ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

DOMAIN DECOMPOSITION SOLUTION SCHEMES FOR LARGE-SCALE IGA PROBLEMS

George Stavroulakis[1], **Dimitris Tsapetis**[1] **and Manolis Papadrakakis**[1]

[1] Institute of Structural Analysis and Seismic Research, National Technical University of Athens 9, Iroon Polytechneiou, Zografou Campus, GR-15780 Athens, Greece
e-mail: stavroulakis@nessos.gr

Keywords: Isogeometric Analysis, Refinement, PCG, IETI, Solutions scheme, Refinement, Mapping.

Abstract. *Isogeometric Analysis (IGA) [5] is a promising concept that establishes a close link between the technologies of CAD (computer aided design) and numerical simulation via finite element analysis (FEA). In the IGA framework, the same function spaces, which are used for the geometric representation of the computational domain, are used for the approximation of the problem unknowns. There are several computational geometry technologies that could serve as a basis for IGA with Non-Uniform Rational B-Splines (NURBS) being the most widely adopted due to their popularity in CAD software. In contrast with FEA where there is a very broad spectrum of solution techniques for the fast and efficient solution of the linear or linearized systems that occur [1-4], IGA solution schemes are still an open issue. With respect to domain decomposition techniques, the Isogeometric Tearing and Interconnecting (IETI) [8] method combines the advanced solver design of dual domain decomposition methods with the exact geometry representation of IGA, relying on patches for the subdivision of the domain. In this work, an innovative solution scheme is presented, showcasing greatly enhanced performance when compared to the established and tested solution schemes for IGA and its numerical performance is exhibited in numerical examples.*

1 SOLUTION TECHNIQUES

Iterative methods to solve a linear system such as Conjugate Gradient (CG) method [1, 2] are commonly used with the aid of a preconditioner [3, 4] to ensure fast convergence of the method. In our case, we employ a Preconditioned Conjugate Gradient (PCG) method in which the preconditioner is not a matrix as commonly thought of in numerical analysis, but a transformation in the wider mathematical sense. In the following sections, we will explain the need that led us to the proposal of a transformation, compare two different approaches of the new "preconditioner" and explain thoroughly the optimal among the two.

1.1 Large – scale problems in IGA

IGA is a method that was initially proposed by Hughes et al. [5] and envelops both the geometric information of the structures as well as the analysis mesh and data in one single model. Since 2005, that first appeared, IGA has gained a wide acceptance by the scientific community as it gradually establishes its background and grows rapidly. This raises the need of researchers to solve constantly large and more complex problems utilizing this promising method. These problems are of the form:

$$Ku = f \tag{1}$$

where K is the stiffness matrix, f is the applied force field and u is the displacement field. Unfortunately, the major advantage of IGA i.e.: higher Continuity Shape Functions, becomes a disadvantage. For example, NURBS Shape Functions span through several Isogeometric Elements which inevitably leads to a denser Stiffness Matrix with a greater bandwidth in comparison to simple Finite Elements.

Since complexity of a solver increases with increasing continuity [6, 7] it is crucial for the further evolution of IGA, the development of an algorithm that will provide both high accuracy and minimized time complexity compared to the existing techniques. IETI that was proposed in [8], is a powerful method that divides the problem into a multitude of independent problems and thus reduces the complexity of a large scale system. The major drawback of this approach is that the number of subdomains must be identical to the number of patches present to an IGA model. This means that subdivisioning is mesh-dependent which makes both load balancing and scalability to suffer in high performance computing environments. On the other hand, inducing discontinuities in a model by subdividing it into a number of subdomains, increases the error and discontinuities in second-order characteristics like stress and strain.

1.2 The PCG-IETI method

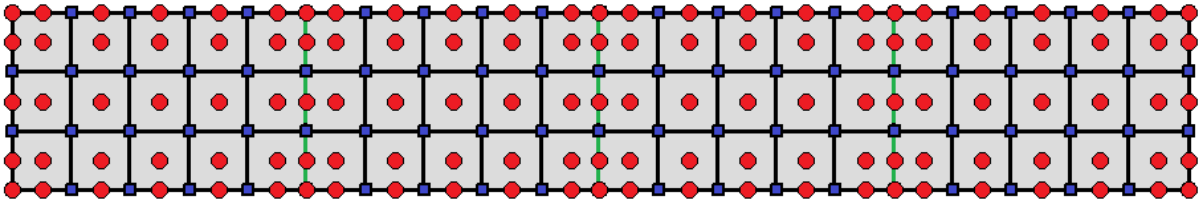
In order to circumvent this drawback, a new model with stiffness matrix K_p is introduced which we will refer to as "patched model" and has the geometry of the original model with appropriately induced discontinuities in the form of patches, in order to subdivide the original model appropriately. Due to the fact that there are errors introduced due to these discontinuities but the geometry of the problem is identical, the original and subdivided models constitute near-by problems with the stiffness matrix of the original model being equal to $K = K_p + \Delta K$.

Let's consider the PCG algorithm, equipped with a preconditioner following the rationale of incomplete Cholesky preconditionings which feature an error matrix E_i. This matrix is usually defined by the computed positions of small elements in the lower triangular matrix produced by the incomplete Cholesky factorization procedure, which do not satisfy a specified magnitude criterion and therefore are discarded [15]. Considering the near-by problem of the form $\left(K_p + \Delta K\right)u = f$, if matrix E_i is taken as ΔK, the preconditioning matrix becomes the initial matrix $\tilde{A} = K_p$. The repeated solutions required for the evaluation of the preconditioned residual vector $z^k = \tilde{A}^{-1} r^k$ are performed using IETI and are treated as problems with multiple right-hand sides.

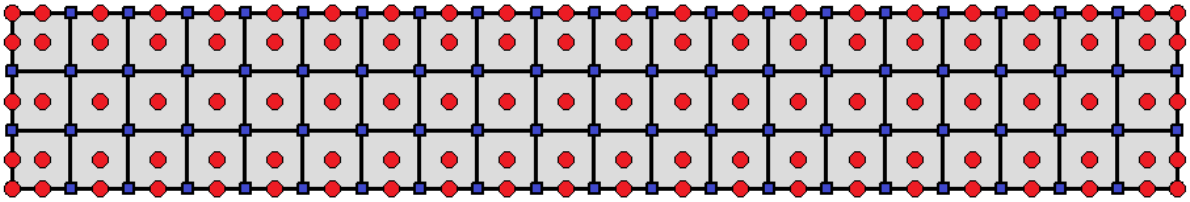
In contrast to models generated using FEA, IGA-generated models with the same geometry but different patches have different number of degrees of freedom (dof). This means that the residual vector r^k cannot be used in order to perform the evaluation of the preconditioned re-

sidual vector using matrix K_p. Even if that was possible, this evaluation would produce a vector with a different size compared to the vectors produced using matrix *K*. In order to circumvent this problem, a transformation between these two IGA meshes is employed. Specifically, with the aid of double transformation, the residual of the original model is being mapped to the patched model space and after the preconditioned residual vector evaluation, the solution is being mapped back to the original model space.

