

Implementation of regularized isogeometric boundary element methods for gradient-based shape optimization in two-dimensional linear elasticity

H. Lian¹, P. Kerfriden¹, S.P.A. Bordas^{2,1,*}

¹*School of Engineering, Cardiff University, Queen's Buildings, The Parade, Cardiff, CF24 3AA, UK*

²*Faculté des Sciences, de La Technologie, Luxembourg University, 6, rue Richard Coudenhove-Kalergi, L-1359 Luxembourg*

SUMMARY

The present work addresses shape sensitivity analysis and optimization in two-dimensional elasticity with a regularised isogeometric boundary element method (IGABEM). NURBS are used both for the geometry and the basis functions to discretize the regularised boundary integral equations. With the advantage of tight integration of design and analysis, the application of IGABEM in shape optimization reduces the mesh generation/regeneration burden greatly. The work is distinct from the previous literatures in IGABEM shape optimization mainly in two aspects: 1) the structural and sensitivity analysis takes advantage of the regularized form of the boundary integral equations, eliminating completely the need of evaluating strongly singular integrals and jump terms and their shape derivatives, which were the main implementation difficulty in IGABEM, and 2) although based on the same CAD model, the mesh for structural and shape sensitivity analysis is separated from the geometrical design mesh, thus achieving a balance between less design variables for efficiency and refined mesh for accuracy. This technique was initially used in isogeometric finite element method and was incorporated into the present IGABEM implementation. Copyright © 0000 John Wiley & Sons, Ltd.

Received ...

KEY WORDS: shape optimization; isogeometric boundary element method; regularised Boundary Integral Equations

1. INTRODUCTION

This paper presents an isogeometric boundary element method for shape optimization assuming a linear, homogeneous, isotropic material. Shape optimization is a process to find the optimal shape of a component or structure under given constraints and objectives. To achieve automatic shape optimization, the finite element method (FEM) [1] combined with mathematical programming algorithms is probably the most commonly used approach [2]. However, a mesh must be created in

*Correspondence to: Faculté des Sciences, de La Technologie, Luxembourg University, 6, rue Richard Coudenhove-Kalergi, L-1359 Luxembourg. E-mail: stephane.bordas@uni.lu

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 for a number of existing methods, causes a cumbersome remeshing procedure (Fig. 4(a)). The meshing/remeshing procedure is time-consuming and far from being automated. It may happen in industrial practice that the geometry is so complex that available mesh generators fail, or require significant human intervention. For linear elastic problems, authors report that the mesh generation step represents as much as 80% of the total time. To alleviate the mesh burden, numerous works were proposed, and reviewed in [3].

Meshfree/meshless methods Meshfree (meshless) methods [4], refer to a broad collection of numerical methods, including the smoothed particle hydrodynamics method (SPH) [5], the element-free Galerkin Method (EFG) [6], the reproducing kernel particle Method (RKPM) [7], the meshless local Petrov-Galerkin method (MLPG) [8], the *hp*-cloud method [9], the partition of unity finite element method (PUFEM) [10], *etc.* Although different in the way of formulating shape and test functions, they share the same characteristic of lifting the strict connectivity requirements posed by the FEM. In contrast to the 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). 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 [11, 12, 13]. 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 [14], 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 integration. Background meshes [15] or other advanced techniques must thus be used, such as direct nodal integration [16, 17, 18], stabilized nodal integration [19], stress point integration [20, 21], and support-based integration [22, 23], which complicate the approach and somewhat detract from the “meshfree” concept.

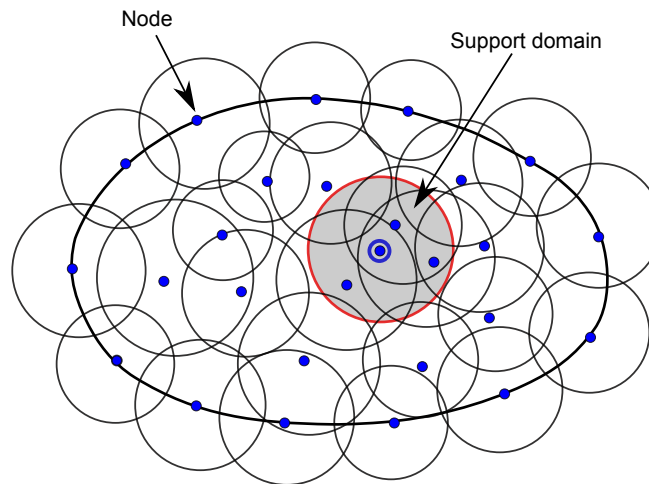


Figure 1. Meshfree method domain discretization

Boundary element methods Boundary Element Methods (BEM) [24, 25, 26, 27, 28] 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. The main advantage of the BEM for 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 [29, 30, 31]. However, the surface meshing of the BEM is still cumbersome for shape optimization in large scale problems. Moreover, BEM's advantages are mostly visible in cases where Green's functions are available.

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, including immersed boundary methods [32], fictitious domain [33], embedded boundary [34], virtual boundary [35] and Cartesian grid methods [36]. The Extended finite element method (XFEM) [37, 15, 38, 39] also falls into this category and its application in shape optimization can be read in [40, 41, 42, 43]. 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. Moës [44] proposed an XFEM-based method for complex microstructures, which was generalized by Moumnassi *et al.* [45] to treat arbitrary CAD geometries implicitly, including corners and sharp edges.

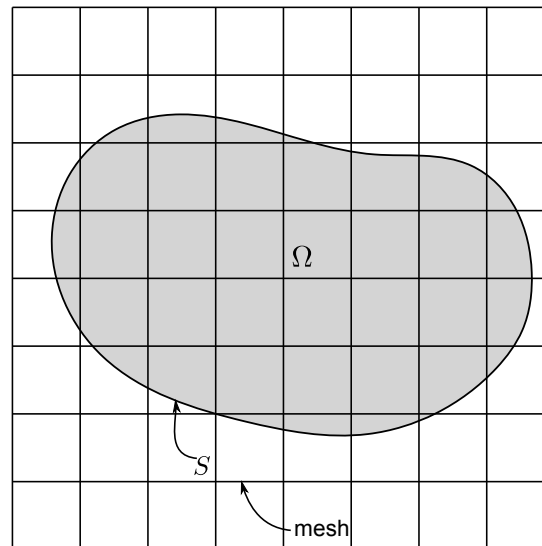


Figure 2. Implicit boundary methods

Isogeometric analysis A recent trend in shape optimization is isogeometric analysis (IGA) [46], which integrates the geometry and analysis representations. 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 idea is that the meshing procedure is bypassed because an existing CAD geometry is used directly for analysis, meanwhile keeping the exact geometry. Yet, in its finite element form, IGA is limited in reaching such benefits, because a mesh (or

parameterization) for the interior of the domain must be generated, which causes difficulties and remains an unsolved problem. Moreover, local refinement of the basis functions is difficult or impossible since tensor product splines are typically used. The application of IGA in shape optimization can be found in [47, 48, 49, 50, 51].

NURBS-enhanced finite element methods NURBS-enhanced finite element methods (NEFEM) [52, 53] 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. 3. 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 with FEM.

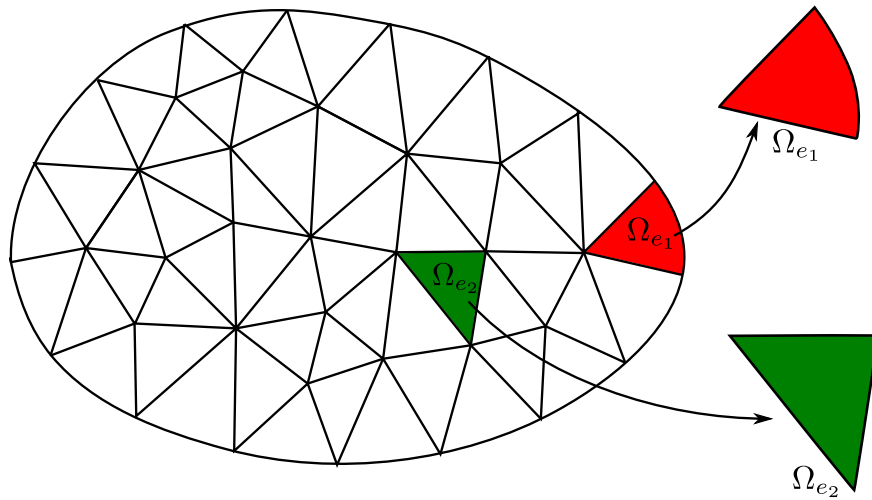


Figure 3. NURBS-enhanced FEM mesh

Geometry-independent field approximation In the geometry-independent field approximation (GIFT) [54], 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 require a volume parameterization which is an open problem as mentioned above.

The isogeometric boundary element method (IGABEM) alleviates most of the above difficulties and allows a truly general and automatic CAD to analysis transitions. It was proposed by the Cardiff group in [55, 56, 57] and, almost simultaneously by [58]. 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 is a particularly suitable choice for applications in shape optimization, which is the topic of the present work. The comparison between FEM and IGABEM in shape optimization procedures can be clearly

seen in Fig. 4. The shape optimization in three dimensional linear elasticity was conducted by the IGABEM with NURBS [59]. [60] incorporated T-splines to IGABEM for shape-hull optimization in hydrodynamic problems. Our work is different in the following aspects: 1) The approach proposed by [50] in Isogeometric Finite Element Methods to separate the geometry representation and the field variable representation were incorporated, which can increase the analysis accuracy with a refined mesh, but limit the number of design variables by using a coarse mesh concerning the efficiency. 2) Shape sensitivity analyses are carried out in detail in order to fully verify the method, and the optimization problems have included displacement minimization, compliance minimization, in addition to stress constraint optimization in [59]. 3) More importantly, we use regularized boundary integral equations [64, 65, 66, 56] for sensitivity analysis, which is in opposition to existing work in the literature [55, 59]. This avoids the need to compute strongly singular integrals and jump terms and the related shape derivatives, which is the main difficulty to implement isogeometric boundary element method. 4) From the implementation perspective, our paper gives all required details so that the results can be fully reproduced.

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 form. 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 proposes conclusions and directions for future work.

2. B-SPLINES AND NURBS

2.1. Knot vector

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

$$\{\xi_1, \xi_2, \dots, \xi_{n+p+1}\} \quad \xi_A \in \mathbb{R}, \quad (1)$$

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. 5.

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, $[0, 1]$, $[1, 2]$, $[2, 3]$.

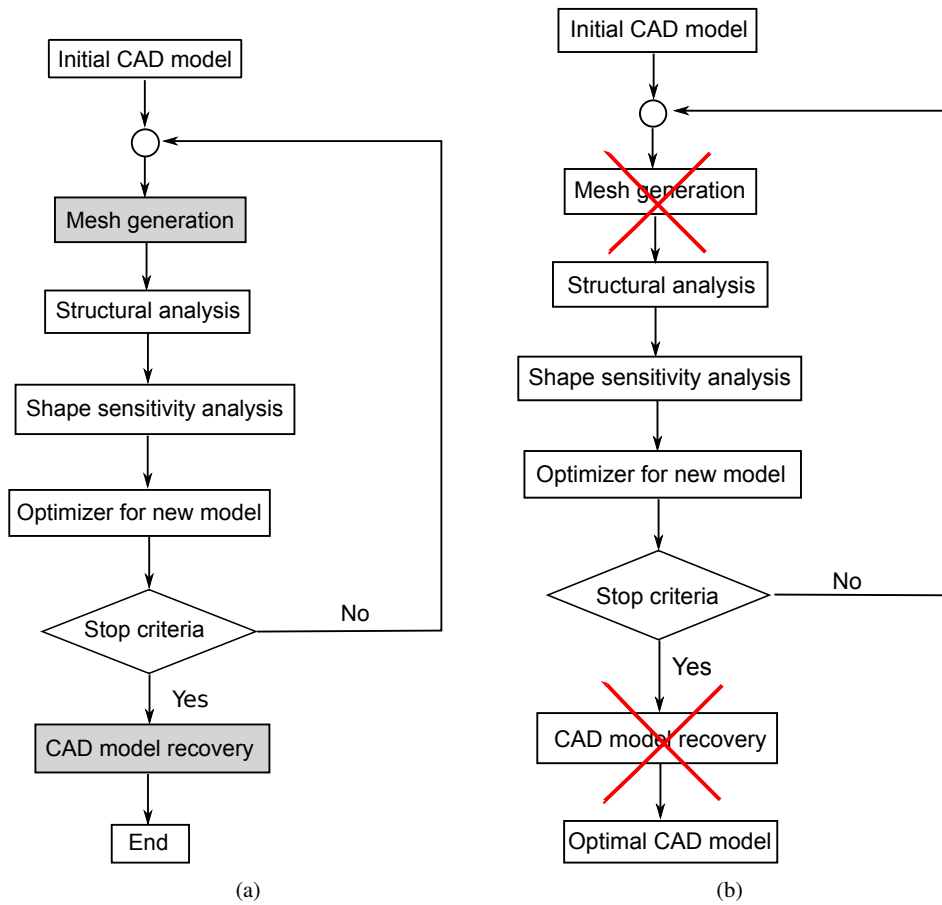


Figure 4. (a) FEM shape optimization flowchart and (b) IGABEM shape optimization flowchart

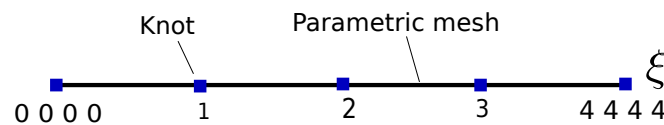


Figure 5. Knot vector

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, $\{0, 0, 0, 1, 2, 3, 4, 5, 5, 5\}$.

2.2. 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** [61, 62]

$$N_{A,0}(\xi) = \begin{cases} 1 & \text{if } \xi_A \leq \xi < \xi_{A+1}, \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

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

In essence a B-spline basis function is a piecewise polynomial function. The functions 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. 6):

- **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.

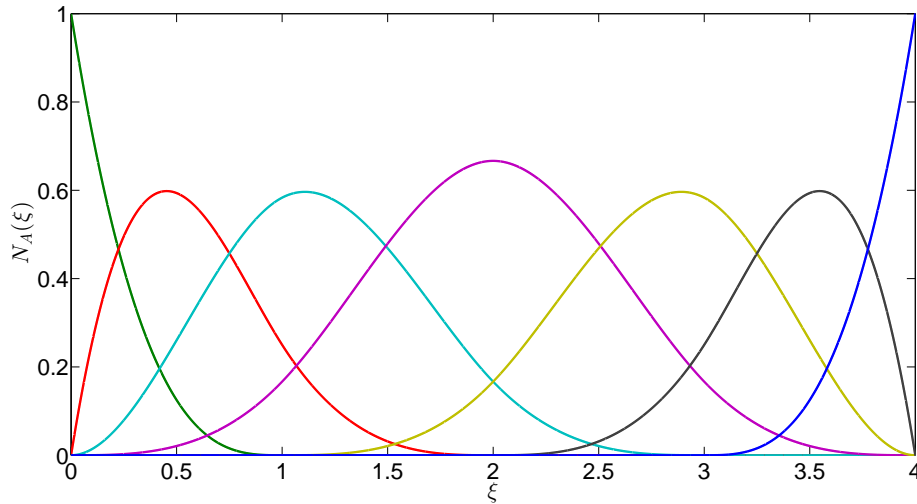


Figure 6. B-spline basis functions ($p = 3$) for knot vector $\{0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4\}$

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

$$\frac{d}{d\xi} N_{A,p}(\xi) = \frac{p}{\xi_{A+p} - \xi_A} N_{A,p-1}(\xi) - \frac{p}{\xi_{A+p+1} - \xi_{A+1}} N_{A+1,p-1}(\xi). \quad (4)$$

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

$$\begin{aligned} \frac{d^k}{d\xi^k} N_{A,p}(\xi) &= \frac{p}{\xi_{A+p} - \xi_A} \left(\frac{d^{k-1}}{d\xi^{k-1}} N_{A,p-1}(\xi) \right) \\ &\quad - \frac{p}{\xi_{A+p+1} - \xi_{A+1}} \left(\frac{d^{k-1}}{d\xi^{k-1}} N_{A+1,p-1}(\xi) \right). \end{aligned} \quad (5)$$

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

$$\frac{d^k}{d\xi^k} N_{A,p}(\xi) = \frac{p!}{(p-k)!} \sum_{j=0}^k \alpha_{k,j} N_{A+j,p-k}(\xi), \quad (6)$$

with

$$\begin{aligned} \alpha_{0,0} &= 1, \\ \alpha_{k,0} &= \frac{\alpha_{k-1,0}}{\xi_{A+p-k+1} - \xi_A}, \\ \alpha_{k,j} &= \frac{\alpha_{k-1,j} - \alpha_{k-1,j-1}}{\xi_{A+p+j-k+1} - \xi_{A+j}}, \quad j = 1, \dots, k-1, \\ \alpha_{k,k} &= \frac{-\alpha_{k-1,k-1}}{\xi_{A+p+1} - \xi_{A+k}}. \end{aligned} \quad (7)$$

2.3. NURBS basis functions

Non-uniform Rational B-Splines (NURBS) [63] 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 geometries in the present work are represented by NURBS.

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

$$R_{A,p}(\xi) = \frac{N_{A,p}(\xi) w_A}{W(\xi)}, \quad (8)$$

with

$$W(\xi) = \sum_{A=1}^n w_A N_{A,p}(\xi), \quad (9)$$

where w_A denotes a weight associated with 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. 7). When all of the weights are equal to 1, the NURBS curve reduces to a B-spline curve.

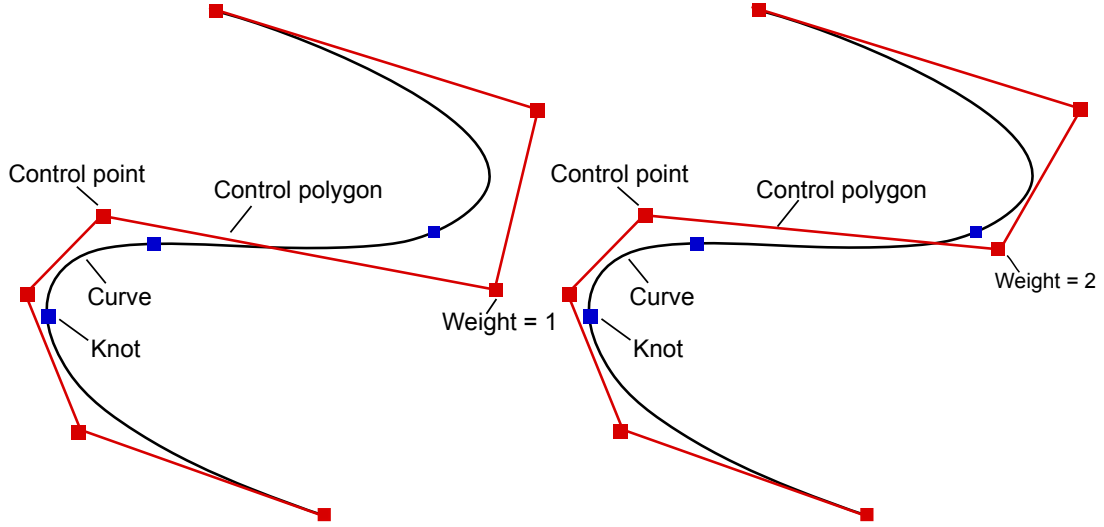


Figure 7. The comparison between NURBS curve with different weights

The derivatives of a NURBS basis function are expressed by

$$\frac{d}{d\xi} R_{A,p}(\xi) = w_A \frac{W(\xi) \frac{d}{d\xi} N_{A,p}(\xi) - \frac{d}{d\xi} W(\xi) N_{A,p}(\xi)}{(W(\xi))^2}, \quad (10)$$

and

$$\frac{d}{d\xi} W(\xi) = \sum_{A=1}^n \frac{d}{d\xi} N_{A,p}(\xi) w_A. \quad (11)$$

2.4. NURBS geometries

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

$$R_A(\xi|\Xi_A) \equiv \prod_{i=1}^{d_p} R_A^i(\xi_A^i|\Xi_A^i), \quad (12)$$

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

$$R_{A,B}^{p,q}(\xi, \eta) = \frac{N_{A,p}(\xi) M_{B,q}(\eta) w_{A,B}}{\sum_{\hat{A}=1}^n \sum_{\hat{B}=1}^m N_{\hat{A},p}(\xi) M_{\hat{B},q}(\eta) w_{\hat{A},\hat{B}}}, \quad (13)$$

$$R_{A,B,C}^{p,q,r}(\xi, \eta, \zeta) = \frac{N_{A,p}(\xi) M_{B,q}(\eta) L_{C,r}(\zeta) w_{A,B,C}}{\sum_{\hat{A}=1}^n \sum_{\hat{B}=1}^m \sum_{\hat{C}=1}^l N_{\hat{A},p}(\xi) M_{\hat{B},q}(\eta) L_{\hat{C},r}(\zeta) w_{\hat{A},\hat{B},\hat{C}}}. \quad (14)$$

A 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. A NURBS curve can be expressed as

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

where $\mathbf{x}(\xi)$ denotes the physical curve of interest, ξ is the coordinate in parametric space, \mathbf{P}_A the control points, $N_{A,p}$ the B-spline basis functions of order p . See Fig. 8. NURBS surfaces can be constructed in a similar way (Fig. 9).

