Shape Sensitivity Analysis and Optimization using Isogeometric Boundary Element Methods in Two-dimensional Linear Elasticity

Haojie Lian[1], Robert N. Simpson[2], Stéphane P.A. Bordas[1,3,*], Pierre Kerfriden[1]

[1]School of Engineering, Cardiff University, Queen's Buildings, The parade, Cardiff, UK
[2]School of Engineering, Glasgow University, Rankine Building, Glasgow, UK
[3]Faculté des Sciences, de la Technologie, Luxembourg University, Luxembourg

*BordasS@cardiff.ac.uk

Abstract

The present work addresses shape sensitivity analysis and optimization in two-dimensional elasticity with an isogeometric boundary element method. The NURBS which generates geometric models are used as the basis functions to discretize the Boundary Integral Equations. The structural and sensitivity analysis use the regularized form of boundary integral equations, to avoid the difficulty in addressing strong singular integrals and jump terms. For the shape optimization, the control points are used as design variables. The main advantage of the present work is that it tightly integrates design models and analysis models, thus greatly reducing the mesh generation burden.

1 Introduction

Shape optimization is a process to find the optimal shape of a component or structure meanwhile satisfying the given requirements. To achieve a rational and automatic shape optimization, the Finite Element Method (FEM) [52] was applied combined with mathematical programming algorithms [51]. 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, which inevitably causes a cumbersome remeshing procedure (Fig. 1). The meshing/remeshing procedure is time-consuming and far from being automated. It may happen in industrial practices that the geometry is so complex that available mesh generators fail, or require significant human intervention. To alleviate the mesh burden, numerous works are proposed from various perspectives:

- Meshfree/Meshless methods

  Meshfree Methods [31], or called Meshless Methods, refer to a broad collection of numerical analysis methods, including the Element-Free Galerkin (EFG) Method [5], Reproducing Kernel Particle Method (RKPM) [24], Meshless Local Petrov Galerkin(MLPG) [1] method, *hp*-cloud method [18], Partition of Unity Finite Element Method (PUFEM) [2], *etc*. Although different in the way of formulating shape functions or test functions, they share the same characteristic of 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. Obviously, 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 [7, 6, 50]. 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. Moreover, the node positions and refinement need to be determined by experience or numerical experiments .

- Boundary element methods

  Boundary Element Methods (BEM) [15, 14, 36, 22, 42] take the advantage of Boundary Integral Equations to decrease the dimension of the problem by one, *i.e.* only line integral needed for two-dimensional problems, and surface integral for three-dimensional problems. The main advantage of BEM for shape optimization is that it greatly alleviates mesh generation burden because surface mesh generation is much easier and faster than the domain mesh. The application of BEM in shape optimization can be seen in [47, 48, 49]. However, the surface meshing of the BEM is still cumbersome for shape optimization in large scale problems.

- Extended finite element methods

  The main difficulty in mesh generation process emanates from the requirement of the mesh to conform to the (usually complex) geometry of the domain. To separate the FEM mesh and geometry representation, Extended Finite Element Method (XFEM) [4, 8, 9, 17] was proposed by introducing enrichment functions. XFEM is initially employed for crack propagation and then extended to moving boundary problems including shape optimization [45, 19, 30, 44]. However, due to the separation of geometry and analysis meshes, XFEM has the following shortcomings for its application in shape optimization

  - An implicit geometry description have to be used, *e.g.* level set. 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 objects. The shortcomings are that a care must be taken for constructing the velocity field and the geometry advancing needs to solve an differential equation.

  - The domain integration around the boundary requires a special treatment.

- Isogeometric analysis

  A recent trend in shape optimization is applying Isogeometric Analysis (IGA) [21], which is proposed to integrate the geometry and analysis representation. This is achieved by using the data provided by CAD models *directly* rather than converting it through a preprocessing routine into a form suitable for analysis. The main advantage is that a meshing procedure is bypassed because an existing CAD geometry model will be used for analysis, meanwhile keeping the exact geometry. The application of IGA in shape optimization can be found in the [10, 20, 29, 35, 46]. However, a bottleneck still exists, that is, current IGA is in a domain discretization framework, but CAD typically only provides surface representation. To get an analysis-suitable trivariate discretization from surface is far from a trivial task.

Inspired by IGA but intended to overcome its limitations, an isogeometric approach using the framework of the boundary element method - coined the isogeometric boundary element method (IGABEM)- was proposed [39, 33, 38, 40]. The idea relies on the fact that both CAD models and boundary element methods rely on quantities defined entirely on the boundary. With the advantage of achieving a tight integration of CAD and analysis, IGABEM has been a particularly suitable choice for the application in shape optimization, which is the present work built on.

This paper is organized as follows. Section 2 reviews B-splines and NURBS, which are in central of IGABEM. Section 3 presents the formulation of IGABEM in a regularized from. The shape sensitivity analysis with IGABEM is introduced in Section 4. Section 5 demonstrates the IGABEM shape optimization, followed by numerical examples in Section 6. Finally, Section 7 contains the conclusions.

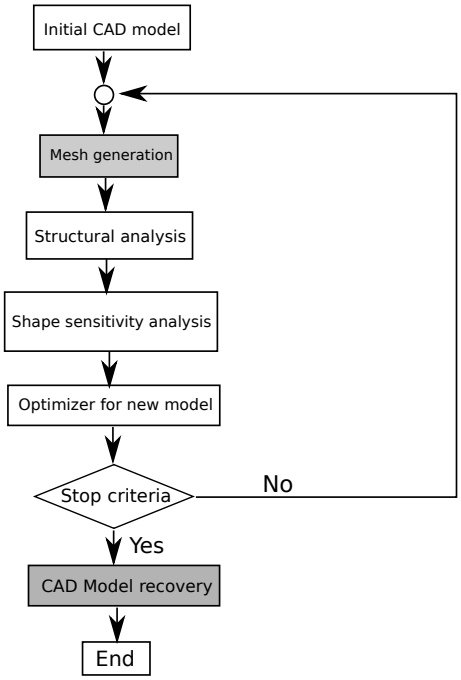

Figure 1. FEM shape optimization procedure

2 B-splines and NURBS

2.1 Knot vector

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

$$\{\xi_1, \xi_2, \cdots, \xi_{n+p+1}\},\ \xi_A \in \mathbb{R}$$

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.

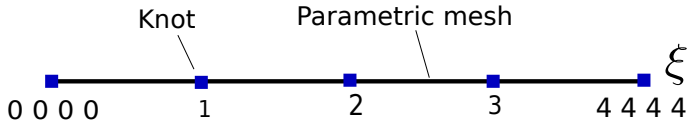

Figure 2. Knot vector

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 different 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}. 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-spline. 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 an uniform knot vector if the knots are uniformly spaced, for example, {0, 0, 0, 1, 2, 3, 4, 5, 5, 5}.

2.2 B-spline basis functions

With the concept of a knot vector, we can now define B-spline basis function using the **Cox-de Boor recursion formula** [13, 16]

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

$$N_{A,p}(\xi) = \frac{\xi - \xi_A}{\xi_{A+p} - \xi_A} N_{A,p-1}(\xi) + \frac{\xi_{A+p+1} - \xi}{\xi_{A+p+1} - \xi_{A+1}} N_{A+1,p-1}(\xi). \tag{2}$$

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. 3):

- 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 has an important significance for interactive design: the change of one control point only affects the local part of the curve, giving great convenience for curve modification.
- 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 3. B-spline basis functions ($p=3$)

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

$$\frac{\mathrm{d}}{\mathrm{d}\xi} N_{A,p}(\xi) = \frac{p}{\xi_{A+p} - \xi_A} N_{A,p-1}(\xi) - \frac{p}{\xi_{A+p+1} - \xi_{A+1}} N_{A+1,p-1}(\xi). \tag{3}$$

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

$$\begin{aligned}\frac{\mathrm{d}^k}{\mathrm{d}^k\xi}N_{A,p}(\xi) &= \frac{p}{\xi_{A+p}-\xi_A}\left(\frac{\mathrm{d}^{k-1}}{\mathrm{d}^{k-1}\xi}N_{A,p-1}(\xi)\right)\\ &\quad -\frac{p}{\xi_{A+p+1}-\xi_{A+1}}\left(\frac{\mathrm{d}^{k-1}}{\mathrm{d}^{k-1}\xi}N_{A+1,p-1}(\xi)\right).\end{aligned} \tag{4}$$

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 the following

$$\frac{\mathrm{d}^k}{\mathrm{d}^k\xi}N_{A,p}(\xi)=\frac{p!}{(p-k)!}\sum_{j=0}^{k}\alpha_{k,j}N_{A+j,p-k}(\xi), \tag{5}$$

with

$$\begin{aligned}\alpha_{0,0} &= 1,\\ \alpha_{k,0} &= \frac{\alpha_{k-1,0}}{\xi_{A+p-k+1}-\xi_A},\\ \alpha_{k,j} &= \frac{\alpha_{k-1,j}-\alpha_{k-1,j-1}}{\xi_{A+p+j-k+1}-\xi_{A+j}}, \qquad j=1,....,k-1,\\ \alpha_{k,k} &= \frac{-\alpha_{k-1,k-1}}{\xi_{A+p+1}-\xi_{A+k}}.\end{aligned} \tag{6}$$

2.3 NURBS basis functions

Non-uniform Rational B-Splines (NURBS) [34, 37] 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. In addition, all numerical examples in the present work are represented by NURBS.

NURBS basis function $R_{A,p}$ is defined as

$$R_{A,p}(\xi)=\frac{N_{A,p}(\xi)w_A}{W(\xi)}, \tag{7}$$

with

$$W(\xi)=\sum_{A=1}^{n}w_AN_{A,p}(\xi). \tag{8}$$

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 NURBS geometry, with higher values drawing the curve closer to that point (Fig. 4). When all of the weights are equal to 1, the NURBS reduces to a B-spline curve.

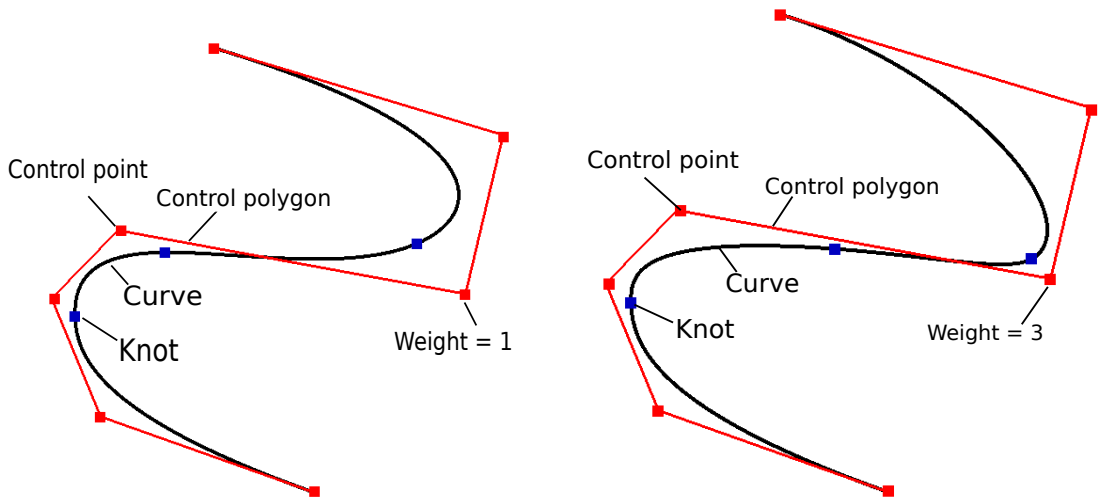

Figure 4. The comparison beween NURBS curve with different weights

The derivatives of NURBS basis function are expressed by

$$\frac{\mathrm{d}}{\mathrm{d}\xi}R_{A,p}(\xi)=w_A\frac{W(\xi)\frac{\mathrm{d}}{\mathrm{d}\xi}N_{A,p}(\xi)-\frac{\mathrm{d}}{\mathrm{d}\xi}W(\xi)N_{A,p}(\xi)}{(W(\xi))^2}, \tag{9}$$

and

$$\frac{\mathrm{d}}{\mathrm{d}\xi}W(\xi)=\sum_{A=1}^{n}\frac{\mathrm{d}}{\mathrm{d}\xi}N_{A,p}(\xi)w_A. \tag{10}$$

2.4 NURBS geometries

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

$$R_A(\boldsymbol{\xi}|\boldsymbol{\Xi}_A)\equiv\prod_{i=1}^{d_p}R_A^i(\xi_A^i|\Xi_A^i), \tag{11}$$

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

$$R_{A,B}^{p,q}(\xi,\eta) = \frac{N_{A,p}(\xi)M_{B,q}(\eta)w_{A,B}}{\sum_{\hat{A}=1}^{n}\sum_{\hat{B}=1}^{m}N_{\hat{A},p}(\xi)M_{\hat{B},q}(\eta)w_{\hat{A},\hat{B}}}, \tag{12}$$

$$R_{A,B,C}^{p,q,r}(\xi,\eta,\zeta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)L_{k,r}(\zeta)w_{i,j,k}}{\sum_{\hat{A}=1}^{n}\sum_{\hat{B}=1}^{m}\sum_{\hat{C}=1}^{l}N_{\hat{A},p}(\xi)M_{\hat{B},q}(\eta)L_{\hat{C},r}(\zeta)w_{\hat{A},\hat{B},\hat{C}}}. \tag{13}$$

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

$$\mathbf{x}(\xi)=\sum_{A}^{n}R_{A,p}(\xi)\mathbf{P}_A, \tag{14}$$

where $\mathbf{x}(\xi)$ denotes the physical curve we are interested, ξ the coordinate in parametric space, $\mathbf{P}_A$ the control points, $N_{A,p}$ the B-spline basis functions of order p. See Fig. 5. NURBS surface can be constructed using the similar way (Fig. 6).

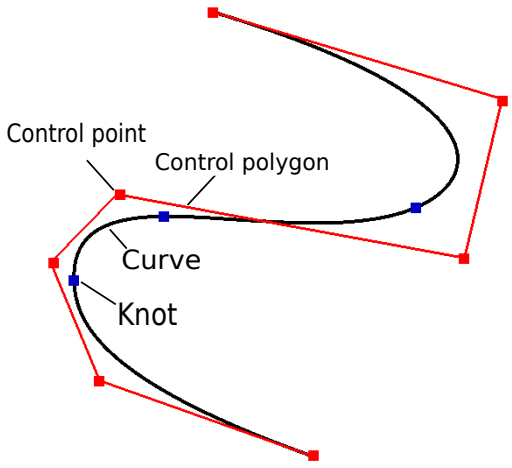

Figure 5. NURBS curve

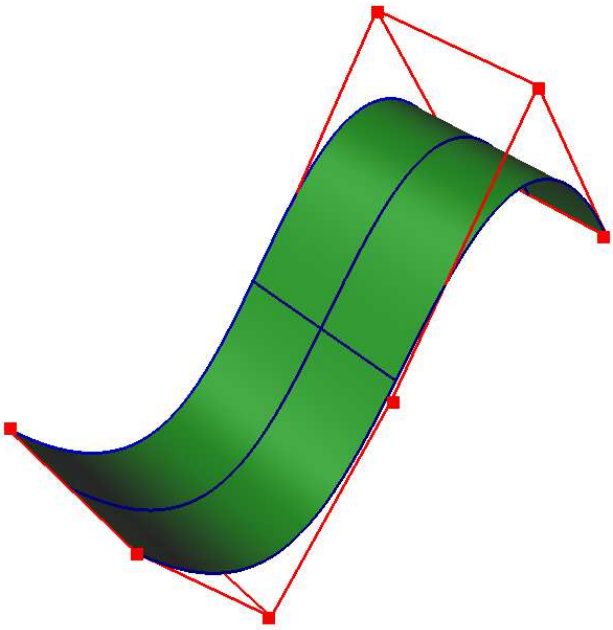

Figure 6. NURBS surface

Another important interpretation of NURBS geometry is a linear combination of standard B-spline basis functions and weighted control points

$$\mathbf{x}(\xi) = \sum_{A=1}^{n} N_{A,p}(\xi)\tilde{\mathbf{P}}_A, \tag{15}$$

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

NURBS geometries possess the following properties

- The convex hull property. The NURBS geometry is contained in the convex hull constructed by the control grid, which is a mesh interpolated by control points. See Fig. (6).
- The variation diminishing property. No plane has more intersections with the curve than it has with the control grids. This property renders NURBS less oscillatory than Lagrangian polynomials.
- The transformation invariance property. An affine transformation to NURBS can be achieved by applying an affine transformation to the control points.
- Non-interpolatory. The NURBS geometry do not interpolate the control points except at the start point, end point and any point whose knot value is repeated p times.

