Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement

T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs

HAL Id: hal-01513346

https://hal.archives-ouvertes.fr/hal-01513346

Submitted on 25 Apr 2017

Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement

T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs

Institute for Computational Engineering and Sciences, The University of Texas at Austin, 201 East 24th Street, 1 University Station C0200, Austin, TX 78712-0027, United States

The concept of isogeometric analysis is proposed. Basis functions generated from NURBS (Non-Uniform Rational B-Splines) are employed to construct an exact geometric model. For purposes of analysis, the basis is refined and/or its order elevated without changing the geometry or its parameterization. Analogues of finite element h- and p-refinement schemes are presented and a new, more efficient, higher-order concept, k-refinement, is introduced. Refinements are easily implemented and exact geometry is maintained at all levels without the necessity of subsequent communication with a CAD (Computer Aided Design) description. In the context of structural mechanics, it is established that the basis functions are complete with respect to affine transformations, meaning that all rigid body motions and constant strain states are exactly represented. Standard patch tests are likewise satisfied. Numerical examples exhibit optimal rates of convergence for linear elasticity problems and convergence to thin elastic shell solutions. A k-refinement strategy is shown to converge toward monotone solutions for advection–diffusion processes with sharp internal and boundary layers, a very surprising result. It is argued that isogeometric analysis is a viable alternative to standard, polynomial-based, finite element analysis and possesses several advantages.

1. Introduction

In this paper we introduce a new method for the analysis of problems governed by partial differential equations such as, for example, solids, structures and fluids. The method has many features in common

with the finite element method and some features in common with meshless methods. However, it is more geometrically based and takes inspiration from Computer Aided Design (CAD). A primary goal is to be geometrically exact no matter how coarse the discretization. Another goal is to simplify mesh refinement by eliminating the need for communication with the CAD geometry once the initial mesh is constructed. Yet another goal is to more tightly weave the mesh generation process within CAD. In this work we introduce ideas in pursuit of these goals.

It is interesting to note that finite element analysis in engineering had its origins in the 1950s and 1960s. Aerospace engineering was the focal point of activity during this time. By the late 1960s the first commercial computer programs (ASKA, NASTRAN, Stardyne, etc.) appeared. Subsequently, the finite element method spread to other engineering and scientific disciplines, and now its use is widespread and many commercial programs are available. Despite the fact that geometry is the underpinning of analysis, CAD, as we know it today, had its origins later, in the 1970s and 1980s. A highly-recommended introductory book, with historical insights, is Rogers [1]. This perhaps explains why the geometric representations in finite element analysis and CAD are so different. Major finite element programs were technically mature long before modern CAD was widely adopted. Presently, CAD is a much bigger industry than analysis. Analysis is usually referred to as Computer Aided Engineering (CAE) in market research. It is difficult to precisely quantify the size of the CAE and CAD industries but current estimates are that CAE is in the \$1–\$2 billion range and CAD is in the \$5–\$10 billion range. The typical situation in engineering practice is that designs are encapsulated in CAD systems and meshes are generated from CAD data. This amounts to adopting a totally different geometric description for analysis and one that is only approximate. In some instances mesh generation can be done automatically but in most circumstances it can be done at best semi-automatically. There are still situations in major industries in which drawings are made of CAD designs and meshes are built from them. It is estimated that about 80% of overall analysis time is devoted to mesh generation in the automotive, aerospace, and ship building industries. In the automotive industry, a mesh for an entire vehicle takes about four months to create. Design changes are made on a daily basis, limiting the utility of analysis in design if new meshes cannot be generated within that time frame. Once a mesh is constructed, refinement requires communication with the CAD system during each refinement iteration. This link is often unavailable, which perhaps explains why adaptive refinement is still primarily an academic endeavor rather than an industrial technology.

The geometric approximation inherent in the mesh can lead to accuracy problems. One example of this is in thin shell analysis, which is notoriously sensitive to geometric imperfections; see Fig. 1. The sensitivity to imperfections is shown in Fig. 1b in which the buckling load of a geometrically perfect cylindrical shell is compared with shells in which geometric imperfections are introduced with magnitudes of 1%, 10%, and 50% of the thickness. As may be seen, there is a very considerable reduction in buckling load with increased imperfection.

Sensitivity to geometry has also been noted in fluid mechanics. Spurious entropy layers about aerodynamic shapes were the bane of compressible Euler solvers in the 1980s and 1990s. The problem and its solution were identified in the thesis of Barth [2]. Piecewise linear approximations of geometry were the root cause. Smooth geometry completely eliminated the entropy layers even when the flow fields were approximated by linear elements on the curved geometry; see Fig. 2. This result explains why methods which employ smooth geometric mappings are widely used in airfoil analysis (see [3]). It is also well known in computational fluid dynamics that good quality boundary layer meshes significantly improve the accuracy of computed wall quantities, such as pressure, friction coefficient, and heat flux; see Fig. 3.

The construction of finite element geometry (i.e., the mesh) is costly, time consuming and creates inaccuracies. It is clear from the smaller size of the CAE industry compared with the CAD industry that the most fruitful direction would be to attempt to change, or replace, finite element analysis with something more CAD-like. This direction was taken in the development of the RASNA program Mechanica, in which exact geometry in conjunction with a p-adaptive finite element procedure was utilized. However, the lack of satisfaction of the isoparametric concept led to theoretical questions which were addressed in later versions
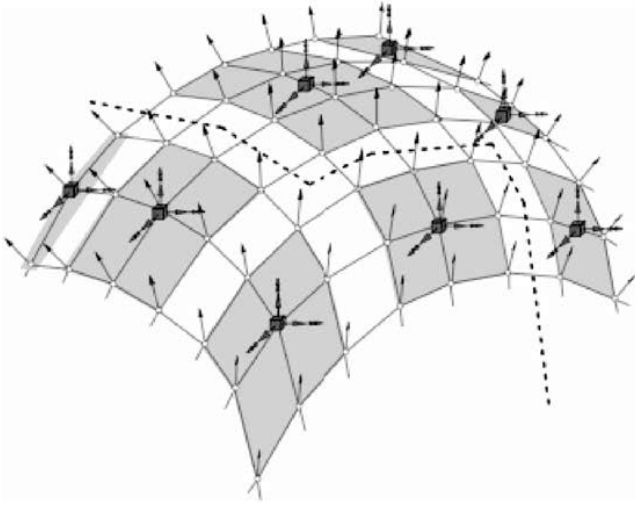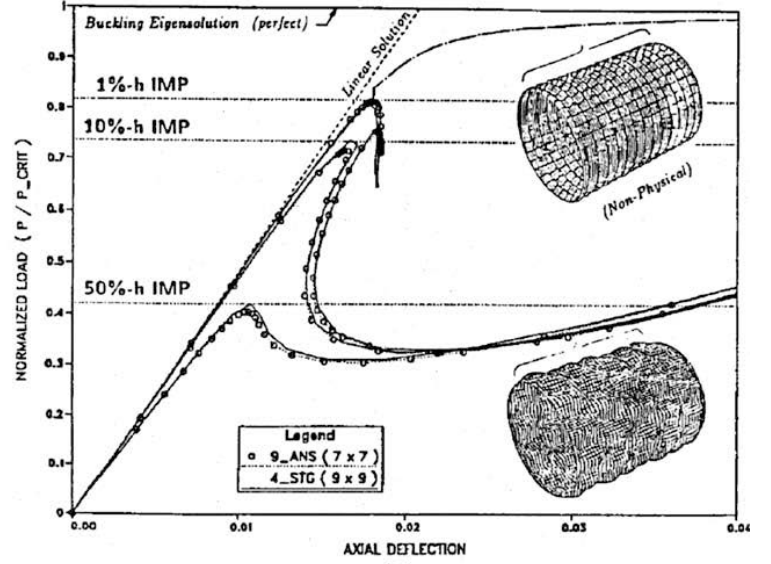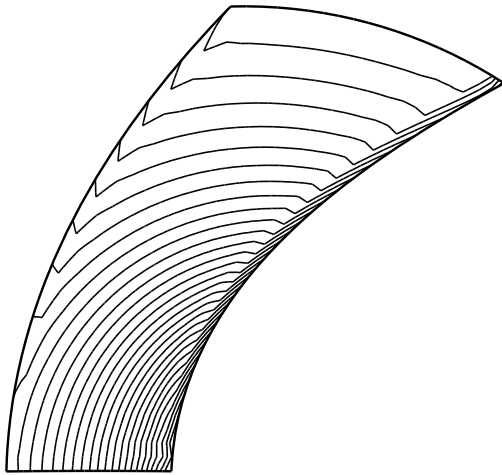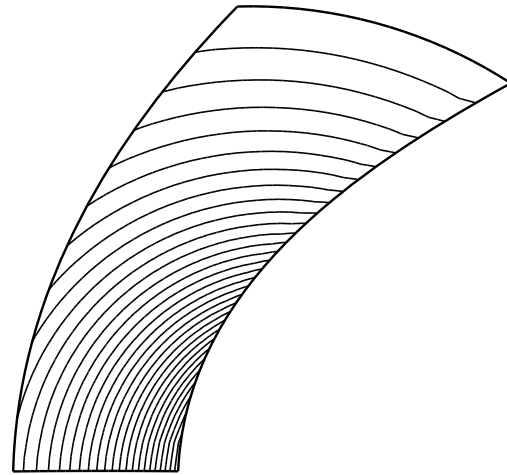
Fig. 1. Thin shell structures exhibit significant imperfection sensitivity: (a) faceted geometry of typical finite element meshes introduces geometric imperfections [4] and (b) buckling of cylindrical shell with random geometric imperfections [5].

Fig. 2. Isodensity contours of GLS discretization of Ringleb flow. (a) Isoparametric linear Lagrange element approximation: both solution space and geometry space are represented by piecewise linear functions. (b) Super-isoparametric element approximation: solution space is piecewise linear, while geometry is piecewise quadratic. Smooth geometry avoids spurious entropy layers associated with piecewise-linear geometric approximations (from [6]).

of the code by abandoning the exact geometry in favor of high-order polynomial approximations [7]. The use of a fixed polynomial approximation to geometry has been shown by Szabó et al. [8] to be limiting. As solution polynomial order is increased, the error plateaus at some level and cannot be further reduced (see Fig. 4). The seriousness of this result is compounded by the fact that computed quantities defined on boundaries are usually the most important ones in engineering applications, and this is where geometric errors are most harmful. Furthermore, most finite element analyses are still performed with low-order elements for which geometric errors are largest. The success of RASNA, which was later acquired by Parametric Technology Corporation (PTC), a CAD company, was due to its tight linkage with CAD
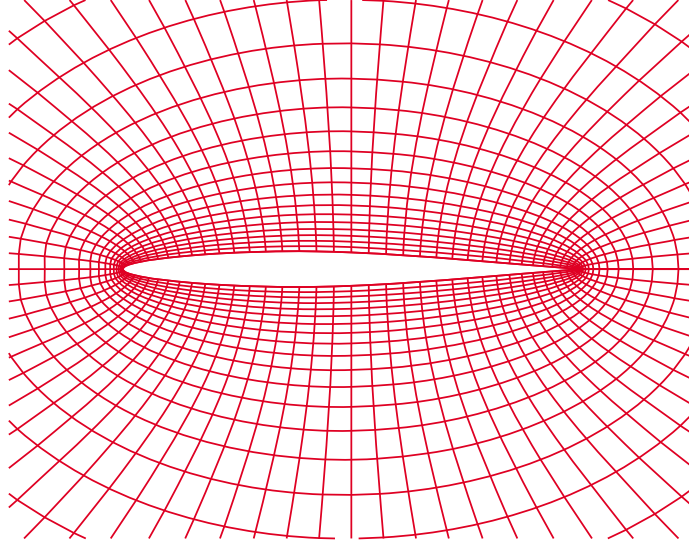
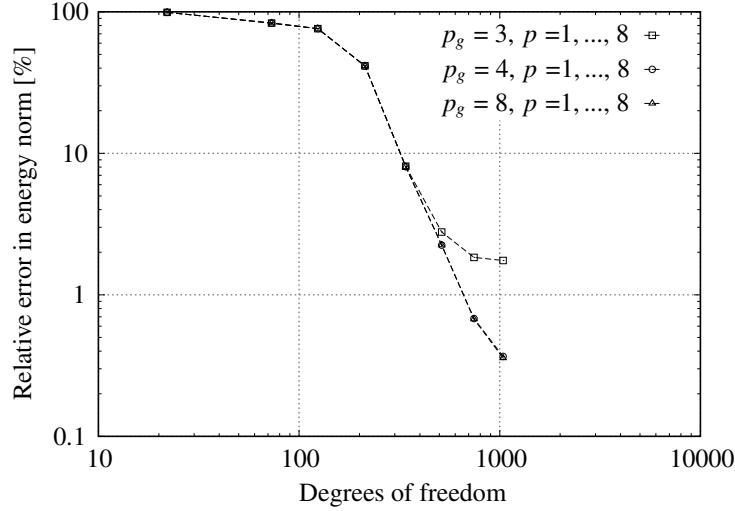Fig. 3. Quality boundary layer meshes significantly improve accuracy (from [3]).

Fig. 4. Convergence study of the Scordelis–Lo roof problem: p_g represents the polynomial degree of geometry representation, p corresponds to the polynomial degree of the approximation space (from [10]).

geometry and, perhaps more importantly, its consequent ability to provide adaptive p-refinement and thus more reliable results. The present methodology is similarly inspired, but attempts to more faithfully adhere to CAD geometry and eliminate the finite element polynomial description entirely. (The p-method is described in Szabó and Babuška [9] and Szabó et al. [8].)

The approach we have developed is based on NURBS (Non-Uniform Rational B-Splines), a standard technology employed in CAD systems. We propose to match the exact CAD geometry by NURBS surfaces, then construct a coarse mesh of "NURBS elements". These would be solid elements in three-dimensions that exactly represent the geometry. [1] This is obviously not a trivial task and one that deserves much study but when it can be accomplished it opens a door to powerful applications. Subsequent refinement does not require any further communication with the CAD system and is so simple that it may facilitate more widespread adoption of this technology in industry. There are analogues of h-, p-, and hp-refinement

---

[1] We assume that unnecessary features, such as the rivets on an airplane wing, are removed from the geometry prior to mesh generation for analysis. Feature removal is a necessary but complex process.

strategies, and a new, higher-order methodology emerges, *k-refinement*, which seems to have advantages of efficiency and robustness over traditional *p*-refinement. All subsequent meshes retain exact geometry. Throughout, the isoparametric philosophy is invoked, that is, the solution space for dependent variables is represented in terms of the same functions which represent the geometry. For this reason, we have dubbed the methodology *isogeometric analysis*.

NURBS are not a requisite ingredient in isogeometric analysis. We might envision developing isogeometric procedures based on "A-patches" (see [11–16]) or "subdivision surfaces" (see [17–19]). However, NURBS seem to be the most thoroughly developed CAD technology and the one in most widespread use.

The body of this paper begins with a tutorial on B-splines (B-splines are the progenitors of NURBS), followed by one on NURBS. We then describe an analysis framework based on NURBS. This is followed by sample applications in linear solid and structural mechanics and some introductory calculations in fluids, namely, ones involving classical test cases for the advection–diffusion equation. Various refinement strategies are studied and, in cases for which exact elasticity solutions are available, optimal rates of convergence are attained. The structural problems include some applications to thin shells modeled as solids. The approach is seen to handle these situations remarkably well. In the fluid calculations, we employ the SUPG formulation and consider difficult test cases involving internal and boundary layers. We observe that, by employing high-order, *k*-refinement strategies, convergence toward monotone solutions is obtained. This surprising result seems to contradict numerical analysis intuitions and suggests the possibility of linear difference methods that are simultaneously robust and highly accurate. We close with conclusions and suggestions for future work.

2. B-splines and NURBS

2.1. Knot vectors

NURBS are built from B-splines. 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. A *knot vector* in one dimension is a set of coordinates in the parametric space, written $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$, where $\xi_i \in \mathbb{R}$ is the *i*th it knot, *i* is the knot index, $i = 1, 2, \ldots, n + p + 1$, *p* is the polynomial order, and *n* is the number of basis functions which comprise the B-spline.

Remark. The convention we will adopt is that the order $p = 0, 1, 2, 3$, etc., refers to constant, linear, quadratic, cubic, etc., piecewise polynomials, respectively. This is the usual terminology in the finite element literature. What we refer to as "order" is usually referred to as "degree" in the computational geometry literature.

If knots are equally-spaced in the parametric space, they are said to be *uniform*. If they are unequally spaced, they are *non-uniform*. More than one knot can be located at the same coordinate in the parametric space. These are referred to as *repeated* knots. 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 at the corners of patches in multiple dimensions but they are not, in general, interpolatory at interior knots. This is a distinguishing feature between knots and "nodes" in finite element analysis.

2.2. Basis functions

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

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leqslant \xi < \xi_{i+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$
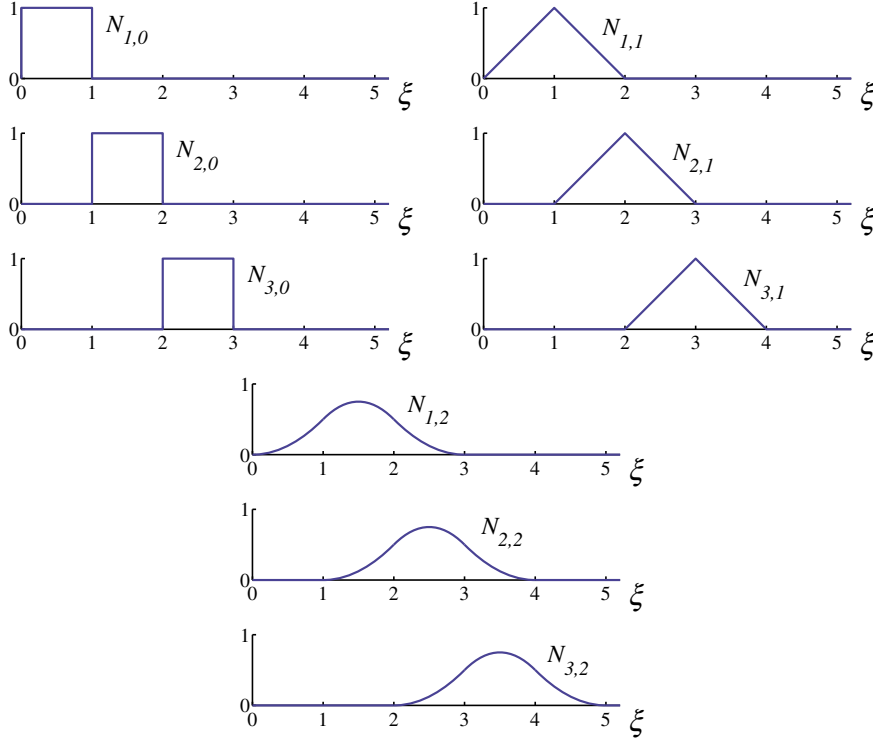
Fig. 5. Basis functions of order $0, 1, 2$ for uniform knot vector $\Xi = \{0, 1, 2, 3, 4, \ldots\}$.

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

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

Derivatives with respect to spatial coordinates may be computed by way of standard techniques described in Hughes [20, Chapter 3]. An initial example of the results of applying (1) and (2) to a uniform knot vector is presented in Fig. 5. Note that, for $p = 0$ and 1, the basis functions are the same as for standard piecewise constant and linear finite element functions, respectively. However, for $p \geqslant 2$, they are different. Quadratic B-spline basis functions (and NURBS basis functions, as will be shown later) are identical but shifted. This is in contrast with quadratic finite element functions which are different for internal and end nodes. This "homogeneous" pattern continues as we go to higher-order B-splines and may result in significant advantages in equation solving over finite element functions, which are quite "heterogeneous".

An example of quadratic basis functions for an open, non-uniform knot vector is presented in Fig. 6. 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 $p - 1$ continuous derivatives. If a knot is repeated k times, then the number of continuous derivatives decreases by k. When the multiplicity of a knot is exactly p, the basis function is interpolatory.
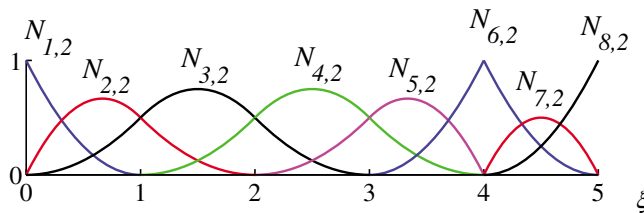

Fig. 6. Quadratic basis functions for open, non-uniform knot vector $\Xi = \{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$.

Important properties of B-spline basis functions are:

(1) They constitute a partition of unity, that is, $\forall \xi$

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

(2) The support of each $N_{i,p}$ is compact and contained in the interval $[\xi_i, \xi_{i+p+1}]$.
(3) Each basis function is non-negative, that is, $N_{i,p}(\xi) \geqslant 0, \forall \xi$. Consequently, *all* of the coefficients of a mass matrix computed from a B-spline basis are greater than, or equal to, zero.

2.3. B-spline curves

B-spline curves in $\mathbb{R}^d$ are constructed by taking a linear combination of B-spline basis functions. The coefficients of the basis functions are referred to as *control points*. These are somewhat analogous to nodal coordinates in finite element analysis. Piecewise linear interpolation of the control points gives the so-called *control polygon*. In general, control points are not interpolated by B-spline curves. Given n basis functions, $N_{i,p}$, $i = 1, 2, \ldots, n$, and corresponding control points $B_i \in \mathbb{R}^d$, $i = 1, 2, \ldots, n$, a piecewise-polynomial *B-spline curve* is given by

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

An example is shown in Fig. 7 for the quadratic basis functions considered previously. Note that the curve is interpolatory at the first and last control points, due to the fact that the knot vector is open, and also at the sixth control point, due to the fact that the multiplicity of the knot $\xi = 4$ 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.
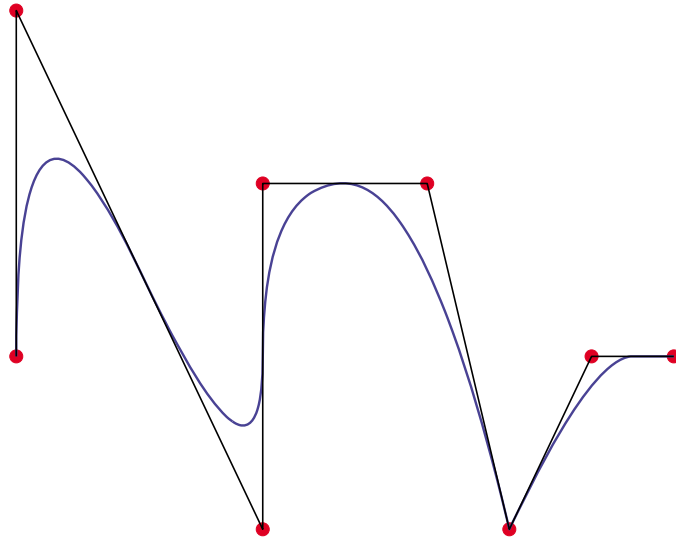
Fig. 7. B-spline, piecewise quadratic curve in $\mathbb{R}^2$. Control point locations are denoted by •. Basis functions and knot vector as in Fig. 6.

Important properties of B-spline curves are:

(1) They have continuous derivatives of order $p-1$ in the absence of repeated knots or control points.
(2) Repeating a knot or control point k times decreases the number of continuous derivatives by k.
(3) An affine transformation of a B-spline curve is obtained by applying the transformation to the control points. [2] We refer to this property as *affine covariance*.

2.4. h-refinement: knot insertion

The analogue of *h*-refinement is *knot insertion*. Knots may be inserted without changing a curve geometrically or parametrically. Given a knot vector $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$, let $\bar{\bar{\xi}} \in [\xi_k, \xi_{k+1}[$ be a desired new knot. The new $n+1$ basis functions are formed recursively, using (1) and (2), with the new knot vector $\Xi = \{\xi_1, \xi_2, \ldots, \xi_k, \bar{\bar{\xi}}, \xi_{k+1}, \ldots, \xi_{n+p+1}\}$. The new $n+1$ control points, $\{\overline{B}_1, \overline{B}_2, \ldots, \overline{B}_{n+1}\}$, are formed from the original control points, $\{B_1, B_2, \ldots, B_n\}$, by

$$\overline{B}_i = \alpha_i B_i + (1 - \alpha_i)B_{i-1}, \tag{5}$$

where

$$\alpha_i = \begin{cases} 1, & 1 \leqslant i \leqslant k - p, \\ \dfrac{\bar{\bar{\xi}} - \xi_i}{\xi_{i+p} - \xi_i}, & k - p + 1 \leqslant i \leqslant k, \\ 0, & k + 1 \leqslant i \leqslant n + p + 2. \end{cases} \tag{6}$$

Knot values already present in the knot vector may be repeated as above but as described in Section 2.2, the continuity of the basis will be reduced. Continuity of the curve is preserved by choosing the control points as in (5) and (6). Each unique internal knot value may appear no more than p times or the curve becomes discontinuous.

An example of knot insertion is presented in Fig. 8. The original curve consists of quadratic B-splines. The knot vector is $\Xi = \{0, 0, 0, 1, 1, 1\}$. The curve is shown on the left with basis functions below. A new knot is inserted at $\bar{\bar{\xi}} = 0.5$. The new curve, shown on the right, is geometrically and parametrically identical to the original curve, but the basis functions, below the curve, and control points are changed. There is one more of each. This process may be repeated to enrich the solution space by adding more basis functions of the same order while leaving the curve unchanged. This subdivision strategy is seen to be analogous to the classical *h*-refinement strategy in finite element analysis.

2.5. p-refinement: order elevation

The polynomial order of basis functions may be increased without changing the geometry or parameterization. It is important to note that each unique knot value in Ξ must be repeated to preserve discontinuities in the *p*th derivative of the curve being elevated. The number of new control points depends on the multiplicities of existing knots. This strategy of *order elevation* is an analogue of *p*-refinement in finite element analysis.

As is the case of *h*-refinement by way of knot insertion, the solution space spanned by the order elevated basis functions contains the space spanned by the original functions. Thus, it is possible to order elevate without changing the geometry of the B-spline curve. Less obviously, it can be done so as to leave

---

[2] This turns out to be the essential property for satisfying so-called "patch tests". More about this later.

Original curve with $\Xi = \{0,0,0,1,1,1\}$ Refined curve with $\Xi = \{0,0,0,0,0.5,1,1,1\}$
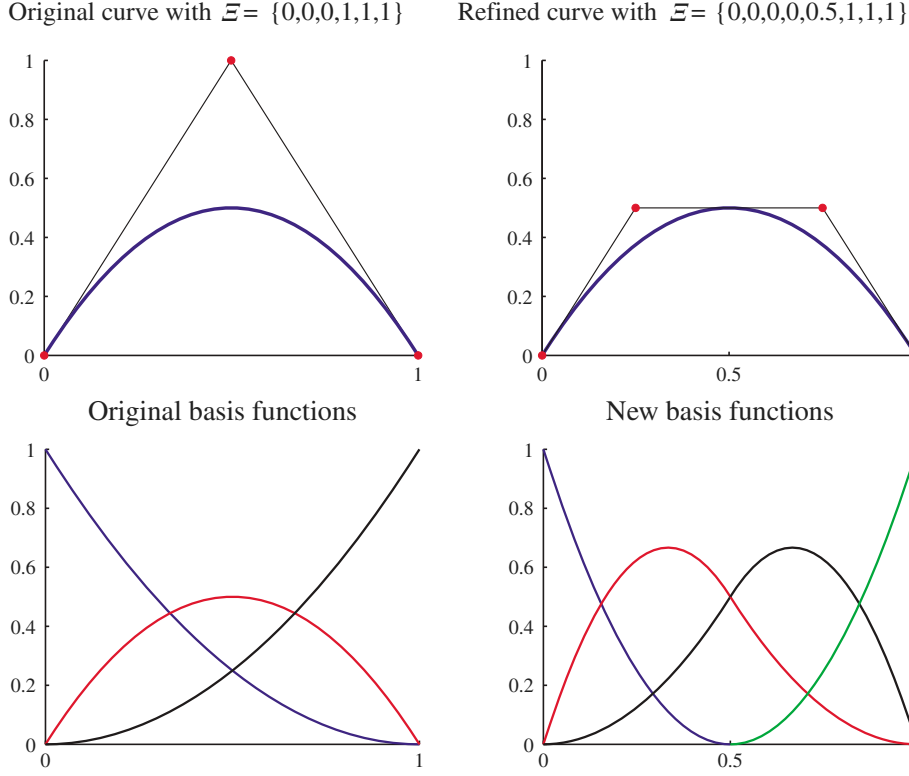

Fig. 8. Knot insertion. Control points are denoted by •.

the parameterization of the curve in tact. The process for doing this involves subdividing the curve into many Bézier curves by knot insertion (see [1] or [21] for a discussion of Bézier curves), order elevating each of these individual segments, and then removing the unnecessary knots 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. We omit the details for the sake of brevity. For a thorough treatment, see Piegl and Tiller [21]. (Note that the CAD community refers to this process as "degree elevation".)

An example of order elevation is depicted in Fig. 9. The original curve and quadratic basis functions, shown on the left, are the same as considered in the previous example. This time the multiplicity of the knots is increased by one. The numbers of control points and basis functions each increase by one. The locations of the control points change, but the elevated curve is geometrically and parametrically identical to the original curve. There are now four cubic basis functions. The locations of control points for this elevated curve are different than those in the previous example (cf. Fig. 8).

