INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 0000; **00**:1–26
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme

Powell-Sabin B-splines and unstructured standard T-splines for the solution of the Kirchhoff-Love plate theory exploiting Bézier extraction

Stefan May[1], Julien Vignollet[1], René de Borst[1]*

[1] *University of Glasgow, School of Engineering, Rankine Building, Oakfield Avenue, Glasgow G12 8LT, UK.*

SUMMARY

The equations that govern Kirchhoff-Love plate theory are solved using quadratic Powell-Sabin B-splines and unstructured standard T-splines. Bézier extraction is exploited to make the formulation computationally efficient. Since quadratic Powell-Sabin B-splines result in $\mathcal{C}^1_{\mathrm{A}}$-continuous shape functions, they are of sufficiently high continuity to capture Kirchoff-Love plate theory when cast in a weak form. Unlike Non-Uniform Rational B-Splines (NURBS) which are commonly used in isogeometric analysis, Powell-Sabin B-splines do not necessarily capture the geometry exactly. However, the fact that they are defined on triangles instead of on quadrilaterals increases their flexibility in meshing, and can make them competitive with respect to NURBS, as no bending strip method for joined NURBS patches is needed. This paper further illustrates how unstructured T-splines can be modified such that they are $\mathcal{C}^1_{\mathrm{A}}$-continuous around extraordinary points, and that the blending functions fulfil the partition of unity property. The performance of quadratic NURBS, unstructured T-splines, Powell-Sabin B-splines and NURBS-to-NURPS (Non-Uniform Rational Powell-Sabin B-splines which are obtained by a transformation from a NURBS patch) is compared in a study of a circular plate. Copyright © 0000 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: isogeometric analysis, Powell-Sabin B-splines, NURBS-to-NURPS, NURBS, unstructured T-splines, Bézier extraction

1. INTRODUCTION

Kirchhoff-Love plate theory [1] is well suited for the analysis of thin plates, but requires $\mathcal{C}^1_{\mathrm{A}}$-continuous shape functions in finite element analyses (the use of the subscript "A" is explained in Section 2). Other problems which require higher-order continuous shape functions include higher-order gradient damage models [2], higher-order shear deformation theory [3], or the Cahn-Hilliard equation [4]. References [2–4] utilise either B-splines, Non-Uniform Rational B-splines (NURBS) or T-splines [5]. These higher-order continuous shape functions are suitable for problems which require an interelement continuity that is higher than $\mathcal{C}^0_{\mathrm{A}}$. The idea of applying basis functions like B-splines, NURBS or T-splines, which are routinely used in Computer Aided Design (CAD), to analysis as well, was introduced in [6] and termed IsoGeometric Analysis (IGA) in [7].

A vibration analysis for Kirchhoff-Love plates using isogeometric analysis has been carried out in [8], while an isogeometric formulation for Kirchhoff-Love shell elements was proposed in [9]. Since multiple NURBS patches are joined with $\mathcal{C}^0_{\mathrm{A}}$-continuity, the bending strip method was proposed in [10] and adds a penalty stiffness between adjacent NURBS patches. This approach

*Correspondence to: René de Borst, University of Glasgow, School of Engineering, Oakfield Avenue, Rankine Building, Glasgow G12 8LT, UK. E-mail: Rene.DeBorst@glasgow.ac.uk requires a geometric continuity $\mathcal{G}^1$ along the $\mathcal{C}^0_{\mathrm{A}}$-continuity lines. Hence, the control points need to be positioned such that a $\mathcal{G}^1$-continuity is obtained. Unfortunately, this ($\mathcal{G}^1$, $\mathcal{C}^0_{\mathrm{A}}$)-construction has been applied only to two or four adjacent NURBS patches – it has not been demonstrated how to position the control points when three or more than four NURBS patches meet at an extraordinary point.

T-splines are an alternative to NURBS and consist of a single patch. However, when three or more than four elements in a T-spline mesh meet at an extraordinary point, a $\mathcal{C}^0_{\mathrm{A}}$-continuity line is introduced in the vicinity of the extraordinary point and the blending functions of the T-spline mesh do not fulfil the partition of unity property. It was demonstrated in [11] how to obtain $\mathcal{G}^1$-continuous blending functions by modifying the coefficients of the Bézier extraction operators in the one-ring neighbourhood elements of an extraordinary point. It was pointed out in [12] that this $\mathcal{G}^1$-construction also yields $\mathcal{C}^1_{\mathrm{A}}$-continuous blending functions. However, the approach in [11] does not result in blending functions that fulfil the partition of unity property.

Quadratic Powell-Sabin (PS) B-splines also provide shape functions that give a higher continuity [13]. They are based on a linear triangulation and have quadratic shape functions that are $\mathcal{C}^1_{\mathrm{A}}$-continuous across elements. Powell-Sabin B-splines have been cast in terms of Bézier ordinates in [14, 15] in order to obtain an efficient calculation. It is noted that Powell-Sabin B-splines are not based on the isogeometric concept as they only *approximate* the exact geometry and are not based on a NURBS patch but on a triangulation. To address this issue, a method was recently proposed to transform a single NURBS patch into Non-Uniform Rational Powell-Sabin B-splines (NURPS) [16] and denoted NURBS-to-NURPS. The boundary of the NURBS-to-NURPS then exactly matches the boundary of the NURBS patch while the interior domain of the NURBS-to-NURPS only approximates the NURBS patch which is relevant for non-planar geometries. Powell-Sabin B-splines and NURBS-to-NURPS have been used for analysis in [16–19].

After clarifying the term continuity from a CAD and from an analysis perspective, this paper proposes a Powell-Sabin B-spline formulation for Kirchhoff-Love plates. In this class of problems the $\mathcal{C}^1_{\mathrm{A}}$-continuity of Powell-Sabin splines is fully exploited. Moreover, we will show how Bézier extraction can be implemented, thus making the method computationally efficient, similar to Bézier extraction for NURBS and T-splines [20, 21]. Then, unstructured quadratic T-splines will be modified such that the blending functions fulfil the partition of unity property and are $\mathcal{C}^1_{\mathrm{A}}$-continuous around the extraordinary point. Finally, we compare NURBS, unstructured T-splines and NURBS-to-NURPS with Powell-Sabin B-splines. As a test problem we take a circular Kirchhoff-Love plate, and we study the cases with clamped and with simply supported boundary conditions. Our study goes beyond that in [22], which was for the Poisson equation, and hence required only $\mathcal{C}^0_{\mathrm{A}}$-continuous shape functions. Particular attention is given to cases with extraordinary points in the mesh, as their effect on the convergence behaviour of fourth-order partial differential equations has not been studied hitherto.

2. CONTINUITY FOR CAD AND ANALYSIS

In this section we elaborate on the term "continuity" from a CAD perspective as well as from an analysis perspective.

2.1. Parametric continuity $\mathcal{C}$

We consider the two curved segments $\underline{\boldsymbol{S}}_1(t_1)$ and $\underline{\boldsymbol{S}}_2(t_2)$ which depend on the parametric coordinates t_1 and t_2:

$$\begin{aligned} \underline{\boldsymbol{S}}_1(t_1) &= (1-t_1)\underline{\boldsymbol{P}}_1 + t_1\underline{\boldsymbol{P}}_2 \quad 0 \le t_1 \le 1, \\ \underline{\boldsymbol{S}}_2(t_2) &= (1-t_2)\underline{\boldsymbol{P}}_2 + t_2\underline{\boldsymbol{P}}_3 \quad 0 \le t_2 \le 1 \end{aligned} \tag{1}$$

with the coordinates in the physical domain $\underline{\boldsymbol{x}} = (x, y)$

$$\underline{\boldsymbol{P}}_1 = (1,1), \quad \underline{\boldsymbol{P}}_2 = (2,2), \quad \underline{\boldsymbol{P}}_3 = (3,3). \tag{2}$$

Both curves $\mathcal{S}_1(t_1)$ and $\mathcal{S}_2(t_2)$ have been plotted in Figure 1(a).

DOI: 10.1002/nme

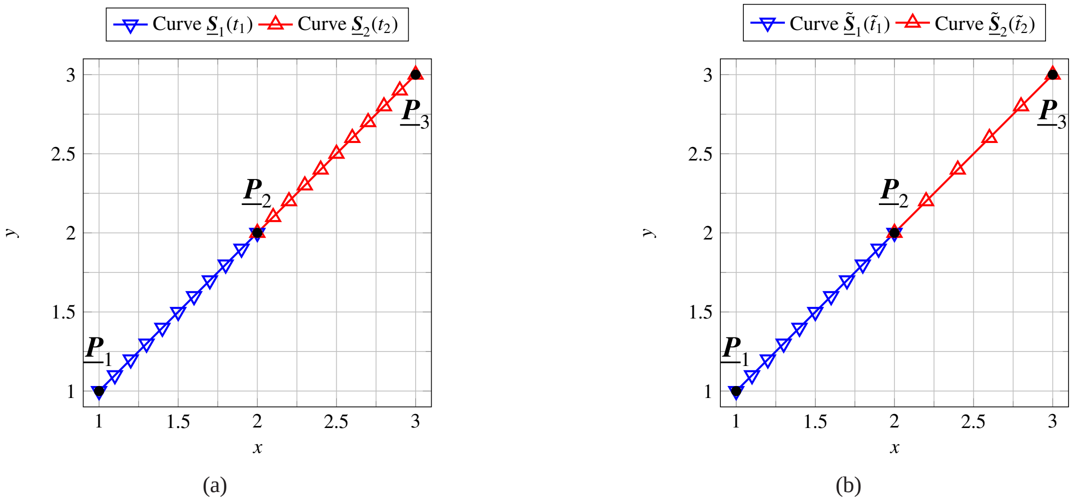


Figure 1. Parametric curves in the physical domain $\underline{\boldsymbol{x}} = (x, y)$, (a) $\mathcal{S}_1(t_1)$ and $\mathcal{S}_2(t_2)$ are *parametrically* $\mathcal{C}^1$-continuous at $\underline{\boldsymbol{P}}_2$, (b) $\tilde{\mathcal{S}}_1(\tilde{t}_1)$ and $\tilde{\mathcal{S}}_2(\tilde{t}_2)$ are *geometrically* $\mathcal{G}^1$-continuous at their joint $\underline{\boldsymbol{P}}_2$. The triangles mark isoparametric distances of $\Delta t_1 = \Delta t_2 = \Delta\tilde{t}_1 = \Delta\tilde{t}_2 = 0.1$.

Evaluating the first derivative with respect to the parametric coordinate yields for both curves:

$$\left.\frac{\partial \underline{\boldsymbol{S}}_1(t_1)}{\partial t_1}\right|_{t_1=1} = \left.\frac{\partial \underline{\boldsymbol{S}}_2(t_2)}{\partial t_2}\right|_{t_2=0} = (1,1). \tag{3}$$

Both curves have the same first derivative at their joint: the *parametric* continuity of the first derivative in Equation (3) will be denoted as $\mathcal{C}^1$.

2.2. Geometric continuity $\mathcal{G}$

Next, we consider the two curved segments $\underline{\tilde{\boldsymbol{S}}}_1(\tilde{t}_1)$ and $\underline{\tilde{\boldsymbol{S}}}_2(\tilde{t}_2)$, which depend on the parametric coordinates $\tilde{t}_1$ and $\tilde{t}_2$ with:

$$\begin{aligned} \underline{\tilde{\boldsymbol{S}}}_1(\tilde{t}_1) &= (1-\tilde{t}_1)\underline{\boldsymbol{P}}_1 + \tilde{t}_1\underline{\boldsymbol{P}}_2 \qquad && 0 \le \tilde{t}_1 \le 1, \\ \underline{\tilde{\boldsymbol{S}}}_2(\tilde{t}_2) &= (1-2\tilde{t}_2)\underline{\boldsymbol{P}}_2 + 2\tilde{t}_2\underline{\boldsymbol{P}}_3 \qquad && 0 \le \tilde{t}_2 \le \frac{1}{2}. \end{aligned} \tag{4}$$

The first derivatives with respect to the parametric coordinate now read:

$$(1,1) = \left.\frac{\partial \underline{\tilde{\boldsymbol{S}}}_1(\tilde{t}_1)}{\partial \tilde{t}_1}\right|_{\tilde{t}_1=1} \neq \left.\frac{\partial \underline{\tilde{\boldsymbol{S}}}_2(\tilde{t}_2)}{\partial \tilde{t}_2}\right|_{\tilde{t}_2=0} = (2,2). \tag{5}$$

Hence, both curves are not $\mathcal{C}^1$-continuous. However, Figure 1(b) shows that both curves still have a continuous geometry in the physical domain $\underline{\boldsymbol{x}}$. In order to account for this, the term *geometric* continuity was introduced in [23]. If two curved segments satisfy the condition

$$\left.\frac{\partial \underline{\tilde{\boldsymbol{S}}}_1(\tilde{t}_1)}{\partial \tilde{t}_1}\right|_{\tilde{t}_1=1} = c\left.\frac{\partial \underline{\tilde{\boldsymbol{S}}}_2(\tilde{t}_2)}{\partial \tilde{t}_2}\right|_{\tilde{t}_2=0}, \tag{6}$$

with a scalar c, then they are called *geometric* continuous with respect to the first derivative. This continuity is denoted as $\mathcal{G}^1$. The step from Equation (4) to Equation (1) – replacing $\tilde{t}_2$ with $\frac{t_2}{2}$ – is called reparameterisation. It was pointed out in [24] that the definitions for parametric and geometric continuity in Equations (3) and (6) may not be clear for overlapping control points.

2.3. *Continuity $\mathcal{C}_{\mathrm{A}}$ for analysis*

Now, we consider the notion "continuity" from an analysis perspective. Equation (1) is equivalent to the parameterisation:

$$\bar{\underline{\boldsymbol{S}}}(t) = N_1(t)\underline{\boldsymbol{P}}_1 + N_2(t)\underline{\boldsymbol{P}}_2 + N_3(t)\underline{\boldsymbol{P}}_3 \tag{7}$$

with $0 \leq t \leq 2$ and the Lagrangian shape functions (see Figure 2)

$$N_1(t) = \begin{cases} 1-t & \text{for } 0 \leq t \leq 1 \\ 0 & \text{for } 1 \leq t \leq 2 \end{cases}, \quad N_2(t) = \begin{cases} t & \text{for } 0 \leq t \leq 1 \\ 2-t & \text{for } 1 \leq t \leq 2 \end{cases}, \\ N_3(t) = \begin{cases} 0 & \text{for } 0 \leq t \leq 1 \\ t-1 & \text{for } 1 \leq t \leq 2 \end{cases}. \tag{8}$$

It is noted that the shape functions $N_i(t)$ in Equation (8) can also be derived from the knot vector $T = \{0, 0, 1, 2, 2\}$. The Lagrangian shape functions in Equation (8) have a continuity of $\mathcal{C}^0_{\mathrm{A}}$. In order to distinguish between the *parametric* continuity $\mathcal{C}$ of the *curve* and the continuity $\mathcal{C}^0_{\mathrm{A}}$ of the *basis functions*, the subscript A (for analysis) was added. It follows that the continuity which is required for analysis needs to be distinguished from the parametric or geometric continuity which is interesting from a designer's point of view.

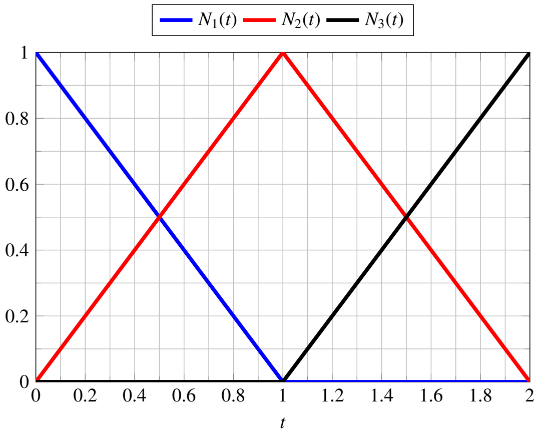


Figure 2. $\mathcal{C}^0_{\mathrm{A}}$-continuous linear Lagrangian shape functions.