2.5 Knot insertion in NURBS

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 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,

$$\bar{\tilde{\mathbf{P}}}_A = \begin{cases} \tilde{\mathbf{P}}_1, & A = 1, \\ \alpha_A\tilde{\mathbf{P}}_A + (1-\alpha_A)\tilde{\mathbf{P}}_{A-1}, & 1 < A < m, \\ \tilde{\mathbf{P}}_n, & A = m, \end{cases} \tag{16}$$

with

$$\alpha_A = \begin{cases} 1, & 1 \leqslant A \leqslant k-p, \\ \dfrac{\bar{\xi}_A - \xi_A}{\xi_{A+p} - \xi_A}, & k-p+1 \leqslant A \leqslant k, \\ 0, & A \geqslant k+1. \end{cases} \tag{17}$$

2.6 NURBS element structure

Knot vector used for defining the NURBS basis functions provides a natural element structure which is very useful for numerical analysis using FEM or BEM. We can view the non-zero knot interval as the element in each dimension. The only difference from the isoparametric element which is widely used in analysis is that each element in NURBS has a different set of basis functions. To employ the Gauss-Legendre quadrature rule, we can transfer the space defined in each knot intervals into a standard $[-1,+1]$ space $\hat{\Omega}$, and $\mathrm{d}\Omega_e = J_e \mathrm{d}\hat{\Omega}_e$, with

$$J_e = \frac{\xi_b - \xi_a}{2}, \tag{18}$$

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 transforming parent elements to parametric elements and varies with the element.

3 Isogeometric boundary element methods

3.1 Isogeometric boundary element method

The boundary integral equation in linear elasticity is written as

$$C_{ij}(\mathbf{s})u_j(\mathbf{s}) + \int_S T_{ij}(\mathbf{s},\mathbf{x})u_j(\mathbf{x})\mathrm{d}S(\mathbf{x}) = \int_S U_{ij}(\mathbf{s},\mathbf{x})t_j(\mathbf{x})\mathrm{d}S(\mathbf{x}), \tag{19}$$

where C_{ij} is the jump term, u_j and t_j are the displacement field and traction field around the boundary, respectively. T_{ij} and U_{ij} are the displacement and traction fundamental solutions, which in two-dimensional elasticity are written as

$$U_{ij}(\mathbf{s},\mathbf{x}) = \frac{1}{8\pi\mu(1-\nu)}\left[(3-4\nu)\ln\frac{1}{r}\delta_{ij} + r_{,i}r_{,j}\right], \tag{20}$$

$$T_{ij}(\mathbf{s},\mathbf{x}) = -\frac{1}{4\pi(1-\nu)r}\left\{\frac{\partial r}{\partial n}[(1-2\nu)\delta_{ij} + 2r_{,i}r_{,j}] - (1-2\nu)(r_{,i}n_j - r_{,j}n_i)\right\}, \tag{21}$$

where $\mathbf{x}$ is the field point on the boundary, $\mathbf{s}$ the source point, $r = ||\mathbf{x}-\mathbf{s}||$ the distance between the source point and field point (Fig. 7).

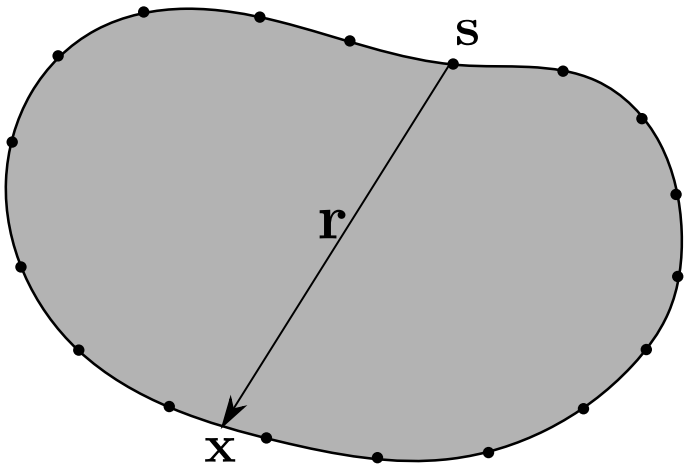

Figure 7. The distance beween the source point and field point

Eq. (19) is a singular from and requires an explicit evaluation of jump terms and strong singular integrals in IGABEM. To overcome this difficulty, we employ a regularized boundary integral equation proposed by Liu in [25, 27, 26], which is written as

$$\int_S T_{ij}(\mathbf{s},\mathbf{x})[u_j(\mathbf{x})-u_j(\mathbf{s})]\mathrm{d}S(\mathbf{x}) = \int_S U_{ij}(\mathbf{s},\mathbf{x})t_j(\mathbf{x})\mathrm{d}S(\mathbf{x}). \tag{22}$$

The regularized form cancels the singularity of the left-hand side of Eq. (22) because

$$T_{ij}(\mathbf{s},\mathbf{x})[u_j(\mathbf{x})-u_j(\mathbf{s})] \sim O\left(\frac{1}{r}\right)O(r) = O(1), \text{ in 2D.} \tag{23}$$

The right-hand of Eq. (22) can be solved using a Telles transformation [43].

The geometry in Eq. (22) has been parameterized using NURBS basis functions

$$\mathbf{x}(\xi) = R_A(\xi)\mathbf{P}_A, \tag{24}$$

where A denotes the global index of the basis function, $\mathbf{P}$ the control point, $\boldsymbol{\xi}$ the parametric coordinates of the field points.

The displacement and traction fields around the boundary are discretized using NURBS basis functions, which is the main difference from the traditional BEM,

$$u_j(\xi) = R_A(\xi)\tilde{u}_j^A, \tag{25}$$

$$t_j(\xi) = R_A(\xi)\tilde{t}_j^A, \tag{26}$$

where $\tilde{u}_j^A$ and $\tilde{t}_j^A$ are the nodal unknowns related to displacements and tractions, and $\hat{\boldsymbol{\xi}}$ are the intrinsic coordinates of the field points in parent elements.

Substituting Eqs. (24), (25) and (26) to Eq. (22) leads to a discrete form of regularized BIE

$$\begin{aligned} &\int_S T_{ij}(\zeta_c,\xi)[R_A(\xi)-R_A(\zeta_c)]J(\xi)\mathrm{d}S(\xi)\tilde{u}_j^A \\ =\ &\int_S U_{ij}(\zeta_c,\xi)R_A(\xi)J_e(\xi)\mathrm{d}S(\xi)\tilde{t}_j^A. \end{aligned} \tag{27}$$

where $\boldsymbol{\zeta}_c$ denotes the parametric coordinates of collocation points and c the index of the collocation points.

To perform a numerical integration using Gauss-Legendre rule, the above integral should be transformed into parent elements with a Jacobian $\tilde{J}_e$. So the above integral becomes

$$\begin{aligned} &\int_{-1}^{+1} T_{ij}(\zeta_c,\xi)[R_A(\xi)-R_A(\zeta_c)]J(\xi)\tilde{J}_e\big(\tilde{\xi}\big)\mathrm{d}S\big(\tilde{\xi}\big)\tilde{u}_j^A \\ =\ &\int_{-1}^{+1} U_{ij}(\zeta_c,\xi)R_A(\xi)J(\xi)\tilde{J}_e\big(\tilde{\xi}\big)\mathrm{d}S\big(\tilde{\xi}\big)\tilde{t}_j^A. \end{aligned} \tag{28}$$

By denoting the terms of the above equation as

$$H_{ij}^{cA} = \int_{-1}^{+1} T_{ij}(\zeta_c,\xi)[R_A(\xi)-R_A(\zeta_c)]J(\xi)\tilde{J}_e\big(\tilde{\xi}\big)\mathrm{d}S\big(\tilde{\xi}\big), \tag{29}$$

$$G_{ij}^{cA} = \int_{-1}^{+1} U_{ij}(\zeta_c,\xi)R_A(\xi)J(\xi)\tilde{J}_e\big(\tilde{\xi}\big)\mathrm{d}S\big(\tilde{\xi}\big), \tag{30}$$

we can rewrite Eq. (27) in a matrix form

$$\mathbf{Hu}{=}\mathbf{Gt}, \tag{31}$$

where $\mathbf{H}$ and $\mathbf{G}$ are the matrix collecting the terms of H_{ij}^{cA} and G_{ij}^{cA}, respectively. $\mathbf{u}$ contains the nodal displacements, and $\mathbf{t}$ the nodal tractions. Both $\mathbf{u}$ and $\mathbf{t}$ include unknowns and the known values given by boundary conditions. By swapping the unknowns of both sides of Eq. (31), we can rearrange it as

$$\mathbf{Az} = \mathbf{By} = \mathbf{f}. \tag{32}$$

The 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 vector $\mathbf{f}$ on the right-hand side.

3.2 Imposition of boundary conditions

NURBS basis functions lack *kronecker delta* property, so the nodal parameters do not possess a clear physical interpretation. Hence, the boundary conditions cannot be substituted directory into nodal parameters. In isogeometric finite element method, this task can be done through Lagrange multiplier methods, penalty methods, or Nitsche method. However, these methods are not available in collocation IGABEM, because it is not based on variational problems. Hence, a nodal parameter extraction approach should be used, which can be conducted by collocation method or Galerkin method.

3.2.1 Collocation method

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

$$\mathbf{u}(\tilde{\boldsymbol{\xi}}_c) = \bar{\mathbf{u}}(\tilde{\boldsymbol{\xi}}_c), \quad \text{on } S_u, \tag{33}$$

$$\mathbf{t}(\tilde{\boldsymbol{\xi}}_c) = \bar{\mathbf{t}}(\tilde{\boldsymbol{\xi}}_c), \quad \text{on } S_t, \tag{34}$$

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

Substituting Eqs.(25) and (26) into above equations and using matrix form produces

$$\mathbf{R}(\tilde{\boldsymbol{\xi}}_c)\tilde{\mathbf{u}} = \bar{\mathbf{u}}(\tilde{\boldsymbol{\xi}}_c), \quad \text{on } S_u, \tag{35}$$

$$\mathbf{R}(\tilde{\boldsymbol{\xi}}_c)\tilde{\mathbf{t}} = \bar{\mathbf{t}}(\tilde{\boldsymbol{\xi}}_c), \quad \text{on } S_t, \tag{36}$$

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 $\mathbf{I}$ the identity matrix.

After obtaining the $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{t}}$, we can substitute them into the governing equation for analysis.

3.2.2 Galerkin method

Galerkin method is to force boundary conditions satisfied in a "average" sens, *i.e.*

$$\int_{S_u} \mathbf{R}^{\mathrm{T}}\mathbf{u}\mathrm{d}S = \int_{S_u} \mathbf{R}^{\mathrm{T}}\bar{\mathbf{u}}\mathrm{d}S, \quad \text{on } S_u, \tag{37}$$

$$\int_{S_t} \mathbf{R}^{\mathrm{T}}\mathbf{u}\mathrm{d}S = \int_{S_t} \mathbf{R}^{\mathrm{T}}\bar{\mathbf{u}}\mathrm{d}S, \quad \text{on } S_t, \tag{38}$$

where the shape function $\mathbf{R}$ is used as weighting function. Substituting Eqs.(25) and (26) into above equations leads to

$$\int_{S_u} \mathbf{R}^{\mathrm{T}}\mathbf{R}\tilde{\mathbf{u}}\mathrm{d}S = \int_{S_u} \mathbf{R}^{\mathrm{T}}\bar{\mathbf{u}}\mathrm{d}S, \quad \text{on } S_u, \tag{39}$$

$$\int_{S_t} \mathbf{R}^{\mathrm{T}}\mathbf{R}\tilde{\mathbf{t}}\mathrm{d}S = \int_{S_t} \mathbf{R}^{\mathrm{T}}\bar{\mathbf{t}}\mathrm{d}S, \quad \text{on } S_t. \tag{40}$$

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

$$\begin{aligned}\mathbf{A}_1\tilde{\mathbf{u}} &= \mathbf{z}_1, \quad \text{on } S_u,\\ \mathbf{A}_2\tilde{\mathbf{t}} &= \mathbf{z}_2, \quad \text{on } S_t,\end{aligned}$$

where

$$\begin{aligned}\mathbf{A}_1 &= \int_{S_u} \mathbf{R}^{\mathrm{T}}\mathbf{R}\mathrm{d}S \quad \text{on } S_u,\\ \mathbf{A}_2 &= \int_{S_t} \mathbf{R}^{\mathrm{T}}\mathbf{R}\mathrm{d}S \quad \text{on } S_t.\end{aligned}$$

and

$$\mathbf{z}_1 = \int_{S_u} \mathbf{R}^{\mathrm{T}}\bar{\mathbf{u}}\mathrm{d}S, \quad \text{on } S_u, \tag{41}$$

$$\mathbf{z}_2 = \int_{S_t} \mathbf{R}^{\mathrm{T}}\bar{\mathbf{t}}\mathrm{d}S, \quad \text{on } S_t. \tag{42}$$

Without needing integrals, the collocation method is more efficient than Galerkin method. However, the Galerkin method is more elegant in the case of the geometry with corners where a care must be taken for choosing collocation point positions.

4 Shape sensitivity analysis with IGABEM

Shape sensitivity analysis refers to the evaluation of the derivatives with respect to design variables. Shape sensitivity analysis is a critical step for gradient-based shape optimization, although its application is not limited to it. In the context of boundary integral equation, typically three ways are used to conduct sensitivity analysis, 1) finite difference method, 2) ajoint variable method [12, 28], and 3) implicit differentiation method [3, 11, 23]. Finite difference methods is very easy to implement but the accuracy is not enough satisfying. Ajoint variable method uses an ajoint state to obtain an sensitivity expression for each design variable, particularly useful for the case with a great number of design variables and small number of constraints. However, an ajoint variable normally corresponds to a concentrated point force, which is not consistent with the distributed traction used in BEM. So the concentrated force is always approximated using a traction exerted on a small area, which decrease the accuracy and robustness of the algorithm. Implicit differentiation method is directly differentiating BIE with respect to design variables, and generates analytical forms of BIE sensitivities. Due to its accuracy and convenience for BIE, the present work will employ the implicit differentiation method, and adopt regularized BIE to generate its differentiation form.

4.1 Implicit differentiation with regularized IGABEM

In IGABEM, we will take the regularized form as the basis of shape sensitivity analysis. Hence, we took shape derivatives with respect to the regularized BIE and obtain the following implicit differentiation equation

$$\begin{aligned}&\int_S \left\{\dot{T}_{ij}(\mathbf{s},\mathbf{x})[u_j(\mathbf{x})-u_j(\mathbf{s})] + T_{ij}(\mathbf{s},\mathbf{x})[\dot{u}_j(\mathbf{x})-\dot{u}_j(\mathbf{s})\right\}\mathrm{d}S(\mathbf{x})\\ &+\int_S T_{ij}(\mathbf{s},\mathbf{x})[u_j(\mathbf{x})-u_j(\mathbf{s})][\mathrm{d}\dot{S}(\mathbf{x})]\\ =\ &\int_S \left[\dot{U}_{ij}(\mathbf{s},\mathbf{x})t_j(\mathbf{x}) + U_{ij}(\mathbf{s},\mathbf{x})\dot{t}_j(\mathbf{x})\right]\mathrm{d}S(\mathbf{x})\\ &+\int_S U_{ij}(\mathbf{s},\mathbf{x})t_j(\mathbf{x})[\mathrm{d}\dot{S}(\mathbf{x})]\end{aligned} \tag{43}$$

We use subscript $\dot{}$ denotes the material derivatives. 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 weak singular.

