23rd International Congress on Modelling and Simulation, Canberra, ACT, Australia, 1 to 6 December 2019
mssanz.org.au/modsim2019

T-splines and Bézier extraction in modeling shape of boundary in Parametric Integral Equation System for 3D problems of linear elasticity

Krzysztof Szerszeń and Eugeniusz Zieniuk

Institute of Informatics, University of Bialystok
Konstantego Ciołkowskiego 1M, 15-245
Bialystok, Poland
Email: ezieniuk@ii.uwb.edu.pl, kszerszen@ii.uwb.edu.pl

Abstract: T-splines are a generalization of NURBS surfaces. They retain all the properties of NURBS, but are increasingly considered as a better alternative. While the control points in NURBS lie in a rectangular grid of rows and columns, the control points in the T-splines may be incomplete. Moreover, T-splines allow for local insertion of new control points. The possibility of adding control points locally has many advantages. For the same geometric representation, T-splines require a much smaller number of control points than NURBS. Moreover, T-splines naturally eliminate the formation of gaps when connecting adjacent surfaces, which is a disadvantage of NURBS. Due to these advantages, T-splines have received increasing research attention in recent years.

In the present study, we will use T-splines in combination with the Parametric Integral Equation System (PIES) to solve 3D linear elasticity problems. PIES is a computational method for finding solutions to 2D and 3D boundary value problems and is based on the formal separation of the approximation of the boundary geometry from the boundary functions. The PIES formulation is independent from the physical representation of the boundary and, for any shape, it is always defined in the parametric domain and not on the physical boundary as in the traditional Boundary Integral Equations (BIE). As a result, the boundary can be described analytically in the form of mathematical functions directly included in the mathematical formula of PIES. In the previous studies, the 2D boundary could be defined by parametric curves such as Bézier, B-spline and Hermitte, whereas in the 3D case, we can apply parametric surfaces. These were flat Coons as well as smooth Bézier surfaces. The latter have particularly positive properties with respect to their flexibility and simplicity of implementation. This correlates with the fact that parametric curves and surfaces are widespread in CAD systems, e.g. NURBS. They have been intensively studied and modified into more effective variants such as hierarchical B-splines, PB-splines (Point based splines), T-splines, LR-splines (Locally refined splines) and PHT-splines (Polynomial splines over hierarchical T-meshes).

The aim of the paper is to extend the application of PIES to objects with boundaries represented by T-splines. In order to simplify the problem, we use a Bézier extraction mechanism for automatic conversion of T-splines surfaces into standard Bézier surfaces, previously used in PIES. The Bézier extraction transforms the T-splines basis functions into linear combination of standard Bernstein basis functions. Therefore, if we can transform the T-splines basis functions into Bernstein polynomials, this allows the T-spline surface to be replaced by a set of Bézier patches with the traditional parametric domain for each Bézier patch.

The reliability of the proposed approach is evaluated through numerical tests based on linear elastic problems. We show how to represent the boundary by T-splines and compare the accuracy of the results with known analytical solutions. We use a commercial T-spline plugin for Rhino from Autodesk to model and manipulate T-spline surfaces. The plugin also includes the Bézier extraction module that directly converts T-spline models to the appropriate ones constructed with Bézier surfaces. The presented examples confirm the applicability of the approach with high accuracy of the solutions.

Keywords: *Parametric Integral Equation System (PIES), T-splines, Bézier extraction, 3D linear elasticity*

1. INTRODUCTION

When solving a boundary value problem, we need to specify the shape of the boundary and the boundary conditions for the governing differential (integral) equation. Classical numerical methods subdivide the whole domain (boundary) into finite (boundary) elements. This practice, taking the form of Finite Element Method (FEM) and Boundary Element Method (BEM), has developed significantly during recent decades and has already exhausted its development potential.

