Studies of Refinement and Continuity in Isogeometric Structural Analysis

J.A. Cottrell [1], T.J.R. Hughes [2], and A. Reali [3]

Institute for Computational Engineering and Sciences, The University of Texas at Austin,
201 East 24th Street, 1 University Station C0200, Austin, TX 78712, USA

Abstract

We investigate the effects of smoothness of basis functions on solution accuracy within the isogeometric analysis framework. We consider two simple one-dimensional structural eigenvalue problems and two static shell boundary value problems modeled with trivariate NURBS solids. We also develop a local refinement strategy that we utilize in one of the shell analyses. We find that increased smoothness, that is, the "k-method," leads to a significant increase in accuracy for the problems of structural vibrations over the classical C^0-continuous "p-method," whereas a judicious insertion of C^0-continuous surfaces about singularities in a mesh otherwise generated by the k-method, usually outperforms a mesh in which all basis functions attain their maximum level of smoothness. We conclude that the potential for the k-method is high, but smoothness is an issue that is not well understood due to the historical dominance of C^0-continuous finite elements and therefore further studies are warranted.

Key words: isogeometric analysis, finite element analysis, *k*-method, *p*-method, refinement, continuity, smoothness, structural eigenvalue problems, shells, singularities

[1] Graduate Research Assistant (ICES)

[2] Professor of Aerospace Engineering and Engineering Mechanics, Computational and Applied Mathematics Chair III

[3] Postdoctoral Fellow, Department of Structural Mechanics, University of Pavia

Preprint submitted to Elsevier Science *26 January 2007*

Contents

1 Introduction

The concept of Isogeometric Analysis, introduced by Hughes, Cottrell and Bazilevs [7] and further developed by Cottrell *et al.* [3] and Bazilevs *et al.* [1], was initially motivated by the gap existing between Computer Aided Design (CAD) and Finite Element Analysis (FEA). The first manifestation of the gap is in the initial mesh generation process. The design is encapsulated in some type of CAD model. This model often includes ambiguities, such as gaps and overlaps, and levels of detail, such as individual bolts, welds, etc., that make it inappropriate for analysis. The ambiguities must be removed and defeaturing must be performed to arrive at an Analysis Suitable Geometry (ASG) that exactly represents the features of interest for the calculation (see Figure 1). This ASG must then be replaced with a finite element mesh, usually a piecewise polynomial approximation of the actual geometry. Creating a mesh can be one of the more time consuming steps in the analysis process.

Though initial mesh generation can be a significant bottleneck, additional difficulties are encountered during refinement. Frequently, if an accurate solution is to be obtained through a series of refinements, the quality of the geometric approxima-

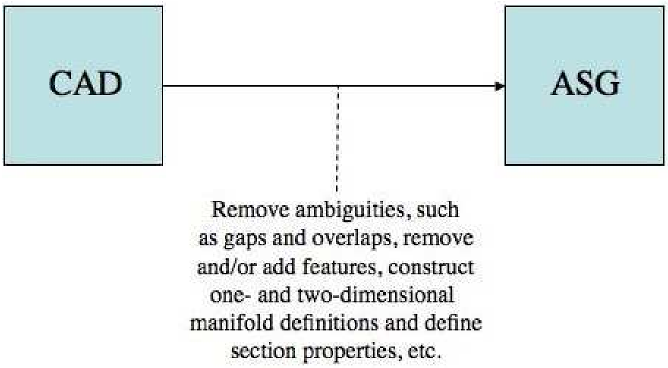

CAD
ASG
Remove ambiguities, such as gaps and overlaps, remove and/or add features, construct one- and two-dimensional manifold definitions and define section properties, etc.

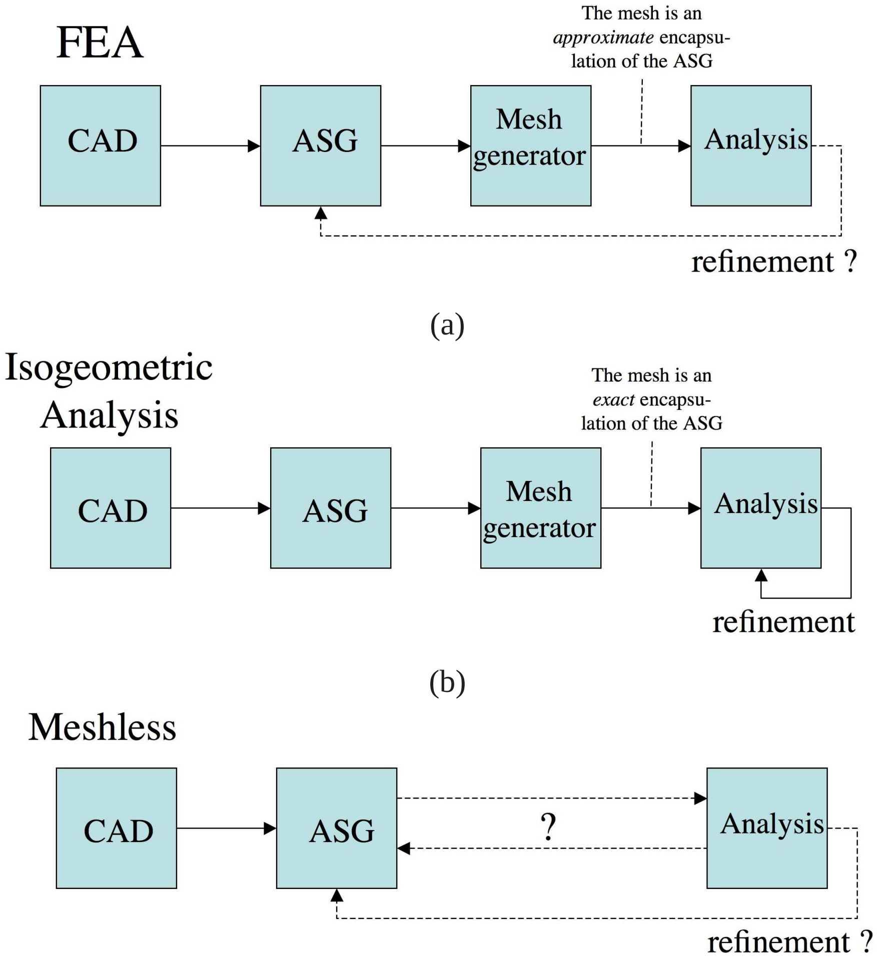
Fig. 1. The geometry of an object of engineering interest is initially encapuslated in a Computer Aided Design (CAD) package. The CAD description must frequently be changed significantly to create an Analysis Suitable Geometry (ASG).

tion must be simultaneously improved or else the error will reach a plateau from which it cannot be reduced. If such geometric refinement is to take place, a link must be established between the ASG and the refinement routine. This link usually does not exist in practice (see Figure 2a). This may be one of the reasons why automatic refinement has had little impact in industry despite the great promise shown in academic research studies.

Isogeometric analysis is a methodology for addressing these problems. The idea is to have one and only one representation of the geometry which exactly encapsulates the ASG and is more faithful to the initial CAD representation. While an ASG must still be constructed, using functions and technologies of the sort found in CAD packages may facilitate the development of links between the design and analysis software. More importantly, if the finite element mesh were to exactly encapsulate the ASG, refinement to any level could take place completely within the analysis framework. The need for reestablishing the link with an external description of the geometry would be completely obviated as the mesh would *be* the exact geometry, as in Figure 2b.

Our current implementation of the isogeometric analysis concept, based on Non-Uniform Rational B-Splines (NURBS), accomplishes this last task in almost all situations. The geometric flexibility of the NURBS basis allows for the exact representation of a much larger class of objects than standard finite element technology. Most notably, all conic sections can be represented exactly. At this point, generation of the initial mesh can still be a time consuming process but once it has been performed, the isogeometric mesh *encapsulates the exact geometry and may be refined to any level without ever altering this geometry in any way.*

Meshless methods do seem to share certain features with the isogeometric approach. The description of complicated geometries within such methods, however, has been almost entirely ignored in the literature. Notable exceptions are found in the papers of Subbarayan and colleagues [11, 20] and the recent work of Simkins



(a)

(b)

(c)

Fig. 2. The analysis process. a) In finite element analysis, mesh refinement requires interaction with an external description of the geometry if the quality of the geometric approximation is to be improved. The lack of such interaction is an impediment to adaptive mesh refinement procedures. b) In isogeometric analysis, the mesh *is* the exact geometry and so refinement can take place completely within the analysis framework. c) The literature on meshless methods is yet to present a comprehensive view how complex geometries may be represented and how that representation interacts with the process of refining the solution space.

et al. [15]. While meshless methods do show great promise in certain areas, a clear view of the proper way in which to define a geometry, as well as how that description affects both refinement of the solution space and, perhaps more importantly, numerical integration of the basis functions, is yet to emerge; see Figure 2c.

An important feature of the NURBS-based approach to isogeometric analysis, that was not one of its initial motivations, is the ability to use functions of higher order *and* higher continuity. Section 2 will describe the construction of NURBS basis functions that may have up to $p-1$ continuous derivatives across element boundaries, where p is the order of the underlying polynomial. This is seen in Section

3.1 to have a profound effect in structural vibration problems. The NURBS functions of higher continuity offer a much more compact representation of the vibrational modes of structures than do standard finite element functions, yielding much greater accuracy per degree of freedom, even at the same polynomial order. In Sections 3.2 and 3.3 we study shells modeled as trivariate NURBS solids. We explore local refinement and control of continuity. Results indicate that in regions with very large gradients, the use of functions with reduced continuity leads to more accurate results on a per-degree-of-freedom basis, at least on coarse meshes. These observations lead to the conclusion that local *control* of the continuity of the basis is a tool to be exploited in efficiently representing many types of solutions. In Section 4 we draw conclusions.

2 Overview of the Isogeometric Analysis Framework

Our current implementation of the isogeometric analysis concept is based on Non-Uniform Rational B-Splines (NURBS). This section will present an in depth discussion of the NURBS functions and their usage in representing various geometries comprised of a single NURBS patch. The myriad of refinement options encompassing classical h- and p-refinement, as well as the new k-refinement described in [7], will be discussed in detail. Lastly, we will describe the use of multiple patches and local refinement using constraint equations.

2.1 B-splines and NURBS

2.1.1 Knot Vectors

NURBS are built from B-splines and so a discussion of B-splines is a natural starting point for the investigation of NURBS. Unlike in standard FEA, the B-spline parametric space is local to "patches" rather than elements. Patches play the role of ***subdomains*** within which element types and material models are assumed to be uniform. However, a variety of refinement options may exist within a single patch. More about this later. Many simple domains can be represented by a single patch.

Note that the distinction between "elements" and "patches" may be thought of in two different ways. In [8] and [9], the patches themselves are referred to as elements. This is not unreasonable as the parametric space is local to patches and a finite element code must include a loop over the patches during assembly. As mentioned previously, we take the alternate view that patches are subdomains comprised of many elements, namely the "knot spans". This latter view seems more appropriate as, in our current code, numerical quadrature is being carried out at the knot span level. Furthermore, in the case of B-splines, the functions are piecewise polynomials where the different "pieces" join along knot lines. In this way the functions are C^∞ within an element. Lastly, surprisingly complicated domains may be described by a single patch (*e.g.*, all of the numerical examples in [7]). Describing such domains as being comprised of one element seems unnatural.

A ***knot vector*** in one dimension is a set of coordinates in the parametric space, written $\Xi = \{\xi_1, \xi_2, ..., \xi_{n+p+1}\}$, where $\xi_i \in \mathbb{R}$ is the i^{th} ***knot***, i is the knot index, $i = 1, 2, ..., n+p+1,\ \ p$ is the polynomial order, and n is the number of basis functions which comprise the B-spline. The knots partition the parameter space into elements. Element boundaries in the physical space are simply the images of knot lines under the B-Spline mapping, as shown in Figure 3.

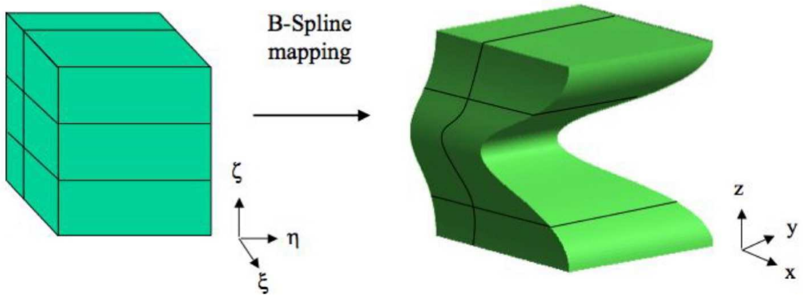


Fig. 3. The parametric space is local to "patches" rather than elements. The knots partition the patch into elements.

Knot vectors may be ***uniform*** if the knots are equally spaced in the parametric domain, or if they are unequally spaced, they are ***non-uniform***. Knot values may be repeated, that is, more than one knot may take on the same value. The multiplicities of knot values have important implications for the continuity properties of the basis. A knot vector is said to be ***open*** if its first and last knots appear $p+1$ times. Open knot vectors are standard in the CAD literature. In one dimension, basis functions formed from open knot vectors are interpolatory at the ends of the parametric space interval, $[\xi_1, \xi_{n+p+1}]$, and in multiple dimensions they are interpolatory at the corners of patches, but they are not, in general, interpolatory at interior knots. This is a distinguishing feature between knots and "nodes" in finite element analysis.

2.1.2 Basis functions

B-spline basis functions are defined recursively starting with piecewise constants $(p = 0)$:

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

For $p = 1, 2, 3, ...,$ they are defined by

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \tag{2}$$

The results of applying (1) and (2) to a uniform knot vector are presented in Figure 4. For B-spline functions with $p = 0$ and 1, we have the same result as for standard piecewise constant and linear finite element functions, respectively. Quadratic B-spline basis functions, however, are different than those in FEM. Each quadratic B-spline is identical but shifted. This distinguishes them from quadratic finite element functions, which are different for internal and end nodes. The homogeneous nature of the basis has implications for the quality of the approximation and the potential for efficient solution. In the case of structural vibrations, where the heterogeneity of finite element functions leads to a branching of the spectrum that degrades the accuracy of a large percentage of the computed frequencies, the homogeneity of B-spline functions leads to dramatic improvements, as will be shown later in Section 3.1.

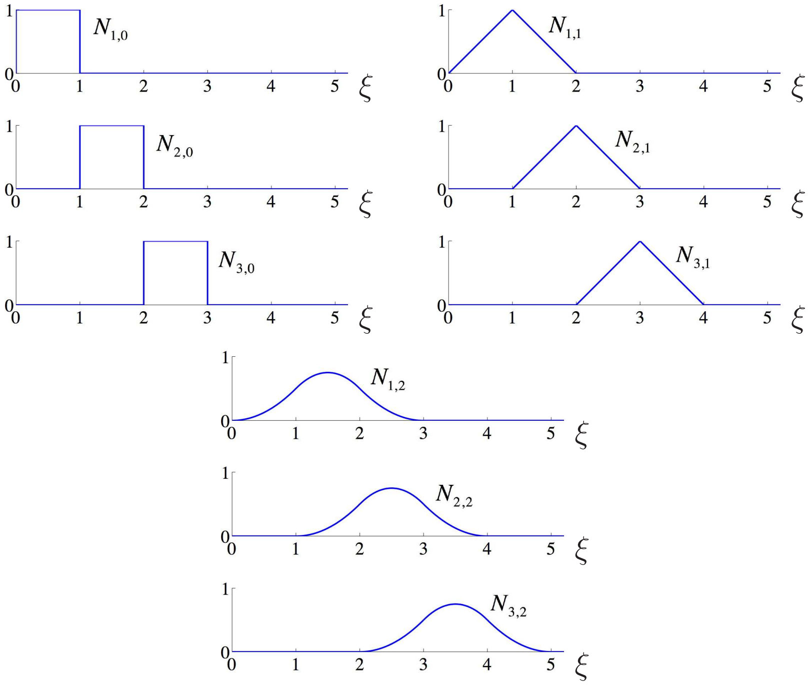


Fig. 4. Basis functions of order $0, 1, 2$ for uniform knot vector $\Xi = \{0, 1, 2, 3, 4, ...\}$.