The PCG algorithm, equipped with the latter double-transformation preconditioner throughout the entire solution process, constitutes the PCG-IETI method providing both load balancing and scalability properties.



(a) Example of a Simple Cantilever split into four patches (patched model).



(b) Example of the same Cantilever modeled as a single patch (original model).

Figure 1: Cantilever examples. Black continuous lines illustrate the patch border and squares the Control Points in each case. Dotted lines depict the Control Net.

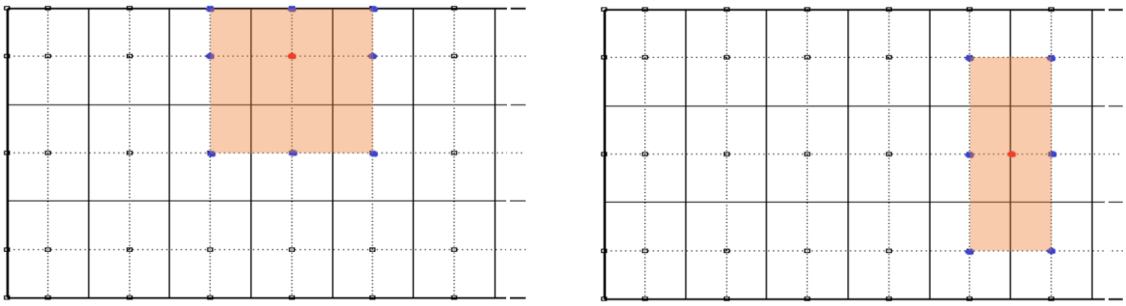
2 MAPPING

2.1 Shape Functions Mapping

Since the implementation of PCG-IETI requires a transformation between two different meshes, we utilized the intergrid transfer operator ***R*** - Restriction operator for the transfer of the fine grid information to the coarse one and the prolongation or interpolation Operator ***I*** for the transverse transfer from the coarse to the fine grid [10]. In our case, we use the full weighting Operator ***R*** [11] which is derives from interpolation Operator using:

$$\mathrm{R} = \mathrm{c} \cdot \mathrm{I}^{\mathrm{T}} \tag{2}$$

So by defining the interpolation operator ***I*** the transformation of the two meshes will be completed for application in the iterative solver. For finite elements purposes, a simple linear interpolation between the nodal values is adequate for the intergrid transfer, but since the application will be in the field of Isogeometric Analysis, the choice was to use a more complex interpolation that would provide better results taking into account the increased continuity of the Isogeometric method. Our choice is to use NURBS shape functions. The interpolation between the two control meshes is performed considering the Control Point parametric coordinates when the isoparametric domain in each direction spans in [0, 1]. Given the same parametric domain both coarse and fine mesh can transfer information between each other with simple interpolation.



(a) Approximation of an existing Control Point with the aid of NURBS interpolation. We observe that the point's value is defined with the aid of nine Control Points.

(b) Approximation of a non-existing Control Point with the aid of NURBS interpolation. We observe that the point's value is defined with the aid of six Control Points.

Figure 2: Domain of influence using NURBS interpolation. Rhinoceros [9] software was used.

In the next subsections we provide a brief description of Shape Function in Isogeometric Analysis.

2.1.1. B-Splines

Given a degree $p \geq 0$ of the shape functions and an ascending sequence $\Xi = \{\xi_1, \xi_2, ..., \xi_{n+p+1}\}$ of n+p+1 real values called Knot Vector, where n is the number of Control Points the B-Splines can be calculated using the recursive Cox-de Boor algorithm. Note that we only use open Knot Value Vectors which implies that the first and last Knot Value has multiplicity of p+1 and the internal values multiplicity m lower or equal to p. The recursive formulation of B-Splines starts with constant shape functions of degree p=0.

$$N_{i,0}(\xi) = \begin{cases} 1, & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

For degree p>0 B-Splines are defined using p-1 degree according to the following formula:

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} \cdot N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} \cdot N_{i+1,p-1}(\xi) \tag{4}$$

It must be noted that in the fractions of equation (3), when a denominator equals zero then the fraction's value is considered as zero.

2.1.2. NURBS

NURBS are produced using both B-Spline functions and the weights of the Control Points. NURBS curves are given by:

$$R_i^p(\xi) = \frac{N_{i,p}(\xi) \cdot w_i}{\sum_{\hat{i}=1}^{n} N_{\hat{i},p}(\xi) \cdot w_{\hat{i}}}$$
$$C(\xi) = \sum_{i=1}^{n} R_i^p(\xi) \cdot B_i \tag{5}$$

In a similar way we can define two and three dimensional NURBS Shape Function.

$$R_{i,j}^{p,q}(\xi,\eta)=\frac{N_{i,p}(\xi)\cdot M_{j,q}(\eta)\cdot w_{i,j}}{\sum_{\hat{i}=1}^{n}\sum_{\hat{j}=1}^{m}N_{\hat{i},p}(\xi)\cdot M_{\hat{j},q}(\eta)\cdot w_{\hat{i},\hat{j}}}$$
$$R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta)=\frac{N_{i,p}(\xi)\cdot M_{j,q}(\eta)\cdot L_{k,r}(\eta)\cdot w_{i,j,k}}{\sum_{\hat{i}=1}^{n}\sum_{\hat{j}=1}^{m}\sum_{\hat{k}}^{n}N_{\hat{i},p}(\xi)\cdot M_{\hat{j},q}(\eta)\cdot L_{\hat{k},r}\cdot w_{\hat{i},\hat{j},\hat{k}}} \quad (6)$$

