A Large Deformation, Rotation-Free, Isogeometric Shell

D.J. Benson [a,1], Y. Bazilevs [a,2], M.-C. Hsu [a,3], and T.J.R. Hughes [b,4]

[a] *Department of Structural Engineering, University of California, San Diego 9500 Gilman Drive, La Jolla, CA 92093, USA*

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

Abstract

Conventional finite shell element formulations use rotational degrees of freedom to describe the motion of the fiber in the Reissner-Mindlin shear deformable shell theory, resulting in an element with five or six degrees of freedom per node. These additional degrees of freedom are frequently the source of convergence difficulties in implicit structural analyses, and, unless the rotational inertias are scaled, control the time step size in explicit analyses. Structural formulations that are based on only the translational degrees of freedom are therefore attractive. Although rotation-free formulations using C^0 basis functions are possible, they are complicated in comparison to their C^1 counterparts. A C^k-continuous, $k \geq 1$, NURBS-based isogeometric shell for large deformations formulated without rotational degrees of freedom is presented here. The effect of different choices for defining the shell normal vector is demonstrated using a simple eigenvalue problem, and a simple lifting operator is shown to provide the most accurate solution. Higher order elements are commonly regarded as inefficient for large deformation analyses, but a traditional shell benchmark problem demonstrates the contrary for isogeometric analysis. The rapid convergenge of the quadratic element is demonstrated for the NUMASHEET S-rail benchmark metal stamping problem.

Key words: isogeometric analysis, NURBS, shells, rotation-free, metal stamping

[1] Professor
[2] Assistant Professor
[3] Graduate Research Assistant
[4] Professor of Aerospace Engineering and Engineering Mechanics, Computational and Applied Mathematics Chair III

1 Introduction

Structural elements, like solid elements, have traditionally used C^0 Lagrange polynomials as basis functions. The development of C^1 shell elements based on Kirchhoff-Love theory proved very difficult, and the structural analysis community eventually adopted the C^0 shear deformable Reissner-Mindlin theory. Since the fiber vector is no longer defined as being normal to the reference surface, two rotational coordinates are needed to define its orientation, but three rotational coordinates are commonly used in practice to simplify the implementation.

Using six degrees of freedom per node doubles the number of degrees of freedom in comparison to the requirements of thin shell theory. Doubling the degrees of freedom increases the stiffness matrix size by a factor of four, and the cost of the matrix factorization by a factor of eight. Since structural elements are typically very ill-conditioned, a direct solver is needed. Sparse direct solvers currently scale poorly beyond a few dozen processors, and therefore the size of structural problems that may be efficiently solved implicitly is capped. Additionally, the rotational degrees of freedom are frequently the source of convergence problems in large deformation analyses. Even large scale quasi-static problems are currently solved using explicit time integration methods to achieve scalability on massively parallel computers. The stable time step is determined by the maximum eigenvalue of the system, which is invariably a mode dominated by the rotational degrees of freedom. Unless the rotational inertias are scaled up, the stable time step size for explicit time integration methods is therefore governed by the thickness of the shell, making explicit thin shell calculations impractical. Shell formulations that use only translational degrees of freedom are therefore attractive for many reasons.

Isogeometric analysis is a new computational method that is based on geometry representations (i.e., basis functions) developed in computer-aided design (CAD), computer graphics (CG), and animation, with a far-reaching goal to bridge the existing gap between CAD and analysis [17, 18, 29]. For the first instantiation of the isogeometric methodology, non-uniform rational B-splines (NURBS) were chosen as a basis, due to their relative simplicity and ubiquity in the worlds of CAD, CG, and animation. It was found that not only were NURBS applicable to engineering analysis, they were better suited for many applications, and were able to deliver accuracy superior to standard finite elements (see, e.g., [1, 5–8, 18, 30, 37]). Subdivision surfaces [14–16] and, more recently, T-Splines [4, 19], were also successfully employed in the analysis context. It should be emphasized that *isogeometric* refers only to using the same basis functions for analysis as for the geometry, and is not restricted to NURBS [29].

Rotation-free isogeometric Bernoulli-Euler beams and Poisson-Kirchhoff plates were introduced and studied in [18]. See also [17].

A rotation-free isogeometric shell formulation is presented here using C^k-continuous, $k \geq 1$, NURBS. The efficiency of the element is enhanced by interpolating the normal from the control points. The normal directions are not uniquely defined and the effects of different choices are explored. A simply lifting operator is found to be a good option. The accuracy and efficiency of the proposed methodology is illustrated with numerical examples.

2 Kinematics

The translational degrees of freedom of the control points define the motion of the reference surface of the shell. Without additional degrees of freedom, the orientation of the fiber with respect to the reference surface is fixed. The simplest assumption regarding the fiber is that it is normal to the reference surface, i.e., the Kirchhoff-Love hypothesis. In the context of a finite element formulation, the Kirchhoff-Love hypothesis may be imposed at a number of different levels, ranging from imposing it at a set of discrete points to imposing it over the entire reference surface. In the current formulation, the normal vectors are associated with the control points. Since the basis functions in isogeometric analysis are generally not interpolatory and the control points do not lie on the shell surface, the normals associated with the control points are not uniquely defined. The consequences of different choices are examined subsequently.

The summation convention for repeated lower-case latin indices is assumed to hold over the range 1 to 3 unless explicitly indicated otherwise.

The kinematics for the rotation-free shell are chosen to be very similar to those for our Reissner-Mindlin isogeometric element [13] in order to achieve commensurate speed and robustness, and to facilitate direct comparisons. The spatial coordinates $\boldsymbol{x}$ of a point with parametric coordinates $\boldsymbol{s}$ are

$$\boldsymbol{x} = \sum_A N_A(s_1, s_2) \left(\boldsymbol{q}_A + s_3 \frac{h_A}{2} \boldsymbol{n}_A \right). \tag{1}$$

The subscript A is the control point index, h_A is the shell thickness, and $\boldsymbol{n}_A$ is the normal. The basis functions N_A are functions only of the two parametric coordinates on the reference lamina ($s_3 = 0$). The coordinates of the control points are denoted by the notation $\boldsymbol{q}_A$ to emphasize their role as generalized coordinates that in general do not lie on the shell surface. This equation is identical to the one used in our previous Reissner-Mindlin element [13] but with the interpretation that the current normal is associated with the control point.

The exact normal is computed by taking the cross product of the derivatives of $\boldsymbol{x}$ with respect to the two parametric coordinates,

$$\boldsymbol{p}_A = \boldsymbol{t}_{A1} \times \boldsymbol{t}_{A2} \tag{2}$$

$$\boldsymbol{t}_{Ai} = \frac{\partial \boldsymbol{x}}{\partial s_i}|_A = \sum_B \frac{\partial N_B}{\partial s_i}|_A \boldsymbol{q}_B \tag{3}$$

$$\boldsymbol{n}_A = \frac{\boldsymbol{p}_A}{|\boldsymbol{p}_A|}. \tag{4}$$

The subscript A on the tangent vectors indicates that they are associated with the basis function A. There is not, however, a unique material point on the shell associated with basis function A, and three choices are explored in the subsequent sections.

The velocity is obtained by differentiating Equation 1 with respect to time,

$$\dot{\boldsymbol{x}} = \sum_A N_A(s_1, s_2) \left(\dot{\boldsymbol{q}}_A + s_3 \frac{h_A}{2} \dot{\boldsymbol{n}}_A \right). \tag{5}$$

For a traditional shell formulation with rotational degrees of freedom, the velocity of the normal is expressed in terms of the angular velocity, $\boldsymbol{\omega}$, namely,

$$\dot{\boldsymbol{n}}_A = \boldsymbol{\omega}_A \times \boldsymbol{n}_A. \tag{6}$$

This has the unintended consequence of introducing a spurious singularity in the stiffness matrix about the axis of the normal if three rotational degrees of freedom per node, or control point, are used. For the rotation-free shell, the velocity of the normal is calculated by directly differentiating Equation 4 with respect to time,

$$\dot{\boldsymbol{n}} = \frac{1}{|\boldsymbol{p}|} \left(\boldsymbol{I} - \boldsymbol{n} \otimes \boldsymbol{n} \right) \dot{\boldsymbol{p}} \tag{7}$$

$$\dot{\boldsymbol{p}} = \frac{\partial \dot{\boldsymbol{x}}}{\partial s_1} \times \frac{\partial \boldsymbol{x}}{\partial s_2} + \frac{\partial \boldsymbol{x}}{\partial s_1} \times \frac{\partial \dot{\boldsymbol{x}}}{\partial s_2} \tag{8}$$

2.1 The discrete gradient operator

The Jacobian $J_{jk} = \frac{\partial x_j}{\partial s_k}$ is defined as

$$J_{ik} = \left[\sum_A \frac{\partial N_A}{\partial s_k} q_{Ai} \right] + s_3 \left[\sum_A \frac{\partial N_A}{\partial s_k} \frac{h_A}{2} n_{Ai} \right], \quad k = 1, 2 \tag{9}$$

$$J_{i3} = \left[\sum_A N_A \frac{h_A}{2} n_{Ai} \right] \tag{10}$$

where $i = 1, 2, 3$ and the terms enclosed by square brackets are independent of s_3. Its inverse $J^{-1}_{kj} = \frac{\partial s_k}{\partial x_j}$ is used frequently in subsequent equations.