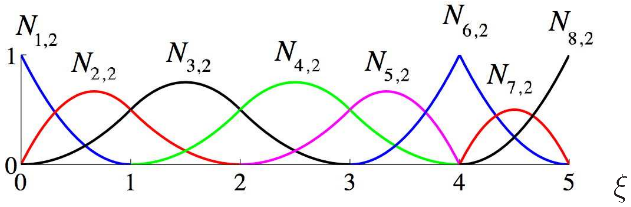


Fig. 5. Quadratic basis functions for open, non-uniform knot vector $\Xi = \{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$.

For an open, non-uniform knot vector we can attain much richer behavior. An example is presented in Figure 5. Note that the basis functions are interpolatory at the ends of the interval and also at $\xi = 4$, the location of a repeated knot, where only C^0-continuity is attained. Elsewhere, the functions are C^1-continuous. In general, basis functions of order p have $p - m_i$ continuous derivatives across knot ξ_i, where m_i is the multiplicity of the value of ξ_i in the knot vector. When the multiplicity of a knot value is exactly $p,$ the basis is interpolatory there. When the multiplicity is $p + 1,$ the basis becomes discontinuous and the patch is effectively split into two separate patches.

An important property of B-spline basis functions is that they constitute a partition of unity, that is, $\forall \xi$,

$$\sum_{i=1}^{n} N_{i,p}(\xi) = 1. \tag{3}$$

This is a feature they share with finite elements and meshless methods. Also of note is that the support of each $N_{i,p}$ is compact and contained in the interval $[\xi_i, \xi_{i+p+1}]$. Lastly, observe that each basis function is point-wise non-negative over the entire domain, that is, $N_{i,p}(\xi) \geq 0, \forall \xi$. This means that all of the entries of a mass matrix will be positive, which has implications for developing lumped mass schemes.

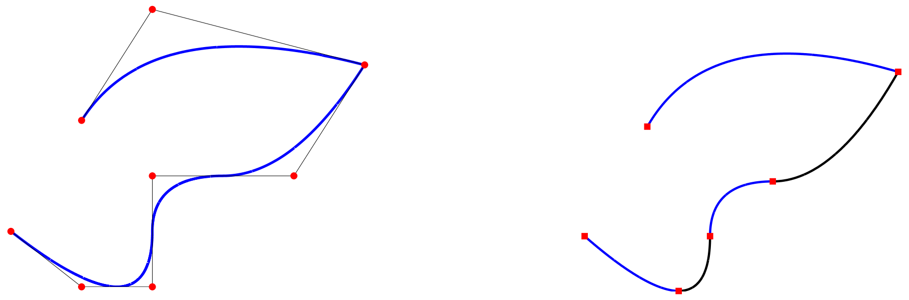
2.1.3 B-spline curves

B-spline curves in $\mathbb{R}^d$ are constructed by taking a linear combination of B-spline basis functions. The vector-valued coefficients of the basis functions are referred to as ***control points***. These are analogous to nodal coordinates in finite element analysis in that they are the coefficients of the basis functions, but the non-interpolatory nature of the basis does not lead to the usual interpretation of the control point values. Piecewise linear interpolation of the control points gives the so-called ***control polygon***. Again note that, in general, control points are not interpolated by B-spline curves. Given n basis functions, $N_{i,p}, i = 1, 2, ..., n,$ and corresponding control points $B_i \in \mathbb{R}^d, i = 1, 2, ..., n,$ a piecewise-polynomial ***B-spline curve*** is given by

$$C(\xi) = \sum_{i=1}^{n} N_{i,p}(\xi) B_i. \tag{4}$$

The example shown in Figure 6 is built from the quadratic basis functions considered in Figure 5. The curve is interpolatory at the first and last control points, a general feature of a curve built from an open knot vector. Note that it is also interpolatory at the sixth control point. This is due to the fact that the multiplicity of the knot at $\xi = 4$ is equal to the polynomial order. Note also that the curve is tangent to the control polygon at the first, last, and sixth control points. The curve is $C^{p-1} = C^1$-continuous everywhere except at the location of the repeated knot, $\xi = 4$, where it is $C^{p-2} = C^0$-continuous.

The properties of B-spline curves follow directly from the properties of their basis functions. For example, B-spline curves have continuous derivatives up to order $p-1$ in the absence of repeated knots or control points. Repeating a knot or control point k times decreases the number of continuous derivatives by k.



(a) Curve and control points (b) Curve and mesh denoted by knot locations

Fig. 6. B-spline, piecewise quadratic curve in $\mathbb{R}^2$. a) Control point locations are denoted by ●'s. b) The knots, which define a mesh by partitioning the curve into elements, are denoted by ■'s. Basis functions and knot vector as in Figure 5.

Affine transformations of a B-spline curve are obtained by applying the transformations directly to the control points. This turns out to be the essential property for satisfying so-called "patch tests," as discussed in [7]. This property is referred to as ***affine covariance***.

2.1.4 h-refinement: Knot insertion

The mechanism for implementing *h*-refinement is ***knot insertion***.[4] Knots may be inserted without changing a curve geometrically or parametrically. Given a knot vector $\Xi = \{\xi_1, \xi_2, ..., \xi_{n+p+1}\}$, let $\bar{\Xi} = \{\bar{\xi}_1 = \xi_1, \bar{\xi}_2, ..., \bar{\xi}_{n+m+p+1} = \xi_{n+p+1}\}$ be an *extended* knot vector such that $\Xi \subset \bar{\Xi}$. The new $n+m$ basis functions are formed as before by applying (1) and (2) to the new knot vector $\bar{\Xi}$. The new $n+m$ control points, $\bar{\mathbf{B}} = \{\bar{B}_1, \bar{B}_2, ..., \bar{B}_{n+m}\}^{\mathrm{T}}$, are formed from the original control points, $\mathbf{B} = \{B_1, B_2, ..., B_n\}^{\mathrm{T}}$, by

$$\bar{\mathbf{B}} = \mathbf{T}^p \mathbf{B} \tag{5}$$

where

$$T^0_{ij} = \begin{cases} 1 \; \bar{\xi}_i \in \quad \text{if} \quad [\xi_j, \xi_{j+1}) \\ 0 \; \text{otherwise} \end{cases} \tag{6}$$

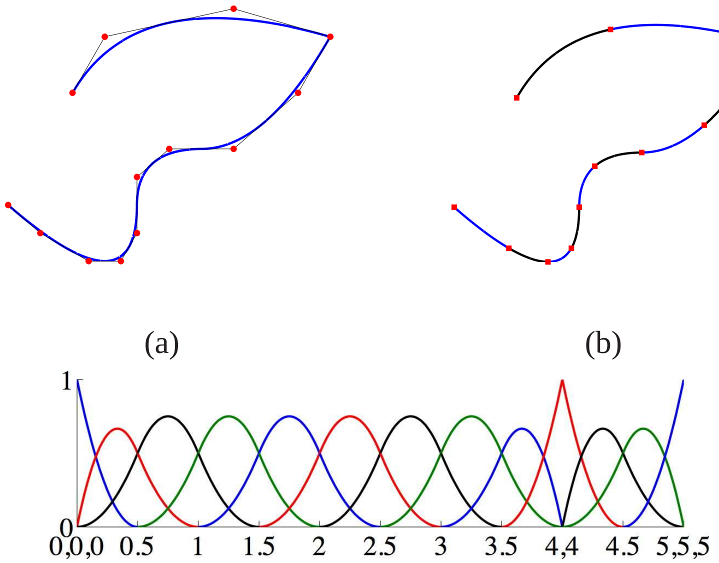
and

$$T^{q+1}_{ij} = \frac{\bar{\xi}_{i+q} - \xi_j}{\xi_{j+q} - \xi_j} T^q_{ij} + \frac{\xi_{j+q+1} - \bar{\xi}_{i+q}}{\xi_{j+q+1} - \xi_{j+1}} T^q_{ij+1} \quad \text{for} \quad q = 0, 1, 2, ..., p-1 \tag{7}$$

[4] Note that in the CAD literature "knot insertion" refers to inserting a single knot into a knot vector, whereas "knot refinement" refers to inserting multiple knots simultaneously. Here, we make no distinction and use "knot insertion" to refer to both cases. For an algorithm for inserting an individual knot, see [7].

Knot values already present in the knot vector may be repeated as above but, as described subsequently in Section 2.1.2, the continuity of the *basis* will be reduced. Continuity of the *curve* is preserved by choosing the control points as in (5), (6) and (7).

Figure 7 shows the case of a global refinement of the curve from Figure 6. Insertion of new knot values has parallels with the classical *h*-refinement strategy in finite element analysis as it splits existing elements into smaller ones. Repeating existing knot values to decrease the continuity of basis does not have an analogue in FEA. We will return to this idea later.


(a)

(b)

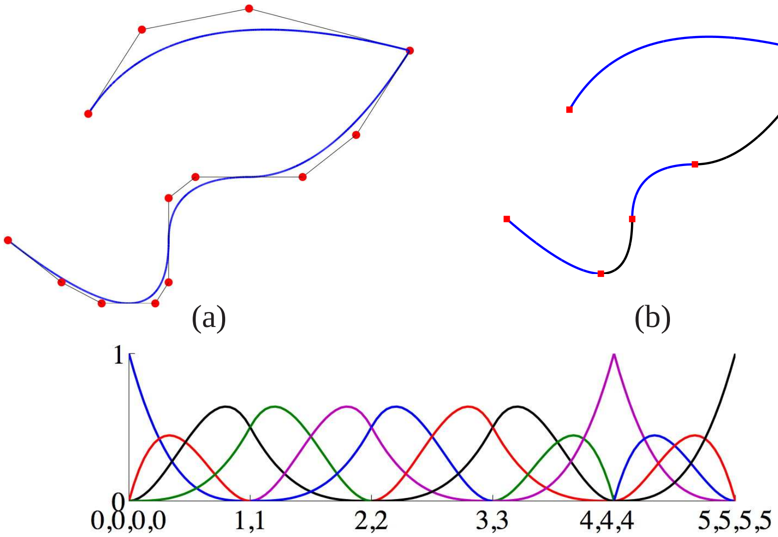
$$\bar{\Xi} = \{0, 0, 0, .5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4, 4.5, 5, 5, 5\}$$

Fig. 7. Knot insertion. a) New control points are computed from the original control points using (5). b) Each element has been split by inserting a new knot at the midpoint of each knot span.

2.1.5 p-refinement: Order elevation

The mechanism for implementing *p*-refinement is ***order elevation***[5] . As its name implies, the process involves raising the polynomial order of the basis functions used to represent the geometry (and the solution space, as our finite element implementation will be isoparametric). Recalling from Section 2.1.1 that the basis has $p - m_i$ continuous derivatives across element boundaries, it is clear that, when p is increased, m_i must also be increased if we are to preserve the discontinuities in the derivatives of our original curve. During order elevation, the *multiplicity* of each existing knot value is increased by one, but no new knot *values* are added. As with knot insertion, neither the geometry nor the parameterization are changed.

[5] sometimes also called “degree elevation.”



$$\bar{\Xi} = \{0, 0, 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 4, 5, 5, 5, 5\}$$

Fig. 8. Order elevation. a) New control points are calculated so as to preserve the geometry and parameterization. b) The mesh remains unchanged as no new elements have been created. Note the increased multiplicity of internal knots. This is done to preserve discontinuities in the derivatives of the curve.

The process for order elevation begins by replicating existing knots until their multiplicity is equal to the polynomial order, thus effectively subdividing the curve into many Bézier curves by knot insertion (see Rogers [14] or Farin [5] for a discussion of Bézier curves; we may think of them as one element B-spline curves). The next step is to elevate the order of the polynomial on each of these individual segments. Lastly, excess knots are removed to combine the segments into one, order-elevated, B-spline curve. Several efficient algorithms exist which combine the steps so as to minimize the computational cost of the process. Details are omitted for the sake of brevity. For a thorough treatment, see Piegl and Tiller [12].

Figure 8 shows this process applied to the curve in Figure 6. The multiplicities of the knots have been increased but no new elements created. Note that the locations of control points for these order-elevated curves are different than those in the *h*-refinement example (cf. Figure 7).

2.1.6 k-refinement: Higher order and higher continuity

As we have seen, the two primitive operations for B-splines are knot insertion and order elevation. Knot insertion is similar to *h*-refinement, but for it to be a perfect analogue, each new knot value would have to be inserted with multiplicity $m_i = p$ to ensure a C^0 basis everywhere. Similarly, if we begin with a mesh where all functions are already C^0 across element boundaries, order elevation coincides exactly with our traditional notion of *p*-refinement. Knot insertion and order elevation, however, provide us with more possibilities than the two standard notions of refinement.

As mentioned above, we can insert new knot values with multiplicities of 1 to define new elements across whose boundaries functions will be C^{p-1}. We can also repeat existing knot values to lower the continuity of the basis across existing element boundaries. This makes knot insertion a more flexible process than simple *h*-refinement. Similarly, we have a more flexible higher-order refinement as well. It stems from the fact that the processes of order elevation and knot insertion do not commute. If a unique knot value, $\bar{\xi}$, is inserted between two distinct knot values in a curve of order p, the number of continuous derivatives of the basis functions at $\bar{\xi}$ is $p - 1$. As described above, if we subsequently elevate to some higher order, q, the multiplicity of every distinct knot value (including the knot just inserted) is incremented $(q - p)$ times so that discontinuities in the p^{th} derivative of the basis are preserved. That is, the basis still has only $p - 1$ continuous derivatives at $\bar{\xi}$, although the order is now q. If instead we elevated the order of the original, coarsest curve to q and only then inserted the unique knot value $\bar{\xi}$, the basis would have $q-1$ continuous derivatives at $\bar{\xi}$. We refer to this latter procedure as ***k-refinement***. It has no analogue in standard finite element analysis [6] .

[6] This notion of *k*-refinement is *not* the same as the "*k*-convergence" described in [8] in which the position of the knots is altered. It bears more in common with the "k-version finite

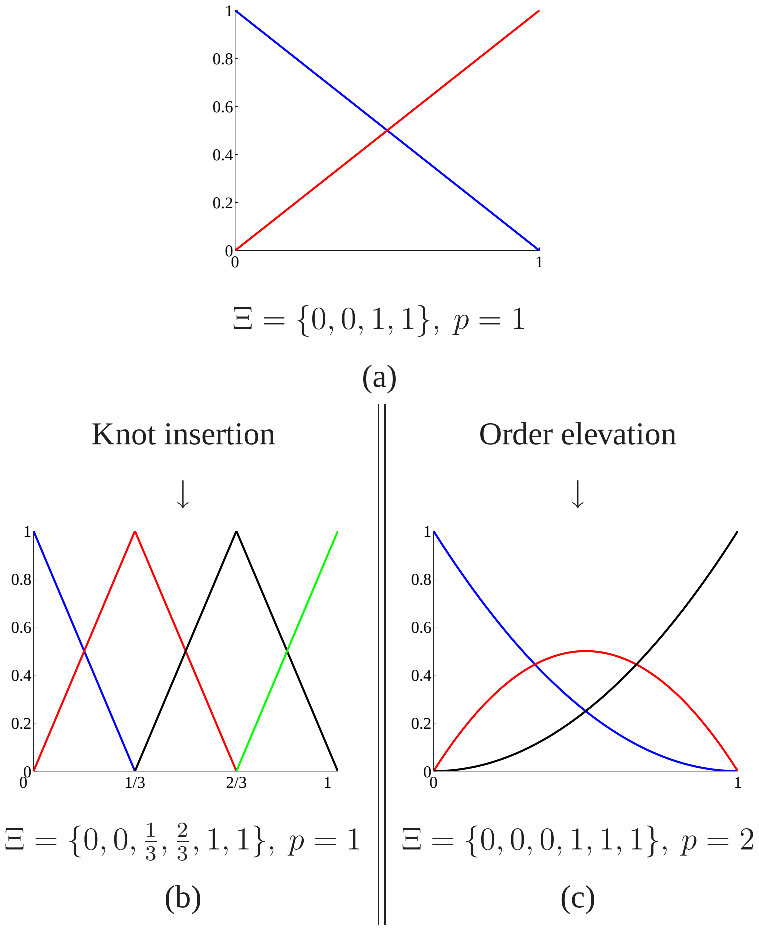
The concept of *k*-refinement is important because isogeometric analysis is fundamentally a higher-order approach. While linear finite elements can be represented within a NURBS context, it takes quadratic-level NURBS to represent conic sections – one of the key features of the method. In traditional *p*-refinement there is a very inhomogeneous structure to arrays due to the different basis functions associated with surface, edge, vertex and interior nodes. In addition, there is a proliferation in the number of nodes because C^0-continuity is maintained in the refinement process. In *k*-refinement, there is a homogeneous structure within patches and growth in the number of control variables is limited. Let us emphasize that an "element" in one dimension is defined as the span between two *distinct* knot values. The number of elements in a curve will then be the number of non-zero knot spans in the knot vector (*e.g.*, the domain associated with the knot vector $\Xi = \{0, 0, 0, 1, 2, 3, 3, 4, 4, 4\}$ consists of four elements).

