



University
of Glasgow

May, Stefan (2016) *Splines for damage and fracture in solids*.
PhD thesis.

<http://theses.gla.ac.uk/7284/>

Copyright and moral rights for this thesis are retained by the author

A copy can be downloaded for personal non-commercial research or study

This thesis cannot be reproduced or quoted extensively from without first
obtaining permission in writing from the Author

The content must not be changed in any way or sold commercially in any format
or medium without the formal permission of the Author

When referring to this work, full bibliographic details including the author, title,
awarding institution and date of the thesis must be given

Splines for damage and fracture in solids

Stefan May

Submitted in fulfilment of the requirements for the Degree of
Doctor of Philosophy



School of Engineering
University of Glasgow

May 2016

Declaration

I declare that, except where explicit reference is made to the contribution of others, that this dissertation is the result of my own work and has not been submitted for any other degree at the University of Glasgow or any other institution.

Stefan May

Abstract

This thesis addresses different aspects of numerical fracture mechanics and spline technology for analysis.

An energy-based arc-length control for physically non-linear problems is proposed. It switches between an internal energy-based and a dissipation-based arc-length method. The arc-length control allows to trace an equilibrium path with multiple snap-through and / or snap-back phenomena and only requires two parameters.

Phase field models for brittle and cohesive fracture are numerically assessed. The impact of different parameters and boundary conditions on the phase field model for brittle fracture is investigated. It is demonstrated that Γ -convergence is not attained numerically for the phase field model for brittle fracture and that the phase field model for cohesive fracture does not pass a two-dimensional patch test when using an unstructured mesh.

The properties of the BÉZIER extraction operator for T-splines are exploited for the determination of linear dependencies, partition of unity properties, nesting behaviour and local refinement. Unstructured T-spline meshes with extraordinary points are modified such that the blending functions fulfil the partition of unity property and possess a higher continuity.

BÉZIER extraction for POWELL–SABIN B-splines is introduced. Different spline technologies are compared when solving KIRCHHOFF–LOVE plate theory on a disc with simply supported and clamped boundary conditions.

POWELL–SABIN B-splines are utilised for smeared and discrete approaches to fracture. Due to the higher continuity of POWELL–SABIN B-splines, the implicit fourth order gradient damage model for quasi-brittle materials can be solved and stresses can be computed directly at the crack tip when considering the cohesive zone method.

Contents

1	Introduction	1
2	Energy-based arc-length control for physically non-linear problems	5
2.1	Necessity of an arc-length control	5
2.2	Arc-length control based on the rate of dissipated energy	7
2.3	Arc-length control based on the rate of internal energy	10
2.4	Numerical examples	13
2.4.1	Phase field model for brittle fracture	13
2.4.2	Perforated beam with interface elements	16
3	Fracture models using phase field	19
3.1	Phase field expression of the crack	19
3.2	Phase field model for brittle fracture	20
3.2.1	Continuum formulation	21
3.2.2	Finite element formulation	25
3.2.3	Numerical examples	27
3.3	Phase field model for cohesive fracture	37
3.3.1	Continuum formulation	39
3.3.2	Finite element formulation	44
3.3.3	Numerical examples	46
4	T-splines	51
4.1	Continuity for design and analysis	52
4.1.1	Parametric continuity \mathcal{C}	52
4.1.2	Geometric continuity \mathcal{G}	53
4.1.3	Continuity \mathcal{C}_A for analysis	53
4.1.4	Geometric continuous basis functions	54
4.2	Defining T-splines	56
4.2.1	Definition of the domains	56
4.2.2	Definition of the local knot vector	57
4.2.3	Construction of the blending functions	57
4.2.4	Element definition	58
4.3	BÉZIER extraction for T-splines	59

4.4	Classification of T-splines	62
4.4.1	Classification of T-splines according to the type of linear dependence	63
4.4.2	Partition of unity property for T-splines	65
4.5	Local refinement of T-splines by adding anchors	71
4.5.1	Computation of the refinement matrix and nesting behaviour	72
4.5.2	Determination of the coordinates for the anchors in the refined T-spline mesh	73
4.5.3	The algorithm for local refinement of standard T-splines	75
4.5.4	Local refinement of standard T-splines of even degree by adding anchors	76
4.5.5	Summary for the local refinement of standard T-splines	81
4.6	Hierarchical refinement of T-splines using the reconstruction operator	82
4.6.1	Splitting elements	82
4.6.2	Example	84
4.7	Unstructured quadratic T-splines	86
4.7.1	The unstructured T-spline mesh	86
4.7.2	Generalised BÉZIER extraction	87
4.7.3	Modifying the BÉZIER extraction operator	89
4.8	Analysis-suitable T-splines	94
5	POWELL–SABIN B-splines	97
5.1	KIRCHHOFF–LOVE plate theory	98
5.1.1	Continuum formulation	98
5.1.2	Finite element formulation	99
5.2	BÉZIER extraction for quadratic POWELL–SABIN B-splines	100
5.2.1	Quadratic POWELL–SABIN B-splines	100
5.2.2	BÉZIER extraction	102
5.2.3	Patch test	106
5.3	The representation of a disc using NURBS, T-splines, NURBS-to-NURPS and POWELL–SABIN B-splines	106
5.3.1	Representation of a disc using NURBS	107
5.3.2	Representation of a disc using unstructured T-splines	109
5.3.3	Representation of a disc using the NURBS-to-NURPS methodology	111
5.3.4	Representation of a disc using POWELL–SABIN B-splines	111
5.4	Numerical examples	113

6 POWELL–SABIN B-splines for smeared and discrete fracture approaches	117
6.1 Implicit higher order gradient damage model	117
6.1.1 Continuum formulation	117
6.1.2 Finite element formulation	120
6.1.3 Numerical examples	122
6.2 Cohesive zone modelling	129
6.2.1 Continuum formulation	130
6.2.2 Finite element formulation	132
6.2.3 Re-meshing	135
6.2.4 Single-edge notched beam	136
6.2.5 Structured T-splines for the single-edge notched beam	138
7 Summary	141
A Energy-based arc-length control for physically non-linear problems	143
A.1 SHERMAN–MORRISON formula	143
A.2 Time discretisation scheme for the arc-length control	145
A.2.1 Time discretisation for the rate of internal energy	145
A.2.2 Time discretisation for the rate of dissipated energy	146
B T-splines	147
B.1 Local refinement of T-splines by adding anchors	147
B.1.1 Local refinement of standard T-splines of odd degree by adding anchors	147
B.1.2 Local refinement of non-standard T-splines by adding anchors	153
B.1.3 Obtaining the optimised number of additionally inserted anchors	154
B.2 Modified local knot vectors for the hierarchical refinement	156
B.3 Unstructured cubic T-splines	156
B.3.1 The unstructured T-spline mesh	156
B.3.2 Modifying the BÉZIER extraction operator	157
Bibliography	171

1 Introduction

The prediction of crack propagation and failure of materials is one of the big issues engineering mechanics is currently dealing with. Fracture can be modelled by either a discrete or a smeared approach utilising the finite element method. In the discrete approach, the crack is modelled explicitly whereas in the smeared approach, the crack is distributed over a damage zone with a finite width.

Re-meshing allows the mesh to follow the crack in the discrete approach as demonstrated by INGRAFFEA, SWENSON and WAWRZYNEK [108, 121] for the theory of linear elastic fracture mechanics (LEFM). LEFM can be used to describe crack phenomena in brittle materials such as glass (GRIFFITH [45]), and ductile materials such as steel when GRIFFITH's theory is further modified, see IRWIN [53] and OROWAN [81]. GRIFFITH's fracture theory is based on energy rather than local stresses. However, LEFM is only applicable when the plastic region at the crack tip is small. It is not suited to predict failure in quasi-brittle materials such as concrete and rock which exhibit the formation of micro-cracks ahead of the crack tip. Therefore, HILLERBORG *et al.* [50] added cohesive zones in the vicinity of the crack tip for describing failure in quasi-brittle materials in a finite element computation. Cohesive zone modelling represents an application of the approach by BARENBLATT [5] and DUGDALE [38]. Yet, complications arise for discrete fracture models when describing phenomena like crack branching or modelling curved cracks in three dimensions.

This has motivated the use of smeared fracture approaches. They do not require re-meshing which eases their implementation. Smeared approaches represent / approximate a crack by introducing a damage variable and can be straightforwardly extended to three dimensions. However, damage approaches yield different solutions upon mesh refinement as discussed by CRISFIELD [26], DE BORST [28] and KUHLE [62]. Thus, non-local damage approaches were developed by BAŽANT and PIJAUDIER-CABOT [8, 87]. Non-local approaches introduce an additional parameter – the length scale parameter. The implicit gradient damage model for quasi-brittle materials by PEERLINGS *et al.* [82] allows for a more effective finite element solution when introducing non-locality.

Closely related to the implicit gradient damage model is the phase field model for brittle fracture which also introduces a length scale parameter, see BOURDIN *et al.* [17], FRANCFORT & MARIGO [43] and MIEHE *et al.* [76]. While the implicit gradient damage model for quasi-brittle materials requires an additional field problem for the non-local

equivalent strain, the phase field model for brittle fracture introduces an additional field problem for the damage parameter.

Recently, VERHOUSEL & DE BORST [117] proposed a phase field model for cohesive fracture. They introduced a second additional auxiliary field in the smeared crack zone which describes the displacement jump.

However, issues were encountered for the phase field models for brittle and cohesive fracture by MAY *et al.* [71] and VIGNOLLET *et al.* [118]. It was demonstrated by MAY *et al.* [71] that Γ -convergence is not attained numerically for the phase field model for brittle fracture and that the phase field model for cohesive fracture does not pass a two-dimensional patch test for unstructured meshes.

Tracing an equilibrium path for discrete and smeared approaches to fracture often requires an arc-length method due to the occurrence of snap-through and / or snap-back phenomena. For these physically non-linear problems, the arc-length methods developed by RIKS [90] and WEMPNER [123] may fail to converge due to strain localisation in small zones, see CRISFIELD [26]. To overcome this, DE BORST [28] developed the indirect displacement method. This method considers only degrees of freedom involved in the failure zone and therefore requires an a priori knowledge of the expected failure zone.

The fact that physically non-linear problems can involve a monotonically increasing dissipation has been exploited by GUTIÉRREZ [49] and applied to the implicit gradient damage model for quasi-brittle materials. VERHOUSEL *et al.* [114] further pursued and enhanced the idea and applied it to plasticity models and geometrically non-linear problems with damage.

MAY *et al.* [73] proposed an arc-length method which switches between controlling the amount of internal and dissipated energy and hence, is completely energy-based. This path-following method only requires two parameters for tracing an entire equilibrium path and is especially useful when the equilibrium path exhibits several snap-through and / or snap-back phenomena.

B-splines, Non-Uniform Rational B-Splines (NURBS) and T-splines have been exploited for discrete and smeared approaches to fracture by VERHOUSEL *et al.* [115, 116]. The concept of using functions that are used in Computer Aided Design (CAD) also for the analysis was introduced by CIRAK *et al.* [24] and KAGAN *et al.* [55], and termed later isogeometric analysis (IGA) by HUGHES *et al.* [52]. It is motivated from the time-consuming mesh generation and geometric inaccuracy present in most of today's analysis tools. NURBS and T-splines meet a growing acceptance in the engineering community.

The technique of BÉZIER extraction by BORDEN *et al.* [13] and SCOTT *et al.* [94] contributes considerably to this acceptance as it allows for an implementation that is identical to that typically used in finite element codes. However, BUFFA *et al.* [20] raised the concern that not all T-spline blending functions are linearly independent, which is a necessary condition to perform the analysis. LI *et al.* [66] introduced a method to determine whether the blending functions in a T-spline mesh are linearly independent – and therefore analysis-suitable – or not. Later, SCOTT *et al.* [96] developed an algorithm to refine analysis-suitable T-spline meshes.

While the approach in LI *et al.* [66] and SCOTT *et al.* [96] is based on the topology of the T-spline mesh, MAY *et al.* [72] demonstrated that the BÉZIER extraction operator can also be used to determine linear dependencies, partition of unity and nesting behaviour of locally refined meshes. Moreover, the BÉZIER extraction operator of unstructured T-splines with extraordinary points can be modified such that the resulting mesh fulfils the partition of unity and is \mathcal{C}_A^1 -continuous around an extraordinary point, cf. MAY *et al.* [74].

Another spline technology are the POWELL–SABIN [88] B-splines. Quadratic POWELL–SABIN B-splines are based on a linear triangulation and \mathcal{C}_A^1 -continuous in the entire domain. Since they provide basis functions of higher continuity, they can be applied to problems involving fourth order partial differential equations: KIRCHHOFF–LOVE [60, 70] plate theory or the implicit fourth order gradient damage model for quasi-brittle materials. POWELL–SABIN B-splines further allow a direct computation of stresses at the crack tip for discrete fracture approaches as a result of their \mathcal{C}_A^1 -continuity, see also MAY *et al.* [75]. Since triangles are usually favoured for discrete approaches due to their re-meshing flexibility, POWELL–SABIN B-splines are also appealing for these kind of problems.

This thesis comprises different aspects related to numerical fracture mechanics and spline technology for analysis. Each of the following chapters interacts to some extent with the other chapters but can mostly be read independently. **Chapter 2** introduces a new arc-length method that utilises the rate of internal and dissipated energy. This chapter is based on MAY *et al.* [73]. **Chapter 3** is about phase field models for brittle and cohesive fracture. It addresses the issues observed by MAY *et al.* [71] and VIGNOLLET *et al.* [118]: Γ -convergence for the phase field model for brittle fracture and stress oscillations for the phase field model for cohesive fracture. **Chapter 4** shows how the BÉZIER extraction operator for T-splines of arbitrary degree can be used for the determination of linear dependencies, partition of unity property, nesting behaviour and local refinement as demonstrated in MAY *et al.* [72]. Furthermore, unstructured T-spline meshes will be discussed

and it will be explained how the BÉZIER extraction operator can be modified in order to obtain blending functions that fulfil the partition of unity and are \mathcal{C}_A^1 -continuous around an extraordinary point, see also MAY *et al.* [74]. **Chapter 5** introduces BÉZIER extraction for POWELL–SABIN B-splines. Also, it compares different spline technologies and points out their advantages and disadvantages when applied to KIRCHHOFF–LOVE plate theory on a disc as shown in MAY *et al.* [74]. **Chapter 6** focuses on POWELL–SABIN B-splines for smeared and discrete approaches to fracture, cf. MAY *et al.* [75]. Due to the concerns experienced in Chapter 3, POWELL–SABIN B-splines will not be utilised for a phase field approach. Finally, **Chapter 7** gives a concise summary of this thesis.

2 Energy-based arc-length control for physically non-linear problems

This chapter focuses on the energy-based arc-length control for physically non-linear problems proposed by MAY *et al.* [73]. It is based on the rate of internal and dissipated energy and allows to trace an equilibrium path with multiple snap-through and / or snap-back phenomena. After addressing briefly the need for an arc-length method, an arc-length control which is based on the rate of dissipated and internal energy is introduced. The dissipation-based arc-length control is derived from the second law of thermodynamics whereas the internal energy-based arc-length control is derived from the time derivative of the energy density. The method is applied to a plate with an eccentric hole using the phase field model for brittle fracture and to a perforated beam using interface elements with decohesion. Index notation is adopted throughout with respect to a CARTESIAN frame.

2.1 Necessity of an arc-length control

The state of a solid is governed by the equilibrium of external and internal forces

$$\underline{\mathbf{f}}^{\text{int}}(\underline{\mathbf{u}}) = \underline{\mathbf{f}}^{\text{ext}}, \quad (2.1)$$

while the external force vector can be represented by a normalised load vector $\hat{\underline{\mathbf{f}}}$ and the loading parameter λ

$$\underline{\mathbf{f}}^{\text{ext}} = \lambda \hat{\underline{\mathbf{f}}}. \quad (2.2)$$

As it will be shown next, neither a force nor a displacement control is in general suitable for following the equilibrium path. During a force control, the loading parameter λ is prescribed, while for a displacement control, the displacement u is prescribed for some points of the solid. Fig. 2.1 gives a typical force-displacement curve which can occur during loading.

Up to point A in Fig. 2.1(a) a monotonically increasing force control could be employed,

$$\lambda_1 < \lambda_2 < \dots < \lambda_k, \quad (2.3)$$

whereas up to point B in Fig. 2.1(b) a monotonically increasing displacement control

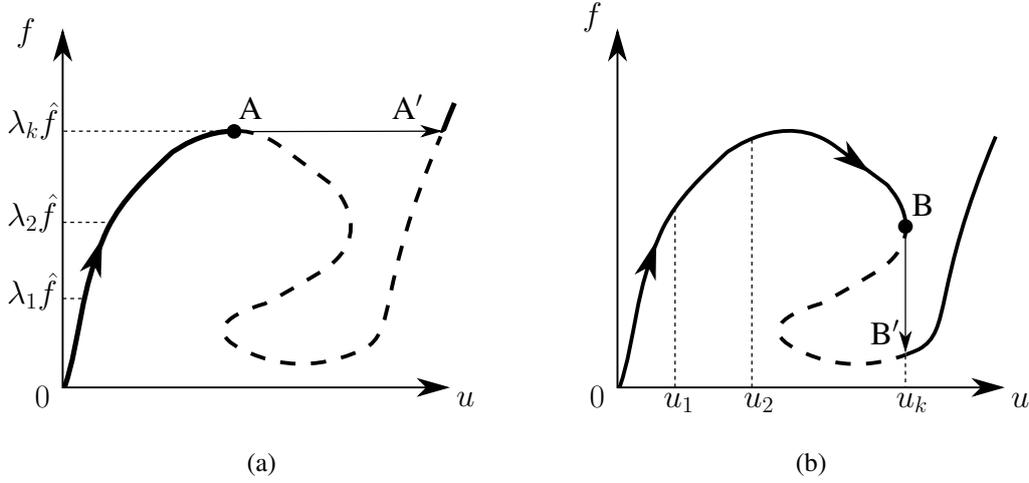


Fig. 2.1: (a) The force control is not able to capture the dashed equilibrium path between the points A and A' (snap-through). (b) The displacement control is unable to trace the dashed equilibrium path between the points B and B' (snap-back).

could be used,

$$u_1 < u_2 < \dots < u_k. \quad (2.4)$$

However, the monotonically increasing force control is not able to capture the snap-through after the peak load at point A, while the monotonically increasing displacement control is unable to capture the snap-back at point B.

Therefore, an arc-length control is necessary in order to properly trace the equilibrium path during loading. An arc-length control adds to Eq. (2.1) an additional constraint equation $\varphi = \varphi(\underline{\mathbf{u}}, \lambda)$, which ensures that the equilibrium path can be followed. By adding the additional constraint equation, the following system of equations must be solved

$$\underline{\mathbf{H}}(\underline{\mathbf{u}}, \lambda) = \begin{bmatrix} \underline{\mathbf{f}}^{\text{int}}(\underline{\mathbf{u}}) - \lambda \hat{\underline{\mathbf{f}}} \\ \varphi(\underline{\mathbf{u}}, \lambda) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}. \quad (2.5)$$

Assuming that the solution in the $k+1$ -th increment and i -th iteration is known for $\underline{\mathbf{u}}_{k+1}^i$ and λ_{k+1}^i , Eq. (2.5) can be linearised using a TAYLOR series around $\underline{\mathbf{u}}_{k+1}^i, \lambda_{k+1}^i$ as follows

$$\underline{\mathbf{H}}(\underline{\mathbf{u}}, \lambda) \approx \underline{\mathbf{H}}(\underline{\mathbf{u}}_{k+1}^i, \lambda_{k+1}^i) + \underline{\mathbf{K}}_T(\underline{\mathbf{u}}_{k+1}^i, \lambda_{k+1}^i) \cdot \begin{bmatrix} \underline{\mathbf{u}} - \underline{\mathbf{u}}_{k+1}^i \\ \lambda - \lambda_{k+1}^i \end{bmatrix} = \mathbf{0}, \quad (2.6)$$

with the tangential stiffness matrix

$$\underline{\underline{\mathbf{K}}}_T(\underline{\mathbf{u}}, \lambda) = \begin{bmatrix} \frac{\partial \underline{\mathbf{f}}^{\text{int}}(\underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}} & -\underline{\hat{\mathbf{f}}} \\ \frac{\partial \varphi(\underline{\mathbf{u}}, \lambda)}{\partial \underline{\mathbf{u}}} & \frac{\partial \varphi(\underline{\mathbf{u}}, \lambda)}{\partial \lambda} \end{bmatrix} = \begin{bmatrix} \underline{\underline{\mathbf{K}}} & -\underline{\hat{\mathbf{f}}} \\ \underline{\mathbf{v}}^T & w \end{bmatrix}. \quad (2.7)$$

The solution for $\underline{\mathbf{u}}_{k+1}^{i+1}$, λ_{k+1}^{i+1} in the $k+1$ -th increment in the $i+1$ -th iteration in Eq. (2.6),

$$\underline{\underline{\mathbf{H}}}(\underline{\mathbf{u}}_{k+1}^{i+1}, \lambda_{k+1}^{i+1}) = \underline{\mathbf{0}}, \quad (2.8)$$

can be obtained by solving

$$\begin{bmatrix} \underline{\mathbf{u}} \\ \lambda \end{bmatrix}_{k+1}^{i+1} = \begin{bmatrix} \underline{\mathbf{u}} \\ \lambda \end{bmatrix}_{k+1}^i - \underline{\underline{\mathbf{K}}}_T^{-1} \Big|_{k+1}^i \cdot \begin{bmatrix} \underline{\mathbf{f}}^{\text{int}}(\underline{\mathbf{u}}) - \lambda \underline{\hat{\mathbf{f}}} \\ \varphi(\underline{\mathbf{u}}, \lambda) \end{bmatrix}_{k+1}^i. \quad (2.9)$$

In order to save computational time for the solution of Eq. (2.9), the inverse of the tangential stiffness matrix $\underline{\underline{\mathbf{K}}}_T$ can be evaluated utilising the SHERMAN–MORRISON [101] formula, see Appendix A.1.

2.2 Arc-length control based on the rate of dissipated energy

An arc-length function based on the rate of dissipated energy has been introduced for damage models by GUTIÉRREZ [49]. The procedure uses a force control at the beginning of the loading, and when the dissipated energy reaches a certain limit switches to a dissipation-based arc-length control. The dissipation-based arc-length control is motivated by the fact that during loading the amount of dissipated energy can only increase monotonically. Therefore, by prescribing the amount of energy which should be dissipated in each loading step, the equilibrium path can be traced.

The first law of thermodynamics gives a statement about the conservation of energy – energy can neither be destroyed nor created. However, the first law of thermodynamics does not give a statement about the dissipative nature of a process. The dissipative behaviour of a process can be described by the second law of thermodynamics. The second law of thermodynamics reads in a *local* form for a constant temperature (see JIRÁSEK & BAŽANT [54, Chapter 23]),

$$\dot{\mathcal{D}} = \sigma_{ij} \dot{\epsilon}_{ij} - \dot{\psi} \geq 0, \quad (2.10)$$

where $\dot{\mathcal{D}}$ is the rate of dissipation and ψ the energy density that describes the energy stored in the bulk of the solid per unit volume. It is now assumed that for the constitutive

behaviour between the stress σ_{ij} and the strain ε_{ij} a damage law of the form

$$\sigma_{ij}(\varepsilon_{ij}, d) = g(d)C_{ijkl}\varepsilon_{kl} \quad (2.11)$$

is used with the damage parameter $d \in [0, 1]$ (0: undamaged state, 1: fully broken state), the degradation function $g(d)$, the infinitesimal strain tensor

$$\varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}) \quad (2.12)$$

and the elasticity tensor equipped with the usual major and minor symmetries

$$C_{ijkl} = C_{jikl}, \quad C_{ijkl} = C_{ijlk}, \quad C_{ijkl} = C_{klij}. \quad (2.13)$$

The energy density ψ reads

$$\psi(\varepsilon_{ij}, d) = \frac{1}{2}\sigma_{ij}(\varepsilon_{ij}, d)\varepsilon_{ij}, \quad (2.14)$$

while

$$\begin{aligned} \frac{\partial\psi}{\partial\varepsilon_{kl}} &= \frac{1}{2}\frac{\partial\sigma_{ij}}{\partial\varepsilon_{kl}}\varepsilon_{ij} + \frac{1}{2}\sigma_{ij}\frac{\partial\varepsilon_{ij}}{\partial\varepsilon_{kl}} = \frac{1}{2}g(d)C_{ijkl}\varepsilon_{ij} + \frac{1}{2}g(d)C_{ijkl}\varepsilon_{kl}\delta_{ik}\delta_{jl} \\ &= g(d)C_{ijkl}\varepsilon_{ij} = g(d)C_{klij}\varepsilon_{ij} = \sigma_{kl}. \end{aligned} \quad (2.15)$$

There are two ways of expressing the time derivative of the energy density ψ in Eq. (2.10). The first option is to use the chain rule with Eq. (2.15)

$$\dot{\psi} = \frac{\partial\psi}{\partial\varepsilon_{ij}}\dot{\varepsilon}_{ij} + \frac{\partial\psi}{\partial d}\dot{d} = \sigma_{ij}\dot{\varepsilon}_{ij} + \frac{\partial\psi}{\partial d}\dot{d}, \quad (2.16)$$

which yields for Eq. (2.10)

$$\dot{\mathcal{D}} = -\frac{\partial\psi}{\partial d}\dot{d} \geq 0. \quad (2.17)$$

The second option is to apply the product rule

$$\dot{\psi} = \frac{1}{2}\dot{\sigma}_{ij}\varepsilon_{ij} + \frac{1}{2}\sigma_{ij}\dot{\varepsilon}_{ij}, \quad (2.18)$$

which results for Eq. (2.10) in

$$\dot{\mathcal{D}} = \frac{1}{2}\sigma_{ij}\dot{\varepsilon}_{ij} - \frac{1}{2}\dot{\sigma}_{ij}\varepsilon_{ij} \geq 0. \quad (2.19)$$

Assuming that there are no discontinuities in the solid, the *global* forms of Eq. (2.17) and Eq. (2.19) can be written as

$$\dot{\mathcal{E}}^D = \int_{\Omega} \dot{\mathcal{D}} \, dV = \int_{\Omega} \frac{1}{2} \sigma_{ij} \dot{\varepsilon}_{ij} - \frac{1}{2} \dot{\sigma}_{ij} \varepsilon_{ij} \, dV = \int_{\Omega} -\frac{\partial \psi}{\partial d} \dot{d} \, dV, \quad (2.20)$$

where $\dot{\mathcal{E}}^D$ is the rate of dissipated energy. It can be observed from Eq. (2.20) that $\dot{\mathcal{E}}^D$ directly follows from the evolution of the damage variable d . The dissipated energy \mathcal{E}^D increases monotonically since $\dot{\mathcal{E}}^D \geq 0$ follows from $\dot{d} \geq 0$ and $\frac{\partial \psi}{\partial d} \leq 0$ in Eq. (2.20).

The second integral in Eq. (2.20) can be expressed in matrix-vector format with

$$\underline{\varepsilon} = \underline{\mathbf{B}} \underline{\mathbf{u}}, \quad (2.21)$$

Eq. (2.1) and Eq. (2.2) as

$$\begin{aligned} \dot{\mathcal{E}}^D &= \int_{\Omega} \frac{1}{2} \dot{\underline{\mathbf{u}}}^T \underline{\mathbf{B}}^T \underline{\boldsymbol{\sigma}} \, dV - \int_{\Omega} \frac{1}{2} \underline{\mathbf{u}}^T \underline{\mathbf{B}}^T \dot{\underline{\boldsymbol{\sigma}}} \, dV \\ &= \frac{1}{2} \dot{\underline{\mathbf{u}}} \underline{\mathbf{f}}^{\text{int}}(\underline{\mathbf{u}}) - \frac{1}{2} \underline{\mathbf{u}} \dot{\underline{\mathbf{f}}}^{\text{int}}(\underline{\mathbf{u}}) = \frac{1}{2} \dot{\underline{\mathbf{u}}} \lambda \hat{\underline{\mathbf{f}}} - \frac{1}{2} \underline{\mathbf{u}} \dot{\lambda} \hat{\underline{\mathbf{f}}}. \end{aligned} \quad (2.22)$$

Replacing $\dot{\mathcal{E}}^D$ in Eq. (2.22) with the rate of the path parameter $\dot{\tau}^D$ yields

$$\frac{1}{2} (\lambda \dot{\underline{\mathbf{u}}}^T - \dot{\lambda} \underline{\mathbf{u}}^T) \hat{\underline{\mathbf{f}}} - \dot{\tau}^D = 0. \quad (2.23)$$

Any time discretisation scheme would result in

$$\frac{1}{2} (\lambda_k \underline{\mathbf{u}}_{k+1}^T - \lambda_{k+1} \underline{\mathbf{u}}_k^T) \hat{\underline{\mathbf{f}}} - \Delta \tau^D = 0, \quad (2.24)$$

see Appendix A.2.2. The time discretisation of the last term in Eq. (2.20) is in general not equal to Eq. (2.24), although the second and the third integral in Eq. (2.20) are equal from a continuity perspective. Eq. (2.24) can now be used as the constraint equation in Eq. (2.5) in the following form

$$\varphi^D(\underline{\mathbf{u}}, \lambda) = \frac{1}{2} (\lambda_k \underline{\mathbf{u}}^T - \lambda \underline{\mathbf{u}}_k^T) \hat{\underline{\mathbf{f}}} - \Delta \tau^D. \quad (2.25)$$

The parameter $\Delta \tau^D$ in Eq. (2.25) can be interpreted as the prescribed step size for each increment – it prescribes the amount of energy which needs to be dissipated in one increment.

2.3 Arc-length control based on the rate of internal energy

Next, a new arc-length function will be introduced for the regime when the rate of dissipated energy $\dot{\mathcal{E}}^D$ due to the evolution of the damage variable d is very small, e. g., at the onset of loading. Assuming that there are no discontinuities in the solid, and using Eqs. (2.1), (2.2) and (2.21), Eq. (2.18) can be written in the *global* form to yield the rate of the internal energy $\dot{\mathcal{U}}$ in matrix-vector format

$$\begin{aligned} \dot{\mathcal{U}} &= \int_{\Omega} \dot{\psi} \, dV = \int_{\Omega} \frac{1}{2} \dot{\underline{\mathbf{u}}}^T \underline{\mathbf{B}}^T \underline{\boldsymbol{\sigma}} + \frac{1}{2} \underline{\mathbf{u}}^T \underline{\mathbf{B}}^T \dot{\underline{\boldsymbol{\sigma}}} \, dV \\ &= \frac{1}{2} \dot{\underline{\mathbf{u}}}^T \underline{\mathbf{f}}^{\text{int}}(\underline{\mathbf{u}}) + \underline{\mathbf{u}}^T \underline{\mathbf{f}}^{\text{int}}(\underline{\mathbf{u}}) = \frac{1}{2} \left(\dot{\underline{\mathbf{u}}}^T \lambda + \underline{\mathbf{u}}^T \dot{\lambda} \right) \underline{\hat{\mathbf{f}}}. \end{aligned} \quad (2.26)$$

Replacing $\dot{\mathcal{U}}$ with the path parameter $\dot{\tau}^U$ in Eq. (2.19) and applying the midpoint rule (see Appendix A.2.1) results in

$$\frac{1}{2} \left(\lambda_{k+1} \underline{\mathbf{u}}_{k+1}^T - \lambda_k \underline{\mathbf{u}}_k^T \right) \underline{\hat{\mathbf{f}}} - \Delta\tau^U = 0, \quad (2.27)$$

which can now be used as a constraint equation in Eq. (2.5) as follows

$$\varphi^U(\underline{\mathbf{u}}, \lambda) = \frac{1}{2} \left(\lambda \underline{\mathbf{u}}^T - \lambda_k \underline{\mathbf{u}}_k^T \right) \underline{\hat{\mathbf{f}}} - \Delta\tau^U. \quad (2.28)$$

The parameter $\Delta\tau^U$ in Eq. (2.28) can be interpreted as the prescribed step size for an increment – it prescribes the amount of internal energy which needs to be introduced into the system in one increment.

In the first iteration ($i = 1$) of the first increment ($k = 1$)

$$\left. \frac{\partial \varphi^U(\underline{\mathbf{u}}, \lambda)}{\partial \lambda} \right|_{k=1}^{i=1} = \left. \frac{1}{2} \underline{\mathbf{u}}^T \underline{\hat{\mathbf{f}}} \right|_{k=1}^{i=1} = \frac{1}{2} \underline{\mathbf{u}}_1^T \underline{\hat{\mathbf{f}}} = 0, \quad (2.29)$$

and Eq. (2.7) would result with $\underline{\mathbf{u}}_1^1 = \underline{\mathbf{u}}_0 = \underline{\mathbf{0}}$ in a singular matrix. Hence, in the first increment $k = 1$ the following arc-length expression is used

$$\varphi_1^F(\lambda) = \lambda - \Delta\tau_1^F, \quad \left. \frac{\partial \varphi_1^F(\lambda)}{\partial \lambda} \right|_{k=1} = 1, \quad (2.30)$$

which is equivalent to a force control. After the first increment, the solution for $\underline{\mathbf{u}}_1$ and λ_1 is known. From the solution for $\underline{\mathbf{u}}_1$ and λ_1 , the rate of the internal energy for the first increment $\Delta\tau_1^U$ can be evaluated using Eq. (2.28),

$$\Delta\tau_1^U = \frac{1}{2} \left(\lambda_1 \underline{\mathbf{u}}_1^T - \lambda_0 \underline{\mathbf{u}}_0^T \right) \underline{\hat{\mathbf{f}}} = \frac{1}{2} \lambda_1 \underline{\mathbf{u}}_1^T \underline{\hat{\mathbf{f}}}. \quad (2.31)$$

$\Delta\tau_1^U$ from Eq. (2.31) can then be utilised in the following increments as the prescribed step size. No adaptive step size scheme will be used for $\Delta\tau_1^U$.

Next to $\Delta\tau_1^F$, a ratio a needs to be defined. This parameter specifies when the arc-length control has to switch from internal energy-based arc-length control to dissipation-based arc-length control and is defined as

$$a = \frac{\Delta\tau^D}{\Delta\tau^U}. \quad (2.32)$$

When the force-displacement curve exhibits a more brittle behaviour – i. e. little damage occurs before the maximum peak force – a smaller value must be assigned to the parameter a . The simulation cannot switch to the dissipation-based arc-length control and will not find an equilibrium at a snap-through / snap-back when the parameter a is too large. If a is taken too small the simulation switches too early to the dissipation-based arc-length control. In order to determine a , one can start a simulation assuming a large a . Then, a value can be assigned to a that is smaller than the term $\frac{\Delta\tau^D}{\Delta\tau^U}$ from the last increment where an equilibrium could be found. The algorithm is summarised in Algorithm 2.1.

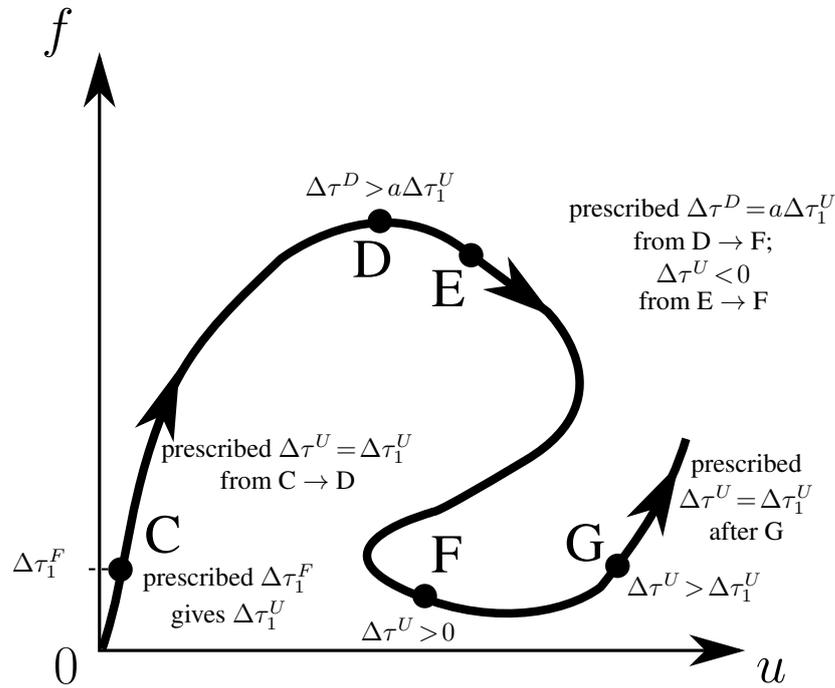


Fig. 2.2: Path following technique using an arc-length control which is based on the rate of internal energy \dot{U} and the rate of dissipated energy $\dot{\mathcal{E}}^D$. Switch from rate of internal energy-based arc-length control φ^U to dissipation-based arc-length control φ^D at point D; switch from dissipation-based arc-length control φ^D to internal energy-based arc-length control φ^U at point G.

```

k = 0;
InternalEnergyArclength=1;
Prescribe  $\Delta\tau_1^F$  and ratio a;
while k < kmax do
    k = k + 1; i = 0; error = 1;
    while error > errormax do
        i = i + 1;
        if k = 1 then
             $\varphi_1^F = \lambda_1 - \Delta\tau_1^F$ ;
        else
            if InternalEnergyArclength = 1 then
                 $\varphi^U = \frac{1}{2} (\lambda_{k+1}^i \underline{\mathbf{u}}_{k+1}^i - \lambda_k \underline{\mathbf{u}}_k^T) \hat{\underline{\mathbf{f}}} - \Delta\tau^U$ ;
            else
                 $\varphi^D = \frac{1}{2} (\lambda_k \underline{\mathbf{u}}_{k+1}^i - \lambda_{k+1}^i \underline{\mathbf{u}}_k^T) \hat{\underline{\mathbf{f}}} - \Delta\tau^D$ ;
            end
        end
        // Solve for  $\underline{\mathbf{u}}_{k+1}^{i+1}$  and  $\lambda_{k+1}^{i+1}$  using Eq. (2.9) and evaluate the error
        error = error( $\underline{\mathbf{u}}_{k+1}^{i+1}, \lambda_{k+1}^{i+1}$ );
    end
    // Define the arc-length function for the next increment
    if k = 1 then
         $\Delta\tau_1^U = \frac{1}{2} \lambda_1 \underline{\mathbf{u}}_1^T \hat{\underline{\mathbf{f}}}$ ;
         $\Delta\tau^U = \Delta\tau_1^U$ ;
    else
        if InternalEnergyArclength = 1 then
             $\Delta\tau^D = \frac{1}{2} (\lambda_k \underline{\mathbf{u}}_{k+1}^T - \lambda_{k+1} \underline{\mathbf{u}}_k^T) \hat{\underline{\mathbf{f}}}$ ;
            if  $\Delta\tau^D > a\Delta\tau_1^U$  then
                // use now arc-length based on rate of dissipated energy
                InternalEnergyArclength = 0;
                InternalEnergyNegative = 0;
                 $\Delta\tau^D = a\Delta\tau_1^U$ ;
            end
        else
             $\Delta\tau^U = \frac{1}{2} (\lambda_{k+1} \underline{\mathbf{u}}_{k+1}^T - \lambda_k \underline{\mathbf{u}}_k^T) \hat{\underline{\mathbf{f}}}$ ;
            if  $\Delta\tau^U < 0$  and InternalEnergyNegative = 0 then
                InternalEnergyNegative = 1;
            else if  $\Delta\tau^U > \Delta\tau_1^U$  and InternalEnergyNegative = 1 then
                // use now arc-length based on rate of internal energy
                InternalEnergyArclength = 1;
                 $\Delta\tau^U = \Delta\tau_1^U$ ;
            end
        end
    end
end
end
end

```

Algorithm 2.1: Algorithm for the loading process for the arc-length control based on the rate of internal energy \dot{U} and the rate of the dissipated energy $\dot{\mathcal{E}}^D$.

A simulation starts with a prescribed step size $\Delta\tau_1^F$ for the force, which gives after convergence the step size $\Delta\tau_1^U$ for the rate of internal energy at point C, cf. Fig. 2.2. $\Delta\tau_1^U$ is then used as the prescribed step size for the arc-length control φ^U based on the rate of internal

energy from point C to D, $\Delta\tau^U = \Delta\tau_1^U$. If, at the end of an increment, the incremental dissipated energy times the ratio factor a is larger than the prescribed increment for the internal energy, $\Delta\tau^D > a\Delta\tau^U = a\Delta\tau_1^U$ (point D in Fig. 2.2), the loading process switches from internal energy-based (φ^U) to dissipation-based (φ^D) arc-length control with a prescribed step size $\Delta\tau^D = a\Delta\tau_1^U$. The increment in the internal energy becomes negative, $\Delta\tau^U < 0$, between point E and F in Fig. 2.2. When the incremental internal energy $\Delta\tau^U$ becomes again larger than $\Delta\tau_1^U$ (point G in Fig. 2.2), the loading process switches back to an internal energy-based arc-length control φ^U with a prescribed step size $\Delta\tau^U = \Delta\tau_1^U$. The arc-length method requires two parameters: $\Delta\tau_1^F$ and a .

2.4 Numerical examples

In this section, two numerical examples are considered which exhibit multiple snap-through and / or snap-back phenomena.

2.4.1 Phase field model for brittle fracture

Consider the phase field problem for brittle fracture governed by the equations

$$\sigma_{ij,i} = 0, \quad (2.33)$$

$$\frac{\mathcal{G}_c}{2\ell}[d - 4\ell^2 d_{,ii}] + \frac{\partial g}{\partial d}\mathcal{H} = 0 \quad (2.34)$$

and subject to the boundary conditions

$$\sigma_{ij}n_j = h_i \quad \text{on } \partial\Omega_h, \quad u_i = \bar{u}_i \quad \text{on } \partial\Omega_u, \quad d_{,i}n_i = 0 \quad \text{on } \partial\Omega, \quad (2.35)$$

with the decomposition of the boundary $\partial\Omega$ into the parts $\partial\Omega_h$ and $\partial\Omega_u$ ($\partial\Omega_h \cap \partial\Omega_u = \emptyset$, $\partial\Omega_h \cup \partial\Omega_u = \partial\Omega$), the prescribed surface traction \mathbf{h} , prescribed displacement $\bar{\mathbf{u}}$ and normal vector \mathbf{n} on the boundary, see also Chapter 3 for more details. The stress tensor is computed via

$$\sigma_{ij} = g(d)C_{ijkl}\varepsilon_{kl}, \quad (2.36)$$

with the degradation function

$$g(d) = (1 - d)^2. \quad (2.37)$$

In Eq. (2.34), ℓ denotes the length scale parameter, \mathcal{G}_c the critical energy release rate and \mathcal{H} the history field

$$\mathcal{H} = \max \psi, \quad (2.38)$$

which was introduced by MIEHE *et al.* [76] in order to ensure irreversibility of the phase field variable d . The energy density ψ for the damaged solid is computed with

$$\psi = g(d)\psi^{\text{el}}, \quad (2.39)$$

where ψ^{el} corresponds to the energy density of the undamaged solid

$$\psi^{\text{el}} = \frac{1}{2}\lambda\varepsilon_{ii}^{\text{el}}\varepsilon_{jj}^{\text{el}} + \mu\varepsilon_{ij}^{\text{el}}\varepsilon_{ij}^{\text{el}} = \frac{1}{2}\lambda\varepsilon_{ii}\varepsilon_{jj} + \mu\varepsilon_{ij}\varepsilon_{ij}, \quad (2.40)$$

with the LAMÉ constants λ and μ . Energy is dissipated upon creating a new crack surface area

$$\mathcal{E}^D = \int_{\Omega} \mathcal{G}_c \, dA, \quad (2.41)$$

which becomes due to the smearing

$$\mathcal{E}^D = \int_{\Omega} \mathcal{G}_c \gamma_{\ell} \, dV, \quad (2.42)$$

with the crack surface density function

$$\gamma_{\ell} = \frac{1}{4\ell} (d^2 + 4\ell^2 d_{,i} d_{,i}). \quad (2.43)$$

Recalling Eq. (2.20) gives for the rate of dissipated energy

$$\dot{\mathcal{E}}^D = \int_{\Omega} -\frac{\partial\psi}{\partial d} \dot{d} \, dV = \int_{\Omega} \mathcal{G}_c \dot{\gamma}_{\ell} \, dV, \quad (2.44)$$

i. e. the energy which is dissipated in the bulk is equal to the energy dissipated upon propagation of the smeared crack surface.

The phase field problem for brittle fracture is applied to the plate with an eccentric hole in Fig. 2.3 which has been considered with different dimensions in LORENTZ & BADEL [69]. The mesh consists of 9494 linear quads which gives the mesh size $h \approx 0.01$ mm. Plane strain is assumed. The material parameters are taken as: YOUNG's modulus $E = 210$ MPa, POISSON's ratio $\nu = 0.3$, critical energy release rate $\mathcal{G}_c = 2.7 \cdot 10^{-3}$ N/mm, and the length scale parameter is set to $\ell = 0.02$ mm. The two parameters for the arc-length control are $\Delta\tau_1^F = 0.2$ N and $a = 0.25$. The force-displacement curve and the development of the phase field variable d are depicted in Figs. 2.4 and 2.5.

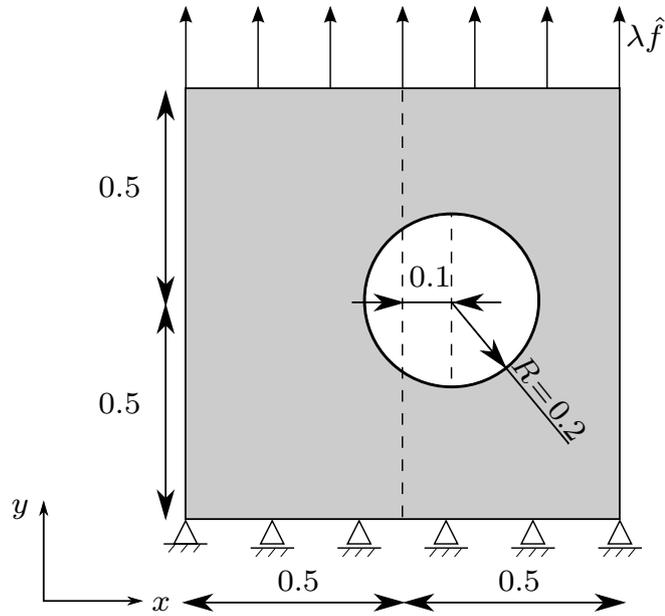


Fig. 2.3: Plate under tension with an eccentric hole, dimensions in mm.

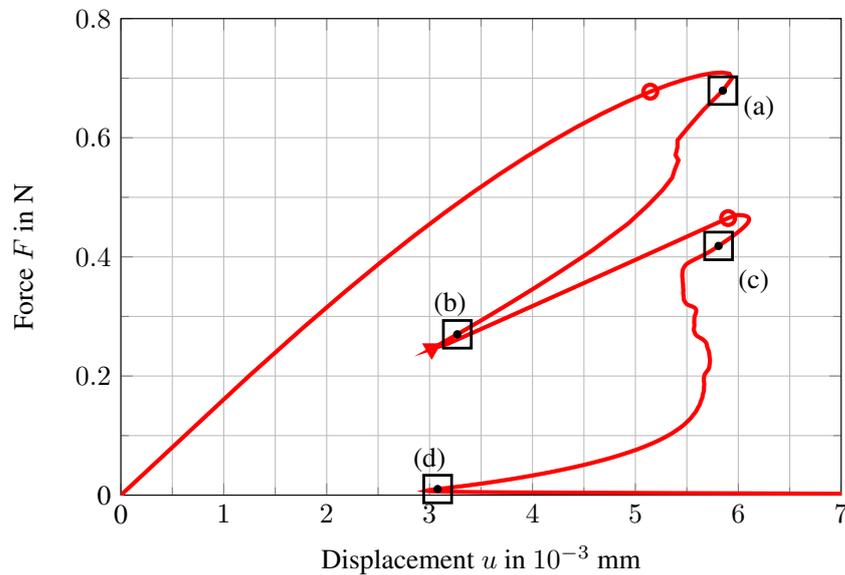


Fig. 2.4: Force-displacement curve for the plate under tension with an eccentric hole for a mesh size $h \approx 0.01$ mm and a length scale parameter $\ell = 0.02$ mm. Circles denote the switch from internal energy to dissipation-based arc-length control, the triangle denotes the switch from dissipation-based to internal energy-based arc-length control. Squares correspond to the phase field distributions for d in Fig. 2.5.

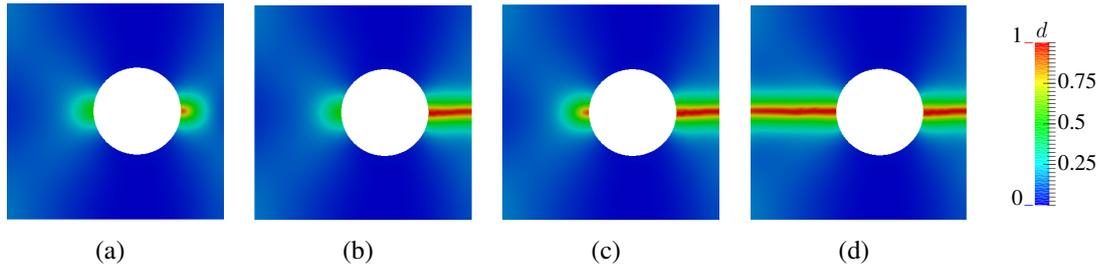


Fig. 2.5: Propagation of the phase field variable d for the plate with an eccentric hole under tension; plots correspond to the squares in Fig. 2.4.

2.4.2 Perforated beam with interface elements

As a second numerical example, the perforated beam shown in Fig. 2.6(a) is considered, see also VERHOUSEL *et al.* [114].

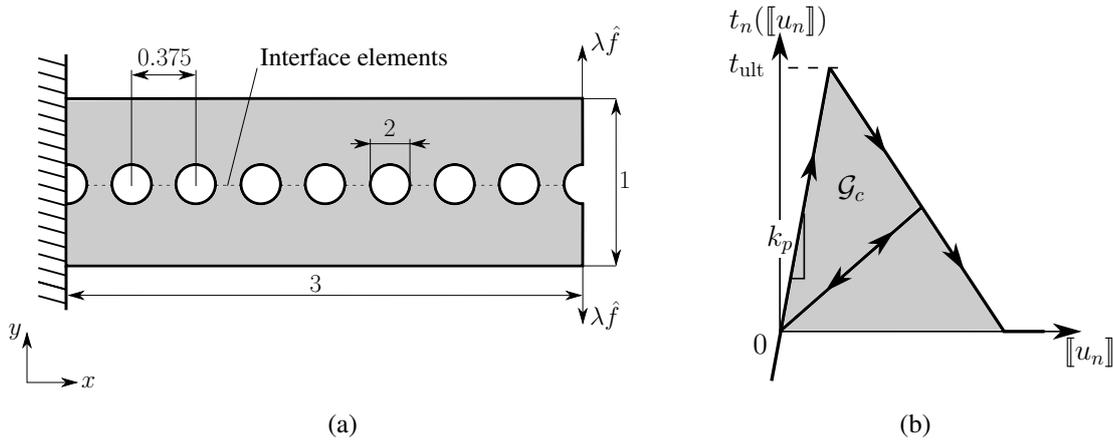


Fig. 2.6: (a) Set-up for the perforated beam, dimensions in mm. (b) Bi-linear cohesive law for the interface elements. The shaded grey area is equivalent to the critical energy release rate \mathcal{G}_c .

The stress equilibrium

$$\sigma_{ij,i} = 0 \quad (2.45)$$

is subject to the boundary conditions

$$\sigma_{ij}n_j = h_i \quad \text{on } \partial\Omega_h, \quad u_i = \bar{u}_i \quad \text{on } \partial\Omega_u. \quad (2.46)$$

The boundary $\partial\Omega$ is decomposed into the parts $\partial\Omega_h$ and $\partial\Omega_u$ ($\partial\Omega_h \cap \partial\Omega_u = \emptyset$, $\partial\Omega_h \cup \partial\Omega_u = \partial\Omega$); \mathbf{h} is the prescribed surface traction, $\bar{\mathbf{u}}$ the prescribed displacement and \mathbf{n} the normal vector on the boundary. The bulk is assumed to be linearly elastic, i. e.

no damage law is used

$$\sigma_{ij} = C_{ijkl}\varepsilon_{kl}. \quad (2.47)$$

Along the interface Γ_d the cohesive traction \mathbf{t} is imposed, see also Section 6.2 for more details. The cohesive traction \mathbf{t} corresponds to the global coordinate system and is obtained by transforming the cohesive traction \mathbf{t}_d of the local coordinate system. \mathbf{t}_d consists of two components: one for the normal (t_n) and one for the shear direction (t_s). In the normal direction, the bi-linear cohesive law $t_n(\llbracket u_n \rrbracket)$ of Fig. 2.6(b) with the ultimate traction $t_{\text{ult}} = 1$ MPa, critical energy release rate $\mathcal{G}_c = 2.5 \cdot 10^{-3}$ N/mm and penetration stiffness $k_p = 10^4$ MPa/mm is applied to the interface elements. $\llbracket u_n \rrbracket$ denotes the jump in the normal direction. In the shear direction, t_s is set to zero. Due to the elastic bulk, energy is dissipated only in the interface elements. The increment of dissipated energy $\Delta\tau^D$ can be derived by geometric considerations, see also Fig. 2.7(a)

$$\Delta\tau^D = A - B - C - D = \frac{1}{2} (t_{nk}\llbracket u_n \rrbracket_{k+1} - t_{nk+1}\llbracket u_n \rrbracket_k). \quad (2.48)$$

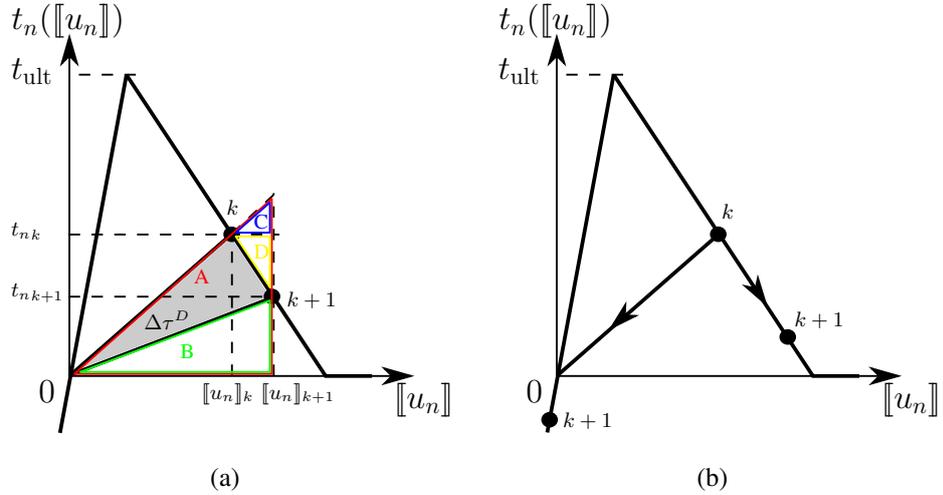


Fig. 2.7: (a) The increment of dissipated energy $\Delta\tau^D$ (shaded grey area) can be derived from geometric considerations. (b) Both options for increment $k+1$ could follow increment k since they both yield a positive increment of dissipated energy $\Delta\tau^D$ in Eq. (2.48).

Eq. (2.48) does not have a unique solution for increment $k+1$, see also Fig. 2.7(b). Still, no problems were encountered for the numerical simulation, i. e. no undesired unloading occurred.

The material parameters are taken as: YOUNG's modulus $E = 100$ MPa, POISSON's ratio $\nu = 0.3$ with a plane strain assumption. 15354 linear triangular elements are used and along the interface a two-point NEWTON-COTES integration scheme is utilised in order

to avoid stress oscillations along the interface, see SCHELLEKENS & DE BORST [91], and further VIGNOLLET *et al.* [119] for a discussion on the integration of interface elements in an isogeometric context. The two parameters for the arc-length control are $\Delta\tau_1^F = 0.025$ N and $a = 0.1$. The resulting force-displacement curve is given in Fig. 2.8.

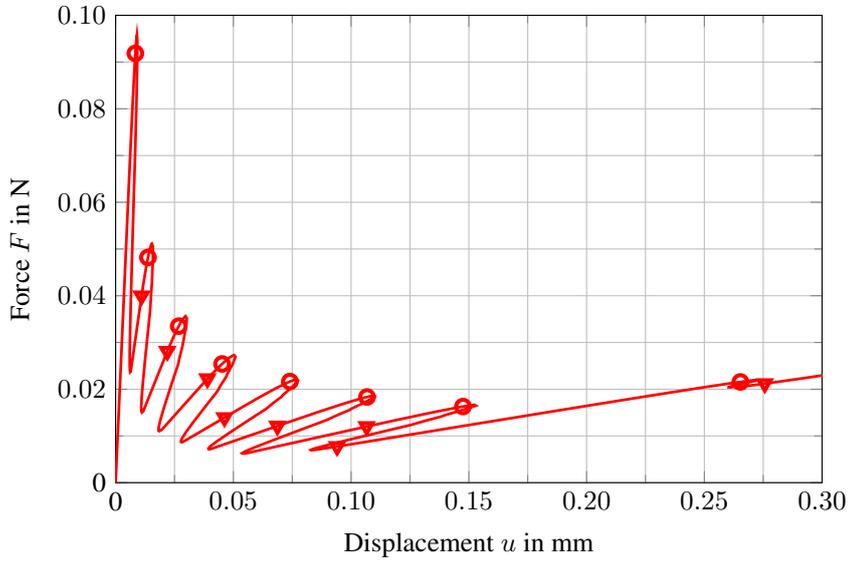


Fig. 2.8: Force-displacement curve for the perforated beam; circles denote the switch from internal energy to dissipation-based arc-length control, triangles denote the switch from dissipation to internal energy-based arc-length control.

3 Fracture models using phase field

The next section explains how a crack can be smeared by introducing a phase field variable. Afterwards, the phase field models for brittle and cohesive fracture are studied.

3.1 Phase field expression of the crack

The basic idea of phase field models for fracture is to approximate a discrete crack Γ by a smeared crack Γ_ℓ . The exponential function

$$d(x) = \exp\left(-\frac{|x|}{2\ell}\right), \quad (3.1)$$

where ℓ denotes the length scale parameter, can be utilised for the approximated crack surface in the one-dimensional case, see Fig. 3.1.

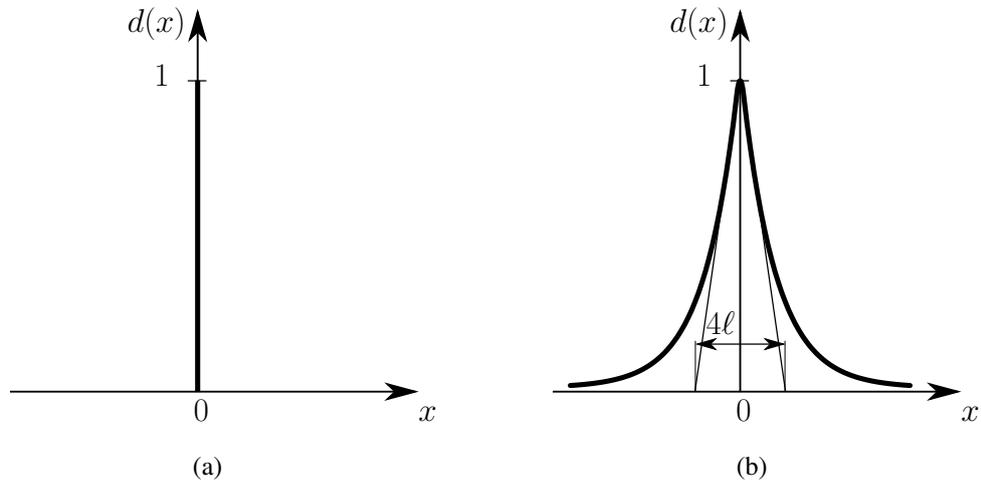


Fig. 3.1: (a) Sharp crack with $d=1$ at $x=0$ and $d=0$ otherwise, (b) smeared crack modelled with the length scale parameter ℓ .

The phase field variable d describes the phase field and varies between zero and one, $d \in [0, 1]$. Herein, $d=0$ characterises the unbroken state of the material far away from the crack while $d=1$ represents the fully broken material inside the crack. The differential equation

$$d - 4\ell^2 d_{,xx} = 0 \quad (3.2)$$

subject to the boundary conditions

$$d(0) = 1, \quad (3.3)$$

$$d(\pm\infty) = 0 \quad (3.4)$$

has the solution in Eq. (3.1) for the one-dimensional case. This can be demonstrated by applying to Eq. (3.2) the ansatz $d = e^{-\lambda|x|}$ (which already satisfies the boundary condition Eq. (3.4)), solving for λ and subsequently using for the resulting problem Eq. (3.3) to determine the constant parameter.

The discrete crack surface Γ can now be expressed by the functional / potential Γ_ℓ

$$\Gamma = \int_{\Gamma} dA = \int_{\Omega} \underbrace{\frac{1}{4\ell} (d^2 + 4\ell^2 d_{,x}^2)}_{\gamma_\ell} dV = \Gamma_\ell, \quad (3.5)$$

with the crack surface density function γ_ℓ , see MIEHE *et al.* [77] for more details. Eq. (3.5) is for the one-dimensional case an equality and not an approximation. From Eq. (3.5) it can be observed that the crack surface Γ – a discontinuity – has been smeared over the whole domain to yield the smeared crack surface Γ_ℓ .

Minimising the crack potential Γ_ℓ (see also REDDY [89, Chapter 4] for an extensive overview of minimising functionals)

$$\delta\Gamma_\ell = \frac{\partial\Gamma_\ell}{\partial d} \delta d + \frac{\partial\Gamma_\ell}{\partial d_{,x}} \delta d_{,x} = 0 \quad (3.6)$$

yields the differential equation in Eq. (3.2) and the boundary condition Eq. (3.4). In the subsequent sections, the boundary condition Eq. (3.3) will be enforced in the case of the phase field model for brittle fracture by a driving force and in the case of the phase field model for cohesive fracture by prescribing $d=1$ at nodes.

In the multi-dimensional case, the crack surface density function γ_ℓ can be extended as follows

$$\gamma_\ell = \frac{1}{4\ell} (d^2 + 4\ell^2 d_{,i} d_{,i}). \quad (3.7)$$

In Eq. (3.7), index notation is adopted with respect to a CARTESIAN frame.

3.2 Phase field model for brittle fracture

FRANCFORT & MARIGO [43] suggested to model fracture by a variational approach which minimises the energy of the bulk and the energy of the surface associated to the crack. The variational approach in FRANCFORT & MARIGO [43] was motivated by the

work of MUMFORD & SHAH [78] who introduced a potential for image segmentation. AMBROSIO & TORTORELLI [2] approximated the MUMFORD-SHAH potential by elliptic functionals using a control variable which ranges between zero and one. This idea of introducing an additional scalar field – also called phase field since it varies between the phases zero and one – was then adapted to fracture mechanics by BOURDIN *et al.* [17]. An auxiliary field (phase field) was introduced which smears the fracture energy over the volume of the solid.

The model by BOURDIN *et al.* [17] has been commonly adopted by others, for instance AMOR *et al.* [3], BORDEN *et al.* [14], KUHN & MÜLLER [63], LANCIONI & ROYER-CARFAGNI [65] and MIEHE *et al.* [77], to name just a few, with slight modifications of the length scale parameter ℓ and the convention whether $d=0$ or $d=1$ defines the fully broken state. Recently, BORDEN *et al.* [16] proposed a higher order phase field model for brittle fracture exploiting the higher continuity of splines.

An important assumption in the phase field model for brittle fracture is that the smeared potential Π_ℓ with the smeared crack surface Γ_ℓ Γ -converges to the discrete potential Π with the discrete crack surface Γ for a vanishing length scale parameter ℓ . Notice that $\Pi_\ell \rightarrow \Pi$ for $\ell \rightarrow 0$ implies that $\Gamma_\ell \rightarrow \Gamma$ for $\ell \rightarrow 0$. CHAMBOLLE [22] proved that this is the case for continuous media. BELLETTINI & COSCIA [11] considered the discretised version $\Pi_{\ell,h}$ of Π_ℓ and showed that $\Pi_{\ell,h}$ Γ -converges to Π for $\ell \rightarrow 0$ under the condition that $h \ll \ell$, h denoting the mesh size. Nevertheless, the *mathematical* consideration in BELLETTINI & COSCIA [11] has been done for image segmentation and not for the phase field approach to brittle fracture. It will be demonstrated in this section that in the context of the phase field model for brittle fracture, Γ -convergence is not attained *numerically*, i. e. $\Pi_{\ell,h}$ does not Γ -converge to Π for $\ell \rightarrow 0$ and $h \ll \ell$, see also MAY *et al.* [71]. After introducing the continuum formulation and Γ -convergence for the phase field model for brittle fracture in the next section, the finite element formulation will be derived. Then, the impact of different factors and parameters will be examined by means of some numerical examples.

3.2.1 Continuum formulation

In the following, the model by BOURDIN *et al.* [17] is briefly outlined. The potential for a solid with a discrete crack reads

$$\Pi = \int_{\Omega} \psi^{\text{el}} \, dV + \int_{\Gamma} \mathcal{G}_c \, dA, \quad (3.8)$$

where the first term denotes the elastic energy in the bulk and the second term the fracture energy which is created / dissipated upon crack propagation. The elastic energy density

ψ^{el} can be expressed by HOOKE's law

$$\psi^{\text{el}} = \frac{1}{2} \lambda \varepsilon_{ii}^{\text{el}} \varepsilon_{jj}^{\text{el}} + \mu \varepsilon_{ij}^{\text{el}} \varepsilon_{ij}^{\text{el}} = \frac{1}{2} \lambda \varepsilon_{ii} \varepsilon_{jj} + \mu \varepsilon_{ij} \varepsilon_{ij} \quad (3.9)$$

as a function of the infinitesimal strain tensor

$$\varepsilon_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i}), \quad (3.10)$$

with λ and μ the LAMÉ constants. In Eq. (3.10), u_i denotes the displacement. Notice that $\varepsilon_{ij}^{\text{el}} = \varepsilon_{ij}$ in Eq. (3.9). \mathcal{G}_c denotes the critical energy release rate in Eq. (3.8), i. e. the amount of energy needed to create a unit area of fracture surface. Using Eqs. (3.5) and (3.7), the fracture energy necessary to create a diffusive crack can be expressed by

$$\int_{\Gamma} \mathcal{G}_c \, dA = \int_{\Omega} \mathcal{G}_c \gamma_{\ell} \, dV. \quad (3.11)$$

Plugging the smeared fracture energy Eq. (3.11) into Eq. (3.8) requires the introduction of a link between the elastic energy density ψ^{el} and the phase field variable d . This link is inspired by damage models where a degradation function $g(d)$ reduces the stiffness of the bulk of the solid. Due to the smearing, the elastic energy density ψ^{el} associated to the undamaged solid needs to be replaced by the energy density ψ which corresponds to the damaged solid. It is split into two parts – a damaged part ψ^{d} on which a degradation function $g(d)$ acts, and an intact part ψ^{i}

$$\psi(\varepsilon_{ij}, d) = g(d) \psi^{\text{d}}(\varepsilon_{ij}) + \psi^{\text{i}}(\varepsilon_{ij}). \quad (3.12)$$

This split is motivated by the observation that the tensile strain components contribute to the damage process that results in fracture, while the compression strain components do not. Various splits of the energy density ψ have been investigated by AMOR *et al.* [3].

The degradation function $g(d)$ in Eq. (3.12) has to fulfil the following properties

- $g(0) = 1$ since for $d = 0$ no damage occurs;
- $g(1) = 0$ since for $d = 1$ the damaged part ψ^{d} has to vanish;
- $g'(0) \neq 0$ since the damage has to be initiated at the onset;
- $g'(1) = 0$ since the energy must converge to a finite value for the fully broken state.

For the degradation function $g(d)$, use has often been made of the quadratic function

$$g(d) = (1 - d)^2. \quad (3.13)$$

BORDEN [15] has introduced a cubic degradation function,

$$g(d) = s((1-d)^3 - (1-d)^2) + 3(1-d)^2 - 2(1-d)^3, \quad (3.14)$$

with the additional parameter s . Eq. (3.14) results in force-displacement curves which better reflect the behaviour of brittle materials, as less damage occurs before reaching the peak load, see also Section 3.2.3.

Notice that the crack in Eq. (3.11) has been smeared on purely mathematical grounds, whereas the introduction of the degradation function $g(d)$ in Eq. (3.12) is inspired by a phenomenological concept that is commonly used in damage mechanics.