2.6. k-refinement

An alternative order elevation strategy takes advantage of the fact that the processes of knot insertion and order elevation do not commute. If a unique knot value, $\bar{\bar{\xi}}$, is inserted between two distinct knots in a curve of order p, the number of continuous derivatives of the basis functions at $\bar{\bar{\xi}}$ is $p - 1$. As described above, if we subsequently elevate the order to q, the multiplicity of every distinct knot value (including the knot just inserted) is increased so that discontinuities in the pth derivative of the basis are preserved, that is, the basis still has $p - 1$ continuous derivatives at $\bar{\bar{\xi}}$. If instead we elevated the order of the original curve to q and only then inserted a unique knot value, the basis would have $q - 1$ continuous derivatives at $\bar{\bar{\xi}}$. We refer to this latter procedure as *k-refinement*. It has no analogue in standard finite element analysis.
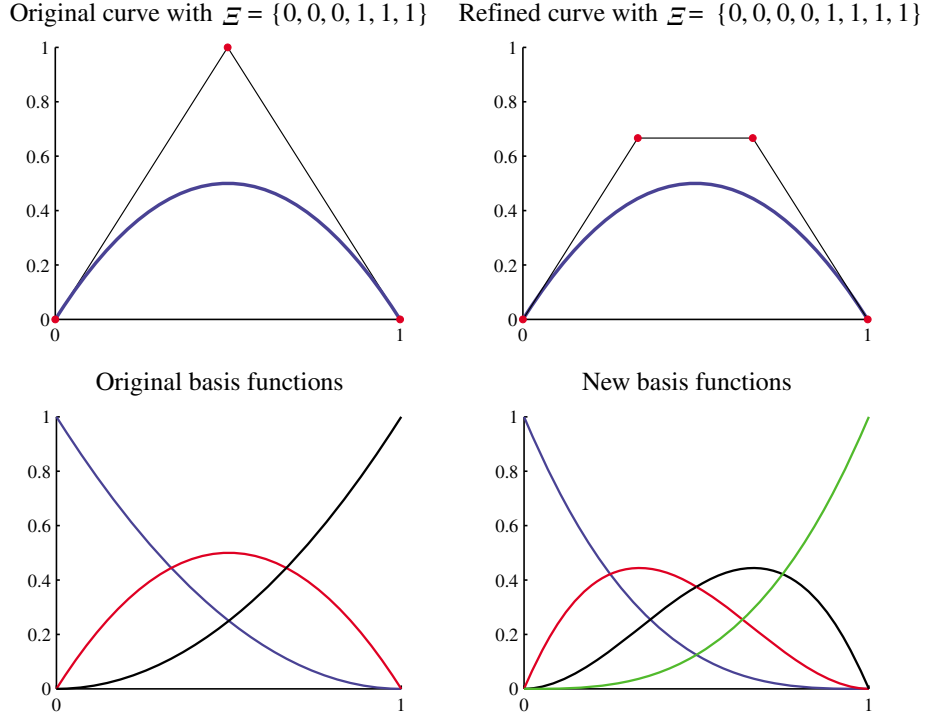
9

Fig. 9. Order elevation. Control points are denoted by •.

We believe the concept of k-refinement is important and potentially a superior approach to high-precision analysis than p-refinement. 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. This will be made clear with an example but first we need a definition. Let us define an "element" in one dimension 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). This definition anticipates our NURBS-based finite element implementation, described in Section 3.

With this notion of an element, consider the refinement processes depicted in Fig. 10. Assume the initial domain consists of one element and $p + 1$ basis functions (assuming an open knot vector), which we then h-refine until we have $n - p$ elements and n basis functions. We then perform order elevation by p-refinement, 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, begin 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 h-refine 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 $(r + 1)n - rp$. Keep in mind too that in d dimensions these numbers are raised to the d power. There may also be other advantages in that smoother derivatives may lead to more accurate physical quantities, such as strains and stresses, and increased smoothness may, surprisingly, lead to better capturing of thin layers. An example of this phenomenon is presented later on. Of course, if the physical situation dictates a certain lower level of continuity at a knot value
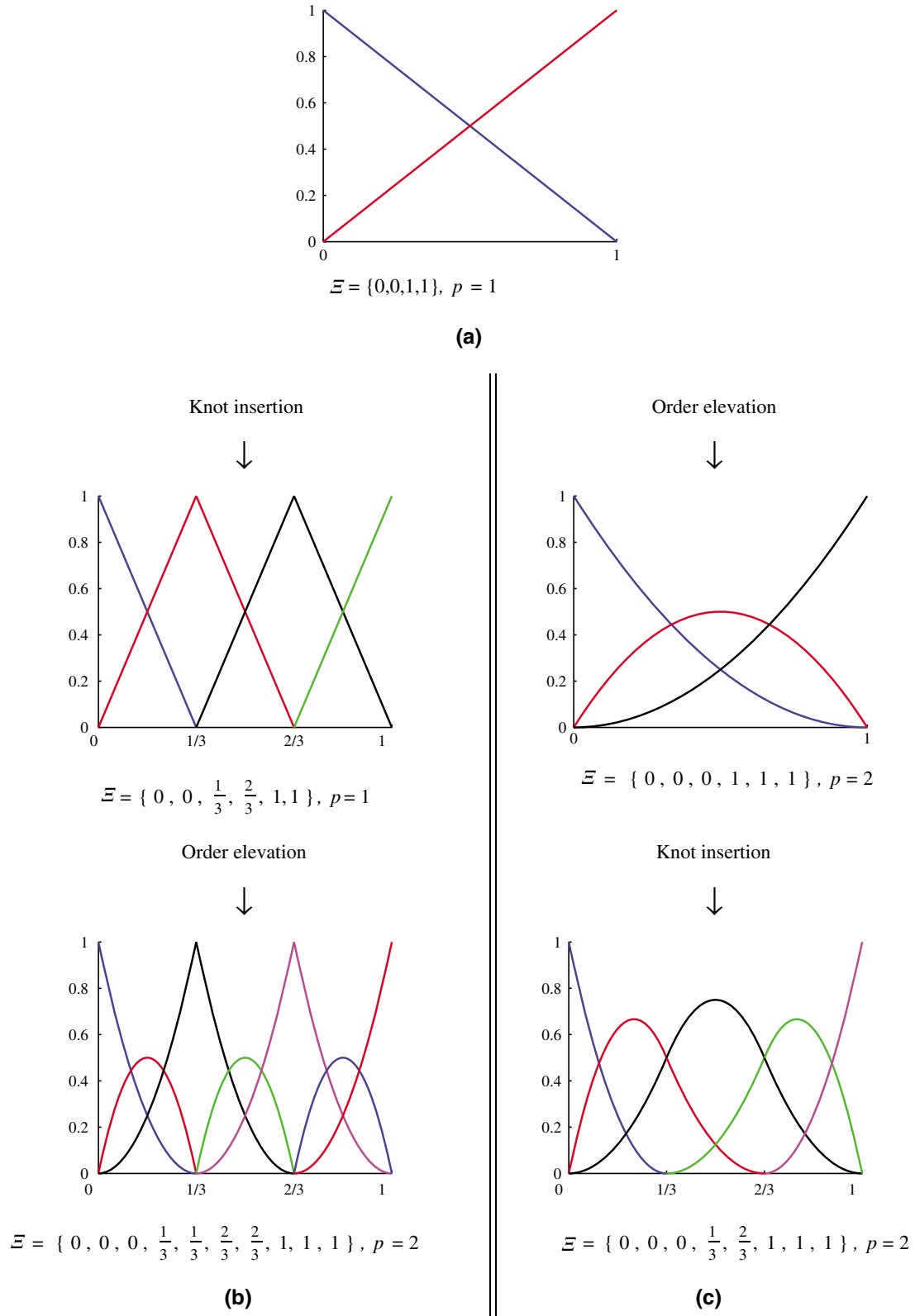
Fig. 10. *k*-refinement takes advantage of the fact that knot insertion and order elevation do not commute. (a) Base case of one linear element. (b) Classic *p*-refinement approach: knot insertion followed by order elevation results in seven piecewise quadratic basis functions that are C^0 at internal knots. (c) New *k*-refinement approach: order elevation followed by knot insertion results in five piecewise quadratic basis functions that are C^1 at internal knots.
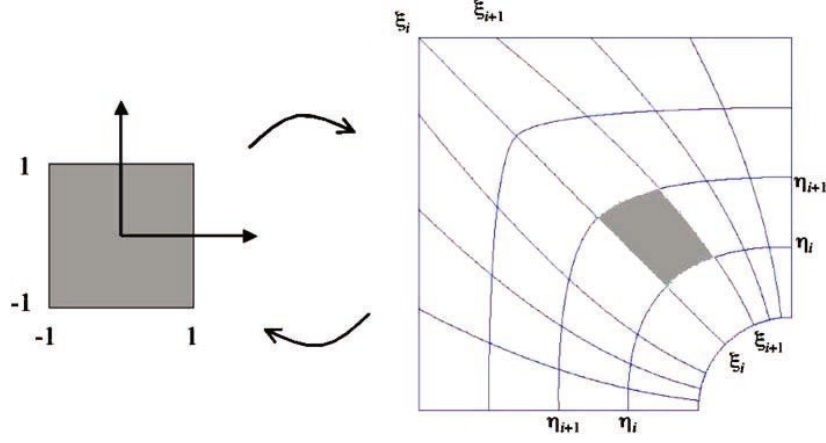
11

Fig. 11. Elements as knot spans.

(e.g., the corners in the geometry, discontinuous material properties, etc.) this can always be incorporated into the process by knot duplication, so no generality is lost.

2.7. B-spline surfaces

Given a *control net* $\{B_{i,j}\}$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$, and knot vectors $\varXi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$, and $\mathscr{H} = \{\eta_1, \eta_2, \ldots, \eta_{m+q+1}\}$, a tensor product B-spline surface is defined by

$$S(\xi, \eta) = \sum_{i=1}^{n} \sum_{j=1}^{m} N_{i,p}(\xi) M_{j,q}(\eta) B_{i,j}, \tag{7}$$

where $N_{i,p}$ and $M_{j,q}$ are basis functions of B-spline curves. For purposes of numerically integrating arrays constructed from B-splines, "elements" are taken to be knot spans, namely, $[\xi_i, \xi_{i+1}] \times [\eta_j, \eta_{j+1}]$. See Fig. 11 for an illustration of a standard bi-unit parent element and its image in physical space. Integrals are pulled back to the parent element by the classical change-of-variables formula and standard Gaussian quadrature rules are employed; see [20, Chapter 3].

2.8. B-spline solids

Tensor product B-spline solids are defined in analogous fashion to B-spline surfaces. Given a control net $\{B_{i,j,k}\}$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$, $k = 1, 2, \ldots, l$, and knot vectors $\varXi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$, $\mathscr{H} = \{\eta_1, \eta_2, \ldots, \eta_{m+q+1}\}$, and $\mathscr{Z} = \{\zeta_1, \zeta_2, \ldots, \zeta_{l+r+1}\}$, a B-spline solid is defined by

$$S(\xi, \eta, \zeta) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} N_{i,p}(\xi) M_{j,q}(\eta) L_{k,r}(\zeta) B_{i,j,k}. \tag{8}$$

2.9. Rational B-splines

Desired geometric 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 quadratic curves. This is illustrated in Fig. 12 in which a circle in $\mathbb{R}^2$ is constructed from a piecewise quadratic B-spline curve in $\mathbb{R}^3$. 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
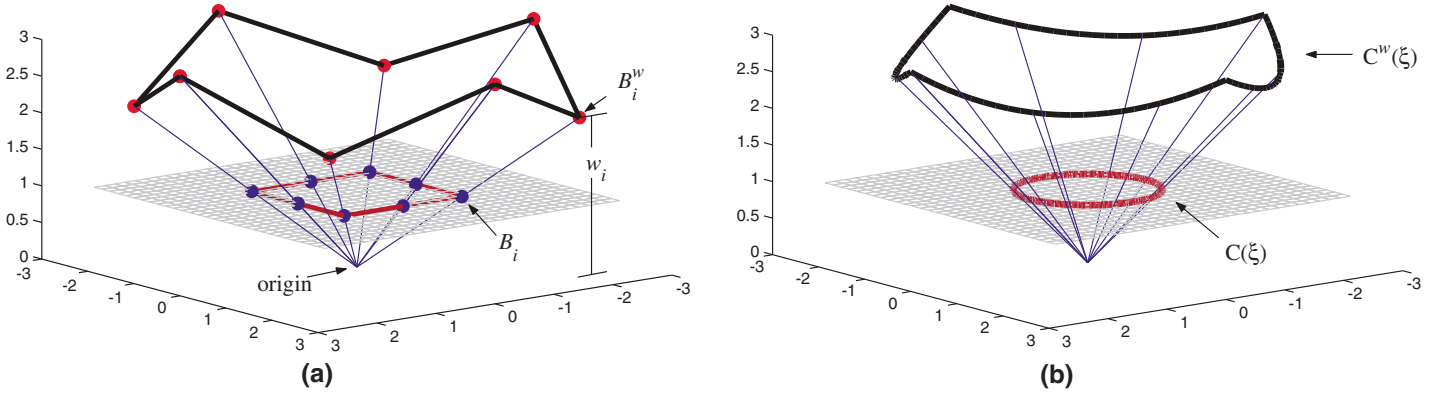
Fig. 12. Circle in $\mathbb{R}^2$ constructed by projective transformation of piecewise quadratic B-spline in $\mathbb{R}^3$. (a) Projective transformation of "projective control points" yields control points. Weight w_i is the z-component of B_i^w. (b) Projective transformation of B-spline curve $C^w(\xi)$ yields NURBS curve $C(\xi)$. See Appendix A.1 for details.

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 control points by the following relations:

$$(B_i)_j = (B_i^w)_j / w_i, \quad j = 1, \ldots, d, \tag{9}$$

$$w_i = (B_i^w)_{d+1}. \tag{10}$$

where $(B_i)_j$ is the jth component of the vector B_i, etc. and w_i is referred to as the ith *weight*. In Fig. 12a, the weights are the vertical coordinates of the control points defining the piecewise quadratic B-spline curve in $\mathbb{R}^3$. The rational basis functions and NURBS curve are given by

$$R_i^p(\xi) = \frac{N_{i,p}(\xi) w_i}{\sum_{\hat{i}=1}^n N_{\hat{i},p}(\xi) w_{\hat{i}}}, \tag{11}$$

$$C(\xi) = \sum_{i=1}^n R_i^p(\xi) B_i. \tag{12}$$

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

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

$$R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) = \frac{N_{i,p}(\xi) M_{j,q}(\eta) L_{k,r}(\zeta) w_{i,j,k}}{\sum_{\hat{i}=1}^n \sum_{\hat{j}=1}^m \sum_{\hat{k}=1}^l N_{\hat{i},p}(\xi) M_{\hat{j},q}(\eta) L_{\hat{k},r}(\zeta) w_{\hat{i},\hat{j},\hat{k}}}. \tag{14}$$

As an example, Fig. 13 shows a control net and the corresponding NURBS surface description of a torus. The paraphernalia describing the constructions in Figs. 12 and 13 are provided in Appendices A.1 and A.2, respectively.

Important properties of NURBS are:

(1) NURBS basis functions form a partition of unity.
(2) The continuity and support of NURBS basis functions are the same as for B-splines.
(3) Affine transformations in physical space are obtained by applying the transformation to the control points, that is, NURBS possess the property of affine covariance.
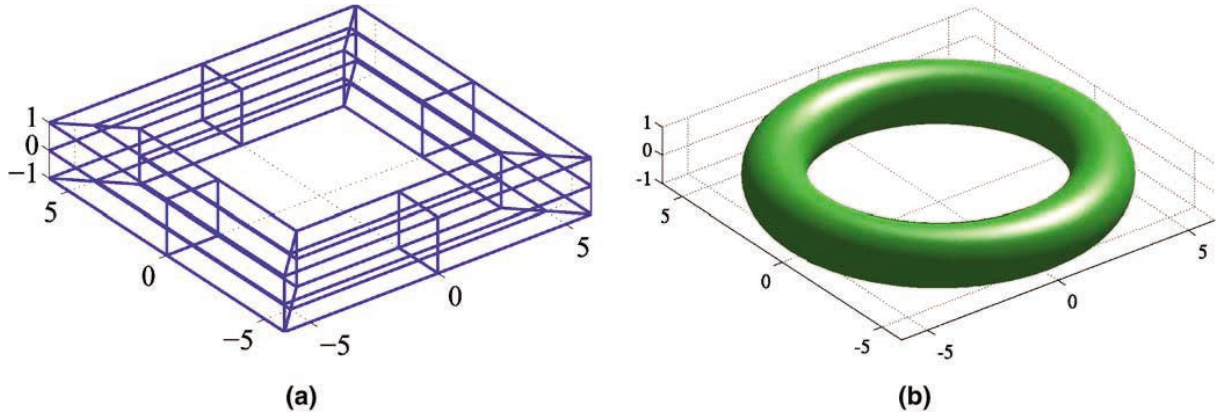
Fig. 13. (a) Control net for toroidal surface and (b) toroidal surface. See Appendix A.2 for details.

(4) If weights are equal, NURBS become B-splines (i.e., piecewise polynomials).
(5) NURBS surfaces and solids are the projective transformations of tensor product, piecewise polynomial entities.

3. NURBS as a basis for analysis

An analysis framework based on NURBS consists of the following items and features:

(1) A mesh for a NURBS patch is defined by the product of knot vectors. For example, in three dimensions, a mesh is given by $\varXi \times \mathscr{H} \times \mathscr{L}$.
(2) Knot spans subdivide the domain into "elements".
(3) The support of each basis function consists of a small number of "elements".
(4) The control points associated with the basis functions define the geometry.
(5) The isoparametric concept is invoked, that is, the fields in question (e.g., displacement, velocity, temperature, etc.) are represented in terms of the same basis functions as the geometry. The coefficients of the basis functions are the degrees-of-freedom, or *control variables*.
(6) Mesh refinement strategies are developed from a combination of knot insertion and order elevation techniques. These enable analogues of classical *h*-refinement and *p*-refinement methods, and the new possibility of *k*-refinement.
(7) Arrays constructed from isoparametric NURBS patches can be assembled into global arrays in the same way as finite elements; see Hughes [20, Chapter 2]. Compatibility of NURBS patches is attained by employing the same NURBS edge and surface representations on both sides of patch interfaces; see Fig. 14a. This gives rise to a standard continuous Galerkin method. Refinement necessarily propagates from patch to patch. There are two alternatives corresponding to Fig. 14b. One is the discontinuous Galerkin method, which can be employed at the patch level. Compatibility is enforced weakly by the variational formulation. The other is to utilize constraint equations for the control points and control variables to attain pointwise compatibility at patch interfaces. See Kagan et al. [22] for a procedure of this kind in the context of B-splines. Local refinement is an important and challenging research topic.
(8) The easiest way to set Dirichlet boundary conditions is to apply them to the control variables. In the case of homogeneous Dirichlet conditions, this results in exact, pointwise satisfaction. In the case of inhomogeneous Dirichlet conditions, the boundary values must be approximated by functions lying within the NURBS space. This amounts to "strong" but approximate satisfaction of the boundary con-
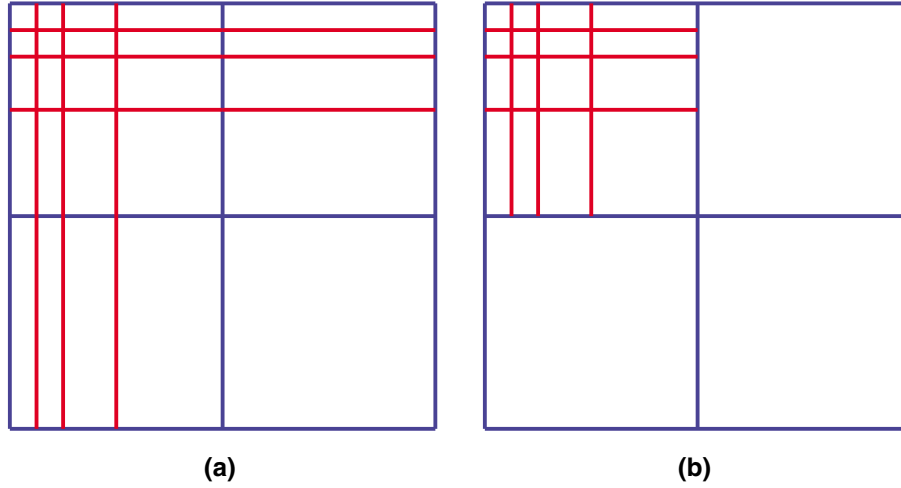
Fig. 14. (a) Global refinement employing the continuous Galerkin method and (b) local refinement employing the discontinuous Galerkin method or constraint equations at the patch level.

ditions. Constraint equations can also be written to ensure "strong" exact, interpolated, or least-squares best-fit satisfaction of boundary conditions. An alternative formulation of Dirichlet conditions can be based on "weak" satisfaction, a standard feature of the discontinuous Galerkin method. Given the variety of possibilities, Dirichlet boundary conditions need to be researched more thoroughly to determine optimal strategies. Neumann boundary conditions are satisfied naturally, in precisely the same way as in standard finite element formulations; see Hughes [20, Chapters 1 and 2].

Remark. It is well known that typical finite element interpolation functions oscillate in attempting to fit discontinuous data. An example is illustrated in Fig. 15a where Lagrange polynomials of orders three, five, and seven interpolate a discontinuity represented by eight data points in $\mathbb{R}^2$. Note that as the order is increased, the amplitude of the oscillations increases. This is sometimes referred to as *Gibbs phenomena*. NURBS behave very differently when the data are viewed as control points. In Fig. 15b the NURBS curves are monotone, illustrating the *variation diminishing* property of NURBS (see [1, Chapter 3]). This property has advantages in representing sharp layers. An example illustrating this will be presented later.
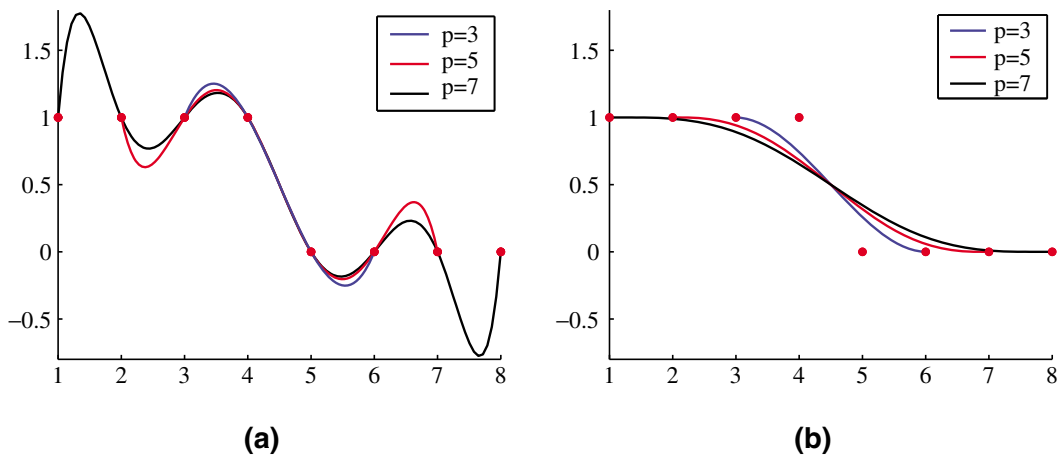

Fig. 15. (a) Lagrange interpolation oscillates when faced with discontinuous data and (b) NURBS exhibit the variation diminishing property for the same data.

Table 1
Comparison of finite element analysis and isogeometric analysis based on NURBS

Finite element analysis		Isogeometric analysis
Nodal points		Control points
Nodal variables		Control variables
Mesh		Knots
Basis interpolates nodal points and variables		Basis does *not* interpolate control points and variables
Approximate geometry		Exact geometry
Polynomial basis		NURBS basis
Gibbs phenomena		Variation diminishing
Subdomains		Patches
	Compact support	
	Partition of unity	
	Isoparametric concept	
	Affine covariance	
	Patch tests satisfied	

A summary of similar and dissimilar finite element and isogeometric analysis concepts is presented in Table 1. A salient feature of isogeometric analysis, shared by meshless methods, is the non-interpolatory nature of the basis. Other shared features are the partition of unity property and compact support. For a state-of-the-art summary of meshless methods, see [23–40].

4. Isogeometric structural analysis

Isoparametric NURBS patches represent all rigid body motions and constant strain states exactly. This follows from the properties of NURBS with respect to affine transformations. Structures assembled from compatible NURBS patches pass standard "patch tests". (Patch tests are described in Chapters 3 and 4 of Hughes [20]. They are considered by engineers to be practical ways of assessing whether or not finite elements are convergent and whether or not they have been programmed correctly.) We have verified this assertion in numerous test cases (not shown). In what follows, we present numerical solutions for linear elastic solids and structures. The Galerkin formulation of linear elasticity is employed (see [20, Chapter 2]). The first example is two-dimensional and the remainder are three-dimensional. Several of the calculations involve thin shells. These are modeled as NURBS solids and *no* shell assumptions are employed. Gaussian quadrature is used on elements. The rule of thumb that we typically employed is to use the lowest-order rule that would exactly integrate the integrand assuming the NURBS are B-splines of the same polynomial order and the Jacobian determinants are constants. This represents an approximation in the case of NURBS. To assess its validity, we performed tests in which we systematically increase the number of quadrature points. For sufficiently fine meshes no differences in results were discernible. However, coarse meshes required more integration points due to large variations in the geometrical mapping. More research needs to be done to determine a robust strategy covering all situations.

Remark. We employed both direct and iterative linear algebraic equation solvers. The direct solver had a profile architecture and was typically more efficient for shell structures than the iterative, diagonally preconditioned, conjugate gradient solver. However, this was not the case for NURBS solids. The code we developed was restricted to a single processor and many of the finer-mesh cases could not be solved with the direct solver because they exceeded available memory. The aspect ratios of elements in some cases were quite large and the number of equations approached a quarter of a million. Nevertheless, in all

cases the iterative procedure converged without difficulty. We did not test the corresponding finite element cases but we would have been surprised if iteration behaved this reliably in those cases too. Consequently, we conjecture that the NURBS cases may be better conditioned than the corresponding finite element cases. The evidence we have for this is rather skimpy but we think it warrants further investigation.

4.1. Infinite plate with circular hole under constant in-plane tension in the x-direction

In this two-dimensional example, we present in some detail the NURBS analysis of a problem in solid mechanics having an exact solution. We also systematically explore h- and k-refinement. The infinite plate is modeled by a finite quarter plate. The exact solution [41, pp. 120–123], evaluated at the boundary of the finite quarter plate, is applied as a Neumann boundary condition and is given here for reference:

$$\sigma_{rr}(r,\theta) = \frac{T_x}{2}\left(1 - \frac{R^2}{r^2}\right) + \frac{T_x}{2}\left(1 - 4\frac{R^2}{r^2} + 3\frac{R^4}{r^4}\right)\cos 2\theta, \tag{15}$$

$$\sigma_{\theta\theta}(r,\theta) = \frac{T_x}{2}\left(1 + \frac{R^2}{r^2}\right) - \frac{T_x}{2}\left(1 + 3\frac{R^4}{r^4}\right)\cos 2\theta, \tag{16}$$

$$\sigma_{r\theta}(r,\theta) = -\frac{T_x}{2}\left(1 + 2\frac{R^2}{r^2} - 3\frac{R^4}{r^4}\right)\sin 2\theta, \tag{17}$$

where T_x is the magnitude of the applied stress for the infinite plate case. The setup is illustrated in Fig. 16. R is the radius of the hole, L is the length of the finite quarter plate, E is Young's modulus, and v is Poisson's ratio. A rational quadratic basis is the minimum order capable of representing a circular hole. The coarsest mesh, $\varXi \times \mathscr{H}$, is defined by the knot vectors

$$\varXi = \{0, 0, 0, 0.5, 1, 1, 1\}, \tag{18}$$

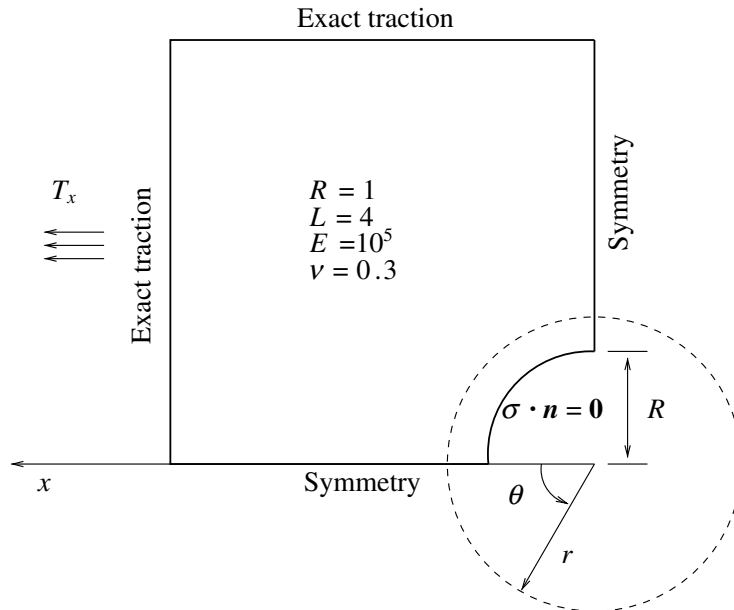$$\mathscr{H} = \{0, 0, 0, 1, 1, 1\}. \tag{19}$$

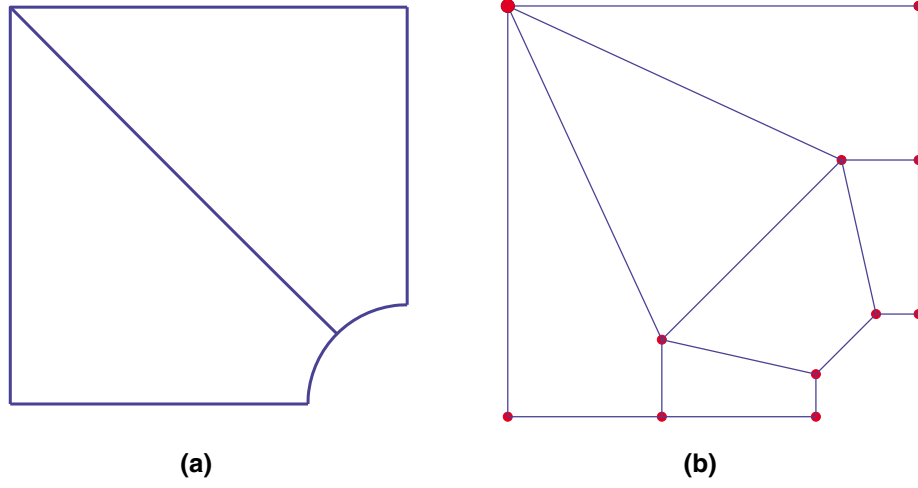Fig. 16. Elastic plate with a circular hole: problem definition.

