

# Projekt Simulationstools und ihre Anwendung

Diskontinuierliche Galerkin Methode und  
ihre Anwendung in der Elastostatik

Tilo Laufer (328 449)  
Donat Rudenskiy(309 073)

27.9.2014

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Allgemeine Bilanzgleichung</b>	<b>4</b>
2.1	Bilanzgleichungen für reguläre Punkte . . . . .	9
2.1.1	Massenbilanz in regulären Punkten . . . . .	10
2.1.2	Impulsbilanz in regulären Punkten . . . . .	11
2.2	Bilanzgleichungen für Punkte auf singulären Flächen . . . . .	11
2.2.1	Massenbilanz in singulären Punkten . . . . .	11
2.2.2	Impulsbilanz in singulären Punkten . . . . .	12
2.3	Zusammenfassung - Allgemeine Bilanzgleichung . . . . .	13
<b>3</b>	<b>Materialmodell</b>	<b>13</b>
3.1	Linear elastisches isotropes Material . . . . .	13
3.2	Oberflächenspannungstensor . . . . .	14
3.3	Messung der isotropen Oberflächenspannung . . . . .	15
<b>4</b>	<b>Diskontinuierliche Galerkin Methode</b>	<b>16</b>
4.1	Ableitung der schwachen Form . . . . .	17
4.2	Numerische Flüsse . . . . .	19
4.3	Implementierung der schwachen Form . . . . .	20
4.4	Einbau von Rissen . . . . .	20
4.4.1	Rissinitialisierung . . . . .	21
4.4.2	Rissfortschritt . . . . .	21
<b>5</b>	<b>Ergebnisse und Auswertung</b>	<b>25</b>
5.1	Bestimmung vom <i>penalty factor</i> . . . . .	25
5.2	Aussehen eines Risses . . . . .	27
5.3	Ausblick . . . . .	28
<b>Anhang</b>		<b>29</b>
A.1	Gaussscher Satz mit singulären Flächen . . . . .	31
A.2	Transporttheorem für Volumenintegrale . . . . .	31
A.3	Transporttheorem für Flächenintegrale . . . . .	32
A.4	FEniCS Code zur Berechnung von Rissen . . . . .	34
<b>Literatur</b>		<b>41</b>

# 1 Einleitung

Ziel dieses Projektes ist die Prognose von Rissen in Festkörpern mit Hilfe der diskontinuierlichen Galerkin Methode. Risse führen zum Versagen von Bauteilen. Das Versagen eines belasteten Bolzens, aufgrund von Rissbildung, ist in Abbildung 1 dargestellt.



**Abbildung 1:** Spannungsriss unter Kerbwirkung an einem Bolzen [6]

Oft werden Risse in der Kontinuumsphysik mit Hilfe der *kontinuierlichen* finiten Elementmethode simuliert. Dabei werden semiempirische Ansätze in den Elementformulierungen verwendet, die die Rissbildung stark beeinflussen.

Bei der *diskontinuierlichen* Galerkin Methode werden Risse durch singuläre Flächen abgebildet. Die singulären Flächen erfordern die Verwendung von Bilanzgleichungen für Punkte auf singulären Flächen. Die Impuls- und Massenbilanzgleichungen für Punkte auf singulären Flächen wird zusammen mit den entsprechenden Bilanzgleichungen für Punkte in regulären Volumen ausführlich in dieser Arbeit hergeleitet.

Dafür wird in Kapitel 2 von der allgemeinen Bilanzgleichung ausgegangen. Ihre Form ist in der aktuellen (EULER) Konfiguration bekannt und kann aus MÜLLER [14, Kapitel 3.3] entnommen werden. Anschließend wird die Bilanzgleichung in die Referenzkonfiguration (LAGRANGE) überführt. Die Überführung ist sinnvoll, da die Referenzkonfiguration für Festkörper eine geeignetere Darstellung als die aktuelle Konfiguration ist. Nach der allgemeinen Herleitung der Gleichungen wird die Spezialisierung auf Impuls- und Massenbilanz durchgeführt.

Die Bilanzgleichungen werden in Kapitel 4 auf lineare finite Elemente spezialisiert, um daraus die schwache Formulierung abzuleiten. Bei der Herleitung werden numerische Flüsse über die Elementkanten hinweg eingeführt. Die Flüsse sind in der Literatur nicht einheitlich definiert. In dieser Arbeit wird eine Formulierung verwendet, die die Erfüllung der Impulsbilanz für Punkte auf singulären Flächen ermöglicht.

Zum Abschluss wird die hergeleitete Formulierung verwendet, um Rissbildung und Risswachstum zu simulieren. Dafür wird ein vereinfachtes Modell verwendet, das die Kerbwirkung im Bolzen aus Abbildung 1 abbildet.

## 2 Allgemeine Bilanzgleichung

Eine Bilanzgleichung beschreibt den zeitlichen und räumlichen Zustand einer zu bilanzierenden Größe:

$$\underbrace{\Psi^\bullet}_{\Psi: \text{bilanzierende Größe}} = \underbrace{M}_{\text{materieller Zufluss}} + \underbrace{F}_{\text{Zufluss}} + \underbrace{P}_{\text{Produktion} = 0} + \underbrace{S}_{\text{Zufuhr (Supply)}} . \quad (2.1)$$

Dabei wird in diesem Projekt angenommen, dass es zu keiner Produktion der zu bilanzierenden Größe kommt, wodurch chemische Reaktionen ausgeschlossen werden. Dies ist die Definition einer Erhaltungsgröße. Die Massen- und Impulsbilanz sind solche Erhaltungsgrößen. Durch die Beachtung von materiellen Zuflüssen, handelt es sich hier um eine Bilanzgleichung für offene Volumen.

Nun wird die Bilanzgleichung in integraler Form unter Verwendung der EINSTEINSchen Summenkonvention geschrieben:

$$\underbrace{\left( \int_{B^+ \cup B^-} B_\psi \, dv + \int_{^s a} {}^a \psi \, da \right)}_{\text{Änderung der zu bilanzierende Größe}} = \underbrace{\int_{a^+ \cup a^-} B_\psi (w_i - v_i) n_i \, da + \int_{\partial {}^s a} {}^a \psi (w_i - v_i) \nu_i \, dl}_{\text{materieller Zufluss}} - \underbrace{\int_{a^+ \cup a^-} {}^a \phi_k n_k \, da - \int_{\partial {}^s a} {}^l \phi_k \nu_k \, dl}_{\text{Zufluss}} + \underbrace{\int_{B^+ \cup B^-} B_s \, dv + \int_{^s a} {}^s \psi \, da}_{\text{Zufuhr (Supply)}} . \quad (2.2)$$

Erläuterung der verwendeten Variablen:

- $w_i$ : Geschwindigkeitsvektor der Abbildung
- $v_i$ : Geschwindigkeitsvektor eines materiellen Punktes
- ${}^a s$ : Dichte der Zufuhr auf der Singulärenfläche
- $B_s$ : Dichte der Zufuhr im Volumen
- ${}^l \phi_k$ : nichtmaterieller Flussdichtenvektor über den Rand der Singulärenfläche
- ${}^a \phi_k$ : nichtmaterieller Flussdichtenvektor über die Singulärenfläche

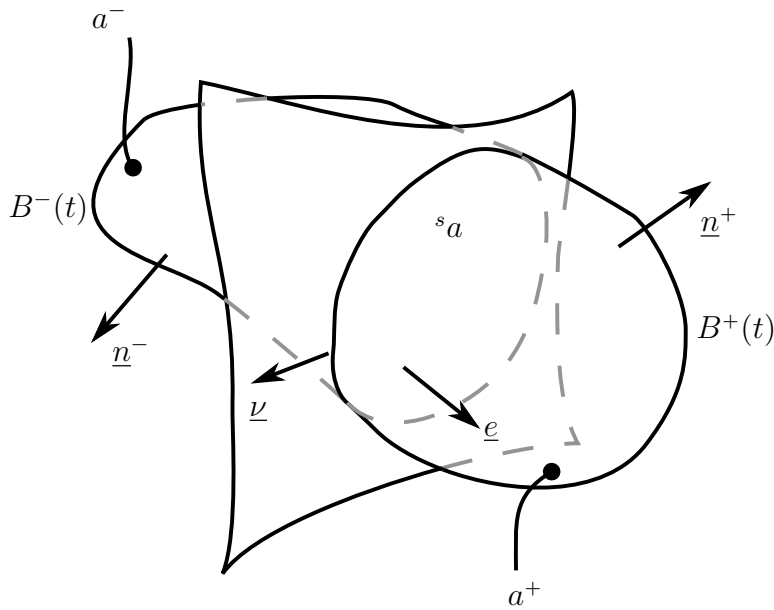
Weiterhin ist in der Gleichung (2.2)  $B_\psi$  die Dichte (pro Volumen) der zu bilanzierenden Größe  $\Psi$ . Die Größe  ${}^a \psi$  ist die Flächendichte der Größe  $\Psi$  und ist nur auf der singulären Fläche definiert. Durch die Beachtung der singulären Fläche müssen die Integralgrenzen genau unterschieden werden. Es wird idealisiert angenommen, dass sich ein solches Volumen mit singulärer Fläche als Kontinuumskartoffel approximieren lässt. Ein Volumen in der aktuellen Konfiguration ist in der Abbildung 2 dargestellt. Hier wird das Kontinuum  $B(t)$  durch eine singuläre Fläche  ${}^s a$  in zwei Untervolumen geteilt,  $B^+(t)$  und  $B^-(t)$ . Die Oberfläche der Teilvolumen setzt sich wie folgt zusammen:

$$\partial B^\pm(t) = a^\pm \cup {}^s a \cup \partial {}^s a . \quad (2.3)$$

Mit  $a^+ \cup a^-$  als Berandung von  $B^+ \cup B^-$ . Über die singuläre Fläche  ${}^s a$  zwischen  $B^+$  und  $B^-$  können Feldgrößen unstetig sein. So kann sich der linke Grenzwert  ${}^B \underline{\psi}^+$  vom rechten Grenzwert  ${}^B \underline{\psi}^-$  unterscheiden. Der Rand von  ${}^s a$  ist  $\partial {}^s a$  und hat die Dimension einer Linie. Weitere in der Abbildung 2 gezeigte Größen sind:

- $\underline{n}^\pm$  der auf der Fläche  $a^\pm$  definierte Normalenvektor,
- $\underline{e}$  der auf der Fläche  ${}^s a$  definierte Normalenvektor und
- $\underline{\nu}$  der auf dem Rand  $\partial {}^s a$  definierte Normalenvektor.

Wobei  $\underline{\nu}$  orthogonal zu der Linie  $\partial {}^s a$  und zu dem Normalenvektor  $\underline{e}$  ist.

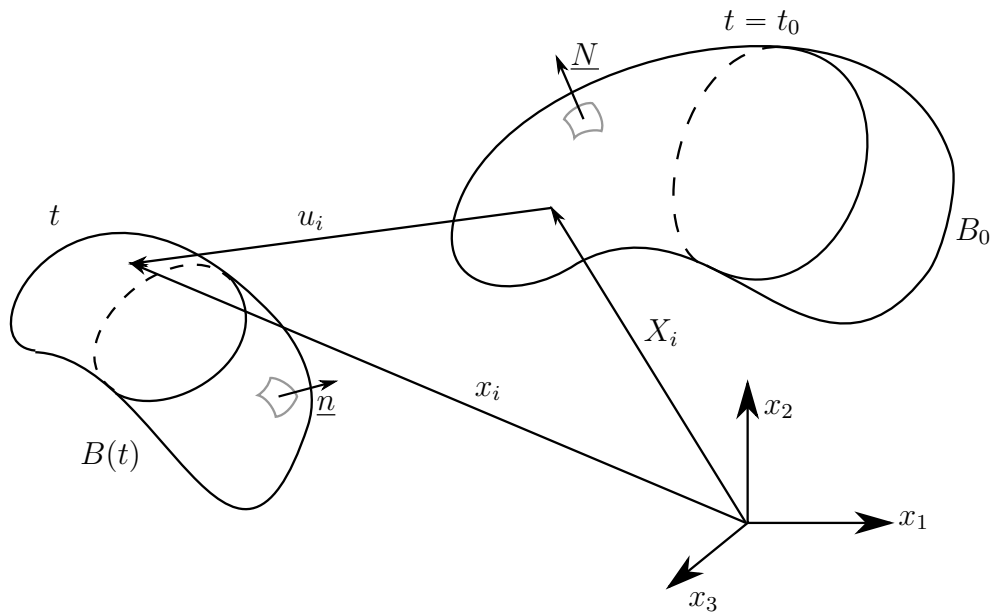


**Abbildung 2:** Kontinuumskartoffel mit singulärer Fläche in der aktuellen Konfiguration

Der Definitionsbereich der verschiedenen Größen ist nachfolgend aufgelistet:

- $\underline{n} = \underline{n}(\underline{x}, t), \forall \underline{x} \in (a^+ \cup a^-)$
  - $\underline{\nu} = \underline{\nu}(\underline{x}, t), \forall \underline{x} \in \partial {}^s a$
  - $\underline{e} = \underline{e}(\underline{x}, t), \forall \underline{x} \in {}^s a$
  - ${}^B \underline{\psi} = {}^B \underline{\psi}(\underline{x}, t), \forall \underline{x} \in (B^+ \cup B^-)$
  - ${}^a \underline{\psi} = {}^a \underline{\psi}(\underline{x}, t), \forall \underline{x} \in {}^s a$
  - ${}^{B_0} \underline{\psi} = {}^{B_0} \underline{\psi}(\underline{X}, t), \forall \underline{x} \in (B_0^+ \cup B_0^-)$
  - ${}^A \underline{\psi} = {}^A \underline{\psi}(\underline{X}, t), \forall \underline{x} \in {}^s A$
- } EULER
- } LAGRANGE

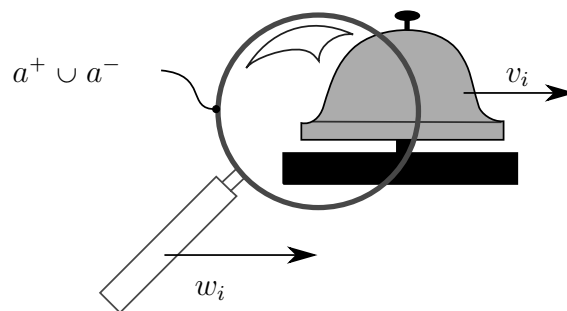
In der obigen Auflistung lässt sich gut erkennen, dass die Feldgröße  $\underline{\psi}$  in der aktuellen Konfiguration (EULER) von  $\underline{x}$  abhängt. In der Referenzkonfiguration (LAGRANGE) ist die Feldgröße  $\underline{\psi}$  abhängig von  $\underline{X}$ . Um beide Ortsvektoren zu unterscheiden, wird die Abbildung 2 (EULER) vereinfacht noch einmal in Abbildung 3 dargestellt.



**Abbildung 3:** Vereinfachte Kontinuumskartoffel in LAGRANGE und EULER Darstellung

Hier sieht man zweimal dasselbe Kontinuum. Die EULER-Darstellung ( $B(t)$ ) erkennt man daran, dass die Größen kleingeschrieben werden (Normalenvektor:  $\underline{n}$ ). Sie stellt den Körper in der aktuellen Konfiguration dar und alle Feldgrößen sind an dem aktuellen Ort  $\underline{x}$  definiert. Zusätzlich wird das gleiche Kontinuum ( $B_0$ ) in seiner Referenzlage (LAGRANGE) gezeichnet. Zu erkennen ist die LAGRANGE-Darstellung an großgeschriebenen Größen (Normalenvektor:  $\underline{N}$ ). Hier sind alle Feldgrößen an dem Initialort  $\underline{X}$  definiert.

