Technische Universität Graz
Institut für Festigkeitslehre
Kopernikusgasse 24/I
8010 Graz

Matthias Rambausek, BSc

An Energy-Based r-Adaptive Finite Element Method Applied to Hyperelastostatics

Masterarbeit
zur Erlangung des akademischen Grades eines
Diplom–Ingenieurs

Studienrichtung: Maschinenbau

Technische Universität Graz
Fakultät für Maschinenbau und Wirtschaftswissenschaften

Betreuer: Dipl.-Ing. Dr.techn. Michael Hammer
Beurteiler: O.Univ.–Prof. Dipl.–Ing. Dr.techn. Christian C. Celigoj

Graz, 2013

Abstract In Finite Element Analysis (FEA) the discretisation (the mesh and the shape functions' order) has great influence on the quality of analysis. In r-adaptive FEA it is tried to improve the FE solution by finding the optimal mesh without changing the mesh topology and the element order. This means, the mesh's nodes are moved to improve the overall result. The energy-based r-adaptivity approach is driven by energy minimisation, which also is the governing principle in hyperelastostatics. Therefore, a variational Arbitrary Lagrangian Eulerian (ALE) formulation is used to obtain the equilibrium equations for spatial and material motion. The FE solution of both equilibrium equations yields an optimised mesh and thus better results concerning energy minimisation. This reduces the overall error, compared to the FE solution of the pure spatial equilibrium on the initial mesh.

Kurzfassung Die Diskretisierung (das Rechennetz und die Ordnung der Formfunktionen) hat großen Einfluss auf die Qualität der Lösung von Finite Elemente (FE) Simulationen. Bei der r-adaptiven FE-Methode wird zur Verbesserung der FE-Lösung versucht, das optimale Netz unter Beibehaltung der Netztopologie und der Element-Ordnung für die gegebene Aufgabenstellung zu finden. Energie-basierte r-Adaptivität folgt dem Prinzip der Energieminimierung, das auch in der statischen Berechnung bei hyperelastischen Materialien angewandt wird. Die Umsetzung erfolgt über einen speziellen Variationsansatz, bei dem sowohl die räumlichen Verschiebungen des verformten Körpers, als auch die Lage der Knoten des Rechennetzes variiert werden. Dieser als "variational ALE" Formulierung bekannte Ansatz wird zur Formulierung der räumlichen und materiellen Gleichgewichte verwendet. Die FE-Lösung dieser beiden Gleichgewichte ergibt ein optimiertes Netz und damit bessere Resultate gemessen an der erreichten Energiereduktion. Dadurch wiederum wird der globale Fehler verglichen mit der FE-Lösung allein für das räumliche Gleichgewicht am ursprünglichen Rechennetz verringert.

Graz University of Technology

Senat

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am …………………………… ……………………………………………..
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

…………………………… ……………………………………………..
date (signature)

Contents

List of Figures

List of Tables

1. Preliminaries

This chapter gives an overview of the basic concepts that will be used in the framework presented dealing with r-adaptivity. This is a method for improving the solution of discretised partial differential equations by introducing the material coordinates as additional unknowns into the system of equations. Although r-adaptivity has a wide range of applications, this work is restricted to the static analysis of hyperelastic solids.

1.1. Equilibrium

A static analysis of solids is usually set up on the well known spatial equilibrium equations given in equation (1.1.1) and (1.1.2) respectively.
The local spatial translational equilibrium (for statics) in terms of Cauchy stresses is,

$$\frac{\partial \sigma_{ij}}{\partial x_j} + f_i^B = 0 \qquad i, j = x, y, z \tag{1.1.1}$$

This differential (strong) form of the equilibrium equation can be transferred into an integral (weak) form

$$\int\limits_{V_t} \left(\frac{\partial \sigma_{ij}}{\partial x_j} + f_i^B \right) \mathrm{d}V_t = 0 \tag{1.1.2}$$

where the first index of $\boldsymbol{\sigma}$ denotes the stress direction and the second one denotes the direction normal to the surface on which the stress is acting.

Investigating the rotational equilibrium it can be shown that the Cauchy stress tensor is symmetric ($\sigma_{ij} = \sigma_{ji}$)(Parkus 1960).

1.1.1. Principle of Virtual Power

Multiplying equation (1.1.1) with a virtual velocity $\delta\boldsymbol{v}$ yields after integration over the volume V_t (Bonet and Wood 1997),

$$\delta\dot{W} = \int\limits_{V_t} \left(\frac{\partial\sigma_{ij}}{\partial x_j} + f_i^B \right) \delta v_i \, \mathrm{d}V_t = 0 \tag{1.1.3}$$

Application of the product rule

$$\frac{\partial\sigma_{ij}}{\partial x_j}\delta v_i = \frac{\partial\left(\sigma_{ij}\delta v_i\right)}{\partial x_j} - \sigma_{ij}\frac{\partial\delta v_i}{\partial x_j} \tag{1.1.4}$$

and the Gauss theorem

$$\int\limits_{V_t} \frac{\partial\left(\sigma_{ij}\delta v_i\right)}{\partial x_j}\mathrm{d}V_t = \int\limits_{\partial V_t} n_j\left(\sigma_{ij}\delta v_i\right)\,\mathrm{d}A_t \tag{1.1.5}$$

on equation (1.1.3) gives

$$\delta\dot{W} = \int\limits_{\partial V_t} \sigma_{ij}\delta v_i n_j \,\mathrm{d}A_t - \int\limits_{V_t} \sigma_{ij}\frac{\partial\delta v_i}{\partial x_j}\,\mathrm{d}V_t + \int\limits_{V_t} f_i^B\delta v_i\,\mathrm{d}V_t = 0 \tag{1.1.6}$$

Finally, introducing the virtual symmetric velocity gradient $\delta\boldsymbol{d}$,

$$\delta d_{ij} = \frac{1}{2}\left(\frac{\partial\delta v_i}{\partial x_j} + \frac{\partial\delta v_j}{\partial x_i}\right) \tag{1.1.7}$$

yields,

$$\delta\dot{W} = \int\limits_{V_t} \sigma_{ij}\delta d_{ij}\,\mathrm{d}V_t - \int\limits_{V_t} f_i^B\delta v_i\,\mathrm{d}V_t - \int\limits_{\partial V_t} \underbrace{\sigma_{ij}n_j}_{t_i}\delta v_i\,\mathrm{d}A_t = 0 \tag{1.1.8}$$

The above equation is a weak formulation of the equilibrium equation, based on the principle of virtual power. In the scope of this thesis the principle of virtual power (Jourdain's principle) is equivalent to the principle of virtual work (d'Alembert's principle) (Glocker 1998).

In elastostatics, these principles represent tools corresponding to the more general concept of calculus of variations. They can be interpreted as an energy functional's arbitrary admissible variation with respect to spatial coordinates and are common in cases when the observer takes up the Lagrangian point of view. However, for r-adaptivity the Eulerian viewpoint is also considered. Thus, the concept of calculus of variations is used in this work, resulting in the formulation of Euler-Lagrange equations. Introducing the energy functional for elastostatic analysis,

$$\varPi = \int\limits_{V} \varPsi^{int} + \varPsi^{ext} \, \mathrm{d}V \tag{1.1.9}$$

where $\varPsi^{int}$ denotes the internal potential energy density. Hence, the functional $\varPi$ is the body's total potential energy. In this context, the Euler-Lagrange equations describe stationary points of the total energy which represent local points of equilibrium. They do not distinguish between stable and non-stable equilibrium. In standard elastostatic analysis this is a minor restriction, not so for r-adaptivity as will be shown later.

Furthermore, the connection between the Euler-Lagrange equations for spatial motion and the principle of virtual power or virtual work respectively will be outlined in the following sections.

1.2. Introduction to Spatial and Material Motion

R-adaptivity is usually settled in a framework called arbitrary Lagrangian-Eulerian (ALE) formulation (Thoutireddy 2003; Mosler and Ortiz 2006; Kuhl et al. 2004). This means that the deformation of a solid is described from the Lagrangian and from the Eulerian viewpoint. Classic elastostatics, as treated in many text books, work with the Lagrangian viewpoint describing a physical particle's spatial motion. This setting will be called spatial motion problem. The Eulerian viewpoint is used to describe changes in the material configuration, the material motion problem (Steinmann 2002). While the Lagrangian viewpoint is widely used in solid mechanics, the Eulerian viewpoint is far more common in fluid dynamics. Since both viewpoints describe the same physics, the exact solutions obtained from different viewpoints have to be identical. This holds for both spatial and material motion problem. However, finite element discretisation usually destroys this property, even in the case of homogeneous isotropic material (Steinmann et al. 2009). Depending on the discretisation, the differences are larger or smaller.

Interpretation of Material Motion - Configurational Mechanics: Keeping things simple, the spatial motion problem is motivated by the question, "Where will a given point of the undeformed body move to when the body is subjected to a certain load?". The material motion problem asks, "From where has a given point of a deformed body moved to its current position?". Assuming one has solved the spatial motion problem, it should not be difficult to obtain the inverse map and thus, obtain the solution of the material motion problem. This holds for undiscretised continua but not so for discretised as mentioned before.

Configurational mechanics deal with disturbed continua, inclusions, phase transitions et cetera. Despite the physical background, a rather formal variational approach to energy minimisation will be used to derive all formulations concerning r-adaptivity in hyperelastostatics as presented in (Kuhl et al. 2004).

1.2.1. ALE Kinematics

Configurations used in the ALE formulation presented are the material configuration (MC, undeformed body), the spatial configuration (SC, deformed body) and the reference configuration (RC, fixed). It is important to note that the material motion problem and the spatial motion problem can be formulated in terms of MC, SC and RC independently. The mapping between those three configurations is shown in Figure 1.2.1.

$\boldsymbol{\varphi}$ Spatial motion deformation map

$\boldsymbol{F}$ Spatial motion deformation gradient

$\boldsymbol{\Phi}$ Material motion deformation map

$\boldsymbol{f}$ Material motion deformation gradient

In mappings with a superscript * the reference configuration takes over the part of the material configuration, a subscript $_*$ indicates that the reference configuration "replaces" the spatial configuration. Thus following relationships between the three mappings hold,

$$\boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{X}, t) = \boldsymbol{\varphi}^*(\boldsymbol{\xi}, t) \tag{1.2.1}$$

$$\boldsymbol{\varphi}(\boldsymbol{X}, t) = \boldsymbol{\varphi}^*(\boldsymbol{\xi}, t) \circ \boldsymbol{\varphi}_*(\boldsymbol{X}, t) \tag{1.2.2}$$

$$\boldsymbol{F} = \frac{\partial \boldsymbol{\varphi}(\boldsymbol{X}, t)}{\partial \boldsymbol{X}} = \frac{\partial \boldsymbol{\varphi}(\boldsymbol{X}, t)}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}} = \boldsymbol{F}^* \cdot \boldsymbol{F}_* \tag{1.2.3}$$

$$\boldsymbol{X} = \boldsymbol{\Phi}(\boldsymbol{x}, t) = \boldsymbol{\Phi}_*(\boldsymbol{\xi}, t) \tag{1.2.4}$$

$$\boldsymbol{\Phi}(\boldsymbol{x}, t) = \boldsymbol{\Phi}_*(\boldsymbol{\xi}, t) \circ \boldsymbol{\Phi}^*(\boldsymbol{x}, t) \tag{1.2.5}$$

$$\boldsymbol{f} = \frac{\partial \boldsymbol{\Phi}(\boldsymbol{x}, t)}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{\Phi}(\boldsymbol{x}, t)}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}} = \boldsymbol{f}_* \cdot \boldsymbol{f}^* \tag{1.2.6}$$

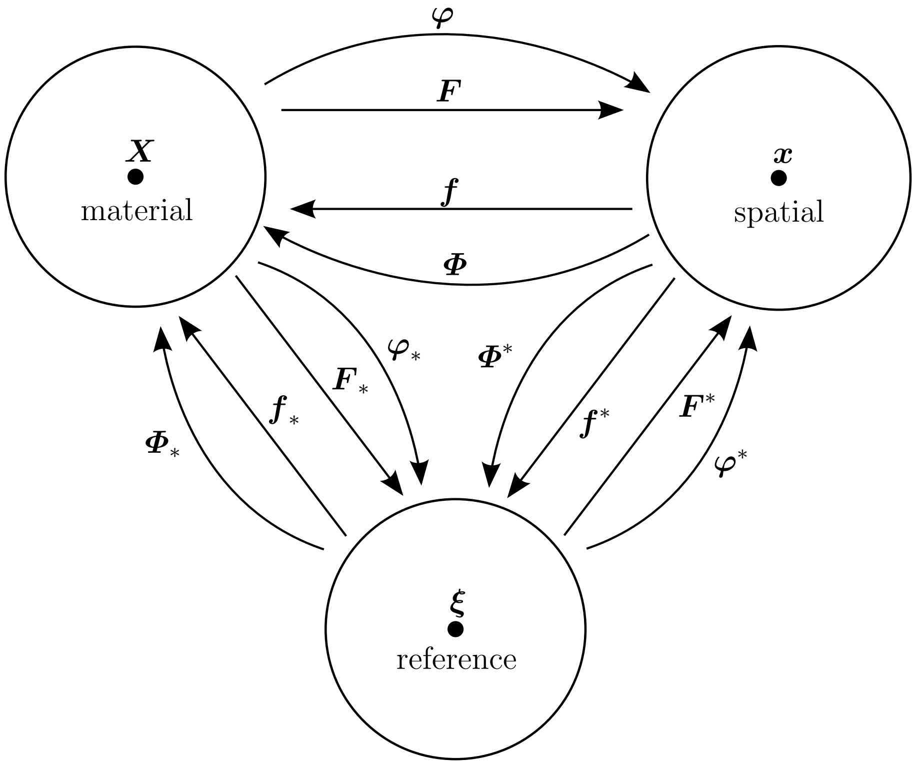

Figure 1.2.1. Mappings between configurations

More definitions used in the following sections:
Variation

$$\delta\{\bullet\} = \delta_{\boldsymbol{x}}\{\bullet\} + \delta_{\boldsymbol{X}}\{\bullet\} \tag{1.2.7}$$

$$\delta_{\boldsymbol{x}}\{\bullet\} = \delta\{\bullet\}|_{\boldsymbol{X}} \tag{1.2.8}$$

$$\delta_{\boldsymbol{X}}\{\bullet\} = \delta\{\bullet\}|_{\boldsymbol{x}} \tag{1.2.9}$$

and analogously:
Incrementation

$$\varDelta\{\bullet\} = \varDelta_{\boldsymbol{x}}\{\bullet\} + \varDelta_{\boldsymbol{X}}\{\bullet\} \tag{1.2.10}$$

$$\varDelta_{\boldsymbol{x}}\{\bullet\} = \varDelta\{\bullet\}|_{\boldsymbol{X}} \tag{1.2.11}$$

$$\varDelta_{\boldsymbol{X}}\{\bullet\} = \varDelta\{\bullet\}|_{\boldsymbol{x}} \tag{1.2.12}$$

1.2.2. Transformations Between Configurations

The relation between the infinitesimal volume calculated on the material configuration $\mathrm{d}V_0$ and the infinitesimal volume calculated on the spatial configuration $\mathrm{d}V_t$ is given as,

$$\mathrm{d}V_t = J\mathrm{d}V_0 \tag{1.2.13}$$
$$\mathrm{d}V_0 = j\mathrm{d}V_t \tag{1.2.14}$$

with,

$$J = \det \boldsymbol{F}$$
$$j = \det \boldsymbol{f} = 1/J$$

This leads to,

$$\int_{V_0} \{\bullet\}_0 \, \mathrm{d}V_0 = \int_{V_t} \{\bullet\}_t \, \mathrm{d}V_t = \int_{V_t} \{\bullet\}_t \, J\mathrm{d}V_0$$

and thus,

$$\{\bullet\}_t = 1/J \, \{\bullet\}_0 = j \, \{\bullet\}_0 \tag{1.2.15}$$

Equation (1.2.15) gives the transformation of scalar quantities between configurations.

For the derivation of an equilibrium equation through variation of the potential energy $\varPi$ with respect to the spatial coordinates $\boldsymbol{x}$, the following expressions will be used (Steinmann 2002):

$$\delta_{\boldsymbol{x}}\{\bullet\}_0(\boldsymbol{F},\boldsymbol{\varphi};\boldsymbol{X}) = \mathrm{D}_{\boldsymbol{F}}\{\bullet\}_0 \colon \delta_{\boldsymbol{x}}\boldsymbol{F} + \frac{\partial\{\bullet\}_0}{\partial \boldsymbol{x}} \cdot \delta_{\boldsymbol{x}}\boldsymbol{\varphi} \tag{1.2.16}$$

In equation (1.2.16) the operator $\mathrm{D}_{\boldsymbol{F}}$ is the derivative with respect to the spatial motion deformation gradient, whereby the quantity $\{\bullet\}$ is expressed for the spatial motion problem[1]. The index $_0$ indicates that $\{\bullet\}$ is a quantity on the material configuration. An index $_t$ denotes a quantity on the spatial configuration. Besides $\mathrm{D}_{\boldsymbol{F}}$ there also exist the operators $\mathrm{D}_{\boldsymbol{f}}$, $\mathrm{d}_{\boldsymbol{F}}$ and $\mathrm{d}_{\boldsymbol{f}}$. A capital "D" is used for functions expressed for the spatial motion problem and a lowercase "d" for functions expressed for the material motion problem. In the hyperelastic static case the distinction between $\mathrm{D}_{\boldsymbol{F}}$ and $\mathrm{d}_{\boldsymbol{F}}$ is not necessary. The same holds for $\mathrm{D}_{\boldsymbol{f}}$ and $\mathrm{d}_{\boldsymbol{f}}$ (Steinmann 2002). Thus, the derivatives with respect the deformation gradients, will be written as $\partial_{\boldsymbol{F}}$ or $\partial_{\boldsymbol{f}}$ respectively.

According to Steinmann (2002), the following is valid for a scalar object,

$$\mathrm{D}_{\boldsymbol{F}} \{\bullet\}_0 = \{\bullet\}_0 \, \boldsymbol{f}^T - J\boldsymbol{f}^T \cdot \mathrm{D}_{\boldsymbol{f}} \{\bullet\}_t \cdot \boldsymbol{f}^T \tag{1.2.17}$$

As mentioned above, in the hyperelastic static case this can be written as,

$$\partial_{\boldsymbol{F}} \{\bullet\}_0 = \{\bullet\}_0 \, \boldsymbol{f}^T - J\boldsymbol{f}^T \cdot \partial_{\boldsymbol{f}} \{\bullet\}_t \cdot \boldsymbol{f}^T \tag{1.2.18}$$

A proof for equation (1.2.18) is given in the Appendix (A.1.1).

[1] "Expressed for the spatial motion problem" means that all quantities are parametrised in terms of material quantities, e.g. $\boldsymbol{\varphi} = \boldsymbol{\varphi}(\boldsymbol{X})$. Consequently, "expressed for the material motion problem" means parametrised with spatial quantities, e.g. $\boldsymbol{\varPhi} = \boldsymbol{\varPhi}(\boldsymbol{x})$.

1.3. Spatial Motion Problem

The description of the position of a physical particle in the spatial configuration by means of the nonlinear deformation map $\boldsymbol{\varphi}$ is the spatial motion problem.

Making use of the transformations given above, the variation of the total potential energy yields,

$$\begin{aligned}\delta_{\boldsymbol{x}} \mathit{\Pi} &= \int\limits_{V_0} \left[\delta_{\boldsymbol{x}} \mathit{\Psi}_0^{int} + \delta_{\boldsymbol{x}} \mathit{\Psi}_0^{ext}\right] \mathrm{d}V_0 \\ &= \int\limits_{V_0} \left[\left(\partial_{\boldsymbol{F}} \mathit{\Psi}_0^{int} \colon \delta_{\boldsymbol{x}} \boldsymbol{F} + \partial_{\boldsymbol{x}} \mathit{\Psi}_0^{int} \cdot \delta_{\boldsymbol{x}} \boldsymbol{\varphi}\right) + \left(\partial_{\boldsymbol{F}} \mathit{\Psi}_0^{ext} \colon \delta_{\boldsymbol{x}} \boldsymbol{F} + \partial_{\boldsymbol{x}} \mathit{\Psi}_0^{ext} \cdot \delta_{\boldsymbol{x}} \boldsymbol{\varphi}\right)\right] \mathrm{d}V_0 \quad (1.3.1)\end{aligned}$$

Considering the common parametrisations $\mathit{\Psi}_0^{int} = \mathit{\Psi}_0^{int}(\boldsymbol{F}; \boldsymbol{X})$ and $\mathit{\Psi}_0^{ext} = \mathit{\Psi}_0^{ext}(\boldsymbol{\varphi}; \boldsymbol{X})$, equation (1.3.1) can be simplified to,

$$\delta_{\boldsymbol{x}} \mathit{\Pi} = \int\limits_{V_0} \left[\partial_{\boldsymbol{F}} \mathit{\Psi}_0^{int} \colon \delta_{\boldsymbol{x}} \boldsymbol{F} + \partial_{\boldsymbol{x}} \mathit{\Psi}_0^{ext} \cdot \delta_{\boldsymbol{x}} \boldsymbol{\varphi}\right] \mathrm{d}V_0 \qquad (1.3.2)$$

Finally, the transformation of equation (1.3.2) from the material to the spatial domain yields,

$$\begin{aligned}\delta_{\boldsymbol{x}} \mathit{\Pi} &= \int\limits_{V_0} \left[\left(\mathit{\Psi}_0^{int} \boldsymbol{f}^T - J \boldsymbol{f}^T \cdot \mathrm{D}_{\boldsymbol{f}} \mathit{\Psi}_t^{int} \cdot \boldsymbol{f}^T\right) : \delta_{\boldsymbol{x}} \boldsymbol{F} + \partial_{\boldsymbol{x}} \mathit{\Psi}_0^{ext} \cdot \delta_{\boldsymbol{x}} \boldsymbol{\varphi}\right] \mathrm{d}V_0 \\ &= \int\limits_{V_t} \left[\left(\mathit{\Psi}_t^{int} \boldsymbol{f}^T - \boldsymbol{f}^T \cdot \mathrm{d}_{\boldsymbol{f}} \mathit{\Psi}_t^{int} \cdot \boldsymbol{f}^T\right) : \delta_{\boldsymbol{x}} \boldsymbol{F} + \partial_{\boldsymbol{x}} \mathit{\Psi}_t^{ext} \cdot \delta_{\boldsymbol{x}} \boldsymbol{\varphi}\right] \mathrm{d}V_t \\ &= \int\limits_{V_t} \left[\left(\mathit{\Psi}_t^{int} \boldsymbol{I} - \boldsymbol{f}^T \cdot \mathrm{d}_{\boldsymbol{f}} \mathit{\Psi}_t^{int}\right) : \frac{\partial \delta_{\boldsymbol{x}} \boldsymbol{\varphi}}{\partial \boldsymbol{x}} + \partial_{\boldsymbol{x}} \mathit{\Psi}_t^{ext} \cdot \delta_{\boldsymbol{x}} \boldsymbol{\varphi}\right] \mathrm{d}V_t \qquad (1.3.3)\end{aligned}$$

where $\boldsymbol{I}$ is the (appropriate) unit tensor.

1.3.1. Stress Tensors for the Spatial Motion Problem

In equation (1.3.2), expression $\mathrm{D}_{\boldsymbol{F}} \mathit{\Psi}_0^{int}$ is the First Piola-Kirchhoff stress tensor $\boldsymbol{P}$ (Steinmann 2002). Changing the integration domain from material to spatial, $\mathrm{D}_{\boldsymbol{F}} \mathit{\Psi}_0^{int} \colon \delta_{\boldsymbol{x}} \boldsymbol{F}$ ends up in

$\left(\mathit{\Psi}_t^{int}\boldsymbol{I} - \boldsymbol{f}^T \cdot \mathrm{d}_{\boldsymbol{f}}\mathit{\Psi}_t^{int}\right) : \frac{\partial \delta_{\boldsymbol{x}}\boldsymbol{\varphi}}{\partial \boldsymbol{x}}$, which is interesting, because it results in an alternative expression for the Cauchy stress tensor

$$\boldsymbol{\sigma} = j\boldsymbol{P} \cdot \boldsymbol{F}^T = j\mathrm{D}_{\boldsymbol{F}}\mathit{\Psi}_0^{int} \cdot \boldsymbol{F}^T = \mathit{\Psi}_t^{int}\boldsymbol{I} - \boldsymbol{f}^T \cdot \mathrm{d}_{\boldsymbol{f}}\mathit{\Psi}_t^{int} \tag{1.3.4}$$

and the transformation between $\boldsymbol{P}$ and $\boldsymbol{\sigma}$ is a Piola transformation [2]. Inserting equation (1.3.4) into (1.3.3) gives,

$$\delta_{\boldsymbol{x}}\mathit{\Pi} = \int\limits_{V_t} \left[\boldsymbol{\sigma} : \frac{\partial \delta_{\boldsymbol{x}}\boldsymbol{\varphi}}{\partial \boldsymbol{x}} + \partial_{\boldsymbol{x}}\mathit{\Psi}_t^{ext} \cdot \delta_{\boldsymbol{x}}\boldsymbol{\varphi}\right] \mathrm{d}V_t \tag{1.3.5}$$

Comparison of the above equation with (1.1.8) shows the equivalence of the Principle of Virtual Power and the variational approach used in this thesis. However, this statement only holds for the elastostatic conservative case and things become more complicated for dynamic problems. Steinmann (2002) also discusses the elastodynamic case using the Hamiltonian framework.

It should be mentioned that many authors, including Steinmann, write,

$$\mathrm{D}_{\boldsymbol{F}}\mathit{\Psi}_0^{int} = \boldsymbol{P}^T \quad \text{and} \tag{1.3.6}$$

$$\boldsymbol{\sigma}^T = j\boldsymbol{P}^T \cdot \boldsymbol{F}^T \tag{1.3.7}$$

As long as these tensors are used consistently within a more or less closed framework, this makes no difference at all. The transposition may result from different “sortings” of the Cauchy stress tensor’s components, depending on their interpretation. In this work, the first index (i) of σ_{ij} is the stress component’s direction, the second index j denotes the direction of the surface, on which the component is “acting”. The second possible interpretation is exactly the opposite, or in other words, $\boldsymbol{\sigma}_{notation1} = \boldsymbol{\sigma}^T_{notation2}$.

