Liu Ning (Orcid ID: 0000-0001-8380-0668)

A geometrically exact isogeometric Kirchhoff plate: feature-preserving automatic meshing and C^1 rational triangular Bézier spline discretizations

Ning Liu, Ann E. Jeffers

Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI, 48109, USA

Abstract

The analysis of the Kirchhoff plate is performed using rational Bézier triangles in isogeometric analysis (IGA) coupled with a feature-preserving automatic meshing algorithm. IGA employs the same basis function for geometric design as well as for numerical analysis. The proposed approach also features an automatic meshing algorithm that admits localized geometric features (e.g., small geometric details, sharp corners) with high resolution. Moreover, the use of rational triangular Bézier splines for domain triangulation significantly increases the flexibility in discretizing spaces bounded by complicated NURBS curves. To raise the global continuity to C^1 for the solution of the plate bending problem, Lagrange multipliers are leveraged to impose continuity constraints. The proposed approach also manipulates the control points at domain boundaries in such a way that the geometry is exactly described. A number of numerical examples consisting of static bending and free vibration analysis of thin plates bounded by complicated NURBS curves are used to demonstrate the advantage of the proposed approach.

Keywords: isogeometric analysis; Kirchhoff plate; Bézier triangle; automatic meshing; feature preserving; Lagrange multiplier

1. Introduction

Numerical modeling and analysis of plates of complicated shapes has continuously been a popular research topic because of the widespread applications of plate structures in various fields. Finite element analysis (FEA) of plates can be categorized into thin plate analysis based on the Kirchhoff plate theory and thick plate analysis based on the Reissner-Mindlin plate theory. The main difference between the two prevailing theories lies in the fact that thin plate analysis assumes that the vector normal to the plate mid-surface remains normal to the mid-surface during deformation and thus does not take into account transverse shear deformations, whereas thick plate analysis does. Due to the fact that Reissner-Mindlin plate elements can be joined with C^0 continuity, the use of very simple basis functions is allowed. On the contrary, in the Kirchhoff plate formulation, because of the presence of second-order derivatives, C^1 continuity is demanded between elements which requires higher order basis functions. For this reason, the C^0 shear deformable Reissner-Mindlin plate element is more propagated in commercial finite

element codes. However, most of the plate structures in reality belong to thin and very thin plates, and the use of C^0 basis functions would usually result in various shear locking problems.

Although the numerical analysis of thin plates is already a very mature field, to exactly describe the plate geometry can be rather difficult and sometimes inaccurate, particularly when the structures have curved boundaries or complicated cutouts are involved. The main reason for this lack of accuracy lies in the fact that the model created from standard FEA, which is represented by Lagrange basis functions, is only an approximation of the original computer-aided design (CAD) model, which is described by Non-Uniform Rational B-spines (NURBS). About a decade ago, isogeometric analysis (IGA) was proposed by Hughes et al. [1] as a novel approach to bridge the gap between design and analysis. By employing the same basis functions used in geometric design to approximate field variables in an isoparametric sense, the models created using IGA possess geometric exactness. Other appealing features of IGA include high-order continuity of basis functions, which further leads to more stable numerical conditioning, faster convergence of solutions, and so on. NURBS-based IGA has since been applied to the analysis of thin and thick plates ([2–8], to name a few).

Nevertheless, NURBS functions, which are the main tool for IGA, exhibit a number of defects. First of all, NURBS *h*-refinement propagates across the entire domain, which compromises the efficiency of the method. Secondly, the control mesh generated is restricted to a quadrilateral shape and is therefore not flexible in discretizing domains of arbitrary topology. Moreover, the smoothness in multi-patch analysis using NURBS is not satisfactory. The patch interface is either C^0-continuous or simply not closed (i.e., non-physical gaps). To regain control of the smoothness across the patch boundaries, additional efforts such as the imposition of geometric constraints [9] or the bending strip method [10] are necessary, which requires extra computational time. On the other hand, a variety of local refinement techniques have been developed to overcome the problematic tensor-product structure of NURBS, such as hierarchical B-splines [11,12], truncated hierarchical B-splines (THB-splines) [13], T-splines [14], locally refined splines [15] and polynomial splines over hierarchical T-meshes [16]. However, the construction of the aforementioned local refinement splines relies on complicated algorithms and the resulting mesh is still dependent on the four-sided geometry. On the contrary, the use of spline basis functions for domain triangulation increases the flexibility in discretizing complex spaces. One way to realize this is to use certain triangle-splitting algorithms such as the Powell-Sabin splines [17–19] and the Clough-Tocher splines [20], depending on particular macro-triangle structures. Higher-order Powell-Sabin splines are also available to triangulate a given space [18,21,22]. However, for a given space for triangulation, the Powell-Sabin triangles are sometimes not unique [23]. Recently, NURBS have been successfully converted to the non-uniform rational Powell-Sabin splines (NURPS) [24]. To exactly recover the boundary NURBS curve of degree p, NURPS of degree p or higher should be used. For the recovery of the interior domain described by NURBS of bi-degree (p1, p2), NURPS of degree p1+p2 or higher should be

used. Powell-Sabin B-splines have been applied to study Kirchhoff-Love plate problems [25] and fracture mechanics [26] with satisfactory results.

An alternative is to construct the domain triangulation through the use of rational Bézier triangles [27]. This approach is more general and does not depend on specific triangle-splitting schemes. Since rational basis functions of the Bézier-Bernstein form are used to represent the parametric space, it has the potential to describe the exact geometry as well. In our work, the C^0 rational Bézier triangles are employed for the representation of the triangulated space. Since the Kirchhoff plate formulation involves second-order derivatives of the basis function, at least C^1 continuity is required. For this reason, the global continuity of the triangular Bézier splines is raised to C^1. Note that the approach we adopted can be used to elevate the splines to any desired continuity C^r. Lagrange multipliers are used to impose the Dirichlet boundary conditions and the continuity constraints. Considering that the use of Lagrange multipliers results in an increase of unknowns in the system equations, which hinders efficiency, an iterative approach for the solution of the Lagrange multiplier augmented system is provided as well.

As to the parameterization of the boundary and interior space of the model, we leverage on the recently developed algorithm TriGA [28]. Specifically, a polygonal approximation of the NURBS boundary is first established through *h*-refinement and a dynamic quadtree decomposition algorithm. This procedure allows us to capture sharp geometric features with very good accuracy. With the polygonal approximation of the original NURBS curves computed, a linear domain triangulation can then be constructed by resorting to the meshing package mesh2d [29] that is available online. After that, the linear triangular elements are raised to cubic such that there are sufficient control points for imposing inter-element continuity constraints. The last step is to replace the control points at boundary with those governing the original NURBS curves. This boundary replacement algorithm is also discussed in [23]. Thus, a geometrically exact domain triangulation admitting sharp geometric features can be established.

A Kirchhoff plate formulation is implemented into the algorithm. To verify our modeling approach, a number of plate models bounded by complicated NURBS curves are investigated in the context of static bending and free vibration analysis. Numerical results prove the accuracy and efficiency of the proposed method.

2. A brief review of NURBS and rational Bézier triangles

In this section, we give a brief review on the fundamentals of Bézier curves [30], NURBS [1] and the construction of rational triangular Bézier spline spaces [28].

2.1. Bézier and NURBS curve

In one dimension, a degree-*n* Bernstein polynomial is defined as follows

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i} \tag{1}$$

where $t \in [0,1]$, $\binom{n}{i} = \frac{n!}{i!(n-i)!}$ is the binomial coefficient.

A Bézier curve $S(t)$ is constructed as a linear combination of $n+1$ Bernstein polynomials $B_i^n(t)$ and the corresponding set of control points b_i, which can be expressed as

$$S(t) = \sum_{i=1}^{n+1} B_i^n(t) b_i \tag{2}$$

The B-spline basis functions are related to the Bernstein basis through a Bézier extraction operator C [31] uniquely defined by a specified knot vector on the parametric space, and is written as

$$N(t) = CB(t) \tag{3}$$

Through projection of the B-splines from $\Re^d$ to $\Re^{d+1}$ using the weights associated with the corresponding control points, a degree-n NURBS curve is then given as

$$R_i^n(t) = \frac{N_i^n(t) w_i}{\sum_{j=1}^{ncp} N_j^n(t) w_j} \tag{4}$$

where ncp denotes the number of control points used to define the curve.

2.2. Rational Bézier triangles

A bivariate Bernstein polynomial can be constructed on a triangular domain as

$$B_{ijk}^n(\mathbf{u}) = \frac{n!}{i!\, j!\, k!} u_1^i u_2^j u_3^k \tag{5}$$

where n is the polynomial order, the triplet (i, j, k) represent the ordinate index that sum to n, and $\mathbf{u} = \{u_1, u_2, u_3\}$ denote the barycentric coordinates of a point in the triangle.

The rational form of the above Bernstein basis functions is written as

$$R_{ijk}^{n} = \frac{B_{ijk}^{n}(\mathbf{u}) w_{ijk}}{\sum_{r+s+t=n} B_{rst}^{n}(\mathbf{u}) w_{rst}} \tag{6}$$

where w is the weight.

Finally, a rational triangular Bézier space can be defined as a linear combination of the rational Bernstein basis functions $R_{ijk}^{n}(\mathbf{u})$ and the corresponding control points b_{ijk}

$$T(\mathbf{u}) = \sum_{i+j+k=n} R_{ijk}^{n}(\mathbf{u}) b_{ijk} \tag{7}$$

Fig. 1 illustrates the control lattice of linear, quadratic and cubic Bézier triangles on the parametric domain, respectively.