Nun sollen die Geschwindigkeiten  $w_i$  und  $v_i$  aus der Gleichung (2.2) näher diskutiert werden. Betrachtet man Abbildung 4, so erkennt man gut den Unterschied zwischen materieller Geschwindigkeit  $v_i$  und Geschwindigkeit der Bilanzgrenzen  $w_i$  (Abbildungsgeschwindigkeit). Für den materiellen Fluss der Größe  ${}^B\psi$ , d.h. Materie, die die Bilanzgrenzen durchquert, ist nur die Relativgeschwindigkeit,  $w_i - v_i$ , von Interesse.



**Abbildung 4:** Unterscheidung von materieller Geschwindigkeit und Geschwindigkeit der Bilanzgrenzen

Die allgemeine Bilanzgleichung (2.2) soll nun von der EULER-Darstellung (aktuelle Konfiguration) in die LAGRANGE-Darstellung (Referenzkonfiguration) überführt werden. Dafür erinnern

wir uns zunächst an die integrale Form der Bilanzgleichung:

$$\begin{aligned} \left( \int_{B^+ \cup B^-} {}^B\psi \, dv + \int_{s_a} {}^a\psi \, da \right) \cdot &= \int_{a^+ \cup a^-} {}^B\psi(w_i - v_i)n_i \, da + \int_{\partial s_a} {}^a\psi(w_i - v_i)\nu_i \, dl - \\ &- \int_{a^+ \cup a^-} {}^a\phi_k n_k \, da - \int_{\partial s_a} {}^l\phi_k \nu_k \, dl + \int_{B^+ \cup B^-} {}^B s \, dv + \int_{s_a} {}^a s \, da. \end{aligned} \quad (2.4)$$

Zunächst wird die linke Seite behandelt. Dafür werden die Transporttheoreme im Anhang A.2 und A.3 auf Seite 31 ausführlich hergeleitet. Das Resultat ist nachfolgend noch einmal hingeschrieben:

$$\begin{aligned} \left( \int_{B^+ \cup B^-} {}^B\psi \, dv \right) \cdot &= \int_{B^+ \cup B^-} \left( {}^B\psi \cdot + {}^B\psi \frac{\partial w_i}{\partial x_i} \right) dv, \\ \left( \int_{s_a} {}^a\psi \, da \right) \cdot &= \int_{s_a} {}^a\psi \, da + \int_{s_a} {}^a\psi (w^\alpha_{;\alpha} - 2K_m w_\perp) da. \end{aligned} \quad (2.5)$$

Die Transporttheoreme (2.5) werden angewendet, um die zeitliche Ableitung über den Integralen auf der linken Seite umzuschreiben. Die Bilanzgleichung (2.4) ergibt sich damit zu:

$$\begin{aligned} &\int_{B^+ \cup B^-} \left( {}^B\psi \cdot + {}^B\psi \frac{\partial w_i}{\partial x_i} \right) dv + \int_{s_a} {}^a\psi + {}^a\psi (w^\alpha_{;\alpha} - 2K_m w_\perp) da = \\ &= \int_{a^+ \cup a^-} {}^B\psi(w_i - v_i)n_i \, da + \int_{\partial s_a} {}^a\psi(w_i - v_i)\nu_i \, dl - \\ &- \int_{a^+ \cup a^-} {}^a\phi_k n_k \, da - \int_{\partial s_a} {}^l\phi_k \nu_k \, dl + \int_{B^+ \cup B^-} {}^B s \, dv + \int_{s_a} {}^a s \, da. \end{aligned} \quad (2.6)$$

Zum Überführen der Bilanzgleichung von der EULER-Darstellung in die LAGERANGE-Darstellung müssen nun die integralen Grenzen umgeschrieben werden. Aus dem Anhang auf Seite 29 sind folgende Beziehungen zwischen EULER- und LAGERANGE-Darstellung bekannt:

$$\begin{aligned} dv &= J \, dV, \\ n_r \, da &= J(\mathbf{F}^{-1})_{pr} N_p \, dA, \\ dl_i &= F_{ik} \, dL \quad \text{und} \\ da &= \sqrt{\frac{g}{G}} \, dA. \end{aligned} \quad (2.7)$$

Damit können nun die Bilanzgleichungen in LAGERANGE-Darstellung geschrieben werden:

$$\begin{aligned}
& \int_{B_0^+ \cup B_0^-} \left( B_{0\psi} \cdot + B_{0\psi} \frac{\partial w_k}{\partial x_k} \right) J \, dV + \int_{sA} \left( {}^A\psi + {}^A\psi (w_{;\alpha}^\alpha - 2K_m w_\perp) \right) \sqrt{\frac{g}{G}} \, dA = \\
& = \int_{A^+ \cup A^-} \left( B_{0\psi}(w_k - v_k) - {}^A\phi_k \right) J(\mathbf{F}^{-1})_{pk} N_p \, dA + \int_{\partial sA} \left( {}^A\psi(w_k - v_k) - {}^L\phi_k \right) F_{ki} \nu_i^0 \, dL + \quad (2.8) \\
& + \int_{B_0^+ \cup B_0^-} B_{0s} J \, dV + \int_{sA} {}^A s \sqrt{\frac{g}{G}} \, dA.
\end{aligned}$$

Mit Hilfe des GAUSSSchen Satzes für Volumen (Gleichung (A.7)) und für Flächen (Gleichung (A.9)) kann die Bilanzgleichung weiter zusammengefasst werden:

$$\begin{aligned}
& \int_{B_0^+ \cup B_0^-} \left( B_{0\psi} \cdot J + B_{0\psi} J \frac{\partial w_k}{\partial x_k} - \frac{\partial}{\partial X_p} \left( B_{0\psi}(w_k - v_k) - {}^A\phi_k \right) J(\mathbf{F}^{-1})_{pk} - B_{0s} J \right) \, dV = \\
& = \int_{sA} \left( \left( - {}^A\psi \sqrt{\frac{g}{G}} - {}^A\psi (w_{;\alpha}^\alpha - 2K_m w_\perp) \sqrt{\frac{g}{G}} + \left( ({}^A\psi(w_k - v_k) - {}^L\phi_k) F_k^\Delta \right)_{;\Delta} + {}^A s \sqrt{\frac{g}{G}} \right) + \right. \\
& \left. + \left[ ({}^B_{0\psi}(w_k - v_k) - {}^A\phi_k) J(\mathbf{F}^{-1})_{pk} \right] E_p \right) \, dA. \quad (2.9)
\end{aligned}$$

Hierbei beschreibt  $[\cdot]$  die Sprungklammer, die wie folgt definiert ist:

$$[a] = (a^+ - a^-). \quad (2.10)$$

Anschließend wird nun noch der zweite Term in dem Volumenintegral, unter Verwendung der Hilfsformel:

$$\frac{\partial J(\mathbf{F}^{-1})_{sk}}{\partial X_s} = 0, \quad (2.11)$$

wie folgt umgeformt:

$$\begin{aligned}
B_{0\psi} J \frac{\partial w_i}{\partial x_i} &= B_{0\psi} J \frac{\partial w_i}{\partial X_k} \frac{\partial X_k}{\partial x_i} \\
&= B_{0\psi} J \frac{\partial w_i}{\partial X_k} (\mathbf{F}^{-1})_{ki} \\
&= B_{0\psi} \frac{\partial J w_i (\mathbf{F}^{-1})_{ki}}{\partial X_k}. \quad (2.12)
\end{aligned}$$



Somit ergibt sich die allgemeine Bilanzgleichung für offene Volumen zu:

$$\begin{aligned}
& \int_{B_0^+ \cup B_0^-} \left( {}^{B_0}\psi \cdot J + {}^{B_0}\psi \frac{\partial J w_i(\mathbf{F}^{-1})_{ki}}{\partial X_k} - \frac{\partial}{\partial X_p} ({}^{B_0}\psi(w_k - v_k) - {}^A\phi_k) J(\mathbf{F}^{-1})_{pk} - {}^{B_0}{}_S J \right) dV = \\
& = \int_{sA} \left( \left( - {}^A\psi \sqrt{\frac{g}{G}} - {}^A\psi (w^\alpha_{;\alpha} - 2K_m w_\perp) \sqrt{\frac{g}{G}} + \left( ({}^A\psi(w_k - v_k) - {}^L\phi_k) F_k^\Delta \right)_{;\Delta} + {}^A{}_S \sqrt{\frac{g}{G}} \right) + \right. \\
& \left. + [({}^{B_0}\psi(w_k - v_k) - {}^A\phi_k) J(\mathbf{F}^{-1})_{pk}] E_p \right) dA.
\end{aligned} \tag{2.13}$$

Wie gut in der Gleichung 2.13 zu erkennen ist, gibt es auf der linken Seite nur Integrale über reguläre Punkte. Auf der rechten Seite gibt es nur Integrale über singuläre Punkte. Beide Seiten müssen somit einzeln zu null werden. Damit ergibt sich die allgemeine Bilanzgleichung zu:

$$0 = \int_{B_0^+ \cup B_0^-} \left( {}^{B_0}\psi \cdot J + {}^{B_0}\psi \frac{\partial J w_i(\mathbf{F}^{-1})_{ki}}{\partial X_k} - \frac{\partial}{\partial X_p} ({}^{B_0}\psi(w_k - v_k) - {}^A\phi_k) J(\mathbf{F}^{-1})_{pk} - {}^{B_0}{}_S J \right) dV \tag{2.14}$$

für reguläre Punkte und zu:

$$\begin{aligned}
0 = \int_{sA} & \left( \left( - {}^A\psi \sqrt{\frac{g}{G}} - {}^A\psi (w^\alpha_{;\alpha} - 2K_m w_\perp) \sqrt{\frac{g}{G}} + \left( ({}^A\psi(w_k - v_k) - {}^L\phi_k) F_k^\Delta \right)_{;\Delta} + {}^A{}_S \sqrt{\frac{g}{G}} \right) + \right. \\
& \left. + [({}^{B_0}\psi(w_k - v_k) - {}^A\phi_k) J(\mathbf{F}^{-1})_{pk}] E_p \right) dA
\end{aligned} \tag{2.15}$$

für Punkte auf singulären Flächen.

## 2.1 Bilanzgleichungen für reguläre Punkte

In diesem Abschnitt wird nun die in Gleichung (2.14) ermittelte Bilanzgleichung für offene Volumen in regulären Punkten verwendet. Die Bilanzgleichungen müssen für Kontrollvolumen erfüllt werden. Auch bei immer kleiner werdenden Kontrollvolumen (infinitesimal kleine Volumen) verlieren sie ihre Gültigkeit nicht, sodass im folgenden die Integrale weggelassen werden können:

$${}^{B_0}\psi \cdot J + {}^{B_0}\psi \frac{\partial J w_i(\mathbf{F}^{-1})_{ki}}{\partial X_k} - \frac{\partial}{\partial X_p} ({}^{B_0}\psi(w_k - v_k) - {}^A\phi_k) J(\mathbf{F}^{-1})_{pk} - {}^{B_0}{}_S J = 0. \tag{2.16}$$

Für eine kürzere Schreibweise wird nun die Bezeichnung  $B_0$ , zur Kennzeichnung einer Volu-

mengröße, nicht weiter mitgeschrieben. Mit einer weiteren Umformung folgt:

$$\begin{aligned}
0 &= \psi \cdot J + \psi \frac{\partial J w_i (\mathbf{F}^{-1})_{ki}}{\partial X_k} - \frac{\partial}{\partial X_p} \left( \psi (w_k - v_k) - {}^A\phi_k \right) J (\mathbf{F}^{-1})_{pk} - sJ \\
&= \psi \cdot J + \underbrace{\psi \frac{\partial J w_i (\mathbf{F}^{-1})_{ki}}{\partial X_k} - \frac{\partial \psi w_k J (\mathbf{F}^{-1})_{pk}}{\partial X_p}}_{\frac{\partial \psi}{\partial X_p} w_k J (\mathbf{F}^{-1})_{pk}} + \frac{\partial \psi v_k J (\mathbf{F}^{-1})_{pk}}{\partial X_p} + \frac{\partial {}^A\phi_k J (\mathbf{F}^{-1})_{pk}}{\partial X_p} - sJ \\
&= \underbrace{\psi \cdot}_{\frac{\partial \psi}{\partial t} + \frac{\partial \psi}{\partial X_p} (\mathbf{F}^{-1})_{pi} w_i} J + \frac{\partial \psi}{\partial X_p} w_k J (\mathbf{F}^{-1})_{pk} + \frac{\partial \psi v_k J (\mathbf{F}^{-1})_{pk}}{\partial X_p} + \frac{\partial {}^A\phi_k J (\mathbf{F}^{-1})_{pk}}{\partial X_p} - sJ \quad (2.17) \\
&= \frac{\partial \psi}{\partial t} J + \psi \underbrace{\frac{\partial v_k}{\partial X_p} J (\mathbf{F}^{-1})_{pk}}_{\frac{dJ}{dt}} + v_k \underbrace{\frac{\partial \psi}{\partial X_p} J (\mathbf{F}^{-1})_{pk}}_{J \frac{\partial \psi}{\partial x_k}} + \frac{\partial {}^A\phi_k J (\mathbf{F}^{-1})_{pk}}{\partial X_p} - sJ \\
&= \frac{\partial \psi}{\partial t} J + \psi \frac{dJ}{dt} + v_k J \frac{\partial \psi}{\partial x_k} + \frac{\partial {}^A\phi_k J (\mathbf{F}^{-1})_{pk}}{\partial X_p} - sJ.
\end{aligned}$$

Daraus resultiert schlussendlich in der allgemeinen Bilanzgleichung für Punkte in regulären Volumen:

$$0 = \frac{d\psi J}{dt} + \frac{\partial {}^A\phi_k J (\mathbf{F}^{-1})_{pk}}{\partial X_p} - sJ. \quad (2.18)$$

Diese Gleichung wird nun nachfolgend benutzt um die Massen- und die Impulsbilanz für reguläre Punkte herzuleiten.

### 2.1.1 Massenbilanz in regulären Punkten

Mit Hilfe der Tabelle 1, in der die zu bilanzierende Größe konkretisiert wird, kann nun die in Gleichung (2.18) gefundene allgemeine Bilanz auf die Massenbilanz spezialisiert werden:

$$\begin{aligned}
0 &= \frac{d\rho J}{dt} \\
\Rightarrow \rho J &= \rho_0 = \text{const.}|_t.
\end{aligned} \quad (2.19)$$

Dies ist die Lösung der Massenbilanz für reguläre Punkte.

**Tabelle 1:** Übersicht der verschiedenen Feldgrößen in unterschiedlichen Bilanzgleichungen

	$B_\psi$	$a_\psi$	$a_\phi_k$	$l_\phi_k$	$B_s$	$a_s$
Massenbilanz	$B_\rho$	$a_\rho$	0	0	0	0
Impulsbilanz	$B_\rho B_{v_i}$	$a_\rho a_{v_i}$	$l_{\sigma_{ki}}$	$l_{\sigma_{ki}}$	$B_\rho f_i$	$a_\rho f_i$

### 2.1.2 Impulsbilanz in regulären Punkten

In analoger Weise lässt sich auch die Impulsbilanz aus der Gleichung (2.18) mit Tabelle 1 für reguläre Punkte finden:

$$0 = \frac{d\rho v_i J}{dt} + \frac{\partial}{\partial X_k} \left( \underbrace{\sigma_{ji} J(\mathbf{F}^{-1})_{kj}}_{P_{ki}} \right) - \rho f_i J. \quad (2.20)$$

Wobei hier der 1. PIOLA-KIRCHHOFF-Spannungstensor  $P_{ki} = \sigma_{ji} J(\mathbf{F}^{-1})_{kj}$  eingeführt wurde. Nun wird die Lösung der Massenbilanz ( $\rho J = \rho_0$ ) in die Impulsbilanz eingesetzt, sodass die Massenbilanz implizit erfüllt wird:

$$\begin{aligned} 0 &= \frac{d}{dt} (\rho_0 v_i) + \frac{\partial P_{ki}}{\partial X_k} - \rho_0 f_i \quad \left| \quad \text{mit: } v_i = \frac{d}{dt} (u_i + X_i) = \frac{d}{dt} u_i \right. \\ &= \rho_0 \frac{d^2 u_i}{dt^2} + \frac{\partial P_{ki}}{\partial X_k} - \rho_0 f_i. \end{aligned} \quad (2.21)$$

Dies ist die Lösung der Impulsbilanz. Sie erfüllt die Massenbilanz und ist für viele Arbeiten in der Kontinuumsphysik der Startpunkt.