Consider the classical *p*-refinement process. Assume the initial domain consists of one element and $p + 1$ basis functions (assuming an open knot vector), which we then refine by inserting new knot values until we have $n - p$ elements and n basis functions, all C^{p-1}. We then perform order elevation, maintaining continuity at the $p-1$ level. This requires replicating each distinct knot value, adding a basis function in each element and so increasing the total number of basis functions by $n - p$ to $2n - p$. After a total of r order elevations of this type, we have $(r + 1)n - rp$ basis functions, where p is still the order of our original basis functions. This is seen to be a large number of functions when one considers that in most cases of practical interest the number of elements will be quite a bit larger than the order of the basis. By comparison, consider beginning with the same one element domain and proceed by *k*-refinement. That is, order elevate r times adding only *one* basis function at each refinement, then insert knots until we have $n-p$ elements as before. The final number of basis functions is $n + r$, each having $r + p - 1$ continuity. This amounts to an enormous savings as $n+r$ is considerably smaller than $(r+1)n-rp$. Bear in mind that in d dimensions these numbers are raised to the d power. Recall that the mesh, defined by the knot *locations*, is fixed and is the same for *p*- and *k*-refinements. See Figures 9 and 10.

It is important to note that "pure" *k*-refinement, where all functions maintain C^{p-1} continuity across element boundaries, is only possible if the coarsest mesh is comprised of one element. If the initial mesh places constraints on the continuity across certain element boundaries, these constraints will exist on all meshes. In general, though some such constraints will exist, the number of elements desired for analysis

element method" of [16, 17] in that k refers to continuity, but the motivations are different. The increased continuity in [16] is required so that a least-squares finite element approach is possible. Such an approach requires that the solution space have the same number of continuous derivatives as found in the highest order derivative of the differential operator. Our motivations for using basis functions of higher continuity are efficiency and robustness of the solution space in a classical Galerkin finite element formulation of the problem.

will be much higher than the number needed for modeling the geometry. Refinements may be performed such that the functions have $p-1$ continuous derivatives across these new element boundaries and the benefits of *k*-refinement will still be significant.



$\Xi = \{0, 0, 1, 1\},\ p = 1$

(a)

$\Xi = \{0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1\},\ p = 1$

(b)

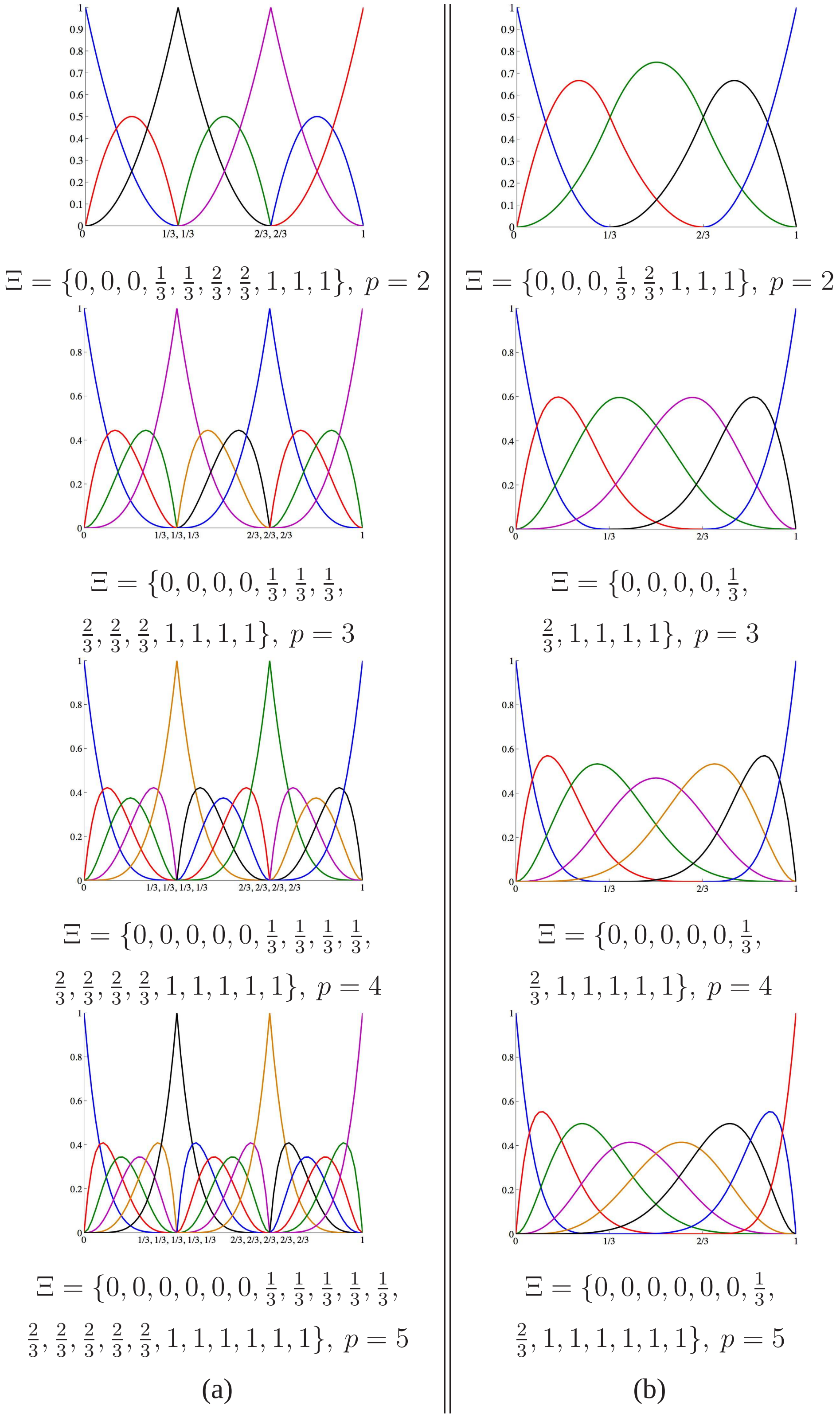
$\Xi = \{0, 0, 0, 1, 1, 1\},\ p = 2$

(c)

Fig. 9. When refining a coarse, low-order mesh to create a fine, higher-order mesh, one may choose between a p- or k-refinement strategy. Here we see the initial step for each case. (a) Base case of one linear element. (b) Classic *p*-refinement approach: knot insertion is performed first to create many low-order elements. Subsequent order elevation will preserve the C^0 continuity across element boundaries. c) New *k*-refinement approach: order elevation is performed on the coarsest discretization. Subsequent knot insertion will result a basis which is C^{p-1} across the newly created element boundaries. See the results of p- and k-refinement for several different polynomial orders in Figure 10.

2.1.7 The hpk-refinement space

As we have shown, knot insertion and order elevation are the primitive operations by which classical h- and p-refinements, as well as the new k-refinement, can be implemented. Recognizing their flexibility as compared with classical refinement procedures makes feasible the notion of an hpk-refinement space. Recalling that B-spline curves may have no more than $p-1$ continuous derivatives across an element



$\Xi = \{0,0,0,\frac{1}{3},\frac{1}{3},\frac{2}{3},\frac{2}{3},1,1,1\},\ p=2$

$\Xi = \{0,0,0,0,\frac{1}{3},\frac{1}{3},\frac{1}{3},$
$\frac{2}{3},\frac{2}{3},\frac{2}{3},1,1,1,1\},\ p=3$

$\Xi = \{0,0,0,0,0,\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3},$
$\frac{2}{3},\frac{2}{3},\frac{2}{3},\frac{2}{3},1,1,1,1,1\},\ p=4$

$\Xi = \{0,0,0,0,0,0,\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3},$
$\frac{2}{3},\frac{2}{3},\frac{2}{3},\frac{2}{3},\frac{2}{3},1,1,1,1,1,1\},\ p=5$

(a)

$\Xi = \{0,0,0,\frac{1}{3},\frac{2}{3},1,1,1\},\ p=2$

$\Xi = \{0,0,0,0,\frac{1}{3},$
$\frac{2}{3},1,1,1,1\},\ p=3$

$\Xi = \{0,0,0,0,0,\frac{1}{3},$
$\frac{2}{3},1,1,1,1,1\},\ p=4$

$\Xi = \{0,0,0,0,0,0,\frac{1}{3},$
$\frac{2}{3},1,1,1,1,1,1\},\ p=5$

(b)

Fig. 10. Three element, higher-order meshes for p- and k-refinement. a) The p-refinement approach results in many functions that are C^0 across element boundaries. b) In comparison, k-refinement results in a much smaller number of functions, each of which is C^{p-1} across element boundaries.

boundary, the set of possible refinements may be characterized as in Figure 11. Pure k-refinement keeps h fixed but increases the continuity along with the polynomial order, as in Figure 12. Pure p-refinement increases the polynomial order while the basis remains C^0, as in Figure 13. Increasing the multiplicity of existing knot values decreases the continuity without introducing new elements, as in Figure 14. Inserting new knot values with a multiplicity of p results in classical h-refinement, whereby new elements are introduced that have C^0 boundaries, shown in Figure 15. Inserting new knot values with a multiplicity of 1 decreases h without decreasing the minimum continuity already found in the mesh, as in Figure 16. Considering all of the aforementioned techniques results in a multitude of refinement options beyond simple h-, p- and k-refinement, see Figure 17.

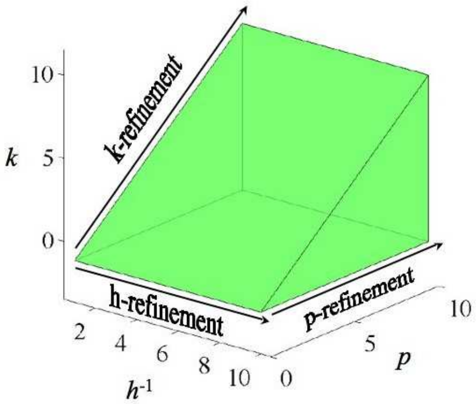


Fig. 11. The hpk-space. The set of all allowable refinements is contained in the region shown in green. Note that this region extends in the direction of the arrows.

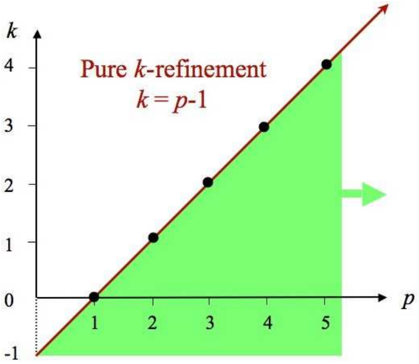


Fig. 12. The hpk-space. In pure k-refinement, the locations of the element boundaries (and thus element size, h) are fixed. As the polynomial order, p, is increased, the continuity of the functions across element boundaries, k, is increased such that $k = p - 1$ at all levels of refinement.

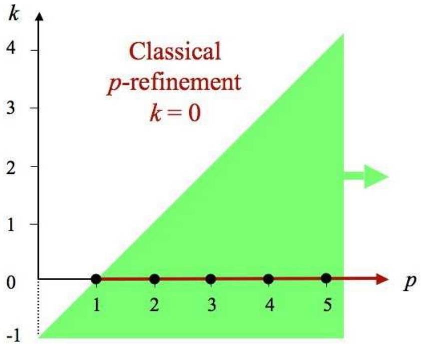


Fig. 13. The hpk-space. In pure p-refinement, the locations of the element boundaries (and thus element size, h) are fixed. As the polynomial order, p, is increased, the continuity of the functions across element boundaries, k, is fixed at $k = 0$ for all levels of refinement.

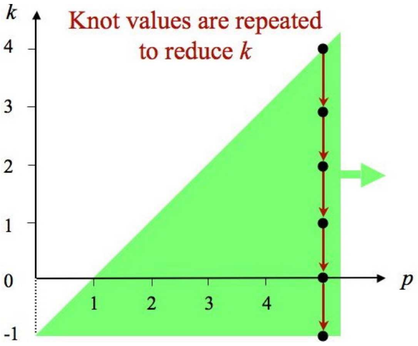


Fig. 14. The hpk-space. Repetition of existing knot values decreases the continuity across the corresponding element boundary without creating new elements or changing the polynomial order. The basis has $p - m_i$ continuous derivatives across knot ξ_i, where m_i is the multiplicity of that knot value.

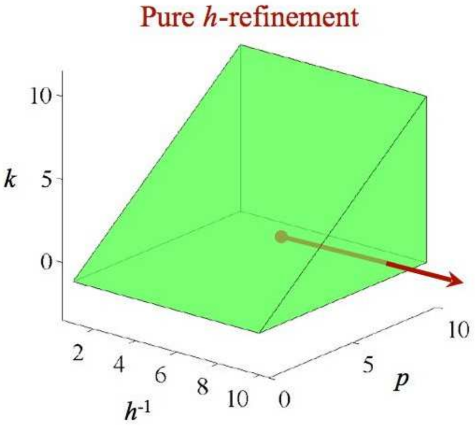


Fig. 15. The hpk-space. If we insert new knot values with multiplicity of p, new elements are created and the basis remains C^0 across all element boundaries. In this way classical h-refinement is exactly replicated.

2.1.8 Rational B-splines

As described in the beginning of this section, NURBS are formed from B-splines. Specifically, NURBS entities in $\mathbb{R}^d$ can be obtained by projective transformations of B-spline entities in $\mathbb{R}^{d+1}$, in particular, conic sections, such as circles and ellipses, can be *exactly* constructed by projective transformations of piecewise rational quadratic curves. The projective transformation of a B-spline curve yields a rational polynomial of the form $C_R(\xi) = f(\xi)/g(\xi)$, where f and g are piecewise polynomials. The construction of a rational B-spline curve in $\mathbb{R}^d$ proceeds as follows. Let $\{B_i^w\}$ be a set of control points for a B-spline curve in $\mathbb{R}^{d+1}$ with knot vector Ξ. These are referred to as the “projective control points” for the desired NURBS curve in $\mathbb{R}^d$. The control points in $\mathbb{R}^d$ are derived from the projective

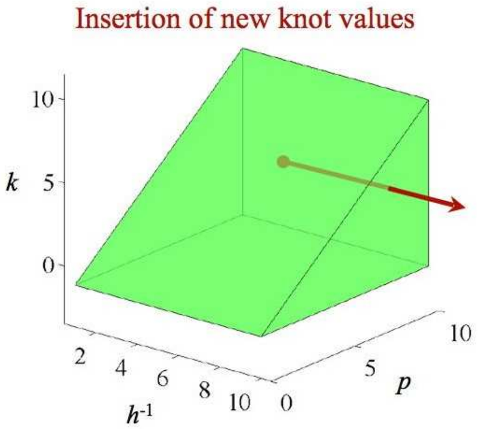


Fig. 16. The hpk-space. Insertion of new knot values with a multiplicity of 1 results in a splitting of elements, and thus a decrease in h (shown in the figure as an increase in h^{-1}). The basis has $p-1$ continuous derivatives across these new element boundaries, and so the (possibly lower) minimum continuity already existing in the mesh is unchanged, as is the polynomial order.

