Proceedings of the ASME 2013 International Mechanical Engineering Congress and Exposition
IMECE2013
November 15-21, 2013, San Diego, California, USA

IMECE2013-63228

AUTOMATIC REFINEMENT OF FE SHELL MODELS BASED ON A LOCAL ENERGY FUNCTION

Antonio Carminelli
CIRI-MAM, University of Bologna,
Viale del Risorgimento 2, 40136 Bologna, Italy
phone: +39 051 2093451, fax: +39 051 2093446
e-mail: antonio.carminelli@ unibo.it

Giuseppe Catania
DIN, Dept. of Mechanical Engineering,
University of Bologna, Viale del Risorgimento 2,
40136 Bologna, Italy
phone +39 051 2093447, fax +39 051 2093446
e-mail: giuseppe.catania@unibo.it

ABSTRACT

This paper deals with an adaptive refinement technique of a B-spline degenerate shell finite element model, for the free vibration analysis of curved thin and moderately thick-walled structures. The automatic refinement of the solution is based on an error functional related to the density of the total potential energy. The model refinement is generated by locally increasing, in a sub-domain R of a local patch domain, the number of shape functions while maintaining constant the functions polynomial order. The local refinement strategy is described in a companion paper, written by the same authors of this paper and presented in this Conference. A two-step iterative procedure is proposed. In the first step, one or more sub domains to be refined are identified by means of a point-wise error functional based on the system total potential energy local density. In the second step, the number of shape functions to be added is iteratively increased until the difference of the total potential energy, calculated on the sub domain between two iteration, is below a user defined tolerance. A numerical example is presented in order to test the proposed approach. Strengths and limits of the approach are critically discussed.

INTRODUCTION

Finite Element (FE) techniques are widely used for modeling the vibration behavior of industrial component and structures. Within the FE method, the infinite dimensional solution space is approximated with a finite dimensional one, consequently the obtained solution is usually approximated. In order to reduce the error, the user is usually involved in an iterative,time consuming process, in order to improve the accuracy of FE analysis results and keep low its computational cost. Efficient adaptive procedures can help users improving the accuracy of computed eigensolutions, by means of automatic, optimal, mesh refinement.

Literature on FE adaptive methods is very vast [1-10], demonstrating the interest of the research community on the topic. However, most of it deals with static analysis and less attention has been given to kinetic applications [11,12]. Nevertheless, adaptive methods are only implemented in a few known FE software applications [13-16], such as the ones based on the p-refinement technique [16]. Hughes et al. [17], [18] and the authors of this paper [30] proposed using NURBS functions as the basis for the approximation of a solution field in FE analysis.

An adaptive technique should take into account the following three sub-problems:

- identification of optimal local density error indicator, by identifying the subdomains needing a refinement;
- the definition of a refinement technique. Dofs are added on the subdomains without modifying the geometric shape of the component under study;
- the definition of a global error estimator and of a stopping criterion to determine the level of refinement needed.

The local error indicator value, in standard FE techniques, is usually associated with each element of the FE mesh: if the element error indicator is above a predefined value, then the element will be refined. If macroelements are used, this error indicator is not suitable to provide the needed local error indication. In [19] this local error indicator was associated with a partition of the macroelement, defined by means of the associated knot-vectors. In this paper a point-wise error, based on energy density, is adopted.

In order to make the local refinement of B-spline FE models possible, some limitations inherent in tensorial product B-spline representation must be dealt with. To this purpose, several researchers [19-25] adopt various generalizations of B-spline functions, in which the rectangular form of tensor product B-spline manifolds is not used anymore. T-spline

functions were introduced by Sederberg [26] for geometric modeling applications. In the same paper Sederberg introduced PB-spline functions that are a meshless generalization of both T-spline and B-spline functions. A local refinement method for FE models was proposed by Dorfel *et al.* [19] and by Bazilevs *et.al* [21] making use of T-spline functions. However, these refinement methods can insert more control points (CPs) than the user choice, in order to satisfy some constraints resulting from the T-spline knot-insertion algorithm [19]. In [22] the authors proposed an adaptive local refinement technique by means of the hierarchical B-spline approach. This technique being based on functions subdivision, is not capable of refining the solution on a point inside a knot span without the refinement of all the basis functions active on that knot span. More recently the LR spline over Box-partition paradigm was presented [27]. LR-spline are B-spline meshless functions, as PB-spline are, in which each function is augmented with a scaling coefficient. The refinement algorithm is based on a subdivision technique but scaling coefficients values are calculated such that the resulting subdivided functions remain polynomial. This is in contrast with both the T-spline and PB-spline where a rational scaling is used in order to maintain the partition-of-unity property [26]. Since all the proposed techniques are based on a function subdivision approach, the smallest refinable domain is given by the knot span that defines the support of the functions on which the subdivision operation is applied.

In this paper a B2-spline model is adopted. An automatic refinement procedure based on the gradient of the energy density function is proposed, and the solution can be improved by locally refining on a limited sub-domain. The B2-spline model is described in a companion paper written by the same authors. The approach is based on the superposition of another B-spline (refinement) patch on a small portion of the starting (base) patch. This strategy, being not based on function subdivision, allow the refinement subdomain to be as small as needed.

The choice of the criterion used to stop the iterative procedure is also dealt with. An example is presented in order to test the automatic approach. Critical discussion on advantages and drawbacks of the proposed approach then follows.

B2-SPLINE SHELL MODEL

The B2-spline FE model, adopted for the automatic refinement procedure, is briefly introduced. For further details, [31] can be referred to.