The resulting univariate, bivariate or trivariate NURBS Shape Function are used for interpolation of the new Control Points with known parametric coordinates. Interpolation is used for approximation of both displacements and forces of the new Control Net using the following:

$$u_x=\sum N_i\cdot u_{x,i} \qquad u_y=\sum N_i\cdot u_{y,i} \qquad u_z=\sum N_i\cdot u_{z,i} \quad (7)$$

As eq. (6) shows that the approximation of the displacements of e.g. the patched Control Net is performed using the displacements and Shape Functions of the unpatched one and vice versa. In the case of forces, the transpose of the Shape Functions is used. Specifically, for calculation of the forces of the patched model, the forces of the unpatched are used with the transpose of the patched shape functions and vice versa which sums up to:

$$f_x=\sum N_i^T\cdot f_{x,i} \qquad f_y=\sum N_i^T\cdot f_{y,i} \qquad f_z=\sum N_i^T\cdot f_{z,i} \quad (8)$$

2.1.3. Cook Cantilever Example

In order to illustrate the performance of the NURBS Shape Function transformation, a cook cantilever was examined. The transformation had to transfer the information from the single-patch cantilever with 11x3 Control Points to a Cantilever subdivided into 9 patches with resulting 19x3 Control Points as shown in Fig. 1. By applying the transformation to this model we approximate displacements and forces. In Table 1 we can see how the approximation works by using norms of the resulting values compared to the initial ones. Even thought at first the approximated displacements seem close enough based on the total norms, the third and norm clearly depicts that an error up to 8% is induced using this technique.

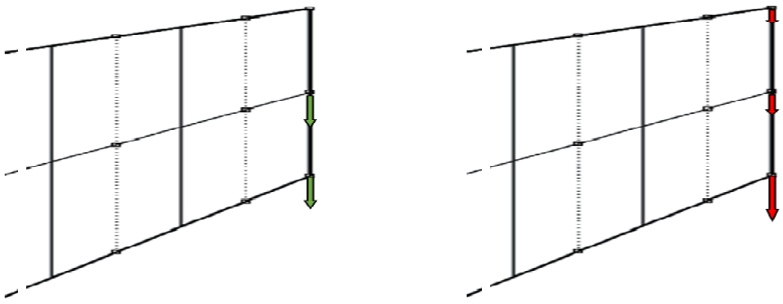
	Unpatched	Patched
$\text{norm}(U_{\text{exact}})$	0.1283	0.1561
$\text{norm}(U_{\text{approximate}})$	0.1294	0.1571
Error norm	7.15%	7.94%

Table 1: Performance of transformation on displacements using Shape Functions.

In case of forces, the error as shown in Table 2 dramatically increases. Unfortunately, this approach does not only create an error of the value but also changes the loading as clearly illustrated in Fig.3. This causes the algorithm either to converge real slow or in some cases even diverge due to the large error created in every intergrid approximation. In case of non-linear analysis, wrong estimation of both internal and external forces, can result to much greater error induced in the solution procedure and thus wrong results.

	Unpatched	Patched
$\text{norm}\left(F_{\text{exact}}\right)$	141.4214	141.4214
$\text{norm}\left(F_{\text{approximate}}\right)$	136.9306	136.9306
$\frac{\text{norm}\left(F_{\text{exact}} - F_{\text{approximate}}\right)}{\text{norm}\left(F_{\text{exact}}\right)}$	43.3%	43.3%

Table 2: Performance of transformation on forces using Shape Functions.



(a) Loads on a Cantilever before transformation

(b) Load on the Cantilever after the transformation

Figure 3: Changes of the Cantilever Loading due to the transformation. Rhinoceros [9] software was used.

2.2 Isogeometric Refinement Mapping

The second approach of the mapping was based on Isogeometric Refinement. Since our goal is to transfer information between two meshes that are based on the same structure, refinement as proposed in [5] can be considered the best tool applicable. It is used to calculate coordinates and weights of the new Control Points keeping the geometry unchanged, so it seems logical to approximate the data needed using this method. In [12] it is clearly illustrated how a structure can be analyzed using multiple separate patches, yet it is very important to explain the link with Isogeometric refinement in our case. Following the inverse procedure of [12] we want to segment one single patch into multiple individual subdomains for the sake of our iterative solver.

2.2.1. h-Refinement

The refinement algorithm that will be used in the aforementioned transformation is h-refinement or knot insertion. For a known knot value vector $\Xi=\{\xi_1, \xi_{2,}\ldots,\xi_{n+p+1}\}$ we want to introduce new knot values on increase the multiplicity of already known ones. For each one of the new inserted knot values the number of control points is increased by one. Their coordinates and weights $\bar{B}$ are computed

$$\bar{B} = T^p \cdot B \tag{9}$$

Where

$$T_{ij}^0(\xi) = \begin{cases} 1, & \text{if } \xi_j \le \bar{\xi}_i < \xi_{j+1} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

and

$$T_{ij}^{q+1} = \frac{\bar{\xi}_{i+q} - \xi_j}{\xi_{j+q} - \xi_j} \cdot T_{ij}^q + \frac{\xi_{j+q+1} - \bar{\xi}_{i+q}}{\xi_{j+p+1} - \xi_{j+1}} \cdot T_{ij+1}^q \tag{11}$$

2.2.2. Multiple patches segmentation

In Fig.4 we observe the 1D shape functions of a structure model as one patch. It is known from[13]that NURBS [13] Shape functions have a continuity of C^{p-m} over knots, where p is the polynomial degree of the function and m is the multiplicity of the knot. Fig. 4 shows NURBS of degree p=2 with C^1 continuity on knots. Using Knot Insertion algorithm of Isogeometric Refinement we raise the multiplicity of selected knots creating a C^0 continuity resulting in Fig. 5.

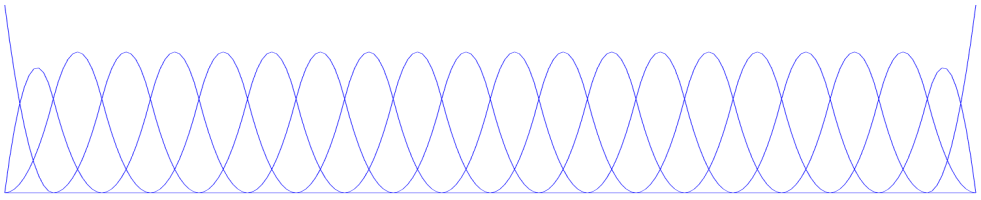


Figure 4: 1D NURBS of one single patch.