More recently, new technologies for effective description and modeling of geometric objects have emerged. We can also interpret the domain of the solved boundary problem as the geometric object and connect it with the solved differential (integral) equation. However, this connection is not easy to achieve, because the formalism of the classical differential equations (integral) is not adapted to new modeling techniques. This is also the reason for applying meshing techniques because such representation of the domain (boundary) is the simplest from the point of view of its connection with the solved differential (integral) equations.

Therefore, to adapt the classic Boundary Integral Equation (BIE) to modern modeling techniques, the authors had previously modified BIE and obtained Parametric Integral Equation System (PIES), which allowed to eliminate elements for modeling of the boundary. PIES eliminates the need for both boundary and domain discretizations and as a result allows for effective boundary shape representation by 2D and 3D computer visualization techniques, such as parametric curves or surfaces. This correlates with the fact that parametric curves and surfaces are widespread in CAD systems, e.g. NURBS (Piegl et al. 2012). They have been intensively studied which leads to the development of more effective variants, such as hierarchical B-splines (Forsey et al. 1988), PB-splines (Point based splines) (Sederberg et al. 2003), T-splines (Sederberg et al. 2003, 2004), LR-splines (Locally refined splines) (Tor et al. 2013) and PHT-splines (Polynomial splines over hierarchical T-meshes) (Deng et al. 2008). Moreover, PIES separates the boundary shape represented by the aforementioned curves or surfaces from the approximation of field values on the boundary, so the accuracy improvement is independent from the boundary description. Previous research about the practical application of boundary representation by parametric surfaces in PIES in the case of 3D problems are concerned mainly with the boundary value problems described for example by the Laplace (Zieniuk et al. 2014), Navier-Lamé (Zieniuk et al. 2018), and Stokes (Zieniuk et al. 2019) equations with the boundary described by Coons and Bézier surfaces.

The aim of the paper is to extend the application of PIES to the boundary value problems with boundaries represented by T-splines. T-splines set a new direction in CAD technology due to their flexibility in modeling with an irregular mesh of control points. Moreover, the number of control points for models generated by T-splines is definitely smaller in relation to the analogue described using NURBS. Therefore, introducing T-splines for boundary representation in PIES is reasonable. We use the CAD system Rhino for creating T-spline models and employ build-in Bézier extraction module to convert T-splines into Bézier surfaces. The approach is presented and verified on the basis of 3D problems of linear elasticity described by the Navier-Lamé differential equation.

2. T-SPLINES AND BÉZIER EXTRACTION

2.1. Comparison of NURBS and T-splines surfaces

In 2003, Sederberg and others (Sederberg et al. 2003, 2004) introduced T-splines as a generalization of NURBS technology, due to the limitations imposed by the NURBS tensor product (Farin 1999, Piegl et al. 2012). The concept of T-splines was preceded by the introduction of Point Based (PB) splines (Sederberg et al. 2003). PB-splines are the surfaces where their control points do not have a topological relationship with each other.

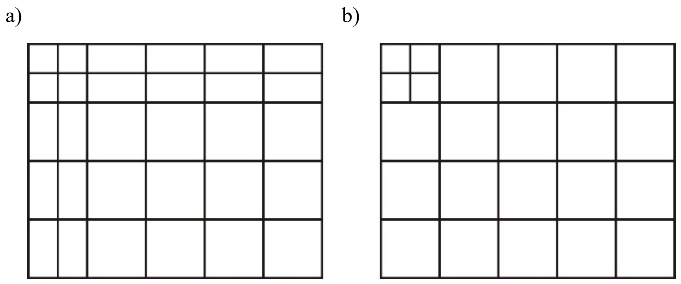

Figure 1. The topology of control points in NURBS (a) and T-splines (b) during refinement.

T-splines are the extension of PB-splines. While the control points in NURBS lie in a rectangular grid of rows and columns, the control points in the T-splines may be incomplete. In addition, T-splines allow for local insertion of new control points, as shown in Fig. 1.

