frac{1}{r^2}}\right)\{-(1-4\nu)n_k\delta_{ij}\},
\end{aligned} \tag{4.3.38}$$

$$\begin{aligned}
\dot{S}^{2}_{kij}\left(\mathbf{s},\ \mathbf{x}\right)=&\frac{\mu}{2\pi(1-\nu)r^2}\left\{2\left(\dot{\frac{\partial r}{\partial n}}\right)[(1-2\nu)\delta_{ij}r_{,k}+\nu(r_{,j}\delta_{ik}+r_{,i}\delta_{jk})]\right\} \qquad (4.3.39)\\
&+\frac{\mu}{2\pi(1-\nu)r^2}\left\{2\left(\dot{\frac{\partial r}{\partial n}}\right)(-4r_{,i}r_{,j}r_{,k})\right\}\\
&+\frac{\mu}{2\pi(1-\nu)r^2}\left\{2\left(\frac{\partial r}{\partial n}\right)[(1-2\nu)\delta_{ij}(\dot{r}_{,k})+\nu((\dot{r}_{,j})\delta_{ik}+(\dot{r}_{,i})\delta_{jk})]\right\}\\
&+\frac{\mu}{2\pi(1-\nu)r^2}\left\{2\left(\frac{\partial r}{\partial n}\right)[-4((\dot{r}_{,i})r_{,j}r_{,k}+r_{,i}(\dot{r}_{,j})r_{,k}+r_{,i}r_{,j}(\dot{r}_{,k}))]\right\},
\end{aligned}$$

$$\begin{aligned}
\dot{S}^{3}_{kij}\left(\mathbf{s},\ \mathbf{x}\right)=&\frac{\mu}{2\pi(1-\nu)r^2}\{2\nu(\dot{n}_ir_{,j}r_{,k}+n_i(\dot{r}_{,j})r_{,k}+n_ir_{,j}(\dot{r}_{,k}))\}\\
&+\frac{\mu}{2\pi(1-\nu)r^2}\{2\nu(\dot{n}_jr_{,i}r_{,k}+n_j(\dot{r}_{,j})r_{,k}+n_jr_{,i}(\dot{r}_{,k}))\}\\
&+\frac{\mu}{2\pi(1-\nu)r^2}\{(1-2\nu)[2(\dot{n}_kr_{,i}r_{,j}+n_k(\dot{r}_{,i})r_{,j}+n_kr_{,i}(\dot{r}_{,j}))]\}\\
&+\frac{\mu}{2\pi(1-\nu)r^2}\{(1-2\nu)(\dot{n}_j\delta_{ik}+\dot{n}_i\delta_{jk})-(1-4\nu)\dot{n}_k\delta_{ij}\}.
\end{aligned} \tag{4.3.40}$$

4.3.2 Three-dimensional problem

The shape derivatives of displacement and traction fundamental solutions in three dimensions are given by

$$\dot{U}_{ij}\left(\mathbf{s},\ \mathbf{x}\right)=\frac{1}{16\pi\mu(1-\nu)}\left\{\left(\dot{\frac{1}{r}}\right)[(3-4\nu)\delta_{ij}+r_{,i}r_{,j}]\right\} \tag{4.3.41}$$

$$+\frac{1}{16\pi\mu(1-\nu)}\left\{\frac{1}{r}[(\dot{r_{,i}})r_{,j}+r_{,i}(\dot{r_{,j}})]\right\},$$

$$\begin{aligned}\dot{T}_{ij}\left(\mathbf{s},\ \mathbf{x}\right) = & \frac{-1}{8\pi(1-\nu)}\left(\dot{\frac{1}{r^2}}\right)\left\{\frac{\partial r}{\partial n}[(1-2\nu)\delta_{ij}+3r_{,i}r_{,j}]\right\}\\ & +\frac{-1}{8\pi(1-\nu)}\left(\dot{\frac{1}{r^2}}\right)\{(1-2\nu)(n_i r_{,j}-n_j r_{,i})\}\\ & +\frac{-1}{8\pi(1-2\nu)r^2}\left\{\left(\dot{\frac{\partial r}{\partial n}}\right)[(1-2\nu)\delta_{ij}+3r_{,i}r_{,j}]\right\}\\ & +\frac{-1}{8\pi(1-2\nu)r^2}\left\{3\frac{\partial r}{\partial n}[(\dot{r_{,i}})r_{,j}+r_{,i}(\dot{r_{,j}})]\right\},\\ & +\frac{-1}{8\pi(1-2\nu)r^2}\{(1-2\nu)[\dot{n}_i r_{,j}+n_i(\dot{r_{,j}})-\dot{n}_j r_{,i}-n_j(\dot{r_{,i}})]\}\end{aligned} \tag{4.3.42}$$

where

$$\left(\dot{\frac{1}{r}}\right)=-\frac{\dot{r}}{r^2},\quad \left(\dot{\frac{1}{r^2}}\right)=-\frac{2\dot{r}}{r^3}, \tag{4.3.43}$$

$$(\dot{r_{,i}})=\left(\dot{\frac{x_i-s_i}{r}}\right)=\frac{(\dot{x}_i-\dot{s}_i)r-(x_i-s_i)\dot{r}}{r^2}, \tag{4.3.44}$$

$$\left(\dot{\frac{\partial r}{\partial n}}\right)=(\dot{r_{,i}n_i})=(\dot{r_{,i}})n_i+r_{,i}\dot{n}_i, \tag{4.3.45}$$

$$\dot{r}=\left[\dot{\sqrt{(x_i-s_i)(x_i-s_i)}}\right]=\frac{(x_i\dot{-}s_i)(x_i-s_i)}{r}=\frac{(\dot{x}_i-\dot{s}_i)(x_i-s_i)}{r}. \tag{4.3.46}$$

The sensitivity of unit outward normal n_i on the boundary is

$$\dot{n}_i=\frac{\dot{J}_i(\tilde{\boldsymbol{\xi}})J(\tilde{\boldsymbol{\xi}})-J_i(\tilde{\boldsymbol{\xi}})\dot{J}(\tilde{\boldsymbol{\xi}})}{J^2(\tilde{\boldsymbol{\xi}})}, \tag{4.3.47}$$

where J is Jacobian determinant. Its expression and shape derivatives are given by

$$J_i(\tilde{\boldsymbol{\xi}})=\varepsilon_{ijk}\frac{\partial x_j}{\partial\tilde{\xi}_1}\frac{\partial x_k}{\partial\tilde{\xi}_2}, \tag{4.3.48}$$

$$J(\tilde{\boldsymbol{\xi}})=\sqrt{J_i(\tilde{\boldsymbol{\xi}})J_i(\tilde{\boldsymbol{\xi}})}, \tag{4.3.49}$$

$$\dot{J}_i(\tilde{\boldsymbol{\xi}})=\varepsilon_{ijk}\left(\dot{\frac{\partial x_j}{\partial\tilde{\xi}_1}}\right)\frac{\partial x_k}{\partial\tilde{\xi}_2}+\varepsilon_{ijk}\frac{\partial x_j}{\partial\tilde{\xi}_1}\left(\dot{\frac{\partial x_k}{\partial\tilde{\xi}_2}}\right), \tag{4.3.50}$$

$$\dot{J}(\tilde{\boldsymbol{\xi}})=\frac{\dot{J}_i(\tilde{\boldsymbol{\xi}})J_i(\tilde{\boldsymbol{\xi}})}{J(\tilde{\boldsymbol{\xi}})}, \tag{4.3.51}$$

with ε_{ijk} the permutation operator

$$\varepsilon_{ijk} = \begin{cases} 1 & \text{for cyclic suffix order : 123, 231, 312,} \\ -1 & \text{for cyclic suffix order : 132, 213, 321,} \\ 0 & \text{if any two indices are the same.} \end{cases} \tag{4.3.52}$$

The sensitivities of hypersingular fundamental solutions are

$$\begin{aligned} \dot{D}_{kij} = & \frac{1}{8\pi(1-\nu)} \left(\dot{\frac{1}{r^2}}\right) [(1-2\nu)(\delta_{ik} r_{,j} + \delta_{jk} r_{,i} - \delta_{ij} r_{,k}) + 3 r_{,i} r_{,j} r_{,k}] \\ & + \frac{1}{8\pi(1-\nu)r^2} [(1-2\nu)(\delta_{ik}(\dot{r}_{,j}) + \delta_{jk}(\dot{r}_{,i}) - \delta_{ij}(\dot{r}_{,k}))] \\ & + \frac{1}{8\pi(1-\nu)r^2} [3((\dot{r}_{,i}) r_{,j} r_{,k} + r_{,i}(\dot{r}_{,j}) r_{,k} + r_{,i} r_{,j}(\dot{r}_{,k}))], \end{aligned} \tag{4.3.53}$$

$$\dot{S}_{kij} = \dot{S}^1_{kij}(\mathbf{s},\, \mathbf{x}) + \dot{S}^2_{kij}(\mathbf{s},\, \mathbf{x}) + \dot{S}^3_{kij}(\mathbf{s},\, \mathbf{x}) + \dot{S}^4_{kij}(\mathbf{s},\, \mathbf{x}), \tag{4.3.54}$$

with

$$\begin{aligned} \dot{S}^1_{kij} = & \frac{\mu}{4\pi(1-\nu)} \left(\dot{\frac{1}{r^3}}\right) \left\{ 3\frac{\partial r}{\partial n} [(1-2\nu)\delta_{ij} r_{,k} + \nu(r_{,j}\delta_{ik} + r_{,i}\delta_{jk}) - 5 r_{,i} r_{,j} r_{,k}] \right\} \\ & + \frac{\mu}{4\pi(1-\nu)} \left(\dot{\frac{1}{r^3}}\right) \{3\nu(n_i r_{,j} r_{,k} + n_j r_{,i} r_{,k})\} \\ & + \frac{\mu}{4\pi(1-\nu)} \left(\dot{\frac{1}{r^3}}\right) \{(1-2\nu)(3 n_k r_{,i} r_{,j} + n_j \delta_{ik} + n_i \delta_{jk}) - (1-4\nu) n_k \delta_{ij}\}, \end{aligned} \tag{4.3.55}$$

$$\begin{aligned} \dot{S}^2_{kij} = & \frac{\mu}{4\pi(1-\nu)r^3} \left\{ 3\left(\dot{\frac{\partial r}{\partial n}}\right) [(1-2\nu)\delta_{ij} r_{,k} + \nu(r_{,j}\delta_{ik} + r_{,i}\delta_{jk}) - 5 r_{,i} r_{,j} r_{,k}] \right\} \\ & + \frac{\mu}{4\pi(1-\nu)r^3} \left\{ 3\frac{\partial r}{\partial n} [(1-2\nu)\delta_{ij}(\dot{r}_{,k}) + \nu((\dot{r}_{,j})\delta_{ik} + (\dot{r}_{,i})\delta_{jk})] \right\} \\ & - \frac{\mu}{4\pi(1-\nu)r^3} \left\{ 3\frac{\partial r}{\partial n} [5((\dot{r}_{,i}) r_{,j} r_{,k} + r_{,i}(\dot{r}_{,j}) r_{,k} + r_{,i} r_{,j}(\dot{r}_{,k}))] \right\}, \end{aligned} \tag{4.3.56}$$

$$\begin{aligned} \dot{S}^3_{kij} = & \frac{\mu}{4\pi(1-\nu)r^3} \{3\nu(\dot{n}_i r_{,j} r_{,k} + n_i(\dot{r}_{,j}) r_{,k} + n_i r_{,j}(\dot{r}_{,k}))\} \\ & + \frac{\mu}{4\pi(1-\nu)r^3} \{3\nu(\dot{n}_j r_{,i} r_{,k} + n_j(\dot{r}_{,i}) r_{,k} + n_j r_{,i}(\dot{r}_{,k}))\}, \end{aligned} \tag{4.3.57}$$

$$\begin{aligned} \dot{S}^4_{kij} = & \frac{\mu}{4\pi(1-\nu)r^3} \{(1-2\nu)(3\dot{n}_k r_{,i} r_{,j} + 3 n_k(\dot{r}_{,i}) r_{,j} + 3 n_k r_{,i}(\dot{r}_{,j}))\} \\ & + \frac{\mu}{4\pi(1-\nu)r^3} \{(1-2\nu)(\dot{n}_j \delta_{ik} + \dot{n}_i \delta_{jk}) - (1-4\nu)\dot{n}_k \delta_{ij}\}. \end{aligned} \tag{4.3.58}$$

Tab. 4.1 shows the singularity order of the fundamental solution sensitivities, where we can see that they have the same order as the fundamental solutions.

Kernel sensitivity	Order	Singularity type	Dimension
$\dot{U}_{ij}$	$O(\ln(1/r))$	weakly singular	2D
$\dot{T}_{ij}$	$O(1/r)$	strongly singular	2D
$\dot{U}_{ij}$	$O(1/r)$	weakly singular	3D
$\dot{T}_{ij}$	$O(1/r^2)$	strongly singular	3D
$\dot{D}_{kij}$	$O(1/r)$	strongly singular	2D
$\dot{S}_{kij}$	$O(1/r^2)$	hypersingular	2D
$\dot{D}_{kij}$	$O(1/r^2)$	strongly singular	3D
$\dot{S}_{kij}$	$O(1/r^3)$	hypersingular	3D

Table 4.1: The singularity of kernel function sensitivities

4.4 Postprocessing

Similar to structural analysis, the displacement and stress shape derivatives are also treated differently for the surface and domain points. After solving Eq. (4.2.21), we have

$$\dot{u}_j^e(\tilde{\boldsymbol{\xi}}) = \sum_{a=1}^{n_a} R_{ea}(\tilde{\boldsymbol{\xi}}) \dot{u}_j{}^{ea}, \tag{4.4.59}$$

$$\frac{\mathrm{d}\dot{u}_j^e(\tilde{\boldsymbol{\xi}})}{\mathrm{d}\tilde{\boldsymbol{\xi}}} = \sum_{a=1}^{n_a} \frac{\mathrm{d}R_{ea}(\tilde{\boldsymbol{\xi}})}{\mathrm{d}\tilde{\boldsymbol{\xi}}} \dot{u}_j{}^{ea}, \tag{4.4.60}$$

$$\dot{t}_j^e(\tilde{\boldsymbol{\xi}}) = \sum_{a=1}^{n_a} R_{ea}(\tilde{\boldsymbol{\xi}}) \dot{t}_j^{ea}. \tag{4.4.61}$$

The above quantities will be used in the sensitivity evaluation.

4.4.1 Evaluate shape sensitivity at interior points

The shape sensitivity at the interior points can be obtained by taking shape derivatives with respect to design variables for Somigliana's identities. By ignoring domain integrals, the displacement shape sensitivity is

$$\begin{aligned}\dot{u}_i\left(\mathbf{S}\right) = &\int_S \dot{U}_{ij}\left(\mathbf{S},\mathbf{x}\right)t_j\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right) + \int_\Omega U_{ij}\left(\mathbf{S},\mathbf{x}\right)\dot{t}_j\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right)\\ &+ \int_\Omega U_{ij}\left(\mathbf{S},\mathbf{x}\right)t_j\left(\mathbf{x}\right)[\mathrm{d}S\dot{}(\mathbf{x})] - \int_S \dot{T}_{ij}\left(\mathbf{S},\mathbf{x}\right)u_j\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right)\\ &- \int_S T_{ij}\left(\mathbf{S},\mathbf{x}\right)\dot{u}_j\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right) - \int_S T_{ij}\left(\mathbf{S},\mathbf{x}\right)u_j\left(\mathbf{x}\right)[\mathrm{d}S\dot{}(\mathbf{x})],\end{aligned}\tag{4.4.62}$$

and stress shape sensitivity is

$$\begin{aligned}\dot{\sigma}_{ij}\left(\mathbf{S}\right) = &\int_S \dot{D}_{kij}\left(\mathbf{S},\mathbf{x}\right)t_k\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right) + \int_S D_{kij}\left(\mathbf{S},\mathbf{x}\right)\dot{t}_k\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right)\\ &+ \int_S D_{kij}\left(\mathbf{S},\mathbf{x}\right)t_k\left(\mathbf{x}\right)[\mathrm{d}S\dot{}(\mathbf{x})] - \int_S \dot{S}_{kij}\left(\mathbf{S},\mathbf{x}\right)u_k\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right)\\ &- \int_S S_{kij}\left(\mathbf{S},\mathbf{x}\right)\dot{u}_k\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right) - \int_S S_{kij}\left(\mathbf{S},\mathbf{x}\right)u_k\left(\mathbf{x}\right)[\mathrm{d}S\dot{}(\mathbf{x})].\end{aligned}\tag{4.4.63}$$

4.4.2 Evaluate stress shape sensitivity at boundary points

Two-dimension problem

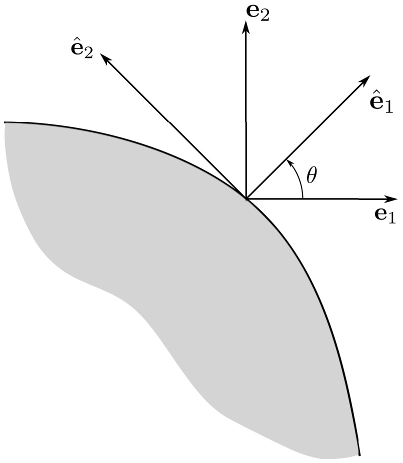


Figure 4.2: Local coordinate system for sensitivity analysis in 2D

Defining a local coordinate system such that $\hat{e}_{1j}(j = 1, 2)$ are the unit vector in the normal directions and $\hat{e}_{2j}$ are the unit vector in the tangential directions to the boundary element so that

$$\hat{\mathbf{x}} = \hat{x}_1\hat{\mathbf{e}}_1 + \hat{x}_2\hat{\mathbf{e}}_2. \tag{4.4.64}$$

The tangential vector $\mathbf{m}$ is expressed with

$$\mathbf{m} = \frac{\partial \mathbf{x}(\tilde{\xi})}{\partial \tilde{\xi}}. \tag{4.4.65}$$

And the shape derivatives of $|\mathbf{m}|$ are

$$|\dot{\mathbf{m}}| = \overset{\cdot}{\left(\sqrt{\frac{\partial x_i}{\partial \tilde{\xi}}\frac{\partial x_i}{\partial \tilde{\xi}}}\right)} = \frac{\overset{\cdot}{\left(\frac{\partial x_i}{\partial \tilde{\xi}}\right)} \frac{\partial x_i}{\partial \tilde{\xi}}}{|\mathbf{m}|}, \tag{4.4.66}$$

with

$$\overset{\cdot}{\left(\frac{\partial \mathbf{x}}{\partial \tilde{\xi}}\right)} = \frac{\partial N_a(\tilde{\xi})}{\partial \tilde{\xi}} \dot{\mathbf{x}}^a. \tag{4.4.67}$$

The normalized local tangential vector

$$\dot{\hat{\mathbf{e}}}_1 = \dot{\mathbf{n}}, \tag{4.4.68}$$

$$\dot{\hat{\mathbf{e}}}_2 = \overset{\cdot}{\left(\frac{\mathbf{m}}{|\mathbf{m}|}\right)}. \tag{4.4.69}$$

Defining the displacements, strains, stresses and tractions in the local coordinates $\hat{x}_j$ as $\hat{u}_j$, $\hat{\epsilon}_{ij}$, $\hat{\sigma}_{ij}$ and $\hat{t}_j$ respectively, the corresponding stress components $\hat{\sigma}_{ij}$ can be written as

$$\begin{aligned} \dot{\hat{\epsilon}}(\tilde{\xi}) = \dot{\hat{u}}_{2,2}(\tilde{\xi}) = \overset{\cdot}{\left(\frac{\partial \hat{u}_2}{\partial \tilde{\xi}} \frac{\partial \tilde{\xi}}{\partial \hat{x}_2}\right)} &= \overset{\cdot}{\left(A_{2i} \frac{\partial u_i}{\partial \tilde{\xi}} \frac{\partial \tilde{\xi}}{\partial \hat{x}_2}\right)} \\ &= (\dot{A}_{2i}) \frac{\partial u_i}{\partial \tilde{\xi}} \frac{\partial \tilde{\xi}}{\partial \hat{x}_2} + A_{2i} \overset{\cdot}{\left(\frac{\partial u_i}{\partial \tilde{\xi}}\right)} \frac{\partial \tilde{\xi}}{\partial \hat{x}_2} + A_{2i} \frac{\partial u_i}{\partial \tilde{\xi}} \overset{\cdot}{\left(\frac{\partial \tilde{\xi}}{\partial \hat{x}_2}\right)}, \end{aligned} \tag{4.4.70}$$

with

$$\overset{\cdot}{\left(\frac{\partial \tilde{\xi}}{\partial \hat{x}_2}\right)} = \overset{\cdot}{\left(\frac{1}{|\mathbf{m}_2|}\right)}. \tag{4.4.71}$$

The sensitivity of the transformation matrix from the global coordinate system to the local tangential system is

$$\dot{A} = \begin{bmatrix} \dot{\hat{\mathbf{e}}}_1 \\ \dot{\hat{\mathbf{e}}}_2 \end{bmatrix} = \begin{bmatrix} \dot{n}_1 & \dot{n}_2 \\ -\dot{n}_2 & \dot{n}_1 \end{bmatrix}. \tag{4.4.72}$$

The stress sensitivity tensor in the local coordinate system is

$$\dot{\hat{\sigma}}_{11} = \dot{\hat{t}}_1, \tag{4.4.73}$$

$$\dot{\hat{\sigma}}_{12} = \dot{\hat{t}}_2, \tag{4.4.74}$$

$$\dot{\hat{\sigma}}_{22} = \left(\frac{E}{1-\nu^2}\right)\dot{\hat{\epsilon}}_{22} + \left(\frac{\nu}{1-\nu}\right)\dot{\hat{t}}_1. \tag{4.4.75}$$

Transferring the stress sensitivity back to the global Cartesian coordinate system writes

$$\begin{aligned}\dot{\sigma}_{ij} &= \dot{\left(A_{ki}A_{nj}\hat{\sigma}_{kn}\right)} \\ &= \dot{A}_{ki}A_{nj}\hat{\sigma}_{kn} + A_{ki}\dot{A}_{nj}\hat{\sigma}_{kn} + A_{ki}A_{nj}\dot{\hat{\sigma}}_{kn}.\end{aligned} \tag{4.4.76}$$

Three-dimension problems

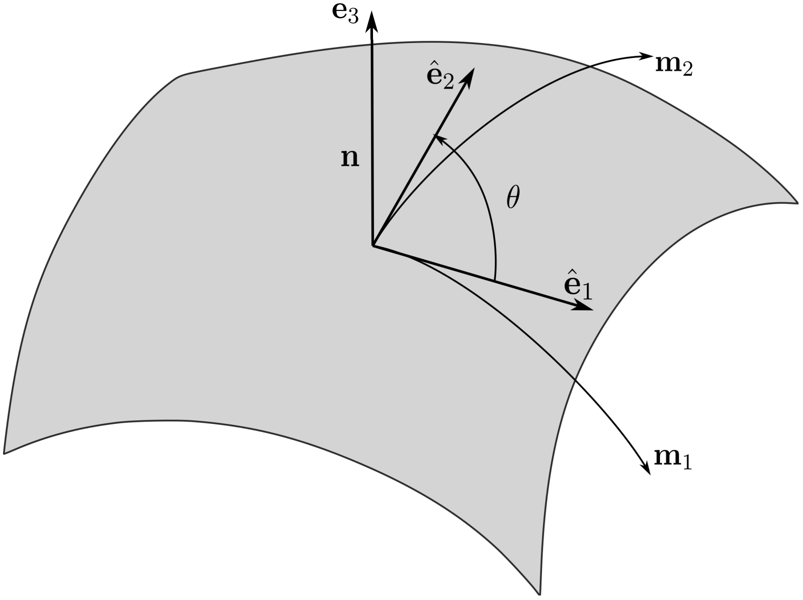


Figure 4.3: Local coordinate system for sensitivity analysis in 3D

By constructing a local system as in structural analysis (Fig. 4.3), the boundary strain sensitivity can be evaluated

$$\begin{aligned}\dot{\hat{\epsilon}}_{ij} &= \dot{\left(\tfrac{\partial \hat{u}_i}{\partial \hat{x}_j}\right)} = \dot{\left(\tfrac{\partial \hat{u}_i}{\partial \tilde{\xi}_k}\tfrac{\partial \tilde{\xi}_k}{\partial \hat{x}_j}\right)} = \dot{\left(A_{il}\tfrac{\partial u_l}{\partial \tilde{\xi}_k}\tfrac{\partial \tilde{\xi}_k}{\partial \hat{x}_j}\right)} \\ &= \dot{A}_{il}\frac{\partial u_l}{\partial \tilde{\xi}_k}\frac{\partial \xi_k}{\partial \hat{x}_j} + A_{il}\dot{\left(\tfrac{\partial u_l}{\partial \tilde{\xi}_k}\right)}\frac{\partial \xi_k}{\partial \hat{x}_j} + A_{il}\frac{\partial u_l}{\partial \tilde{\xi}_k}\dot{\left(\tfrac{\partial \tilde{\xi}_k}{\partial \hat{x}_j}\right)},\end{aligned} \tag{4.4.77}$$

where the shape sensitivities of the derivatives of $\tilde{\xi}_j$ with respect to $\hat{x}_j$ are

$$\dot{\left(\tfrac{\partial \tilde{\xi}_1}{\partial \hat{x}_1}\right)} = \dot{\left(\tfrac{1}{|\mathbf{m}_1|}\right)}, \tag{4.4.78}$$

$$\overset{\cdot}{\left(\frac{\partial\tilde{\xi}_1}{\partial\hat{x}_2}\right)} = \overset{\cdot}{\left(\frac{-\cos\theta}{|\mathbf{m}_1|\sin\theta}\right)} = \overset{\cdot}{\left(\frac{1}{|\mathbf{m}_1|}\right)}\frac{-\cos\theta}{\sin\theta} + \frac{1}{|\mathbf{m}_1|}\overset{\cdot}{\left(\frac{-\cos\theta}{\sin\theta}\right)}, \tag{4.4.79}$$

$$\overset{\cdot}{\left(\frac{\partial\tilde{\xi}_2}{\partial\hat{x}_1}\right)} = 0, \tag{4.4.80}$$

$$\overset{\cdot}{\left(\frac{\partial\tilde{\xi}_2}{\partial\hat{x}_2}\right)} = \overset{\cdot}{\left(\frac{1}{|\mathbf{m}_2|}\sin\theta\right)} = \overset{\cdot}{\left(\frac{1}{|\mathbf{m}_2|}\right)}\sin\theta + \frac{1}{|\mathbf{m}_2|}(\overset{\cdot}{\sin}\,\theta). \tag{4.4.81}$$

The shape sensitivities related to the angle θ are

$$(\overset{\cdot}{\cos}\,\theta) = \overset{\cdot}{\left(\frac{\mathbf{m}_1\cdot\mathbf{m}_2}{|\mathbf{m}_1|\,|\mathbf{m}_2|}\right)} = \frac{(\overset{\cdot}{\mathbf{m}_1\cdot\mathbf{m}_2})\,(|\mathbf{m}_1|\,|\mathbf{m}_2|) - (\mathbf{m}_1\cdot\mathbf{m}_2)\,(\overset{\cdot}{|\mathbf{m}_1|\,|\mathbf{m}_2|})}{(|\mathbf{m}_1|\,|\mathbf{m}_2|)^2}, \tag{4.4.82}$$

$$\dot{\theta} = \overset{\cdot}{\left[\arccos\left(\frac{\mathbf{m}_1\cdot\mathbf{m}_2}{|\mathbf{m}_1|\,|\mathbf{m}_2|}\right)\right]} = \frac{-1}{1-\left(\frac{\mathbf{m}_1\cdot\mathbf{m}_2}{|\mathbf{m}_1||\mathbf{m}_2|}\right)^2}\overset{\cdot}{\left(\frac{\mathbf{m}_1\cdot\mathbf{m}_2}{|\mathbf{m}_1|\,|\mathbf{m}_2|}\right)}, \tag{4.4.83}$$

$$(\overset{\cdot}{\sin}\,\theta) = \dot{\theta}\cos\theta, \tag{4.4.84}$$

$$\overset{\cdot}{\left(\frac{\cos\theta}{\sin\theta}\right)} = \frac{(\overset{\cdot}{\cos}\,\theta)\sin\theta - \cos\theta(\overset{\cdot}{\sin}\,\theta)}{\sin^2\theta}. \tag{4.4.85}$$

with

$$\overset{\cdot}{\left(\frac{1}{|\mathbf{m}_1|}\right)} = -\frac{|\dot{\mathbf{m}}_1|}{|\mathbf{m}_1|^2}, \tag{4.4.86}$$

$$\overset{\cdot}{\left(\frac{1}{|\mathbf{m}_2|}\right)} = -\frac{|\dot{\mathbf{m}}_2|}{|\mathbf{m}_2|^2}. \tag{4.4.87}$$

The shape sensitivities of the base vectors of the tangential system are

$$|\dot{\mathbf{m}}_1| = \overset{\cdot}{\left(\sqrt{\frac{\partial x_i}{\partial\tilde{\xi}_1}\frac{\partial x_i}{\partial\tilde{\xi}_1}}\right)} = \frac{\overset{\cdot}{\left(\frac{\partial x_i}{\partial\tilde{\xi}_1}\right)}\frac{\partial x_i}{\partial\tilde{\xi}_1}}{|\mathbf{m}_1|}, \tag{4.4.88}$$

$$|\dot{\mathbf{m}}_2| = \overset{\cdot}{\left(\sqrt{\frac{\partial x_i}{\partial\tilde{\xi}_2}\frac{\partial x_i}{\partial\tilde{\xi}_2}}\right)} = \frac{\overset{\cdot}{\left(\frac{\partial x_i}{\partial\tilde{\xi}_2}\right)}\frac{\partial x_i}{\partial\tilde{\xi}_2}}{|\mathbf{m}_2|}. \tag{4.4.89}$$

According to the Hooke's law, we can get

$$\dot{\hat{\sigma}}_{11} = \frac{E}{1-\nu^2}(\dot{\hat{\epsilon}}_{11} + \nu\dot{\hat{\epsilon}}_{22}) + \frac{\nu}{1-\nu}\dot{\hat{t}}_3, \tag{4.4.90}$$

$$\dot{\hat{\sigma}}_{12} = \frac{E}{1+\nu}\dot{\hat{\epsilon}}_{12}, \tag{4.4.91}$$

$$\dot{\hat{\sigma}}_{22} = \frac{E}{1-\nu^2}(\dot{\hat{\epsilon}}_{22} + \nu\dot{\hat{\epsilon}}_{11}) + \frac{\nu}{1-\nu}\dot{\hat{t}}_3, \tag{4.4.92}$$

$$\dot{\hat{\sigma}}_{33} = \dot{\hat{t}}_3, \tag{4.4.93}$$

$$\dot{\hat{\sigma}}_{23} = \dot{\hat{t}}_2, \tag{4.4.94}$$

$$\dot{\hat{\sigma}}_{13} = \dot{\hat{t}}_1. \tag{4.4.95}$$

The shape sensitivity of the boundary stress is finally transferred to the global Cartesian coordinate system as

$$\sigma_{ij} = \dot{A}_{ki} A_{nj} \hat{\sigma}_{kn} + A_{ki} \dot{A}_{nj} \hat{\sigma}_{kn} + A_{ki} A_{nj} \dot{\hat{\sigma}}_{kn}. \tag{4.4.96}$$

4.5 Shape sensitivity analysis numerical examples

In this section we will investigate the performance of IGABEM for shape sensitivity analysis through some numerical examples with closed-form solutions. All the geometries in two-dimensions are modelled using NURBS and that in three-dimensions using T-splines. To study the accuracy of numerical results $(\cdot)_h$ against analytical results $(\cdot)$, we define the following errors:

$$e_{L_2}(\cdot)_h = \frac{\|(\cdot)_h - (\cdot)\|_{L_2}}{\|(\cdot)\|_{L_2}}, \tag{4.5.97}$$

and

$$e_{L_\infty}(\cdot)_h = \frac{\|(\cdot)_h - (\cdot)\|_{L_2}}{\|(\cdot)\|_{L_\infty}}, \tag{4.5.98}$$

with

$$\|(\cdot)\|_{L_2} = \sqrt{\int_S (\cdot) \cdot (\cdot) \mathrm{d}S}, \tag{4.5.99}$$

and

$$\|(\cdot)\|_{L_\infty} = \max_{1 \leqslant i \leqslant n} |(\cdot)_i|. \tag{4.5.100}$$

4.5.1 Lamé problem

Consider a thick cylinder subject to uniform pressure $p = 10^5$ on the inner surface in the normal direction. The radius of the inner surface and outer surface is $a = 3$, and $b = 8$, respectively. The material parameters are Young's modulus $E = 10^5$, and Poisson's ration ν =0.3. The analytical displacement and stress in polar coordinates (r, θ) are given by

$$u_r(r, \theta) = \frac{pa^2}{E(b^2 - a^2)} \left[(1 - \nu) r + \frac{b^2(1 + \nu)}{r} \right], \tag{4.5.101}$$

$$\sigma_{rr}(r,\theta) = \frac{pa^2}{b^2-a^2}\left(1-\frac{b^2}{r^2}\right), \tag{4.5.102}$$

$$\sigma_{\theta\theta}(r,\theta) = \frac{pa^2}{b^2-a^2}\left(1+\frac{b^2}{r^2}\right). \tag{4.5.103}$$

Choosing the radius of the outer boundary b as the design variable, the analytical displacement and stress sensitivities are given by

$$\begin{aligned}\dot{u}_r(r,\theta) = &-\frac{2Pa^2b}{E(b^2-a^2)^2}\left[(1-\nu)r + \frac{b^2(1+\nu)}{r}\right] \\ &+\frac{Pa^2}{E(b^2-a^2)}\left[(1-\nu)\dot{r} + (1+\nu)\frac{2br-b^2\dot{r}}{r^2}\right],\end{aligned} \tag{4.5.104}$$

$$\dot{\sigma}_{rr}(r,\theta) = \frac{-2a^2bP}{(b^2-a^2)^2}\left(1-\frac{b^2}{r^2}\right) + \frac{Pa^2}{b^2-a^2}\left(\frac{2br^2-2b^2r\dot{r}}{r^4}\right), \tag{4.5.105}$$

$$\dot{\sigma}_{\theta\theta}(r,\theta) = \frac{-2a^2bP}{(b^2-a^2)^2}\left(1+\frac{b^2}{r^2}\right) + \frac{Pa^2}{b^2-a^2}\left(\frac{2br^2-2b^2r\dot{r}}{r^4}\right), \tag{4.5.106}$$

where the symbol $\dot{(\cdot)}$ refers to the shape derivatives of the superposed quantities.