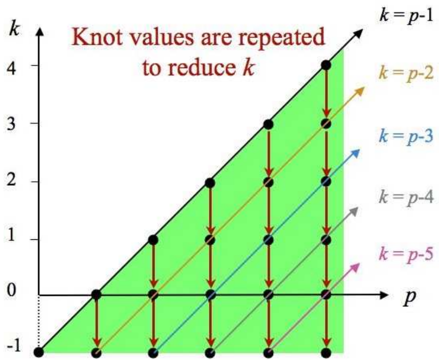


Fig. 17. The hpk-space. Combining knot insertion and order elevation in various permutations allows us to traverse the entire allowable refinement space.

control points by the following relations:

$$(B_i)_j = (B_i^w)_j / w_i, \quad j = 1, ..., d \tag{8}$$

$$w_i = (B_i^w)_{d+1} \tag{9}$$

where $(B_i)_j$ is the j^{th} component of the vector B_i, etc. and w_i is referred to as the i^{th} ***weight***. The rational basis functions and NURBS curve are given by

$$R_i^p(\xi) = \frac{N_{i,p}(\xi) w_i}{\sum_{\hat{i}=1}^{n} N_{\hat{i},p}(\xi) w_{\hat{i}}} \tag{10}$$

$$C(\xi) = \sum_{i=1}^{n} R_i^p(\xi) B_i \ . \tag{11}$$

Rational surfaces and solids are defined analogously in terms of the rational basis functions

$$R_{i,j}^{p,q}(\xi,\eta) = \frac{N_{i,p}(\xi) M_{j,q}(\eta) w_{i,j}}{\sum_{\hat{i}=1}^{n} \sum_{\hat{j}=1}^{m} N_{\hat{i},p}(\xi) M_{\hat{j},q}(\eta) w_{\hat{i},\hat{j}}} \tag{12}$$

$$R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta) = \frac{N_{i,p}(\xi) M_{j,q}(\eta) L_{k,r}(\zeta) w_{i,j,k}}{\sum_{\hat{i}=1}^{n} \sum_{\hat{j}=1}^{m} \sum_{\hat{k}=1}^{l} N_{\hat{i},p}(\xi) M_{\hat{j},q}(\eta) L_{\hat{k},r}(\zeta) w_{\hat{i},\hat{j},\hat{k}}} \tag{13}$$

The powerful thing about the construction of the NURBS basis functions is that, as NURBS in $\mathbb{R}^d$ are B-splines in $\mathbb{R}^{d+1}$, all of the refinement techniques we have discussed are applied to NURBS by operating directly on those higher dimensional B-splines. The NURBS basis functions also form a partition of unity. The continuity and support of NURBS are the same as for B-splines. Affine transformations in physical space are still obtained by applying the transformation to the control points, that is, NURBS possess the property of affine covariance.

2.2 Multiple patches and local refinement

In almost all practical circumstances, it will be required to describe a domain with multiple NURBS patches. For example, if different material or physical models are to be used in different parts of the domain, it might simplify things to describe these subdomains by different patches. Also, if different subdomains are to be assembled in parallel on a multiple processor machine, it is convenient from the point of view of data structures to not have a single patch split between different processors. Most common is the case where the domain simply differs topologically from a cube. The tensor product structure of the parameter space of a patch makes it poorly suited for representing complex, multiply connected domains. Such geometries can frequently be handled quite simply by using multiple patches (see, *e.g.*, Figure 18).

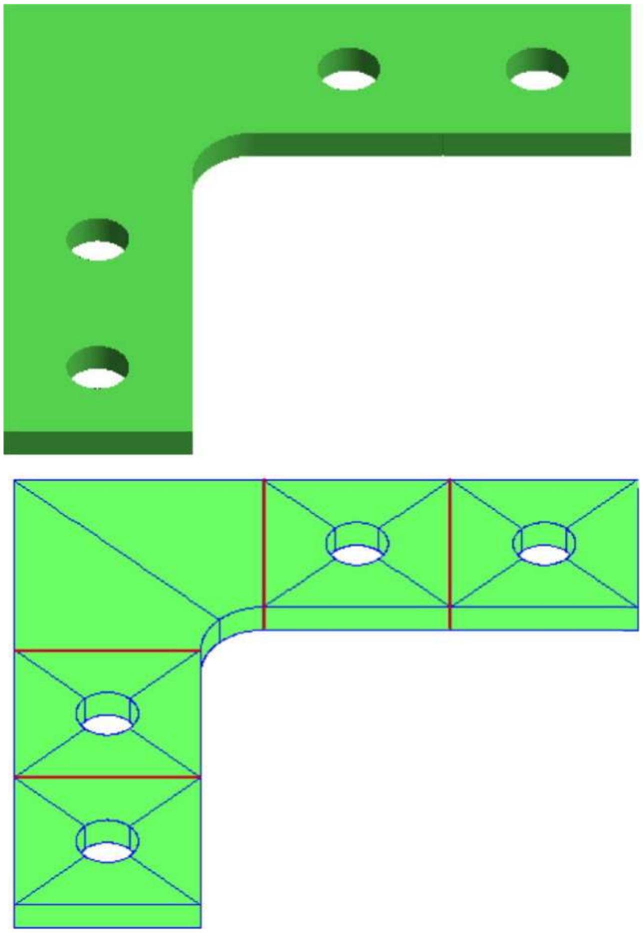


Fig. 18. The bracket on the top is exactly and concisely represented by five simple NURBS patches (patch boundaries are shown in red, element boundaries in blue). The patches match geometrically and parametrically on the internal faces where they meet.

Even in cases where a cube can be mapped into the desired object, doing so might introduce such extreme mesh distortion and widely varying Jacobians within elements that analysis will be adversely affected. Figure 19b (from [7]) shows the amount of mesh distortion needed to represent the “stiffened shell” of Figure 19a with a single NURBS patch. A mesh using multiple patches, shown in Figure 19c, exhibits far less distortion and yields a much more “natural” mesh.

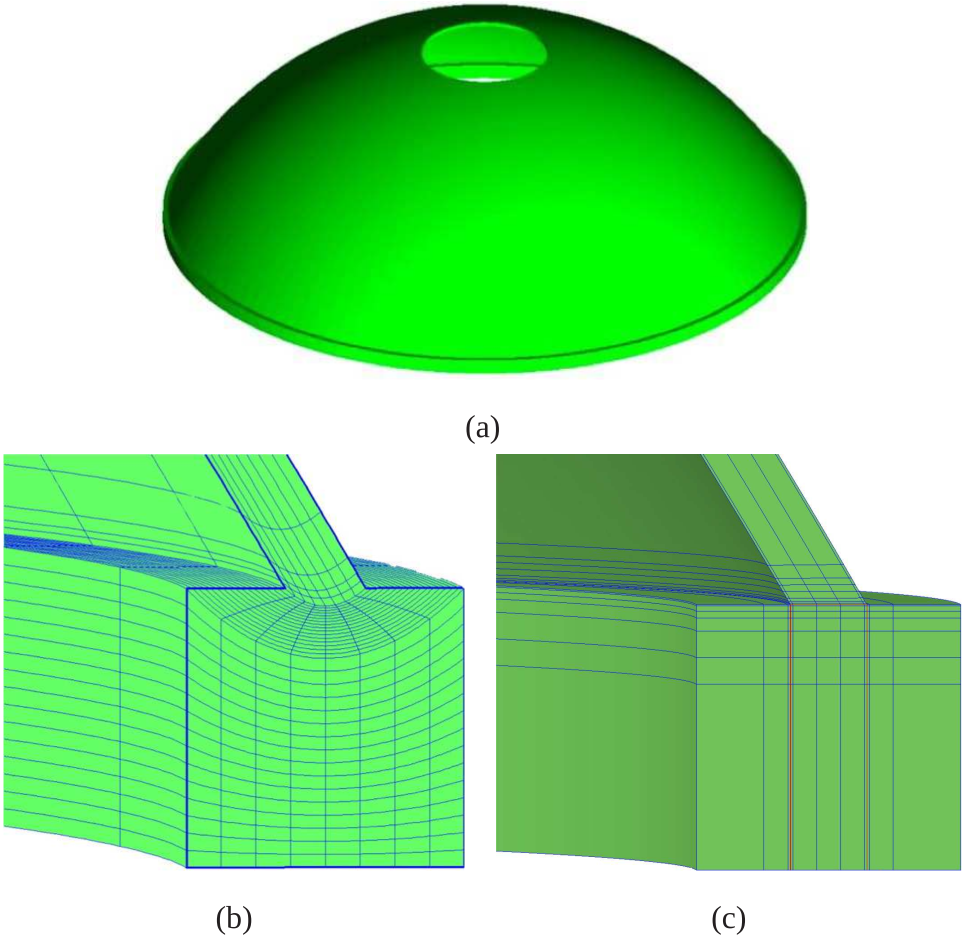


Fig. 19. Multiple patches usually produce better quality meshes. (a) The stiffened shell of [7] can be modeled using a single NURBS patch. (b) Such a mapping produces severe mesh distortion that is unavoidable when using a single patch. (c) Allowing the shell and the stiffener to be modeled by different patches creates a much more natural mesh. Patch boundaries shown in red.

Another reason for using multiple patches is that it makes local refinement possible. The situation is represented in Figure 20. Even with multiple patches, if we want the control points of the two patches on their interface to be in one-to-one correspondence, we need to have matching knot vectors. This means that refinements of one patch must necessarily propagate from that patch to the next. If we are to allow knots to be inserted on one side and not the other (*i.e.*, local refinement), we may proceed as follows.

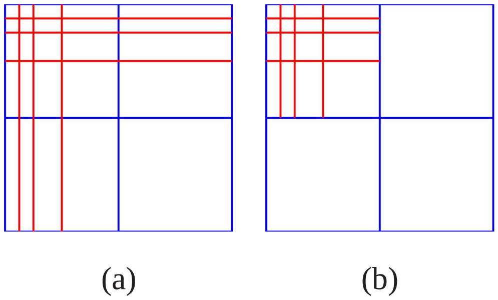


Fig. 20. (a) Global refinement employing the continuous Galerkin method. (b) Local refinement employing the discontinuous Galerkin method or constraint equations at the patch level. With constraint equations, at least C^0-continuity can be attained across patches, and higher-order continuity can be achieved in certain cases if desired.

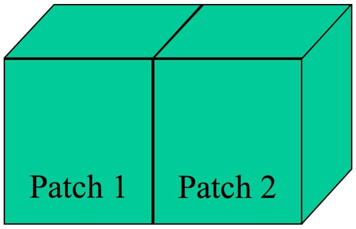


Fig. 21. The two patches share a common interface. On the coarsest mesh, their control points on that interface are in one-to-one correspondence, trivially enforcing C^0 continuity.

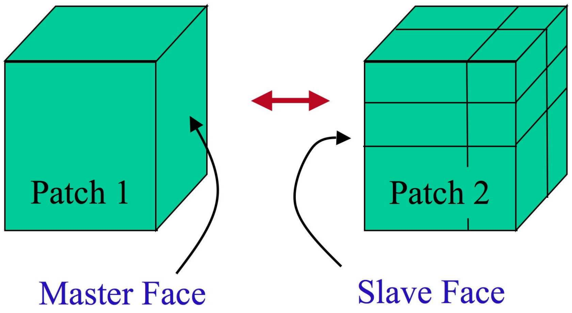


Fig. 22. As Patch 2 is refined by knot insertion and the one-to-one correspondence of the interface control points is lost. Constraint equations may be employed to ensure that continuity is maintained.

Consider the two B-spline [7] patches that meet on an interface, as shown in Figure 21. On the coarsest mesh, we assume that the control points and knot vectors in the plane of the face are identical on both patches, thus ensuring that the patches match geometrically and parametrically on that shared face. Using superscripts 1 and 2 to identify the patch numbers, a subscript f to denote control points on the face where the patches meet, and a subscript n to denote control points *not* on that face, we

[7] We will discuss the B-spline case here, but it is crucial to note that if we were to use NURBS rather than B-splines, all of the relationships in this section must hold for the *projective* control points and *projective* control variables.

may write the control points for Patches 1 and 2 as

$$\mathbf{B}^1 = \begin{pmatrix} \mathbf{B}_n^1 \\ \mathbf{B}_f^1 \end{pmatrix} \quad \text{and} \quad \mathbf{B}^2 = \begin{pmatrix} \mathbf{B}_n^2 \\ \mathbf{B}_f^2 \end{pmatrix}, \tag{14}$$

respectively, where

$$\mathbf{B}_f^2 = \mathbf{B}_f^1. \tag{15}$$

If we now refine the basis of Patch 2 by knot insertion, then we have the following new set of control points for Patch 2:

$$\tilde{\mathbf{B}}^2 = \tilde{\mathbf{T}}\mathbf{B}^2 = \begin{pmatrix} \tilde{\mathbf{T}}_n & 0 \\ 0 & \tilde{\mathbf{T}}_f \end{pmatrix} \begin{pmatrix} \mathbf{B}_n^2 \\ \mathbf{B}_f^2 \end{pmatrix}, \tag{16}$$

where $\tilde{\mathbf{T}}$ is the multi-dimensional generalization of the extension operator defined in (7). As before, it is sparse and its values are entirely defined by the knot vectors and the polynomial orders. The block diagonal structure follows from the fact that we are using open knot vectors. When open knot vectors are used, each face of a NURBS solid is influenced only by the control points on that face. Put simply, each face of the NURBS solid is a NURBS surface.

Combining (15) and (16), we see that C^0-continuity of the geometry is maintained by the relationship

$$\tilde{\mathbf{B}}_f^2 = \tilde{\mathbf{T}}_f \mathbf{B}_f^1. \tag{17}$$

Building on the approach of Kagan, Fischer and Bar-Yoseph [9] [8] , it follows that for our solution space to enforce the same continuity constraints, we need our control variables to obey precisely the same relationship. Let

$$\mathbf{u}^1 = \begin{pmatrix} \mathbf{u}_n^1 \\ \mathbf{u}_f^1 \end{pmatrix} \quad \text{and} \quad \mathbf{u}^2 = \begin{pmatrix} \mathbf{u}_n^2 \\ \mathbf{u}_f^2 \end{pmatrix} \tag{18}$$

be the control variables on Patch 1 and the refined Patch 2, respectively. Then C^0-continuity of the solution across the interface between the patches may be maintained by enforcing the constraint

$$\mathbf{u}_f^2 = \tilde{\mathbf{T}}_f \mathbf{u}_f^1. \tag{19}$$

From an implementational point of view, the two patches may be assembled locally to create the two local problems

$$\mathbf{K}^1\mathbf{u}^1 = \mathbf{b}^1 \tag{20}$$

[8] In [9], a similar approach was taken for B-Splines *surfaces*. Here we extend that to NURBS *solids*.

and

$$\mathbf{K}^2\mathbf{u}^2 = \mathbf{b}^2 \tag{21}$$

for the control points on either patch. Consistent with the partitioning of the control variables in (18), we partition the stiffness matrices as

$$\mathbf{K}^1 = \begin{pmatrix} \mathbf{K}^1_{nn} & \mathbf{K}^1_{nf} \\ \mathbf{K}^1_{fn} & \mathbf{K}^1_{ff} \end{pmatrix} \quad \text{and} \quad \mathbf{K}^2 = \begin{pmatrix} \mathbf{K}^2_{nn} & \mathbf{K}^2_{nf} \\ \mathbf{K}^2_{fn} & \mathbf{K}^2_{ff} \end{pmatrix}. \tag{22}$$

Before solving, we must assemble problems (20) and (21) into one global problem accounting for the behavior of both patches, as well as their interaction. We should have three coupled blocks of equations: one corresponding to weighting functions with support in Patch 1 that vanish on the face shared by the two patches, one corresponding to weighting functions with support on either or both patches that do *not* vanish on the shared face, and one corresponding to weighting functions with support on Patch 2 that vanish on the shared face. We begin by expanding (20) using the partitioning of (22) to get

$$\mathbf{K}^1_{nn}\mathbf{u}^1_n + \mathbf{K}^1_{nf}\mathbf{u}^1_f = \mathbf{b}^1_n \tag{23}$$

and

$$\mathbf{K}^1_{fn}\mathbf{u}^1_n + \mathbf{K}^1_{ff}\mathbf{u}^1_f = \mathbf{b}^1_f. \tag{24}$$

Inserting (19) into (21) and expanding yields

$$\mathbf{K}^2_{nn}\mathbf{u}^2_n + \mathbf{K}^2_{nf}\tilde{\mathbf{T}}_f\mathbf{u}^1_f = \mathbf{b}^2_n \tag{25}$$

and

$$\mathbf{K}^2_{fn}\mathbf{u}^2_n + \mathbf{K}^2_{ff}\tilde{\mathbf{T}}_f\mathbf{u}^1_f = \mathbf{b}^2_f. \tag{26}$$

Note that (23) is the block of equations corresponding to weighting functions in Patch 1 that vanish on the shared face. Similarly, (25) is the block of equations corresponding to weighting functions in Patch 2 that vanish on the shared face. Now (24) and (26) both correspond to weighting functions with support on the shared face and as such we would like to add them together to get a final expression for that block. Unfortunately, they contain different numbers of equations. This is because we assembled the two patches independently. We correctly generated the equations in (24) by testing against functions in the "master" weighting space associated with Patch 1, but we generated the equations in (26) by testing against all of the functions in the larger "slave" weighting space on Patch 2 without regard for the constraint. Just as the basis functions of the slave solution space on Patch 2 corresponding to the shared face are restricted to act only in the linear combinations defined by $\tilde{\mathbf{T}}_f$ that result in functions existing in the master solution space, so too must the functions in the slave weighting space act only in such linear combinations as replicate functions in the master weighting space. This constraint may be enforced by now premultiplying (26) by $\tilde{\mathbf{T}}_f^{\mathrm{T}}$, thus constraining the weighting functions and reducing the number of equations to match that of (24):

$$\tilde{\mathbf{T}}_f^{\mathrm{T}}\mathbf{K}_{fn}^2\mathbf{u}_n^2 + \tilde{\mathbf{T}}_f^{\mathrm{T}}\mathbf{K}_{ff}^2\tilde{\mathbf{T}}_f\mathbf{u}_f^1 = \tilde{\mathbf{T}}_f^{\mathrm{T}}\mathbf{b}_f^2. \tag{27}$$

We may now express the global system comprised of (23), (25), and $\Big($(24)+(27)$\Big)$ as

$$\mathbf{K}\mathbf{u} = \mathbf{b}, \tag{28}$$

where

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_{nn}^1 & \mathbf{K}_{nf}^1 & 0 \\ \mathbf{K}_{fn}^1 & (\mathbf{K}_{ff}^1 + \tilde{\mathbf{T}}_f^{\mathrm{T}}\mathbf{K}_{ff}^2\tilde{\mathbf{T}}_f) & \tilde{\mathbf{T}}_f^{\mathrm{T}}\mathbf{K}_{fn}^2 \\ 0 & \mathbf{K}_{nf}^2\tilde{\mathbf{T}}_f & \mathbf{K}_{nn}^2 \end{pmatrix}, \tag{29}$$

$$\mathbf{u} = \begin{pmatrix} \mathbf{u}_n^1 \\ \mathbf{u}_f^1 \\ \mathbf{u}_n^2 \end{pmatrix}, \tag{30}$$

and

$$\mathbf{b} = \begin{pmatrix} \mathbf{b}_n^1 \\ \mathbf{b}_f^1 + \tilde{\mathbf{T}}_f^{\mathrm{T}}\mathbf{b}_f^2 \\ \mathbf{b}_n^2 \end{pmatrix}. \tag{31}$$

We may recover $\mathbf{u}_f^2$ via (19) after solving (28).