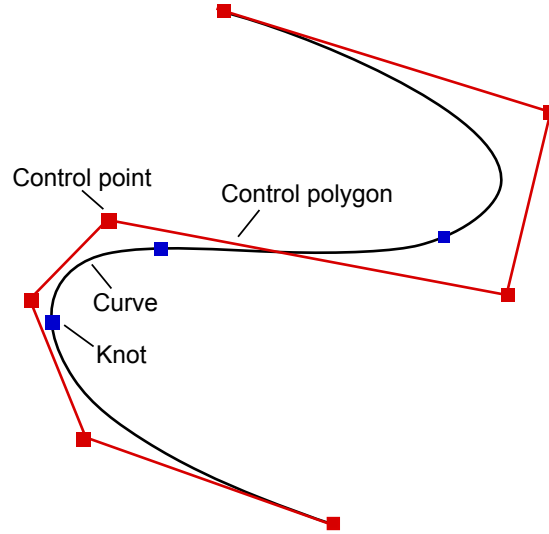


Figure 8. NURBS curve

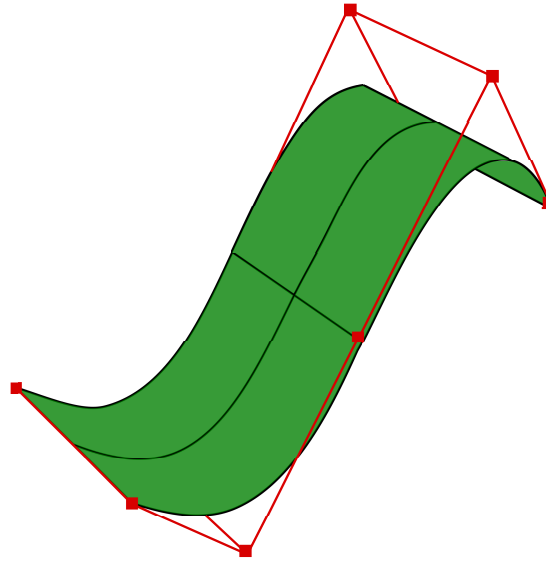


Figure 9. NURBS surface

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

$$\mathbf{x}(\xi) = \sum_{A=1}^n N_{A,p}(\xi) \tilde{\mathbf{P}}_A, \quad (16)$$

where $\tilde{\mathbf{P}}_A = \{w_A \mathbf{P}_A, w_A\}^T$ are 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. 9.
- The variation diminishing property. No plane has more intersections with the curve than it has with the control grid. This property renders NURBS less oscillatory than Lagrangian polynomials.
- The transformation invariance property. An affine transformation of a NURBS can be achieved by applying an affine transformation to the control points.
- Non-interpolatory. The NURBS 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.

2.5. Knot insertion in NURBS

Knot insertion is used to enrich the NURBS basis function space. Let $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$ be a knot vector, $\tilde{\mathbf{P}}$ the corresponding weighted control points. If we insert a new knot $\tilde{\xi} \in [\xi_k, \xi_{k+1}]$, the added control point $\tilde{\tilde{\mathbf{P}}}$ can be obtained as follows without changing the geometry,

$$\tilde{\tilde{\mathbf{P}}}_A = \begin{cases} \tilde{\mathbf{P}}_1 & A = 1, \\ \alpha_A \tilde{\mathbf{P}}_A + (1 - \alpha_A) \tilde{\mathbf{P}}_{A-1} & 1 < A < m, \\ \tilde{\mathbf{P}}_n & A = m, \end{cases} \quad (17)$$

with

$$\alpha_A = \begin{cases} 1 & 1 \leq A \leq k - p, \\ \frac{\tilde{\xi}_A - \xi_A}{\xi_{A+p} - \xi_A} & k - p + 1 \leq A \leq k, \\ 0 & A \geq k + 1. \end{cases} \quad (18)$$

2.6. NURBS element structure

Knot vectors used to define the NURBS basis functions provide natural element structures which are very useful for numerical analysis using FEM or BEM. We can view the non-zero knot interval as an element in each dimension. The only difference from isoparametric elements which are 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 on each knot interval into a standard space $\hat{\Omega} [-1, +1]$, and $d\Omega_e = J_e d\hat{\Omega}_e$, with

$$J_e = \frac{\xi_b - \xi_a}{2}, \quad (19)$$

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 for linear elastostatics

In the linear elastic problem as shown in Fig. 10, the equilibrium equations are expressed as

$$\sigma_{ij,j} + b_i = 0, \quad (20)$$

where i, j denote Cartesian components and range from 1 to 2 in two dimensions, σ_{ij} are the components of stress tensor, ϵ_{ij} the strain tensor, b_i the body force. The strain tensor ϵ_{ij} is defined by

$$\epsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}), \quad (21)$$

where u_i is the displacement field and a comma implies a differentiation. The boundary conditions are

$$u_i = \bar{u}_i \quad \text{on } S_u \subseteq S, \quad (22)$$

$$\sigma_{ij}n_j = \bar{t}_i \quad \text{on } S_t \subseteq S. \quad (23)$$

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

$$S_u \cup S_t = S, \quad S_u \cap S_t = \emptyset. \quad (24)$$

The displacement boundary integral equation (DBIE) can be written as

$$C_{ij}(\mathbf{s})u_j(\mathbf{s}) + \oint_S T_{ij}(\mathbf{s}, \mathbf{x})u_j(\mathbf{x})dS(\mathbf{x}) = \int_S U_{ij}(\mathbf{s}, \mathbf{x})t_j(\mathbf{x})dS(\mathbf{x}), \quad (25)$$

where i, j denote Cartesian component and range from 1 to 2 in two dimensional cases. 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

$$U_{ij}(\mathbf{s}, \mathbf{x}) = \frac{1}{8\pi\mu(1-\nu)} \left[(3-4\nu) \ln\left(\frac{1}{r}\right) \delta_{ij} + r_{,i}r_{,j} \right], \quad (26)$$

$$T_{ij}(\mathbf{s}, \mathbf{x}) = -\frac{1}{4\pi(1-\nu)r} \left\{ \frac{\partial r}{\partial n} [(1-2\nu)\delta_{ij} + 2r_{,i}r_{,j}] - (1-2\nu)(r_{,i}n_j - r_{,j}n_i) \right\}, \quad (27)$$

where \mathbf{x} is the field point on the boundary, \mathbf{s} the source point, $r = r(\mathbf{x}, \mathbf{s}) = \|\mathbf{x} - \mathbf{s}\|$ the distance between the source point and field point (Fig. 11), $\lambda = 2\nu\mu/(1-2\nu)$ is Lamé constant, and $\mu = E/2(1+\nu)$ is shear modulus of elasticity, with E denoting Young's modulus, and ν Poisson's ratio, and δ_{ij} is the Kronecker delta symbol with the property

$$\delta_{ij} = \begin{cases} 0 & i \neq j, \\ 1 & i = j. \end{cases} \quad (28)$$

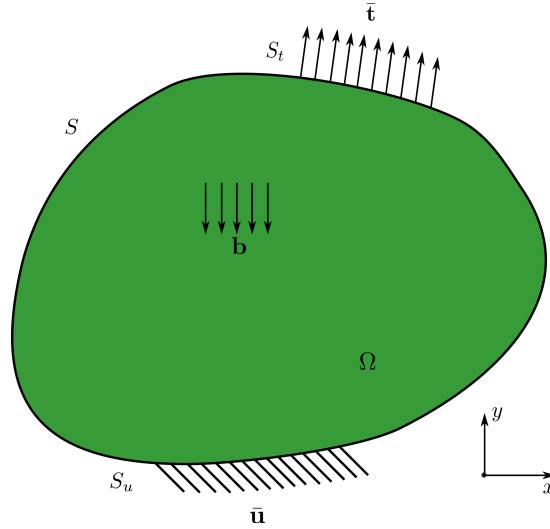


Figure 10. The distance between the source point and field point

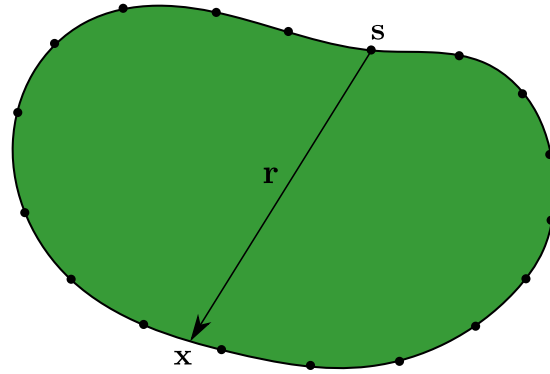


Figure 11. The distance between the source point and field point

Eq. (25) is a singular form and requires an explicit evaluation of jump terms and strongly singular integrals in IGABEM. To overcome this difficulty, we employ a regularized boundary integral equation proposed by Liu in [64, 65, 66], which is written as

$$\int_S T_{ij}(\mathbf{s}, \mathbf{x}) [u_j(\mathbf{x}) - u_j(\mathbf{s})] dS(\mathbf{x}) = \int_S U_{ij}(\mathbf{s}, \mathbf{x}) t_j(\mathbf{x}) dS(\mathbf{x}). \quad (29)$$

The regularized form cancels the singularity of the left-hand side of Eq. (29) because

$$T_{ij}(\mathbf{s}, \mathbf{x}) [u_j(\mathbf{x}) - u_j(\mathbf{s})] \sim \mathcal{O}\left(\frac{1}{r}\right) \mathcal{O}(r) = \mathcal{O}(1) \quad \text{in 2D.} \quad (30)$$

The weakly singular integrals in the right-hand side of Eq. (29) can be evaluated using a Telles transformation [67].

The geometry in Eq. (29) was parameterized using NURBS basis functions

$$\mathbf{x}(\xi) = \sum_{A=1}^n R_A(\xi) \mathbf{P}_A, \quad (31)$$

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

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

$$u_j(\xi) = \sum_{A=1}^n R_A(\xi) \tilde{u}_j^A, \quad (32)$$

$$t_j(\xi) = \sum_{A=1}^n R_A(\xi) \tilde{t}_j^A, \quad (33)$$

where \tilde{u}_j^A and \tilde{t}_j^A are the nodal unknowns related to displacements and tractions.

Substituting Eqs. (31,32,33) to Eq. (29) leads to a discrete form of regularized BIE

$$\begin{aligned} & \sum_{A=1}^n \int_S T_{ij}(\zeta_c, \xi) [R_A(\xi) - R_A(\zeta_c)] J(\xi) dS(\xi) \tilde{u}_j^A \\ &= \sum_{A=1}^n \int_S U_{ij}(\zeta_c, \xi) R_A(\xi) J(\xi) dS(\xi) \tilde{t}_j^A, \end{aligned} \quad (34)$$

where ζ_c denotes the parametric coordinate of the collocation point and c the global index of the collocation point. To perform a numerical integration using the Gauss-Legendre rule, the above

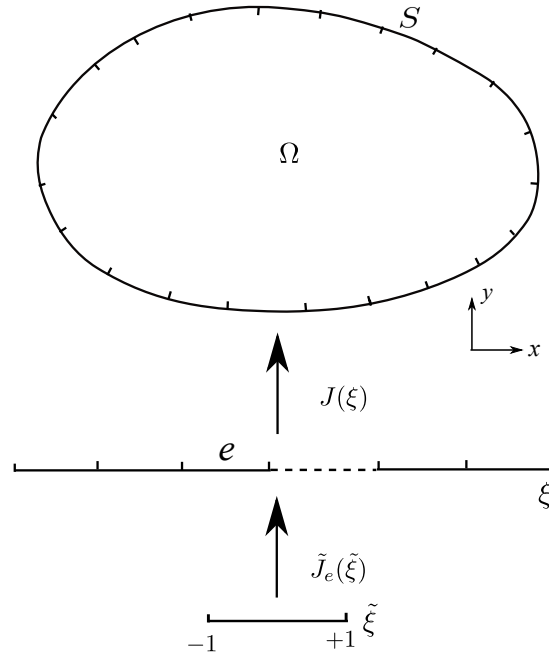


Figure 12. The mapping from parent elements to physical space

integral should be transformed into parent elements with the Jacobian \tilde{J}_e (see Fig. 12). So the above integral becomes

$$\sum_{A=1}^n \int_{-1}^{+1} T_{ij}(\zeta_c, \xi) [R_A(\xi) - R_A(\zeta_c)] J(\xi) \tilde{J}_e(\tilde{\xi}) d\tilde{S}(\tilde{\xi}) \tilde{u}_j^A$$

$$= \sum_{A=1}^n \int_{-1}^{+1} U_{ij}(\zeta_c, \xi) R_A(\xi) J(\xi) \tilde{J}_e(\tilde{\xi}) dS(\tilde{\xi}) \tilde{t}_j^A, \quad (35)$$

where $\tilde{\xi}$ is the intrinsic coordinate of the field point in the parent element. By denoting the terms of the above equation as

$$H_{ij}^{cA} = \int_{-1}^{+1} T_{ij}(\zeta_c, \xi) [R_A(\xi) - R_A(\zeta_c)] J(\xi) \tilde{J}_e(\tilde{\xi}) dS(\tilde{\xi}), \quad (36)$$

$$G_{ij}^{cA} = \int_{-1}^{+1} U_{ij}(\zeta_c, \xi) R_A(\xi) J(\xi) \tilde{J}_e(\tilde{\xi}) dS(\tilde{\xi}), \quad (37)$$

we can rewrite Eq. (35) in matrix form

$$\mathbf{H}\mathbf{u} = \mathbf{G}\mathbf{t}, \quad (38)$$

where \mathbf{H} and \mathbf{G} are the matrices collecting the terms of H_{ij}^{cA} and G_{ij}^{cA} , respectively. \mathbf{u} contains the nodal displacement parameters, and \mathbf{t} the nodal traction parameters. Both \mathbf{u} and \mathbf{t} include unknown and known values given by boundary conditions. By swapping the unknowns of both sides of Eq. (38), we can rearrange it as

$$\mathbf{A}\mathbf{z} = \mathbf{B}\mathbf{y} = \mathbf{f}. \quad (39)$$

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 directly into nodal parameters. In the isogeometric finite element method, this task can be done through Lagrange multiplier methods, penalty methods, or Nitsche's method [68]. However, these methods are not available in collocation IGABEM, because the method is not based on a variational equality. Hence, a nodal parameter extraction approach should be used, which can be conducted by a point collocation method or a L_2 projection method.

3.2.1. Collocation method The collocation method enforces boundary conditions to be satisfied 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,

$$\mathbf{u}(\xi_c) = \bar{\mathbf{u}}(\xi_c) \quad \text{on } S_u, \quad (40)$$

$$\mathbf{t}(\xi_c) = \bar{\mathbf{t}}(\xi_c) \quad \text{on } S_t, \quad (41)$$

where S_u is the portion of the boundary with displacement boundary conditions, and S_t with traction boundary conditions. ξ_c denotes the collocation point with index c , which can be chosen identical to that used for constructing IGABEM equations.

Substituting Eqs.(32, 33) into the above equations and using a matrix form produces

$$\mathbf{R}(\xi_c)\tilde{\mathbf{u}} = \bar{\mathbf{u}}(\xi_c) \quad \text{on } S_u, \quad (42)$$

$$\mathbf{R}(\xi_c)\tilde{\mathbf{t}} = \bar{\mathbf{t}}(\xi_c) \quad \text{on } S_t, \quad (43)$$

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

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

3.2.2. L_2 projection The Galerkin method enforces boundary conditions in an “average” sense, *i.e.*

$$\int_{S_u} \mathbf{R}^T \mathbf{u} dS = \int_{S_u} \mathbf{R}^T \bar{\mathbf{u}} dS \quad \text{on } S_u, \quad (44)$$

$$\int_{S_t} \mathbf{R}^T \mathbf{t} dS = \int_{S_t} \mathbf{R}^T \bar{\mathbf{t}} dS \quad \text{on } S_t, \quad (45)$$

where the shape function \mathbf{R} is used as a weighting function. Substituting Eqs.(32) and (33) into the above equations leads to

$$\int_{S_u} \mathbf{R}^T \mathbf{R} \tilde{\mathbf{u}} dS = \int_{S_u} \mathbf{R}^T \bar{\mathbf{u}} dS \quad \text{on } S_u, \quad (46)$$

$$\int_{S_t} \mathbf{R}^T \mathbf{R} \tilde{\mathbf{t}} dS = \int_{S_t} \mathbf{R}^T \bar{\mathbf{t}} dS \quad \text{on } S_t. \quad (47)$$

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

$$\mathbf{A}_1 \tilde{\mathbf{u}} = \mathbf{z}_1 \quad \text{on } S_u, \quad (48)$$

$$\mathbf{A}_2 \tilde{\mathbf{t}} = \mathbf{z}_2 \quad \text{on } S_t, \quad (49)$$

where

$$\mathbf{A}_1 = \int_{S_u} \mathbf{R}^T \mathbf{R} dS \quad \text{on } S_u, \quad (50)$$

$$\mathbf{A}_2 = \int_{S_t} \mathbf{R}^T \mathbf{R} dS \quad \text{on } S_t. \quad (51)$$

and

$$\mathbf{z}_1 = \int_{S_u} \mathbf{R}^T \bar{\mathbf{u}} dS \quad \text{on } S_u, \quad (52)$$

$$\mathbf{z}_2 = \int_{S_t} \mathbf{R}^T \bar{\mathbf{t}} dS \quad \text{on } S_t. \quad (53)$$

Because it is integration-free, the collocation method is more efficient than the L_2 projection. However, the L_2 projection method is more elegant in the case of geometries with corners where special care must be taken for choosing collocation point positions. Moreover, collocation methods may lead to instabilities along interfaces between Dirichlet and Neumann boundary conditions since the equations are satisfied only pointwise.

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 aspect of sensitivity analysis of finite element method with NURBS, it is referred to [43, 50]. In the context of boundary integral equations, three methods are available to conduct sensitivity analysis, 1) finite difference method, 2) adjoint variable method [69, 70], and 3) implicit differentiation method [71, 72, 73]. Finite difference methods are easy to implement but the accuracy is limited. The adjoint variable method uses an adjoint state to obtain a sensitivity expression for each design variable, and is thus particularly useful for a large number of design variables and 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 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

Different from traditional BEM where the singular BIE was differentiated with respect to the design variables, In IGABEM we use the regularized form as the basis of shape sensitivity analysis. By taking the parametric coordinates as the material coordinates which are independent of design variables, the shape differentiation form of the regularized BIE is

$$\begin{aligned}
 & \int_S \{ \dot{T}_{ij}(\mathbf{s}, \mathbf{x}) [u_j(\mathbf{x}) - u_j(\mathbf{s})] + T_{ij}(\mathbf{s}, \mathbf{x}) [\dot{u}_j(\mathbf{x}) - \dot{u}_j(\mathbf{s})] \} dS(\mathbf{x}) \\
 & + \int_S T_{ij}(\mathbf{s}, \mathbf{x}) [u_j(\mathbf{x}) - u_j(\mathbf{s})] [d\dot{S}(\mathbf{x})] \\
 & = \int_S [\dot{U}_{ij}(\mathbf{s}, \mathbf{x}) t_j(\mathbf{x}) + U_{ij}(\mathbf{s}, \mathbf{x}) \dot{t}_j(\mathbf{x})] dS(\mathbf{x}) \\
 & + \int_S U_{ij}(\mathbf{s}, \mathbf{x}) t_j(\mathbf{x}) [d\dot{S}(\mathbf{x})],
 \end{aligned} \tag{54}$$

We use a superimposed dot ($\dot{\cdot}$) to denote 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 weakly singular.