Because of the symmetry, only a quarter of the cylinder needs to be modelled as shown in Fig. 4.4. The geometry is constructed using quadratic NURBS and the minimum number of elements and control points to represent the geometry are shown in Fig. 4.5. The shape sensitivity analysis is performed using a refined mesh with 8 elements on each segment. Figs. 4.6 and 4.7 show the IGABEM solutions of the displacement and stress sensitivities on the bottom edge AB, respectively. Figs. 4.8 and 4.9 show the corresponding errors of the numerical displacement and stress sensitivities. An excellent agreement between the analytical and numerical solutions is observed. To investigate the accuracy of shape sensitivities at the interior points, we select the points on the line of $a+0.5 \leqslant r \leqslant b-0.5$ and $\theta = \pi/4$. Supposing the domain points to be linearly varied in the radial direction, *i.e.*

$$\dot{r} = \frac{r-a}{b-a}, \tag{4.5.107}$$

the displacement and stress sensitivities can be evaluated using Eqs. (4.4.62) and (4.4.63). The numerical solutions are shown in Figs. 4.10 and 4.11, respectively.

And the errors are shown in Figs. 4.12 and 4.13. An excellent agreement with the analytical solution is seen again.

Fig. 4.14 illustrates the convergence of the errors $e_{L_2}(\mathbf{u}_h)$ and $e_{L_2}(\dot{\mathbf{u}}_h)$ against the number of degrees of freedom. Both the structural analysis and shape sensitivity analysis converge to the exact results. The reason for the large error in the shape sensitivity analysis compared with that of structural analysis is due to the fact that the numerical results from the structural analysis are used in the shape sensitivity analysis, which leads to error accumulation.

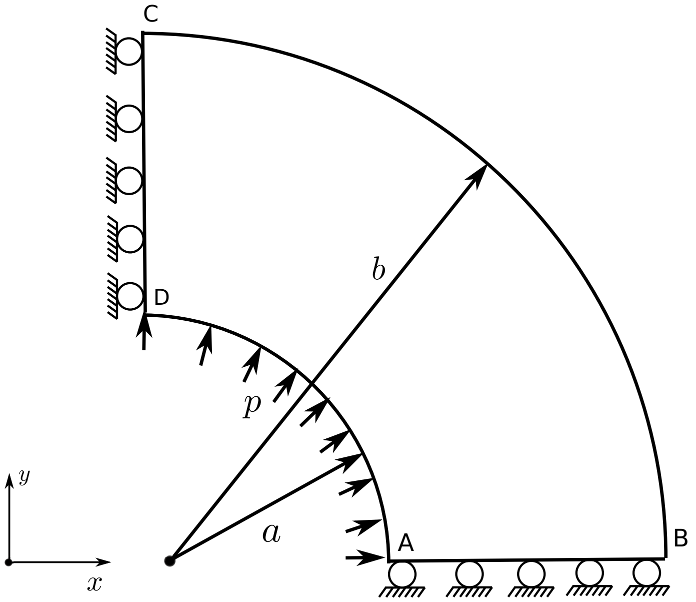


Figure 4.4: Definition of Lamé's problem

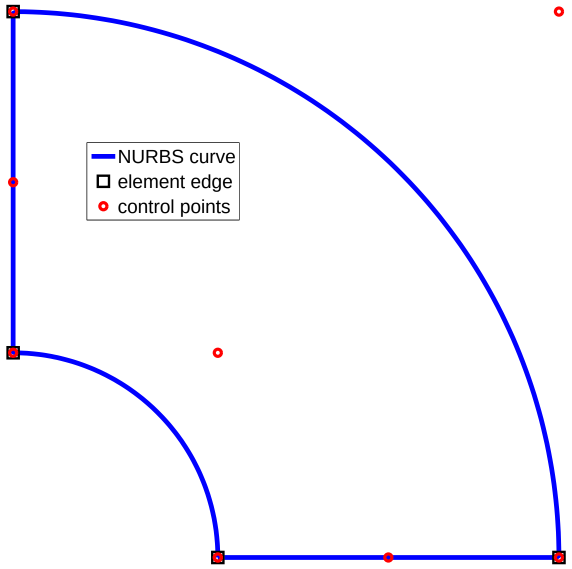


Figure 4.5: Geometric model of Lamé's problem

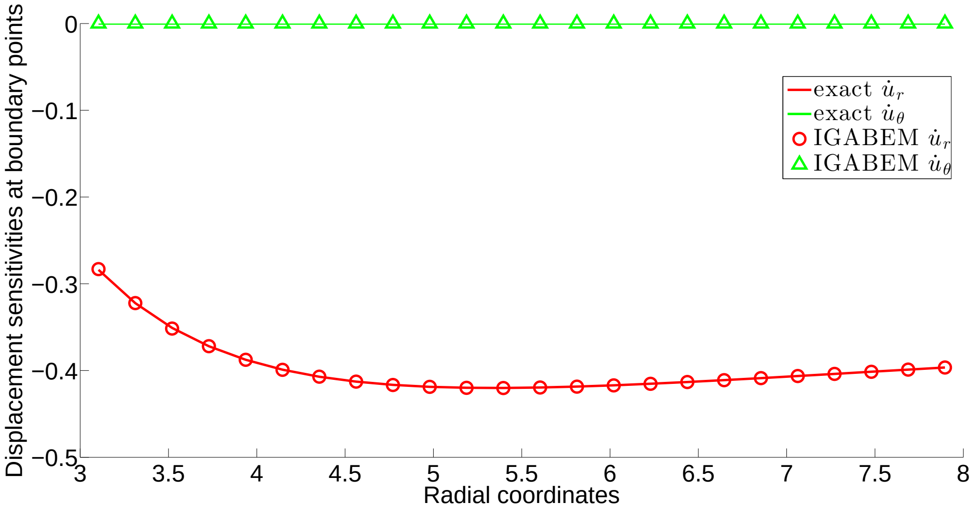


Figure 4.6: Displacement sensitivities on the boundary points for Lamé's problem

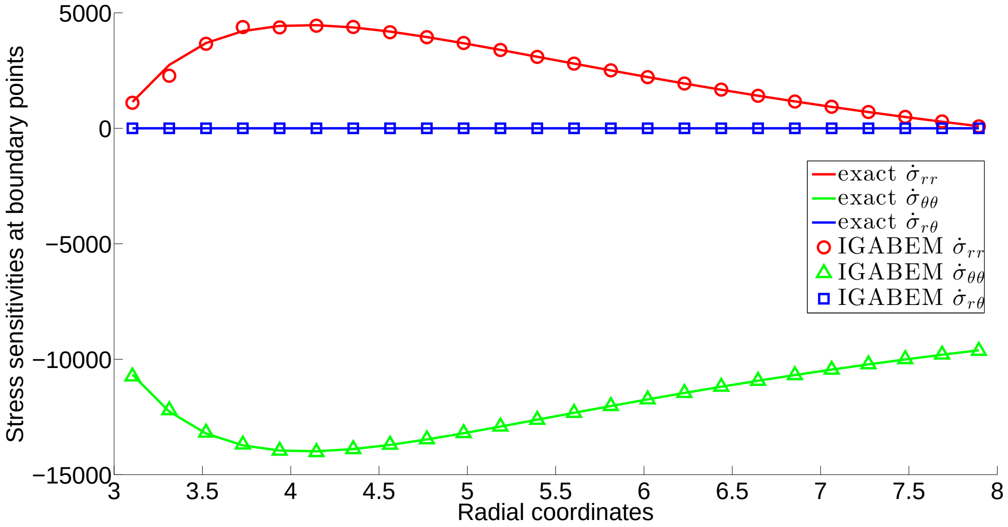


Figure 4.7: Stress sensitivities on the boundary points for Lamé's problem

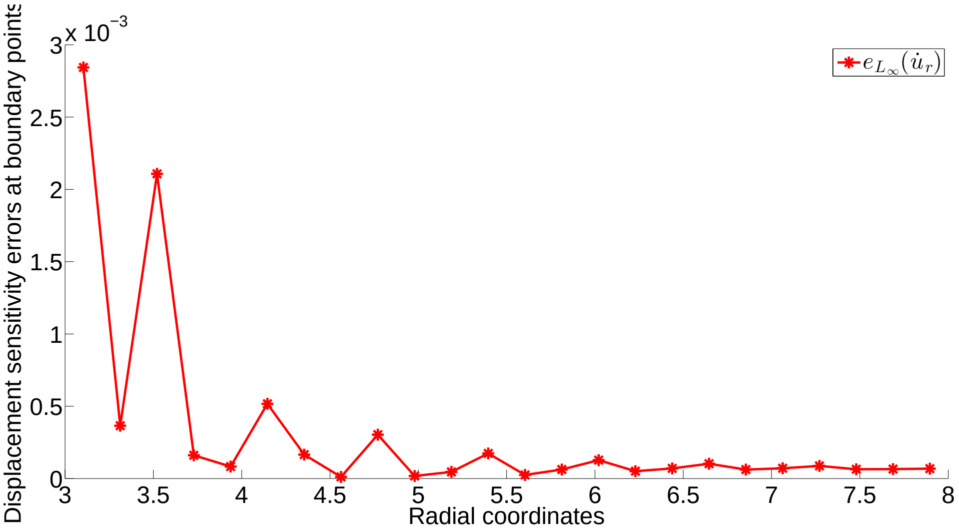


Figure 4.8: Displacement sensitivity errors on the boundary points for Lamé's problem

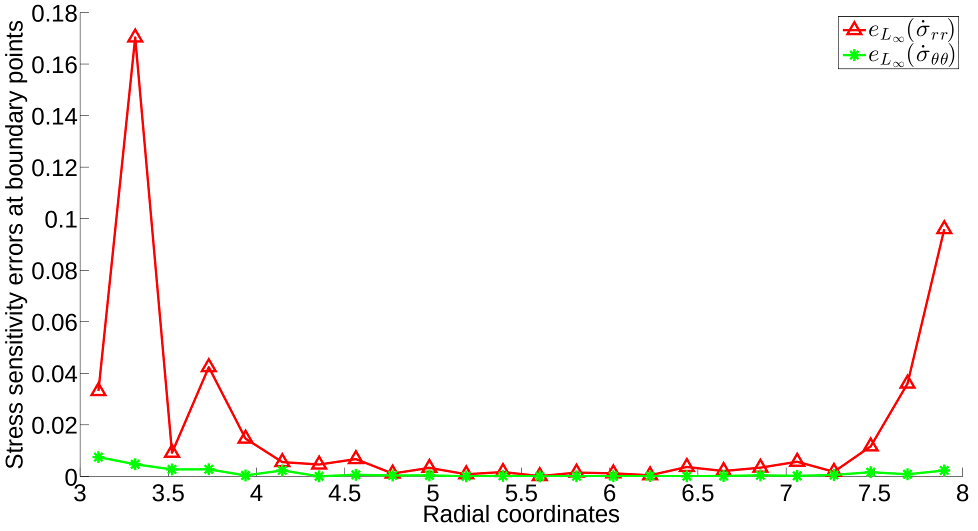


Figure 4.9: Stress sensitivity errors on the boundary points for Lamé's problem

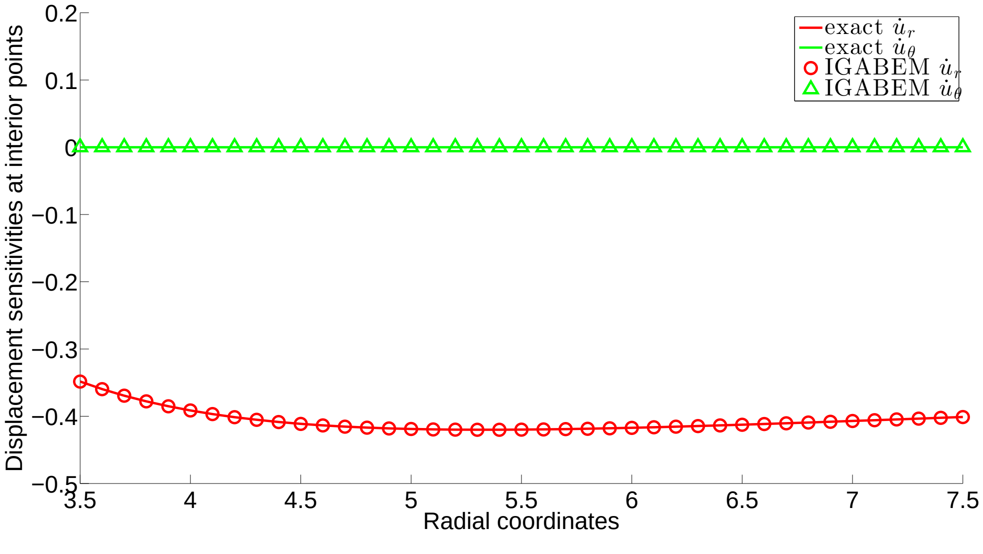


Figure 4.10: Displacement sensitivities at the interior points for Lamé's problem

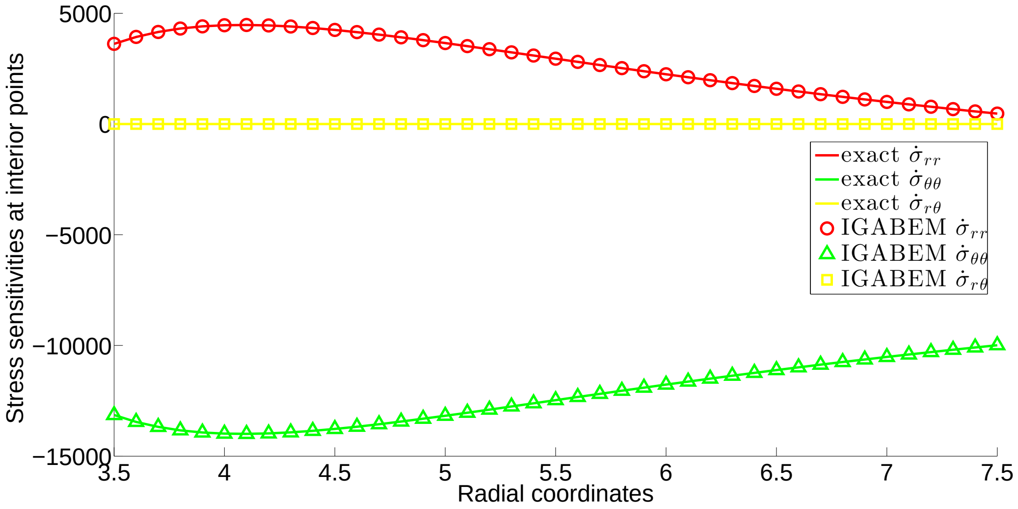


Figure 4.11: Stress sensitivities at the interior points for Lamé's problem

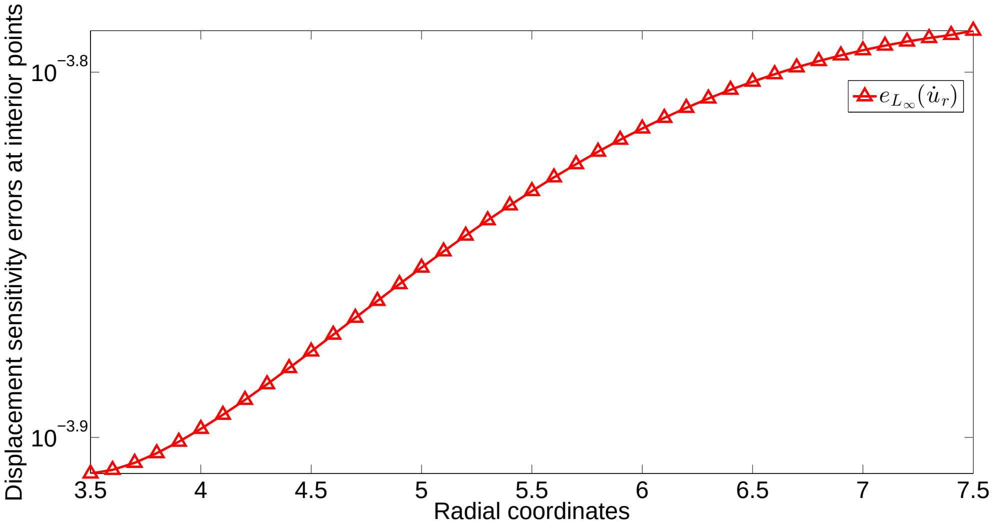


Figure 4.12: Displacement sensitivity error at the interior points for Lamé's problem

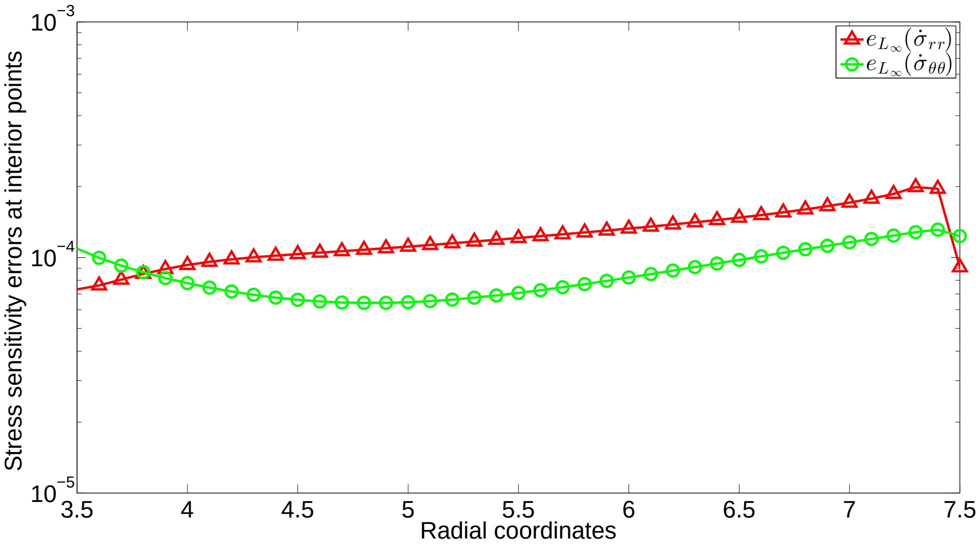


Figure 4.13: Stress sensitivity errors at the interior points for Lamé's problem

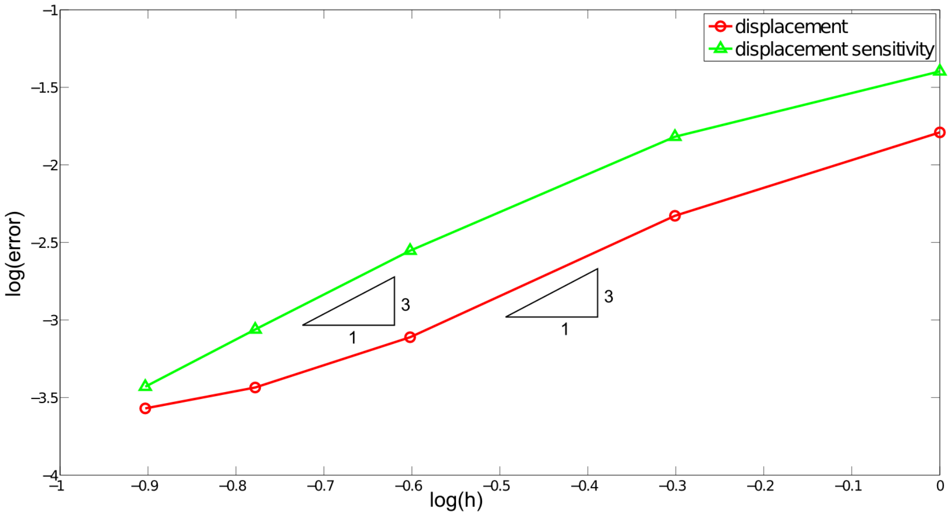


Figure 4.14: $e_{L_2}(\mathbf{u})$ and $e_{L_2}(\dot{\mathbf{u}})$ for Lamé's problem

4.5.2 Kirsch problem

The Kirsch problem is an infinitely large plate with a circular hole, subject to a far field uniform tension T. This problem can be modelled by extracting a finite domain and imposing the exact solution as boundary conditions around the boundary. Due to the symmetry, only a quarter of the plate is modelled, as shown in Fig. 4.15. The length of the plate is $b = 4$, and the radius of the hole is $a = 1$. The material parameters are $E = 10^5$, and $\nu = 0.3$. The traction boundary conditions on the top and left edge are from the analytical solutions. In the polar coordinates (r, θ), the analytical solutions for displacements and stresses are given by

$$u_r(r,\theta) = -\frac{Ta^2}{4Gr}\left\{(1+K) - (1-K)\left[4(1-\nu) - \frac{a^2}{r^2}\right]\cos\theta\right\}, \tag{4.5.108}$$

$$u_\theta(r,\theta) = -\frac{Ta^2}{4Gr}\left\{(1-K)\left[2(1-2\nu) + \frac{a^2}{r^2}\right]\sin 2\theta\right\}, \tag{4.5.109}$$

and

$$\sigma_{rr}(r,\theta) = \frac{T}{2}\left(1 - \frac{a^2}{r^2}\right) + \frac{T}{2}\left(1 - 4\frac{a^2}{r^2} + 3\frac{a^4}{r^4}\right)\cos 2\theta, \tag{4.5.110}$$

$$\sigma_{\theta\theta}(r,\theta) = \frac{T}{2}\left(1 + \frac{a^2}{r^2}\right) - \frac{T}{2}\left(1 + 3\frac{a^4}{r^4}\right)\cos 2\theta, \tag{4.5.111}$$

$$\sigma_{r\theta}(r,\theta) = -\frac{T}{2}\left(1 + 2\frac{a^2}{r^2} - 3\frac{a^4}{r^4}\right)\sin 2\theta. \tag{4.5.112}$$

with

$$K = 3 - 4\nu. \tag{4.5.113}$$

Assuming the design variable to be the hole radius a, the analytical displacement sensitivities are

$$\begin{aligned}\dot{u}_r(r,\theta) = &-\frac{Ta^2}{4G}\overset{\cdot}{\left(\frac{1}{r}\right)}\left\{(1+K) - (1-K)\left[4(1-\nu) - \frac{a^2}{r^2}\right]\cos\theta\right\} \\ &-\frac{Ta^2}{4Gr}\left\{-(1-K)\overset{\cdot}{\left(\frac{a^2}{r^2}\right)}\cos\theta\right\},\end{aligned} \tag{4.5.114}$$

$$\dot{u}_\theta(r,\theta) = -\frac{Ta^2}{4G}\overset{\cdot}{\left(\frac{1}{r}\right)}\left\{(1-K)\left[2(1-2\nu) + \frac{a^2}{r^2}\right]\sin 2\theta\right\} \tag{4.5.115}$$

$$-\frac{Ta^2}{4Gr}\left\{(1-K)\left[2(1-2\nu)+\frac{a^2}{r^2}\right]\sin 2\theta\right\},$$

with

$$\dot{\left(\frac{1}{r}\right)} = -\frac{\dot{r}}{r^2}, \tag{4.5.116}$$
$$\dot{\left(\frac{a^2}{r^2}\right)} = \frac{2ar^2-2a^2r\dot{r}}{r^4}.$$

The analytical stress sensitivities are

$$\begin{aligned}\dot{\sigma}_{rr}(r,\theta) = & -\frac{T}{2}\dot{\left(\frac{a^2}{r^2}\right)}+\frac{T}{2}\left[-4\dot{\left(\frac{a^2}{r^2}\right)}+3\dot{\left(\frac{a^4}{r^4}\right)}\right]\cos 2\theta \\ & +\frac{T}{2}\left(1-4\frac{a^2}{r^2}+3\frac{a^4}{r^4}\right)(\dot{\cos 2\theta}),\end{aligned} \tag{4.5.117}$$

$$\dot{\sigma}_{\theta\theta}(r,\theta) = \frac{T}{2}\dot{\left(\frac{a^2}{r^2}\right)}-\frac{3T}{2}\dot{\left(\frac{a^4}{r^4}\right)}\cos 2\theta-\frac{T}{2}\left(1+3\frac{a^4}{r^4}\right)(\dot{\cos 2\theta}), \tag{4.5.118}$$

$$\dot{\sigma}_{r\theta}(r,\theta) = -\frac{T}{2}\left[2\dot{\left(\frac{a^2}{r^2}\right)}-3\dot{\left(\frac{a^4}{r^4}\right)}\right]\sin 2\theta-\frac{T}{2}\left(1+2\frac{a^2}{r^2}-3\frac{a^4}{r^4}\right)(\dot{\sin 2\theta}), \tag{4.5.119}$$

with

$$\dot{\left(\frac{a^2}{r^2}\right)} = \frac{2ar^2-2a^2r\dot{r}}{r^4}, \tag{4.5.120}$$

$$\dot{\left(\frac{a^4}{r^4}\right)} = \frac{4a^3r^4-4a^4r^3\dot{r}}{r^8}. \tag{4.5.121}$$

Fig. 4.16 shows the NURBS geometry model with the minimum number of control points. The NURBS order is $p=2$ and the knot vector is $[0,0,0,1,1,2,2,3,3,4,4,5,5,5]$. The analysis model uses 12 elements per boundary segment.

Figs. 4.17 and 4.18 show the IGABEM solutions for the displacement and stress sensitivities on edge AB, respectively. The corresponding errors are shown in Figs. 4.19 and 4.20. Figs. 4.21 and 4.22 show the displacement and stress sensitivities at the interior points along the line $a+0.5 \leqslant r \leqslant \sqrt{2}L-0.5$ and $\theta = 3\pi/4$. And the corresponding errors are shown in Figs. 4.23 and 4.24. The domain points are assumed to be linearly spaced in the radial direction, *i.e.*

$$\dot{r} = \begin{cases} \frac{L/\cos\theta - r}{L/\cos\theta - a} & \text{for } \theta \geqslant \frac{3\pi}{4}, \\ \\ \frac{L/\sin\theta - r}{L/\sin\theta - a}, & \text{for } \theta < \frac{3\pi}{4}. \end{cases} \tag{4.5.122}$$

The numerical solutions agree with the analytical solutions very well. And the convergence of the structural and sensitivity analysis solutions is shown in Fig. 4.25.