3. Automatic domain triangulation based on dynamic quadtree decomposition

Automatic mesh generation has been a very popular topic over the last a few decades and is by itself a very complicated process. Mesh generation using triangular segmentations is a very well developed field, and therefore our work resorts to a number of mesh generation algorithms that are available online, with a special leverage on TriGA [28] and mesh2d [29]. Our goal is to automatically generate a domain triangulation that is capable of capturing the local sharp features with high resolution and maintaining the exact geometry from the input NURBS curves. This is done via a dynamic quadtree decomposition algorithm presented in the work of Engvall and Evans [28]. To make this paper self-contained, we briefly discuss the algorithm. For details of the implementation aspects, the readers are recommended to look at the original paper.

As illustrated in Fig. 2, the automatic mesh generation algorithm can roughly be divided into four steps:

1). **Constructing a polygonal approximation**: the input NURBS curves are firstly subdivided through *h*-refinement (i.e., knot insertion) until a sufficiently close polygon approximating the NURBS curve is generated by connecting control points. This process is guided by a prescribed threshold φ defined as the relative difference between the length of the NURBS curve on each knot span and the length of the polygon approximating it. Mid-span knot insertion is performed until the relative difference on every span is within the given limit.

2). **Generating quadtree background mesh**: the mesh is further refined by evaluating a linear sizing function for every side of the polygonal approximation and subsequently comparing it to the side length. If the ratio exceeds a given threshold (normally 1.5), mid-span knot insertion is performed to enrich the local curve. The process iterates until the above ratio for all the sides is

below a prescribed tolerance. This part of the algorithm can be realized using the mesh2d.m function [29].

3). **Triangulating polygonal domain**: based on the polygonal approximation of the input NURBS boundary curves, a linear domain triangulation can be automatically constructed with any standard meshing tool available in commercial software or open source toolkit (e.g., the meshpoly function in mesh2d).

4). **Elevating polynomial degree and recovering exact boundary**: the global continuity of the polynomials is raised to a desired order (usually bicubic as it is the standard in CAD) and knots that correspond to the polygon vertices are repeated $n-1$ times such that C^0 continuity at the vertices is imposed. Additionally, the control points at the boundary edges are substituted by the control points obtained from *h*-refinement of the original NURBS curves. In this way, the exact boundary is recovered.

4. Enforcing high-order continuity via Lagrange multipliers

In this section, the C^r continuity constraints for inter-element continuity are explicitly defined. In order to impose the continuity constraints, a number of approaches can be used: (1) the master-slave method, (2) penalty method, (3) boundary minimum determining set (BMDS) approach, and (4) Lagrange multipliers. The master-slave method is inferior in handling arbitrary constraints. The penalty method requires careful selection of the penalty weight to avoid ill-conditioning. In the BMDS used in [23], solving for the reduced row echelon form is computationally expensive. Moreover, relaxing the constraints on the boundary vertices by restraining the internal free vertices that have influence on the constrained boundary vertices requires user intervention. In addition, careful selection of the free internal vertices is necessary to avoid inaccurate results. On the other hand, the continuity constraints can be exactly enforced through the use of Lagrange multipliers, but this method increases the size of the problem by the number of constraint equations. In the following, we provide an iterative solution procedure presented in [32] that solves the Lagrange multiplier augmented system without increasing the system size.

In the domain discretized by Bézier triangles, the neighboring triangles are connected with C^0 continuity. However, the formulation of Kirchhoff plate involves second order derivatives of the basis functions, and therefore raising the degree of continuity at the common edges is necessary.

Assuming two adjacent triangles $T(v_1, v_2, v_3)$ and $\tilde{T}(v_4, v_3, v_2)$ that share the edge v_2v_3, they can be joined with C^r differentiability if and only if [27]:

$$\tilde{b}_{\gamma jk} = \sum_{\mu+\nu+\kappa=\gamma} \frac{\gamma!}{\mu!\nu!\kappa!} b_{\mu,k+\nu,j+\kappa} u_1^{\mu} u_2^{\nu} u_3^{\kappa} \tag{8}$$

where $0 \le \gamma \le r$, $\gamma + j + k = n$, $\{u_1, u_2, u_3\}$ are the barycentric coordinates of the vertex v_4 relative to T. An example is shown in Fig. 3 where two bicubic triangles are joined with C^1 continuity.

Collecting the continuity constraints computed in Eq. (8), we can write them in matrix form as

$$\mathbf{Ld} = \mathbf{G} \tag{9}$$

where $\mathbf{L}$ is the matrix containing the coefficients of the constraints, $\mathbf{d}$ is the vector including the ordinate information, and $\mathbf{G}$ is the right-hand side of the continuity constraint equations. $\mathbf{G} = \mathbf{0}$ in the case of enforcing the continuity constraints. We use $\mathbf{G}$ in the derivation to keep the method general. Note that, to avoid the ill-conditioning of the augmented stiffness matrix, in which the $\mathbf{L}$ matrix is not of full rank, a preprocessing step is recommended to sort out the linearly independent rows in $\mathbf{L}$ to use for enforcing higher-order continuity.

The Lagrange multiplier augmented system can be expressed as follows

$$\begin{bmatrix} \mathbf{K} & \mathbf{L}^T \\ \mathbf{L} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{G} \end{bmatrix} \tag{10}$$

where $\mathbf{K}$ is the stiffness matrix, $\mathbf{F}$ is the forcing vector, and $\boldsymbol{\lambda}$ denotes the Lagrange multiplier vector.

When a large number of continuity constraints are to be enforced, solving Eq. (10) can be very costly. Alternatively, an iterative approach can be used to solve the problem without increasing the matrix size. Consider a variant of Eq. (10) where the lower diagonal block of zeros is replaced by a diagonal matrix consisting of small numbers, termed a constraint-scaling diagonal matrix, i.e.

$$\begin{bmatrix} \mathbf{K} & \mathbf{L}^T \\ \mathbf{L} & -\varepsilon\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{G} - \varepsilon\boldsymbol{\lambda} \end{bmatrix} \tag{11}$$

where ε is a small number and $\mathbf{I}$ is the identity matrix.

The above system can be expressed in an iterative form as

$$\mathbf{K}\mathbf{d}^{(i+1)} + \mathbf{L}^T\boldsymbol{\lambda}^{(i+1)} = \mathbf{F} \tag{12}$$

$$\mathbf{L}\mathbf{d}^{(i+1)} - \varepsilon\boldsymbol{\lambda}^{(i+1)} = \mathbf{G} - \varepsilon\boldsymbol{\lambda}^{(i)}$$

where i indicates the i^{th} iteration.

Multiplication of $\mathbf{L}^T$ to the second equation in Eq. (12) and rearranging terms yields

$$\mathbf{L}^T\boldsymbol{\lambda}^{(i+1)} = w\mathbf{L}^T\mathbf{L}\mathbf{d}^{(i+1)} - w\mathbf{L}^T\mathbf{G} + \mathbf{L}^T\boldsymbol{\lambda}^{(i)} \tag{13}$$

where $w = \dfrac{1}{\varepsilon}$ is the weight.

Combining Eq. (13) with the first equation in Eq. (12) results in the following

$$\left(\mathbf{K} + w\mathbf{L}^T\mathbf{L}\right)\mathbf{d}^{(i+1)} = \mathbf{F} + w\mathbf{L}^T\mathbf{G} - \mathbf{L}^T\boldsymbol{\lambda}^{(i)} \tag{14}$$

Taking an initial guess of $\boldsymbol{\lambda}^{(0)} = \mathbf{0}$ yields

$$\mathbf{d}^{(1)} = \left(\mathbf{K} + w\mathbf{L}^T\mathbf{L}\right)^{-1}\left(\mathbf{F} + w\mathbf{L}^T\mathbf{G}\right) \tag{15}$$

Recall from the first equation of Eq. (12) that $\mathbf{F} = \mathbf{K}\mathbf{d}^{(i)} + \mathbf{L}^T\boldsymbol{\lambda}^{(i)}$. Substituting this into Eq. (14) leads to

$$\mathbf{d}^{(i+1)} = \left(\mathbf{K} + w\mathbf{L}^T\mathbf{L}\right)^{-1}\left(\mathbf{K}\mathbf{d}^{(i)} + w\mathbf{L}^T\mathbf{G}\right) \tag{16}$$

Eq. (15) and Eq. (16) can be used to solve the augmented Lagrangian system iteratively. Note that this method essentially combines the penalty method with the Lagrange multiplier method. By solving the problem iteratively, it circumvents the problematic ill-conditioning issue exhibited in the penalty method.

5. Governing equations for the Kirchhoff plate

5.1. Kinematics

Let m_{ij} be the bending moment of a plate and q the external distributed load vector. The equilibrium equation for a Kirchhoff plate can be expressed as

$$m_{ij,ij} = q \tag{17}$$

where the comma indicates differentiation and i, j are the indexes ranging from one to two, since the out-of-plane stresses are assumed to be zero.

Multiplying Eq. (17) with the variation in transverse displacement δd and integrating over the entire domain Ω yields the following

$$\int_\Omega \delta d_{,ij} m_{ij} d\Omega + \int_{\partial\Omega} \left(\delta d m_{ij,j} n_i - Q\right) ds + \int_{\partial\Omega} \left(-\delta d_{,i} m_{ij} n_j + M_i\right) ds = \int_\Omega \delta d q d\Omega \tag{18}$$

where $d_{,ij} = \kappa_{ij}$ is the curvature The second and third terms on the left-hand side are the shear and moment boundary conditions on the boundary $\partial\Omega$, respectively. Neglecting the boundary terms yields the weak form

$$\int_\Omega \delta\kappa_{ij} m_{ij} d\Omega = \int_\Omega \delta d q d\Omega \tag{19}$$

In Eq. (19), the bending moments m_{ij} can be computed as

$$m_{ij} = -\int_{-t/2}^{t/2} \sigma_{ij} z dz \tag{20}$$

where t is the thickness of the plate.