## 2.2 Bilanzgleichungen für Punkte auf singulären Flächen

In diesem Abschnitt wird nun die in Gleichung (2.15) ermittelte Bilanzgleichung für Punkte auf singulären Flächen verwendet. Auch hier werden die Integrale in der Bilanzgleichung fallen gelassen, was sich durch das Pillendosenargument begründen lässt, siehe MÜLLER [14, S. 68]. Damit ergibt sich die allgemeine Bilanzgleichung für Punkte auf singulären Flächen zu:

$$\begin{aligned} 0 &= - {}^A\psi \sqrt{\frac{g}{G}} - {}^A\psi (w_{;\alpha}^\alpha - 2K_m w_\perp) \sqrt{\frac{g}{G}} + \left( ({}^A\psi (w_k - v_k) - {}^L\phi_k) F_k^\Delta \right)_{;\Delta} + {}^A s \sqrt{\frac{g}{G}} + \\ &+ \left[ ({}^{B_0}\psi (w_k - v_k) - {}^A\phi_k) J(\mathbf{F}^{-1})_{pk} \right] E_p. \end{aligned} \quad (2.22)$$

Mit der Definition der eckigen Klammer aus Gleichung (2.10).

### 2.2.1 Massenbilanz in singulären Punkten

Auch hier wird mit Hilfe der Tabelle 1 eine Spezialisierung zur Massenbilanz durchgeführt. Die Gleichung (2.22) wird somit zu:

$$\begin{aligned} - {}^A\rho \sqrt{\frac{g}{G}} - {}^A\rho (w_{;\alpha}^\alpha - 2K_m w_\perp) \sqrt{\frac{g}{G}} + \left( {}^A\rho (w_k - v_k) F_k^\Delta \right)_{;\Delta} + \\ + \left[ \rho (w_k - v_k) J(\mathbf{F}^{-1})_{pk} \right] E_p = 0. \end{aligned} \quad (2.23)$$

Die erste Annahme ist, dass die singuläre Fläche kein Eigenleben besitzt. Was damit zu begründen ist, dass es sich bei der singulären Fläche um eine erdachte Fläche handelt. Damit

ergibt sich die Dichte zu  ${}^A\rho = 0$  und die Bilanzgleichung vereinfacht sich mit der Benutzung der Massenbilanz aus der Gleichung (2.19) wie folgt:

$$\left[ \underbrace{\rho J}_{\rho_0} (w_k - v_k) (\mathbf{F}^{-1})_{pk} \right] E_p = 0. \quad (2.24)$$

Das ist die Lösung der Massenbilanz für Punkte auf singulären Flächen.

### 2.2.2 Impulsbilanz in singulären Punkten

Die Spezialisierung der allgemeinen Bilanzgleichung für Punkte auf singulären Flächen (2.22) ergibt für die Impulsbilanz:

$$\begin{aligned} & - ({}^A\rho w_i) \cdot \sqrt{\frac{g}{G}} - {}^A\rho w_i (w^\alpha_{;\alpha} - 2K_m w_\perp) \sqrt{\frac{g}{G}} \\ & + \left( ({}^A\rho w_i (w_k - v_k) - {}^l\sigma_{ki}) F_k^\Delta \right)_{;\Delta} + {}^A\rho f_i \sqrt{\frac{g}{G}} + \\ & + \left[ (\rho v_i (w_k - v_k) - \sigma_{ki}) J(\mathbf{F}^{-1})_{pk} \right] E_p = 0. \end{aligned} \quad (2.25)$$

Und ohne Eigenleben  ${}^A\rho = 0$  der singulären Fläche folgt:

$$({}^l\sigma_{ki} F_k^\Delta)_{;\Delta} - \left[ (\rho v_i (w_k - v_k) - \sigma_{ki}) J(\mathbf{F}^{-1})_{pk} \right] E_p = 0. \quad (2.26)$$

Hier beschreibt  ${}^l\sigma_{ki}$  den Oberflächenspannungstensor (*surface stress*). Eine genauere Definition dazu erfolgt im Kapitel 3.2 auf der Seite 14. Wie bei der regulären Impulsbilanz, ist es auch bei der Impulsbilanz für Punkte auf singulären Flächen möglich, die entsprechende Massenbilanz implizit zu erfüllen. Durch Einsetzen ergibt sich:

$$\begin{aligned} 0 & = ({}^l\sigma_{ki} F_k^\Delta)_{;\Delta} - \left[ (\rho v_i (w_k - v_k) - \sigma_{ki}) J(\mathbf{F}^{-1})_{pk} \right] E_p \\ & = ({}^l\sigma_{ki} F_k^\Delta)_{;\Delta} - \left[ \underbrace{v_i \rho_0 (w_k - v_k) (\mathbf{F}^{-1})_{pk} - \sigma_{ki} J(\mathbf{F}^{-1})_{pk}}_{\text{Massenbilanz (2.24) = 0}} \right] E_p = 0 \\ & = ({}^l\sigma_{ki} F_k^\Delta)_{;\Delta} - \left[ \underbrace{J \sigma_{ki} (\mathbf{F}^{-1})_{pk}}_{P_{pi}} \right] E_p \\ & = ({}^l\sigma_{ki} F_k^\Delta)_{;\Delta} - [P_{pi}] E_p. \end{aligned} \quad (2.27)$$

Diese Differentialgleichung ist eine Sonderform der Impulsbilanz für Punkte auf singulären Flächen. Eine Lösung dieser Differentialgleichung erfüllt die Impulsbilanz sowie die Massenbilanz (Gleichung (2.24)) für Punkte auf Singulärenflächen.

## 2.3 Zusammenfassung - Allgemeine Bilanzgleichung

Zusammenfassend ergibt sich damit der folgende Satz von Bilanzgleichungen:

$$\begin{aligned} 0 &= \rho_0 \frac{d^2 u_i}{dt^2} + \frac{\partial P_{ki}}{\partial X_k} - \rho_0 f_i, \\ 0 &= ({}^l\sigma_{ki} F_k^\Delta)_{;\Delta} - [P_{pi}] E_p, \end{aligned} \quad (2.28)$$

mit welchen das Verschiebungsfeld,  $u_i$ , eines beliebigen Randanfangswertproblems gelöst werden kann, falls  $P_{ij}$  und  ${}^l\sigma_{ij}$  in Bezug auf  $u_i$  definiert werden. Für die nun noch fehlenden Größen,  $P_{ij}$  und  ${}^l\sigma_{ij}$ , werden Ausdrücke im dem folgenden Kapitel gefunden.

## 3 Materialmodell

Um die Bilanzgleichungen abschließen zu können, wird ein Materialmodell benötigt. In dieser Arbeit wird ein elastisches Materialmodell inklusive Oberflächenspannung für Titan angewendet.

### 3.1 Linear elastisches isotropes Material

Das Verhalten von Titan wird als linear elastisch angenommen. Der GREEN-LAGRANGE-Dehnungstensor wird nun als Dehnungsmaß eingeführt:

$$E_{ij} = \frac{1}{2} (C_{ij} - \delta_{ij}) . \quad (3.1)$$

$C_{ij}$  bezeichnet dabei den CAUCHY-GREEN-Tensor und ist definiert als:

$$C_{ij} = F_{ki} F_{kj} , \quad (3.2)$$

mit  $F_{ki}$  als Deformationsgradient. Lässt sich nun der Spannungstensor  $P_{ki}$  durch ein Materialmodell in Abhängigkeit des Dehnungstensor  $E_{ij}$  ausdrücken, so ist die Gleichung (2.28)<sub>1</sub> abgeschlossen. Somit kann das Verschiebungsfeld für ein reguläres Kontinuum berechnet werden.

Soll nun der Spannungstensor mit dem Dehnungstensor verknüpft werden, so ist die Tatsache problematisch, dass der 1. PIOLA-KIRCHHOFF-Spannungstensor nicht symmetrisch ist. Zur Lösung wird der Spannungstensor symmetrisiert. Dadurch erhalten wir den 2. PIOLA-KIRCHHOFF-Spannungstensor  $S_{kl}$ :

$$\begin{aligned} S_{kl} &= (\mathbf{F}^{-1})_{li} \sigma_{ji} J (\mathbf{F}^{-1})_{kj} = (\mathbf{F}^{-1})_{li} P_{ki} = \text{symm.} \\ F_{nl} S_{kl} &= \underbrace{F_{nl} (\mathbf{F}^{-1})_{li}}_{\delta_{ni}} P_{ki} = P_{kn} . \end{aligned} \quad (3.3)$$

Für isotropes Material lässt sich nun mit Hilfe des HOOKEschen Gesetzes der Dehnungstensor mit dem Spannungstensor verknüpfen:

$$S_{ij} = \lambda E_{kk} \delta_{ij} + 2\mu E_{ij} . \quad (3.4)$$

Dabei sind  $\lambda$  und  $\mu$  die Lamé-Konstanten und lassen sich wie folgt durch Ingenieurkonstanten berechnen:

- $\lambda = \frac{2G\nu}{1-2\nu}$ ,
- $\mu = G$ .

Die Ingenieurkonstanten für Titan sind:

- $E = 170 \text{ GPa}$ ,
- $\nu = 0,39$ ,
- $G = \frac{E}{2(1+\nu)}$ .

Damit ist die Bilanzgleichung (2.28)<sub>1</sub> für reguläre Punkte komplett geschlossen. Zur Schließung der Bilanzgleichung (2.28)<sub>2</sub> für Punkte auf singulären Flächen fehlt noch eine Beschreibung für den Oberflächenspannungstensor  ${}^l\sigma_{ki}$ .

### 3.2 Oberflächenspannungstensor

Der Oberflächenspannungstensor wird bei der Herleitung der Impulsbilanz für Punkte auf singulären Flächen eingeführt. Die recht unbekannt GröÙe ist jedoch schon in verschiedenen Veröffentlichungen untersucht worden [10, 11]. Siehe [4] für eine ausführliche Einführung, sowie numerische und experimentelle Methoden zur Messung der Oberflächenspannung.

Der Oberflächenspannungstensor ist wie folgt definiert:

$${}^l\sigma_{ki} = \gamma\delta_{ki} + \frac{\partial\gamma}{\partial\varepsilon_{ki}}. \quad (3.5)$$

Da in der englischen Sprache eine bessere Differenzierung zwischen den GröÙen  ${}^l\sigma_{ki}$  und  $\gamma$  existiert, wird in diesem Abschnitt darauf zurückgegriffen.

Nach VERMAAK [10] gibt es eine weit verbreitete Fehlinterpretation und Verwirrung zwischen den GröÙen *surface tension* und *surface stress*. Folgt man den Erläuterungen von VERMAAK, so ist die zentrale thermodynamische GröÙe die eine Oberfläche beschreibt  $\gamma$  und wird *surface tension* genannt. Die *surface tension* ist die Arbeit, die benötigt wird, um eine Einheitsfläche bei definierter Umgebung (Temperatur, chemisches Potential und Volumen) zu erzeugen.

Der *surface stress* ist  ${}^l\sigma_{ki}$  und beschreibt die Arbeit, die benötigt wird, um eine existierende Fläche um das Maß der Einheitsfläche zu vergrößern. Die Indizes  $k$  und  $i$  stehen dabei für:

- $k$ : Normalenvektor  $n_k$  der zu vergrößerten Oberfläche,
- $i$ : Richtung, in der die Kraft wirkt.

VERMAAK sagt in [10, Kapitel 3.] das der Term  $\frac{\partial\gamma}{\partial\varepsilon_{ki}}$  für Flüssigkeiten zu null wird. Dadurch werden *surface tension* und *surface stress* für Flüssigkeiten vom Zahlenwert her identisch. Dies gilt jedoch nicht für Festkörper. Für diese Arbeit wird die folgende Bezeichnung der relevanten GröÙen verwendet:

- ${}^l\sigma_{ki}$ : die Oberflächenspannungstensor [N/m],
- $\gamma$ : die Oberflächenspannung [N/m] und
- $\varepsilon_{ki}$ : der Dehnungstensor [-].

Für isotropes Verhalten auf der Oberfläche, wie z. B. bei Titan, lässt sich die Gleichung (3.5) wie folgt vereinfachen:

$${}^l\sigma_{ki} = f\delta_{ki}. \quad (3.6)$$

Die isotrope Oberflächenspannung  $f$  für Titan wurde von CAMMARATA [4, S. 15] mit  $f = 2,574 \text{ J/m}^2$  angegeben. Die Messung des Parameters ist allgemein nicht bekannt, darum wird ihr Vorgehen im nächsten Abschnitt genauer beschrieben.

### 3.3 Messung der isotropen Oberflächenspannung

Dieser Abschnitt ist in Anlehnung an CAMMARATA [4, chap. 2.F.] geschrieben. Die Oberflächenspannung lässt sich auf zwei grundsätzlich verschiedene Weisen bestimmen. Die eine Methode bestimmt die Oberflächenspannung mittels molekulardynamischer Simulation und die andere Methode ist eine experimentelle Bestimmung.

Für eine experimentelle Bestimmung wird meist die elastische Dehnung gemessen, die aus der aufgebrachten Spannung resultiert.

Laut CAMMARATA [4] wird nach Messung der Dehnung der Wert der Spannung mit Hilfe des HOOKEschen-Gesetz ermittelt. Das HOOKEschen-Gesetz stimmt in den meisten Fällen auch für die Messung der Oberflächenspannung.

Für die Messung der Oberflächenspannung wird meist eine Vollkugel betrachtet. Die Kugel hat den Radius  $r$  und wird von einer isotropen Oberflächenspannung  ${}^l\sigma_{ij} = f\delta_{ij}$  belastet. Nun wird angenommen das sich die isotropen Oberflächenspannung wie ein hydrostatischer Druck verhält. Damit ist der LAPLACE-Druck für diese Kugel bekannt als:

$$\Delta P = \frac{2f}{r}. \quad (3.7)$$

Die daraus resultierende Dehnung in der Kugel lässt sich nun mit einem angepassten HOOKEschen-Gesetz mit dem Druck  $\Delta P$  verknüpfen:

$$-\Delta P = K\varepsilon_V = K\frac{\Delta V}{V}. \quad (3.8)$$

Hierbei ist  $K = \frac{E}{3(1-2\nu)}$  der Kompressionsmodul und  $\varepsilon_V = \frac{\Delta V}{V}$  die Volumendilatation. Die Oberflächenspannung lässt sich damit wie folgt berechnen:

$$f = -\frac{r}{2}K\frac{\Delta V}{V}. \quad (3.9)$$

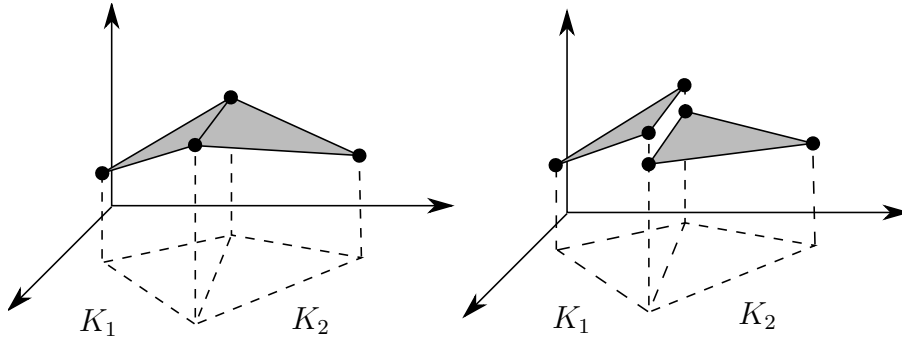
Die Änderung des Volumens  $\Delta V$  erfolgt dabei durch Messung der Gitterkonstante. Hierbei wird davon ausgegangen, dass sich die Atome in einer idealen Gitterstruktur befinden, wenn die Kugel unbelastet ist. Der Abstand zwischen den Atomen wird mit der Gitterkonstante  $a$  bezeichnet. Durch die Oberflächenspannung sind die Atome nun nicht mehr in ihrer idealen Gitterstruktur angeordnet. Die Abweichung der Gitterkonstante wird als  $\Delta a$  bezeichnet. Nach MAYS et. al. [8] berechnet sich nun die Oberflächenspannung wie folgt:

$$f = -\frac{3}{2}Kr\frac{\Delta a}{a}. \quad (3.10)$$

Die Messung bei MAYS [8] erfolgt an sehr kleinen Kugeln mit Durchmessern von 35Å bis 125Å. Ein Ångström, Å, entspricht:  $1\text{Å} = 10^{-10}\text{ m}$ . Für Interessierte sei die weiterführende Literatur von STONEHAM [15] genannt.