Remark 1.1. Covariance and contravariance: Taking contravariant components from position vectors $\boldsymbol{x}$ and $\boldsymbol{X}$, the deformation gradient is, in index notation, written as,

$$F^i_{\ I} = \frac{\partial x^i}{\partial X^I}$$

and hence,

$$\frac{\partial \mathit{\Psi}_0^{int}}{\partial F^i_{\ I}} = P_i^{\ I}$$

$P_i^{\ I}$ are mixed components of the first Piola-Kirchhoff tensor. This is of theoretical importance for transforms like the Piola transform between $\boldsymbol{P}$ and $\boldsymbol{\sigma}$. Applying index notation to equation (1.3.4) yields,

$$j\, P_i^{\ I}\, F^j_{\ I} = \sigma_i^{\ j}$$

[2] In this case, the transformation can be obtained by applying Nanson’s formula $da\,\boldsymbol{n} = J\,dA\,\boldsymbol{F}^{-T} \cdot \boldsymbol{N}$

but the commonly used components of Cauchy stresses are contravariant σ^{ij}. This means a metric tensor has been omitted. This is not a big deal when using rectangular Cartesian reference frames, where metric tensors are unit tensors $\boldsymbol{I}$ in "their" frame. As this thesis only uses such reference frames, the distinction between covariant and contravariant components is not necessary and in most equations a simplified index format will be used, e.g.,

$$F^i_I = \frac{\partial x^i}{\partial X^I} \rightarrow F_{ij} = \frac{\partial x_i}{\partial X_j}$$

1.3.2. Spatial Equilibrium

To keep things simple, this work concentrates on displacement driven examples and therefore, the external potential energy will not be considered any further. This leads to a short equation for spatial equilibrium given on the material and spatial domain:

$$\delta_{\boldsymbol{x}} \Pi = \int\limits_{V_0} \delta_{\boldsymbol{x}} \boldsymbol{F} \colon \boldsymbol{P} \,\mathrm{d}V_0 = 0 \tag{1.3.8}$$

$$\delta_{\boldsymbol{x}} \Pi = \int\limits_{V_t} \frac{\partial \delta_{\boldsymbol{x}} \boldsymbol{\varphi}}{\partial \boldsymbol{x}} \colon \boldsymbol{\sigma} \,\mathrm{d}V_t = 0 \tag{1.3.9}$$

or in index notation, considering Einstein's sum convention,

$$\delta_{\boldsymbol{x}} \Pi = \int\limits_{V_0} \delta_{\boldsymbol{x}} F_{ij} \, P_{ij} \,\mathrm{d}V_0 = 0 \tag{1.3.10}$$

$$\delta_{\boldsymbol{x}} \Pi = \int\limits_{V_t} \frac{\partial \delta_{\boldsymbol{x}} \varphi_i}{\partial x_j} \, \sigma_{ij} \,\mathrm{d}V_t = 0 \tag{1.3.11}$$

1.4. Material Motion Problem

Proceeding in analogy to the spatial motion problem, an equation for material equilibrium can be derived as follows.

$$\delta_{\boldsymbol{X}} \Pi = \int\limits_{V_t} \left[\delta_{\boldsymbol{X}} \Psi_t^{int} + \delta_{\boldsymbol{X}} \Psi_t^{ext} \right] \mathrm{d}V_t \tag{1.4.1}$$

Seeking a variation with respect to material coordinates $\boldsymbol{X}$, the corresponding variations are introduced,

$$\delta_{\boldsymbol{X}} \left\{ \bullet \right\}_t (\boldsymbol{f}, \boldsymbol{\Phi}; \boldsymbol{x}) = \mathrm{d}_{\boldsymbol{f}} \left\{ \bullet \right\}_t : \delta_{\boldsymbol{X}} \boldsymbol{f} + \frac{\partial \left\{ \bullet \right\}_t}{\partial \boldsymbol{x}} \cdot \delta_{\boldsymbol{X}} \boldsymbol{\Phi} \tag{1.4.2}$$

Above expression as well as considering the absence of $\varPsi^{ext}$, applied in the variation of the total potential energy with respect to material coordinates, gives,

$$\delta_{\boldsymbol{X}} \varPi = \int\limits_{V_t} \left[\mathrm{d}_{\boldsymbol{f}} \varPsi_t^{int} \colon \delta_{\boldsymbol{X}} \boldsymbol{f} + \partial_{\boldsymbol{X}} \varPsi_t^{int} \cdot \delta_{\boldsymbol{X}} \boldsymbol{\varPhi} \right] \mathrm{d}V_t \tag{1.4.3}$$

As $\varPsi_t^{int} = \varPsi_t^{int}(\boldsymbol{f}; \boldsymbol{x})$ the second term in equation (1.4.1) vanishes and we obtain,

$$\delta_{\boldsymbol{X}} \varPi = \int\limits_{V_t} \mathrm{d}_{\boldsymbol{f}} \varPsi_t^{int} \colon \delta_{\boldsymbol{X}} \boldsymbol{f} \, \mathrm{d}V_t \tag{1.4.4}$$

To show the connection to Eshelbian mechanics, $\mathrm{d}_{\boldsymbol{f}} \left\{ \bullet \right\}_t$ or $\partial_{\boldsymbol{f}} \left\{ \bullet \right\}_t$ respectively is obtained from equation (1.2.18),

$$\partial_{\boldsymbol{f}} \left\{ \bullet \right\}_t = \left\{ \bullet \right\}_t \boldsymbol{F}^T - j \, \boldsymbol{F}^T \cdot \partial_{\boldsymbol{F}} \left\{ \bullet \right\}_0 \cdot \boldsymbol{F}^T \tag{1.4.5}$$

With equations (1.2.14) and (1.2.15) the pull back of equation (1.4.4) can be computed:

$$\begin{aligned} \delta_{\boldsymbol{X}} \varPi &= \int\limits_{V_t} \left(\varPsi_t^{int} \, \boldsymbol{F}^T - j \, \boldsymbol{F}^T \cdot \mathrm{D}_{\boldsymbol{F}} \varPsi_0^{int} \cdot \boldsymbol{F}^T \right) : \delta_{\boldsymbol{X}} \boldsymbol{f} \, \mathrm{d}V_t \\ &= \int\limits_{V_0} \left(\varPsi_0^{int} \, \boldsymbol{F}^T - \boldsymbol{F}^T \cdot \mathrm{D}_{\boldsymbol{F}} \varPsi_0^{int} \cdot \boldsymbol{F}^T \right) : \delta_{\boldsymbol{X}} \boldsymbol{f} \, \mathrm{d}V_0 \\ &= \int\limits_{V_0} \left(\varPsi_0^{int} \, \boldsymbol{I} - \boldsymbol{F}^T \cdot \mathrm{D}_{\boldsymbol{F}} \varPsi_0^{int} \right) : \frac{\partial \delta_{\boldsymbol{X}} \boldsymbol{\varPhi}}{\partial \boldsymbol{X}} \, \mathrm{d}V_0 \end{aligned} \tag{1.4.6}$$

While,

$$\boldsymbol{p} = \mathrm{d}_{\boldsymbol{f}} \varPsi_t^{int} \tag{1.4.7}$$

is the material motion analogon to the First Piola-Kirchhoff stress tensor for the spatial motion problem,

$$\boldsymbol{\varSigma} = \varPsi_0^{int} \, \boldsymbol{I} - \boldsymbol{F}^T \cdot \mathrm{D}_{\boldsymbol{F}} \varPsi_0^{int} = J \, p_{ik} f_{jk} \tag{1.4.8}$$

is the Eshelby stress tensor. The structure of (1.4.8) for the Eshelby stress tensor is similar to the structure of (1.3.4) for the Cauchy stress tensor.

1.4.1. Material Equilibrium

To summarise the results above, the material equilibrium equation with all simplifications as it will be used in this work is given for the material and spatial domain as follows,

$$\delta_{\boldsymbol{X}} \Pi = \int\limits_{V_t} \delta_{\boldsymbol{X}} \boldsymbol{f} : \boldsymbol{p} \, \mathrm{d}V_t = 0 \tag{1.4.9}$$

$$\delta_{\boldsymbol{X}} \Pi = \int\limits_{V_0} \frac{\partial \delta_{\boldsymbol{X}} \boldsymbol{\Phi}}{\partial \boldsymbol{X}} : \boldsymbol{\Sigma} \, \mathrm{d}V_0 = 0 \tag{1.4.10}$$

$$\delta_{\boldsymbol{X}} \Pi = \int\limits_{V_t} \delta_{\boldsymbol{X}} f_{ij} \, p_{ij} \, \mathrm{d}V_t = 0 \tag{1.4.11}$$

$$\delta_{\boldsymbol{X}} \Pi = \int\limits_{V_0} \frac{\partial \delta_{\boldsymbol{X}} \Phi_i}{\partial X_j} \, \Sigma_{ij} \, \mathrm{d}V_0 = 0 \tag{1.4.12}$$

1.5. Nonlinear Equations

Nonlinear equations are equations

$$f(x) = 0$$

where $f(x)$ is nonlinear in x.

The stress tensors in the spatial and material equilibrium equations are defined as functions of the spatial or material motion deformation gradient respectively. These functions are nonlinear for most materials, including neo-Hookean material, which is used later in this thesis. Hence, the equilibrium equations are treated as nonlinear equations. Before introducing the Finite Element Method (chapter 2) some aspects of nonlinear optimisation problems and resulting nonlinear equations will be discussed.

1.5.1. Newton-Raphson Method for Nonlinear Systems of Equations

The Newton-Raphson method is an iterative root finding algorithm for nonlinear equations and nonlinear systems of equations.

Take the nonlinear system $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}$: Linearisation starting from an initial guess $\boldsymbol{x}^0$ yields,

$$\boldsymbol{F}(\boldsymbol{x}^0) + \left.\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}^0} \Delta \boldsymbol{x}^0 = 0 \tag{1.5.1}$$

Now, calculating $\boldsymbol{x}^1 = \boldsymbol{x}^0 + \Delta \boldsymbol{x}^0$ and generally it will come out that $\boldsymbol{F}(\boldsymbol{x}^1) \neq \boldsymbol{0}$. However, if $\boldsymbol{x}^0$ is sufficiently close to the exact solution, then $||\boldsymbol{F}(\boldsymbol{x}^1)|| < ||\boldsymbol{F}(\boldsymbol{x}^0)||$.

Hence, calculating $\boldsymbol{x}^2$ in the same way yields an approximate solution $\boldsymbol{x}^2$ with $||\boldsymbol{F}(\boldsymbol{x}^2)|| < ||\boldsymbol{F}(\boldsymbol{x}^1)||$. This procedure is repeated until the error $||\boldsymbol{F}(\boldsymbol{x}^n)||$ is sufficiently small. A 1d illustration of Newton's method is given in Figure 1.5.1. The algorithm is given in Table 1.1. A geometrical interpretation of the Newton-Raphson scheme is given in Figure 1.5.1.

Table 1.1. General Newton-Raphson Algorithm

```
set CONVERGED = False
choose initial guess $\boldsymbol{x}^0$, set iteration counter $n = 0$
**while** CONVERGED = False **do**
    compute $\boldsymbol{F}(\boldsymbol{x}^n)$ and $\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}\Big|_n$
    solve $\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}\Big|_n \Delta\boldsymbol{x}^n = -\boldsymbol{F}(\boldsymbol{x}^n)$
    compute next position $\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + \Delta\boldsymbol{x}^n$
    check the error $\epsilon^{n+1} = ||\boldsymbol{F}(\boldsymbol{x}^{n+1})||$[3]
    **if** $\epsilon^{n+1} \leq$ tol **then**
        set CONVERGED = True
    **else**
        increment iteration counter $n = n + 1$
    **end if**
**end while**
```

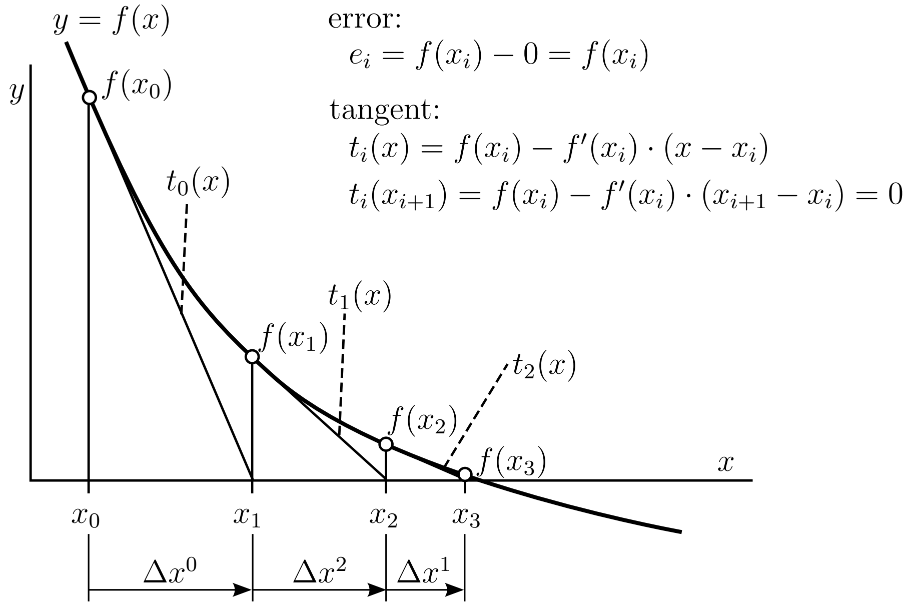

Figure 1.5.1. Geometrical interpretation of the Newton-Raphson scheme applied to a 1d function ("Newton's method")

[3] There exist some other error measures like the unbalanced energy introduced in Section 2.6.1

1.5.2. A One-dimensional Minimisation Problem

Considering a more or less arbitrary potential

$$\Pi(x) = 0.2\,x^2 - \cos{(x)} \tag{1.5.2}$$

that should be minimised. The potential Π is depicted in Figure 1.5.2.

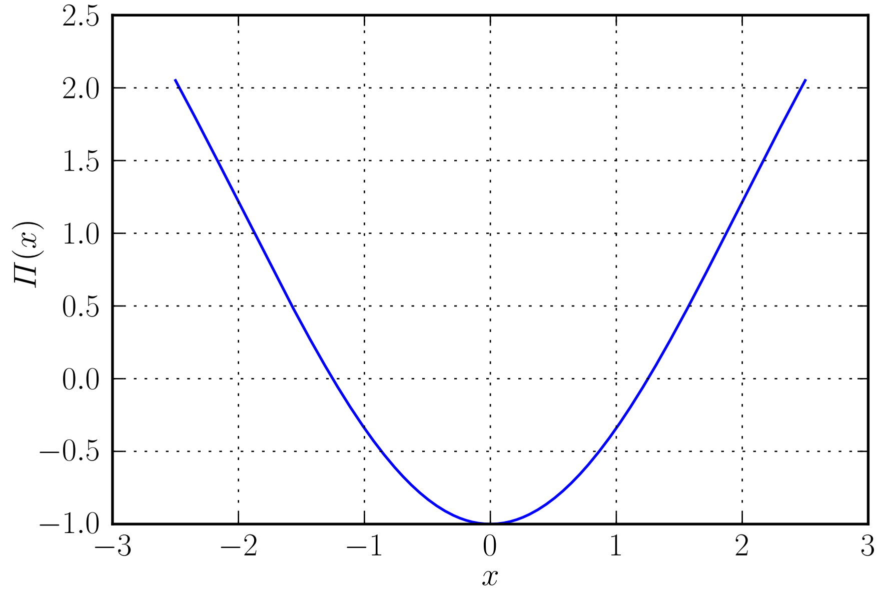

Figure 1.5.2. Example potential $\Pi(x)$

To find a (local) minimum, $\Pi(x)$ will be derived with respect to x. At an extremum, this derivative equals zero and if this extremum is a local minimum the second derivative of Π has to be greater than zero. Thus, one is seeking a point $\bar{x}$ where,

$$\left.\frac{\mathrm{d}\Pi(x)}{\mathrm{d}x}\right|_{\bar{x}} = 0 \tag{1.5.3}$$

$$\left.\frac{\mathrm{d}^2\Pi(x)}{\mathrm{d}x^2}\right|_{\bar{x}} > 0 \tag{1.5.4}$$

hold. Equation (1.5.3) has to be solved and equation (1.5.4) is the criterion for a local minimum.

The first and second order derivatives of Π are given as,

$$\frac{\mathrm{d}\Pi(x)}{\mathrm{d}x} = 0.4\,x + sin(x) \tag{1.5.5}$$

$$\frac{\mathrm{d}^2\Pi(x)}{\mathrm{d}x^2} = 0.4 + cos(x) \tag{1.5.6}$$

Obviously, equation (1.5.5) is nonlinear. Finding a local minimum of the given potential with the Newton-Raphson scheme with an initial guess $x^0 = -1$ is depicted below. Figure 1.5.3 shows the Newton-Raphson algorithm applied to the gradient of Π given in equation (1.5.5). At $\bar{x} = 0$, equation (1.5.3) and condition (1.5.4) hold. Therefore, at $\bar{x} = 0$ the potential Π has a (local) minimum. Figure 1.5.4 depicts the values of Π evaluated at the Newton-Raphson iterations x^i.

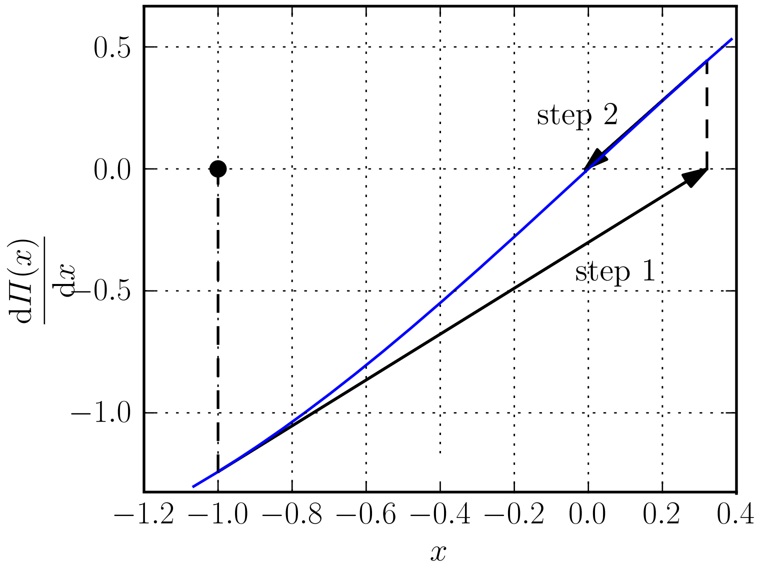

Figure 1.5.3. The Newton-Raphson algorithm applied to equation (1.5.5) with $x^0 = -1.0$

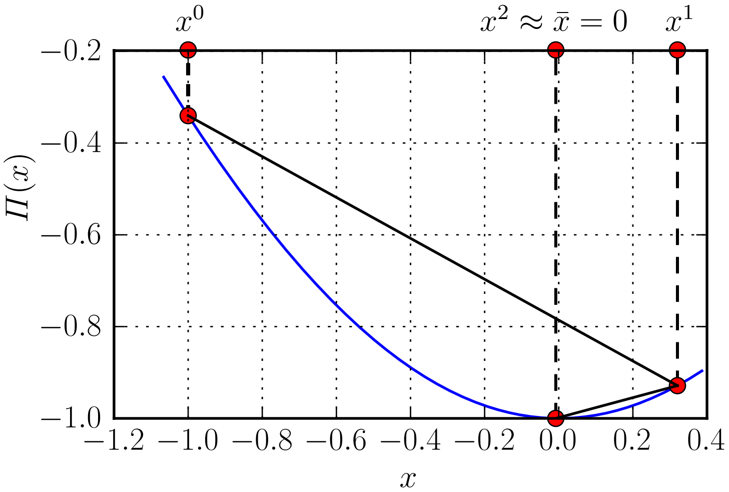

Figure 1.5.4. Potential Π evaluated at Newton-Raphson iterations with $x^0 = -1.0$

However, setting $x^0 = -1.5$ causes severe convergence problems for the Newton-Raphson algorithm. Figure 1.5.5 depicts the first 3 Newton-Raphson iterations (steps). At step 3 the solution $x^3 \approx -13$ is far away from the exact solution $\bar{x} = 0$ and also far away from the initial guess $x^0 = -1.5$. Again, the potential $\varPi$ is evaluated at the Newton-Raphson iterations x^i (see Figure 1.5.6.

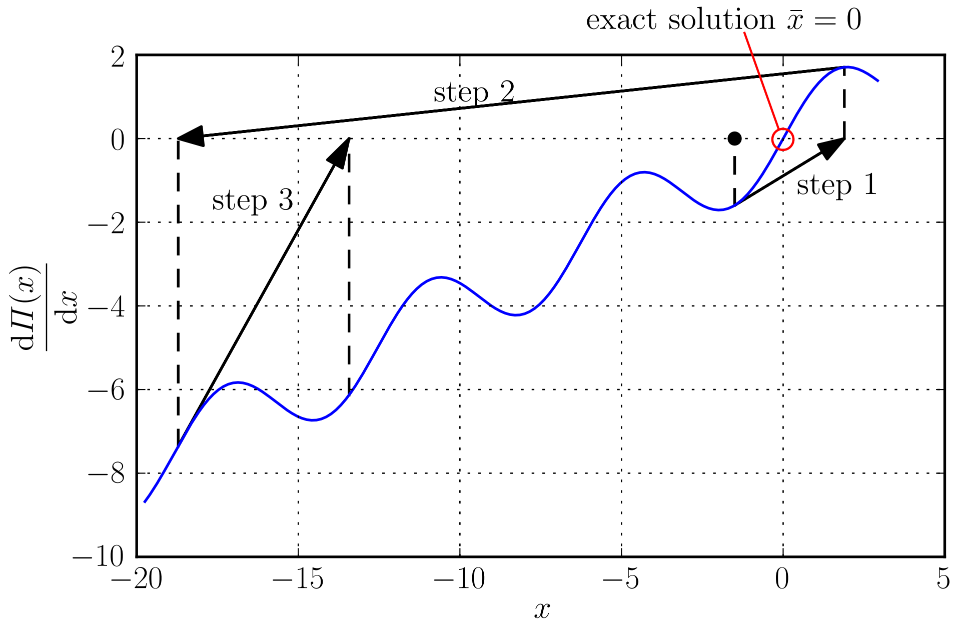

Figure 1.5.5. The Newton-Raphson algorithm applied to equation (1.5.5) with $x^0 = -1.5$

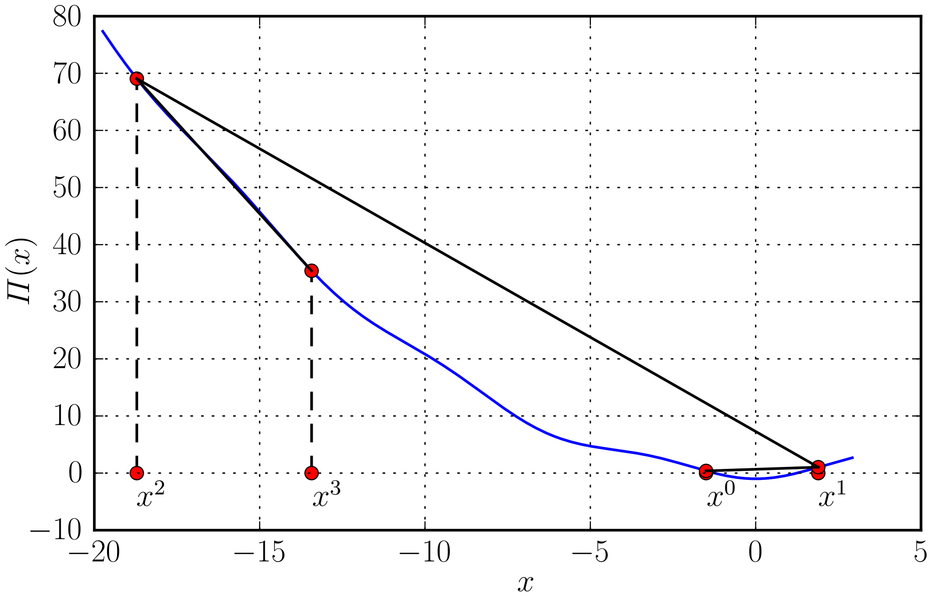

Figure 1.5.6. Potential $\varPi$ evaluated at Newton-Raphson iterations with $x^0 = -1.5$

The reason for the convergence problems shown above is a Hessian of almost zero at x_1. This is indicated by the nearly horizontal tangent to the gradient for Newton-step 2 in

Figure 1.5.5. The Hessian of the potential Π as given in equation (1.5.2) is depicted in Figure 1.5.7.

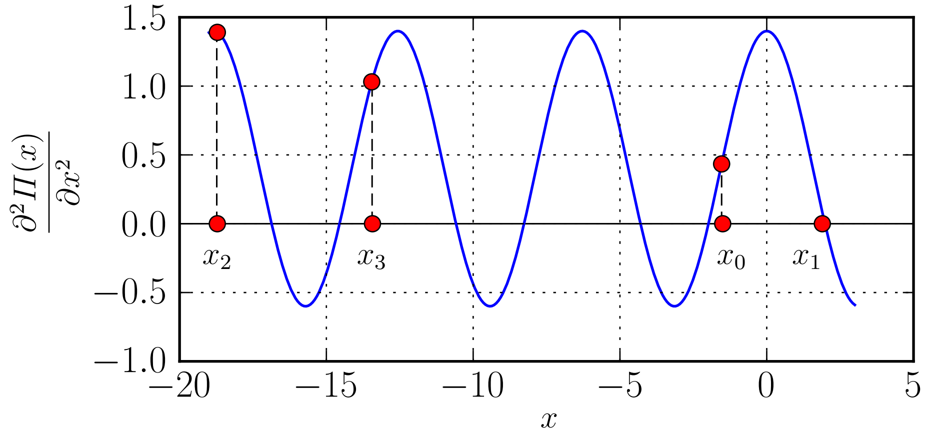

Figure 1.5.7. Hessian of potential Π

2. Finite Element Method

The Finite Element Method (FEM) is the standard discretisation method in mechanics of solids. In this chapter the basic concept of the displacement-based FEM will be introduced without going into too much detail.

2.1. The Concept of Finite Elements

The idea is to partition a body into finite elements and obtain approximations of the exact solution on these elements. Of course, these approximations have to fulfil a set of criteria, e.g. gaps must not emerge between the elements. These criteria imply additional equations coupling the corresponding elements, leading to a system of equations for the global set of differential equations. The approximations themselves are usually based on interpolations of the exact solution.

2.2. Finite Element Shapes

The first decision in any finite element analysis (FEA) is the selection of finite element shapes, the discretisation of the body itself. There exist shapes for one-, two-, three-dimensional and even for higher dimensional problems. Besides very basic elements like hexahedra or tetrahedra that are called volume elements, structural elements like shell-elements have been developed, which are able to decrease the computational effort significantly. These elements are dedicated to specific applications and therefore already include further modelling and are not generally applicable. However, this work is restricted to volume elements and their equivalents for lower dimensional problems. In Figure 2.2.1, discretisations of a rectangle and a quadrant are shown. The first mesh is called structured and consists of quadrilateral elements. The latter discretisation is of an unstructured type, the shapes are linear triangular. Observing this discretisation at the curved borders, one can see the difference between the outline of the continuous body and the discretised mesh consisting of triangles.

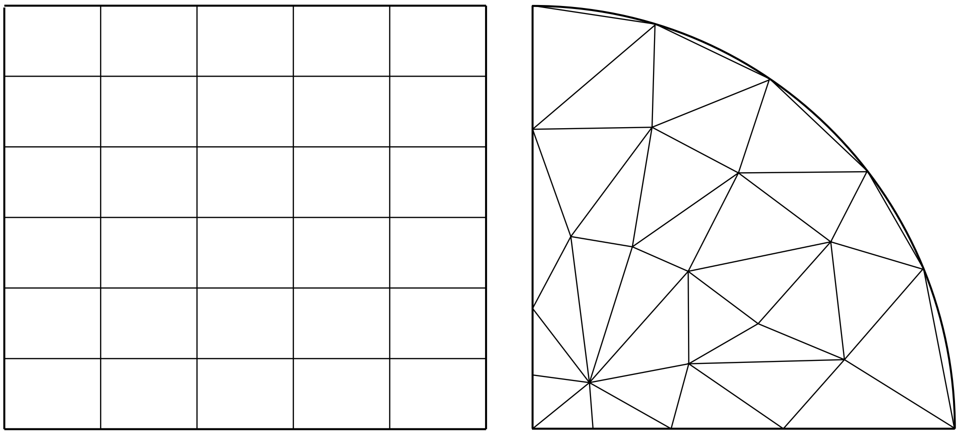

Figure 2.2.1. Example of a structured (left) and an unstructured (right) two-dimensional mesh

2.3. Interpolation

Interpolation is a very basic tool for the approximation of a function. Basically, there are two possible approaches to interpolation:

- global interpolation and
- piecewise interpolation.

An example for global and piecewise interpolation is illustrated in Figure 2.3.1.

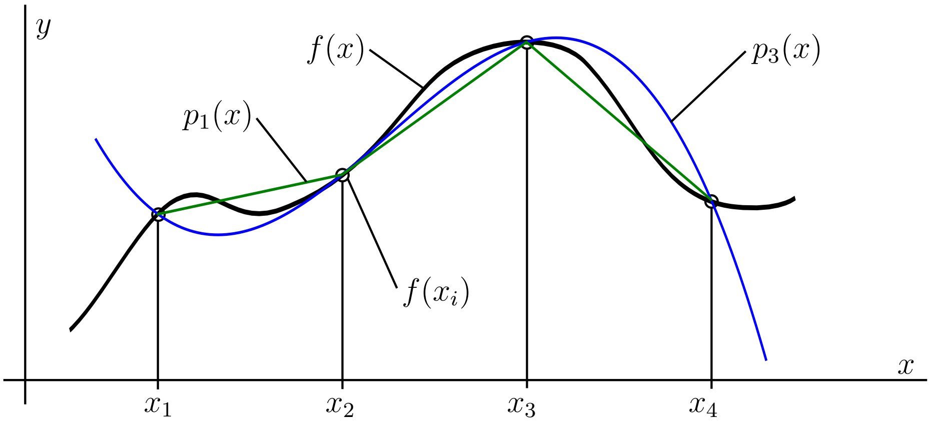

Figure 2.3.1. Piece-wise linear and global 3rd order polynomial interpolation

In Figure 2.3.1 the function $f(x)$ should be interpolated, $p_1(x)$ is a piecewise linear interpolation polynomial and $p_3(x)$ is a global interpolation polynomial of order 3. While $p_3(x)$ is a single curve that fulfils the interpolation condition at all four interpolation points $(x_1, \ldots, x_4)$, $p_1(x)$ consists of three concatenated linear interpolation polynomials. Each of these linear polynomials fulfil the interpolation condition only at a subset of interpolation points.

2.3.1. Interpolation Error

Error analysis of interpolation tells us that the error can be reduced by using higher order interpolation polynomials. However, higher order polynomials cause a higher computational effort and sometimes numerical difficulties arise. Polynomials become more "sensitive" with increasing order. For piecewise polynomials it can be shown that their approximation error additionally depends on the size of their partitions. Smaller partitions cause smaller errors and the power of this error reduction on the other hand depends on the polynomial order. As the partitions or pieces of the piecewise polynomials correspond to the finite elements discretising the solid, refinement of this finite element mesh is used to reduce the approximation error. This increases the quality and accuracy of the finite element solution.

