

Montanuniversität Leoben

**SIMULATION EINER NANOINDENTATION IN DAS IDEALISIERTE
MODELL ("STAGGERED-MODEL" = GESTAFFELT AUFGEBAUTES MODELL)
EINES BIOLOGISCHEN MATERIALS (KNOCHEN, DENTIN, PERLMUT,..)
MIT HILFE DES FEM-PAKETES "ABAQUS"**



erstellt am Department Materialphysik
betreut von
a.o.Univ.Prof. Dr. Ingomar JÄGER

Leoben, Jänner 2007

Danksagung / Widmung

Ich bedanke mich bei allen Personen, die es mir ermöglicht haben, das Studium in Leoben zu absolvieren, allen voran **meiner Familie** und hier insbesondere meinem **Bruder Andreas**, der die finanzielle Grundlage dafür schaffen konnte.

Auch möchte ich mich bei meinem Diplomarbeitbetreuer **Dr. Ingomar Jäger** für die Führung und Unterstützung während der Zeit der Betreuung bedanken. Mein Dank gilt auch **Dr. Thomas Antretter**, der mir bei meinen ABAQUS-Problemen stets mit gutem Rat zur Seite stand und ständig ein offenes Ohr dafür hatte, sowie **Dr. Johannes Meenen** der Firma ABAQUS, der produktive Vorschläge für weitere Simulationen machte.

Kurzdarstellung

Die vorliegende Arbeit beschäftigt sich mit der Simulation einer Nanoindentation in einen Verbundwerkstoff, dessen Aufbau dem idealisierten Modell für biologische Materialien ("staggered model") entspricht. Eine Nanoindentation ist eine Härteeindruckmessung vergleichbar dem Verfahren nach Rockwell jedoch im Bereich von Nanometern meist mit einer dreiseitigen Berkovich-Pyramide zur Bestimmung einer Kraft-Eindringtiefen-Kurve. Mittels solcher Kurven kann der Indentationsmodul eines Materials ermittelt werden, welcher im Falle eines Knochens einen Vergleichs-Materialkennwert zur Beurteilung der Knochenqualität darstellt. Ziel der Arbeit ist es, neben der Untersuchung des Spannungsfeldes unter der Indenterspitze mit Hilfe des FEM-Paketes ABAQUS, die Härte sowie den Indentationsmodul laut Oliver/Pharr zu ermitteln und diesen mit vorliegenden Messergebnissen bzw. Literaturdaten zu vergleichen. Es werden unterschiedliche Verfahrensrouten und Programmadaptierungen verwendet um ein optimales Vergleichsergebnis zu gewährleisten. Wie gezeigt wird, lassen sich mit den zur Verfügung stehenden Ressourcen mittels des Programmpaketes ABAQUS für die gewählten Materialdaten derzeit noch keine aussagekräftigen Simulationsergebnisse erhalten.

Abstract

The present work deals with a nanoindentation simulation of a composite material corresponding to the idealized model for biological materials ("staggered model"). Nanoindentation is a kind of a hardness test comparable to Rockwell but on a nanometer scale and with a three-sided Berkovich pyramid. It is determining the load-displacement-curve. Such curves can be used to determine the Indentation modulus of a material. In case of bone applications this characteristic value is a reference material property to evaluate the bone quality. The FEM package ABAQUS was used to examine the stress field under the Indenter tip. Moreover, the package was utilised to determine the hardness and the Indentation modulus according to Oliver/Pharr and to compare the calculated results with the literature existing data. Different method routes and programme adaptations have been followed to ensure results reliability. As shown in the present work the present resources are not sufficient to get comparable meaningful simulation results by using the program package ABAQUS for the chosen material data.

Inhaltsverzeichnis

Kurzdarstellung	I
Inhaltsverzeichnis	II
1 Einleitung	1
1.1 Forschungsleitendes Interesse	1
1.1.1 Knochenumbau - Osteoblasten, Osteoklasten und Osteozyten	1
1.1.2 Osteoporose	2
1.1.3 Knochenmasse vs. Knochenqualität	4
1.1.4 Aufbau eines Knochens – Komponenten und Bausteine	4
1.1.4.1 Struktur und Eigenschaften des Hydroxyapatits	5
1.1.4.2 Struktur und Eigenschaften des Kollagens	7
1.1.5 Aufbau eines Knochens – hierarchische Struktur	10
1.2 Forschungsleitende Fragestellungen	12
2 Arbeitshypothese	13
2.1 Aufbau (Anordnung und Abmessung) auf Nanometerebene	13
2.2 Das "Staggered Model"	19
2.3 Die Indentationsmodul-Bestimmung nach Oliver/Pharr	22
2.4 Die Vergleichsspannung nach von Mises	27
3 Die Finite Elemente Methode (FEM)	30
3.1 Theoretische Grundlagen	30
3.2 Allgemeines Vorgehen	31
3.3 Diskretisierung	32
3.4 Element-Ansatz	33
3.5 Das Prinzip vom Minimum des Potentials	34
3.6 Implizite vs. explizite Problemlösung	34
3.6.1 Implizite Problemformulierung	36
3.6.2 Explizite Problemformulierung	37
4 Vorgehensweise – Simulationsschritte – Modellentwicklung	38
4.1 Verwendete Hard- und Software	38
4.1.1 Arbeitsplatzrechner	38
4.1.2 Universitäts-Applicationsserver "zalpha1"	38
4.1.3 Universitäts-Applikationsserver "HP®-Cluster"	39
4.1.4 SMMP Shared-Memory-Applikationsserver "SGI®-Altix®350"	39
4.1.5 Die Netzwerkeinrichtung	40

4.1.6 ABAQUS 5.8, 6.3, 6.4, 6.5 und 6.6	40
4.1.7 Fortran F77 und Microsoft Visual C++ 6.0	41
4.1.8 Exceed und Exceed3D (Hummingbird Connectivity V7.1)	42
4.1.9 SSH® Secure Shell™	42
4.1.10 ORIGIN, MS-Excel und MS-Word	43
4.2 Die gewählten Abmessungen und Materialdaten	43
4.3 Modellaufbau	44
4.4 Modellvernetzung	45
4.4.1 Voll (3D) Simulation – Netzgenerierung durch ABAQUS/CAE	47
4.4.2 Reduzierte (2D) Simulation – ABAQUS/Standard	49
4.4.2.1 Netzgenerierung mittels CAE	50
4.4.2.2 Netzgenerierung mittels Programmierung in FORTRAN/C++	56
4.4.3 Reduzierte (2D) Simulation – ABAQUS/Explicit	61
4.4.3.1 Netzgenerierung mittels Programmierung in C++	62
4.4.3.2 Netzgenerierung mittels CAE	63
4.4.4 Reduzierte (2D) Simulation – ABAQUS Standard	64
4.4.4.1 Netzgenerierung mittels Programmierung in C++	64
4.4.4.2 Netzgenerierung mittels CAE	64
5 Die Simulationsergebnisse	64
5.1 Indentation senkrecht auf die Mineralplättchen	64
5.1.1 Geometrie, Netz und Randbedingungen	65
5.1.2 Visualisierung der Simulationsresultate	67
5.1.3 Ermittlung der validierbaren projizierten Indentationsfläche	71
5.1.4 Ermittlung der validierbaren Indenterkraft	74
5.1.5 Ermittlung der "wahren Härte"	76
5.1.6 Auswertung nach Oliver/Pharr – Indentationsmodul	76
5.1.7 Ergebnisdiskussion und Vergleich mit Literaturwerten	79
5.2 Indentation parallel zu den Mineralplättchen	81
5.2.1 Auswertung nach Oliver/Pharr – Indentationsmodul	83
5.2.2 Ergebnisdiskussion und Vergleich mit Literaturwerten	83
6 Zusammenfassung und Schlusswort	84
Abbildungsverzeichnis	85
Literaturverzeichnis	93
Anhang	95

1 Einleitung

1.1 Forschungsleitendes Interesse

Eine der verbreitetsten Skeletterkrankungen ist die Osteoporose (Knochenschwund). Weltweit sind rund 250 Millionen Menschen betroffen, davon allein in Deutschland mehr als 7 Millionen und in Österreich rund 750.000 Menschen, wobei der Frauenanteil hormonell und physiologisch bedingt bei über 80% liegt. So leidet jede vierte Frau über 50 heute an Osteoporose. Zur Bestimmung dieser Krankheit bzw. zur Charakterisierung der Knochenqualität ist es notwendig, einen aussagekräftigen Vergleichswert, also einen "Materialkennwert" des Knochens zu kennen. Einen solchen Wert stellt der E-Modul des Knochens dar, ermittelt nach der Methode von Oliver/Pharr (vgl. **Kapitel 2.3**).

In werkstoffkundiger Hinsicht ist diese Arbeit von Interesse, da das gewählte Modell ebenfalls den idealisierten Aufbau eines Verbundwerkstoffes (Kurzfaserverstärkter Kunststoff,...) darstellt und somit Aussagen über einen solchen abgeleitet werden könnten.

1.1.1 Knochenumbau - Osteoblasten, Osteoklasten und Osteozyten

Die menschlichen Knochen als lebendes Gewebe befinden sich in einem kontinuierlichen, permanenten Umbauprozess und werden dabei fortwährend erneuert. Die Geschlechtshormone (Östrogen, Testosteron) steuern neben anderen Funktionen auch diesen Knochenumbau. Dieser erfolgt, wie in **Abbildung 1** zu sehen, durch Osteoklasten (die knochenfressenden Zellen) und Osteoblasten (die knochenaufbauenden Zellen), die im Team zusammenarbeiten. Die Osteoklasten tragen dabei den alten Knochen ab, um Platz für neuen Knochen zu schaffen. Anschließend wird durch die Osteoblasten, welche Knochengewebe bilden, das später mineralisiert (calcifiziert und somit gehärtet) wird, neuer Knochen gebildet. Bei gesunden Menschen sind Auf- und Abbautätigkeit des Knochens so aufeinander abgestimmt, dass die Knochenmasse erhalten bleibt. Hormone regulieren unter anderem die Aufnahme von Calcium in das Knochengewebe und bremsen den Knochenabbau. [1] Werden Osteoblasten oder Osteoklasten während der Knochenneubildung "eingeschlossen", werden sie zu Osteozyten, die mittels ihrer, in den Knochen "eingewachsenen" flüssigkeitsgefüllten Kanälen die Spannungssituation innerhalb des Knochens wahrnehmen, den jeweils optimalen Knochenumbauplan "erstellen" und an die Osteoklasten und -blasten weiterleiten. So wird sichergestellt, dass lange nicht benötigte Knochenmasse (Aufenthalt im Weltall, Bettlägerigkeit,...) abgebaut bzw. bei erhöhten Spannungen (Gewichtszunahme,

erhöhte sportliche Aktivitäten,...) zusätzliche Knochenmasse erzeugt wird und zwar dort, wo diese überflüssig ist bzw. benötigt wird. [2]

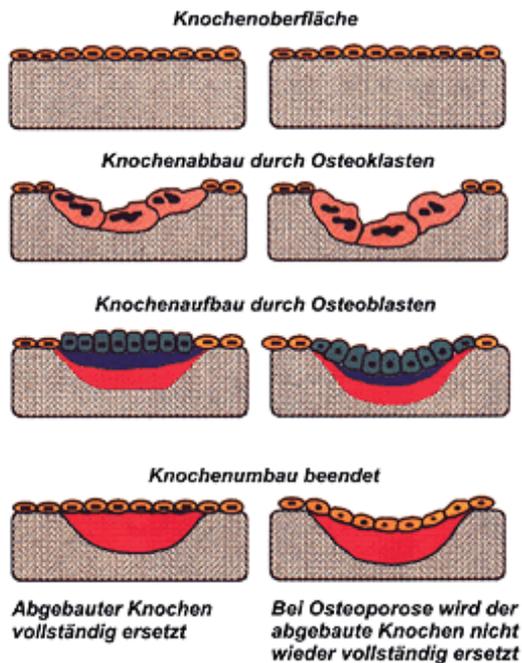


Abbildung 1: Schematische Darstellung des Knochenaufbaues und Knochenabbaues durch die Osteoklasten und Osteoblasten ohne (links) und mit (rechts) Osteoporose-Erkrankung [<http://www.osteoporose.com>].

Sowohl Knochenaufbau als auch Knochenabbau können bei lebenden Menschen mittels der Quote des Knochenumbaus gemessen werden. Dieser Knochenumbau lässt sich durch Blut- oder Urintests bestimmen, welche die entsprechenden Knochenmarker (z.B. Gewebeproteine und von den Osteoklasten bzw. Osteoblasten abgesonderten Enzymen) anzeigen. Es gibt zwei Arten von Knochenmarkern: Resorptionsmarker (diese sind ein Zeichen für Knochenabbau) und Formationsmarker (diese sind bei Knochenaufbau zu finden).

Darüber hinaus gibt es Messmethoden um die Knochendichte bzw. den E-Modul des Knochens zu bestimmen. Die vorliegende Arbeit beschäftigt sich mit der Simulation der E-Modul Bestimmung des Knochens.

1.1.2 Osteoporose

Übersetzt man Osteoporose wörtlich, erhält man das Problem mit seinem Namen, nämlich "poröser Knochen". Die Osteoporose ist durch eine erniedrigte Knochenmasse und Zerstörung der Mikroarchitektur des Knochengewebes (poröse Knochenstruktur) mit nachfolgend erhöhtem Knochenbruchrisiko charakterisiert, welches vorwiegend die Wirbelsäule, den Oberschenkelhals und den Unterarm bzw. das Handgelenk betrifft.

Etwa 95 Prozent aller betroffenen Patienten leiden an einer "primären Osteoporose" für die keine direkte Ursache auszumachen ist. Sie steht in engem Zusammenhang mit dem Alter und

mit dem Hormon- und Calciumstoffwechsel. Zierliche Menschen scheinen häufiger zu erkranken, denn sie haben eine geringere Ausgangsknochenmasse und das Skelett reagiert stärker auf Belastung.

Bei den restlichen Patienten können verschiedene Krankheiten die Entstehung einer Osteoporose begünstigen. Man spricht dann von einer "sekundären Osteoporose", ausgelöst durch z.B. Störung des Cortisonstoffwechsels, Geschlechtshormonmangel beim Mann, Störung des Calciumstoffwechsels oder Schilddrüsenüberfunktion. [1] Welche Auswirkung der Knochenverlust auf die innere Struktur hat, verdeutlicht **Abbildung 2**.

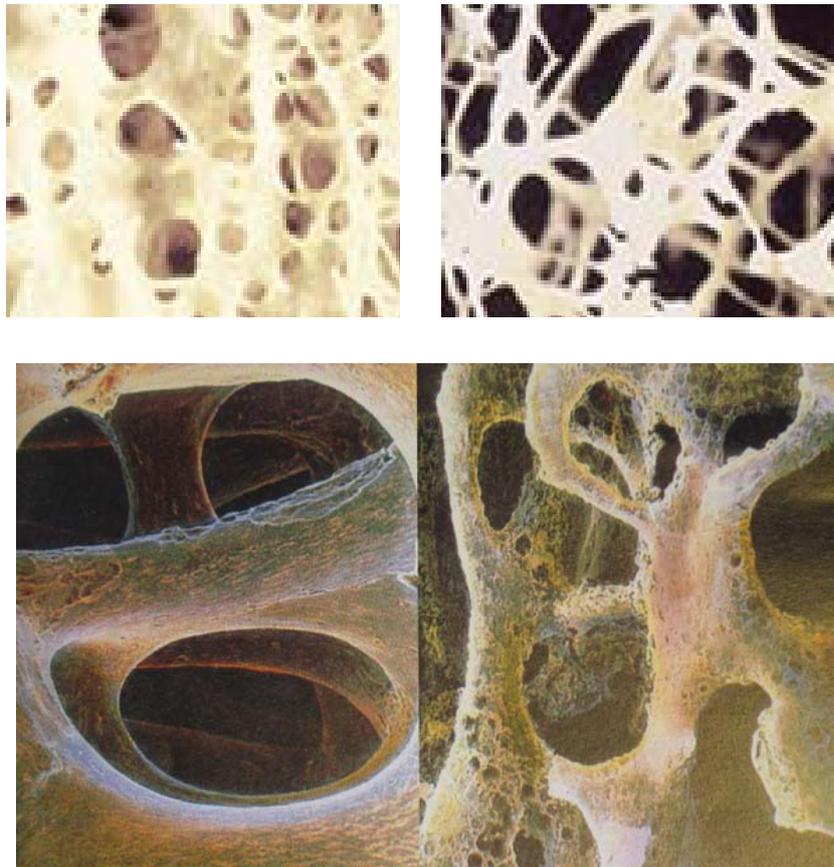


Abbildung 2: Aufnahmen von gesunden (links) bzw. von erkrankten (rechts) Knochentrabekelnetzwerkes (entspricht Ebene 4 der in **Kapitel 1.1.5** erwähnten Einteilung) [<http://www.osteoporose.com>, APOTHECUM Herbst 2004].

Ergänzend sei hier angeführt, dass die Osteoporose bisher vor allem mit Medikamenten behandelt wurde, die den Knochenabbau hemmen. Das sind Resorptionshemmer wie die Bisphosphonate und Raloxifen. Mit neueren Medikamenten z.B. Parathormon wird hingegen der Knochenaufbau gefördert. Die Osteoblasten werden aktiviert. Der Knochenaufbau wird

dadurch mehr gesteigert als die Knochenabbau-Tätigkeit. Parathormon (PTH) wird in den Zellen der Nebenschilddrüse produziert und wirkt auf verschiedene Weise. Ziel der Wirkung ist in jedem Fall eine Anhebung des Serum-Calcium-Spiegels.

1.1.3 Knochenmasse vs. Knochenqualität

Frakturresistenter Knochen bedeutet starker Knochen und Knochenstärke ist abhängig von zwei Hauptfaktoren: der Knochenmasse und der Knochenqualität. Während die Messung der Knochenmasse zur Bestimmung des künftigen Frakturrisikos hilfreich sein kann, führt die alleinige Erhöhung der Knochenmasse durch eine Behandlung nicht unbedingt zum Schutz vor Frakturen. Wichtig zur Bestimmung der Knochenstärke und letztendlich zur Bestimmung der Frakturresistenz ist die Knochenqualität. Eine Vielzahl von Faktoren tragen zur Knochenqualität bei, vor allem die Mikroarchitektur (die mikroskopische Struktur des trabekulären Knochens), der Knochenumbau (ausgeglichenes Abbauen, Erneuern und mikroskopisches Reparieren des Knochengewebes) und die Materialeigenschaften des Knochengewebes (Mineralisierung und die Kombination der Kristallinität mit der Kollagenmatrix). Neueste Studien zeigten, dass einige Therapieformen alle Faktoren der Knochenqualität beeinflussen, die Knochenstärke bewahren und Frakturen reduzieren. Vor allem die Mikroarchitektur (Knochenstruktur mikroskopisch betrachtet) spielt eine wichtige Rolle hinsichtlich der allgemeinen Knochenstärke. Bei Patienten mit Osteoporose führt der beschleunigte Umbau des Knochens zum kompletten Abbau oder zu einer allgemeinen Schwächung der Trabekel bzw. der dreidimensionalen Knochenmatrix. Schließlich werden die vertikalen und horizontalen trabekulären Verbindungen unterbrochen, und es entstehen freie Enden. Diese in ihrer Kontinuität unterbrochenen trabekulären Verbindungen reduzieren die Knochenstabilität und erhöhen somit das Frakturrisiko.

1.1.4 Aufbau eines Knochens – Komponenten und Bausteine

Der Knochen ist ein lebendes Gewebe, das aus einem Fasergerüst (Kollagenfibrillen, der organischen Komponente des Knochens) besteht, welches mittels Calcium, Phosphor und anderen Mineralien unter Bildung von nichtorganischen Calcium-Phosphat-Kristallen (Hydroxyapatit z.B. $\text{Ca}_{10}(\text{PO}_4)_6(\text{OH})_2$) teilweise mineralisiert und so die Stabilität des Knochens gewährleistet.

1.1.4.1 Struktur und Eigenschaften des Hydroxyapatits

Der Name Hydroxylapatit (Hydroxyapatit, Apatite, ICSD 203027, PDF 9-432, Dana No: 41.8.1.3, Strunz No: VII/B.39-30) leitet sich einerseits von der Hydroxylgruppe (-OH) und andererseits vom griechischen "apadao" ("Ich bin irreführend") ab.

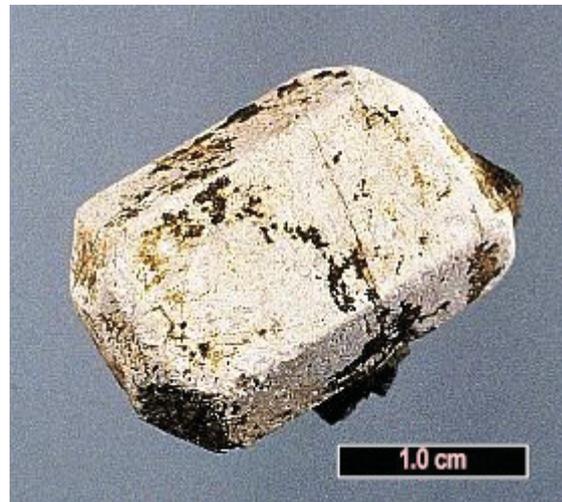


Abbildung 3: Ein weißer Hydroxyapatitkristall [<http://webmineral.com>].

Durch seine Strukturformel $\text{Ca}_{10}(\text{PO}_4)_6(\text{OH})_2$ hat es eine Molmasse von 502.31g/mol, was sich auf 55.82 % CaO, 42.39 % P_2O_5 und 1.79 % H_2O aufteilt. Es besitzt, wie sich anhand von **Abbildung 3** erahnen lässt und wie man in **Abbildung 4** sieht, eine hexagonal-dipyramidale Kristallstruktur (H-M Symbol: 6/m; Space Group: P 63/m) mit einem Achsenverhältnis von $a = a_1 = a_2 = a_3$ und $a : c = 1 : 0.72998$. Somit ergibt sich eine Dichte von 3.16g/cm^3 . Bei der Röntgendiffraktometrie (Powder X-Ray Diffraction (XRD)) kann folgende Hauptintensität und daraus die interplanaren Abstände [\AA] bei den auftretenden relativen Intensitäten (I/I_0) ermittelt werden:

$$D_1(I_0) = 2.814 (1) ; D_2(I/I_0) = 2.72 (0.6) ; D_3(I/I_0) = 2.778 (0.6)$$

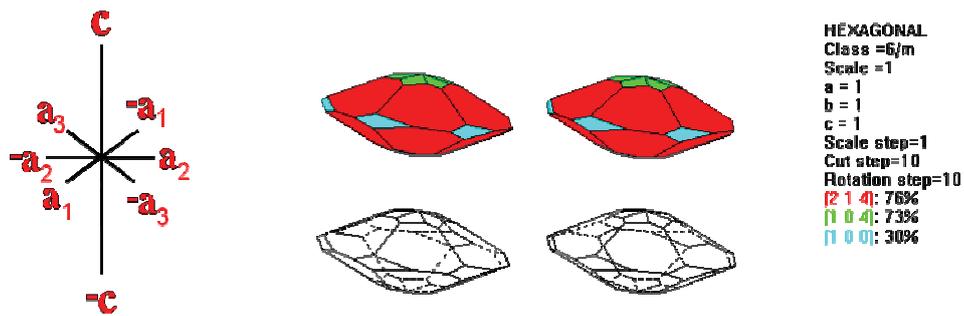


Abbildung 4: hexagonal-dipyramidale Kristallstruktur des Hydroxyapatits
[<http://webmineral.com>].

Hier sei angemerkt, dass die Funktionsgrundlagenidee von Fluor-Zahnpasten darauf beruht, dass die Hydroxylgruppe (-OH) aus den obersten Hydroxyapatitplatten durch Fluor (F) ersetzt wird und der chemisch stabilere (säurebeständigere) Fluorapatit $\text{Ca}_5(\text{PO}_4)_3\text{F}$ entsteht.

Die **Abbildung 5** zeigt die derzeit einzigen veröffentlichten AFM-Aufnahmen von physiologischen Hydroxyapatitkristallen, deren Größe etwa 40nm mal 20nm beträgt.

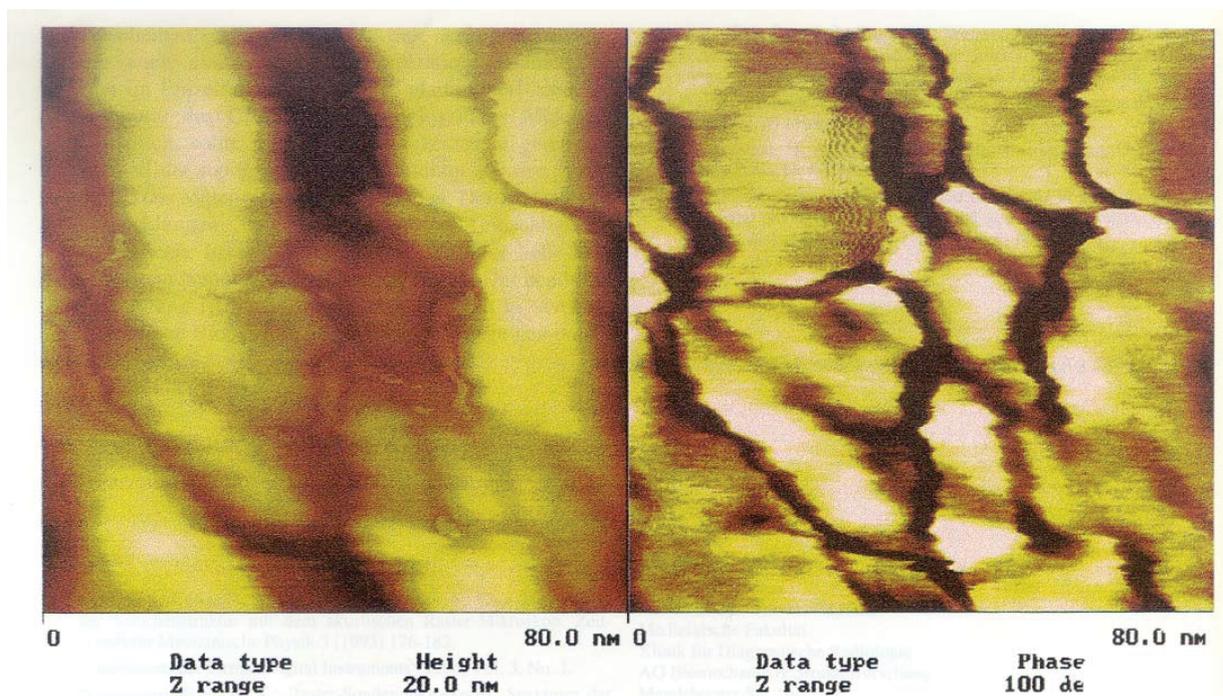


Abbildung 5: Raster-Sondenmikroskopaufnahme (AFM) einiger Apatitkristalle in der Kollagenfasermatrix. Die linke Aufnahme wurde mit dem "kontaktlosen Modus" (20nm über der Oberfläche) erstellt, wobei das rechte Bild eine Aufnahme mit dem "Tapping Modus" zeigt. Die Apatitkristalle lassen sich dabei als helle Flächen und die Kollagenfibrillen als dunkle Areale erkennen [18].

1.1.4.2 Struktur und Eigenschaften des Kollagens

Kollagene sind eine Proteinfamilie der extrazellulären Matrix, die von Bindegewebszellen gebildet werden. Sie stellen eine heterogene Gruppe von Proteinen dar, die etwa ein Viertel der Gesamtproteinmenge im menschlichen Organismus ausmachen. Kollagen ist der wichtigste Faserbestandteil von Haut, Knochen, Sehnen, Knorpel, Blutgefäßen und Zähnen. Kollagenmoleküle weisen einen hohen Anteil an Prolin und Glycin (meist an dritter Stelle) auf, daneben eine Reihe hydroxylierter Aminosäuren wie Hydroxyprolin und Hydroxylysin, die eine Quervernetzung der Proteine und die Ausbildung eines stabilen Kollagenmatrix ermöglichen.

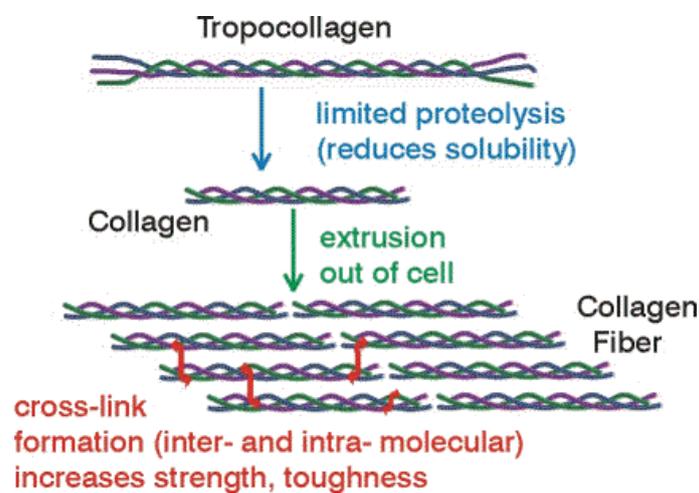


Abbildung 6: Bildung von Kollagen Fibrillen (Collagen Fiber) aus Kollagenmolekülen (Collagen) und dessen Ausgangsbaustein innerhalb der Zelle (Tropokollagen) [<http://labs.ansci.uiuc.edu>].

Die Moleküle der vier häufigsten Kollagentypen (I > II > III > V, Einteilung siehe **Abbildung 8**) sind etwa 300nm lang und 1.5nm dick. Sie neigen aufgrund ihrer Aminosäuresequenzen unter physiologischen Bedingungen zu lateraler Assoziation mit einer Versetzungsperiode von 67nm zu den direkten Nachbarmolekülen. **Abbildung 6** zeigt die Bildung dieser koaxialen Aggregate, die Fibrillen genannt werden. Sie können durch Querbrücken stabilisiert werden und sind somit vielseitige Baustoffe für die mechanische Zugbelastbarkeit und räumliche Organisation von Geweben.

Wie in **Abbildung 7** ersichtlich, zeichnet sich ihre Konformation durch einen sehr hohen triplehelikalen Anteil aus. Kollagenfasern (Fibrillen) bestehen dabei aus drei α -helikalen Kollagenmolekülen, von denen jeweils drei Moleküle sich in Form einer Superhelix umeinander winden und auf diese Weise Fasern ausbilden.

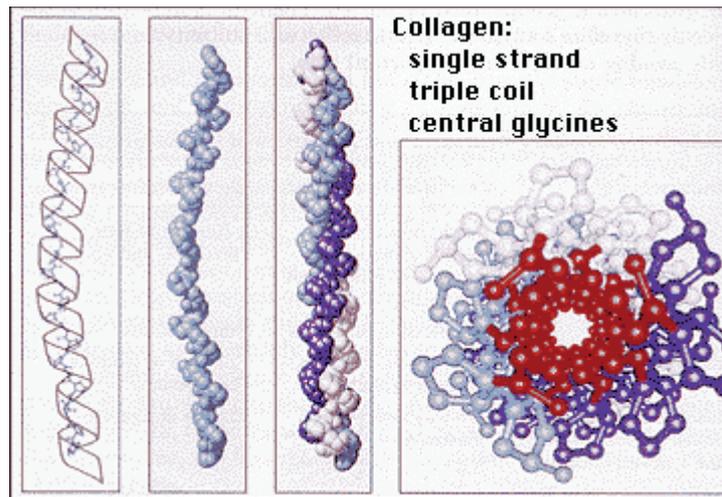


Abbildung 7: Der Aufbau eines Kollagenmoleküls. Dabei winden sich drei α -helikale Kollagenmoleküle in Form einer Superhelix umeinander [www.mad-cow.org] [de.wikipedia.org].

Kollagen findet sich im Organismus vor allem in Bindegewebe, wo es je nach Beschaffenheit den Großteil der extrazellulären Matrix ausmachen kann. Die unterschiedliche Beschaffenheit der verschiedenen Kollagenklassen liegt einer Vielzahl von Eigenschaften des Binde- und Stützgewebes zugrunde. Die Reißfestigkeit von Bändern und Sehnen, die Flexibilität von Knochen oder die Druckresistenz der Gelenkknorpel ist größtenteils durch das im Gewebe vorherrschende Kollagen bedingt. Die Synthese von Kollagen ist von Ascorbinsäure (Vitamin C) abhängig. Ein Mangel daran ruft das Krankheitsbild des Skorbut hervor. Eine Vielzahl genetischer Erkrankungen beruhen auf Defekten des Kollagens oder damit assoziierter Proteine (beispielsweise Ehlers-Danlos-Syndrom, Osteogenesis imperfecta, Stickler-Syndrom, Alport-Syndrom). Bisher konnten 25 Kollagenpolypeptide beschrieben werden, die über 19 unterschiedliche Kollagentypen (siehe **Abbildung 8**), deren Funktion jedoch teilweise noch ungeklärt ist, in der extrazellulären Matrix aufbauen. Aufgrund der unterschiedlichen Kollagentypen und Strukturen ist eine Angabe von allgemein gültigen Eigenschaftswerten nicht möglich.

- Kollagen I (faserbildend): Haut, Sehnen, Knochen, Dentin, Faserknorpel
- Kollagen II (faserbildend): Hyaliner Knorpel, Faserknorpel, Elastischer Knorpel, Glaskörper
- Kollagen III (faserbildend): Haut, Skelettmuskulatur, Blutgefäße
- Kollagen IV (netzförmig): Basallamina
- Kollagen V (faserbildend): Fetale Gewebe, Plazenta, interstitielles Bindegewebe
- Kollagen VI (faserassoziiert, kugelförmig): Bindegewebe
- Kollagen VII (Ankerfibrillen): Verankerung von Epithelien auf Basalmembran und Stroma
- Kollagen VIII (netzförmig): Descemet-Membran
- Kollagen IX (faserassoziiert): Knorpel, Glaskörper
- Kollagen X (netzförmig): Wachstumszone des Knorpels
- Kollagen XI (faserbildend): Knorpel
- Kollagen XII (faserassoziiert): embryonale Haut, embryonale Sehnen
- Kollagen XIII: unbekannt
- Kollagen XIV: unbekannt
- Kollagen XV: unbekannt
- Kollagen XVI: unbekannt
- Kollagen XVII (transmembranär): Assoziation mit Hemidesmosomen von Plattenepithelzellen der Epidermis
- Kollagen XVIII: unbekannt
- Kollagen XIX: (faserassoziiert): fetale Haut, fetale Sehnen

Abbildung 8: Unterteilung und Auftreten der 19 unterschiedlichen Kollagentypen [2].

Wirtschaftliche Bedeutung hatte das tierische Kollagen in der Vergangenheit, als es durch Säure oder Hitze denaturiert, zu "Knochenleim" verarbeitet wurde. Heute werden aus Kollagenfasern resorbierbare Nahtmaterialien und Hauttransplantate synthetisiert. In der kosmetischen Chirurgie wird Kollagen als Injektionsmaterial zur Formung äußerer Strukturen des Körpers (z.B. Lippen) verwendet. Die **Abbildung 9** zeigt REM-Aufnahmen von Kollagenfibrillen in physiologischem Zusammenhang.

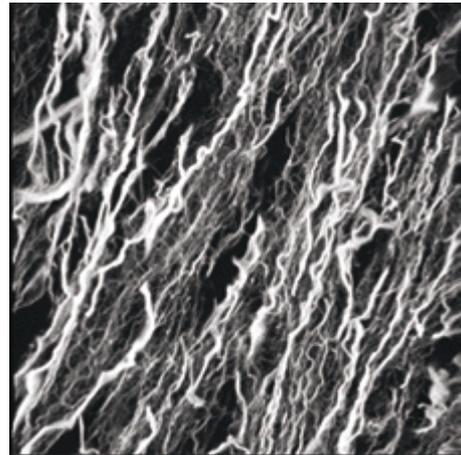
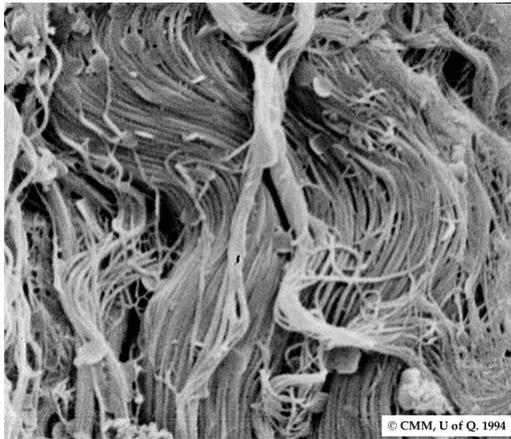


Abbildung 9: REM-Bild einzelner Kollagenfasern links [www.wellesley.edu] sowie eines Mäuseherzens rechts [www.sdsc.edu].

1.1.5 Aufbau eines Knochens – hierarchische Struktur

Die Struktur des Knochens teilt sich in kortikalen, osteonalen ("festen") und trabekulären, spongiösen ("porösen") Knochen. Der kortikale Knochen ist die äußere Schicht - eine starke dichte Struktur aus Gewebefasern. Innerhalb dieser äußeren Schicht befindet sich, wie man in **Abbildung 10** erkennt, eine schwammartige (poröse, trabekuläre, spongiöse) Knochenstruktur (Wirbelkörper, Hüftknochen,...). Dieser besteht aus verbundenen Platten und Stäben, die ein inneres Gerüst bilden. Die Anzahl und Größe der Trabekulae und wie diese miteinander verbunden sind macht die strukturell bedingte Tragkraft des Knochengewebes aus.

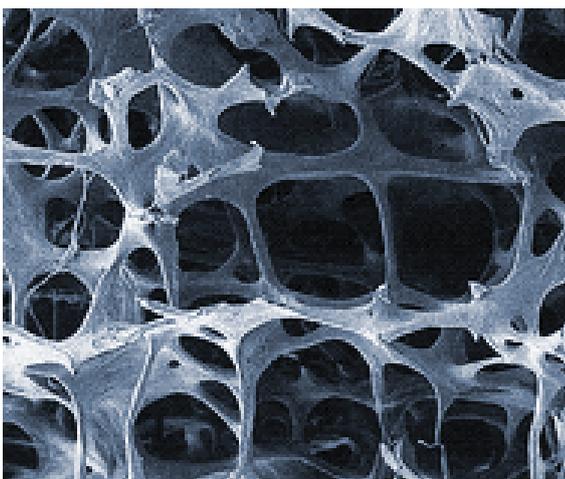


Abbildung 10: REM-Bild der Mikroarchitektur der tragenden Struktur (Trabekelstruktur) eines Knochens [Prof. Dr. Hans-Jürgen Pesch, Erlangen].

Zur weiteren Charakterisierung des Knochenaufbaus ist die folgende Unterteilung nach der hierarchischen Struktur weitestgehend akzeptiert:

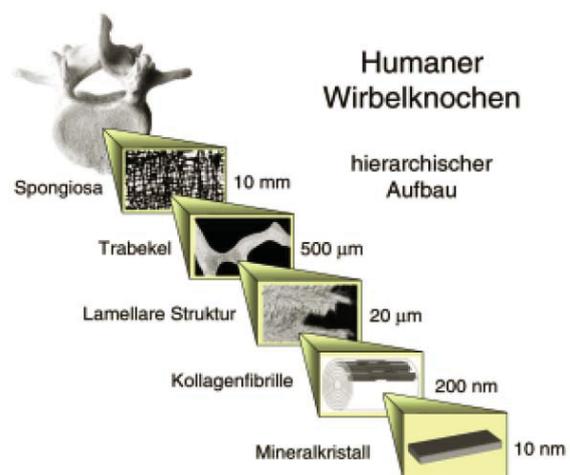


Abbildung 11: Schematische Darstellung des Aufbaus einer hierarchischen Struktur am Beispiel eines spongiösen Menschenknochens [21].

Von hierarchischen Strukturen spricht man, wie man anhand von **Abbildung 11** erkennt, wenn bei der mikroskopischen Vergrößerung eines Materials immer neue Unterstrukturen erkennbar werden. Dieser Effekt wird vor allem bei biologischen, gewachsenen Materialien ("Verbundwerkstoffen") beobachtet.

Üblicherweise wird bei der Untersuchung von Knochen über die, in **Abbildung 12** angeführten, 6 hierarchischen Ebenen gesprochen.

- Ebene 5 : Äußere Form
- Ebene 4 : Innere Architektur (osteonal und trabekulär)
- Ebene 3 : Mineralverteilung
- Ebene 2 : Lamellare Organisation der Knochenmatrix
- Ebene 1 : Mineralisierte Kollagenfibrille
- Ebene 0 : Kollagenmolekül/Mineralkristall Komposite

Abbildung 12: Die Unterteilung in die 6 hierarchischen Ebenen des Knochens [2].

Während die Ebene 5 noch die Größenordnung von einigen mm bis dm hat, weisen die Kollagenfibrillen (Ebene 1) nur noch einen Durchmesser von einigen 100nm auf und die Ebene 0 beschränkt sich auf einige nm. Somit wird verdeutlicht, dass die hierarchischen Ebenen mehr als 8 Größenordnungen (von Nanometer [nm] bis Dezimeter [dm]) überspannen [4].

Wie bereits der Name des in dieser Arbeit untersuchten Verfahrens verrät, beschäftigt sich dieses mit der Ebene 0, also mit Vorgängen auf der Größenskala von Nanometern [nm].

1.2 Forschungsleitende Fragestellungen

Um eine Aussage über die "Materialqualität" eines Knochens treffen und eine Erkrankung (z.B. Osteoporose) somit eingehender bzw. systematischer untersuchen zu können, hat es sich durchgesetzt, mittels einer Nanoindentation die Härte sowie den Elastizitätsmodul nach z.B. Oliver/Pharr zu ermitteln und diese als "Materialkennwerte" zu nutzen.

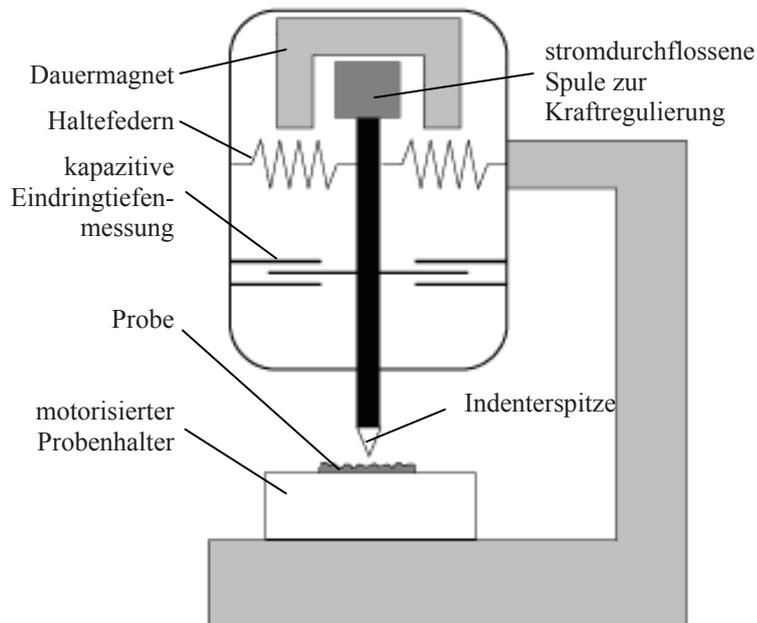


Abbildung 13: Schematische Darstellung des Aufbaus eines Nanoindenters

[www.nanoindentation.cornell.edu].

Obwohl heutzutage viele solcher Messungen durchgeführt werden, gibt es derzeit keinerlei Untersuchungsergebnisse, was dabei wirklich gemessen wird bzw. welche Vorgänge sich dabei im Knochen unterhalb der Indenterspitze abspielen. Da wir uns in der Größenordnung [nm] bewegen, gibt es derzeit nur unzureichende Untersuchungsverfahren um diese Frage zu klären. So wird in der vorliegenden Arbeit versucht, mittels einer FEM - Simulation eine Vorhersage über das Verhalten bzw. die Spannungen innerhalb des Materials zu treffen.

Besonders folgende Fragestellungen werden behandelt:

- Spannungen, Dehnungen sowohl makroskopisch als auch innerhalb der Komponenten. - z.B. Spannungsspitzen in Plättchen (=> Bruch).
- Indenttiefe für den Übergang vom rein elastischen Indent zur plastischen Verformung.
- Größe und Lage der projizierten Indentfläche in Abhängigkeit von der Indentkraft. => Härte.
- "Contact stiffness" S , d.h. dF/du bei maximaler Eindringtiefe, daraus mit Hilfe der Oliver/Pharr-Auswertung des Indentationsmoduls.

2 Arbeitshypothese

2.1 Aufbau (Anordnung und Abmessung) auf Nanometerebene

Wie bereits in **Kapitel 1.1.4.2** erläutert, haben die Moleküle der vier häufigsten Kollagentypen eine Länge von etwa 300nm und eine Dicke von etwa 1.1nm - 1.6nm. Unter physiologischen Bedingungen neigen diese zu lateraler Assoziation mit einer Versetzungsperiode von 67nm. Diese koaxialen Aggregate nennt man Fibrillen. Mineralisiert eine solche Kollagenmatrix, gibt es etliche Möglichkeiten, wie die Hydroxyapatitplatten eingelagert werden können. Dies soll in **Abbildung 14**, ausgehend von der allgemein anerkannten, parallelen Hodge-Petruska Anordnung (a) des unmineralisierten Kollagens, angedeutet werden.

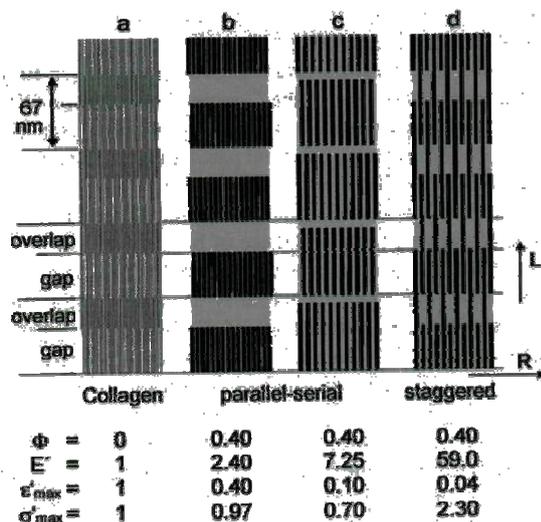


Abbildung 14: Hodge-Petruska Schema für unmineralisiertes Kollagen (a) sowie mögliche parallel-serielle (b-c) bzw. staggered (d) Mineralplattenanordnungen mit zugehörigen, auf reines Kollagen normierten Materialkennwerten [5].

Da die Länge der Moleküle kein ganzzahliges Vielfaches der Periodizität darstellt, kommt es zu Überlappungen (overlap) und Lücken (gap), wobei der Anteil an Bereichen mit Lücken in Längsrichtung, wie man aus **Abbildung 15** erkennen kann, bei rund 60% liegt.

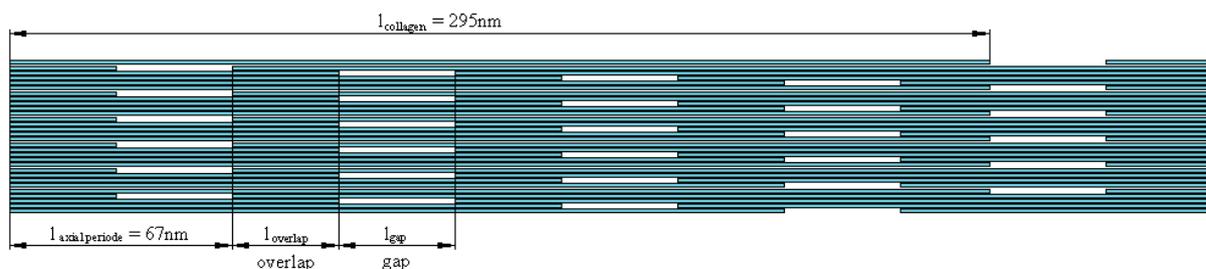


Abbildung 15: Detailliertere Skizze des Hodge-Petruska Schemas für eine unmineralisierte Kollagenfibrille.

Wenn man berücksichtigt, dass die Kollagenmoleküle nicht ganz 300nm lang sind, ist ein Volumenanteil von $f_{gap} = 0.6$ wie dargestellt durchaus korrekt. Die overlap-Bereich-Länge errechnet sich somit, wie ersichtlich ist aus $295 - 4 \cdot 67 = 27\text{nm}$ ($27/67 = 0.40 = 40\%$) damit bleiben für den gap-Bereich eine Länge von $67 - 27 = 40\text{nm}$ ($40/67 = 0.60 = 60\%$). In den gap-Bereichen liegt der Kollagenanteil, wie man ebenfalls sehen kann, rund 20% unter dem der overlap-Bereiche (Die "Periodizität" liegt bei 5 "Lagen", wobei in jedem gap-Bereich eine Lage pro Periode fehlt, also $1/5=20\%$).

Darüber hinaus ergaben Messungen, wie in **Abbildung 16** angeführt, dass Kollagenmoleküle einen mittleren Abstand (senkrecht zur Fibrillenorientierung) von ca. 1.1nm (völlig trockenes Kollagen, $d_{dry} = d_{Kollagen}$), 1.25nm (Schienbeinknochen (mineralisiert), d_{bone}) bzw. 1.55nm (feuchtes Kollagen, d_{wet}) aufweisen. Die Erhöhung des Abstandes ergibt sich aufgrund von Wasser- bzw. Mineraleinlagerungen.

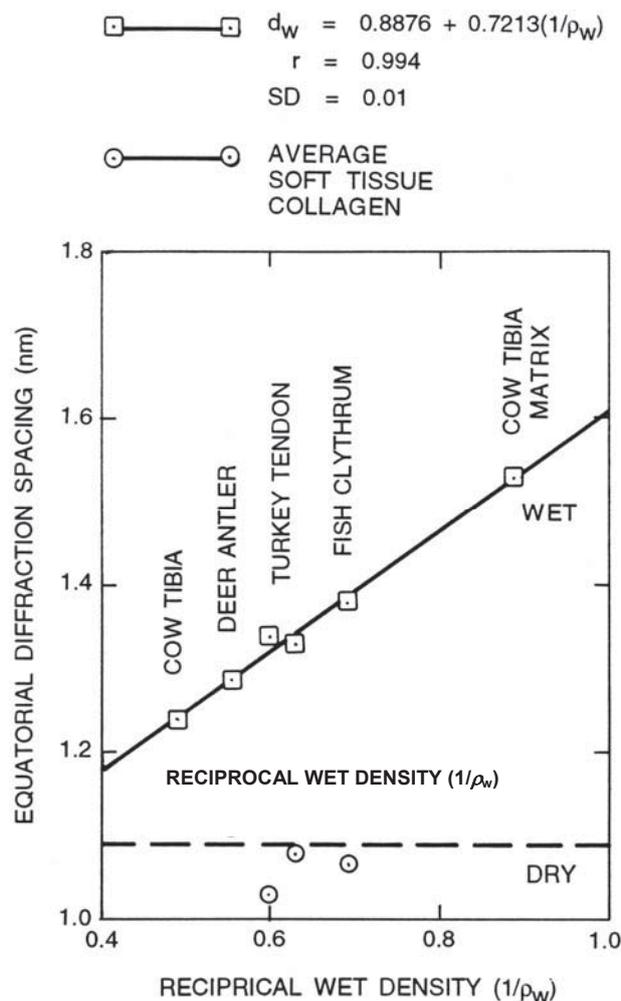
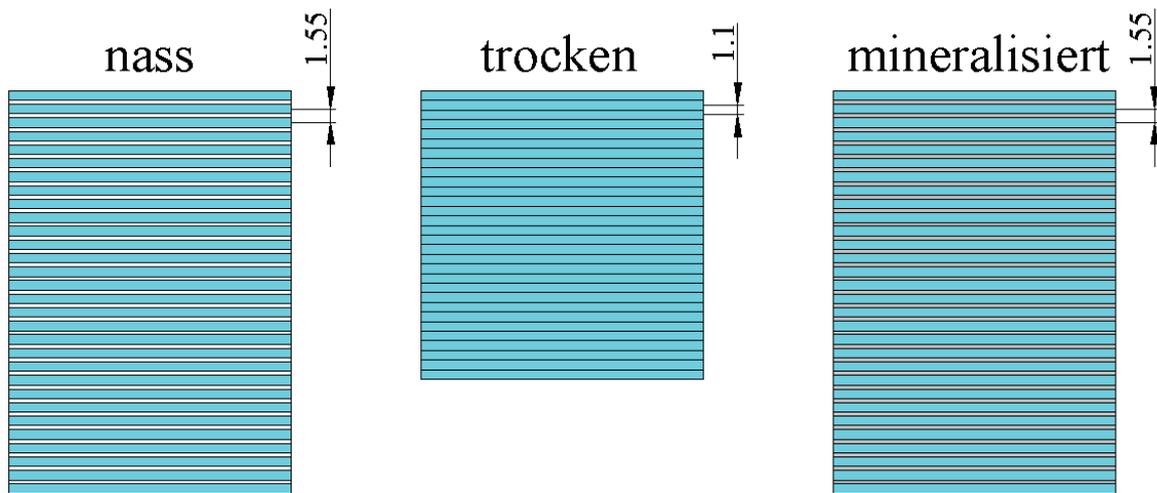
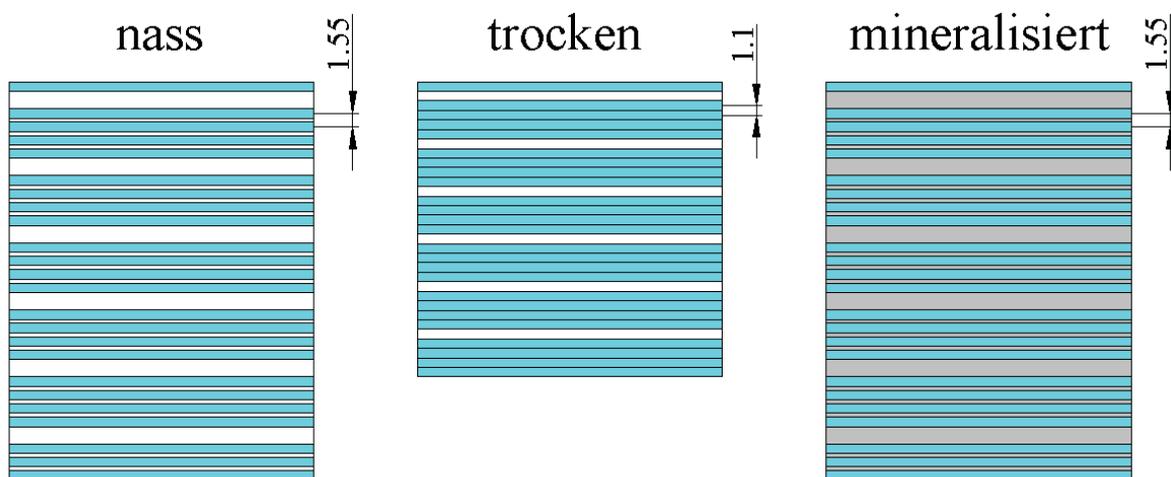


Abbildung 16: Kollagenmolekülabstand über reziproker, auf den nassen Zustand normierter Dichte unterschiedlicher organischer Materialien (Gewebe). Die oberen Punkte beziehen sich auf nasse (feuchte) Proben während die unteren Messpunkte mit trockenen Proben durchgeführt wurden. Die Messpunkte der trockenen Proben streuen zwar mehr, haben allerdings für alle Gewebe etwa den gleichen Wert von 1.1nm. Zum Vergleich kamen Schienbeinknochen einer Kuh (cow tibia), Hirschgeweihstangen (deer antler), Truthahnsehnen (Turkey tendon), Fischgräten (fish clythrum) und Schienbeinknochenmatrix einer Kuh (cow tibia matrix) [Lees et al., 1984].



overlap-Bereich

Abbildung 17: Kollagenfibrillenordnung ($n = 30$) im overlap-Bereich in nassem (wet), trockenem (dry) und 100% mineralisiertem (mineralized) Zustand. Gut zu erkennen ist die Reduktion des Kollagenmolekülabstandes bei der Trocknung auf 1.1 nm (dies entspricht dem Kollagenfibrillendurchmesser). Im ideal vollkommen mineralisierten Zustand sind alle im feuchten Zustand mit Wasser gefüllten Abstände nun mit Mineral gefüllt und daher ist der Kollagenmolekülabstand mit 1.55 nm gleich dem nassen Zustand.



gap-Bereich

Abbildung 18: Kollagenfibrillenordnung ($n = 30$) im gap-Bereich in nassem (wet), trockenem (dry) und 100% mineralisiertem (mineralized) Zustand. Abweichend von **Abbildung 17** ist nun statt jeder fünften Kollagenfibrille aufgrund der Periodizität ebenfalls Hydroxyapatit vorhanden, d.h. der Mineralisierungsgrad in den gap-Bereichen ist höher.