This approach ensures C^0 continuity in the solution across the patch boundary when one patch is a knot refined version of the other patch on their common interface. Higher continuity has also been implemented by applying similar constraint equations in the normal direction. As long as the geometries are compatible, the patch boundary may be seen as the result of inserting a knot into some “metapatch” $p+1$ times. It should be noted that these are strong, exact constraints, not approximations. An approach that would allow for weak enforcement of continuity, as well as allowing for local order elevation is to use discontinuous Galerkin techniques at the patch level. That is, weakly enforce continuity of appropriate fluxes across patch boundaries while strongly enforcing them across element boundaries within the patch.

Remark

It is important to note that these operations could also be applied over the entire domain rather than just for the interface between patches. These could be used in a multigrid scheme where the grid transfer operator would be $\tilde{\mathbf{T}}$. This could potentially be very efficient as $\tilde{\mathbf{T}}$ is uniquely defined by the knot vectors and thus its construction is very inexpensive.

3 Numerical examples

In Section 2.1.6 we compared the number of degrees-of-freedom in k- and p-refined meshes. We found that, for the same mesh and polynomial order, k-refinement involved many fewer degrees-of-freedom than p-refinement. This suggests to us that k-refinement may be a more *efficient* procedure than p-refinement. However, this is not completely clear because other factors are at play. A traditional way to assess efficiency is by comparing accuracy on a per degree-of-freedom basis, although this may not be entirely satisfactory either. Nevertheless, to get some sense of the relative efficiency of k-refined and p-refined meshes, we will adopt this approach in the following numerical examples. We will often refer to p-refined meshes simply as "finite elements" and k-refined meshes as "NURBS."

3.1 Vibrations of beams and rods

We study the problem of the structural vibrations of an elastic fixed-fixed rod of unit length, whose natural frequencies and modes, assuming unit material parameters, are governed by:

$$\begin{aligned} &u_{,xx} + \omega^2 u = 0 \text{ for } x \in \left]0,1\right[\\ &u(0) = u(1) = 0, \end{aligned} \tag{32}$$

and for which the exact natural frequencies are:

$$\omega_n = n\pi, \text{ with } n = 1, 2, 3... \tag{33}$$

As a first numerical experiment, the eigenproblem is solved with both finite elements and isogeometric analysis using quadratic basis functions. The resulting natural frequencies, ω_n^h, are presented in Figure 23, normalized with respect to the exact solution (33), and plotted versus the mode number, n, normalized by the total number of degrees-of-freedom, N. To produce the spectra of Figure 23, we used $N = 999$ but the results are in fact independent of N.

Figure 23 illustrates the superior behavior of NURBS basis functions compared with finite elements. In this case, the finite element results depict an acoustical branch for $n/N < 0.5$ and an optical branch for $n/N > 0.5$ (see Brillouin [2]). As we go to higher-order, the disparity becomes even greater. Higher-order NURBS outperform higher-order finite elements by an ever increasing margin, see Figure 24.

Additionally, transverse vibrations of a simply-supported, unit length Bernoulli-Euler beam are considered (see Hughes [6], Chapter 7). For this case, the natural frequencies and modes, assuming unit material and cross-sectional parameters, are

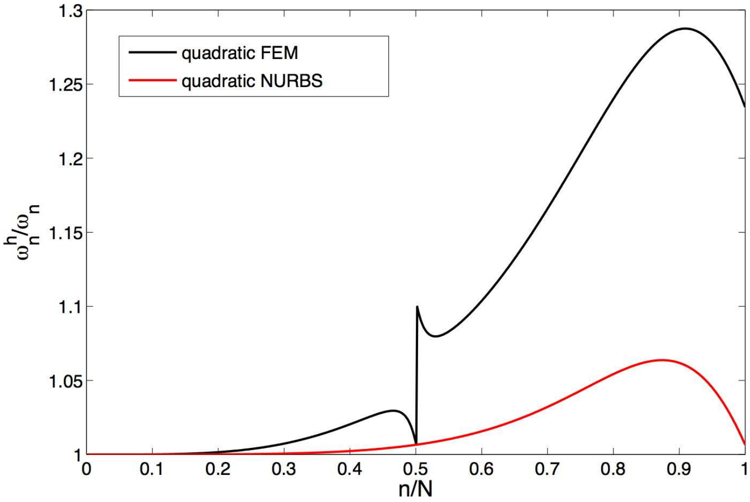


Fig. 23. Fixed-fixed-rod. Normalized discrete spectra for quadratic finite elements and NURBS.

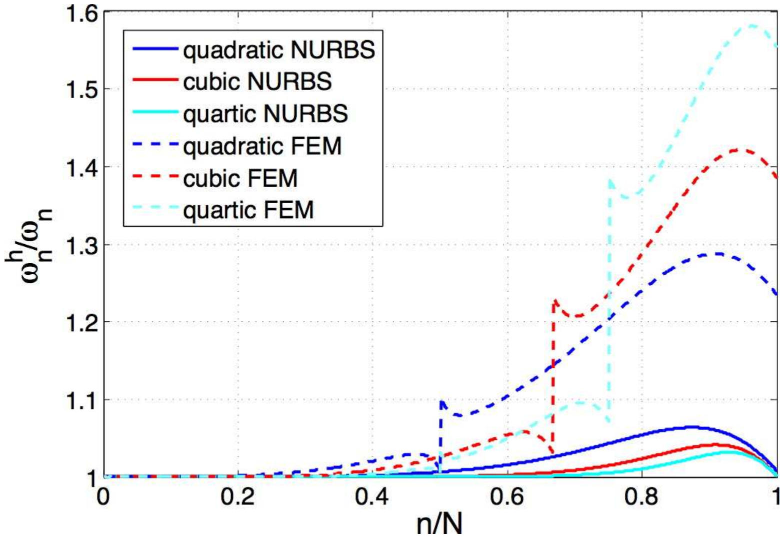


Fig. 24. Fixed-fixed-rod. Normalized discrete spectra for higher-order finite elements and NURBS.

governed by:

$$\begin{aligned} &u_{,xxxx} - \omega^2 u = 0 \text{ for } x \in]0,1[\\ &u(0) = u(1) = u_{,xx}(0) = u_{,xx}(1) = 0, \end{aligned} \tag{34}$$

where

$$\omega_n = (n\pi)^2, \text{ with } n = 1, 2, 3, ... \tag{35}$$

The numerical experiments and results for the Bernoulli-Euler beam problem are analogous to the ones reported for the rod. Note that the classical beam finite element employed to solve problem (34) is a two-node Hermite cubic element with two degrees-of-freedom per node (transverse displacement and rotation), whereas our isogeometric analysis formulation is rotation-free (see, for example, Engel *et al*. [4]). Figure 25 presents the discrete spectra obtained using different order finite element and NURBS basis functions. Again, *k*-refinement results are dramatically better on a per degree-of-freedom basis.

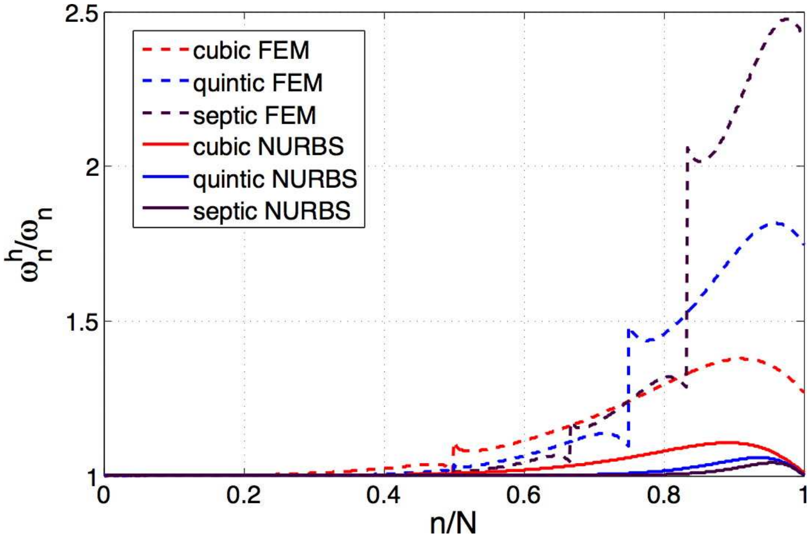


Fig. 25. Simply-supported beam. Normalized discrete spectra for higher-order finite elements and NURBS.

Remark

It is very important to observe the trends in Figures 24 and 25. For finite elements, the optical branches of the frequency spectra *diverge* as p is increased. That is, the errors in the higher frequencies get worse as p is increased. It is well-known that higher frequencies are inaccurate in finite element analysis, but it is apparently a new observation that they get progressively worse as p is increased. On the other hand, for NURBS, *the entire spectrum converges* as p is increased. These opposite trends may be very important in applications such as wave propagation and turbulence, in which the *entire* discrete spectrum may participate significantly in the solution. We conjecture that NURBS, capable of attaining almost spectral accuracy on patches, as evidenced by Figures 24 and 25, may be superior to classical, higher-order, C^0-continuous finite elements in these applications. It may also be noted that, based on similar studies, NURBS exhibit superior accuracy compared to finite elements for first-order spatial operators. This buttresses the belief that NURBS should be capable of attaining better accuracy than finite elements in representing wave phenomena and turbulence.

3.2 Hyperboloidal shell

The hyperboloidal shell problem was introduced to us by Prof. Barna Szabó. His group had analyzed the structure using a p-refinement strategy on meshes created using quasi-regional mappings based on optimal collocation at Babuška points [18]. He was interested in an independent estimate of the limit value of the potential energy and suggested we investigate it as our isogeometric approach is capable of exactly representing the conic sections in the geometry. The problem was considered previously by Lee and Bathe [10] using shell elements, but unfortunately they do not report the potential energy in their results.

The domain is the thin-walled solid seen in Figure 26, whose mid-surface is defined by

$$x^2 + z^2 - y^2 = 1, \quad y \in [-1, 1]. \tag{36}$$

The structure has a thickness of $t = 0.001$ in the direction normal to this mid-surface (all distances are in meters). The loading is a smoothly varying pressure normal to the surface,

$$p(\theta) = p_0 \cos(2\theta), \tag{37}$$

with $p_0 = 1.0$ MPa. The top and bottom of the structure are fixed.

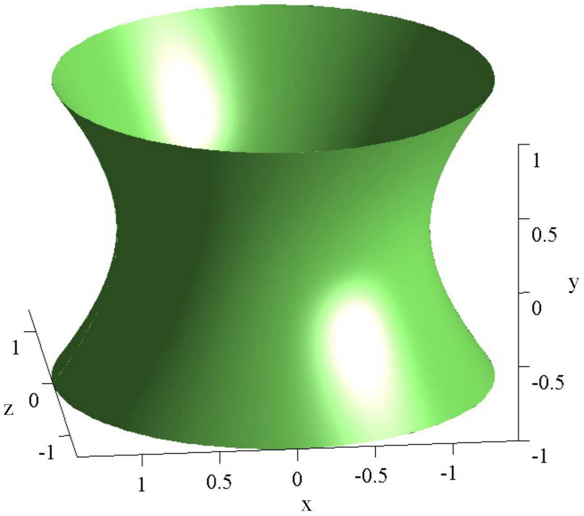


Fig. 26. The geometry of the hyperboloidal shell.

3.2.1 Mesh generation and implementation

Only a quarter of the structure is modeled due to symmetry. The mid-surface is a conic section, namely a hyperbola, extruded in a path defined by another conic section, a circle. As rational quadratic NURBS are capable of representing all conic sections, this hyperboloidal surface of revolution can be represented exactly. However, the inner and outer surfaces of the structure are defined as offsets of the mid-surface, shifted by $\pm t/2$ in the normal direction, and are not conic sections. Moreover, they are not in the NURBS space, so our mesh will inherently be an approximate geometry [9] .

The decision was made to use two quadratic elements through the thickness of the structure (see Figure 27). The knot value defining the boundary between the elements has a multiplicity equal to its polynomial order, 2, thereby making the geometrically exact mid-surface a discernible entity within the mesh – it is the boundary between the inner and outer layers of elements. Knots are then inserted into the appropriate knot vectors to define the elements in the mid-surface of the coarsest mesh. This mid-surface mesh is identical for all polynomial orders.

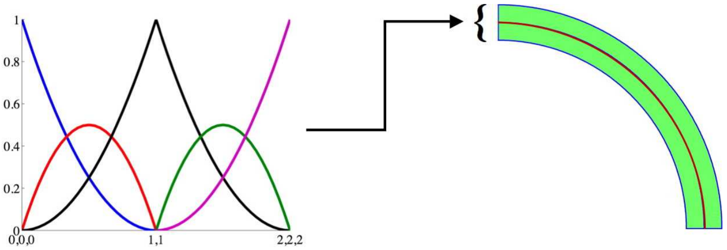


Fig. 27. Two quadratic elements are used through the thickness. The basis is C^0 across the interior element boundary, thus making the boundary itself a NURBS surface. By this construction, the geometrically exact mid-surface is a discernible entity within the mesh.

Once the coarse mid-surface mesh is fixed, so too is the number of basis functions in the axial direction. The offset curves that define the inner and outer surfaces of revolution must now be interpolated. The number of points along each curve that may be interpolated is equal to the number of basis functions in the axial direction. Due to the use of open knot vectors (see [7]), the number of functions for a fixed mesh grows with p. Specifically, for the chosen mesh there are $14 + p$ basis functions in the axial (*i.e.*, y) direction. As a result, $14 + p$ points, equispaced in the parametric domain, are calculated along the hyperbola, then offset by $\pm t/2$ using the analytically computed normals to the curve. These offset points are then interpolated using C^{p-1} B-splines to create the approximate geometry. In this way, the quality of the overall geometric approximation improves as the polynomial or-

[9] Note that it is the offset of the hyperbola which is not represented exactly. In the radial direction, the offset of a circle is again a circle and therefor exists in our NURBS space. It is the *radii* of the circles denoting the inner and outer surfaces of the structure at a given height y that are not exact.

der increases, though the mid-surface mesh is the same for all orders. The loading, however, does not differ as it is applied directly to the mid-surface itself.