2.4. *Geometric continuous basis functions*

It was demonstrated in [12] that $\mathcal{G}^k$-continuous basis functions are $\mathcal{C}^k_{\mathrm{A}}$-continuous. This will be illustrated for the case ($\mathcal{G}^1$, $\mathcal{C}^1_{\mathrm{A}}$) in one dimension. Consider the four basis functions N_i

$$\begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \end{bmatrix}, \tag{9}$$

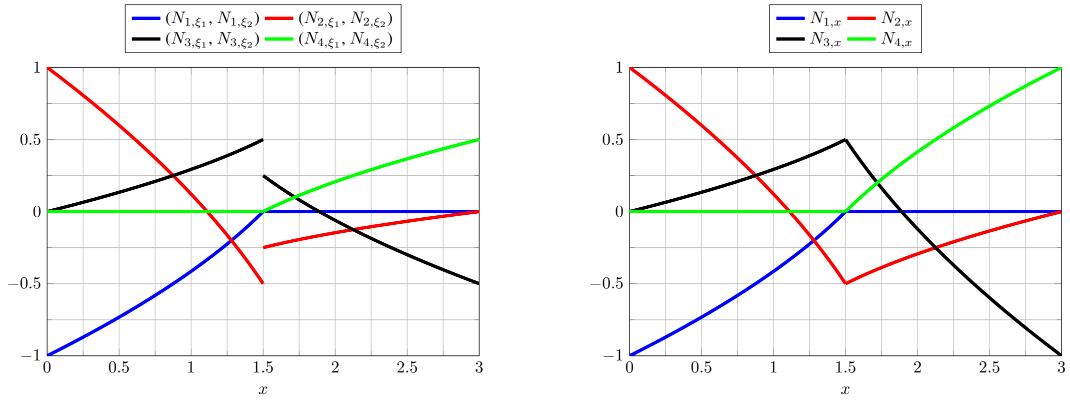
with

$$\left.\begin{aligned} B_1(\xi_1) &= \tfrac{1}{4}(1-\xi_1)^2 \\ B_2(\xi_1) &= \tfrac{1}{2}(1-\xi_1^2) \\ B_3(\xi_1) &= \tfrac{1}{4}(1+\xi_1)^2 \end{aligned}\right\} -1 \leq \xi_1 \leq 1, \qquad \left.\begin{aligned} B_4(\xi_2) &= \tfrac{1}{16}(2-\xi_2)^2 \\ B_5(\xi_2) &= \tfrac{1}{8}(4-\xi_2^2) \\ B_6(\xi_2) &= \tfrac{1}{16}(2+\xi_2)^2 \end{aligned}\right\} -2 \leq \xi_2 \leq 2. \tag{10}$$

The following physical coordinate $P_i = x_i$ corresponds to each basis function

$$P_1 = 0, \quad P_2 = 1, \quad P_3 = 2, \quad P_4 = 3. \tag{11}$$

The derivatives N_{i,ξ_1} and N_{i,ξ_2} with respect to the parameter coordinates ξ_1 and ξ_2 are plotted in Figure 3(a) over the physical domain x for all four basis functions.



(a)

(b)

Figure 3. (a) The derivatives $(N_{2,\xi_1},\, N_{2,\xi_2})$ and $(N_{3,\xi_1},\, N_{3,\xi_2})$ are discontinuous while (b) the derivatives $N_{2,x}$ and $N_{3,x}$ are continuous: $\mathcal{G}^1$-continuous basis functions are also $\mathcal{C}^1_{\mathrm{A}}$-continuous.

It can be observed that $(N_{2,\xi_1},\, N_{2,\xi_2})$ and $(N_{3,\xi_1},\, N_{3,\xi_2})$ are discontinuous. However, since the $\mathcal{G}^1$-continuity condition

$$\left.\frac{\partial N_i}{\partial \xi_1}\right|_{\xi_1=1} = 2\left.\frac{\partial N_i}{\partial \xi_2}\right|_{\xi_2=-2} \tag{12}$$

holds for all four basis function, they have to be $\mathcal{C}^1_{\mathrm{A}}$-continuous: Figure 3(b) shows the derivatives $N_{i,x}$ of all four basis functions with respect to the physical coordinate x. All four derivatives $N_{i,x}$ are now continuous. Thus, the four basis function N_i are $\mathcal{C}^1_{\mathrm{A}}$-continuous. The property that $\mathcal{G}^1$-continuous basis functions are $\mathcal{C}^1_{\mathrm{A}}$-continuous is especially useful for connecting surfaces at an extraordinary point such that the basis functions possess the required $\mathcal{C}^1_{\mathrm{A}}$-continuity (see Section 5). In sum, basis functions that fulfil a $\mathcal{G}^1$-constraint are $\mathcal{C}^1_{\mathrm{A}}$-continuous and the geometry is $\mathcal{G}^1$-continuous for any set of control points. If the basis functions are only $\mathcal{C}^0_{\mathrm{A}}$-continuous the geometry can still be $\mathcal{G}^1$-continuous by an appropriate choice of the location of the coordinates in the physical domain. Such a $(\mathcal{G}^1,\, \mathcal{C}^0_{\mathrm{A}})$-construction can also be used for solving fourth order partial differential equations when a rigid link between neighbouring control points along the $\mathcal{C}^0_{\mathrm{A}}$-continuity is introduced, as for the bending strip method [10]. This rigid link transfers the $\mathcal{G}^1$-continuity from the coordinates to the displacement degrees of freedom.

3. KIRCHHOFF-LOVE PLATE THEORY

3.1. *Continuum formulation*

The moment equilibrium for a Kirchhoff-Love plate reads [25]:

$$m_{\alpha\beta,\alpha\beta} = p_0 \tag{13}$$

with p_0 the force per unit area, and

$$m_{\alpha\beta} = -\int_{-h/2}^{h/2} \sigma_{\alpha\beta} z \,\mathrm{d}z \tag{14}$$

the bending moments. Greek indices take values one and two, a comma denotes differentiation, while h is the thickness of the plate. $\sigma_{\alpha\beta}$ is the stress, and Hooke's law for plane stress is used:

$$\begin{bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{12}\end{bmatrix} = \frac{E}{1-\nu^2}\begin{bmatrix}1 & \nu & 0\\ \nu & 1 & 0\\ 0 & 0 & \frac{1-\nu}{2}\end{bmatrix}\begin{bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ 2\varepsilon_{12}\end{bmatrix} \tag{15}$$

since it is assumed that $|\sigma_{33}|, |\sigma_{13}|, |\sigma_{23}| \ll |\sigma_{11}|, |\sigma_{22}|, |\sigma_{12}|$. The non-zero strain components are assumed as follows:

$$\varepsilon_{11} = -z\frac{\partial^2 w}{\partial x^2}, \quad \varepsilon_{22} = -z\frac{\partial^2 w}{\partial y^2}, \quad 2\varepsilon_{12} = -2z\frac{\partial^2 w}{\partial x \partial y} \tag{16}$$

with the deflection $w(x, y)$. Upon substitution of Equation (15) and Equation (16) into Equation (14) we obtain:

$$\underbrace{\begin{bmatrix}m_{11}\\ m_{22}\\ m_{12}\end{bmatrix}}_{\underline{\boldsymbol{m}}} = \underbrace{\frac{Eh^3}{12(1-\nu^2)}\begin{bmatrix}1 & \nu & 0\\ \nu & 1 & 0\\ 0 & 0 & \frac{1-\nu}{2}\end{bmatrix}}_{\underline{\underline{\mathbf{D}}}}\underbrace{\begin{bmatrix}\kappa_{11}\\ \kappa_{22}\\ 2\kappa_{12}\end{bmatrix}}_{\underline{\boldsymbol{\kappa}}} \tag{17}$$

with $\underline{\underline{\mathbf{D}}}$ the elastic stiffness matrix and

$$\kappa_{\alpha\beta} = w_{,\alpha\beta} \tag{18}$$

the curvature. The bending moments are assembled in the array $\underline{\boldsymbol{m}}$, while $\underline{\boldsymbol{\kappa}}$ contains the curvatures. Multiplying Equation (13) by a test function δw, integrating over the domain Ω and exploiting Gauss' theorem then results in:

$$\int_\Omega \delta w_{,\alpha\beta} m_{\alpha\beta}\,\mathrm{d}x\mathrm{d}y + \int_\Gamma \delta w m_{\alpha\beta,\alpha} n_\beta\,\mathrm{d}\Gamma - \int_\Gamma \delta w_{,\beta} m_{\alpha\beta} n_\alpha\,\mathrm{d}\Gamma = \int_\Omega \delta w p_0\,\mathrm{d}x\mathrm{d}y. \tag{19}$$

Substitution of Equation (18) and omitting the boundary terms that relate to imposed moments and forces, the resulting weak form becomes:

$$\int_\Omega \delta\kappa_{\alpha\beta} m_{\alpha\beta}\,\mathrm{d}x\mathrm{d}y = \int_\Omega \delta w p_0\,\mathrm{d}x\mathrm{d}y. \tag{20}$$

Since second derivatives appear in this equation, $\mathcal{C}^1_{\mathrm{A}}$-continuous functions (functions of class H^2) are necessary with square integrable second derivatives [26].

3.2. Discretisation

Discretisation of the domain Ω into E elements, $\Omega = \bigcup_{e=1}^{E} \Omega^e$, with w the deflection and δw its variation, leads to:

$$w^e = \underline{\boldsymbol{N}}^T \underline{\boldsymbol{w}}, \quad \delta w^e = \underline{\boldsymbol{N}}^T \delta\underline{\boldsymbol{w}}, \quad \underline{\boldsymbol{\kappa}}^e = \underline{\underline{\mathbf{B}}}\,\underline{\boldsymbol{w}}, \quad \delta\underline{\boldsymbol{\kappa}}^e = \underline{\underline{\mathbf{B}}}\delta\underline{\boldsymbol{w}} \tag{21}$$

where $\underline{\boldsymbol{N}}$ contains the shape functions, and $\underline{\underline{\mathbf{B}}}^T = [N_{,xx}, N_{,yy}, 2N_{,xy}]$ contains the second derivatives of the shape functions. It is recalled that $\underline{\underline{\mathbf{D}}}$ is the elastic stiffness matrix. Use of Equation (17) results in the matrix-vector equation:

$$\int_\Omega \delta\underline{\boldsymbol{\kappa}}^T \underline{\underline{\mathbf{D}}}\,\underline{\boldsymbol{\kappa}}\,\mathrm{d}x\mathrm{d}y = \int_\Omega \delta\underline{\boldsymbol{w}}^T \underline{\boldsymbol{N}}\,p_0\,\mathrm{d}x\mathrm{d}y \tag{22}$$

which for arbitrary $\delta\underline{\boldsymbol{w}}$ gives:

$$\underbrace{\int_\Omega \underline{\underline{\mathbf{B}}}^T \underline{\underline{\mathbf{D}}}\,\underline{\underline{\mathbf{B}}}\,\mathrm{d}x\mathrm{d}y}_{\underline{\underline{\mathbf{K}}}}\ \underline{\boldsymbol{w}} = \underbrace{\int_\Omega \underline{\boldsymbol{N}}\,p_0\,\mathrm{d}x\mathrm{d}y}_{\underline{\boldsymbol{f}}} \tag{23}$$

with $\underline{\underline{\mathbf{K}}}$ the stiffness matrix and $\underline{\boldsymbol{f}}$ the force vector.

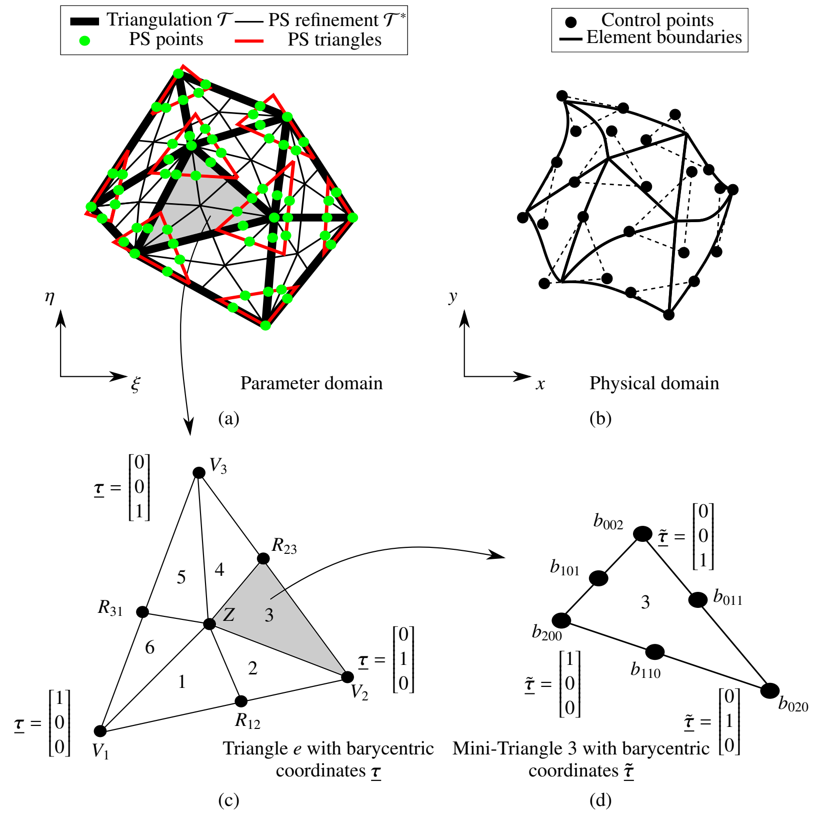

Triangulation $\mathcal{T}$
PS refinement $\mathcal{T}^*$
PS points
PS triangles
Control points
Element boundaries
η
ξ
Parameter domain
(a)
y
x
Physical domain
(b)
$\underline{\boldsymbol{\tau}} = \begin{bmatrix}0\\0\\1\end{bmatrix}$
V_3
R_{23}
R_{31}
5
4
Z
3
6
2
1
$\underline{\boldsymbol{\tau}} = \begin{bmatrix}0\\1\\0\end{bmatrix}$
V_2
R_{12}
$\underline{\boldsymbol{\tau}} = \begin{bmatrix}1\\0\\0\end{bmatrix}$
V_1
Triangle e with barycentric coordinates $\underline{\boldsymbol{\tau}}$
(c)
b_{002}
$\underline{\tilde{\boldsymbol{\tau}}} = \begin{bmatrix}0\\0\\1\end{bmatrix}$
b_{101}
b_{011}
3
b_{200}
$\underline{\tilde{\boldsymbol{\tau}}} = \begin{bmatrix}1\\0\\0\end{bmatrix}$
b_{110}
b_{020}
$\underline{\tilde{\boldsymbol{\tau}}} = \begin{bmatrix}0\\1\\0\end{bmatrix}$
Mini-Triangle 3 with barycentric coordinates $\underline{\tilde{\boldsymbol{\tau}}}$
(d)

Figure 4. The parameter and physical domains, ($\underline{\boldsymbol{\xi}}$) and ($\underline{\boldsymbol{x}}$), respectively, for Powell-Sabin B-splines. Each triangle e has a barycentric coordinate system $\underline{\boldsymbol{\tau}}$ and can further be subdivided into six mini-triangles with a barycentric coordinate system $\underline{\tilde{\boldsymbol{\tau}}}$.

4. BÉZIER EXTRACTION FOR QUADRATIC POWELL-SABIN B-SPLINES

This Section starts with a concise description of Powell-Sabin splines, including notions like Powell-Sabin refinement, Powell-Sabin points, and Powell-Sabin triangles. For a more in-depth treatment reference is made to [15].

We consider the parameter domain $\underline{\boldsymbol{\xi}} = (\xi, \eta)$ for a triangulation $\mathcal{T}$ (thick black lines) with $e = 1, 2, \dots E$ triangles and N_v vertices in Figure 4(a). A vertex k of the triangulation $\mathcal{T}$ has the coordinates $\underline{\boldsymbol{V}}_k = (\xi_k, \eta_k)$ in the parameter domain. m_k triangles are attached to vertex k and will be denoted as the molecule Ω_k.

Each triangle e of the triangulation $\mathcal{T}$ has a barycentric coordinate system with $\underline{\boldsymbol{\tau}} = (\tau_1, \tau_2, \tau_3)$, see Figure 4(c). The points $\underline{\boldsymbol{V}}_1$, $\underline{\boldsymbol{V}}_2$, $\underline{\boldsymbol{V}}_3$, $\underline{\boldsymbol{R}}_{12}$, $\underline{\boldsymbol{R}}_{23}$, $\underline{\boldsymbol{R}}_{31}$ and $\underline{\boldsymbol{Z}}$ have the following barycentric coordinates (τ_1, τ_2, τ_3):

$$\begin{gathered} \underline{\boldsymbol{V}}_1 = (1,0,0), \quad \underline{\boldsymbol{V}}_2 = (0,1,0), \quad \underline{\boldsymbol{V}}_3 = (0,0,1), \\ \underline{\boldsymbol{R}}_{12} = (\lambda_1, \lambda_2, 0), \quad \underline{\boldsymbol{R}}_{23} = (0, \mu_2, \mu_3), \quad \underline{\boldsymbol{R}}_{31} = (\nu_1, 0, \nu_3), \quad \underline{\boldsymbol{Z}} = (a, b, c). \end{gathered} \tag{24}$$

Splitting each triangle e of the triangulation $\mathcal{T}$ in Figure 4(a) into six mini-triangles ($n = 1, 2, \dots, 6$) yields the Powell-Sabin refinement $\mathcal{T}^*$ (thin black lines). Each mini-triangle n has a barycentric coordinate system $\underline{\tilde{\boldsymbol{\tau}}} = (\tilde{\tau}_1, \tilde{\tau}_2, \tilde{\tau}_3)$ and Bézier ordinates $b_{r,s,t}$, cf. Figure 4(d).

For each vertex k, its Powell-Sabin points are the vertex k itself and the midpoints of all edges of the Powell-Sabin refinement $\mathcal{T}^*$ containing the vertex k. A Powell-Sabin triangle (shown in red), which contains all Powell-Sabin points (denoted by green dots) is associated with each vertex k. The Powell-Sabin triangles are chosen such that they share two edges with the convex hull of the

DOI: 10.1002/nme

Powell-Sabin points [18, 27]. In this way the solution of an expensive optimisation algorithm is avoided [15].

Three ($j = 1, 2, 3$) Powell-Sabin B-splines $N_k^j(\underline{\boldsymbol{\xi}})$ are associated to each vertex k, i. e. one for each corner of the Powell-Sabin triangle of vertex k, and have a support in the molecule Ω_k of $\underline{\boldsymbol{V}}_k$. For an elaborate demonstration for the derivation of the calculation of the Powell-Sabin B-splines $N_k^j(\underline{\boldsymbol{\xi}})$ reference is made to [15]. Here, we only give a brief outline of the procedure.