The position vector of the degenerate B2-spline solid shell, with respect to a Cartesian fixed, global reference frame $\{x, y, z\}$, can be expressed as the sum of *np* standard tensorial product of B-spline functions:

$$\mathbf{s}(\xi,\eta,\zeta) = \frac{\sum_{k=0}^{np} \mathbf{s}_k(\xi,\eta,\zeta)}{S} \qquad (1)$$

where

$$\mathbf{s}_k(\xi,\eta,\zeta) = \sum_{i=1}^{m_k}\sum_{j=1}^{n_k} N_i^k(\xi)N_j^k(\eta)\left[\mathbf{P}_{\mathbf{ij}}^k + t_{ij}^k\left(\zeta-\frac{1}{2}\right)\mathbf{v}_{\mathbf{ij}}^{\mathbf{3}k}\right] \qquad (2)$$

and

$$S = \sum_{k=0}^{np}\left(\sum_{i=1}^{m_k}\sum_{j=1}^{n_k} N_i^k(\xi)N_j^k(\eta)\right) \qquad (3)$$

Each $\mathbf{s}_k$ is defined on the domain Ω^k by means of:

- a control net of $m_k \times n_k$ CPs $\mathbf{P}_{\mathbf{ij}}^k$, versors $\mathbf{v}_{\mathbf{ij}}^{\mathbf{3}k}$ and thickness coefficients t_{ij}^k;
- the uni-variate normalized B-spline functions $N_i^k(\xi)$ of degree *p*, defined with respect to the curvilinear coordinates ξ by means of the knot vector $\mathbf{U}^k$;
- the uni-variate normalized B-spline functions $N_j^k(\eta)$ of degree *q*, defined with respect to the curvilinear coordinates η by means of the knot vector $\mathbf{V}^k$.

The versors $\mathbf{v}_{\mathbf{ij}}^{\mathbf{3}k}$ and the thickness values t_{ij}^k can be calculated from the interpolation algorithm proposed in [30] by the authors.

For *k=0* the knot-vectors are partition of the parametric interval [0,1] as follows:

$$\mathbf{U}^0 = \left\{\xi_1^0,\ldots,\xi_r^0\right\} = \left\{\underbrace{0,\ldots,0}_{p+1},\xi_{p+1}^0,\ldots,\xi_{r-p-1}^0,\underbrace{1,\ldots,1}_{p+1}\right\}$$

and

$$\mathbf{V}^0 = \left\{\eta_1^0,\ldots,\eta_s^0\right\} = \left\{\underbrace{0,\ldots,0}_{q+1},\eta_{q+1}^0,\ldots,\eta_{s-q-1}^0,\underbrace{1,\ldots,1}_{q+1}\right\}.$$

For *k>0* the knot-vectors $\mathbf{U}^k$ and $\mathbf{V}^k$ are partition of the knot-span of $\mathbf{U}^0$ and $\mathbf{V}^0$.

The displacement field can be defined by following the isoparametric approach and enforcing the fiber inextensibility in the thickness direction [29]:

$$\mathbf{d}(\xi,\eta,\zeta) = \begin{Bmatrix} d_x \\ d_y \\ d_z \end{Bmatrix} = \frac{\sum_{k=0}^{np}\mathbf{d}^k(\xi,\eta,\zeta)}{S} \qquad (4)$$

where:

$$\mathbf{d}^k(\xi,\eta,\zeta) = \mathbf{N^k}\cdot\boldsymbol{\delta}^{\mathbf{k}} = \mathbf{N^k}\cdot\begin{Bmatrix}\boldsymbol{\delta}_i^{\mathbf{k}}\\ \boldsymbol{\delta}_b^{\mathbf{k}}\end{Bmatrix}$$
$$= \sum_{i=1}^{m_k}\sum_{j=1}^{n_k} N_i^k(\xi)N_j^k(\eta)\left(\begin{Bmatrix}u_{ij}^k\\ v_{ij}^k\\ w_{ij}^k\end{Bmatrix} + t_{ij}^k\left(\zeta-\frac{1}{2}\right)[\mathbf{-v}_{ij}^{\mathbf{2}k} \quad \mathbf{v}_{ij}^{\mathbf{1}k}]\begin{Bmatrix}\alpha_{ij}^k\\ \beta_{ij}^k\end{Bmatrix}\right) \quad (5)$$

and $\left(\mathbf{v}_{ij}^{\mathbf{1}k}, \mathbf{v}_{ij}^{\mathbf{2}k}, \mathbf{v}_{ij}^{\mathbf{3}k}\right)$ is an orthonormal set defined on the $\mathbf{P}_{ij}^k$ CPs starting from the vector $\mathbf{v}_{ij}^{\mathbf{3}k}$ [29], $u_{ij}^k, v_{ij}^k, w_{ij}^k$ are the three translational dofs, α_{ij}^k and β_{ij}^k are two rotational dofs for each CP. $\boldsymbol{\delta}_b^{\mathbf{k}}$ are the dofs on the boundary of the domain on which $\mathbf{d}^k$ is defined, $\boldsymbol{\delta}_i^{\mathbf{k}}$ are the dofs on the interior of the domain.

In order to assure C^0 continuity of the displacement field Eq.4, some of the dofs in $\mathbf{d}^k$ (with $k>0$) are expressed as linear function of dofs in $\mathbf{d}^0$ by means of the transformation explained in [31], so that the displacement field can be expressed in matrix form as:

$$\begin{Bmatrix}d_x\\ d_y\\ d_z\end{Bmatrix} = \mathbf{N}\cdot\begin{Bmatrix}\boldsymbol{\delta}^0\\ \boldsymbol{\delta}_{\mathbf{i}}^1\\ \vdots\\ \boldsymbol{\delta}_{\mathbf{i}}^K\end{Bmatrix} = \mathbf{N}\cdot\boldsymbol{\delta} \quad (6)$$

The equation of motion for harmonic forced vibration analysis can be obtained by means of the principle of minimum total potential energy as shown in [15]:

$$\mathbf{M}\cdot\ddot{\boldsymbol{\delta}} + \mathbf{K}\cdot\boldsymbol{\delta} = \mathbf{F}_0 e^{j\omega t} \quad (7)$$

The expressions of the elasticity, inertia matrices and surface tractions are:

$$\mathbf{K} = \int_{\Omega^{\mathbf{P}}} \mathbf{B}^{\mathbf{T}}\mathbf{E}\mathbf{B}\, d\Omega \quad (8)$$

$$\mathbf{M} = \int_{\Omega^{\mathbf{P}}} \rho\mathbf{N}^{\mathbf{T}}\mathbf{N}\, d\Omega \quad (9)$$

$$\mathbf{F} = \int_{\Sigma^{\mathbf{P}}} \mathbf{N}^{\mathbf{T}}\boldsymbol{\Phi}\, d\Sigma \quad (10)$$

where $\mathbf{B}$ is the strain-displacement matrix, $\mathbf{E}$ is the plane stress constitutive matrix obtained according to the Mindlin theory [29], ρ is the mass density, $\mathbf{N}$ is the matrix of basis functions, $\boldsymbol{\Phi}$ the surface tractions, Ω^P being the solid geometry structure under analysis and Σ^P the surface where loads are prescribed.