Differentiating the velocity with respect to the coordinates gives the velocity gradient $\boldsymbol{L}$ in the global coordinate system

$$\begin{aligned} L_{ij} = &\sum_{k=1,2} J^{-1}_{kj} \left[\sum_A \frac{\partial N_A}{\partial s_k} \dot{x}_{Ai} \right] \\ &+ s_3 \sum_{k=1,2} J^{-1}_{kj} \left[\sum_A \frac{\partial N_A}{\partial s_k} \frac{h_A}{2} \dot{n}_{Ai} \right] \\ &+ J^{-1}_{j3} \left[\sum_A N_A \frac{h_A}{2} \dot{n}_{Ai} \right] \end{aligned} \tag{11}$$

and the rate of deformation $\boldsymbol{D}$ is

$$\boldsymbol{D} = \frac{1}{2} \left(\boldsymbol{L} + \boldsymbol{L}^T \right). \tag{12}$$

The relationship between the time derivative of the normal and the velocities of the control points is subsequently required. First, the projection matrix $\boldsymbol{P}_A$ is defined as

$$P_{Aij} = \frac{1}{|\boldsymbol{p}_A|} \left(\delta_{ij} - n_{Ai} n_{Aj} \right) \tag{13}$$

and then Equation 8 is expanded as follows,

$$\dot{p}_{Ai} = e_{ijk} \sum_B \left[\frac{\partial N_B}{\partial s_1} (\boldsymbol{s}_A) t_{A2k} - \frac{\partial N_B}{\partial s_2} (\boldsymbol{s}_A) t_{A1k} \right] \dot{x}_{Bj} \tag{14}$$

$$t_{Ajk} = \sum_C \frac{\partial N_C}{\partial s_j} (\boldsymbol{s}_A) x_{Ck} \quad j = 1, 2 \tag{15}$$

where e_{ijk} is the alternator tensor. The time rate of the normal $\boldsymbol{n}_A$ is therefore the product of two terms, the first of which depends only on control point A, an important fact for efficient implementation. To simplify subsequent equations, the array $\boldsymbol{c}$ is defined as

$$c^A_{Bk} = \frac{\partial N_B}{\partial s_1} (\boldsymbol{s}_A) t_{A2k} - \frac{\partial N_B}{\partial s_2} (\boldsymbol{s}_A) t_{A1k} \tag{16}$$

allowing $\dot{\boldsymbol{n}}_A$ to be written succintly as

$$\dot{p}_{Aj} = \sum_B e_{jkn} c^A_{Bn} \dot{x}_{Bk} \tag{17}$$

$$\dot{n}_{Ai} = P_{Aij} \dot{p}_{Aj} = P_{Aij} \sum_B e_{jkn} c^A_{Bn} \dot{x}_{Bk}. \tag{18}$$

The discrete gradient operator $\boldsymbol{B}$ is defined through its action on $\dot{\boldsymbol{x}}_A$,

$$L_{ij} = \sum_A B_{ijAk} \dot{x}_{Ak}. \tag{19}$$

Substituting Equation 18 into Equation 11 provides a complete expression for the velocity gradient in terms of the control point velocities, namely

$$\begin{aligned} L_{ij} = &\sum_{k=1,2} J^{-1}_{kj} \left[\sum_A \frac{\partial N_A}{\partial s_k} \dot{x}_{Ai} \right] \\ &+ \sum_B \left\{ \left(s_3 \sum_{k=1,2} J^{-1}_{kj} \frac{\partial N_B}{\partial s_k} + J^{-1}_{3j} N_B \right) \frac{h_B}{2} P_{Bi\ell} \sum_A e_{\ell mn} c^B_{An} \dot{x}_{Am} \right\} \end{aligned} \tag{20}$$

Collecting the coefficients of $\dot{x}_{Ak}$ gives the individual terms in $\boldsymbol{B}$,

$$\begin{aligned} B_{ijAk} = &\sum_{\ell=1,2} J^{-1}_{\ell j} \frac{\partial N_A}{\partial s_\ell} \delta_{ik} \\ &+ \sum_B \left\{ \left(s_3 \sum_{\ell=1,2} J^{-1}_{\ell j} \frac{\partial N_B}{\partial s_\ell} + J^{-1}_{3j} N_B \right) \frac{h_B}{2} P_{Bim} e_{mkn} c^B_{An} \right\} \end{aligned} \tag{21}$$

The contribution to the internal force vector may be expressed using $\boldsymbol{B}$ as

$$F^{\text{int}}_{Ak} = \int_V \sigma_{ij} B_{ijAk} \mathrm{d}V, \tag{22}$$

We note that there is a more efficient way to compute Equation 22 than a direct computation of it using Gaussian quadrature. This is described in Section 3.5.

3 Implementation

The element is implemented using the generalized element formulation [12], and therefore may use any of the CAD basis functions that are, at the minimum, C^1 continuous. The example calculations are performed with NURBS [17, 34] but T-splines [4, 19], for example, would work as well.

The first term of $\boldsymbol{B}$ in Equation 21 has the usual form of the discrete gradient in solids and membranes, while the second involves a summation over all the control points in the element. The number of operations required to directly evaluate this expression therefore scales with the square of the number of control points instead of the usual linear scaling associated with elements having rotational degrees of freedom. Even for fairly low-order elements (e.g., quadratic and cubic),

the rotation-free formulation might be expected to be more expensive than the conventional formulation. There are, however, three mitigating factors. First, the cost of calculating a tangent matrix is small in comparison to the cost of factorizing it. Second, the $\boldsymbol{B}$ matrix is needed only for the tangent matrix since the evaluation of the internal force may be performed without the explicit evaluation of $\boldsymbol{B}$ (see Section 3.5). Thirdly, and perhaps most importantly, by design, the terms (e.g., c^B_{An}) associated with the double summations are nodal quantities that are independent of the integration points, and therefore only need to be evaluated once per residual (or stiffness) evaluation.

Additionally, many of the same strategies used in the shear deformable element [13] may be carried over to reduce the cost of evaluating the residual. The most important strategy is noting that many of the expressions that need to be evaluated are linear in s_3. To take advantage of this, the integration of an element is performed in two nested loops: The inner loop is through the thickness of the shell, and is associated with s_3, while the outer loop is associated with the quadrature points specified over the reference lamina, $(s_1, s_2, 0)$. To distinguish quantities associated with these two loops, the superscripts G and g refer to the outer and inner loops respectively. For example, the volume of the element, V, is evaluated as

$$V = \int_{\boldsymbol{x}} \mathrm{d}\boldsymbol{x} = \int_{\boldsymbol{s}} \det\left(\boldsymbol{J}\right) \mathrm{d}\boldsymbol{s} = \sum_G \sum_g W^G w^g \det\left(\boldsymbol{J}^{Gg}\right) = \sum_G \sum_g V^{Gg}. \quad (23)$$

where the integration weights for G and g are W^G and w^g, respectively. For compactness, the evaluation of a function at the specific integration point (s_1^G, s_2^G, s_3^g) is denoted by the dual superscripts "Gg" and the volume contribution of integration point Gg is denoted V^{Gg}. To emphasize where efficiency has been gained by taking advantage of the linearity of a function in s_3, the alternate notation $\boldsymbol{J}^G(s_3^g)$ may also be used.

3.1 Evaluation of the normals

The normals are evaluated using Equation 4 and as they are not functions of the lamina integration points, they only need to be evaluated only once per element residual evaluation. As will be seen below, adopting this convention reduces the number of times that the double summation over the control points is performed to only two: once during the evaluation of terms required to calculated the velocity gradient, and once during the final residual evaluation. A traditional curved thin shell formulation typically requires that the double summation be performed twice at each lamina integration point, an expense that may be insignificant for an implicit calculation, but significant for an explicit calculation.

Unlike traditional Lagrangian basis functions, which are interpolatory at the nodes, there is not a natural, unique location for evaluating the normal associated to a

control point. The effect of the choice of the normal location is not evident for many test cases solved with a degenerated solid formulation with rotations. A flat plate, for example, has a constant normal direction and the solution obtained is therefore independent of where the normals are evaluated. A formulation without rotational degrees of freedom, however, involves the derivatives of the normal vectors, and the accuracy of the solution clearly depends on how they are evaluated.

Three choices for the normals are compared in a later section to explore the sensitivity of of the element to how the normal is evaluated. The first one evaluates the normal at the location where the basis function achieves its maximum. Note that for NURBS, each basis function has a single maximum, giving a unique location for each normal. The second projects the location of each control point to the closest point on the shell surface, and uses the parametric coordinates of that point for evaluating the normal. There are other possible choices for evaluation points, for example the Greville and Demko abscissae [3], but these are not considered in this work.