A Powell-Sabin B-spline $N_k^j(\underline{\boldsymbol{\xi}})$ in the parameter domain $\underline{\boldsymbol{\xi}}$ must be defined in each of the m_k triangles of the molecule Ω_k of vertex k,

$$N_k^j(\underline{\boldsymbol{\xi}}) = N_k^j\big(\underline{\boldsymbol{\xi}}(\underline{\boldsymbol{\tau}})\big) = \sum_{e_k=1}^{m_k} N_k^{j,e_k}(\underline{\boldsymbol{\tau}}). \tag{25}$$

Since each triangle of the molecule Ω_k is split into six mini-triangles, the Powell-Sabin B-splines $N_k^{j,e_k}(\underline{\boldsymbol{\tau}})$ over a triangle e_k can be written as:

$$N_k^{j,e_k}(\underline{\boldsymbol{\tau}}) = N_k^{j,e_k}\big(\underline{\boldsymbol{\tau}}(\underline{\tilde{\boldsymbol{\tau}}})\big) = \sum_{n=1}^{6} N_{k,n}^{j,e_k}(\underline{\tilde{\boldsymbol{\tau}}}). \tag{26}$$

For clarity of notation, the indices ${}^{e_k}_{n}$ are omitted in the following. The Powell-Sabin B-splines over each mini-triangle, $N_k^j(\underline{\tilde{\boldsymbol{\tau}}})$ in Equation (26), can be expressed using the Bézier ordinates $b_{r,s,t}$, see Figure 4(d),

$$N_k^j(\underline{\tilde{\boldsymbol{\tau}}}) = \sum_{r+s+t=2} b_{r,s,t} B_{r,s,t}^2(\underline{\tilde{\boldsymbol{\tau}}}), \tag{27}$$

where $B_{r,s,t}^2(\underline{\tilde{\boldsymbol{\tau}}})$ denote the Bernstein polynomials of degree two:

$$B_{r,s,t}^2(\underline{\tilde{\boldsymbol{\tau}}}) = \frac{2!}{r!s!t!}\tilde{\tau}_1^r\tilde{\tau}_2^s\tilde{\tau}_3^t. \tag{28}$$

In order to determine the Bézier ordinates $b_{r,s,t}$ in Equation (27), the following properties are assigned to the Powell-Sabin B-splines: For any vertex $l \neq k$ we have

$$N_k^j(V_l) = 0, \quad \frac{\partial}{\partial \xi}N_k^j(V_l) = 0, \quad \frac{\partial}{\partial \eta}N_k^j(V_l) = 0, \tag{29}$$

and otherwise

$$N_k^j(V_k) = \alpha_k^j, \quad \frac{\partial}{\partial \xi}N_k^j(V_k) = \beta_k^j, \quad \frac{\partial}{\partial \eta}N_k^j(V_k) = \gamma_k^j \tag{30}$$

with

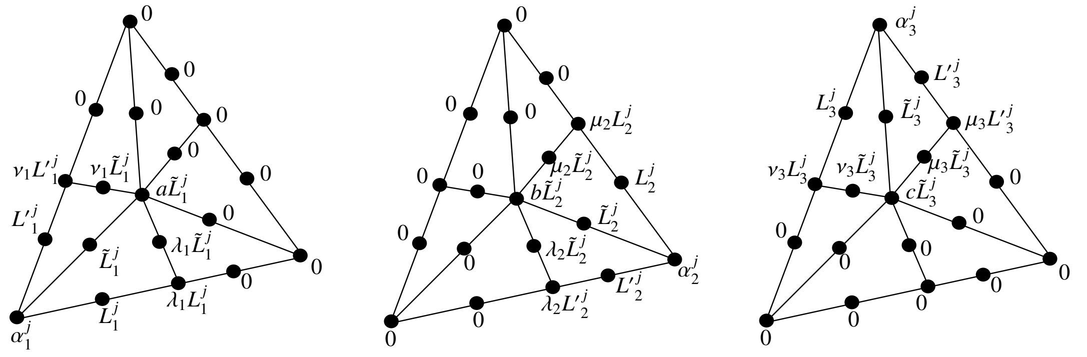
$$\sum_{j=1}^{3}\alpha_k^j = 1, \quad \sum_{j=1}^{3}\beta_k^j = 0, \quad \sum_{j=1}^{3}\gamma_k^j = 0. \tag{31}$$

The corners of each Powell-Sabin triangle (red in Figure 4(a)) have the coordinates $\underline{\boldsymbol{Q}}_k^j = (\xi_k^j, \eta_k^j)$, which gives the map from the triangle domain $\underline{\boldsymbol{\tau}}$ to the parameter domain $\underline{\boldsymbol{\xi}}$ for a surface $\underline{\boldsymbol{S}}_\xi(\underline{\boldsymbol{\tau}})$, as follows

$$\underline{\boldsymbol{S}}_\xi(\underline{\boldsymbol{\tau}}) = \sum_{k=1}^{N_v}\sum_{j=1}^{3} N_k^j(\underline{\boldsymbol{\tau}})\underline{\boldsymbol{Q}}_k^j. \tag{32}$$

For each vertex k with coordinates (ξ_k, η_k) the parameters α_k^j, β_k^j and γ_k^j in Equation (31) are then obtained by solving:

$$\begin{bmatrix} \alpha_k^1 & \alpha_k^2 & \alpha_k^3 \\ \beta_k^1 & \beta_k^2 & \beta_k^3 \\ \gamma_k^1 & \gamma_k^2 & \gamma_k^3 \end{bmatrix} \begin{bmatrix} \xi_k^1 & \eta_k^1 & 1 \\ \xi_k^2 & \eta_k^2 & 1 \\ \xi_k^3 & \eta_k^3 & 1 \end{bmatrix} = \begin{bmatrix} \xi_k & \eta_k & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{33}$$



(a) Bézier ordinates for the three Powell-Sabin B-splines $N^j_1(\underline{\tilde{\boldsymbol{\tau}}})$ corresponding to vertex V_1

(b) Bézier ordinates for the three Powell-Sabin B-splines $N^j_2(\underline{\tilde{\boldsymbol{\tau}}})$ corresponding to vertex V_2

(c) Bézier ordinates for the three Powell-Sabin B-splines $N^j_3(\underline{\tilde{\boldsymbol{\tau}}})$ corresponding to vertex V_3

Figure 5. Bézier ordinates $b_{r,s,t}$ for the six $(n = 1, 2, \dots, 6)$ mini-triangles of the Powell-Sabin B-splines $N^j_1(\underline{\tilde{\boldsymbol{\tau}}})$, $N^j_2(\underline{\tilde{\boldsymbol{\tau}}})$ and $N^j_3(\underline{\tilde{\boldsymbol{\tau}}})$.

This equation follows by combining Equation (30), Equation (31) and Equation (32).

Using Equation (24) and Equation (33), the Bézier ordinates $b_{r,s,t}$ of the mini-triangles in Figure 5 can be evaluated. The Bézier ordinates corresponding to V_1 are given by:

$$L^j_1 = \alpha^j_1 + \frac{1-\lambda_1}{2}\bar{\beta}^j_1, \quad L'^j_1 = \alpha^j_1 + \frac{(1-\nu_1)}{2}\bar{\gamma}^j_1, \quad \tilde{L}^j_1 = \alpha^j_1 + \frac{b}{2}\bar{\beta}^j_1 + \frac{c}{2}\bar{\gamma}^j_1, \tag{34}$$

$$\bar{\beta}^j_1 = \beta^j_1(\xi_2 - \xi_1) + \gamma^j_1(\eta_2 - \eta_1), \quad \bar{\gamma}^j_1 = \beta^j_1(\xi_3 - \xi_1) + \gamma^j_1(\eta_3 - \eta_1), \tag{35}$$

while for V_2 they read,

$$L^j_2 = \alpha^j_2 + \frac{1-\mu_2}{2}\bar{\beta}^j_2, \quad L'^j_2 = \alpha^j_2 + \frac{(1-\lambda_2)}{2}\bar{\gamma}^j_2, \quad \tilde{L}^j_2 = \alpha^j_2 + \frac{c}{2}\bar{\beta}^j_2 + \frac{a}{2}\bar{\gamma}^j_2, \tag{36}$$

$$\bar{\beta}^j_2 = \beta^j_2(\xi_3 - \xi_2) + \gamma^j_2(\eta_3 - \eta_2), \quad \bar{\gamma}^j_2 = \beta^j_2(\xi_1 - \xi_2) + \gamma^j_2(\eta_1 - \eta_2) \tag{37}$$

and for V_3 we have:

$$L^j_3 = \alpha^j_3 + \frac{1-\nu_3}{2}\bar{\beta}^j_3, \quad L'^j_3 = \alpha^j_3 + \frac{(1-\mu_3)}{2}\bar{\gamma}^j_3, \quad \tilde{L}^j_3 = \alpha^j_3 + \frac{a}{2}\bar{\beta}^j_3 + \frac{b}{2}\bar{\gamma}^j_3, \tag{38}$$

$$\bar{\beta}^j_3 = \beta^j_3(\xi_1 - \xi_3) + \gamma^j_3(\eta_1 - \eta_3), \quad \bar{\gamma}^j_3 = \beta^j_3(\xi_2 - \xi_3) + \gamma^j_3(\eta_2 - \eta_3). \tag{39}$$

For example, the Powell-Sabin B-splines $N^j_k(\underline{\tilde{\boldsymbol{\tau}}})$ in the mini-triangle $n = 3$ of Figure 4(d) can be expressed as

$$\begin{bmatrix} N^1_1(\underline{\tilde{\boldsymbol{\tau}}}) \\ N^2_1(\underline{\tilde{\boldsymbol{\tau}}}) \\ N^3_1(\underline{\tilde{\boldsymbol{\tau}}}) \\ N^1_2(\underline{\tilde{\boldsymbol{\tau}}}) \\ N^2_2(\underline{\tilde{\boldsymbol{\tau}}}) \\ N^3_2(\underline{\tilde{\boldsymbol{\tau}}}) \\ N^1_3(\underline{\tilde{\boldsymbol{\tau}}}) \\ N^2_3(\underline{\tilde{\boldsymbol{\tau}}}) \\ N^3_3(\underline{\tilde{\boldsymbol{\tau}}}) \end{bmatrix} = \begin{bmatrix} a\tilde{L}^1_1 & 0 & 0 & 0 & 0 & 0 \\ a\tilde{L}^2_1 & 0 & 0 & 0 & 0 & 0 \\ a\tilde{L}^3_1 & 0 & 0 & 0 & 0 & 0 \\ b\tilde{L}^1_2 & \tilde{L}^1_2 & \mu_2\tilde{L}^1_2 & \alpha^1_2 & L^1_2 & \mu_2 L^1_2 \\ b\tilde{L}^2_2 & \tilde{L}^2_2 & \mu_2\tilde{L}^2_2 & \alpha^2_2 & L^2_2 & \mu_2 L^2_2 \\ b\tilde{L}^3_2 & \tilde{L}^3_2 & \mu_2\tilde{L}^3_2 & \alpha^3_2 & L^3_2 & \mu_2 L^3_2 \\ c\tilde{L}^1_3 & 0 & \mu_3\tilde{L}^1_3 & 0 & 0 & \mu_3 L'^1_3 \\ c\tilde{L}^2_3 & 0 & \mu_3\tilde{L}^2_3 & 0 & 0 & \mu_3 L'^2_3 \\ c\tilde{L}^3_3 & 0 & \mu_3\tilde{L}^3_3 & 0 & 0 & \mu_3 L'^3_3 \end{bmatrix} \begin{bmatrix} B^2_{200}(\underline{\tilde{\boldsymbol{\tau}}}) \\ B^2_{110}(\underline{\tilde{\boldsymbol{\tau}}}) \\ B^2_{101}(\underline{\tilde{\boldsymbol{\tau}}}) \\ B^2_{020}(\underline{\tilde{\boldsymbol{\tau}}}) \\ B^2_{011}(\underline{\tilde{\boldsymbol{\tau}}}) \\ B^2_{002}(\underline{\tilde{\boldsymbol{\tau}}}) \end{bmatrix}, \tag{40}$$

or in matrix-vector format:

$$\underline{\boldsymbol{N}}^e_n(\underline{\tilde{\boldsymbol{\tau}}}) = \underline{\underline{\mathbf{C}}}^e_n \underline{\boldsymbol{B}}(\underline{\tilde{\boldsymbol{\tau}}}), \tag{41}$$

DOI: 10.1002/nme

where the Bézier extraction operator $\underline{\underline{\mathbf{C}}}^e_n$ in Equation (40) contains the Bézier ordinates $b_{r,s,t}$ for each Powell-Sabin B-spline $N_k^j(\underline{\tilde{\boldsymbol{\tau}}})$ in the mini-triangle $n = 3$ of triangle e from Figure 4(c),(d). Hence, it is possible to apply the Bézier extraction procedure to Powell-Sabin splines, in a fashion similar to NURBS and T-splines [20, 21].

Non-Uniform Rational Powell-Sabin B-splines (NURPS) $R_k^j(\underline{\boldsymbol{\tau}})$ can be computed as follows:

$$R_k^j(\underline{\boldsymbol{\tau}}) = \frac{w_k^j N_k^j(\underline{\boldsymbol{\tau}})}{\sum_{\hat{k}=1}^{N_v} \sum_{\hat{j}=1}^{3} w_{\hat{k}}^{\hat{j}} N_{\hat{k}}^{\hat{j}}(\underline{\boldsymbol{\tau}})} \tag{42}$$

with w_k^j the weights associated with each corner $\underline{\boldsymbol{Q}}_k^j$ of a Powell-Sabin triangle. A NURPS-surface $\underline{\boldsymbol{S}}(\underline{\boldsymbol{\tau}})$ in the physical domain $\underline{\boldsymbol{x}}$ can be expressed by a map from the triangle domain $\underline{\boldsymbol{\tau}}$ with

$$\underline{\boldsymbol{S}}(\underline{\boldsymbol{\tau}}) = \sum_{k=1}^{N_v} \sum_{j=1}^{3} R_k^j(\underline{\boldsymbol{\tau}}) \underline{\boldsymbol{P}}_k^j \tag{43}$$

where the control points $\underline{\boldsymbol{P}}_k^j$ correspond to each $\underline{\boldsymbol{Q}}_k^j$, see also Figure 4(b). In matrix-vector format, the NURPS $\underline{\boldsymbol{R}}_n^e$ for one mini-triangle n of triangle e is obtained from

$$\underline{\boldsymbol{R}}_n^e(\underline{\tilde{\boldsymbol{\tau}}}) = \underline{\underline{\mathbf{W}}}^e \underline{\underline{\mathbf{C}}}^e_n \frac{\underline{\boldsymbol{B}}(\underline{\tilde{\boldsymbol{\tau}}})}{W^{be}_{\;n}(\underline{\tilde{\boldsymbol{\tau}}})} \tag{44}$$

with

$$W^{be}_{\;n}(\underline{\tilde{\boldsymbol{\tau}}}) = (\underline{\boldsymbol{w}}^{be}_{\;n})^T \underline{\boldsymbol{B}}(\underline{\tilde{\boldsymbol{\tau}}}), \quad \underline{\boldsymbol{w}}^{be}_{\;n} = (\underline{\underline{\mathbf{C}}}^e_n)^T \underline{\boldsymbol{w}}^e, \quad \underline{\underline{\mathbf{W}}}^e = \mathrm{diag}(\underline{\boldsymbol{w}}^e) \tag{45}$$

where $\underline{\boldsymbol{w}}^e$ is the vector containing the weights of triangle e. The derivatives with respect to coordinates in the domain of the mini-triangle, $\underline{\tilde{\boldsymbol{\tau}}}$, read:

$$\frac{\partial \underline{\boldsymbol{R}}_n^e(\underline{\tilde{\boldsymbol{\tau}}})}{\partial \tilde{\tau}_i} = \underline{\underline{\mathbf{W}}}^e \underline{\underline{\mathbf{C}}}^e_n \frac{\partial}{\partial \tilde{\tau}_i} \left(\frac{\underline{\boldsymbol{B}}(\underline{\tilde{\boldsymbol{\tau}}})}{W^{be}_{\;n}(\underline{\tilde{\boldsymbol{\tau}}})} \right) = \underline{\underline{\mathbf{W}}}^e \underline{\underline{\mathbf{C}}}^e_n \left(\frac{1}{W^{be}_{\;n}(\underline{\tilde{\boldsymbol{\tau}}})} \frac{\partial \underline{\boldsymbol{B}}(\underline{\tilde{\boldsymbol{\tau}}})}{\partial \tilde{\tau}_i} - \frac{\partial W^{be}_{\;n}(\underline{\tilde{\boldsymbol{\tau}}})}{\partial \tilde{\tau}_i} \frac{\underline{\boldsymbol{B}}(\underline{\tilde{\boldsymbol{\tau}}})}{\left(W^{be}_{\;n}(\underline{\tilde{\boldsymbol{\tau}}})\right)^2} \right). \tag{46}$$

The derivatives in the physical domain $\underline{\boldsymbol{x}}$ are subsequently obtained as:

$$\frac{\partial \underline{\boldsymbol{R}}_n^e(\underline{\tilde{\boldsymbol{\tau}}})}{\partial x_i^e} = \sum_{j=1}^{2} \frac{\partial \underline{\boldsymbol{R}}_n^e(\underline{\tilde{\boldsymbol{\tau}}})}{\partial \tilde{\tau}_j} \frac{\partial \tilde{\tau}_j}{\partial x_i^e} \tag{47}$$

where

$$\frac{\partial \tilde{\tau}_j}{\partial x_i^e} = \begin{bmatrix} \frac{\partial x^e}{\partial \tilde{\tau}_1} & \frac{\partial x^e}{\partial \tilde{\tau}_2} \\ \frac{\partial y^e}{\partial \tilde{\tau}_1} & \frac{\partial y^e}{\partial \tilde{\tau}_2} \end{bmatrix}^{-1} = \underline{\underline{\mathbf{J}}}^{-1}, \tag{48}$$

with the Jacobian matrix $\underline{\underline{\mathbf{J}}}$ of the geometry mapping. To further illustrate the procedure, a Matlab code snippet is given in Appendix A for the Bézier extraction procedure for NURPS, which computes the second derivatives in the global coordinate system $\underline{\boldsymbol{x}}$ and assembles the stiffness matrix $\underline{\underline{\mathbf{K}}}$ and the force vector $\underline{\boldsymbol{f}}$ in Equation (23).