As already reported on the companion paper [31], the numerical integration of vector and matrices in Eq.7 is a major problem when dealing with the B2-spline shell model. Therefore the correct evaluation of the matrices (Eq.8,9) are needed at each iteration step. In this example, all the integrals on Ω^0 are numerically evaluated by means of *(p+1)x(q+1)* Gauss rule in (ξ,η) for each knot-span given by the knot-vectors $\mathbf{U}$ and $\mathbf{V}$. Lower order Gauss rules were adopted in evaluating the integrals on Ω^k (with $k>1$) on subdomains defined partitioning the knot span in $\mathbf{U}^k$ and $\mathbf{V}^k$. Two-points Gauss rule is adopted along the thickness coordinate ζ.

ADAPTIVE REFINEMENT STRATEGY

The automatic refinement strategy is based on the gradient of the density of total potential energy. The potential energy, of internal and inertia forces, of a system modeled by means of the FE method is:

$$\Pi = \boldsymbol{\delta}^T\cdot\mathbf{M}\cdot\ddot{\boldsymbol{\delta}} + \frac{1}{2}\boldsymbol{\delta}^T\cdot\mathbf{K}\cdot\boldsymbol{\delta}$$
$$= \boldsymbol{\delta}^T\cdot\int_{\Omega^{\mathbf{P}}}\rho\mathbf{N}^{\mathbf{T}}\mathbf{N}\, d\Omega\cdot\ddot{\boldsymbol{\delta}} + \frac{1}{2}\boldsymbol{\delta}^T\cdot\int_{\Omega^{\mathbf{P}}}\mathbf{B}^{\mathbf{T}}\mathbf{E}\mathbf{B}\, d\Omega\cdot\boldsymbol{\delta} \quad (11)$$

By integrating only along the thickness, by adopting a Gauss rule with two points:

$$\Pi = \boldsymbol{\delta}^T\cdot\int_0^1\int_0^1(\tilde{\mathbf{m}}(\xi,\eta)\cdot\ddot{\boldsymbol{\delta}} + \frac{1}{2}\cdot\tilde{\mathbf{k}}(\xi,\eta)\cdot\boldsymbol{\delta})\cdot d\xi\cdot d\eta \quad (12)$$

where

$$\tilde{\mathbf{k}}(\xi,\eta) = \sum_{i=1}^{2} w_i\,\mathbf{B}^{\mathbf{T}}(\xi,\eta,\zeta_i)\mathbf{E}\,\mathbf{B}(\xi,\eta,\zeta_i)\det J(\xi,\eta,\zeta_i) \quad (13)$$

and

$$\tilde{\mathbf{m}}(\xi,\eta) = \sum_{i=1}^{2} w_i\rho\mathbf{N}(\xi,\eta,\zeta_i)^{\mathbf{T}}\,\mathbf{N}(\xi,\eta,\zeta_i)\det J(\xi,\eta,\zeta_i) \quad (14)$$

Introducing the solution:

$$\boldsymbol{\delta} = \boldsymbol{\Phi}\cdot\mathbf{q}(t) = \sum_{r=1}^{n}\boldsymbol{\varphi}_r\cdot Q_r e^{j\omega t} = \sum_{r=1}^{n}\boldsymbol{\varphi}_r\cdot q_r$$
$$\ddot{\boldsymbol{\delta}} = -\omega^2\cdot\boldsymbol{\delta} \quad (15)$$

in Eq.(12), the following equation holds:

$$\Pi = \int_{\Omega} d\Pi = \int_0^1\int_0^1 g(\xi,\eta)d\xi d\eta =$$
$$= \int_0^1\int_0^1\left[\sum_{r=1}^{n} q_r^2\,\boldsymbol{\varphi}_r^T\left(\frac{1}{2}\tilde{\mathbf{k}}(\xi,\eta) - \omega^2\tilde{\mathbf{m}}(\xi,\eta)\right)\boldsymbol{\varphi}_r\right]d\xi\, d\eta \quad (16)$$

As a consequence, the surface density of the total potential energy is:

$$g(\xi,\eta)=\sum_{r=1}^{n}\gamma_r\cdot\psi_r(\xi,\eta) \tag{17a}$$

with $\gamma_r=q_r^2$ and:

$$\psi_r(\xi,\eta)=\boldsymbol{\varphi}_r^T\left(\frac{1}{2}\tilde{\mathbf{k}}(\xi,\eta)-\omega^2\tilde{\mathbf{m}}(\xi,\eta)\right)\boldsymbol{\varphi}_r\ . \tag{17b}$$

The local indicator $LI(\xi,\eta)$ is defined:

$$\begin{aligned}LI(\xi,\eta)&=\left(\frac{\partial g}{\partial\xi}\right)^2+\left(\frac{\partial g}{\partial\eta}\right)^2=\\&=\left(\sum_r\gamma_r\frac{\partial\psi_r}{\partial\xi}\right)^2+\left(\sum_r\gamma_r\frac{\partial\psi_r}{\partial\eta}\right)^2\end{aligned} \tag{18}$$

where only a few modes, $r\in[r_{\min},r_{\max}]$, are generally taken into account.

A grid of LI values can be evaluated with a predefined uniform resolution on the patch to be refined: $n^{\xi}\cdot n^{\eta}$ evaluation points on the (ξ,η) parametric space result. Each calculated *LI* value is compared with the following value:

$$GI=\frac{\alpha}{n^{\xi}n^{\eta}}\sum_{i=1}^{n^{\xi}}\sum_{j=1}^{n^{\eta}}LI(\xi_i,\eta_j) \tag{19}$$

Each value $LI>GI$ identifies a point where a refinement could be needed. Contiguous values are grouped so that they identifies a subdomain where the refinement procedure will be applied. If a point is marked as to be refined, while the contiguous points LI value are below the *GI* value, it is disregarded. This allow geometric singularity to be dealt with; as a matter of fact, in presence of a geometric singularity, while refining the model, the energy usually grows only on the singularity point.