The third approach is motivated by a desire to minimize shear locking by approximating the kinematics of classical thin shell theory, where the fiber is always normal to the reference surface. While a $\bar{\boldsymbol{B}}$-type formulation [21] is possible, it might result in a formulation that involves operations over the entire NURBS patch. For the sake of computational efficiency in an explicit formulation, a procedure that is local to the element is strongly preferred. Unlike the two previous choices, this one does not associate a physical point in the shell mid-surface with each basis function. Instead, the tangent vectors at integration points, given by the exact formula,

$$\boldsymbol{t}_i(\boldsymbol{s}_g) = \sum_B \frac{\partial N_B}{\partial s_i}(\boldsymbol{s}_g)\boldsymbol{q}_B, \tag{24}$$

are required to be interpolates of the tangent vectors associated with the basis functions, that is,

$$\boldsymbol{t}_i(\boldsymbol{s}_g) = \sum_A N_A(\boldsymbol{s}_g)\boldsymbol{t}_{Ai}. \tag{25}$$

The $\boldsymbol{t}_{Ai}$ are assumed to be linear functions of the generalized coordinates $\boldsymbol{q}_b$,

$$\boldsymbol{t}_{Ai} = \sum_B C^A_{Bi}\boldsymbol{q}_B, \tag{26}$$

but, in contradistinction to Equation 4, the coefficients C^A_{Bi} are *not* equal to $\frac{\partial N_B}{\partial s_i}$ evaluated at some $\boldsymbol{s}_A$ associated with N_A. Rather, we determine C^A_{Bi} by combining Equations 24, 25, and 26 to obtain

$$\sum_B \frac{\partial N_B}{\partial s_i}(\boldsymbol{s}_g)\boldsymbol{q}_B = \sum_B \sum_A N_A(\boldsymbol{s}_g)C^A_{Bi}\boldsymbol{q}_B, \tag{27}$$

and require it to hold for all values of $\boldsymbol{q}_B$. This results in a system of linear equations

for the coefficients C^A_{Bi},

$$\sum_A N_A(\boldsymbol{s}_g) C^A_{Bi} = \frac{\partial N_B}{\partial s_i}(\boldsymbol{s}_g). \tag{28}$$

We refer to this method of defining the $\boldsymbol{t}_{Ai}$ as the *lifting operator* method because it *lifts* values from the integration points to the basis functions. It ensures that the tangent vectors at the integration points, given by Equation 25, and hence the normals at the integration points,

$$\boldsymbol{n}(\boldsymbol{s}_g) = \frac{\boldsymbol{p}(\boldsymbol{s}_g)}{|\boldsymbol{p}(\boldsymbol{s}_g)|}, \quad \text{where} \quad \boldsymbol{p}(\boldsymbol{s}_g) = \boldsymbol{t}_1(\boldsymbol{s}_g) \times \boldsymbol{t}_2(\boldsymbol{s}_g), \tag{29}$$

are *exact*. This may be interpreted as a discrete Kirchhoff approximation. For elements with a polynomial or B-spline basis (equivalent to setting the fourth control point coordinate $w = 1$ everywhere for NURBS), the Kirchhoff condition is satisfied point-wise over the entire element. There is no difference between a discrete Kirchhoff approximation at the integration points and its point-wise exact satisfaction over the element in terms of numerical performance since only the integration points are evaluated during the numerical solution.

The example calculations use $p + 1$ points in each in-plane direction, providing a sufficient number equations for determining $\hat{N}^A_{Bi}$. The "reduced" $p \times p$ rule does not, and additional points must be chosen. The additional five points used in the example calculations for a quadratic NURBS element with 2×2 integration are $(\pm 1/\sqrt{3}, 0)$, $(0, \pm 1/\sqrt{3})$, and $(0, 0)$.

3.2 Evaluation of the Jacobian

The tangent vectors, $\boldsymbol{t}_{Ak}$ for $k = 1, 2$ are evaluated at the control points using Equation 26, and the normal vectors are calculated from Equations 2 and 4. Since these quantities are evaluated at the control points, they are calculated once per residual evaluation and stored.

The Jacobian is efficiently evaluated as an affine function of s_3,

$$J^G_{ik}(s_3) = J^{G1}_{ik} + s_3 J^{G2}_{ik} \tag{30}$$

$$J^{G1}_{ik} = \frac{\partial x_i}{\partial s_k} = \sum_A \frac{\partial N_A}{\partial s_k} q_{Ai} \quad k = 1, 2 \tag{31}$$

$$J^{G1}_{i3} = \sum_A N_A \frac{h_A}{2} n_{Ai} \tag{32}$$

$$J^{G2}_{ik} = \sum_A \frac{\partial N_A}{\partial s_k} \frac{h_A}{2} n_{Ai} \quad k = 1, 2 \tag{33}$$

3.3 Evaluation of the velocity gradient

The velocity gradient is an affine function of s_3 at lamina integration point G, allowing it to be efficiently evaluated as

$$L_{ij}^{G} = L_{ij}^{G1} + s_3 L_{ij}^{G2} \tag{34}$$

$$L_{ij}^{G1} = \sum_{k=1,2} J_{kj}^{-1} \left[\sum_{A} \frac{\partial N_A}{\partial s_k} \dot{x}_{Ai} \right] + J_{j3}^{-1} \left[\sum_{A} N_A \frac{h_A}{2} \dot{n}_{Ai} \right] \tag{35}$$

$$L_{ij}^{G2} = \sum_{k=1,2} J_{kj}^{-1} \left[\sum_{A} \frac{\partial N_A}{\partial s_k} \frac{h_A}{2} \dot{n}_{Ai} \right]. \tag{36}$$

3.4 Stress update

The co-rotational stress update procedure used here is the same one used in our previous paper [13]. The local corotational coordinate system, $\boldsymbol{e}_i^{\ell}, i = 1, 2, 3$, for the stress is defined at the integration points using the invariant scheme of Hughes and Liu [26] with the convention that $\boldsymbol{e}_3^{\ell}$ is the direction normal to the reference surface. The tangent vectors $\boldsymbol{t}^i, i = 1, 2$ are defined as

$$\boldsymbol{t}_i = \sum_{A} \frac{\partial N_A}{\partial s_i} \boldsymbol{q}_A \tag{37}$$

and the normal vector is

$$\boldsymbol{p} = \boldsymbol{t}_1 \times \boldsymbol{t}_2 \tag{38}$$

$$\boldsymbol{n} = \boldsymbol{e}_3^{\ell} = \frac{\boldsymbol{p}}{||\boldsymbol{p}||} \tag{39}$$

where $||\cdot||$ is the usual three-dimensional Euclidean norm. Defining $\boldsymbol{t}_\alpha$ and $\boldsymbol{t}_\beta$ by

$$\boldsymbol{t}_\alpha = \frac{\boldsymbol{t}_1 + \boldsymbol{t}_2}{||\boldsymbol{t}_1 + \boldsymbol{t}_2||}, \tag{40}$$

$$\boldsymbol{t}_\beta = \frac{\boldsymbol{n} \times \boldsymbol{t}_\alpha}{||\boldsymbol{n} \times \boldsymbol{t}_\alpha||}, \tag{41}$$

the remaining local coordinate vectors are

$$\boldsymbol{e}_1^{\ell} = \frac{\sqrt{2}}{2} \left(\boldsymbol{t}_\alpha - \boldsymbol{t}_\beta \right), \tag{42}$$

$$\boldsymbol{e}_2^{\ell} = \frac{\sqrt{2}}{2} \left(\boldsymbol{t}_\alpha + \boldsymbol{t}_\beta \right). \tag{43}$$

The rotation matrix from the local to the global coordinate system, $v^g = \boldsymbol{R} v^\ell$, is

$$\boldsymbol{R} = \left[\boldsymbol{e}_1^\ell \; \boldsymbol{e}_2^\ell \; \boldsymbol{e}_3^\ell \right]. \tag{44}$$

The stress is updated in the local coordinate system using the local rate of deformation, $\boldsymbol{D}^\ell$,

$$\boldsymbol{D}^\ell = \boldsymbol{R}^T \boldsymbol{D}^g \boldsymbol{R} \tag{45}$$

The normal component, D_{33}^ℓ, is calculated within the constitutive model to satisfy the zero normal stress condition $\sigma_{33}^\ell = 0$. The algorithm for calculating D_{33}^ℓ depends on the particular constitutive model [24, 35, 36]. The updated stress is rotated into the global coordinate system at the end of the time step for evaluating the residual,

$$\boldsymbol{\sigma} = \boldsymbol{R} \boldsymbol{\sigma}^\ell \boldsymbol{R}^T. \tag{46}$$

3.5 Evaluation of the residual and stiffness matrix

The internal forces may be directly evaluated by performing the appropriate sparse matrix multiply and numerical quadrature of Equation 22, but the operation count may be reduced by taking advantage of the structure of the internal forces. Integrating through the thickness, the force resultants at G are defined as

$$R_{ip}^G = \sum_g \sigma_{ij}^g (J^{Gg})_{pj}^{-1} V^{Gg} \tag{47}$$

$$\hat{R}_{ip}^G = \sum_g s_3^g \sigma_{ij}^g (J^{Gg})_{pj}^{-1} V^{Gg}. \tag{48}$$

