



ARTICLE

Isogeometric Boundary Element Analysis for 2D Transient Heat Conduction Problem with Radial Integration Method

Leilei Chen¹, Kunpeng Li¹, Xuan Peng², Haojie Lian^{3,4,*}, Xiao Lin⁵ and Zhuojia Fu⁶

¹College of Architecture and Civil Engineering, Xinyang Normal University, Xinyang, 464000, China

²Department of Mechanical Engineering, Suzhou University of Science and Technology, Suzhou, 215009, China

³Key Laboratory of In-Situ Property-Improving Mining of Ministry of Education, Taiyuan University of Technology, Taiyuan, 030024, China

⁴Institute for Computational Engineering, Faculty of Science, Technology and Communication, University of Luxembourg, Luxembourg

⁵The York Management School, University of York, York, YO10 5GD, UK

⁶College of Mechanics and Materials, Hohai University, Nanjing, 211100, China

*Corresponding Author: Haojie Lian. Email: haojie.lian@outlook.com

Received: 16 July 2020 Accepted: 04 September 2020

ABSTRACT

This paper presents an isogeometric boundary element method (IGABEM) for transient heat conduction analysis. The Non-Uniform Rational B-spline (NURBS) basis functions, which are used to construct the geometry of the structures, are employed to discretize the physical unknowns in the boundary integral formulations of the governing equations. Bézier extraction technique is employed to accelerate the evaluation of NURBS basis functions. We adopt a radial integration method to address the additional domain integrals. The numerical examples demonstrate the advantage of IGABEM in dimension reduction and the seamless connection between CAD and numerical analysis.

KEYWORDS

Isogeometric analysis; NURBS; boundary element method; heat conduction; radial integration method

1 Introduction

Isogeometric analysis (IGA) has drawn extensive attentions in engineering science community since the seminal works of Hughes et al. [1]. As an alternative to the traditional analysis based on Lagrange polynomial, IGA uses the same spline functions that are used in the geometric expression in CAD, for example Non-Uniform Rational B-splines (NURBS), as the basis functions to approximate the unknown field. Hence, IGA is able to perform numerical analysis directly from CAD without cumbersome preprocessing procedure and geometric errors. Additionally, IGA offers high-order continuity and flexible refinement schemes which are particularly attractive in numerical simulation. Due to the aforementioned salient features, IGA has been successfully applied in



many fields [2–6]. To facilitate the implementation of IGA, Bézier extraction [7] technique is proposed that enables one to develop IGA codes in existing finite element codes, without changing the program framework. By using the extraction operator, the complicated iterative process for evaluating B-spline basis functions is eliminated, which improves the efficiency of simulation significantly.

Although the concept of IGA was originally proposed in the context of finite element methods, its application in engineering practice is restricted because FEM relies on volume parameterization which conflicts with the boundary representation in CAD field. Mierzwiczak et al. [8] and Fu et al. [9] applied the isogeometric boundary element method to the heat conduction problem, and proposed the origin intensity factor to eliminate the singularity of the integral. The order reduction of POD model for transient heat transfer was studied by Li et al. [10]. Wang et al. [11] solved the three-dimensional elasto-plastic problem by boundary element method. On the contrary, boundary element method (BEM) [12–19] only needs boundary parameterization and thus is compatible with CAD models naturally. The isogeometric analysis with boundary element method (IGABEM) was first applied to potential problems [20] and elasticity analysis [21,22]. Since then, IGABEM has been applied successfully to a wide range of fields, such as heat conduction [23], linear elasticity [24,25], fracture mechanics [26–28], electromagnetics [29], acoustics [30–35], structural optimization [34,36–40], etc.

In this paper, we are aimed to extend the application of IGABEM to heat conduction problems which are an increasingly important topic in many engineering fields. The radial integration method proposed by Gao [41] is incorporated to solve the additional domain integrals. The problems of constant and variable coefficients, transient and steady state of temperature field are studied. The rest of the paper is arranged as follows: Section 2 gives an overview of the B-spline and NURBS basis. Bézier extraction is detailed in Section 3. In Section 4, the boundary integral equation for heat conduction is introduced and the discrete formula derivation of the boundary integral equation, as well as the time marching method are introduced in detail. Section 5 provides two sets of numerical examples to verify the algorithm, followed by the conclusion in Section 6.

2 NURBS

For the sake of completeness, the fundamentals of NURBS are briefed in this section. NURBS are generalized from B-splines, whose basis functions are defined over a knot vector which is a monotonic increasing real number sequence, denoted by $U = \{u_1, u_2, \dots, u_m\}$ where u_i is the i -th knot in the vector U . B-spline basis function is evaluated iteratively as:

$$N_{i,0}(u) = \begin{cases} 1, & \text{if } u_i \leq u < u_{i+1} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$N_{i,p} = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u), \quad (2)$$

where p is the order of the polynomial in B-spline basis functions, and n is the number of the basis functions.

The derivative expression of the basis function can be derived as follows:

$$N'_{i,p} = \frac{p}{u_{i+p} - u_i} N_{i,p-1} - \frac{p}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \quad (3)$$

The B-spline curve is described by the linear combination of NURBS basis functions and the corresponding control points,

$$\mathbf{x}(u) = \sum_{i=1}^n N_{i,p}(u) \mathbf{P}_i, \quad (4)$$

where \mathbf{P}_i is the Cartesian coordinates of control points.

Non-uniform rational B-spline (NURBS) basis functions are obtained by introducing the weight coefficient ω to rationalize the B-spline basis,

$$R_{i,p}(u) = \frac{N_{i,p}(u) \omega_i}{W(u)}, \quad (5)$$

where

$$W(u) = \sum_{j=1}^n N_{j,p} \omega_j. \quad (6)$$

Similar to B-splines, the NURBS curve is expressed by:

$$\mathbf{x}(u) = \sum_{i=1}^n R_{j,p}(u) \mathbf{P}_i \quad (7)$$

Since the NURBS basis function adopts the rational form composed of the B-spline basis function and the weight factor ω , it is able to exactly express conic curves, such as ellipses and circles.

3 Bézier Extraction Operation

The computation of IGABEM can be accelerated using Bézier extraction. The idea is to extract linear operators and map the Bernstein polynomial basis functions of Bézier elements to global B-spline basis functions. The linear transformation is defined by a matrix called an extraction operator. The transpose of the extraction operator maps the control points of the global B-spline to that of Bézier curves. The process of Bézier extraction operation is proposed in this work, also see [7].

To perform Bézier decomposition, all internal knots of a knot vector are repeated until their multiplicity equals p . For example, we insert \bar{u} after the k -th knot of the original knot vector $\Xi = \{u_1, u_2, \dots, u_{n+p+1}\}$. The new knot $\xi = \{u_1, u_2, \dots, u_k, \bar{u}, u_{k+1}, u_{n+p+1}\}$ with $k > p$. The introduction of new knots will bring about changes in the basis functions and the generation of new control points. The relationship between the original control point \mathbf{P} and the new control point \mathbf{Q} can be deduced:

$$\mathbf{Q}_i = \begin{cases} \mathbf{P}_1, & i = 1 \\ \alpha_i \mathbf{P}_i + (1 - \alpha_i) \mathbf{P}_{i-1}, & 0 < i < m \\ \mathbf{P}_n, & i = m \end{cases} \quad (8)$$

where $m = n + 1$. The expression for calculating factor α_i in Eq. (8) is as follows:

$$\alpha_i = \begin{cases} 1, & i \leq k - p; \\ \frac{\bar{u} - u_i}{u_{i+p} - u_i}, & k - p + 1 \leq i \leq k \\ 0, & i \geq k + 1 \end{cases} \quad (9)$$

After the knot is inserted, it brings about a change in the control point. Let $\{\bar{u}_1, \bar{u}_2, \dots, \bar{u}_m\}$ be the set of insertion knots required for the Bézier decomposition of the B-spline. Then for each new knot, $\bar{u}_j, j = 1, 2, \dots, m$, using Eq. (9), we define $\alpha_i^j, i = 1, 2, \dots, n + j$, to be the i th α . So the expression for the conversion matrix \mathbf{C} is as follows:

$$\mathbf{C}_j = \begin{bmatrix} \alpha_1 & 1 - \alpha_2 & 0 & \dots & & 0 \\ 0 & \alpha_2 & 1 - \alpha_3 & 0 & \dots & 0 \\ 0 & 0 & \alpha_3 & 1 - \alpha_4 & 0 & \dots & 0 \\ \vdots & & & & & & \\ 0 & \dots & & & 0 & \alpha_{n+j-1} & 1 - \alpha_{n+j} \end{bmatrix} \quad (10)$$

Then, the corresponding control points can be calculated by:

$$\mathbf{Q} = \mathbf{C}_m^T \dots \mathbf{C}_1^T \mathbf{P} \quad (11)$$

We rewrite Eq. (11) with matrix form:

$$\mathbf{Q} = \mathbf{C}^T \mathbf{P} \quad (12)$$

To decompose a B-spline curve defined in the knot vector $\{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$ with Bézier extraction, we will insert the knots $\{0.25, 0.25, 0.5, 0.5, 0.75, 0.75\}$ into the existing knot vector. Without changing the geometry, the B-spline curve is expressed using the Bézier basis function as follows:

$$\mathbf{X}(u) = \sum_{i=1}^n N_{i,p} \mathbf{P}_i = \sum_{i=1}^m \mathbf{B}_{i,n} \mathbf{Q}_i \quad (13)$$