Es wird allgemein angenommen, dass sich in diesen Freiräumen im unmineralisierten Zustand Wasser befindet, welches mit zunehmender Mineralisierung, wie in **Abbildung 17** bzw. **Abbildung 18** zu sehen, verdrängt (entsprechend mit zunehmender Trocknung entzogen) wird. Die "Keimbildung" für die zunehmende Mineralisierung erfolgt aus energetischen Gründen vorwiegend in den großen Freiräumen der gap-Bereiche. Bei voranschreitender Mineralisierung wachsen die Mineralplatten dann auch in die Freiräume zwischen die Kollagenmoleküle in die overlap-Regionen. Ausgehend von diesen Erkenntnissen über die Dimension und Lage der Kollagenlücken ergibt sich nun für den maximal erreichbaren Mineralisierungsgrad (sämtliches Wasser durch Mineral ersetzt):

$$V_{gesamt,overlap} = [n \cdot d_{Collagen} + (n-1) \cdot d_{H_2O}] \cdot l_{overlap} \cdot b = (n \cdot d_{wet} - d_{Collagen}) \cdot l_{overlap} \cdot b \quad [1]$$

$$V_{gesamt,gap} = [n \cdot d_{Collagen} + (n-1) \cdot d_{H_2O}] \cdot l_{gap} \cdot b = (n \cdot d_{wet} - d_{Collagen}) \cdot l_{gap} \cdot b \quad [2]$$

Betrachtet man eine hohe Anzahl von Schichten ($n \hat{=}$) kann man die Vereinfachung $n = n-1 = n+1$ treffen bzw. einzelne d_{MP} vernachlässigen. Diese Konvention kann auch als zulässig betrachtet werden, da die genaue Mineralplättchenanzahl bzw. die Tatsache ob jede Schicht mit Mineralplättchen in Dickenrichtung beendet wird, nicht für jede Probe mit Sicherheit festgestellt werden kann. Daher ergibt sich für das Gesamtvolumen in den jeweiligen Bereichen aus Gleichung [1] und [2] vereinfacht:

$$V_{gesamt,overlap} = n \cdot d_{wet} \cdot l_{overlap} \cdot b \quad [3]$$

$$V_{gesamt,gap} = n \cdot d_{wet} \cdot l_{gap} \cdot b \quad [4]$$

Der bei Trocknung auftretende Dickenverlust, also der Freiraum des Wassers kann wie folgt angegeben werden:

$$\Delta d = d_{wet} - d_{dry} \quad [5]$$

Wenn das gesamte durch das Wasser ausgefüllte Volumen, also das bei Trocknung verloren gehende Volumen, durch Hydroxyapatit aufgefüllt werden würde, ergäbe sich in den jeweiligen Bereichen mit den obigen Vereinfachungen:

$$V_{\text{Mineral,overlap,max}} = V_{\text{Water,overlap}} = \Delta V_{\text{overlap}} = n \cdot \Delta d \cdot l_{\text{overlap}} \cdot b \quad [6]$$

$$V_{\text{Mineral,gap,max}} = V_{\text{Water,gap}} = \Delta V_{\text{gap}} + V_{\text{gap}} = n \cdot \Delta d \cdot l_{\text{gap}} \cdot b + \left(\frac{1}{5} \cdot n\right) \cdot d_{\text{Collagen}} \cdot l_{\text{gap}} \cdot b \quad [7]$$

Damit ergeben sich für den theoretisch maximal erreichbaren Mineralisierungsgrad mit Gleichung [3] – [7]:

$$\Phi_{\text{overlap,max}} = \frac{V_{\text{Mineral,overlap,max}}}{V_{\text{gesamt,overlap}}} = \frac{n \cdot \Delta d \cdot l_{\text{overlap}} \cdot b}{n \cdot d_{\text{wet}} \cdot l_{\text{overlap}} \cdot b} = \frac{\Delta d}{d_{\text{wet}}} \quad [8]$$

$$\Phi_{\text{gap,max}} = \frac{V_{\text{Mineral,gap,max}}}{V_{\text{gesamt,gap}}} = \frac{n \cdot \Delta d \cdot l_{\text{gap}} \cdot b + \frac{1}{5} \cdot n \cdot d_{\text{Collagen}} \cdot l_{\text{gap}} \cdot b}{n \cdot d_{\text{wet}} \cdot l_{\text{gap}} \cdot b} = \frac{\Delta d + \frac{1}{5} \cdot d_{\text{Collagen}}}{d_{\text{wet}}} \quad [9]$$

Mit den angegebenen Werten ergibt sich:

$$\Phi_{\text{overlap,max}} = \frac{\Delta d}{d_{\text{wet}}} = \frac{d_{\text{wet}} - d_{\text{dry}}}{d_{\text{wet}}} = \frac{1.55 - 1.10}{1.55} = 0.29$$

$$\Phi_{\text{gap,max}} = \frac{\Delta d + \frac{1}{5} \cdot d_{\text{Collagen}}}{d_{\text{wet}}} = \frac{d_{\text{wet}} - d_{\text{dry}} + \frac{1}{5} \cdot d_{\text{Collagen}}}{d_{\text{wet}}} = \frac{1.55 - 1.10 + \frac{1}{5} \cdot 1.10}{1.55} = 0.43$$

Im physiologischen Zustand ist allerdings nicht das gesamte Wasser durch Hydroxyapatit ersetzt und es ergibt sich für eine "reale" Knochenstruktur (mineralisiert und feucht):

$$\Phi_{\text{overlap,real}} = \frac{d_{\text{wet}} - d_{\text{bone}}}{d_{\text{wet}}} = \frac{1.55 - 1.25}{1.55} = 0.19$$

$$\Phi_{\text{gap,real}} = \frac{d_{\text{wet}} - d_{\text{bone}} + \frac{1}{5} \cdot d_{\text{Collagen}}}{d_{\text{wet}}} = \frac{1.55 - 1.25 + \frac{1}{5} \cdot 1.10}{1.55} = 0.34$$

$$\text{Volumenanteil :} \quad f_{\text{overlap}} + f_{\text{gap}} = 1 \quad [10]$$

$$\text{Durchschnittlicher Mineralisierungsgrad :} \quad \Phi_{\text{average}} = f_{\text{overlap}} \cdot \Phi_{\text{overlap}} + f_{\text{gap}} \cdot \Phi_{\text{gap}} \quad [11]$$

Wenn man nun von den 40% ($f_{\text{overlap}} = 0.4$) overlap-Regionen und damit 60% ($f_{\text{gap}} = 0.6$) gap-Regionen über die Länge ausgeht, ergibt sich ein durchschnittlicher maximaler Mineralisierungsgrad für "realen" Knochen von:

$$\Phi_{\text{average,real}} = f_{\text{overlap}} \cdot \Phi_{\text{overlap,real}} + f_{\text{gap}} \cdot \Phi_{\text{gap,real}} = 0.4 \cdot 0.19 + 0.6 \cdot 0.34 = 0.28$$

bzw. ein limitierender maximaler Mineralisierungsgrad (wenn alles Wasser durch Hydroxyapatit ersetzt ist) von:

$$\Phi_{\text{average,max}} = f_{\text{overlap}} \cdot \Phi_{\text{overlap,max}} + f_{\text{gap}} \cdot \Phi_{\text{gap,max}} = 0.4 \cdot 0.29 + 0.6 \cdot 0.43 = 0.37$$

Bei dieser vereinfachten Betrachtungsweise, ist eine zweidimensionale Anordnung zugrunde gelegt. Betrachtet man hingegen eine realitätsnähere, dreidimensionale Anordnung ergeben sich die folgenden Mineralisierungsgrade [5]:

$$\Phi_{\text{overlap,dry}} < 1 - (d_{\text{dry}} / d_{\text{wet}})^2 \approx 0.50$$

$$\Phi_{\text{overlap,real}} < 1 - (d_{\text{real}} / d_{\text{wet}})^2 \approx 0.35$$

$$\Phi_{\text{gap,dry}} < 1 - 0.8 \cdot (d_{\text{dry}} / d_{\text{wet}})^2 \approx 0.60$$

$$\Phi_{\text{gap,real}} < 1 - 0.8 \cdot (d_{\text{real}} / d_{\text{wet}})^2 \approx 0.48$$

$$\Phi_{\text{average,real}} = f_{\text{overlap}} \cdot \Phi_{\text{overlap,real}} + f_{\text{gap}} \cdot \Phi_{\text{gap,real}} = 0.4 \cdot 0.35 + 0.6 \cdot 0.48 = 0.43$$

Während der Wert für den vereinfachten 2D-Fall mit $\Phi_{\text{average,real}} = 0.28$ noch unterhalb der in der Realität gemessenen Mineralisierungsgrad von 0.40 bis 0.45 [19, 20] liegt, ist der für den 3D-Fall ermittelte Wert mit $\Phi_{\text{average,real}} = 0.43$ [5] schon relativ realistisch. Der Mineralisierungsgrad kann sich noch etwas erhöhen, wenn man berücksichtigt, dass sich die Mineralplättchen in ihrer gesamten Stärke teilweise in die overlap-Bereiche erstrecken und ihre Länge somit über der gap-Länge liegt. Dieser Schluss wird durch die physiologischen Mineralplattenabmessungen unterstützt. So haben die Mineralplatten innerhalb eines Knochens die typischerweise auftretenden Abmessungen von 2nm - 4nm (etwa die Kollagen-

moleküldicke plus zweimal den Kollagenmolekülfreiraum) mal 100nm, wobei die längere Abmessung in Fibrillenrichtung gefunden wird [19].

Passt man den Mineralisierungsgrad an die Realität an, ergeben sich Anordnungen laut **Abbildung 14**, in welcher auch die unterschiedlichen Mineralisierungsgrade (Φ), E-Moduli (E'), maximalen Dehnungen (ε'_{max}) und maximalen Spannungen (σ'_{max}) angegeben sind. Die Komposite-Materialkenn Daten sind dabei auf die Daten für reines Kollagen normiert (\cdot) [5]. Basierend auf den vorangegangenen Überlegungen und Berechnungen wurde das nachfolgende Modell entwickelt.

2.2 Das "Staggered Model"

Die in **Abbildung 14** (d) dargestellte Anordnung bezeichnet man als "Staggered Model". Für dieses Modell spricht neben der Realisierbarkeit realer Mineralisierungsgrade auch die Tatsache, dass in der Realität mit steigendem Mineralisierungsgrad (Φ) sowohl der E-Modul als auch die Zugfestigkeit zunimmt, was sich mit den in **Abbildung 14** (b, c) abgebildeten Anordnungen nicht realisieren lässt. Diese stellen Kombinationen aus einer seriellen (Reuss) und einer parallelen (Voigt) Anordnung dar. Für das Voigt-Modell gilt allerdings, wie in **Abbildung 19** zu sehen, die Einschränkung, dass die Dehnung der beiden Komponenten gleich sein muss. Daraus ergibt sich eine Limitierung der maximalen Dehnung auf den kleineren Wert des "spröden" ($E_{Mineral} \gg E_{Collagen}$) Hydroxyapatits ($\varepsilon_{Composit} = \varepsilon_{Mineral}$).

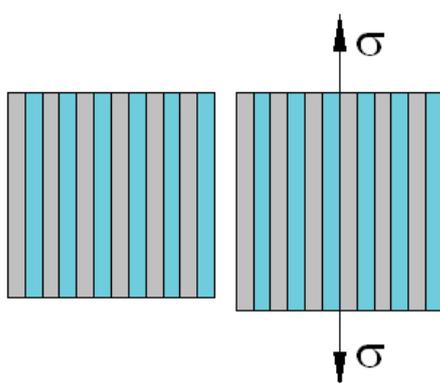


Abbildung 19: Prinzipskizze des Voigt-Modells im ent- und belasteten Zustand mit den zugehörigen Hauptzusammenhängen.

$$\varepsilon_{Composit} = \varepsilon_{Mineral} = \varepsilon_{Collagen} = \varepsilon_{max} \quad [12]$$

$$\varepsilon_{max} = \frac{\sigma_{Composit}}{E_{Composit}} = \frac{\sigma_{Mineral}}{E_{Mineral}} = \frac{\sigma_{Collagen}}{E_{Collagen}} \quad [13]$$

Beim Reuss-Modell hingegen müssen, wie in **Abbildung 20** zu sehen ist, die Spannungen innerhalb der beiden Komponenten gleich sein, daher wird die Verbundmaterial-Zugspannung auf die niedrigere Zugspannung des Kollagens limitiert. ($\sigma_{Composit}^{max} = \sigma_{Collagen}^{max}$).

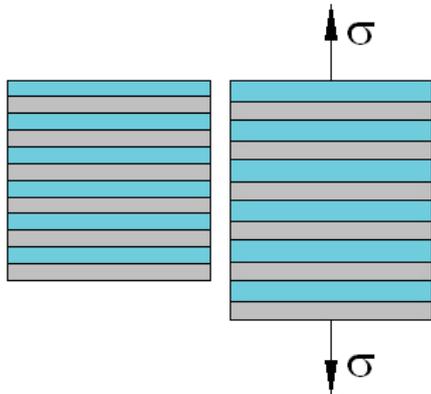


Abbildung 20: Prinzipskizze des Reuss-Modells im ent- und belasteten Zustand mit den zugehörigen Hauptzusammenhängen.

$$\sigma_{Composit} = \sigma_{Mineral} = \sigma_{Collagen} = \sigma_{max} \quad [14]$$

Da im Experiment sowohl Bruchdehnung als auch Zugfestigkeit steigen, muss neben der Normalkraftkomponente auch eine weitere Kraftübertragung stattfinden. Dies ist über Scherkräfte, wie beim "Staggered-Model" welches in **Abbildung 21** gezeigt ist, möglich.

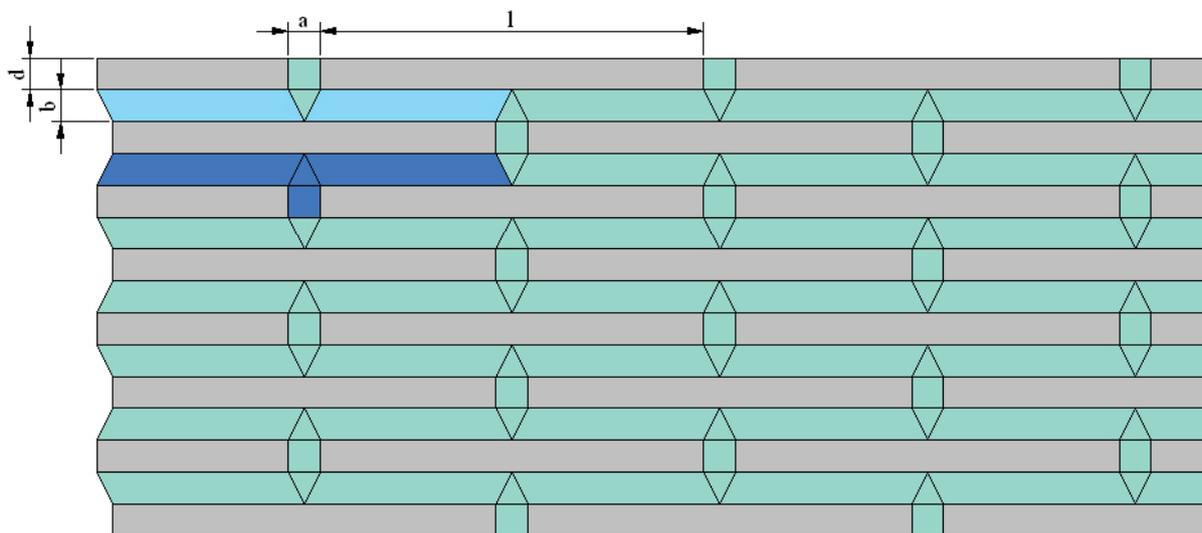


Abbildung 21: Skizze des "Staggered Model", wobei die Mineralplattenbereiche grau und die Kollagenbereiche blaugrün dargestellt sind. Jene Kollagenbereiche, die Normalspannungen übertragen, sind dunkelblau und jene Kollagenbereiche, die Schubspannungen übertragen, sind hellblau dargestellt.

Wie gezeigt werden kann, hat dieses Modell für den 2D-Fall die Form

$$\Phi_{average} \cdot (l+a) \cdot (b+d) = l \cdot d \Rightarrow \Phi_{average} = \frac{l \cdot d}{(l+a) \cdot (b+d)} = \frac{l}{l+a} \cdot \frac{d}{b+d} \quad [15]$$

und damit bei fixiertem, konstantem Mineralisierungsgrad folgende Grenzen:

$a = 0$ würde den Limitierungen des Reuss-Modell (durchgehende Mineralplatten) entsprechen und $a > l$ den Limits des Voigts-Modells (Kopplung reiner Kollagenbereiche mit Kollagen-Mineralplatten-Komposite):

$$\text{Für } a > 0 \text{ ergibt sich } \frac{l}{l+a} < 1 \Rightarrow \frac{d}{b+d} > \Phi_{average} \quad [16]$$

$$\text{für } a \leq l \text{ ergibt sich } \frac{l}{l+a} > 0.5 \Rightarrow \frac{d}{b+d} \leq 2 \cdot \Phi_{average} \quad [17]$$

Für den Mineralisierungsgrad der beiden Überlagerungsbereiche ergibt sich somit für $f_{gap} = 0.6$ mit Gleichung [11]:

$$\Phi_{average} \leq \Phi_{gap} \leq \frac{5}{4} \cdot \Phi_{average} \quad [18] \qquad \frac{5}{8} \cdot \Phi_{average} \leq \Phi_{overlap} \leq \Phi_{average} \quad [19]$$

Die Komposite-Eigenschaften variieren dabei, bei fixiertem Mineralisierungsgrad (also fixiertem d und l), vor allem durch Änderung der Abmessungen a und b . Die Zusammenhänge verdeutlicht **Abbildung 22**, wobei auffällt, dass sich die Härte (der E-Modul) in beiden Fällen erhöht, also wenn a oder b minimiert wird.

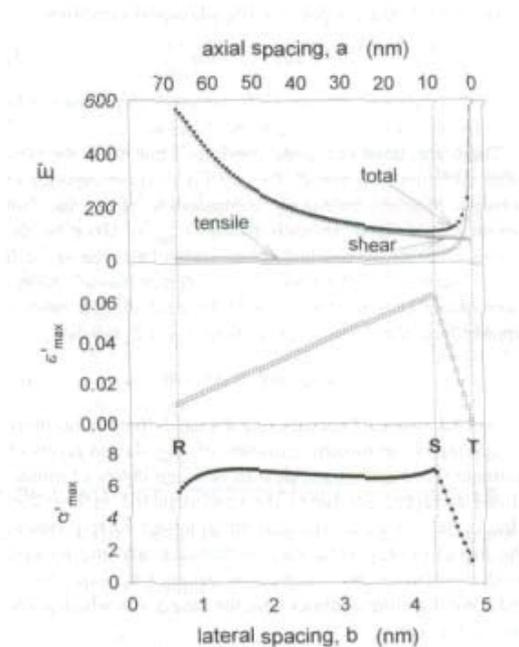


Abbildung 22: Die auf unmineralisierten Kollagen normierten Materialkennwerte E-Modul E' , mit den dazugehörigen Schub- (shear) und Zug- (tensile) anteil, maximale Dehnung ϵ_{max}' und maximale Spannung σ_{max}' bei Variation der Plättchenabstände a und b für Plättchenabmessungen von $d = 3.5\text{nm}$ und $l = 70\text{nm}$ sowie einen Mineralisierungsgrad von $\Phi = 0.42$. Die Buchstaben stehen dabei für die Grenzfälle $a = 1$ (R), $a = 2 \cdot b$ (S) und $a = 0$ (T) des Modells [5].

2.3 Die Indentationsmodul-Bestimmung nach Oliver/Pharr

In der Erforschung von Materialverhalten auf Mikroebene gab es große Fortschritte seit es Messgeräte gibt, die während eines Härteeindruckversuches auf dieser Größenskala kontinuierlich Kraft und Eindringtiefe messen können [9]. Diese Messdaten können somit zur Auswertung herangezogen werden, selbst wenn die Eindrücke selbst zu klein sind um visuell auswertbar zu sein. Zwei der so am häufigsten ermittelten Kennwerte sind die Härte H und der E-Modul E . Kontaktprobleme, so wie sie bei einer Härteeindruckmessung auftreten, wurden bereits am Ende des 19 Jahrhunderts wissenschaftlich behandelt, wobei besonders Boussinesq und Hertz anzuführen sind [12]. Boussinesq entwickelte eine Methode, basierend auf der Potentialtheorie um Spannungen und Verschiebungen in einem rein elastischen Körper durch einen starren, achsensymmetrischen Indenter zu berechnen. Daraus wurden Lösungen für die verschiedensten Indenterformen abgeleitet. Hertz analysierte den Kontakt zwischen zwei elastischen Kugeln unterschiedlichen Durchmessers und E-Moduls. Seine Lösungsvorschläge bilden die Grundlage für viele Problemstellungen der Kontakt-Mechanik und liefern das Gerüst zur Einbindung nicht starrer Indenter. Ein weiterer wichtiger Beitrag wurde von Sneddon geliefert, der generelle Zusammenhänge zwischen Kraft, Verschiebung und Kontaktfläche für unterschiedlichste Indenter aufzeigte. So gilt für viele einfache Indenterformen folgender Zusammenhang zwischen Kraft und Verformung:

$$F = \alpha \cdot u^m \quad [20]$$

F ist dabei die Indenterlast (Kraft), u die elastische Indentationstiefe (Verformung) und α sowie m sind indenterformabhängige Konstanten mit den typischen Werten von $m = 1$ (flacher Zylinder), $m = 1.5$ (parabolischer Indenter und Kugel (bei geringem u)) und $m = 2$ (Kegel). Verlassen wir nun das Gebiet des rein elastischen Materialverhaltens, so stellen wir fest, dass Rechenmodelle für jegliches plastische Materialverhalten ein komplexeres Problem darstellen. Dabei werden nicht nur die einzelnen Gleichungen nichtlinear sondern es müssen auch diverse Materialparameter, die das Materialverhalten wiedergeben, berücksichtigt werden. Daher werden für solche Problemstellungen, wie in der vorliegenden Arbeit, anstatt der analytischen Lösungen oft durch Simulationen ermittelte Näherungslösungen gesucht. Das hier zu beschreibende Verfahren geht von Messdaten aus, welche während eines kompletten Be- und Entlastungszyklus aufgenommen werden. Die Entlastungskurve wird dann entsprechend eines Modells der Verformung eines Symmetrievolumens mit rein elastischem Verformungsverhalten durch einen elastischen Eindruckkörper analysiert, wobei

die Kontaktfläche bei Hauptlast zum E-Modul in Relation gesetzt wird. Unter Berücksichtigung der Indenterform können dann H und E ausgewertet werden. Untersuchungen ergaben darüber hinaus folgende Ergebnisse:

- Bei Verwendung von konischen Indentern bleibt der Eindruckquerschnitt während der Entlastung, aufgrund des "sink-in"-Effekts ("Verformung der Ränder"), wie in **Abbildung 23** gezeigt, konstant und lediglich die Eindringtiefe nimmt bei der Entlastung ab. Bei Entlastung ändert sich durch elastische Rückverformung die Eindringtiefe von der maximalen Eindringtiefe (bei Maximallast) u_{max} auf die bleibende Eindringtiefe (plastische Verformung) u_f .

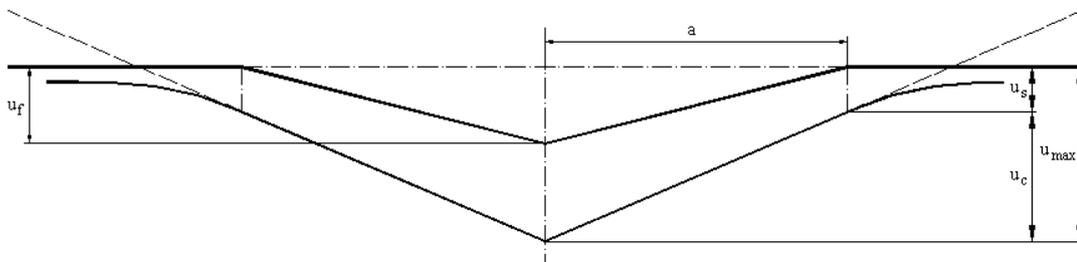


Abbildung 23: Skizze des "sink-in"-Effekts, wobei die Probenoberfläche strichpunktiert und der Indenter bei Maximallast strichliert dargestellt ist. Das Maß a repräsentiert die Projektionsfläche A und die auftretenden Tiefen sind die maximale Eindringtiefe u_{max} (kombiniert aus der Kontakttiefe u_c und der Oberflächenverformung u_s) sowie die bleibende Eindringtiefe u_f .

Voraussetzung für den gezeigten "sink-in"-Effekt ist ein hohes Verhältnis:

$$\Theta = \frac{H}{E} = \frac{\text{wahreHärte}}{E - \text{Modul}_{\text{Probe}}} \left[\frac{\frac{N}{\text{mm}^2}}{\frac{N}{\text{mm}^2}} \right] \quad [21]$$

Ist das Material zu "weich" kommt es beim Eindringversuch zu elastischen Verformungen und zur Ausbildung eines "Wulstes" welcher über die ursprüngliche Oberfläche hinausragt. Dies wäre der so genannte "pile-up"-Effekt, wie er in **Abbildung 24** dargestellt ist.

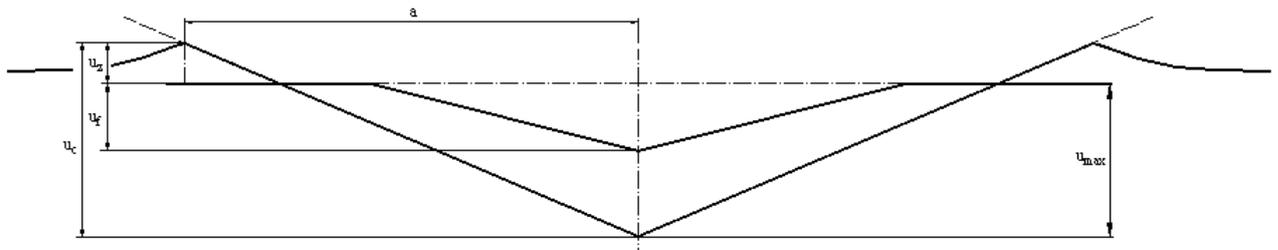


Abbildung 24: Skizze des "pile-up"-Effekts, wobei die Probenoberfläche strichpunktiertere und der Indenter bei Maximallast strichliert dargestellt ist. Das Maß a repräsentiert die Projektionsfläche A und die auftretenden Tiefen sind die maximale Eindringtiefe u_{max} (kombiniert aus der Kontakttiefe u_c und der Oberflächenverformung u_s) sowie die bleibende Eindringtiefe u_f .

Die Grenze zwischen diesen Effekten liegt bei etwa $\Theta = 0.03$ [10].

- Um eine vollkommen reversible Be- und Entlastungskurve zu bekommen muss vorher mehrmals eine Be- und Entlastung durchgeführt werden.
- Der Einfluss eines nicht starren Indenters (non rigid tip) kann wie folgt berücksichtigt werden:

$$\frac{1}{E_r} = \frac{(1 - \nu_{Pr}^2)}{E_{Pr}} + \frac{(1 - \nu_{In}^2)}{E_{In}} \quad [22]$$

Dabei ist E_r der reduzierte E-Modul (reduced Young' modulus), E_{Pr} und ν_{Pr} der E-Modul und die Poissonzahl der Probe und E_{In} und ν_{In} der E-Modul und die Poissonzahl des Indenters.

Die Kontakttiefe u_c , die Oberflächenverformungstiefe am ersten Kontaktpunkt u_s , die sich daraus addierte, maximale Eindringtiefe (bei F_{max}) u_{max} , der Kontaktflächenabstand (bei rotationssymmetrischer Betrachtungsweise der Kontaktflächenradius) a , sowie die bleibende (plastische Verformung) Eindringtiefe nach Entlastung u_f kann aus **Abbildung 23** und **Abbildung 24** ersehen werden. Aus einer experimentellen Kraft-Eindringtiefen-Kurve (load-displacement-curve), wie sie in **Abbildung 25** schematisch dargestellt ist, und der elastischen Kontakt-Theorie kann man die Härte (Entlastungssteifigkeit) auch wie folgt näherungsweise ermitteln:

$$S = \frac{dF}{du} = \frac{2}{\sqrt{\pi}} \cdot E_r \cdot \sqrt{A} \quad [23]$$

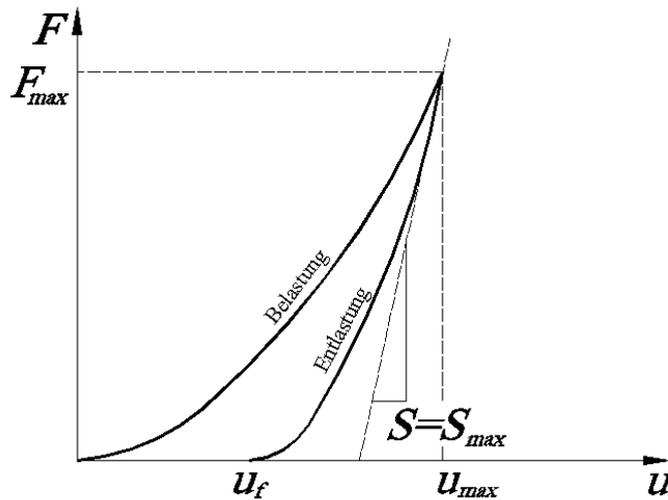


Abbildung 25: Prinzip einer Kraft-Eindringtiefe-Kurve (load displacement curve). Mit steigender Kraft F steigt auch die Eindringtiefe u , wobei das Eindringen auch aufgrund der größer werdenden Kontaktfläche schwerer wird. Die maximale Eindringtiefe u_{max} verringert sich bei Entlastung aufgrund elastischer Rückverformung auf die bleibende Eindringtiefe u_f . S ist die Tangente an die Entlastungskurve, wobei S_{max} die Entlastungssteifigkeit (S bei u_{max}) darstellt.

S ist also die Steigung an die Entlastungskurve am Punkt der höchsten Last, wobei A die Projektionsfläche des elastischen Kontakts und E_r den zuvor erwähnten reduzierten E-Modul darstellen. Diese Gleichung gilt, wie diverse Veröffentlichungen belegen, für jegliche Indenterform. Die beiden für die Ermittlung der Härte ausschlaggebenden Parameter F_{max} (größte auftretende Last) und u_{max} (maximale Eindringtiefe) sowie die Entlastungssteifigkeit (initial unloading contact stiffness) S_{max} (Anstieg der Tangente in diesem Punkt) sind in **Abbildung 25** angeführt. Hier sei erwähnt, dass die Entlastungssteifigkeit immer bei F_{max} ausgewertet wird, unabhängig davon, ob die Entlastungskurve in irgendeinem Teilgebiet einen linearen Verlauf hat. Um nun den E-Modul nach Oliver/Pharr zu bestimmen, kann man die oben angeführte Gleichung umformen und erhält:

$$E_{OP} = E_r = \frac{\sqrt{\pi}}{2} \cdot \frac{S}{\sqrt{A}} \quad [24]$$

Der E-Modul nach Oliver/Pharr ist genauer betrachtet allerdings auch noch von einem empirischen Indenterformfaktor β abhängig, daher ergäbe sich:

$$E_{OP} = E_r = \beta \cdot \frac{\sqrt{\pi}}{2} \cdot \frac{S}{\sqrt{A}} \quad [24a]$$

Da der Kontaktflächenabstand a (zur Bestimmung der Kontaktfläche A) im Experiment nur schwer bestimmbar ist, kann man die Kontaktfläche A auch als Funktion der Indentationstiefe angeben:

$$A = f(u_c) \quad [25]$$

wobei die verwendete Funktion aus dem Experiment bzw. dem Versuchsaufbau abzuleiten ist. Allerdings ist auch der Abstand u_c nicht einfach zugänglich. Daher wird auf die Gleichungen von Sneddon für konische Indenter zurückgegriffen und man erhält, wenn man u durch $u - u_f$ ersetzt, da Sneddon's Ansatz ja für rein elastisches Materialverhalten ermittelt wurde:

$$u_s = \frac{\pi - 2}{\pi} \cdot (u - u_f) \quad [26]$$

$$(u - u_f) = 2 \cdot \frac{F_{\max}}{S} \quad [27]$$

Durch Einsetzen dieser Gleichungen und Umformung ergibt sich somit für konische Indenter:

$$u_s = \frac{\pi - 2}{\pi} \cdot 2 \cdot \frac{F_{\max}}{S} \quad [28]$$

bzw. die allgemeine Gleichung:

$$u_s = \varepsilon \cdot \frac{F_{\max}}{S} \quad [29]$$

ε ist dabei eine indentergeometrieabhängige Konstante und hat eine Größe von $\varepsilon = 2 \cdot \frac{\pi - 2}{\pi} = 0.73$ (konischer Indenter), $\varepsilon = 0.75$ (parabolischer Indenter)

bzw. $\varepsilon = 1$ (flacher Indenter). Diese Größe ist somit implizit in der in Gleichung [24a] angeführten Indenterformfaktor β enthalten. Der Einfluss der Indentergeometrie ist in **Abbildung 26** gezeigt.

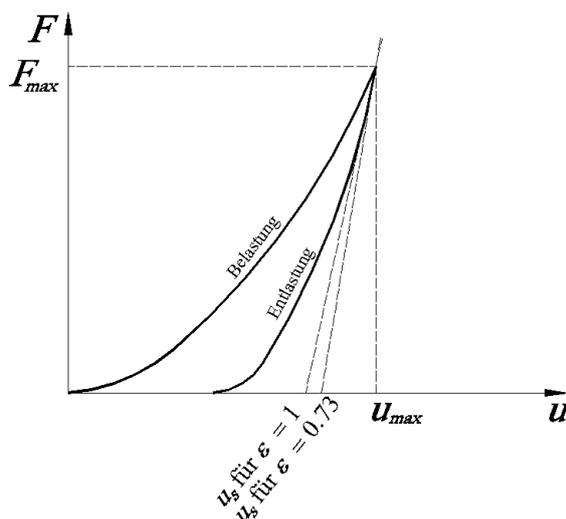


Abbildung 26: Prinzip einer Kraft-Eindringtiefe-Kurve (load displacement curve) mit der maximalen Eindringtiefe u_{\max} (bei Maximalkraft F_{\max}) sowie der indenterformabhängigen (durch unterschiedliche ε -Werte wiedergegebenen) Kontakttiefe u_s .

Genau in der gleichen Weise kann der zweite Materialkennwert, die Härte H (bezieht man sich bei der Berechnung auf die projizierte Fläche, spricht man oft auch von der "wahren Härte") mit folgender Gleichung bestimmt werden:

$$H = \frac{F_{\max}}{A} \quad [30]$$

Hier sei noch angemerkt, dass die ermittelten Härtewerte von dem verwendeten Messverfahren und dem zu untersuchenden Material abhängen. Grund dafür ist beispielsweise die Tatsache dass bei anderen Methoden (z.B. Härtemessverfahren nach Vickers) der entstandene, plastische Härteabdruck zur Flächenbestimmung herangezogen wird. Die so ermittelte Fläche kann allerdings geringer sein, als die Fläche bei F_{\max} , da es nicht immer zu einer hundertprozentigen Plastifizierung kommt. Der Härtewert kann somit für dieses Verfahren höher liegen. Härte ist ganz allgemein der Widerstand eines Materials gegen das Eindringen eines (meist härteren) Körpers. Zur Veranschaulichungen sei hier folgendes Beispiel angeführt. Um in einem Material mit kristalliner Struktur eine Eindringung zu erreichen, muss der Abstand zwischen den regelmäßig im Gitter angeordneten Gitteratomen verändert werden. Die Gitteratome kann man sich vorstellen wie durch Federn gekoppelte Punktmassen, wobei der E-Modul der Federkonstanten dieser Kopplungsfedern gleichkommt. Die Federkräfte entsprechen somit den Bindungskräften innerhalb des Kristalls. Ein hoher E-Modul korrespondiert daher meist mit einer höheren Härte, da der Widerstand gegen eine Abstandsvergrößerung mit höherem E-Modul zunimmt. So sind beispielsweise die Bindungskräfte der sp^3 -Hybridisierung des Diamant extrem hoch, was einen hohen E-Modul zur Folge hat und somit die Härte dem Diamant erklärt.

2.4 Die Vergleichsspannung nach von Mises

Für die Ergebnisvisualisierung ist es erforderlich, die Belastung jeder einzelnen Zelle darzustellen. Denkbar wäre eine Darstellung der Zug- bzw. Druckspannungen $\sigma_{z,d}$, allerdings würden dabei die durch Scherung usw. auftretenden Schubspannungen τ vernachlässigt werden. Da Zug- und Schubspannungen rechtwinkelig aufeinander stehen, ist eine skalare Addition unzulässig. Eine vektorielle Addition wäre gestattet, ist allerdings nicht übersichtlich darzustellen. Die Lösung dieses Problems stellt die Überlegung dar, eine Vergleichsspannung σ_v heranzuziehen, die den Werkstoff in gleicher Weise belastet wie der reale Belastungsfall.

Eine im Maschinenbau übliche Vergleichspannungsbestimmung, die auch innerhalb von ABAQUS Anwendung findet, geht auf den österreichischen Mathematiker Richard Edler von Mises (1883-1953) zurück, der diese für den 2D-Fall wie folgt berechnet:

$$\sigma_{v.Mises} = \sqrt{\sigma_{xx}^2 + \sigma_{yy}^2 - \sigma_{xx} \cdot \sigma_{yy} + 3 \cdot \tau_{xy}^2} \quad [31]$$

Hierbei sind σ_{xx} und σ_{yy} , wie in **Abbildung 27** gezeigt, die Zugspannungen in x- bzw. y-Richtung und τ_{xy} die Schubspannung in der xy-Ebene.

Analog ergibt sich für den 3D-Fall:

$$\sigma_{v.Mises} = \sqrt{\sigma_{xx}^2 + \sigma_{yy}^2 + \sigma_{zz}^2 - \sigma_{xx} \cdot \sigma_{yy} - \sigma_{xx} \cdot \sigma_{zz} - \sigma_{yy} \cdot \sigma_{zz} + 3 \cdot \tau_{xy}^2 + 3 \cdot \tau_{xz}^2 + 3 \cdot \tau_{yz}^2} \quad [32]$$

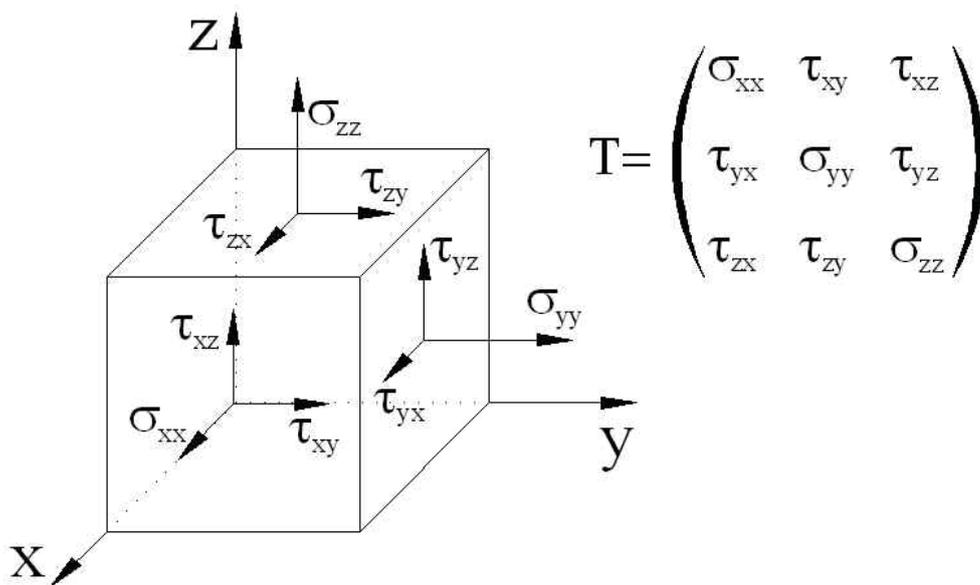


Abbildung 27: Skizze der auftretenden Spannungskomponenten in einem kartesischen Koordinatensystem. Es wird deutlich, dass die Schubspannungen τ_{xy} und τ_{yx} gleich groß sein müssen (analog $\tau_{xz} = \tau_{zx}$, $\tau_{yz} = \tau_{zy}$), da sonst die Volumenzelle an der Kante "aufreißen" würde. Dies führt zu einer Symmetrie, die als Transformationsregel aus der Matrix einen Tensor macht. Der Transformationstensor ist dadurch definiert, dass sich in einem Koordinatensystem seine Komponenten auf bestimmte Weise transformieren lassen. Der Spannungstensor T ist darüber hinaus ein Tensor (lat.: ‚tendo‘ = "ich spanne") zweiter Stufe, da er eine quadratische Matrix darstellt. (Ein Tensor erster Stufe wäre ein Vektor und Tensor nullter Stufe ein Skalar.) Somit repräsentiert jeder Tensor eine Matrix, aber nicht jede Matrix ist ein Tensor.

In beiden Fällen, Gleichung [31] und [32], stellt der erste Teil mit den Quadraten den Betrag des resultierenden Spannungsvektors dar. Die gemischten Glieder der einzelnen Komponenten resultieren aus der Querkontraktion des Werkstoffes. Die Terme mit den Schubspannungen dienen der Drehung des Koordinatensystems. Dadurch muss nicht auf die drei Hauptspannungen umgerechnet werden. Würde man zuerst die Hauptspannungen σ_1 , σ_2 und σ_3 errechnen, würden die Terme mit den Schubspannungen entfallen, aber das Ergebnis wäre dasselbe. Angemerkt sei hierbei, dass diese zur Vergleichspannungsberechnung verwendete Anstrengungshypothese, ausgehend von der Gestaltänderungsarbeit, für zähe, plastifizierungsfähige Werkstoffe aufgestellt wurde. Sie dient dabei zur Fließgrenzenbestimmung. Erreicht die v.Mises-Spannung ($\sigma_{v.Mises}$) den Wert der Streckgrenze (R_e), fängt der Werkstoff an zu plastizieren. Dies ist gleichbedeutend mit dem Verlassen der Hook'schen Gerade im Spannungs-Dehnungs-Diagramm. Gekennzeichnet ist dies bei einem solchen Werkstoff durch das Auftreten von Lüdersbändern, d.h. die Versetzungen werden z.B. durch Legierungselemente oder Verunreinigungen (Cottrell-Wolke) in ihrer Bewegung behindert, reißen sich durch plastische Verformung des Werkstoffes von diesen los. Da es für die interstitiell gelösten (aus Zwischengitterplätzen befindlichen), meist größeren Atome energetisch günstiger ist, sich wegen des sie umgebenden Druckspannungsfeldes aufgrund ihres unterschiedlichen Durchmessers, in den Zugbereichen der Versetzungen anzulagern, folgen diese den Versetzungen. Werden die Versetzungen von den nachdiffundierenden Atomen (Wolke) eingeholt, kommt es erneut zu einer Fixierung der Versetzung. Diese Wechselwirkung kann sich mehrmals wiederholen bis die Versetzungen sich aufgrund der steigenden Spannungen schneller bewegen als die Diffusionsgeschwindigkeit der blockierenden Atome. Dringen diese Versetzungen an die Materialoberfläche vor, bilden sich Versetzungsstufen (Intrusionen bzw. Extrusionen). Die dadurch, z.B. an der Blechoberfläche, auftretenden Oberflächenrauigkeiten, sind dann als Linien erkennbar und werden als Lüdersbänder bezeichnet.

Auf den in dieser Arbeit behandelten Verbundwerkstoff, aus einem spröden (Mineralplättchen) und einem zähen (Kollagenfasern) Werkstoff, ist diese Vergleichspannungsberechnung damit nur bedingt übertragbar. Hier wäre die Rechnung mittels der Hauptspannungen empfehlenswert. Die sich daraus ergebenden Unstimmigkeiten werden allerdings als unerheblich eingestuft und daher in dieser Arbeit vernachlässigt, zumal verwendete Programmpaket ABAQUS eine solche Analyse auch nicht anbietet.

3 Die Finite Elemente Methode (FEM)

3.1 Theoretische Grundlagen

Die Finite Elemente Methode ist ein numerisches Verfahren zur näherungsweise Lösung, insbesondere partieller Differentialgleichungen in ihrer integralen Form mittels z.B. Variationsrechnung unter beliebig variierbaren Randbedingungen. Sie ist auch ein weit verbreitetes modernes Berechnungsverfahren im Ingenieurwesen, mit welcher Problemstellungen aus den verschiedensten Disziplinen behandelt werden können. Die Problemstellungen unterscheiden sich dabei lediglich in der zugrunde liegenden Differentialgleichung. So liefert die Lösung der Gleichungen der Elastizitätstheorie das Verhalten, also die Verformungen und die Spannungen, von Festkörpern unter äußeren Belastungen, während die Lösung der Laplace-Gleichung die Beschreibung von Temperaturfeldern ermöglicht. Löst man die so genannten Navier-Stokes-Gleichungen ergibt sich das Strömungsverhalten eines Fluids. Die Lösung der Maxwell-Gleichungen hingegen ermöglicht die mathematische Behandlung von Magnetfeldern. Die Lösung der Differentialgleichungen ist für einfache Geometrien und Aufgabenstellungen analytisch möglich, nicht aber für komplexe, diskontinuierliche Systeme, wie sie in der Praxis auftreten. Hierfür sind numerische Näherungsverfahren besser geeignet bzw. stellen oft die einzige Möglichkeit dar. Ein Verfahren, das sich als besonders "mächtig" bzw. zielführend erwiesen hat, ist dabei neben dem FVM (Finiten-Volumen-Verfahren), dem FDM (Finiten-Differenzen-Verfahren), der REM (Randelementmethode) und der FPM (Finite-Punkte-Methode) die von ABAQUS verwendete FEM. Wie bei den meisten anderen Verfahren wird dabei das Berechnungsgebiet in eine große Zahl kleiner, aber endlich vieler Teilgebiete unterteilt. Die dabei bei der FEM erhaltenen Elemente sind also endlich (finit) und nicht unendlich (infini) klein, woraus sich der Name der Methode ableitet [12]. Die FEM unterscheidet sich dabei von anderen Näherungsverfahren, da bei der FEM nicht über das gesamte Lösungsgebiet sondern auf jedem dieser einzelnen Elemente eine Ansatzfunktion (linear, quadratisch, kubisch,...) definiert wird, aus denen sich dann durch Algebraisierung der partiellen Differentialgleichung (z.B. durch partielle Integration bzw. Lösung der Variationsformulierung) unter Einbau der Randbedingungen ein i.a. nicht lineares Gleichungssystem aufbaut. Aus der Lösung dieses Gleichungssystems werden danach die weiteren Feldvariablen abgeleitet. Das Verfahren wurde in den 1950er Jahren entwickelt und findet dank enormer Rechnerleistungssteigerungen immer mehr Anwendungen.

3.2 Allgemeines Vorgehen

Das untersuchte Lösungsgebiet G wird zunächst in Teilgebiete G_e , die finiten Elemente eingeteilt [12]:

$$G = \sum_{e=1}^m G_e \tag{34}$$

Diese Elemente werden durch die Elementknoten miteinander verbunden. Jeder Knoten hat eine bestimmte Anzahl von Freiheitsgraden (DOF's). Entsprechend diesen werden für die gesuchte Lösung je n Ansatzfunktionen Ψ_i definiert. Diese Funktionen sind nur innerhalb jenes Patches, der den zugehörigen Knoten enthält/umgibt ungleich Null. Dies veranschaulicht **Abbildung 28**. Im unteren Teil sieht man 6 lineare Ansatzfunktionen eine "pyramidenförmige" Ansatzfunktion in einem Patch um den Knoten n aufspannen (schraffierter Bereich in der x_1 - x_2 -Ebene). Sie hat in n den Wert 1, innerhalb des Patch einen Wert zwischen 1 und 0 und außerhalb des Patches den Wert 0.