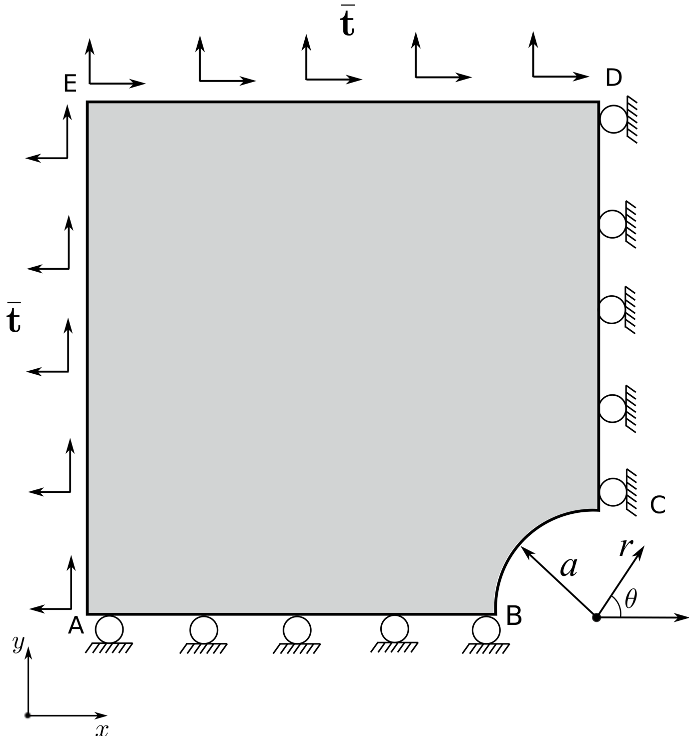


Figure 4.15: Definition of the Kirsch problem

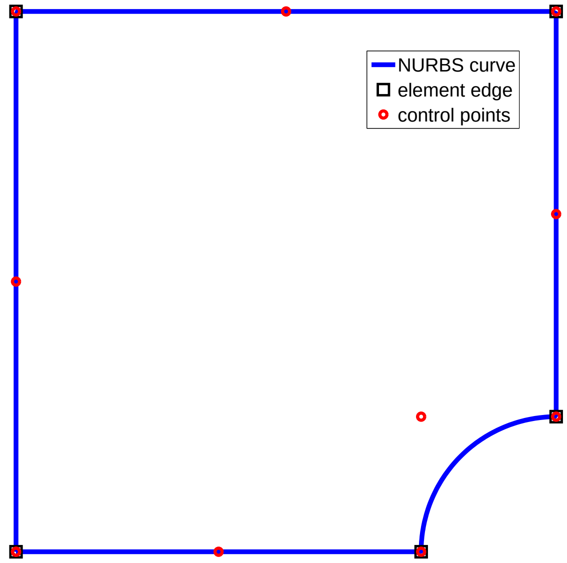


Figure 4.16: Geometric model of the Kirsch problem

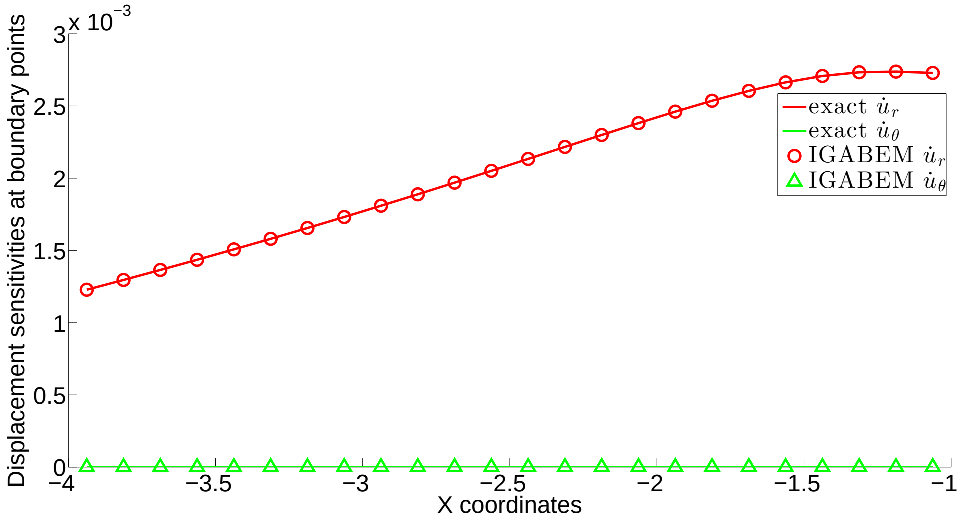


Figure 4.17: Displacement sensitivities on the edge AB of the plate

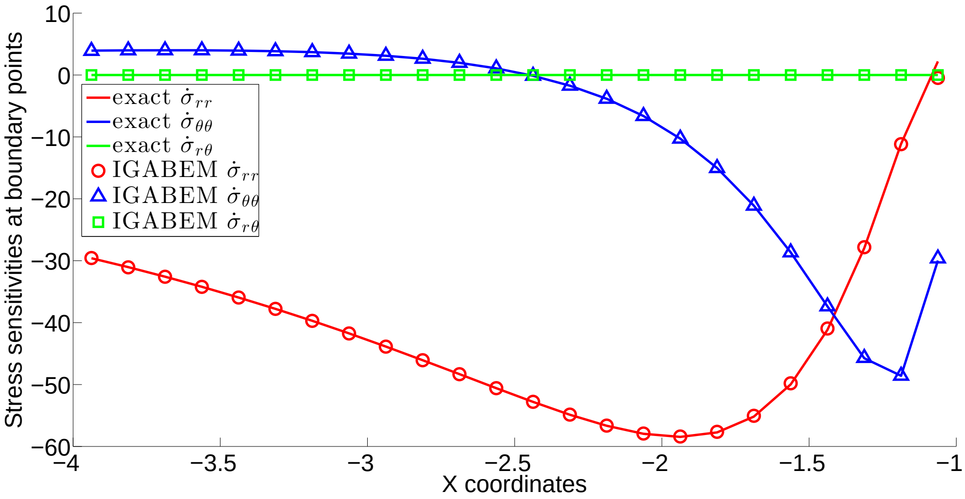


Figure 4.18: Stress sensitivities on the edge AB of the plate

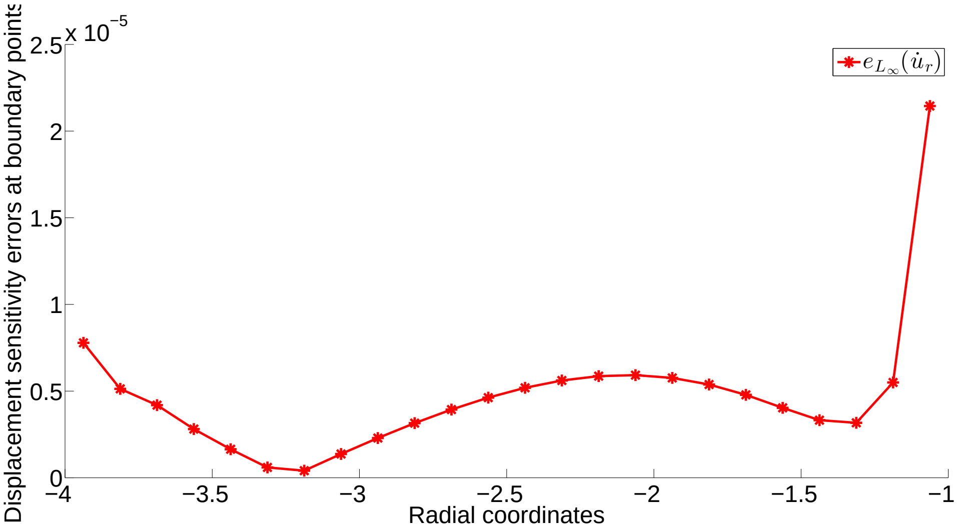


Figure 4.19: Displacement sensitivity errors on the edge AB of the plate

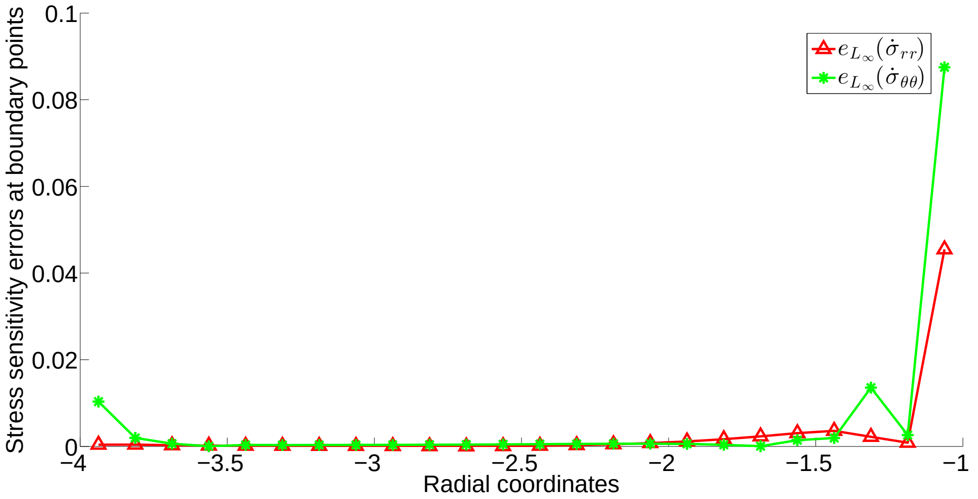


Figure 4.20: Stress sensitivity errors on the edge AB of the plate

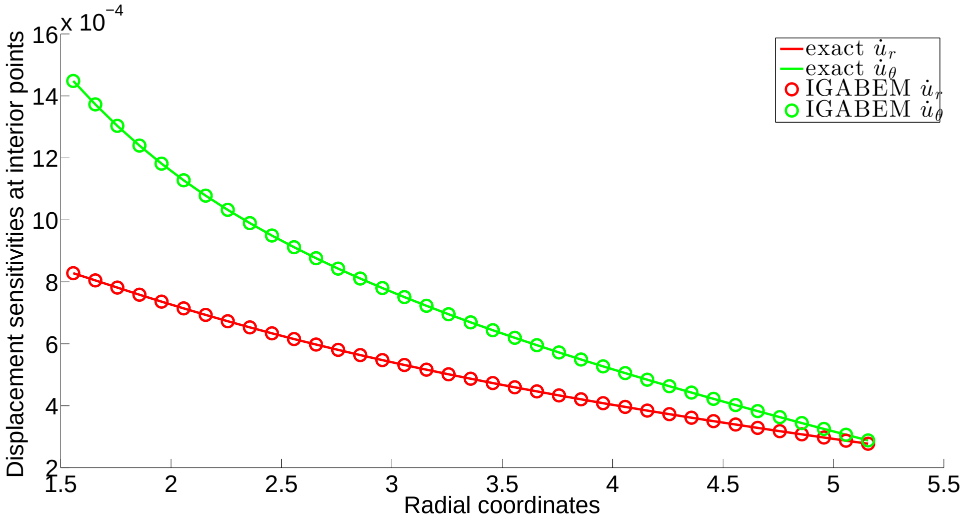


Figure 4.21: Displacement sensitivities at the interior points of the plate

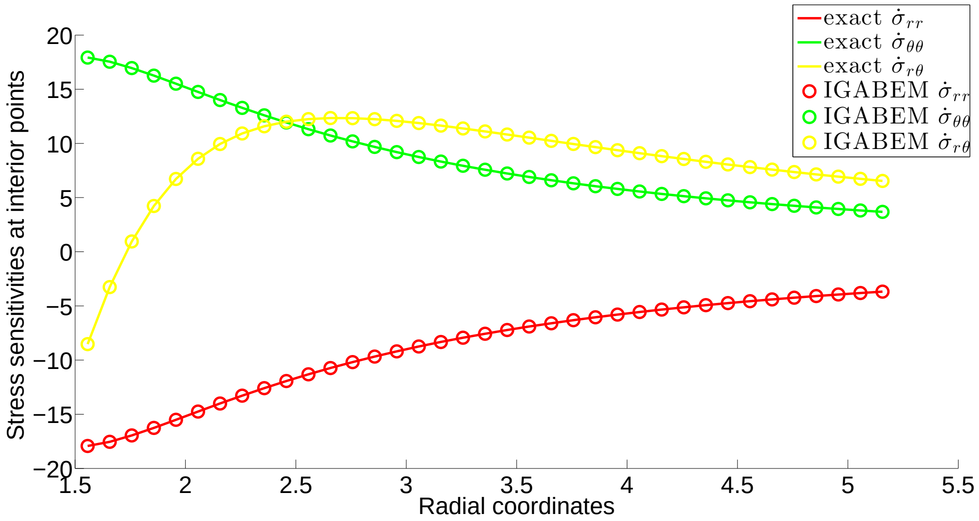


Figure 4.22: Stress sensitivities at the interior points of the plate

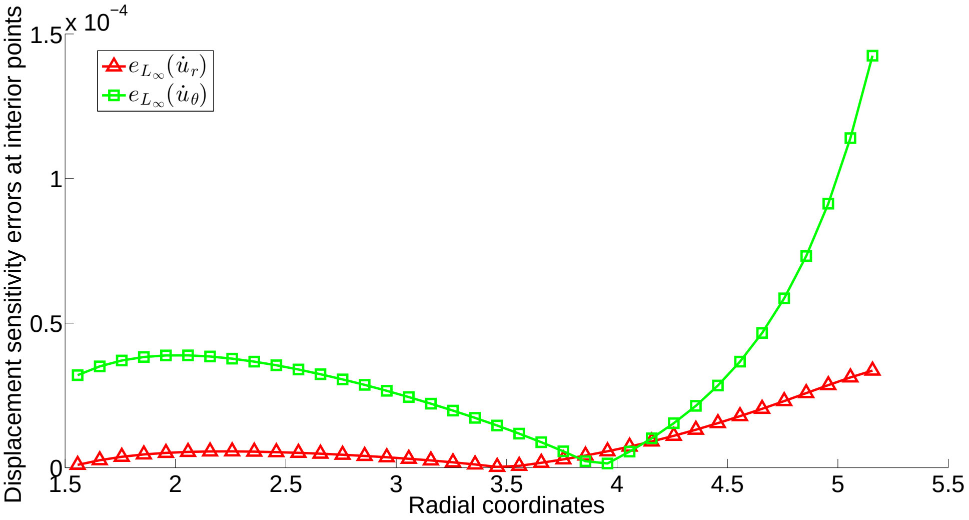


Figure 4.23: Displacement sensitivities at the interior points of the plate

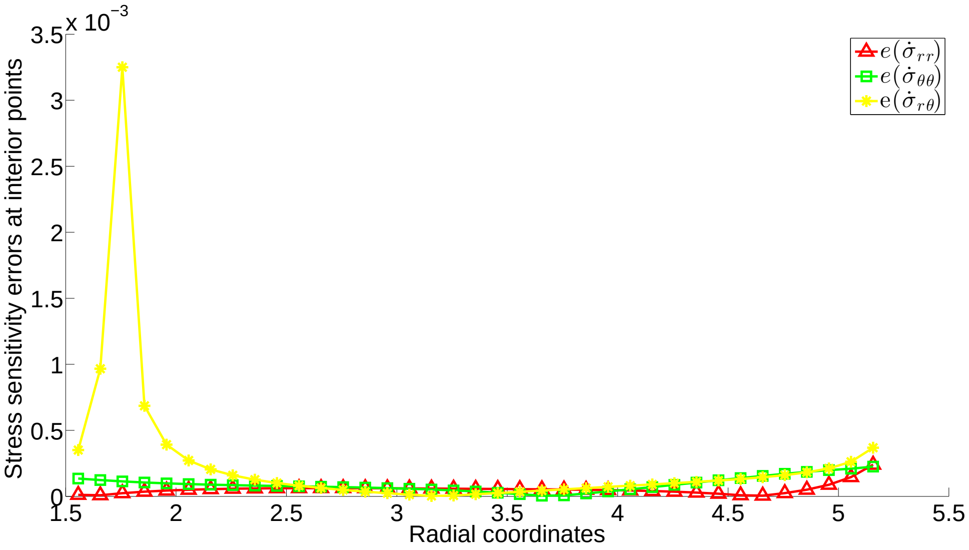


Figure 4.24: Stress sensitivity errors at the interior points of the plate

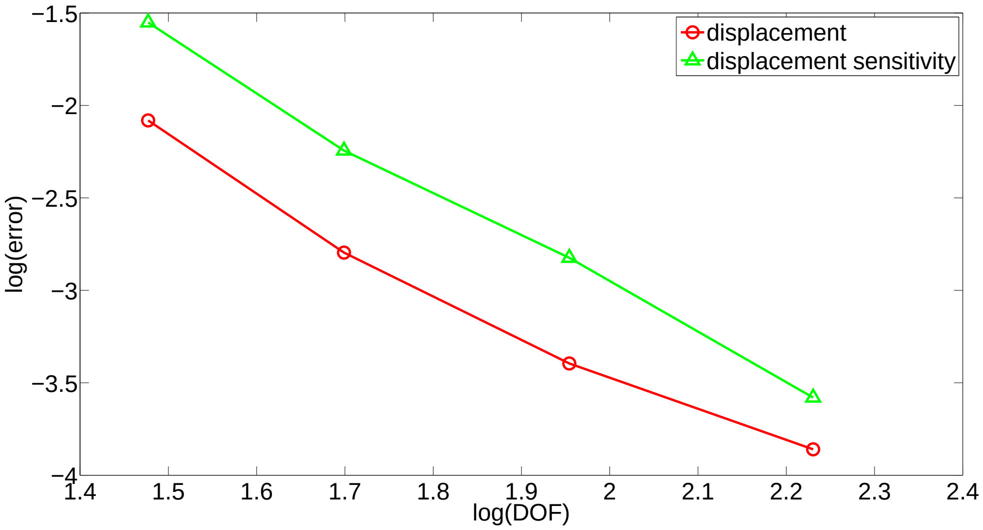


Figure 4.25: $e_{L_2}(\mathbf{u})$ and $e_{L_\infty}(\dot{\mathbf{u}})$ for the Kirsch problem

4.5.3 3D cylinder

Consider a 3D cylinder subject to the inner pressure $p = 1200$ in the direction normal to its lateral surface (Fig. 4.26). The geometry parameters are: inner radius $a = 0.5$, outer radius $b = 1$, height $h = 4$. The Young's modulus is $E = 10^5$, and Poisson's ration ν is 0. The problem is modelled by exerting the analytical displacement on the inner surface while zero tractions are enforced on all other surfaces.

The analytical displacement and stress fields in polar coordinates (r, θ) are given by

$$u_r(r,\theta) = \frac{Pa^2}{E(b^2-a^2)}\left[(1-\nu)r + \frac{b^2(1+\nu)}{r}\right], \tag{4.5.123}$$

$$\sigma_{rr}(r,\theta) = \frac{Pa^2}{b^2-a^2}\left(1-\frac{b^2}{r^2}\right), \tag{4.5.124}$$

$$\sigma_{\theta\theta}(r,\theta) = \frac{Pa^2}{b^2-a^2}\left(1+\frac{b^2}{r^2}\right). \tag{4.5.125}$$

Given the design variable as the radius of the outer boundary b, the analytical solutions of the displacement and stress sensitivities are written as

$$\begin{aligned}\dot{u}_r(r,\theta) = &-\frac{2Pa^2b}{E(b^2-a^2)^2}\left[(1-\nu)r + \frac{b^2(1+\nu)}{r}\right] \\ &+\frac{Pa^2}{E(b^2-a^2)}\left[(1-\nu)\dot{r} + (1+\nu)\frac{2br - b^2\dot{r}}{r^2}\right],\end{aligned} \tag{4.5.126}$$

$$\dot{\sigma}_{rr}(r,\theta) = \frac{-2a^2bP}{(b^2-a^2)^2}\left(1-\frac{b^2}{r^2}\right) - \frac{Pa^2}{b^2-a^2}\left(\frac{2br^2-2b^2r\dot{r}}{r^4}\right), \tag{4.5.127}$$

$$\dot{\sigma}_{\theta\theta}(r,\theta) = \frac{-2a^2bP}{(b^2-a^2)^2}\left(1+\frac{b^2}{r^2}\right) + \frac{Pa^2}{b^2-a^2}\left(\frac{2br^2-2b^2r\dot{r}}{r^4}\right). \tag{4.5.128}$$

The analysis model uses 384 Bézier elements and 864 control points. Fig. 4.27 shows the comparison between the analytical and numerical displacement sensitivities on the surface and Fig. 4.28 shows the corresponding errors. A good agreement is observed. Figs. 4.29 and 4.30 show the stress sensitivities and the errors, respectively. The numerical stress sensitivities agree with the analytical solutions well, except in the small regions close to the sharp edges where large errors are present. A possible reason is that we used a collocation scheme which shifted the collocation points from the edges, and thus cannot capture the solutions in the local region around the edges accurately.

To investigate the stress sensitivities in the domain, we assume the shape derivatives of the domain points to be

$$\dot{r} = \frac{b-r}{b-a}. \tag{4.5.129}$$

The comparison between the analytical and numerical stress sensitivities are shown in Fig. 4.31, and the corresponding relative errors are shown in Fig. 4.32. The errors are relatively larger at the inner points close to the boundary, which is caused by the nearly singular integrals. But a reasonable accuracy can still be observed. The convergence of the shape sensitivity analysis is shown in Fig. 4.33.

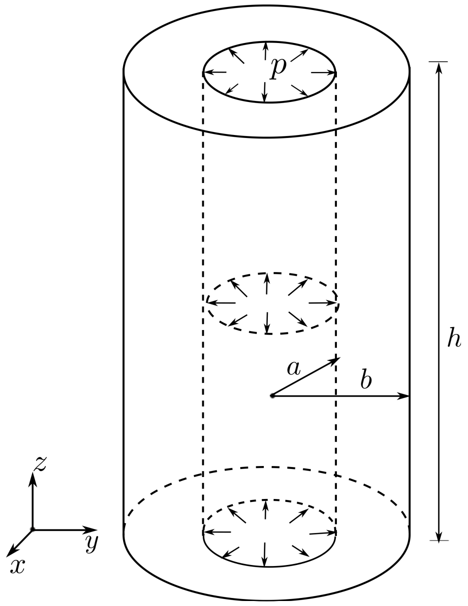
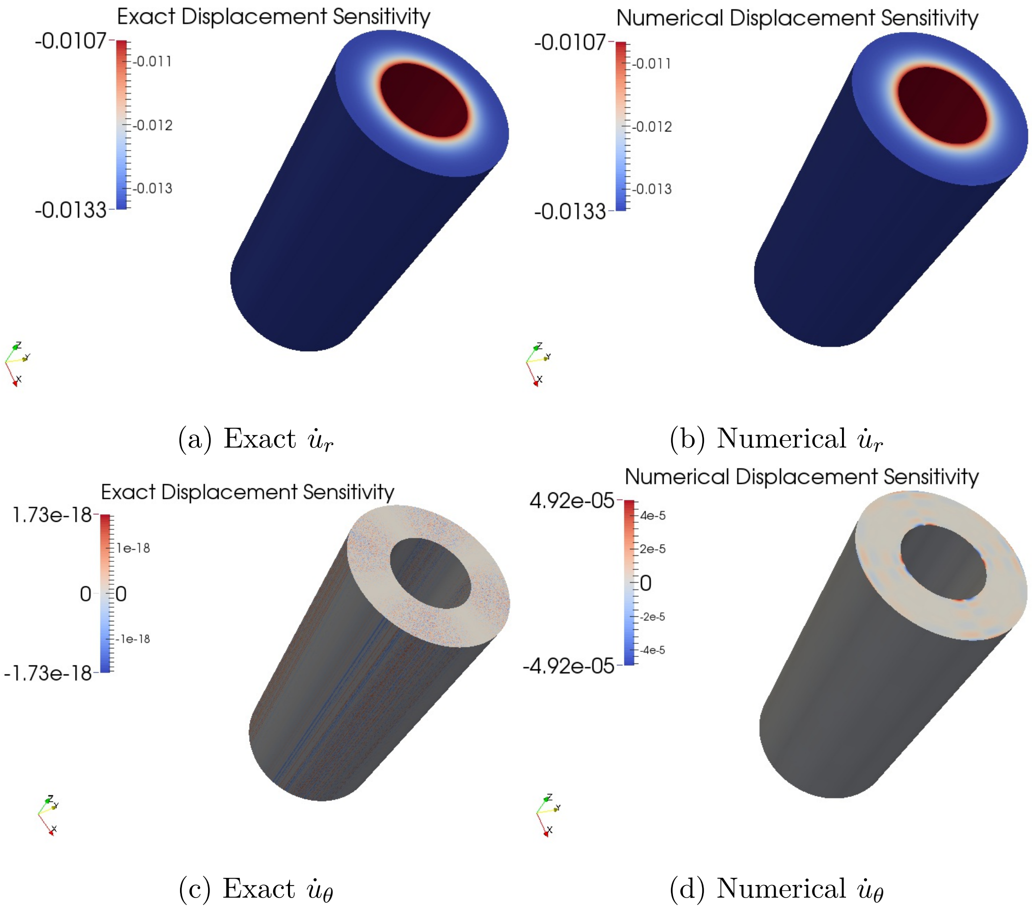


Figure 4.26: The definition of the 3D cylinder problem



(a) Exact $\dot{u}_r$

(b) Numerical $\dot{u}_r$

(c) Exact $\dot{u}_\theta$

(d) Numerical $\dot{u}_\theta$

Figure 4.27: Displacement sensitivities on the cylinder surface

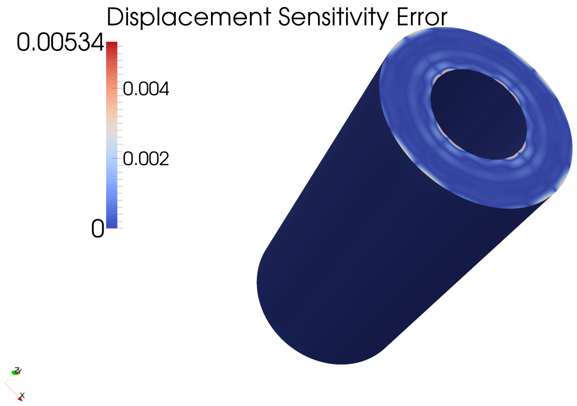
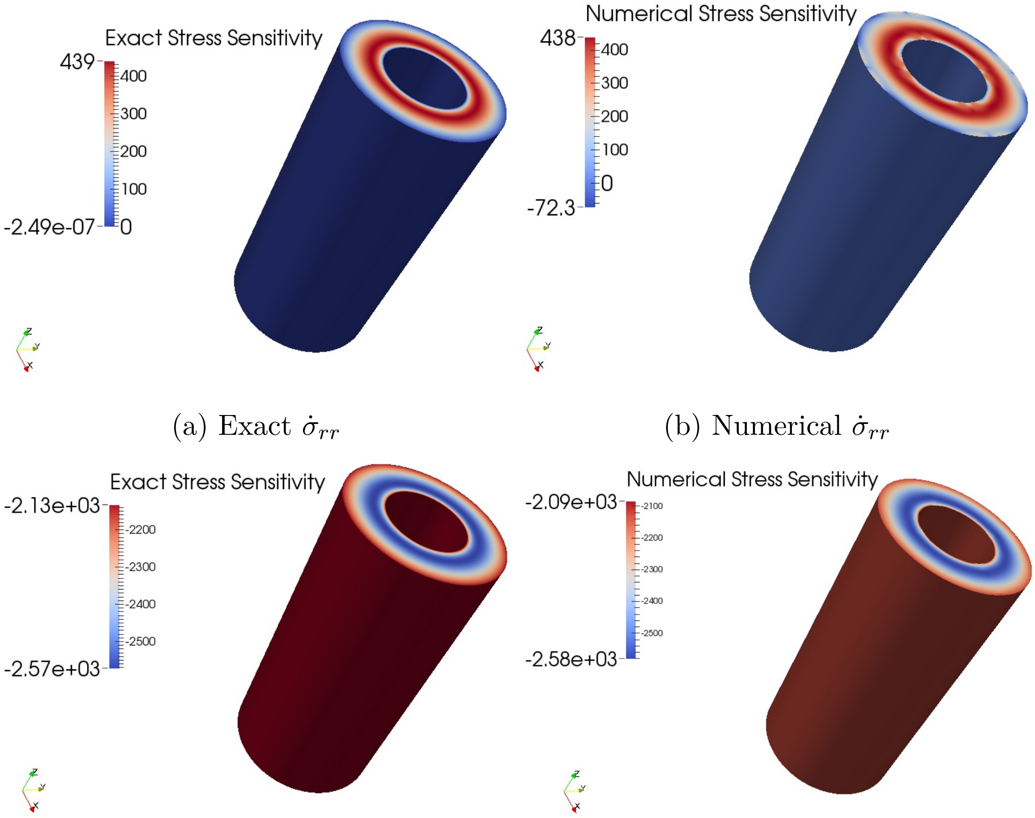


Figure 4.28: $e_{L_\infty}(\dot{u}_r)$ on the cylinder surface



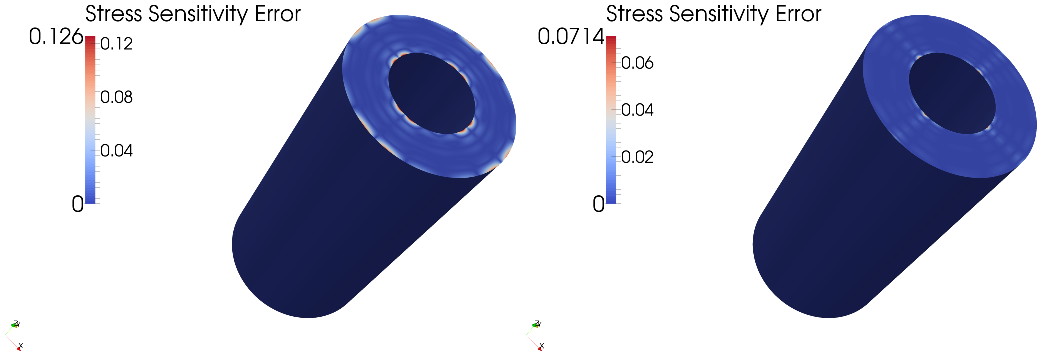
(a) Exact $\dot{\sigma}_{rr}$

(b) Numerical $\dot{\sigma}_{rr}$

(c) Exact $\dot{\sigma}_{\theta\theta}$

(d) Numerical $\dot{\sigma}_{\theta\theta}$

Figure 4.29: Stress sensitivities on the cylinder surface



(a) $e_{L_\infty}(\dot{\sigma}_{rr})$

(b) $e_{L_\infty}(\dot{\sigma}_{\theta\theta})$

Figure 4.30: Stress sensitivity errors on the cylinder surface

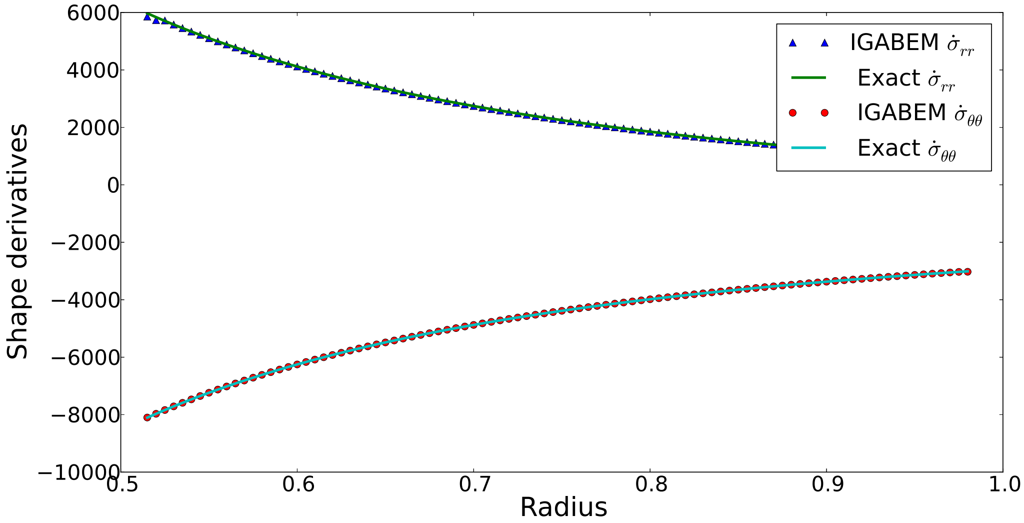


Figure 4.31: Stress sensitivities at interior points of the cylinder

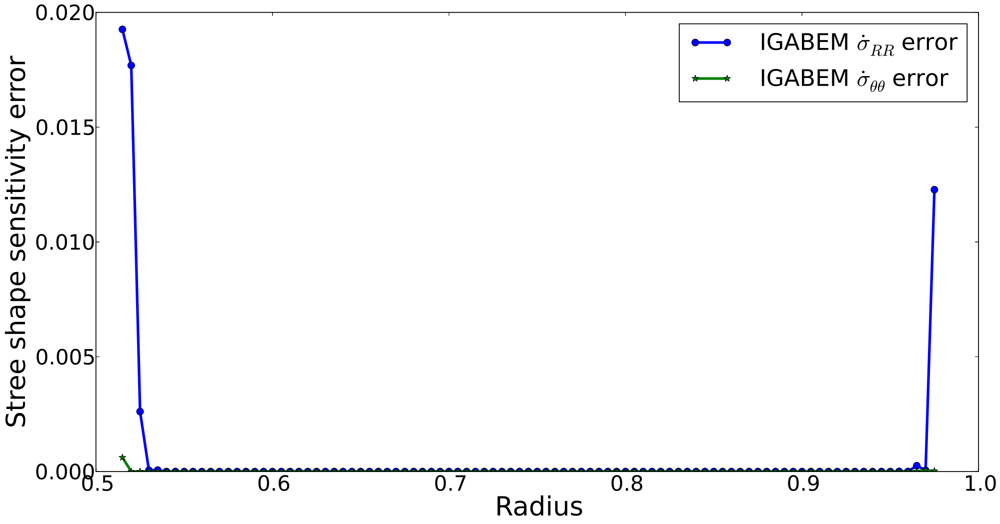


Figure 4.32: Stress sensitivity errors $e_{L_\infty}(\dot{\sigma}_{rr})$ and $e_{L_\infty}(\dot{\sigma}_{\theta\theta})$ at interior points of the cylinder

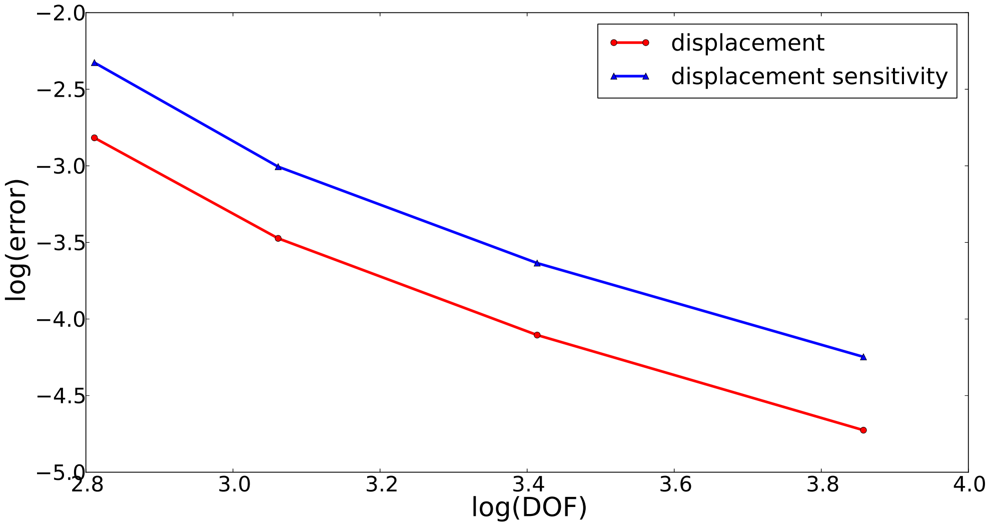


Figure 4.33: $e_{L_2}(\mathbf{u}_h)$ and $e_{L_2}(\dot{\mathbf{u}}_h\ )$ against DOF for the 3D cylinder problem