Before we present T-splines, we give a brief description of standard NURBS surfaces for which T-splines are enhanced. The NURBS surface of degree 3 is given by the formula (Farin 1999)

$$\boldsymbol{P}(v,u)=\frac{\sum_{i=0}^{n}\sum_{j=0}^{m}\boldsymbol{P}_{ij}w_{ij}B_{v,i}^{3}(v)B_{u,j}^{3}(u)}{\sum_{i=0}^{n}\sum_{j=0}^{m}w_{ij}B_{v,i}^{3}(v)B_{u,j}^{3}(u)}, \tag{1}$$

where $B_{v,i}^{3}(v)$ and $B_{u,j}^{3}(u)$ are B-spline basis functions associated with the parameters v and u. A significant drawback of NURBS surfaces is the fact that their control points need to form a rectangular topological grid.

2.2. Advantages of T-splines surfaces

T-splines are a generalization of NURBS surfaces. They retain all the properties of NURBS, but are increasingly considered as the better alternative. Similar to NURBS, T-splines are the tensor product of the B-spline base functions, meaning that the T-spline control points influence the shape of the surface based on the B-spline basis functions. While they both are based on the same type of basis functions, the formula of T-splines is slightly different and is written as follows

$$\boldsymbol{P}(v,u)=\frac{\sum_{i=0}^{n}\boldsymbol{P}_{i}w_{i}N_{i}(v,u)}{\sum_{i=0}^{n}w_{i}N_{i}(v,u)}, \tag{2}$$

where $N_i(v,u)=B_{v,i}^{3}(v)B_{u,i}^{3}(u)$ are T-splines basis functions corresponding to control point $\boldsymbol{P}_i$. The main difference in the case of T-splines is that functions $B_{v,i}^{3}(v)$ and $B_{u,i}^{3}(u)$ are unique for each of the control points, while in the case of NURBS $B_{v,i}^{3}(v)$ are the same for all control points in a given column and $B_{u,i}^{3}(u)$ are the same for all control points in a given row. This gives great flexibility in the modeling of a T-splines surface with fewer control points compared to the analogous NURBS surface. In addition, the gaps generated as a result of the NURBS joint are categorically unacceptable for modeling the boundary surfaces during solving boundary value problems.

Since NURBS are a special case of T-splines, they are compatible with each other. Each NURBS is a T-spline, but on the other hand, each T-spline can be transformed into one or more NURBS surfaces by performing a local grid of control points to eliminate all T-junctions.

2.3. Bézier extraction

Bézier extraction consists of the representation of the T-splines basis functions as a linear combination of Bernstein polynomials. It was first proposed for NURBS by (Borden et al. 2011), and then for T-splines in (Scott et al. 2011). Therefore, if we can transform the T-splines basis functions into Bernstein polynomials, this allows the T-spline surface to be replaced by a set of Bézier patches with the traditional parametric domain. We can write it down as follows (Scott et al. 2011)

$$\boldsymbol{N}^{e}(\tilde{\xi})=\boldsymbol{C}^{e}\boldsymbol{B}(\tilde{\xi}), \tag{3}$$

where $\tilde{\xi}$ is the coordinate in the standard domain of an individual Bézier patch, $\boldsymbol{N}^{e}(\tilde{\xi})$ is a T-splines vector of the basis functions that are non-zero over the Bézier surface e, whereas $\boldsymbol{B}(\tilde{\xi})$ is a vector of the tensor product of the basis functions of Bernstein polynomials associated with the Bézier surface e, while $\boldsymbol{C}^{e}$ is the extraction operator. From a practical point of view, Bézier extraction consists of duplicating existing knots until they reach a multiple of the degree of the T-splines surface.