Substituting Eqs. (3.11), (3.12) and (3.13) into Eq. (3.8) yields the potential Π_ℓ for a solid with a smeared crack

$$\Pi_\ell = \int_{\Omega} ((1-d)^2 \psi^d + \psi^i) \, dV + \int_{\Omega} \mathcal{G}_c \gamma_\ell \, dV. \quad (3.15)$$

Γ -convergence (with $\psi^d = \psi^{\text{el}}$ and $\psi^i = 0$) is then defined such that the potential Π_ℓ with the smeared crack Γ_ℓ in Eq. (3.15) converges to the potential Π with the discrete crack Γ in Eq. (3.8) when $\ell \rightarrow 0$, i. e.

$$\Pi_\ell|_{\ell \rightarrow 0} = \left(\int_{\Omega} (1-d)^2 \psi^{\text{el}} \, dV + \int_{\Omega} \mathcal{G}_c \gamma_\ell \, dV \right) \Big|_{\ell \rightarrow 0} = \int_{\Omega} \psi^{\text{el}} \, dV + \int_{\Gamma} \mathcal{G}_c \, dA = \Pi. \quad (3.16)$$

According to CHAMBOLLE [22], the potential

$$\Pi_\ell(u_\ell, d) = \int_{\Omega} ((1-d)^2 + \eta) \psi^{\text{el}}(u_\ell) \, dV + \int_{\Omega} \mathcal{G}_c \gamma_\ell(d) \, dV, \quad (3.17)$$

with the stabilisation parameter η , Γ -converges for $\eta \rightarrow 0$ and $\ell \rightarrow 0$ ($\eta \ll \ell$) to

$$\Pi(u) = \int_{\Omega} \psi^{\text{el}}(u) \, dV + \int_{\Gamma} \mathcal{G}_c \, dA \quad (3.18)$$

if the global minimisers u_ℓ of Π_ℓ converge to the global minimisers u of Π . Furthermore, the Γ -convergence result of BELLETTINI & COSCIA [11] reads in the mechanical context: the discretised version $\Pi_{\ell,h}$ of Π_ℓ with

$$\Pi_{\ell,h}(u_{\ell,h}, d_h) = \int_{\Omega} ((1-d_h)^2 + \eta) \psi^{\text{el}}(u_{\ell,h}) \, dV + \int_{\Omega} \mathcal{G}_c \gamma_\ell(d_h) \, dV \quad (3.19)$$

Γ -converges to Π for $\eta \rightarrow 0$, $\ell \rightarrow 0$ and $h \rightarrow 0$ ($\eta \ll \ell$, $h \ll \ell$). In the numerical examples in Section 3.2.3, $\eta=0$ similar to BORDEN *et al.* [14].

According to BOURDIN *et al.* [19], a correction factor must be applied to the critical

energy release rate \mathcal{G}_c when it is compared with the critical energy release rate of the discretised solid $\mathcal{G}_{c,h}$ as follows

$$\mathcal{G}_{c,h} = \left(1 + \frac{h}{4\ell}\right) \mathcal{G}_c, \quad (3.20)$$

i. e. the critical energy release rate \mathcal{G}_c is amplified by a factor $1 + \frac{h}{4\ell}$. Hence, when investigating Γ -convergence in Section 3.2.3, the smeared crack $\Gamma_{\ell,h}$ in Eq. (3.5) needs to be compared with $(1 + \frac{h}{4\ell}) \Gamma$ when the correction factor is taken into account, see also BORDEN *et al.* [16] where this correction has been considered in numerical studies of Γ -convergence for second and fourth-order phase field models. In that study, Γ -convergence was obtained numerically for a prescribed displacement field. This is different from MAY *et al.* [71] where the entire displacement field was not prescribed.

In practical computations, the restriction $h \ll \ell$ can be difficult to fulfil, especially since the length scale parameter ℓ already needs to be small in order to resolve the crack properly. In numerical simulations the weaker condition $h < \ell$ is usually adopted: AMOR *et al.* [3], BOURDIN [18], BOURDIN *et al.* [19], BORDEN *et al.* [14, 16], DEL PIERO *et al.* [31], KUHN & MÜLLER [63], MIEHE *et al.* [76].

For a given equilibrium configuration the variation of the potential Π_ℓ with the smeared crack Γ_ℓ in Eq. (3.15) is required to be zero

$$\delta\Pi_\ell = \frac{\partial\Pi_\ell}{\partial\varepsilon_{ij}}\delta\varepsilon_{ij} + \frac{\partial\Pi_\ell}{\partial d}\delta d + \frac{\partial\Pi_\ell}{\partial d_{,i}}\delta d_{,i} = 0. \quad (3.21)$$

Since $\delta\Pi_\ell = 0$ must hold for any $\delta\varepsilon_{ij}$, δd and $\delta d_{,i}$, this leads to the following system of equations

$$\sigma_{ij,i} = 0, \quad (3.22)$$

$$\frac{\mathcal{G}_c}{2\ell}(d - 4\ell^2 d_{,ii}) + \frac{\partial g}{\partial d}\mathcal{H} = 0 \quad (3.23)$$

subject to the boundary conditions

$$\sigma_{ij}n_j = h_i \quad \text{on } \partial\Omega_h, \quad u_i = \bar{u}_i \quad \text{on } \partial\Omega_u, \quad d_{,i}n_i = 0 \quad \text{on } \partial\Omega, \quad (3.24)$$

with $\partial\Omega_h \cap \partial\Omega_u = \emptyset$, $\partial\Omega_h \cup \partial\Omega_u = \partial\Omega$, the prescribed surface traction \mathbf{h} , the prescribed displacement $\bar{\mathbf{u}}$ and the normal vector \mathbf{n} on the boundary. The stress tensor σ_{ij} in Eq. (3.22) is defined as

$$\sigma_{ij} = \frac{\partial\psi}{\partial\varepsilon_{ij}} = g(d)\frac{\partial\psi^d}{\partial\varepsilon_{ij}} + \frac{\partial\psi^i}{\partial\varepsilon_{ij}}. \quad (3.25)$$

The formulation by MIEHE *et al.* [76] has been applied to Eq. (3.23). \mathcal{H} describes the history field

$$\mathcal{H} = \max \psi^d, \quad (3.26)$$

which ensures irreversibility since cracks can only grow ($\dot{d} \geq 0$). Alternatively, in KUHN & MÜLLER [63], irreversibility has been enforced by setting $d=1$ when d approaches one. The term $\frac{\partial g}{\partial d} \mathcal{H}$ in Eq. (3.23) can be interpreted as the driving force which drives the damage evolution and ensures that $d \rightarrow 1$ for $\psi^d \rightarrow \infty$.

Multiplying the strong forms of Eqs. (3.22) and (3.23) by a test function δu_j and δd , integrating over the domain Ω and applying integration by parts results with GAUSS' theorem in the weak forms

$$\begin{aligned} \int_{\Omega} \delta u_{i,j} \sigma_{ij} \, dV &= \int_{\partial\Omega_h} \delta u_i h_i \, dA, \\ \int_{\Omega} \frac{\mathcal{G}_c}{2\ell} (\delta d d + 4\ell^2 \delta d_{,i} d_{,i}) + \delta d \frac{\partial g}{\partial d} \mathcal{H} \, dV &= 0. \end{aligned} \quad (3.27)$$

3.2.2 Finite element formulation

Discretising the domain Ω into E elements,

$$\Omega = \bigcup_{e=1}^E \Omega^e, \quad (3.28)$$

and approximating the field variables and their derivatives,

$$\underline{\mathbf{u}}^e = \underline{\mathbf{N}}_u \underline{\mathbf{u}}, \quad \delta \underline{\mathbf{u}}^e = \underline{\mathbf{N}}_u \delta \underline{\mathbf{u}}, \quad \underline{\boldsymbol{\varepsilon}}^e = \underline{\mathbf{B}}_u \underline{\mathbf{u}}, \quad \delta \underline{\boldsymbol{\varepsilon}}^e = \underline{\mathbf{B}}_u \delta \underline{\mathbf{u}}, \quad (3.29)$$

$$d^e = \underline{\mathbf{N}}_d^T \underline{\mathbf{d}}, \quad \delta d^e = \underline{\mathbf{N}}_d^T \delta \underline{\mathbf{d}}, \quad \underline{\mathbf{d}}_{,i}^e = \underline{\mathbf{B}}_d \underline{\mathbf{d}}, \quad \delta \underline{\mathbf{d}}_{,i}^e = \underline{\mathbf{B}}_d \delta \underline{\mathbf{d}}, \quad (3.30)$$

with

$$\begin{aligned} \underline{\mathbf{u}}^e &= \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix}, \quad \delta \underline{\mathbf{u}}^e = \begin{bmatrix} \delta u_1^e \\ \delta u_2^e \end{bmatrix}, \quad \underline{\boldsymbol{\varepsilon}}^e = \begin{bmatrix} \varepsilon_{11}^e \\ \varepsilon_{22}^e \\ 2\varepsilon_{12}^e \end{bmatrix}, \quad \delta \underline{\boldsymbol{\varepsilon}}^e = \begin{bmatrix} \delta \varepsilon_{11}^e \\ \delta \varepsilon_{22}^e \\ 2\delta \varepsilon_{12}^e \end{bmatrix}, \\ \underline{\mathbf{d}}_{,i}^e &= \begin{bmatrix} d_{,1}^e \\ d_{,2}^e \end{bmatrix}, \quad \delta \underline{\mathbf{d}}_{,i}^e = \begin{bmatrix} \delta d_{,1}^e \\ \delta d_{,2}^e \end{bmatrix} \end{aligned} \quad (3.31)$$

and

$$\underline{\underline{\mathbf{N}}}_u = \begin{bmatrix} N_1 & 0 & N_2 & 0 & \dots \\ 0 & N_1 & 0 & N_2 & \dots \end{bmatrix}, \quad \underline{\underline{\mathbf{B}}}_u = \begin{bmatrix} N_{1,1} & 0 & N_{2,1} & 0 & \dots \\ 0 & N_{1,2} & 0 & N_{2,2} & \dots \\ N_{1,2} & N_{1,1} & N_{2,2} & N_{2,1} & \dots \end{bmatrix}, \quad (3.32)$$

$$\underline{\underline{\mathbf{N}}}_d^T = \begin{bmatrix} N_1 & N_2 & \dots \end{bmatrix}, \quad \underline{\underline{\mathbf{B}}}_d = \begin{bmatrix} N_{1,1} & N_{2,1} & \dots \\ N_{1,2} & N_{2,2} & \dots \end{bmatrix}$$

results in the following matrix-vector equation for Eq. (3.27)

$$\delta \underline{\underline{\mathbf{u}}}^T \underbrace{\int_{\Omega} \underline{\underline{\mathbf{B}}}_u^T (g \underline{\underline{\mathbf{C}}}^d + \underline{\underline{\mathbf{C}}}^i) \underline{\underline{\mathbf{B}}}_u \underline{\underline{\mathbf{u}}} \, dV}_{\underline{\underline{\mathbf{f}}}_u^{\text{int}}(\underline{\underline{\mathbf{d}}}, \underline{\underline{\mathbf{u}}})} - \delta \underline{\underline{\mathbf{u}}}^T \underbrace{\int_{\Gamma} \underline{\underline{\mathbf{N}}}_u^T \underline{\underline{\mathbf{h}}} \, dA}_{\underline{\underline{\mathbf{f}}}_u^{\text{ext}}} = 0, \quad (3.33)$$

$$\delta \underline{\underline{\mathbf{d}}}^T \underbrace{\int_{\Omega} \frac{\mathcal{G}_c}{2\ell} (\underline{\underline{\mathbf{N}}}_d \underline{\underline{\mathbf{N}}}_d^T + 4\ell^2 \underline{\underline{\mathbf{B}}}_d^T \underline{\underline{\mathbf{B}}}_d) \underline{\underline{\mathbf{d}}} + \underline{\underline{\mathbf{N}}}_d \frac{\partial g}{\partial \underline{\underline{\mathbf{d}}}} \mathcal{H} \, dV}_{\underline{\underline{\mathbf{f}}}_d^{\text{int}}(\underline{\underline{\mathbf{d}}}, \underline{\underline{\mathbf{u}}})} = 0, \quad (3.34)$$

where $\underline{\underline{\mathbf{C}}}^d$ corresponds to the damaged part of the elasticity matrix, and $\underline{\underline{\mathbf{C}}}^i$ to the intact part of the elasticity matrix. $\underline{\underline{\mathbf{f}}}_u^{\text{int}}(\underline{\underline{\mathbf{d}}}, \underline{\underline{\mathbf{u}}})$ is the internal force vector corresponding to the mechanical field problem and $\underline{\underline{\mathbf{f}}}_d^{\text{int}}(\underline{\underline{\mathbf{d}}}, \underline{\underline{\mathbf{u}}})$ is the internal force vector related to the phase field. Eqs. (3.33) and (3.34) are required to hold for any $\delta \underline{\underline{\mathbf{u}}}$ and $\delta \underline{\underline{\mathbf{d}}}$. Setting $\underline{\underline{\mathbf{h}}} = \underline{\underline{\mathbf{0}}}$ and taking into account nodal forces yields for Eq. (3.33)

$$\underline{\underline{\mathbf{f}}}_u^{\text{int}}(\underline{\underline{\mathbf{d}}}, \underline{\underline{\mathbf{u}}}) - \lambda \hat{\underline{\underline{\mathbf{f}}}} = \underline{\underline{\mathbf{0}}}, \quad (3.35)$$

with $\hat{\underline{\underline{\mathbf{f}}}}$ a normalised external force vector, and λ a loading parameter. In order to trace the equilibrium path, the arc-length control method from Chapter 2 is utilised. Denoting the arc-length function by φ , the following system of equations must be solved

$$\underline{\underline{\mathbf{H}}}(\underline{\underline{\mathbf{d}}}, \underline{\underline{\mathbf{u}}}, \lambda) = \begin{bmatrix} \underline{\underline{\mathbf{f}}}_d^{\text{int}}(\underline{\underline{\mathbf{d}}}, \underline{\underline{\mathbf{u}}}) \\ \underline{\underline{\mathbf{f}}}_u^{\text{int}}(\underline{\underline{\mathbf{d}}}, \underline{\underline{\mathbf{u}}}) - \lambda \hat{\underline{\underline{\mathbf{f}}}} \\ \varphi(\underline{\underline{\mathbf{u}}}, \lambda) \end{bmatrix} = \begin{bmatrix} \underline{\underline{\mathbf{0}}} \\ \underline{\underline{\mathbf{0}}} \\ 0 \end{bmatrix}. \quad (3.36)$$

Linearisation of Eq. (3.36) yields the solution at iteration $i+1$ in the increment $k+1$

$$\begin{bmatrix} \underline{\underline{\mathbf{d}}} \\ \underline{\underline{\mathbf{u}}} \\ \lambda \end{bmatrix}_{k+1}^{i+1} = \begin{bmatrix} \underline{\underline{\mathbf{d}}} \\ \underline{\underline{\mathbf{u}}} \\ \lambda \end{bmatrix}_{k+1}^i - \underline{\underline{\mathbf{K}}}_T^{-1} \Big|_{k+1}^i \cdot \begin{bmatrix} \underline{\underline{\mathbf{f}}}_d^{\text{int}}(\underline{\underline{\mathbf{d}}}, \underline{\underline{\mathbf{u}}}) \\ \underline{\underline{\mathbf{f}}}_u^{\text{int}}(\underline{\underline{\mathbf{d}}}, \underline{\underline{\mathbf{u}}}) - \lambda \hat{\underline{\underline{\mathbf{f}}}} \\ \varphi(\underline{\underline{\mathbf{u}}}, \lambda) \end{bmatrix}_{k+1}^i, \quad (3.37)$$

with the tangential stiffness matrix

$$\underline{\underline{\mathbf{K}}}_T(\underline{\mathbf{d}}, \underline{\mathbf{u}}, \lambda) = \begin{bmatrix} \frac{\partial \underline{\mathbf{f}}_d^{\text{int}}(\underline{\mathbf{d}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{d}}} & \frac{\partial \underline{\mathbf{f}}_d^{\text{int}}(\underline{\mathbf{d}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}} & \underline{\mathbf{0}} \\ \frac{\partial \underline{\mathbf{f}}_u^{\text{int}}(\underline{\mathbf{d}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{d}}} & \frac{\partial \underline{\mathbf{f}}_u^{\text{int}}(\underline{\mathbf{d}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}} & -\hat{\underline{\mathbf{f}}} \\ \underline{\mathbf{0}}^T & \frac{\partial \varphi(\underline{\mathbf{u}}, \lambda)}{\partial \underline{\mathbf{u}}} & \frac{\partial \varphi(\underline{\mathbf{u}}, \lambda)}{\partial \lambda} \end{bmatrix}. \quad (3.38)$$

3.2.3 Numerical examples

This section studies the performance of the phase field model for brittle fracture using a one-dimensional bar, see also MAY *et al.* [71] and VIGNOLLET *et al.* [118]. The following factors are considered: mesh convergence study, varying length scale parameter ℓ , different degradation function $g(d)$ and monolithic vs. staggered approach. Then, it will be demonstrated that Γ -convergence is not attained numerically for the one-dimensional bar. Subsequently, the impact of different boundary conditions on the phase field model for brittle fracture is addressed. Only linear interpolation functions for displacement and phase field will be considered in this section.

3.2.3.1 One-dimensional bar with reduced thickness in the middle under tension

Consider the one-dimensional bar depicted in Fig. 3.2.

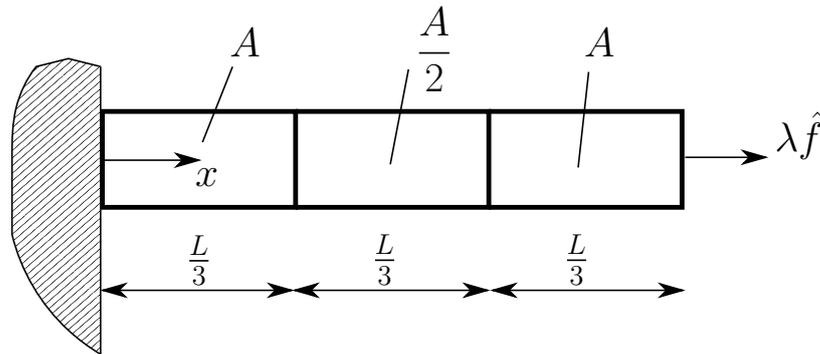


Fig. 3.2: Bar with a reduced thickness in the middle subject to the load $\lambda \hat{f}$ applied to the right edge.

The bar has a reduced cross-section in the middle and the load $\lambda \hat{f}$ is applied to the right edge. The parameters are $E = 10$ MPa for the YOUNG's modulus, $\mathcal{G}_c = 0.1$ N/mm for the critical energy release rate, $L = 1$ mm for the length of the bar, $A = 1$ mm² and $\ell = \frac{L}{20}$ for the length scale parameter. The quadratic degradation function $g(d) = (1 - d)^2$ acts directly on the YOUNG's modulus E since $\psi^d = E$ and $\psi^i = 0$.

Mesh size h

The force-displacement curves for different mesh sizes h are depicted in Fig. 3.3. The parameters for the arc-length control in Chapter 2 are $\Delta\tau_1^F = 0.1$ N and $a = 1$. Here and in the following, circles denote the arc-length control based on the rate of internal energy while triangles denote the arc-length control based on the rate of dissipated energy.

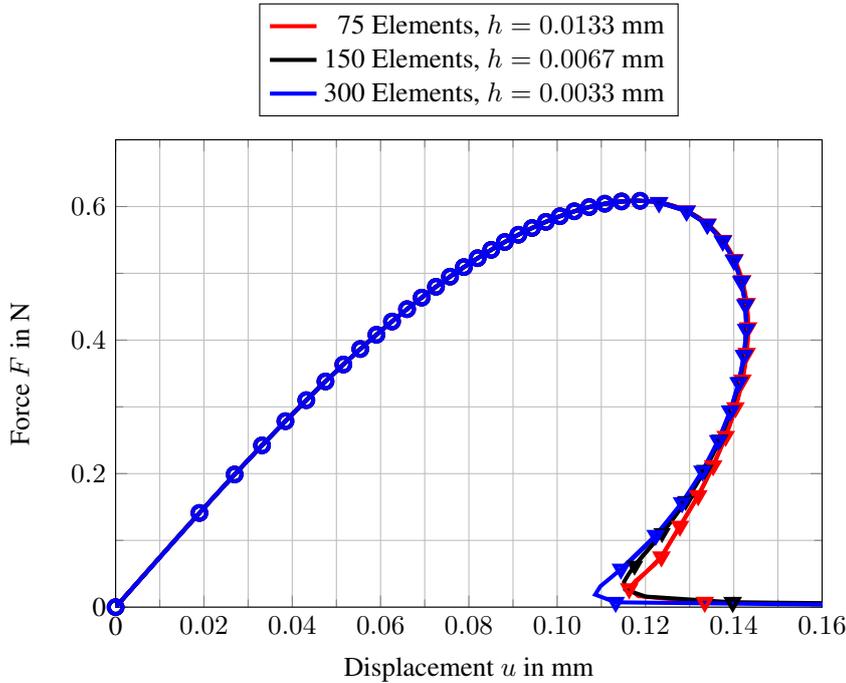


Fig. 3.3: Force-displacement curves for different mesh sizes h with $\ell = 0.05$ mm, $\Delta\tau_1^F = 0.1$ N and $a = 1$.

Length scale parameter ℓ

The impact of a varying length scale parameter ℓ on the force-displacement curve is shown in Fig. 3.4 for a constant mesh of 150 elements, $h = 0.0067$ mm. The rule of thumb $\ell > h$ is respected in order to accurately approximate the crack topology MIEHE *et al.* [77]. The parameters for the arc-length control are $\Delta\tau_1^F = 0.1$ N and $a = 1$.

A decreasing length scale parameter ℓ results in a higher peak force. Furthermore, a smaller length scale parameter ℓ results in a more pronounced snap-back behaviour. BORDEN *et al.* [14] mentioned that the length scale parameter ℓ may be interpreted as a material parameter since it influences the critical stress. This makes it difficult to decide how ℓ should be treated for the phase field model for brittle fracture. On the one hand, the length scale parameter ℓ has been originally introduced in Section 3.1 as a mathematical approximation of the crack, but on the other hand, numerical experiments show that it should be treated as a material parameter.

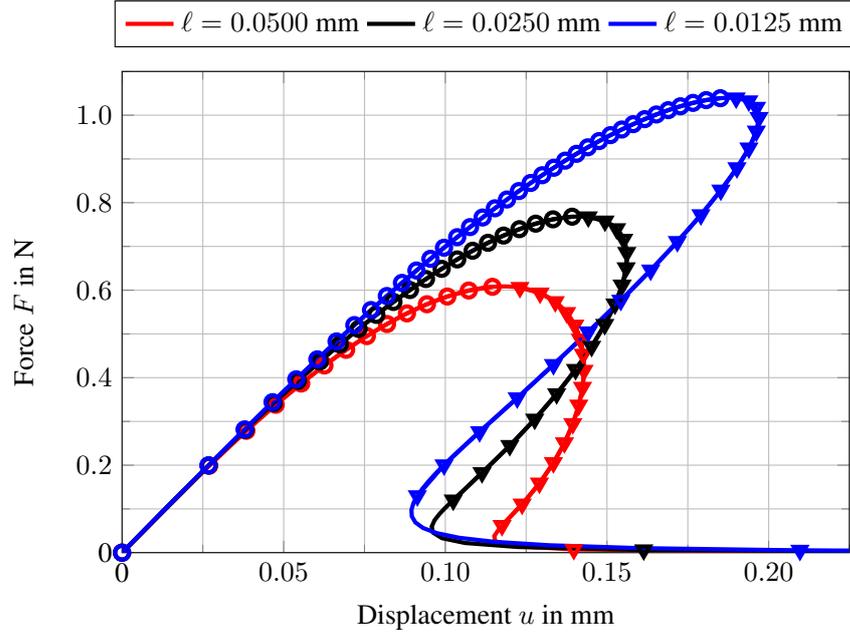


Fig. 3.4: Force-displacement curves for different length scale parameters ℓ and a constant mesh consisting of 150 elements, mesh size is $h=0.0067$ mm; arc-length control parameters are $\Delta\tau_1^F=0.1$ N and $a=1$.

Degradation function $g(d)$

It appears from Figs. 3.3 and 3.4 that the phase field model for brittle fracture does not exhibit brittle behaviour. It can be observed that the force-displacement curves depart early from linearly elastic behaviour since at the onset of loading, the driving force term $\frac{\partial g}{\partial d}\mathcal{H}$ in Eq. (3.23) starts to grow. Hence, the phase field variable d and the degradation function $g(d)$ start to grow as well and reduce the stiffness of the bar. Therefore, the cubic degradation function in Eq. (3.14) was proposed by BORDEN [15]. Its impact is studied next. The force-displacement curves for the quadratic and cubic degradation function $g(d)$ are shown in Fig. 3.5 for a constant length scale parameter $\ell=0.05$ mm and a constant mesh size of 150 elements, so that the mesh size is $h=0.0067$ mm.

The arc-length parameters are $\Delta\tau_1^F=0.1$ N, $a=1$ for the quadratic case and $\Delta\tau_1^F=0.1$ N, $a=0.5$ for the cubic case. It can be observed that the additional parameter s influences the behaviour of the force-displacement curve. Furthermore, the cubic degradation function results in a more linear behaviour at the onset of loading, and the snap-back behaviour is more pronounced.

Monolithic vs. staggered approach

As a next step, the importance of a monolithic solver is investigated. In MIEHE *et al.* [77] a staggered approach has been used. The staggered approach solves the system of equations in Eqs. (3.33) and (3.34) in an alternate scheme. It does not invoke an iteration loop between both field problems in each increment. Instead, in each increment two steps are

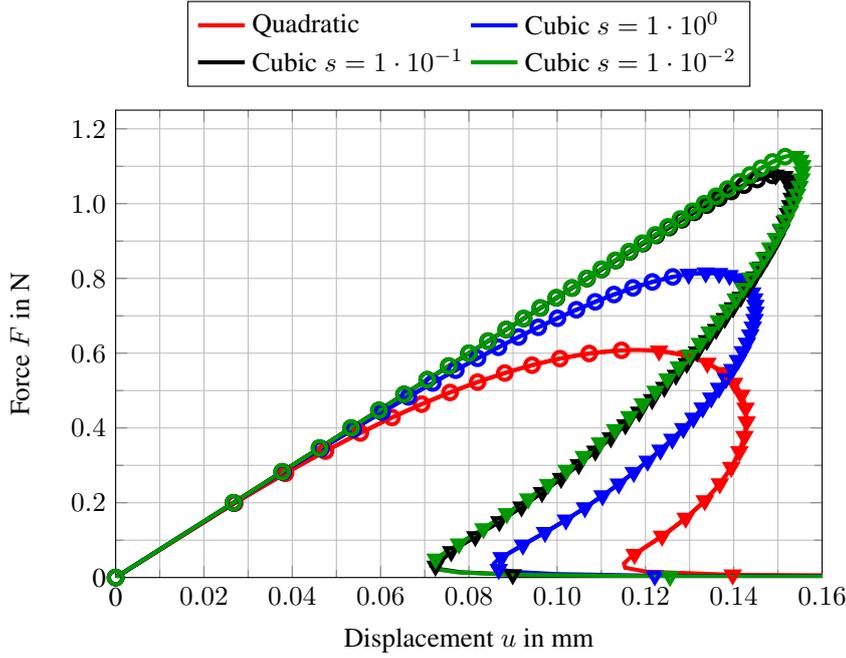


Fig. 3.5: Comparison of the quadratic and the cubic degradation function $g(d)$ for different parameters s , a constant mesh size of 150 elements, mesh size is $h=0.0067$ mm and a constant length scale parameter $\ell=0.05$ mm; parameters for the arc-length control are $\Delta\tau_1^F=0.1$ N, $a=1$ for the quadratic case and $\Delta\tau_1^F=0.1$ N, $a=0.5$ for the cubic case.

performed. In a first step, the state of the displacement field u_i is frozen and Eq. (3.34) is solved for the phase field d . In the second step, the state of d is frozen and Eq. (3.33) is solved for the displacement field u_i . Fig. 3.6 shows that the staggered approach is not able to capture the snap-back behaviour.

The staggered approach is robust in the sense that a solution is always obtained, but this solution does not necessarily represent an equilibrium state – a converged equilibrium solution can be obtained only with a monolithic scheme, or a staggered scheme with an iteration loop in each increment between both field problems.

Γ -convergence

Next, Γ -convergence will be checked numerically by examining the numerical final crack surface $\Gamma_{\ell,h}$ of the one-dimensional bar. For the simulation, $\Gamma_{\ell,h}$ can be obtained by calculating the discretised version of Eq. (3.5),

$$\Gamma_{\ell,h} = \int_{\Omega} \frac{1}{4\ell} (d_h^2 + 4\ell^2 d_{h,x}^2) \, dV, \quad (3.39)$$

when $\max d_h > 0.99$. The error Γ_E is defined as follows

$$\Gamma_E = \frac{|\Gamma_{\ell,h} - \Gamma|}{\Gamma}. \quad (3.40)$$

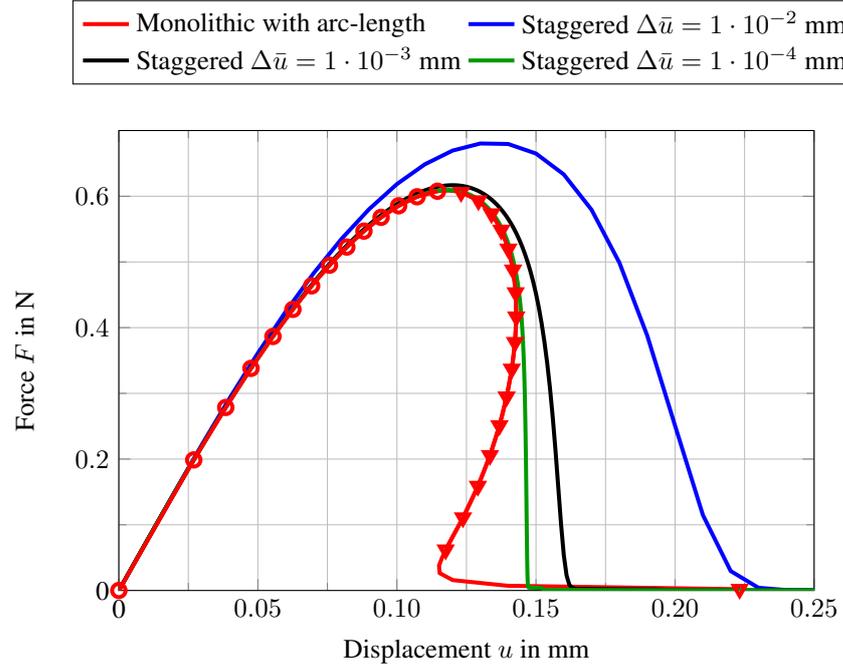


Fig. 3.6: Comparison of the staggered and monolithic approach for the length scale parameter $\ell=0.05$ mm and a constant mesh consisting of 150 elements, mesh size is $h=0.0067$ mm; for the monolithic approach, the arc-length parameters are $\Delta\tau_1^F=0.1$ N and $a=1$.

When the correction factor in BOURDIN *et al.* [19] is taken into account, the error Γ_E is computed with

$$\Gamma_E = \frac{|\Gamma_{\ell,h} - (1 + \frac{h}{4\ell}) \Gamma|}{(1 + \frac{h}{4\ell}) \Gamma}. \quad (3.41)$$

The theoretical final crack surface for the bar in Fig. 3.2 is $\Gamma = A/2$, which is equal to the cross section of the segment in the centre of the bar.

Fig. 3.7 gives the convergence study of the final crack surface $\Gamma_{\ell,h}$ when using the quadratic degradation function $g(d)$ in Eq. (3.13) for different values of the length scale parameter ℓ . Three different mesh sizes h have been used.

Fig. 3.7 shows that all results for $\Gamma_{\ell,h}$ give a poor approximation of the theoretical final crack surface Γ since Eq. (3.40) yields for the error $\Gamma_E > 0.1$. Below a certain value of the ratio ℓ/h , a further *decrease* of the ratio ℓ/h results in an *increase* of the error Γ_E . Fig. 3.8 shows that this also holds when the correction factor in Eq. (3.41) proposed by BOURDIN *et al.* [19] is taken into account.

The minimum in Figs. 3.7 and 3.8 occurs for the same values of the internal length scale ($\ell=0.05$ mm). Also, it seems that there is a range for ℓ/h for each discretisation for which Γ_E becomes minimal. Surprisingly, the error Γ_E does not decrease for a fixed length scale parameter ℓ upon mesh refinement.

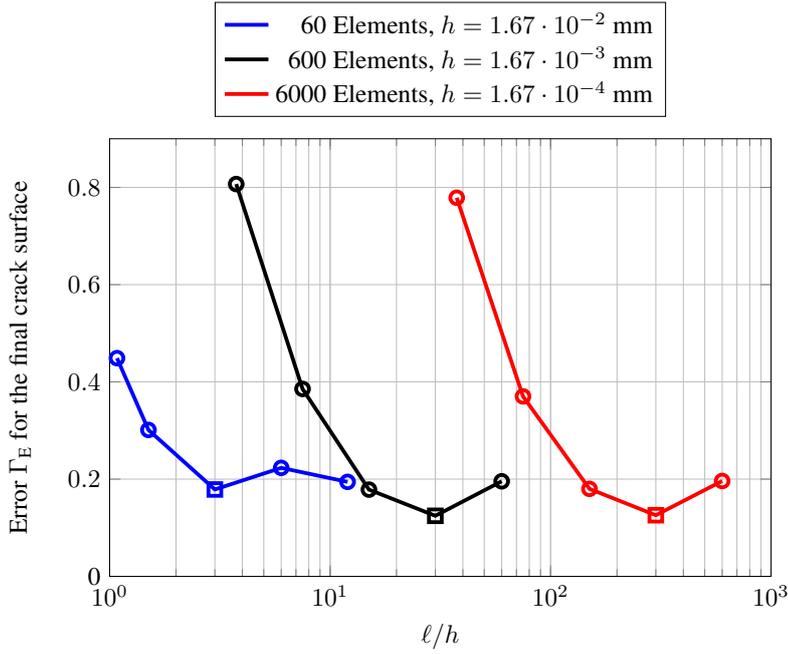


Fig. 3.7: Convergence study for the final crack surface $\Gamma_{\ell,h}$ for a one-dimensional bar. The squares correspond to $\ell = 0.05$ mm.

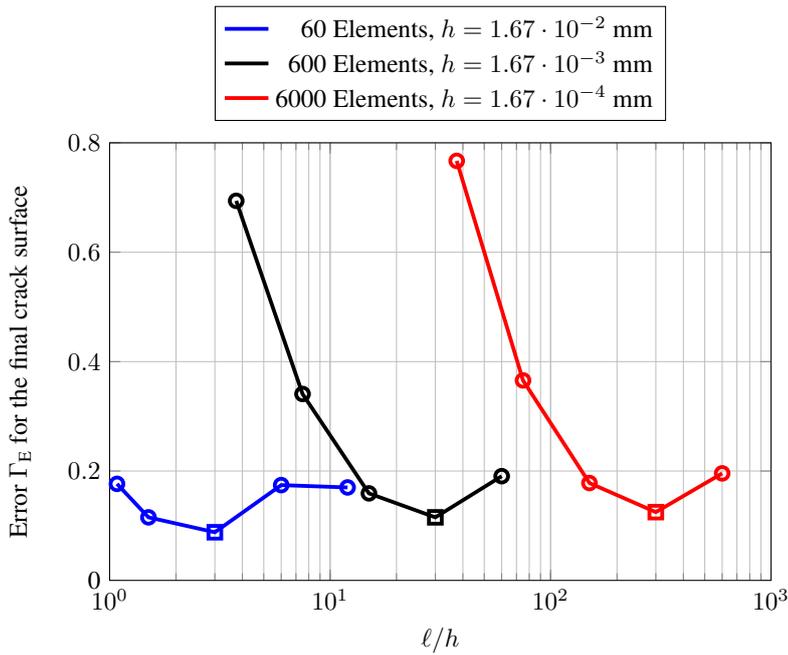


Fig. 3.8: Convergence study for the final crack surface $\Gamma_{\ell,h}$ for a one-dimensional bar taking into account the correction factor in Eq. (3.41) proposed by BOURDIN *et al.* [19]. The squares correspond to $\ell = 0.05$ mm.

Fig. 3.9 shows the evolution of the phase field variable d during the loading process for the mesh with 600 elements.

The distribution seems to reasonably match the theoretical profile when $d = 1$ is prescribed

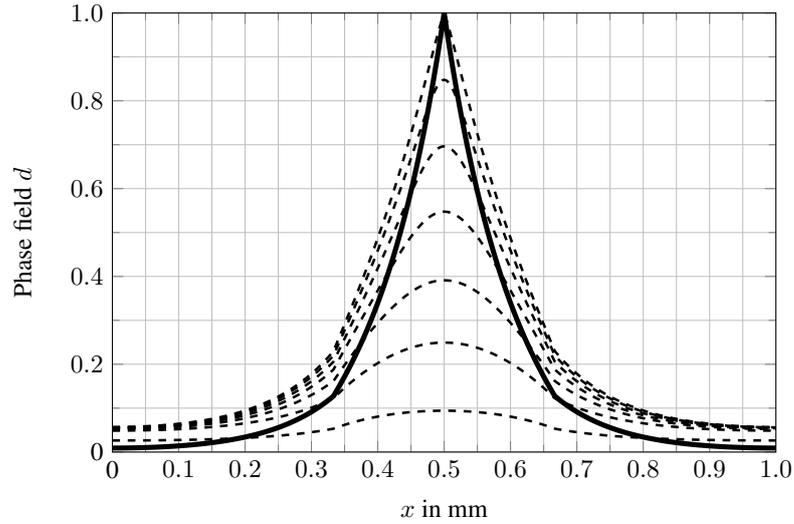


Fig. 3.9: Evolution of the phase field variable d at various stages in the loading process (dashed black). The solid black curve represents the distribution for d when $d=1$ is prescribed in the centre of the bar. The mesh size is $h = 1.67 \cdot 10^{-3}$ mm for 600 elements and the length scale parameter is $\ell = 0.00625$ mm.

in the centre of the bar. However, the phase field variable d is larger than the optimal profile along the entire length of the bar.

The considered convergence study in Figs. 3.7 and 3.8 for the numerical final crack length $\Gamma_{\ell,h}$ indicates that $\Gamma_{\ell,h}$ does not converge to the discrete crack surface Γ for $\ell \rightarrow 0$. Hence, it can be concluded that for phase field model for brittle fracture Γ -convergence is not attained,

$$\Pi_{\ell,h}|_{\ell \rightarrow 0} \neq \Pi \quad \text{since} \quad \Gamma_{\ell,h}|_{\ell \rightarrow 0} \neq \Gamma. \quad (3.42)$$

In view of the results for the one-dimensional bar in this section, the phase field variable d should rather be interpreted as a damage variable than a variable that represents a smeared crack since Γ -convergence does not manifest numerically. Moreover, the degradation function $g(d)$ is motivated from damage mechanics and the differential equation in Eq. (3.23) should be seen as an equation which describes the evolution of the damage variable d .

3.2.3.2 Impact of the boundary conditions

Next, it will be demonstrated that the phase field model for brittle fracture can give different results depending on how the boundary conditions are imposed. Consider the plate in Fig. 3.10 subject to a shear load. Notice that a shear load is obtained only under the assumption of small deformations.

The bottom edge is fixed in the x -direction. All edges are fixed in the y -direction – also the notch in order to prevent penetration. The material parameters are:

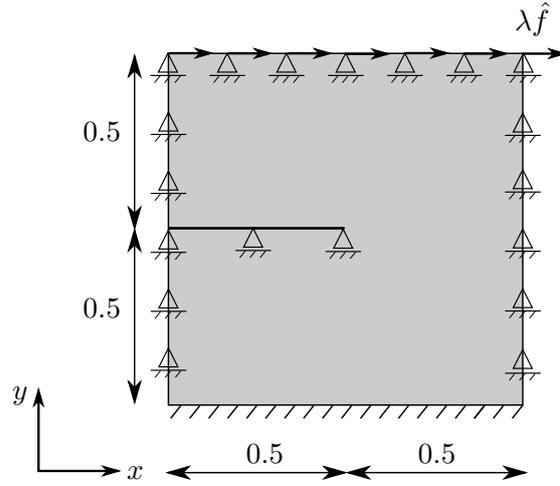


Fig. 3.10: Plate under shear (for small deformations), dimensions in mm; the initial notch is modelled as a discrete crack.

YOUNG's modulus $E = 210$ MPa, POISSON's ratio $\nu = 0.3$ and critical energy release rate $\mathcal{G}_c = 2.7 \cdot 10^{-3}$ N/mm. Plane strain is assumed. The arc-length parameters for the path following technique in Chapter 2 are $\Delta\tau_1^F = 0.1$ N and $a = 0.1$. In order to account for damage only in tension, the strain tensor $\boldsymbol{\varepsilon}$ will be decomposed into a positive and negative part as by MIEHE *et al.* [76],

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^+ + \boldsymbol{\varepsilon}^-, \quad (3.43)$$

with

$$\boldsymbol{\varepsilon}_{ij}^+ = P_{ijkl}^+ \boldsymbol{\varepsilon}_{kl}, \quad \boldsymbol{\varepsilon}_{ij}^- = P_{ijkl}^- \boldsymbol{\varepsilon}_{kl}, \quad (3.44)$$

where P_{ijkl}^+ and P_{ijkl}^- are two projection tensors. Due to the decomposition, ψ^d contains contributions that stem from the positive / tensile strains

$$\psi^d = \frac{1}{2} \lambda \varepsilon_{ii}^+ \varepsilon_{jj}^+ + \mu \varepsilon_{ij}^+ \varepsilon_{ij}^+, \quad (3.45)$$

and ψ^i contains those from the negative / compressive strains

$$\psi^i = \frac{1}{2} \lambda \varepsilon_{ii}^- \varepsilon_{jj}^- + \mu \varepsilon_{ij}^- \varepsilon_{ij}^-. \quad (3.46)$$

The notch is first modelled in a discrete sense, by applying the boundary conditions shown in Fig. 3.10. The load $\lambda \hat{f}$ is applied at the top edge in the positive x -direction. The length scale parameter is taken as $\ell = 0.01$ mm and the mesh size $h = 0.01$ mm for a mesh with 100×100 elements. The force-displacement curve and the evolution of the phase field variable d are depicted in Fig. 3.11 and Fig. 3.12, respectively.

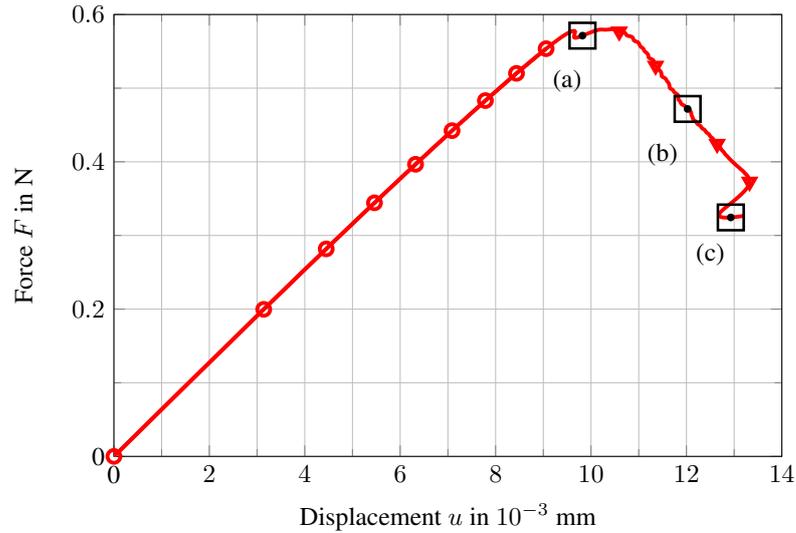


Fig. 3.11: Force-displacement curve for the plate under shear (assuming small deformations) when the initial notch is modelled as a discrete crack; the mesh consists of 100×100 elements, so that the mesh size is $h = 0.01$ mm; the length scale parameter is $\ell = 0.01$ mm. Squares correspond to the phase field distributions for d in Fig. 3.12.

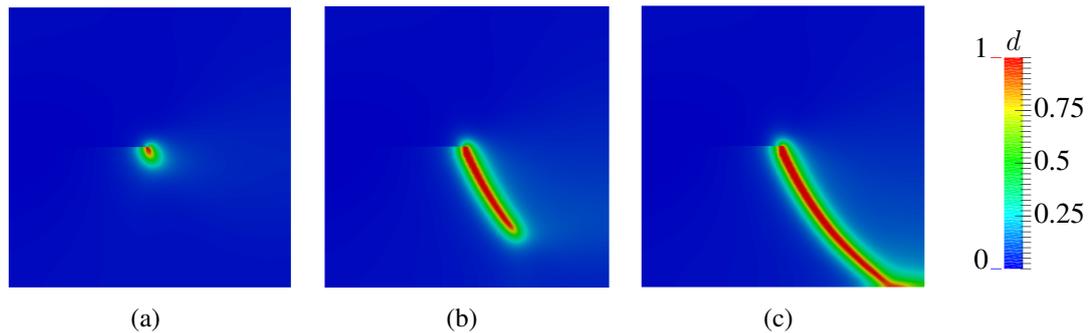


Fig. 3.12: Propagation of the variable d for the plate under shear (assuming small deformations) when the initial notch is modelled as a discrete crack; plots correspond to the squares in Fig. 3.11.

Next, the initial notch is introduced by prescribing $d = 1$, cf. Fig. 3.13.

With this boundary condition, the force-displacement curve in Fig. 3.14 is obtained. The corresponding patterns for the propagation of the phase field variable d are depicted in Fig. 3.15, and are very different from those in Fig. 3.12 since the crack propagates along the left top edge.

The boundary condition $d = 1$ penalises only the positive / tensile part of the strain tensor ε since in Eq. (3.45) $\psi^d = \psi^d(\varepsilon_{ij}^+)$. Assuming $\psi^d = \psi^{\text{el}}$ and $\psi^i = 0$ in the left part of the plate yields different results, see the resulting force-displacement curve and the crack path in Figs. 3.16 and 3.17 which are in better agreement with the results for the discrete notch. However, a secondary crack starts to propagate at a certain stage in the loading process.

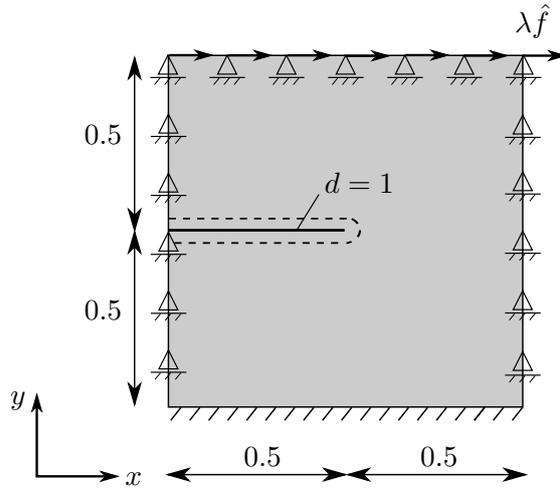


Fig. 3.13: Plate under shear (for small deformations), dimensions in mm; the initial notch is modelled with $d = 1$.

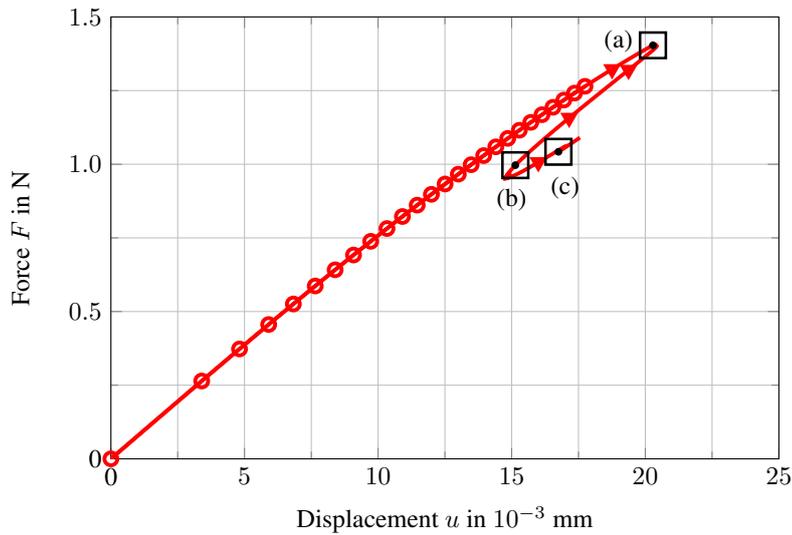


Fig. 3.14: Force-displacement curve for the plate under shear (assuming small deformations) when the initial notch modelled with $d = 1$; the mesh consists of 100×100 elements, so that the mesh size is $h = 0.01$ mm; the length scale parameter is $\ell = 0.01$ mm. Squares correspond to the phase field distributions for d in Fig. 3.15.

Moreover, it is not clear where $\psi^d = \psi^{el}$ and $\psi^i = 0$ should be prescribed since the crack is smeared and also extends into the right part of the plate.

A further calculation was carried out by prescribing $d = 0$ along the top left edge as suggested by AMOR *et al.* [3]. The resulting force-displacement curve and crack path are again different from the case where the notch is modelled in a discrete sense, see Figs. 3.18 and 3.19.

As in the previous case, a secondary crack emerges. Apparently, the phase field model for

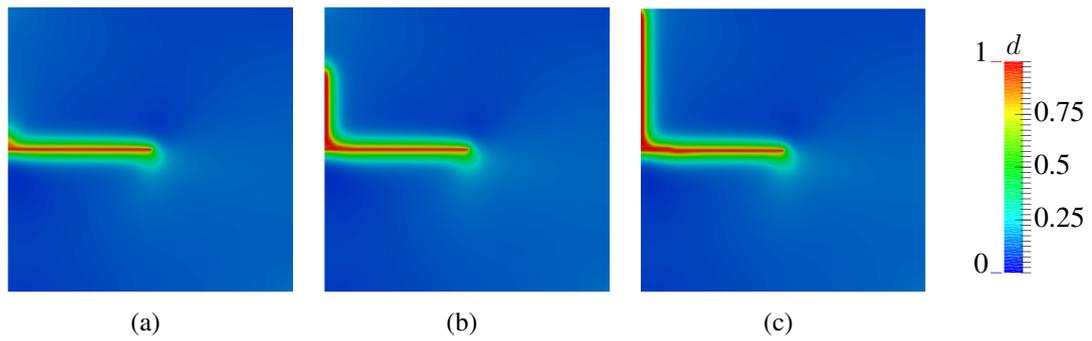


Fig. 3.15: Propagation of the variable d for the plate under shear (assuming small deformations) when the initial notch is modelled with $d=1$; plots correspond to the squares in Fig. 3.14.

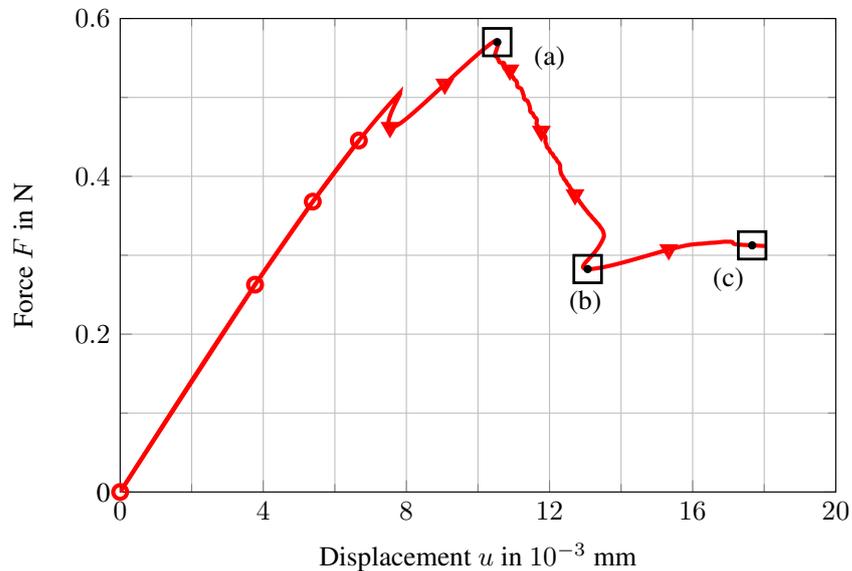


Fig. 3.16: Force-displacement curve for the plate under shear (assuming small deformations) when for the left half of the plate $\psi^d = \psi^{el}$ is set. The mesh consists of 100×100 elements, so that the mesh size is $h=0.01$ mm; the length scale parameter is taken as $\ell=0.01$ mm. Squares correspond to the phase field distributions for d in Fig. 3.17.

brittle fracture is sensitive to the applied boundary conditions.

3.3 Phase field model for cohesive fracture

A phase field model for cohesive fracture was introduced by VERHOOSSEL & DE BORST [117]. Apart from the phase field variable d , the introduction of a smeared jump v_i is necessary. Furthermore, the model requires an appropriate choice of the degree of the basis functions, i. e. linear for phase field d and smeared jump v_i , and cubic for displacement field u_i . Nevertheless, in the two-dimensional case, stress oscillations were

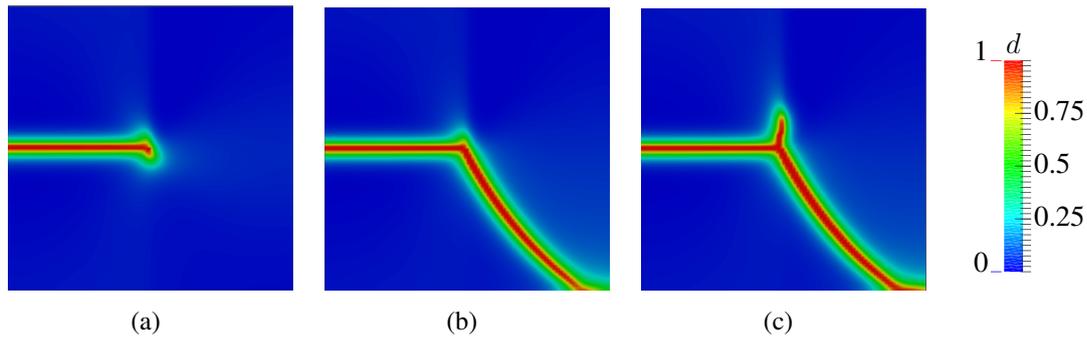


Fig. 3.17: Propagation of the variable d for the plate under shear (assuming small deformations); plots correspond to the squares in Fig. 3.16.

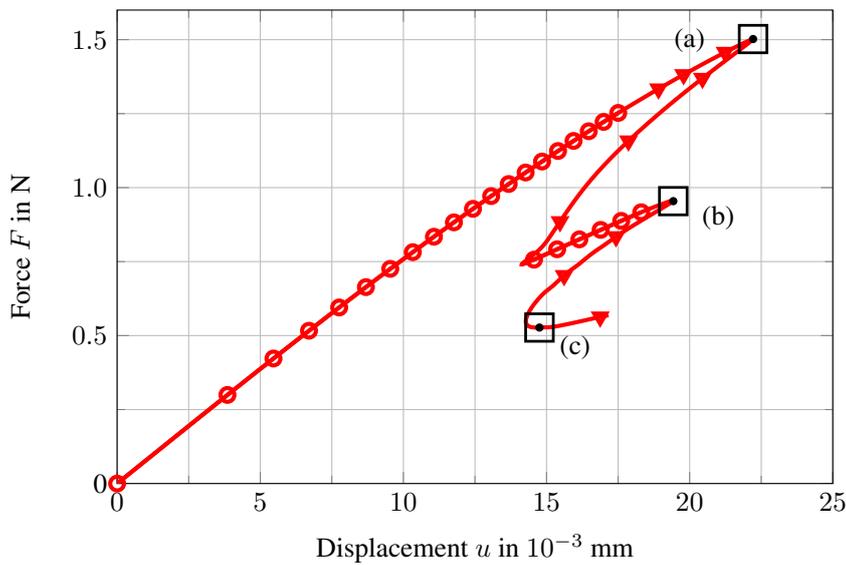


Fig. 3.18: Force-displacement curve for the plate under shear (assuming small deformations) when for the top left edge of the plate $d=0$ is prescribed. The mesh consists of 100×100 elements, so that the mesh size is $h=0.01$ mm; the length scale parameter is $\ell=0.01$ mm. Squares correspond to the phase field distributions for d in Fig. 3.19.

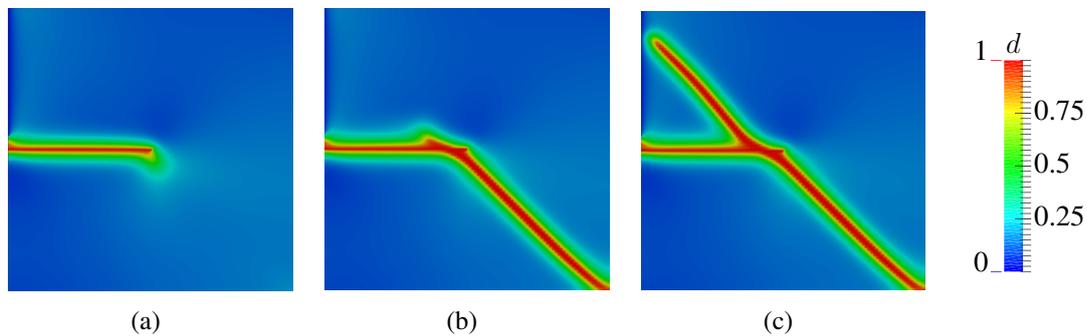


Fig. 3.19: Propagation of the variable d for the plate under shear (assuming small deformations). The plots correspond to the squares in Fig. 3.18.

observed for unstructured meshes by VIGNOLLET *et al.* [118]. This issues was then further pursued in MAY *et al.* [71] utilising a two-dimensional patch test for an unstructured mesh.

In this section, the model and its limitations are considered. After introducing the continuum formulation and deriving the finite element formulation, numerical examples are studied.

3.3.1 Continuum formulation

The following equation needs to be solved for the phase field problem

$$d - 4\ell^2 d_{,ii} = 0 \quad (3.47)$$

subject to the boundary conditions

$$d = 1 \quad \text{on } \Gamma, \quad d_{,i} n_i = 0 \quad \text{on } \partial\Omega. \quad (3.48)$$

This is different from the phase field model for brittle fracture in Section 3.2 where the partial differential equations for mechanical and phase field problem are linked, see Eqs. (3.22) and (3.23). Here, the phase field model for cohesive fracture is linked to the mechanical field problem via a set of nodes on the discrete crack surface Γ for which $d = 1$ is prescribed. Prescribing $d = 1$ at GAUSS points leads to $d > 1$ at the nodes as observed by VIGNOLLET *et al.* [118].

In the following, the link between mechanical and phase field problem for the cohesive case will be derived.

While for brittle fracture models the critical energy release rate \mathcal{G}_c is instantly dissipated, for cohesive models the energy release rate \mathcal{G} is released gradually and governed by the function

$$\mathcal{G} = \mathcal{G}(\llbracket \mathbf{u}_d \rrbracket). \quad (3.49)$$

The energy release rate \mathcal{G} depends on the crack opening in the local coordinate system $\llbracket \mathbf{u}_d \rrbracket$ at the interface and is equal to the critical energy release rate \mathcal{G}_c at full crack opening, cf. Fig. 3.20(a).

The energy release rate \mathcal{G} in Eq. (3.49) depends on the local crack opening in the normal and shear direction. In Fig. 3.20(a), it is assumed that \mathcal{G} depends only on the crack opening in the normal direction, i. e. only mode I is considered.

The crack mouth opening $\llbracket \mathbf{u}_d \rrbracket$ in the local coordinate system is obtained by a transfor-

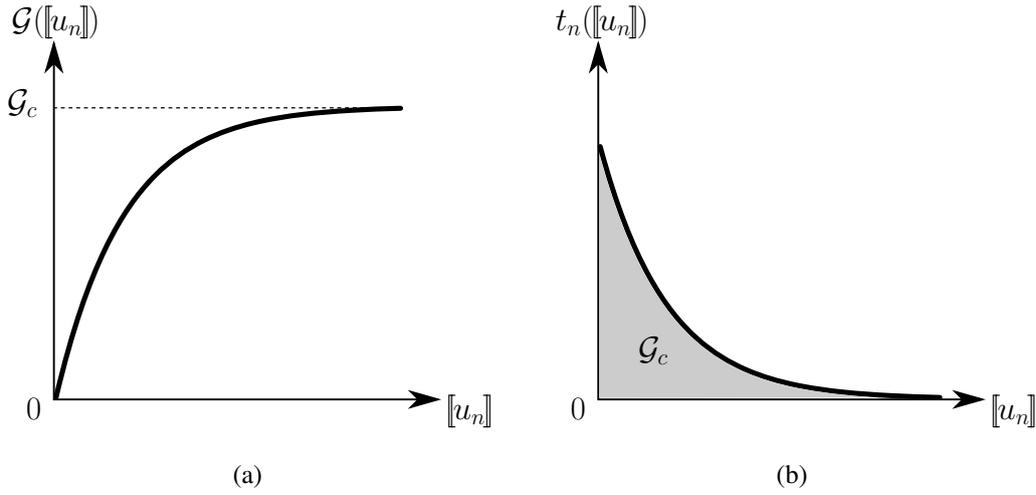


Fig. 3.20: (a) Energy release rate $\mathcal{G}(\llbracket u_n \rrbracket)$ and (b) normal traction $t_n(\llbracket u_n \rrbracket)$ in the cohesive zone for the one-dimensional case. The shaded grey area is equivalent to the critical energy release rate \mathcal{G}_c .

mation of the global crack mouth opening $\llbracket \underline{\mathbf{u}} \rrbracket$ as follows

$$\llbracket \underline{\mathbf{u}}_d \rrbracket = \underline{\underline{\mathbf{Q}}} \llbracket \underline{\mathbf{u}} \rrbracket, \quad (3.50)$$

with the orthogonal rotation matrix $\underline{\underline{\mathbf{Q}}}$, see also Section 6.2 for more details. The global traction $\underline{\mathbf{t}}$ in the cohesive zone is evaluated with the transformation

$$\underline{\mathbf{t}} = \underline{\underline{\mathbf{Q}}}^T \underline{\mathbf{t}}_d, \quad (3.51)$$

where $\underline{\mathbf{t}}_d$ is the cohesive traction in the local coordinate system. Both, $\underline{\mathbf{t}}_d$ and $\llbracket \underline{\mathbf{u}}_d \rrbracket$, consist of a normal and a shear component

$$\underline{\mathbf{t}}_d = \begin{bmatrix} t_n \\ t_s \end{bmatrix}, \quad \llbracket \underline{\mathbf{u}}_d \rrbracket = \begin{bmatrix} \llbracket u_n \rrbracket \\ \llbracket u_s \rrbracket \end{bmatrix}. \quad (3.52)$$

The local traction $\underline{\mathbf{t}}_d$ in the cohesive zone is computed via

$$t_{di}(\llbracket u_{dj} \rrbracket) = \frac{\partial \mathcal{G}(\llbracket u_{dj} \rrbracket)}{\partial \llbracket u_{di} \rrbracket}, \quad (3.53)$$

see Fig. 3.20(b).

In the phase field model for cohesive fracture, the crack is again smeared over the solid by employing Eqs. (3.5) and (3.7)

$$\int_{\Gamma} \mathcal{G}(\llbracket \underline{\mathbf{u}}_d \rrbracket) dA = \int_{\Omega} \mathcal{G}(\llbracket \underline{\mathbf{u}}_d \rrbracket) \gamma_{\ell} dV. \quad (3.54)$$

The smearing in Eq. (3.54) must not affect the term $\mathcal{G}(\llbracket \underline{\mathbf{u}}_d \rrbracket)$ in the direction normal to the crack since

$$\int_{\Gamma} \mathcal{B} \, dA = \int_{\Omega} \mathcal{B} \gamma_{\ell} \, dV \quad (3.55)$$

must hold for any quantity \mathcal{B} . The energy release rate $\mathcal{G}(\llbracket \underline{\mathbf{u}}_d \rrbracket)$ and the local crack mouth opening $\llbracket \underline{\mathbf{u}}_d \rrbracket$ exist in the discrete case only at the crack surface Γ . In the smeared case, $\mathcal{G}(\llbracket \underline{\mathbf{u}}_d \rrbracket)$ and $\llbracket \underline{\mathbf{u}}_d \rrbracket$ exist in the entire volume Ω while the value $\mathcal{G}(\llbracket \underline{\mathbf{u}}_d \rrbracket)$ itself must not change and has to remain constant in the direction normal to the crack. This can be achieved by replacing the local discrete crack mouth opening $\llbracket \underline{\mathbf{u}}_d \rrbracket$ with an auxiliary field – the smeared local crack mouth opening $\underline{\mathbf{v}}_d$ which exists in the entire volume Ω – and enforcing each component of the auxiliary field to remain constant in the direction normal to the crack $\underline{\mathbf{n}}_d$,

$$\frac{\partial \underline{\mathbf{v}}_d}{\partial \underline{\mathbf{n}}_d} = \underline{\mathbf{v}}_{d,j} n_{dj} = (v_{di})_{,j} n_{dj} = \frac{\partial v_{di}}{\partial x_j} n_{dj} = \underline{\mathbf{0}}. \quad (3.56)$$

The smeared local crack mouth opening $\underline{\mathbf{v}}_d$ is obtained from the smeared global crack mouth opening $\underline{\mathbf{v}}$ from the transformation (cf. Eq. (3.50))

$$\underline{\mathbf{v}}_d = \underline{\underline{\mathbf{Q}}} \underline{\mathbf{v}}. \quad (3.57)$$

$\underline{\mathbf{v}}_d$ being constant in the direction normal to the crack implies that $\mathcal{G}(\underline{\mathbf{v}}_d)$ is constant as well. The final expression for Eq. (3.54) reads

$$\int_{\Gamma} \mathcal{G}(\llbracket \underline{\mathbf{u}}_d \rrbracket) \, dA = \int_{\Omega} \mathcal{G}(\underline{\mathbf{v}}_d) \gamma_{\ell} \, dV \quad \text{subject to } \frac{\partial \underline{\mathbf{v}}_d}{\partial \underline{\mathbf{n}}_d} = \underline{\mathbf{0}}. \quad (3.58)$$

For the brittle case (see Section 3.2), \mathcal{G}_c is dissipated instantly and does not depend on the crack mouth opening $\llbracket \underline{\mathbf{u}}_d \rrbracket$. Therefore, no auxiliary field $\underline{\mathbf{v}}_d$ can be introduced for the phase field model for brittle fracture.

The phase field model for cohesive fracture assumes a split of the strain tensor into an elastic component and a component that accounts for damage

$$\varepsilon_{ij} = \varepsilon_{ij}^{\text{el}} + \varepsilon_{ij}^{\text{d}}. \quad (3.59)$$

Hence, the elastic energy density is now expressed as

$$\psi^{\text{el}} = \psi = \frac{1}{2} \lambda \varepsilon_{ii}^{\text{el}} \varepsilon_{jj}^{\text{el}} + \mu \varepsilon_{ij}^{\text{el}} \varepsilon_{ij}^{\text{el}}. \quad (3.60)$$

Notice the differences between the phase field model for brittle and cohesive fracture. For the phase field model for brittle fracture, $\psi^{\text{el}} \neq \psi$ and $\varepsilon_{ij}^{\text{el}} = \varepsilon_{ij}$ (Eqs. (3.9) and (3.12)), whereas for the phase field model for cohesive fracture $\psi^{\text{el}} = \psi$ and $\varepsilon_{ij}^{\text{el}} \neq \varepsilon_{ij}$ (Eqs. (3.59))

and (3.60)). The stress tensor σ_{ij} is computed with

$$\sigma_{ij} = \frac{\partial \psi^{\text{el}}}{\partial \varepsilon_{ij}^{\text{el}}} = \frac{\partial \psi^{\text{el}}}{\partial \varepsilon_{kl}} \frac{\partial \varepsilon_{kl}}{\partial \varepsilon_{ij}^{\text{el}}} = \frac{\partial \psi^{\text{el}}}{\partial \varepsilon_{kl}} \frac{\partial (\varepsilon_{kl}^{\text{el}} + \varepsilon_{kl}^{\text{d}})}{\partial \varepsilon_{ij}^{\text{el}}} = \frac{\partial \psi^{\text{el}}}{\partial \varepsilon_{kl}} \delta_{ki} \delta_{lj} = \frac{\partial \psi^{\text{el}}}{\partial \varepsilon_{ij}}, \quad (3.61)$$

or alternatively

$$\sigma_{ij} = C_{ijkl}(\varepsilon_{kl} - \varepsilon_{kl}^{\text{d}}) = C_{ijkl} \varepsilon_{kl}^{\text{el}}. \quad (3.62)$$

The tensor $\varepsilon_{ij}^{\text{d}}$ that accounts for damage can be derived from thermodynamical considerations. The second law of thermodynamics gives (see JIRÁSEK & BAŽANT [54, Chapter 23])

$$\dot{\mathcal{D}} = \sigma_{ij} \dot{\varepsilon}_{ij} - \dot{\psi} = \sigma_{ij} (\dot{\varepsilon}_{ij}^{\text{el}} + \dot{\varepsilon}_{ij}^{\text{d}}) - \frac{\partial \psi}{\partial \varepsilon_{ij}^{\text{el}}} \dot{\varepsilon}_{ij}^{\text{el}} = \sigma_{ij} (\dot{\varepsilon}_{ij}^{\text{el}} + \dot{\varepsilon}_{ij}^{\text{d}}) - \sigma_{ij} \dot{\varepsilon}_{ij}^{\text{el}} = \sigma_{ij} \dot{\varepsilon}_{ij}^{\text{d}} \geq 0. \quad (3.63)$$

The dissipation $\dot{\mathcal{D}}$ for the smeared form in Eq. (3.58) can be evaluated explicitly in the local coordinate system with

$$\dot{\mathcal{D}} = \frac{d}{dt} (\gamma_\ell(d) \mathcal{G}(v_{di})) = \mathcal{G} \frac{\partial \gamma_\ell}{\partial d} \dot{d} + \gamma_\ell \frac{\partial \mathcal{G}(v_{dj})}{\partial v_{di}} \dot{v}_{di} = \mathcal{G} \frac{\partial \gamma_\ell}{\partial d} \dot{d} + \gamma_\ell t_{di} \dot{v}_{di}. \quad (3.64)$$

The first term in Eq. (3.64) corresponds to the energy that is dissipated when advancing the cohesive zone by \dot{d} . Assuming that the smeared local jump v_{di} is initially zero in the newly created cohesive zone, the first term does not contribute to dissipation of energy since $\mathcal{G}(0) = 0$, Fig. 3.20(a). The second term in Eq. (3.64) represents the energy dissipation as the result of further crack opening by \dot{v}_{di} . Under these assumptions, the dissipation $\dot{\mathcal{D}}$ in Eq. (3.64) reads with the global quantities \underline{t} and \underline{v} exploiting Eqs. (3.51) and (3.57)

$$\dot{\mathcal{D}} = \gamma_\ell \underline{t}_d^T \dot{\underline{v}}_d = \gamma_\ell (\underline{\mathbf{Q}} \underline{t})^T \underline{\mathbf{Q}} \dot{\underline{v}} = \gamma_\ell \underline{t}^T \underline{\mathbf{Q}}^T \underline{\mathbf{Q}} \dot{\underline{v}} = \gamma_\ell \underline{t}^T \dot{\underline{v}} = \gamma_\ell t_i \dot{v}_i. \quad (3.65)$$

Equating Eq. (3.65) and Eq. (3.63),

$$\gamma_\ell t_i \dot{v}_i = \sigma_{ij} \gamma_\ell \text{sym}(\dot{v}_i n_{dj}) = \sigma_{ij} \dot{\varepsilon}_{ij}^{\text{d}}, \quad (3.66)$$

yields the contribution of the strain tensor $\varepsilon_{ij}^{\text{d}}$ accounting for damage

$$\varepsilon_{ij}^{\text{d}} = \gamma_\ell \text{sym}(v_i n_{dj}). \quad (3.67)$$

It has been taken into account in Eq. (3.66) that the traction t_i in the smeared crack zone Γ_ℓ is also distributed over the solid and therefore $t_i = \sigma_{ij} n_{dj}$. The potential of the phase

field model for cohesive fracture now reads

$$\Pi_\ell = \int_{\Omega} \psi^{\text{el}} + \mathcal{G}(v_i)\gamma_\ell + \frac{\alpha}{2} \left\| \frac{\partial v_i}{\partial x_j} n_{dj} \right\|^2 dV, \quad (3.68)$$

where the last term has been introduced in order to enforce the smeared jump \underline{v} to remain constant in the direction normal to the crack, cf. Eq. (3.56). Eq. (3.68) is written in terms of the global smeared jump \underline{v} . The local smeared jump \underline{v}_d can be evaluated utilising Eq. (3.57). Furthermore, \underline{v} being constant in the normal direction implies that \underline{v}_d is constant in the normal direction. The term γ_ℓ in Eq. (3.68) is computed from the solution of the phase field in Eq. (3.47).