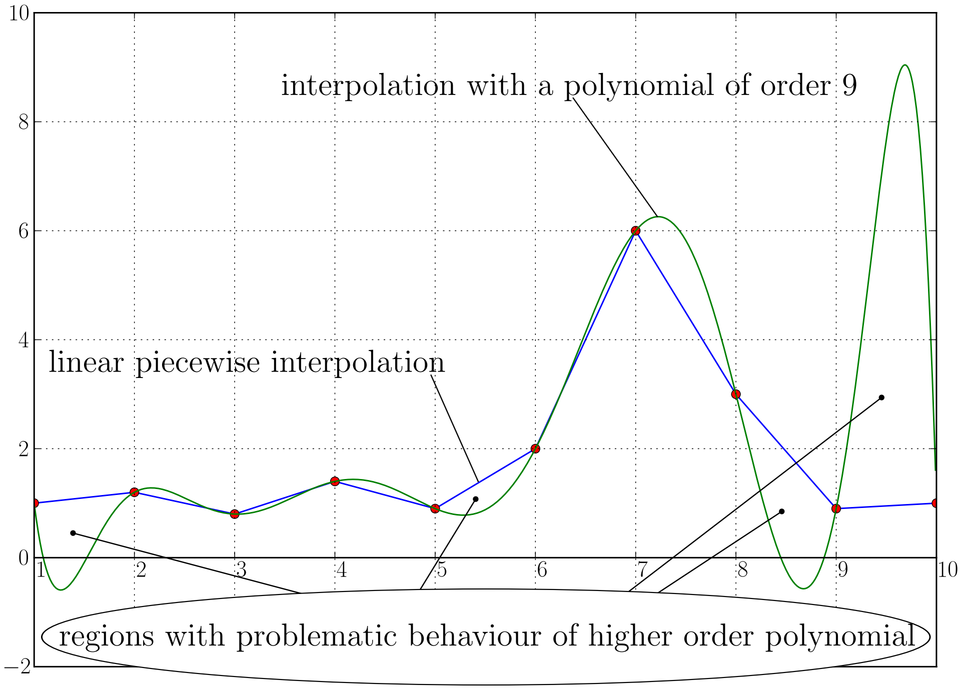

Figure 2.3.2. Comparison of piecewise linear (blue lines) and global polynomial (green curve) interpolation. Both of them fulfilling the interpolation condition.

2.3.2. Lagrange Polynomials

Looking for a way to determine interpolation polynomials for a given set of data without having to solve a system of equations each time, Joseph Louis Lagrange found that the interpolation polynomial $p(x)$ for an arbitrary function $f(x)$ with interpolation points x_i can be given as,

$$p(x) = \sum_{i=1}^{n} f(x_i) \cdot \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \tag{2.3.1}$$

The products

$$l_i^{n-1}(x) = \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \tag{2.3.2}$$

are known as Lagrange basis polynomials of order $n-1$ for the interpolation point x_i. With,

$$l_i^{n-1}(x_k) = \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x_k - x_j}{x_i - x_j} = \begin{matrix} 0 & \text{for } k \neq i \text{ and} \\ 1 & \text{for } k = i \end{matrix} \tag{2.3.3}$$

it can be proven, that $p(x_i)$ fulfils the interpolation condition:

$$f(x_i) = p(x_i) = \sum_{i=1}^{n} f(x_i) \cdot \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x_i - x_j}{x_i - x_j} = f(x_i) \cdot l_i^{n-1}(x_i) = f(x_i) \cdot 1 \tag{2.3.4}$$

The form of interpolation polynomials presented can be interpreted as a polynomial of order $n-1$ in an n-dimensional (polynomial) space, of the basis polynomials which are of order $n-1$ each and with the function values at the interpolation points as components.

Linear and quadratic Lagrange basis functions for 1d domains are depicted in Figure 2.3.3.

2.4. Isoparametric Finite Element Method

In this work isoparametric finite elements will be used. Isoparametric means that displacements and geometry are interpolated in the same way (Bathe 1996).

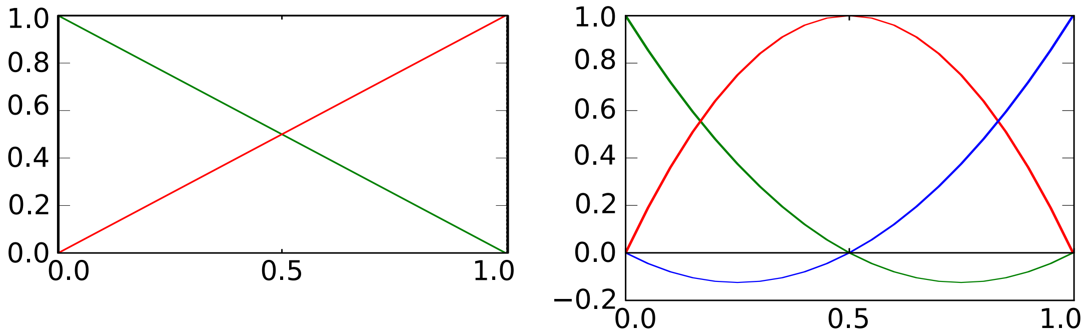

Figure 2.3.3. Linear and quadratic Lagrange basis (1d)

2.4.1. Reference Element

In finite element analysis, the computation of integrals is usually performed by numeric integration methods. Therefore it is beneficial to map every finite element of the "original" domain on a reference element where the integration takes place. This reference element is a normalised form of the same shape type as the concrete elements. The coordinates used in these elements are natural coordinates. Figure 2.4.1 shows an example for the mapping of a triangular element.

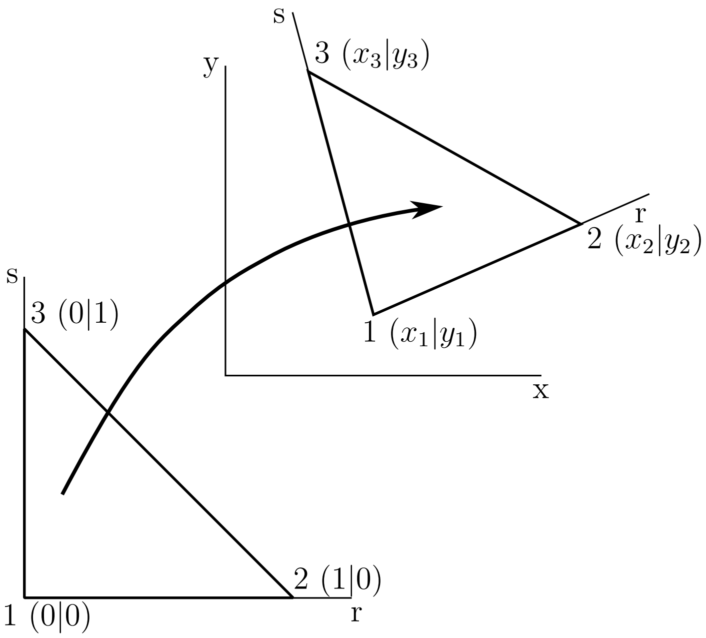

Figure 2.4.1. Triangular reference element and general triangular element

Figure 2.4.2 shows the mapping of a bilinear quadrilateral element.

The mapping itself is realised through geometry interpolation as discussed in the follow-

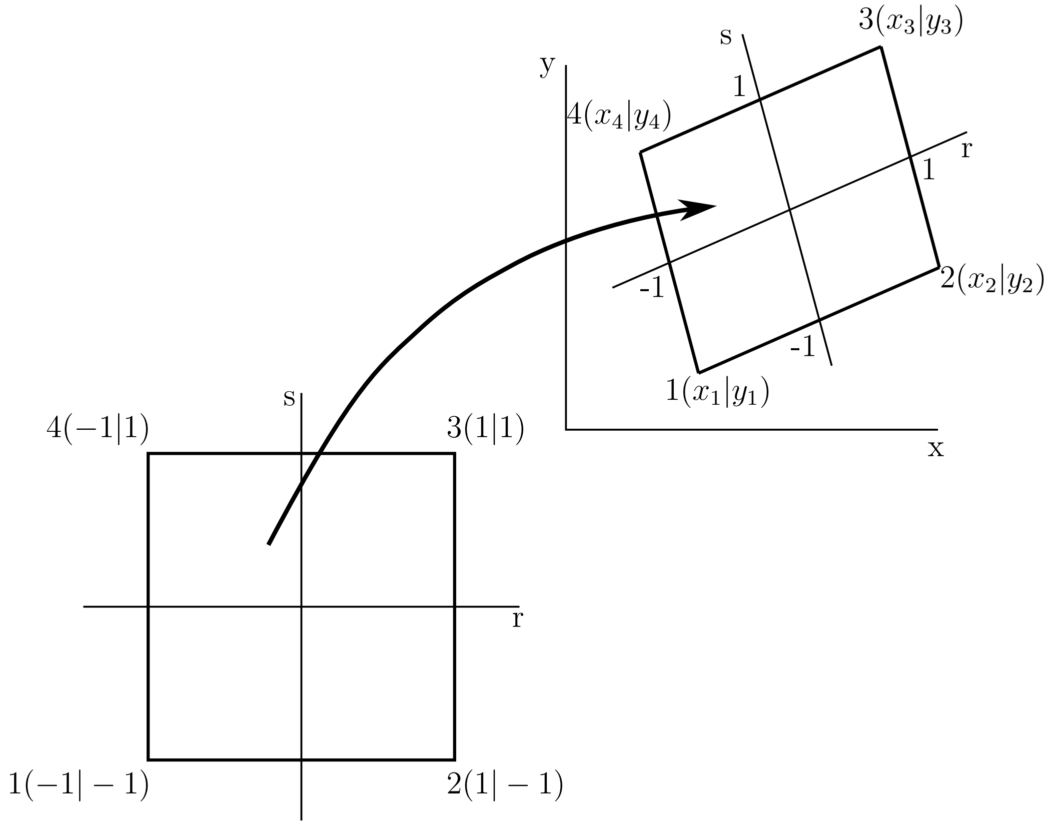

Figure 2.4.2. Quadrilateral reference element and general quadrilateral element

ing section.

2.4.2. Interpolation of Geometry and Displacements

General approach to interpolation of geometry

$$x_i(\boldsymbol{r}) = \sum_{j=1}^{n} \hat{x}_{ij}\, h_j(\boldsymbol{r}) \tag{2.4.1}$$

with

$x_i(\boldsymbol{r})$ i-th component of coordinates $\boldsymbol{x}$ at natural coordinates $\boldsymbol{r}$ with $i = 1, \ldots, \dim$

$h_j(\boldsymbol{r})$ shape function of node j evaluated at natural coordinates $\boldsymbol{r}$ with $j = 1, \ldots, n$ (n is the number of nodes)

$\hat{x}_{ij}$ i-th component of material or spatial coordinates $\hat{\boldsymbol{x}}$ of node j

General approach to interpolation of displacements

$$u_i(\boldsymbol{r}) = \sum_{j=1}^{n} h_j(\boldsymbol{r})\hat{u}_{ij} \tag{2.4.2}$$

with

$u_i(\boldsymbol{r})$ i-th component of displacement $\boldsymbol{u}$ at natural coordinates $\boldsymbol{r}$ with $i = 1, \dots, \mathrm{dim}$

$\hat{u}_{ij}$ i-th component of displacement $\hat{\boldsymbol{u}}$ of node j

Shape functions Comparing Lagrange interpolation and the interpolation methods presented above, the similarity between both methods is obvious and the only question is, how to obtain shape functions from the Lagrangian basis polynomials. Fortunately, the shape functions for standard shapes like quadrilaterals and triangles are documented in many textbooks, such as (Zienkiewicz and Taylor 2000).

For quadrilateral shape functions they are given as the tensor product of Lagrange polynomials,

$$\tilde{h}^n_{[k,l]}(r,s) = l^n_k(r) \cdot l^n_l(s) \tag{2.4.3}$$

where $\tilde{h}^n_{[k,l]}(r,s)$ is the shape function for the node with the local node indices k for coordinate r and l for coordinate s. The mapping between local node numbers and local node indices is established by appropriate adjacency tables, i.e. $\tilde{h}^n_{[1,0]}(r,s) = h_2(r,s)$.

For triangular shape functions, things are a little more complicated:

$$\tilde{h}^n_{[k,l,m]}(L_1,L_2,L_3) = l^k_k(L_1) \cdot l^l_l(L_2) \cdot l^m_m(L_3) \tag{2.4.4}$$

where natural coordinates are replaced by area coordinates. The area coordinates themselves can be expressed in terms of natural coordinates,

$$L_1 = 1 - r - s \tag{2.4.5a}$$
$$L_2 = r \tag{2.4.5b}$$
$$L_3 = s \tag{2.4.5c}$$
$$\tag{2.4.5d}$$

2.4.3. Coordinate Transformation for Integration on Reference Element

To evaluate the integral of a function $\boldsymbol{Q}(\boldsymbol{X})$ over a single finite element m in a standardised way, the computation is performed on the reference element. Therefore, a change of variables is used to write the integral in terms of the (natural) coordinates of the reference element,

$$\int\limits_{V^{(m)}} \boldsymbol{Q}(\boldsymbol{X})\mathrm{d}V^{(m)} = \int\limits_{\Omega} \boldsymbol{Q}[\boldsymbol{X}(\boldsymbol{r})]\ \det[\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{r}}]\mathrm{d}\Omega \tag{2.4.6}$$

2.4.4. Numerical Integration

The standard method for computing integrals in isoparametric finite element analysis is numerical integration with Gaussian quadrature. With precomputed integration points (or Gauss points) ${}_a r$ it is possible to compute integrals as the sum over the function values at the integration points (labelled $a = 1 \dots q$) multiplied with an integration weight ${}_a w$.

$$\int\limits_{\Omega} \boldsymbol{Q}[\boldsymbol{X}(\boldsymbol{r})]\ \det[(\boldsymbol{J}(\boldsymbol{r})]\,\mathrm{d}\Omega \doteq \sum_{a=1}^{q} \boldsymbol{Q}[\boldsymbol{X}({}_a\boldsymbol{r})]\ \det[\boldsymbol{J}({}_a\boldsymbol{r})]\,{}_a w \tag{2.4.7}$$

Although this method is not exact in general, it is for polynomials of order $2q-1$. However, if the integrand is not polynomial and does not have polynomial character, the integration error can be significant, e.g. if the integration point is near to a singularity. The integration points in natural coordinates and integration weights for Gaussian quadrature are listed in tables, at least for common integration domains like straight lines, squares, triangles, cubes and tetrahedra etc. (Zienkiewicz and Taylor 2000).

2.5. Finite Element Formulation of Spatial Equilibrium Equation

After introducing FEM basis techniques, they should be applied to the spatial equilibrium equation as given in equation (1.4.12). Partitioning the overall material domain V_0 in finite elements m with volume $V^{(m)}$ yields,

$$\int\limits_{V_0} \delta_{\boldsymbol{x}}\boldsymbol{F} : \boldsymbol{P}\,\mathrm{d}V_0 = \underset{m}{\mathsf{A}} \int\limits_{V_0^{(m)}} \delta_{\boldsymbol{x}}F_{ij}\,P_{ij}\,\mathrm{d}V_0^{(m)} = 0 \tag{2.5.1}$$

Because elements are coupled with their neighbour elements, equation (2.5.1) is not a sum of equations but a system of equations and $\underset{m}{\mathsf{A}}$ indicates an assembling procedure respecting the interdependencies between elements. Using equations (2.4.7), equation

(2.5.1) with a focus on a single element (m) can be written as,

$$\int\limits_{V_0^{(m)}} \delta_{\boldsymbol{x}} F_{ij}\, P_{ij}\, \mathrm{d}V_0^{(m)} \doteq \sum_{a=1}^{q} {}_a\left(\delta_{\boldsymbol{x}} F_{ik}\right)\, {}_aP_{ij}\, \det({}_a\boldsymbol{J})\, {}_aw = 0 \tag{2.5.2}$$

Since the first Piola-Kirchhoff tensor depends on $\boldsymbol{F}$, and $\boldsymbol{P} = \boldsymbol{P}(\boldsymbol{F})$ is a nonlinear function, equation (2.5.1) is nonlinear too and $\boldsymbol{P}(\boldsymbol{F})$ has to be linearised:

$$P_{ij}(\boldsymbol{F} + \varDelta\boldsymbol{F}) = P_{ij}\big|_{\boldsymbol{F}} + \left.\frac{\partial P_{ij}}{\partial F_{kl}}\right|_{\boldsymbol{F}} \varDelta F_{kl} + \mathcal{O}\left[(\varDelta\boldsymbol{F})^2\right] \doteq P_{ij}\big|_{\boldsymbol{F}} + \left.\frac{\partial P_{ij}}{\partial F_{kl}}\right|_{\boldsymbol{F}} \varDelta F_{kl} \tag{2.5.3}$$

Beginning with the deformation gradient in terms of nodal values,

$$\begin{aligned} F_{ij} = \frac{\partial \varphi_i}{\partial X_j} = \frac{\partial x_i}{\partial X_j} = \frac{\partial\left(X_i + u_i\right)}{\partial X_j} &= \delta_{ij} + \frac{\partial u_i}{\partial X_j} \\ &= \delta_{ij} + J_{kl}^{(h)}\, J_{lj}^{-1}\, \hat{u}_{ik} \end{aligned} \tag{2.5.4}$$

the virtual deformation gradient in terms of virtual nodal displacements is obtained straight-forwardly:

$$\begin{aligned} \delta_{\boldsymbol{x}} F_{ij} = \frac{\partial \delta_{\boldsymbol{x}} \varphi_i}{\partial X_j} = \frac{\partial \delta_{\boldsymbol{x}} \hat{u}_{ik}\, h_k}{\partial X_j} &= \frac{\partial h_k}{\partial X_j}\, \delta_{\boldsymbol{x}} \hat{u}_{ik} \\ &= J_{kl}^{(h)}\, J_{lj}^{-1}\, \delta_{\boldsymbol{x}} \hat{u}_{ik} \end{aligned} \tag{2.5.5}$$

The same procedure leads to the incremental deformation gradient in terms of incremental spatial displacements,

$$\varDelta_{\boldsymbol{x}} F_{ij} = J_{kl}^{(h)}\, J_{lj}^{-1}\, \varDelta_{\boldsymbol{x}} \hat{u}_{ik} \tag{2.5.6}$$

In above equations some Jacobians have been introduced, they are given as,

$$J_{kl}^{(h)} = \frac{\partial h_k}{\partial r_l} \tag{2.5.7}$$

$$J_{ik} = \frac{\partial \hat{X}_{ij}\, h_j}{\partial r_k} = J_{jk}^{(h)}\, \hat{X}_{ij} \tag{2.5.8}$$

where (2.5.8) is the Jacobian of the interpolation of nodal material coordinates $\hat{\boldsymbol{X}}$ which is used to compute $\boldsymbol{J}^{-1}$.

With

$$\mathbb{J}_{ij} = J_{ik}^{(h)}\, J_{kj}^{-1} \tag{2.5.9}$$

equation (2.5.2) reads now as,

$$\int\limits_{V_0^{(m)}} \delta_{\boldsymbol{x}} F_{ij}\, P_{ij}\, \mathrm{d}V_0^{(m)} \doteq \sum_{a=1}^{q} \delta_{\boldsymbol{x}} \hat{u}_{ik}\, {}_a\mathbb{J}_{kj} \left(P_{ij}\big|_{{}_a\boldsymbol{F}} + \left.\frac{\partial P_{ij}}{\partial F_{lm}}\right|_{{}_a\boldsymbol{F}} {}_a\mathbb{J}_{nm}\, \varDelta_{\boldsymbol{x}} \hat{u}_{ln} \right) {}_aw = 0 \tag{2.5.10}$$

Remembering that the equilibrium equation must hold for all allowed variations, a system of equations for incremental nodal displacements $\varDelta_{\boldsymbol{x}} \hat{\boldsymbol{u}}$ is obtained:

$$\begin{aligned} \left(\sum_{a}^{q} {}_a\mathbb{J}_{kj} \left. \frac{\partial P_{ij}}{\partial F_{lm}} \right|_{{}_a\boldsymbol{F}} {}_a\mathbb{J}_{nm}\, {}_a w \right) \varDelta_{\boldsymbol{x}} \hat{u}_{ln} &= - \left(\sum_{a=1}^{q} {}_a\mathbb{J}_{kj} \left. P_{ij} \right|_{{}_a\boldsymbol{F}} {}_a w \right) \\ K^{(m)}_{ikln} \varDelta_{\boldsymbol{x}} \hat{u}^{(m)}_{ln} &= R^{(m)}_{ik} \end{aligned} \tag{2.5.11}$$

where $\boldsymbol{K}^{(m)}$ is the element stiffness matrix, $\varDelta_{\boldsymbol{x}} \hat{\boldsymbol{u}}^{(m)}$ are the element's nodal displacement increments and $\boldsymbol{R}^{(m)}$ is the element residual. These element-level equations have to be fulfilled on each element simultaneously which is realised by an assembling operation (indicated by $\underset{m}{\mathsf{A}}$) that builds a global system of equations,

$$\begin{aligned} \boldsymbol{K} &= \underset{m}{\mathsf{A}} \boldsymbol{K}^{(m)} \\ \boldsymbol{R} &= \underset{m}{\mathsf{A}} \boldsymbol{R}^{(m)} \\ \boldsymbol{K} \varDelta_{\boldsymbol{x}} \hat{\boldsymbol{u}} &= \boldsymbol{R} \end{aligned}$$

Details of an exemplary assembling scheme are given in the next section.

2.5.1. Assembling the Global System of Equations

To obtain a system of linear equations in classic matrix-vector form from equation (2.5.11), it has to be assembled, indicated by $\underset{m}{\mathsf{A}}$. The assembling map used in this work is quite simple and examples for two and four dimensional objects are presented below:

$$R_I \hat{=} R^{(m)}_{ij}, \text{ with } I = i + dof \cdot (n_j - 1) \tag{2.5.12}$$

$$K_{IJ} \hat{=} K^{(m)}_{ijkl}, \text{ with } I = i + dof \cdot (n_j - 1), J = k + dof \cdot (n_l - 1) \tag{2.5.13}$$

In equations (2.5.12) and (2.5.13) the first and third indices indicate the coordinate or degree of freedom, e.g. $0 \hat{=} x, 1 \hat{=} y$, the second and fourth indices of the unassembled entities indicate the local node number (node number within the element m), n_j is the global node number of the local node with local number j and dof is the number of degrees of freedom of each node[1].

Finally, the resulting system is written as,

$$\boldsymbol{K} \cdot \varDelta_{\boldsymbol{x}} \hat{\boldsymbol{u}} = \boldsymbol{R} \tag{2.5.14}$$

[1] This assembling scheme only works if each node has the same number of degrees of freedom. Furthermore, the scheme is for 1 as the first index of arrays, node numbering etc. In cases where array indices start at 0, the $(n_{j/l} - 1)$ changes to $n_{j/l}$.

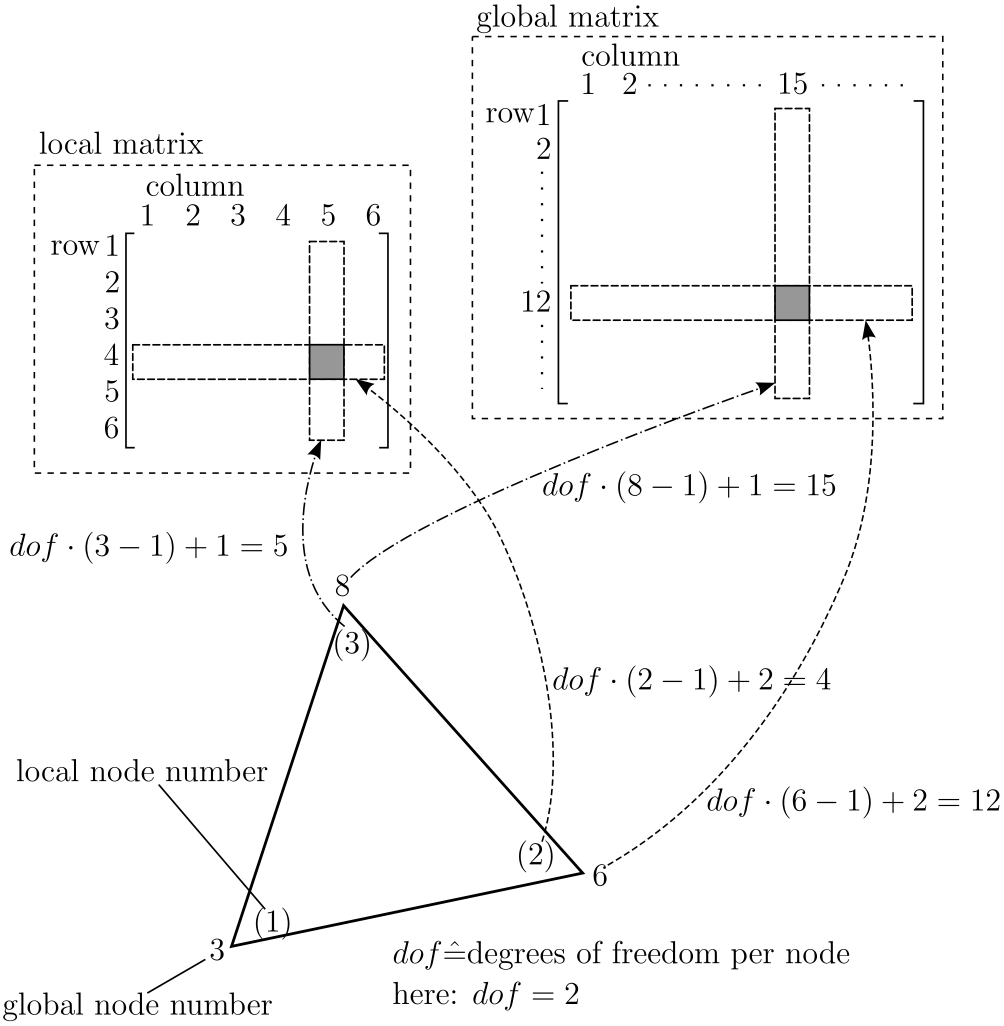

Figure 2.5.1. Assembly of matrix entry corresponding to the second coordinate in local node (1) and to the first coordinate in local node (2)

2.5.2. Integration of Dirichlet Boundary Conditions

If the value of the quantity sought, e.g. displacements, is known at some point, this is called a Dirichlet or necessary boundary condition. In mechanics such boundary conditions may be introduced by infinitely stiff support, where displacements are zero a priori. Another source of Dirichlet boundary conditions is the exploitation of a mechanical system's symmetry. In displacement driven analysis, "load" application is also performed by Dirichlet boundary conditions, whereby in this case the prescribed displacements are different from zero. A proper way to impose this kind of boundary conditions is given in (Wu et al. 2008).

2.6. Nonlinear Problem

In case of a significant magnitude of nonlinearity in equation (2.5.1) simple linearisation and solution of the resulting linear system of equations is not sufficient to obtain high quality results. One approach to the solution of nonlinear systems of equations is the Newton-Raphson method. The concept of this method has already been introduced in Section 1.5.1. Below, the Newton-Raphson method will be written in terms of FEM expressions.

2.6.1. The Newton-Raphson Algorithm Applied in Finite Element Analysis

Comparing equations (1.5.1) and (2.5.14) we find that,

$$\left.\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}\right|_0 \hat{=} \left.\boldsymbol{K}\right|_0 \tag{2.6.1a}$$

$$\Delta \boldsymbol{x}^0 \hat{=} \left.\Delta_{\boldsymbol{x}} \hat{\boldsymbol{u}}\right|_0 \tag{2.6.1b}$$

$$\boldsymbol{F}(\boldsymbol{x}^0) \hat{=} \left.\boldsymbol{R}\right|_0 \tag{2.6.1c}$$

where $|_0$ indicates terms from the linearisation at the initial state. The Newton-Raphson algorithm implemented for the pure spatial motion problem, as it has been implemented for this work, is listed in Table 2.1.

Table 2.1. Newton-Raphson Algorithm for Finite Element Method

```
set CONVERGED = False
choose initial guess $\hat{\boldsymbol{u}}_0 = \mathbf{0}$, set iteration counter $n = 0$
while CONVERGED = False do
    compute deformation gradient $\boldsymbol{F}(\boldsymbol{x})|_n$ for each integration point of each element
    compute element right hand side $\boldsymbol{R}^{(m)}(\boldsymbol{F})\Big|_n$ and element stiffness matrix $\boldsymbol{K}^{(m)}(\boldsymbol{F})\Big|_n$
    assemble global right hand side $\boldsymbol{R}(\boldsymbol{F})|_n$ and stiffness matrix $\boldsymbol{K}(\boldsymbol{F})|_n$
    integrate Dirichlet boundary conditions $\mathcal{BC}^n$ as proposed in (Wu et al. 2008)[2]
    Solve $\boldsymbol{K}(\boldsymbol{F})|_n \; \Delta_{\boldsymbol{x}}\hat{\boldsymbol{u}}|_n = \boldsymbol{R}(\boldsymbol{F})|_n$
    update nodal data and integration points: Coordinates $\hat{\boldsymbol{x}}^{n+1} = \hat{\boldsymbol{x}}^n + \Delta_{\boldsymbol{x}}\hat{\boldsymbol{u}}|_n$, Jacobians, ...
    check convergence with an appropriate criterion, e.g. absolute value of "unbalanced energy" $\epsilon^n = |\,\Delta_{\boldsymbol{x}}\hat{\boldsymbol{u}}|_n \cdot \boldsymbol{R}(\boldsymbol{F})|_n|$
    if $\epsilon^n \leq$ tol then
        set CONVERGED = True
    else
        increment iteration counter $n = n + 1$
    end if
end while
write out the results, postprocessing
```

[2] usually, for $n = 0$ the "driving" displacement is the boundary condition, if $n > 0$ all nodes with Dirichlet boundary condition have $\Delta\hat{\boldsymbol{u}}^n = \mathbf{0}$, since after the initial displacements there are no more increments in their displacement.

