Goal-adaptive Meshing of Isogeometric Kirchhoff–Love Shells

Verhelst, H. M.; Mantzaflaris, A.; Möller, M.; Den Besten, J. H.

DOI
10.1007/s00366-024-01958-4

Publication date
2024

Document Version
Final published version

Published in
Engineering with Computers

Engineering with Computers
https://doi.org/10.1007/s00366-024-01958-4

ORIGINAL ARTICLE

Goal-adaptive Meshing of Isogeometric Kirchhoff–Love Shells

H. M. Verhelst[1,2] · A. Mantzaflaris[3] · M. Möller[2] · J. H. Den Besten[1]

Received: 14 July 2023 / Accepted: 11 February 2024

Abstract

Mesh adaptivity is a technique to provide detail in numerical solutions without the need to refine the mesh over the whole domain. Mesh adaptivity in isogeometric analysis can be driven by Truncated Hierarchical B-splines (THB-splines) which add degrees of freedom locally based on finer B-spline bases. Labeling of elements for refinement is typically done using residual-based error estimators. In this paper, an adaptive meshing workflow for isogeometric Kirchhoff–Love shell analysis is developed. This framework includes THB-splines, mesh admissibility for combined refinement and coarsening and the Dual-Weighted Residual (DWR) method for computing element-wise error contributions. The DWR can be used in several structural analysis problems, allowing the user to specify a goal quantity of interest which is used to mark elements and refine the mesh. This goal functional can involve, for example, displacements, stresses, eigenfrequencies etc. The proposed framework is evaluated through a set of different benchmark problems, including modal analysis, buckling analysis and non-linear snap-through and bifurcation problems, showing high accuracy of the DWR estimator and efficient allocation of degrees of freedom for advanced shell computations.

Keywords Isogeometric analysis · Buckling · Dual-weighted residual method · Adaptive meshing · Kirchhoff–Love shell

1 Introduction

The idea behind isogeometric analysis (IGA) [1] is to bridge the gap between computer aided design (CAD) and finite element analysis (FEA). By employing B-splines or Non-uniform rational B-splines (NURBS) as the basis for FEA, IGA not only provides geometrically exact analysis, but the high smoothness of the spline bases also provides high accuracy per degree of freedom [2]. The close link with conventional engineering fields such as automotive, offshore, aircraft or civil engineering makes structural analysis with isogeometric analysis a particular field of interest. Besides the performance of the different isogeometric element formulations for thin Kirchhoff–Love shells [3–6], moderately thick Reissner-Mindlin shells [7–11] or thicker solid-like shells [12, 13] in static and dynamic simulations, conventional engineering disciplines also rely on accurate modal and (post-)buckling simulations. In addition, the ability to handle complex (multipatch) CAD geometries via trimming [14–16] or patch coupling methods [17–21] improves the applicability of IGA in structural engineering. For problems with a large number of degrees of freedom or problems with a large number of load/time steps, mesh adaptivity can play a key role in providing efficient simulations for industrial applications.

A loop in an adaptive isogeometric method (AIGM) consists of the steps *solve* the Partial Differential Equation (PDE) at hand, *estimate* element-wise error contributions, *mark* regions for refinement, *refine (coarsen)* marked regions [22], see Fig. 1. Here, localised regions can be defined

H. M. Verhelst
h.m.verhelst@tudelft.nl

A. Mantzaflaris
angelos.mantzaflaris@inria.fr

M. Möller
m.moller@tudelft.nl

J. H. Den Besten
henk.denbesten@tudelft.nl

[1] Department of Maritime and Transport Technology, Delft University of Technology, Mekelweg 2, Delft 2628 CD, The Netherlands

[2] Department of Applied Mathematics, Delft University of Technology, Mekelweg 2, Delft 2628 CD, The Netherlands

[3] AlgebRa, geOmetry, Modeling and AlgoriTHms, Inria Sophia Antipolis - Méditerranée, Université Côte d'Azur, 2004 route des Lucioles, Sophia Antipolis cedex 06902, France

Published online: 31 March 2024

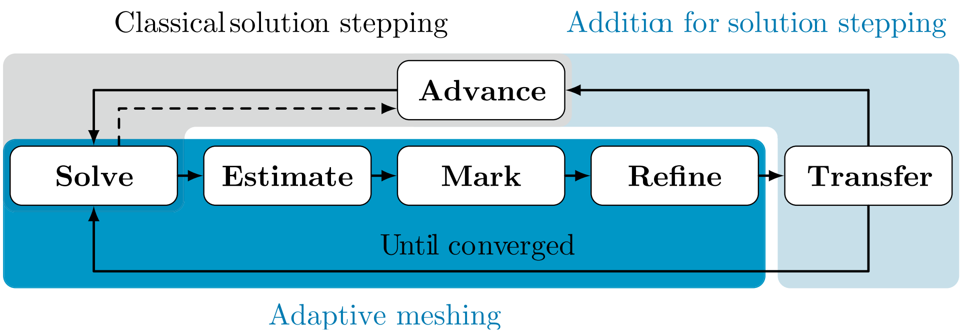


Fig. 1 A typical flowchart for an adaptive meshing routine. The *classical solution stepping* depicts a process without adaptive meshing. Here, a solution is obtained by the *solve* and the solution is advanced (e.g. in time or load step) and recomputed. The *adaptive meshing step* denotes the additional operations for mesh adaptivity and the *Addition for solution stepping* includes an additional *transfer* step in case the adaptive meshing method is applied to solution. The *Estimate* block provides an error estimation with local contributions per element or per degree of freedom (DoF). The *Mark* block contains a marking rule that marks regions for refinement based on a specific rule. The *Refine* block transforms the current mesh to a new mesh, where regions are refined and coarsened based on the marking rule. The block *Transfer* transfers the previous solution to the new mesh, so that it can be used to recompute the present interval on a modified mesh. This recomputation is performed again in the *Solve* block and follows through the subsequent blocks, until an adaptivity criterion is reached. For example, a criterion that requires the total error in the mesh to be within certain bounds

element-wise or function-wise. The AIGM process can be repeated in an iterative manner (e.g. for static, buckling or modal analysis), until satisfactory accuracy is achieved or it can be applied (iteratively) within a time/load stepping procedure. A broad overview of the mathematical foundations of AIGMs is given in [23]. In previous works, AIGMs are developed for different applications (*solve*), using different *estimation* strategies, *marking* strategies and often for mesh *refinement*, with a few also providing *coarsening* strategies [24–28].

Solve The *solve* block contains the partial differential equation (PDE) at hand. It can be a physics-based problem, e.g., to solve shell [14, 29, 30], linear elasticity [31] or free-surface flow [32] problems. Alternatively, the solve step can involve a non-physics PDE, e.g. for mesh generation [33].

Estimate Determination of localised errors is done in the *estimate* block. In the works [14, 29, 30], an error estimator based on a residual-like variational problem in the so-called *bubble-space* was presented for Kirchhoff plates, Kirchhoff–Love shells and trimmed domains. This method has proven a large decrease in CPU time compared to a residual-based error estimator in the strong form, due to its easy parallelisation and the small block-structure of the linear system to solve. As an alternative to this method, error estimation can also be performed in a goal-oriented fashion, e.g., by the Dual-Weighted Residual (DWR) method. This method has been applied in the FEA context in various works [34–41] and was used in the works [32, 42] for Poisson and free-boundary problems, in [43] for a geometrically non-linear rod, in [33] for PDE-based domain parametrisations and in [31] for micromechanical modelling of trabecular bone. Goal-oriented refinement in general provides localised error estimates by solving a linear adjoint problem on the current space and an enriched space.

Mark As soon as localised error contributions are known, regions can be marked for refinement. This marking is mostly done using the Dörfler marking strategy [44], as in [22, 25, 35], which involves marking the regions with the largest error contributions until their sum exceeds a certain percentage of the total error. An alternative is to mark the regions with an error higher than a threshold (an absolute threshold based on the maximum error) [29, 45] or based on a relative threshold taking a fixed percentage of the total number of cells for refinement. In [45] the latter two strategies are discussed.

Refine Local refinement for adaptive meshing in isogeometric analysis is enabled by the use of Hierarchical B-splines (HB-splines) [46], Truncated

Hierarchical B-splines (THB-splines) [45, 47] or T-splines [48] amongst other spline constructions, which are reviewed in [49]. HB-splines provides a nested, linear dependent space that violates the partition of unity property. To preserve the latter, THB-splines have been introduced in [47]. For (T)HB-splines *suitable grading* to generate admissible meshes should be taken into account in order to guarantee a bounded error [22], for which algorithms have been presented in [50]. In the work of [51], a distinction is made between *greedy* and *safe* refinement, the former being a refinement of cells with a 1-level difference with adjacent cells and the latter being a refinement complying with the refinement neighborhoods defined in [22]. Besides for adaptive meshing for solving PDEs [24–28, 33], THB-splines have also been successfully applied in the context of fitting [52, 53].

Transfer The *transfer* from previous time/load-steps onto a new mesh can be done using different methodologies. In the work of [27] different least-squares approaches are provided. Furthermore, quasi-interpolation [54, 55] is a technique that can be used to transfer solutions between hierarchical meshes.

In this paper, we employ goal-oriented adaptive refinement for isogeometric thin shell analysis to facilitate THB-adaptive meshing for a variation of structural analysis problems. The developed framework is versatile in terms of the goal functional being used and provides an adaptive meshing strategy for linear an non-linear static, modal, buckling and post-buckling problems. In brief, the contribution of the paper is threefold. Firstly, we use the Dual-Weighted Residual (DWR) method to derive novel error estimators for structural shell analysis, given goal functionals based on displacements, (principal) stresses and strains, forces and moments. Secondly, we employ the eigenvalue DWR from [34, 56] for error estimations for modal and buckling analyses. Thirdly, the goal functionals are used to drive an adaptive meshing strategy with suitable grading and efficient transfer of solutions by quasi-interpolation method on hierarchical spline spaces [54, 55]. This adaptive meshing strategy is applied to non-linear shell analysis with focus on buckling problems with snap-through and bifurcation instabilities - being new applications in the realm of adaptive meshing research for nonlinear shell problems. It should be noted that the present framework is developed for isogeometric Kirchhoff–Love shells - since it provides a natural separation of bending and membrane terms - but it is easily adapted for other shell formulations. By defining a frame work for 2-dimensional parametric domains and by presenting a wide range of mechanics-inspired goal functionals, the present work extends an earlier work by [43] for geometrically non-linear rods.

The paper is structured as follows. In Sect. 2, the isogeometric Kirchhoff–Love shell analysis proposed by [3] is briefly revised and some basic concepts for structural analysis computations are given. In Sect. 3, the Dual-Weighted Residual (DWR) method is provided for the isogeometric Kirchhoff–Love shell using the membrane and flexural strain split. However, it can be used for general elasticity problems. Moreover, the section provides the DWR method for eigenvalue problems to compute error estimators for modal and buckling analyses. Thereafter, Sect. 4 provides the details for adaptivity for isogeometric analysis. This includes the concept of Truncated-Hierarchical B-splines (THB-splines) and admissible refinement. Furthermore, the *mark* and *transfer* operations are described. In Sect. 5, a summary of the preceeding sections is provided by means of a global algorithm for the AIGM for structural analysis computations with load-stepping. In Sect. 6 the present work is evaluated on numerical benchmark problems, ranging from linear problems with analytical solutions to non-linear shell problems. Finally, Sect. 7 provides conclusions and and outlook based on this work.

2 Isogeometric Kirchhoff–Love shell analysis

In this section, we provide a brief background on the Kirchhoff–Love shell formulation. For more details on this formulation, we refer to [3–5, 57, 58].

Check punctuation!!

2.1 Shell kinematics

Since Kirchhoff–Love shells satisfy the Kirchhoff Hypothesis [59], the coordinates $\mathbf{x}$ of any parametric point $\boldsymbol{\theta} = (\theta^1, \theta^2, \theta^3)$ in the shell surface can be represented by the surface position $\mathbf{r}(\theta^1, \theta^2)$ and contribution in normal direction $\theta^3 \mathbf{a}_3$ as

$$\mathbf{x}(\boldsymbol{\theta}) = \mathbf{r}(\theta^1, \theta^2) + \theta^3 \mathbf{a}_3, \tag{1}$$

Given the covariant basis of the surface $\mathbf{r}$, defined by $\mathbf{a}_\alpha, \alpha = 1, 2$ and the orthogonal unit normal $\mathbf{a}_3$, the covariant basis of $\mathbf{x}$ is defined as follows:

$$\mathbf{g}_\alpha = \mathbf{x}_{,\alpha} = \mathbf{a}_\alpha + \theta^3 \mathbf{a}_{3,\alpha}, \quad \mathbf{g}_3 = \mathbf{x}_{,3} = \mathbf{a}_3. \tag{2}$$

Given the second fundamental form $b_{\alpha\beta} = \mathbf{a}_3 \cdot \mathbf{a}_{\alpha,\beta} = -\mathbf{a}_{\,3,\beta} \cdot \mathbf{a}_\alpha$ and the metric coefficients defined as

$$g_{\alpha\beta} = \mathbf{g}_\alpha \cdot \mathbf{g}_\beta = a_{\alpha\beta} - 2\theta^3 b_{\alpha\beta}, \tag{3}$$

the contravariant basis vectors $\mathbf{g}^\alpha$ can simply be obtained by $\mathbf{g}^\alpha = g^{\alpha\beta}\mathbf{g}_\beta$. The undeformed configuration $\mathbf{r}$ and the deformed configuration $\mathring{\mathbf{r}}$ of the surface are related by $\mathbf{r} = \mathring{\mathbf{r}} + \mathbf{u}$. From the defintion of the deformation gradient $\mathbf{F} = \mathbf{g}_i \otimes \mathring{\mathbf{g}}^i$, the deformation tensor $\mathbf{C}$ can be obtained:

$$\mathbf{C} = \mathbf{F}^\top \mathbf{F} = \mathbf{g}_i \cdot \mathbf{g}_j \, \mathring{\mathbf{g}}^i \otimes \mathring{\mathbf{g}}^j = g_{ij} \, \mathring{\mathbf{g}}^i \otimes \mathring{\mathbf{g}}^j. \tag{4}$$

Note that the deformation tensor is defined in the contravariant undeformed basis $\mathring{\mathbf{g}}^i \otimes \mathring{\mathbf{g}}^j$. For Kirchhoff–Love shells, it is known that $g_{\alpha 3} = g_{3\alpha} = 0$, hence this implies $C_{\alpha 3} = C_{3\alpha} = 0$, since $g_{33} = 1$, which implies C_{33} to be one and meaning that the thickness remains constant under deformation. As a result, the Green-Lagrange strain tensor $\mathbf{E} = E_{\alpha\beta} \, \mathring{\mathbf{g}}^\alpha \otimes \mathring{\mathbf{g}}^\beta$ and its decomposition to membrane and bending contributions (ε and κ, respectively) is [3, 4]:

$$\begin{aligned} E_{\alpha\beta} &= \frac{1}{2}\left(g_{\alpha\beta} - \mathring{g}_{\alpha\beta}\right) = \frac{1}{2}\left((a_{\alpha\beta} - \mathring{a}_{\alpha\beta}) - 2\theta^3\left(b_{\alpha\beta} - \mathring{b}_{\alpha\beta}\right)\right) \\ &= \varepsilon_{\alpha\beta} + \theta_3 \kappa_{\alpha\beta}. \end{aligned} \tag{5}$$

2.2 Constitutive relation

The constitutive relations for the Kirchhoff–Love shell relate the Green-Lagrange strain tensor $\mathbf{E}$ to the second Piola-Kirchhoff stress tensor $\mathbf{S}$. For linear elastic materials, this is achieved by:

$$S^{\alpha\beta} = \mathbb{C}^{\alpha\beta\gamma\delta} E_{\gamma\delta} \tag{6}$$

where $\mathbb{C} = \mathbb{C}^{\alpha\beta\gamma\delta} \, \mathring{\mathbf{g}}_i \otimes \mathring{\mathbf{g}}_j \otimes \mathring{\mathbf{g}}_k \otimes \mathring{\mathbf{g}}_l$ is the material tensor, which takes for linear materials the form $\mathbb{C}^{\alpha\beta\gamma\delta} = 4\frac{\lambda\mu}{\lambda+2\mu} \mathring{g}^{\alpha\beta}\mathring{g}^{\gamma\delta} + 2\mu\left(\mathring{g}^{\alpha\delta}\mathring{g}^{\beta\gamma} + \mathring{g}^{\alpha\gamma}\mathring{g}^{\beta\delta}\right)$ [60]. For nonlinear hyperelastic constitutive relations, the stress and material tensors are derived from the 3D constitutive relations for (in)compressible materials and due to through-thickness integration, the shell normal force and bending moment tensors $\boldsymbol{n} = n^{\alpha\beta} \, \mathring{\mathbf{g}}_\alpha \otimes \mathring{\mathbf{g}}_\beta$ and $\boldsymbol{m} = m^{\alpha\beta} \, \mathring{\mathbf{g}}_\alpha \otimes \mathring{\mathbf{g}}_\beta$, respectively, are defined as

$$n^{\alpha\beta}(\mathbf{u}) = \int_T S^{\alpha\beta}(\mathbf{u}) \mathrm{d}\theta^3, \quad m^{\alpha\beta}(\mathbf{u}) = \int_T \theta^3 S^{\alpha\beta}(\mathbf{u}) \mathrm{d}\theta^3, \tag{7}$$

where $T = [-t/2, t/2]$ is the through-thickness domain. For more details on hyperelastic material models, the reader is referred to [4, 57] and specifically for stretch-based ones to [5].

2.3 Variational formulation

The shell internal and external equilibrium equations in variational form are derived by the principle of virtual work [3, 4]. The weak formulation follows from the principle of virtual work with virtual displacements $\boldsymbol{\phi}$:

$$\begin{aligned} &\text{Find } \mathbf{u} \in \boldsymbol{\mathcal{S}} \text{ s.t.} \boldsymbol{\mathcal{W}}(\mathbf{u}, \boldsymbol{\phi}) := \delta W^{\text{int}} - \delta W^{\text{ext}} \\ &\quad = \int_\Omega \boldsymbol{n}(\mathbf{u}) : \boldsymbol{\varepsilon}'(\mathbf{u}, \boldsymbol{\phi}) + \boldsymbol{m}(\mathbf{u}) : \boldsymbol{\kappa}'(\mathbf{u}, \boldsymbol{\phi}) \mathrm{d}\Omega - \int_\Omega \mathbf{f}(\mathbf{u}) \cdot \boldsymbol{\phi} \mathrm{d}\Omega, \\ &\quad \forall \boldsymbol{\phi} \in \boldsymbol{\mathcal{S}} \end{aligned} \tag{8}$$

With $\mathbf{f}(\mathbf{u})$ the surface load acting on the mid-surface, for the sake of generality defined as a function of the displacements $\mathbf{u}$ (e.g. a follower pressure p gives $\mathbf{f}(\mathbf{u}) = p\mathbf{a}_3(\mathbf{u})$). Furthermore, $\boldsymbol{\varepsilon}'(\mathbf{u}, \boldsymbol{\phi})$ and $\boldsymbol{\kappa}'(\mathbf{u}, \boldsymbol{\phi})$ are the virtual strain components given displacements $\mathbf{u}$ and being linear with respect to variation $\boldsymbol{\phi}$, hence $\boldsymbol{\mathcal{W}}(\mathbf{u}, \boldsymbol{\phi})$ is also linear in its second argument. The coefficients of the variations of the membrane force and bending moment tensors are

$$\begin{aligned} (n')^{\alpha\beta}(\mathbf{u}, \boldsymbol{\phi}) &= \int_T \mathbb{C}^{\alpha\beta\gamma\delta}(\mathbf{u}) \mathrm{d}\theta^3 \varepsilon'_{\gamma\delta}(\mathbf{u}, \boldsymbol{\phi}) \\ &\quad + \int_T \theta^3 \mathbb{C}^{\alpha\beta\gamma\delta}(\mathbf{u}) \mathrm{d}\theta^3 \kappa'_{\gamma\delta}(\mathbf{u}, \boldsymbol{\phi}), \\ (m')^{\alpha\beta}(\mathbf{u}, \boldsymbol{\phi}) &= \int_T \theta^3 \mathbb{C}^{\alpha\beta\gamma\delta}(\mathbf{u}) \mathrm{d}\theta^3 \varepsilon'_{\gamma\delta}(\mathbf{u}, \boldsymbol{\phi}) \\ &\quad + \int_T \left(\theta^3\right)^2 \mathbb{C}^{\alpha\beta\gamma\delta}(\mathbf{u}) \mathrm{d}\theta^3 \kappa'_{\gamma\delta}(\mathbf{u}, \boldsymbol{\phi}). \end{aligned} \tag{9}$$

Linearizing the virtual work from Eq. (8) provides the continuous equivalent of the Jacobian or *tangential stiffness matrix* for Newton iterations which will be performed to solve the non-linear weak formulation Eq. (8) in a discrete setting [58]:

$$\begin{aligned} \boldsymbol{\mathcal{W}}'(\mathbf{u}, \boldsymbol{\phi}, \boldsymbol{\psi}) &:= \int_\Omega \boldsymbol{n}'(\mathbf{u}, \boldsymbol{\psi}) : \boldsymbol{\varepsilon}'(\mathbf{u}, \boldsymbol{\phi}) + \boldsymbol{n}(\mathbf{u}) : \boldsymbol{\varepsilon}''(\mathbf{u}, \boldsymbol{\phi}, \boldsymbol{\psi}) \\ &\quad + \boldsymbol{m}'(\mathbf{u}, \boldsymbol{\psi}) : \boldsymbol{\kappa}'(\mathbf{u}, \boldsymbol{\phi}) + \boldsymbol{m} : \boldsymbol{\kappa}''(\mathbf{u}, \boldsymbol{\phi}, \boldsymbol{\psi}) \mathrm{d}\Omega \\ &\quad - \int_\Omega \mathbf{f}'(\mathbf{u}, \boldsymbol{\psi}) \cdot \boldsymbol{\phi} \mathrm{d}\Omega, \end{aligned} \tag{10}$$

where $\boldsymbol{\varepsilon}''(\mathbf{u}, \boldsymbol{\phi}, \boldsymbol{\psi})$ and $\boldsymbol{\kappa}(\mathbf{u}, \boldsymbol{\phi}, \boldsymbol{\psi})$ are the second variations of the membrane and bending strains and $\mathbf{f}'$ is the first variation of the applied force, being nonzero when the force is depending on the displacements $\mathbf{u}$. For the details on these formulations, we refer to previous publications [3, 58, 60]. It should be noted that in the undeformed case, $\mathbf{u} = \mathbf{0}$, the internal membrane forces and bending forces, $\boldsymbol{n}(\mathbf{u})$ and $\boldsymbol{m}(\mathbf{u})$, respectively, vanish. As a result, the continuous equivalent for the linear stiffness matrix is:

$$\mathring{\mathcal{W}}'(\boldsymbol{\phi}, \boldsymbol{\psi}) = \int_\Omega \mathring{\boldsymbol{n}}'(\boldsymbol{\psi}) : \mathring{\boldsymbol{\varepsilon}}'(\boldsymbol{\phi}) + \mathring{\boldsymbol{m}}'(\boldsymbol{\psi}) : \mathring{\boldsymbol{\kappa}}'(\boldsymbol{\phi})\mathrm{d}\Omega - \int_\Omega \mathring{\mathbf{f}}'(\boldsymbol{\psi}) \cdot \boldsymbol{\phi}\mathrm{d}\Omega, \tag{11}$$

where the $\mathring{\cdot}$ denotes tensors and functions on the undeformed geometry, i.e. with $\mathbf{u} = \mathbf{0}$.