## 4 Diskontinuierliche Galerkin Methode

Nachdem die partiellen Differentialgleichungen (2.21) und (4.5) hergeleitet sind, sollen die Gleichungen mit Hilfe der finiten Elementmethode (FEM) gelöst werden. In der kontinuierlichen FEM werden  $C^0$ -stetige Elemente verwendet. Hier soll auf die  $C^0$ -Stetigkeit verzichtet werden. Der Unterschied ist exemplarisch in Abbildung 5 dargestellt. Die daraus resultierenden Sprünge sollen durch die Bilanzgleichung für Punkte auf singulären Flächen beschränkt werden.



**Abbildung 5:** kontinuierliche Galerkin Methode neben diskontinuierlicher Galerkin Methode

Nun wird noch einmal an die Bilanzgleichungen (2.28) aus Kapitel 3 erinnert:

$$\begin{aligned} 0 &= \rho_0 \frac{d^2 u_i}{dt^2} + \frac{\partial P_{ki}}{\partial X_k} - \rho_0 f_i \\ 0 &= ({}^l \sigma_{ki} F_k^\Delta)_{;\Delta} - [P_{pi}] E_p. \end{aligned} \quad (4.1)$$

Betrachtet man nun Abbildung 5, so erkennt man, dass die finiten Elemente  $K_i$  gradlinig berandet sind. Dies liegt daran, dass in dieser Arbeit nur Elemente mit linearem Polynomansatz verwendet werden. In der Impulsbilanz für Punkte auf singulären Flächen ist eine kovariante Ableitung  $(\bullet)_{;\Delta}$ , die für lineare Elemente (Elemente mit linearem Polynomansatz) zu einer partiellen Ableitung  $(\bullet)_{,\Delta}$  wird. Zusätzlich muss für lineare Elemente nicht zwischen ko- und kontravarianter Schreibweise unterschieden werden. Mit diesen Informationen lässt sich die Bilanzgleichung für Punkte auf singulären Flächen wie folgt vereinfachen:

$$\begin{aligned} 0 &= ({}^l \sigma_{ki} F_{k\Delta})_{;\Delta} - [P_{pi}] E_p \text{ mit: } F_{k\Delta} = G_{\alpha\Delta} F_{kl} T_{\alpha l} \\ &= ({}^l \sigma_{ki} G_{\alpha\Delta} F_{kl} T_{\alpha l})_{,\Delta} - [P_{pi}] E_p \\ &= ({}^l \sigma_{ki} \delta_{\alpha\Delta} F_{kl} T_{\alpha l})_{,\Delta} - [P_{pi}] E_p \\ &= ({}^l \sigma_{ki} F_{kl})_{,\Delta} T_{\Delta l} - [P_{pi}] E_p \\ &= \frac{\partial}{\partial \xi^\Delta} ({}^l \sigma_{ki} F_{kl}) T_{\Delta l} - [P_{pi}] E_p. \end{aligned} \quad (4.2)$$

Es ist das Ziel nur Ableitungen nach  $X_i$  in den Gleichungen zu verwenden. Das ist in der Gleichung (4.2)<sub>5</sub> nicht der Fall. Um die Ableitung nach  $\xi^\Delta$  zu ersetzen, wird die Gleichung (4.2)<sub>5</sub>



wie folgt umgeschrieben:

$$\begin{aligned}
0 &= \frac{\partial}{\partial \xi^\Delta} ({}^l\sigma_{ki} F_{kl}) T_{\Delta l} - [P_{pi}] E_p \\
&= ({}^l\sigma_{ki} F_{kl})_{,m} \underbrace{\frac{\partial X_m}{\partial \xi^\Delta}}_{T_{\Delta u}} T_{\Delta l} - [P_{pi}] E_p \\
&= ({}^l\sigma_{ki} F_{kl})_{,m} \underbrace{T_{\Delta m} T_{\Delta l}}_{G_{lm}} - [P_{pi}] E_p,
\end{aligned} \tag{4.3}$$

wobei die Oberfläche in einer kartesischen Basis definiert ist, deshalb gilt  $G_{lm} = \delta_{lm}$  und die Gleichung (4.3)<sub>3</sub> wird zu:

$$0 = ({}^l\sigma_{ki} F_{kl})_{,m} \delta_{lm} - [P_{pi}] E_p. \tag{4.4}$$

Die Impulsbilanz für Punkte auf singulären Flächen vereinfacht sich somit zu:

$$0 = ({}^l\sigma_{ki} F_{kl})_{,l} - [P_{pi}] E_p. \tag{4.5}$$

Damit die Bilanzgleichungen nun mit Hilfe der finiten Elementmethode gelöst werden kann, wird zunächst die schwache Form der Differentialgleichung (2.28)<sub>1</sub> benötigt. Diese wird im nachfolgenden Kapitel hergeleitet.

## 4.1 Ableitung der schwachen Form

Im Rahmen dieser Arbeit werden statische Probleme untersucht. Dafür wird die Gleichung (2.28)<sub>1</sub> auf stationäre Probleme spezialisiert:

$$\begin{aligned}
0 &= \underbrace{\rho_0 \frac{d^2 u_i}{dt^2}}_{=0} + \frac{\partial P_{ki}}{\partial X_k} - \rho_0 f_i \\
&= \frac{\partial P_{ki}}{\partial X_k} - \rho_0 f_i.
\end{aligned} \tag{4.6}$$

Die schwache Form ist bekannt für eine Differentialgleichung (DGL) erster Ordnung. Deshalb wird die DGL zweiter Ordnung in Gleichung (4.6) zunächst in ein System von Differentialgleichungen erster Ordnung überführt. Dafür wird eine neue Variable  $K_{ij} = u_{i,j}$  eingeführt:

$$\begin{aligned}
P_{ki,k} &= f_i, \\
K_{ij} &= u_{i,j},
\end{aligned} \tag{4.7}$$

mit  $P_{ki} = P_{ki}(\mathbf{K})$ , wodurch  $P_{ki}$  als eine Funktion von  $\mathbf{K}$  aufgefasst wird. Die bekannten schwachen Formen zu den Gleichungen (4.7)<sub>1</sub> und (4.7)<sub>2</sub> lauten:

$$\begin{aligned}
\int_K (P_{ki} \delta u_{i,k} - f_i \delta u_i) dx - \int_{\partial K} \hat{P}_{ki} \delta u_i n_k ds &= 0 \text{ und} \\
\int_K (K_{ij} \delta K_{ij} + u_i \delta K_{ij,j}) dx - \int_{\partial K} \hat{u}_i \delta K_{ij} n_j ds &= 0.
\end{aligned} \tag{4.8}$$

Das Gebiet  $K$ , über das integriert wird, ist ein einziges finites Element ohne Singularitäten. In der schwachen Form wurden die numerischen Flüsse  $\hat{P}_{ki}$  und  $\hat{u}_i$  eingeführt, die im Abschnitt 4.2

ab Seite 19 näher erläutert werden. Durch das Einsetzen von  $\delta K_{ij} = \delta u_{i,j}$  lassen sich nun beide schwachen Formen addieren:

$$\int_K \left( u_{i,j} \delta u_{i,j} + u_i \delta u_{i,jj} + P_{ki} \delta u_{i,k} - f_i \delta u_i \right) dx - \int_{\partial K} \left( \hat{u}_i \delta u_{i,j} n_j + \hat{P}_{ki} \delta u_i n_k \right) ds = 0. \quad (4.9)$$

Durch das Rücksubstituieren von  $\mathbf{K}$  in der Gleichung (4.9) ist  $P_{ki}$  wieder eine Funktion von  $u_i$ . Mit Hilfe der partiellen Integration von  $\delta u_{i,jj}$  lässt sich die Gleichung wie folgt vereinfachen:

$$\begin{aligned} 0 &= \int_K \left( u_{i,j} \delta u_{i,j} + \underbrace{u_i \delta u_{i,jj}}_{-\int_K u_{i,j} \delta u_{i,j} dx + \int_{\partial K} u_i \delta u_{i,j} n_j ds} + P_{ki} \delta u_{i,k} - f_i \delta u_i \right) dx - \\ &\quad - \int_{\partial K} \left( \hat{P}_{ji} \delta u_i + \hat{u}_i \delta u_{i,j} \right) n_j ds \\ &= \int_K \left( P_{ki} \delta u_{i,k} - f_i \delta u_i \right) dx + \int_{\partial K} \left( (u_i - \hat{u}_i) \delta u_{i,j} - \hat{P}_{ji} \delta u_i \right) n_j ds = 0. \end{aligned} \quad (4.10)$$

Als nächstes werden die Integrale über alle finiten Elemente aufsummiert. Dadurch ergibt sich:

$$0 = \sum_{k=1}^n \int_{K^{(k)}} \left( P_{ki} \delta u_{i,k} - f_i \delta u_i \right) dx + \int_{\partial K^{(k)}} \left( (u_i - \hat{u}_i) \delta u_{i,j} - \hat{P}_{ji} \delta u_i \right) n_j ds = 0. \quad (4.11)$$

Durch Auswerten der Summe ergeben sich neue Gebiete über die integriert wird. Diese sind in Abbildung 6 dargestellt. Dabei beschreibt  $\Omega$  das gesamte Gebiet,  $\Gamma$  die Ränder zwischen den Elementen (im Inneren von  $\Omega$ ) und  $\partial\Omega$  die äußeren Ränder vom Gebiet  $\Omega$ , die nur ein angrenzendes Element besitzen. Unter Berücksichtigung der neuen Integrationsgebiete ergibt sich Gleichung (4.11) zu:

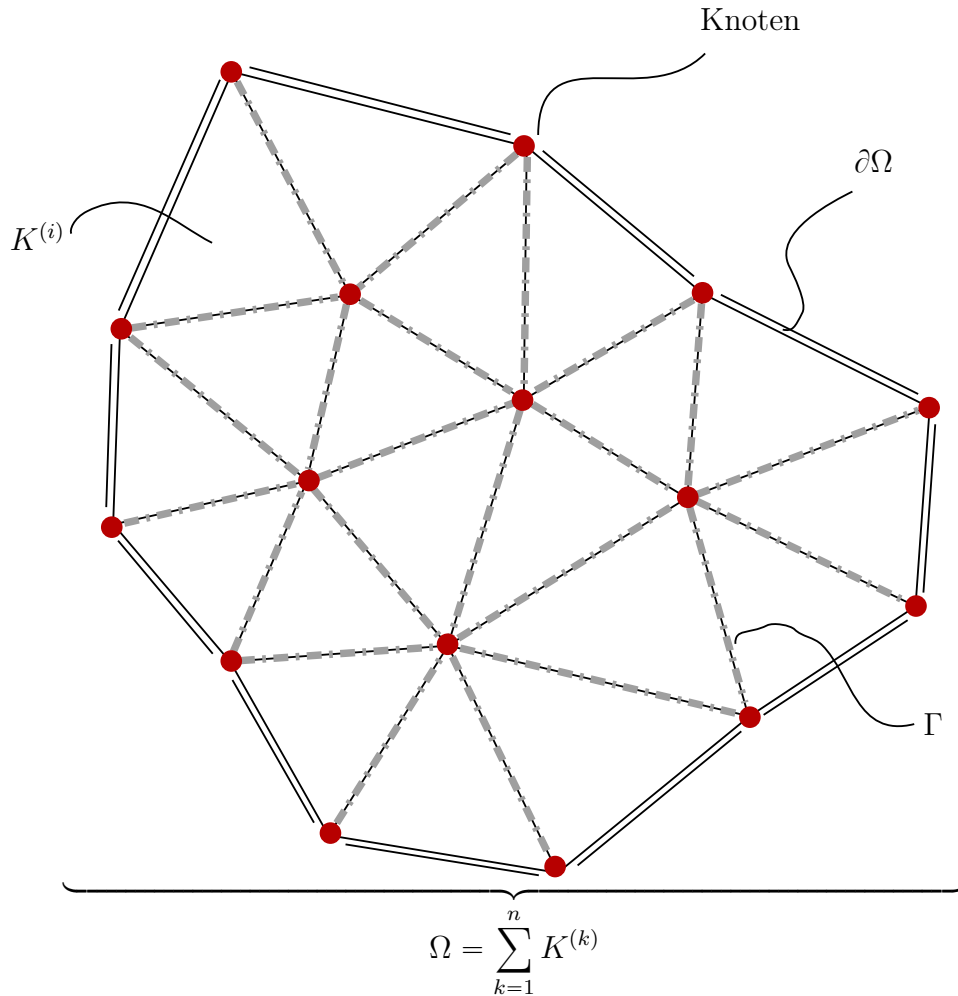
$$\begin{aligned} 0 &= \int_{\Omega} \left( P_{ji} \delta u_{i,j} - f_i \delta u_i \right) dx + \int_{\Gamma} \left( \{u_i - \hat{u}_i\} \llbracket \delta u_{i,j} \rrbracket + \llbracket u_i - \hat{u}_i \rrbracket \{ \delta u_{i,j} \} - \right. \\ &\quad \left. - \{ \delta u_i \} \llbracket \hat{P}_{ji} \rrbracket - \{ \hat{P}_{ji} \} \llbracket \delta u_i \rrbracket \right) dS + \int_{\partial\Omega} \left( (u_i - \hat{u}_i) \delta u_{i,j} - \hat{P}_{ji} \delta u_i \right) n_j ds. \end{aligned} \quad (4.12)$$

Wobei hier eine neue Sprungklammer,  $\llbracket \cdot \rrbracket$ , eingeführt wurde. Sie unterscheidet zwischen Tensoren erster Stufe und Tensoren zweiter Stufe und ist wie folgt definiert:

$$\begin{aligned} \llbracket A_i \rrbracket &= (A_i^+ n_j^+ + A_i^- n_j^-) \\ \llbracket B_{ij} \rrbracket &= (B_{ij}^+ n_j^+ + B_{ij}^- n_j^-) = (B_{ij}^+ - B_{ij}^-) n_j^+ = -[B_{ij}] E_j. \end{aligned} \quad (4.13)$$

