Contents lists available at ScienceDirect

Engineering Analysis with Boundary Elements

journal homepage: www.elsevier.com/locate/enganabound

B-spline approximation in boundary face method for three-dimensional linear elasticity

Jinliang Gu, Jianming Zhang*, Xiaomin Sheng, Guanyao Li

State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, China

ARTICLE INFO

Article history:
Received 7 November 2010
Accepted 31 May 2011
Available online 1 July 2011

Keywords:
B-spline basis functions
Linear elasticity
Boundary face method
BIEs

ABSTRACT

In this paper, basis functions generated from B-spline or Non-Uniform Rational B-spline (NURBS), are used for approximating the boundary variables to solve the 3D linear elasticity Boundary Integral Equations (BIEs). The implementation is based on the BFM framework in which both boundary integration and variable approximation are performed in the parametric spaces of the boundary surfaces to keep the exact geometric information in the BIEs. In order to reduce the influence of tensor product of B-spline and make the discretization of a body surface easier, the basis functions defined in global intervals are translated into local form. B-spline fitting function built with the local basis functions is converted into an interpolation type of function in which the nodal values of the boundary variables are used for control points. Numerical tests for 3D linear elasticity problems show that the BFM with B-spline basis functions outperforms that with the well-known Moving Least Square (MLS) approximation.

1. Introduction

In the Finite Element Analysis (FEA), not only physical variables but also the geometric models are approximated using elements. The geometric error of the model may result in serious accuracy problem, which has attracted many researchers' attention. Hughes et al. firstly introduced the conception of isogeometric analysis [1] to deal with it. In Ref. [1], they have given a detailed description to the concept using some numerical examples, such as thin shell and fluid mechanics analysis, which are notoriously sensitive to the geometric imperfection in FEA. In order to obtain acceptable numerical results, much time and computational resource are consumed for the needless work of further mesh generation to construct more accurate finite element geometry. Actually, the problem also exists in the area of Boundary Element Analysis (BEA). Recently, Zhang et al. [2] proposed the Boundary Face Method (BFM), which uses the exact geometry for analysis in the BIEs. It thoroughly overcomes the shortcomings associated with approximate geometric models. In the implementation of BFM, he makes use of the approximation functions constructed by the Moving Least-squares (MLS) method. Here in this paper, we present a new implementation of the BFM, where the B-spline basis functions [3,4] instead of the MLS are used for approximating the boundary values. Our work is largely stimulated by the conception of isogeometric analysis.

Isogeometric analysis is proposed to overcome some difficulties that occur in classical design loop [4–7]. Within the concept, basis functions generated from non-uniform rational B-spline (NURBS) play a key role in offering exact geometry representation, simplification of design optimization and tighter integration of analysis and CAD. Although NURBS is not the unique tool for implementation of isogeometric analysis, it is the one in the most widespread use so far. It may be owing to the two facts, one is that NURBS is the standard approach for representation of free form curves and sculptured surfaces in CAD, and can represent elementary shapes such as sphere, cylinders, and torus exactly. These make it possible for NURBS to express all of CAD models in a uniform geometric representation. Another is that the basis functions originated in the field of approximation theory and inherently have some useful mathematic and geometric properties, such as the local support, nonnegative and partition of unit. These are attractive properties with regard to numerical stability. Actually, much work that focus on the B-spline basis functions had been carried out in the realm of Finite Element Analysis (FEA) before isogeometric analysis. For example, the basis functions were used for solving partial differential equations [8] and problems where continuity of higher derivatives is required [9–11]. Interpolation function derived from the basis functions used in the analysis of engineering problems shows that these can result in a significant reduction of the number of degrees of freedom [12].

* Correspondence author: Tel.: +86 731 8823061.
E-mail address: zhangjianm@gmail.com (J. Zhang).

0955-7997/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.enganabound.2011.05.013

However, there are little available literatures that refer to the application of B-spline basis functions in the BEA. Here, we introduce them in the framework of the BFM. Coupling the BFM with the B-spline basis functions, two obvious advantages are absorbed into the new implementation comparing with traditional BEM. The first one is that the integrand quantities, such as the coordinates of Gauss integration points, Jacobian and out normal are calculated directly from exact geometry instead of element approximation, an important property inherited from BFM. The second one is that B-spline approximation represents continuity between elements, a distinct property that must be explicitly expressed in the BEA cases. Furthermore, a richer set of refinement operations for the basis functions makes B-spline more flexibly to be used for adaptive analysis in BFM. It should be noted that the B-spline basis functions are only resorted to the boundary of a geometric body in our method, making these implemented much simpler than the FEM, and also the boundary can be related directly to a CAD model. In our scheme, B-spline basis functions defined in global interval are converted into local form. Thus, construction of B-spline approximation functions will be no longer fully restricted by the fashion of tensor product, which requires that the control points must lie topologically in a rectangular grid. This strategy largely reduces the load for computation and saves a lot of computational resource. The surface of a geometric model is discretized by a set of nodes. Boundary values on these nodes, such as the displacement and stress, are used as the control points for constructing B-spline approximation functions. In that the original constructed bivariate B-spline functions do not automatically satisfy the Kronecker delta property, the fitting type of that is converted into the interpolation type using the inverse transformation. The interpolation type is also called as B-spline approximation functions. To further compute the boundary unknowns, e.g., the components of stress in the BIEs, the partial derivatives of basis functions are first expressed in the local forms, and then evaluated.

For comparison, The MLS [13] is also introduced in the same framework used for approximation function. As a reliable method for meshless analysis, the MLS is widely applied in data fitting and interpolation. The construction MLS approximation function only needs some points in the local support region, and reasonable results associated with high accuracy usually are obtained in numerical tests. B-spline basis functions have some properties similar with MLS, such as the local support. Some researchers feel that B-spline basis functions have the potential to impact in the area of meshless methods [14]. The implementation of B-spline approximation is much simpler than that of MLS. Moreover, in the B-spline implementation, there is no need to choose the local support size for a node, which is a troublesome issue in the MLS. Comparison between the two methods by numerical examples has fully demonstrated that B-spline basis functions are more efficient than the MLS for solving potential problems.

This paper is organized as follows. In Section 2, there are expressions of B-spline basis functions with degree two and three, and bivariate B-spline and NURBS functions together with their derivatives. In Section 3, B-spline as an approximate tool is used in the discretization of BIEs. Numerical examples for 3D linear elasticity problems are given in Section 4. Finally, we present the conclusions for our work in Section 5.

2. Bivariate B-spline and NURBS functions

NURBS and B-spline functions are built from B-spline basis functions [3,4]. In traditional way, these basis functions are expressed in global form. Here, we transform the functions from global form to local form to make them more flexible for subsequent application. The B-spline basis functions in global form are defined recursively for zero degree

$$B_{i,k}(\xi)=\begin{cases}1, & \xi_i \le \xi \le \xi_{i+1}\\ 0, & \text{otherwise}\end{cases} \quad k=0 \tag{1}$$

and for non-zero degrees

$$B_{i,k}(\xi)=\frac{\xi-\xi_i}{\xi_{i+k}-\xi_i}B_{i,k-1}(\xi)+\frac{\xi_{i+k+1}-\xi}{\xi_{i+k+1}-\xi_{i+1}}B_{i+1,k-1}(\xi), \quad k>0 \tag{2}$$

Assuming that 0/0=0 and $\Xi=[\xi_1,\xi_2,\ldots,\xi_{n+k+1}]$, where Ξ is a knot vector; ξ_i is the ith knot and n is the total number of basis functions corresponding to the number of control points. It should be noted that knot values presented in the knot vector Ξ, in this paper, are given by: $\xi_1=\xi_2=\ldots=\xi_{k+1}=0$, $\xi_{n+1}=\xi_{n+2}=\ldots=\xi_{n+k+1}=1$, and $\xi_i=i/(n-k)$ for $i=k+1, k+2,\ldots n$.