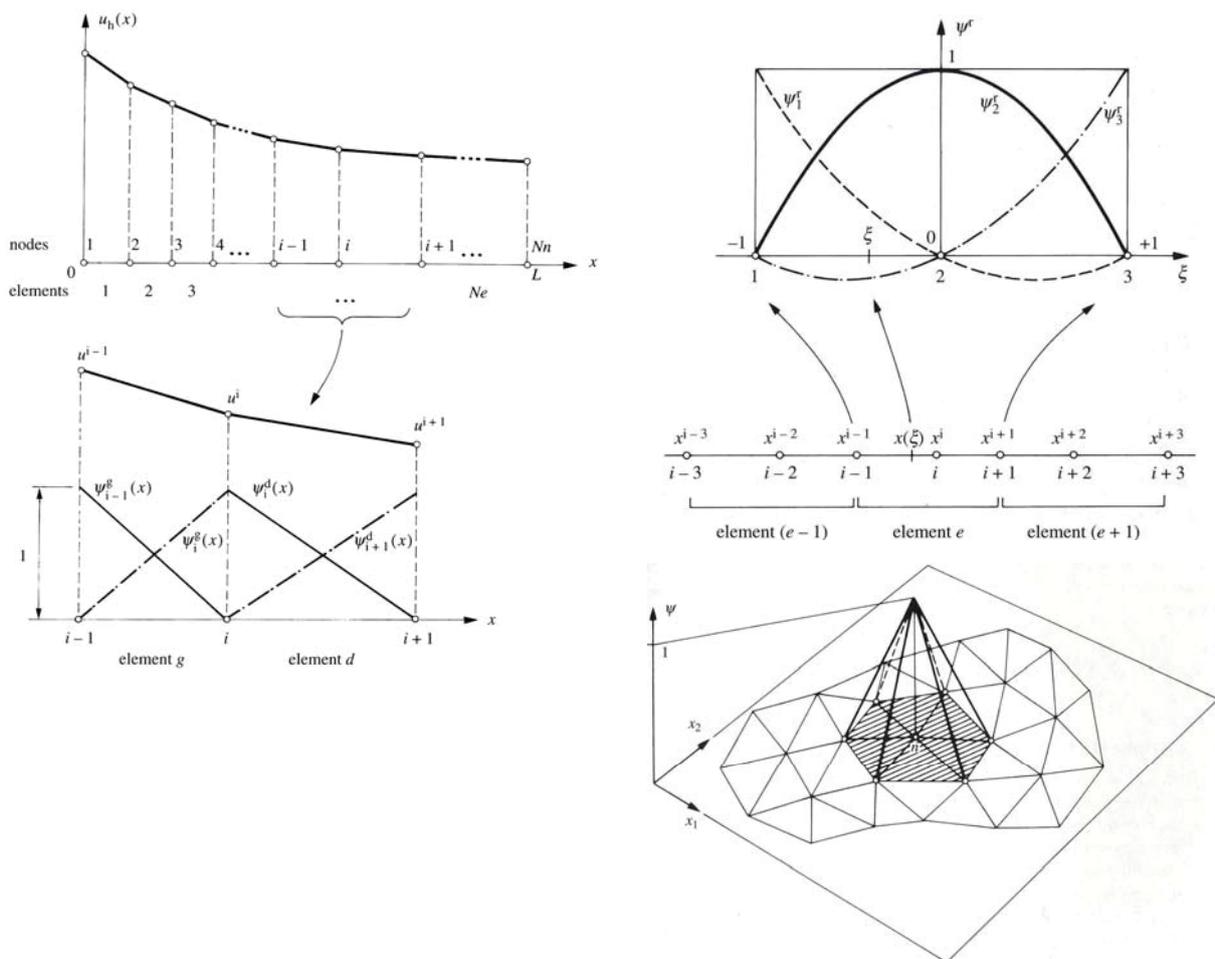


Abbildung 28: Einfaches Beispiel einer stückweise linearen (oben links) bzw. quadratischen (oben rechts) Ansatzfunktion Ψ für einen 1D-Fall sowie einer stückweise linearen Ansatzfunktion für einen 2D-Fall (unten). [8]

Durch eine Linearkombination der n Ansatzfunktionen innerhalb des Elementes wird die numerische Näherungslösung festgelegt [12]:

$$y|_{G_e} \approx \sum_{i=1}^n c_{e,n} \cdot \Psi_{e,n} \quad [35]$$

Innerhalb eines Finiten Element kann dann mit klassischen Methoden der Kontinuumsmechanik bzw. Physik (Ritz, Gallerkin,...) das Randwertproblem algebraisiert werden. Die dabei auftretenden integrale können i.a. nur noch numerisch gelöst werden. Da die Ansatzfunktionen, wie man anhand von **Abbildung 28** erkennt, nur auf wenigen der Elemente (den an den betrachteten Knoten angrenzenden Elementen) ungleich Null sind, ergibt sich ein dünnbesetztes, häufig sehr großes, lineares Gleichungssystem (Systemmatrix mit Bandstruktur), bei dem die Faktoren der Linearkombination unbekannt sind:

$$\underline{K} \cdot \underline{u} = \underline{F} \quad [36]$$

\underline{K} ist dabei die Steifigkeitsmatrix (Materialkennwerte und geometrische Größen), \underline{u} der Lösungsvektor (z.B. der Verschiebungsvektor) und \underline{F} der Lastvektor.

Dieses Gleichungssystem könnte man zwar prinzipiell direkt (Gaußsches Eliminationsverfahren,...) lösen, allerdings würde der Berechnungsaufwand dort äußerst hoch (bei N Gleichungen $O(N^3)$) werden und die "dünnbesetzte" Struktur (die sich effizient speichern lässt) verloren gehen. Darüber hinaus würde eine schlechte Konditionierung des Gleichungssystems numerische Schwierigkeiten bereiten. Daher verwendet man i.a. iterative Lösungsverfahren, die schrittweise nach einer besseren Lösung suchen. Einfache Beispiele dafür sind das Jacobi- und Gauss-Seidel-Verfahren. In der Praxis werden aber eher Mehrgitterverfahren oder vorkonditionierte Krylow-Unterraum-Verfahren, wie das Verfahren der konjugierten Gradienten, verwendet [12]. Es gibt heute eine Vielzahl von kommerziellen Computerprogrammen, die nach der Methode der Finiten Elemente arbeiten, wobei das in dieser Arbeit verwendete Programmpaket ABAQUS eines der bekanntesten und universellsten darstellt.

3.3 Diskretisierung

Die gegebene Aufgabe wird diskretisiert, indem man ganz allgemein das Grundgebiet in einfache Teilgebiete, die so genannten Elemente, zerlegt (vgl. **Kapitel 4.4**). Bei gewissen Aufgabenstellungen ist die Aufteilung in Elemente durch das Problem bereits weitgehend vorgegeben. Das ist beispielsweise bei einem räumlichen Fachwerk der Fall, bei welchem die einzelnen Stäbe die Elemente der Konstruktion bilden. Dasselbe gilt etwa auch bei Rahmenkonstruktionen, wo die einzelnen Balken oder unterteilte Balkenstücke die Elemente

der Aufgabe darstellen. Im Fall von zweidimensionalen Problemen wird das Grundgebiet in Dreiecke, Parallelogramme, krummlinige Dreiecke oder Vierecke eingeteilt. Selbst wenn nur geradlinige Elemente verwendet werden, erreicht man mit einer entsprechend feinen Diskretisierung eine recht gute Approximation (Annäherung) des Grundgebietes. Die Diskretisierungsgüte beliebig berandeter Gebiete kann man durch die Verwendung von krummlinigen Elementen selbstverständlich erhöhen. Jedenfalls erlaubt die räumliche Diskretisierung eine äußerst flexible und auch dem Problem angepasste Erfassung des Grundgebietes. Allerdings muss darauf geachtet werden, dass sehr spitze oder überstumpfe Winkel in den Elementknoten vermieden werden, um numerische Schwierigkeiten auszuschließen. Das gegebene Gebiet G wird somit durch die Fläche der approximierenden Elemente ersetzt. Bei räumlichen Problemen erfolgt eine Diskretisierung des dreidimensionalen Gebietes in Tetraederelemente, Quaderelemente oder andere dem Problem angepasste, möglicherweise auch krummflächig berandete Elemente (In der Regel sind dies Serendipity- oder Lagrange-Elemente.) [12].

3.4 Element-Ansatz

In jedem der Elemente wird für die das Problem beschreibenden Funktionen, ein problemgerechter Ansatz gewählt. Im Besonderen eignen sich dazu ganze rationale Funktionen in den unabhängigen Raumkoordinaten. Für eindimensionale Elemente (Stäbe, Balken) kommen Polynome ersten, zweiten, dritten und gelegentlich sogar höheren Grades in Frage. Bei zweidimensionalen Problemen finden bilineare, biquadratische oder höhergradige Polynome Verwendung. Die Art des Ansatzes hängt dabei von der Wahl des Elementtyps ab. Die Ansatzfunktionen müssen beim Übergang von einem Element ins benachbarte ganz bestimmte Stetigkeitsbedingungen erfüllen. Die Stetigkeitsanforderungen sind häufig aus physikalischen Gründen offensichtlich und aus mathematischen Gründen erforderlich. Zum Beispiel muss die Verschiebung eines zusammenhängenden Körpers in einer Richtung beim Übergang von einem Element zum anderen stetig sein um die Kontinuität des Materials zu gewährleisten. Im Fall der Balken- oder Plattenbiegung sind die Stetigkeitsanforderungen höher, da dort aus analogen physikalischen Gründen sogar die Stetigkeit der ersten Ableitung bzw. der beiden ersten partiellen Ableitungen gefordert werden muss. Elemente mit Ansatzfunktionen, welche den Stetigkeitsbedingungen genügen, heißen konform. Um nun die Stetigkeitsanforderungen tatsächlich zu erfüllen, muss der Funktionsverlauf im Element durch Funktionswerte und auch durch Werte von (partiellen) Ableitungen (den Knotenpunktverschiebungen) in bestimmten Punkten des Elementes, den Knotenpunkten,

ausgedrückt werden. Die in den Knotenpunkten benutzten Funktionswerte und Werte von Ableitungen nennt man die Knotenvariablen des Elements. Mit Hilfe dieser Knotenvariablen stellt sich die Ansatzfunktion als Linearkombination von sogenannten Formfunktionen mit den Knotenvariablen als Koeffizienten dar. Es ist zweckmäßig, für die Knotenpunktkoordinaten neben einem elementbezogenen lokalen ein globales Koordinatensystem zu verwenden. Beide werden durch Transformationsfunktionen miteinander verknüpft. Werden für diese Transformation dieselben Formfunktionen wie für den Verformungsansatz benutzt, so spricht man von isoparametrischen Elementen, bei Funktionen niedrigeren bzw. höheren Grades von sub- bzw. superparametrischen Elementen [12].

3.5 Das Prinzip vom Minimum des Potentials

Die Lösungsgrößen (z.B. Knotenpunktverschiebungen) werden nun aufgrund der Bedingung ermittelt, dass im gesuchten Gleichgewichtszustand das Gesamtpotential ein Minimum hat. Das Prinzip vom Minimum des Potentials bildet eine der möglichen Variationsmethoden zur direkten Bestimmung von Steifigkeitsgleichungen finiter Elemente [12, 24]. Dieses Minimum wird mit Hilfe einer Variationsformulierung als

$$\delta\Pi = 0 \quad [37]$$

ausgedrückt. $\delta\Pi$ stellt dabei die Variationsableitung δ des Funktionals Π (ein Funktional ist eine Funktion von Funktionen) der Summe der inneren und der äußeren virtuellen Arbeit dar. Für den statischen Fall gilt somit:

$$\delta A^{(a)} + \delta A^{(i)} = 0 \quad [38]$$

Allgemein sei hier noch angemerkt, dass die Natur immer danach bestrebt ist, den Zustand der geringsten Energie zu erreichen, d.h. dieser ist am stabilsten. Daher ist eine, beinahe jede Fragestellung naturwissenschaftlichen Charakters nach dem "Warum" klärende, Antwort: "Weil es energetisch günstiger ist."

3.6 Implizite vs. explizite Problemlösung

In dieser Arbeit werden beide von Softwarepaket ABAQUS angebotenen Lösungsvarianten verwendet. Dies ist einerseits ABAQUS/Standard und andererseits ABAQUS/Explicit. Hinter "Standard" verbirgt sich ein implizites Lösungsverfahren, während bereits aus dem Namen "Explizit" das dabei verwendete Lösungsverfahren ersichtlich ist. Die beiden Verfahren unterscheiden sich wie folgt. Die Variation erfolgt, für die vorliegende Problemstellung, über

die auftretenden Verschiebungen. Daher lautet das Prinzip der virtuellen Arbeit für diesen Fall [22]:

$$\delta\Pi = (\underline{F} - \underline{P})\delta\underline{u} = \left((\underline{m} \cdot \ddot{\underline{u}} + \underline{c} \cdot \dot{\underline{u}} + \underline{k} \cdot \underline{u}) - \underline{P} \right) \delta\underline{u} = 0 \quad [39]$$

$$\underline{F} = \underline{m} \cdot \ddot{\underline{u}} + \underline{c} \cdot \dot{\underline{u}} + \underline{k} \cdot \underline{u} \quad [40]$$

Dabei ist \underline{F} der innere und \underline{P} der äußere Kraftvektor, \underline{m} die Massenmatrix, \underline{c} die Dämpfungsmatrix, \underline{k} die Steifigkeitsmatrix (\sim Federkonstante), \underline{u} der Verschiebungsvektor mit den zugehörigen Ableitungen Beschleunigung $\ddot{\underline{u}}$ und Geschwindigkeit $\dot{\underline{u}}$. Die Lösung erfolgt durch zeitliche Diskretisierung und Iteration. Iteration (lat.: ‚iterare‘ = "wiederholen") ist ein Begriff aus der numerischen Mathematik und bezeichnet eine Methode, sich der Lösung eines Rechenproblems schrittweise (mit Schrittweite Δt), aber zielgerichtet anzunähern. Sie besteht in der wiederholten Anwendung desselben Rechenverfahrens. Man errechnet dabei, je nach verwendetem Verfahren (Zentral-, Vorwärts- bzw. Rückwärtsdifferenz.) die Lösung zum jeweiligen Zeitpunkt aus den vorangegangenen Lösungen. Die Ergebnisse eines Iterationsschrittes werden als Ausgangswerte des jeweils nächsten Schrittes genommen bis das Ergebnis (beziehungsweise Veränderung einer Bestandsgröße) sich nahe genug der exakten Lösung genähert hat. Ein Beispiel dafür ist das Newton-Verfahren. Manchmal setzt man den nächsten Schritt aus den Ergebnissen der vorherigen zwei Schritte (oder von noch mehr Schritten) an, zum Beispiel bei der Regula Falsi [12]. Abhängig davon, zu welchem Zeitpunkt t oder $t+\Delta t$ man das diskretisierte Gleichungssystem anschreibt, unterscheidet man zwischen impliziter (Zeitpunkt $t+\Delta t$) und der expliziter (Zeitpunkt t) Methode. Für die explizite Methode ergibt sich somit [23]

$$\underline{F}_t = \underline{m} \cdot \ddot{\underline{u}}_t + \underline{c} \cdot \dot{\underline{u}}_t + \underline{k} \cdot \underline{u}_t \quad [41]$$

wo hingegen für die implizite Methode folgendes gilt:

$$\underline{F}_{t+\Delta t} = \underline{m} \cdot \ddot{\underline{u}}_{t+\Delta t} + \underline{c} \cdot \dot{\underline{u}}_{t+\Delta t} + \underline{k} \cdot \underline{u}_{t+\Delta t} \quad [42]$$

Allerdings ist man in beiden Fällen an der Lösung $\underline{u}_{t+\Delta t}$ interessiert. Man wird anhand der folgenden Kapitel erkennen, dass während bei der expliziten Lösungsmethode die gesuchte

Lösungsgröße des jeweiligen Zeitschritts ($u_{t+\Delta t}$) nur auf der linken Gleichungsseite vorkommt, also explizit (ausdrücklich, genau) mittels vergangener Lösungen ($u_t, u_{t-\Delta t}$) ausgedrückt werden kann, kommt bei der impliziten Lösungsmethodik die Lösungsgröße des jeweiligen Zeitschritts ($u_{t+\Delta t}$) auf beiden Seiten der Gleichung vor, ist also "von sich selbst" (implizit = innewohnend) abhängig und es bedarf zur Lösung daher weiterer mathematischer Ansätze und Umformschritte. Solche Problemstellungen können meist nur mehr iterativ gelöst werden kann. Die folgenden Kapitel sollen dies veranschaulichen.

3.6.1 Implizite Problemformulierung

Für den Zeitpunkt $t+\Delta t$ werden die Geschwindigkeits- und die Beschleunigungsgleichung als Funktionen f bzw. g der Verschiebungen zu verschiedenen Zeitpunkten bis $t+\Delta t$ berechnet [23]:

$$\underline{\dot{u}}_{t+\Delta t} = \frac{1}{\Delta t} \cdot f(\underline{u}_{t+\Delta t}, \underline{u}_t, \underline{u}_{t-\Delta t}, \underline{u}_{t-2\Delta t}, \dots) \quad \underline{\ddot{u}}_t = \frac{1}{\Delta t^2} \cdot g(\underline{u}_{t+\Delta t}, \underline{u}_t, \underline{u}_{t-\Delta t}, \underline{u}_{t-2\Delta t}, \dots)$$

Damit ergibt sich die Bewegungsgleichung:

$$\underline{F}_{t+\Delta t} = \underline{m} \cdot \frac{1}{\Delta t^2} \cdot f(\underline{u}_{t+\Delta t}, \underline{u}_t, \underline{u}_{t-\Delta t}, \underline{u}_{t-2\Delta t}, \dots) + \underline{c} \cdot \frac{1}{\Delta t} \cdot g(\underline{u}_{t+\Delta t}, \underline{u}_t, \underline{u}_{t-\Delta t}, \underline{u}_{t-2\Delta t}, \dots) + \underline{k} \cdot \underline{u}_{t+\Delta t} \quad [43]$$

Daraus lässt sich nun aus den vorangegangenen Zeitschritten die Lösung $\underline{u}_{t+\Delta t}$ berechnen, wobei zu erkennen ist, dass die Lösungsgröße implizit gegeben ist und von den Funktionen f sowie g abhängt. Zu seiner Berechnung muss also in jedem Iterationsschritt ein (je nach Art der Funktionen f und g) lineares oder nichtlineares Gleichungssystem gelöst werden. Die berechneten Werte stellen dann Approximationen an die tatsächlichen Werte der exakten Lösung des Anfangswert-Problems dar. Es gibt für die implizite Methode keine Einschränkungen an die Zeitschrittweite aufgrund von Stabilitätsüberlegungen, der Zeitschritt kann groß gewählt werden, was den Aufwand des iterativen Lösens des Gleichungssystems in jedem Schritt wettmacht. Dies umso mehr, da diese Lösungsmethodik eine hohen Konvergenzordnung (die Geschwindigkeit, mit der sich die Glieder einer konvergenten Folge dem Grenzwert nähern) besitzt.

3.6.2 Explizite Problemformulierung

Stellt man aus den Verschiebungen \underline{u} die Geschwindigkeits- bzw. Beschleunigungsgleichung zum Zeitpunkt t auf, ergibt sich [23]:

$$\dot{\underline{u}}_t = \frac{1}{2 \cdot \Delta t} \cdot (\underline{u}_{t-\Delta t} + \underline{u}_{t+\Delta t}) \quad \ddot{\underline{u}}_t = \frac{1}{\Delta t^2} \cdot (\underline{u}_{t-\Delta t} - 2 \cdot \underline{u}_t + \underline{u}_{t+\Delta t})$$

und somit:

$$\begin{aligned} \underline{F}_t &= \underline{m} \cdot \frac{1}{\Delta t^2} \cdot (\underline{u}_{t-\Delta t} - 2 \cdot \underline{u}_t + \underline{u}_{t+\Delta t}) + \underline{c} \cdot \frac{1}{2 \cdot \Delta t} \cdot (\underline{u}_{t-\Delta t} + \underline{u}_{t+\Delta t}) + \underline{k} \cdot \underline{u}_t \\ \underline{F}_t &= \underline{m} \cdot \frac{1}{\Delta t^2} \cdot \underline{u}_{t-\Delta t} - \underline{m} \cdot \frac{1}{\Delta t^2} \cdot 2 \cdot \underline{u}_t + \underline{m} \cdot \frac{1}{\Delta t^2} \cdot \underline{u}_{t+\Delta t} + \underline{c} \cdot \frac{1}{2 \cdot \Delta t} \cdot \underline{u}_{t-\Delta t} + \underline{c} \cdot \frac{1}{2 \cdot \Delta t} \cdot \underline{u}_{t+\Delta t} + \underline{k} \cdot \underline{u}_t \end{aligned}$$

Damit ergibt sich:

$$\underline{u}_{t+\Delta t} \cdot \left(\frac{1}{\Delta t^2} \cdot \underline{m} + \frac{1}{2 \cdot \Delta t} \cdot \underline{c} \right) = \underline{F}_t - \left(\underline{k} - \frac{1}{\Delta t^2} \cdot \underline{m} \right) \cdot \underline{u}_t - \left(\frac{1}{\Delta t^2} \cdot \underline{m} - \frac{1}{2 \cdot \Delta t} \cdot \underline{c} \right) \cdot \underline{u}_{t-\Delta t} \quad [44]$$

Die Lösungsgröße $\underline{u}_{t+\Delta t}$ kann somit explizit aus vergangener Lösungen ($\underline{u}_t, \underline{u}_{t-\Delta t}$) berechnet werden. Die Schrittweite Δt kann, da man sich nur auf vergangene Werte bezieht, variabel gewählt werden. Eine sinnvolle Veränderung der Schrittweite setzt einen Algorithmus zur Schrittweiten-Steuerung voraus, der den Fehler im aktuellen Schritt abschätzt und dann die Schrittweite für den nächsten Schritt dementsprechend wählt. Die explizite Methode ist allerdings nur bedingt stabil. Die Stabilität ist nur dann gewährleistet, wenn der Zeitschritt Δt kleiner ist als die Zeit, die eine Schallwelle (c_d Wellenausbreitungsgeschwindigkeit) zur Ausbreitung durch das kleinste Element (L^e charakteristische Elementlänge) in der Netzstruktur benötigt.

$$\Delta t_{stable} \leq \frac{L^e}{c_d} \quad [45]$$

Für eine Berechnung heißt dies, dass mit zunehmender Netzfeinheit auch die Diskretisierungsschrittweite verkleinert werden muss. In ABAQUS stehen beide Lösungsmethodiken (implizit/STANDARD und explizit/EXPLICIT) zur Verfügung und wie den folgenden Kapiteln zu entnehmen ist, werden für die vorliegende Arbeit auch beide, wenn auch in unterschiedlichem Ausmaß, verwendet.

4 Vorgehensweise – Simulationsschritte – Modellentwicklung

In diesem Kapitel wird nun ein Überblick über die verwendeten Ressourcen (Hilfsmittel) gegeben und das zeitliche Vorgehen erläutert. Aufgrund der zahlreichen Variationen werden nur einzelne, repräsentative Simulationsergebnisse gezeigt, ohne näher auf die einzelnen Simulationsparameter im Detail einzugehen.

4.1 Verwendete Hard- und Software

Aufgabenstellung war es, die Indentation mittels des Softwarepaketes ABAQUS zu simulieren. Die dafür zur Verfügung stehende Hard- und Software wird nachstehenden beschrieben.

4.1.1 Arbeitsplatzrechner

Zur Anwendung kommt ein, in **Abbildung 29** gezeigter, Big-Tower mit ASUS-P4 Motherboard und einem Intel Pentium-IV (Northwood) Socket-478 Prozessor mit 2.4 GHz und 533FSB sowie 1024MB RAM. Um die Datensicherung zu gewährleisten werden 4 separate 160GB Datenfestplatten im RAID 1 (mirror) betrieben. Zur Anwendung kommt sowohl das Betriebssystem Windows2000 als auch WindowsXP/home.



Abbildung 29: Der verwendete Arbeitsplatzrechner. Gut zu erkennen sind die Wechselrahmen für die im RAID verwendeten Festplatten sowie die für lange Rechenzeiten notwendige Kühlung (IP-Adresse: 213.229.22.92).

4.1.2 Universitäts-Applicationsserver "alpha1"

Bei dieser Maschine, die in **Abbildung 30** abgebildet ist, handelt es sich um einen HP/Compaq Alphaserver ES 45 mit 4 Stück 1000MHz Alpha EV68 Prozessoren und 16GB

RAM sowie 200GB Festplattenkapazität mit dem Betriebssystem TRU64 (Batch Substem Platform). Aufgrund der programmtechnischen Auslastung ist die ABAQUS Nutzung dabei auf 2 CPU limitiert, welche sich alle Anwender teilen. Daher muss während der Arbeit darauf geachtet werden, dass alle Nutzer ihre Rechenjobs abarbeiten lassen können.



Abbildung 30: Der Applikationsserver der Montanuniversität (zalpha1.unileoben.ac.at, IP-Adresse: 193.171.87.200).

4.1.3 Universitäts-Applikationsserver "HP®-Cluster"

Es wird versucht, die letzten Simulationen für die vorliegende Arbeit auf diesem, die "zalpha1" ersetzende Clusterlösung zu rechnen.

4.1.4 SMMP Shared-Memory-Applikationsserver "SGI®-Altix®350"

Da die Komplexität der Simulationen im Laufe der Arbeit ständig reduziert werden muss (**Kapitel 4.4**) können noch einige Simulationen mit dem Shared-Memory-Applikationsserver des Lehrstuhl "Simulation und Modellierung Metallurgischer Prozesse" (SMMP) durchgeführt und somit die hardwarebedingten Grenzen etwas erweitert werden. Die in **Abbildung 31** gezeigte Altix®350 hat 16 1.5GHz Intel® Itanium2® CPU's welche in Ringarchitektur angeordnet sind mit 32GB RAM.



Abbildung 31: Der SGI®-Altix®350 Shared-Memory-Server des Lehrstuhl für "Simulation und Modellierung metallurgischer Prozesse" (SMMP) (IP-Adresse: 192.168.1.204).

4.1.5 Die Netzwerkeinrichtung

Um es dem Autor zu ermöglichen, ABAQUS auch von seinem heimischen Arbeitsplatz zu verwenden, muss einerseits ein Internetanschluss mit statischer IP-Adresse (diese wird vom Lizenzserver während der Arbeit ständig abgefragt) eingerichtet werden und andererseits diese IP-Adresse am Universitäts-Lizenzserver zur Beziehung einer Lizenz freigeschaltet werden. Dies stellt einen nicht zu verachtenden Arbeitsaufwand dar, da ein solcher Zugang von der Montanuniversität erstmals für Nutzer außerhalb dieser zur Verfügung gestellt wird. Auch die Benutzungsrechte auf dem Universitäts-Applikationsserver ("zalpha1") werden geschaffen.

4.1.6 ABAQUS 5.8, 6.3, 6.4, 6.5 und 6.6

Für eine FEM-Analyse mittels ABAQUS hat der Anwender ein Inputfile (*.inp) zu verfassen (siehe **Anhang a**) in welchem sämtliche Knoten- und Elementdefinitionen als auch Befehls- und Definitionszeilen nach genau vorgegebenen Formaten programmiert werden müssen. Ausgliederung und Laden von Programmteilen während der Programmabarbeitung ist allerdings möglich und kommt auch zur Anwendung, so wird beispielsweise die Geometrie- bzw. Netzgenerierung (Knoten- und Elementdefinitionen) teilweise in die gesonderte Datei "KnotenElemente.i" (siehe **Anhang b**) ausgegliedert. ABAQUS stellt grundsätzlich die zwei Analysemethoden standard (= implicit) und explicit zur Verfügung, wobei sich die Programmsyntax und die Programmbefehle unterscheidet. Der Programmstart erfolgt über eine einfache Befehlszeile, etwa:

abaqus job=diplomarbeit.inp interactive

"abaqus" ist dabei der Programmaufrufbefehl, "job=" weist ABAQUS auf die folgende Inputdatei mit dem Namen "diplomarbeit.inp" hin, der folgende, zusätzliche Befehle (davon gibt es etwa 20 weitere) weist ABAQUS an während der "Abarbeitung" eine Bildschirmausgabe der *.log-Datei (Programmschritte,...) durchzuführen. Dies ist anzuraten, da daran gesehen werden kann, ob das Programm wunschgemäß arbeitet. Näheres zur Syntax kann dem **Anhang c**) entnommen werden, da dort Wissenswertes in Kommentarzeilen eingefügt wurde. Die Kalkulation, also Rechnung wird "Processing" genannt, unter Pre- (= Vor-) Processing versteht man die Vorbereitungsarbeiten wie Geometrierstellung, Vernetzung, Materialzuordnung,... während man unter Post- (= Nach-) Processing die Ergebnisausgabe bzw. Nachbearbeitung versteht. Bei den zur Erstellung dieser Arbeit verwendeten Versionen ABAQUS 6.3 (zu Beginn der Arbeit am Arbeitsplatzrechner, der

Autor benutzte bis dahin ABAQUS 5.8), ABAQUS 6.4 (auf dem Applikationsserver) bzw. ABAQUS 6.5 (am Arbeitsplatzrechner) und ABAQUS 6.6 (auf dem neuen Applikationsserver sowie dem SMMP-Server) steht mit dem Programmteil ABAQUS/CAE auch ein GUI (Grafik User Interface) zur Verfügung, mit welchem das Pre- als auch das Postprocessing durchgeführt werden kann. Das GUI hilft dabei, die *.inp Datei zu erstellen. Bis zu dieser Version von ABAQUS wurden spezielle Softwarepakete (wie etwa MSC/PATRAN) zum Preprocessing und andere Programme (wie etwa ABAQUS/Viewer) zum Postprocessing verwendet.

4.1.7 Fortran F77 und Microsoft Visual C++ 6.0

Es gibt zahlreiche Möglichkeiten mittels Preprocessingprogrammen eine Geometrie zu erstellen und diese zu vernetzen. Gemeinsamer Nachteil all dieser Programme ist, dass man ein unstrukturiertes Netz (unstructured mesh) anstelle eines strukturierten Netzes (structured mesh) erhält und daher anhand einer Zellennummer nicht die Nachbarzellennummern eruierbar sind. Ebenso kann man anhand der Zellennummer eines unstrukturierten Netzes nicht die exakte Position der Zelle bestimmen. Möchte man bestimmten Gebieten bestimmte Eigenschaften zuweisen (z.B. Mineralmaterialkennwerte für Mineralplättchen) und berücksichtigt die Tatsache, dass in ABAQUS Knoten- und Element-Sets definiert werden können, benötigt man jedoch strukturierte Netze um solche Sets zu erstellen und diesen die Werte zuzuordnen. Eine in dieser Arbeit inkludierte Möglichkeit stellt die Netzgenerierung mittels eines selbst erstellten Programmes dar, das alle Geometriepunkte und Elemente strukturiert erstellt und nummeriert sowie in Sets gruppiert sowie diesen bestimmte Werte (Materialkennwerte, Elementtyp,...) zuweist.

Fortran77 wird benutzt, da diese Programmiersprache wissenschaftlich etabliert und dem Programmpaket ABAQUS aus dessen Vergangenheit heraus zuordenbar ist. Eine weitere Möglichkeit stellt die Programmierung mittels Microsoft Visual C++ dar. Diese Programmiersyntax ist weiter entwickelt und stellt bessere Programmkontrollmechanismen zur Verfügung. So kann der Quellcode Zeile für Zeile abgearbeitet werden, was die Nachvollziehung der Änderung einzelner Programmvariablen ermöglicht. Da die Programmierung mittels des Fortran77-Compilers erfolglos bleibt (nach mehrwöchigen ergebnislosen Versuchen wird festgestellt, dass der Compiler (Freeware aus dem Internet) leider fehlerhaft ist) wird die Generierung der Datei "KnotenElemente.i" mit C++ durchgeführt (C++ Quellcode sowie ein Auszug aus der Netzgenerierungsdatei siehe **Anhang b**)).

4.1.8 Exceed und Exceed3D (Hummingbird Connectivity V7.1)

Wie sehr viele andere wissenschaftliche Spezialsoftwarepakete ist auch ABAQUS standardmäßig für das Betriebssystem Linux geschrieben. Um solche Programme auf einer Windows-Oberfläche verwenden zu können, wird eine Software benötigt, die eine Linux-Benutzerplattform auf Windows generiert. Auf dieser wird das Softwarepaket ausgeführt. Ein Programm, welches diese Generierung übernimmt ist das verwendete Programm Exceed der Firma Hummingbird. Zur korrekten Darstellung von 3D-Geometrie innerhalb der GUI wird auch Exceed3d benötigt. Zur Anwendung kommt dabei die Version 7.1.

4.1.9 SSH® Secure Shell™

Für den Datentransfer zwischen dem Arbeitsrechner (Windows) und dem Applikationsserver zalpha1 (Linux) wird ein Programm benötigt, das die Dateistruktur beider Betriebssysteme beherrscht. Dieses Programm stellt eine, wie in **Abbildung 32** gezeigt, GUI zur Verfügung mittels welchem man mit einfachen Kopierbefehlen ("drag and drop") die benötigten Dateien von einem auf den anderen Rechner übertragen kann. Sinnvoll ist dies vor allen aufgrund der Tatsache, dass das Pre- sowie Postprocessing mittels ABAQUS/CAE durchgeführt wird, dieses auf dem Applikationsserver aber nicht zur Verfügung steht, um die Serverperformance mit diesen Prozessen nicht zu belasten. Auch steht ABAQUS/CAE nicht für alle Prozessortypen (z.B. auch nicht für die Intel Itanium2 des SMMP-Servers) zur Verfügung. Zur Anwendung kommt die Version SSH Secure Shell for Workstations 3.2.

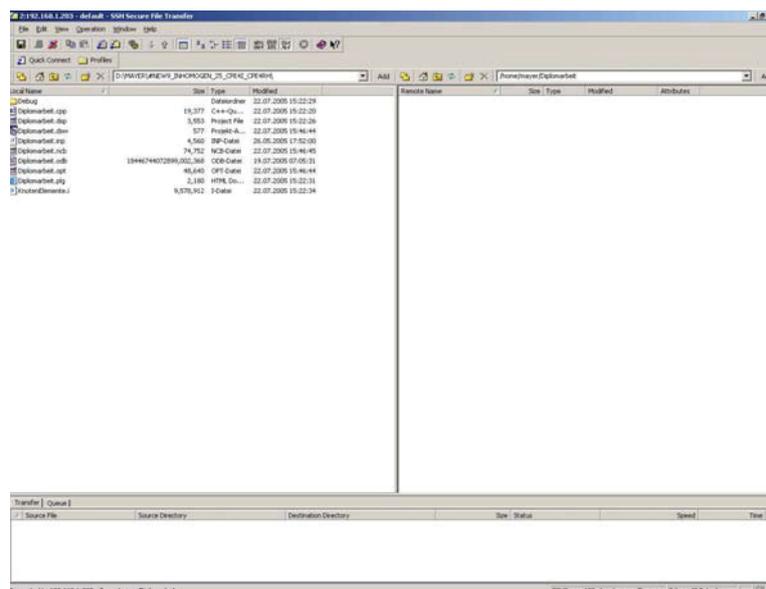


Abbildung 32: Das einfach zu bedienende GUI des SSH Secure Shell arbeitet ähnlich wie der Windows Explorer. Besonders sei hier auf die aufgrund der Dateigröße bereits fehlerhafte Größenanzeige der *.odb Datei hingewiesen (18446744072899.002.368!!).

4.1.10 ORIGIN, MS-Excel und MS-Word

Zur Auswertung der Ergebnisse wird neben der Tabellenkalkulationssoftware Microsoft Excel 2000 auch das Softwarepaket ORIGIN 7G SR2 der Firma OriginLab Corporation verwendet. Dabei handelt es sich um ein aus einem reinen Statistikanalyseprogramm weiterentwickeltes Grafik- und Datenanalyseprogramm mit dem man beispielsweise diverse Fittkurven unterschiedlichen Gleichungstyps erstellen kann. Die Arbeit wurde mittels Microsoft Word 2000 sowie Microsoft Word 2003 zusammengefasst.

4.2 Die gewählten Abmessungen und Materialdaten

Die gewählten, vereinfachten Materialdaten sind in **Abbildung 33** dargestellt und können wie folgt zusammengefasst werden:

- Das Mineral hat einen E-Modul von $E_{Mineral} = 114\text{GPa}$, ist isotrop und hat ein ideal elastisches Materialverhalten.
- Die Matrix zwischen den Plättchen (Kollagen) wird, als isotropes Material mit einem E-Modul von $E_{Kollagen} = 1.5\text{GPa}$ beschrieben, welches ein rein elastisch-plastisches Materialverhalten aufweist mit einem Übergang bei $\varepsilon = 3\%$.

Da die Poissonzahlen der einzelnen Komponenten nicht bekannt sind, wird für das Mineral $\nu_{Mineral} = 0.30$ und für die Matrix $\nu_{Kollagen} = 0.30$ verwendet. Die geometrischen Dimensionen werden mit einer Länge von $l = 100\text{nm}$, einer Breite von $b = 40\text{nm}$ und einer Dicke von $d = 2.5\text{nm}$ angenommen. Der Abstand der Plättchen in der Ebene (Spaltbreite) wurde mit $a = 2.5\text{nm}$ und die Matrixdicke zwischen den einzelnen Ebenen mit $b = 2.5\text{nm}$ fixiert. Im Laufe der Arbeit wird versucht, mittels Änderung dieser Geometrieabstände die Konvergenz positiv zu beeinflussen. Der Indenter erhält die Materialdaten von Diamant mit $E_{Indenter} = 1140\text{GPa}$ und $\nu_{Indenter} = 0.07$. Die Indenterform stellt eine Berkovich-Pyramide, wie in **Abbildung 34** gezeigt, dar, die einen Spitzenradius von $r = 100\text{nm}$ aufweist.

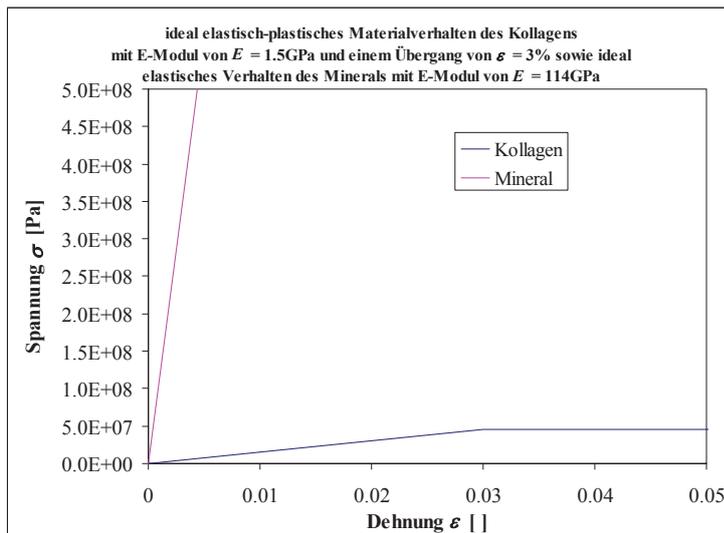
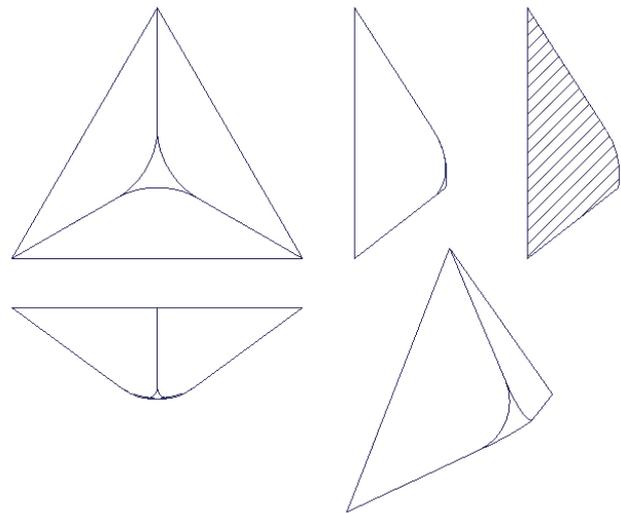


Abbildung 33: Die Spannungs-Dehnungsverläufe der verwendeten Materialien. Man erkennt die großen Unterschiede im E-Modul (Steigung). Das ideal plastische Materialverhalten des Kollagens setzt bei 3% Dehnung ein.

Abbildung 34: Skizze einer Berkovich-Pyramidenspitze in Grund-, Auf- und Seitenriss sowie einer Isometrieansicht und ein Schnitt durch die Pyramidenspitze.



4.3 Modellaufbau

Für die vorliegende Arbeit werden die Mineralplättchen rechteckig und in einer Ebene regelmäßig angeordnet angenommen, wobei die Plättchen der nächsten Ebene so versetzt sind, dass jeweils der Mittelpunkt eines Plättchens genau über dem Mittelpunkt der Zwischenräume der darunter befindlichen Ebene liegt. Dabei wird davon ausgegangen, dass die Plättchen regulär, identisch und ideal gepackt sind. Um sich diese dreidimensionale Anordnung besser vorstellen zu können wurde mit Hilfe des 3D-Zeichenprogrammes Autodesk INVENTOR 5™ eine, wie in **Abbildung 35** gezeigt, Zeichnung erstellt. **Abbildung 36** zeigt die mittels des Zeichenprogramms Autodesk AUTOCAD 2005 erstellte 2D-Anordnung.

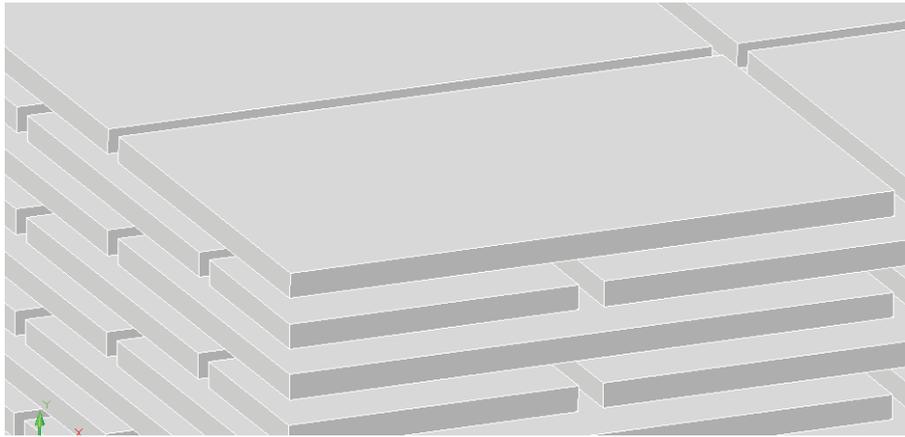


Abbildung 35: Skizze der 3D-Mineralplättchenanordnung. Die Mineralplättchen haben die Abmessungen $l = 100\text{nm}$, $b = 40\text{nm}$ und $d = 2.5\text{nm}$.

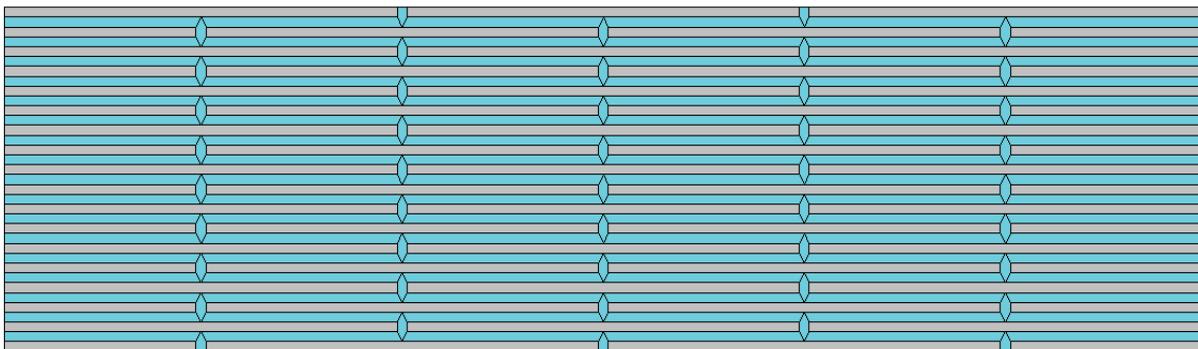


Abbildung 36: Skizze der 2D-Mineralplättchenanordnung. Die Mineralplättchen haben die Abmessungen $l = 100\text{nm}$ und $d = 2.5\text{nm}$.

Indentationen erreichen in der Praxis eine Tiefe von 100nm - 500nm (manchmal werden Messungen bis 1000nm durchgeführt) bei verwendeten Kräften von 0.2mN - 2mN. Die Hertz'sche Theorie besagt, dass um die Spannungsfelder korrekt erfassen zu können, ein Volumen von 9-facher Tiefe und 7-facher Weite zu berücksichtigen ist.

4.4 Modellvernetzung

Bei der räumlichen Diskretisierung werden, wie in **Kapitel 3.3** beschrieben, einzelne Gebiete (domains) der Geometrie in finite Teilgebiete geteilt. Diese Teilgebiete nennt man Elemente. Prinzipiell unterteilen sich diese aufgrund ihrer Dimension (2D bzw. 3D-Elemente) bzw. ihrer Elementdefinition (Balken- (truss), Kontinuums-(continums), Strahl-(beam), Schalenelemente(shell) usw.) die sich vor allem in der durch sie aufgrund ihrer Freiheitsgrade übertragbaren Kräfte bzw. Momente unterscheiden. (Stabelemente können beispielsweise nur axiale Schub- und Druckkräfte übertragen usw.) In ABAQUS stehen eine große Anzahl verschiedener Elemente zur Verfügung, einen Auszug zeigt **Abbildung 37**.

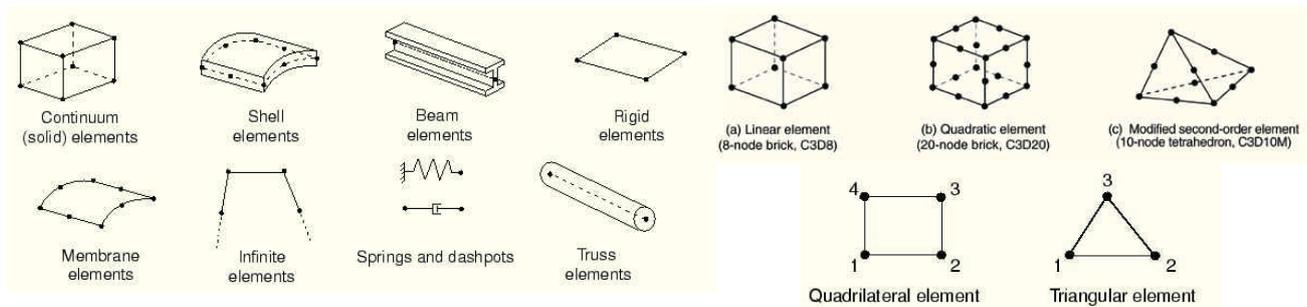


Abbildung 37: Eine Auswahl der in ABAQUS zur Verfügung stehenden Elementtypen. Gut zu erkennen ist die unterschiedliche Anzahl von Knoten, je nach Ansatzfunktion (linear, quadratisch) [8].

Als weitere Unterscheidung wird die Arte der Ansatzfunktion (linear, quadratisch,...), wie bereits in **Kapitel 3.2** beschrieben, und die rechnerische Behandlung innerhalb von ABAQUS (reduzierte Integration, Inkompressibilität, Axialität,...) herangezogen. Daraus ergeben sich die von ABAQUS verwendeten Elementtypenbezeichnungen wie sie auch innerhalb des Quelltextes im **Anhang c)** angeführt und erklärt sind. Jedes Element hat eine bestimmte Anzahl von Knoten, welche die Elementgeometrie aufbauen. Je nach Elementtyp kommen neben den Eckpunkten auch noch weitere Punkte (z.B. Elementmittelpunkte als Stützstellen für quadratische Ansatzfunktionen der Elementdefinition,...) hinzu. Die Wahl des Elementtyps ist von entscheidender Bedeutung für die Güte der erhaltenen Resultate. Dies veranschaulicht **Abbildung 38**.

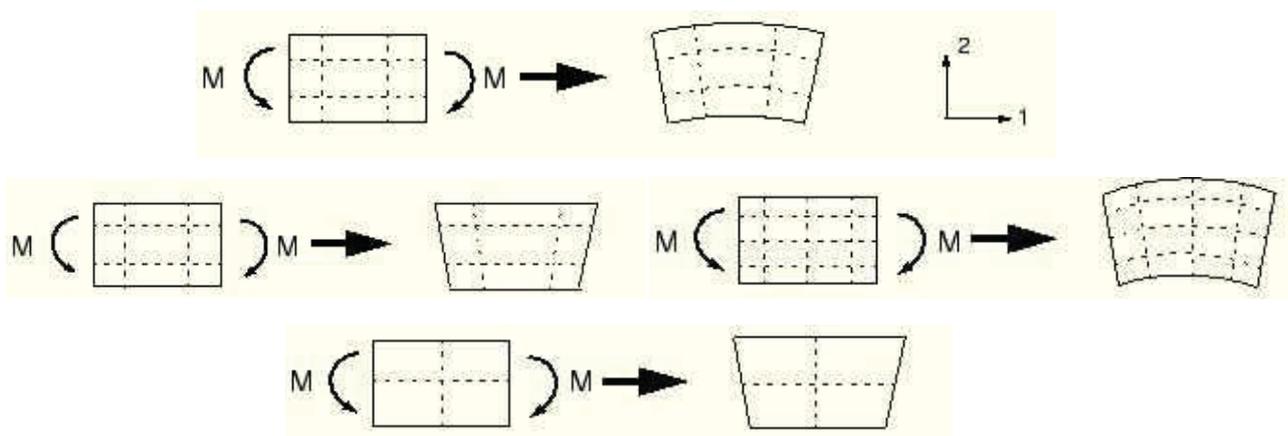


Abbildung 38: Verformung aufgrund des angelegten Momentes M . Exakte (analytische) Lösung (oben), Simulationsergebnis für volle Integration mit linearer Ansatzfunktion (z.B. CPS4, Kontinuumelemente (C..) mit ebenem Spannungszustand (Plain Stress ..PS..)) (Mitte links) sowie quadratischer Ansatzfunktion (z.B. CPS8) (Mitte rechts) als auch für reduzierte Integration mit linearer Ansatzfunktion (z.B. CPS4R) (unten) [8].

Die Gesamtheit von Elementen (Zellen) und diese beschreibende Knoten nennt man Gitter (grid). Fügt man noch Randbedingungen sowie Zelleneigenschaften (Materialkennwerte,...) hinzu, wird daraus ein Netz (mesh).

4.4.1 Voll (3D) Simulation – Netzgenerierung durch ABAQUS/CAE

Wenn man die in **Kapitel 4.3** beschriebenen Informationen (laut Hertz 9-fache Tiefe und 7-fache Weite) kombiniert und vereinfacht von einem Indentationswinkel von 45° ausgeht (damit ist der Indentationsdurchmesser etwa gleich der doppelten Indentationstiefe), muss man ein Simulationsvolumen von $900 \times 1400 \times 1400 \text{ nm}$ (für 100 nm Indentationstiefe) bzw. $4500 \times 7000 \times 7000 \text{ nm}$ (für 500 nm Indentationstiefe) modellieren. Berücksichtigt man die angegebenen Mineralplättchen- und Kollagenabmessungen ergibt sich eine Plättchenlagenanzahl von:

$$n = \frac{900}{d_{\text{Mineral}} + d_{\text{Collagen}}} = \frac{900}{2.5 + 2.5} = \frac{900}{5} = 180 \text{ Plättchenlagen / Tiefe} \quad [46]$$

$$n = \frac{4500}{d_{\text{Mineral}} + d_{\text{Collagen}}} = \frac{4500}{2.5 + 2.5} = \frac{4500}{5} = 900 \text{ Plättchenlagen / Tiefe} \quad [47]$$

und

$$n = \frac{1400}{l_{\text{Mineral}} + l_{\text{Collagen}}} = \frac{1400}{100 + 2.5} = \frac{1400}{102.5} = 14 \text{ Plättchenlagen / Länge} \quad [48]$$

$$n = \frac{7000}{l_{\text{Mineral}} + l_{\text{Collagen}}} = \frac{7000}{100 + 2.5} = \frac{7000}{102.5} = 70 \text{ Plättchenlagen / Länge} \quad [49]$$

sowie

$$n = \frac{1400}{b_{\text{Mineral}} + b_{\text{Collagen}}} = \frac{1400}{40 + 2.5} = \frac{1400}{42.5} = 34 \text{ Plättchenlagen / Breite} \quad [50]$$

$$n = \frac{7000}{b_{\text{Mineral}} + b_{\text{Collagen}}} = \frac{7000}{40 + 2.5} = \frac{7000}{42.5} = 164 \text{ Plättchenlagen / Breite} \quad [51]$$

Aus diesen Überlegungen ergibt sich somit eine Gesamtzahl der zu modellierenden Mineralplättchen von:

$$N_{100 \text{ nm}} = 180 \times 14 \times 34 = 85680 \text{ Plättchen} \quad [52]$$

$$N_{500 \text{ nm}} = 900 \times 70 \times 164 = 10332000 \text{ Plättchen} \quad [53]$$

wie sie in **Abbildung 39** ausschnittsweise dargestellt.

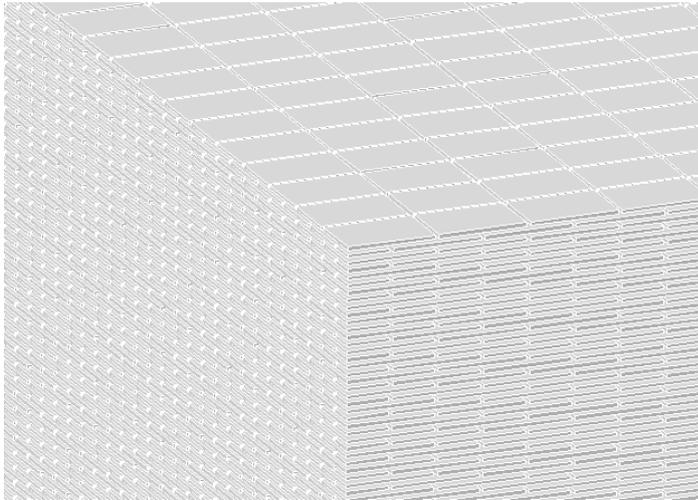


Abbildung 39: Ausschnitt der gesamten Mineralplättchenanordnung in der Rechendomäne unter Berücksichtigung der "Staggered" Anordnung.

Die Plättchenanzahl kann durch Einführung von Symmetrieebenen reduziert werden. Aufgrund der Plättchengeometrie ist für einen kegeligen Indenter als Modellierungsvolumen dennoch ein Raumviertel und für eine Berkovich-Pyramide gar ein Halbvolumen erforderlich. Für die große Indentationstiefe hatte das Zeichenprogramm bereits erhebliche Schwierigkeiten, die errechnete Plättchenanzahl darzustellen. Wenn man berücksichtigt, dass für die Diskretisierung (Vernetzung) jedes Mineralplättchen in zumindest 2 Elemente über die Dicke und 6 Elemente über die Länge sowie 4 Elemente über die Breite zu teilen ist, ergibt sich eine Elementzahl von:

$$N_e = 2 \cdot 6 \cdot 4 = 48 \text{ Elemente / Mineralplättchen} \quad [54]$$

Wenn man nun noch berücksichtigt, dass dies, wie man anhand einer anzustrebenden Netzfeinheit in **Abbildung 40** erkennen kann, noch eine äußerst grobe Vernetzung darstellt und der Kollagenanteil ebenfalls, aufgrund seines geringeren E-Moduls sogar noch erheblich feiner vernetzt werden muss, ergibt sich eine Elementanzahl, die derzeit weder von der Software ABAQUS 6.5 (maximale Elementanzahl liegt bei 999999999 $\sim 10^9$) noch von der zur Verfügung stehenden Rechnerhardware zufrieden stellend behandelt werden kann. Parallel zur Visualisierung mittels INVENTOR, wie in **Abbildung 40** ersichtlich, werden die Möglichkeiten der neuen ABAQUS Version vor allem der neuen GUI ABAQUS/CAE mit in Betracht gezogen.