Außerdem wurde auch die Mittelwertsklammer  $\{ \cdot \}$  eingeführt, die folgendermaßen definiert ist:

$$\{ a \} = \frac{1}{2} (a^+ + a^-). \quad (4.14)$$



**Abbildung 6:** Diskretisierung der Kontinuumskartoffel

## 4.2 Numerische Flüsse

Die numerischen Flüsse sind nicht einheitlich implementiert, siehe [5, 9, 7, 13, 16]. Es gibt bis zu sechs verschiedene Stabilisierungsparameter. Allerdings sagt KAUFMANN in [12] das BABUSKA und ZLAMAL nur einen *penalty-factor* verwenden,  $\eta$ , welcher in vielen vorgeschlagenen Definitionen der numerischen Flüsse vorkommt. Beispiel wurden nachfolgend verschiedene Definitionen zusammengefasst:

**Tabelle 2:** Auflistung von numerischen Flüssen in COCKBURN [5]:

Methode:	$\hat{P}_{ji}$	$\hat{u}_i$
BABUSKA–ZLAMAL [1]	$C_{11} [u_h]$	$u_h$
BREZZI [3]	$-\alpha^r [u_h]$	$u_h$
BASSI–REBAY [2]	$\{P_{ji}\}$	$\{u_h\}$

Wir implementieren, in Anlehnung an BABUSKA und ZLAMAL, eine eigene Formulierung wie

folgt:

$$\begin{aligned}\hat{u}_i &= u_i \\ \hat{P}_{ji} &= P_{ji} - \eta \llbracket u_i \rrbracket .\end{aligned}\tag{4.15}$$

Im numerischen Fluss  $\hat{P}_{ji}$  wurde ein einziger Stabilisierungsfaktor (*penalty-factor*)  $\eta$  eingeführt. Der Stabilisierungsfaktor soll an Stellen ohne Riss die Sprünge  $\llbracket u_i \rrbracket$  und  $\llbracket \delta u_i \rrbracket$  zu Null zwingen. Die eigene Formulierung ermöglicht die Erfüllung der Gleichung (2.28)<sub>2</sub> in ihrer vereinfachten Form (4.5). Um dies zu sehen werden der Sprung  $\llbracket \hat{P}_{ji} \rrbracket$  und der Mittelwert  $\{ \hat{P}_{ji} \}$  berechnet:

$$\begin{aligned}\llbracket \hat{P}_{ji} \rrbracket &= - [P_{ji}] E_j = - ({}^l\sigma_{ki} F_{kj})_{,j} \\ \{ \hat{P}_{ji} \} &= \frac{1}{2} (\hat{P}_{ji}^+ + \hat{P}_{ji}^-) = \{ P_{ji} \} - \eta \llbracket u_i \rrbracket .\end{aligned}\tag{4.16}$$

Nach Einsetzen der Flüsse in Gleichung (4.12) ergibt sich die schwache Form wie folgt:

$$\begin{aligned}0 &= \int_{\Omega} (P_{ji} \delta u_{i,j} - f_i \delta u_i) dx + \\ &+ \int_{\Gamma} \left( \{ \delta u_i \} ({}^l\sigma_{ki} F_{kj})_{,j} - \{ P_{ji} \} \llbracket \delta u_i \rrbracket + \llbracket u_i \rrbracket \eta \llbracket \delta u_i \rrbracket \right) dS - \\ &- \int_{\partial\Omega} P_{ji} n_j \delta u_i ds .\end{aligned}\tag{4.17}$$

Diese schwache Form ist nun mit Hilfe der finiten Elementmethode lösbar und im nachfolgenden Kapitel numerisch umgesetzt.

### 4.3 Implementierung der schwachen Form

Die zuvor gefundene schwache Form soll nun numerisch gelöst werden. Dafür wird das quell-offene Programmpaket FEniCS verwendet, da FEniCS die Verwendung von einem diskontinuierlichen Galerkin Raum für die finiten Elemente erlaubt. Nun kann die in Gleichung (4.17) gefundene schwache Form in für FEniCS verständliche Syntax übersetzt werden:

```
1 Form = (P[j,i]*del_u[i].dx(j) - f[i]*del_u[i])*dx \
2 + (0.5 * (del_u('++')+del_u('--'))[i]*P_int[j,i].dx(j) \
3 - 0.5 * (P('++')+P('--'))[j,i]*(del_u('++')-del_u('--'))[i]*n('--')[j] \
4 + (del_u('++')-del_u('--'))[i]*eta*(u('++')-u('--'))[i])*dS \
5 - t[i]*del_u[i]*ds
```

Wobei diese Syntax eine starke Ähnlichkeit zur mathematischen Formulierung in Gleichung (4.17) besitzt. Damit bietet die Programmierung mit FEniCS eine intuitive Möglichkeit zur Diskretisierung von partiellen Differentialgleichungen.

### 4.4 Einbau von Rissen

In der hier verwendeten Methode (diskontinuierliche Galerkin Methode) kann eine Rissbildung und das Wachstum des Risses simuliert werden. Dabei werden Risse als singuläre Flächen aufgefasst. Die singulären Flächen befinden sich in der diskontinuierlichen Galerkin Methode zwischen den Elementen. Somit breiten sich die Risse entlang von Elementberandungen aus.

Durch Einführen des *penalty-factors*  $\eta$  wird die Rissbildung unterbunden. Deshalb muss an Stellen, an denen sich ein Riss bildet, der *penalty-factor* auf null gesetzt werden, um Sprünge in den Verschiebungen zu ermöglichen.

Bei der Herleitung der Bilanzgleichungen im Kapitel 2 wurde von einem Kontinuum mit singulärer Fläche ausgegangen, auf der alle Größen stetig sind. Dadurch gibt es keine singulären Linien auf der singulären Fläche. Für die Rissentwicklung bedeutet das, dass es zu keiner Verzweigung von Rissen kommen kann. Denn eine Verzweigung eines Risses wäre eine Singularität auf der singulären Fläche. Dies wurde in der Herleitung jedoch nicht berücksichtigt. Das in dieser Arbeit verwendete Risswachstum<sup>1</sup> wird in zwei Phasen unterschieden:

1. Rissinitialisierung und
2. Rissfortschritt.

Ein Struktogramm der Rissalgorithmus ist in Abbildung 7 zu sehen und eine Beschreibung der einzelnen Phasen wird in den nächsten Abschnitten erläutert.

#### 4.4.1 Rissinitialisierung

Während der Rissinitialisierung (solange wie `len(crackPeakList) == 0`) wird der Anfang vom Riss gesucht. Dabei wird nach dem Aufbringen jedes Last-increments der Knoten im Netz mit der höchsten Vergleichsspannung gesucht. Ist die maximale Vergleichsspannung größer als die Bruchspannung (`sigma_f`), beginnt dort der Riss. Dieser Knoten wird nun als Risspitze definiert (erste Knoten-ID in `crackPeakList.append(tmpIndex)`).

Quelltext 1: Rissinitiiierung im Risswachstumsalgorithmus

```

1 | if len(crackPeakList) == 0:
2 |     tmpValue = 0
3 |     tmpIndex = 0
4 |     print("search startpoint")
5 |     for n__ in vertices(mesh):
6 |         ## vertex nodes
7 |         value = crackStress(n__.point())
8 |         if tmpValue < value:
9 |             tmpValue = value
10 |            tmpIndex = n__.index()
11 |
12 |    if sigma_f < tmpValue:
13 |        crackPeakList.append(tmpIndex)

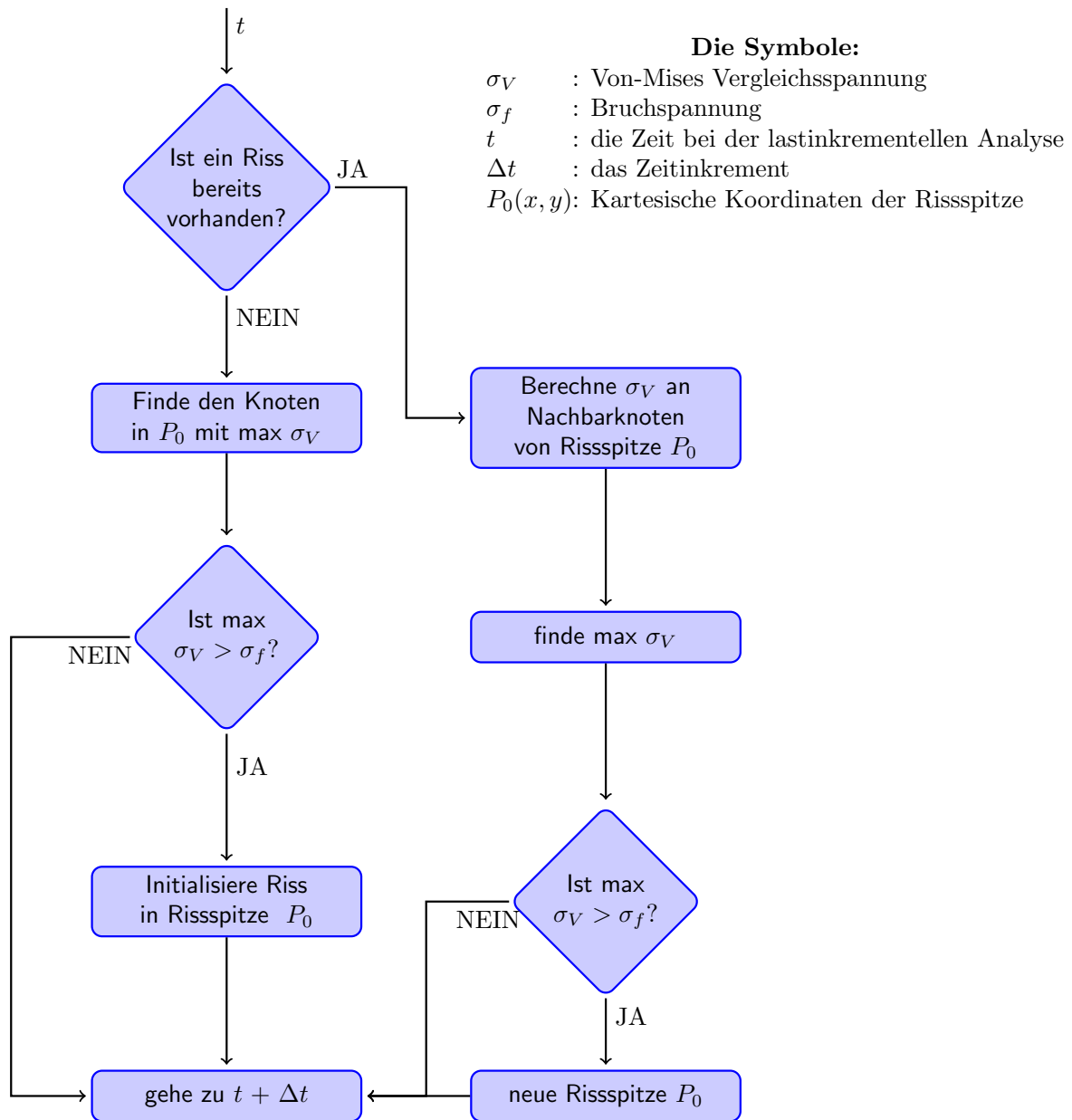
```

Wenn eine Knoten-ID in `crackPeakList` existiert, wird nun das Risswachstum simuliert.

#### 4.4.2 Rissfortschritt

Dafür werden zunächst alle zur Risspitze benachbarten Knoten gesucht die nicht in der Liste (`crackPeakList`) sind (Quelltext 2 Zeile 14-22). Dies ist wichtig, damit es nicht zu dem in Abbildung 8 gezeigtem Freischneiden von Starrkörpern kommen kann.

<sup>1</sup>Der in dieser Arbeit implementierte Algorithmus ist auf ein 2D-Modell beschränkt.



**Abbildung 7:** Struktogramm vom Rissalgorithmus

Es werden drei Listen gespeichert. Eine, in der die ID's der Nachbarknoten gespeichert wird und eine zweite Liste mit den dazugehörigen ID's der Kanten. In der dritten Liste werden die Werte der Vergleichsspannung am Mittelpunkt der Elementkanten gespeichert.

Nach dem Speichern dieser drei Listen wird nun der höchste Wert in der Liste mit den Vergleichsspannungen ermittelt. Ist dieser Wert über der Bruchspannung `sigma_f` wird der Riss verlängert. Dafür wird zunächst auf der entsprechenden Kante der *penalty-factor* zu null gesetzt. Dies geschieht durch Markieren der Kante mit "`facet_domains[listFacets[ID]] = 1`". Dadurch wird das Oberflächenintegral mit dem *penalty-factor* nicht mehr auf dieser Kante ausgewertet, was einem *penalty-factor* von null entspricht. Außerdem wird die Liste mit den Rissknoten verlängert. Dafür wird die neue Risspitze an die Liste mit den Knoten entlang des

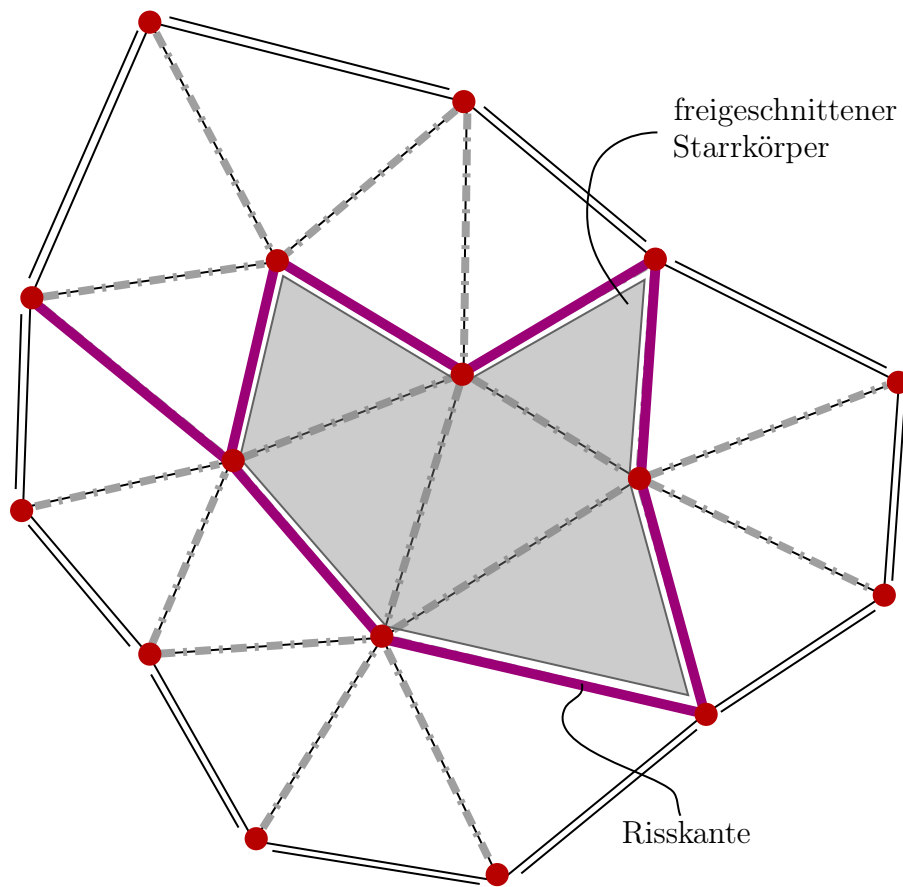


Abbildung 8: Freischneiden von Starrkörpern

Risses angehängt `"crackPeakList.append(listNodes[ID])"`. Nun wird die Belastung um ein Lastinkrement vergrößert.