4.5.4 Spherical cavity

Fig. 4.34a shows a problem of a traction free spherical cavity in an infinite domain subject to a tension $S = 10^5$ at infinity. The problem is analyzed by extracting a finite cube domain around the cavity (Fig. 4.34b) and by exerting the analytical displacement solutions around the cube surface as boundary conditions. The radius of the cavity is $a = 0.5$, and the length of the cube is $2b = 10$. The Young's modulus is $E = 10^5$, and Poisson's ratio $\nu = 0.3$. The analytical solution of the displacement is given by

$$\begin{aligned} 2\mu u_R(R,\beta,\theta) &= -A_1 R + \frac{3}{2}\frac{A_2}{R^4} - \frac{A_3}{R^2} \\ &\quad + \left(3A_1 R - \frac{9}{2}\frac{A_2}{R^4} + B_1(4\nu - 2)R + \frac{B_2(4\nu-5)}{R^2}\right)\cos\beta, \qquad (4.5.130) \\ 2\mu u_\beta(R,\beta,\theta) &= -\left[-3A_1 R - \frac{3A_2}{R^4} + \left(B_1 R + \frac{B_2}{R^2}\right)(2-4\nu)\right]\sin\beta\cos\beta, \qquad (4.5.131) \\ u_\theta(R,\beta,\theta) &= 0, \qquad (4.5.132) \end{aligned}$$

where

$$\begin{aligned} A_1 &= \frac{S\nu}{1+\nu}, \qquad A_2 = \frac{Sa^5}{7-5\nu}, \qquad A_3 = \frac{Sa^3(6-5\nu)}{2(7-5\nu)}, \\ B_1 &= -\frac{S}{2(1+\nu)}, \quad B_2 = -\frac{5Sa^3}{2(7-5\nu)}. \qquad (4.5.133) \end{aligned}$$

The analytical stress is given by

$$\begin{aligned} \sigma_{RR}(R,\beta,\theta) &= S\cos^2\beta + \frac{S}{7-5\nu}\left\{\frac{a^3}{R^3}[6 - 5(5-\nu)\cos^2\beta] + \frac{6a^5}{R^5}(3\cos^2\beta - 1)\right\} \\ &\quad + \frac{S}{7-5\nu}\left\{\frac{6a^5}{R^5}(3\cos^2\beta - 1)\right\}, \qquad (4.5.134) \\ \sigma_{\theta\theta}(R,\beta,\theta) &= \frac{S}{2(7-5\nu)}\left\{\frac{a^3}{R^3}[5\nu - 2 + 5(1-2\nu)\cos^2\beta] + \frac{a^5}{R^5}(1 - 5\cos^2\beta)\right\} \\ &\quad + \frac{S}{2(7-5\nu)}\left\{\frac{a^5}{R^5}(1 - 5\cos^2\beta)\right\}, \qquad (4.5.135) \\ \sigma_{\beta\beta}(R,\beta,\theta) &= S\sin^2\beta + \frac{S}{2(7-5\nu)}\left\{\frac{a^3}{R^3}[4 - 5\nu + 5(1-2\nu)\cos^2\beta] + \frac{3a^5}{R^5}(3 - 7\cos^2\beta)\right\} \\ &\quad + \frac{S}{2(7-5\nu)}\left\{\frac{3a^5}{R^5}(3 - 7\cos^2\beta)\right\}, \qquad (4.5.136) \\ \sigma_{R\beta}(R,\beta,\theta) &= S\left\{-1 + \frac{1}{7-5\nu}\left[-\frac{5a^3(1+\nu)}{R^3} + \frac{12a^5}{R^5}\right]\right\}\sin\beta\cos\beta. \qquad (4.5.137) \end{aligned}$$

We take the cavity radius a as the design variable, and the analytical displacement sensitivity can be written as

$$\begin{aligned}2\mu\dot{u}_R(R,\beta,\theta) &= -(A_1\dot{R}+\dot{A}_1R)+\frac{3}{2}\dot{\left(\frac{A_2}{R^4}\right)}-\dot{\left(\frac{A_3}{R^2}\right)} \\ &\quad+\left[3A_1\dot{R}-\frac{9}{2}\dot{\left(\frac{A_2}{R^4}\right)}+B_1(4\nu-2)\dot{R}+(4\nu-5)\dot{\left(\frac{B_2}{R^2}\right)}\right]\cos\beta,\end{aligned} \tag{4.5.138}$$

$$\begin{aligned}2\mu\dot{u}_\beta(R,\beta,\theta) &= -\left\{-3A_1\dot{R}-3\dot{\left(\frac{A_2}{R^4}\right)}+\left[B_1\dot{R}+\dot{B}_1R+\dot{\left(\frac{B_2}{R^2}\right)}\right](2-4\nu)\right\} \\ &\quad\times\sin\beta\cos\beta,\end{aligned} \tag{4.5.139}$$

$$\dot{u}_\theta(R,\beta,\theta) = 0, \tag{4.5.140}$$

with

$$\begin{aligned}&\dot{A}_1=0,\quad \dot{A}_2=\frac{5Sa^4}{7-5\nu},\qquad \dot{A}_3=\frac{3Sa^2(6-5\nu)}{2(7-5\nu)},\\ &\dot{B}_1=0,\quad \dot{B}_2=-\frac{15Sa^2}{2(7-5\nu)},\end{aligned} \tag{4.5.141}$$

and

$$\dot{\left(\frac{A_2}{R^4}\right)}=\frac{\dot{A}_2R^4-4A_2R^3\dot{R}}{R^8}, \tag{4.5.142}$$

$$\dot{\left(\frac{A_3}{R^2}\right)}=\frac{\dot{A}_3R^2-2A_3R\dot{R}}{R^4}, \tag{4.5.143}$$

$$\dot{\left(\frac{B_2}{R^2}\right)}=\frac{\dot{B}_2R^2-2B_2R\dot{R}}{R^4}. \tag{4.5.144}$$

And the analytical stress sensitivity is expressed by

$$\begin{aligned}\dot{\sigma}_{RR}(R,\beta,\theta) &= \frac{S}{7-5\nu}\left[\dot{\left(\frac{a^3}{R^3}\right)}(6-5(5-\nu)\cos^2\beta)\right] \\ &\quad+\frac{S}{7-5\nu}\left[6\dot{\left(\frac{a^5}{R^5}\right)}(3\cos^2\beta-1)\right],\end{aligned} \tag{4.5.145}$$

$$\begin{aligned}\dot{\sigma}_{\theta\theta}(R,\beta,\theta) &= \frac{S}{2(7-5\nu)}\left[\dot{\left(\frac{a^3}{R^3}\right)}(5\nu-2+5(1-2\nu)\cos^2\beta)\right] \\ &\quad+\frac{S}{2(7-5\nu)}\left[\dot{\left(\frac{a^5}{R^5}\right)}(1-5\cos^2\beta)\right],\end{aligned} \tag{4.5.146}$$

$$
\begin{aligned}
\dot{\sigma}_{\beta\beta}(R,\beta,\theta) &= \frac{S}{2(7-5\nu)}\left[\left(\dot{\frac{a^3}{R^3}}\right)(4-5\nu+5(1-2\nu)\cos^2\beta)\right] \\
&+\frac{S}{2(7-5\nu)}\left[3\left(\dot{\frac{a^5}{R^5}}\right)(3-7\cos^2\beta)\right], \qquad (4.5.147)\\
\dot{\sigma}_{R\beta}(R,\beta,\theta) &= S\left\{\frac{1}{7-5\nu}\left[-5(1+\nu)\left(\dot{\frac{a^3}{R^3}}\right)+12\left(\dot{\frac{a^5}{R^5}}\right)\right]\right\}\sin\beta\cos\beta, \qquad (4.5.148)
\end{aligned}
$$

with

$$
\left(\dot{\frac{a^3}{R^3}}\right) = \frac{3a^2R^3-3a^3R^2\dot{R}}{R^6}, \qquad (4.5.149)
$$

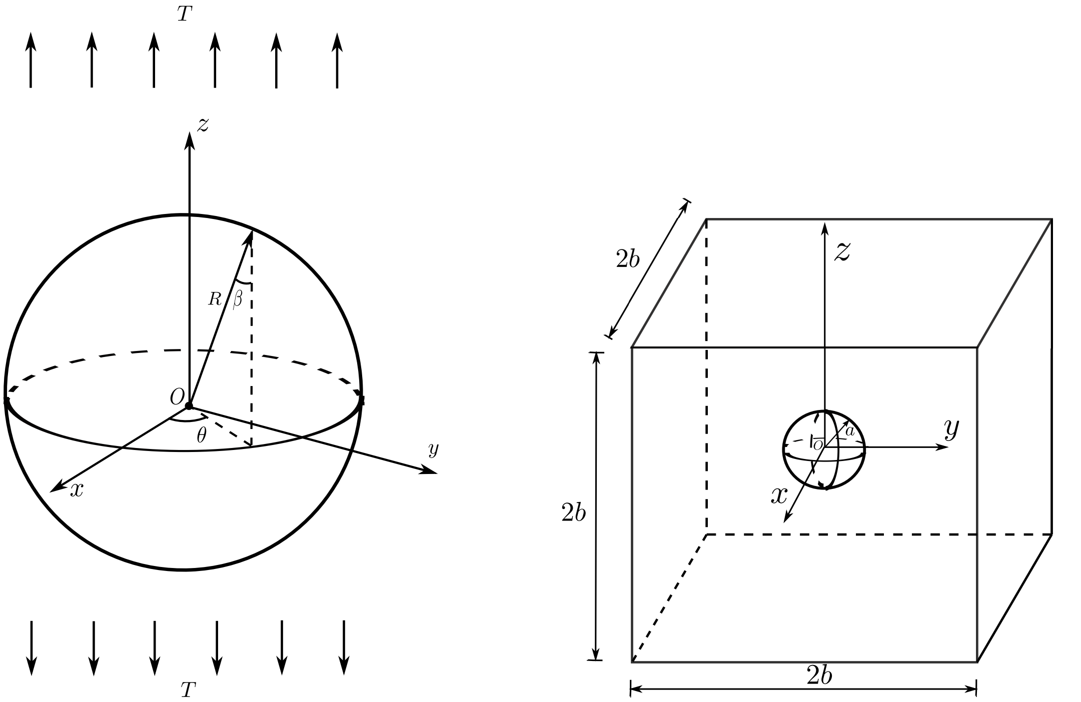
$$
\left(\dot{\frac{a^5}{R^5}}\right) = \frac{5a^4R^5-5a^5R^4\dot{R}}{R^{10}}. \qquad (4.5.150)
$$

The boundary conditions for structural and shape sensitivity analysis are enforced using the Galerkin nodal parameter extraction method. The analysis model has 224 Bézier elements on the surface, and 560 control points.

The comparison between the analytical and the numerical displacement sensitivities on the surface is shown in Fig. 4.35, and the displacement sensitivity errors are shown in Fig. 4.36. An excellent agreement can be seen. To investigate the shape sensitivities at the interior points, we take an inner spherical surface in the domain with the radius R =2.5. The points in the domain are assumed to be regularly distributed along the radial line, *i.e.*

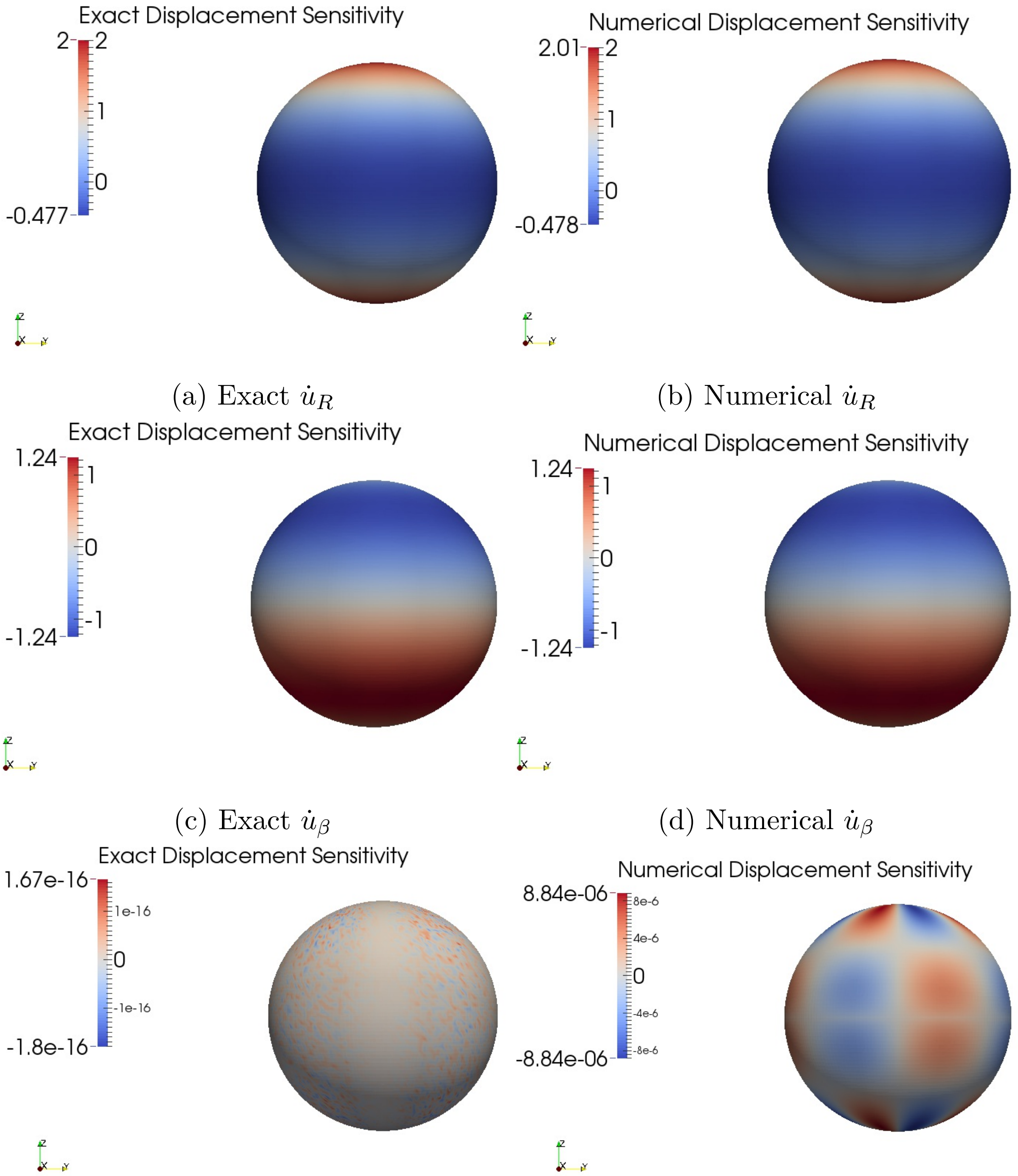
$$
\dot{R} = \begin{cases}
\dfrac{\left|\frac{L}{\cos\theta\sin\beta}\right|-R}{\left|\frac{L}{\cos\theta\sin\beta}\right|-a} & \text{if } |x| \geqslant |y| \text{ and } \frac{L}{|L/\cos\theta|} \geqslant \left|\frac{z}{R\sin\beta}\right|, \\[2ex]
\dfrac{\left|\frac{L}{\cos\beta}\right|-R}{\left|\frac{L}{\cos\beta}\right|-a} & \text{if } |x| \geqslant |y| \text{ and } \frac{L}{|L/\cos\theta|} < \left|\frac{z}{R\sin\beta}\right|, \\[2ex]
\dfrac{\left|\frac{L}{\sin\theta\sin\beta}\right|-R}{\left|\frac{L}{\sin\theta\sin\beta}\right|-a} & \text{if } |x| < |y| \text{ and } \frac{L}{|L/\sin\theta|} \geqslant \left|\frac{z}{R\sin\beta}\right|, \\[2ex]
\dfrac{\left|\frac{L}{\cos\beta}\right|-R}{\left|\frac{L}{\cos\beta}\right|-a} & \text{if } |x| < |y| \text{ and } \frac{L}{|L/\sin\theta|} < \left|\frac{z}{R\sin\beta}\right|.
\end{cases} \qquad (4.5.151)
$$

The good agreements with analytical displacement and stress sensitivities are shown in Figs. 4.37 and 4.39, respectively. The related errors are shown in Figs. 4.38 and 4.40. The convergence study can be found in Fig. 4.41.



(a) (b)

Figure 4.34: (a) The definition of the spherical cavity problem, and (2) the analysis model of the spherical cavity problem



(a) Exact $\dot{u}_R$

(b) Numerical $\dot{u}_R$

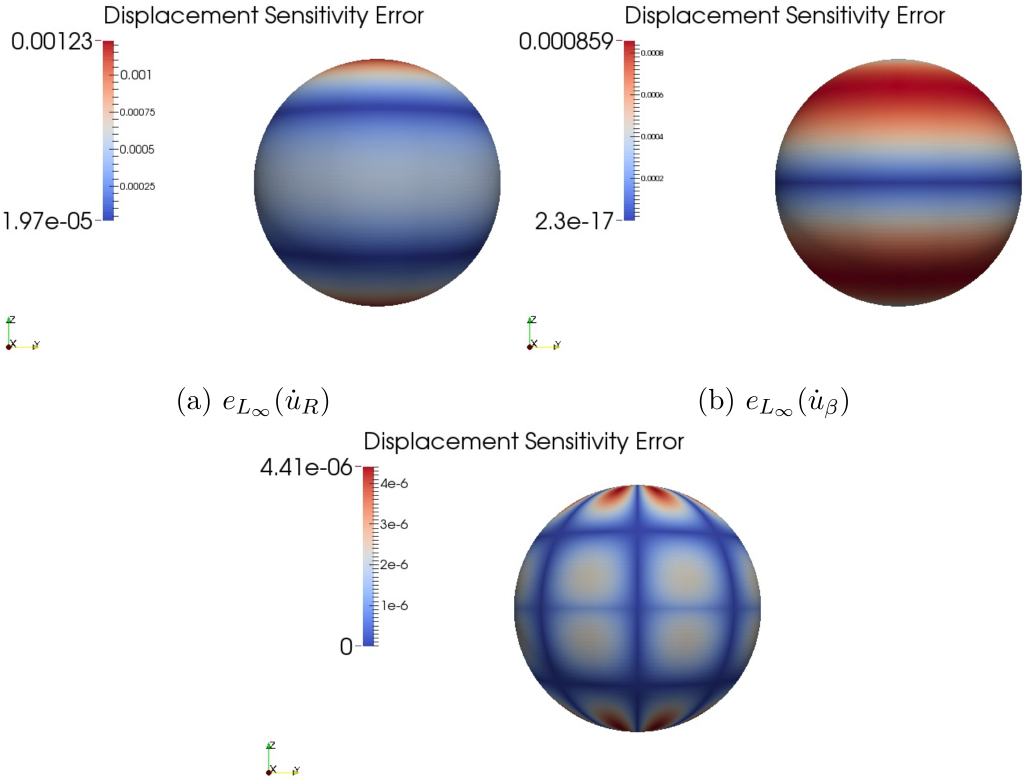
(c) Exact $\dot{u}_\beta$

(d) Numerical $\dot{u}_\beta$

(e) Exact $\dot{u}_\theta$

(f) Numerical $\dot{u}_\theta$

Figure 4.35: Displacement sensitivities on the cavity surface

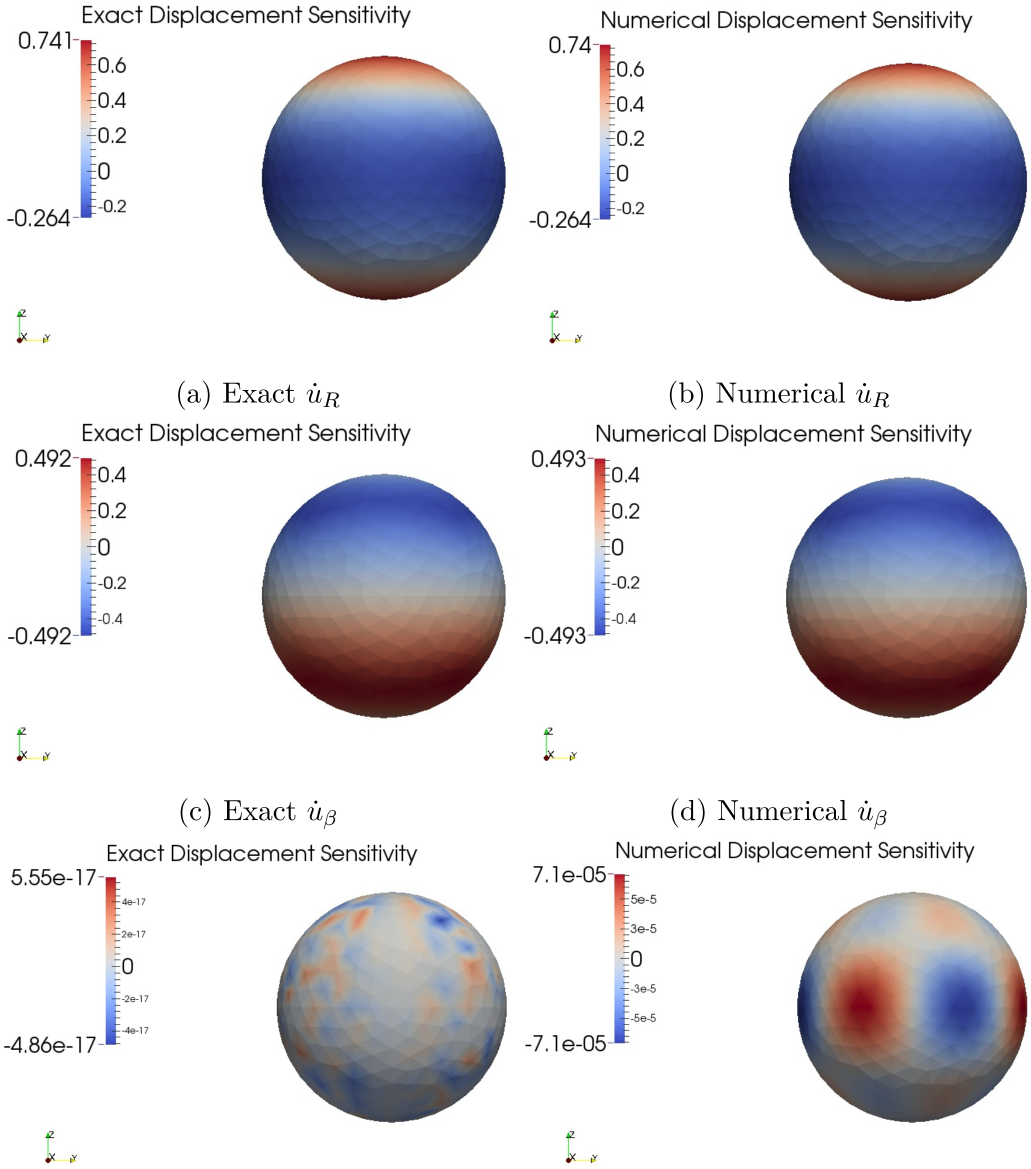


(a) $e_{L_\infty}(\dot{u}_R)$

(b) $e_{L_\infty}(\dot{u}_\beta)$

(c) $e_{L_\infty}(\dot{u}_\theta)$

Figure 4.36: Displacement sensitivity errors on the cavity surface



(a) Exact $\dot{u}_R$

(b) Numerical $\dot{u}_R$

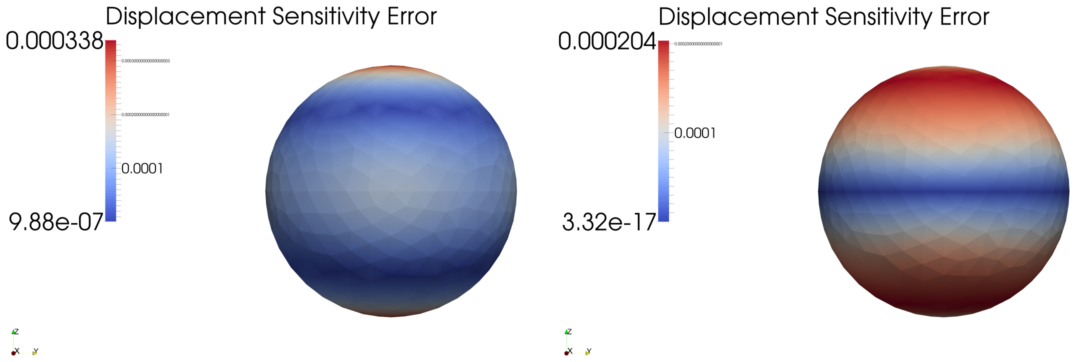
(c) Exact $\dot{u}_\beta$

(d) Numerical $\dot{u}_\beta$

(e) Exact $\dot{u}_\theta$

(f) Numerical $\dot{u}_\theta$

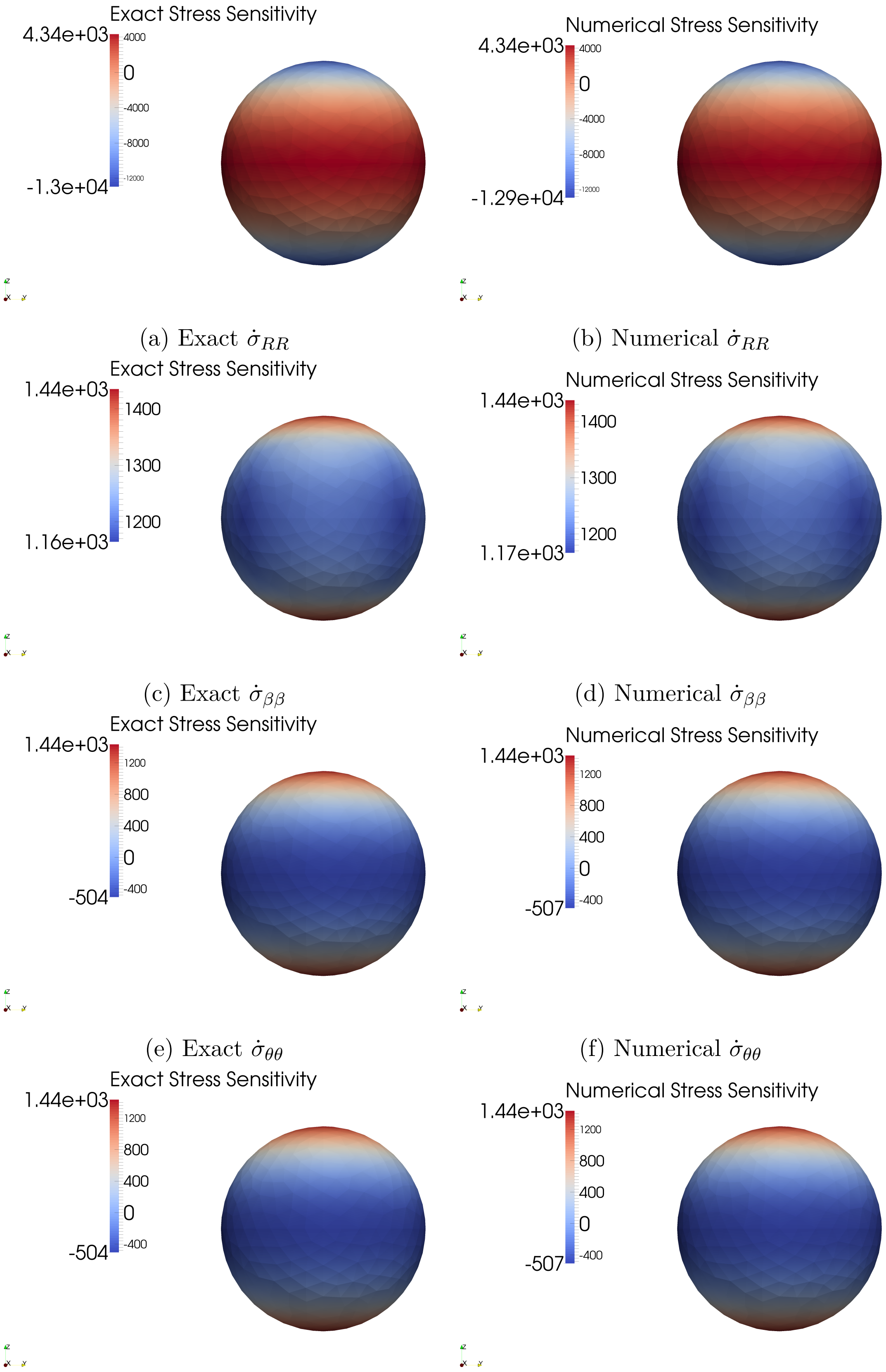
Figure 4.37: Displacement sensitivities at interior points of the cavity ($R = 2.5$)



(a) $e_{L_\infty}(\dot{u}_R)$

(b) $e_{L_\infty}(\dot{u}_\beta)$

Figure 4.38: Displacement sensitivity errors at interior points of the cavity ($R = 2.5$)



(a) Exact $\dot{\sigma}_{RR}$

(b) Numerical $\dot{\sigma}_{RR}$

(c) Exact $\dot{\sigma}_{\beta\beta}$

(d) Numerical $\dot{\sigma}_{\beta\beta}$

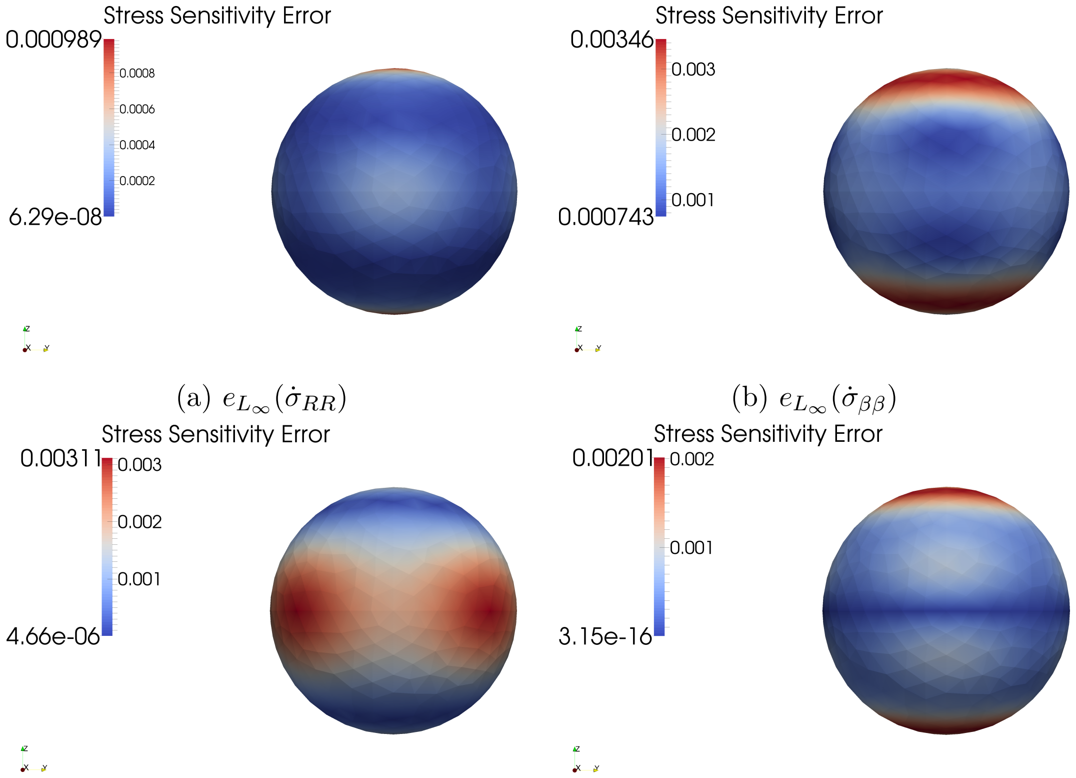
(e) Exact $\dot{\sigma}_{\theta\theta}$

(f) Numerical $\dot{\sigma}_{\theta\theta}$

(g) Exact $\dot{\sigma}_{R\beta}$

(h) Numerical $\dot{\sigma}_{R\beta}$

Figure 4.39: Stress sensitivities at interior points of the cavity ($R = 2.5$)



(a) $e_{L_\infty}(\dot{\sigma}_{RR})$

(b) $e_{L_\infty}(\dot{\sigma}_{\beta\beta})$

(c) $e_{L_\infty}(\dot{\sigma}_{\theta\theta})$

(d) $e_{L_\infty}(\dot{\sigma}_{R\beta})$

Figure 4.40: Stress sensitivity errors at interior points of the cavity ($R = 2.5$)

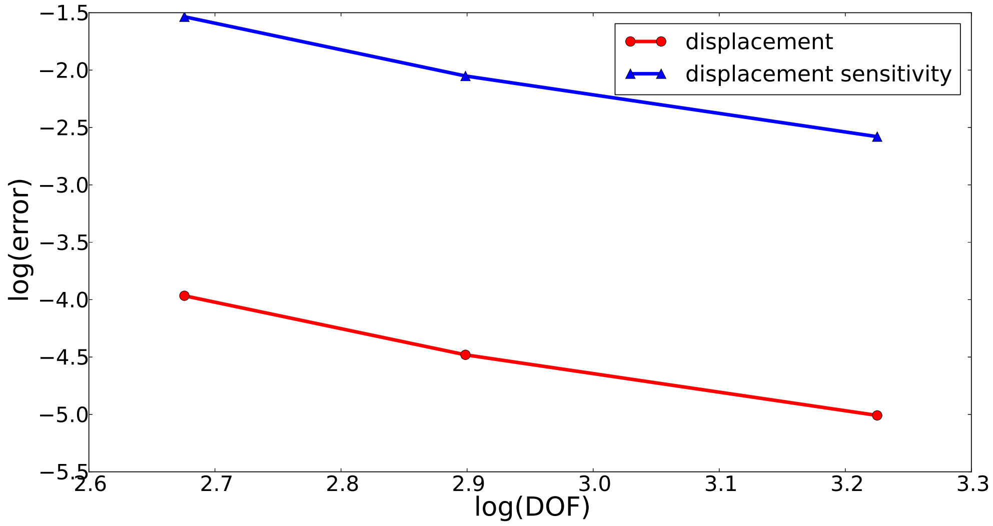


Figure 4.41: $e_{L_2}(\mathbf{u}_h)$ and $e_{L_2}(\dot{\mathbf{u}}_h)$ against DOF for the cavity problem

4.6 Conclusions

The formulation of shape sensitivity analysis using IGABEM has been presented. The NURBS and T-splines are used to discretize the material differentiation form of the regularized boundary integral equation. The shape sensitivity analysis is based on the exact geometry as CAD, thereby removing the geometric errors.