The stress-strain relationship for a homogeneous and isotropic plate is

$$\sigma_{ij} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \dfrac{1-\nu}{2} \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{bmatrix} \tag{21}$$

where E and ν represent the Young's modulus and Poisson's ratio, respectively. Eq. (21) can be written in shorthand as $\sigma_{ij} = \mathbf{C}\varepsilon_{ij}$. The strain vector can be rewritten in terms of transverse displacement as

$$\varepsilon_{11} = -z\frac{\partial^2 d}{\partial x^2},\ \varepsilon_{22} = -z\frac{\partial^2 d}{\partial y^2},\ \varepsilon_{12} = -z\frac{\partial^2 d}{\partial x \partial y} \tag{22}$$

Substituting Eq. (21) and Eq. (22) into Eq. (20) leads to the moment-curvature relationship

$$\begin{bmatrix} m_{11} \\ m_{22} \\ m_{12} \end{bmatrix} = \frac{Et^3}{12\left(1-\nu^2\right)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{bmatrix} \kappa_{11} \\ \kappa_{22} \\ 2\kappa_{12} \end{bmatrix} \tag{23}$$

Combining Eq. (23) and Eq. (19) gives

$$\int_\Omega \delta\kappa_{ij} \mathbf{C} \kappa_{ij} d\Omega = \int_\Omega \delta d q d\Omega \tag{24}$$

where $\mathbf{C}$ is the material matrix shown in Eq. (23).

5.2. Discretized form

The input NURBS geometry is triangulated using the approach discussed in Section 3 along with the rational triangular Bézier splines. Recall from Section 2 that $\mathbf{R}$ is the rational Bézier basis function used to represent a triangular patch. The transverse displacement in one patch can then be represented using the following

$$d = \mathbf{R}\mathbf{d} \tag{25}$$

Differentiating Eq. (25) twice with respect to the physical coordinates results in the expression for the curvature

$$\kappa = \mathbf{B}\mathbf{d} \tag{26}$$

where $\mathbf{B} = \begin{bmatrix} \frac{\partial^2 \mathbf{R}}{\partial x^2} & \frac{\partial^2 \mathbf{R}}{\partial y^2} & 2\frac{\partial^2 \mathbf{R}}{\partial x \partial y} \end{bmatrix}^T$.

Substituting Eq. (25) and Eq. (26) into Eq. (24) leads to the final expression of the weak form

$$\int_\Omega \delta\mathbf{d}^T \mathbf{B}^T \mathbf{C}\mathbf{B}\mathbf{d} d\Omega = \int_\Omega \delta\mathbf{d}^T \mathbf{R} q d\Omega \tag{27}$$

From the virtual work equation (i.e., Eq. (27)), we obtain the stiffness $\mathbf{K}$ and forcing terms $\mathbf{F}$, i.e.,

$$\mathbf{K} = \int_\Omega \mathbf{B}^T \mathbf{C}\mathbf{B} d\Omega \tag{28}$$

$$\mathbf{F} = \int_\Omega \mathbf{R} q d\Omega$$

5.3. Free vibration analysis

For free vibration analysis of the Kirchhoff plate, the weak form of the elastodynamic equilibrium equation is written as follows:

$$\int_{\Omega} \delta\boldsymbol{\varepsilon}^T \mathbf{C}\boldsymbol{\varepsilon} d\Omega + \int_{\Omega} \delta\mathbf{u}^T \rho t \ddot{\mathbf{u}} d\Omega = 0 \tag{29}$$

where ρ is the mass density, $\mathbf{u}$ is the displacement tensor and $\ddot{\mathbf{u}}$ represents the acceleration tensor. The displacement tensor is defined as

$$\mathbf{u} = \begin{bmatrix} u & v & d \end{bmatrix}^T = \begin{bmatrix} -z\dfrac{\partial}{\partial x} & -z\dfrac{\partial}{\partial y} & 1 \end{bmatrix}^T d \tag{30}$$

and the acceleration tensor is obtained from Eq. (30) by differentiating twice in time.

Eq. (29) can be concisely expressed in the form

$$\mathbf{K}\mathbf{d} + \mathbf{M}\ddot{\mathbf{d}} = \mathbf{0} \tag{31}$$

Based on Eq. (25) and integrating over the thickness, the mass matrix takes the form

$$M_{ij} = \int_{\Omega} \rho \left(R_i R_j t + \frac{\partial R_i}{\partial x}\frac{\partial R_j}{\partial x}\frac{t^3}{12} + \frac{\partial R_i}{\partial y}\frac{\partial R_j}{\partial y}\frac{t^3}{12} \right) d\Omega \tag{32}$$

The general solution of Eq. (31) is

$$\mathbf{d} = \mathbf{d}_{\phi} \sin\left(\omega t + \theta\right) \tag{33}$$

where ω is the frequency and $\mathbf{d}_{\phi}$ denotes the eigenmode obtained from the following eigenvalue problem

$$\left(\mathbf{K} - \omega^2 \mathbf{M}\right)\mathbf{d} = \mathbf{0} \tag{34}$$

The above problem essentially amounts to a generalized constrained eigenvalue problem, the solution of which requires special treatment. To solve Eq. (34), a solution procedure presented in [33] is used to compute a constrained stiffness matrix incorporating Lagrange multipliers. The natural frequencies and eigenmodes governing the vibration are further obtained from the eigenvalue analysis.

6. Numerical examples

In this section, four numerical examples of Kirchhoff plates of complicated geometries are demonstrated in the context of static bending and free vibration analysis. For all of the examples shown, a 28-point quadrature rule is used for the integration over the triangle, and a 5-point quadrature rule is employed for the integration over the edges to ensure the accuracy of the solution. The results are compared to analytical solutions, if available, or converged finite element solutions using Abaqus.

6.1. Bending of a simply supported circular plate

To verify the plate formulation, a simply supported circular plate subjected to uniform loading is analyzed. For this example, an exact solution is available in [34] and is reproduced here in Eq. (35). The geometry and material properties are illustrated in Fig. 4 along with the deformed shape.

$$d(r) = -\frac{qr_0^2(3+\nu)r^2}{32D(1+\nu)} + \frac{qr^4}{64D} + \frac{qr_0^4(5+\nu)}{64D(1+\nu)} \tag{35}$$

where r is the radius of the point at query and $D = \frac{Et^3}{12(1-\nu^2)}$.

Four meshes with different thresholds φ are illustrated in Fig. 5, from which it is easy to see that the boundary mesh becomes finer as the threshold value φ gets smaller. The relative error at the center of the plate and the L^2 relative error norm e_{L^2} are measured against the analytical solution. Results are also compared with uniform meshing using Abaqus linear triangular shell element S3 (see Table 1). Note that, Eq. (36) was used to calculate the L^2 norm e_{L^2}. In terms of the relative error at the center of the plate, we observe that our solution with $\varphi \le 3\%$ already outperforms the Abaqus model with 604 nodes. The L^2 relative error norm also shows a faster convergence with our proposed model. Worth noting is that the Abaqus S3 element for thin plate analysis employs five degrees of freedom (DOF) per node, whereas our proposed element has only one DOF per node, which further demonstrates a significant saving in computational cost.

$$e_{L^2} = \frac{\sqrt{\int_\Omega \left(d^h - d\right)^2 d\Omega}}{\sqrt{\int_\Omega d^2 d\Omega}} \tag{36}$$

6.2. Bending of a perforated circular plate

In this example, a more complicated geometry is used to demonstrate the ability of the proposed approach in capturing local geometric features. Specifically, a perforated circular plate with simply supported boundary condition is subjected to uniformly distributed load. The dimensions, loading condition, and the simulation results are shown in Fig. 6. The material properties are the same as in the first example.

To illustrate the capability of the proposed approach in discretizing space bounded by complicated NURBS curves, four meshes of the perforated plate are shown in Fig. 7. As we can see, the holes in the plate are accurately captured. To verify the deformation, our results are compared with the converged solution using the Abaqus linear shell element S3, as listed in Table 2. Again, we observe that the results agree very well.

Note that the relative error at $\varphi \le 3\%$ is fairly large, but it does not indicate that $\varphi \le 3\%$ is not a good setting for all cases. The threshold φ is merely a control parameter relative to the dimension of the local feature. In other words, $\varphi \le 3\%$ is likely to result in very satisfying result if the radius of the holes in the plate is not very small.

6.3. Free vibration of a square plate with an elliptical hole

In this section, the undamped free vibration analysis of a simply supported square plate with an elliptical hole of varying radius is investigated. The dimension of the plate is illustrated in Fig. 8. The thickness of the plate is $t = 0.05m$. The material properties are: Young's modulus $E = 2\times10^{11}\, N/m^2$, Poisson's ratio $\nu = 0.3$ and mass density $\rho = 8000\, kg/m^3$. The mesh generated with $\varphi \le 1\%$ is shown in Fig. 9.

The dimensionless parameter Ω_{nd} is used to measure the natural frequency and is defined as

$$\Omega_{nd} = \left(\frac{\omega^2 \rho t a^4}{D} \right)^{\frac{1}{4}} \tag{37}$$

The solutions are compared with those modeled using the Abaqus linear shell element S4R and are listed in Table 3. As we can see, the results are in very good agreement. The first 10 vibration modes are plotted in Fig. 10 for illustrative purposes.