It was suggested in [17] to evaluate the integrals for each component of the stiffness matrix in Equation (23) analytically since the integral of a Powell-Sabin B-spline $N_k^j(\underline{\tilde{\boldsymbol{\tau}}})$ over a mini-triangle n with area A_n can be computed as [28]:

$$\int_{\Omega_n} N_k^j(\underline{\tilde{\boldsymbol{\tau}}}) \mathrm{d}\tilde{\tau}_1 \mathrm{d}\tilde{\tau}_2 = \frac{A_n}{6} \sum_{r+s+t=2} b_{r,s,t}. \tag{49}$$

Relations for the computation of the product of the derivatives of a Powell-Sabin B-spline can then be derived accordingly [17]. However, this is computationally more expensive than using the Bézier extraction procedure.

5. UNSTRUCTURED QUADRATIC T-SPLINES

This section addresses unstructured T-spline meshes of quadratic degree [29]. An unstructured T-spline mesh contains points with valence three or more than four. These points are called star, irregular or extraordinary points. Without special treatment, these meshes are non-standard and only $\mathcal{C}^0_\mathrm{A}$-continuous around the extraordinary points. The blending functions of a non-standard T-spline mesh do not fulfil the partition of unity property [30]. It was demonstrated in [31] how this can be detected utilising the global Bézier extraction operator. Herein, it will be demonstrated how these meshes can be manipulated such that they are standard (blending functions N^i fulfil the partition of unity property) and $\mathcal{C}^1_\mathrm{A}$-continuous around the extraordinary points.

The technique presented here is very similar to that in [11]. However, the approach in [11] does not fulfil the partition of unity property for the blending function N^i in the one-ring neighbourhood elements of an extraordinary point. Herein it is demonstrated how this deficiency can be repaired.

5.1. *The unstructured T-spline mesh*

Figure 6 shows an unstructured quadratic T-spline mesh. For definition of the terms anchors, edges and elements, reference is made to [31].

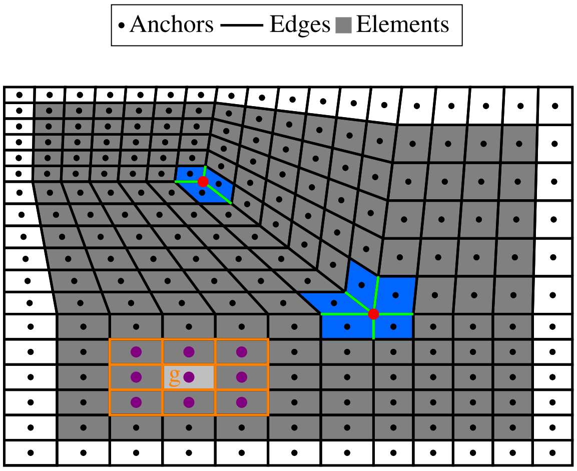


Figure 6. Example for an unstructured quadratic T-spline mesh. All non-zero knot intervals are assumed to be uniform. The two extraordinary points of valence three and five are marked with red. Spoke edges (green) touch an extraordinary point. Anchors with support in element g (light grey) are marked violet.

Extraordinary points are indicated with a red circle. Spoke edges are marked with green: they touch an extraordinary point. In order to build the Bézier extraction operator for each purple anchor that has a support in the light grey element g, the knot intervals of the neighbouring rectangles (marked orange) for element g are required. Some anchors with support in g do not require all their individual knot intervals in order to determine their Bézier extraction operator in g. This construction cannot be applied to the blue elements in the one-ring neighbourhood of the extraordinary points. These elements are called irregular elements, whereas the other elements are regular elements. For the irregular elements, generalised Bézier extraction will be utilised. Generalised Bézier extraction defines the transpose of the Bézier extraction operator, $\underline{\underline{\mathbf{C}}}_e^T$. The blending functions $\underline{\boldsymbol{N}}_e$ in an element of a T-spline mesh can be expressed as a linear combination of the Bernstein polynomials utilising the Bézier extraction operator

$$\underline{\boldsymbol{N}}_e = \underline{\underline{\mathbf{C}}}_e \underline{\boldsymbol{B}}_e, \tag{50}$$

where the vector $\underline{\boldsymbol{B}}_e$ contains the bivariate Bernstein polynomials. The $a = 1 \ldots p+1$ univariate Bernstein polynomials B^a_ℓ of order p_ℓ are defined over the interval $\tilde{\xi}_\ell \in [-1, 1]$ by

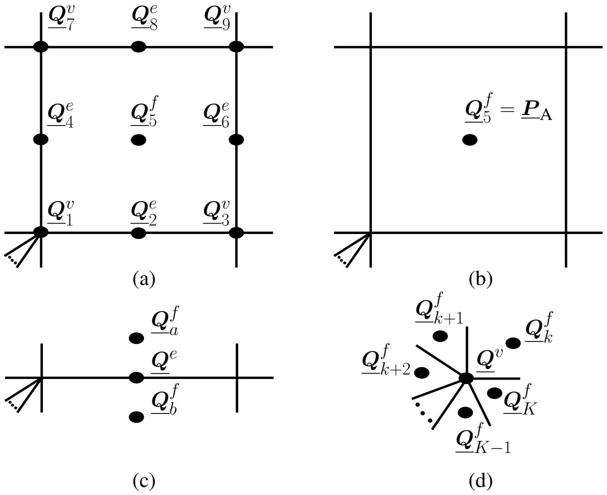
$$B^a_\ell(\tilde{\xi}_\ell) = \frac{1}{2^{p_\ell}} \binom{p_\ell}{a-1} (1-\tilde{\xi}_\ell)^{p_\ell-(a-1)}(1+\tilde{\xi}_\ell)^{a-1}. \tag{51}$$

5.2. Generalised Bézier extraction

Generalised Bézier extraction yields a relation between Bézier control points $\underline{\boldsymbol{Q}}_e$ and the control points $\underline{\boldsymbol{P}}_e$ with support in element e

$$\underline{\boldsymbol{Q}}_e = \underline{\underline{\mathbf{C}}}^T_e \underline{\boldsymbol{P}}_e. \tag{52}$$

Each quadratic element has nine Bézier control points – one face point $\underline{\boldsymbol{Q}}^f_5$, four edge points $\underline{\boldsymbol{Q}}^e_2$, $\underline{\boldsymbol{Q}}^e_4$, $\underline{\boldsymbol{Q}}^e_6$, $\underline{\boldsymbol{Q}}^e_8$ and four vertex points $\underline{\boldsymbol{Q}}^v_1$, $\underline{\boldsymbol{Q}}^v_3$, $\underline{\boldsymbol{Q}}^v_7$, $\underline{\boldsymbol{Q}}^v_9$ as depicted in Figure 7(a). Herein, it is assumed that all non-zero knot intervals are uniform. The general case for non-uniform knot-intervals is considered in [11] for the cubic case.


$\underline{\boldsymbol{Q}}^v_7$
$\underline{\boldsymbol{Q}}^e_8$
$\underline{\boldsymbol{Q}}^v_9$
$\underline{\boldsymbol{Q}}^e_4$
$\underline{\boldsymbol{Q}}^f_5$
$\underline{\boldsymbol{Q}}^e_6$
$\underline{\boldsymbol{Q}}^v_1$
$\underline{\boldsymbol{Q}}^e_2$
$\underline{\boldsymbol{Q}}^v_3$
$\underline{\boldsymbol{Q}}^f_5 = \underline{\boldsymbol{P}}_{\mathrm{A}}$
$\underline{\boldsymbol{Q}}^f_a$
$\underline{\boldsymbol{Q}}^e$
$\underline{\boldsymbol{Q}}^f_b$
$\underline{\boldsymbol{Q}}^f_{k+1}$
$\underline{\boldsymbol{Q}}^f_k$
$\underline{\boldsymbol{Q}}^f_{k+2}$
$\underline{\boldsymbol{Q}}^v$
$\underline{\boldsymbol{Q}}^f_K$
$\underline{\boldsymbol{Q}}^f_{K-1}$

(a) (b)

(c) (d)

Figure 7. (a) The nine Bézier control points $\underline{\boldsymbol{Q}}_e$ for a Bézier element. (b) The control point $\underline{\boldsymbol{P}}_A$ corresponds to the anchor in this element. (c) An edge Bézier control point $\underline{\boldsymbol{Q}}^e$ is written in terms of neighbouring face Bézier control points $\underline{\boldsymbol{Q}}^f$ in Equation (54). (d) A vertex Bézier control point $\underline{\boldsymbol{Q}}^v$ is written in terms of neighbouring face Bézier control points $\underline{\boldsymbol{Q}}^f$ in Equation (55).

The face point $\underline{\boldsymbol{Q}}^f_5$ (cf. Figure 7(b)) is determined as

$$\underline{\boldsymbol{Q}}^f_5 = \underline{\boldsymbol{P}}_{\mathrm{A}}, \tag{53}$$

where $\underline{\boldsymbol{P}}_{\mathrm{A}}$ denotes the control point coordinate of anchor A. The edge vertex $\underline{\boldsymbol{Q}}^e$ in Figure 7(c) is computed with

$$\underline{\boldsymbol{Q}}^e = \frac{1}{2}\underline{\boldsymbol{Q}}^f_a + \frac{1}{2}\underline{\boldsymbol{Q}}^f_b, \tag{54}$$

and the vertex point $\underline{\boldsymbol{Q}}^v$ of Figure 7(d) is obtained utilising

$$\underline{\boldsymbol{Q}}^v = \sum_{k=1}^{K} \frac{1}{4}\underline{\boldsymbol{Q}}^f_k, \tag{55}$$

where it was assumed that $\underline{\boldsymbol{Q}}^v$ is the vertex of K elements.

After computing $\underline{\underline{\mathbf{C}}}^T_e$ exploiting the generalised Bézier extraction for the one-ring neighbourhood elements, the T-spline mesh is non-standard, i. e. the blending functions N do not fulfil the partition of unity property. Also, the Bézier extraction operators are non-square matrices for the elements in the one-ring neighbourhood of an extraordinary point. $\underline{\underline{\mathbf{C}}}_e$ has full row rank for the one-ring neighbourhood elements of an extraordinary point with valence three – the blending functions are locally linearly independent. For the one-ring neighbourhood elements of an extraordinary point with valence five, $\underline{\underline{\mathbf{C}}}_e$ does not have full row rank and therefore, the blending functions are locally linearly dependent. Along spoke edges there is $\mathcal{C}^0_{\mathrm{A}}$-continuity. Furthermore, the one-ring neighbourhood elements of the extraordinary point (the irregular elements) are $\mathcal{C}^1_{\mathrm{A}}$-continuous with the two-ring neighbourhood elements of the extraordinary points. Next, it will be explained how to enforce $\mathcal{C}^1_{\mathrm{A}}$-continuity along spoke edges while preserving $\mathcal{C}^1_{\mathrm{A}}$-continuity between the one- and two-ring neighbourhood elements. Moreover, the partition of unity property of the blending functions N will be fulfilled.

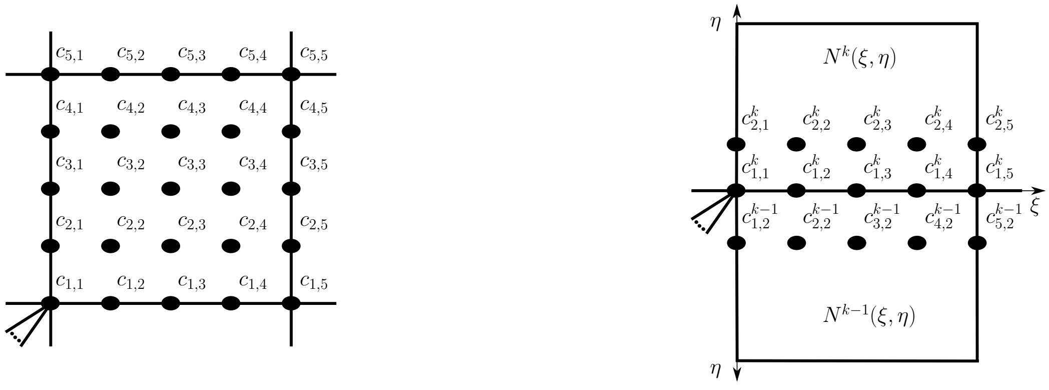
5.3. *Modifying the Bézier extraction operator*

This section shows how to modify the coefficients of the Bézier extraction operator $\underline{\underline{\mathbf{C}}}_e$ in the one-ring neighbourhood elements of an extraordinary point. In a first step, the Bézier extraction operator $\underline{\underline{\mathbf{C}}}_e$ is elevated from degree two to degree four. This is achieved by utilising the degree elevation matrix $\underline{\underline{\mathbf{E}}}^{2,4}$, cf. [32]. Degree elevating the blending functions $\underline{\boldsymbol{N}}_e$ in Equation (50) with support in element e results in

$$\underline{\boldsymbol{N}}_e = \underline{\underline{\mathbf{C}}}^2_e \underline{\boldsymbol{B}}^2_e = \underline{\underline{\mathbf{C}}}^2_e \underline{\underline{\mathbf{E}}}^{2,3} \underline{\boldsymbol{B}}^3_e = \underline{\underline{\mathbf{C}}}^2_e \underline{\underline{\mathbf{E}}}^{2,3} \underline{\underline{\mathbf{E}}}^{3,4} \underline{\boldsymbol{B}}^4_e = \underline{\underline{\mathbf{C}}}^2_e \underline{\underline{\mathbf{E}}}^{2,4} \underline{\boldsymbol{B}}^4_e = \underline{\underline{\mathbf{C}}}^4_e \underline{\boldsymbol{B}}^4_e, \tag{56}$$

where the superscript was added in order to indicate the degree. It can be observed from Equation (56) that degree elevation does not change the blending functions $\underline{\boldsymbol{N}}_e$. Also, degree elevation does not change local dependencies that may exist, i. e. the row rank of $\underline{\underline{\mathbf{C}}}_e$ is not affected. After degree elevation, each blending function N with support over a Bézier element in the one-ring neighbourhood has 25 Bézier coefficients $c_{\alpha,\beta}$ (cf. Figure 8(a))

$$N(\underline{\boldsymbol{\xi}}) = \sum_{\alpha=1}^{5} \sum_{\beta=1}^{5} c_{\alpha,\beta} B_{\alpha,\beta}(\underline{\boldsymbol{\xi}}). \tag{57}$$


$c_{5,1}$ $c_{5,2}$ $c_{5,3}$ $c_{5,4}$ $c_{5,5}$
$c_{4,1}$ $c_{4,2}$ $c_{4,3}$ $c_{4,4}$ $c_{4,5}$
$c_{3,1}$ $c_{3,2}$ $c_{3,3}$ $c_{3,4}$ $c_{3,5}$
$c_{2,1}$ $c_{2,2}$ $c_{2,3}$ $c_{2,4}$ $c_{2,5}$
$c_{1,1}$ $c_{1,2}$ $c_{1,3}$ $c_{1,4}$ $c_{1,5}$
η
$N^k(\xi,\eta)$
$c^k_{2,1}$ $c^k_{2,2}$ $c^k_{2,3}$ $c^k_{2,4}$ $c^k_{2,5}$
$c^k_{1,1}$ $c^k_{1,2}$ $c^k_{1,3}$ $c^k_{1,4}$ $c^k_{1,5}$
ξ
$c^{k-1}_{1,2}$ $c^{k-1}_{2,2}$ $c^{k-1}_{3,2}$ $c^{k-1}_{4,2}$ $c^{k-1}_{5,2}$
$N^{k-1}(\xi,\eta)$
η

(a)

(b)

Figure 8. After degree elevation, a blending function with support in a one-ring neighbourhood element has 25 Bézier coefficients $c_{\alpha,\beta}$ in each one-ring neighbourhood element. (b) Bézier coefficients along a spoke edge that are involved in the $\mathcal{G}^1$-continuity constraint of Equation (62).