Minimising Π_ℓ yields

$$\begin{aligned} \delta\Pi_\ell &= \frac{\partial\Pi_\ell}{\partial\varepsilon_{ij}}\delta\varepsilon_{ij} + \frac{\partial\Pi_\ell}{\partial v_i}\delta v_i + \frac{\partial\Pi_\ell}{\partial\left(\frac{\partial v_i}{\partial x_j}n_{dj}\right)}\delta\left(\frac{\partial v_i}{\partial x_j}n_{dj}\right) \\ &= \frac{\partial\psi^{\text{el}}}{\partial\varepsilon_{ij}}\delta\varepsilon_{ij} + \frac{\partial\psi^{\text{el}}}{\partial v_i}\delta v_i + \gamma_\ell\frac{\partial\mathcal{G}(v_j)}{\partial v_i}\delta v_i + \frac{\partial\Pi_\ell}{\partial\left(\frac{\partial v_i}{\partial x_j}n_{dj}\right)}\delta\left(\frac{\partial v_i}{\partial x_j}n_{dj}\right) = 0, \end{aligned} \quad (3.69)$$

and the following equations result

$$\sigma_{ij,i} = 0 \quad \text{in } \Omega, \quad (3.70)$$

$$\gamma_\ell[t_i(v_j) - \sigma_{ij}n_{dj}] = \alpha\frac{\partial^2 v_i}{\partial x_j \partial x_k}n_{dj}n_{dk} \quad \text{in } \Gamma_\ell \quad (3.71)$$

subject to the boundary conditions

$$\sigma_{ij}n_j = h_i \quad \text{on } \partial\Omega_h, \quad u_i = \bar{u}_i \quad \text{on } \partial\Omega_u, \quad \frac{\partial v_i}{\partial x_j}n_{dj} = 0 \quad \text{on } \partial\Gamma_\ell, \quad (3.72)$$

while Γ_ℓ denotes the domain where the phase field variable d is larger than a small tolerance that truncates the support of the smeared crack. \mathbf{h} is the prescribed surface traction, $\bar{\mathbf{u}}$ the prescribed displacement and \mathbf{n} the normal vector on the boundary. The boundary $\partial\Omega$ is decomposed into the parts $\partial\Omega_h$ and $\partial\Omega_u$ with $\partial\Omega_h \cap \partial\Omega_u = \emptyset$, $\partial\Omega_h \cup \partial\Omega_u = \partial\Omega$. Multiplying the strong forms of Eqs. (3.47), (3.70) and (3.71) by a test function δd , δu_j and δv_i , integrating over the domain Ω and Γ_ℓ , applying integration by parts and GAUSS'

theorem results in the following weak forms

$$\begin{aligned}
 \int_{\Omega} \delta d d + 4\ell^2 \delta d_{,i} d_{,i} dV &= 0, \\
 \int_{\Omega} \delta u_{i,j} \sigma_{ij} dV &= \int_{\partial\Omega_h} \delta u_i h_i dA, \\
 \int_{\Gamma_\ell} \gamma_\ell (\delta v_i t_i - \delta v_i \sigma_{ij} n_{dj}) + \alpha \frac{\partial v_i}{\partial x_j} n_j \frac{\partial v_i}{\partial x_k} n_k dV &= 0.
 \end{aligned} \tag{3.73}$$

3.3.2 Finite element formulation

Discretisation of the domain Ω into E elements,

$$\Omega = \bigcup_{e=1}^E \Omega^e, \tag{3.74}$$

and approximation of the phase field d , displacement field u_i and smeared global jump field v_i and their derivatives,

$$d^e = \underline{\mathbf{N}}_d^T \underline{\mathbf{d}}, \quad \delta d^e = \underline{\mathbf{N}}_d^T \delta \underline{\mathbf{d}}, \quad \underline{\mathbf{d}}_{,i}^e = \underline{\mathbf{B}}_d \underline{\mathbf{d}}, \quad \delta \underline{\mathbf{d}}_{,i}^e = \underline{\mathbf{B}}_d \delta \underline{\mathbf{d}}, \tag{3.75}$$

$$\underline{\mathbf{u}}^e = \underline{\mathbf{N}}_u \underline{\mathbf{u}}, \quad \delta \underline{\mathbf{u}}^e = \underline{\mathbf{N}}_u \delta \underline{\mathbf{u}}, \quad \underline{\boldsymbol{\varepsilon}}^e = \underline{\mathbf{B}}_u \underline{\mathbf{u}}, \quad \delta \underline{\boldsymbol{\varepsilon}}^e = \underline{\mathbf{B}}_u \delta \underline{\mathbf{u}}, \tag{3.76}$$

$$\underline{\mathbf{v}}^e = \underline{\mathbf{N}}_v \underline{\mathbf{v}}, \quad \delta \underline{\mathbf{v}}^e = \underline{\mathbf{N}}_v \delta \underline{\mathbf{v}}, \quad [\text{sym}(\underline{\mathbf{v}} \otimes \underline{\mathbf{n}}_d)]^e = \underline{\mathbf{B}}_v \underline{\mathbf{v}}, \tag{3.77}$$

$$[\text{sym}(\delta \underline{\mathbf{v}} \otimes \underline{\mathbf{n}}_d)]^e = \underline{\mathbf{B}}_v \delta \underline{\mathbf{v}}, \quad \left[\frac{\partial \underline{\mathbf{v}}}{\partial \underline{\mathbf{n}}_d} \right]^e = \underline{\mathbf{G}}_v \underline{\mathbf{v}}, \quad \left[\delta \frac{\partial \underline{\mathbf{v}}}{\partial \underline{\mathbf{n}}_d} \right]^e = \underline{\mathbf{G}}_v \delta \underline{\mathbf{v}}, \tag{3.78}$$

with

$$\begin{aligned}
 \underline{\mathbf{u}}^e &= \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix}, \quad \delta \underline{\mathbf{u}}^e = \begin{bmatrix} \delta u_1^e \\ \delta u_2^e \end{bmatrix}, \quad \underline{\boldsymbol{\varepsilon}}^e = \begin{bmatrix} \varepsilon_{11}^e \\ \varepsilon_{22}^e \\ 2\varepsilon_{12}^e \end{bmatrix}, \quad \delta \underline{\boldsymbol{\varepsilon}}^e = \begin{bmatrix} \delta \varepsilon_{11}^e \\ \delta \varepsilon_{22}^e \\ 2\delta \varepsilon_{12}^e \end{bmatrix}, \\
 \underline{\mathbf{d}}_{,i}^e &= \begin{bmatrix} d_{,1}^e \\ d_{,2}^e \end{bmatrix}, \quad \delta \underline{\mathbf{d}}_{,i}^e = \begin{bmatrix} \delta d_{,1}^e \\ \delta d_{,2}^e \end{bmatrix}, \quad \underline{\mathbf{v}}^e = \begin{bmatrix} v_1^e \\ v_2^e \end{bmatrix}, \quad \delta \underline{\mathbf{v}}^e = \begin{bmatrix} \delta v_1^e \\ \delta v_2^e \end{bmatrix}, \\
 [\text{sym}(\underline{\mathbf{v}} \otimes \underline{\mathbf{n}}_d)]^e &= \begin{bmatrix} v_1 n_{d1} \\ v_2 n_{d2} \\ v_1 n_{d2} + v_2 n_{d1} \end{bmatrix}, \quad [\text{sym}(\delta \underline{\mathbf{v}} \otimes \underline{\mathbf{n}}_d)]^e = \begin{bmatrix} \delta v_1 n_{d1} \\ \delta v_2 n_{d2} \\ \delta v_1 n_{d2} + \delta v_2 n_{d1} \end{bmatrix}, \\
 \left[\frac{\partial \underline{\mathbf{v}}}{\partial \underline{\mathbf{n}}_d} \right]^e &= \begin{bmatrix} v_{1,1} n_{d1} + v_{1,2} n_{d2} \\ v_{2,1} n_{d1} + v_{2,2} n_{d2} \end{bmatrix}, \quad \left[\delta \frac{\partial \underline{\mathbf{v}}}{\partial \underline{\mathbf{n}}_d} \right]^e = \begin{bmatrix} \delta v_{1,1} n_{d1} + \delta v_{1,2} n_{d2} \\ \delta v_{2,1} n_{d1} + \delta v_{2,2} n_{d2} \end{bmatrix}
 \end{aligned} \tag{3.79}$$

and

$$\begin{aligned}
 \underline{\underline{\mathbf{N}}}_u &= \begin{bmatrix} N_1 & 0 & N_2 & 0 & \dots \\ 0 & N_1 & 0 & N_2 & \dots \end{bmatrix}, \quad \underline{\underline{\mathbf{B}}}_u = \begin{bmatrix} N_{1,1} & 0 & N_{2,1} & 0 & \dots \\ 0 & N_{1,2} & 0 & N_{2,2} & \dots \\ N_{1,2} & N_{1,1} & N_{2,2} & N_{2,1} & \dots \end{bmatrix}, \\
 \underline{\underline{\mathbf{N}}}_d^T &= \begin{bmatrix} N_1 & N_2 & \dots \end{bmatrix}, \quad \underline{\underline{\mathbf{B}}}_d = \begin{bmatrix} N_{1,1} & N_{2,1} & \dots \\ N_{1,2} & N_{2,2} & \dots \end{bmatrix}, \\
 \underline{\underline{\mathbf{N}}}_v &= \begin{bmatrix} N_1 & 0 & N_2 & 0 & \dots \\ 0 & N_1 & 0 & N_2 & \dots \end{bmatrix}, \quad \underline{\underline{\mathbf{B}}}_v = \begin{bmatrix} n_{d1}N_1 & 0 & n_{d1}N_2 & 0 & \dots \\ 0 & n_{d2}N_1 & 0 & n_{d2}N_2 & \dots \\ n_{d2}N_1 & n_{d1}N_1 & n_{d2}N_2 & n_{d1}N_2 & \dots \end{bmatrix}, \\
 \underline{\underline{\mathbf{G}}}_v &= \begin{bmatrix} n_{d1}N_{1,1} + n_{d2}N_{1,2} & 0 & n_{d1}N_{2,1} + n_{d2}N_{2,2} & 0 & \dots \\ 0 & n_{d1}N_{1,1} + n_{d2}N_{1,2} & 0 & n_{d1}N_{2,1} + n_{d2}N_{2,2} & \dots \end{bmatrix}
 \end{aligned} \tag{3.80}$$

results in the following matrix-vector equation for Eq. (3.73)

$$\delta \underline{\underline{\mathbf{d}}}^T \int_{\Omega} \underbrace{(\underline{\underline{\mathbf{N}}}_d \underline{\underline{\mathbf{N}}}_d^T + 4\ell^2 \underline{\underline{\mathbf{B}}}_d^T \underline{\underline{\mathbf{B}}}_d)}_{\underline{\underline{\mathbf{f}}}_d^{\text{int}}(\underline{\underline{\mathbf{d}}})} \underline{\underline{\mathbf{d}}} \, dV = 0, \tag{3.81}$$

$$\delta \underline{\underline{\mathbf{u}}}^T \int_{\Omega} \underbrace{\underline{\underline{\mathbf{B}}}_u^T (\underline{\underline{\mathbf{C}}} \underline{\underline{\mathbf{B}}}_u \underline{\underline{\mathbf{u}}} - \gamma_{\ell} \underline{\underline{\mathbf{C}}} \underline{\underline{\mathbf{B}}}_v \underline{\underline{\mathbf{v}}})}_{\underline{\underline{\mathbf{f}}}_u^{\text{int}}(\underline{\underline{\mathbf{v}}}, \underline{\underline{\mathbf{u}}})} \, dV - \delta \underline{\underline{\mathbf{u}}}^T \int_{\Omega} \underbrace{\underline{\underline{\mathbf{N}}}_u^T \underline{\underline{\mathbf{h}}}}_{\underline{\underline{\mathbf{f}}}_u^{\text{ext}}} \, dA = 0, \tag{3.82}$$

$$\delta \underline{\underline{\mathbf{v}}}^T \int_{\Omega} \underbrace{-\gamma_{\ell} \underline{\underline{\mathbf{B}}}_v^T (\underline{\underline{\mathbf{C}}} \underline{\underline{\mathbf{B}}}_u \underline{\underline{\mathbf{u}}} - \gamma_{\ell} \underline{\underline{\mathbf{C}}} \underline{\underline{\mathbf{B}}}_v \underline{\underline{\mathbf{v}}}) + \gamma_{\ell} \underline{\underline{\mathbf{N}}}_v^T \underline{\underline{\mathbf{t}}} + \alpha \underline{\underline{\mathbf{G}}}_v^T \underline{\underline{\mathbf{G}}}_v \underline{\underline{\mathbf{v}}}}_{\underline{\underline{\mathbf{f}}}_v^{\text{int}}(\underline{\underline{\mathbf{v}}}, \underline{\underline{\mathbf{u}}})} \, dV = 0, \tag{3.83}$$

with the elasticity matrix $\underline{\underline{\mathbf{C}}}$. After solving Eq. (3.81) for arbitrary $\delta \underline{\underline{\mathbf{d}}}$, γ_{ℓ} can be computed for Eqs. (3.82) and (3.83). In the numerical cases studies of Section 3.3.3 it is sufficient to perform a single loading step under displacement control. Hence, no arc-length control is required. Considering that the system of Eqs. (3.82) and (3.83) must hold for any $\delta \underline{\underline{\mathbf{u}}}$ and $\delta \underline{\underline{\mathbf{v}}}$, and that the external forces vanish, $\underline{\underline{\mathbf{h}}} = \underline{\underline{\mathbf{0}}}$, yields

$$\underline{\underline{\mathbf{H}}}(\underline{\underline{\mathbf{v}}}, \underline{\underline{\mathbf{u}}}) = \begin{bmatrix} \underline{\underline{\mathbf{f}}}_v^{\text{int}}(\underline{\underline{\mathbf{v}}}, \underline{\underline{\mathbf{u}}}) \\ \underline{\underline{\mathbf{f}}}_u^{\text{int}}(\underline{\underline{\mathbf{v}}}, \underline{\underline{\mathbf{u}}}) \end{bmatrix} = \underline{\underline{\mathbf{0}}}. \tag{3.84}$$

Linearisation of Eq. (3.84) yields the solution for iteration $i+1$

$$\begin{bmatrix} \underline{\underline{\mathbf{v}}} \\ \underline{\underline{\mathbf{u}}} \end{bmatrix}^{i+1} = \begin{bmatrix} \underline{\underline{\mathbf{v}}} \\ \underline{\underline{\mathbf{u}}} \end{bmatrix}^i - \underline{\underline{\mathbf{K}}}_T^{-1} \Big|_i \cdot \begin{bmatrix} \underline{\underline{\mathbf{f}}}_v^{\text{int}}(\underline{\underline{\mathbf{v}}}, \underline{\underline{\mathbf{u}}}) \\ \underline{\underline{\mathbf{f}}}_u^{\text{int}}(\underline{\underline{\mathbf{v}}}, \underline{\underline{\mathbf{u}}}) \end{bmatrix}^i, \tag{3.85}$$

with the tangential stiffness matrix

$$\underline{\underline{\mathbf{K}}}_T(\underline{\mathbf{v}}, \underline{\mathbf{u}}) = \begin{bmatrix} \frac{\partial \underline{\mathbf{f}}_v^{\text{int}}(\underline{\mathbf{v}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{v}}} & \frac{\partial \underline{\mathbf{f}}_v^{\text{int}}(\underline{\mathbf{v}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}} \\ \frac{\partial \underline{\mathbf{f}}_u^{\text{int}}(\underline{\mathbf{v}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{v}}} & \frac{\partial \underline{\mathbf{f}}_u^{\text{int}}(\underline{\mathbf{v}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}} \end{bmatrix}. \quad (3.86)$$

3.3.3 Numerical examples

This section shows that an appropriate selection of the polynomial degree of the basis functions for displacement, smeared jump and phase field removes stress oscillations in the one-dimensional case. It will also be demonstrated that stress oscillations still occur for a two-dimensional unstructured mesh. Two possible causes for the stress oscillations are given.

3.3.3.1 One-dimensional bar

Consider the one-dimensional bar with an elastic interface in Fig. 3.21.

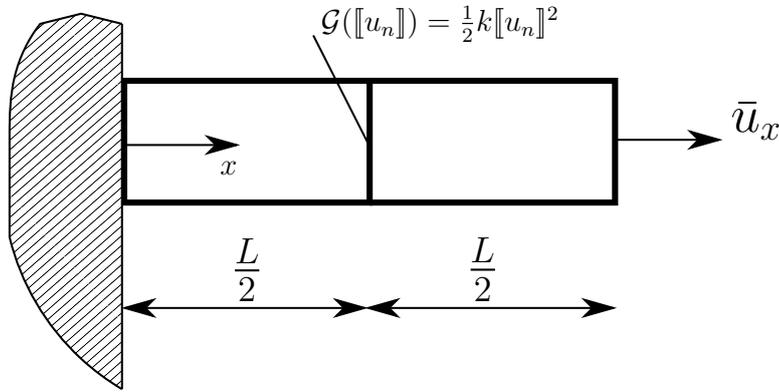


Fig. 3.21: Bar with an elastic interface $\mathcal{G} = \frac{1}{2}k[[u_n]] = \frac{1}{2}kv_n^2$ in the centre; with Eq. (3.53) the cohesive traction becomes $t_n = k[[u_n]] = kv_n$.

First, the bar is modelled with one-dimensional elements. The YOUNG's modulus is $E = 10$ MPa, the stiffness of the interface $k = 10$ MPa/mm, the length $L = 1$ mm and the length scale parameter $\ell = L/10$. The penalty parameter is taken as $\alpha = 1$ MPa. $d = 1$ is prescribed at the elastic interface, i. e. at the node in the centre of the bar. The bar consists of ten elements, i. e. five elements in each segment, so that the mesh size is $h = 0.1$ mm. The prescribed displacement at the right edge is $\bar{u}_x = 0.1$ mm.

Application of linear basis functions for the displacement u_x , the smeared jump v_x and

the phase field d results in stress oscillations, Fig. 3.22(a), as was also observed by VERHOESEL & DE BORST [117].

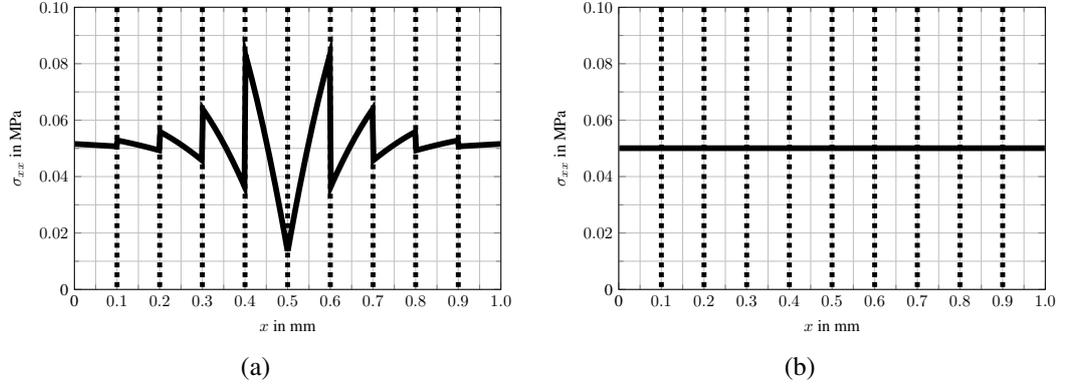


Fig. 3.22: Stress distribution along the bar in Fig. 3.21 for (a) linear basis functions for u_x , v_x , d and (b) cubic basis functions for u_x and linear basis functions for v_x , d ; the dashed lines mark element boundaries.

In one dimension, Eq. (3.59) can be rewritten using Eq. (3.5) as follows

$$\varepsilon_{xx}^{\text{el}} = \varepsilon_{xx} - \varepsilon_{xx}^{\text{d}} = \frac{du_x}{dx} - \gamma_\ell v_x = \frac{du_x}{dx} - \frac{1}{4\ell}(d^2 + 4\ell^2 d_{,x}^2)v_x. \quad (3.87)$$

Since v_x is enforced to be constant, the strain $\varepsilon_{xx}^{\text{d}}$ that accounts for damage has a quadratic distribution when linear basis functions are used for d . Therefore, the total strain ε_{xx} must have a quadratic distribution as well. This can be achieved when cubic basis functions are used for the displacement u_x . Fig. 3.22(b) shows that this is a successful remedy.

3.3.3.2 One-dimensional bar in a two-dimensional setting

Keeping the interpolation of the displacement of the third order while those for the phase field and the crack opening remain linear, the bar is now reconsidered in a two-dimensional setting with a POISSON'S ratio $\nu=0$ and by prescribing that all shear components of the smeared jump are zero, $v_s = v_y = 0$, see Fig. 3.23.

A structured mesh with 10×10 elements is used. The width is $c=1$ mm and $d=1$ is prescribed at all nodes for which $x=L/2$. The other parameters are the same as in the purely one-dimensional case. Along the line $y=0.51$ mm the same results are obtained for the stress distribution σ_{xx} as in case of the purely one-dimensional study, see Fig. 3.24. No stress oscillations are observed.

As a next step, some nodes are slightly displaced, cf. Fig. 3.25(a). Now, stress oscillations result along the line $y=0.51$ mm, see Fig. 3.25(b).

This is in agreement with the results obtained in VIGNOLLET *et al.* [118], where the use of

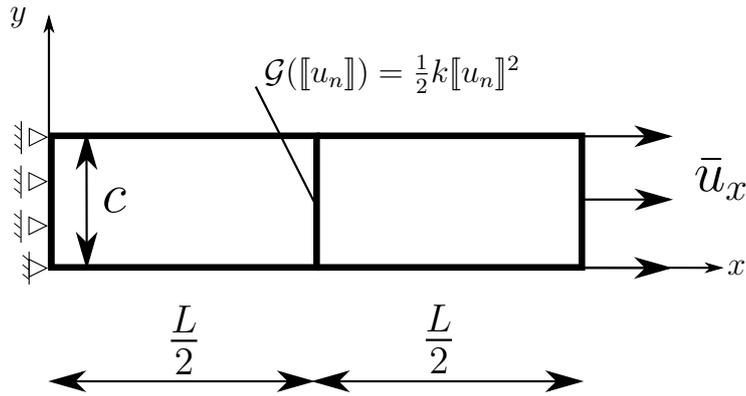


Fig. 3.23: Bar from Fig. 3.21 in a two-dimensional setting with $\nu=0$ and all $v_s=v_y=0$.

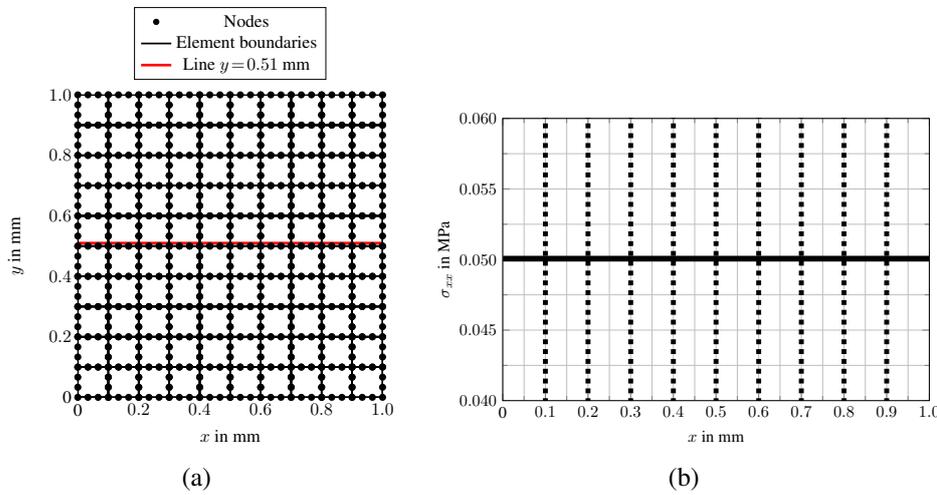


Fig. 3.24: For the structured mesh in (a) no stress oscillations are observed along the line $y=0.51$ mm in (b). Dashed lines correspond to element boundaries.

unstructured meshes for a peel test also resulted in stress oscillations. The present simulation can be considered as a patch test since a homogeneous stress state should be obtained when prescribing a uniform traction or displacement at the boundary, irrespective of the mesh lay-out. Unfortunately, this is not obtained for the present three-field formulation of the phase field model for cohesive fracture.

3.3.3.3 Causes for the stress oscillations

Next, two possible causes are given for the stress oscillations.

Contradiction in the continuum formulation

Consider the total strain ε_{xx} from Eq. (3.59) in a one-dimensional format

$$\varepsilon_{xx} = \frac{du_x}{dx} = \varepsilon_{xx}^{\text{el}} + \varepsilon_{xx}^{\text{d}} = \frac{du_x^{\text{el}}}{dx} + \gamma \ell v_x, \quad (3.88)$$

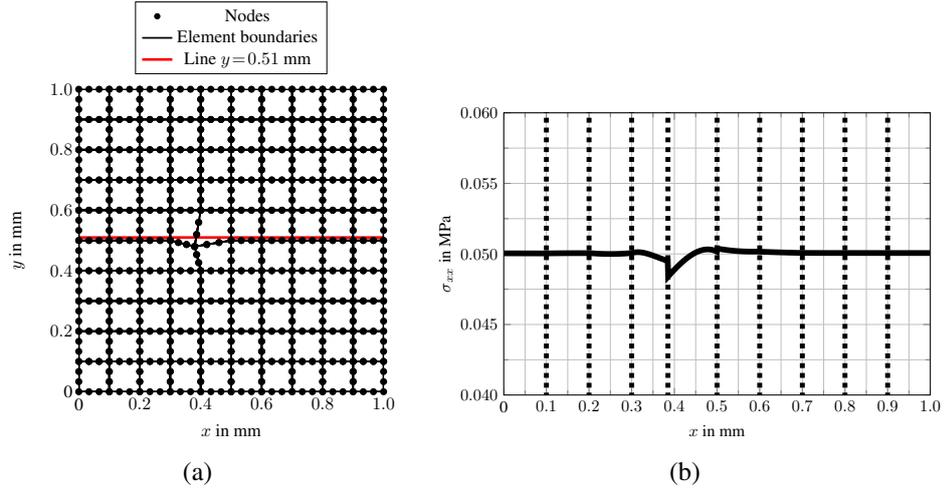


Fig. 3.25: For the unstructured mesh in (a) stress oscillations can be observed along the line $y = 0.51$ mm in (b). Dashed lines mark element boundaries.

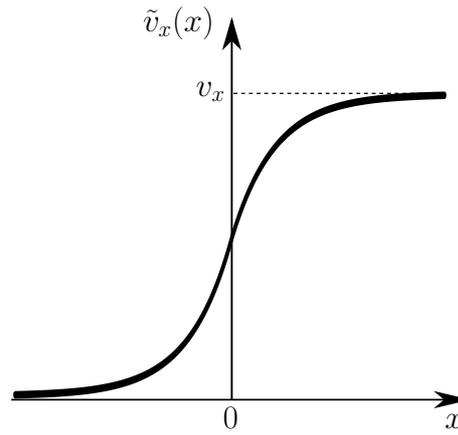


Fig. 3.26: Non-constant jump $\tilde{v}_x(x)$.

with the elastic displacement u_x^{el} . Integrating Eq. (3.88) with a constant smeared jump in the normal direction, $\frac{\partial v_x}{\partial x} = 0$, yields

$$u_x(x) = u_x^{\text{el}}(x) + v_x \underbrace{\int_{-\infty}^x \gamma_\ell(\tilde{x}) d\tilde{x}}_{\tilde{v}_x(x)}. \quad (3.89)$$

The integral in Eq. (3.89) can be interpreted as a smeared HEAVISIDE step function \mathcal{H} which is used in partition of unity approaches

$$u_x(x) = u_x^{\text{el}}(x) + \mathcal{H}v_x(x). \quad (3.90)$$

It has been derived in Eq. (3.58) that the smeared jump field v_x needs to be constant in the direction normal to the crack in order to have a constant $\mathcal{G}(v_x)$ in the normal crack direction. However, when the second term \tilde{v}_x in Eq. (3.89) is interpreted as the jump for the smeared model, it is observed that this term is not constant in the normal direction to the crack, see also Fig. 3.26. Hence, there are two different interpretations for the smeared jump in the phase field model for cohesive fracture, v_x and \tilde{v}_x . Both interpretations cannot hold simultaneously, and the phase field model for cohesive fracture seems to embody a contradiction.

Different polynomial orders

Another explanation for the oscillations may be the different orders of polynomials for the following terms. The distribution for the stress

$$\sigma_{xx} = E\varepsilon_{xx}^{\text{el}} \quad (3.91)$$

is quadratic when cubic basis functions are used for the displacement field u_i and linear basis functions for the smeared global jump v_i and the phase field d in Eq. (3.87). Since the cohesive traction t_n in the normal direction is constant,

$$t_n = k\llbracket u_n \rrbracket = kv_n, \quad (3.92)$$

due to a constant v_n , it may be that the different orders of approximation for t_n (constant) and for σ_{xx} (quadratic) in Eq. (3.71) contribute to stress oscillations.

4 T-splines

NURBS are commonly used in CAD. Since NURBS have a tensor product structure, refinement occurs globally. Furthermore, modelling watertight surfaces using NURBS can be difficult, if not impossible for certain cases. Also, several NURBS-patches join only with C_A^0 -continuity. T-splines, which can be seen as a generalisation of NURBS, were introduced by SEDERBERG *et al.* [99, 100] and do not suffer from the limitations that are inherent in NURBS. Local refinement is now possible and watertight surfaces can be created. Moreover, T-splines allow for the reduction of superfluous control points and a T-spline consists of a single patch. Use of T-spline blending functions as basis functions in a finite element context has been proposed by BAZILEVS *et al.* [9] and DÖRFEL *et al.* [36]. SEDERBERG *et al.* [100] used the term *blending* instead of *basis* function since it was not clear at that point whether T-spline blending functions are always linearly independent. Later, BUFFA *et al.* [20] demonstrated that the T-spline blending functions can be linearly dependent.

Using the BÉZIER extraction procedure introduced by SCOTT *et al.* [94], each blending function can be defined in a normalised fashion by a linear combination of BERNSTEIN polynomials. It was demonstrated by WANG *et al.* [120] that, if the BÉZIER extraction operator over all elements for all anchors (i. e. the global BÉZIER extraction operator) is linearly independent then the blending functions for all anchors are linearly independent as well. The BÉZIER extraction operator further allows to check the partition of unity and to refine T-splines locally, see MAY *et al.* [72].

Before discussing T-splines, the next section addresses the different definitions of the term continuity in design and analysis. Afterwards, a brief overview on the construction of the BÉZIER extraction operator for T-splines is given. Subsequently, linear independence and the partition of unity property of T-spline meshes are examined utilising the BÉZIER extraction operator. Then, a refinement method is proposed for T-spline meshes by adding anchors while the BÉZIER extraction operator is used for the determination of the nesting behaviour between two T-spline meshes. The capabilities of the method are demonstrated for T-spline meshes of an even and of an odd polynomial degree. Thereafter, hierarchical refinement for standard, semi-standard and non-standard T-spline meshes is introduced. Further, unstructured T-spline meshes of quadratic and cubic degree will be considered. It will be demonstrated how the BÉZIER extraction operator can be modified in order to obtain T-spline meshes that fulfil the partition of unity and are C_A^1 -continuous around an extraordinary point. Finally, analysis-suitable T-splines are discussed.

4.1 Continuity for design and analysis

In this section, the term *continuity* is elaborated from a design and from an analysis perspective.

4.1.1 Parametric continuity \mathcal{C}

Consider the two curve segments $\underline{\mathcal{S}}_1(t_1)$ and $\underline{\mathcal{S}}_2(t_2)$ depending on the parametric coordinates t_1 and t_2

$$\begin{aligned}\underline{\mathcal{S}}_1(t_1) &= (1 - t_1)\underline{\mathcal{P}}_1 + t_1\underline{\mathcal{P}}_2 \quad 0 \leq t_1 \leq 1, \\ \underline{\mathcal{S}}_2(t_2) &= (1 - t_2)\underline{\mathcal{P}}_2 + t_2\underline{\mathcal{P}}_3 \quad 0 \leq t_2 \leq 1,\end{aligned}\tag{4.1}$$

with the coordinates of the control points in the physical domain $\underline{\mathbf{x}} = (x, y)$

$$\underline{\mathcal{P}}_1 = (1, 1), \quad \underline{\mathcal{P}}_2 = (2, 2), \quad \underline{\mathcal{P}}_3 = (3, 3).\tag{4.2}$$

Both curves $\mathcal{S}_1(t_1)$ and $\mathcal{S}_2(t_2)$ are plotted in Fig. 4.1(a).

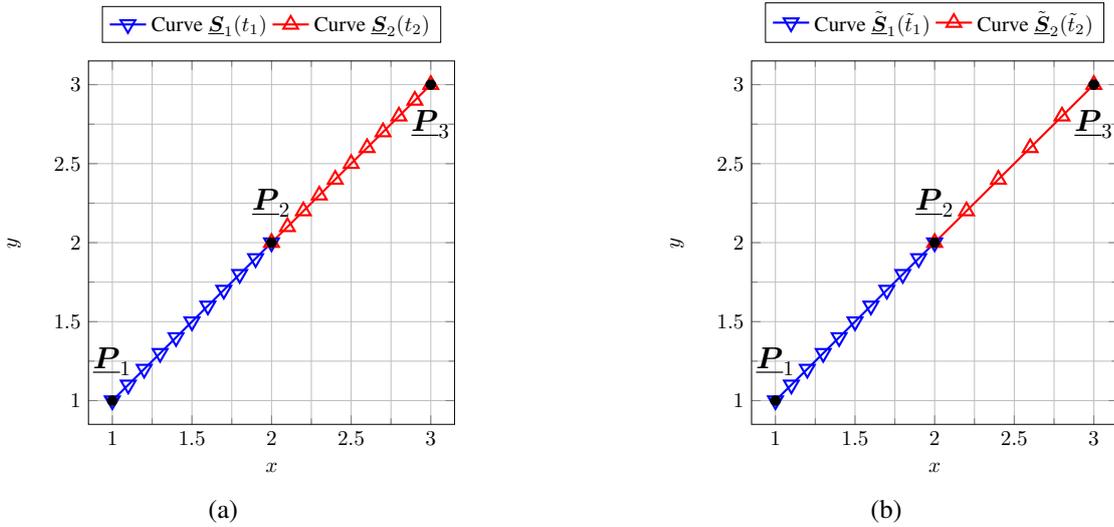


Fig. 4.1: Parametric curves in physical domain $\underline{\mathbf{x}} = (x, y)$, (a) $\mathcal{S}_1(t_1)$ and $\mathcal{S}_2(t_2)$ are *parametric* \mathcal{C}^1 -continuous at $\underline{\mathcal{P}}_2$; (b) $\tilde{\mathcal{S}}_1(\tilde{t}_1)$ and $\tilde{\mathcal{S}}_2(\tilde{t}_2)$ are *geometric* \mathcal{G}^1 -continuous at their joint $\underline{\mathcal{P}}_2$. The triangles mark isoparametric distances of $\Delta t_1 = \Delta t_2 = \Delta \tilde{t}_1 = \Delta \tilde{t}_2 = 0.1$.

Evaluating the first derivative with respect to the parametric coordinate yields for both curves

$$\left. \frac{\partial \underline{\mathcal{S}}_1(t_1)}{\partial t_1} \right|_{t_1=1} = \left. \frac{\partial \underline{\mathcal{S}}_2(t_2)}{\partial t_2} \right|_{t_2=0} = (1, 1).\tag{4.3}$$

Both curves have the same first derivative at their joint; the *parametric* continuity of the first derivative in Eq. (4.3) will be denoted as \mathcal{C}^1 .

4.1.2 Geometric continuity \mathcal{G}

As a next example, consider the two curve segments $\tilde{\mathbf{S}}_1(\tilde{t}_1)$ and $\tilde{\mathbf{S}}_2(\tilde{t}_2)$ depending on the parametric coordinates \tilde{t}_1 and \tilde{t}_2 with

$$\begin{aligned}\tilde{\mathbf{S}}_1(\tilde{t}_1) &= (1 - \tilde{t}_1)\mathbf{P}_1 + \tilde{t}_1\mathbf{P}_2 & 0 \leq \tilde{t}_1 \leq 1, \\ \tilde{\mathbf{S}}_2(\tilde{t}_2) &= (1 - 2\tilde{t}_2)\mathbf{P}_2 + 2\tilde{t}_2\mathbf{P}_3 & 0 \leq \tilde{t}_2 \leq \frac{1}{2}.\end{aligned}\quad (4.4)$$

The first derivative with respect to the parametric coordinate reads for both curves

$$(1, 1) = \left. \frac{\partial \tilde{\mathbf{S}}_1(\tilde{t}_1)}{\partial \tilde{t}_1} \right|_{\tilde{t}_1=1} \neq \left. \frac{\partial \tilde{\mathbf{S}}_2(\tilde{t}_2)}{\partial \tilde{t}_2} \right|_{\tilde{t}_2=0} = (2, 2). \quad (4.5)$$

Hence, both curves are not \mathcal{C}^1 -continuous. However, Fig. 4.1(b) reveals that both curves still have a continuous geometry in the physical domain \underline{x} . In order to account for this, the term *geometric* continuity was introduced by BARSKY & DEROSE [6, 7]. If two curve segments satisfy the condition

$$\left. \frac{\partial \tilde{\mathbf{S}}_1(\tilde{t}_1)}{\partial \tilde{t}_1} \right|_{\tilde{t}_1=1} = c \left. \frac{\partial \tilde{\mathbf{S}}_2(\tilde{t}_2)}{\partial \tilde{t}_2} \right|_{\tilde{t}_2=0}, \quad (4.6)$$

with a scalar $c > 0$, then they are called *geometric* continuous with respect to the first derivative. This continuity is denoted with \mathcal{G}^1 . The step from Eq. (4.4) to Eq. (4.1) – replacing \tilde{t}_2 with $\frac{t_2}{2}$ – is called reparameterisation. PETERS [84] pointed out that the definitions for parametric and geometric continuity in Eqs. (4.3) and (4.6) may not be clear for overlapping control points.

4.1.3 Continuity \mathcal{C}_A for analysis

Now, the term *continuity* is considered from an analysis perspective. Eq. (4.1) is equivalent to the parameterisation

$$\underline{\mathbf{S}}(t) = N_1(t)\mathbf{P}_1 + N_2(t)\mathbf{P}_2 + N_3(t)\mathbf{P}_3, \quad (4.7)$$

with $0 \leq t \leq 2$ and the LAGRANGIAN basis functions (see Fig. 4.2)

$$\begin{aligned}N_1(t) &= \begin{cases} 1 - t & \text{for } 0 \leq t \leq 1 \\ 0 & \text{for } 1 \leq t \leq 2 \end{cases}, & N_2(t) &= \begin{cases} t & \text{for } 0 \leq t \leq 1 \\ 2 - t & \text{for } 1 \leq t \leq 2 \end{cases}, \\ N_3(t) &= \begin{cases} 0 & \text{for } 0 \leq t \leq 1 \\ t - 1 & \text{for } 1 \leq t \leq 2 \end{cases}.\end{aligned}\quad (4.8)$$

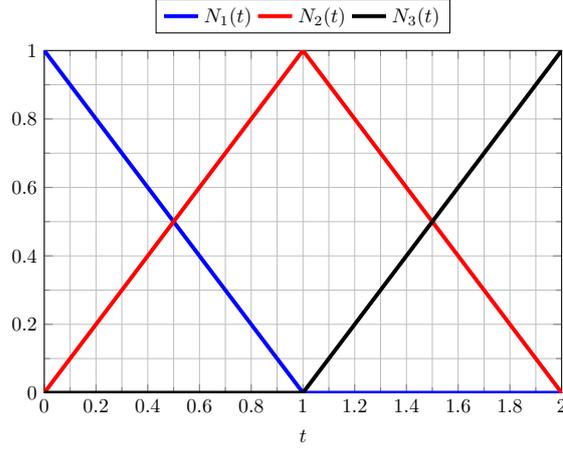


Fig. 4.2: \mathcal{C}_A^0 -continuous linear LAGRANGIAN basis functions.

The LAGRANGIAN basis functions in Eq. (4.8) have a continuity of \mathcal{C}_A^0 . In order to distinguish between the *parametric* continuity \mathcal{C} of the *curve* and the continuity \mathcal{C}_A of the *basis functions*, the subscript A (for analysis) was added. It follows that the continuity which is required for analysis needs to be distinguished from the parametric or geometric continuity which is interesting from a designer's point of view.

4.1.4 Geometric continuous basis functions

GROISSER & PETERS [46] showed that \mathcal{G}^k -continuous basis functions are \mathcal{C}_A^k -continuous. This will be illustrated for the case $(\mathcal{G}^1, \mathcal{C}_A^1)$ in one dimension. Consider the four basis functions N_i

$$\begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \end{bmatrix}, \quad (4.9)$$

with

$$\left. \begin{array}{l} B_1(\xi_1) = \frac{1}{4}(1 - \xi_1)^2 \\ B_2(\xi_1) = \frac{1}{2}(1 - \xi_1^2) \\ B_3(\xi_1) = \frac{1}{4}(1 + \xi_1)^2 \end{array} \right\} -1 \leq \xi_1 \leq 1, \quad \left. \begin{array}{l} B_4(\xi_2) = \frac{1}{16}(2 - \xi_2)^2 \\ B_5(\xi_2) = \frac{1}{8}(4 - \xi_2^2) \\ B_6(\xi_2) = \frac{1}{16}(2 + \xi_2)^2 \end{array} \right\} -2 \leq \xi_2 \leq 2. \quad (4.10)$$

The matrix in Eq. (4.9) contains the BÉZIER coefficients for the knot vector $\Xi = \{0, 0, 0, \frac{1}{2}, 1, 1, 1\}$. The following physical coordinate $P_i = x_i$ corresponds to each

basis function

$$P_1 = 0, \quad P_2 = 1, \quad P_3 = 2, \quad P_4 = 3. \quad (4.11)$$

The derivatives N_{i,ξ_1} and N_{i,ξ_2} with respect to the parameter coordinates ξ_1 and ξ_2 are plotted in Fig. 4.3(a) over the physical domain x for all four basis functions.

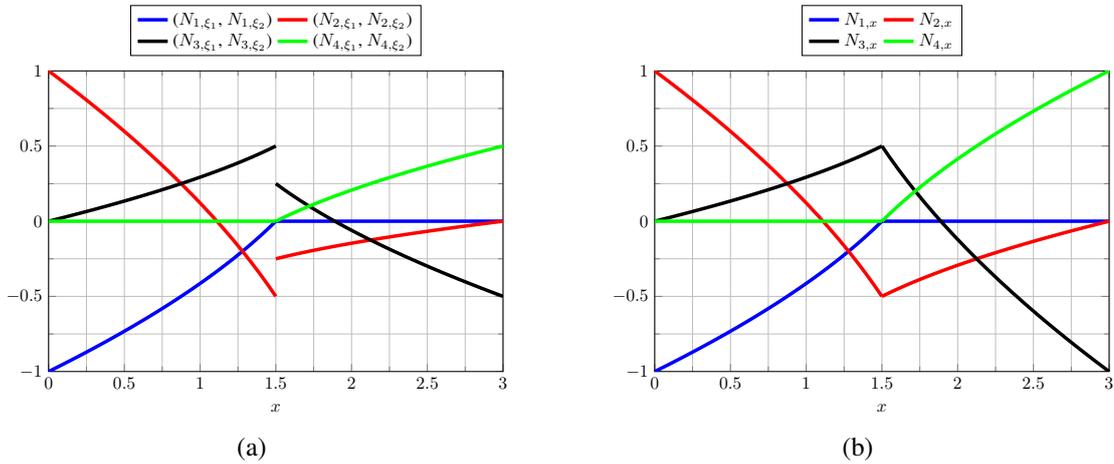


Fig. 4.3: (a) The derivatives $(N_{2,\xi_1}, N_{2,\xi_2})$ and $(N_{3,\xi_1}, N_{3,\xi_2})$ are discontinuous while (b) the derivatives $N_{2,x}$ and $N_{3,x}$ are continuous: \mathcal{G}^1 -continuous basis functions are also \mathcal{C}_A^1 -continuous.

It can be observed that $(N_{2,\xi_1}, N_{2,\xi_2})$ and $(N_{3,\xi_1}, N_{3,\xi_2})$ are discontinuous. However, since the \mathcal{G}^1 -continuity condition

$$\left. \frac{\partial N_i}{\partial \xi_1} \right|_{\xi_1=1} = 2 \left. \frac{\partial N_i}{\partial \xi_2} \right|_{\xi_2=-2} \quad (4.12)$$

holds for all four basis function, they have to be \mathcal{C}_A^1 -continuous: Fig. 4.3(b) shows the derivatives $N_{i,x}$ of all four basis functions with respect to the physical coordinate x . All four derivatives $N_{i,x}$ are now continuous. Thus, the four basis function N_i are \mathcal{C}_A^1 -continuous. The property that \mathcal{G}^1 -continuous basis functions are \mathcal{C}_A^1 -continuous is especially useful for connecting surfaces at an extraordinary point such that the basis functions possess the required \mathcal{C}_A^1 -continuity (see Section 4.7).

In sum, basis functions that fulfil a \mathcal{G}^1 -constraint are \mathcal{C}_A^1 -continuous and the geometry is \mathcal{G}^1 -continuous for any set of control points. If the basis functions are only \mathcal{C}_A^0 -continuous the geometry can still be \mathcal{G}^1 -continuous by an appropriate choice of the location of the coordinates in the physical domain. Such a $(\mathcal{G}^1, \mathcal{C}_A^0)$ -construction can also be used for solving fourth order partial differential equations when a rigid link between neighbouring control points along the \mathcal{C}_A^0 -continuity is introduced, see the bending strip method by KIENDL *et al.* [58]. This rigid link transfers the \mathcal{G}^1 -continuity from the coordinates to the displacement degrees of freedom.

4.2 Defining T-splines

This section provides a brief overview of T-splines. For a more elaborate demonstration of T-splines in a finite element environment reference is made to BAZILEVS *et al.* [9]. Index notation is adopted throughout with respect to a CARTESIAN frame.

4.2.1 Definition of the domains

In Fig. 4.4, the physical domain (x_ℓ), the parent domain ($\tilde{\xi}_\ell$), the index domain (u_ℓ), the parameter domain (ξ_ℓ^u), and the sub-parameter domain (ξ_ℓ) are shown for T-splines. Each element e can be mapped from the physical domain x_ℓ onto the parent domain $\tilde{\xi}_\ell \in [-1, 1]$, where GAUSSIAN integration can be carried out. The sub-parameter domain ξ_ℓ is obtained when only the unique values of the parameter domain ξ_ℓ^u are considered.

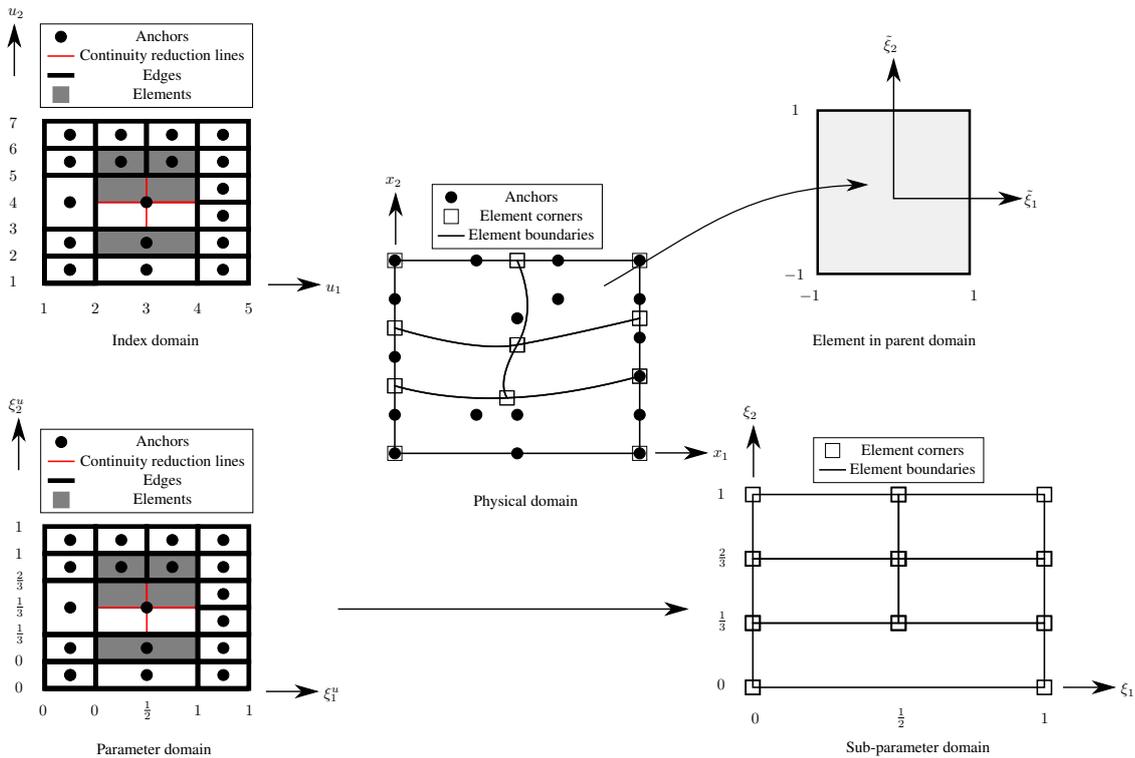


Fig. 4.4: Illustration of the physical domain (x_ℓ), the parent domain ($\tilde{\xi}_\ell$), the index domain (u_ℓ), the parameter domain (ξ_ℓ^u), and the sub-parameter domain (ξ_ℓ) for a quadratic T-spline mesh.

4.2.2 Definition of the local knot vector

The index domain in Fig. 4.4 represents a tiling of a region in \mathbb{R}^2 while all edges of each rectangle have a positive integer value. T-spline meshes of odd and of even polynomial order have to be treated differently when defining the local knot vectors from the parameter domain. The local knot vectors are necessary for defining the blending functions, see Section 4.2.3.

For a T-spline mesh of even degree p_ℓ in both directions, a so-called anchor – to which a single multivariate blending function is attached – is placed in the centre of each rectangle, see the quadratic T-spline mesh in Fig. 4.5(a). A local knot vector for a T-spline mesh of even degree is obtained from the parameter domain by – starting at the anchor – marching horizontally (both left and right) and vertically (both up and down), until a number of $p_\ell/2+1$ edges are crossed in all four directions, thus giving a vector length of $p_\ell+2$. Every time an edge is crossed, the corresponding parameter value is added to the local knot vector. If fewer than $p_\ell/2+1$ edges are crossed, and there are no more edges left to be crossed, the parameter value that has been added last is repeated until $p_\ell/2+1$ parameter values are added in this direction. For the blue anchor A sitting at $(3.5, 5.5)$ in the index domain in Fig. 4.5(a), the local knot vectors are $\Xi_1^A = \{0, \frac{1}{2}, 1, 1\}$ and $\Xi_2^A = \{\frac{1}{3}, \frac{2}{3}, 1, 1\}$, respectively.

For a T-spline mesh of odd degree p_ℓ in both directions, anchors are located at the vertices of the rectangles, see the cubic T-spline mesh in Fig. 4.5(b). In order to obtain the local knot vector of an anchor, the parameter ξ_ℓ^u at the vertex is added to the local knot vectors for each direction. Afterwards, marching (starting at the location of the anchor) horizontally to the right and left, and vertically up and down, until $(p_\ell+1)/2$ edges have been crossed in all four directions, yields again a local knot vector of length $p_\ell+2$. If there are no more edges to be crossed, then the value of the last added parameter is repeated until $(p_\ell+1)/2$ values are added in this direction to the local knot vector. Consider, for instance, the blue anchor B sitting at $(2, 2)$ in the index domain for the cubic T-spline mesh in Fig. 4.5(b). The local knot vectors are $\Xi_1^B = \{0, 0, 0, 1, 1\}$ and $\Xi_2^B = \{0, 0, 0, \frac{1}{3}, \frac{2}{3}\}$.

4.2.3 Construction of the blending functions

Consider a T-spline mesh with n anchors. Each anchor i is equipped with a single multivariate blending function N^i . Each multivariate blending function N^i is defined in the sub-parameter domain ξ_ℓ as follows

$$N^i(\underline{\xi}) = \prod_{\ell=1}^d N_\ell^i(\xi_\ell), \quad (4.13)$$

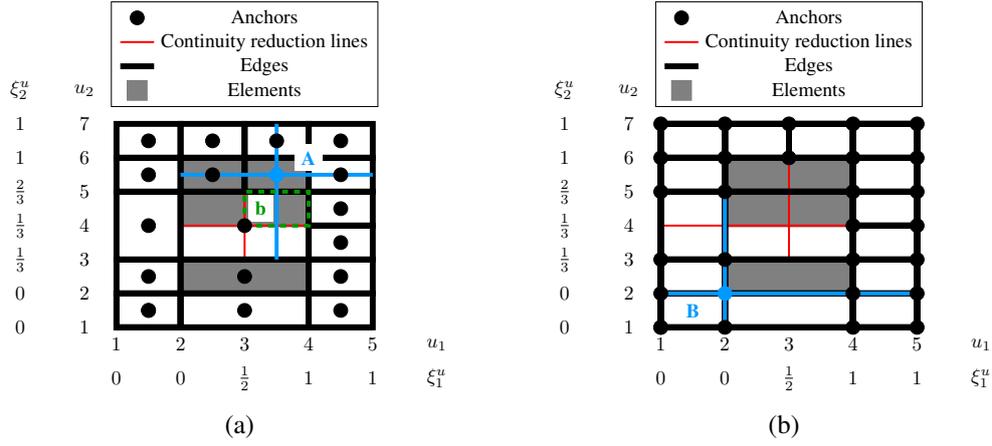


Fig. 4.5: Determination of the local knot vectors for a T-spline mesh of (a) even (quadratic, $p_\ell=2$) and (b) odd (cubic, $p_\ell=3$) degree: every time an edge is crossed in all four directions, the corresponding parameter value is added to the local knot vector. (a) The local knot vectors for the blue anchor A are $\Xi_1^A = \{\xi_1^2, \xi_1^3, \xi_1^4, \xi_1^5\} = \{0, \frac{1}{2}, 1, 1\}$ and $\Xi_2^A = \{\xi_2^3, \xi_2^5, \xi_2^6, \xi_2^7\} = \{\frac{1}{3}, \frac{2}{3}, 1, 1\}$, (b) The local knot vectors for the blue anchor B are $\Xi_1^B = \{\xi_1^1, \xi_1^2, \xi_1^3, \xi_1^4, \xi_1^5\} = \{0, 0, 0, 1, 1\}$ and $\Xi_2^B = \{\xi_2^1, \xi_2^2, \xi_2^3, \xi_2^4, \xi_2^5\} = \{0, 0, 0, \frac{1}{3}, \frac{2}{3}\}$.

with the univariate blending functions N_ℓ^i for each anchor i and the dimension d . The univariate blending function N_ℓ^i of order p_ℓ for anchor i is given by

$$N_\ell^i(\xi_\ell) = N_{\ell,1,p_\ell}^i(\xi_\ell), \quad (4.14)$$

where the N_{ℓ,a,p_ℓ}^i (with $a=1$ the single blending function for anchor i is obtained) can be defined with the local knot vector $\Xi_\ell^i = \{\xi_{\ell,1}^i, \xi_{\ell,2}^i, \dots, \xi_{\ell,p_\ell+2}^i\}$ of anchor i for $p_\ell=0$ with

$$N_{\ell,a,0}^i(\xi_\ell) = \begin{cases} 1 & \text{if } \xi_{\ell,a}^i \leq \xi_\ell < \xi_{\ell,a+1}^i \\ 0 & \text{otherwise} \end{cases}. \quad (4.15)$$

For $p_\ell \geq 1$ they are given by the COX - DE BOOR [25, 27] recursion formula

$$N_{\ell,a,p_\ell}^i(\xi_\ell) = \frac{\xi_\ell - \xi_{\ell,a}^i}{\xi_{\ell,a+p_\ell}^i - \xi_{\ell,a}^i} N_{\ell,a,p_\ell-1}^i(\xi_\ell) + \frac{\xi_{\ell,a+p_\ell+1}^i - \xi_\ell}{\xi_{\ell,a+p_\ell+1}^i - \xi_{\ell,a+1}^i} N_{\ell,a+1,p_\ell-1}^i(\xi_\ell). \quad (4.16)$$

Herein, only cases will be considered with an equal polynomial order p_ℓ in the ξ_1 -direction and the ξ_2 -direction.

4.2.4 Element definition

The red anchor A with index coordinates (3.5, 5.5) for the quadratic T-spline mesh in Fig. 4.6(a) has the local knot vectors $\Xi_1^A = \{\xi_1^2, \xi_1^3, \xi_1^4, \xi_1^5\}$ and $\Xi_2^A = \{\xi_2^3, \xi_2^5, \xi_2^6, \xi_2^7\}$. An-

chor A has a non-zero blending function in the green parameter domain $[\xi_1^2, \xi_1^5] \times [\xi_2^3, \xi_2^7]$. Within this domain, the net of red dashed lines depicted in Fig. 4.6(a) is obtained upon drawing all the values contained in the local knot vectors Ξ_ℓ^A . Along those lines, the blending function of anchor A has a reduced continuity, which is indicated by a multiplicity larger than zero in the local knot vectors. If one of these lines is not already an edge, this line is added to the T-spline mesh, see Fig. 4.6(b). The added line is called a continuity reduction line. For T-splines, elements are defined by the union of all edges and continuity reduction lines with non-zero parametric area in the parameter space ξ_ℓ^u , see also Fig. 4.5(a).

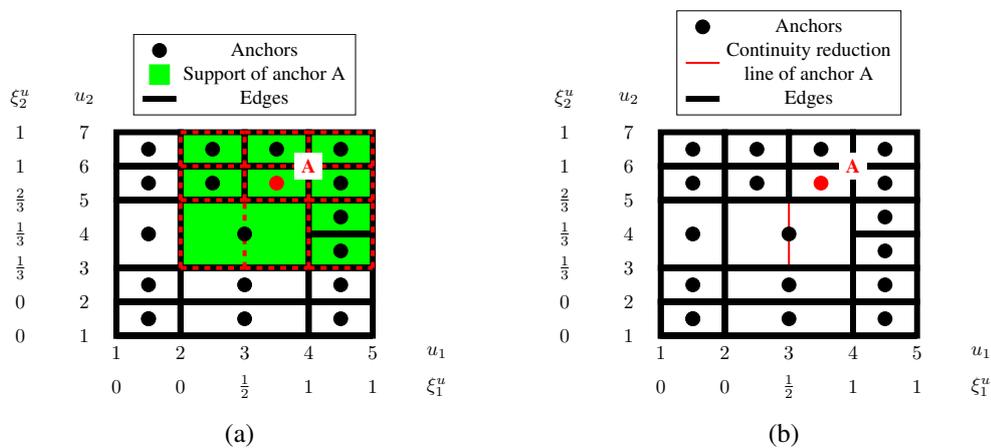


Fig. 4.6: Continuity reduction lines: consider the red anchor A with index coordinates $(3.5, 5.5)$. (a) This anchor has a support (non-zero blending functions) in the green shaded domain $[\xi_1^2, \xi_1^5] \times [\xi_2^3, \xi_2^7]$. Drawing all the values contained in the local knot vectors $\Xi_1^A = \{\xi_1^2, \xi_1^3, \xi_1^4, \xi_1^5\}$ and $\Xi_2^A = \{\xi_2^3, \xi_2^5, \xi_2^6, \xi_2^7\}$ gives the net of dashed red lines. (b) If a red dashed line in (a) is not already an edge then it is added to the T-spline mesh.

4.3 BÉZIER extraction for T-splines

For details on the BÉZIER extraction method for T-splines, reference is made to SCOTT *et al.* [94]. Here, a succinct summary on the calculation of the BÉZIER extraction operator is given and the method is illustrated by means of an example.

Suppose that the domain is divided into $e = 1 \dots E$ elements. Then, the blending function N_e^i of anchor i over element e can be written as a linear combination of the BERNSTEIN polynomials

$$N_e^i(\underline{\xi}) = \underline{C}_e^i \underline{B}_e(\underline{\xi}), \quad (4.17)$$

where the $(p_\ell+1)^2$ bivariate BERNSTEIN polynomials \underline{B}_e for element e in the two-dimensional case are expressed as follows

$$\underline{B}_e(\underline{\xi}) = \begin{bmatrix} B_{1e}^1(\xi_1)B_{2e}^1(\xi_2) \\ \vdots \\ B_{1e}^{p_\ell+1}(\xi_1)B_{2e}^1(\xi_2) \\ \vdots \\ B_{1e}^{p_\ell+1}(\xi_1)B_{2e}^{p_\ell+1}(\xi_2) \end{bmatrix}. \quad (4.18)$$

The bivariate BERNSTEIN polynomials \underline{B}_e are equal for each element e in the parent domain $\tilde{\xi}_\ell$. A univariate blending function $N_{\ell e}^i$ of anchor i over element e can be expressed in terms of the univariate BERNSTEIN basis $B_{\ell e}^a$ with

$$N_{\ell e}^i(\xi_\ell) = \begin{bmatrix} C_{\ell e}^{i1} & \dots & C_{\ell e}^{ip_\ell+1} \end{bmatrix} \begin{bmatrix} B_{\ell e}^1(\xi_\ell) \\ \vdots \\ B_{\ell e}^{p_\ell+1}(\xi_\ell) \end{bmatrix}, \quad (4.19)$$

where $C_{\ell e}^{ia}$ are the coefficients for anchor i and element e corresponding to $B_{\ell e}^a$. The $a=1 \dots p_\ell+1$ univariate BERNSTEIN polynomials $B_{\ell e}^a$ of order p_ℓ are defined over the interval $\tilde{\xi}_\ell \in [-1, 1]$ by

$$B_{\ell e}^a(\tilde{\xi}_\ell) = \frac{1}{2^{p_\ell}} \binom{p_\ell}{a-1} (1 - \tilde{\xi}_\ell)^{p_\ell-(a-1)} (1 + \tilde{\xi}_\ell)^{a-1}. \quad (4.20)$$

The univariate BERNSTEIN polynomials $B_{\ell e}^a$ are equal in the parent domain $\tilde{\xi}_\ell$ for each element e in each direction. In Eq. (4.17), \underline{C}_e^i is the BÉZIER extraction operator of anchor i with support over element e

$$\underline{C}_e^i = \begin{bmatrix} C_{1e}^{i1}C_{2e}^{i1} \\ \vdots \\ C_{1e}^{ip_\ell+1}C_{2e}^{i1} \\ \vdots \\ C_{1e}^{ip_\ell+1}C_{2e}^{ip_\ell+1} \end{bmatrix}. \quad (4.21)$$

In order to illustrate the notation, consider again the anchor A at (3.5, 5.5) in Fig. 4.5(a). The local knot vectors are $\Xi_1^A = \{0, \frac{1}{2}, 1, 1\}$ and $\Xi_2^A = \{\frac{1}{3}, \frac{2}{3}, 1, 1\}$ for the ξ_1 -direction and the ξ_2 -direction, respectively. Next, the BÉZIER extraction operator over the element b in Fig. 4.5(a) with range $[3, 4] \times [4, 5]$ in the index domain is evaluated for anchor A. The range for the element b is $[\xi_1^3, \xi_1^4] \times [\xi_2^4, \xi_2^5]$ in the parameter domain and $[\frac{1}{2}, 1] \times [\frac{1}{3}, \frac{2}{3}]$ in

the sub-parameter domain. In Fig. 4.7, the blending functions N_{ℓ}^A are shown for each direction in the sub-parameter domain ξ_{ℓ} . The part of the blending functions N_{ℓ}^A which has a support over element b with range $[\frac{1}{2}, 1] \times [\frac{1}{3}, \frac{2}{3}]$ in the sub-parameter domain ξ_{ℓ} – i. e. $N_{\ell b}^A$ – has been plotted with a solid black line. Expressing the blending functions $N_{\ell b}^A$ of anchor A with support over element b for each direction ξ_{ℓ} in terms of the BERNSTEIN basis $B_{\ell b}^a$ of element b, gives for the BÉZIER extraction operator in each direction

$$N_{1b}^A = \begin{bmatrix} C_{1b}^{A1} & C_{1b}^{A2} & C_{1b}^{A3} \end{bmatrix} \begin{bmatrix} B_{1b}^1 \\ B_{1b}^2 \\ B_{1b}^3 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 1 & 0 \end{bmatrix} \begin{bmatrix} B_{1b}^1 \\ B_{1b}^2 \\ B_{1b}^3 \end{bmatrix}, \quad (4.22)$$

$$N_{2b}^A = \begin{bmatrix} C_{2b}^{A1} & C_{2b}^{A2} & C_{2b}^{A3} \end{bmatrix} \begin{bmatrix} B_{2b}^1 \\ B_{2b}^2 \\ B_{2b}^3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} B_{2b}^1 \\ B_{2b}^2 \\ B_{2b}^3 \end{bmatrix}. \quad (4.23)$$

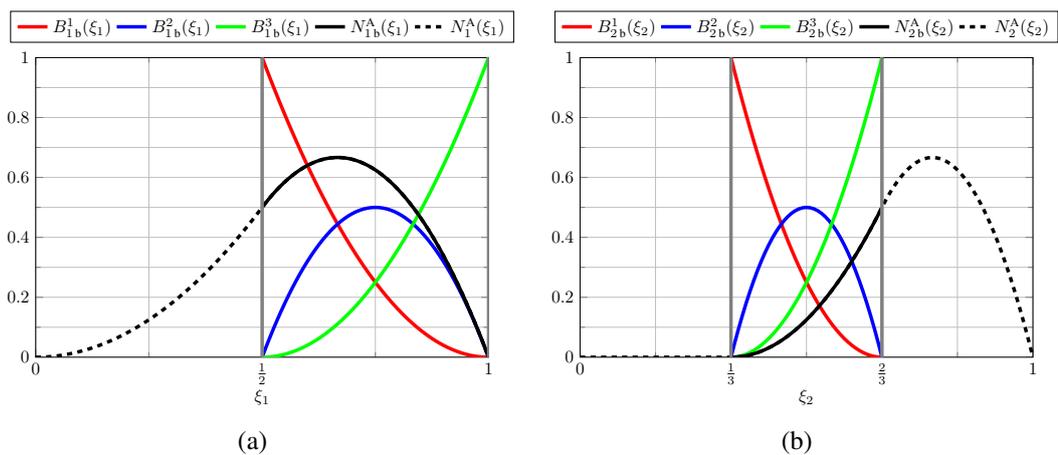


Fig. 4.7: Illustration of the blending functions N_{ℓ}^A for anchor A and the BERNSTEIN polynomials $B_{\ell b}^a$ for element b in Fig. 4.5(a) over the sub-parameter domain in (a) the ξ_1 -direction and (b) the ξ_2 -direction.

