ISSN 2176-5480

22nd International Congress of Mechanical Engineering (COBEM 2013)
November 3-7, 2013, Ribeirão Preto, SP, Brazil

IMMERSED NORMALIZED WEIGHTED B-SPLINE (I-SPLINE) FINITE ELEMENT METHOD CONVERGENCE FOR 2D ELASTICITY PROBLEMS

Patrícia Tonon
Rodolfo André Kuche Sanches
Universidade Tecnológica Federal do Paraná, Campus de Pato Branco
rodolfosanches@utfpr.edu.br

Abstract. *Sanches et al. (2011) introduces a novel finite element method based on immersed B-splines (i-spline) for problems with complex geometry and topology. The analysis using the finite element method is performed in a block structured logically Cartesian mesh, which is slightly larger than the domain of interest. The boundary is defined by the zero-th level-set of a signed distance function. Far from the boundary, the standard B-spline basis functions are used as shape functions, however, close to domain boundaries, the B-spline basis functions are modified so that they satisfy the boundary conditions of Dirichlet. Although the new technique is robust and efficient, in the first tests it appeared not to converge as it was expect to. In this work we study the convergence properties of the method when applied to two-dimensional elasticity problems, furthermore, it was shown efficiency and robustness of the method and also that good convergence rates are possible.*

Keywords: *b-splines, immersed boundary, finite elements, linear elasticity, convergence*

1. INTRODUCTION

B-Splines and correlated functions are widely employed in engineering design, constituting the base for most of the computer graphics and computer aided design (CAD) softwares. However, in the finite element context, the use of b-splines is relatively recent subject and has gained more attention with the introduction of the *isogeometric* analysis paradigm by Hughes *et al.* (2005).

The integration CAD-numerical analysis is not the only advantage of B-Splines basis for finite element. Another important advantage is related to its properties of positivity and smoothness. This means, e.g., that all the components of a mass matrix are positive and, hence, the related lumped mass matrix is always positive definite and B-spline basis functions are also variation diminishing, enabling stable representation with high degree polynomials (Strang and Fix, 2008; Sanches *et al.*, 2011) and making them less prone to numerical oscillations typically encountered with higher degree Lagrangian basis functions.

However, due to the non-local nature of the B-splines functions, the essential boundary conditions are not directly satisfied, and B-splines are closely related to Cartesian grids. For many practically relevant geometries, Cartesian meshes are not flexible enough and for some geometries, like spherical objects, not possible (Sanches *et al.*, 2011).

One good way to deal with such problems is to combine B-Splines basis to immersed boundary methods. The key of the immersed boundary methods is to employ Cartesian meshes slightly larger than the physical domain and use auxiliary algorithms to enforce the Dirichlet and Neumann boundary conditions (Glowinski *et al.*, 1994) and (Belytschko *et al.*, 2003).

Due to the non-local nature of B-splines, conventional algorithms for enforcing boundary conditions in immersed boundary methods are not directly applicable to an immersed b-spline finite element method. One may suggest the use of Lagrange multipliers or the penalty method, however there are some drawbacks, for instance, the Lagrange multiplier does not lead to positive definite systems, and the penalty method requires very delicate balance of terms in the variation functional. Moreover, both methods generate approximations which, in general, do not satisfy exactly the boundary condition (Höllig *et al.*, 2002).

Based on the approach introduced by L. W. Kantorowitsch (1964) for enforcing homogeneous Dirichlet boundary conditions, by multiplying the basis functions with a weight function which has zero value ovoer the boundaries, Höllig *et al.* (2002), introduces the Web-splines (*weighted extended b-splines*) finite element method. In this method, the basis function close to the domain boundary which have only a small overlap with the physical domain are coupled to the interior basis functions avoiding detrimental effect on the numerical stability of the discretized problem. Although the web-spline method fails the patch test, it can be shown to be convergent. Furthermore, inhomogeneous (i.e., non-zero) Dirichlet boundary conditions are approximately considered by applying a suitable domain load close to the boundaries.

Sanches *et al.* (2011) introduces a novel and robust immersed b-spline finite element method (*i-spline* - Immersed Normalized Weighted B-Spline), which consists on a simple technique sharing many advantages of the web-spline method and at the same time fulfilling the patch test and enabling easy prescription of inhomogeneous Dirichlet boundary con-

ISSN 2176-5480

ditions exactly over the boundaries. The i-spline approach consists basically in firstly deriving B-spline basis functions which are interpolating at the domain boundaries. From outset, B-splines with small support on the physical domain are omitted for interpolation purposes, which alleviates the stability issues associated to this functions. In such paper we obtained for cubic immersed normalized weighted b-splines the same convergence order as the one of finite elements with linear polynomials. In the referenced work, the authors show the robustness and efficiency of the method, however a good discussion on the convergence properties of the methods is missing.

The aim of this work is to apply the *i-splines* to 2D elasticity problems and to study the factors that affect its convergence.

2. B-splines

We present in this section a summary with relevant informations for understanding the proposed method. A detailed description of the B-spline functions may be found in Piegl and Tiller (1997) or Rogers (2001).

A spline curve bay be conveniently represented by a linear combination of smaller B functions, which are called B-splines. On an one-dimensional domain with knots ξ_i, the univariate b-spline shape functions are given with the following recurrence relationship:

$$B_i^0(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi \leq \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

and

$$B_i^n(\xi) = \frac{\xi - \xi_i}{\xi_{i+n} - \xi_i} B_i^{n-1}(\xi) + \frac{\xi_{i+n+1} - \xi}{\xi_{i+n+1} - \xi_{i+1}} B_{i+1}^{n-1}(\xi), \tag{2}$$

whereby n is the polynomial order of the b-spline. From these equations, it can be deduced that the support of a b-spline basis function B_i^n of polynomial order n reaches between $\xi_i \leq \xi \leq \xi_{i+n+1}$ (see Fig. 1). Furthermore, B_i^n is $n-1$ times continuously differentiable over the knots and is a complete polynomial of order n between the knots. In this paper we use only uniform B-splines so that the distance between the knots is uniform, i.e. for all grid nodes i $\xi_{i+1} - \xi_i = const.$ Figure 1 shows B-splines of several polynomial degree, where it is possible to note the increasing smoothness and support size with increasing degree.