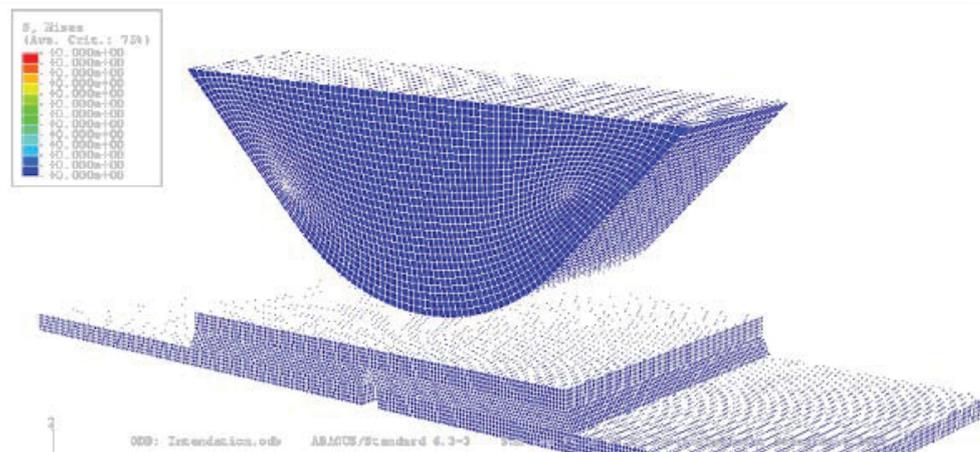


Abbildung 40: Ein stark vereinfachter, erster 3D-Versuch. Der Indenter ist dabei noch mittels Kontinuumelementen vernetzt.

4.4.2 Reduzierte (2D) Simulation – ABAQUS/Standard

Wie im vorangegangenen Kapitel erläutert, muss von einer 3D-Simulation des Problems Abstand genommen werden. Für die Simulation einer Indentation mit einer Tiefe von 500nm wären in 2D etwa 63000 Mineralplättchen (900x70) und das dazwischen liegende Kollagen zu vernetzen was eine Elementanzahl von etwa 3000000 (20 Elemente/Mineralplättchen und eine etwas höhere Anzahl an Kollagenelementen) ergeben würde. Auch diese Anzahl ist für eine Simulation nicht praktikabel. Aus diesem Grund wird eine Indentationstiefe von 100nm in Betracht gezogen. Die Elementanzahl ist hier bei sehr grober Vernetzung etwa 120000 (180x14 Mineralplättchen mal 20 Elemente/Mineralplättchen und einem Faktor für Kollagenelementanzahl von 2.4). Um repräsentative und somit vergleichbare Ergebnisse zu erzielen und die Konvergenz der Rechnung zu gewährleisten muss allerdings eine feinere Vernetzung durchgeführt werden. Die Rechengenauigkeit ist bei sämtlichen Simulationen, da es sich um Näherungsrechnungen handelt, ebenso von der Elementgröße (Güte der Diskretisierung) abhängig wie die Konvergenz. Allerdings muss darauf hingewiesen werden, dass es bei sehr feiner Diskretisierung zur Summierung von Rechengenauigkeiten kommen kann, die bei Wahl eines gröberen Netzes nicht auftreten würden. Solche Fehler fallen bei transienten (zeitabhängigen) Simulationen oft durch ihr periodisches Auftreten auf. Daher ist durch mehrere Simulationen mit unterschiedlich feiner Diskretisierung immer eine Netzfeinheitsunabhängigkeit sicherzustellen.

4.4.2.1 Netzgenerierung mittels CAE

Es wird versucht, eine vorerst kleinere Probegeometrie mittels ABAQUS/CAE zu simulieren.

Abbildung 41 zeigt die verwendete Grafikoberfläche (GUI) von ABAQUS/CAE.

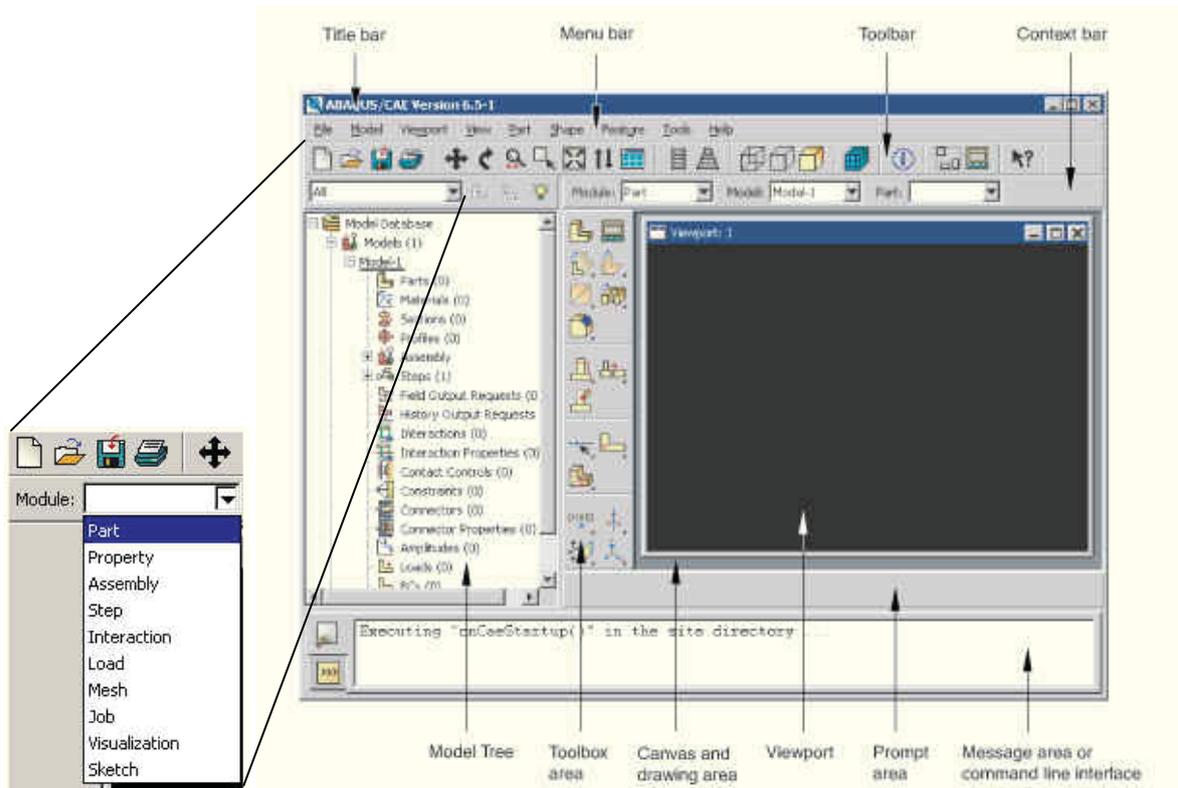


Abbildung 41: Das Grafik-User-Interface, (GUI) von ABAQUS/CAE 6.5. Gut zu erkennen ist, im "pull down Menü" in der linken Bildhälfte, der strukturierte Aufbau und rechts das Modellverzeichnis (model tree), welchem schrittweise gefolgt werden kann. Dies beginnt mit der Bauteilerstellung (Part) und der Materialdefinition (Property) sowie dem Modellaufbau (Assembly) usw. und geht hin bis zum Starten der Analyse (Job - die zur Abarbeitung ("Processing") gegebenen Aufgaben werden als Jobs bezeichnet) und dem Postprocessing (Visualization).

Da noch nicht exakt geklärt ist, welches Material in den Abständen a zwischen den Materialplatten zu finden ist (Kollagen, Wasser, Luft,...), wird vorgesehen, diese "Zwickel" gesondert zu behandeln. Diesen kann man dann dieselben Elementtypen, Materialdaten und -gesetze des Kollagens zuordnen oder durch die Wahl anderer Elementtypen (in ABAQUS gibt es etwa 50 verschiedene Elementtypen) und eines dritten Materialverhaltens den Einfluss auf das Ergebnis untersuchen. Nimmt man beispielsweise Wasser an, würden sich die ebenfalls in ABAQUS verfügbaren Hybrid-Elemente (inkompressibel, ...) mit zugehörigen Materialdaten anbieten. Es zeigt sich, wie man beispielhaft anhand des Simulationsresultats in

Abbildung 42 erkennt, dass eine solche Simulation nicht zielführend ist. Grund dafür ist die Tatsache, dass ein Oberflächenkontaktproblem bei Selbst- als auch Fremdkörperkontakt Konvergenzprobleme ergibt. Dies tritt umso öfter auf, je mehr solcher Kontaktflächen vorhanden sind. Dazu kommt, dass ABAQUS grundsätzlich beträchtliche Probleme hat ein Kontaktproblem zu lösen, wenn beide Kontaktkörper verformbar sind. Hier sei angemerkt, dass Kontakt- und Eindringprobleme generell ein Anwendungsgebiet darstellt, bei dem komplizierteste Differentialgleichungen auftreten und es bei der Verwendung von ABAQUS oft zu Divergenzproblemen kommt. Wenn eine Lösung sich während der Simulation immer weiter von der exakten Lösung entfernt, spricht man von Divergenz. Um ein aussagekräftiges Ergebnis zu erhalten muss die Näherungslösung allerdings zu einem Grenzwert, nämlich der exakten Lösung, konvergieren.

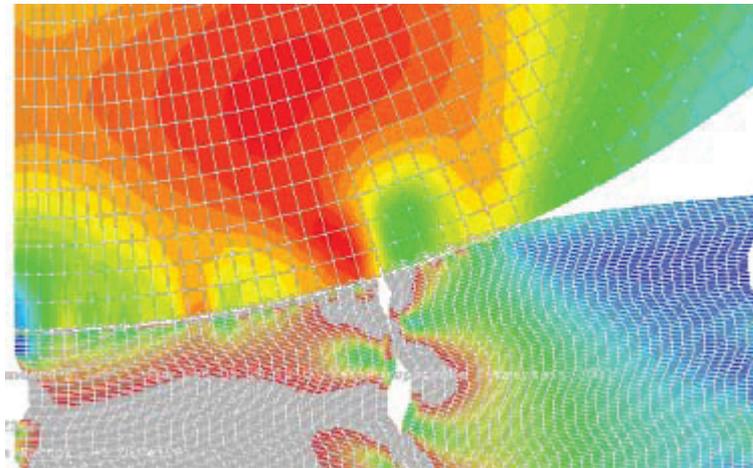


Abbildung 42: Netz und Spannungsfeld einer Indentation mit vernetztem Indenter und nicht gefüllten Modellbereichen (Zwickel) in denen es im Laufe der Simulation zu Selbstkontakt kommt. Zu sehen sind Probleme an der Kontaktfläche sowie stark verzerrte Kollagenelemente, die sogar über die Symmetrieebene hinaus stehen. Die Größenfarbskala wird, da es sich um einen Simulationsversuch handelt nicht angezeigt.

Wie bereits in **Abbildung 42** ersichtlich, kommt es zu extremen Verformungen der Kollagenelemente sowie zu den eben angegebenen Konvergenzproblemen, welche auch "irreguläre" Ergebnisse liefert, wie man verdeutlicht auch anhand von **Abbildung 43**, bei welcher das Netz ausgeblendet ist, erkennt.

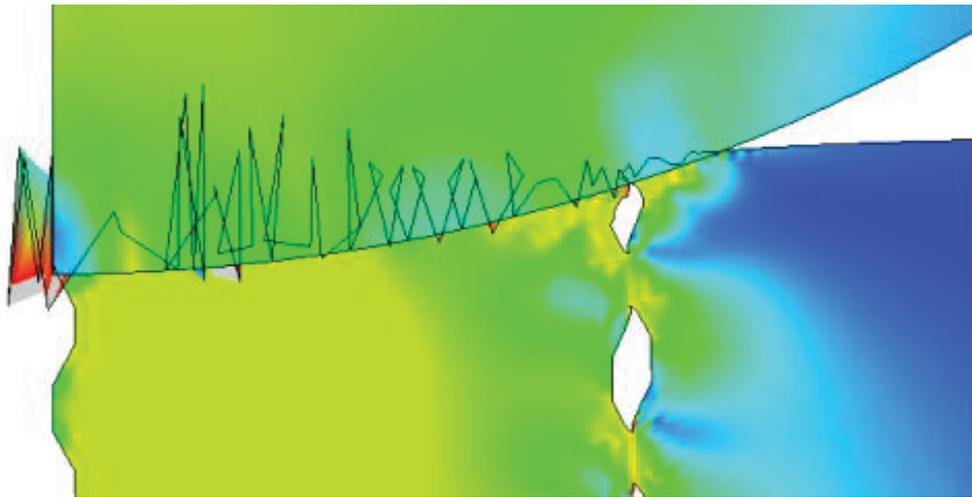


Abbildung 43: Die beim Oberflächenkontakt des vernetzten Indenters mit dem Komposite auftretenden Probleme. Die dabei ersichtliche Abweichung von der "regulären" Oberflächenform (Kontaktfläche) führt zur Divergenz der Simulation.

Daneben sind die Spannungen bei größer werdenden Indentationstiefen, aufgrund der sehr unterschiedlichen E-Moduli, dermaßen unterschiedlich, dass man um Farbverläufe zu erhalten entweder die höchsten, wie in den erwähnten Abbildungen ersichtlich, oder die niedrigsten Spannungen durch selbst definierte Anzeigegrenzen ausblenden muss. Solche Bereiche werden bei den gewählten Anzeigeeigenschaften grau dargestellt (siehe **Abbildung 43**). Würde man diese Einschränkung nicht durchführen, hätte man lediglich blaue und rote Bereiche, wie in **Abbildung 44** ersichtlich. Hier sei darauf hingewiesen, dass die angezeigten Mises-Spannungen von ABAQUS in den Integrationspunkten (aus den dortigen Verschiebungen und dem zugrunde gelegten Materialgesetz) ermittelt werden. Um diese nun flächig darzustellen, stellt ABAQUS mehrere Mittelungsverfahren zur Verfügung, bei denen ein Durchschnittsbildungskriterium (Average Criteria) angegeben werden kann. Dieses ist standardmäßig auf 75% eingestellt und wurde für alle ABAQUS-Postprocessingbilder in der vorliegenden Arbeit auch so verwendet, da eine Änderung dieses Wertes zu keinerlei qualitativer Verbesserung der Ergebnisse führte.

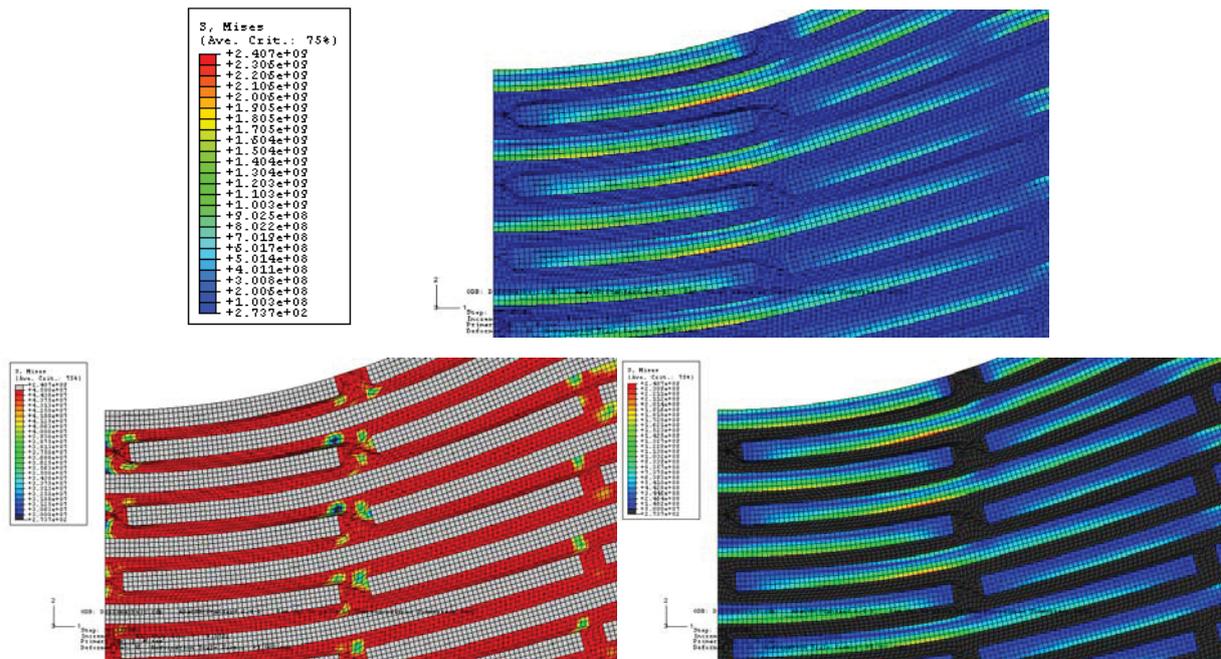


Abbildung 44: Darstellung desselben Ergebnisses mit unterschiedlicher Skalierung. Bei der Skalierung über die Größe aller vorhandenen Spannungen ($2.7e+2$ bis $2.4e+9$, Bild oben) lässt sich das Spannungsfeld innerhalb des Kollagens nicht und das der Mineralplättchen nur gering auflösen. Skaliert man auf die Größe der Spannungen im Kollagen ($3e+7$ bis $4.5e+7$ für den dargestellten Bildabschnitt, Bild unten links) erscheinen die Mineralplättchen zwar grau da die Spannungen generell höher liegen, aber der Farbverlauf innerhalb des Kollagens ist erkennbar. Für eine genauere Farbverlaufsverteilung innerhalb der Mineralplättchen ist eine andere Skalierung zu wählen ($5e+7$ bis $2.4e+9$, Bild unten rechts), wobei die Spannungsverläufe im Kollagen nicht mehr zu sehen sind, da diese aufgrund des E-Modulunterschiedes unterhalb dieses Minimalwertes liegt.

Um die Anzahl an Selbstkontakten (**Abbildung 43**) zu reduzieren wurden die leeren Bereiche zwischen den Mineralplättchen als Kollagenbereiche definiert (**Abbildung 44**). Um eine weitere Fehlerquellen auszuschließen, wurde eine Indenterformanpassung, wie sie in **Abbildung 45** zu sehen ist durchgeführt, um einerseits die Bereiche im Inneren des Indenters, wo es aufgrund des hohen E-Moduls kaum zu Spannungen bzw. Dehnungen kommt, nicht mitzusimulieren und andererseits bei gleich bleibender Elementzahl die Möglichkeit zu erhalten die oberflächennahen Bereiche feiner vernetzten zu können.

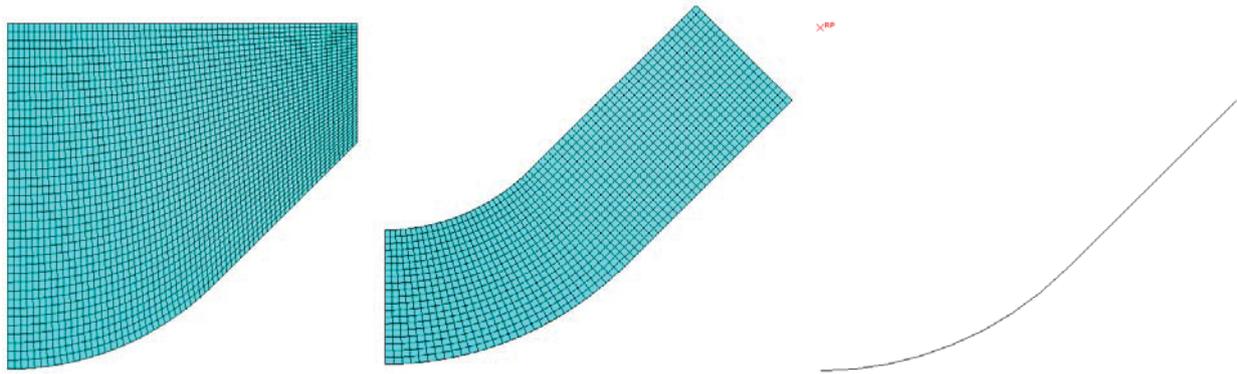


Abbildung 45: Die Indenterform wird mit fortlaufendem Simulationszeitpunkt angepasst. Hier wurde der Indenter für die ersten Simulationen noch voll mitvernetz (links) bzw. um ein gleichmäßigeres Netz zu erhalten als "Schale" mit gleicher Netzgüte simuliert (Mitte). Um die Konvergenz zu verbessern, hat es sich als günstig erwiesen, den Indenter als "analytical rigid" also als starr anzunehmen (rechts). Diese Vereinfachung ist aufgrund der Materialdaten des Indenters ($E_{Indenter} = 1140\text{GPa}$ und $\nu_{Indenter} = 0.07$) durchaus zulässig. Erkennbar ist der für die Definition des starren Indenters notwendige Referenzpunkt (RP).

Ausgehend von diesen Ergebnissen wird der Indenter für die weiteren Simulationen als ideal starr ("rigid body" Definition in ABAQUS) angenommen, was auch die Rechnung dahingehend vereinfacht, dass es nicht nötig ist, den reduzierten E-Modul E_r nach Gleichung [22] zu bestimmen. Darüber hinaus werden Simulationen durchgeführt, bei denen die Materialdaten für Mineral, Kollagen und die Zwickel gleich gesetzt und somit eine Indentation in einen homogenen Materialblock simuliert werden. Das Ergebnis einer solchen Simulation ist in **Abbildung 46** zu sehen. Die Simulation erfolgt dabei kraftgesteuert, was bedeutet, dass die angegebene Maximalkraft (100mN – 10N) mittels einer Stufenfunktion von ABAQUS gleichmäßig steigend aufgebracht wird. Bei der Arbeit mit ABAQUS/CAE fällt darüber hinaus auf, dass das Programm bei der Darstellung der angestrebten Elementgröße im Nanometerbereich Skalierprobleme hat. Dies ist auch der Grund, warum der starre Indenter nicht zu sehen ist. Ein weiterer Nachteil von ABAQUS/CAE liegt darin begründet, dass der Befehl "pin" nicht zur Verfügung steht. Bei der verwendeten Geometrierstellungsmethode müssen die sich deckenden Knoten der unterschiedlichen Bereiche miteinander gekoppelt werden. ABAQUS stellt unterschiedliche Verfahren dafür zur Verfügung, die sich vor allem dadurch unterscheiden, welche Freiheitsgrade ("degree of freedom", DOF) übergeben bzw. fixiert werden. Die Zuordnung innerhalb von ABAQUS erläutert **Abbildung 47**.

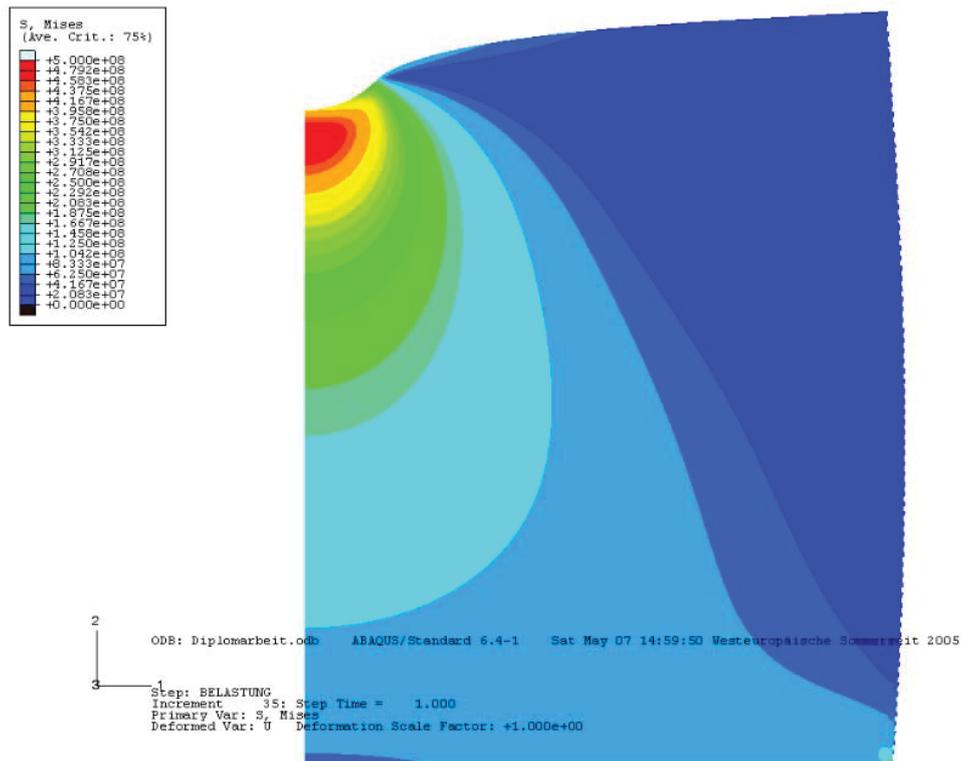


Abbildung 46: Indent in „homogenen“ Block mittels eines "rigid" Indenters. Gut zu erkennen ist, dass für tiefere Indentationen die Gesamtgeometrie vergrößert werden muss. Einerseits reicht die "Spannungsbirne", also der Bereich welcher Spannungen ausgesetzt ist, immer tiefer in das Material hinein und andererseits beginnt sich die rechte Außenseite merklich zu verformen.

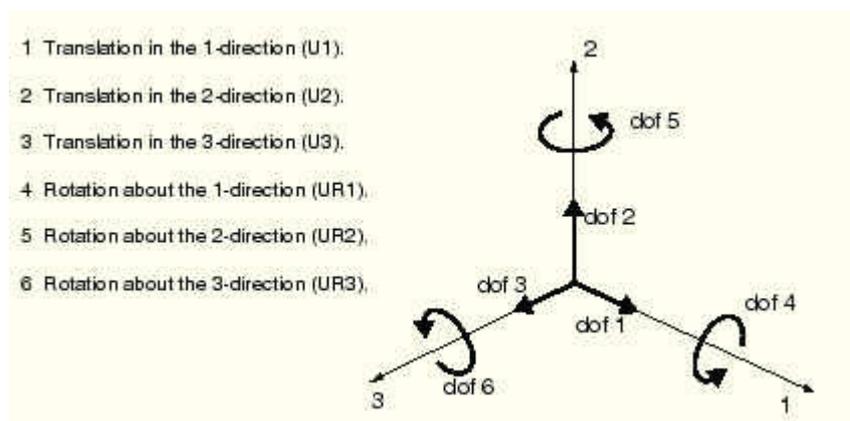


Abbildung 47: Die Nummernzuordnung der Freiheitsgrade innerhalb von ABAQUS. Darüber hinaus gibt es noch Freiheitsgrade wie etwa Temperatur und/oder elektrisches Potential [8].

Der unter ABAQUS/CAE zur Verfügung stehende und verwendete Befehl "tie" übergibt in der durchgeführten 2D-Simulation neben den beiden Verschiebungen auch die Momente. Da die verwendeten Elemente allerdings Kontinuumelemente darstellen, welche sowieso keine

Freiheitsgrade für Momente aufweisen, sollte der Befehl "pin" verwendet werden, da dieser nur die Punktverschiebungen übergibt. Dies minimiert den Rechenaufwand sowie den Speicherbedarf. Aus diesen Gründen wird dazu übergegangen, das Netz mittels Programmierung selbst zu erstellen. So kann der gewünschte Befehl verwendet werden und man erhält ein strukturiertes Netz, welches die Auffindung einzelner Elemente und Knoten erleichtert.

4.4.2.2 Netzgenerierung mittels Programmierung in FORTRAN/C++

Nach der Programmiersprache FORTRAN77 (aufgrund eines Compilerfehlers nicht verwendbare Gratissoftware aus dem Internet) wird auf die Programmiersprache Microsoft Visual C++ 6.0 gewechselt, für die eine gekaufte Vollversion zur Verfügung steht. Der große Vorteil der durch ein selbstgeschriebenes Programm automatisierten Netzgenerierung liegt in den dadurch vereinfachten Variationsmöglichkeiten in Geometrie (Plättchenzahl, Abmessungen,...) und Elementtyp.

4.4.2.2.1 Die Geometrie – Modell der "Einheitszelle"

Für eine Automatisierung der Vernetzung hat es sich als notwendig erwiesen, eine Unterteilung in "Einheitszellen", wie in **Abbildung 48** gezeigt, durchzuführen.

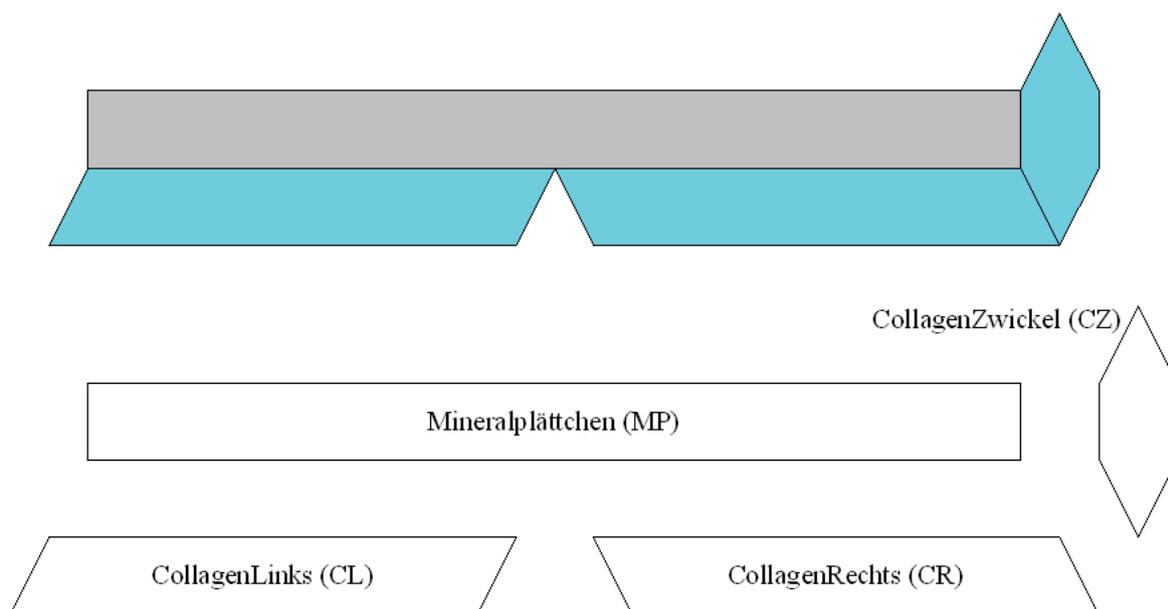


Abbildung 48: Der Aufbau einer "Einheitszelle". Diese basiert auf dem Mineralplättchen MP (grau), den beiden darunter liegenden Kollagenstreifen CL und CR (cyan) sowie dem Kollagenzwickel CZ (cyan) rechts davon.

Eine solche Einheitszelle besteht aus dem Mineralplättchen, den beiden Teilbereichen unterhalb der Mineralplättchens sowie dem Kollagenzwickel. Diese Unterteilung wird notwendig, da ganze als auch halbe Mineralplättchen innerhalb einer Halbgeometrie enthalten sind. Daraus resultierend ergeben sich zwei unterschiedliche Kollagenformen unterhalb des Mineralplättchens. Mit der getroffenen Definition, dass die Mineralplättchenanzahl immer ungerade ist, dass die letzte Mineralplättchenreihe keine Kollagenbereiche unterhalb hat und dass die Gesamtgeometrie am rechten Rand nicht eben sein muss, ergeben sich insgesamt 4 unterschiedliche Kollagenzwickelformen, wie in **Abbildung 49** abgebildet.

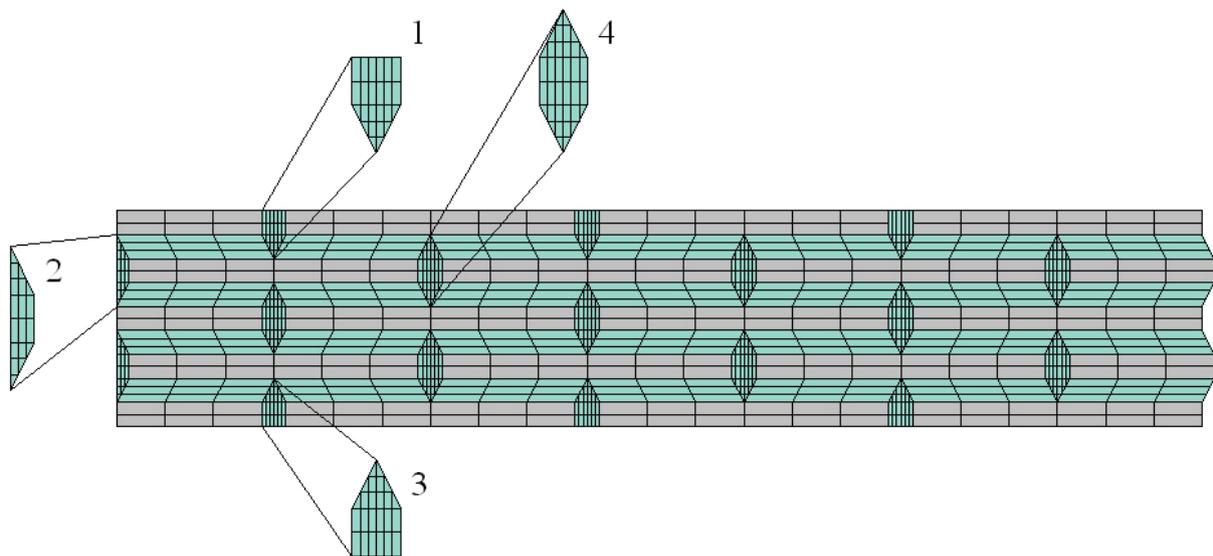


Abbildung 49: Die vier, anhand der im Text beschriebenen Definitionen, erhaltenen Zwickelformen. Die Form 1 tritt nur innerhalb der ersten Mineralplättchenreihe auf, die Zwickelform 3 ist nur in der letzten Mineralplättchenreihe zu finden, die Zwickelform 2 ist nur in der Symmetrieebene (also in geraden Mineralplättchenreihen) vorhanden und die restlichen Kollagenzwickel haben die Kollagenzwickelform 4.

Aus diesen Einheitszellen lässt sich die Gesamtgeometrie aufbauen. Gekoppelt mit der Einführung einer definierter Knoten- und Elementnummerierung lässt sich ein strukturiertes Netz erstellen, wie dies in **Abbildung 50** ersichtlich ist.

Folgendes Schema für die Nummernvergabe kommt zur Anwendung:

zwwwyyxx

Dabei gilt

z definiert, um welche Definition es sich handelt:

- 0.....Knoten Mineralplättchen
- 1.....Knoten Kollagen links
- 2.....Knoten Kollagen rechts
- 3.....Knoten Kollagen Zwickel
- 5.....Element Mineralplättchen
- 6.....Element Kollagen links
- 7.....Element Kollagen rechts
- 8.....Element Kollagen Zwickel

www definiert das Mineralplättchen (Nummer des Mineralplättchens)

yy definiert die Elementreihe (-schicht)

xx definiert die Elementspalte

	0www0101	0www0102	0www0103	0www0104	0www0105	0www0106	0www0107	
	5www0101	5www0102	5www0103	5www0104	5www0105	5www0106	5www0107	
	0www0201	0www0202	0www0203	0www0204	0www0205	0www0206	0www0207	
	5www0201	5www0202	5www0203	5www0204	5www0205	5www0206	5www0207	
	0www0301	0www0302	0www0303	0www0304	0www0305	0www0306	0www0307	
	1www0101	1www0102	1www0103	1www0104	2www0101	2www0102	2www0103	2www0104
	6www0101	6www0102	6www0103	6www0104	7www0101	7www0102	7www0103	7www0104
	1www0201	1www0202	1www0203	1www0204	2www0201	2www0202	2www0203	2www0204
	6www0201	6www0202	6www0203	6www0204	7www0201	7www0202	7www0203	7www0204
	1www0301	1www0302	1www0303	1www0304	2www0301	2www0302	2www0303	2www0304
	6www0301	6www0302	6www0303	6www0304	7www0301	7www0302	7www0303	7www0304
	1www0401	1www0402	1www0403	1www0404	2www0401	2www0402	2www0403	2www0404

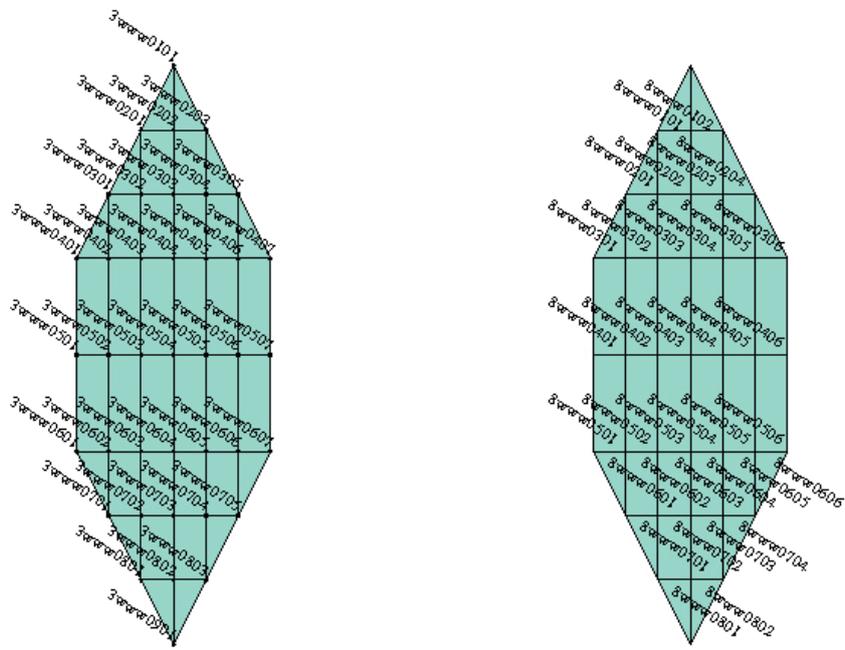


Abbildung 50: Knoten- und Elementnummerierung zum Aufbau eines strukturierten Netzes. Die Ziffern sind die Knoten- bzw. Elementnummern. Es fällt der logische Zusammenhang auf, dass es bei einer Unterteilung immer eine Knotennummernreihe mehr als Elementreihen gibt.

Mit dieser Nummerierung lässt sich ein Netz mit 999 Mineralplättchen realisieren, bei denen jeder Mineralplättchen als auch jeder Kollagenbereich 99 Elementschichten und 99 Elementspalten haben dürfte. Bei der Programmierung muss, wie in **Abbildung 51** ersichtlich ist, auf die korrekte Kopplung der deckungsgleichen Knoten geachtet werden. Eine Erläuterung zu den einzelnen Variablen (z.B. $aempl = \text{Anzahl Elemente MineralPlättchenLänge, \dots}$) kann dem Quellcode im **Anhang a)** (Diplomarbeit.cpp) entnommen werden. Die Gruppenbezeichnungen ergeben sich aus der Teilbereichsbezeichnung, der Position des Teilbereichs sowie der Position innerhalb des Teilbereichs (z.B. MPLO = MineralPlättchen Links Oben).

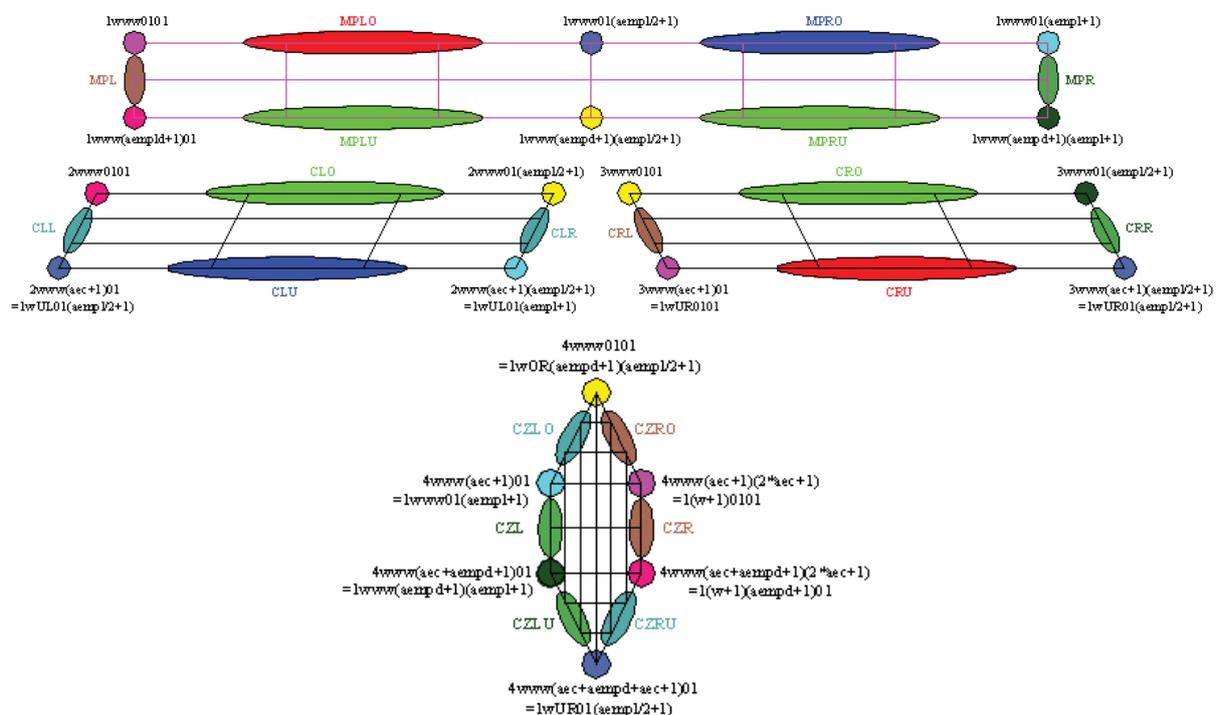


Abbildung 51: Knotennummernvergabe und -gruppierung der vier Teilbereiche der Einheitszelle. Farblich markiert sind die zu verknüpfenden Knotengruppen (nsets) bzw. Einzelknoten.

Ein Hauptproblem bei der Programmierung mittels des "downgeloadeten FORTRAN F77 Compilers" ist die Tatsache, dass dieser nur Integer-Zahlen (ganzzahlig) bis $\pm 2097152 (= 2^{21})$ verarbeiten kann. Damit lässt sich kein strukturiertes Netz in der zu verwendenden Größenordnung herstellen. Diese Unzulänglichkeit wird lange nicht erkannt, da laut zugehörigem Benutzerhandbuch Integer-Zahlen bis $2^{31} = \pm 2147483648$ verarbeitet werden können und der Compiler auch keine Möglichkeit aufweist, das Programm Schritt für Schritt abzuarbeiten, was die Fehlersuche immens erschwert.

Mit der Verwendung von C++ kann ein strukturiertes Netz erstellt werden, welches aufgrund der getroffenen Unterteilung neben Viereck- auch Dreieckelemente beinhaltet. Durch die getroffene Automatisierung der Vernetzung geht allerdings die Möglichkeit verloren das Netz dort, wo es erforderlich ist (wo große Belastungen bzw. Verschiebungen auftreten) ohne großen Aufwand feiner zu vernetzen.

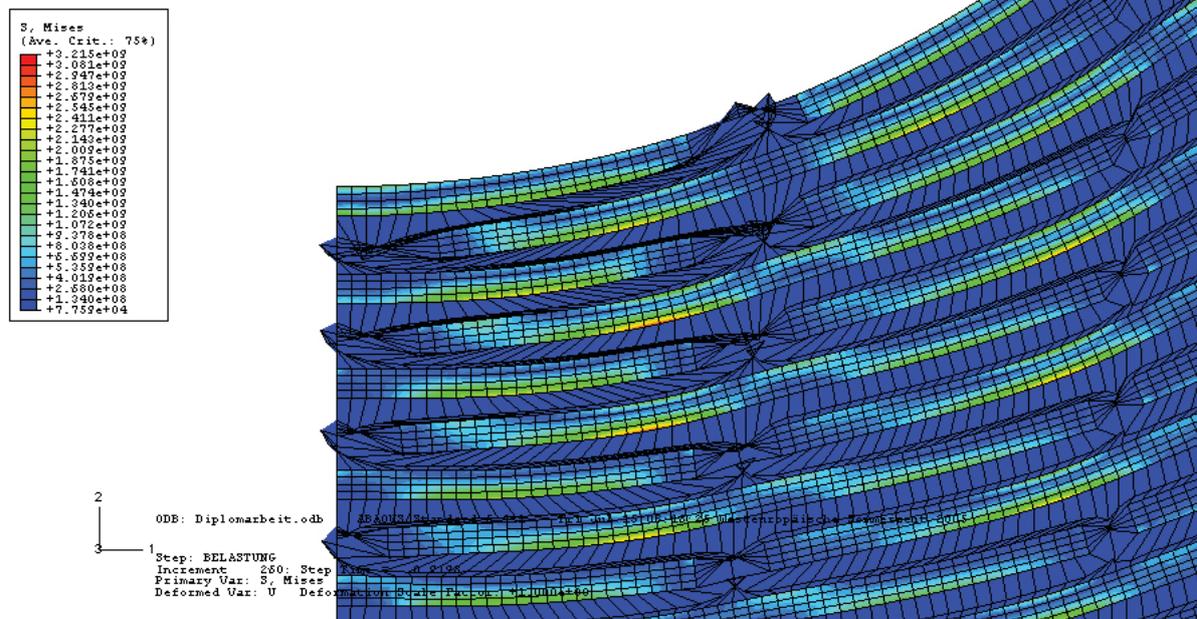


Abbildung 52: Die bei zunehmender Eindringtiefe immer stärker werdenden Elementverzerrungen führen zu Simulationsabbrüchen bzw. nicht vollkommen glaubwürdigen Ergebnissen (hier Eindringtiefe etwa 30nm).

Wie in **Abbildung 52** zu sehen, kommt es bei zunehmender Eindringtiefe zu starker Elementverzerrungen vor allem der Kollagenelemente unterhalb der Indenterspitze. Zahlreiche Simulation mit unterschiedlicher Elementgröße, unterschiedlichen Elementtypen und auch Geometrieabmessungen (a und b größer angenommen z.B. 5nm), zur Minimierung des Mineralisierungsgrades, wie dies auch in **Abbildung 52** zu sehen ist, konnte diese Verzerrungen nicht wirksam minimieren. So wird auch versucht, mit achsensymmetrischen Elementen eine Verzerrungsminimierung herbeizuführen, da ABAQUS dadurch aus der 2D-Geometrie kreisrunde "3D-Mineralringe" interpretiert und diese starrer sind als rechteckige Platten. Auch eine Vergrößerung der Gesamtgeometrie auf eine Dicke von 1500nm verhindert diese Verzerrungen nicht, erhöht allerdings den Rechenaufwand und den benötigten Speicherbedarf sowohl auf der Festplatte als auch im Arbeitsspeicher. Der im Lokalrechner installierte RAM von 1024MB reicht für die Abarbeitung eines solchen Problems nicht mehr aus. Daher wird mit dieser Simulation auf den Applikationsserver "zalpha1" gewechselt.

Eine Möglichkeit, diese starken Verzerrungen zu verhindern, ist die Verwendung eines Programmwerkzeuges, welches die Ergebnisse zwischenspeichert, eine Neuvernetzung ("remeshing") durchführt und die zwischengespeicherten Lösungen dem neuen Netz zuordnet. Eine weitere Möglichkeit ist die Benutzung einer adaptiven Vernetzung ("adaptiv mesh"). Dabei werden einzelne Knotenpunkte unter Beibehaltung der Gleichgewichtsbedingungen (Verschiebungsenergien usw.) innerhalb der Geometrie verschoben und die Verzerrung somit minimiert. Die erste angeführte Möglichkeit bietet ABAQUS generell nicht an. Die zweite Möglichkeit bietet ABAQUS/Standard auch nicht an, sie steht allerdings unter ABAQUS/Explicit zur Verfügung.

4.4.3 Reduzierte (2D) Simulation – ABAQUS/Explicit

Der Quellcode den ABAQUS/Standard verwendet wird auf ABAQUS/Explicit adaptiert. Neben der Vertiefung von ABAQUS/Standard und der Einarbeitung in FORTRAN F77 sowie C++ muss auch die dafür zu verwendende Befehlsyntax erlernt werden. Sowohl die Übereinstimmung mit der Realität, als auch die Konvergenz erhöht sich mit steigender Elementzahl. Für ABAQUS/Explicit kann man nachvollziehen, dass dies gleichbedeutend mit einem sinkendem Zeitschritt ist. Dieser ist (unter Vernachlässigung der Poisson-Zahl), wie in **Kapitel 3.6.2** ausgeführt, begrenzt durch die Beziehung [45]:

$$\Delta t_{stable} = \frac{2}{\omega_{max}} = \frac{L^e}{c_d}$$

wobei ω_{max} die höchste Frequenz des Systems, L^e die charakteristische Elementlänge (z.B. die kürzere Seite eines rechteckigen Elements) und c_d die Wellenausbreitungsgeschwindigkeit

$$c_d = \sqrt{\frac{E}{\rho}} \quad [55]$$

innerhalb der Materie darstellt. Man erkennt, dass c_d von den Materialkenndaten abhängt. Aufgrund der stark unterschiedlichen E-Moduli der beiden verwendeten Materialien schwanken die zugehörigen Geschwindigkeiten ebenfalls stark.

Mit einer Verdopplung der Elemente steigt die Rechenzeit allerdings um den Faktor vier (4-fache Elementanzahl bedeutet also theoretisch 16-fache Rechenzeit, die in der Praxis, aufgrund Rechenungenauigkeiten und Datenaustauschzeiten, noch etwas steigen kann.). Bei ABAQUS/Explicit wird nun wie bei expliziter Rechnung notwendig nicht mehr masselos gerechnet. Dies führt, wie man in **Abbildung 53** ansatzweise erkennen kann, zu Schwingungsproblemen (wellenförmige Ausbreitungen des induzierten Indents) und muss

durch geschickte, durch weitere Simulationen auszutestende, Materialkennwerte verhindert bzw. unterdrückt werden.

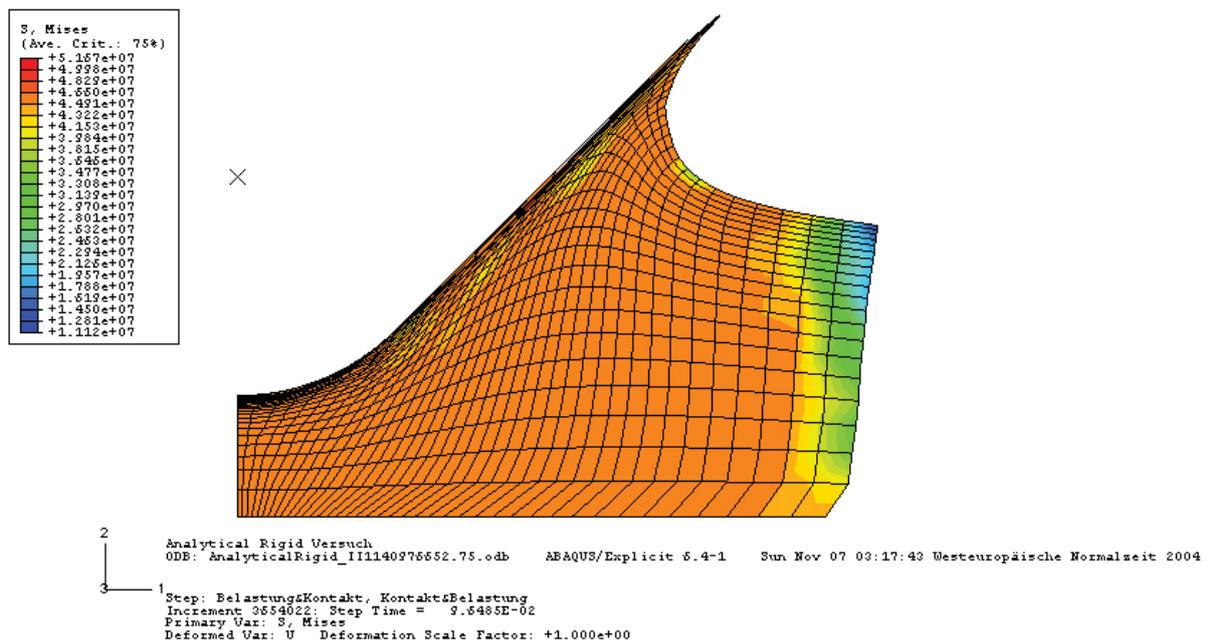


Abbildung 53: Ergebnis eines Eindringversuchs mittels ABAQUS/Explicit in ein homogenes Material. Bei dieser Simulation kam es durch ungeeignete Wahl der Dämpfungskonstanten zu einem wellenartigen "Wegfließen" des Materials.