The refinement algorithm proposed in [31] is then applied on the identified subdomains. The number of added dofs are iteratively increased until, for more than one point, on the refining subdomain at the *k-th* step, the following condition holds:

$$\frac{\left|GI^{k-1}-GI^{k}\right|}{GI^{k-1}}<\beta \tag{20}$$

where GI^k is the GI value calculated at the *k-th* step of the iterative procedure, and β is a user defined value used to stop the iteration if the percent change of the GI indicator, between two successive steps, is below the β value.

NUMERICAL TEST

An example case is considered for testing the approach. The example concerns a curved roof with a circular section and a small cutout (Fig.1). The roof is fully constrained on one edge (*y*=1) and loaded with a constant pressure on a subdomain delimited by the red lines in Fig.1. A stress singularity is expected near the corner C (green circle in Fig.2).

The B-spline shell model is made by means of four B-spline patches, connected as shown in Fig.2. Each patch is defined by means of 7x7 CPs and sixth degree B-spline functions defined on the knot-vectors $\mathbf{U}=\mathbf{V}=\left\{\underbrace{0,\ldots,0}_{7},\underbrace{1,\ldots,1}_{7}\right\}$.

The CPs (black dot in Figs.1-2) defining the shell position vector are included in the appendix. The geometry position vector and the displacement field are C^0 on the boundary connecting the patches. Moreover, the following parameters are used:

Young's modulus $E=2.3\cdot10^9\ N/m^2$;
Poisson's ratio ν=0.35;
Density ρ= 1000 Kg/m^3;
Thickness $t=0.005\,m$.

The automatic refinement strategy is adopted in a forced vibration problem. A uniform pressure *p*=1000 N/m^2, directed along the negative *z* axis, is applied inside the domain $\boldsymbol{\Omega}^{\mathbf{F}}$ delimited by the red lines (Fig.2), corresponding to half the domain of the upper-left patch. The forcing function $\boldsymbol{\Phi}$ is:

$$\boldsymbol{\Phi}=\begin{Bmatrix}0\\0\\-p\end{Bmatrix}e^{j\omega t}$$

with ω=16Hz. With this forcing frequency, given the starting eigensolutions in Tab.1, an approximated solution can be obtained by considering only the fourth mode in Eqs.18-20, so that γ_4=6.8121e-4 results. The plot of the fourth mode shape is

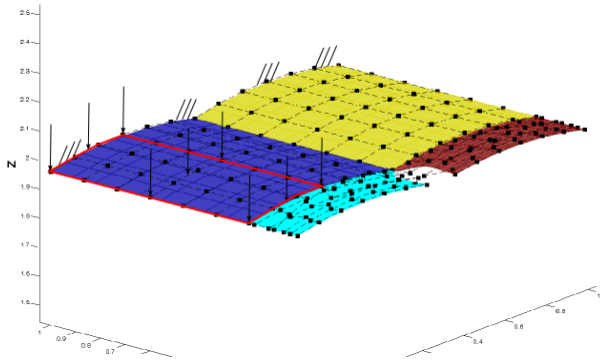

Figure 1. Cantilevered shell roof with a distributed pressure.

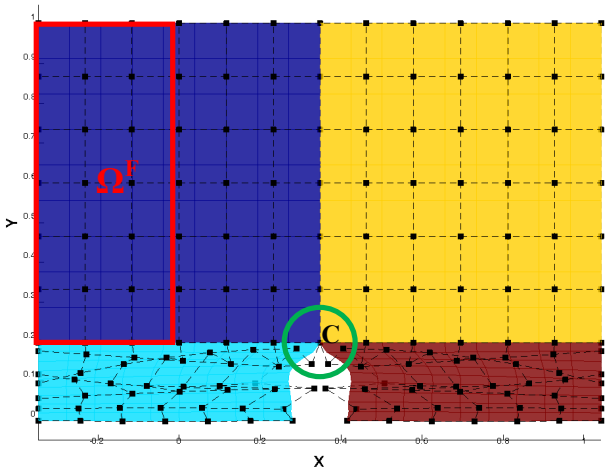

Figure 2. Top view of the shell roof: the forcing pressure is applied on Ω^F (bounded by red lines) and a singularity is expected near the point C (green circle).

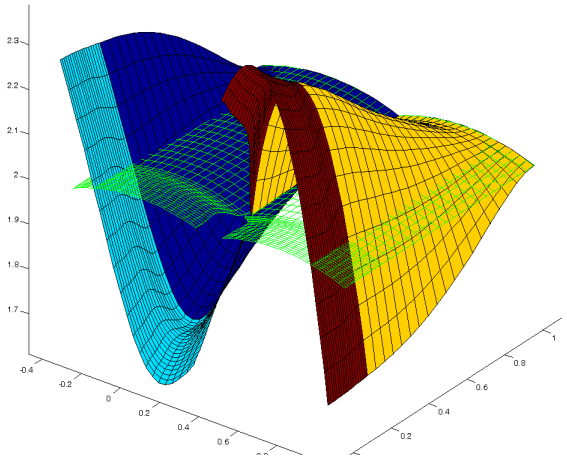

Figure 3. shell roof: fourth modeshape. In green the undeformed model.

reported in Fig.3. Figure 4 shows the energy density of the 4th mode shape and its gradient is reported in Fig.5. The model damping is not considered in the example. Real modal damping can be easily included.

The adaptive refinement procedure is started using α=20, $n^{\xi}=n^{\eta}$=40 and β=0.05 (Eqs.19-20). With these values the subdomains to be refined near the point *C* are identified (Fig.6) so that the iterative part of the procedure, in which the dofs are added at each step, can be started.