Using Eqs. (12), (13) can be expressed as follows:

$$\mathbf{X}(u) = \mathbf{Q}^T \mathbf{B}(u) = \left(\mathbf{C}^T \mathbf{P} \right)^T \mathbf{B}(u) = \mathbf{P}^T \mathbf{C} \mathbf{B}(u) = \mathbf{P}^T \mathbf{N}(u) \quad (14)$$

Since \mathbf{P} is arbitrary, we can get the following equation

$$\mathbf{N}(u) = \mathbf{C} \mathbf{B}(u) \quad (15)$$

It should be noted that only the knot vector determines the C matrix. In other words, the extraction operator is a product of parameterization and does not depend on control points

or basis functions. Therefore, we can apply the extraction operator directly to NURBS. Define weights as diagonal matrices

$$\mathbf{W} = \begin{bmatrix} \omega_1 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \omega_n \end{bmatrix} \quad (16)$$

Converting Eq. (5) to matrix form, we have

$$\mathbf{R}(u) = \frac{1}{W(u)} \mathbf{W} \mathbf{N}(u) \quad (17)$$

Using Eq. (15), the NURBS basis function can be expressed in terms of the Bernstein basis as

$$\mathbf{R} = \frac{1}{\omega^T \mathbf{N}} \mathbf{W} \mathbf{N} = \frac{1}{\omega^T \mathbf{C} \mathbf{B}} \mathbf{W} \mathbf{C} \mathbf{B} \quad (18)$$

The NURBS curve is expressed as:

$$\mathbf{T}(\mathbf{u}) = \frac{1}{\omega^T \mathbf{C} \mathbf{B}(\mathbf{u})} \mathbf{P}^T \mathbf{W} \mathbf{C} \mathbf{B}(\mathbf{u}) \quad (19)$$

In Eq. (19), $\mathbf{B}(u)$ can be used to express the NURBS curve. Let $\omega^b = \mathbf{C}^T \omega$, and define \mathbf{W}^b to be the diagonal matrix

$$\mathbf{W}^b = \begin{bmatrix} w_1^b & & & \\ & w_2^b & & \\ & & \ddots & \\ & & & w_{n+m}^b \end{bmatrix} \quad (20)$$

Eq. (19) can be written as follows:

$$\mathbf{T}(\mathbf{u}) = \frac{1}{(\omega^b)^T \mathbf{B}(\mathbf{u})} \mathbf{P}^T \mathbf{W} \mathbf{C} \mathbf{B}(\mathbf{u}) \quad (21)$$

The expression of the Bézier control points can refer to the method of homogeneous coordinates:

$$\mathbf{P}^b = (\mathbf{W}^b)^{-1} \mathbf{C}^T \mathbf{W} \mathbf{P} \quad (22)$$

Both sides are multiplied by \mathbf{W}^b at the same time:

$$\mathbf{W}^b \mathbf{P}^b = \mathbf{C}^T \mathbf{W} \mathbf{P} \quad (23)$$

After solving the weight factor ω^b and the control point P^b after the Bézier operation, we arrive at the final Bézier representation of the NURBS:

$$\mathbf{T}(\mathbf{u}) = \frac{\mathbf{1}}{\mathbf{W}^b(\mathbf{u})} \left(\mathbf{W}^b \mathbf{P}^b \right)^T \mathbf{B}(\mathbf{u}) = \sum_{i=1}^{n+m} \frac{\mathbf{P}_i^b \omega_i^b \mathbf{B}_i(\mathbf{u})}{\mathbf{W}^b(\mathbf{u})} \quad (24)$$

Thus, we can express NURBS equivalently with a series of C^0 Bézier elements.

4 Governing Equation

The governing partial differential equation of transient heat conduction in isotropic medium with internal heat source is expressed by:

$$k \nabla^2 T(\mathbf{x}, t) + b(\mathbf{x}, t) = \rho c_x \frac{\partial T(\mathbf{x}, t)}{\partial t} \quad (t > t_0, \mathbf{x} \in \Omega) \quad (25)$$

where $T(\mathbf{x}, t)$ represents the temperature at point \mathbf{x} at time t , k is the thermal conductivity, $b(\mathbf{x}, t)$ shows the heat source value at point \mathbf{x} at time t , ρ is the density of the material, and c_x is specific heat capacity.

The following three kinds of boundary conditions are considered:

$$\begin{aligned} T(\mathbf{x}, t) &= \bar{T}(\mathbf{x}, t) \quad (t > t_0, \mathbf{x} \in \Gamma_T) \\ q(\mathbf{x}, t) &= -k \frac{\partial T(\mathbf{x}, t)}{\partial \mathbf{n}(\mathbf{x})} = \bar{q}(\mathbf{x}, t) \quad (t > t_0, \mathbf{x} \in \Gamma_q) \\ T_f(\mathbf{x}, t) &= \bar{T}_f(\mathbf{x}, t) \quad (t > t_0, \mathbf{x} \in \Gamma_f) \end{aligned} \quad (26)$$

where γ_T represents the dirichlet boundary, γ_q the neumann boundary and γ_f convective boundary condition. $\mathbf{n}(\mathbf{x})$ is the unit external normal direction vector at the boundary point.

The transient problem is time-dependent, so the initial conditions need to be prescribed:

$$T(\mathbf{x}, t_0) = T_0(\mathbf{x}) \quad (\mathbf{x} \in \Omega)$$

where T_0 indicates initial temperature on the domain.

4.1 Boundary Element Method of Transient Temperature Field

Through weighted residual operation and partial integration operation, Eq. (25) can be expressed as the following boundary-domain integral equation:

$$c(\mathbf{x}) \hat{T}(\mathbf{x}, t) + \left(H^* \hat{T} \right) (\mathbf{x}) = \left(G^* q \right) (\mathbf{x}) + \left(\tilde{G} b \right) (\mathbf{x}) - \left(\tilde{G} \dot{T} \right) (\mathbf{x}) \quad (27)$$

where the coefficient $c(\mathbf{x}) = 1$ when the source point \mathbf{x} is located in the domain, $c(\mathbf{x}) = 0.5$ when the source point \mathbf{x} is on the boundary of structure, and the integrals are

$$\begin{cases} \left(H^* \hat{T} \right) (\mathbf{x}) := \int_{\Gamma} H(\mathbf{x}, \mathbf{y}) \hat{T}(\mathbf{y}, t) d\Gamma(\mathbf{y}) \\ \left(G^* q \right) (\mathbf{x}) := \int_{\Gamma} G(\mathbf{x}, \mathbf{y}) q(\mathbf{y}, t) d\Gamma(\mathbf{y}) \\ \left(\tilde{G} b \right) (\mathbf{x}) := \int_{\Omega} G(\mathbf{x}, \tilde{\mathbf{y}}) b(\tilde{\mathbf{y}}, t) d\Omega(\tilde{\mathbf{y}}) \\ \left(\tilde{G} \dot{T} \right) (\mathbf{x}) := \int_{\Omega} G(\mathbf{x}, \tilde{\mathbf{y}}) \tilde{\rho} \dot{\hat{T}}(\tilde{\mathbf{y}}, t) d\Omega(\tilde{\mathbf{y}}) \end{cases} \quad (28)$$

where $\hat{T} = kT$ indicates normalized temperature, $\dot{\hat{T}} = \frac{\partial \hat{T}}{\partial t}$, $\tilde{\rho} = \rho c_x/k$ is reciprocal of thermal diffusivity, $G(\mathbf{x}, \mathbf{y})$ and $H(\mathbf{x}, \mathbf{y})$ are Green's function and it's derivative, as follows:

$$\begin{cases} G(\mathbf{x}, \mathbf{y}) = \frac{1}{2\pi} \ln\left(\frac{1}{r}\right) \\ H(\mathbf{x}, \mathbf{y}) = \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} = \frac{-1}{2\pi r} \frac{\partial r}{\partial \mathbf{n}(\mathbf{y})} \end{cases} \quad (29)$$

In this work, we do not consider the source. Thus, Eq. (27) can be rewritten as

$$c(\mathbf{x}) \hat{T}(\mathbf{x}, t) + (H^* \hat{T})(\mathbf{x}) = (G^* q)(\mathbf{x}) - (\tilde{G} \dot{\hat{T}})(\mathbf{x}) \quad (30)$$

By observing Eq. (28), we can find that the last term on the right hand of Eq. (30) is domain integral term rather than boundary integral term. In this work, the radial integral method is used to transform domain integral into boundary integral to retain the advantage of dimension reduction of IGABEM.