By Eq. (2), higher degrees of B-spline basis functions can be deducted recursively. Here, we focus on the quadratic and cubic functions.

Global form of quadratic B-spline basis functions can be written as

$$B_{i,2}=\begin{cases}\frac{(\xi-\xi_i)^2}{(\xi_{i+2}-\xi_i)(\xi_{i+1}-\xi_i)}, & \xi_i \le \xi < \xi_{i+1}\\ \frac{(\xi-\xi_i)(v)}{(\xi_{i+2}-\xi_i)(\xi_{i+2}-\xi_{i+1})}+\frac{(\xi_{i+3}-\xi)(\xi-\xi_{i+1})}{(\xi_{i+3}-\xi_{i+1})(\xi_{i+2}-\xi_{i+1})}, & \xi_{i+1} \le \xi < \xi_{i+2}\\ \frac{(\xi_{i+3}-\xi)^2}{(\xi_{i+3}-\xi_{i+1})(\xi_{i+3}-\xi_{i+2})}, & \xi_{i+2} \le \xi < \xi_{i+3}\end{cases} \tag{3}$$

The above three piecewise polynomials can be transformed into a local form, which is defined in the non-zero subinterval $[\xi_i,\xi_{i+1})$ and expressed as

$$\begin{cases}B^{(3)}_{i-2,2}(\xi)=\frac{(\xi_{i+1}-\xi)^2}{(\xi_{i+1}-\xi_{i-1})(\xi_{i+1}-\xi_i)}\\ B^{(2)}_{i-1,2}(\xi)=\frac{(\xi-\xi_{i-1})(\xi_{i+1}-\xi)}{(\xi_{i+1}-\xi_{i-1})(\xi_{i+1}-\xi_i)}+\frac{(\xi_{i+2}-\xi)(\xi-\xi_i)}{(\xi_{i+2}-\xi_i)(\xi_{i+1}-\xi_i)}\\ B^{(1)}_{i,2}(\xi)=\frac{(\xi-\xi_i)^2}{(\xi_{i+2}-\xi_i)(\xi_{i+1}-\xi_i)}\end{cases} \tag{4}$$

where $B^{(l)}_{i-j,k}(\xi)$ is the lth segment of the $(i-j)$th global basis function in the $[\xi_i,\xi_{i+1})$, k is the degree of basis functions, and $l=1,2,\ldots,k+1$.

Analogously, the cubic global form of B-spline basis function also can be transformed into a local form, which is defined in the non-zero interval $[\xi_i,\xi_{i+1})$ and expressed as

$$\begin{cases}B^{(4)}_{i-3,3}(\xi)=\frac{(\xi_{i+1}-\xi)^3}{(\xi_{i+1}-\xi_{i-2})(\xi_{i+1}-\xi_{i-1})(\xi_{i+1}-\xi_i)}\\ B^{(3)}_{i-2,3}(\xi)=\frac{(\xi_{i+1}-\xi)^2(\xi-\xi_{i-2})}{(\xi_{i+1}-\xi_{i-1})(\xi_{i+1}-\xi_{i-2})(\xi_{i+1}-\xi_i)}\\ \qquad +\frac{(\xi_{i+2}-\xi)(\xi_{i+1}-\xi)(\xi-\xi_{i-1})}{(\xi_{i+2}-\xi_{i-1})(\xi_{i+1}-\xi_{i-1})(\xi_{i+1}-\xi_i)}\\ \qquad +\frac{(\xi_{i+2}-\xi)^2(\xi-\xi_i)}{(\xi_{i+2}-\xi_{i-1})(\xi_{i+2}-\xi_i)(\xi_{i+1}-\xi_i)}\\ B^{(2)}_{i-1,3}(\xi)=\frac{(\xi_{i+1}-\xi)(\xi-\xi_{i-1})^2}{(\xi_{i+2}-\xi_{i-1})(\xi_{i+1}-\xi_{i-1})(\xi_{i+1}-\xi_i)}\\ \qquad +\frac{(\xi_{i+2}-\xi)(\xi-\xi_i)(\xi-\xi_{i-1})}{(\xi_{i+2}-\xi_i)(\xi_{i+2}-\xi_{i-1})(\xi_{i+1}-\xi_i)}\\ \qquad +\frac{(\xi_{i+3}-\xi)(\xi-\xi_i)^2}{(\xi_{i+3}-\xi_i)(\xi_{i+2}-\xi_i)(\xi_{i+1}-\xi_i)}\\ B^{(1)}_{i,3}(\xi)=\frac{(\xi-\xi_i)^3}{(\xi_{i+3}-\xi_i)(\xi_{i+2}-\xi_i)(\xi_{i+1}-\xi_i)}\end{cases} \tag{5}$$

The transformation for the B-spline basis functions can be described intuitively by Fig. 1. We firstly list all of the cubic global basis functions in the interval [0,1] in Fig. 1(a). In each subinterval, there are four different piecewise polynomials overlapped, which are from four successive global basis functions. Then, we can extract the superposed segments that exist in the same subinterval, for example those in the [0.4, 0.6] depicted in the Fig. 1(b). These piecewise polynomials actually are a local form defined in a local subinterval.

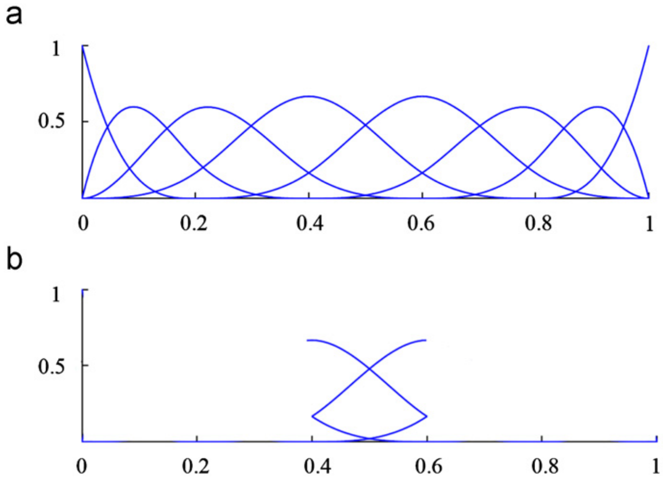

Fig. 1. B-spline basis functions located in the subinterval [0.4, 0.6].

Using the local B-spline basis functions, we can easily obtain the local form of bivariate B-spline function. It is defined by

$$P(\xi,\eta)=\sum_{i=1}^{n}\sum_{j=1}^{m}B_{i,k}(\xi)B_{j,l}(\eta)p_{ij}=\sum_{r=0}^{k}\sum_{s=0}^{l}N_{rs}(\xi,\eta)p_{rs} \tag{6}$$

where $N_{rs}(\xi,\eta)$ is expressed as

$$N_{rs}(\xi,\eta)=\begin{cases}B^{(r+1)}_{i_r-r,k}(\xi)B^{(s+1)}_{j_s-s,l}(\eta), & \xi\in\left[\xi_{i_r},\xi_{i_r+1}\right),\ \eta\in\left[\eta_{j_s},\eta_{j_s+1}\right)\\ 0, & \text{otherwise}\end{cases} \tag{7}$$

$p_{ij}(x_{ij},y_{ij},z_{ij})$ are a set of control points. ξ and η are parametric coordinates with their values lying in the $[\xi_1,\xi_2,\ldots,\xi_{n+k+1}]$ and $[\eta_1,\eta_2,\ldots,\eta_{m+l+1}]$, respectively. Where $p_{rs}=p_{(ir-r)(js-s)}$, $i_r\in\{k+1,\ldots,n\}$ and $j_s\in\{l+1,\ldots,m\}$. $[\xi_{ir},\xi_{ir+1})$ and $[\eta_{js},\eta_{js+1})$ are non-zero intervals.