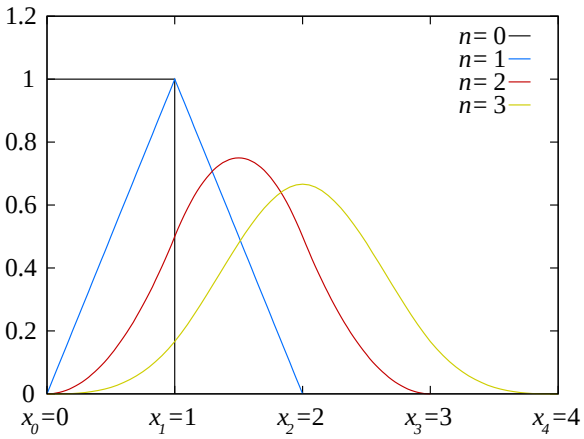


Figure 1. B-splines of polynomial degree zero, one, two and three where ξ matches with the Cartesian coordinates x.

The univariate b-splines can be extended to multi-dimensions using the tensor product formalism, i. e., a B-spline on a n-dimensional space is obtained by combining univariate B-splines in coordinate directions ξ^1, ξ^2, ... ξ^n:

$$B(\xi, \eta) = B(\xi^1) \times B(\xi^2) \times \cdots \times B(\xi^n), \tag{3}$$

where indices have been omitted for clarity. Figure 2 depicts a bivariate cubic B-spline generation.

2.1 B-splines and finite elements

For introducing the B-splines in the finite element method context, we consider an linear elasticity boundary value problem over a domain Ω, with the Neumann boundary Γ_N and the Dirichlet boundary Γ_D (see Fig. 3.

ISSN 2176-5480

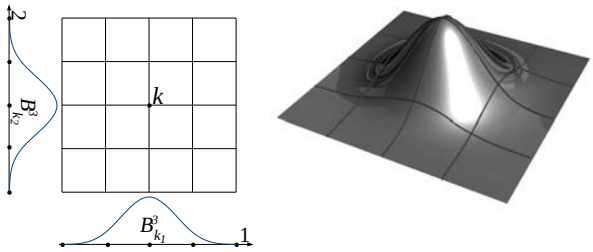


Figure 2. Bivariate cubic B-spline shape function generated as the tensor product of two univariate cubic B-spline shape functions.

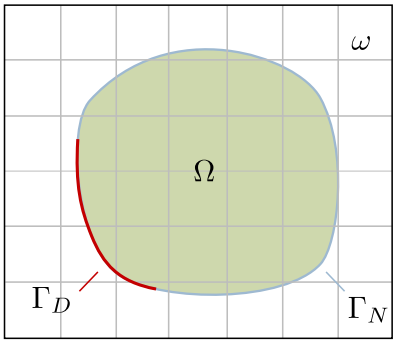


Figure 3. Cartesian domain ω and the immersed physical domain Ω with its Dirichlet and Neumann boundaries Γ_D and Γ_N, respectively.

The test function v for the weak form of the mechanical problem may be obtained by a linear combination of the B-splines basis functions and have to be homogeneous ($v = 0$) on the Dirichlet boundary (Γ_D). For discretizing the weak form with spline shape functions, the problem domain Ω is embedded into a slightly larger domain ω, which is suitable for a Cartesian mesh (see Fig. 3). As discussed, we use the Cartesian mesh in order to facilitate the finite element discretization with tensor product splines. The spline shape functions on the Cartesian mesh provide a means to approximate the displacements and test functions:

$$u(\xi) = \sum_{i=1}^{np} B_i^n(\xi) u_i \qquad v(\xi) = \sum_{i=1}^{np} B_i^n(xi) v_i, \tag{4}$$

where B_i^n are the spline basis functions of polynomial order n, u_i and v_i are the nodal displacements and test functions at node i of the Cartesian mesh and np is the number of nodes. It is evident that this spline approximation does not conform to the physical domain boundary Γ. Hence, it is not directly suitable for discretizing the weak form in Eq. (**??**). However, as will be discussed in the coming sections it is possible to modify the spline shape functions so that they can accommodate the boundary.

3. Normalized weighted b-splines

3.1 Implicit boundary representation

Before moving on to the discussion of the modified b-spline shape functions, we first elaborate on the representation of the physical domain on the Cartesian mesh. The proposed modification of the spline shape functions requires to identify all the cells close to the boundary Γ and if they are inside or outside the physical domain Ω. To this end, a computationally efficient and scalable approach is to use a signed distance function (or, level set function):

$$\phi(P,\Gamma) = \begin{cases} \text{distance}(P,\Gamma) & \text{if } P \in \Omega \\ 0 & \text{if } P \in \Gamma \\ -\text{distance}(P,\Gamma) & \text{otherwise} \end{cases}, \tag{5}$$

ISSN 2176-5480

where P is any point on the Cartesian domain. A discrete representation of the signed distance function can be conveniently obtained by combining the spline shape functions with the signed distance values at the nodes ϕ_i:

$$\phi = \sum_{i=1}^{np} B_i^n(\xi)\phi_i. \tag{6}$$

Note that the shape functions B_i^n do not need to conform to the physical boundary since they are only used for interpolation of a scalar field ϕ.