The shape derivatives of field points are given by

$$\dot{\mathbf{x}}(\xi) = R_A(\xi)\dot{\mathbf{P}}_A. \tag{44}$$

Discretize the displacement and traction field around the boundary as

$$u_j(\xi) = R_A(\xi)\tilde{u}_j^A, \tag{45}$$

$$t_j(\xi) = R_A(\xi)\tilde{t}_j^A, \tag{46}$$

and also discretize the shape derivatives of boundary displacement and traction field as

$$\dot{u}_j(\xi) = R_A(\xi)\dot{\tilde{u}}_j^A, \tag{47}$$

$$\dot{t}_j(\xi) = R_{ea}(\xi)\dot{\tilde{t}}_j^A. \tag{48}$$

After discretization, Eq. (22) can be expressed by

$$\begin{aligned}
&\int_S \left\{\dot{T}_{ij}(\mathbf{s},\mathbf{x})[R_A(\xi)-R_A(\zeta)]J(\xi) + T_{ij}(\mathbf{s},\mathbf{x})[R_A(\xi)-R_A(\zeta)]\dot{J}(\xi)\right\}\mathrm{d}S(\xi)\tilde{u}_j^A \\
&+\int_S T_{ij}(\mathbf{s},\mathbf{x})[R_A(\xi)-R_A(\zeta)]J(\xi)\mathrm{d}S(\xi)\dot{\tilde{u}}_j^A \\
=\ &\int_S \left[\dot{U}_{ij}(\mathbf{s},\mathbf{x})R_A(\xi)J(\xi) + U_{ij}(\mathbf{s},\mathbf{x})R_A(\xi)\dot{J}(\xi)\right]\mathrm{d}S(\xi)\tilde{t}_j^A \\
&+\int_S U_{ij}(\mathbf{s},\mathbf{x})R_A(\xi)J(\xi)\mathrm{d}S(\xi)\dot{\tilde{t}}_j^A.
\end{aligned} \tag{49}$$

The above integral also needs to transfer to parent elements where Gauss-Legendre quadrature rules can be used,

$$\begin{aligned}
&\int_{-1}^{+1} \left\{\dot{T}_{ij}(\mathbf{s},\mathbf{x})[R_A(\xi)-R_A(\zeta)]J(\xi) + T_{ij}(\mathbf{s},\mathbf{x})[R_A(\xi)-R_A(\zeta)]\dot{J}(\xi)\right\}\tilde{J}_e\big(\tilde{\xi}\big)\mathrm{d}S\big(\tilde{\xi}\big)\tilde{u}_j^A \\
&+\int_{-1}^{+1} T_{ij}(\mathbf{s},\mathbf{x})[R_A(\xi)-R_A(\zeta)]J(\xi)\tilde{J}_e\big(\tilde{\xi}\big)\mathrm{d}S\big(\tilde{\xi}\big)\dot{\tilde{u}}_j^A \\
=\ &\int_{-1}^{+1} \left[\dot{U}_{ij}(\mathbf{s},\mathbf{x})R_A(\xi)J(\xi) + U_{ij}(\mathbf{s},\mathbf{x})R_A(\xi)\dot{J}(\xi)\right]\tilde{J}_e\big(\tilde{\xi}\big)\mathrm{d}S\big(\tilde{\xi}\big)\tilde{t}_j^A \\
&+\int_{-1}^{+1} U_{ij}(\mathbf{s},\mathbf{x})R_A(\xi)J(\xi)\tilde{J}_e\big(\tilde{\xi}\big)\mathrm{d}S\big(\tilde{\xi}\big)\dot{\tilde{t}}_j^A
\end{aligned} \tag{50}$$

The above equation can be assembled to a matrix form, yielding the following form

$$\dot{\mathbf{H}}\mathbf{u} + \mathbf{H}\dot{\mathbf{u}} = \dot{\mathbf{G}}\mathbf{t} + \mathbf{G}\dot{\mathbf{t}}. \tag{51}$$

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 to the unknown field sensitivities $\dot{\mathbf{u}}$ and $\dot{\mathbf{t}}$.

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

$$\dot{u}_j(\mathbf{x}) = \dot{\bar{u}}_j(\mathbf{x}), \qquad \text{on } S_u, \tag{52}$$

$$\dot{t}_j(\mathbf{x}) = \dot{\bar{t}}_j(\mathbf{x}), \qquad \text{on } S_t, \tag{53}$$

where $\dot{\bar{u}}_j$ and $\dot{\bar{t}}_j$ are the displacement and traction sensitivity boundary conditions, respectively. The sensitivity boundary condition should be exerted using a nodal parameter extraction scheme similar to structural analysis procedure with IGABEM.

By swapping the unknowns in Eq. (51), a final matrix form is obtained as

$$\mathbf{A}\dot{\mathbf{z}} = \dot{\mathbf{f}} + \dot{\mathbf{A}}\mathbf{z} \tag{54}$$

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.

4.2 Sensitivities of fundamental solutions

As shown above, the sensitivities of fundamental solutions play an important role in implicit differentiation method. The analytical forms of the displacement and traction fundamental solution sensitivities $\dot{U}_{ij}$ and $\dot{T}_{ij}$ are derived by taking shape differentiation on Eqs. (19) and (20),

$$\dot{U}_{ij}(\mathbf{s},\mathbf{x}) = \frac{1}{8\pi\mu(1-\nu)}\left[(3-4\nu)\left(\dot{\ln\frac{1}{r}}\right)\delta_{ij} + (\dot{r_{,i}})r_{,j} + r_{,i}(\dot{r_{,j}})\right], \tag{55}$$

$$\begin{aligned}\dot{T}_{ij}(\mathbf{s},\mathbf{x}) = & -\frac{1}{4\pi(1-\nu)}\left(\dot{\frac{1}{r}}\right)\left\{\frac{\partial r}{\partial n}[(1-2\nu)\delta_{ij} + 2r_{,i}r_{,j}]\right\} \\ & -\frac{1}{4\pi(1-\nu)}\left(\dot{\frac{1}{r}}\right)[-(1-2\nu)(r_{,i}n_j - r_{,j}n_i)] \\ & -\frac{1}{4\pi(1-\nu)r}\left\{\left(\dot{\frac{\partial r}{\partial n}}\right)[(1-2\nu)\delta_{ij} + 2r_{,i}r_{,j}] - 2\frac{\partial r}{\partial n}\big[(\dot{r_{,i}})r_{,j} + r_{,i}(\dot{r_{,j}})\big] - (1- \right. \\ & \left. 2\nu)\big[(\dot{r_{,i}})n_j + r_{,i}\dot{n}_j - (\dot{r_{,j}})n_i - r_{,j}\dot{n}_i\big]\right\},\end{aligned} \tag{56}$$

where the quantities superposed by $\dot{}$ mean taking derivatives with respect to the given design variable, and

$$\left(\dot{\frac{\partial r}{\partial n}}\right) = (\dot{r_{,i}n_i}) = (\dot{r_{,i}})n_i + r_{,i}\dot{n}_i, \quad \left(\dot{\frac{1}{r}}\right) = -\frac{\dot{r}}{r^2}, \tag{57}$$

$$(\dot{r_{,i}}) = \left(\dot{\frac{x_i - s_i}{r}}\right) = \frac{(\dot{x}_i - \dot{s}_i)r - (x_i - s_i)\dot{r}}{r^2}, \quad \left(\dot{\ln\frac{1}{r}}\right) = -\frac{\dot{r}}{r}. \tag{58}$$

The Jacobian is

$$J(\xi) = \sqrt{J_i(\xi)J_i(\xi)}, \tag{59}$$

with

$$J_i(\xi) = \frac{\mathrm{d}x_i}{\mathrm{d}\xi}. \tag{60}$$

The shape derivative of Jacobian is given by

$$\dot{J}(\xi) = \frac{\dot{J}_i(\xi)J_i(\xi)}{J(\xi)}. \tag{61}$$

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

$$\dot{n}_i = \left[\frac{J_i(\dot{\xi})}{J(\xi)}\right] = \frac{\dot{J}_i(\xi)J(\xi) - J_i(\xi)\dot{J}(\xi)}{J^2(\xi)}. \tag{62}$$

The shape derivatives of hypersingular fundamental solutions are

$$\begin{aligned}\dot{D}_{kij}(\mathbf{s},\mathbf{x}) &= \frac{1}{4\pi(1-\nu)}\left(\frac{\dot{1}}{r}\right)[(1-2\nu)(r_{,i}\delta_{jk}+r_{,j}\delta_{ki}-r_{,k}\delta_{ij})+2r_{,i}r_{,j}r_{,k}]\\ &+\frac{1}{4\pi(1-\nu)r}\big\{\big[(1-2\nu)\big[(\dot{r}_{,i})\delta_{jk}+(\dot{r}_{,j})\delta_{ki}-(\dot{r}_{,k})\delta_{ij}\big)\big]\\ &+2\big[(\dot{r}_{,i})r_{,j}r_{,k}+r_{,i}(\dot{r}_{,j})r_{,k}+r_{,i}r_{,j}(\dot{r}_{,k})\big]\big\}, \end{aligned} \tag{63}$$

$$\begin{aligned}\dot{S}_{kij}(\mathbf{s},\mathbf{x}) &= \frac{\mu}{2\pi(1-\nu)}\left(\frac{\dot{1}}{r^2}\right)\Big\{2\frac{\partial r}{\partial n}[(1-2\nu)\delta_{ij}r_{,k}+\nu(r_{,j}\delta_{ik}+r_{,i}\delta_{jk})-4r_{,i}r_{,j}r_{,k}]+\\ &2\nu(n_ir_{,j}r_{,k}+n_jr_{,i}r_{,k})+(1-2\nu)(2n_kr_{,i}r_{,j}+n_j\delta_{ik}+n_i\delta_{jk})-(1-4\nu)n_k\delta_{ij}\Big\}+\\ &\frac{\mu}{2\pi(1-\nu)r^2}\Big\{2\left(\frac{\dot{\partial r}}{\partial n}\right)[(1-2\nu)\delta_{ij}r_{,k}+\nu(r_{,j}\delta_{ik}+r_{,i}\delta_{jk})-4r_{,i}r_{,j}r_{,k}]+\\ &2\left(\frac{\partial r}{\partial n}\right)\Big\{\big[(1-2\nu)\delta_{ij}(\dot{r}_{,k})+\nu\big[(\dot{r}_{,j})\delta_{ik}+(\dot{r}_{,i})\delta_{jk}\big]-4\big[(\dot{r}_{,i})r_{,j}r_{,k}+r_{,i}(\dot{r}_{,j})r_{,k}+\\ &r_{,i}r_{,j}(\dot{r}_{,k})\big]\big]\Big\}+2\nu\big[\dot{n}_ir_{,j}r_{,k}+n_i(\dot{r}_{,j})r_{,k}+n_ir_{,j}(\dot{r}_{,k})+\dot{n}_jr_{,i}r_{,k}+n_j(\dot{r}_{,j})r_{,k}+\\ &n_jr_{,i}(\dot{r}_{,k})\big]+(1-2\nu)\big(2\dot{n}_kr_{,i}r_{,j}+2n_k(\dot{r}_{,i})r_{,j}+2n_kr_{,i}(\dot{r}_{,j})+\dot{n}_j\delta_{ik}+\dot{n}_i\delta_{jk}\big)-\\ &(1-4\nu)\dot{n}_k\delta_{ij}\Big\}. \end{aligned} \tag{64}$$

Table 1 shows the singularity order of the fundamental solution sensitivities, where we can find that they have the same order as fundamental solutions.

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

Table 1. The singularity of kernel function sensitivities

4.3 Stress and displacement shape sensitivity recovery

4.3.1 Evaluate sensitivities at interior points

After getting the displacement and traction of the boundary by solving Eq.(31), we can evaluate the displacement or stress in the domain if necessary. The displacement and the stress of the interior point is a straightforward use of Somigliana's identities. Ignoring the body force, for interior displacement the expression is

$$\begin{aligned}u_i(\mathbf{S}) &= \int_{\tilde{S}} U_{ij}(\mathbf{S},\tilde{\boldsymbol{\xi}})N_{ea}t_j^{ea}(\tilde{\boldsymbol{\xi}})J(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\\ &-\int_{\tilde{S}} T_{ij}(\mathbf{S},\tilde{\boldsymbol{\xi}})N_{ea}u_j^{ea}(\tilde{\boldsymbol{\xi}})J(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}), \end{aligned} \tag{65}$$

and for interior stress the expression is

$$\begin{aligned}\sigma_{ij}(\mathbf{S}) &= \int_{\tilde{S}} D_{kij}(\mathbf{S},\tilde{\boldsymbol{\xi}})N_{ea}t_k^{ea}(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\\ &-\int_{\tilde{S}} S_{kij}(\mathbf{s},\tilde{\boldsymbol{\xi}})N_{ea}u_k^{ea}(\tilde{\boldsymbol{\xi}})J(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}). \end{aligned} \tag{66}$$

The shape sensitivity at the interior points can be obtained by taking shape derivatives for the Somigliana's identities. So the displacement shape sensitivity is

$$\begin{aligned}\dot{u}_i(\mathbf{S}) &= \int_S \dot{U}_{ij}(\mathbf{S},\mathbf{x})t_j(\mathbf{x})\mathrm{d}S(\mathbf{x}) + \int_\Omega U_{ij}(\mathbf{S},\mathbf{x})\dot{t}_j(\mathbf{x})\mathrm{d}S(\mathbf{x}) \\ &+ \int_\Omega U_{ij}(\mathbf{S},\mathbf{x})t_j(\mathbf{x})[\mathrm{d}\dot{S}(\mathbf{x})] - \int_S \dot{T}_{ij}(\mathbf{S},\mathbf{x})u_j(\mathbf{x})\mathrm{d}S(\mathbf{x}) \\ &- \int_S T_{ij}(\mathbf{S},\mathbf{x})\dot{u}_j(\mathbf{x})\mathrm{d}S(\mathbf{x}) - \int_S T_{ij}(\mathbf{S},\mathbf{x})u_j(\mathbf{x})[\mathrm{d}\dot{S}(\mathbf{x})], \end{aligned} \tag{67}$$

and stress shape sensitivity is

$$\begin{aligned}\dot{\sigma}_{ij}(\mathbf{S}) &= \int_S \dot{D}_{kij}(\mathbf{S},\mathbf{x})t_k(\mathbf{x})\mathrm{d}S(\mathbf{x}) + \int_S D_{kij}(\mathbf{S},\mathbf{x})\dot{t}_k(\mathbf{x})\mathrm{d}S(\mathbf{x}) \\ &+ \int_S D_{kij}(\mathbf{S},\mathbf{x})t_k(\mathbf{x})[\mathrm{d}\dot{S}(\mathbf{x})] - \int_S \dot{S}_{kij}(\mathbf{S},\mathbf{x})u_k(\mathbf{x})\mathrm{d}S(\mathbf{x}) \\ &- \int_S S_{kij}(\mathbf{S},\mathbf{x})\dot{u}_k(\mathbf{x})\mathrm{d}S(\mathbf{x}) - \int_S S_{kij}(\mathbf{S},\mathbf{x})u_k(\mathbf{x})[\mathrm{d}\dot{S}(\mathbf{x})]. \end{aligned} \tag{68}$$

4.4 Evaluate stress sensitivities at boundary points

Evaluation of the stress at boundary points can also employ the somigliana's equations, but a singular integral needs to be treated and a integral surface is time-consuming. So an easy and efficient way is to recover the stress by Hooke's law and Cauchy's formula from the displacement, displacement gradient and traction

$$\mathbf{u}(\tilde{\xi}) = N_{ea}(\tilde{\xi})\mathbf{u}^{ea}, \tag{69}$$

$$\frac{\mathrm{d}\mathbf{u}(\tilde{\xi})}{\mathrm{d}\tilde{\xi}} = \frac{\mathrm{d}N_{ea}(\tilde{\xi})}{\mathrm{d}\tilde{\xi}}\mathbf{u}^{ea}, \tag{70}$$

$$\mathbf{t}(\tilde{\xi}) = N_{ea}(\tilde{\xi})\mathbf{t}^{ea}. \tag{71}$$

As shown in Fig. 8, 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, and the vectors in this system can be represented as

$$\hat{\mathbf{x}} = \hat{x}_1\hat{\mathbf{e}}_1 + \hat{x}_2\hat{\mathbf{e}}_2. \tag{72}$$