Analogously, the local form of bivariate NURBS function also can be obtained. It is defined by

$$P(\xi,\eta)=\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{B_{i,k}(\xi)B_{j,l}(\eta)w_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{m}B_{i,k}(\xi)B_{j,l}(\eta)w_{ij}}p_{ij}=\sum_{r=0}^{k}\sum_{s=0}^{l}N_{rs}(\xi,\eta)p_{rs} \tag{8}$$

where the $N_{rs}(\xi,\eta)$ is expressed as

$$N_{rs}(\xi,\eta)=\begin{cases}\dfrac{B^{(r+1)}_{i_r-r,k}(\xi)B^{(s+1)}_{j_s-s,l}(\eta)w_{(i_r-r)(j_s-s)}}{\sum_{r=0}^{k}\sum_{s=0}^{l}B^{(r+1)}_{i_r-r,k}(\xi)B^{(s+1)}_{j_s-s,l}(\eta)w_{(i_r-r)(j_s-s)}}, & \xi\in[\xi_{i_r},\xi_{i_r+1}),\ \eta\in[\eta_{j_s},\eta_{j_s+1})\\ 0, & \text{otherwise}\end{cases} \tag{9}$$

when all of weights (i.e.,w_{ij}) are set to 1, Eq. (8) degenerates into a local bivariate B-spline function.

Note that there are two attracting features introduced in the local bivariate B-spline or NURBS functions. Firstly, to compute the value of an arbitrary point it no longer requires to look for all of control points and evaluate all of basis functions. Only those relates to the point are needed. Secondly, because of the local definition, there is no need to keep a fixed number in every row and column to discretize a geometric body surface. These make our implementation of B-spline approximation much more efficient and easy. For example, in Fig. 2(a), a number of elements inevitably concentrate on the area of the two poles of the sphere to discretize. Actually, most of them have no contribution to the accuracy of the numerical results. Their existence is merely restricted by the tensor product form of the B-spline functions. It is doubtless that these superfluous meshes will increase the cost for computation. Whereas, in Fig. 2(b), the number of elements located along circle direction can be modified in every layer using the local form. Moreover, local refinement operations can be implemented in each layer easily as that to the B-spline curve [4].

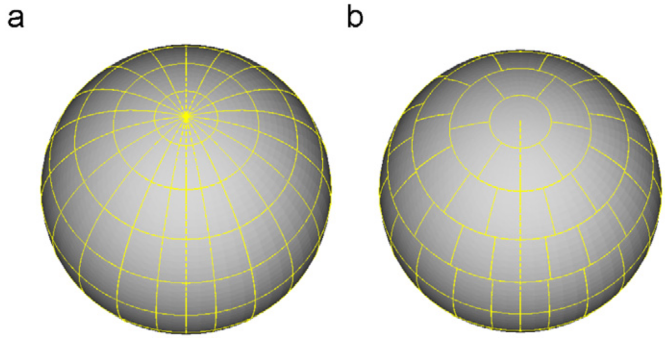

Fig. 2. Different nodes distribution patterns on the surface of a sphere for two different definitions of B-spline basis functions: (a) Nodes distribution using the global definition; (b) nodes distribution using the local definition.

Because Eqs. (7) and (9) do not satisfy the Kronecker delta property, inverse transformation is required to guarantee that B-spline approximation function interpolate the given boundary values on a body surface. The detailed process for the inverse transformation is omitted here, which can be found in the article [1,14].

If the unknown boundary values have been solved from the BIEs in the BFM framework, to obtain the components of the boundary values, the derivatives of B-spline approximation function are also required. These actually can be resorted to the derivation of the B-spline basis functions. There are two effective approaches that can be used to obtain their derivatives. One approach is using the derivation directly to the formula (2), so we can obtain the general recursive formulas, which are expressed as

$$\begin{cases}B'_{i,0}(\xi)=0,k=0\\ B'_{i,k}(\xi)=\frac{1}{\xi_{i+k}-\xi_i}B_{i,k-1}(\xi)+\frac{\xi-\xi_i}{\xi_{i+k}-\xi_i}B'_{i,k-1}(\xi)\\ \qquad -\frac{1}{\xi_{i+k+1}-\xi_{i+1}}B_{i+1,k-1}(\xi)+\frac{\xi_{i+k+1}-\xi}{\xi_{i+k+1}-\xi_{i+1}}B'_{i+1,k-1}(\xi),k\geq 1\end{cases} \tag{10}$$

Higher degree derivatives of B-spline basis functions can be obtained using the above recursive formulas. However, these derivatives are defined in a global interval, and called global B-spline basis function derivatives.

Alternative approach is direct derivation of the local B-spline basis functions. The derivatives of quadratic basis functions are written as

$$\begin{cases}B'^{(3)}_{i-2,2}(\xi)=\frac{2(\xi-\xi_{i+1})}{(\xi_{i+1}-\xi_{i-1})(\xi_{i+1}-\xi_i)}\\ B'^{(2)}_{i-1,2}(\xi)=\frac{(\xi_{i+1}-2\xi+\xi_{i-1})}{(\xi_{i+1}-\xi_{i-1})(\xi_{i+1}-\xi_i)}+\frac{(\xi_{i+2}-2\xi+\xi_i)}{(\xi_{i+2}-\xi_i)(\xi_{i+1}-\xi_i)}\\ B'^{(1)}_{i,2}(\xi)=\frac{2(\xi-\xi_i)}{(\xi_{i+2}-\xi_i)(\xi_{i+1}-\xi_i)}\end{cases} \tag{11}$$

The above piecewise polynomials are defined in a non-zero subinterval $[\xi_i,\xi_{i+1})$, and called local B-spline basis function derivatives. Where$B'^{(l)}_{i-j,k}(\xi)$ is the lth segment of the (i–j)th derivatives of B-spline basis functions, k is the degree of the basis functions and $l=1,2,\ldots,k+1$. Local form of higher degree derivatives also can be obtained in this manner, here we omit them.

Then, the partial derivative of local bivariate B-spline function can be evaluated by the derivation of the local B-spline basis functions. Its derivative with respect to ξ is given by

$$P_\xi(\xi,\eta)=\sum_{i=1}^{n}\sum_{j=1}^{m}B'_{i,k}(\xi)B_{j,l}(\eta)p_{ij}=\sum_{r=0}^{k}\sum_{s=0}^{l}N'_{rs}(\xi,\eta)p_{rs} \tag{12}$$

where the derivative of $N_{rs}(\xi,\eta)$ with respect to ξ is expressed as

$$N'_{rs}(\xi,\eta)=\begin{cases} B\prime^{(r+1)}_{i_r-r,k}(\xi)B^{(s+1)}_{j_s-s,l}(\eta), & \xi\in[\xi_{i_r},\ \xi_{i_r+1}),\ \eta\in[\eta_{j_s},\ \eta_{j_s+1}) \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where $p_{rs}=p_{(ir-r)(js-s)}$, $[\xi_{ir},\xi_{ir+1})$ and $[\eta_{js},\eta_{js+1})$ are non-zero subintervals.