Quelltext 2: Risswachstum im Risswachstumsalgorithmus

```

1 else:
2     print("search new facet")
3     listNodes = []
4     listValues = []
5     listFacets = []
6
7     for f__ in facets(mesh):
8         node1ID = f__.entities(0)[0]
9         node2ID = f__.entities(0)[1]
10        node1 = Vertex(mesh,node1ID).midpoint()
11        node2 = Vertex(mesh,node2ID).midpoint()
12        tmpStress = crackStress((node1+node2)/2.0)
13
14        if node1ID == crackPeakList[-1] and not (node2ID in crackPeakList):
15            listNodes.append(node2ID)
16            listValues.append(tmpStress)
17            listFacets.append(f__)
18
19        elif node2ID == crackPeakList[-1] and not (node1ID in crackPeakList):
20            listNodes.append(node1ID)

```

```
21     listValues.append(tmpStress)
22     listFacets.append(f__)
23
24     if max(listValues) > sigma_f:
25         ID = listValues.index(max(listValues))
26         facet_domains[listFacets[ID]] = 1
27         crackPeakList.append(listNodes[ID])
28         print("newCrack")
```



## 5 Ergebnisse und Auswertung

Das in dieser Arbeit verwendete Modell ist ein 2D-Balken, der eine Kerbe bei halber Länge besitzt (Abbildung 9). Die Abmessungen des Balkens betragen:

- $l_1 = 20 \text{ cm}$  ,
- $l_2 = 5 \text{ cm}$  ,
- $\delta l_1 = 2 \text{ mm}$  und
- $\delta l_2 = 10 \text{ mm}$  .

Als Randbedingung wurden am rechten und linken Rand des Balkens DIRICHLET Randbedingungen vorgegeben und an allen anderen Rändern wurden NEUMANN Randbedingungen vorgegeben. Die rechte Seite wurde fest eingespannt und auf der linken Seite wurde eine Verschiebung von  $\Delta l = 0,45 \text{ mm}$  inkrementell vorgegeben.

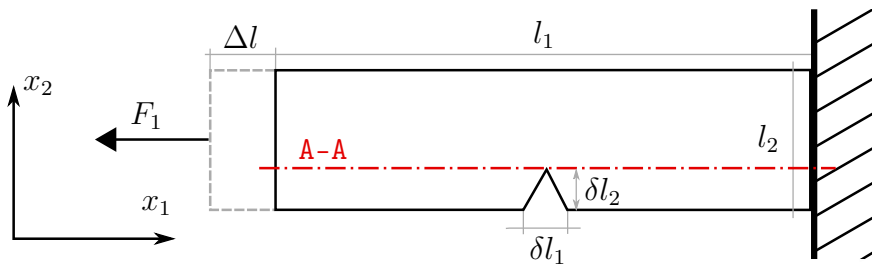
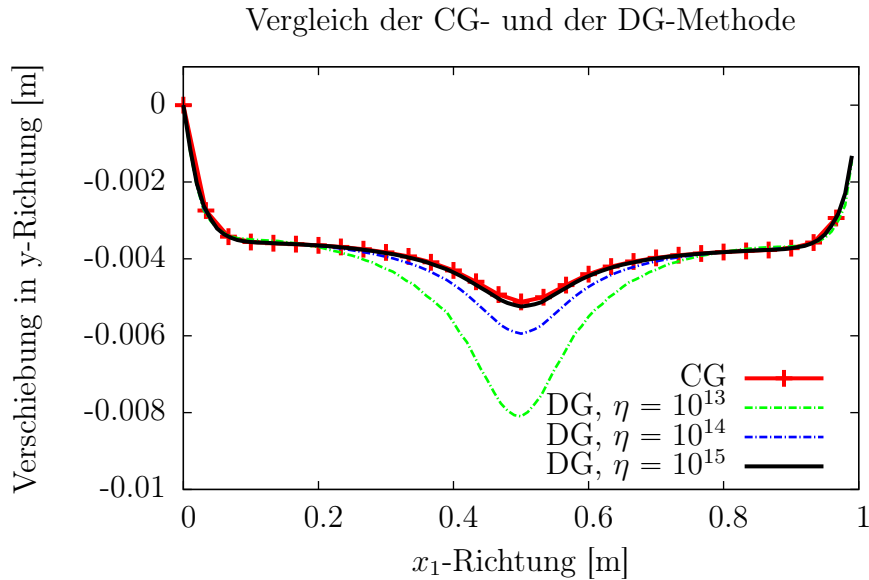


Abbildung 9: Prinzipskizze des untersuchten Balkens

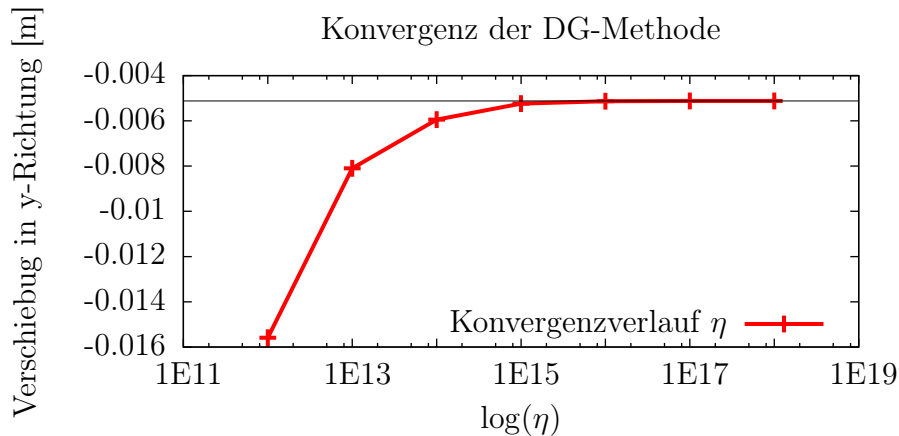
### 5.1 Bestimmung vom *penalty factor*

Zunächst bestand ein großes Interesse daran, zu untersuchen, ob die Lösung mit der diskontinuierlichen Galerkin Methode ohne Risse mit der Lösung der kontinuierlichen Galerkin Methode übereinstimmt. Dafür wurde zunächst das Risswachstum abgeschaltet und Simulationen mit sich unterscheidenden *penalty-factoren* durchgeführt. Der Vergleich der Verschiebungsergebnisse ( $u_2$ -Richtung) ist in Abbildung 10 dargestellt. Die Vergleichslösung der kontinuierlichen Galerkin (CG) Methode ist in rot eingezeichnet.



**Abbildung 10:** Verlauf der Verschiebungen an der oberen Kante ( $x_2 = l_2$ )

Die Ergebnisse aus Abbildung 10 haben den Anschein einer Konvergenz. Um dies genauer untersuchen zu können, wurde anschließend eine Konvergenzanalyse durchgeführt. Dafür wurde die Verschiebung in  $x_2$ -Richtung über die *penalty-factor* aufgetragen. Die Verschiebungen wurden am Punkt der größten Einschnürung, also über der Kerbe im Schnitt A-A (Abbildung 9) ausgewertet. Dargestellt sind die Ergebnisse in Abbildung 11.



**Abbildung 11:** Konvergenz des *penalty factors*

Aus der Untersuchung ist gut zu erkennen, dass die Lösung der diskontinuierlichen Galerkin Methode bei größer werdendem *penalty-factor* gegen die Lösung der kontinuierlichen Galerkin Methode konvergiert. Dies dient dazu, dass wir  $\eta$  bestimmen können. Sogar wenn keine kontinuierliche Lösung für das Problem existiert, kann der *penalty-factor*  $\eta$  bestimmt werden. Dafür wird die Eigenschaft ausgenutzt, dass die diskontinuierliche Lösung für wachsende  $\eta$  konvergiert. Eine untere Schranke für  $\eta$  lässt sich somit leicht bestimmen, indem die Änderung der Verschiebungen bei wachsendem *penalty-factor* untersucht wird.

## 5.2 Aussehen eines Risses

Das in Abbildung 13 gezeigte Rissbild ist das Ergebnis, der in dieser Arbeit hergeleiteten Methode zur Simulation von Rissen.

Es ist dabei zu beachten, dass in der Simulation nur ein kleines Risswachstum simuliert werden konnte. Ein Weiterführen der Simulation führte immer im nächsten Iterationsschritt zum Abbruch. Der Grund dafür konnte nicht ermittelt werden.

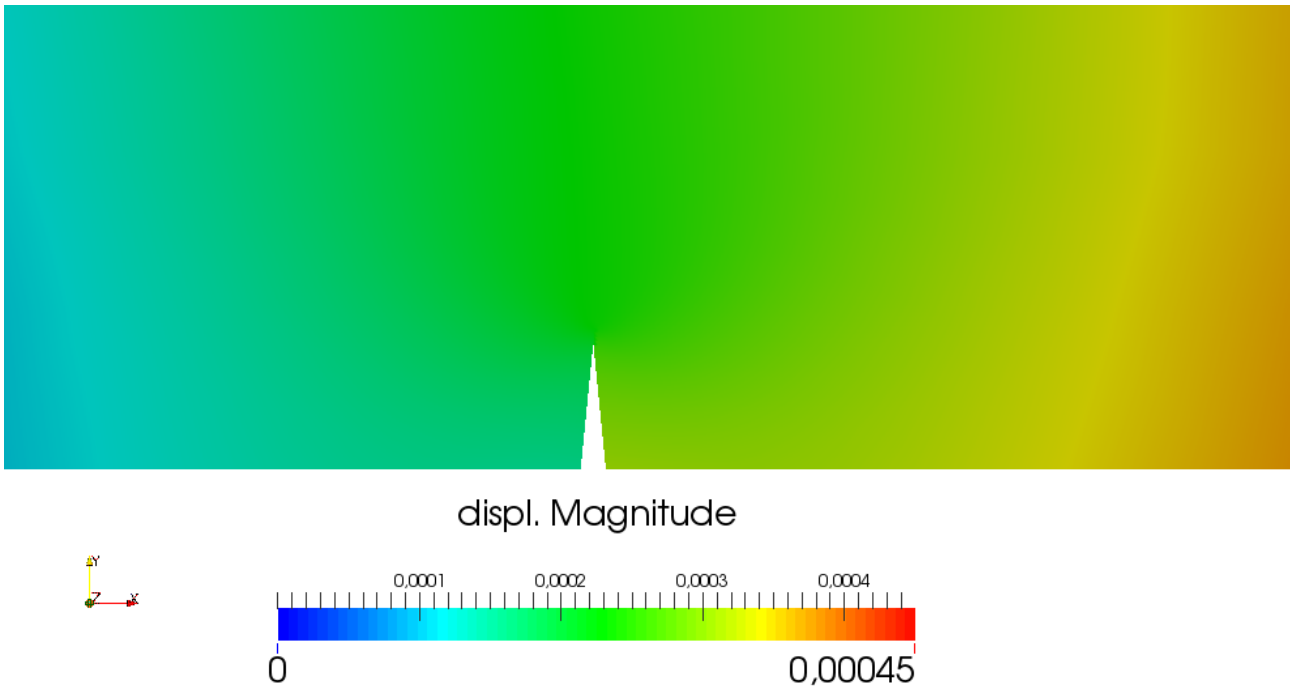


Abbildung 12: Verschiebungsmagnitude des Balkens bei  $\Delta l = 0,225$  mm

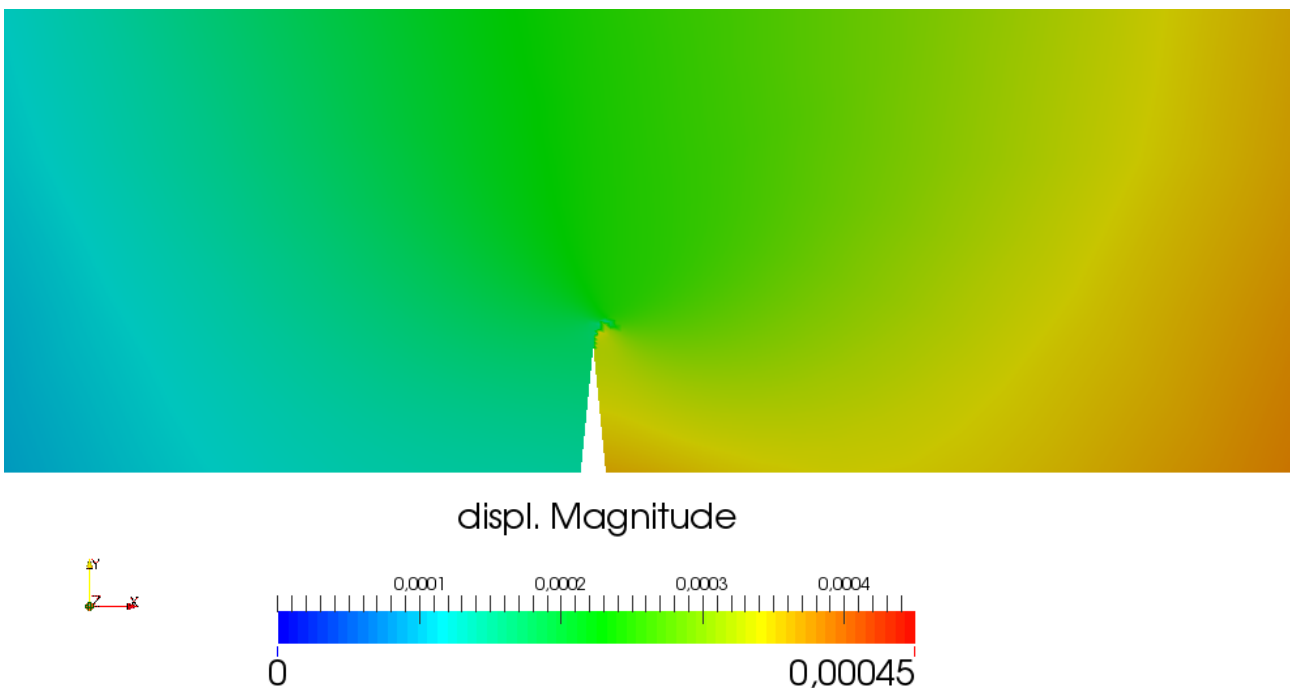


Abbildung 13: Verschiebungsmagnitude des Balkens bei  $\Delta l = 0,45$  mm

### 5.3 Ausblick

Zur Weiterführung dieses Projektes bietet es sich zunächst an, zu untersuchen, warum die Rechnung nur ein kleines Risswachstum erlaubt. Das ist wichtig, um anschließend die Netzunabhängigkeit des Risswachstums zu untersuchen. Netzunabhängigkeit des Risswachstums heißt, dass es ab einer gewissen charakteristischen Elementgröße keine Veränderung des Rissbildes gibt, wenn das Netz weiter verfeinert wird.

Auch ist eine Untersuchung des Oberflächenspannungstensors interessant. Der Oberflächenspannungstensor ist in der klassischen Kontinuumsmechanik eine wenig verbreitete Größe. Sie wurde im Kontext der Rissbildung noch nicht untersucht. Dadurch ist nicht geklärt, wie diese Größe das Kontinuum beeinflusst.

Ein weiterer Punkt ist die Herleitung der Bilanzgleichungen mit Singularitäten auf der singulären Fläche. Dies würde die Rissverzweigung ermöglichen.

# Anhang

Für die Herleitung der Bilanzgleichungen werden einige Vorbetrachtungen benötigt.