We firstly approximate unknown temperature gradient $\dot{\hat{T}}$ by interpolation calculation. The commonly used interpolation function is radial basis function (RBF). We select a number of collocation points and interpolation to get the approximate formula of variables $\dot{\hat{T}}$,

$$\dot{\hat{T}}(\tilde{\mathbf{y}}, t) = \sum_{i=1}^N \beta_i(t) f_i(r(\tilde{\mathbf{y}})) \quad (31)$$

where N is the number of collocation points, β_i is the undetermined coefficient at the i -th collocation point, $f_i(r)$ is the radial basis function at the i -th collocation point, $r = |\tilde{\mathbf{y}} - \mathbf{x}^i|$, and \mathbf{x}^i is the coordinates of the i -th collocation point.

It is worth noting that collocation points can be boundary points or internal points. Generally, combining boundary points and internal points to form collocation points can effectively improve the approximation accuracy. In addition, the accuracy of approximation can be further improved by combining radial basis functions and polynomials, which can be expressed in combination with quadratic polynomials,

$$\dot{\hat{T}}(\tilde{\mathbf{y}}, t) = \sum_{i=1}^N \beta_i(t) f_i(r(\tilde{\mathbf{y}})) + \gamma_0(t) + \sum_{i=1}^2 \gamma_i(t) \tilde{\mathbf{y}}_i + \sum_{i=1}^2 \sum_{j=i}^2 \gamma_{ij}(t) \tilde{\mathbf{y}}_i \tilde{\mathbf{y}}_j \quad (32)$$

where $\tilde{\mathbf{y}}_i$ and $\tilde{\mathbf{y}}_j$ are Cartesian coordinate components of the point $\tilde{\mathbf{y}}$. The following conditions are met:

$$\begin{cases} \sum_{i=1}^N \beta_i = 0 \\ \sum_{i=1}^N \beta_i \mathbf{x}_1^i = \sum_{i=1}^N \beta_i \mathbf{x}_2^i = 0 \\ \sum_{i=1}^N \beta_i \mathbf{x}_1^i \mathbf{x}_1^i = \sum_{i=1}^N \beta_i \mathbf{x}_2^i \mathbf{x}_2^i = \sum_{i=1}^N \beta_i \mathbf{x}_1^i \mathbf{x}_2^i = 0 \end{cases} \quad (33)$$

It is particularly important to determine these coefficients β and γ in Eq. (32). Although it is not easy to directly express these coefficients, we can get the implicit expression of these coefficients through a series of transformation operations. By setting collocation point as $\tilde{\mathbf{y}}$ in Eq. (32) and assembling them into matrix-vector formulation, we can get the following expression:

$$\mathbf{f}\mathbf{v} = \hat{\mathbf{T}} \quad (34)$$

where

$$\mathbf{f} = \begin{bmatrix} f^{11} & f^{12} & \dots & f^{1N} & 1 & \mathbf{x}_1^1 & \mathbf{x}_2^1 & \mathbf{x}_1^1 \mathbf{x}_2^1 & \mathbf{x}_1^1 \mathbf{x}_1^1 & \mathbf{x}_2^1 \mathbf{x}_2^1 \\ f^{21} & f^{22} & \dots & f^{2N} & 1 & \mathbf{x}_1^2 & \mathbf{x}_2^2 & \mathbf{x}_1^2 \mathbf{x}_2^2 & \mathbf{x}_1^2 \mathbf{x}_1^2 & \mathbf{x}_2^2 \mathbf{x}_2^2 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f^{N1} & f^{N2} & \dots & f^{NN} & 1 & \mathbf{x}_1^N & \mathbf{x}_2^N & \mathbf{x}_1^N \mathbf{x}_2^N & \mathbf{x}_1^N \mathbf{x}_1^N & \mathbf{x}_2^N \mathbf{x}_2^N \\ 1 & 1 & \dots & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{x}_1^1 & \mathbf{x}_1^2 & \dots & \mathbf{x}_1^N & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{x}_2^1 & \mathbf{x}_2^2 & \dots & \mathbf{x}_2^N & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{x}_1^1 \mathbf{x}_2^1 & \mathbf{x}_1^2 \mathbf{x}_2^2 & \dots & \mathbf{x}_1^N \mathbf{x}_2^N & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{x}_1^1 \mathbf{x}_1^1 & \mathbf{x}_1^2 \mathbf{x}_1^2 & \dots & \mathbf{x}_1^N \mathbf{x}_1^N & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{x}_2^1 \mathbf{x}_2^1 & \mathbf{x}_2^2 \mathbf{x}_2^2 & \dots & \mathbf{x}_2^N \mathbf{x}_2^N & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (35)$$

and

$$\mathbf{v} = [\beta_1 \quad \beta_2 \quad \dots \quad \beta_N \quad \gamma_0 \quad \gamma_1 \quad \gamma_2 \quad \gamma_{11} \quad \gamma_{12} \quad \gamma_{22}]^T \quad (36)$$

$$\hat{\mathbf{T}} = [\hat{T}_1 \quad \hat{T}_2 \quad \dots \quad \hat{T}_N \quad 0 \quad \dots \quad 0]^T \quad (37)$$

where $f^{ij} = f_j(r)$ in Eq. (35), $r = |\mathbf{x}^i - \mathbf{x}^j|$, and \hat{T}_i denotes the temperature derivative at the i -th collocation point.

When the collocation point is not repeated, the matrix \mathbf{f} is invertible. Thus, the coefficient vector \mathbf{v} can be expressed as:

$$\mathbf{v} = \mathbf{f}^{-1} \dot{\hat{\mathbf{T}}} \quad (38)$$

By substituting Eq. (33) into the last term on the right of Eq. (27) and using the radial integration method, the following boundary integral formula can be obtained

$$(\tilde{G}\dot{\hat{T}})(\mathbf{x}) := I_0 + I_1 + I_2 \quad (39)$$

where

$$\begin{cases} I_0 = \sum_{i=1}^N \beta_i \int_{\Gamma} \frac{B_i(\mathbf{x}, \mathbf{y})}{r(\mathbf{x}, \mathbf{y})} \frac{\partial r}{\partial \mathbf{n}(\mathbf{y})} d\Gamma(\mathbf{y}) \\ I_1 = \sum_{i=1}^2 \gamma_i \int_{\Gamma} \frac{r_{,k} \tilde{B}^1(\mathbf{x}, \mathbf{y})}{r(\mathbf{x}, \mathbf{y})} \frac{\partial r}{\partial \mathbf{n}(\mathbf{y})} d\Gamma(\mathbf{y}) \\ I_2 = \left(\gamma_0 + \sum_{i=1}^2 \gamma_i \mathbf{x}_i \right) \int_{\Gamma} \frac{B_0(\mathbf{x}, \mathbf{y})}{r(\mathbf{x}, \mathbf{y})} \frac{\partial r}{\partial \mathbf{n}(\mathbf{y})} d\Gamma(\mathbf{y}) \end{cases} \quad (40)$$

and

$$\begin{cases} B_i(\mathbf{x}, \mathbf{y}) = \int_0^{r(\mathbf{x}, \mathbf{y})} G(\mathbf{x}, \hat{\mathbf{y}}) \tilde{\rho} f_i(R) r(\mathbf{x}, \hat{\mathbf{y}}) dr(\hat{\mathbf{y}}) \\ \tilde{B}^1(\mathbf{x}, \mathbf{y}) = \int_0^{r(\mathbf{x}, \mathbf{y})} G(\mathbf{x}, \hat{\mathbf{y}}) \tilde{\rho} r^2(\mathbf{x}, \hat{\mathbf{y}}) dr(\hat{\mathbf{y}}) \\ B_0(\mathbf{x}, \mathbf{y}) = \int_0^{r(\mathbf{x}, \mathbf{y})} G(\mathbf{x}, \hat{\mathbf{y}}) \tilde{\rho} r(\mathbf{x}, \hat{\mathbf{y}}) dr(\hat{\mathbf{y}}) \end{cases} \quad (41)$$

It can be found that the domain integral in Eq. (39) is successfully transformed into the boundary integral. For the detailed derivation process of the above results, also see [42].

It is worth noting that these coefficients in Eq. (40) are unknown, so it is impossible to obtain the results of I_0 , I_1 , and I_2 directly through the boundary integral calculation. In fact, these coefficients do not need to be calculated directly.

When the source point \mathbf{x} is set as the i -th collocation point, the terms I_0 , I_1 , and I_2 can be rewritten as a formulas of vector and vector multiplication, respectively. By combining the three terms, we can get the following formulas

$$(\tilde{G}\dot{\hat{T}})(\mathbf{x}) := \mathbf{A}^i \mathbf{v} \quad (42)$$

where the vector \mathbf{A}^i contains the result of the boundary integrals in Eq. (40). By substituting Eq. (38) into Eq. (42), we can obtain the following formulation

$$(\tilde{G}\dot{\hat{T}})(\mathbf{x}) := \mathbf{A}^i \mathbf{f}^{-1} \dot{\hat{\mathbf{T}}} = \mathbf{C}^i \dot{\hat{\mathbf{T}}} \quad (43)$$

where $\mathbf{C}^i = \mathbf{A}^i \mathbf{f}^{-1}$. When the point \mathbf{x} in Eq. (43) is chosen as collocation point, by assembling the equations at all points, we could get a formulation of matrix vector multiplication.