Assume that $a = 1 \ldots A$ blending functions N^a have a support over *at least two* of the $b = 1 \ldots B$ one-ring neighbourhood elements and $c = 1 \ldots C$ blending functions N^c have a support in *only one* one-ring neighbourhood elements. Now, the coefficients of the Bézier extraction operator of all blending functions N^a and N^c have to be perturbed. The initial coefficients of the Bézier extraction operator are denoted with $c^{a,b}_{\alpha,\beta}$, $c^{c,b}_{\alpha,\beta}$ and the modified coefficients with $\tilde{c}^{a,b}_{\alpha,\beta}$, $\tilde{c}^{c,b}_{\alpha,\beta}$. The following constrained linear least square problem needs to be solved for each extraordinary point

$$\min_{\underline{\tilde{\boldsymbol{c}}}\,\in\, S} \|\underline{\underline{\mathbf{F}}}\,\underline{\tilde{\boldsymbol{c}}} - \underline{\boldsymbol{f}}\|_2, \qquad S = \left\{ \underline{\tilde{\boldsymbol{c}}} \mid \|\underline{\underline{\mathbf{G}}}\,\underline{\tilde{\boldsymbol{c}}} - \underline{\boldsymbol{g}}\|_2 = \min \right\}, \tag{58}$$

with the fairing matrix $\underline{\underline{\mathbf{F}}}$, the fairing vector $\underline{\boldsymbol{f}}$, the constraint matrix $\underline{\underline{\mathbf{G}}}$ and the constraint vector $\underline{\boldsymbol{g}}$. The problem in Equation (58) can be transformed into an unconstrained linear least square problem [33].

The following constraints are assembled into $\underline{\underline{\mathbf{F}}}$ and $\underline{\boldsymbol{f}}$ in Equation (58)

$$\begin{aligned}
\tilde{c}^{a,b}_{\alpha,\beta} - \tilde{c}^{a,b}_{\alpha,\beta+1} &= c^{a,b}_{\alpha,\beta} - c^{a,b}_{\alpha,\beta+1} && \text{for } 1 \le \alpha \le 5,\, 1 \le \beta \le 4,\, 1 \le a \le A,\, 1 \le b \le B,\\
\tilde{c}^{a,b}_{\alpha,\beta} - \tilde{c}^{a,b}_{\alpha+1,\beta} &= c^{a,b}_{\alpha,\beta} - c^{a,b}_{\alpha+1,\beta} && \text{for } 1 \le \alpha \le 4,\, 1 \le \beta \le 5,\, 1 \le a \le A,\, 1 \le b \le B,\\
\tilde{c}^{c,b}_{\alpha,\beta} - \tilde{c}^{c,b}_{\alpha+1,\beta} &= c^{c,b}_{\alpha,\beta} - c^{c,b}_{\alpha+1,\beta} && \text{for } 1 \le \alpha \le 5,\, 1 \le \beta \le 4,\, 1 \le c \le C,\, 1 \le b \le B,\\
\tilde{c}^{c,b}_{\alpha,\beta} - \tilde{c}^{c,b}_{\alpha+1,\beta} &= c^{c,b}_{\alpha,\beta} - c^{c,b}_{\alpha+1,\beta} && \text{for } 1 \le \alpha \le 4,\, 1 \le \beta \le 5,\, 1 \le c \le C,\, 1 \le b \le B.
\end{aligned} \tag{59}$$

The fairing equations in Equation (59) prevent oscillations between neighbouring coefficients of the Bézier extraction operator when perturbing the coefficients $c_{\alpha,\beta}$.

Suppose that $\alpha = 1$, $\beta = 1$ marks the Bézier control point at the extraordinary point, see Figure 8(a). In order to preserve $\mathcal{C}^1_{\mathrm{A}}$-continuity between one and two-ring neighbourhood elements, the constraints

$$\begin{aligned}
\tilde{c}^{a,b}_{\alpha,\beta} &= c^{a,b}_{\alpha,\beta} && \text{for } 1 \le \alpha \le 5,\, 4 \le \beta \le 5,\, 1 \le a \le A,\, 1 \le b \le B,\\
\tilde{c}^{a,b}_{\alpha,\beta} &= c^{a,b}_{\alpha,\beta} && \text{for } 4 \le \alpha \le 5,\, 2 \le \beta \le 3,\, 1 \le a \le A,\, 1 \le b \le B
\end{aligned} \tag{60}$$

are assembled into $\underline{\underline{\mathbf{G}}}$ and $\underline{\boldsymbol{g}}$ in Equation (58) for blending functions that are non-zero in *at least two* one-ring neighbourhood elements. Blending functions that are non-zero in *only one* one-ring neighbourhood element are not allowed to change by enforcing

$$\tilde{c}^{c,b}_{\alpha,\beta} = c^{c,b}_{\alpha,\beta} \quad \text{for } 1 \le \alpha \le 5, 1 \le \beta \le 5,\, 1 \le c \le C,\, 1 \le b \le B. \tag{61}$$

In order to get $\mathcal{C}^1_{\mathrm{A}}$-continuity along spoke edges (between one-ring neighbourhood elements $k-1$ and k in Figure 8(b)) for the blending function N that is non-zero in *at least two* one-ring neighbourhood elements, the $\mathcal{G}^1$-continuity condition (see also [34, 35])

$$\begin{aligned}
f(\xi) = r(\xi)\frac{\partial N^{k-1}(\xi,\eta)}{\partial \eta}\bigg|_{\eta=0} + s(\xi)\frac{\partial N^{k}(\xi,\eta)}{\partial \xi}\bigg|_{\eta=0} + t(\xi)\frac{\partial N^{k}(\xi,\eta)}{\partial \eta}\bigg|_{\eta=0}&\\
= r(\xi)N^{k-1}_{,\eta}(\xi) + s(\xi)N^{k}_{,\xi}(\xi) + t(\xi)N^{k}_{,\eta}(\xi) = 0&
\end{aligned} \tag{62}$$

can be exploited since it was pointed out in [12] that this $\mathcal{G}^1$-condition yields $\mathcal{C}^1_{\mathrm{A}}$-continuous blending functions (see also Section 2.4).

In the following, the notation

$$\sum_{i=1}^{p+1} c_i B^p_i(\xi) = \langle c_1, c_2, \ldots, c_{p+1}\rangle^p(\xi) \tag{63}$$

will be used. The polynomials $r(\xi)$, $s(\xi)$ and $t(\xi)$ in Equation (62) are taken as

$$r(\xi) = 1, \qquad s(\xi) = \langle \zeta, 0, 0\rangle^2(\xi), \qquad t(\xi) = 1, \tag{64}$$

where ζ is computed from

$$\zeta = -2\cos(\theta), \quad \theta = \frac{2\pi}{B}. \tag{65}$$

Using Equations (63) and (64) the three terms in Equation (62) can be reworked as:

$$r(\xi)N^{k-1}_{,\eta}(\xi) = 4\langle \tilde{c}^{k-1}_{1,2} - \tilde{c}^{k}_{1,1}, \tilde{c}^{k-1}_{2,2} - \tilde{c}^{k}_{1,2}, \tilde{c}^{k-1}_{3,2} - \tilde{c}^{k}_{1,3}, \tilde{c}^{k-1}_{4,2} - \tilde{c}^{k}_{1,4}, \tilde{c}^{k-1}_{5,2} - \tilde{c}^{k}_{1,5}\rangle^{4}(\xi), \tag{66}$$

$$s(\xi)N^{k}_{,\xi}(\xi) = \langle \zeta, 0, 0\rangle^{2}(\xi) 4\langle \tilde{c}^{k}_{1,2} - \tilde{c}^{k}_{1,1}, \tilde{c}^{k}_{1,3} - \tilde{c}^{k}_{1,2}, \tilde{c}^{k}_{1,4} - \tilde{c}^{k}_{1,3}, \tilde{c}^{k}_{1,5} - \tilde{c}^{k}_{1,4}\rangle^{3}(\xi), \tag{67}$$

$$t(\xi)N^{k}_{,\eta}(\xi) = 4\langle \tilde{c}^{k}_{2,1} - \tilde{c}^{k}_{1,1}, \tilde{c}^{k}_{2,2} - \tilde{c}^{k}_{1,2}, \tilde{c}^{k}_{2,3} - \tilde{c}^{k}_{1,3}, \tilde{c}^{k}_{2,4} - \tilde{c}^{k}_{1,4}, \tilde{c}^{k}_{2,5} - \tilde{c}^{k}_{1,5}\rangle^{4}(\xi). \tag{68}$$

Since the term in Equation (67) is quintic, the degree of $N^{k}_{,\xi}$ needs to be reduced using the transformation matrix $\underline{\underline{\mathbf{D}}}^{3,2}$ [32]

$$\underline{\underline{\mathbf{D}}}^{3,2} = \left(\underline{\underline{\mathbf{E}}}^{2,3}\right)^{T} \left[\underline{\underline{\mathbf{E}}}^{2,3} \left(\underline{\underline{\mathbf{E}}}^{2,3}\right)^{T}\right]^{-1}. \tag{69}$$

Reducing the degree of $N^{k}_{,\xi}(\xi)$ in Equation (67) then results in

$$\begin{aligned} N^{k}_{,\xi}(\xi) = 4\langle &-\frac{19}{20}\tilde{c}^{k}_{1,1} + \frac{16}{20}\tilde{c}^{k}_{1,2} + \frac{6}{20}\tilde{c}^{k}_{1,3} - \frac{4}{20}\tilde{c}^{k}_{1,4} + \frac{1}{20}\tilde{c}^{k}_{1,5}, \\ &\frac{5}{20}\tilde{c}^{k}_{1,1} - \tilde{c}^{k}_{1,2} + \tilde{c}^{k}_{1,4} - \frac{5}{20}\tilde{c}^{k}_{1,5}, \\ &-\frac{1}{20}\tilde{c}^{k}_{1,1} + \frac{4}{20}\tilde{c}^{k}_{1,2} - \frac{6}{20}\tilde{c}^{k}_{1,3} - \frac{16}{20}\tilde{c}^{k}_{1,4} + \frac{19}{20}\tilde{c}^{k}_{1,5}\rangle^{2}(\xi), \end{aligned} \tag{70}$$

which, for Equation (67), gives:

$$\begin{aligned} s(\xi)N^{k}_{,\xi}(\xi) = 4\zeta\langle &-\frac{19}{20}\tilde{c}^{k}_{1,1} + \frac{16}{20}\tilde{c}^{k}_{1,2} + \frac{6}{20}\tilde{c}^{k}_{1,3} - \frac{4}{20}\tilde{c}^{k}_{1,4} + \frac{1}{20}\tilde{c}^{k}_{1,5}, \\ &\frac{5}{40}\tilde{c}^{k}_{1,1} - \frac{1}{2}\tilde{c}^{k}_{1,2} + \frac{1}{2}\tilde{c}^{k}_{1,4} - \frac{5}{40}\tilde{c}^{k}_{1,5}, \\ &-\frac{1}{120}\tilde{c}^{k}_{1,1} + \frac{4}{120}\tilde{c}^{k}_{1,2} - \frac{6}{120}\tilde{c}^{k}_{1,3} - \frac{16}{120}\tilde{c}^{k}_{1,4} + \frac{19}{120}\tilde{c}^{k}_{1,5}, 0, 0\rangle^{4}(\xi). \end{aligned} \tag{71}$$

In order to satisfy Equation (62), Equations (66), (68) and (71) are exploited to make vanish the following terms:

$$\tilde{c}^{k-1}_{1,2} - \tilde{c}^{k}_{1,1} + \zeta\left(-\frac{19}{20}\tilde{c}^{k}_{1,1} + \frac{16}{20}\tilde{c}^{k}_{1,2} + \frac{6}{20}\tilde{c}^{k}_{1,3} - \frac{4}{20}\tilde{c}^{k}_{1,4} + \frac{1}{20}\tilde{c}^{k}_{1,5}\right) + \tilde{c}^{k}_{2,1} - \tilde{c}^{k}_{1,1} = 0, \tag{72}$$

$$4\left(\tilde{c}^{k-1}_{2,2} - \tilde{c}^{k}_{1,2}\right) + \zeta\left(\frac{1}{2}\tilde{c}^{k}_{1,1} - 2\tilde{c}^{k}_{1,2} + 2\tilde{c}^{k}_{1,4} - \frac{1}{2}\tilde{c}^{k}_{1,5}\right) + 4\left(\tilde{c}^{k}_{2,2} - \tilde{c}^{k}_{1,2}\right) = 0, \tag{73}$$

$$4\left(\tilde{c}^{k-1}_{3,2} - \tilde{c}^{k}_{1,3}\right) + \zeta\left(-\frac{1}{30}\tilde{c}^{k}_{1,1} + \frac{4}{30}\tilde{c}^{k}_{1,2} - \frac{6}{30}\tilde{c}^{k}_{1,3} - \frac{16}{30}\tilde{c}^{k}_{1,4} + \frac{19}{30}\tilde{c}^{k}_{1,5}\right) + 4\left(\tilde{c}^{k}_{2,3} - \tilde{c}^{k}_{1,3}\right) = 0, \tag{74}$$

$$\tilde{c}^{k-1}_{4,2} - \tilde{c}^{k}_{1,4} + \tilde{c}^{k}_{2,4} - \tilde{c}^{k}_{1,4} = 0, \tag{75}$$

$$\tilde{c}^{k-1}_{5,2} - \tilde{c}^{k}_{1,5} + \tilde{c}^{k}_{2,5} - \tilde{c}^{k}_{1,5} = 0. \tag{76}$$

Moreover, the fourth derivative of $f(\xi)$ has to vanish. This constraint results in

$$\tilde{c}^{k}_{1,1} - 4\tilde{c}^{k}_{1,2} + 6\tilde{c}^{k}_{1,3} - 4\tilde{c}^{k}_{1,4} + \tilde{c}^{k}_{1,5} = 0. \tag{77}$$

Equations (72) – (77) need to be assembled in $\underline{\underline{\mathbf{G}}}$ and $\underline{\boldsymbol{g}}$ along all spoke edges for all blending functions that are non-zero in *at least two* one-ring neighbourhood elements.

In order to fulfil the partition of unity property [31], the following equation must be satisfied for all blending function that are non-zero in *at least one* one-ring neighbourhood element

$$\sum_{a=1}^{A}\tilde{c}^{a,b}_{\alpha,\beta} + \sum_{c=1}^{C}\tilde{c}^{c,b}_{\alpha,\beta} = 1 \quad \begin{cases} \text{for } 1 \le \alpha \le 3,\ 1 \le \beta \le 3 \\ \text{in all one-ring neighbourhood elements } b = 1\ldots B. \end{cases} \tag{78}$$

Computing the term

$$\|\underline{\underline{\mathbf{G}}}\,\underline{\tilde{\boldsymbol{c}}} - \underline{\boldsymbol{g}}\|_{2} \tag{79}$$

DOI: 10.1002/nme

after solving Equation (58) results in a computed zero (i.e. within machine precision), so that all the imposed constraints are satisfied.

Upon solution of the constrained linear least square problem in Equation (58), the support of the blending functions that are non-zero in *two* one-ring neighbourhood elements has changed, see Figure 9.

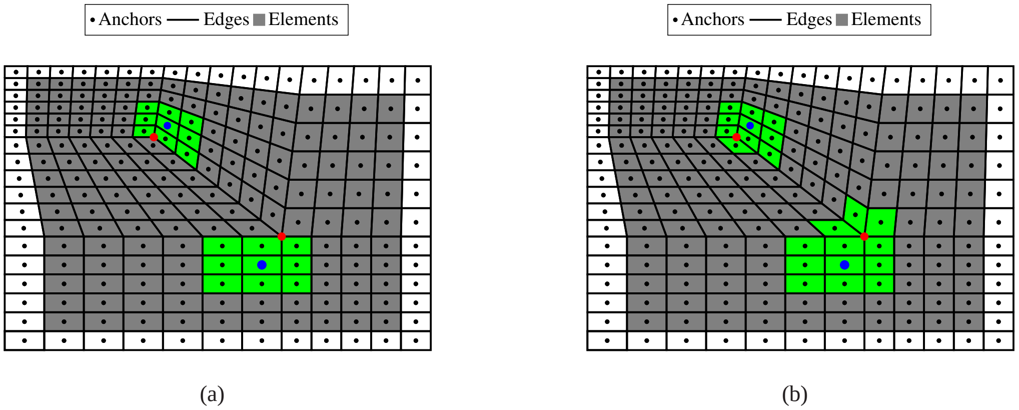


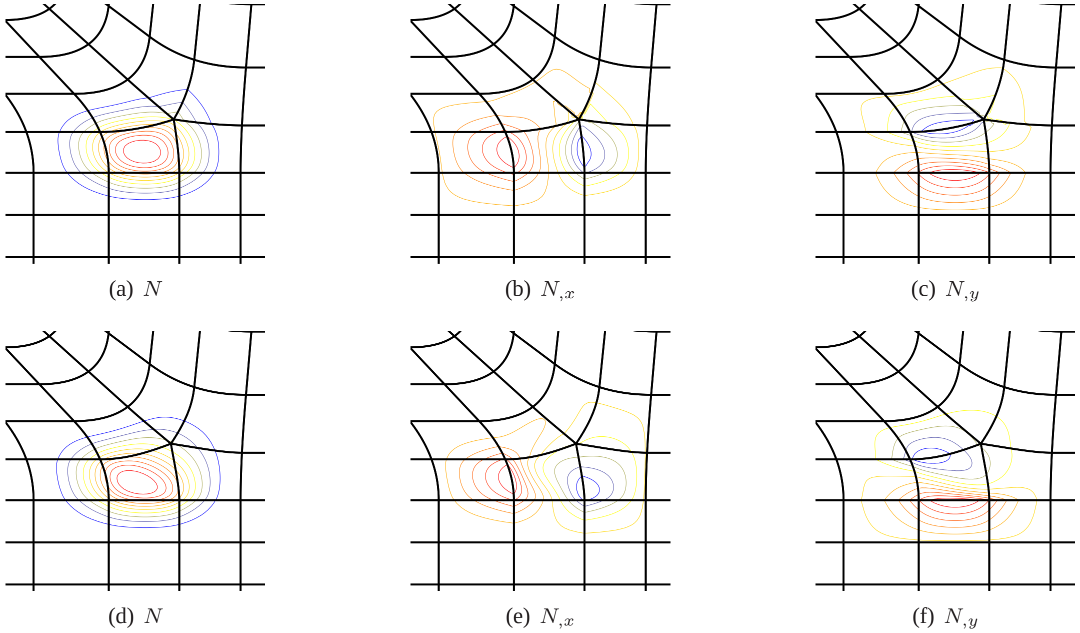
Figure 9. Modifying the Bézier coefficients $c_{\alpha,\beta}$ in the one-ring neighbourhood elements of an extraordinary point (red) results in a modified support of blending functions N that are non-zero in *two* one-ring neighbourhood elements after generalised Bézier extraction: green marks the support of the two blue blending functions (a) before and (b) after solving the constrained least square problem in Equation (58).