3. PARAMETRIC INTEGRAL EQUATION SYSTEM (PIES)

The previous section shows that it is possible to transform the T-splines surface into a set of corresponding Bézier patches. Due to this, we can apply PIES for the analysis of the boundary value problems with the boundary described by T-splines. The PIES formula in this case for the Navier-Lamé equation without body forces is as follows (Zieniuk et al. 2018)

$$0.5\boldsymbol{u}_l(\overline{v},\overline{w}) = \sum_{j=1}^{n} \int_{v_{j-1}}^{v_j} \int_{w_{j-1}}^{w_j} \left\{ \overline{\boldsymbol{U}}_{lj}^{*}(\overline{v},\overline{w},v,w)\boldsymbol{p}_j(v,w) - \overline{\boldsymbol{P}}_{lj}^{*}(\overline{v},\overline{w},v,w)\boldsymbol{u}_j(v,w) \right\} J_j(v,w)\,dv\,dw, \tag{4}$$

$$v_{l-1} \le \overline{v} \le v_l,\;\; w_{l-1} \le \overline{w} \le w_l,\; v_{j-1} \le v \le v_j,\;\; w_{j-1} \le w \le w_j,\; l = 1,2,3,...,n.$$

The first integral function in (4) is written as

$$\overline{\boldsymbol{U}}_{lj}^{*}(\overline{v},\overline{w},v,w) = \frac{1}{16\pi(1-\upsilon)\mu\eta} \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ U_{21} & U_{22} & U_{23} \\ U_{31} & U_{32} & U_{33} \end{bmatrix}, \qquad \mu = \frac{E}{2(1+\upsilon)}, \tag{5}$$

where

$$U_{11} = (3-4\upsilon) + \frac{\eta_1^2}{\eta^2},\; U_{12} = \frac{\eta_1\eta_2}{\eta^2},\; U_{13} = \frac{\eta_1\eta_3}{\eta^2},\; U_{21} = \frac{\eta_2\eta_1}{\eta^2},\; U_{22} = (3-4\upsilon) + \frac{\eta_2^2}{\eta^2},$$

$$U_{23} = \frac{\eta_2\eta_3}{\eta^2},\; U_{31} = \frac{\eta_3\eta_1}{\eta^2},\; U_{32} = \frac{\eta_3\eta_2}{\eta^2},\; U_{33} = (3-4\upsilon) + \frac{\eta_3^2}{\eta^2},$$

and E is Young's modulus, υ is Poisson's ratio, and μ is shear modulus.

The second integral function in (4) is represented by the following expression

$$\overline{\boldsymbol{P}}_{lj}^{*}(\overline{v},\overline{w},v,w) = \frac{-1}{8\pi(1-\upsilon)\eta^2} \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix}, \tag{6}$$

where

$$P_{11} = \left\{(1-2\upsilon) + 3\frac{\eta_1^2}{\eta^2}\right\}\frac{\partial\eta}{\partial n},\; P_{12} = 3\frac{\eta_1\eta_2}{\eta^2}\frac{\partial\eta}{\partial n} - (1-2\upsilon)\frac{\eta_1 n_2 - \eta_2 n_1}{\eta},$$

$$P_{13} = 3\frac{\eta_1\eta_3}{\eta^2}\frac{\partial\eta}{\partial n} - (1-2\upsilon)\frac{\eta_1 n_3 - \eta_3 n_1}{\eta},\; P_{21} = 3\frac{\eta_2\eta_1}{\eta^2}\frac{\partial\eta}{\partial n} - (1-2\upsilon)\frac{\eta_2 n_1 - \eta_1 n_2}{\eta},$$

$$P_{22} = \left\{(1-2\upsilon) + 3\frac{\eta_2^2}{\eta^2}\right\}\frac{\partial\eta}{\partial n},\; P_{23} = 3\frac{\eta_2\eta_3}{\eta^2}\frac{\partial\eta}{\partial n} - (1-2\upsilon)\frac{\eta_2 n_3 - \eta_3 n_2}{\eta},$$