Chapter 5

Shape Optimization with IGABEM

In this chapter, we apply the isogeometric boundary element method (IGABEM) to gradient-based shape optimization in linear elasticity. A seamless integration of CAD and analysis renders IGABEM with immediate advantages for its application in shape optimization: 1) meshing procedure is completely bypassed, and 2) the returned optimized model can be directly used in CAD without needing any smoothing or recovery procedure. See Fig. 5.1. In the remainder of the chapter, a remark on the implementation of the IGABEM will be given firstly. Thereafter, we will present numerical examples to demonstrate the application of the IGABEM in shape optimization in linear elasticity.

5.1 Remarks on the effective implementation

5.1.1 Shape derivatives of some quantities of interest

The displacement and stress shape sensitivities can be obtained from the procedure demonstrated in Chapter 6. However, it remains to calculate the sensitivities of some commonly used quantities dependent on displacements and stresses. To be consistent with our CAD and analysis models, all of the domain integrals involved should be transformed into boundary integrals.

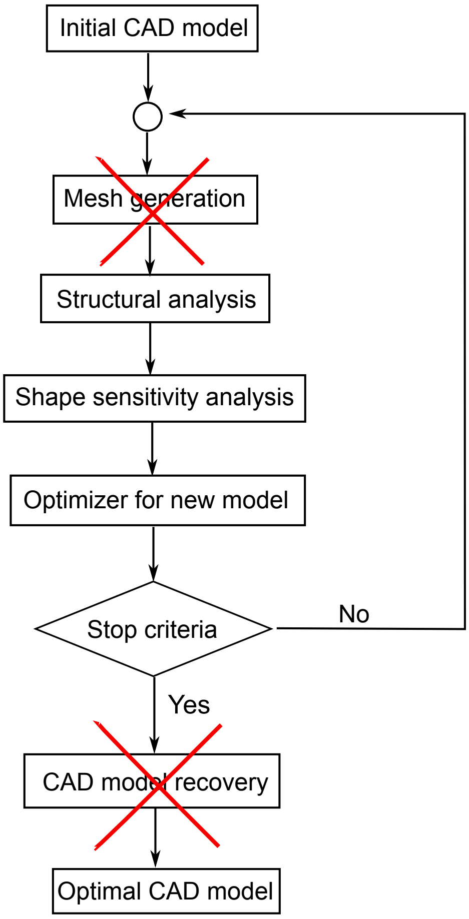


Figure 5.1: IGABEM shape optimization flowchart

- The shape derivatives of the volume V or area A. The volume and area can be transferred into boundary integral readily by using the divergence theorem

$$V=\int_{\Omega}\mathrm{d}\Omega=\frac{1}{3}\int_{\Omega}\nabla\cdot\mathbf{x}\mathrm{d}\Omega=\frac{1}{3}\int_{S}\mathbf{x}\cdot\mathbf{n}\mathrm{d}S=\frac{1}{3}\sum_{e=1}^{n_e}\int_{\tilde{S}}\mathbf{x}\cdot\mathbf{n}J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S},\quad(5.1.1)$$

$$A=\int_{\Omega}\mathrm{d}\Omega=\frac{1}{2}\int_{\Omega}\nabla\cdot\mathbf{x}\mathrm{d}\Omega=\frac{1}{2}\int_{S}\mathbf{x}\cdot\mathbf{n}\mathrm{d}S=\frac{1}{2}\sum_{e=1}^{n_e}\int_{\tilde{S}}\mathbf{x}\cdot\mathbf{n}J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}.\quad(5.1.2)$$

So the shape derivatives are

$$\begin{aligned}\dot{V}&=\frac{1}{3}\sum_{e=1}^{n_e}\int_{\tilde{S}}\left[\mathbf{x}\cdot\mathbf{n}J_e(\tilde{\boldsymbol{\xi}})\right]\mathrm{d}\tilde{S}\\&=\frac{1}{3}\sum_{e=1}^{n_e}\int_{\tilde{S}}\left[\dot{\mathbf{x}}\cdot\mathbf{n}J_e(\tilde{\boldsymbol{\xi}})+\mathbf{x}\cdot\dot{\mathbf{n}}J_e(\tilde{\boldsymbol{\xi}})+\mathbf{x}\cdot\mathbf{n}\dot{J}_e(\tilde{\boldsymbol{\xi}})\right]\mathrm{d}\tilde{S},\end{aligned}\qquad(5.1.3)$$

$$\dot{A}=\frac{1}{2}\sum_{e=1}^{n_e}\int_{\tilde{S}}\left[\mathbf{x}\cdot\mathbf{n}J_e(\tilde{\xi})\right]\mathrm{d}\tilde{S}$$

$$= \frac{1}{2}\sum_{e=1}^{n_e}\int_{\tilde{S}}\left[\dot{\mathbf{x}}\cdot\mathbf{n}J_e(\tilde{\xi})+\mathbf{x}\cdot\dot{\mathbf{n}}J_e(\tilde{\xi})+\mathbf{x}\cdot\mathbf{n}\dot{J}_e(\tilde{\xi})\right]\mathrm{d}\tilde{S}. \tag{5.1.4}$$

- The shape derivatives of the von Mises stress σ_{vm}. The expression of σ_{vm} is given by

$$\sigma_{\mathrm{vm}} = \left(\frac{3}{2}s_{ij}s_{ij}\right)^{\frac{1}{2}}, \tag{5.1.5}$$

with s_{ij} the components of the stress deviator tensor, which is given by

$$s_{ij} = \sigma_{ij} - \frac{1}{3}\sigma_{kk}\delta_{ij}. \tag{5.1.6}$$

In two dimensional problems, the von Mises stress can be written as

$$\sigma_{\mathrm{vm}} = \sqrt{\sigma_{11}^2+\sigma_{22}^2+3\sigma_{12}^2-\sigma_{11}\sigma_{22}}, \tag{5.1.7}$$

and its sensitivity is

$$\dot{\sigma}_{\mathrm{vm}} = \frac{(2\sigma_{11}-\sigma_{22})\dot{\sigma}_{11}+(2\sigma_{22}-\sigma_{11})\dot{\sigma}_{22}+6\sigma_{12}\dot{\sigma}_{12}}{2\sigma_{\mathrm{vm}}}. \tag{5.1.8}$$

In three dimensional problems, von Mises stress is

$$\sigma_{\mathrm{vm}} = \sqrt{\frac{1}{2}[(\sigma_{11}-\sigma_{22})^2+(\sigma_{22}-\sigma_{33})^2+(\sigma_{33}-\sigma_{11})^2+6(\sigma_{12}^2+\sigma_{23}^2+\sigma_{31}^2)]}, \tag{5.1.9}$$

and its sensitivity is expressed by

$$\begin{aligned}\dot{\sigma}_{\mathrm{vm}} =& \frac{(\sigma_{11}-\sigma_{22})(\dot{\sigma}_{11}-\dot{\sigma}_{22})+(\sigma_{22}-\sigma_{33})(\dot{\sigma}_{22}-\dot{\sigma}_{33})+(\sigma_{33}-\sigma_{11})(\dot{\sigma}_{33}-\dot{\sigma}_{11})}{2\sigma_{\mathrm{vm}}}\\ &+\frac{6(\sigma_{12}\dot{\sigma}_{12}+\sigma_{23}\dot{\sigma}_{23}+\sigma_{31}\dot{\sigma}_{31})}{2\sigma_{\mathrm{vm}}}.\end{aligned} \tag{5.1.10}$$

- The shape derivatives of the conserved energy E,

$$E = \int_{\Omega}\sigma_{ij}\epsilon_{ij}\mathrm{d}\Omega = \int_S t_i u_i \mathrm{d}S. \tag{5.1.11}$$

Its shape derivative $\dot{E}$ is given by

$$\begin{aligned}\dot{\left[\int_S t_i(\mathbf{x})\,u_i(\mathbf{x})\,\mathrm{d}S(\mathbf{x})\right]} =& \int_{\tilde{S}}[\dot{t}_i(\tilde{\boldsymbol{\xi}})u_i(\tilde{\boldsymbol{\xi}})+t_i(\tilde{\boldsymbol{\xi}})\dot{u}_i(\tilde{\boldsymbol{\xi}})]J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}\\ &+\int_{\tilde{S}}t_i(\tilde{\boldsymbol{\xi}})u_i(\tilde{\boldsymbol{\xi}})\dot{J}_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}.\end{aligned} \tag{5.1.12}$$

5.1.2 Shape sensitivity transition in NURBS

Numerical analysis always requires a sufficiently refined control mesh to reproduce the solution accurately. In contrast, a relatively coarse geometrical mesh is preferred in CAD and shape optimization, because an unnecessary refinement will introduce redundant design variables, leading to a costly shape sensitivity analysis and oscillatory geometries. To take advantages of refined meshes for stress analysis, and coarse meshes for model design and optimization, we evaluated the shape derivatives of the quantities in refined meshes with respect to the design variables in the coarse meshes [81].

Recall the knot insertion algorithm in NURBS for adding new control points while keeping the geometry unchanged,

$$\bar{\tilde{\mathbf{P}}}_A = \begin{cases} \tilde{\mathbf{P}}_1 & A = 1, \\ \alpha_A \tilde{\mathbf{P}}_A + (1 - \alpha_A)\tilde{\mathbf{P}}_{A-1} & 1 < A < m, \\ \tilde{\mathbf{P}}_n & A = m, \end{cases} \tag{5.1.13}$$

with

$$\alpha_A = \begin{cases} 1 & 1 \leqslant A \leqslant k - p, \\ \frac{\bar{\xi}_A - \xi_A}{\xi_{A+p} - \xi_A} & k - p + 1 \leqslant A \leqslant k, \\ 0 & A \geqslant k + 1, \end{cases} \tag{5.1.14}$$

where $\tilde{\mathbf{P}}$ are the weighted control points in NURBS before refinement, $\bar{\tilde{\mathbf{P}}}_A$ the added weighted control points by knot insertion or repetition. Given the shape derivatives in the mesh of $\dot{\tilde{\mathbf{P}}}$ with respect to a given design variable, which can be a control point in the same mesh, the shape derivatives of a weighted point $\bar{\tilde{\mathbf{P}}}_A$ in the refined mesh can be obtained by taking derivatives in Eq. (5.1.13)

$$\dot{\bar{\tilde{\mathbf{P}}}}_A = \begin{cases} \dot{\tilde{\mathbf{P}}}_1 & A = 1, \\ \alpha_A \dot{\tilde{\mathbf{P}}}_A + (1 - \alpha_A)\dot{\tilde{\mathbf{P}}}_{A-1} & 1 < A < m, \\ \dot{\tilde{\mathbf{P}}}_n & A = m. \end{cases} \tag{5.1.15}$$

After that, the control point derivatives $\dot{\bar{\mathbf{P}}}_A$ are obtained by dividing $\dot{\bar{\tilde{\mathbf{P}}}}_A$ by the weights. Now the shape derivatives transited from a coarse mesh to a refined mesh. Through this approach, the shape sensitivity analysis mesh is separated from the design mesh. It should be noted that the analysis and geometry design still share the same model, only in different levels of refinement.

5.1.3 The control point derivatives

The shape derivatives of the geometry point is determined by that of the control points

$$\dot{\mathbf{x}} = \sum_{A}^{n_A} R_A \dot{\mathbf{P}}_A, \tag{5.1.16}$$

where the control point sensitivities are evaluated depending on the following three cases:

- Design control points. For the control points which are set to be the design variables, the associated shape derivatives are unity.

- Fixed control points. Some control points are not influenced by design variables in the optimization procedure, and the corresponding sensitivities are zero.

- Adjoint control points. Some control points should move in the same way to keep a required geometry. So the shape derivatives are the same for these the design points.

- Linked control points. To keep the geometry vary reasonably, some control points which are not the design variables also need to move according to some rules. The shape derivatives can be derived from the design points. The relationships between design points and linked points can be exerted by the designer in any way provided it can lead to a reasonable geometry.

5.1.4 Side constraint

The side constraints are used to specify the range of the design variables, such as

$$t_i^l \leqslant t_i \leqslant t_i^u, \tag{5.1.17}$$

where t_i^l and t_i^u are the lower bound and upper bound for the ith design variables. The side constraints can guarantee that the result is not a meaningless geometry, for example, not splitting or crossing over.

5.2 Shape optimization numerical examples

In this section we will investigate the performance of the IGABEM in shape optimization. In the following examples, the geometries in two dimensions are modelled using NURBS and the shape sensitivity transition technique is employed. In three dimensions, the geometries are designed using T-splines, with the models exported from the Rhino T-spline plugin [97]. The optimization solver uses the method of moving asymptotes (MMA) [95].

5.2.1 Cantilever beam

The problem is a cantilever beam subject to a distributed traction $\bar{t} = 2$ on the beam end (Fig. 5.2). In the implementation, the traction is imposed on the end element of the beam bottom segment in the design model(Fig. 5.3). The initial geometry parameters are length $a = 30$ and height $b = 6$. All of the control point weights are 1. The material parameters are Young's modulus $E = 210 \times 10^3$ and Poisson's ratio $\nu = 0.3$. The optimization objective is to minimize the displacement of the beam end. The design model to be optimized uses quadratic NURBS curve with 20 control points and 16 elements, as shown in Fig. 5.3. The design variables are the vertical positions of the nine control points on the beam's top surface. The control points on the bottom are fixed during optimization, and that on the two sides will be linearly distributed along the y-direction. The constraint is that the beam area should not be beyond $\hat{A} = 220$. The side constraints can be seen in Tab. 5.1. The analysis mesh is refined from the design mesh and has 32 elements (Fig. 5.4). After the iterative procedure (Fig. 5.6), an optimized design is obtained with the final geometry shown in Fig. 5.5. The optimization objective reduces to around 30% meanwhile keeping a smooth geometry and satisfying the constraints.

The final positions of the control points can be seen in Tab. 5.1. Those are sufficient to construct a CAD model of the structure which can be used immediately by the designers, and displayed on the CAD software. By comparing Fig. 5.7 with Fig. 5.8, we can see that the stresses on the beam upper surface become smooth after the shape optimization, which coincides with the prediction of the beam theory. The stress oscillation on the points adjacent to the beam ends is because the the movement of the control points on the beam ends is restricted after reaching the corresponding movement bounds.

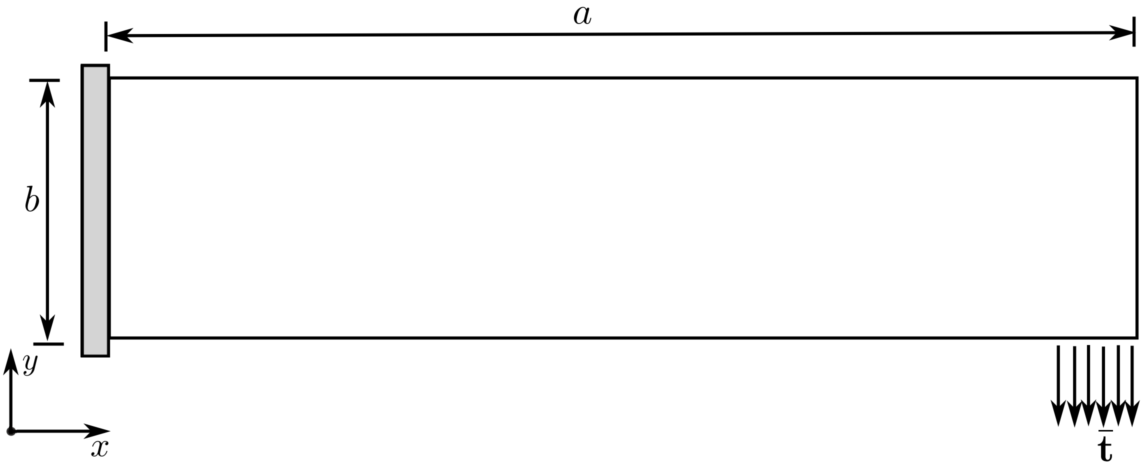


Figure 5.2: The definition of the cantilever beam problem

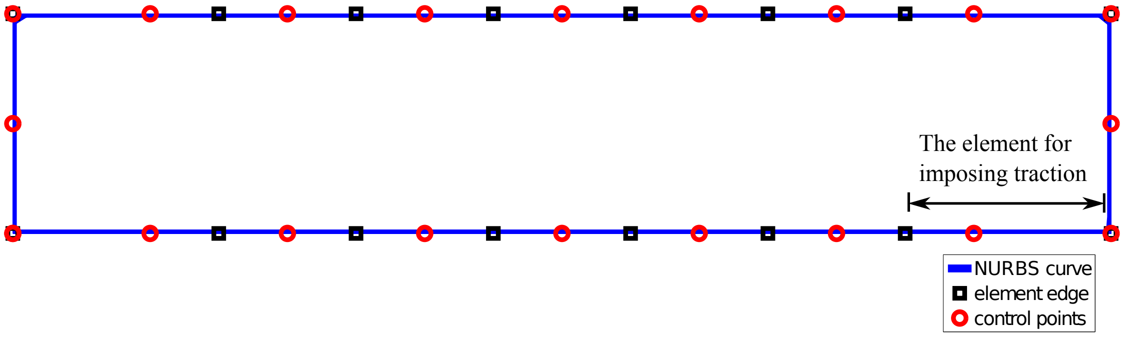


Figure 5.3: The initial design mesh of the cantilever beam

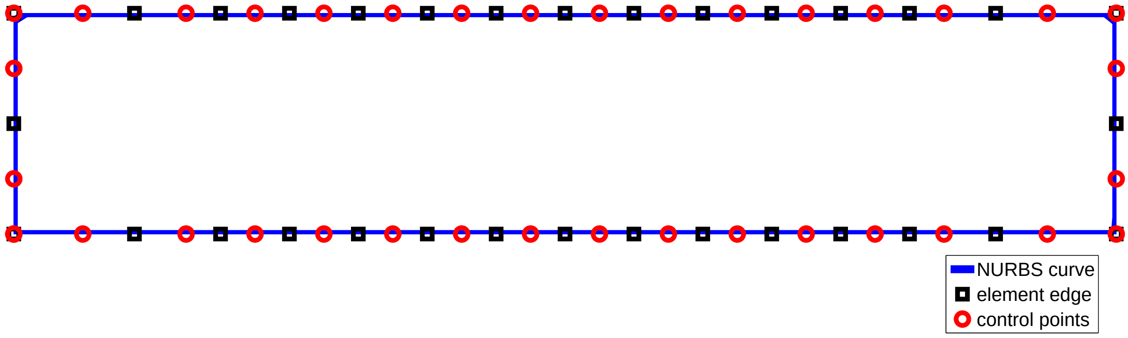


Figure 5.4: The analysis mesh of the cantilever beam

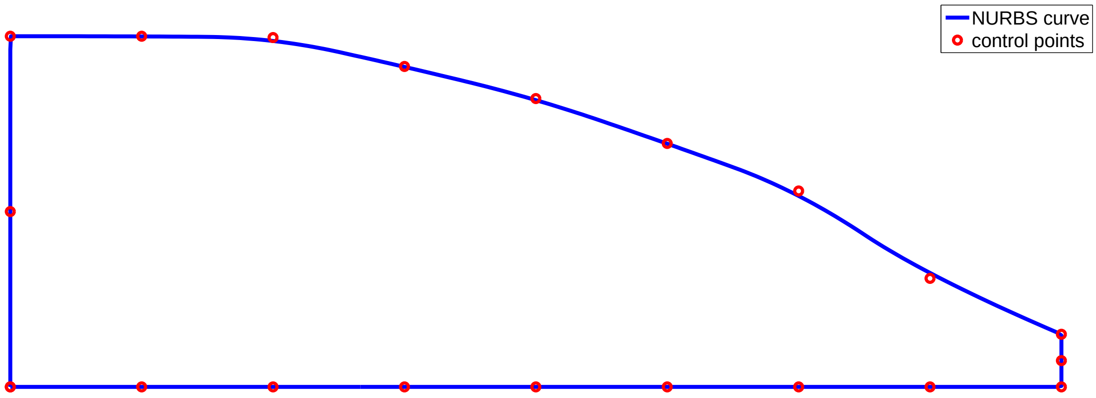


Figure 5.5: The optimized design for the cantilever beam

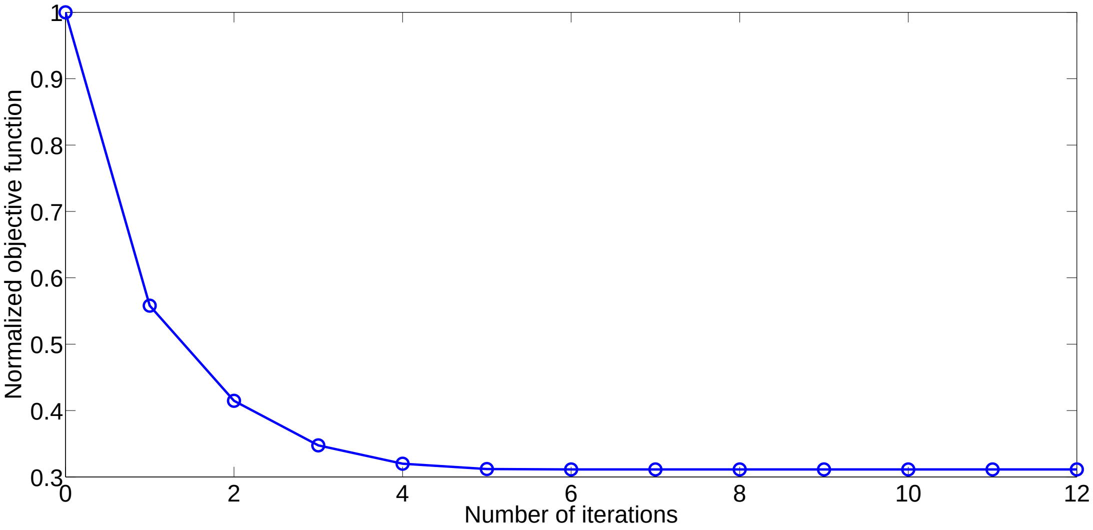


Figure 5.6: The convergence of the iterative process for the cantilever beam optimization

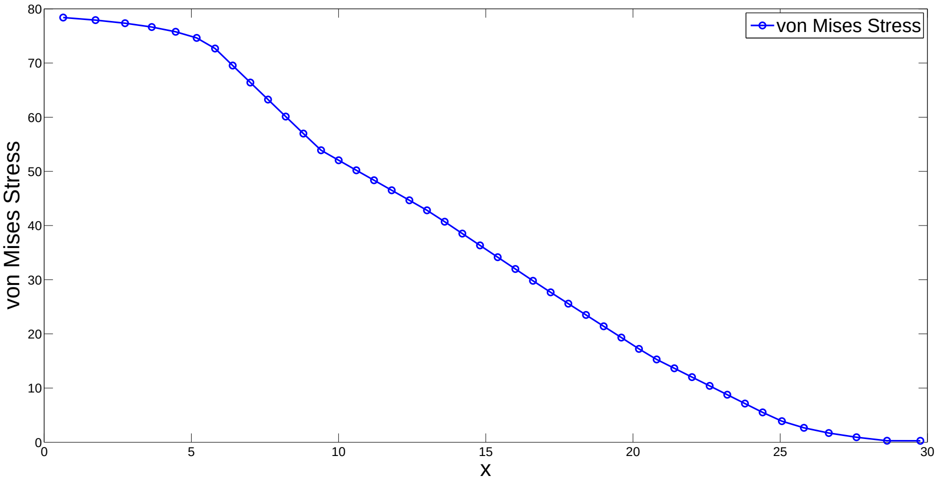


Figure 5.7: Distribution of von Mises stress on the upper surface before optimization

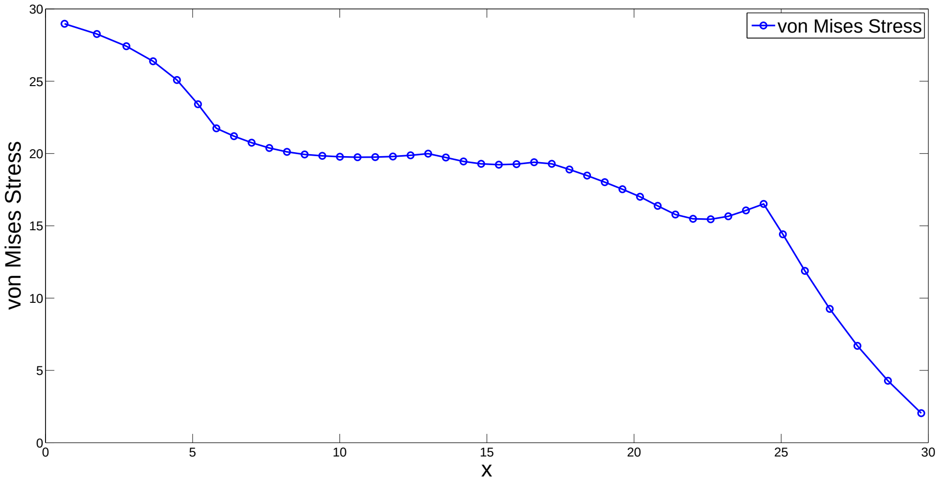


Figure 5.8: Distribution of von Mises stress on the upper surface after optimization

Design variable	Lower bound	Upper bound	Initial value	Final value
t_1	1.5	10	6	1.5001
t_2	1.5	10	6	3.0951
t_3	1.5	10	6	5.5876
t_4	1.5	10	6	6.9434
t_5	1.5	10	6	8.2222
t_6	1.5	10	6	9.1364
t_7	1.5	10	6	9.9619
t_8	1.5	10	6	9.9999
t_9	1.5	10	6	10.0000

Table 5.1: Design variables in the cantilever beam optimization procedure

5.2.2 Fillet

Consider a fillet subject to a traction $\bar{t} = 100$ in the x-direction (see Fig. 5.9). The objective is to minimize its area while keeping the von Mises stress below the allowable value $\hat{\sigma}_{\mathrm{vm}} = 125$. Due to symmetry, only a half model is needed, as shown in Fig. 5.9. The length of the segments are AB = 20, BC = 9, and DE = 9. The Young's modulus is $E = 10^7$, and Poisson's ratio $\nu = 0.3$. The initial positions of the control points of the design model are shown in Fig. 5.10 and the coordinates are given by Tab. A.1 in the appendix. For the shape optimization, the design boundary portion is the curve CD while the vertical positions of the three control points (a, b, c) between CD are set as design variables (Fig. 5.10). The lower and upper bounds for the design variables are 4.5 and 9, respectively. To exert allowable stress constraints, we set a series of monitoring points along CD in the analysis mesh as shown in Fig. 5.11, which is used for structural and sensitivity analysis. The optimized design of the fillet is shown in Fig. 5.12, with the final values of design variables in Tab. 5.2. After the optimization, a smooth stress distribution (Fig. 5.14) on the monitor points is obtained from 5.13. The area is reduced to 138.4132 from 145.1602, and the final design agrees with the reported result using the Boundary Contour Method [78] very well. However, the present method requires no meshing procedure.