The procedure added the dofs on four subdomains (Fig.7) near the point C whose parametric coordinates are:

- Patch 1 [0.8,1]x[0.8,1]
- Patch 2 [0,0.3]x[0.575,1]
- Patch 3 [0.8,1]x[0,0.2]
- Patch 4 [0,0.3]x[0,0.425]

The eigensolutions after the refinement are reported in Tab.2: the fourth mode shows a good improvement.

Mode index	B-spline unrefined (875 dofs)	B2-spline refined (3315 dofs)
1	6.7310	6.4986
2	6.7581	6.7339
3	9.8154	9.8531
4	15.603	14.451
5	20.140	19.691
6	20.193	20.130
7	33.487	30.418
8	35.185	32.593
9	35.777	35.603
10	39.979	39.904

Tab2. Numerical frequencies results (Hz) for the shell roof model.

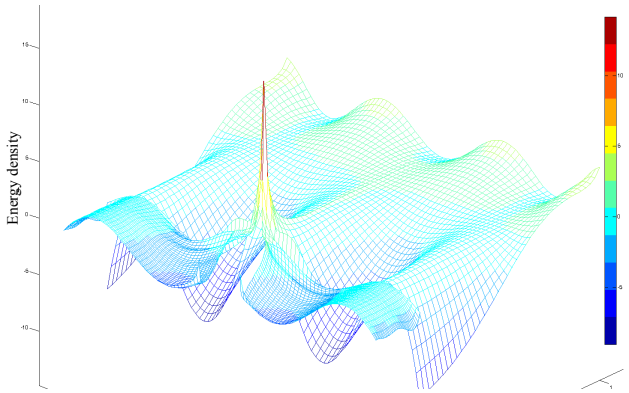

Figure 4. Plot of the energy density: a peak is shown near the point C.

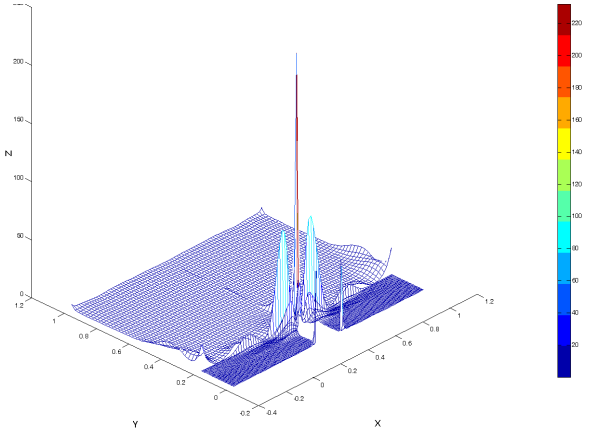

Figure 5. Plot of the local indicator *LI*.

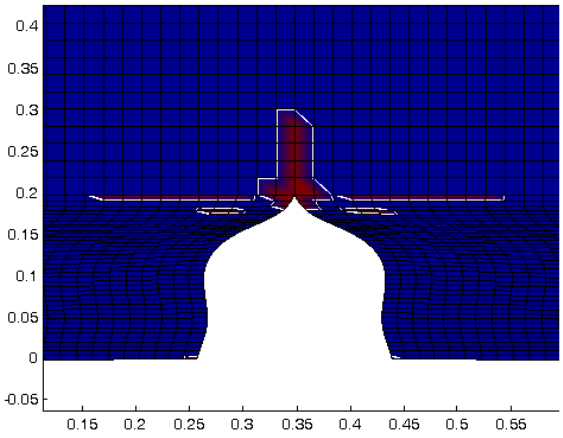

Figure 6. In red the subdomain to be refined.

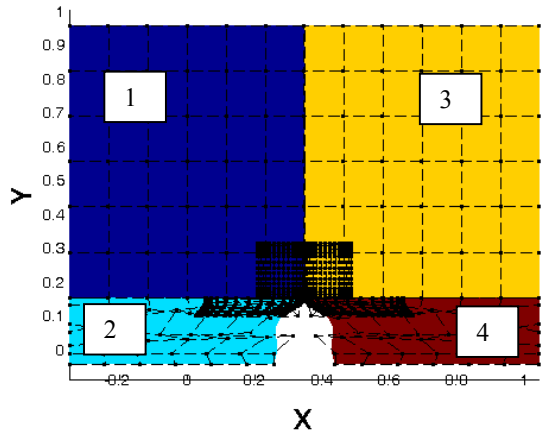

Figure 7. Black dots represents the CPs: the dofs added on the subdomains are clearly visible.

CONCLUSION

In the present study an automatic procedure for refining the eigensolution obtained by means of B-spline shell FE is proposed. The refinement algorithm adopted is shown in a companion paper written by the same authors. The indicator used for locating the subdomains to be refined is based on the gradient of the total energy density function. With respect to other known techniques, an advantage of the present approach is the ability to refine a very small zone, within a knot-span, with a minimum number of added dofs. The refinement approach does not need that dofs are inserted outside the subdomain to be refined. The automatc choice of the starting solution accuracy was not taken into account in this paper. If a small number of dofs are used in the unrefined model, high gradients can not be modeled with sufficient accuracy and the LI functional is not expected to perform correctly. Optimal numerical integration of model parameters is critical with B2 FE modeling, and requires further study. In this paper low order Gauss rules were adopted. The correct choice of the optimal integration order in the reported example was obtained by a trial and error iterative procedure.

A proposed numerical example showed the validity of the approach for the selected tested model, but more case studies are needed in order to test the approach. Future studies will address towards investigating the outlined drawbacks and towards the optimal choice of the α and β values.