Now, using Eq. (4.21) and combining the unidirectional BÉZIER extraction operators defined in Eqs. (4.22) and (4.23), the BÉZIER extraction operator \underline{C}_b^A for the anchor A with support over element b, see Fig. 4.5(a), reads

$$\underline{C}_b^A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{2} & 0 \end{bmatrix}^T. \quad (4.24)$$

If, for an anchor i , this procedure is applied to all elements E , then the BÉZIER extraction

operator for anchor i is obtained

$$\underline{\underline{C}}^i = \begin{bmatrix} \underline{C}_1^i \\ \vdots \\ \underline{C}_E^i \end{bmatrix}. \quad (4.25)$$

The BÉZIER extraction operator for all n anchors is then given by

$$\underline{\underline{C}} = \begin{bmatrix} \underline{C}^{1T} \\ \vdots \\ \underline{C}^{nT} \end{bmatrix}. \quad (4.26)$$

$\underline{\underline{C}}$ is called global BÉZIER extraction operator. Hence, the vector with all n blending functions

$$\underline{N}(\underline{\xi}) = \begin{bmatrix} N^1(\underline{\xi}) \\ \vdots \\ N^n(\underline{\xi}) \end{bmatrix} \quad (4.27)$$

can be written as

$$\underline{N}(\underline{\xi}) = \underline{\underline{C}} \underline{B}(\underline{\xi}), \quad (4.28)$$

where \underline{B} is the vector which contains the elemental BERNSTEIN polynomials \underline{B}_e

$$\underline{B}(\underline{\xi}) = \begin{bmatrix} \underline{B}_1(\underline{\xi}) \\ \vdots \\ \underline{B}_E(\underline{\xi}) \end{bmatrix}. \quad (4.29)$$

A single blending function N^i can be expressed as

$$N^i(\underline{\xi}) = \underline{C}^{iT} \underline{B}(\underline{\xi}). \quad (4.30)$$

The blending functions \underline{N}_e with support in element e are determined by

$$\underline{N}_e(\underline{\xi}) = \underline{C}_e \underline{B}_e(\underline{\xi}), \quad (4.31)$$

with the elemental BÉZIER extraction operator \underline{C}_e .

4.4 Classification of T-splines

In this section, T-spline meshes are classified according to the linear dependencies exhibited by their blending functions. The partition of unity property is also investigated. The

classification methods in this section can be applied to T-spline meshes of arbitrary degree, T-spline meshes with extraordinary points and three-dimensional T-spline meshes: only the BÉZIER extraction operator is required.

4.4.1 Classification of T-splines according to the type of linear dependence

In the following, the BÉZIER extraction operator is used to gather meshes into three categories based on the type of linear dependence of their blending functions:

- globally linearly independent,
- locally linearly independent with a non-square matrix $\underline{\underline{C}}_e$,
- locally linearly independent with a square matrix $\underline{\underline{C}}_e$.

4.4.1.1 Global linear independence

A T-spline mesh with n anchors has globally linearly independent blending functions if and only if the solution for

$$\sum_{i=1}^n \alpha^i N^i(\underline{\xi}) = 0 \quad (4.32)$$

is $\alpha^i = 0$ for $i = 1 \dots n$. Recall, that each blending function N^i of anchor i can be expressed using the BÉZIER extraction operator. Substituting Eq. (4.30) into Eq. (4.32) leads to

$$\sum_{i=1}^n \alpha^i \underline{C}^{iT} \underline{B}(\underline{\xi}) = 0. \quad (4.33)$$

Since the BERNSTEIN polynomials in \underline{B} are linearly independent, Eq. (4.33) can be replaced by

$$\sum_{i=1}^n \alpha^i \underline{C}^{iT} = \underline{\mathbf{0}}^T, \quad (4.34)$$

which is equivalent to

$$\begin{bmatrix} \underline{C}^1 & \dots & \underline{C}^n \end{bmatrix} \begin{bmatrix} \alpha^1 \\ \vdots \\ \alpha^n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (4.35)$$

Thus, using Eq. (4.26), Eq. (4.35) can be rewritten as

$$\underline{\underline{C}}^T \underline{\alpha} = \underline{\mathbf{0}}. \quad (4.36)$$

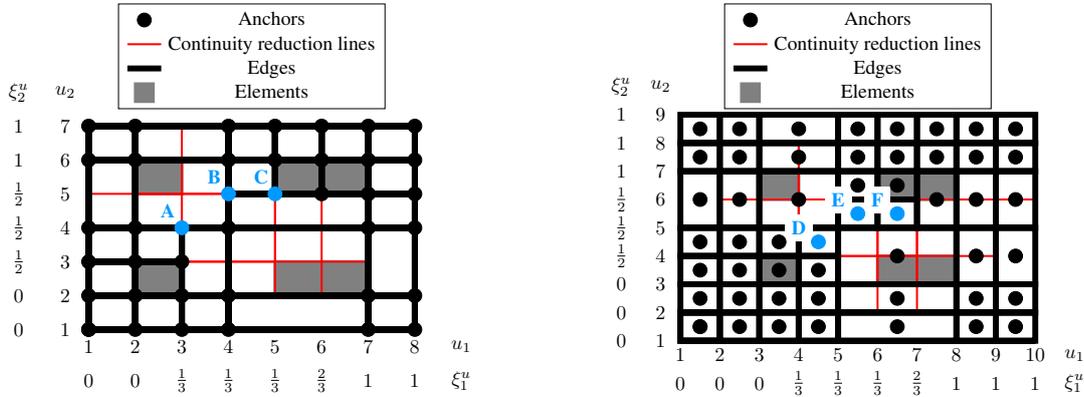
Since a T-spline mesh has globally linearly independent blending functions N^i when the only solution for Eq. (4.36) is $\alpha^i = 0$ for $i = 1 \dots n$, it follows directly from rank inspection of the global BÉZIER extraction operator $\underline{\underline{C}}$ in Eq. (4.36) whether the blending functions N^i of a T-spline mesh are globally linearly independent, cf. WANG *et al.* [120]. If the global BÉZIER extraction operator $\underline{\underline{C}}$ has full row rank, then the rank of $\underline{\underline{C}}$ is equal to the number of anchors n and consequently, the blending functions N^i are globally linearly independent. In sum, the condition for global linear independence is

$$\text{rank}(\underline{\underline{C}}) = n. \tag{4.37}$$

The size of the global BÉZIER extraction operator is

$$\text{size}(\underline{\underline{C}}) = n \times \left(E \cdot \prod_{\ell=1}^d p_{\ell} + 1 \right). \tag{4.38}$$

If a T-spline mesh is globally linearly dependent, GAUSSIAN elimination can transform Eq. (4.36) into a characteristic shape – the row echelon form – and the dependencies between anchors can be detected from the transformed system, see Fig. 4.8.



(a) Globally linearly dependent cubic T-spline mesh from BUFFA *et al.* [20], transforming Eq. (4.36) into row echelon form yields the following linear dependencies between the anchors A, B and C: $-3N^A(\xi) + 3N^B(\xi) + N^C(\xi) = 0$.

(b) Globally linearly dependent quartic T-spline mesh, transforming Eq. (4.36) into row echelon form yields the following linear dependencies between the anchors D, E and F: $-3N^D(\xi) + 2N^E(\xi) + N^F(\xi) = 0$.

Fig. 4.8: Globally linearly dependent T-spline meshes (a) of cubic and (b) of quartic polynomial degree.

A T-spline mesh with globally linearly dependent blending functions cannot be used for analysis since in a finite element context, this results in a system of equations that cannot be solved.

4.4.1.2 Local linear independence

Repeating the procedure of the previous section at the elemental level, the condition for local linear independence is

$$\text{rank}(\underline{\underline{\mathbf{C}}}_e) = n_e \quad \text{for } e = 1 \dots E, \quad (4.39)$$

with the elemental BÉZIER extraction operator $\underline{\underline{\mathbf{C}}}_e$ and the number of anchors n_e with support in element e . When Eq. (4.39) holds, the size of $\underline{\underline{\mathbf{C}}}_e$ is

$$\text{size}(\underline{\underline{\mathbf{C}}}_e) = n_e \times \left(\prod_{\ell=1}^d p_\ell + 1 \right). \quad (4.40)$$

4.4.1.3 Local linear independence with a square matrix $\underline{\underline{\mathbf{C}}}_e$

A subset of locally linearly independent T-spline meshes (i. e. when Eq. (4.39) holds) can be defined when the following additional property is valid for each element e

$$\text{rank}(\underline{\underline{\mathbf{C}}}_e) = \prod_{\ell=1}^d p_\ell + 1 = n_e \quad \text{for } e = 1 \dots E. \quad (4.41)$$

When Eq. (4.41) holds, the size of $\underline{\underline{\mathbf{C}}}_e$ is

$$\text{size}(\underline{\underline{\mathbf{C}}}_e) = \left(\prod_{\ell=1}^d p_\ell + 1 \right) \times \left(\prod_{\ell=1}^d p_\ell + 1 \right). \quad (4.42)$$

Eq. (4.41) implies Eq. (4.39). Also, Eq. (4.39) implies Eq. (4.37) – local linear independence inherently results in global linear independence.

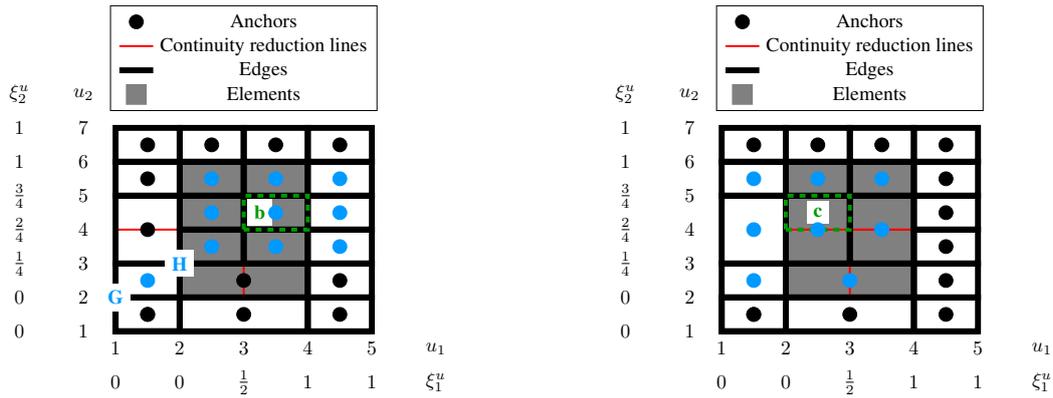
If a T-spline mesh is locally linearly dependent, then the non-zero coefficients $\underline{\underline{\alpha}}_e$ are obtained analogously to the global case by transforming

$$\underline{\underline{\mathbf{C}}}_e^T \underline{\underline{\alpha}}_e = \mathbf{0} \quad (4.43)$$

into row echelon form. In Fig. 4.9, examples are given for a locally linearly dependent T-spline mesh, a T-spline mesh for which Eq. (4.39) holds and a T-spline mesh for which Eq. (4.41) holds.

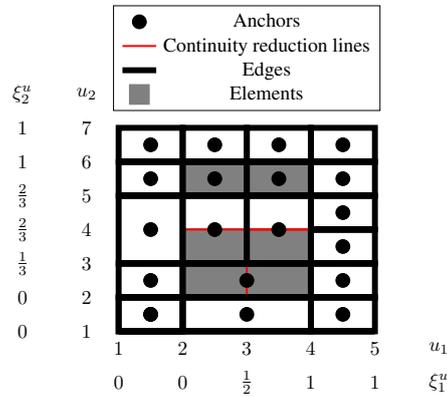
4.4.2 Partition of unity property for T-splines

This section addresses the partition of unity property of the blending functions N^i and of the rational blending functions R^i , respectively. It will be elaborated how T-spline



(a) Locally linearly dependent T-spline mesh, ten anchors (blue) have a support in element b (dashed green); transforming Eq. (4.43) into row echelon form yields the dependencies in element b between anchors G and H: $6N^G(\xi) - N^H(\xi) = 0$.

(b) Locally linearly independent T-spline mesh; $\text{rank}(\underline{\underline{C}}_e) = n_e$ for element c; in element c (dashed green) are only eight anchors (blue) with a support and therefore $\underline{\underline{C}}_e$ is not a square matrix for element c.



(c) Locally linearly independent T-spline mesh;
 $\text{rank}(\underline{\underline{C}}_e) = \prod_{\ell=1}^d p_\ell + 1$ for $e = 1 \dots E$.

Fig. 4.9: Local dependencies in a quadratic T-spline mesh.

meshes can be classified as standard, semi-standard and non-standard using the BÉZIER extraction operator. It will also be shown that an affine transformation exists only when the partition of unity property is satisfied. However, the patch test is not automatically satisfied if an affine transformation exists, see the example utilising POWELL–SABIN B-splines in Section 5.2.3.

4.4.2.1 Partition of unity property of the rational blending functions R^i

The multivariate rational T-spline blending function for an anchor i can be constructed as

$$R^i(\underline{\xi}) = \frac{w^i N^i(\underline{\xi})}{\sum_{j=1}^n w^j N^j(\underline{\xi})}, \quad (4.44)$$

with the weight w^i associated to anchor i . In view of Eq. (4.44), the rational blending functions R^i always form a partition of unity (all R^i sum to one).

4.4.2.2 Partition of unity property of the blending functions N^i

SEDERBERG *et al.* [100] classified T-spline meshes according to the partition of unity property of the blending functions N^i ,

$$\sum_{i=1}^n \beta^i N^i(\underline{\xi}) = 1, \quad (4.45)$$

into

- Standard T-spline meshes: all $\beta^i = 1$,
- Semi-standard T-spline meshes: some $\beta^i \neq 1$,
- Non-standard T-spline meshes: no solution for β^i .

Only for standard T-spline meshes the blending functions N^i and the rational blending functions R^i satisfy the partition of unity property.

4.4.2.3 Partition of unity property of the blending functions N^i using the BÉZIER extraction operator

Next, it will be shown how the global BÉZIER extraction operator can be used to determine the partition of unity property of the blending functions N^i . Rewriting Eq. (4.45) using Eq. (4.30) yields

$$\sum_{i=1}^n \beta^i \underline{\mathbf{C}}_i^T \underline{\mathbf{B}}(\underline{\xi}) = 1. \quad (4.46)$$

Substituting Eqs. (4.25) and (4.29) into Eq. (4.46) and elaboration gives

$$\underbrace{\left(\beta^1 \underline{\mathbf{C}}_1^T + \dots + \beta^n \underline{\mathbf{C}}_1^T \right)}_{\underline{\gamma}_1^T} \underline{\mathbf{B}}_1(\underline{\xi}) + \dots + \underbrace{\left(\beta^1 \underline{\mathbf{C}}_E^T + \dots + \beta^n \underline{\mathbf{C}}_E^T \right)}_{\underline{\gamma}_E^T} \underline{\mathbf{B}}_E(\underline{\xi}) = 1. \quad (4.47)$$

The BERNSTEIN polynomials \underline{B}_e in Eq. (4.47) form a partition of unity if and only if $\underline{\gamma}_e = \underline{\mathbf{1}}$ for each element e . This statement can be expressed in a matrix-vector format as

$$\begin{bmatrix} \beta^1 \underline{C}_1^1 + \dots + \beta^n \underline{C}_1^n \\ \vdots \\ \beta^1 \underline{C}_E^1 + \dots + \beta^n \underline{C}_E^n \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{1}} \\ \vdots \\ \underline{\mathbf{1}} \end{bmatrix}, \quad (4.48)$$

which is equivalent to

$$\begin{bmatrix} \underline{C}^1 & \dots & \underline{C}^n \end{bmatrix} \begin{bmatrix} \beta^1 \\ \vdots \\ \beta^n \end{bmatrix} = \underline{\mathbf{1}}. \quad (4.49)$$

Exploiting the global BÉZIER extraction operator $\underline{\underline{C}}$ in Eq. (4.26) yields

$$\underline{\underline{C}}^T \underline{\beta} = \underline{\mathbf{1}}. \quad (4.50)$$

The row echelon form of Eq. (4.50) then provides the means to assess whether a T-spline mesh is standard, semi-standard or non-standard.

Figs. 4.10 and 4.11 show a set of T-spline meshes of quadratic and cubic degree and their classification of linear dependence and of the partition of unity property according to the previous definitions using the BÉZIER extraction operator. As can be observed from Figs. 4.10 and 4.11, changing the knot intervals gives a different classification for the T-spline mesh – (a), (c), (e) are standard T-spline meshes whereas (b), (d) and (f) are non-standard or semi-standard T-spline meshes.

4.4.2.4 Affine transformation requires partition of unity

Any T-spline surface \underline{T} in the physical domain (x_ℓ) can be expressed by the mapping from the sub-parameter (ξ_ℓ) domain as follows

$$\underline{T}(\underline{\xi}) = \sum_{i=1}^n R^i(\underline{\xi}) \underline{P}^i, \quad (4.51)$$

where $\underline{P}^i = (x_1^i, x_2^i)^T$ are the control points associated to anchor i . Applying a transformation to the control points \underline{P}^i of the form

$$\underline{P}_T^i = \underline{\underline{A}} \underline{P}^i + \underline{b}, \quad (4.52)$$

with the transformation matrix $\underline{\underline{A}}$, the displacement vector \underline{b} , and the control points \underline{P}_T^i after transformation, results in an affine transformation since the rational blending func-

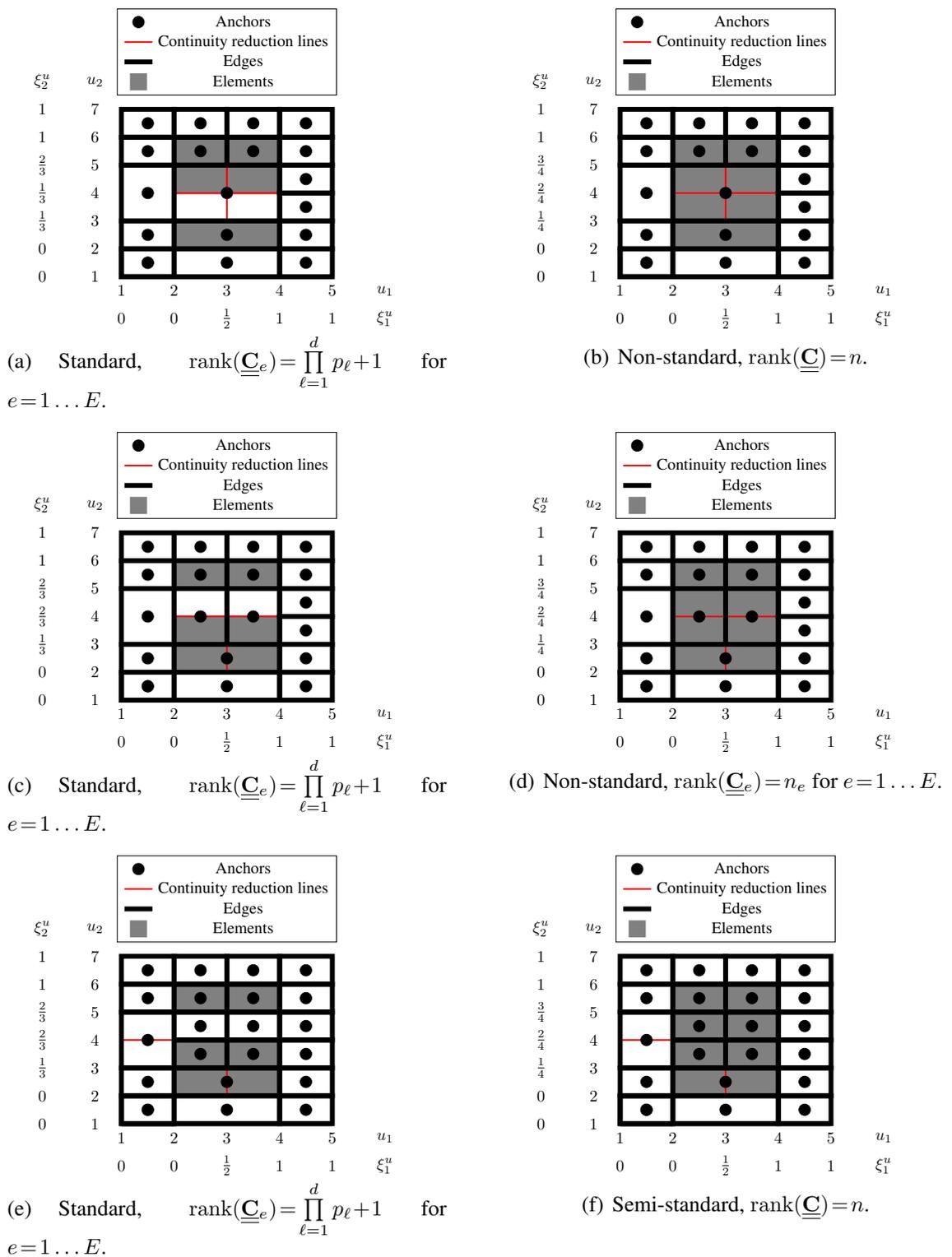


Fig. 4.10: Classification of quadratic T-spline meshes according to the level of linear independence and the partition of unity property. (a), (c) and (e) are standard T-spline meshes, changing the knot intervals results in non-standard or semi-standard T-spline meshes.

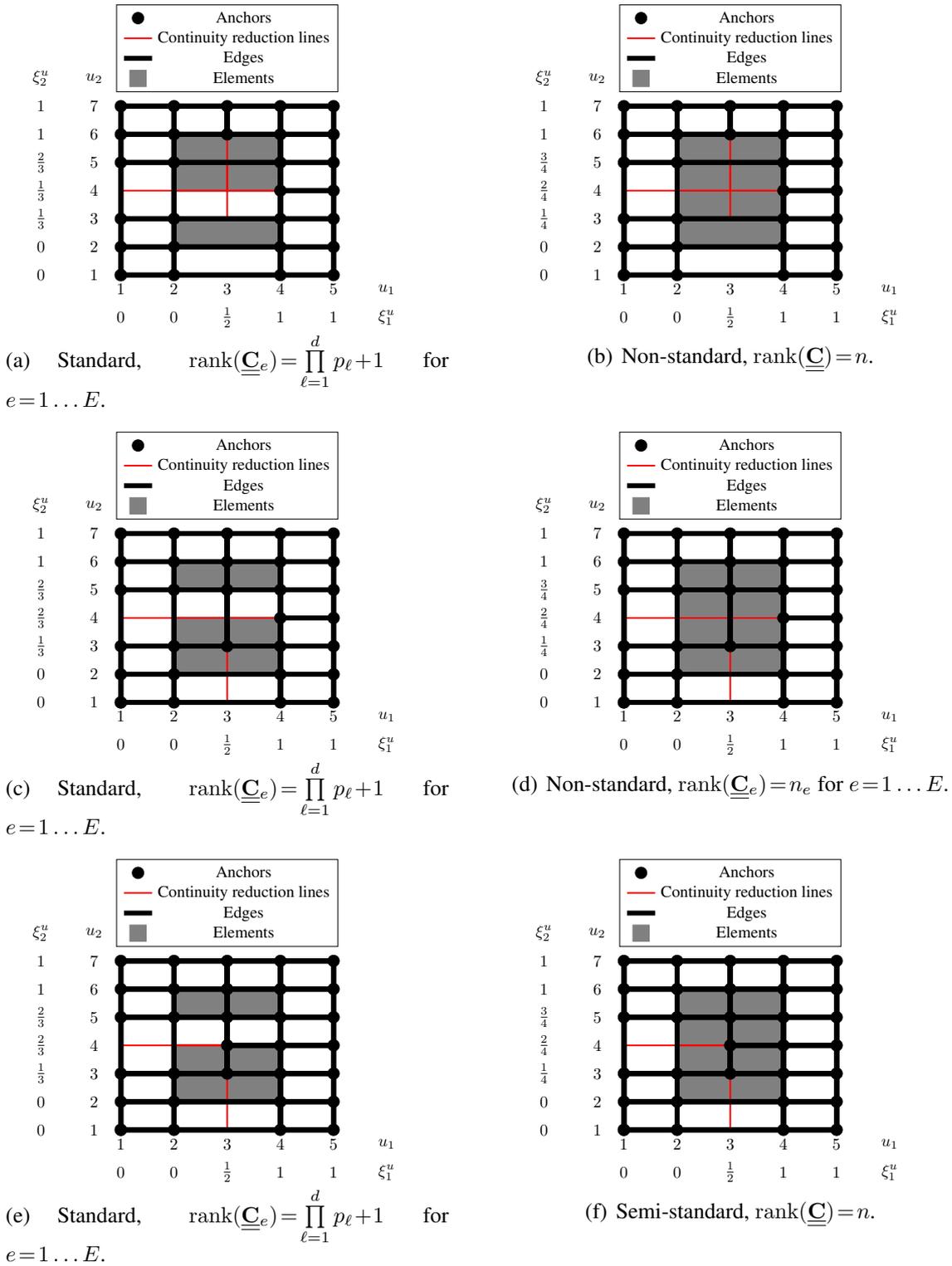


Fig. 4.11: Classification of cubic T-spline meshes according to the level of linear independence and the partition of unity property. (a), (c) and (e) are standard T-spline meshes, changing the knot intervals results in non-standard or semi-standard T-spline meshes.

tions R^i in Eq. (4.51) form a partition of unity.

However, defining the T-spline in Eq. (4.51) with the blending functions N^i instead of the rational blending functions R^i yields an affine transformation only for standard T-spline meshes. For semi-standard and non-standard T-spline meshes, the blending functions N^i do not possess the partition of unity property, see also Fig. 4.12 with a rigid body motion applied to the control points of the anchors.

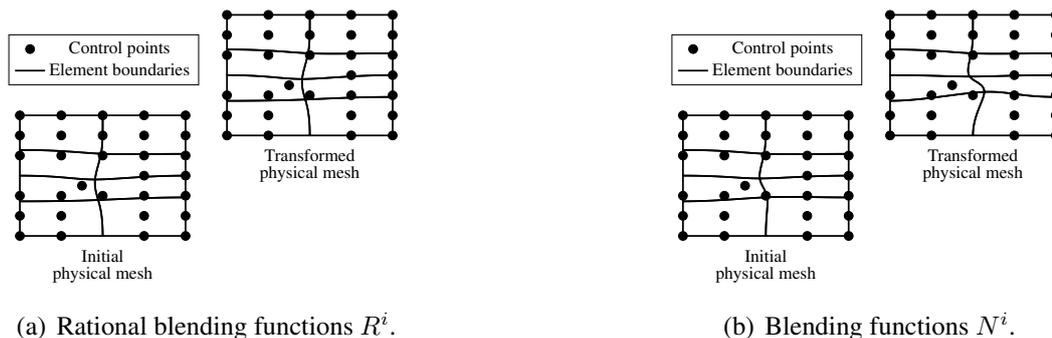


Fig. 4.12: Applying a rigid body motion to the control points of the anchors results in an affine transformation when the partition of unity property is fulfilled. (a) An affine transformation for the semi-standard T-spline mesh in Fig. 4.11(f) is obtained for the rational blending functions R^i ; (b) using the blending functions N^i instead of R^i in Eq. (4.51) gives no affine transformation – the element boundaries are different – since the N^i do not form a partition of unity for semi-standard meshes.

4.5 Local refinement of T-splines by adding anchors

It will be demonstrated in this section how standard, semi-standard and non-standard T-spline meshes of even and odd polynomial degree can be refined locally by adding anchors using information from the BÉZIER extraction operator. This refinement procedure is applicable only to partition of meshes without extraordinary points. The local refinement algorithm ensures that $\underline{\underline{C}}_e$ is a square matrix for each element. This then also allows for hierarchical refinement, see Section 4.6.

A requirement for the refinement algorithm is that the initial and the refined T-spline mesh are nested – this condition will be defined in the following section, together with a method to fulfil it using the BÉZIER extraction operator. Also, it will be shown how the location of the control points in the refined T-spline mesh can be obtained. Afterwards, the algorithm for the local refinement of T-splines will be explained by means of some examples: after demonstrating the refinement of standard T-splines (Section 4.5.4, Appendix B.1.1), it will be shown that also non-standard T-spline meshes can be refined locally by adding anchors (Appendix B.1.2).

Standard T-spline meshes do not necessarily have a square matrix $\underline{\underline{C}}_e$ for each element and global partition of unity does not imply local linear independence – this is not correctly addressed by MAY *et al.* [72]. For instance, unstructured T-spline meshes with extraordinary points (see Section 4.7) can be standard *and* locally linearly dependent. Furthermore, unstructured T-spline meshes that are standard can have a BÉZIER extraction operator that is not a square matrix for elements in the one-ring neighbourhood of an extraordinary point.

4.5.1 Computation of the refinement matrix and nesting behaviour

A refinement matrix $\underline{\underline{M}}$ of size $n \times n_R$ gives the relation between the blending functions \underline{N}_R of a refined mesh with n_R anchors and the blending functions \underline{N} of an initial mesh which has n anchors

$$\underline{N}(\xi) = \underline{\underline{M}} \underline{N}_R(\xi). \quad (4.53)$$

Expressing the blending functions on both sides using the BERNSTEIN polynomials, Eq. (4.28), results in

$$\underline{\underline{C}} \underline{B}_R(\xi) = \underline{\underline{M}} \underline{\underline{C}}_R \underline{B}_R(\xi), \quad (4.54)$$

while the blending functions \underline{N} on the initial mesh must be defined in terms of the elements of the refined mesh with the BERNSTEIN polynomials \underline{B}_R . The linear independence of the BERNSTEIN polynomials \underline{B}_R in Eq. (4.54) gives

$$\underline{\underline{C}} = \underline{\underline{M}} \underline{\underline{C}}_R. \quad (4.55)$$

The coefficients of a row of the refinement matrix $\underline{\underline{M}}$ can be evaluated as follows. Expanding Eq. (4.55) using Eq. (4.26) yields

$$\begin{bmatrix} \underline{C}^{1T} \\ \vdots \\ \underline{C}^{nT} \end{bmatrix} = \begin{bmatrix} \underline{M}^{1T} \\ \vdots \\ \underline{M}^{nT} \end{bmatrix} \begin{bmatrix} \underline{C}_R^{1T} \\ \vdots \\ \underline{C}_R^{n_R T} \end{bmatrix}. \quad (4.56)$$

Applying the transpose to both sides results in

$$\begin{bmatrix} \underline{C}^1 & \dots & \underline{C}^n \end{bmatrix} = \begin{bmatrix} \underline{C}_R^1 & \dots & \underline{C}_R^{n_R} \end{bmatrix} \begin{bmatrix} \underline{M}^1 & \dots & \underline{M}^n \end{bmatrix}, \quad (4.57)$$

which allows to determine the rows \underline{M}^{iT} for $i=1 \dots n$ of the refinement matrix $\underline{\underline{M}}$ by transforming the system

$$\underline{C}^i = \underline{\underline{C}}_R^T \underline{M}^i \quad \text{for } i = 1 \dots n \quad (4.58)$$

into a row echelon form. In the case that there is no solution for \underline{M}^i of anchor i in Eq. (4.58), the initial and the refined T-spline mesh are not nested, which means that it is not possible to represent *all* blending functions \underline{N} of the initial T-spline mesh as a linear combination of the blending functions \underline{N}_R of the refined T-spline mesh. One can resolve this as will be explained in Section 4.5.4 (quadratic case, $p_\ell = 2$) and Appendix B.1.1 (cubic case, $p_\ell = 3$).

When nestedness is ensured and the initial mesh is standard, the refined T-spline mesh can be only a standard or semi-standard T-spline mesh: knowing that the initial T-spline mesh is standard and satisfies the partition of unity property in Eq. (4.45) (with all $\underline{\beta} = \underline{1}$) and using the rows \underline{M}^{iT} of the refinement matrix \underline{M} from Eq. (4.58) results in

$$1 = \sum_{i=1}^n \beta^i N^i(\underline{\xi}) = \sum_{i=1}^n \beta^i \underline{M}^{iT} \underline{N}_R(\underline{\xi}) = \sum_{j=1}^{n_R} \beta_R^j N_R^j(\underline{\xi}), \quad (4.59)$$

where the coefficients $\underline{\beta}_R$ are computed from

$$\underline{\beta}_R = \underline{M}^T \underline{\beta}. \quad (4.60)$$

From Eq. (4.60), it can be concluded that there always exists a solution for the coefficients $\underline{\beta}_R$ when nestedness is ensured (\underline{M} exists) and when the initial mesh is standard. Therefore, the refined T-spline mesh can be only a standard or semi-standard T-spline.

4.5.2 Determination of the coordinates for the anchors in the refined T-spline mesh

In this section, it is assumed that the initial and the refined T-spline mesh are nested. It will be demonstrated how the coordinates and weights of the anchors in a refined T-spline mesh can be determined. The weighted (non-rational) surface of Eq. (4.51) reads (see PIEGL & TILLER [86, Chapter 4])

$$\underline{T}_w(\underline{\xi}) = \sum_{i=1}^n N^i(\underline{\xi}) \underline{P}_w^i, \quad (4.61)$$

with the weighted control points

$$\underline{P}_w^i = (w^i x_1^i, w^i x_2^i, w^i)^T. \quad (4.62)$$

The refined and the initial weighted curves – \underline{T}_{wR} and \underline{T}_w , respectively – are required to represent the same geometry

$$\underline{T}_{wR}(\underline{\xi}) = \underline{T}_w(\underline{\xi}). \quad (4.63)$$

Inserting Eq. (4.61) into the left- and right-hand side of Eq. (4.63) yields

$$\sum_{j=1}^{n_R} N_R^j(\underline{\xi}) \underline{P}_{wR}^j = \sum_{i=1}^n N^i(\underline{\xi}) \underline{P}_w^i. \quad (4.64)$$

Using the BÉZIER extraction operator subsequently gives

$$\sum_{j=1}^{n_R} \underline{P}_{wR}^j \underline{C}_R^{jT} \underline{B}_R(\underline{\xi}) = \sum_{i=1}^n \underline{P}_w^i \underline{C}^{iT} \underline{B}_R(\underline{\xi}), \quad (4.65)$$

or, since the BERNSTEIN polynomials \underline{B}_R are linearly independent,

$$\sum_{j=1}^{n_R} \underline{P}_{wR}^j \underline{C}_R^{jT} = \sum_{i=1}^n \underline{P}_w^i \underline{C}^{iT}. \quad (4.66)$$

Elaborating Eq. (4.66) results in

$$\begin{bmatrix} \underline{P}_{wR}^1 & \cdots & \underline{P}_{wR}^{n_R} \end{bmatrix} \begin{bmatrix} \underline{C}_R^{1T} \\ \vdots \\ \underline{C}_R^{n_RT} \end{bmatrix} = \begin{bmatrix} \underline{P}_w^1 & \cdots & \underline{P}_w^n \end{bmatrix} \begin{bmatrix} \underline{C}^{1T} \\ \vdots \\ \underline{C}^{nT} \end{bmatrix}, \quad (4.67)$$

or in the global form

$$\underline{C}_R^T \underline{P}_{wR} = \underline{C}^T \underline{P}_w, \quad (4.68)$$

with the matrices

$$\underline{P}_{wR} = \begin{bmatrix} \underline{P}_{wR}^{1T} \\ \vdots \\ \underline{P}_{wR}^{n_RT} \end{bmatrix}, \quad \underline{P}_w = \begin{bmatrix} \underline{P}_w^{1T} \\ \vdots \\ \underline{P}_w^{nT} \end{bmatrix}. \quad (4.69)$$

Plugging Eq. (4.55) into Eq. (4.68) leads to

$$\underline{C}_R^T \underline{P}_{wR} = \underline{C}_R^T \underline{M}^T \underline{P}_w. \quad (4.70)$$

Hence, the weighted control points \underline{P}_{wR} for the refined mesh follow from

$$\underline{P}_{wR} = \underline{M}^T \underline{P}_w. \quad (4.71)$$

4.5.3 The algorithm for local refinement of standard T-splines

The local refinement algorithm (see also Algorithm 4.1) for standard T-splines proceeds as follows: after inserting new anchors into the T-spline mesh (refining), it is checked whether each element has a BÉZIER extraction operator $\underline{\underline{C}}_e$ that is a square matrix. If this is not the case, Eq. (4.43) is transformed into row echelon form, and linear dependencies can be detected and removed. Once the BÉZIER extraction operator is a square matrix in each element, each row of the refinement matrix $\underline{\underline{M}}$ in Eq. (4.58) is evaluated. Should the blending functions of some anchors of the initial mesh not be nested in the refined mesh, the mesh needs to be modified accordingly. Finally, when nestedness is satisfied, Eq. (4.50) is assessed whether $\underline{\underline{\beta}} = \underline{\underline{1}}$ holds. If not, then the refined mesh is semi-standard according to Eq. (4.60) and anchors are added to the mesh within the support of anchors for which $\beta^i \neq 1$. Otherwise, the initial and the refined mesh are nested standard T-spline meshes.

```

// Start with a standard T-spline mesh
// Number of refinement steps: N
for i = 1 : N do
    RefinementSuccessful = 0;
    while RefinementSuccessful = 0 do
        // Check whether BÉZIER extraction operator is a square matrix in each element:
        if size( $\underline{\underline{C}}_e$ )  $\neq \left( \prod_{\ell=1}^d p_\ell + 1 \right) \times \left( \prod_{\ell=1}^d p_\ell + 1 \right)$  for e = 1 ... E then
            // add additional anchors by inspecting the BÉZIER extraction operator in Eq. (4.43):
            // (a) ensure that  $\underline{\underline{C}}_e$  is a square matrix
            // (b) remove linear dependencies
        else
            // Check with Eq. (4.58) whether the initial and the refined mesh are nested:
            if Refinement matrix  $\underline{\underline{M}}$  cannot be computed then
                // add additional anchors by assessing the BÉZIER extraction operators of the initial
                // and the refined mesh: localise, which anchors are not nested in Eq. (4.58)
            else
                // Check whether T-spline mesh is standard by assessing Eq. (4.50):
                if  $\underline{\underline{\beta}} \neq \underline{\underline{1}}$  then
                    // mesh is semi-standard according to Eq. (4.60)
                    // add anchors to the mesh within the support of the anchors i for which  $\beta^i \neq 1$ 
                else
                    // Compute the weighted control points  $\underline{\underline{P}}_{wR}$  of the refined mesh using
                    // Eq. (4.71)
                    RefinementSuccessful = 1;
                end
            end
        end
    end
end
end
end

```

Algorithm 4.1: Local refinement algorithm based on the insertion of new anchors for standard T-spline meshes.

4.5.4 Local refinement of standard T-splines of even degree by adding anchors

This section explains how standard T-spline meshes of even degree can be refined using the BÉZIER extraction operator. In order to be able to implement the methods described in the following, the local knot vectors for each anchor are required in the index (u_ℓ) and sub-parameter (ξ_ℓ) domain – it is not sufficient to have only access to the BÉZIER extraction operator.

4.5.4.1 Example 1: Ensuring that $\underline{\underline{C}}_e$ is a square matrix and nestedness

Initial refinement

Consider the quadratic standard T-spline mesh in the index domain and the physical domain in Fig. 4.13.

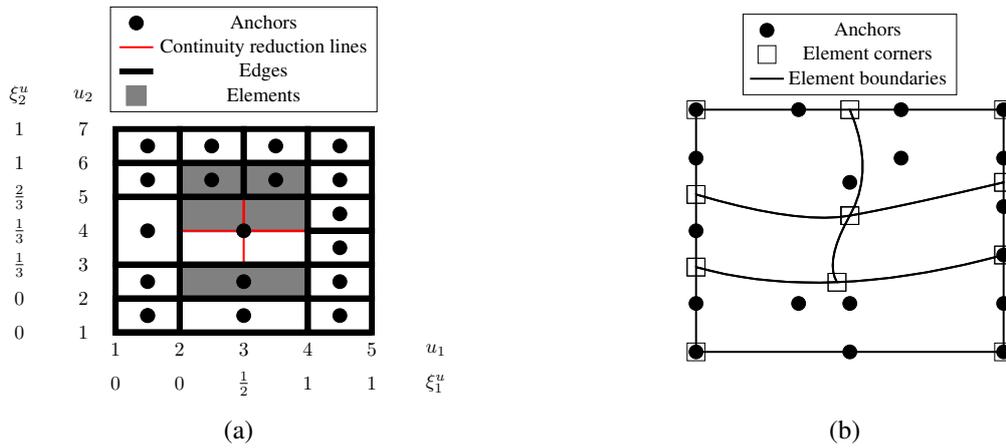


Fig. 4.13: Initial quadratic standard T-spline mesh in (a) the index domain and (b) the physical domain.

It is refined by insertion of an anchor which results in the rectangle $[\xi_1^2, \xi_1^4] \times [\xi_2^3, \xi_2^5]$ being split vertically, see Fig. 4.14(a).

Ensuring that $\underline{\underline{C}}_e$ is a square matrix

The resulting mesh is locally linearly independent but non-standard, and $\underline{\underline{C}}_e$ is not a square matrix for all elements: for element b (bounded by a dashed green line), $\text{rank}(\underline{\underline{C}}_e) = n_e$, as there are only eight anchors (blue) with a support, $n_e = 8$, see Fig. 4.14(b). Hence, additional anchors need to be inserted in order to obtain a square matrix $\underline{\underline{C}}_e$. Each local knot vector of the blue anchors with support in element b in Fig. 4.14(b) contains the sub-parameter values of the boundaries of element b – $[0, \frac{1}{2}] \times [\frac{1}{3}, \frac{2}{3}]$ in the ξ_1 -direction and the ξ_2 -direction, respectively, except for the anchors A and B in Fig. 4.15(a). The local knot vectors of the anchors A and B in the ξ_1 -direction do not contain the sub-parameter

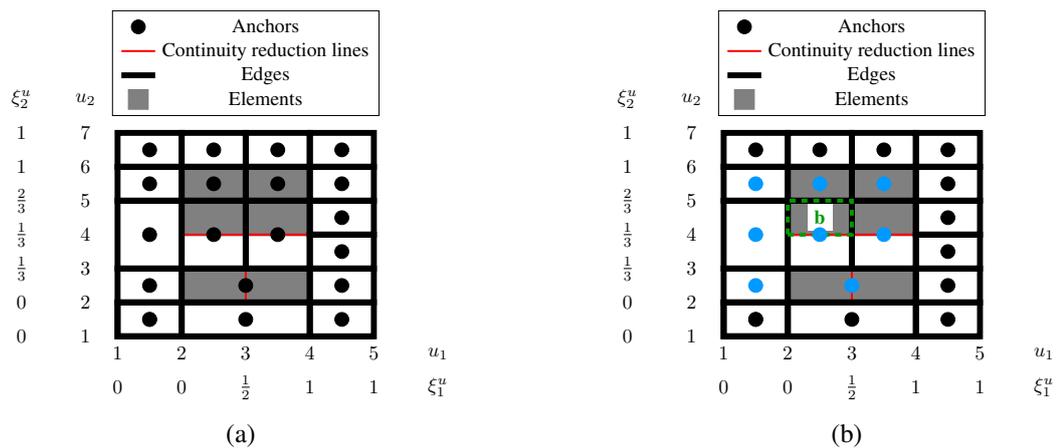


Fig. 4.14: (a) Refined quadratic non-standard T-spline mesh of Fig. 4.13(a) in the index domain. (b) The T-spline mesh is locally linearly independent – but as only eight anchors (blue) have a support in element b (dashed green line), \underline{C}_e is not a square matrix for element b.

value $\xi_1 = \frac{1}{2}$. Therefore, rectangle c needs to be split. This results in the standard T-spline mesh in Fig. 4.15(b).

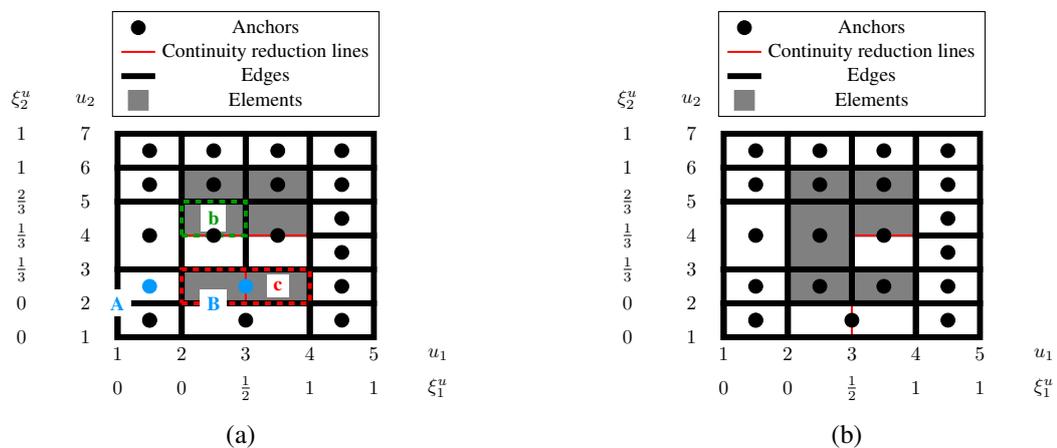


Fig. 4.15: Procedure to obtain a square matrix \underline{C}_e . (a) The local knot vectors of the anchors A and B (blue) do not contain the sub-parameter value $\xi_1 = \frac{1}{2}$, which is a boundary of element b (dashed green). The local knot vectors of all other anchors with support in element b (see Fig. 4.14(b)) contain the sub-parameter values 0, $\frac{1}{2}$ in the ξ_1 -direction and $\frac{1}{3}$, $\frac{2}{3}$ in the ξ_2 -direction – $[0, \frac{1}{2}] \times [\frac{1}{3}, \frac{2}{3}]$ represents the boundary values of element b in the sub-parameter domain. Hence, the rectangle c needs to be split so that the local knot vectors of the anchors A and B also contain the knot $\xi_1 = \frac{1}{2}$. (b) The resulting standard mesh and the initial mesh in 4.13(a) are not nested.

Nestedness

The initial T-spline mesh in Fig. 4.13(a) and the refined mesh in Fig. 4.15(b) are not

nested: the blending functions of anchors C, D and E (see mesh in Fig. 4.16) cannot be expressed as a linear combination of the blending functions of the refined mesh in Fig. 4.15(b). This can be identified by inspection of the row echelon form of Eq. (4.58) for these anchors.

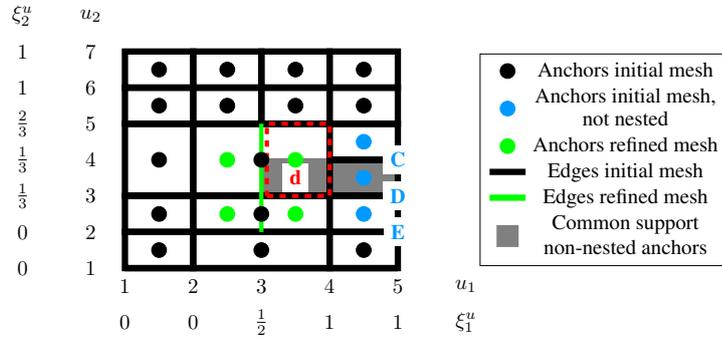


Fig. 4.16: Superposition of the initial T-spline mesh in the index domain from Fig. 4.13(a) and the refined mesh in Fig. 4.15(b). Transforming Eq. (4.58) into row echelon form gives no results for the anchors C, D and E (blue) since the meshes in Fig. 4.13(a) and Fig. 4.15(b) are not nested. Edges and anchors from the refined mesh in Fig. 4.15(b), which were added during refinement, are inserted in the initial mesh from Fig. 4.13(a) and marked with green. Within the grey domain all three anchors C, D and E from the initial mesh have a support, while the grey domain is bounded by the newly inserted green edges. In this grey domain an additional anchor needs to be inserted, i. e. the dashed red rectangle d needs to be subdivided, see Fig. 4.17.

Therefore, an additional anchor has to be inserted. The new edges and anchors of the refined mesh in Fig. 4.15(b) are drawn in the initial mesh of Fig. 4.13(a) as illustrated with solid green lines and green points in Fig. 4.16. The grey domain highlights the common support of the three anchors C, D and E and is bounded by the new green edges. Within the grey domain a new anchor needs to be inserted, i. e. the dashed red rectangle d needs to be subdivided. The resulting refined mesh has now the sought properties: it is standard and is nested with the initial (non-refined) mesh, i. e. the blending function \underline{N} of each anchor in Fig. 4.13(a) can be represented as a linear combination of the blending functions \underline{N}_R of the anchors in the refined mesh in Fig. 4.17(a).

Refined physical mesh

So far, refinement has been considered only in the index domain in order to obtain a standard and nested T-spline mesh. Next, the evaluation of the weighted control points in the physical domain is addressed.

The location of the weighted control points for the refined mesh \underline{P}_{wR} is determined using Eq. (4.71). The physical mesh is shown in Fig. 4.17(b) which preserves the same geometry as the physical mesh in Fig. 4.13(b). This can be observed by comparing for instance the shape of the element boundaries of the initial and the refined physical mesh.

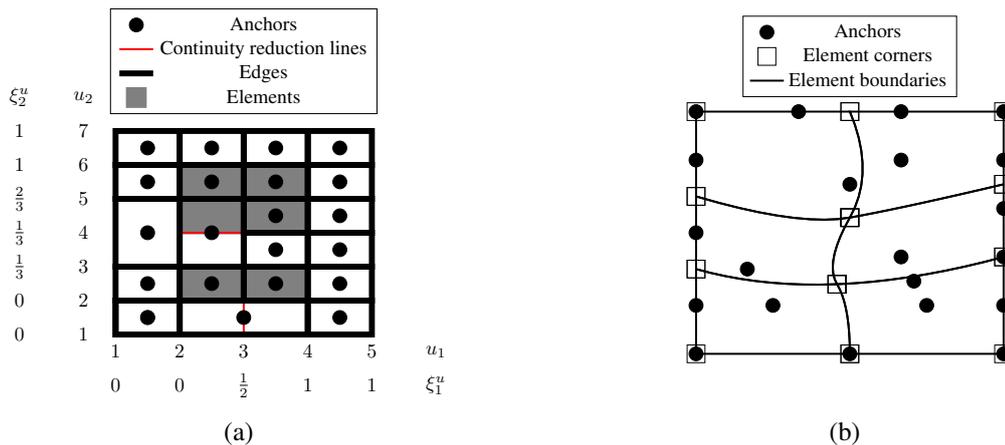
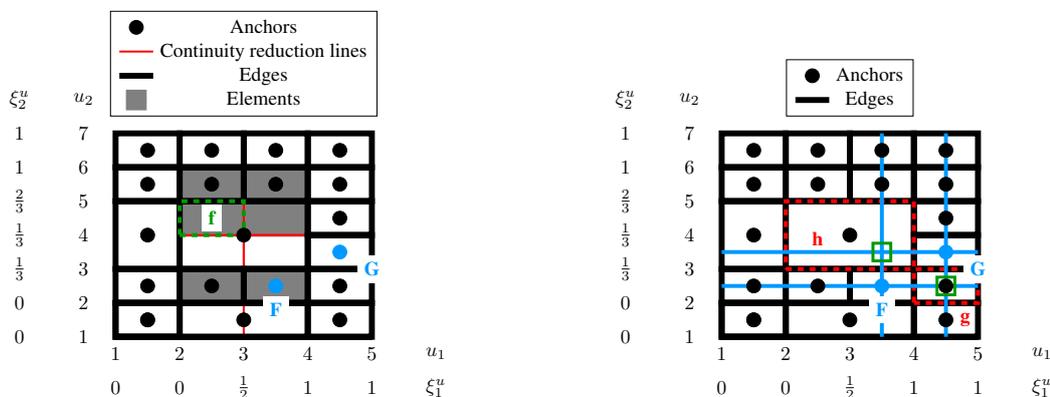


Fig. 4.17: Refined quadratic T-spline mesh of Fig. 4.13 in (a) the index domain and (b) the physical domain. This T-spline mesh is standard and nested with the initial T-spline mesh in Fig. 4.13.

4.5.4.2 Example 2: Removing linear dependencies

Initial refinement

As a next example, the initial quadratic T-spline mesh in Fig. 4.13 is now refined as shown in Fig. 4.18(a).



(a) The T-spline mesh is locally linearly dependent – the row echelon version of Eq. (4.43) gives the dependency $N^F(\xi) - N^G(\xi) = 0$ in element f.

(b) Extension lines (solid blue) for the anchors F and G intersect at the location of the green squares. The rectangles g and h (dashed red line) contain the green squares. Rectangle g cannot be further subdivided. All options for subdividing rectangle h are given in Fig. 4.19.

Fig. 4.18: Refined (non-standard) quadratic T-spline mesh of Fig. 4.13(a) in the index domain.

Removing linear dependencies

The T-spline mesh of Fig. 4.18(a) is non-standard using Eq. (4.50). Furthermore, the BÉZIER extraction operator $\underline{\underline{C}}_e$ is not a square matrix and does not have full row rank

for element f . Transforming Eq. (4.43) into row echelon form yields the dependency $N^F(\xi) - N^G(\xi) = 0$ in element f . In order to break this dependence, new anchors need to be inserted. In the following, it will be shown how to identify potential locations for these new anchors and how to select the ideal one.

Extension lines (solid blue) are drawn between the anchors F and G as depicted in Fig. 4.18(b). These extension lines intersect at the location of the green squares. These squares are located in the rectangles g and h (dashed red line). Rectangle g cannot be further subdivided, but rectangle h can, as shown in Fig. 4.19.

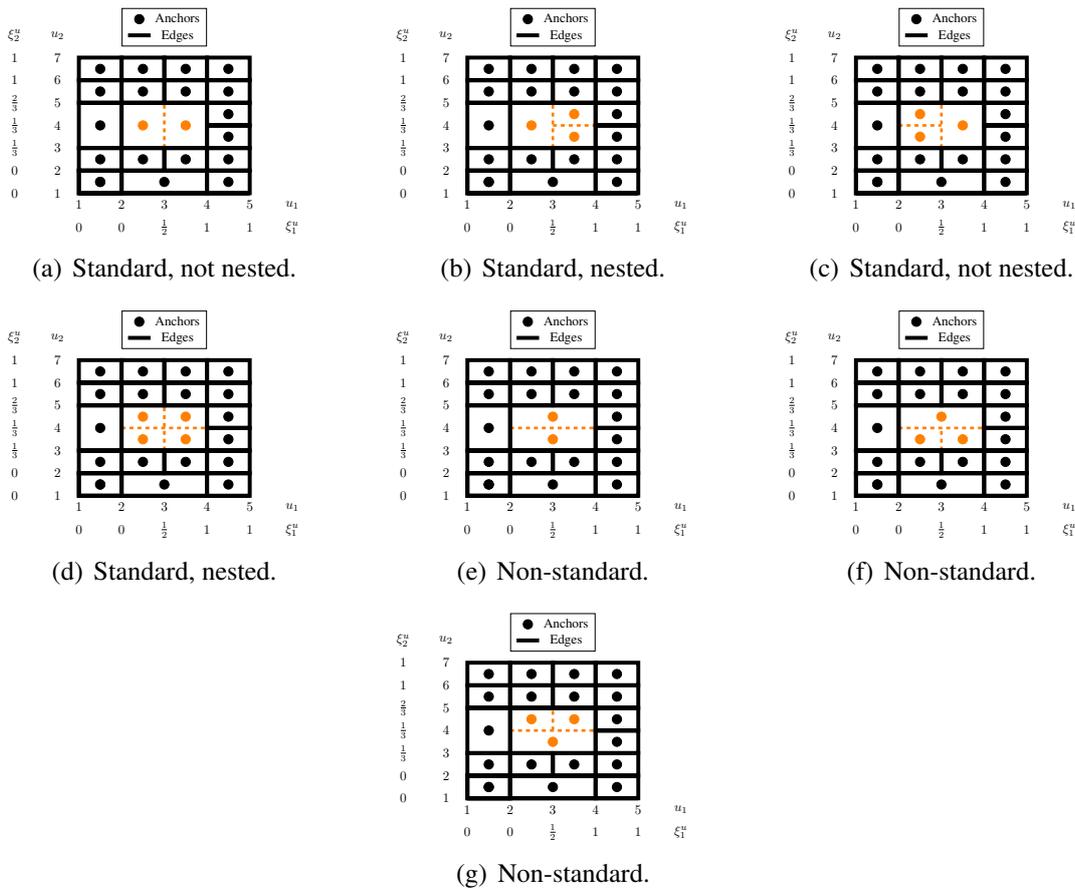


Fig. 4.19: All possible subdivisions for the rectangle h in Fig. 4.18(b): the dashed orange lines indicate the new edges to be inserted, the orange points denote the locations of the new anchors.

Table 4.1 gives a summary of the number of pairs of anchors with linearly dependent blending functions, number of non-square matrices $\underline{\underline{C}}_e$, nestedness and number of additionally inserted anchors for the options in Fig. 4.19. This information can be used in order to determine the best location and optimum number of additional anchors.

According to Fig. 4.19 and Table 4.1, only the options (b) and (d) are suitable for refinement of the T-spline mesh in Fig. 4.13 since they are standard and nested with the initial

Tab. 4.1: Summary of the number of pairs of anchors with linearly dependent blending functions, number of non-square matrices $\underline{\underline{\mathbf{C}}}_e$, nestedness and number of additionally inserted anchors for the options in Fig. 4.19.

Figure	4.19(a)	4.19(b)	4.19(c)	4.19(d)	4.19(e)	4.19(f)	4.19(g)
Number of pairs of anchors with linearly dependent blending functions	0	0	0	0	0	0	0
Number of non-square matrices $\underline{\underline{\mathbf{C}}}_e$	0	0	0	0	1	2	2
Nestedness	✗	✓	✗	✓	✗	✗	✗
Number of additional anchors	2	3	3	4	2	3	3

mesh. Regarding efficiency, the option which introduces the smallest amount of new anchors could be selected, i. e. option (b).

In case that no refinement option results in a standard and nested T-spline mesh, one can select either the option with the smallest number of pairs of anchors with linearly dependent blending functions or the option with the smallest number of non-square matrices $\underline{\underline{\mathbf{C}}}_e$ and then continue with the next refinement step until a standard and nested mesh is obtained, see Appendix B.1.3.

4.5.5 Summary for the local refinement of standard T-splines

The examples for the local refinement of standard T-spline meshes by adding anchors demonstrate how to exploit the information from the BÉZIER extraction operator:

- enforce a square matrix $\underline{\underline{\mathbf{C}}}_e$ in each element:
 - the BÉZIER extraction operator indicates, which element does not have enough anchors with a support (Fig. 4.15(a));
 - when there are local linear dependencies, the BÉZIER extraction operator shows, where new anchors and edges need to be inserted (Fig. 4.18(b))
- pinpoint for which blending functions two T-spline meshes are not nested (Fig. 4.16).

It was found that a square BÉZIER extraction operator $\underline{\underline{\mathbf{C}}}_e$ for each element and a computable refinement matrix $\underline{\underline{\mathbf{M}}}$ in Eq. (4.58) resulted in a nested standard T-spline mesh. It was not observed that this resulted in a nested semi-standard T-spline mesh. However, should such a case arise, one can pinpoint for which anchors $\beta_R^i \neq 1$ using Eq. (4.60) and insert an additional anchor in the supported domain of these anchors.

The local refinement of standard T-spline meshes of odd degree is treated in Appendix B.1.1. Furthermore, Appendix B.1.2 demonstrates that also non-standard T-splines can be refined locally when nestedness exists.

4.6 Hierarchical refinement of T-splines using the reconstruction operator

THOMAS *et al.* [110] introduced another refinement strategy based on the reconstruction operator. Instead of adding new anchors to the mesh as was proposed in the previous section, the method is based on the division of elements while an invertible elemental BÉZIER extraction operator $\underline{\underline{\mathbf{C}}}_e$ is needed for the reconstruction operator. The hierarchical refinement method by THOMAS *et al.* [110] has been applied to analysis-suitable T-splines (see Section 4.8). Here, it will be demonstrated how the idea of this concept can also be applied to standard, semi-standard and non-standard T-spline meshes.

4.6.1 Splitting elements

The hierarchical refinement algorithm based on the reconstruction operator requires that $\underline{\underline{\mathbf{C}}}_e$ is a square matrix of full rank for the element e that is subdivided,

$$\text{rank}(\underline{\underline{\mathbf{C}}}_e) = \prod_{\ell=1}^d p_{\ell} + 1, \quad (4.72)$$

since the reconstruction operator, defined as

$$\underline{\underline{\mathbf{R}}}_e = \underline{\underline{\mathbf{C}}}_e^{-1}, \quad (4.73)$$

is needed. Therefore, for this hierarchical refinement algorithm the BÉZIER extraction operator plays again a key role: when Eq. (4.72) is satisfied for element e , this element can be refined hierarchically. Thus, this algorithm can be applied to standard, semi-standard and non-standard T-spline meshes.

Consider an element with range $[-1, 1]$ and suppose that it needs to be split in half: $[-1, 0]$ and $[0, 1]$. The first BERNSTEIN basis B_1^1 with the knot vector $\{-1, -1, -1, 1\}$ (black curve) in Fig. 4.20 in the element $[-1, 1]$ can be defined in the two sub-elements $[-1, 0]$ and $[0, 1]$ as a linear combination of the BERNSTEIN polynomials in the two sub-elements: the BERNSTEIN basis functions for the left part of the element with support in $[-1, 0]$ are

given by the local knot vectors

$$B_{1l}^1 \text{ for } \{-1, -1, -1, 0\}, \quad B_{1l}^2 \text{ for } \{-1, -1, 0, 0\}, \quad B_{1l}^3 \text{ for } \{-1, 0, 0, 0\}. \quad (4.74)$$

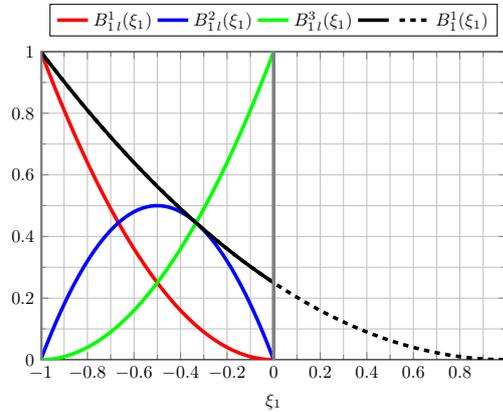


Fig. 4.20: The BERNSTEIN polynomial B_1^1 with support over the element $[-1, 1]$ can be expressed in the sub-element e_l with range $[-1, 0]$ as a linear combination of the BERNSTEIN polynomials B_{1l}^a : $B_1^1(\xi_1) = B_{1l}^1(\xi_1) + \frac{1}{2}B_{1l}^2(\xi_1) + \frac{1}{4}B_{1l}^3(\xi_1)$.

The BERNSTEIN polynomial B_1^1 in the left part of the element (solid black line) can now be expressed as a linear combination of the BERNSTEIN polynomials B_{1l}^i as follows

$$B_1^1(\xi_1) = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} B_{1l}^1(\xi_1) \\ B_{1l}^2(\xi_1) \\ B_{1l}^3(\xi_1) \end{bmatrix}. \quad (4.75)$$

The coefficients in Eq. (4.75) can either be obtained using the algorithm from SCOTT *et al.* [94] for the knot vector $\{-1, -1, -1, 1\}$ with an interior knot at $\xi_1=0$ or, alternatively, using the relations in FAROUKI & NEFF [39]. Applying the same procedure to B_1^2 , B_1^3 , and on the right part of the element (with the range $[0, 1]$) gives

$$\underbrace{\begin{bmatrix} B_1^1(\xi_1) \\ B_1^2(\xi_1) \\ B_1^3(\xi_1) \end{bmatrix}}_{\underline{\mathbf{B}}_1(\xi_1)} = \underbrace{\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{4} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{4} \end{bmatrix}}_{\underline{\mathbf{A}}_{1l}} \underbrace{\begin{bmatrix} B_{1l}^1(\xi_1) \\ B_{1l}^2(\xi_1) \\ B_{1l}^3(\xi_1) \end{bmatrix}}_{\underline{\mathbf{B}}_{1l}(\xi_1)} + \underbrace{\begin{bmatrix} \frac{1}{4} & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{4} & \frac{1}{2} & 1 \end{bmatrix}}_{\underline{\mathbf{A}}_{1r}} \underbrace{\begin{bmatrix} B_{1r}^1(\xi_1) \\ B_{1r}^2(\xi_1) \\ B_{1r}^3(\xi_1) \end{bmatrix}}_{\underline{\mathbf{B}}_{1r}(\xi_1)}. \quad (4.76)$$

Hence, the BERNSTEIN polynomials $\underline{\mathbf{B}}_1$ over one element e with the span $[-1, 1]$ can be expressed as a linear combination of the BERNSTEIN polynomials $\underline{\mathbf{B}}_{1l}$ and $\underline{\mathbf{B}}_{1r}$ over the two smaller elements e_l with the span $[-1, 0]$ and e_r with the span $[0, 1]$. Extending

Eq. (4.76) into more dimensions gives

$$\underline{\mathbf{B}}(\underline{\xi}) = \underline{\mathbf{A}}_l \underline{\mathbf{B}}_l(\underline{\xi}) + \underline{\mathbf{A}}_r \underline{\mathbf{B}}_r(\underline{\xi}). \quad (4.77)$$

Next, it is assumed that the BÉZIER extraction operator is known for the original, single element $\underline{\mathbf{C}}_e$ and for the two sub-elements $\underline{\mathbf{C}}_{el}$ and $\underline{\mathbf{C}}_{er}$. Then, the weighted surface $\underline{\mathbf{T}}_{we}(\underline{\xi})$ with the weighted control points $\underline{\mathbf{P}}_{we}^i$ can be expressed over element e using Eq. (4.77)

$$\underline{\mathbf{T}}_{we}(\underline{\xi}) = \sum_{i=1}^{n_e} \underline{\mathbf{C}}_e^i T \underline{\mathbf{B}}(\underline{\xi}) \underline{\mathbf{P}}_{we}^i = \sum_{i=1}^{n_e} \underline{\mathbf{C}}_e^i T (\underline{\mathbf{A}}_l \underline{\mathbf{B}}_l(\underline{\xi}) + \underline{\mathbf{A}}_r \underline{\mathbf{B}}_r(\underline{\xi})) \underline{\mathbf{P}}_{we}^i. \quad (4.78)$$

Alternatively, $\underline{\mathbf{T}}_{we}(\underline{\xi})$ can be written over the two sub-elements e_l and e_r as

$$\underline{\mathbf{T}}_{we}(\underline{\xi}) = \underline{\mathbf{T}}_{wel}(\underline{\xi}) + \underline{\mathbf{T}}_{wer}(\underline{\xi}) = \sum_{j=1}^{n_{el}} \underline{\mathbf{C}}_{el}^j T \underline{\mathbf{B}}_l(\underline{\xi}) \underline{\mathbf{P}}_{wel}^j + \sum_{k=1}^{n_{er}} \underline{\mathbf{C}}_{er}^k T \underline{\mathbf{B}}_r(\underline{\xi}) \underline{\mathbf{P}}_{wer}^k, \quad (4.79)$$

with the weighted control points $\underline{\mathbf{P}}_{wel}^j$ and $\underline{\mathbf{P}}_{wer}^k$ for element e_l and e_r , respectively. Comparing Eq. (4.78) and Eq. (4.79) results in

$$\sum_{i=1}^{n_e} \underline{\mathbf{C}}_e^i T \underline{\mathbf{A}}_l \underline{\mathbf{B}}_l(\underline{\xi}) \underline{\mathbf{P}}_{we}^i = \sum_{j=1}^{n_{el}} \underline{\mathbf{C}}_{el}^j T \underline{\mathbf{B}}_l(\underline{\xi}) \underline{\mathbf{P}}_{wel}^j, \quad (4.80)$$

$$\sum_{i=1}^{n_e} \underline{\mathbf{C}}_e^i T \underline{\mathbf{A}}_r \underline{\mathbf{B}}_r(\underline{\xi}) \underline{\mathbf{P}}_{we}^i = \sum_{k=1}^{n_{er}} \underline{\mathbf{C}}_{er}^k T \underline{\mathbf{B}}_r(\underline{\xi}) \underline{\mathbf{P}}_{wer}^k, \quad (4.81)$$

or in matrix form

$$\underline{\mathbf{C}}_e^T \underline{\mathbf{A}}_l \underline{\mathbf{P}}_{we} = \underline{\mathbf{C}}_{el}^T \underline{\mathbf{P}}_{wel}, \quad (4.82)$$

$$\underline{\mathbf{C}}_e^T \underline{\mathbf{A}}_r \underline{\mathbf{P}}_{we} = \underline{\mathbf{C}}_{er}^T \underline{\mathbf{P}}_{wer}. \quad (4.83)$$

Hence, the weighted coordinates of the two sub-elements are obtained with the reconstruction operator in Eq. (4.73) as

$$\underline{\mathbf{P}}_{wel} = \underline{\mathbf{R}}_{el}^T \underline{\mathbf{C}}_e^T \underline{\mathbf{A}}_l \underline{\mathbf{P}}_{we}, \quad \underline{\mathbf{P}}_{wer} = \underline{\mathbf{R}}_{er}^T \underline{\mathbf{C}}_e^T \underline{\mathbf{A}}_r \underline{\mathbf{P}}_{we}. \quad (4.84)$$

4.6.2 Example

Consider the quadratic non-standard T-spline mesh of Fig. 4.21 which is globally linearly independent but locally linearly dependent.