In our implementation, the tangent stiffness matrix is computed using appropriate Gauss–Lengendre quadrature for each element in the hierarchical mesh. We note that more efficient numerical integration approaches exist [61–63] for hierarchical splines that might further reduce the computational cost.

2.4 Structural analysis

In the present paper, we will provide different goal functionals for different structural analysis applications. Therefore, we briefly recall the different structural analysis types. Firstly, in case of *static analysis*, the problem as in Eq. (8) is solved. In case of *quasi-static analysis*, load and/or displacement steps are performed successively and in each step, a static solve is performed. Typically, one writes Eq. (8) for load-control as

$$\begin{aligned}&\text{Find } \mathbf{u} \in \mathcal{S} \text{ s.t.} \mathcal{W}(\mathbf{u}, \boldsymbol{\phi}, \lambda) := \delta W^{\text{int}} - \lambda \delta W^{\text{ext}} \\ &\quad = \int_\Omega \boldsymbol{n}(\mathbf{u}) : \boldsymbol{\varepsilon}'(\mathbf{u}, \boldsymbol{\phi}) + \boldsymbol{m}(\mathbf{u}) : \boldsymbol{\kappa}'(\mathbf{u}, \boldsymbol{\phi})\mathrm{d}\Omega - \int_\Omega \lambda \mathbf{f}_0 \cdot \boldsymbol{\phi}\mathrm{d}\Omega, \\ &\quad \forall \boldsymbol{\phi} \in \mathcal{S}, \lambda \in \mathbb{R},\end{aligned} \tag{12}$$

where λ is the load factor scaling the reference load $\mathbf{f}_0$. Quasi-static simulations can be solved using simple load or displacement controlled schemes, using arc-length continuation such as Riks' method or Crisfield's method [64, 65]. When quasi-static analysis is performed for post-buckling analysis, one or multiple bifurcation points are passed by definition. On a bifurcation point, the determinant of the tangential stiffness matrix K is equal to zero, hence this matrix is singular. To cope with instabilities, *a priori* perturbations can be applied to the geometry, or a procedure for approximating singular points [66] can be used. In our previous work, we provide more details on arc-length continuation for post-buckling analysis without providing *a priori* perturbations [67].

In the case of *modal analysis* and *buckling analysis*, a generalised eigenvalue problem needs to be solved. These eigenvalue problems have the general form

$$\text{find } (\mu, \mathbf{v}) \in \mathbb{R} \times \mathcal{S} \text{ s.t. } \mathcal{A}(\mathbf{v}, \boldsymbol{\phi}) = \mu \mathcal{B}(\mathbf{v}, \boldsymbol{\phi}) \, \forall \boldsymbol{\phi} \in \mathcal{S} \tag{13}$$

Where μ provides the eigenfrequency in modal analysis and the critical load factor in buckling analysis and where $\mathbf{v}$ denotes the vibration or buckling mode shape. The operators $\mathcal{A}$ and $\mathcal{B}$ are bi-linear. For buckling analysis, $\mathcal{A}(\mathbf{v}, \boldsymbol{\phi}) = \mathcal{W}'(\mathbf{u}_L, \mathbf{v}, \boldsymbol{\phi})$ and $\mathcal{B}(\mathbf{v}, \boldsymbol{\phi}) = \mathcal{W}'(\mathbf{0}, \mathbf{v}, \boldsymbol{\phi})$ with $\mathbf{u}_L$ the pre-buckling solution given load λ_L. For modal analysis, $\mathcal{A}(\mathbf{v}, \boldsymbol{\phi}) = \mathcal{W}'(\mathbf{0}, \mathbf{v}, \boldsymbol{\phi})$ and $\mathcal{B}(\mathbf{v}, \boldsymbol{\phi}) = \mathcal{M}(\mathbf{v}, \boldsymbol{\phi})$ with $\mathcal{M}$ the mass operator:

$$\mathcal{M}(\mathbf{v}, \boldsymbol{\phi}) = \int_\Omega \rho \mathbf{v} \boldsymbol{\phi} \mathrm{d}\Omega \tag{14}$$

Where ρ is the density function over the surface.

3 Dual-weighted residual method

This section elaborates on the Dual-Weighted Residual (DWR) method [68, 69], which is used in the *Estimate* step of Fig. 1. The DWR is a method to compute the *a posteriori* error of a solution in terms of a given goal functional of interest, by solving a linear dual problem. The DWR provides a global estimate of the error, but given a partition of unity of the spline space, it can be used to provide an error contribution per basis function.

3.1 General framework

The general framework of the dual weighted residual (DWR) is presented by [37, 68, 69]. For the sake of completeness, we provide a brief overview of the DWR here. Consider the following non-linear problem to solve

$$\text{find } \mathbf{u} \in \mathcal{S} \text{ s.t. } \mathcal{W}(\mathbf{u}, \boldsymbol{\phi}) = 0 \, \forall \boldsymbol{\phi} \in \mathcal{S}, \tag{15}$$

where $\mathcal{W}(\mathbf{u})$ is a semi-linear operator, $\mathbf{u}$ is the solution, $\boldsymbol{\phi}$ is a test function and $\mathcal{S}$ is a suitably chosen vector space including $\mathbf{u} \in \mathcal{S}$. The approximation of $\mathbf{u}$, denoted by $\mathbf{u}_h$ can be found by solving the discrete counterpart of Eq. (15)

$$\text{find } \mathbf{u}_h \in \mathcal{S}_h^p \text{ s.t. } \mathcal{W}(\mathbf{u}_h, \boldsymbol{\phi}_h) = 0 \, \forall \boldsymbol{\phi} \in \mathcal{S}_h^p, \tag{16}$$

where $\mathbf{u}_h$ and $\boldsymbol{\phi}_h$ are the discrete counterparts of $\mathbf{u}$ and $\boldsymbol{\phi}$, respectively, and the space $\mathbf{S}_h^p \subset \mathbf{S}$ is a function space on the (isogeometric) mesh $\mathcal{T}_h^p(\Omega)$ with mesh size h and order p

covering the computational domain Ω. The solution to this problem is typically obtained by iteratively solving

$$\text{find } \boldsymbol{\phi}_h \in \mathcal{S}_h^p \text{ s.t. } \mathcal{W}'(\mathbf{u}_h, \boldsymbol{\phi}_h, \boldsymbol{\psi}_h) = \mathcal{R}(\mathbf{u}_h, \boldsymbol{\psi}_h)\, \forall \boldsymbol{\psi}_h \in \mathcal{S}_h^p, \tag{17}$$

while updating the discrete solution. Here, the residual is defined as

$$\mathcal{R}(\mathbf{u}_h, \boldsymbol{\phi}_h) = -\mathcal{W}(\mathbf{u}_h, \boldsymbol{\phi}_h). \tag{18}$$

Let us now define a non-linear and differentiable goal functional $\mathcal{L}(\mathbf{u})$ or quantity of interest, such that

$$\Delta\mathcal{L}(\mathbf{u}_h) = \mathcal{L}(\mathbf{u}) - \mathcal{L}(\mathbf{u}_h). \tag{19}$$

Then, from Proposition 4.1 of [70], it follows that:

$$\Delta\mathcal{L}(\mathbf{u}_h) = \mathcal{R}(\mathbf{u}_h, \xi - \xi_h) \approx \mathcal{R}(\mathbf{u}_h, \tilde{\xi} - \xi_h). \tag{20}$$

Here, the solutions $\xi \in \mathcal{S}$ and $\xi_h \in \mathcal{S}_h^p$ are the exact and discrete solutions to the adjoint problem defined using the mean value linearizations of $\mathcal{W}$ and $\mathcal{L}$, see Equations 10 to 12 of [70]. Sinze the exact dual solution ξ is not available, it is approximated by $\tilde{\xi} \in \tilde{\mathcal{S}}$. The discrete dual solution $\xi_h \in \mathcal{S}_h^p$ is obtained by solving the following discrete adjoint problem:

$$\text{find } \xi_h \in \mathcal{S}_h^p \text{ s.t. } \mathcal{W}'(\mathbf{u}_h, \zeta_h, \xi_h) = \mathcal{L}'(\mathbf{u}_h, \zeta_h)\, \forall \zeta_h \in \mathcal{S}_h^p. \tag{21}$$

The approximation $\tilde{\boldsymbol{\psi}} \in \tilde{\mathcal{S}}$ is obtained by solving the adjoint problem in an enriched space, i.e.

$$\text{find } \tilde{\xi}_h \in \tilde{\mathcal{S}}_h^p \text{ s.t. } \mathcal{W}'(\mathbf{u}_h, \tilde{\zeta}_h, \tilde{\xi}_h) = \mathcal{L}'(\mathbf{u}_h, \tilde{\zeta}_h)\, \forall \tilde{\zeta}_h \in \tilde{\mathcal{S}}_h^p, \tag{22}$$

with $\tilde{\xi}_h$ and $\tilde{\zeta}_h$ the dual solution and test functions on the enriched space $\tilde{\mathcal{S}}_h^p$, respectively. A choice for $\tilde{\mathcal{S}}_h^p$ is to use the same mesh as for $\mathcal{S}_h^p$, with the same regularity but with a higher degree, i.e. $\tilde{\mathcal{S}}_h^p = \mathcal{S}_h^{p+1}$. This is easily achieved using spline bases. When using B-Splines, one can repeat all knots of the knot vector an extra time compared to the original basis, such that $\mathcal{S}_h^p \subset \mathcal{S}_h^{p+1} \subset \mathcal{S}$ is a nested space.

Finally, using Eq. (20) together with the dual solution $\xi_h \in \mathcal{S}_h^p$ and the enriched dual solution $\tilde{\xi}_h \in \tilde{\mathcal{S}}_h^p$, an estimate for the global error with respect to the goal functional $\mathcal{L}$ can be obtained. To obtain the local element-wise error estimations r_i for element $\omega_i \in \mathcal{T}_h^p(\Omega)$, such that

$$\Delta\mathcal{L}(\mathbf{u}_h) = \mathcal{R}(\mathbf{u}_h, \tilde{\xi}_h - \xi_h) = \sum_{\omega_i \in \mathcal{T}_h^p(\Omega)} r_i, \tag{23}$$

element-wise integration of Eq. (20) is simply performed to obtain r_i. However, as discussed in Sect. 4.3 it can be beneficial to have strictly positive element error contributions for element labeling. One can either take the absolute values of r_i or one can integrate the squared norm of the integrant in Eq. (20) to ensure positivity of element error contributions. Obviously, the sum of the element errors would not be equal to $\Delta\mathcal{L}$.

For Kirchhoff–Love shells specifically, the operator $\mathcal{W}(\mathbf{u}, \boldsymbol{\phi})$ and its linearisation $\mathcal{W}'(\mathbf{u}, \boldsymbol{\phi}, \boldsymbol{\psi})$ are used to perform the DWR analysis.

3.2 Eigenvalue problems

When the problem of interest is an eigenvalue problem, the DWR routine is slightly different. Here, we follow the works [35, 41, 56, 68, 69, 71] to give a brief overview of the DWR for eigenvalue problems.

Let us consider the following eigenvalue problem

$$\text{find } (\mu, \mathbf{v}) \in \mathbb{R} \times \mathcal{S} \text{ s.t. } \mathcal{A}(\mathbf{v}, \boldsymbol{\phi}) = \mu\mathcal{B}(\mathbf{v}, \boldsymbol{\phi})\, \forall \boldsymbol{\phi} \in \mathcal{S}. \tag{24}$$

Here, $\mathcal{A}$ and $\mathcal{B}$ are bi-linear operators. For uniqueness of the problem, the discrete eigenvectors $\mathbf{v}_h$ are normalised by the condition [56]

$$\mathcal{B}(\mathbf{v}, \mathbf{v}) = 1. \tag{25}$$

Typically, discretizing the system gives the following:

$$\text{find } (\mu_h, \mathbf{v}_h) \in \mathbb{R} \times \mathcal{S}_h^p \text{ s.t. } \mathcal{A}(\mathbf{v}_h, \boldsymbol{\phi}_h) = \mu\mathcal{B}(\mathbf{v}_h, \boldsymbol{\phi}_h)\, \forall \boldsymbol{\phi}_h \in \mathcal{S}_h^p, \tag{26}$$

where the eigenpairs $\hat{\mathbf{v}}_h = (\mu_h, \mathbf{v}_h)$ are the solutions of the eigenvalue problem. In addition, the adjoint eigenvalue problem is defined by the eigenvalue problem [69]:

$$\text{find } (\eta, \boldsymbol{\psi}) \in \mathbb{R} \times \mathcal{S} \text{ s.t. } \mathcal{A}(\boldsymbol{\psi}, \boldsymbol{\phi}) = \eta\mathcal{B}(\boldsymbol{\psi}, \boldsymbol{\phi})\, \forall (\boldsymbol{\phi}) \in \mathcal{S}, \tag{27}$$

Of for which the normalization similar to Eq. (25) is used for the dual eigenvectors $\boldsymbol{\psi}$

$$\mathcal{B}(\mathbf{v}, \boldsymbol{\psi}) = 1. \tag{28}$$

To derive the DWR method for the eigenvalue problem in Eq. (24), the functional $\mathcal{V}(\cdot, \cdot)$ is defined, such that the following problem should be solved

$$\begin{aligned} &\text{Find } \hat{\mathbf{v}} = (\mu, \mathbf{v}) \in \mathbb{R} \times \mathcal{S} \text{ s.t.} \\ &\quad \mathcal{V}(\hat{\mathbf{v}}, \hat{\boldsymbol{\phi}}) = \mu\mathcal{B}(\mathbf{v}, \boldsymbol{\phi}) - \mathcal{A}(\mathbf{v}, \boldsymbol{\phi}) + \tau(\mathcal{B}(\mathbf{v}, \mathbf{v}) - 1) = 0, \\ &\quad \forall \hat{\boldsymbol{\phi}} = (\tau, \boldsymbol{\phi}) \in \mathbb{R} \times \mathcal{S}, \end{aligned} \tag{29}$$

where the normalisation condition from Eq. (25) is enforced weakly. The discrete counterpart of this equation reads:

$$\begin{aligned} &\text{Find } \hat{\mathbf{v}}_h = (\mu_h, \mathbf{v}_h) \in \mathbb{R} \times \mathcal{S}_h^p \text{ s.t.} \\ &\quad \mathcal{V}(\hat{\mathbf{v}}_h, \hat{\boldsymbol{\phi}}_h) = \mu\mathcal{B}(\mathbf{v}_h, \boldsymbol{\phi}_h) - \mathcal{A}(\mathbf{v}_h, \boldsymbol{\phi}_h) + \tau_h\big(\mathcal{B}(\mathbf{v}_h, \mathbf{v}_h) - 1\big) = 0, \\ &\quad \forall \hat{\boldsymbol{\phi}}_h = (\tau_h, \boldsymbol{\phi}_h) \in \mathbb{R} \times \mathcal{S}_h^p. \end{aligned} \tag{30}$$

Furthermore, a goal-function for the eigenvalues is defined as follows:

$$\mathcal{L}(\hat{\mathbf{v}}) = \mu = \mu\mathcal{B}(\mathbf{v},\mathbf{v}), \tag{31}$$

giving

$$\Delta\mathcal{L}(\hat{\boldsymbol{\phi}}_h) = \mu - \mu_h. \tag{32}$$

Using the non-linear functional $\mathcal{V}$ and the goal functional $\mathcal{L}$, the same derivations as in Sect. 3.1 can be followed to find a system of equations to solve the DWR eigenvalue problem. The Gateaux derivative of $\mathcal{V}$, denoted by $\mathcal{V}'$ is given by:

$$\begin{aligned}\mathcal{V}'(\hat{\mathbf{v}},\hat{\boldsymbol{\phi}},\hat{\boldsymbol{\psi}}) = {}& \eta\mathcal{B}(\mathbf{v},\boldsymbol{\psi}) + \mu\mathcal{B}(\boldsymbol{\psi},\boldsymbol{\phi})\\ &- \mathcal{A}(\boldsymbol{\psi},\boldsymbol{\phi}) + \tau(\mathcal{B}(\mathbf{v},\boldsymbol{\psi}) + \mathcal{B}(\boldsymbol{\psi},\mathbf{v})),\end{aligned} \tag{33}$$

where the derivatives $\mathcal{A}'(\boldsymbol{\psi},\boldsymbol{\phi})$ and $\mathcal{B}'(\boldsymbol{\psi},\boldsymbol{\phi})$ are equal to the bi-linear operators $\mathcal{A}(\mathbf{u},\boldsymbol{\phi})$ and $\mathcal{B}(\mathbf{u},\boldsymbol{\phi})$ themselves. Furthermore, the solution around which the linerisation is performed is denoted by $\hat{\mathbf{v}} = (\mu,\mathbf{v})$, the test functions are denoted by $\hat{\boldsymbol{\phi}} = (\tau,\boldsymbol{\phi})$ and the trial functions are denoted by $\hat{\boldsymbol{\psi}} = (\eta,\boldsymbol{\psi})$. Furthermore, the linearisation of the goal functional Eq. (31) is

$$\mathcal{L}'(\hat{\mathbf{v}},\hat{\boldsymbol{\psi}}) = \eta\mathcal{B}(\mathbf{v},\mathbf{v}) + \mu[\mathcal{B}(\mathbf{v},\boldsymbol{\psi}) + \mathcal{B}(\boldsymbol{\psi},\mathbf{v})], \tag{34}$$

such that the adjoint eigenvalue problem, analoguously to Eq. (21), given by

$$\begin{aligned}&\text{Find } \hat{\boldsymbol{\phi}} = (\tau,\boldsymbol{\phi}) \in \mathbb{R}\times\mathcal{S} \text{ s.t. } \mathcal{V}'(\hat{\mathbf{v}},\hat{\boldsymbol{\phi}},\hat{\boldsymbol{\psi}}) = \mathcal{L}'(\hat{\mathbf{v}},\hat{\boldsymbol{\psi}})\\ &\quad\forall\hat{\boldsymbol{\psi}} = (\eta,\psi) \in \mathbb{R}\times\mathcal{S},\end{aligned} \tag{35}$$

becomes [56, 69]:

$$\begin{aligned}&\text{Find } \hat{\boldsymbol{\phi}} = (\tau,\boldsymbol{\phi}) \in \mathbb{R}\times\mathcal{S} \text{ s.t.}\\ &\quad \eta\mathcal{B}(\mathbf{v},\boldsymbol{\phi}) + \mu\mathcal{B}(\boldsymbol{\psi},\boldsymbol{\phi}) - \mathcal{A}(\boldsymbol{\psi},\boldsymbol{\phi}) + \tau(\mathcal{B}(\mathbf{v},\boldsymbol{\psi})\\ &\quad + \mathcal{B}(\boldsymbol{\psi},\mathbf{v})) = \eta\mathcal{B}(\mathbf{v},\mathbf{v})\\ &\quad + \mu[\mathcal{B}(\mathbf{v},\boldsymbol{\psi}) + \mathcal{B}(\boldsymbol{\psi},\mathbf{v})],\\ &\quad \forall\hat{\boldsymbol{\psi}} = (\eta,\psi) \in \mathbb{R}\times\mathcal{S}.\end{aligned} \tag{36}$$

This equation can be simplified to obtain the following [56, 69]:

$$\begin{aligned}&\text{Find } \hat{\boldsymbol{\phi}} = (\tau,\boldsymbol{\phi}) \in \mathbb{R}\times\mathcal{S} \text{ s.t.}\\ &\quad \mu\mathcal{B}(\boldsymbol{\psi},\boldsymbol{\phi}) - \mathcal{A}(\boldsymbol{\psi},\boldsymbol{\phi}) + \eta[\mathcal{B}(\mathbf{v},\boldsymbol{\phi}) - \mathcal{B}(\mathbf{v},\mathbf{v})]\\ &\quad + (\tau-\mu)[\mathcal{B}(\mathbf{v},\boldsymbol{\psi}) + \mathcal{B}(\boldsymbol{\psi},\mathbf{v})] = 0,\\ &\quad \forall\hat{\boldsymbol{\psi}} = (\eta,\psi) \in \mathbb{R}\times\mathcal{S}.\end{aligned} \tag{37}$$

Using the normalizations from Eqs. (25) and (28) and the fact that Eq. (27) solves the same equation as Eq. (24), it follows that Eq. (37) is solved by Eq. (27) [69].