APPENDIX

The CPs coordinates of the shell roof used in the numerical example are:

CP	x	y	z
1	-0.347296355300000	1	1.96961550600000
2	-0.232714925000000	1	1.98986257560000
3	-0.116697472200000	1	2.00196063190000
4	-1.564540000000e-05	1	2.00608974120000
5	0.116724799200000	1	2.00211999290000
6	0.232701595500000	1	1.98976406760000
7	0.347296355300000	1	1.96961550600000
8	-0.347296355300000	0.866089372400000	1.96961550600000
9	-0.232714925000000	0.866089370600000	1.98986257560000
10	-0.116697472200000	0.866089368700000	2.00196063190000
11	-1.564540000000e-05	0.866089366800000	2.00608974120000
12	0.116724799200000	0.866089365000000	2.00211999290000
13	0.232701595500000	0.866089363100000	1.98976406760000
14	0.347296355300000	0.866089361300000	1.96961550600000
15	-0.347296355300000	0.732178528600000	1.96961550600000
16	-0.232714925000000	0.732178527000000	1.98986257560000
17	-0.116697472200000	0.732178525200000	2.00196063190000
18	-1.564540000000e-05	0.732178523500000	2.00608974120000
19	0.116724799200000	0.732178521800000	2.00211999290000
20	0.232701595500000	0.732178520100000	1.98976406760000
21	0.347296355300000	0.732178518400000	1.96961550600000
22	-0.347296355300000	0.598268103200000	1.96961550600000
23	-0.232714925000000	0.598268102700000	1.98986257560000
24	-0.116697472200000	0.598268102200000	2.00196063190000
25	-1.564540000000e-05	0.598268101700000	2.00608974120000
26	0.116724799200000	0.598268101200000	2.00211999290000
27	0.232701595500000	0.598268100700000	1.98976406760000
28	0.347296355300000	0.598268100200000	1.96961550600000
29	-0.347296355300000	0.464357165500000	1.96961550600000
30	-0.232714925000000	0.464357166300000	1.98986257560000
31	-0.116697472200000	0.464357167100000	2.00196063190000
32	-1.564540000000e-05	0.464357167900000	2.00608974120000
33	0.116724799200000	0.464357168800000	2.00211999290000
34	0.232701595500000	0.464357169600000	1.98976406760000
35	0.347296355300000	0.464357170400000	1.96961550600000
36	-0.347296355300000	0.330446630900000	1.96961550600000
37	-0.232714925000000	0.330446632200000	1.98986257560000
38	-0.116697472200000	0.330446633500000	2.00196063190000
39	-1.564540000000e-05	0.330446634800000	2.00608974120000
40	0.116724799200000	0.330446636200000	2.00211999290000
41	0.232701595500000	0.330446637500000	1.98976406760000
42	0.347296355300000	0.330446638800000	1.96961550600000