6.4. Free vibration of a square plate with a heart-shape cutout

In the last example, a simply supported square plate with a heart-shape cutout is used to demonstrate the performance of the proposed plate model. The dimension of the plate is shown in Fig. 11. The thickness of the plate is $t = 0.05m$. The material properties are the same as the third example. Free vibration analysis is conducted, for which a number of reference solutions

are available in the literature [8-10]. The dimensionless parameter Ω_{nd} defined in Eq. (37) is used to measure the natural frequency.

The automatically generated meshes are shown in Fig. 12, in which we can observe that the proposed approach is able to handle sharp geometric corners fairly easily. The natural frequencies and mode shapes of the first 10 modes are listed and plotted in Table 4 and Fig. 13, respectively. As we can see, the free vibration results match very well with those in literature.

7. Concluding remarks

In this paper, we solved the Kirchhoff plate problem using isogeometric analysis (IGA). The parameter space was represented by rational Bézier triangles, and the analysis was further facilitated by an automatic meshing algorithm that admits local geometric features with high resolution. Due to the use of rational Bézier splines, the proposed model was extremely flexible for representing geometries comprised of complex topologies. In addition, the replacement of control points at domain boundaries yielded a geometrically exact model to be analyzed. By resorting to the Lagrange multipliers, the global continuity of the domain triangulation was elevated to C^1, which is suitable for Kirchhoff plate analysis. Numerical examples comprised of static bending and free vibration analysis of plates bounded by complicated NURBS curves verify the accuracy and efficiency of the proposed modeling approach. In the future, we intend to investigate the performance of the developed plate model for stability analysis.

Acknowledgement

This study is supported by the University of Michigan. Any opinions, findings, conclusions or recommendations are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

References

1. Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput Methods Appl Mech Eng* 2005; **194**:4135–4195.

2. Shojaee S, Izadpanah E, Valizadeh N, Kiendl J. Free vibration analysis of thin plates by using a NURBS-based isogeometric approach. *Finite Elem Anal Des* 2012; **61**:23–34.

3. Liu N, Jeffers AE. Isogeometric analysis of laminated composite and functionally graded sandwich plates based on a layerwise displacement theory. *Compos Struct* 2017; **176**:143–153.

4. Thai CH, Nguyen-Xuan H, Nguyen-Thanh N, Le T-H, Nguyen-Thoi T, Rabczuk T. Static, free vibration, and buckling analysis of laminated composite Reissner-Mindlin plates using NURBS-based isogeometric approach. *Int J Numer Methods Eng* 2012; **91**:571–603.

5. Beirão Da Veiga L, Hughes TJR, Kiendl J, *et al.* A locking-free model for Reissner–Mindlin plates: Analysis and isogeometric implementation via NURBS and triangular NURPS. *Math*

Model Methods Appl Sci 2015; **25**:1519–1551.

6. Reali A, Gomez H. An isogeometric collocation approach for Bernoulli-Euler beams and Kirchhoff plates. *Comput Methods Appl Mech Eng* 2015; **284**:623–636.

7. Fantuzzi N, Tornabene F. Strong Formulation Isogeometric Analysis (SFIGA) for laminated composite arbitrarily shaped plates. *Compos Part B Eng* 2016; **96**:173–203.

8. Tornabene F, Fantuzzi N, Bacciocchi M. A new doubly-curved shell element for the free vibrations of arbitrarily shaped laminated structures based on Weak Formulation IsoGeometric Analysis. *Compos Struct* 2017; **171**:429–461.

9. Liu N, Jeffers AE. Adaptive isogeometric analysis in structural frames using a layer-based discretization to model spread of plasticity. *Comput Struct* 2018; **196**:1–11.

10. Kiendl J, Bazilevs Y, Hsu MC, Wüchner R, Bletzinger KU. The bending strip method for isogeometric analysis of Kirchhoff-Love shell structures comprised of multiple patches. *Comput Methods Appl Mech Eng* 2010; **199**:2403–2416.

11. Vuong A V., Giannelli C, Jüttler B, Simeon B. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Comput Methods Appl Mech Eng* 2011; **200**:3554–3567.

12. Schillinger D, Rank E. An unfitted hp-adaptive finite element method based on hierarchical B-splines for interface problems of complex geometry. *Comput Methods Appl Mech Eng* 2011; **200**:3358–3380.

13. Giannelli C, Jüttler B, Speleers H. THB-splines: The truncated basis for hierarchical splines. *Comput Aided Geom Des* 2012; **29**:485–498.

14. Bazilevs Y, Calo VM, Cottrell JA, *et al.* Isogeometric analysis using T-splines. *Comput Methods Appl Mech Eng* 2010; **199**:229–263.

15. Dokken T, Lyche T, Pettersen KF. Polynomial splines over locally refined box-partitions. *Comput Aided Geom Des* 2013; **30**:331–356.

16. Deng J, Chen F, Li X, *et al.* Polynomial splines over hierarchical T-meshes. *Graph Models* 2008; **70**:76–86.

17. Speleers H, Manni C, Pelosi F, Sampoli ML. Isogeometric analysis with Powell-Sabin splines for advection-diffusion-reaction problems. *Comput Methods Appl Mech Eng* 2012; **221–222**:132–148.

18. Speleers H. A normalized basis for quintic Powell-Sabin splines. *Comput Aided Geom Des* 2010; **27**:438–457.

19. Speleers H, Dierckx P, Vandewalle S. Numerical solution of partial differential equations

with Powell-Sabin splines. *J Comput Appl Math* 2006; **189**:643–659.

20. Speleers H. A normalized basis for reduced Clough-Tocher splines. *Comput Aided Geom Des* 2010; **27**:700–712.

21. Speleers H. Construction of Normalized B-Splines for a Family of Smooth Spline Spaces Over Powell-Sabin Triangulations. *Constr Approx* 2013; **37**:41–72.

22. Grošelj J. A normalized representation of super splines of arbitrary degree on Powell–Sabin triangulations. *BIT Numer Math* 2016; **56**:1257–1280.

23. Jaxon N, Qian X. Isogeometric analysis on triangulations. *CAD Comput Aided Des* 2014; **46**:45–47.

24. Speleers H, Manni C, Pelosi F. From NURBS to NURPS geometries. *Comput Methods Appl Mech Eng* 2013; **255**:238–254.

25. May S, Vignollet J, Borst R de. Powell-Sabin B-splines and unstructured standard T-splines for the solution of the Kirchhoff-Love plate theory exploiting Bézier extraction. *Int J Numer Methods Eng* 2016; **107**:205–233.

26. May S, de Borst R, Vignollet J. Powell-Sabin B-splines for smeared and discrete approaches to fracture in quasi-brittle materials. *Comput Methods Appl Mech Eng* 2016; **307**:193–214.

27. Lai, Ming-Jun; Schumaker LL. *Spline Functions on Triangulations*. Cambridge University Press; 2007.

28. Engvall L, Evans JA. Isogeometric triangular Bernstein-Bézier discretizations: Automatic mesh generation and geometrically exact finite element analysis. *Comput Methods Appl Mech Eng* 2016; **304**:378–407.

29. Engwirda D. Mesh2d-automatic mesh generation. *http://www.mathworks.com/matlabcentral/fileexchange/25555-mesh2d-automatic-mesh-generation* 2009.

30. Farin G. *Curves and Surfaces for CAGD: A Practical Guide*.; 2002.

31. Borden MJ, Scott MA, Evans JA, Hughes TJR. Isogeometric finite element data structures based on Bézier extraction of NURBS. *Int J Numer Methods Eng* 2011; **87**:15–47.

32. Lai MJ, Wenston P. Bivariate splines for fluid flows. *Comput Fluids* 2004; **33**:1047–1073.

33. Lidström P, Olsson P. On the natural vibrations of linear structures with constraints. *J Sound Vib* 2007; **301**:341–354.

34. Rostamiyan Y, Fereidoon A, Davoudabadi M, Yaghoobi H, Ganji DD. Analytical Approach to Investigation of Deflection of Circular Plate Under Uniform Load by Homotopy Perturbation

Method. *Math Comput Appl* 2010; **15**:816–821.

35. Bui TQ, Nguyen MN. A moving Kriging interpolation-based meshfree method for free vibration analysis of Kirchhoff plates. *Comput Struct* 2011; **89**:380–394.

36. Cui XY, Liu GR, Li GY, Zhang GY. A thin plate formulation without rotation DOFs based on the radial point interpolation method and triangular cells. *Int J Numer Methods Eng* 2011; **85**:958–986.