To complete Mesh 1 for each polynomial order, knots are inserted near the fixed ends creating two rows of small elements in order to better resolve the boundary layer. The multiplicity of these knots is p, and so the basis functions are C^0 across these element boundaries, shown in red in Figure 28. As discussed above, we could have introduced the *same number* of new degrees-of-freedom into the region by creating many small elements, each having $p - 1$ continuous derivatives across their boundaries. Instead we have chosen fewer elements with lower continuity. The motivation for introducing these C^0 mesh-lines is that previous experience has indicated that doing so helps to prevent the behavior in the layer from polluting results elsewhere in the domain. The main reason for this is that introducing a C^0 mesh-line results in a localizing of the support of the basis functions and thereby decreasing the coupling of functions within the boundary layer with functions outside of the boundary layer (see Figure 29). The result is crisper layers and more compact representations of the global solution, particularly on coarse meshes [10] .

Meshes 2 and 3, seen in Figures 30 and 31, respectively, are the result of subsequent uniform *h*-refinements. The basis is C^{p-1} across the element boundaries introduced through these refinements. The geometry and parameterization remain unchanged, and so Mesh 1 fixes the geometry for each polynomial order. In this way, our analysis is an *h*-method, repeated for several different polynomial orders, rather than a *p*-method repeated for several meshes as in [18]. Recall, however, that the mid-surface meshes for Mesh 1, Mesh 2 and Mesh 3 are independent of the polynomial order. This was done in an effort to make the results from one polynomial order to the next as comparable as possible.

3.2.2 Results

The numbers of degrees-of-freedom and the numerically computed volume are reported in Table 1. The potential energy for each mesh, as well as the limit as *h* $\rightarrow 0$ estimated using Richardson extrapolation (see the appendix), are reported in Table 2. Plots of the deformed geometry as seen from several different angles are shown in Figures 32-34. The displacement has been amplified by a factor of 10 to make it more visible. Due to the sinusoidal character of the loading, the deformed structure has “compression lobes” and “expansion lobes.” In both cases, the largest gradients of the solution are contained in thin layers near the fixed ends of the structure. Plots

[10] Note that we would expect a very fine mesh (*e.g.*, one with all of the elements the size of those in the boundary layer) comprised of highly continuous functions to represent the solution more efficiently than a classical p-method on a per degree-of-freedom basis. On a coarse mesh, however, the efficiency of the method is degraded if a large percentage of the functions have support in both very small and very large elements. This issue is investigated in more detail in Section 3.3.

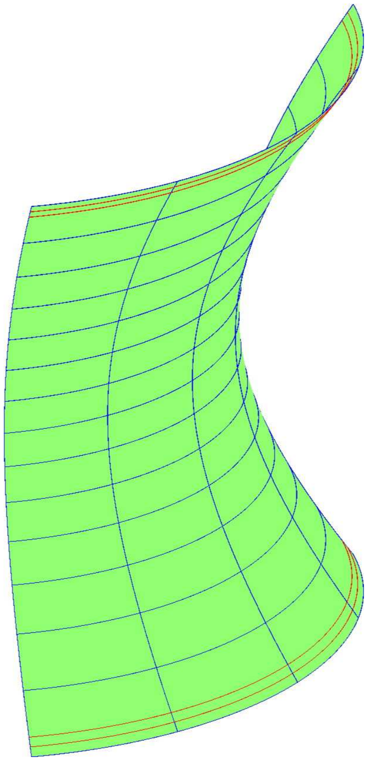


Fig. 28. Mesh 1. Basis functions have $p - 1$ continuous derivatives across blue element boundaries. They are only C^0 across red element boundaries.

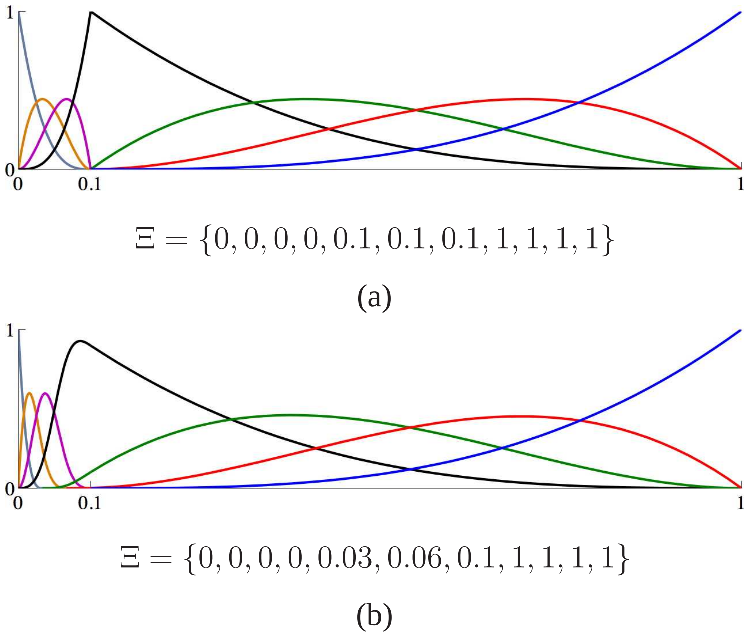
of the radial and vertical displacement at a compression lobe and at an expansion lobe are shown in Figures 35-42. In all of these plots, results for each of the polynomial orders on Mesh 3 are shown. While the quadratics are far from converged, and cubics seem to be showing signs of the geometry error, the quartics and quintics lie practically on top of each other. It is likely that if the study were extended to higher polynomial orders, the curves would be virtually indistinguishable from the quintic solution.

p, q	**Mesh 1 DOF**	**Mesh 2 DOF**	**Mesh 3 DOF**	**Vol. of shell**
2, 2	2160	6300	21060	$1.597535 * 10^{-2}$ m^3
3, 2	3045	7755	23655	$1.597527 * 10^{-2}$ m^3
4, 2	4080	9360	26400	$1.597530 * 10^{-2}$ m^3
5, 2	5265	11115	29295	$1.597530 * 10^{-2}$ m^3

Table 1

Mesh data. Here p denotes the polynomial order in the plane of the surface, while q is the polynomial order through the thickness. The exact volume of the shell is $1.597530 * 10^{-2}$ m^3.

After this initial study was completed, a second study was performed using the maximum continuity possible. The shell geometries were identical to those presented above, as were the meshes outside of the boundary layer region. The width



$\Xi = \{0, 0, 0, 0, 0.1, 0.1, 0.1, 1, 1, 1, 1\}$

(a)

$\Xi = \{0, 0, 0, 0, 0.03, 0.06, 0.1, 1, 1, 1, 1\}$

(b)

Fig. 29. Example boundary layer meshes. Both meshes have the same number of basis functions. a) Only one of the seven basis functions has support both inside the layer $(\xi < 0.1)$ and outside of the layer $(\xi > 0.1)$. b) Three of the seven basis functions have support both inside and outside of the layer.

p, q	**Mesh 1**	**Mesh 2**	**Mesh 3**	**Est. limit**
2, 2	-4.668902	-4.751145	-4.796779	-4.799821
3, 2	-4.783082	-4.794395	-4.801991	-4.802112
4, 2	-4.787948	-4.795994	-4.799878	-4.799893
5, 2	-4.791334	-4.798373	-4.799941	-4.799942
	$* 10^{-2}$ MNm	$* 10^{-2}$ MNm	$* 10^{-2}$ MNm	$* 10^{-2}$ MNm

Table 2
Potential energy. Estimated limit calculated using Richardson extrapolation.

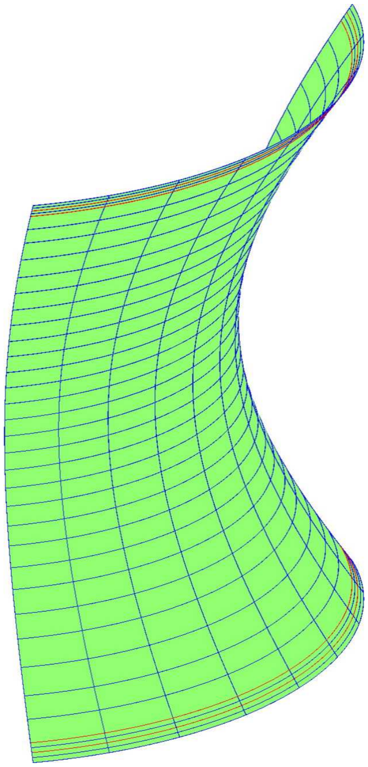


Fig. 30. Mesh 2. The second mesh is generated by uniform *h*-refinement of Mesh 1. The basis is C^{p-1} across the new element boundaries.

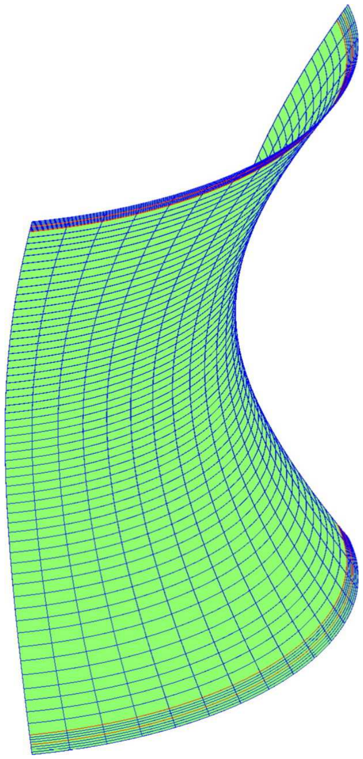


Fig. 31. Mesh 3. The third mesh is generated by uniform *h*-refinement of Mesh 2. The basis is C^{p-1} across the new element boundaries.

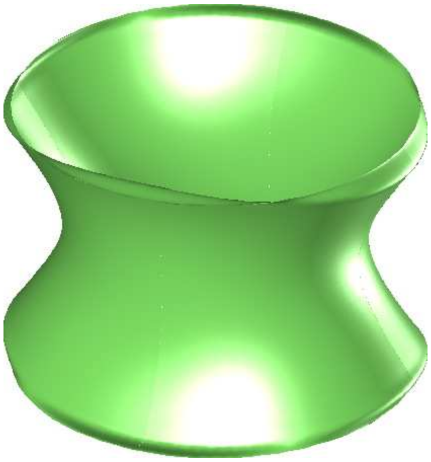


Fig. 32. The deformed configuration. Displacements have been amplified by a factor of 10 for visibility.

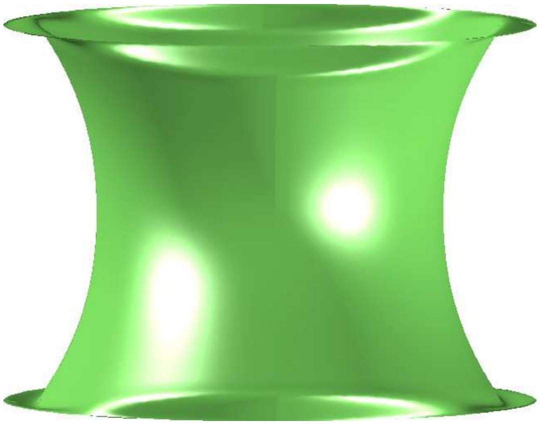


Fig. 33. The deformed configuration. Compression lobes are visible where the loading is directed inward.

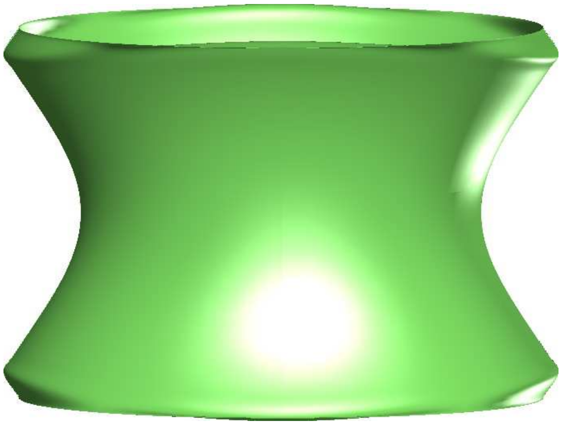


Fig. 34. The deformed configuration. Expansion lobes are visible where the loading is directed outward.

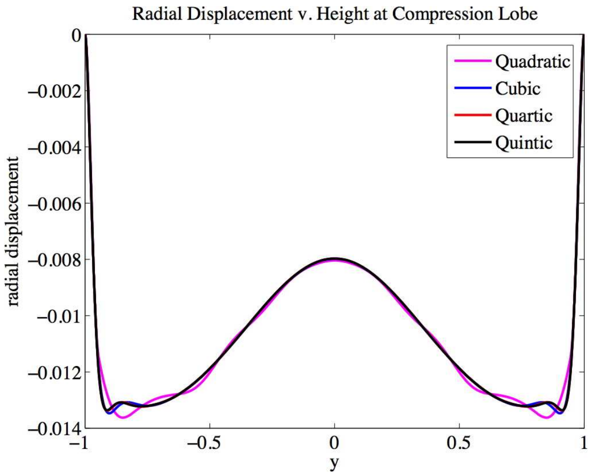


Fig. 35. Compression lobe. Radial displacement versus height.

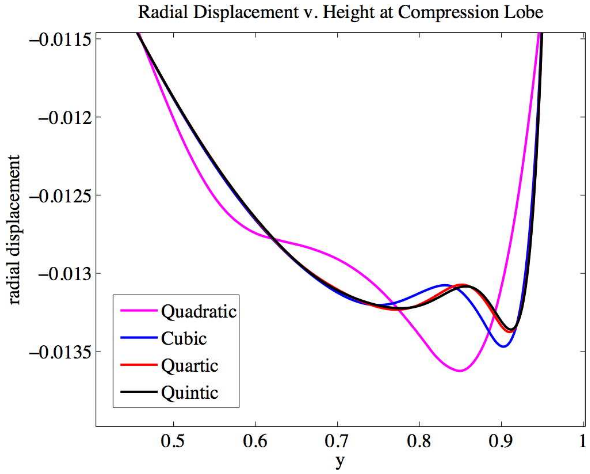


Fig. 36. Compression lobe. Detail of radial displacement versus height.

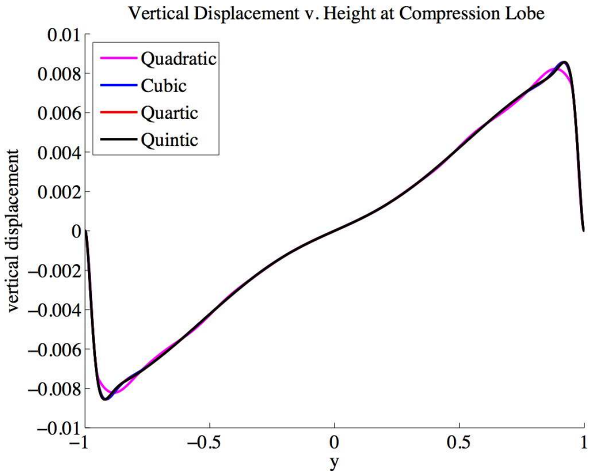


Fig. 37. Compression lobe. Vertical displacement versus height.

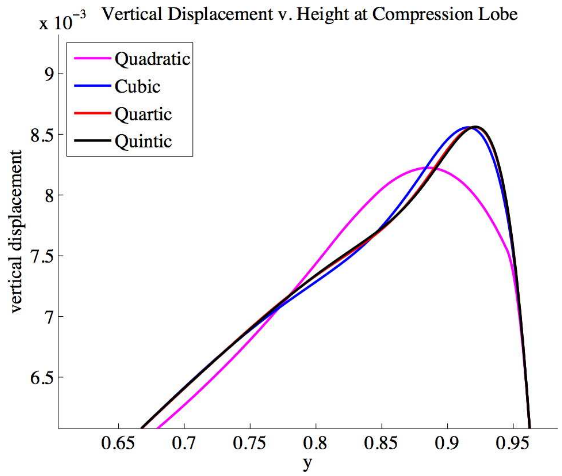


Fig. 38. Compression lobe. Detail of vertical displacement versus height.

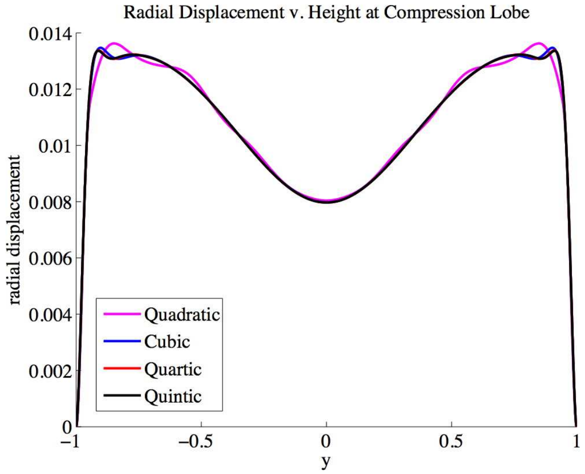


Fig. 39. Expansion lobe. Radial displacement versus height.