Design variable	Lower bound	Upper bound	Initial value	Final value
t_1	0	4	5.625	4.6895
t_2	0	4	6.750	5.1486
t_3	0	4	7.875	6.0814

Table 5.2: Design variables in the fillet optimization procedure

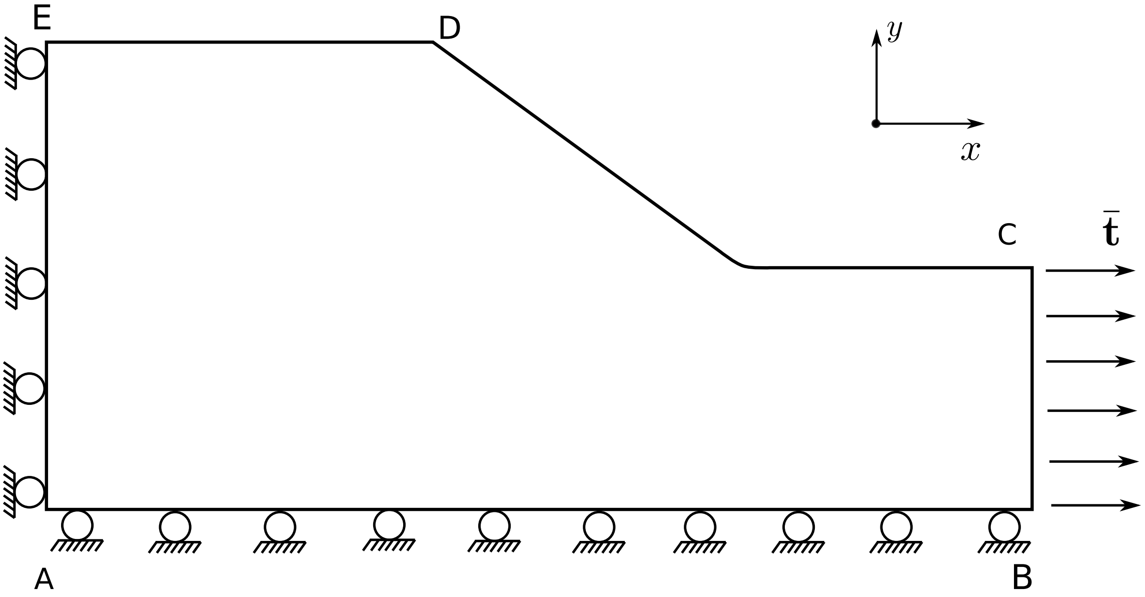


Figure 5.9: The definition of the fillet problem

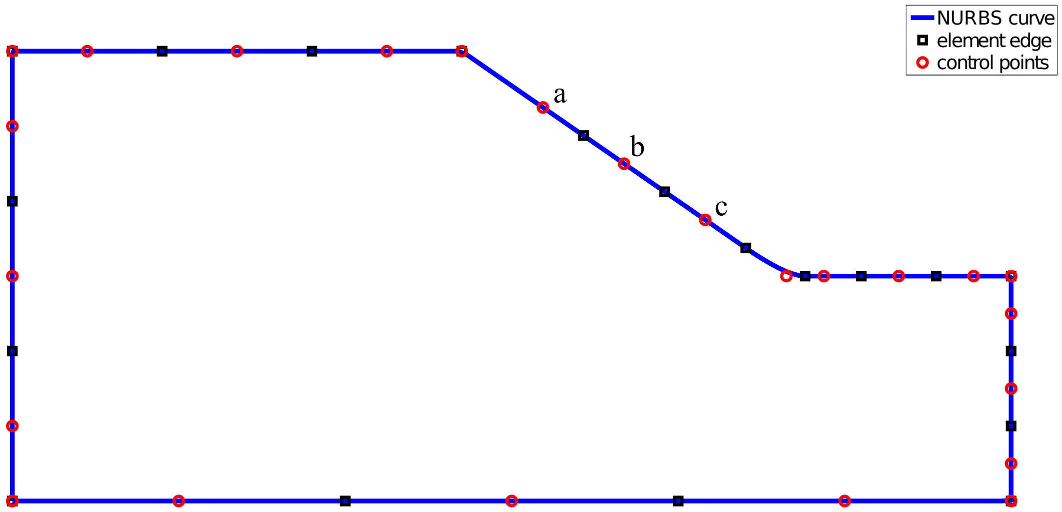


Figure 5.10: The design mesh of the fillet problem

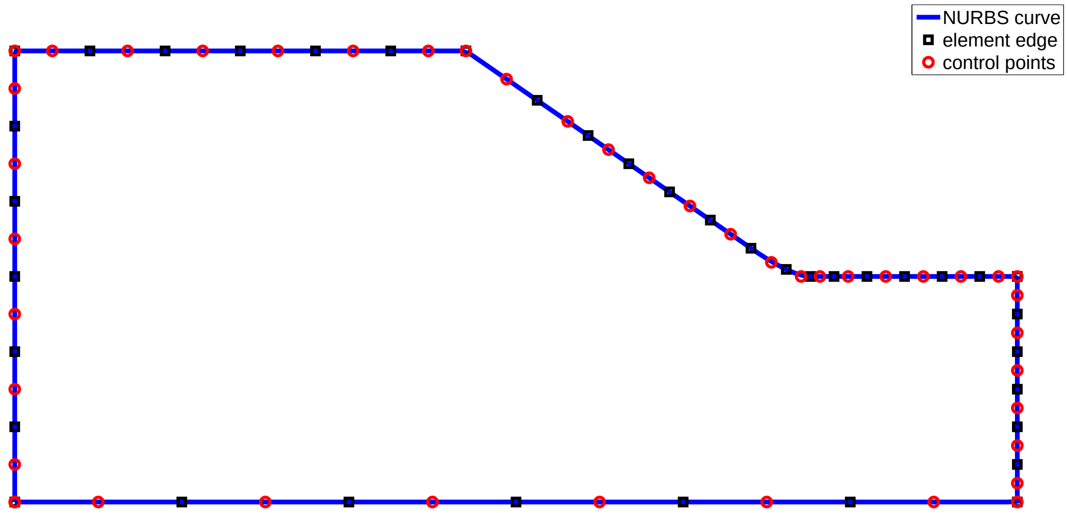


Figure 5.11: The analysis mesh of the fillet problem

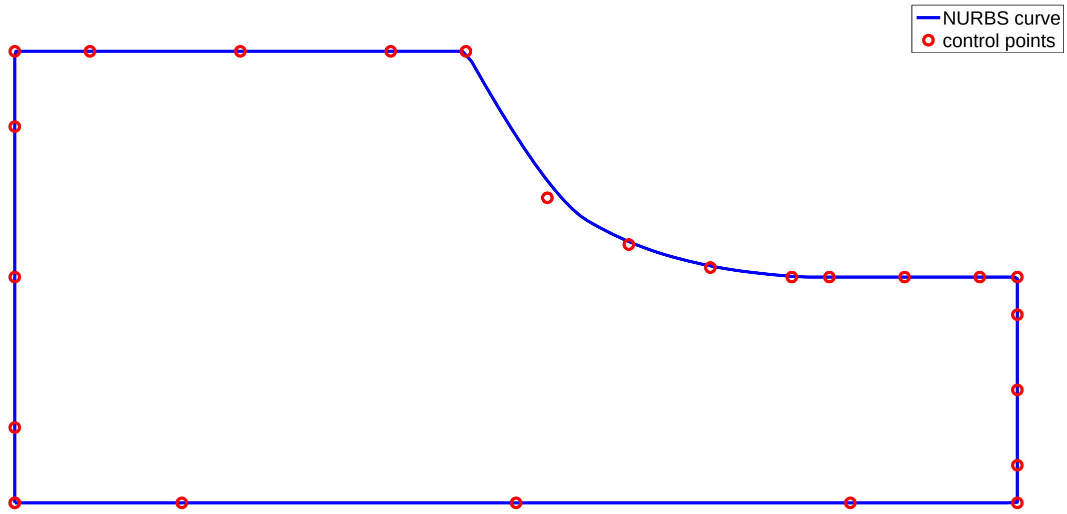


Figure 5.12: The optimized design of the fillet

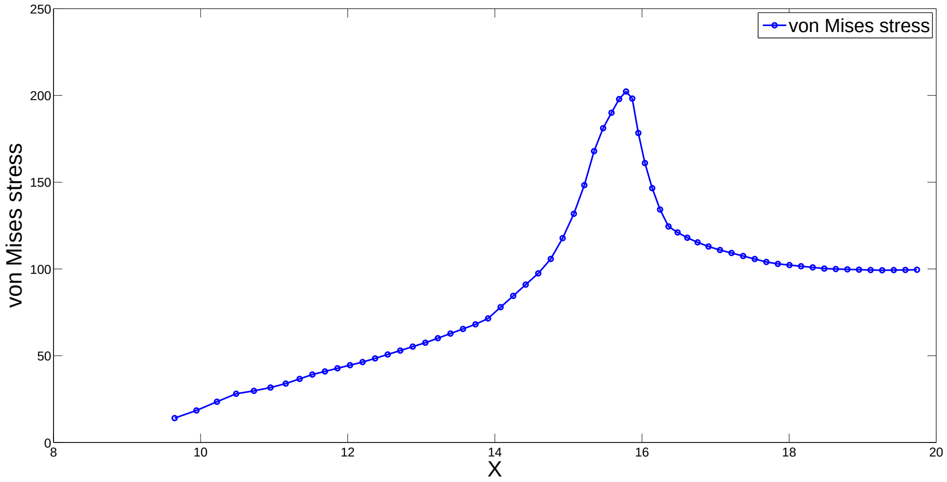


Figure 5.13: Distribution of von Mises stress on the stress monitor points before the optimization (in the curve segment CD)

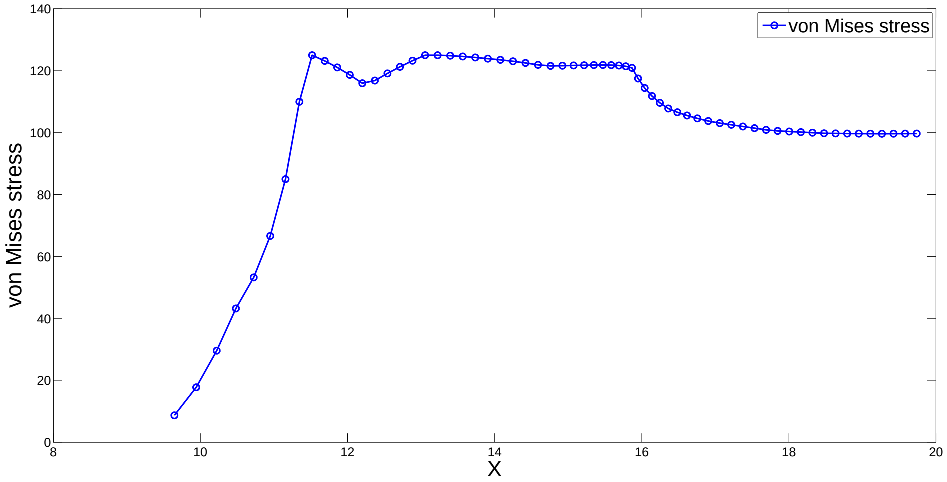


Figure 5.14: Distribution of von Mises stress on the stress monitor points after the optimization (in the curve segment CD)

5.2.3 Connecting rod

The objective is to minimize the area of a connecting rod without violating the maximum von Mises stress constraints. Due to the symmetry, only a half is modelled. The geometry of the initial design and the boundary conditions are shown in Fig. 5.15. The geometry parameters are AB = 110, BC = 90, CD = 10, EF = 9, HA = 15, GE = 30, $a = 45$, $\theta = \pi/4$. The Young's modulus is $E = 10^7$, and Poisson's ratio $\nu = 0.3$. The pressure is $p = 100$ in the normal direction of the half arc. In the structural and shape sensitivity analysis, the traction boundary condition is exerted through the Galerkin nodal parameter extraction method. The initial positions of the control points of the design model are shown in Fig. 5.16 and the coordinates are given by Tab. A.2 in the appendix. The design boundary is the line HG while end points G and H are fixed, and its allowable von Mises stress is $\hat{\sigma}_{\mathrm{vm}} = 600$. The vertical positions of the four control points on the design curve in the design mesh are set as design variables. The lower bound is $[45, 15, 15, 15]$, and the upper bound is $[70, 70, 70, 70]$. The monitoring points are chosen on GH. The mesh for structural and shape sensitivity analysis is shown Fig. (5.17). The optimized geometry is shown in Fig. (5.18), with the coordinates of the converged control points in Tab. 5.3.

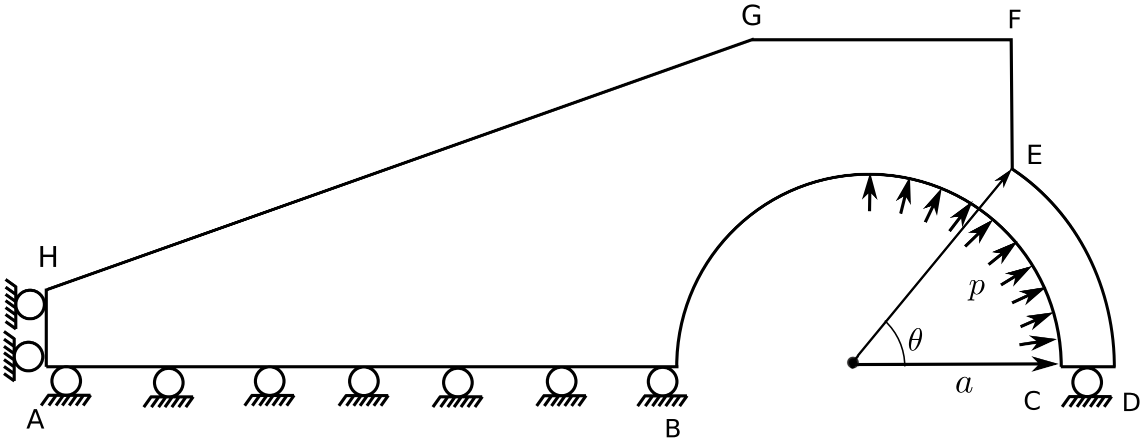


Figure 5.15: The definition of the connecting rod problem

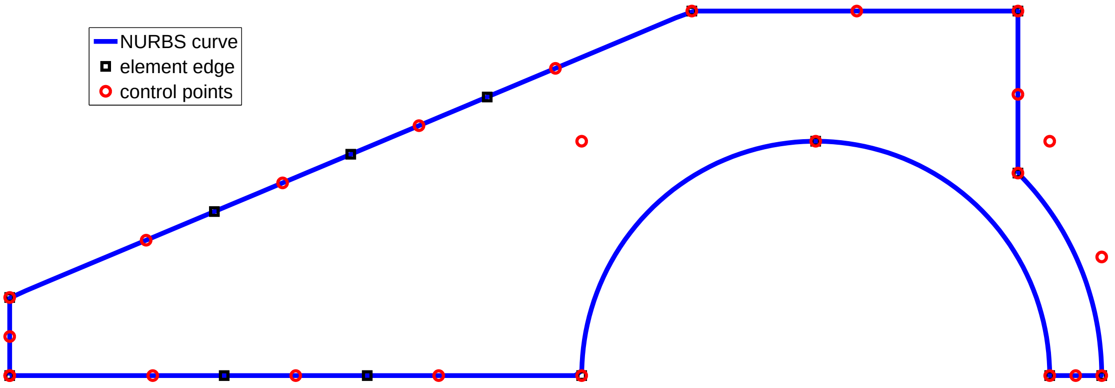


Figure 5.16: The design mesh of the connecting rod problem

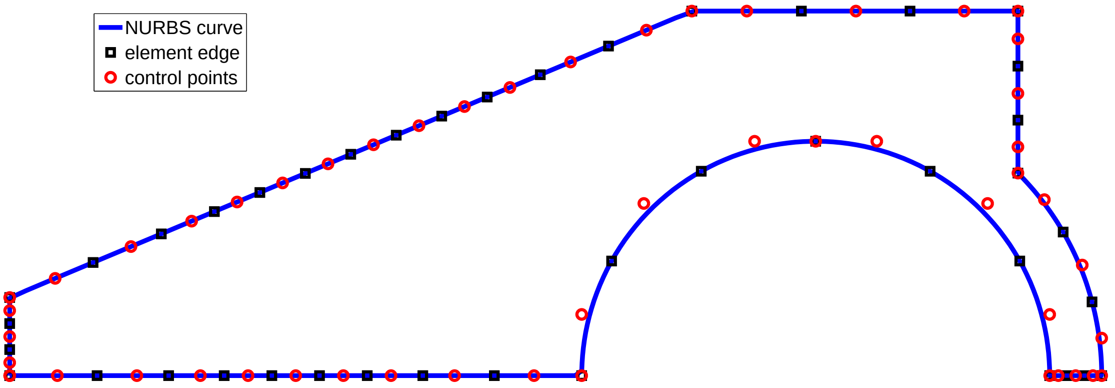


Figure 5.17: The analysis mesh of the connecting rod problem

Design variable	Lower bound	Upper bound	Initial value	Final value
t_1	45	70	59	53.9400
t_2	15	70	48	42.3105
t_3	15	70	37	20.2241
t_4	15	70	26	15.1259

Table 5.3: Design variables in the connecting rod optimization procedure

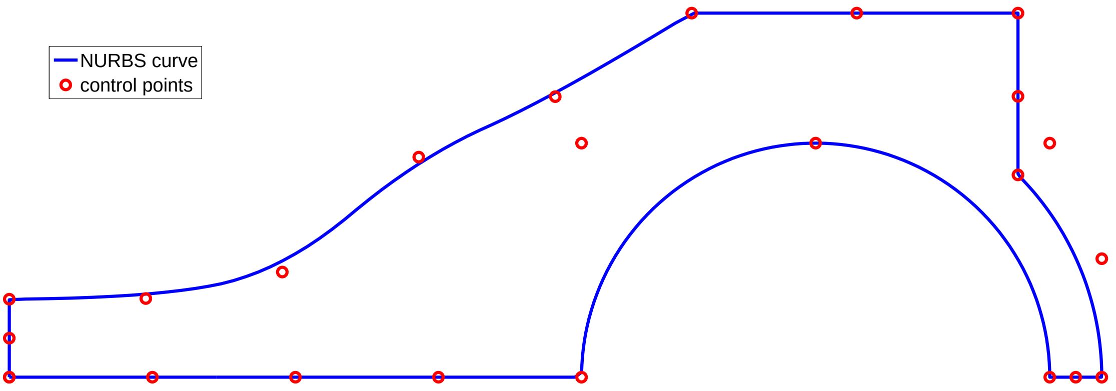


Figure 5.18: The optimized design of the connecting rod

5.2.4 Cantilever beam

Consider a three dimensional cantilever beam, fixed at the left side and subject to to a traction on the bottom. The traction is in the z-direction and linearly distributed along the length as $\bar{t} = -100y$. The material parameters are Young's modulus $E = 10^5$, and Poisson ratio $\nu = 0.3$. The length of the beam is $L = 60$, and the width and height is $h = 20$ (Fig. 5.19). A cubic T-spline model of the cantilever beam is exported from Rhino [97] (Fig. 5.20a) with 336 Bézier elements and 125 control points (Fig. 5.20b).

The objective is to minimize the displacement of the beam's end. The design variables are the control points on the top fibre/surface except that on the left side. The vertical positions of the bottom fibre/surface are fixed during optimization, with that of the remaining control points varying linearly, as Fig. (5.21). The volume constraint is $V \leqslant 26400$. The side constraint is $10 \leqslant z \leqslant 30$. After the iterative process (Fig. 5.23), an optimized geometry and vertical displacement distribution is produced (Fig. 5.22b), against to that of the initial design (Fig. 5.22a). The final positions of the control points can be seen in Tab. 5.4. It is noted that the slight oscillation in the iterative process is due to the violation of the constraints.

Design variable	Lower bound	Upper bound	Initial value	Final value
t_1	10	30	20	30
t_2	10	30	20	27.3548
t_3	10	30	20	19.7754
t_4	10	30	20	13.2369

Table 5.4: Design variables in 3D beam optimization procedure

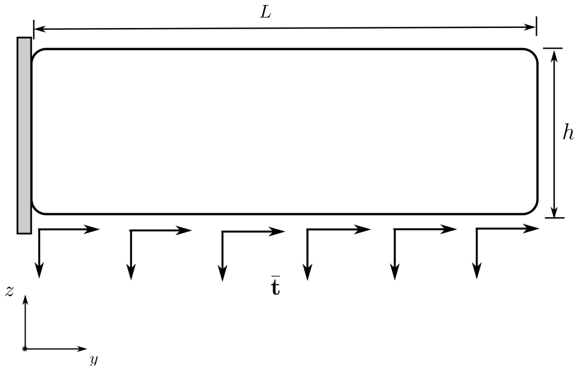
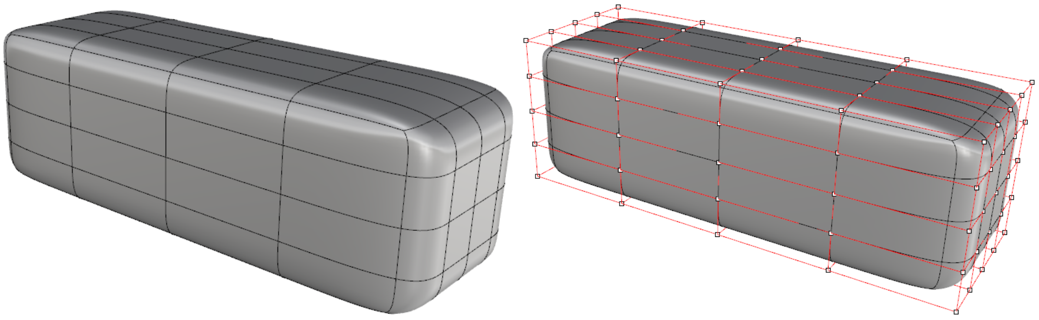


Figure 5.19: The definition of the 3D beam problem


(a)

(b)

Figure 5.20: (a) The geometry of the 3D beam problem, and (b) the control points of the 3D beam

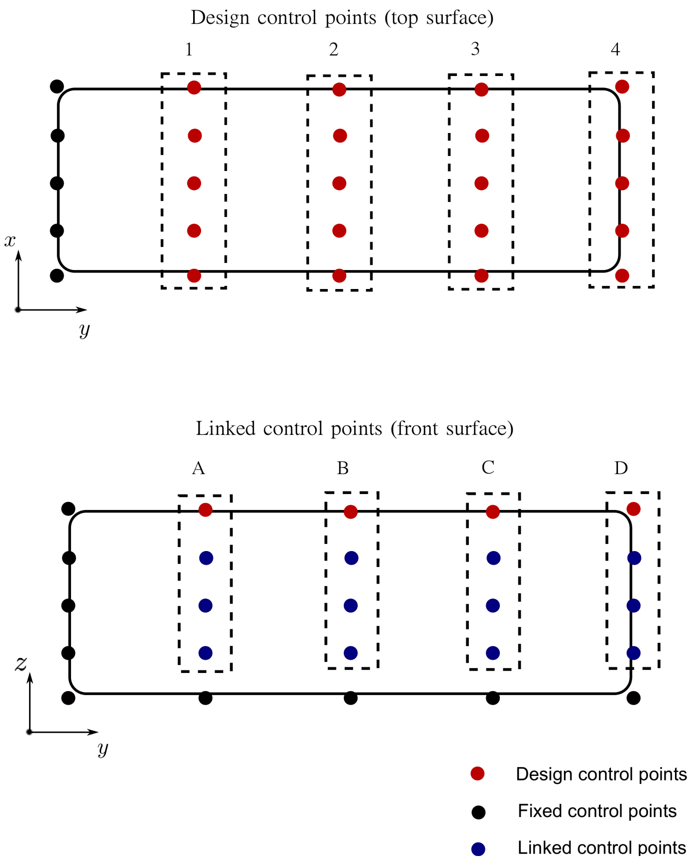
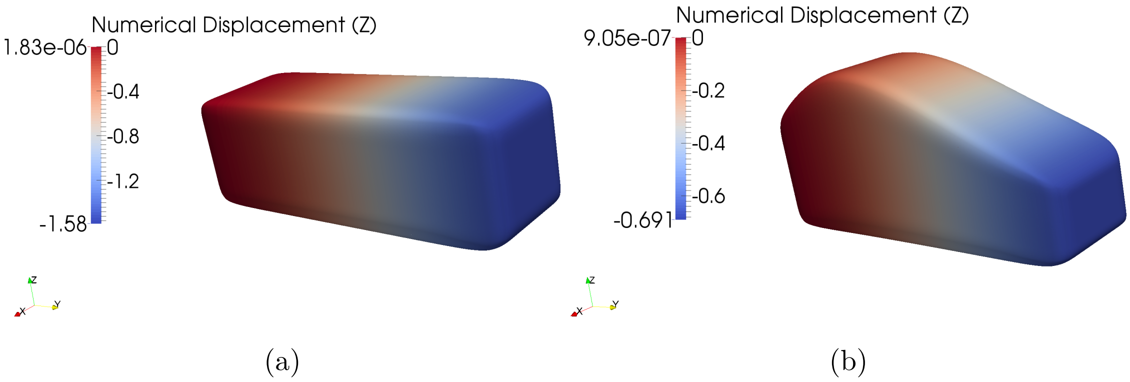


Figure 5.21: The design and linked control points of the 3D beam



(a)

(b)

Figure 5.22: (a) The initial geometry of the 3D beam, and (b) the optimized geometry of the 3D beam

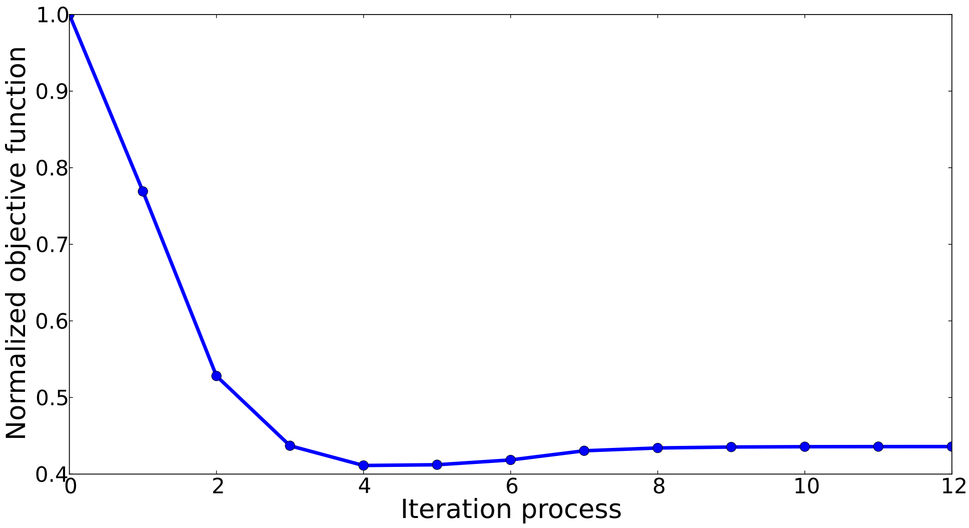


Figure 5.23: The iterative process of the 3D beam optimization

5.2.5 Hammer

The objective is to minimize the conserved energy of a hammer with a volume constraint. The T-spline model of the hammer and the related control points are shown in Figs. 5.24b and 5.25a. The coordinates of the control points are given by Tab. A.3 in the appendix. The hammer is fixed at the bottom, and subject to a uniform traction $\bar{t} = 10^2$ in y-direction on the front (Fig. 5.24a). The Young's modulus is $E = 10^5$, Poisson's ratio ν =0.3. The design control points are shown in Fig. 5.25b and the components in y-direction of the control points A(B), C(D), E(F), G(H), I(J) are set as design variables. The initial values of the design variables are $[2.45, 1.25, 1.33, 1.28, 2.30]$ and the side constraints are $0 \leqslant y \leqslant 4$ for all the control points. The initial volume is $V = 1257.63$ and the volume constraint is $V \leqslant 1307.94$. Fig. 5.27 illustrates the convergence of the iterative process, leading to an optimized geometry as shown in Fig. 5.26b, compared to the initial geometry in Fig. 5.26a. The final values of the design variables can be found in Tab. 5.5. Through the whole optimization procedure, the structural and shape sensitivity analysis can communicate with the the CAD model, and no meshing/remeshing is needed. The final optimized model can be returned directly to the CAD designer without any postprocessing or smoothing procedure.

Design variable	Lower bound	Upper bound	Initial value	Final value
t_1	0	4	2.45	1.8977
t_2	0	4	1.25	1.8353
t_3	0	4	1.33	1.4129
t_4	0	4	1.28	0
t_5	0	4	2.30	0

Table 5.5: Design variables in the hammer optimization procedure

To further test the robustness of the present methodology, we take two other hammer geometries with different initial values for the design parameters. One is with the initial parameters [0.2, 0.2, 0.2, 0.2] and the initial shape is shown in Fig. 5.28a. From Fig. 5.29 we can see that the optimization process initially violates the

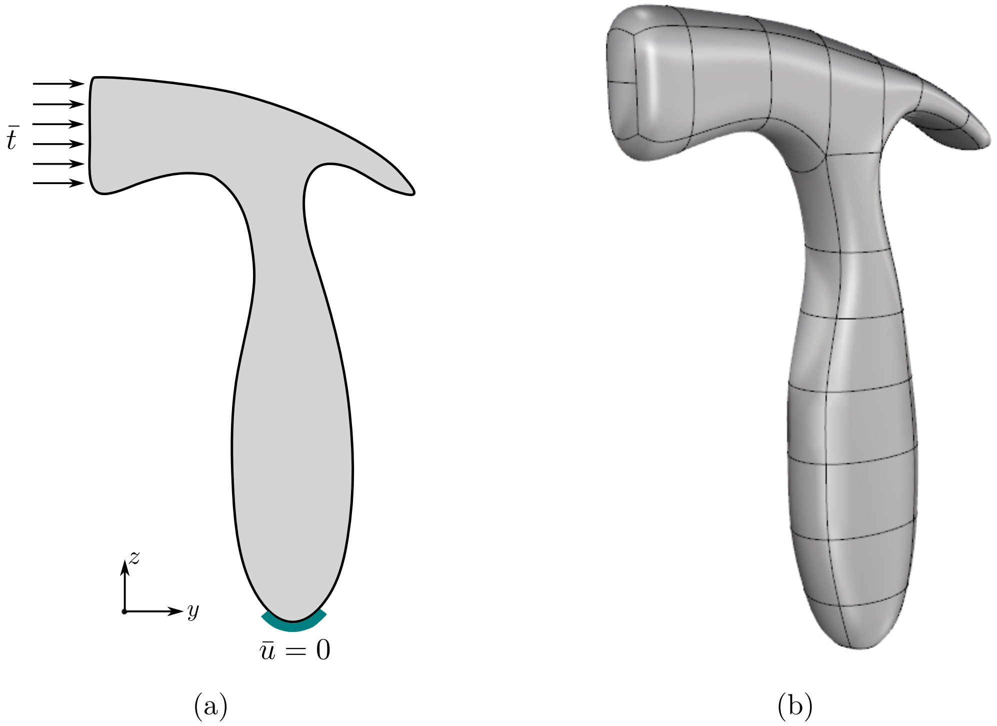


Figure 5.24: (a) Hammer problem definitions, and (b) hammer T-spline model

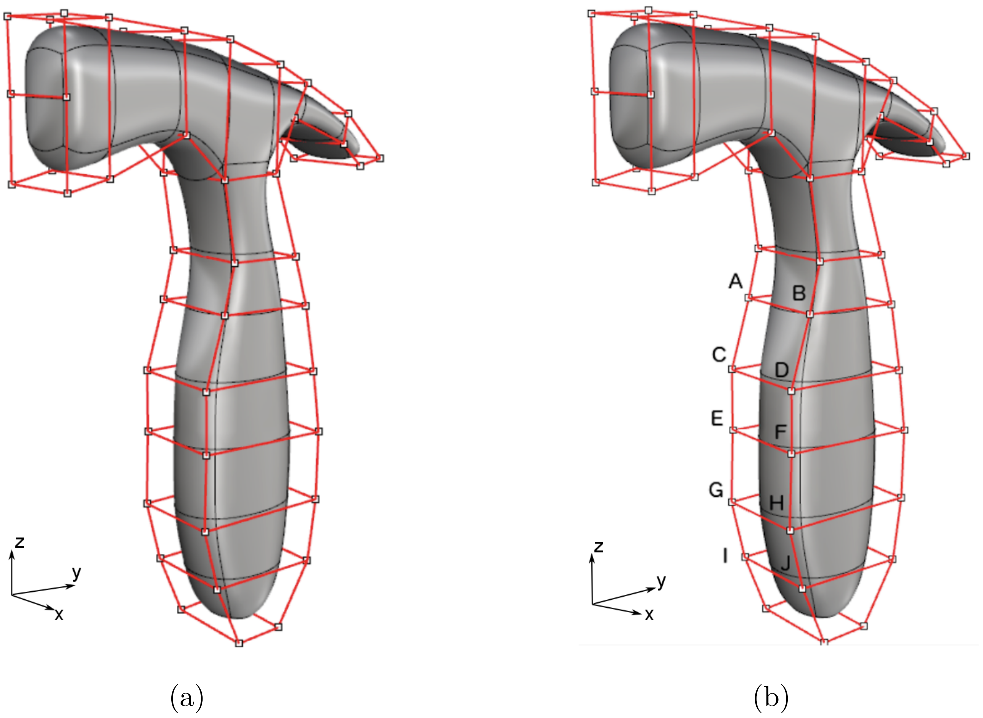
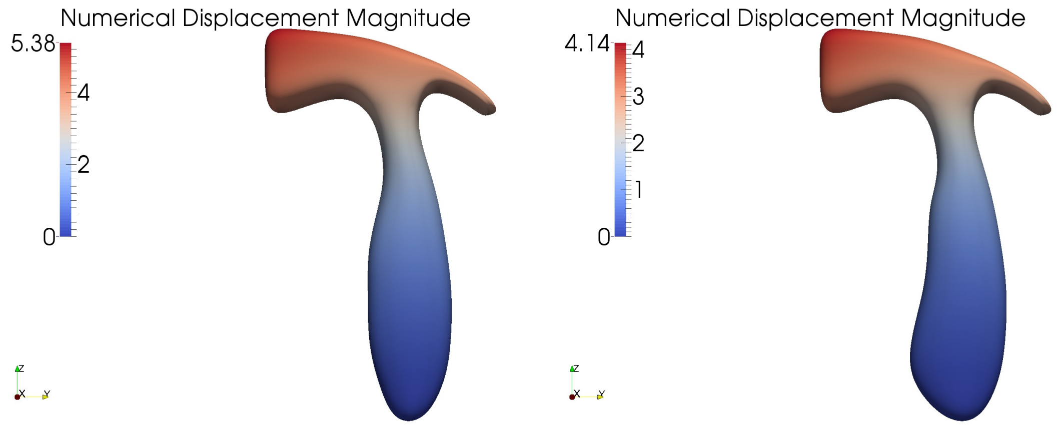


Figure 5.25: (a) Hammer control points, and (b) hammer design points


Numerical Displacement Magnitude
5.38
4
2
0
z
x
y
Numerical Displacement Magnitude
4.14
4
3
2
1
0
z
x
y

(a)

(b)

Figure 5.26: (a) The initial shape of the hammer, and (b) the optimized shape of the hammer

volume constraints, thus leading to an increase of strain energy at the first steps. After locating a feasible region, the strain energy decreases and converges. The other set of parameters is [3.8, 3.8, 3.8, 3.8] with the initial shape as shown in Fig. 5.28b and the iterative process in Fig. 5.30. Both experiments converge to the same result as the initial one, reaching the same value of conserved energy 5916 and design parameters [1.9, 1.8, 1.4, 0.0, 0.0].

Next we choose a T-shape component as the initial geometry of the shape optimization, as shown in Fig. 5.31. The coordinates of the control points are given by Tab. A.4 in the appendix. The objective function, boundary conditions and material parameters are the same as above. The volume constraint is $1537 \leqslant V \leqslant 1564$. As shown in Fig 5.32, the design control points are divided to seven groups, and the design variables are listed as follows,

- t_1, the y-coordinates of the control points A(1, 2), B(1, 2), and C(1, 2).
- t_2, the y-coordinates of the control points D(1, 2) and E(1, 2).
- t_3, the y-coordinates of the control points F(1, 2), G(1, 2), and H(1, 2).
- $t_{4,}$ the y-coordinates of the control points I(1, 2) and J(1, 2).
- t_5, the z-coordinates of the control points K(1, 2), L(1, 2), and M(1, 2).