3. r-Adaptivity

In chapter 2 the FEM was applied to the spatial motion problem only. R-adaptivity means the solution of the spatial motion and material motion problem with the aim of further minimising the total energy resulting in a smaller error. The nature of the r-adaptivity problem contains several numerical difficulties as highlighted in (Askes et al. 2004) and (Mosler and Ortiz 2006). The r-adaptivity problem is not strictly convex in contrast to the classic FEM approach to the pure spatial motion problem. Furthermore the resulting systems may contain singularities differing from one iteration to another (Askes et al. 2004). These difficulties motivate different regularisation approaches and solution strategies, which will be compared by means of example problems. Beside the class of strategies based on the already introduced concept of linearisation and application of the Newton-Raphson scheme there exist other methods to solve nonlinear problems that sometimes do not require any linearisation in the classic sense at all, e.g. nonlinear conjugate gradient type methods used in (Thoutireddy 2003). However, this work is restricted to the former class of strategies.

3.1. Linearisation of Spatial and Material Motion Quantities

Seeking stationary points of the total energy $\varPi$, the starting point to r-adaptivity is,

$$\delta\varPi = \delta_{\boldsymbol{x}}\varPi + \delta_{\boldsymbol{X}}\varPi = 0 \tag{3.1.1}$$

the full linearisation consists of four linearised terms,

$$\begin{aligned}\delta\varPi(\boldsymbol{x}+\varDelta\boldsymbol{x},\boldsymbol{X}+\varDelta\boldsymbol{X}) &\doteq \delta\varPi(\boldsymbol{x},\boldsymbol{X}) + \varDelta\delta\varPi \\ &= \delta\varPi(\boldsymbol{x},\boldsymbol{X}) + \varDelta_{\boldsymbol{x}}\delta_{\boldsymbol{x}}\varPi + \varDelta_{\boldsymbol{x}}\delta_{\boldsymbol{X}}\varPi + \varDelta_{\boldsymbol{X}}\delta_{\boldsymbol{x}}\varPi + \varDelta_{\boldsymbol{X}}\delta_{\boldsymbol{X}}\varPi\end{aligned} \tag{3.1.2}$$

Starting from equation (1.3.10),

$$\begin{aligned}\mathit{\Delta}_{\boldsymbol{x}}\delta_{\boldsymbol{x}}\mathit{\Pi} &= \int\limits_{V_0} \delta_{\boldsymbol{x}}F_{ij}\, \mathit{\Delta}_{\boldsymbol{x}}P_{ij}\, \mathrm{d}V_0 \\ &= \int\limits_{V_0} \delta_{\boldsymbol{x}}F_{ij}\, (\partial_{\boldsymbol{F}}\boldsymbol{P})_{ijkl}\, \mathit{\Delta}_{\boldsymbol{x}}F_{kl}\, \mathrm{d}V_0\end{aligned}$$

where,

$$(\partial_{\boldsymbol{F}}\boldsymbol{P})_{ijkl} = \frac{\partial P_{ij}}{\partial F_{kl}} \tag{3.1.3}$$

with $\boldsymbol{P}$ expressed for the spatial motion problem. Using,

$$P_{ij} = \frac{\partial \mathit{\Psi}_0^{int}}{\partial F_{ij}} \tag{3.1.4}$$

$$\delta_{\boldsymbol{x}}\boldsymbol{\varphi} = \delta\boldsymbol{\varphi}^* \tag{3.1.5}$$

$$\delta_{\boldsymbol{x}}\boldsymbol{F} = \frac{\partial \delta\boldsymbol{\varphi}^*}{\partial \boldsymbol{X}} \tag{3.1.6}$$

above equation can be written as,

$$\mathit{\Delta}_{\boldsymbol{x}}\delta_{\boldsymbol{x}}\mathit{\Pi} = \int\limits_{V_0} \frac{\partial \delta\varphi_i^*}{\partial X_j}\, \frac{\partial^2 \mathit{\Psi}_0^{int}}{\partial F_{kl}\partial F_{ij}}\, \frac{\partial \mathit{\Delta}\varphi_k^*}{\partial X_l}\, \mathrm{d}V_0 \tag{3.1.7}$$

The linearisation with respect to material coordinates $\boldsymbol{X}$ of the variation with respect to spatial coordinates $\boldsymbol{x}$ is done starting from equation (1.3.11), since this avoids the linearisation of the infinitesimal volume dV_0.

$$\begin{aligned}\mathit{\Delta}_{\boldsymbol{X}}\delta_{\boldsymbol{x}}\mathit{\Pi} &= \int\limits_{V_t} \frac{\partial \delta_{\boldsymbol{x}}\varphi_i}{\partial x_j}\, \mathit{\Delta}_{\boldsymbol{X}}\sigma_{ij}\, \mathrm{d}V_t \\ &= \int\limits_{V_t} \frac{\partial \delta\varphi_i^*}{\partial x_j}\, \frac{\partial \sigma_{ij}}{\partial f_{kl}}\, \mathit{\Delta}_{\boldsymbol{X}}f_{kl}\, \mathrm{d}V_t \\ &= \int\limits_{V_t} \frac{\partial \delta\varphi_i^*}{\partial X_m} F_{mj}^{-1}\, \frac{\partial \sigma_{ij}}{\partial f_{kl}}\, \frac{\partial \mathit{\Delta}\mathit{\Phi}_{*k}}{\partial x_l}\, \mathrm{d}V_t\end{aligned} \tag{3.1.8}$$

As shown in Section 1.3.1 Cauchy stresses $\boldsymbol{\sigma}$ can be written as,

$$\sigma_{ij} = \mathit{\Psi}_t^{int}\, \delta_{ij} - f_{ki}\, \frac{\partial \mathit{\Psi}_t^{int}}{\partial f_{kj}} \tag{3.1.9}$$

Inserting equation (3.1.9) with index k replaced by n, equation (3.1.8) becomes,

$$\mathit{\Delta}_{\boldsymbol{X}}\delta_{\boldsymbol{x}}\mathit{\Pi} = \int\limits_{V_t} \frac{\partial \delta\varphi_i^*}{\partial X_m} F_{mj}^{-1} \left(\delta_{ij}\, \frac{\partial \mathit{\Psi}_t^{int}}{\partial f_{kl}} - \frac{\partial f_{ni}}{\partial f_{kl}}\, \frac{\partial \mathit{\Psi}_t^{int}}{\partial f_{nj}} - f_{ni}\, \frac{\partial^2 \mathit{\Psi}_t^{int}}{\partial f_{kl}\partial f_{nj}} \right) \frac{\partial \mathit{\Delta}\mathit{\Phi}_{*k}}{\partial x_l}\, \mathrm{d}V_t \tag{3.1.10}$$

The next step is the linearisation with respect to $\boldsymbol{x}$ of the variation with respect to material coordinates $\boldsymbol{X}$:

$$\begin{aligned}
\varDelta_{\boldsymbol{x}}\delta_{\boldsymbol{X}}\varPi &= \int\limits_{V_0} \frac{\partial \delta_{\boldsymbol{X}}\varPhi_i}{\partial X_j}\, \varDelta_{\boldsymbol{x}}\varSigma_{ij}\, \mathrm{d}V_0 \\
&= \int\limits_{V_0} \frac{\partial \delta\varPhi_{*i}}{\partial X_j}\, \frac{\partial \varSigma_{ij}}{\partial F_{kl}}\, \varDelta_{\boldsymbol{x}}F_{kl}\mathrm{d}V_0 \\
&= \int\limits_{V_0} \frac{\partial \delta\varPhi_{*i}}{\partial x_m} f^{-1}_{mj}\, \frac{\partial \varSigma_{ij}}{\partial F_{kl}}\, \frac{\partial \varDelta\varphi^*_k}{\partial X_l}\mathrm{d}V_0
\end{aligned} \tag{3.1.11}$$

Again, expressing the Eshelby stress tensor as a derivation of the internal potential energy using equation (1.4.8), equation (3.1.11) can be written in terms of the internal energy:

$$\varDelta_{\boldsymbol{x}}\delta_{\boldsymbol{X}}\varPi = \int\limits_{V_0} \frac{\partial \delta\varPhi_{*i}}{\partial x_m} f^{-1}_{mj} \left(\delta_{ij}\frac{\partial \varPsi^{int}_0}{\partial F_{kl}} - \frac{\partial F_{ni}}{\partial F_{kl}}\,\frac{\partial \varPsi^{int}_0}{\partial F_{nj}} - F_{ni}\,\frac{\partial^2 \varPsi^{int}_0}{\partial F_{kl}\partial F_{nj}}\right) \frac{\partial \varDelta\varphi^*_k}{\partial X_l}\mathrm{d}V_0 \tag{3.1.12}$$

Finally, linearisation of variation, both with respect to material coordinates $\boldsymbol{X}$ is obtained in a similar manner,

$$\begin{aligned}
\varDelta_{\boldsymbol{X}}\delta_{\boldsymbol{X}}\varPi &= \int\limits_{V_t} \delta_{\boldsymbol{X}} f_{ij}\, \varDelta_{\boldsymbol{X}} p_{ij}\, \mathrm{d}V_t \\
&= \int\limits_{V_t} \delta_{\boldsymbol{X}} f_{ij}\, \frac{\partial p_{ij}}{\partial f_{kl}}\, \varDelta_{\boldsymbol{X}} f_{kl}\, \mathrm{d}V_t \\
&= \int\limits_{V_t} \frac{\partial \delta\varPhi_{*i}}{\partial x_j}\, \frac{\partial p_{ij}}{\partial f_{kl}}\, \frac{\partial \varDelta\varPhi_{*k}}{\partial x_l}\, \mathrm{d}V_t
\end{aligned} \tag{3.1.13}$$

and using

$$p_{ij} = \frac{\partial \varPsi^{int}_t}{\partial f_{ij}} \tag{3.1.14}$$

yields,

$$\varDelta_{\boldsymbol{X}}\delta_{\boldsymbol{X}}\varPi = \int\limits_{V_t} \frac{\partial \delta\varPhi_{*i}}{\partial x_j}\, \frac{\partial^2 \varPsi^{int}_t}{\partial f_{kl}\partial f_{ij}}\, \frac{\partial \varDelta\varPhi_{*k}}{\partial x_l}\, \mathrm{d}V_t \tag{3.1.15}$$

Remembering,

$$\begin{aligned}
\delta_{\boldsymbol{x}} F_{ij} &= \frac{\partial \delta\varphi^*_i}{\partial X_j} \\
\delta_{\boldsymbol{X}} f_{ij} &= \frac{\partial \delta\varPhi_{*i}}{\partial x_j}
\end{aligned}$$

the introduced linearisations fit into a matrix scheme as given below,

$$\Delta\delta\mathit{\Pi} \hat{=} \int\limits_V \begin{bmatrix} \delta_{\boldsymbol{x}} F_{ij} & \delta_{\boldsymbol{X}} f_{ij} \end{bmatrix} \cdot \begin{bmatrix} \tilde{A}_{ijkl} & F^{-1}_{jm} \tilde{b}_{imkl} \\ f^{-1}_{jm} \tilde{B}_{imkl} & \tilde{a}_{ijkl} \end{bmatrix} \cdot \begin{bmatrix} \Delta_{\boldsymbol{x}} F_{kl} \\ \Delta_{\boldsymbol{X}} f_{kl} \end{bmatrix} \mathrm{d}V \tag{3.1.17}$$

with,

$$\tilde{A}_{ijkl} = \frac{\partial^2 \mathit{\Psi}_0^{int}}{\partial F_{kl} \partial F_{ij}} \tag{3.1.18a}$$

$$\tilde{b}_{ijkl} = \delta_{ij} \frac{\partial \mathit{\Psi}_t^{int}}{\partial f_{kl}} - \frac{\partial f_{mi}}{\partial f_{kl}} \frac{\partial \mathit{\Psi}_t^{int}}{\partial f_{mj}} - f_{mi} \frac{\partial^2 \mathit{\Psi}_t^{int}}{\partial f_{kl} \partial f_{mj}} \tag{3.1.18b}$$

$$\tilde{B}_{ijkl} = \delta_{ij} \frac{\partial \mathit{\Psi}_0^{int}}{\partial F_{kl}} - \frac{\partial F_{mi}}{\partial F_{kl}} \frac{\partial \mathit{\Psi}_0^{int}}{\partial F_{mj}} - F_{mi} \frac{\partial^2 \mathit{\Psi}_0^{int}}{\partial F_{kl} \partial F_{mj}} \tag{3.1.18c}$$

$$\tilde{a}_{ijkl} = \frac{\partial^2 \mathit{\Psi}_t^{int}}{\partial f_{kl} \partial f_{ij}} \tag{3.1.18d}$$

Note that in the scheme from equation (3.1.17) the integration domains are not specified, and therefore some additional operations presented in the next section are necessary to obtain a scheme of practical use.

3.1.1. Discretised r-Adaptivity Equations

Now, to obtain these linearisations in terms of virtual and incremental displacements $\delta\varphi^*, \delta\mathit{\Phi}_*$ and $\Delta\varphi^*, \Delta\mathit{\Phi}_*$, the FEM formulations of the material motion and spatial motion deformation gradients is used (see Section 2.5, equation (2.5.4)) and the following for the derivation for δF_{ij}):

$$\delta_{\boldsymbol{x}} F_{ij} = J^{(h)}_{kl} \, J^{-1}_{lj} \delta \hat{\varphi}^*_{ik} \tag{3.1.19a}$$

$$\delta_{\boldsymbol{X}} f_{ij} = J^{(h)}_{kl} \, j^{-1}_{lj} \delta \hat{\mathit{\Phi}}_{*ik} \tag{3.1.19b}$$

where,

$$j_{ij} = J^{(h)}_{kj} \, \hat{x}_{ik} \tag{3.1.20}$$

is similar to equation (2.5.8). Combining the two Jacobians in each formulation, the variations and increments read as,

$$\delta_{\boldsymbol{x}} F_{ij} = \mathbb{J}_{kj} \delta_{\boldsymbol{x}} \hat{\varphi}^*_{ik} \tag{3.1.21a}$$

$$\delta_{\boldsymbol{X}} f_{ij} = \mathbb{j}_{kj} \delta_{\boldsymbol{X}} \hat{\mathit{\Phi}}_{*ik} \tag{3.1.21b}$$

$$\Delta_{\boldsymbol{x}} F_{ij} = \mathbb{J}_{kj} \Delta_{\boldsymbol{x}} \hat{\varphi}^*_{ik} \tag{3.1.21c}$$

$$\Delta_{\boldsymbol{X}} f_{ij} = \mathbb{j}_{kj} \Delta_{\boldsymbol{X}} \hat{\mathit{\Phi}}_{*ik} \tag{3.1.21d}$$

Using these expressions in equation (3.1.17) and specifying integration domains yields,

$$\Delta\delta\Pi_h^{(m)} = \begin{bmatrix} \delta\hat{\varphi}^*_{in} & \delta\hat{\Phi}_{*in} \end{bmatrix} \cdot \begin{bmatrix} \int_{V_0^{(m)}} \mathbb{J}_{nj}\, \tilde{A}_{ijkl} \mathbb{J}_{ol}\, \mathrm{d}V_0 & \int_{V_t^{(m)}} \mathbb{J}_{nj}\, F^{-1}_{jm}\, \tilde{b}_{imkl}\, \mathbb{j}_{ol}\, \mathrm{d}V_t \\ \int_{V_0^{(m)}} \mathbb{j}_{nj}\, f^{-1}_{jm}\, \tilde{B}_{imkl}\, \mathbb{J}_{ol}\, \mathrm{d}V_0 & \int_{V_t^{(m)}} \mathbb{j}_{nj}\, \tilde{a}_{ijkl}\, \mathbb{j}_{ol}\, \mathrm{d}V_t \end{bmatrix} \cdot \begin{bmatrix} \Delta\hat{\varphi}^*_{ko} \\ \Delta\hat{\Phi}_{*ko} \end{bmatrix} \tag{3.1.22}$$

Now, applying,

$$\mathbb{J}_{ij} = \frac{\partial h_i}{\partial X_j} = \frac{\partial h_i}{\partial x_k}\frac{\partial x_k}{\partial X_j} = \frac{\partial h_i}{\partial x_k} F_{kj} = \mathbb{j}_{ik} F_{kj} \tag{3.1.23a}$$

$$\mathbb{j}_{ij} = \frac{\partial h_i}{\partial x_k} = \frac{\partial h_i}{\partial X_k} f_{kj} = \mathbb{J}_{ik} f_{kj} \tag{3.1.23b}$$

in equation (3.1.22) yields,

$$\Delta\delta\Pi_h^{(m)} = \begin{bmatrix} \delta\hat{\varphi}^*_{in} & \delta\hat{\Phi}_{*in} \end{bmatrix} \cdot \begin{bmatrix} \int_{V_0^{(m)}} \mathbb{J}_{nj}\, \tilde{A}_{ijkl} \mathbb{J}_{ol}\, \mathrm{d}V_0 & \int_{V_t^{(m)}} \mathbb{j}_{nj}\, \tilde{b}_{ijkl}\, \mathbb{j}_{ol}\, \mathrm{d}V_t \\ \int_{V_0^{(m)}} \mathbb{J}_{nj}\, \tilde{B}_{ijkl}\, \mathbb{J}_{ol}\, \mathrm{d}V_0 & \int_{V_t^{(m)}} \mathbb{j}_{nj}\, \tilde{a}_{ijkl}\, \mathbb{j}_{ol}\, \mathrm{d}V_t \end{bmatrix} \cdot \begin{bmatrix} \Delta\hat{\varphi}^*_{ko} \\ \Delta\hat{\Phi}_{*ko} \end{bmatrix} \tag{3.1.24}$$

3.1.2. General Newton Scheme for r-Adaptivity

Seeking a stationary point of the total potential energy can be expressed as,

$$\delta\Pi = 0 \tag{3.1.25}$$

or considering nonlinear behaviour and prepared for Newton-Raphson scheme:

$$0 = \delta\Pi\big|_n + \Delta\delta\Pi\big|_n \tag{3.1.26}$$

$$\Delta\delta\Pi\big|_n = -\,\delta\Pi\big|_n \tag{3.1.27}$$

Finally, inserting all variations (equilibrium equations (1.3.10) and (1.4.11)) and linearisations (equation (3.1.22)) in the above equation (3.1.27) gives:

$$\begin{bmatrix} \delta\hat{\varphi}^*_{in} & \delta\hat{\Phi}_{*in} \end{bmatrix} \cdot \begin{bmatrix} \int_{V_0^{(m)}} \mathbb{J}_{nj}\, \tilde{A}_{ijkl} \mathbb{J}_{oj}\, \mathrm{d}V_0 & \int_{V_t^{(m)}} \mathbb{J}_{nj}\, F^{-1}_{jm}\, \tilde{b}_{imkl}\, \mathbb{j}_{ol}\, \mathrm{d}V_t \\ \int_{V_0^{(m)}} \mathbb{j}_{nj}\, f^{-1}_{jm}\, \tilde{B}_{imkl}\, \mathbb{J}_{ol}\, \mathrm{d}V_0 & \int_{V_t^{(m)}} \mathbb{j}_{nj}\, \tilde{a}_{ijkl}\, \mathbb{j}_{ol}\, \mathrm{d}V_t \end{bmatrix}\Bigg|_n \cdot \begin{bmatrix} \Delta\hat{\varphi}^*_{ko} \\ \Delta\hat{\Phi}_{*ko} \end{bmatrix}\Bigg|_n = \\ - \begin{bmatrix} \delta\hat{\varphi}^*_{in} & \delta\hat{\Phi}_{*in} \end{bmatrix} \cdot \begin{bmatrix} \int_{V_0^{(m)}} \mathbb{J}_{nj} P_{ij}\, \mathrm{d}V_0 \\ \int_{V_t^{(m)}} \mathbb{j}_{nj} p_{ij}\, \mathrm{d}V_t \end{bmatrix}\Bigg|_n \tag{3.1.28}$$

Since these equations must hold for all allowed variations, one can write,

$$\left.\begin{bmatrix} \int_{V_0^{(m)}} \mathbb{J}_{nj}\, \tilde{A}_{ijkl} \mathbb{J}_{oj}\, \mathrm{d}V_0 & \int_{V_t^{(m)}} \mathbb{J}_{nj}\, F^{-1}_{jm}\, \tilde{b}_{imkl}\, \mathring{\mathbb{j}}_{ol}\, \mathrm{d}V_t \\ \int_{V_0^{(m)}} \mathring{\mathbb{j}}_{nj}\, f^{-1}_{jm}\, \tilde{B}_{imkl}\, \mathbb{J}_{ol}\, \mathrm{d}V_0 & \int_{V_t^{(m)}} \mathring{\mathbb{j}}_{nj}\, \tilde{a}_{ijkl}\, \mathring{\mathbb{j}}_{ol}\, \mathrm{d}V_t \end{bmatrix}\right|_n \cdot \left.\begin{bmatrix} \Delta\hat{\varphi}^*_{ko} \\ \Delta\hat{\Phi}_{*ko} \end{bmatrix}\right|_n = - \left.\begin{bmatrix} \int_{V_0^{(m)}} \mathbb{J}_{nj} P_{ij}\, \mathrm{d}V_0 \\ \int_{V_t^{(m)}} \mathring{\mathbb{j}}_{nj} p_{ij}\, \mathrm{d}V_t \end{bmatrix}\right|_n \qquad (3.1.29)$$

and obtain the fully linearised equations to the r-adaptivity problem, where $\Delta\hat{\boldsymbol{\varphi}}^*$ and $\Delta\hat{\boldsymbol{\Phi}}_*$ are the incremental spatial and material displacements calculated in each Newton-Raphson iteration.
Instead of using the equilibrium equations (1.3.10) and (1.4.11) one could also have used equations (1.3.11) and (1.4.12).

Symmetry: Since equation (3.1.29) is derived from a potential Π, the system matrix corresponding to second order derivatives,

$$\begin{bmatrix} \mathbb{A} & \mathbb{b} \\ \mathbb{B} & \mathbb{a} \end{bmatrix} = \begin{bmatrix} \int_{V_0^{(m)}} \mathbb{J}_{nj}\, \tilde{A}_{ijkl} \mathbb{J}_{ol}\, \mathrm{d}V_0 & \int_{V_t^{(m)}} \mathbb{J}_{nj}\, F^{-1}_{jm}\, \tilde{b}_{imkl}\, \mathring{\mathbb{j}}_{ol}\, \mathrm{d}V_t \\ \int_{V_0^{(m)}} \mathring{\mathbb{j}}_{nj}\, f^{-1}_{jm}\, \tilde{B}_{imkl}\, \mathbb{J}_{ol}\, \mathrm{d}V_0 & \int_{V_t^{(m)}} \mathring{\mathbb{j}}_{nj}\, \tilde{a}_{ijkl}\, \mathring{\mathbb{j}}_{ol}\, \mathrm{d}V_t \end{bmatrix} \qquad (3.1.30)$$

has to be symmetric and

$$\mathbb{b} = \mathbb{B}^T \qquad (3.1.31)$$

holds for the off-diagonal block matrices (mixed second order derivatives of scalar potentials are symmetric).

3.1.3. A More Compact Notation

The derivations given above contain rather specific expressions for gradient and Hessian. In terms of Euler-Lagrange equations, the system given in equation (3.1.29) can be written as,

$$-\begin{bmatrix} \frac{\partial\Pi}{\partial\boldsymbol{x}} \\ \frac{\partial\Pi}{\partial\boldsymbol{X}} \end{bmatrix}_n = -\begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{R} \end{bmatrix}_n = \begin{bmatrix} \frac{\partial\boldsymbol{r}}{\partial\boldsymbol{x}} & \frac{\partial\boldsymbol{r}}{\partial\boldsymbol{X}} \\ \frac{\partial\boldsymbol{R}}{\partial\boldsymbol{x}} & \frac{\partial\boldsymbol{R}}{\partial\boldsymbol{X}} \end{bmatrix}_n \cdot \begin{bmatrix} \Delta\boldsymbol{x} \\ \Delta\boldsymbol{X} \end{bmatrix}_n \qquad (3.1.32)$$

This compact notation will be used in the sequel, if more general aspects of r-adaptivity are concerned.

3.2. Properties of the Basic r-Adaptivity Problem

All previous chapters and sections dealt with the setup of a proper framework including equilibrium equations, linearisations and a standard algorithm to solve nonlinear problems. Now, having all the tools together and applying them to quite simple examples reveals some minor and major difficulties stemming from the nature of r-adaptivity. In the upcoming sections the major phenomena one has to deal with, when solving the discussed r-adaptivity problem, will be presented. As there already exists a popular model problem (see Figure 3.2.1) used in preceding works, e.g. (Mueller and Maugin 2002) and (Askes et al. 2004), this setting is used for first investigations. In this 2d example plane strain is assumed. The material is of neo-Hookean type with parameters $\lambda = E/[3(1-2\nu)] - 2/3\mu$ and $\mu = E/[2(1+\nu)]$ with $E = 2.0\text{e}11\text{Pa}$ and $\nu = 0.3$. The length $L = 1\text{m}$ and thickness $d = 1\text{m}$. The prescribed overall displacement $w = 0.5\text{m}$. However, this overall displacement will be applied incrementally (10 increments). As will be shown in the next subsection, these "load-", or better displacement-increments, yield interesting effects in material motion and r-adaptivity.

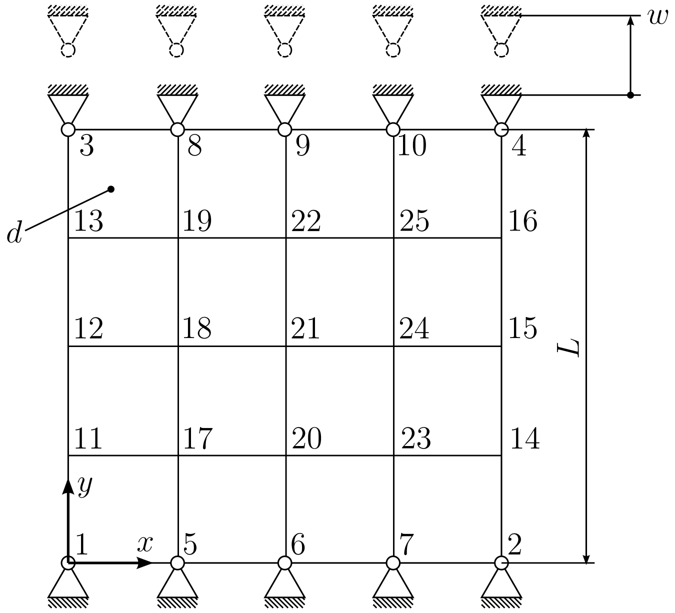

Figure 3.2.1. Model problem 1: square block tension

3.2.1. Displacement Increments

For this first study, only two inner nodes ("20", "22") were allowed to move in the material configuration. Observing material displacements for each spatial displacement increment shows that small spatial displacement increments in general do not yield small material displacements. Since the problem is symmetric, only node "22" will be observed. For the resulting spatial and material motion in the y-direction see Table 3.1 and Figure 3.2.2.

Increment #	spat. displ. [m]	mat. displ [m]
1	0.0399396	0.0285456
5	0.197885	0.0218019
10	0.391258	0.0143212

Table 3.1. Material displacement and spatial displacement of node "22" at different displacement increments

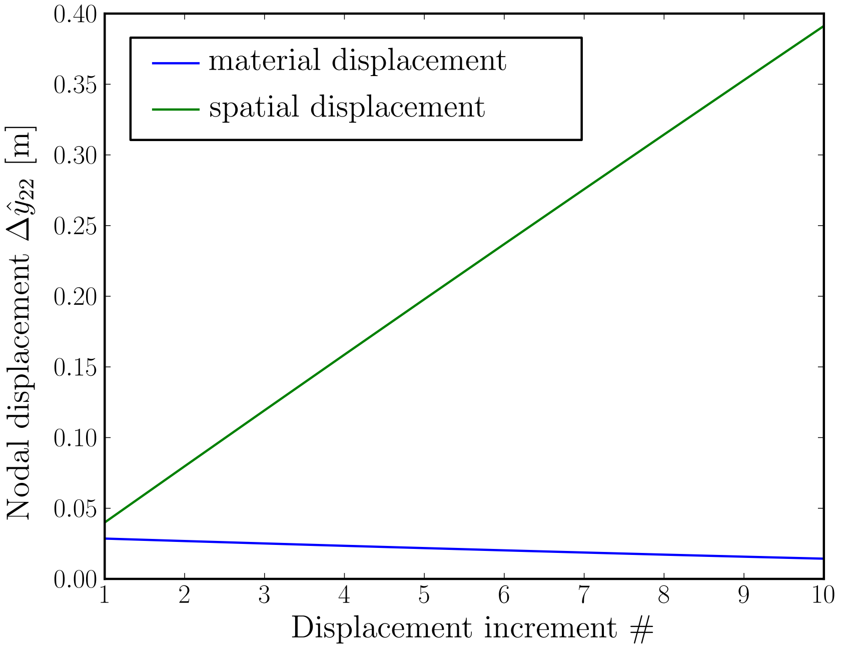

Figure 3.2.2. Material displacement of node "22" decreasing with increasing spatial displacement

This reproduces a behaviour mentioned in (Askes et al. 2004; Remark 7). Furthermore, higher material displacements at lower spatial displacements are related to the problem with singularities discussed in Section 3.2.5.