## Deformationsgradient

Der Deformationsgradient ist definiert als:

$$F_{ij} = \frac{\partial x_i}{\partial X_j}. \quad (\text{A.1})$$

## Differentielles Linienelement

Das differentielle Linienelement  $dL_i$  lässt sich mit Hilfe des tangential Vektors  $dT_i$  wie folgt parametrisieren:

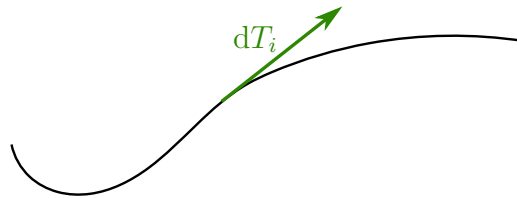


Abbildung 14: differentielle Linien mit Basisvektor

$$\begin{aligned} dL_i &= \frac{\partial X_i}{\partial \xi} d\xi = dT_i d\xi \\ dl_i &= \frac{\partial x_i}{\partial \xi} d\xi = \frac{\partial x_i}{\partial X_k} \frac{\partial X_k}{\partial \xi} d\xi \\ dl_i &= F_{ik} dT_k d\xi \\ dl_i &= F_{ik} dL_k. \end{aligned} \quad (\text{A.2})$$

## Differentielles Flächenelement

Das differentielle Flächenelement  $dA_i$  lässt sich mit Hilfe der tangentialen Vektoren  $dT_i^\alpha$  folgendermaßen parametrisieren:

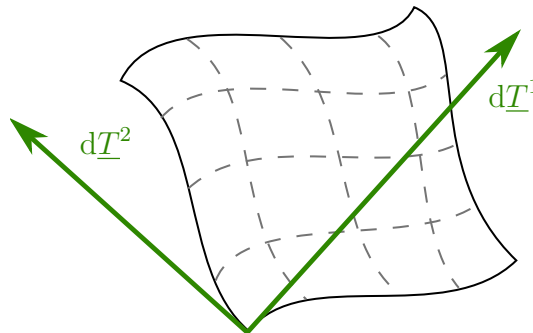


Abbildung 15: differentielles Flächenelement mit Orthonormalbasis

$$\begin{aligned}
dA_i &= \varepsilon_{ijk} dT_j^1 dT_k^2 \\
da_i &= \varepsilon_{ijk} \frac{\partial x_j}{\partial X_l} \frac{\partial x_k}{\partial X_m} dT_l^1 dT_m^2 \\
&= \varepsilon_{ijk} F_{jl} F_{km} dT_l^1 dT_m^2.
\end{aligned} \tag{A.3}$$

Es gilt dabei die Relation:

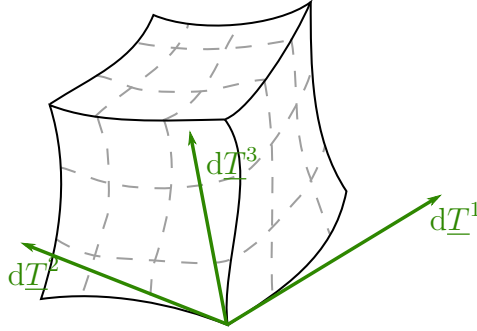
$$\varepsilon_{ijk} F_{il} F_{jm} F_{kn} = \det(\mathbf{F}) \varepsilon_{lmn}. \tag{A.4}$$

Damit ergibt sich das differentielle Flächenelement  $da_i$  in der aktuellen Konfiguration zu:

$$\begin{aligned}
F_{ip} da_i &= F_{il} \varepsilon_{ijk} F_{jl} F_{km} dT_l^1 dT_m^2 \\
&= \underbrace{\det(\mathbf{F})}_{J} \varepsilon_{plm} dT_l^1 dT_m^2 \\
\delta_{ip} da_i &= J(\mathbf{F}^{-1})_{pr} \underbrace{\varepsilon_{plm} dT_l^1 dT_m^2}_{dA_p} \\
da_p &= J(\mathbf{F}^{-1})_{pr} dA_p \\
n_p da &= J(\mathbf{F}^{-1})_{pr} N_p dA.
\end{aligned} \tag{A.5}$$

## Differentielles Volumenelement

Das differentielle Volumenelement lässt sich mit Hilfe der Orthonormalbasis  $dT^i$  aus der Referenzkonfiguration bestimmen.



**Abbildung 16:** differentielles Volumenelement mit Orthonormalbasis

$$\begin{aligned}
dV &= \varepsilon_{ijk} dT_i^1 dT_j^2 dT_k^3 \\
dv &= \varepsilon_{ijk} d\tau_i^1 d\tau_j^2 d\tau_k^3 \\
dv &= J \varepsilon_{lmnk} F_{il} dT_l^1 F_{jm} dT_m^2 F_{kn} dT_n^3, \text{ mit Gleichung A.4} \\
dv &= J dV
\end{aligned} \tag{A.6}$$

## A.1 GAUSSscher Satz mit singulären Flächen

In diesem Abschnitt wird nun der GAUSSsche Satz für Volumen mit singulären Flächen hergeleitet:

$$\begin{aligned}
\int_{B^+ \cup B^-} \frac{\partial f}{\partial X_k} dV &= \int_{B^+} \frac{\partial f}{\partial X_k} dV + \int_{B^-} \frac{\partial f}{\partial X_k} dV \\
&= \int_{\partial B^+ = A^+ \cup {}^s A} f N_k dA + \int_{\partial B^- = A^- \cup {}^s A} f N_k dA \\
&= \int_{A^+} f N_k dA - \int_{{}^s A} f^+ E_k dA + \int_{A^-} f N_k dA + \int_{{}^s A} f^- E_k dA \\
&= \int_{A^+ \cup A^-} f N_k dA - \int_{{}^s A} [f] E_k dA.
\end{aligned} \tag{A.7}$$

Dabei steht  $[ \cdot ]$  für die Sprungklammer und wird wie folgt berechnet:

$$[f] = f^+ - f^-. \tag{A.8}$$

Nun wird noch der GAUSSsche Satz für Flächen ohne Singularitäten eingeführt. Dabei wurde  $f_k$  auf der rechten Seite in den Anteil in Normalenrichtung der Fläche  $f_\perp E^k$  und in den Anteil Tangential zur Fläche  $f^\Delta T_\Delta^k$  zerlegt. Durch Ausnutzen von  $E^k \nu_k^0 = 0$  folgt:

$$\begin{aligned}
\int_{\partial {}^s A} f_k \nu_k^0 dL &= \int_{\partial {}^s A} (f^\Delta T_\Delta^k + f_\perp E^k) \nu_k^0 dL \\
&= \int_{\partial {}^s A} f^\Delta T_\Delta^k \nu_k^0 dL \\
&= \int_{\partial {}^s A} f^\Delta T_\Delta^k (\nu_0^\Gamma T_\Gamma^k + \nu_{0\perp} E^k) dL \\
&= \int_{\partial {}^s A} f^\Delta T_\Delta^k \nu_0^\Gamma T_\Gamma^k dL \\
&= \int_{\partial {}^s A} f^\Delta G_{\Delta\Gamma} \nu_0^\Gamma dL \\
&= \int_{\partial {}^s A} f^\Delta \nu_0^\Delta dL \\
&= \int_{{}^s A} f^\Delta_{;\Delta} dA.
\end{aligned} \tag{A.9}$$

Das ist das Ergebnis für den GAUSSschen Satz auf Flächen angewendet.

## A.2 Transporttheorem für Volumenintegrale

Für die Überführung der allgemeinen Bilanzgleichungen für materielle Volumen in eine Bilanzgleichung für reguläre Punkte und Punkte auf einer singulären Fläche werden Transporttheo-

reme benötigt. Diese werden nachfolgend eingeführt:

$$\left( \int_{v^+ \cup v^-} {}^B\psi \, dv \right)^\cdot = \int {}^B\psi^\cdot \, dv + \int {}^B\psi (dv)^\cdot = \int \left( {}^B\psi^\cdot + {}^B\psi \frac{\partial v_i}{\partial x_i} \right) dv. \quad (\text{A.10})$$

Dabei wurde verwendet, dass die Ableitung des differentiellen Volumenelements wie folgt geschrieben werden kann:

$$(dv)^\cdot = (dx \, dy \, dz)^\cdot = dv \left( \frac{dx^\cdot}{dx} + \frac{dy^\cdot}{dy} + \frac{dz^\cdot}{dz} \right) = \frac{\partial v_i}{\partial x_i} dv \quad (\text{A.11})$$

### A.3 Transporttheorem für Flächenintegrale

Für das Transporttheorem auf Flächen gilt:

$$\left( \int_{sa} {}^a\psi \, da \right)^\cdot = \int {}^a\psi^\cdot \, da + \int {}^a\psi (da)^\cdot. \quad (\text{A.12})$$

Nun wird die Zeitableitung des differentiellen Flächenelements  $(da)^\cdot$  gesucht. Dafür wird zunächst an das differentiellen Flächenelement  $da$  erinnert:

$$da = \sqrt{g} \, d\tau^1 \, d\tau^2. \quad (\text{A.13})$$

Mit  $\tau_\alpha$  und  $\tau_\beta$ ,  $\alpha, \beta \in [1, 2]$  als Tangentenvektoren (siehe Gleichung A.5). Es gilt die Relation:

$$g_{\alpha\beta} = \tau_\alpha^i \tau_\beta^i, \quad (\text{A.14})$$

wobei  $g_{\alpha\beta}$  den Metriktenor in kovarianter Schreibweise darstellt. Nach Ausführen der Zeitableitung in Gleichung (A.12) folgt mit der Determinante,  $g = \det(g_{\alpha\beta})$ ,

$$\begin{aligned} (da)^\cdot &= (\sqrt{g})^\cdot \, d\tau^1 \, d\tau^2, \\ (da)^\cdot &= \frac{1}{2} \frac{1}{\sqrt{g}} g^\cdot \, d\tau^1 \, d\tau^2. \end{aligned} \quad (\text{A.15})$$

Die zeitliche Ableitung ergibt sich zu:

$$g^\cdot = \frac{\partial g}{\partial g_{\alpha\beta}} g_{\alpha\beta}^\cdot. \quad (\text{A.16})$$

Nach Umschreiben der partiellen Ableitung folgt:

$$g^\cdot = g^{\alpha\beta} g_{\alpha\beta}^\cdot. \quad (\text{A.17})$$

Zur Berechnung der zeitlichen Ableitung der Oberflächenmetrik werden die Tangentialvektoren eingesetzt:

$$\begin{aligned} g_{\alpha\beta}^\cdot &= (\tau_\alpha^i \tau_\beta^i)^\cdot \\ &= 2\tau_{(\alpha}^\cdot \tau_{\beta)}^i = \text{symm.} \end{aligned} \quad (\text{A.18})$$



und:

$$\begin{aligned}
\tau_\alpha^{\cdot i} &= \left( \frac{\partial x^i}{\partial \xi^\alpha} \right) \cdot \\
&= \frac{\partial^2 x_i}{\partial \xi^\alpha \partial t} \\
&= \frac{\partial w_i}{\partial \xi^\alpha} \\
&= \frac{\partial}{\partial \xi} (w^\beta \tau_\beta^i + w_\perp e^i) .
\end{aligned} \tag{A.19}$$

$w_i$  ist die Geschwindigkeit der Partikel auf der singulären Fläche. Diese setzt sich aus tangentialem Anteil  $w_i \tau_\beta^i$  und dem senkrechten Anteil  $w_\perp e^i$  zusammen. Im Folgenden werden die Christoffelsymbole  $\Gamma$  eingeführt:

$$\begin{aligned}
\frac{\partial \tau_\beta^i}{\partial \xi^\alpha} &= \Gamma_{\beta\alpha}^\gamma \tau_\gamma^i + b_{\beta\alpha} e^i, \text{ mit} \\
b_{\beta\alpha} &= \frac{\partial^2 x_i}{\partial \xi^\alpha \partial \xi^\beta} e^i .
\end{aligned} \tag{A.20}$$

Dabei ist  $b_{\beta\alpha}$  der Krümmungstensor und  $e_i$  ist der Normalenvektor auf der betrachteten singulären Oberfläche. Die Darstellung der Christoffelsymbole mit Hilfe der Oberflächenmetrik  $g_{ij}$  lautet:

$$\Gamma_{\beta\alpha}^\gamma = \frac{1}{2} g^{\gamma\Lambda} \left( \frac{\partial g_{\beta\Lambda}}{\partial \xi^\gamma} + \frac{\partial g_{\alpha\Lambda}}{\partial \xi^\beta} - \frac{\partial g_{\beta\alpha}}{\partial \xi^\Lambda} \right) \tag{A.21}$$

$$g_{\alpha\beta}^{\cdot} = 2\tau_\beta^i \left( \frac{\partial w^\omega}{\partial \xi^{(\alpha}} \tau_\omega^i + w^\omega \underbrace{\frac{\partial \tau_\omega^i}{\partial \xi^{(\alpha}}}_{\Gamma_{w(\alpha}^\gamma \tau_\gamma^i + b_{w(\alpha} e^i} + \frac{\partial w_\perp}{\partial \xi^{(\alpha}} e^i + w_\perp \frac{\partial e^i}{\partial \xi^{(\alpha}} \right), \text{ mit Gl. A.19} \tag{A.22}$$

$$g_{\alpha\beta}^{\cdot} = \underbrace{2g_{\omega\beta} \frac{\partial w^\omega}{\partial \xi^{(\alpha}} + 2g_{\gamma\beta} w^\omega \Gamma_{w(\alpha}^\gamma}_{2g_{\omega\beta} w_{(\alpha}^\gamma} + 2\tau_\beta^i w_\perp \frac{\partial e^i}{\partial \xi^{(\alpha}} . \tag{A.23}$$

Der Ausdruck  $w_{;\alpha}^\gamma$  bezeichnet die kovariante Ableitung. Die Ableitung von dem Metriktensor ist 0. Einsetzen in Gl. (A.15):

$$\begin{aligned}
(da)^\cdot &= \sqrt{g} g^{\alpha\beta} \left( g_{\gamma\beta} w_{;\alpha}^\gamma + \tau_\beta^i w_\perp \frac{\partial e^i}{\partial \xi^{(\alpha}} \right) d\tau^1 d\tau^2 \\
&= \sqrt{g} \left( \underbrace{\delta_{\gamma\alpha}^\alpha w_{;\alpha}^\gamma}_{w_{;\alpha}^\alpha} + g^{\alpha\beta} \tau_\beta^i w_\perp \frac{\partial e^i}{\partial \xi^{(\alpha}} \right) d\tau^1 d\tau^2
\end{aligned} \tag{A.24}$$

Die Ableitung des Normalenvektors:

$$\frac{\partial e^i}{\partial \xi^\alpha} = A_\alpha^\gamma \tau_\gamma + B_\alpha e^i, \tag{A.25}$$

wird im Folgenden gesucht, wobei  $A$  und  $B$  erstmal unbekannt sind. Beide Seiten werden mit dem Normalenvektor  $e^i$  multipliziert:

$$\begin{aligned}
\frac{\partial e^i}{\partial \xi^\alpha} e^i &= \cancel{A_\alpha^\gamma \tau_\gamma e^i} + B_\alpha \underbrace{e^i e^i}_1 \\
\Rightarrow B_\alpha &= \frac{\partial e^i}{\partial \xi^\alpha} e^i = \cancel{\frac{\partial e^i e^i}{\partial \xi^\alpha}} - e^i \frac{\partial e^i}{\partial \xi^\alpha} = 0. \\
\frac{\partial e^i}{\partial \xi^\alpha} \tau^i &= A_\alpha^\gamma g_{\gamma\Lambda} \\
\Rightarrow A_\alpha^\gamma &= g^{\gamma\Lambda} \frac{\partial e^i}{\partial \xi^\alpha} \tau_\Lambda^i \\
&= g^{\gamma\Lambda} \left( \frac{\partial e^i \tau_\Lambda^i}{\partial \xi^\alpha} - e^i \frac{\partial \tau_\Lambda^i}{\partial \xi^\alpha} \right) \\
&= -g^{\gamma\Lambda} \underbrace{e^i \frac{\partial \tau_\Lambda^i}{\partial \xi^\alpha}}_{b_{\Lambda\alpha}} \\
&= -g^{\gamma\Lambda} b_{\Lambda\alpha}.
\end{aligned} \tag{A.26}$$

Die Ableitung des Normalenvektors nach  $\xi^\alpha$  ergibt sich damit zu:

$$\frac{\partial e^i}{\partial \xi^\alpha} = -g^{\gamma\Lambda} b_{\Lambda\alpha} \tau_\gamma^i. \tag{A.27}$$

Wobei  $b_{\Lambda\alpha}$  der Krümmungstensor ist. Einsetzen in Gl. (A.24) liefert:

$$\begin{aligned}
(da)^\bullet &= \sqrt{g} (w^\alpha_{;\alpha} + g^{\alpha\beta} \tau_\beta^i w_\perp (-g^{\gamma\Lambda} b_{\Lambda(\alpha} \tau_\gamma^i)) d\tau^1 d\tau^2 \\
&= (w^\alpha_{;\alpha} - g^{\alpha\Lambda} b_{\Lambda\alpha} w_\perp) da, \text{ mit Gl. A.13} \\
K_m &= \frac{1}{2} g^{\alpha\beta} b_{\alpha\beta} \\
\Rightarrow (da)^\bullet &= (w^\alpha_{;\alpha} - 2K_m w_\perp) da.
\end{aligned} \tag{A.28}$$

$K_m$  wird als Krümmungsradius bezeichnet.