Using Eqs. (20), (32) and (15) with $\mathcal{W} = \mathcal{V}$ according to Eq. (29) and with $\boldsymbol{\psi}$ denoting the dual eigenvector and η the dual eigenvalue, the error estimation according to the DWR method for an eigenvalue problem is

$$\begin{aligned}\Delta\mathcal{L}(\hat{\mathbf{v}}_h) = {}& \mathcal{A}(\mathbf{v}_h,\boldsymbol{\psi}-\boldsymbol{\psi}_h) - \mu_h\mathcal{B}(\mathbf{v}_h,\boldsymbol{\psi}-\boldsymbol{\psi}_h)\\ &+ (\eta-\eta_h)(\mathcal{B}(\mathbf{v}_h,\mathbf{v}_h) - 1),\end{aligned} \tag{38}$$

for $\hat{\mathbf{v}}_h = (\mu_h,\mathbf{v}_h) \in \mathbb{R}\times\mathcal{S}^p_h$, $\hat{\boldsymbol{\psi}}_h = (\eta_h,\boldsymbol{\psi}_h) \in \mathbb{R}\times\mathcal{S}^p_h$ and $\hat{\boldsymbol{\psi}} = (\eta,\boldsymbol{\psi}) \in \mathbb{R}\times\mathcal{S}$. The exact adjoint solution $\hat{\boldsymbol{\psi}}_h$ is again approximated by solving Eq. (27) on an *enriched* space $\tilde{\mathcal{S}}^p_h \subset \mathcal{S}, \tilde{\mathcal{S}}^p_h \supset \mathcal{S}^p_h$, providing $(\tilde{\eta}_h,\tilde{\boldsymbol{\psi}}_h) \in \mathbb{R}\times\tilde{\mathcal{S}}^p_h$. In [38] different choices for constructing $\tilde{\mathcal{S}}^p_h$ are given, including an h-refinement and a p-refinement. As in the work of [33], the second approach is used in the present paper, with the same mesh as for $\mathcal{S}^p_h$, but with a higher order and with the same regularity, i.e. $\tilde{\mathcal{S}}^p_h = \mathcal{S}^{p+1}_h$, as it introduces less degrees of freedom compared to an h-refinement.

As specificed in the end of Sect. 2.4, the DWR method for modal analysis requires $\mathcal{A}(\mathbf{v},\boldsymbol{\phi}) = \mathcal{W}'(\mathbf{0},\mathbf{v},\boldsymbol{\phi})$ and $\mathcal{B}(\mathbf{v},\boldsymbol{\phi}) = \mathcal{M}(\mathbf{v},\boldsymbol{\phi})$. For buckling analysis, $\mathcal{A}(\mathbf{v},\boldsymbol{\phi}) = \mathcal{W}'(\mathbf{u}_L,\mathbf{v},\boldsymbol{\phi})$ and $\mathcal{B}(\mathbf{v},\boldsymbol{\phi}) = \mathcal{W}'(\mathbf{0},\mathbf{v},\boldsymbol{\phi})$ with the first operator defined about a pre-buckling solution $\mathbf{u}_L$.

3.3 Goal functionals for Isogeometric Kirchhoff–Love shells

The remainder of this section focusses on defining the goal functional $\mathcal{L}(\boldsymbol{u})$, see Eq. (19), together with its variation $\mathcal{L}'(\boldsymbol{u})$ such that the dual problem (Eq. (21)) can be solved and the error estimate (Eq. (20)) can be computed.

In general, the goal functional can be defined in a point, on a boundary or over the domain:

$$\mathcal{L}(\cdot) = \int_{\Omega} \updownarrow(\cdot,\mathbf{x})\,\mathrm{d}\Omega, \quad \text{Domain-wise}, \tag{39}$$

$$\mathcal{L}(\cdot) = \int_{\partial\Omega} \updownarrow(\cdot,\mathbf{x})\,\mathrm{d}\Gamma, \quad \text{Boundary-wise}, \tag{40}$$

$$\mathcal{L}(\cdot) = \sum_{i\in\mathcal{I}} \updownarrow(\cdot,\mathbf{x}_i), \quad \text{Point-wise}, \tag{41}$$

where Ω denotes the integration domain, $\partial\Omega$ a side of Ω and $\mathcal{I}$ a set of indices corresponding to points $\mathbf{x}_i \in \Omega$. Furthermore, $\updownarrow(\cdot,\mathbf{x}_i)$ denotes a goal functional summant or integrant, which has a variation denoted by $\updownarrow'(\cdot,\boldsymbol{\phi}\mathbf{x}_i)$. The variation of $\mathcal{L}$, denoted by $\mathcal{L}'(\cdot,\boldsymbol{\phi},\mathbf{x}_i)$, directly follows from $\updownarrow'(\cdot,\boldsymbol{\phi}\mathbf{x}_i)$ due to linearity of integrals and summation. In addition, we classify two different types of goal functional integrants, resulting in norm-based and component-based goal functionals. In the former case, $\updownarrow$ is of the form $\updownarrow = \|\mathbf{A}\|^2$ with variation $\updownarrow' = 2\mathbf{A}\cdot\mathbf{A}'$. For component-based goal functionals,

Table 1 Overview of the goal functionals. Here $\mathbf{u}_h = \mathbf{u}(\mathbf{x})$ is the discrete deformation tensor depending on position coordinate $\mathbf{x}$, $\mathbf{C}_h = \mathbf{C}(\mathbf{u}_h)$ is the deformation tensor based on $\mathbf{u}_h$ and $\mathbf{C}(\boldsymbol{\phi}'_h) = \mathbf{C}'(\mathbf{u}_h, \boldsymbol{\phi})$ is its variation, $\mathcal{T}(\mathbf{A})$ is the transformation of a second-order tensor $\mathbf{A}$ from the undeformed contravariant basis to the basis spanned by the principal directions. Note that the variation $[\mathcal{T}(\mathbf{A})]'$ of $\mathcal{T}(\mathbf{A})$ is $\mathcal{T}(\mathbf{A}')$, since the spectral decomposition of the deformation tensor itself is just a linear change of tensor basis

Displacement-norm	$\updownarrow_{\lVert\mathbf{u}\rVert}(\mathbf{u}_h) = \lVert\mathbf{u}_h(\mathbf{x})\rVert^2$	$\updownarrow'_{\lVert\mathbf{u}\rVert}(\mathbf{u}_h, \boldsymbol{\phi}_h) = 2\mathbf{u}_h \cdot \boldsymbol{\phi}_h$
Displacement-component	$\updownarrow_{\mathbf{u}_i}(\mathbf{u}_h) = \mathbf{u}_h(\mathbf{x}) \cdot \mathbf{e}_i$	$\updownarrow'_{\lVert\mathbf{u}\rVert}(\mathbf{u}_h, \boldsymbol{\phi}_h) = \mathbf{e}_i \cdot \boldsymbol{\phi}_h$
Stretch-norm	$\updownarrow_{\lVert\lambda\rVert}(\mathbf{u}_h) = \lVert\mathcal{T}(\mathbf{C}_h)\rVert^2$	$\updownarrow'_{\lVert\lambda\rVert}(\mathbf{u}_h, \boldsymbol{\phi}_h) = 2\mathcal{T}(\mathbf{C}_h) \cdot \boldsymbol{\phi}_h$
Stretch-component	$\updownarrow_{\lambda_i}(\mathbf{u}_h) = \mathcal{T}(\mathbf{C}_h) \cdot \mathbf{e}_i$	$\updownarrow'_{\lVert\lambda\rVert}(\mathbf{u}_h, \boldsymbol{\phi}_h) = \mathbf{e}_i \cdot \mathcal{T}(\mathbf{C}'_h)$
Tensor-norm	$\updownarrow_{\lVert\mathbf{A}\rVert}(\mathbf{u}_h) = \lVert\mathbf{A}\rVert^2$	$\updownarrow'_{\lVert\mathbf{A}\rVert}(\mathbf{u}_h, \boldsymbol{\phi}_h) = 2\mathbf{A} \cdot \boldsymbol{\phi}_h$
Tensor-component	$\updownarrow_{\mathbf{A}_i}(\mathbf{u}_h) = \mathbf{A} \cdot \mathbf{e}_i$	$\updownarrow'_{\lVert\mathbf{A}\rVert}(\mathbf{u}_h, \boldsymbol{\phi}_h) = \mathbf{e}_i \cdot \mathbf{A}$

we define $\updownarrow = \mathbf{A} \cdot \mathbf{e}_i$ with variation $\updownarrow' = A' \cdot \mathbf{e}_i$. Here, $\mathbf{e}_i$ is a unit vector in direction i. It should be noted that the goal functional $\mathcal{L}(\cdot)$ needs to be bounded in all cases, therefore making point-wise goal functionals not always suitable, e.g. in case of a stress singularity in a point.

In Table 1 we provide some goal functional integrants or summants $\updownarrow(\cdot, \mathbf{x}_i)$. Together with their variations $\updownarrow'(\cdot, \boldsymbol{\phi}\mathbf{x}_i)$, these provide $\mathcal{L}'(\cdot, \boldsymbol{\phi}, \mathbf{x}_i)$ due to linearity of integrals and summation. The *tensor-based* goal functionals refer to goal functionals that could be used for any second-order tensor, e.g. the membrane strain tensor $\boldsymbol{\varepsilon}(\mathbf{u})$ or the flexural moment tensor $\mathbf{m}(\mathbf{u})$.

4 Coarsening and refinement using THB splines

This section elaborates on coarsening and refinement of isogeometric meshes using THB-splines. In particular, this section elaborates on the *Mark*, *Refine* and *Transfer* blocks of Fig. 1. Firstly, Sect. 4.1 will provide a brief background on THB-splines, which enable the *Refine* step of the adaptive meshing flowchart. Then, Sect. 4.2 elaborates on methods for suitable grading for refinement meshes; which is required to provide admissible refinement with (Truncated) Hierarchical B-spline ((T)HB) bases, given labelled elements. Section 4.3 elaborates on the labeling method for the *Mark* step, given an element-wise error distribution, taking admissibility into account. Lastly, Sect. 4.4 elaborates on the quasi-interpolation method that is used to *Transfer* the solution of one solution step to the next. The notations in this section will be closely related to those used in [25, 50].

4.1 (T)HB-Splines

Refinement of B-spline meshes can be done using (Truncated) Hierarchical B-splines ((T)HB-splines), of which the details can be found in [46, 47]. The conceptual idea behind (T)HB-splines is that they are constructed from a sequence of N nested tensor B-spline spaces in different

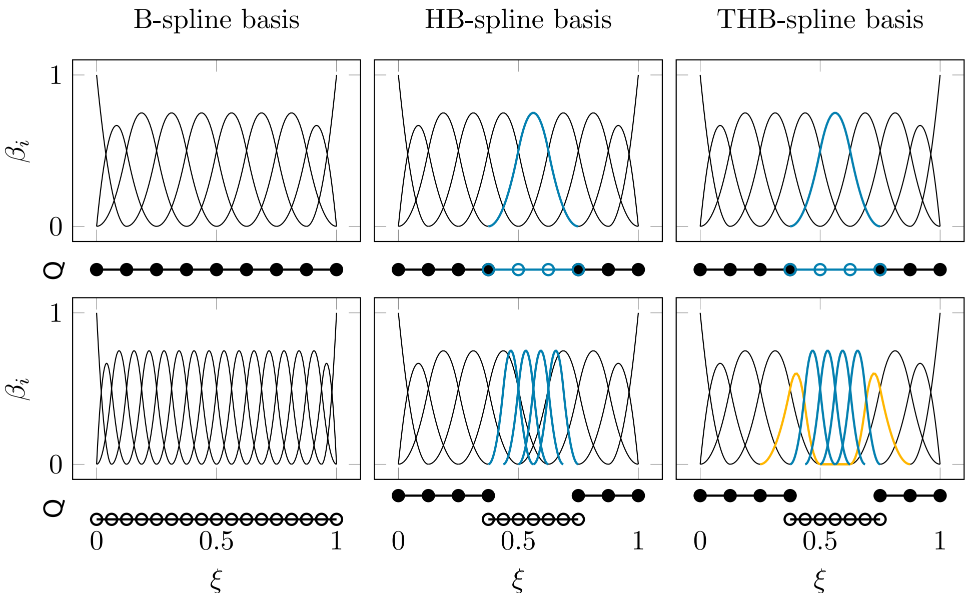


Fig. 2 Principles of refinement for different spline bases. The top figures represent the basis on level $\mathcal{B}^0$, optionally with refined basis functions coloured blue. Bottom pictures represent refined bases. (left) uniform refinement (hence $\mathcal{B}^1$; middle) HB-refinement; (right) THB-refinement, with truncated basis functions coloured yellow. Note that the refinement basis functions are from V^1. The unrefined unique knot vector in all cases is $\Xi = \{0, 1/8, 2/8, \dots, 7/8, 1\}$ and the degree of the basis is 2. The bases are generated in `G+Smo` [72]

levels $l = 0, \ldots, N-1$, denoted by $V^0 \subset V^1 \subset, \ldots, V^{N-1}$ with associated bases $\mathcal{B}^\ell$ of degree p on a grid G^ℓ with elements Q. The parametric domains are defined as $\Omega = \Omega^0 \supseteq \Omega^1 \supseteq \ldots \supseteq \Omega^{N-1} = \emptyset$. By defining the set of active cells by $\mathcal{G}^\ell := \{Q \in G^\ell \,:\, Q \subset \Omega^\ell \wedge Q \not\subset \Omega^{\ell+1}\}$, the hierarchical mesh is defined as $\mathcal{Q} = \{Q \in \mathcal{G}^\ell \,:\, \ell = 0, \ldots, N-1\}$. In Fig. 2, an illustration is given for a refined B-spline basis (left), a refined HB-spline basis (middle) and a refined THB-spline basis (right). For the (T)HB-spline basis, this picture depicts the refinement of a single basis function, corresponding to the elements in its support. The (T)HB-spline bases show that for THB-splines a truncation is performed to ensure partition of unity, which is discussed in more detail in [47].

4.2 Admissible meshing

The concept of admissible meshing was discussed in [22, 23, 50]. An admissible mesh of class m is a mesh of which the truncated basis functions belong to at most m successive levels and mesh admissibility ensures that the number of basis functions acting on a mesh elements does not depend on the number of levels in the hierarchy, but on the parameter m. In order to guarantee mesh admissibility for refinement and coarsening operations, refinement and coarsening neighborhoods are defined such that admissible meshes can be constructed recursively, which is discussed in more detail in [22, 23, 50]. In Fig. 3, we illustrate a simple mesh together with the refinement neighborhood of some selected elements. The $\mathcal{T}$-refinement neighborhood $\mathcal{N}_r(\mathcal{Q}, Q, m)$ of element Q is defined as

$$\begin{aligned}&\mathcal{N}_r^{\mathrm{THB}}(\mathcal{Q}, Q, m)\\ &= \left\{Q' \in \mathcal{G}^{\ell-m+1} : \exists Q'' \in S(Q, \ell-m+2), Q'' \subseteq Q'\right\},\end{aligned} \tag{42}$$

where $S(Q, k)$ is the *multi-level* support extension with respect to level k.

The coarsening neighborhood $\mathcal{N}_c(Q)$ of element $Q \in G^\ell$ is defined by [25]. When coarsening element Q^ℓ, the coarsening neighborhood ensures that the newly activated basis functions are not active on the surrounding basis functions of level $\ell + m$. In other words, if element Q of level ℓ is the element to be coarsened, then the coarsening neighborhood defined by

$$\begin{aligned}&\mathcal{N}_c(\mathcal{Q}, Q, m)\\ &= \left\{Q' \in \mathcal{G}^{\ell+m-1} : \exists Q'' \in \mathcal{G}^\ell \text{ and } Q'' \subset P(Q),\right.\\ &\quad \left.\text{with } Q' \in S(Q'', \ell)\right\},\end{aligned} \tag{43}$$

must be empty. Here, $P(Q)$ denotes the parent of Q, i.e. the unique cell $Q' \in G^{\ell-1}$ such that $Q \subset Q'$. Note the small difference with respect to the definintion given in [25], since the coarsening neighborhood in their work is defined for the element $\hat{Q}$ of level ℓ which will be activated, i.e. $\hat{Q}$ is the parent of Q for which the coarsening neighborhood is

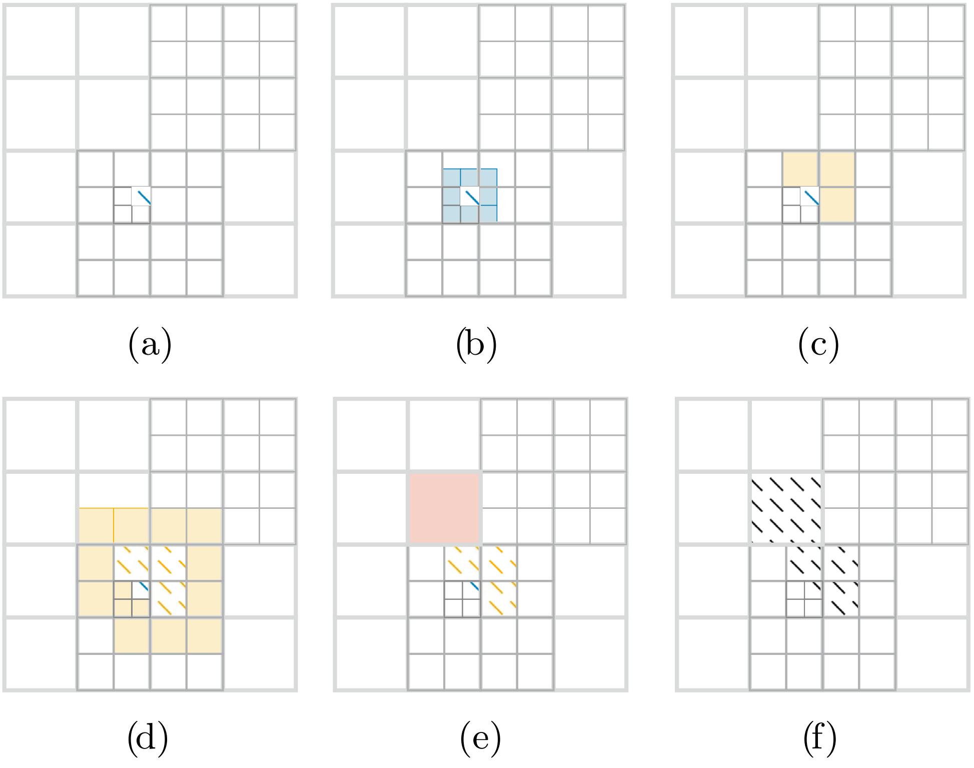


Fig. 3 Recursive marking strategy on the marked element of level ℓ on the initial mesh represented in (**a**). As a first step, the support extension of the marked element is obtained (**b**), from which the parents that are active on level $\ell - 1$ define the $\mathcal{T}$-neighborhood of the marked cell (**c**). Starting the same procedure on the marked cells of level $\ell - 1$, the support extension can again be obtained (**d**) with their corresponding parents on level $\ell - 2$, marking the $\mathcal{T}$-neighborhood of the marked elements of level $\ell - 1$ (**e**). The complete recursive marking from the marked element in **a** is depicted in (f)

defined here. Given the definition in Eq. (43) and given a set of elements marked for refinement $\mathcal{M}_r$, a coarsening neighborhood checking elements marked for refinement, can be defined:

$$\mathcal{N}_c^r(\mathcal{Q}, Q, m, \mathcal{M}_r): = \left\{Q' \in \mathcal{G}^{\ell+m-2} : Q' \in \mathcal{M}_r, \exists Q'' \in \mathcal{G}^{\ell-1} \text{ and } Q'' \subset P(Q), \text{ with } Q' \in S(Q'', \ell - 1)\right\}. \tag{44}$$

In other words, this is the coarsening neighborhood that checks whether for element Q of level ℓ to be coarsened there are elements in the marked set $\mathcal{M}_r$ that will be part of the coarsening neighboorhood as soon as they are refined; thus it uses Eq. (43) with $\ell - 1$. This neighborhood ensures that coarse labeling can be performed conforming with the Dörfler marking strategy and without refining first. This avoids to compute element-error contributions on an in-between mesh which has been refined first. Obviously, if another element with the same parent as Q is marked for refinement, no coarsening should take place. An element can be coarsened if $\mathcal{N}_c^r(\mathcal{Q}, Q, m, \mathcal{M}_r) = \emptyset$. Combining both neighborhoods, an element Q of level ℓ can be coarsened if and only if $\hat{\mathcal{N}}_c(\mathcal{Q}, Q, m, \mathcal{M}_r) = \mathcal{N}_c(\mathcal{Q}, Q, m) \cup \mathcal{N}_c^r(\mathcal{Q}, Q, m, \mathcal{M}_r) = \emptyset$. In Fig. 4, the coarsening neighborhood is illustrated for a simple mesh.