The boundary will be represented by the zero-th level-set of ϕ. In contrast to the usual parametric mesh based boundary representations (using segments or facets), level set based representations are more suitable for problems with large deformations and topology changes.

3.2 The modified shape functions

In this section we discuss the method proposed by Sanches *et al.* (2011).

As discussed before, the b-spline shape functions defined over the Cartesian mesh do not conform to the boundary of the physical domain. In particular, they are non-interpolating at Dirichlet boundaries and, hence, cannot be used for discretizing the weak form In the following, we develop an approach for modifying the shape functions associated with the Cartesian mesh so that they are interpolating at the boundaries. First, we define similar to Höllig *et al.* (2002) a weight function which is zero outside the domain, one inside the domain and has a smooth transition zone in between

$$w(\xi) = \begin{cases} 0 & \text{for } \phi(\xi) \leq 0 \\ 1 - \left(1 - \dfrac{\phi(\xi)}{\delta}\right)^k & \text{for } 0 > \phi(\xi) \leq \delta \\ 1 & \text{for } \phi(\xi) > \delta \end{cases} \tag{7}$$

where $\phi(\xi)$ is the signed distance, δ is the transition length and k is an integer with $k \geq 1$ which controls the smoothness of the weight function inside the domain. Note that the weight function w has a non-zero gradient and is C^{k-1} continuous at $\phi(\xi) = \delta$.

Next, all the Cartesian mesh cells are tagged as *physical*, *fictitious* or *boundary* depending on their position with respect to the physical domain. This classification is performed by computing for each cell $\square_e$ the minimum and maximum signed distance $\min \phi(\square_e)$ and $\max \phi(\square_e)$, respectively.

- physical cell: $\min \phi(\square_e) \geq 0$
- fictitious cell: $\max \phi(\square_e) < 0$
- boundary cell: neither a physical nor a fictitious cell

This classification is unique as it is based on the signed distance function and not on the parametric representation of the physical domain (e.g., via a surface mesh). The purpose of the cell tags is to identify and modify the shape functions which will be used for discretizing the weak form. To this purpose, all the b-spline shape functions are tagged as *active*, *inactive* or *semi-active*. This classification is based on the tags of all the nodes attached to a cell. In the following we denote the set of all cells around the control point (node) of a b-spline, center of the central cell for an odd b-spline degree, or center knot, for an even b-spline degree as the *one-ring of the B_i function*, and classify the shape functions as:

- active B_i: the one-ring contains only physical cells;
- inactive B_i: the one-ring contains only fictitious cells;
- semi-active B_i: the one-ring contains physical as well as fictitious cells.

The preceding definitions are used for defining a weighting function z_I associate with each node (or shape function) of the Cartesian mesh.

$$z_i(\xi) = \begin{cases} 0 & \text{if } B_i \text{ is inactive} \\ 1 & \text{if } B_i \text{ is semi-active} \\ w(x) & \text{if } B_i \text{ is active} \end{cases}. \tag{8}$$

The modified shape functions can be finally written as

$$N_i(\xi) = \frac{z_i B_i^n(\xi)}{\sum z_i(\xi) B_i^n(\xi)}. \tag{9}$$

ISSN 2176-5480

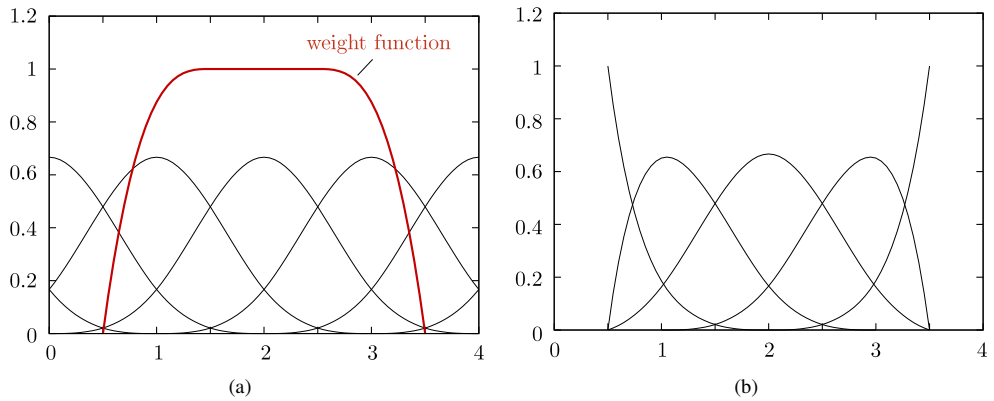


Figure 4. (a) Cubic b-splines and a smooth weight function for a domain $\Omega = (0.5, 3.5)$. (b) Modified shape functions computed with the proposed approach. Note that the modified shape functions are interpolating at the boundaries.

Figure 4 shows the construction of the modified shape functions for an one-dimensional domain $\Omega = (0.5, 3.5)$ and cubic b-spline shape functions. Figure 5 shows the original and modified shape functions for a two-dimensional problem. The i-splines keep most of the b-spline properties, such as:

- *Linear independence*: As the cardinal B-splines are linearly independent, and $w_i \neq a\frac{w_j B_j}{B_i}$ for i and $j = 1..nf$, where nf is the number of non zero N_i functions over any given cell, the functions N_i are also linearly independent.
- *Partition of unity*: The partition of unity is fulfilled due to the rationalization.
- *Positivity and compact support*: As w_i and B_i are always positive inside the physical domain,this property is fulfilled.
- *Smoothness*: As w_i^n is of the same order of continuity as the b-splines of degree n, B_i^n, the functions N_i^n, build based on these b-splines form a bases $(n-1)$-times continuously differentiable with discontinuities of the n-th derivative at the break points.