4.2 Discretization of Boundary Integral Equation

First, we discretize the boundary into N_e non-overlapping NURBS elements,

$$\Gamma = \bigcup_{e=1}^{N_e} \Gamma_e, \quad \Gamma_i \cap \Gamma_j = \mathbf{0}, \quad i \neq j \quad (44)$$

NURBS basis functions are used for approximation. In any NURBS element e , variables can be approximately expressed as follows:

$$\begin{cases} \hat{T}(\xi) = \sum_{l=1}^m R_{l,p}(\xi) \hat{T}^l \\ q(\xi) = \sum_{l=1}^m R_{l,p}(\xi) q^l \end{cases} \quad (45)$$

where m is the number of control points, $\xi \in [-1, 1]$ represents local parametric coordinates, \hat{T}^l and q^l are normalized temperature and its flux at the l -th collocation points. In this work, the Greville abscissa is used to define the positions of collocation points in parameter space, also see [39].

By substituting Eq. (45) into Eq. (27), the two boundary integral terms can be rewritten as

$$\begin{cases} (H^* \hat{T})(\mathbf{x}) := \sum_{e=1}^{N_e} \sum_{l=1}^m \left[\int_{-1}^1 H(\mathbf{x}, \mathbf{y}(\xi)) R_{l,p}(\xi) J^e(\xi) d\xi \right] \hat{T}^l \\ (G^* q)(\mathbf{x}) := \sum_{e=1}^{N_e} \sum_{l=1}^m \left[\int_{-1}^1 G(\mathbf{x}, \mathbf{y}(\xi)) R_{l,p}(\xi) J^e(\xi) d\xi \right] q^l \end{cases} \quad (46)$$

where $J^e = d\Gamma/d\xi$ denotes Jacobian. Since the last term in Eq. (30) is calculated by radial basis function approximation, only geometric interpolation is needed in the final boundary integrals. Therefore, we only need to replace $d\Gamma$ in Eq. (40) with $J^e d\xi$. On the other hand, the integrals in the above equations are solved with Gauss integral method. In this work, the number of Gauss quadrature points is set as eight.

After collecting the equations for all collocation points and expressing them in matrix forms, one can obtain the following system of linear algebraic equations:

$$\mathbf{H}\hat{\mathbf{T}} - \mathbf{G}\mathbf{q} = -\mathbf{C}\hat{\mathbf{T}} \quad (47)$$

In this work, the finite difference method is used for solution of the above time domain equations.

It is worth noting that weakly singular integrals exist in Eq. (30). For BEM, it is very important to deal with this kind of singular integral effectively, because it directly determines the accuracy of BEM. Qu et al. [43] and Gong et al. [44] used exponential transformation to calculate accurately the weak singular integrals. Gao et al. [45] used the subtraction of singularity technique to overcome the singularity problem. In this work, the subtraction of singularity technique is also used for calculation of weak singular integrals, see [39].

4.3 Time Marching Method for Solving Transient Heat Conduction Problems

Eq. (27) contains a set of integral equations about time. First, we assume that the total time period is from t_0 to t_m . And then, divide it into m intervals. Thus, the i th moment can be expressed as:

$$t_i = t_0 + \frac{t_m - t_0}{m}i \quad (48)$$

Herein, \hat{T}^i denotes the temperature at t_i , and \hat{T}^{i+1} denotes the temperature at t_{i+1} . When the time marching method is used to solve the equations, we can obtain the derivative of temperature with respect to time, as follows:

$$\dot{\hat{T}} = \frac{\hat{T}^{i+1} - \hat{T}^i}{\Delta t} \quad (49)$$

where the time step Δt is equal to $\frac{t_m - t_0}{m}$. After interpolation approximation, we can get the temperature \hat{T} and its derivative q at $t \in [t_i, t_{i+1}]$, as follows:

$$\begin{cases} \hat{T} = \beta \hat{T}^{i+1} + (1 - \beta) \hat{T}^i \\ q = \beta q^{i+1} + (1 - \beta) q^i \end{cases} \quad (50)$$

where β changes from 0 to 1. When $\beta = 0$, it means forward difference scheme is used, when $\beta = 1$, it means backward difference scheme is used, otherwise it means central difference scheme is used. By substituting Eq. (50) into Eq. (30), we can obtain the following formulations

$$\begin{cases} (H^* \hat{T})(\mathbf{x}) := \beta (H^* \hat{T}^{i+1})(\mathbf{x}) + (1 - \beta) (H^* \hat{T}^i)(\mathbf{x}) \\ (G^* q)(\mathbf{x}) := \beta (G^* q^{i+1})(\mathbf{x}) + (1 - \beta) (G^* q^i)(\mathbf{x}) \end{cases} \quad (51)$$

Thus, Eq. (47) can be rewritten as

$$\beta \mathbf{H} \hat{\mathbf{T}}^{i+1} + (1 - \beta) \mathbf{H} \hat{\mathbf{T}}^i - \beta \mathbf{G} \mathbf{q}^{i+1} - (1 - \beta) \mathbf{G} \mathbf{q}^i = -\mathbf{C} \frac{\hat{\mathbf{T}}^{i+1} - \hat{\mathbf{T}}^i}{\Delta t} \quad (52)$$

After merging the similar terms, we can get the following expression:

$$\mathbf{P} \hat{\mathbf{T}}^{i+1} - \beta \mathbf{G} \mathbf{q}^{i+1} = -\mathbf{Q} \hat{\mathbf{T}}^i + (1 - \beta) \mathbf{G} \mathbf{q}^i \quad (53)$$

where

$$\begin{cases} \mathbf{P} = \beta \mathbf{H} + \mathbf{C} / \Delta t \\ \mathbf{Q} = (1 - \beta) \mathbf{H} - \mathbf{C} / \Delta t \end{cases} \quad (54)$$

When solving the unknown temperature $\hat{\mathbf{T}}^{i+1}$ or its derivative \mathbf{q}^{i+1} at t_{i+1} , the temperature $\hat{\mathbf{T}}^i$ and its derivative \mathbf{q}^i at t_i are all known. Moving all the unknown terms of Eq. (53) to the left-hand side and all the known terms to the right-hand side by considering the boundary conditions, one finally obtain the following system of linear equations:

$$\mathbf{B} \mathbf{x}^{i+1} = \mathbf{y}^{i+1} \quad (55)$$

where \mathbf{B} is the coefficient matrix, \mathbf{x}^{i+1} is the vector of unknown variable values at the collocation points, \mathbf{y}^{i+1} is the known vector. All the unknown state values can be obtained after Eq. (55) is solved.

5 Numerical Examples

5.1 Square Example

Consider a two-dimensional flat plate, the initial temperature is 100 K, the upper and lower boundaries are adiabatic, the left boundary temperature is maintained at 100 K. Convective boundary condition is applied to the right boundary, and the ambient temperature is 400 K, as shown in the Fig. 1. Material density $\rho = 271 \text{ kg/m}^3$, specific heat capacity $c_x = 871 \text{ J/(kg} \cdot \text{K)}$, thermal conductivity $k = 202.4 \text{ W/(m} \cdot \text{K)}$, convective heat transfer coefficient on the right boundary $h = 80 \text{ W/(m}^2 \cdot \text{K)}$ and thermal emissivity $\varepsilon = 1$. The boundary element model is as shown in the Fig. 1b. Each boundary is divided into 16 equally spaced linear elements, with a total of 64 boundary elements, 64 boundary nodes and 15 internal points.

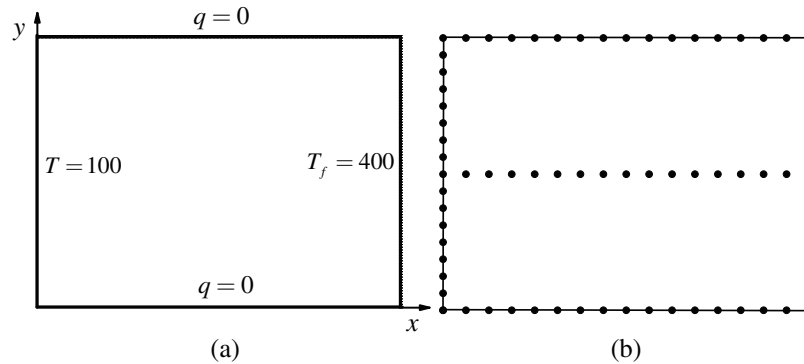


Figure 1: Plate model diagram and boundary conditions. (a) Plate boundary conditions. (b) Plate model diagram

Fig. 2 shows the distribution of temperature along the lower boundary of plate when $t = 100, 200, 400$ and 600 s. In this figure, the calculation results obtained by using IGABEM are compared with those of the finite volume method (Fluent), and it demonstrates the correctness and validity of this algorithm proposed in this work.