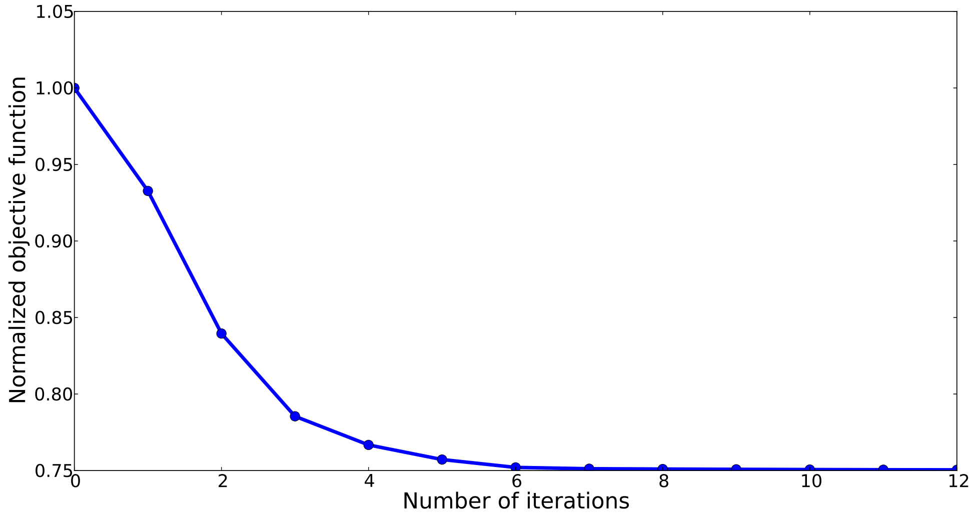


Figure 5.27: The iterative procedure of the hammer optimization

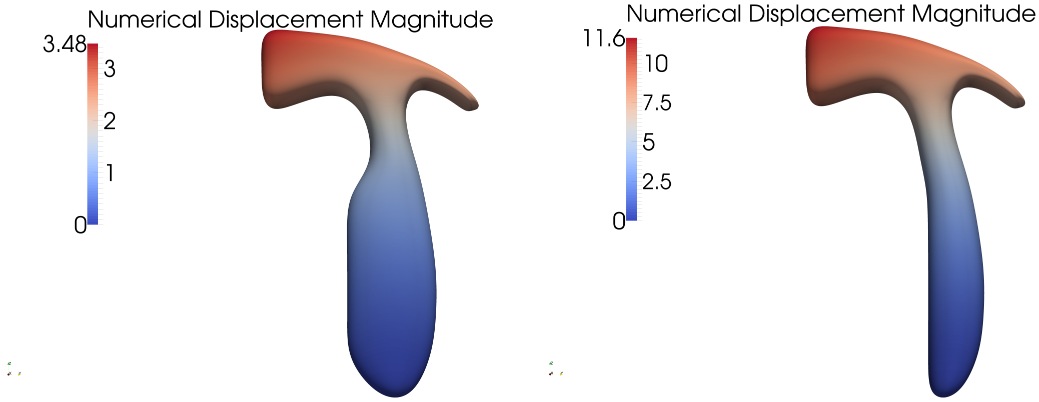
- t_6, the z-coordinates of the control points N(1, 2), O(1, 2), and P(1, 2).

- t_7, the z-coordinates of Q(1, 2).

The initial values of the design variables and the side constraints can be seen in Tab. 5.6. Fig. 5.34 illustrates the convergence of the iterative process and Tab. 5.6 shows the side constraints and the final values of the design variables. It can be observed that, starting from a T-shape geometry, (Fig. 5.33a), the optimization procedure leads to a hammer-shape geometry (Fig. 5.33b).

Design variable	Lower bound	Upper bound	Initial value	Final value
t_1	-2	3.5	1.65	1.05
t_2	-2	3.5	1.4	-1.12
t_3	5.5	9	7.5	7.53
t_4	5.5	9	7.7	9
t_5	10.5	16	13.5	10.5
t_6	3	7.5	5.5	7.5
t_7	2	8	5.4	8

Table 5.6: Design variables in the T-shape component optimization procedure


Numerical Displacement Magnitude
3.48
3
2
1
0
Numerical Displacement Magnitude
11.6
10
7.5
5
2.5
0

(a) (b)

Figure 5.28: (a) The initial shape of the hammer in the second test, and (2) the initial shape of the hammer in the third test

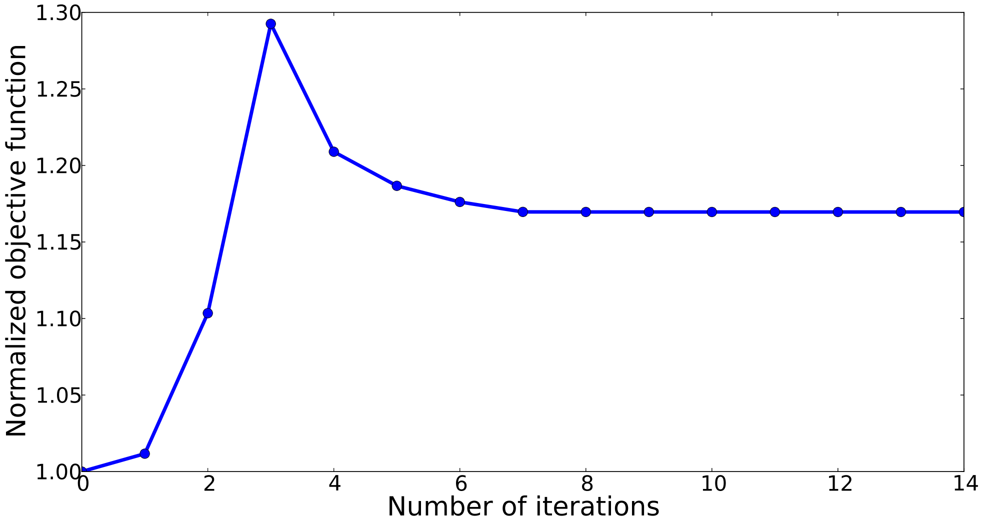


Figure 5.29: The iterative procedure of the hammer optimization (the second test)

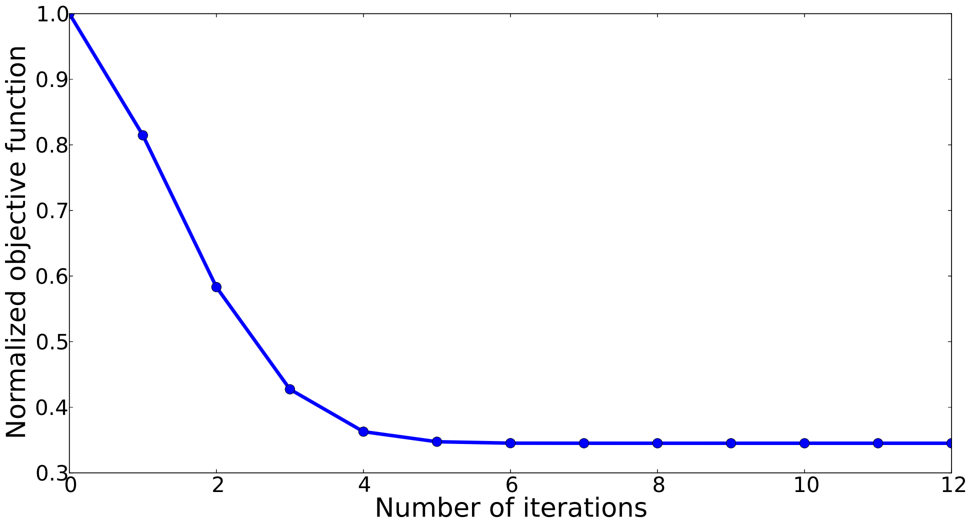


Figure 5.30: The iterative procedure of the hammer optimization (the third test)

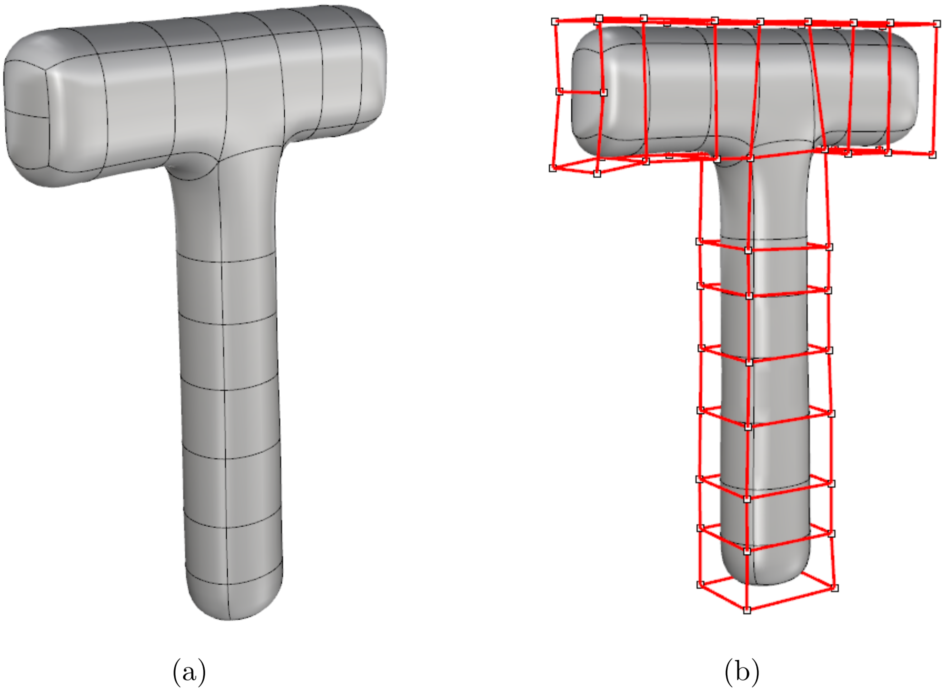


Figure 5.31: (a) T-shape component geometry, and (b) T-shape component control points

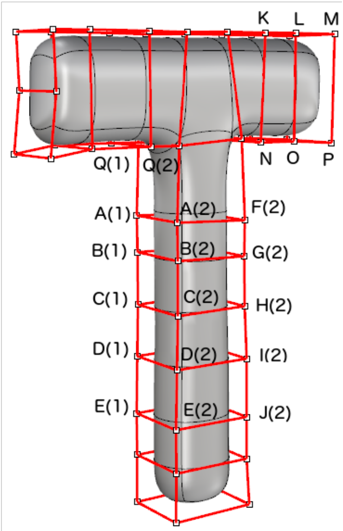
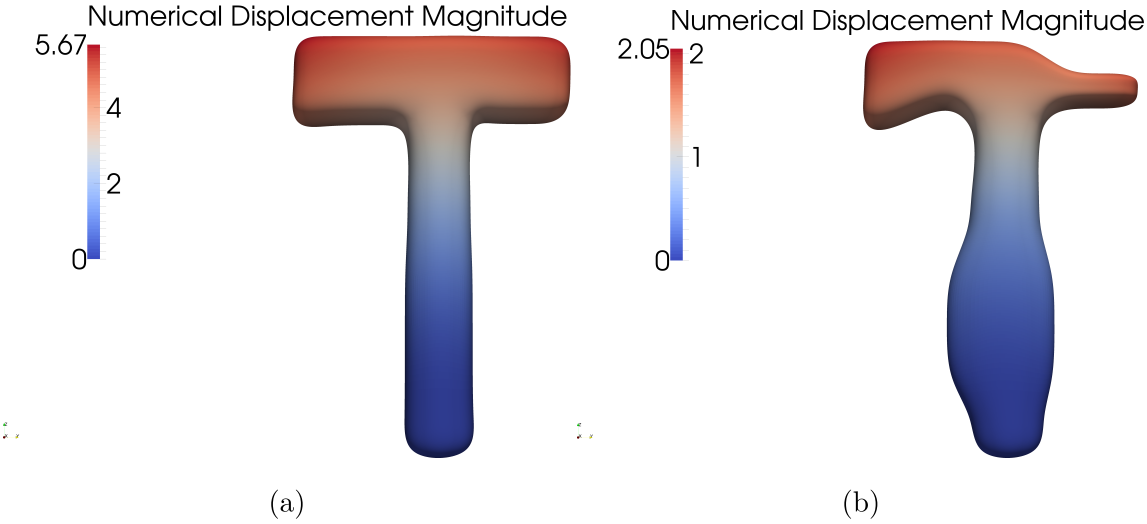


Figure 5.32: The design control points of the T-shape component


Numerical Displacement Magnitude
5.67
4
2
0
Numerical Displacement Magnitude
2.05
2
1
0

(a)

(b)

Figure 5.33: (a) The initial shape of the T-shape component, and (2) the optimized shape of the T-shape component

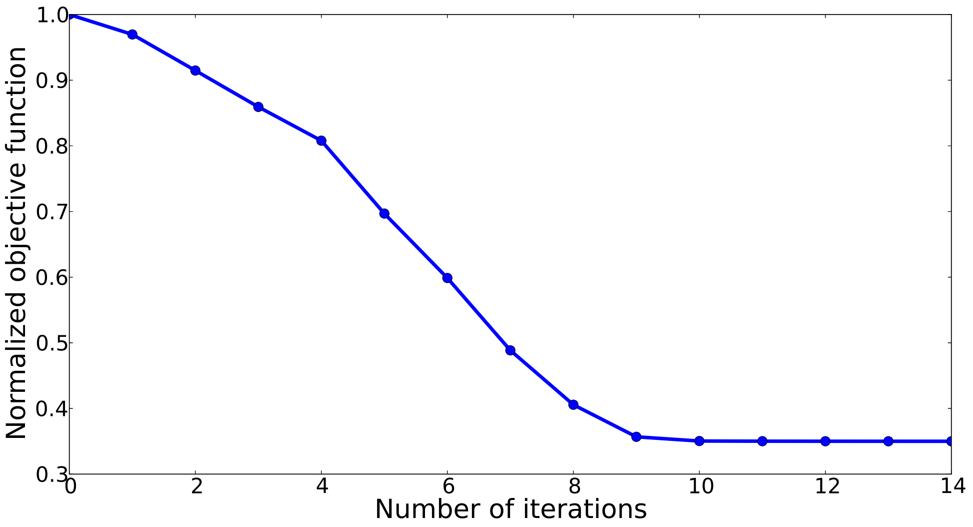


Figure 5.34: The iterative procedure of the T-shape component optimization

5.2.6 Chair

To test the present method on more realistic geometries, consider a chair problem with a watertight geometry constructed by T-splines as given by Fig. 5.35. The original geometry file is sourced from [98] and contains 922 Bézier elements. The coordinates of the control points are given by Tab. A.5 in the appendix. The chair is fixed on the bottom and subject to a uniformly distributed traction with the magnitude of 50 along the opposite z-direction on the face. The Young's modulus is $E = 10^5$, and Poisson's ratio is $\nu = 0.3$. The optimization objective is to minimize the displacement magnitude of the center on the chair face.

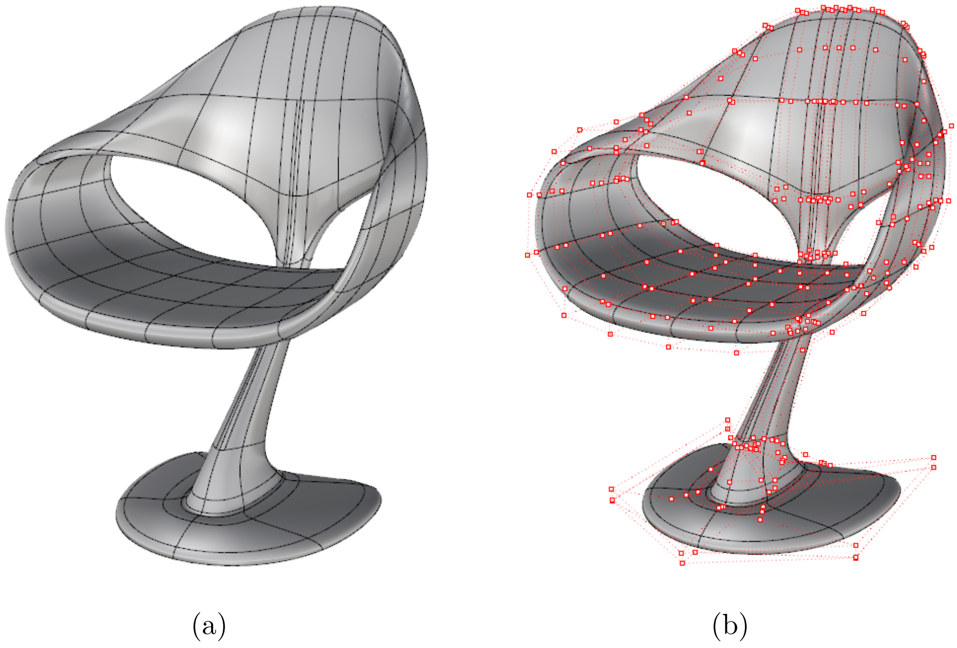


Figure 5.35: (a) Chair geometry, and (b) chair control points

In Fig. 5.36, A, B, and C denote three sets of control points. Each set has five control points which share the same values of y-coordinates. The y-coordinates of the control point set are set as design variables and their initial values and the corresponding side constraints can be found in Tab. 5.7. The initial value of the volume V is 1353.45 and the constraint is $1347 \leqslant V \leqslant 1385$. The initial geometry and the displacement field are shown in Figs. 5.37. The optimized solution reduces the objective function and the final geometry is shown in Fig. 5.38. The change of the design variables can be found in Tab. 5.7 and the iterative procedure is illustrated by Fig. 5.39. The example shows the ability of the present method of optimizing problems with complicated geometries. Throughout the optimization procedure no mesh generation is needed and the optimized geometry remains a CAD model.

Design variable	Lower bound	Upper bound	Initial value	Final value
t_1	8.2	12.5	10.5	9.48
t_2	5.4	10.4	7.5	9.84
t_3	-4.0	-0.5	-2.0	-2.56

Table 5.7: Design variables in the chair optimization procedure

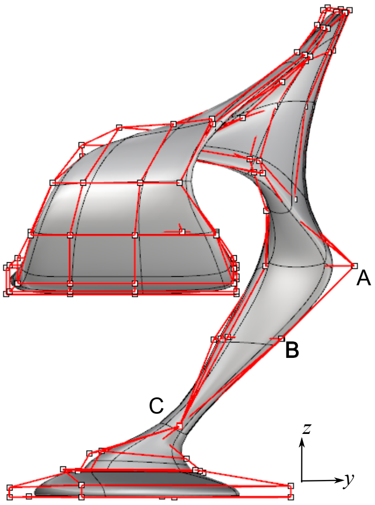


Figure 5.36: The design points of the chair

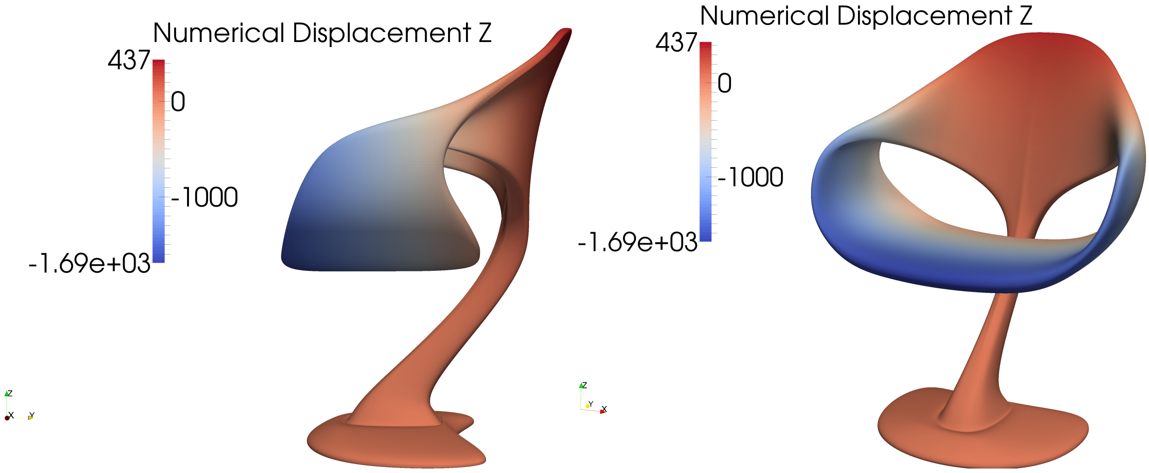


Figure 5.37: The initial shape of the chair

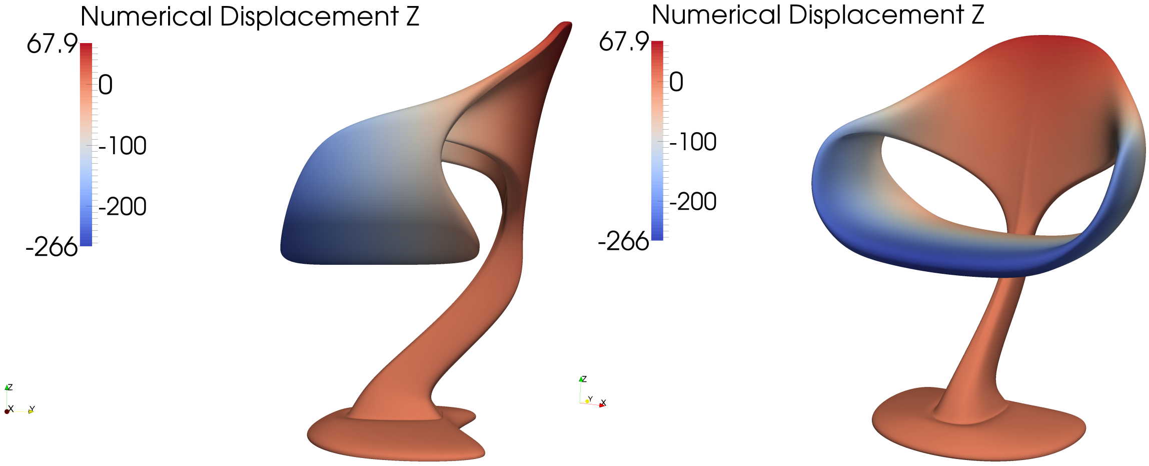


Figure 5.38: The optimal shape of the chair

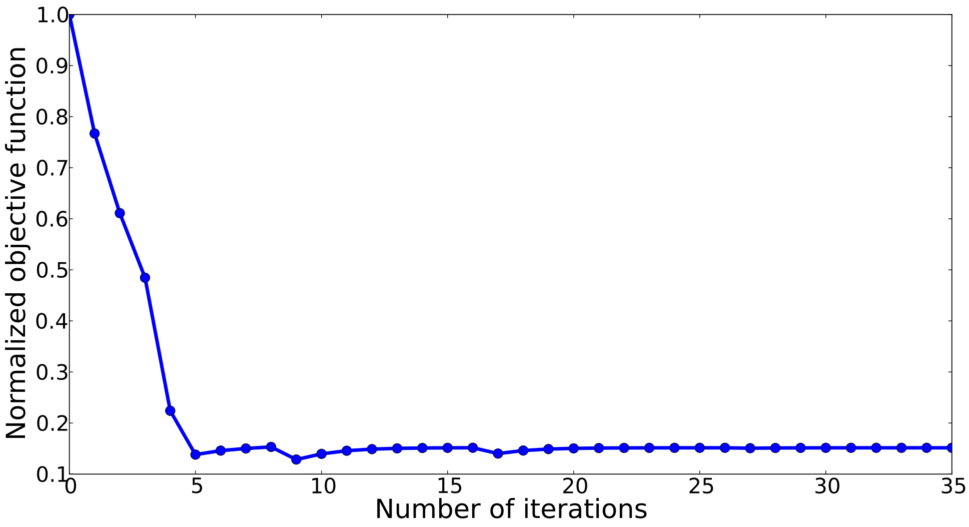


Figure 5.39: The iterative procedure of the chair optimization

5.3 Conclusions

The shape optimization with isogeometric boundary element methods in linear elasticity were formulated. The numerical examples were presented for two-dimensional problems with NURBS geometries, and three-dimensional problems with T-spline geometries. In all the numerical examples, the meshing/remeshing procedures are avoided completely and the optimized geometries are still with the CAD representations. We also adopted a shape sensitivity transition method to separate when needed, the shape sensitivity analysis mesh from the geometrical design mesh.

Chapter 6

Conclusions

An isogeometric boundary element method (IGABEM) was applied to structural shape sensitivity analysis and optimization in two-dimensional and three-dimensional linear elastic problems. The IGABEM adopts the same basis functions as used in the CAD to discretize the boundary integral equation (BIE). Compared with the IGA in the finite element form, the IGABEM is based on the boundary representation and thus compatible with the CAD geometric models. Hence, IGA can achieve a truly integration of analysis and CAD, which property plays a key role in automating and accelerating shape optimization. To facilitate the implementation of IGABEM for its structural and sensitivity analyses, the IGABEM formulation is produced by discretizing the regularized form of BIE, to bypass the difficulties in the evaluation of jump terms and strongly singular integrals. The numerical examples show that the present work possesses the following advantages in shape optimization:

- Meshing/remeshing procedure is completely bypassed, which is a significant improvement in computational efficiency, and eliminates human effort and intervention during the optimization procedure.

- The structural and shape sensitivity analysis is performed on the exact geometry as the CAD model.

- NURBS curves are used for two-dimensional problems, and T-spline surface for three-dimensional problems. Hence, the ability to represent a water-tight and locally-refined geometry can be guaranteed.

- A free-form representation for shape optimization can be naturally achieved, and the control mesh provides an elegant choice of design variables.
- The final optimal model can be returned to the CAD designers directly, so the optimal model will not be perturbed in the "smoothing" step.

The work can be extended in the future as follows:

- A fast algorithm is needed to address the full matrix of isogeometric boundary element method.
- An error estimation can be developed to allow an adaptive refinement scheme.
- Acoustic and electromagnetic problems can be applied to, where the application exhibits the advantage of isogeometric boundary element method in infinite domain problems.

Bibliography

[1] M. H. Aliabadi, D. P. Rooke, and D. J. Cartwright. An improved boundary element formulation for calculating stress intensity factors: Application to aerospace structures. *The Journal of Strain Analysis for Engineering Design*, 22(4):203–207, 1987.

[2] H. Andrä. Integration of singular integrals for the Galerkin-type boundary element method in 3D elasticity. *Computer Methods in Applied Mechanics and Engineering*, 157(3):239–249, 1998.

[3] S. N. Atluri and T. Zhu. A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics*, 22(2):117–127, 1998.

[4] F. Auricchio, L. B. Da Veiga, T. J. R. Hughes, A. Reali, and G. Sangalli. Isogeometric collocation methods. *Mathematical Models and Methods in Applied Sciences*, 20(11):2075–2107, 2010.

[5] H. Azegami and K. Takeuchi. A smoothing method for shape optimization: traction method using the robin condition. *International Journal of Computational Methods*, 3(01):21–33, 2006.

[6] M. R. Barone and R. Yang. A boundary element approach for recovery of share sensitivities in three-dimensional elastic solids. *Computer Methods in Applied Mechanics and Engineering*, 74(1):69–82, 1989.

[7] Y. Bazilevs, V. M. Calo, J. A. Cottrell, J. A. Evans, T. J. R. Hughes, S. Lipton, M. A. Scott, and T. W. Sederberg. Isogeometric analysis using T-splines. *Computers and Mathematics with Applications*, 199(5–8):229–263, 2010.

[8] S. Beissel and T. Belytschko. Nodal integration of the element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 139(1):49–74, 1996.

[9] T. Belytschko and T. Black. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45(5):601–620, 1999.

[10] T. Belytschko, Y. Guo, W. K. Liu, and S. P. Xiao. A unified stability analysis of meshless particle methods. *International Journal for Numerical Methods in Engineering*, 48(9):1359–1400, 2000.

[11] T. Belytschko, Y. Krongauz, J. Dolbow, and C. Gerlach. On the completeness of meshfree particle methods. *International Journal for Numerical Methods in Engineering*, 43(5):785–819, 1998.

[12] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139(1):3–47, 1996.

[13] T. Belytschko, Y. Y. Lu, and L. Gu. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37:229–256, 1994.

[14] M. P. Bendsøe. Optimal shape design as a material distribution problem. *Structural Optimization*, 1(4):193–202, 1989.

[15] M. P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197–224, 1988.

[16] M. P. Bendsoe and O. Sigmund. *Topology optimization: theory, methods and applications.* Springer, 2003.

[17] K.-U. Bletzinger, M. Firl, J. Linhard, and R. Wüchner. Optimal shapes of mechanically motivated surfaces. *Computer Methods in Applied Mechanics and Engineering*, 199(5):324–333, 2010.

[18] F. Bobaru and S. Mukherjee. Shape sensitivity analysis and shape optimization in planar elasticity using the element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 190(32):4319–4337, 2001.

[19] F. Bobaru and S. Mukherjee. Meshless approach to shape optimization of linear thermoelastic solids. *International Journal for Numerical Methods in Engineering*, 53(4):765–796, 2002.

[20] M. Bonnet, G. Maier, and C. Polizzotto. Symmetric Galerkin boundary element method. *Applied Mechanics Reviews*, 51:669–704, 1998.

[21] S. P. A. Bordas and B. Moran. Enriched finite elements and level sets for damage tolerance assessment of complex structures. *Engineering Fracture Mechanics*, 73(9):1176–1201, Jan 2006.

[22] S. P. A. Bordas, P. V. Nguyen, C. Dunant, A. Guidoum, and H. Nguyen-Dang. An extended finite element library. *International Journal for Numerical Methods in Engineering*, 71(6):703–732, Jan 2007.

[23] M. J. Borden, M. A. Scott, J. A. Evans, and T. J. R. Hughes. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering*, 87(1-5):15–47, 2011.

[24] A. Carpinteri, G. Ferro, and G. Ventura. The partition of unity quadrature in element-free crack modelling. *Computers and Structures*, 81(18):1783–1794, 2003.

[25] J.-S. Chen, C.-T. Wu, S. Yoon, and Y. You. A stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 50(2):435–466, 2001.

[26] S. Cho and S. H. Ha. Isogeometric shape design optimization: exact geometry and enhanced sensitivity. *Structural and Multidisciplinary Optimization*, 38(1):53–70, 2009.

[27] J. H. Choi and B. M. Kwak. Boundary integral equation method for shape optimization of elastic structures. *International Journal for Numerical Methods in Engineering*, 26(7):1579–1595, 1988.

[28] K. K. Choi. *Shape design sensitivity analysis and optimal design of structural systems*. Springer, 1987.

[29] K. K. Choi and E. J. Haug. Shape design sensitivity analysis of elastic structures. *Journal of Structural Mechanics*, 11(2):231–269, 1983.

[30] A. N. Christiansen, J. A. Bærentzen, M. Nobel-Jørgensen, N. Aage, and O. Sigmund. Combined shape and topology optimization of 3d structures. *Computers & Graphics*, 46:25–35, 2015.

[31] A. N. Christiansen, M. Nobel-Jørgensen, N. Aage, O. Sigmund, and J. A. Bærentzen. Topology optimization using an explicit interface representation. *Structural and Multidisciplinary Optimization*, 49(3):387–399, 2014.

[32] F. Cirak and M. Ortiz. Fully C1-conforming subdivision elements for finite deformation thin-shell analysis. *International Journal for Numerical Methods in Engineering*, 51(7):813–833, 2001.

[33] F. Cirak, M. Ortiz, and P. Schroder. Subdivision surfaces: A new paradigm for thin-shell finite-element analysis. *International Journal for Numerical Methods in Engineering*, 47(12):2039–2072, 2000.

[34] F. Cirak, M. Scott, E. Antonsson, M. Ortiz, and P. Schröder. Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision. *Computer-Aided Design*, 34(2):137–148, 2002.

[35] A. P. Cisilino, M. H. Aliabadi, and J. L. Otegui. A three-dimensional boundary element formulation for the elastoplastic analysis of cracked bodies. *International Journal for Numerical Methods in Engineering*, 42(2):237–256, 1998.

[36] M. G. Cox. The numerical evaluation of B-splines. *IMA Journal of Applied Mathematics*, 10(2):134–149, 1972.

[37] T. A. Cruse. A direct formulation and numerical solution of the general transient elastodynamic problem. II. *Journal of Mathematical Analysis and Applications*, 22(2):341–355, 1968.

[38] T. A. Cruse. Numerical solutions in three dimensional elastostatics. *International Journal of Solids and Structures*, 5:1259–1274, 1969.

[39] C. De Boor. On calculating with B-splines. *Journal of Approximation Theory*, 6(1):50–62, 1972.

[40] J. Dolbow and T. Belytschko. Numerical integration of the Galerkin weak form in meshfree methods. *Computational Mechanics*, 23(3):219–230, 1999.

[41] M. Dörfel, B. Jüttler, and B. Simeon. Adaptive isogeometric analysis by local h-refinement with T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8):264–275, 2010.