3.3 Iso-parametric geometry representation of immersed domain

Due to the functions being rational near the boundaries, the new basis can not represent exactly linear functions over the space of coordinates $\xi = \xi_1, \xi_2, \xi_3$. However it is possible over another space of coordinates $\mathbf{x} = x_1, x_2, x_3$, since this coordinates are written as a linear combination of the shape functions N_i.

$$\mathbf{x}^h(\xi) = \sum_{i=1} \mathbf{x}_i N_i(\xi) \tag{10}$$

For simplicity, let us consider a linear function in one dimension, $f(x) = ax^h + b$ (with a and b constants). This function may be approximated by the i-spline shape functions $N_i(\xi)$ and, although this approximation is not linear over ξ, it is still linear over the iso-parametric approximation x.

$$f^h(\xi) = \sum_{i=1} f_i N_i(\xi) = \sum_{i=1} (ax_i + b) N_i(\xi) = a \sum_{i=1} x_i N_i(\xi) + b. \tag{11}$$

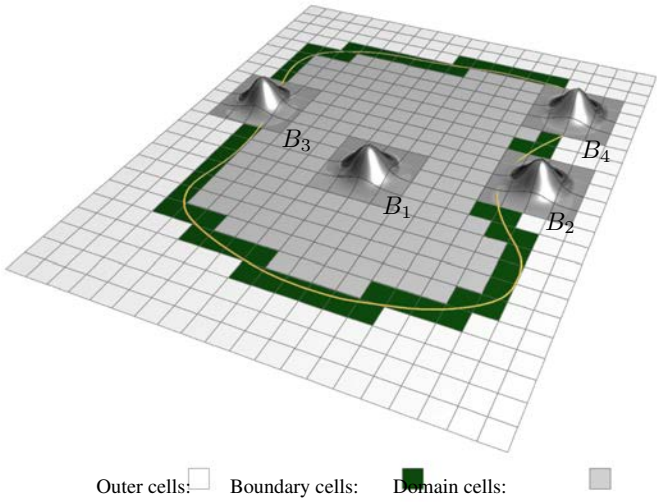
In our approach, the Cartesian space is considered to be a parametric domain of coordinates ξ, and the physical domain is obtained by Eq. 10. For active shape functions, the $\mathbf{x}_i$ values are the coordinates vector ξ_i of the node (control point) associated with the respective B-spline. However, it is necessary to make the boundary of the physical match with the boundary of the parametric (Cartesian) domain defined by level-set. There are different options are found for this purpose (see Sanches *et al.* (2011), and the one we employ consist on taking $\mathbf{x}_i$ for the semi-active shape functions as the boundary closest point to its related B-spline node.

This iso-parametric representation also makes easy the numerical integration over boundary cells.

3.3.1 Numerical quadrature over boundary cells

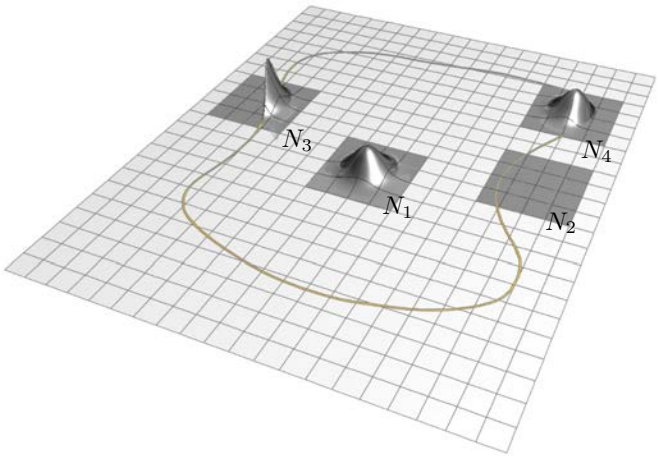
Although it is possible to integrate on the physical cells directly with an ordinary Gauss quadrature, this cannot be performed on boundary cells characterized in Fig. 6 by $\bigcirc_e \subset \square_e$. However, the iso-parametric geometry mapping allows to consider each boundary cell as a re-shaped Cartesian cell.

ISSN 2176-5480



Active B-splines: B_1 and B_4. Semi-active B-spline: B_3. Inactive B-spline: B_2.

(a) 2D b-splines classification.



(b) 2D shape functions based on cubic b-splines.

Figure 5. (a) Original b-spline shape functions which do not conform to the physical domain boundary. (b) Modified shape functions which are interpolating at the boundaries. Note that $N_1 \equiv B_1$, $N_2 \equiv 0$, $N_3 \neq B_3$ and $N_4 \neq B_4$.

For this purpose, Sanches *et al.* (2011) employs additional element-wise parameters $\mathbf{S}_e$ which are mapped to ξ using cubic B-splines. Considering ξ as the cell local coordinates on the Cartesian parametric domain, with cell sizes $h_{\xi_1} = h_{\xi_2} = h_{\xi_3} = 1$, x^c as the coordinates on the Cartesian domain with physical dimensions, with cells size h_1, h_2 and h_3, and $\mathbf{x}^h$ as the coordinates on physical domain. the grid with real physical dimensions, we can consider the cells on $\mathbf{S}_e$ reshaped parametric domain with size $h_{S_e} = 1$, cf. Fig. 6, and map to ξ using the cubic b-splines, so that a new local position on the Cartesian domain $\tilde{\xi}_g$ is set to the gauss points by

$$\tilde{\xi}_g = \xi(\mathbf{S}_{eg}) = \sum B_i(\mathbf{S}_{eg})(\xi_i + \mathbf{G} \cdot (\mathbf{x}_i^h - \mathbf{x}_i^c), \tag{12}$$

where $G = \text{diag}(\frac{h_{\xi_1}}{h_1}, \frac{h_{\xi_2}}{h_2}, \frac{h_{\xi_3}}{h_3})$. The new weigh for the Gauss point $\tilde{\xi_g}$ is given by $\tilde{\omega}_g = \det(\xi_{,\mathbf{S}_e})\tilde{w}_g$, where $\mathbf{S}_{eg}$ and w_g are the original rectangular gauss point and weight. The use of B-splines for mapping is very robust, as degenerated