Fig. 17. Mesh and control net for the elastic plate with circular hole. Two control points at same location create the upper-left corner. Knot values define "elements", not the control net: (a) coarse mesh and (b) control net.
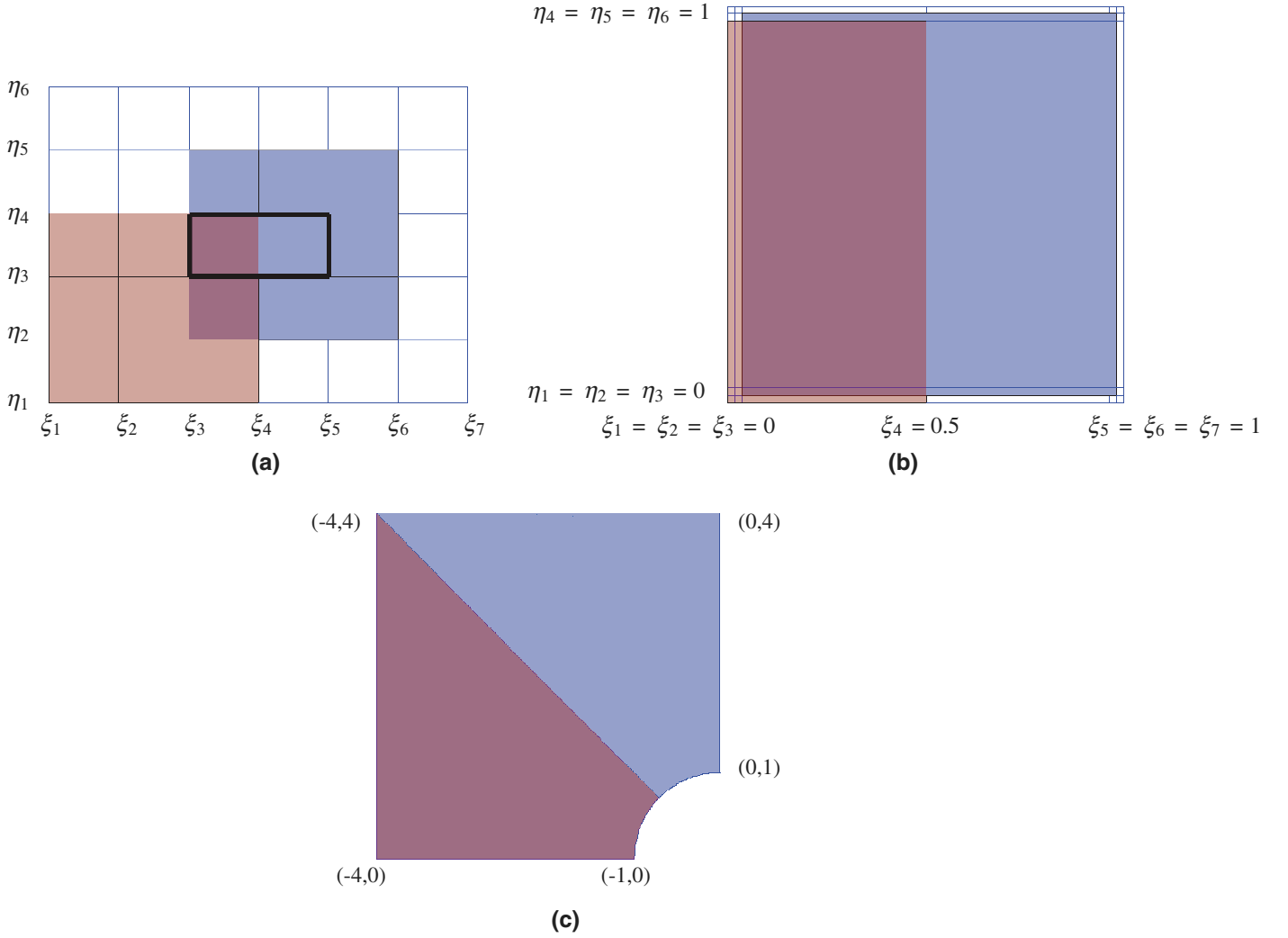

Fig. 18. Different views of the coarsest mesh and support of two of the basis functions. The support of rational quadratic basis functions consists of 3×3 knot spans. Open knot vectors require that knot spans near the boundary have zero measure: (a) Index space; (b) parameter space and (c) physical space.
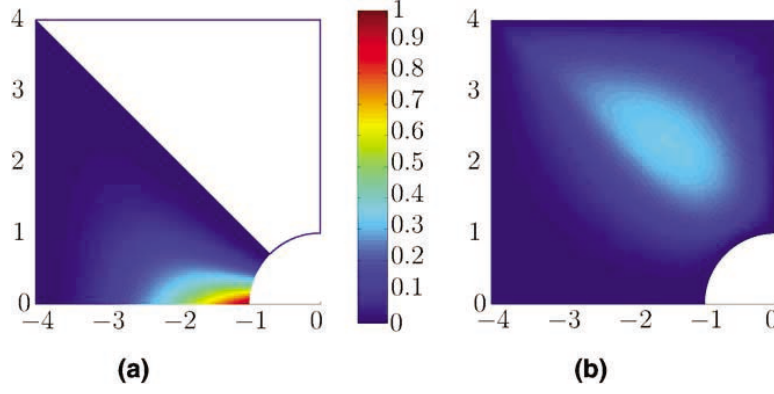
Fig. 19. Basis function contours: (a) The first basis function has support in one element and takes on the value 1 at $x = -1$, $y = 0$ and (b) the second basis function has support in both elements. The maximum amplitude of this function is approximately 0.4.

The exact geometry is represented with only two elements, as shown in Fig. 17a. The corresponding control net is shown in Fig. 17b. A summary of the geometric data is given in Appendix A.3. A repeated control point is responsible for the upper-left hand corner. Fig. 18 depicts three different views of the mesh and support of two of the basis functions. These are the index space, parameter space, and physical space views. The index space illustrates how only parts of the basis functions are supported by the physical space, identified by the dark-bordered rectangle (i.e., $[\xi_3, \xi_5] \times [\eta_3, \eta_4]$) in Fig. 18a. This is due to the open knot vectors. Consequently, knot spans near the boundary have zero measure and may be ignored in the element formation phase.

Contour plots of the same two basis functions are shown in Fig. 19. The first six meshes used in the analysis are shown in Fig. 20. The scheme used to select knots for insertion is described in Appendix B.

Contour plots of results obtained on meshes 1, 4, and 7 are presented in Fig. 21. The applied stress is $T_x = 10$ and the contours show that the stress concentration of $\sigma_{xx} = 30$ at the edge of the hole (i.e., at $r = R$, $\theta = 3/2\pi$) is obtained as the mesh is refined.

Convergence results in the L^2-norm of stresses are shown in Fig. 22. The cubic and quartic NURBS are obtained by order elevation of the quadratic NURBS on the coarsest mesh. Since the parameterization of
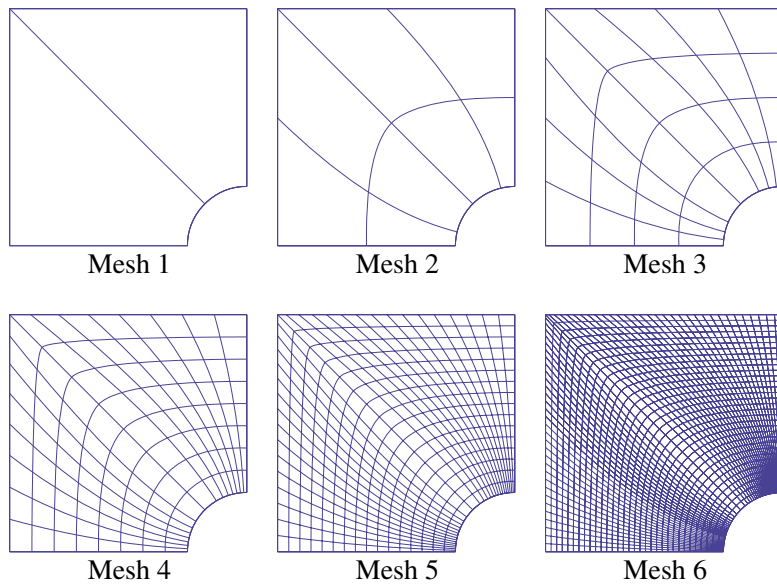

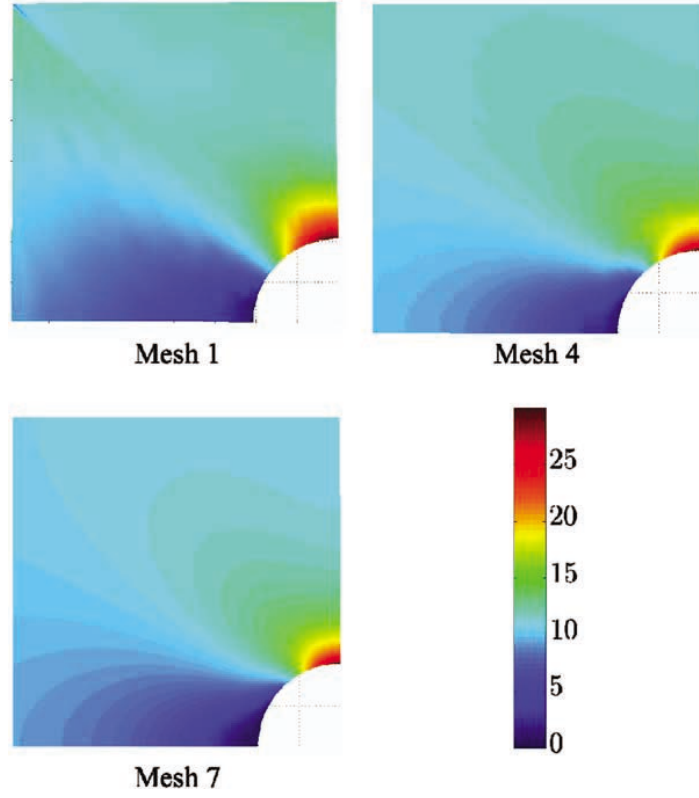Fig. 20. Elastic plate with circular hole. Meshes produced by h-refinement (knot insertion).

Fig. 21. Contour plots of σ_{xx} obtained with quadratic NURBS. The applied stress is $T_x = 10$ and the stress concentration is $\sigma_{xx} = 30$ at $r = R$, $\theta = 3/2\pi$.
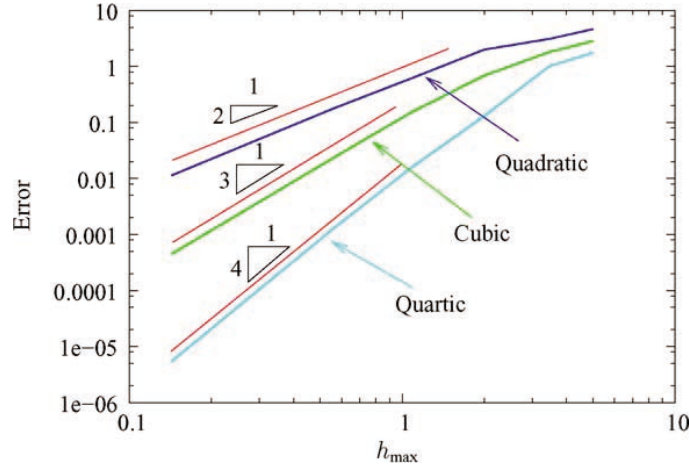

Fig. 22. Error measured in the L^2-norm of stress vs. mesh parameter.

the geometrical mapping does not change, the *h*-refinement algorithm generates identical meshes for all polynomial orders. As a result, the continuity of the basis is C^{p-1} everywhere, except along the line which joins the center of the circular edge with the upper left-hand corner of the domain (see Fig. 17a). There it is C^1 as is dictated by the coarsest mesh employing rational quadratic parameterization. The mesh parameter, $h_{\max}$, is defined as the maximum distance, in physical space, between diagonally opposite knot locations. As
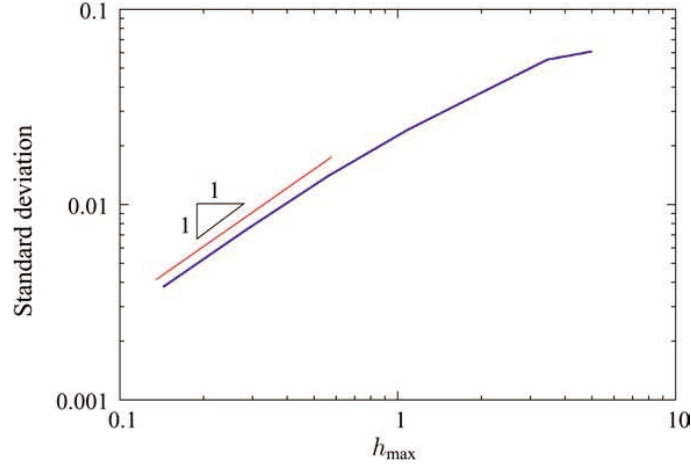
Fig. 23. Standard deviation of weights for rational quadratic elements vs. mesh parameter.

can be seen, the L^2-convergence rates of stress for quadratic, cubic, and quartic NURBS are approximately 2, 3, and 4, respectively. This is the best one could reasonably hope for.

The convergence of the standard deviation of weights for rational quadratic elements is depicted in Fig. 23. As may be seen, the standard deviation converges to zero, meaning the weights converge to a constant within each element and the NURBS approach polynomial B-splines. The geometric way to see this is to note that the mesh in physical space (i.e., $\mathbb{R}^2$) can be "lifted" to projective space (i.e., $\mathbb{R}^3$). As the elements in projective space shrink in size, the projective control points move closer together. As the weights are just the third component of the projective control points, within an element they necessarily converge to a constant value. This geometric argument suggests the rate of convergence should be O(h_{max}), as observed in Fig. 23.

4.2. Solid circular cylinder subjected to internal pressure loading

The problem specification is shown in Fig. 24. Plane strain conditions are assumed to hold in the axial direction. It is a simple matter to obtain an exact solution to this problem, assuming the pressure varies at most circumferentially (see [41, 117–119]). We provide the exact solution for the case with constant pressure as a reference:
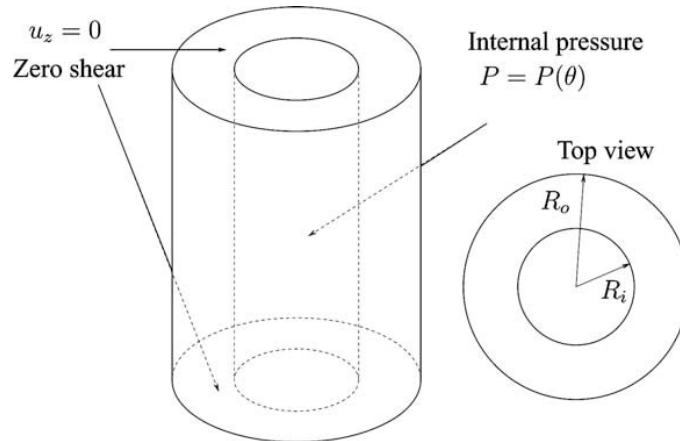

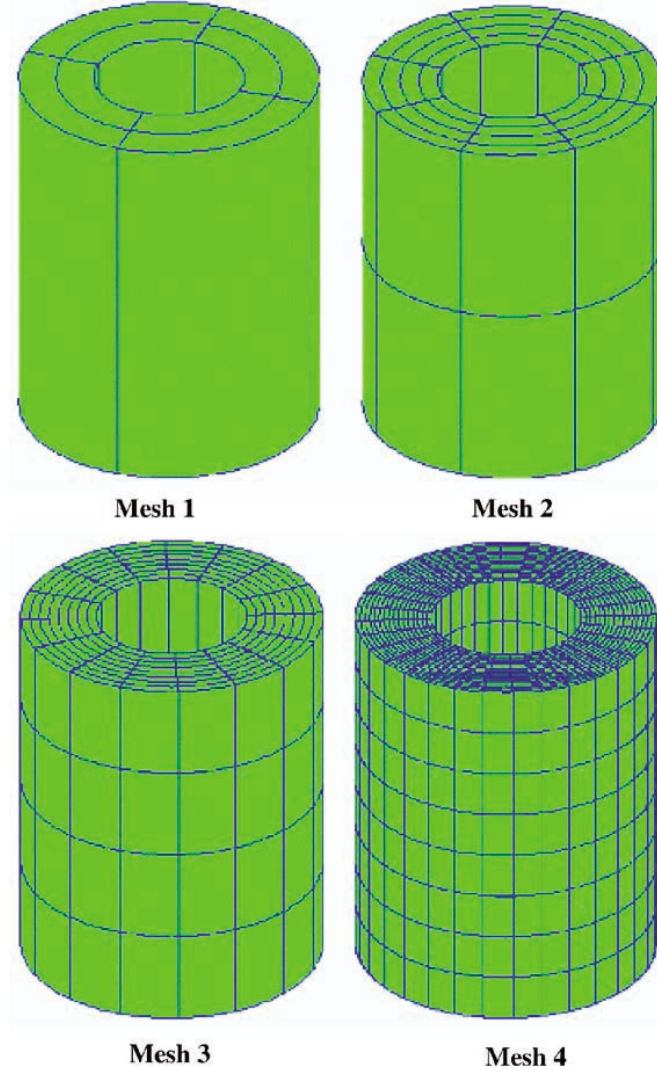Fig. 24. Thick cylinder pressurized internally.

Fig. 25. Solid circular cylinder meshes produced by *h*-refinement (knot insertion).

$$u_r(r) = \frac{1}{E} \frac{PR_i^2}{R_o^2 - R_i^2} \left((1 - v)r + \frac{R_o^2(1 + v)}{r} \right), \tag{20}$$

$$\sigma_{rr}(r) = \frac{PR_i^2}{R_o^2 - R_i^2} - \frac{PR_i^2 R_o^2}{r^2(R_o^2 - R_i^2)}, \tag{21}$$

$$\sigma_{\theta\theta}(r) = \frac{PR_i^2}{R_o^2 - R_i^2} + \frac{PR_i^2 R_o^2}{r^2(R_o^2 - R_i^2)}. \tag{22}$$

Meshes developed from *h*-refinement are shown in Fig. 25. The NURBS paraphernalia for the coarsest mesh is given in Appendix A.4.

Results for the constant pressure case, employing quadratic NURBS, are presented in Fig. 26. In Fig. 26a the axisymmetric deformation of the cylinder is clearly apparent for Mesh 4. Axisymmetric response is attained for all meshes. Errors in radial displacement are plotted in Fig. 26b. For the first (coarsest) mesh, the maximum error through the wall thickness is approximately 1%. For the second mesh it is approximately 0.1% and for the third 0.01%. Order elevated solutions employing cubic and quartic NURBS are, for all practical purposes, exact on all meshes. Hence, results for these cases are not presented.
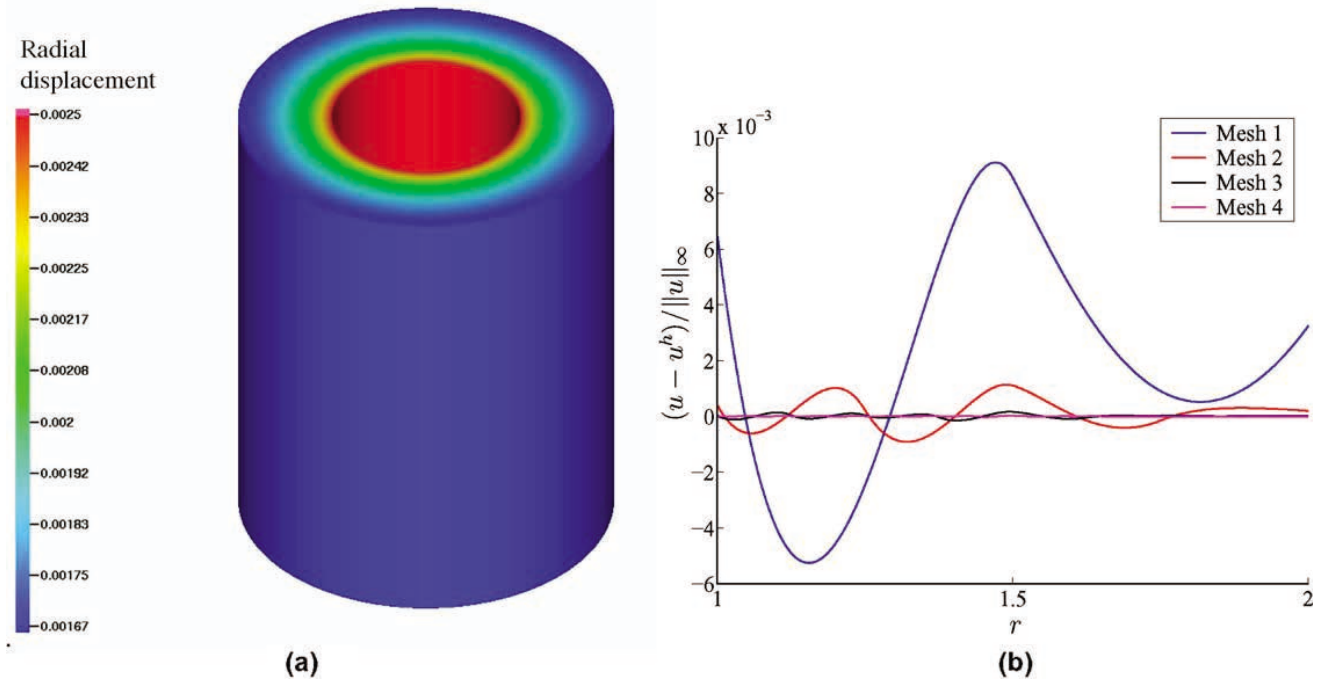
Fig. 26. Solid circular cylinder subjected to constant internal pressure: (a) radial displacement contours and (b) convergence of radial displacement.

In order to determine convergence rates, a somewhat more complex loading was considered in which the internal pressure was assumed to vary as $\cos 2\theta$. Quadratic, cubic, and quartic NURBS were considered. The cubic and quartic cases were obtained from the quadratic case by k-refinement, in which case the degree of continuity was increased to C^2 and C^3, respectively. The rates of convergence of the error measured in the energy norm are presented in Fig. 27. As in the example of the plate with a circular hole, the rates of convergence for quadratic, cubic, and quartic NURBS elements are 2, 3, and 4, respectively.

4.3. Solid "horseshoe" subjected to equal and opposite in-plane flat edge displacements

The problem setup is presented in Fig. 28a. The top surfaces are displaced in the directions shown. This introduces an asymmetry to the loading that is visible in the stress contours. Meshes 1, 3, and 5 are depicted
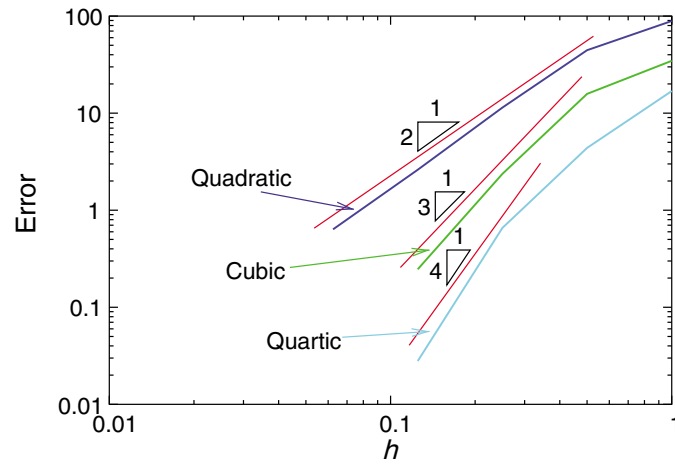

Fig. 27. Solid circular cylinder with varying internal pressure. Convergence of the error in the energy norm for quadratic, cubic, and quartic NURBS discretizations.
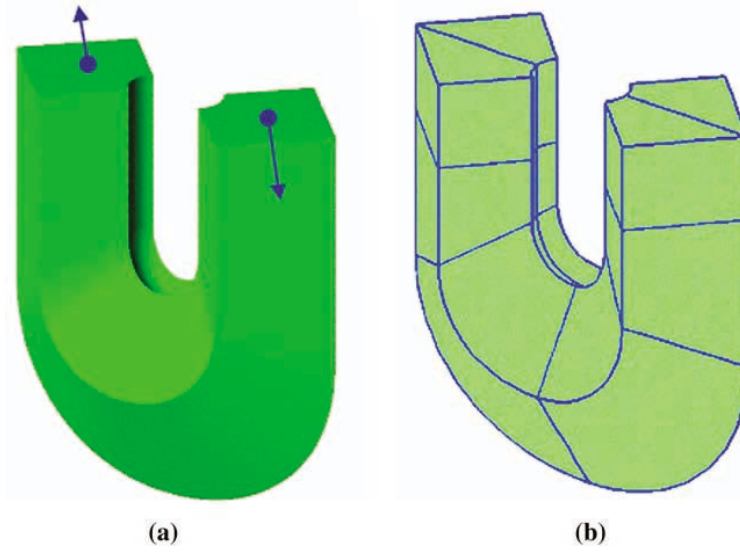
23

Fig. 28. Solid "horseshoe" subjected to equal and opposite in-plane flat edge displacements: (a) problem geometry and (b) coarsest mesh.
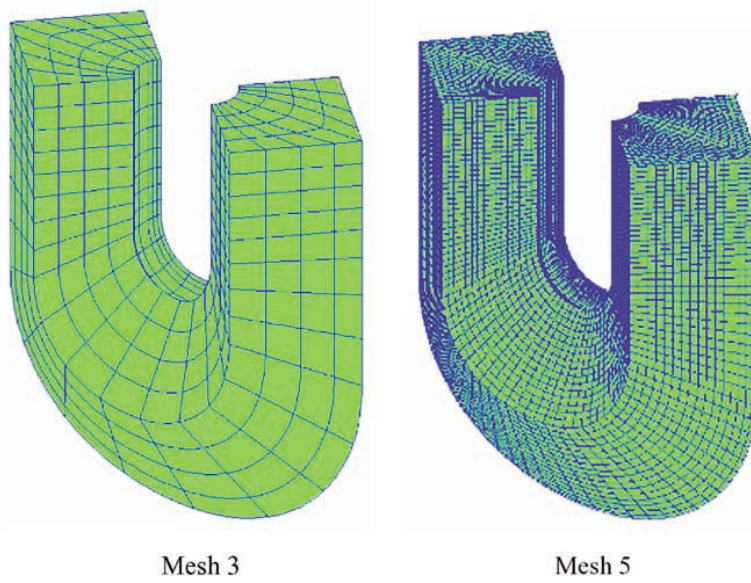

Mesh 3 Mesh 5

Fig. 29. Other meshes for the solid "horseshoe".

in Figs. 28b and 29. The standard *h*-refinement procedure, as described in Appendix B, is employed to obtain finer meshes. It is clear for this case that good quality, refined meshes can be simply generated, once an exact parameterized geometry is constructed in the form of the first mesh. Fig. 30 shows sample plots of stress contours for a quadratic NURBS solution on Mesh 5. Note the smoothness of the stress contours.