The Bézier extraction operators $\underline{\underline{\mathbf{C}}}_e$ for the one-ring neighbourhood elements of an extraordinary point are not square matrices, i. e. hierarchical refinement [31] or Bézier projection [32] are not applicable. For the one-ring neighbourhood elements of the extraordinary point with valence three, the Bézier extraction operator $\underline{\underline{\mathbf{C}}}_e$ has full row rank – the blending functions N are locally linearly independent. The blending functions N are locally linearly dependent for the one-ring neighbourhood elements of the extraordinary point with valence five since $\underline{\underline{\mathbf{C}}}_e$ does not have full row rank.

Figures 10 and 11 show a blending function N and its first derivatives $N_{,x}$ and $N_{,y}$ in the physical domain before and after modifying the Bézier coefficients $c_{\alpha,\beta}$. It can be seen that $N_{,x}$ and $N_{,y}$ are continuous after smoothing. Thus, the blending functions N are $\mathcal{C}^1_{\mathrm{A}}$-continuous. Modifying the Bézier coefficients $c_{\alpha,\beta}$ for the cubic case such that the unstructured T-spline mesh fulfils the partition of unity property for the blending functions N and is C^1_{A}-continuous around the extraordinary points has been considered in [29].

If a T-spline mesh does not fulfil the partition of unity property it is non-analysis-suitable according to [36]. However, analysis can also be performed with non-analysis-suitable T-spline meshes, see also the discussion of the term “analysis-suitable” in [31]. For instance, the unstructured T-spline meshes utilised in [37–45] are non-analysis-suitable T-spline meshes since the blending functions do not fulfil the partition of unity property in the one-ring neighbourhood elements of an extraordinary point.

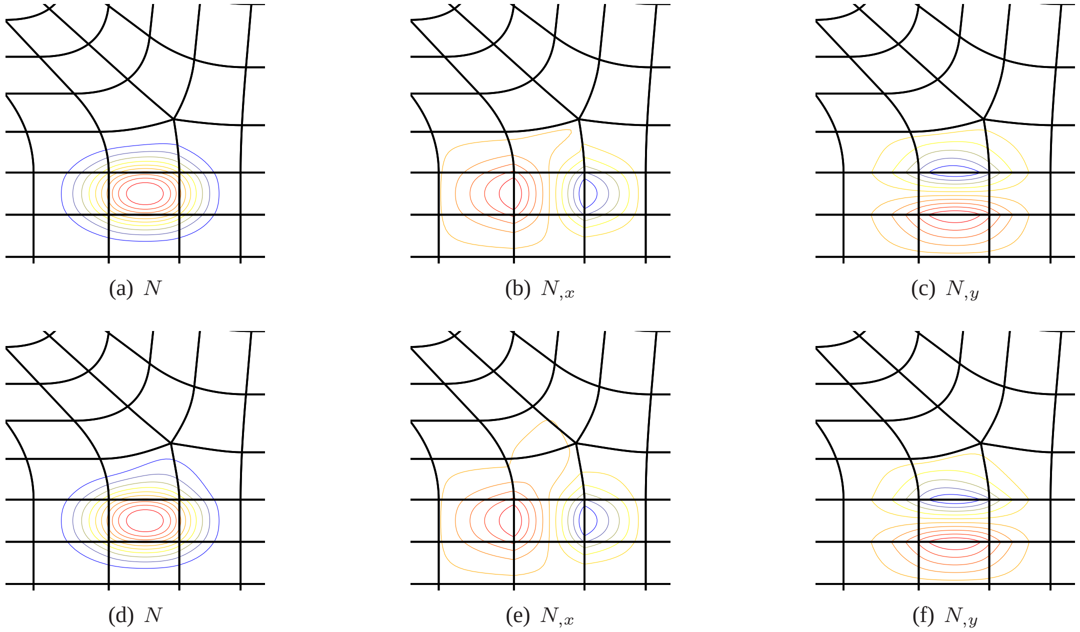
Furthermore, it was concluded in [46] that the unstructured T-spline mesh in [42] is an analysis-suitable T-spline. Unfortunately, it was not taken into account in [46] that the blending functions of an analysis-suitable T-spline have to fulfil the partition of unity property which is not the case for the unstructured T-spline meshes in [42]. This shows that it may not be possible to conclude from the topology of an unstructured T-spline mesh whether it fulfils the partition of unity property or not and that instead, the Bézier extraction operator should be exploited as in [31].



(a) N (b) $N_{,x}$ (c) $N_{,y}$

(d) N (e) $N_{,x}$ (f) $N_{,y}$

Figure 10. Contour plots of a quadratic blending function N and its derivatives $N_{,x}$, $N_{,y}$ in the physical domain before (a)-(c) and after (d)-(f) smoothing. The blending function corresponds to an anchor that is located in the one-ring neighbourhood of the extraordinary point of valence five.



(a) N (b) $N_{,x}$ (c) $N_{,y}$

(d) N (e) $N_{,x}$ (f) $N_{,y}$

Figure 11. Contour plots of a quadratic blending function N and its derivatives $N_{,x}$, $N_{,y}$ in the physical domain before (a)-(c) and after (d)-(f) smoothing. The blending function corresponds to an anchor that is located in the two-ring neighbourhood of the extraordinary point of valence five.

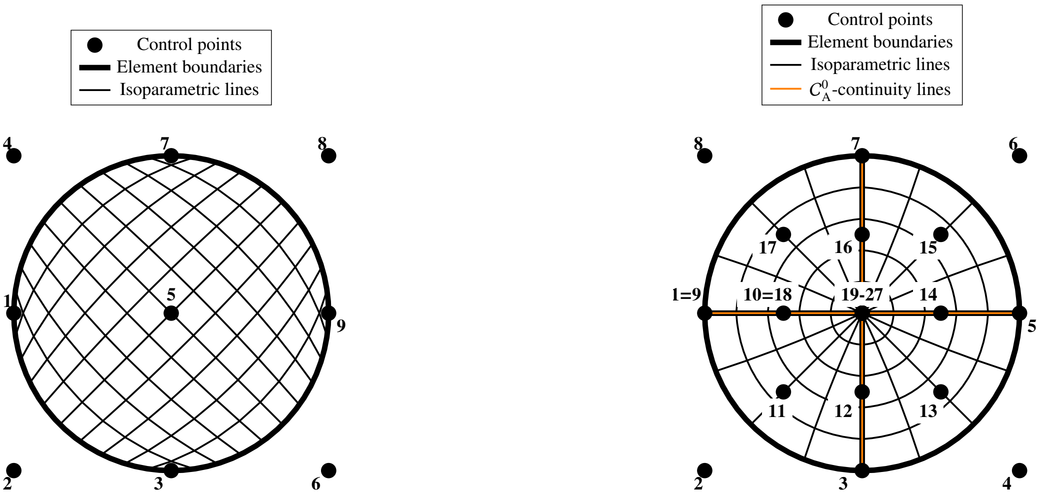
6. THE REPRESENTATION OF A DISC WITH NURBS, T-SPLINES, NURBS-TO-NURPS AND POWELL-SABIN B-SPLINES

In Section 7, an analysis will be carried out for a Kirchhoff-Love plate with a circular geometry. Therefore, we now create a number of discretisations for this geometry using NURBS, NURBS-to-NURPS, unstructured T-splines and Powell-Sabin B-splines. The first two methods can represent a circular boundary exactly, while the unstructured T-spline and Powell-Sabin B-spline mesh only approximate the circle.

6.1. Representation of a disc with NURBS

6.1.1. Single patch

An area with a circular boundary can be created with a single quadratic NURBS patch (or one element) using nine control points, see Figure 12(a). The isoparametric lines indicate where singularities blue (i. e. the determinant of the Jacobian matrix $\underline{\underline{\mathbf{J}}}$ in Equation (48) vanishes) are introduced: at control points one, three, seven and nine, see also [47]. Uniform h-refinement will be applied for the convergence study in Section 7.



(a)

(b)

Figure 12. Representation of a disc using NURBS with (a) one single quadratic element / patch and (b) a polar parameterisation using four patches.

6.1.2. Polar parameterisation

Another possibility to construct a circle is by a polar parameterisation with 27 control points that uses four NURBS patches, Figure 12(b). This results in a singular / degenerated point in the centre where the determinant of the Jacobian matrix $\underline{\underline{\mathbf{J}}}$ in Equation (48) vanishes. Also, four $\mathcal{C}^0_\mathrm{A}$-continuity lines (orange) are introduced. As for the single NURBS-patch from the previous section, uniform h-refinement will be applied for the convergence study in Section 7. h-refinement does not change the number of $\mathcal{C}^0_\mathrm{A}$-continuity lines.

The bending strip method proposed in [10] will be employed in Section 7 at these $\mathcal{C}^0_\mathrm{A}$-continuity lines. The bending strip method adds a stiffness matrix of the form

$$\underline{\underline{\mathbf{K}}}_{BS} = \int_\Omega \underline{\underline{\mathbf{B}}}^T \, \underline{\underline{\mathbf{Q}}}^T \, \underline{\underline{\mathbf{D}}}_{BS} \, \underline{\underline{\mathbf{Q}}} \, \underline{\underline{\mathbf{B}}} \, \underline{\boldsymbol{w}} \, \mathrm{d}x\mathrm{d}y \tag{80}$$

to Equation (23) at the interfaces between patches in order to regain $\mathcal{C}^1_\mathrm{A}$-continuity in an approximate sense. In Equation (80)

$$\underline{\underline{\mathbf{D}}}_{BS} = \frac{h^3}{12} \begin{bmatrix} E_{BS} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{81}$$

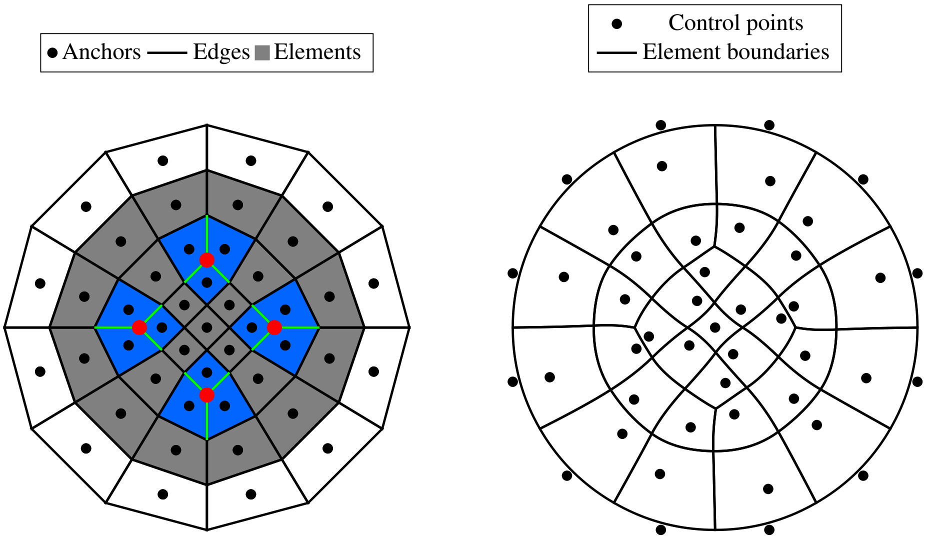
represents the penalty matrix with the penalty bending stiffness E_{BS} and

$$\underline{\underline{\mathbf{Q}}} = \begin{bmatrix} n_1 & n_2 & 0 \\ -n_2 & n_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{82}$$

is the rotation matrix with the normal vector $\underline{\boldsymbol{n}}$ of the bending strip. The rotation matrix $\underline{\underline{\mathbf{Q}}}$ aligns the bending matrix $\underline{\underline{\mathbf{D}}}_{BS}$ with the bending direction.

6.2. Representation of a disc using unstructured T-splines

The approach from Section 5 generates $\mathcal{C}^1_{\mathrm{A}}$-continuous blending functions that fulfil the partition of unity property for an unstructured quadratic T-spline mesh with extraordinary points. The T-spline mesh for the circular disc is shown for the index domain in Figure 13(a).



(a)

(b)

Figure 13. An unstructured quadratic T-spline mesh in (a) the index domain and (b) in the physical domain. (a) Extraordinary points are marked red, spoke edges green and one-ring neighbourhood elements of an extraordinary point blue. (b) The T-spline mesh in the physical domain only approximates the circular boundary.

A circular geometry cannot be represented exactly without double knots, see [48]. Hence, the unstructured quadratic T-spline mesh can only approximate the circular geometry since it is $\mathcal{C}^1_{\mathrm{A}}$-continuous in the entire domain. The n_B control points on the boundary $\underline{\boldsymbol{P}}^i_B$ are determined by solving

$$\sum_{i=1}^{n_B} N^i(\bar{\xi}_k)\underline{\boldsymbol{P}}^i_B = \underline{\boldsymbol{S}}_{\text{Circle}}(\bar{\xi}_k) \quad \text{for } k=1\ldots n_B. \tag{83}$$

$\bar{\xi}$ is the coordinate along the circle with $0 \leq \bar{\xi} \leq 360$, and $\underline{\boldsymbol{S}}_{\text{Circle}}(\bar{\xi}_k)$ the curve representing the circle. Along the boundary are n_B elements. The centre of each boundary element is located at

$$\bar{\xi}_k = \frac{360}{n_B}k + \frac{180}{n_B} \quad \text{for } k=1\ldots n_B. \tag{84}$$

Substituting $\bar{\xi}_k$ from Equation (84) into Equation (83) results in the following system

$$\begin{bmatrix} \frac{6}{8} & \frac{1}{8} & 0 & 0 & 0 & \dots & 0 & \frac{1}{8} \\ \frac{1}{8} & \frac{6}{8} & \frac{1}{8} & 0 & 0 & \dots & 0 & 0 \\ 0 & \frac{1}{8} & \frac{6}{8} & \frac{1}{8} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{1}{8} & 0 & 0 & 0 & 0 & \dots & \frac{1}{8} & \frac{6}{8} \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{P}}_B^{1\;T} \\ \underline{\boldsymbol{P}}_B^{2\;T} \\ \underline{\boldsymbol{P}}_B^{3\;T} \\ \vdots \\ \underline{\boldsymbol{P}}_B^{n_B\;T} \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{S}}_{\text{Circle}}^T(\bar{\xi}_1) \\ \underline{\boldsymbol{S}}_{\text{Circle}}^T(\bar{\xi}_2) \\ \underline{\boldsymbol{S}}_{\text{Circle}}^T(\bar{\xi}_3) \\ \vdots \\ \underline{\boldsymbol{S}}_{\text{Circle}}^T(\bar{\xi}_{n_B}) \end{bmatrix}, \tag{85}$$

which can be solved for the control points on the boundary $\underline{\boldsymbol{P}}_B^i$. The values in the matrix of Equation (85) stem from

$$\begin{bmatrix} N^{k-1}(\xi_k) \\ N^k(\xi_k) \\ N^{k+1}(\xi_k) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 1 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} B^1(\xi_k) \\ B^2(\xi_k) \\ B^3(\xi_k) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 1 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{4} \\ \frac{2}{4} \\ \frac{1}{4} \end{bmatrix} = \begin{bmatrix} \frac{1}{8} \\ \frac{6}{8} \\ \frac{1}{8} \end{bmatrix}. \tag{86}$$

The location of the control points in the interior of the T-spline mesh in Figure 13 is determined by solving

$$\frac{\partial}{\partial \xi_\beta}\left(\frac{\partial x_\alpha}{\partial \xi_\beta} + \frac{\partial x_\beta}{\partial \xi_\alpha}\right) = 0 \tag{87}$$

in the sub-parameter domain ξ_α, while the location of the control points on the boundary $\underline{\boldsymbol{P}}_B^i$ of the disc is prescribed. The resulting T-spline mesh in the physical domain is depicted in Figure 13(b). Two T-spline meshes with a different refinement level are not nested – for each refinement level, Equations (85) and (87) are solved for the determination of the control points. As already discussed in Section 5, the Bézier extraction operators $\underline{\underline{\mathbf{C}}}_e$ for the elements in the one-ring neighbourhood of an extraordinary point are not square matrices. Hence, these elements cannot be refined hierarchically as in [31] and the Bézier projection procedure [32] cannot be applied since the inverse of Bézier extraction operator – the reconstruction operator – is required.