The partial derivative of bivariate NURBS function with respect to ξ also can be obtained, which is given by

$$\begin{aligned} P_\xi(\xi,\eta) &= \frac{\partial\left(\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n}\frac{B_{i,k}(\xi)B_{j,l}(\eta)w_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n}B_{i,k}(\xi)B_{j,l}(\eta)w_{ij}}p_{ij}\right)}{\partial\xi} \\ &= \sum_{r=0}^{k}\sum_{s=0}^{l}N'_{rs}(\xi,\eta)p_{rs} \end{aligned} \tag{14}$$

where the derivative of $N_{rs}(\xi,\eta)$ with respect to ξ is given by

$$N'_{rs}(\xi,\eta)=\begin{cases} \frac{B\prime^{(r+1)}_{i_r-r,k}(\xi)B^{(s+1)}_{j_s-s,l}(\eta)w_{(i_r-r)(j_s-s)}}{D(\xi,\eta)}-\frac{B^{(r+1)}_{i_r-r,k}(\xi)B^{(s+1)}_{j_s-s,l}(\eta)w_{(i_r-r)(j_s-s)}D_\xi(\xi,\eta)}{(D(\xi,\eta))^2}, \\ \xi\in[\xi_{i_r},\xi_{i_r+1}),\eta\in[\eta_{j_s},\eta_{j_s+1}) \\ \qquad 0, \quad \text{otherwise} \end{cases} \tag{15}$$

$$D(\xi,\eta)=\sum_{r=0}^{k}\sum_{s=0}^{l}B^{(r+1)}_{i_r-r,k}(\xi)B^{(s+1)}_{j_s-s,l}(\eta)w_{(i_r-r)(j_s-s)} \tag{16}$$

$$D_\xi(\xi,\eta)=\sum_{r=0}^{k}\sum_{s=0}^{l}B\prime^{(r+1)}_{i_r-r,k}(\xi)B^{(s+1)}_{j_s-s,l}(\eta)w_{(i_r-r)(j_s-s)} \tag{17}$$

The partial derivative of NURBS shape function with respect to η can also be obtained similarly. Its expression is symmetrical with the above.

3. Discretization of the BIE for linear elasticity

The original linear elasticity in terms of partial differential equations can be transformed into integral equations over the boundary using Kelvin's fundamental solutions and Betti's reciprocity theorem (for more details see [15]). The 3D regularized displacement BIEs for linear elasticity can be written as

$$0=\int_\Gamma t_j(s)U_{ij}(s,y)d\Gamma(s)-\int_\Gamma(u_j(s)-u_j(y))T_{ij}(s,y)d\Gamma(s) \tag{18}$$

where $u_j(s)$ and $t_j(s)$ are the displacement and stress function on the boundary Γ, respectively, which can be approximated by the B-spline approximation functions in this paper. Where s is the field point (integration point), and y is the source point (collocation point). The kernels $U_{ij}(s,y)$ and $T_{ij}(s,y)$ are Kelvin's fundamental solutions, which are given as

$$U_{ij}(s,y)=\frac{1}{16\pi G(1-\mu)r}[(3-4\mu)\delta_{ij}+r_{,i}r_{,j}] \tag{19}$$

$$\begin{aligned} T_{ij}(s,y)=\frac{-1}{8\pi(1-\mu)r^2}\Big\{&\frac{\partial r}{\partial n(s)}[(1-2\mu)\delta_{ij}+3r_{,i}r_{,j}] \\ &-(1-2\mu)(r_{,i}n_j(s)-r_{,j}n_i(s))\Big\} \end{aligned} \tag{20}$$

In our schema, both the geometric faces to represent the integral boundary Γ of a body and the B-spline function to approximate the boundary values: $u_j(s)$ and $t_j(s)$, are expressed in parametric form. Therefore, the discretization of Eq. (18) can be performed in the parametric domain. Fig. 3 shows a parametric mesh, which maps to a 3D mesh model of a boundary face. In it, the red dots denote the locations of nodes; the purple mesh-grids are integral background elements, which are one-to-one corresponding to the nodes; the green rectangular is a parametric domain of an integral boundary element, for example the labeled domain of an integral element Γ_{up}; the black mesh-girds is parametric subdivision according to the knot vectors of quadratic beverage B-spline: $\Xi_\xi=[0,0,0,1/3,2/3,1,1,1]$ and $\Xi_\eta=[0,0,0,1/2,1,1,1]$; the virtual values, considered as the control points of B-spline function, are located on the corner points of each black grids, denoted by $\hat{p}_{ij}$; the given Dirichlet boundary condition is imposed on the red dots, denoted by $\tilde{p}_{ij}$.

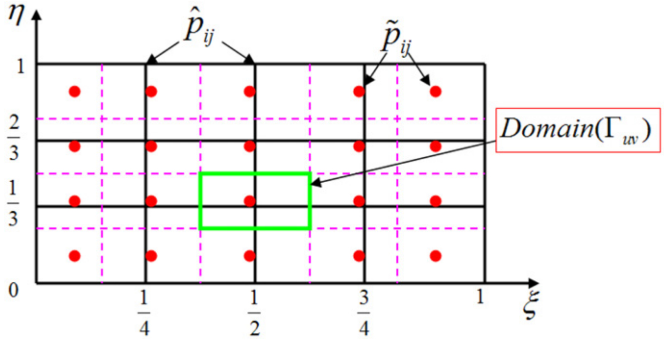

Fig. 3. The parametric mesh of an integral boundary face. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

To make use of the mesh fashion described as the Fig. 3, Eq. (18) can be discreted as

$$0=\sum_{uv}\int_{\Gamma_{uv}}t_j(s)U_{ij}(s,y)d\Gamma(s)-\sum_{uv}\int_{\Gamma_{uv}}(u_j(s)-u_j(y))T_{ij}(s,y)d\Gamma(s) \tag{21}$$

Subsequently, the bivariate B-spline approximation function can also be constructed with the $\hat{p}_{ij}$ as control points, which is expressed as

$$P(\xi,\eta)=\sum_{r=0}^{k}\sum_{s=0}^{l}N_{rs}(\xi,\eta)p_{rs} \tag{22}$$

where $p_{rs}=\hat{p}_{(i_r-r)(j_s-s)}$, $i_r\in\{k+1,\dots,n\}$ and $j_s\in\{l+1,\dots,m\}$. In Fig. 3, $n=5$, $m=4$ and $k=l=2$. Eq. (22) can also be expanded and then expressed as another form, such as

$$\begin{aligned} P(\xi,\eta) &= \sum_{r=0}^{k}\sum_{s=0}^{l}N_{rs}(\xi,\eta)p_{rs} \\ &= \sum_{r=0}^{k}N_{r0}(\xi,\eta)p_{r0}+\sum_{r=1}^{k}N_{r1}(\xi,\eta)p_{r1}+\ldots+\sum_{r=l}^{k}N_{rl}(\xi,\eta)p_{rl} \\ &= \sum_i N_i(\xi,\eta)\hat{p}_i \end{aligned} \tag{23}$$

where $N_i(\xi,\eta)$ and $\hat{p}_i$ are the elements of two vectors corresponding to $N_{rs}(\xi,\eta)$ and p_{rs}, respectively. In Fig. 3, $N_{r0}(\xi,\eta)=N_{r1}(\xi,\eta)=\ldots=N_{rl}(\xi,\eta)$. If the number of elements in every rows is different from each other, we have $N_{r0}(\xi,\eta)\neq N_{r1}(\xi,\eta)\neq\ldots\neq N_{rl}(\xi,\eta)$. $N_{rs}(\xi,\eta)$ does not satisfy the Kronecker delta property, but it is constricted to interpolate the node values $\tilde{p}_{ij}$, i.e.,

$$\tilde{p}_j=P(\xi_j,\eta_j)=\sum_i N_i(\xi_j,\eta_j)\hat{p}_i \tag{24}$$

where $\tilde{p}_j$ is an arbitrary vector element corresponding to the node values $\tilde{p}_{ij}$.