Introducing the resultants and the expression for $\boldsymbol{B}$, Equation 21, into Equation 22 gives

$$F_{Ai}^{\text{int}} = \sum_G \left\{ \left[\sum_{p=1,2} R_{ip}^G \frac{\partial N_A^G}{\partial s_p} \right] + \sum_B \left[\frac{h_B}{2} \left(\sum_{p=1,2} \hat{R}_{jp}^G \frac{\partial N_B^G}{\partial s_p} + R_{j3}^G N_B^G \right) P_{jk}^B \right] e_{ik\ell} c_{A\ell}^B \right\}. \tag{49}$$

The terms in the square brackets may be summed across the lamina integration points, G, giving the partial internal force contributions defined in Equations 50 and 51.

$$\bar{F}_{Ai} = \sum_G \sum_{p=1,2} R_{ip}^G \frac{\partial N_A^G}{\partial s_p} \tag{50}$$

$$\hat{F}_{Aj} = \sum_G \frac{h_A}{2} \left(\sum_{p=1,2} \hat{R}_{jp}^G \frac{\partial N_A^G}{\partial s_p} + R_{j3}^G N_A^G \right) \tag{51}$$

The final internal force evaluation, Equation 52, is evaluated outside of the integration loop so that the summation over B is only performed once per element residual evaluation, thereby reducing calculations.

$$F_{Ai} = \bar{F}_{Ai} + \sum_B e_{ik\ell} \hat{F}_{Bj} P^B_{jk} c^B_{A\ell} \tag{52}$$

The material tangent contribution to the stiffness matrix is

$$\boldsymbol{K} = \int_V \boldsymbol{B}^T \boldsymbol{C} \boldsymbol{B} \mathrm{d}V \tag{53}$$

where $\boldsymbol{C}$ is the material tangent constitutive matrix, and $\boldsymbol{B}$ is evaluated using Equation 21.

4 Dynamics and the mass matrix

The first variation of $\boldsymbol{x}$ is

$$\delta \boldsymbol{x} = \sum_A N_A \left(\delta \boldsymbol{q}_A + \frac{h_A}{2} s_3 \sum_B \frac{\partial \boldsymbol{n}_A}{\partial \boldsymbol{q}_B} \delta \boldsymbol{q}_B \right) \tag{54}$$

and the acceleration, obtained by differentiating the velocity, is

$$\ddot{\boldsymbol{x}} = \sum_C N_C \left(\ddot{\boldsymbol{q}}_C + \frac{h_C}{2} s_3 \sum_D \left\{ \frac{\partial \boldsymbol{n}_C}{\partial \boldsymbol{q}_D} \ddot{\boldsymbol{q}}_D + \sum_E \frac{\partial^2 \boldsymbol{n}_C}{\partial \boldsymbol{q}_C \boldsymbol{q}_D} \dot{\boldsymbol{q}}_C \dot{\boldsymbol{q}}_D \right\} \right). \tag{55}$$

The inertial contribution to the principle of virtual work therefore has the form

$$\begin{aligned} \int_{\boldsymbol{s}} \delta \boldsymbol{x} \cdot \ddot{\boldsymbol{x}} \rho \det(\boldsymbol{J}) \, \mathrm{d}\boldsymbol{s} = & \int_{\boldsymbol{s}} \sum_{A,B} N_A N_B \delta \boldsymbol{q}_A \cdot \ddot{\boldsymbol{q}}_B \rho \det(\boldsymbol{J}) \, \mathrm{d}\boldsymbol{s} \\ & + \int_{\boldsymbol{s}} \sum_A \frac{h_A}{2} s_3 (\ldots) \rho \det(\boldsymbol{J}) \, \mathrm{d}\boldsymbol{s} \\ & + \int_{\boldsymbol{s}} \sum_{A,B} \frac{h_A h_B}{4} s_3^2 (\ldots) \rho \det(\boldsymbol{J}) \, \mathrm{d}\boldsymbol{s}. \end{aligned} \tag{56}$$

The first term on the right hand side has the same form as the consistent mass matrix for solid elements. The second term is small provided that $\rho \det(\boldsymbol{J})$ is nearly constant through the thickness, which holds when the thickness is small in comparison to the radius of curvature. The third term is small when the thickness of the element is small relative to its characteristic length. Only the first term on the right hand side is therefore retained for the consistent mass matrix for thin shells,

$$M_{AB} = \int_{\boldsymbol{s}} N_A N_B \rho \det(\boldsymbol{J}) \, \mathrm{d}\boldsymbol{s} \tag{57}$$

$$\boldsymbol{F}^{\text{inertial}} = \boldsymbol{M}\ddot{\boldsymbol{q}}. \tag{58}$$

For higher order elements, the consistent mass matrix provides a more accurate solution for eigenvalue problems than a lumped mass matrix. However a lumped mass matrix is a requirement for explicit time integration methods. The NURBS basis functions are non-negative everywhere and therefore none of the problems associated with the development of lumped mass matrices for higher order Lagrange and serendipity basis functions occur. Simple row summing [25] is used in the explicit calculations presented later.

4.1 Time step size estimation

Since the cost of an explicit simulation is inversely proportional to the stable time step size, an accurate estimate of time step size is necessary for efficient simulations. The traditional approach to obtaining the time step size invokes two mathematical bounds on the maximum frequency of the structural model. The first bound is the maximum frequency of the system is bounded from above by the maximum of the individual maximum frequencies of the elements,

$$\omega_{\max}^{\text{system}} \leq \max_{e=1,NEL} (\omega_{\max}^{e}). \tag{59}$$

A short proof is given in [25], where the original proofs were attributed to Irons [31] and Hughes *et al.* [27].

The second required bound is on the maximum frequency of the individual elements. The first bounds of this type were originally obtained for finite difference methods using linear basis functions in one dimension,

$$\triangle t^e \leq \frac{2}{\omega_{\max}^{e}} = \frac{\ell^e}{c_{\text{eff}}^{e}} \tag{60}$$

where ℓ^e is the length of the element and c_{eff}^{e} is the effective sound speed (which is a function not only of the material properties but also the shock viscosity formulation [10]). Each constitutive model used in an explicit analysis therefore calculates effective elastic moduli in addition to updating the stress and internal history variables. This applies even to materials that may not have a classical elastic response, e.g., a viscous fluid. In two and three dimensions, the same expression is often used with a heuristic formula for a characteristic element length. A more rigorous bound [22] has been obtained in terms of the discrete gradient operator.

A more precise estimate of the maximum system frequency may be obtained using the power iteration method developed by Benson [11] for multi-material arbitrary

Lagrangian Eulerian (MMALE) methods. Power iteration [39] obtains the maximum eigenvalue of the system by the iterative scheme

$$\left(\omega_{\max}^2 \boldsymbol{\Psi}\right)^{i+1} = \boldsymbol{M}^{-1}\boldsymbol{K}\boldsymbol{\Psi}^i \tag{61}$$

$$(\omega_{\max}^2)^{i+1} = \max_{n=1,NEQN}\left(\frac{(\omega_{\max}^2 \boldsymbol{\Psi})_n^{i+1}}{\boldsymbol{\Psi}_n^i}\right) \tag{62}$$

where $\boldsymbol{M}$ and $\boldsymbol{K}$ are the mass and elastic stiffness matrices of the system, i is the iteration number, n is the equation number, and $NEQN$ is the number of equations in the system. Twelve iterations are usually sufficient for convergence.

In practice, in an explicit code, the product $\boldsymbol{K}\boldsymbol{\Psi}^i$ is evaluated using the standard strategy of evaluating the internal force contribution on the element level from the strain calculated from the eigenvector and the effective elasticity matrix, $\boldsymbol{C}$, that is calculated from the same effective elastic moduli used in the sound speed calculation, namely,

$$\boldsymbol{\epsilon}^{\Psi} = \boldsymbol{B}\boldsymbol{\Psi}^i \tag{63}$$

$$\boldsymbol{\sigma}^{\Psi} = \boldsymbol{C}\boldsymbol{\epsilon}^{\Psi} \tag{64}$$

$$\boldsymbol{K}\boldsymbol{\Psi}^i = \int \boldsymbol{B}^T\boldsymbol{\sigma}^{\Psi}\mathrm{d}\boldsymbol{x} \tag{65}$$

and then the element contribution is assembled into the global vector. Note that boundary conditions, such as contact, that contribute terms to the stiffness matrix also contribute to the product $\boldsymbol{K}\boldsymbol{\Psi}^i$.

The ratio, R, of the actual maximum system eigenvalue and the bound obtained from a traditional element characteristic length evaluation has been observed to change slowly with time even for problems involving large strains and material nonlinearities [11], allowing the actual maximum stable time step size to be safely approximated as

$$\triangle t_{\text{actual}} \approx R\triangle t^e. \tag{66}$$

where $\triangle t^e$ is the traditional bound obtained by taking minimum time step size over all the elements. In practice, the ratio R is updated infrequently, with one power iteration performed each time step, for a fixed small number of time steps in a larger time interval. The time interval between the updates of R depends on the particular problem; see Benson [11] for more details.