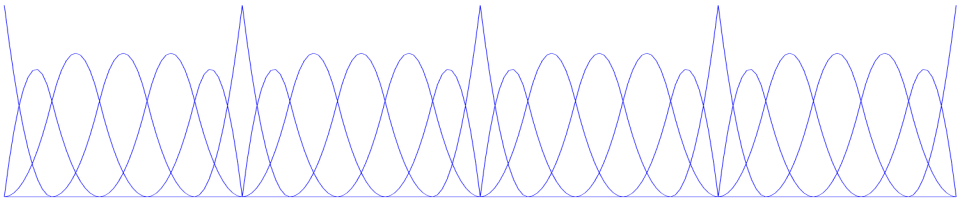


Figure 5: 1D NURBS of one single patch with C^0 continuity over selected knots.

In Fig.6. we observe the domain after the knot value insertion resembles to several independent subdomains united as one by merging their C^{-1} boundary shape function into a new C^0 one. In addition to the shape function the boundary Control Points between the two subdomains coincide. Considering that they have an independent role for each subdomain we result in Fig.7. where now every patches are independent and when combined with its data can be processed as a separate entity like in Fig.8. only take into account the boundary interconnection with other subdomains.

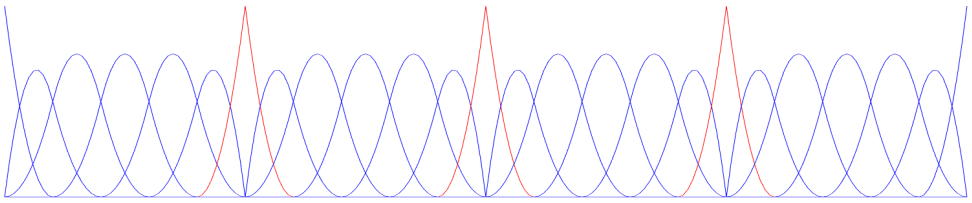


Figure 6: 1D NURBS of one single patch with C^0 continuity over selected knots.

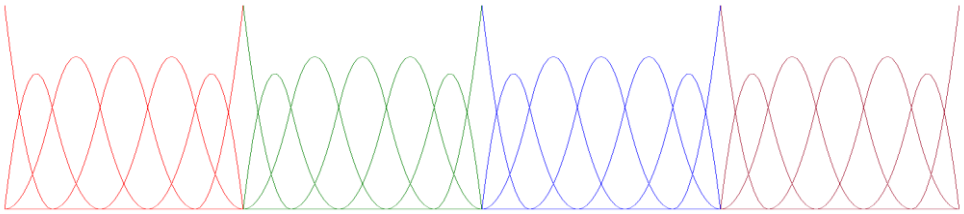


Figure 7: 1D NURBS of one single patch with C^0 continuity over selected knots.

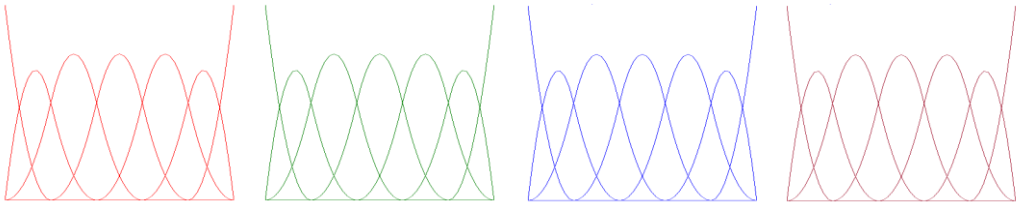


Figure 8: 1D NURBS of four discrete patches

2.2.3. Refinement transformation

As described above the Control Points of the segmented structure are given by h-refinement of the single-patched domain. The transformation matrix in this 1D case as described in [5, 12, and 13] is:

$$\underset{(mx3)}{\left\{P_{(\xi)}^{Multipatch}\right\}} = \underset{(mxn)}{\left[T^{MS}\right]} \cdot \underset{(nx3)}{\left\{P_{(\xi)}^{Singlepatch}\right\}} \tag{12}$$

In order to create the inverse transformation, we need to remove the inserted knot values. Since they are removable without altering the geometry a matrix is needed that will provide us with the inverse transformation.

$$\underset{(nx3)}{\left\{P_{(\xi)}^{Singlepatch}\right\}} = \underset{(nxm)}{\left[T^{SM}\right]} \cdot \underset{(mx3)}{\left\{P_{(\xi)}^{Multipatch}\right\}} \tag{13}$$

Using eq.11 and since knot values are removable we have:

$$\begin{aligned}
&\underset{(mx3)}{\left\{P^{Multipatch}_{(\xi)}\right\}} = \underset{(mxn)}{\left[T^{MS}\right]}\cdot\underset{(nx3)}{\left\{P^{Singlepatch}_{(\xi)}\right\}} \Rightarrow \\
&\underset{(nxm)}{\left[T^{MS}\right]^{T}}\cdot\underset{(mx3)}{\left\{P^{Multipatch}_{(\xi)}\right\}} = \underset{(nxn)}{\left(\underset{(nxm)}{\left[T^{MS}\right]^{T}}\cdot\underset{(mxn)}{\left[T^{MS}\right]}\right)}\cdot\underset{(nx3)}{\left\{P^{Singlepatch}_{(\xi)}\right\}} \Rightarrow \\
&\underset{(nx3)}{\left\{P^{Singlepatch}_{(\xi)}\right\}} = \underset{(nxn)}{\left(\underset{(nxm)}{\left[T^{MS}\right]^{T}}\cdot\underset{(mxn)}{\left[T^{MS}\right]}\right)^{-1}}\cdot\underset{(nxm)}{\left[T^{MS}\right]^{T}}\cdot\underset{(mx3)}{\left\{P^{Multipatch}_{(\xi)}\right\}}
\end{aligned} \tag{14}$$

So the inverse transformation can be created using the initial one by:

$$\underset{(nxm)}{\left[T^{SM}\right]} = \underset{(nxn)}{\left(\underset{(nxm)}{\left[T^{MS}\right]^{T}}\cdot\underset{(mxn)}{\left[T^{MS}\right]}\right)^{-1}}\cdot\underset{(nxm)}{\left[T^{MS}\right]^{T}} \tag{15}$$