4.4. Thin cylindrical shell with fixed ends subjected to constant internal pressure

The problem setup and a radial displacement profile are shown in Fig. 31. Note that the fixed ends create boundary layers which are difficult to accurately capture with low order finite element methods. The exact shell theory solution is given in Timoshenko and Woinowsky-Krieger [42, pp. 476–477], for plane stress and is provided as a reference below
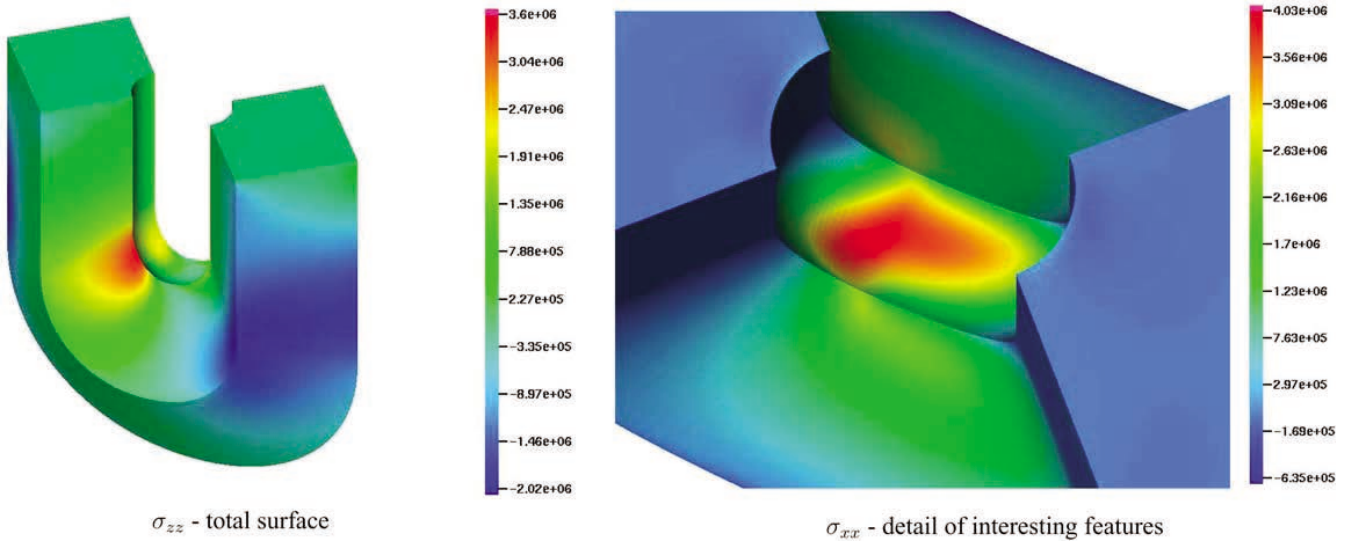
σ_{zz} - total surface

σ_{xx} - detail of interesting features

Fig. 30. Solid "horseshoe". Contour plots of stress components.
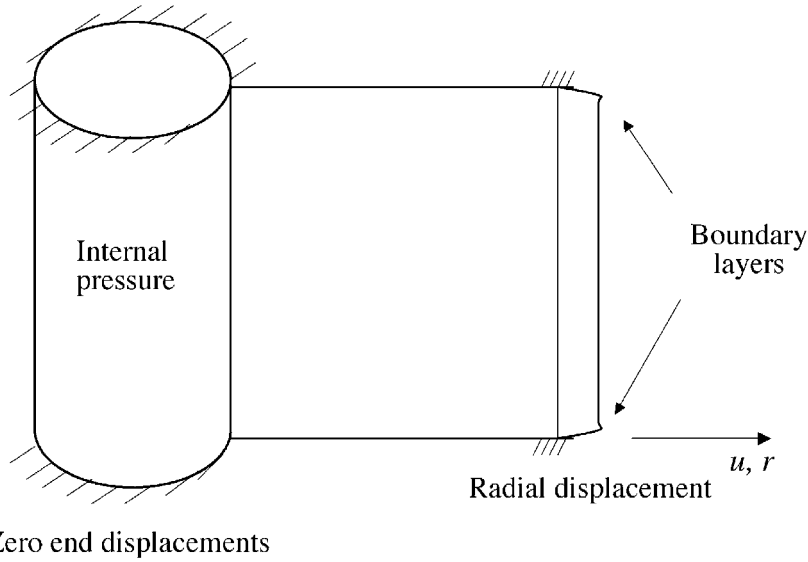
Fig. 31. Thin cylindrical shell. Problem statement and displacement profile.

$$u(x) = -\frac{PR^2}{Et}(1 - C_1 \sin \beta x \sinh \beta x - C_2 \cos \beta x \cosh \beta x) \quad x \in (-L/2, L/2), \tag{23}$$

$$C_1 = \frac{\sin \alpha \cosh \alpha - \cos \alpha \sinh \alpha}{\sinh \alpha \cosh \alpha + \sin \alpha \cos \alpha}, \tag{24}$$

$$C_2 = \frac{\cos \alpha \sinh \alpha + \sin \alpha \cosh \alpha}{\sinh \alpha \cosh \alpha + \sin \alpha \cos \alpha}, \tag{25}$$

$$\beta = \left(\frac{Et}{4R^2D}\right)^{1/4}, \quad \alpha = \frac{\beta L}{2}, \quad D = \frac{Et^3}{12(1 - v^3)}. \tag{26}$$

25

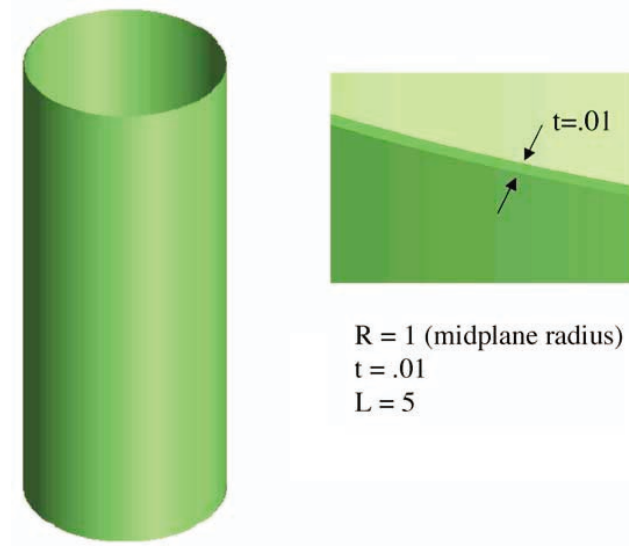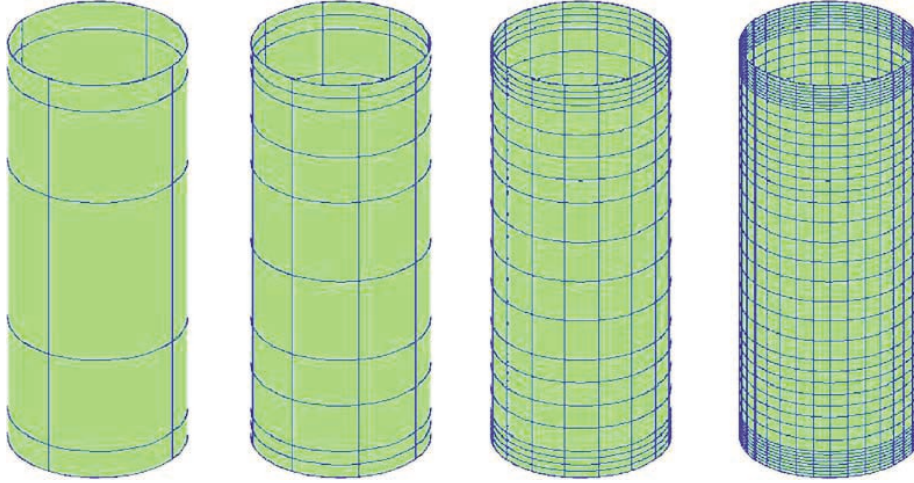Fig. 32. Thin cylindrical shell geometry.

Fig. 33. Thin cylindrical shell surface meshes. Meshes 1–4.

The Young's modulus and Poisson's ratio in this solution need to be replaced by $\frac{E}{1-\nu^2}$ and $\frac{\nu}{1-\nu}$, respectively, to account for the fixed-end conditions assumed here. The geometry of the shell is shown in Fig. 32. Note that the radius to thickness ratio is 100. The meshes are depicted in Figs. 33 and 34. The first four surface meshes are shown in Fig. 33. Note the added refinement in the region of the boundary layer. We wish to emphasize that, despite the shell being very thin, we are modeling it with solid elements. (For an excellent comprehensive review of approaches to shell modeling, see Bischoff et al. [43].)

The through-thickness mesh resolution for Mesh 1 is shown in Fig. 34. Note that there are two elements in the radial direction and four in the circumferential direction. As the surface mesh is refined, the number of elements in the circumferential direction increases accordingly. However, the number of elements in the radial direction is fixed at two throughout the refinement process. The functions employed for this problem are quadratic NURBS. Their appearance in the radial, or through-thickness, direction is presented in Fig. 34a. Radial displacement contours are presented in Fig. 35 for Meshes 2–5. Note that, for all meshes, perfectly axisymmetric response is obtained. Note also the appearance of boundary layers. The convergence of the radial displacement profile is shown on Fig. 36. Mesh 1 is too coarse to represent both the boundary layers and the plateau between. Mesh 3 picks up the plateau but the boundary layers are still not accurately

26
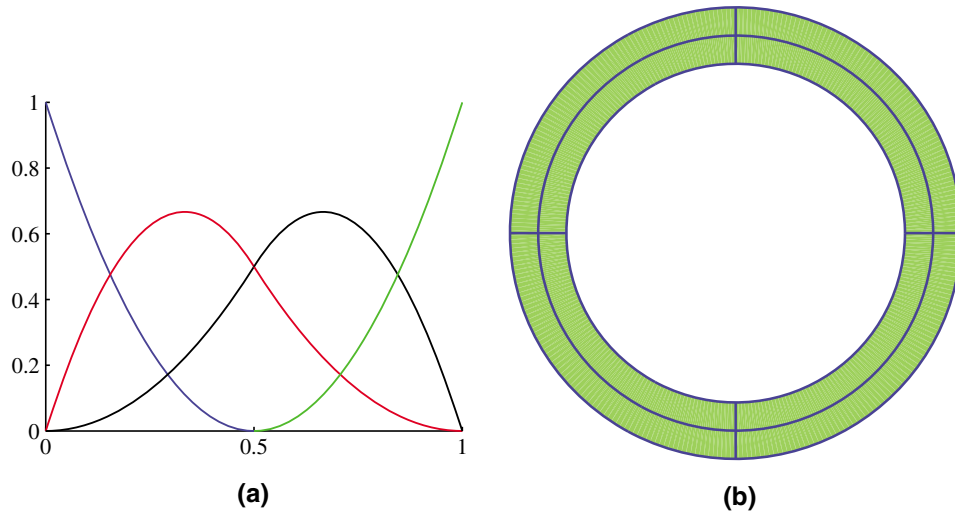
Fig. 34. Thin cylindrical shell: (a) quadratic basis functions through the thickness and (b) end view of the coarse mesh.
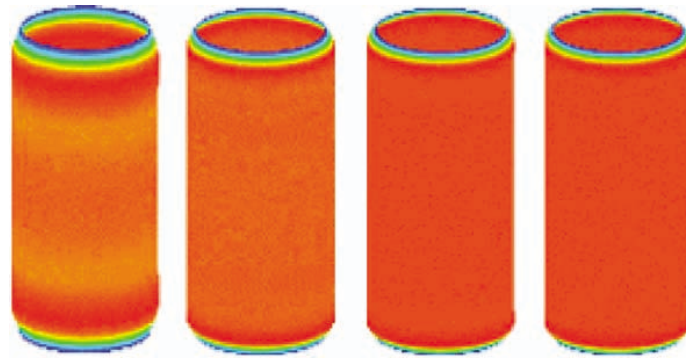
Fig. 35. Thin cylindrical shell. Meshes 2–5. Radial displacement contour plots.
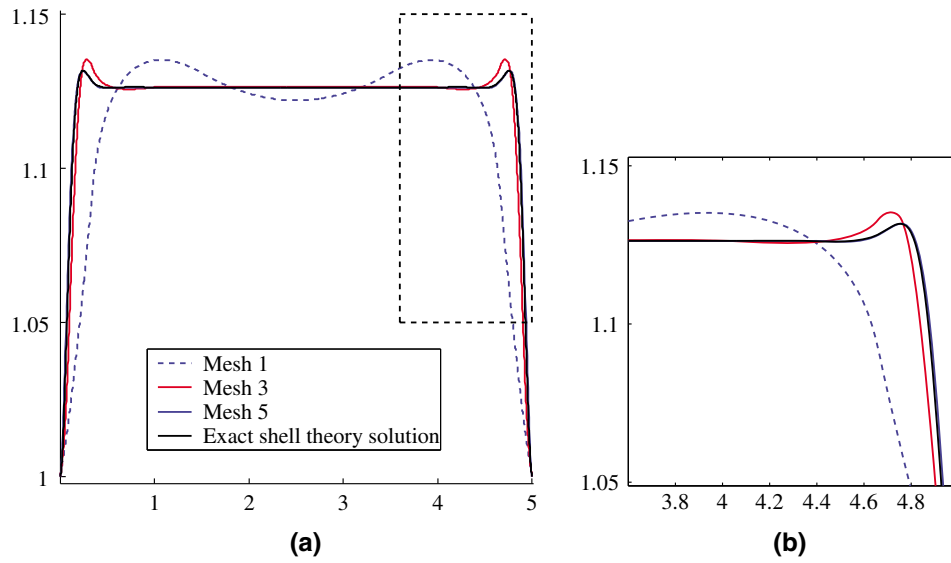
Fig. 36. Thin cylindrical shell. Convergence of radial displacement to exact shell theory solution. Mesh 5 solution indistinguishable from exact. NURBS are capable of accurately resolving shell boundary layers: (a) global radial displacement and (b) detail of the boundary layer.

captured. The Mesh 5 solution is indistinguishable from the exact shell theory solution. In the detail on the right, the exact shell solution and Mesh 5 solution are seen to overlap in the boundary layer region.

4.5. Shell obstacle course

The so-called shell obstacle course consists of three problems, the Scordelis–Lo roof, the pinched hemi-sphere, and the pinched cylinder. These problems, and their relevance to the assessment of shell analysis procedures, have been discussed extensively in the literature. The problem descriptions presented in Figs. 37–39, are adapted from Felippa [44] and Belytschko et al. [45]. As in the previous example, two quadratic

E	$=$	4.32×10^8
ν	$=$	0.0
g	$=$	90
R	$=$	25
L	$=$	50
t	$=$	0.25
ϕ	$=$	$40°$

Fig. 37. Shell obstacle course. Scordelis–Lo roof problem description and data.

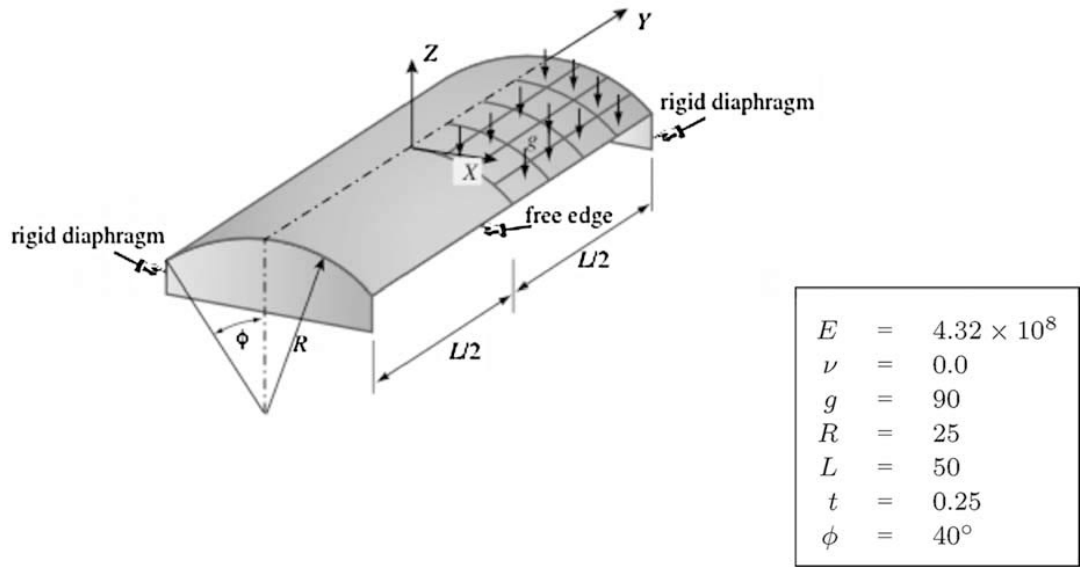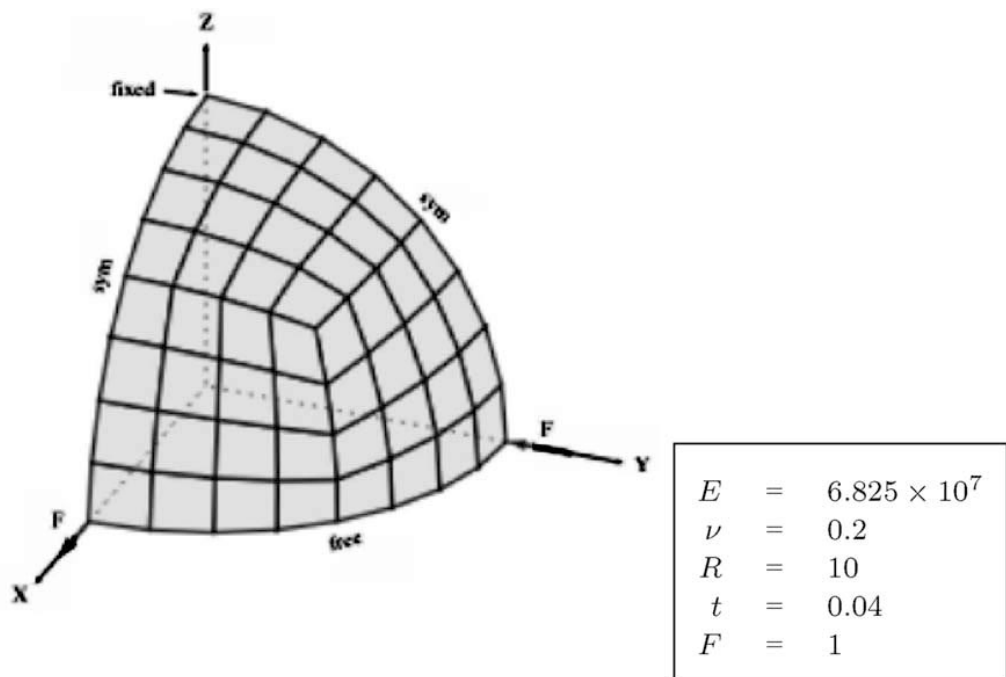|		
E	$=$	6.825×10^7
ν	$=$	0.2
R	$=$	10
t	$=$	0.04
F	$=$	1

Fig. 38. Shell obstacle course. Pinched hemisphere problem description and data.
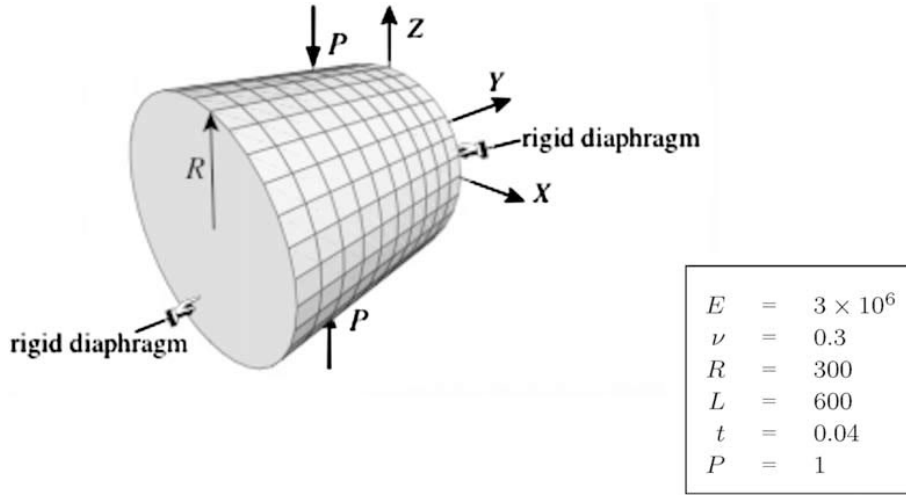
Fig. 39. Shell obstacle course. Pinched cylinder problem description and data.

NURBS elements are employed in the through-thickness direction (see Fig. 34), whereas *h*-refinement and *k*-refinement are utilized for surface meshing. Quadratic through quintic surface NURBS are employed in the convergence analysis of all cases. In one case, the pinched hemisphere, a one-element surface solution, starting with rational quadratics, the lowest-order NURBS capable of exactly representing spherical geometry, and culminating with 10th-order NURBS, is used to assess convergence. This analysis has the flavor of what are usually referred to as "spectral methods," which are accurate and efficient procedures, typically utilized for performing detailed studies of geometrically simple but physically complex phenomena, such as turbulence (see [46,47] for detailed descriptions and applications of spectral methods). Convergence is assessed by comparing the displacement of certain points in the shell with benchmark solutions presented in Belytschko et al. [45]. Sample contour plots of the solutions for quadratic elements and the finest meshes studied are presented in Figs. 40–42. Note that in each case the contours are very smooth.

4.5.1. Scordelis–Lo roof

The Scordelis–Lo roof is subjected to gravity loading. The ends are supported by fixed diaphragms and the side edges are free (see Fig. 37). The vertical displacement of the midpoint of the side edge is the quantity used to assess convergence. The second, fourth, and sixth meshes used in the study are shown in Fig. 43. These meshes have 2, 8, and 32 surface elements per side, respectively. Due to symmetry, only one quadrant
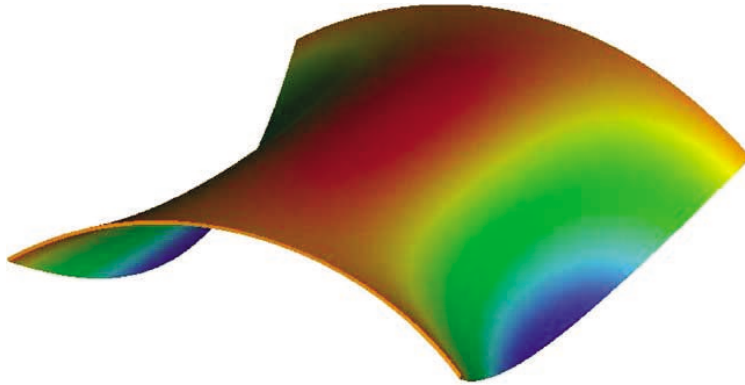
Fig. 40. Shell obstacle course. Scordelis–Lo roof in deformed configuration (scaling factor of 200 used). Contours of displacement in the direction of the gravity load.
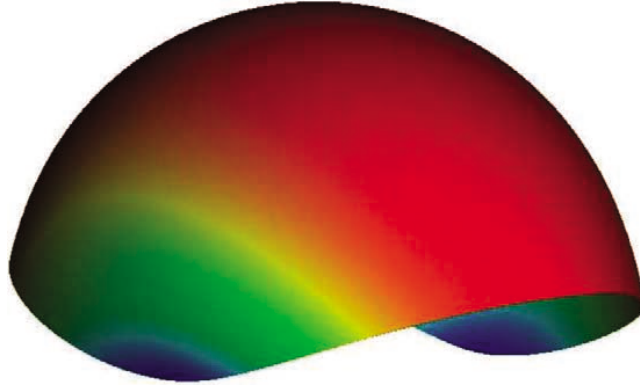
Fig. 41. Shell obstacle course. Pinched hemisphere in deformed configuration (scaling factor of 33.3 used). Contours of displacement in the direction of inward directed point load.
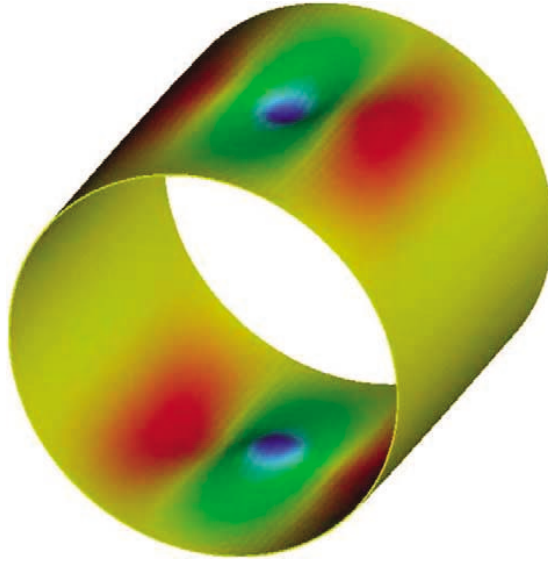
Fig. 42. Shell obstacle course. Pinched cylinder in deformed configuration (scaling factor of 3×10^6 used). Contours of displacement in the direction of the point load. Notice the highly localized displacement in the vicinity of the load.
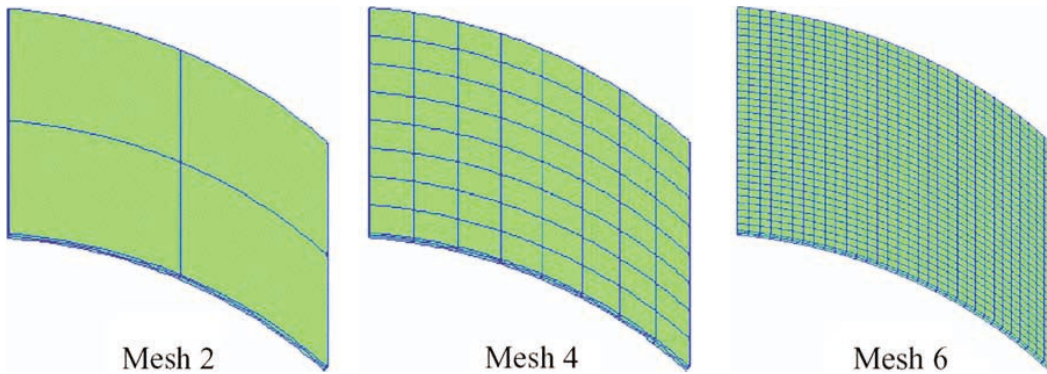
Fig. 43. Shell obstacle course. Scordelis–Lo roof meshes.

is meshed. Convergence of the displacement to the benchmark value is shown in Fig. 44. In all cases, convergence is quite rapid. For the higher-order cases, namely, quartic and quintic, even one element provides a very accurate solution.
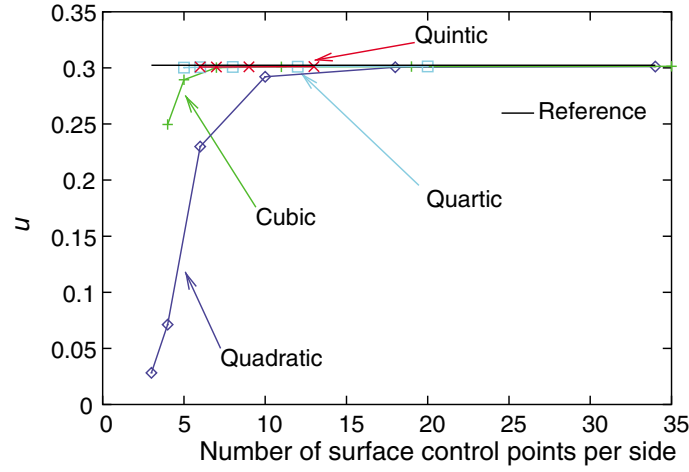
Fig. 44. Shell obstacle course. Scordelis–Lo roof displacement convergence.

4.5.2. Pinched hemisphere

In the pinched hemisphere, equal and opposite concentrated forces are applied at antipodal points of the equator. The equator is otherwise considered to be free (see Fig. 38). The second, fourth, and sixth meshes are shown in Fig. 45. Due to symmetry, only one quadrant is meshed. Convergence of the displacement under the inward directed load is presented in Fig. 46. The quadratic case converges very slowly, which is not surprising as quadratic, fully-integrated, solid finite elements are known to "lock" in shell analysis. Cubic solid finite elements also exhibit locking in similar circumstances but in the present case cubic NURBS behave reasonably well. Fig. 47 presents convergence of the displacement for one surface element meshes. Notice that the lowest-order meshes lock but eventually accurate results are obtained. One 10th-order NURBS surface element is seen to provide an essentially exact result. To assess whether there is any tendency to oscillate, displacement in the direction of the inward directed point load is plotted for the single 10th-order NURBS surface element case in Fig. 48. As is evident, the displacements are very smooth and monotone.

4.5.3. Pinched cylinder

The pinched cylinder is subjected to equal and opposite concentrated forces at its midspan (see Fig. 39). The ends are supported by rigid diaphragms. This constraint results in highly localized deformation under the loads (see Fig. 42). Only one octant of the cylinder is used in the calculation due to symmetry. The second, fourth, and sixth meshes are shown in Fig. 49. Convergence of the displacement under the load is

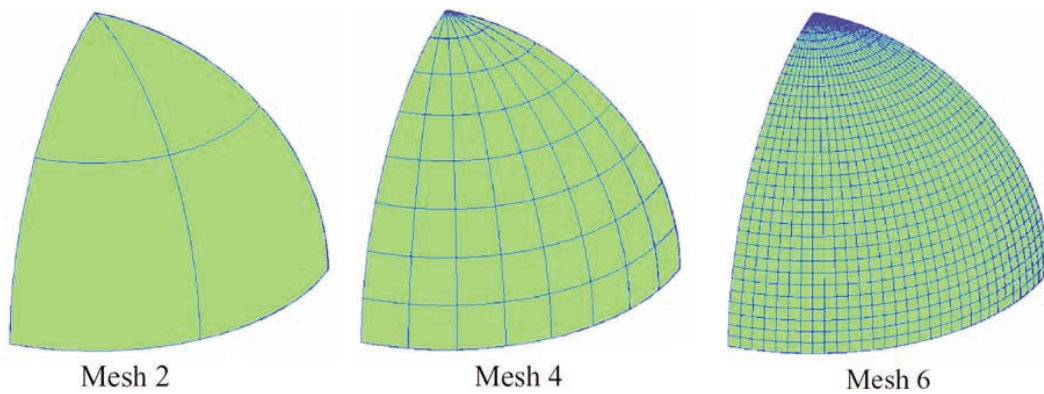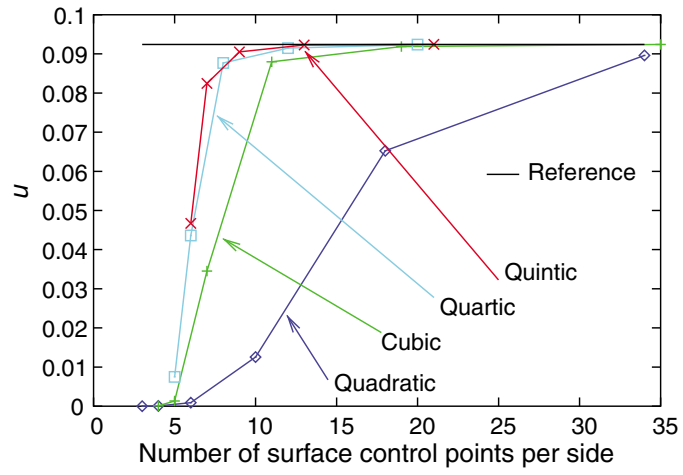Fig. 45. Shell obstacle course. Pinched hemisphere meshes.