The dashed green element b is now divided vertically into two sub-element b_l and b_r with

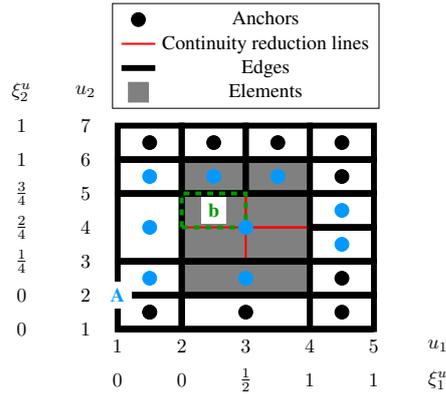


Fig. 4.21: Non-standard, globally linearly independent T-spline mesh in the index domain (from Fig. 4.10(b)). Element b (dashed green line) with range $[0, \frac{1}{2}] \times [\frac{2}{4}, \frac{3}{4}]$ is split vertically into two sub-elements b_l and b_r with range $[0, \frac{1}{4}] \times [\frac{2}{4}, \frac{3}{4}]$ and $[\frac{1}{4}, \frac{1}{2}] \times [\frac{2}{4}, \frac{3}{4}]$, respectively. Each local knot vector associated to a blue anchor (i. e. those having a support in element b) needs to be modified. For instance, the anchor A with $\Xi_1^A = \{0, 0, 0, 1\}$ becomes $\Xi_{1l}^A = \{0, 0, 0, \frac{1}{4}\}$ in element b_l and $\Xi_{1r}^A = \{0, 0, \frac{1}{4}, 1\}$ in element b_r . $\Xi_1^A = \{0, 0, 0, 1\}$ remains unchanged for the other elements. The modified local knot vectors for the other blue anchors are given in Appendix B.2.

range $[0, \frac{1}{4}] \times [\frac{2}{4}, \frac{3}{4}]$ and $[\frac{1}{4}, \frac{1}{2}] \times [\frac{2}{4}, \frac{3}{4}]$, i. e. the knot value $\xi_1 = \frac{1}{4}$ is inserted in element b . Element b can be subdivided since Eq. (4.72) holds for it. In order to obtain the weighted control points $\underline{\underline{P}}_{w_{el}}$ and $\underline{\underline{P}}_{w_{er}}$ in Eq. (4.84), the reconstruction operators $\underline{\underline{R}}_{el}$ and $\underline{\underline{R}}_{er}$, which follow from the BÉZIER extraction operators $\underline{\underline{C}}_{el}$ and $\underline{\underline{C}}_{er}$, respectively, are needed for element b . These BÉZIER extraction operators are based on the modified local knot vectors of the sub-elements b_l and b_r which are obtained as follows. Take from each blue anchor i which has a support over element b in Fig. 4.21 the local knot vector Ξ_1^i . Then insert into this local knot vector the knot value $\xi_1 = \frac{1}{4}$ and split the resulting knot vector into two knot vectors of length $p_\ell + 2$, where one knot vector contains the first $p_\ell + 2$ entries and the other one the last $p_\ell + 2$ entries. For instance, taking the anchor A in Fig. 4.21 gives the local knot vector $\Xi_1^A = \{0, 0, 0, 1\}$. The local knot vectors for the elements b_l and b_r are then $\Xi_{1l}^A = \{0, 0, 0, \frac{1}{4}\}$ and $\Xi_{1r}^A = \{0, 0, \frac{1}{4}, 1\}$. The local knot vector Ξ_1^A is modified only in the elements b_l and b_r , while for the other elements Ξ_1^A remains unchanged. The local knot vectors for the blue anchors in the sub-elements b_l and b_r are listed in Appendix B.2.

The initial non-standard T-spline mesh and the hierarchically refined non-standard T-spline mesh in the physical domain are depicted in Fig. 4.22. Both physical meshes represent the same geometry.

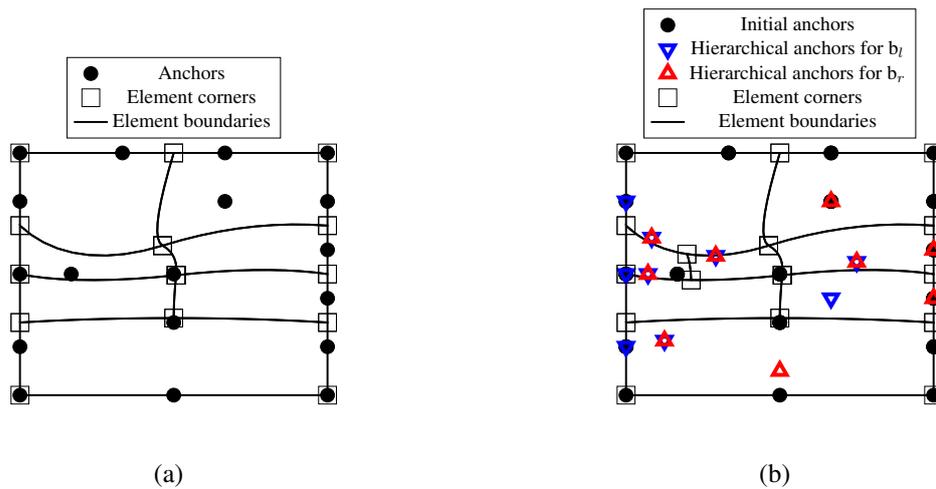


Fig. 4.22: (a) Initial and (b) hierarchically refined non-standard T-spline mesh from Fig. 4.21 in the physical domain.

4.7 Unstructured quadratic T-splines

This section addresses unstructured T-spline meshes of quadratic degree – the cubic degree is considered in Appendix B.3. An unstructured T-spline mesh contains points with valence three or more than four. These points are called star, irregular or extraordinary points. Without special treatment, these meshes are non-standard and only C_A^0 -continuous around the extraordinary points. It will be demonstrated how these meshes can be manipulated such that they are standard (blending functions N^i fulfil the partition of unity) and C_A^1 -continuous around the extraordinary points.

The technique presented here is very similar to the idea by SCOTT [95]. However, the approach in SCOTT [95] does not fulfil the partition of unity for the blending function N^i in the one-ring neighbourhood elements of an extraordinary point. Herein, it will be demonstrated how this deficiency can be repaired.

4.7.1 The unstructured T-spline mesh

Fig. 4.23 shows an unstructured quadratic T-spline mesh.

Extraordinary points are indicated by a red circle. Spoke edges are marked with green: they touch an extraordinary point. In order to build the BÉZIER extraction operator for each purple anchor that has a support in the light grey element g , the knot intervals of the neighbouring rectangles (marked orange) for element g are required. Some anchors with support in g do not require all their individual knot intervals in order to determine their BÉZIER extraction operator in g . This construction cannot be applied to the blue elements in the one-ring neighbourhood of the extraordinary points. These elements are called

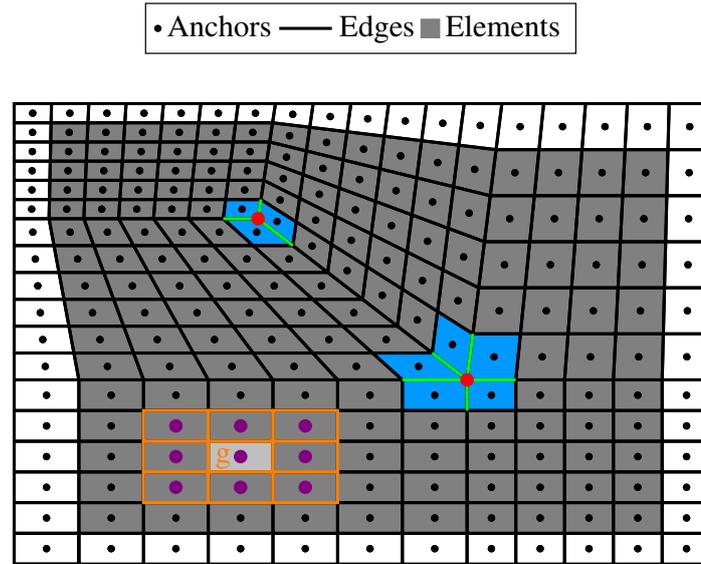


Fig. 4.23: Example for an unstructured quadratic T-spline mesh. All non-zero knot intervals are assumed to be uniform. The two extraordinary points of valence three and five are marked with red. Spoke edges (green) touch an extraordinary point. Anchors with support in element g (light grey) are marked violet. In order to determine the BÉZIER extraction operator for these anchors in element g , the knot intervals of the neighbouring rectangles of g are required. Some of the violet anchors do not need all of their individual knot intervals. This technique is not applicable to the one-ring neighbourhood elements (blue) of an extraordinary point. Generalised BÉZIER extraction must be applied to these irregular elements.

irregular elements, whereas the other elements are regular elements. For the irregular elements, generalised BÉZIER extraction will be utilised. Generalised BÉZIER extraction defines the transpose of the BÉZIER extraction operator, $\underline{\underline{C}}_e^T$.

4.7.2 Generalised BÉZIER extraction

Generalised BÉZIER extraction yields a relation between BÉZIER control points \underline{Q}_e and the control points \underline{P}_e with support in element e

$$\underline{Q}_e = \underline{\underline{C}}_e^T \underline{P}_e. \quad (4.85)$$

Each quadratic element has nine BÉZIER control points – one face point \underline{Q}_5^f , four edge points $\underline{Q}_2^e, \underline{Q}_4^e, \underline{Q}_6^e, \underline{Q}_8^e$ and four vertex points $\underline{Q}_1^v, \underline{Q}_3^v, \underline{Q}_7^v, \underline{Q}_9^v$ as depicted in Fig. 4.24(a). Herein, it is assumed that all non-zero knot intervals are uniform. The general case for non-uniform knot-intervals is considered by SCOTT [95] for the cubic case.

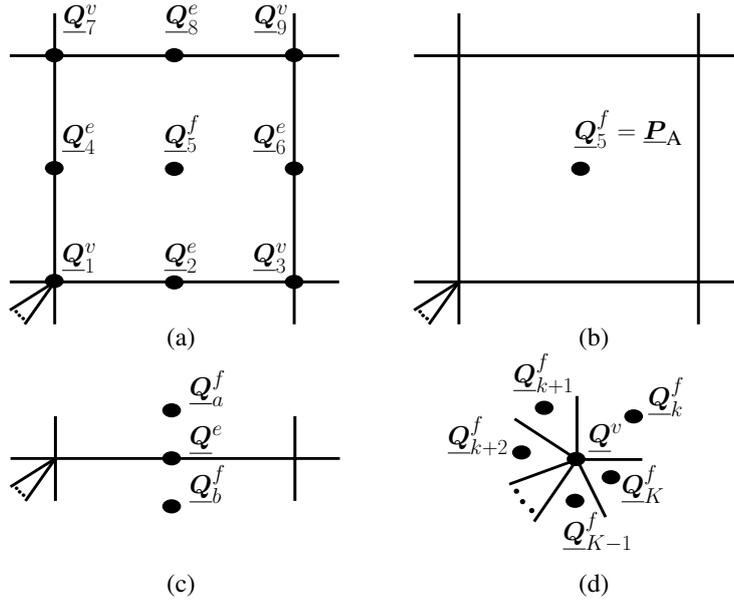


Fig. 4.24: (a) The nine BÉZIER control points \underline{Q}_e for a BÉZIER element. (b) The control point \underline{P}_A corresponds to the anchor in this element. (c) An edge BÉZIER control point \underline{Q}^e is written in terms of neighbouring face BÉZIER control points \underline{Q}^f in Eq. (4.87). (d) A vertex BÉZIER control point \underline{Q}^v is written in terms of neighbouring face BÉZIER control points \underline{Q}^f in Eq. (4.88).

The face point \underline{Q}_5^f (cf. Fig. 4.24(b)) is determined as

$$\underline{Q}_5^f = \underline{P}_A, \tag{4.86}$$

where \underline{P}_A denotes the control point coordinates of anchor A. The edge vertex \underline{Q}^e in Fig. 4.24(c) is computed with

$$\underline{Q}^e = \frac{1}{2}\underline{Q}_a^f + \frac{1}{2}\underline{Q}_b^f, \tag{4.87}$$

and the vertex point \underline{Q}^v of Fig. 4.24(d) is obtained utilising

$$\underline{Q}^v = \sum_{k=1}^K \frac{1}{4}\underline{Q}_k^f, \tag{4.88}$$

where it was assumed that \underline{Q}^v is the vertex of K elements.

After computing \underline{C}_e^T exploiting the generalised BÉZIER extraction for the one-ring neighbourhood elements, the T-spline mesh is non-standard, i. e. the blending functions N do not fulfil the partition of unity. Also, the BÉZIER extraction operators are non-square matrices for the elements in the one-ring neighbourhood of an extraordinary point. \underline{C}_e

has full row rank for the one-ring neighbourhood elements of an extraordinary point with valence three – the blending functions are locally linearly independent. For the one-ring neighbourhood elements of an extraordinary point with valence five, $\underline{\mathbf{C}}_e$ does not have full row rank and therefore, the blending functions are locally linearly dependent. Along spoke edges exists \mathcal{C}_A^0 -continuity. Furthermore, the one-ring neighbourhood elements of the extraordinary point (the irregular elements) are \mathcal{C}_A^1 -continuous with the two-ring neighbourhood elements of the extraordinary points. Next, it will be explained how to enforce \mathcal{C}_A^1 -continuity along spoke edges while preserving \mathcal{C}_A^1 -continuity between the one- and two-ring neighbourhood elements. Moreover, the partition of unity of the blending functions N will be fulfilled.

4.7.3 Modifying the BÉZIER extraction operator

This section shows how to modify the coefficients of the BÉZIER extraction operator $\underline{\mathbf{C}}_e$ in the one-ring neighbourhood elements of an extraordinary point. In a first step, the BÉZIER extraction operator $\underline{\mathbf{C}}_e$ is elevated from degree two to degree four. This is achieved by utilising the degree elevation matrix $\underline{\mathbf{E}}^{2,4}$, cf. THOMAS *et al.* [110]. Degree elevating Eq. (4.31) results in

$$\underline{\mathbf{N}}_e = \underline{\mathbf{C}}_e^2 \underline{\mathbf{B}}_e^2 = \underline{\mathbf{C}}_e^2 \underline{\mathbf{E}}^{2,3} \underline{\mathbf{B}}_e^3 = \underline{\mathbf{C}}_e^2 \underline{\mathbf{E}}^{2,3} \underline{\mathbf{E}}^{3,4} \underline{\mathbf{B}}_e^4 = \underline{\mathbf{C}}_e^2 \underline{\mathbf{E}}^{2,4} \underline{\mathbf{B}}_e^4 = \underline{\mathbf{C}}_e^4 \underline{\mathbf{B}}_e^4, \quad (4.89)$$

where the superscript was added in order to indicate the degree. It can be observed from Eq. (4.89) that degree elevation does not change the blending functions $\underline{\mathbf{N}}_e$. Also, degree elevation does not change local dependencies that may exist, i. e. the row rank of $\underline{\mathbf{C}}_e$ is not affected. After degree elevation, each blending function N with support over a BÉZIER element in the one-ring neighbourhood has 25 BÉZIER coefficients $c_{\alpha,\beta}$ (cf. Eq. (4.30) and Fig. 4.25(a))

$$N(\underline{\boldsymbol{\xi}}) = \sum_{\alpha=1}^5 \sum_{\beta=1}^5 c_{\alpha,\beta} B_{\alpha,\beta}(\underline{\boldsymbol{\xi}}). \quad (4.90)$$

Assume that $a=1 \dots A$ blending functions N^a have a support over *at least two* of the $b=1 \dots B$ one-ring neighbourhood elements and $c=1 \dots C$ blending functions N^c have a support in *only one* one-ring neighbourhood element. Now, the coefficients of the BÉZIER extraction operator of all blending functions N^a and N^c have to be perturbed. The initial coefficients of the BÉZIER extraction operator are denoted with $c_{\alpha,\beta}^{a,b}$, $c_{\alpha,\beta}^{c,b}$ and the modified coefficients with $\tilde{c}_{\alpha,\beta}^{a,b}$, $\tilde{c}_{\alpha,\beta}^{c,b}$. The following constrained linear least square problem needs to be solved for each extraordinary point

$$\min_{\tilde{\mathbf{c}} \in S} \|\underline{\mathbf{F}} \tilde{\mathbf{c}} - \underline{\mathbf{f}}\|_2, \quad S = \left\{ \tilde{\mathbf{c}} \mid \|\underline{\mathbf{G}} \tilde{\mathbf{c}} - \underline{\mathbf{g}}\|_2 = \min \right\}, \quad (4.91)$$

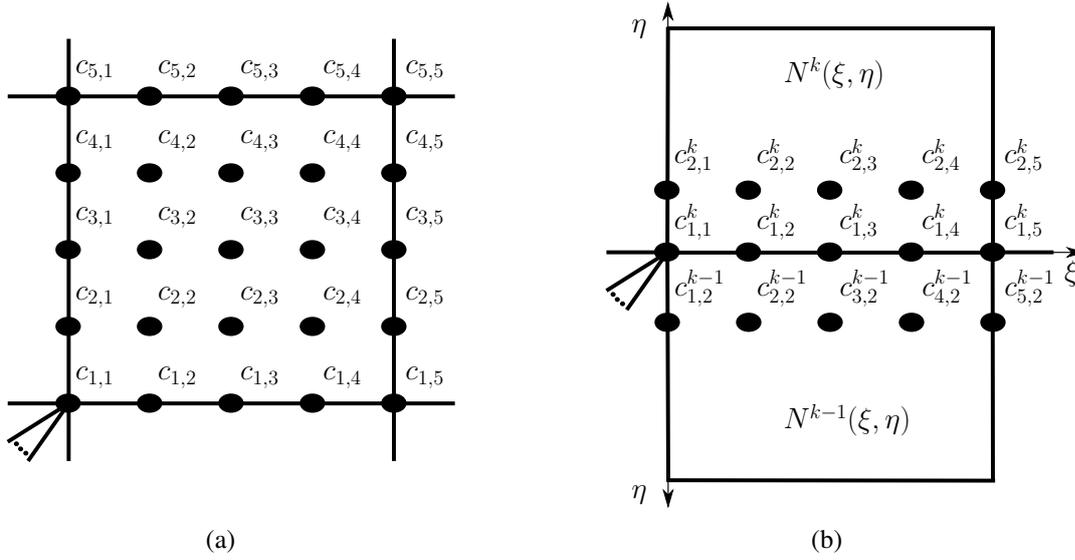


Fig. 4.25: (a) After degree elevation, a blending function with support in a one-ring neighbourhood element has 25 BÉZIER coefficients $c_{\alpha,\beta}$ in each one-ring neighbourhood element. (b) BÉZIER coefficients along a spoke edge that are involved in the \mathcal{G}^1 -continuity constraint of Eq. (4.95).

with the fairing matrix $\underline{\mathbf{F}}$, the fairing vector $\underline{\mathbf{f}}$, the constraint matrix $\underline{\mathbf{G}}$ and the constraint vector $\underline{\mathbf{g}}$. The problem in Eq. (4.91) can be transformed into an unconstrained linear least square problem as explained by BJÖRK [12, Chapter 5].

The following constraints are assembled in $\underline{\mathbf{F}}$ and $\underline{\mathbf{f}}$ in Eq. (4.91)

$$\begin{aligned}
 \tilde{c}_{\alpha,\beta}^{a,b} - \tilde{c}_{\alpha,\beta+1}^{a,b} &= c_{\alpha,\beta}^{a,b} - c_{\alpha,\beta+1}^{a,b} & \text{for } 1 \leq \alpha \leq 5, 1 \leq \beta \leq 4, 1 \leq a \leq A, 1 \leq b \leq B, \\
 \tilde{c}_{\alpha,\beta}^{a,b} - \tilde{c}_{\alpha+1,\beta}^{a,b} &= c_{\alpha,\beta}^{a,b} - c_{\alpha+1,\beta}^{a,b} & \text{for } 1 \leq \alpha \leq 4, 1 \leq \beta \leq 5, 1 \leq a \leq A, 1 \leq b \leq B, \\
 \tilde{c}_{\alpha,\beta}^{c,b} - \tilde{c}_{\alpha+1,\beta}^{c,b} &= c_{\alpha,\beta}^{c,b} - c_{\alpha+1,\beta}^{c,b} & \text{for } 1 \leq \alpha \leq 5, 1 \leq \beta \leq 4, 1 \leq c \leq C, 1 \leq b \leq B, \\
 \tilde{c}_{\alpha,\beta}^{c,b} - \tilde{c}_{\alpha+1,\beta}^{c,b} &= c_{\alpha,\beta}^{c,b} - c_{\alpha+1,\beta}^{c,b} & \text{for } 1 \leq \alpha \leq 4, 1 \leq \beta \leq 5, 1 \leq c \leq C, 1 \leq b \leq B.
 \end{aligned} \tag{4.92}$$

The fairing equations in Eq. (4.92) prevent oscillations between neighbouring coefficients of the BÉZIER extraction operator when perturbing the coefficients $c_{\alpha,\beta}$.

Suppose that $\alpha = 1, \beta = 1$ marks the BÉZIER control point at the extraordinary point, see Fig. 4.25(a). In order to preserve \mathcal{C}_A^1 -continuity between one- and two-ring neighbourhood elements, the constraints

$$\begin{aligned}
 \tilde{c}_{\alpha,\beta}^{a,b} &= c_{\alpha,\beta}^{a,b} & \text{for } 1 \leq \alpha \leq 5, 4 \leq \beta \leq 5, 1 \leq a \leq A, 1 \leq b \leq B, \\
 \tilde{c}_{\alpha,\beta}^{a,b} &= c_{\alpha,\beta}^{a,b} & \text{for } 4 \leq \alpha \leq 5, 2 \leq \beta \leq 3, 1 \leq a \leq A, 1 \leq b \leq B
 \end{aligned} \tag{4.93}$$

are assembled in $\underline{\mathbf{G}}$ and $\underline{\mathbf{g}}$ in Eq. (4.91) for blending functions that are non-zero in *at least two* one-ring neighbourhood elements. Blending functions that are non-zero in *only one*

one-ring neighbourhood element are not allowed to change by enforcing

$$\tilde{c}_{\alpha,\beta}^{c,b} = c_{\alpha,\beta}^{c,b} \quad \text{for } 1 \leq \alpha \leq 5, 1 \leq \beta \leq 5, 1 \leq c \leq C, 1 \leq b \leq B. \quad (4.94)$$

In order to get C_A^1 -continuity along spoke edges (between one-ring neighbourhood elements $k-1$ and k in Fig. 4.25(b)) for the blending function N that is non-zero in *at least two* one-ring neighbourhood elements, the \mathcal{G}^1 -continuity condition (see also DU & SCHMITT [37] or FAUX & PRATT [40, Chapter 7])

$$\begin{aligned} f(\xi) &= r(\xi) \frac{\partial N^{k-1}(\xi, \eta)}{\partial \eta} \Big|_{\eta=0} + s(\xi) \frac{\partial N^k(\xi, \eta)}{\partial \xi} \Big|_{\eta=0} + t(\xi) \frac{\partial N^k(\xi, \eta)}{\partial \eta} \Big|_{\eta=0} \\ &= r(\xi) N_{,\eta}^{k-1}(\xi) + s(\xi) N_{,\xi}^k(\xi) + t(\xi) N_{,\eta}^k(\xi) = 0 \end{aligned} \quad (4.95)$$

can be exploited since it was pointed out by GROISSER & PETERS [46] that this \mathcal{G}^1 -condition yields C_A^1 -continuous blending functions (see also Section 4.1.4).

In the following, the notation

$$\sum_{i=1}^{p+1} c_i B_i^p(\xi) = \langle c_1, c_2, \dots, c_{p+1} \rangle^p(\xi) \quad (4.96)$$

will be used. The polynomials $r(\xi)$, $s(\xi)$ and $t(\xi)$ in Eq. (4.95) are taken as

$$r(\xi) = 1, \quad s(\xi) = \langle \zeta, 0, 0 \rangle^2(\xi), \quad t(\xi) = 1, \quad (4.97)$$

where ζ is computed from

$$\zeta = -2 \cos(\theta), \quad \theta = \frac{2\pi}{B}. \quad (4.98)$$

The three individual terms in Eq. (4.95) result with Eqs. (4.96) and (4.97) in

$$r(\xi) N_{,\eta}^{k-1}(\xi) = 4 \langle \tilde{c}_{1,2}^{k-1} - \tilde{c}_{1,1}^k, \tilde{c}_{2,2}^{k-1} - \tilde{c}_{1,2}^k, \tilde{c}_{3,2}^{k-1} - \tilde{c}_{1,3}^k, \tilde{c}_{4,2}^{k-1} - \tilde{c}_{1,4}^k, \tilde{c}_{5,2}^{k-1} - \tilde{c}_{1,5}^k \rangle^4(\xi), \quad (4.99)$$

$$s(\xi) N_{,\xi}^k(\xi) = \langle \zeta, 0, 0 \rangle^2(\xi) 4 \langle \tilde{c}_{1,2}^k - \tilde{c}_{1,1}^k, \tilde{c}_{1,3}^k - \tilde{c}_{1,2}^k, \tilde{c}_{1,4}^k - \tilde{c}_{1,3}^k, \tilde{c}_{1,5}^k - \tilde{c}_{1,4}^k \rangle^3(\xi), \quad (4.100)$$

$$t(\xi) N_{,\eta}^k(\xi) = 4 \langle \tilde{c}_{2,1}^k - \tilde{c}_{1,1}^k, \tilde{c}_{2,2}^k - \tilde{c}_{1,2}^k, \tilde{c}_{2,3}^k - \tilde{c}_{1,3}^k, \tilde{c}_{2,4}^k - \tilde{c}_{1,4}^k, \tilde{c}_{2,5}^k - \tilde{c}_{1,5}^k \rangle^4(\xi). \quad (4.101)$$

Since the term in Eq. (4.100) is quintic, $N_{,\xi}^k$ needs to be degree reduced utilising the transformation matrix $\underline{\underline{D}}^{3,2}$ (see THOMAS *et al.* [110])

$$\underline{\underline{D}}^{3,2} = (\underline{\underline{E}}^{2,3})^T \left[\underline{\underline{E}}^{2,3} (\underline{\underline{E}}^{2,3})^T \right]^{-1}. \quad (4.102)$$

Degree reducing $N_{,\xi}^k(\xi)$ in Eq. (4.100) results in

$$N_{,\xi}^k(\xi) = 4 \left\langle -\frac{19}{20} \tilde{c}_{1,1}^k + \frac{16}{20} \tilde{c}_{1,2}^k + \frac{6}{20} \tilde{c}_{1,3}^k - \frac{4}{20} \tilde{c}_{1,4}^k + \frac{1}{20} \tilde{c}_{1,5}^k, \right. \\ \left. \frac{5}{20} \tilde{c}_{1,1}^k - \tilde{c}_{1,2}^k + \tilde{c}_{1,4}^k - \frac{5}{20} \tilde{c}_{1,5}^k, \right. \\ \left. -\frac{1}{20} \tilde{c}_{1,1}^k + \frac{4}{20} \tilde{c}_{1,2}^k - \frac{6}{20} \tilde{c}_{1,3}^k - \frac{16}{20} \tilde{c}_{1,4}^k + \frac{19}{20} \tilde{c}_{1,5}^k \right\rangle^2(\xi), \quad (4.103)$$

which gives for Eq. (4.100)

$$s(\xi) N_{,\xi}^k(\xi) = 4 \zeta \left\langle -\frac{19}{20} \tilde{c}_{1,1}^k + \frac{16}{20} \tilde{c}_{1,2}^k + \frac{6}{20} \tilde{c}_{1,3}^k - \frac{4}{20} \tilde{c}_{1,4}^k + \frac{1}{20} \tilde{c}_{1,5}^k, \right. \\ \left. \frac{5}{40} \tilde{c}_{1,1}^k - \frac{1}{2} \tilde{c}_{1,2}^k + \frac{1}{2} \tilde{c}_{1,4}^k - \frac{5}{40} \tilde{c}_{1,5}^k, \right. \\ \left. -\frac{1}{120} \tilde{c}_{1,1}^k + \frac{4}{120} \tilde{c}_{1,2}^k - \frac{6}{120} \tilde{c}_{1,3}^k - \frac{16}{120} \tilde{c}_{1,4}^k + \frac{19}{120} \tilde{c}_{1,5}^k, 0, 0 \right\rangle^4(\xi). \quad (4.104)$$

In order to fulfil Eq. (4.95), the following terms must vanish exploiting Eqs. (4.99), (4.101) and (4.104)

$$\tilde{c}_{1,2}^{k-1} - \tilde{c}_{1,1}^k + \zeta \left(-\frac{19}{20} \tilde{c}_{1,1}^k + \frac{16}{20} \tilde{c}_{1,2}^k + \frac{6}{20} \tilde{c}_{1,3}^k - \frac{4}{20} \tilde{c}_{1,4}^k + \frac{1}{20} \tilde{c}_{1,5}^k \right) + \tilde{c}_{2,1}^k - \tilde{c}_{1,1}^k = 0, \quad (4.105)$$

$$4 \left(\tilde{c}_{2,2}^{k-1} - \tilde{c}_{1,2}^k \right) + \zeta \left(\frac{1}{2} \tilde{c}_{1,1}^k - 2 \tilde{c}_{1,2}^k + 2 \tilde{c}_{1,4}^k - \frac{1}{2} \tilde{c}_{1,5}^k \right) + 4 \left(\tilde{c}_{2,2}^k - \tilde{c}_{1,2}^k \right) = 0, \quad (4.106)$$

$$4 \left(\tilde{c}_{3,2}^{k-1} - \tilde{c}_{1,3}^k \right) + \zeta \left(-\frac{1}{30} \tilde{c}_{1,1}^k + \frac{4}{30} \tilde{c}_{1,2}^k - \frac{6}{30} \tilde{c}_{1,3}^k - \frac{16}{30} \tilde{c}_{1,4}^k + \frac{19}{30} \tilde{c}_{1,5}^k \right) + 4 \left(\tilde{c}_{2,3}^k - \tilde{c}_{1,3}^k \right) = 0, \quad (4.107)$$

$$\tilde{c}_{4,2}^{k-1} - \tilde{c}_{1,4}^k + \tilde{c}_{2,4}^k - \tilde{c}_{1,4}^k = 0, \quad (4.108)$$

$$\tilde{c}_{5,2}^{k-1} - \tilde{c}_{1,5}^k + \tilde{c}_{2,5}^k - \tilde{c}_{1,5}^k = 0. \quad (4.109)$$

Moreover, the fourth derivative of $f(\xi)$ has to vanish. This constraint results in

$$\tilde{c}_{1,1}^k - 4 \tilde{c}_{1,2}^k + 6 \tilde{c}_{1,3}^k - 4 \tilde{c}_{1,4}^k + \tilde{c}_{1,5}^k = 0. \quad (4.110)$$

Eqs. (4.105) – (4.110) need to be assembled in $\underline{\mathbf{G}}$ and $\underline{\mathbf{g}}$ along all spoke edges for all blending functions that are non-zero in *at least two* one-ring neighbourhood elements.

In order to fulfil the partition of unity, the following equation must be satisfied for all blending function that are non-zero in *at least one* one-ring neighbourhood element

$$\sum_{a=1}^A \tilde{c}_{\alpha,\beta}^{a,b} + \sum_{c=1}^C \tilde{c}_{\alpha,\beta}^{c,b} = 1 \quad \begin{cases} \text{for } 1 \leq \alpha \leq 3, 1 \leq \beta \leq 3 \\ \text{in all one-ring neighbourhood elements } b = 1 \dots B. \end{cases} \quad (4.111)$$

Eq. (4.111) follows from Section 4.4.2.3. Computing the term

$$\|\underline{\mathbf{G}}\tilde{\mathbf{c}} - \underline{\mathbf{g}}\|_2 \quad (4.112)$$

after solving Eq. (4.91) yields zero within machine precision, i. e. all imposed constraints are satisfied.

After solving the constrained linear least square problem in Eq. (4.91), the support of the blending functions that are non-zero in *two* one-ring neighbourhood elements has changed, see Fig. 4.26.

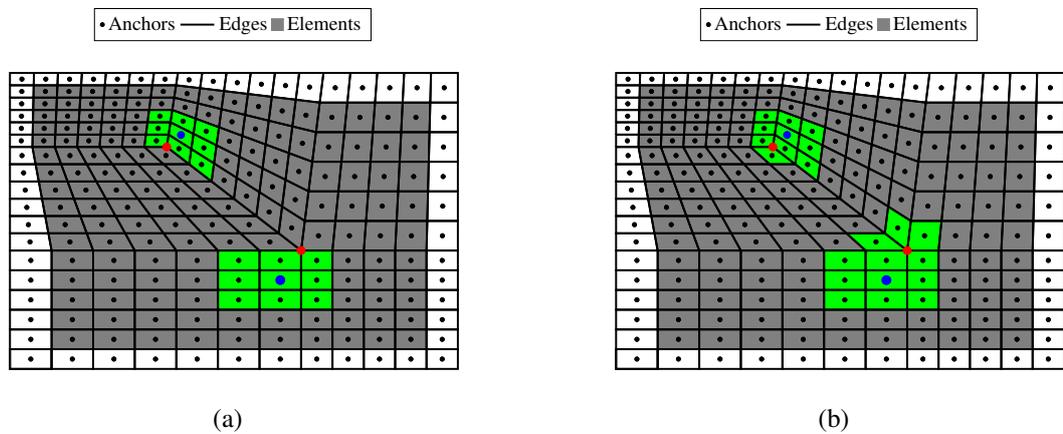


Fig. 4.26: Modifying the BÉZIER coefficients $c_{\alpha,\beta}$ in the one-ring neighbourhood elements of an extraordinary point (red) results in a modified support of blending functions N that are non-zero in *two* one-ring neighbourhood elements after generalised BÉZIER extraction: green marks the support of the two blue blending functions (a) before and (b) after solving the constrained least square problem in Eq. (4.91).

Moreover, the BÉZIER extraction operators $\underline{\mathbf{C}}_e$ for the one-ring neighbourhood elements of an extraordinary point are still not square matrices, i. e. hierarchical refinement (Section 4.6) or BÉZIER projection (THOMAS *et al.* [110]) are not applicable. For the one-ring neighbourhood elements of the extraordinary point with valence three, the BÉZIER extraction operator $\underline{\mathbf{C}}_e$ has full row rank – the blending functions N are locally linearly independent. The blending functions N are locally linearly dependent for the one-ring neighbourhood elements of the extraordinary point with valence five since $\underline{\mathbf{C}}_e$ does not have full row rank.

Figs. 4.27 and 4.28 show a blending function N and its first derivatives $N_{,x}$ and $N_{,y}$ in the physical domain before and after modifying the BÉZIER coefficients $c_{\alpha,\beta}$. It can be seen that $N_{,x}$ and $N_{,y}$ are continuous after smoothing. Thus, the blending functions N are \mathcal{C}_A^1 -continuous.

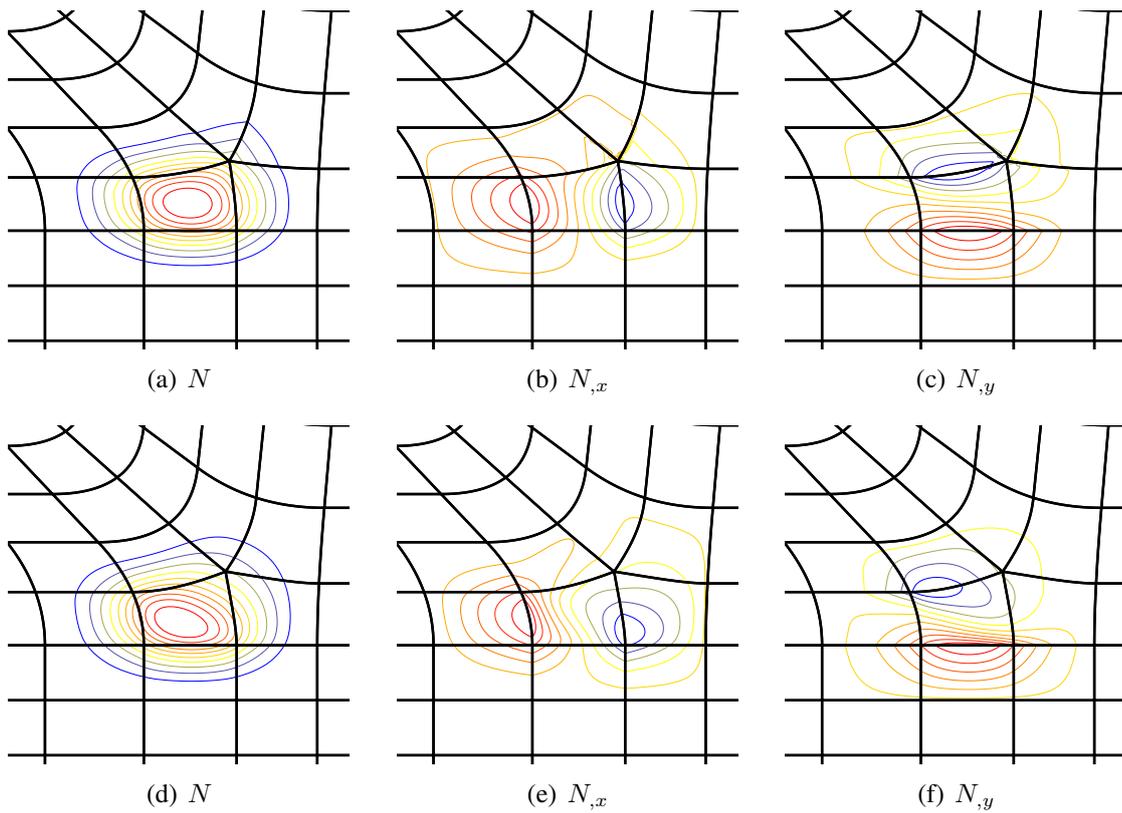


Fig. 4.27: Contour plots of a quadratic blending function N and its derivatives $N_{,x}$, $N_{,y}$ in the physical domain before (a)-(c) and after (d)-(f) smoothing. The blending function corresponds to an anchor that is located in the one-ring neighbourhood of the extraordinary point of valence five.

4.8 Analysis-suitable T-splines

The blending functions of an analysis-suitable T-spline mesh fulfil the partition of unity. Analysis-suitable T-splines have been defined by LI *et al.* [66]. In order to detect them, the extended T-spline mesh was introduced, and a mesh was deemed analysis-suitable when there are no two orthogonal T-node extensions which intersect in the extended T-spline mesh. This definition holds for any knot interval and is of topological nature; it allows to distinguish between analysis-suitable and non-analysis-suitable T-splines.

The approach presented in the previous sections is based on the BÉZIER extraction operator. It is an algebraic viewpoint and allows the determination of linear dependencies (see Section 4.4.1) and the classification of T-splines into standard, semi-standard and non-standard with Eq. (4.50).

Fig. 4.29 reveals that a standard T-spline is not necessarily an analysis-suitable T-spline. In Fig. 4.29, T-node extensions intersect in the extended T-spline mesh and the T-spline meshes are therefore non-analysis-suitable. It was observed in Figs. 4.11(a), 4.11(c)

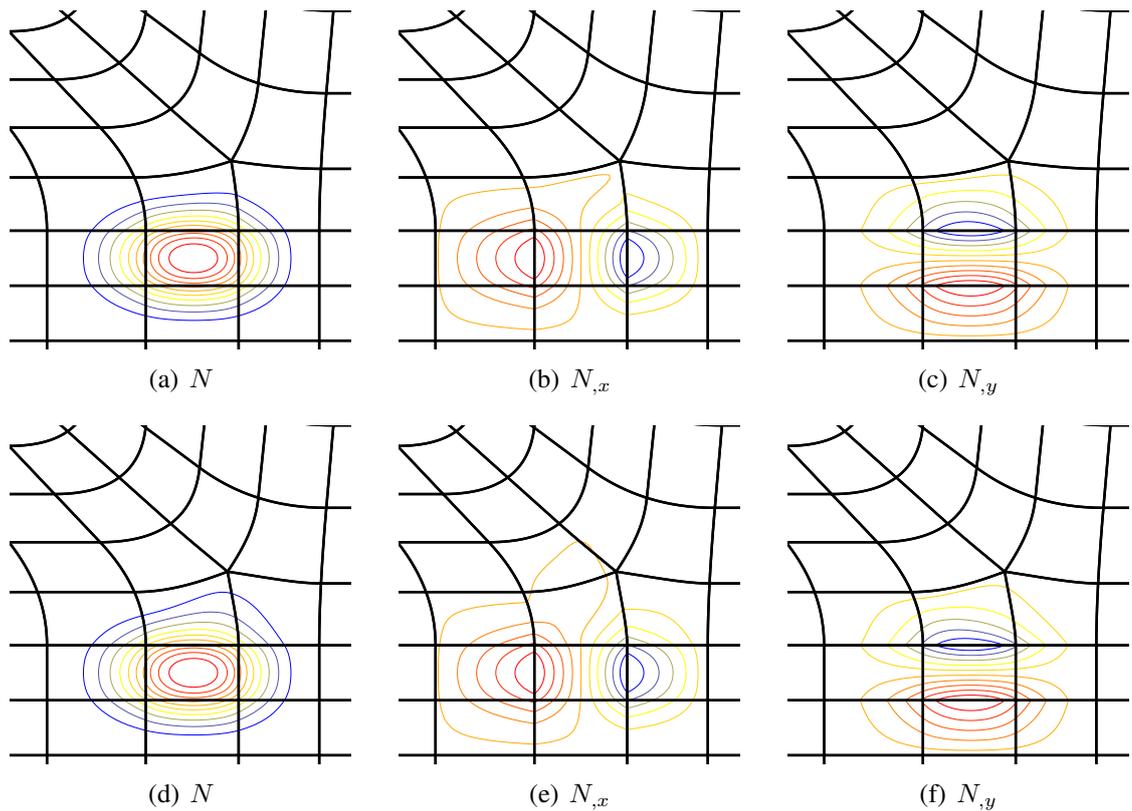


Fig. 4.28: Contour plots of a quadratic blending function N and its derivatives $N_{,x}$, $N_{,y}$ in the physical domain before (a)-(c) and after (d)-(f) smoothing. The blending function corresponds to an anchor that is located in the two-ring neighbourhood of the extraordinary point of valence five.

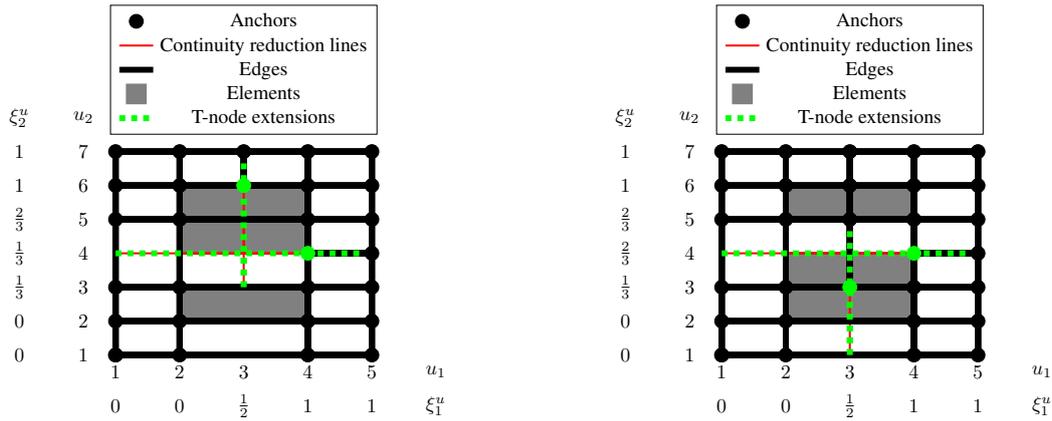
and 4.11(e) that the T-spline meshes in Fig. 4.29 are standard.

Notice that the term “analysis-suitable” might cause confusion. Analysis can also be performed with non-analysis-suitable T-spline meshes. For instance, the unstructured T-spline meshes utilised by BAZILEVS *et al.* [10], CASQUERO *et al.* [21], DIMITRI *et al.* [34], DIMITRI [35], HSU *et al.* [51], KIENDL *et al.* [59], KOSTAS *et al.* [61], SCOTT *et al.* [97, 98], SIMPSON *et al.* [103] and THOMAS & SCOTT [111] are non-analysis-suitable T-spline meshes since the blending functions N^i do not fulfil the partition of unity in the one-ring neighbourhood elements of an extraordinary point (see also Section 4.7 and Appendix B.3).

The requirements for a T-spline mesh to be suitable for analysis are

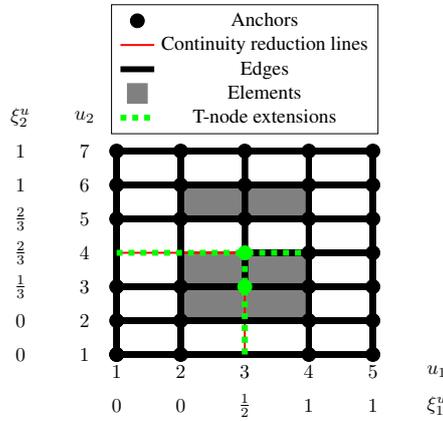
- the blending functions N^i are globally linearly independent (Eq. (4.37) holds)
- the partition of unity property holds in order to satisfy the affine transformation.

The local / global linear independence of the blending functions N^i results in the local / global linear independence of the rational blending functions R^i . Hence, globally



(a) T-node extensions for the T-spline mesh in Fig. 4.11(a).

(b) T-node extensions for the T-spline mesh in Fig. 4.11(c).



(c) T-node extensions for the T-spline mesh in Fig. 4.11(e).

Fig. 4.29: Extended T-spline meshes for Fig. 4.11(a), 4.11(c) and 4.11(e); these standard T-spline meshes are non-analysis-suitable according to LI *et al.* [66] since T-node extensions intersect in the extended T-spline mesh.

linearly independent semi-standard and non-standard T-spline meshes which employ the rational blending functions R^i in Eq. (4.44) can be used for analysis since the rational blending functions R^i always form a partition of unity.

Furthermore, LI [67] concluded that the unstructured T-spline mesh in SCOTT *et al.* [97] is an analysis-suitable T-spline. Unfortunately, LI [67] did not take into account that the blending functions of an analysis-suitable T-spline also fulfil the partition of unity which is not the case for the unstructured T-spline meshes in SCOTT *et al.* [97]. This shows that it may not be possible to conclude from the topology of an unstructured T-spline mesh whether it fulfils the partition of unity or not and that instead, the BÉZIER extraction operator should be exploited.

5 POWELL–SABIN B-splines

KIRCHHOFF–LOVE plate theory [60, 70] is suited for the analysis of thin plates, but requires C_A^1 -continuous basis functions in finite element analyses. The subdivision surfaces technique has been applied to KIRCHHOFF–LOVE shell elements by CIRAK *et al.* [24]. A NURBS-based formulation for KIRCHHOFF–LOVE shell elements was proposed by KIENDL *et al.* [57]. Since multiple NURBS-patches are joined with C_A^0 -continuity, GUO & RUESS [48] used NITSCHKE’s method [80] for coupling NURBS-patches. Alternatively, the bending strip method was developed by KIENDL *et al.* [58] which adds a penalty stiffness between adjacent NURBS-patches. This approach requires a geometric continuity \mathcal{G}^1 along the C_A^0 -continuity joints. Hence, the control points need to be positioned such that a \mathcal{G}^1 -continuity is obtained. Unfortunately, this (\mathcal{G}^1, C_A^0) -construction has been applied only to two or four adjacent NURBS-patches – it has not been demonstrated how to position the control points when three or more than four NURBS-patches meet at an extraordinary point.

T-splines are an alternative to NURBS and consist of a single patch. It was further demonstrated in Section 4.7 how to obtain C_A^1 -continuity around extraordinary points for unstructured quadratic T-splines by enforcing \mathcal{G}^1 -continuous blending functions. This construction also fulfils the partition of unity of the blending functions.

While T-splines have a layout that is based on a quadrilateral structure, quadratic POWELL–SABIN [88] B-splines are based on a linear triangulation. They likewise provide basis functions that are C_A^1 -continuous across elements. POWELL–SABIN B-splines have been cast in terms of BÉZIER ordinates by DIERCKX *et al.* [32] and DIERCKX [33] for an efficient calculation. POWELL–SABIN B-splines are not based on the isogeometric concept as they only *approximate* the exact geometry. To address this issue the NURBS-to-NURPS method was recently proposed by SPELEERS *et al.* [106] which transforms a single NURBS-patch into Non-Uniform Rational POWELL–SABIN B-splines (NURPS). The boundary of the NURBS-to-NURPS then exactly matches the boundary of the NURBS-patch while the interior domain of the NURBS-to-NURPS only approximates the NURBS-patch which is relevant for non-planar geometries. POWELL–SABIN B-splines and NURBS-to-NURPS have been used for analysis by STOGNER & CAREY [107] and SPELEERS *et al.* [104–106].

Currently, POWELL–SABIN B-splines cannot be obtained from arbitrary tetrahedral meshes in the three-dimensional case since certain constraints with neighbouring tetrahedrons have to be fulfilled, see WORSEY & PIPER [124]. Hence, three-dimensional

POWELL–SABIN B-splines are presently available only for structured meshes. Moreover, there is still no “perfect” C_A^1 -continuous tetrahedron, i. e. a tetrahedron with low polynomial degree, little splits and applicable to arbitrary meshes (see also the overview by ALFELD & SOROKINA [1]).

The KIRCHHOFF–LOVE plate theory is introduced in the next section. Then, it will be shown how BÉZIER extraction for POWELL–SABIN B-splines can be implemented, thus making the method computationally efficient, similar to BÉZIER extraction for NURBS and T-splines by BORDEN *et al.* [13] and SCOTT *et al.* [94]. The performance of POWELL–SABIN B-splines is compared with NURBS, unstructured T-splines and NURBS-to-NURPS when solving the KIRCHHOFF–LOVE plate with clamped and simply supported boundary conditions. This study goes beyond the study by NGUYEN *et al.* [79], which was for the POISSON equation, and hence only required C_A^0 -continuous basis functions.

5.1 KIRCHHOFF–LOVE plate theory

The continuum and finite element formulation are derived for the KIRCHHOFF–LOVE plate theory in this section.

5.1.1 Continuum formulation

The moment equilibrium for a plate reads (cf. GROSS *et al.* [47, Chapter 3])

$$m_{\alpha\beta,\alpha\beta} = p_0, \quad (5.1)$$

with the force per unit area p_0 and the bending moments

$$m_{\alpha\beta} = - \int_{-h/2}^{h/2} \sigma_{\alpha\beta} z \, dz. \quad (5.2)$$

Greek indices take values one and two, a comma denotes differentiation, while h is the thickness of the plate. $\sigma_{\alpha\beta}$ is the stress, and HOOKE’s law for plane stress with the YOUNG’s modulus E and POISSON’s ratio ν is used

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{bmatrix}, \quad (5.3)$$

since it is assumed that $|\sigma_{33}|, |\sigma_{13}|, |\sigma_{23}| \ll |\sigma_{11}|, |\sigma_{22}|, |\sigma_{12}|$. The non-zero strain components are assumed as follows

$$\varepsilon_{11} = -z \frac{\partial^2 w}{\partial x^2}, \quad \varepsilon_{22} = -z \frac{\partial^2 w}{\partial y^2}, \quad 2\varepsilon_{12} = -2z \frac{\partial^2 w}{\partial x \partial y}, \quad (5.4)$$

with the deflection $w(x, y)$. Plugging Eqs. (5.3) and (5.4) into Eq. (5.2) yields

$$\underbrace{\begin{bmatrix} m_{11} \\ m_{22} \\ m_{12} \end{bmatrix}}_{\underline{\mathbf{m}}} = \frac{Eh^3}{12(1-\nu^2)} \underbrace{\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}}_{\underline{\mathbf{D}}} \underbrace{\begin{bmatrix} \kappa_{11} \\ \kappa_{22} \\ 2\kappa_{12} \end{bmatrix}}_{\underline{\mathbf{\kappa}}}, \quad (5.5)$$

with the elastic stiffness matrix $\underline{\mathbf{D}}$ and the curvature

$$\kappa_{\alpha\beta} = w_{,\alpha\beta}. \quad (5.6)$$

The bending moments are assembled in the vector $\underline{\mathbf{m}}$, while $\underline{\mathbf{\kappa}}$ contains the curvatures. Multiplying Eq. (5.1) by a test function δw , integrating over the domain Ω and twice by parts results with GAUSS' theorem in

$$\int_{\Omega} \delta w_{,\alpha\beta} m_{\alpha\beta} \, dx dy + \int_{\partial\Omega} \delta w m_{\alpha\beta, \alpha} n_{\beta} \, d\Gamma - \int_{\partial\Omega} \delta w_{,\beta} m_{\alpha\beta} n_{\alpha} \, d\Gamma = \int_{\Omega} \delta w p_0 \, dx dy, \quad (5.7)$$

where \mathbf{n} is the normal vector on the boundary. Boundary terms are neglected since no moments or forces are imposed on $\partial\Omega$ in Section 5.4. Substitution of Eq. (5.6) into Eq. (5.7), results into the weak form

$$\int_{\Omega} \delta \kappa_{\alpha\beta} m_{\alpha\beta} \, dx dy = \int_{\Omega} \delta w p_0 \, dx dy. \quad (5.8)$$

Since second derivatives appear in this equation, C_A^1 -continuous functions (functions of class H^2) are necessary with square integrable second derivatives (cf. FISH & BELYTSCHKO [42, Chapter 3]).

5.1.2 Finite element formulation

Discretisation of the domain Ω into E elements,

$$\Omega = \bigcup_{e=1}^E \Omega^e, \quad (5.9)$$

and approximation of the deflection w and its derivatives,

$$w^e = \underline{\mathbf{N}}^T \underline{\mathbf{w}}, \quad \delta w^e = \underline{\mathbf{N}}^T \delta \underline{\mathbf{w}}, \quad \underline{\boldsymbol{\kappa}}^e = \underline{\mathbf{B}} \underline{\mathbf{w}}, \quad \delta \underline{\boldsymbol{\kappa}}^e = \underline{\mathbf{B}} \delta \underline{\mathbf{w}}, \quad (5.10)$$

where

$$\underline{\mathbf{N}}^T = \begin{bmatrix} N_1 & N_2 & \dots \end{bmatrix}, \quad \underline{\mathbf{B}} = \begin{bmatrix} N_{1,11} & N_{2,11} & \dots \\ N_{1,22} & N_{2,22} & \dots \\ 2N_{1,12} & 2N_{2,12} & \dots \end{bmatrix} \quad (5.11)$$

results with Eq. (5.5) in the following matrix-vector equation for Eq. (5.8)

$$\int_{\Omega} \delta \underline{\boldsymbol{\kappa}}^T \underline{\mathbf{D}} \underline{\boldsymbol{\kappa}} \, dx dy = \int_{\Omega} \delta \underline{\mathbf{w}}^T \underline{\mathbf{N}} p_0 \, dx dy, \quad (5.12)$$

which gives for arbitrary $\delta \underline{\mathbf{w}}$

$$\underbrace{\int_{\Omega} \underline{\mathbf{B}}^T \underline{\mathbf{D}} \underline{\mathbf{B}} \, dx dy}_{\underline{\mathbf{K}}} \underline{\mathbf{w}} = \underbrace{\int_{\Omega} \underline{\mathbf{N}} p_0 \, dx dy}_{\underline{\mathbf{f}}}, \quad (5.13)$$

with the stiffness matrix $\underline{\mathbf{K}}$ and the force vector $\underline{\mathbf{f}}$.

5.2 BÉZIER extraction for quadratic POWELL–SABIN B-splines

This section starts with a concise description of POWELL–SABIN B-splines, including notions like POWELL–SABIN refinement, POWELL–SABIN points, and POWELL–SABIN triangles. For an in-depth treatment reference is made to DIERCKX [33]. Then, BÉZIER extraction for POWELL–SABIN B-splines is introduced, followed by a discussion on the patch test for POWELL–SABIN B-splines.

5.2.1 Quadratic POWELL–SABIN B-splines

In order to obtain basis functions with C_A^1 -continuity for a triangulation \mathcal{T} , some pre-processing steps are necessary. Consider the parameter domain $\underline{\boldsymbol{\xi}} = (\xi, \eta)$ for a triangulation \mathcal{T} (thick black lines) with $e = 1, \dots, E$ triangles and N_v vertices in Fig. 5.1(a). A vertex k of the triangulation \mathcal{T} has the coordinates $\underline{\mathbf{V}}_k = (\xi_k, \eta_k)$ in the parameter domain. m_k triangles are attached to vertex k and will be denoted as the molecule Ω_k .

Each triangle e of the triangulation \mathcal{T} has a barycentric coordinate system with $\underline{\boldsymbol{\tau}} = (\tau_1, \tau_2, \tau_3)$, see Fig. 5.1(c). The points $\underline{\mathbf{V}}_1, \underline{\mathbf{V}}_2, \underline{\mathbf{V}}_3, \underline{\mathbf{R}}_{12}, \underline{\mathbf{R}}_{23}, \underline{\mathbf{R}}_{31}$ and $\underline{\mathbf{Z}}$ have

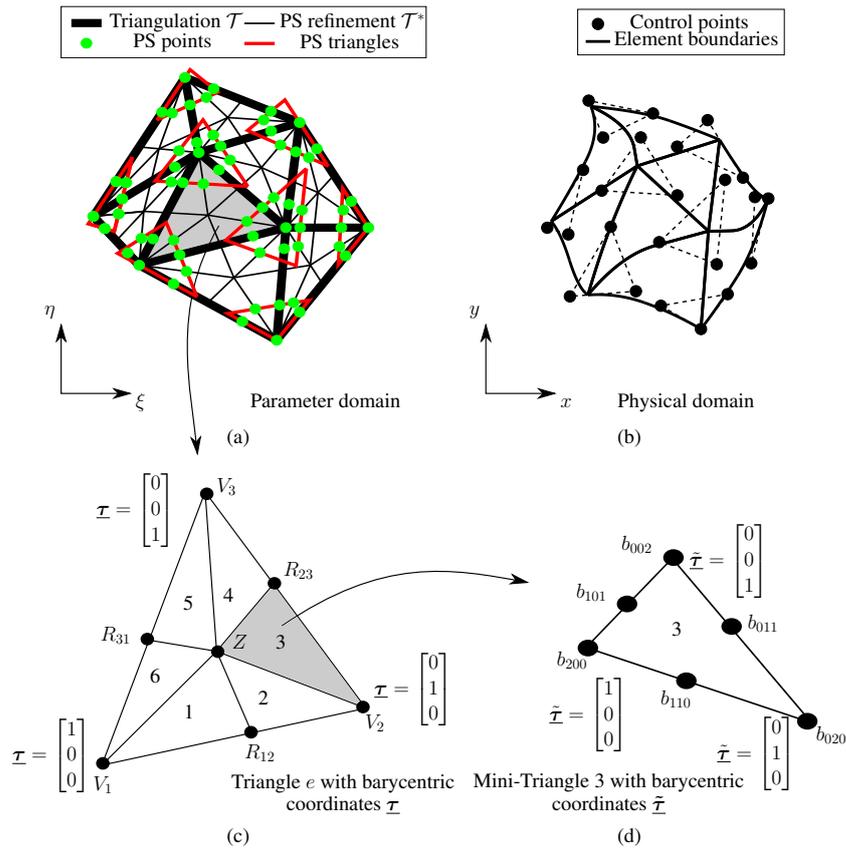


Fig. 5.1: The (a) parameter and (b) physical domains, $(\underline{\xi})$ and (\underline{x}) , respectively, for POWELL–SABIN B-splines. (c) Each triangle e has a barycentric coordinate system $\underline{\tau}$ and (d) can further be subdivided into six mini-triangles with a barycentric coordinate system $\tilde{\underline{\tau}}$.

the following barycentric coordinates (τ_1, τ_2, τ_3)

$$\begin{aligned} \underline{V}_1 &= (1, 0, 0), & \underline{V}_2 &= (0, 1, 0), & \underline{V}_3 &= (0, 0, 1), \\ \underline{R}_{12} &= (\lambda_1, \lambda_2, 0), & \underline{R}_{23} &= (0, \mu_2, \mu_3), & \underline{R}_{31} &= (\nu_1, 0, \nu_3), & \underline{Z} &= (a, b, c). \end{aligned} \quad (5.14)$$

Splitting each triangle e of the triangulation \mathcal{T} in Fig. 5.1(a) into six mini-triangles ($n=1, \dots, 6$) yields the POWELL–SABIN refinement \mathcal{T}^* (thin black lines). Each mini-triangle n has a barycentric coordinate system $\tilde{\underline{\tau}} = (\tilde{\tau}_1, \tilde{\tau}_2, \tilde{\tau}_3)$ and BÉZIER ordinates $b_{r,s,t}$, cf. Fig. 5.1(d).

POWELL–SABIN points (denoted by green dots) are now defined for each vertex k of the triangulation \mathcal{T} in Fig. 5.1(a): they are the vertex k itself and the midpoints of all edges of the POWELL–SABIN refinement \mathcal{T}^* containing the vertex k . A POWELL–SABIN triangle (shown in red), which contains all POWELL–SABIN points, is associated with each vertex k . The POWELL–SABIN triangle needs to contain all POWELL–SABIN points in order to get positive basis functions. Furthermore, the triangle needs to be small for a low

condition number of the stiffness matrix $\underline{\mathbf{K}}$ in Eq. (5.13). The POWELL–SABIN triangles are chosen such that they share two edges with the convex hull of the POWELL–SABIN points, see SPELEERS *et al.* [105] and VANRAES *et al.* [113]. In this way the solution of the expensive optimisation algorithm in DIERCKX [33] is avoided.

Three ($j = 1, 2, 3$) POWELL–SABIN B-splines $N_k^j(\underline{\xi})$ are associated to each vertex k , i. e. one for each corner of the POWELL–SABIN triangle of vertex k , and have a support in the molecule Ω_k of $\underline{\mathbf{V}}_k$. For an elaborate demonstration for the derivation of the calculation of the POWELL–SABIN B-splines $N_k^j(\underline{\xi})$ reference is made to DIERCKX [33]. Here, only a brief outline of the procedure is given.

A POWELL–SABIN B-spline $N_k^j(\underline{\xi})$ in the parameter domain $\underline{\xi}$ must be defined in each of the m_k triangles of the molecule Ω_k of vertex k ,

$$N_k^j(\underline{\xi}) = N_k^j(\underline{\xi}(\underline{\tau})) = \sum_{e_k=1}^{m_k} N_k^{j,e_k}(\underline{\tau}). \quad (5.15)$$

Since each triangle of the molecule Ω_k is split into six mini-triangles, the POWELL–SABIN B-splines $N_k^{j,e_k}(\underline{\tau})$ over a triangle e_k can be written as

$$N_k^{j,e_k}(\underline{\tau}) = N_k^{j,e_k}(\underline{\tau}(\tilde{\tau})) = \sum_{n=1}^6 N_{k,n}^{j,e_k}(\tilde{\tau}). \quad (5.16)$$

For clarity of notation, the indices e_k in $N_{k,n}^{j,e_k}(\tilde{\tau})$ are omitted in the following.

5.2.2 BÉZIER extraction

The POWELL–SABIN B-splines over each mini-triangle, $N_k^j(\tilde{\tau})$ in Eq. (5.16), can be expressed using the BÉZIER ordinates $b_{r,s,t}$, see Fig. 5.1(d),

$$N_k^j(\tilde{\tau}) = \sum_{r+s+t=2} b_{r,s,t} B_{r,s,t}^2(\tilde{\tau}), \quad (5.17)$$

where $B_{r,s,t}^2(\tilde{\tau})$ denote the BERNSTEIN polynomials of degree two

$$B_{r,s,t}^2(\tilde{\tau}) = \frac{2!}{r!s!t!} \tilde{\tau}_1^r \tilde{\tau}_2^s \tilde{\tau}_3^t. \quad (5.18)$$

In order to determine the BÉZIER ordinates $b_{r,s,t}$ in Eq. (5.17), the following properties are assigned to the POWELL–SABIN B-splines: for any vertex $l \neq k$

$$N_k^j(V_l) = 0, \quad \frac{\partial}{\partial \xi} N_k^j(V_l) = 0, \quad \frac{\partial}{\partial \eta} N_k^j(V_l) = 0, \quad (5.19)$$

and otherwise

$$N_k^j(V_k) = \alpha_k^j, \quad \frac{\partial}{\partial \xi} N_k^j(V_k) = \beta_k^j, \quad \frac{\partial}{\partial \eta} N_k^j(V_k) = \gamma_k^j, \quad (5.20)$$

with

$$\sum_{j=1}^3 \alpha_k^j = 1, \quad \sum_{j=1}^3 \beta_k^j = 0, \quad \sum_{j=1}^3 \gamma_k^j = 0. \quad (5.21)$$

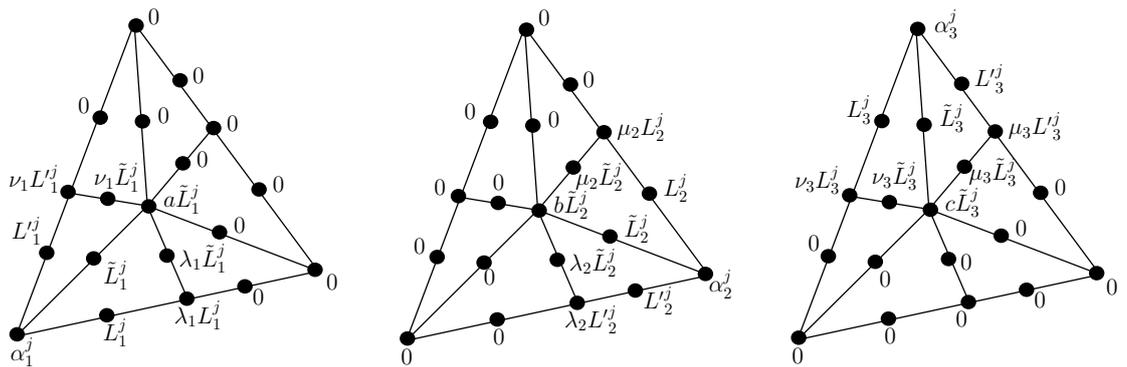
The corners of each POWELL–SABIN triangle (red in Fig. 5.1(a)) have the coordinates $\underline{Q}_k^j = (\xi_k^j, \eta_k^j)$, which gives the map from the mini-triangle domain $\tilde{\tau}$ to the parameter domain $\underline{\xi}$ for a surface $\underline{S}_\xi(\tilde{\tau})$ as follows

$$\underline{S}_\xi(\tilde{\tau}) = \sum_{k=1}^{N_v} \sum_{j=1}^3 N_k^j(\tilde{\tau}) \underline{Q}_k^j. \quad (5.22)$$

For each vertex k with coordinates (ξ_k, η_k) , the parameters α_k^j , β_k^j and γ_k^j in Eq. (5.21) are then obtained by solving

$$\begin{bmatrix} \alpha_k^1 & \alpha_k^2 & \alpha_k^3 \\ \beta_k^1 & \beta_k^2 & \beta_k^3 \\ \gamma_k^1 & \gamma_k^2 & \gamma_k^3 \end{bmatrix} \begin{bmatrix} \xi_k^1 & \eta_k^1 & 1 \\ \xi_k^2 & \eta_k^2 & 1 \\ \xi_k^3 & \eta_k^3 & 1 \end{bmatrix} = \begin{bmatrix} \xi_k & \eta_k & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \quad (5.23)$$

This equation follows by combining Eqs. (5.20) – (5.22). Using Eqs. (5.14) and (5.23), the BÉZIER ordinates $b_{r,s,t}$ of the mini-triangles in Fig. 5.2 can be evaluated.



(a) BÉZIER ordinates for the three POWELL–SABIN B-splines $N_1^j(\tilde{\tau})$ corresponding to vertex V_1 .

(b) BÉZIER ordinates for the three POWELL–SABIN B-splines $N_2^j(\tilde{\tau})$ corresponding to vertex V_2 .

(c) BÉZIER ordinates for the three POWELL–SABIN B-splines $N_3^j(\tilde{\tau})$ corresponding to vertex V_3 .

Fig. 5.2: BÉZIER ordinates $b_{r,s,t}$ for the six $(n=1, \dots, 6)$ mini-triangles of the POWELL–SABIN B-splines $N_1^j(\tilde{\tau})$, $N_2^j(\tilde{\tau})$ and $N_3^j(\tilde{\tau})$.