4.2 Time Integration

The explicit central difference method and implicit Newmark method are used in LS-DYNA to advance the solution in time, and either may be used with the element formulation presented here. Since the example calculations are explicit, the central

difference method is briefly summarized. Using the lumped mass matrix, the accelerations are calculated from the assembled forces at time step n, and the velocities are updated to $n + 1/2$,

$$\dot{\boldsymbol{q}}_A^{n+1/2} = \dot{\boldsymbol{q}}_A^{n-1/2} + \triangle t \ddot{\boldsymbol{q}}_A^n \tag{67}$$

The translational velocities are integrated to give the current control point coordinates,

$$\boldsymbol{q}_A^{n+1} = \boldsymbol{q}_A^n + \triangle t \dot{\boldsymbol{q}}_A^{n+1/2}. \tag{68}$$

4.3 Boundary conditions

Natural force and moment boundary condtions are treated in the same way as in standard finite element analysis. Essential displacement boundary conditions can be satisfied strongly by assigning control variable values, analogous to assigning nodal values as in standard finite element analysis. Essential rotation boundary conditions can also be satisfied strongly by assigning control variable values at the boundary and values adjacent to the boundary. This simple procedure was first utilized by Aurrichio et al. [2] in their "stream function" formulation of elasticity. It is based on the observation that, for open knot vectors, the slope of a NURBS surface at its boundary is defined by the slope of the first row of bilinear control elements [17] adjacent to the boundary. The analogous case for curves is illustrated in Figure 2.20 of [17]. Alternatively, rotation boundary conditions can be satisfied weakly, using penalties, and/or Lagrange multipliers, as in [18].

5 Example Calculations

In this section we present one linear elastic and two nonlinear elastic-plastic computational examples. Unless otherwise specified, there are $p + 1$ integration points in each in-plane direction between knots where p is the polynomial order of the NURBS basis. We denote by E, ν, ρ, σ_y, and E_H the Young's modulus, Poisson's ratio, density, initial yield stress, and plastic hardening modulus, respectively.

All calculations were performed in double precision on a Dell laptop computer with an Intel Core Duo using a single processor.

5.1 Vibration of a square plate

The vibration of a simply supported square plate demonstrates the accuracy of the rotation-free formulation in linear analysis, and provides a convenient means of

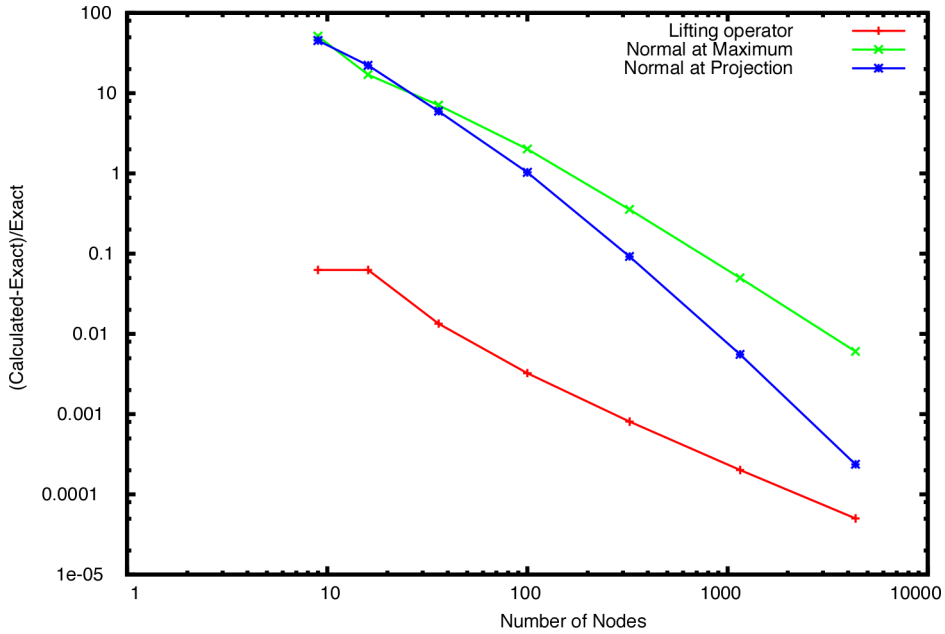


Fig. 1. The accuracy of the first eigenvalue for a square plate as a function of the choice of normal and the number of control points.

exploring the different choices for the normals for the rotation-free formulation.

The exact eigenvalues in radians for a square plate of length L and thickness h, using thin plate theory, are

$$\omega_{ij} = C(i^2 + j^2) \quad 0 < i, j \tag{69}$$

$$C = \pi^2 \sqrt{\frac{E}{\rho(12(1-\nu^2))}} \frac{h}{L^2} \tag{70}$$

where E is Young's modulus, ν is Poisson's ratio, and ρ is the density. For the calculations shown here, the values are $E = 10^7$, $\nu = 0.3$, $\rho = 1$, $L = 10.0$, and $h = 0.05$.

A plot of the lowest eigenvalue ($i = j = 1$) as a function of the number of control points for quadratic elements ($p = 2$) is shown in Figure 1. A consistent mass matrix, neglecting rotary inertia, was used for these calculations. Two integration points were used in the through-thickness direction. All the control points in the mesh are counted, including those on the boundary that have all their degrees of freedom constrained. The plot starts at nine control points because a single quadratic isogeometric element requires nine control points, the same as a Lagrangian element, but only one control point is unconstrained.

The results using the lifting operator are nearly three orders of magnitude more accurate than the other two for coarse meshes. On the finest mesh, the lifting operator is still two orders of magnitude more accurate than the normal calculated at the

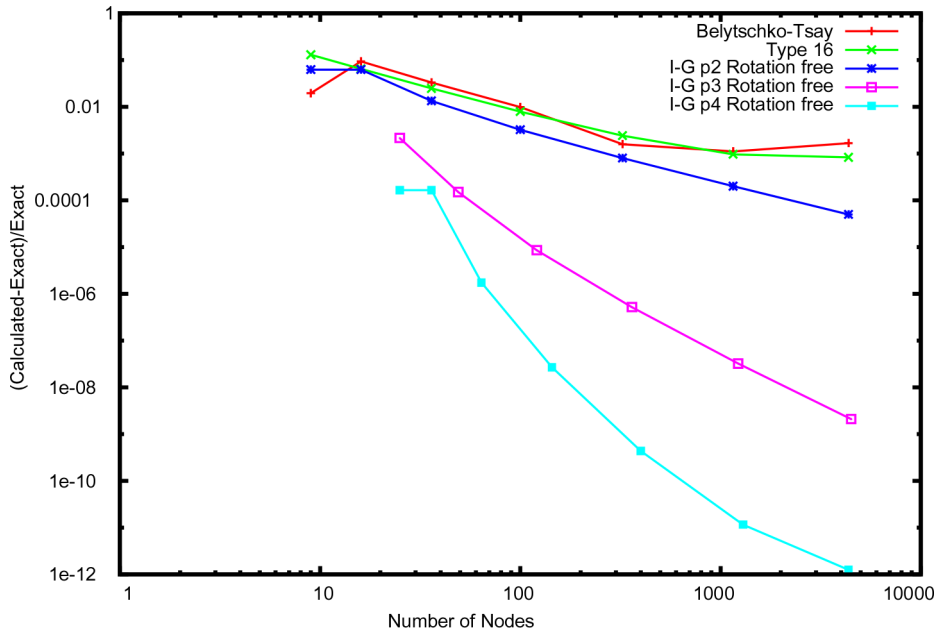


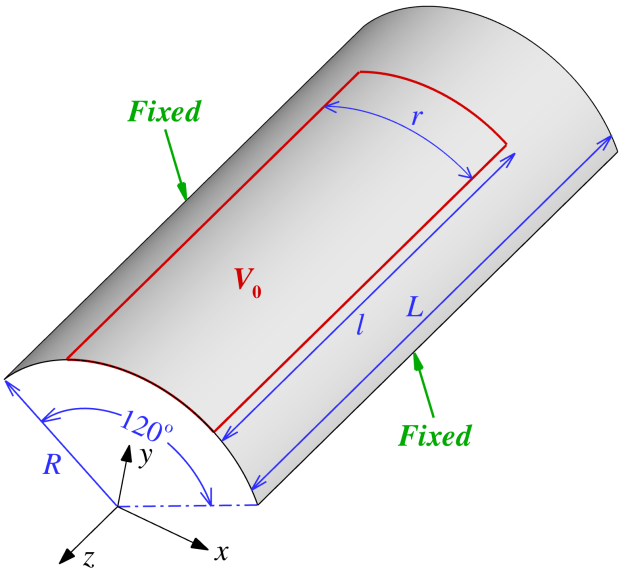
Fig. 2. The accuracy of the first eigenvalue for a square plate as a function of the degree of the NURBS basis functions and the number of control points.

maximum of the basis function. The orthogonal projection converges towards the result obtained by the lifting operator, but calculations with higher order NURBS ($p = 3$ and $p = 4$) indicate that it converges to an eigenvalue below the theoretical value, and the absolute value of the error increases past a certain point.