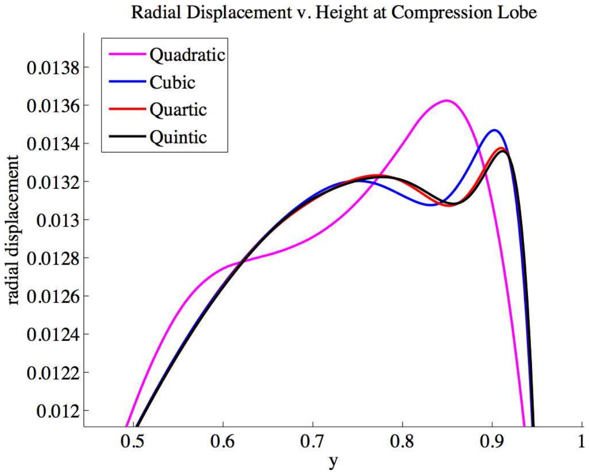


Fig. 40. Expansion lobe. Detail of radial displacement versus height.

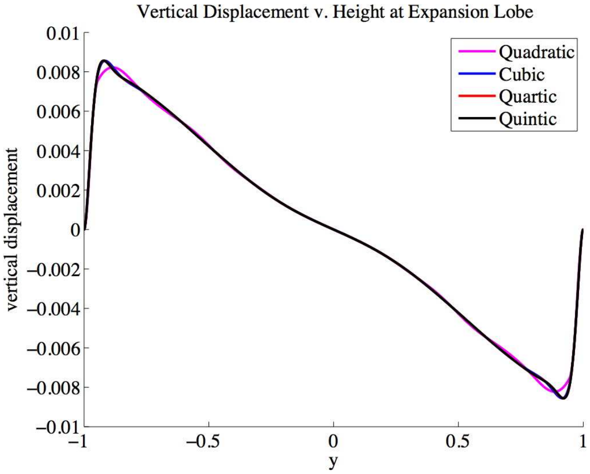


Fig. 41. Expansion lobe. Vertical displacement versus height.

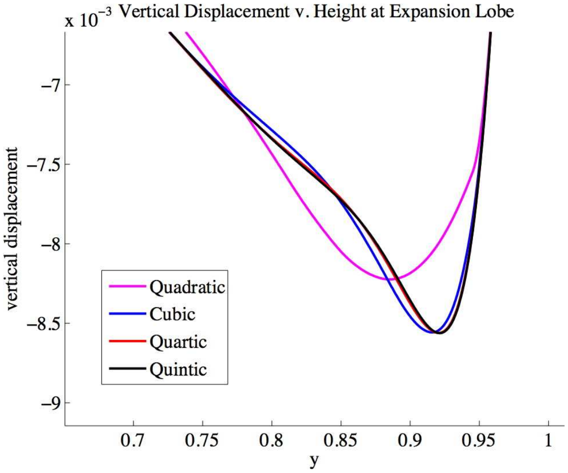


Fig. 42. Expansion lobe. Detail of vertical displacement versus height.

functions may result in a more accurate solution.

	Mesh 1	**Mesh 2**	**Mesh 3**
C^0 Boundary layer	0.1804%	0.0337%	0.0011%
C^{p-1} Boundary layer	0.3379%	0.0206%	0.0007%

Table 3
Comparison of the potential energy errors in the two approaches to boundary layer meshing, $p = 5$. Though the coarse mesh favors the C^0 boundary layer, smooth functions prove to be more accurate once the meshes are sufficiently fine. The reference solution used in the error calculation is the estimated limit for the quintic case from the previous table.

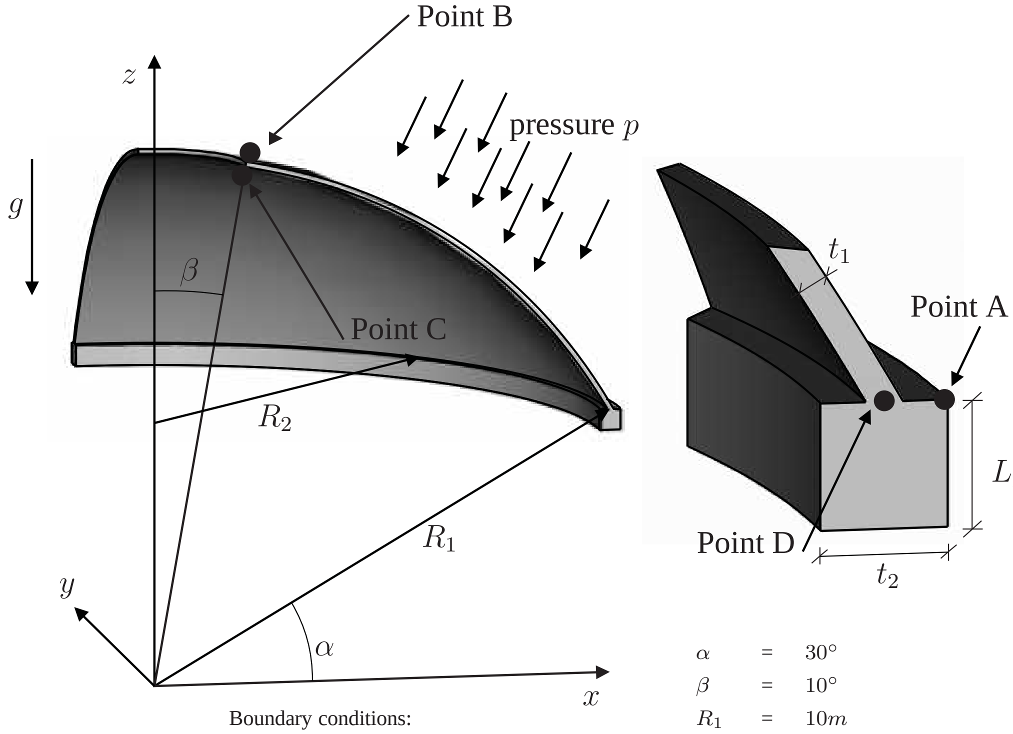
Remark

In preliminary investigations of this problem, a mesh was used that had one cubic element through the thickness, rather than the two quadratic elements as described above. The interpolation, and thus the geometry itself, was identical. The pressure loading was applied to the inner surface of the structure. Though a rigorous study was not performed under these conditions, it was clear that the potential energy was consistently greater (less negative) by about 3% than the limit values reported by Szabo [18]. The meshes used in the present study were generated in an effort to improve upon these early efforts, as indeed they did. Pressurizing the mid-surface instead of the inner surface not only allowed the use of accurate normals (which was seen as the most likely source of error in the preliminary efforts), but also made the study more consistent as the loading is now the same for all polynomial orders.

When comparing with the results of Szabó [18], our value for the potential energy is slightly more negative but the difference is only $\approx 0.1\%$ of the total energy. It is likely that the discrepancy is due to geometrical differences in our models, though the implementations of the loading may have had an effect as well. Despite having the exact geometrical mid-surface, our approach to interpolation was somewhat arbitrary, though the volume calculations imply that our geometries for $p = 4$ and $p = 5$ are more accurate than those in [18].

3.3 Hemispherical shell with a stiffener

The hemispherical shell with a stiffener problem (see Figure 43) was modeled with a single NURBS patch in [7]. As was shown in Section 2.2, use of a single patch leads to substantial distortion of the elements. While it speaks well of the overall robustness of the method that accurate results were still obtained, efficiency clearly suffered. The *p*-method used in that convergence study was not competitive on a per degree-of-freedom basis with the original results of Rank *et al.* [13], who used a trunk space p-refinement strategy. Such an approach does not use the full tensor product space of basis functions, but the much smaller *trunk* space, just large enough to ensure the optimal convergence rate at a given polynomial order (see Szabó, Düster and Rank [19] for a discussion of the trunk space and the *p*-method in general). As NURBS necessarily have an underlying tensor product structure, at least on patches, an analogous isogeometric analysis approach exploiting the trunk space has not been attempted thus far.



Boundary conditions:

At bottom surface of stiffener:
$u_z = 0$

Symmetry at x-z-plane:
$u_y = 0$

Symmetry at y-z-plane:
$u_x = 0$

α	=	$30°$
β	=	$10°$
R_1	=	$10m$
R_2	=	$5\sqrt{3}m$
t_1	=	$0.1m$
t_2	=	$0.4m$
L_2	=	$0.4m$
E	=	$6.825 \cdot 10^7 \frac{kN}{m^2}$
ν	=	0.3
ρ	=	$500 \frac{kg}{m^3}$
g	=	$10.0 \frac{m}{s^2}$
p	=	$100.0 \frac{kN}{m^2}$

Fig. 43. Hemispherical shell with stiffener. Problem description from Rank *et al.* [13].

Despite the tensor product structure of NURBS, k-refinement presents the possibility of improved efficiency. In fact, *k*-refinement, in conjunction with the use of multiple patches to create better quality meshes, and the use of local refinement to avoid placing functions in regions where they are not needed, enables the NURBS based approach to show an accuracy per degree-of-freedom comparable to the results presented by Rank *et al.* in [13]; see Figures 46-49.

In studying the stiffened shell problem with a *k*-refinement approach, certain previously unobserved features begin to emerge. As above, if a given mesh of higher-order and high continuity does not achieve the level of accuracy desired, one can

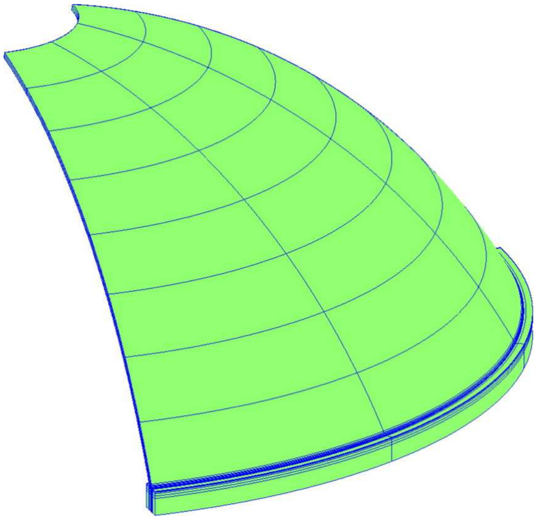
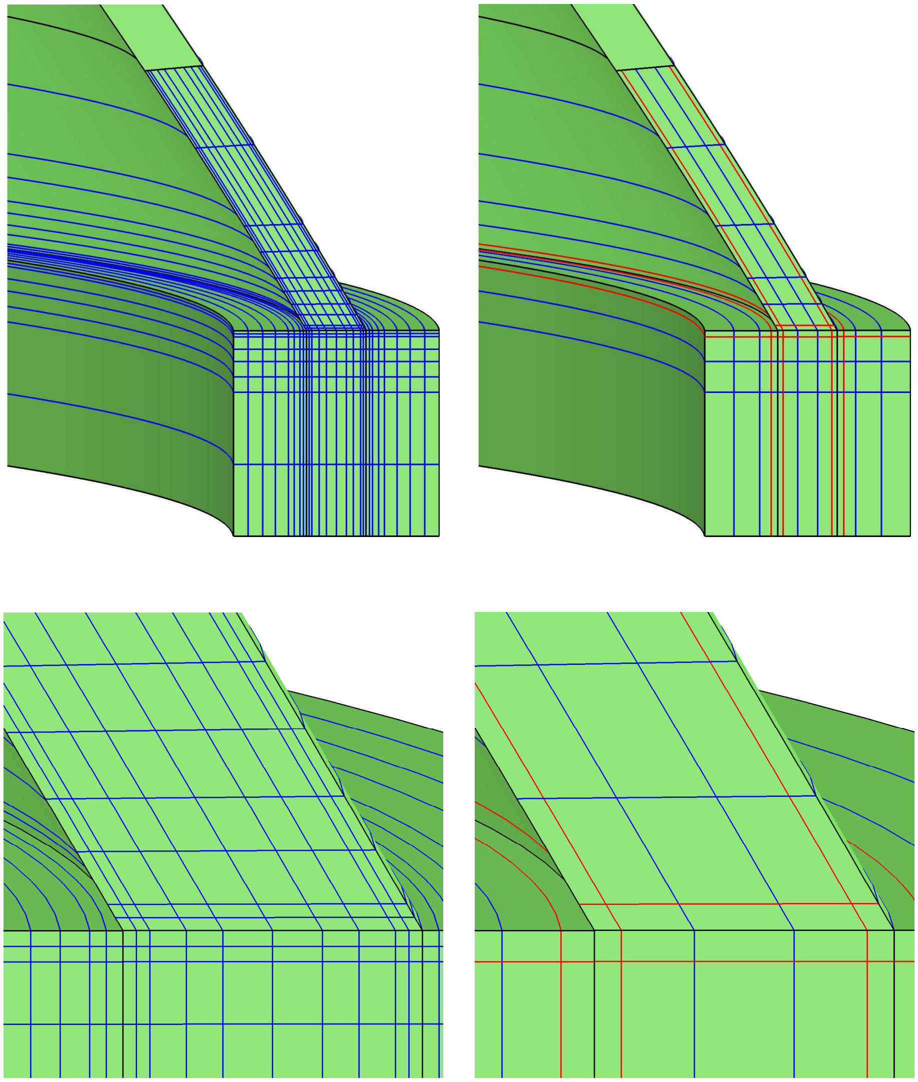


Fig. 44. Hemispherical shell with stiffener. The coarse mesh may be refined in multiple ways.



(a) Local k-refinement

(b) Local k^*-refinement

Fig. 45. Hemispherical shell with stiffener. (a) A k-refinement approach with C^{p-1} continuity across element boundaries. Many small elements are used to get a well-resolved solution. (b) Functions are C^0 across the element boundaries in red, C^{p-1} elsewhere. Fewer elements are needed than in (a). In both cases, the basis is C^0 across patch boundaries, shown in black, and local refinement is implemented at the patch level.

add more degrees-of-freedom by inserting a knot in one of the parametric directions. The number of new degrees-of-freedom is *exactly* the same regardless of whether a new knot value is inserted (creating new elements by splitting existing ones), or whether an existing knot value is repeated (creating no new elements, but decreasing the continuity of the basis across the corresponding element boundaries). While a rigorous analysis of the two approaches has not yet been performed, in the present results it seems clear that in regions where the solution is very smooth (such as in the shell, a reasonable distance away from the stiffener), inserting a new knot, and thus more functions that maintain high continuity, was the more beneficial refinement. In the vicinity of a singularity (such as near the reentrant corner where the shell meets the stiffener and the stress is singular), it is more beneficial to repeat an existing knot value, decreasing the continuity of the basis and simultaneously decreasing the support of the basis functions in the physical space. Both of these effects help localize the singularity and prevent it from polluting the results elsewhere in the domain [11] .

The meshes for the multiple-patch treatment of the stiffened shell are shown in Figures 44 and 45. The locally refined, *k*-method meshes are seen in Figure 45a. In Figure 45b, we see the case where fewer elements are used. A *k*-type refinement is used everywhere except at the knot lines marked in red. The multiplicities of these knots were increased with the polynomial order such that the basis remained C^0 across them. The results for this mesh are labeled "Local *k**-ref" to indicate that the *k*-refinement paradigm was altered near the singularity. The displacements are plotted versus the number of degrees of freedom in Figures 46-49. The calculated von Mises stresses are plotted versus the number of degrees of freedom in Figures 50-53. The trunk space *p*-method results from Rank *et al.* [13] are plotted for comparison. For displacements, the single patch results from [7] are plotted as well.

4 Conclusions

We described the possibility of h-, p-, and k-refinement strategies and explored their behavior on four numerical examples. The first two examples concerned the free vibration of an elastic rod and a thin beam. In these cases the superiority of the k-method over the classical p-method is quite dramatic. On a per degree-of-freedom basis, the results are significantly more accurate across the entire spectrum. In addition, spurious optical branches for the p-method are found to diverge with p, where as optical branches are eliminated for the k-method. In the latter case the entire

[11] This is reminiscent of the heuristic notion that an *hp*-method should use large elements with higher-order in smooth regions and small elements of lower-order near singularities. Coupling this with control over the continuity across elements opens the door to the possibility of an *hpk*-method.