Table 1. Relative error in deflection using different thresholds

Threshold φ	$\leq 3\%$	$\leq 1\%$	$\leq 0.5\%$	$\leq 0.1\%$	Abaqus				exact solution
#nodes	169	481	1273	3004	37	105	237	604	-
#DOF	169	481	1273	3004	185	525	1185	3020	-
center deflection $\times 10^{-3}$	-8.699	-8.690	-8.694	-8.699	-7.996	-8.430	-8.564	-8.655	-8.695
relative error	0.046%	0.058%	0.005%	0.050%	8.039%	3.048%	1.507%	0.460%	-
e_{L^2}	0.803%	0.181%	0.065%	0.012%	7.724%	2.287%	0.910%	0.311%	-

Table 2. Relative error in deflection using different thresholds

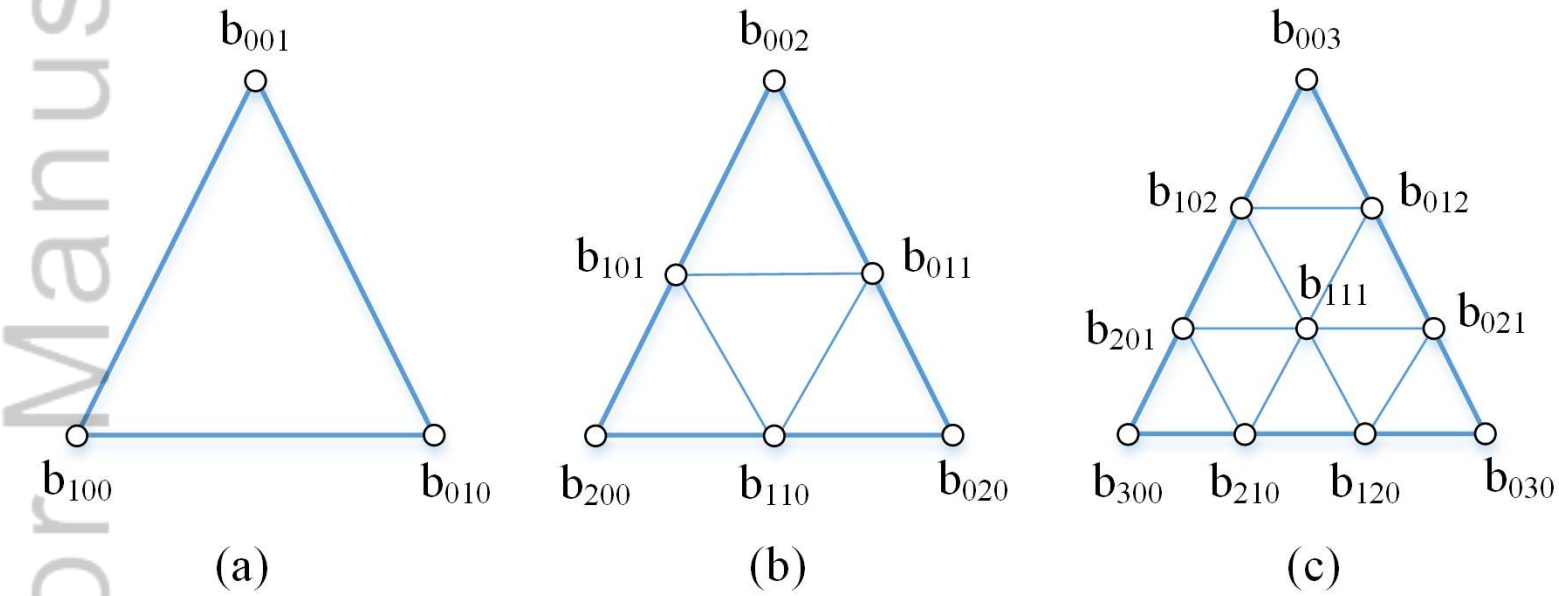
Threshold φ	$\leq 3\%$	$\leq 1\%$	$\leq 0.5\%$	$\leq 0.1\%$	Abaqus
#DOF	2949	5154	9219	17109	162600
max deflection	-0.007367	-0.008273	-0.008608	-0.008772	-0.008950
relative error	17.687%	7.564%	3.821%	1.988%	-

Table 3. Natural frequencies of the square plate with an elliptical hole

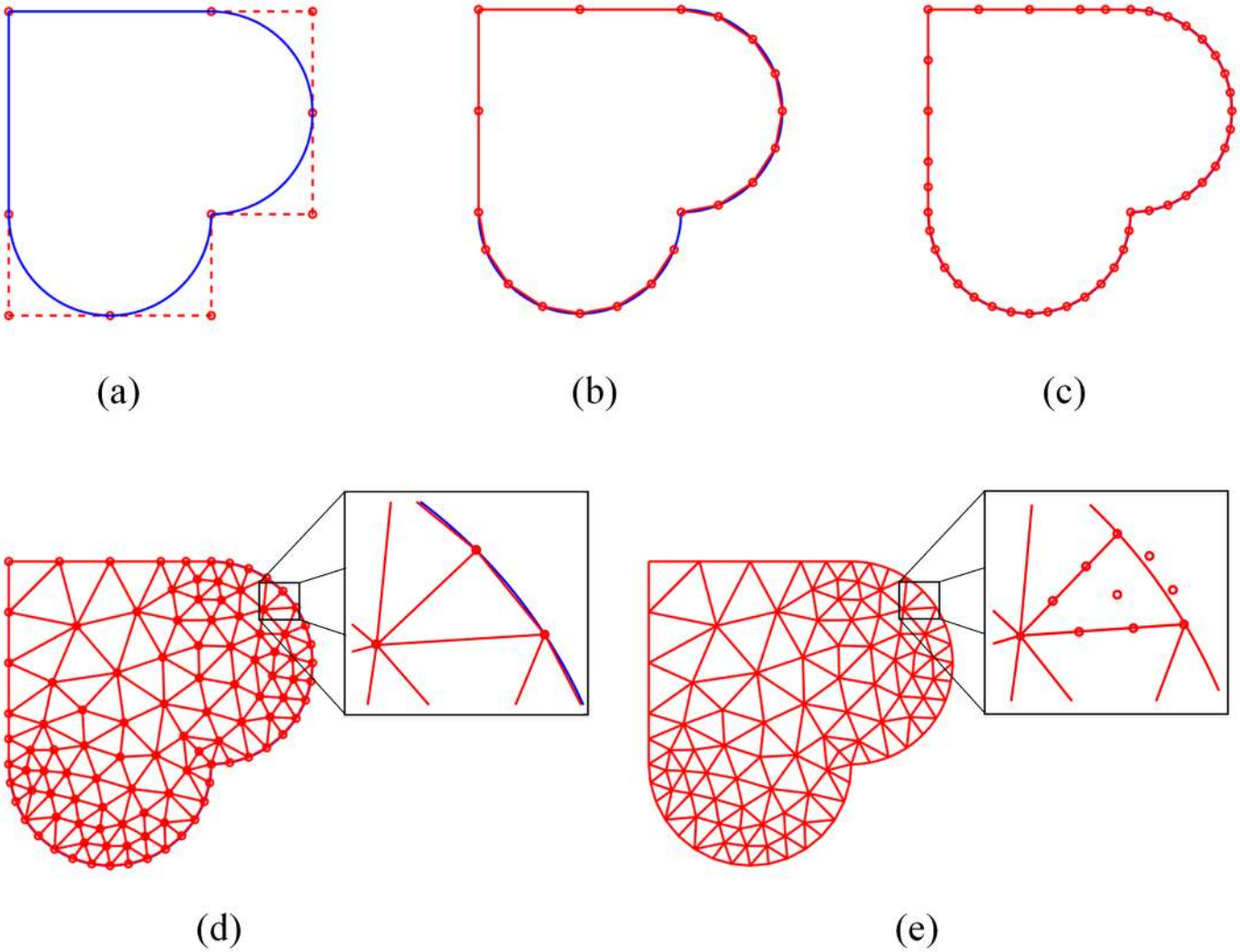
Mode	$a/b=1$				$a/b=2$			$a/b=3$			$a/b=4$		
	Abaqus	$\varphi \le 3\%$	$\varphi \le 1\%$	$\varphi \le 0.5\%$	Abaqus	$\varphi \le 1\%$	$\varphi \le 0.5\%$	Abaqus	$\varphi \le 1\%$	$\varphi \le 0.5\%$	Abaqus	$\varphi \le 1\%$	$\varphi \le 0.5\%$
#nodes	11734	585	1128	2001	11382	1314	2034	10835	1644	2505	10623	1530	2196
1	4.3876	4.5419	4.4809	4.4423	4.3725	4.5058	4.4587	4.3550	4.5108	4.4755	4.3436	4.5455	4.4970
2	6.9581	7.1086	6.9936	6.9669	6.6411	6.6689	6.6613	6.0098	6.0724	6.0484	5.3019	5.3728	5.3734
3	6.9582	7.1720	6.9937	6.9683	6.9127	6.9162	6.9227	6.9298	6.9783	6.9681	6.9975	7.1451	7.0998
4	8.7803	8.8150	8.8040	8.8067	8.6904	8.6915	8.7006	8.4595	8.4901	8.4783	7.9516	8.0177	8.0112
5	9.7965	9.8998	9.8113	9.8027	9.6815	9.6819	9.6840	9.6498	9.6792	9.6723	9.7759	9.9202	9.8772
6	10.0848	10.4552	10.3084	10.2298	10.3845	10.5686	10.5129	10.4431	10.5391	10.5203	10.3834	10.4428	10.4390
7	11.2292	11.3397	11.2882	11.2987	10.8277	10.9352	10.9038	10.4642	10.5433	10.5284	10.4189	10.4877	10.4871
8	11.2293	11.3483	11.2883	11.2994	11.2325	11.2764	11.2692	11.3071	11.3736	11.3502	10.9105	11.0077	11.0073
9	12.7027	12.9518	12.8121	12.8005	12.0485	12.1568	12.1390	11.5094	11.5863	11.5666	11.5470	11.6823	11.6525
10	12.7032	12.9707	12.8124	12.8012	12.9157	13.0627	13.0437	12.8471	13.0027	12.9357	12.2924	12.4050	12.4054