Given that the exact solution is based on thin plate theory, the response of the lifting operator therefore appears to be providing a very good approximation of thin shell theory. Since the accuracy of the lifting operator is clearly superior to the other choices, the remaining calculations are performed using it.

The accuracy of the rotation-free shell formulation as a function of the degree p is shown in Figure 2. For comparison, the results obtained using some of the more popular 4-node shear deformable shells implemented in LS-DYNA are plotted with them.

The Belytschko-Lin-Tsay element [9] uses one-point integration with hourglass control. Several forms of hourglass control are available with LS-DYNA, but no attempt was made to optimize the results by changing the default stiffness or formulation. The fully-integrated element (denoted "Type 16" in Figure 2) is based on the Hughes-Liu [26] formulation with some simplifications for speed [24], and is popular for implicit, large deformation calculations. Consistent and lumped mass matrices are available for these shells in LS-DYNA, and the consistent form was used for all the calculations in Figure 2. The converged value for the first frequency for the shear deformable shells is lower than the analytical thin plate value. The error, calculated using the thin plate value, changes sign as the mesh is refined. Its absolute value, used in the log-log graph, creates the illusion that the values for



$$
\begin{aligned}
L &= 12.56 \text{ in} \\
l &= 10.205 \text{ in} \\
R &= 3.0 \text{ in} \\
r &= 3.08 \text{ in} \\
h &= 0.125 \text{ in} \\
E &= 1.05 \times 10^7 \text{ psi} \\
\nu &= 0.33 \\
\rho &= 2.5 \times 10^{-4} \text{ lb-s}^2/\text{in}^4 \\
\sigma_y &= 4.4 \times 10^4 \text{ psi} \\
V_0 &= 5650 \text{ in/s}
\end{aligned}
$$

Fig. 3. Roof loaded by velocity impulse: problem description. The curved ends of the roof are hinged and the lateral boundaries are fixed.

shear deformable elements start to diverge at 500 nodes. The lumped mass calculations (not shown) do not exhibit this artifact because their first eigenvalues are consistently on one side of the thin plate solution for the range of meshes considered in this example, but the magnitude of their errors is larger than for the consistent mass calculations.

The convergence rates are, as expected, increasing monotonically with the degree of the basis functions. The quadratic, cubic, and quartic isogeometric elements are much more accurate than the traditional elements with linear basis functions. The displacements in the x, y-plane are zero for all the elements, and therefore there are three active degrees of freedom for the linear, shear deformable elements, and only one for the rotation-free elements.

For the finest meshes using quartic ($p = 4$) NURBS, the accuracy reaches the round-off error of the eigenvalue solver, which uses the block-shifted Lanczos method. To obtain this accuracy, a special version of LS-DYNA was created that wrote out the eigenvalues to fifteen decimal places instead of the default eight. Additionally, the value of π used in the exact solution, that is,

$$\pi \approx 3.14159265358979323846264338327 95, \tag{71}$$

was obtained from the article on π in Wikipedia [38] because the value obtained from $\pi \approx \cos^{-1}(-1)$ is not sufficiently accurate.

5.2 *Roof loaded by velocity impulse*

The influence of the integration rule on the nonlinear response of the rotation free shell is explored using the problem setup given in Figure 3. This problem was taken from [9] and consists of a 120° cylindrical panel loaded impulsively by specifying an initial velocity distribution. The problem dimensions and material data are summarized in Figure 3.

The normal is evaluated using the lifting operator and quadratic ($p = 2$) NURBS are used in the calculations.

An initial velocity normal to the shell surface is specified over a region marked on the figure. The normal velocity is specified on the control points by first extracting a consistent normal (see [23]) and then multiplying it with a prescribed velocity magnitude. The problem is solved on the entire domain with no symmetry assumptions.

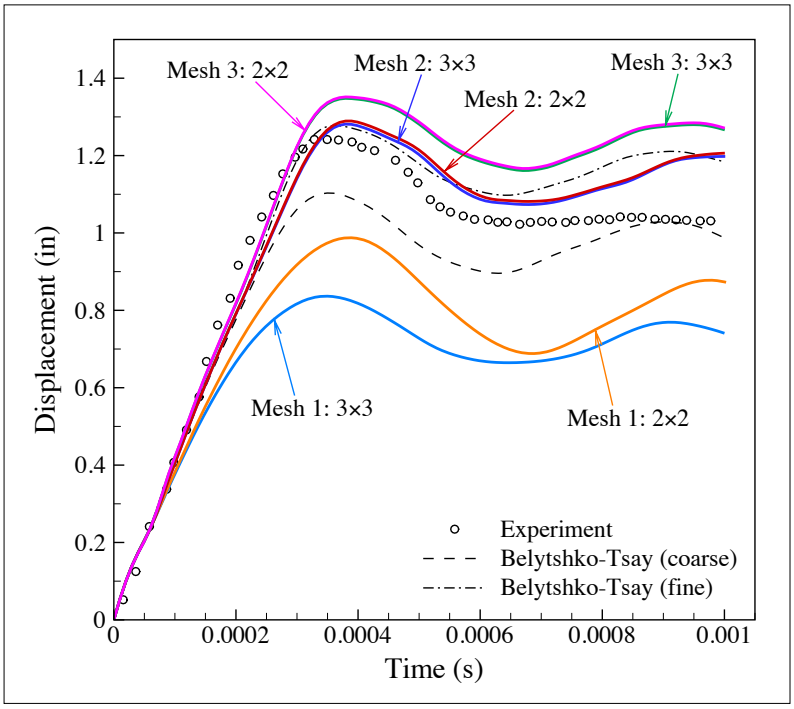


Fig. 4. Roof loaded by velocity impulse: displacement histories and convergence to the reference solution under $h-$refinement on C^1 meshes.

For convenience, the 3×3 and 2×2 lamina integration rules are referred to as "full" and "reduced" integraton, respectively. Note that element quadrature for isogeometric elements is an area of active research, e.g., [28], and therefore neither rule should be regarded as the "standard" one or as optimal. Three integration points are used in the through-thickness direction.

Displacement histories of the point, initially located at $x = 0$, $y = 3$, and $z = 6$, on

Table 1
Roof Results

Element Type	Number of Cntrl. Pnts.	Number of Elements	Integration Rule	Time Steps	CPU (seconds)	Maximum Displacement
NURBS	180	130	2×2	364	0.54	0.988
NURBS	180	130	3×3	367	0.81	0.836
NURBS	540	450	2×2	740	2.90	1.289
NURBS	540	450	3×3	743	5.28	1.281
NURBS	1836	1666	2×2	1502	20.87	1.351
NURBS	1836	1666	3×3	1502	36.92	1.348
B-T	191	224	1×1	578	0.16	1.103
B-T	4656	4512	1×1	2027	10.5	1.277

each mesh are shown in Figure 4 for the two rules. The results are compared with the reference computations employing two meshes of 224 and 4512 Belytschko-Tsay elements, as well as experimental data from [33]. Table 5.2 summarizes the results.

As might be expected for explicit time integration, the reduced integration rule is almost twice as fast as the full integration rule. There is, however, an overhead associated with the gather of the element data and the assembly of the force vector that prevents the element speed from being perfectly linear in the number of integration points. The reduced integration rule produces a larger displacement than the full rule for the coarsest mesh, denoted Mesh 1, but the difference disappears as the mesh is refined.

The medium NURBS mesh with 450 elements, denoted Mesh 2, has a peak displacement that is approximately the same as the reference solution obtained with 4656 Belytschko-Tsay elements, and the finest NURBS mesh has a peak displacement that is approximately five percent higher. In terms of CPU cost, the medium mesh with 3×3 integration is approximately twice as fast as the reference solution, and the 2×2 integration is more than three times as fast.

5.3 Metal Stamping

The last example is the S-rail benchmark stamping problem from the NUMISHEET 1996 Conference [32]. Our simulation is based on the LS-DYNA analysis performed by researchers at Alcoa [20], using their input file in its entirety except for the isogeometric elements modeling the blank.