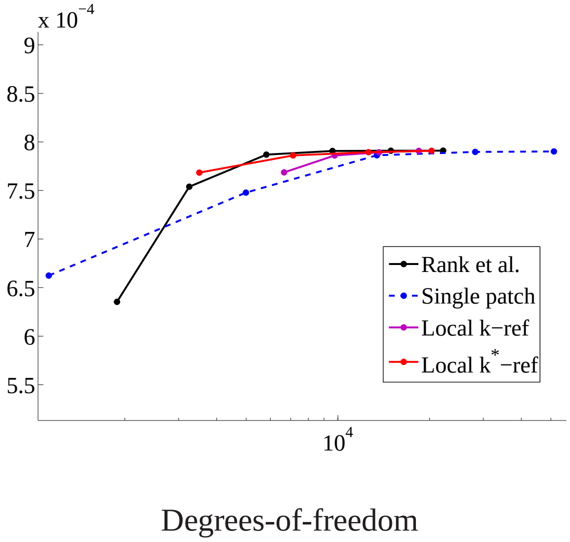


Fig. 46. Hemispherical shell with stiffener. The displacement at point A is plotted versus the total number of degrees-of-freedom.

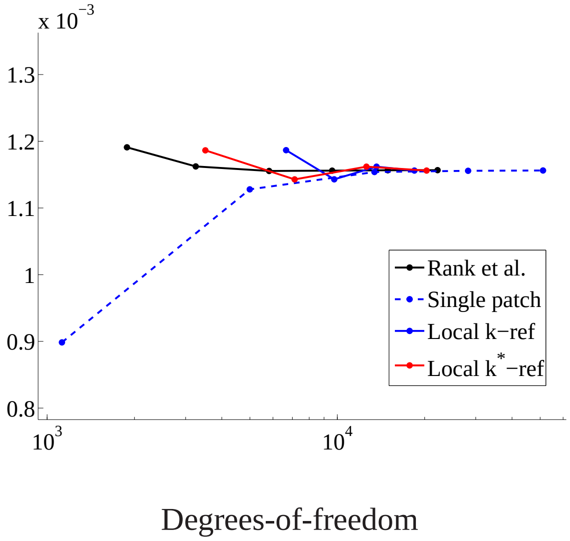


Fig. 47. Hemispherical shell with stiffener. The displacement at point B is plotted versus the total number of degrees-of-freedom.

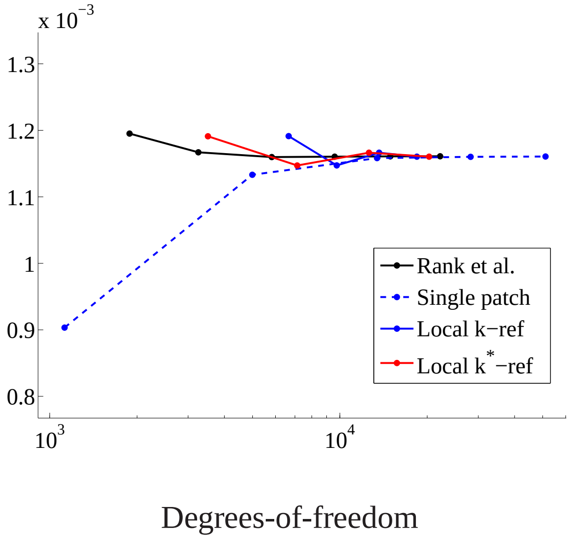


Fig. 48. Hemispherical shell with stiffener. The displacement at point C is plotted versus the total number of degrees-of-freedom.

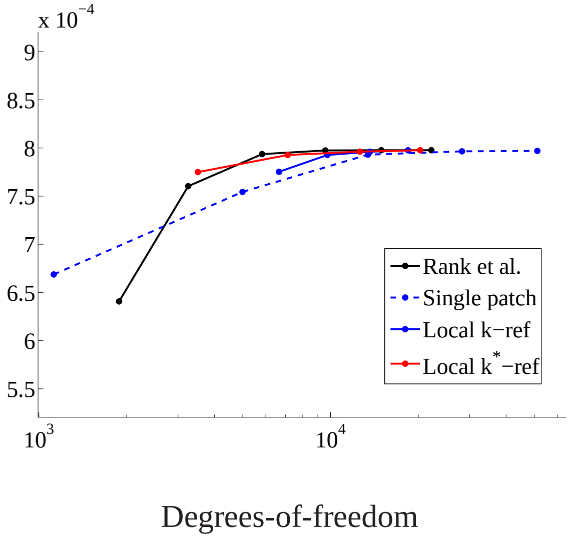


Fig. 49. Hemispherical shell with stiffener. The displacement at point D is plotted versus the total number of degrees-of-freedom.

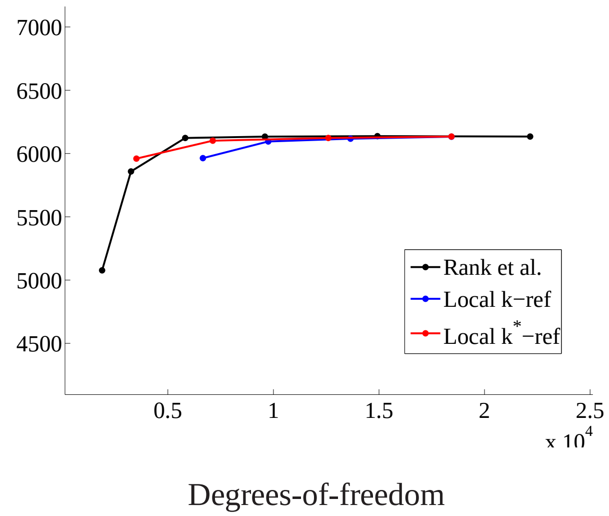


Fig. 50. Hemispherical shell with stiffener. The von Mises stress at point A is plotted versus the total number of degrees-of-freedom.

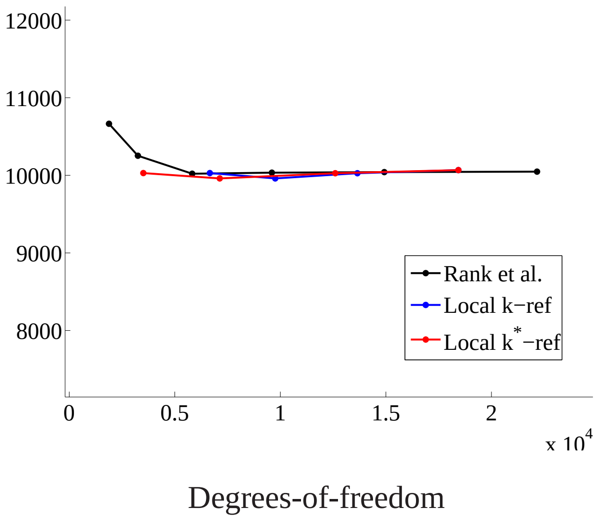


Fig. 51. Hemispherical shell with stiffener. The von Mises stress at point B is plotted versus the total number of degrees-of-freedom.

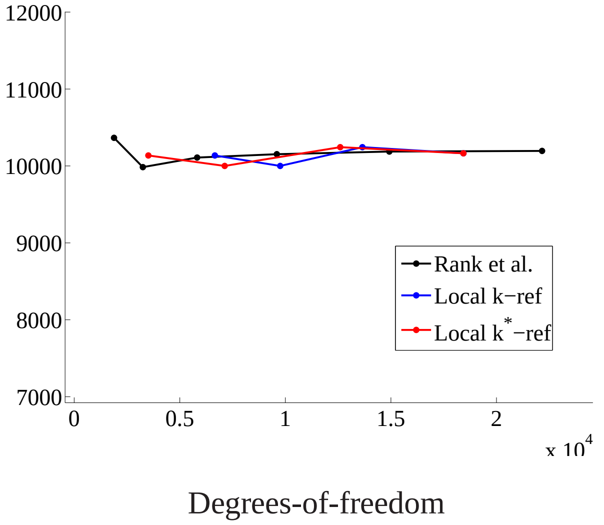


Fig. 52. Hemispherical shell with stiffener. The von Mises stress at point C is plotted versus the total number of degrees-of-freedom.

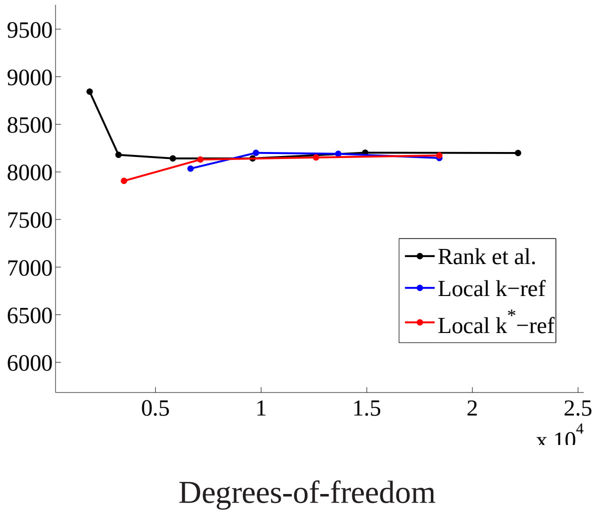


Fig. 53. Hemispherical shell with stiffener. The von Mises stress at point D is plotted versus the total number of degrees-of-freedom.

spectrum converges as order is increased. We feel in these cases that the increased smoothness of the k-method basis functions better exploits the smoothness of the exact analytical eigensolutions than do the C^0-continuous basis functions of the p-method. For further studies of the behavior of isogeometric approaches to structural vibrations, see Cottrell *et al.* [3].

We then considered two elliptic boundary value problems for shell structures exhibiting singular behavior. The first shell was a hyperboloid subjected to circumferentially varying pressure (Lee and Bathe [10]). There is a weak boundary layer singularity. The second shell was a spherical cap, with a hole at the apex, built into a solid ring stiffener (Rank *et al.* [13]). The intersection of the shell and the stiffener creates a strong line singularity. These problems provided the opportunity to study the effects of smoothness of basis functions in the vicinity of singularities. In both cases, we used a trivariate NURBS solid description instead of a shell theory. For the hyperboloidal shell, we reduced smoothness locally in the vicinity of the boundary layer. In the case of coarse meshes, this improved accuracy, where as for finer meshes the pure k-method was more accurate. In the case of the spherical cap, we employed a multipatch approach with local refinement and again compared smooth discretizations within the patches with ones in which continuity was reduced to C^0 in the vicinity of the singularity. We found this latter approach led to more rapid convergence. The reason for this seems to be that basis functions having support in the vicinity of the singularity tend to propagate information away from the singularity. The support of smooth k-method basis functions is greater than the support of the same order p-method functions when there are approximately the same numbers of degrees-of-freedom. As a result, the errors created by the singularities tend to propagate further for the smoother basis functions of the k-method. By judiciously locating a few surfaces of reduced continuity, the “pollution” created by the singularities seemed to be more locally confined.

In conclusion, or studies revealed some cases where increased smoothness dramatically improved accuracy and other cases where some local reduction in smoothness also enhanced accuracy. Historically, the dominance of C^0 finite elements has precluded the opportunity for assessing the potential of increased smoothness. It is clear from the studies described herein, and those in our other recent papers [1, 3, 7], that the potential is very significant. However, it does not seem to be a black and white issue, but rather to depend strongly on the nature of the exact solution. Further studies need to be performed to assess the tradeoffs in a wider variety of problems.

Acknowledgements We thank Barna Szabó, Ernst Rank and Alexander Düster for helpful discussions on the shell problems considered herein. The support of Office of Naval Research under contract N00014-03-0263 is gratefully acknowledged.

Appendix
Richardson Extrapolation

Richardson extrapolation is a technique in which two approximations of a known order of accuracy relative to some parameter are combined to obtain an approximation with a higher order of accuracy. In the case of finite elements, the parameter is the mesh size, h, and the order of accuracy is dictated by the polynomial order of the basis functions [12] . In general, we generate an approximation $A(h)$ to a desired quantity A, where the order of the error is known. That is,

$$A = A(h) + Bh^k + Ch^{k+1} + Dh^{k+2} + ... \tag{A.1}$$

where k is known but $B, C, D,$ etc. are unknown constants. More concisely, we write

$$A = A(h) + Bh^k + O(h^{k+1}). \tag{A.2}$$

We can get a second such equation by generating a second approximation with a different mesh size, for example

$$A = A(h/2) + B(h/2)^k + O(h^{k+1}). \tag{A.3}$$

We can remove the lowest order term in the error by taking 2^k times A.3 and subtracting A.2, which yields

$$(2^k - 1)A = 2^k A(h/2) - A(h) + O(h^{k+1}). \tag{A.4}$$

Simplifying, we arrive at

$$A = \frac{2^k A(h/2) - A(h)}{2^k - 1} + O(h^{k+1}). \tag{A.5}$$

We now have a new approximation

$$\tilde{A} \equiv \frac{2^k A(h/2) - A(h)}{2^k - 1} \tag{A.6}$$

whose order of accuracy is higher than either of the two approximations that generated it.

References

[1] Y. Bazilevs, L. Beirao de Veiga, J.A. Cottrell, T.J.R. Hughes, and G. Sangalli. Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16:1031–1090, 2006.

[2] L. Brillouin. *Wave Propagation in Periodic Structures.* Dover Publications, Inc., Mineola, NY, 1953.

[12] The same is true of NURBS based isogeometric analysis. See [1].

[3] J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195:5257–5296, 2006.

[4] G. Engel, K. Garikipati, T. J. R. Hughes, M. G. Larson, and L. Mazzei. Continuous /discontinuous finite element approximations of fourth-order elliptic problems in structural and continuum mechanics with applications to thin beams and plates, and strain gradient elasticity. *Computer Methods in Applied Mechanics and Engineering*, 191:3669–3750, 2002.

[5] G.E. Farin. *NURBS Curves and Surfaces: from Projective Geometry to Practical Use*. A. K. Peters, Ltd., Natick, MA, 1995.

[6] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, Mineola, NY, 2000.

[7] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.

[8] P. Kagan, A. Fischer, and P. Z. Bar-Yoseph. New B-spline finite element approach for geometrical design and mechanical analysis. *International Journal of Numerical Methods in Engineering*, 41:435–458, 1998.

[9] P. Kagan, A. Fischer, and P. Z. Bar-Yoseph. Mechanically based models: Adaptive refinement for B-spline finite element. *International Journal of Numerical Methods in Engineering*, 57:1145–1175, 2003.

[10] P. S. Lee and K. J. Bathe. Development of MITC isotropic triangular shell finite elements. *Computers and Structures*, 82:945–962, 2004.

[11] D. Natekar, X. F. Zhang, and G. Subbarayan. Constructive solid analysis: a hierarchical, geometry-based meshless analysis procedure for integrated design and analysis. *Computer-Aided Design*, 36(5):473–486, 2004.

[12] L. Piegl and W. Tiller. *The NURBS Book (Monographs in Visual Communication), 2nd ed.* Springer-Verlag, New York, 1997.

[13] E. Rank, A. Düster, V. Nübel, K. Preusch, and O. T. Bruhns. High order finite elements for shells. *Computer Methods in Applied Mechanics and Engineering*, 194 (21-24):2494–2512, 2005.

[14] D. F. Rogers. *An Introduction to NURBS With Historical Perspective*. Academic Press, San Diego, CA, 2001.

[15] D.C. Simkins, A. Dumar, N. Collier, and L.B. Whitenack. Geometry representation, modification and iterative design using RKEM. *Computer Methods in Applied Mechanics and Engineering*, Submitted.

[16] K. S. Surana, A. R. Ahmadi, and J. N. Reddy. The *k*-version finite element method for self-adjoint operators in BVP. *International Journal of Computational Engineering Science*, 3:155–218, 2002.

[17] K. S. Surana, R. K. Maduri, P. W. TenPas, and J. N. Reddy. Elastic wave propagation in laminated composites using the space-time least-squares formulation in *h*, *p*, *k* framework. *Mechanics of Advanced Materials and Structures*, 13:161–196, 2006.

[18] B. Szabó. Private communications, 2004-2006.

[19] B. Szabó, A. Düster, and E. Rank. The p-version of the finite element method.

In E. Stein, R. de Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics, Vol. 1, Fundamentals*, chapter 5. Wiley, 2004.

[20] X. F. Zhang and G. Subbarayan. jNURBS: An object-oriented, symbolic framework for integrated, meshless analysis and optimal design. *Advances in Engineering Software*, 37(5):287–311, 2006.