6.3. *Representation of a disc using the NURBS-to-NURPS methodology*

We now transform the single NURBS patch of Section 6.1.1 into a NURPS mesh (NURBS-to-NURPS) following [16]. The boundary of the NURBS-to-NURPS matches exactly the boundary defined by the single NURBS patch, see Figure 14. In order to represent the circular boundary exactly, the Powell-Sabin triangles that correspond to the corners $\underline{\boldsymbol{\xi}} = (0,0); (0,1); (1,0); (1,1)$ in the parameter domain need to be degenerated into a line in the physical domain $\underline{\boldsymbol{x}}$ (dashed lines in Figure 14(c) and Figure 14(d)). It is noted that the NURBS-to-NURPS approach is based on a single NURBS patch, and that a method for transforming multiple NURBS patches into a NURPS has so far not been proposed.

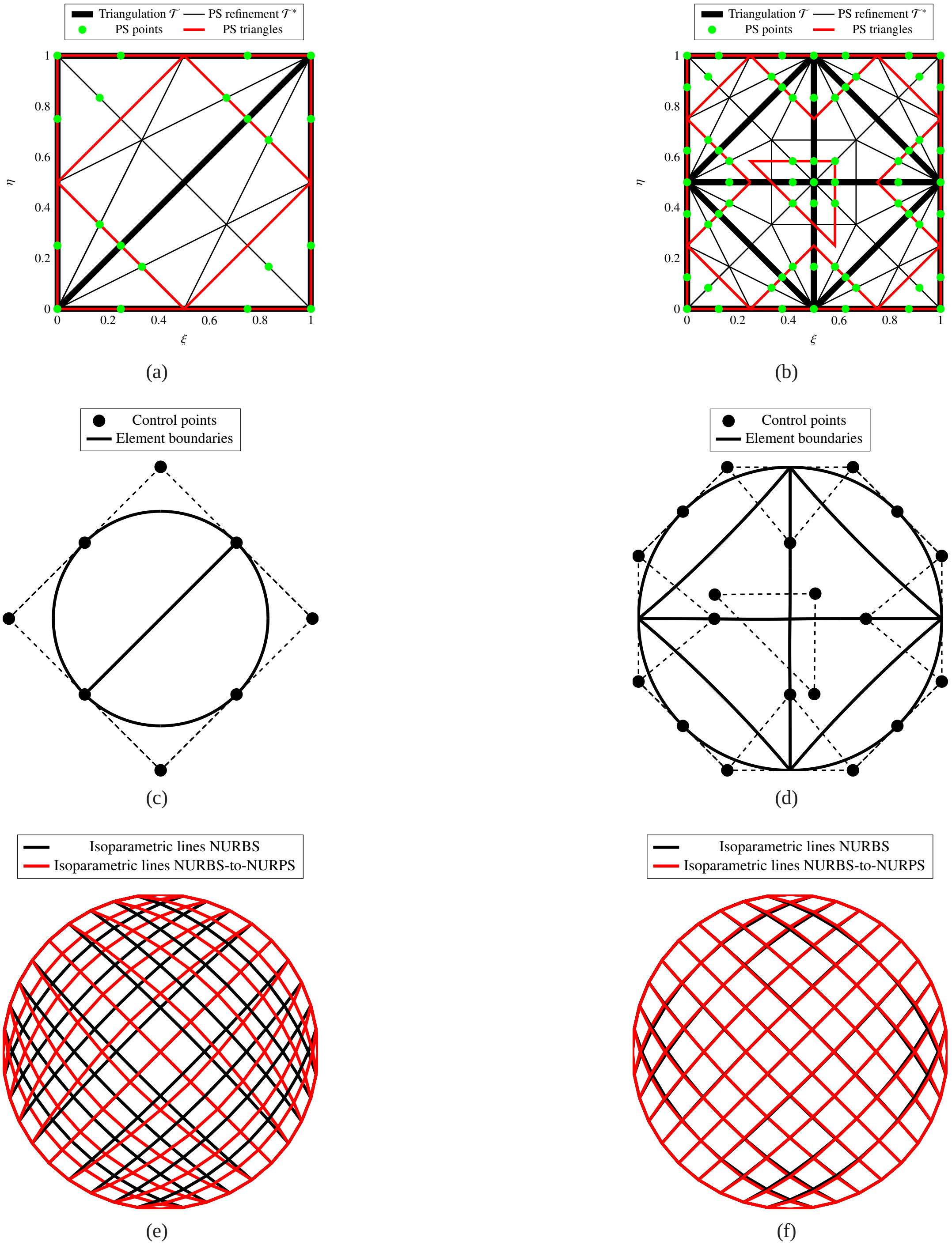
6.4. *Representation of a disc with Powell-Sabin B-splines*

A linear $\mathcal{C}_{\text{A}}^0$ finite element triangulation $\mathcal{T}$ can be transformed into a $\mathcal{C}_{\text{A}}^1$ Powell-Sabin B-spline mesh $\mathcal{T}^*$, see [14,15]. This corresponds to a NURPS mesh with the location of the control points $\underline{\boldsymbol{P}}_i^j = \underline{\boldsymbol{Q}}_i^j$ and for all weights $w_i^j = 1$ in Equation (43), i. e. the parametric and the physical domains, $\underline{\boldsymbol{\xi}}$ and $\underline{\boldsymbol{x}}$, respectively, are identical. For this case, the circular boundary of the disc is only approximated. The Powell-Sabin triangles on the boundary are constrained in such a way that one corner of the Powell-Sabin triangle is always equivalent to the vertex coordinate, see Figure 15. Upon mesh refinement, the Powell-Sabin triangles on the boundary progressively deteriorate into lines, see Figure 15(b).

7. NUMERICAL RESULTS

In this section the circular Kirchhoff-Love plate of Figure 16 is considered for two different boundary conditions: simply supported and clamped. The parameters are $E = 2.1 \times 10^5$MPa,

DOI: 10.1002/nme


Triangulation $\mathcal{T}$
PS refinement $\mathcal{T}^*$
PS points
PS triangles
η
ξ
0
0.2
0.4
0.6
0.8
1
Control points
Element boundaries
Isoparametric lines NURBS
Isoparametric lines NURBS-to-NURPS

(a) (b) (c) (d) (e) (f)

Figure 14. Generation of a NURBS-to-NURPS mesh from a single NURBS patch. From the left to the right column, the triangulation in the NURBS patch is refined. (a) and (b) show the triangulation $\mathcal{T}$ and Powell-Sabin refinement $\mathcal{T}^*$ in the parameter domain $\underline{\boldsymbol{\xi}}$. (c) and (d) show the NURBS-to-NURPS mesh in the physical domain $\underline{\boldsymbol{x}}$; dashed lines connect the control points of the corners of the Powell-Sabin triangles. (e) and (f) show isoparametric lines for the NURBS and the NURBS-to-NURPS meshes in the physical domain $\underline{\boldsymbol{x}}$. Upon refinement, the NURBS-to-NURPS representation converges to the NURBS parameterisation.

DOI: 10.1002/nme

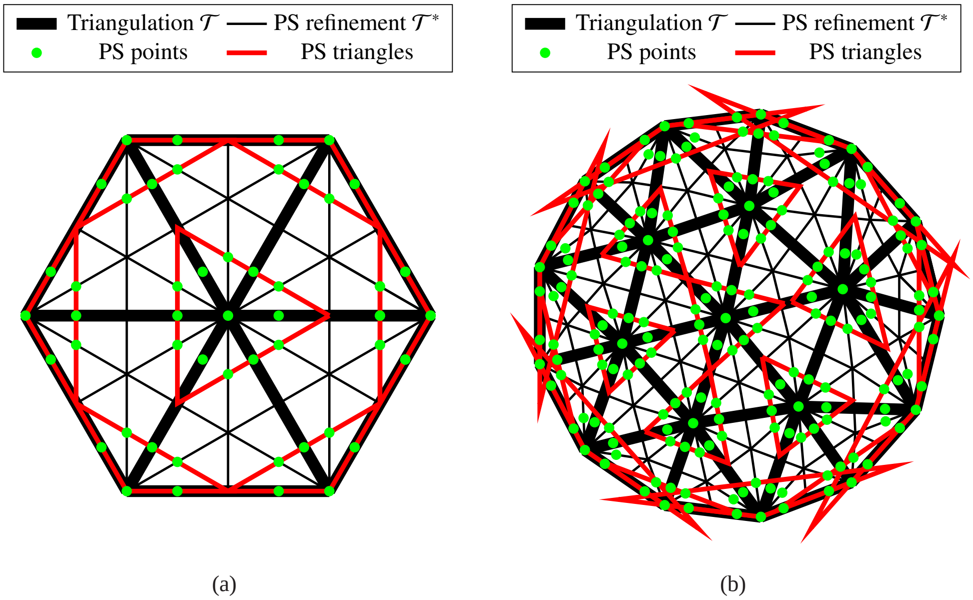

Triangulation $\mathcal{T}$
PS refinement $\mathcal{T}^*$
PS points
PS triangles
Triangulation $\mathcal{T}$
PS refinement $\mathcal{T}^*$
PS points
PS triangles
(a)
(b)

Figure 15. Approximation of a circle using Powell-Sabin B-splines for two different levels of refinement. The Powell-Sabin triangles on the boundary are constrained in such a way that one corner of each Powell-Sabin triangle is equivalent to the vertex.

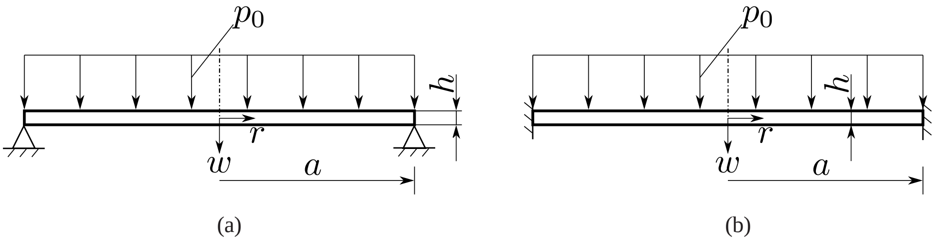


Figure 16. Computational set-up for (a) the simply supported and (b) the clamped circular Kirchhoff-Love plate.

$\nu = 0.3$, $p_0 = 0.16\text{MPa}$, $a = 250\text{mm}$ and $h = 10\text{mm}$. The analytical solutions w_{ex} for both cases can be found in [25]. In the following, the results for the single NURBS patch from Section 6.1.1 will be plotted in all convergence plots for comparison. The L_2 error norm is computed from

$$w_{L_2} = \frac{\sqrt{\int_\Omega \left(w - w_{\text{ex}}\right)^2 \mathrm{d}x\mathrm{d}y}}{\sqrt{\int_\Omega w_{\text{ex}}^2 \,\mathrm{d}x\mathrm{d}y}}. \tag{88}$$

The convergence rate for the fourth order partial differential equation in Equation (13) is equal to two for quadratic basis functions ($p = 2$) according to [49],

$$w_{L_2} \leq Ch^{\min(p+1,2p-2)} = Ch^2 \tag{89}$$

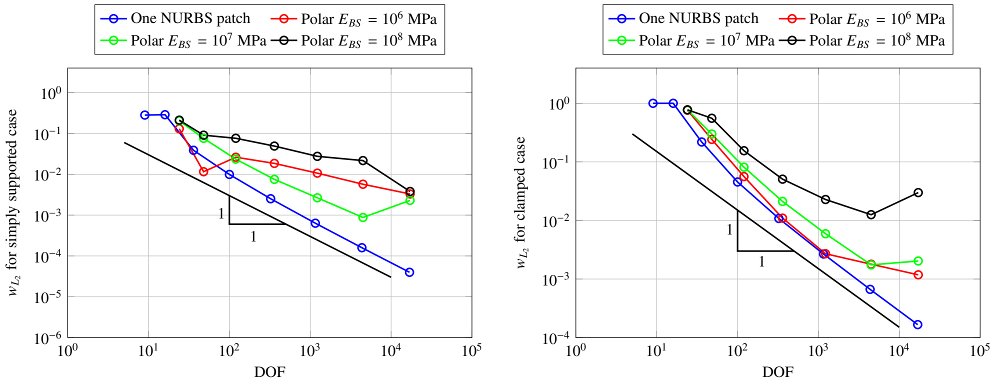
with the mesh size h and a constant C. In order to transform Equation (89) onto the degrees of freedom DOF, the relation

$$h = \frac{p+1}{\sqrt{\text{DOF}}} \tag{90}$$

is used so that we obtain

$$w_{L_2} \leq \tilde{C}\,\text{DOF}^{-1}. \tag{91}$$

For the NURBS and T-spline meshes, the deflection w for the outer control points is set to zero for the simply supported case, while the deflection w of the two outer rows of control points is set to zero for the clamped case. For the Powell-Sabin B-splines and NURBS-to-NURPS, the deflection w of a control point associated to a vertex on the boundary is set to zero for the simply supported case if $\alpha_k^j > 0$ holds in Equation (30)(a). For the clampled case, the deflection w of all three control points which are associated to a vertex that lies on the boundary is set to zero.

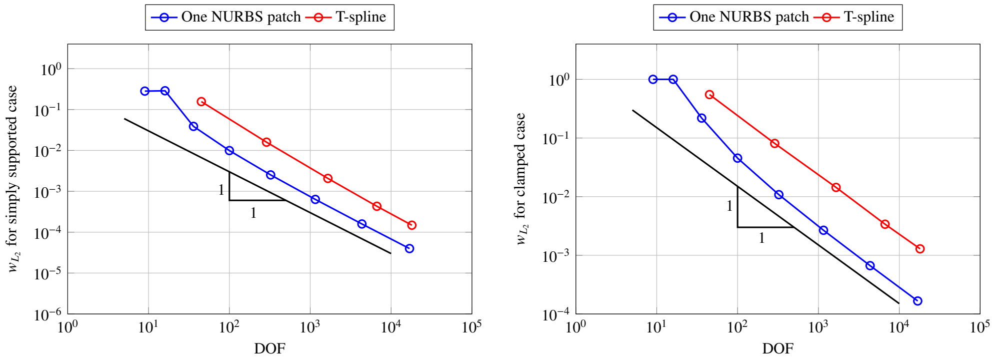


(a)

(b)

Figure 17. Convergence plots for the single NURBS patch and for the polar NURBS parameterisation using four NURBS patches for (a) the simply supported and (b) the clamped boundary conditions.

Figure 17 shows the error in the L_2-norm when the circle is represented with four NURBS patches and a polar parameterisation with a singular point in the centre as in Section 6.1.2. For this $(\mathcal{G}^1, \mathcal{C}^0_\mathrm{A})$-construction, the bending strip method has been applied along the $\mathcal{C}^0_\mathrm{A}$-continuity lines for different values of the penalty stiffness E_{BS}. The results do not converge well for the finer meshes. Moreover, the value of the penalty stiffness E_{BS} for which the lowest error w_{L_2} is obtained, is different for both boundary conditions.

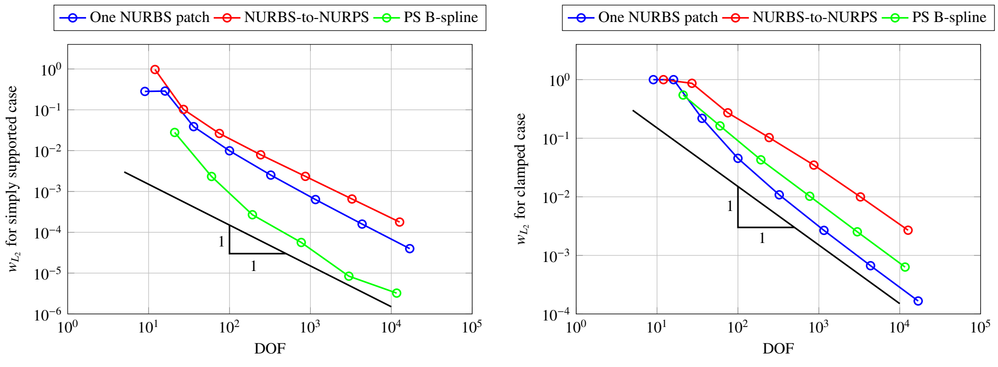


(a)

(b)

Figure 18. Convergence plots for the single NURBS patch and the T-spline mesh for (a) the simply supported and (b) the clamped boundary conditions.

Figure 18 gives the results upon mesh refinement for the unstructured T-spline mesh from Section 6.2, again together with the results that stem from the single NURBS patch for comparison. In contrast to the previous $(\mathcal{G}^1, \mathcal{C}^0_\mathrm{A})$-construction using the bending strip method, a constant convergence rate is observed for the simply supported and clamped case for the unstructured T-spline mesh. The error w_{L_2} for the unstructured T-spline mesh is larger than the error w_{L_2} for the single NURBS patch. A convergence study was also carried out for an unstructured T-spline mesh without the $\mathcal{G}^1$-construction that yields $\mathcal{C}^1_A$-continuous blending functions for the one-ring neighbourhood elements, i. e. a T-spline mesh that is only $\mathcal{C}^0_A$-continuous along spoke edges and non-standard. This T-spline mesh gives almost identical results for the error w_{L_2} and the results are not plotted in Figure 18 since they are not distinguishable from the error for standard T-spline meshes with $\mathcal{C}^1_A$-continuity along spoke edges.