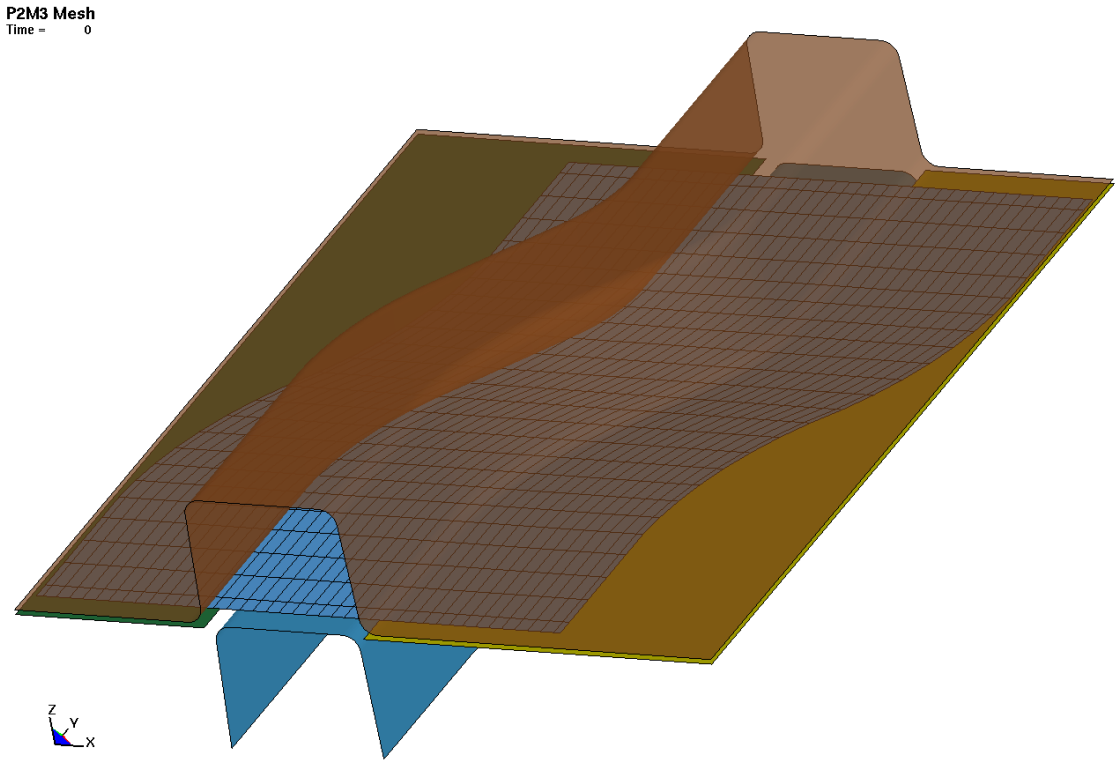


Fig. 5. The initial configuration of the NUMISHEET 1996 stamping problem. The upper die is semi-transparent to show the mesh for the blank.

An aluminum sheet metal blank of 6111-T4 aloy, 0.92 mm thick, is modeled with the transversely anisotropic plasticity model in LS-DYNA, using a Young's modulus of 69 GPa, Poisson's ration of 0.33, and an anisotropic hardening parameter of 0.64. The blank holder force is 10 kN, the Coulomb coefficient of friction is 0.1, and the punch stroke is 37 mm. The initial state is shown in Figure 5, with the upper die semi-transparent to show the mesh for the blank.

The original simulation from Alcoa used 42804 fully-integrated (type 16) shells in LS-DYNA, with 11550 elements in the blank and the remainder defining the geometry of the rigid body dies. Five integration points are used in the through-thickness direction.

Simulations were run using 240, 1092, 3840, and 7680 quadratic NURBS elements for the blank. The corresponding numbers of control points are 308, 960, 4100, and 8036. Note that the ratio of control points to the number of elements is much less than asymptotic value of 4 expected for traditional 9-node quadratic shells. The lamina integration rule is 2×2 Gauss and the normal is evaluated using the lifting operator. In these simulations, the time step size is governed by the penalty stiffness in the contact algorithm, and not the elements.

Plots of the final sheet, viewed from above, are shown in Figure 6. The vertical

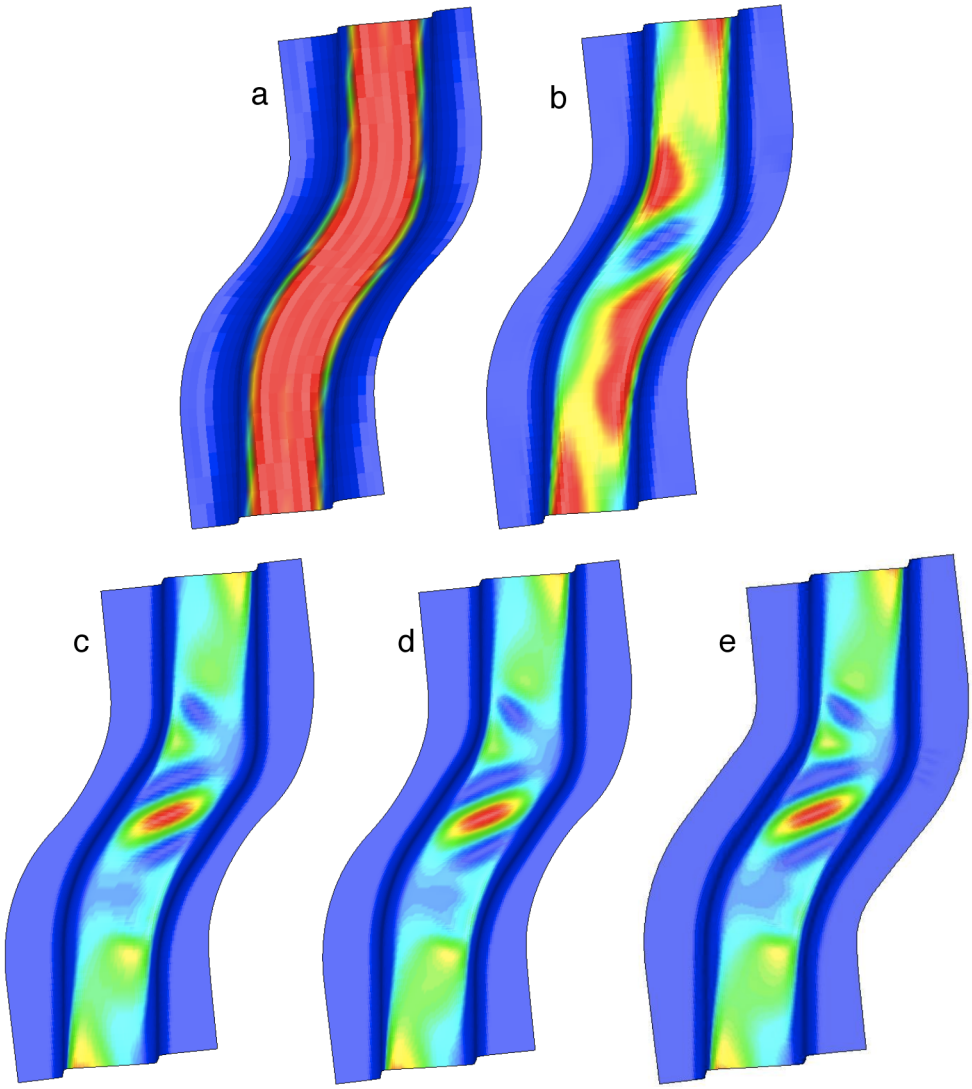


Fig. 6. The top view of the deformed blank for the NUMISHEET 1996 stamping problem: a) 240 quadratic NURBS elements, b) 1092 quadratic NURBS elements, c) 3840 quadratic NURBS elements, d) 7680 quadratic NURBS elements, and e) 11550 LS-DYNA type 16 elements.

displacement has been contoured over a range of 3 mm to highlight the wrinkling on the top of the S-rail. No wrinkling appears in the coarsest isogeometric mesh. The overall character of the wrinkling appears in the 1092 element mesh, however it has the opposite sign of the remaining isogeometric meshes and Alcoa's solution. The two finest isogeometric meshes capture the wrinkling accurately.

The equivalent plastic strain is shown in Figure 7. Alcoa obtained a maximum value of 0.291. The coarsest mesh predicts a peak value of only 0.198, about two-thirds of the reference solution. The peak plastic strains are 0.239, 0.290, and 0.289 for the 1092, 3840, and 7680 element meshes, respectively.