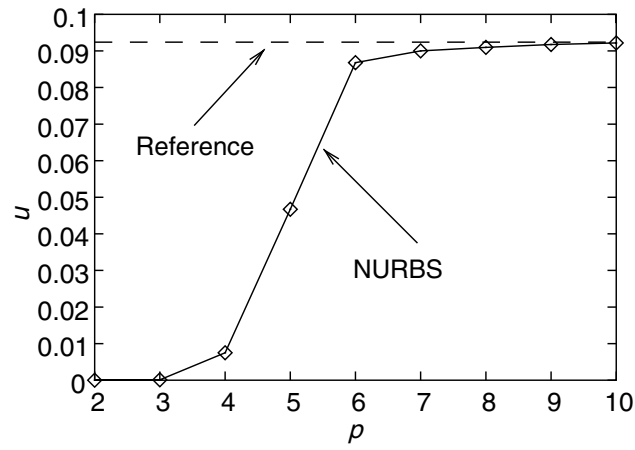Fig. 46. Shell obstacle course. Pinched hemisphere displacement convergence.

Fig. 47. Shell obstacle course. Pinched hemisphere displacement convergence for one NURBS surface element.
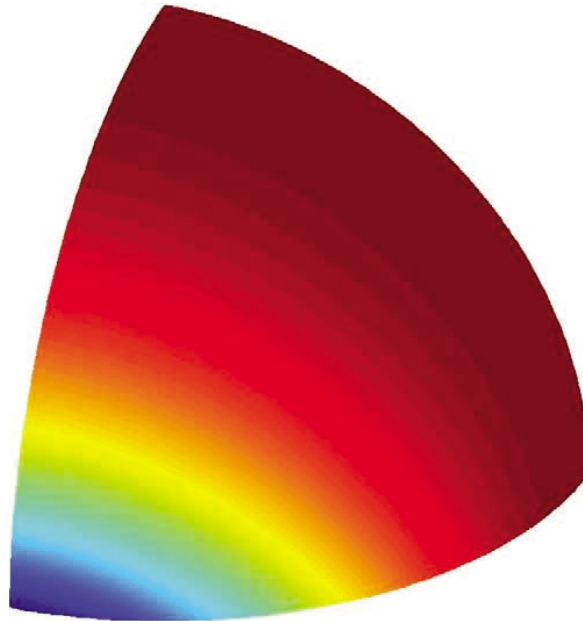
Fig. 48. Shell obstacle course. Pinched hemisphere displacement in the direction of the inward directed point load for one surface element with $p = 10$.
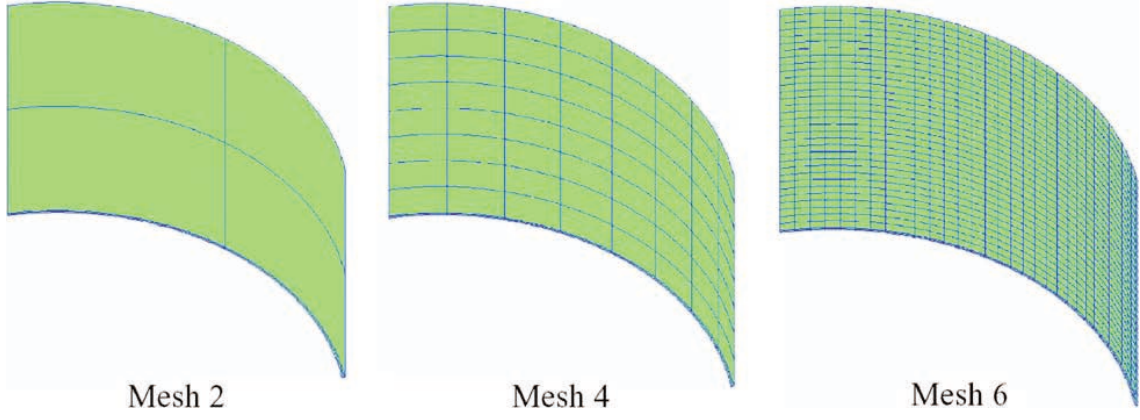
Fig. 49. Shell obstacle course. Pinched cylinder meshes.
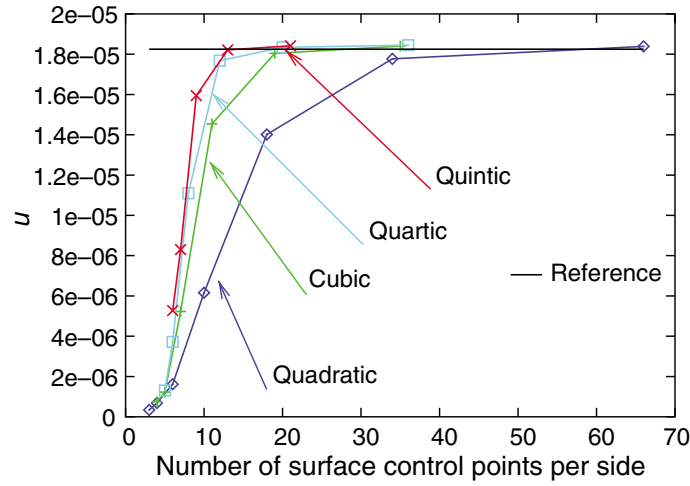
Fig. 50. Shell obstacle course. Pinched cylinder displacement convergence.

presented in Fig. 50. The NURBS elements converge to a very slightly softer solution than the benchmark solution. This may be due to transverse shear effects. It is well known that, as long as the characteristic surface element dimension is large compared with the thickness, formulations which permit transverse shear deformations typically closely approximate formulations which satisfy the Kirchhoff constraint (i.e., zero transverse shear). When this trend reverses, that is as the surface element dimension approaches zero, holding the thickness constant, the displacement under a concentrated load grows, and converges to infinity (see [48] for elaboration).

Remark. We believe that two quadratic NURBS through the thickness are unnecessary for typical thin shell analysis. We performed some tests with one quadratic NURBS through the thickness and the results were indistinguishable when compared with the two-quadratic-NURBS case. However, when we reduced to linear variation through the thickness, convergence to correct solutions was not obtained. Classic shell theory hypotheses, such as invoking the plane stress condition in the through-thickness direction, may be sufficient to correct the deficiency of linear through-thickness displacement variation. Such a formulation would certainly be competitive with traditional shell element formulations which employ displacement and rotation degrees of freedom at a reference surface. Using only displacement degrees of freedom (i.e., control variables) in the NURBS case considerably simplifies shell analysis, especially in non-linear analysis

wherein rotations are no longer vectorial and additive, but require a multiplicative group structure. A further simplification might be just to use a NURBS surface and employ "rotationless" formulations, such as the one for plates described in Engel et al. [49]. In this reference a discontinuous Galerkin formulation is proposed but all interface discontinuity (i.e., "jump") terms in it disappear if C^1, or higher, continuity is satisfied. This is simply attained with NURBS but very difficult to achieve in finite element analysis. Rotationless shell elements have recently gained popularity in computational mechanics (see [17–19,49–53]). The formulation of Engel et al. [49] has also been proposed as being an appropriate basis for so-called "strain gradient theories". These theories also require C^1-continuity, and NURBS would appear to be naturally suited to them.

4.6. Hemispherical shell with a stiffener

We consider the hemispherical shell with stiffener presented in Rank et al. [10] who solved the problem using a finite element method and p-refinement strategy. The geometry and data are depicted in Fig. 51. The structure is subjected to gravity loading and external pressure, and vertical displacements are set to zero on the bottom surface of the stiffener. Only a quarter of the domain is modeled due to symmetry. The initial mesh is constructed using rational quadratic NURBS and is shown in Fig. 52. Isotropic p-refinement was performed to assess convergence. Polynomial orders 2–6 were employed. Figs. 53 and 54 show the vertical displacement and von Mises stress on the deformed configuration. The von Misses stress is given by

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	=	$100.0 \frac{kN}{m^2}$
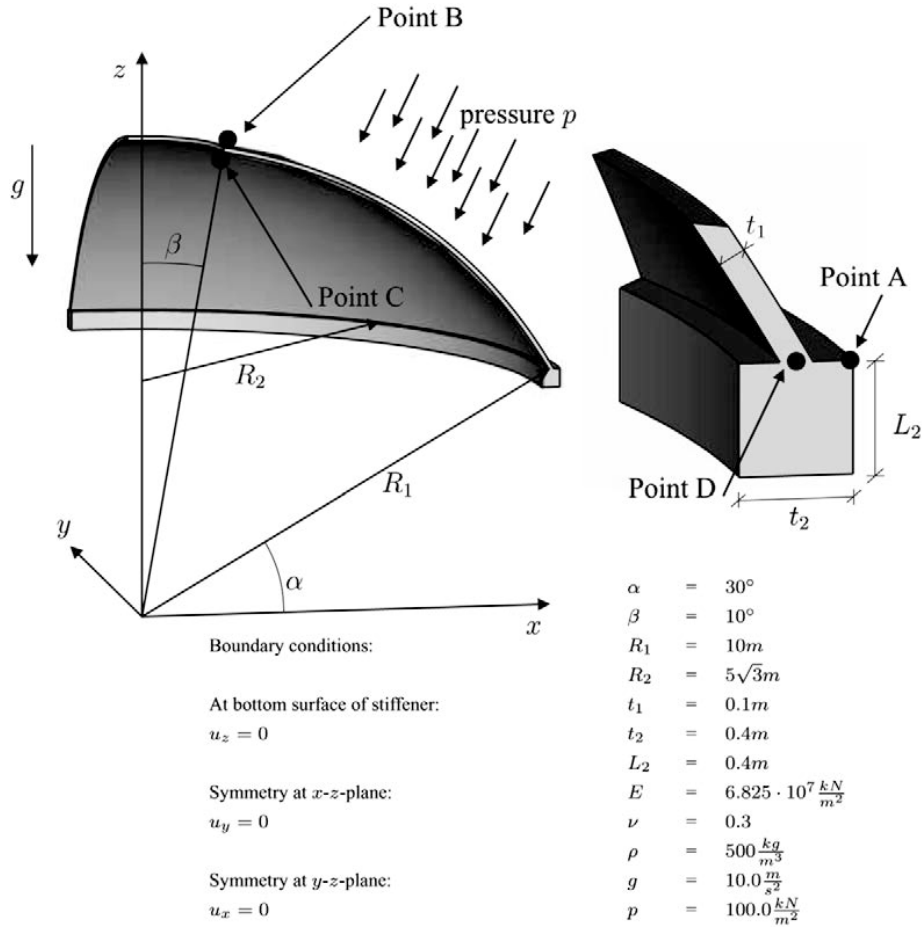
Fig. 51. Hemispherical shell with stiffener. Problem description from Rank et al. [10].
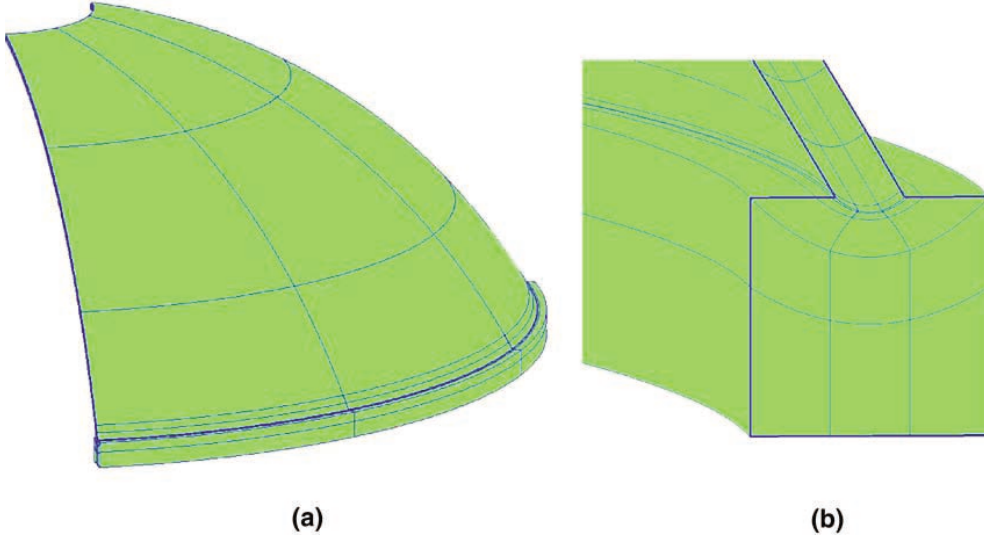
Fig. 52. Hemispherical shell with a stiffener: (a) coarse mesh and (b) detail of the stiffener.

Fig. 53. Hemispherical shell with a stiffener. Vertical displacement contours on the deformed configuration for refined, $p = 2$, solution (scaling factor of 500 used).

$$\sigma_{\mathrm{vm}} = \left(\frac{3}{2}\sigma'_{ij}\sigma'_{ij}\right)^{\frac{1}{2}}, \quad \sigma'_{ij} = \sigma_{ij} - \frac{1}{3}\sigma_{kk}\delta_{ij}, \tag{27}$$

where σ_{vm} is the von Mises stress, σ_{ij} is the stress tensor and σ'_{ij} is the deviatoric stress. The results in Figs. 53 and 54 came from a quadratic NURBS simulation. There is qualitative agreement with the results given in Rank et al. [10]. The Euclidean norm of the displacement and the von Mises stress were calculated at points A–D, identified in Fig. 51. Since points A–D are located on the boundary of the computational domain, values of σ_{ij} imposed by the boundary conditions were used in (27), where applicable, as opposed to the computed quantities. Results for $p = 2$ to $p = 6$ are plotted in Figs. 55–62 along with results from Rank et al. [10] for their finest simulation, that is, $p = 8$, which we take as a reference. Good agreement in the converged displacements is observed. Figs. 56 and 57 indicate that at points B and C, the displacements for the NURBS discretization converge from the "stiff" side (consistent with the results of the previous section). Agreement in converged values of the von Mises stress is also apparent.
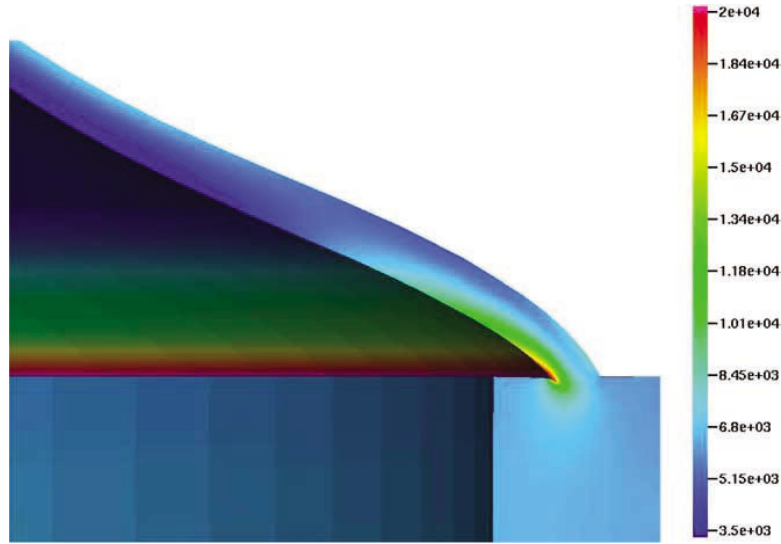
Fig. 54. Hemispherical shell with a stiffener. von Mises stress contours on the deformed configuration for refined, $p = 2$, solution (scaling factor of 500 used).
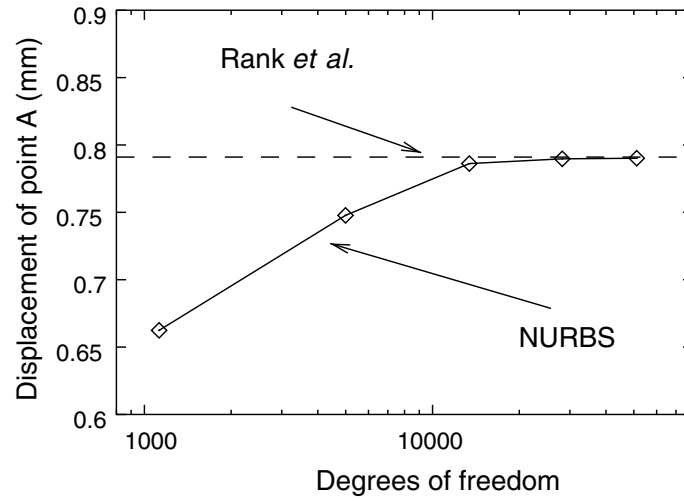
Fig. 55. Hemispherical shell with a stiffener. Convergence of the Euclidean norm of the displacement at point A.
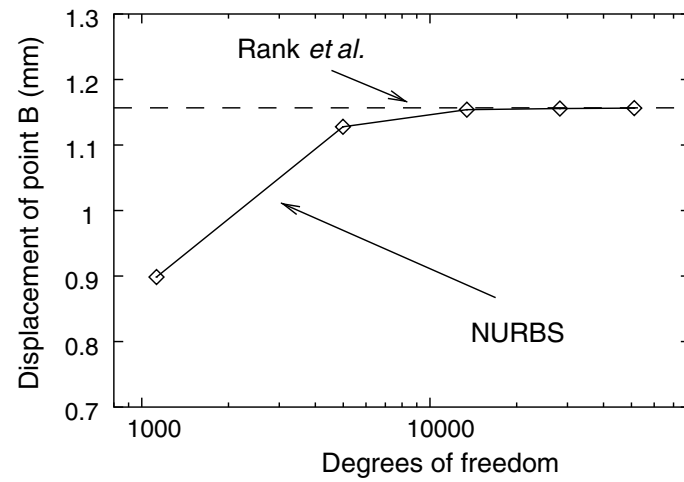
Fig. 56. Hemispherical shell with a stiffener. Convergence of the Euclidean norm of the displacement at point B.
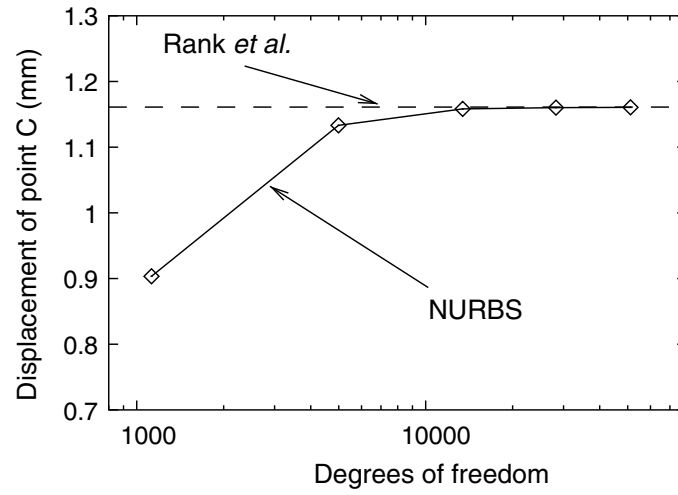
Fig. 57. Hemispherical shell with a stiffener. Convergence of the Euclidean norm of the displacement at point C.
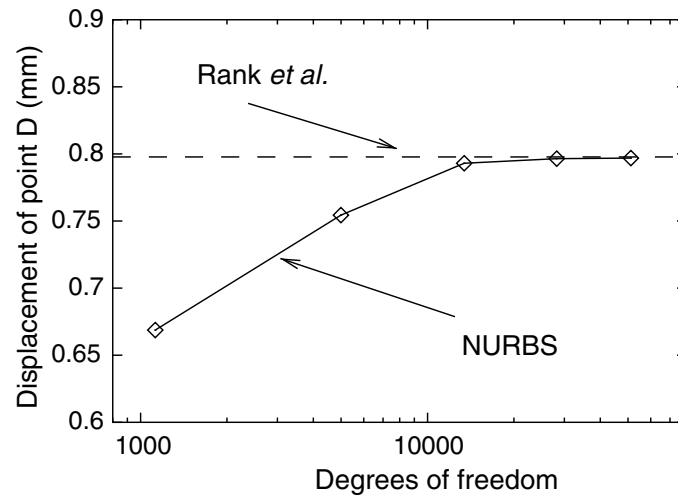
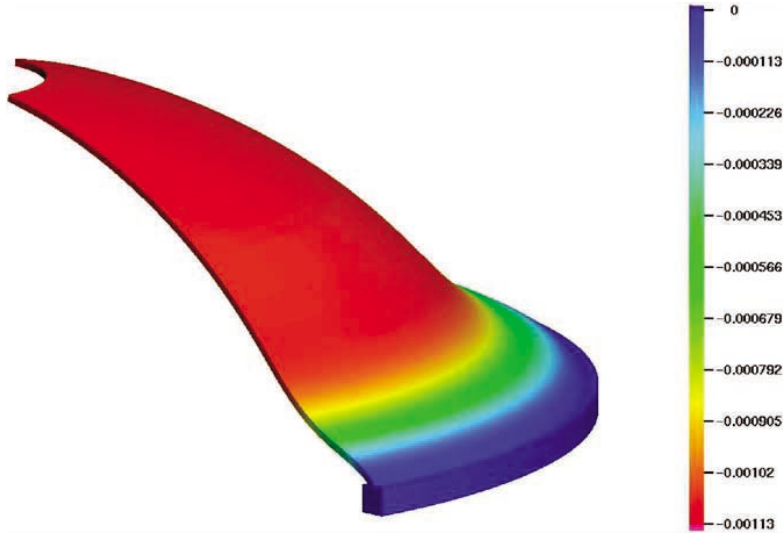Fig. 58. Hemispherical shell with a stiffener. Convergence of the Euclidean norm of the displacement at point D.
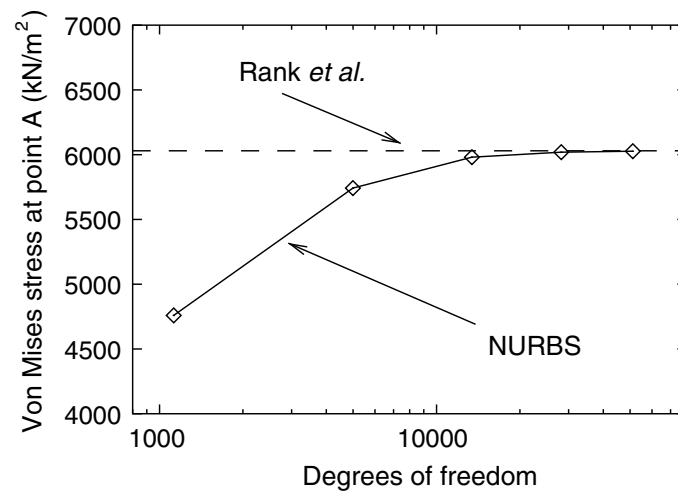
Fig. 59. Hemispherical shell with a stiffener. Convergence of the von Mises stress at point A.
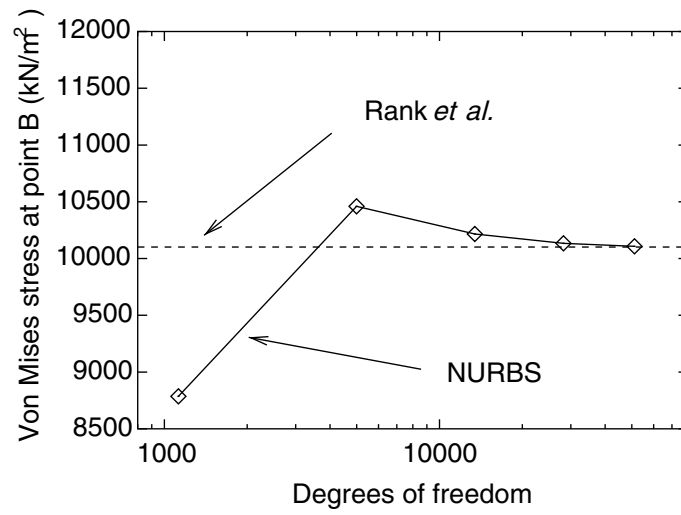
Fig. 60. Hemispherical shell with a stiffener. Convergence of the von Mises stress at point B.
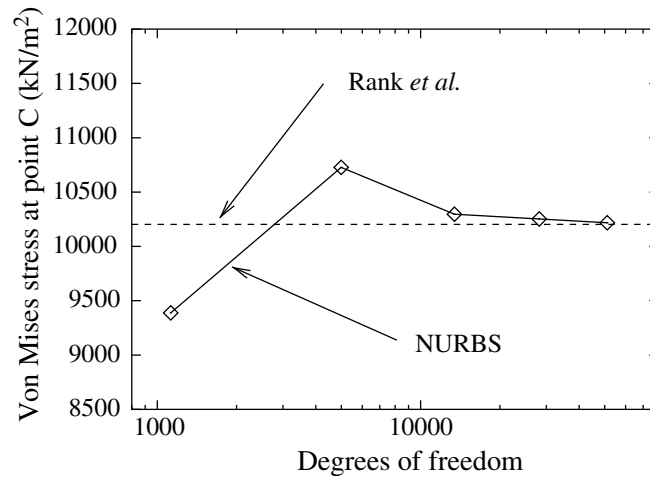
Fig. 61. Hemispherical shell with a stiffener. Convergence of the von Mises stress at point C.
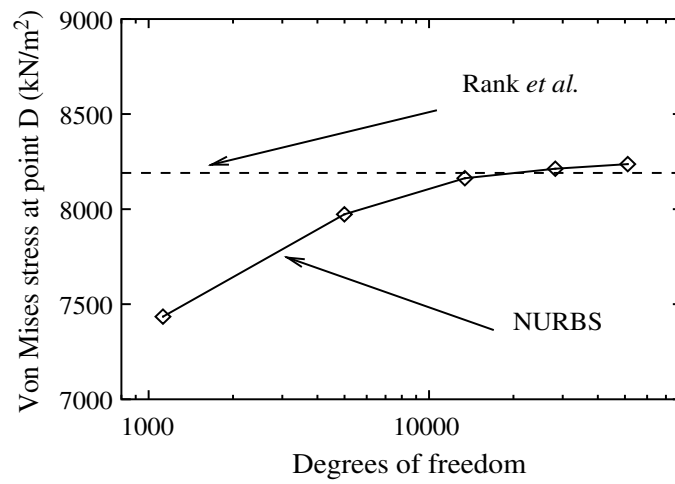
Fig. 62. Hemispherical shell with a stiffener. Convergence of the von Mises stress at point D.

5. Isogeometric fluids analysis

We wish to perform preliminary assessments of isogeometric analysis in fluids. The fluid equations are fundamentally different than the structural equations in that advection dominated flow phenomena are characterized by essentially skew-symmetric differential operators, the antithesis of the symmetric, definite operators which characterize typical structural analysis models. Another complicated flow feature is sharp layers involving very strong gradients. Within these layers, diffusive behavior prevails and the equations transition to being predominantly symmetric-definite. As a result, successful fluid analysis formulations need to automatically account for the local competition between advective and diffusive effects, and adjust the discretization accordingly. The standard Galerkin formulation optimally treats symmetric-definite operators but produces unstable discretizations of skew-symmetric operators. This is well known in finite element analysis and, unfortunately, carries over to the NURBS-based approach. The first finite element methodology exhibiting uniform stability and convergence behavior across the full range of advective and diffusive phenomena was SUPG, a "stabilized method" which had its origins in the late 1970s and early 1980s. (A good summary of the very early literature is contained in Brooks and Hughes [54]. Recent, state-of-the-art literature on stabilized methods is presented in [55–66].) SUPG is a residual-based modification of the Galerkin method which increases stability without degrading accuracy. It is sufficient for most flow phenomena but it is not a monotone technology because its higher-order accuracy normally precludes monotone behavior except in sufficiently resolved cases. There is a well known "barrier theorem" that asserts that the class of higher-order accurate, linear difference methods cannot be monotone, and SUPG falls into that class. Consequently, SUPG by itself is not a sufficiently robust technology for shock wave phenomena in fluids but in combination with so-called "discontinuity-capturing" operators has proved to be an industrial-strength technology for applications with shocks (see [67]). SUPG possesses a rich mathematical theory and there is an extensive literature on its mathematical properties beginning with Johnson et al. [68]; see Hughes et al. [69] and references therein for a review of the literature. The identification of stabilized methods as multiscale methods was made in Hughes [70] and Hughes et al. [71]. These works also explored the link between stabilized methods and subgrid-scale modeling [69], an important direction in current research.

In this section we investigate the ability of the isogeometric approach, in conjunction with SUPG, to solve challenging test cases for the advection–diffusion equation. Isogeometric analysis is fundamentally a higher-order approach and one would not expect good behavior in situations with unresolved interior and boundary layers. In fact, the Gibbs phenomena noted for polynomial-based finite element methods tends to become more pronounced as polynomial order is increased. This is the reason that most practical fluids formulations employ lower-order, typically constant and linear, interpolation of flow variables. However, the variation diminishing property of Dirichlet boundary condition specification, plus the notion of k-refinement, leads to some remarkable results in the case of NURBS. This is illustrated by the first example.