4.3 Labeling methods

Let $\tilde{\mathcal{Q}}$ be the ordered set of $\mathcal{Q}$ such that $e_k \geq e_{k+1}\ \forall k \in \tilde{\mathcal{Q}}$, where e_k denotes the error of element k. Then, the Dörfler marking strategy [44] is defined as the elements $Q_i \in \tilde{\mathcal{Q}}$, $i = 0, \dots, k$ such that the sum of their respective errors is smaller than a fraction ρ_r of the total element error $e = \sum_i e_i$:

$$\mathcal{M}_r^{\text{Dörfler}} = \left\{ Q_i \in \mathcal{Q} : \sum_{k=0}^{i} e_k < \rho_r e \right\}. \tag{45}$$

This marking strategy, however, does not take into account the contributions of the elements that are marked because they are part of a refinement neighborhood of a marked element $Q_i \in \mathcal{M}_r$. Therefore, we define the index set $\mathcal{I}_r^K$ as the set of element indices whose span contains elements $Q_k \in \tilde{\mathcal{Q}}$ and their refinement neighborhoods:

$$\mathcal{I}_r^K := \left\{k \in 1, \dots, K : \mathcal{N}_r(Q_k, \mathcal{Q}, m) \cup Q_k,\ Q_k \in \tilde{\mathcal{Q}}\right\}, \tag{46}$$

and we define κ_r as the maximum index for which the sum of all elements with indices i in $\mathcal{I}_r^{\kappa_r}$ is smaller than the error tolerance $\rho_r e$:

$$\kappa_r := \arg\max \sum_{i \in \mathcal{I}_r^K} e_i < \rho_r e, \tag{47}$$

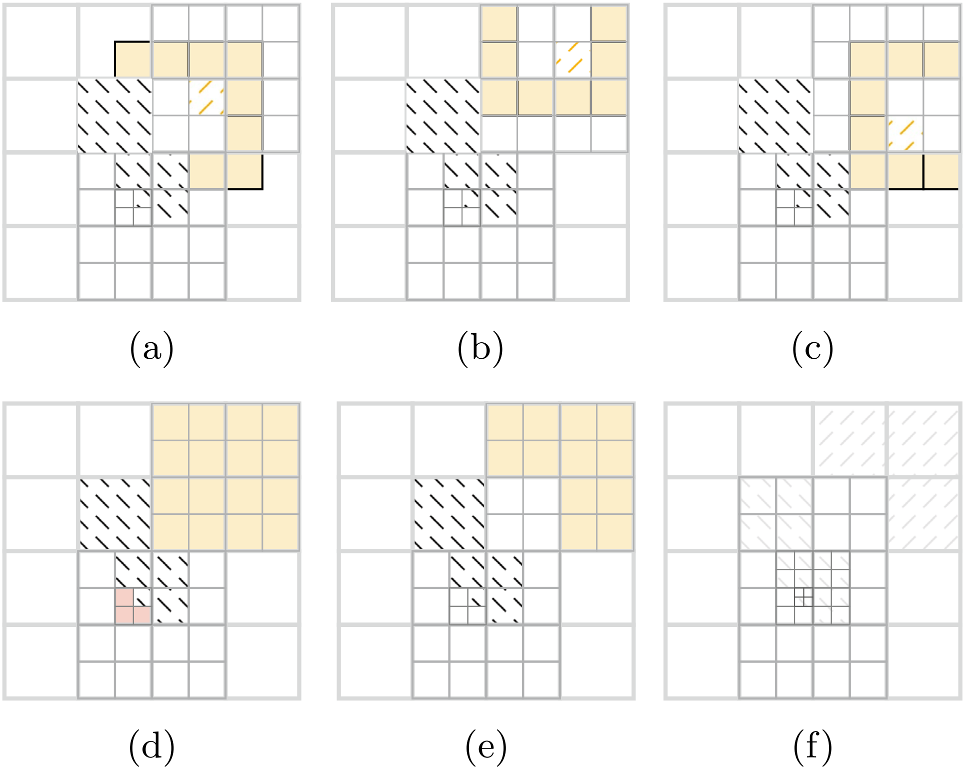


Fig. 4 Given the mesh from Fig. 3f, the coarsening neighborhoods are evaluated in **a**–**c** of this figure. The cell ▨ marks the cell ℓ of level ℓ that is marked for coarsening to its parent and the cells ▧ mark cells that are marked for refinement. The ring around the cell marked for coarsening depicts the region that should be checked for the coarsening neighborhood. That is, it defines the region that should not contain cells of level $\ell + 1$ (for $\mathcal{N}_c$) or cells of level ℓ that are marked for refinement (for $\mathcal{N}_c^r$). The cells for which $\mathcal{N}_c = \emptyset$ are marked in **d** and the cells with $\mathcal{N}_c^r = \emptyset$ are marked in **e**. The final mesh after refinement and coarsening is depicted in **f**. The coarsened elements that satisfy $\mathcal{N}_c \cup \mathcal{N}_c^r = \emptyset$ are marked as coarsened elements

such that the Dörfler marking including refinement neighborhoods is

$$\mathcal{M}_r = \left\{ Q_k \in \mathcal{Q} \,:\, k \in \mathcal{I}_r^{\kappa_r} \right\}. \tag{48}$$

For marking a set of coarsening elements, $\mathcal{M}_c$ the Dörfler marking procedure can be followed again. The original Dörfler marking strategy would be coarsening the elements $Q_i \in \mathcal{Q}$ such that their total element error is smaller than a fraction of the total element error $\rho_c e$, with coarsening parameter ρ_c:

$$\mathcal{M}_c^{\text{Dörfler}} = \left\{ Q_i \in \mathcal{Q} \,:\, \sum_{k=i}^{N} e_k < \rho_c e \right\}. \tag{49}$$

Similar to marking for refinement, the marking rule for coarsening can be specified more precisely by including admissible coarsening. In this case, the elements for which $\hat{\mathcal{N}}_c(\mathcal{Q}, Q, m, \mathcal{M}_r) = \emptyset$ holds are added in the sum of marked elements. Therefore, let us define the index set $\mathcal{J}_K$ that contains all elements $Q_k \in \mathcal{Q}$ for which the admissible coarsening condition holds, starting from the element with the smallest error, i.e. Q_N.

$$\mathcal{I}_c^K := \left\{ k \in 1, ..., K \,:\, \hat{\mathcal{N}}_c(\mathcal{Q}, Q, m, \mathcal{M}_r) = \emptyset,\; Q_{N-k-1} \in \tilde{\mathcal{Q}} \right\}. \tag{50}$$

Similar to κ_r, we define κ_c as the maximum index for which the sum of all elements with indices i in $\mathcal{I}_c^{\kappa_c}$ is smaller than the error tolerance $\rho_c e$:

$$\kappa_c := \arg\max \sum_{i \in \mathcal{I}_r^K} e_i < \rho_c e, \tag{51}$$

such that the Dörfler marking strategy taking into account coarsening admissibility is defined as

$$\mathcal{M}_c = \left\{ Q_k \in \mathcal{Q} \,:\, k \in \mathcal{I}_c^{\kappa_c} \right\}. \tag{52}$$

An alternative to the Dörfler marking strategy is a strategy where a given fraction of the total number of elements is marked. In that case, the formulations from Eqs. (48) and (52) would still hold, but in Eqs. (46) (50) the indices κ_r and κ_c are defined by the sum of the marked elements in respectively $\mathcal{I}_r^K$ and $\mathcal{I}_c^K$.

Whether to mark a set for refinement or coarsening, i.e. to construct $\mathcal{M}_r$ and $\mathcal{M}_c$ depends on the global error $\Delta\mathcal{L}$ following from the DWR and user-defined tolerances for refinement and coarsening. Let tol_r be the tolerance for refinement and tol_c the tolerance for coarsening, such that $\mathcal{M}_r \neq \emptyset$ if and only if $\Delta\mathcal{L} > \text{tol}_r$ and $\mathcal{M}_c \neq \emptyset$ if and only if $\Delta\mathcal{L} < \text{tol}_r$. As a consequence, if $\text{tol}_r \geq \text{tol}_c$, refinement and coarsening are never performed simultaneously. If $\text{tol}_r < \text{tol}_c$ a band with bandwidth $\text{tol}_c - \text{tol}_r$ is defined, in which refinement and coarsening are performed simultaneously. In the present work, tolerances are defined such that the latter condition is satisfied, and the adaptivity iterations are terminated when $\Delta\mathcal{L} \in [\text{tol}_r, \text{tol}_c]$, i.e.:

$$\begin{cases} \mathcal{M}_r = \emptyset,\; \mathcal{M}_c \neq \emptyset & \text{if}\,\Delta\mathcal{L} < \text{tol}_r, \\ \mathcal{M}_r \neq \emptyset,\; \mathcal{M}_c = \emptyset & \text{if}\,\Delta\mathcal{L} > \text{tol}_c, \\ \mathcal{M}_r \neq \emptyset,\; \mathcal{M}_c \neq \emptyset & \text{if}\,\text{tol}_r \geq \Delta\mathcal{L} \geq \text{tol}_c, \end{cases} \tag{53}$$

given $\text{tol}_r \leq \text{tol}_c$.

Note that the total element error e and the total estimated error of the system of equations $\Delta\mathcal{L}$ are not necessarily the same, since the element error measure e_k can be defined in different ways. In case of the DWR method, a natural choice is to choose e_k as the element-wise integrals of $\Delta\mathcal{L}$ from Eq. (20). However, integrating the squared norm of the integrant from Eq. (20) would yield strictly positive element errors, making the ordering of the set of element errors simple.

4.4 Quasi-interpolation

In the discrete setting, solution of the problem $\mathbf{u}_h$ is represented by the THB-spline basis $\boldsymbol{\phi}_i \in \mathcal{S}_h^p$ together with the solution coefficients $\alpha_i \in \mathbb{R}$. In case of analyses with multiple solution steps (e.g. dynamic or quasi-static analysis), mesh refinements can be performed after each solution step. As a consequence, the solution at load step $k+1$ is defined on another set of basis functions $\{\bar{\boldsymbol{\phi}}_i\} \in \mathcal{S}_h^p$ with corresponding coefficients $\bar{\alpha}_i^k$ compared to the previous solution at step k. In order to transfer the coefficients α_i^k to the new basis, an interpolation scheme needs to be used.

Interpolation on a spline basis can be a costly part of the simulation. Global interpolation implies that the contributions of all basis functions are taken into account in the interpolation. This requires solving a large dense system. An efficient way of interpolating spline coefficients for hierarchical basis is a so-called quasi-interpolation scheme [54, 55]. Here, on each level of the hierarchical basis a quasi-interpolant is constructed. This quasi-interpolant interpolates a given function f over the support of each basis function individually, to find the coefficient related to that basis function. More precisely, given a function $f \in C(\Omega^0)$, the quasi-interpolant for level ℓ is defined as

$$\Lambda^\ell(f) = \sum_{i=1}^{N} \lambda_{i,\ell}(f) B_{i,\ell}, \quad \ell = 0, \ldots, n-1, \tag{54}$$

where the coefficients $\lambda_{i,\ell}$ are suitable linear functionals on $C(\Omega^0)$. Across all levels $\ell = 0, \dots, N-1$, the interpolant for the function becomes:

$$\Lambda(f) = \sum_{\ell=0}^{N-1} \sum_{i\in\mathcal{I}_{\ell,\Omega_n}} \lambda_{i,\ell}(f) B_{i,\ell}, \qquad (55)$$

where $B^{\mathcal{T}}_{i,\ell,\Omega_n}$ is a THB spline of level ℓ constructed on domain Ω_n. For any basis function $B_{i,\ell}$, the coefficient $\lambda_{i,\ell}$ are found by locally interpolating the function f onto all active basis functions $B_{j,\ell}, j \in \mathcal{J}$ in the support of $B_{i,\ell}$. This gives coefficients $\lambda_{j,\ell}, j \in \mathcal{J}$ of which coefficient i gives $\lambda_{i,\ell}$. This quasi-interpolation scheme is used in the present framework to express the solution obtained from the previous load-step in terms of the newly, adaptively refined and coarsened basis. In the case of non-nested spaces – which can occur when coarsening – this implies that the quasi-interpolation scheme is not exact.

5 Algorithmic overview

In Fig. 1 the adaptive isogeometric method for solution stepping problems has been presented. Based on Sects. 2–4, a summarised workflow for adaptive isogeometric shell analysis is depicted in Fig. 5 and Algorithm 1.

The *Solve* block involves solving the non-linear isogeometric Kirchhoff–Love shell equation from Eq. (16). This variational formulation involves geometric and material non-linearities and can potentially also involve load non-linearities. After solving the Kirchhoff–Love shell problem, the discrete solution vector $\mathbf{u}_h$ is passed to the *Estimate* block. Here, the DWR method is solved by computing the adjoint problem in the primal space (Eq. (21)) and in the enriched space (Eq. (22)). Then, the element-wise error estimate can be obtained by integrating Eq. (20) element-wise. The element-wise errors e_k can be passed to the *Mark* block, where elements are marked for refinement (Eq. (52)) if the total error $\Delta\mathcal{L}$ is larger than a lower (refinement) tolerance tol_r and a coarsening marking (Eq. (52)) is performed if the total error is above an upper (coarsening) tolerance tol_c. This implies that if $\text{tol}_c < \Delta\mathcal{L} < \text{tol}_r$, a combined coarsening and refinement step is performed, as described in Eq. (53). In this case, the coarsening marking from Eq. (52) is performed given $\mathcal{M}_r$. Given the elements marked for refinement and coarsening, collected in $\mathcal{M}_r$ and $\mathcal{M}_c$ respectively, the mesh can be *Adapt*ed. In order to start the solution interval again, the start point should be *Transfer*red to the new mesh and the governing equation can be solved again if the error is not in the interval $[\text{tol}_r, \text{tol}_c]$ or if the number of refinement iterations i exceeds the maximum number of refinement iterations, $I_{\max}$. If the total error is in the interval $[\text{tol}_r, \text{tol}_c]$ or if the number of refinement iterations i exceeds the maximum number of refinement iterations, $I_{\max}$, the solution can be advanced, e.g. using a load-stepping or an arc-length method. Thereafter, the governing equations can be *Solve*d again. Note that if $I_{\max} = 1$, no inner iterations for adaptive meshing are performed.

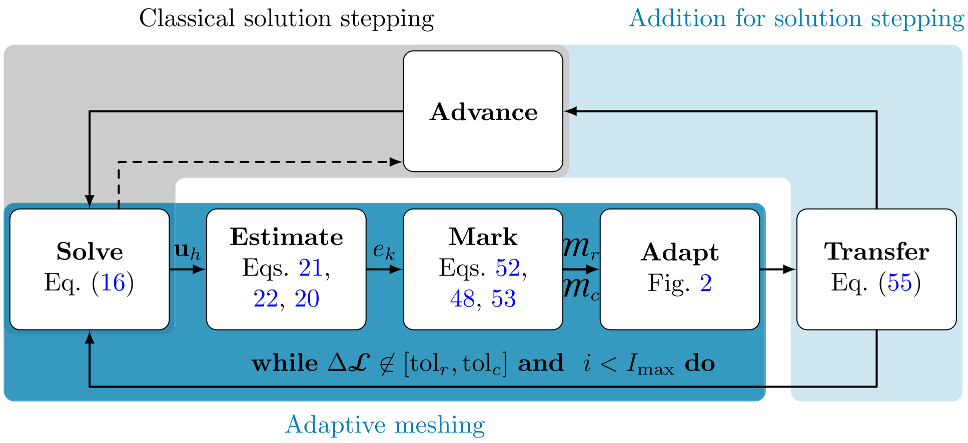


Fig. 5 A graphical summary of the adaptive meshing flowchart from Fig. 1 used in the present work. The equations which are used in each step are indicated in the blocks. The adaptive meshing iterations are performed within each solution step until the total error $\Delta\mathcal{L}$ is contained in the interval $[\text{tol}_r, \text{tol}_c]$, $\text{tol}_r < \text{tol}_c$, following the tolerances in Sect. 4.3. In case of convergence, the solution is advanced, e.g. with an arc-length iteration. Algorithm 1 provides an algorithm corresponding to this flow-chart

Algorithm 1 An algorithmic summary of the goal-adaptive meshing routine employed in the present work. Figure 5 provides a graphical summary of this algorithm.

```
           for loadsteps do
               while $\Delta\mathcal{L} \notin [\text{tol}_l, \text{tol}_u]$ and $i < I_{\max}$ do
Solve     |        Compute the primal solution $\mathbf{u}_h$ (Eq. (16))
          |        Compute the dual solution, $\boldsymbol{\xi}_h$ given $\mathbf{u}_h$ (Eq. (21))
Estimate  |        Compute the enriched dual solution, $\tilde{\boldsymbol{\xi}}_h$ given $\mathbf{u}_h$ (Eq. (22))
          |        Compute the total error estimation $\Delta\mathcal{L}$ according to Eq. (20) and find the element-wise errors $e_k$ and the total element error $e$.
          |        if $\Delta\mathcal{L} > \text{tol}_r$ then
          |            Mark elements for refinement into $\mathcal{M}_r$ using $e$, see Eq. (48).
Mark      |        end if
          |        if $\Delta\mathcal{L} < \text{tol}_c$ then
          |            Mark elements for coarsening into $\mathcal{M}_c$ using $e$, see Eq. (52).
          |        end if
Adapt     |        Refine all $Q \in \mathcal{M}_r$ using THB-splines, see Fig. 2
          |        Coarsen all $Q \in \mathcal{M}_c$ using THB-splines, see Fig. 2
Transfer  |        Transfer the solutions required to start the new solution step to the new mesh using Quasi-Interpolation, see Eq. (55).
               end while
Advance   |    Advance the solution to the next solution step
           end for
```

6 Numerical examples

In this section, several numerical examples are presented. The examples represent different applications of the theory presented in this paper and - without loss of generality - all employ Isogeometric Kirchhoff–Love shells. The first three examples illustrate the performance of the DWR error estimator and the last three example demonstrate the use of this error estimator for adaptive meshing. More precisely, the numerical examples performed in this section, as well as their purpose, are:

Linear static analysis of a square plate
(Sect. 6.1) A simple example of linear Kirchhoff–Love shell theory is presented. In this case, error estimators using the DWR are computed for different goal functionals and verified using the actual error computed from manufactured solutions. The goal of this benchmark problem is to evaluate the accuracy of the error estimators in linear static analysis.

Modal analysis of a circular plate
(Sect. 6.2) Since the analytical eigenvalues and eigenmodes are known for this case, the goal of this benchmark problem is to verify the error estimator for a vibration eigenvalue problem, given in Eq. (26) with the stiffness and mass operators $\mathcal{A}(\mathbf{v},\boldsymbol{\phi}) = \mathcal{W}'(\mathbf{0},\mathbf{v},\boldsymbol{\phi})$ and $\mathcal{B}(\mathbf{v},\boldsymbol{\phi}) = \mathcal{M}(\mathbf{v},\boldsymbol{\phi})$.

Linear buckling analysis of a square plate
(Sect. 6.3) Analytical critical buckling loads and mode shapes are also known for this case. Therefore, the goal of this benchmark problem is to provide verification for the buckling error estimator from Eq. (26) with the buckling operators $\mathcal{A}(\mathbf{v},\boldsymbol{\phi}) = \mathcal{W}'(\mathbf{u}_L,\mathbf{v},\boldsymbol{\phi})$ and $\mathcal{B}(\mathbf{v},\boldsymbol{\phi}) = \mathcal{M}(\mathbf{v},\boldsymbol{\phi})$.