3.2.2. Convexity

Here all inner nodes are allowed to move in the material configuration and the system matrix is analysed at some Newton iterations. This yields results different from pure spatial motion computation where eigenvalues λ_i of the system matrix are usually positive. During the first two ("purely spatial") iterations only the spatial motion problem is treated, in all following iterations (beginning from number 3) the full r-adaptivity system is solved. As shown in Table 3.2, negative eigenvalues emerge beginning from iteration 3. The setup used in this case was the same as before, only the displacement (now $w = 0.1$m) is applied at once (no increments).

Iteration #	λ_{min}	λ_{max}	with r-Adaptivity
1	$3.26647640 \times 10^{10}$	$8.99231655 \times 10^{11}$	no
2	$3.35893754 \times 10^{10}$	$8.89363870 \times 10^{11}$	no
3	$-1.67829331 \times 10^{08}$	$1.39358464 \times 10^{12}$	yes
4	$-1.62069864 \times 10^{05}$	$1.85436530 \times 10^{12}$	yes
5	$-1.87446387 \times 10^{13}$	$1.39274754 \times 10^{13}$	yes

Table 3.2. Smallest and largest eigenvalues at different iterations

For this setting, the Newton-Raphson algorithm failed to converge. However, performing three purely spatial iterations instead of two results in a better conditioned system, the Newton-Raphson algorithm does converge and all eigenvalues are positive throughout all iterations.

An indefinite system matrix, in fact, is no exception but the common case. Hence, the pure r-adaptivity problem is not strictly convex. This is of great importance since strict convexity of a potential guarantees the existence of a unique local minimum (point of equilibrium, stationary point). Therefore, the optimisation problem may become more difficult.

Furthermore, the Newton-Raphson scheme may fail to converge as already illustrated in Figure 1.5.5.

3.2.3. Boundary Nodes

In the above computations only a few inner nodes were allowed to change their material position. Now, with the same parameters as used in Section 3.2.2, at first only the internal nodes are allowed to move in the material configuration. When this first stage has converged, all inner nodes are fixed and the boundary nodes are allowed to move tangentially to the boundary. After convergence of the second stage, the internal and boundary nodes are released and the final solution is obtained. The material motion boundary condition (for the final stage) are depicted in Figure 3.2.3.

Comparing the resulting total energy at the different stages (see Figure 3.2.4), it is obvious that the energy release achieved by material motion boundary nodes is significantly higher than that of inner nodes. In this case, the separation of the overall solution process into internal-only, boundary-only and overall r-adaptivity was necessary for the Newton-Raphson solver to converge. However, this strategy does not work in general, and therefore more sophisticated stabilisation procedures will be introduced below.

The sharp peaks in Figure 3.2.4 result from changes in material boundary conditions. The total energies at steps 0 to 2 are not shown and therefore, the peak at step 11 seems

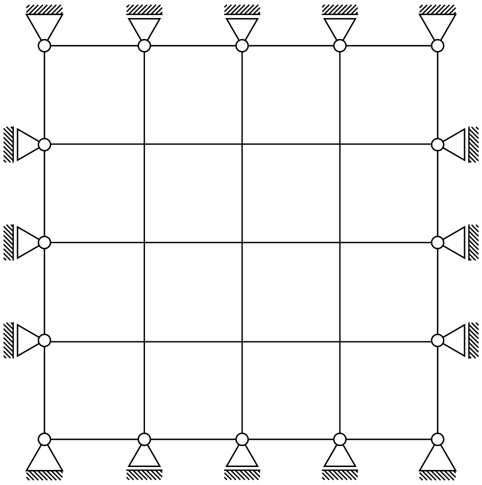

Figure 3.2.3. Model problem 1: material motion boundary conditions

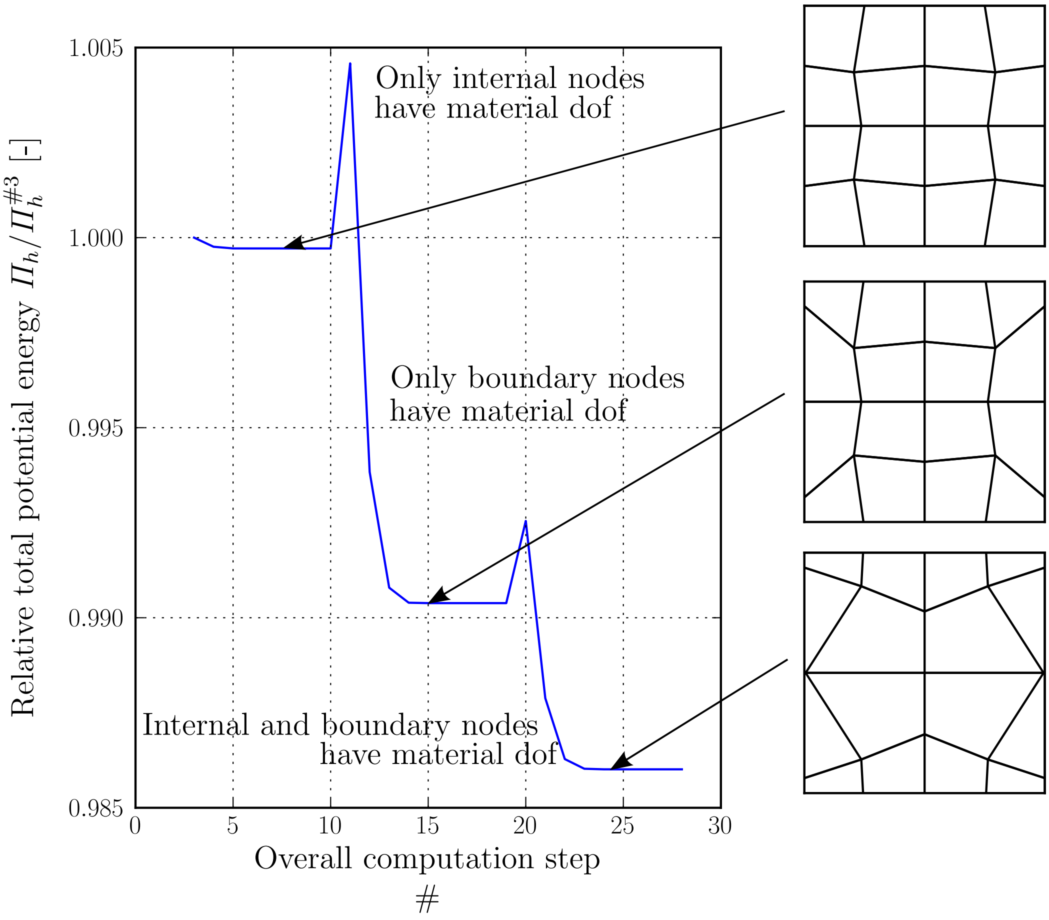

Figure 3.2.4. Model Problem 1: total potential energy with different material boundary conditions

to be a global maximum, however, this is not the case at all.
Applying material boundary conditions in the above manner can help to stabilise the solver procedure in very simple cases only. Therefore relying on this approach is not recommended in general.

3.2.4. Stationary Point Outside Admissible Domain

Allowing all nodes except vertices to move in the material configuration shows some nodes having large material displacements degenerating their elements or even leaving the material configuration's borders. Noteworthy, as in Figure 3.2.5, nodes that left the material domain may return in subsequent iterations. This phenomenon has been

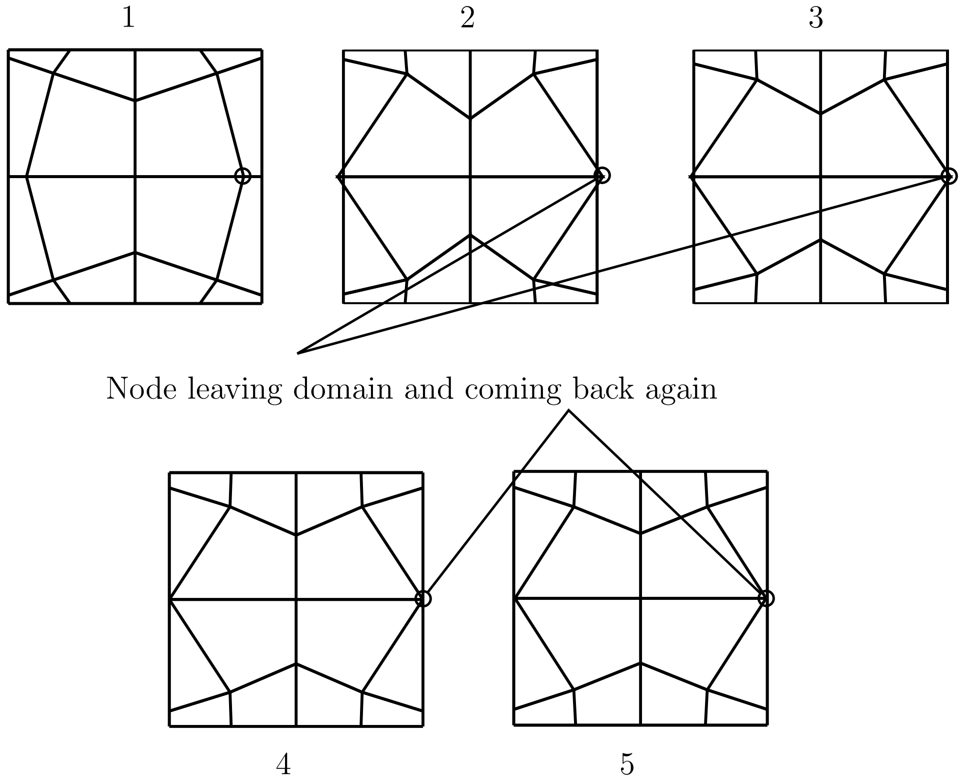

Figure 3.2.5. Model Problem 1: node leaving domain but coming back again

observed in overall steps 20 to 24 of the computation of which the setup and the results were already discussed in Section 3.2.3 and Figure 3.2.4.

More often there exists no admissible solution, since having nodes traversing (or leaving) the body's borders or degenerating elements is not the result one is looking for, even if that would minimise the total energy (Scherer et al. 2007; 2008). This corresponds with convexity or non-convexity respectively of the r-adaptivity problem. Figure 3.2.5 only describes tendencies of domain-leaving phenomenon. In Section 4.5 this problem will be discussed in more detail.

3.2.5. Singularities

To demonstrate the occurrence of singularities, the current setting has to be adapted. The mesh of the square block is refined and symmetry is used to solve the problem on a quarter of the whole block. Herein, $L/2 = 1\text{m}$ and $w/2 = 0.05\text{m}$, material and thickness d remain unchanged. Figure 3.2.6 gives an overview of the adapted example.

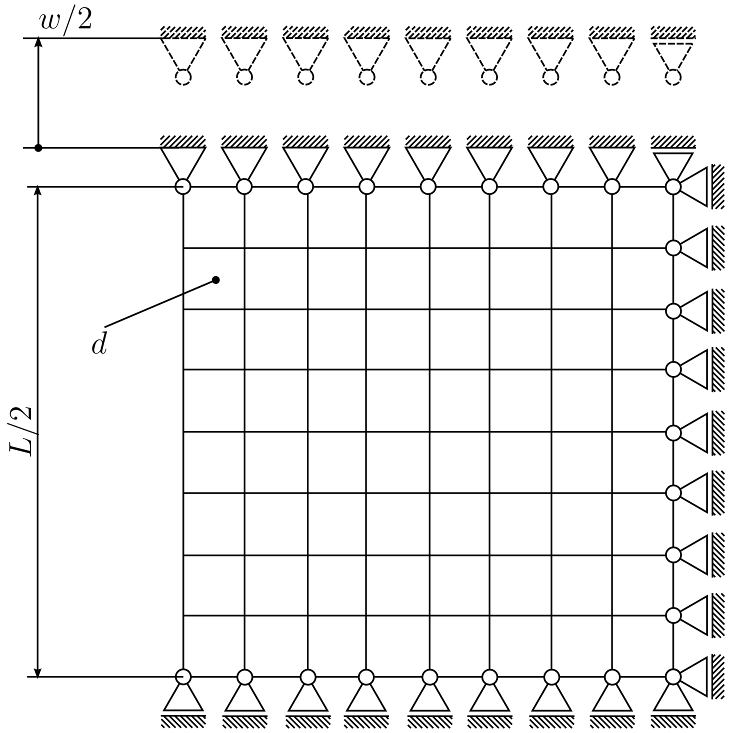

Figure 3.2.6. Model problem 2: square block tension on quarter system

The pure spatial motion problem yields rather small displacements in the lower right region in figure 3.2.7. Despite the fact that spatial equilibrium has been obtained before the first r-adaptivity step, the Newton-Raphson solver diverges immediately. This behaviour marks the extremum of the already discussed tendency of large material motion of nodes with rather small spatial displacement. In (Askes et al. 2004) it is mentioned that nodes with vanishing configurational forces cause singularities. As a remedy it is proposed to impose Dirichlet boundary condition on these nodes' material motion. However, in (Mosler and Ortiz 2006) it is shown that these nodes are not the only reason for singularities. In case of small spatial displacements, too "similar" spatial and material

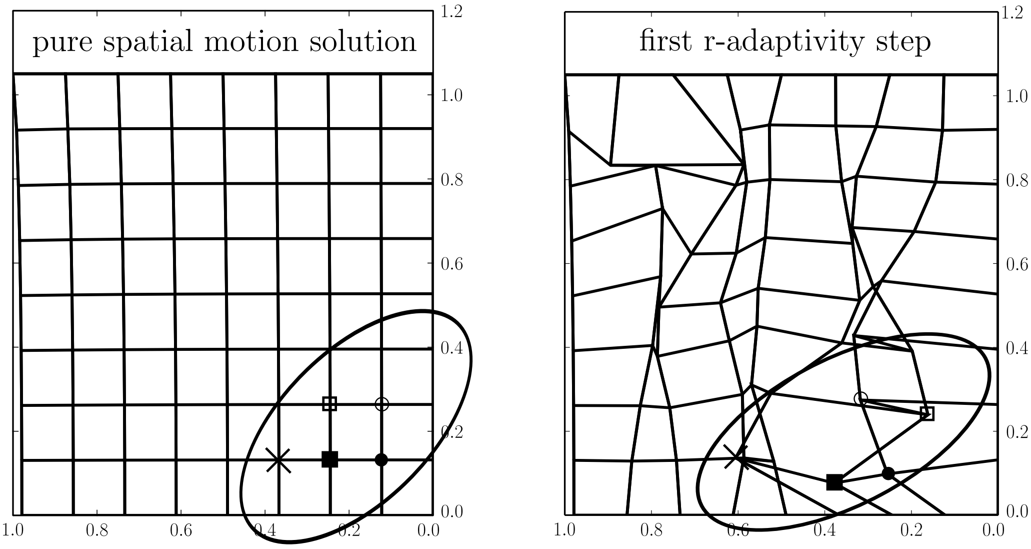

Figure 3.2.7. Model problem 2: nodes with too large material displacements

configurations may also cause problems (see section 4.5). This is the case in the above example, where configurational forces at boundary nodes in tangential direction to the surface are of the same magnitude as configurational forces of internal nodes. Therefore, there are no nodes with vanishing configurational forces.

Table 3.3 shows eigenvalues and condition numbers of the Hessian for the last pure spatial motion step and the immediately following r-adaptivity step. Comparing condition numbers, the Hessian of the first r-adaptivity step is far worse conditioned than the Hessian of the last pure spatial step. On the other hand, the r-adaptivity Hessian is not positive definite. Nevertheless, the system is ill-conditioned enough to cause inadmissible material displacements, even if it is not singular in a narrower sense. In pure spatial analysis inadmissible displacements are avoided by applying the load incrementally. However, these increments only concern the spatial motion problem and the material displacements may even be higher at low increments, as discussed in Section 3.2.1. Hence, other ways to incrementally apply the “material” load have to be found. Several more or less successful approaches (solution strategies) will be introduced in the next chapter.

Step	$\lvert\lambda\rvert_{min}$	λ_{min}	λ_{max}	κ_2
last pure spatial	8.538e9	8.538e9	1.032e12	1.209e2
first r-adaptivity	3.024e3	−1.411e4	2.020e12	6.641e8

Table 3.3. Eigenvalues and condition numbers at last pure spatial and first r-adaptivity step

4. Solution Strategies

As pointed out in Section 3.2 there is more to r-adaptivity than just variating an energy functional and then linearising it. Different strategies to overcome these difficulties have been proposed by various authors. Probably the most important publications on this topic are those of Mosler and Ortiz (2006) and Scherer et al. (2007; 2008). In (Kuhl et al. 2004; Askes et al. 2004) the VALE framework for r-adaptivity was introduced and prepared for the Newton-Raphson scheme. First proposals have been made, for example the dynamic selective imposition of material Dirichlet boundary conditions. In (Mosler and Ortiz 2006) the nature of singularities of the system matrix have been further investigated and a first regularisation approach has been presented. Subsequently, in (Scherer et al. 2007; 2008) Scherer et al., introduced more sophisticated regularisation methods based on fictitious potentials and mesh quality measures. Another aspect is the type of linearisation and iterative solver procedure. In (Askes et al. 2004) a plain staggered (completely uncoupled linearisations) as well as a monolithic (coupled, fully linearised system) Newton-Raphson scheme had been tested. An improved staggered scheme considering the coupling between spatial and material motion was applied in (Scherer et al. 2007). In (Mosler and Ortiz 2006) a monolithic scheme making use of line search and angular criteria was used. The following sections will discuss the approaches introduced above in more detail.

4.1. Staggered Newton-Raphson Scheme

The staggered scheme from (Askes et al. 2004) solves the pure spatial motion problem and the pure material motion problem alternatingly.

Performance considerations given in that work show the implications of neglecting the strong coupling of spatial and material motion. It is important to note that in this first staggered scheme the spatial motion and material motion were iterated until convergence was reached before switching the type of motion.

Besides simultaneously solving the fully linearised system another possibility to consider the coupling was given by Scherer et al. (2007). Therein, an improved staggered scheme has been introduced that uses the implicit dependence of the discrete energy functional $\varPi^h(\boldsymbol{X}) = \varPi^h(\boldsymbol{\varphi}(\boldsymbol{X});\boldsymbol{X})$. Furthermore, the spatial equilibrium is recovered after each material motion increment. Now, as $\varPi^h$ is a function of only one variable and $\varPi^*(\boldsymbol{\varphi}^*(\boldsymbol{X});\boldsymbol{X})$ denotes $\varPi^h$ at spatial equilibrium, the coupled scheme given in equation (3.1.32) is rewritten:

$$\begin{aligned}\boldsymbol{R}^* = \frac{\partial \varPi^*(\boldsymbol{\varphi}(\boldsymbol{X});\boldsymbol{X})}{\partial \boldsymbol{X}} &= \left.\frac{\partial \varPi^h(\boldsymbol{\varphi}(\boldsymbol{X});\boldsymbol{X})}{\partial \boldsymbol{\varphi}(\boldsymbol{X})}\right|_{\boldsymbol{\varphi}^*,\boldsymbol{X}} \cdot \left.\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{X}}\right|_{\boldsymbol{\varphi}^*,\boldsymbol{X}} + \left.\frac{\partial \varPi^h}{\partial \boldsymbol{X}}\right|_{\boldsymbol{\varphi}^*,\boldsymbol{X}} \\ &= \left.\frac{\partial \varPi^h}{\partial \boldsymbol{X}}\right|_{\boldsymbol{\varphi}^*,\boldsymbol{X}} \end{aligned} \tag{4.1.1}$$

$$\frac{\partial^2 \varPi^*}{\partial \boldsymbol{X}^2} = \frac{\partial \boldsymbol{R}^*}{\partial \boldsymbol{X}} = \left.\frac{\partial \boldsymbol{R}^*}{\partial \boldsymbol{X}}\right|_{\boldsymbol{\varphi}^*,\boldsymbol{X}} + \left.\frac{\partial \boldsymbol{R}^*}{\partial \boldsymbol{\varphi}}\right|_{\boldsymbol{\varphi}^*,\boldsymbol{X}} \cdot \left.\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{X}}\right|_{\boldsymbol{\varphi}^*,\boldsymbol{X}} \tag{4.1.2}$$

The term that does not occur in equation (3.1.32) is $\dfrac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{X}}$. This term can be computed as

$$\left.\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{X}}\right|_{\boldsymbol{\varphi}^*,\boldsymbol{X}} = -\left.\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\varphi}}\right|_{\boldsymbol{\varphi}^*,\boldsymbol{X}}^{-1} \cdot \left.\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{X}}\right|_{\boldsymbol{\varphi}^*,\boldsymbol{X}}, \tag{4.1.3}$$

which is an expensive operation since it involves the inversion of a matrix and the multiplication of two square matrices.

Note that because of symmetry (see equation (3.1.31)) $\left.\dfrac{\partial \boldsymbol{r}}{\partial \boldsymbol{X}}\right|_{\boldsymbol{\varphi}^*,\boldsymbol{X}} = \left.\dfrac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{\varphi}^*,\boldsymbol{X}}^{T}$. The staggered scheme from (Scherer et al. 2007) that also has been used in (Scherer et al. 2008) is given below:

Table 4.1. Staggered r-adaptivity scheme

```
start from state of spatial equilibrium
while $||\boldsymbol{R}|| > \epsilon_1$ do
    compute deformation gradient matrix using equation (4.1.3)
    compute material motion gradient $\boldsymbol{R}^*$ and Hessian $\frac{\partial \boldsymbol{R}^*}{\partial \boldsymbol{X}}$ using equation
    (4.1.1) and equation (4.1.2) respectively.
    solve $\frac{\partial \boldsymbol{R}^*}{\partial \boldsymbol{X}} \cdot \Delta\hat{\boldsymbol{X}} = -\boldsymbol{R}^*$
    apply material motion increment $\Delta\hat{\boldsymbol{X}}$: Update nodal material coordinates and integration point data
    recover spatial equilibrium: Standard spatial motion FEM
end while
```

Because it respects the implicit dependence, this scheme uses exactly the same block matrices (except from $\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{X}}$) as the coupled ("monolithic") scheme. In a comparison of both schemes applied to model problem 1 (see Section 3.2) the convergence rates of both schemes were approximately equal. However, within a staggered scheme the systems of equations to solve have N degrees of freedom whereas within the coupled scheme systems of $2N$ equations have to be solved. Assuming a solver of complexity $\mathcal{O}(N^3)$ as the procedure's bottle-neck, the staggered scheme outperforms the coupled scheme as soon as the recovery of spatial equilibrium takes less than approximatetly five iterations which is the case for most staggered iterations.

4.2. Monolithic Newton-Raphson Scheme

In contrast to staggered schemes, in monolithic schemes it is not necessary to regain spatial equilibrium after solving the material motion problem as both problems, spatial and material motion, are solved simultaneously. Therefore, except for the matrices that have to be assembled (see Section 3.1.2) the procedure is the same as for standard spatial motion FEM. However, while for staggered schemes the system of equations is of size N, for monolithic schemes it is of size $2\,N$. So, the former method results in a higher number of less expensive iterations (more systems to solve). Assuming an overall complexity for one iteration (assembling and solving one system of equations) of $\mathcal{O}(N^3)$, solving about eight systems of equations in the staggered scheme causes the same effort as solving one system in the monolithic scheme. The coupled scheme might be beneficial in case of line search methods, as for each line search step material and spatial configuration have to be updated, before the internal potential energy can be computed. Latter updates involve solving systems of equations to recover spatial equilibrium after each material motion update.

4.3. Line Search

Because of the limitations of the Newton-Raphson scheme concerning starting points not close to the final solution, a frequently used class of tools to "globalise" the standard Newton-Raphson scheme are line search methods. All line-search-enhanced Newton methods have in common that they use the increment obtained by the Newton-Raphson scheme, the Newton direction, and take this direction to find the best increment determined by an appropriate criterion. The range of scaling factors and the distribution of line search points are the main differences between line search methods. Furthermore, many methods use an angular criterion to ensure that the Newton direction is a descending direction. An introduction into line search methods is given in (Ulbrich and Ulbrich 2011). The line search algorithm used in this work is presented in Table 4.2.

Table 4.2. Line search algorithm

```
obtain Newton direction $\Delta\boldsymbol{x}$ through a standard Newton iteration
set SUCCESS = False
set scaling increment $\Delta s$
set initial scaling $s = \Delta s$
set best scaling $s_{best} = 0$
while SUCCESS == False and $s \leq 1$ do
    apply increment $s\,\Delta\boldsymbol{x}$, update nodal data and integration points
    compute total potential energy change $\Delta\Pi_h = \Pi_h(s_{best}) - \Pi_h(s)$
    if $\Delta\Pi_h < 0$ then
        if $s > 0$ then
            SUCCESS = True
        else
            break
        end if
    else
        $s_{best} = s$
        $s = s + \Delta s$
    end if
end while
return SUCCESS, $s_{best}$
```

Since this algorithm has not been applied in every example, its use is mentioned in the example's or figure's description explicitly. The main purpose was to produce plots of energy over the scaled increment to better understand the problem's behaviour. It also has been more or less successfully used to stabilise the solution procedure. Although in some cases the convergence behaviour had been improved, the presented line search method turned out to be insufficient to overcome all solver failures.

4.4. Selective Material Motion Restriction

As outlined in (Mosler and Ortiz 2006) there are very few cases where the full linearised system (3.1.29) can be solved without means of stabilisation. To overcome singularities arising from nodes with (nearly) vanishing material motion residual, selective fixation of such nodes has been proposed. A special subclass of that kind of nodes are those located on symmetry axes or planes. They may cause problems when numeric errors destroy symmetry. The fact that symmetry is often used to reduce computational effort, for example by doing analysis on only a quarter of the original model and introducing boundary conditions at symmetry axes or planes, indicates the special role of “symmetry nodes“ for r-adaptivity as some kind of “boundary nodes inside the domain“.

Model problem 1 as introduced in Section 3.2 will be used to demonstrate the effects of this strategy. For this purpose, a minimal ratio β has to be defined, to filter (material) degrees of freedom that have to be removed. The filtering condition reads as,

$$\beta \cdot \max(|RHS_{material}|) \leq |RHS_{material}| \leq \max(|RHS_{material}|) \tag{4.4.1}$$

where $RHS_{material}$ denotes the set of Right-Hand-Side (RHS) values corresponding to the material motion problem. $|RHS_{material}|$ denotes the set of absolute RHS values corresponding to the material motion problem.

The evolution of the final solution when the presented strategy is applied is depicted in Figure 4.4.1. Herein, the filter ratio was set to $\beta = 0.7$. All “material-dofs” not fulfilling the filtering condition given in equation (4.4.1) were removed by restricting their motion via Dirichlet boundary conditions.

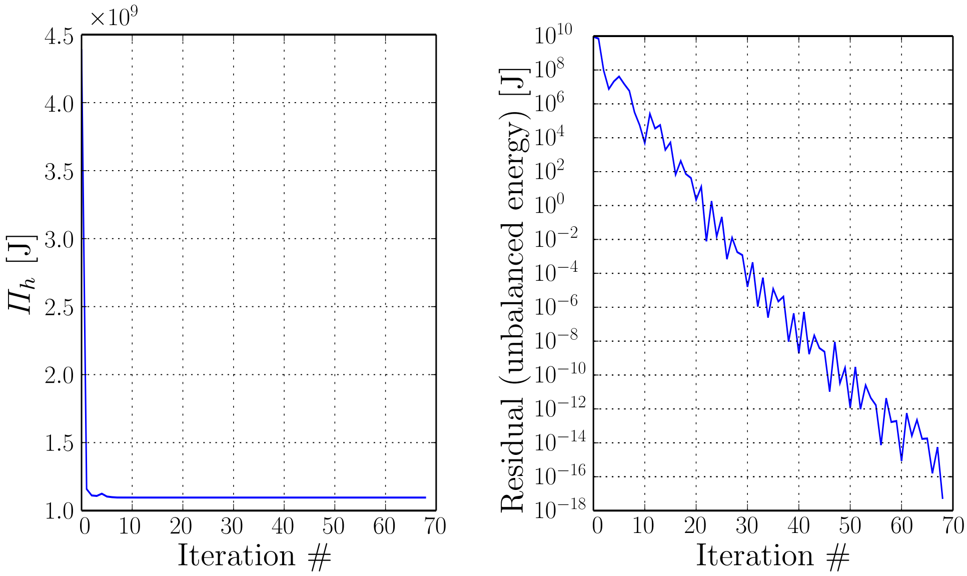

Figure 4.4.1. Model problem 1 - strategies: selecting nodes with high configurational forces, total energy and residuals

Choosing the selection parameter β sufficiently large ($\beta = 0.5$ failed), the computation converges without additional stabilisation such as line search methods. However, convergence behaviour is rather weak. The final solution is identical to the results obtained in Section 3.2.3.