The local tangential vector can be obtained by

$$\hat{\mathbf{e}}_1 = \mathbf{n}, \tag{73}$$

$$\hat{\mathbf{e}}_2 = \frac{\mathbf{m}}{|\mathbf{m}|}, \tag{74}$$

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

$$\mathbf{m} = \frac{\mathrm{d}\mathbf{x}(\tilde{\xi})}{\mathrm{d}\tilde{\xi}}. \tag{75}$$

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

$$A = \begin{bmatrix} \hat{\mathbf{e}}_1 \\ \hat{\mathbf{e}}_2 \end{bmatrix} = \begin{bmatrix} n_1 & n_2 \\ -n_2 & n_1 \end{bmatrix}. \tag{76}$$

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, then $\hat{\epsilon}_{22}$ can be evaluated through the displacement gradient in global coordinates,

$$\hat{\epsilon}_{22}(\tilde{\xi}) = \hat{u}_{2,2}(\tilde{\xi}) = \frac{\partial \hat{u}_2}{\partial \tilde{\xi}}\frac{\partial \tilde{\xi}}{\partial \hat{x}_2} = A_{2j}\frac{\partial u_j}{\partial \tilde{\xi}}\frac{\partial \tilde{\xi}}{\partial \hat{x}_2}, \tag{77}$$

with

$$\frac{\partial\tilde{\xi}}{\partial\hat{x}_2}=\frac{1}{|\mathbf{m}|}. \tag{78}$$

The stress tensor in the local coordinate system is

$$\hat{\sigma}_{11} = \hat{t}_1, \tag{79}$$
$$\hat{\sigma}_{12} = \hat{t}_2, \tag{80}$$
$$\hat{\sigma}_{22} = \left(\frac{E}{1-\nu^2}\right)\hat{\epsilon}_{22}+\left(\frac{\nu}{1-\nu}\right)\hat{t}_1. \tag{81}$$

Finally, the stress in the global cartesian coordinate system can be obtained

$$\sigma_{ij}=A_{ki}A_{nj}\hat{\sigma}_{kn}. \tag{82}$$

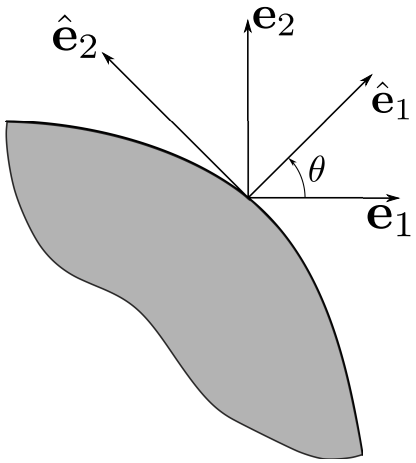

Figure 8. Local coordinate system on curve

Now we consider stress sensitivities. After solving Eq. (54), we can get

$$\dot{\mathbf{u}}(\xi)=N_{ea}(\xi)\dot{\mathbf{u}}^{ea}, \tag{83}$$

$$\frac{\mathrm{d}\dot{\mathbf{u}}(\xi)}{\mathrm{d}\xi}=\frac{\mathrm{d}N_{ea}(\xi)}{\mathrm{d}\xi}\dot{\mathbf{u}}^{ea}, \tag{84}$$

$$\dot{\mathbf{t}}(\xi)=N_{ea}(\xi)\dot{\mathbf{t}}^{ea}. \tag{85}$$

And the sensitivity of tangential vector $\mathbf{m}$ is

$$\dot{\mathbf{m}}=\left(\frac{\mathrm{d}\dot{\mathbf{x}}(\xi)}{\mathrm{d}\xi}\right), \tag{86}$$

and

$$|\dot{\mathbf{m}}|=\left(\dot{\sqrt{\frac{\mathrm{d}x_i}{\mathrm{d}\xi}\frac{\mathrm{d}x_i}{\mathrm{d}\xi}}}\right)=\frac{\left(\frac{\dot{\mathrm{d}x_i}}{\mathrm{d}\xi}\right)\frac{\mathrm{d}x_i}{\mathrm{d}\xi}}{|\mathbf{m}|^2}, \tag{87}$$

with

$$\left(\frac{\mathrm{d}\dot{\mathbf{x}}(\xi)}{\mathrm{d}\xi}\right)=\frac{\mathrm{d}N_{ea}(\xi)}{\mathrm{d}\xi}\dot{\mathbf{x}}^{ea}. \tag{88}$$

The normalized local tangential vector

$$\dot{\hat{\mathbf{e}}}_1 = \left(\frac{\dot{\mathbf{m}}}{|\mathbf{m}|}\right)=-\frac{|\dot{\mathbf{m}}|}{|\mathbf{m}|^2}, \tag{89}$$
$$\dot{\hat{\mathbf{e}}}_2 = \dot{\mathbf{n}}. \tag{90}$$

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

$$\begin{aligned}\dot{\hat{\epsilon}}(\xi) &= \dot{\hat{u}}_{1,1}(\xi) = \left(\frac{\partial \hat{u}_1}{\partial \xi}\dot{\frac{\partial \xi}{\partial \hat{x}_1}}\right) = \left(A_{i1}\dot{\frac{\partial u_1}{\partial \xi}}\frac{\partial \xi}{\partial \hat{x}_1}\right) \\ &= (\dot{A_{i1}})\frac{\partial u_1}{\partial \xi}\frac{\partial \xi}{\partial \hat{x}_1} + A_{i1}\left(\dot{\frac{\partial u_1}{\partial \xi}}\right)\frac{\partial \xi}{\partial \hat{x}_1} + A_{i1}\frac{\partial u_1}{\partial \xi}\left(\dot{\frac{\partial \xi}{\partial \hat{x}_1}}\right),\end{aligned} \tag{91}$$

with

$$\left(\dot{\frac{\partial \xi}{\partial \hat{x}_1}}\right) = \left(\dot{\frac{1}{|\mathbf{m}_1|}}\right). \tag{92}$$

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

$$\dot{A} = \begin{bmatrix} \dot{\hat{\mathbf{e}}}_1 \\ \dot{\hat{\mathbf{e}}}_2 \end{bmatrix} = \begin{bmatrix} n_1 & n_2 \\ -n_2 & n_1 \end{bmatrix}. \tag{93}$$

The stress sensitivity tensor in the local coordinate system is

$$\dot{\hat{\sigma}}_{11} = \left(\frac{E}{1-\nu^2}\right)\dot{\hat{\epsilon}}_{11} + \left(\frac{\nu}{1-\nu}\right)\dot{\hat{t}}_2, \tag{94}$$

$$\dot{\hat{\sigma}}_{22} = \dot{\hat{t}}_2, \tag{95}$$

$$\dot{\hat{\sigma}}_{12} = \dot{\hat{t}}_1. \tag{96}$$

Transferring the stress sensitivity back to the global cartesian coordinate system generates

$$\begin{aligned}\dot{\sigma}_{ij} &= \left(A_{ki}\dot{A_{nj}}\hat{\sigma}_{kn}\right) \\ &= \left(\dot{A_{ki}}\right)A_{nj}\hat{\sigma}_{kn} + A_{ki}(\dot{A_{nj}})\hat{\sigma}_{kn} + A_{ki}A_{nj}(\dot{\hat{\sigma}_{kn}}).\end{aligned} \tag{97}$$

5 Shape optimization with IGABEM

Shape optimization can be conducted through a gradient-less or gradient-based method. The gradient-less shape optimization is without needing to evaluate the shape derivatives, but normally prohibitively time-consuming and not supported by a mathematical theory. So a gradient-based method is normally preferred and also used in the present work. The gradient-based shape optimization has a well-sounded mathematical foundation rooted in optimal-control theory. A shape optimization problem can be formulated as minimizing an objective function

$$f(\mathbf{t}) \qquad (\mathbf{t} \in R^n), \tag{98}$$

subject to the constraints

$$g_i(\mathbf{t}) \leqslant 0, \quad \text{for } i = 1, ..., m, \tag{99}$$

$$t_i^l \leqslant t_i \leqslant t_i^u. \tag{100}$$

where $\mathbf{t}$ is a vector of parameters which controls geometry configurations, also called design variables. $f(\mathbf{t})$ is the objective function, $g_i(\mathbf{t})$ the constraint functions, i the constraint function index, m the number of constraints. Eq. (100) is side constraints to limit the search region for the optimum, where t_i^l and t_i^u are lower and upper bounds of design variables, respectively. A design is called feasible if all constraints are satisfied.

To find the minimum value, numerical optimization algorithms employ the gradient of objective functions and constraint functions to find the next value in iterative steps, *i.e.*

$$\left(f_k, g_i^k, \frac{\mathrm{d}}{\mathrm{d}\mathbf{t}}f_k, \frac{\mathrm{d}}{\mathrm{d}\mathbf{t}}g_i^k\right) \rightarrow \left(f_{k+1}, g_i^{k+1}\right), \tag{101}$$

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

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

From Fig. (9), we can find that the steps of meshing/remeshing has been removed in IGABEM optimization.

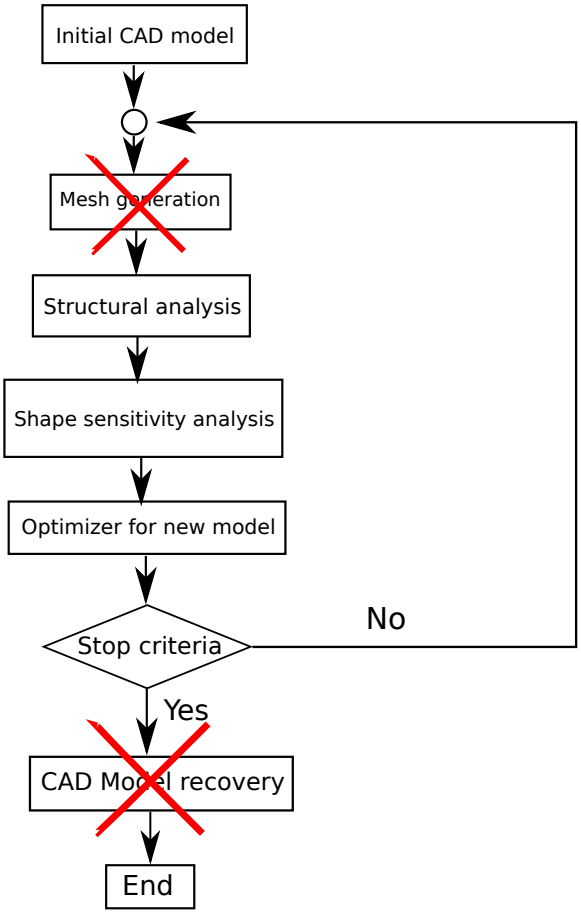

Figure 9. IGABEM shape optimization flowchart

5.1 Shape derivatives of volume and von Mises stress

The displacement and stress shape sensitivities can be obtained from the procedure demonstrated in Section 4. However, a bit more effort is needed to calculate the sensitivities of some other commonly used quantities. To be consistent with our CAD and analysis model, the domain integrals involved should be transformed to boundary integral forms.

- The shape derivatives of area A. The area can be transfered into boundary integral readily by using divergence theorem

$$A = \int_\Omega \mathrm{d}\Omega = \frac{1}{2}\int_\Omega \nabla \cdot \mathbf{x}\,\mathrm{d}\Omega = \frac{1}{2}\int_\Gamma \mathbf{x}\cdot\mathbf{n}\,\mathrm{d}\Gamma = \frac{1}{2}\int_\Gamma \mathbf{x}\cdot\mathbf{n} J(\tilde{\boldsymbol{\xi}})\mathbf{x}\mathrm{d}\Gamma(\tilde{\boldsymbol{\xi}}). \tag{102}$$

So the shape derivatives are

$$\begin{aligned}\dot{A} &= \frac{1}{2}\int_{\Gamma}\left[\mathbf{x}\cdot\mathbf{n}J\left(\tilde{\boldsymbol{\xi}}\right)\right]\mathrm{d}\Gamma\left(\tilde{\boldsymbol{\xi}}\right)\\ &= \frac{1}{2}\int_{\Gamma}\left[\dot{\mathbf{x}}\cdot\mathbf{n}J\left(\tilde{\boldsymbol{\xi}}\right)+\mathbf{x}\cdot\dot{\mathbf{n}}J\left(\tilde{\boldsymbol{\xi}}\right)+\mathbf{x}\cdot\mathbf{n}\dot{J}\left(\tilde{\boldsymbol{\xi}}\right)\right]\mathrm{d}\Gamma\left(\tilde{\boldsymbol{\xi}}\right).\end{aligned} \tag{103}$$

- The shape derivatives of von Mises stress σ_{vm}. The expression of σ_{vm} is given by

$$\sigma_{\mathrm{vm}}=\left(\frac{3}{2}s_{ij}s_{ij}\right)^{\frac{1}{2}}, \tag{104}$$

with s_{ij} the components of the stress deviator tensor, which is given by

$$s_{ij}=\sigma_{ij}-\frac{1}{3}\sigma_{kk}\delta_{ij}. \tag{105}$$

In two-dimensional problems, von Mises stress can be written as

$$\sigma_{\mathrm{vm}}=\sqrt{\sigma_{11}^2+\sigma_{22}^2+3\sigma_{12}^2-\sigma_{11}\sigma_{22}}, \tag{106}$$

and its sensitivity is given by

$$\dot{\sigma}_{\mathrm{vm}}=\frac{(2\sigma_{11}-\sigma_{22})\dot{\sigma}_{11}+(2\sigma_{22}-\sigma_{11})\dot{\sigma}_{22}+6\sigma_{12}\dot{\sigma}_{12}}{2\sigma_{\mathrm{vm}}}. \tag{107}$$

5.2 Shape sensitivity transition in NURBS

Numerical analysis always requires a very refined control mesh to achieve sufficient accuracy. In contrast, a relative coarse 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 in sensitivity analysis, and coarse meshes for model design and optimization, we propose a method to evaluate the shape derivatives of the quantities in refined meshes with respect to the design variables in the meshes before refinement.

Recall the knot insertion algorithm in NURBS for adding new control points while keeping the geometry unchanged,

$$\bar{\tilde{\mathbf{P}}}_A=\begin{cases}\tilde{\mathbf{P}}_1 & A=1\\ \alpha_A\tilde{\mathbf{P}}_A+(1-\alpha_A)\tilde{\mathbf{P}}_{A-1} & 1<A<m\\ \tilde{\mathbf{P}}_n & A=m\end{cases} \tag{108}$$

with

$$\alpha_A=\begin{cases}1 & 1\leqslant A\leqslant k-p\\ \dfrac{\bar{\xi}_A-\xi_A}{\xi_{A+p}-\xi_A} & k-p+1\leqslant A\leqslant k\\ 0 & A\geqslant k+1\end{cases} \tag{109}$$

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 weighted point $\bar{\tilde{\mathbf{P}}}_A$ in the refined mesh can be obtained through taking derivatives in Eq. (108)

$$\dot{\bar{\tilde{\mathbf{P}}}}_A=\begin{cases}\dot{\tilde{\mathbf{P}}}_1, & A=1,\\ \alpha_A\dot{\tilde{\mathbf{P}}}_A+(1-\alpha_A)\dot{\tilde{\mathbf{P}}}_{A-1}, & 1<A<m,\\ \dot{\tilde{\mathbf{P}}}_n, & A=m.\end{cases} \tag{110}$$

After that, the control point derivatives $\dot{\mathbf{P}}_A$ is obtained by dividing the weights of $\dot{\tilde{\mathbf{P}}}_A$. Now the shape derivatives transited from coarse mesh to refined mesh have been completed. Through this approach, the shape sensitivity analysis mesh has been separated from the design mesh. It should be noted that analysis and the geometry design still share the same model, only in the different level of refinement.