5.1. Advection skew to the mesh with outflow Dirichlet boundary conditions

The problem setup is described in Fig. 63. Here, a is the advective velocity magnitude, κ is the diffusivity, and L is the side length of the domain. The Peclet number, $Pe = aL/\kappa = 10^6$. When this number is greater than one, advection dominates and diffusion is only important in very small layers. In the present case, diffusion is important in a region of thickness $\mathrm{O}(Pe^{-1}\ln Pe)$ in the outflow boundary layers and $\mathrm{O}(Pe^{-1/2}\ln Pe)$ in the internal layer (see [72, p. 468]). In all calculations the mesh is uniform, consisting of a 20×20 grid of square elements, with element side length $h = 1/20 = 0.05$. Refinement is performed by the k-method, and solutions from $p = 1$ to $p = 12$ are calculated. In all cases, the standard SUPG formulation is used with $\tau = h_a/2a$, where h_a is the element length in the direction of the flow velocity which, in the present case, is simply, $h_a = h/\max\{\cos\theta, \sin\theta\}$.
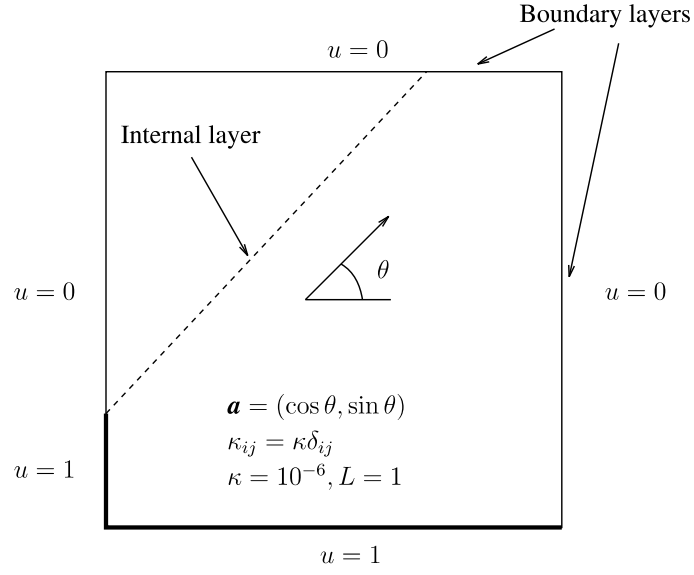
Fig. 63. Advection skew to mesh. Problem description and data.
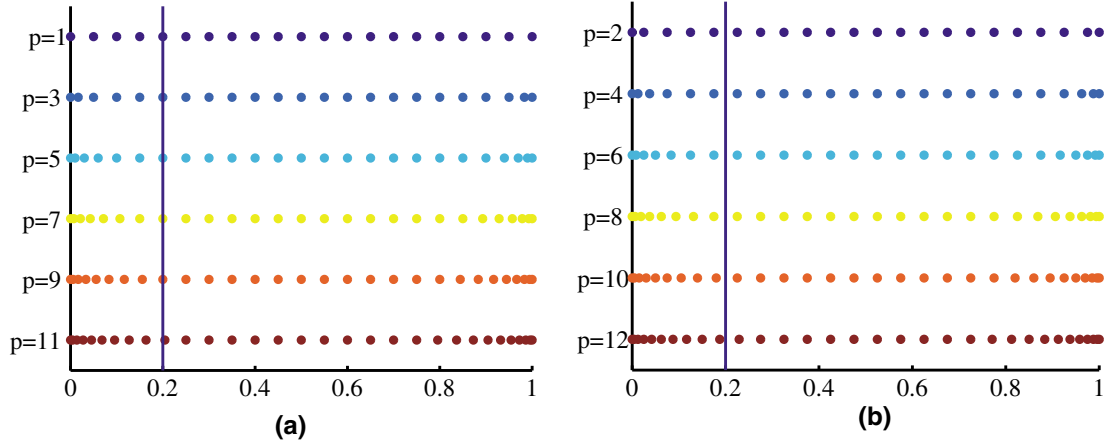
Fig. 64. The *y*-coordinate of the control points along the left edge of the domain: (a) odd polynomial orders and (b) even polynomial orders.
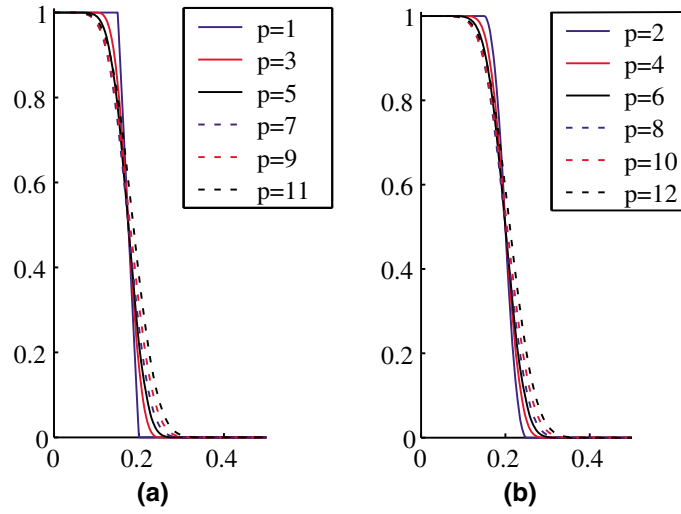
Fig. 65. Dirichlet boundary conditions along the left edge of the domain: (a) odd polynomial orders and (b) even polynomial orders.
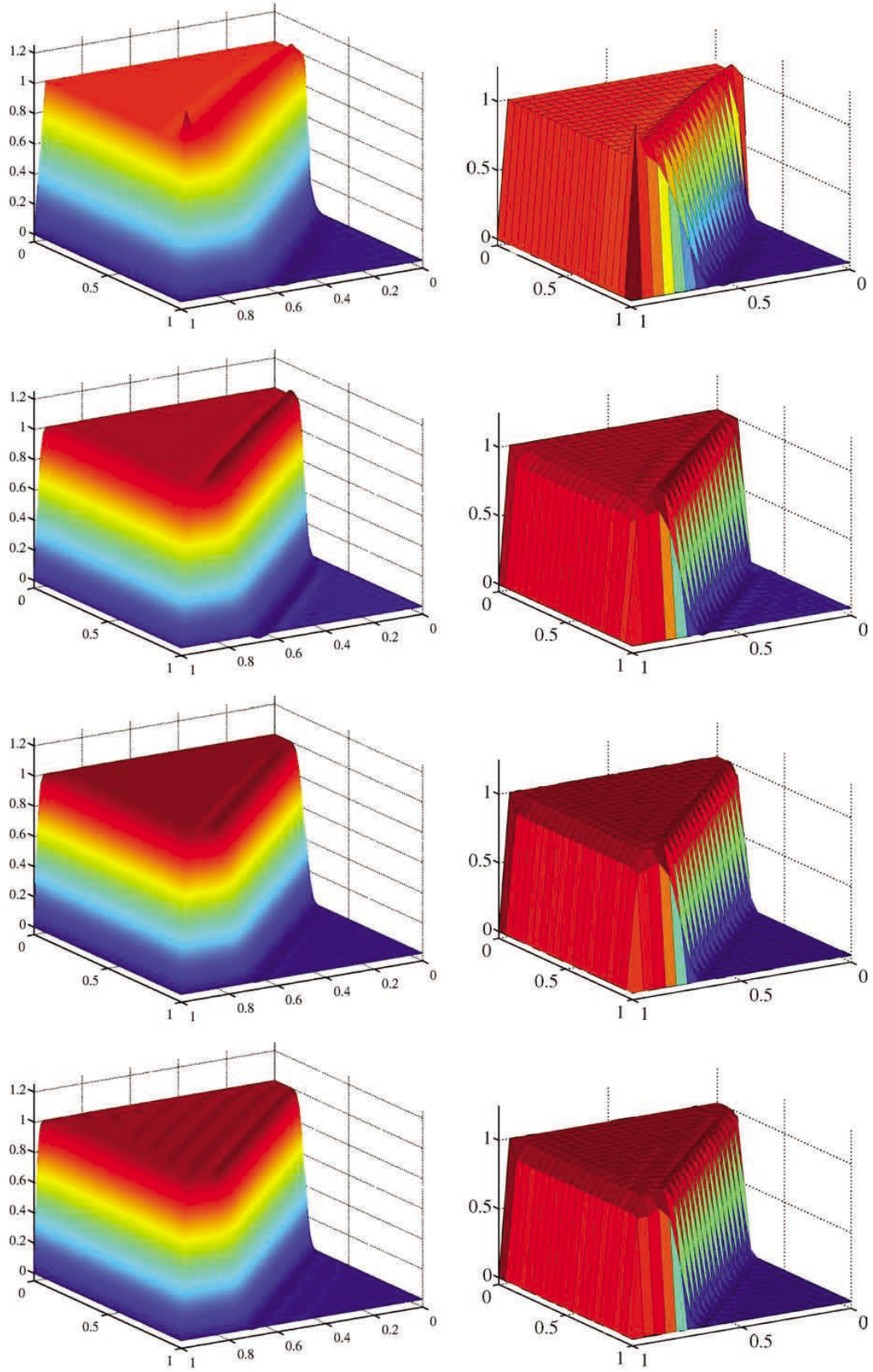
Fig. 66. Advection skew to the mesh, $\theta = 45°$. Top to bottom: results for $p = 1$ to $p = 4$. Left: plot with 100×100 points, Phong shaded. Right: plot with 21×21 points.
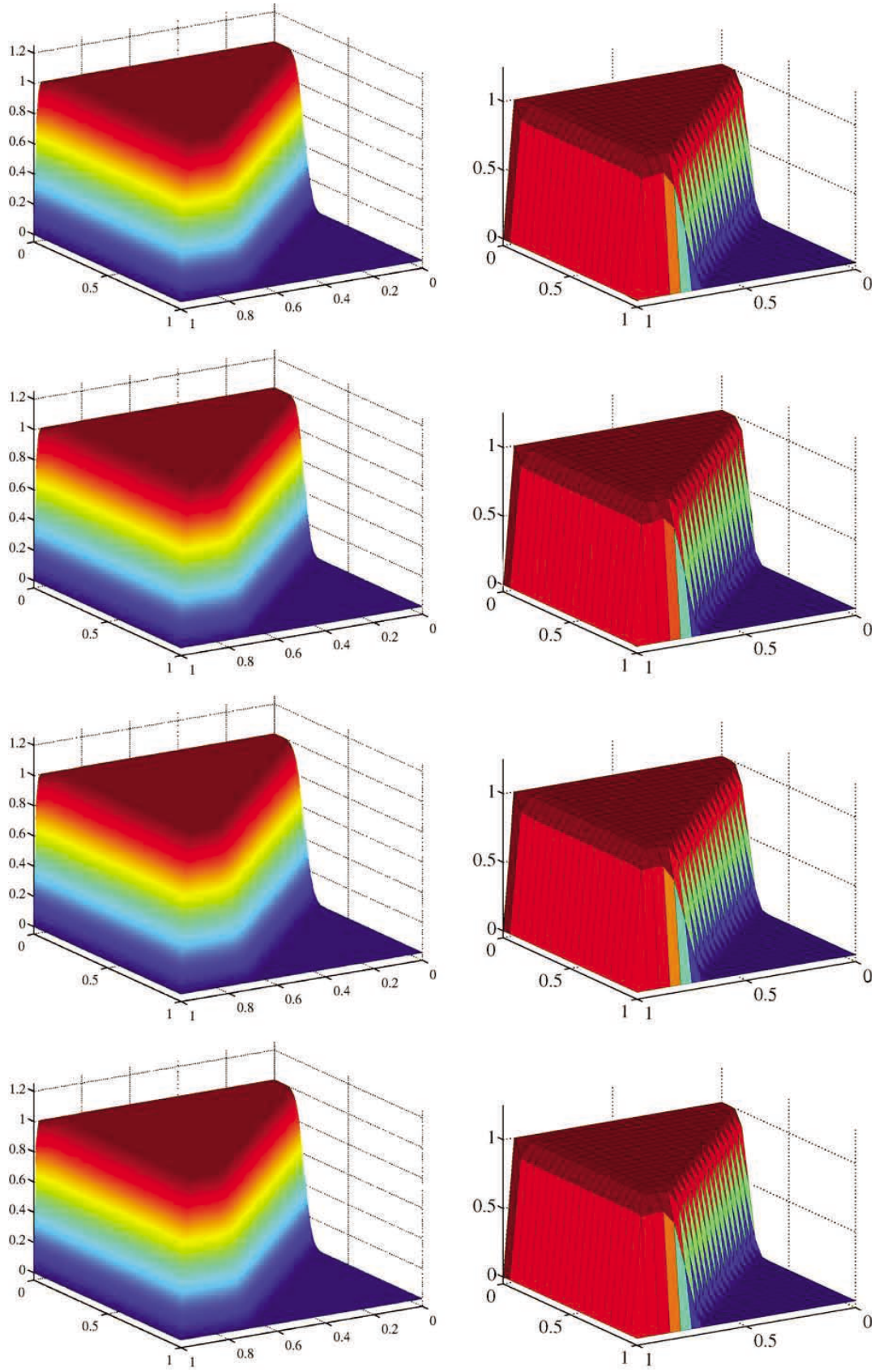
Fig. 67. Advection skew to the mesh, $\theta = 45°$. Top to bottom: results for $p = 5$ to $p = 8$. Left: plot with 100×100 points, Phong shaded. Right: plot with 21×21 points.
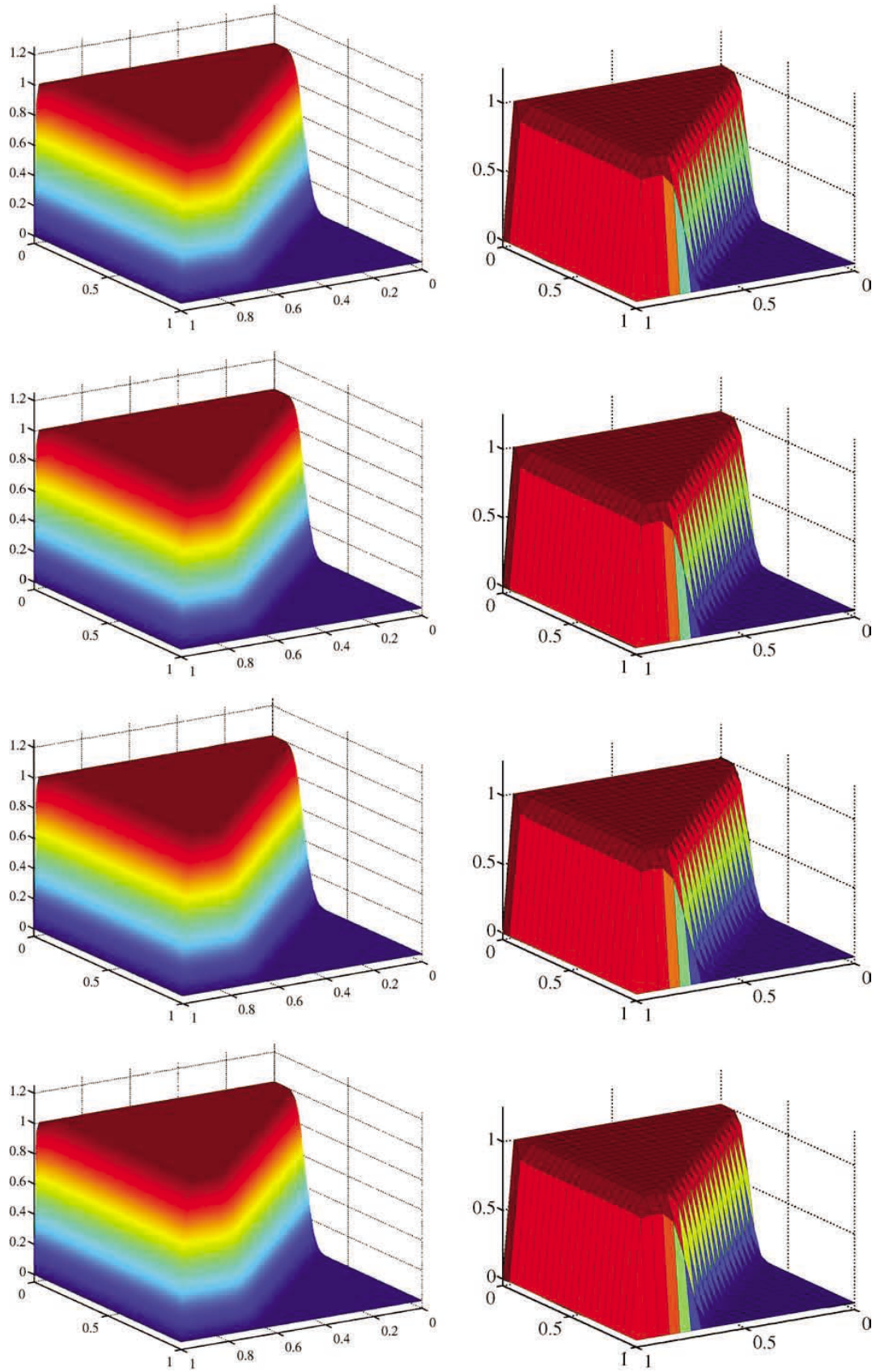
Fig. 68. Advection skew to the mesh, $\theta = 45°$. Top to bottom: results for $p = 9$ to $p = 12$. Left: plot with 100×100 points, Phong shaded. Right: plot with 21×21 points.

The boundary condition is set by specifying the control variables. On the top and right edges of the domain, all control variables are set to 0 and the boundary condition is exactly satisfied along these edges. On the bottom, the control variable corresponding to the lower right-hand corner is set to 0 and the remainder are set to 1. The result is that the boundary value is identically 1 up to the last element in which it smoothly decreases to 0 at the corner. The left-hand-side boundary is more interesting. If we think of our control variables as control points in $\mathbb{R}^3$ defining the surface plot of the solution, where the x and y coordinates have been fixed by the two-dimensional geometrical mapping and are no longer to be chosen by the user, then what we have done along the left side of our domain is to set the z-component (our actual control variable) equal to 1 for each control point that falls in the interval $[0, 0.2[$, and equal to 0 if it falls in $[0.2, 1]$. The locations of the control points are shown in Fig. 64. Note the clustering of points near the open knots. The resulting boundary conditions are shown in Fig. 65. For $p = 1$, the boundary condition is interpolated, whereas for $p > 1$ it is fit to the control variables in monotone fashion as the variation diminishing property of B-splines prevents the curve from over- and under-shooting. We wish to emphasize that k-refinement produces non-nested solution spaces, which prevents us from having the exact same boundary condition at each stage of the refinement process. As a result of this technique, the discontinuity "smears" about the location 0.2. For odd polynomial orders with $p \leqslant 9$, a control point falls directly on 0.2, whereas for even polynomial orders one does not (see Fig. 64). For this reason we have plotted the even and odd cases separately. When p is larger than 9, the "fringing" due to the open knot vectors disrupts this pattern. Clearly, a better scheme for setting the boundary control variables is required. One possibility is to obtain the desired boundary condition by setting up a projection problem over the boundary, for which a suitable norm needs to be chosen. Our next computational example will demonstrate that technique. Nevertheless, the main point of the present study is to assess the ability of NURBS (in this case, B-splines because of the simplicity of the domain) to deal with unresolved boundary and interior layers. We first consider the case in which $\theta = 45°$. The results for $p = 1$ to $p = 12$ are presented in Figs. 66–68. Two views are presented for each p, one in which the plotting routine sampled the solution with a 100×100 grid of uniformly distributed points and one in which it is sampled with a 21×21 uniform grid. In the former case the plot is Phong shaded and in the latter, it is represented by bilinear interpolation on each element and the element edges are drawn. The philosophy behind the dual views is that the 100×100 grid plots are a more faithful rendering of the higher-order cases, whereas the 21×21 point piecewise bilinear interpolates are the type of plots that have appeared in numerous research articles over the years and these may be more easily visually compared with results in the literature. The $p = 1$, $\theta = 45°$, case is unusual in that the selection of τ, the SUPG parameter, is the tuned value that captures the boundary layer very well. In general, the boundary layer is very poorly captured with $p = 1$
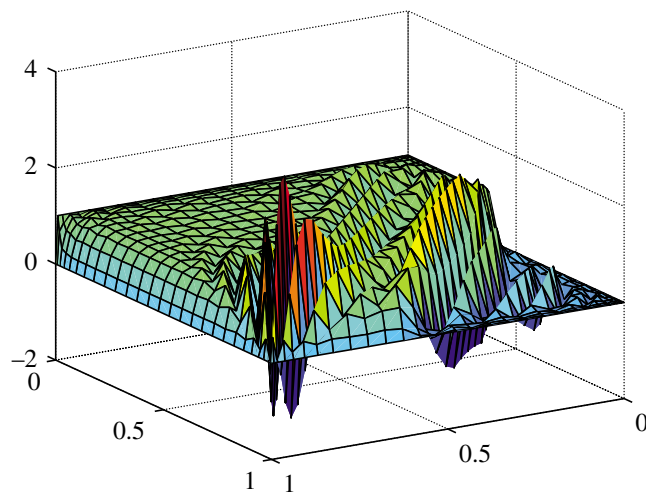
Fig. 69. Advection skew to the mesh, $\theta = 45°$. Control variables for $p = 8$.
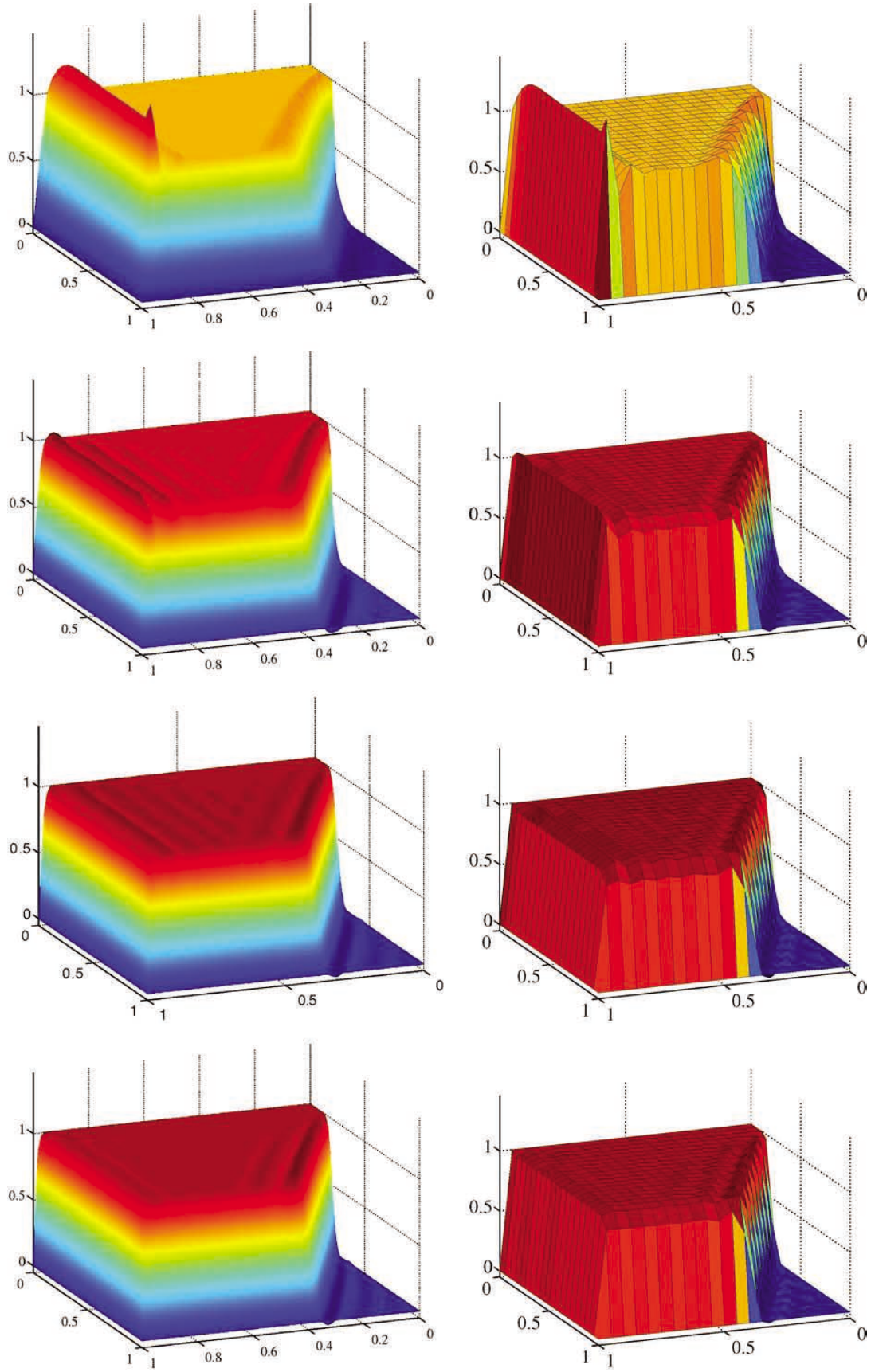
Fig. 70. Advection skew to the mesh, $\theta \approx 63.4°$. Top to bottom: results for $p = 1$ to $p = 4$. Left: plot with 100×100 points, Phong shaded. Right: plot with 21×21 points.

45
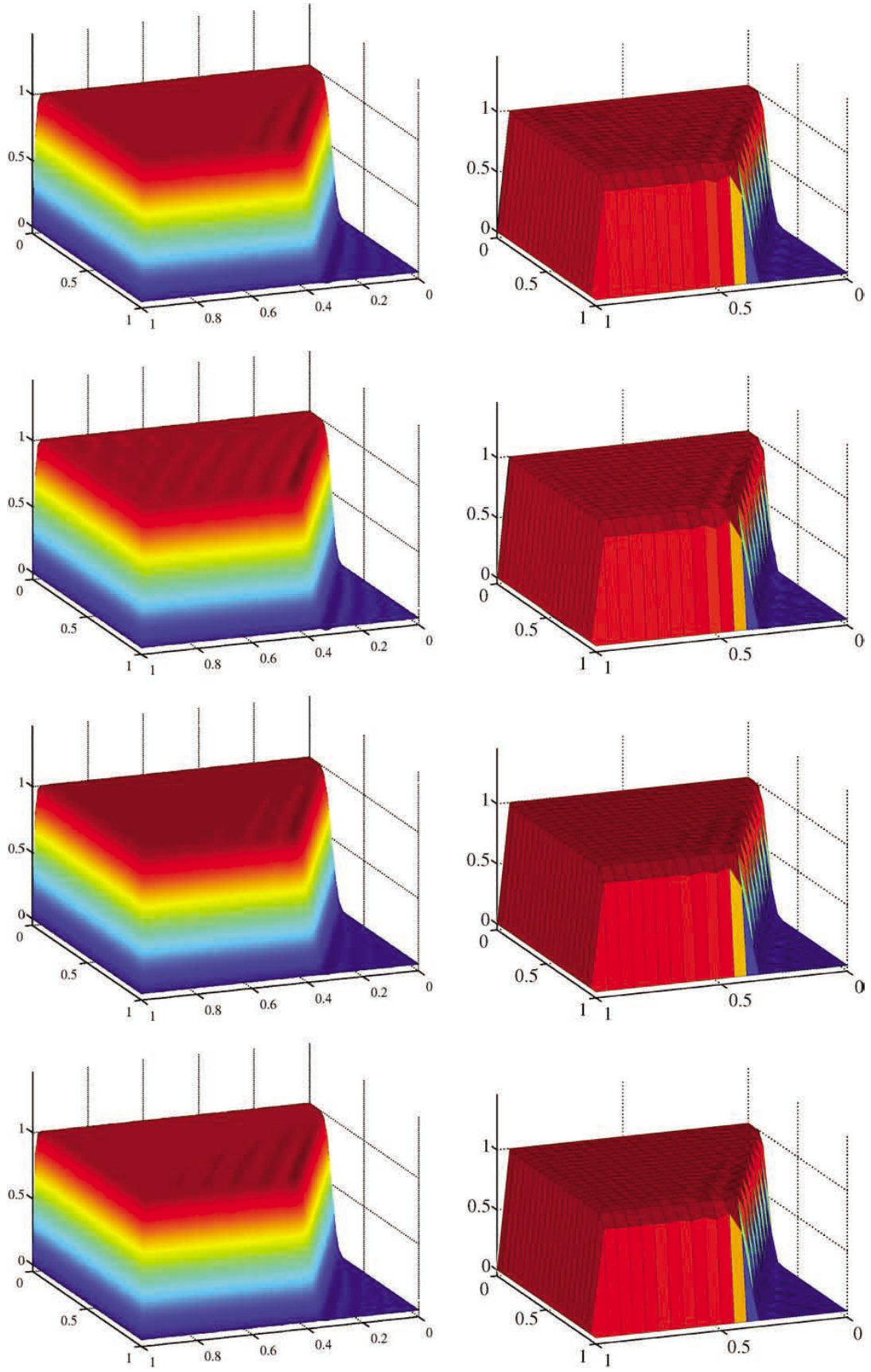
Fig. 71. Advection skew to the mesh, $\theta \approx 63.4°$. Top to bottom: results for $p = 5$ to $p = 8$. Left: plot with 100×100 points, Phong shaded. Right: plot with 21×21 points.
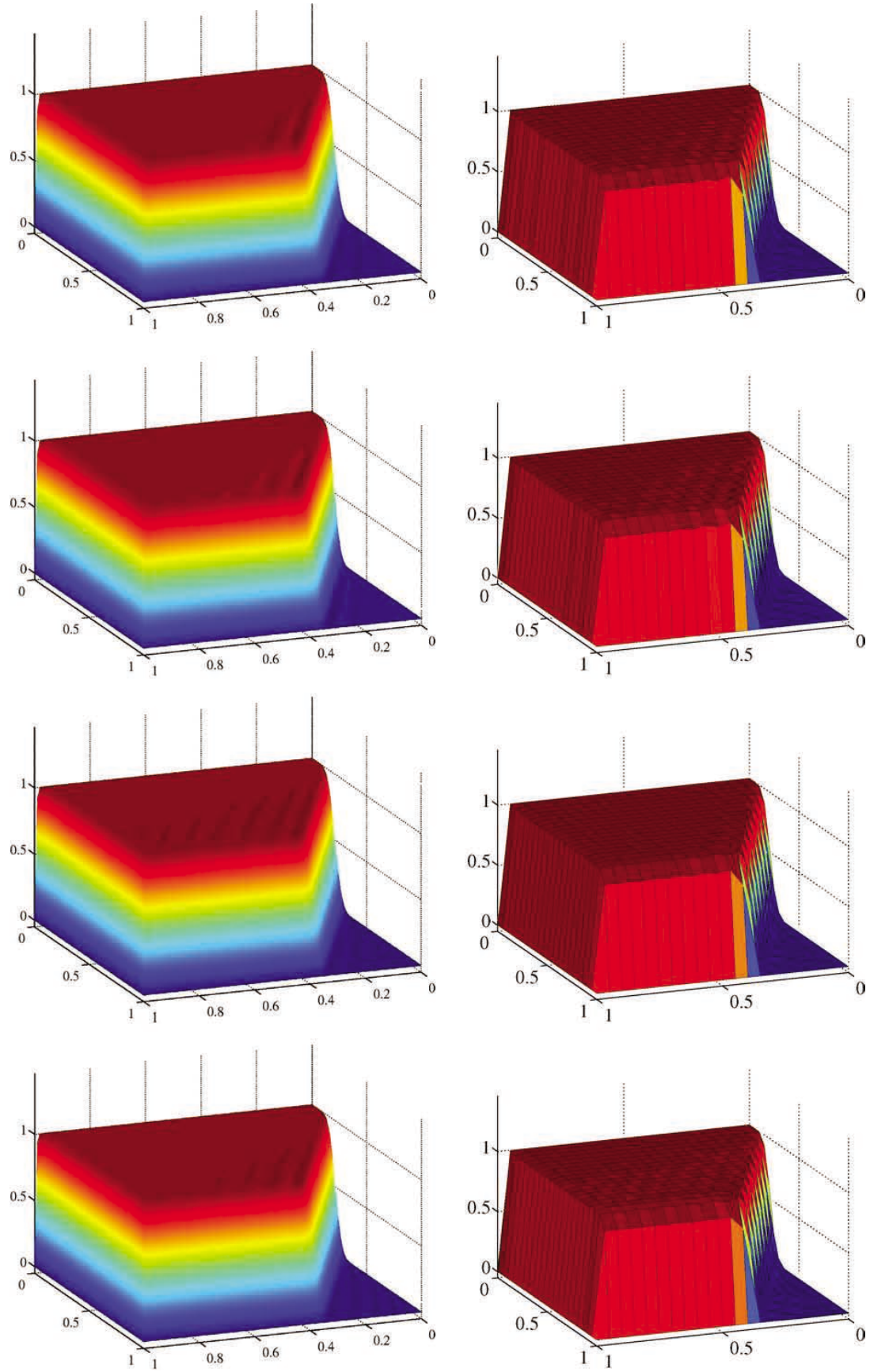
Fig. 72. Advection skew to the mesh, $\theta \approx 63.4°$. Top to bottom: results for $p = 9$ to $p = 12$. Left: plot with 100×100 points, Phong shaded. Right: plot with 21×21 points.

unless τ takes on a specific value. As one examines the results, it is clear that they improve as p increases and are converging toward monotone results with quite sharp layers. One would perhaps expect that oscillations would increase with increasing p but this is not the case. This is certainly due in part to the smearing of the boundary condition but we suspect that the high continuity of the basis obtained through k-refinement plays a part in this as well. Further studies need to be undertaken to clarify these issues.