## A.4 FEniCS Code zur Berechnung von Rissen mit der diskontinuierlichen Galerkin Methode

**Quelltext 3:** FEniCS Code zur Rissberechnung mit der diskontinuierlichen Galerkin Methode

```

1 from dolfin import *
2 import numpy as np
3 from ufl.algorithms import *
4
5 parameters["form_compiler"]["cpp_optimize"] = True
6 parameters["form_compiler"]["optimize"] = True
7 parameters["form_compiler"]["representation"] = "quadrature"
8 parameters["form_compiler"]["quadrature_degree"] = 2
9 parameters['linear_algebra_backend'] = 'PETSc' #'PETSc', 'uBLAS', 'STL'

```

```

10
11 parameters['allow_extrapolation'] = True
12 #Dimention = D
13 D = 2
14 xlength = 0.20 # [m]
15 ylength = 0.05 # [m]
16
17
18 # index notation
19 i, j, k, l, m = indices(5)
20 # metric
21 delta = Identity(D)
22
23 # =====
24 # ===== DEFINE GEOMETRY / MESH =====
25 # =====
26 #
27 if not has_cgal():
28     print "DOLFIN must be compiled with CGAL to run this demo."
29     exit(0)
30
31 # Create empty Mesh
32 mesh = Mesh() # mesh for calculation
33 mesh2 = Mesh() # fine mesh for plot
34 meshSize = 0.04
35
36 # Create list of polygonal domain vertices
37 notchx_offs = xlength/2.0 # xlength/2.0 is center
38 notchy_offs = ylength/5.0
39 domain_vertices = [Point(0.0          , 0.0),
40                    Point((notchx_offs)-1e-3, 0.0),
41                    Point((notchx_offs)  , notchy_offs),
42                    Point((notchx_offs)+1e-3, 0.0),
43                    Point(xlength        , 0.0),
44                    Point(xlength        , ylength),
45                    Point(0.0            , ylength),
46                    Point(0.0            , 0.0)]
47
48 # Generate mesh and plot
49 PolygonalMeshGenerator.generate(mesh , domain_vertices, meshSize);
50
51 center=Point(notchx_offs, notchy_offs)
52 def refine_interval(radius,mesh):
53     markers = MeshFunction("bool", mesh, D)
54     markers.set_all(False)
55     # Mark cells for refinement
56     for cell in cells(mesh):
57         if cell.midpoint().distance(center) < radius:
58             markers[cell.index()] = True
59     # Refine mesh
60     mesh = refine(mesh, markers)

```

```

61     return mesh
62
63 mesh=refine_interval(0.06 ,mesh)
64 mesh=refine_interval(0.05 ,mesh)
65 mesh=refine_interval(0.04 ,mesh)
66 mesh=refine_interval(0.03 ,mesh)
67 mesh=refine_interval(0.02 ,mesh)
68 mesh=refine_interval(0.015,mesh)
69 mesh=refine_interval(0.01 ,mesh)
70 mesh=refine_interval(0.01 ,mesh)
71 mesh=refine_interval(0.01 ,mesh)
72
73 #plot(mesh, interactive=True)
74 mesh2=mesh
75 mesh2=refine_interval(0.06,mesh2)
76
77 # origin is in the middle of the plate
78 space      = VectorFunctionSpace(mesh, 'DG', 1)
79 boundaries = FacetFunction('uint', mesh)
80 left       = CompiledSubDomain('near(x[0], 0.0) && on_boundary')
81 right      = CompiledSubDomain('near(x[0], 1) && on_boundary', l=xlength)
82 top        = CompiledSubDomain('near(x[1], 1) && on_boundary', l=ylength)
83 bottom     = CompiledSubDomain('near(x[1], 0.0) && on_boundary')
84
85 boundaries.set_all(0)
86 left.mark(boundaries,1)
87 right.mark(boundaries,2)
88
89 del_u      = TestFunction(space)
90 d_u        = TrialFunction(space)
91 u          = Function(space)
92 u0         = Function(space)
93 u_t0       = Function(space)
94
95 left_u     = Constant((0.0,0.0)) # [m]
96 right_u    = Expression(('0.001*t', '0.0'),t=0) # [m]
97 null_ij    = Constant(((0.0, 0.0),(0.0, 0.0))) # [-]
98
99 bc1        = DirichletBC(space, left_u, left, method='geometric')
100 bc2        = DirichletBC(space, right_u, right, method='geometric')
101 bc         = [bc1,bc2]
102
103 # =====
104 # ===== MARK GAP =====
105 # =====
106 #
107 cell_domains = CellFunction('uint', mesh)
108 facet_domains = FacetFunction('uint', mesh)
109 facet_domains_t0 = FacetFunction('uint', mesh)
110
111 # =====

```

```

112 # ===== MATERIAL PARAMETER =====
113 # =====
114 #
115
116 rho_0 = 1.0
117 # material coeff.s of Pt
118 # mat vals from: http://www.goodfellow.com/G/Platin.html
119 nu = 0.39 # [-]
120 E = 170e+09 # [Pa]
121 G = E/(2.0*(1.0+nu))
122
123 # Lamé parameters (lambda has another meaning in python as def,
124 # thus is forbidden to use as a variable)
125 lambda = 2.0 * G * nu / (1.0 - 2.0 * nu)
126 mu = G
127
128 f = as_tensor([0.,0.]) # Body Force
129 trac = as_tensor([0.,0.]) # surface traction
130
131 # =====
132 # ===== define python functions =====
133 # =====
134 #
135 #Element surface normals
136 n = FacetNormal(mesh)
137
138 # deformation gradient
139 F = as_tensor(u[i].dx(j) + delta[i,j], (i,j))
140 F0 = as_tensor(u0[i].dx(j) + delta[i,j], (i,j))
141
142 # right Cauchy-Green deformation tensor
143 C = as_tensor(F[i,k]*F[i,j], (k,j))
144 C0 = as_tensor(F0[i,k]*F0[i,j], (k,j))
145
146 # Green - Lagrange strain tensor
147 E = as_tensor(1./2.*(C[k,j] - delta[k,j]), (k,j))
148 E0 = as_tensor(1./2.*(C0[k,j] - delta[k,j]), (k,j))
149
150 # second Piola - Kirchhoff tensor
151 S = as_tensor(lambda*E[1,1]*delta[k,j]+2.0*mu*E[k,j], (k,j))
152 S0 = as_tensor(lambda*E0[1,1]*delta[k,j]+2.0*mu*E0[k,j], (k,j))
153
154 # first Piola -Kirchhoff tensor
155 P = as_tensor(F[i,j]*S[j,k], (k,i))
156 P0 = as_tensor(F0[i,j]*S0[j,k], (k,i))
157
158 # Zugfestigkeit Titan
159 # Wikipedia
160 sigma_f = 235.0e+6
161
162 # surface tension

```

```

163 #ff = 2.574 # [J/m*m] - Pt
164 ff = Expression('2.574*p', p=1.0, cell=triangle) # [m]
165
166 sigma_l = ff*delta
167 interfaceP = as_tensor(sigma_l('+')[k,i]*F('+')[k,j] , (j,i))
168
169 # cauchy stress tensor
170 sigma0 = as_tensor(F0[j,k]*P0[k,i], (j,i))
171 sigma0_dev = as_tensor(sigma0[i,j] - 1.0/3.0 * sigma0[k,k] * delta[i,j], (i,j))
172 eqStress0 = as_tensor((3.0/2.0 * sigma0_dev[i,j] * sigma0_dev[i,j])**0.5, ())
173 crackStress = project(eqStress0, FunctionSpace(mesh, 'DG', 1))
174
175 penalty = 1E15
176
177 F1 = (P[j,i]*del_u[i].dx(j)-f[i]*del_u[i]) *dx
178 F2 = 0.5*(del_u('+')[i]+del_u('-')[i]) * interfaceP[j,i].dx(j) *dS
179 F3 = -0.5*(P('+')+P('-'))[j,i]*(del_u('+')-del_u('-'))[i]*n('+')[j] *dS
180 F4 = (del_u('+')-del_u('-'))[i]*(u('+')-u('-'))[i]*penalty *dS(0)
181 F5 = - trac[i]*del_u[i] *ds
182
183 Form = F1+F2+F3+F4+F5
184
185 Gain = derivative(Form, u, d_u)
186
187 # =====
188 # ===== DEFINE SOLVER PARAMETER =====
189 # =====
190 #
191 # abstract interface for a Newton solver,
192 # iterate is an object and has its own attributes self.attribute
193 class iterate(NonlinearProblem):
194     def __init__(self, a, L, bc, exter_B, inter_B):
195         NonlinearProblem.__init__(self)
196         self.L = L
197         self.a = a
198         self.bc = bc
199         self.e = exter_B
200         self.i = inter_B
201     def F(self, b, x):
202         assemble(self.L, tensor=b, exterior_facet_domains=self.e, interior_facet_domains=self.i)
203         for condition in self.bc: condition.apply(b, x)
204     def J(self, A, x):
205         assemble(self.a, tensor=A, exterior_facet_domains=self.e, interior_facet_domains=self.i)
206         for condition in self.bc: condition.apply(A)
207
208 solver = NewtonSolver()
209 solver.parameters['linear_solver']='lu' #'mumps' #'cg' conjugated gradients
210 solver.parameters['convergence_criterion'] = 'incremental'
211 solver.parameters['relative_tolerance'] = 1.0e-3
212 solver.parameters['absolute_tolerance'] = 1.0e-6
213 solver.parameters['maximum_iterations'] = 30

```

```

214
215 pfile = File('example_DG_mesh5/nonlin_elastostatics.pvd')
216 sfile = File('example_DG_mesh5/nonlin_elastostatics_stress.pvd')
217 mfile = File('example_DG_mesh5/nonlin_elastostatics_mesh.pvd')
218
219 t = 0.0
220 newCrack = 0.0
221 crackPeakList = []
222
223 while t < 0.50:
224     print ("#####");
225     print ("#");
226     print ("# Solving Time = " + str(t));
227     print ("#");
228     print ("#####");
229     right_u.t = t
230     problem = iterate(Gain, Form, bc, boundaries, facet_domains)
231
232     solver.solve(problem, u.vector())
233
234     # =====
235     # ===== DEFINE OUTPUT =====
236     # =====
237     #
238     u_prj = project(u, VectorFunctionSpace(mesh2, 'CG', 1))
239     pfile << (u_prj, t)
240     u0.assign(u)
241     crackStress = project(eqStress0, FunctionSpace(mesh, 'DG', 1))
242
243     # =====
244     # ===== CRACK GROWING =====
245     # =====
246     #
247     if len(crackPeakList) == 0:
248         tmpValue = 0
249         tmpIndex = 0
250         print("search startpoint")
251         for n__ in vertices(mesh):
252             ## vertex nodes
253             value = crackStress(n__.point())
254             if tmpValue < value:
255                 tmpValue = value
256                 tmpIndex = n__.index()
257
258         if sigma_f < tmpValue:
259             crackPeakList.append(tmpIndex)
260     else :
261         print("search new facet")
262         listNodes = []
263         listValues = []
264         listFacets = []

```

```

265
266     for f__ in facets(mesh):
267         node1ID = f__.entities(0)[0]
268         node2ID = f__.entities(0)[1]
269         testTpm = (mesh.coordinates()[node1ID] + mesh.coordinates()[node2ID] )/2.0
270         tmpStress = crackStress(testTpm)
271
272         if node1ID == crackPeakList[-1] and not (node2ID in crackPeakList):
273             listNodes.append(node2ID)
274             listValues.append(tmpStress)
275             listFacets.append(f__)
276
277         elif node2ID == crackPeakList[-1] and not (node1ID in crackPeakList):
278             listNodes.append(node1ID)
279             listValues.append(tmpStress)
280             listFacets.append(f__)
281
282         if max(listValues) > sigma_f:
283             ID = listValues.index(max(listValues))
284             facet_domains[listFacets[ID]] = 1
285             crackPeakList.append(listNodes[ID])
286             print("newCrack")
287
288     crackStress_plot = project(eqStress0, FunctionSpace(mesh2, 'CG', 1))
289     sfile << (crackStress_plot, t)
290     mfile << (facet_domains, t)
291
292     x = np.arange((notchx_offs)-1e-3, (notchx_offs)+1e-3, xlength/300.)
293     lst = []
294     for x_ in x:
295         lst.append(u((x_, notchy_offs))[0])
296     np.savetxt('./example_DG_mesh5/displ_mag.csv', np.transpose(np.asarray([ x, lst])), \
297         delimiter=",")
298
299     t = t + 0.025

```



## Literatur

- [1] I. Babuska and M. Zlamal. Nonconforming elements in the finite element method with penalty. pages 863–875. SIAM J. Numer. Anal., 1973.
- [2] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible navier-stokes equations. *J. Comput. Phys.*, 131(2):267–279, 1997.
- [3] F. Brezzi, G. Manzini, D. Marini, P. Pietra, and A. Russo. Discontinuous galerkin approximations for elliptic problems. pages 365–378. Numer. Methods Partial Differential Eq., 2000.
- [4] R. C. Cammarata. Surface and stress effects in thin films. Elsevier Science Ltd., 1994.
- [5] B. Cockburn. Discontinuous galerkin methods. *Z. Angew. Math. Mech*, 2003.
- [6] Metallurgical Consulting. [http://www.metalconsult.com/images/thread\\_of\\_a\\_stainless\\_steel\\_stud\\_large.jpg](http://www.metalconsult.com/images/thread_of_a_stainless_steel_stud_large.jpg).
- [7] B. Rivière et al. Discontinuous galerkin finite element methods for linear elasticity and quasistatic linear viscoelasticity. Springer-Verlag, 2003.
- [8] C. W. Mays et al. On surface stress and surface tension. North-Holland Publishing Co., 1968.
- [9] J. Mergheim et al. A hybrid discontinuous galerkin/interface method for the computational modelling of failure. *Commun. Numer. Meth. Engng*, 2004.
- [10] J. S. Vermaak et al. On surface stress and surface tension. North-Holland Publishing Co., 1967.
- [11] R. Shuttleworth et al. The surface tension of solids. *Proc. Phys. Soc.*, 1949.
- [12] Peter Kaufmann, Sebastian Martin, Mario Botsch, and Markus Gross. Flexible simulation of deformable models using discontinuous galerkin fem. pages 105–115. Eurographics Association, 2008.
- [13] R. Radovitzkyv L. Noels. Alternative approaches for the derivation of discontinuous galerkin methods for nonlinear mechanics. *Journal of Applied Mechanics*, 2007.
- [14] W.H. Müller. *Streifzüge durch die Kontinuumstheorie*. Springer, 2011.
- [15] A. M. Stoneham. Measurement of surface tension by lattice parameter changes: theory for faceted microcrystals. *J. Phys. C: Solid State Phys.*, 1977.
- [16] A. TenEyck. Discontinuos galerkin methods for approximating the solution to problems in nonlinear elasticity. 2008.