4.4.3.1 Netzgenerierung mittels Programmierung in C++

Die Programmierung des Netzes kann änderungslos übernommen werden. Die Knoten- und Elementdefinition ist für beide ABAQUS-Varianten identisch. Leider muss festgestellt werden, dass der von ABAQUS/Explicit zur Verfügung gestellte "Adaptive mesh" - Befehl zu keiner merklichen Verhinderung der starken Elementverzerrungen führt. Der Rechenaufwand und die damit verbundene Rechenzeit sowie der benötigte Arbeitsspeicher steigen allerdings enorm. Dieses Problem lässt sich nur mehr auf einem Cluster (Zusammenschluss mehrerer Computer wobei jeder die Aufgabe unterteilt und den einzelnen Rechnern zugeteilt wird) oder einem "shared memory"-System (mehrere CPU's teilen sich einen Rechenjob wobei sie sich den Arbeitsspeicher teilen, was einen schnelleren Datentransfer ermöglicht) abarbeiten. Der Applikationsserver der Montanuniversität "zalpha1" gehört zu den Zweitgenannten, wobei dieser bereits zu den älteren Geräten zu zählen ist und lediglich 2 CPU's zur Verwendung mit ABAQUS zur Verfügung stehen. Die so durchgeführten Simulationen brechen, wie die ABAQUS/Standard Simulationen mit hohen Elementzahlen ab, wenn der vom Betriebssystem zur Verfügung gestellte, limitierte Festplattenspeicher von etwa 4GB von einer Ausgabedatei

(meist *.odb) erreicht wird. Auch wenn die Ausgabehäufigkeit bzw. die Anzahl der Variablen, für die eine Ausgabe geschrieben wird, reduziert werden, können diese Simulationen nur durch Umschreiben der *.inp-Datei und Neustart (Restart) dieser geänderten Datei zum Weiterrechnen gebracht werden. Solche "wiedererweckte" Simulationen stürzen immer wieder, ohne Angabe von Gründen in der Nachrichten- (*.msg) bzw. Protokolldatei (*.dat) ab.

In einem Gespräch mit Dr. Johannes Meenen (ABAQUS/Deutschland) auf dem User-Meeting der FSI-Kopplungssoftware MpCCI (Fluid-Structure-Interaction) des Fraunhoferinstituts zur Kopplung eines FEM- (Finite-Element-Methode) und eines CFD- (Computerized-Fluid-Dynamic) Programms wurden die Probleme angesprochen und seinerseits folgende Vorschläge gemacht:

- Vermeidung von parallelgramm- und dreiecksförmigen Elementen
- Vermeidung von Elementen mit sehr unterschiedlichen Seitenverhältnissen
- Verwendung von CPE4I-Elementen für die Mineralplättchenbereiche
- Verwendung vom CPE4R- bzw. CPE4RH-Elemente für die Kollagenbereiche
- Vorerst Weglassen des Kontakts und Drücken des Kontaktpunktes zum Indenter
- Weggesteuerte Indentation anstelle der kraftgesteuerten (bessere Konvergenz)
- Einbau des Kontaktes und weggesteuertes Verfahren des Indenters
- Verwendung des Befehls "merge" innerhalb ABAQUS/CAE
- Weglassen der Adaptierung und Rechnen mit ABAQUS/Standard

Schon der erste Punkt ist durch eine einfache Umprogrammierung des C++-Quellcodes nicht zu verwirklichen, da eine solche zu umfangreich und aufwendig ist. Dies würde eine komplette Neuprogrammierung nötig machen.

4.4.3.2 Netzgenerierung mittels CAE

Eine Durchführung weiterer ABAQUS/Explicit Simulationen erscheint aufgrund der gemachten Vorschläge nicht zielführend.

4.4.4 Reduzierte (2D) Simulation – ABAQUS Standard

Auf die Informationen durch das oben angeführte Gespräch gründend wird die Kenntnis in ABAQUS/CAE weiter vertieft und weitere Simulationen mit den vorgeschlagenen Änderungen durchgeführt.

4.4.4.1 Netzgenerierung mittels Programmierung in C++

Wie bereits in **Kapitel 4.4.3.1** beschrieben ist eine Umprogrammierung nicht zweckmäßig und widerspricht auch den gemachten Vorschlägen.

4.4.4.2 Netzgenerierung mittels CAE

Mit Hilfe des Befehls "merge" stellt die Erstellung auch großer Geometrien kein Problem mehr dar. Es müssen die Mineralplättchen nun nicht mehr einzeln gezeichnet und dann verknüpft werden, sondern es wird eine Maske erstellt mit deren Hilfe alle Mineralplättchen gleichzeitig aus einem Block "herausgeschnitten" werden. Der übrig bleibende Block entspricht den Kollagenbereichen. Man hat nun zwei Domänen, die man vereinen (merge) kann. Jeder Domäne werden die entsprechenden Materialkenndaten zugeordnet und die Simulation gestartet. Trotz zahlreicher Versuche konnte auch mit diesen Änderungen kein zufriedenstellendes Ergebnis produziert werden.

5 Die Simulationsergebnisse

5.1 Indentation senkrecht auf die Mineralplättchen

Mit den zur Verfügung stehenden Mitteln (Hard- und Software) konnte in angemessener Zeit trotz zahlreicher Versuche mit unterschiedlichsten Verfahrensrouten (ABAQUS/Standard, ABAQUS/Explicit, mit/ohne adaptivem Netz, Elementtypen, ABAQUS/CAE Netz, FORTRAN/C++ Netz,.....) kein validierbares Ergebnis erhalten werden. In der Realität liegen die Indentationstiefen bei etwa 100nm - 600nm [15]. Um den Verfahrensweg wiederzugeben, wird die Auswertung anhand einer Simulation veranschaulicht, deren Eindringtiefe 22nm beträgt. Zur Auswertung kommt eine Simulation mit dieser geringen Eindringtiefe, da nur für eine solche ein gesamter Be- und Entlastungszyklus durchlaufen werden kann. Wie zu sehen ist (vgl. **Kapitel 4.4**), führt eine höhere Eindringtiefe für die gewählten Prozessparameter und Materialkennwerte zu unzulässigen Elementverzerrungen bzw. Programmabbrüchen aufgrund von auftretenden Divergenzen.

5.1.1 Geometrie, Netz und Randbedingungen

Die Simulation wird mittels ABAQUS/Standard weggesteuert durchgeführt (bei kraftgesteuerten Eindringversuchen kam es aufgrund deren numerischen Stabilität bereits bei geringeren Eindringtiefen zu Divergenzen), wobei die in **Abbildung 54** dargestellt Geometrie verwendet wird. Bei dieser Simulationsvariante wird ein Weg (und eine zugehörige Zeit, also eigentlich eine Geschwindigkeit) vorgegeben, welcher unabhängig von der benötigten Kraft gleichförmig erreicht wird.

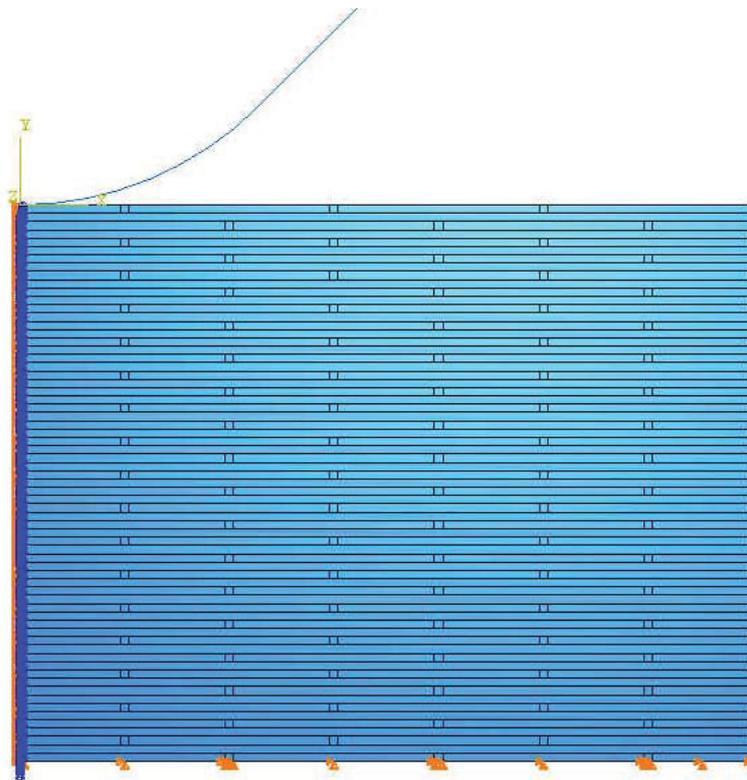


Abbildung 54: Aufbau der analysierten Geometrie mit Darstellung, farblich markiert sind wie im Text beschrieben die unterschiedlichen Randbedingungen.

Die in ABAQUS/CAE mittels des "merge"-Befehls erstellte (also mittels Vorgehen laut **Kapitel 4.4.4.2**) und vernetzte Geometrie besteht dabei aus 136 Mineralplättchen (68 davon sind halbe Mineralplättchen) mit $l = 60\text{nm}$ und $d = 2.5\text{nm}$. Die Kollagenabstände sind dabei mit $a = b = 2.5\text{nm}$ angenommen. Die Gesamtgeometrie hat somit eine Höhe $H = 167.5\text{nm}$ und eine Breite von $B = 218.75\text{nm}$. Da es sich um eine 2D-Simulation handelt wird von ABAQUS standardmäßig eine theoretische Tiefe von 1m angenommen. Dies hat zwar keine Auswirkung auf die visualisierten Ergebnisse, muss allerdings bei der Bewertung der Eindringkräfte berücksichtigt werden. Ebenfalls zu sehen sind die Randbedingungen. Die orangen Dreiecke am unteren Geometrierand stehen für eine Fixierung der Punkte in allen Raumrichtungen

("encasted"), wobei die blau-orangen Markierungen an der linken Geometrieseite sowie am Indenterradiusmittelpunkt (Referenzpunkt RP) eine Symmetriebedingung wiedergeben (Verschiebung nur in y-Richtung möglich). Verwendet wird ein "rigid" Indenter welcher weggesteuert in die durch die Randbedingungen fixierte Geometrie eindringt.

Das verwendete Netz unterteilt jedes Mineralplättchen in 4 Elemente/Mineralplättchenhöhe und 96 Elemente/Mineralplättchenlänge. Die Kollagenbereiche werden dementsprechend (deckungsgleiche Knotenpunkte) unterteilt, wodurch sich, wie in **Abbildung 55** ersichtlich, ein quadratisches Netz mit einem Gitterabstand von etwa 0.625nm ergibt. Jedes Mineralplättchen besteht somit aus 384 Elementen und somit die Gesamtgeometrie aus 93800 Elementen.

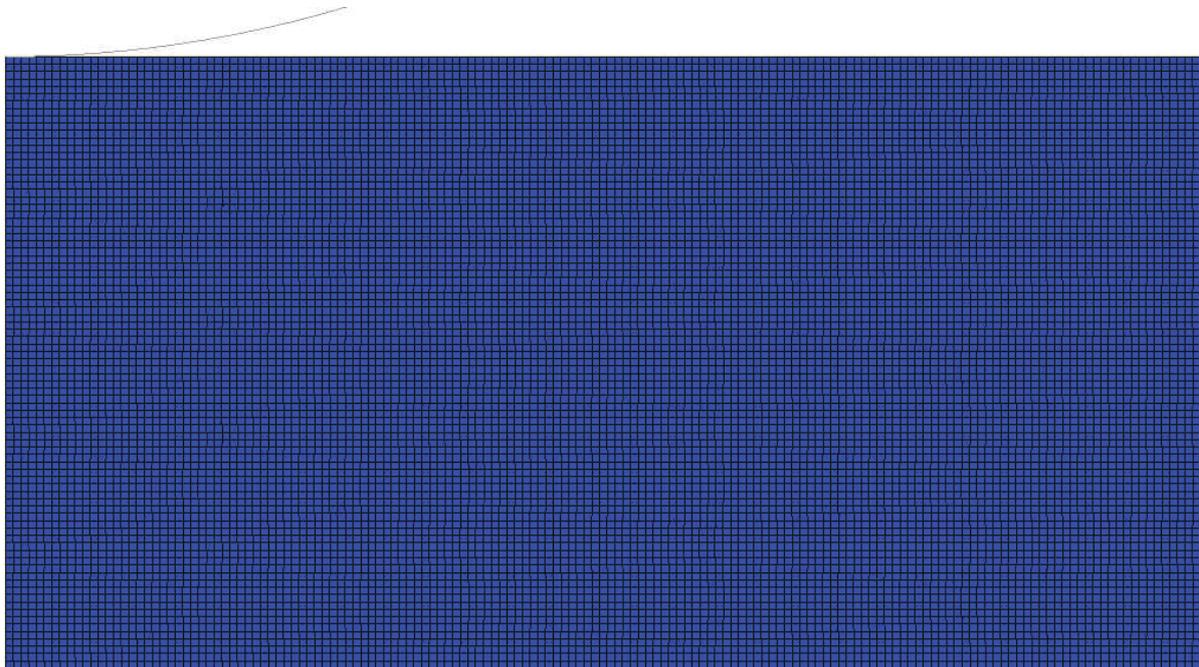


Abbildung 55: Teilausschnitt des verwendeten Netzes im Bereich der Indenterspitze.

Für die Mineralplättchen wurde der Elementtyp CPE4I und für das Kollagen der Elementtyp CPE4RH verwendet. Es handelt sich also um 4-knotige, bilineare Kontinuumselemente (C) für ebenen Dehnungszustand (PE), wobei die Mineralplättchen eine "incompatible mode"-Formulierung besitzen, wo hingegen die für das Kollagen verwendeten Elemente eine "reduced integration with hourglass control"-Formulierung verwenden, da sich diese Formulierungskombination laut ABAQUS am besten für die vorliegende Aufgabenstellung (gebettete, starre Körper in verformbarer Matrix) eignet. Details zu diesen Definitionen kann den Erläuterungen innerhalb des Quellcodes im **Anhang a)** sowie **Anhang c)** entnommen werden.

5.1.2 Visualisierung der Simulationsergebnisse

Wie aus **Abbildung 56** ersichtlich, sind die Spannungen in den Mineralplättchen wesentlich höher als im Kollagen, was aufgrund der unterschiedlichen E-Moduli allerdings zu erwarten ist.

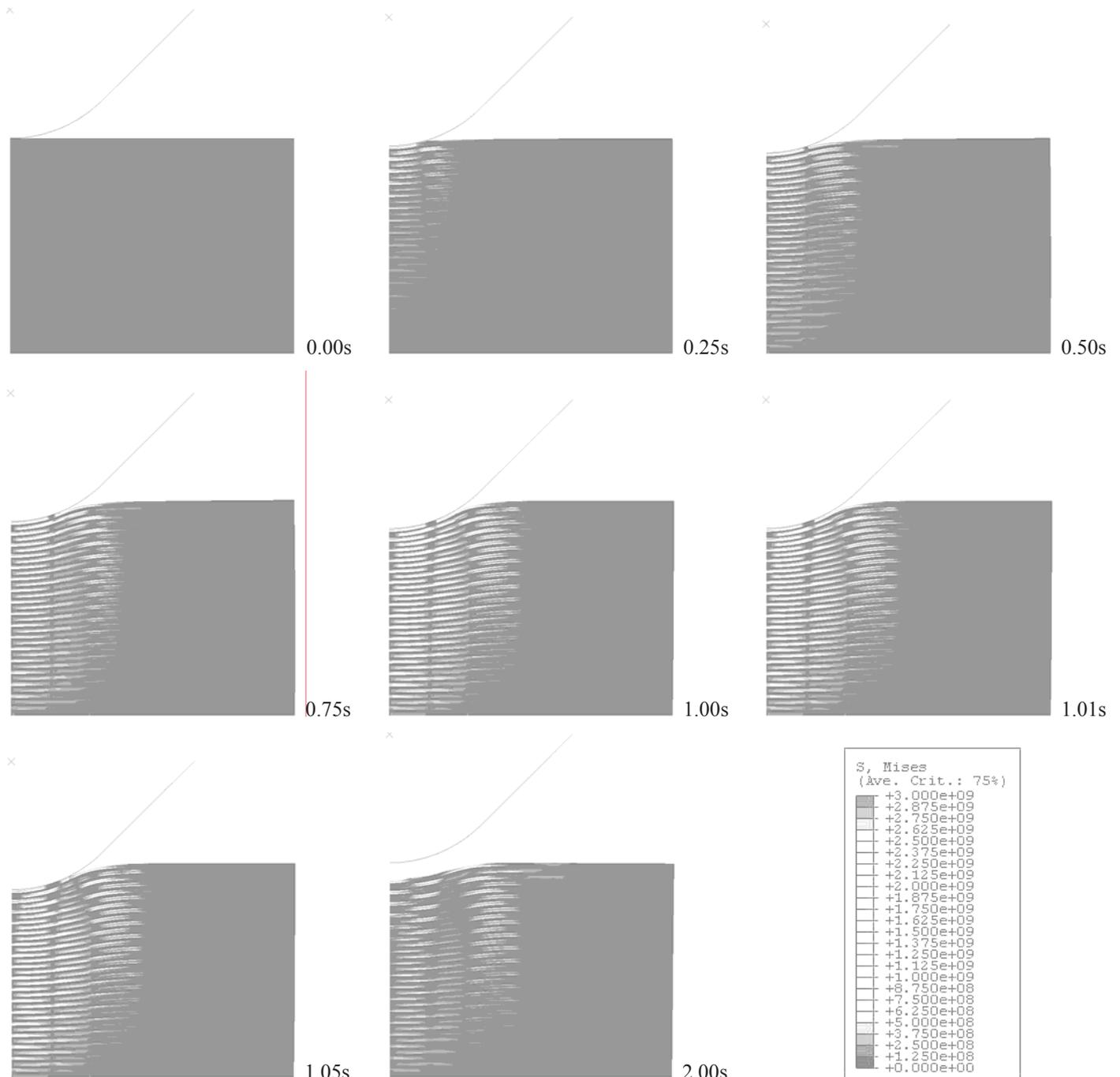


Abbildung 56: Übersicht über den Be- und Entlastungszyklus. Die ersten 4 Bilder zeigen das Eindringen des Indenters bis zu einer maximalen Tiefe von 22nm (Bild5, 1.00s) während die letzten 3 Bilder die Entlastung zeigen. Gut zu erkennen ist die bleibende (plastische) Verformung und das dadurch auftretende Lösen der Indenterspitze vom Bulk-Material. Die Skalierung ist dabei von 0MPa bis 3e9Pa linear gewählt.

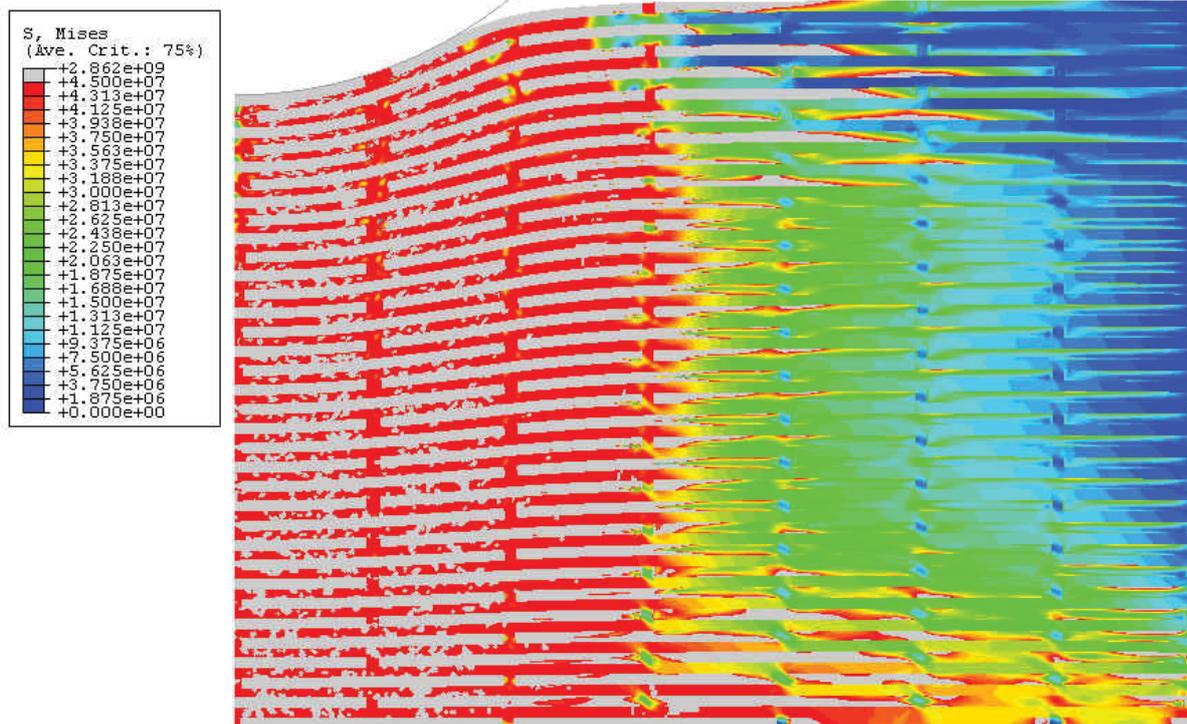


Abbildung 57: Übersicht über die Spannungsverteilung im Bulk-Material. Man erkennt, dass Spannungen bis tief in das Verbundmaterial (Bulk) vorhanden sind. Auch der Einfluss der unteren Randbedingung ist zu erkennen. Die Skalierung wurde so gewählt, dass die rot-grau Grenze bei der Plastifizierungsspannung des Kollagens liegt.

Die "Spannungs-Birne" in **Abbildung 57** bestätigt die Theorie von Hertz, nach welcher bei einem Eindringversuch ein Volumen von 9-fache Eindringtiefe und 7-fache Eindringweite zu berücksichtigen ist. So erkennt man, dass ein großer Teil des Kollagens (rote Bereiche) Spannungen in der Höhe der Plastifizierungsgrenze aufweist. Wie auch anhand **Abbildung 59** ersichtlich ist, steigt die Spannung im Kollagen sehr schnell (Eindringtiefen kleiner einer halben Mineralplättchendicke, also ca. 2nm) bis zu dieser Grenze an. Somit ist die Antwort auf die erste forschungsleitende Fragestellung gefunden. Der Übergang von rein elastischer zu plastischer Verformung tritt aufgrund des niedrigen E-Moduls des Kollagens also beinahe sofort, also schon bei extrem kleinen Eindringtiefen, ein.

Während in **Abbildung 56** versucht wird, eine Übersicht über den gesamten Eindringversuchszyklus und damit die gesamten Werteskala darzustellen, ist aus **Abbildung 58** die Spannungsverteilung in den Mineralplättchen und in **Abbildung 59** die Spannungsverteilung im Kollagen bis zur maximalen Eindringtiefe (22nm) mit jeweils darauf angepasster Skalierung ersichtlich, was durch die genauere Auflösung die Bewertbarkeit der auftretenden Spannungen verbessert.

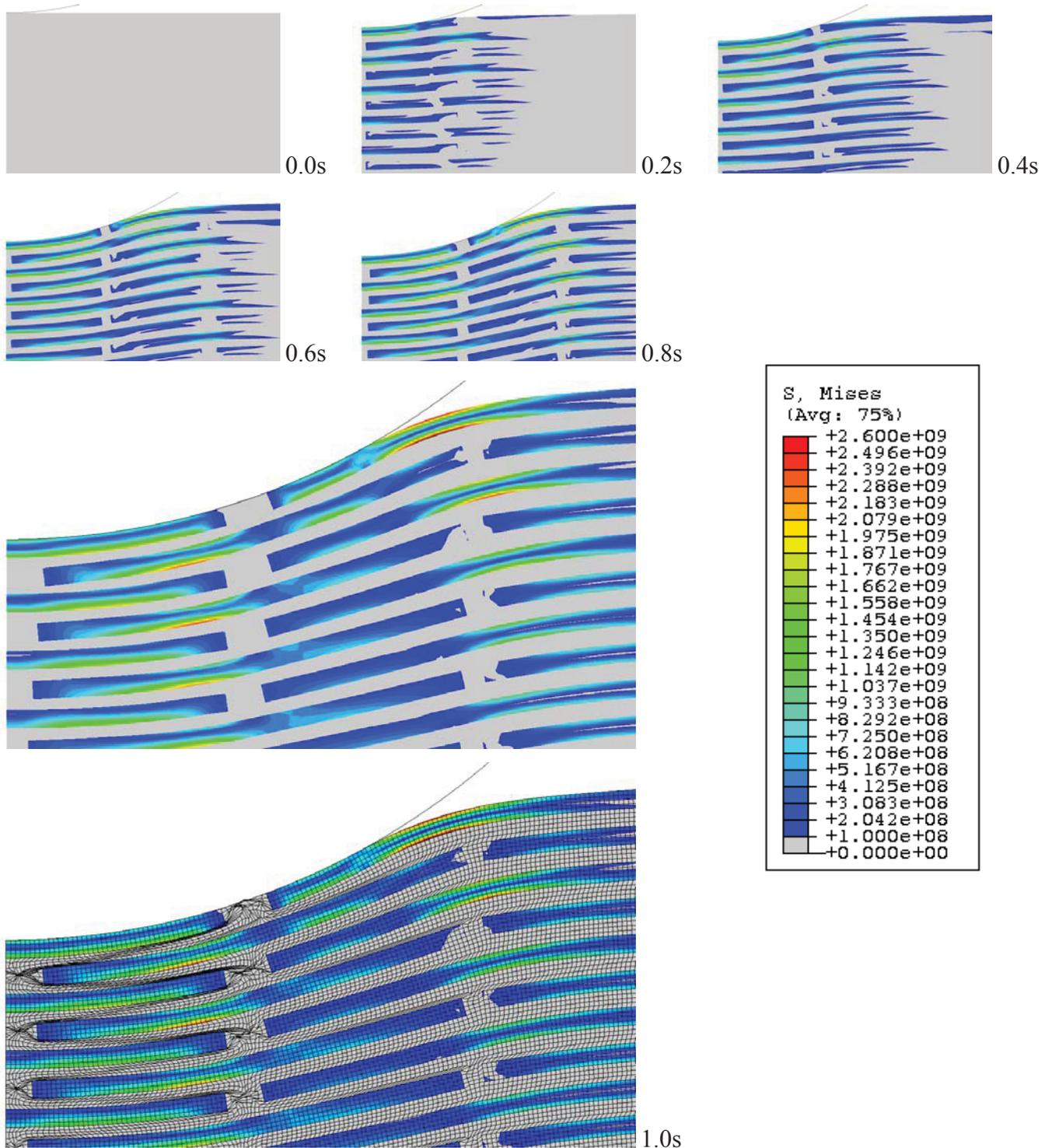


Abbildung 58: Die Spannungsverteilung innerhalb der Mineralplättchen bis zur maximalen Eindringtiefe von 22nm. Die Skalierung wurde dementsprechend angepasst und geht nun linear skaliert von $1e8\text{Pa}$ (blau) bis $2.6e9\text{Pa}$ (rot). Die nicht mehr unerheblichen Elementverzerrungen, vor allem der Kollagenbereiche in der Nähe der Mineralplättchenkanten, sind gut erkennbar.

Wie aus **Abbildung 58** ersichtlich ist, liegt die maximale Spannung innerhalb der Mineralplättchen bereits für diese geringe Eindringtiefe bei etwa $2.5e9\text{Pa}$ (2.5GPa) und damit bereits in der Größen-

ordnung der theoretischen Festigkeit von einem Dreißigstel des E-Moduls:

$$\sigma_{\text{theoretisch}} = \frac{E_{\text{Mineral}}}{30} = \frac{114\text{GPa}}{30} = 3.8\text{GPa}$$

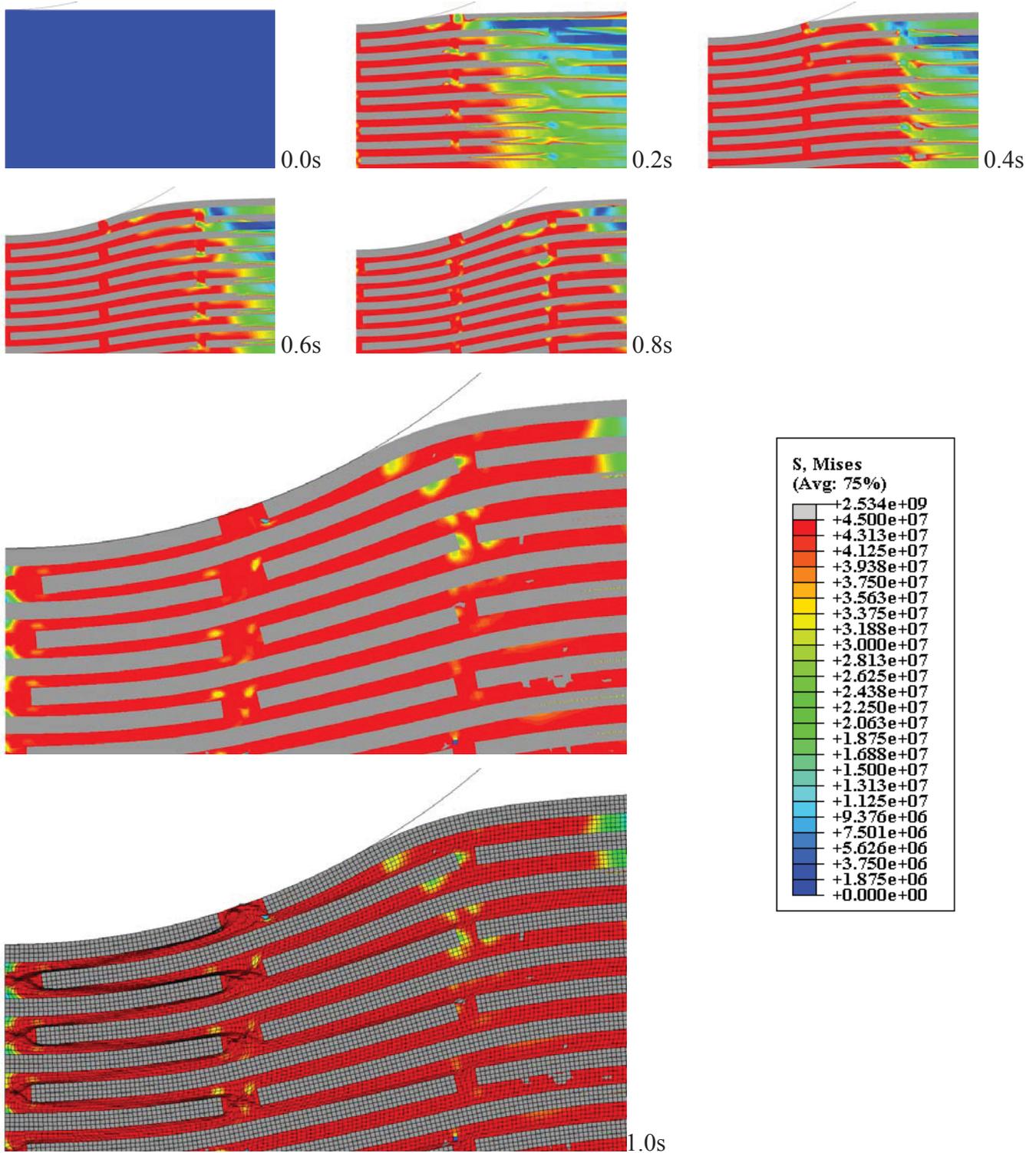


Abbildung 59: Die Spannungsverteilung innerhalb des Kollagens bis zur maximalen Eindringtiefe von 22nm. Die Skalierung wurde dementsprechend angepasst und geht nun linear skaliert von 0 (blau) bis $4.5e7$ (rot).

Während die maximale Festigkeit von makroskopischen Hydroxyapatit noch bei etwa 100MPa liegt, gibt es keine eindeutigen Aussagen über die Festigkeit von nanometergroßen Hydroxyapatitplättchen. Die Annahme, dass diese beinahe ihre theoretische Festigkeit erreichen können, liegt nahe. Die Festigkeitsabnahme diverser Materialien geht zumeist auf Verunreinigungen oder andere Inhomogenitäten zurück. Je kleiner die Materialteilchen sind, desto geringer ist die Wahrscheinlichkeit, dass sich festigkeitsminimierende (z.B. rissinduzierende) Inhomogenitäten darin befinden. Dies gilt insbesondere, wenn die Materialteilchen kleiner werden als die Größenordnung der meisten Inhomogenitäten. So braucht nicht nur ein Verunreinigungsteilchen ein bestimmtes Volumen, sondern auch eine Versetzung um die "Versetzungsschleife" zu schließen.

Wie zu erwarten war und man dies **Abbildung 59** entnehmen kann, ist die maximale Spannung im Kollagen, allerdings großflächig, mit $4.5e7\text{Pa}$ (45MPa) begrenzt, was das ideal-elastisch-plastischen Materialverhalten (vgl. **Kapitel 4.2**) des Kollagens widerspiegelt, welches ab diesem Spannungswert ein rein plastisches Materialverhalten (keine weitere Spannungserhöhung) zeigt (vgl. **Abbildung 33**).

5.1.3 Ermittlung der validierbaren projizierten Indentationsfläche

Auch wenn sich nur der unter **Kapitel 5.1.1** beschriebene Versuch auswerten lässt und sich dessen Ergebnisse aufgrund der niedrigen Eindringtiefe zu keinem aussagekräftigen Vergleich mit Literaturwerten heranziehen lässt, wird dessen Auswertung durchgeführt, um den Ablauf zu erläutern. Für die folgenden Auswertungen muss berücksichtigt werden, dass die ausgewerteten Simulationsergebnisse aus einer zweidimensionalen Simulation stammen, ABAQUS zur Kraft- bzw. Spannungsermittlung allerdings eine Fläche A und somit eine "Referenztiefe" benötigt, die in diesem Fall mit $d = 1\text{m}$ angenommen wurde. Andererseits ist es in ABAQUS/CAE nicht möglich Geometrien in [nm] zu erstellen. Somit ergibt sich für die Umrechnung, das $d = 10^9\text{m}$ [nm auf m] als Referenztiefe verwendet werden muss. Für die Spannungs- und Verformungsermittlung in einer 2D-Simulation stellt diese Skalierung kein Problem dar, was schon daran gesehen werden kann, dass die Spannungen in den Materialien bei Simulationen mit in C++ erstellten Geometrien, bei welchen die Einheit [nm] verwendet werden konnte, die selbe Höhe aufweisen. Will man, wie im folgenden Kapitel, die benötigte Indenterkraft berechnen, muss man berücksichtigen, dass die in der Simulation ermittelte Kraft jener entspricht, welche benötigt wird, um eine 10^9m lange "Indenterkante" mit einem Indenterradius von 100nm in das Bulk-Material zu drücken.

Erschwert wird diese Umrechnung durch den Umstand, dass es zu einer Oberflächenverformung (Sink-in Effekt) kommt. Ohne den Sink-In-Effekt würde sich für eine Eindringtiefe von $h = 22\text{nm}$ bei einem Indenterradius von $r = 100\text{nm}$ mit Hilfe der Kreisabschnittsformel folgende projizierende Länge ergeben:

$$l' = 2 \cdot \sqrt{b \cdot (2 \cdot r - b)} = 2 \cdot \sqrt{22 \cdot (2 \cdot 100 - 22)} = 125.2\text{nm}$$

bzw.

$$a' = \frac{l'}{2} = 62.6\text{nm}$$

Da es allerdings, wie man aus **Abbildung 60** erkennt, zu einem solchen Effekt kommt, verringert sich dieser Abstand und es kann ein Abstand von $a = l/2 = 50.7\text{nm}$ abgemessen werden, womit sich eine folgende projizierende Länge ergibt:

$$l = 101.4\text{nm}$$

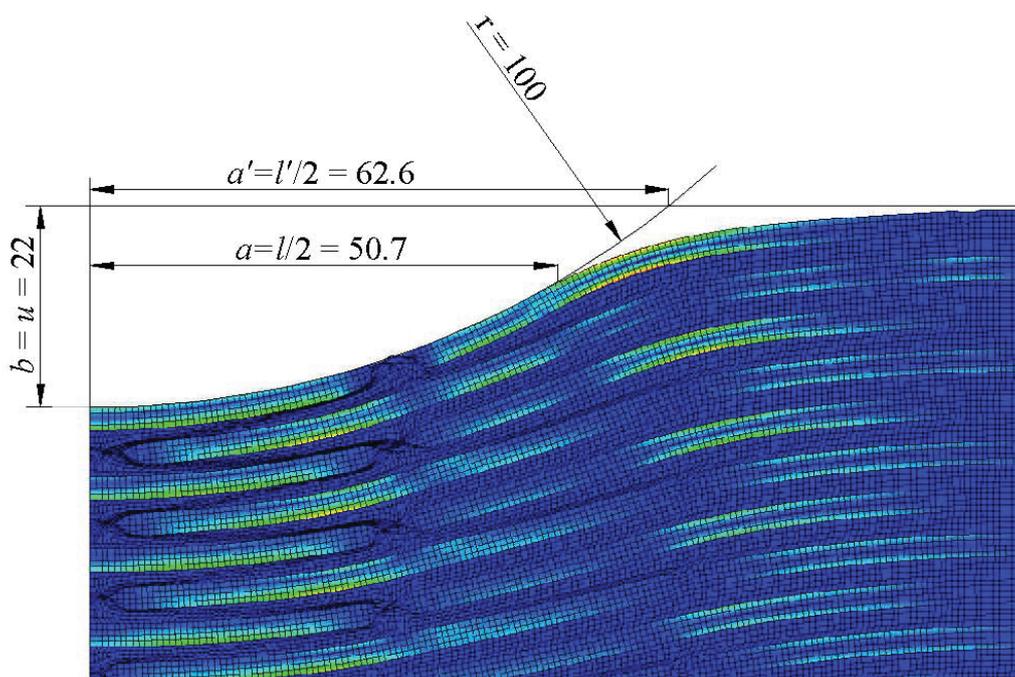


Abbildung 60: Aus dem Postprocessingbild lässt sich für die maximale Eindringtiefe von $u = 22\text{nm}$ und einem Indenterspitzenradius von $r = 100\text{nm}$ ein für die projizierende Fläche maßgeblicher Maß $a = 50.7\text{nm}$ ermitteln. Laut Sehnenformel würde sich, also unter Vernachlässigung des Sink-in Effekts, ein Maß von $a' = 62.6\text{nm}$ ergeben.

Ausgehend davon, dass zu Beginn der Indentation die projizierende Fläche für beide Fälle, 0 ist und sich mit der Sehnenformel für jede Eindringtiefe ein Maß a' berechnen lässt, kann

man darüber nachdenken, dass für die jeweilige Indentationsfläche maßgebliche Maß a mittels einer vereinfachten Annahme zu berechnen. Diese Annahme geht davon aus, dass das Verhältnis zwischen a und a' während des gesamten Indentationsversuches konstant bleibt. Man erhält also einen Skalierungsfaktor f :

$$f = \frac{a}{a'} = \frac{50.7}{62.6} = 0.8099$$

Mit Hilfe dieses Skalierungsfaktors kann die für die projizierte Fläche maßgebliche Maß a , wie aus **Abbildung 61** ersichtlich, direkt aus der Eindringtiefe bestimmt werden.

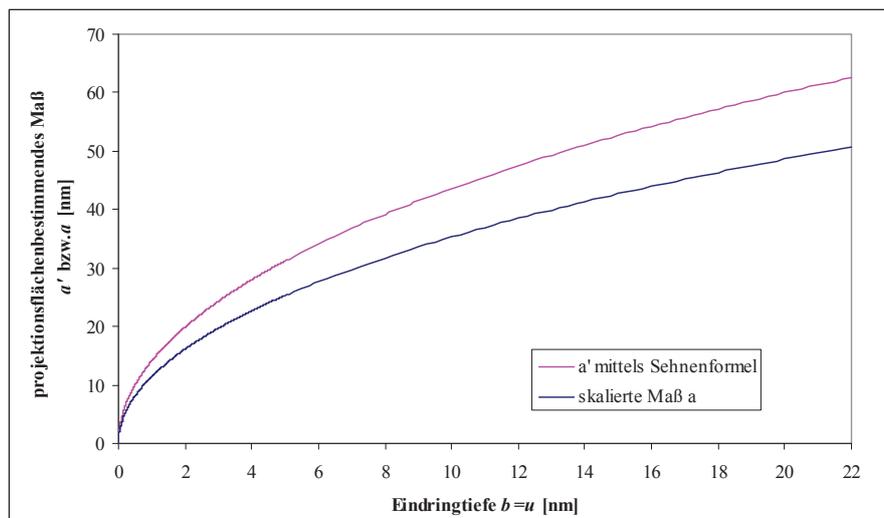


Abbildung 61: Die nichtlinearen Zusammenhänge Eindringtiefe b (bzw. u) zu projektionsflächenbestimmendem Maß a' (kein Sink-in Effekt berücksichtigt, mittels Kreis-Sehnformel, magenta) bzw. a (mit Berücksichtigung des Sink-in Effektes, mittels Skalierungsfaktors, blau). Bei der Skalierung wird davon ausgegangen, dass der Sink-in Effekt mit steigender Eindringtiefe kontinuierlich zunimmt.

Mit Hilfe der ermittelten projizierten Länge $l = 2a$, ergibt sich beispielsweise für die maximale Eindringtiefe, eine projizierende Fläche von:

$$A_{Kante} = l \cdot d = 101.4 \cdot 10^{-9} \cdot 1 \cdot 10^9 = 1.014 \cdot 10^2 m^2$$

Nimmt man den Berkovich-Pyramidenspitzes in erster Näherung als rotationssymmetrischen gerundeten Kegelstumpf an, erhält man als projizierende Fläche bei gleicher Eindringtiefe, einen Kreis mit dem Durchmesser $l = 2a$ und somit als Fläche:

$$A_{Berkovich} = a^2 \cdot \pi = \frac{l^2 \cdot \pi}{4} = \frac{(101.4 \cdot 10^{-9})^2 \cdot \pi}{4} = 8.075 \cdot 10^{-15} m^2$$

5.1.4 Ermittlung der validierbaren Indenterkraft

Die von ABAQUS erhaltene und der Auswertung zugrunde liegende Kraft-Eindringtiefen-Kurve kann **Abbildung 62** entnommen werden. Dazu wurde in ABAQUS der Kontaktknoten unterhalb der Indenterspitze (10010101) als Ausgabepunkt für Knotenpunktverschiebung (u_2) und der Referenzpunkt des Indenters (RP) als Ausgabepunkt für die Indentationskraft (Reaktionskraft RF_2) gewählt und die ermittelten Werte visualisiert.

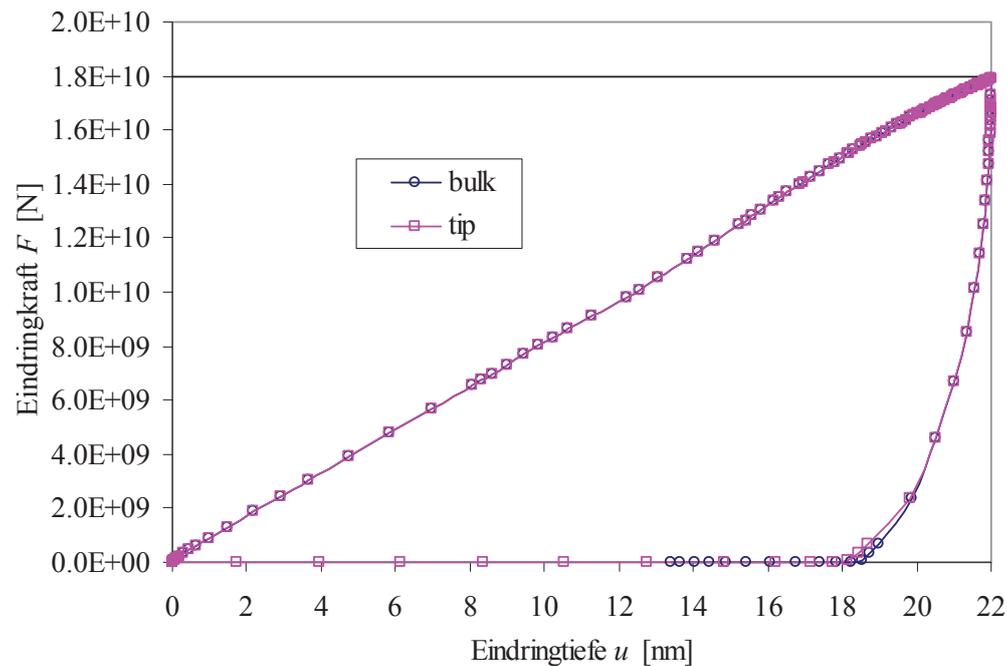


Abbildung 62: Von ABAQUS ausgegebene "Load-Displacement-Curve" für den Referenzpunkt (RP) des Indenters (tip, magenta) sowie für den Materialknoten 10010101 genau unterhalb des Indenters (bulk, königsblau). Eine genauere Erläuterung zu den Kurvenverläufen kann **Kapitel 5.1.6** entnommen werden.

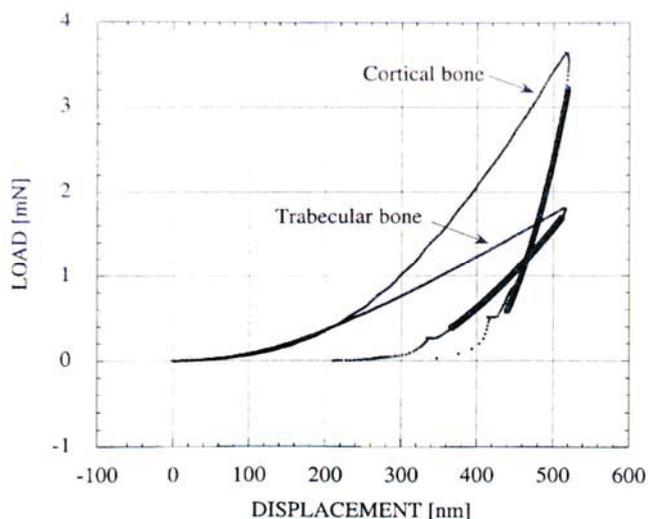


Abbildung 63: Aus Versuchen ermittelte "Load-Displacement-Curve" für kortikalen und trabekulären Knochen. Gut zu erkennen sind die typischen Größenordnungen bei solchen Indentationsversuchen [16].

Wie aus **Abbildung 63** sofort gesehen werden kann, werden im Versuch für Eindringtiefen von einigen 100 nm Indentationskräfte von einigen mN gemessen [16]. Die Abweichungen des Simulationsergebnisses lassen sich aus den im vorangegangenen Kapitel gemachten Geometriebetrachtungen ableiten. Verwendet man die dort angeführte, etwas vereinfachte Korrektur (Skalierungsfaktor), kommt man auf die in **Abbildung 64** dargestellte Kraft-Eindringtiefen-Kurve.

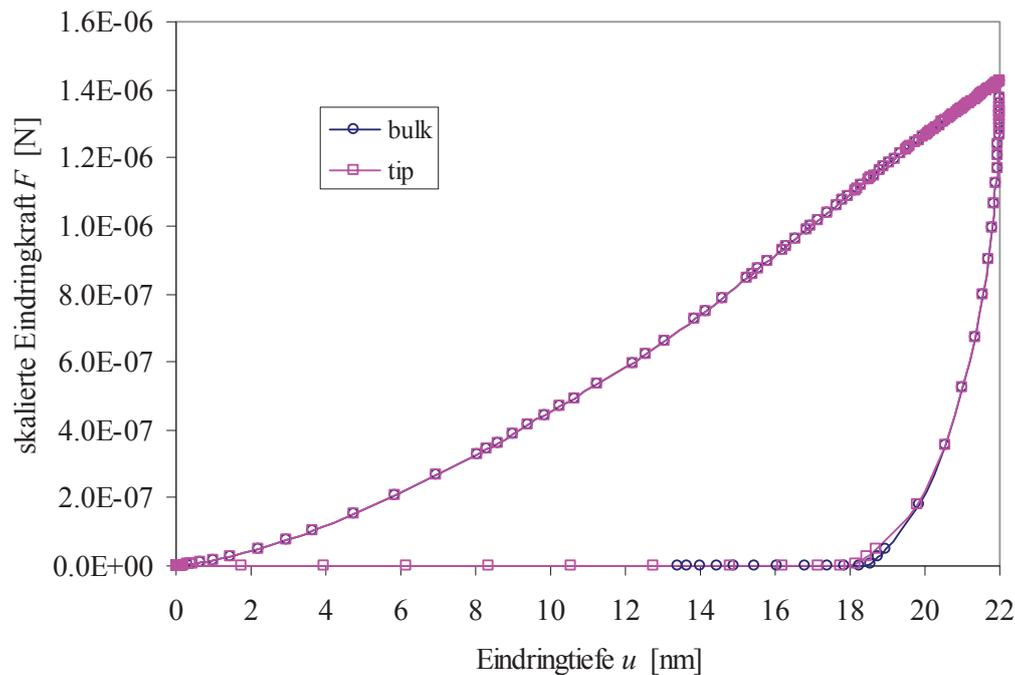


Abbildung 64: Die von ABAQUS ausgegebene "Load-Displacement-Curve" skaliert für die idealisiert gedachte Berkovich-Pyramidenspitze für den Referenzpunkt (RP) des Indenters (tip, magenta) sowie für den Materialknoten 10010101 genau unterhalb des Indenters (bulk, königsblau). Eine genauere Erläuterung zu den Kurvenverläufen kann **Kapitel 5.1.6** entnommen werden.

Für die maximale Indentationskraft bedeutet dies:

$$\sigma = \frac{F_{Kante}}{A_{Kante}} = \frac{F_{Berkovich}}{A_{Berkovich}} \Rightarrow$$

$$F_{Berkovich} = F_{Kante} \cdot \frac{A_{Berkovich}}{A_{Kante}} = F_{Kante} \cdot \frac{l^2 \cdot \pi}{4 \cdot l \cdot d} = 1.8 \cdot 10^{10} \cdot \frac{101.4 \cdot 10^{-9} \cdot \pi}{4 \cdot 10^9} = 1.43 \cdot 10^{-6} \text{ N} = 1.43 \cdot 10^{-3} \text{ mN}$$

Dieser nun sehr kleine Wert lässt sich damit erklären, dass wir mit 22nm eine nur sehr kleine Eindringtiefe simulieren können. Laut **Abbildung 63** erwartet man dementsprechend eine kleinere Kraft, womit sich das Simulationsergebnis bestätigt.

5.1.5 Ermittlung der "wahren Härte"

Damit ergibt sich für die verwendeten Materialkennwerte und Geometriedaten entsprechend Gleichung [30] eine "wahren Härte" von:

$$H = \frac{F}{A} = \frac{F}{\frac{l^2 \cdot \pi}{4}} = \frac{4 \cdot 1.43 \cdot 10^{-6}}{(101.4 \cdot 10^{-9})^2 \cdot \pi} = 1.77 \cdot 10^8 \frac{N}{m^2} = 177 MPa$$

Mit Hilfe von Gleichung [21] und dem Hinweis, dass der ermittelte Sink-In-Effekt nur bei einem Verhältnis von $\Theta > 0.03$ auftritt, lässt sich die obere Grenze des Proben-E-Modul abschätzen:

$$\Theta = \frac{H}{E} \Rightarrow E = \frac{H}{\Theta} = \frac{177}{0.03} \leq 5.9 GPa$$

Eine genauere Bestimmung des Proben-E-Moduls lässt sich mit Hilfe der Auswertung nach Oliver/Pharr durchführen.

5.1.6 Auswertung nach Oliver/Pharr – Indentationsmodul

Man erkennt an der Indentationskurve, dargestellt in **Abbildung 65**, dass es nach dem Belastungskurvenabschnitt, welcher die elastisch-ideal plastische Materialverformung widerspiegelt, bei der Entlastung nur noch zu einer Rückverformung der elastisch verformten Gebiete kommt. Diese wird von den plastifizierten Gebieten behindert, wodurch der gekrümmte Kurvenverlauf erklärt werden kann. Gegen Ende der Entlastung kommt es zur Trennung der Indenterspitze von der Bulk-Oberfläche. Dass weiterhin eine Kraft auf den Indenter wirkt, wie in **Abbildung 66** zu sehen ist, kann darauf zurückgeführt werden, dass die Mineralplättchen bei der Rückverformung ihre Form ändern und der Kontaktpunkt daher nicht mehr ident mit dem betrachteten Kontaktknoten unterhalb der Indentationsspitze (10010101) ist. Begründet ist dies in ihrem reinelastischen Materialverhalten sowie dem

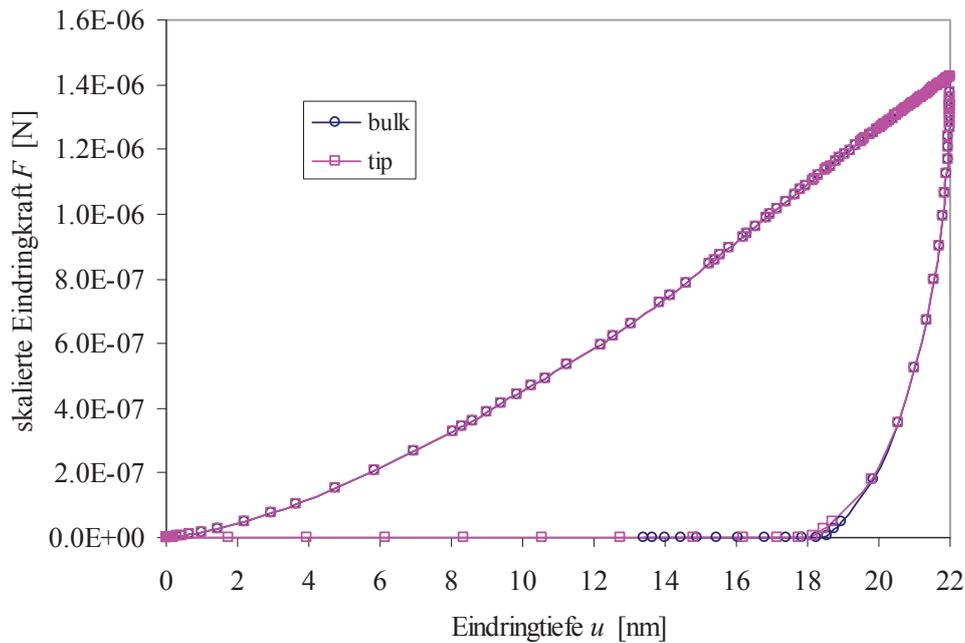


Abbildung 65: "Load-Displacement-Curve" für den Referenzpunkt (RP) des Indenters (tip, magenta) sowie für den Materialknoten 10010101 genau unterhalb des Indenters (bulk, königsblau). Auffällig dabei ist, dass sich der Indenter bereits vor der maximalen, elastischen "Rückverformung" von der Materialoberfläche löst (bei etwa 20nm). Unerwarteterweise kommt es, ebenfalls wegen der rückformungsverhindernden plastischen Verformung der Mineralplättchen, auch zu einer "kraftlosen" Rückverformung (von etwa 18nm bis 13nm) des Bulk-Materials.