5.2 An Example of Elliptic Plate

Consider an elliptical plate with a long radius of 2 m and a short radius of 1 m. The initial temperature is 100 K, and the ambient temperature of the plate boundary is 400 K. Material density $\rho = 271 \text{ kg/m}^3$, specific heat capacity $c_x = 871 \text{ J/(kg} \cdot \text{K)}$, thermal conductivity $k = 202.4 \text{ W/(m} \cdot \text{K)}$, and boundary convective heat transfer coefficient $h = 80 \text{ W/(m}^2 \cdot \text{K)}$.

The shape of ellipse and the distribution of internal points are shown in Fig. 3. Fig. 4a shows the initial NURBS curve and the position of control points. After normalizing the knot vector, the parameter space vector of boundary element can be obtained, which is expressed as $\tilde{\Xi} = [0, 0.25, 0.5, 0.75, 1]$. Therefore, four boundary elements are formed, and the parameter space intervals are $[0, 0.25]; [0.25, 0.5]; [0.5, 0.75]; [0.75, 1]$ respectively. By splitting each NURBS elements equally and inserting new knots, the new control point sequence, knot vector and weight factor

vector after refinement can be obtained. For example, insert a new control point on the middle point of each initial NURBS cell parameter space to get the refined control point sequence shown in Fig. 4b, as shown by the black circle point “inserted knots.” After Bézier extraction operation, a new set of control point sequence is obtained, as shown in “BE operation knots.” For “initial knots” in Fig. 4a and “inserted knots” in Fig. 4b, we can find that inserting a new point will also change the position of the original point, but still describe the same curve. However, Bézier extraction operation does not change the position of the original control points, but inserts a new control point at the middle point of some elements. Then, the new control point sequence and Bézier extraction sequence when the initial NURBS cells are divided into 3, 4, 5 and 6 subunits are given in turn, as shown in Figs. 4c–4f respectively.

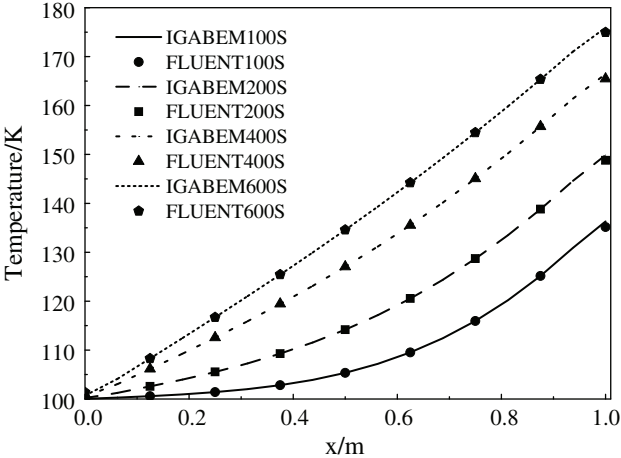


Figure 2: Temperature distribution along the lower boundary of the plate at different times

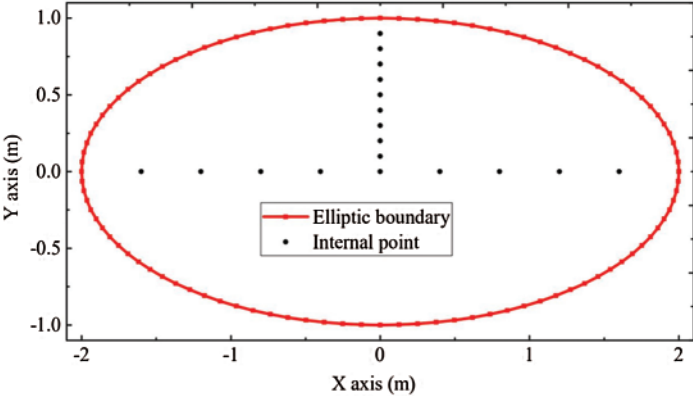


Figure 3: The NURBS curve and internal points for elliptic

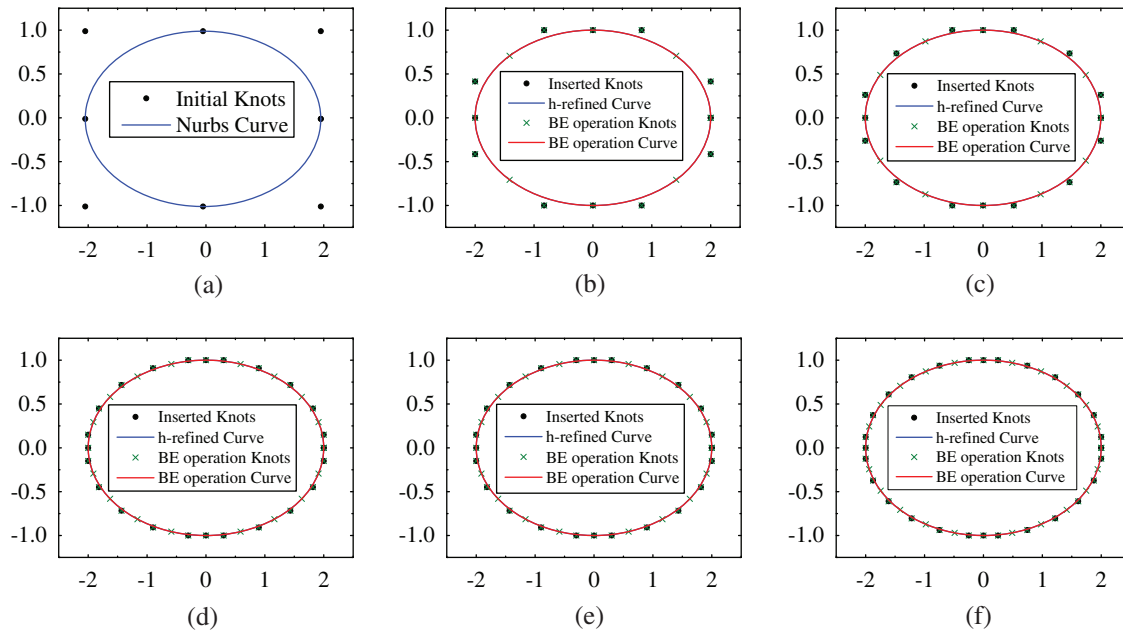


Figure 4: Geometric control points and NURBS curves of elliptical model with different refinement levels. (a) Initial. (b) Refine2. (c) Refine3. (d) Refine4. (e) Refine5. (f) Refine6

In this work, the finite difference method is used to solve the time-domain equation. Therefore, we first investigate the influence of time step on the calculation results, as shown in [Tab. 1](#). The temperature values at nine points on the x axis are calculated, where different time step is used, for example $\Delta t = 5, 6, 7, 8, 9$, and 10 . From this table, we can find that the results with different time steps have small deviation. In fact, too large step value may lead to large error of temperature gradient value, even the calculation is not correct at all. If the time step is too small, it will also lead to inaccurate calculation of temperature gradient value. Therefore, it is very important to select an appropriate time step value. It can be seen from [Tabs. 1](#) and [2](#) that it is appropriate to set the time step from 5 to 10 .

Table 1: Temperature values at several internal points with different time step values in 200 s

NSTEP	NODE 1 (-1.6, 0)	NODE 2 (-1.2, 0)	NODE 3 (-0.8, 0)	NODE 4 (-0.4, 0)	NODE 5 (0, 0)	NODE 6 (0.4, 0)	NODE 7 (0.8, 0)	NODE 8 (1.2, 0)	NODE 9 (1.6, 0)
5	136.156	117.783	109.520	106.530	105.821	106.535	109.522	117.784	136.159
6	135.715	117.454	109.308	106.386	105.691	106.386	109.309	117.456	135.717
7	135.273	117.127	109.096	106.238	105.562	106.238	109.097	117.128	135.275
8	136.043	117.776	109.588	106.621	105.908	106.621	109.588	117.778	136.045
9	135.603	117.450	109.376	106.472	105.778	106.472	109.377	117.452	135.605
10	135.969	117.773	109.632	106.678	105.966	106.678	109.633	117.775	135.971

In this work, we use the radial basis method to approximate the temperature gradient. Therefore, it is significant to investigate the influence of different types of radial basis functions on the numerical results. Herein, eight different types of radial basis functions are given, as shown in Tab. 3. Tab. 4 shows the comparison of temperature values at several calculation points when using different radial basis functions, where the time step is 5. By observing Tab. 4, it can be found that the deviation of calculation results in 200 s is very small when using different types of radial basis functions, more results in 400 s are also found in Tab. 5. It verifies the stability of the algorithm proposed.