$$P_{31} = 3\frac{\eta_3\eta_1}{\eta^2}\frac{\partial\eta}{\partial n} - (1-2\upsilon)\frac{\eta_3 n_1 - \eta_1 n_3}{\eta},\; P_{32} = 3\frac{\eta_3\eta_2}{\eta^2}\frac{\partial\eta}{\partial n} - (1-2\upsilon)\frac{\eta_3 n_2 - \eta_2 n_3}{\eta},$$

$$P_{33} = \left\{(1-2\upsilon) + 3\frac{\eta_3^2}{\eta^2}\right\}\frac{\partial\eta}{\partial n},\; \frac{\partial\eta}{\partial n} = \frac{\eta_1}{\eta}n_1 + \frac{\eta_2}{\eta}n_2 + \frac{\eta_3}{\eta}n_3,$$

and n_1, n_2, n_3 are components of the normal vector $\boldsymbol{n}_j$ to the Bézier patch of the boundary with index j.

The shape of the boundary represented by Bézier patches $(l, j = 1,2,...,n)$ i.e. in the form obtained from T-splines after Bézier extraction is included into kernels (5) and (6) by

$$\eta_1 = P_j^{(1)}(v,w) - P_l^{(1)}(\overline{v},\overline{w}),\, \eta_2 = P_j^{(2)}(v,w) - P_l^{(2)}(\overline{v},\overline{w}),\, \eta_3 = P_j^{(3)}(v,w) - P_l^{(3)}(\overline{v},\overline{w}),\; \eta = \left[\eta_1^2 + \eta_2^2 + \eta_3^2\right]^{0.5},$$

where $P_j^{(1)}(v,w), P_j^{(2)}(v,w), P_j^{(3)}(v,w)$ are scalar components of the individual Bézier patches and depend upon parameters v and w defined as the coordinate of the standard domain of individual Bézier patch, $\tilde{\boldsymbol{\xi}} \equiv \{v,w\}$.

In addition, functions $\boldsymbol{u}_j(v,w)$ and $\boldsymbol{p}_j(v,w)$ in (4) represent the boundary field variables in the form of the boundary conditions as well as the unknown values on the boundary. We call them the boundary functions. Let us note, that the form of boundary functions in PIES used to approximate the boundary field variables is completely independent from the functions used to describe the boundary shape. Hence, we can describe this as a strict separation between the approximation of the boundary shape and the boundary functions. The direct inclusion of the boundary shape in (4), through functions (5) and (6), is the main advantage of PIES over the traditional BIE. As a result, the proposed PIES is characterized in that its numerical solution does not require, in contrast to the BEM, discretization at the level of the aforementioned boundary declaration as well as the discretization of the boundary functions.

As mentioned earlier, the shape of the boundary in PIES is defined directly in the kernels (5), (6) using surface patches, and not by the boundary integral as in the classic BIE. This formulation is very advantageous since the boundary shape modification does not require re-discretization and PIES can automatically adapt to the modified shape. This is a significant advantage, especially in inverse problems relating to, for example, the identification of the optimal shape of the boundary. For more details of an analytical modification of BIE in order to obtain PIES with indicated differences between them, we refer the reader to (Zieniuk et al. 2018).

Furthermore, functions $\boldsymbol{u}_j(v,w)$, $\boldsymbol{p}_j(v,w)$ represent the fields and its normal derivative to the boundary defined on each of Bézier patch and are approximated by Lagrange polynomials in the following form

$$\boldsymbol{u}_j(v,w)=\sum_{p=0}^{N}\sum_{r=0}^{M}\boldsymbol{u}_j^{(pr)}L_j^{(p)}(v)L_j^{(r)}(w), \tag{7}$$

$$\boldsymbol{p}_j(v,w)=\sum_{p=0}^{N}\sum_{r=0}^{M}\boldsymbol{p}_j^{(pr)}L_j^{(p)}(v)L_j^{(r)}(w), \tag{8}$$

where $\boldsymbol{u}_j^{(pr)}$, $\boldsymbol{p}_j^{(pr)}$ are coefficients defined as the boundary conditions or obtained as the result of the solution of PIES, $L_j^{(p)}(v)$, $L_j^{(r)}(w)$ are basis functions in the form of the Lagrange polynomials and $\overline{n}=N*M$ is the number of the Lagrange polynomials on each Bézier surface.