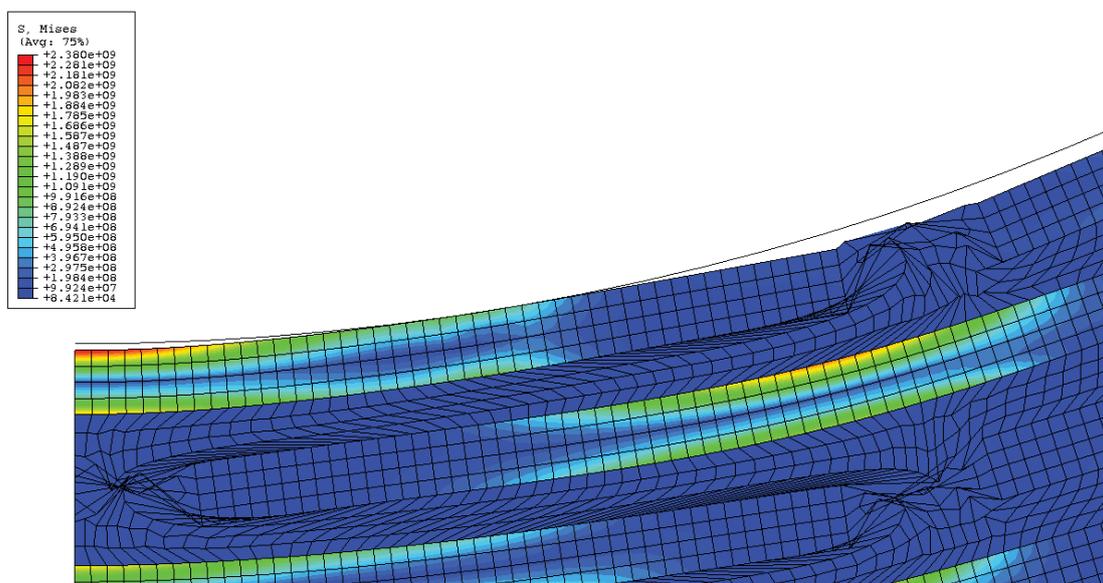


Abbildung 66: Der letzte "Kontaktpunkt" des Indenters mit dem Mineralplättchen liegt aufgrund der angenommenen Materialgesetze sowie der auftretenden Verformungen und Spannungen nicht direkt unter dem Indenter. Dieser Umstand führt zu den in **Abbildung 62** gezeigten Kurvenverläufen.

ideal-elastisch-plastischen Materialverhalten der Kollagenbereiche. In beiden Materialien ist eine (elastische) "Verformungsspannung" gespeichert, die während der Rückverformung abgebaut wird. Die plastifizierten Teile des Kollagens möchten dies verhindern, behindern daher diese Rückverformung und üben dabei Kräfte auf die Mineralplättchen aus. Obwohl, wie in **Abbildung 66** gezeigt, der letzte Kontaktpunkt zwischen Indenter und Bulk-Material nicht exakt unterhalb der Indenterspitze liegt, treten dort aufgrund der größten Verformungen die höchsten Spannungen auf. Sieht man sich den Verlauf der Entlastungskurve etwas näher an und legt, wie in **Abbildung 67** gezeigt, eine Tangente an den höchsten Punkt der Kurve, erhält man deren Steigung S (Entlastungssteifigkeitswert). Dazu wird eine Fittkurve durch die Datenpunkte gelegt, diese abgeleitet und am gegebenen Punkt ausgewertet. Verwendet man als erste Näherung die beiden obersten Datenpunkte und macht eine lineare Regression, erhält man den angegebenen Wert von:

$$S = 0.000244 \frac{N}{nm} = 0.000244 \cdot \frac{1}{10^{-9}} = 2.44 \cdot 10^5 \frac{N}{m}$$

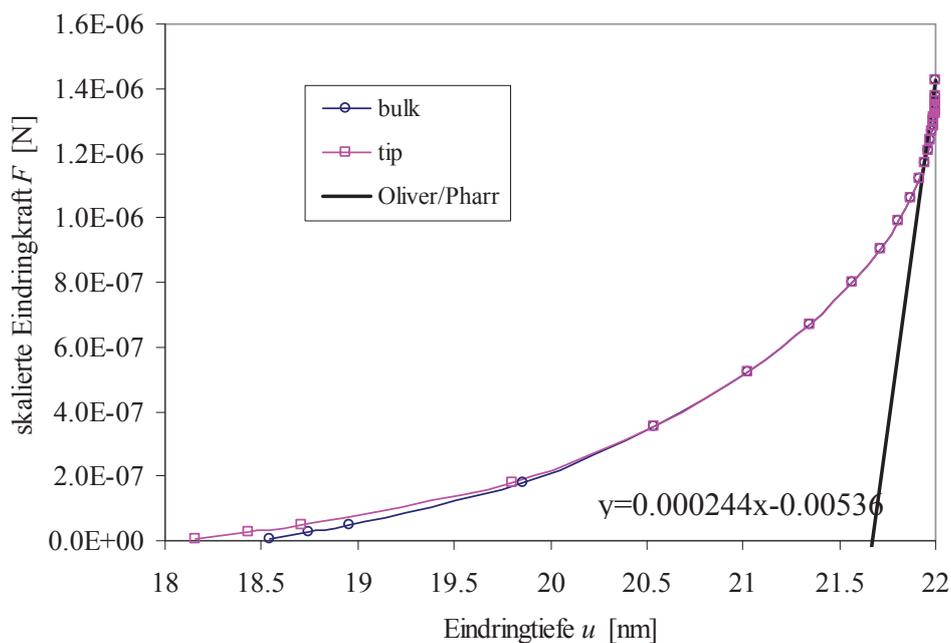


Abbildung 67: Entlastungskurvenabschnitt mit der für die Oliver/Pharr-Auswertung erforderlichen Tangente samt zugehöriger Geradengleichung.

Mit diesem Entlastungssteifigkeitswert S kann man nun mittels Gleichung [24] den Indentationsmodul nach Oliver/Pharr berechnen:

$$E_{OP} = E_r = \frac{\sqrt{\pi}}{2} \cdot \frac{S}{\sqrt{A}} = \frac{\sqrt{\pi}}{2} \cdot \frac{S}{\sqrt{l^2 \cdot \pi}} = \frac{S}{l} = \frac{2.44 \cdot 10^5}{101.4 \cdot 10^{-9}} = 2.4 \cdot 10^{12} \frac{N}{m^2} = 2400 GPa$$

Wie zu sehen ist, liegt dieser Wert weit über dem, in **Kapitel 5.1.5** aus dem für einen Sink-In-Effekt als Grenze gegebenen Verhältnis berechneten Modulwert von $E \leq 5.9GPa$. Wie aus **Abbildung 63** gesehen werden kann, wird der Entlastungssteifigkeitswert für höhere Eindringtiefen noch höhere Werte erreichen. Der ermittelte, zu hohe Wert kann damit erklärt werden, dass die verwendete Formel nach Oliver/Pharr für einen dreidimensionalen Berkovich-Pyramidenversuch gedacht ist, die Simulation allerdings zweidimensional durchgeführt wurde. Die versuchte, einfache "Umrechnung" ist scheinbar nicht zulässig, worauf zurückzuführen ist, dass die Mineralplättchen auch in der dritten Dimension gestaffelt sind, wie dies aus **Abbildung 35** ersichtlich ist. Das "Gesamtsystem" wird durch eine solche Anordnung viel flexibler, da die in Dickenrichtung "dahinterliegenden" Mineralplättchen sich ebenfalls leichter verformen als die in der 2D-Simulation verwendeten 1m langen "Mineralplattenlatten".

5.1.7 Ergebnisdiskussion und Vergleich mit Literaturwerten

Wie bereits ausführlich geschildert und anhand von **Abbildung 63** erläutert, konnten keine mit der Realität vergleichbaren Simulationsergebnisse erhalten werden. Gründe hierfür sind:

- Die unzureichende Eindringtiefe bedingt durch
 - die um 2 Zehnerpotenzen unterschiedlichen E-Moduli.
 - die geringe Simulations-Geometriehöhe.
 - die "unzureichenden" Remeshing-Werkzeuge von ABAQUS.
- Die begrenzte Elementanzahl in ABAQUS
 - die keine 3D-Simulation ermöglichte, welche für die Indenterform sowie die "Geometrieschichtung" (Anisotropie) notwendig wäre.
- Die numerischen Fehler ausgelöst durch
 - die starken Elementverzerrungen.

Die Rechenleistung der verwendeten Hardware stellt kein wirkliches Hindernis dar, da diese für eine größere Anzahl von Elementen ausreichen würde. Die starken Elementverzerrungen verhindern allerdings die Ergebnisfindung. Würden die dadurch verursachten Divergenzen nicht auftreten, müsste man bei Elementzahlerhöhung allerdings dennoch mit sehr langen Simulationslaufzeiten rechnen. Für die gewählten Material- und Geometriedaten liegt die Grenze für konvergierende Simulationsergebnisse bei einer Eindringtiefe von etwa 22nm, ab

dieser Grenze kommt es zu dermaßen starken Elementverzerrungen, die es ABAQUS unmöglich machen weiter zu konvergieren. Auch mittels ABAQUS-explicit und dem dort verfügbaren Remeshing-Werkzeug kann, wie **Abbildung 68** zeigt, die Elementverzerrungen nicht in dem Maße reduziert werden, dass eine größere Eindringtiefe erreicht werden kann. Man erkennt zwar eine deutliche Minimierung der Elementverzerrungen, diese Verbesserungen können allerdings die stärker werdenden Elementverzerrungen nicht mehr ausgleichen. Darüber hinaus ist eine solche Simulation auch deutlich zeitintensiver. Ein genauer Faktor kann, da auf unterschiedlichen Rechenanlagen und mit unterschiedlichen ABAQUS-Versionen gerechnet wird, nicht exakt angegeben werden, aber je nach "Dämpfungsparametern" liegt dieser bei etwa 3 bis 6.

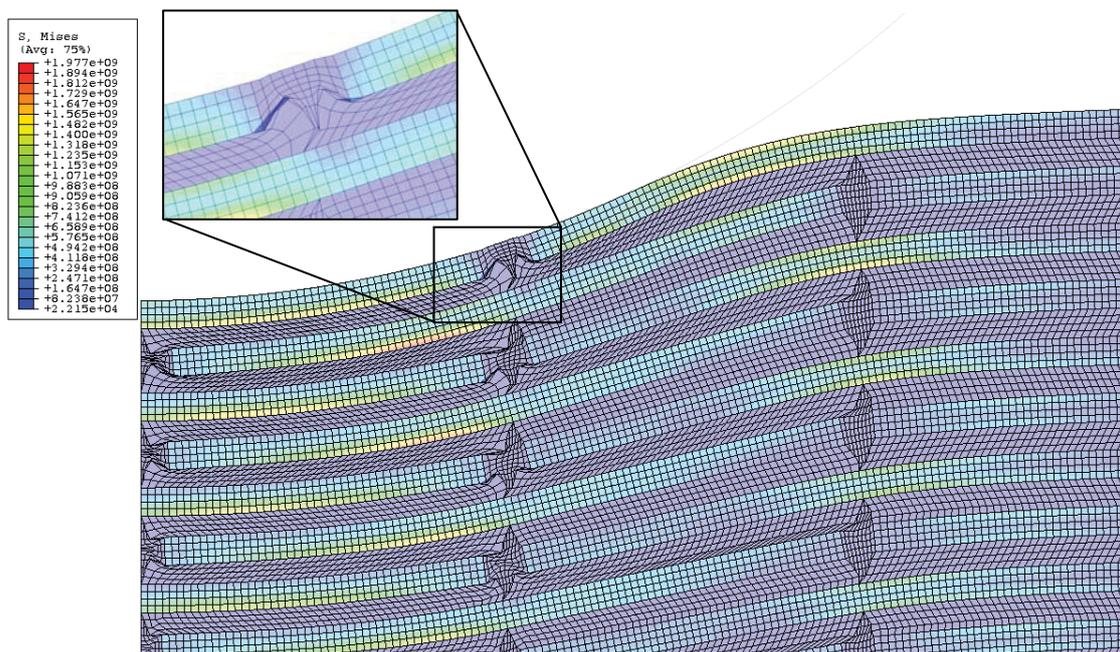


Abbildung 68: Auch mit ABAQUS-explicit können, obwohl mit dem dort verfügbaren Remeshing-Tools die Elementverzerrungen etwas minimiert werden können, keine größeren Eindringtiefen erzielt werden. Die transparent-blauen Dreiecke in der Detailansicht stellen "real (physikalisch) unmögliche Überlagerungen" von Kollagenelementen mit Mineralplättchenelementen dar.

Die starken Elementverzerrungen legen die Vermutung nahe, dass die Kollagenfasern in der Realität extremen Belastungen ausgesetzt sind. Es kann daher davon ausgegangen werden, dass bei einem biologischen Material wie etwa Knochen, da hier der Aufbau in der Regel nicht geometrisch perfekt (exakte Reihen,...) gestaltet ist, zu einem doppelten Versagen kommen wird. Einerseits wird in den Mineralplättchen die theoretische Festigkeit erreicht/überschritten und es kommt zum Bruch, andererseits werden die Kollagenfasern die starken Dehnungen und Verzerrungen (vor allem die Dehnungen über die

Mineralplättchenkanten, auch wenn diese in der Realität nicht so scharfkantig ausgebildet sein dürfte) nicht ertragen und ihrerseits zerstört werden. Ein Knochen stellt damit, wie Gao et al. [17] festgestellt hat, einen idealen Verbundwerkstoff dar. Bei einem solchen ist es am günstigsten, wenn beide Komponenten zur selben Zeit versagen, da es keinen Sinn macht, wenn eine Komponente mehr Belastung verträgt als die andere, da diese gemeinsam einen Werkstoffverbund darstellen. Die Kette ist nur so stark wie ihr schwächstes Glied, also ist es "ökonomisch" sinnvoll, wenn alle Kettenglieder gleich stark sind.

5.2 Indentation parallel zu den Mineralplättchen

Es gilt hier dasselbe wie unter **Kapitel 5.1** geschrieben, wobei in diesem Fall die Ergebnisfindung sich noch komplizierter darstellt, wie das die Ergebnisse in **Abbildung 69** und **Abbildung 70** verdeutlicht. Die Verzerrungen des Kollagens sowie des gesamten Systems sind, bedingt durch die starren Mineralplättchen (Trägheitsmoment $I = \frac{b \cdot h^3}{12}$ wobei nun $h = 60\text{nm}$ und $b = 5\text{nm}$), noch größer. Aufgrund dieser "Starrheit" steigen, wie in **Abbildung 69** ersichtlich, auch die Spannungen innerhalb der Mineralplättchen sehr schnell stark an.

Wird der Indenter weiter abgesenkt, kommt es bei dieser Mineralplättchenanordnung zu noch stärkeren Kollagenelementverzerrungen und dies bereits bei noch geringeren Eindringtiefen. Neben diesen starken Verzerrungen kann man, wie in **Abbildung 70** gezeigt, sogar erkennen, dass Mineralplättchenelemente in den Indenter "eindringen". Dies ist darauf zurückzuführen, dass man in ABAQUS bei Kontaktproblemen zwei Oberflächen definieren muss, eine als "Master" sowie eine als "Slave", wobei ABAQUS nun sicherstellt, dass keiner der Knoten der "Slave"-Oberfläche die "Master"-Oberfläche "durchdringt". Dieses Konzept ist auch bei den in **Abbildung 70** dargestellten Ergebnis umgesetzt, allerdings sind die Verzerrungen dermaßen groß, dass die unter der "Slave"-Oberfläche liegenden Bulk-Knoten, für welche dieses Konzept nicht gilt, den Indenter nicht "wahrnehmen". Deren Belastung/Verschiebung ist nur von den Kräften und Verschiebungen seiner Nachbarknoten abhängig. Eine Definition des gesamten Bulk-Knoten als "Slave"-Knoten könnte diesen Effekt möglicherweise verhindern. Da eine solche Möglichkeit innerhalb von ABAQUS im Moment nicht bekannt ist, kann dieser Einfluss nicht ermittelt bzw. abgeschätzt werden.

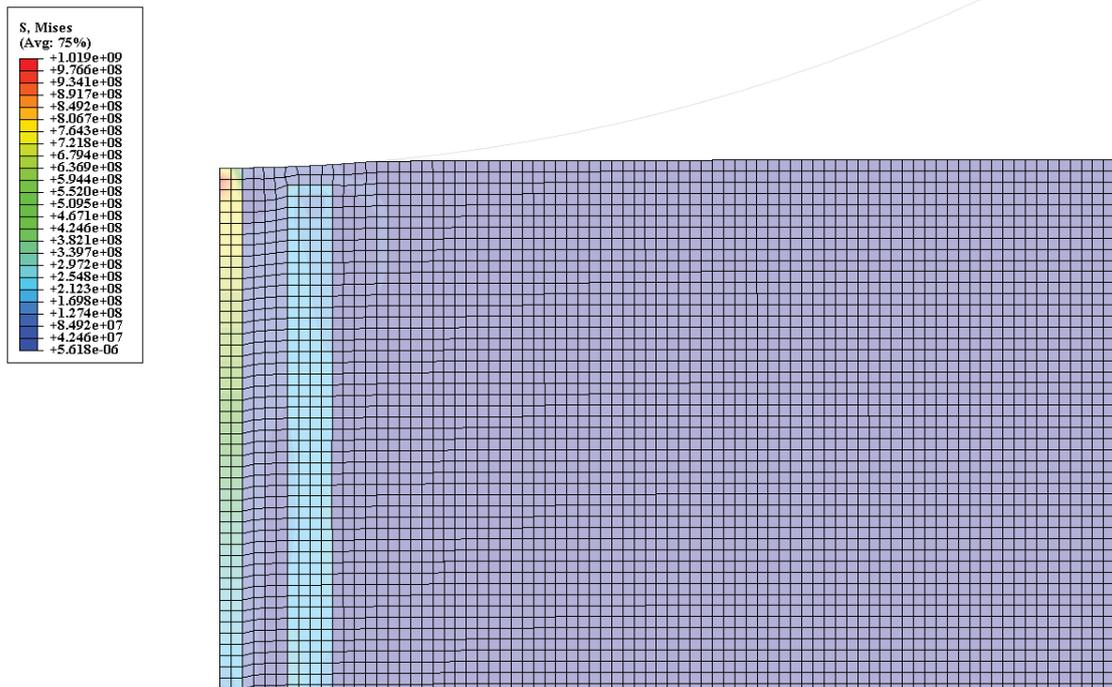


Abbildung 69: Bereits bei sehr kleinen Eindringtiefen kommt es zu sehr starken Spannungen innerhalb der Mineralplättchen. Auch erkennt man bereits die stärker werdenden Elementverzerrungen der Kollagenelemente welche zwischen den "starrten" Mineralplättchen und dem Indenter "eingequetscht" werden.

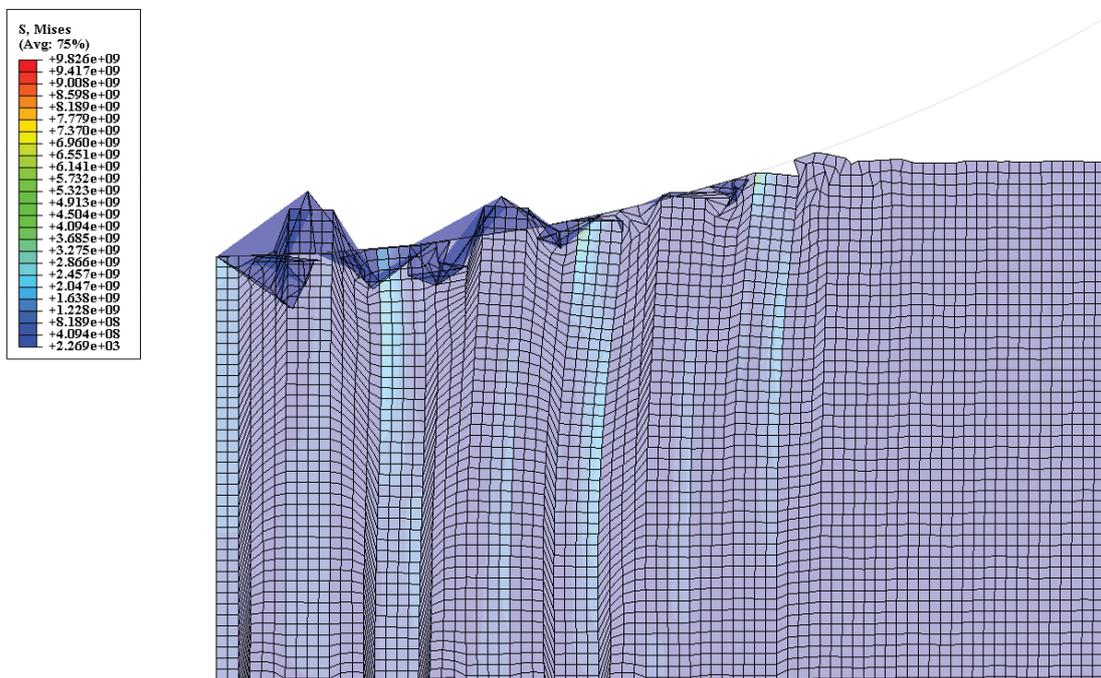


Abbildung 70: "Abbruchergebnis" einer Simulation für eine Indentation parallel zu den Mineralplättchen. Die Elementverzerrungen sind dabei noch eklatanter als bei der Indentation rechtwinkelig zu den Mineralplättchen (**Kapitel 5.1**). Die transparent-blauen Flächen stellen eine "real unmögliche Überlagerung" von Kollagenelementen untereinander bzw. mit Mineralplättchenelementen dar.

5.2.1 Auswertung nach Oliver/Pharr – Indentationsmodul

Aufgrund der noch stärkeren Verzerrung konnte keine Simulation mit einem gesamten Be- und Entlastungszyklus durchgeführt werden. Anzunehmen ist allerdings, dass der E-Modul in einer solchen Anordnung wesentlich höher ist, da aufgrund der "Steifigkeit" einer solchen Anordnung wesentlich höhere Kräfte angewandt werden müssten um eine Verformung zu erzielen. Die Kraft-Eindringtiefen-Kurve eines solchen Versuches würde daher wesentlich größere Steigungen für die Be- als auch die Entlastung aufweisen. Die erreichbare Eindringtiefe bis zum Simulationsabbruch könnte wahrscheinlich durch eine Vergrößerung der Geometriehöhe etwas erhöht werden, es ist allerdings zu befürchten, dass dieser Effekt marginal ist.

5.2.2 Ergebnisdiskussion und Vergleich mit Literaturwerten

Wie im vorangegangenen Kapitel festgestellt, liegen keine auswertbaren Ergebnisse vor.

6 Zusammenfassung und Schlusswort

Innerhalb der vorliegenden Arbeit war es nicht möglich, aussagekräftige Resultate zu liefern um diese mit Literaturwerten zu vergleichen. Für zukünftige Studien könnten weitere Parameterstudien (Elementtypenkombinationen, Adaptierungsparameter für "adaptiv mesh",...) nützlich sein, um diese Ziele zu erreichen. Möglicherweise würden sich andere Softwarepakete, die speziell für große Verformungen ausgelegt sind wie etwa DEFORM (Scientific Forming Technologies Corporation, Columbus, Ohio) besser eignen, das zur Behandlung stehende Problem zu lösen.

Zur Beantwortung der forschungsleitenden Fragestellungen (**Kapitel 1.2**) wurden Auswertungen von Indentationssimulationen durchgeführt, dessen Ergebnisse sich wie folgt zusammenfassen lassen:

- Bei einer Indentation senkrecht zu den Mineralplättchen, kann für die getroffenen Prozessparameter, Material- und Geometriedaten eine Indentationstiefe von rund 22nm erreicht werden, bei weiterem Eindringen wird die Verformung der Elemente zu groß, was eine Aussagekraft der Daten in Zweifel zieht bzw. es kommt zum Abbruch der Simulation aufgrund Divergenz.
- In **Kapitel 5.1** wurde eine Simulation mit 22nm Indentationstiefe ausgewertet (die realen Indentationsversuche werden allerdings mit Indentationstiefen von rund 100nm - 600nm durchgeführt).
- Die Mineralplättchen erreichen bei der Simulation unterhalb der Indenterspitze, auch bereits bei kleineren Eindringtiefen, ihre theoretische Festigkeit von $E/30$.
- Die Kollagenelementverzerrungen in der Simulation legen es nahe, dass die Kollagenfasern in der Realität unterhalb der Indenterspitze versagen werden bzw. geschädigt werden.
- Ein nach dem "staggered Model" aufgebauter Verbundwerkstoff (z.B. Knochen) stellt, wie Gao et al. [17] festgestellt hat, einen idealen Verbundwerkstoff dar, da es bei einem solchen am günstigsten ist, wenn beide Komponenten zur selben Zeit versagen.
- Bei einer Indentation parallel zu den Mineralplättchen (**Kapitel 5.2**) kommt es für die getroffenen Prozessparameter, Material- und Geometriedaten bereits nach geringer Indentationstiefe zu extremen Elementverzerrungen, die eine Auswertung unmöglich machen.

Abbildungsverzeichnis

- Abbildung 1:** Schematische Darstellung des Knochenaufbaues und Knochenabbaues durch die Osteoklasten und Osteoblasten ohne (links) und mit (rechts) Osteoporose-Erkrankung [<http://www.osteoporose.com>]. 2
- Abbildung 2:** Aufnahmen von gesunden (links) bzw. von erkrankten (rechts) Knochentrabekelnetzwerkes (entspricht Ebene 4 der in **Kapitel 1.1.5** erwähnten Einteilung) [<http://www.osteoporose.com>, APOTHECUM Herbst 2004]. 3
- Abbildung 3:** Ein weißer Hydroxyapatitkristall [<http://webmineral.com>]. 5
- Abbildung 4:** hexagonal-dipyramidale Kristallstruktur des Hydroxyapatits 6
- Abbildung 5:** Raster-Sondenmikroskopaufnahme (AFM) einiger Apatitkristalle in der Kollagenfasermatrix. Die linke Aufnahme wurde mit dem "kontaktlosen Modus" (20nm über der Oberfläche) erstellt, wobei das rechte Bild eine Aufnahme mit dem "Tapping Modus" zeigt. Die Apatitkristalle lassen sich dabei als helle Flächen und die Kollagenfibrillen als dunkle Areale erkennen [18]. 6
- Abbildung 6:** Bildung von Kollagen Fibrillen (Collagen Fiber) aus Kollagenmolekülen (Collagen) und dessen Ausgangsbaustein innerhalb der Zelle (Tropokollagen) [<http://labs.ansci.uiuc.edu>]. 7
- Abbildung 7:** Der Aufbau eines Kollagenmoleküls. Dabei winden sich drei α -helikale Kollagenmoleküle in Form einer Superhelix umeinander [www.mad-cow.org] [de.wikipedia.org]. 8
- Abbildung 8:** Unterteilung und Auftreten der 19 unterschiedlichen Kollagentypen [2]. 9
- Abbildung 9:** REM-Bild einzelner Kollagenfasern links [www.wellesley.edu] sowie eines Mäuseherzens rechts [www.sdsc.edu]. 10
- Abbildung 10:** REM-Bild der Mikroarchitektur der tragenden Struktur (Trabekelstruktur) eines Knochens [Prof. Dr. Hans-Jürgen Pesch, Erlangen]. 10
- Abbildung 11:** Schematische Darstellung des Aufbaus einer hierarchischen Struktur am Beispiel eines spongiösen Menschenknochens [21]. 11
- Abbildung 12:** Die Unterteilung in die 6 hierarchischen Ebenen des Knochens[2]. 11
- Abbildung 13:** Schematische Darstellung des Aufbaus eines Nanoindenters [www.nanoindentation.cornell.edu]. 12

- Abbildung 14:** Hodge-Petruska Schema für unmineralisiertes Kollagen (a) sowie mögliche parallel-serielle (b-c) bzw. staggered (d) Mineralplattenanordnungen mit zugehörigen, auf reines Kollagen normierten Materialkennwerten [5]. 13
- Abbildung 15:** Detailliertere Skizze des Hodge-Petruska Schemas für eine unmineralisierte Kollagenfibrille. 13
- Abbildung 16:** Kollagenmolekülabstand über reziproker, auf den nassen Zustand normierter Dichte unterschiedlicher organischer Materialien (Gewebe). Die oberen Punkte beziehen sich auf nasse (feuchte) Proben während die unteren Messpunkte mit trockenen Proben durchgeführt wurden. Die Messpunkte der trockenen Proben streuen zwar mehr, haben allerdings für alle Gewebe etwa den gleichen Wert von 1.1nm. Zum Vergleich kamen Schienbeinknochen einer Kuh (cow tibia), Hirschgeweihstangen (deer antler), Truthahnsehnen (Turkey tendon), Fischgräten (fish clythrum) und Schienbeinknochenmatrix einer Kuh (cow tibia matrix) [Lees et al., 1984]. 14
- Abbildung 17:** Kollagenfibrillenordnung ($n = 30$) im overlap-Bereich in nassem (wet), trockenem (dry) und 100% mineralisiertem (mineralized) Zustand. Gut zu erkennen ist die Reduktion des Kollagenmolekülabstandes bei der Trocknung auf 1.1nm (dies entspricht dem Kollagenfibrillendurchmesser). Im ideal vollkommen mineralisierten Zustand sind alle im feuchten Zustand mit Wasser gefüllten Abstände nun mit Mineral gefüllt und daher ist der Kollagenmolekülabstand mit 1.55nm gleich dem nassen Zustand. 15
- Abbildung 18:** Kollagenfibrillenordnung ($n = 30$) im gap-Bereich in nassem (wet), trockenem (dry) und 100% mineralisiertem (mineralized) Zustand. Abweichend von **Abbildung 17** ist nun statt jeder fünften Kollagenfibrille aufgrund der Periodizität ebenfalls Hydroxyapatit vorhanden, d.h. der Mineralisierungsgrad in den gap-Bereichen ist höher. 15
- Abbildung 19:** Prinzipskizze des Voigt-Modells im ent- und belasteten Zustand mit den zugehörigen Hauptzusammenhängen. 19
- Abbildung 20:** Prinzipskizze des Reuss-Modells im ent- und belasteten Zustand mit den zugehörigen Hauptzusammenhängen. 20
- Abbildung 21:** Skizze des "Staggered Model", wobei die Mineralplattenbereiche grau und die Kollagenbereiche blaugrün dargestellt sind. Jene Kollagenbereiche, die Normalspannungen übertragen, sind dunkelblau und jene Kollagenbereiche, die Schubspannungen übertragen, sind hellblau dargestellt. 20

- Abbildung 22:** Die auf unmineralisierten Kollagen normierten Materialkennwerte E-Modul E' , mit den dazugehörigen Schub- (shear) und Zug-(tensile)anteil, maximale Dehnung ε_{max} und maximale Spannung σ_{max} bei Variation der Plättchenabstände a und b für Plättchenabmessungen von $d = 3.5\text{nm}$ und $l = 70\text{nm}$ sowie einen Mineralisierungsgrad von $\Phi = 0.42$. Die Buchstaben stehen dabei für die Grenzfälle $a = 1$ (R), $a = 2 \cdot b$ (S) und $a = 0$ (T) des Modells [5]. 21
- Abbildung 23:** Skizze des "sink-in"-Effekts, wobei die Probenoberfläche strichpunktiert und der Indenter bei Maximallast strichliert dargestellt ist. Das Maß a repräsentiert die Projektionsfläche A und die auftretenden Tiefen sind die maximale Eindringtiefe u_{max} (kombiniert aus der Kontakttiefe u_c und der Oberflächenverformung u_s) sowie die bleibende Eindringtiefe u_f . 23
- Abbildung 24:** Skizze des "pile-up"-Effekts, wobei die Probenoberfläche strichpunktiert und der Indenter bei Maximallast strichliert dargestellt ist. Das Maß a repräsentiert die Projektionsfläche A und die auftretenden Tiefen sind die maximale Eindringtiefe u_{max} (kombiniert aus der Kontakttiefe u_c und der Oberflächenverformung u_s) sowie die bleibende Eindringtiefe u_f . 24
- Abbildung 25:** Prinzip einer Kraft-Eindringtiefen-Kurve (load displacement curve). Mit steigender Kraft F steigt auch die Eindringtiefe u , wobei das Eindringen auch aufgrund der größer werdenden Kontaktfläche schwerer wird. Die maximale Eindringtiefe u_{max} verringert sich bei Entlastung aufgrund elastischer Rückverformung auf die bleibende Eindringtiefe u_f . S ist die Tangente an die Entlastungskurve, wobei S_{max} die Entlastungssteifigkeit (S bei u_{max}) darstellt. 25
- Abbildung 26:** Prinzip einer Kraft-Eindringtiefen-Kurve (load displacement curve) mit der maximalen Eindringtiefe u_{max} (bei Maximalkraft F_{max}) sowie der indenterformabhängigen (durch unterschiedliche ε -Werte wiedergegebenen) Kontakttiefe u_s . 26
- Abbildung 27:** Skizze der auftretenden Spannungskomponenten in einem kartesischen Koordinatensystem. Es wird deutlich, dass die Schubspannungen τ_{xy} und τ_{yx} gleich groß sein müssen (analog $\tau_{xz} = \tau_{zx}$, $\tau_{yz} = \tau_{zy}$), da sonst die Volumenzelle an der Kante "aufreißen" würde. Dies führt zu einer Symmetrie, die als Transformationsregel aus der Matrix einen Tensor macht. Der Transformationstensor ist dadurch definiert, dass sich in einem Koordinatensystem seine Komponenten auf bestimmte Weise transformieren lassen. Der Spannungstensor T ist darüber hinaus ein Tensor (lat.: ‚tendo‘ =

- "ich spanne") zweiter Stufe, da er eine quadratische Matrix darstellt. (Ein Tensor erster Stufe wäre ein Vektor und Tensor nullter Stufe ein Skalar.) Somit repräsentiert jeder Tensor eine Matrix, aber nicht jede Matrix ist ein Tensor. 28
- Abbildung 28:** Einfaches Beispiel einer stückweise linearen (oben links) bzw. quadratischen (oben rechts) Ansatzfunktion Ψ für einen 1D-Fall sowie einer stückweise linearen Ansatzfunktion für einen 2D-Fall (unten). [8] 31
- Abbildung 29:** Der verwendete Arbeitsplatzrechner. Gut zu erkennen sind die Wechselrahmen für die im RAID verwendeten Festplatten sowie die für lange Rechenzeiten notwendige Kühlung (IP-Adresse: 213.229.22.92). 38
- Abbildung 30:** Der Applikationsserver der Montanuniversität (zalpha1.unileoben.ac.at, IP-Adresse: 193.171.87.200). 39
- Abbildung 31:** Der SGI®-Altix®350 Shared-Memory-Server des Lehrstuhl für "Simulation und Modellierung metallurgischer Prozesse" (SMMP) (IP-Adresse: 192.168.1.204). 39
- Abbildung 32:** Das einfach zu bedienende GUI des SSH Secure Shell arbeitet ähnlich wie der Windows Explorer. Besonders sei hier auf die aufgrund der Dateigröße bereits fehlerhafte Größenanzeige der *.odb Datei hingewiesen (18446744072899.002.368!!). 42
- Abbildung 33:** Die Spannungs-Dehnungsverläufe der verwendeten Materialien. Man erkennt die großen Unterschiede im E-Modul (Steigung). Das ideal plastische Materialverhalten des Kollagens setzt bei 3% Dehnung ein. 44
- Abbildung 34:** Skizze einer Berkovich-Pyramidenspitze in Grund-, Auf- und Seitenriss sowie einer Isometrieansicht und ein Schnitt durch die Pyramidenspitze. 44
- Abbildung 35:** Skizze der 3D-Mineralplättchenanordnung. Die Mineralplättchen haben die Abmessungen $l = 100\text{nm}$, $b = 40\text{nm}$ und $d = 2.5\text{nm}$. 45
- Abbildung 36:** Skizze der 2D-Mineralplättchenanordnung. Die Mineralplättchen haben die Abmessungen $l = 100\text{nm}$ und $d = 2.5\text{nm}$. 45
- Abbildung 37:** Eine Auswahl der in ABAQUS zur Verfügung stehenden Elementtypen. Gut zu erkennen ist die unterschiedliche Anzahl von Knoten, je nach Ansatzfunktion (linear, quadratisch) [8]. 46
- Abbildung 38:** Verformung aufgrund des angelegten Momentes M . Exakte (analytische) Lösung (oben), Simulationsergebnis für volle Integration mit linearer Ansatzfunktion (z.B. CPS4, Kontinuumelemente (C..)) mit ebenem Spannungszustand (Plain Stress ..PS..) (Mitte links) sowie quadratischer

- Ansatzfunktion (z.B.CPS8) (Mitte rechts) als auch für reduzierte Integration mit linearer Ansatzfunktion (z.B.CPS4R) (unten) [8]. 46
- Abbildung 39:** Ausschnitt der gesamten Mineralplättchenanordnung in der Rechendomain unter Berücksichtigung der "Staggered" Anordnung. 48
- Abbildung 40:** Ein stark vereinfachter, erster 3D-Versuch. Der Indenter ist dabei noch mittels Kontinuumelementen vernetzt. 49
- Abbildung 41:** Das Grafik-User-Interface, (GUI) von ABAQUS/CAE 6.5. Gut zu erkennen ist, im "pull down Menü" in der linken Bildhälfte, der strukturierte Aufbau und rechts das Modellverzeichnis (model tree), welchem schrittweise gefolgt werden kann. Dies beginnt mit der Bauteilerstellung (Part) und der Materialdefinition (Property) sowie dem Modellaufbau (Assembly) usw. und geht hin bis zum Starten der Analyse (Job - die zur Abarbeitung ("Processing") gegebenen Aufgaben werden als Jobs bezeichnet) und dem Postprocessing (Visualization). 50
- Abbildung 42:** Netz und Spannungsfeld einer Indentation mit vernetztem Indenter und nicht gefüllten Modellbereichen (Zwickel) in denen es im Laufe der Simulation zu Selbstkontakt kommt. Zu sehen sind Probleme an der Kontaktfläche sowie stark verzerrte Kollagenelemente, die sogar über die Symmetrieebene hinaus stehen. Die Größenfarbskala wird, da es sich um einen Simulationsversuch handelt nicht angezeigt. 51
- Abbildung 43:** Die beim Oberflächenkontakt des vernetzten Indenters mit dem Komposite auftretenden Probleme. Die dabei ersichtliche Abweichung von der "regulären" Oberflächenform (Kontaktfläche) führt zur Divergenz der Simulation. 52
- Abbildung 44:** Darstellung desselben Ergebnisses mit unterschiedlicher Skalierung. Bei der Skalierung über die Größe aller vorhandenen Spannungen ($2.7e+2$ bis $2.4e+9$, Bild oben) lässt sich das Spannungsfeld innerhalb des Kollagens nicht und das der Mineralplättchen nur gering auflösen. Skaliert man auf die Größe der Spannungen im Kollagen ($3e+7$ bis $4.5e+7$ für den dargestellten Bildabschnitt, Bild unten links) erscheinen die Mineralplättchen zwar grau da die Spannungen generell höher liegen, aber der Farbverlauf innerhalb des Kollagens ist erkennbar. Für eine genauere Farbverlaufsverteilung innerhalb der Mineralplättchen ist eine andere Skalierung zu wählen ($5e+7$ bis $2.4e+9$, Bild unten rechts), wobei die Spannungsverläufe im Kollagen nicht mehr zu sehen sind, da diese aufgrund des E-Modulunterschiedes unterhalb dieses Minimalwertes liegt. 53

- Abbildung 45:** Die Indenterform wird mit fortlaufendem Simulationszeitpunkt angepasst. Hier wurde der Indenter für die ersten Simulationen noch voll mitvernetzt (links) bzw. um ein gleichmäßigeres Netz zu erhalten als "Schale" mit gleicher Netzgüte simuliert (Mitte). Um die Konvergenz zu verbessern, hat es sich als günstig erwiesen, den Indenter als "analytical rigid" also als starr anzunehmen (rechts). Diese Vereinfachung ist aufgrund der Materialdaten des Indenters ($E_{Indenter} = 1140\text{GPa}$ und $\nu_{Indenter} = 0.07$) durchaus zulässig. Erkennbar ist der für die Definition des starren Indenters notwendige Referenzpunkt (RP). 54
- Abbildung 46:** Indent in „homogenen“ Block mittels eines "rigid" Indenters. Gut zu erkennen ist, dass für tiefere Indentationen die Gesamtgeometrie vergrößert werden muss. Einerseits reicht die "Spannungsbirne", also der Bereich welcher Spannungen ausgesetzt ist, immer tiefer in das Material hinein und andererseits beginnt sich die rechte Außenseite merklich zu verformen. 55
- Abbildung 47:** Die Nummernzuordnung der Freiheitsgrade innerhalb von ABAQUS. Darüber hinaus gibt es noch Freiheitsgrade wie etwa Temperatur und/oder elektrisches Potential [8]. 55
- Abbildung 48:** Der Aufbau einer "Einheitszelle". Diese basiert auf dem Mineralplättchen MP (grau), den beiden darunter liegenden Kollagenstreifen CL und CR (cyan) sowie dem Kollagenzwickel CZ (cyan) rechts davon. 56
- Abbildung 49:** Die vier, anhand der im Text beschriebenen Definitionen, erhaltenen Zwickelformen. Die Form 1 tritt nur innerhalb der ersten Mineralplättchenreihe auf, die Zwickelform 3 ist nur in der letzten Mineralplättchenreihe zu finden, die Zwickelform 2 ist nur in der Symmetrieebene (also in geraden Mineralplättchenreihen) vorhanden und die restlichen Kollagenzwickel haben die Kollagenzwickelform 4. 57
- Abbildung 50:** Knoten- und Elementnummerierung zum Aufbau eines strukturierten Netzes. Die Ziffern sind die Knoten- bzw. Elementnummern. Es fällt der logische Zusammenhang auf, dass es bei einer Unterteilung immer eine Knotennummernreihe mehr als Elementreihen gibt. 58
- Abbildung 51:** Knotennummernvergabe und -gruppierung der vier Teilbereiche der Einheitszelle. Farblich markiert sind die zu verknüpfenden Knotengruppen (nsets) bzw. Einzelknoten. 59
- Abbildung 52:** Die bei zunehmender Eindringtiefe immer stärker werdenden Elementverzerrungen führen zu Simulationsabbrüchen bzw. nicht vollkommen glaubwürdigen Ergebnissen (hier Eindringtiefe etwa 30nm). 60

- Abbildung 53:** Ergebnis eines Eindringversuchs mittels ABAQUS/Explicit in ein homogenes Material. Bei dieser Simulation kam er durch ungeeignete Wahl der Dämpfungskonstanten zu einem wellenartigen "Wegfließens" des Materials. 62
- Abbildung 54:** Aufbau der analysierten Geometrie mit Darstellung, farbig markiert sind wie im Text beschrieben die unterschiedlichen Randbedingungen. 65
- Abbildung 55:** Teilausschnitt des verwendeten Netzes im Bereich der Indenterspitze. 66
- Abbildung 56:** Übersicht über den Be- und Entlastungszyklus. Die ersten 4 Bilder zeigen das Eindringen des Indenters bis zu einer maximalen Tiefe von 22nm (Bild5, 1.00s) während die letzten 3 Bilder die Entlastung zeigen. Gut zu erkennen ist die bleibende (plastische) Verformung und das dadurch auftretende Lösen der Indenterspitze vom Bulk-Material. Die Skalierung ist dabei von 0MPa bis $3e9$ Pa linear gewählt. 67
- Abbildung 57:** Übersicht über die Spannungsverteilung im Bulk-Material. Man erkennt, dass Spannungen bis tief in das Verbundmaterial (Bulk) vorhanden sind. Auch der Einfluss der unteren Randbedingung ist zu erkennen. Die Skalierung wurde so gewählt, dass die rot-grau Grenze bei der Plastifizierungsspannung des Kollagens liegt. 68
- Abbildung 58:** Die Spannungsverteilung innerhalb der Mineralplättchen bis zur maximalen Eindringtiefe von 22nm. Die Skalierung wurde dementsprechend angepasst und geht nun linear skaliert von $1e8$ Pa (blau) bis $2.6e9$ Pa (rot). Die nicht mehr unerheblichen Elementverzerrungen, vor allem der Kollagenbereiche in der Nähe der Mineralplättchenkanten, sind gut erkennbar. 69
- Abbildung 59:** Die Spannungsverteilung innerhalb des Kollagens bis zur maximalen Eindringtiefe von 22nm. Die Skalierung wurde dementsprechend angepasst und geht nun linear skaliert von 0 (blau) bis $4.5e7$ (rot). 70
- Abbildung 60:** Aus dem Postprocessingbild lässt sich für die maximale Eindringtiefe von $u = 22$ nm und einem Indenterspitzenradius von $r = 100$ nm ein für die projizierende Fläche maßgeblicher Maß $a = 50.7$ nm ermitteln. Laut Sehnensformel würde sich, also unter Vernachlässigung des Sink-in Effekts, ein Maß von $a' = 62.6$ nm ergeben. 72
- Abbildung 61:** Die nichtlinearen Zusammenhänge Eindringtiefe b (bzw. u) zu projektionsflächenbestimmendem Maß a' (kein Sink-in Effekt berücksichtigt, mittels Kreis-Sehnensformel, magenta) bzw. a (mit Berücksichtigung des Sink-in Effektes, mittels Skalierungsfaktors, blau). Bei der Skalierung wird davon

- ausgegangen, dass der Sink-in Effekt mit steigender Eindringtiefe kontinuierlich zunimmt. 73
- Abbildung 62:** Von ABAQUS ausgegebene "Load-Displacement-Curve" für den Referenzpunkt (RP) des Indenters (tip, magenta) sowie für den Materialknoten 10010101 genau unterhalb des Indenters (bulk, königsblau). Eine genauere Erläuterung zu den Kurvenverläufen kann **Kapitel 5.1.6** entnommen werden. 74
- Abbildung 63:** Aus Versuchen ermittelte "Load-Displacement-Curve" für kortikalen und trabekulären Knochen. Gut zu erkennen sind die typischen Größenordnungen bei solchen Indentationsversuchen [16]. 74
- Abbildung 64:** Die von ABAQUS ausgegebene "Load-Displacement-Curve" skaliert für die idealisiert gedachte Berkovich-Pyramidenspitze für den Referenzpunkt (RP) des Indenters (tip, magenta) sowie für den Materialknoten 10010101 genau unterhalb des Indenters (bulk, königsblau). Eine genauere Erläuterung zu den Kurvenverläufen kann **Kapitel 5.1.6** entnommen werden. 75
- Abbildung 65:** "Load-Displacement-Curve" für den Referenzpunkt (RP) des Indenters (tip, magenta) sowie für den Materialknoten 10010101 genau unterhalb des Indenters (bulk, königsblau). Auffällig dabei ist, dass sich der Indenter bereits vor der maximalen, elastischen "Rückverformung" von der Materialoberfläche löst (bei etwa 20nm). Unerwarteterweise kommt es, ebenfalls wegen der rückformungsverhindernden plastischen Verformung der Mineralplättchen, auch zu einer "kraftlosen" Rückverformung (von etwa 18nm bis 13nm) des Bulk-Materials. 77
- Abbildung 66:** Der letzte "Kontaktpunkt" des Indenters mit dem Mineralplättchen liegt aufgrund der angenommenen Materialgesetze sowie der auftretenden Verformungen und Spannungen nicht direkt unter dem Indenter. Dieser Umstand führt zu den in **Abbildung 62** gezeigten Kurvenverläufen. 77
- Abbildung 67:** Entlastungskurvenabschnitt mit der für die Oliver/Pharr-Auswertung erforderlichen Tangente samt zugehöriger Geradengleichung. 78
- Abbildung 68:** Auch mit ABAQUS-explicit können, obwohl mit dem dort verfügbaren Remeshing-Tools die Elementverzerrungen etwas minimiert werden können, keine größeren Eindringtiefen erzielt werden. Die transparent-blauen Dreiecke in der Detailansicht stellen "real (physikalisch) unmögliche Überlagerungen" von Kollagenelementen mit Mineralplättchenelementen dar. 80

Abbildung 69: Bereits bei sehr kleinen Eindringtiefen kommt es zu sehr starken Spannungen innerhalb der Mineralplättchen. Auch erkennt man bereits die stärker werdenden Elementverzerrungen der Kollagenelemente welche zwischen den "starren" Mineralplättchen und dem Indenter "eingequetscht" werden. 82

Abbildung 70: "Abbruchergebnis" einer Simulation für eine Indentation parallel zu den Mineralplättchen. Die Elementverzerrungen sind dabei noch eklatanter als bei der Indentation rechtwinkelig zu den Mineralplättchen (**Kapitel 5.1**). Die transparent-blauen Flächen stellen eine "real unmögliche Überlagerung" von Kollagenelementen untereinander bzw. mit Mineralplättchenelementen dar. 82

Literaturverzeichnis

Nummer	Literaturquelle
1	http://www.osteoporose.com
2	Roschger P. (2003) Knochen und Bindegewebe. Vorlesungsskriptum zu Lehrveranstaltung 430040 an der Montanuniversität Leoben
3	Fratzl P., Fratzl-Zelmann N. and Klaushofer K. (1993) Collagen packing and mineralization. Biophys. J. 64:260-266
4	Fratzl P., Klaushofer K. (2001) Röntgenkleinwinkelstreuung in der Osteologie. Journal für Mineralstoffwechsel. 2/2001:12-19
5	Jäger I., Fratzl P. (2000) Mineralized Collagen Fibrils: A Mechanical Model with a Staggered Arrangement of Mineral Particles. Biophysical Journal Vol.79:1737-1746
6	Rappaz M., Bellet M., Deville M. (2003) Numerical Modeling in Materials Science and Engineering. Springer Verlag ISBN 3-540-42676-0
7	Bathe Klaus-Jürgen (1990) Finite-Elemente-Methoden. Springer Verlag ISBN 3-540-15602-X
8	ABAQUS Version 6.5 Manuals
9	Oliver W.C., Pharr G.M. (1992) An improved technique for determining hardness and elastic modulus using load and displacement sensing indentation experiments. Journal for Material Research Vol.7 No.6:1564-1583

- 10 Marx V., Balke H. (1997) A critical investigation of the unloading behavior of sharp indentation. *Acta mater* Vol.45 No.9:3791-3800
- 11 <http://webmineral.com>
- 12 <http://de.wikipedia.org>
- 13 <http://labs.ansci.uiuc.edu>
- 14 www.mad-cow.org
- 15 Hengsberger S., Kulik A., Zysset Ph. (2002) Nanoindentation discriminates the elastic properties of individual human bone lamellae under dry and physiological conditions. *Elsevier Science Bone* Vol.30 No.1:178-184
- 16 Zysset Ph., Guo E., Hoffler E., Moore K., Goldstein S. (1999) Elastic modulus and hardness of cortical and trabecular bone lamellae measured by nanoindentation in the human femur. *Journal of Biomechanics* 32:1005-1012
- 17 Gao H., Ji B., Jäger I., Arzt E., Fratzl P. (2003) Material become insensitive to flaws at nanoscale: Lesson from nature. *PNAS* Vol.100 No.10:5597-5600
- 18 Hein H.J., Weisser L., Engler T., Bernstein A., Brandt J. (1997) Zur Darstellung der Mikrostruktur des Knochens mit dem Raster-Sondenmikroskop. *Z Med Phys* 7
- 19 Currey J. (1984) *The Mechanical Adaptation of Bones*. Princeton Univ. Press, Princeton N.Y.
- 20 Landis W.J. (1995) The strenght of a calcified tissue depends in part on the molecular structure and organization of its constituent mineral crystals in their matrix. *Bone* 16:533-544
- 21 Peter Frazl (2002) Von Knochen, Holz und Zähnen. *Physik Journal* 1 Nr. 5: 49-55
- 22 Buchmayr B. (2003) *Werkstoffmodelle in der Umformtechnik*. Vorlesungsskriptum zu Lehrveranstaltung 560030 an der Montanuniversität Leoben
- 23 Antretter T. (2003) *Numerische Lösung von Ingenieurproblemen mit dem Finite Element Paket ABAQUS*. Vorlesungsskriptum zu Lehrveranstaltung 400082 an der Montanuniversität Leoben

Anhang

a) C++ Quellcode

Hier ist einer der letzten Quellcodevarianten angeführt, mit der die ins ABAQUS eingelesene Datei "KnotenElemente.i" (**Anhang b**) erstellt wird, welche die Geometrie- bzw. die Netzdaten enthält. Kommentarzeilenmarke sind hierbei mit (//) gekennzeichnet (in der Programmiersprache C ist auch oft (/* */) üblich) und grün dargestellt. Blau gekennzeichnet sind die C++ internen Variablen. Jede Befehls- allerdings nicht jede Programmzeile (auch wenn hier in mehreren Zeilen abgedruckt) endet mit einem (;).