(a) (b)

Figure 19. Convergence plots for the single NURBS patch, the NURBS-to-NURPS approach and the Powell-Sabin B-splines for (a) the simply supported and (b) the clamped boundary conditions.

Finally, the results for the NURBS-to-NURPS configuration and for the standard Powell-Sabin B-spline computations are given in Figure 19. Unexpectedly, the Powell-Sabin B-spline mesh, which only *approximates* the geometry, gives the lowest error w_{L_2} in case of the simply supported boundary conditions, even lower than the single NURBS patch. This is not the case for the clamped boundary conditions, but also then the standard Powell-Sabin B-spline mesh, which approximates the geometry, gives results that are superior to those from the NURBS-to-NURPS. A possible explanation is that the effect of the distorted elements (degenerated Powell-Sabin triangles), which are introduced by the NURBS-to-NURPS approach, is not compensated by the improved (exact) capturing of the boundary.

8. CONCLUDING REMARKS

In this paper a method has been developed to solve boundary value problems that stem from Kirchhoff-Love plate theory using quadratic Powell-Sabin B-splines. By virtue of the fact that these interpolation functions are $\mathcal{C}^1_{\mathrm{A}}$-continuous, the resulting fourth-order partial differential equation can be solved properly. Numerical efficiency is achieved by exploiting Bézier extraction, similar to procedures that have been developed for NURBS and T-splines [20, 21]. Further, the Bézier coefficients in an unstructured quadratic T-spline meshes have been modified such that the resulting mesh is $\mathcal{C}^1_{\mathrm{A}}$-continuous in the entire domain and fulfils the partition of unity property.

Quadratic Powell-Sabin B-splines have a disadvantage compared to NURBS as the geometry is generally not captured exactly. Conversely, the fact that they are $\mathcal{C}^1_{\mathrm{A}}$-continuous everywhere avoids using the bending strip method [10] for joined NURBS patches. Furthermore, using the Powell-Sabin technique, a $\mathcal{C}^1_{\mathrm{A}}$-continuous triangulations can be created from arbitrary discretisation as a pre-processing step and is simpler than having to deal with several NURBS patches or complex T-spline technology.

In a first assessment of the advantages and drawbacks of quadratic Powell-Sabin B-splines compared to NURBS and T-splines, a circular Kirchhoff-Love plate has been considered. Different representations of the disc have been examined, including a single NURBS patch, a polar parameterisation consisting of four joined NURBS patches, a T-spline mesh with extraordinary points, a NURBS-to-NURPS configuration, and Powell-Sabin B-splines. The numerical results show that the use of the bending strip method can impair convergence when the discretisation is refined. In all cases the single NURBS patch performed better than the NURBS patches or the T-splines. The Powell-Sabin B-spline mesh performed almost as well as the single NURBS patch in the calculations for the clamped boundary conditions, and better than the single NURBS patch for the case with simply supported boundary conditions. For both cases the Powell-Sabin splines yielded more accurate results than the NURBS-to-NURPS approach. This observation comes somewhat at a surprise, since the geometry of the boundary is *approximated* in the Powell-Sabin B-spline mesh. The explanation may be that the inability of Powell-Sabin B-splines to capture the boundary exactly is more than off-set by the loss of accuracy that stems for the badly shaped triangles in the NURBS-to-NURPS discretisations.

A. CODE SNIPPET FOR THE BÉZIER EXTRACTION PROCEDURE FOR NURPS

Algorithm 1 gives the Bézier extraction procedure in a Matlab code snippet for the evaluation of the stiffness matrix $\underline{\underline{\mathbf{K}}}$ and the force vector $\underline{\boldsymbol{f}}$ in Equation (23).

```
Input: ControlPts, Weights, Bézier extraction operator C_Operator for all mini-triangles,
Connectivity Conn, Number of Gauss points ngauss
Output: Stiffness matrix K, force vector f

% get the values of the Bernstein polynomials and their derivatives at the GPs of the barycentric coordinate system
$\underline{\tilde{\tau}}$
for $i = 1 : ngauss$ do
    [B(:,i), dBdt1(:,i), dBdt2(:,i), d2Bdt1(:,i), d2Bdt2(:,i), d2Bdt1dt2(:,i),] =
    f_d1Bernstein(tildetau1(i),tildetau2(i));
end
```

REFERENCES

1. Kirchhoff G. Über das Gleichgewicht und die Bewegung einer elastischen Scheibe. *Journal für die reine und angewandte Mathematik* 1850; **40**:51–88.
2. Verhoosel CV, Scott MA, Hughes TJR, de Borst R. An isogeometric analysis approach to gradient damage models. *International Journal for Numerical Methods in Engineering* 2011; **86**(1):115–134.
3. Tran LV, Ferreira AJM, Nguyen-Xuan H. Isogeometric analysis of functionally graded plates using higher-order shear deformation theory. *Composites Part B: Engineering* 2013; **51**(0):368–383.
4. Gómez H, Calo VM, Bazilevs Y, Hughes TJR. Isogeometric analysis of the Cahn–Hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering* 2008; **197**(49):4333–4352.
5. Sederberg TW, Zheng J, Bakenov A, Nasri A. T-splines and T-NURCCs. *ACM Transactions on Graphics* 2003; **22**(3):477–484.
6. 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**(3):435–458.
7. Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 2005; **194**(39–41):4135–4195.
8. Cottrell JA, Reali A, Bazilevs Y, Hughes TJR. Isogeometric analysis of structural vibrations. *Computer methods in applied mechanics and engineering* 2006; **195**(41):5257–5296.
9. Kiendl J, Bletzinger KU, Linhard J, Wüchner R. Isogeometric shell analysis with Kirchhoff-Love elements. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**(49–52):3902–3914.
10. Kiendl J, Bazilevs Y, Hsu MC, Wüchner R, Bletzinger KU. The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**(37–40):2403–2416.
11. Scott MA. *T-splines as a Design-Through-Analysis Technology*. Ph.D. thesis; The University of Texas at Austin, 2011.
12. Groisser D, Peters J. Matched G^k-constructions always yield C^k-continuous isogeometric elements. *Computer Aided Geometric Design* 2015; **34**:67–72.
13. Powell MJD, Sabin MA. Piecewise Quadratic Approximations on Triangles. *ACM Transactions on Mathematical Software* 1977; **3**(4):316–325.
14. Dierckx P, van Leemput S, Vermeire T. Algorithms for surface fitting using Powell-Sabin splines. *IMA Journal of Numerical Analysis* 1992; **12**(2):271–299.
15. Dierckx P. On calculating normalized Powell-Sabin B-splines. *Computer Aided Geometric Design* 1997; **15**(1):61–78.
16. Speleers H, Manni C, Pelosi F. From NURBS to NURPS geometries. *Computer Methods in Applied Mechanics and Engineering* 2013; **255**:238–254.

17. Speleers H, Dierckx P, Vandewalle S. Numerical solution of partial differential equations with Powell-Sabin splines. *Journal of Computational and Applied Mathematics* 2006; **189**:643–659.
18. Speleers H, Manni C, Pelosi F, Sampoli ML. Isogeometric analysis with Powell–Sabin splines for advection–diffusion–reaction problems. *Computer Methods in Applied Mechanics and Engineering* 2012; **221–222**:132–148.
19. Stogner RH, Carey GF. C^1 macroelements in adaptive finite element methods. *International Journal for Numerical Methods in Engineering* 2007; **70**(9):1076–1095.
20. Borden MJ, Scott MA, Evans JA, Hughes TJR. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering* 2011; **87**(1–5):15–47.
21. Scott MA, Borden MJ, Verhoosel CV, Sederberg TW, Hughes TJR. Isogeometric finite element data structures based on Bézier extraction of T-splines. *International Journal for Numerical Methods in Engineering* 2011; **88**(2):126–156.
22. Nguyen T, Karčiauskas K, Peters J. A Comparative Study of Several Classical, Discrete Differential and Isogeometric Methods for Solving Poisson's Equation on the Disk. *Axioms* 2014; **3**(2):280–299.
23. Barsky BA, DeRose TD. *Geometric Continuity of Parametric Curves*. Technical Report UCB/CSD 84/205; University of California at Berkeley, 1984.
24. Peters J. Geometric continuity. *Handbook of Computer Aided Geometric Design* 2002; :193–229.
25. Gross D, Hauger W, Wriggers P. *Technische Mechanik: Band 4: Hydromechanik, Elemente der Höheren Mechanik, Numerische Methoden*, vol. 7. Springer, 2009.
26. Fish J, Belytschko T. *A First Course in Finite Elements*. John Wiley & Sons, 2007.
27. Vanraes E, Dierckx P, Bultheel A. *On the choice of the PS-triangles*. Technical Report 353; Department of Computer Science, K.U. Leuven, 2003.
28. Chui CK, Lai MJ. Multivariate vertex splines and finite elements. *Journal of Approximation Theory* 1990; **60**(3):245–343.
29. May S. *Splines for damage and fracture in solids*. Ph.D. thesis; University of Glagow, 2015.
30. Sederberg TW, Cardon DL, Finnigan GT, North NS, Zheng J, Lyche T. T-spline Simplification and Local Refinement. *ACM Transactions on Graphics* Aug 2004; **23**(3):276–283.
31. May S, Vignollet J, de Borst R. The role of the Bézier extraction operator for T-splines of arbitrary degree: linear dependencies, partition of unity property, nesting behaviour and local refinement. *International Journal for Numerical Methods in Engineering* 2015; **103**(8):547–581.
32. Thomas DC, Scott MA, Evans JA, Tew K, Evans EJ. Bézier projection: A unified approach for local projection and quadrature-free refinement and coarsening of NURBS and T-splines with particular application to isogeometric design and analysis. *Computer Methods in Applied Mechanics and Engineering* 2015; **284**:55–105.
33. Björck Å. *Numerical Methods for Least Squares Problems*. Siam, 1996.
34. Du WH, Schmitt FJM. On the G^1 continuity of piecewise Bézier surfaces: a review with new results. *Computer-Aided Design* 1990; **22**(9):556–573.
35. Faux ID, Pratt MJ. *Computational Geometry for Design and Manufacture*. Ellis Horwood Ltd, 1979.
36. Li X, Zheng J, Sederberg TW, Hughes TJR, Scott MA. On linear independence of T-spline blending functions. *Computer Aided Geometric Design* 2012; **29**(1):63–76.
37. Bazilevs Y, Hsu MC, Scott MA. Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with applications to wind turbines. *Computer Methods in Applied Mechanics and Engineering* 2012; **249–252**:28–41.
38. Dimitri R, De Lorenzis L, Scott MA, Wriggers P, Taylor RL, Zavarise G. Isogeometric large deformation frictionless contact using T-splines. *Computer Methods in Applied Mechanics and Engineering* 2014; **269**:394–414.
39. Hsu MC, Kamensky D, Xu F, Kiendl J, Wang C, Wu MCH, Mineroff J, Reali A, Bazilevs Y, Sacks MS. Dynamic and fluid–structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models. *Computational Mechanics* 2015; **55**(6):1211–1225.
40. Kiendl J, Hsu MC, Wu MCH, Reali A. Isogeometric Kirchhoff–Love shell formulations for general hyperelastic materials. *Computer Methods in Applied Mechanics and Engineering* 2015; **291**:280–303.
41. Kostas KV, Ginnis AI, Politis CG, Kaklis PD. Ship-hull shape optimization with a T-spline based BEM–isogeometric solver. *Computer Methods in Applied Mechanics and Engineering* 2015; **284**:611–622.
42. Scott MA, Simpson RN, Evans JA, Lipton S, Bordas SPA, Hughes TJR, Sederberg TW. Isogeometric boundary element analysis using unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering* 2013; **254**:197–221.
43. Scott MA, Hughes TJR, Sederberg TW, Sederberg MT. *An integrated approach to engineering design and analysis using the Autodesk T-spline plugin for Rhino3d*. ICES Report 14-33; The Institute for Computational Engineering and Sciences, The University of Texas at Austin, 2014.
44. Simpson RN, Scott MA, Taus M, Thomas DC, Lian H. Acoustic isogeometric boundary element analysis. *Computer Methods in Applied Mechanics and Engineering* 2014; **269**:265–290.
45. Thomas DC, Scott MA. Isogeometric analysis based on T-splines. *Isogeometric Methods for Numerical Simulation* 2015; :205–232.
46. Li X. Some properties for analysis-suitable T-splines. *Journal of Computational Mathematics* 2015; **33**(4):428–442.
47. Schmidt R, Kiendl J, Bletzinger KU, Wüchner R. Realization of an integrated structural design process: analysis-suitable geometric modelling and isogeometric analysis. *Computing and Visualization in Science* 2010; **13**(7):315–330.
48. Piegl L, Tiller W. A menagerie of rational B-spline circles. *IEEE Computer Graphics and Applications* 1989; **9**(5):48–56.
49. Tagliabue A, Dedè L, Quarteroni A. Isogeometric analysis and error estimates for high order partial differential equations in fluid dynamics. *Computers & Fluids* 2014; **102**:277–303.

```
K = zeros(NoControlPts,NoControlPts);
f = zeros(NoControlPts,1);
% element loop (loop over triangles of triangulation $\mathcal{T}$)
for $e = 1 : NoElements$ do
    % connectivity for this element
    sctr = Conn(e,:);
    elco = ControlPts(sctr,1:2);
    We = diag(Weights(sctr));
    we = Weights(sctr);
    Ke = zeros(length(sctr),length(sctr));
    fe = zeros(length(sctr),1);
    % loop over mini-triangles of triangulation $\mathcal{T}^*$
    for $n = 1 : 6$ do
        % Bézier extraction operator for this mini-triangle
        Cen = C_Operator{e,n};
        wb = Cen'*we;
        % loop over Gauss points
        for $i = 1 : ngauss$ do
            Wb = wb'*B(:,i);
            dWbdt1 = wb'*dBdt1(:,i);
            dWbdt2 = wb'*dBdt2(:,i);
            d2Wbdt1 = wb'*d2Bdt1;
            d2Wbdt2 = wb'*d2Bdt2;
            d2Wbdt1dt2 = wb'*d2Bdt1dt2;

            R = We*Cen*B/Wb;
            dRdt1 = We*Cen*(dBdt1/Wb - B/Wb^2*dWbdt1);
            dRdt2 = We*Cen*(dBdt2/Wb - B/Wb^2*dWbdt2);
            d2Rdt1 = We*Cen*(d2Bdt1/Wb - 2*dBdt1/Wb^2*dWbdt1 + 2*B/Wb^3*dWbdt1^2 -
            B/Wb^2*d2Wbdt1);
            d2Rdt2 = We*Cen*(d2Bdt2/Wb - 2*dBdt2/Wb^2*dWbdt2 + 2*B/Wb^3*dWbdt2^2 -
            B/Wb^2*d2Wbdt2);
            d2Rdt1dt2= We*Cen*(d2Bdt1dt2/Wb - dBdt2*dWbdt1/Wb^2 - dBdt1*dWbdt2/Wb^2 -
            B*d2Wbdt1dt2/Wb^2 + 2*B*dWbdt2*dWbdt1/Wb^3);

            dxdt1 = elco(:,1)'*dRdt1;
            dydt1 = elco(:,2)'*dRdt1;
            dxdt2 = elco(:,1)'*dRdt2;
            dydt2 = elco(:,2)'*dRdt2;
            d2xdt1 = elco(:,1)'*d2Rdt1;
            d2ydt1 = elco(:,2)'*d2Rdt1;
            d2xdt2 = elco(:,1)'*d2Rdt2;
            d2ydt2 = elco(:,2)'*d2Rdt2;
            d2xdt1dt2 = elco(:,1)'*d2Rdt1dt2;
            d2ydt1dt2 = elco(:,2)'*d2Rdt1dt2;

            jacob = [dxdt1 dxdt2;dydt1 dydt2];
            detJelem = det(jacob);
            invJacob = inv(jacob);
            dRdx = [dRdt1 dRdt2] * invJacob;

            d2Rdt1t2 = [d2Rdt1'; d2Rdt2'; d2Rdt1dt2'];
            d2xydt1t2 = [d2xdt1 d2ydt1; d2xdt2 d2ydt2; d2xdt1dt2 d2ydt1dt2];
            dxydt1t2 = [ dxdt1*dxdt1 dydt1*dydt1 2*dxdt1*dydt1;
                         dxdt2*dxdt2 dydt2*dydt2 2*dxdt2*dydt2;
                         dxdt1*dxdt2 dydt1*dydt2 dxdt1*dydt2 + dxdt2*dydt1];
            d2Rdx = dxydt1t2\(d2Rdt1t2 - d2xydt1t2*dRdx');

            d2Rdx2(1,:) = d2Rdx(1,:);
            d2Rdy2(1,:) = d2Rdx(2,:);
            d2Rdxdy(1,:)= d2Rdx(3,:);

            Be = [d2Rdx2; d2Rdy2; 2*d2Rdxdy];
            Ke = Ke + 1/2*w(i)*(Be'*D*Be)*detJelem;
            fe = fe + 1/2*w(i)*(R'*p0)*detJelem;
        end
    end
    K(sctr,sctr) = K(sctr,sctr) + Ke;
    f(sctr,1) = f(sctr,1) + fe;
end
```

Algorithm 1: Algorithm for the Bézier extraction procedure for NURPS for the calculation of the stiffness matrix and the force vector in Equation (23).