This transformation matrix is one dimensional so it can be applied to each set of Control Points on axis Ξ. In case, we want to apply a refinement in more than one axes, a new equivalent transformation matrix for each refined axis must be created and consecutively applied in order to create the final Control Net. It is important to note that the null difference of the loads is a property that derives from knot insertion in IGA and thus it not an approximation but an exact transformation. Specifically, for the Shape Functions we know:

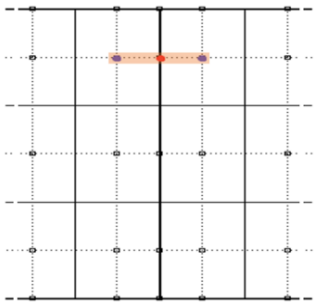
$$\underset{(mx1)}{\left\{P^{Multipatch}_{(\xi,\eta,\zeta)}\right\}} = \underset{(mxn)}{\left[T^{MS}\right]^{T}}\cdot\underset{(nx1)}{\left\{P^{Singlepatch}_{(\xi,\eta,\zeta)}\right\}} \tag{16}$$

The calculation of loads on the multipatch domain is performed by:

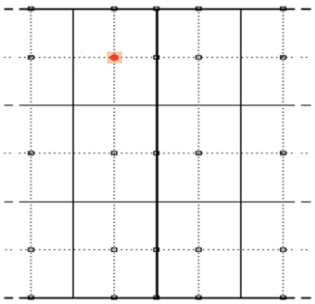
$$\begin{aligned}
&\underset{(mx3)}{\left\{L^{Multipatch}\right\}}\int_{\Omega}\underset{(mx1)}{\left\{R^{Multipatch}_{(\xi,\eta,\zeta)}\right\}}\underset{(1x3)}{f\left(\xi,\eta,\zeta\right)}\det\left[J\right]d\Omega = \\
&\int_{\Omega}\underset{(mxn)}{\left[T^{MS}\right]^{T}}\underset{(nx1)}{\left\{R^{Single-patch}_{(\xi,\eta,\zeta)}\right\}}\underset{(1x3)}{f\left(\xi,\eta,\zeta\right)}\det\left[J\right]d\Omega = \\
&\underset{(mxn)}{\left[T^{MS}\right]^{T}}\int_{\Omega}\underset{(nx1)}{\left\{R^{Single-patch}_{(\xi,\eta,\zeta)}\right\}}\underset{(1x3)}{f\left(\xi,\eta,\zeta\right)}\det\left[J\right]d\Omega \Rightarrow \\
&\underset{(mx3)}{\left\{L^{Multipatch}\right\}} = \underset{(mxn)}{\left[T^{MS}\right]^{T}}\cdot\underset{(nx3)}{\left\{L^{Single-patch}\right\}}
\end{aligned} \tag{17}$$

Finally, Fig. 8 shows the influence to the approximation of the new mesh. In case of a new Control Point its value is an interpolation of the Control Points closest to the inserted knot value and only on the axis knot insertion is performed. On the other hand, when a Control Point is not in proximity to an h-refinement its value and position is identical to its previous state. Fig.9 and 10 illustrate how NURBS affect the data continuity. For instance, in Fig. 9 examining the yellow element separation line of axis Ksi, we observe that full continuity allows NURBS to spans smoothly across many elements. On the other hand, when the same yellow line is distorted in Fig.10 by raising the multiplicity of the knot value, only one shape function now connects different subdomains. In addition, as expected the insertion of a knot

value resulted to the insertion of a single shape function in axis Ksi. Distortion of the single patch mesh only occurred on the subdomain interface while all other Control Points remained intact. In case we had a larger degree of shape functions this change would have also affected Control Points close to the knot inserted by changing their coordinates and weights.



(a) *Approximation of a non-existing Control Point with the aid of Refinement Transformation. We observe that the point's value is defined with the aid of two Control Points.*



(b) Approximation of an existing Control Point with the aid of Refinement Transformation. We observe that the point's value is retained to the identical Control Point of the previous mesh.

Figure 8: Domain of influence using Refinement Transformation.

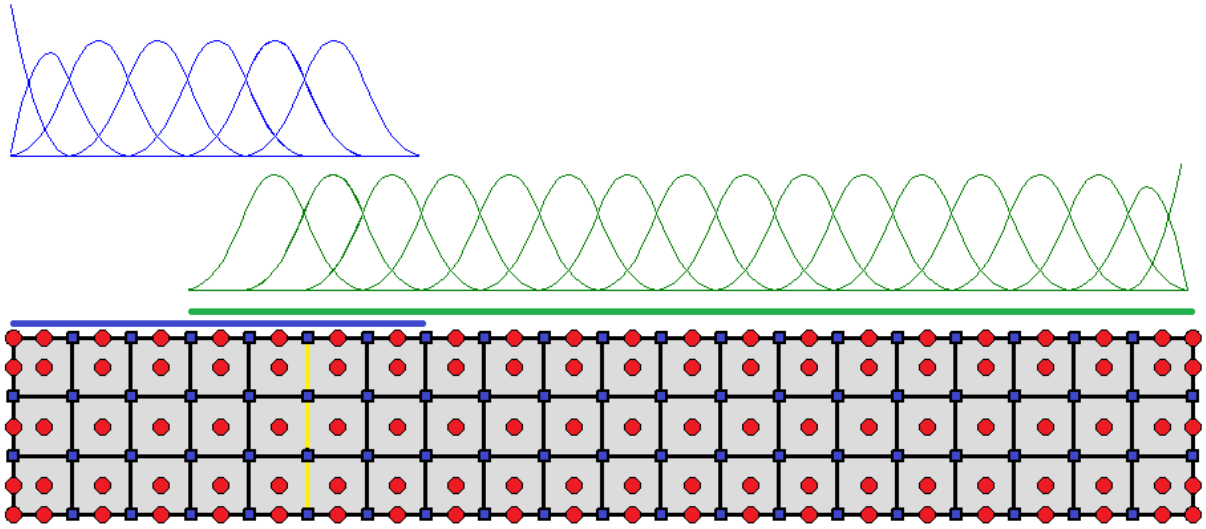


Figure 9: Shape functions and continuity of a single patch.