The shape derivatives of field points are given by

$$\dot{\mathbf{x}}(\xi) = \sum_{A=1}^n R_A(\xi) \dot{\mathbf{P}}_A. \tag{55}$$

We discretize the displacement and traction field around the boundary as

$$u_j(\xi) = \sum_{A=1}^n R_A(\xi) \tilde{u}_j^A, \tag{56}$$

$$t_j(\xi) = \sum_{A=1}^n R_A(\xi) \tilde{t}_j^A, \quad (57)$$

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

$$\dot{u}_j(\xi) = \sum_{A=1}^n R_A(\xi) \dot{\tilde{u}}_j^A, \quad (58)$$

$$\dot{t}_j(\xi) = \sum_{A=1}^n R_A(\xi) \dot{\tilde{t}}_j^A. \quad (59)$$

After discretization, Eq. (29) can be expressed by

$$\begin{aligned} & \sum_{A=1}^n \left\{ \int_S \{ \dot{T}_{ij}(\zeta_c, \xi) [R_A(\xi) - R_A(\zeta_c)] J(\xi) \} dS(\xi) \right\} \tilde{u}_j^A \\ & + \sum_{A=1}^n \left\{ \int_S \{ T_{ij}(\zeta_c, \xi) [R_A(\xi) - R_A(\zeta_c)] \dot{J}(\xi) \} dS(\xi) \right\} \tilde{u}_j^A \\ & + \sum_{A=1}^n \left\{ \int_S \{ T_{ij}(\zeta_c, \xi) [R_A(\xi) - R_A(\zeta_c)] J(\xi) \} dS(\xi) \right\} \dot{\tilde{u}}_j^A \\ & = \sum_{A=1}^n \left\{ \int_S [\dot{U}_{ij}(\zeta_c, \xi) R_A(\xi) J(\xi) + U_{ij}(\zeta_c, \xi) R_A(\xi) \dot{J}(\xi)] dS(\xi) \right\} \tilde{t}_j^A \\ & + \sum_{A=1}^n \left\{ \int_S [U_{ij}(\zeta_c, \xi) R_A(\xi) J(\xi)] dS(\xi) \right\} \dot{\tilde{t}}_j^A. \end{aligned} \quad (60)$$

The above integral is also evaluated over parent elements where Gauss-Legendre quadrature rules can be used,

$$\begin{aligned} & \sum_{A=1}^n \left\{ \int_{-1}^{+1} \{ \dot{T}_{ij}(\zeta_c, \xi) [R_A(\xi) - R_A(\zeta)] J(\xi) \} \tilde{J}_e(\tilde{\xi}) d\tilde{S}(\tilde{\xi}) \right\} \tilde{u}_j^A \\ & + \sum_{A=1}^n \left\{ \int_{-1}^{+1} \{ T_{ij}(\zeta_c, \xi) [R_A(\xi) - R_A(\zeta)] \dot{J}(\xi) \} \tilde{J}_e(\tilde{\xi}) d\tilde{S}(\tilde{\xi}) \right\} \tilde{u}_j^A \\ & + \sum_{A=1}^n \left\{ \int_{-1}^{+1} \{ T_{ij}(\zeta_c, \xi) [R_A(\xi) - R_A(\zeta)] J(\xi) \} \tilde{J}_e(\tilde{\xi}) d\tilde{S}(\tilde{\xi}) \right\} \dot{\tilde{u}}_j^A \\ & = \sum_{A=1}^n \left\{ \int_{-1}^{+1} [\dot{U}_{ij}(\zeta_c, \xi) R_A(\xi) J(\xi) + U_{ij}(\zeta_c, \xi) R_A(\xi) \dot{J}(\xi)] \tilde{J}_e(\tilde{\xi}) d\tilde{S}(\tilde{\xi}) \right\} \tilde{t}_j^A \\ & + \sum_{A=1}^n \left\{ \int_{-1}^{+1} [U_{ij}(\zeta_c, \xi) R_A(\xi) J(\xi)] \tilde{J}_e(\tilde{\xi}) d\tilde{S}(\tilde{\xi}) \right\} \dot{\tilde{t}}_j^A. \end{aligned} \quad (61)$$

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

$$\dot{\mathbf{H}}\mathbf{u} + \mathbf{H}\dot{\mathbf{u}} = \dot{\mathbf{G}}\mathbf{t} + \mathbf{G}\dot{\mathbf{t}}. \quad (62)$$

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 $\dot{\mathbf{t}}$.

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

$$\dot{u}_j(\mathbf{x}) = \dot{\bar{u}}_j(\mathbf{x}), \quad \text{on } S_u \quad (63)$$

$$\dot{t}_j(\mathbf{x}) = \dot{\bar{t}}_j(\mathbf{x}), \quad \text{on } S_t \quad (64)$$

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. (62), a final matrix form is obtained as

$$\mathbf{A}\dot{\mathbf{z}} = \dot{\mathbf{f}} + \dot{\mathbf{A}}\mathbf{z} \quad (65)$$

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 the 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. (26) and (27),

$$\dot{U}_{ij}(\mathbf{s}, \mathbf{x}) = \frac{1}{8\pi\mu(1-\nu)} \left[(3-4\nu) \left(\ln \frac{1}{r} \right) \delta_{ij} + (\dot{r}_{,i})r_{,j} + r_{,i}(\dot{r}_{,j}) \right], \quad (66)$$

$$\begin{aligned} \dot{T}_{ij}(\mathbf{s}, \mathbf{x}) = & \frac{-1}{4\pi(1-\nu)} \left(\frac{1}{r} \right) \left\{ \frac{\partial r}{\partial n} [(1-2\nu)\delta_{ij} + 2r_{,i}r_{,j}] \right\} \\ & \frac{-1}{4\pi(1-\nu)} \left(\frac{1}{r} \right) [-(1-2\nu)(r_{,i}n_j - r_{,j}n_i)] \\ & \frac{-1}{4\pi(1-\nu)r} \left\{ \left(\frac{\partial r}{\partial n} \right) [(1-2\nu)\delta_{ij} + 2r_{,i}r_{,j}] - 2 \frac{\partial r}{\partial n} [(\dot{r}_{,i})r_{,j} + r_{,i}(\dot{r}_{,j})] \right. \\ & \left. - (1-2\nu)[(\dot{r}_{,i})n_j + r_{,i}\dot{n}_j - (\dot{r}_{,j})n_i - r_{,j}\dot{n}_i] \right\}, \end{aligned} \quad (67)$$

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

$$\left(\frac{\partial r}{\partial n} \right) = (r_{,i}\dot{n}_i) = (\dot{r}_{,i})n_i + r_{,i}\dot{n}_i, \quad \left(\frac{1}{r} \right) = -\frac{\dot{r}}{r^2}, \quad (68)$$

$$(\dot{r}_{,i}) = \left(\frac{x_i \dot{s}_i - s_i \dot{x}_i}{r} \right) = \frac{(\dot{x}_i - \dot{s}_i)r - (x_i - s_i)\dot{r}}{r^2}, \quad \left(\ln \frac{1}{r} \right) = -\frac{\dot{r}}{r}. \quad (69)$$

The Jacobian is

$$J(\xi) = \sqrt{J_i(\xi)J_i(\xi)}, \quad (70)$$

with

$$J_i(\xi) = \frac{dx_i}{d\xi}. \quad (71)$$

The shape derivative of the Jacobian is given by

$$\dot{J}(\xi) = \frac{\dot{J}_i(\xi)J_i(\xi)}{J(\xi)}. \quad (72)$$

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

$$\dot{n}_i = \left[\frac{\dot{J}_i(\xi)}{J(\xi)} \right] = \frac{\dot{J}_i(\xi)J(\xi) - J_i(\xi)\dot{J}(\xi)}{J^2(\xi)}. \quad (73)$$

The hypersingular fundamental solutions are

$$D_{kij} = \frac{1}{4\pi(1-\nu)r} [(1-2\nu)(r_{,i}\delta_{jk} + r_{,j}\delta_{ki} - r_{,k}\delta_{ij}) + 2r_{,i}r_{,j}r_{,k}], \quad (74)$$

$$\begin{aligned} S_{kij} = & \frac{\mu}{2\pi(1-\nu)r^2} \left\{ 2\frac{\partial r}{\partial n} [(1-2\nu)\delta_{ij}r_{,k} + \nu(r_{,j}\delta_{ik} + r_{,i}\delta_{jk}) - 4r_{,i}r_{,j}r_{,k}] \right\} \\ & + \frac{\mu}{2\pi(1-\nu)r^2} \{ 2\nu(n_i r_{,j}r_{,k} + n_j r_{,i}r_{,k}) \} \\ & + \frac{\mu}{2\pi(1-\nu)r^2} \{ (1-2\nu)(2n_k r_{,i}r_{,j} + n_j \delta_{ik} + n_i \delta_{jk}) - (1-4\nu)n_k \delta_{ij} \}. \end{aligned} \quad (75)$$

Hence the shape derivatives of hypersingular fundamental solutions are

$$\begin{aligned} \dot{D}_{kij}(\mathbf{s}, \mathbf{x}) = & \frac{1}{4\pi(1-\nu)} \left(\frac{\dot{1}}{r} \right) [(1-2\nu)(r_{,i}\delta_{jk} + r_{,j}\delta_{ki} - r_{,k}\delta_{ij}) + 2r_{,i}r_{,j}r_{,k}] \\ & + \frac{1}{4\pi(1-\nu)r} [(1-2\nu)((\dot{r}_{,i})\delta_{jk} + (\dot{r}_{,j})\delta_{ki} - (\dot{r}_{,k})\delta_{ij})] \\ & + \frac{1}{4\pi(1-\nu)r} [2((\dot{r}_{,i})r_{,j}r_{,k} + r_{,i}(\dot{r}_{,j})r_{,k} + r_{,i}r_{,j}(\dot{r}_{,k}))], \end{aligned} \quad (76)$$

$$\dot{S}_{kij}(\mathbf{s}, \mathbf{x}) = \dot{S}_{kij}^1(\mathbf{s}, \mathbf{x}) + \dot{S}_{kij}^2(\mathbf{s}, \mathbf{x}) + \dot{S}_{kij}^3(\mathbf{s}, \mathbf{x}) \quad (77)$$

with

$$\begin{aligned} \dot{S}_{kij}^1(\mathbf{s}, \mathbf{x}) = & \frac{\mu}{2\pi(1-\nu)} \left(\frac{\dot{1}}{r^2} \right) \left\{ 2\frac{\partial r}{\partial n} [(1-2\nu)\delta_{ij}r_{,k} + \nu(r_{,j}\delta_{ik} + r_{,i}\delta_{jk})] \right\} \\ & + \frac{\mu}{2\pi(1-\nu)} \left(\frac{\dot{1}}{r^2} \right) \left\{ 2\frac{\partial r}{\partial n} (-4r_{,i}r_{,j}r_{,k}) + 2\nu(n_i r_{,j}r_{,k} + n_j r_{,i}r_{,k}) \right\} \\ & + \frac{\mu}{2\pi(1-\nu)} \left(\frac{\dot{1}}{r^2} \right) \{ (1-2\nu)(2n_k r_{,i}r_{,j} + n_j \delta_{ik} + n_i \delta_{jk}) \}, \\ & + \frac{\mu}{2\pi(1-\nu)} \left(\frac{\dot{1}}{r^2} \right) \{ -(1-4\nu)n_k \delta_{ij} \}, \end{aligned} \quad (78)$$

$$\dot{S}_{kij}^2(\mathbf{s}, \mathbf{x}) = \frac{\mu}{2\pi(1-\nu)r^2} \left\{ 2 \left(\frac{\dot{\partial r}}{\partial n} \right) [(1-2\nu)\delta_{ij}r_{,k} + \nu(r_{,j}\delta_{ik} + r_{,i}\delta_{jk})] \right\} \quad (79)$$

$$\begin{aligned}
& + \frac{\mu}{2\pi(1-\nu)r^2} \left\{ 2 \left(\frac{\partial r}{\partial n} \right) (-4r_{,i}r_{,j}r_{,k}) \right\} \\
& + \frac{\mu}{2\pi(1-\nu)r^2} \left\{ 2 \left(\frac{\partial r}{\partial n} \right) [(1-2\nu)\delta_{ij}(\dot{r}_{,k}) + \nu((\dot{r}_{,j})\delta_{ik} + (\dot{r}_{,i})\delta_{jk})] \right\} \\
& + \frac{\mu}{2\pi(1-\nu)r^2} \left\{ 2 \left(\frac{\partial r}{\partial n} \right) [-4((\dot{r}_{,i})r_{,j}r_{,k} + r_{,i}(\dot{r}_{,j})r_{,k} + r_{,i}r_{,j}(\dot{r}_{,k}))] \right\}, \\
\dot{S}_{kij}^3(\mathbf{s}, \mathbf{x}) &= \frac{\mu}{2\pi(1-\nu)r^2} \{ 2\nu(\dot{n}_i r_{,j}r_{,k} + n_i(\dot{r}_{,j})r_{,k} + n_i r_{,j}(\dot{r}_{,k})) \} \\
& + \frac{\mu}{2\pi(1-\nu)r^2} \{ 2\nu(\dot{n}_j r_{,i}r_{,k} + n_j(\dot{r}_{,i})r_{,k} + n_j r_{,i}(\dot{r}_{,k})) \} \\
& + \frac{\mu}{2\pi(1-\nu)r^2} \{ (1-2\nu)[2(\dot{n}_k r_{,i}r_{,j} + n_k(\dot{r}_{,i})r_{,j} + n_k r_{,i}(\dot{r}_{,j}))] \} \\
& + \frac{\mu}{2\pi(1-\nu)r^2} \{ (1-2\nu)(\dot{n}_j \delta_{ik} + \dot{n}_i \delta_{jk}) - (1-4\nu)\dot{n}_k \delta_{ij} \}. \tag{80}
\end{aligned}$$

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

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

Table I. The singularity of kernel function sensitivities

4.3. Stress and displacement shape sensitivity recovery

4.3.1. Evaluate sensitivities at interior points After computing the displacement and traction of the boundary by solving Eq. (38), 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. Ignoring body forces, for the interior displacement field the expression is

$$u_i(\mathbf{S}) = \int_S U_{ij}(\mathbf{S}, \mathbf{x}) t_j(\mathbf{x}) - \int_S T_{ij}(\mathbf{S}, \mathbf{x}) u_j(\mathbf{x}) dS(\mathbf{x}), \tag{81}$$

and for the interior stress field the expression is

$$\sigma_{ij}(\mathbf{S}) = \int_S D_{kij}(\mathbf{S}, \mathbf{x}) t_k dS(\mathbf{x}) - \int_S S_{kij}(\mathbf{S}, \mathbf{x}) u_k(\mathbf{x}) dS(\mathbf{x}), \tag{82}$$

where \mathbf{S} denote the interior point, and \mathbf{x} the boundary point. The shape sensitivity at interior points can be obtained by taking shape derivatives of the Somigliana's identities. So the displacement shape sensitivity is

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

$$- \int_S T_{ij}(\mathbf{S}, \mathbf{x}) \dot{u}_j(\mathbf{x}) dS(\mathbf{x}) - \int_S T_{ij}(\mathbf{S}, \mathbf{x}) u_j(\mathbf{x}) [dS(\mathbf{x})], \quad (83)$$

and the stress shape sensitivity is

$$\begin{aligned} \dot{\sigma}_{ij}(\mathbf{S}) = & \int_S \dot{D}_{kij}(\mathbf{S}, \mathbf{x}) t_k(\mathbf{x}) dS(\mathbf{x}) + \int_S D_{kij}(\mathbf{S}, \mathbf{x}) \dot{t}_k(\mathbf{x}) dS(\mathbf{x}) \\ & + \int_S D_{kij}(\mathbf{S}, \mathbf{x}) t_k(\mathbf{x}) [dS(\mathbf{x})] - \int_S \dot{S}_{kij}(\mathbf{S}, \mathbf{x}) u_k(\mathbf{x}) dS(\mathbf{x}) \\ & - \int_S S_{kij}(\mathbf{S}, \mathbf{x}) \dot{u}_k(\mathbf{x}) dS(\mathbf{x}) - \int_S S_{kij}(\mathbf{S}, \mathbf{x}) u_k(\mathbf{x}) [dS(\mathbf{x})]. \end{aligned} \quad (84)$$

4.4. Evaluate stress sensitivities at boundary points

The evaluation of the stress at boundary points can also be done using Somigliana's equations, but a singular integral needs to be computed and an integral 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:

$$\mathbf{u}(\xi) = \sum_{A=1}^n R_A(\xi) \mathbf{u}^A, \quad (85)$$

$$\frac{d\mathbf{u}(\xi)}{d\xi} = \sum_{A=1}^n \frac{dR_A(\xi)}{d\xi} \mathbf{u}^A, \quad (86)$$

$$\mathbf{t}(\xi) = \sum_{A=1}^n R_A(\xi) \mathbf{t}^A. \quad (87)$$

As shown in Fig. 13, 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

$$\hat{\mathbf{x}} = \hat{x}_1 \hat{\mathbf{e}}_1 + \hat{x}_2 \hat{\mathbf{e}}_2. \quad (88)$$

The local tangential vector can be obtained by

$$\hat{\mathbf{e}}_1 = \mathbf{n}, \quad (89)$$

$$\hat{\mathbf{e}}_2 = \frac{\mathbf{m}}{|\mathbf{m}|}, \quad (90)$$

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

$$\mathbf{m} = \frac{d\mathbf{x}(\xi)}{d\xi}. \quad (91)$$

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

$$A = \begin{bmatrix} \hat{\mathbf{e}}_1 \\ \hat{\mathbf{e}}_2 \end{bmatrix} = \begin{bmatrix} n_1 & n_2 \\ -n_2 & n_1 \end{bmatrix}. \quad (92)$$

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 global coordinates,

$$\hat{\epsilon}_{22}(\xi) = \hat{u}_{2,2}(\xi) = \frac{\partial \hat{u}_2}{\partial \xi} \frac{\partial \xi}{\partial \hat{x}_2} = A_{2j} \frac{\partial u_j}{\partial \xi} \frac{\partial \xi}{\partial \hat{x}_2}, \quad (93)$$

with

$$\frac{\partial \xi}{\partial \hat{x}_2} = \frac{1}{|\mathbf{m}|}. \quad (94)$$

The stress tensor in the local coordinate system is

$$\hat{\sigma}_{11} = \hat{t}_1, \quad (95)$$

$$\hat{\sigma}_{12} = \hat{t}_2, \quad (96)$$

$$\hat{\sigma}_{22} = \left(\frac{E}{1-\nu^2} \right) \hat{\epsilon}_{22} + \left(\frac{\nu}{1-\nu} \right) \hat{t}_1. \quad (97)$$

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

$$\sigma_{ij} = A_{ki} A_{nj} \hat{\sigma}_{kn}. \quad (98)$$

Now we consider stress sensitivities. After solving Eq. (65), we can get

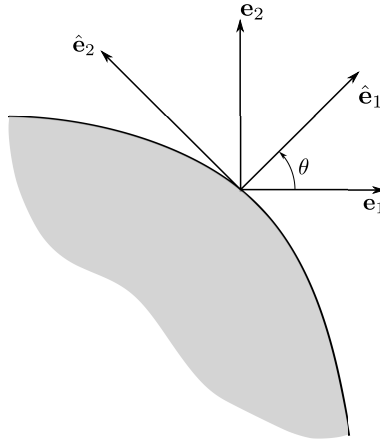


Figure 13. Local coordinate system on curve

$$\dot{\mathbf{u}}(\xi) = R_A(\xi) \dot{\mathbf{u}}^A, \quad (99)$$

$$\frac{d\dot{\mathbf{u}}(\xi)}{d\xi} = \frac{dR_A(\xi)}{d\xi} \dot{\mathbf{u}}^A, \quad (100)$$

$$\dot{\mathbf{t}}(\xi) = R_A(\xi) \dot{\mathbf{t}}^A. \quad (101)$$

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

$$\dot{\mathbf{m}} = \left(\frac{d\mathbf{x}(\xi)}{d\xi} \right), \quad (102)$$

and

$$|\dot{\mathbf{m}}| = \left(\sqrt{\frac{\dot{dx}_i}{d\xi} \frac{dx_i}{d\xi}} \right) = \frac{\left(\frac{dx_i}{d\xi} \right) \frac{dx_i}{d\xi}}{|\mathbf{m}|}, \quad (103)$$

with

$$\left(\frac{d\mathbf{x}(\xi)}{d\xi} \right) = \frac{dR_A(\xi)}{d\xi} \dot{\mathbf{x}}^A. \quad (104)$$