Table 4. Natural frequencies of the square plate with a heart-shape cutout

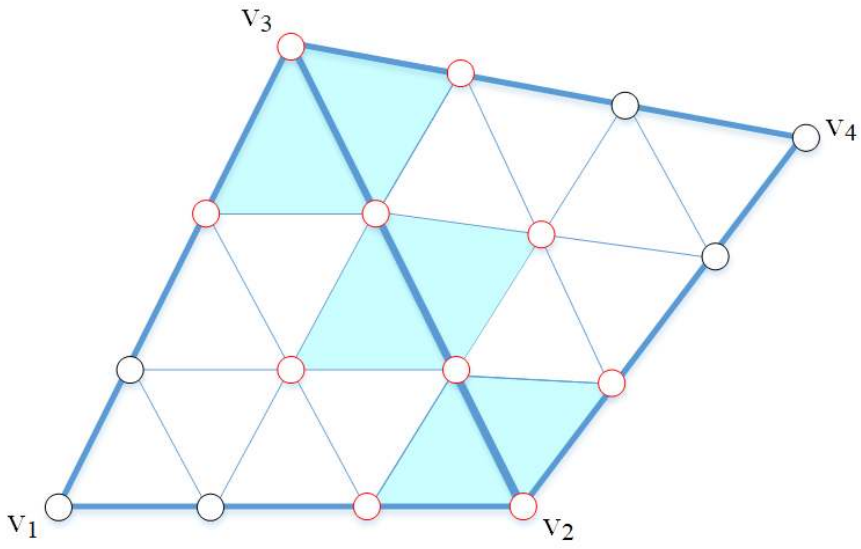
Mode	Threshold φ			Cubic NURBS [2]	Moving Kriging interpolation [35]	Radial point interpolation [36]
	$\leq 3\%$	$\leq 1\%$	$\leq 0.5\%$			
#nodes	384	777	1671	512	506	-
1	5.3688	5.1618	5.0386	5.193	5.3898	4.919
2	6.6041	6.4982	6.4205	6.579	7.5023	6.398
3	7.1068	6.9956	6.8818	6.597	8.3470	6.775
4	8.7629	8.6670	8.6151	7.819	10.6358	8.613
5	9.3238	9.1539	9.0555	8.812	11.0484	9.016
6	10.9440	10.7818	10.7140	9.420	12.8945	10.738
7	11.1755	11.0085	10.9480	10.742	13.7100	10.930
8	11.9959	11.7631	11.6683	10.776	14.0620	11.601
9	13.4042	12.9533	12.8590	11.919	16.6492	12.903
10	13.6026	13.3453	13.2412	13.200	17.3641	13.283



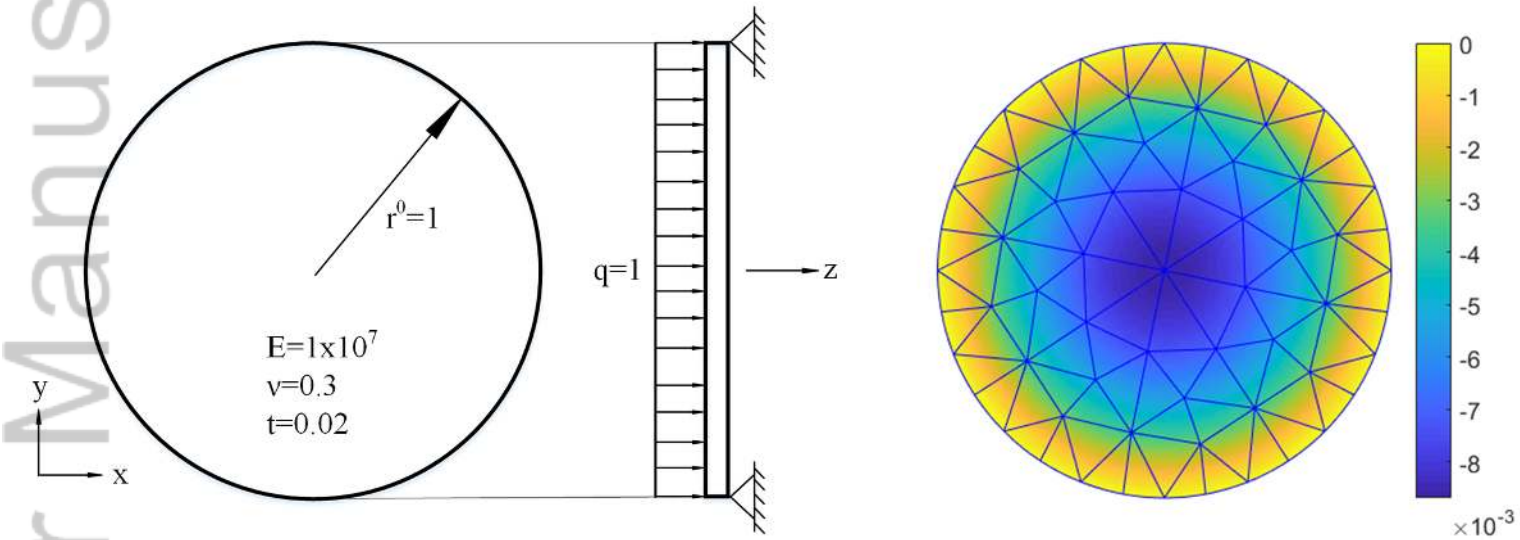
NME_5809_F1.tif



NME_5809_F2.tif

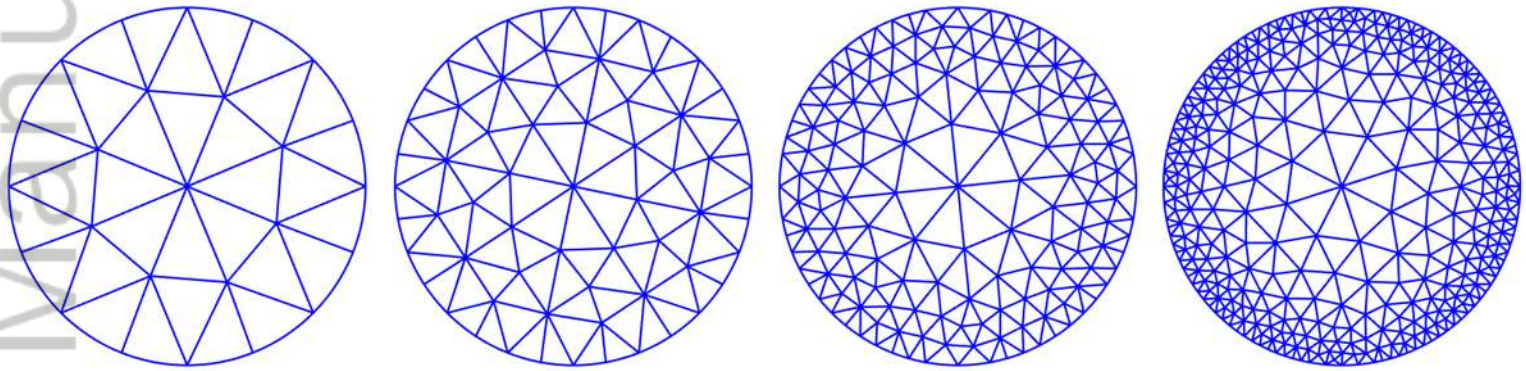


NME_5809_F3.tif



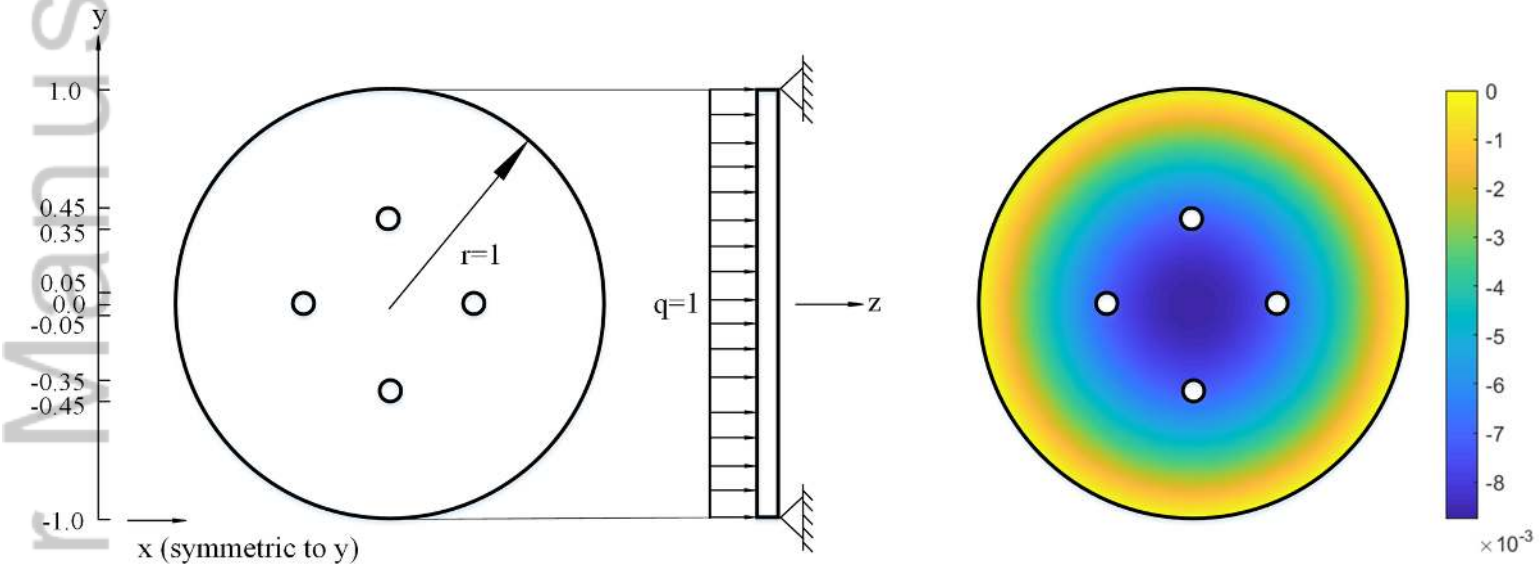
(a) (b)