6 Numerical examples

We will investigate the performance of IGABEM for sensitivity analysis and shape optimization through some numerical examples with closed-form solutions. All the geometries are modeled using NURBS. To study the analysis convergence, we define the error of structural analysis as

$$e(\mathbf{u}_h) = \frac{\|\mathbf{u}_h - \mathbf{u}\|_{L_2}}{\|\mathbf{u}\|_{L_2}}, \tag{111}$$

and that of shape sensitivity analysis as

$$e(\dot{\mathbf{u}}_h) = \frac{\|\dot{\mathbf{u}}_h - \dot{\mathbf{u}}\|_{L_2}}{\|\dot{\mathbf{u}}\|_{L_2}}, \tag{112}$$

with

$$\|\mathbf{u}\|_{L_2} = \sqrt{\int_\Gamma (\mathbf{u}\cdot\mathbf{u})\mathrm{d}\Gamma}, \tag{113}$$

and

$$\|\dot{\mathbf{u}}\|_{L_2} = \sqrt{\int_\Gamma (\dot{\mathbf{u}}\cdot\dot{\mathbf{u}})\mathrm{d}\Gamma}. \tag{114}$$

And the shape sensitivity transition technique is employed. The optimization solver is taking the method of moving asymptotes (MMA) [41].

6.1 Shape sensitivity analysis examples

6.1.1 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 $\nu = 0.3$. The analytical displacement and stress in polar coordinates (r, θ) are given by

$$u_r(r,\theta) = \frac{pa^2}{E(b^2-a^2)}\left[(1-\nu)r + \frac{b^2(1+\nu)}{r}\right], \tag{115}$$

$$\sigma_{rr}(r,\theta) = \frac{pb^2}{b^2-a^2}\left(1-\frac{b^2}{r^2}\right), \tag{116}$$

$$\sigma_{\theta\theta}(r,\theta) = \frac{pb^2}{b^2-a^2}\left(1+\frac{b^2}{r^2}\right). \tag{117}$$

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

$$\begin{aligned}\dot{u}_r(r,\theta) = &-\frac{2Pa^2b}{E(b^2-a^2)^2}\left[(1-\nu)r + \frac{b^2(1+\nu)}{r}\right] \\ &+\frac{Pa^2}{E(b^2-a^2)}\left[(1-\nu)\dot{r} + (1+\nu)\frac{2br - b^2\dot{r}}{r^2}\right],\end{aligned} \tag{118}$$

$$\dot{\sigma}_{rr}(r,\theta) = \frac{-2a^2bP}{(b^2-a^2)^2}\left(1-\frac{b^2}{r^2}\right)+\frac{Pa^2}{b^2-a^2}\left(\frac{2br^2-2b^2r\,\dot{r}}{r^4}\right), \tag{119}$$

$$\dot{\sigma}_{\theta\theta}(r,\theta) = \frac{-2a^2bP}{(b^2-a^2)^2}\left(1+\frac{b^2}{r^2}\right)+\frac{Pa^2}{b^2-a^2}\left(\frac{2br^2-2b^2r\,\dot{r}}{r^4}\right), \tag{120}$$

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

Because of the symmetry, only a quarter of the cylinder needs to be modeled as shown in Fig. 10. The geometry is constructed using quadratic NURBS and the minimum number of elements and control points to represent the geometry are shown in Fig. 11. The shape sensitivity analysis is performed using a refined mesh with 8 elements on each segment. The Figs. 12 and 13 show the IGABEM solutions of the displacement and stress sensitivities on the bottom edge AB. An excellent agreement with the analytical 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.*

$$\dot{r}=\frac{b-r}{b-a}, \tag{121}$$

the displacement and stress sensitivities can be evaluated using Eqs. (66) and (67). The numerical solutions are shown in Figs. 14 and 15, respectively. An excellent agreement with the analytical solution is seen again.

Fig. 16. illustrates the convergence of the errors $e(\mathbf{u}_h)$ and $e(\dot{\mathbf{u}}_h)$ against the degree of freedoms. Both the structural analysis and shape sensitivity analysis converge to the exact results. The reason that the error of shape sensitivity analysis is larger than structural analysis is because the numerical results from structural analysis will used in the shape sensitivity analysis, thus accumulating the errors.

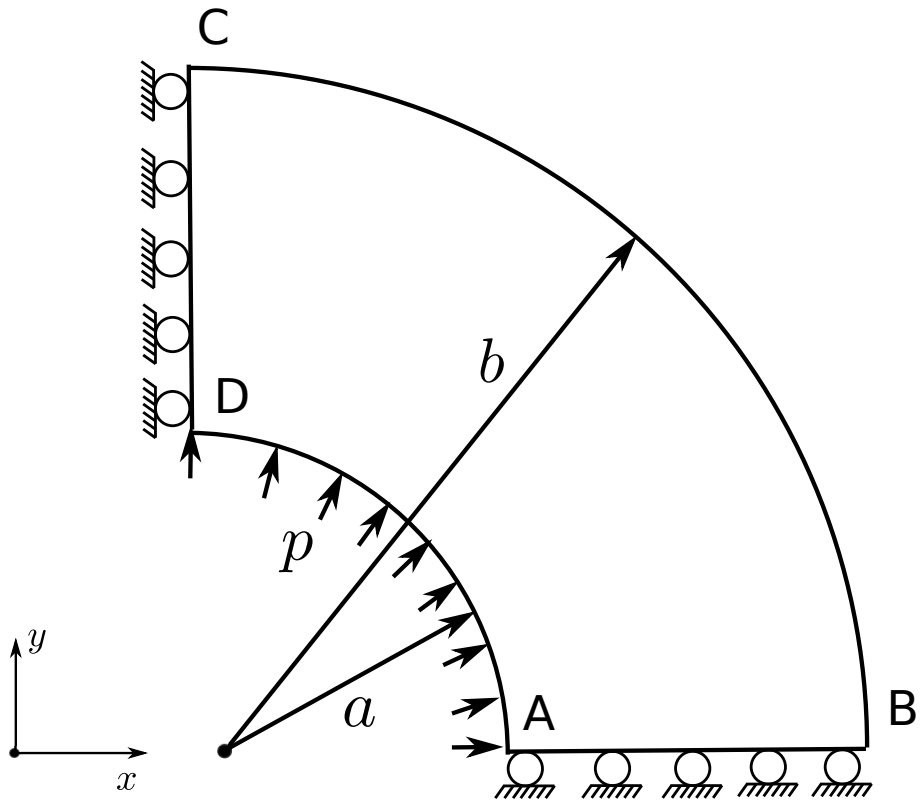

Figure 10. Definition of Lamé problem

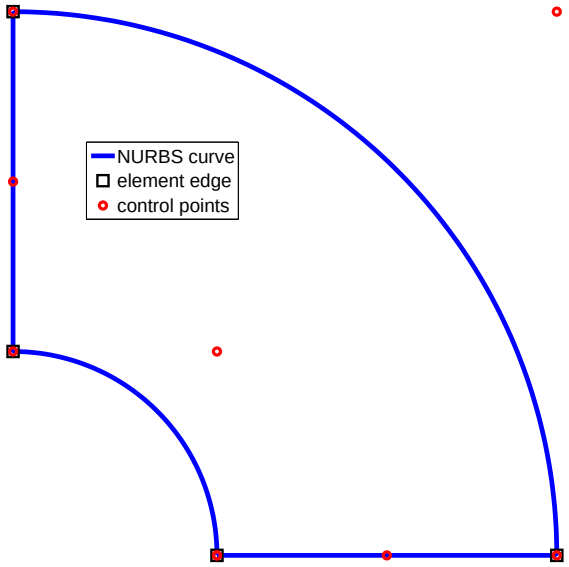

Figure 11. Geometric model of Lamé problem

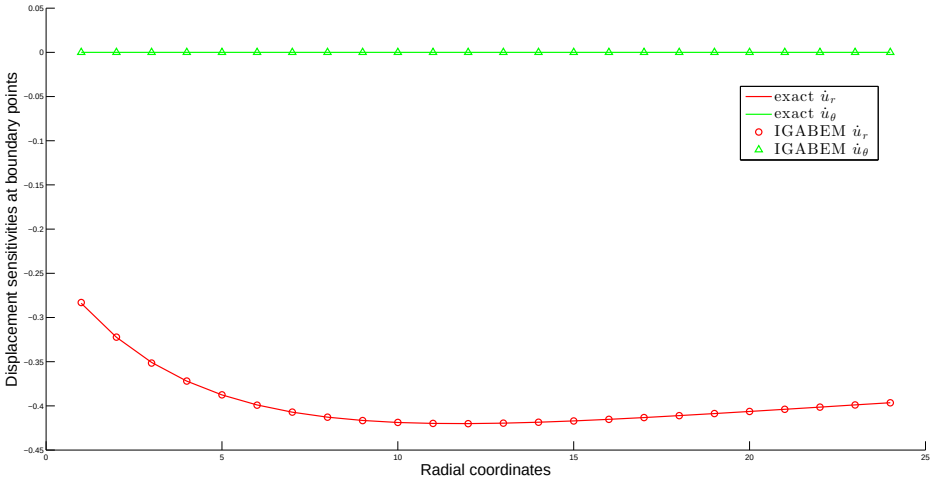

Figure 12. Displacement sensitivities on the boundary points for Lamé problem

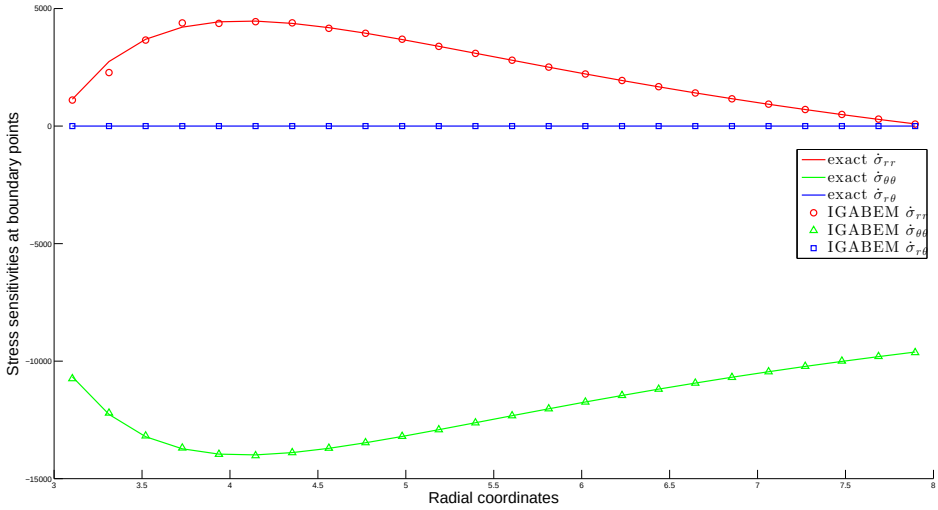

Figure 13. Stress sensitivities on the boundary points for Lamé problem

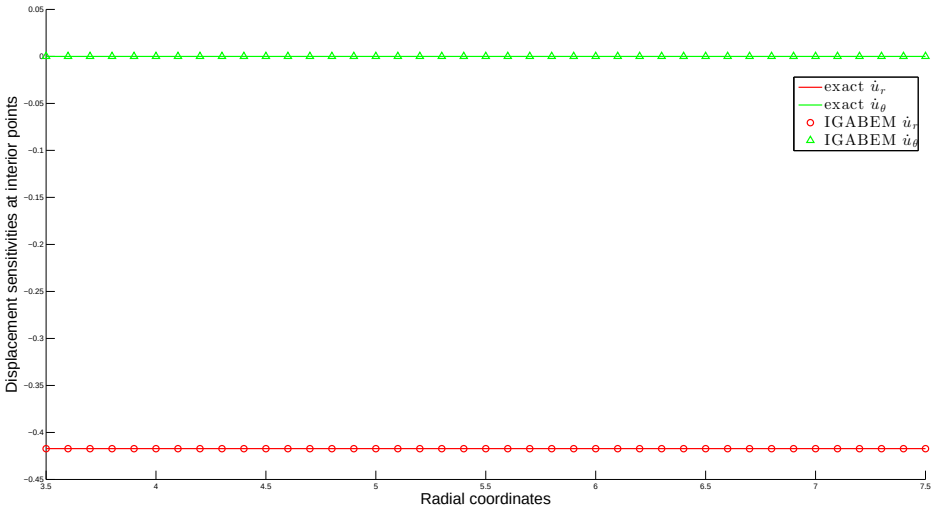

Figure 14. Displacement sensitivities at the interior points for Lamé problem

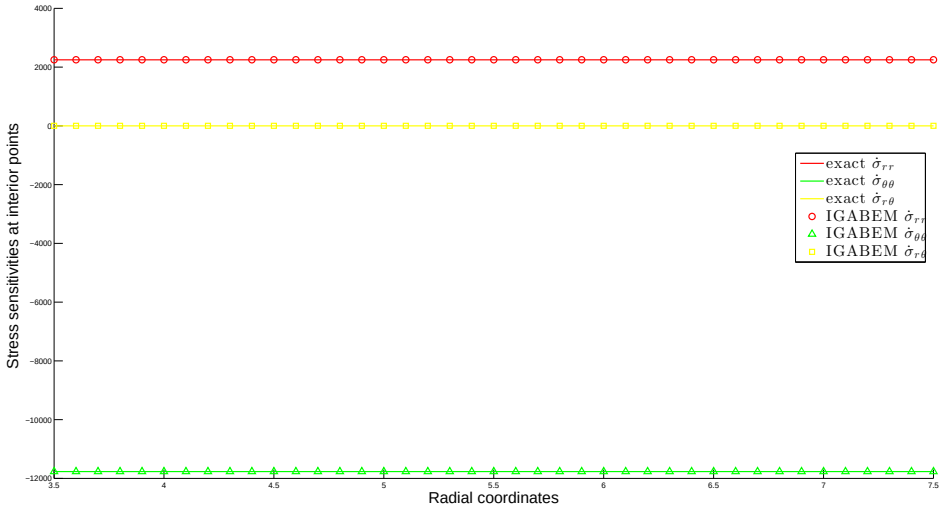

Figure 15. Stress sensitivities at the interior points for Lamé problem

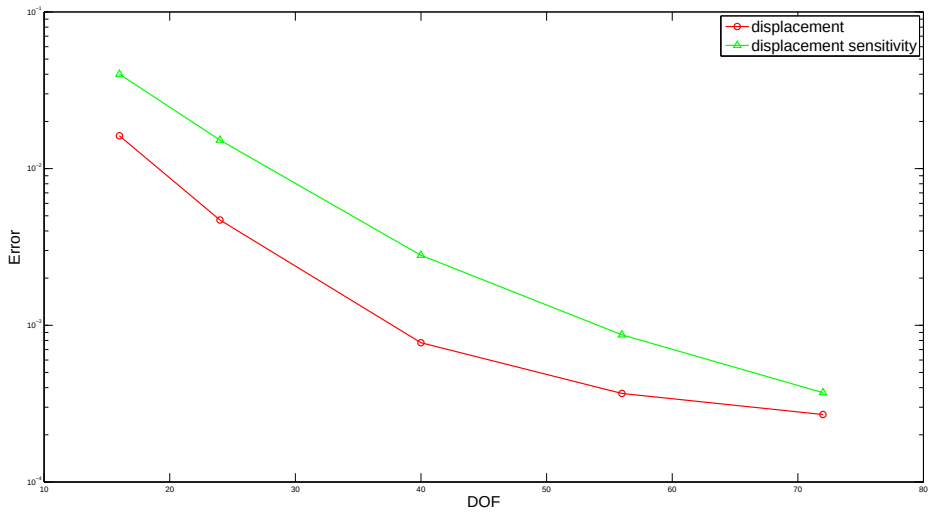

Figure 16. Relative error norm of displacement sensitivities for Lamé problem

6.1.2 Kirsch problem

The Kirsch problem is a infinitely large plate with a circular holde, subject to a far field uniform tension T. This problem can be modeled by extracting a finite domain and imposing he exact solutions as boundary conditions around the boundary. Due to the symmetry, only a quater of the plate is modeled, as shown in Fig. 17. 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 displacement and stress are given by