Substituting Eq. (24) into Eq. (23) with inverse transformation, we have

$$\begin{aligned} P(\xi,\eta) &= \sum_i N_i(\xi,\eta)\sum_j([N]^{-1}_{ij}\tilde{p}_j)=\sum_i\sum_j N_i(\xi,\eta)[N]^{-1}_{ij}\tilde{p}_j \\ &= \sum_j\tilde{N}_j(\xi,\eta)\tilde{p}_j \end{aligned} \tag{25}$$

where $\tilde{N}_j(\xi,\eta)=\sum_i N_i(\xi,\eta)[N]^{-1}_{ij}$.

The boundary displacement and stress approximation functions can be respectively built, such as

$$u_j(\xi,\eta)=\sum_i \tilde{N}_i(\xi,\eta)\tilde{u}_j(\xi_i,\eta_i) \tag{26}$$

$$t_j(\xi,\eta)=\sum_i \tilde{N}_i(\xi,\eta)\tilde{t}_j(\xi_i,\eta_i) \tag{27}$$

Making the following substitution,

$$I_{uv}^{(1)}=\int_{\Gamma_{uv}}(u_j(s)-u_j(y))T_{ij}(s,y)d\Gamma(s) \tag{28}$$

$$I_{uv}^{(2)}=\int_{\Gamma_{uv}}t_j(s)U_{ij}(s,y)d\Gamma(s) \tag{29}$$

and substituting Eq. (26) and Eq. (27) into them, we have

$$I_{uv}^{(1)}=\sum_l\left(\int_{-1}^{1}\int_{-1}^{1}(\tilde{N}_j(\xi,\eta)-\tilde{N}_j(\xi_0,\eta_0))T_{ij}(\xi,\eta)J(\xi,\eta)d\xi d\eta\right)(\tilde{u}_j)_l \tag{30}$$

$$I_{uv}^{(2)}=\sum_l\left(\int_{-1}^{1}\int_{-1}^{1}\tilde{N}_{rs}(\xi,\eta)U_{ij}(s,y)J(\xi,\eta)d\xi d\eta\right)(\tilde{t}_j)_l \tag{31}$$

where $(\tilde{u}_j)_l=\tilde{u}_j(\xi_l,\eta_l)$ and $(\tilde{t}_j)_l=\tilde{t}_j(\xi_l,\eta_l)$.

Then Eq. (21) can be described as a matrix form

$$H\tilde{\mathbf{u}}-G\tilde{\mathbf{t}}=\mathbf{0} \tag{32}$$

Solving the above equation, we can obtain the unknown displacement and stress on the boundary. To further compute the boundary stress components, the following equations are used:

$$\begin{cases}\sigma_{ij}n_j=t_i\\ \frac{\partial u_i}{\partial x_j}\frac{\partial x_j}{\partial \xi_k}=\frac{\partial u_i}{\partial \xi_k}(i,j=1,2,3\quad k=1,2)\\ \sigma_{ij}=G(u_{i,j}+u_{j,i})+\lambda u_{l,l}\delta_{ij}\end{cases} \tag{33}$$

This relationship can be founded from geometric information of a body and linear elasticity theory.

If the boundary values have been solved from Eq. (32), the solution of domain stress components needs the following domain stress integral equation, which is written as

$$\sigma_{ij}(y)=\int_\Gamma t_k(s)U_{ijk}(s,y)d\Gamma(s)-\int_\Gamma u_k(s)T_{ijk}(s,y)d\Gamma(s) \tag{34}$$

with the fundamental solution described as

$$U_{ijk}=\frac{1}{8\pi(1-\mu)r^2}\left[(1-2\mu)(\delta_{ik}r_{,j}+\delta_{jk}r_{,i}-r_{,k}\delta_{ij})+3r_{,i}r_{,j}r_{,k}\right] \tag{35}$$

$$\begin{aligned}T_{ijk}=\frac{G}{4\pi(1-\mu)r^3}\Big\{&3\frac{\partial r}{\partial n(s)}\left[(1-2\mu)\delta_{ij}r_{,k}+\mu(\delta_{ik}r_{,j}+\delta_{jk}r_{,i})-5r_{,i}r_{,j}r_{,k}\right]\\&-(1-4\mu)\delta_{ij}n_k(s)+(1-2\mu)[\delta_{ik}n_j(s)+\delta_{jk}n_i(s)+3r_{,i}r_{,j}n_k(s)]\\&+3\mu(r_{,j}r_{,k}n_i(s)+r_{,i}r_{,k}n_j(s))\Big\}\end{aligned} \tag{36}$$

4. Numerical examples

Two 3D exact geometric models for linear elasticity problems are employed in the BFM to illustrate the efficiency and accuracy of B-spline approximation. As a tool to discretize the surfaces of the geometric bodies, nodes described here have two different meanings. One is that these nodes are only a series of discrete points, which locate in the inner of rectangular grids. So B-spline approximation functions can be constructed by the prescribed boundary values on these nodes. Another is that these nodes are a series of background elements. To solve the BIEs, information for Gauss quadrature can be obtained from these nodes, which have one-to-one correspondence to the background elements. These background elements are expressed as the rectangular grids due to the form of tensor product of the bivariate B-spline functions. Comparison between the B-spline approximation and the MLS approximation is performed in the same framework of BFM. The boundary geometric information used for solving the BIEs, such as the Jacobian, the out normal and the distance between two points, can be exactly obtained from the exact geometry of a body based on the conception of BFM.

In order to estimate numerical error and convergence, a 'global' L_2 norm error, normalized by $|v_{\max}|$ is defined by [16]

$$e=\frac{1}{|v|_{\max}}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(v_i^{(e)}-v_i^{(n)})^2} \tag{37}$$

where $|v_{\max}|$ is the maximum value of sample points, the superscripts (e) and (n) refer to the exact and numerical solutions, respectively.

4.1. The linear elasticity problem

To assess the accuracy of the current approximation method, we use the following two analytical fields:

(i) Linear solution:

$$\begin{aligned}u_x&=x+0.5y+0.5z\\ u_y&=0.5x+y+0.5z\\ u_z&=0.5x+0.5y+z\end{aligned} \tag{38}$$

(ii) Quadratic solution:

$$\begin{aligned}u_x&=-2x^2+3y^2+3z^2\\ u_y&=3x^2-2y^2+3z^2\\ u_z&=3x^2+3y^2-2z^2\end{aligned} \tag{39}$$

In all cases, the 3D regularized displacement BIEs for linear elasticity is solved, combined with reasonable prescribed boundary conditions corresponding to the above analytical solutions.

4.1.1. Linear elasticity problem for a sphere

A sphere is firstly used for discussion, with radius 2 unit and center at the origin. The usual spherical polar coordinates θ and φ are used. The linear and quadratic analytical fields (Eqs. (38) and (39)) are used as exact solutions. Displacement boundary condition is employed in the discussion. Comparison among linear, quadratic and cubic B-spline approximation, as well as MLS will be carried out on the sphere. Displacement boundary conditions corresponding to the exact solutions (Eqs. (38) and (39)) are imposed on the surface of the sphere. Two mesh models of the sphere are described in the Fig. 3. The left one and the right one in Fig. 4 are meshed according to the B-spline local definition and global definition, respectively.