After substituting the series (7) and (8) into formula (4), we obtain the following approximate expression

$$\begin{aligned}0.5\boldsymbol{u}_l(\overline{v},\overline{w})=\sum_{j=1}^{n}\sum_{p=0}^{N}\sum_{r=0}^{M}\Bigg\{&\boldsymbol{p}_j^{(pr)}\int_{\overline{v}_{j-1}}^{\overline{v}_j}\int_{\overline{w}_{j-1}}^{\overline{w}_j}\overline{\boldsymbol{U}}_{lj}^{*}(\overline{v},\overline{w},v,w)L_j^{(p)}(v)L_j^{(r)}(w)J_j(v,w)dvdw\\&-\boldsymbol{u}_j^{(pr)}\int_{\overline{v}_{j-1}}^{\overline{v}_j}\int_{\overline{w}_{j-1}}^{\overline{w}_j}\overline{\boldsymbol{P}}_{lj}^{*}(\overline{v},\overline{w},v,w)L_j^{(p)}(v)L_j^{(r)}(w)J_j(v,w)dvdw\Bigg\}.\end{aligned} \tag{9}$$

After solving PIES (8) by pseudospectral method (Gottlieb et al. 1977) in the collocation points, we obtain the unknown functions $\boldsymbol{u}_j(v,w)$ or $\boldsymbol{p}_j(v,w)$ for the individual surface patches. PIES differs from classical BIE because in PIES the boundary is formally defined by surface patches. Therefore PIES is physically defined not on the boundary as in BIE, but within the parametric domain of these patches. This is caused by the direct inclusion of the boundary shape into the kernels of the PIES. They are called the fundamental and singular boundary solutions to distinguish them from their classical counterparts, which are not include the boundary. This advantage allows to separate the approximation of the boundary shape from the boundary functions

4. EXAMPLES

An application of the PIES method with proposed boundary representation is illustrated by numerical examples. We present how to generate the boundary using T-splines with Bézier extraction discussed in section 3. In addition, the numerical results are compared with exact solutions known from literature. The considered domain (torus) is generated using the Autodesk T-spline plugin for Rhino3d. This plugin also includes the Bézier extraction module that directly converts T-splines to Bézier surfaces. This module is described in more detail in (Scott et al. 2013). As shown in Fig. 2, the boundary of a considered domain is defined using 32 rectangular Bézier patches of degree 3.

In order to verify the proposed approach, inside the boundary created by Bézier patches the Navier-Lamé equation with displacement Dirichlet boundary conditions is solved. Presented Dirichlet boundary conditions depend on the Cartesian coordinates $\boldsymbol{x} \equiv \{x_1, x_2, x_3\}$ are obtained for the following two independent exact solutions functions of displacements

$$u_1 = 0.5(2x_1 + x_2 + x_3)\,,\ \ u_2 = 0.5(x_1 + 2x_2 + x_3)\,,\ \ u_3 = 0.5(x_1 + x_2 + 2x_3)\,, \tag{10}$$

and

$$u_1 = x_2^3 - 3x_2x_3^2\,,\ \ u_2 = x_3^3 - 3x_3x_1^2\,,\ \ u_3 = x_1^3 - 3x_1x_2^2\,, \tag{11}$$

that satisfy the Navier-Lamé equation. The material properties are as follows: Young's modulus $E = 1$, Poisson's ratio $\upsilon = 0.25$ and shear modulus $\mu = 0.4$.