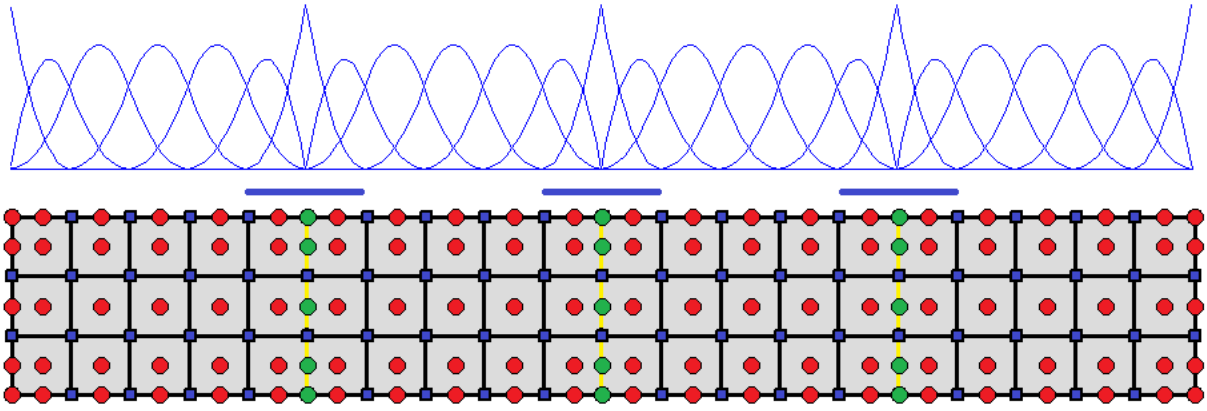


Figure 10: Shape functions and continuity of multiple patched structure.

2.2.4. Cook Cantilever Example

The same cantilever example of 1.2.3 will be used here in order to compare the results of the two approaches.

	Unpatched	Patched
$\text{norm}\left(U_{\text{exact}}\right)$	0.1283	0.1561
$\text{norm}\left(U_{\text{approximate}}\right)$	0.1284	0.1560
Error norm	0.23%	0.34%

Table 3: Performance of transformation on displacements using Refinement Transformation Matrix

	Unpatched	Patched
$\text{norm}\left(F_{\text{exact}}\right)$	141.4214	141.4214
$\text{norm}\left(F_{\text{approximate}}\right)$	141.4214	141.4214
Error norm	0%	0%

Table 4: Performance of transformation on forces using Refinement Transformation Matrix

It is clear given the two comparison matrices that the error of the intergrid transformation is now minimized. The transformation between the meshes can now act as a preconditioner that will not induce error and approximate with increased accuracy the single patch solution. As result the refinement transformation considered ideal and it will be implemented and tested in three dimensional examples.

3 NUMERICAL EXAMPLES

Since the comparison of the two proposed transformation methods resulted in advantage of the Isogeometric Refinement method, two numerical examples will be provided that will illustrate the performance of the method in real life three dimensional problems. Note that, our proposal was tested on the simple cantilever beams that was described above providing satisfactory results, yet the numerical examples chosen will be of different degrees of freedom magnitudes in order to illustrate the wide range of applications for the proposed method.

3.1 Pipe

The first application of the PCG-IETI method is a pipe and this method is tested on two models with 1k and 10k dof respectively.

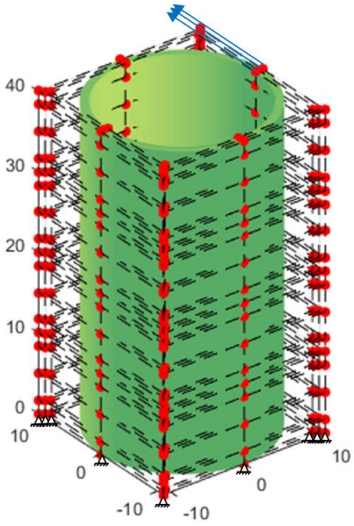
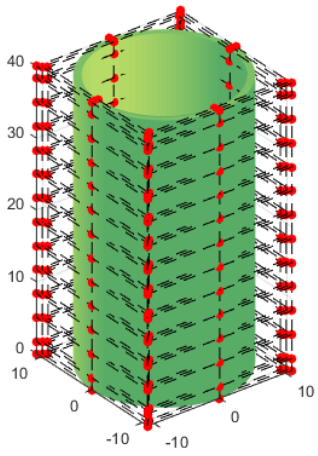
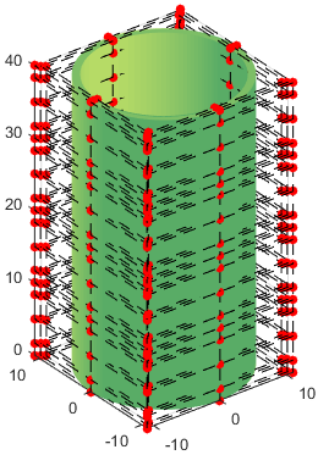


Figure 11: Patched Pipe with 459 Control Points.

In Figure 11 we can see the patched model counterpart of the pipe model. It is created with three dimensional NURBS using NURBS Toolbox [14]. The object is created by 9 Control Points per axis ξ that create the circumference of the circle, 3 per axis η that create the radius of the annulus and 17 per axis η that give height to the pipe. The degree of NURBS Shape functions is considered to be consistent in all parametric directions and equal to 2 for the sake of simplicity. The aforementioned Control Points result to (9x3x17)x3=1377 degrees of freedom. As depicted in Fig.11, all Control Points at the base are clamped, in order to constrain the base displacements. In addition, concentrated loads are applied to Control Points interpolatory to the geometry, introducing both a bending and torsional strain to the structure. The Pipe is segmented into four equal patches of (9x3x5)x3=405 degrees of freedom for the implementation of the IETI part of the implemented method. This patched Pipe is used to approximate a single-patched pipe of similar load and boundary conditions. Both geometries are depicted in Fig.12. The single patched structure is composed now by 9x3x14 Control Points resulting to 1134 degrees of freedom, which are of similar magnitude to the ones of the patched structure.



(a)



(b)

Figure 12: (a) Pipe designed as a single patch. (b) Pipe designed as four separate patches.

In similar fashion, the Pipe structure is analyzed in the case of 10k Dof magnitude. Fig.13 shows the new geometries as created with NURBS Toolbox both patched and unpatched.

(a) (b)

Figure 13: (a) Pipe designed as a single patch. (b) Pipe designed as ten separate patches.