Das Programm besteht aus einem Hauptteil (MAIN) und ausgegliederten Subprozeduren (SUBROUTINE), die sich nach der Definition von C++ Programmen vor dem Hauptteil befinden müssen. Das Programm erstellt ein 2D-Netz mit $d = 2.5\text{nm}$, $l = 60\text{nm}$ und $a = b = 2.5\text{nm}$, nach der Skizze in **Abbildung 50**.

```
// =====
// Verfasser: Florian Markus MAYER,                2003/2006   *
// Diplomarbeit : "Nanointendierung in eine 'brick and mortar'-Struktur"   *
// C++Programm zur Erstellung des FEM-Netzes fuer "staggard2D"           *
// =====
//  H A U P T P R O G R A M M
// =====
// C-Header-Dateien (Funktion,Ein-Ausgabe,...) laden
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <process.h>
// Definition der verwendeten Ein- Ausgabe-Dateinamen
FILE *Ausgabe;
// Globale Variablendefinitionen
// double = +/- 1.79e+308
double mpl, mpd, cl, cd, lec, dec, a, rTip, xTip, yTip;
double demp, lemp, xs, ys, xcs, ycs, xsh;
double ych, nr_durch_ampl, h1, h2, ychh, xss;
double hpx1, hpx2, hpy1, hpy2, hp, yh;
double Pi = 3.1415926535;
double xc[9999], yc[9999];
double x[520000], y[520000];
// int = +/- 2 147 483 648
int nr, ampl, ampd, aec, ic, zw, b, b10er, b1er,c, c10er, c1er;
int rei, aempl, aempd, aemplh, fknh, gx, gy, mod, nr000;
int hMPaA, hMPaE, gMPaE, gerei, bc, BC_Nr, i, OoU, LoR;
int kn, knh, akn, fkn, mo, ia, ib, id, AMPid, ACid, enh, en, fen;
int mod100,ia10er,ia1er,ib10er,ib1er, kn_reihe, aknh;
int nr1er, nr10er, nrL, nrL1er, nrL10er;
```

```

int nrOLg, nrOLg1er, nrOLg10er, nrOLu, nrOLu1er, nrOLu10er, nrOL, nrOL1er, nrOL10er;
int nrORg, nrORg1er, nrORg10er, nrORu, nrORu1er, nrORu10er, nrOR, nrOR1er, nrOR10er;
int nrULg, nrULg1er, nrULg10er, nrULu, nrULu1er, nrULu10er, nrUL, nrUL1er, nrUL10er;
int f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, ZoMP, keinZ, ie, ig;
int MP_korrOR, MP_korrUL, MP_korrUR, MP_korr, y_korr;
int Schleife=1;
// sinnlos, aber mit 'void' laufen die Funktionen nicht!!
// arrays (Felder)
int n[5];
// float = +/- 3.40e+38
// =====
// *****
// SUBROUTINE Moeglichkeiten Knotenausgabe für BC_Nr=13 bis 23 (BC's)
// *****
int Moeglichkeiten(int i)
{
    // Moeglichkeiten Beginn
    if (xc[akn]<0)
    {
        if (xc[akn]>1e-10)
            {xc[akn]=0;}
    }
// CLO
if (BC_Nr==5)
{
    if (x[akn]>xs)
    {
        if (x[akn]<xcs)
        {
            if (yh==(ys-mpd))
                {fprintf(Ausgabe, "%10d,\n",kn);}
        }
    }
}
// CLU
if (BC_Nr==6)
{
    if (x[akn]>(xs-cd/2+1e-10)) // rundungsfehlerausbessernd
    {
        if (x[akn]<(xcs-cd/2-1e-10)) // rundungsfehlerausbessernd
        {
            if (yh==(ys-mpd-cd))
                {fprintf(Ausgabe, "%10d,\n",kn);}
        }
    }
}
// CLR
if (BC_Nr==7)
{
    for (ie=2; ie<=aec; ie++) // for (ie=1; ie<=(aec+1); ie++)
    {
        if (akn==(100*ie+aempl/2+1)) //((1xx,) 2xx, 3xx,..., 99xx

```

```

        {fprintf(Ausgabe, "%10d,\n",kn);}
    }
}
// CLL
if (BC_Nr==8)
{
    for (ie=2; ie<=aec; ie++)                // for (ie=1; ie<=(aec+1); ie++)
    {
        if (akn==(1+100*ie))                // 101, 201, 301,..., 9901
        {fprintf(Ausgabe, "%10d,\n",kn);}
    }
}
// CRO
if (BC_Nr==9)
{
    if (x[akn]<(xcs+mpl/2))
    {
        if (x[akn]>(xcs))
        {
            if (yh==(ys-mpd))
            {fprintf(Ausgabe, "%10d,\n",kn);}
        }
    }
}
// CRU
if (BC_Nr==10)
{
    if (x[akn]<(xcs+mpl/2+cd/2-1e-10))        // rundungsfehlerausbessernd
    {
        if (x[akn]>(xcs+cd/2+1e-10))        // rundungsfehlerausbessernd
        {
            if (yh==(ys-mpd-cd))
            {fprintf(Ausgabe, "%10d,\n",kn);}
        }
    }
}
// CRR
if (BC_Nr==11)
{
    for (ie=2; ie<=aec; ie++)
    {
        if (akn==(1+aempl/2+100*ie))        // 1xx, 2xx, 3xx,..., 99xx
        {fprintf(Ausgabe, "%10d,\n",kn);}
    }
}
// CRL
if (BC_Nr==12)
{
    for (ie=2; ie<=aec; ie++)
    {
        if (akn==(1+100*ie))                // 101, 201, 301,..., 9901

```

```

        {fprintf(Ausgabe, "%10d,\n",kn);}
    }
}
// CZO (L&R)
ychh=(yces-cd);
if (ych<yces)
{
    if (ych>ychh) // if (ych>=ychh)
    {
// CZOL
        if (BC_Nr==13)
        {
            for (ie=1; ie<=99; ie++)
            {
                if (akn==1+100*ie) // 101, 201, 301,..., 9901
                {fprintf(Ausgabe, "%10d,\n",kn);}
            }
        }
// CZOR
        if (BC_Nr==15)
        {
            for (ie=1; ie<=99; ie++)
            {
                if (akn==(ie*101+(ie-1))) // 101, 203, 305, 407,...9999
                {fprintf(Ausgabe, "%10d,\n",kn);}
            }
        }
// CZOM
        if (ych>ychh)
        {
            if (BC_Nr==14)
            {
                for (ie=1; ie<=(aec+1); ie++)
                {
                    if (akn==(101*ie)) // 101, 202, 303,...9999
                    {fprintf(Ausgabe, "%10d,\n",kn);}
                }
            }
        }
// CZU
        if (zw!=2)
        {ychh=(yces-cd-mpd);}
        else
        {ychh=(yces-cd-mpd-cd-mpd);}
// CZUL
        if (BC_Nr==16)
        {
            if (ych>(yces-cd-mpd-cd))
            {
                if (ych<ychh) // if (ych<=ychh)

```

```
{
    for (ie=1; ie<=99; ie++)
    {
        if (akn==1+100*ie) // 101, 201, 301, ..., 9901
            {fprintf(Ausgabe, "%10d,\n",kn);}
    }
}
}
}
// CZUM
if (BC_Nr==17)
{
    for (ie=2; ie<=aec; ie++)
    {
        if (akn==((aec+aempd)+ie)*100+((aec+2)-ie)) // y0x, ..., z01
            {fprintf(Ausgabe, "%10d,\n",kn);}
    }
}
// CZUR
if (BC_Nr==18)
{
    for (ie=2; ie<=aec; ie++)
    {
        if (akn==((aec+aempd)+ie)*100+((2*aec+1)-(ie-1)*2)) // 101, 203, 305, 407, ..., 9999
            {fprintf(Ausgabe, "%10d,\n",kn);}
    }
}
// CZOQ
if (BC_Nr==19)
{
    for (ie=1; ie<=(2*aec+1); ie++)
    {
        if (akn==(aec+1)*100+ie) // x01, x02, x03, ..., x99
            {fprintf(Ausgabe, "%10d,\n",kn);}
    }
}
// CZUQ
if (BC_Nr==20)
{
    for (ie=1; ie<=(2*aec+1); ie++)
    {
        if (akn==((aec+aempd+1)*100+ie)) // y01, y02, y03, ..., y99
            {fprintf(Ausgabe, "%10d,\n",kn);}
    }
}
// CZL
if (BC_Nr==21)
{
    if (xc[akn]==(xcs+mpl/2))
    {
        if (ych<(ycc-cd))
        {
```

```

        if (ych > (yces - cd - mpd))
            {fprintf(Ausgabe, "%10d,\n", kn);}
    }
}
}
// CZM
if (zw != 2)
    {ychh = (yces - cd);}
else
    {ychh = -((mpd + cd) * (nr / ampl));}
if (BC_Nr == 22)
    {
        if (ych < (ychh + 1e-10)) // rundungsfehlerbedingt
            {
                if (ych >= (ychh - mpd))
                    {
                        if (zw != 2)
                            {
                                if (xc[akn] == xcs + mpl/2 + cl/2)
                                    {fprintf(Ausgabe, "%10d,\n", kn);}
                                }
                            else
                                {
                                    if (xc[akn] == 0)
                                        {fprintf(Ausgabe, "%10d,\n", kn);}
                                    }
                                }
                            }
                    }
            }
}
// CZR
if (BC_Nr == 23)
    {
        if (zw != 2)
            {
                if (xc[akn] == (xcs + mpl/2 + cl/2 + cl/2))
                    {
                        if (ych < (yces - cd))
                            {
                                if (ych > (yces - cd - mpd))
                                    {fprintf(Ausgabe, "%10d,\n", kn);}
                                }
                            }
                    }
            }
        else
            {
                if (xc[akn] == (cl/2))
                    {
                        if (ych < (yces - cd))
                            {
                                if (ych > (yces - cd - mpd))
                                    {fprintf(Ausgabe, "%10d,\n", kn);}
                                }
                            }
                    }
            }
    }
}

```

```

    }
    }
}
return Schleife;
} // Moeglichkeiten Ende
// *****
// SUBROUTINE Knotengenerierung SchleifeK (Knotenausgabe MP&C)
// *****
int SchleifeK(int i)
{ // SchleifeK Beginn
    fkn=fknh+nr*10000;
    xsh=xs;
    for (id=0; id<=gy; id++)
    {
        akn=101+id*100;
        kn=fkn+akn;
        y[akn]=(ys+hpy1)+(id*hpy2);
        yh=y[akn];
// *****
        for (ic=0; ic<=gx; ic++)
        {
            x[akn]=(xsh+hp1)+ic*hp2;
            if (hMPaA==1)
                {xss=xs;}
            else
                {xss=xs+mpl/2;}
// normale Knotenausgabe
            if (bc==0)
                {fprintf(Ausgabe, "%10d,\t%2.4e,\t%2.4e\n",kn,x[akn],yh);} // allgemeine C-Kn
// BC-NodeSet AMP des MP
            if (bc==1)
            {
                if (x[akn]==0) // AMP y-Symmetrie bei x=0
                    {fprintf(Ausgabe, "%10d,\n",kn);}
            }
// NodesSets-Ausgabe
            if (bc==2)
            {
                if (BC_Nr==1) // OBEN (MPO)
                {
                    if (yh==ys)
                    {
                        if (LoR==1) // LINKS
                        {
                            if (x[akn]<xss-1e-10) // RUNDUNGSBEDINGT
                            {
                                if (x[akn]>xs+1e-10)
                                    {fprintf(Ausgabe, "%10d,\n",kn);}
                            }
                        }
                    }
                }
                if (LoR==2) // RECHTS
                {

```

```

        if (x[akn]>xss+1e-10)
        {
            if (x[akn]<(xss+mpl/2-1e-10))
            {fprintf(Ausgabe, "%10d,\n",kn);}
        }
    }
}
}
}
// ***** UNTEN *****
if (BC_Nr==2)                                // UNTEN (MPU)
{
    if (yh==(ys-mpd))
    {
        if (LoR==1)
        {
            if (x[akn]<xss-1e-10)
            {
                if (x[akn]>xs+1e-10)
                {fprintf(Ausgabe, "%10d,\n",kn);}
            }
        }
        if (LoR==2)
        {
            if (x[akn]>xss+1e-10)
            {
                if (x[akn]<(xss+mpl/2-1e-10))
                {fprintf(Ausgabe, "%10d,\n",kn);}
            }
        }
    }
}
//
if (BC_Nr==3)                                // MPR rechte Grenze (z.B. aemplh*lemp)
{
    if (x[akn]==(xsh+(gx*hp2)))
    {
        if (yh<ys-1e-10)
        {
            if (yh>(ys-mpd+1e-10))
            {fprintf(Ausgabe, "%10d,\n",kn);}
        }
    }
}
if (BC_Nr==4)                                // MPL linke Grenze = xs
{
    if (x[akn]==xsh)
    {
        if (yh<ys-1e-10)
        {
            if (yh>(ys-mpd+1e-10))
            {fprintf(Ausgabe, "%10d,\n",kn);}
        }
    }
}

```

```

    }
  }
}
// Randbedingungen-Ausgabe des Collagens (CLO,CRO,CRU.CZOR..)
if (bc==3) // BC_Nr=5 ... 23
  {Schleife=Moeglichkeiten(Schleife);}
kn=kn+1; // +1
akn=akn+1; // +0001
}
// *****
xsh=xsh+hp; // xsh bei MP gleich, beim Collagen wird "Schräge" berücksichtigt
}
return (Schleife);
} //SchleifeK Ende
// =====
// *****
// SUBROUTINE Elementgenerierung SchleifeE
// *****
int SchleifeE(int i)
{ // SchleifeE Beginn
  fen=enh+nr*10000;
  fkn=fknh+nr*10000;
  kn_reihe=0;
  for (ib=1; ib<=gy; ib++)
  {
    akn=ib*100+1; // akn=0101....
    kn=fkn+akn;
    en=fen+akn;
    en=enh+nr*10000+0101+0100;
    kn_reihe=kn_reihe+1;
    for (ia=1; ia<=gx; ia++)
    {
      n[1]=kn;
      n[2]=kn+100;
      n[3]=kn+100+1;
      n[4]=kn+1;
      if (ib==1)
      {
        if (ia==1)
        {
          if (b==1)
          {
            fprintf(Ausgabe, "*ELSET,ELSET=MP%d%d%dE,Instance=MP%d%d%d\n"
              ,nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
          }
        }
        if (b==2)
        {
          fprintf(Ausgabe, "*ELSET,ELSET=CZ%d%d%dE,Instance=CZ%d%d%d\n"
            ,nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
        }
      }
    }
  }
}
}

```

```

if (zw!=2)
{
    if (b!=4) // NORMALE Ausgabe mit zugehörigen Knoten!!
    {fprintf(Ausgabe, "%10d,%12d,%12d,%12d,%12d\n",en,n[1],n[2],n[3],n[4]);}
    else // für ElSet'AlleE'
    {fprintf(Ausgabe, "%10d,\n",en);}
}
if (zw==2)
{
    if (ZoMP==1) // Zwickel (2) oder MP (1)
    {
        if (b!=4) // NORMALE Ausgabe!!
        {fprintf(Ausgabe, "%10d,%12d,%12d,%12d,%12d\n",en,n[1],n[2],n[3],n[4]);}
    }
    else
    {
        if (ia>gx/2)
        {
            if (b!=4) // NORMALE Ausgabe!!
            {fprintf(Ausgabe, "%10d,%12d,%12d,%12d,%12d\n",en,n[1],n[2],n[3],n[4]);}
            else // nur El-Nummern-Ausgabe (b=4)
            {fprintf(Ausgabe, "%10d,\n",en);}
        }
    }
}
kn=kn+1; // kn=kn+0001
en=en+1; // en=en+0001
} // for ia ENDE
// *****
} // for ib ENDE
return Schleife;
} // SchleifeE Ende
// =====
// *****
// SUBROUTINE Knotenausgabe KAus = Knotenausgabe für Spitzenknoten
// *****
int KAus(int i)
{ // KAus Beginn
    if (xc[akn]>-1e-10)
    {
        if (xc[akn]<1e-10) // Ausgleich der Rundungsfehler
        {xc[akn]=0;}
    }
    if (BC_Nr>=13) // Randbedingungsausgabe
    {
        if (zw==2) // halbe Symmetrie-Zwickel
        {
            if (xc[akn]>=0)
            {Schleife=Moeglichkeiten(Schleife);}
        }
        else // NICHT halber Zwickel
        {Schleife=Moeglichkeiten(Schleife);}
    }
}

```

```

}
else // normale Ausgabe
{
  if (OoU==2) // unterer Spitz (je nach dem aus welcher Schleife aufgerufen)
  {
    if (zw==1)
    {fprintf(Ausgabe, "%10d,\t%2.4e,\t%2.4e\n",kn,xc[akn],ych);}
    }
    if (zw==2) // halbe Symmetrie-Zwickel
    {
      if (xc[akn]>=0)
      {
        fprintf(Ausgabe, "%10d,\t%2.4e,\t%2.4e\n",kn,xc[akn],ych);
      }
    }
    if (zw==3) // normale, vollständige Zwickel
    {fprintf(Ausgabe, "%10d,\t%2.4e,\t%2.4e\n",kn,xc[akn],ych);}
    if (OoU==1) // oberer Spitz (je nach dem aus welcher Schleife aufgerufen)
    {
      if (zw==4) // Zwickel der OBERTSEN Reihe
      {fprintf(Ausgabe, "%10d,\t%2.4e,\t%2.4e\n",kn,xc[akn],ych);}
    }
  }
}
return Schleife;
} // KAus Ende
// =====
// *****
// SUBROUTINE Knotengenerierung SchleifeKZS = KnotenZwickelSpitz
// *****
int SchleifeKZS(int i)
{ // SchleifeKZS Beginn
  fknh=30000000; // KnotenCollagenZwickel 3wwwyyxx
  fkn=fknh+nr*10000;
  // #####
  // Spitz
  // #####
  // *****
  for (ia=0; ia<=(aec-1); ia++)
  {
    if (OoU==1) // OBEN
    {
      akn=101+100*(ia);
      kn=fkn+akn;
      if (gerei==0) // UNGERADE Reihe
      {MP_korr=-ampl;}
      else // GERADE Reihe
      {MP_korr=-(ampl-1);}
      y_korr=aempd+1;
      yc[akn]=(yca)-ia*dec;
      ig=ia;
    }
    if (OoU==2) // UNTEN

```

```

{
    akn=101+100*(aec+aempd+1+ia);
    kn=fkn+akn;
    if (gerei==0) // UNGERADE Reihe
    {MP_korr=ampl;}
    else // GERADE Reihe
    {MP_korr=ampl+1;}
    y_korr=1;
    yc[akn]=(ycc-cd-mpd)-(ia+1)*dec;
    ig=(aec-1)-ia;
}
ych=yc[akn];
if (ig==0) // keine Schleife mit von -0 bis +0 in 1er-Schritten möglich!!
{
    xc[akn]=(xcs+mpl/2+cl/2);
    Schleife=KAus(Schleife);
}
else
{
    for (ib=-ig; ib<=ig; ib++)
    {
        xc[akn]=(xcs+mpl/2+cl/2)+ib*lec;
        Schleife=KAus(Schleife);
        kn=kn+1;
        akn=akn+1; //akn=akn+0001
    }
}
}
return Schleife; // SchleifeKZS Ende
// =====
// *****
// SUBROUTINE Knotengenerierung SchleifeKZR = KnotenZwickelRechteck
// *****
int SchleifeKZR(int i)
{ // SchleifeKZR Beginn
    for (ia=0; ia<=aempd; ia++) // y-Schleife
    {
        akn=101+100*(aec+ia); // akn=0101+0100*(aec+ia)
        kn=fkn+akn;
        yc[akn]=(ycc-cl)-ia*demp; // yc[akn]=ys-ia*demp
        ych=yc[akn];
// *****
        for (ib=-aec; ib<=aec; ib++) // x-Schleife
        {
            if (zw!=2) // NICHT halber Zwickel vorne
            {
                xc[akn]=(xcs+mpl/2+cl/2)+ib*lec;
                if (BC_Nr>=13) // Randbedingungsabgabe (bei <13 nie durchlaufen)
                {Schleife=Moeglichkeiten(Schleife);}
                else // normale Ausgabe aller Knoten
                {fprintf(Ausgabe, "%10d,\t%2.4e,\t%2.4e\n",kn,xc[akn],ych);}
            }
        }
    }
}

```

```

    }
    else // halbe Zwickel bei y-Symmetrie
    {
        xc[akn]=(0+ib*lec);
        if (xc[akn]<0)
        {
            if (xc[akn]>1e-10) // rundungsbedingte Fehler korrigieren
            {xc[akn]=0;}
        }
        if (xc[akn]>=0) // NUR Knoten mit POSITIVEM x-Wert
        {
            if (BC_Nr>=13) // Randbedingungsabgabe
            {Schleife=Moeglichkeiten(Schleife);}
            else // normale Ausgabe aller Knoten
            {fprintf(Ausgabe, "%10d,\t%2.4e,\t%2.4e\n",kn,xc[akn],ych);}
        }
    }
    kn=kn+1;
    akn=akn+1; // akn=akn+0001
}
}
return Schleife; // SchleifeKZR Ende
}
// =====
// *****
// SUBROUTINE Elementgenerierung SchleifeEZSD = ElementeZwickelSpitzDreiecke
// *****
int SchleifeEZSD(int i)
{ // SchleifeEZSD Beginn
    for (ia=1; ia<=aec; ia++)
    { // LINKE Seite der Dreieckelemente
        akn=(ia+f1)*100+1; // OBEN : ia*100+1 UNTEN : (ia+aec+aempd)*100+1
        kn=fkn+akn;
        en=fen+akn;
        if (zw!=2) // NICHT halbe Zwickel vorne (diese hätten keine Dreiecke LINKS)
        {
            n[1]=kn;
            n[2]=kn+100;
            n[3]=kn+f2+1; // OBEN : kn+100+1 UNTEN : kn+1
        }
        // Ausgabe
        if (b!=3) // normale Ausgabe. also NICHT für ElSet 'AlleE'
        {fprintf(Ausgabe, "%10d,%12d,%12d,%12d\n",en,n[1],n[2],n[3]);}
        else // für ElSet'AlleE'
        {fprintf(Ausgabe, "%10d,\n",en);}
    }
    // RECHTE Seite der Dreieckelemente
    en=en+1+(f3+ia+f4)*f5; // OBEN : en+1+(ia-1)*2 UNTEN : en+1+(aec-ia)*2
    // en+1+(aec-ia)*2 = en+1+(-aec+ia)*(-2)
    n[1]=(kn+(f3+ia+f4)*f5)+f6; // OBEN : (kn+(ia-1)*2) UNTEN : (kn+(aec-ia)*2)+1
    n[2]=(kn+(f3+ia+f4)*f5)+f7; // OBEN : (kn+(ia-1)*2)+101 UNTEN : (kn+(aec-ia)*2)+100
    n[3]=(kn+(f3+ia+f4)*f5)+f8; // OBEN : (kn+(ia-1)*2) UNTEN : (kn+(aec-ia)*2)+1
    // Ausgabe
    if (b!=3) // normale Ausgabe. also NICHT für ElSet 'AlleE'

```

```

    {fprintf(Ausgabe, "%10d,%12d,%12d,%12d\n",en,n[1],n[2],n[3]);}
    else // für ElSet'AlleE'
    {fprintf(Ausgabe, "%10d,\n",en);}
    }
return Schleife;
} // SchleifeEZSD Ende
// =====
// *****
// SUBROUTINE Elementgenerierung SchleifeEZSV = ElementeZwickelSpitzVierecke
// *****
int SchleifeEZSV(int i)
{ // SchleifeEZSV Beginn
for (ib=2; ib<=aec; ib++)
{
akn=(f1+ib+f10)*100+1; // OBEN : (ib*100+1) UNTEN : (ib-1+aec+aempd)*100+1
kn=fkn+akn;
en=fen+akn+1;
for (ia=1; ia<=((f3+ib+f4+f10)*f5); ia++) // OBEN : (ib-1)*2 UNTEN : (aec-ib+1)*2
{
n[1]=kn+f11; // OBEN : kn+0 UNTEN : kn+1+99
n[2]=kn+f11+f9; // OBEN : kn+0+101 UNTEN : kn+1+99
n[3]=kn+f11+f9+1; // OBEN : kn+0+101+1 UNTEN : kn+1+99+1
n[4]=kn+f11+1;
if (zw!=2)
{
if (b!=3)
{fprintf(Ausgabe, "%10d,%12d,%12d,%12d,%12d\n",en,n[1],n[2],n[3],n[4]);}
else
{fprintf(Ausgabe, "%10d,\n",en);} // für AlleE
}
}
else
{
if (b!=3)
{
if (ia>(((f3+ib+f4+f10)*f5)/2))
{fprintf(Ausgabe, "%10d,%12d,%12d,%12d,%12d\n",en,n[1],n[2],n[3],n[4]);}
}
else
{
if (ia>(((f3+ib+f4+f10)*f5)/2))
{fprintf(Ausgabe, "%10d,\n",en);}
}
}
}
kn=kn+1; // kn=kn+0001
en=en+1; // en=en+0001
}
}
return Schleife;
} // SchleifeEZSV Ende
// =====
// *****
// SUBROUTINE Knoten-&Elementgenerierung ZwickelKE

```

```

// *****
int ZwickelKE(int i)
{
    // ZwickelKE Beginn
    if (zw!=5)
    {
        if (BC_Nr<5)
        {
            fprintf(Ausgabe, "*** *****\n");
            fprintf(Ausgabe, "** **          Collagen          **\n");
            fprintf(Ausgabe, "** **          Zwickel          **\n");
            fprintf(Ausgabe, "** ** Knoten plus Elemente **\n");
            fprintf(Ausgabe, "*** *****\n");
            fprintf(Ausgabe, "*Instance, name=CZ%d%d%d,
                part=CollagenZwickel\n",nr/100,nr10er,nr1er);
            printf("*Instance, name=CZ%d%d%d, part=CollagenZwickel\n",nr/100,nr10er,nr1er);
            fprintf(Ausgabe, "*NODE\n");
        }
    }
    // #####
    // KNOTEN
    // #####
    // #####
    // Spitz OBEN
    // #####
    OoU=1;
    Schleife=SchleifeKZS(Schleife);
    // #####
    // Rechteck
    // #####
    Schleife=SchleifeKZR(Schleife);
    // #####
    // Spitz UNTEN
    // #####
    OoU=2;
    Schleife=SchleifeKZS(Schleife);
    // #####
    // ELEMENTE
    // #####
    if (BC_Nr<5)
    {
        if (bc<2)
        {
            enh=80000000;
            fen=enh+nr*10000;
            fkn=fknh+nr*10000;
            b=5;
            // 0 für normale Ausgabe, >0 für Collagenausgabe
            if (zw!=1)
            {
                // #####
                // Spitz OBEN Dreiecke (Zwickel Oben Dreiecke) (CPE3 = 3-node linear)
                // #####
                fprintf(Ausgabe, "*ELEMENT,TYPE=CPE3\n");
                printf("*ELEMENT,ELSET=ZwickelOd%dE\n",nr);
            }
        }
    }
}

```

```

    OoU=1;
    f1=f3=f6=f10=0;
    f2=100;
    f4=-1;
    f5=2;
    f7=f9=101;
    f8=102;
    f11=0;
    Schleife=SchleifeEZSD(Schleife);
// #####
// Spitz OBEN Vierecke (Zwickel Oben Vierecke) (CPE4R = 4-node bilinear)
// #####
    fprintf(Ausgabe, "*ELEMENT,TYPE=CPE4R\n");
    printf("*ELEMENT,ELSET=ZwickelOV%dE\n",nr);
    Schleife=SchleifeEZSV(Schleife);
}
// #####
// Rechteck (Zwickel) (CPE4R = 4-node bilinear)
// #####
    fprintf(Ausgabe, "*ELEMENT,TYPE=CPE4R\n");
    printf("*ELEMENT,ELSET=ZwickelR%dE\n",nr);
    gx=aec*2;
    gy=aempd;
    enh=80000000+aec*100;           // ElementCollagenZwickel 8wwwyyxx
    fknh=30000000+aec*100;         // KnotenCollagenZwickel 3wwwyyxx
    ZoMP=2; // Zwickel oder MP => bei MP Ausgabe von allen Elementen bei zw=2
    Schleife=SchleifeE(Schleife);
    if (zw!=4) // != entspricht UNGLEICH
    {
// #####
// Spitz UNTEN Dreiecke (Zwickel Unten Dreiecke) (CPE3 = 3-node linear)
// #####
    fprintf(Ausgabe, "*ELEMENT,TYPE=CPE3\n");
    printf("*ELEMENT,ELSET=ZwickelUd%dE\n",nr);
    fen=80000000+nr*10000;
    fkn=30000000+nr*10000;
    OoU=2;
    f1=(aec+aempd);
    f2=f4=0;
    f3=-aec,
    f5=-2;
    f6=1;
    f7=100;
    f8=2;
    f9=99;
    f10=-1;
    f11=1;
    Schleife=SchleifeEZSD(Schleife);
// #####
// Spitz UNTEN Vierecke (Zwickel Unten Vierecke) (CPE4R = 4-node bilinear)
// #####
    fprintf(Ausgabe, "*ELEMENT,TYPE=CPE4R\n");

```

```

        printf("*ELEMENT,ELSET=ZwickelUV%dE\n",nr);
        Schleife=SchleifeEZSV(Schleife);
    }
// ++++++
// ELEMENTSET AlleE
// ++++++
    fprintf(Ausgabe, "*Elset, elset=AlleE\n");
    printf("*Elset, elset=AlleE\n");
    if (bc<2)
    {
        enh=80000000; // ElementCollagenZwickel 8wwwyyxx
        fen=enh+nr*10000;
        fkn=fknh+nr*10000;
        if (zw!=1)
        {
// #####
// ELEMENTSET AlleE - Zwickel Oben Dreiecke (CPE3 = 3-node linear)
// #####
            OoU=1;
            f1=f3=f6=f10=0;
            f2=100;
            f4=-1;
            f5=2;
            f7=f9=101;
            f8=102;
            f11=0;
            b=3;
            Schleife=SchleifeEZSD(Schleife);
// #####
// ELEMENTSET AlleE - Zwickel Oben Vierecke (CPE4R = 4-node bilinear)
// #####
            Schleife=SchleifeEZSV(Schleife);
        }
// #####
// ELEMENTSET AlleE - Rechteck (Zwickel) (CPE4R = 4-node bilinear)
// #####
            gx=aec*2;
            gy=aempd;
            enh=80000000+aec*100; // ElementCollagenZwickel 8wwwyyxx
            fknh=30000000+aec*100; // KnotenCollagenZwickel 3wwwyy
            ZoMP=2;
// Zwickel oder MP => bei MP Ausgabe von allen Elementen bei zw=2
            b=4;
// 0 für normale Ausgabe, 1 für Ausgabe von ObenE
            Schleife=SchleifeE(Schleife);
            if (zw!=4) // != entspricht UNGLEICH
            {
// #####
// ELEMENTSET AlleE - Zwickel Unten Dreiecke (CPE3 = 3-node linear)
// #####
                fen=80000000+nr*10000;
                fkn=30000000+nr*10000;

```

```

    OoU=2;
    f1=(aec+aempd);
    f2=f4=0;
    f3=-aec,
    f5=-2;
    f6=1;
    f7=100;
    f8=2;
    f9=99;
    f10=-1;
    f11=1;
    b=3;
    Schleife=SchleifeEZSD(Schleife);
// #####
// ELEMENTSET AlleE - Zwickel Unten Vierecke (CPE4R = 4-node bilinear)
// #####
    Schleife=SchleifeEZSV(Schleife);
    }
    }
    fprintf(Ausgabe, "*SOLID SECTION, ELSET=AlleE,
                MATERIAL=CollagenZ\n",nr/100,nr10er,nr1er);
    fprintf(Ausgabe, "1.\n");
    fprintf(Ausgabe, "*END Instance\n");
    printf("*END Instance\n");
    }
    }
    }
    return Schleife;
} // ZwickelKE Ende
// =====
// *****
// SUBROUTINE Knoten-&Elementgenerierung KE
// *****
// Funktionsdeklarationen
int KE(int i)
{ // KE Beginn
    div_t div_result;
    fprintf(Ausgabe, "** #####\n");
    fprintf(Ausgabe, "** ## Bereich : Mineralplättchen %d%d%d ##\n",nr/100,nr10er,nr1er);
    fprintf(Ausgabe, "** #####\n");
// #####
// Plättchen
// #####
    fknh=00000000; // KnotenMineral 0wwwyyxx
    enh=50000000; // ElementMineral 5wwwyyxx (früher nur bis 9999999)
// #####
// KNOTEN Plättchen
// #####
    fprintf(Ausgabe, "** *****\n");
    fprintf(Ausgabe, "** ** Mineralplättchen **\n");
    fprintf(Ausgabe, "** ** Knoten plus Elemente **\n");
    fprintf(Ausgabe, "** *****\n");

```

```

fprintf(Ausgabe, "*Instance, name=MP%d%d%d,
                part=Mineralplättchen\n",nr/100,nr10er,nr1er);
printf("\n*Instance, name=MP%d%d%d, part=Mineralplaettchen\n",nr/100,nr10er,nr1er);
fprintf(Ausgabe, "*NODE\n");
hpx1=0;
hpx2=lemp;
hpy1=0;
hpy2=-demp;
hp=0;
Schleife=SchleifeK(Schleife);
// #####
// ELEMENTE Plättchen (CPE4R = 4-node bilinear)
// #####
fprintf(Ausgabe, "*ELEMENT,TYPE=CPE4R\n");
printf("*ELEMENT,TYPE=CPE4R\n");
ZoMP=1;
b=0;
Schleife=SchleifeE(Schleife);
// ++++++
// ELEMENTSET AlleE
// ++++++
fprintf(Ausgabe, "** Region:\n");
fprintf(Ausgabe, "*Elset, elset=AlleE\n");
printf("*Elset, elset=AlleE\n");
ic=aempl;
if (hMPaA==1)
{ic=aempl/2;}
if (hMPaE==1)
{ic=aempl/2;}
for (ia=1; ia<=aempd; ia++)
{
    ia10er=ia/10;
    div_result = div(ia,10);
    ia1er=div_result.rem;
    for (ib=1; ib<=ic; ib++)
    {
        ib10er=ib/10;
        div_result = div(ib,10);
        ib1er=div_result.rem;
        fprintf(Ausgabe, " 5%d%d%d%d%d%d%d\n",
                    nr/100,nr10er,nr1er,ia10er,ia1er,ib10er,ib1er);
    }
}
}
fprintf(Ausgabe, "** Section:\n");
fprintf(Ausgabe, "*SOLID SECTION, ELSET=AlleE,
                MATERIAL=Mineral\n",nr/100,nr10er,nr1er);
fprintf(Ausgabe, "1.\n");
fprintf(Ausgabe, "*END Instance\n");
printf("*END Instance\n");
// #####
// BC's für Plättchen (MPO, MPU,...)
// #####

```

```

// MPOLwww Mineralplättchen Oben Links
// MPORwww Mineralplättchen Oben Rechts
// MPULwww Mineralplättchen Unten Links
// MPURwww Mineralplättchen Unten Rechts
// MPRwww Mineralplättchen Rechts
// MPLwww Mineralplättchen Links
fprintf(Ausgabe, "*** *****\n");
fprintf(Ausgabe, "*** ** MineralPlättchen **\n");
fprintf(Ausgabe, "*** ** NSETs für BC **\n");
fprintf(Ausgabe, "*** *****\n");
if (hMPaA==1) // halbes MP am Anfang
{
    div_result = div(AMPid,10);
    mod=div_result.rem;
    fprintf(Ausgabe, "*NSET,NSET=AMP%d%d,Instance=MP%d%d%d\n",AMPid/10,
            mod,nr/100,nr10er,nr1er); // AMP ... Anfang MineralPlättchen
    printf("*NSET,NSET=AMP%d\n",AMPid);
    bc=1; // Variable welche if-Anweisung bei SchleifeK Anwendung findet
    Schleife = SchleifeK(Schleife); // BC-Bedingung Symetrie um y => KnotenSet
    AMPid=AMPid+1;
}
bc=2;
if (hMPaA!=1)
{
    LoR=1;
    fprintf(Ausgabe, "*NSET,NSET=MPOL%d%d%d,Instance=MP%d%d%d\n",
            nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
    printf("*NSET,NSET=MPOL%d\n",nr); // MPOL
    BC_Nr=1;
    Schleife=SchleifeK(Schleife);
    fprintf(Ausgabe, "*NSET,NSET=MPUL%d%d%d,Instance=MP%d%d%d\n",nr/100,
            nr10er,nr1er,nr/100,nr10er,nr1er);
    printf("*NSET,NSET=MPUL%d\n",nr); // MPUL
    BC_Nr=2;
    Schleife=SchleifeK(Schleife);
}
if (hMPaE!=1)
{
    LoR=2;
    fprintf(Ausgabe, "*NSET,NSET=MPOR%d%d%d,Instance=MP%d%d%d\n",
            nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
    printf("*NSET,NSET=MPOR%d\n",nr);
    BC_Nr=1;
    Schleife=SchleifeK(Schleife);
    fprintf(Ausgabe, "*NSET,NSET=MPUR%d%d%d,Instance=MP%d%d%d\n",
            nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
    printf("*NSET,NSET=MPUR%d\n",nr);
    BC_Nr=2;
    Schleife=SchleifeK(Schleife);
}
fprintf(Ausgabe, "*NSET,NSET=MPR%d%d%d,Instance=MP%d%d%d\n",
            nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);

```

```

printf("*NSET,NSET=MPR%d\n",nr);
BC_Nr=3;
Schleife=SchleifeK(Schleife);
fprintf(Ausgabe, "*NSET,NSET=MPL%d%d%d,Instance=MP%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
printf("*NSET,NSET=MPL%d\n",nr);
BC_Nr=4;
Schleife=SchleifeK(Schleife);
printf("*NSET,NSET=MPEcken\n");
if (hMPaA!=1)
{
    fprintf(Ausgabe, "*NSET,NSET=MPEckOL%d%d%d,Instance=MP%d%d%d\n",
            nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
    kn=00000000+nr*10000+1*100+1*1;
    fprintf(Ausgabe, "%10d,\n",kn);
    fprintf(Ausgabe, "*NSET,NSET=MPEckUL%d%d%d,Instance=MP%d%d%d\n",
            nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
    kn=00000000+nr*10000+(aempd+1)*100+1*1;
    fprintf(Ausgabe, "%10d,\n",kn);
}
fprintf(Ausgabe, "*NSET,NSET=MPEckOM%d%d%d,Instance=MP%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
if (hMPaA==1)
{kn=00000000+nr*10000+1*100+1*1;}
else
{kn=00000000+nr*10000+1*100+(aempl/2+1)*1;}
fprintf(Ausgabe, "%10d,\n",kn);
fprintf(Ausgabe, "*NSET,NSET=MPEckUM%d%d%d,Instance=MP%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
if (hMPaA==1)
{kn=00000000+nr*10000+(aempd+1)*100+1*1;}
else
{kn=00000000+nr*10000+(aempd+1)*100+(aempl/2+1)*1;}
fprintf(Ausgabe, "%10d,\n",kn);
if (hMPaE!=1)
{
    fprintf(Ausgabe, "*NSET,NSET=MPEckOR%d%d%d,Instance=MP%d%d%d\n",
            nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);

    if (hMPaA==1)
    {kn=00000000+nr*10000+1*100+(aempl/2+1)*1;}
    else
    {kn=00000000+nr*10000+1*100+(aempl+1)*1;}
    fprintf(Ausgabe, "%10d,\n",kn);
    fprintf(Ausgabe, "*NSET,NSET=MPEckUR%d%d%d,Instance=MP%d%d%d\n",
            nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);

    if (hMPaA==1)
    {kn=00000000+nr*10000+(aempd+1)*100+(aempl/2+1)*1;}
    else
    {kn=00000000+nr*10000+(aempd+1)*100+(aempl+1)*1;}
    fprintf(Ausgabe, "%10d,\n",kn);
}
}
bc=0;

```

```

if (rei<ampd)                // MP in letzter Reihe haben kein Collagen, nur Zwickel!!
{
  if (hMPaA!=1)              // ausser halbes MP am Anfang der Reihe (=1)
  {
    fprintf(Ausgabe, "*** *****\n");
    fprintf(Ausgabe, "*** **  Collagen Links  **\n");
    fprintf(Ausgabe, "*** **  Knoten plus Elemente **\n");
    fprintf(Ausgabe, "*** *****\n");
// #####
// Collagen LINKS
// #####
    fknh=10000000;           // KnotenCollagenLinks 1wwyyxx
    Links=60000000;         // ElementCollagenLinks 6wwyyxx
    gx=aempl/2;
    gy=aec;
// #####
// KNOTEN Collagen LINKS
// #####
    fprintf(Ausgabe, "*Instance, name=CL%d%d%d,
                    part=CollagenLinks\n",nr/100,nr10er,nr1er);
    printf("*Instance, name=CL%d%d%d, part=CollagenLinks\n",nr/100,nr10er,nr1er);
    fprintf(Ausgabe, "*NODE\n");
    hpx1=0;
    hpx2=lemp;
    hpy1=-mpd;
    hpy2=-dec;
    hp=-lec;
    xs=xcs-mpl/2;           // da in SchleifeK xs als Variable
    Schleife=SchleifeK(Schleife);
// #####
// ELEMENTE Collagen LINKS (CPE4R = 4-node bilinear)
// #####
    fprintf(Ausgabe, "*ELEMENT,TYPE=CPE4R\n");
    printf("*ELEMENT,TYPE=CPE4R\n");
    b=0;
    Schleife=SchleifeE(Schleife);
// ++++++
// ELEMENTSET AlleE
// ++++++
    fprintf(Ausgabe, "*Elset, elset=AlleE\n");
    printf("*Elset, elset=AlleE\n");
    for (ia=1; ia<=(aempd+1); ia++)
    {
      ia10er=ia/10;
      div_result = div(ia,10);
      ia1er=div_result.rem;
      for (ib=1; ib<=(aempl/2); ib++)
      {
        ib10er=ib/10;
        div_result = div(ib,10);
        ib1er=div_result.rem;
        fprintf(Ausgabe, " 6%d%d%d%d%d%d%d,\n",

```

```

        nr/100,nr10er,nr1er,ia10er,ia1er,ib10er,ib1er);
    }
}
fprintf(Ausgabe, "**SOLID SECTION, ELSET=AlleE, MATERIAL=CollagenLR\n",
        nr/100,nr10er,nr1er);
fprintf(Ausgabe, "1.\n");
fprintf(Ausgabe, "*END Instance\n");
printf("*END Instance\n");
fprintf(Ausgabe, "*** *****\n");
fprintf(Ausgabe, "*** ** Collagen Links **\n");
fprintf(Ausgabe, "*** ** NSETs für BC **\n");
fprintf(Ausgabe, "*** *****\n");
// CLOWww Collagen LINKS Oben
// CLUww Collagen LINKS Unten
// CLRww Collagen LINKS Rechts
// CLLww Collagen LINKS Links
bc=3; // verhindert Ausgabe in SchleifeK = Moeglichkeiten
fprintf(Ausgabe, "**NSET,NSET=CLO%d%d%d,Instance=CL%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
printf("**NSET,NSET=CLO%d\n",nr);
BC_Nr=5;
Schleife=SchleifeK(Schleife);
fprintf(Ausgabe, "**NSET,NSET=CLU%d%d%d,Instance=CL%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
printf("**NSET,NSET=CLU%d\n",nr);
BC_Nr=6;
Schleife=SchleifeK(Schleife);
fprintf(Ausgabe, "**NSET,NSET=CLR%d%d%d,Instance=CL%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
printf("**NSET,NSET=CRL%d\n",nr);
BC_Nr=7;
Schleife=SchleifeK(Schleife);
fprintf(Ausgabe, "**NSET,NSET=CLL%d%d%d,Instance=CL%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
printf("**NSET,NSET=CLL%d\n",nr);
BC_Nr=8;
Schleife=SchleifeK(Schleife);
fprintf(Ausgabe, "**NSET,NSET=CLEckOL%d%d%d,Instance=CL%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
kn=10000000+nr*10000+1*100+1*1;
fprintf(Ausgabe, "%10d,\n",kn);
fprintf(Ausgabe, "**NSET,NSET=CLEckOR%d%d%d,Instance=CL%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
kn=10000000+nr*10000+1*100+(aempl/2+1)*1;
fprintf(Ausgabe, "%10d,\n",kn);
fprintf(Ausgabe, "**NSET,NSET=CLEckUL%d%d%d,Instance=CL%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
kn=10000000+nr*10000+(aec+1)*100+1*1;
fprintf(Ausgabe, "%10d,\n",kn);
fprintf(Ausgabe, "**NSET,NSET=CLEckUR%d%d%d,Instance=CL%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
kn=10000000+nr*10000+(aec+1)*100+(aempl/2+1)*1;

```

```

    fprintf(Ausgabe, "%10d,\n",kn);
    bc=0;
}
if (hMPaE!=1) // ausser halbes MP am Ende der Reihe
{
    fprintf(Ausgabe, "*** *****\n");
    fprintf(Ausgabe, "*** ** Collagen Rechts **\n");
    fprintf(Ausgabe, "*** ** Knoten plus Elemente **\n");
    fprintf(Ausgabe, "*** *****\n");
// #####
// Collagen RECHTS
// #####
    fknh=20000000; // KnotenCollagenRechts 2wwwyyxx
    enh=70000000; // ElementCollagenRechts 7wwwyyxx
    gx=aempl/2;
    gy=aec;
// #####
// KNOTEN RECHTS Collagen
// #####
    fprintf(Ausgabe, "*Instance, name=CR%d%d%d, part=CollagenRechts\n",
            nr/100,nr10er,nr1er);
    printf("*Instance, name=CR%d%d%d, part=CollagenRechts\n",nr/100,nr10er,nr1er);
    fprintf(Ausgabe, "*NODE\n");
    printf("*NODE,NSET=CR%dKN\n",nr);
    hpx1=0;
    hpx2=lemp;
    hpy1=-mpd;
    hpy2=-dec;
    hp=lec;
    xs=xcs;
    Schleife=SchleifeK(Schleife);
// #####
// ELEMETE RECHTS Collagen (CPE4R = 4-node bilinear)
// #####
    fprintf(Ausgabe, "*ELEMENT,TYPE=CPE4R\n");
    printf("*ELEMENT,ELSET=CR%dE\n",nr);
    b=0;
    Schleife=SchleifeE(Schleife);
// ++++++
// ELEMENTSET AlleE
// ++++++
    fprintf(Ausgabe, "*Elset, elset=AlleE\n");
    printf("*Elset, elset=AlleE\n");
    for (ia=1; ia<=(aempd+1); ia++)
    {
        ia10er=ia/10;
        div_result = div(ia,10);
        ia1er=div_result.rem;
        for (ib=1; ib<=(aempl/2); ib++)
        {
            ib10er=ib/10;
            div_result = div(ib,10);

```

```

        ib1er=div_result.rem;
        fprintf(Ausgabe, " 7%d%d%d%d%d%d\n",
                nr/100,nr10er,nr1er,ia10er,ia1er,ib10er,ib1er);
    }
}
fprintf(Ausgabe, "*SOLID SECTION, ELSET=AlleE, MATERIAL=CollagenLR\n",
        nr/100,nr10er,nr1er);
fprintf(Ausgabe, "1.\n");
fprintf(Ausgabe, "*END Instance\n");
printf("*END Instance\n");
fprintf(Ausgabe, "***\n");
fprintf(Ausgabe, "*** ** Collagen Rechts **\n");
fprintf(Ausgabe, "*** ** NSETs für BC **\n");
fprintf(Ausgabe, "***\n");
// CROwww Collagen RECHTS Oben
// CRUwww Collagen RECHTS Unten
// CRRwww Collagen RECHTS Rechts
// CRLwww Collagen RECHTS Links
bc=3;
fprintf(Ausgabe, "*NSET,NSET=CRO%d%d%d,Instance=CR%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
printf("*NSET,NSET=CRO%d\n",nr);
BC_Nr=9;
Schleife=SchleifeK(Schleife);
fprintf(Ausgabe, "*NSET,NSET=CRU%d%d%d,Instance=CR%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
printf("*NSET,NSET=CRU%d\n",nr);
BC_Nr=10;
Schleife=SchleifeK(Schleife);
fprintf(Ausgabe, "*NSET,NSET=CRR%d%d%d,Instance=CR%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
printf("*NSET,NSET=CRR%d\n",nr);
BC_Nr=11;
Schleife=SchleifeK(Schleife);
fprintf(Ausgabe, "*NSET,NSET=CRL%d%d%d,Instance=CR%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
printf("*NSET,NSET=CRL%d\n",nr);
BC_Nr=12;
Schleife=SchleifeK(Schleife);
BC_Nr=1;
fprintf(Ausgabe, "*NSET,NSET=CREckOL%d%d%d,Instance=CR%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
kn=20000000+nr*10000+1*100+1*1;
fprintf(Ausgabe, "%10d\n",kn);
fprintf(Ausgabe, "*NSET,NSET=CREckOR%d%d%d,Instance=CR%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
kn=20000000+nr*10000+1*100+(aempl/2+1)*1;
fprintf(Ausgabe, "%10d\n",kn);
fprintf(Ausgabe, "*NSET,NSET=CREckUL%d%d%d,Instance=CR%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
kn=20000000+nr*10000+(aec+1)*100+1*1;
fprintf(Ausgabe, "%10d\n",kn);

```

```

    fprintf(Ausgabe, "**NSET,NSET=CREckUR%d%d%d,Instance=CR%d%d%d\n",
            nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
    kn=20000000+nr*10000+(aec+1)*100+(aempl/2+1)*1;
    fprintf(Ausgabe, "%10d,\n",kn);
    bc=0;
}
}
return Schleife;
} // KE Ende
// =====
// jedes C++Programm besitzt mindestens eine Funtion 'main( )' hier void (no return value)
// =====
void main(void)
{ // Hauptprogramm Beginn
// Öffnen bzw. Erstellen (w) der Ausgabedatei mit Namen "KnotenElemente.i"
Ausgabe = fopen("KnotenElemente.i", "w");
// Programmteildefinition => Division und Aufteilung in Ergebnis (quot) & Rest (rem)
div_t div_result;
//Knoten (yy in 2-Richtung, xx in 1-Richtung)
//0wwwyyxx... Knoten des Mineralplättchens www
//1wwwyyxx... Knoten des linken (unteren) Collagenteils
//2wwwyyxx... Knoten des rechten (unteren) Collagenteils
//3wwwyyxx... Knoten des Collagenzwickels
//Elemente
//5wwwyyxx... Elemente des Mineralplättchens www
//6wwwyyxx... Elemente des linken (unteren) Collagenteils
//7wwwyyxx... Elemente des rechten (unteren) Collagenteils
//8wwwyyxx... Elemente des Collagenzwickels
//
// Initialisierung
mpl=60; // Mineralplättchen Länge (nm = 1e-9m !!)
mpd=2.5; // Mineralplättchen Dicke
cl=2.5; // Collagen Länge (Zwischenraum zwischen Mineralplättchen)
cd=2.5; // Collagen Dicke (Zwischenraum zwischen Mineralplättchen)
// ampl soll GERADE sein
ampl=4; // Anzahl der Mineralplättchen in Längsrichtung (Richtung 1)
// ampd muss UNGERADE sein (halbes Plättchen in erster und letzter Reihe!)
ampd=11; // Anzahl der Mineralplättchen in Dickenrichtung (Richtung 2) UNGERADE!!!
// aempl muss GERADE sein (halbe Plättchen am Anfang und am Ende!)
aempl=8; // Anzahl der Mineralplättchenelemente/Plättchen in Längsrichtung (min.4)
aempd=3; // Anzahl der Mineralplättchenelemente/Plättchen in Dickenrichtung (min.2)
// aec=aecd=aempd=aecl/2
aec=4; // Anzahl der Collagenelemente/Zwischenraum (aecl=aecd*2) (max.9,min.2)
//
// Berechnung der internen Variablen
lemp=mpl/aempl; // Länge der Elemente des MineralPlättchens
demp=mpd/aempd; // Dicke der Elemente des MineralPlättchens
lec=(cl/2)/aec; // Länge der Elemente des Collagens
dec=cd/aec; // Dicke der Elemente des Collagens
//
AMPid=1; // Zähler für MP-BC Symmetrie
ACid=1; // Zähler für C-BC Symmetrie

```

```

ys=0;
fknh=00000000;           // KnotenMineral 0wwwyxx
enh=50000000;           // ElementMineral 5wwwyxx
akn=101;                 // 0101
gy=aempd;                // für SchleifeK gy Schleifenlaufvariable 0..2
hpx1=0;
hpx2=lemp;
hpy1=0;
hpy2=-demp;
gerei=1;
bc=0;                     // normale Ausgabe bei KE nicht NodeSet abhängig
//
// *****
// HAUPTPROGRAMM-Schleife
// *****
// nr = Plättchennummer
for(nr=1; nr<=(ampl*ampd); nr++)
{
    // Beginn der Hauptschleife innerhalb MAIN
    div_result = div(nr,100);
    mod100=div_result.rem;
    nr10er=mod100/10;      // mo=mod(nr,100)
    div_result = div(nr,10);
    nr1er=div_result.rem;
    h1=nr;
    h2=ampl;
    nr_durch_ampl=h1/h2;
// Berechnung der Reihe (nächste ganze Zahl = ceil)
// 0.00..1<nr_durch_ampl<1.00..0 => rei=1 ,....
    rei=ceil(nr_durch_ampl); // Calculates the ceiling of a value (next integer)
// hMPaA
    div_result = div((nr-1),(2*ampl));
    hMPaA=div_result.rem; // 0 (kein Rest) wenn halbes Plättchen am Anfang
    if (hMPaA==0)
        {hMPaA=1;} // binäres System => wenn, dann 1
    else
        {hMPaA=0;}
// hMPaE
    div_result = div((nr),(2*ampl));
    hMPaE=div_result.rem; // 0 (kein Rest) wenn halbes Plättchen am Ende
    if (hMPaE==0)
        {hMPaE=1;} // binäres System => wenn JA, dann 1
    else
        {hMPaE=0;}
// gMPaE
    div_result = div(nr,ampl);
    gMPaE=div_result.rem; // 0 (kein Rest) wenn letztes (ganzes) MP in Reihe
    if (gMPaE==0)
        {gMPaE=1;}
    else
        {gMPaE=0;}
// gerei
    div_result = div(rei,2);