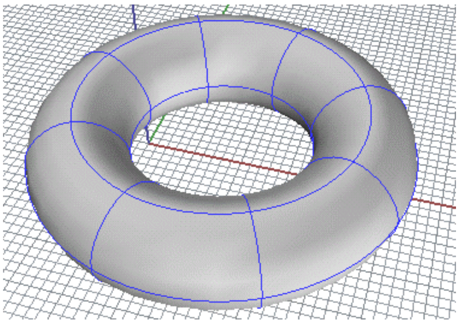

Figure 2. Description of the boundary from example by 32 Bézier patches after the extraction.

A pseudospectral method with collocation points is used to solve PIES. Fig. 3 shows the convergence of solutions in PIES depending on the number of solved algebraic equations. The results expressed by the L_2 error norm are obtained for both functions (10) and (11).

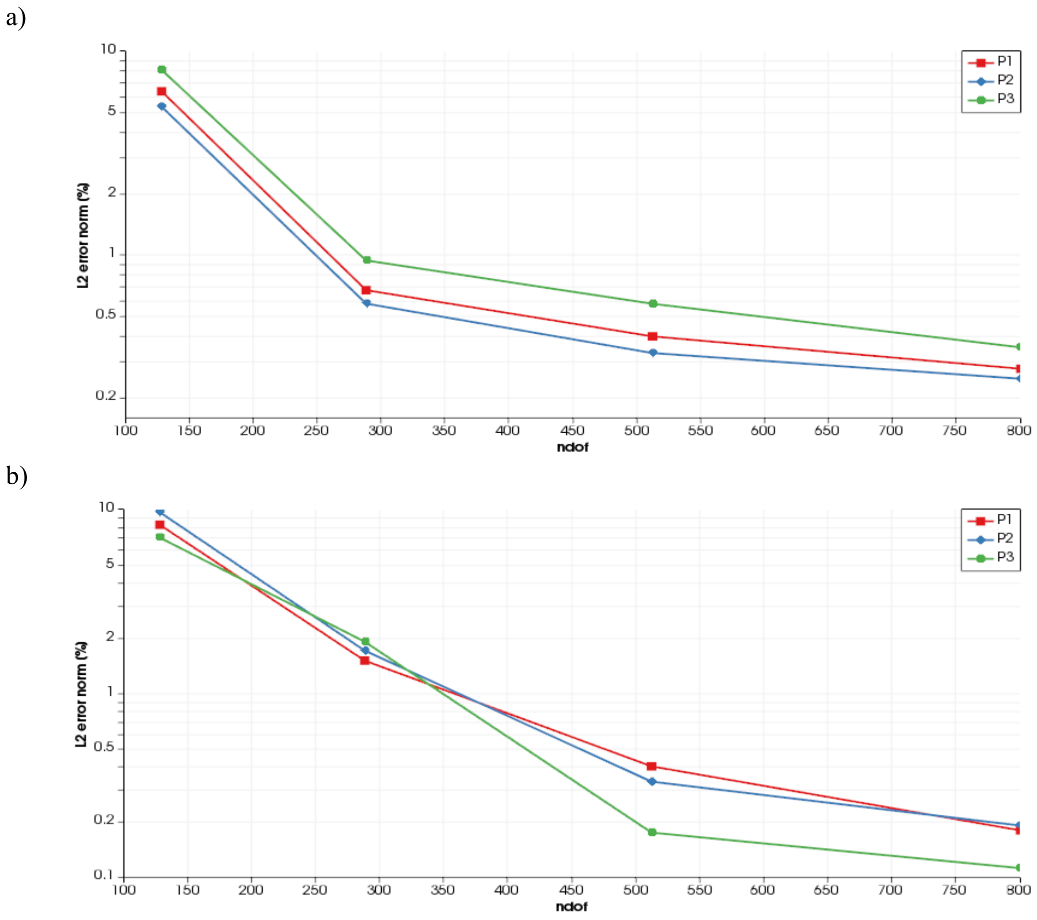

Figure. 3. Relation between the number of solved algebraic equations and the level of error L2 for solutions on the boundary for boundary conditions (10) (a) and (11) (b).