Two kinds of comparison will be performed on the two models to illustrate the B-spline approximation. In the first one, the two

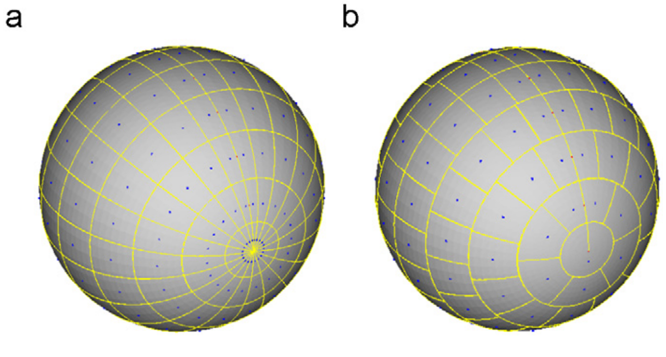

Fig. 4. Two different mesh fashion of a sphere. (a) Discretization according to B-spline local definition; (b) discretization according to B-spline global definition.

types of sphere mesh models are adopted to testify their computational efficiency. Mesh displayed in Fig. 4(b) indicates that the elements (also called nodes) are located along woofs, the number of elements can be modified according to the woofs length. So the element size in the full surface can be modified near to a uniform size. Whereas, mesh displayed in Fig. 4(a) indicates that the number of nodes along each woof keeps a fixed quantity according to B-spline basis global definition. So elements are dense in the areas of two poles of the sphere. In the two models, we distribute the same number of elements (40 elements) on the two sides of equators. 200 nodes are used for subdividing the full surface in Fig. 4(a), but only 118 nodes are used in Fig. 4(b). The L_2 errors of displacement and stress components evaluated by Eq. (37) and time required for numerical results for quadratic analytical fields are described in Table 1. The quadratic and cubic B-splines are employed to approximate the boundary values of displacement and stress. Each group of the data in Table 1(a) and (b) shows that less nodes are required for computation, but more accurate results can be obtained with the local B-spline, compared with the global B-spline. Furthermore, the less time cost is need for obtain high accuracy numerical result. That is because not only the less nodes are used for analysis, but also much more compute time spent for estimation of the singular integral of BIEs is saved due to the local B-spline definition effectively avoid the nodes largely concentrate in two poles of the sphere where many singular integrals need to be computed.

The second comparison is performed between the MLS and the B-spline in the same BFM framework. In this case, the displacement boundary condition corresponding to the linear and quadratic analytical field (Eqs. (38) and (39)) are imposed on the surface of the sphere. The linear, quadratic and cubic local B-spline, as well as the MLS are used for approximating the surface components of displacement u and stress t. The L_2 errors in terms of these boundary values approximated by quadratic B-spline and MLS are shown in Table 2, which are obtained from four sets of nodes: (a) 48 nodes; (b) 80 nodes; (c) 124 nodes; (d) 283 nodes, associated with quadratic displacement boundary condition corresponding to quadratic analytical field Eq. (39). According to the Table 2, we can clearly see that the quadratic B-spline outperforms than the quadratic MLS, not only in exactness but also in computational time.

In Figs. 5–7, six sets of nodes: (a) 22 nodes; (b) 48 nodes; (c) 64 nodes; (d) 80 nodes; (e) 124 nodes; (f) 283 nodes, associated with quadratic displacement boundary condition corresponding to quadratic analytical field Eq. (39) are used to obtain the numerical results. The boundary stress components are approximated by the four methods: the linear, quadratic and cubic B-spline, and the MLS. From the three figures, we can find that quadratic and cubic local B-splines have higher rates of convergence over MLS. Although the linear B-spline is less efficient than MLS in this

Table 2
The L_2 errors of the surface components of displacement and stress obtained from quadratic MLS and B-spline approximation (M: MLS, B2: quadratic B-spline), which are signed with percentage (%).

Number of nodes		u_{xx}	u_{yy}	u_{zz}	S_{xx}	S_{yy}	S_{zz}	Time (s)
48	M	4.594	5.502	3.007	1.935	2.023	2.746	24
	B2	1.454	1.588	1.752	1.512	1.434	1.694	10
80	M	1.737	1.938	0.847	0.8897	0.9412	1.267	75
	B2	0.5055	0.1008	0.4729	0.5218	0.4817	0.5812	41
124	M	0.8548	0.9225	0.3046	0.4582	0.4935	0.802	228
	B2	0.2109	0.07857	0.2042	0.2158	0.2148	0.2591	94
283	M	0.2367	0.2555	0.04826	0.1373	0.1535	0.3691	1825
	B2	0.03675	0.02347	0.04427	0.05364	0.06814	0.07372	747

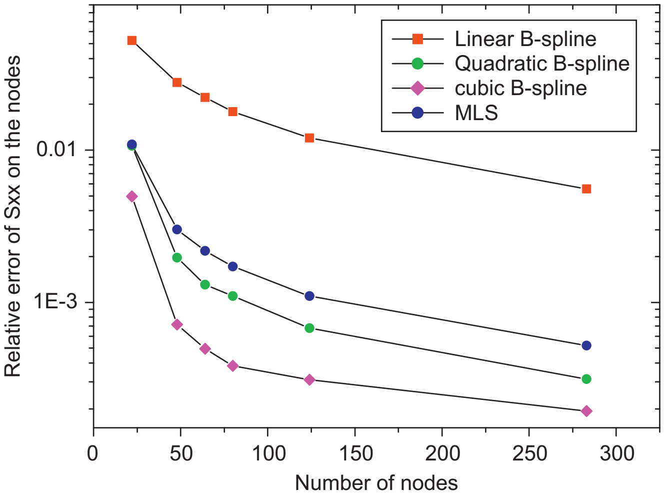

Fig. 5. Relative error of stress component S_{xx} and convergence of the four methods.

Table 1
The L_2 errors of components of displacement and stress on the sphere surface obtained from two different kinds of meshes (L: local B-spline, G: global B-spline), which are signed with percentage (%).

Number of nodes		u_x	u_y	u_z	S_x	S_y	S_z	Time (s)
(a) Quadratic B-spline approximation for quadratic displacement boundary condition								
L	64	0.897	0.3483	0.8665	0.8063	0.7795	0.9657	20
G	72	1.696	0.5649	1.752	1.366	1.305	1.569	128
L	80	0.5055	0.1008	0.4729	0.5281	0.4817	0.5812	41
G	98	0.8009	0.2693	0.8665	0.8056	0.7702	1.006	152
L	101	0.3054	0.04275	0.2994	0.3045	0.3034	0.3713	56
G	128	0.4518	0.1433	0.4729	0.5152	0.4984	0.6092	296
L	124	0.2109	0.07857	0.2042	0.2158	0.2148	0.2591	94
G	162	0.2905	0.08319	0.2994	0.3147	0.3068	0.3827	559
(b) Cubic B-spline approximation for quadratic displacement boundary condition								
L	64	1.22	1.443	1.356	0.6248	0.6403	0.9339	22
G	72	2.88	0.3547	2.886	2.198	2.063	2.24	97
L	80	0.2982	0.3287	0.257	0.1789	0.1653	0.2082	50
G	98	1.359	0.1504	1.356	0.8843	0.8328	0.9693	156
L	101	0.1213	0.06067	0.1225	0.1513	0.1654	0.1518	122
G	128	0.271	0.07346	0.257	0.1571	0.1485	0.1667	359
L	124	0.08022	0.02466	0.1007	0.08135	0.09845	0.08793	194
G	162	0.13	0.03994	0.1225	0.1761	0.1672	0.2007	564

example, B-spline has a flexible selection of high degree, which makes the B-spline more freely and efficiently used in approximation problems than the MLS.