43	-0.347296355300000	0.196535904600000	1.96961550600000
44	-0.232714925000000	0.196535904600000	1.98986257560000
45	-0.116697472200000	0.196535904600000	2.00196063190000
46	-1.564540000000e-05	0.196535904600000	2.00608974120000
47	0.116724799200000	0.196535904600000	2.00211999290000
48	0.232701595500000	0.196535904600000	1.98976406760000
49	0.347296355300000	0.196535904600000	1.96961550600000
50	-0.347296355300000	0.175296002600000	1.96961550600000
51	-0.229706720400000	0.167009917800000	1.99001701520000
52	-0.110753142500000	0.158854561900000	2.00184394340000
53	0.00867211920000000	0.150876638800000	2.00556958520000
54	0.128107739800000	0.143110117500000	2.00105714340000
55	0.246534014500000	0.135641893400000	1.98754547170000
56	0.327729621700000	0.142258308300000	1.96625336500000
57	-0.347296355300000	0.0941547212000000	1.96961550600000
58	-0.243275073400000	0.120353796600000	1.98933012660000
59	-0.137497952200000	0.141902478700000	2.00188133050000
60	-0.0307934504000000	0.158955132600000	2.00764646280000
61	0.0764338766000000	0.171377061000000	2.00650056260000
62	0.183318404900000	0.178612578100000	1.9977825360000
63	0.289343363800000	0.181285318900000	1.98250399910000
64	-0.347296355300000	0.116703429500000	1.96961550600000
65	-0.258996263700000	0.102906911300000	1.98986201500000
66	-0.169539663800000	0.0931748476000000	2.00326442210000
67	-0.0796059994000000	0.0873107869000000	2.01017371430000
68	0.0104591650000000	0.0853960681000000	2.01046486100000
69	0.0998988874000000	0.0878304358000000	2.00349291930000
70	0.188323931100000	0.0940463557000000	1.99022092800000
71	-0.347296355300000	0.0561123765000000	1.96961550600000
72	-0.232378578700000	0.0625036994000000	1.98964676940000
73	-0.115842741500000	0.0681127450000000	2.00164570500000
74	0.00141884710000000	0.0729520649000000	2.00604945670000
75	0.118957181000000	0.0769910682000000	2.00272620030000
76	0.235794629200000	0.0801175465000000	1.99095423920000
77	0.313098505800000	0.0813419595000000	1.97187302070000
78	-0.347296355300000	0.0299820101000000	1.96961550600000
79	-0.247891531600000	0.0311887017000000	1.98973300080000
80	-0.147071868600000	0.0318635332000000	2.00255703010000
81	-0.0456103051000000	0.0320362592000000	2.00847148560000
82	0.0561048632000000	0.0316986293000000	2.00734895810000
83	0.157228985300000	0.0308039864000000	1.99851592710000
84	0.257307164300000	0.0294280743000000	1.98300226350000
85	-0.347296355300000	0	1.96961550600000
86	-0.2442025258000	-0.0007764283000000	1.98968461660000
87	-0.139650738200000	-0.001235430500000	2.00229612720000
88	-0.0344419463000000	-0.001394384500000	2.00784595690000
89	0.0710210833000000	-0.001248141500000	2.00620583240000
90	0.175861731600000	-0.000768324400000	1.9966918072000
91	0.257307164300000	0	1.9803581661000
92	0.461891115200000	1	1.9897640676000
93	0.577867911400000	1	2.0021199929000
94	0.694608356000000	1	2.0060897412000
95	0.811290182900000	1	2.0019606319000
96	0.927307635700000	1	1.9898625756000
97	1.04188906600000	1	1.9696155060000
98	0.461891115200000	0.866089363100000	1.9897640676000
99	0.577867911400000	0.866089365000000	2.0021199929000
100	0.694608356000000	0.866089366800000	2.0060897412000
101	0.811290182900000	0.866089368700000	2.00196063190000
102	0.927307635700000	0.866089370600000	1.98986257560000
103	1.04188906600000	0.866089372400000	1.96961550600000
104	0.461891115200000	0.732178520100000	1.98976406760000
105	0.577867911400000	0.732178521800000	2.00211999290000
106	0.694608356000000	0.732178523500000	2.00608974120000
107	0.811290182900000	0.732178525200000	2.00196063190000
108	0.927307635700000	0.732178527000000	1.98986257560000
109	1.04188906600000	0.732178528600000	1.96961550600000
110	0.461891115200000	0.598268100700000	1.98976406760000
111	0.577867911400000	0.598268101200000	2.00211999290000
112	0.694608356000000	0.598268101700000	2.00608974120000
113	0.811290182900000	0.598268102200000	2.00196063190000
114	0.927307635700000	0.598268102700000	1.98986257560000
115	1.04188906600000	0.598268103200000	1.96961550600000
116	0.461891115200000	0.464357169600000	1.98976406760000
117	0.577867911400000	0.464357168800000	2.00211999290000
118	0.694608356000000	0.464357167900000	2.00608974120000
119	0.811290182900000	0.464357167100000	2.00196063190000
120	0.927307635700000	0.464357166300000	1.98986257560000
121	1.04188906600000	0.464357165500000	1.96961550600000
122	0.461891115200000	0.330446637500000	1.98976406760000
123	0.577867911400000	0.330446636200000	2.00211999290000
124	0.694608356000000	0.330446634800000	2.00608974120000
125	0.811290182900000	0.330446633500000	2.00196063190000
126	0.927307635700000	0.330446632200000	1.98986257560000
127	1.04188906600000	0.330446630900000	1.96961550600000
128	0.461891115200000	0.196535904600000	1.98976406760000
129	0.577867911400000	0.196535904600000	2.00211999290000
130	0.694608356000000	0.196535904600000	2.00608974120000
131	0.811290182900000	0.196535904600000	2.00196063190000
132	0.927307635700000	0.196535904600000	1.98986257560000
133	1.04188906600000	0.196535904600000	1.96961550600000
134	0.366863089000000	0.142258308300000	1.96625336500000
135	0.448058696200000	0.135641893400000	1.98754547170000
136	0.566484970800000	0.143110117500000	2.00105714340000
137	0.685920591500000	0.150876638800000	2.00556958520000
138	0.805345853100000	0.158854561900000	2.00184394340000
139	0.924299431100000	0.167009917800000	1.99001701520000
140	1.04188906600000	0.175296002600000	1.96961550600000
141	0.405249346900000	0.181285318900000	1.98250399910000
142	0.511274305800000	0.178612578100000	1.99778253600000
143	0.618158834000000	0.171377061000000	2.00650056260000
144	0.725386161100000	0.158955132600000	2.00764646280000
145	0.832090662900000	0.141902478700000	2.00188133050000
146	0.937867784100000	0.120353796600000	1.98933012660000
147	1.04188906600000	0.0941547212000000	1.96961550600000
148	0.506268779500000	0.0940463557000000	1.99022092800000
149	0.594693823200000	0.0878304358000000	2.00349291930000
150	0.684133545700000	0.0853960681000000	2.01046486100000
151	0.774198710100000	0.0873107869000000	2.01017371430000
152	0.864132374500000	0.0931748476000000	2.00326442210000
153	0.953588974400000	0.102906911300000	1.98986201500000
154	1.04188906600000	0.116703429500000	1.96961550600000
155	0.381494204800000	0.0813419595000000	1.97187302070000
156	0.458798081500000	0.0801175465000000	1.99095423920000
157	0.575635529700000	0.0769910682000000	2.00272620030000
158	0.693173863600000	0.0729520649000000	2.00604945670000
159	0.810435452100000	0.0681127450000000	2.00164570500000
160	0.926971289300000	0.0625036994000000	1.98964676940000
161	1.04188906600000	0.0561123765000000	1.96961550600000
162	0.437285546400000	0.0294280743000000	1.98300226350000
163	0.537363725400000	0.0308039864000000	1.99851592710000
164	0.638487847400000	0.0316986293000000	2.00734895810000
165	0.740203015700000	0.0320362592000000	2.00847148560000
166	0.841664579300000	0.0318635332000000	2.00255703010000
167	0.942484242300000	0.0311887017000000	1.98973300080000
168	1.04188906600000	0.0299820101000000	1.96961550600000
169	0.437285546400000	0	1.98035816610000
170	0.518730979000000	-0.000768324400000	1.99669180720000
171	0.623571627300000	-0.001248141500000	2.00620583240000
172	0.729034657000000	-0.001394384500000	2.00784595690000
173	0.834243448800000	-0.001235430500000	2.00229612720000
174	0.938795236500000	-0.000776428300000	1.98968461660000

175	1.04188906600000	0	1.96961550600000

The CPs number of each patch are:
Patch1:

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49

Patch2:

43	44	45	46	47	48	49
50	51	52	53	54	55	56
57	58	59	60	61	62	63
64	65	66	67	68	69	70
71	72	73	74	75	76	77
78	79	80	81	82	83	84
85	86	87	88	89	90	91

Patch3:

7	92	93	94	95	96	97
14	98	99	100	101	102	103
21	104	105	106	107	108	109
28	110	111	112	113	114	115
35	116	117	118	119	120	121
42	122	123	124	125	126	127
49	128	129	130	131	132	133

Patch4:

49	128	129	130	131	132	133
134	135	136	137	138	139	140
141	142	143	144	145	146	147
148	149	150	151	152	153	154
155	156	157	158	159	160	161
162	163	164	165	166	167	168
169	170	171	172	173	174	175

ACKNOWLEDGMENTS

The present study was supported by the Regione Emilia-Romagna, Progetto Tecnopolo POR FESR 2007-2013, Asse 1 Attività I.1.1.

REFERENCES

[1] Deuflhard, P., Weiser, M., 2012, Adaptive Numerical Solution of PDEs, Berlin, Boston: De Gruyter.

[2] Szabo, Â., BabusÏka, I., 1991, Finite Element Analysis, John Wiley & Sons, New York.

[3] Zienkiewicz, O.C., Taylor, R.L., 1989, The Finite Element Method, Vol. 1: Basic Formulation and Linear Problems. McGraw-Hill, London.