The BÉZIER ordinates corresponding to V_1 are given by

$$L_1^j = \alpha_1^j + \frac{1 - \lambda_1}{2} \bar{\beta}_1^j, \quad L_1^{\prime j} = \alpha_1^j + \frac{(1 - \nu_1)}{2} \bar{\gamma}_1^j, \quad \tilde{L}_1^j = \alpha_1^j + \frac{b}{2} \bar{\beta}_1^j + \frac{c}{2} \bar{\gamma}_1^j, \quad (5.24)$$

$$\bar{\beta}_1^j = \beta_1^j(\xi_2 - \xi_1) + \gamma_1^j(\eta_2 - \eta_1), \quad \bar{\gamma}_1^j = \beta_1^j(\xi_3 - \xi_1) + \gamma_1^j(\eta_3 - \eta_1), \quad (5.25)$$

while for V_2 they read,

$$L_2^j = \alpha_2^j + \frac{1 - \mu_2}{2} \bar{\beta}_2^j, \quad L_2^{\prime j} = \alpha_2^j + \frac{(1 - \lambda_2)}{2} \bar{\gamma}_2^j, \quad \tilde{L}_2^j = \alpha_2^j + \frac{c}{2} \bar{\beta}_2^j + \frac{a}{2} \bar{\gamma}_2^j, \quad (5.26)$$

$$\bar{\beta}_2^j = \beta_2^j(\xi_3 - \xi_2) + \gamma_2^j(\eta_3 - \eta_2), \quad \bar{\gamma}_2^j = \beta_2^j(\xi_1 - \xi_2) + \gamma_2^j(\eta_1 - \eta_2), \quad (5.27)$$

and for V_3 they are

$$L_3^j = \alpha_3^j + \frac{1 - \nu_3}{2} \bar{\beta}_3^j, \quad L_3^{\prime j} = \alpha_3^j + \frac{(1 - \mu_3)}{2} \bar{\gamma}_3^j, \quad \tilde{L}_3^j = \alpha_3^j + \frac{a}{2} \bar{\beta}_3^j + \frac{b}{2} \bar{\gamma}_3^j, \quad (5.28)$$

$$\bar{\beta}_3^j = \beta_3^j(\xi_1 - \xi_3) + \gamma_3^j(\eta_1 - \eta_3), \quad \bar{\gamma}_3^j = \beta_3^j(\xi_2 - \xi_3) + \gamma_3^j(\eta_2 - \eta_3). \quad (5.29)$$

For example, the POWELL–SABIN B-splines $N_k^j(\tilde{\boldsymbol{\tau}})$ in the mini-triangle $n=3$ of Fig. 5.1(d) can be expressed as

$$\begin{bmatrix} N_1^1(\tilde{\boldsymbol{\tau}}) \\ N_1^2(\tilde{\boldsymbol{\tau}}) \\ N_1^3(\tilde{\boldsymbol{\tau}}) \\ N_2^1(\tilde{\boldsymbol{\tau}}) \\ N_2^2(\tilde{\boldsymbol{\tau}}) \\ N_2^3(\tilde{\boldsymbol{\tau}}) \\ N_3^1(\tilde{\boldsymbol{\tau}}) \\ N_3^2(\tilde{\boldsymbol{\tau}}) \\ N_3^3(\tilde{\boldsymbol{\tau}}) \end{bmatrix} = \begin{bmatrix} a\tilde{L}_1^1 & 0 & 0 & 0 & 0 & 0 \\ a\tilde{L}_1^2 & 0 & 0 & 0 & 0 & 0 \\ a\tilde{L}_1^3 & 0 & 0 & 0 & 0 & 0 \\ b\tilde{L}_2^1 & \tilde{L}_2^1 & \mu_2\tilde{L}_2^1 & \alpha_2^1 & L_2^1 & \mu_2L_2^1 \\ b\tilde{L}_2^2 & \tilde{L}_2^2 & \mu_2\tilde{L}_2^2 & \alpha_2^2 & L_2^2 & \mu_2L_2^2 \\ b\tilde{L}_2^3 & \tilde{L}_2^3 & \mu_2\tilde{L}_2^3 & \alpha_2^3 & L_2^3 & \mu_2L_2^3 \\ c\tilde{L}_3^1 & 0 & \mu_3\tilde{L}_3^1 & 0 & 0 & \mu_3L_3^1 \\ c\tilde{L}_3^2 & 0 & \mu_3\tilde{L}_3^2 & 0 & 0 & \mu_3L_3^2 \\ c\tilde{L}_3^3 & 0 & \mu_3\tilde{L}_3^3 & 0 & 0 & \mu_3L_3^3 \end{bmatrix} \begin{bmatrix} B_{200}^2(\tilde{\boldsymbol{\tau}}) \\ B_{110}^2(\tilde{\boldsymbol{\tau}}) \\ B_{101}^2(\tilde{\boldsymbol{\tau}}) \\ B_{020}^2(\tilde{\boldsymbol{\tau}}) \\ B_{011}^2(\tilde{\boldsymbol{\tau}}) \\ B_{002}^2(\tilde{\boldsymbol{\tau}}) \end{bmatrix}, \quad (5.30)$$

or in matrix-vector format

$$\underline{\mathbf{N}}_n^e(\tilde{\boldsymbol{\tau}}) = \underline{\mathbf{C}}_n^e \underline{\mathbf{B}}(\tilde{\boldsymbol{\tau}}), \quad (5.31)$$

where the BÉZIER extraction operator $\underline{\mathbf{C}}_n^e$ in Eq. (5.30) contains the BÉZIER ordinates $b_{r,s,t}$ for each POWELL–SABIN B-spline $N_k^j(\tilde{\boldsymbol{\tau}})$ in the mini-triangle $n=3$ of triangle e from Fig. 5.1(c),(d). Hence, it is possible to apply the BÉZIER extraction procedure to POWELL–SABIN B-splines, in a fashion similar to NURBS and T-splines (see BORDEN *et al.* [13], SCOTT *et al.* [94]).

Non-Uniform Rational POWELL–SABIN B-splines (NURPS) $R_k^j(\tilde{\boldsymbol{\tau}})$ can be computed as

follows

$$R_k^j(\tilde{\boldsymbol{x}}) = \frac{w_k^j N_k^j(\tilde{\boldsymbol{x}})}{\sum_{k=1}^{N_v} \sum_{j=1}^3 w_k^j N_k^j(\tilde{\boldsymbol{x}})}, \quad (5.32)$$

with w_k^j the weights associated to each corner \mathbf{Q}_k^j of a POWELL–SABIN triangle. A NURPS-surface $\underline{\mathbf{S}}(\tilde{\boldsymbol{x}})$ in the physical domain $\underline{\boldsymbol{x}}$ can be expressed by a map from the mini-triangle domain $\tilde{\boldsymbol{x}}$ with

$$\underline{\mathbf{S}}(\tilde{\boldsymbol{x}}) = \sum_{k=1}^{N_v} \sum_{j=1}^3 R_k^j(\tilde{\boldsymbol{x}}) \underline{\mathbf{P}}_k^j, \quad (5.33)$$

where the control points $\underline{\mathbf{P}}_k^j$ correspond to each \mathbf{Q}_k^j , see also Fig. 5.1(b). In matrix-vector format, the NURPS $\underline{\mathbf{R}}_n^e$ for one mini-triangle n of triangle e is obtained from

$$\underline{\mathbf{R}}_n^e(\tilde{\boldsymbol{x}}) = \underline{\mathbf{W}}^e \underline{\mathbf{C}}_n^e \frac{\underline{\mathbf{B}}(\tilde{\boldsymbol{x}})}{W_n^{be}(\tilde{\boldsymbol{x}})}, \quad (5.34)$$

with

$$W_n^{be}(\tilde{\boldsymbol{x}}) = (\underline{\mathbf{w}}_n^{be})^T \underline{\mathbf{B}}(\tilde{\boldsymbol{x}}), \quad \underline{\mathbf{w}}_n^{be} = (\underline{\mathbf{C}}_n^e)^T \underline{\mathbf{w}}^e, \quad \underline{\mathbf{W}}^e = \text{diag}(\underline{\mathbf{w}}^e), \quad (5.35)$$

where $\underline{\mathbf{w}}^e$ is the vector containing the weights of triangle e . The derivatives with respect to coordinates in the domain of the mini-triangle, $\tilde{\boldsymbol{x}}$, read

$$\frac{\partial \underline{\mathbf{R}}_n^e(\tilde{\boldsymbol{x}})}{\partial \tilde{\tau}_i} = \underline{\mathbf{W}}^e \underline{\mathbf{C}}_n^e \frac{\partial}{\partial \tilde{\tau}_i} \left(\frac{\underline{\mathbf{B}}(\tilde{\boldsymbol{x}})}{W_n^{be}(\tilde{\boldsymbol{x}})} \right) = \underline{\mathbf{W}}^e \underline{\mathbf{C}}_n^e \left(\frac{1}{W_n^{be}(\tilde{\boldsymbol{x}})} \frac{\partial \underline{\mathbf{B}}(\tilde{\boldsymbol{x}})}{\partial \tilde{\tau}_i} - \frac{\partial W_n^{be}(\tilde{\boldsymbol{x}})}{\partial \tilde{\tau}_i} \frac{\underline{\mathbf{B}}(\tilde{\boldsymbol{x}})}{(W_n^{be}(\tilde{\boldsymbol{x}}))^2} \right). \quad (5.36)$$

The derivatives in the physical domain $\underline{\boldsymbol{x}}$ are subsequently obtained as

$$\frac{\partial \underline{\mathbf{R}}_n^e(\tilde{\boldsymbol{x}})}{\partial x_i^e} = \sum_{j=1}^2 \frac{\partial \underline{\mathbf{R}}_n^e(\tilde{\boldsymbol{x}})}{\partial \tilde{\tau}_j} \frac{\partial \tilde{\tau}_j}{\partial x_i^e}, \quad (5.37)$$

where

$$\frac{\partial \tilde{\tau}_j}{\partial x_i^e} = \begin{bmatrix} \frac{\partial x^e}{\partial \tilde{\tau}_1} & \frac{\partial x^e}{\partial \tilde{\tau}_2} \\ \frac{\partial y^e}{\partial \tilde{\tau}_1} & \frac{\partial y^e}{\partial \tilde{\tau}_2} \end{bmatrix}^{-1} = \underline{\mathbf{J}}^{-1}, \quad (5.38)$$

with the JACOBIAN matrix $\underline{\mathbf{J}}$ of the geometry mapping. It was suggested by SPELEERS *et al.* [104] to evaluate the integrals for each component of the stiffness matrix in Eq. (5.13) analytically since the integral of a POWELL–SABIN B-spline $N_k^j(\tilde{\boldsymbol{x}})$ over a

mini-triangle n with area A_n can be computed as (cf. CHUI & LAI [23])

$$\int_{\Omega_n} N_k^j(\tilde{\boldsymbol{\tau}}) d\tilde{\tau}_1 d\tilde{\tau}_2 = \frac{A_n}{6} \sum_{r+s+t=2} b_{r,s,t}. \quad (5.39)$$

Relations for the computation of the product of the derivatives of a POWELL–SABIN B-spline can then be derived accordingly, see SPELEERS *et al.* [104]. However, this is computationally more expensive than using the BÉZIER extraction procedure.

5.2.3 Patch test

Special care must be taken for the selection of the POWELL–SABIN triangles in order to pass the patch test. For instance, consider the triangulation \mathcal{T} in Figs. 5.3(a) and 5.3(b) consisting of two elements.

Figs. 5.3(a) and 5.3(b) show two different options for the bottom left POWELL–SABIN triangle. The bottom left POWELL–SABIN triangle in Fig. 5.3(b) is valid since it contains all POWELL–SABIN points that correspond to the bottom left vertex. Both options yield an affine transformation when applying the transformation from Eq. (4.52) to the control points $\underline{\mathbf{P}}^i$ as depicted in Figs. 5.3(c) and 5.3(d) since the basis functions fulfil the partition of unity.

The mesh in Fig. 5.3(a) also passes the patch test depicted in Fig. 5.4(a). Fig. 5.4(b) shows the contour of the basis function for the red control point corresponding to the mesh in Fig. 5.3(a). The basis function in Fig. 5.4(b) is zero along the left edge. Shifting the corner of the bottom left POWELL–SABIN triangle as in Fig. 5.3(b) results in a different basis function for the red control point in Fig. 5.4(c): the basis function is now non-zero along the left edge. Furthermore, the mesh in Fig. 5.3(b) does not pass the patch test – applying the boundary conditions in Fig. 5.4(a) does not result in a constant stress in the x -direction.

5.3 The representation of a disc using NURBS, T-splines, NURBS-to-NURPS and POWELL–SABIN B-splines

In Section 5.4, an analysis will be carried out for a KIRCHHOFF–LOVE plate with a circular geometry. Therefore, a number of representations for this geometry using NURBS, unstructured T-splines, NURBS-to-NURPS and POWELL–SABIN B-splines are created. The NURBS and NURBS-to-NURPS meshes capture the circular geometry of the disc exactly while the unstructured T-spline and POWELL–SABIN B-spline mesh only approximate the circle.

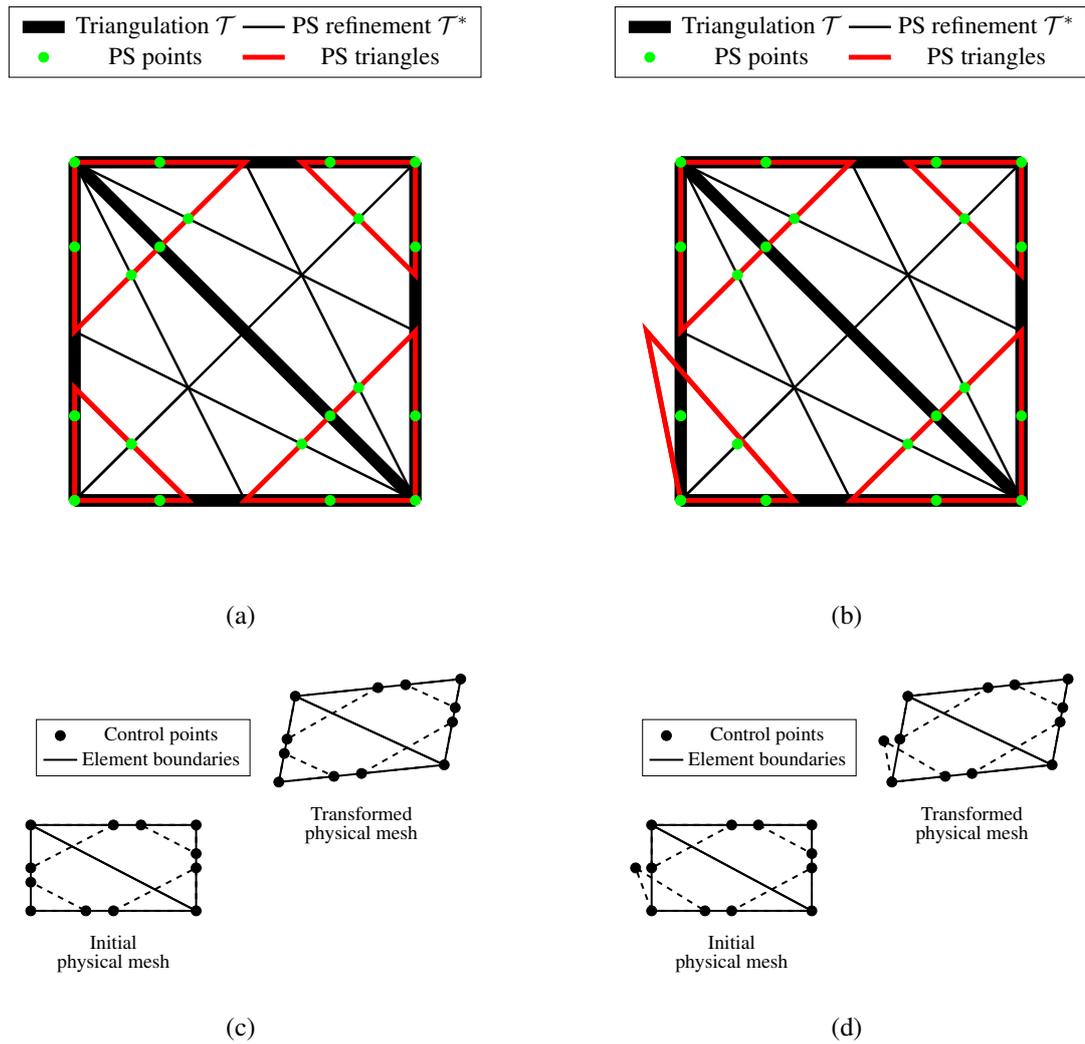


Fig. 5.3: A triangulation \mathcal{T} consisting of two elements with a different choice for the bottom left POWELL–SABIN triangle in (a) and (b). Both options yield an affine transformation in (c) and (d) since the POWELL–SABIN B-splines form a partition of unity.

5.3.1 Representation of a disc using NURBS

5.3.1.1 Single patch

An area with a circular boundary can be created with a single quadratic NURBS-patch (or one element) using nine control points, see Fig. 5.5(a). The isoparametric lines indicate where singularities (i. e. the determinant of the JACOBIAN matrix $\underline{\underline{\mathbf{J}}}$ in Eq. (5.38) vanishes) are introduced: at control points one, three, seven and nine, see also SCHMIDT *et al.* [93]. Uniform h -refinement will be applied for the convergence study in Section 5.4.

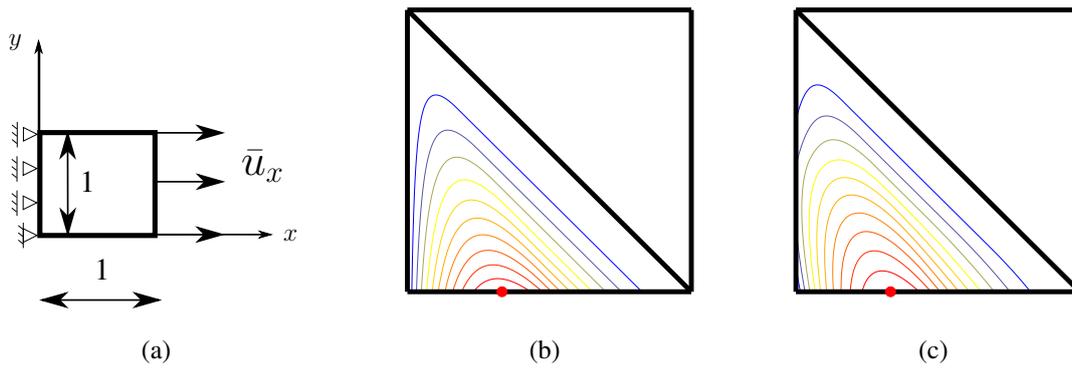


Fig. 5.4: (a) Patch test for the meshes in Figs. 5.3(a) and 5.3(b). (b) The mesh in Fig. 5.3(a) passes the patch test – the basis function of the red control point is zero along the left edge. (c) The mesh in Fig. 5.3(b) with the modified bottom left POWELL–SABIN triangle does not pass the patch test – the red control point has a non-zero basis function along the left edge due to the modified POWELL–SABIN triangle.

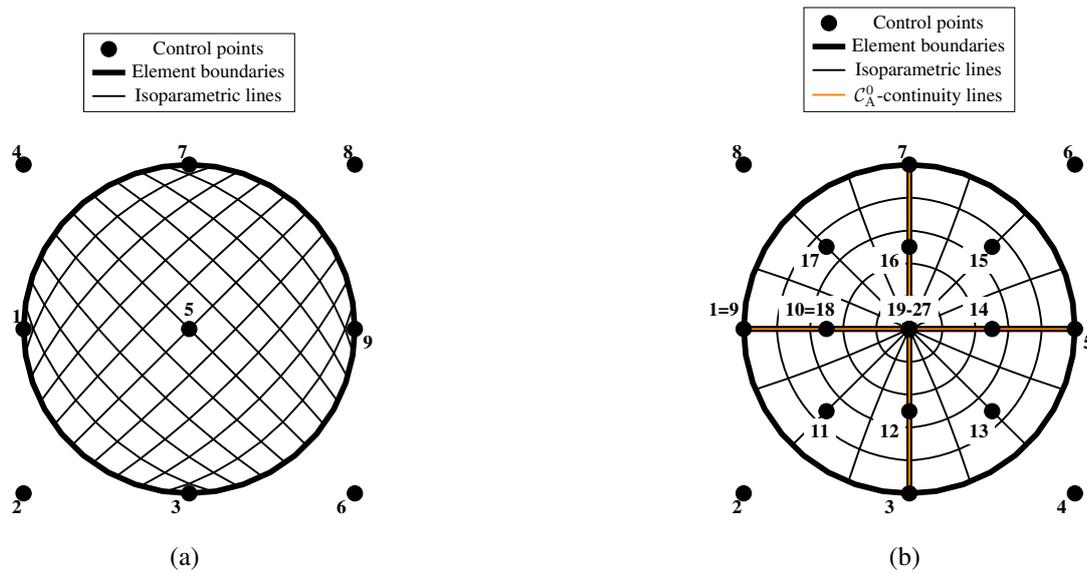


Fig. 5.5: Representation of a disc using NURBS with (a) one single quadratic element / patch and (b) a polar parameterisation using four patches.

5.3.1.2 Polar parameterisation

Another possibility to construct a circle is a polar parameterisation that uses four NURBS-patches and 27 control points, Fig. 5.5(b). This results in a singular / degenerated point in the centre where the determinant of the JACOBIAN matrix $\underline{\underline{J}}$ in Eq. (5.38) vanishes. Also, four C_A^0 -continuity lines (orange) are introduced. As for the single NURBS-patch from the previous section, uniform h -refinement will be applied for the convergence study in Section 5.4. h -refinement does not change the number of

\mathcal{C}_A^0 -continuity lines.

Since Fig. 5.5(b) represents a $(\mathcal{G}^1, \mathcal{C}_A^0)$ -construction, the bending strip method proposed by KIENDL *et al.* [58] can be employed along the \mathcal{C}_A^0 -continuity lines. The bending strip method adds a stiffness matrix of the form

$$\underline{\underline{\mathbf{K}}}_{BS} = \int_{\Omega} \underline{\underline{\mathbf{B}}}^T \underline{\underline{\mathbf{Q}}}^T \underline{\underline{\mathbf{D}}}_{BS} \underline{\underline{\mathbf{Q}}} \underline{\underline{\mathbf{B}}} \mathbf{w} \, dx dy \quad (5.40)$$

to Eq. (5.13) at the interfaces between patches. In Eq. (5.40),

$$\underline{\underline{\mathbf{D}}}_{BS} = \frac{h^3}{12} \begin{bmatrix} E_{BS} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (5.41)$$

represents the penalty matrix with the penalty bending stiffness E_{BS} and

$$\underline{\underline{\mathbf{Q}}} = \begin{bmatrix} n_1 & n_2 & 0 \\ -n_2 & n_1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (5.42)$$

is the orthogonal rotation matrix with the normal vector $\underline{\mathbf{n}}$ of the bending strip. The rotation matrix $\underline{\underline{\mathbf{Q}}}$ aligns the penalty matrix $\underline{\underline{\mathbf{D}}}_{BS}$ with the bending direction.

5.3.2 Representation of a disc using unstructured T-splines

The approach from Section 4.7 generates \mathcal{C}_A^1 -continuous blending functions that fulfil the partition of unity for an unstructured quadratic T-spline mesh with extraordinary points. The T-spline mesh for the circular disc is shown for the index domain in Fig. 5.6(a).

A circular geometry cannot be represented exactly without double knots, see PIEGL & TILLER [85]. Hence, the unstructured quadratic T-spline mesh can only approximate the circular geometry since it is \mathcal{C}_A^1 -continuous in the entire domain. The n_B control points on the boundary $\underline{\underline{\mathbf{P}}}_B^i$ are determined by solving

$$\sum_{i=1}^{n_B} N^i(\bar{\xi}_k) \underline{\underline{\mathbf{P}}}_B^i = \underline{\underline{\mathbf{S}}}_{\text{Circle}}(\bar{\xi}_k) \quad \text{for } k=1 \dots n_B. \quad (5.43)$$

$\bar{\xi}$ is the coordinate along the circle with $0 \leq \bar{\xi} \leq 360$, and $\underline{\underline{\mathbf{S}}}_{\text{Circle}}(\bar{\xi}_k)$ the curve representing the circle. Along the boundary are n_B elements. The centre of each boundary element is located at

$$\bar{\xi}_k = \frac{360}{n_B} k + \frac{180}{n_B} \quad \text{for } k=1 \dots n_B. \quad (5.44)$$

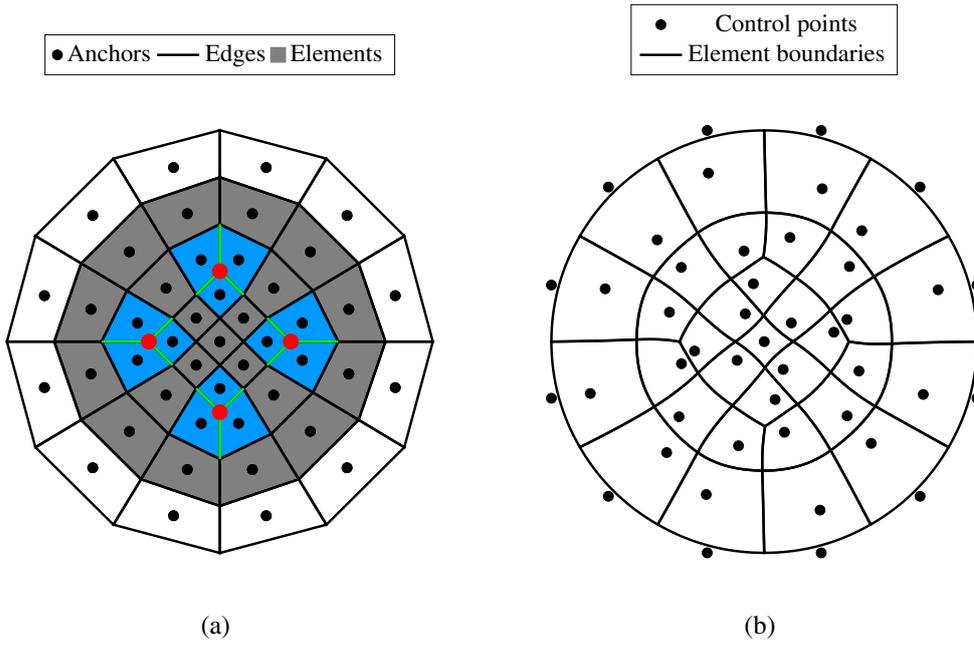


Fig. 5.6: An unstructured quadratic T-spline mesh in (a) the index domain and (b) in the physical domain. (a) Extraordinary points are marked red, spoke edges green and one-ring neighbourhood elements of an extraordinary point blue. (b) The T-spline mesh in the physical domain only approximates the circular boundary.

Plugging $\bar{\xi}_k$ from Eq. (5.44) into Eq. (5.43) results in the following system

$$\begin{bmatrix} \frac{6}{8} & \frac{1}{8} & 0 & 0 & 0 & \dots & 0 & \frac{1}{8} \\ \frac{1}{8} & \frac{6}{8} & \frac{1}{8} & 0 & 0 & \dots & 0 & 0 \\ 0 & \frac{1}{8} & \frac{6}{8} & \frac{1}{8} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{1}{8} & 0 & 0 & 0 & 0 & \dots & \frac{1}{8} & \frac{6}{8} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{P}}_B^{1T} \\ \underline{\mathbf{P}}_B^{2T} \\ \underline{\mathbf{P}}_B^{3T} \\ \vdots \\ \underline{\mathbf{P}}_B^{n_B T} \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{S}}_{\text{Circle}}^T(\bar{\xi}_1) \\ \underline{\mathbf{S}}_{\text{Circle}}^T(\bar{\xi}_2) \\ \underline{\mathbf{S}}_{\text{Circle}}^T(\bar{\xi}_3) \\ \vdots \\ \underline{\mathbf{S}}_{\text{Circle}}^T(\bar{\xi}_{n_B}) \end{bmatrix}, \quad (5.45)$$

which can be solved for the control points on the boundary $\underline{\mathbf{P}}_B^i$. The values in the matrix of Eq. (5.45) result from

$$\begin{bmatrix} N^{k-1}(\xi_k) \\ N^k(\xi_k) \\ N^{k+1}(\xi_k) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 1 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} B^1(\xi_k) \\ B^2(\xi_k) \\ B^3(\xi_k) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 1 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{4} \\ \frac{2}{4} \\ \frac{1}{4} \end{bmatrix} = \begin{bmatrix} \frac{1}{8} \\ \frac{6}{8} \\ \frac{1}{8} \end{bmatrix}. \quad (5.46)$$

The location of the control points in the interior of the T-spline mesh in Fig. 5.6 is determined by solving the problem

$$\frac{\partial}{\partial \xi_\beta} \left(\frac{\partial x_\alpha}{\partial \xi_\beta} + \frac{\partial x_\beta}{\partial \xi_\alpha} \right) = 0 \quad (5.47)$$

in the sub-parameter domain ξ_α , while the location of the control points on the boundary \underline{P}_B^i of the disc is prescribed. The resulting T-spline mesh in the physical domain is depicted in Fig. 5.6(b). Two T-spline meshes with a different refinement level are not nested – for each refinement level, Eqs. (5.45) and (5.47) are solved for the determination of the control points. As already discussed in Section 4.7, the BÉZIER extraction operators \underline{C}_e for the elements in the one-ring neighbourhood of an extraordinary point are not square matrices. Hence, these elements cannot be refined hierarchically as in Section 4.6 and the BÉZIER projection procedure by THOMAS *et al.* [110] cannot be applied since the inverse of BÉZIER extraction operator – the reconstruction operator – is required.

5.3.3 Representation of a disc using the NURBS-to-NURPS methodology

Next, the single NURBS-patch from Section 5.3.1.1 is transformed into a NURPS mesh (NURBS-to-NURPS) following SPELEERS *et al.* [106]. The boundary of the NURBS-to-NURPS matches exactly the boundary defined by the single NURBS-patch, see Fig. 5.7. In order to represent the circular boundary exactly, the POWELL–SABIN triangles that correspond to the corners $\underline{\xi} = (0, 0); (0, 1); (1, 0); (1, 1)$ in the parameter domain need to be degenerated into a line in the physical domain \underline{x} (dashed lines in Fig. 5.7(c) and Fig. 5.7(d)). The NURBS-to-NURPS approach is based on a single NURBS-patch, and a method for transforming multiple NURBS-patches into a NURPS has so far not been proposed.

5.3.4 Representation of a disc using POWELL–SABIN B-splines

A linear \mathcal{C}_A^0 -continuous finite element triangulation \mathcal{T} can be transformed into a \mathcal{C}_A^1 -continuous POWELL–SABIN B-spline mesh \mathcal{T}^* . This corresponds to a NURPS mesh with the location of the control points $\underline{P}_i^j = \underline{Q}_i^j$ and for all weights $w_i^j = 1$ in Eq. (5.33), i. e. the parametric and the physical domains, $\underline{\xi}$ and \underline{x} , respectively, are identical. For this case, the circular boundary of the disc is only approximated. The POWELL–SABIN triangles on the boundary are constrained in such a way that one corner of the POWELL–SABIN triangle is always equivalent to the vertex coordinate, see Fig. 5.8. Upon mesh refinement, the POWELL–SABIN triangles on the boundary progressively deteriorate into lines, see Fig. 5.8(b).

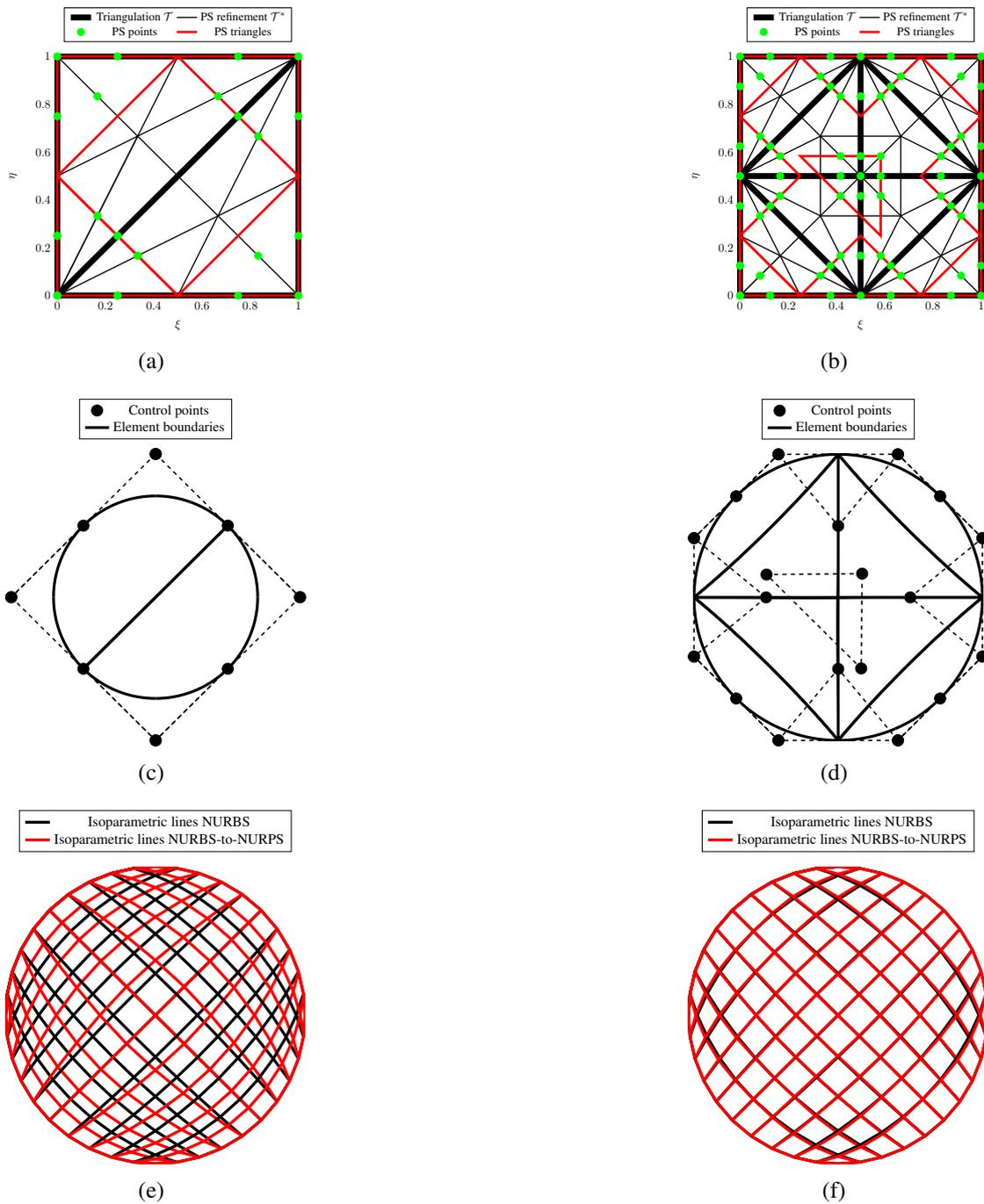


Fig. 5.7: Generation of a NURBS-to-NURPS mesh from a single NURBS-patch. From the left to the right column, the triangulation inside the NURBS-patch is refined. (a) and (b) show the triangulation \mathcal{T} and POWELL–SABIN refinement \mathcal{T}^* in the parameter domain ξ . (c) and (d) show the NURBS-to-NURPS mesh in the physical domain \underline{x} ; dashed lines connect the control points of the corners of the POWELL–SABIN triangles. (e) and (f) show isoparametric lines for the NURBS and the NURBS-to-NURPS meshes in the physical domain \underline{x} . Upon refinement, the NURBS-to-NURPS representation converges to the NURBS parameterisation.

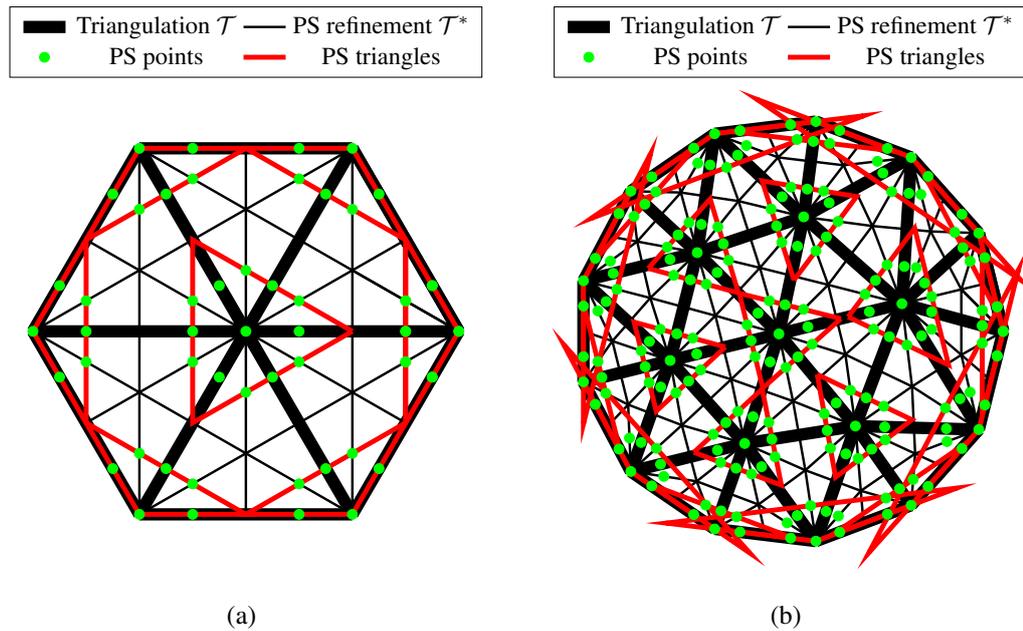


Fig. 5.8: Approximation of a circle using POWELL–SABIN B-splines for two different levels of refinement. The POWELL–SABIN triangles on the boundary are constrained in such a way that one corner of each POWELL–SABIN triangle is equivalent to the vertex.

5.4 Numerical examples

In this section, the circular KIRCHHOFF–LOVE plate in Fig. 5.9 is considered for two different boundary conditions: simply supported and clamped.

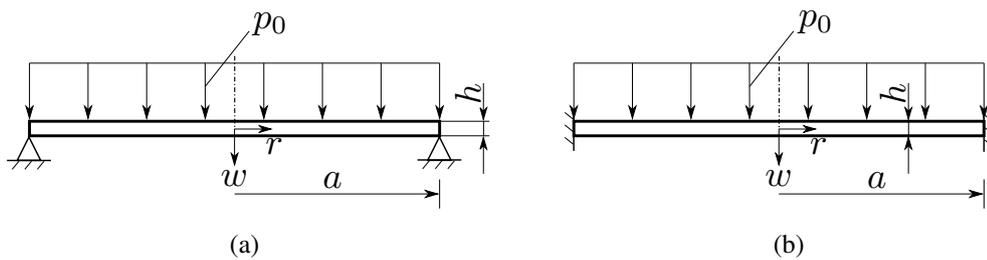


Fig. 5.9: Computational set-up for (a) the simply supported and (b) the clamped circular disc.

The parameters are: YOUNG’s modulus $E = 2.1 \cdot 10^5$ MPa, POISSON’s ratio $\nu = 0.3$, force per unit area $p_0 = 0.16$ MPa, radius $a = 250$ mm and thickness $h = 10$ mm. The analytical solutions w_{ex} for both cases can be found in TIMOSHENKO & WOINOWSKY-KRIEGER [112, Chapter 3]. In the following, the results for the single NURBS-patch from Section 5.3.1.1 will be plotted in all convergence plots for compari-

son. The L_2 error norm is computed from

$$w_{L_2} = \frac{\sqrt{\int_{\Omega} (w - w_{\text{ex}})^2 \, dx dy}}{\sqrt{\int_{\Omega} w_{\text{ex}}^2 \, dx dy}}. \quad (5.48)$$

The convergence rate for the fourth order partial differential equation in Eq. (5.1) is equal to two for quadratic basis functions ($p=2$) according to TAGLIABUE *et al.* [109],

$$w_{L_2} \leq Ch^{\min(p+1, 2p-2)} = Ch^2, \quad (5.49)$$

with the mesh size h and a constant C . In order to transform Eq. (5.49) onto the degrees of freedom (DOF), the relation

$$h = \frac{p+1}{\sqrt{\text{DOF}}} \quad (5.50)$$

is used so that

$$w_{L_2} \leq \tilde{C} \text{DOF}^{-1} \quad (5.51)$$

is obtained. For the NURBS and T-spline meshes, the deflection w for the outer control points is set to zero for the simply supported case, while the deflection w of the two outer rows of control points is set to zero for the clamped case. For the POWELL–SABIN B-splines and NURBS-to-NURPS, the deflection w of a control point associated to a vertex on the boundary is set to zero for the simply supported case if $\alpha_k^j > 0$ holds in Eq. (5.20a). For the clamped case, the deflection w of all three control points which are associated to a vertex that lies on the boundary is set to zero.

Fig. 5.10 shows the error in the L_2 error norm when the circle is represented with four NURBS-patches and a polar parameterisation with a singular point in the centre as in Section 5.3.1.2.

For this (\mathcal{G}^1, C_A^0) -construction, the bending strip method has been applied along the C_A^0 -continuity lines for different values of the penalty stiffness E_{BS} . The results do not converge well for the finer meshes. Moreover, the value of the penalty stiffness E_{BS} for which the lowest error w_{L_2} is obtained, is different for both boundary conditions.

Fig. 5.11 gives the results upon mesh refinement for the unstructured T-spline mesh from Section 5.3.2, again together with the results that stem from the single NURBS-patch for comparison.

In contrast to the previous (\mathcal{G}^1, C_A^0) -construction using the bending strip method, a constant convergence rate is observed for the simply supported and clamped case for the unstructured T-spline mesh. The error w_{L_2} for the unstructured T-spline mesh is larger than the error w_{L_2} for the single NURBS-patch. A convergence study was also carried out for an unstructured T-spline mesh without the \mathcal{G}^1 -construction that yields C_A^1 -continuous

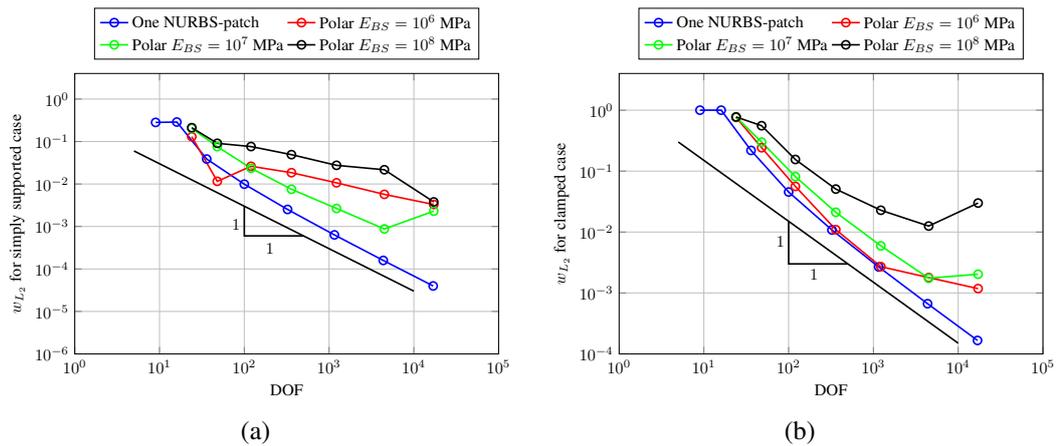


Fig. 5.10: Convergence plots for the single NURBS-patch and for the polar NURBS parameterisation using four NURBS-patches for (a) the simply supported and (b) the clamped boundary conditions.

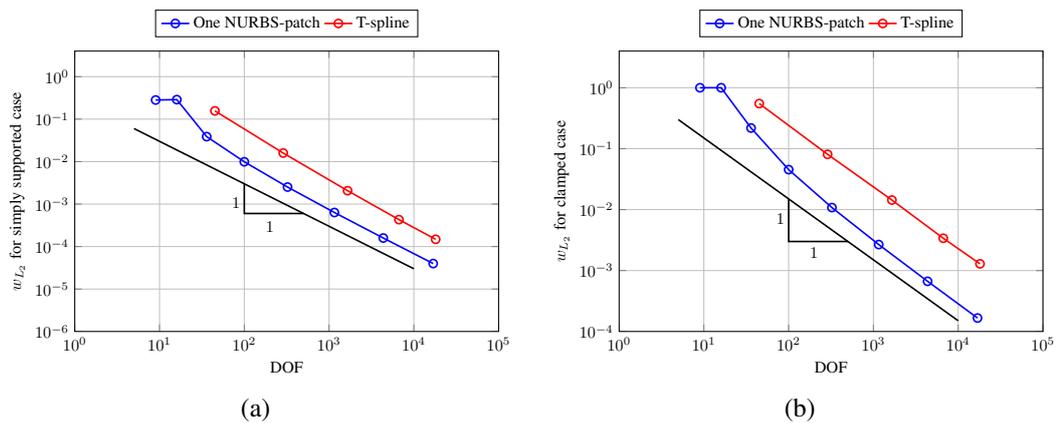


Fig. 5.11: Convergence plots for the single NURBS-patch and the unstructured T-spline mesh for (a) the simply supported and (b) the clamped boundary conditions.

blending functions for the one-ring neighbourhood elements, i. e. a T-spline mesh that is only \mathcal{C}_A^0 -continuous along spoke edges and non-standard. This T-spline mesh gives almost identical results for the error w_{L_2} and the results are not plotted in Fig. 5.11 since they are not distinguishable from the error for standard T-spline meshes with \mathcal{C}_A^1 -continuity along spoke edges.

Finally, the results for the NURBS-to-NURPS configuration and for the POWELL–SABIN B-spline computations are given in Fig. 5.12.

Unexpectedly, the POWELL–SABIN B-spline mesh, which only approximates the geometry, gives the lowest error w_{L_2} in case of the simply supported boundary conditions – even lower than the single NURBS-patch. This is not the case for the clamped boundary conditions, but also then the POWELL–SABIN B-spline mesh gives results that

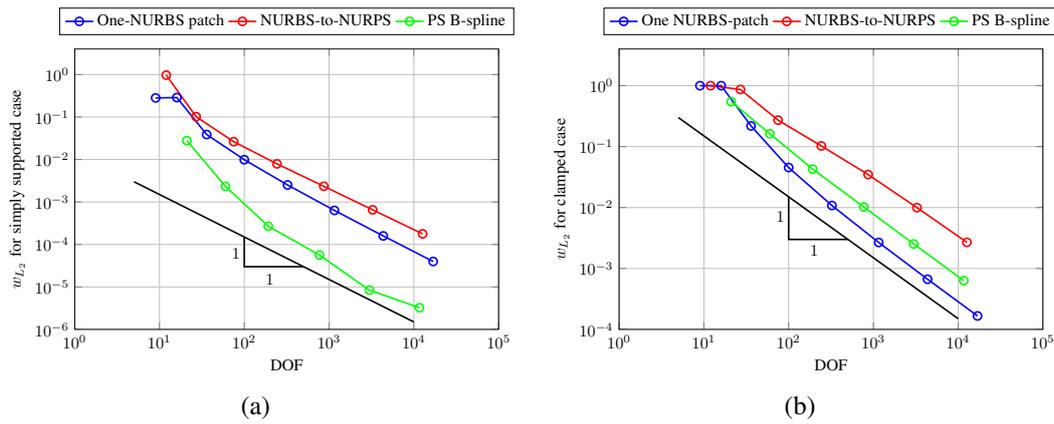


Fig. 5.12: Convergence plots for the single NURBS-patch, the NURBS-to-NURPS approach and the POWELL–SABIN B-splines for (a) the simply supported and (b) the clamped boundary conditions.

are superior to those from the NURBS-to-NURPS approach. A possible explanation is that the effect of the distorted elements (degenerated POWELL–SABIN triangles), which are introduced by the NURBS-to-NURPS approach, is not compensated by the improved (exact) representation of the boundary.

6 POWELL–SABIN B-splines for smeared and discrete fracture approaches

In the following, the implicit fourth order gradient damage model for quasi-brittle materials exploiting the higher continuity of POWELL–SABIN B-splines is considered. Afterwards, POWELL–SABIN B-splines are applied to discrete fracture while cohesive elements are inserted along the crack.

Throughout, index notation is adopted with respect to a CARTESIAN frame.

6.1 Implicit higher order gradient damage model

NURBS and T-splines have been utilised for the implicit higher order gradient damage model by VERHOOSSEL *et al.* [115]. Herein, it will be demonstrated how POWELL–SABIN B-splines can be applied to these kind of problems. They are \mathcal{C}_A^1 -continuous for any triangulation and therefore do not require any user intervention when generating the meshes. Next, the continuum and numerical formulation for the implicit higher order gradient damage model is introduced. Thereafter, two numerical examples are considered, including the advantages of POWELL–SABIN B-splines compared to NURBS and T-splines regarding the discretisation.

6.1.1 Continuum formulation

The implicit higher order gradient damage model requires the solution of two coupled field problems. The first field problem to be solved is the stress equilibrium

$$\sigma_{ij,i} = 0 \quad (6.1)$$

subject to the boundary conditions

$$\sigma_{ij}n_j = h_i \quad \text{on } \partial\Omega_h, \quad u_i = \bar{u}_i \quad \text{on } \partial\Omega_u, \quad (6.2)$$

with the decomposition of the boundary $\partial\Omega$ into the parts $\partial\Omega_h$ and $\partial\Omega_u$ ($\partial\Omega_h \cap \partial\Omega_u = \emptyset$, $\partial\Omega_h \cup \partial\Omega_u = \partial\Omega$), the prescribed surface traction \mathbf{h} , prescribed displacement $\bar{\mathbf{u}}$ and nor-

mal vector \mathbf{n} on the boundary. In Eq. (6.1), $\boldsymbol{\sigma}$ is the stress tensor

$$\sigma_{ij} = (1 - \omega) C_{ijkl} \varepsilon_{kl}, \quad (6.3)$$

with the elasticity tensor \mathbf{C} of the undamaged material, the infinitesimal strain tensor

$$\varepsilon_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i}), \quad (6.4)$$

where \mathbf{u} is the displacement, and $\omega \in [0, 1]$ the scalar damage parameter ($\omega=0$ undamaged, $\omega=1$ fully broken). The damage ω is related to the monotonically increasing history parameter κ , $\omega = \omega(\kappa)$. In order to ensure that κ can only grow, the KARUSH–KUHN–TUCKER [56, 64] conditions

$$f \leq 0, \quad \dot{\kappa} \geq 0, \quad \dot{\kappa} f = 0 \quad (6.5)$$

need to be satisfied. Monotonicity of κ involves monotonicity of ω . The loading function

$$f = \bar{\eta} - \kappa \quad (6.6)$$

will be used where $\bar{\eta}$ is the non-local equivalent strain. The non-local equivalent strain $\bar{\eta}$ was introduced in PEERLINGS *et al.* [82] since strain softening in conventional continuum damage models does not result in a unique solution upon mesh refinement. In the following, a brief outline of introducing non-locality into the model is given.

The non-local equivalent strain $\bar{\eta}$ is defined as the volume average of the local equivalent strain η

$$\bar{\eta}(\mathbf{x}) = \frac{\int_{\Omega} g(\mathbf{x}, \tilde{\mathbf{x}}) \eta(\tilde{\mathbf{x}}) d\tilde{\mathbf{x}}}{\int_{\Omega} g(\mathbf{x}, \tilde{\mathbf{x}}) d\tilde{\mathbf{x}}}, \quad (6.7)$$

with the weighting function

$$g(\mathbf{x}, \tilde{\mathbf{x}}) = \exp\left(-\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|_2^2}{2l_c^2}\right), \quad (6.8)$$

where l_c is the length scale parameter. Expressing the local equivalent strain $\eta(\tilde{\mathbf{x}})$ by a TAYLOR series around the point $\tilde{\mathbf{x}} = \mathbf{x}$ gives

$$\begin{aligned} \eta(\tilde{\mathbf{x}}) = & \eta(\mathbf{x}) + \eta_{,\tilde{i}}|_{\mathbf{x}}(\tilde{x}_i - x_i) + \frac{1}{2}\eta_{,\tilde{i}\tilde{j}}|_{\mathbf{x}}(\tilde{x}_i - x_i)(\tilde{x}_j - x_j) + \\ & \frac{1}{6}\eta_{,\tilde{i}\tilde{j}\tilde{k}}|_{\mathbf{x}}(\tilde{x}_i - x_i)(\tilde{x}_j - x_j)(\tilde{x}_k - x_k) + \\ & \frac{1}{24}\eta_{,\tilde{i}\tilde{j}\tilde{k}\tilde{l}}|_{\mathbf{x}}(\tilde{x}_i - x_i)(\tilde{x}_j - x_j)(\tilde{x}_k - x_k)(\tilde{x}_l - x_l) + \dots, \end{aligned} \quad (6.9)$$

which results for the non-local equivalent strain $\bar{\eta}$ with Eq. (6.7) in

$$\bar{\eta}(\mathbf{x}) = \eta(\mathbf{x}) + \frac{l_c^2}{2}\eta_{,\tilde{i}\tilde{i}}(\mathbf{x}) + \frac{l_c^4}{8}\eta_{,\tilde{i}\tilde{i}\tilde{j}\tilde{j}}(\mathbf{x}) + \dots \quad (6.10)$$

Eq. (6.10) contains the derivatives with respect to the coordinate system $\tilde{\mathbf{x}}$, and \mathbf{x} is the location of a certain point. Since the point \mathbf{x} takes all values in $\tilde{\mathbf{x}}$, i. e. \mathbf{x} and $\tilde{\mathbf{x}}$ are equivalent, Eq. (6.10) can be written as

$$\bar{\eta}(\mathbf{x}) = \eta(\mathbf{x}) + \frac{l_c^2}{2}\eta_{,ii}(\mathbf{x}) + \frac{l_c^4}{8}\eta_{,iiij}(\mathbf{x}) + \dots \quad (6.11)$$

Eq. (6.11) represents the explicit gradient damage formulation. The implicit formulation can be obtained as follows. Differentiating Eq. (6.11) twice and multiplying by $\frac{l_c^2}{2}$ results in

$$\frac{l_c^2}{2}\bar{\eta}_{,ii} = \frac{l_c^2}{2}\eta_{,ii} + \frac{l_c^4}{4}\eta_{,iiij} + \frac{l_c^6}{16}\eta_{,iiijkk} + \dots, \quad (6.12)$$

while differentiating Eq. (6.11) four times and multiplying by $\frac{l_c^2}{8}$ gives

$$\frac{l_c^4}{8}\bar{\eta}_{,iiij} = \frac{l_c^4}{8}\eta_{,iiij} + \frac{l_c^6}{16}\eta_{,iiijkk} + \frac{l_c^8}{64}\eta_{,iiijkkll} + \dots \quad (6.13)$$

Subtracting and adding Eqs. (6.12) and (6.13) from Eq. (6.11) yields

$$\bar{\eta} - \frac{l_c^2}{2}\bar{\eta}_{,ii} + \frac{l_c^4}{8}\bar{\eta}_{,iiij} = \eta + \frac{l_c^8}{64}\eta_{,iiijkkll} + \dots \quad (6.14)$$

Neglecting higher order terms in Eq. (6.14) results in a fourth order partial differential equation for the non-local equivalent strain $\bar{\eta}$

$$\bar{\eta} - \frac{l_c^2}{2}\bar{\eta}_{,ii} + \frac{l_c^4}{8}\bar{\eta}_{,iiij} = \eta, \quad (6.15)$$

which has to be solved together with Eq. (6.1). The boundary conditions

$$\bar{\eta}_{,i}n_i = 0, \quad \bar{\eta}_{,ij}n_i = 0, \quad \bar{\eta}_{,iij}n_i = 0 \quad (6.16)$$

are imposed since it was observed by VERHOESEL *et al.* [115] that ignoring the boundary terms has a minor effect on the results. Still, it is not clear how to interpret the boundary conditions physically. Multiplying Eq. (6.1) by a test function δu_j , Eq. (6.15) by a test function $\delta \bar{\eta}$ and integrating over the domain Ω results after integration by parts, GAUSS' theorem and the boundary conditions in Eqs. (6.2) and (6.16) in

$$\int_{\Omega} \delta u_{i,j} \sigma_{ij} dV = \int_{\partial\Omega_h} \delta u_i h_i dA, \quad (6.17)$$

$$\int_{\Omega} \delta \bar{\eta} \bar{\eta} + \frac{l_c^2}{2} \delta \bar{\eta}_{,i} \bar{\eta}_{,i} + \frac{l_c^4}{8} \delta \bar{\eta}_{,ij} \bar{\eta}_{,ij} dV = \int_{\Omega} \delta \bar{\eta} \eta dV.$$

6.1.2 Finite element formulation

Discretisation of the domain Ω into E elements,

$$\Omega = \bigcup_{e=1}^E \Omega^e, \quad (6.18)$$

and approximation of the displacement u_i , the non-local strain $\bar{\eta}$ and their derivatives,

$$\begin{aligned} \underline{\mathbf{u}}^e &= \underline{\mathbf{N}}_u \mathbf{u}, & \delta \underline{\mathbf{u}}^e &= \underline{\mathbf{N}}_u \delta \mathbf{u}, & \underline{\boldsymbol{\varepsilon}}^e &= \underline{\mathbf{B}}_u \mathbf{u}, & \delta \underline{\boldsymbol{\varepsilon}}^e &= \underline{\mathbf{B}}_u \delta \mathbf{u}, \\ \bar{\boldsymbol{\eta}}^e &= \underline{\mathbf{N}}_{\bar{\eta}}^T \bar{\boldsymbol{\eta}}, & \delta \bar{\boldsymbol{\eta}}^e &= \underline{\mathbf{N}}_{\bar{\eta}}^T \delta \bar{\boldsymbol{\eta}}, & \bar{\boldsymbol{\eta}}_{,i}^e &= \underline{\mathbf{B}}_{\bar{\eta}} \bar{\boldsymbol{\eta}}, & \delta \bar{\boldsymbol{\eta}}_{,i}^e &= \underline{\mathbf{B}}_{\bar{\eta}} \delta \bar{\boldsymbol{\eta}}, \\ \bar{\boldsymbol{\eta}}_{,ij}^e &= \underline{\mathbf{B}}_{\bar{\eta}\bar{\eta}} \bar{\boldsymbol{\eta}}, & \delta \bar{\boldsymbol{\eta}}_{,ij}^e &= \underline{\mathbf{B}}_{\bar{\eta}\bar{\eta}} \delta \bar{\boldsymbol{\eta}}, \end{aligned} \quad (6.19)$$

with

$$\begin{aligned} \underline{\mathbf{u}}^e &= \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix}, & \delta \underline{\mathbf{u}}^e &= \begin{bmatrix} \delta u_1^e \\ \delta u_2^e \end{bmatrix}, & \underline{\boldsymbol{\varepsilon}}^e &= \begin{bmatrix} \varepsilon_{11}^e \\ \varepsilon_{22}^e \\ 2\varepsilon_{12}^e \end{bmatrix}, & \delta \underline{\boldsymbol{\varepsilon}}^e &= \begin{bmatrix} \delta \varepsilon_{11}^e \\ \delta \varepsilon_{22}^e \\ 2\delta \varepsilon_{12}^e \end{bmatrix}, \\ \bar{\boldsymbol{\eta}}_{,i}^e &= \begin{bmatrix} \bar{\eta}_{,1}^e \\ \bar{\eta}_{,2}^e \end{bmatrix}, & \delta \bar{\boldsymbol{\eta}}_{,i}^e &= \begin{bmatrix} \delta \bar{\eta}_{,1}^e \\ \delta \bar{\eta}_{,2}^e \end{bmatrix}, & \bar{\boldsymbol{\eta}}_{,ij}^e &= \begin{bmatrix} \bar{\eta}_{,11}^e \\ \bar{\eta}_{,22}^e \\ \sqrt{2}\bar{\eta}_{,12}^e \end{bmatrix}, & \delta \bar{\boldsymbol{\eta}}_{,ij}^e &= \begin{bmatrix} \delta \bar{\eta}_{,11}^e \\ \delta \bar{\eta}_{,22}^e \\ \sqrt{2}\delta \bar{\eta}_{,12}^e \end{bmatrix} \end{aligned} \quad (6.20)$$

and

$$\begin{aligned} \underline{\underline{\mathbf{N}}}_u &= \begin{bmatrix} N_1 & 0 & N_2 & 0 & \dots \\ 0 & N_1 & 0 & N_2 & \dots \end{bmatrix}, \quad \underline{\underline{\mathbf{B}}}_u = \begin{bmatrix} N_{1,1} & 0 & N_{2,1} & 0 & \dots \\ 0 & N_{1,2} & 0 & N_{2,2} & \dots \\ N_{1,2} & N_{1,1} & N_{2,2} & N_{2,1} & \dots \end{bmatrix}, \\ \underline{\underline{\mathbf{N}}}_{\bar{\eta}}^T &= \begin{bmatrix} N_1 & N_2 & \dots \end{bmatrix}, \quad \underline{\underline{\mathbf{B}}}_{\bar{\eta}} = \begin{bmatrix} N_{1,1} & N_{2,1} & \dots \\ N_{1,2} & N_{2,2} & \dots \end{bmatrix}, \quad \underline{\underline{\mathbf{B}}}_{\bar{\eta}\bar{\eta}} = \begin{bmatrix} N_{1,11} & N_{2,11} & \dots \\ N_{1,22} & N_{2,22} & \dots \\ \sqrt{2}N_{1,12} & \sqrt{2}N_{2,12} & \dots \end{bmatrix} \end{aligned} \quad (6.21)$$

gives the following matrix-vector notation for Eq. (6.17)

$$\begin{aligned} \delta \underline{\underline{\mathbf{u}}}^T \underbrace{\int_{\Omega} \underline{\underline{\mathbf{B}}}_u^T (1 - \omega) \underline{\underline{\mathbf{C}}} \underline{\underline{\mathbf{B}}}_u \, dV \underline{\underline{\mathbf{u}}}}_{\underline{\underline{\mathbf{f}}}_u^{\text{int}}(\bar{\eta}, \underline{\underline{\mathbf{u}}})} - \delta \underline{\underline{\mathbf{u}}}^T \underbrace{\int_{\partial\Omega_h} \underline{\underline{\mathbf{N}}}_u^T \underline{\underline{\mathbf{h}}} \, dA}_{\underline{\underline{\mathbf{f}}}_u^{\text{ext}}} = 0, \\ \delta \underline{\underline{\bar{\eta}}}^T \underbrace{\left(\int_{\Omega} \underline{\underline{\mathbf{N}}}_{\bar{\eta}} \underline{\underline{\mathbf{N}}}_{\bar{\eta}}^T + \frac{l_c^2}{2} \underline{\underline{\mathbf{B}}}_{\bar{\eta}}^T \underline{\underline{\mathbf{B}}}_{\bar{\eta}} + \frac{l_c^4}{8} \underline{\underline{\mathbf{B}}}_{\bar{\eta}\bar{\eta}}^T \underline{\underline{\mathbf{B}}}_{\bar{\eta}\bar{\eta}} \, dV \underline{\underline{\bar{\eta}}} - \int_{\Omega} \underline{\underline{\mathbf{N}}}_{\bar{\eta}} \eta \, dV \right)}_{\underline{\underline{\mathbf{f}}}_{\bar{\eta}}^{\text{int}}(\bar{\eta}, \underline{\underline{\mathbf{u}}})} = 0. \end{aligned} \quad (6.22)$$

Setting $\underline{\underline{\mathbf{h}}} = \mathbf{0}$, using arbitrary $\delta \underline{\underline{\mathbf{u}}}$, $\delta \underline{\underline{\bar{\eta}}}$ and taking into account nodal forces $\lambda \hat{\underline{\underline{\mathbf{f}}}}$ yields the following system of equations to be solved

$$\underline{\underline{\mathbf{H}}}(\bar{\eta}, \underline{\underline{\mathbf{u}}}, \lambda) = \begin{bmatrix} \underline{\underline{\mathbf{f}}}_{\bar{\eta}}^{\text{int}}(\bar{\eta}, \underline{\underline{\mathbf{u}}}) \\ \underline{\underline{\mathbf{f}}}_u^{\text{int}}(\bar{\eta}, \underline{\underline{\mathbf{u}}}) - \lambda \hat{\underline{\underline{\mathbf{f}}}} \\ \varphi(\underline{\underline{\mathbf{u}}}, \lambda) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ 0 \end{bmatrix}, \quad (6.23)$$

where λ is the loading parameter, $\hat{\underline{\underline{\mathbf{f}}}}$ the normalised load vector and φ an arc-length function. Here, the arc-length function based on the rates of internal $\dot{\mathcal{U}}$ and dissipated energy $\dot{\mathcal{E}}^D$ from Chapter 2 is exploited. Linearisation of Eq. (6.23) yields the solution for the $i+1$ -th iteration in the $k+1$ -th increment for $\bar{\eta}_{k+1}^{i+1}$, $\underline{\underline{\mathbf{u}}}_{k+1}^{i+1}$ and λ_{k+1}^{i+1}

$$\begin{bmatrix} \bar{\eta} \\ \underline{\underline{\mathbf{u}}} \\ \lambda \end{bmatrix}_{k+1}^{i+1} = \begin{bmatrix} \bar{\eta} \\ \underline{\underline{\mathbf{u}}} \\ \lambda \end{bmatrix}_{k+1}^i - \underline{\underline{\mathbf{K}}}_T^{-1} \Big|_{k+1}^i \cdot \begin{bmatrix} \underline{\underline{\mathbf{f}}}_{\bar{\eta}}^{\text{int}}(\bar{\eta}, \underline{\underline{\mathbf{u}}}) \\ \underline{\underline{\mathbf{f}}}_u^{\text{int}}(\bar{\eta}, \underline{\underline{\mathbf{u}}}) - \lambda \hat{\underline{\underline{\mathbf{f}}}} \\ \varphi(\underline{\underline{\mathbf{u}}}, \lambda) \end{bmatrix}_{k+1}^i, \quad (6.24)$$

with the tangential stiffness matrix

$$\underline{\mathbf{K}}_T(\underline{\bar{\eta}}, \underline{\mathbf{u}}, \lambda) = \begin{bmatrix} \frac{\partial \underline{\mathbf{f}}_{\bar{\eta}}^{\text{int}}(\underline{\bar{\eta}}, \underline{\mathbf{u}})}{\partial \underline{\bar{\eta}}} & \frac{\partial \underline{\mathbf{f}}_{\bar{\eta}}^{\text{int}}(\underline{\bar{\eta}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}} & \underline{\mathbf{0}} \\ \frac{\partial \underline{\mathbf{f}}_{\mathbf{u}}^{\text{int}}(\underline{\bar{\eta}}, \underline{\mathbf{u}})}{\partial \underline{\bar{\eta}}} & \frac{\partial \underline{\mathbf{f}}_{\mathbf{u}}^{\text{int}}(\underline{\bar{\eta}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}} & -\underline{\hat{\mathbf{f}}} \\ \underline{\mathbf{0}}^T & \frac{\partial \varphi(\underline{\mathbf{u}}, \lambda)}{\partial \underline{\mathbf{u}}} & \frac{\partial \varphi(\underline{\mathbf{u}}, \lambda)}{\partial \lambda} \end{bmatrix}. \quad (6.25)$$

6.1.3 Numerical examples

In this section, POWELL–SABIN B-splines are used for the L-shaped specimen considered by VERHOOSSEL *et al.* [115]. Afterwards, POWELL–SABIN B-splines are utilised for a single-edge notched beam (SENB) subject to an antisymmetric four-point shear loading.

6.1.3.1 L-shaped specimen

The set-up for the L-shaped specimen is given in Fig. 6.1. Parameters and coefficients are taken in accordance with VERHOOSSEL *et al.* [115].

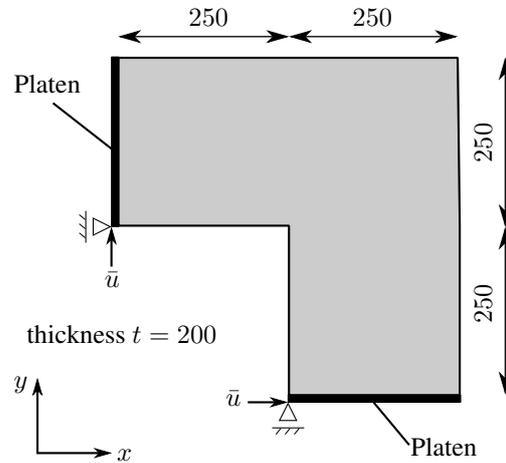


Fig. 6.1: Set-up for the L-shaped specimen, dimensions in mm.

The L-shaped sample has been discretised using NURBS, T-splines and POWELL–SABIN B-splines. A C_A^1 -continuous NURBS mesh is depicted in Fig. 6.2. This mesh contains two pairs of overlapping control points which results in a vanishing determinant of the JACOBIAN matrix $J_{ij} = \frac{\partial x_i}{\partial \xi_j}$. It has been demonstrated by LIPTON *et al.* [68] that these discretisations can still be robust. However, when several NURBS-patches are used, then they join with C_A^0 -continuity.

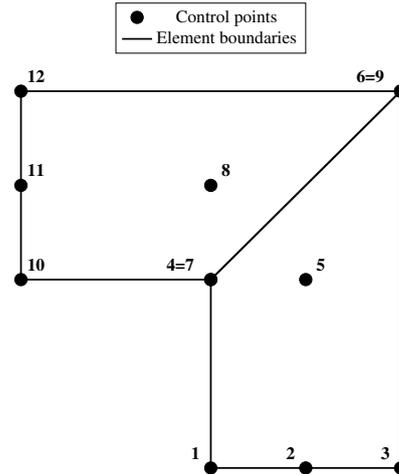


Fig. 6.2: Quadratic C_A^1 -continuous NURBS mesh for the knot vectors $\Xi = \{0, 0, 0, 1, 1, 1\}$ and $\mathcal{H} = \{0, 0, 0, \frac{1}{2}, 1, 1, 1\}$.

In order to obtain for an L-shaped geometry C_A^1 -continuity everywhere using T-splines, the mesh depicted in Fig. 6.3 can be used. Utilising the BÉZIER extraction operator as in Chapter 4 yields that the T-spline mesh is standard, i. e. all blending functions sum to one. Furthermore, the BÉZIER extraction operator is a square matrix in each element. The blue marked corner in Fig. 6.3(b) is not interpolatory, i. e. there is no control point. Modelling the L-shaped geometry with an interpolatory control point would introduce a C_A^0 -continuity into the mesh.

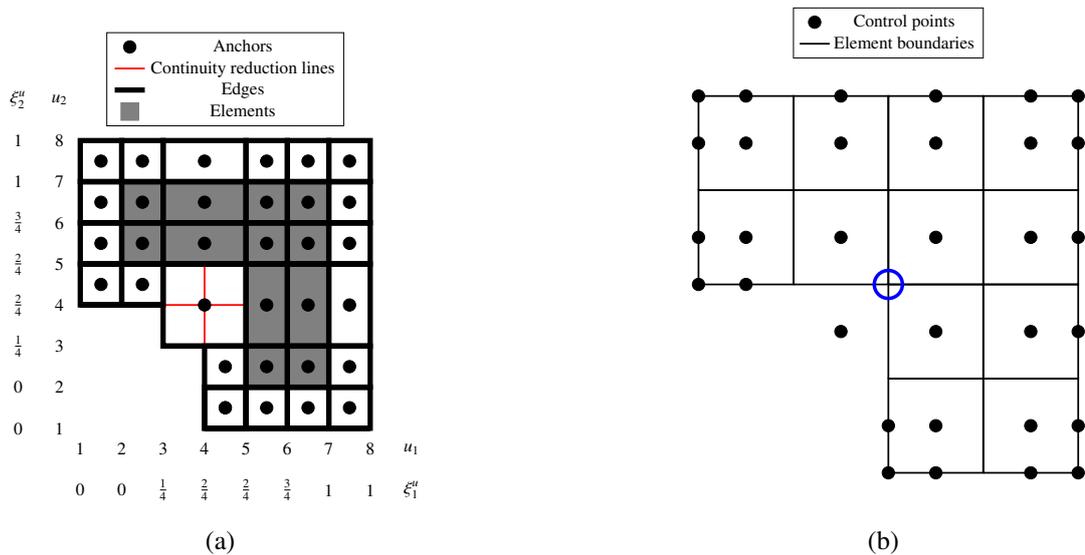


Fig. 6.3: Quadratic C_A^1 -continuous T-spline mesh for the L-shaped geometry in (a) the index (u_ℓ) and parameter (ξ_ℓ^u) domain and (b) the physical (x_ℓ) domain.