Enhancing this method by extending the Newton-Raphson algorithm with a simple line search routine allows lower values for β. Figure 4.4.2 shows a comparison of the potential energy and residuals over iterations for this strategy solved with the line search enhanced

Newton-Raphson solver and different parameters β. As frequently altering material motion boundary conditions affect convergence in a negative manner, lower values for β are beneficial. Despite these promising results, this method fails when applied to model problem 2 and shows a behaviour approximately similar to the one depicted in Figure 3.2.7. Therefore, in the following sections more sophisticated methods will be introduced.

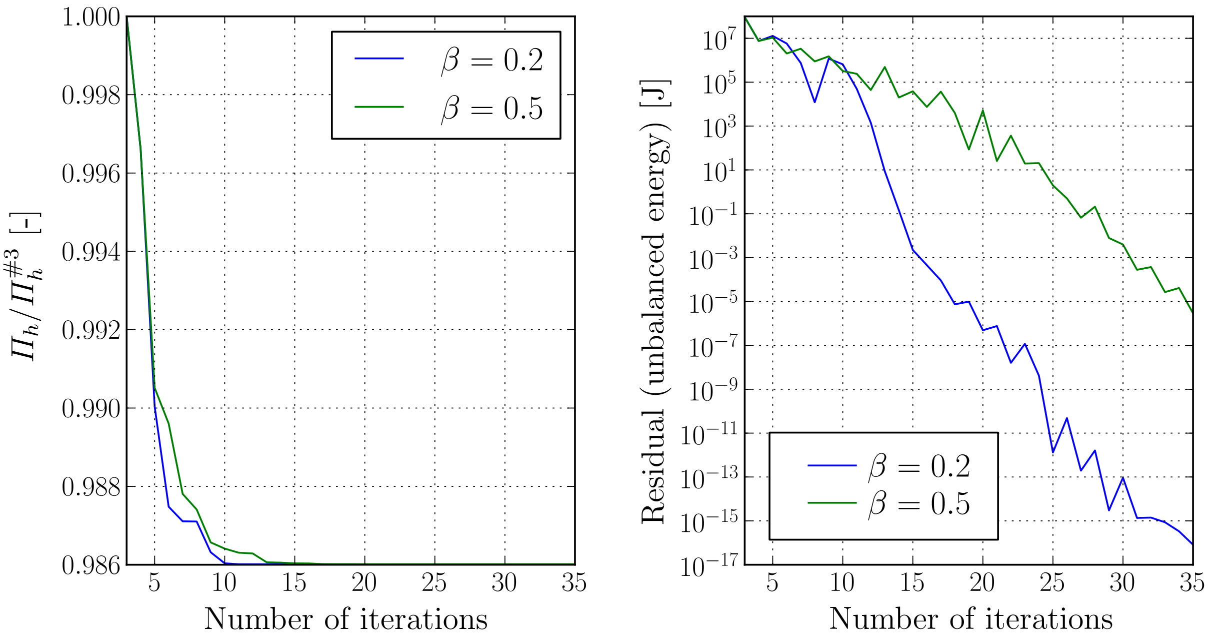

Figure 4.4.2. Model problem 1 - strategies: selective node fixation with line search, comparison of fixation parameters

Remark 4.1. Total energy, residuals and convergence criteria: In traditional spatial motion analysis an unbalanced energy type criterion such as $|\Delta \boldsymbol{u} \cdot \boldsymbol{r}| < tol$ is well suited convergence criterion. However, if a problem is not strictly convex, a solution may be a local minimum, a local maximum, a saddle point or a neutral state. Therefore it may be also necessary to validate results by evaluating the total energy or total energy change along the Newton increment. In the above example it turned out that at low residuals the change of potential energy is too small to be evaluated with double precision floating point numbers. Thus, validation by means of total energy may not be possible in general. A line search plot of discrete energy Π_h at iteration 27 with $\beta = 0.2$ is depicted in Figure 4.4.3. All eigenvalues of the Hessian at this iteration are positive.

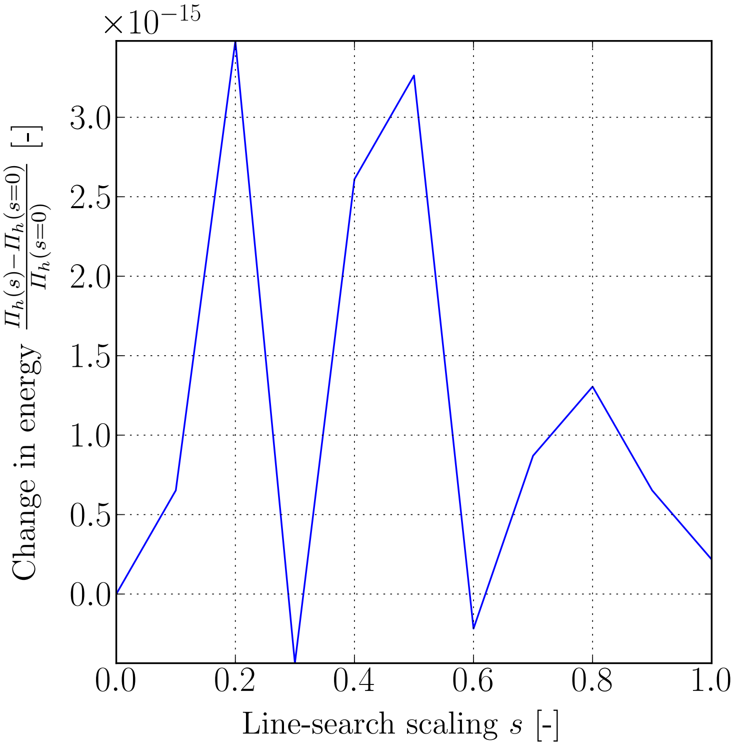

Figure 4.4.3. Model problem 1 - strategies: selective node fixation with line search, change of potential energy at iteration 27

4.5. Regularisation and Stabilisation

Besides vanishing material motion residuals however, Mosler and Ortiz (2006) pointed out that there is one more source of singularities. In the case of regions with rather small spatial displacements there is only a very small difference between spatial and material configuration and so the same holds for derivatives with respect to spatial and material coordinates. Since both derivatives are part of the linearised system, they yield almost equal blocks in each of the four sub-matrices $\mathbb{A}, \mathbb{a}, \mathbb{B}$ and $\mathbb{b}$ from (3.1.30) resulting in almost linear dependent equations and an ill-conditioned system.

This section deals with several approaches to regularise the optimisation problem. These methods can be applied to both staggered and monolithic schemes.

4.5.1. Viscous Regularisation

Mosler and Ortiz (Mosler and Ortiz 2006) used a viscous regularisation term only depending on material coordinates:

$$\Pi_{n+1}^{visc} = \Pi_{n+1} + \alpha \left\| \boldsymbol{X}_{n+1} - \boldsymbol{X}_n \right\|^2 \tag{4.5.1}$$

resulting in,

$$\left.\frac{\partial \Pi^{visc}}{\partial \boldsymbol{X}}\right|_{n+1} = \left.\frac{\partial \Pi}{\partial \boldsymbol{X}}\right|_{n+1} + 2\alpha\,(\boldsymbol{X}_{n+1} - \boldsymbol{X}_n) \tag{4.5.2}$$

$$\left.\frac{\partial^2 \Pi^{visc}}{\partial \boldsymbol{X}^2}\right|_{n+1} = \left.\frac{\partial^2 \Pi}{\partial \boldsymbol{X}^2}\right|_{n+1} + 2\alpha\,\boldsymbol{I} \tag{4.5.3}$$

where $\boldsymbol{I}$ is the unit tensor.
This regularisation not only overcomes problems of rank-deficiency but also makes it possible to obtain a strictly convex problem, given that parameter α is large enough. On the other hand, α influences convergence behaviour. While being close to the exact solution, when displacement increments $\boldsymbol{X}_{n+1} - \boldsymbol{X}_n$ are small, the viscous term has only minor influence, in case of large increments and high values for parameter α the viscous term significantly alters the overall behaviour. In fact, when the additional term on the RHS becomes dominant, the method fails. This may be induced by bad predictors for the unknown $\boldsymbol{X}_{n+1} - \boldsymbol{X}_n$ or too large values for α. Therefore, as Mosler and Ortiz pointed out, α has to be chosen with care. Aiming for a positive definite system, one has to ensure that all eigenvalues are positive and therefore to choose α large enough. Hence, for finding the smallest parameter α that yields a positive definite system, one has to compute the smallest eigenvalue and increase α if this eigenvalue is not positive or larger than a given tolerance. However, this procedure is rather expensive and has to be repeated for each iteration. Thus, it may be beneficial to estimate α high enough and adapt heuristically in cases of weak convergence behaviour. The latter approach should be used with care since the sequence of values determines whether the procedure converges or not. Furthermore, the line search enhanced Newton-Raphson procedure used in the previous section will be used again in the following example. As shown in Figure 4.5.1 the problematic nodes (see Section 3.2.5, Figure 3.2.7) in the lower right region reduced their material motion. However, large material motion of nodes in the upper left region induce degenerated elements. At this state, configurational forces have been reduced, but their magnitude is still too high. Since the proposed method is intended to not influence the final solution, setting the parameter α to higher values, will not prevent nodes from leaving the domain.

In Section 3.2.4, it has been outlined that large material displacements may induce degenerated elements or even let inner nodes leave the domain. Viscous stabilisation is in general not able to prevent these effects and thus, material motion has to be restricted by additional means, e.g. restricting incremental material motion.

4.5.2. Fictitious Potentials

Acknowledging that in many cases the solution (stationary point) cannot be obtained due to inadmissible material displacements, Scherer et al. (2008) proposed a stabilisation

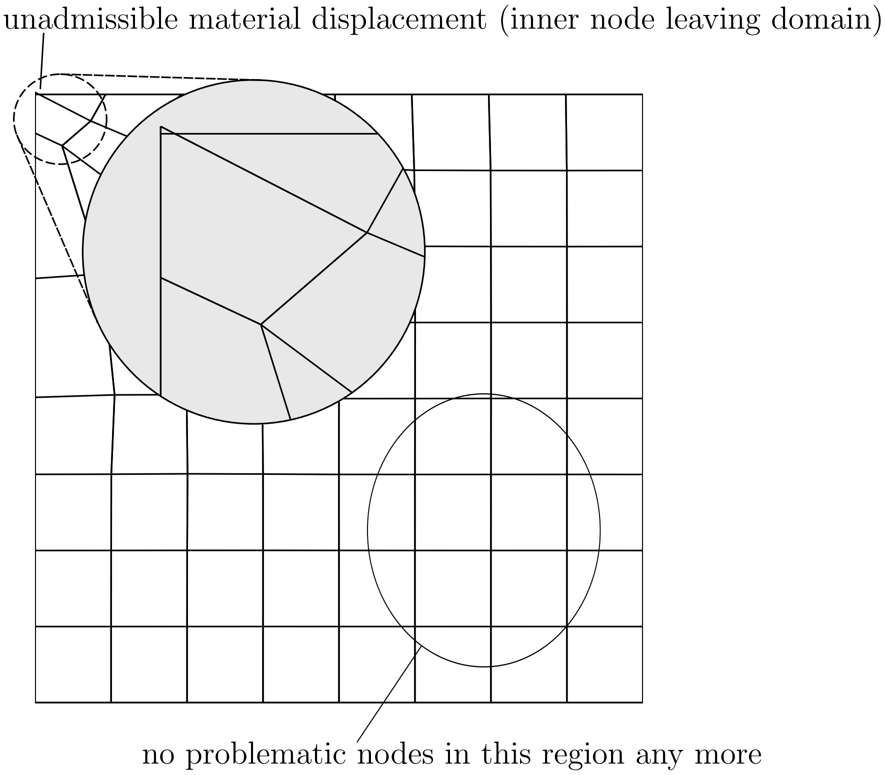
unadmissible material displacement (inner node leaving domain)
no problematic nodes in this region any more

Figure 4.5.1. Model problem 2 (deformed mesh) - strategies: viscous regularisation, $\alpha = 2 \times 10^8$ yields inadmissible material displacements

term that acts as a barrier function limiting material displacements. This stabilisation term is a type of "fictitious potential". The idea behind this approach is the addition of a potential introducing an artificial material motion stiffness between the material configuration and the reference configuration. Thereby, this stiffness tends locally to infinitely high values, if an element's deformation reaches a predefined level and therefore, this artificial stiffness is intended to act as a barrier for material motion. In (Scherer et al. 2008) a neo-Hooke-like potential $\varPsi^+$ has been constructed:

$$\varPsi^+ = \frac{1}{2(1-\delta)} \, f_{*ij} \, f_{*ij} - \ln\left(j_r - \delta\right) \tag{4.5.4}$$

$$\varPi^{+(m)} = \int\limits_{V_r^{(m)}} \varPsi^+ \mathrm{d}V_r \tag{4.5.5}$$

where

δ is a limit for change of volume,

f_{*ij} denotes the deformation gradient mapping from the reference configuration to the material configuration and

j_r is the determinant of f_{*ij}.

Following Section 1.2.1,

$$f_{*ij} = \frac{\partial \varPhi_i}{\partial \xi_j} \tag{4.5.6}$$

Since this fictitious potential considers constraints of the r-adaptivity solution, the new (constrained) overall total energy $\varPi^c$ that will be subjected to minimisation is written as,

$$\varPi^c = \varPi + \gamma\, \varPi^+ \tag{4.5.7}$$

Herein, γ is a weighting parameter similar to α in the viscous stabilisation approach. However, in comparison with the viscous regularisation, the fictitious potential approach inherently misses the property to lose influence when the Newton-Raphson procedure converges. On the one hand, this behaviour ensures that constraints such as admissible elements are pertained but on the other hand it counteracts minimisation of the total inner potential energy. Hence, it is reasonable to lower γ after each successful Newton run and raise γ in case of failures (Scherer et al. 2008).

Remark 4.2. The term barrier function refers to a whole class of solution methods for optimisation problems, constrained by (nonlinear) inequalities. In contrast to penalty methods barrier methods do not allow violation of inequality constraints. This is achieved through replacing inequality constraints with functions containing singularities. Thus, they may behave more problematically than penalty methods but prevent inadmissible solutions. Detailed information on barrier methods is available in Boyd and Vandenberghe (2004).

4.5.2.1. Variations of Fictitious Potential

The first order variation with respect to material coordinates reads as,

$$\begin{aligned} \delta \varPi^+ = \delta_{\boldsymbol{X}} \varPi^+ &= \int\limits_{V_r} \frac{\partial \varPsi^+}{\partial f_{*ij}} \delta_{\boldsymbol{X}} f_{*ij} \,\mathrm{d}V_r \\ &= \int\limits_{V_r} \frac{\partial \varPsi^+}{\partial f_{*ij}} \frac{\partial \delta \varPhi_{*i}}{\partial \xi_j} \,\mathrm{d}V_r . \end{aligned} \tag{4.5.8}$$

The discretised form,

$$\begin{aligned} \delta \varPi_h^{+(m)} &= \int\limits_{V_r^{(m)}} \delta \hat{\varPhi}_{*ik} \frac{\partial \varPsi^+}{\partial f_{*ij}} \frac{\partial h_k}{\partial \xi_j} \,\mathrm{d}V_r^{(m)} \\ &= \delta \hat{\varPhi}_{*ik} \int\limits_{V_r^{(m)}} \frac{\partial \varPsi^+}{\partial f_{*ij}} \mathring{\mathbb{J}}_{*kj} \,\mathrm{d}V_r^{(m)} . \end{aligned} \tag{4.5.9}$$

The linearisation with respect to material coordinates yields,

$$\varDelta\delta\varPi_h^{+(m)} = \delta\hat{\varPhi}_{*im} \int\limits_{V_r^{(m)}} \mathring{\mathbb{j}}_{*mj} \frac{\partial^2 \varPsi^+}{\partial f_{*kl} \partial f_{*ij}} \mathring{\mathbb{j}}_{*nl} \, \mathrm{d}V_r^{(m)} \, \varDelta\hat{\varPhi}_{*kn}. \tag{4.5.10}$$

The given variation and linearisation are completely compatible with equation 3.1.27 and therefore, can easily be integrated into the basic schemes.

4.5.2.2. Fictitious Potential Method Applied to Model Problem 2

For all examples within this section, the volume change parameter δ was set to 0.1. A value of $\delta = 0.05$ allowed nodes to leave the domain as already observed in Section 4.5.1. This may occur because the evaluation of all integrals is done by Gaussian quadrature, where function values are evaluated at certain interior points (Gauss points) only. The highly nonlinear and nonpolynomial character of the fictitious energy function is not integrated exactly in general and therefore, not only the gradient and the Hessian may contain significant errors, but all admissibility checks based on Gaussian quadrature of the fictitious energy function are not exact. Using better suited integration methods or at least a higher number of Gauss points may improve the overall performance of the fictitious potential method. The mesh quality approach presented in the upcoming section may also benefit from proper integration methods. However, for linear triangular elements, Jacobians and other derivatives are constant over the element. Thus, the admissibility checks based on numerical integration are exact for these elements.

The figures 4.5.2 and 4.5.3 show a comparison of the fictitious potential approach with and without line search. The energy graphs in Figure 4.5.2 start from the converged solutions for the initial weighting parameters γ. Note that line search allows a significantly lower initial weighting parameter. Hence, the graph corresponding to the line search version starts at a lower discrete total energy $\varPi_h$.

While the final solution of both approaches is nearly identical, the line search enhanced procedure allows a lower initial weighting parameter. However, the number of iterations until convergence (marked by coloured circles) is higher when line search is used. Figure 4.5.3 shows stagnating weighting parameters somewhere around 2×10^6 for both approaches and line search does not show any advantage in this region. In fact, there is a rather significant kink in both figures, from which any further reduction of the weighting parameter is computationally expensive compared to the rather small improvement in energy $\varPi_h$. Figure 4.5.4 compares the first adapted mesh and the final mesh obtained by both procedures. Here again, the first solution shows significant differences, the final result is practically identical. Noteworthy, the nearly degenerated element at the left top of the final meshes is the one causing problems with lower values for δ mentioned before.

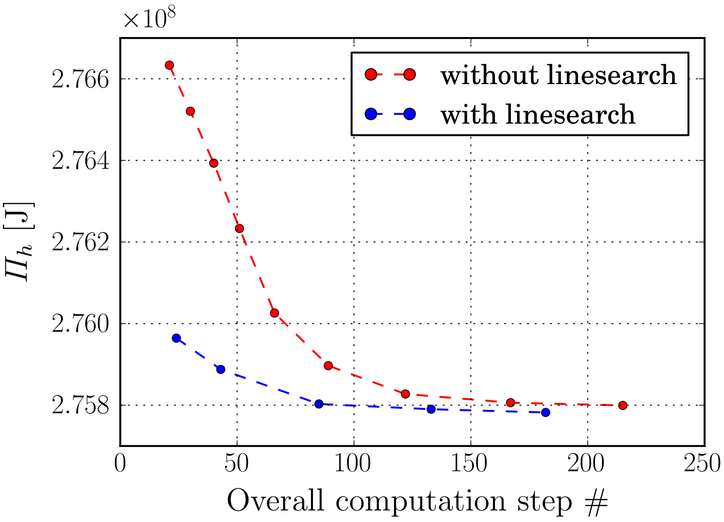

Figure 4.5.2. Model problem 2 - strategies: fictitious potential, total energy

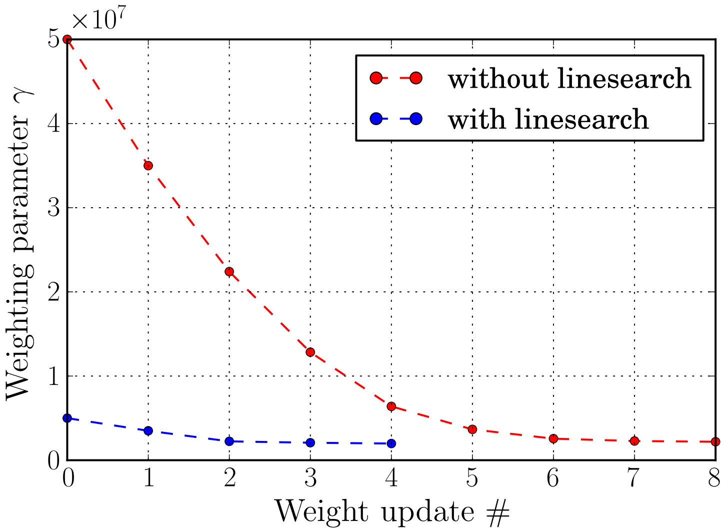

Figure 4.5.3. Model problem 2 - strategies: fictitious potential, weighting parameters

4.5.3. Mesh Quality-based Regularisation

While the fictitious potential approach shows satisfying regularisation and convergence properties, it does not take mesh quality into account explicitly. Moreover, decreasing the weighting parameter may result in admissible (in sense of positive Jacobian determinants) but nearly degenerated meshes. Another regularisation approach introduced in (Scherer et al. 2007) works with a barrier method based on a mesh quality measure for triangular and tetrahedral shapes with linear shape functions,

$$\upsilon^{(m)} = \frac{f_{*ij}\, f_{*ij}}{(j_r)^{2/n^{dim}}} \tag{4.5.11}$$

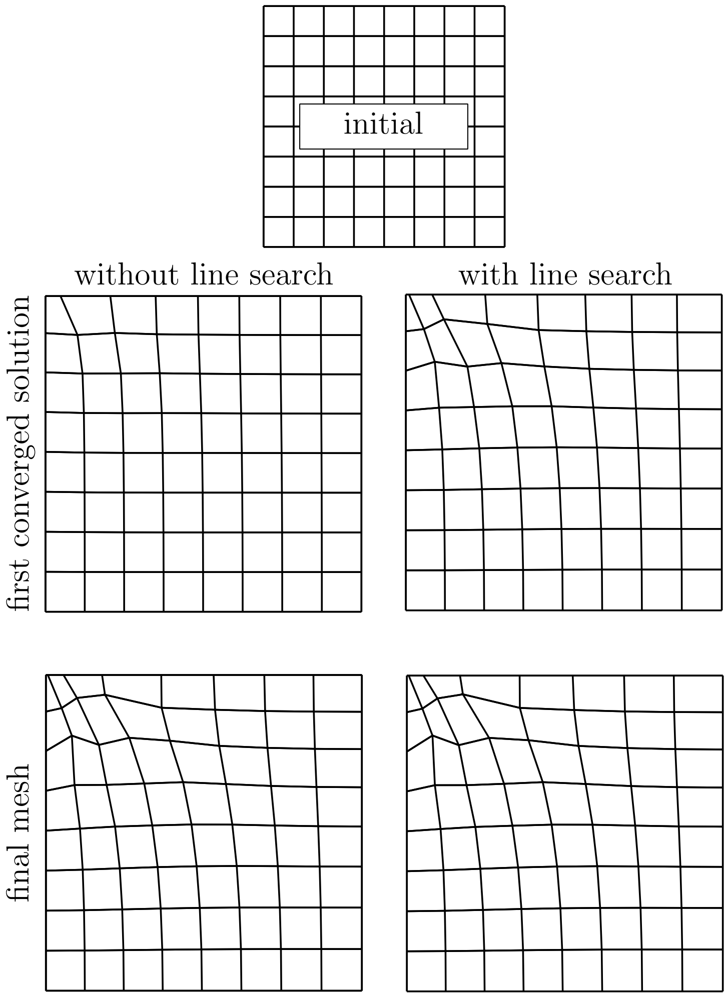

Figure 4.5.4. Model problem 2 - strategies: mesh deformation with and without line search (material configuration)

whereby the deformation gradient mapping the reference configuration (initial mesh) and the material configuration (deformed mesh) is used. This causes the quality measure to be only relative and therefore the quality of single elements depends on their quality at the initial state. Note that only for linear shape functions υ is constant on each element and can be used as a measure for shape quality directly. Thus, within this thesis, the mesh quality approach presented will be applied to linear triangular or tetrahedral shapes only.

The global barrier function constructed from 4.5.11 reads as,

$$\Pi^{+(m)} = \frac{-1}{\delta - \upsilon^{(m)^{-1}}} \tag{4.5.12}$$

The further implementation is the same as for fictitious potentials.

4.6. Comparison of Mesh Quality and Fictitious Potential Approaches

To demonstrate the different behaviour of the mesh quality and of the fictitious potential approach, a new model problem is introduced (see Section 4.6.1). The calculations were performed with staggered and coupled schemes, all without line search methods.

4.6.1. Model Problem 3: Cracked Specimen

This example was shown in (Scherer et al. 2007) and (Scherer et al. 2008). The values used in this thesis are based on (Scherer et al. 2008). The full problem as used in these publications is depicted in Figure 4.6.1.

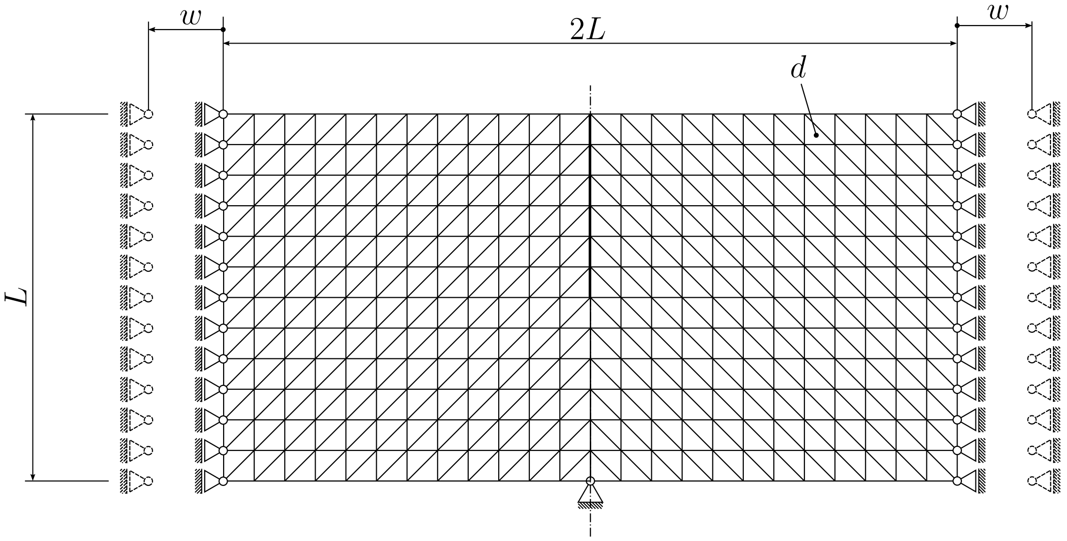

Figure 4.6.1. Model problem 3: cracked specimen, full problem

Since the problem is symmetric, all of the following computations were performed with the equivalent half system. Figure 4.6.2 shows the mesh and spatial boundary conditions of model problem 3.

Mesh and spatial boundary conditions: The length $L = 10\text{m}$, thickness $d = 1\text{m}$ and the prescribed deformation $w = L/40 = 0.25\text{m}$. The type of spatial boundary conditions is depicted in Figure 4.6.2.

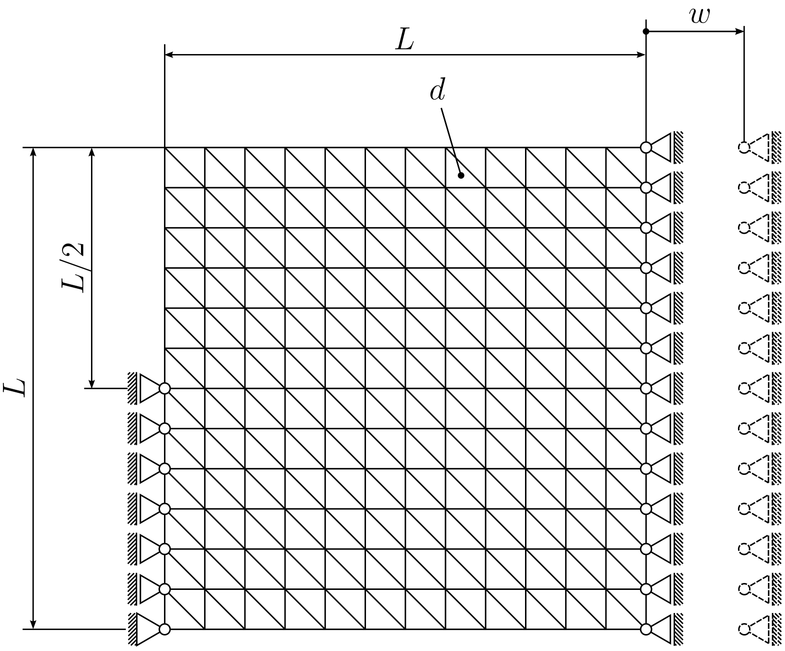

Figure 4.6.2. Model problem 3: cracked specimen, spatial motion boundary conditions

Material motion restrictions: Boundary nodes were allowed to move tangentially to the boundary, material motion of all vertices is fully restricted. All other nodes do not have any restrictions at all.

Material properties: For this example, a neo-Hookean type material with parameters $\lambda = 1.15 \times 10^3$Pa and $\mu = 7.69 \times 10^2$Pa is used. These parameter values correspond to a Young's modulus $E = 2.0 \times 10^3$Pa and a Poisson's ratio $\nu = 0.3$.