The normalized local tangential vector

$$\dot{\mathbf{e}}_1 = \dot{\mathbf{n}}, \quad (105)$$

$$\dot{\mathbf{e}}_2 = \left(\frac{\dot{\mathbf{m}}}{|\mathbf{m}|} \right). \quad (106)$$

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

$$\begin{aligned} \dot{\epsilon}_{22}(\xi) &= \dot{u}_{2,2}(\xi) = \left(A_{2i} \frac{du_i}{d\xi} \frac{d\xi}{d\hat{x}_2} \right) \\ &= (\dot{A}_{2i}) \frac{du_i}{d\xi} \frac{d\xi}{d\hat{x}_2} + A_{2i} \left(\frac{du_i}{d\xi} \right) \frac{d\xi}{d\hat{x}_2} + A_{2i} \frac{du_i}{d\xi} \left(\frac{d\xi}{d\hat{x}_2} \right), \end{aligned} \quad (107)$$

with

$$\left(\frac{d\xi}{d\hat{x}_1} \right) = \left(\frac{1}{|\mathbf{m}_1|} \right). \quad (108)$$

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

$$\dot{A} = \begin{bmatrix} \dot{\mathbf{e}}_1 \\ \dot{\mathbf{e}}_2 \end{bmatrix}. \quad (109)$$

The stress sensitivity tensor in the local coordinate system is

$$\dot{\hat{\sigma}}_{11} = \dot{\hat{t}}_1, \quad (110)$$

$$\dot{\hat{\sigma}}_{12} = \dot{\hat{t}}_2, \quad (111)$$

$$\dot{\hat{\sigma}}_{22} = \left(\frac{E}{1-\nu^2} \right) \dot{\epsilon}_{22} + \left(\frac{\nu}{1-\nu} \right) \dot{\hat{t}}_1. \quad (112)$$

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

$$\begin{aligned} \dot{\sigma}_{ij} &= (A_{ki} \dot{A}_{nj} \hat{\sigma}_{kn}) \\ &= (\dot{A}_{ki}) A_{nj} \hat{\sigma}_{kn} + A_{ki} (\dot{A}_{nj}) \hat{\sigma}_{kn} + A_{ki} A_{nj} (\dot{\hat{\sigma}}_{kn}). \end{aligned} \quad (113)$$

5. SHAPE OPTIMIZATION WITH IGABEM

Shape optimization can be conducted through a gradient-less or gradient-based method. The gradient-less shape optimization does not require evaluating the shape derivatives, but can be slow to converge and thus prohibitively time-consuming. Moreover, gradient-less methods are not supported

by a mathematical theory. Gradient-based methods are thus normally preferred and also used in the present work. Gradient-based shape optimization methods have a well-sound mathematical foundation rooted in optimal-control theory. A shape optimization problem can be formulated as minimizing an objective function

$$f : \mathbb{R}^n \rightarrow \mathbb{R}, \quad (114)$$

$$f : \mathbf{t} \mapsto f(\mathbf{t}), \quad (115)$$

subject to the constraints

$$g_i(\mathbf{t}) \leq 0 \quad \text{for } i = 1, \dots, m, \quad (116)$$

$$t_i^l \leq t_i \leq t_i^u. \quad (117)$$

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. (117) gives side constraints to limit the search region for the optimum, where t_i^l and t_i^u are lower and upper bounds of the 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 the objective functions and constraint functions to find the next value within an iterative algorithm, *i.e.*

$$\left(f_k, \mathbf{g}_i^k, \frac{d}{dt} f_k, \frac{d}{dt} \mathbf{g}_i^k \right) \rightarrow (f_{k+1}, \mathbf{g}_i^{k+1}), \quad (118)$$

where k denotes the k th iteration step, $\frac{d}{dt} f_k$ and $\frac{d}{dt} \mathbf{g}_i^k$ 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.

From Fig. 4(b), we can find that the steps of meshing/remeshing has been removed in IGABEM optimization.

5.1. Shape derivatives of the volume and von Mises stress

The displacement and stress shape sensitivities can be obtained from the procedure demonstrated in Section 4. However, 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 into boundary integrals.

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

$$A = \int_{\Omega} d\Omega = \frac{1}{2} \int_{\Omega} \nabla \cdot \mathbf{x} d\Omega = \frac{1}{2} \int_S \mathbf{x} \cdot \mathbf{n} dS = \frac{1}{2} \int_S \mathbf{x} \cdot \mathbf{n} J(\xi) dS(\xi). \quad (119)$$

So the shape derivatives are

$$\begin{aligned} \dot{A} &= \frac{1}{2} \int_S [\mathbf{x} \cdot \mathbf{n} J(\xi)] dS(\xi) \\ &= \frac{1}{2} \int_S [\dot{\mathbf{x}} \cdot \mathbf{n} J(\xi) + \mathbf{x} \cdot \dot{\mathbf{n}} J(\xi) + \mathbf{x} \cdot \mathbf{n} \dot{J}(\xi)] dS(\xi). \end{aligned} \quad (120)$$

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

$$\sigma_{\text{vm}} = \left(\frac{3}{2} s_{ij} s_{ij} \right)^{\frac{1}{2}}, \quad (121)$$

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

$$s_{ij} = \sigma_{ij} - \frac{1}{3} \sigma_{kk} \delta_{ij}. \quad (122)$$

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

$$\sigma_{\text{vm}} = \sqrt{\sigma_{11}^2 + \sigma_{22}^2 + 3\sigma_{12}^2 - \sigma_{11}\sigma_{22}}, \quad (123)$$

and its sensitivity is given by

$$\dot{\sigma}_{\text{vm}} = \frac{(2\sigma_{11} - \sigma_{22})\dot{\sigma}_{11} + (2\sigma_{22} - \sigma_{11})\dot{\sigma}_{22} + 6\sigma_{12}\dot{\sigma}_{12}}{2\sigma_{\text{vm}}}. \quad (124)$$

5.2. 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 incorporated the method proposed by [50] in IGAFEM to evaluate the shape derivatives of the quantities in refined meshes with respect to the design variables in the meshes before refinement. The approached is reviewed as below:

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

$$\tilde{\mathbf{P}}_A = \begin{cases} \tilde{\mathbf{P}}_1 & A = 1, \\ \alpha_A \tilde{\mathbf{P}}_A + (1 - \alpha_A) \tilde{\mathbf{P}}_{A-1} & 1 < A < m, \\ \tilde{\mathbf{P}}_n & A = m, \end{cases} \quad (125)$$

with

$$\alpha_A = \begin{cases} 1 & A = 1, \\ \frac{\bar{\xi}_A - \xi_A}{\xi_{A+p} - \xi_A} & 1 < A < m, \\ \bar{\mathbf{P}}_n & A = m, \end{cases} \quad (126)$$

where $\bar{\mathbf{P}}$ are the weighted control points in NURBS before refinement, $\bar{\bar{\mathbf{P}}}_A$ the added weighted control points by knot insertion or repetition. Given the shape derivatives in the mesh of $\dot{\bar{\mathbf{P}}}$ with respect to a given design variable, which can be a control point in the same mesh, the shape derivatives of a weighted point $\bar{\bar{\mathbf{P}}}_A$ in the refined mesh can be obtained by taking derivatives in Eq. (126)

$$\dot{\bar{\bar{\mathbf{P}}}}_A = \begin{cases} \dot{\bar{\mathbf{P}}}_1 & A = 1, \\ \alpha_A \dot{\bar{\mathbf{P}}}_A + (1 - \alpha_A) \dot{\bar{\mathbf{P}}}_{A-1} & 1 < A < m, \\ \dot{\bar{\mathbf{P}}}_n & A = m. \end{cases} \quad (127)$$

After that, the control point derivatives $\dot{\bar{\bar{\mathbf{P}}}}_A$ is recovered by dividing the weights of $\dot{\bar{\bar{\mathbf{P}}}}_A$. 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 analysis and the geometry design still share the same model, only in different levels of refinement.

6. NUMERICAL EXAMPLES

We will investigate the performance of IGABEM for sensitivity analysis and shape optimization through some numerical examples. All the geometries are modelled using NURBS. To study the accuracy of numerical results $(\cdot)_h$ against analytical solutions (\cdot) , we define the relative errors as

$$e_{L_2}(\cdot)_h = \frac{\|(\cdot)_h - (\cdot)\|_{L_2}}{\|(\cdot)\|_{L_2}}, \quad (128)$$

and maximum errors as

$$e_{L_\infty}(\cdot)_h = \frac{\|(\cdot)_h - (\cdot)\|_{L_2}}{\|(\cdot)\|_{L_\infty}}, \quad (129)$$

with

$$\|(\cdot)\|_{L_2} = \sqrt{\int_S (\cdot) \cdot (\cdot) dS}, \quad (130)$$

and

$$\|(\cdot)\|_{L_\infty} = \max_{1 \leq i \leq n} |(\cdot)_i|. \quad (131)$$

And the shape sensitivity transition technique is employed. The optimization solver uses the method of moving asymptotes (MMA) [74].

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

$$u_r(r, \theta) = \frac{pa^2}{E(b^2 - a^2)} \left[(1 - \nu)r + \frac{b^2(1 + \nu)}{r} \right], \quad (132)$$

$$\sigma_{rr}(r, \theta) = \frac{pa^2}{b^2 - a^2} \left(1 - \frac{b^2}{r^2} \right), \quad (133)$$

$$\sigma_{\theta\theta}(r, \theta) = \frac{pa^2}{b^2 - a^2} \left(1 + \frac{b^2}{r^2} \right). \quad (134)$$

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

$$\begin{aligned} \dot{u}_r(r, \theta) = & -\frac{2Pa^2b}{E(b^2 - a^2)^2} \left[(1 - \nu)r + \frac{b^2(1 + \nu)}{r} \right] \\ & + \frac{Pa^2}{E(b^2 - a^2)} \left[(1 - \nu)\dot{r} + (1 + \nu)\frac{2br - b^2\dot{r}}{r^2} \right], \end{aligned} \quad (135)$$

$$\dot{\sigma}_{rr}(r, \theta) = \frac{-2a^2bP}{(b^2 - a^2)^2} \left(1 - \frac{b^2}{r^2} \right) + \frac{Pa^2}{b^2 - a^2} \left(\frac{2br^2 - 2b^2r\dot{r}}{r^4} \right), \quad (136)$$

$$\dot{\sigma}_{\theta\theta}(r, \theta) = \frac{-2a^2bP}{(b^2 - a^2)^2} \left(1 + \frac{b^2}{r^2} \right) + \frac{Pa^2}{b^2 - a^2} \left(\frac{2br^2 - 2b^2r\dot{r}}{r^4} \right), \quad (137)$$

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. 14. The geometry is constructed using quadratic NURBS and the minimum number of elements and control points to represent the geometry are shown in Fig. 15. The shape sensitivity analysis is performed using a refined mesh with 8 elements on each segment. Figs. 16 and 17 show the IGABEM solutions of the displacement and stress sensitivities on the bottom edge AB, respectively. Figs. 18 and 19 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 $a + 0.5 \leq r \leq b - 0.5$ and $\theta = \pi/4$. Supposing the domain points to be linearly varied in the radial direction, *i.e.*

$$\dot{r} = \frac{b - r}{b - a}, \quad (138)$$

the displacement and stress sensitivities can be evaluated using Eqs. (83) and (84). The numerical solutions are shown in Figs. 20 and 21, respectively. And the errors are shown in Figs. 22 and 23. An excellent agreement with the analytical solution is seen again.

Fig. 24 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.

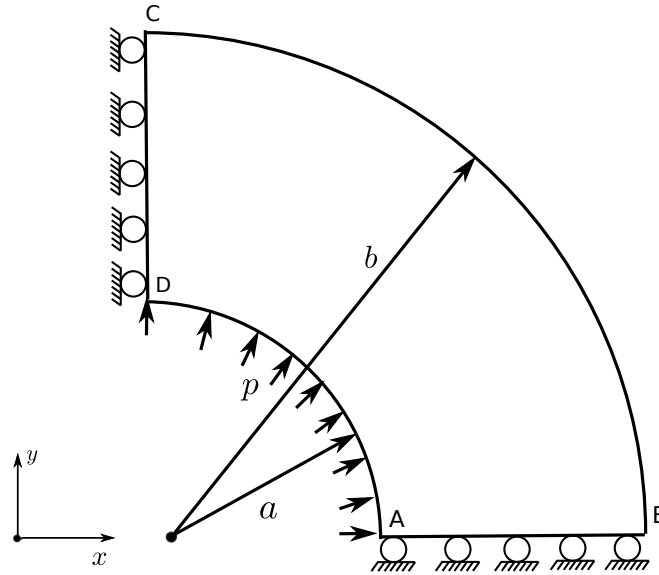


Figure 14. Definition of Lamé's problem

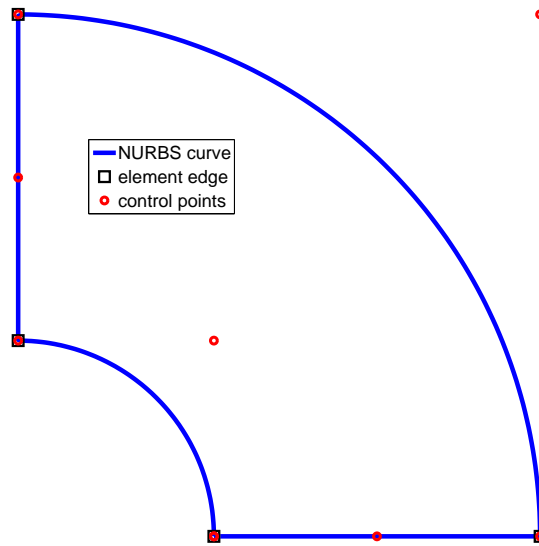


Figure 15. Geometric model of Lamé's problem

6.1.2. 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. 25. 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

$$u_r(r, \theta) = -\frac{Ta^2}{4Gr} \left\{ (1 + K) - (1 - K) \left[4(1 - \nu) - \frac{a^2}{r^2} \right] \cos \theta \right\}, \quad (139)$$

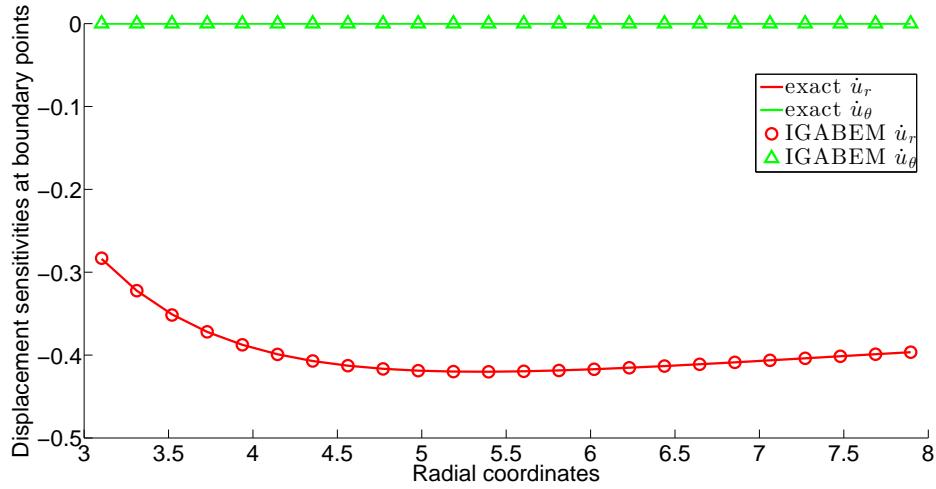


Figure 16. Displacement sensitivities on the boundary points for Lamé's problem

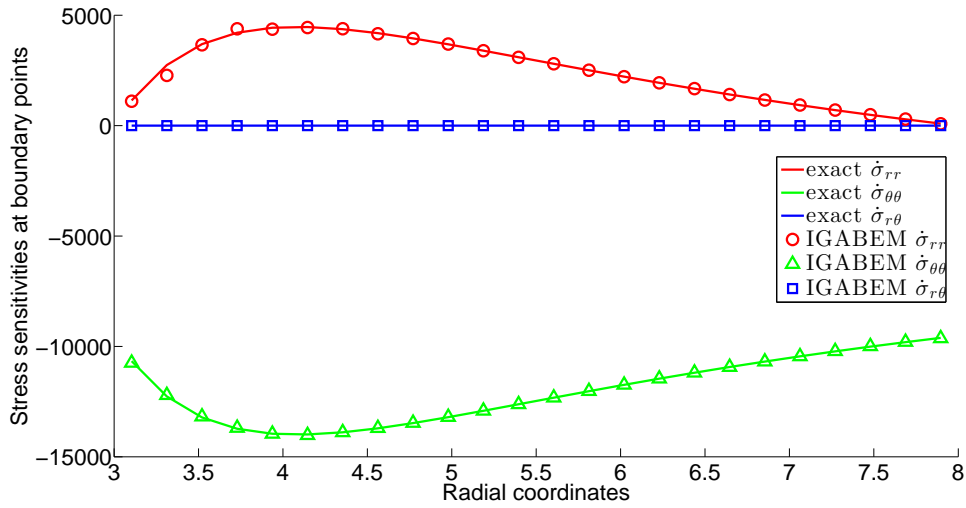


Figure 17. Stress sensitivities on the boundary points for Lamé's problem

$$u_\theta(r, \theta) = -\frac{Ta^2}{4Gr} \left\{ (1-K) \left[2(1-2\nu) + \frac{a^2}{r^2} \right] \sin 2\theta \right\}, \quad (140)$$

and

$$\sigma_{rr}(r, \theta) = \frac{T}{2} \left(1 - \frac{a^2}{r^2} \right) + \frac{T}{2} \left(1 - 4\frac{a^2}{r^2} + 3\frac{a^4}{r^4} \right) \cos 2\theta, \quad (141)$$

$$\sigma_{\theta\theta}(r, \theta) = \frac{T}{2} \left(1 + \frac{a^2}{r^2} \right) - \frac{T}{2} \left(1 + 3\frac{a^4}{r^4} \right) \cos 2\theta, \quad (142)$$

$$\sigma_{r\theta}(r, \theta) = -\frac{T}{2} \left(1 + 2\frac{a^2}{r^2} - 3\frac{a^4}{r^4} \right) \sin 2\theta. \quad (143)$$

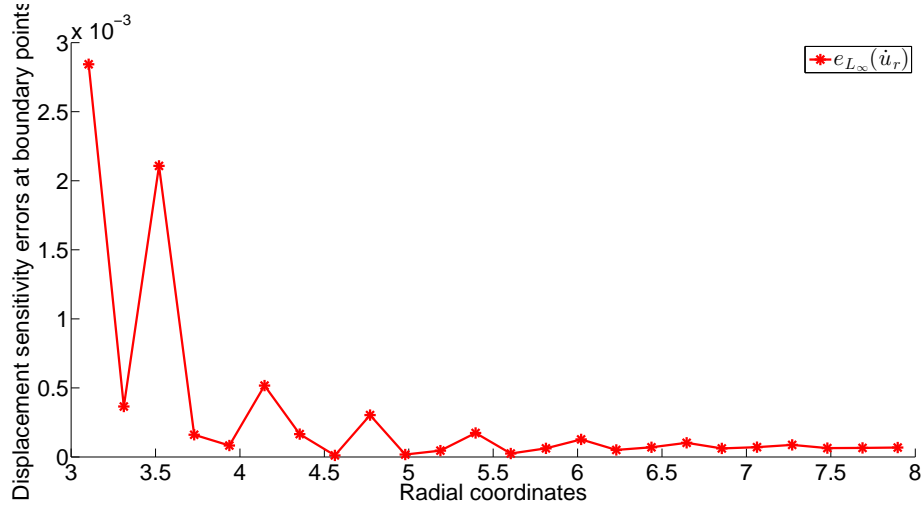


Figure 18. Displacement sensitivity errors on the boundary points for Lamé's problem