The new structures are considered now to examine a greater order of magnitude. Specifically, the single patch Pipe has now 17 Control Points per axis ξ, 3 Control Points per axis η and 62 Control Points per axis ζ. This results to (17x3x62)x3=9486 degrees of freedom while all other properties, such as degree of Shape Functions, Loads and Boundary Condition are retained. The patched structure is now defined by 17x3x71 Control Points resulting to 10863 degrees of freedom.

	Pipe 1K	Pipe 10K
PCG-IETI iterations	10	6
Total IETI iterations	22	21
Mean IETI iterations	2,2	3,5

Table 5: Performance metrics of the PCG-IETI method for the two pipe models.

In table 5, the performance metrics of the PCG-IETI method are presented, considering an error tolerance of 10^{-5}. The total iterations are depicted on the first row where it can be seen that the 1K model needs more iterations to converge, compared to the 10K one. This is due to the fact that the errors introduced from the patched model are larger for the 1K model, compared to the 10K model. Moreover, a mean of 2.2 and 3.5 IETI iterations where needed for each PCG-IETI iteration. The error for each iteration is depicted on Fig. 14

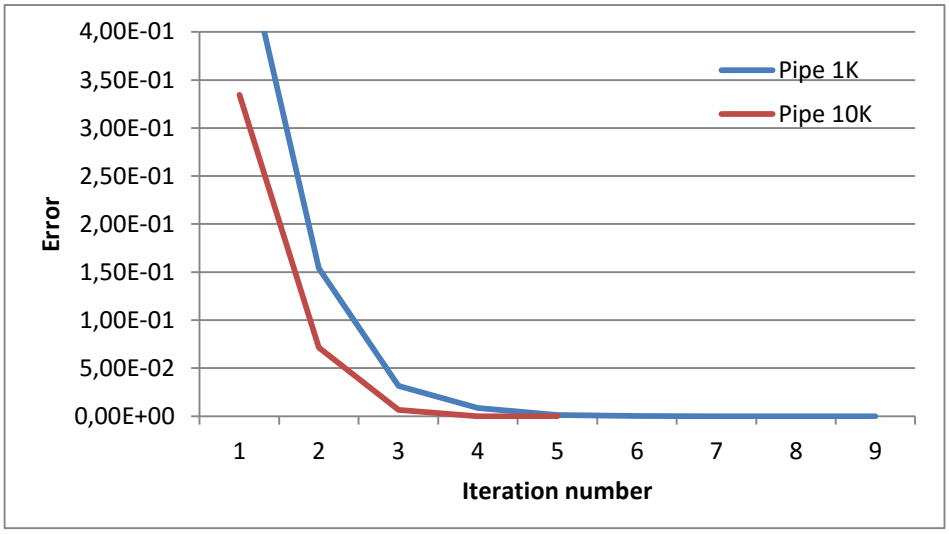


Figure 14: Iterations vs. error for the two pipe models.

3.2 Shell

Second example for the proposed methodology is a Shell once more tested in the same orders of magnitude as the Pipe example.

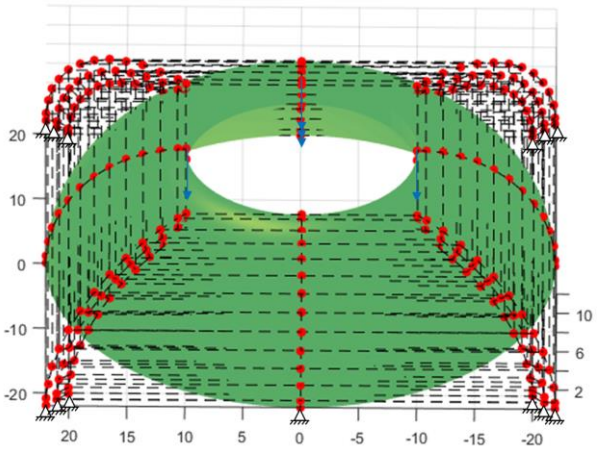
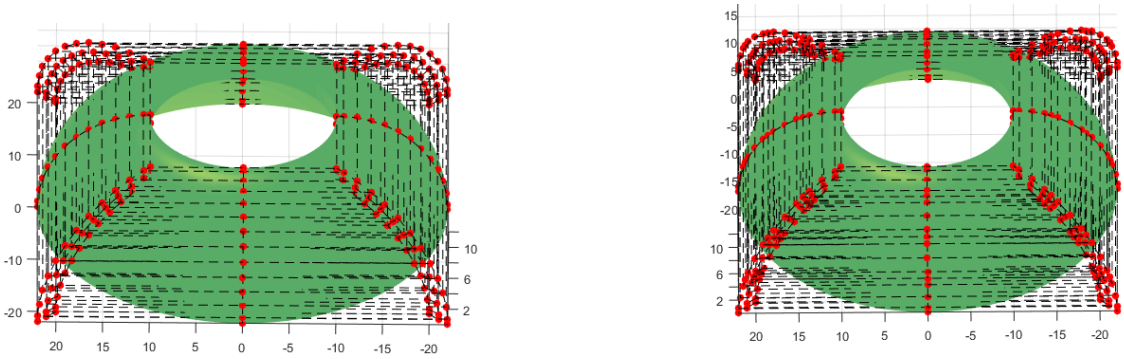


Figure 15: 3D Shell designed with NURBS Toolbox.

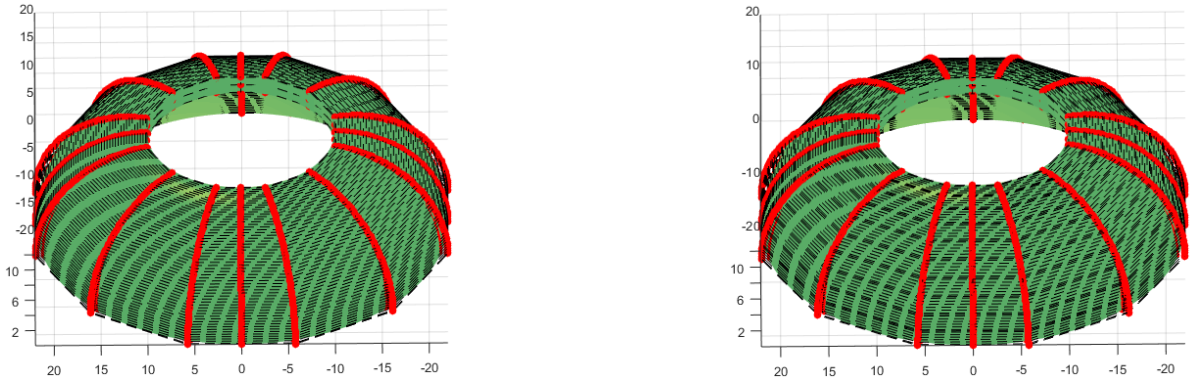
The unpatched shell structure is created by using 9 Control Points per axis ξ, 14 per axis η and 3 per axis ζ, producing 1134 degrees of freedom. As shown in Fig.15, the all the base Control Points are clamped to constrain the base displacements. Loads are once again applied to interpolatory Control Points of the Shell as blue arrows illustrates. Four patches are again used to approximate the exact solution. Both patched and unpatched geometries are shown in Fig.16. Each patch consist of (9x5x3)x3=405 degrees of freedom creating a 1377 Dofs structure.