NME_5809_F4.tif



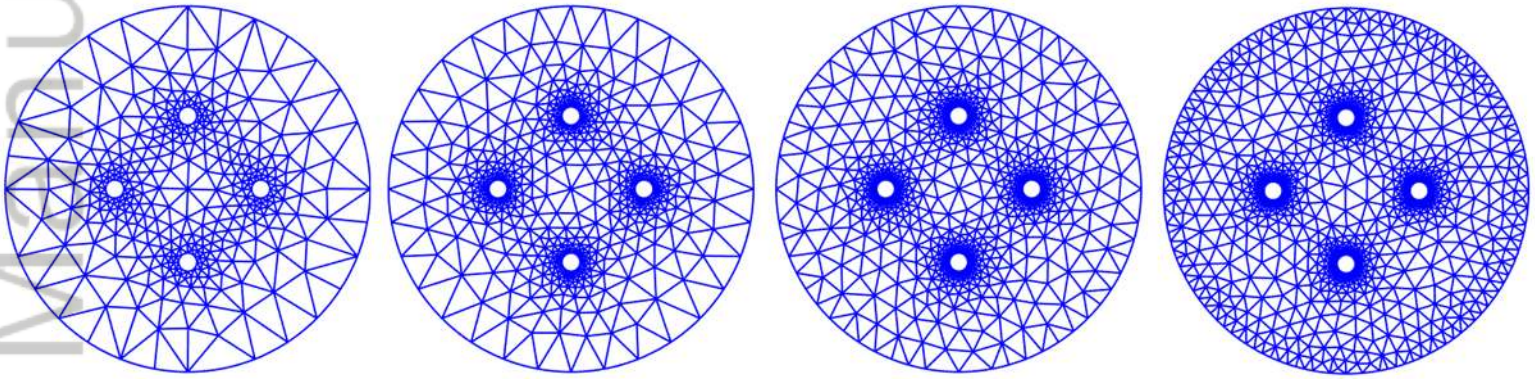
(a) (b) (c) (d)

NME_5809_F5.tif



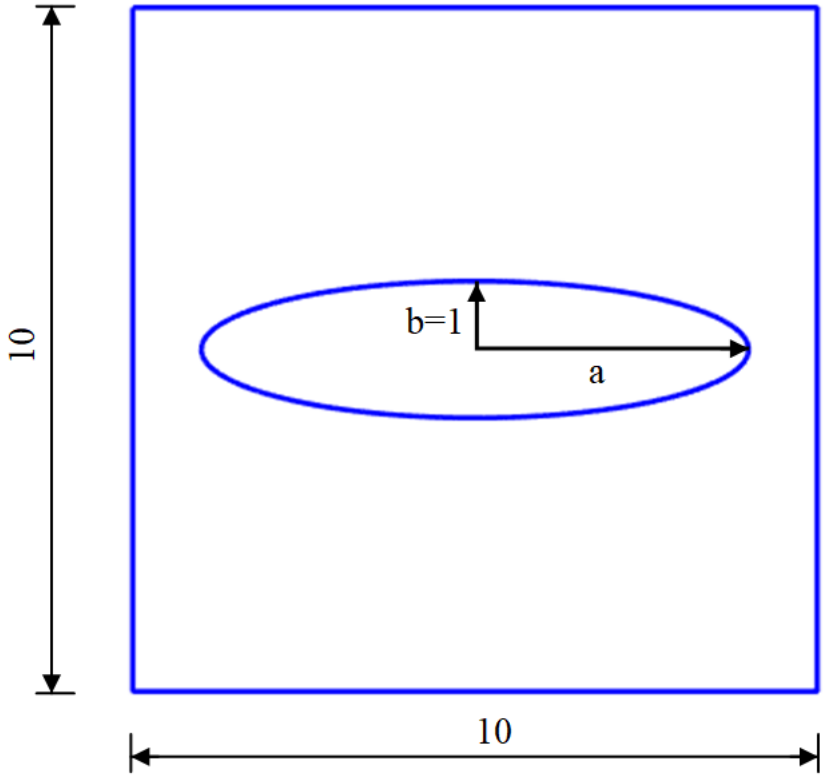
(a) (b)

NME_5809_F6.tif

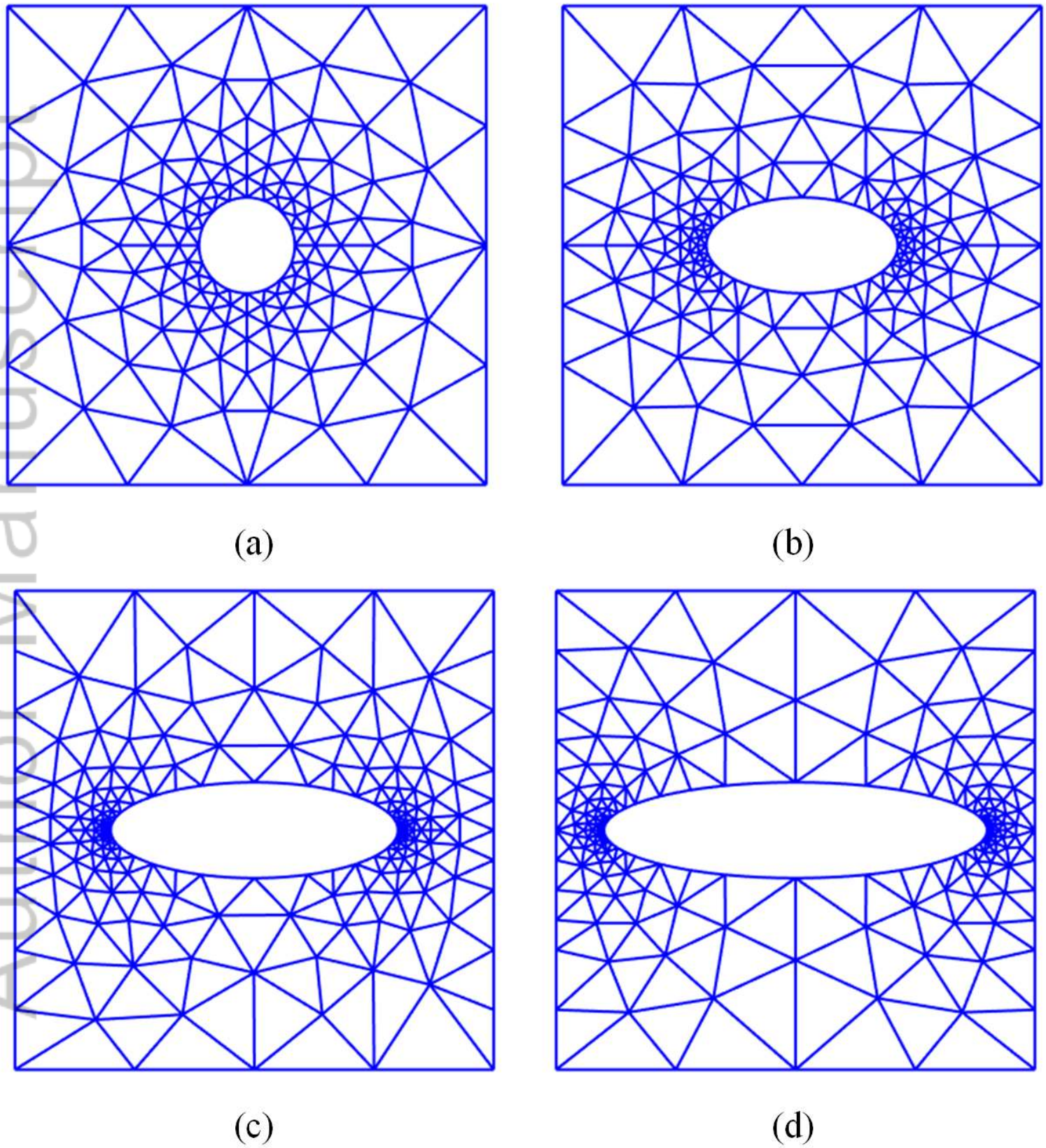


(a) (b) (c) (d)