ISSN 2176-5480

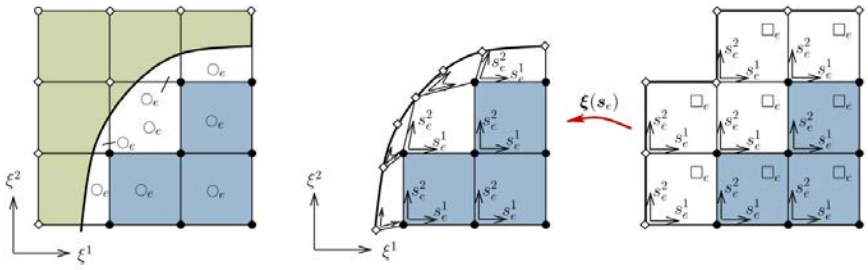


Figure 6. Parametric mapping for numerical quadrature adapted from Sanches *et al.* (2011).

cells, cells taking on the shape of a triangle for instance are tolerated. However, the mapping using B-splines smoothes the boundary and make some gauss points outside the physical domain, where the shape function values are set to zero. It may produce errors, which relatively increases when the mesh is refined, lowering the convergence rate.

To avoid such problems, we employ a linear mapping, so that we have a parametric domain defined by the moved points reshaped using linear approximation which matches exactly to the physical domain after interpolated by the i-splines, so that Eq. 12 becomes:

$$\tilde{\xi}_g = \xi(\mathbf{S}_{eg}) = \sum L_i(\mathbf{S}_{eg})(\xi_i + \mathbf{G} \cdot (\mathbf{x}_i^h - \mathbf{x}_i^c), \tag{13}$$

where L_i is a linear, or bilinear, shape function associated to the node i. It is important to notice that this transformation using linear approximation is employed only between parametric spaces, as the physical space is still defined by Eq. 10. This procedure is automatic using bilinear shape functions defined over the parametric Cartesian grid for convex shapes, however, for concave shapes a boundary cell may become a triangle or even worse. For 2D it can be easily solved by subdividing the boundary cells by the smallest diagonal (see Fig. 7), generating two triangles with triangular quadrature points, which can be reshaped with no problem. For 3D a more sophisticated integration may be employed, as the one presented by Gueziec and Hummel (1995).

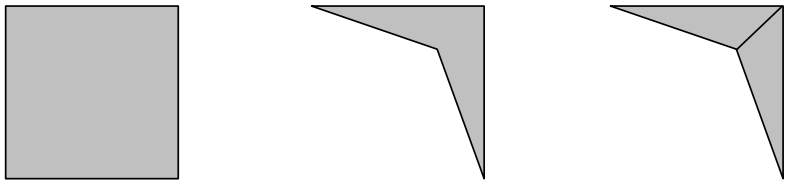


Figure 7. Boundary cell subdivision for numerical integration.

4. Convergence

To establish the convergence characteristics of the method we make use of one- and two-dimensional elasticity and Poisson boundary value problems. The error norms used for the convergence tests are:

$$e_{L_2}^h = \|\mathbf{u}^h - \mathbf{u}\|_{L_2(\Omega)} = \sqrt{\int_\Omega (\mathbf{u}^h - \mathbf{u}) \cdot (\mathbf{u}^h - \mathbf{u}) d\Omega} \tag{14}$$

and

$$e_{H_1}^h = \|\mathbf{u}^h - \mathbf{u}\|_{H_1(\Omega)} = \sqrt{\int_\Omega (\nabla\mathbf{u}^h - \nabla\mathbf{u}) : (\nabla\mathbf{u}^h - \nabla\mathbf{u}) d\Omega}. \tag{15}$$

4.1 Patch test and convergence

In essence, if the deformations and constant stresses are correctly predicted, the patch test is passed, and, as at least constant stresses can be correctly predicted, convergence rate in stresses (or in strains for linear elasticity) will be at least

ISSN 2176-5480

$O(h)$. In the same way, a given discretization will present stresses convergence rate $O(h^2)$ if a linear stress variation is correctly represented (Bathe, 1995).

Due to the local modifications, immersed methods may present a smaller convergence rate compared to the same polynomial degree for standard finite elements, as one may observe from Höllig *et al.* (2002) and Ausas *et al.* (2010). As the method under investigation is able to represent constant and linear fields, we expect a convergence rate at least linear ($O(h)$) for the H_1 norm and quadratic ($O(h^2)$) for L_2 norm.

5. LINEAR ELASTICITY

The finite element solution for elasticity may be obtained from minimization of the total energy functional given by:

$$\Pi(u) = \frac{1}{2}\int_\Omega \mathbf{T}\colon \boldsymbol{\varepsilon} d\Omega - \int_\Omega \mathbf{b}^T\mathbf{u} d\Omega - \int_{\Gamma_N} \mathbf{f}^T\mathbf{u} d\Gamma_N, \tag{16}$$

where $\mathbf{T}$ is the Cauchy stress tensor, ε is the engineering strain pseudo-tensor, $\mathbf{b}$ is the body forces vector and $\mathbf{f}$ is the traction vector. As the stress tensor and the strain pseudo-tensor are linked by the elastic constitutive law (Hooke's Law as considered here), the functional is dependent on the displacements fields and external forces.

The variational problem solution is to find one displacements function $\mathbf{u}(\mathbf{x})$ which minimizes 16 and respects the Dirichlet boundary conditions Assan (2003).

The Ritz method consists of the choice of one trial function $\tilde{\mathbf{u}}(\mathbf{x})$ which agrees with the boundary conditions and has adjustable parameters. The adjustable parameters are chosen so that the functional is minimized.