Table 2: Temperature values at several internal points with different time step values in 400 s

NSTEP	NODE 1 (-1.6, 0)	NODE 2 (-1.2, 0)	NODE 3 (-0.8, 0)	NODE 4 (-0.4, 0)	NODE 5 (0, 0)	NODE 6 (0.4, 0)	NODE 7 (0.8, 0)	NODE 8 (1.2, 0)	NODE 9 (1.6, 0)
5	170.298	148.477	134.829	127.678	125.473	127.678	134.830	148.479	170.300
6	169.674	147.887	134.314	127.229	125.049	127.229	134.316	147.889	169.678
7	170.076	148.287	134.688	127.575	125.384	127.575	134.690	148.289	170.080
8	170.185	148.404	134.808	127.694	125.502	127.694	134.809	148.407	170.187
9	169.559	147.815	134.294	127.246	125.079	127.247	134.295	147.817	169.563
10	170.108	148.356	134.794	127.705	125.521	127.705	134.795	148.358	170.112

Table 3: Types of radial basis functions

NTYP	1	2	3	4	5	6	7	8
	R	$(1 + R^2)^{\frac{1}{2}}$	$(1 + R^2)^{-\frac{1}{2}}$	$(1 + R^2)^{-\frac{5}{2}}$	$(1 + R^2)^{-\frac{7}{2}}$	$(1 + R)^{-1}$	e^{-R}	e^{-R^2}

Table 4: Comparison of calculation results with different types of radial basis functions in 200 s

NTYP	NODE 1 (-1.6, 0)	NODE 2 (-1.2, 0)	NODE 3 (-0.8, 0)	NODE 4 (-0.4, 0)	NODE 5 (0, 0)	NODE 6 (0.4, 0)	NODE 7 (0.8, 0)	NODE 8 (1.2, 0)	NODE 9 (1.6, 0)
1	136.417	117.892	109.492	106.436	105.70	106.436	109.491	117.892	136.416
2	136.541	118.114	109.663	106.374	105.545	106.370	109.648	118.109	136.559
3	136.156	117.783	109.521	106.535	105.82	106.535	109.522	117.784	136.159
4	135.574	117.337	109.233	106.515	105.737	106.515	109.232	117.338	135.575
5	135.595	117.413	109.232	106.419	105.539	106.419	109.232	117.414	135.595
6	135.991	117.575	109.300	106.550	105.903	106.549	109.300	117.575	135.992
7	135.878	117.538	109.341	106.606	105.948	106.607	109.341	117.537	135.877
8	135.530	117.347	109.292	106.615	105.954	106.615	109.291	117.347	135.529

Table 5: Comparison of calculation results with different types of radial basis functions in 400 s

NTYP	NODE 1 (-1.6, 0)	NODE 2 (-1.2, 0)	NODE 3 (-0.8, 0)	NODE 4 (-0.4, 0)	NODE 5 (0, 0)	NODE 6 (0.4, 0)	NODE 7 (0.8, 0)	NODE 8 (1.2, 0)	NODE 9 (1.6, 0)
1	170.288	148.415	134.783	127.719	125.568	127.719	134.783	148.415	170.288
2	170.101	148.329	134.821	127.816	125.700	127.813	134.812	148.319	170.107
3	170.298	148.477	134.829	127.678	125.473	127.678	134.830	148.479	170.301
4	170.851	148.993	134.882	127.143	124.591	127.143	134.882	148.993	170.851
5	171.047	149.173	134.912	126.998	124.356	126.998	134.913	149.173	171.050
6	170.625	148.741	134.815	127.396	125.048	127.395	134.816	148.741	170.625
7	170.633	148.781	134.852	127.383	125.004	127.384	134.852	148.781	170.632
8	170.524	148.743	134.853	127.370	124.965	127.370	134.853	148.744	170.524

Figs. 5a and 5b respectively show the comparison of temperature values at points on the x axis and y axis based on traditional BEM and IGABEM algorithm developed in this paper. From these two graphs, it can be found that the results of the IGABEM algorithm are consistent with the results of conventional BEM, which verifies the correctness of the algorithm developed in this work.

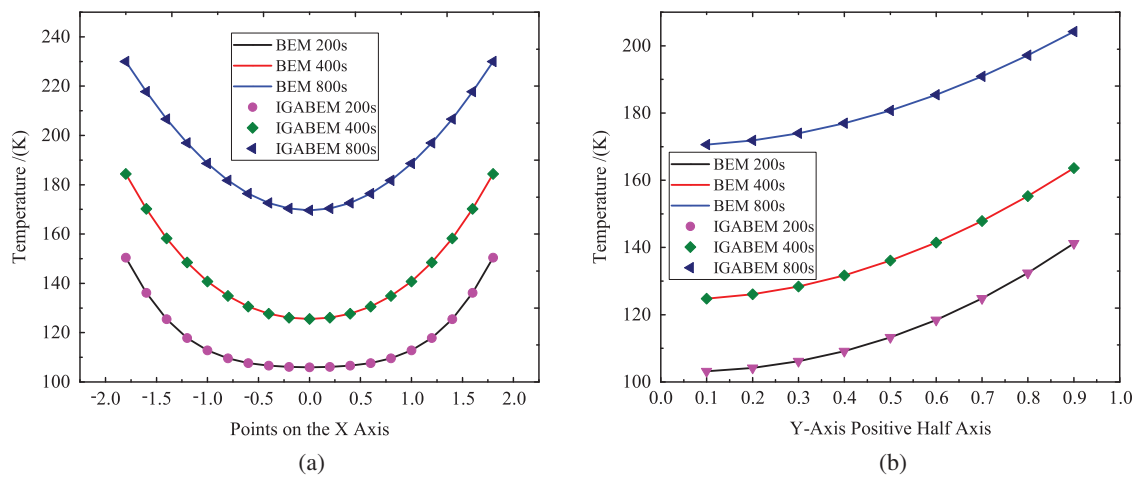


Figure 5: Temperature distribution on X-axis and Y-axis, respectively. (a) Temperature of X-axis symmetry point. (b) Temperature of positive half axis of Y-axis

5.3 An Example of Octagonal Leaf

In this section, we consider an example of octagonal leaf, as shown in Fig. 6. The initial temperature is 100 K, and the ambient temperature of the structural boundary is 600 K. Other physical parameters are consistent with the previous example. The new control point sequence and Bézier extraction sequence when the initial NURBS cells are divided into 2, 3, 4, 5 and 6 subunits are given in turn, as shown in Figs. 7b–7f respectively.

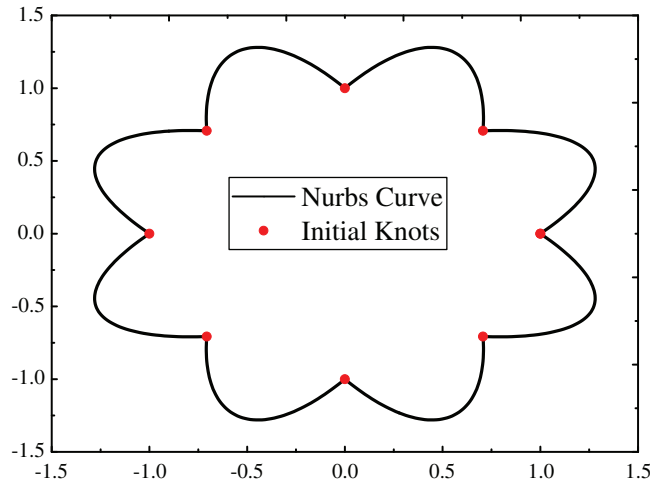


Figure 6: The NURBS curve and internal points for octagonal model

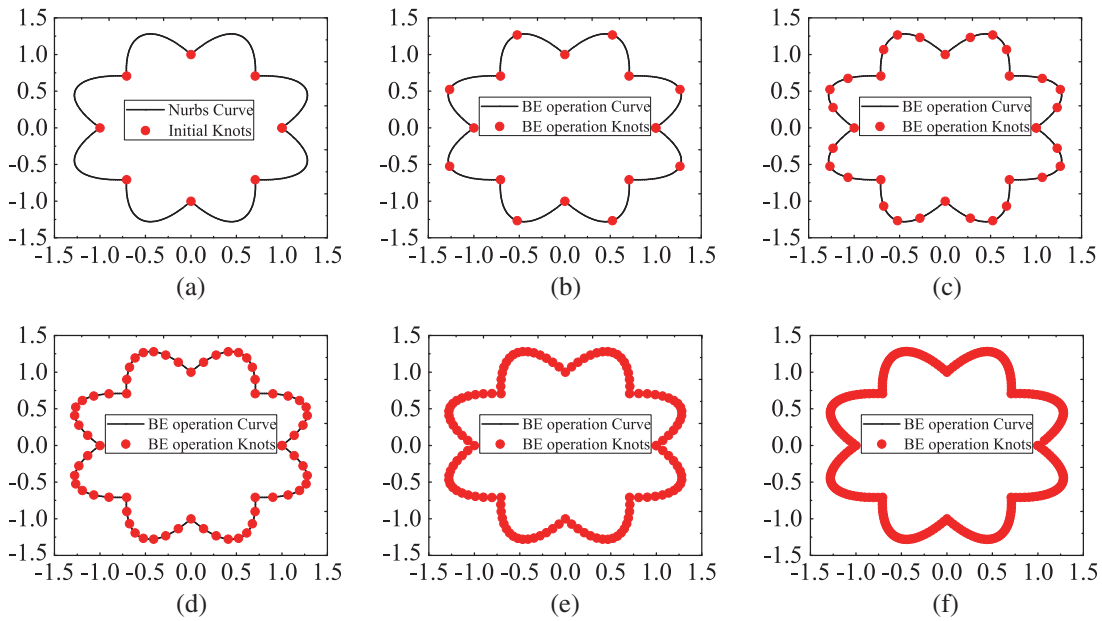


Figure 7: Generation of NURBS curve control points for octagon model. (a) Initial. (b) Refine2. (c) Refine3. (d) Refine4. (e) Refine5. (f) Refine6

Similar to the previous example, we investigate the influence of different radial basis functions on the calculation results, as shown in [Tab. 6](#) for 200 s and in [Tab. 7](#) for 400 s. It shows that there is a small deviation in the calculation results when different radial basis functions are used, which verifies the stability of the algorithm in this work.