It is interesting to view the control variables (see Fig. 69). The solution for $p = 8$ is nearly monotone, whereas the corresponding control variables are quite oscillatory. It is not clear what physical information is provided by the control variables. However, as the mesh is h-refined, the control net will approach the solution surface, that is, the basis becomes interpolatory in the limit as $h \rightarrow 0$. The 20×20 mesh is obviously far from this limit.

The $\theta = \tan^{-1}(2)$ case (i.e., $\theta \approx 63.4°$) is presented in Figs. 70–72. This problem represents possibly the most severe standard test for the advection–diffusion equation. Refinement is performed by the k-method, and solutions from $p = 1$ to $p = 12$ are calculated. This time, for $p = 1$, the overshoot in the outflow boundary layer is approximately 45% but, as the order is elevated by k-refinement, convergence toward monotone results occurs. This angle of flow seems to be somewhat more difficult than the $\theta = 45°$ case. Nevertheless, the same conclusion may be drawn: Sufficiently high-order k-refinement produces extraordinary results in dealing with sharp internal and boundary layers. The maximum percentage overshoot vs. polynomial order for both cases is plotted in Fig. 73. The convergence toward monotone solutions is apparent in both cases.
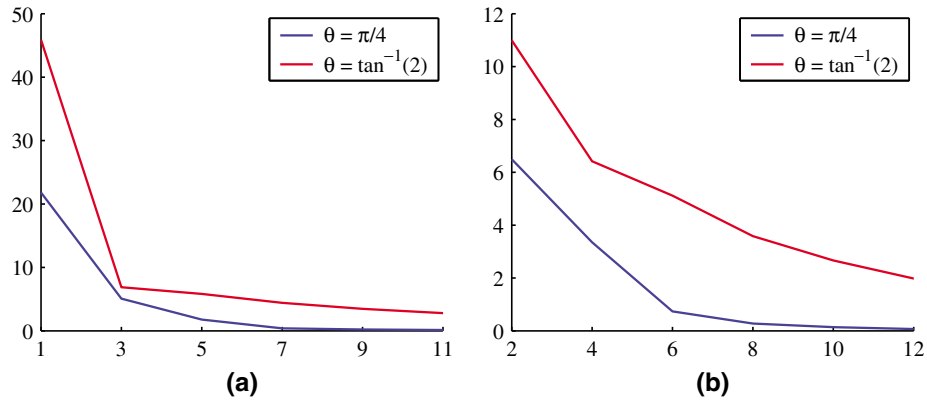
Fig. 73. Maximum solution over-shoot (%) vs. polynomial order: (a) odd polynomial orders and (b) even polynomial orders.
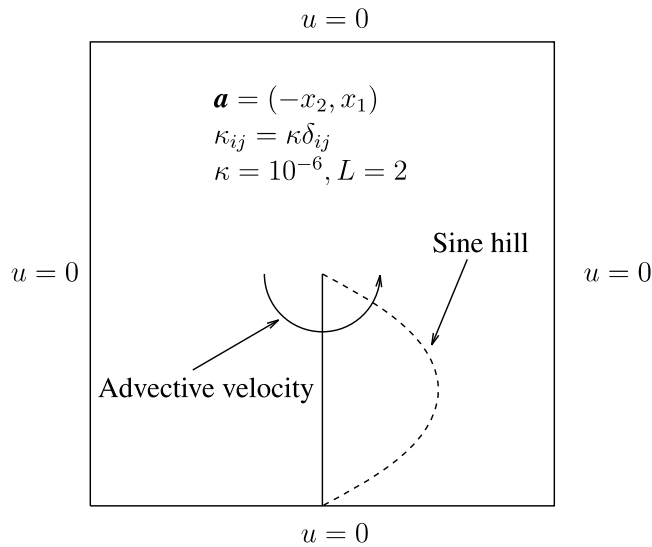
Fig. 74. Advection of a sine hill. Problem description and data.
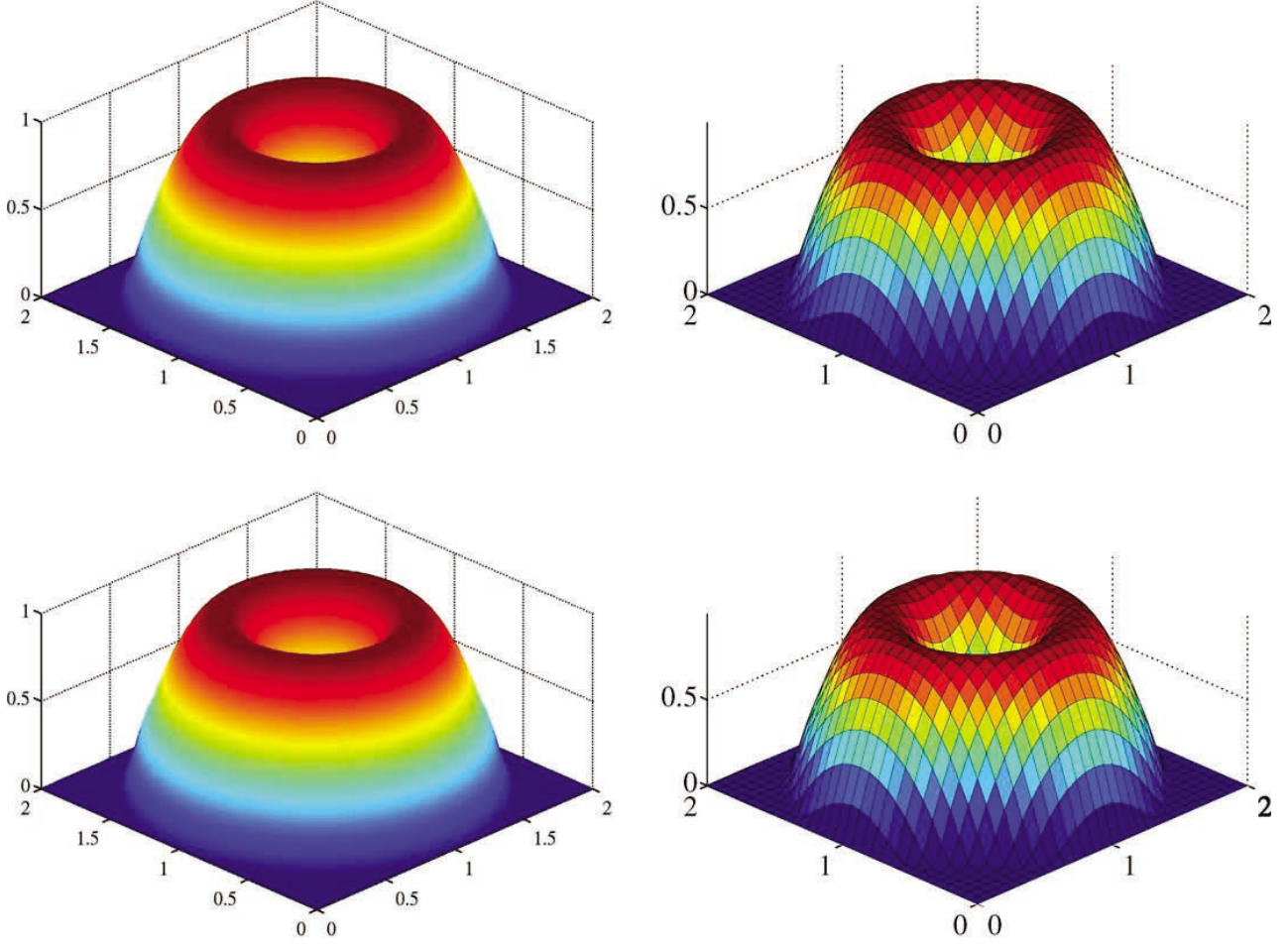
Fig. 75. Advection of a sine hill. Top to bottom: results for $p = 2$ and $p = 6$. Left: plot with 100×100 points, Phong shaded. Right: plot with 31×31 points.

5.2. Advection of a sine hill in a rotating flow field

The problem statement is shown in Fig. 74. The flow is circular about the center of a square domain $[-1, 1] \times [-1, 1]$ with velocity components given by $a_1 = -x_2$ and $a_2 = x_1$. Diffusivity is taken to be 10^{-6}, which makes the problem advection-dominated. Homogeneous essential boundary conditions are imposed around the perimeter of the domain, while the $u = -\sin(2\pi x)$ condition is imposed on the "slit," as shown in Fig. 74. Since the B-spline space does not contain trigonometric functions, H_0^1-minimization was used to set the control point values on the "slit". Computations were done on a 30×30 uniform mesh with $p = 2$ and $p = 6$. Continuity was maintained at the C^1 and C^5 levels, respectively, everywhere in the domain except along the lines $x = 0$ and $y = 0$ where C^0 continuity was maintained. This choice was made to facilitate imposition of boundary conditions. Results are presented in Fig. 75. The exact solution is essentially a pure advection of the boundary condition along the circular streamlines. Both $p = 2$ and $p = 6$ produce excellent results.

6. Conclusions

We have introduced a new analysis framework, called isogeometric analysis, which is based on NURBS. It has many similarities with finite element analysis, but also important differences. A key feature is to represent geometry exactly by NURBS elements and then invoke the isoparametric concept to define field variables, such as displacement, temperature, etc. The coarsest mesh encapsulates the exact geometry. This

means that mesh refinement is simply accomplished by reindexing the parametric space. The refinement process can proceed without interaction with the CAD system, a distinct advantage over finite element procedures. The fact that finite element mesh refinement strategies require interaction with the CAD system at each stage may be the reason that adaptive mesh refinement, despite its benefits, is still primarily an academic activity and one that has not significantly penetrated the industrial sector. There are NURBS analogues of finite element h-refinement and p-refinement, and there is also a variant of p-refinement, which we call k-refinement, in which the continuity of functions is systematically increased. This seems to have no analogue in traditional finite element analysis but is a feature shared by some meshless methods. The approach is fundamentally higher-order. For example, in order to represent circles, cylinders and spheres, rational polynomials of at least quadratic order are necessary. The generation of refined NURBS bases of all orders is facilitated by simple recursion relationships. The versatility and power of recursive NURBS basis representations are truly remarkable. Equation systems generated by NURBS tend to be more homogeneous than those generated by higher-order finite elements and this may have some benefit in equation solving strategies. NURBS satisfy a "variation diminishing" property. For example, they give monotone fits to discontinuous control data and become smoother as order is increased, unlike Lagrange interpolatory polynomials which oscillate more violently as order is increased. NURBS of all orders are non-negative pointwise. This means that every entry of the NURBS mass matrix is non-negative. These properties are not attained in finite element analysis. On the other hand, NURBS are not interpolatory. They are fit to nets of control points and control variables. This aspect is less transparent to deal with than the corresponding finite element concepts of interpolated nodal points and nodal variables but somewhat similar to the situation for meshless methods.

We performed initiatory calculations with isogeometric analysis procedures on some linear structural and fluids problems. h-, p-, and k-refinement strategies were investigated. In the structural applications, convergence to exact solutions was shown to occur at optimal rates for two- and three-dimensional solids. We performed shell analysis with three-dimensional solid NURBS elements. Convergence to thin shell solutions and benchmark results occurred in all cases. The ability to accurately resolve shell boundary layers appears to be particularly noteworthy.

We applied a stabilized SUPG formulation of isogeometric analysis to standard test cases involving the advection–diffusion equation in order to assess its potential for fluid dynamics. A stringent two-dimensional test case, advection skew to a mesh with outflow Dirichlet boundary conditions, was studied on a fixed mesh using k-refinement. As order was increased, solutions improved and converged toward monotone results with sharp layers. This surprising result seems to contradict numerical analysis preconceptions about high-order accurate methods.

There seem to be numerous opportunities for research in isogeometric analysis. Some areas of particular interest are: (1) integrating isogeometric analysis with CAD systems; (2) developing isogeometric mesh generators based on state-of-the-art technologies such as CUBIT (see [73]); (3) developing triangular and tetrahedral isogeometric elements and associated unstructured meshing techniques (see [74]); (4) developing procedures which account for trimmed surface description (this is a significant challenge, the approach of Höllig [75] may be useful, see Fig. 76); (5) developing isogeometric surface elements for traditional and rotationless shell analysis [17–19,49–53] and boundary integral formulations, and combining isogeometric surface descriptions with meshless methods (exact surface descriptions would seem to be particularly easy to generate from CAD descriptions); (6) applying isogeometric technology to thin plate and shell theories, strain-gradient theories, and other higher-order theories; (7) developing hierarchical isogeometric bases for multiscale analysis; (8) applying isogeometric analysis to fluid mechanics and, in particular, boundary layer turbulence (the pioneering studies [76–79] should be mentioned in this regard); (9) applying isogeometric elements to problems of engineering interest; (10) developing a mathematical theory of convergence and error analysis (we have already made some headway on this topic and out initial results will be reported upon in a forthcoming publication); (11) applying isogeometric analysis to contact problems and,
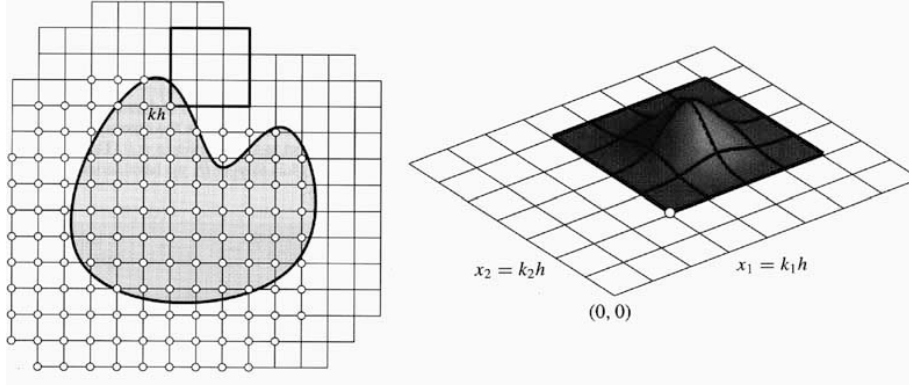
Fig. 76. Höllig [75] has developed finite element methods incorporating a B-spline basis on a rectangular grid. The object's actual geometry is introduced through so-called weighting functions. The idea may be useful in dealing directly with trimmed surfaces.
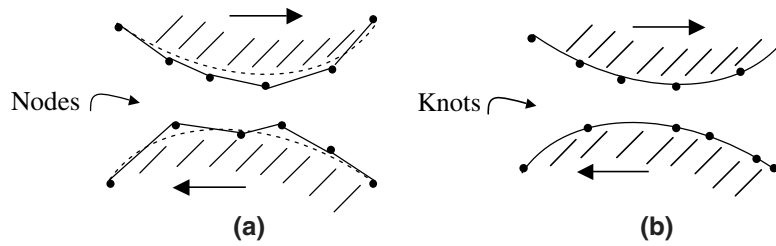
Fig. 77. (a) Faceted polynomial finite elements create problems in sliding contact (see [80,81]). (b) NURBS geometries can attain smoothness of real bodies.

in particular, frictional contact and sliding (see Fig. 77); (12) developing isogeometric dynamic analysis, including time-harmonic and explicit, implicit, and space–time time-stepping approaches (see Fig. 78); (13) applying isogeometric analysis to shape optimization (see Fig. 79); (14) generalizing to other important application areas, such as, for example, acoustics and electromagnetics; (15) various non-linear applications, etc.

In conclusion, we believe the isogeometric approach has considerable potential in practical problem solving and is a promising alternative to current analysis procedures.
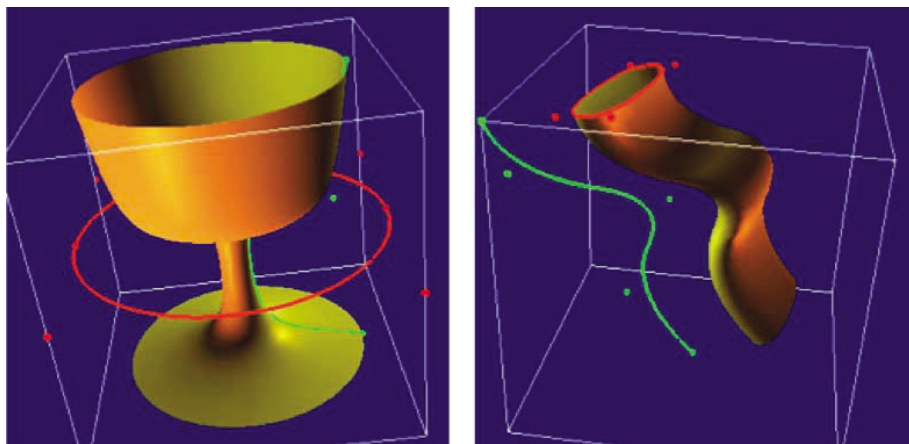
Fig. 78. D-NURBS (Dynamic NURBS) have been proposed by Qin and Terzopoulos [82] for animating objects defined by NURBS surfaces.

51
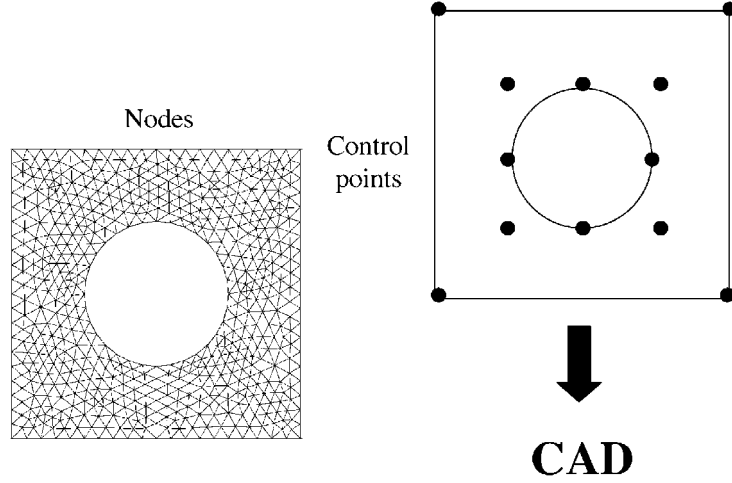
Fig. 79. Isogeometric shape optimization. Left: the finite element geometric description is inconvenient for shape optimization. Right: NURBS present a more concise parameterization of design variables. Optimized NURBS descriptions can be returned to CAD systems for manufacturing, a distinct advantage over finite elements.

Acknowledgement

Appendix A. Control data for NURBS objects

A.1. Circular curve

A circular curve can be described in many different ways using NURBS. The approach shown in Fig. 12, utilizing four rational quadratic segments, is one of the simplest. Given the knot vector

$$\Xi = \{0, 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 4\} \tag{A.1}$$

we use (1) and (2) to generate the B-spline basis functions (with $p = 2$) shown in Table A.1. The curve itself is then generated using (11) and (12) together with the control points and weights given in Table A.2. Note

Table A.1
One-dimensional quadratic basis functions for creating circular NURBS curve

Function	[0, 1]	[1, 2]	[2, 3]	[3, 4]
$N_{1,2}(\xi)$	$(1 - \xi)^2$	0	0	0
$N_{2,2}(\xi)$	$2\xi(1 - \xi)$	0	0	0
$N_{3,2}(\xi)$	ξ^2	$(2 - \xi)^2$	0	0
$N_{4,2}(\xi)$	0	$2(\xi - 1)(2 - \xi)$	0	0
$N_{5,2}(\xi)$	0	$(\xi - 1)^2$	$(3 - \xi)^2$	0
$N_{6,2}(\xi)$	0	0	$2(\xi - 2)(3 - \xi)$	0
$N_{7,2}(\xi)$	0	0	$(\xi - 2)^2$	$(4 - \xi)^2$
$N_{8,2}(\xi)$	0	0	0	$2(\xi - 3)(4 - \xi)$
$N_{9,2}(\xi)$	0	0	0	$(\xi - 3)^2$

Table A.2
Control points and weights for a circle of radius 1

i	B_i	w_i
1	$(1,0)$	1
2	$(1,1)$	$1/\sqrt{2}$
3	$(0,1)$	1
4	$(-1,1)$	$1/\sqrt{2}$
5	$(-1,0)$	1
6	$(-1,-1)$	$1/\sqrt{2}$
7	$(0,-1)$	1
8	$(1,-1)$	$1/\sqrt{2}$
9	$(1,0)$	1

that scaling all of the weights by a constant factor does not change the final curve. In Fig. 12a, the weights (and, correspondingly, the projective control points) have all been scaled by 3 for visual clarity.

A.2. Toroidal surface

The toroidal surface shown in Fig. 13b is created by sweeping a circular NURBS curve of radius 1, as defined in the previous section, around a circular path of radius 5, also defined by a NURBS curve. In this instance, the η coordinate describes our initial circular curve and the ξ coordinate traverses the path along which we sweep. The basis is rational quadratic in each of the two directions, with knot vectors given by

$$\Xi = \{0, 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 4\} \tag{A.2}$$

and

$$\mathscr{H} = \{0, 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 4\}. \tag{A.3}$$

As Ξ is the same as in our previous example, the set of functions, $\{N_{i,2}(\xi)\}$, are the same as in Table A.1. As $\Xi = \mathscr{H}$, the set $\{M_{j,2}(\eta)\}$ have the same form, but with independent variable η rather than ξ.

The control net for the torus may be thought of as given by "sweeping" the control polygon for a circle of radius 1 about the control polygon for a circle of radius 5. See Fig. 13a. The resulting set of control points is given in Table A.3, with weights in Table A.4. The rational surface basis functions are formed from the one-dimensional basis functions and weights using (13).

Table A.3
Control points for toroidal surface

i	$B_{i,1}$	$B_{i,2}$	$B_{i,3}$	$B_{i,4}$	$B_{i,5}$	$B_{i,6}$	$B_{i,7}$	$B_{i,8}$	$B_{i,9}$
1	$(5,0,-1)$	$(6,0,-1)$	$(6,0,0)$	$(6,0,1)$	$(5,0,1)$	$(4,0,1)$	$(4,0,0)$	$(4,0,-1)$	$(5,0,-1)$
2	$(5,5,-1)$	$(6,6,-1)$	$(6,6,0)$	$(6,6,1)$	$(5,5,1)$	$(4,4,1)$	$(4,4,0)$	$(4,4,-1)$	$(5,5,-1)$
3	$(0,5,-1)$	$(0,6,-1)$	$(0,6,0)$	$(0,6,1)$	$(0,5,1)$	$(0,4,1)$	$(0,4,0)$	$(0,4,-1)$	$(0,5,-1)$
4	$(-5,5,-1)$	$(-6,6,-1)$	$(-6,6,0)$	$(-6,6,1)$	$(-5,5,1)$	$(-4,4,1)$	$(-4,4,0)$	$(-4,4,-1)$	$(-5,5,-1)$
5	$(-5,0,-1)$	$(-6,0,-1)$	$(-6,0,0)$	$(-6,0,1)$	$(-5,0,1)$	$(-4,0,1)$	$(-4,0,0)$	$(-4,0,-1)$	$(-5,0,-1)$
6	$(-5,-5,-1)$	$(-6,-6,-1)$	$(-6,-6,0)$	$(-6,-6,1)$	$(-5,-5,1)$	$(-4,-4,1)$	$(-4,-4,0)$	$(-4,-4,-1)$	$(-5,-5,-1)$
7	$(0,-5,-1)$	$(0,-6,-1)$	$(0,-6,0)$	$(0,-6,1)$	$(0,-5,1)$	$(0,-4,1)$	$(0,-4,0)$	$(0,-4,-1)$	$(0,-5,-1)$
8	$(5,-5,-1)$	$(6,-6,-1)$	$(6,-6,0)$	$(6,-6,1)$	$(5,-5,1)$	$(4,-4,1)$	$(4,-4,0)$	$(4,-4,-1)$	$(5,-5,-1)$
9	$(5,0,-1)$	$(6,0,-1)$	$(6,0,0)$	$(6,0,1)$	$(5,0,1)$	$(4,0,1)$	$(4,0,0)$	$(4,0,-1)$	$(5,0,-1)$

Table A.4
Weights for toroidal surface

i	$w_{i,1}$	$w_{i,2}$	$w_{i,3}$	$w_{i,4}$	$w_{i,5}$	$w_{i,6}$	$w_{i,7}$	$w_{i,8}$	$w_{i,9}$
1	1	$1/\sqrt{2}$	1	$1/\sqrt{2}$	1	$1/\sqrt{2}$	1	$1/\sqrt{2}$	1
2	$1/\sqrt{2}$	1/2	$1/\sqrt{2}$	1/2	$1/\sqrt{2}$	1/2	$1/\sqrt{2}$	1/2	$1/\sqrt{2}$
3	1	$1/\sqrt{2}$	1	$1/\sqrt{2}$	1	$1/\sqrt{2}$	1	$1/\sqrt{2}$	1
4	$1/\sqrt{2}$	1/2	$1/\sqrt{2}$	1/2	$1/\sqrt{2}$	1/2	$1/\sqrt{2}$	1/2	$1/\sqrt{2}$
5	1	$1/\sqrt{2}$	1	$1/\sqrt{2}$	1	$1/\sqrt{2}$	1	$1/\sqrt{2}$	1
6	$1/\sqrt{2}$	1/2	$1/\sqrt{2}$	1/2	$1/\sqrt{2}$	1/2	$1/\sqrt{2}$	1/2	$1/\sqrt{2}$
7	1	$1/\sqrt{2}$	1	$1/\sqrt{2}$	1	$1/\sqrt{2}$	1	$1/\sqrt{2}$	1
8	$1/\sqrt{2}$	1/2	$1/\sqrt{2}$	1/2	$1/\sqrt{2}$	1/2	$1/\sqrt{2}$	1/2	$1/\sqrt{2}$
9	1	$1/\sqrt{2}$	1	$1/\sqrt{2}$	1	$1/\sqrt{2}$	1	$1/\sqrt{2}$	1