$$u_r = -\frac{Ta^2}{4Gr}\left\{(1+K)-(1-K)\left[4(1-\nu)-\frac{a^2}{r^2}\right]\cos\theta\right\}, \tag{122}$$

$$u_\theta = -\frac{Ta^2}{4Gr}\left\{(1-K)\left[2(1-2\nu)+\frac{a^2}{r^2}\right]\sin 2\theta\right\}, \tag{123}$$

and

$$\sigma_{rr}(r,\theta) = \frac{T}{2}\left(1-\frac{a^2}{r^2}\right)+\frac{T}{2}\left(1-4\frac{a^2}{r^2}+3\frac{a^4}{r^4}\right)\cos 2\theta, \tag{124}$$

$$\sigma_{\theta\theta}(r,\theta) = \frac{T}{2}\left(1+\frac{a^2}{r^2}\right)-\frac{T}{2}\left(1+3\frac{a^4}{r^4}\right)\cos 2\theta, \tag{125}$$

$$\sigma_{r\theta}(r,\theta) = -\frac{T}{2}\left(1+2\frac{a^2}{r^2}-3\frac{a^4}{r^4}\right)\sin 2\theta. \tag{126}$$

with

$$K = 3-4\nu. \tag{127}$$

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

$$\begin{aligned}\dot{u}_r = &-\frac{Ta^2}{4G}\left(\dot{\frac{1}{r}}\right)\left\{(1+K)-(1-K)\left[4(1-\nu)-\frac{a^2}{r^2}\right]\cos\theta\right\} \\ &-\frac{Ta^2}{4Gr}\left\{-(1-K)\left(\dot{\frac{a^2}{r^2}}\right)\cos\theta\right\},\end{aligned} \tag{128}$$

$$\begin{aligned}\dot{u}_\theta = &-\frac{Ta^2}{4G}\left(\dot{\frac{1}{r}}\right)\left\{(1-K)\left[2(1-2\nu)+\frac{a^2}{r^2}\right]\sin 2\theta\right\} \\ &-\frac{Ta^2}{4Gr}\left\{(1-K)\left[2(1-2\nu)+\frac{a^2}{r^2}\right]\sin 2\theta\right\},\end{aligned} \tag{129}$$

with

$$\begin{aligned}\left(\dot{\frac{1}{r}}\right) &= -\frac{\dot{r}}{r^2}, \\ \left(\dot{\frac{a^2}{r^2}}\right) &= \frac{2ar^2-2a^2r\dot{r}}{r^4}.\end{aligned} \tag{130}$$

The analytical stress sensitivities are

$$\begin{aligned}\dot{\sigma}_{rr}(r,\theta) = &-\frac{T}{2}\left(\dot{\frac{a^2}{r^2}}\right)+\frac{T}{2}\left[-4\left(\dot{\frac{a^2}{r^2}}\right)+3\left(\dot{\frac{a^4}{r^4}}\right)\right]\cos 2\theta \\ &+\frac{T}{2}\left(1-4\frac{a^2}{r^2}+3\frac{a^4}{r^4}\right)(\dot{\cos 2\theta}),\end{aligned} \tag{131}$$

$$\dot{\sigma}_{\theta\theta}(r,\theta) = \frac{T}{2}\left(\dot{\frac{a^2}{r^2}}\right)-\frac{3T}{2}\left(\dot{\frac{a^4}{r^4}}\right)\cos 2\theta-\frac{T}{2}\left(1+3\frac{a^4}{r^4}\right)(\dot{\cos 2\theta}), \tag{132}$$

$$\dot{\sigma}_{r\theta}(r,\theta) = -\frac{T}{2}\left[2\left(\dot{\frac{a^2}{r^2}}\right)-3\left(\dot{\frac{a^4}{r^4}}\right)\right]\sin 2\theta-\frac{T}{2}\left(1+2\frac{a^2}{r^2}-3\frac{a^4}{r^4}\right)(\dot{\sin 2\theta}), \tag{133}$$

with

$$\left(\dot{\frac{a^2}{r^2}}\right) = \frac{2ar^2-2a^2r\dot{r}}{r^4}, \tag{134}$$

$$\left(\frac{\dot{a^4}}{r^4}\right) = \frac{4a^3r^4 - 4a^4r^3\dot{r}}{r^8}. \tag{135}$$

Fig. 18 shows the NURBS geometry model with 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 each segment.

Figs. 19 and 20 show the IGABEM solutions of displacement and stress sensitivities on edge AB, respectively. Figs. 21 and 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 $\theta = 3\pi/4$. The domain points are assumed to be linearly varied in the radial direction, *i.e.*

$$\dot{r} = \begin{cases} \dfrac{L/\cos\theta - r}{L/\cos\theta - a}, & \text{for } \theta \geqslant \dfrac{3\pi}{4}, \\ \dfrac{L/\sin\theta - r}{L/\sin\theta - a}, & \text{for } \theta < \dfrac{3\pi}{4}. \end{cases} \tag{136}$$

The numerical solutions agree with the analytical solutions very well. And the convergence of the structural and sensitivity analysis solutions is shown in Fig. 23.

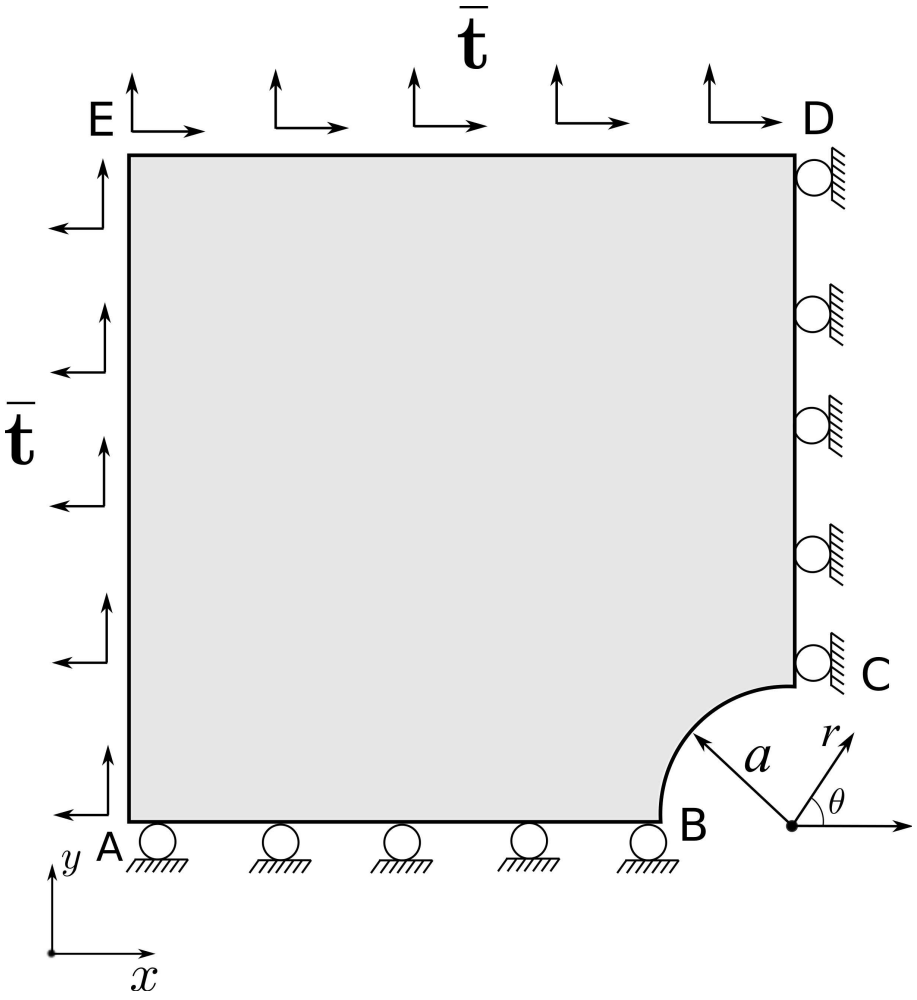

Figure 17. Definition of Kirsch problem

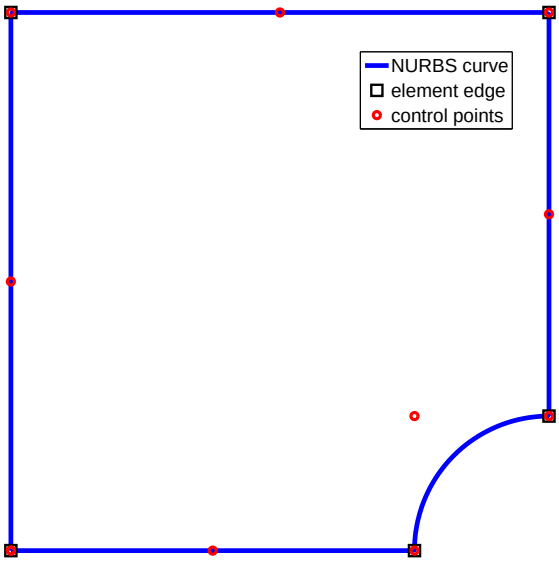

Figure 18. Geometric Model of Kirsch problem

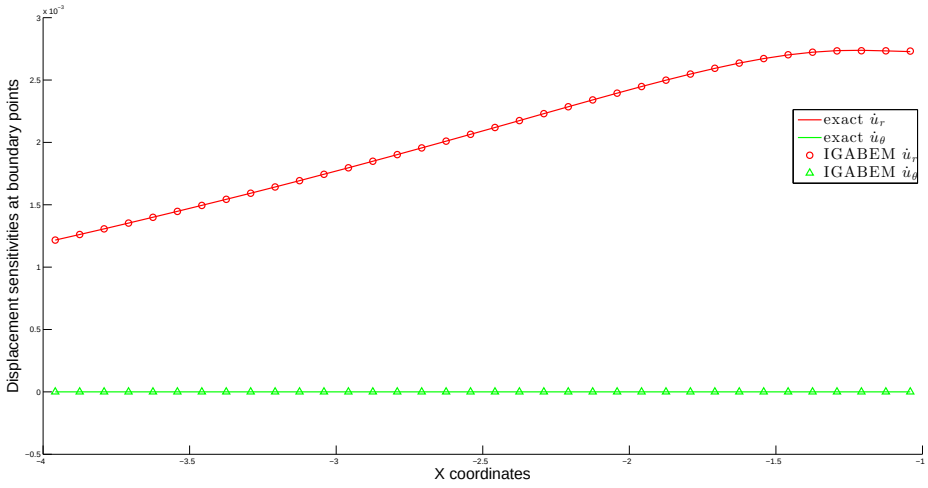

Figure 19. Displacement sensitivities on the edge AB of the plate

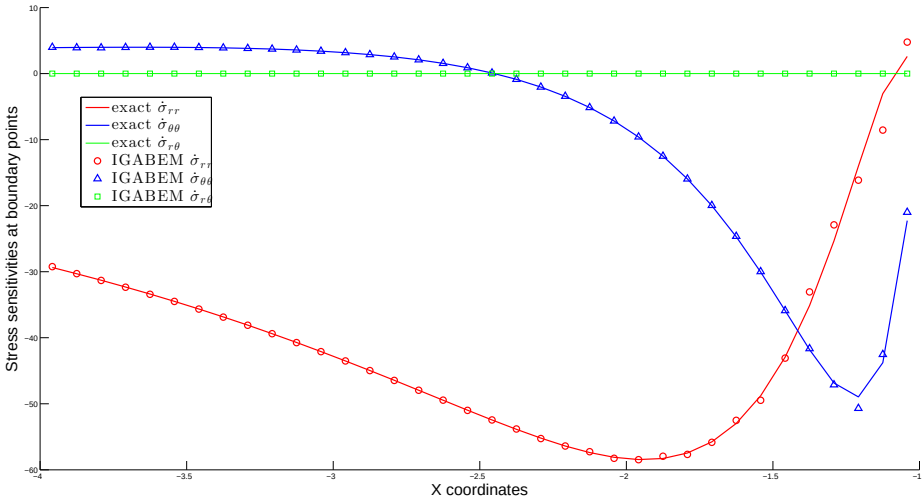

Figure 20. Stress sensitivities on the edge AB of the plate

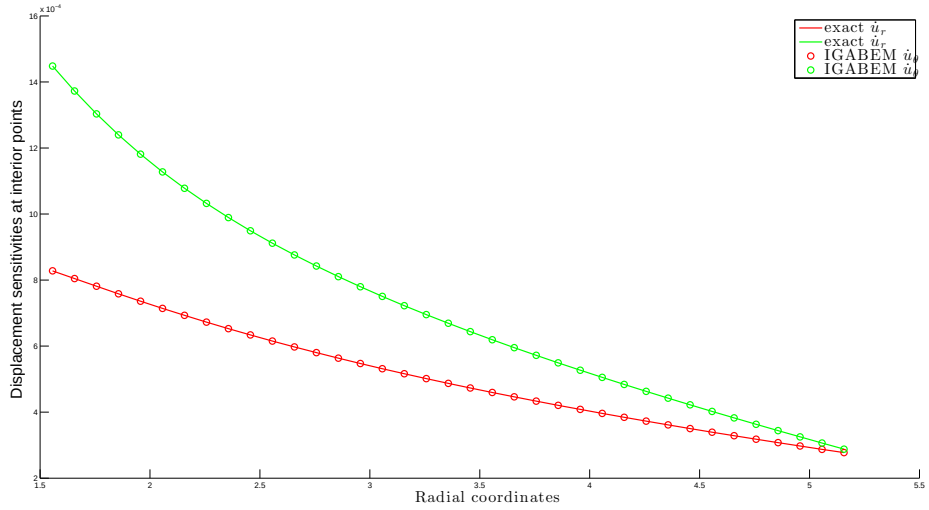

Figure 21. Displacement sensitivities at the interior points of the plate

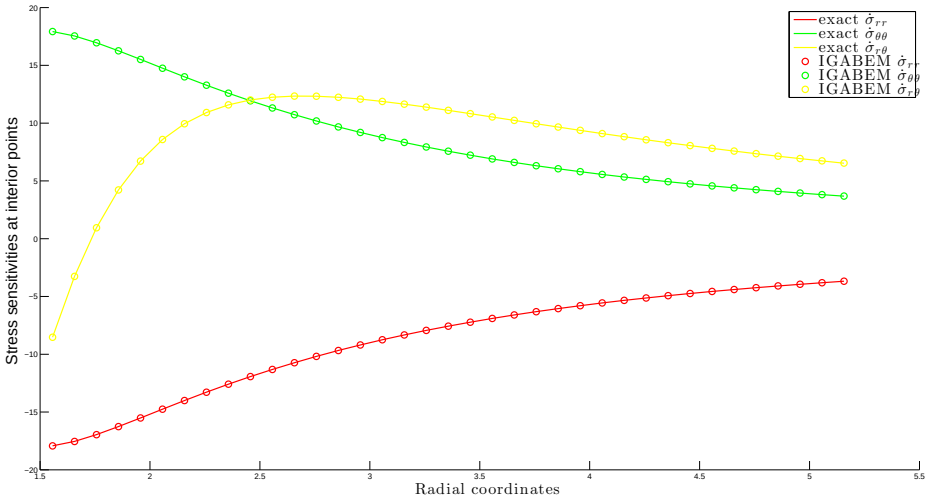

Figure 22. Stress sensitivities at the interior points of the plate

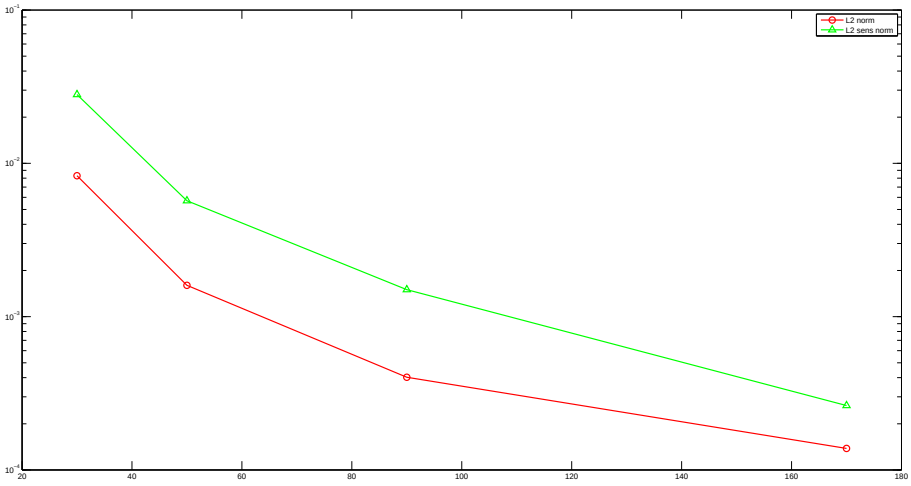

Figure 23. Relative error norm of sensitivity analysis for Kirsch problem

6.2 Shape optimization numerical examples

6.2.1 Cantilever beam

The problem is a cantilever beam subject to a traction $\bar{t} = 2$ at the end of the bottom (Fig. 24.).