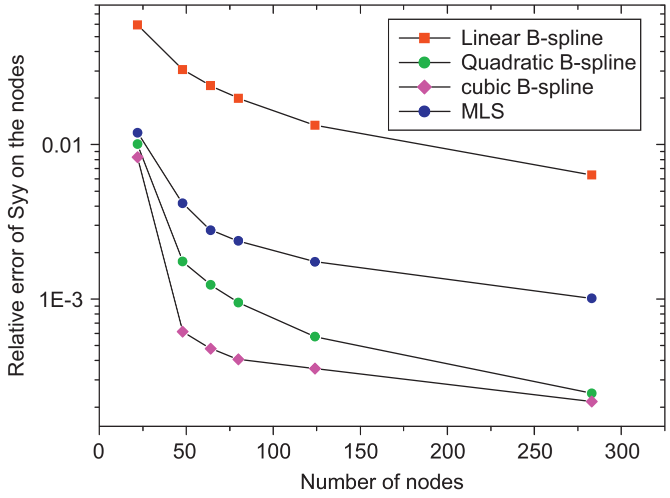

Fig. 6. Relative error of stress component S_{yy} and convergence of the four methods.

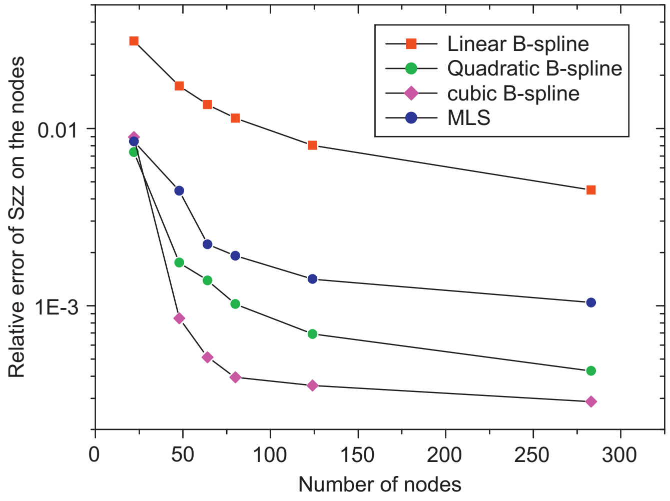

Fig. 7. Relative error of stress component S_{zz} and convergence of the four methods.

4.1.2. *Linear elastic problem for an elbow pipe*

Here, we use a more complicated body to determinate the convergence rates of B-spline approximation in the BFM. Its geometry and main size are described in Fig. 8. The body is closed by six different torus surface parts and four cylinder surfaces. Quadratic displacement boundary condition corresponding to the exact solution (Eq. (39)) is imposed on all of surfaces. There are three sets of nodes: (a) 462 nodes; (b) 652 nodes; (c) 856 nodes, to be used for discretizing this body surface. These mesh models are presented in Fig. 9. B-spline approximation functions for each face can be constructed by a free selection of degrees in two directions respectively. For example the No. 1 torus surface in Fig. 8, we use cubic basis function for approximation in the circumference direction, whereas use linear or quadratic basis function in another direction.

In order to determine the convergence rates, the stress components for three surfaces are considered, which are respectively labeled by three different colored dashes in the Fig. 8(b). The No. 1 surface is of small size when compared with the whole pipe, and especially, the two ends of the pipe are thin. To perform the boundary integral on it, there are many nearly singular integrals needed for treatment, which may deteriorate the final numerical results. But Fig. 10 shows that the numerical results restricted to the quadratic boundary conditions on the No. 1 slightness surface are convergent to the exact solutions following the increment of nodes, even if the results initially obtained from the 462 nodes fluctuate widely. Figs. 11 and 12 indicate that the numerical results obtained from the three sets of nodes are in good agreement with the quadratic stress exact solutions. Comparing with the Fig. 10, we can find the two surface size is bigger than that of the No. 1 surface. So this example fully proves that B-spline has well stability to approximate boundary values in BIEs, not only for a large size surface, but also for a small feature.

5. Conclusions

The bivariate B-spline function as an efficient approximation method has been successfully implemented in the BFM for solving the linear elasticity problem. The new implementation inherits the advantages of the BFM. For example, the geometric model of a body is directly used for analysis, thus no geometric error introduced. To alleviate the difficulties in meshing associated with the tensor product form of the B-spline bivariate functions, the traditional global form of B-spline basis functions are converted into local form. As the B-spline bivariate functions

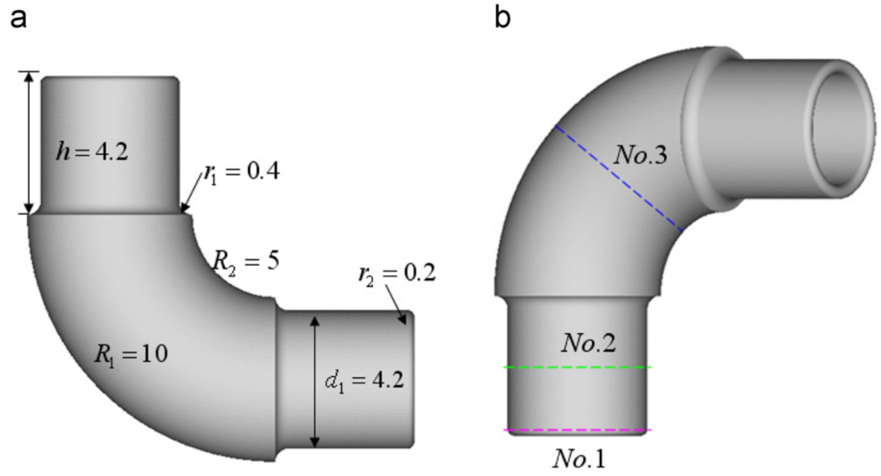

Fig. 8. An elbow pipe and its main size.

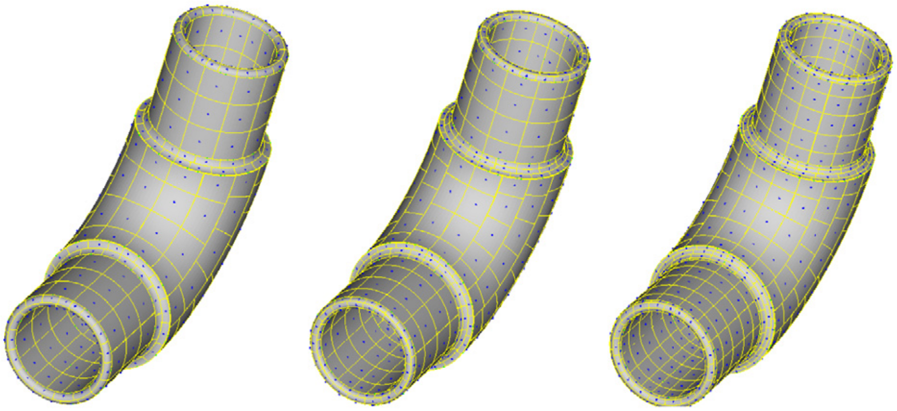

Fig. 9. Three different mesh models for the elbow pipe.