The trial function may be written as a linear of the proposed shape functions as:

$$\mathbf{u}(\mathbf{x}) \approx \tilde{\mathbf{u}}(\mathbf{x}) \sum_{i=1}^{i=n} \alpha_i \phi_i(\mathbf{x}). \tag{17}$$

where α_i's are the control points of the displacements fields, and at same time the adjustable parameters.

6. CONVERGENCE STUDY

6.1 1D problem

In order to start the convergence study, we simulate the problem of one elastic bar of length $L = 1$, Young's modulus $E = 1$ and cross-section area $A = 1$, fixed on both ends. One axial distributed force according to Eq. 18 is applied.

$$f = sin(2x) \tag{18}$$

The differential that describes this problem is:

$$\frac{u^2}{x^2} + sin(2x) = 0 \; \Omega = (a, b), \tag{19}$$

where u is the axial displacement. The analytical solution is given by:

$$u(x) = \frac{1}{4}sin(2x) + \frac{sin(2b) - sin(2a)}{4L}x - \frac{Lsin(2a) + asin(2a) - asin(2b)}{4L}. \tag{20}$$

For the numerical solution we employed 19 Gauss-Legendre quadrature points by elements in order tom minimize numerical integration error. The weight function employed was cubic in order to maintain the smoothness of the b-splines. The boundary cell quadrature points were linearly displaced to inside the domain.

In the first step we analyzed the influence of the boundary cell cutting position, starting by placing the bar centered in a 1D mesh with domain from -1.0 m to 1.0 m, divided initially into 4 elements. Following the bar length was changed. This analysis showed that the error is minimum when the boundary cells are cut by the middle and maximum when the boundary are over some cell end (node) (see Fig. 6.).

In the following step we performed one convergence analysis in which the boundaries are always over some nodes, and another convergence analysis in which the boundary cells are cut in the middle.

In figure 9 one may see the errors (e^h_{H1} and e^h_{L2}), where both simulations are found with convergence order O(2h) for the H1 norm and O(3h) for the L2 norm. The error for a given discretization should be placed between the superior limit (when the boundary cell is cut on is edge) and the inferior limit (when the boundary cell is cut in the middle). This makes us not to expect monotonic convergence for any problems.

The error reduction when cutting the cell in the middle may be understood as and adequate balance between the domain portion with larger number of shape functions and the reduction of the influence of the semi-active shape functions on physical domain.

ISSN 2176-5480

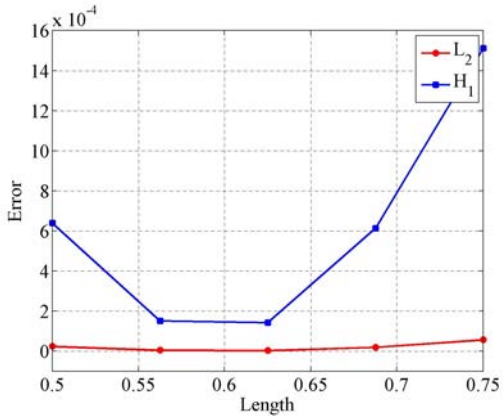


Figure 8. Error norms for (a) bar moved over the fixed grid; (b) Length variation with fixed grid .

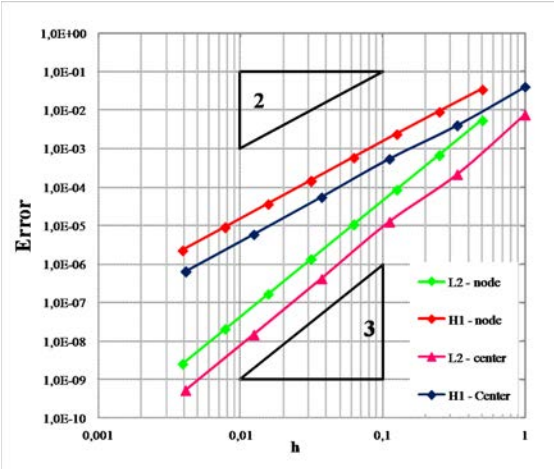


Figure 9. Elastic Bar Convergence

6.2 Elastic ring

For a 2D convergence analysis we selected the problem of an elastic ring with internal pressure, assumed to be under plane stress state. This problem consists of an elastic ring with external radius $R_e = 2$ and internal radius $R_i = 1$, according to figure 8, subjected to an internal radial displacement $u_{r0} = 1$, with null tension and body forces.

The physical properties are: Young's Modulus $E = 1$ and Poisson ration $\nu = 0$. The analytic solution for the axi-symmetric problem is given by (See Timoshenko (1970)):

$$u_r = \frac{R_i}{u_{r0} cos(\theta)} R_i^2 + R_e^2 \left(\frac{R_e^2}{r} + r \right). \tag{21}$$

Two different analysis were considered. In the first we employed 100 Gauss points by cell with no cell subdivision during integration, while in the second we divided the boundary cells into two triangles as explained in section 3.3.1

From figure 10 we conclude that by employing adequate numerical quadrature (subdividing boundary cells), cubic convergence is achieved for the error norm L_2 and quadratic for the error norm H_1.

The displacements and normal stress fields are shown in figures 11 and 12.

7. Other Numerical example

7.1 Diametral compression test

This is an interesting test as it has a load applied in a very small area, and is a Brazilian Standard test employed for determining concrete tension strength.