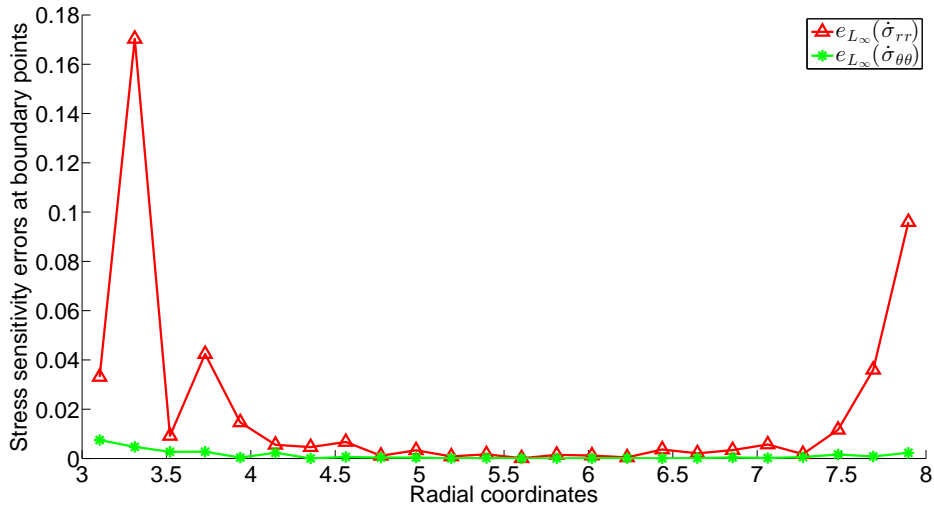


Figure 19. Stress sensitivity errors on the boundary points for Lamé's problem

with

$$K = 3 - 4\nu. \quad (144)$$

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

$$\dot{u}_r(r, \theta) = -\frac{Ta^2}{4G} \left(\frac{1}{r} \right) \left\{ (1+K) - (1-K) \left[4(1-\nu) - \frac{a^2}{r^2} \right] \cos \theta \right\} - \frac{Ta^2}{4Gr} \left\{ -(1-K) \left(\frac{a^2}{r^2} \right) \cos \theta \right\}, \quad (145)$$

$$\dot{u}_\theta(r, \theta) = -\frac{Ta^2}{4G} \left(\frac{1}{r} \right) \left\{ (1-K) \left[2(1-2\nu) + \frac{a^2}{r^2} \right] \sin 2\theta \right\} \quad (146)$$

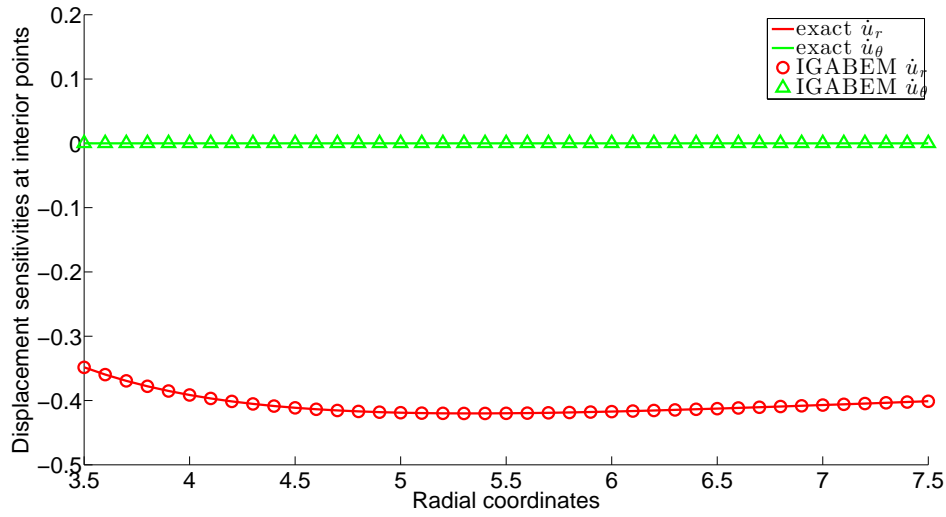


Figure 20. Displacement sensitivities at the interior points for Lamé's problem

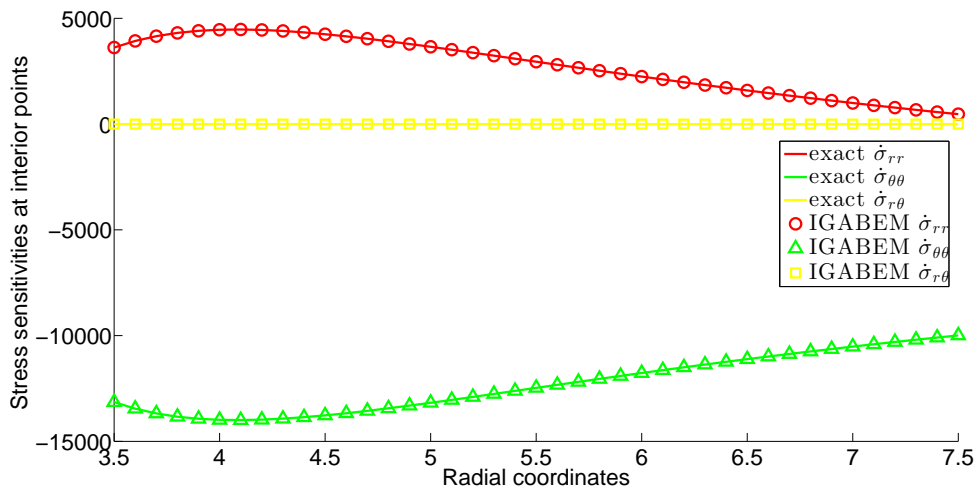


Figure 21. Stress sensitivities at the interior points for Lamé's problem

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

with

$$\begin{aligned} \left(\frac{1}{r} \right) &= -\frac{\dot{r}}{r^2}, \\ \left(\frac{a^2}{r^2} \right) &= \frac{2ar^2 - 2a^2r\dot{r}}{r^4}. \end{aligned} \quad (147)$$

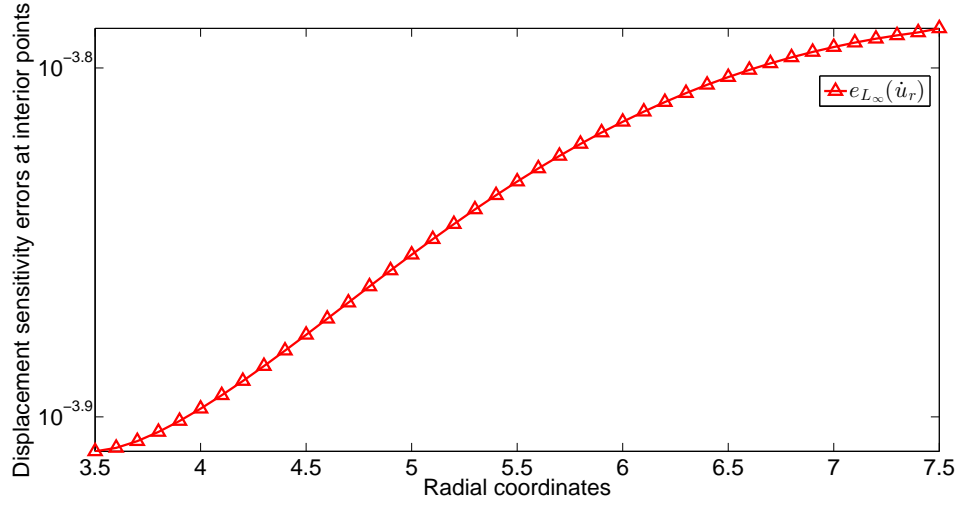


Figure 22. Displacement sensitivity errors at the interior points for Lamé's problem

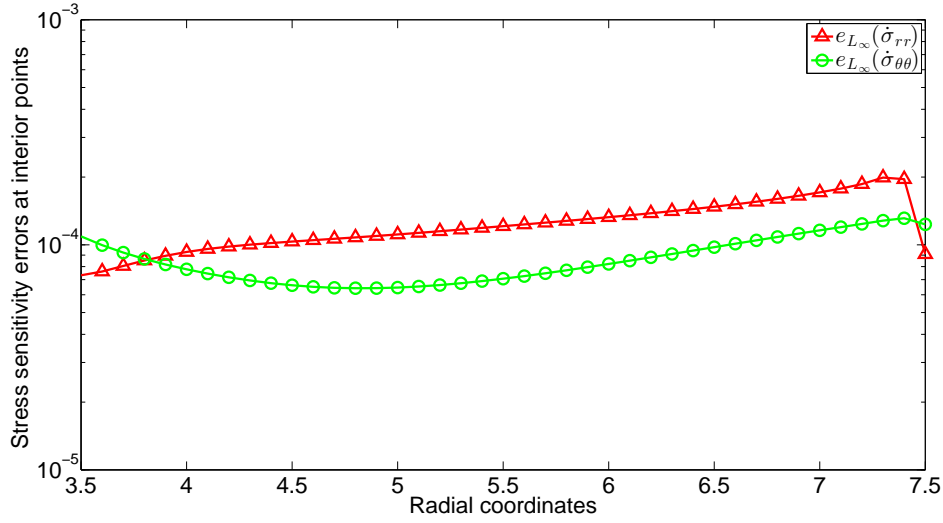


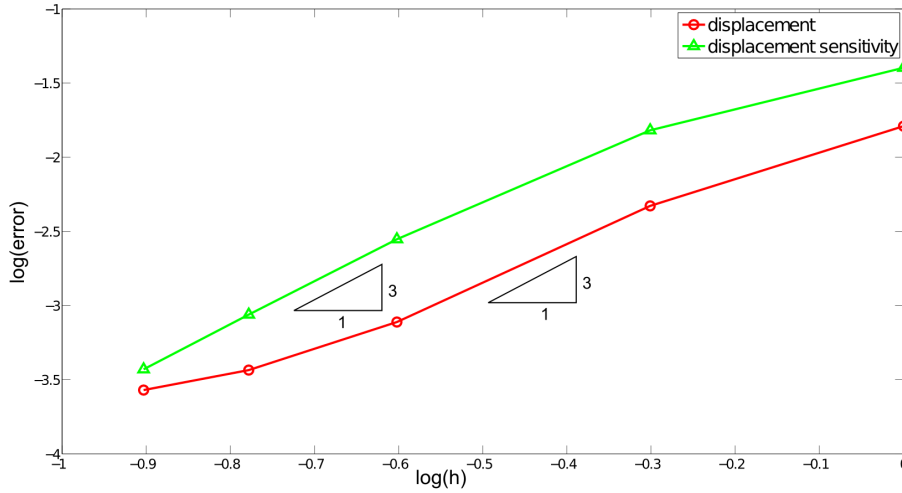
Figure 23. Stress sensitivity errors at the interior points for Lamé's problem

The analytical stress sensitivities are

$$\begin{aligned} \dot{\sigma}_{rr}(r, \theta) = & -\frac{T}{2} \left(\frac{\dot{a}^2}{r^2} \right) + \frac{T}{2} \left[-4 \left(\frac{\dot{a}^2}{r^2} \right) + 3 \left(\frac{\dot{a}^4}{r^4} \right) \right] \cos 2\theta \\ & + \frac{T}{2} \left(1 - 4 \frac{a^2}{r^2} + 3 \frac{a^4}{r^4} \right) (\cos 2\theta), \end{aligned} \quad (148)$$

$$\dot{\sigma}_{\theta\theta}(r, \theta) = \frac{T}{2} \left(\frac{\dot{a}^2}{r^2} \right) - \frac{3T}{2} \left(\frac{\dot{a}^4}{r^4} \right) \cos 2\theta - \frac{T}{2} \left(1 + 3 \frac{a^4}{r^4} \right) (\cos 2\theta), \quad (149)$$

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

Figure 24. $e_{L_2}(\mathbf{u})$ and $e_{L_2}(\dot{\mathbf{u}})$ for Lamé's problem

(150)

with

$$\left(\frac{\dot{a}^2}{r^2}\right) = \frac{2ar^2 - 2a^2r\dot{r}}{r^4}, \quad (151)$$

$$\left(\frac{\dot{a}^4}{r^4}\right) = \frac{4a^3r^4 - 4a^4r^3\dot{r}}{r^8}. \quad (152)$$

Fig. 26 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. 27 and 28 show the IGABEM solutions for the displacement and stress sensitivities on edge AB, respectively. The corresponding errors are shown in Figs. 29 and 30. Figs. 31 and 32 show the displacement and stress sensitivities at the interior points along the line $a + 0.5 \leq r \leq \sqrt{2}L - 0.5$ and $\theta = 3\pi/4$. And the corresponding errors are shown in Figs. 33 and 34. The domain points are assumed to be linearly spaced in the radial direction, *i.e.*

$$\dot{r} = \begin{cases} \frac{L/\cos\theta - r}{L/\cos\theta - a} & \text{for } \theta \geq \frac{3\pi}{4}, \\ \frac{L/\sin\theta - r}{L/\sin\theta - a}, & \text{for } \theta < \frac{3\pi}{4}. \end{cases} \quad (153)$$

The numerical solutions agree with the analytical solutions very well. And the convergence of the structural and sensitivity analysis solutions is shown in Fig. 35.