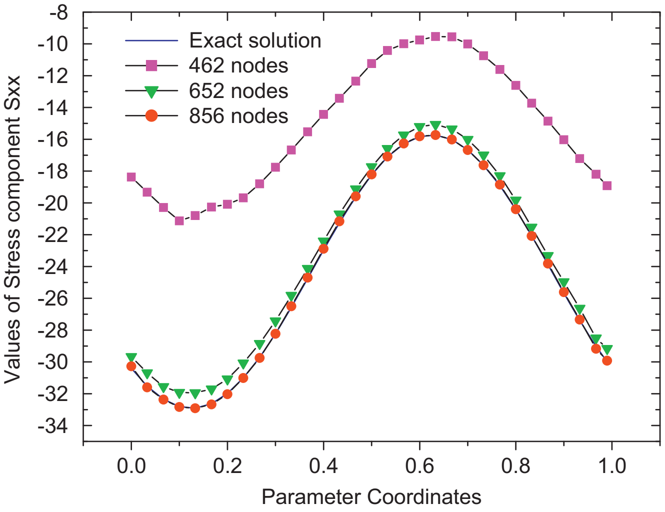

Fig. 10. The stress component S_{xx} along circle direction on the No. 1 surface.

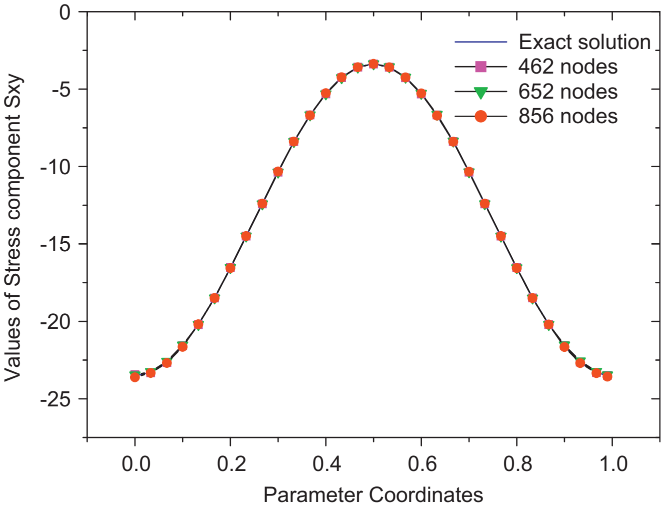

Fig. 11. The Stress component S_{xy} along circle direction on the No. 2 surface.

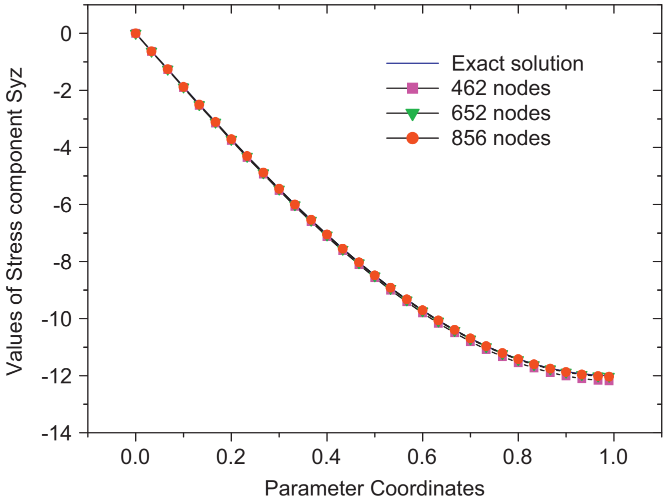

Fig. 12. The Stress component S_{yz} along circle direction on the No. 3 surface.

are fitting type functions, i.e. they lack the Kronecker delta property, an inverse transformation is performed to convert them into ones of interpolation type.

Numerical results have demonstrated that our implementation is feasible for solution of BIEs. As a comparison, the MLS approximation, which is widely applied in meshless analysis, is also implemented into the same framework of BFM. Comparisons between the B-spline bivariate function and the MLS approximation regarding to accuracy, stability and efficiency have been performed using examples of linear elasticity problems. Results show that our method performs better in all mentioned aspects.

Despite the advantages, however, the bivariate B-spline functions suffer from the defection of tensor product, which limits their application in engineering. T-spline is a new promising method [17], which is also based on the B-spline basis functions. Use of T-spline for analysis instead of B-spline is an ongoing work.

To deal with the large-scale computations for complicated geometric bodies, the Fast Multipole Method (FMM) [18–20] can be applied to reduce the computation expense. And this is also planned.

Acknowledgments

This work was supported in part by National Science Foundation of China under grant number 10972074 and in part by National 973 Project of China under grant number 2010CB328005.

References

[1] Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Computer Methods in Applied Mechanics and Engineering 2005;194:4135–95.

[2] Zhang JM, Qin XY, Han X, Li. GY. A boundary face method for potential problems in three dimensions. International Journal of Numerical Methods in Engineering 2009;80:320–37.

[3] de Boor Carl. On calculating with B-spline. Journal of Approximation Theory 1972;6:50–62.

[4] Ramshaw Lyle. Blossoms are polar forms. Computer Aided Geometric Design 1989;6:323–58.

[5] Cottrell J, Hughes T, Reali A. Studies of refinement and continuity in isogeometric analysis. Computer Methods in Applied Mechanics and Engineering 2007;196:4160–83.

[6] Bazilevs Y, de Veiga LB, Cottrell J, Hughes T, Sangalli G. Isogeometric analysis: approximation, stability and error estimates for refined meshes. Mathematical Models and Methods in Applied Sciences 2006;6:1031–90.

[7] Wall WA, Frenzel MA, Cyron C. Isogeometric structural shape optimization. Computer Methods in Applied Mechanics and Engineering. 2008;197:2976–88.

[8] Prenter PM. Splines and Variational Methods. New York: Wiley; 1975.

[9] Hollig K. Finite Element Methods with B-Splines, Frontiers in Applied Mathematics, vol. 26. Philadelphia: SIAM; 2003.

[10] Sabin MA. Spline Finite Elements, PHd Dissertation. U.K: Cambridge University; 1997.

[11] Kagan P, Fischer A, Bar-Yoseph PZ. New B-spline finite element approach for geometrical design and mechanical analysis. International Journal for Numerical Methods in Engineering 1998;41:435–58.

[12] Kagan P, Fischer A. Integrated mechanically based CAE system using B-spline finite elements. Computer Aided Design 2000;32:539–52.

[13] Zhang JM, Yao ZH. Analysis of 2-D thin structures by the meshless regular hybrid boundary node method. Acta Mechanica Sinica 2002;15:36–44.

[14] Shaw Amit, Roy D. NURBS-based parametric mesh-free methods. International Journal of Numerical Methods in Engineering 2008;197:1541–67.

[15] Brebbia CA, Telles JCF, Wrobel LC. Boundary Element Techniques-Theory and Applications in Engineering. Berlin: Springer; 1984.

[16] Zhang Jianming, Tanaka Masataka, Matsumoto Toshiro. Meshless analysis of potential problems in three dimensions with the hybrid boundary node method. International Journal of Numerical Methods in Engineering 2004:1147–66.

[17] Sederberg TW, Zheng J, Bakenov A, Nasri A. T-splines and T-NURCCSs. ACM Transactions on Graphics 2003;22:477–84.

[18] Zhang JM, Tanaka M, Endo M. The hybrid boundary node method accelerated by fast multipole method for 3D potential problems. International Journal for Numerical Methods in Engineering 2005;63:660–80.

[19] Zhang JM, Tanaka M. Fast HdBNM for large-scale thermal analysis of CNT-reinforced composites. Computational Mechanics 2008;41:777–87.

[20] Zhang JM, Tanaka M. Adaptive spatial decomposition in fast multipole method. Journal of Computational Physics 2007;226:17–28.