ISSN 2176-5480

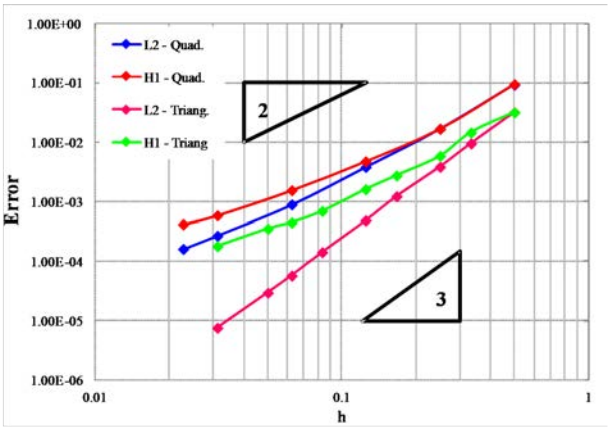
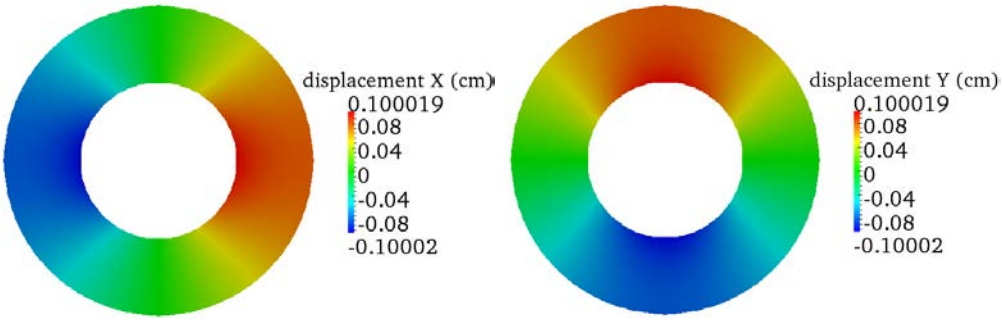
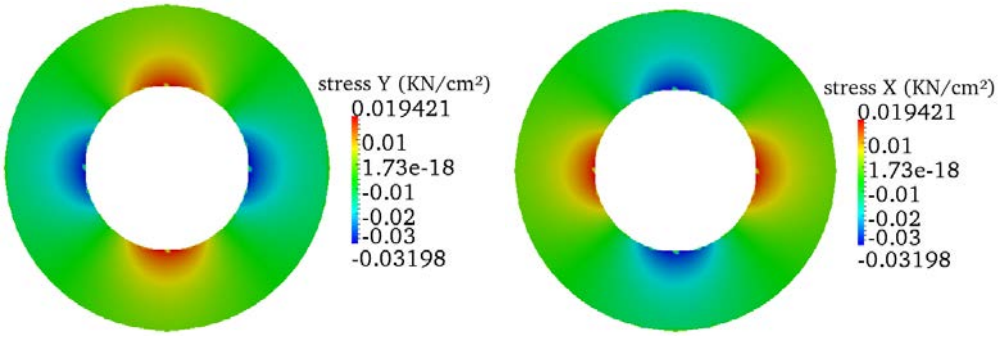


Figure 10. Ring Convergence



(a) Horizontal displacement (b) Vertical displacement

Figure 11. Ring Displacements



(a) σ_x (b) σ_y

Figure 12. Ring Normal Stresses

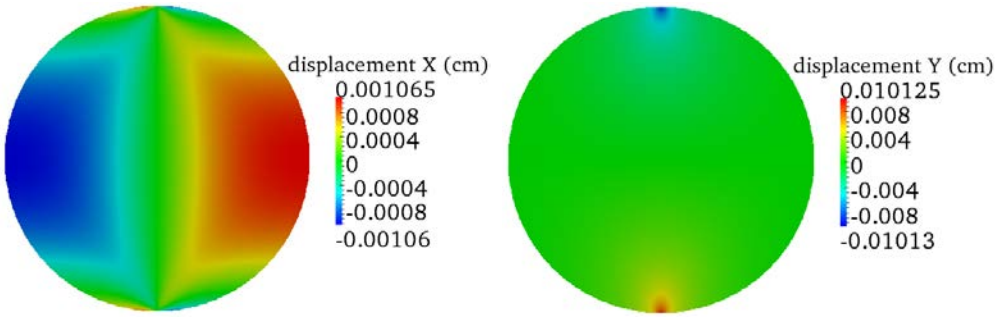
We considered one cylinder with diameter $d = 5$ cm, Young's Modulus $E = 28$ GPa and Poisson Ratio $\nu = 0.27$. To this cylinder we applied a dimetral compression point load $P = 20$ kN. This problem was discretized using a square mesh with dimensions 5.5 cmx5.5 cm, with 10000 elements.

The resulting displacements are depicted in figure 13. The maximum shear stresses contour is compared to the photoelastic results presented by **?**, showing excellent agreement.

7.2 Bicicle chain link

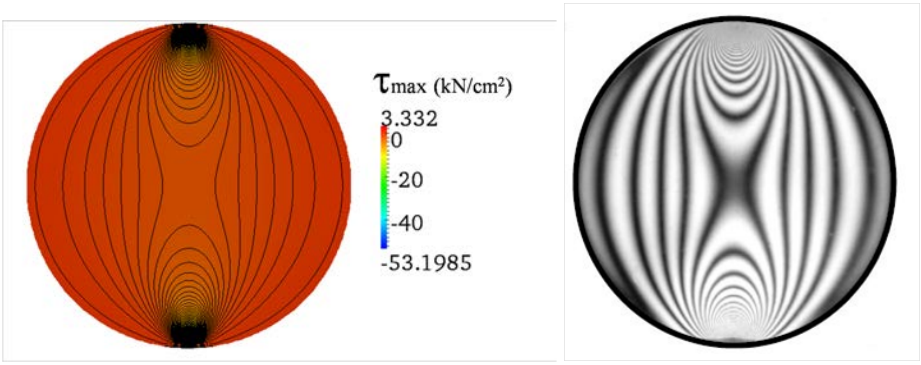
Finally we considered a bicycle chain link model Shimano CN-6700, clamped on the left hole and with an horizontal displacement of 0.5 mm in the write one. The material properties are Young's Modulus $E = 105$ GPa and Poisson Ration $\nu = 0.3$.