Non-linear analysis of a pinched thin plate
(Sect. 6.4) In this example a thin plate with very low bending stiffness subject to a out-of-plane load is analysed. The error estimator is used to provide mesh adaptivity to compare to uniform refinement. The goal of this benchmark problem is to evaluate the performance of the DWR as driver for adaptive meshing in a static load case with geometric non-linearities.

Snap-through instability of a cylindrical roof (Sect. 6.5) The snap-through behaviour of a cylindrical roof are considered in this example. The benchmark problem is a well-known application of arc-length methods and shells. The goal of solving this problem is to test the full adaptive solution stepping procedure from Fig. 5 on a benchmark problem.

Wrinkling analysis (Sect. 6.6) In the last example, the procedure from Fig. 5 is applied to the modeling of membrane wrinkling. This problem contains geometric non-linearities and material non-linearities. The results are compared to uniformly refinements to evaluate the efficiency of adaptive meshing for such applications. The goal of this example is to demonstrate the use of the adaptive meshing routine from Fig. 5 on a complex load-stepping problem with geometric and material non-linearities.

In the following subsections, the short-hand notations $\mathcal{L}_{\text{an}} = \mathcal{L}(\boldsymbol{u}_{\text{an}})$, $\mathcal{L}_{\text{num}} = \mathcal{L}(\boldsymbol{u}_{\text{num}})$, $\Delta\mathcal{L}_{\text{an}} = \mathcal{L}_{\text{an}} - \mathcal{L}_{\text{num}}$ and $\Delta\mathcal{L}_{\text{num}} = \mathcal{R}(\boldsymbol{u}_{\text{num}}, \tilde{\xi}_h - \xi_h)$ (see Eq. (20)) are used, given the analytical and numerical solutions $\boldsymbol{u}_{\text{an}}$ and $\boldsymbol{u}_{\text{num}}$, respectively. Furthermore, where relevant, the parameters ρ_r, ρ_c, tol_r and tol_c (see Eqs. (53), (52) and (48)) are fixed per example. A study on finding optimal values for these parameters is out of the scope of this paper. Lastly, all simulations are performed using the open-source Geometry+Simulation modules [72].

6.1 Linear static analysis of a square plate

For the linear shell example, let us consider a flat plate with unit dimensions $L = W = 1$ [m], a thickness of $t = 10^{-2}$ [m] and with material parameters $E = 10^6$ [Pa], $\nu = 0.3$, which is clamped on all sides, see Fig. 6. A load vector of

$$\mathbf{f} = \frac{2EAt^3}{1-\nu^2}\left(x^4 - 2x^3 + 3x^2 - 2x + y^4 - 2y^3 + 3y^2 - 2y + 12x^2y^2 - 12x^2y - 12xy^2 + 12xy + \frac{1}{3}\right)\mathbf{e}_z \tag{56}$$

is applied, based on the manufactured solution given by

$$\boldsymbol{u}_{\text{an}} = Ax^2(x-1)^2y^2(y-1)^2\mathbf{e}_z. \tag{57}$$

Using this manufactured solution, any goal functional $\mathcal{L}_{\text{an}}$ can be evaluated. Solving the primal problem for this linear shell example gives $\boldsymbol{u}_{\text{num}}$, which can be used to compute the DWR error estimate of $\Delta\mathcal{L}_{\text{num}}$.

In Fig. 7, the results for the linear shell problem are given. The title of each column represents the goal-function that is used for error estimation in this column. The top row provides the errors $\Delta\mathcal{L}_{\text{an}}$ and $\Delta\mathcal{L}_{\text{num}}$ with respect to an uniformly refined mesh size. As can be seen in this figure, $\Delta\mathcal{L}_{\text{num}}$ quickly converges to $\Delta\mathcal{L}_{\text{an}}$ for different spline orders p. In addition, the bottom row of Fig. 7 provides the efficiency of the error estimator, given by $\Delta\mathcal{L}_{\text{num}}/\Delta\mathcal{L}_{\text{an}}$. These figures confirm convergence of the DWR estimates to the analytical goal functional errors for all considered goal functionals. Only for the membrane strain norm goal functional the error estimate for coarse meshes is inaccurate. This can possibly be explained by the in-plane shear strain that cancels out over the whole domain but which does contribute in the norm $\|\boldsymbol{\varepsilon}\|$

Concluding, the linear shell benchmarks shows that for different goal functionals the DWR method provides accurate estimation of the error $\Delta\mathcal{L}$ of the goal functional $\mathcal{L}$ starting at relatively small mesh sizes of $h < 10^{-1}$.

6.2 Modal analysis of a circular plate

As a next example, the vibration modes of a circular plate with clamped boundary conditions are computed. The geometry with boundary conditions is illustrated in Fig. 8. The circular plate has unit diameter, a thickness $t = 10^{-2}$ [m], Young's modulus $E = 10^6$ [Pa], Poisson's ratio $\nu = 0.3$

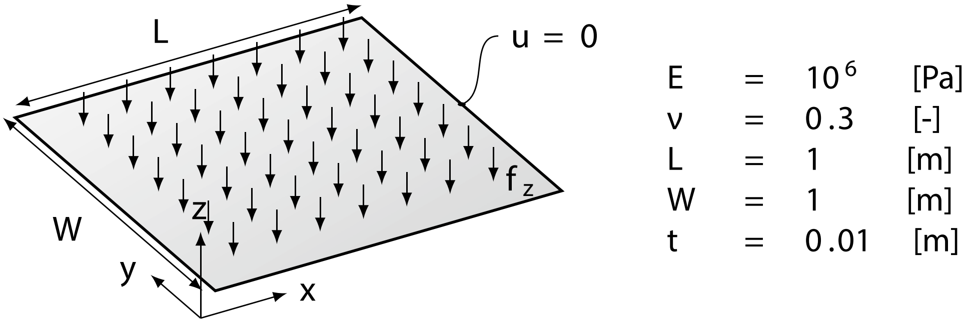


Fig. 6 Geometry and parameters for the example of a unit-square plate with a distributed vertical load f_z given by Eq. (56). The displacements are fixed on all edges

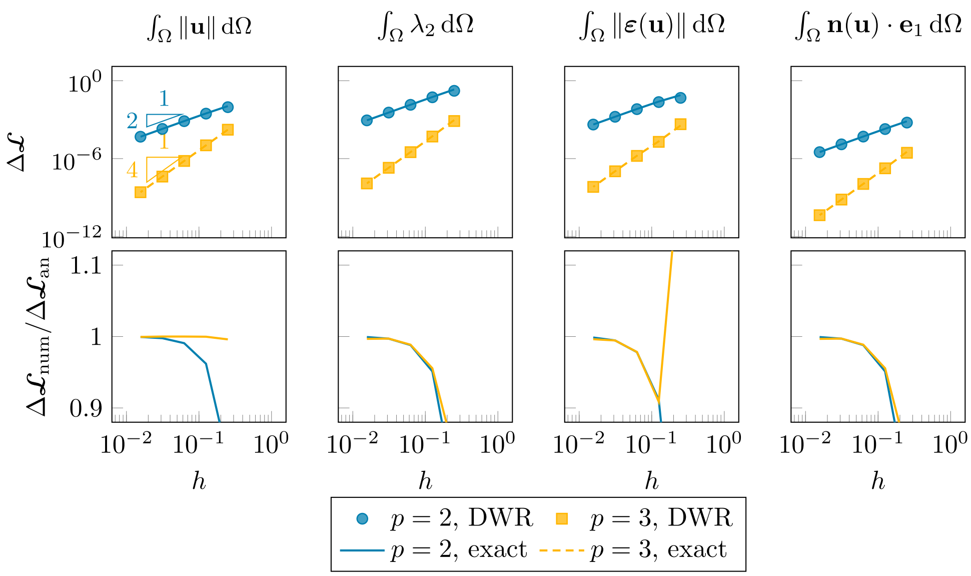
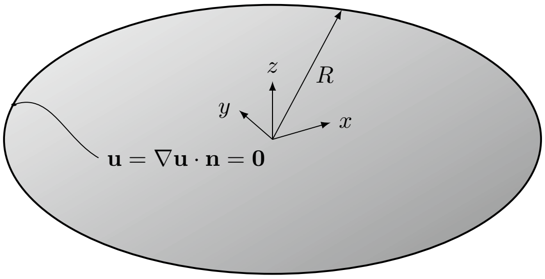


Fig. 7 Linear static analysis of a clamped plate with a uniformly distributed load according to Eq. (56) according to a manufactured solution from Eq. (57). The top row provides $\Delta\mathcal{L}$ against the uniform mesh size h. The bottom row presents the efficiency of the error estimators against the mesh size h. The markers represent error estimates computed via the DWR and the lines represent the exact error, i.e. the error of the numerical solution with respect to the analytical solution. The results are given for goal functionals (from left to right) $\mathcal{L}=\int_\Omega \|\mathbf{u}\|\mathrm{d}\Omega$ (displacement norm), $\mathcal{L}=\int_\Omega \lambda_2\mathrm{d}\Omega$ (second principal stretch), $\mathcal{L}=\int_\Omega \|\varepsilon(\mathbf{u})\|\mathrm{d}\Omega$ (membrane strain norm) and $\mathcal{L}=\int_\Omega \mathbf{n}(\mathbf{u})\cdot\mathbf{e}_1\mathrm{d}\Omega$ (first component of membrane force)

Fig. 8 Geometry and parameters for a vibrating circular plate with a clamped boundary



$$\begin{array}{lcll} E & = & 10^6 & [\mathrm{Pa}] \\ \nu & = & 0.3 & [\text{-}] \\ \rho & = & 1 & [\mathrm{kg/m^3}] \\ R & = & 0.5 & [\mathrm{m}] \\ t & = & 0.01 & [\mathrm{m}] \end{array}$$

and density $\rho = 1\ [\mathrm{kg/m^3}]$. The analytical solutions for the eigenfrequencies of the circular plate are obtained by

$$\omega_n = \gamma_n^2\sqrt{D/\rho t}, \tag{58}$$

where γ_n is the n^{th} root of the equation $(I_{m-1}(\gamma R) - m/RI_m(\gamma R))J_m(\gamma R) - I_m(\gamma R)(J_{m-1}(\gamma R)$ $-m/RJ_m(\gamma R)) = 0$ following from a separation of variables solution [67], R is the radius of the plate and $D = Et^3/(12(1-\nu^2)) = 9.16\cdot 10^{-8}$ is the flexural rigidity. As stated in Sect. 3.2, the goal functional eigenvalue problems is given in Eq. (31), and requires the eigenvalue problem in Eq. (24) to be solved with linear operators $\mathcal{A}(\mathbf{v},\boldsymbol{\phi}) = \mathcal{W}'(\mathbf{0},\mathbf{v},\boldsymbol{\phi})$ and $\mathcal{B}(\mathbf{v},\boldsymbol{\phi}) = \mathcal{M}(\mathbf{v},\boldsymbol{\phi})$, see Eqs. (14) and (10).

Figure 9 presents the first four eigenmodes in the top row. Furthermore, $\Delta\mathcal{L}_{\mathrm{an}}$ and $\Delta\mathcal{L}_{\mathrm{num}}$ as a function of the element size for uniformly refined domain are given in the middle row for the first four eigenmodes. These plots show that the approximation for the error $\Delta\mathcal{L}_{\mathrm{num}}$ approximates $\Delta\mathcal{L}_{\mathrm{an}}$. In the bottom row of Fig. 9 the efficiencies also show that the approximation converges to an efficiency equal to 1. However, for the $p = 4$ line, the efficiency degrades when the ‘exact’ error obtained by the analytical solution approaches values around 10^{-11}, which is attributed to the approximation of the roots γ_i and the precision of the eigenvalue solver.

Concluding, the modal analysis benchmark shows that the DWR method provides accurate estimation of the eigenfrequency error for different considered mode shapes.

6.3 Linear buckling analysis of a square plate

Similar to modal analysis, DWR error estimation for buckling analysis also relies on the formulations in Sect. 3.2. The difference with the modal analysis error estimation is that the buckling analysis error estimation involves the solution

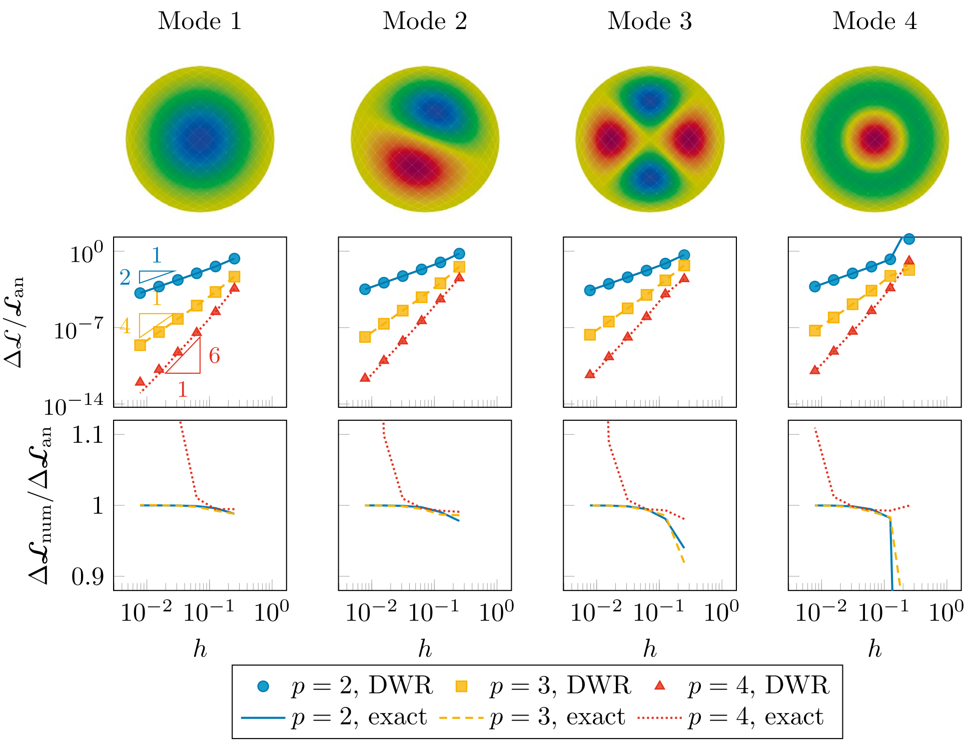


Fig. 9 Modal analysis of a circular plate. The top row provides the mode shapes from mode 1 up to 4. The mid row provides $\Delta\mathcal{L}$ against the uniform mesh size h and the bottom row provides the efficiency of the error estimators against the mesh size h. The markers represent error estimates computed via the DWR and the lines represent the exact error, i.e. the error of the numerical solution with respect to the analytical solution. The eigenfrequencies for mode 1 up to 4 are, respectively, $\omega_1 = 9.56 \cdot 10^{-4}$ [Hz], $\omega_2 = 4.14 \cdot 10^{-3}$ [Hz], $\omega_3 = 1.11 \cdot 10^{-2}$ [Hz] and $\omega_4 = 1.45 \cdot 10^{-2}$ [Hz]

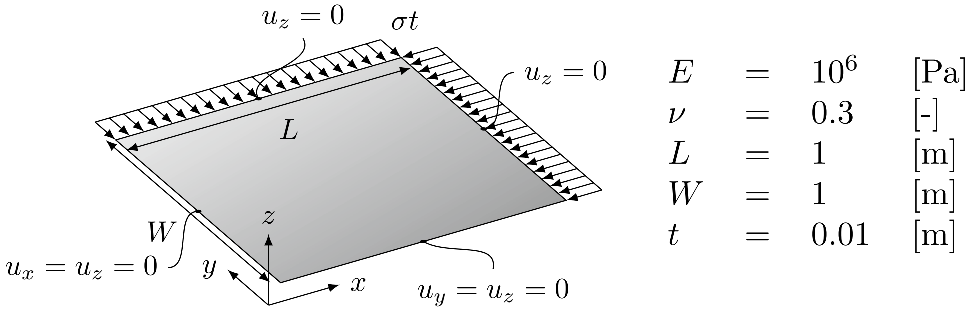


Fig. 10 Geometry and parameters for the plate buckling example. A distributed load of σt is acting on two boundaries in two different directions, and the other boundaries are simply supported and fixed in out-of-plane direction

of a pre-buckling solution and no mass matrix. As an example for buckling analysis, a square simply supported plate is considered, see Fig. 10, with a Saint-Venant Kirchhoff constitutive law with Young's modulus $E = 10^6$ [Pa] and Poisson's ratio $\nu = 0.3$ [-]. The dimensions of the plate are $L \times W \times t = 1 \times 1 \times 0.01$ [m^3]. The plate is subject to a distributed line load of σt in both directions. The analytical solution for the buckling load with m half waves in x-direction and n half waves in y-direction for a square plate with sides L and with equal loads is given in [73] and reads:

$$\sigma_c^{m,n} t = \frac{D\pi^2}{L^2}\left(m^2 + n^2\right), \tag{59}$$

with $D = Et^3/12(1-\nu^2)$ the flexural rigidity of the plate. Using this expression, the first four unique modes are, indexed in ascending order: $\sigma_c^{1,1}t = \sigma_c^1 t = 1.808$ [N/m], $\sigma_c^{2,1}t = \sigma_c^{1,2}t = \sigma_c^2 t = 4.519$ [N/m], $\sigma_c^{2,2}t = \sigma_c^3 t = 7.230$ [N/m], $\sigma_c^{3,1}t = \sigma_c^{1,3}t = \sigma_c^4 t = 9.038$ [N/m].

Figure 11 depicts the analytical error $\Delta\mathcal{L}_{\text{an}}$ and the DWR error estimate $\Delta\mathcal{L}_{\text{num}}$ as a function of the mesh size h for uniform refinements. Both errors are normalised with the analytical value of the critical buckling load. As can be seen in this figure, the DWR prediction of the error converges with a rate of convergence of $2(p-1)$ for all degrees p until it reaches values of around 10^{-10} after which the errors stagnate and increase again (in particular for $p = 4$). This behaviour is similar to the behaviour observed in [74] and can be attributed to the round-off errors as discussed. These errors occur when the number of degrees of freedom is large enough and the machine precision is limited. In case of a buckling simulation, where the non-linear stiffness operator is constructed on an initial solution of a linear simulation, the influence of round-off errors is

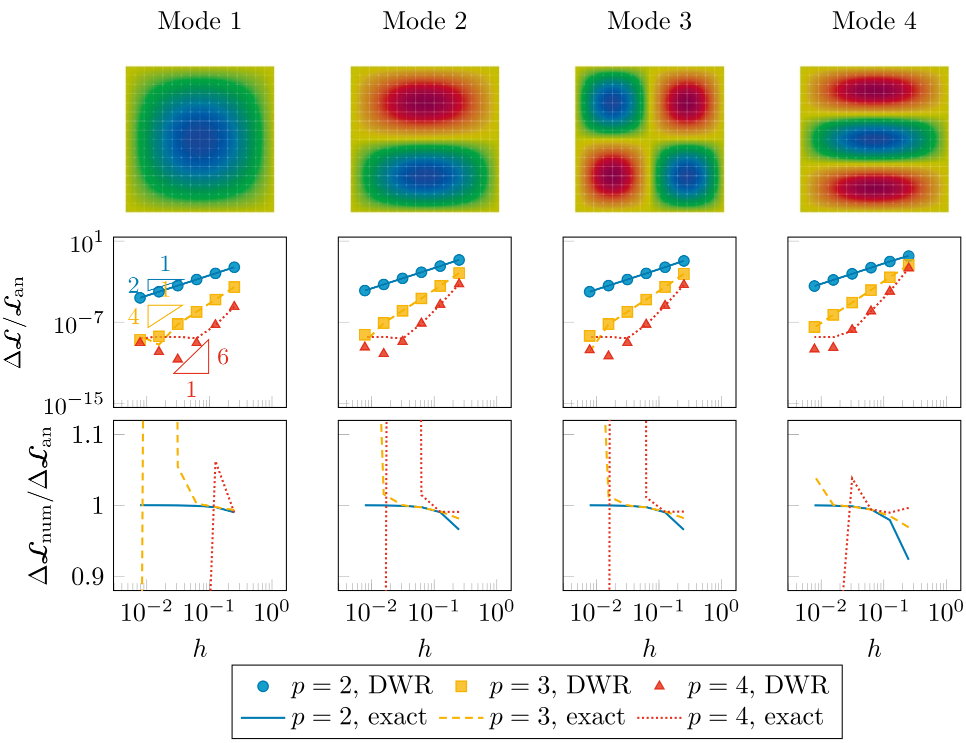
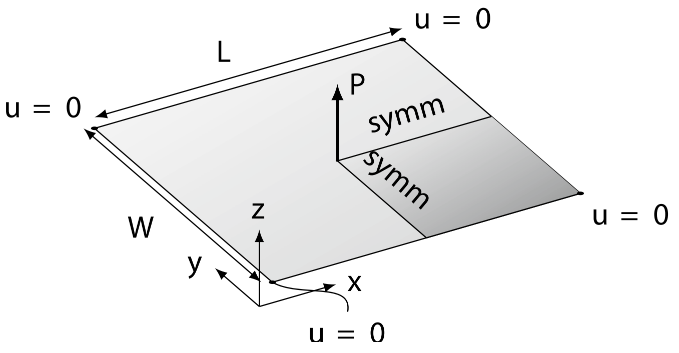


Fig. 11 Buckling analysis of a square plate with simply supported boundary conditions. The top row provides the mode shapes from mode 1 up to 4. The bottom row provides $\Delta\mathcal{L}$ against the uniform mesh sise h. The markers represent error estimates computed via the DWR and the lines represent the exact error, i.e. the error of the numerical solution with respect to the analytical solution. The critical loads for unique modes 1 up to 4 are, respectively, $\sigma_c^{1,1}t = \sigma_c^1 t = 1.808$ [N/m], $\sigma_c^{2,1}t = \sigma_c^{1,2}t = \sigma_c^2 t = 4.519$ [N/m], $\sigma_c^{2,2}t = \sigma_c^3 t = 7.230$ [N/m], $\sigma_c^{3,1}t = \sigma_c^{1,3}t = \sigma_c^4 t = 9.038$ [N/m]



E	=	1.0	[MPa]
ν	=	0.3	[-]
P	=	$4 \cdot 10^{-7}$	[N]
L	=	1	[mm]
W	=	1	[mm]
t	=	10^{-3}	[mm]

Fig. 12 Geometry and parameters for a square thin plate subject to a point load P in the middle. The plate is fully constrained in every corner. Because the problem is symmetric, only a quarter of the domain is modelled. Hence, symmetry conditions are applied. On the x-aligned symmetry axis, this implies that $u_y = \frac{\partial u_z}{\partial y} = \frac{\partial u_x}{\partial y} = 0$ and on the y-aligned symmetry axis this implies that $u_x = \frac{\partial u_z}{\partial x} = \frac{\partial u_y}{\partial x} = 0$

expected to occur sooner. In addition, it can also be seen that the error computed using the analytical solution stagnates. This is due to the fact that the numerical approximation of the critical buckling load shows small variations depending on the solution to the linear problem that is solved to obtain the tangential stiffness matrix to compute the generalised eigenvalue problem for buckling. For the results presented in Fig. 11, the load $\sigma t = 10^{-4}$ [N/m] was used to compute the buckling linearisation.