The influence of time step parameters on the calculation accuracy can be found in [Tab. 8](#) for 200 s and in [Tab. 9](#) for 400 s. Similar to the previous results, the change of time step parameters

from 5 to 10 has little effect on the calculation results, which verifies the stability of the algorithm in this work.

Table 6: Comparison of calculation results with different types of radial basis functions in 200 s for octagonal leaf

NTYP	NODE 1 (-0.6, 0)	NODE 2 (-0.2, 0)	NODE 3 (0, 0)	NODE 4 (0.4, 0)	NODE 5 (0, 0.5)	NODE 6 (0, -0.3)	NODE 7 (0, -0.7)
1	135.010	111.721	109.027	119.731	126.151	115.106	145.490
2	135.254	112.784	110.301	120.490	126.821	115.988	145.524
3	135.570	112.847	110.281	120.632	126.975	116.130	145.897
4	135.335	111.355	108.550	119.775	126.252	114.895	146.002
5	135.087	109.657	106.608	118.714	125.596	113.462	146.163
6	134.604	108.323	105.146	117.715	124.901	112.269	145.893
7	134.215	107.510	104.270	117.068	124.414	111.519	145.591
8	134.725	109.683	106.691	118.533	125.347	113.404	145.643

Table 7: Comparison of calculation results with different types of radial basis functions in 400 s for octagonal leaf

NTYP	NODE 1 (-0.6, 0)	NODE 2 (-0.2, 0)	NODE 3 (0, 0)	NODE 4 (0.4, 0)	NODE 5 (0, 0.5)	NODE 6 (0, -0.3)	NODE 7 (0, -0.7)
1	197.535	168.822	165.221	179.613	187.604	173.319	209.127
2	197.461	169.004	165.495	179.734	187.701	173.441	209.002
3	197.869	169.333	165.777	180.052	188.015	173.806	209.420
4	198.077	169.287	165.685	180.174	188.180	173.813	209.715
5	198.025	169.040	165.398	179.983	188.084	173.606	209.729
6	197.647	168.557	164.890	179.494	187.650	173.136	209.365
7	197.342	168.207	164.522	179.133	187.315	172.786	209.064
8	197.590	168.590	164.942	179.512	187.604	173.151	209.280

Table 8: Temperature at internal points with different time step values in 200 s for octagonal leaf model

NSTEP	NODE 1 (-0.6, 0)	NODE 2 (-0.2, 0)	NODE 3 (0, 0)	NODE 4 (0.4, 0)	NODE 5 (0, 0.5)	NODE 6 (0, -0.3)	NODE 7 (0, -0.7)
5	135.335	111.355	108.550	119.775	126.252	114.895	146.002
6	134.721	111.012	108.241	119.303	125.704	114.502	145.315
7	134.112	110.675	107.943	118.834	125.159	114.114	144.629
8	135.381	111.702	108.934	119.978	126.376	115.187	145.960
9	134.771	111.361	108.630	119.507	125.831	114.795	145.276
10	135.413	111.930	109.187	120.112	126.459	115.379	145.935

Table 9: Temperature at internal points with different time step values in 400 s for octagonal leaf model

NSTEP	NODE 1 (−0.6, 0)	NODE 2 (−0.2, 0)	NODE 3 (0, 0)	NODE 4 (0.4, 0)	NODE 5 (0, 0.5)	NODE 6 (0, −0.3)	NODE 7 (0, −0.7)
5	198.077	169.287	165.685	180.174	188.180	173.813	209.715
6	196.833	168.046	164.445	178.926	186.929	172.568	208.479
7	197.713	168.968	165.370	179.833	187.825	173.484	209.341
8	197.985	169.272	165.677	180.124	188.108	173.783	209.602
9	196.744	168.037	164.444	178.881	186.859	172.544	208.367
10	197.925	169.263	165.676	180.093	188.060	173.764	209.527

6 Conclusion

This paper applied IGABEM to the heat conduction problem of temperature field. IGABEM can accurately represent the geometric model. In addition to the advantages of dimensionality reduction, IGABEM can also realize the seamless integration of CAD modeling and numerical analysis. The use of Bézier extraction technique further simplifies the implementation of IGA. This method transformed the basis functions of NURBS into Bernstein polynomials and maintained the consistency of geometric models. The main feature of this method was that the iterative calculation of NURBS basis function was avoided in the process of element physical interpolation calculation of BEM, which can effectively improve the computational efficiency. The domain integral term was successfully converted to the boundary integral in IGABEM with radial integration method. The present method provides a powerful tool for fast and high fidelity simulation of transient heat conduction problems commonly encountered in numerous industrial sectors.

Funding Statement: This research was funded by National Natural Science Foundation of China (NSFC) under Grant Nos. 11702238, 51904202, and 11902212, and Nanhu Scholars Program for Young Scholars of XYNU.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Hughes, T. J. R., Cottrell, J. A., Bazilevs, Y. (2005). Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39–41), 4135–4195. DOI 10.1016/j.cma.2004.10.008.
2. Cottrell, J. A., Hughes, T. J. R., Reali, A. (2007). Studies of refinement and continuity in isogeometric structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 196(41–44), 4160–4183. DOI 10.1016/j.cma.2007.04.007.
3. Bazilevs, Y., Michler, C., Calo, V. M., Hughes, T. J. R. (2010). Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly enforced boundary conditions on unstretched meshes. *Computer Methods in Applied Mechanics and Engineering*, 199(13–16), 780–790. DOI 10.1016/j.cma.2008.11.020.

4. Nagy, A. P., Abdalla, M. M., Gürdal, Z. (2010). Isogeometric sizing and shape optimisation of beam structures. *Computer Methods in Applied Mechanics and Engineering*, 199(17–20), 1216–1230. DOI 10.1016/j.cma.2009.12.010.
5. Benson, D. J., Bazilevs, Y., Hsu, M. C., Hughes, T. J. R. (2010). Isogeometric shell analysis: The Reissner–Mindlin shell. *Computer Methods in Applied Mechanics and Engineering*, 199(5–8), 276–289. DOI 10.1016/j.cma.2009.05.011.
6. De Lorenzis, L., Temizer, İ., Wriggers, P., Zavarise, G. (2011). A large deformation frictional contact formulation using NURBS-based isogeometric analysis. *International Journal for Numerical Methods in Engineering*, 87(13), 1278–1300. DOI 10.1002/nme.3159.
7. Borden, M. J., Scott, M. A., Evans, J. A., Hughes, T. J. R. (2011). Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering*, 87(1–5), 15–47. DOI 10.1002/nme.2968.
8. Mierzwiczak, M., Chen, W., Fu, Z. (2015). The singular boundary method for steady-state nonlinear heat conduction problem with temperature-dependent thermal conductivity. *International Journal of Heat & Mass Transfer*, 91, 205–217. DOI 10.1016/j.ijheatmasstransfer.2015.07.051.
9. Fu, Z. J., Qin, Q. H., Chen, W. (2011). Hybrid-trefftz finite element method for heat conduction in nonlinear functionally graded materials. *Engineering Computations*, 28(5), 578–599. DOI 10.1108/02644401111141028.
10. Li, T., Gao, Y., Han, D., Yang, F., Yu, B. (2020). A novel POD reduced-order model based on EDFM for steady-state and transient heat transfer in fractured geothermal reservoir. *International Journal of Heat and Mass Transfer*, 146, 118783. DOI 10.1016/j.ijheatmasstransfer.2019.118783.
11. Wang, Y., Liao, Z., Shi, S., Wang, Z., Hien, Poh L. (2020). Data-driven structural design optimization for petal-shaped auxetics using isogeometric analysis. *Computer Modeling in Engineering & Sciences*, 122(2), 433–458. DOI 10.32604/cmescs.2020.08680.
12. Chen, L., Marburg, S., Chen, H., Zhang, H., Gao, H. (2017). An adjoint operator approach for sensitivity analysis of radiated sound power in fully coupled structural-acoustic systems. *Journal of Computational Acoustics*, 25(1), 1750003. DOI 10.1142/S0218396X17500035.
13. Banerjee, P. K., Cathie, D. N. (1980). A direct formulation and numerical implementation of the boundary element method for two-dimensional problems of elasto-plasticity. *International Journal of Mechanical Sciences*, 22(4), 233–245. DOI 10.1016/0020-7403(80)90038-7.
14. Cruse, T. A. (1996). Bie fracture mechanics analysis: 25 years of developments. *Computational Mechanics*, 18(1), 1–11. DOI 10.1007/BF00384172.
15. Seybert, A. F., Soenarko, B., Rizzo, F. J., Shippy, D. J. (1985). An advanced computational method for radiation and scattering of acoustic waves in three dimensions. *Journal of the Acoustical Society of America*, 77(2), 362–368. DOI 10.1121/1.391908.
16. Chen, L. L., Zhao, W. C., Liu, C., Chen, H. B., Marburg, S. (2019). Isogeometric fast multipole boundary element method based on burton-miller formulation for 3d acoustic problems. *Archives of Acoustics*, 44, 475–492.
17. Chen, L., Chen, H., Zheng, C., Marburg, S. (2016). Structural-acoustic sensitivity analysis of radiated sound power using a finite element/discontinuous fast multipole boundary element scheme. *International Journal for Numerical Methods in Fluids*, 82(12), 858–878. DOI 10.1002/flid.4244.
18. Zhao, W. C., Chen, L. L., Chen, H. B., Marburg, S. (2019). Topology optimization of exterior acoustic-structure interaction systems using the coupled fem-bem method. *International Journal for Numerical Methods in Engineering*, 119(1), 1–28. DOI 10.1002/nme.6039.
19. Chen, L. L., Zhang, Y., Lian, H., Atroshchenko, E., Ding, C. et al. (2020). Seamless integration of computer-aided geometric modeling and acoustic simulation: Isogeometric boundary element methods based on Catmull-Clark subdivision surfaces. *Advances in Engineering Software*, 149, 102879. DOI 10.1016/j.advengsoft.2020.102879.
20. Politis, C., Ginnis, A. I., Kaklis, P. D., Belibassakis, K., Feurer, C. (2009). An isogeometric BEM for exterior potential-flow problems in the plane. *2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling*, San Francisco, California, USA: ACM.