ISSN 2176-5480



(a) Horizontal displacement (b) Vertical displacement

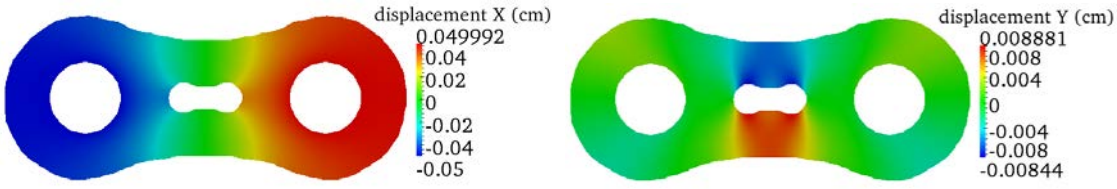
Figure 13. Cylinder Displacements



(a) Present work - Numerical results (b) Martin (2005) - Photoelastic results

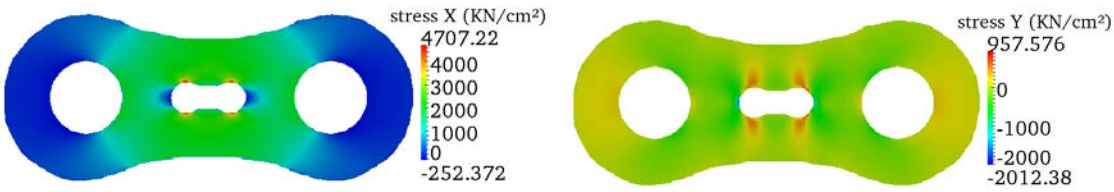
Figure 14. Maximum Shear Stress Contour

We employed a rectangular mesh of dimensions 12 mm x 24 mm with 60x120 divisions. The displacements and the normal stresses are shown respectively in figures 15 and 16. This results show the robustness of the proposed method to deal with complex shapes.



(a) Horizontal displacement (b) Vertical displacement

Figure 15. Bicycle Chain Link Displacements



(a) σ_x (b) σ_y

Figure 16. Bicycle Chain Link Normal Stresses

ISSN 2176-5480

8. CONCLUSIONS

From the present study we confirm the efficiency and robustness of the method introduced by Sanches *et al.* (2011) when applied to 2D elasticity and conclude that it reaches quadratic H_1 convergence rate and cubic L_2 convergence rate. For future studies we consider application of the proposed finite elements basis to other mechanical problems, including problems with moving boundaries.

ACKNOWLEDGEMENTS

The authors would like to thank the Brazilian council CNPq (*Conselho Nacional de Desenvolvimento Científico e Tecnológico*) and the State of Paraná Foundation *Fundação Araucária* for the financial support.

9. REFERENCES

Assan, A.E., 2003. *Método dos Elementos Finitos - Primeiros Passos*. Editora Unicamp, Campinas, SP, Brasil, 2nd edition.

Ausas, R.F., Sousa, F.S. and Buscaglia, G.C., 2010. "An improved finite element space for discontinuous pressures". *Computer Methods in Applied Mechanics and Engineering*, Vol. 199, No. 17-20, pp. 1019 – 1031.

Bathe, K.J., 1995. *Finite Element Procedures*. Prentice Hall, 2nd edition.

Belytschko, T., Parimi, C., Moës, N., Sukumar, N. and Usui, S., 2003. "Structured extended finite element methods for solids defined by implicit surfaces". *International Journal for Numerical Methods in Engineering*, Vol. 56, No. 4, pp. 609–635. ISSN 1097-0207. doi:10.1002/nme.686. URL `http://dx.doi.org/10.1002/nme.686`.

Glowinski, R., Pan, T.W. and Periaux, J., 1994. "A fictitious domain method for dirichlet problem and applications". *Computer Methods in Applied Mechanics and Engineering*, Vol. 111, No. 3-4, pp. 283 – 303. ISSN 0045-7825. doi:DOI: 10.1016/0045-7825(94)90135-X. URL `http://www.sciencedirect.com/science/article/pii/004578259490135X`.

Gueziec, A. and Hummel, R., 1995. "Exploiting triangulated surface extraction using tetrahedral decomposition". *Visualization and Computer Graphics, IEEE Transactions on*, Vol. 1, No. 4, pp. 328 –342.

Höllig, K., Reif, U. and Wipper, J., 2002. "Weighted extended b-spline aproximation of dirichlet problems". *SIAM Journal on Numerical Analysis*, Vol. 39, No. 2, pp. 442–462.

Hughes, T., Cottrell, J. and Bazilevs, Y., 2005. "Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement". *Computer Methods in Applied Mechanics and Engineering*, Vol. 194, pp. 4135–4195.

L. W. Kantorowitsch, W.I.K., 1964. *Approximate Methods of Higher Analysis*. Interscience Publishers (New York), translated from the 4th rusian edition edition.

Piegl, L. and Tiller, W., 1997. *The NURBS book*. Monographs in visual communication. Springer.

Rogers, D., 2001. *An introduction to NURBS*. Academic Press.

Sanches, R., Bornemann, P. and Cirak, F., 2011. "Immersed b-spline (i-spline) finite element method for geometrically complex domains". *Computer Methods in Applied Mechanics and Engineering*, Vol. 200, No. 13-16, pp. 1432 – 1445.

Strang, G. and Fix, G., 2008. *An analysis of the Finite Element Method*. Wesley-Cambridge Press, 2nd edition.

Timoshenko, S., 1970. *Theory of elasticity*. McGraw-Hill Higher Education, 3rd edition.