6.4 Non-linear analysis of a pinched thin plate

In a next example, we consider a square membrane subject to a point-load in the middle and with corners fixed in all directions, see Fig. 12. The membrane is modelled with a Saint-Venant Kirchhoff constitutive law with Young's modulus $E = 1.0\,[MPa]$ and a Poisson ratio $\nu = 0.3$. The thickness of the membrane is $t = 10^{-3}$ [mm] and the length and with are $L \times W = 1 \times 1$ [mm]. The simulation is performed on a quarter of the domain, employing symmetry conditions as depicted in Fig. 12. A load of $P = 4 \cdot 10^{-7}$ [N] is applied in the center of the sheet. The static load case is solved using

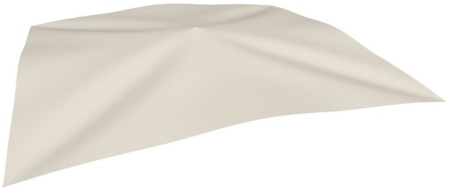


Fig. 13 Deformed surface from the benchmark presented in Fig. 12. The result is the last solution from the adaptive meshing routine with deformation norm goal-functional of which the results are presented in Fig. 14

an arc-length method to ensure convergence of the solution. Furthermore, an adaptive refinement strategy is employed with admissible refinement. The jump parameter m is set to 2 and the maximum number of refinement levels is 8 or 11, which equals a tensor basis level with $2^8 \times 2^8 = 256 \times 256$ or $2^{11} \times 2^{11} = 2048 \times 2048$ elements, respectively. The refinement parameter is set to $\rho_r = 0.5$. The goal functionals considered in this case are based on displacements as well as on principal stresses:

$$\mathcal{L}(\mathbf{u}) = \|\mathbf{u}(\mathbf{x}_P)\|,$$
$$\mathcal{L}(\mathbf{u}) = \int_\Omega \boldsymbol{\sigma}_p \cdot \mathbf{e}_y \mathrm{d}\Omega. \tag{60}$$

Instead of being a domain-integrated goal functional, the first goal functional is evaluated on the point $\mathbf{x}_P$ where the force P is applied, see Fig. 12.

Figure 13 presents the deformed membrane for the last step of the adaptive simulation. Furthermore, Fig. 14 presents the estimated error $\Delta\mathcal{L}$ given the goal functionals in Eq. (60) for the uniform refinement as well as for the adaptive refinement simulation with maximum level 8 or 11. Moreover, Figs. 15 and 16 provide the absolute element-wise errors for the adaptive refinement simulation and for the uniform refinement series for both considered goal functionals. The contour lines in these error fields represent the vertical deflection of the sheet.

From all provided results, it can be observed that the adaptive mesh provides for both goal functionals an efficient converging mesh, where the accuracy per degree of freedom is higher compared to uniformly refined meshes. However, it can also be seen that the total error $\Delta\mathcal{L}$ is not strictly decreasing for both goal functionals. The bottom plots of Fig. 14 indicate that this point is closely related to the maximum depth that is reached: the percentage of the total element error e that can still be refined rapidly decreases, meaning that the only elements that are still available for refinement are the ones that have insignificant contribution to the total element error e, deeming refinement of these elements meaningless. It can be seen from Figs. 15 and 16 that the maximum element depth is reached in the corner where the sheet is fixed and in the corner where the load is applied. After the maximum level is reached, the refined elements start distributing over the diagonal of the domain. For both goal functionals, Fig. 14 shows that some further decrease in the total error $\Delta\mathcal{L}$ can be gained after the maximum error is reached, but that it is most effective to increase the maximum refinement depth. Comparing the error fields and meshes for both goal functionals (see Figs. 15 and 16), it can be seen that the second-principal stress-based error field shows slightly wider error bands in the finest depicted uniform refinement error fields than the displacement-based

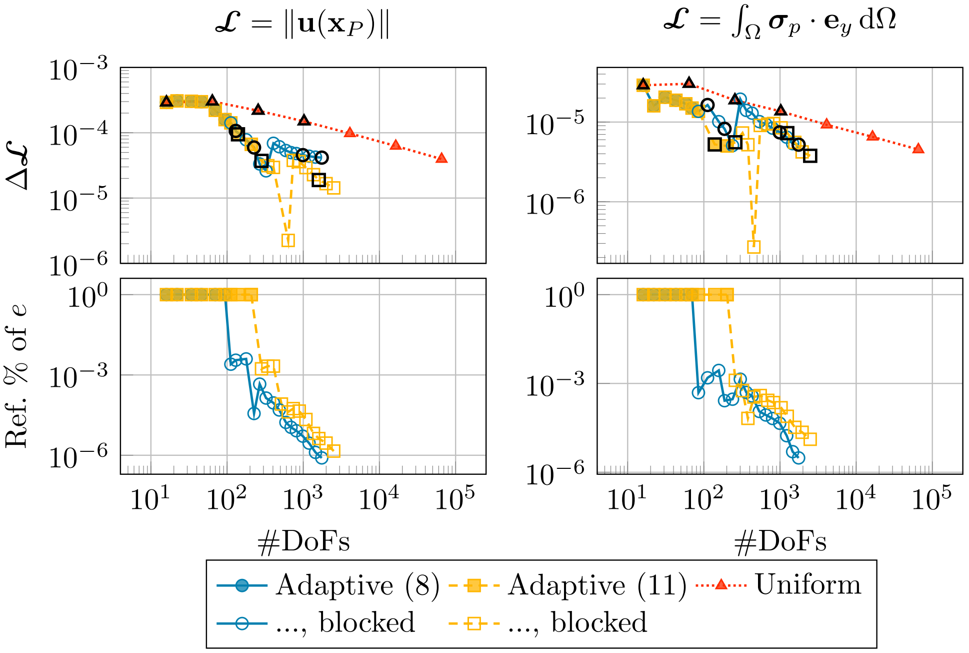
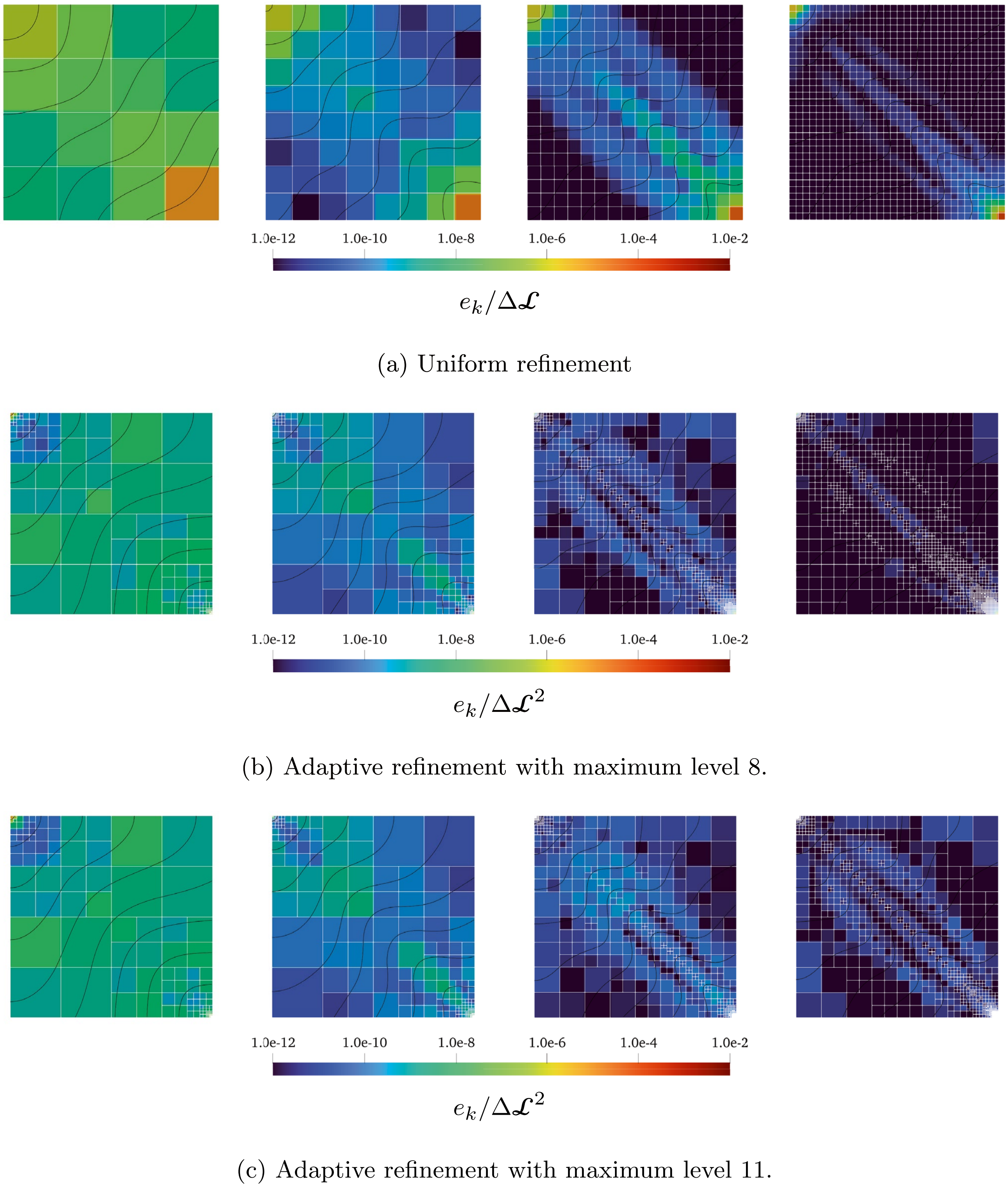


Fig. 14 Estimated error convergence (top) and the percentage of the total element error e that is available for refinement (bottom) against the number of degrees of freedom (DoFs) for adaptively and uniformly refined meshes with respect to the goal functionals from Eq. (60). The markers labeled with a black border are the markers for which the mesh is plotted in Figs. 15 and 16. The filled markers represent points where refinement is not blocked, and empty markers points where refinement is blocked because the maximum refinement depth is reached. Note that the errors are computed before refinement, hence blocked elements in iteration i have effect on the error computation in iteration $i + 1$

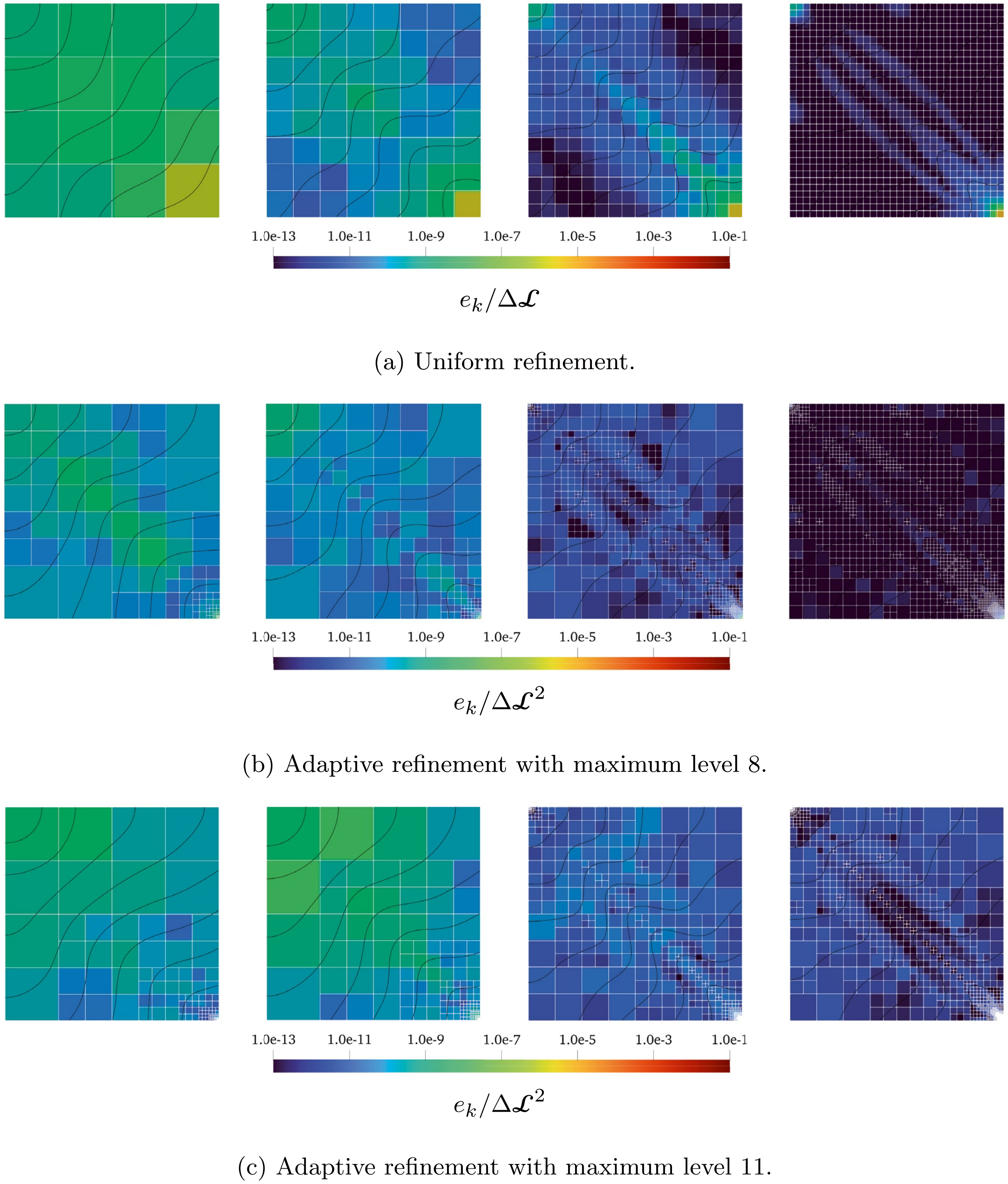


(a) Uniform refinement

(b) Adaptive refinement with maximum level 8.

(c) Adaptive refinement with maximum level 11.

Fig. 15 Normalised element error values $e_k/\Delta\mathcal{L}$ for uniformly (**a**) and adaptively (**b**, **c**) refined meshes using goal-function $\mathcal{L}(\mathbf{u}) = \|\mathbf{u}(\mathbf{x}_P)\|$. The meshing steps increase from left to right. The contour lines represent the displacement of the membrane, with intervals of 0.1 [mm]. The bottom right corner of the pictures indicates the fixed corner and the top left corner is the corner where the load is applied



(a) Uniform refinement.

(b) Adaptive refinement with maximum level 8.

(c) Adaptive refinement with maximum level 11.

Fig. 16 Normalised element error values $e_k/\Delta\mathcal{L}^2$ for uniformly (**a**) and adaptively (**b**, **c**) refined meshes using goal-function $\mathcal{L}(\mathbf{u}) = \int_\Omega \boldsymbol{\sigma}_p \cdot \mathbf{e}_y \mathrm{d}\Omega$. The meshing steps increase from left to right. The contour lines represent the displacement of the membrane, with intervals of 0.1 [mm]. The bottom right corner of the pictures indicates the fixed corner and the top left corner is the corner where the load is applied

error estimator. As a consequence, the corresponding adaptive meshes show that the elements indeed tend to be broader distributed along the diagonal of the domain in case of the displacement-driven refinement (Fig. 15).

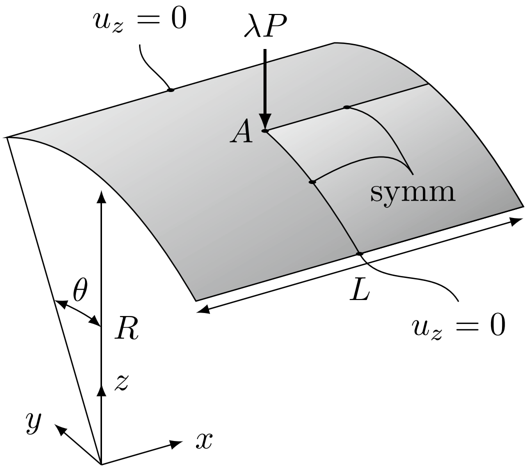
Concluding, the non-linear shell benchmark shows that an adaptive meshing strategy provides accurate solutions on meshes with a small number of degrees of freedom compared to uniform meshes. Furthermore, the benchmarks show the importance of refinement levels, meaning that convergence of the adaptive meshing strategy vanishes as soon as the elements contribution to a large extent to the total error are on the lowest allowed level. This stresses the relevance of spline constructions that allow for deep levels of refinement with moderate computational costs.

6.5 Snap-through instability of a cylindrical roof

In order to present results of the adaptive isogeometric method developed in this paper for quasi-static problems, hence completing full cycles in Fig. 1, the well-known benchmark of a collapsing cylindrical roof [75] subject to a point-load is considered. The goal of this benchmark problem is to evaluate whether the presented adaptive isogeometric method provides DoF-wise efficient solutions compared to solutions with uniform refinements.

The geometry for the benchmark problem is presented in Fig. 17. Here, the radius of the roof $R = 2.540$[m], the angle is $\theta = 0.1$ [rad] and the length and thickness are, respectively, $L = 0.508$ [m] and $t = 6.35 \cdot 10^3$ [mm]. Moreover, the material properties are $E = 3102$ [MPa] and $\nu = 0.3$ [-] for a Saint-Venant Kirchhoff material. Only a quarter of the roof is modelled, as depicted in Fig. 17, because of symmetry. The simulation is performed using a Crisfield arc-length method [65] with arc-length $\Delta L = 25$ and a zero force-scaling. The goal functional that is used for error evaluation and adaptivity is based on the norm of the flexural strain tensor over the whole domain $\mathcal{L} = \int_\Omega \|\mathbf{m}(\mathbf{u})\| \mathrm{d}\Omega$. The jump parameter for admissible meshing is set to $m = 2$. The mesh will be refined when $\Delta\mathcal{L} > \text{tol}_r$ and coarsened when $\Delta\mathcal{L} < \text{tol}_c$. As discussed in Sect. 5, $\text{tol}_r < \text{tol}_c$ such that the mesh is refined and coarsened simultaneously when $\Delta\mathcal{L} \in [\text{tol}_r, \text{tol}_c]$, which is also the condition for termination. The maximum number of mesh adaptivity iterations is set to 5 in this case. The tolerances tol_r and tol_c are determined based on the results of uniformly refined simulations with 16×16 (918 DoFs) and 32×32 elements (3366 DoFs) by taking a wide band around the error envelopes in Sect. 6.5 excluding peaks. The tolerances are $(\text{tol}_r, \text{tol}_c) = (10^{-10}, 10^{-8})$. It should be noted that these tolerances can also be based on requirements in engineering, or they can be determined during the computations; both are beyond the scope of this paper. The refinement parameter is set to $\rho_r = 0.5$, the coarsening parameter to $\rho_c = 0.05$ and the maximum refinement level is 11. The adaptive and uniform meshes are modelled with bi-cubic B-spline basis functions (i.e. $p = 3$).