A.3. Plate with circular hole

The two-element coarse mesh shown in Fig. 17a and defined by the control net in Fig. 17b has an interesting feature. The circle requires using a rational quadratic basis, but this in turn necessitates introducing a discontinuity in the first derivative to create the opposite corner. This could be done either by repeating knots and thus creating a basis that is only C^0 along the line between the initial two elements, or by placing two control points at the same location in physical space. We chose the latter option in order to ensure that our basis had C^1 continuity throughout the interior of the domain. The only place where the basis is not C^1 is on the boundary itself, at the location of the repeated control points. As stated previously, this mesh uses knot vectors

$$\Xi = \{0, 0, 0, 0.5, 1, 1, 1\} \tag{A.4}$$

Table A.5
Quadratic basis functions in ξ-direction for the quarter plate with circular hole

Function	[0, 0.5]	[0.5, 1]
$N_{1,2}(\xi)$	$(2\xi - 1)^2$	0
$N_{2,2}(\xi)$	$2\xi(2 - 3\xi)$	$2(\xi - 1)^2$
$N_{3,2}(\xi)$	$2\xi^2$	$2(3\xi - 1)(1 - \xi)$
$N_{4,2}(\xi)$	0	$(2\xi - 1)^2$

Table A.6
Quadratic basis functions in η-direction for the quarter plate with circular hole

Function	[0, 1]
$M_{1,2}(\eta)$	$(1 - \eta)^2$
$M_{2,2}(\eta)$	$2\eta(1 - \eta)$
$M_{3,2}(\eta)$	η^2

Table A.7
Control net for the plate with a circular hole

i	$B_{i,1}$	$B_{i,2}$	$B_{i,3}$
1	$(-1, 0)$	$(-2.5, 0)$	$(-4, 0)$
2	$(-1, \sqrt{2} - 1)$	$(-2.5, 0.75)$	$(-4, 4)$
3	$(1 - \sqrt{2}, 1)$	$(-0.75, 2.5)$	$(-4, 4)$
4	$(0, 1)$	$(0, 2.5)$	$(0, 4)$

Table A.8
Weights for the plate with a circular hole

i	$w_{i,1}$	$w_{i,2}$	$w_{i,3}$
1	1	1	1
2	$(1 + 1/\sqrt{2})/2$	1	1
3	$(1 + 1/\sqrt{2})/2$	1	1
4	1	1	1

Table A.9
Control points and weights for cylindrical solid

i	j	$B_{i,j,1}$	$B_{i,j,2}$	$B_{i,j,3}$	$w_{i,j,1}$	$w_{i,j,2}$	$w_{i,j,3}$
1	1	$(1,0,0)$	$(1,0,2.5)$	$(1,0,5)$	1	1	1
1	2	$(4/3,0,0)$	$(4/3,0,2.5)$	$(4/3,0,5)$	1	1	1
1	3	$(5/3,0,0)$	$(5/3,0,2.5)$	$(5/3,0,5)$	1	1	1
1	4	$(2,0,0)$	$(2,0,2.5)$	$(2,0,5)$	1	1	1
2	1	$(1,1,0)$	$(1,1,2.5)$	$(1,1,5)$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
2	2	$(4/3,4/3,0)$	$(4/3,4/3,2.5)$	$(4/3,4/3,5)$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
2	3	$(5/3,5/3,0)$	$(5/3,5/3,2.5)$	$(5/3,5/3,5)$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
2	4	$(2,2,0)$	$(2,2,2.5)$	$(2,2,5)$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
3	1	$(0,1,0)$	$(0,1,2.5)$	$(0,1,5)$	1	1	1
3	2	$(0,4/3,0)$	$(0,4/3,2.5)$	$(0,4/3,5)$	1	1	1
3	3	$(0,5/3,0)$	$(0,5/3,2.5)$	$(0,5/3,5)$	1	1	1
3	4	$(0,2,0)$	$(0,2,2.5)$	$(0,2,5)$	1	1	1
4	1	$(-1,1,0)$	$(-1,1,2.5)$	$(-1,1,5)$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
4	2	$(-4/3,4/3,0)$	$(-4/3,4/3,2.5)$	$(-4/3,4/3,5)$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
4	3	$(-5/3,5/3,0)$	$(-5/3,5/3,2.5)$	$(-5/3,5/3,5)$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
4	4	$(-2,2,0)$	$(-2,2,2.5)$	$(-2,2,5)$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
5	1	$(-1,0,0)$	$(-1,0,2.5)$	$(-1,0,5)$	1	1	1
5	2	$(-4/3,0,0)$	$(-4/3,0,2.5)$	$(-4/3,0,5)$	1	1	1
5	3	$(-5/3,0,0)$	$(-5/3,0,2.5)$	$(-5/3,0,5)$	1	1	1
5	4	$(-2,0,0)$	$(-2,0,2.5)$	$(-2,0,5)$	1	1	1
6	1	$(-1,-1,0)$	$(-1,-1,2.5)$	$(-1,-1,5)$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
6	2	$(-4/3,-4/3,0)$	$(-4/3,-4/3,2.5)$	$(-4/3,-4/3,5)$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
6	3	$(-5/3,-5/3,0)$	$(-5/3,-5/3,2.5)$	$(-5/3,-5/3,5)$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
6	4	$(-2,-2,0)$	$(-2,-2,2.5)$	$(-2,-2,5)$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
7	1	$(0,-1,0)$	$(0,-1,2.5)$	$(0,-1,5)$	1	1	1
7	2	$(0,-4/3,0)$	$(0,-4/3,2.5)$	$(0,-4/3,5)$	1	1	1
7	3	$(0,-5/3,0)$	$(0,-5/3,2.5)$	$(0,-5/3,5)$	1	1	1
7	4	$(0,-2,0)$	$(0,-2,2.5)$	$(0,-2,5)$	1	1	1
8	1	$(1,-1,0)$	$(1,-1,2.5)$	$(1,-1,5)$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
8	2	$(4/3,-4/3,0)$	$(4/3,-4/3,2.5)$	$(4/3,-4/3,5)$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
8	3	$(5/3,-5/3,0)$	$(5/3,-5/3,2.5)$	$(5/3,-5/3,5)$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
8	4	$(2,-2,0)$	$(2,-2,2.5)$	$(2,-2,5)$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
9	1	$(1,0,0)$	$(1,0,2.5)$	$(1,0,5)$	1	1	1
9	2	$(4/3,0,0)$	$(4/3,0,2.5)$	$(4/3,0,5)$	1	1	1
9	3	$(5/3,0,0)$	$(5/3,0,2.5)$	$(5/3,0,5)$	1	1	1
9	4	$(2,0,0)$	$(2,0,2.5)$	$(2,0,5)$	1	1	1

and

$$\mathscr{H} = \{0, 0, 0, 1, 1, 1\}. \tag{A.5}$$

The corresponding one-dimensional basis functions are given in Tables A.5 and A.6. The control net and weights are given in Tables A.7 and A.8, respectively.

A.4. Cylindrical solid

The cylinder shown in Fig. 25 has an inner radius of $r_{\mathrm{in}} = 1$, and an outer radius of $r_{\mathrm{out}} = 2$. The length of the cylinder is $L = 5$. The coarse mesh uses quadratic basis functions in all three parametric directions. The ξ coordinate traverses the circumferential direction, the η coordinate traverses the thickness, and the ζ coordinate traverses the length.

The knot vectors are

$$\varXi = \{0, 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 4\}, \tag{A.6}$$

$$\mathscr{H} = \{0, 0, 0, 0.5, 1, 1, 1\} \tag{A.7}$$

and

$$\mathscr{Z} = \{0, 0, 0, 1, 1, 1\}. \tag{A.8}$$

As each of these knot vectors have appeared in a previous example, analytic expressions for the corresponding one-dimensional basis functions may be found in the above tables. The control net and weights are given in Table A.9. Rational solid basis functions are defined by combining the weights and one-dimensional basis functions using (14).

A hollow cylindrical solid can be developed with a single NURBS patch by mapping opposite faces of the parametric domain to the same physical location in the obvious way. For open knot vectors, the control points describing the two faces are coincident. This leads to a C^0-continuous solution across the interface.

Appendix B. Knot selection during *h*-refinement

The difficulty encountered in *h*-refinement is how to determine where to insert knots. Knot locations may always be chosen by the user to increase the resolution in the vicinity of specific features but an automated global refinement strategy is desired. The parameterization of a NURBS patch in physical space is dependent not only on the locations of control points and the polynomial order of the basis functions but also on the weights. In practice, placing a new knot in the middle of an existing knot span, thus splitting the element into two equal parts in the parametric domain, often does not even come close to splitting the element in half in physical space. Complicating matters further is the fact that knot insertion is inherently a global process. For example, consider a mesh in $\mathbb{R}^2$ defined by $\varXi \times \mathscr{H}$. Inserting a knot $\bar{\xi} \in [\xi_i, \xi_{i+1}]$ into $\varXi$ may split the image of the element edge $[\xi_i, \xi_{i+1}] \times \eta_1$ in physical space in half, but fail to split the image of $[\xi_i, \xi_{i+1}] \times \eta_{m+q+1}$ in half, and so no unique knot value will exactly halve every element in the row. Still, in meshes described by one patch, refinement requires selecting one value. The approach we have developed has worked well in the examples we have considered, and is described below for the two-dimensional case.

When inserting a knot into $[\xi_i, \xi_{i+1}]$, perform the following steps:

(1) Find the physical coordinates corresponding to (ξ_i, η_1) and (ξ_{i+1}, η_1), and calculate the midpoint of the chord connecting them; call this point P_1. Likewise, find P_2, the midpoint of the chord from (ξ_i, η_{m+q+1}) and $(\xi_{i+1}, \eta_{m+q+1})$. See Fig. B.1.
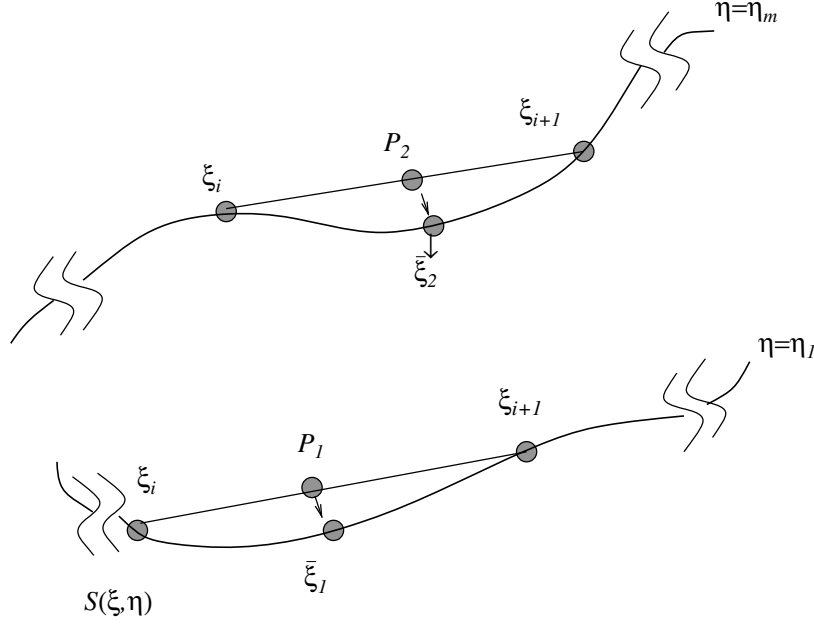
Fig. B.1. Knot selection during *h*-refinement. Approximate the midpoint of the element edge by P_1, the midpoint of the chord connecting the corners of the element. Perform point inversion to find $\bar{\bar{\xi}}_1$, the parameter of the point on the curve that is closest to P_1 in the Euclidean norm. Repeat the process on the opposite side of the domain to find $\bar{\bar{\xi}}_2$. The knot to be inserted is a weighted average of $\bar{\bar{\xi}}_1$ and $\bar{\bar{\xi}}_2$.

(2) Perform *point inversion* to find $\bar{\bar{\xi}}_1$ and $\bar{\bar{\xi}}_2$, the ξ coordinates minimizing the distance between P_1 and P_2 and points on the images of $[\xi_i, \xi_{i+1}] \times \eta_1$ and $[\xi_i, \xi_{i+1}] \times \eta_{m+q+1}$, respectively. See Fig. B.1. This is done using a Newton iteration; for details, see Piegl and Tiller [21].

(3) Take the weighted average of $\bar{\bar{\xi}}_1$ and $\bar{\bar{\xi}}_2$ to get $\bar{\xi}$. Weight by the magnitude of the parametric derivative, $\left\| \frac{d\boldsymbol{x}}{d\xi} \right\|$, where $\boldsymbol{x}$ is the position in physical space and $\|\cdot\|$ denotes the Euclidean norm. That is, the new knot is given by

$$\bar{\xi} = \frac{\left\| \frac{d\boldsymbol{x}}{d\xi} \right\|_{(\bar{\bar{\xi}}_1, \eta_1)} \bar{\bar{\xi}}_1 + \left\| \frac{d\boldsymbol{x}}{d\xi} \right\|_{(\bar{\bar{\xi}}_2, \eta_{m+q+1})} \bar{\bar{\xi}}_2}{\left\| \frac{d\boldsymbol{x}}{d\xi} \right\|_{(\bar{\bar{\xi}}_1, \eta_1)} + \left\| \frac{d\boldsymbol{x}}{d\xi} \right\|_{(\bar{\bar{\xi}}_2, \eta_{m+q+1})}} \tag{B.1}$$

This weighted average produces a better splitting of an edge where the physical position changes rapidly with variation in the parameter. Most notably, this yields a better splitting of longer edges. Independent of different edge lengths, if $\bar{\bar{\xi}}_1 = \bar{\bar{\xi}}_2$, then $\bar{\xi}$ will be set to that value as well. In this way no accuracy is lost in the "easy" cases.

Fig. B.2 compares this approach to knot selection with uniform splitting of the element in the parametric domain. Both methods are applied to the original two-element mesh shown in Fig. 17a. Note that uniform splitting in the parametric space translates to uniform splitting of the short edge (i.e., the circle) but quite non-uniform splitting of the longer edges opposite to it. The advocated scheme behaves in opposite fashion. The former approach produces a better mesh near the circular hole whereas the latter approach produces a somewhat more balanced mesh globally. Both approaches should be useful. The amount of time taken for the advocated refinement process is negligible in comparison with analysis.
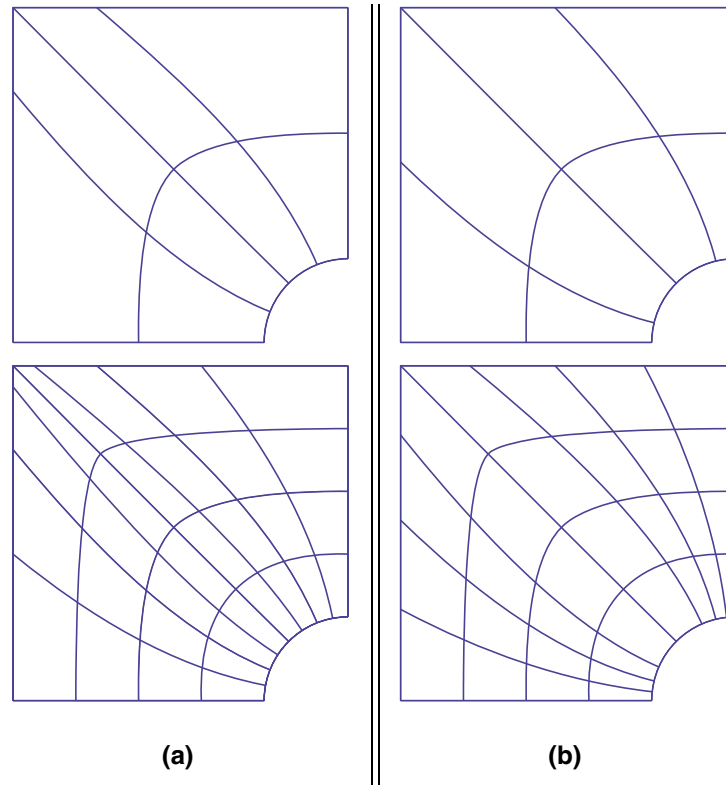
Fig. B.2. Two *h*-refinements of the two-element mesh of Fig. 17a. (a) Elements split uniformly in the parametric space. (b) Elements split using the scheme described above.

References

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

[2] T.J. Barth, Simplified numerical methods for gas dynamics systems on triangulated domains, Ph.D. Thesis, Department of Aeronautics and Astronautics, Stanford University (1998).

[3] M. McMullen, A. Jameson, J.J. Alonso, Application of a non-linear frequency domain solver to the Euler and Navier–Stokes equations, 40th AIAA Aerospace Sciences Meeting and Exhibit, AIAA 2002-0120, Reno, NV, 2002.

[4] M. Gee, W.A. Wall, E. Ramm, Parallel multilevel solutions of nonlinear shell structures, Comput. Methods Appl. Mech. Engrg. (accepted).

[5] G.M. Stanley, Continuum-based shell elements, Ph.D. Thesis, Division of Applied Mechanics, Stanford University, 1985.

[6] T.J. Barth, Numerical methods for gasdynamic systems on unstructured meshes, in: Kröner, Ohlberger, Rohde (Eds.), An Introduction to Recent Developments in Theory and Numerics for Conservation Laws, Lecture Notes in Computational Science and Engineering, Vol. 5, Springer-Verlag, Heidelberg, 1998, pp. 195–285.

[7] S. Yunus, private communication.

[8] B. Szabó, A. Düster, E. Rank, The *p*-version of the finite element method, in: E. Stein, R. de Borst, T.J.R. Hughes (Eds.), Fundamentals, Encyclopedia of Computational Mechanics, vol. 1, Wiley, New York, 2004 (Chapter 5).

[9] B. Szabó, I. Babuška, Finite Element Analysis, Wiley, New York, 1991.

[10] E. Rank, A. Düster, V. Nübel, K. Preusch, O.T. Bruhns, High order finite elements for shells, Comput. Methods Appl. Mech. Engrg. (accepted).

[11] S. Abhyankar, C. Bajaj, Automatic parameterization of rational curves and surfaces I: conics and conicoids, Comput.-Aided Des. 19 (1987) 11–14.

[12] S. Abhyankar, C. Bajaj, Automatic parameterization of rational curves and surfaces II: cubics and cubicoids, Comput.-Aided Des. 19 (1987) 499–502.

[13] S. Abhyankar, C. Bajaj, Automatic parameterization of rational curves and surfaces III: algebraic plane curves, Comput.-Aided Geomet. Des. 5 (1988) 309–321.

[14] S. Abhyankar, C. Bajaj, Automatic parameterization of rational curves and surfaces IV: algebraic space curves, ACM Transactions on Graphics 4 (1989) 325–334.

[15] C. Bajaj, J. Chen, R. Holt, A. Netravali, Energy formulations of A-splines, Comput.-Aided Geomet. Des. 16 (1999) 39–59.

[16] C. Bajaj, J. Chen, G. Xu, Modeling with cubic A-patches, ACM Transactions on Graphics 14 (1995) 103–133.

[17] F. Cirak, M. Ortiz, Fully C^1-conforming subdivision elements for finite deformation thin shell analysis, Int. J. Numer. Methods Engrg. 51 (2001) 813–833.

[18] F. Cirak, M. Ortiz, P. Schröder, Subdivision surfaces: a new paradigm for thin shell analysis, Int. J. Numer. Methods Engrg. 47 (2000) 2039–2072.

[19] F. Cirak, M.J. Scott, E.K. Antonsson, M. Ortiz, P. Schröder, Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision, Comput.-Aided Des. 34 (2002) 137–148.

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

[21] L. Piegl, W. Tiller, The NURBS Book (Monographs in Visual Communication), Second ed., Springer-Verlag, New York, 1997.

[22] P. Kagan, A. Fischer, P.Z. Bar-Yoseph, Mechanically based models: adaptive refinement for B-spline finite element, Int. J. Numer. Methods Engrg. 57 (2003) 1145–1175.

[23] J. Bonet, S. Kulasegaram, M.X. Rodriguez-Paz, M. Profit, Variational formulation for the smooth particle hydrodynamics (SPH) simulation of fluid and solid problems, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1245–1256.

[24] P. Breitkopf, A. Rassineux, J. Savignat, P. Villon, Integration constraint in diffuse element method, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1203–1220.

[25] J.S. Chen, W. Dongdong, S.B. Dong, An extended meshfree method for boundary value problems, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1085–1103.

[26] J.S. Chen, V. Kotta, H. Lu, D. Wang, D. Moldovan, D. Wolf, A variational formulation and a double-grid method for meso-scale modeling of stressed grain growth in polycrystalline materials, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1277–1303.

[27] W. Dongdong, J.S. Chen, Locking-free stabilized conforming nodal integration for meshfree Mindlin–Reissner plate formulation, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1065–1083.

[28] G.E. Fasshauer, Toward approximate moving least squares approximation with irregularly spaced centers, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1231–1243.

[29] S. Fernandez-Mendez, A. Huerta, Imposing essential boundary conditions in mesh-free methods, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1257–1275.

[30] A. Huerta, S. Fernandez-Mendez, W.K. Liu, A comparison of two formulations to blend finite elements and mesh-free methods, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1105–1117.

[31] A. Huerta, Y. Vidal, P. Villon, Pseudo-divergence-free element free Galerkin method for incompressible fluid flow, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1119–1136.

[32] G. Li, N.R. Aluru, Positivity conditions in meshless collocation methods, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1171–1202.

[33] S. Li, H. Lu, W. Han, W.K. Liu, D. Simkins Jr., Reproducing kernel element method. Part II: globally conforming I^m/C^n hierarchies, Comput. Methods Appl. Mech. Engrg. 193 (2004) 953–987.

[34] W.K. Liu, W. Han, H. Lu, S. Li, J. Cao, Reproducing kernel element method. Part I: theoretical formulation, Comput. Methods Appl. Mech. Engrg. 193 (2004) 933–951.

[35] H. Lu, S. Li, D. Simkins Jr., W.K. Liu, J. Cao, Reproducing kernel element method. Part III: generalized enrichment and applications, Comput. Methods Appl. Mech. Engrg. 193 (2004) 989–1011.

[36] T. Rabczuk, T. Belytschko, S.P. Xiao, Stable particle methods based on Lagrangian kernels, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1035–1063.

[37] D. Simkins Jr., S. Li, H. Lu, W.K. Liu, Reproducing kernel element method. Part IV: globally compatible C^n ($n \geqslant 1$) triangular hierarchy, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1013–1034.

[38] D. Sulsky, A. Kaul, Implicit dynamics in the material-point method, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1137–1170.

[39] X. Wang, W.K. Liu, Extended immersed boundary method using FEM and RKPM, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1305–1321.

[40] Z. Wu, Dynamically knots setting in meshless method for solving time dependent propagations equation, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1221–1229.

[41] P.L. Gould, Introduction to Linear Elasticity, Springer Verlag, Berlin, 1999.

[42] S. Timoshenko, S. Woinowsky-Krieger, Theory of Plates and Shells (Engineering Societies Monograph), second ed., McGraw-Hill, New York, 1959.

[43] M. Bischoff, Wall W.A., K.U. Bletzinger, E. Ramm, Models and finite elements for thin-walled structures, in: E. Stein, R. de Borst, T.J.R. Hughes (Eds.), Solids, Structures and Coupled Problems, Encyclopedia of Computational Mechanics, vol. 2, Wiley, New York, 2004 (Chapter 3).

[44] C. Felippa, Course notes for advanced finite element methods, http://caswww.colorado.edu/Felippa.d/FelippaHome.d/Home.html.

[45] T. Belytschko, H. Stolarski, W.K. Liu, N. Carpenter, J.S.-J. Ong, Stress projection for membrane and shear locking in shell finite elements, Comput. Methods Appl. Mech. Engrg. 51 (1985) 221–258.

[46] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods in Fluid Dynamics, Springer-Verlag, Berlin Heidelberg, 1988.

[47] P. Moin, Fundamentals of Engineering Numerical Analysis, Cambridge University Press, 2001.

[48] T.J.R. Hughes, L.P. Franca, A mixed finite element formulation for Reissner–Mindlin plate theory: uniform convergence of all higher order spaces, Comput. Methods Appl. Mech. Engrg. 67 (1988) 223–240.

[49] G. Engel, K. Garikipati, T.J.R. Hughes, M.G. Larson, 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, Comput. Methods Appl. Mech. Engrg. 191 (2002) 3669–3750.

[50] E. Oñate, M. Cervera, Derivation of thin plate bending elements with one degree of freedom per node: a simple three node triangle, Engrg. Comput. 10 (1993) 543–561.

[51] E. Oñate, F. Zarate, Rotation-free triangular plate and shell elements, Int. J. Numer. Methods Engrg. 47 (2000) 557–603.

[52] R. Phaal, C.R. Calladine, A simple class of finite-elements for plate and shell problems: 1. Elements for beams and thin flat plates, Int. J. Numer. Methods Engrg. 35 (1992) 955–977.

[53] R. Phaal, C.R. Calladine, A simple class of finite-elements for plate and shell problems: 2. An element for thin shells with only translational degrees of freedom, Int. J. Numer. Methods Engrg. 35 (1992) 979–996.

[54] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 32 (1982) 199–259.

[55] J.E. Akin, T.E. Tezduyar, Calculation of the advective limit of the SUPG stabilization parameter for linear and higher-order elements, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1909–1922.

[56] M. Bischoff, K. Bletzinger, Improving stability and accuracy of Reissner–Mindlin plate finite elements via algebraic subgrid scale stabilization, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1491–1516.

[57] P.B. Bochev, M.D. Gunzburger, J.N. Shadid, On inf-sup stabilized finite element methods for transient problems, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1471–1489.

[58] E. Burman, P. Hansbo, Edge stabilization for Galerkin approximations of convection–diffusion–reaction problems, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1437–1453.

[59] R. Codina, O. Soto, Approximation of the incompressible Navier–Stokes equations using orthogonal subscale stabilization and pressure segregation on anisotropic finite element meshes, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1403–1419.

[60] A.L.G.A. Coutinho, C.M. Diaz, J.L.D. Alvez, L. Landau, A.F.D. Loula, S.M.C. Malta, R.G.S. Castro, E.L.M. Garcia, Stabilized methods and post-processing techniques for miscible displacements, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1421–1436.

[61] V. Gravemeier, W.A. Wall, E. Ramm, A three-level finite element method for the instationary incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1323–1366.

[62] I. Harari, Stability of semidiscrete formulations for parabolic problems at small time steps, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1491–1516.

[63] G. Hauke, L. Valiño, Computing reactive flows with a field Monte Carlo formulation and multi-scale methods, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1455–1470.

[64] B. Koobus, C. Farhat, A variational multiscale method for the large eddy simulation of compressible turbulent flows on unstructured meshes - application to vortex shedding, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1367–1383.

[65] A. Masud, R.A. Khurram, A multiscale/stabilized finite element method for the advection–diffusion equation, Comput. Methods Appl. Mech. Engrg. 193 (2004) 1997–2018.

[66] T.E. Tezduyar, S. Sathe, Enhanced-discretization space–time technique (EDSTT), Comput. Methods Appl. Mech. Engrg. 193 (2004) 1385–1401.

[67] F.L. Chalot, Industrial aerodynamics, in: E. Stein, R. de Borst, T.J.R. Hughes (Eds.), Computational Fluid Dynamics, Encyclopedia of Computational Mechanics, Vol. 3, Wiley, New York, 2004 (Chapter 12).

[68] C. Johnson, U. Nävert, J. Pitkäranta, Finite element methods for linear hyperbolic problems, Comput. Methods Appl. Mech. Engrg. 45 (1984) 285–312.

[69] T.J.R. Hughes, G. Scovazzi, L.P. Franca, Multiscale and stabilized methods, in: E. Stein, R.D. Borst, T.J.R. Hughes (Eds.), Computational Fluid Dynamics, Encyclopedia of Computational Mechanics, vol. 3, Wiley, New York, 2004 (Chapter 2).

[70] T.J.R. Hughes, Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods, Comput. Methods Appl. Mech. Engrg. 127 (1995) 387–401.

[71] T.J.R. Hughes, G. Feijóo., L. Mazzei, J.B. Quincy, The variational multiscale method—a paradigm for computational mechanics, Comput. Methods Appl. Mech. Engrg. 166 (1998) 3–24.

[72] L.B. Wahlbin, Local behavior in finite element methods, in: P.G. Ciarlet, J.L. Lions (Eds.), Finite Element Methods (Part 1), Handbook of Numerical Analysis, vol. 2, North-Holland, Amsterdam, 1991, pp. 353–522.

[73] T.D. Blacker, CUBIT mesh generation environment users manual vol. 1, Technical Report, Sandia National Laboratories, Albuquerque, NM, 1994.

[74] H. Qin, D. Terzopoulos, Triangular NURBS and their dynamic generalizations, Comput.-Aided Geomet. Des. 14 (1996) 325–347.

[75] K. Höllig, Finite Element Methods with B-Splines, SIAM, Philadelpia, 2003.

[76] A.G. Kravchenko, P. Moin, R. Moser, Zonal embedded grids for numerical simulation of wall-bounded turbulent flows, J. Computat. Phys. 127 (1996) 412–423.

[77] A.G. Kravchenko, P. Moin, K. Shariff, B-spline method and zonal grids for simulation of complex turbulent flows, J. Computat. Phys. 151 (1999) 757–789.

[78] W.Y. Kwok, R.D. Moser, J. Jiménez, A critical evaluation of the resolution properties of B-spline and compact finite difference methods, J. Computat. Phys. 174 (2001) 510–551.

[79] K. Shariff, R.D. Moser, Two-dimensional mesh embedding for B-spline methods, J. Computat. Phys. 145 (1998) 471–488.

[80] T.A. Laursen, Computational Contact and Impact Mechanics, Springer-Verlag, Berlin and Heidelberg, 2002.

[81] P. Wriggers, Computational Contact Mechanics, Wiley, New York, 2002.

[82] H. Qin, D. Terzopoulos, D-NURBS: a physics-based framework for geometric design, IEEE Trans. Visual. Comput. Graph. 2 (1996) 85–96.