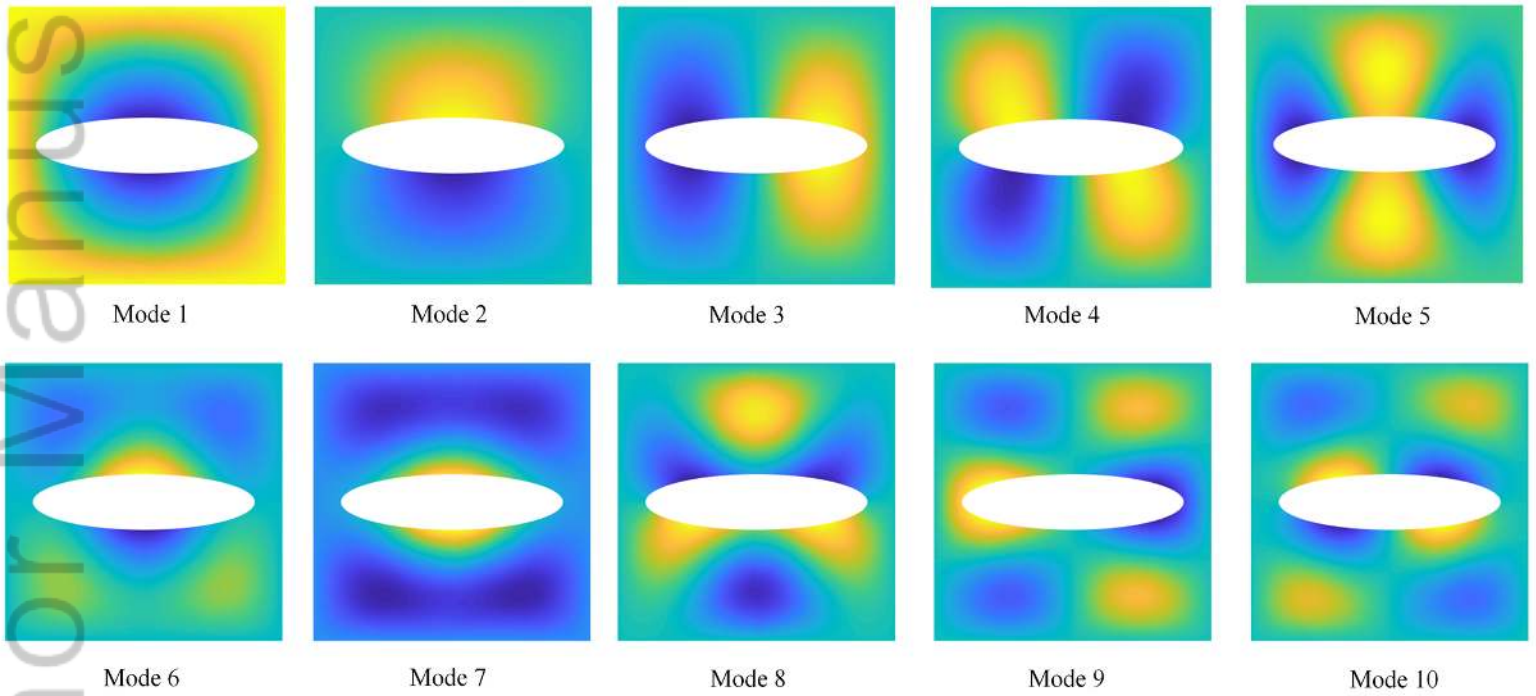
NME_5809_F7.tif



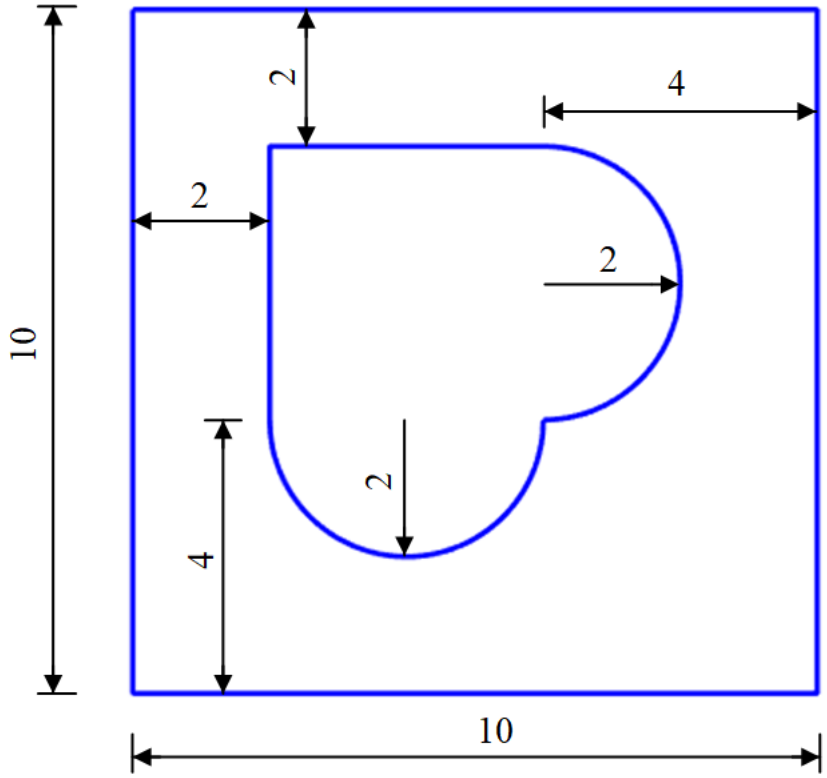
NME_5809_F8.tif



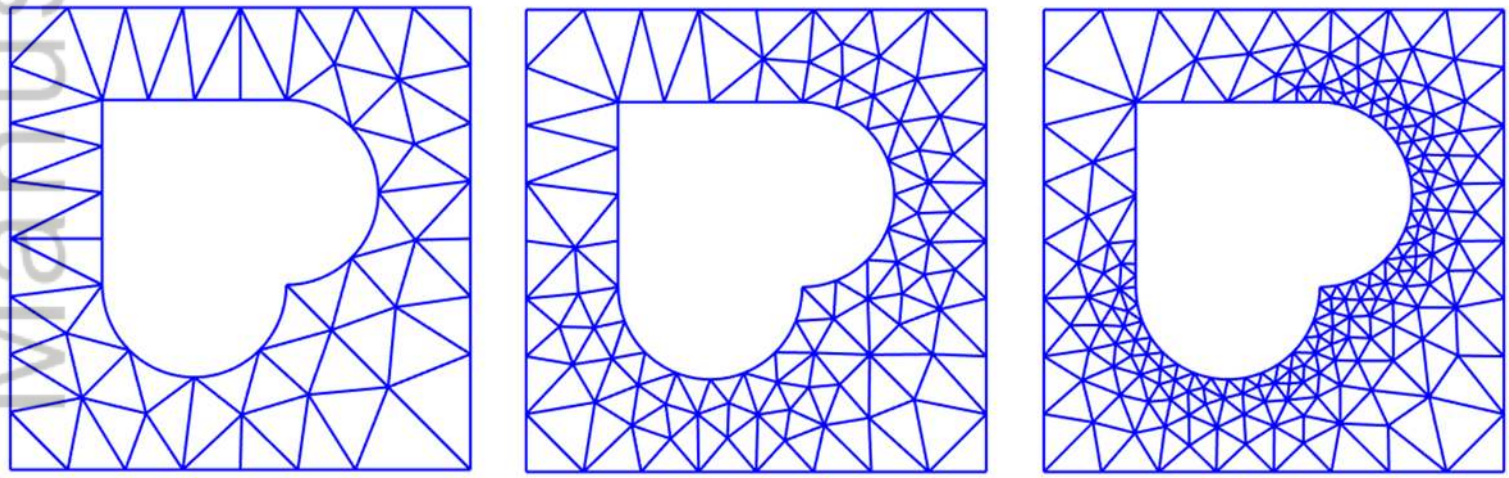
NME_5809_F9.tif



NME_5809_F10.tif

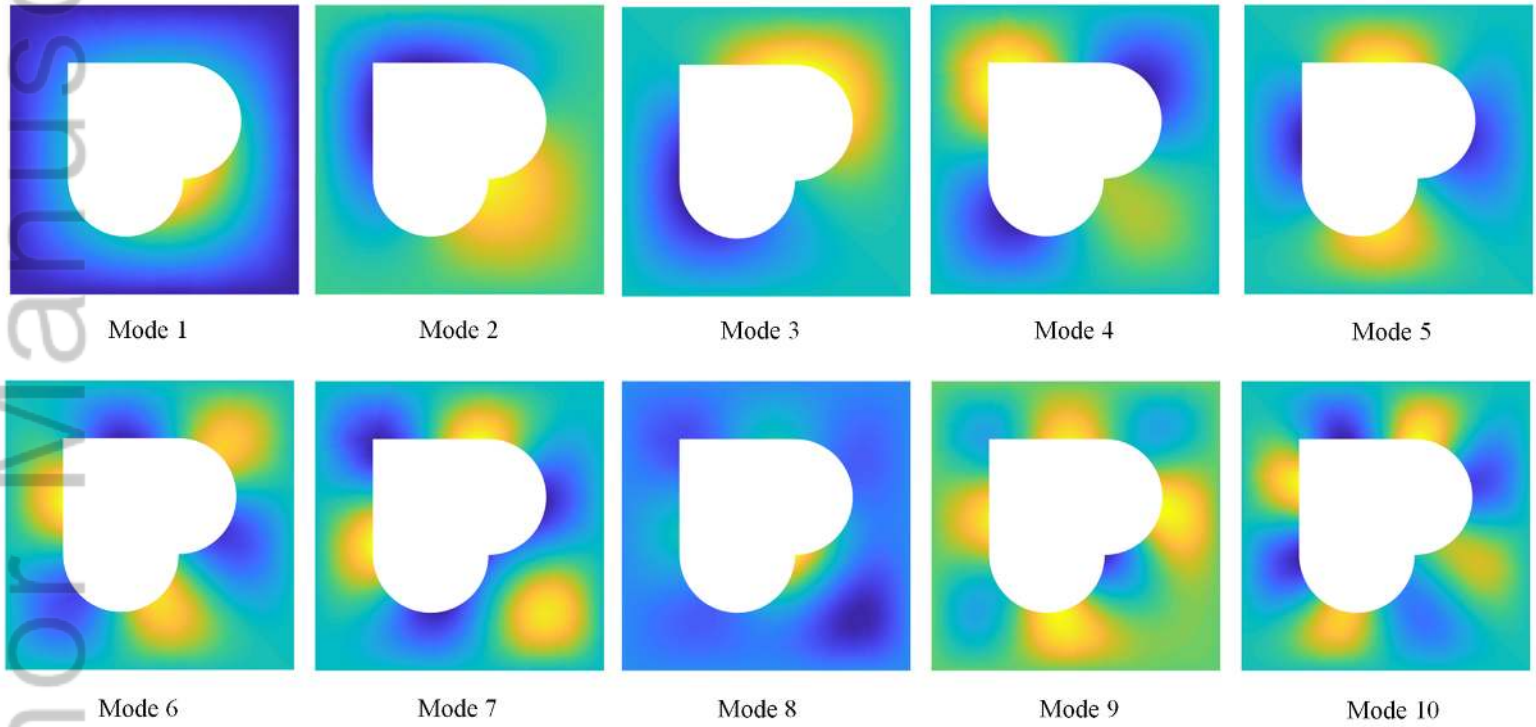


NME_5809_F11.tif



(a) (b) (c)

NME_5809_F12.tif



NME_5809_F13.tif