In Fig. 19, the results of a simulation of the collapsing roof for the adaptive mesh are plotted in a $\|\mathbf{u}\|$, λP, w_A-space. Reference solutions for the present benchmark problem are typically given in the λP, w_A-space, but since the solution curve is not bijective an alternative coordinate $\|\mathbf{u}\|$ is used to represent solutions for this benchmark problem. This is



E	=	3102	[MPa]
ν	=	0.3	[-]
P	=	$3 \cdot 10^3$	[N]
L	=	0.508	[m]
R	=	2.540	[m]
θ	=	0.1	[rad]
t	=	$6.35 \cdot 10^{-3}$	[m]

Fig. 17 Geometry and parameters for cylindrical roof with a point-load P in the middle. The bottom-right corner of each domain corresponds to the point A: the reference point for which the z-displacements are plotted. The roof is free on the curved edges and simply supported ($u_z = 0$) on the straight edges. As a consequence, the problem is symmetric and a quarter of the domain is modelled. On the x-aligned symmetry axis, this implies that $u_y = \frac{\partial u_z}{\partial y} = \frac{\partial u_x}{\partial y} = 0$ and on the y-aligned symmetry axis this implies that $u_x = \frac{\partial u_z}{\partial x} = \frac{\partial u_y}{\partial x} = 0$

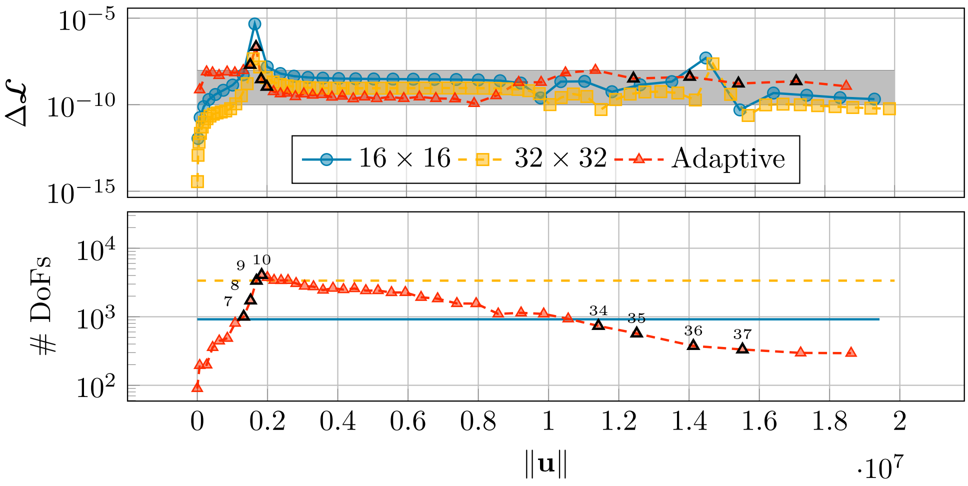


Fig. 18 Errors (top) and number of degrees of freedom (DoFs, bottom) for the goal functional $\mathcal{L} = \int_\Omega \|\mathbf{m}(\mathbf{u})\| \mathrm{d}\Omega$ with tolerances $\mathrm{tol}_r = 10^{-10}$ and $\mathrm{tol}_c = 10^{-8}$ for the collapsing roof subject to the displacement norm $\|\mathbf{u}\|$. The gray region represents the region $\Delta\mathcal{L} \in [\mathrm{tol}_r, \mathrm{tol}_c]$ where refinement and coarsening are performed and where the adaptivity iterations are terminated. Above the gray region only refinement is performed and below the gray region only coarsening is performed

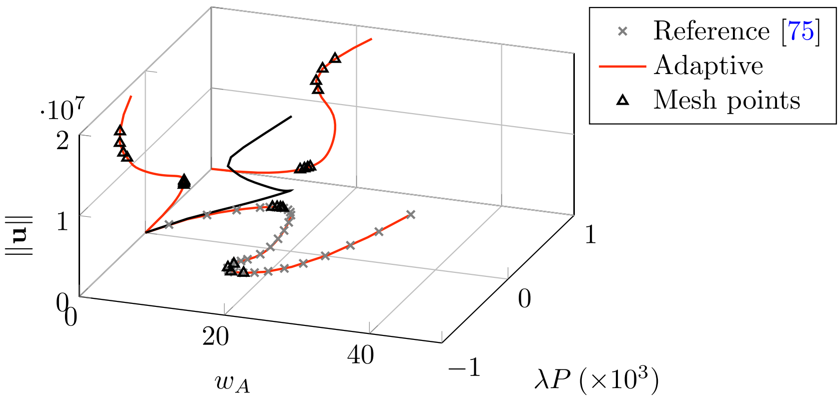
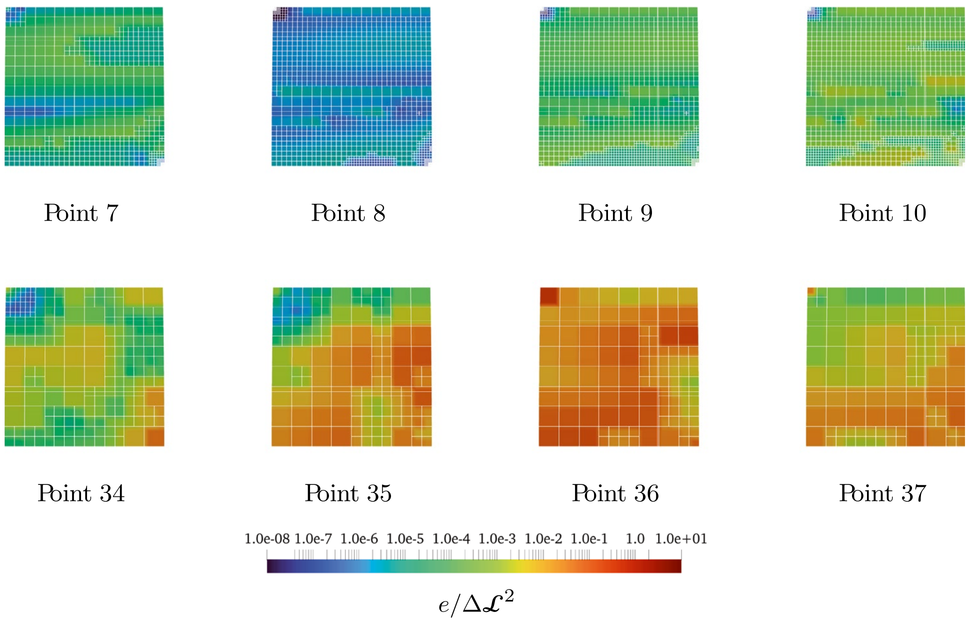


Fig. 19 Projection the adaptively refined result from the commonly used $\lambda P, w_A$-space [75] onto the displacement norm $\|\mathbf{u}\|$. The solid lines correspond to the results obtained by the adaptively refined mesh, the triangular markers correspond to points of which the mesh is provided in Fig. 20 and the cross-markers indicate the result from [75]


Point 7

Point 8

Point 9

Point 10

Point 34

Point 35

Point 36

Point 37

Fig. 20 Meshes corresponding to the points marked in Figs. 19 and 18 by the black-bordered marks. The point A marks the point where the load is applied in Fig. 17. The top row of meshes corresponds to the first limit point in Fig. 19 and the bottom row corresponds to the second limit points. The meshes are ordered from left to right for increasing solution steps. The elements are coloured according to the squared error e_k normalised by the total error $\Delta\mathcal{L}$, i.e. $e_k/\Delta\mathcal{L}^2$

motivated by the projection of the solutions λP and w_A projected against $\|\mathbf{u}\|$ in Fig. 19. The results of [75] are provided as a reference.

In Fig. 18, the error and the number of degrees of freedom are plotted against $\|\mathbf{u}\|$. The error envelopes for the uniform meshes show that a large peak occurs around $\|\mathbf{u}\| = 0.2$, relating to the first limit point of the collapse of the roof, as seen by the markers in Fig. 19. Additionally, it can be seen that the present algorithm providing mesh adaptivity manages to keep the error within specified bounds (see Fig. 18, top), except on the peak just before $\|\mathbf{u}\|$ where the maximum number of adaptivity iterations is insufficient. Furthermore, it has consistently less degrees of freedom than the uniform mesh with 32×32 elements. In Fig. 20 a selection of meshes is provided. The meshes are provided as series of 4 consecutive meshes around the limit points of the solution curve, as indicated in Fig. 19, and are depicted in increasing order from left to right for the first (top) and second (bottom) limit points. The black-bordered markers in Figs. 19 and 18 indicate the points of which the meshes are shown. From the first row of meshes in Fig. 20, it can be seen that the first limit point requires relatively fine meshes and that the elements start concentrated around point A and its diagonal opposite and spread out on the bottom symmetry boundary as the snapping takes place. Furthermore, in the bottom row of Fig. 20, it can be seen that the second limit point does not require many elements, hence the number of elements is slowly decreasing throughout this section of the load–displacement curve. For a complete overview of the mesh in each load-step, we refer to Video 1 in the supplementary material of this paper.

Concluding, this example shows that the goal-adaptive meshing procedure is capable of keeping the error in terms of a goal functions within pre-defined bounds for a solution stepping simulation with limit-point instabilities. Throughout the simulation, the procedure keeps a relatively high efficiency per degree of freedom compared to uniform meshes. It should be noted, however, that the adaptive refinement iterations require a higher computational demand. Therefore, the next and final example provides a procedure where no adaptivity iterations are performed.

6.6 Wrinkling analysis

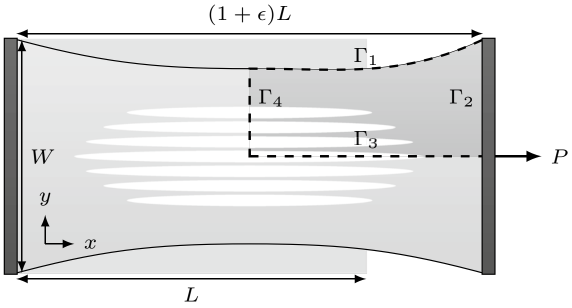
As a last example, the wrinkling analysis of a thin membrane subject to a tensile load is considered. This problem is inspired by [76–78] and was previously modelled using isogeometric Kirchhoff–Love shells in [5], to which we refer for a detailed problem set-up. The goal of this benchmark in the present paper is to demonstrate the use of the goal-adaptive meshing procedure on a bifurcation problem with geometric and material non-linearities.

Given the geometry in Fig. 21, a quarter of the domain is considered with a symmetry boundary condition on Γ_4 and an antisymmetry condition on Γ_3. Furthermore, the boundary Γ_1 is free and on Γ_2 the x-displacement is constant and a horizontal load is applied on this side. The sheet has dimensions $L = 280\,[\text{mm}]$, $W = 140\,[\text{mm}]$ and $t = 0.14\,[\text{mm}]$ such that $L/W = 2$ and $t/W = 10^3$. Furthermore, the material is modelled using a Mooney-Rivlin material model with strain energy density function

$$\Psi(\mathbf{C}) = \frac{c_1}{2}\left(I_1 - 3\right) + \frac{c_2}{2}\left(I_2 - 3\right) \tag{61}$$

and with parameters $c_1 = 3.16 \cdot 10^5\,[\text{Pa}]$ and $c_1 = 1.24 \cdot 10^5\,[\text{Pa}]$. Reference solutions are given by [5] for isogeometric Kirchhoff–Love shell analysis and using ANSYS and LS-DYNA FEA models, respectively. Furthermore, [78] provides experimental data of the maximum amplitude with respect to the strain of the sheet. In the present paper, the reference simulations are performed on uniform cubic meshes with 32×32 and 64×64 elements, respectively.

For the adaptive simulation, a THB spline mesh with initially 32×32 elements is used and mesh adaptivity is activated after wrinkling initiation since the errors in the pre-wrinkling regime are small due to of the lack of out-of-plane deformations of the sheet. The goal-functional is a displacement-based functional on the z-component, i.e. $\mathcal{L}(\mathbf{u}) = \int_\Omega \mathbf{u} \cdot \mathbf{e}_z \mathrm{d}\Omega$. The tolerances for refinement and coarsening are $\text{tol}_r = 10^{-14}$ and $\text{tol}_c = 10^{-10}$, respectively, and they are chosen based on the error envelope of the uniform refinement. The adaptive meshing parameters are chosen as $(\rho_r, \rho_c) = (0.5, 0.005)$. These parameters are chosen based on



c_1	=	$3.13 \cdot 10^5$	[Pa]
c_2	=	$1.24 \cdot 10^5$	[Pa]
L	=	280	[mm]
W	=	140	[mm]
t	=	0.14	[mm]
P	=	Variable	

Fig. 21 Geometry and parameters for the wrinkling problem. The boundary Γ_1 is free, the boundary Γ_2 is fixed in x and y direction and rotations around the y-axis are fixed, the boundaries Γ_3 and Γ_4 have symmetry conditions applied

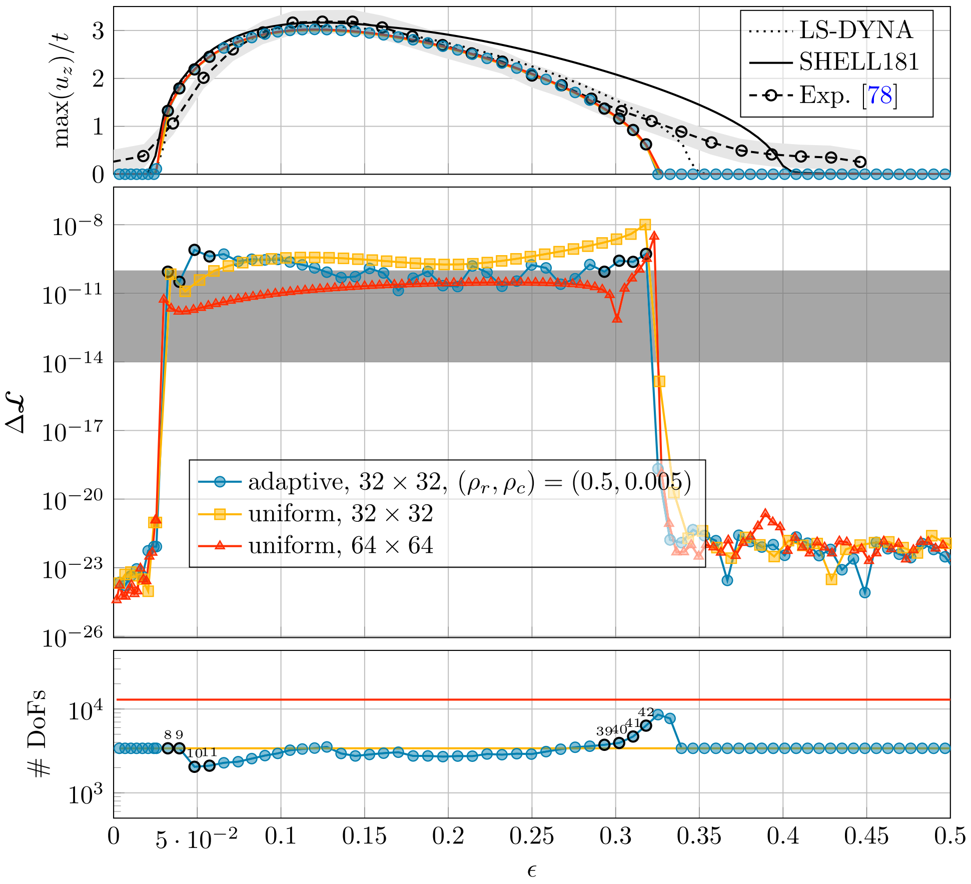


Fig. 22 The non-dimensional maximal out-of-plane deformation max$(u_z)/t$ (top), the goal functional error $\Delta\mathcal{L}$ (mid) and the number of degrees of freedom of the computational mesh (bottom) with respect to the strain of the sheet ϵ. The markers with a black border represent the points of which the meshes are provided in Fig. 23. The colored lines are the solutions obtained by the present model with a Mooney–Rivlin material model with uniform or adaptive meshes. The solid line in the top figure is a SHELL181 result obtained using ANSYS, the dotted line in the top figure is a result obtained using the fully integrated shell in LS-DYNA and the dashed line with markers represents experimental data obtained by [78]

the behaviour of the global error in the first load-steps afer bifurcation. Contrary to the previous example in Sect. 6.5, there are no refinement iterations performed within the load step. This means that the refinement and coarsening operations are performed after the load step based on the magnitude of the error compared to the tolerances. Furthermore, when the error is below a tolerance $\rho_{c,\text{min}}$, the initial mesh is used again. This is done to prevent further coarsening after re-stabilisation of the wrinkles, i.e. the moment when the wrinkles have dissapeared. For the refinement algorithm, a maximum depth of the THB grid is fixed to 11 levels and the jump parameter is set to $m = 2$. The wrinkling simulation is performed using a Crisfield arc-length method [65] with a quadratic procedure to compute the mode shape at the bifurcation [66]. This procedure further described in [67].

In Fig. 22, the results for an adaptive isogeometric wrinkling simulation are provided. The top figure provides the normalised wrinkling amplitude with respect to the strain of the sheet (ϵ), compared to the IGA and ANSYS SHELL181 element reference results from [5] as well as the experimental results from [78]. The figure in the middle provides the error estimate in terms of the goal functional $\mathcal{L}$ with respect to the strain of the sheet and the bottom figure provides the number of degrees of freedom with respect to the strain of the sheet. Firstly, the results from both uniform meshes show that the error estimate $\Delta\mathcal{L}$ is close to zero when the wrinkles initiate, since the sheet is perfectly flat. As soon as the wrinkles form, the error estimate becomes non-zero and it peaks at the moment of re-stabilisation (i.e. the moment when the amplitude vanishes). After re-stabilisation, the error estimate is low, but slightly higher than before wrinkling, probably because the sheet is not numerically flat.

The adaptive meshing simulations show that even with zero inner-iterations for mesh adaptivity the adaptive mesh provides accurate results for a relatively small number of degrees of freedom. Just after wrinkling initiates, the adaptive meshing error peaks due to the coarsening of a large number of elements (as can be seen by the drop of degrees of freedom in the bottom figure). However, the mesh adaptively refines until in the gray region in the middle figure, where combined refinement and coarsening imply that the error balances around tol$_c$, i.e. the upper bound of the marked region. Towards the end of the wrinkling phase, the error increases for the uniformly refined mesh, explaining the increase in number of degrees of freedom for the adaptive mesh. Nevertheless, the adaptive mesh provides more accurate results compared to the 64 × 64 uniform mesh with less degrees of freedom.

In Fig. 23 a selection of meshes from both adaptive simulations are provided, specifically for the wrinkling initiation and

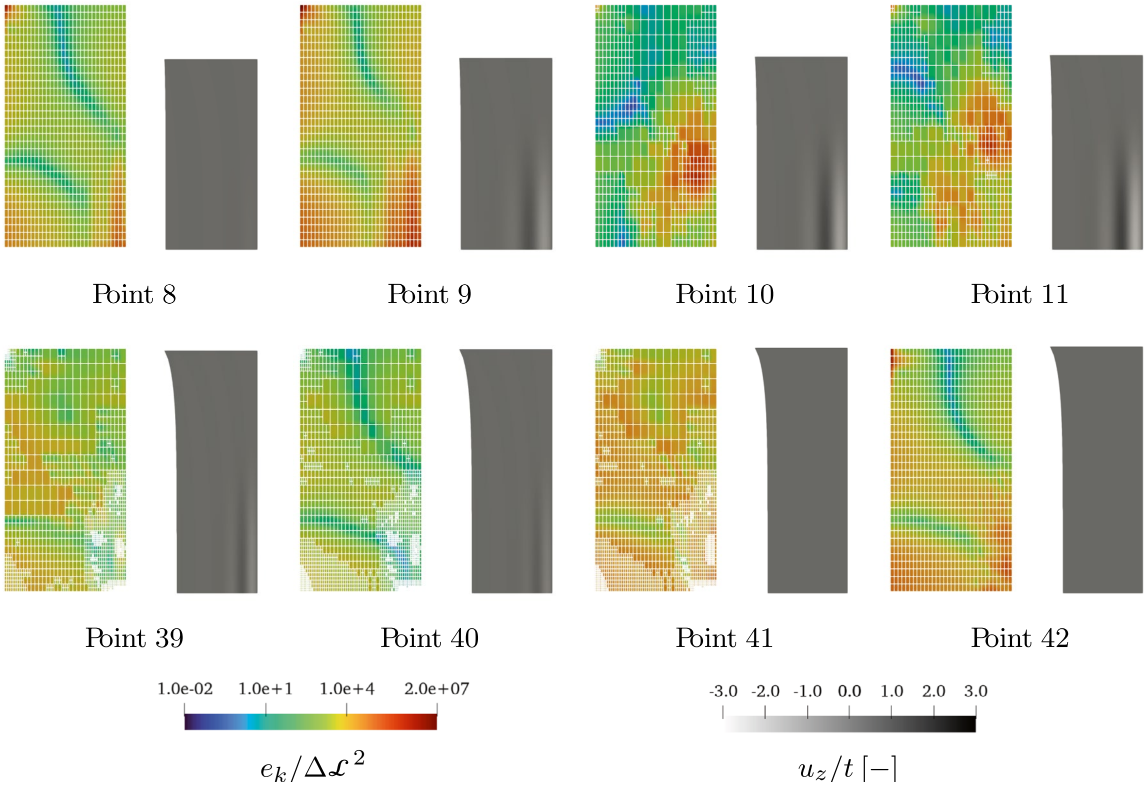


Fig. 23 Normalised element errors (left) and the normalised out-of-plane deformation (right) for the wrinkling benchmark plotted on the undeformed geometry with the corresponding elements

re-stabilisation points. For a complete overview of the mesh in each load-step, we refer to Video 2 in the supplementary material of this paper. The evolution of the meshes shows that the mesh elements concentrate around the wrinkles (bottom-right) and in the top-left corner, which represents the corner between the clamped edge (top) and the free edge. The former is expected since the employed goal functional is based on out-of-plane deformations. However, the fact that the mesh concentrates around the top-left corner is non-intuitive, but tells that this area is important to reduce the global error in terms of the out-of-plane deformations. Furthermore, it can be seen that around the re-stabilisation of the wrinkles the mesh concentrates around the bottom-left corner, indicating that this corner is of importance in accurately modelling the wrinkling amplitudes in the whole domain.