[4] Vejchodsky, T., 2006, "The Problem of Adaptivity for hp-FEM″, in *Proceedings of the Conference Presentation of Mathematics* 2006, Liberec, 247-254.

[5] Ainsworth, M., Oden, J. T., 2011, "A posteriori error estimation in finite element analysis″, Wiley-Interscience.

[6] Bank, R. E., Smith, R. K., 1993, "A posteriori error estimates based on hierarchical bases". *SIAM Journal on Numerical Analysis*, *30*(4), pp. 921-935.

[7] Babuvška, I., Rheinboldt, W. C., 1978, "Error estimates for adaptive finite element computations", *SIAM Journal on Numerical Analysis*, *15*(4), pp. 736-754.

[8] Bank, R. E., Xu, J., 2004, "Asymptotically exact A posteriori error estimators, part II: General unstructured grids", *SIAM Journal on Numerical Analysis*, 41(6), pp. 2313-2332.

[9] Carstensen, C., 2004, "Some remarks on the history and future of averaging techniques in a posteriori finite element error analysis", ZAMM, 84(4), pp. 3-21.

[10] Paulino, G. H., Menezes, I. F. M., Neto, J. C., Martha, L. F., 1999, "A methodology for adaptive finite element analysis: towards an integrated computational environment", *Computational Mechanics*, *23*(5-6), pp. 361-388.

[11]Neumann, J., Schweizerhof, K., 2007, "Computation of Single Eigenfrequencies and Eigenfunctions of Plate and Shell Structures Using an H-adaptive FE-method", *Computational Mechanics*, 40(1), pp- 111-126.

[12]Mukherjee, S., Jafarali, P., Prathap, G., 2005, "A variational basis for error analysis in finite element elastodynamic problems", *Journal of sound and vibration*, *285*(3), pp. 615-635.

[13] Bastian, P., Wittum, G., 1993, "Adaptive multigrid methods: The UG concept", in Adaptive methods - algorithms, theory and applications, Proceedings of the 9th GAMM Seminar, Notes Numer. Fluid Mech. (46), pp. 7-37, Vieweg., Braunschweig.

[14] Schmidt, A., Siebert, K.G., 2005, "Design of Adaptive Finite Element Software – The Finite Element Toolbox ALBERTA", Lecture Notes in *Computational Science and Engeneering* (42), Springer.

[15] Tristano, J. R., Chen, Z., HANCQ, A., Kwok, W., 2002, "Fully automatic adaptive mesh refinement integrated into the solution process", In *International Meshing Roundtable,* Vol. 12, pp. 307-314.

[16] Toogood, R., Eng, P., 2001, "Pro-mechanica tutorial structure". Schroff Development Corporation

[17] Hughes, T.J.R., Cottrell, J.A., Bazilevs, Y., 2005, "Isogeometric analysis: CAD, finite elements, NURBS,exact geometry and mesh refinement", *Computer methods in Applied Mechanical Engineering*, 194, pp. 4135-4195.

[18] Cottrell, J.,A., Reali, A., Bazilevs, Y., Hughes, T.,J.,R., 2006, "Isogeometric analysis of structural vibrations", *Comput. Methods Appl. Mech. Engrg.*, 195, pp. 5257–5296.

[19] Dörfel, M. R., Jüttler, B., Simeon, B., 2010, "Adaptive isogeometric analysis by local h-refinement with T-splines", *Computer methods in applied mechanics and engineering*, *199*(5), pp. 264-275.

[20] Kleiss, S. K., Jüttler, B., Zulehner, W., 2012, "Enhancing isogeometric analysis by a finite element-based local refinement strategy", *Computer Methods in Applied Mechanics and Engineering*, *213*, pp. 168-182.

[21] Bazilevs, Y., Calo, V. M., Cottrell, J. A., Evans, J. A., Hughes, T. J. R., Lipton, S., Scott, M.A., Sederberg, T. W., 2010, "Isogeometric analysis using T-splines", *Computer Methods in Applied Mechanics and Engineering*, *199*(5), pp 229-263.

[22] Vuong, A. V., Giannelli, C., Jüttler, B., Simeon, B., 2011, "A hierarchical approach to adaptive local refinement in isogeometric analysis", *Computer Methods in Applied Mechanics and Engineering*, *200*(49), pp. 3554-3567.

[23] Scott, M. A., Li, X., Sederberg, T. W., Hughes, T. J. R., 2012, "Local refinement of analysis-suitable T-splines", *Computer Methods in Applied Mechanics and Engineering*, *213*, pp. 206-222.

[24] Wang, P., Xu, J., Deng, J., Chen, F., 2011, "Adaptive isogeometric analysis using rational PHT-splines", *Computer-Aided Design*, *43*(11), pp. 1438-1448.

[25] Schillinger, D., Dede, L., Scott, M. A., Evans, J. A., Borden, M. J., Rank, E., Hughes, T. J. R., 2012, "An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces", *Computer Methods in Applied Mechanics and Engineering,* 249-252, pp. 116-150.

[26] Sederberg, T.,W., Zheng, J., Bakenov, A., Nasri, A., 2003, "T-splines and TNURCCs", *ACM Transactions on Graphics,* 22 (3), pp. 161–172.

[27] Dokken, T., Lyche, T., 2013, "Polynomial splines over locally refined box-partitions", *Computer Aided Geometric Design*, 30(3), pp. 331-356.

[28] Piegl L, Tiller W, 1997. The NURBS book. 2nd ed. Springer.

[29] Cook, R.D., Malkus, D.S., Plesha, M.E., 1989, Concepts and applications of finite element analysis, J. Wiley & Sons, Singapore.

[30] Carminelli, A., Catania, G., 2007, "Free vibration analysis of double curvature thin walled structures by a B-spline finite element approach", Proceedings of ASME IMECE 2007, pp.1-7, 2007 November 11-15, Seattle (Washington), USA.

[31] Carminelli, A., Catania, G., 2013, "A local refinement technique for B-spline finite element shell modeling", Proceedings of the ASME IMECE 2013, pp.1-7, November 15-21, 2013, San Diego (CA), USA. Under Review.