[42] C. A. Duarte and J. T. Oden. *Hp* clouds-an hp meshless method. *Numerical Methods for Partial Differential Equations*, 12(6):673–706, 1996.

[43] M. Duflot and H. Nguyen-Dang. A truly meshless Galerkin method based on a moving least squares quadrature. *Communications in Numerical Methods in Engineering*, 18(6):441–449, 2002.

[44] M. Firl, R. Wüchner, and K.-U. Bletzinger. Regularization of shape optimization problems using fe-based parametrization. *Structural and Multidisciplinary Optimization*, 47(4):507–521, 2013.

[45] R. Glowinski, T. Pan, and J. Periaux. A fictitious domain method for Dirichlet problem and applications. *Computer Methods in Applied Mechanics and Engineering*, 111(3-4):283–303, 1994.

[46] M. Guiggiani and P. Casalini. Direct computation of Cauchy principal value integrals in advanced boundary elements. *International Journal for Numerical Methods in Engineering*, 24(9):1711–1720, 1987.

[47] S.-H. Ha, K. Choi, and S. Cho. Numerical method for shape optimization using T-spline based isogeometric method. *Structural and Multidisciplinary Optimization*, 42(3):417–428, 2010.

[48] R. T. Haftka and Z. Gürdal. *Elements of structural optimization*, volume 11. Springer Science & Business Media, 1992.

[49] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39–41):4135–4195, 2005.

[50] M. A. Jaswon. Integral equation methods in potential theory. I. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 275(1360):23–32, 1963.

[51] H. Johansen and P. Colella. A Cartesian grid embedded boundary method for Poisson's equation on irregular domains. *Journal of Computational Physics*, 147(1):60–85, 1998.

[52] P. Kagan, A. Fischer, and P. Z. Bar-Yoseph. New B-spline finite element approach for geometrical design and mechanical analysis. *International Journal for Numerical Methods in Engineering*, 41:435–458, 1998.

[53] J. H. Kane and S. Saigai. Design sensitivity analysis of solids using BEM. *Journal of Engineering Mechanics*, 114(10):1703–1722, 1988.

[54] B. Karihaloo and Q. Xiao. Direct evaluation of accurate SIF with PUM. *Advances in fracture research (Proc ICF 10). Pergamon*, page 0177OR, 2001.

[55] V. Komkov, K. K. Choi, and E. J. Haug. *Design Sensitivity Analysis of Structural Systems*, volume 177. Academic press, 1986.

[56] K. Kostas, A. Ginnis, C. Politis, and P. Kaklis. Ship-hull shape optimization with a t-spline based bem–isogeometric solver. *Computer Methods in Applied Mechanics and Engineering*, 284:611–622, 2015.

[57] C. Le, T. Bruns, and D. Tortorelli. A gradient-based, parameter-free approach to shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 200(9):985–996, 2011.

[58] G. Legrain. A NURBS enhanced extended finite element approach for unfitted CAD analysis. *Computational Mechanics*, 52(4):913–929, 2013.

[59] V. Leitao, M. H. Aliabadi, and D. P. Rooke. The dual boundary element formulation for elastoplastic fracture mechanics. *International journal for numerical methods in engineering*, 38(2):315–333, 1995.

[60] K. Li and X. Qian. Isogeometric analysis and shape optimization via boundary integral. *Computer-Aided Design*, 43(11):1427–1437, 2011.

[61] X. Li, J. Zheng, T. W. Sederberg, T. J. R. Hughes, and M. A. Scott. On linear independence of T-spline blending functions. *Computer Aided Geometric Design*, 29:63 – 76, 2012.

[62] L. Liu, Y. Zhang, T. J. Hughes, M. A. Scott, and T. W. Sederberg. Volumetric T-spline construction using boolean operations. In *Proceedings of the 22nd International Meshing Roundtable*, pages 405–424. Springer, 2014.

[63] W. Liu, S. Jun, and Y. Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Engineering*, 20:1081–1106, 1995.

[64] X. Liu, Q. Xiao, and B. Karihaloo. XFEM for direct evaluation of mixed mode sifs in homogeneous and bi-materials. *International Journal for Numerical Methods in Engineering*, 59(8):1103–1118, 2004.

[65] Y. Liu. On the simple-solution method and non-singular nature of the BIE/BEM–a review and some new results. *Engineering Analysis with Boundary Elements*, 24(10):789–795, 2000.

[66] Y. Liu and T. J. Rudolphi. Some identities for fundamental solutions and their applications to weakly-singular boundary element formulations. *Engineering Analysis with Boundary Elements*, 8(6):301–311, 1991.

[67] Y. Liu and T. J. Rudolphi. New identities for fundamental solutions and their applications to non-singular boundary element formulations. *Computational Mechanics*, 24(4):286–292, 1999.

[68] A. Longo, J. Unzueta, E. Schaeidt, A. Alvarez, and J. J. Anza. A general related variational approach to shape optimum design. *Advances in Engineering Software*, 16(2):135–142, 1993.

[69] N. D. Manh, A. Evgrafov, A. R. Gersborg, and J. Gravesen. Isogeometric shape optimization of vibrating membranes. *Computer Methods in Applied Mechanics and Engineering*, 200(13):1343–1353, 2011.

[70] J. M. Melenk and I. Babuška. The partition of unity finite element method: basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1):289–314, 1996.

[71] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37:239–261, 2005.

[72] N. Moës, M. Cloirec, P. Cartraud, and J.-F. Remacle. A computational approach to handle complex microstructure geometries. *Computer Methods in Applied Mechanics and Engineering*, 192(28):3163–3177, 2003.

[73] M. Moumnassi, S. Belouettar, É. Béchet, S. Bordas, D. Quoirin, and M. Potier-Ferry. Finite element analysis on implicitly defined domains: An accurate representation based on arbitrary parametric surfaces. *Computer Methods in Applied Mechanics and Engineering*, 200(5):774–796, 2011.

[74] V. Nguyen, T. Rabczuk, S. Bordas, and M. Duflot. Meshless methods: A review and computer implementation aspects. *Mathematics and Computers in Simulation*, 79(3):763–813, 2008.

[75] N. Nguyen-Thanh, J. Kiendl, H. Nguyen-Xuan, R. Wüchner, K. Bletzinger, Y. Bazilevs, and T. Rabczuk. Rotation free isogeometric thin shell analysis using pht-splines. *Computer Methods in Applied Mechanics and Engineering*, 200(47):3410–3424, 2011.

[76] N. Nguyen-Thanh, H. Nguyen-Xuan, S. P. A. Bordas, and T. Rabczuk. Isogeometric analysis using polynomial splines over hierarchical t-meshes for two-dimensional elastic solids. *Computer Methods in Applied Mechanics and Engineering*, 200(21):1892–1908, 2011.

[77] J. Nitsche. Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind. In *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*, volume 36, pages 9–15. Springer, 1971.

[78] A.-V. Phan, S. Mukherjee, and J. R. Mayer. Stresses, stress sensitivities and shape optimization in two-dimensional linear elasticity by the boundary contour method. *International Journal for Numerical Methods in Engineering*, 42(8):1391–1407, 1998.

[79] L. Piegl and W. Tiller. *The NURBS Book*. Springer-Verlag, New York, 1997.

[80] C. Politis, A. I. Ginnis, P. D. Kaklis, K. Belibassakis, and C. Feurer. An isogeometric BEM for exterior potential-flow problems in the plane. In *Proceedings of SIAM/ACM joint conference on geometric and physical modeling*, 2009.

[81] X. Qian. Full analytical sensitivities in NURBS based isogeometric shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 199(29-32):2059–2071, 2010.

[82] P. Randles and L. Libersky. Normalized sph with stress points. *International Journal for Numerical Methods in Engineering*, 48(10):1445–1462, 2000.

[83] F. J. Rizzo. An integral equation approach to boundary value problems of classical elastostatics. *Quarterly of Applied Mathematics*, 25(1):83–95, 1967.

[84] A. Rosolen, D. Millán, and M. Arroyo. On the optimum support size in meshfree methods: A variational adaptivity approach with maximum-entropy approximants. *International Journal for Numerical Methods in Engineering*, 82(7):868–895, 2010.

[85] E. Saiki and S. Biringen. Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method. *Journal of Computational Physics*, 123(2):450–465, 1996.

[86] M. A. Scott, M. J. Borden, C. V. Verhoosel, T. W. Sederberg, and T. J. Hughes. Isogeometric finite element data structures based on Bézier extraction of T-splines. *International Journal for Numerical Methods in Engineering*, 88(2):126–156, 2011.

[87] M. A. Scott, R. N. Simpson, J. A. Evans, S. Lipton, S. P. A. Bordas, T. J. R. Hughes, and T. W. Sederberg. Isogeometric boundary element analysis using unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering*, 254:197–221, 2013.

[88] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCs. *ACM Transactions on Graphics*, 22:477–484, July 2003.

[89] R. Sevilla, S. Fernández-Méndez, and A. Huerta. NURBS-enhanced finite element method (NEFEM). *International Journal for Numerical Methods in Engineering*, 76(1):56–83, 2008.

[90] R. Sevilla, S. Fernández-Méndez, and A. Huerta. NURBS-enhanced finite element method (NEFEM): a seamless bridge between CAD and FEM. *Archives of Computational Methods in Engineering*, 18(4):441–484, 2011.

[91] O. Sigmund. A 99 line topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization*, 21(2):120–127, 2001.

[92] O. Sigmund and J. Petersson. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization*, 16(1):68–75, 1998.

[93] R. N. Simpson, S. P. A. Bordas, J. Trevelyan, and T. Rabczuk. A two-dimensional isogeometric boundary element method for elastostatic analysis. *Computer Methods in Applied Mechanics and Engineering*, 209-212:87–100, 2012.

[94] R. N. Simpson, M. A. Scott, M. Taus, D. C. Thomas, and H. Lian. Acoustic isogeometric boundary element analysis. *Computer Methods in Applied Mechanics and Engineering*, 269:265–290, 2014.

[95] K. Svanberg. The method of moving asymptotes-a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373, 1987.

[96] G. T. Symm. Integral equation methods in potential theory. II. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 275(1360):33–46, 1963.

[97] T-Splines, Inc. http://www.tsplines.com/rhino, 2011.

[98] T-Splines, Inc. http://www.tsplines.com/tsplinemodels, 2011.

[99] J. C. F. Telles. A self-adaptive co-ordinate transformation for efficient numerical evaluation of general boundary element integrals. *International Journal for Numerical Methods in Engineering*, 24(5):959–973, 1987.

[100] L. Van Miegroet and P. Duysinx. Stress concentration minimization of 2D filets using X-FEM and level set description. *Structural and Multidisciplinary Optimization*, 33(4-5):425–438, 2007.

[101] W. A. Wall, M. A. Frenzel, and C. Cyron. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197(33):2976–2988, 2008.

[102] P. Wang, J. Xu, J. Deng, and F. Chen. Adaptive isogeometric analysis using rational PHT-splines. *Computer-Aided Design*, 43(11):1438–1448, 2011.

[103] Q. Xiao and B. Karihaloo. Direct evaluation of accurate coefficients of the linear elastic crack tip asymptotic field. *Fatigue and Fracture of Engineering Materials and Structures*, 26(8):719–729, 2003.

[104] Q. Xiao and B. Karihaloo. Implementation of hybrid crack element on a general finite element mesh and in combination with XFEM. *Computer Methods in Applied Mechanics and Engineering*, 196(13):1864–1873, 2007.

[105] W. Xin, A. Chandra, L. Liang-Jenq, and S. Mukherjee. Shape optimization in elasticity and elasto-viscoplasticity by the boundary element method. *International Journal of Solids and Structures*, 31(4):533–550, 1994.

[106] G. Xu, E. Atroshchenko, and S. P. A. Bordas. Geometry independent field approximation for spline-based finite element methods. Barcelona, 2014. 11^{th} World Congress on Computational Mechanics.

[107] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Parameterization of computational domain in isogeometric analysis: methods and comparison. *Computer Methods in Applied Mechanics and Engineering*, 200(23):2021–2031, 2011.

[108] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications. *Computer-Aided Design*, 45(2):395–404, 2013.

[109] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Constructing analysis-suitable parameterization of computational domain from CAD boundary by

variational harmonic method. *Journal of Computational Physics*, 252:275–289, 2013.

[110] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Optimal analysis-aware parameterization of computational domain in 3D isogeometric analysis. *Computer-Aided Design*, 45(4):812–821, 2013.

[111] K. Yamazaki, J. Sakamoto, and M. Kitano. Three-dimensional shape optimization using the boundary element method. *AIAA journal*, 32(6):1295–1301, 1994.

[112] R. Yang. Component shape optimization using BEM. *Computers and Structures*, 37(4):561–568, 1990.

[113] T. Ye, R. Mittal, H. S. Udaykumar, and W. Shyy. An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics*, 156(2):209–240, 1999.

[114] Z. Q. Zhang, J. X. Zhou, N. Zhou, X. M. Wang, and L. Zhang. Shape optimization using reproducing kernel particle method and an enriched genetic algorithm. *Computer Methods in Applied Mechanics and Engineering*, 194(39):4048–4070, 2005.

[115] O. C. Zienkiewicz and J. S. Campbell. Shape optimization and sequential linear programming. *Optimum Structural Design*, pages 109–126, 1973.

[116] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method.* McGraw-hill London, 1977.

Appendix A

The control point coordinates of the geometries

Index	x	y	weight	Index	x	y	weight
1	0	0	1	13	15.5	4.5	1
2	3.3333	0	1	14	13.875	5.625	1
3	10	0	1	15	12.25	6.75	1
4	16.667	0	1	16	10.625	7.875	1
5	20	0	1	17	9	9	1
6	20	0.75	1	18	7.5	9	1
7	20	2.25	1	19	4.5	9	1
8	20	3.75	1	20	1.5	9	1
9	20	4.5	1	21	0	9	1
10	19.25	4.5	1	22	0	7.5	1
11	17.75	4.5	1	23	0	4.5	1
12	16.25	4.5	1	24	0	1.5	1

Table A.1: The control points of the initial geometry of the fillet

Index	x	y	weight	Index	x	y	weight
1	0	0	1	13	193.89	38.891	1
2	27.5	0	1	14	193.89	54	1
3	55	0	1	15	193.89	70	1
4	82.5	0	1	16	162.89	70	1
5	110	0	1	17	131.18	70	1
6	110	45	0.70711	18	104.95	59	1
7	155	45	1	19	78.71	48	1
8	200	45	0.70711	20	52.474	37	1
9	200	0	1	21	26.237	26	1
10	205	0	1	22	0	15	1
11	210	0	1	23	0	7.5	1
12	210	22.782	0.92388	24	0	0	1

Table A.2: The control points of the initial geometry of the connecting rod

Index	x	y	z	Index	x	y	z
1	0.0000	3.9688	-23.1528	32	0.0000	13.9237	5.4004
2	0.0000	6.9849	-22.8453	33	-5.9566	3.9688	-23.1528
3	0.0000	9.1665	-18.9764	34	-5.9566	6.9849	-22.8453
4	0.0000	2.3000	-19.3471	35	-5.9566	9.1665	-18.9764
5	-5.9566	-7.8055	9.3776	36	-5.9566	2.3000	-19.3471
6	0.0000	9.6726	-15.1169	37	-5.9566	9.6726	-15.1169
7	0.0000	1.2800	-15.5661	38	-5.9566	1.2800	-15.5661
8	0.0000	-7.8055	9.3776	39	-5.9566	9.7359	-10.8778
9	0.0000	9.7359	-10.8778	40	-5.9566	1.3300	-10.9803
10	0.0000	1.3300	-10.9803	41	-5.9566	9.2025	-7.0183
11	0.0000	9.2025	-7.0183	42	-5.9566	1.2500	-7.1840
12	0.0000	1.2500	-7.1840	43	-5.9566	8.4433	-2.9690
13	0.0000	8.4433	-2.9690	44	-5.9566	2.4500	-2.9492
14	0.0000	2.4500	-2.9492	45	-5.9566	3.1054	0.0054
15	0.0000	3.1054	0.0054	46	-5.9566	7.5575	0.0047
16	0.0000	7.5575	0.0047	47	-5.9566	5.9840	4.8760
17	0.0000	5.9840	4.8760	48	-5.9566	2.3300	4.6616
18	0.0000	2.3300	4.6616	49	-5.9566	6.0279	10.9346
19	0.0000	6.0279	10.9346	50	-5.9566	2.5480	12.2633
20	0.0000	2.5480	12.2633	51	-5.9566	-0.5316	12.7097
21	0.0000	-0.5316	12.7097	52	-5.9566	-0.3092	7.2275
22	0.0000	-0.3092	7.2275	53	-5.9566	-5.2611	13.3401
23	0.0000	-5.2611	13.3401	54	-5.9566	-5.2223	5.1458
24	0.0000	-5.2223	5.1458	55	-5.9566	-7.8026	4.5231
25	0.0000	-7.8026	4.5231	56	-5.9566	-7.9245	13.5061
26	0.0000	-7.9245	13.5061	57	-5.9566	7.1933	7.9653
27	0.0000	7.1933	7.9653	58	-5.9566	8.0613	9.8869
28	0.0000	8.0613	9.8869	59	-5.9566	11.2365	8.0934

Continued...

Index	x	y	z	Index	x	y	z
29	0.0000	11.2365	8.0934	60	-5.9566	10.9684	6.3494
30	0.0000	10.9684	6.3494	61	-5.9566	12.2957	5.0262
31	0.0000	12.2957	5.0262	62	-5.9566	13.9237	5.4004

Table A.3: The control points of the initial hammer geometry (all of the weights $w_A = 1$)

Index	x	y	z	Index	x	y	z
1	0.0000	1.3048	-22.9571	32	0.0000	15.9543	13.5912
2	0.0000	7.9927	-22.7503	33	-5.9566	1.3048	-22.9571
3	0.0000	7.7429	-19.0020	34	-5.9566	7.9927	-22.7503
4	0.0000	1.3738	-19.0961	35	-5.9566	7.7429	-19.0020
5	-5.9566	-7.8055	9.3776	36	-5.9566	1.3738	-19.0961
6	0.0000	7.7000	-15.6545	37	-5.9566	7.7000	-15.6545
7	0.0000	1.4000	-15.7177	38	-5.9566	1.4000	-15.7177
8	0.0000	-7.8055	9.3776	39	-5.9566	7.7000	-11.0404
9	0.0000	7.7000	-11.0404	40	-5.9566	1.4000	-11.1547
10	0.0000	1.4000	-11.1547	41	-5.9566	7.7000	-6.9239
11	0.0000	7.7000	-6.9239	42	-5.9566	1.4000	-7.0875
12	0.0000	1.4000	-7.0875	43	-5.9566	7.5000	-3.0789
13	0.0000	7.5000	-3.0789	44	-5.9566	1.6500	-2.9837
14	0.0000	1.6500	-2.9837	45	-5.9566	1.6500	-0.1529
15	0.0000	1.6500	-0.1529	46	-5.9566	7.5000	-0.3425
16	0.0000	7.5000	-0.3425	47	-5.9566	7.3032	5.7972
17	0.0000	7.3032	5.7972	48	-5.9566	1.8695	5.4164
18	0.0000	1.8695	5.4164	49	-5.9566	6.0805	13.5912
19	0.0000	6.0805	13.5912	50	-5.9566	2.5947	13.5124
20	0.0000	2.5947	13.5124	51	-5.9566	-0.4968	13.5439
21	0.0000	-0.4968	13.5439	52	-5.9566	-0.4903	5.4318
22	0.0000	-0.4903	5.4318	53	-5.9566	-5.1971	13.5439
23	0.0000	-5.1971	13.5439	54	-5.9566	-5.1652	5.4006
24	0.0000	-5.1652	5.4006	55	-5.9566	-8.3408	4.8699
25	0.0000	-8.3408	4.8699	56	-5.9566	-7.9245	13.5061
26	0.0000	-7.9245	13.5061	57	-5.9566	9.0269	5.4525
27	0.0000	9.0269	5.4525	58	-5.9566	9.4320	13.5908
28	0.0000	9.4320	13.5908	59	-5.9566	12.2319	13.6070

Continued...

Index	x	y	z	Index	x	y	z
29	0.0000	12.2319	13.6070	60	-5.9566	12.1295	5.4490
30	0.0000	12.1295	5.4490	61	-5.9566	15.7837	5.2422
31	0.0000	15.7837	5.2422	62	-5.9566	15.9543	13.5912

Table A.4: The control points of the initial T-shape geometry (all of the weights $w_A = 1$)

Index	x	y	z	Index	x	y	z
0	-11.8248	-5.5254	18.1044	137	0.0000	-5.5254	15.8521
1	0.7906	7.5000	11.0567	138	4.3897	-5.5254	16.0429
2	0.0000	7.5000	10.9089	139	8.1354	-5.5254	16.5386
3	11.8248	-5.5254	18.1044	140	6.9099	8.5726	26.9186
4	9.7096	1.7408	25.1715	141	11.8531	-0.7124	21.7036
5	0.0000	4.3550	10.9496	142	-3.2353	9.8577	20.5650
6	9.0343	5.7337	27.3118	143	-3.1381	10.6282	26.9340
7	11.8248	-1.0635	18.1044	144	0.0000	-1.0635	15.8521
8	0.7230	5.2923	11.1037	145	4.3897	-1.0635	16.0429
9	1.2093	0.8429	4.1005	146	8.1354	-1.0635	16.5386
10	-9.7096	1.7408	25.1715	147	11.5536	-2.0665	24.2342
11	-9.0343	5.7337	27.3118	148	-11.4759	1.6468	24.5411
12	0.0000	1.4393	3.9528	149	3.2353	9.8577	20.5650
13	12.6838	-4.6143	21.6804	150	-1.2129	9.8197	19.9113
14	0.0000	-2.0000	5.1199	151	-1.0959	10.9445	26.9385
15	8.8457	6.8115	23.2897	152	0.0000	2.1691	2.6246
16	1.1059	2.0000	5.2739	153	0.0000	2.8779	1.7493
17	0.0000	-5.5254	14.0394	154	3.1381	10.6282	26.9340
18	4.4307	-5.5254	14.2320	155	11.4759	1.6468	24.5411
19	8.3585	-5.5254	14.7834	156	-10.8708	-0.1457	23.8054
20	6.7088	7.8334	27.0720	157	1.2129	9.8197	19.9113
21	12.6838	-0.7093	21.6804	158	0.0000	12.5468	30.7859
22	2.6954	3.6164	1.8235	159	-4.3897	-9.6560	16.0429
23	-11.8248	-1.0635	18.1044	160	1.0959	10.9445	26.9385
24	0.0000	-1.0635	14.0394	161	-8.1354	-9.1066	16.5386
25	4.4307	-1.0635	14.2320	162	0.0000	9.9873	19.9147
26	8.3585	-1.0635	14.7834	163	-11.0548	-8.1920	18.3343
27	12.2012	-2.1165	25.4895	164	0.0000	11.1088	26.9390

Continued...

Index	x	y	z	Index	x	y	z
28	-12.6838	-4.6143	21.6804	165	-7.4417	10.9747	30.3230
29	3.1344	8.3358	20.6542	166	10.8708	-0.1457	23.8054
30	0.0000	2.6904	1.6758	167	0.0000	13.0616	33.6076
31	0.0000	-4.9728	1.7165	168	-3.7288	5.9288	16.0429
32	2.5584	-4.1710	1.8706	169	-6.9106	5.7090	16.5386
33	3.6891	3.1133	1.9998	170	-9.1637	7.7478	22.3643
34	3.0405	9.1183	27.0720	171	0.0000	-9.8751	15.8521
35	12.1862	1.5945	25.7384	172	-11.8536	-6.7256	21.7112
36	-8.8457	6.8115	23.2897	173	-5.5891	12.1772	32.2636
37	1.1423	8.2892	19.9854	174	-9.5096	4.5044	18.3343
38	2.8981	-4.9347	1.9441	175	-3.2015	9.4970	19.7039
39	0.0000	3.1403	1.8521	176	4.3897	-9.6560	16.0429
40	1.0484	9.4248	27.0720	177	-3.1521	14.0000	33.3505
41	0.0000	-5.4990	1.8928	178	-1.1009	10.5000	15.9350
42	0.0000	8.1864	19.9854	179	-1.1307	13.6880	33.5333
43	3.3736	-6.0039	2.0469	180	1.1174	13.0608	33.6074
44	0.0000	9.4248	27.0720	181	8.1354	-9.1066	16.5386
45	9.1301	9.6359	0.8981	182	1.1162	12.5449	30.7855
46	10.4367	-0.6280	24.2471	183	-11.6124	-4.6372	23.5141
47	-4.4307	-5.5254	14.2320	184	-11.8527	2.0109	21.6890
48	-8.3585	-5.5254	14.7834	185	-11.2311	4.3120	25.4607
49	0.0000	1.6739	0.7504	186	-0.7906	7.5000	11.0567
50	-6.7088	7.8334	27.0720	187	3.1266	12.7512	33.4266
51	0.0000	-9.8751	14.0394	188	3.1463	12.2312	30.6770
52	-12.6838	-0.7093	21.6804	189	-0.7230	5.2923	11.1037
53	1.9532	-3.3553	3.2886	190	-1.2093	0.8429	4.1005
54	-4.4307	-1.0635	14.2320	191	5.6167	11.8676	32.3628
55	0.0000	-9.6735	0.7911	192	6.1394	10.6753	30.0365

Continued...

Index	x	y	z	Index	x	y	z
56	4.4307	-9.6560	14.2320	193	-1.1059	2.0000	5.2739
57	-8.3585	-1.0635	14.7834	194	-2.6954	3.6164	1.8235
58	-12.2012	-2.1165	25.4895	195	7.4667	10.5979	30.4504
59	-3.1344	8.3358	20.6542	196	8.2112	9.0518	28.5842
60	-3.0405	9.1183	27.0720	197	-2.5584	-4.1710	1.8706
61	8.3585	-9.1066	14.7834	198	11.0548	-8.1920	18.3343
62	9.4216	-4.7046	0.9452	199	7.4417	10.9747	30.3230
63	9.1301	9.6359	0.1140	200	-3.6891	3.1133	1.9998
64	-12.1862	1.5945	25.7384	201	0.0000	5.9288	15.8521
65	-1.1423	8.2892	19.9854	202	3.7288	5.9288	16.0429
66	-1.0484	9.4248	27.0720	203	11.5387	4.2867	25.7199
67	0.0000	1.6739	0.1257	204	11.3331	3.2015	25.0775
68	0.0000	-4.1572	3.1346	205	-3.3736	-6.0039	2.0469
69	0.0000	-5.1920	1.7900	206	-9.1301	9.6359	0.8981
70	0.0000	-9.6735	0.1225	207	-1.1162	12.5449	30.7855
71	9.4216	-4.7046	0.1102	208	6.9106	5.7090	16.5386
72	-10.4367	-0.6280	24.2471	209	9.1637	7.7478	22.3643
73	9.1301	9.6359	0.1450	210	11.8536	-6.7256	21.7112
74	3.1094	3.4067	1.8970	211	5.5891	12.1772	32.2636
75	-4.4307	-9.6560	14.2320	212	9.5096	4.5044	18.3343
76	-8.3585	-9.1066	14.7834	213	3.2015	9.4970	19.7039
77	-11.8248	-8.1920	18.1044	214	3.1521	13.3762	33.3505
78	11.8248	-8.1920	18.1044	215	1.1009	10.5000	15.9350
79	7.5017	10.0703	30.6287	216	1.1307	13.6880	33.5333
80	-7.5017	10.0703	30.6287	217	0.0000	10.5000	15.7873
81	0.0000	5.9288	14.0394	218	0.0000	13.6894	33.5337
82	3.7636	5.9288	14.2320	219	-1.1174	13.0608	33.6074
83	0.0000	0.8141	0.1568	220	11.6124	-4.6372	23.5141

Continued...

Index	x	y	z	Index	x	y	z
84	2.0762	2.4608	2.7723	221	11.8527	2.0109	21.6890
85	0.0000	-8.3391	0.1535	222	-9.4216	-4.7046	0.9452
86	-3.7636	5.9288	14.2320	223	-9.1301	9.6359	0.1140
87	9.4216	-4.7046	0.1413	224	-3.1463	12.2312	30.6770
88	7.1001	5.7090	14.7834	225	11.2311	4.3120	25.4607
89	8.8172	7.0953	22.4229	226	-3.1266	12.7512	33.4266
90	12.6838	-6.7241	21.6804	227	-9.4216	-4.7046	0.1102
91	5.6552	11.4341	32.5015	228	-9.1301	9.6359	0.1450
92	10.6762	4.5044	18.1044	229	-6.1394	10.6753	30.0365
93	3.1059	7.9727	19.7874	230	-5.6167	11.8676	32.3628
94	3.0909	11.8762	33.5332	231	-9.4216	-4.7046	0.1413
95	1.0067	7.9262	15.9821	232	-8.2112	9.0518	28.5842
96	1.0988	12.1827	33.7110	233	-7.4667	10.5979	30.4504
97	0.0000	7.5988	15.8280	234	-11.3331	3.2015	25.0775
98	0.0000	12.1827	33.7110	235	-11.5387	4.2867	25.7199
99	-7.1001	5.7090	14.7834	236	-11.8529	0.8762	21.6951
100	12.2012	-4.7106	24.2648	237	-12.1990	2.0133	21.6854
101	12.6838	2.0165	21.6804	238	-11.0548	2.1845	18.3343
102	-8.8172	7.0953	22.4229	239	-9.7856	4.5044	18.2385
103	-12.6838	-6.7241	21.6804	240	-8.1354	2.8871	16.5386
104	-5.6552	11.4341	32.5015	241	-6.9895	5.7090	15.8073
105	11.9693	4.2514	26.0829	242	-4.3897	3.0153	16.0429
106	-10.6762	4.5044	18.1044	243	-3.7433	5.9288	15.2884
107	-3.1059	7.9727	19.7874	244	0.0000	3.0153	15.8521
108	-3.0909	11.8762	33.5332	245	0.0000	5.9288	15.0968
109	-1.0067	7.9262	15.9821	246	3.7288	3.0153	16.0429
110	-1.0988	12.1827	33.7110	247	3.7433	5.9288	15.2884
111	-1.9532	-3.3553	3.2886	248	6.9106	2.8871	16.5386

Continued...

Index	x	y	z	Index	x	y	z
112	-2.8981	-4.9347	1.9441	249	6.9895	5.7090	15.8073
113	-12.2012	-4.7106	24.2648	250	11.0548	2.1845	18.3343
114	-12.6838	2.0165	21.6804	251	9.7856	4.5044	18.2385
115	-11.9693	4.2514	26.0829	252	11.8529	0.8762	21.6951
116	-11.0548	-5.5254	18.3343	253	12.1990	2.0133	21.6854
117	-10.1089	2.2567	24.1159	254	-0.4566	10.9461	26.9388
118	-9.2885	6.3599	26.1498	255	0.4566	10.9461	26.9388
119	-11.0548	-1.0635	18.3343	256	0.5054	9.8233	19.9133
120	-11.8534	-4.6151	21.7077	257	-0.5054	9.8233	19.9133
121	-9.1568	7.4997	23.2265	258	0.4587	10.5000	15.8488
122	-2.0762	2.4608	2.7723	259	-0.4587	10.5000	15.8488
123	11.0548	-5.5254	18.3343	260	0.3294	7.5000	10.9705
124	10.1089	2.2567	24.1159	261	-0.3294	7.5000	10.9705
125	-4.3897	-5.5254	16.0429	262	-0.5039	1.1908	4.0143
126	9.2885	6.3599	26.1498	263	0.5039	1.1908	4.0143
127	11.0548	-1.0635	18.3343	264	-0.4195	7.7952	15.8922
128	-8.1354	-5.5254	16.5386	265	0.4195	7.7952	15.8922
129	-6.9099	8.5726	26.9186	266	0.4759	8.2892	19.9854
130	-11.8531	-0.7124	21.7036	267	-0.4759	8.2892	19.9854
131	-3.1094	3.4067	1.8970	268	-0.3012	4.8056	11.0138
132	-4.3897	-1.0635	16.0429	269	0.3012	4.8056	11.0138
133	11.8534	-4.6151	21.7077	270	-0.4368	9.4248	27.0720
134	-8.1354	-1.0635	16.5386	271	0.4608	2.0000	5.1840
135	9.1568	7.4997	23.2265	272	0.4368	9.4248	27.0720
136	-11.5536	-2.0665	24.2342	273	-0.4608	-2.0000	5.1840

Table A.5: The control points of the initial chair geometry (all of the weights $w_A = 1$)