(a) (b)

Figure 16: (a) Shell designed as a single patch. (b) Shell designed as four separate patches.

Finally, two Shells are analyzed for the 10k Dof case. The degrees of freedom are now, 9486 for the single patch and 10863 for the patched one. Ten patches are used now with 1224 Control Points each for the IETI part of the method. Fig.17 illustrated these two structures and their Control Nets.



(a) (b)

Figure 17: (a) Shell designed as a single patch. (b) Shell designed as ten separate patches.

	Shell 1K	Shell 10K
PCG-IETI iterations	8	5
Total IETI iterations	20	20
Mean IETI iterations	2,2	4

Table 6: Performance metrics of the PCG-IETI method for the two shell models.

In table 6, the performance metrics of the PCG-IETI method are presented, considering an error tolerance of 10^{-5}. The total iterations are depicted on the first row where it can be seen that the 1K model needs more iterations to converge, compared to the 10K one as for the case of the pipes. Once again, this is due to the fact that the errors introduced from the patched model are larger for the 1K model, compared to the 10K model. Moreover, a mean of 2.2 and 4 IETI iterations were needed for each PCG-IETI iteration. The error for each iteration is depicted on Fig. 18

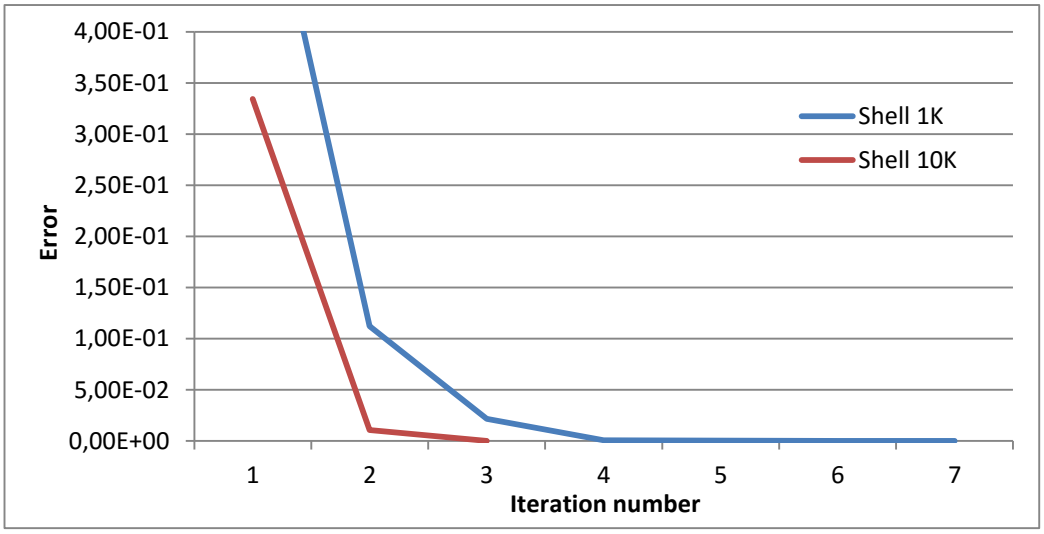


Figure 18: Iterations vs. error for the two pipe models.

REFERENCES

[1] M. Hestenes, E. Stiefel, Methods of Conjugate Gradients for Solving Linear Systems. *Journal of Research of the National Bureau of Standards 49*, 1952.

[2] Y. Saad, Iterative methods for sparse linear systems (2nd Ed.). Philadelphia, *Society for Industrial and Applied Mathematics*, 2003

[3] Axelsson, Owe, Iterative Solution Methods, *Cambridge University Press*, 1996

[4] Y. Saad, H. van der Vorst, Iterative solution of linear systems in the 20th century, §8 Preconditioning methods, pp 193–8 of Numerical Analysis: Historical Developments in the 20th Century, *Elsevier Science Publishers*, 2001.

[5] T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD finite elements NURBS exact geometry and mesh refinement, *Comput. Methods Appl. Mech. Engrg*. 194 (2005) 4135–4195

[6] N. Collier, D. Pardo, L. Dalcin, M. Paszynski, V.M. Calo, The cost of continuity: A study of the performance of isogeometric finite elements using direct solvers, *Comput. Methods Appl. Mech. Engrg*, 2011

[7] N. Collier, L. Dalcin, D. Pardo, V.M. Calo, The cost of continuity: Performance of Iterative Solvers on Isogeometric Finite Elements, *Society for Industrial and Applied Mathematics*, 2013

[8] Stefan K. Kleiss, C. Pechstein, B. Jüttler, S. Tomar, IETI – Isogeometric Tearing and Interconnecting, *Comput. Methods Appl. Mech. Engrg, 2012*

[9] Rhino, CAD Modeling and Design Toolkit. < www.rhino3d.com >

[10] Y. Zhu; A. C. Cangellaris, Multigrid finite element methods for electromagnetic field modeling, *Wiley*, 2006.

[11] W. L. Briggs, Van Emden Henson, S. F. McCormick, A Multigrid Tutorial, *SIAM*, 2000

[12] J. A. Cottrell, T. J. R Hughes, Y. Bazilevs, Isogeometric Analysis: Toward Integration of CAD and FEA, *Wiley*, 2009

[13] L. A. Piegl, W. Tiller, The NURBS Book, *Springer*, 1997

[14] D.M.Spink, NURBS Toolbox

[15] M. Papadrakakis, Solving large-scale linear problems in solid and structural mechanics, in: M. Papadrakakis (Ed.), Solving Large-Scale Problems in Mechanics, John Wiley & Sons, Chichester, 1993, pp. 1–37.