Alternatively, Fig. 6.4 shows the POWELL–SABIN B-spline mesh that can be obtained

directly from an existing triangulation \mathcal{T} , see also Chapter 5 for an introduction into POWELL–SABIN B-splines.

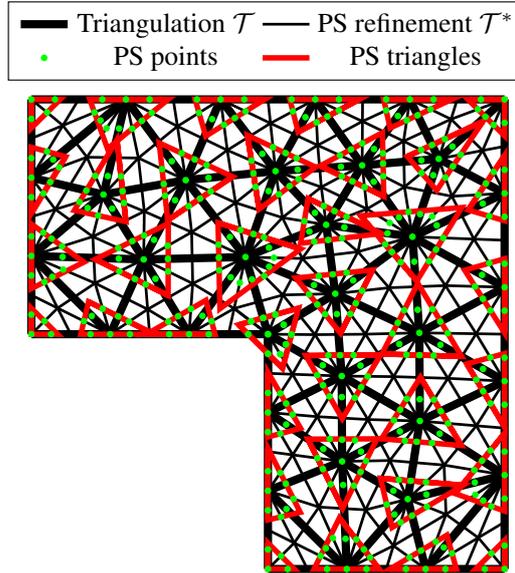


Fig. 6.4: Triangulation \mathcal{T} (thick black lines), POWELL–SABIN refinement \mathcal{T}^* (thin black lines), POWELL–SABIN points (green) and POWELL–SABIN triangles (red) for an L-shaped geometry.

Here and in the following, the POWELL–SABIN triangles on the boundary are constrained as follows: for an angle of $\gamma < 180^\circ$ two edges of the POWELL–SABIN triangle lie on the boundary; for an angle of $\gamma = 180^\circ$, one edge of the POWELL–SABIN triangle lies on the boundary, see also Fig. 6.4. γ refers to the angle at which two boundary edges intersect at a boundary vertex. No restriction is imposed on the POWELL–SABIN triangles on an internal discontinuity (see Section 6.2).

The linear constraints along the platens in Fig. 6.1 are imposed using a master-slave relation as follows (see also Fig. 6.5 for the bottom platen).

Let \underline{u}_A and \underline{u}_B be the displacement at the left and right corner of the bottom platen with coordinates x_A and x_B , respectively. A and B are the master control points. Then, for any slave control point S with coordinates x_S along the edge, its displacement \underline{u}_S is set to

$$\underline{u}_S = \left(1 - \frac{x_S - x_A}{x_B - x_A}\right) \underline{u}_A + \frac{x_S - x_A}{x_B - x_A} \underline{u}_B. \quad (6.26)$$

This master-slave relation needs to be incorporated into Eq. (6.24) as described by FELIPPA [41, Chapter 8]. Eq. (6.26) follows from the intercept theorem

$$\frac{\underline{u}_S - \underline{u}_A}{x_S - x_A} = \frac{\underline{u}_B - \underline{u}_A}{x_B - x_A}. \quad (6.27)$$

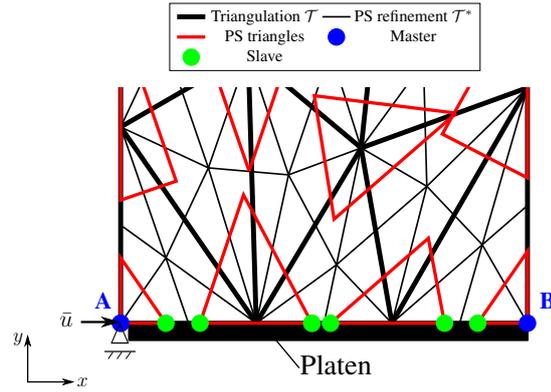


Fig. 6.5: Linear constraints along the bottom platen in Fig. 6.1 with POWELL–SABIN B-splines.

The following parameters are used: YOUNG’s modulus $E = 10 \cdot 10^3$ MPa, POISSON’S ratio $\nu = 0.2$ and length scale $l_c = 5\sqrt{2}$ mm. Plane stress is assumed. The thickness of the specimen is taken as $t = 200$ mm.

The modified VON MISES local equivalent strain defined by DE VREE *et al.* [30]

$$\eta(\boldsymbol{\varepsilon}) = \frac{k-1}{2k(1-2\nu)} I_1(\boldsymbol{\varepsilon}) + \frac{1}{2k} \sqrt{\left(\frac{k-1}{1-2\nu} I_1(\boldsymbol{\varepsilon})\right)^2 + \frac{12k}{(1+\nu)^2} J_2(\boldsymbol{\varepsilon})}, \quad (6.28)$$

with the first invariant of the strain tensor,

$$I_1(\boldsymbol{\varepsilon}) = \varepsilon_{kk}, \quad (6.29)$$

and the second invariant of the deviatoric strain tensor,

$$J_2(\boldsymbol{\varepsilon}) = \frac{1}{2} \varepsilon_{ij} \varepsilon_{ij} - \frac{1}{6} I_1(\boldsymbol{\varepsilon})^2, \quad (6.30)$$

is used. In Eq. (6.28), $k = 10$ in order to account for different strengths in compression and tension. The following damage law from GEERS *et al.* [44]

$$\omega(\kappa) = \begin{cases} 0 & \kappa \leq \kappa_0 \\ 1 - \frac{\kappa_0}{\kappa} \left(1 - \alpha + \alpha \exp(\beta(\kappa_0 - \kappa))\right) & \kappa > \kappa_0 \end{cases}, \quad (6.31)$$

with $\kappa_0 = 4 \cdot 10^{-4}$, $\alpha = 0.98$ and $\beta = 80$ is utilised. A displacement control is applied to the left corner of the bottom platen and the bottom corner of the left platen with an increment of $\Delta \bar{u} = 0.05$ mm. The meshes for the POWELL–SABIN B-splines are refined uniformly. Fig. 6.6 shows the force-displacement curves for four different mesh sizes. The legend entry displays the number of degrees of freedom (DOF).

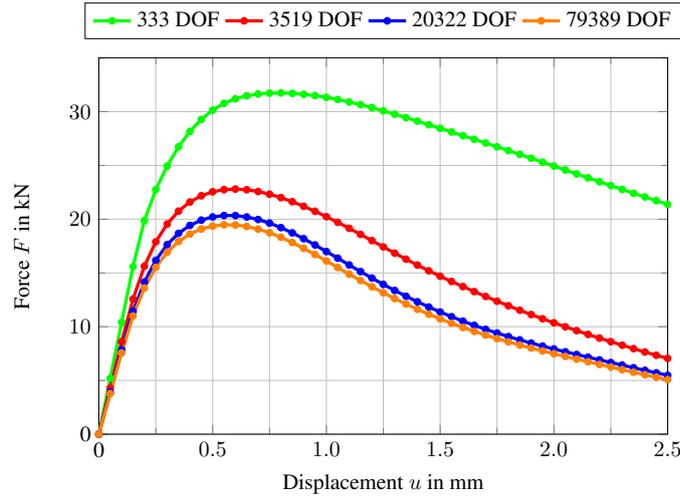


Fig. 6.6: Force-displacement curves for the L-shaped sample using the implicit fourth order gradient damage model for four different mesh sizes.

Fig. 6.7 shows the contours for the damage field ω and the first principal stress σ_1 for the mesh with 20322 DOF at a displacement of $\bar{u} = 1.9$ mm.

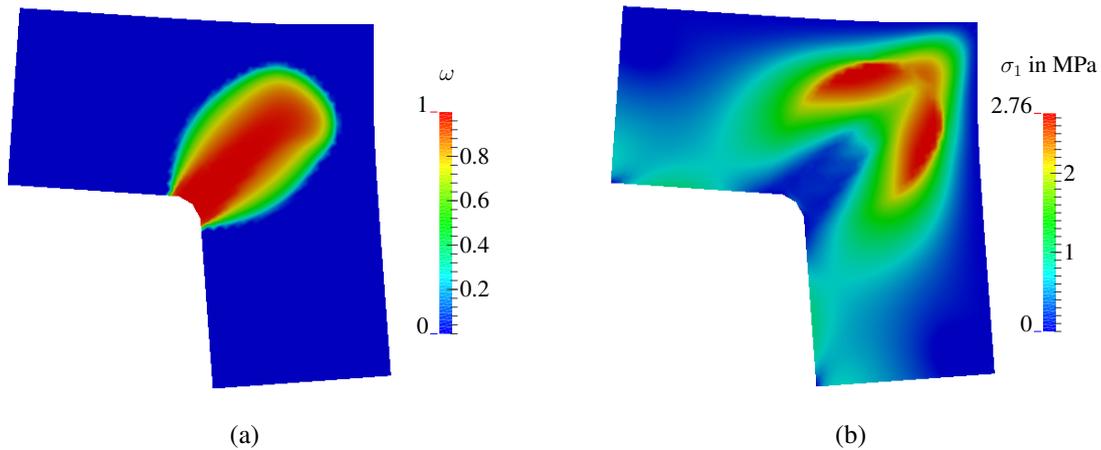


Fig. 6.7: L-shaped sample in the deformed configuration at a displacement of $\bar{u} = 1.9$ mm for the mesh with 20322 DOF. Displacements are amplified by a factor of 20. Contours for (a) damage field ω and (b) first principal stress field σ_1 .

A comparison of the force-displacement curve for the mesh with 20322 DOF when solving the second order and fourth order gradient damage formulation is depicted in Fig. 6.8.

6.1.3.2 Single-edge notched beam

As a second example, the SENB is considered. The parameters and coefficients are taken from GUTIÉRREZ [49]. The geometry and the boundary conditions are given in Fig. 6.9.

boundary, $\alpha_k^j = 0$ for the third control point. This yields with Eq. (5.21a) the required prescribed load for the bottom platen. Furthermore, the relation

$$\underline{u}_S = \left(1 - \frac{x_S - x_E}{x_F - x_E}\right) \underline{u}_E + \frac{x_S - x_E}{x_F - x_E} \underline{u}_F \quad (6.33)$$

is incorporated into Eq. (6.24) where S are the slave control points (green) and E and F are the master control points (blue).

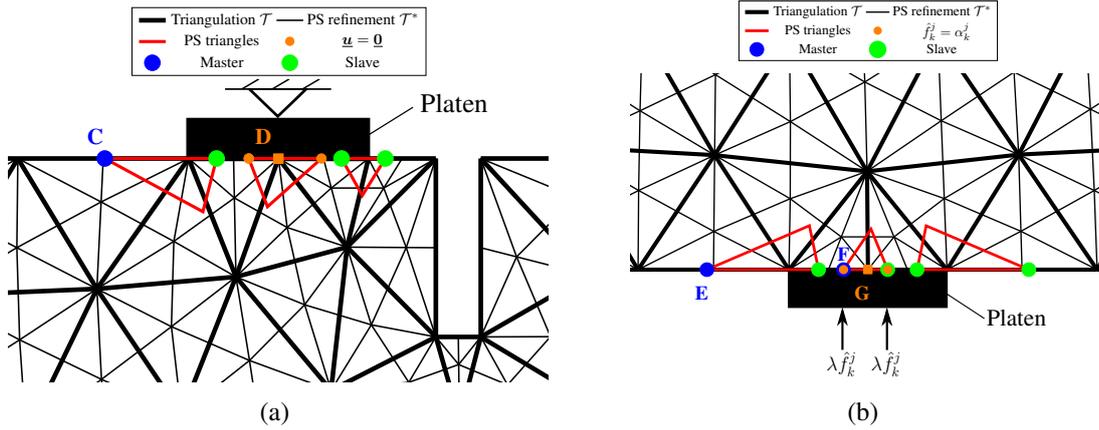


Fig. 6.10: Applying the linear constraints along the platens in Fig. 6.9 to the control points. Control points are indicated by circles, vertices by squares.

For the fixed boundary condition at the top right, the displacement in the vertical direction of the two control points (marked orange) in Fig. 6.11(a) is set to zero. The boundary condition at the bottom left is imposed by setting $\hat{f}_k^j = \alpha_k^j$ for the two control points (orange) in Fig. 6.11(b).

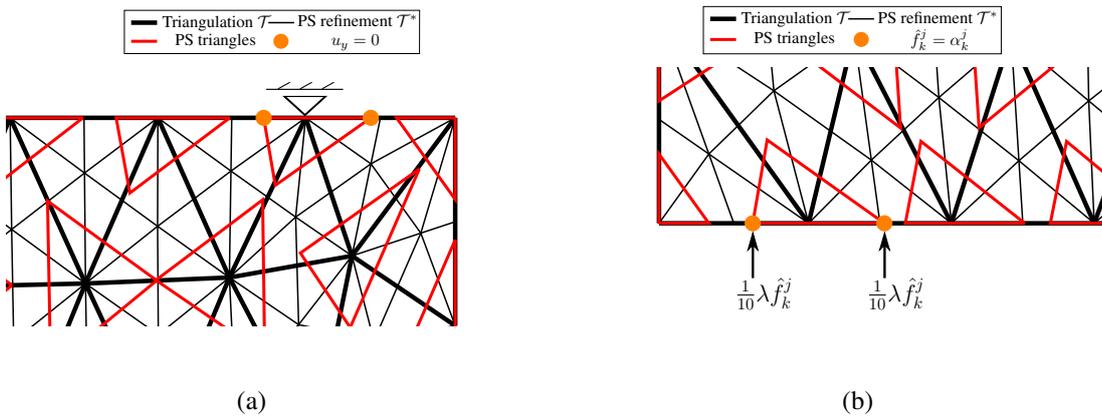


Fig. 6.11: Applying the boundary conditions in Fig. 6.9 to the control points.

The material parameters of the concrete are: YOUNG’s modulus $E = 35 \cdot 10^3$ MPa and POISSON’s ratio $\nu = 0.2$ under a plane stress assumption. The thickness of the specimen

is taken as $t = 100$ mm. The length scale parameter is set to $l_c = \sqrt{2}$ mm². Again, the modified VON MISES local equivalent strain from Eq. (6.28) and the damage law from Eq. (6.31) with the parameters $k = 10$, $\kappa_0 = 6 \cdot 10^{-5}$, $\alpha = 0.96$ and $\beta = 100$ are used. In order to trace the equilibrium path, the arc-length method from Chapter 2 is employed with the parameters $\Delta\tau = 8$ kN and $a = 0.25$.

The meshes are refined uniformly. Fig. 6.12 shows the force-displacement curves. The crack mouth sliding displacement (CMSD, difference in vertical displacement between the right and left notch tips) is plotted along the x -axis and the force $P = \frac{11}{10}\lambda$ is plotted along the y -axis.

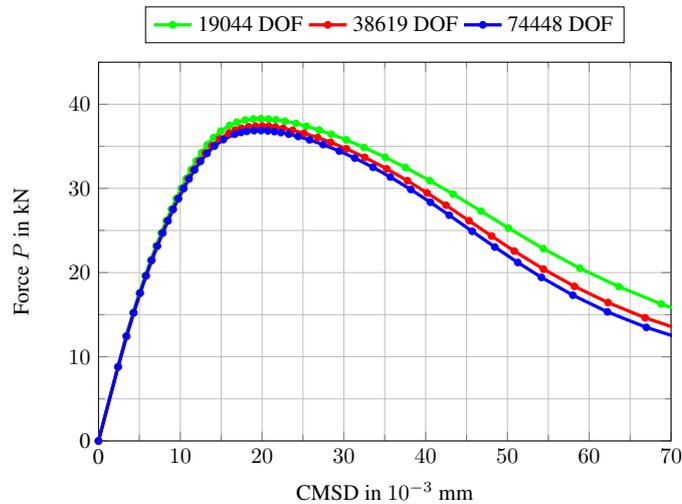


Fig. 6.12: Force-displacement curves for the SENB with the implicit fourth order gradient damage model for three different mesh sizes.

Fig. 6.13(a) shows for the SENB with 74448 DOF at a CMSD of 0.04 mm the damage field ω and stress field σ_{xx} , respectively. For coarser meshes (less than 19044 DOF), the SENB may fail due to the damage zone which starts to grow at the bottom left edge while the damage zone at the notch does not propagate. Hence, fine meshes are required in order to obtain results that are in agreement with the experiments by SCHLANGEN [92].

6.2 Cohesive zone modelling

In the following, the continuum and numerical formulation for a discrete approach to fracture adding cohesive elements along the crack path are derived. Then, the re-meshing algorithm is explained. As a numerical example, the SENB is considered again. After demonstrating the approach with POWELL–SABIN B-splines, the approach using structured T-splines for cohesive zone modelling is discussed.

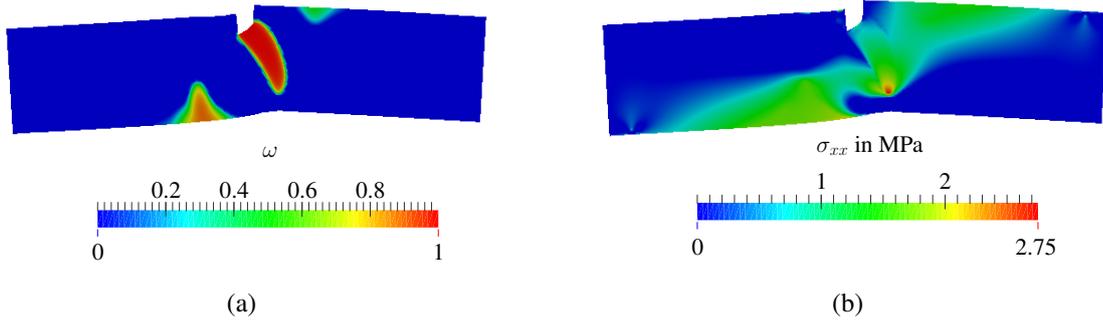


Fig. 6.13: SENB in the deformed configuration at a CMSD of 0.04 mm for the mesh with 74448 DOF. Displacements are amplified by a factor of 200. Contours for (a) damage field ω and (b) stress field σ_{xx} .

6.2.1 Continuum formulation

For the cohesive zone modelling, the stress equilibrium

$$\sigma_{ij,i} = 0 \quad (6.34)$$

subject to the boundary conditions

$$\sigma_{ij}n_j = h_i \quad \text{on } \partial\Omega_h, \quad u_i = \bar{u}_i \quad \text{on } \partial\Omega_u \quad (6.35)$$

needs to be solved. The boundary $\partial\Omega$ is decomposed into the parts $\partial\Omega_h$ and $\partial\Omega_u$ ($\partial\Omega_h \cap \partial\Omega_u = \emptyset$, $\partial\Omega_h \cup \partial\Omega_u = \partial\Omega$), see Fig. 6.14(a). In Eq. (6.35), \mathbf{h} and $\bar{\mathbf{u}}$ are the prescribed surface traction and prescribed displacement, respectively. \mathbf{n} is the normal vector on the boundary $\partial\Omega_h$.

A discontinuity Γ_d is present within the domain Ω . It consists of two overlapping boundaries, Γ_d^+ and Γ_d^- see Fig. 6.14(b). Along each boundary Γ_d^+ and Γ_d^-

$$\sigma_{ij}n_j^+ = t_i^+ \quad \text{on } \Gamma_d^+, \quad \sigma_{ij}n_j^- = t_i^- \quad \text{on } \Gamma_d^- \quad (6.36)$$

are imposed. \mathbf{t}^+ and \mathbf{t}^- are the tractions in the cohesive zone on the positive and negative side, respectively. \mathbf{n}^+ and \mathbf{n}^- are the normal vectors on Γ_d^+ and Γ_d^- . Setting

$$t_i = t_i^- = \sigma_{ij}n_j^- \quad (6.37)$$

results in

$$t_i^+ = \sigma_{ij}n_j^+ = -\sigma_{ij}n_j^- = -t_i. \quad (6.38)$$

In contrast to Eq. (6.3) for the damage model, the stress tensor $\boldsymbol{\sigma}$ in Eq. (6.34) does not

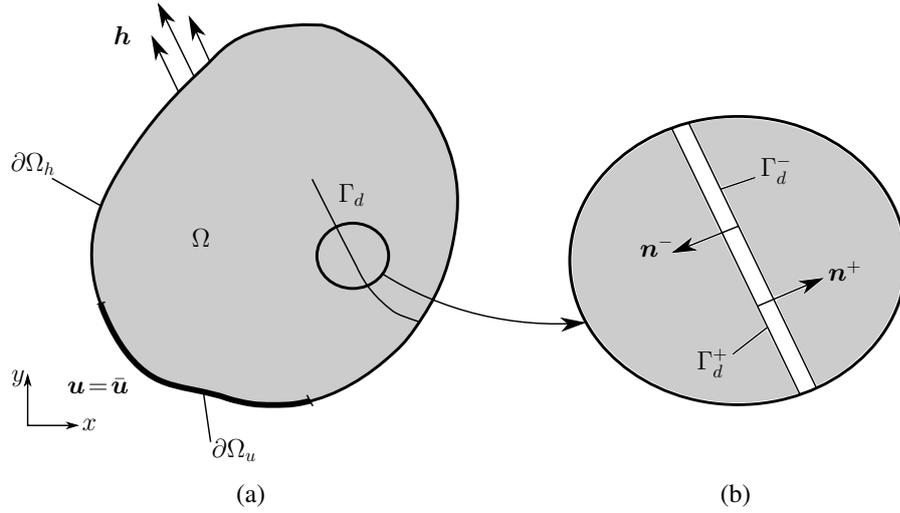


Fig. 6.14: (a) Domain Ω with internal discontinuity Γ_d and decomposition of the boundary $\partial\Omega$ into $\partial\Omega_h$ and $\partial\Omega_u$. (b) Discontinuity Γ_d with overlapping positive and negative side Γ_d^+ and Γ_d^- , respectively.

depend on a damage parameter and is computed with

$$\sigma_{ij} = C_{ijkl}\varepsilon_{kl}, \quad (6.39)$$

where ε is the infinitesimal strain tensor

$$\varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}). \quad (6.40)$$

Multiplying Eq. (6.34) by a test function δu_j and integrating over the domain Ω yields after integration by parts, use of GAUSS' theorem and Eqs. (6.35) – (6.38)

$$\int_{\Omega} \delta u_{i,j} \sigma_{ij} \, dV + \int_{\Gamma_d^+} \delta u_i^+ t_i \, dA - \int_{\Gamma_d^-} \delta u_i^- t_i \, dA = \int_{\partial\Omega_h} \delta u_i h_i \, dA. \quad (6.41)$$

Since

$$\int_{\Gamma_d^+} \dots \, dA = \int_{\Gamma_d^-} \dots \, dA = \int_{\Gamma_d} \dots \, dA, \quad (6.42)$$

Eq. (6.41) can be written with the crack opening $\llbracket u_i \rrbracket = u_i^+ - u_i^-$ as

$$\int_{\Omega} \delta u_{i,j} \sigma_{ij} \, dV + \int_{\Gamma_d} \delta \llbracket u_i \rrbracket t_i \, dA = \int_{\partial\Omega_h} \delta u_i h_i \, dA. \quad (6.43)$$

6.2.2 Finite element formulation

Discretisation of the domain Ω into E elements,

$$\Omega = \bigcup_{e=1}^E \Omega^e, \quad (6.44)$$

and approximation of the the displacement u_i and its derivatives,

$$\underline{\mathbf{u}}^e = \underline{\mathbf{N}}_u \mathbf{u}, \quad \delta \underline{\mathbf{u}}^e = \underline{\mathbf{N}}_u \delta \mathbf{u}, \quad \underline{\boldsymbol{\varepsilon}}^e = \underline{\mathbf{B}}_u \mathbf{u}, \quad \delta \underline{\boldsymbol{\varepsilon}}^e = \underline{\mathbf{B}}_u \delta \mathbf{u}, \quad (6.45)$$

with

$$\underline{\mathbf{u}}^e = \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix}, \quad \delta \underline{\mathbf{u}}^e = \begin{bmatrix} \delta u_1^e \\ \delta u_2^e \end{bmatrix}, \quad \underline{\boldsymbol{\varepsilon}}^e = \begin{bmatrix} \varepsilon_{11}^e \\ \varepsilon_{22}^e \\ 2\varepsilon_{12}^e \end{bmatrix}, \quad \delta \underline{\boldsymbol{\varepsilon}}^e = \begin{bmatrix} \delta \varepsilon_{11}^e \\ \delta \varepsilon_{22}^e \\ 2\delta \varepsilon_{12}^e \end{bmatrix} \quad (6.46)$$

and

$$\underline{\mathbf{N}}_u = \begin{bmatrix} N_1 & 0 & N_2 & 0 & \dots \\ 0 & N_1 & 0 & N_2 & \dots \end{bmatrix}, \quad \underline{\mathbf{B}}_u = \begin{bmatrix} N_{1,1} & 0 & N_{2,1} & 0 & \dots \\ 0 & N_{1,2} & 0 & N_{2,2} & \dots \\ N_{1,2} & N_{1,1} & N_{2,2} & N_{2,1} & \dots \end{bmatrix} \quad (6.47)$$

results in the matrix-vector notation for Eq. (6.43)

$$\delta \underline{\mathbf{u}}^T \left(\underbrace{\int_{\Omega} \underline{\mathbf{B}}_u^T \underline{\mathbf{C}} \underline{\mathbf{B}}_u \, dV \mathbf{u} + \int_{\Gamma_d} (\underline{\mathbf{N}}_u^{+T} - \underline{\mathbf{N}}_u^{-T}) \underline{\mathbf{t}} \, dA}_{\underline{\mathbf{f}}_u^{\text{int}}(\mathbf{u})} \right) - \delta \underline{\mathbf{u}}^T \underbrace{\int_{\partial\Omega_h} \underline{\mathbf{N}}_u^T \underline{\mathbf{h}} \, dA}_{\underline{\mathbf{f}}_u^{\text{ext}}} = 0. \quad (6.48)$$

The traction $\underline{\mathbf{t}}$ in the global coordinate system can be obtained from the local traction $\underline{\mathbf{t}}_d$ with

$$\underline{\mathbf{t}} = \underline{\mathbf{Q}}^T \underline{\mathbf{t}}_d, \quad (6.49)$$

where $\underline{\mathbf{Q}}$ is the orthogonal rotation matrix,

$$\underline{\mathbf{Q}} = \underline{\mathbf{Q}}^- = \begin{bmatrix} n_x^- & n_y^- \\ -n_y^- & n_x^- \end{bmatrix}, \quad (6.50)$$

and $\underline{\mathbf{n}}^-$ the normal vector from the minus side of the internal discontinuity, Γ_d^- , see Fig. 6.14(b). $\underline{\mathbf{t}}_d$ itself depends on the crack opening $[[\underline{\mathbf{u}}_d]]$ in the local coordinate system of the discontinuity Γ_d , $\underline{\mathbf{t}}_d = \underline{\mathbf{t}}_d([[\underline{\mathbf{u}}_d]])$. The local jump $[[\underline{\mathbf{u}}_d]]$ can be evaluated from the transformation

$$[[\underline{\mathbf{u}}_d]] = \underline{\mathbf{Q}} [[\mathbf{u}]]. \quad (6.51)$$

The vectors for the local traction \underline{t}_d and local jump $[[\underline{u}_d]]$ consist of a normal and shear component

$$\underline{t}_d = \begin{bmatrix} t_n \\ t_s \end{bmatrix}, \quad [[\underline{u}_d]] = \begin{bmatrix} [[u_n]] \\ [[u_s]] \end{bmatrix}. \quad (6.52)$$

The BÉZIER ordinates of the BÉZIER extraction operator in Eq. (5.31) that are required to compute the matrices $\underline{\underline{N}}_u^+$ and $\underline{\underline{N}}_u^-$ in Eq. (6.48) are different on both sides of the internal discontinuity Γ_d . This stems from the fact that pairs of POWELL–SABIN triangles on either side of Γ_d are in general not symmetric, see also Fig. 6.15.

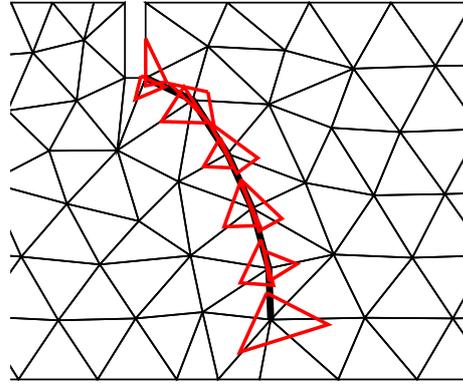


Fig. 6.15: Pairs of POWELL–SABIN triangles (red) along the discontinuity Γ_d (thick black line) are not symmetric. This results in different BÉZIER ordinates for the BÉZIER extraction operator and in different basis functions $\underline{\underline{N}}_u^+$ and $\underline{\underline{N}}_u^-$ on both sides.

With $\underline{h}=\underline{0}$, arbitrary $\delta\underline{u}$ and nodal forces $\lambda\hat{\underline{f}}$ for Eq. (6.48), the following system of equations needs to be solved

$$\underline{\underline{H}}(\underline{u}, \lambda) = \begin{bmatrix} \underline{f}_u^{\text{int}}(\underline{u}) - \lambda\hat{\underline{f}} \\ \varphi(\underline{u}, \lambda) \end{bmatrix} = \underline{0}, \quad (6.53)$$

where λ is the loading parameter, $\hat{\underline{f}}$ the normalised load vector and φ an arc-length function. Linearisation of Eq. (6.53) yields the solution for the $i+1$ -th iteration in the $k+1$ -th increment for $\underline{u}_{k+1}^{i+1}$ and λ_{k+1}^{i+1}

$$\begin{bmatrix} \underline{u} \\ \lambda \end{bmatrix}_{k+1}^{i+1} = \begin{bmatrix} \underline{u} \\ \lambda \end{bmatrix}_{k+1}^i - \underline{\underline{K}}_T^{-1} \Big|_{k+1}^i \cdot \begin{bmatrix} \underline{f}_u^{\text{int}}(\underline{u}) - \lambda\hat{\underline{f}} \\ \varphi(\underline{u}, \lambda) \end{bmatrix}_{k+1}^i, \quad (6.54)$$

with the tangential stiffness matrix

$$\underline{\underline{\mathbf{K}}}_T(\underline{\mathbf{u}}, \lambda) = \begin{bmatrix} \frac{\partial \underline{\mathbf{f}}_u^{\text{int}}(\underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}} & -\hat{\underline{\mathbf{f}}} \\ \frac{\partial \varphi(\underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}} & \frac{\partial \varphi(\underline{\mathbf{u}}, \lambda)}{\partial \lambda} \end{bmatrix}. \quad (6.55)$$

Different from Section 6.1, a dissipation-based arc-length method for φ is not used. Since the crack does not grow in each increment, new cohesive elements are not inserted in each step. Therefore, the amount of energy that can be dissipated is different depending on whether a new cohesive element has been added or not. Thus, an arc-length method which controls the crack mouth sliding displacement (CMSD) is exploited, see also DE BORST [28]. However, the arc-length method employed here is slightly different from DE BORST [28]. The constraint

$$u_k^{\text{R}} - u_k^{\text{L}} = k \Delta \bar{u}_{\text{CMSD}} \quad (6.56)$$

is enforced, where u_k^{R} is the displacement on the right and u_k^{L} the displacement on the left tip of the notch (see Fig. 6.9) in the k -th increment and $\Delta \bar{u}_{\text{CMSD}}$ the prescribed CMSD. In the $k+1$ -th increment,

$$u_{k+1}^{\text{R}} - u_{k+1}^{\text{L}} = (k+1) \Delta \bar{u}_{\text{CMSD}} \quad (6.57)$$

needs to be fulfilled. Subtracting Eq. (6.56) from Eq. (6.57) results in

$$u_{k+1}^{\text{R}} - u_k^{\text{R}} - (u_{k+1}^{\text{L}} - u_k^{\text{L}}) = (k+1 - k) \Delta \bar{u}_{\text{CMSD}}, \quad (6.58)$$

$$\Delta u^{\text{R}} - \Delta u^{\text{L}} = \Delta \bar{u}_{\text{CMSD}}, \quad (6.59)$$

so that the following arc-length method has to be satisfied in each increment

$$\varphi(\underline{\mathbf{u}}_{k+1}) = \Delta u^{\text{R}} - \Delta u^{\text{L}} - \Delta \bar{u}_{\text{CMSD}}. \quad (6.60)$$

Since φ depends only on the displacement vector $\underline{\mathbf{u}}$ in Eq. (6.60), the term $\frac{\partial \varphi(\underline{\mathbf{u}}, \lambda)}{\partial \lambda}$ becomes zero in Eq. (6.55). This would result in a singular tangential stiffness matrix $\underline{\underline{\mathbf{K}}}_T$. Therefore, Eq. (6.54) cannot be used to solve the problem. However, the linearised form of Eq. (6.53) can be solved for $\underline{\mathbf{u}}_{k+1}^{i+1}$ and λ_{k+1}^{i+1} in a staggered manner according to DE

BORST *et al.* [29, Chapter 4]. In a first step,

$$\delta \underline{\mathbf{u}}_{k+1}^{i+1 I} = \left(\frac{\partial \underline{\mathbf{f}}_u^{\text{int}}(\underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}} \Big|_{k+1}^i \right)^{-1} \cdot \hat{\underline{\mathbf{f}}}, \quad (6.61)$$

$$\delta \underline{\mathbf{u}}_{k+1}^{i+1 II} = \left(\frac{\partial \underline{\mathbf{f}}_u^{\text{int}}(\underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}} \Big|_{k+1}^i \right)^{-1} \cdot \left(\lambda_{k+1}^i \hat{\underline{\mathbf{f}}} - \underline{\mathbf{f}}_u^{\text{int}}(\underline{\mathbf{u}}_{k+1}^i) \right) \quad (6.62)$$

are solved. Then,

$$\delta \lambda_{k+1}^{i+1} = \frac{-\varphi - \frac{\partial \varphi}{\partial \underline{\mathbf{u}}} \delta \underline{\mathbf{u}}_{k+1}^{i+1 II}}{\frac{\partial \varphi}{\partial \underline{\mathbf{u}}} \delta \underline{\mathbf{u}}_{k+1}^{i+1 I} + \frac{\partial \varphi}{\partial \lambda}}, \quad (6.63)$$

$$\delta \underline{\mathbf{u}}_{k+1}^{i+1} = \delta \lambda_{k+1}^{i+1} \delta \underline{\mathbf{u}}_{k+1}^{i+1 I} + \delta \underline{\mathbf{u}}_{k+1}^{i+1 II} \quad (6.64)$$

are computed which finally results in

$$\underline{\mathbf{u}}_{k+1}^{i+1} = \underline{\mathbf{u}}_{k+1}^i + \delta \underline{\mathbf{u}}_{k+1}^{i+1}, \quad (6.65)$$

$$\lambda_{k+1}^{i+1} = \lambda_{k+1}^i + \delta \lambda_{k+1}^{i+1}. \quad (6.66)$$

6.2.3 Re-meshing

After finding an equilibrium in increment $k+1$ for Eq. (6.53), the first principal stress σ_1 at the crack tip (black vertex in Fig. 6.16) is evaluated. Due to the \mathcal{C}_A^1 -continuity of the POWELL–SABIN B-splines, σ_1 can be computed directly at the vertex. When the first principal stress exceeds the critical stress t_{ult} , the vector normal to the first principal stress direction is computed. The location, where this vector intersects an element edge is highlighted red in Fig. 6.16. This marks the location of the new crack tip. As a next step, elements in the neighbourhood of the new crack tip have to be re-meshed while the positions of the vertices along the crack path and the new crack tip are held fixed. Also, vertices on and outside the blue polygon in Fig. 6.16 are not allowed to move and re-meshing is applied within the blue polygon. For local re-meshing, the approach by PERSSON & STRANG [83] is used which interprets the mesh as a truss structure and finds an equilibrium such that forces at the vertices become zero.

After re-meshing, the new POWELL–SABIN triangles have to be determined locally and Eq. (6.53) has to be solved again for increment $k+1$ with the extended crack. When solving Eq. (6.53) for increment $k+1$, the displacement vector $\tilde{\underline{\mathbf{u}}}_k$ for increment k on the updated mesh is required. The vector $\tilde{\underline{\mathbf{u}}}_k$ can be obtained by solving locally the following

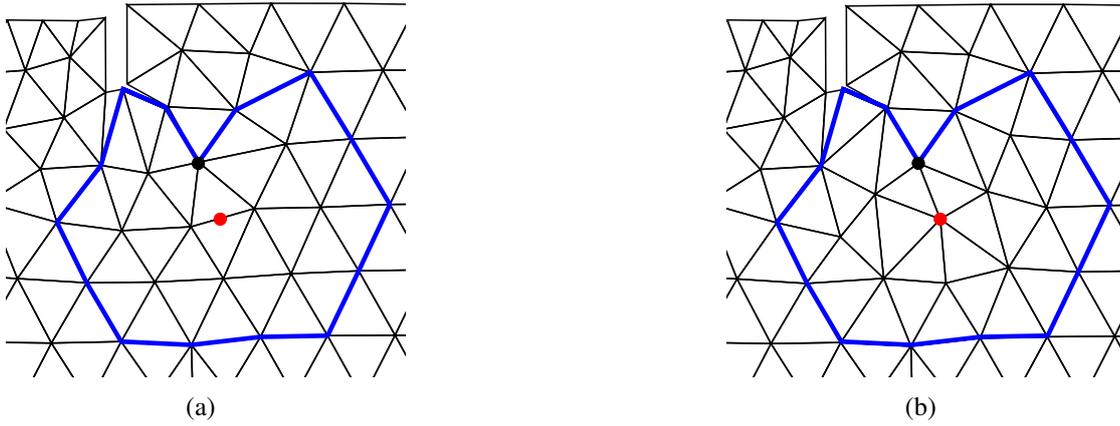


Fig. 6.16: Mesh in the deformed configuration (a) before and (b) after re-meshing. The black circle marks the old crack tip where the critical stress t_{ult} is exceeded. The red circle shows the location of the new crack tip. Within the blue polygon, a local re-meshing procedure is applied. Vertices along the existing crack path, the new crack tip and vertices on and outside the blue polygon are held fixed.

problem in the re-meshed domain

$$\min \left(\frac{1}{2} \int_{\Omega} \|\underline{\tilde{\mathbf{N}}}_u \tilde{\mathbf{u}}_k - \underline{\mathbf{N}}_u \mathbf{u}_k\|_2^2 dV \right), \quad (6.67)$$

where $\underline{\tilde{\mathbf{N}}}_u$ contains the basis functions on the updated mesh. Eq. (6.67) results in the following system

$$\int_{\Omega} \underline{\tilde{\mathbf{N}}}_u^T \underline{\tilde{\mathbf{N}}}_u \tilde{\mathbf{u}}_k dV = \int_{\Omega} \underline{\tilde{\mathbf{N}}}_u \underline{\mathbf{N}}_u \mathbf{u}_k dV, \quad (6.68)$$

which can be solved directly for $\tilde{\mathbf{u}}_k$.

6.2.4 Single-edge notched beam

As a numerical example for cohesive zone modelling, the SENB from Section 6.1.3.2 is re-considered. Parameters and coefficients are taken from VERHOESEL *et al.* [116]. The material parameters are: YOUNG's modulus $E = 35 \cdot 10^3$ MPa and POISSON's ratio $\nu = 0.2$ with a plane stress assumption. The concrete is modelled linearly elastic. The thickness of the beam is $t = 100$ mm and the value of the critical stress is set to $t_{\text{ult}} = 2.8$ MPa. For the local traction \underline{t}_d , the relations from WELLS & SLUYS [122] are used. The cohesive law (see Fig. 6.17) for the normal direction reads

$$t_n = t_{\text{ult}} \exp \left(-\frac{t_{\text{ult}}}{\mathcal{G}_c} \kappa \right), \quad (6.69)$$

where the history parameter κ is determined by the loading function

$$f = \llbracket u_n \rrbracket - \kappa, \quad (6.70)$$

and evolves according to the KARUSH–KUHN–TUCKER conditions in Eq. (6.5). The critical energy release rate and penetration stiffness are taken as $\mathcal{G}_c = 0.1 \text{ N/mm}$ and $k_p = 10^4 \text{ MPa/mm}$, respectively.

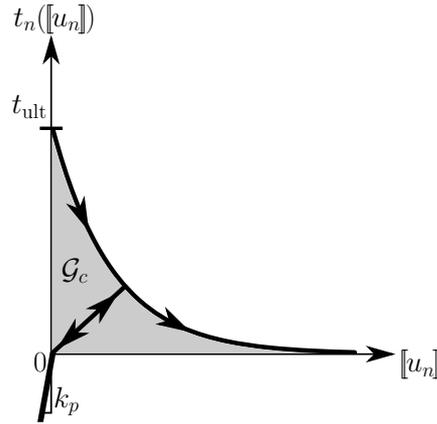


Fig. 6.17: Cohesive traction law $t_n(\llbracket u_n \rrbracket)$ in the normal direction. The branch represents unloading for the case $\llbracket u_n \rrbracket < \kappa$. The shaded grey area is equivalent to the critical energy release rate \mathcal{G}_c . The assumption that $t_n(\llbracket u_n \rrbracket)$ is a material law is not always correct since $t_n(\llbracket u_n \rrbracket)$ also depends on the specimen size, see ANDERSON [4, Chapter 6].

In the shear direction,

$$t_s = d_{\text{int}} \llbracket u_s \rrbracket \quad (6.71)$$

is utilised, with the initial shear stiffness $d_{\text{int}} = 1 \text{ MPa/mm}$. At the onset of the simulation, $\Delta \bar{u}_{\text{CMSD}} = 1.5 \cdot 10^{-3} \text{ mm}$. Fig. 6.18 shows the resulting force-displacement curves for the SENB for six different mesh sizes. The mesh is refined uniformly. Along the x -axis, the CMSD is plotted, while the y -axis shows the reaction force $P = \frac{11}{10} \lambda$. The legend entry gives the number of DOF at the beginning and at the end of the simulation.

Fig. 6.19 shows the final crack paths for six different mesh sizes.

In Fig. 6.20, the contour plot for the stress field σ_{xx} at a CMSD of 0.04 mm for the mesh with 49632 – 49860 DOF is depicted. The mesh size is equivalent to the one using the implicit fourth order gradient damage model with 74448 DOF in Fig. 6.13.

Finally, Fig. 6.21 gives an overlap of the initial and final mesh with 744 and 768 DOF, respectively.

The discrete approach with cohesive zones along the crack path requires less degrees of freedom than the implicit fourth order gradient damage model for numerical results that are in good agreement with the experiments carried out by SCHLANGEN [92]. Also, since

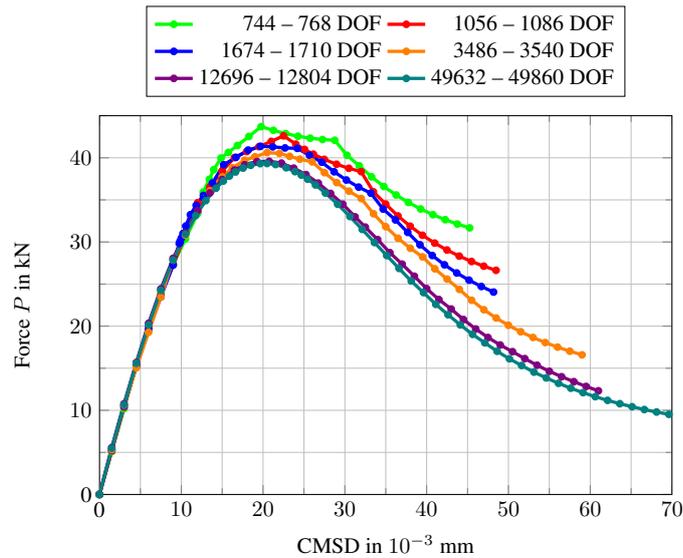


Fig. 6.18: Force-displacement curves for the SENB with cohesive zone modelling for six different mesh sizes.

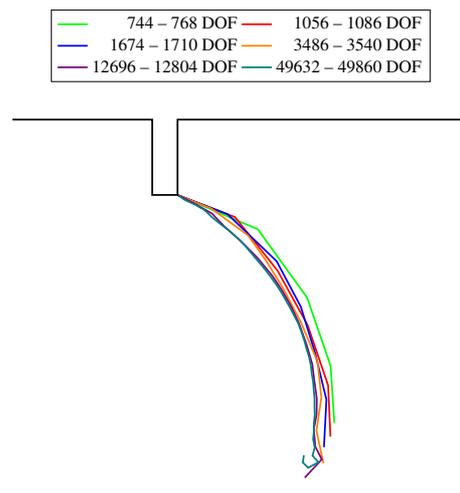


Fig. 6.19: Final crack paths for six different mesh sizes.

derivatives of second order are required for the implicit fourth order gradient damage model, the computation of the tangential stiffness matrix and force vector is more expensive. Nevertheless, implementation is easier for the implicit fourth order gradient damage model since no re-meshing is necessary.

6.2.5 Structured T-splines for the single-edge notched beam

Structured T-splines have already been used for cohesive zone modelling for the SENB by VERHOOSSEL *et al.* [116]. Fig. 6.22(a) shows the initial structured T-spline mesh. In order to determine the position of the control points, Eq. (5.47) has to be solved, while

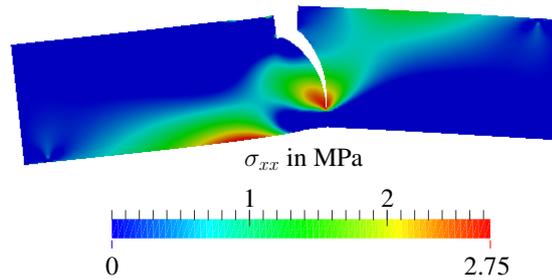


Fig. 6.20: Stress field σ_{xx} for the SENB in the deformed configuration at a CMSD of 0.04 mm for the mesh with 49632 – 49860 DOF. Displacements are amplified by a factor of 200. The mesh size is equivalent to the one showing the results for the implicit fourth order gradient damage model in Fig. 6.13.

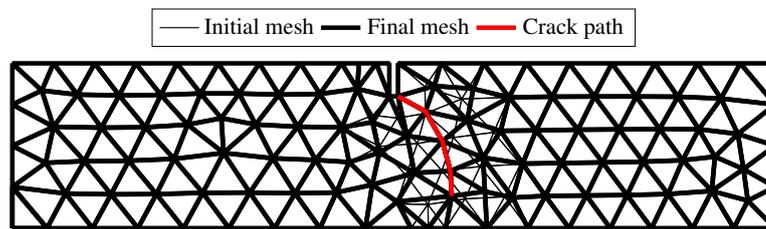


Fig. 6.21: Comparison of the initial (744 DOF) and the final mesh (768 DOF).

the location of some control points has to be prescribed, for instance control points on the boundary. It can be observed from Fig. 6.22 that apart from the position of the control points on the boundary also the position of the blue control points in the interior has to be prescribed. Furthermore, the blue control points have to be positioned such that the initial mesh has to be aligned with the final crack path (red). Then, elements can be split vertically. Not prescribing the location of the blue control points results in the mesh in Fig. 6.22(b) after solving Eq. (5.47). It can be seen that this would result in inverted elements around the notch. Since the final crack path is not always known a priori, structured T-splines may not be ideal for modelling discrete fracture. Furthermore, the continuity across elements sharing one edge that touches the crack tip is only C_A^0 , so that stresses are discontinuous and need to be evaluated and weighted at the integration points in the vicinity of the crack tip in order to determine when the critical stress t_{ult} is reached. This is similar to LAGRANGIAN basis functions.

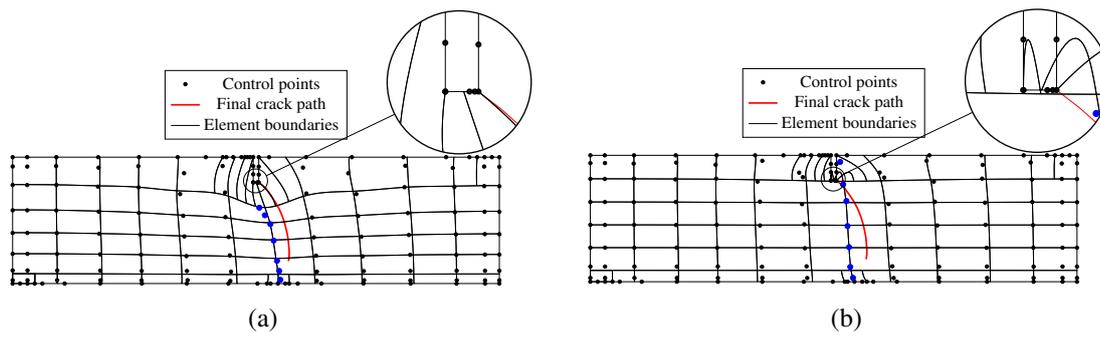


Fig. 6.22: Cohesive zone modelling using structured T-splines. (a) T-spline mesh from VERHOUSEL *et al.* [116] when the location of the blue control points is prescribed. (b) Not prescribing the location of the blue control points results in inverted elements around the notch.

7 Summary

In the present thesis, an energy-based arc-length method has been proposed which is especially useful for tracing an equilibrium path with multiple snap-through and / or snap-back phenomena for physically non-linear problems. It switches between internal energy-based and dissipation-based arc-length control. The arc-length method has been applied to interface elements with a cohesive traction law, the phase field model for brittle fracture and the implicit higher order gradient damage model for quasi-brittle materials.

Concerns were raised about the phase field models for brittle and cohesive fracture. While the phase field model for cohesive fracture does not pass a two-dimensional patch test for an unstructured mesh, Γ -convergence is not attained numerically for the phase field model for brittle fracture using a one-dimensional bar. It was concluded that since the smeared numerical final crack surface $\Gamma_{\ell,h}$ does not converge to the discrete crack surface Γ for a vanishing length scale parameter ℓ and a mesh size $h \ll \ell$, the smeared numerical potential $\Pi_{\ell,h}$ cannot converge to the discrete potential Π either. Moreover, it was demonstrated that the phase field model for brittle fracture is sensitive to the imposed boundary conditions. These issues must be further improved.

The BÉZIER extraction operator has been exploited for the determination of linear dependencies in T-spline meshes and classification of T-spline meshes into standard, semi-standard and non-standard. The BÉZIER extraction operator was further used for refining T-spline meshes by adding anchors, such that the initial and the refined T-spline mesh are nested. Furthermore, it has been demonstrated that hierarchical refinement of standard, semi-standard and non-standard T-spline meshes using the reconstruction operator basically also involves the BÉZIER extraction operator since the reconstruction operator is its inverse. It has been explained how to modify the BÉZIER extraction operator for quadratic and cubic unstructured T-spline meshes with extraordinary points such that the resulting T-spline mesh is standard and C_A^1 -continuous around the extraordinary point. Extending these methodologies to the three-dimensional case leaves room for future research.

BÉZIER extraction for POWELL–SABIN B-splines has been introduced for an efficient computation. POWELL–SABIN B-splines have been utilised for smeared and discrete fracture approaches. Unfortunately, POWELL–SABIN B-splines are not based on the isogeometric concept since they only approximate the exact geometry. However, they provide basis functions which are C_A^1 -continuous in the entire domain. This allows a direct computation of stresses at the crack tip for discrete fracture approaches which eases the

implementation. Also, POWELL–SABIN B-splines gave lower errors than unstructured T-splines or multiple NURBS-patches for solving the KIRCHHOFF–LOVE plate with simply supported and clamped boundary conditions. Although there has been a lot of research in extending POWELL–SABIN B-splines to the three-dimensional case in the past years, there still exists no solution for unstructured meshes. However, this would contribute to a wider acceptance of this element technology.

A Energy-based arc-length control for physically non-linear problems

A.1 SHERMAN–MORRISON formula

The inverse of the matrix in Eq. (2.7)

$$\underline{\underline{\mathbf{K}}}_T = \begin{bmatrix} \underline{\underline{\mathbf{K}}} & -\hat{\underline{\underline{f}}} \\ \underline{\underline{\mathbf{v}}}^T & w \end{bmatrix} \quad (\text{A.1})$$

can be obtained as follows. Rewriting Eq. (A.1) gives

$$\underline{\underline{\mathbf{K}}}_T = \begin{bmatrix} \underline{\underline{\mathbf{K}}} & \underline{\underline{\mathbf{0}}} \\ \underline{\underline{\mathbf{0}}}^T & 1 \end{bmatrix} - \underline{\underline{\mathbf{x}}}_1 \underline{\underline{\mathbf{y}}}_1^T - \underline{\underline{\mathbf{x}}}_2 \underline{\underline{\mathbf{y}}}_2^T = \underbrace{\underline{\underline{\mathbf{A}}} - \underline{\underline{\mathbf{A}}}_1 - \underline{\underline{\mathbf{A}}}_2}_{\underline{\underline{\mathbf{B}}}} \quad (\text{A.2})$$

with

$$\underline{\underline{\mathbf{x}}}_1 = \begin{bmatrix} \hat{\underline{\underline{f}}} \\ 0 \end{bmatrix}, \quad \underline{\underline{\mathbf{y}}}_1^T = \begin{bmatrix} \underline{\underline{\mathbf{0}}}^T & 1 \end{bmatrix} \quad \rightarrow \underline{\underline{\mathbf{x}}}_1 \underline{\underline{\mathbf{y}}}_1^T = \begin{bmatrix} \underline{\underline{\mathbf{0}}} & \hat{\underline{\underline{f}}} \\ \underline{\underline{\mathbf{0}}}^T & 0 \end{bmatrix}, \quad (\text{A.3})$$

$$\underline{\underline{\mathbf{x}}}_2 = \begin{bmatrix} \underline{\underline{\mathbf{0}}} \\ -1 \end{bmatrix}, \quad \underline{\underline{\mathbf{y}}}_2^T = \begin{bmatrix} \underline{\underline{\mathbf{v}}}^T & w - 1 \end{bmatrix} \quad \rightarrow \underline{\underline{\mathbf{x}}}_2 \underline{\underline{\mathbf{y}}}_2^T = \begin{bmatrix} \underline{\underline{\mathbf{0}}} & \underline{\underline{\mathbf{0}}} \\ -\underline{\underline{\mathbf{v}}}^T & 1 - w \end{bmatrix}. \quad (\text{A.4})$$

Application of the SHERMAN–MORRISON [101] formula yields the following expressions

$$\underline{\underline{\mathbf{K}}}_T^{-1} = (\underline{\underline{\mathbf{B}}} - \underline{\underline{\mathbf{A}}}_2)^{-1} = (\underline{\underline{\mathbf{B}}} - \underline{\underline{\mathbf{x}}}_2 \underline{\underline{\mathbf{y}}}_2^T)^{-1} = \underline{\underline{\mathbf{B}}}^{-1} + \frac{\underline{\underline{\mathbf{B}}}^{-1} \underline{\underline{\mathbf{x}}}_2 \underline{\underline{\mathbf{y}}}_2^T \underline{\underline{\mathbf{B}}}^{-1}}{1 - \underline{\underline{\mathbf{y}}}_2^T \underline{\underline{\mathbf{B}}}^{-1} \underline{\underline{\mathbf{x}}}_2}, \quad (\text{A.5})$$

$$\underline{\underline{\mathbf{B}}}^{-1} = (\underline{\underline{\mathbf{A}}} - \underline{\underline{\mathbf{A}}}_1)^{-1} = (\underline{\underline{\mathbf{A}}} - \underline{\underline{\mathbf{x}}}_1 \underline{\underline{\mathbf{y}}}_1^T)^{-1} = \underline{\underline{\mathbf{A}}}^{-1} + \frac{\underline{\underline{\mathbf{A}}}^{-1} \underline{\underline{\mathbf{x}}}_1 \underline{\underline{\mathbf{y}}}_1^T \underline{\underline{\mathbf{A}}}^{-1}}{1 - \underline{\underline{\mathbf{y}}}_1^T \underline{\underline{\mathbf{A}}}^{-1} \underline{\underline{\mathbf{x}}}_1}, \quad (\text{A.6})$$

while the terms in Eq. (A.6) can be expressed as

$$\underline{\underline{\mathbf{A}}}^{-1} = \begin{bmatrix} \underline{\underline{\mathbf{K}}}^{-1} & \underline{\mathbf{0}} \\ \underline{\mathbf{0}}^T & 1 \end{bmatrix}, \quad \underline{\underline{\mathbf{A}}}^{-1} \underline{\mathbf{x}}_1 = \begin{bmatrix} \underline{\underline{\mathbf{K}}}^{-1} \hat{\underline{\mathbf{f}}} \\ 0 \end{bmatrix}, \quad \underline{\mathbf{y}}_1^T \underline{\underline{\mathbf{A}}}^{-1} = \begin{bmatrix} \underline{\mathbf{0}}^T & 1 \end{bmatrix}, \quad (\text{A.7})$$

$$\underline{\underline{\mathbf{A}}}^{-1} \underline{\mathbf{x}}_1 \underline{\mathbf{y}}_1^T \underline{\underline{\mathbf{A}}}^{-1} = \begin{bmatrix} \underline{\mathbf{0}} & \underline{\underline{\mathbf{K}}}^{-1} \hat{\underline{\mathbf{f}}} \\ \underline{\mathbf{0}}^T & 0 \end{bmatrix}, \quad \underline{\mathbf{y}}_1^T \underline{\underline{\mathbf{A}}}^{-1} \underline{\mathbf{x}}_1 = 0, \quad (\text{A.8})$$

$$\underline{\underline{\mathbf{B}}}^{-1} = \begin{bmatrix} \underline{\underline{\mathbf{K}}}^{-1} & \underline{\mathbf{0}} \\ \underline{\mathbf{0}}^T & 1 \end{bmatrix} + \begin{bmatrix} \underline{\mathbf{0}} & \underline{\underline{\mathbf{K}}}^{-1} \hat{\underline{\mathbf{f}}} \\ \underline{\mathbf{0}}^T & 0 \end{bmatrix} \quad (\text{A.9})$$

and in Eq. (A.5) as

$$\underline{\underline{\mathbf{B}}}^{-1} \underline{\mathbf{x}}_2 = \begin{bmatrix} -\underline{\underline{\mathbf{K}}}^{-1} \hat{\underline{\mathbf{f}}} \\ -1 \end{bmatrix}, \quad \underline{\mathbf{y}}_2^T \underline{\underline{\mathbf{B}}}^{-1} = \begin{bmatrix} \underline{\mathbf{v}}^T \underline{\underline{\mathbf{K}}}^{-1} & \underline{\mathbf{v}}^T \underline{\underline{\mathbf{K}}}^{-1} \hat{\underline{\mathbf{f}}} + w - 1 \end{bmatrix}, \quad (\text{A.10})$$

$$\underline{\underline{\mathbf{B}}}^{-1} \underline{\mathbf{x}}_2 \underline{\mathbf{y}}_2^T \underline{\underline{\mathbf{B}}}^{-1} = \begin{bmatrix} -\underline{\underline{\mathbf{K}}}^{-1} \hat{\underline{\mathbf{f}}} \underline{\mathbf{v}}^T \underline{\underline{\mathbf{K}}}^{-1} & -\underline{\underline{\mathbf{K}}}^{-1} \hat{\underline{\mathbf{f}}} \underline{\mathbf{v}}^T \underline{\underline{\mathbf{K}}}^{-1} \hat{\underline{\mathbf{f}}} - w \underline{\underline{\mathbf{K}}}^{-1} \hat{\underline{\mathbf{f}}} + \underline{\underline{\mathbf{K}}}^{-1} \hat{\underline{\mathbf{f}}} \\ -\underline{\mathbf{v}}^T \underline{\underline{\mathbf{K}}}^{-1} & -\underline{\mathbf{v}}^T \underline{\underline{\mathbf{K}}}^{-1} \hat{\underline{\mathbf{f}}} - w + 1 \end{bmatrix}, \quad (\text{A.11})$$

$$\underline{\mathbf{y}}_2^T \underline{\underline{\mathbf{B}}}^{-1} \underline{\mathbf{x}}_2 = -\underline{\mathbf{v}}^T \underline{\underline{\mathbf{K}}}^{-1} \hat{\underline{\mathbf{f}}} - w + 1. \quad (\text{A.12})$$

With

$$\underline{\mathbf{q}} = \underline{\underline{\mathbf{K}}}^{-1} \hat{\underline{\mathbf{f}}} \quad \text{and} \quad \underline{\mathbf{q}} = \frac{\underline{\mathbf{q}}(\underline{\mathbf{v}}^T \underline{\mathbf{q}} + w)}{\underline{\mathbf{v}}^T \underline{\mathbf{q}} + w} = \frac{\underline{\mathbf{q}} \underline{\mathbf{v}}^T \underline{\mathbf{q}} + w \underline{\mathbf{q}}}{\underline{\mathbf{v}}^T \underline{\mathbf{q}} + w} \quad (\text{A.13})$$

Eq. (A.1) becomes

$$\underline{\underline{\mathbf{K}}}_T^{-1} = \begin{bmatrix} \underline{\underline{\mathbf{K}}}^{-1} & \underline{\mathbf{0}} \\ \underline{\mathbf{0}}^T & 1 \end{bmatrix} + \begin{bmatrix} \underline{\mathbf{0}} & \underline{\mathbf{q}} \\ \underline{\mathbf{0}}^T & 0 \end{bmatrix} + \frac{\begin{bmatrix} -\underline{\mathbf{q}} \underline{\mathbf{v}}^T \underline{\underline{\mathbf{K}}}^{-1} & -\underline{\mathbf{q}} \underline{\mathbf{v}}^T \underline{\mathbf{q}} - w \underline{\mathbf{q}} + \underline{\mathbf{q}} \\ -\underline{\mathbf{v}}^T \underline{\underline{\mathbf{K}}}^{-1} & -\underline{\mathbf{v}}^T \underline{\mathbf{q}} - w + 1 \end{bmatrix}}{\underline{\mathbf{v}}^T \underline{\mathbf{q}} + w} \quad (\text{A.14})$$

$$= \begin{bmatrix} \underline{\underline{\mathbf{K}}}^{-1} & \underline{\mathbf{0}} \\ \underline{\mathbf{0}}^T & 1 \end{bmatrix} + \frac{\begin{bmatrix} \underline{\mathbf{0}} & \underline{\mathbf{q}} \underline{\mathbf{v}}^T \underline{\mathbf{q}} + w \underline{\mathbf{q}} \\ \underline{\mathbf{0}}^T & 0 \end{bmatrix}}{\underline{\mathbf{v}}^T \underline{\mathbf{q}} + w} + \frac{\begin{bmatrix} -\underline{\mathbf{q}} \underline{\mathbf{v}}^T \underline{\underline{\mathbf{K}}}^{-1} & -\underline{\mathbf{q}} \underline{\mathbf{v}}^T \underline{\mathbf{q}} - w \underline{\mathbf{q}} + \underline{\mathbf{q}} \\ -\underline{\mathbf{v}}^T \underline{\underline{\mathbf{K}}}^{-1} & -\underline{\mathbf{v}}^T \underline{\mathbf{q}} - w + 1 \end{bmatrix}}{\underline{\mathbf{v}}^T \underline{\mathbf{q}} + w} \quad (\text{A.15})$$

$$= \begin{bmatrix} \underline{\underline{\mathbf{K}}}^{-1} & \underline{\mathbf{0}} \\ \underline{\mathbf{0}}^T & 1 \end{bmatrix} + \frac{1}{\underline{\mathbf{v}}^T \underline{\mathbf{q}} + w} \begin{bmatrix} -\underline{\mathbf{q}} \underline{\mathbf{v}}^T \underline{\underline{\mathbf{K}}}^{-1} & \underline{\mathbf{q}} \\ -\underline{\mathbf{v}}^T \underline{\underline{\mathbf{K}}}^{-1} & -\underline{\mathbf{v}}^T \underline{\mathbf{q}} - w + 1 \end{bmatrix}. \quad (\text{A.16})$$

A.2 Time discretisation scheme for the arc-length control

For the initial-value problem with $t \in [0, T]$

$$\dot{x}(t) = f(x(t)) \quad (\text{A.17})$$

$$x(0) = x_k \quad (\text{A.18})$$

the generalised midpoint rule is defined as follows (see SIMO & HUGHES [102, Chapter 1])

$$f(x_{k+\theta}) = \frac{x_{k+1} - x_k}{\Delta t}, \quad x_{k+\theta} = \theta x_{k+1} + (1 - \theta)x_k, \quad \theta \in [0, 1] \quad (\text{A.19})$$

with $\theta=0$ for forward EULER, $\theta=\frac{1}{2}$ for midpoint rule and $\theta=1$ for backward EULER. x_{k+1} and x_k denote in Eq. (A.19) the solution for the variable x at time increment $k+1$ and k , respectively.

A.2.1 Time discretisation for the rate of internal energy

Applying the time discretisation scheme in Eq. (A.19) to Eq. (2.26)

$$\frac{1}{2} \left(\underline{\dot{\mathbf{u}}}^T \lambda + \underline{\mathbf{u}}^T \dot{\lambda} \right) \hat{\underline{\mathbf{f}}} - \dot{\tau}^U = 0 \quad (\text{A.20})$$

gives

$$\begin{aligned} & \frac{1}{2} \left((\theta \lambda_{k+1} + (1 - \theta) \lambda_k) \frac{\underline{\mathbf{u}}_{k+1}^T - \underline{\mathbf{u}}_k^T}{\Delta t} \right. \\ & \quad \left. + \frac{\lambda_{k+1} - \lambda_k}{\Delta t} (\theta \underline{\mathbf{u}}_{k+1}^T + (1 - \theta) \underline{\mathbf{u}}_k^T) \right) \hat{\underline{\mathbf{f}}} - \frac{\tau_{k+1}^U - \tau_k^U}{\Delta t} \\ & = \frac{1}{2} \left(2\theta \frac{\lambda_{k+1} \underline{\mathbf{u}}_{k+1}^T}{\Delta t} + (1 - 2\theta) \frac{\lambda_{k+1} \underline{\mathbf{u}}_k^T}{\Delta t} \right. \\ & \quad \left. + (1 - 2\theta) \frac{\lambda_k \underline{\mathbf{u}}_{k+1}^T}{\Delta t} + (2\theta - 2) \frac{\lambda_k \underline{\mathbf{u}}_k^T}{\Delta t} \right) \hat{\underline{\mathbf{f}}} - \frac{\Delta \tau^U}{\Delta t} = 0. \quad (\text{A.21}) \end{aligned}$$

Using the midpoint rule with $\theta = \frac{1}{2}$ in Eq. (A.21) yields the arc-length function φ^U

$$\varphi^U(\underline{\mathbf{u}}_{k+1}, \lambda_{k+1}) = \frac{1}{2} (\lambda_{k+1} \underline{\mathbf{u}}_{k+1}^T - \lambda_k \underline{\mathbf{u}}_k^T) \hat{\underline{\mathbf{f}}} - \Delta \tau^U. \quad (\text{A.22})$$

A.2.2 Time discretisation for the rate of dissipated energy

Starting with the constraint equation from Eq. (2.23)

$$\frac{1}{2}(\lambda \dot{\underline{\mathbf{u}}}^T - \dot{\lambda} \underline{\mathbf{u}}^T) \hat{\underline{\mathbf{f}}} - \dot{\tau}^D = 0 \quad (\text{A.23})$$

and application of Eq. (A.19) gives

$$\begin{aligned} \frac{1}{2} \left((\theta \lambda_{k+1} + (1 - \theta) \lambda_k) \frac{\underline{\mathbf{u}}_{k+1}^T - \underline{\mathbf{u}}_k^T}{\Delta t} \right. \\ \left. - \frac{\lambda_{k+1} - \lambda_k}{\Delta t} (\theta \underline{\mathbf{u}}_{k+1}^T + (1 - \theta) \underline{\mathbf{u}}_k^T) \right) \hat{\underline{\mathbf{f}}} - \frac{\tau_{k+1}^D - \tau_k^D}{\Delta t} \\ = \frac{1}{2} \left(\frac{\lambda_k \underline{\mathbf{u}}_{k+1}^T}{\Delta t} - \frac{\lambda_{k+1} \underline{\mathbf{u}}_k^T}{\Delta t} \right) \hat{\underline{\mathbf{f}}} - \frac{\Delta \tau^D}{\Delta t} = 0. \end{aligned} \quad (\text{A.24})$$

Therefore, the arc-length function φ^D for any time discretisation scheme can be written as

$$\varphi^D(\underline{\mathbf{u}}_{k+1}, \lambda_{k+1}) = \frac{1}{2} (\lambda_k \underline{\mathbf{u}}_{k+1}^T - \lambda_{k+1} \underline{\mathbf{u}}_k^T) \hat{\underline{\mathbf{f}}} - \Delta \tau^D. \quad (\text{A.25})$$

B T-splines

B.1 Local refinement of T-splines by adding anchors

This section demonstrates local refinement by adding anchors for standard T-spline meshes of odd degree and non-standard T-splines. Moreover, a strategy for finding the optimum number of additionally inserted anchors will be suggested.

B.1.1 Local refinement of standard T-splines of odd degree by adding anchors

Herein, it is explained how standard T-spline meshes of odd degree can be refined locally by adding anchors while the information from the BÉZIER extraction operator is exploited.