The results show that with an increasing number of used collocation points the improvement in accuracy of the solution is achieved. The number of generated equations is directly related to the number of the collocation points. The collocation points are defined in a normalized parametric domain of each Bézier patch with their arrangement in accordance with the roots of the Chebyshev polynomials of the second kind. The calculations are repeated for collocation points 4, 9, 16, and 25 respectively. This is done without any modification of the boundary representation from Fig. 2 and is reduced only to changing the number of collocation points in the parametric domains of the patches. This is due to the aforementioned separation of the boundary representation and the approximation of the boundary functions on it. This number is directly related to the size of the algebraic equation necessary to solve the problem. These were respectively 128, 288, 512 and 800 equations.

5. CONCLUSIONS

T-splines significantly simplify the modeling of geometric objects (the shape of the boundary of solved boundary value problems) in comparison with NURBS surfaces. Easy conversion of T-splines into Bézier surfaces allows the use of PURC to solve boundary value problems with boundaries described with T-splines to obtain accurate solutions. It would be interesting to study the conversion of T-splines into other surfaces previously used in PIES and to study the effectiveness of such an approach. Ultimately, it is planned to apply T-splines directly in PIES without converting these into Bézier patches first. However, this requires extensive research.

REFERENCES

Borden, M.J., Scott, M.A., Evans, J.A., Hughes, T.J. (2011). Isogeometric finite element data structures based on Bézier extraction of NURBS. International Journal for Numerical Methods in Engineering 87.1-5, 15-47.

Farin, G. (1999). NURBS for Curve & Surface Design: From Projective Geometry to Practical Use. AK Peters/CRC Press.

Forsey, D.R., Bartels, R.H. (1988). Hierarchical B-spline refinement ACM Siggraph Computer Graphics 22.4, 205-212.

Gottlieb, D., Orszag S.A. (1977). Numerical analysis of spectral methods: theory and applications. Vol. 26. Siam.

Deng, J., Chen, F., Li, X., Hu, C., Tong, W., Yang, Z., Feng, Y. (2008). Polynomial splines over hierarchical T-meshes. Graphical Models 70.4, 76-86.

Piegl, L., Wayne T. (2012). The NURBS book. Springer Science & Business Media.

Scott, M.A., Borden, M.J., Verhoosel, C.V., Sederberg, T.W., Hughes, T.J. (2011). Isogeometric finite element data structures based on Bézier extraction of T-splines. International Journal for Numerical Methods in Engineering 88.2 , 126-156.

Scott, M.A., Hughes, T.J., Sederberg, T.W., Sederberg, M.T. (2013). An integrated approach to engineering design and analysis using the Autodesk T-spline plugin for Rhino3d. Advances in Engineering Software (in preparation) (2013).

Sederberg, T.W., Zheng, J., Bakenov, A., Nasri, A. (2003). T-splines and T-NURCCs. ACM transactions on graphics (TOG) 22.3, 477-484.

Sederberg, T.W., Cardon, D.L., Finnigan, G.T., North, N.S., Zheng, J., Lyche, T. (2004). T-spline simplification and local refinement. ACM Transactions on Graphics 23.3, 276-283. ACM.

Dokken, T., Lyche, T., Pettersen, K.F. (2013) Polynomial splines over locally refined box-partitions. Computer Aided Geometric Design 30.3, 331-356.

Zieniuk, E., Szerszeń, K. (2014). The PIES for solving 3D potential problems with domains bounded by rectangular Bézier patches. Engineering Computations 31.4, 791-809.

Zieniuk, E., Szerszeń, K. (2018). A separation of the boundary geometry from the boundary functions in PIES for 3D problems modeled by the Navier–Lamé equation. Computers & Mathematics with Applications 75.4, 1067-1094.

Zieniuk, E., Szerszeń, K. (2019). A separation between the boundary shape and the boundary functions in the parametric integral equation system for the 3D Stokes equation. Numerical Algorithms 80.3, 753-780.