4.6.2. Improving Performance of the Mesh Quality Approach

The method presented now can be applied on fictitious potential approaches too. However, since the mesh quality approach seems preferable in terms of mesh quality, it will be used to demonstrate this performance improving technique.

The main drawback of the barrier methods introduced is their need of a sequence of calculations with increasing weighting parameters. Starting with weights that are too low yields an ill-conditioned system since their regularisation property strongly depends on the weighting parameter. Considering the weak convergence behaviour especially for lower weights, the procedures presented before (no matter if they use staggered or monolithic schemes) are expensive. Remembering the viscous regularisation method discussed in Section 4.5, with good regularisation behaviour but yielding inadmissible meshes (see 4.5.1), the idea is to use viscous regularisation as the main regularisation

method (applied on the Left-Hand-Side (LHS) only) and the mesh quality based barrier method (LHS and RHS) for conserving mesh quality and admissibility. This combination of viscous regularisation and barrier methods significantly reduces the initial weighting parameter for the barrier function and therefore the overall computational effort.
The algorithm is outlined in Table 4.3.

Table 4.3. Combination of barrier methods and viscous regularisation

```
set ALPHA_INCREASED = False
set viscous regularisation weight to initial value: $\alpha = \alpha_0$
set viscous regularisation updating parameter: $0 < \eta^{reg} < 1$
solve spatial motion problem
FIRST_R_STEP = True
while $||\boldsymbol{R}|| > \epsilon_1$ or FIRST_R_STEP do
    FIRST_R_STEP = False
    assemble r-adaptivity equations for staggered or coupled scheme
    apply viscous regularisation on the LHS using equation (4.5.3)[1]
    compute incremental displacements (whether with staggered or coupled scheme)
    check mesh properties: e.g. element distortion, Jacobians
    while mesh inadmissible do
        increase viscous regularisation weight: $\alpha \leftarrow \alpha/\eta^{reg}$
        apply viscous regularisation on the LHS using equation (4.5.3)
        set ALPHA_INCREASED = True
        compute incremental displacements
        check mesh
    end while
    compute new $||R||^k$ (convergence criterion for material motion)
    if ALPHA_INCREASED == False then
        compute convergence rate:
        $CR = \left(||RHS||^k - ||RHS||^{k-1}\right) / ||RHS||^{k-1}$
        if $CR > CR_{limit}$ then
            decrease viscous regularisation weight: $\alpha \leftarrow \alpha \cdot \eta^{reg}$
        end if
    end if
end while
```

[1] The viscous term on the RHS of equation (4.5.2) has been omitted because of difficulties arising from bad predictors.

4.6.3. Optimality

In the present case there is an unconstrained problem (the pure r-adaptivity problem) and a constrained problem formulated in terms of the unconstrained problem plus barrier functions enforcing inequality constraints. The vanishing RHS norm or gradient norm respectively acts as the optimality condition. Since only the constrained problem has been solved, the point of optimality has been found for the more or less artificially constrained problem and a specific weighting parameter γ only. Thus, one has to assume that the solution obtained is near the solution of the unconstrained problem. However, lowering the weighting parameter γ can bring the constrained problem near to the unconstrained one. Consequently, the RHS norm of the unconstrained problem at the final state obtained by solving the constrained problem is not zero (optimal) in general.
Thus, two quantities will be used to compare solutions obtained by different barrier and regularisation methods:

- Total potential internal energy Π_{tot} and
- the material gradient norm ($||\boldsymbol{R}||$) of Π_{tot} that is the norm of the vector of nodal configurational forces,

both of them evaluated for the unconstrained problem at the final state of the constrained problem.

Convergence criterion: In all following computations, as convergence criterion for material motion the (constrained system's) RHS norm has been used. It has been preferred over a criterion using unbalanced energy for better comparability with results given in publications of Scherer et al. (2007; 2008).

4.6.4. Comparison of Solution Strategies

In the following, comparison of different aspects of the fictitious potential, mesh quality and the combined regularisation approach will be presented in figures 4.6.3 to 4.6.6.

Because not only regularisation approaches but also the scheme types are compared, the abbreviations “M” denoting monolithic and “S” denoting staggered schemes in the legends of figures 4.6.3 to 4.6.6 are introduced. “VR” stands for viscous regularisation and therefore denotes combined methods.

Fictitious potential parameters: This method has been initialised with $\gamma = 0.5$, $\eta = 0.7$ and $\kappa = 1.0$. The coefficients for reduction and increase of κ have been set to $c_{red} = 0.6$ and $c_{inc} = 1.6666$. Volumetric change is limited a 0.005.

Mesh quality parameters: The initial weighting parameter was $\gamma = 5.0$, the distortion limiting parameter $\delta = 0.2$. Other parameters: $\kappa = 1.0$, $c_{red} = 0.6$ and $c_{inc} = 1.6666$. Additionally, the example was solved using the monolithic scheme with $\delta = 0.1$.

Combined method parameters: $\gamma = 1 \times 10^{-4}$ and $\delta = 0.2$. Other parameters: $\kappa = 1.0$, $c_{red} = 0.6$ and $c_{inc} = 1.6666$. For the viscous regularisation parameter $\alpha = 1 \times 10^{4}$ and $\eta^{visc} = 0.3$. The other parameters κ^{visc}, c_{red}^{visc} and c_{inc}^{visc} were all set to 1.

Convergence limits: A value of $||RHS|| = 1 \times 10^{-8}$ was set as convergence limit for every method. Since coupled schemes deal with larger systems, they tend to have higher residuals even if the solution has the same quality as the solution obtained with staggered schemes.

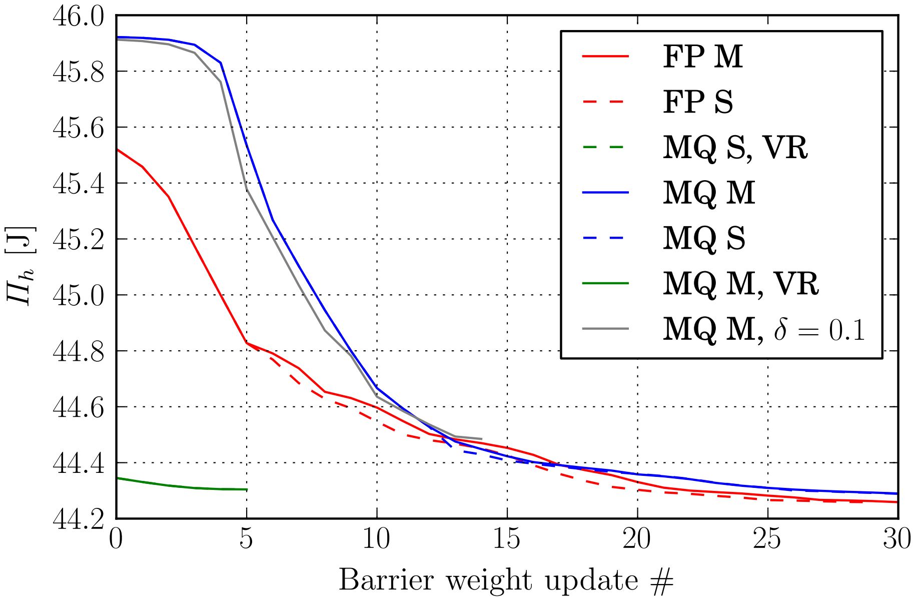

Figure 4.6.3. Model problem 3 - comparison: decrease of discrete potential energy

Due to the rather low initial mesh quality weighting parameter of the combined method, the highest initial (weight update 0) decrease in potential internal energy has been achieved. However, in the long run the methods not using viscous regularisation yield a lower final internal energy. It can be seen that the fictitious potential approach has some weaknesses concerning nodal configurational forces, since, despite lower final internal energy than the pure mesh quality approach, the final result yields a higher material gradient norm $||\boldsymbol{R}_h||$ (see 4.6.4). The pure mesh quality approach as well as the fictitious potential approach were aborted because of stagnation (weight updating factor $\eta^k \approx 1$) in terms of fictitious potential weight γ. Obviously, the additional viscous regularisation helped to overcome this deficiency. However, it turned out that an admissible mesh alone could not prevent occurrence of stagnation. Indeed, the sequence of the viscous parameters α, determined by initial values and updating factors, has great influence on

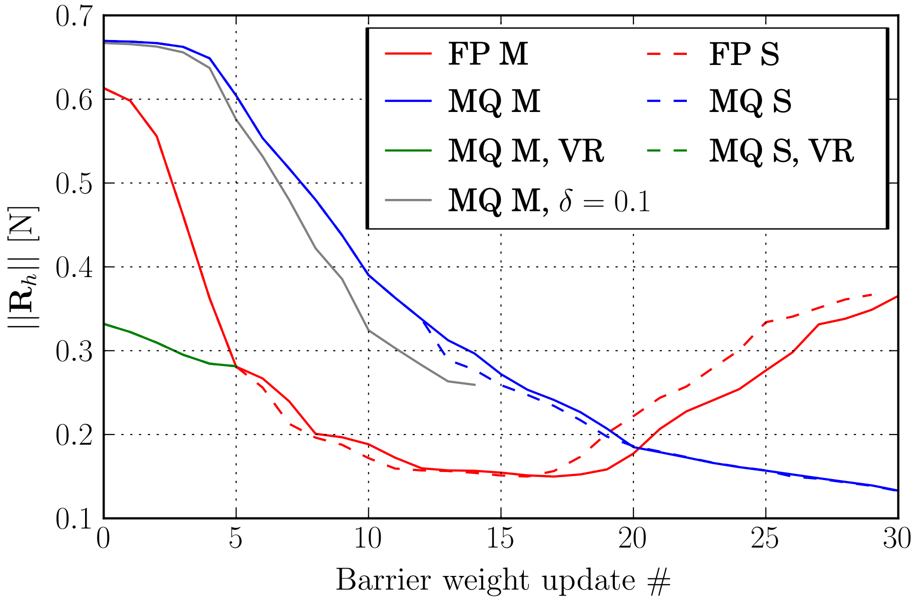

Figure 4.6.4. Model problem 3 - comparison: norm of nodal configurational forces

convergence. A proper (and computationally cheap) criterion for viscous regularisation updating has not been found. Especially the final results obtained by both combined approaches cannot competed with those obtained by the pure mesh quality approach. Thus, there may be some optimisation potential in this naive combined procedure. Line search methods may be a way out and will be considered in further investigations. Because of the staggered calculation of material motion and spatial equilibrium, line search is more expensive for staggered schemes than for coupled schemes. While at any line search point in Newton direction obtained by a coupled scheme the line search criterion (e.g. total potential energy) can be computed directly, in case of staggered schemes, the Newton direction is obtained from the computed material motion. Therefore, when using staggered schemes, for each line search point the spatial equilibrium has to be recovered before the criterion can be computed. Consequently, when using line search the coupled scheme catches up or even outperforms the staggered scheme in terms of overall computational effort per iteration. Remarkably, the plain mesh quality approach has a rather weak convergence behaviour and is inferior in every aspect of performance. However, using the combined approach, the weighting parameter for mesh quality may be chosen lower than the value reached by the pure mesh quality approach. Unfortunately, a lower barrier weight does not necessarily mean better results in terms of internal energy and material forces. The main advantage of the combined approach is the resistance against stagnation of the Newton-Raphson solver. Note that Figures 4.6.3 to 4.6.6 consider successful runs only.

Remark 4.3. Methods not using viscous regularisation may suffer from stagnation during the Newton-Raphson procedure. Therefore, a stagnation detection and additionally,

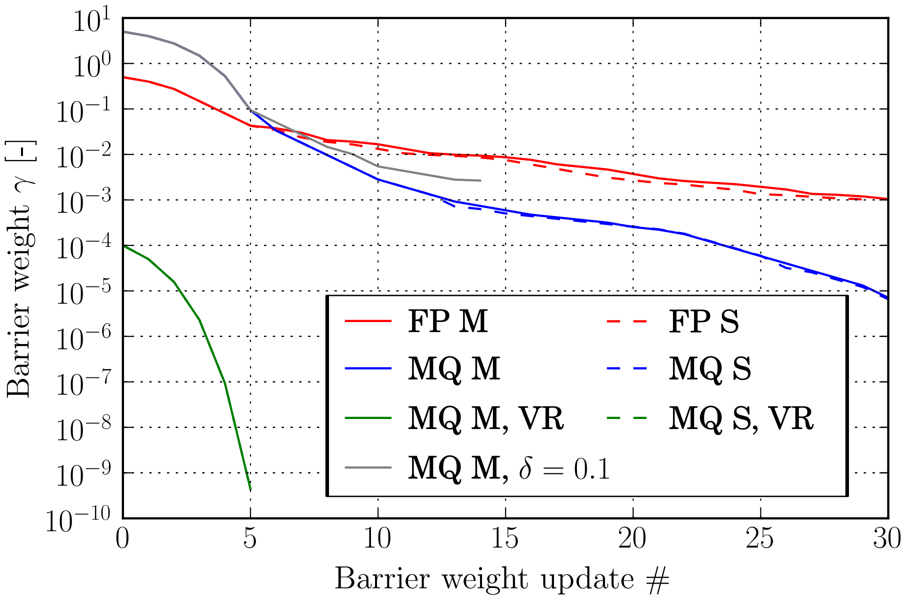

Figure 4.6.5. Model problem 3 - comparison: decrease of weighting parameters

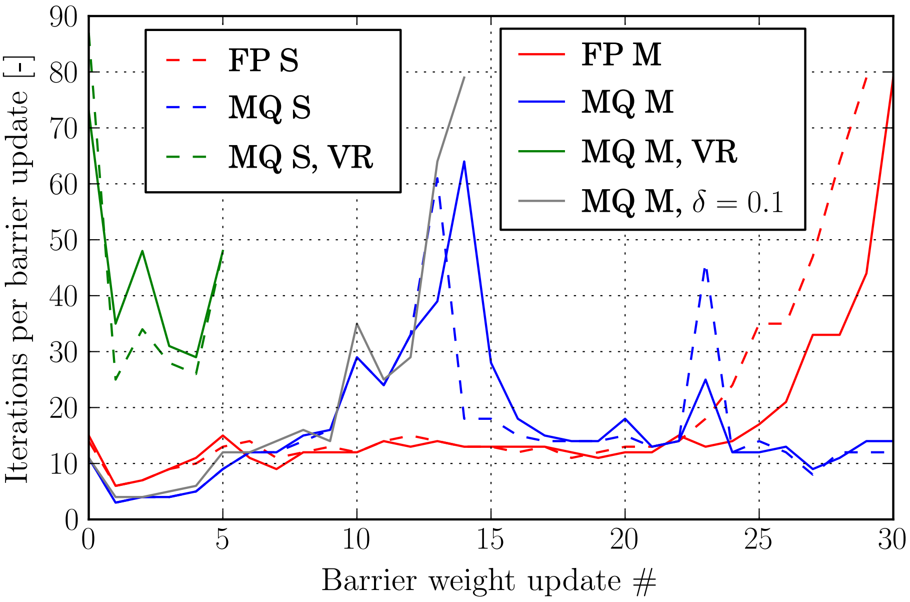

Figure 4.6.6. Model problem 3 - comparison: interations until convergence for each weight update

an upper limit for the number of Newton iterations were applied. If the procedure had stagnated or not converged within that limiting number of iterations, the weighting parameter γ was increased and Newton-Raphson procedure was restarted. Only successful (meaning converged within the maximum number of iterations) Newton-Raphson procedures were considered in Figures 4.6.3 to 4.6.5. The maximum number of iteration per weight update was set to 80, which can be seen in Figure 4.6.6.

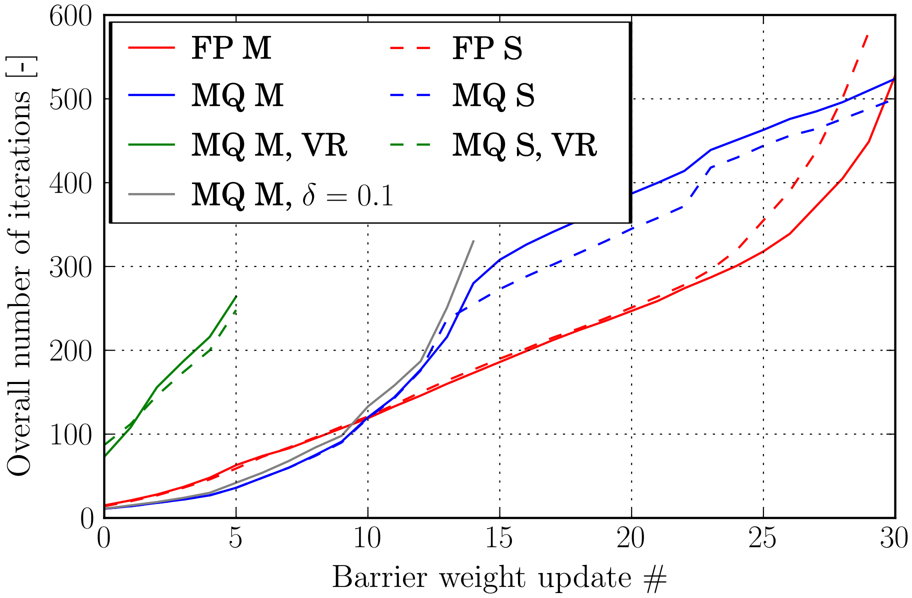

Figure 4.6.7. Model problem 3 - comparison: Accumulated iterations of successful Newton-Raphson procedures

Finally, Figure 4.6.8 shows the final meshes. The final mesh belonging to the pure mesh quality method with limit $\delta = 0.1$ is not depicted, because this setup suffered from stagnation in weight updates and did not yield interesting results. Whether an approach used a staggered or a monolithic scheme had no significant influence on the final mesh as demonstrated by meshes c) and d) in Figure 4.6.8.

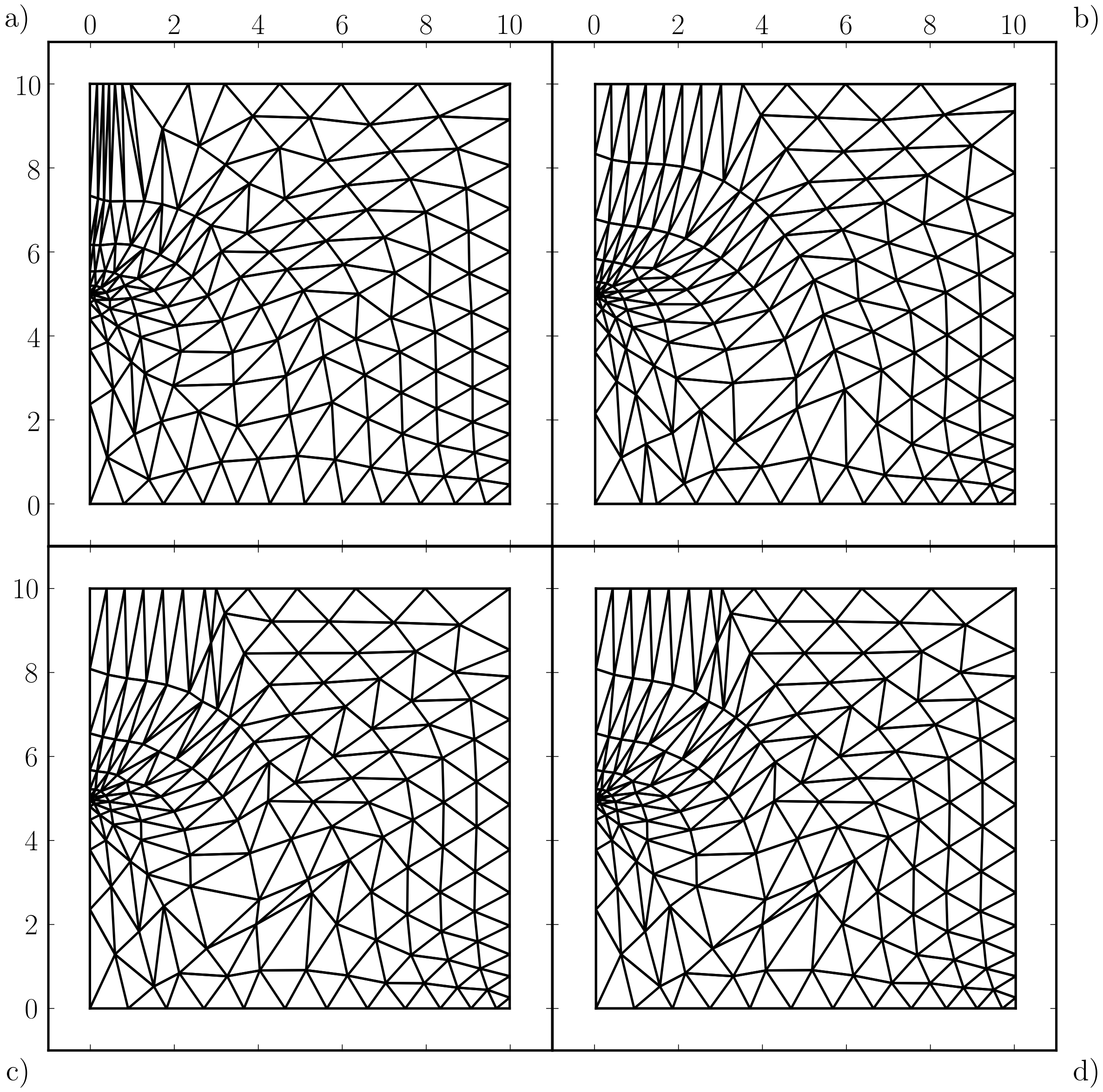

Figure 4.6.8. Model problem 3 - comparison: final meshes obtained with fictitious potential approach a), mesh quality approach b) and combined method with staggered c) and monolithic scheme d)

5. Conclusion

Starting from a variational ALE formulation applied to hyperelastostatics a variational framework for r-adaptive FEA has been presented. The solution method is based on the Newton-Raphson procedure. At first, the plain r-adaptivity framework was applied to very basic examples and soon it turned out that there are some numerical difficulties such as indefinite system matrices. These numerical problems were demonstrated and discussed on various simple examples. Then several regularisation approaches to overcome these difficulties introduced by other authors were tested and compared. Continuing with regularisation methods, a rather simple combination of a viscous regularisation and a mesh-quality-based approach was introduced. Finally, the most promising approaches were applied to the model problem of a cracked specimen.

As discussed in Section 3.2, r-adaptive FEA yields several challenges on the solution algorithm. In fact, regularisation and stabilisation methods have great influence on the overall performance. The most promising regularisation approaches are based on barrier methods. However, the barrier functions introduced were suitable for linear triangular and tetrahedral shapes only. Furthermore, the computational efficiency of the barrier methods presented needs further improvement. The combination of barrier methods and viscous regularisation seems promising concerning performance. The unsatisfying behaviour of the combined method concerning energy minimisation observed in Section 4.6 may result from the non-convexity of the r-adaptivity problem. A further consequence of non-convexity is that the Newton-Raphson method may converge to a saddle point instead of a local minimum. Furthermore, there may exist several local minima. Consequently, a stationary point obtained by a Newton-Raphson type procedure is not guaranteed to be the "best" result concerning energy minimisation in the presented vALE r-adaptivity framework.

Appendix A.

Derivations

A.1. Basics

Before the neo-Hookean material specific derivatives used in this work will be derived, some basic operations and techniques are shown.

Derivative of a tensor determinant with respect to the tensor itself (Bonet and Wood 1997)

$$
\begin{aligned}
J &= \det \boldsymbol{F} \\
\frac{\partial J}{\partial \boldsymbol{F}} &= J \cdot \boldsymbol{F}^{-T} && \text{(A.1.1)} \\
\frac{\partial J}{\partial F_{ij}} &= J\, f_{ji}, && \text{(A.1.2)}
\end{aligned}
$$

and using the chain rule

$$
\begin{aligned}
\frac{\partial \ln J}{\partial F_{ij}} &= \frac{\partial \ln J}{\partial J} \frac{\partial J}{\partial F_{ij}} \\
&= \frac{1}{J}\, J\, f_{ji} = f_{ji}. && \text{(A.1.3)}
\end{aligned}
$$

This also enables to compute,

$$
\begin{aligned}
\frac{\partial \frac{1}{J}}{\partial F_{ij}} &= -\frac{1}{J^2} \frac{\partial J}{\partial F_{ij}} \\
&= -\frac{1}{J} f_{ji}. && \text{(A.1.4)}
\end{aligned}
$$

Derivative of a tensor with respect to its inverse

$$\begin{aligned}
F_{im}f_{mj} &= \delta_{ij} \\
\frac{\partial\left(F_{im}f_{mj}\right)}{\partial f_{kl}} &= 0 \qquad \rightarrow \text{using the product rule} \\
\frac{\partial F_{im}}{\partial f_{kl}}f_{mj} &= -F_{im}\frac{\partial f_{mj}}{\partial f_{kl}} = -F_{im}\delta_{mk}\delta_{jl} \quad \mid \cdot F_{jn} \\
\frac{\partial F_{im}}{\partial f_{kl}}\delta_{mn} &= -F_{im}\delta_{mk}\delta_{jl}F_{jn} = -F_{ik}F_{ln} \quad \mid n \rightarrow j \\
\frac{\partial F_{ij}}{\partial f_{kl}} &= -F_{ik}F_{lj},
\end{aligned} \tag{A.1.5}$$

and swapping i and j yields the derivative of the transpose

$$\frac{\partial F_{ji}}{\partial f_{kl}} = -F_{li}F_{jk}. \tag{A.1.6}$$

A.1.1. Transformation of the Derivative of a Scalar Entity with Respect to the Deformation Gradient

For this derivation, $\mathrm{D}_{\boldsymbol{F}}$ will be written as $\frac{\partial\{\bullet\}}{\partial \boldsymbol{F}}$. Remembering equation (1.2.15),

$$\{\bullet\}_0 = J\{\bullet\}_t$$

the transformation of $\frac{\partial\{\bullet\}_0}{\partial F_{ij}}$ is obtained as follows:

$$\begin{aligned}
\frac{\partial\{\bullet\}_0}{\partial F_{ij}} &= \frac{\partial J\{\bullet\}_t}{\partial F_{ij}} = \{\bullet\}_t\frac{\partial J}{\partial F_{ij}} + J\frac{\partial\{\bullet\}_t}{\partial F_{ij}} = \{\bullet\}_t\, Jf_{ji} + J\frac{\partial\{\bullet\}_t}{\partial f_{kl}}\frac{\partial f_{kl}}{\partial F_{ij}} \\
&= \{\bullet\}_0 f_{ji} - J\frac{\partial\{\bullet\}_t}{\partial f_{kl}}f_{ki}f_{jl},
\end{aligned} \tag{A.1.7}$$

In matrix notation this reads as,

$$\frac{\partial\{\bullet\}_0}{\partial \boldsymbol{F}} = \{\bullet\}_0\boldsymbol{f}^T - J\boldsymbol{f}^T\frac{\partial\{\bullet\}_t}{\partial \boldsymbol{f}}\boldsymbol{f}^T \tag{A.1.8}$$

A.2. Derivatives of the Neo-Hookean Energy Density Function

A.2.1. Spatial Motion Entities

The energy function for compressible neo-Hookean material is given as

$$\Psi_0^{int} = \frac{1}{2}\lambda_0(\ln J)^2 + \frac{1}{2}\mu_0\left(F_{ij}F_{ij} - n^{dim} - 2\ln J\right), \tag{A.2.1}$$

for the spatial motion problem in terms of the spatial motion deformation gradient.
To obtain the first Piola-Kirchhoff stress tensor, the derivative of the neo-Hookean energy function with respect to $\boldsymbol{F}$:

$$\begin{aligned} P_{ij} = \frac{\partial \Psi_0^{int}}{\partial F_{ij}} &= \frac{\partial\left[\frac{1}{2}\lambda_0(\ln J)^2 + \frac{1}{2}\mu_0\left(F_{ij}F_{ij} - n^{dim} - 2\ln J\right)\right]}{\partial F_{ij}} \\ &= \left(\lambda_0 \ln J - \mu_0\right) f_{ji} + \mu_0 F_{ij}, \end{aligned} \tag{A.2.2}$$

with $\boldsymbol{f} = \boldsymbol{F}^{-1}$. Using equation (1.3.4) and expressions $\lambda_t = \lambda_0/J, \mu_t = \mu_0/J$,

$$\sigma_{ij} = \frac{1}{J}P_{ik}F_{jk} = \left(\lambda_t \ln J - \mu_t\right)\delta_{ij} + \mu_t b_{ij}, \tag{A.2.3}$$

where $b_{ij} = F_{ik}F_{jk}$ is the left Cauchy-Green tensor.
Recalling the discretised spatial motion linearisations obtained in Section 3.1,

$$\int\limits_{V_0} \mathbb{J}_{mj}\tilde{A}_{ijkl}\mathbb{J}_{nl}\,\mathrm{d}V_0 = \int\limits_{V_0} \mathbb{J}_{mj}\frac{\partial^2\Psi_0^{int}}{\partial F_{kl}\partial F_{ij}}\mathbb{J}_{nl}\,\mathrm{d}V_0 \tag{A.2.4}$$

$$\int\limits_{V_t} \mathring{\mathbb{j}}_{mj}\tilde{b}_{ijkl}\,\mathring{\mathbb{j}}_{nl}\,\mathrm{d}V_t = \int\limits_{V_t} \mathring{\mathbb{j}}_{mj}\frac{\partial \sigma_{ij}}{\partial f_{kl}}\,\mathring{\mathbb{j}}_{nl}\,\mathrm{d}V_t \tag{A.2.5}$$