Concluding, the wrinkling benchmark shows the potential of mesh adaptivity for such applications. With fewer degrees of freedom, the THB-spline mesh is able to approximate the solution around a pre-defined error, eventhough the selection of the refinement and coarsening parameters and the tolerances has not been optimised in this study.

7 Conclusions

This paper presents an adaptive method for isogeometric Kirchhoff–Love shells. The main contributions of this paper are a goal-adaptive error estimator for isogeometric Kirchhoff–Love shells using the Dual-Weighted Residual method and a slightly modified suitably graded refinement scheme taking into account refined elements in the definition of the coarsening neighborhood.

Using the Dual-Weighted Residual method and given a pre-defined goal functional (e.g the second-principal stress integrated over the domain), an estimator for the error in terms of this goal functional can be defined. The adjoint problem that needs to be solved on the original mesh and on a nested degree-elevated (‘enriched’) mesh has been defined for the isogeometric Kirchhoff–Love shell. In addition, the operators for modal and linear buckling analysis have been derived, implying an additional generalised eigenvalue problem to be solved on the enriched mesh. For suitable grading, the works of [22, 23, 25, 50] have been closely followed. In order to be able to refine and coarsen in the same iteration, refined elements have been added to the original definition of the coarsening neighborhood.

To assess the proposed adaptive isogeometric method for Kirchhoff–Love shells, few numerical benchmark problems have been evaluated. Linear static analysis provided an analytical solution has been used to evaluate the DWR error estimators. The eigenvalue problems for modal and buckling analyses have been evaluated on respectively the problem of circular plate vibration and square plate buckling. Based on the linear, modal and buckling analysis with analytical solutions, it can be concluded that the DWR estimator for Kirchhoff–Love shells can be used with several goal functionals and in several applications, as it provides high accuracy with respect to the exact errors.

Using the problem of a pinched membrane, the error estimator has been used to adaptively refine a mesh in a non-linear example. From this example, it can be concluded that eventhough the challenging non-linear problem, high accuracy per DoF can be obtained compared to uniformly refined meshes based on different goal functionals. Lastly, the adaptive isogeometric method from the present paper has been evaluated in solution-stepping problems for structural instabilities. Firstly, the method was applied to limit-point instability problem of a collapse of a cylindrical roof. Here, inner adaptivity iterations were performed for each load-step until the error was located in a desired interval. Again, the present method provides a high accuracy per degree of freedom. In addition, it was shown that the method is indeed able to provide adaptive meshes with respect to a pre-defined interval for a given goal functional. The last benchmark problem involves a tension-wrinkling bifurcation instability of a thin membrane. In this benchmark problem, adaptive meshing has been applied in the post-buckling regime based on wrinkling amplitudes. Here, no adaptivity iterations within the load steps were performed, showing that the method is still able to provide good adaptivity with respect to the pre-defined tolerances. Also, the results of the wrinkling error estimators show that the error peaks in the re-stabilisation phase, where the highest deviation with experimental results are observed.

Future developments for the present method include the application on multi-patch domains, both coupled with penalty methods as well as with globally continuous bases as presented in [79] to handle more complex geometries. Furthermore, structural dynamics have been left out of the scope of this paper, since the DWR for dynamic problems requires backwards-in-time evaluation of the adjoint problem, which is ideally combined with parallel-in-time methods like ParaReal or MGRIT [80]. Lastly, future work can be done on the (adaptive) determination of the adaptive meshing parameters. On the one hand, one can apply the present method on real-world engineering applications, taking realistic goal functionals and margins into the adaptivity algorithm. On the other hand, advanced schemes for triggering pure or combined refinement or coarsening together with their parameters can be futher investigated.

Author Contributions The authors contributed to this paper in the following way. H. M. Verhelst: conceptualization, formal analysis, investigation, methodology, software, validation, visualization, writing – original draft, writing – review and editing. A. Mantzaflaris: software, writing – review and editing. M. Möller: conceptualization, software, writing – review & editing, funding acquisition, project administration, supervision. J.H. Den Besten: conceptualization, writing – review and editing, funding acquisition, project administration, supervision.

Funding H.M. Verhelst, M. Möller and J.H. Den Besten are greatful to the faculties of Electrical Engineering, Mathematics and Computer Science (EEMCS) and of Mechanical, Materials and Maritime Engineering (3mE) of Delft University of Technology for the financial support to conduct this research. A. Mantzaflaris is greatful for the financial of the EU ITN network GRAPES.

Data availability The data used in the present paper are obtained using the Geometry + Simulation modules version v23.12, and instructions for data reproduction are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Hughes TJR, Cottrell JAA, Bazilevs Y (2005) Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput Methods Appl Mech Eng 194(39–41):4135–4195 arXiv:1608.04366
2. Sande E, Manni C, Speleers H (2020) Explicit error estimates for spline approximation of arbitrary smoothness in isogeometric analysis. Numerische Mathematik 144(4):889–929
3. Kiendl J, Bletzinger K-U, Linhard J, Wüchner R (2009) Isogeometric shell analysis with Kirchhoff–Love elements. Comput Methods Appl Mech Eng 198(49–52):3902–3914
4. Kiendl J, Hsu M-C, Wu MCH, Reali A (2015) Isogeometric Kirchhoff–Love shell formulations for general hyperelastic materials. Comput Methods Appl Mech Eng 291:280–303
5. Verhelst HM, Möller M, Den Besten JH, Mantzaflaris A, Kaminski ML (2021) Stretch-based hyperelastic material formulations for isogeometric Kirchhoff–Love shells with application to wrinkling. Comput Aided Design 139:103075
6. Alaydin MD, Benson DJ, Bazilevs Y (2021) An updated Lagrangian framework for Isogeometric Kirchhoff–Love thin-shell analysis. Comput Methods Appl Mech Eng 384:113977

7. Benson DJ, Bazilevs Y, Hsu M-C, Hughes TJR (2011) A large deformation, rotation-free, isogeometric shell. Comput Methods Appl Mech Eng 200(13–16):1367–1378
8. Hu Q, Xia Y, Natarajan S, Zilian A, Hu P, Bordas SPA (2020) Isogeometric analysis of thin Reissner–Mindlin shells: locking phenomena and B-bar method. Computat Mech 65(5):1323–1341
9. Kiendl J, Auricchio F, Beirão da Veiga L, Lovadina C, Reali A (2015) Isogeometric collocation methods for the Reissner–Mindlin plate problem. Comput Methods Appl Mech Eng 284:489–507
10. Sobota PM, Dornisch W, Müller R, Klinkel S (2017) Implicit dynamic analysis using an isogeometric Reissner–Mindlin shell formulation. Int J Numer Methods Eng 110(9):803–825
11. Benson DJ, Hartmann S, Bazilevs Y, Hsu M-C, Hughes TJR (2013) Blended isogeometric shells. Comput Methods Appl Mech Eng 255:133–146
12. Hosseini S, Remmers JJC, Verhoosel CV, Borst R (2013) An isogeometric solid-like shell element for nonlinear analysis. Int J Numer Methods Eng 95(3):238–256
13. Leonetti L, Liguori F, Magisano D, Garcea G (2018) An efficient isogeometric solid-shell formulation for geometrically nonlinear analysis of elastic shells. Comput Methods Appl Mech Eng 331:159–183
14. Coradello L, D'Angella D, Carraturo M, Kiendl J, Kollmannsberger S, Rank E, Reali A (2020) Hierarchically refined isogeometric analysis of trimmed shells. Comput Mech 66(2):431–447
15. Guo Y, Heller J, Hughes TJR, Ruess M, Schillinger D (2018) Variationally consistent isogeometric analysis of trimmed thin shells at finite deformations, based on the STEP exchange format. Comput Methods Appl Mech Eng 336:39–79
16. Leidinger LF, Breitenberger M, Bauer AM, Hartmann S, Wüchner R, Bletzinger KU, Duddeck F, Song L (2019) Explicit dynamic isogeometric B-Rep analysis of penalty-coupled trimmed NURBS shells. Comput Methods Appl Mech Eng 351:891–927
17. Herrema AJ, Johnson EL, Proserpio D, Wu MCH, Kiendl J, Hsu M-C (2019) Penalty coupling of non-matching isogeometric Kirchhoff–Love shell patches with application to composite wind turbine blades. Comput Methods Appl Mech Eng 346:810–840
18. Leonetti L, Liguori FS, Magisano D, Kiendl J, Reali A, Garcea G (2020) A robust penalty coupling of non-matching isogeometric Kirchhoff–Love shell patches in large deformations. Comput Methods Appl Mech Eng 371:113289
19. Bouclier R, Passieux JC, Salaün M (2017) Development of a new, more regular, mortar method for the coupling of NURBS subdomains within a NURBS patch: application to a non-intrusive local enrichment of NURBS patches. Comput Methods Appl Mech Eng 316:123–150
20. Guo Y, Ruess M (2015) Nitsche's method for a coupling of isogeometric thin shells and blended shell structures. Comput Methods Appl Mech Eng 284:881–905
21. Coradello L, Kiendl J, Buffa A (2021) Coupling of non-conforming trimmed isogeometric Kirchhoff–Love shells via a projected super-penalty approach. Comput Methods Appl Mech Eng 387:114187 arxiv:2104.13804
22. Buffa A, Giannelli C (2016) Adaptive isogeometric methods with hierarchical splines: error estimator and convergence. Math Models Methods Appl Sci 26(01):1–25
23. Buffa A, Gantner G, Giannelli C, Praetorius D, Vázquez R (2022) Mathematical Foundations of Adaptive Isogeometric Analysis. Arch Comput Methods Eng 29(7):4479–4555
24. D'Angella D, Reali A (2020) Efficient extraction of hierarchical B-Splines for local refinement and coarsening of Isogeometric Analysis. Comput Methods Appl Mech Eng 367:113131
25. Carraturo M, Giannelli C, Reali A, Vázquez R (2019) Suitably graded THB-spline refinement and coarsening: towards an adaptive isogeometric analysis of additive manufacturing processes. Comput Methods Appl Mech Eng 348:660–679
26. Lorenzo G, Scott MA, Tew K, Hughes TJR, Gomez H (2017) Hierarchically refined and coarsened splines for moving interface problems, with particular application to phase-field models of prostate tumor growth. Comput Methods Appl Mech Eng 319:515–548
27. Hennig P, Ambati M, De Lorenzis L, Kästner M (2018) Projection and transfer operators in adaptive isogeometric analysis with hierarchical B-splines. Comput Methods Appl Mech Eng 334:313–336
28. Garau EM, Vázquez R (2018) Algorithms for the implementation of adaptive isogeometric methods using hierarchical B-splines. Appl Numer Math 123:58–87
29. Antolin P, Buffa A, Coradello L (2020) A hierarchical approach to the a posteriori error estimation of isogeometric Kirchhoff plates and Kirchhoff–Love shells. Comput Methods Appl Mech Eng 363:112919
30. Coradello L, Davide D'angella, Carraturo M, Kiendl J, Kollmannsberger S, Rank E, Reali A (2020) Hierarchically refined isogeometric analysis of trimmed shells. Computat Mech 66:431–447
31. Verhoosel CV, Van Zwieten GJ, Van Rietbergen B, De Borst R (2015) Image-based goal-oriented adaptive isogeometric analysis with application to the micro-mechanical modeling of trabecular bone. Comput Methods Appl Mech Eng 284:138–164
32. Kuru G, Verhoosel CV, Zee KG, Brummelen EH (2014) Goal-adaptive isogeometric analysis with hierarchical splines. Comput Methods Appl Mech Eng
33. Hinz J, Abdelmalik M, Möller M (2020) Goal-oriented adaptive thb-spline schemes for PDE-based planar parameterization. arXiv:2001.08874
34. Rannacher R (2004) Adaptive finite element methods in flow computations. Recent Adv Adapt Computat Contemp Math 383:176–183
35. Gedicke J, Carstensen C (2013) A posteriori error estimators for convection-diffusion eigenvalue problems
36. Hartmann R, Held J, Leicht T, Prill F (2010) Error estimation and adaptive mesh refinement for aerodynamic flows. Not Numer Fluid Mech Multidiscip Design
37. Hartmann R, Houston P (2002) Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. J Comput Phys 183(2):508–532
38. Hartmann R, Houston P (2003) Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws. SIAM J Sci Comput 24(3):979–1004
39. Hartmann R, Houston P (2002) Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. J Comput Phys
40. Möller M, Kuzmin D (2006) Adaptive mesh refinement for high-resolution finite element schemes. Int J Numer Methods Fluids 52(5):545–569
41. Cliffe KA, Hall EJC, Houston P (2010) Adaptive discontinuous galerkin methods for eigenvalue problems arising in incompressible fluid flows. SIAM J Sci Comput 31(6):4607–4632
42. Van Der Zee KG, Verhoosel CV (2011) Isogeometric analysis-based goal-oriented error estimation for free-boundary problems 47:600–609
43. Dedè L, Santos HAFA (2012) B-spline goal-oriented error estimators for geometrically nonlinear rods. Comput Mech 49(1):35–52
44. Dörfler W (1996) A convergent adaptive algorithm for Poisson's equation. SIAM J Numer Anal 33(3):1106–1124
45. Giannelli C, Jüttler B, Kleiss SK, Mantzaflaris A, Simeon B, Špeh J (2016) THB-splines: an effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis. Comput Methods Appl Mech Eng 299:337–365

46. Vuong A-V, Giannelli C, Jüttler B, Simeon B (2011) A hierarchical approach to adaptive local refinement in isogeometric analysis. Comput Methods Appl Mech Eng 200(49–52):3554–3567
47. Giannelli C, Jüttler B, Speleers H (2012) THB-splines: the truncated basis for hierarchical splines. Comput Aided Geom Design 29(7):485–498
48. Bazilevs Y, Calo VMM, Cottrell JAA, Evans JAA, Hughes TJRJR, Lipton S, Scott MAA, Sederberg TWW (2010) Isogeometric analysis using T-splines. Comput Methods Appl Mech Eng 199(5–8):229–263 arXiv:1010.1724
49. Hennig P, Müller S, Kästner M (2016) Bézier extraction and adaptive refinement of truncated hierarchical NURBS. Comput Methods Appl Mech Eng 305:316–339
50. Bracco C, Giannelli C, Vázquez R (2018) Refinement algorithms for adaptive isogeometric methods with hierarchical splines. Axioms 7(3):43
51. Hennig P, Kästner M, Morgenstern P, Peterseim D (2017) Adaptive mesh refinement strategies in isogeometric analysis—a computational comparison. Comput Methods Appl Mech Eng 316:424–448 arXiv:1605.00825
52. Bracco C, Giannelli C, Großmann D, Imperatore S, Mokriš D, Sestini A (2022) THB-spline approximations for turbine blade design with local B-spline approximations 29:63–82 arXiv:2003.08706
53. Kiss G, Giannelli C, Zore U, Jüttler B, Großmann D, Barner J (2014) Adaptive CAD model (re-)construction with THB-splines. Graph Models 76(5):273–288
54. Speleers H, Manni C (2016) Effortless quasi-interpolation in hierarchical spaces. Numerische Mathematik 132(1):155–184
55. Giust A, Jüttler B, Mantzaflaris A (2020) Local (t)HB-spline projectors via restricted hierarchical spline fitting. Comput Aided Geomet Design 80:101865. https://doi.org/10.1016/j.cagd.2020.101865
56. Lathouwers D (2011) Spatially adaptive eigenvalue estimation for the SN equations on unstructured triangular meshes. Ann Nucl Energy 38(9):1867–1876
57. Roohbakhshan F, Sauer RA (2017) Efficient isogeometric thin shell formulations for soft biological materials. Biomech Model Mechanobiol 16(5):1569–1597
58. Sauer RA, Duong TX (2017) On the theoretical foundations of thin solid and liquid shells. Math Mech Solids 22(3):343–371
59. Reddy JN (2014) An introduction to nonlinear finite element analysis: with applications to heat transfer, fluid mechanics, and solid mechanics. Oxford University Press, Oxford
60. Goyal A (2015) Isogeometric shell discretizations for flexible multibody dynamics. doctoralthesis, Technische Universität Kaiserslautern
61. Pan M, Jüttler B, Mantzaflaris A (2021) Efficient matrix assembly in isogeometric analysis with hierarchical b-splines. J Comput Appl Math 390:113278. https://doi.org/10.1016/j.cam.2020.113278
62. Pan M, Jüttler B, Scholz F (2022) Efficient matrix computation for isogeometric discretizations with hierarchical b-splines in any dimension. Comput Methods Appl Mech Eng 388:114210. https://doi.org/10.1016/j.cma.2021.114210
63. Giannelli C, Kanduč T, Martinelli M, Sangalli G, Tani M (2022) Weighted quadrature for hierarchical b-splines. Comput Methods Appl Mech Eng 400:115465. https://doi.org/10.1016/j.cma.2022.115465
64. Riks E (1972) The application of newton's method to the problem of elastic stability. J Appl Mech 39(4):1060
65. Crisfield MAM (1981) A fast incremental/iterative solution procedure that handles "snap-through": 55–62
66. Wriggers P, Wagner W, Miehe C (1988) A quadratically convergent procedure for the calculation of stability points in finite element analysis. Comput Methods Appli Mech Eng 70(3):329–347
67. Verhelst HM, Moller M, Den Besten JH, Vermolen FJ, Kaminski ML (2020) Equilibrium Path Analysis Including Bifurcations with an Arc-Length Method Avoiding A Priori Perturbations. In: Proceedings of ENUMATH2019 Conference
68. Becker R, Rannacher R (2001) An optimal control approach to a posteriori error estimation in finite element methods. Acta Numerica 10:1–102
69. Bangerth W, Rannacher R (2003) Adaptive finite element methods for differential equations, 1st edn. Birkhäuser Basel, Basel
70. Hartmann R, Houston P (2006) Symmetric interior penalty DG methods for the compressible Navier–Stokes equations II: goal-oriented a posteriori error estimation. Int J Numer Anal Model 3(1):141–162
71. Giani S, Grubišić L, Ovall JS (2012) Benchmark results for testing adaptive finite element eigenvalue procedures. Appl Numer Math 62(2):121–140
72. Jüttler B, Langer U, Mantzaflaris A, Moore SE, Zulehner W (2014) Geometry + simulation modules: implementing isogeometric analysis. PAMM 14(1):961–962
73. Jones RM (2006) Buckling of Bars, Plates, and Shells. Bull Ridge Corporation, Blacksburg, Va
74. Liu J, Möller M, Schuttelaars HM (2021) Balancing truncation and round-off errors in FEM: One-dimensional analysis. J Comput Appl Math 386:113219. https://doi.org/10.1016/j.cam.2020.113219
75. Sze KY, Liu XH, Lo SH (2004) Popular benchmark problems for geometric nonlinear analysis of shells. Finite Elem Anal Design 40(11):1551–1569
76. Cerda E, Ravi-Chandar K, Mahadevan L (2002) Wrinkling of an elastic sheet under tension. Nature 419(6907):579–580
77. Cerda E, Mahadevan L (2003) Geometry and Physics of Wrinkling. Phys Rev Lett
78. Panaitescu A, Xin M, Davidovitch B, Chopin J, Kudrolli A (2019) Birth and decay of tensional wrinkles in hyperelastic sheets. arXiv:1906.10054
79. Farahat A, Verhelst HM, Kiendl J, Kapl M (2023) Isogeometric analysis for multi-patch structured Kirchhoff–Love shells. Comput Methods Appl Mech Eng 411:116060. https://doi.org/10.1016/j.cma.2023.116060
80. Falgout RD, Friedhoff S, Kolev TV, MacLachlan SP, Schroder JB (2014) Parallel time integration with multigrid. SIAM J Sci Comput 36(6):635–661