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 $E=210\times10^3$ 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 12 control points and 8 elements, as shown in Fig. 25. The design variables are the vertical positions of the five control points on the beam top. The control points on the bottom are fixed during optimization, and that on the two sides will be set to vary linearly along the height of the side. The contraint is that the beam area should not be beyond $\hat{A}=210$. The side constraints can be seen in Table (2). The analysis mesh is refined from the design mesh and has 16 elements (Fig. 26). After an iteration procedure (Fig. 28), an optimal design is converged with the final geometry shown in Fig. 27. 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 Table (2), which construct a model which can be used immediately by the CAD.

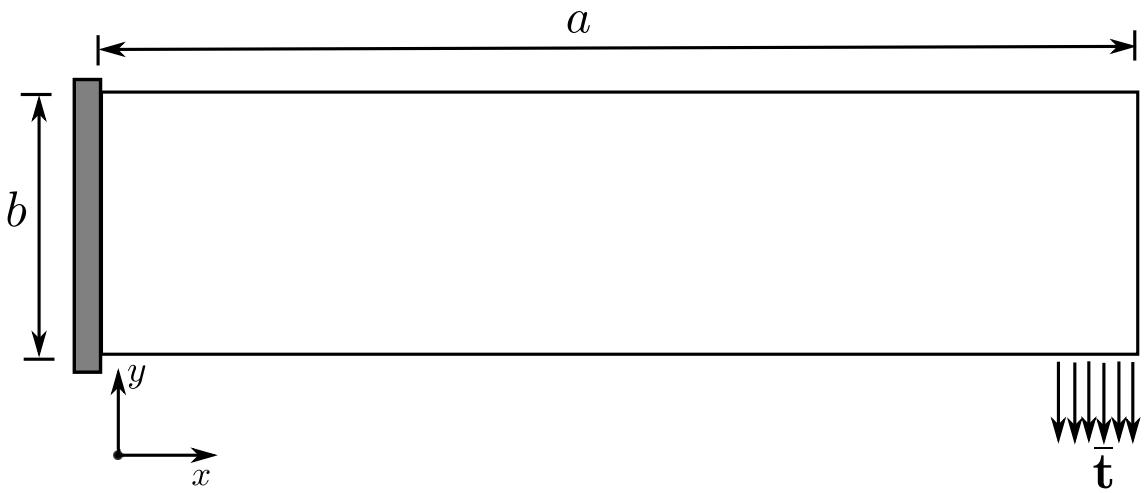

Figure 24. The definition of cantilever beam

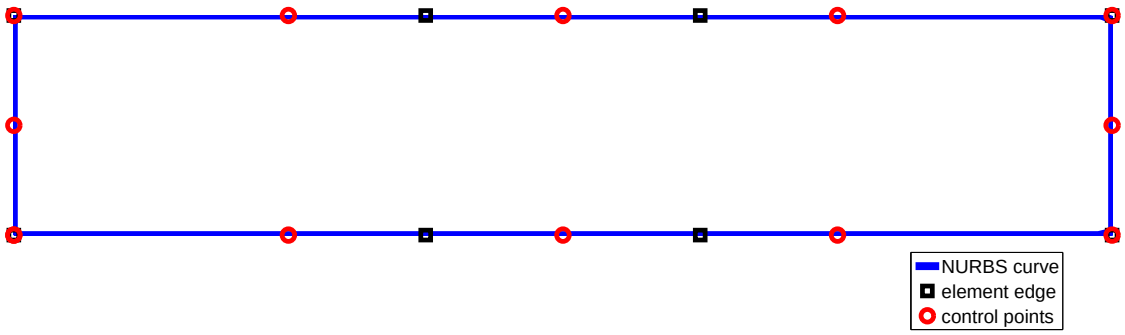

Figure 25. The design mesh of the cantilever beam

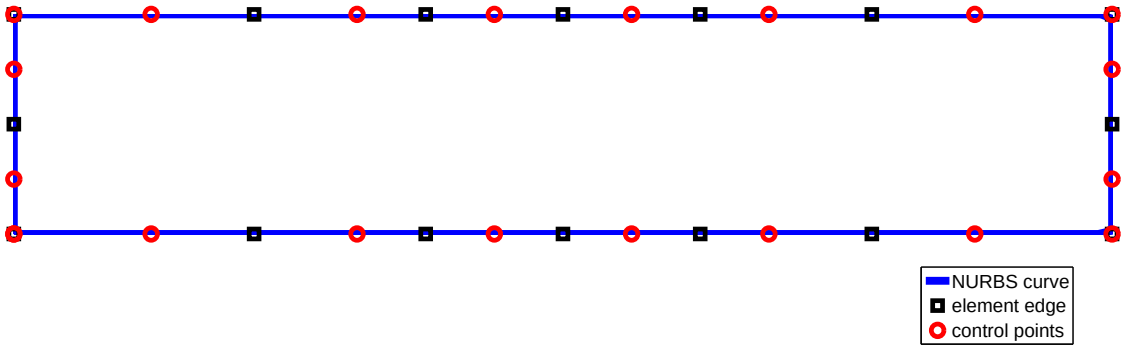

Figure 26. The analysis mesh of the cantilever beam

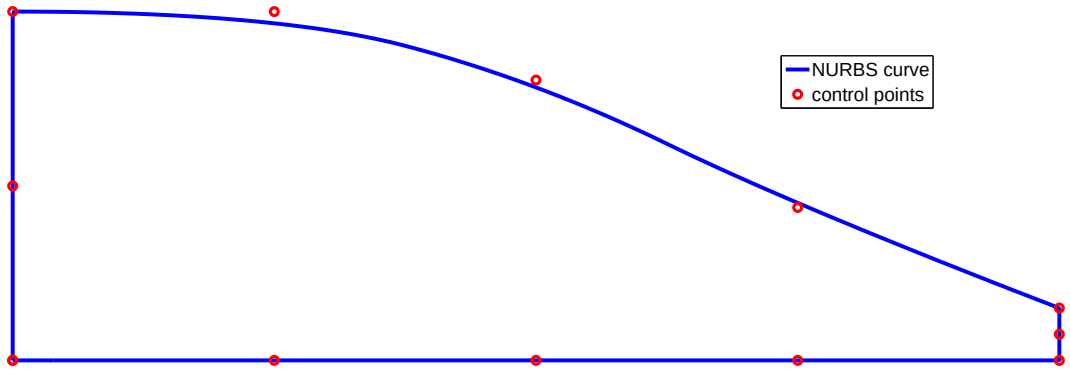

Figure 27. The optimal design for the cantilever beam

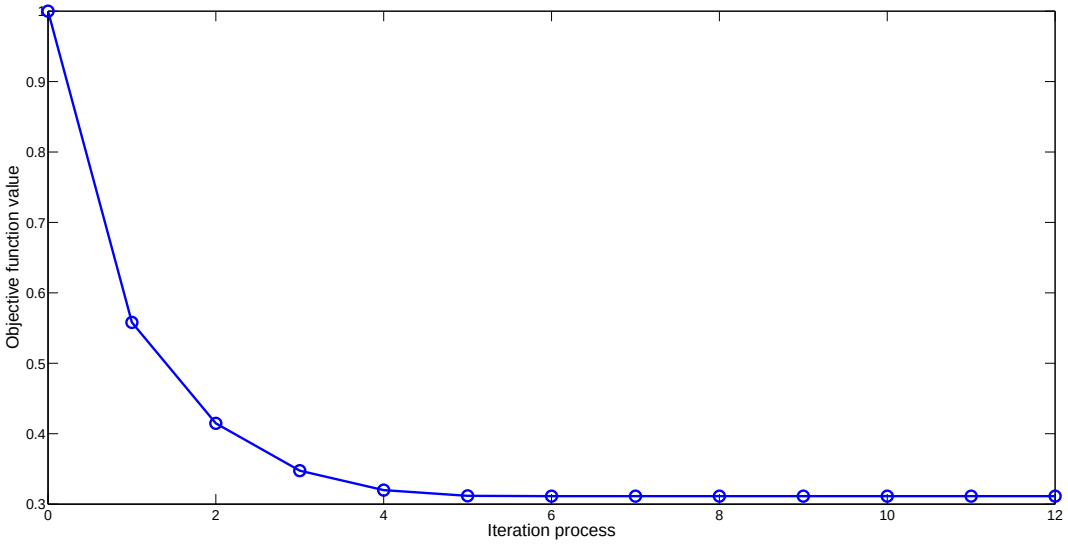

Figure 28. The iteration process of cantilever beam optimization

Design variable	Lower bound	Upper bound	Initial value	Final value
t_1	1.5	10	6	1.5
t_2	1.5	10	6	4.3841
t_3	1.5	10	6	8.0360
t_4	1.5	10	6	9.9999
t_5	1.5	10	6	10.0000

Table 2. Design variables in the cantilever beam optimization procedure

6.2.2 Fillet

Consider a fillet subject to a traction $\bar{t} = 100$ in y-direction. The objective is to minimize its area while keeping the von Mises stress below the allowable value $\hat{\sigma}_{\text{vm}} = 120$. Due to the symmetry, only a half is modeled, as shown in Fig. (29). The length of the segment AB = 20, BC = 9, CD = 4.5, and EF = 9. The Young's modulus is $E = 10^7$, and Poisson's ratio $\nu = 0.3$. For the shape optimization, the design curve is the line ED while point E and D fixed. Hence the vertical positions of the three control points between ED are set to be design variables (Fig. 30). 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 monitor points between ED in the analysis mesh as shown in Fig. 31, which is used for structural and sensitivity analysis. The optimal design of the fillet is shown in Fig. 32, with the final values of design variables seen in Table. (3). The area is reduced to 138.8776 from 145.1250, and the final design agrees with the reported result using Boundary Contour Method [32] very well. However, the present method is without any meshing in the iterative steps and possesses a great advantage.

Design variable	Lower bound	Upper bound	Initial value	Final value
t_1	0	4	5.625	4.8120
t_2	0	4	6.750	5.2156
t_3	0	4	7.875	6.1940