21. Simpson, R. N., Bordas, S. P. A., Trevelyan, J., Rabczuk, T. (2012). A two-dimensional isogeometric boundary element method for elastostatic analysis. *Computer Methods in Applied Mechanics and Engineering*, 209, 87–100. DOI 10.1016/j.cma.2011.08.008.
22. Simpson, R. N., Bordas, S. P. A., Lian, H., Trevelyan, J. (2013). An isogeometric boundary element method for elastostatic analysis: 2D implementation aspects. *Computers & Structures*, 118, 2–12. DOI 10.1016/j.compstruc.2012.12.021.
23. An, Z., Yu, T., Bui, T. Q., Wang, C., Trinh, N. A. (2018). Implementation of isogeometric boundary element method for 2-D steady heat transfer analysis. *Advances in Engineering Software*, 116, 36–49. DOI 10.1016/j.advengsoft.2017.11.008.
24. Scott, M. A., Simpson, R. N., Evans, J. A., Lipton, S., Bordas, S. P. A. et al. (2013). Isogeometric boundary element analysis using unstructured t-splines. *Computer Methods in Applied Mechanics and Engineering*, 254, 197–221. DOI 10.1016/j.cma.2012.11.001.
25. Li, S., Trevelyan, J., Zhang, W., Wang, D. (2018). Accelerating isogeometric boundary element analysis for 3-dimensional elastostatics problems through black-box fast multipole method with proper generalized decomposition. *International Journal for Numerical Methods in Engineering*, 114(9), 975–998. DOI 10.1002/nme.5773.
26. Nguyen, B. H., Tran, H. D., Anitescu, C., Zhuang, X., Rabczuk, T. (2016). An isogeometric symmetric galerkin boundary element method for two-dimensional crack problems. *Computer Methods in Applied Mechanics and Engineering*, 306, 252–275. DOI 10.1016/j.cma.2016.04.002.
27. Peng, X., Atroshchenko, E., Kerfriden, P., Bordas, S. P. A. (2017). Linear elastic fracture simulation directly from CAD: 2D NURBS-based implementation and role of tip enrichment. *International Journal of Fracture*, 204(1), 55–78. DOI 10.1007/s10704-016-0153-3.
28. Peng, X., Atroshchenko, E., Kerfriden, P., Bordas, S. P. A. (2017). Isogeometric boundary element methods for three dimensional static fracture and fatigue crack growth. *Computer Methods in Applied Mechanics and Engineering*, 316, 151–185. DOI 10.1016/j.cma.2016.05.038.
29. Simpson, R. N., Liu, Z., Vazquez, R., Evans, J. A. (2018). An isogeometric boundary element method for electromagnetic scattering with compatible B-spline discretizations. *Journal of Computational Physics*, 362, 264–289. DOI 10.1016/j.jcp.2018.01.025.
30. Simpson, R. N., Scott, M. A., Taus, M., Thomas, D. C., Lian, H. (2014). Acoustic isogeometric boundary element analysis. *Computer Methods in Applied Mechanics and Engineering*, 269, 265–290. DOI 10.1016/j.cma.2013.10.026.
31. Peake, M. J., Trevelyan, J., Coates, G. (2015). Extended isogeometric boundary element method (XIBEM) for three-dimensional medium-wave acoustic scattering problems. *Computer Methods in Applied Mechanics and Engineering*, 284, 762–780. DOI 10.1016/j.cma.2014.10.039.
32. Keuchel, S., Hagelstein, N. C., Zaleski, O., von Estorff, O. (2017). Evaluation of hypersingular and nearly singular integrals in the isogeometric boundary element method for acoustics. *Computer Methods in Applied Mechanics and Engineering*, 325, 488–504. DOI 10.1016/j.cma.2017.07.025.
33. Chen, L., Marburg, S., Zhao, W., Liu, C., Chen, H. (2019). Implementation of isogeometric fast multipole boundary element methods for 2D half-space acoustic scattering problems with absorbing boundary condition. *Journal of Theoretical and Computational Acoustics*, 27(2), 1850024. DOI 10.1142/S259172851850024X.
34. Chen, L. L., Lian, H., Liu, Z., Chen, H. B., Atroshchenko, E. et al. (2019). Structural shape optimization of three dimensional acoustic problems with isogeometric boundary element methods. *Computer Methods in Applied Mechanics and Engineering*, 355, 926–951. DOI 10.1016/j.cma.2019.06.012.
35. Chen, L., Liu, C., Zhao, W., Liu, L. (2018). An isogeometric approach of two dimensional acoustic design sensitivity analysis and topology optimization analysis for absorbing material distribution. *Computer Methods in Applied Mechanics and Engineering*, 336, 507–532. DOI 10.1016/j.cma.2018.03.025.
36. Li, K., Qian, X. (2011). Isogeometric analysis and shape optimization via boundary integral. *Computer-Aided Design*, 43(11), 1427–1437. DOI 10.1016/j.cad.2011.08.031.

37. Lian, H., Kerfriden, P., Bordas, S. P. A. (2016). Implementation of regularized isogeometric boundary element methods for gradient-based shape optimization in two-dimensional linear elasticity. *International Journal for Numerical Methods in Engineering*, 106(12), 972–1017. DOI 10.1002/nme.5149.
38. Lian, H., Kerfriden, P., Bordas, S. P. A. (2017). Shape optimization directly from CAD: An isogeometric boundary element approach using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 317, 1–41. DOI 10.1016/j.cma.2016.11.012.
39. Liu, C., Chen, L., Zhao, W., Chen, H. (2017). Shape optimization of sound barrier using an isogeometric fast multipole boundary element method in two dimensions. *Engineering Analysis with Boundary Elements*, 85, 142–157. DOI 10.1016/j.enganabound.2017.09.009.
40. Chen, L., Lu, C., Lian, H., Liu, Z., Zhao, W. et al. (2020). Acoustic topology optimization of sound absorbing materials directly from subdivision surfaces with isogeometric boundary element methods. *Computer Methods in Applied Mechanics and Engineering*, 362, 112806. DOI 10.1016/j.cma.2019.112806.
41. Gao, X. W. (2002). The radial integration method for evaluation of domain integrals with boundary-only discretization. *Engineering Analysis with Boundary Elements*, 26(10), 905–916. DOI 10.1016/S0955-7997(02)00039-5.
42. Gao, X. W., Peng, H. F. K. Y. J. W. (2015). *Theory and program of advanced boundary element method*. Beijing, China: Science Press.
43. Qu, X. Y., Dong, C. Y., Bai, Y., Gong, Y. P. (2018). Isogeometric boundary element method for calculating effective property of steady state thermal conduction in 2D heterogeneities with a homogeneous interphase. *Journal of Computational and Applied Mathematics*, 343, 124–138. DOI 10.1016/j.cam.2018.04.053.
44. Gong, Y., Trevelyan, J., Hattori, G., Dong, C. (2018). Hybrid nearly singular integration for isogeometric boundary element analysis of coatings and other thin 2D structures. *Computer Methods in Applied Mechanics and Engineering*, 346, 642–673. DOI 10.1016/j.cma.2018.12.019.
45. Gao, X., Trevor, G. D. (2002). *Boundary element programming in mechanics*. Cambridge, UK: Cambridge University Press.