Therein,

$$\begin{aligned} \frac{\partial^2\Psi_0^{int}}{\partial F_{kl}\partial F_{ij}} &= \frac{\partial P_{ij}}{\partial F_{kl}} = \lambda_0\frac{\partial \ln J}{\partial F_{kl}}f_{ji} + \left(\lambda_0\ln J - \mu_0\right)\frac{\partial f_{ji}}{\partial F_{kl}} + \mu_0\frac{\partial F_{ij}}{\partial F_{kl}} \\ &\rightarrow \text{using equations (A.1.3) and (A.1.6)} \\ \frac{\partial^2\Psi_0^{int}}{\partial F_{kl}\partial F_{ij}} &= \lambda_0 f_{ji}f_{lk} - \left(\lambda_0\ln J - \mu_0\right)f_{li}f_{jk} + \mu_0\delta_{ik}\delta_{jl} \end{aligned} \tag{A.2.6}$$

and using $J = 1/j$,

$$\begin{aligned} \frac{\partial \sigma_{ij}}{\partial f_{kl}} &= \left(\lambda_0\ln(J)\delta_{ij} - \mu_0\delta_{ij} + \mu_0 b_{ij}\right)\frac{\partial j}{\partial f_{kl}} + \lambda_t\delta_{ij}\frac{\partial\ln(1/j)}{\partial f_{kl}} + \mu_t\frac{\partial(F_{im}F_{jm})}{\partial f_{kl}} \\ &= \left(\lambda_t\ln(J)\delta_{ij} - \mu_t\delta_{ij} + \mu_t b_{ij}\right)F_{lk} - \lambda_t\delta_{ij}F_{lk} - \mu_t\left(F_{ik}F_{lm}F_{jm} + F_{im}F_{jk}F_{lm}\right) \\ &= \left(\lambda_t\left(\ln(J) - 1\right) - \mu_t\right)\delta_{ij}F_{lk} + \mu_t\left(F_{im}F_{jm}F_{lk} - F_{ik}F_{lm}F_{jm} - F_{im}F_{jk}F_{lm}\right) \end{aligned} \tag{A.2.7}$$

Now, remembering equations (3.1.23),

$$\mathbb{J}_{ij} = \mathbb{j}_{ik} F_{kj} \tag{A.2.8}$$

$$\mathbb{j}_{ij} = \mathbb{J}_{ik} f_{kj} \tag{A.2.9}$$

replacing all $\mathbb{J}$ and $\mathbb{j}$ in equation (A.2.4) results in,

$$\int\limits_{V_0} \mathbb{J}_{mj} \frac{\partial^2 \Psi_0^{int}}{\partial F_{kl} \partial F_{ij}} \mathbb{J}_{nl} \, \mathrm{d}V_0 \rightarrow$$

$$\begin{aligned}
\mathbb{J}_{mj} \frac{\partial^2 \Psi_0^{int}}{\partial F_{kl} \partial F_{ij}} \mathbb{J}_{nl} &= \mathbb{j}_{mo} F_{oj} \frac{\partial^2 \Psi_0^{int}}{\partial F_{kl} \partial F_{ij}} \mathbb{j}_{np} F_{pl} \\
&= \mathbb{j}_{mo} F_{oj} \left[\lambda_0 f_{ji} f_{lk} - \left(\lambda_0 \ln J - \mu_0 \right) f_{li} f_{jk} \mu_0 \delta_{ik} \delta_{jl} \right] \mathbb{j}_{np} F_{pl} \\
&= \mathbb{j}_{mo} \left[\lambda_0 \delta_{io} \delta_{kp} - \left(\lambda_0 \ln J - \mu_0 \right) \delta_{ip} \delta_{ok} + \mu_0 \delta_{ik} b_{op} \right] \mathbb{j}_{np}
\end{aligned} \tag{A.2.10}$$

Note that equations (3.1.18) give rather general expressions for the coupling tangent operators (A.2.5). The tangent operators as implemented for this work: Replacing the second $\mathbb{j}$ in equation (A.2.5),

$$\int\limits_{V_t} \mathbb{j}_{mj} \frac{\partial \sigma_{ij}}{\partial f_{kl}} \mathbb{j}_{nl} \, \mathrm{d}V_t \rightarrow$$

$$\begin{aligned}
\mathbb{j}_{mj} \frac{\partial \sigma_{ij}}{\partial f_{kl}} \mathbb{j}_{nl} &= \mathbb{j}_{mj} \frac{\partial \sigma_{ij}}{\partial f_{kl}} \mathbb{J}_{np} f_{pl} \\
&= \mathbb{j}_{mj} \left[\left[\lambda_t \left(\ln(J) - 1 \right) - \mu_t \right] \delta_{ij} F_{lk} + \mu_t \left(F_{io} F_{jo} F_{lk} - \right. \right. \\
&\quad \left. \left. F_{ik} F_{lo} F_{jo} - F_{io} F_{jk} F_{lo} \right) \right] \mathbb{J}_{np} f_{pl} \\
&= \mathbb{j}_{mj} \left[\left[\lambda_t \left(\ln(J) - 1 \right) - \mu_t \right] \delta_{ji} \delta_{pk} \right. \\
&\quad \left. + \mu_t \left(F_{io} F_{jo} \delta_{pk} - F_{ik} \delta_{po} F_{jo} - F_{io} F_{jk} \delta_{po} \right) \right] \mathbb{J}_{np} \\
&= \mathbb{j}_{mj} \left[\left[\lambda_t \left(\ln(J) - 1 \right) - \mu_t \right] \delta_{ij} \delta_{kp} \right. \\
&\quad \left. + \mu_t \left(b_{ij} \delta_{kp} - F_{ik} F_{jp} - F_{ip} F_{jk} \right) \right] \mathbb{J}_{np}
\end{aligned} \tag{A.2.11}$$

A.2.2. Material Motion Entities

Analogously, the linearisations of the material motion problem for neo-Hookean material can be obtained using the neo-Hookean energy function for material motion in the spatial domain in terms of the material motion deformation gradient,

$$\Psi_t^{int} = \frac{1}{2} \lambda_t \left(\ln j \right)^2 + \frac{1}{2} \mu_t \left(f_{ij}^{-1} f_{ij}^{-1} - n^{dim} + 2 \ln j \right) \tag{A.2.12}$$

and the Eshelby momentum tensor (also see equation (1.4.8)),

$$\Sigma_{ij} = J \, p_{ik} f_{jk}, \tag{A.2.13}$$

wherein

$$\begin{aligned}
p_{ij} = \frac{\partial \Psi_t^{int}}{\partial f_{ij}} &= \left[\frac{1}{2}\lambda_t \left(\ln j\right)^2 + \frac{1}{2}\mu_t \left(f_{mn}^{-1} f_{mn}^{-1} - n^{dim} + 2\ln j\right)\right] F_{ji} \\
&\quad + \lambda_t \ln j F_{ji} + \mu_t \left(\frac{\partial f_{mn}^{-1}}{\partial f_{ij}} f_{mn}^{-1} + F_{ji}\right) \\
&= \left(\Psi_t^{int} + \lambda_t \ln j + \mu_t\right) F_{ji} - \mu_t F_{mi} F_{jn} F_{mn} \\
&= \left(\Psi_t^{int} + \lambda_t \ln j + \mu_t\right) F_{ji} - \mu_t C_{in} F_{jn}, \qquad \text{(A.2.14)}
\end{aligned}$$

where $C_{ij} = F_{ki}F_{kj}$ is the right Cauchy-Green tensor (for rectangular Cartesian reference frame).

Now, having in mind the material motion part of equation 3.1.22,

$$\int\limits_{V_0} \mathbb{J}_{mj} \tilde{B}_{ijkl} \mathbb{J}_{nl} \,\mathrm{d}V_0 = \int\limits_{V_0} \mathbb{J}_{mj} \frac{\partial \Sigma_{ij}}{\partial F_{kl}} \mathbb{J}_{nl} \,\mathrm{d}V_0 \qquad \text{(A.2.15)}$$

$$\int\limits_{V_t} \mathring{\mathbb{j}}_{mj} \tilde{a}_{ijkl} \mathring{\mathbb{j}}_{nl} \,\mathrm{d}V_t = \int\limits_{V_t} \mathring{\mathbb{j}}_{mj} \frac{\partial^2 \Psi_t^{int}}{\partial f_{kl} \partial f_{ij}} \mathring{\mathbb{j}}_{nl} \,\mathrm{d}V_t, \qquad \text{(A.2.16)}$$

one is interested in

$$\begin{aligned}
\frac{\partial^2 \Psi_t^{int}}{\partial f_{kl} \partial f_{ij}} &= \frac{\partial p_{ij}}{\partial f_{kl}} \quad \text{and} \\
\frac{\partial \Sigma_{ij}}{\partial F_{kl}} &= \frac{\partial \left(\Psi_0^{int}\, \delta_{ij} - F_{mi} P_{mj}\right)}{\partial F_{kl}}.
\end{aligned}$$

Beginning with,

$$\begin{aligned}
\frac{\partial p_{ij}}{\partial f_{kl}} &= \left(\frac{\partial \Psi_t^{int}}{\partial f_{kl}} + \lambda_t \ln j F_{lk} + \lambda_t F_{lk} + \mu_t F_{lk}\right) F_{ji} \\
&\quad + \left(\Psi_t^{int} + \lambda_t \ln j + \mu_t\right) \frac{\partial F_{ji}}{\partial f_{kl}} \\
&\quad - \mu_t \left(C_{in} F_{jn} F_{lk} \frac{\partial F_{mi}}{\partial f_{kl}} F_{mn} F_{jn} + F_{mi} \frac{\partial F_{mn}}{\partial f_{kl}} F_{jn} + F_{mi} F_{mn} \frac{\partial F_{jn}}{\partial f_{kl}}\right) \\
&= F_{ji}\, p_{kl} + \left[\lambda_t \left(\ln j + 1\right) + \mu_t\right] F_{ji} F_{lk} \\
&\quad - \left(\Psi_t^{int} + \lambda_t \ln j + \mu_t\right) F_{li} F_{jk} \\
&\quad - \mu_t \left(C_{in} F_{jn} F_{lk} - F_{li} F_{mk} F_{mn} F_{jn}\right. \\
&\quad \left. - F_{mi} F_{ln} F_{mk} F_{jn} - F_{mi} F_{mn} F_{ln} F_{jk}\right) \\
&= F_{ji} \left[\left(\Psi_t^{int} + \lambda_t \ln j + \mu_t\right) F_{lk} - \mu_t C_{kn} F_{ln}\right] \\
&\quad + \left[\lambda_t \left(\ln j + 1\right) + \mu_t\right] F_{ji} F_{lk} \\
&\quad - \left(\Psi_t^{int} + \lambda_t \ln j + \mu_t\right) F_{li} F_{jk} \\
&\quad - \mu_t \left(C_{in} F_{jn} F_{lk} - F_{li} F_{mk} b_{jm} - C_{ik} b_{jl} - C_{in} F_{ln} F_{jk}\right). \qquad \text{(A.2.17)}
\end{aligned}$$

Then,

$$
\begin{aligned}
\frac{\partial \Sigma_{ij}}{\partial F_{kl}} &= \delta_{ij} P_{kl} - \delta_{mk}\delta_{il} P_{mj} - F_{mi}\frac{\partial P_{mj}}{\partial F_{kl}} \\
&= \delta_{ij} P_{kl} - \delta_{il} P_{kj} - F_{mi}\frac{\partial P_{mj}}{\partial F_{kl}} \\
&= \left(\lambda_0 \ln J - \mu_0\right)\delta_{ij} f_{lk} + \mu_0 \delta_{ij} F_{kl} \\
&\quad - \left[\left(\lambda_0 \ln J - \mu_0\right)\delta_{il} f_{jk} + \mu_0 \delta_{il} F_{kj}\right] \\
&\quad - \left[\lambda_0 \delta_{ij} f_{lk} - \left(\lambda_0 \ln J - \mu_0\right)\delta_{il} f_{jk} + \mu_0 F_{ki}\delta_{jl}\right]. \qquad \text{(A.2.18)}
\end{aligned}
$$

Next, using equations (A.2.8) and (A.2.9),

$$
\int\limits_{V_t} \mathring{\mathbb{J}}_{mj} \frac{\partial^2 \Psi_t^{int}}{\partial f_{kl}\partial f_{ij}} \mathring{\mathbb{J}}_{nl}\, \mathrm{d}V_t \rightarrow
$$

$$
\begin{aligned}
\mathring{\mathbb{J}}_{mj} \frac{\partial^2 \Psi_t^{int}}{\partial f_{kl}\partial f_{ij}} \mathring{\mathbb{J}}_{nl} &= \mathbb{J}_{mo} f_{oj} \frac{\partial^2 \Psi_t^{int}}{\partial f_{kl}\partial f_{ij}} \mathbb{J}_{np} f_{pl} \\
&= \mathbb{J}_{mo}\{ f_{oj} F_{ji} \left[\left(\Psi_t^{int} + \lambda_t \ln j + \mu_t\right) f_{pl} F_{lk} - \mu_t C_{kn} f_{pl} F_{ln}\right] \\
&\quad + \left[\lambda_t \left(\ln j + 1\right) + \mu_t\right] f_{oj} F_{ji} f_{pl} F_{lk} \\
&\quad - \left(\Psi_t^{int} + \lambda_t \ln j + \mu_t\right) f_{pl} F_{li} f_{oj} F_{jk} \\
&\quad - \mu_t \left(C_{in} f_{oj} F_{jn} f_{pl} F_{lk} - f_{pl} F_{li} F_{mk} f_{oj} b_{jm}\right. \\
&\quad \left. - C_{ik} f_{oj} f_{pl} b_{jl} - C_{in} f_{pl} F_{ln} f_{oj} F_{jk}\right)\} \mathbb{J}_{np} \\
&= \mathbb{J}_{mo}\{ \left(\Psi_t^{int} + \lambda_t \ln j + \mu_t\right) \delta_{io}\delta_{kp} - \mu_t \delta_{io} C_{kp} \\
&\quad + \left[\lambda_t \left(\ln j + 1\right) + \mu_t\right] \delta_{io}\delta_{kp} \\
&\quad - \left(\Psi_t^{int} + \lambda_t \ln j + \mu_t\right) \delta_{ip}\delta_{ok} \\
&\quad - \mu_t \left(C_{io}\delta_{kp} - \delta_{ip} C_{ok}\right. \\
&\quad \left. - C_{ik}\delta_{op} - C_{ip}\delta_{ok}\right)\} \mathbb{J}_{np} \\
&= \mathbb{J}_{mo}\{ \left(\Psi_t^{int} + 2\lambda_t \ln j + \lambda_t + 2\mu_t\right) \delta_{io}\delta_{kp} \\
&\quad - \left(\Psi_t^{int} + \lambda_t \ln j + \mu_t\right) \delta_{ip}\delta_{ok} \\
&\quad - \mu_t \left(C_{io}\delta_{kp} + \delta_{io} C_{kp} - \delta_{ip} C_{ok}\right. \\
&\quad \left. - C_{ik}\delta_{op} - C_{ip}\delta_{ok}\right)\} \mathbb{J}_{np} \qquad \text{(A.2.19)}
\end{aligned}
$$

Once again, the coupling tangent operator is not implemented as given in equation (3.1.18). The coupling tangent operator in its implemented form is derived below: Replacing the second $\mathbb{J}$ in equation A.2.18,

$$
\begin{aligned}
&\int\limits_{V_0} \mathbb{J}_{mj} \frac{\partial \Sigma_{ij}}{\partial F_{kl}} \mathbb{J}_{nl} \, \mathrm{d}V_0 \rightarrow \\
&\qquad \mathbb{J}_{mj} \frac{\partial \Sigma_{ij}}{\partial F_{kl}} \mathbb{J}_{nl} = \mathbb{J}_{mj} \frac{\partial \Sigma_{ij}}{\partial F_{kl}} \mathring{\mathbb{j}}_{np} F_{pl} \\
&\qquad = \mathbb{J}_{mj} \{ (\lambda_0 \ln J - \mu_0) \, \delta_{ij} F_{pl} f_{lk} + \mu_0 \delta_{ij} F_{pl} F_{kl} \\
&\qquad\quad - [(\lambda_0 \ln J - \mu_0) \, \delta_{il} F_{pl} f_{jk} + \mu_0 \delta_{il} F_{pl} F_{kj}] \\
&\qquad\quad - [\lambda_0 \delta_{ij} F_{pl} f_{lk} - (\lambda_0 \ln J - \mu_0) \, \delta_{il} F_{pl} f_{jk} + \mu_0 F_{ki} \delta_{jl} F_{pl}] \} \mathring{\mathbb{j}}_{np} \\
&\qquad = \mathbb{J}_{mj} \{ (\lambda_0 \ln J - \mu_0) \, \delta_{ij} \delta_{kp} + \mu_0 \delta_{ij} b_{kp} \\
&\qquad\quad - [(\lambda_0 \ln J - \mu_0) \, F_{pi} f_{jk} + \mu_0 F_{pi} F_{kj}] \\
&\qquad\quad - [\lambda_0 \delta_{ij} \delta_{kp} - (\lambda_0 \ln J - \mu_0) \, F_{pi} f_{jk} \\
&\qquad\quad + \mu_0 F_{ki} F_{pj}] \} \mathring{\mathbb{j}}_{np} \\
&\qquad = \mathbb{J}_{mj} \{ [\lambda_0 \, (\ln J - 1) - \mu_0] \, \delta_{ij} \delta_{kp} \\
&\qquad\quad + \mu_0 \, (\delta_{ij} b_{pk} - F_{pi} F_{kj} - F_{ki} F_{pj}) \} \mathring{\mathbb{j}}_{np}.
\end{aligned}
\tag{A.2.20}
$$

A.3. Material Derivatives of the Fictitious Potential

In Section 4.5.2, equation (4.5.4) the fictitious potential was given as,

$$
\mathit{\Psi}^{+} = \frac{1}{2(1-\delta)} \, f_{*ij} \, f_{*ij} - \ln (j_r - \delta) \, . \tag{A.3.1}
$$

The first derivative with respect to the mesh deformation gradient $\boldsymbol{f}_r$ yields,

$$
p^{+}_{ij} = \frac{\partial \mathit{\Psi}^{+}}{\partial f_{*ij}} = \frac{1}{(1-\delta)} f_{*ij} - \frac{j_r}{(j_r - \delta)} f^{-1}_{*ji}. \tag{A.3.2}
$$

Computing the second derivative,

$$
\begin{aligned}
\frac{\partial p^{+}_{ij}}{\partial f_{*kl}} &= \frac{1}{(1-\delta)} \delta_{ik} \delta_{jl} - \frac{j_r}{(j_r - \delta)} \frac{\partial f^{-1}_{*ji}}{\partial f_{*kl}} - \frac{j_r}{(j_r - \delta)} f^{-1}_{*ji} f^{-1}_{*lk} + \frac{{j_r}^2}{(j_r - \delta)^2} f^{-1}_{*ji} f^{-1}_{*lk} \\
&= \frac{1}{(1-\delta)} \delta_{ik} \delta_{jl} + \frac{j_r}{(j_r - \delta)} f^{-1}_{*li} f^{-1}_{*jk} + \frac{\delta j_r}{(j_r - \delta)^2} f^{-1}_{*ji} f^{-1}_{*lk}
\end{aligned}
\tag{A.3.3}
$$

A.4. Material Derivatives of the Mesh Quality Barrier Function

The mesh quality based barrier function presented in Section 4.5.3, equation (4.5.12) was given as,

$$\mathit{\Pi}^{+} = \frac{-1}{\delta - \upsilon^{-1}}, \tag{A.4.1}$$

with

$$\upsilon = \frac{f_{*ij} f_{*ij}}{\left(j_r\right)^{2/n^{dim}}}, \tag{A.4.2}$$

where the superscript $^{(m)}$ indicating evaluating on element m is omitted to reduce clutter.

Note that there is need for an integration over the domain, because a single element's quality should not depend on its size.
Computing the first derivative with respect to $\boldsymbol{f}_*$,

$$\frac{\partial \mathit{\Pi}^{+}}{\partial f_{*ij}} = \frac{1}{\left(\delta - \upsilon^{-1}\right)^2}\, \upsilon^{-2} \frac{\partial \upsilon}{\partial f_{*ij}}, \tag{A.4.3}$$

with

$$\frac{\partial \upsilon}{\partial f_{*ij}} = 2\left(\frac{f_{*ij}}{\left(j_r\right)^{2/n^{dim}}} - \frac{\upsilon}{n^{dim}} f^{-1}_{*ji}\right), \tag{A.4.4}$$

and the second derivative,

$$\begin{aligned}\frac{\partial^2 \mathit{\Pi}^{+}}{\partial f_{*kl} \partial f_{*ij}} =& \frac{-2}{\left(\delta - \upsilon^{-1}\right)^3}\, \upsilon^{-4} \frac{\partial \upsilon}{\partial f_{*ij}} \frac{\partial \upsilon}{\partial f_{*kl}} \\ &- \frac{2}{\left(\delta - \upsilon^{-1}\right)^2}\, \upsilon^{-3} \frac{\partial \upsilon}{\partial f_{*ij}} \frac{\partial \upsilon}{\partial f_{*kl}} \\ &+ \frac{1}{\left(\delta - \upsilon^{-1}\right)^2}\, \upsilon^{-2} \frac{\partial^2 \upsilon}{\partial f_{*kl} \partial f_{*ij}},\end{aligned} \tag{A.4.5}$$

with

$$\begin{aligned}\frac{\partial^2 \upsilon}{\partial f_{*kl} \partial f_{*ij}} =& 2\left[\left(\frac{\delta_{ik}\delta_{jl}}{\left(j_r\right)^{2/n^{dim}}} - \frac{2}{n^{dim}} \frac{f_{*ij} f^{-1}_{*lk}}{\left(j_r\right)^{2/n^{dim}}}\right) - \frac{1}{n^{dim}}\left(f^{-1}_{*ji} \frac{\partial \upsilon}{\partial f_{*kl}} + \upsilon \frac{\partial f^{-1}_{*ji}}{\partial f_{*kl}}\right)\right] \\ =& \frac{2}{n^{dim}}\left[\frac{n^{dim}\delta_{ik}\delta_{jl} - 2 f_{*ij} f^{-1}_{*lk}}{\left(j_r\right)^{2/n^{dim}}} - \left(f^{-1}_{*ji} \frac{\partial \upsilon}{\partial f_{*kl}} - \upsilon f^{-1}_{li} f^{-1}_{jk}\right)\right]\end{aligned} \tag{A.4.6}$$

Appendix B.

Software

Several software packages and libraries have been used within the code written for the computations presented in this work. The code itself is mainly written in Python (Version 2.x). Numerics were primarily done with "NumPy" and "SciPy" classes. Some tensor operations were implemented in C++ using Blitz++. Figures were generated with "matplotlib" and "ParaView", a VTK viewer. Coding itself was done with Eclipse using the PyDev plug-in. In fact, the list of software and tools is far too long to be completely listed here. Nevertheless, an incomplete listing is given below:

- Python
  - The Python programming language and interpreter (http://www.python.org),
  - IPython - an interactive Python shell (http://ipython.org) (Pérez and Granger 2007),
  - NumPy and SciPy - numeric and scientific library (http://www.numpy.org, http://www.scipy.org),
  - Matplotlib - a plotting library (http://matplotlib.org/) (Hunter 2007),
  - PyVTK - library for writing VTK files (http://cens.ioc.ee/projects/pyvtk/),
  - PySoofea - Michael Hammer's Software for Object Oriented Finite Element Analysis in Python, that delivered the foundations for the final software (http://www.soofea.org/, Python version unpublished),
  - PyLint - code analysis (http://pylint.org/), ...
- Blitz++ - C++ class library for scientific computing featuring index operations

(http://sourceforge.net/projects/blitz/)

- ParaView - scientific data visualisation (http://www.paraview.org/),
- Eclipse - IDE (http://www.eclipse.org/), with plug-ins for Python (http://pydev.org/) and C/C++ (Eclipse CDT),
- Gmsh - a mesh generating tool (http://geuz.org/gmsh/),
- Inkscape - vector graphics editor (http://inkscape.org/),
- Kile - a LaTeX-editor (http://kile.sourceforge.net/), and many others

Bibliography

Harm Askes, Ellen Kuhl, and Paul Steinmann. An ALE formulation based on spatial and material settings of continuum mechanics. part 2: Classification and applications. *Computer Methods in Applied Mechanics and Engineering*, 193(39–41): 4223 – 4245, 2004. ISSN 0045-7825. doi: 10.1016/j.cma.2003.09.031. URL http://www.sciencedirect.com/science/article/pii/S0045782504002208.

Klaus-Jürgen Bathe. *Finite Element Procedures*. Prentice Hall, Inc, 1996. ISBN 0-13-301458-4.

J. Bonet and R. D. Wood. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, 2nd edition, 1997.

Stephen Boyd and Lieven Vandenberghe. *Covex Optimization*. Cambridge University Press, 2004.

Christoph Glocker. The principles of d'Alembert, Jourdain, and Gauss in nonsmooth dynamics part i: Scleronomic multibody systems. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 78(1):21–37, 1998. ISSN 1521-4001. doi: 10.1002/(SICI)1521-4001(199801)78:1⟨21::AID-ZAMM21⟩3.0.CO;2-W. URL http://dx.doi.org/10.1002/(SICI)1521-4001(199801)78:1<21::AID-ZAMM21>3.0.CO;2-W.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

Ellen Kuhl, Harm Askes, and Paul Steinmann. An ALE formulation based on spatial and material settings of continuum mechanics. part 1: Generic hyperelastic formulation. *Computer Methods in Applied Mechanics and Engineering*, 193(39–41): 4207 – 4222, 2004. ISSN 0045-7825. doi: 10.1016/j.cma.2003.09.030. URL http://www.sciencedirect.com/science/article/pii/S0045782504002191.

J. Mosler and M. Ortiz. On the numerical implementation of variational arbitrary Lagrangian–Eulerian (VALE) formulations. *International Journal for Numerical Methods in Engineering*, 67(9):1272–1289, 2006. ISSN 1097-0207. doi: 10.1002/nme.1621. URL http://dx.doi.org/10.1002/nme.1621.

R. Mueller and G. A. Maugin. On material forces and finite element discretizations. *Computational Mechanics*, 29:52–60, 2002. ISSN 0178-7675. doi: 10.1007/s00466-002-0322-2. URL http://dx.doi.org/10.1007/s00466-002-0322-2.

H. Parkus. *Mechanik der festen Körper.* Springer-Verlag Vienna, 1960.

Fernando Pérez and Brian E. Granger. IPython: a System for Interactive Scientific Computing. *Comput. Sci. Eng.*, 9(3):21–29, May 2007. URL http://ipython.org.

Michael Scherer, Ralf Denzer, and Paul Steinmann. Energy-based r-adaptivity: a solution strategy and applications to fracture mechanics. *International Journal of Fracture*, 147:117–132, 2007. ISSN 0376-9429. doi: 10.1007/s10704-007-9143-9. URL http://dx.doi.org/10.1007/s10704-007-9143-9.

Michael Scherer, Ralf Denzer, and Paul Steinmann. On a solution strategy for energy-based mesh optimization in finite hyperelastostatics. *Computer Methods in Applied Mechanics and Engineering*, 197(6–8):609 – 622, 2008. ISSN 0045-7825. doi: 10.1016/j.cma.2007.08.005. URL http://www.sciencedirect.com/science/article/pii/S0045782507003635.

P. Steinmann. On spatial and material settings of hyperelastodynamics. *Acta Mechanica*, 156(3-4):193–218, 2002. ISSN 0001-5970. doi: 10.1007/BF01176756. URL http://dx.doi.org/10.1007/BF01176756.

P. Steinmann, M. Scherer, and R. Denzer. Secret and joy of configurational mechanics: From foundations in continuum mechanics to applications in computational mechanics. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 89(8):614–630, 2009. ISSN 1521-4001. doi: 10.1002/zamm.200800132. URL http://dx.doi.org/10.1002/zamm.200800132.

Pururav Thoutireddy. *Variational Arbitrary Lagrangiean-Eulerian method.* PhD thesis, California Institute of Technology, Pasadena, California, 2003.

Michael Ulbrich and Stefan Ulbrich. *Nichtlineare Optimierung.* Springer Basel AG, 2nd edition, 2011. ISBN 978-3-0346-0142-9.

Baisheng Wu, Zhonghai Xu, and Zhengguang Li. A note on imposing displacement boundary conditions in finite element analysis. *Communications in Numerical Methods in Engineering*, 24(9):777–784, 2008. ISSN 1099-0887. doi: 10.1002/cnm.989. URL http://dx.doi.org/10.1002/cnm.989.

O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method - The Basis*, volume 1. Butterworth-Heinemann, 5th edition, 2000.