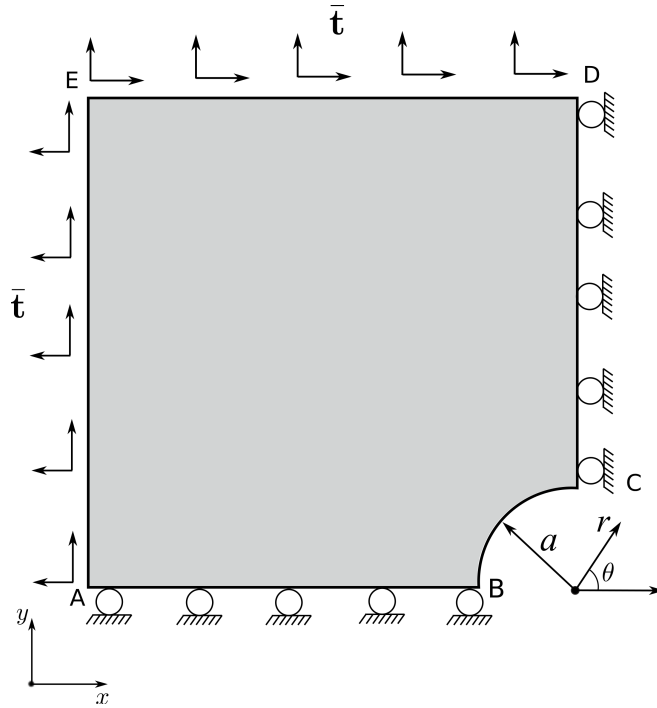


Figure 25. Definition of the Kirsch problem

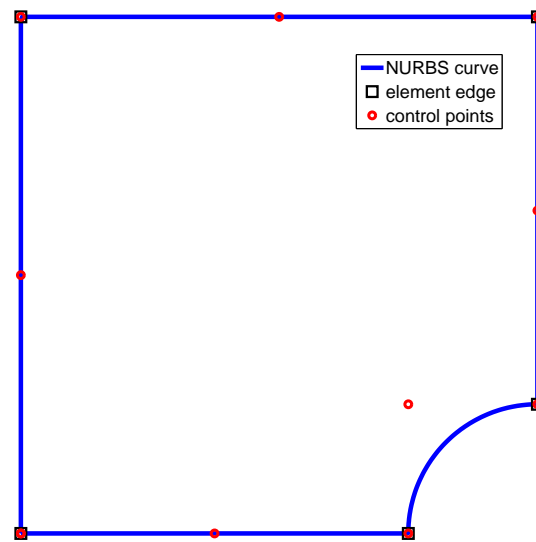


Figure 26. Geometric model of the Kirsch problem

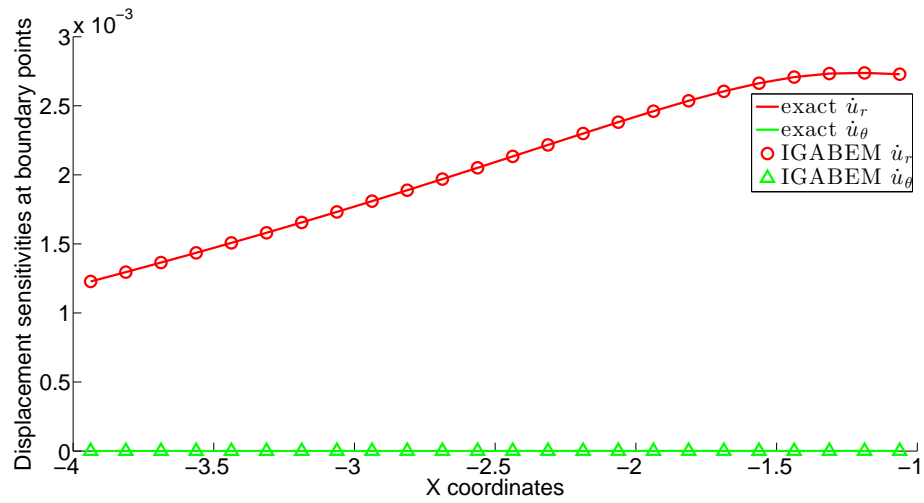


Figure 27. Displacement sensitivities on the edge AB of the plate

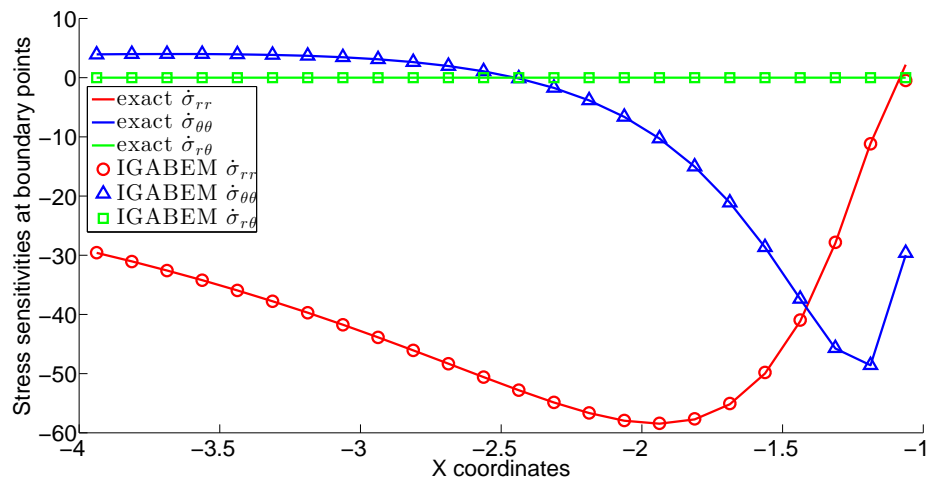


Figure 28. Stress sensitivities on the edge AB of the plate

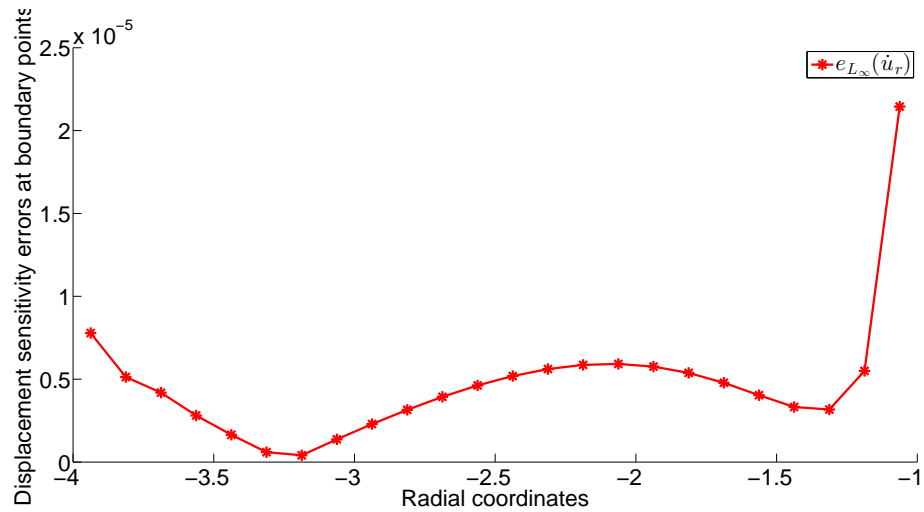


Figure 29. Displacement sensitivity errors on the edge AB of the plate

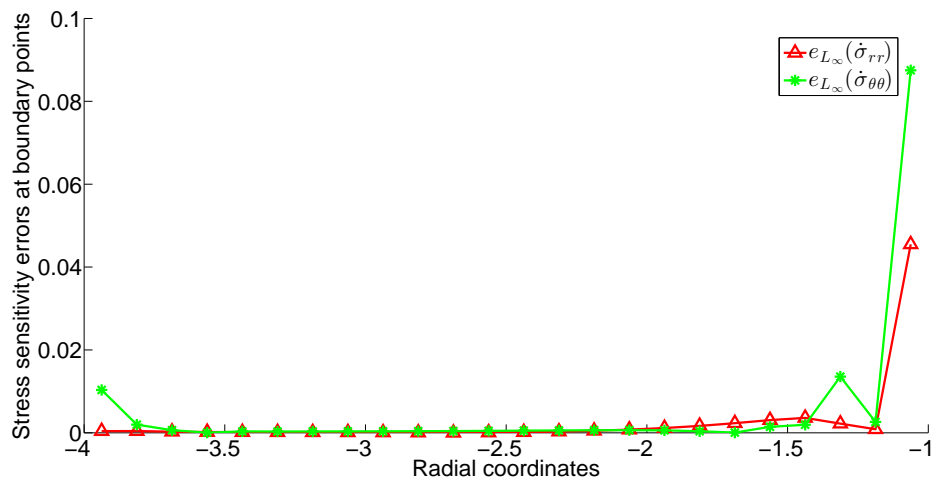


Figure 30. Stress sensitivity errors on the edge AB of the plate

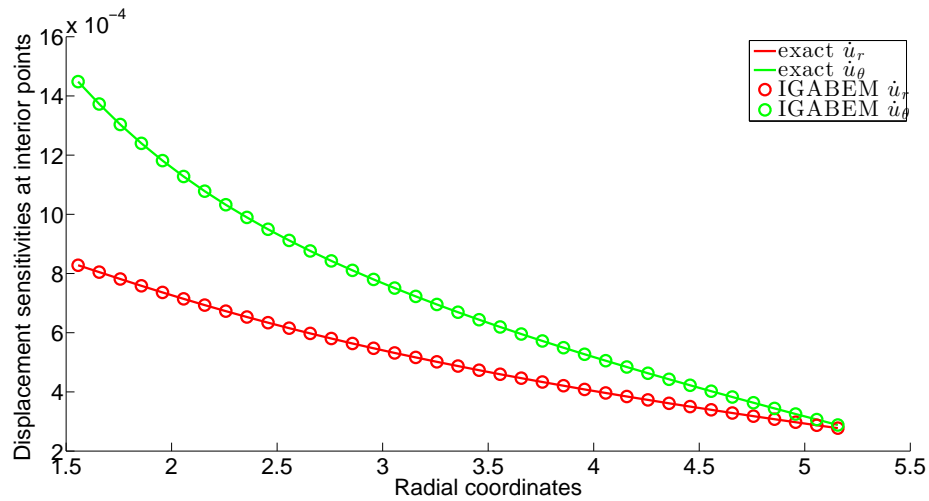


Figure 31. Displacement sensitivities at the interior points of the plate

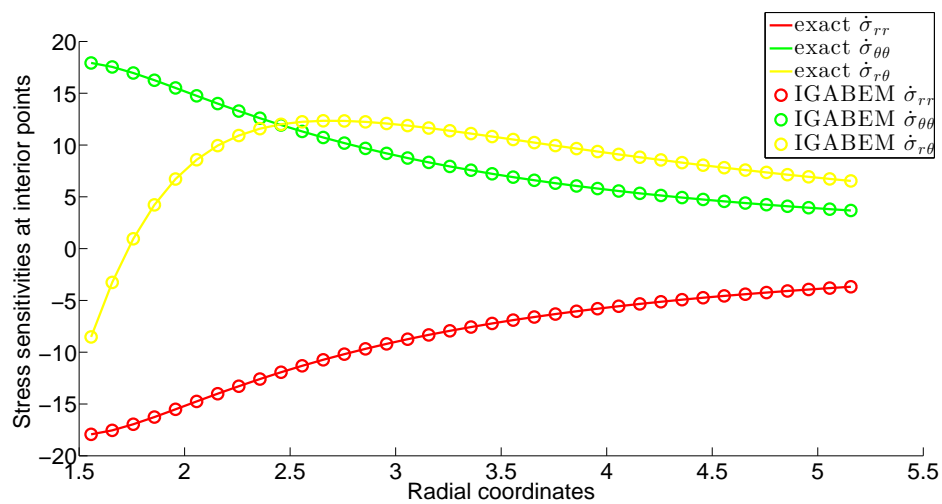


Figure 32. Stress sensitivities at the interior points of the plate

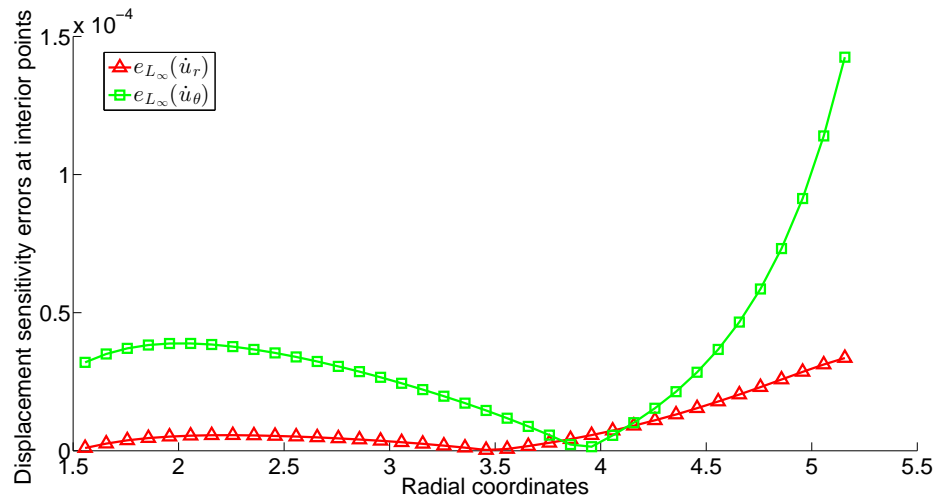


Figure 33. Displacement sensitivities at the interior points of the plate

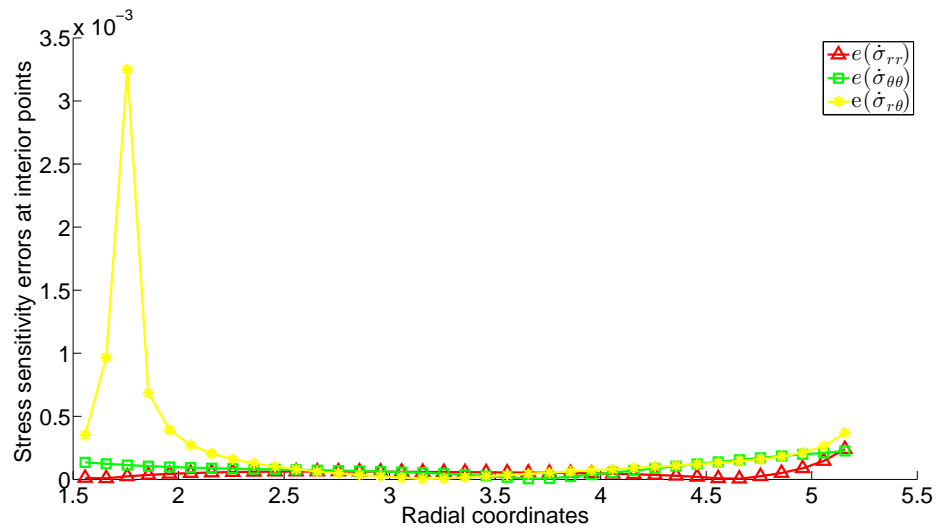


Figure 34. Stress sensitivity errors at the interior points of the plate

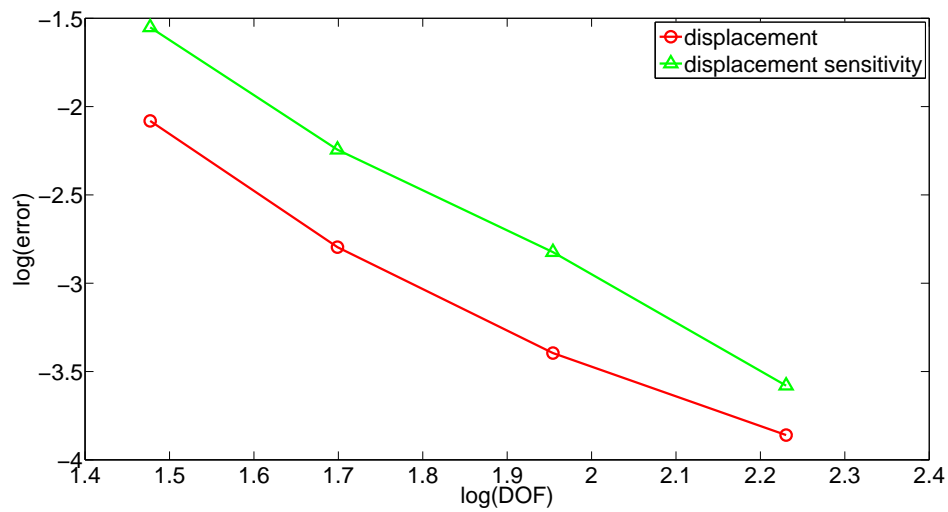


Figure 35. $e_{L_2}(\mathbf{u})$ and $e_{L_\infty}(\dot{\mathbf{u}})$ for the Kirsch problem

6.2. Shape optimization numerical examples

6.2.1. Cantilever beam The problem is a cantilever beam subject to a distributed traction $\bar{t} = 2$ on the beam end (Fig. 36). In the implementation, the traction is imposed on the end element of the beam bottom segment in the design model (Fig. 37). 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. 37. 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. II. The analysis mesh is refined from the design mesh and has 32 elements (Fig. 38). After the iterative procedure (Fig. 40), an optimized design is obtained with the final geometry shown in Fig. 39. 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. II. 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. 41 with Fig. 42, 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 movement of the control points on the beam ends is restricted after reaching the corresponding movement bounds.

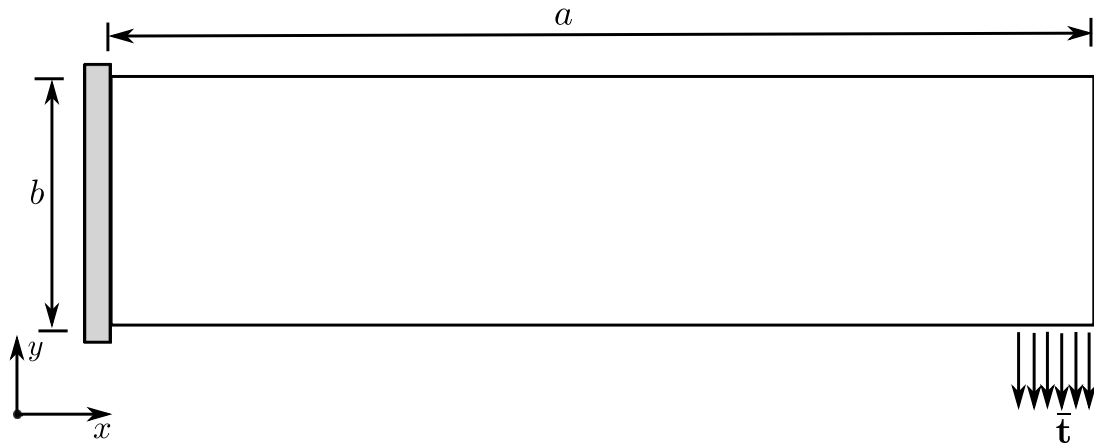


Figure 36. The definition of the cantilever beam problem

6.2.2. Fillet Consider a fillet subject to a traction $\bar{t} = 100$ in the x -direction (see Fig. 43). The objective is to minimize its area while keeping the von Mises stress below the allowable value $\hat{\sigma}_{vm} = 125$. Due to symmetry, only a half model is needed, as shown in Fig. 43. 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. 44 and the coordinates are given by Tab. VI 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

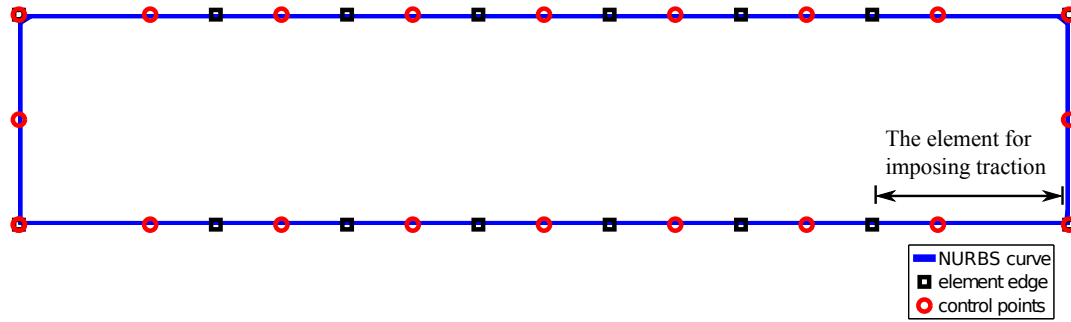


Figure 37. The initial design mesh of the cantilever beam

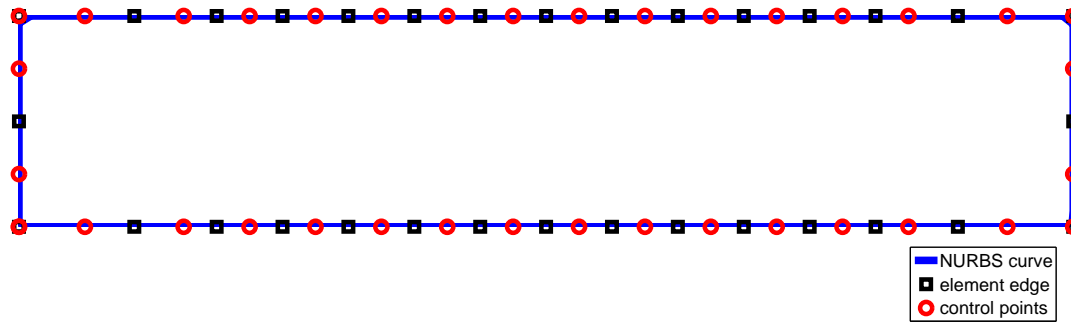


Figure 38. The analysis mesh of the cantilever beam

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 II. Design variables in the cantilever beam optimization procedure

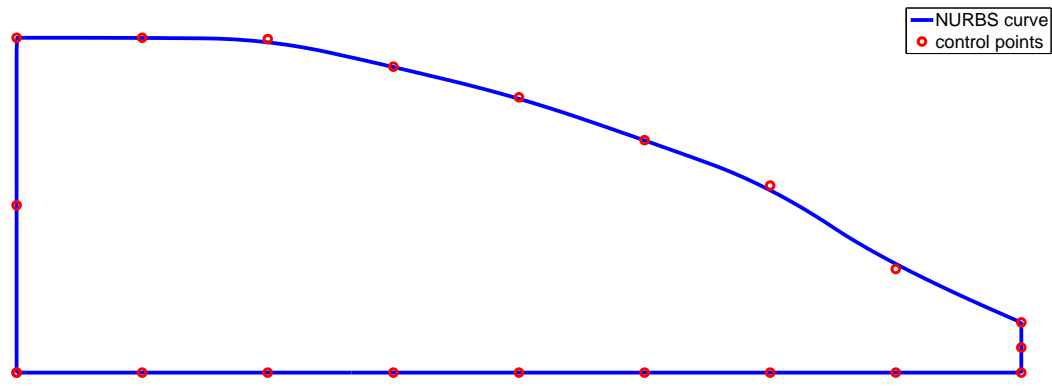


Figure 39. The optimized design for the cantilever beam

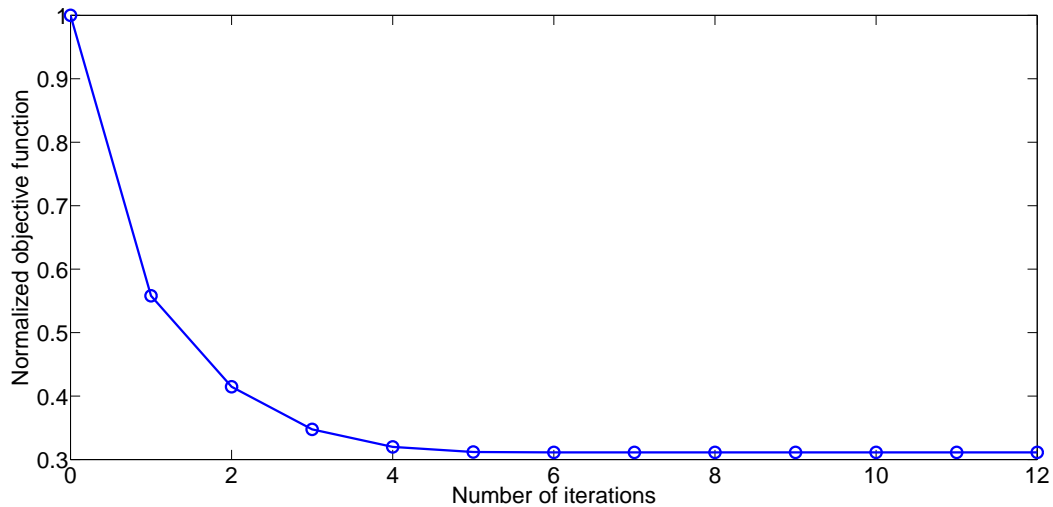


Figure 40. The convergence of the iterative process for the cantilever beam optimization

are set as design variables (Fig. 44). 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. 45, which is used for structural and sensitivity analysis. The optimized design of the fillet is shown in Fig. 46, with the final values of design variables in Tab. III. After the optimization, a smooth stress distribution (Fig. 48) on the monitor points is obtained from 47. The area is reduced to 138.4132 from 145.1602, and the final design agrees with the reported result using the Boundary Contour Method [75] very well. However, the present method requires no meshing procedure.