Table 3. Design variables in fillet optimization procedure

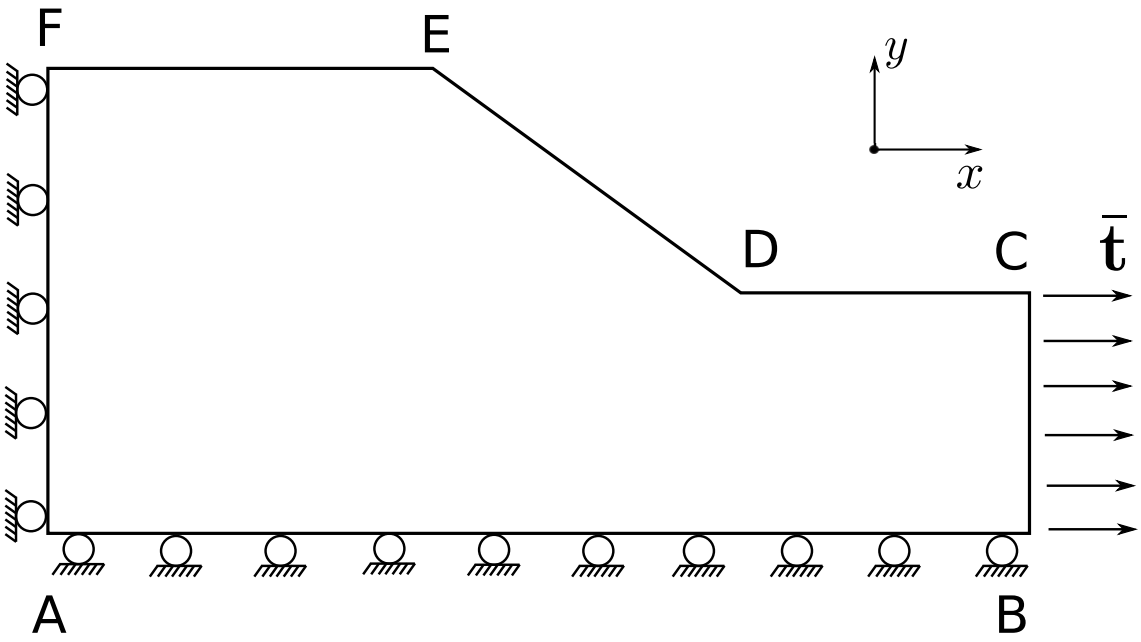

Figure 29. The definition of fillet problem

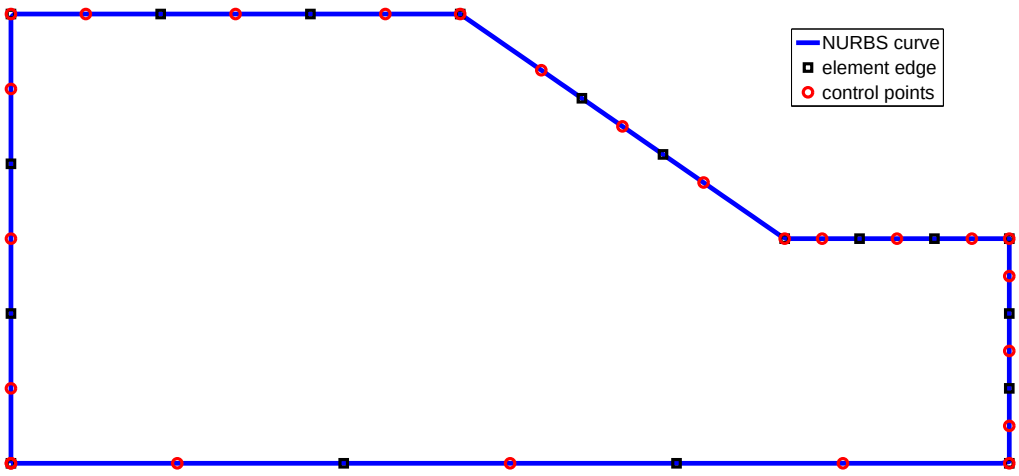

Figure 30. The design mesh of the fillet problem

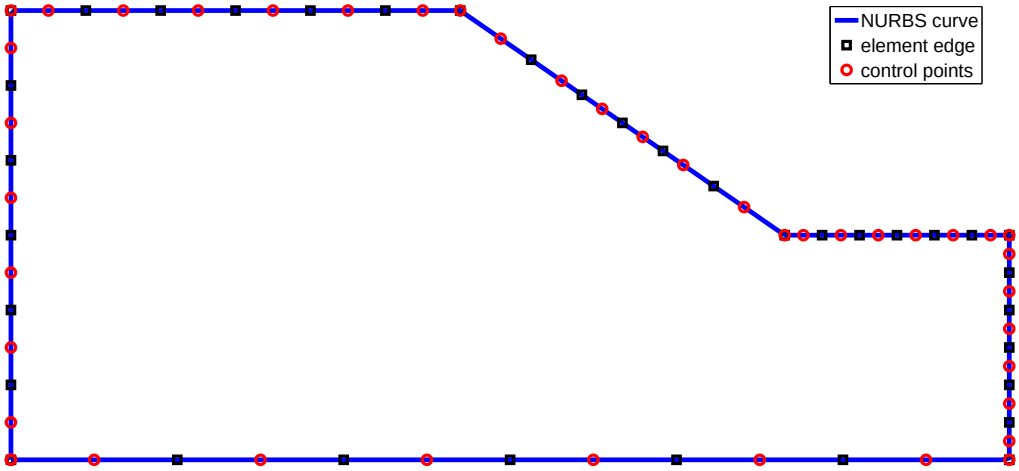

Figure 31. The analysis mesh of the fillet problem

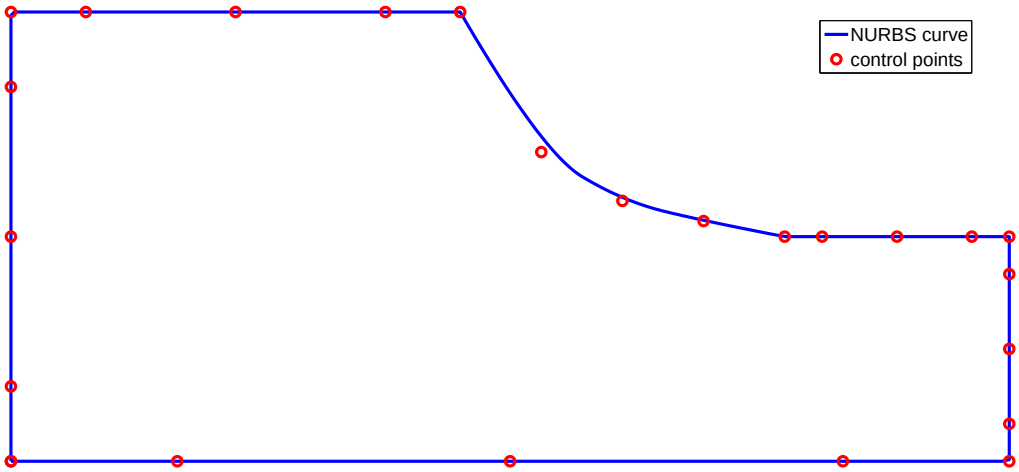

Figure 32. The optimal design of the fillet

6.2.3 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 modeled. The geometry of the initial design and the boundary conditions are shown in Fig. 33. 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 $p = 100$ and 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 design boundary is the line HG while end points G and H fixed, and its allowable von Mises stress is $\hat{\sigma}_{\text{vm}} = 600$. The vertical positions of the four control points on the design curve in the design mesh (Fig. 34) are set as design variables. The lower bound is $[45, 15, 15, 15]$, and the upper bound is $[70, 70, 70, 70]$. The monitor points are chosen between GH. The mesh for structural and shape sensitivity analysis is shown Fig. (35). The optimal geometry is shown in Fig. (36), with the values of final control points in Table (4).

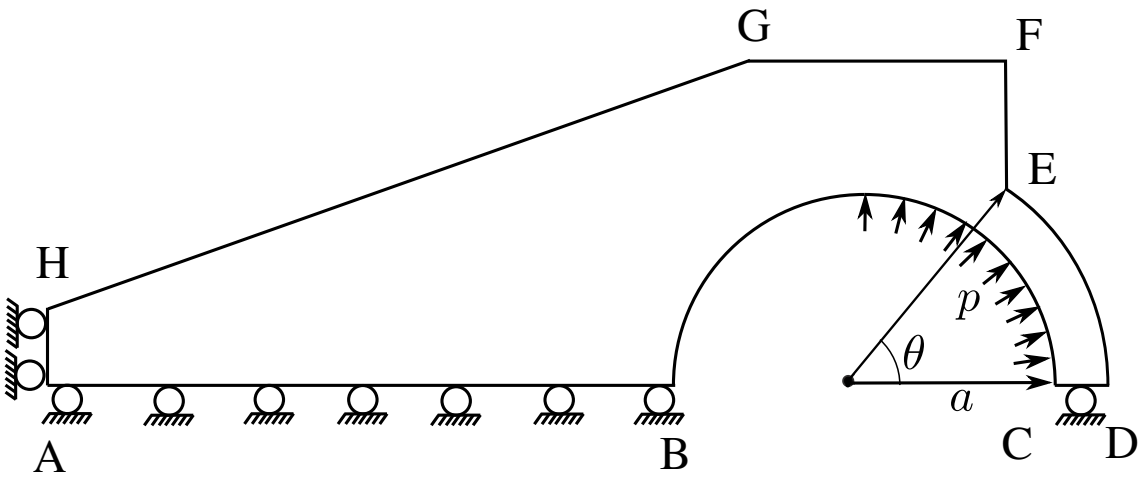

Figure 33. The definition of connecting rod problem

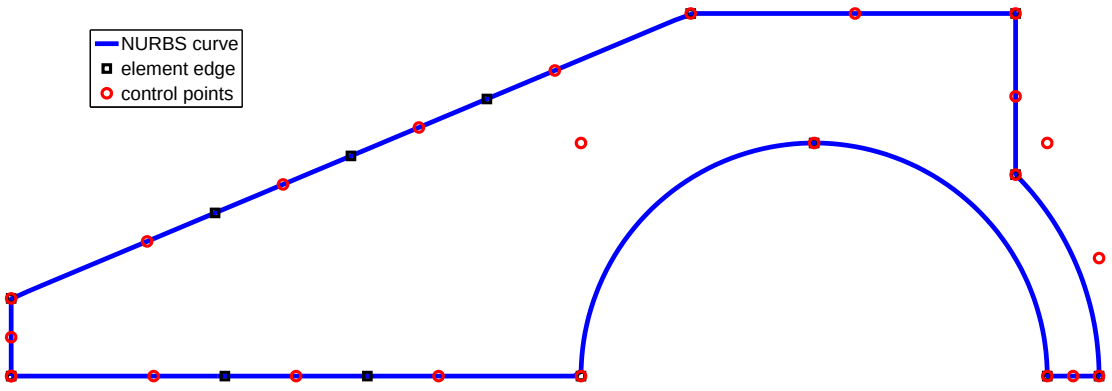

Figure 34. The design mesh of the connecting rod problem

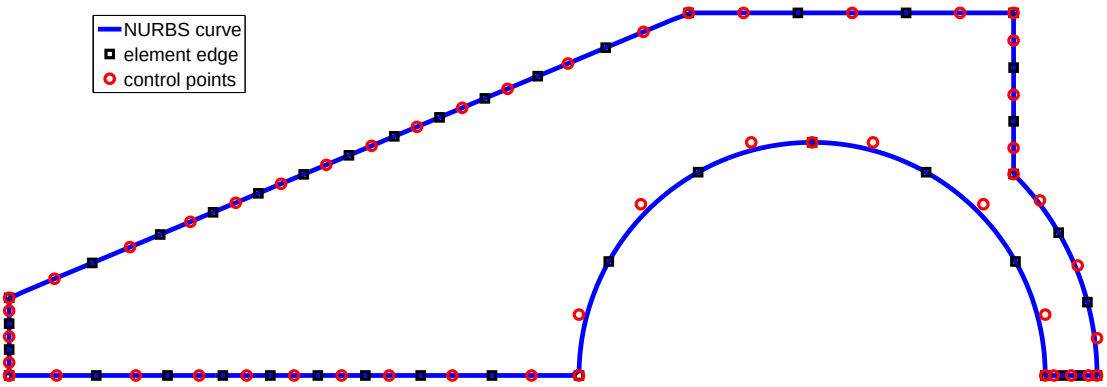

Figure 35. The analysis mesh of the connecting rod problem

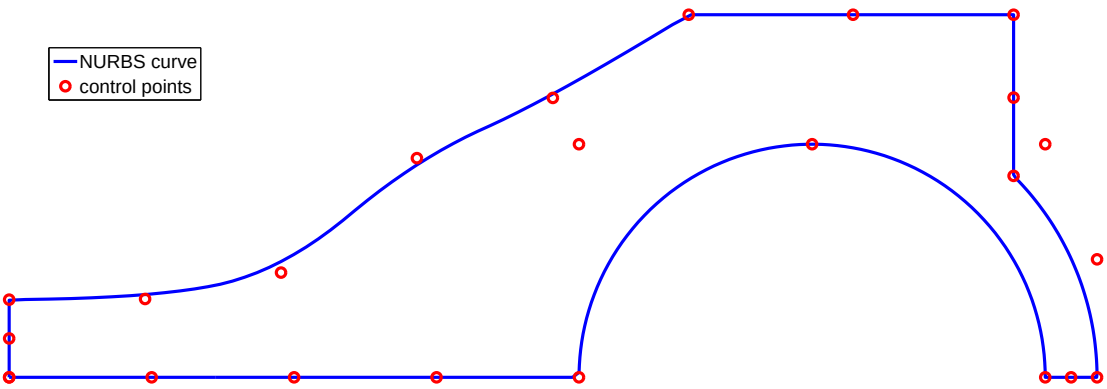

Figure 36. The optimal design of the connecting rod

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 4. Design variables in connecting rod optimization procedure

7 Conclusion

The formulation of shape sensitivity analysis and optimization using IGABEM is presented. IGABEM can achieve a seamless integration between the CAD and analysis, which is partic-

ularly of huge significance in shape optimization. The numerical examples have bee presented and show the advantages for the present work in shape optimization:

- A repeatedly meshing procedure is not needed. This is a significant improvement in computational efficience.
- The returned optimal model can be directly used in CAD without needing any smoothing or recovery procedure.
- The control mesh provides a natural and elegant choice of design variables.
- The structural and shape sensitivity analysis is performed on an exact geometry so the accuracy is improved, further accelerating the convergence of the optimization iterations.

The future works can be extended to acoustic shape optimization, where the advantages of IGABEM can be exhibited fully, without needing to mesh the open domain around the structure. A fast algorithm also needs to be considered to treat the full matrix in IGABEM.

Acknowledgments.

The authors are sincerely grateful for the support of the European Research Council Starting Independent Research Grant (ERC Stg grant agreement No. 279578) entitled "Towards real time multiscale simulation of cutting in non-linear materials with applications to surgical simulation and computer guided surgery".

Bibliography

[1] S. Atluri and T. Zhu. A new meshless local petrov-galerkin (mlpg) approach in computational mechanics. *Computational mechanics*, 22(2):117–127, 1998.

[2] I. Babuska and J. M. Melenk. The partition of unity finite element method. Technical Report, DTIC Document, 1995.

[3] M. 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.

[4] 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.

[5] T. Belytschko, Y. Y. Lu and L. Gu. Element-free galerkin methods. *International journal for numerical methods in engineering*, 37(2):229–256, 1994.

[6] 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.

[7] 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.

[8] S. Bordas and B. Moran. Enriched finite elements and level sets for damage tolerance assessment of complex structures. *Engineering Fracture Mechanics*, 73(9):1176–1201, 2006.

[9] S. 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, 2007.

[10] S. Cho and S.-H. Ha. Isogeometric shape design optimization: exact geometry and enhanced sensitivity. *Structural and Multidisciplinary Optimization*, 38(1):53–70, 2009.

[11] 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.

[12] K. K. Choi and E. J. Haug. Shape design sensitivity analysis of elastic structures. *Journal of Structural Mechanics*, 11(2):231–269, 1983.

[13] M. G. Cox. The numerical evaluation of b-splines. *IMA Journal of Applied Mathematics*, 10(2):134–149, 1972.

[14] T. Cruse. Numerical solutions in three dimensional elastostatics. *International journal of solids and structures*, 5:1259–1274, 1969.

[15] T. 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.

[16] C. De Boor. On calculating with< i> b</i>-splines. *Journal of Approximation Theory*, 6(1):50–62, 1972.

[17] J. Dolbow and T. Belytschko. A finite element method for crack growth without remeshing. *Int. J. Numer. Meth. Engng*, 46:131–150, 1999.

[18] C. A. Duarte and J. T. Oden. Hp clouds-an hp meshless method. *Numerical methods for partial differential equations*, 12(6):673–706, 1996.

[19] P. Duysinx, L. Van Miegroet, T. Jacobs and C. Fleury. Generalized shape optimization using x-fem and level set methods. In *IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials*, pp. 23–32. Springer, 2006.

[20] 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.

[21] T. Hughes, J. 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.

[22] M. 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.

[23] J. Kane and S. Saigai. Design-sensitivity analysis of solids using bem. *Journal of engineering mechanics*, 114(10):1703–1722, 1988.

[24] W. K. Liu, S. Jun and Y. F. Zhang. Reproducing kernel particle methods. *International journal for numerical methods in fluids*, 20(8-9):1081–1106, 1995.

[25] Y. Liu and T. 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.

[26] 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.

[27] Y. Liu and T. Rudolphi. New identities for fundamental solutions and their applications to non-singular boundary element formulations. *Computational mechanics*, 24(4):286–292, 1999.

[28] A. Longo, J. Unzueta, E. Schaeidt, A. Alvarez and J. Anza. A general related variational approach to shape optimum design. *Advances in Engineering Software*, 16(2):135–142, 1993.

[29] 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.

[30] L. V. Miegroet and P. Duysinx. Stress concentration minimization of 2d filets using x-fem and level set description. *Struct Multidiscip O*, 33(4-5):425–438, Jan 2007.

[31] 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, Jan 2008.

[32] A.-V. u. ˜. Phan, S. Mukherjee and J. 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.

[33] C. Politis, A. Ginnis, P. 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.

[34] L. Piegl and W. Tiller. *The NURBS Book*. Springer-Verlag, New York, 1997.

[35] X. Qian. Full analytical sensitivities in nurbs based isogeometric shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 199(29):2059–2071, 2010.

[36] F. Rizzo. An integral equation approach to boundary value problems of classical elastostatics. *Quart. Appl. Math*, 25(1):83–95, 1967.

[37] M. Scott, X. Li, T. Sederberg and T. Hughes. Local refinement of analysis-suitable T-splines. *Computer Methods in Applied Mechanics and Engineering*, accepted for publication, doi:10.1016/j.cma.2011.11.022, 2011.

[38] M. Scott, R. N. Simpson, J. Evans, S. Lipton, S. P. A. Bordas, T. Hughes and T. Sederberg. Isogeometric boundary element analysis using unstructured t-splines. *Computer Methods in Applied Mechanics and Engineering*, 254:197–221, 2013.

[39] R. Simpson, S. 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.

[40] R. Simpson, M. Scott, M. Taus, D. Thomas and H. Lian. Acoustic isogeometric boundary element analysis. *Computer Methods in Applied Mechanics and Engineering*, 269:265–290, 2014.

[41] 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.

[42] G. 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.

[43] J. 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.

[44] L. Van Miegroet and P. Duysinx. 3d shape optimization with x-fem and a level set constructive geometry approach. In *Proceeding of the 8th World Congress on Structural and Multidisciplinary Optimization*. 2009.

[45] L. Van Miegroet, N. Moës, C. Fleury and P. Duysinx. Generalized shape optimization based on the level set method. In *Proceedings of the 6th World Congress of Structural and Multidisciplinary Optimization (WCSMO6)*. 2005.

[46] 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.

[47] 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.

[48] K. Yamazaki, J. Sakamoto and M. Kitano. Three-dimensional shape optimization using the boundary element method. *AIAA journal*, 32(6):1295–1301, 1994.

[49] R. Yang. Component shape optimization using bem. *Computers & Structures*, 37(4):561–568, 1990.

[50] Z. Zhang, J. Zhou, N. Zhou, X. 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.

[51] O. Zienkiewicz and J. Campbell. Shape optimization and sequential linear programming. *Optimum structural design*, :109–126, 1973.

[52] O. C. Zienkiewicz and R. L. Taylor. *The finite element method*, volume 3. McGraw-hill London, 1977.