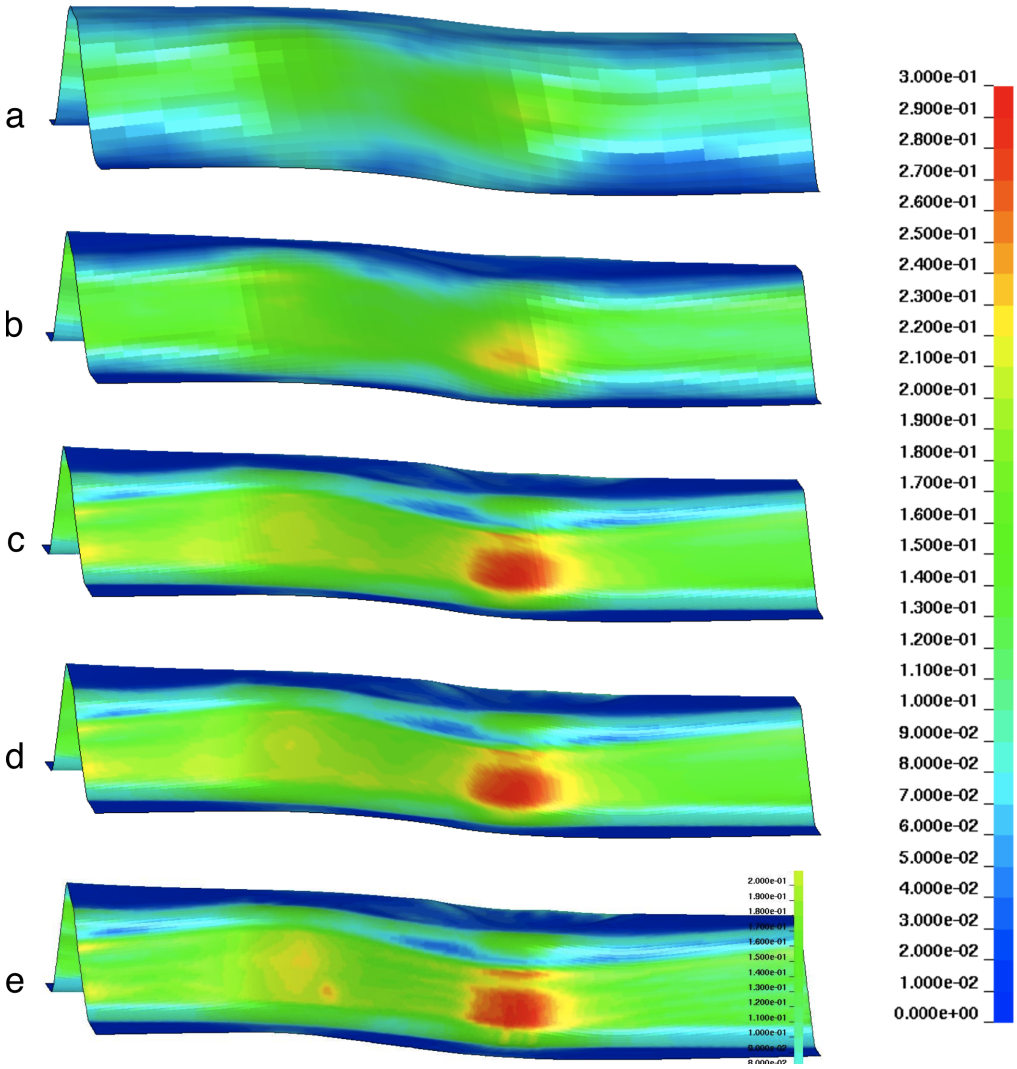


Fig. 7. The side view of the deformed blank for the NUMISHEET 1996 stamping problem a) 240 quadratic NURBS elements, b) 1092 quadratic NURBS elements, c) 3840 quadratic NURBS elements, d) 7680 quadratic NURBS elements, and e) 11550 LS-DYNA type 16 elements.

6 Conclusions

An accurate rotation-free isogeometric shell element has been developed.

The accuracy of the element is largely a function of the NURBS basis functions. Since the basis functions are not interpolatory, the notion of evaluating the normals at the control points is ambiguous. As demonstrated by the eigenvalue problem, a response that accurately approximates thin shell theory is obtained by using a simple lifting operator.

Efficiency for explicit calculations is obtained by using reduced integration, which is made possible by the intrinsic properties of the NURBS basis functions. Fur-

ther gains appear feasible [28]. For implicit analyses, where the linear algebra is frequently the dominant cost, having only half the usual number of degrees of freedom per control point offers another important gain in speed.

Acknowledgments

This research was supported by NSF grant 0700204 (Dr. Joycelyn S. Harrison, program officer) and ONR grant N00014-03-0236 (Dr. Luise Couchman, contract monitor).

References

[1] I. Akkerman, Y. Bazilevs, V.M. Calo, T.J.R. Hughes, and S. Hulshoff. The role of continuity in residual-based variational multiscale modeling of turbulence. *Computational Mechanics*, 41:371–378, 2008.

[2] F. Auricchio, L. Beirao da Veiga, A. Buffa, C. Lovadina, A. Reali, and G. Sangalli. A fully locking-free isogeometric approach for plane linear elasticity problems: A stream function formulation. 197:160–172, 2007.

[3] F. Auricchio, L. B. Da Veiga, T. J. R. Hughes, A. Reali, and G. Sangalli. Isogeometric collocation methods. *Mathematical Models and Methods in Applied Sciences*, in press, 2009.

[4] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 2008. Accepted for publication.

[5] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197:173–201, 2007.

[6] Y. Bazilevs, V.M. Calo, T.J.R. Hughes, and Y. Zhang. Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Computatonal Mechanics*, 43:3–37, 2008.

[7] Y. Bazilevs, V.M. Calo, Y. Zhang, and T.J.R. Hughes. Isogeometric fluid-structure interaction analysis with applications to arterial blood flow. *Computational Mechanics*, 38:310–322, 2006.

[8] Y. Bazilevs and T.J.R. Hughes. NURBS-based isogeometric analysis for the computation of flows about rotating components. *Computatonal Mechanics*, 43:143–150, 2008.

[9] T. Belytschko, J. I. Lin, and C. S. Tsay. Explicit algorithms for the nonlinear dynamics of shells. *Computer Methods in Applied Mechanics and Engineering*, 42:225–251, 1984.

[10] D. J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering*, 99:2356–394, 1992.

[11] D. J. Benson. Stable time step estimation for multi-material Eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering*, 167:191–205, 1998.

[12] D. J. Benson, Y. Bazilevs, E. De Luycker, M.-C. Hsu, M. Scott, T. J. R. Hughes, and T. Belytschko. A generalized finite element formulation for arbitrary basis functions: from isogeometric analysis to xfem. *International Journal for Numerical Methods in Engineering*, submitted for review, 2009.

[13] D. J. Benson, Y. Bazilevs, M. C. Hsu, and T. J. R. Hughes. Isogeometric shell analysis: The Reissner-Mindlin shell. *Computer Methods in Applied Mechanics and Engineering*, page DOI information: 10.1016/j.cma.2009.05.011, 2009.

[14] F. Cirak and M. Ortiz. Fully C^1-conforming subdivision elements for finite deformation thin shell analysis. *International Journal of Numerical Methods in Engineering*, 51:813–833, 2001.

[15] F. Cirak, M. Ortiz, and P. Schröder. Subdivision surfaces: a new paradigm for thin shell analysis. *International Journal of Numerical Methods in Engineering*, 47:2039–2072, 2000.

[16] F. Cirak, M. J. Scott, E. K. Antonsson, M. Ortiz, and P. Schröder. Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision. *Computer-Aided Design*, 34:137–148, 2002.

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

[18] J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195:5257–5297, 2006.

[19] M.R. Dörfel, B. Jüttler, and B. Simeon. Adaptive isogeometric analysis by local $h-$refinement with T-splines. *Computer Methods in Applied Mechanics and Engineering*, 2009. Published online. doi:10.1016/j.cma.2008.07.012.

[20] R. Dick. personal communication, Alcoa, 2009.

[21] T. Elguedj, Y. Bazilevs, V.M. Calo, and T. J. R. Hughes. $\bar{B}$ and $\bar{F}$ projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order NURBS elements. *Computer Methods in Applied Mechanics and Engineering*, 197(33-40), 2008.

[22] D. P. Flanagan and T. Belytschko. A uniform strain hexahedron and quadrilateral with orthogonal hourglass control. *International Journal for Numerical Methods in Engineering*, 17:679–706, 1981.

[23] P. M. Gresho and R. L. Sani. *Incompressible Flow and the Finite Element Method*. Wiley, New York, NY, 1998.

[24] J. O. Hallquist. LS-DYNA theoretical manual. Technical report, Livermore Software Technology Corporation, 1998.

[25] T. J. R. Hughes. *The Finite Element Method, Linear Static and Dynamic Finite Element Analysis*. Dover, 2000.

[26] T. J. R. Hughes and W.K. Liu. Nonlinear finite element analysis of shells: Part I. Three-dimensional shells. *Computer Methods in Applied Mechanics and Engineering*, 26:331–362, 1981.

[27] T. J. R. Hughes, K. S. Pister, and R. L. Taylor. Implicit-explicit finite elements in nonlinear transient analysis. *Computer Methods in Applied Mechanics and Engineering*, 17/18:159–182, 1979.

[28] T. J. R. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, In Press, Corrected Proof:–, 2008.

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

[30] T.J.R. Hughes, A. Reali, and G. Sangalli. Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: Com-

parison of $p-$method finite elements with $k-$method NURBS. *Computer Methods in Applied Mechanics and Engineering*, 197:4104–4124, 2008.

[31] B. M. Irons. Applications of a theorem on eigenvalues to finite element problems. Technical Report CR/132/70, University of Wales, Department of Civil Engineering, Swansea, U. K., 1970.

[32] J.K. Lee, G.L. Kinzel, and R.H. Wagoner. *NUMISHEET 96 3rd International Conference: Numerical Simulation of 3-D Sheet Metal Forming Processes Verification of Simulations and Experiments*. Ohio State University, 1996.

[33] L. Morino, J.W. Leech, and E.A. Wittmer. An improved numerical calculation technique for large elastic-plastic transient deformations of thin shells: Part 2 – Evaluation and applications. *Journal of Applied Mechanics*, pages 429–435, 1971.

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

[35] S. Schoenfeld and D. J. Benson. Quickly convergent integraton methods for plane stress plasticity. *Communications in Applied Numerical Methods*, 9:293–305, 1993.

[36] J. C. Simo and T. J. R. Hughes. *Computational Inelasticity*. Springer, New York, 2000.

[37] W. A. Wall, M. A. Frenzel, and C. Cyron. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197:2976–2988, 2008.

[38] Wikipedia.org. Pi. http://en.wikipedia.org/wiki/Pi.

[39] J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Oxford University Press, 1965.