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 modelled. The geometry of the initial design and the boundary conditions are shown in Fig. 49. The geometry

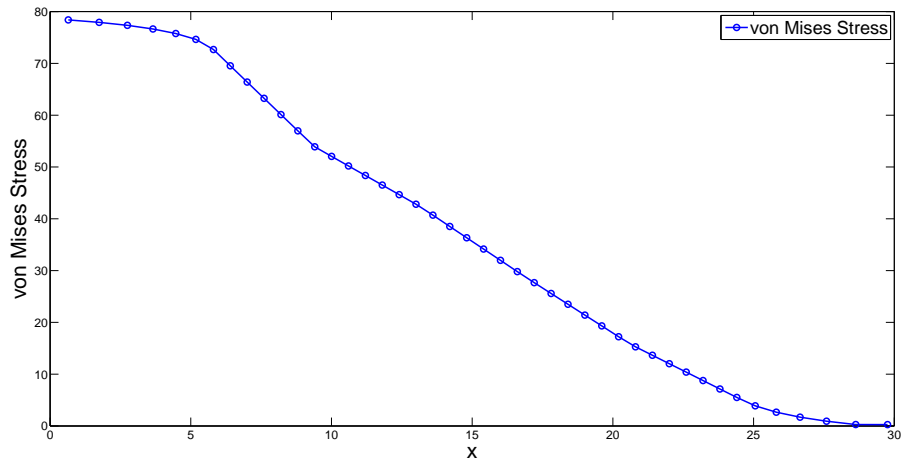


Figure 41. Distribution of von Mises stress on the upper surface before optimization

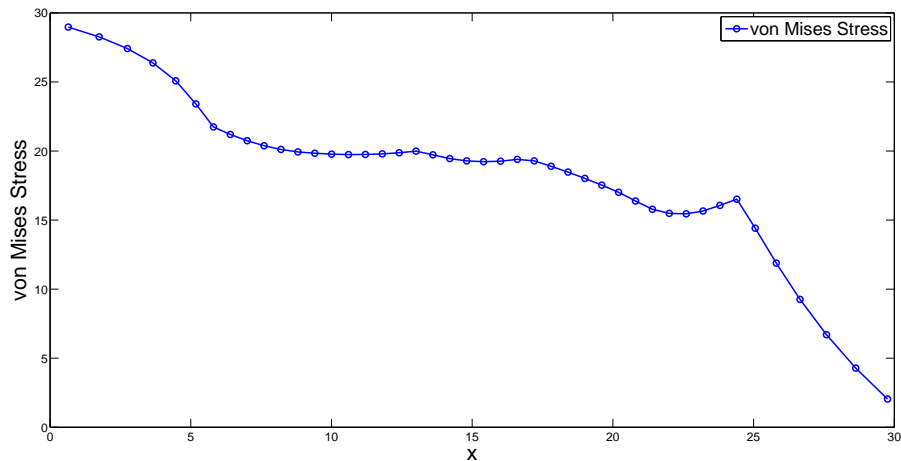


Figure 42. Distribution of von Mises stress on the upper surface after optimization

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 III. Design variables in the fillet optimization procedure

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 L_2 projection method. The initial positions of the control points of the design model are shown in Fig. 50 and the coordinates are given by Tab. VII 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 $\hat{\sigma}_{vm} = 600$. The vertical positions of the four control points on the design curve

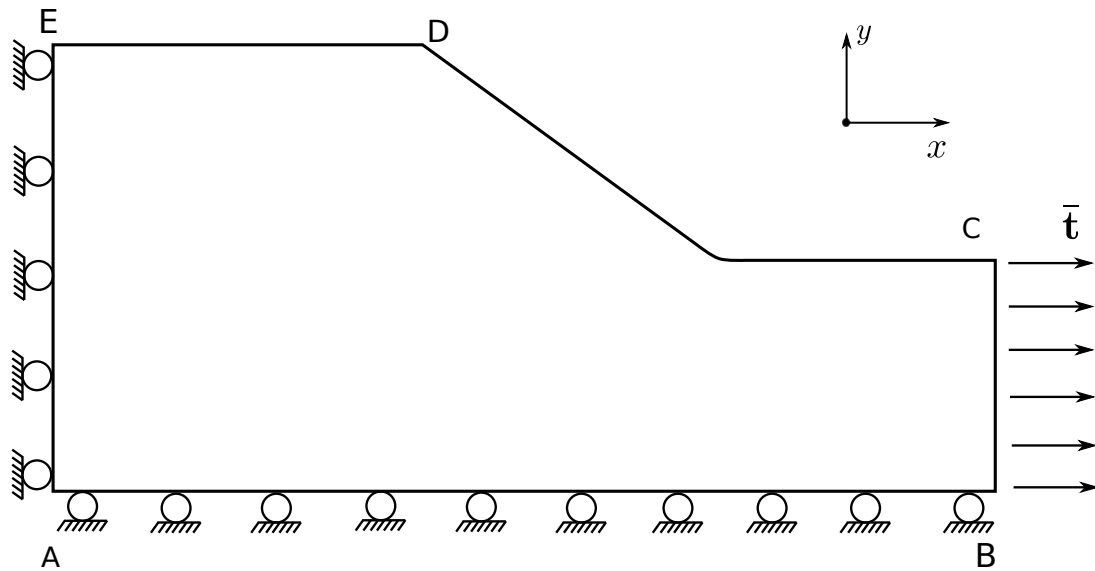


Figure 43. The definition of the fillet problem

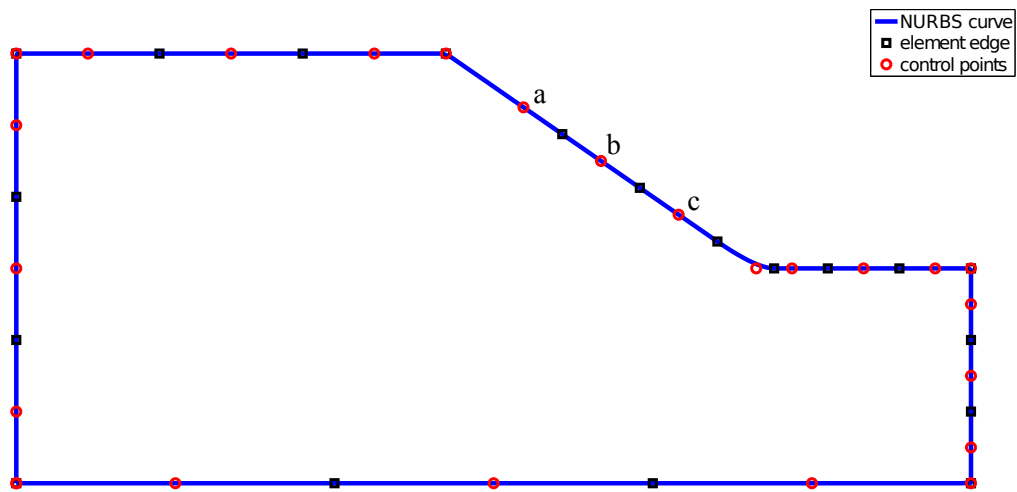


Figure 44. The design mesh of the fillet problem

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. (51). The optimized geometry is shown in Fig. (52), with the coordinates of the converged control points in Tab. IV.

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 IV. Design variables in the connecting rod optimization procedure

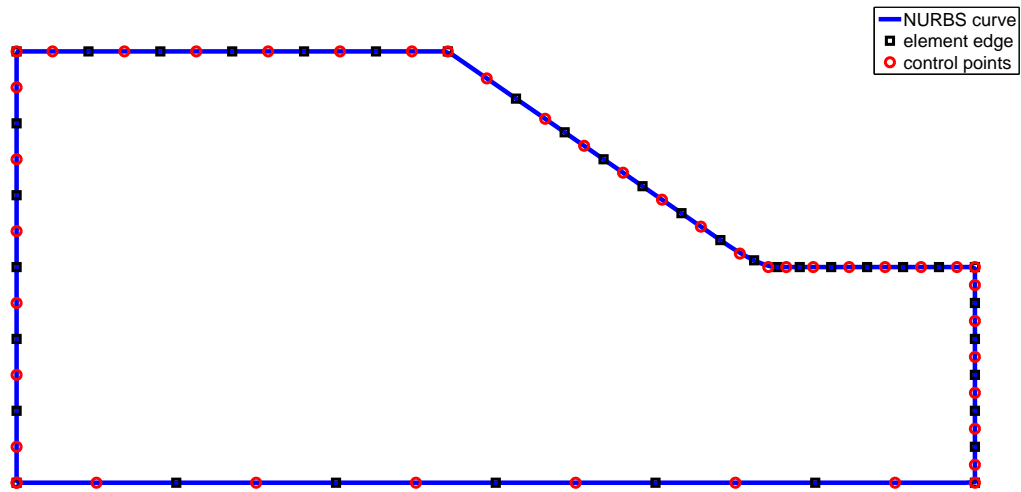


Figure 45. The analysis mesh of the fillet problem

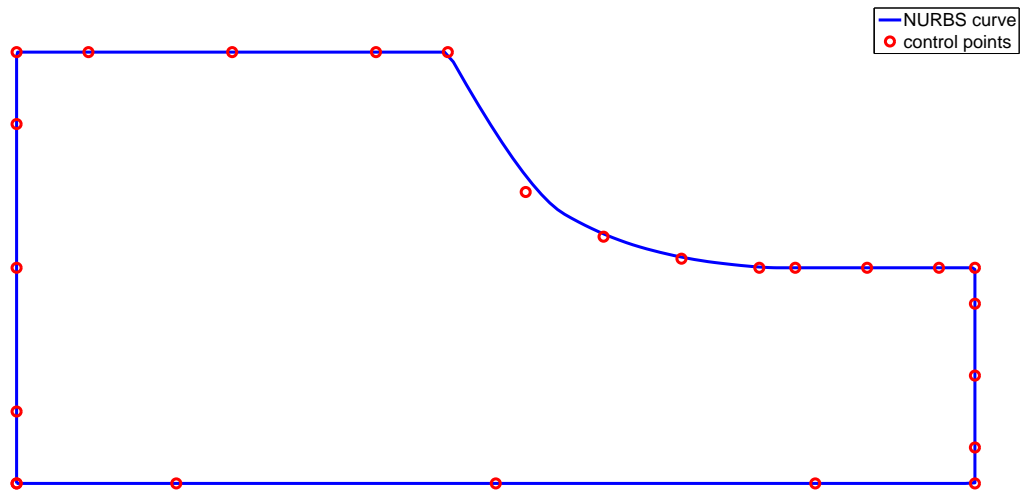


Figure 46. The optimized design of the fillet

6.2.4. Spanner The objective is to minimize the compliance of a spanner subject to the volume constraint $V \leq 90$. The geometry of the initial design and the boundary conditions are shown in Fig. 53. The Young's modulus is $E = 210000$, and Poisson's ratio $\nu = 0.3$. The magnitude of the traction exerted at the spanner end is $p = 100$. The initial positions of the control points of the design model are shown in Fig. 54 and the coordinates are given by Tab. VIII in the appendix. The design variables are the y -coordinates of the points A, B, C, D, E. The points F-K are set to have the same vertical positions as E, and the symmetric condition is enforced for the spanner. The mesh is refined for structural and shape sensitivity analysis, as shown in Fig. (51). The optimized geometry is shown in Fig. (56), with the coordinates of the converged control points in Tab. V.

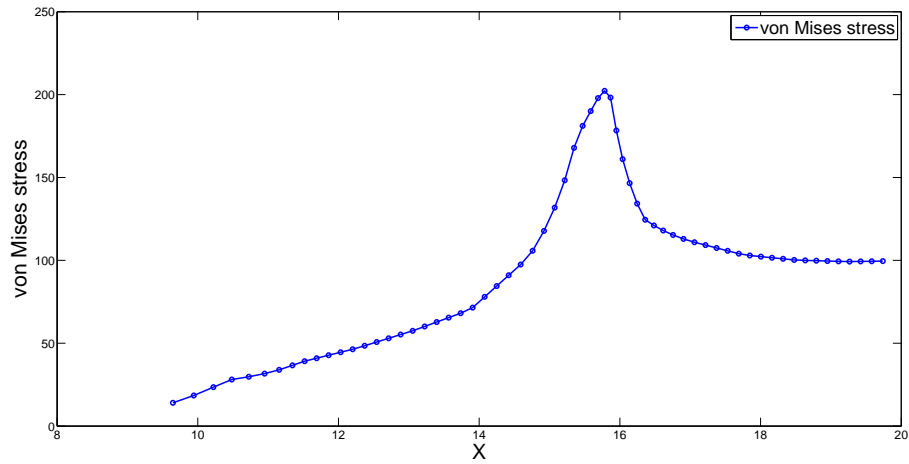


Figure 47. Distribution of von Mises stress on the stress monitor points before the optimization (in the curve segment CD)

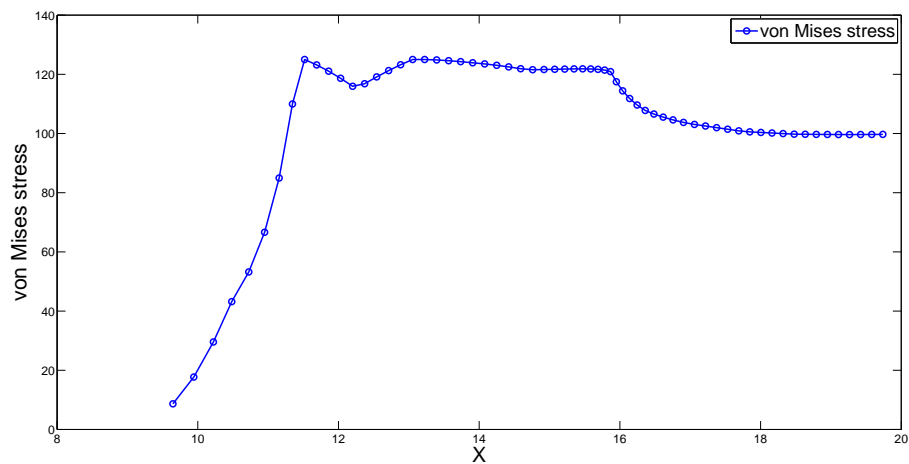


Figure 48. Distribution of von Mises stress on the stress monitor points after the optimization (in the curve segment CD)

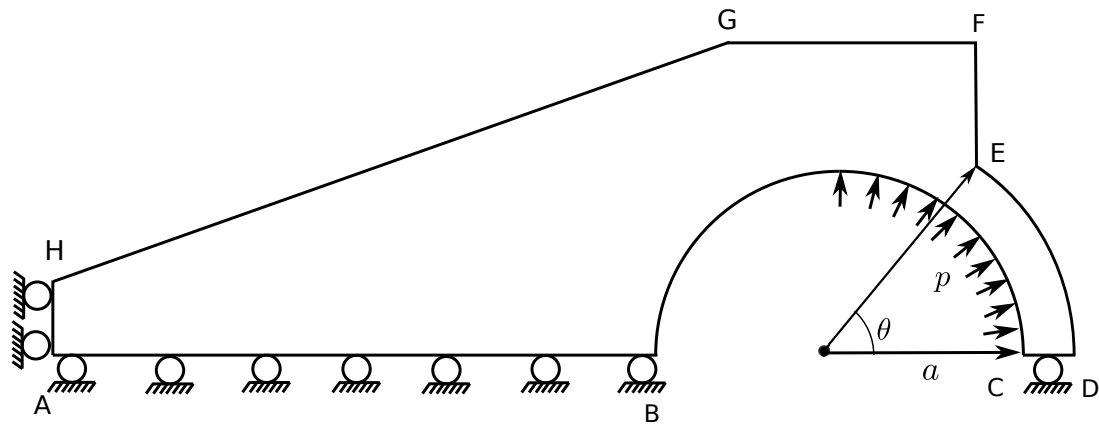


Figure 49. The definition of the connecting rod problem

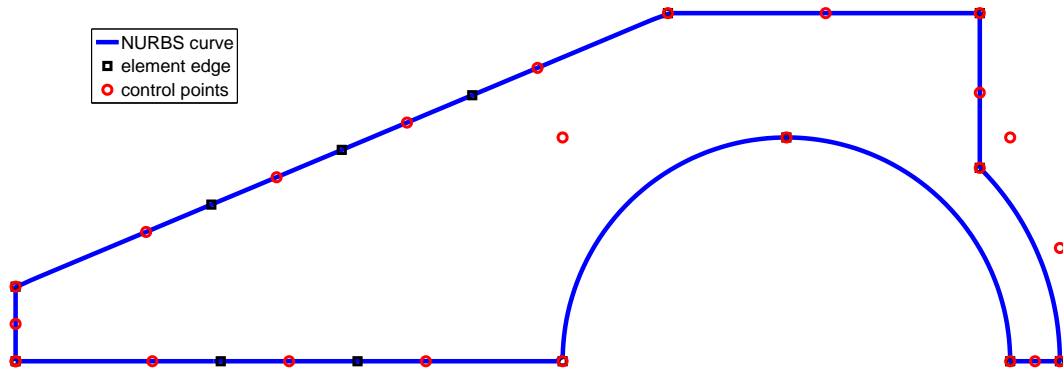


Figure 50. The design mesh of the connecting rod problem

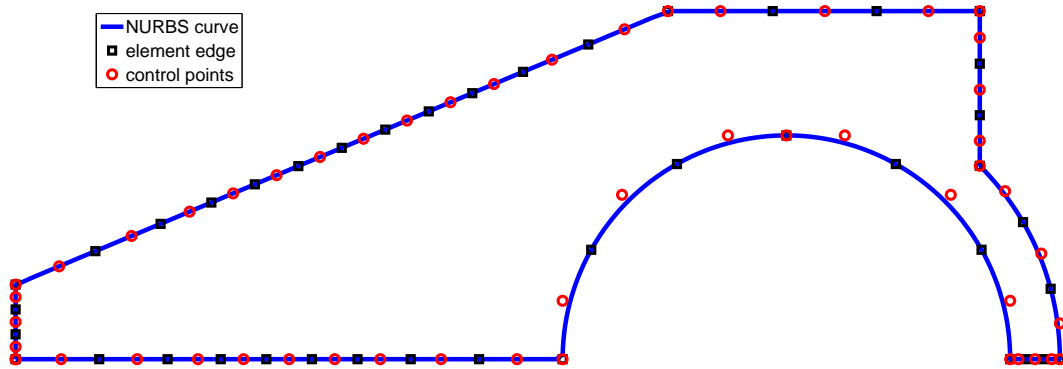


Figure 51. The analysis mesh of the connecting rod problem

Design variable	Lower bound	Upper bound	Initial value	Final value
t_1	5	3	6	3.3931
t_2	5	2	6	2.2976
t_3	5	2	6	2.7169
t_4	5	1	6	2.4159
t_5	5	0.5	6	1.6148

Table V. Design variables in the spanner optimization procedure

7. CONCLUSIONS

Shape sensitivity analysis and optimization is addressed in this paper using the IGABEM. The structure is optimized directly from CAD and the output of the structural shape optimization is a

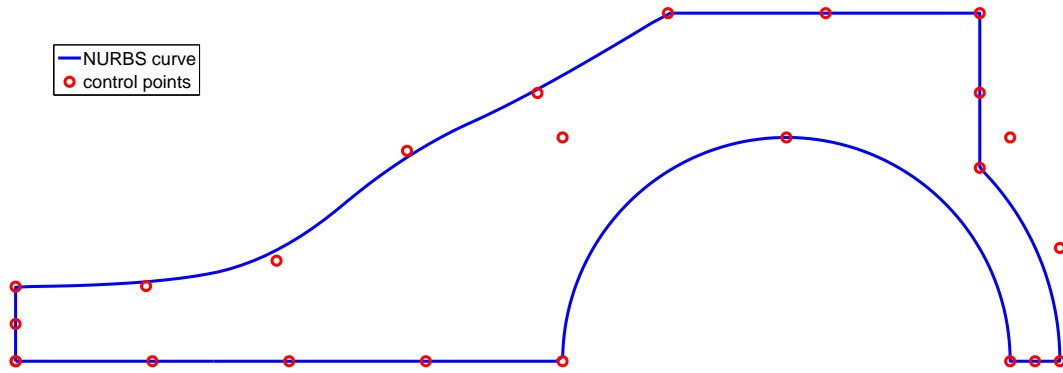


Figure 52. The optimized design of the connecting rod

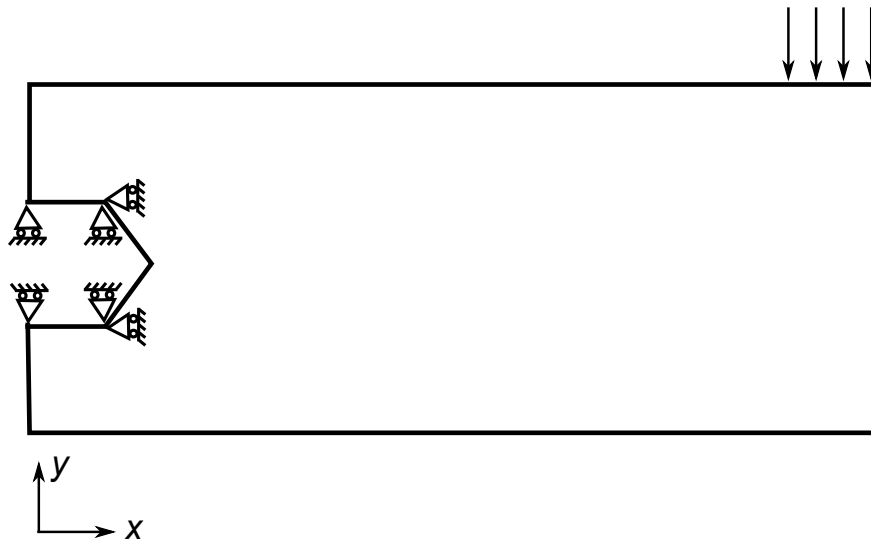


Figure 53. The definition of the spanner problem

CAD model which can be directly visualized in CAD software. IGABEM can achieve a seamless integration between the CAD and analysis, which is particularly significant in shape optimization. The numerical examples show the following advantages:

- The meshing procedure is completely bypassed. This is a significant improvement in computational efficiency, but more importantly in human effort and intervention during the optimization process.
- We incorporated the algorithm in isogeometric finite element method to evaluate the sensitivities of quantities on fine meshes used for analysis with respect to the parameters defined on the coarse meshes provided by CAD. This allows using a coarse mesh for the

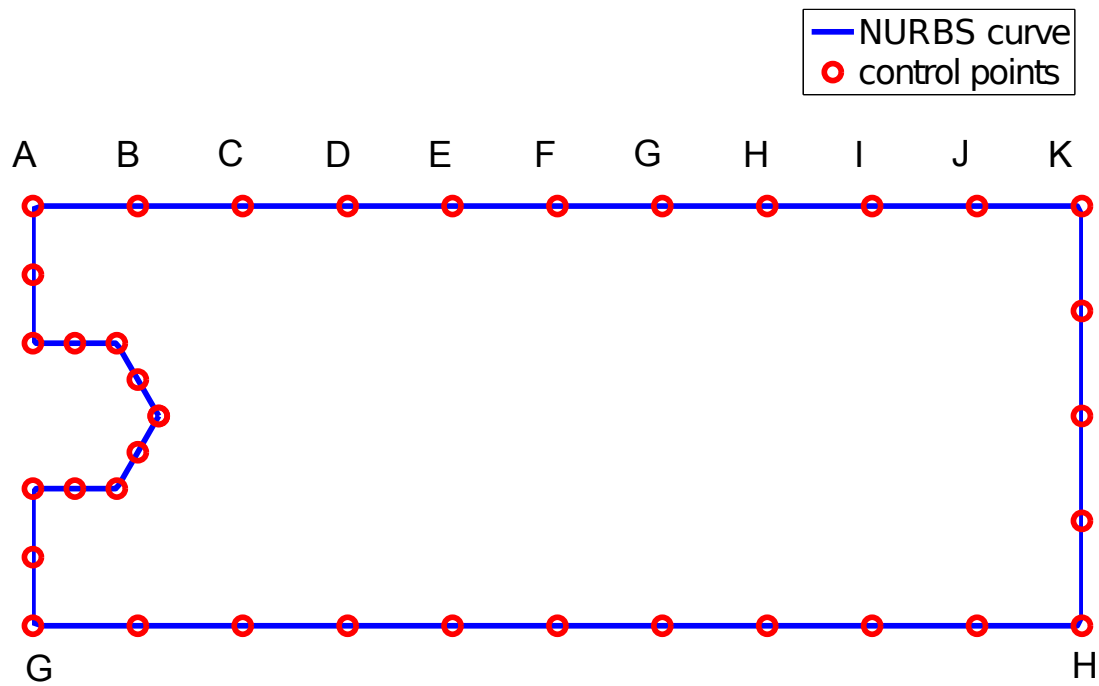


Figure 54. The design mesh of the spanner problem

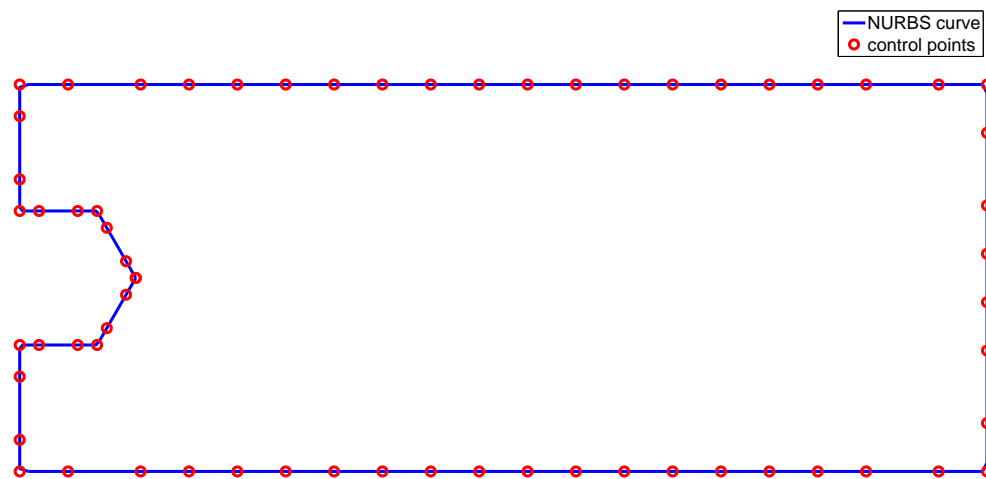


Figure 55. The analysis mesh of the spanner problem

geometry to decrease the number of design variables and fine meshes for the analysis to increase the accuracy of the calculations.

- 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.

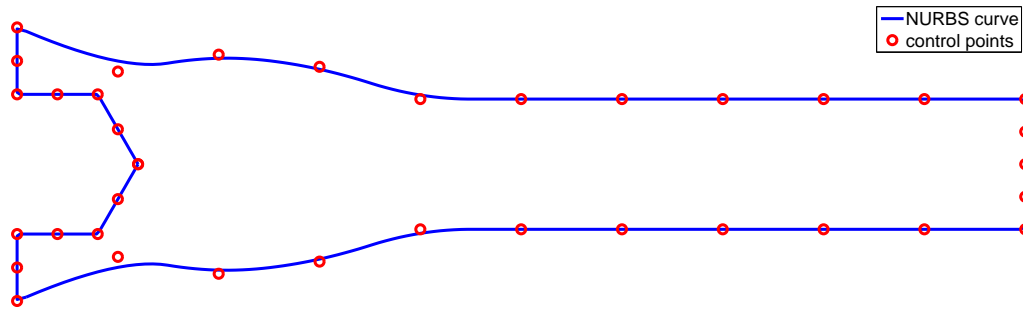


Figure 56. The optimized design of the spanner

Future work includes extension to acoustic and electromagnetic shape optimization, where the advantages of IGABEM can be exhibited fully, without needing to mesh the open domain around the structure. Acceleration algorithms also must be considered to treat more realistic problems. Using geometry independent field approximation (GIFT) would provide added flexibility in the choice of the analysis mesh with respect to the geometrical mesh.

ACKNOWLEDGEMENTS

S.P.A Bordas is 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”.

References

1. Zienkiewicz OC, Taylor RL. *The Finite Element Method*. McGraw-hill London, 1977.
2. Zienkiewicz OC, Campbell JS. Shape optimization and sequential linear programming. *Optimum Structural Design* 1973; :109–126.
3. Lian H, Bordas SPA, Sevilla R, Simpson RN. Recent developments in the integration of computer aided design and analysis. *Computational Technology Reviews* March 2013; **6**:1–36.
4. Nguyen V, Rabczuk T, Bordas S, Duflo M. Meshless methods: A review and computer implementation aspects. *Mathematics and Computers in Simulation* 2008; **79**(3):763–813.
5. Gingold R, Monaghan J. Smoothed particle hydrodynamics: theory and applications to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 1977; **181**:375–389.
6. Belytschko T, Lu YY, Gu L. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering* 1994; **37**:229–256.
7. Liu W, Jun S, Zhang Y. Reproducing kernel particle methods. *International Journal for Numerical Methods in Engineering* 1995; **20**:1081–1106.
8. Atluri SN, Zhu T. A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics* 1998; **22**(2):117–127.
9. Duarte CA, Oden JT. *Hp* clouds-an hp meshless method. *Numerical Methods for Partial Differential Equations* 1996; **12**(6):673–706.
10. Melenk JM, Babuška I. The partition of unity finite element method: basic theory and applications. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**(1):289–314.
11. Bobaru F, Mukherjee S. Meshless approach to shape optimization of linear thermoelastic solids. *International Journal for Numerical Methods in Engineering* 2002; **53**(4):765–796.
12. Bobaru F, Mukherjee S. Shape sensitivity analysis and shape optimization in planar elasticity using the element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**(32):4319–4337.
13. Zhang ZQ, Zhou JX, Zhou N, Wang XM, Zhang L. Shape optimization using reproducing kernel particle method and an enriched genetic algorithm. *Computer Methods in Applied Mechanics and Engineering* 2005; **194**(39):4048–4070.
14. Rosolen A, Millán D, Arroyo M. On the optimum support size in meshfree methods: A variational adaptivity approach with maximum-entropy approximants. *International Journal for Numerical Methods in Engineering* 2010; **82**(7):868–895.
15. Dolbow J, Belytschko T. Numerical integration of the Galerkin weak form in meshfree methods. *Computational Mechanics* 1999; **23**(3):219–230.
16. Belytschko T, Krongauz Y, Organ D, Fleming M, Krysl P. Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**(1):3–47.
17. Beissel S, Belytschko T. Nodal integration of the element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**(1):49–74.
18. Belytschko T, Krongauz Y, Dolbow J, Gerlach C. On the completeness of meshfree particle methods. *International Journal for Numerical Methods in Engineering* 1998; **43**(5):785–819.
19. Chen JS, Wu CT, Yoon S, You Y. A stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering* 2001; **50**(2):435–466.
20. Randles P, Libersky L. Normalized sph with stress points. *International Journal for Numerical Methods in Engineering* 2000; **48**(10):1445–1462.
21. Belytschko T, Guo Y, Liu WK, Xiao SP. A unified stability analysis of meshless particle methods. *International Journal for Numerical Methods in Engineering* 2000; **48**(9):1359–1400.
22. Duflo M, Nguyen-Dang H. A truly meshless Galerkin method based on a moving least squares quadrature. *Communications in Numerical Methods in Engineering* 2002; **18**(6):441–449.
23. Carpinteri A, Ferro G, Ventura G. The partition of unity quadrature in element-free crack modelling. *Computers and Structures* 2003; **81**(18):1783–1794.
24. Cruse TA. A direct formulation and numerical solution of the general transient elastodynamic problem. II. *Journal of Mathematical Analysis and Applications* 1968; **22**(2):341–355.
25. Cruse TA. Numerical solutions in three dimensional elastostatics. *International Journal of Solids and Structures* 1969; **5**:1259–1274.
26. Rizzo FJ. An integral equation approach to boundary value problems of classical elastostatics. *Quarterly of Applied Mathematics* 1967; **25**(1):83–95.
27. Jaswon MA. Integral equation methods in potential theory. I. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 1963; **275**(1360):23–32.

28. Symm GT. Integral equation methods in potential theory. II. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 1963; **275**(1360):33–46.
29. Xin W, Chandra A, Liang-Jenq L, Mukherjee S. Shape optimization in elasticity and elasto-viscoplasticity by the boundary element method. *International Journal of Solids and Structures* 1994; **31**(4):533–550.
30. Yamazaki K, Sakamoto J, Kitano M. Three-dimensional shape optimization using the boundary element method. *AIAA journal* 1994; **32**(6):1295–1301.
31. Yang R. Component shape optimization using BEM. *Computers and Structures* 1990; **37**(4):561–568.
32. Mittal R, Iaccarino G. Immersed boundary methods. *Annual Review of Fluid Mechanics* 2005; **37**:239–261.
33. Glowinski R, Pan T, Periaux J. A fictitious domain method for Dirichlet problem and applications. *Computer Methods in Applied Mechanics and Engineering* 1994; **111**(3-4):283–303.
34. Johansen H, Colella P. A Cartesian grid embedded boundary method for Poisson's equation on irregular domains. *Journal of Computational Physics* 1998; **147**(1):60–85.
35. Saiki E, Biringen S. Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method. *Journal of Computational Physics* 1996; **123**(2):450–465.
36. Ye T, Mittal R, Udaykumar HS, Shyy W. An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics* 1999; **156**(2):209–240.
37. Belytschko T, Black T. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering* 1999; **45**(5):601–620.
38. Bordas SPA, Moran B. Enriched finite elements and level sets for damage tolerance assessment of complex structures. *Engineering Fracture Mechanics* Jan 2006; **73**(9):1176–1201.
39. Bordas SPA, Nguyen PV, Dunant C, Guidoum A, Nguyen-Dang H. An extended finite element library. *International Journal for Numerical Methods in Engineering* Jan 2007; **71**(6):703–732.
40. Duysinx P, Van Miegroet L, Jacobs T, Fleury C. Generalized shape optimization using XFEM and level set methods. *IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials*, Springer, 2006; 23–32.
41. Van Miegroet L, Duysinx P. Stress concentration minimization of 2D filets using X-FEM and level set description. *Structural and Multidisciplinary Optimization* 2007; **33**(4-5):425–438.
42. Nanthakumar S, Lahmer T, Rabczuk T. Detection of multiple flaws in piezoelectric structures using xfem and level sets. *Computer Methods in Applied Mechanics and Engineering* 2014; **275**:98–112.
43. Ghasemi H, Brighenti R, Zhuang X, Muthu J, Rabczuk T. Optimization of fiber distribution in fiber reinforced composite by using nurbs functions. *Computational Materials Science* 2014; **83**:463–473.
44. Moës N, Cloirec M, Cartraud P, Remacle JF. A computational approach to handle complex microstructure geometries. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**(28):3163–3177.
45. Moumnassi M, Belouettar S, Béchet É, Bordas S, Quoirin D, Potier-Ferry M. Finite element analysis on implicitly defined domains: An accurate representation based on arbitrary parametric surfaces. *Computer Methods in Applied Mechanics and Engineering* 2011; **200**(5):774–796.
46. Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 2005; **194**(39–41):4135–4195.
47. Cho S, Ha SH. Isogeometric shape design optimization: exact geometry and enhanced sensitivity. *Structural and Multidisciplinary Optimization* 2009; **38**(1):53–70.
48. Ha SH, Choi K, Cho S. Numerical method for shape optimization using T-spline based isogeometric method. *Structural and Multidisciplinary Optimization* 2010; **42**(3):417–428.
49. Manh ND, Evgrafov A, Gersborg AR, Gravesen J. Isogeometric shape optimization of vibrating membranes. *Computer Methods in Applied Mechanics and Engineering* 2011; **200**(13):1343–1353.
50. Qian X. Full analytical sensitivities in NURBS based isogeometric shape optimization. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**(29-32):2059–2071.
51. Wall WA, Frenzel MA, Cyron C. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering* 2008; **197**(33):2976–2988.
52. Sevilla R, Fernández-Méndez S, Huerta A. NURBS-enhanced finite element method (NEFEM). *International Journal for Numerical Methods in Engineering* 2008; **76**(1):56–83.
53. Sevilla R, Fernández-Méndez S, Huerta A. NURBS-enhanced finite element method (NEFEM): a seamless bridge between CAD and FEM. *Archives of Computational Methods in Engineering* 2011; **18**(4):441–484.
54. Xu G, Atroshchenko E, Bordas SPA. Geometry independent field approximation for spline-based finite element methods. 11th World Congress on Computational Mechanics: Barcelona, 2014.
55. Simpson RN, Bordas SPA, Trevelyan J, Rabczuk T. A two-dimensional isogeometric boundary element method for elastostatic analysis. *Computer Methods in Applied Mechanics and Engineering* 2012; **209-212**:87–100.

56. Scott MA, Simpson RN, Evans JA, Lipton S, Bordas SPA, Hughes TJR, Sederberg TW. Isogeometric boundary element analysis using unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering* 2013; **254**:197–221.
57. Simpson RN, Scott MA, Taus M, Thomas DC, Lian H. Acoustic isogeometric boundary element analysis. *Computer Methods in Applied Mechanics and Engineering* 2014; **269**:265–290.
58. Politis C, Ginnis AI, Kaklis PD, Belibassakis K, Feurer C. An isogeometric BEM for exterior potential-flow problems in the plane. *Proceedings of SIAM/ACM joint conference on geometric and physical modeling*, 2009.
59. Li K, Qian X. Isogeometric analysis and shape optimization via boundary integral. *Computer-Aided Design* 2011; **43**(11):1427–1437.
60. Kostas K, Ginnis A, Politis C, Kaklis P. Ship-hull shape optimization with a t-spline based bem–isogeometric solver. *Computer Methods in Applied Mechanics and Engineering* 2015; **284**:611–622.
61. Cox MG. The numerical evaluation of B-splines. *IMA Journal of Applied Mathematics* 1972; **10**(2):134–149.
62. De Boor C. On calculating with B-splines. *Journal of Approximation Theory* 1972; **6**(1):50–62.
63. Piegl L, Tiller W. *The NURBS Book*. Springer-Verlag: New York, 1997.
64. Liu Y, Rudolphi TJ. Some identities for fundamental solutions and their applications to weakly-singular boundary element formulations. *Engineering Analysis with Boundary Elements* 1991; **8**(6):301–311.
65. Liu Y, Rudolphi TJ. New identities for fundamental solutions and their applications to non-singular boundary element formulations. *Computational Mechanics* 1999; **24**(4):286–292.
66. Liu Y. On the simple-solution method and non-singular nature of the BIE/BEM—a review and some new results. *Engineering Analysis with Boundary Elements* 2000; **24**(10):789–795.
67. Telles JCF. A self-adaptive co-ordinate transformation for efficient numerical evaluation of general boundary element integrals. *International Journal for Numerical Methods in Engineering* 1987; **24**(5):959–973.
68. Nguyen VP, Kerfriden P, Brino M, Bordas SPA, Bonisoli E. Nitsches method for two and three dimensional NURBS patch coupling. *Computational Mechanics* 2014; **53**(6):1163–1182.
69. Choi KK, Haug EJ. Shape design sensitivity analysis of elastic structures. *Journal of Structural Mechanics* 1983; **11**(2):231–269.
70. Longo A, Unzueta J, Schaeidt E, Alvarez A, Anza JJ. A general related variational approach to shape optimum design. *Advances in Engineering Software* 1993; **16**(2):135–142.
71. Barone MR, Yang R. A boundary element approach for recovery of shape sensitivities in three-dimensional elastic solids. *Computer Methods in Applied Mechanics and Engineering* 1989; **74**(1):69–82.
72. Choi JH, Kwak BM. Boundary integral equation method for shape optimization of elastic structures. *International Journal for Numerical Methods in Engineering* 1988; **26**(7):1579–1595.
73. Kane JH, Saigai S. Design sensitivity analysis of solids using BEM. *Journal of Engineering Mechanics* 1988; **114**(10):1703–1722.
74. Svanberg K. The method of moving asymptotes—a new method for structural optimization. *International Journal for Numerical Methods in Engineering* 1987; **24**(2):359–373.
75. Phan AV, Mukherjee S, Mayer JR. Stresses, stress sensitivities and shape optimization in two-dimensional linear elasticity by the boundary contour method. *International Journal for Numerical Methods in Engineering* 1998; **42**(8):1391–1407.

A. THE CONTROL POINTS OF THE INITIAL 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 VI. 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 VII. The control points of the initial geometry of the connecting rod

Index	x	y	weight	Index	x	y	weight
1	3	0	1	19	25	3	1
2	2.5	$-\sqrt{3}/2$	1	20	25	6	1
3	2	$-\sqrt{3}$	1	21	22.5	6	1
4	1	$-\sqrt{3}$	1	22	20	6	1
5	0	$-(6 + \sqrt{3})/2$	1	23	17.5	6	1
6	0	-6	1	24	15	6	1
7	2.5	-6	1	25	12.5	6	1
8	5	-6	1	26	10	6	1
9	7.5	-6	1	27	7.5	6	1
10	10	-6	1	28	5	6	1
11	12.5	-6	1	29	2.5	6	1
12	15	-6	1	30	0	6	1
13	17.5	-6	1	31	0	$(6 + \sqrt{3})/2$	1
14	20	-6	1	32	0	$\sqrt{3}$	1
15	22.5	-6	1	33	1	$\sqrt{3}$	1
16	25	-6	1	34	2	$\sqrt{3}$	1
17	25	-3	1	35	2.5	$\sqrt{3}/2$	1
18	25	0	1	36	3	0	1

Table VIII. The control points of the initial geometry of the spanner