B.1.1.1 Example 1: Ensuring that $\underline{\underline{C}}_e$ is a square matrix and nestedness

Initial refinement

Consider the cubic standard T-spline mesh depicted in Fig. B.1 which is refined as in Fig. B.2(a).

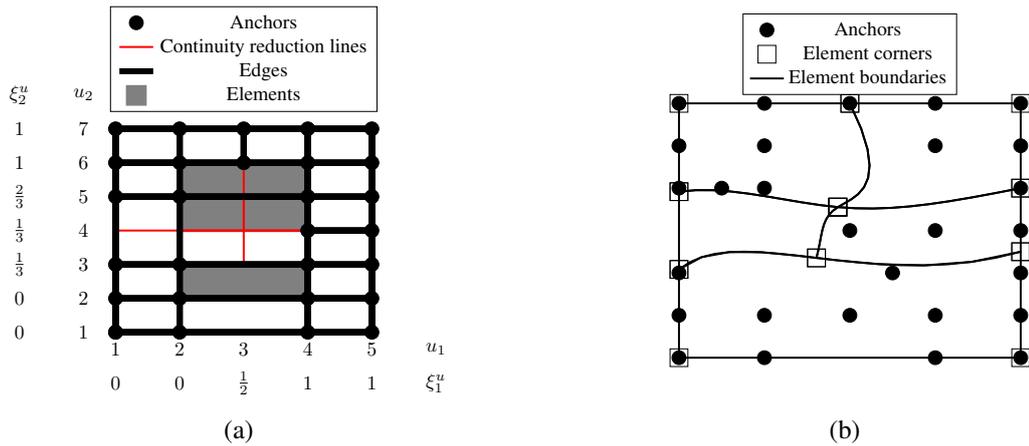


Fig. B.1: Initial cubic standard T-spline mesh in (a) the index domain and (b) the physical domain.

Ensuring that $\underline{\underline{C}}_e$ is a square matrix

The mesh in Fig. B.2(a) is locally linearly independent, but non-standard and $\underline{\underline{C}}_e$ is not a

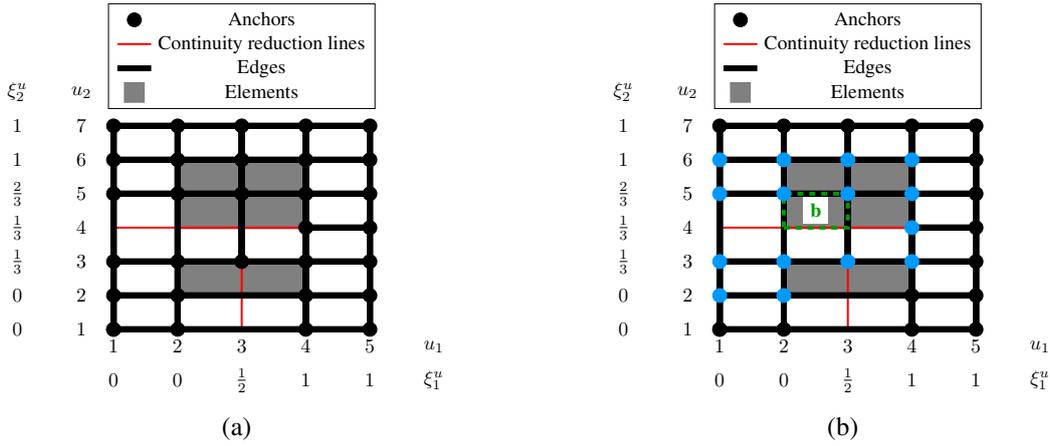


Fig. B.2: (a) Refined cubic non-standard T-spline mesh of Fig. B.1(a) in the index domain; (b) the T-spline mesh is locally linearly independent, but in element b (dashed green line) are only fifteen anchors (blue) with a support and therefore $\underline{\underline{C}}_e$ is not a square matrix for this element.

square matrix for all elements: $\text{rank}(\underline{\underline{C}}_e) = n_e$ in element b as there are only fifteen anchors (blue) with a support, Fig. B.2(b). Hence, an additional anchor needs to be inserted. Each local knot vector of the blue anchors in Fig. B.2(b) contains the sub-parameter values of the boundaries of element b – $[0, \frac{1}{2}] \times [\frac{1}{3}, \frac{2}{3}]$ in the ξ_1 -direction and the ξ_2 -direction, respectively – except for the anchors A and B in Fig. B.3(a). The local knot vectors of the anchors A and B in the ξ_1 -direction do not contain the sub-parameter value $\xi_1 = \frac{1}{2}$. Therefore, an additional anchor needs to be inserted at the location of the red point c. This results in the standard mesh in Fig. B.3(b).

Nestedness

Unfortunately, the initial mesh in Fig. B.1(a) and the refined mesh in Fig. B.3(b) are not nested. Transforming Eq. (4.58) into row echelon form gives no solution for the anchors C, D, E and F in the initial mesh, see Fig. B.4, i. e. the blending functions associated to these anchors in the initial T-spline mesh cannot be represented as a linear combination of the blending functions of the refined T-spline mesh in Fig. B.3(b).

Therefore, an additional anchor has to be inserted. The new edges and anchors of the refined T-spline mesh of Fig. B.3(b) are drawn in the initial T-spline mesh of Fig. B.1(a) as illustrated with solid green lines and points in Fig. B.4. Then, the domain where all four anchors C, D, E and F have a common support is drawn while this domain needs to be cut by the green edge. This domain is indicated by a grey colour. It can be observed that within the grey domain no anchor is sitting at the location of the red point d. Therefore, the red point d represents the location of an anchor which has to be inserted into the T-spline mesh. The resulting T-spline mesh is depicted in Fig. B.5(a). This T-spline mesh

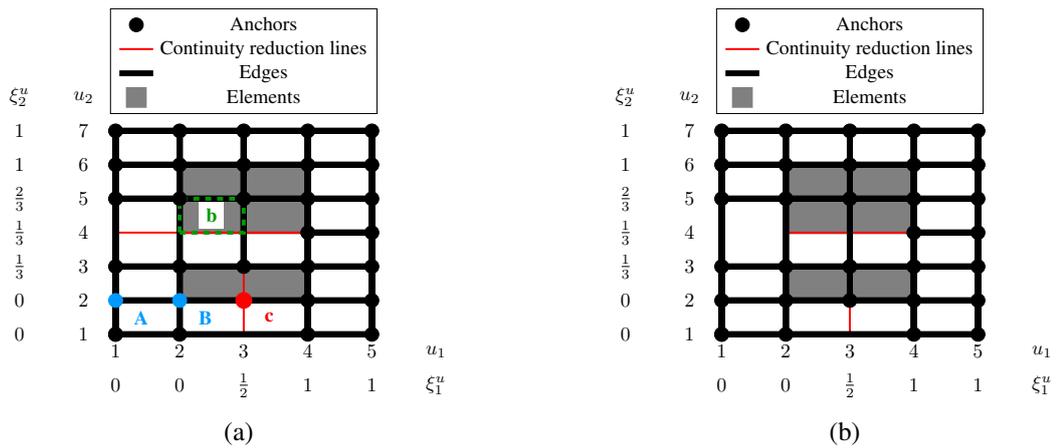


Fig. B.3: Determination of the location of a new anchor when \underline{C}_e is not a square matrix for a cubic T-spline mesh. (a) The local knot vectors in the ξ_1 -direction of the blue anchors A and B do not contain the sub-parameter value $\xi_1 = \frac{1}{2}$, which is a boundary of element b (dashed green). Therefore, an anchor is required at the location of the red point c. (b) The resulting standard T-spline mesh. This T-spline mesh and the initial T-spline mesh in B.1(a) are not nested.

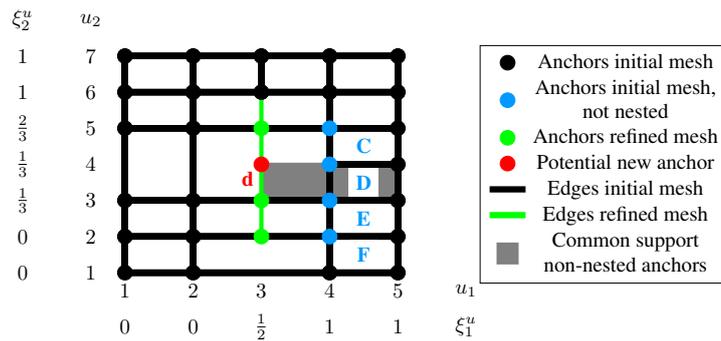


Fig. B.4: Superposition of the initial T-spline mesh in the index domain from Fig. B.1(a) and the refined mesh in Fig. B.3(b). The row echelon form of Eq. (4.58) gives no results for the anchors C, D, E and F (blue) and therefore, the meshes in Fig. B.1(a) and Fig. B.3(b) are not nested. Edges and anchors from the refined mesh in Fig. B.3(b), which were added during refinement, are inserted in the initial mesh from Fig. B.1 and marked with green. In the grey domain all four anchors C, D, E and F have a common support, while the grey domain is bounded by the newly inserted green edges. Within the grey domain, no anchor is at the position of the red point d. In order to obtain a refined mesh which is standard and nested with the initial mesh in Fig. B.1, the anchor d needs to be inserted into the mesh of Fig. B.3(b), see also Fig. B.5.

is standard. Furthermore, the initial T-spline mesh in Fig. B.1(a) and the refined T-spline mesh in Fig. B.5(a) are nested.

Refined physical mesh

After obtaining a standard and nested T-spline mesh in the index domain, the computation

of the physical mesh can be considered. The location of the weighted control points for the refined mesh \underline{P}_{wR} can be determined using Eq. (4.71). The physical mesh after refinement is shown in Fig. B.5(b) which represents the same geometry as the physical mesh in Fig. B.1(b).

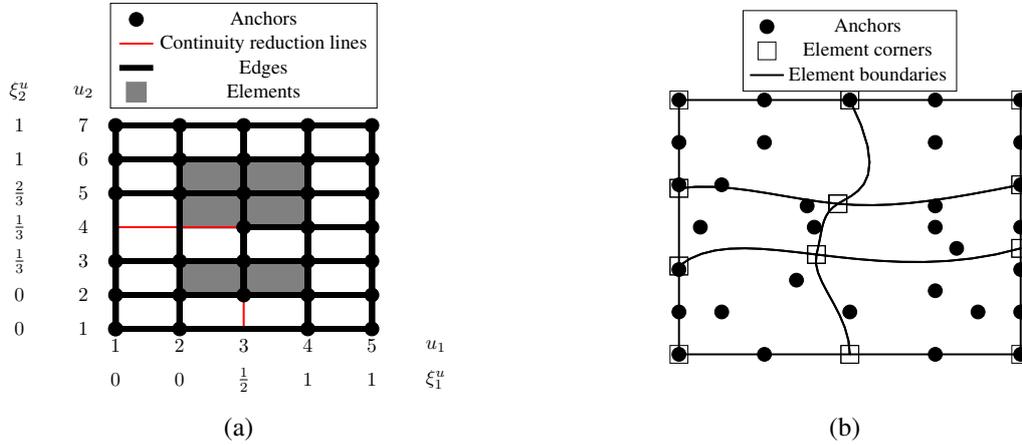


Fig. B.5: Refined cubic T-spline mesh of Fig. B.1 in (a) the index domain and (b) the physical domain. This T-spline mesh is standard and nested with the initial T-spline mesh of Fig. B.1.

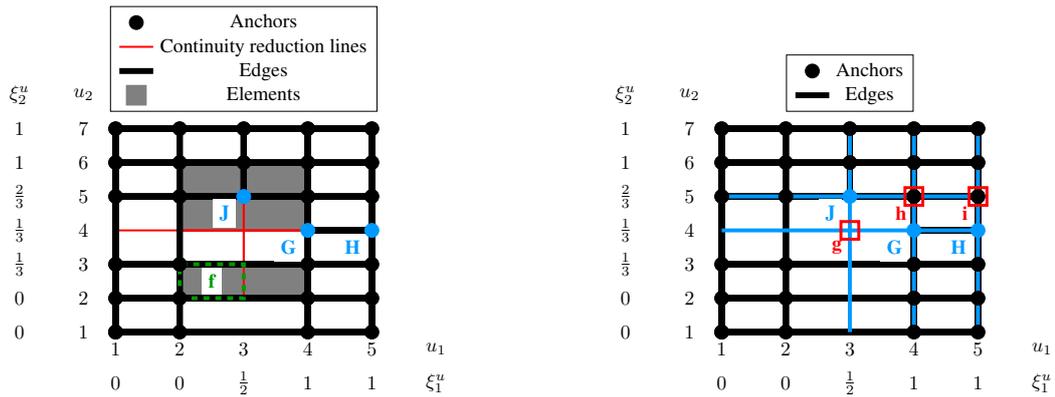
B.1.1.2 Example 2: Removing linear dependencies

Initial refinement

As a next example, the initial cubic standard T-spline mesh in Fig. B.1 is refined as shown in Fig. B.6(a).

Removing linear dependencies

The T-spline mesh in Fig. B.6(a) is non-standard exploiting Eq. (4.50). \underline{C}_e is not a square matrix and does not have full row rank in element f. The row echelon version of Eq. (4.43) yields the dependency $-2N^G(\underline{\xi}) + 2N^H(\underline{\xi}) + 3N^J(\underline{\xi}) = 0$ in element f. Therefore, an additional anchor needs to be inserted. This will be done in a manner similar to Section 4.5.4: extension lines (solid blue) are drawn between the anchors with locally linearly dependent blending functions G, H and J, Fig. B.6(b). The intersections of the extension lines are marked with the red squares. These squares denote possible positions for a new anchor if there does not already exist one. It can be observed from Fig. B.6(b) that only the intersection at the red square g is a candidate for a new anchor. However, the T-spline mesh with a new anchor in Fig. B.7 is still locally linearly dependent and semi-standard, so that more anchors and edges need to be inserted by applying the aforementioned methods until a standard and nested T-spline mesh is obtained.



(a) The T-spline mesh is locally linearly dependent – the row echelon form of Eq. (4.43) results in the dependency $-2N^G(\underline{\xi}) + 2N^H(\underline{\xi}) + 3N^J(\underline{\xi}) = 0$ in element f (dashed green).

(b) Illustration for the determination of the location of new anchors. Extension lines (solid blue) for the anchors with locally linearly dependent blending functions G , H and J are drawn. The extension lines intersect at the location of the red squares g , h and i . Only the square g represents a location for a new anchor (see Fig. B.7) since at h and i anchors are already located.

Fig. B.6: Refined (non-standard) cubic T-spline mesh of Fig. B.1(a) in the index domain.

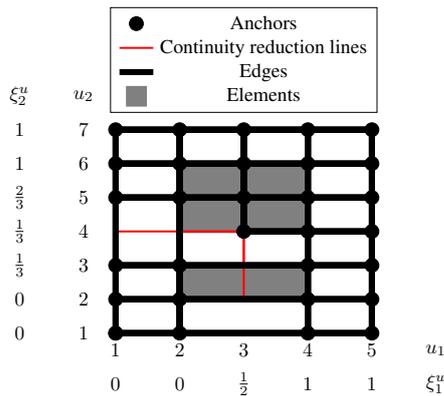


Fig. B.7: Refined cubic T-spline mesh of Fig. B.6(a). This T-spline mesh is semi-standard.

B.1.1.3 Example 3: Non-standard T-spline fulfils necessary condition for standard T-splines

In the examples considered so far, enforcing the BÉZIER extraction operator $\underline{\underline{C}}_e$ to be a square matrix resulted in a standard T-spline mesh. However, this is not always the case as presented in Fig. B.8(b).

Both T-spline meshes in Fig. B.8 are locally linearly independent with a square matrix $\underline{\underline{C}}_e$ for each element. The initial T-spline mesh, Fig. B.8(a), is standard, while the refined T-spline mesh, Fig. B.8(b), is non-standard. Recall, that a standard and a non-standard T-spline mesh cannot be nested according to Eq. (4.60). Evaluating the refinement matrix in Eq. (4.58) gives no solution for the anchor K in Fig. B.9(a). Applying the same

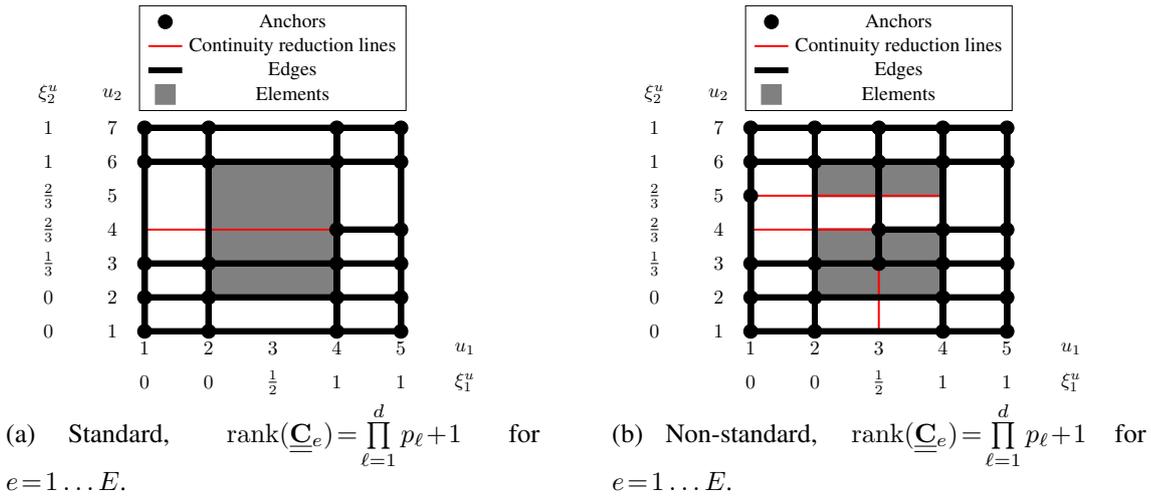


Fig. B.8: (a) Initial standard T-spline mesh and (b) refined non-standard T-spline mesh. The BÉZIER extraction operator $\underline{\underline{C}}_e$ is a square matrix of full rank in each element for both meshes, but the T-spline meshes are not nested as shown in Fig. B.9(a).

procedure as previously explained (see also Fig. B.4) gives the possible locations for new anchors (red) as depicted in Fig. B.9(a).

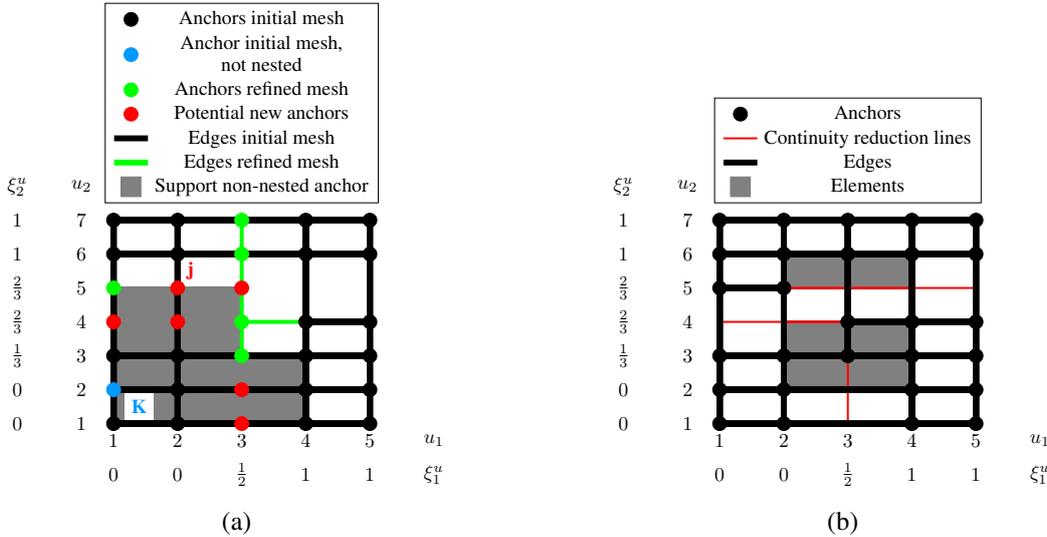


Fig. B.9: (a) Superposition of the initial T-spline mesh from Fig. B.8(a) and the newly inserted edges and anchors (green) from the refined T-spline mesh in Fig. B.8(b). The blending function of anchor K in the initial T-spline mesh cannot be represented as a linear combination of the blending functions of the refined T-spline mesh, i. e. both T-spline meshes are not nested since the row echelon form of Eq. (4.58) gives no result for anchor K. The support of anchor K – bounded by the new green edges and anchors – is depicted with a grey domain. Within the grey domain, new anchors can be inserted at the location of the red points in order to obtain a standard and nested T-spline mesh. For instance, inserting the red anchor j would result in a standard and nested T-spline mesh as presented in (b).

B.1.2 Local refinement of non-standard T-splines by adding anchors

In the previous examples in Section 4.5.4 and Appendix B.1.1, it was demonstrated how to refine a standard T-spline mesh and obtain a standard mesh based on Algorithm 4.1. This section gives an example that non-standard meshes can also be refined locally by adding anchors – the only requirement is that the initial and the refined mesh are nested. Fig. B.10 shows the initial non-standard and Fig. B.11 the refined semi-standard quadratic T-spline mesh in the index and physical domain, respectively. Both meshes are nested, which allows the calculation of the weighted control points \underline{P}_{wR} in Eq. (4.71).

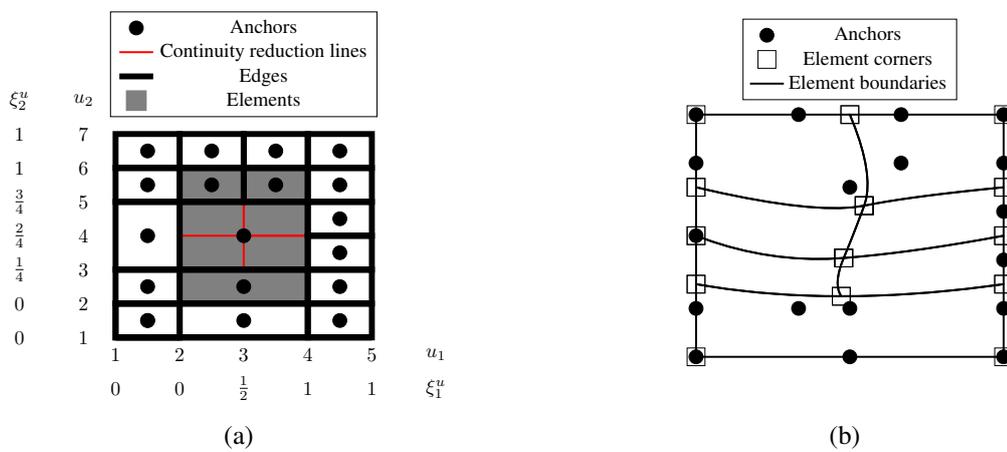


Fig. B.10: Initial (non-standard) quadratic T-spline mesh in (a) the index domain and (b) the physical domain.

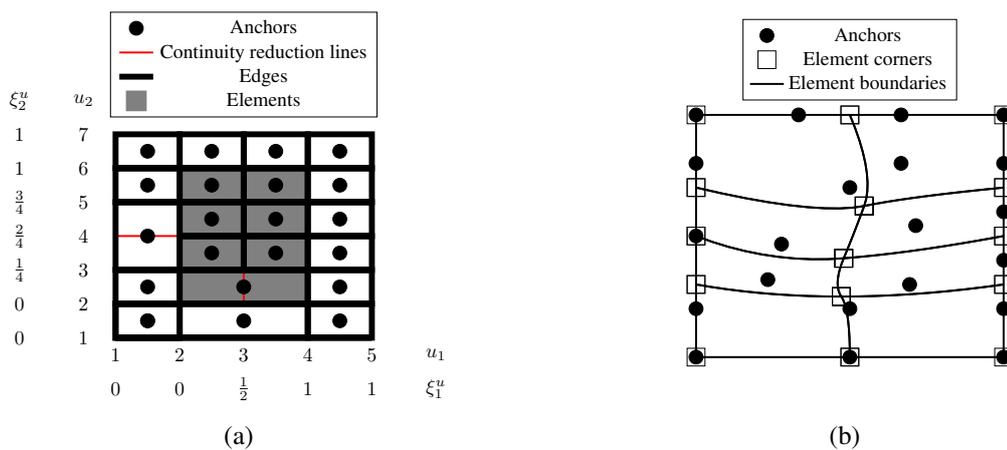


Fig. B.11: Refined (semi-standard) quadratic T-spline mesh of Fig. B.10 in (a) the index domain and (b) the physical domain.

B.1.3 Obtaining the optimised number of additionally inserted anchors

Consider the initial (standard) and refined (non-standard) T-spline mesh in Fig. B.12.

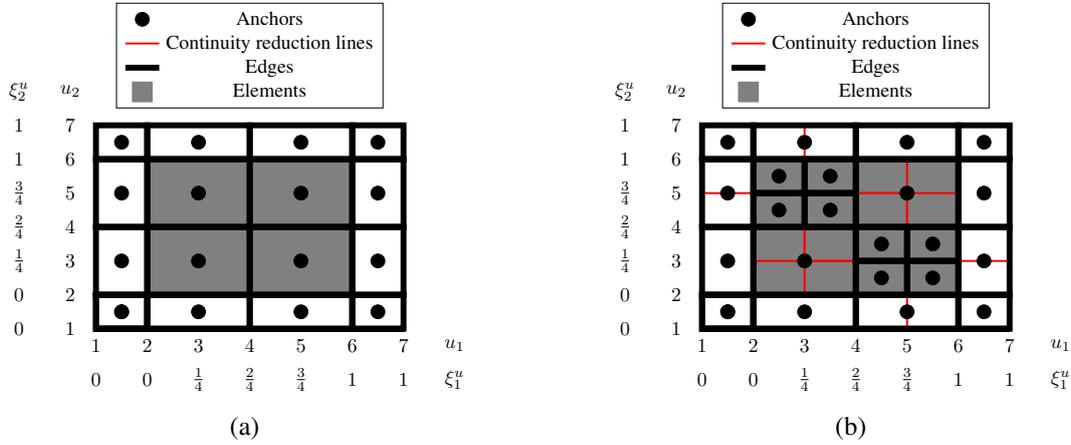


Fig. B.12: (a) Initial standard and (b) refined non-standard T-spline mesh in the index domain.

Fig. B.13 shows all the options where additional anchors can be inserted by applying the routines from Section 4.5.4, while due to symmetry, only the options for the lower part of Fig. B.12(b) are considered.

Table B.1 gives the number of pairs of anchors with linearly dependent blending functions, number of non-square matrices $\underline{\underline{C}}_e$, nestedness and number of additionally inserted anchors for the options in Fig. B.13.

Tab. B.1: Summary of the number of pairs of anchors with linearly dependent blending functions, number of non-square matrices $\underline{\underline{C}}_e$, nestedness and number of additionally inserted anchors for the options in Fig. B.13.

Figure	B.13(a)	B.13(b)	B.13(c)	B.13(d)	B.13(e)	B.13(f)	B.13(g)	B.13(h)	B.13(i)
Number of pairs of anchors with linearly dependent blending functions	5	5	6	6	5	6	5	6	6
Number of non-square matrices $\underline{\underline{C}}_e$	8	8	8	8	6	6	6	6	6
Nestedness	\times								
Number of additional anchors	2	2	2	2	3	3	3	3	4

According to Table B.1, the optimum option would be either Fig. B.13(e) or Fig. B.13(g) since they yield the smallest number of pairs of anchors with linearly dependent blending functions and number of non-square matrices $\underline{\underline{C}}_e$. After inserting the additional anchors, again, the rectangles to be subdivided are determined for the updated mesh and the optimum option is selected. This procedure needs to be repeated until a standard and nested

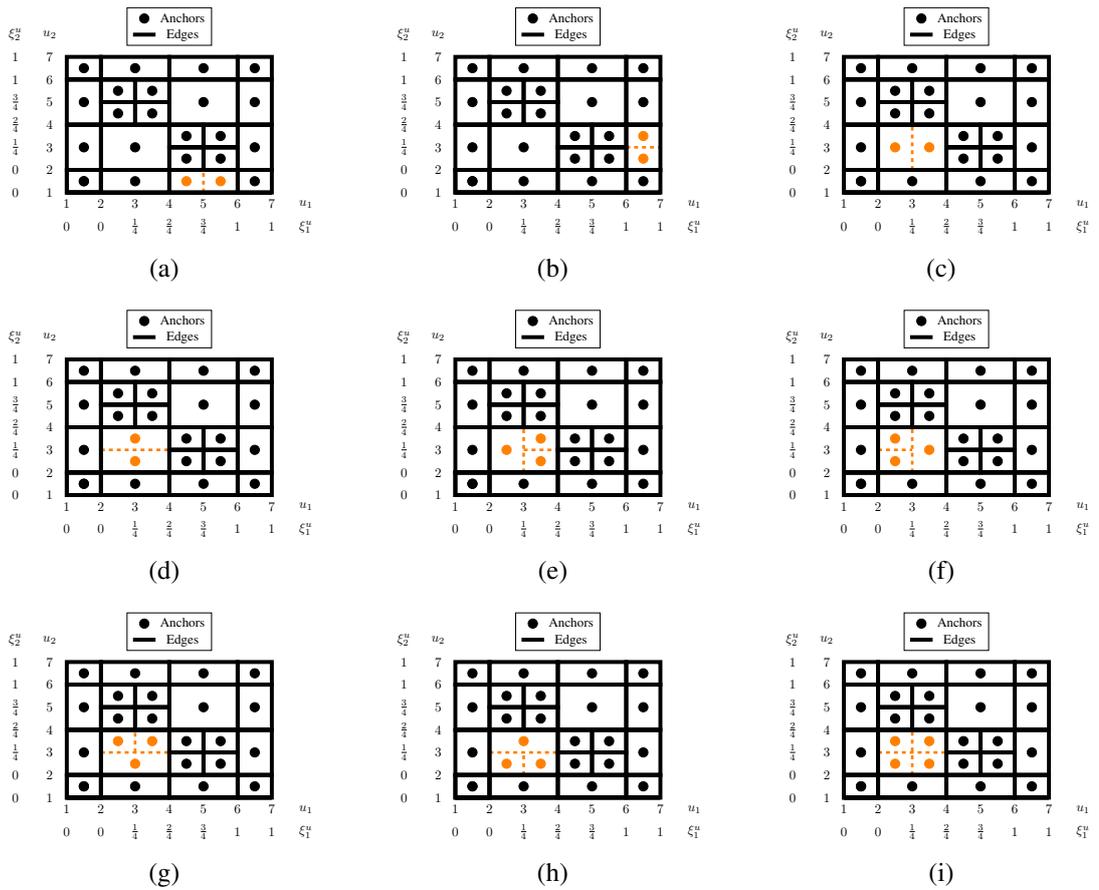


Fig. B.13: All possible subdivisions for Fig. B.12(b): the dashed orange lines indicate the new edges to be inserted, the orange points denote the locations of the new anchors.

T-spline mesh is obtained, see for example Fig. B.14 after six iterations.

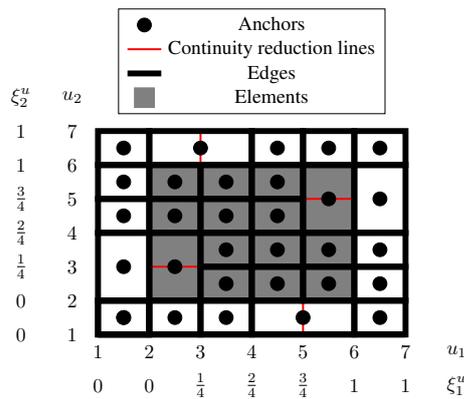


Fig. B.14: Standard and nested T-spline mesh after inserting additional anchors in six iterations into the refined T-spline mesh of Fig. B.12(b).

B.2 Modified local knot vectors for the hierarchical refinement

Table B.2 gives the local knot vector in element b and in the sub-elements b_l and b_r for each of the blue anchors in Fig. 4.21.

Tab. B.2: Local knot vectors Ξ_1 for the blue anchors in Fig. 4.21 in element b and in the sub-elements b_l and b_r .

Coordinates in index domain	Local knot vector	Local knot vector in b_l	Local knot vector in b_r
(1.5, 2.5)	{0, 0, 0, 1}	{0, 0, 0, $\frac{1}{4}$ }	{0, 0, $\frac{1}{4}$, 1}
(3, 2.5)	{0, 0, 1, 1}	{0, 0, $\frac{1}{4}$, 1}	{0, $\frac{1}{4}$, 1, 1}
(1.5, 4)	{0, 0, 0, 1}	{0, 0, 0, $\frac{1}{4}$ }	{0, 0, $\frac{1}{4}$, 1}
(3, 4)	{0, 0, 1, 1}	{0, 0, $\frac{1}{4}$, 1}	{0, $\frac{1}{4}$, 1, 1}
(4.5, 3.5)	{0, 1, 1, 1}	{0, $\frac{1}{4}$, 1, 1}	{ $\frac{1}{4}$, 1, 1, 1}
(4.5, 4.5)	{0, 1, 1, 1}	{0, $\frac{1}{4}$, 1, 1}	{ $\frac{1}{4}$, 1, 1, 1}
(1.5, 5.5)	{0, 0, 0, $\frac{1}{2}$ }	{0, 0, 0, $\frac{1}{4}$ }	{0, 0, $\frac{1}{4}$, $\frac{1}{2}$ }
(2.5, 5.5)	{0, 0, $\frac{1}{2}$, 1}	{0, 0, $\frac{1}{4}$, $\frac{1}{2}$ }	{0, $\frac{1}{4}$, $\frac{1}{2}$, 1}
(3.5, 5.5)	{0, $\frac{1}{2}$, 1, 1}	{0, $\frac{1}{4}$, $\frac{1}{2}$, 1}	{ $\frac{1}{4}$, $\frac{1}{2}$, 1, 1}

B.3 Unstructured cubic T-splines

This section gives an outline how an unstructured cubic T-spline mesh can be created that is \mathcal{C}_A^1 -continuous along spoke edges and between one- and two-ring neighbourhood elements, \mathcal{C}_A^2 -continuous everywhere else and that fulfils the partition of unity of the blending functions N .

B.3.1 The unstructured T-spline mesh

For cubic T-splines, the orange marked knot intervals in the neighbourhood of the light grey element h in Fig. B.15 are necessary for the determination of the BÉZIER extraction operator of each violet anchor with support in h . As for the quadratic case, some anchors with support in element h do not require all their individual knot intervals in order to determine their BÉZIER extraction operator in h .

This method is not applicable to the blue elements in the two-ring neighbourhood of an extraordinary point – these elements are irregular. Generalised BÉZIER extraction must be applied to these elements as explained for arbitrary knot intervals by SCOTT [95].

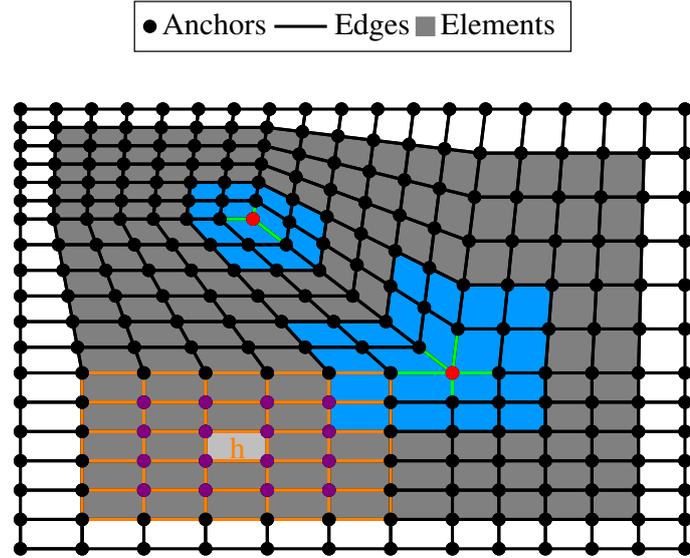


Fig. B.15: Example for an unstructured cubic T-spline mesh. All non-zero knot intervals are assumed to be uniform. The two extraordinary points of valence three and five are marked with red. Spoke edges (green) touch an extraordinary point. The purple anchors have a support in the light grey element h . In order to determine the BÉZIER extraction operator of these anchors in element h , the knot intervals of the two neighbouring rows of rectangles of element h are required (marked orange). Some anchors do not need all of their individual knot intervals. This construction cannot be applied to the two-ring neighbourhood elements (blue) of an extraordinary point. Generalised BÉZIER extraction must be applied to these irregular elements.

After generalised BÉZIER extraction, the continuity along the one- and two-ring neighbourhood elements is C_A^2 . C_A^0 -continuity exists along spoke edges and the T-spline mesh is non-standard – the partition of unity is not fulfilled. The BÉZIER extraction operators $\underline{\underline{C}}_e$ are non-square matrices for the elements in the one-ring neighbourhood of an extraordinary point. The blending functions N in the one-ring neighbourhood elements of an extraordinary point with valence three are locally linearly independent since $\underline{\underline{C}}_e$ has full row rank. For the one-ring neighbourhood elements of the extraordinary point with valence five, $\underline{\underline{C}}_e$ does not have full row rank. Hence, the blending functions are locally linearly dependent.

B.3.2 Modifying the BÉZIER extraction operator

Modifying the BÉZIER extraction operator for the cubic case is similar to the quadratic case in Section 4.7. The BÉZIER extraction operators $\underline{\underline{C}}_e$ in the one-ring neighbourhood elements are elevated to degree four according to

$$\underline{N}_e = \underline{\underline{C}}_e^3 \underline{B}_e^3 = \underline{\underline{C}}_e^3 \underline{\underline{E}}^{3,4} \underline{B}_e^4 = \underline{\underline{C}}_e^3 \underline{\underline{E}}^{3,4} \underline{B}_e^4 = \underline{\underline{C}}_e^4 \underline{B}_e^4. \quad (\text{B.1})$$

For the cubic case, only the coefficients $c_{\alpha,\beta}$ of the blending functions that are non-zero in *at least two* one-ring neighbourhood elements are modified. Assuming that $a = 1 \dots A$ blending functions have a support in *at least two* of $b = 1 \dots B$ one-ring neighbourhood elements, the fairing equations for the constrained linear least square problem in Eq. (4.91) read as follows

$$\begin{aligned} \tilde{c}_{\alpha,\beta}^{a,b} - \tilde{c}_{\alpha,\beta+1}^{a,b} &= c_{\alpha,\beta}^{a,b} - c_{\alpha,\beta+1}^{a,b} \quad \text{for } 1 \leq \alpha \leq 5, 1 \leq \beta \leq 4, 1 \leq a \leq A, 1 \leq b \leq B, \\ \tilde{c}_{\alpha,\beta}^{a,b} - \tilde{c}_{\alpha+1,\beta}^{a,b} &= c_{\alpha,\beta}^{a,b} - c_{\alpha+1,\beta}^{a,b} \quad \text{for } 1 \leq \alpha \leq 4, 1 \leq \beta \leq 5, 1 \leq a \leq A, 1 \leq b \leq B. \end{aligned} \quad (\text{B.2})$$

When perturbing the BÉZIER coefficients $c_{\alpha,\beta}$, C_A^1 -continuity will be enforced along one- and two-ring neighbourhood elements and along spoke edges. The following constraints need to be assembled in $\underline{\mathbf{G}}$ and $\underline{\mathbf{g}}$ for blending functions that are non-zero in *at least two* one-ring neighbourhood elements: Eqs. (4.93) and (4.105) – (4.110). Furthermore, for blending functions that have support in *only two* one-ring neighbourhood elements, the constraint

$$\tilde{c}_{\alpha,\beta}^{a,b} = 0 \quad \text{for } 1 \leq \alpha \leq 3, 1 \leq \beta \leq 3 \quad (\text{B.3})$$

is enforced in one-ring neighbourhood elements where they do not have a support. In order to fulfil the partition of unity,

$$\sum_{a=1}^A \tilde{c}_{\alpha,\beta}^{a,b} = 1 \quad \begin{cases} \text{for } 1 \leq \alpha \leq 3, 1 \leq \beta \leq 3 \\ \text{in all one-ring neighbourhood elements } b = 1 \dots B \end{cases} \quad (\text{B.4})$$

needs to be taken into account. Computing the term in Eq. (4.112) after solving Eq. (4.91) yields zero within machine precision for the unstructured cubic T-spline mesh, i. e. all imposed constraints are satisfied.

For the cubic case, no change in support of an anchor is observed for the valences three and five after modifying the BÉZIER coefficients $c_{\alpha,\beta}$. The BÉZIER extraction operator $\underline{\mathbf{C}}_e$ for the elements in the one-ring neighbourhood of an extraordinary point is still a non-square matrix. Thus, hierarchical refinement (Section 4.6) and BÉZIER projection (THOMAS *et al.* [110]) cannot be applied to these elements. The BÉZIER extraction operator $\underline{\mathbf{C}}_e$ has full row rank in all elements of Fig. B.15 after solving Eq. (4.91). Hence, the blending functions N are locally linearly independent. Recall, that the blending functions N are locally linearly dependent in the one-ring neighbourhood elements of an extraordinary point with valence five for the quadratic case.

Fig. B.16 shows the contour plot for a blending function N and its derivatives $N_{,x}$ and $N_{,y}$ before and after smoothing. The blending function corresponds to an anchor that is equivalent to the extraordinary point of valence five.

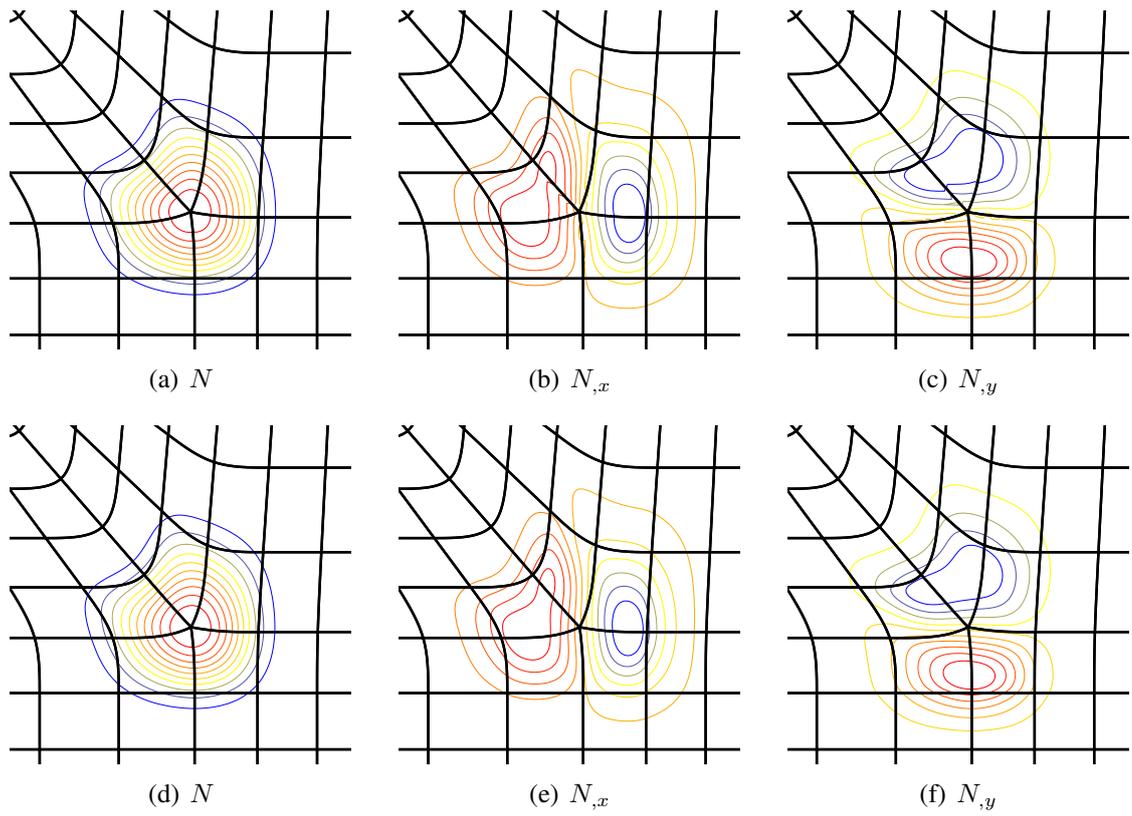


Fig. B.16: Contour plots of a cubic blending function N and its derivatives $N_{,x}$, $N_{,y}$ in the physical domain before (a)-(c) and after (d)-(f) smoothing. The blending function corresponds to an anchor that is equivalent to the extraordinary point of valence five.

Bibliography

- [1] Alfeld, P.; Sorokina, T.: *Two tetrahedral C^1 cubic macro elements*; Journal of Approximation Theory; **157 (1)**; 53–69; 2009
- [2] Ambrosio, L.; Tortorelli, V. M.: *Approximation of functional depending on jumps by elliptic functional via Γ -convergence*; Communications on Pure and Applied Mathematics; **43 (8)**; 999–1036; 1990
- [3] Amor, H.; Marigo, J.-J.; Maurini, C.: *Regularized formulation of the variational brittle fracture with unilateral contact: Numerical experiments*; Journal of the Mechanics and Physics of Solids; **57 (8)**; 1209–1229; 2009
- [4] Anderson, T. L.: *Fracture Mechanics: Fundamentals and Applications*; CRC press; 2005
- [5] Barenblatt, G. I.: *The Mathematical Theory of Equilibrium Cracks in Brittle Fracture*; Advances in Applied Mechanics; **7**; 55–129; 1962
- [6] Barsky, B. A.; DeRose, T. D.: *Geometric Continuity of Parametric Curves*; Technical Report UCB/CSD 84/205; University of California at Berkeley; 1984
- [7] Barsky, B. A.; DeRose, T. D.: *Geometric continuity of parametric curves: three equivalent characterizations*; IEEE Computer Graphics and Applications; **9 (6)**; 60–68; 1989
- [8] Bažant, Z. P.; Pijaudier-Cabot, G.: *Nonlocal Continuum Damage, Localization Instability and Convergence*; Journal of Applied Mechanics; **55 (2)**; 287–293; 1988
- [9] Bazilevs, Y.; Calo, V. M.; Cottrell, J. A.; Evans, J. A.; Hughes, T. J. R.; Lipton, S.; Scott, M. A.; Sederberg, T. W.: *Isogeometric analysis using T-splines*; Computer Methods in Applied Mechanics and Engineering; **199 (5–8)**; 229–263; 2010
- [10] Bazilevs, Y.; Hsu, M. C.; Scott, M. A.: *Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with applications to wind turbines*; Computer Methods in Applied Mechanics and Engineering; **249–252**; 28–41; 2012
- [11] Bellettini, G.; Coscia, A.: *Discrete approximation of a free discontinuity problem*; Numerical Functional Analysis and Optimization; **15 (3–4)**; 201–224; 1994

- [12] Björck, Å.: *Numerical Methods for Least Squares Problems*; Siam; 1996
- [13] Borden, M. J.; Scott, M. A.; Evans, J. A.; Hughes, T. J. R.: *Isogeometric finite element data structures based on Bézier extraction of NURBS*; International Journal for Numerical Methods in Engineering; **87 (1–5)**; 15–47; 2011
- [14] Borden, M. J.; Verhoosel, C. V.; Scott, M. A.; Hughes, T. J. R.; Landis, C. M.: *A phase-field description of dynamic brittle fracture*; Computer Methods in Applied Mechanics and Engineering; **217–220**; 77–95; 2012
- [15] Borden, M. J.: *Isogeometric Analysis of Phase-Field Models for Dynamic Brittle and Ductile Fracture*; Ph.D. thesis; The University of Texas at Austin; 2012
- [16] Borden, M. J.; Hughes, T. J. R.; Landis, C. M.; Verhoosel, C. V.: *A higher-order phase-field model for brittle fracture: Formulation and analysis within the isogeometric analysis framework*; Computer Methods in Applied Mechanics and Engineering; **273**; 100–118; 2014
- [17] Bourdin, B.; Francfort, G. A.; Marigo, J.-J.: *Numerical experiments in revisited brittle fracture*; Journal of the Mechanics and Physics of Solids; **48 (4)**; 797–826; 2000
- [18] Bourdin, B.: *Numerical implementation of the variational formulation for quasi-static brittle fracture*; Interfaces and Free Boundaries; **9 (3)**; 411–430; 2007
- [19] Bourdin, B.; Francfort, G. A.; Marigo, J.-J.: *The variational approach to fracture*; Journal of elasticity; **91 (1–3)**; 5–148; 2008
- [20] Buffa, A.; Cho, D.; Sangalli, G.: *Linear independence of the T-spline blending functions associated with some particular T-meshes*; Computer Methods in Applied Mechanics and Engineering; **199 (23–24)**; 1437–1445; 2010
- [21] Casquero, H.; Liu, L.; Bona-Casas, C.; Zhang, Y.; Gómez, H.: *A hybrid variational-collocation immersed method for fluid-structure interaction using unstructured T-splines*; International Journal for Numerical Methods in Engineering; **105 (11)**; 855–880; 2016
- [22] Chambolle, A.: *An approximation result for special functions with bounded deformation*; Journal de Mathématiques Pures et Appliquées; **83 (7)**; 929–954; 2004
- [23] Chui, C. K.; Lai, M.-J.: *Multivariate vertex splines and finite elements*; Journal of Approximation Theory; **60 (3)**; 245–343; 1990

-
- [24] Cirak, F.; Ortiz, M.; Schröder, P.: *Subdivision surfaces: a new paradigm for thin-shell finite-element analysis*; International Journal for Numerical Methods in Engineering; **47 (12)**; 2039–2072; 2000
- [25] Cox, M. G.: *The numerical evaluation of B-Splines*; IMA Journal of Applied Mathematics; **10 (2)**; 134–149; 1972
- [26] Crisfield, M. A.: *Local instabilities in the non-linear analysis of reinforced concrete beams and slabs*; Institution of Civil Engineers, Proceedings; **73 (1)**; 135–145; 1982
- [27] de Boor, C.: *On calculating with B-splines*; Journal of Approximation Theory; **6 (1)**; 50–62; 1972
- [28] de Borst, R.: *Computation of post-bifurcation and post-failure behavior of strain-softening solids*; Computers & Structures; **25 (2)**; 211–224; 1987
- [29] de Borst, R.; Crisfield, M. A.; Remmers, J. J. C.; Verhoosel, C. V.: *Nonlinear Finite Element Analysis of Solids and Structures*; John Wiley & Sons; 2012
- [30] de Vree, J. H. P.; Brekelmans, W. A. M.; Van Gils, M. A. J.: *Comparison of nonlocal approaches in continuum damage mechanics*; Computers & Structures; **55 (4)**; 581–588; 1995
- [31] Del Piero, G.; Lancioni, G.; March, R.: *A variational model for fracture mechanics: numerical experiments*; Journal of the Mechanics and Physics of Solids; **55 (12)**; 2513–2537; 2007
- [32] Dierckx, P.; van Leemput, S.; Vermeire, T.: *Algorithms for surface fitting using Powell-Sabin splines*; IMA Journal of Numerical Analysis; **12 (2)**; 271–299; 1992
- [33] Dierckx, P.: *On calculating normalized Powell-Sabin B-splines*; Computer Aided Geometric Design; **15 (1)**; 61–78; 1997
- [34] Dimitri, R.; De Lorenzis, L.; Scott, M. A.; Wriggers, P.; Taylor, R. L.; Zavarise, G.: *Isogeometric large deformation frictionless contact using T-splines*; Computer Methods in Applied Mechanics and Engineering; **269**; 394–414; 2014
- [35] Dimitri, R.: *Isogeometric treatment of large deformation contact and debonding problems with T-splines: a review*; Curved and Layered Structures; **2 (1)**; 59–90; 2015
-

- [36] Dörffel, M. R.; Jüttler, B.; Simeon, B.: *Adaptive isogeometric analysis by local h-refinement with T-splines*; Computer Methods in Applied Mechanics and Engineering; **199** (5–8); 264–275; 2010
- [37] Du, W.-H.; Schmitt, F. J. M.: *On the G^1 continuity of piecewise Bézier surfaces: a review with new results*; Computer-Aided Design; **22** (9); 556–573; 1990
- [38] Dugdale, D. S.: *Yielding of steel sheets containing slits*; Journal of the Mechanics and Physics of Solids; **8** (2); 100–104; 1960
- [39] Farouki, R. T.; Neff, C. A.: *On the numerical condition of Bernstein-Bézier subdivision processes*; Mathematics of Computation; **55** (192); 637–647; 1990
- [40] Faux, I. D.; Pratt, M. J.: *Computational Geometry for Design and Manufacture*; Ellis Horwood Ltd; 1979
- [41] Felippa, C. A.: *Introduction to Finite Element Methods*; Course Notes, Department of Aerospace Engineering Sciences, University of Colorado at Boulder; 2004
- [42] Fish, J.; Belytschko, T.: *A First Course in Finite Elements*; John Wiley & Sons; 2007
- [43] Francfort, G. A.; Marigo, J.-J.: *Revisiting brittle fracture as an energy minimization problem*; Journal of the Mechanics and Physics of Solids; **46** (8); 1319–1342; 1998
- [44] Geers, M. G. D.; de Borst, R.; Brekelmans, W. A. M.; Peerlings, R. H. J.: *Strain-based transient-gradient damage model for failure analyses*; Computer Methods in Applied Mechanics and Engineering; **160** (1–2); 133–153; 1998
- [45] Griffith, A. A.: *The phenomena of Rupture and Flow in Solids*; Philosophical Transactions of the Royal Society of London. Series A, containing Papers of a Mathematical or Physical Character; **221**; 163–198; 1921
- [46] Groisser, D.; Peters, J.: *Matched G^k -constructions always yield C^k -continuous isogeometric elements*; Computer Aided Geometric Design; **34**; 67–72; 2015
- [47] Gross, D.; Hauger, W.; Wriggers, P.: *Technische Mechanik: Band 4: Hydromechanik, Elemente der Höheren Mechanik, Numerische Methoden*; Springer; 2009
- [48] Guo, Y.; Ruess, M.: *Nitsche's method for a coupling of isogeometric thin shells and blended shell structures*; Computer Methods in Applied Mechanics and Engineering; **284**; 881–905; 2015

-
- [49] Gutiérrez, M. A.: *Energy release control for numerical simulations of failure in quasi-brittle solids*; Communications in Numerical Methods in Engineering; **20** (1); 19–29; 2004
- [50] Hillerborg, A.; Modéer, M.; Petersson, P.-E.: *Analysis of crack formation and crack growth in concrete by means of fracture mechanics and finite elements*; Cement and Concrete Research; **6** (6); 773–781; 1976
- [51] Hsu, M.-C.; Kamensky, D.; Xu, F.; Kiendl, J.; Wang, C.; Wu, M. C. H.; Mineroff, J.; Reali, A.; Bazilevs, Y.; Sacks, M. S.: *Dynamic and fluid–structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models*; Computational Mechanics; **55** (6); 1211–1225; 2015
- [52] Hughes, T. J. R.; Cottrell, J. A.; Bazilevs, Y.: *Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement*; Computer Methods in Applied Mechanics and Engineering; **194** (39–41); 4135–4195; 2005
- [53] Irwin, G. R.: *Fracture Dynamics*; Fracturing of Metals, American Society for Metals, Cleveland, OH; 147–166; 1948
- [54] Jirásek, M.; Bažant, Z. P.: *Inelastic Analysis of Structures*; John Wiley & Sons; 2001
- [55] Kagan, P.; Fischer, A.; Bar-Yoseph, P. Z.: *New B-Spline Finite Element approach for geometrical design and mechanical analysis*; International Journal for Numerical Methods in Engineering; **41** (3); 435–458; 1998
- [56] Karush, W.: *Minima of functions of several variables with inequalities as side constraints*; Master’s thesis; University of Chicago; 1939
- [57] Kiendl, J.; Bletzinger, K.-U.; Linhard, J.; Wüchner, R.: *Isogeometric shell analysis with Kirchhoff-Love elements*; Computer Methods in Applied Mechanics and Engineering; **198** (49–52); 3902–3914; 2009
- [58] Kiendl, J.; Bazilevs, Y.; Hsu, M.-C.; Wüchner, R.; Bletzinger, K.-U.: *The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches*; Computer Methods in Applied Mechanics and Engineering; **199** (37–40); 2403–2416; 2010
- [59] Kiendl, J.; Hsu, M.-C.; Wu, M. C. H.; Reali, A.: *Isogeometric Kirchhoff–Love shell formulations for general hyperelastic materials*; Computer Methods in Applied Mechanics and Engineering; **291**; 280–303; 2015
-

- [60] Kirchhoff, G.: *Über das Gleichgewicht und die Bewegung einer elastischen Scheibe.*; Journal für die reine und angewandte Mathematik; **40**; 51–88; 1850
- [61] Kostas, K. V.; Ginnis, A. I.; Politis, C. G.; Kaklis, P. D.: *Ship-hull shape optimization with a T-spline based BEM–isogeometric solver*; Computer Methods in Applied Mechanics and Engineering; **284**; 611–622; 2015
- [62] Kuhl, E.: *Numerische Modelle für kohäsive Reibungsmaterialien*; Ph.D. thesis; Universität Stuttgart; 2010
- [63] Kuhn, C.; Müller, R.: *A continuum phase field model for fracture*; Engineering Fracture Mechanics; **77 (18)**; 3625–3634; 2010
- [64] Kuhn, H. W.; Tucker, A. W.: *Nonlinear Programming*; In *Second Berkeley Symposium on Mathematical Statistics and Probability*; 481–492; 1951
- [65] Lancioni, G.; Royer-Carfagni, G.: *The Variational Approach to Fracture Mechanics. A Practical Application to the French Panthéon in Paris*; Journal of Elasticity; **95 (1–2)**; 1–30; 2009
- [66] Li, X.; Zheng, J.; Sederberg, T. W.; Hughes, T. J. R.; Scott, M. A.: *On linear independence of T-spline blending functions*; Computer Aided Geometric Design; **29 (1)**; 63–76; 2012
- [67] Li, X.: *Some properties for analysis-suitable T-splines*; Journal of Computational Mathematics; **33 (4)**; 428–442; 2015
- [68] Lipton, S.; Evans, J. A.; Bazilevs, Y.; Elguedj, T.; Hughes, T. J. R.: *Robustness of isogeometric structural discretizations under severe mesh distortion*; Computer Methods in Applied Mechanics and Engineering; **199 (5–8)**; 357–373; 2010
- [69] Lorentz, E.; Badel, P.: *A new path-following constraint for strain-softening finite element simulations*; International Journal for Numerical Methods in Engineering; **60 (2)**; 499–526; 2004
- [70] Love, A. E. H.: *The small free vibrations and deformation of a thin elastic shell*; Philosophical Transactions of the Royal Society of London. A; **179**; 491–546; 1888
- [71] May, S.; Vignollet, J.; de Borst, R.: *A numerical assessment of phase-field models for brittle and cohesive fracture: Γ -Convergence and stress oscillations*; European Journal of Mechanics - A/Solids; **52**; 72–84; 2015

-
- [72] May, S.; Vignollet, J.; de Borst, R.: *The role of the Bézier extraction operator for T-splines of arbitrary degree: linear dependencies, partition of unity property, nesting behaviour and local refinement*; International Journal for Numerical Methods in Engineering; **103 (8)**; 547–581; 2015
- [73] May, S.; Vignollet, J.; de Borst, R.: *A new arc-length control method based on the rates of the internal and the dissipated energy*; Engineering Computations; **33 (1)**; 100–115; 2016
- [74] May, S.; Vignollet, J.; de Borst, R.: *Powell-Sabin B-splines and unstructured standard T-splines for the solution of the Kirchhoff-Love plate theory exploiting Bézier extraction*; International Journal for Numerical Methods in Engineering; accepted; 2016
- [75] May, S.; de Borst, R.; Vignollet, J.: *Powell-Sabin B-splines for smeared and discrete approaches to fracture in quasi-brittle materials*; Computer Methods in Applied Mechanics and Engineering; accepted; 2016
- [76] Miehe, C.; Hofacker, M.; Welschinger, F.: *A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits*; Computer Methods in Applied Mechanics and Engineering; **199 (45–48)**; 2765–2778; 2010
- [77] Miehe, C.; Welschinger, F.; Hofacker, M.: *Thermodynamically consistent phase-field models of fracture: Variational principles and multi-field FE implementations*; International Journal for Numerical Methods in Engineering; **83 (10)**; 1273–1311; 2010
- [78] Mumford, D.; Shah, J.: *Optimal approximations by piecewise smooth functions and associated variational problems*; Communications on Pure and Applied Mathematics; **42 (5)**; 577–685; 1989
- [79] Nguyen, T.; Karčiauskas, K.; Peters, J.: *A Comparative Study of Several Classical, Discrete Differential and Isogeometric Methods for Solving Poisson's Equation on the Disk*; Axioms; **3 (2)**; 280–299; 2014
- [80] Nitsche, J.: *Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind*; Abhandlungen aus dem mathematischen Seminar der Universität Hamburg; **36 (1)**; 9–15; 1971
-

- [81] Orowan, E.: *Fracture and Strength of Solids*; Reports on Progress in Physics; **12**; 185–232; 1949
- [82] Peerlings, R. H. J.; de Borst, R.; Brekelmans, W. A. M.; de Vree, J. H. P.: *Gradient enhanced damage for quasi-brittle materials*; International Journal for Numerical Methods in Engineering; **39 (19)**; 3391–3403; 1996
- [83] Persson, P.-O.; Strang, G.: *A Simple Mesh Generator in MATLAB*; SIAM Review; **46 (2)**; 329–345; 2004
- [84] Peters, J.: *Geometric continuity*; Handbook of Computer Aided Geometric Design; 193–229; 2002
- [85] Piegl, L.; Tiller, W.: *A menagerie of rational B-spline circles*; IEEE Computer Graphics and Applications; **9 (5)**; 48–56; 1989
- [86] Piegl, L.; Tiller, W.: *The NURBS Book*; Springer; 1996
- [87] Pijaudier-Cabot, G.; Bažant, Z. P.: *Nonlocal Damage Theory*; Journal of Engineering Mechanics; **113 (10)**; 1512–1533; 1987
- [88] Powell, M. J. D.; Sabin, M. A.: *Piecewise Quadratic Approximations on Triangles*; ACM Transactions on Mathematical Software; **3 (4)**; 316–325; 1977
- [89] Reddy, J. N.: *Energy Principles and Variational Methods in Applied Mechanics*; John Wiley & Sons; 2002
- [90] Riks, E.: *An incremental approach to the solution of snapping and buckling problems*; International Journal of Solids and Structures; **15 (7)**; 529–551; 1979
- [91] Schellekens, J. C. J.; de Borst, R.: *On the numerical integration of interface elements*; International Journal for Numerical Methods in Engineering; **36 (1)**; 43–66; 1993
- [92] Schlangen, E.: *Experimental and Numerical Analysis of Fracture Processes in Concrete*; Ph.D. thesis; Delft University of Technology; 1993
- [93] Schmidt, R.; Kiendl, J.; Bletzinger, K.-U.; Wüchner, R.: *Realization of an integrated structural design process: analysis-suitable geometric modelling and isogeometric analysis*; Computing and Visualization in Science; **13 (7)**; 315–330; 2010

-
- [94] Scott, M. A.; Borden, M. J.; Verhoosel, C. V.; Sederberg, T. W.; Hughes, T. J. R.: *Isogeometric finite element data structures based on Bézier extraction of T-splines*; International Journal for Numerical Methods in Engineering; **88** (2); 126–156; 2011
- [95] Scott, M. A.: *T-splines as a Design-Through-Analysis Technology*; Ph.D. thesis; The University of Texas at Austin; 2011
- [96] Scott, M. A.; Li, X.; Sederberg, T. W.; Hughes, T. J. R.: *Local refinement of analysis-suitable T-splines*; Computer Methods in Applied Mechanics and Engineering; **213–216**; 206–222; 2012
- [97] Scott, M. A.; Simpson, R. N.; Evans, J. A.; Lipton, S.; Bordas, S. P. A.; Hughes, T. J. R.; Sederberg, T. W.: *Isogeometric boundary element analysis using unstructured T-splines*; Computer Methods in Applied Mechanics and Engineering; **254**; 197–221; 2013
- [98] Scott, M. A.; Hughes, T. J. R.; Sederberg, T. W.; Sederberg, M. T.: *An integrated approach to engineering design and analysis using the Autodesk T-spline plugin for Rhino3d*; ICES Report 14-33; The Institute for Computational Engineering and Sciences, The University of Texas at Austin; 2014
- [99] Sederberg, T. W.; Zheng, J.; Bakenov, A.; Nasri, A.: *T-splines and T-NURCCs*; ACM Transactions on Graphics; **22** (3); 477–484; 2003
- [100] Sederberg, T. W.; Cardon, D. L.; Finnigan, G. T.; North, N. S.; Zheng, J.; Lyche, T.: *T-spline Simplification and Local Refinement*; ACM Transactions on Graphics; **23** (3); 276–283; 2004
- [101] Sherman, J.; Morrison, W. J.: *Adjustment of an inverse matrix corresponding to a change in one element of a given matrix*; The Annals of Mathematical Statistics; **21** (1); 124–127; 1950
- [102] Simo, J. C.; Hughes, T. J. R.: *Computational Inelasticity*; Springer; 1998
- [103] Simpson, R. N.; Scott, M. A.; Taus, M.; Thomas, D. C.; Lian, H.: *Acoustic isogeometric boundary element analysis*; Computer Methods in Applied Mechanics and Engineering; **269**; 265–290; 2014
- [104] Speleers, H.; Dierckx, P.; Vandewalle, S.: *Numerical solution of partial differential equations with Powell-Sabin splines*; Journal of Computational and Applied Mathematics; **189** (1–2); 643–659; 2006
-

- [105] Speleers, H.; Manni, C.; Pelosi, F.; Sampoli, M. L.: *Isogeometric analysis with Powell–Sabin splines for advection–diffusion–reaction problems*; Computer Methods in Applied Mechanics and Engineering; **221–222**; 132–148; 2012
- [106] Speleers, H.; Manni, C.; Pelosi, F.: *From NURBS to NURPS geometries*; Computer Methods in Applied Mechanics and Engineering; **255**; 238–254; 2013
- [107] Stogner, R. H.; Carey, G. F.: *C^1 macroelements in adaptive finite element methods*; International Journal for Numerical Methods in Engineering; **70 (9)**; 1076–1095; 2007
- [108] Swenson, D. V.; Ingraffea, A. R.: *Modeling mixed-mode dynamic crack propagation using finite elements: Theory and applications*; Computational Mechanics; **3 (6)**; 381–397; 1988
- [109] Tagliabue, A.; Dedè, L.; Quarteroni, A.: *Isogeometric analysis and error estimates for high order partial differential equations in fluid dynamics*; Computers & Fluids; **102**; 277–303; 2014
- [110] Thomas, D. C.; Scott, M. A.; Evans, J. A.; Tew, K.; Evans, E. J.: *Bézier projection: A unified approach for local projection and quadrature-free refinement and coarsening of NURBS and T-splines with particular application to isogeometric design and analysis*; Computer Methods in Applied Mechanics and Engineering; **284**; 55–105; 2015
- [111] Thomas, D. C.; Scott, M. A.: *Isogeometric analysis based on T-splines*; Isogeometric Methods for Numerical Simulation; 205–232; 2015
- [112] Timoshenko, S. P.; Woinowsky-Krieger, S.: *Theory of plates and shells*; McGraw-Hill; 1959
- [113] Vanraes, E.; Dierckx, P.; Bultheel, A.: *On the choice of the PS-triangles*; Technical Report 353; Department of Computer Science, K.U. Leuven; 2003
- [114] Verhoosel, C. V.; Remmers, J. J. C.; Gutiérrez, M. A.: *A dissipation-based arc-length method for robust simulation of brittle and ductile failure*; International Journal for Numerical Methods in Engineering; **77 (9)**; 1290–1321; 2009
- [115] Verhoosel, C. V.; Scott, M. A.; Hughes, T. J. R.; de Borst, R.: *An isogeometric analysis approach to gradient damage models*; International Journal for Numerical Methods in Engineering; **86 (1)**; 115–134; 2011

- [116] Verhoosel, C. V.; Scott, M. A.; de Borst, R.; Hughes, T. J. R.: *An isogeometric approach to cohesive zone modeling*; International Journal for Numerical Methods in Engineering; **87 (1–5)**; 336–360; 2011
- [117] Verhoosel, C. V.; de Borst, R.: *A phase-field model for cohesive fracture*; International Journal for Numerical Methods in Engineering; **96 (1)**; 43–62; 2013
- [118] Vignollet, J.; May, S.; de Borst, R.; Verhoosel, C. V.: *Phase-field models for brittle and cohesive fracture*; Meccanica; **49 (11)**; 2587–2601; 2014
- [119] Vignollet, J.; May, S.; de Borst, R.: *On the numerical integration of isogeometric interface elements*; International Journal for Numerical Methods in Engineering; **102 (11)**; 1733–1749; 2015
- [120] Wang, W.; Zhang, Y.; Xu, G.; Hughes, T. J. R.: *Converting an unstructured quadrilateral/hexahedral mesh to a rational T-spline*; Computational Mechanics; **50 (1)**; 65–84; 2012
- [121] Wawrzynek, P. A.; Ingraffea, A. R.: *Interactive finite element analysis of fracture processes: an integrated approach*; Theoretical and Applied Fracture Mechanics; **8 (2)**; 137–150; 1987
- [122] Wells, G. N.; Sluys, L. J.: *A new method for modelling cohesive cracks using finite elements*; International Journal for Numerical Methods in Engineering; **50 (12)**; 2667–2682; 2001
- [123] Wempner, G. A.: *Discrete approximations related to nonlinear theories of solids*; International Journal of Solids and Structures; **7 (11)**; 1581–1599; 1971
- [124] Worsey, A. J.; Piper, B.: *A trivariate Powell-Sabin interpolant*; Computer Aided Geometric Design; **5 (3)**; 177–186; 1988