```

```

    gerei=div_result.rem;                // 0 (kein Rest) wenn GERADE Reihe (2,4,6,..)
    if (gerei==0)
    {gerei=1;}
    else
    {gerei=0;}
// MP_korr
    if (gerei==0)                        // UNGERADE Reihe
    {
        MP_korrOR=-ampl;
        MP_korrUL=ampl-1;
        MP_korrUR=ampl;
    }
    else                                  // GERADE Reihe
    {
        MP_korrOR=-ampl+1;
        MP_korrUL=ampl;
        MP_korrUR=ampl+1;
    }
// aemplh
    if (hMPaA==1)
// aemplh wird für Anknüpfungspunkte gebraucht (halbe-ganze MP)
    {aemplh=aempl/2;}
    else
    {aemplh=ampl;}
// y-Startkoordinaten
    ys=-((rei-1)*(mpd+cd));              // y-Startkoordinate (links-oben) des MP
    ycs=ys+cd;
// *****
// HALBES Plättchen am Anfang (hMPaA=0) = UNGERADE Reihe, keine zw=2
// *****
    if (hMPaA==1)                        // wenn halbes Plättchen am Anfang, Rest von (nr-1)/(2*ampl)
    {
        gx=aempl/2;                      // =aemplh Hilfs-Anzahl der Elemente über die Länge des MP
        gy=aempd;                        // zuvor wurde Collagen mit aec berechnet (zumindest bei rei>1)
        xs=xcs=0;                        // x-Startkoordinate (links-oben) des Mineralplättchens/Collagens
        hp=0;
        gerei=0;                          // Halbe MP am Anfang haben nur UNGERADE Reihen
        Schleife=KE(Schleife);           // MP&C Knoten&Elemente (halbes am Anfang)
        if (rei==1)                      // oberste Teilzwickel (Reihe 1)
        {
            zw=1;
            Schleife=ZwickelKE(Schleife); // ZwickelKnoten (Elemente)
        }
        else // if (rei!=1)
        {
            if (rei!=ampd)
// ganzer Zwickel
            {
                zw=3;
                Schleife=ZwickelKE(Schleife); // ZwickelKnoten (Elemente)
            }
            else // letzte Reihe : rei=ampd

```

```

    {
        zw=4; // Bodenzwickel (Reihe ampd)
        Schleife=ZwickelKE(Schleife); // ZwickelKnoten (Elemente)
    }
}
}
// *****
// HALBES Plättchen am Ende (hMPaE=1) => KEIN CollagenZwickel!!!!
// *****
if (hMPaE==1) // wenn halbes Plättchen am Ende, Rest von (nr)/(2*ampl)
{
    gx=aempl/2; // =aemplh Hilfs-Anzahl der Elemente über die Länge des MP
    gy=aempd; // zuvor wurde Collagen mit aec berechnet
    xs=cl/2+(cl+mpl)*(ampl-1); // x-Startkoordinate (links-oben) des Mineralplättchens
    xcs=xs+mpl/2; // x-Startkoordinate (mittig) des Collagens
    gerei=1; // Halbe MP am Ende haben nur GERADE Reihen
    Schleife=KE(Schleife); // PlättchenKnoten&Elemente (halbes am Ende)
}
// *****
// GANZES Plättchen (sonst = wenn hMPaA!=1 und hMPaE!=1)
// *****
if (hMPaA!=1) // nicht Halbes am Anfang
{
    if (hMPaE!=1) // nicht Halbes am Ende
    {
        hMPaE=2;
        gx=aempl; // =aemplh Hilfs-Anzahl der Elemente über die Länge des MP
        gy=aempd; // zuvor wurde Collagen mit aec berechnet
// GERANDE Reihenzahl (2, 4, 6, 8, ...)
        if (gerei==1) // gerade Reihe
        {
            xs=cl/2+(nr-((rei-1)*ampl+1))*(cl+mpl); //x-Startkoordinate (links-oben) des MP
            xcs=xs+mpl/2;
            Schleife=KE(Schleife);
// PlättchenKnoten&Elemente (ganzes, gerade)
        }
        else
        {
// UNGERADE Reihenzahl (3, 5, 7, 9, ..) (sonst : gerei!=1)
            xs=(mpl/2+cl)+((nr-((rei-1)*ampl+2))*(cl+mpl);
// xs = x-Startkoordinate (links-oben) des MP
            xcs=xs+mpl/2;
            Schleife=KE(Schleife); // PlättchenKnoten&Elemente (ganzes, ungerade)
        }
    }
    if (rei==1) // OBERSTE Reihe (1)
    {
        if (gMPaE==1) // letztes MP in Reihe => 2 in nächster Reihe
        {
            xcs=-cl/2-mpl/2; // xcs ist im MINUS, da Spitz somit bei x=0
            ycs=ys-mpd; // da nächste Reihe!!
            zw=2;
            Schleife=ZwickelKE(Schleife); //ZwickelKnoten&Elementgenerierung

```

```

    }
    else // GANZES MP (damit nicht am Anfang) der Reihe, nicht am Ende
    {
        zw=1;
        Schleife=ZwickelKE(Schleife);
    }
}
else // NICHT OBERSTE Reihe (rei!=1)
{
    if (gMPaE==1) // ganzes MP am Ende IMMER MPaE=1
    {
        if (rei==ampd)
        {
            xcs=-c1/2-mp1/2; // xcs ist im MINUS, da Spitz somit bei x=0
            ycs=ys-mpd; // da nächste Reihe!!
            zw=2;
            Schleife=ZwickelKE(Schleife);
        }
    }
    else // NICHT ganzes MP am Ende
    {
        if (rei==ampd) // UNTERSTE Reihe (4)
        {
            zw=4;
            Schleife=ZwickelKE(Schleife);
        }
        else // NICHT UNTERSTE Reihe
        {
            zw=3;
            Schleife=ZwickelKE(Schleife);
        }
    }
}
} // if (hMPaA!=1)-Ende
// =====
// *****
// KnotenSets für BC und MPC
// *****
//
// CZOLwww 13 CollagenZwickel ObenLinks
// CZOMwww 14 CollagenZwickel ObenMitte
// CZORwww 15 CollagenZwickel ObenRechts
// CZULwww 16 CollagenZwickel UntenLinks
// CZUMwww 17 CollagenZwickel UntenMitte
// CZURwww 18 CollagenZwickel UntenRechts
// CZOQwww 19 CollagenZwickel ObenQuer
// CZUQwww 20 CollagenZwickel UntenQuer
// CZLwww 21 CollagenZwickel Links
// CZMwww 22 CollagenZwickel Mitte
// CZRwww 23 CollagenZwickel Rechts
//

```

```

if (hMPaE!=1) // nicht halbes Plättchen am Ende
{
if (nr!=(ampd*ampl)) // letztes MP hat überhaupt kein C (weder Raute noch Zwickel)
{
fprintf(Ausgabe, "*** *****\n");
fprintf(Ausgabe, "*** ** Collagen **\n");
fprintf(Ausgabe, "*** ** Zwickel **\n");
fprintf(Ausgabe, "*** ** NSETs für BC **\n");
fprintf(Ausgabe, "*** *****\n");
if (rei!=1) // schließt oberste Reihenzwickel aus
{
if (gMPaE!=1) // NICHT ganzes Plättchen am Ende
{
fprintf(Ausgabe, "*NSET,NSET=CZOL%d%d%d,Instance=CZ%d%d%d\n",
nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
printf("*NSET,NSET=CZOL%d\n",nr);
BC_Nr=13;
Schleife=ZwickelKE(Schleife);
}
fprintf(Ausgabe, "*NSET,NSET=CZOM%d%d%d,Instance=CZ%d%d%d\n",
nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
printf("*NSET,NSET=CZOM%d\n",nr);
BC_Nr=14;
Schleife=ZwickelKE(Schleife);
fprintf(Ausgabe, "*NSET,NSET=CZOR%d%d%d,Instance=CZ%d%d%d\n",
nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
printf("*NSET,NSET=CZOR%d\n",nr);
BC_Nr=15;
Schleife=ZwickelKE(Schleife);
}
}
if (rei==1)
{
if (gMPaE==1) // obwohl ganzes MP in erster Reihe, ist bei zw=2 oberer Spitz vorh.
{
fprintf(Ausgabe, "*NSET,NSET=CZOM%d%d%d,Instance=CZ%d%d%d\n",
nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
printf("*NSET,NSET=CZOM%d\n",nr);
BC_Nr=14;
Schleife=ZwickelKE(Schleife);
fprintf(Ausgabe, "*NSET,NSET=CZOR%d%d%d,Instance=CZ%d%d%d\n",
nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
printf("*NSET,NSET=CZOR%d\n",nr);
BC_Nr=15;
Schleife=ZwickelKE(Schleife);
}
}
}
if (rei!=ampd) // schließt unterste Reihenzwickel aus
{
if (gMPaE!=1) // nicht letztes MP in Reihe
{ // schließt Symmetriewickel aus
fprintf(Ausgabe, "*NSET,NSET=CZUL%d%d%d,Instance=CZ%d%d%d\n",
nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);

```

```

    printf("*NSET,NSET=CZUL%d\n",nr);
    BC_Nr=16;
    Schleife=ZwickelKE(Schleife);
}
fprintf(Ausgabe, "*NSET,NSET=CZUM%d%d%d,Instance=CZ%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
printf("*NSET,NSET=CZUM%d\n",nr);
BC_Nr=17;
Schleife=ZwickelKE(Schleife);
fprintf(Ausgabe, "*NSET,NSET=CZUR%d%d%d,Instance=CZ%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
printf("*NSET,NSET=CZUR%d\n",nr);
BC_Nr=18;
Schleife=ZwickelKE(Schleife);
}
if (gMPaE!=1)
{
    // nicht letztes MP in Reihe
    // schließt Symmetriezwickel aus
    fprintf(Ausgabe, "*NSET,NSET=CZOQ%d%d%d,Instance=CZ%d%d%d\n",
            nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
    printf("*NSET,NSET=CZOQ%d\n",nr);
    BC_Nr=19;
    Schleife=ZwickelKE(Schleife);
    fprintf(Ausgabe, "*NSET,NSET=CZUQ%d%d%d,Instance=CZ%d%d%d\n",
            nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
    printf("*NSET,NSET=CZUQ%d\n",nr);
    BC_Nr=20;
    Schleife=ZwickelKE(Schleife);
    fprintf(Ausgabe, "*NSET,NSET=CZL%d%d%d,Instance=CZ%d%d%d\n",
            nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
    printf("*NSET,NSET=CZL%d\n",nr);
    BC_Nr=21;
    Schleife=ZwickelKE(Schleife);
}
fprintf(Ausgabe, "*NSET,NSET=CZM%d%d%d,Instance=CZ%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
printf("*NSET,NSET=CZM%d\n",nr);
BC_Nr=22;
Schleife=ZwickelKE(Schleife);
fprintf(Ausgabe, "*NSET,NSET=CZR%d%d%d,Instance=CZ%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
printf("*NSET,NSET=CZR%d\n",nr);
BC_Nr=23;
Schleife=ZwickelKE(Schleife);
bc=0;
BC_Nr=1;
if (nr!=(ampl*ampd))
{
    if (hMPaE!=1)
    {
        if (zw!=2)
        // NICHT halbe CZ am Anfang
        {
            fprintf(Ausgabe, "*NSET,NSET=CZEckOL%d%d%d,Instance=CZ%d%d%d\n",

```

```

        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
    kn=30000000+nr*10000+(aec+1)*100+1*1;
    fprintf(Ausgabe, "%10d,\n",kn);
    fprintf(Ausgabe, "*NSET,NSET=CZEckUL%d%d%d,Instance=CZ%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
    kn=30000000+nr*10000+(aec+aempd+1)*100+1*1;
    fprintf(Ausgabe, "%10d,\n",kn);
}
if (nr>(ampl-1)) // für letztes MP in Reihe gilt es, da dann in nächster Zeile
{
    fprintf(Ausgabe, "*NSET,NSET=CZEckOM%d%d%d,Instance=CZ%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
    kn=30000000+nr*10000+1*100+1*1;
    fprintf(Ausgabe, "%10d,\n",kn);
}
fprintf(Ausgabe, "*NSET,NSET=CZEckOR%d%d%d,Instance=CZ%d%d%d\n",
    nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
kn=30000000+nr*10000+(aec+1)*100+(2*aec+1)*1;
fprintf(Ausgabe, "%10d,\n",kn);
fprintf(Ausgabe, "*NSET,NSET=CZEckUR%d%d%d,Instance=CZ%d%d%d\n",
    nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
kn=30000000+nr*10000+(aec+aempd+1)*100+(2*aec+1)*1;
fprintf(Ausgabe, "%10d,\n",kn);
if (rei!=ampd)
{
    fprintf(Ausgabe, "*NSET,NSET=CZEckUM%d%d%d,Instance=CZ%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
    kn=30000000+nr*10000+(aec+aempd+aec+1)*100+1*1;
    fprintf(Ausgabe, "%10d,\n",kn);
}
}
}
}
}
// AC
if (zw==2)
{
    div_result = div(ACid,10);
    mod=div_result.rem;
    fprintf(Ausgabe, "*NSET,NSET=AC%d%d\n",ACid/10,mod); // AC ... Anfang Collagen
    printf("*NSET,NSET=AC%d,\n",ACid);
    fprintf(Ausgabe, "CZOM%d%d%d, CZM%d%d%d, CZUM%d%d%d\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
    ACid=ACid+1;
}
} // ENDE Hauptschleife innerhalb MAIN
// #####
// KnotenSets verbinden - MPC's
// #####
fprintf(Ausgabe, "** #####\n");
fprintf(Ausgabe, "*MPC\n");
printf("\n");

```

```
printf("**MPC's\n");
for(nr=1; nr<=(ampl*ampd); nr++)
{
    /** nr **
    div_result = div(nr,100);
    mod100=div_result.rem;
    nr10er=mod100/10;
    div_result = div(nr,10);
    nr1er=div_result.rem;
    // ** nrOLg ** für MP in GERADEN Reihen
    nrOLg=nr-ampl;
    div_result = div(nrOLg,100);
    mod100=div_result.rem;
    nrOLg10er=mod100/10;
    div_result = div(nrOLg,10);
    nrOLg1er=div_result.rem;
    // ** nrOLu ** für MP in UNGERADEN Reihen
    nrOLu=nr-ampl-1;
    div_result = div(nrOLu,100);
    mod100=div_result.rem;
    nrOLu10er=mod100/10;
    div_result = div(nrOLu,10);
    nrOLu1er=div_result.rem;
    // ** nrORg ** für MP in GERADEN Reihen
    nrORg=nr-ampl+1;
    div_result = div(nrORg,100);
    mod100=div_result.rem;
    nrORg10er=mod100/10;
    div_result = div(nrORg,10);
    nrORg1er=div_result.rem;
    // ** nrORu ** für MP in UNGERADEN Reihen
    nrORu=nr-ampl;
    div_result = div(nrORu,100);
    mod100=div_result.rem;
    nrORu10er=mod100/10;
    div_result = div(nrORu,10);
    nrORu1er=div_result.rem;
    // ** nrL **
    nrL=nr-1;
    div_result = div(nrL,100);
    mod100=div_result.rem;
    nrL10er=mod100/10;
    div_result = div(nrL,10);
    nrL1er=div_result.rem;
    // ** nrULg ** für MP in GERADEN Reihen
    nrULg=nr+ampl;
    div_result = div(nrULg,100);
    mod100=div_result.rem;
    nrULg10er=mod100/10;
    div_result = div(nrULg,10);
    nrULg1er=div_result.rem;
    // ** nrULu ** für MP in UNGERADEN Reihen
```

// Beginn der for-Hauptschleife

```

nrULu=nr+ampl-1;
div_result = div(nrULu,100);
mod100=div_result.rem;
nrULu10er=mod100/10;
div_result = div(nrULu,10);
nrULu1er=div_result.rem;
//
h1=nr;
h2=ampl;
nr_durch_ampl=h1/h2;
// Berechnung der Reihe (nächste ganze Zahl = ceil)
// 0.00..1<nr_durch_ampl<1.00..0 => rei=1 ,....
rei=ceil(nr_durch_ampl); // Calculates the ceiling of a value (next integer)
// gerei
div_result = div(rei,2);
gerei=div_result.rem; // 0 (kein Rest) wenn gerade Reihe (2,4,6,..)
if (gerei==0)
{gerei=1;} // gerei=1 wenn GERADE Reihe
else
{gerei=0;}
// nrOLg, nrOLu,.....
if (gerei==1) // GERADE Reihe
{
nrOL=nrOLg;
nrOL10er=nrOLg10er;
nrOL1er=nrOLg1er;
nrOR=nrORg;
nrOR10er=nrORg10er;
nrOR1er=nrORg1er;
nrUL=nrULg;
nrUL10er=nrULg10er;
nrUL1er=nrULg1er;
}
else // UNGERADE Reihe
{
nrOL=nrOLu;
nrOL10er=nrOLu10er;
nrOL1er=nrOLu1er;
nrOR=nrORu;
nrOR10er=nrORu10er;
nrOR1er=nrORu1er;
nrUL=nrULu;
nrUL10er=nrULu10er;
nrUL1er=nrULu1er;
}
// hMPaA
div_result = div((nr-1),(2*ampl));
hMPaA=div_result.rem; // 0 (kein Rest) wenn halbes Plättchen am Anfang
if (hMPaA==0)
{hMPaA=1;} // binäres System => wenn, dann 1
else
{hMPaA=0;}

```

```

// hMPaE
div_result = div((nr),(2*ampl));
hMPaE=div_result.rem;           // 0 (kein Rest) wenn halbes Plättchen am Ende
if (hMPaE==0)
    {hMPaE=1;}                 // binäres System => wenn, dann 1
else
    {hMPaE=0;}

// gMPaE
div_result = div(nr,ampl);
gMPaE=div_result.rem;          // 0 (kein Rest) wenn letztes (ganzes) MP in Reihe
if (gMPaE==0)
    {gMPaE=1;}                 // binäres System => wenn, dann 1
else
    {gMPaE=0;}

//
fprintf(Ausgabe, "*** #####\n");
fprintf(Ausgabe, "*** ## MPC-Bereich Mineralplättchen %d%d%d ##\n",
        nr/100,nr10er,nr1er);
fprintf(Ausgabe, "*** #####\n");
printf("BC's Mineralplaettchen%d\n",nr);
//OBEN und UNTEN schichtweise verbunden : MP und C-Rauten
if (rei==1)
    {
        if (hMPaA!=1)
            {fprintf(Ausgabe, "PIN,MPUL%d%d%d,CLO%d%d%d,\n",
                nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);}
        fprintf(Ausgabe, "PIN,MPUR%d%d%d,CRO%d%d%d,\n",
            nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
    }
else // if (rei!=1)
    {
        if (rei!=ampd)
            {
                if (gerei==1) // GERADE Reihe
                    {
                        if (hMPaE!=1)
                            {
                                fprintf(Ausgabe, "PIN,MPOR%d%d%d,CLU%d%d%d,\n",
                                    nr/100,nr10er,nr1er,nrORg/100,nrORg10er,nrORg1er);
                                fprintf(Ausgabe, "PIN,MPUR%d%d%d,CRO%d%d%d,\n",
                                    nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
                            }
                        fprintf(Ausgabe, "PIN,MPOL%d%d%d,CRU%d%d%d,\n",
                            nr/100,nr10er,nr1er,nrOLg/100,nrOLg10er,nrOLg1er);
                        fprintf(Ausgabe, "PIN,MPUL%d%d%d,CLO%d%d%d,\n",
                            nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
                    }
            }
        else // UNGERADE Reihe
            {
                if (hMPaA!=1)
                    {
                        fprintf(Ausgabe, "PIN,MPOL%d%d%d,CRU%d%d%d,\n",
                            nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
                    }
            }
    }

```

```

        nr/100,nr10er,nr1er,nrOLu/100,nrOLu10er,nrOLu1er);
    fprintf(Ausgabe, "PIN,MPUL%d%d%d,CLO%d%d%d,\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
    }
    fprintf(Ausgabe, "PIN,MPOR%d%d%d,CLU%d%d%d,\n",
        nr/100,nr10er,nr1er,nrORu/100,nrORu10er,nrORu1er);
    fprintf(Ausgabe, "PIN,MPUR%d%d%d,CRO%d%d%d,\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
    }
}
else // rei=ampl LETZTE Reihe (ist auch immer ungerade)
{
    if (hMPaA!=1)
    {fprintf(Ausgabe, "PIN,MPOL%d%d%d,CRU%d%d%d,\n",
        nr/100,nr10er,nr1er,nrOLu/100,nrOLu10er, nrOLu1er);}
    fprintf(Ausgabe, "PIN,MPOR%d%d%d,CLU%d%d%d,\n",
        nr/100,nr10er,nr1er,nrORu/100,nrORu10er,nrORu1er);
    }
}
//LINKS und RECHTS schichtweise verbunden : MP und C-Zwickel
if (hMPaA!=1)
{fprintf(Ausgabe, "PIN,MPL%d%d%d,CZR%d%d%d,\n",
    nr/100,nr10er,nr1er,nrL/100,nrL10er,nrL1er);}
if (hMPaE!=1)
{
    if (gMPaE!=1)
    {fprintf(Ausgabe, "PIN,MPR%d%d%d,CZL%d%d%d,\n",
        nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);}
}
//LINKS und RECHTS schichtweise verbunden : C-Raute und C-Zwickel
if (rei!=ampd)
{
    if (gerei!=1) // UNGERADE Reihe
    {
        if (hMPaA!=1)
        {
            fprintf(Ausgabe, "PIN,CLL%d%d%d,CZUR%d%d%d,\n",
                nr/100,nr10er,nr1er,nrL/100,nrL10er,nrL1er);
            fprintf(Ausgabe, "PIN,CLR%d%d%d,CZOL%d%d%d,\n",
                nr/100,nr10er,nr1er,nrULu/100,nrULu10er,nrULu1er);
        }
        fprintf(Ausgabe, "PIN,CRL%d%d%d,CZOR%d%d%d,\n",
            nr/100,nr10er,nr1er,nrULu/100,nrULu10er,nrULu1er);
        if (gMPaE!=1)
        {fprintf(Ausgabe, "PIN,CRR%d%d%d,CZUL%d%d%d,\n",
            nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);}
    }
}
else // GERADE Reihe
{
    fprintf(Ausgabe, "PIN,CLL%d%d%d,CZUR%d%d%d,\n",
        nr/100,nr10er,nr1er,nrL/100,nrL10er,nrL1er);
    if (hMPaE!=1)

```

```

    {
        fprintf(Ausgabe, "PIN,CLR%d%d%d,CZOL%d%d%d,\n",
                nr/100,nr10er,nr1er,nrULg/100,nrULg10er,nrULg1er);
        fprintf(Ausgabe, "PIN,CRL%d%d%d,CZOR%d%d%d,\n",
                nr/100,nr10er,nr1er,nrULg/100,nrULg10er,nrULg1er);
        fprintf(Ausgabe, "PIN,CRR%d%d%d,CZUL%d%d%d,\n",
                nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
    }
}
}
// gemeinsame Eckpunkte pinnen
// MP & CZ
// ##### MPEckOL #####
// ACHTUNG : REIHENFOLGE
// ***** OBEN *****
// MPEckOL : CZ_L - CR_OL, CR_OL - MP
if (hMPaA!=1)
{
    if (rei!=1)
    {
        fprintf(Ausgabe, "PIN,CZEckOR%d%d%d,MPEckOL%d%d%d,\n",
                nrL/100,nrL10er,nrL1er,nr/100,nr10er,nr1er);
        fprintf(Ausgabe, "PIN,CREckUL%d%d%d,MPEckOL%d%d%d,\n",
                nrOL/100,nrOL10er,nrOL1er,nr/100,nr10er,nr1er);
    }
    else
    {fprintf(Ausgabe, "PIN,MPEckOL%d%d%d,CZEckOR%d%d%d,\n",
            nr/100,nr10er,nr1er,nrL/100,nrL10er,nrL1er);}
}
// MPEckOM : CZ_OL - CR_OL, CR_OL - CL_OR, CL_OR - MP
if (rei!=1)
{
    if (hMPaA!=1)
    {
        if (hMPaE!=1)
        {
            fprintf(Ausgabe, "PIN,CZEckUM%d%d%d,MPEckOM%d%d%d,\n",
                    nrOL/100,nrOL10er,nrOL1er,nr/100,nr10er,nr1er);
            fprintf(Ausgabe, "PIN,CLEckUL%d%d%d,MPEckOM%d%d%d,\n",
                    nrOR/100,nrOR10er,nrOR1er,nr/100,nr10er,nr1er);
            fprintf(Ausgabe, "PIN,CREckUR%d%d%d,MPEckOM%d%d%d,\n",
                    nrOL/100,nrOL10er,nrOL1er,nr/100,nr10er,nr1er);
        }
        else
        {
            fprintf(Ausgabe, "PIN,CREckUR%d%d%d,MPEckOM%d%d%d,\n",
                    nrOL/100,nrOL10er,nrOL1er,nr/100,nr10er,nr1er);
        }
    }
}
else
{
    fprintf(Ausgabe, "PIN,CZEckUM%d%d%d,MPEckOM%d%d%d,\n",

```

```

        nrOL/100,nrOL10er,nrOL1er,nr/100,nr10er,nr1er);
    fprintf(Ausgabe, "PIN,CLEckUL%d%d%d,MPEckOM%d%d%d,\n",
        nrOR/100,nrOR10er,nrOR1er,nr/100,nr10er,nr1er);
    }
}
// MPEckOR : CZ - CL_OR, CL_OR - MP
if (hMPaE!=1)
{
    if (rei!=1)
    {
        if (gMPaE!=1)
        {
            fprintf(Ausgabe, "PIN,CZEckOL%d%d%d,MPEckOR%d%d%d,\n",
                nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
            fprintf(Ausgabe, "PIN,CLEckUR%d%d%d,MPEckOR%d%d%d,\n",
                nrOR/100,nrOR10er,nrOR1er,nr/100,nr10er,nr1er);
        }
        else
        {fprintf(Ausgabe, "PIN,CLEckUR%d%d%d,MPEckOR%d%d%d,\n",
            nrOR/100,nrOR10er,nrOR1er,nr/100,nr10er,nr1er);}
        }
    else
    {
        if (gMPaE!=1)
        {fprintf(Ausgabe, "PIN,MPEckOR%d%d%d,CZEckOL%d%d%d,\n",
            nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);}
        }
    }
}
// ***** UNTEN *****
// MPEckUL : CZ_L - CL, CL - MP
if (hMPaA!=1)
{
    if (rei!=ampd)
    {
        fprintf(Ausgabe, "PIN,CZEckUR%d%d%d,MPEckUL%d%d%d,\n",
            nrL/100,nrL10er,nrL1er,nr/100,nr10er,nr1er);
        fprintf(Ausgabe, "PIN,CLEckOL%d%d%d,MPEckUL%d%d%d,\n",
            nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
    }
    else
    {fprintf(Ausgabe, "PIN,MPEckUL%d%d%d,CZEckUR%d%d%d,\n",
        nr/100,nr10er,nr1er,nrL/100,nrL10er,nrL1er);}
}
// MPEckUM : CZ_UL - CL, CL - CR, CR - MP
if (rei!=ampd)
{
    if (hMPaA!=1)
    {
        if (hMPaE!=1)
        {
            fprintf(Ausgabe, "PIN,CZEckOM%d%d%d,MPEckUM%d%d%d,\n",
                nrUL/100,nrUL10er,nrUL1er,nr/100,nr10er,nr1er);

```

```

        fprintf(Ausgabe, "PIN,CLEckOR%d%d%d,MPEckUM%d%d%d,\n",
                nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
        fprintf(Ausgabe, "PIN,CREckOL%d%d%d,MPEckUM%d%d%d,\n",
                nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
    }
    else
    {fprintf(Ausgabe, "PIN,CLEckOR%d%d%d,MPEckUM%d%d%d,\n",
            nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);}
}
else
{
    fprintf(Ausgabe, "PIN,CZEckOM%d%d%d,MPEckUM%d%d%d,\n",
            nrUL/100,nrUL10er,nrUL1er,nr/100,nr10er,nr1er);
    fprintf(Ausgabe, "PIN,CREckOL%d%d%d,MPEckUM%d%d%d,\n",
            nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
}
}
// MPEckUR : CZ - CR, CR - MP
if (hMPaE!=1)
{
    if (gMPaE!=1)
    {
        if (rei!=ampd)
        {
            fprintf(Ausgabe, "PIN,CZEckUL%d%d%d,MPEckUR%d%d%d,\n",
                    nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
            fprintf(Ausgabe, "PIN,CREckOR%d%d%d,MPEckUR%d%d%d,\n",
                    nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
        }
        else
        {fprintf(Ausgabe, "PIN,MPEckUR%d%d%d,CZEckUL%d%d%d,\n",
                nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);}
    }
    else
    {
        if (rei!=ampd)
        {fprintf(Ausgabe, "PIN,CREckOR%d%d%d,MPEckUR%d%d%d,\n",
                nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);}
    }
}
}
// Ende der for-Hauptschleife
// #####
// INDENTER
// #####
fprintf(Ausgabe, "*** #####\n");
fprintf(Ausgabe, "*** ##      INDENTER      ##\n");
fprintf(Ausgabe, "*** ##      Knoten plus Elemente      ##\n");
fprintf(Ausgabe, "*** #####\n");
//  $b=1/2*\tan((2*\alpha)/4)$ 
fprintf(Ausgabe, "*Instance, name=IndenterTip, part=Indenter\n");
printf("*Instance, name=IndenterTip, part=Indenter\n");
fprintf(Ausgabe, "*** Region:\n");

```

```

fprintf(Ausgabe, "*Element, type=MASS, elset=TipElset\n");
fprintf(Ausgabe, "1,1\n");
fprintf(Ausgabe, "*Mass, elset=TipElset\n");
fprintf(Ausgabe, "1000.,\n");
fprintf(Ausgabe, "*** Radienmittelpunkt: (0,100nm)\n");
fprintf(Ausgabe, "*NODE\n");
fprintf(Ausgabe, "1,0,100\n");
fprintf(Ausgabe, "*NSET, NSET=TipRP\n");
fprintf(Ausgabe, "1,\n");
fprintf(Ausgabe, "*SURFACE, TYPE=SEGMENTS, name=RigidSurfaceTip\n");
fprintf(Ausgabe, "START,470.71067812,429.28932188\n");
fprintf(Ausgabe, "LINE,70.71067812,29.28932188\n");
fprintf(Ausgabe, "CIRCL,0,0,0,100\n");
fprintf(Ausgabe, "*RIGID BODY,ref node=TipRP,analytical surface=RigidSurfaceTip\n");
// nr=0;
// rTip=100e-09;// 100nm
// for (a=0; a<=450; a++)
// {
//   nr=nr+1;
//   xTip=rTip*sin((Pi/180)*(a/10));
//   yTip=rTip-(rTip*cos((Pi/180)*(a/10))); // +1e+20 stellt Kontakt sicher (kein HCRIT)
//   fprintf(Ausgabe, "%8d,\t%2.4e,\t%2.4e\n",nr,xTip,yTip);
// }
// fprintf(Ausgabe, " 452,\t5.0000e-007,\t%2.4e\n", (yTip+((5e-07)-xTip)));
// fprintf(Ausgabe, " 453,\t5.0000e-007,\t5.0000e-007\n");
// fprintf(Ausgabe, " 454,\t0.0000e+000,\t5.0000e-007\n");
// fprintf(Ausgabe, "*ELEMENT,TYPE=R2D2\n");
// printf(" *ELEMENT,TYPE=R2D2\n");
// for (b=1; b<=453; b++)
// { fprintf(Ausgabe, "%8d,%12d,%12d\n",b, b, (b+1));}
// fprintf(Ausgabe, "*NODE\n");
// fprintf(Ausgabe, " 500,\t0.,\t100e-09,\t0.\n",x[akn]);
// fprintf(Ausgabe, "*NSET, NSET=TipRP\n");
// fprintf(Ausgabe, " 500,\n");
// printf(Ausgabe, "*Elset, elset=IndenterTip,generate\n");
// fprintf(Ausgabe, " 1, 454, 1\n");
fprintf(Ausgabe, "*END Instance\n");
printf(" *END Instance\n");
// fprintf(Ausgabe, "*Elset, elset=TipE,Instance=IndenterTip,generate\n");
// fprintf(Ausgabe, " 1, 450, 1\n");
// fprintf(Ausgabe, "*NSET, NSET=SymTip,Instance=IndenterTip\n");
// fprintf(Ausgabe, " 1, 454,\n");
// #####
// #####
// ELEMENTSET für Kontakt (MP001E, CZ001E, MP002E,...)
// #####
for (nr=1; nr<=ampl; nr++)
{
  div_result = div(nr,100);
  mod100=div_result.rem;
  nr10er=mod100/10;
  div_result = div(nr,10);

```

```

    nr1er=div_result.rem;
// Mineralplättchen
    b=1;
    if (nr==1)
        {gx=aempl/2;}
    else
        {gx=aempl;}
    gy=1;           // 1 reicht, da nur OBERSTE Elementreihe (Rechenzeitverkürzung)
    fknh=00000000; // KnotenMineral 0wwwyxxx
    enh=50000000;  // ElementMineral 5wwwyxxx
    Schleife=SchleifeE(Schleife);
// CollagenZwickel
    if (nr<ampl)
    {
        b=2;
        gx=aec*2;
        fknh=30000000+aec*100; // KnotenCollagenZwickel 3wwwyxxx
        enh=0+aec*100;
        Schleife=SchleifeE(Schleife);
    }
}
// #####
// GLOBALE ElementSets  OBEN, SET1
// #####
// MINERALPLÄTTCHEN
fprintf(Ausgabe, "*** #####\n");
fprintf(Ausgabe, "*** ##  GLOBALE ElementSet's  ##\n");
fprintf(Ausgabe, "*** #####\n");
// OBEN (für Oberflächen-Kontakt)
fprintf(Ausgabe, "*ELSET,ELSET=OBEN\n");
printf("\n*ELSET,ELSET=OBEN\n");
for (nr=1; nr<=ampl; nr++)
{
    // Oben-for Anfang
    div_result = div(nr,100);
    mod100=div_result.rem;
    nr10er=mod100/10;
    div_result = div(nr,10);
    nr1er=div_result.rem;
    fprintf(Ausgabe, "MP%d%d%dE,\n",nr/100,nr10er,nr1er);
    if (nr<ampl)
        {fprintf(Ausgabe, "CZ%d%d%dE,\n",nr/100,nr10er,nr1er);}
}
// Oben-for Ende
// SET1 (für Remeshing => ALLE CollagenElemente)
for (nr=1; nr<=(ampd*ampl); nr++)
{
    // Set1-for Anfang
    div_result = div(nr,100);
    mod100=div_result.rem;
    nr10er=mod100/10;
    div_result = div(nr,10);
    nr1er=div_result.rem;
    h1=nr;
    h2=ampl;

```

```

    nr_durch_ampl=h1/h2;
    rei=ceil(nr_durch_ampl);
// hMPaA
    div_result = div((nr-1),(2*ampl));
    hMPaA=div_result.rem; // 0 (kein Rest) wenn halbes Plättchen am Anfang
    if (hMPaA==0) // binäres System => wenn JA, dann 1
    {hMPaA=1;}
    else
    {hMPaA=0;}
// hMPaE
    div_result = div((nr),(2*ampl));
    hMPaE=div_result.rem; // 0 (kein Rest) wenn halbes Plättchen am Ende
    if (hMPaE==0) // binäres System => wenn JA, dann 1
    {hMPaE=1;}
    else
    {hMPaE=0;}
//
if (rei<ampd)
{
    if (hMPaE!=1)
    {
        fprintf(Ausgabe, "*ELSET,ELSET=%d%d%dR,Instance=CR%d%d%d\n",
                nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
        fprintf(Ausgabe, "AlleE,\n");
    }
    if (hMPaA!=1)
    {
        fprintf(Ausgabe, "*ELSET,ELSET=%d%d%dL,Instance=CL%d%d%d\n",
                nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
        fprintf(Ausgabe, "AlleE,\n");
    }
}
}
if (nr!=(ampl*ampd))
{
    if (hMPaE!=1)
    {
        fprintf(Ausgabe, "*ELSET,ELSET=%d%d%dZ,Instance=CZ%d%d%d\n",
                nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);
        fprintf(Ausgabe, "AlleE,\n");
    }
}
} // Set1-for Ende
fprintf(Ausgabe, "*ELSET,ELSET=SET1\n");
for (nr=1; nr<=(ampd*ampl); nr++)
{
    div_result = div(nr,100);
    mod100=div_result.rem;
    nr10er=mod100/10;
    div_result = div(nr,10);
    nr1er=div_result.rem;
    h1=nr;
    h2=ampl;

```

```

    nr_durch_ampl=h1/h2;
    rei=ceil(nr_durch_ampl);
// hMPaA
    div_result = div((nr-1),(2*ampl));
    hMPaA=div_result.rem;           // 0 (kein Rest) wenn halbes Plättchen am Anfang
    if (hMPaA==0)
        {hMPaA=1;}                 // binäres System => wenn JA, dann 1
    else
        {hMPaA=0;}
// hMPaE
    div_result = div((nr),(2*ampl));
    hMPaE=div_result.rem;         // 0 (kein Rest) wenn halbes Plättchen am Ende
    if (hMPaE==0)
        {hMPaE=1;}                 // binäres System => wenn JA, dann 1
    else
        {hMPaE=0;}
//
    if (rei<ampd)
    {
        if (hMPaE==1)
            {fprintf(Ausgabe, "%d%d%dL,\n",nr/100,nr10er,nr1er);}
        else
        {
            if (hMPaA==1)
                {fprintf(Ausgabe, "%d%d%dR,%d%d%dZ,\n",
                    nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);}
            else
                {fprintf(Ausgabe, "%d%d%dR,%d%d%dL,%d%d%dZ,\n",
                    nr/100,nr10er,nr1er,nr/100,nr10er,nr1er,nr/100,nr10er,nr1er);}
        }
    }
    else
    {
        if (nr!=(ampl*ampd))
            {fprintf(Ausgabe, "%d%d%dZ,\n",nr/100,nr10er,nr1er);}
    }
}
// #####
// GLOBALE KnotenSets SYMy, OBEN
// #####
fprintf(Ausgabe, "*** #####\n");
fprintf(Ausgabe, "*** ## GLOBALE NodeSet's ##\n");
fprintf(Ausgabe, "*** #####\n");
// SYMyMP SymetrieKnotenSet der Mineralplättchen um y
fprintf(Ausgabe, "*NSET,NSET=SYMyMP\n");
printf("\n*NSET,NSET=SYMyMP\n");
for (ic=1; ic<=((ampd-1)/2+1); ic++)
{
    div_result = div(ic,10);
    mod=div_result.rem;
    fprintf(Ausgabe, "AMP%d%d,\n",ic/10,mod);
}

```

```

// SYMyC SymetrieKnotenSet der Mineralplättchen um y
fprintf(Ausgabe, "*NSET,NSET=SYMyC\n");
printf("*NSET,NSET=SYMyC\n");
for (ic=1; ic<=((ampd-1)/2); ic++)
{
    div_result = div(ic,10);
    mod=div_result.rem;
    fprintf(Ausgabe, "AC%d%d,\n",ic/10,mod);
}
// SYMy SymetrieKnotenSet um y
fprintf(Ausgabe, "*NSET,NSET=SYMy\n");
printf("*NSET,NSET=SYMy\n");
fprintf(Ausgabe, "SYMyMP,SYMyC,\n");
// XSYMM GesamtSymetrieKnotenSet um y
fprintf(Ausgabe, "*NSET,NSET=XSYMM\n");
printf("*NSET,NSET=XSYMM\n");
fprintf(Ausgabe, "SYMy,\n");
// fprintf(Ausgabe, "SYMy,SymTip,\n");
// ObenKN (für Oberflächen-Kontakt)
fprintf(Ausgabe, "*NSET,NSET=OBEN\n");
printf("*NSET,NSET=OBEN\n");
for (nr=1; nr<=ampl; nr++)
{
    // ObenKN-for Anfang
    div_result = div(nr,100);
    mod100=div_result.rem;
    nr10er=mod100/10;
    div_result = div(nr,10);
    nr1er=div_result.rem;
    if (nr==1)
    {
        fprintf(Ausgabe, "MPOR001,\n");
        fprintf(Ausgabe, "CZOQ001,\n");
    }
    else
    {
        if (nr!=ampl)
        {
            fprintf(Ausgabe, "MPOL%d%d%d,\n",nr/100,nr10er,nr1er);
            fprintf(Ausgabe, "MPOR%d%d%d,\n",nr/100,nr10er,nr1er);
            fprintf(Ausgabe, "CZOQ%d%d%d,\n",nr/100,nr10er,nr1er);
        }
        else
        {
            fprintf(Ausgabe, "MPOL%d%d%d,\n",nr/100,nr10er,nr1er);
            fprintf(Ausgabe, "MPOR%d%d%d,\n",nr/100,nr10er,nr1er);
        }
    }
}
// ObenKN-for Ende
// UNTEN (für BC uy=0)
fprintf(Ausgabe, "*NSET,NSET=UNTEN\n");
printf("*NSET,NSET=UNTEN\n");
nr=(ampl*ampd-ampl+1);

```

```

div_result = div(nr,100);
mod100=div_result.rem;
nr10er=mod100/10;
div_result = div(nr,10);
nr1er=div_result.rem;
fprintf(Ausgabe, "MPEckUM%d%d%d,\n",nr/100,nr10er,nr1er);
for (nr=((ampl*ampd)-ampl+1); nr<=(ampl*ampd); nr++)
{
    // UNTEN-for Anfang
    div_result = div(nr,100);
    mod100=div_result.rem;
    nr10er=mod100/10;
    div_result = div(nr,10);
    nr1er=div_result.rem;
    if (nr==((ampl*ampd)-ampl+1))
    {
        fprintf(Ausgabe, "MPUR%d%d%d,\n",nr/100,nr10er,nr1er);
        if (nr!=(ampl*ampd))
        {fprintf(Ausgabe, "CZUQ%d%d%d,\n",nr/100,nr10er,nr1er);}
    }
    else
    {
        if (nr!=(ampl*ampd))
        {
            fprintf(Ausgabe, "MPUL%d%d%d,\n",nr/100,nr10er,nr1er);
            fprintf(Ausgabe, "MPUR%d%d%d,\n",nr/100,nr10er,nr1er);
            fprintf(Ausgabe, "CZUQ%d%d%d,\n",nr/100,nr10er,nr1er);
        }
        else
        {
            fprintf(Ausgabe, "MPUL%d%d%d,\n",nr/100,nr10er,nr1er);
            fprintf(Ausgabe, "MPUR%d%d%d,\n",nr/100,nr10er,nr1er);
        }
    }
}
// UNTEN-for Ende
// Hauptprogramm Ende
// =====

```

b) Auszug aus der mittels des Quellcodes erstellten Datei "KnotenElemente.i"

Mittels der in **Anhang a)** angegebenen C++ Datei wird ein Textfile mit Namen "KnotenElemente.i" erzeugt. Man erkennt den strukturierten Aufbau, auch wenn die Datei hier aus Platzgründen gekürzt werden muss. Man erkennt jeweils die Daten für das erste (halbe) und das letzte Mineralplättchen sowie für den Indenter. Auch die für die *.inp Datei nötigen Knotensets sind ersichtlich. Zur besseren Verständlichkeit wurden die selbstdefinierten Namensbezeichnungen in rot und die ABAQUS-spezifischen Variablen (z.B. Elementtyp) in blau dargestellt. Diese Datei wird in die ABAQUS-*.inp Datei (**Anhang c)**) eingelesen.

```

** #####
** ## Bereich : Mineralplättchen 001 ##
** #####
** *****
** ** Mineralplättchen **
** ** Knoten plus Elemente **
** *****
*Instance, name=MP001, part=Mineralplättchen
*NODE
  10101, 0.0000e+000, 0.0000e+000
  10102, 7.5000e+000, 0.0000e+000
  10103, 1.5000e+001, 0.0000e+000
  10104, 2.2500e+001, 0.0000e+000
  10105, 3.0000e+001, 0.0000e+000
  10201, 0.0000e+000, -8.3333e-001
  10202, 7.5000e+000, -8.3333e-001
  10203, 1.5000e+001, -8.3333e-001
  10204, 2.2500e+001, -8.3333e-001
  10205, 3.0000e+001, -8.3333e-001
  10301, 0.0000e+000, -1.6667e+000
  10302, 7.5000e+000, -1.6667e+000
  10303, 1.5000e+001, -1.6667e+000
  10304, 2.2500e+001, -1.6667e+000
  10305, 3.0000e+001, -1.6667e+000
  10401, 0.0000e+000, -2.5000e+000
  10402, 7.5000e+000, -2.5000e+000
  10403, 1.5000e+001, -2.5000e+000
  10404, 2.2500e+001, -2.5000e+000
  10405, 3.0000e+001, -2.5000e+000
*ELEMENT,TYPE=CPE4R
  50010101, 10101, 10201, 10202, 10102
  50010102, 10102, 10202, 10203, 10103
  50010103, 10103, 10203, 10204, 10104
  50010104, 10104, 10204, 10205, 10105
  50010201, 10201, 10301, 10302, 10202
  50010202, 10202, 10302, 10303, 10203
  50010203, 10203, 10303, 10304, 10204
  50010204, 10204, 10304, 10305, 10205
  50010301, 10301, 10401, 10402, 10302
  50010302, 10302, 10402, 10403, 10303
  50010303, 10303, 10403, 10404, 10304
  50010304, 10304, 10404, 10405, 10305
** Region:
*Elset, elset=AlleE
  50010101,
  50010102,
  50010103,
  50010104,
  50010201,
  50010202,
  50010203,
  50010204,

```

```
50010301,
50010302,
50010303,
50010304,
** Section:
*SOLID SECTION, ELSET=AlleE, MATERIAL=Mineral
1.,
*END Instance
** *****
** ** MineralPlättchen **
** ** NSETs für BC **
** *****
*NSET,NSET=AMP01,Instance=MP001
  10101,
  10201,
  10301,
  10401,
*NSET,NSET=MPOR001,Instance=MP001
  10102,
  10103,
  10104,
*NSET,NSET=MPUR001,Instance=MP001
  10402,
  10403,
  10404,
*NSET,NSET=MPR001,Instance=MP001
  10205,
  10305,
*NSET,NSET=MPL001,Instance=MP001
  10201,
  10301,
*NSET,NSET=MPEckOM001,Instance=MP001
  10101,
*NSET,NSET=MPEckUM001,Instance=MP001
  10401,
*NSET,NSET=MPEckOR001,Instance=MP001
  10105,
*NSET,NSET=MPEckUR001,Instance=MP001
  10405,
** *****
** ** Collagen Rechts **
** ** Knoten plus Elemente **
** *****
*Instance, name=CR001, part=CollagenRechts
*NODE
  20010101, 0.0000e+000, -2.5000e+000
  20010102, 7.5000e+000, -2.5000e+000
  20010103, 1.5000e+001, -2.5000e+000
  20010104, 2.2500e+001, -2.5000e+000
  20010105, 3.0000e+001, -2.5000e+000
  20010201, 3.1250e-001, -3.1250e+000
  20010202, 7.8125e+000, -3.1250e+000
```

20010203, 1.5313e+001, -3.1250e+000
20010204, 2.2813e+001, -3.1250e+000
20010205, 3.0313e+001, -3.1250e+000
20010301, 6.2500e-001, -3.7500e+000
20010302, 8.1250e+000, -3.7500e+000
20010303, 1.5625e+001, -3.7500e+000
20010304, 2.3125e+001, -3.7500e+000
20010305, 3.0625e+001, -3.7500e+000
20010401, 9.3750e-001, -4.3750e+000
20010402, 8.4375e+000, -4.3750e+000
20010403, 1.5938e+001, -4.3750e+000
20010404, 2.3438e+001, -4.3750e+000
20010405, 3.0938e+001, -4.3750e+000
20010501, 1.2500e+000, -5.0000e+000
20010502, 8.7500e+000, -5.0000e+000
20010503, 1.6250e+001, -5.0000e+000
20010504, 2.3750e+001, -5.0000e+000
20010505, 3.1250e+001, -5.0000e+000

*ELEMENT,TYPE=CPE4R

70010101, 20010101, 20010201, 20010202, 20010102
70010102, 20010102, 20010202, 20010203, 20010103
70010103, 20010103, 20010203, 20010204, 20010104
70010104, 20010104, 20010204, 20010205, 20010105
70010201, 20010201, 20010301, 20010302, 20010202
70010202, 20010202, 20010302, 20010303, 20010203
70010203, 20010203, 20010303, 20010304, 20010204
70010204, 20010204, 20010304, 20010305, 20010205
70010301, 20010301, 20010401, 20010402, 20010302
70010302, 20010302, 20010402, 20010403, 20010303
70010303, 20010303, 20010403, 20010404, 20010304
70010304, 20010304, 20010404, 20010405, 20010305
70010401, 20010401, 20010501, 20010502, 20010402
70010402, 20010402, 20010502, 20010503, 20010403
70010403, 20010403, 20010503, 20010504, 20010404
70010404, 20010404, 20010504, 20010505, 20010405

*Elset, elset=AlleE

70010101,
70010102,
70010103,
70010104,
70010201,
70010202,
70010203,
70010204,
70010301,
70010302,
70010303,
70010304,
70010401,
70010402,
70010403,
70010404,

```
*SOLID SECTION, ELSET=AlleE, MATERIAL=CollagenLR
```

```
1.,
```

```
*END Instance
```

```
** *****
```

```
** ** Collagen Rechts **
```

```
** ** NSETs für BC **
```

```
** *****
```

```
*NSET,NSET=CRO001,Instance=CR001
```

```
20010102,
```

```
20010103,
```

```
20010104,
```

```
*NSET,NSET=CRU001,Instance=CR001
```

```
20010502,
```

```
20010503,
```

```
20010504,
```

```
*NSET,NSET=CRR001,Instance=CR001
```

```
20010205,
```

```
20010305,
```

```
20010405,
```

```
*NSET,NSET=CRL001,Instance=CR001
```

```
20010201,
```

```
20010301,
```

```
20010401,
```

```
*NSET,NSET=CREckOL001,Instance=CR001
```

```
20010101,
```

```
*NSET,NSET=CREckOR001,Instance=CR001
```

```
20010105,
```

```
*NSET,NSET=CREckUL001,Instance=CR001
```

```
20010501,
```

```
*NSET,NSET=CREckUR001,Instance=CR001
```

```
20010505,
```

```
** *****
```

```
** ** Collagen **
```

```
** ** Zwickel **
```

```
** ** Knoten plus Elemente **
```

```
** *****
```

```
*Instance, name=CZ001, part=CollagenZwickel
```

```
*NODE
```

```
30010501, 3.0000e+001, 0.0000e+000
```

```
30010502, 3.0313e+001, 0.0000e+000
```

```
30010503, 3.0625e+001, 0.0000e+000
```

```
30010504, 3.0938e+001, 0.0000e+000
```

```
30010505, 3.1250e+001, 0.0000e+000
```

```
30010506, 3.1563e+001, 0.0000e+000
```

```
30010507, 3.1875e+001, 0.0000e+000
```

```
30010508, 3.2188e+001, 0.0000e+000
```

```
30010509, 3.2500e+001, 0.0000e+000
```

```
30010601, 3.0000e+001, -8.3333e-001
```

```
30010602, 3.0313e+001, -8.3333e-001
```

```
30010603, 3.0625e+001, -8.3333e-001
```

```
30010604, 3.0938e+001, -8.3333e-001
```

```
30010605, 3.1250e+001, -8.3333e-001
```

30010606, 3.1563e+001, -8.3333e-001
30010607, 3.1875e+001, -8.3333e-001
30010608, 3.2188e+001, -8.3333e-001
30010609, 3.2500e+001, -8.3333e-001
30010701, 3.0000e+001, -1.6667e+000
30010702, 3.0313e+001, -1.6667e+000
30010703, 3.0625e+001, -1.6667e+000
30010704, 3.0938e+001, -1.6667e+000
30010705, 3.1250e+001, -1.6667e+000
30010706, 3.1563e+001, -1.6667e+000
30010707, 3.1875e+001, -1.6667e+000
30010708, 3.2188e+001, -1.6667e+000
30010709, 3.2500e+001, -1.6667e+000
30010801, 3.0000e+001, -2.5000e+000
30010802, 3.0313e+001, -2.5000e+000
30010803, 3.0625e+001, -2.5000e+000
30010804, 3.0938e+001, -2.5000e+000
30010805, 3.1250e+001, -2.5000e+000
30010806, 3.1563e+001, -2.5000e+000
30010807, 3.1875e+001, -2.5000e+000
30010808, 3.2188e+001, -2.5000e+000
30010809, 3.2500e+001, -2.5000e+000
30010901, 3.0313e+001, -3.1250e+000
30010902, 3.0625e+001, -3.1250e+000
30010903, 3.0938e+001, -3.1250e+000
30010904, 3.1250e+001, -3.1250e+000
30010905, 3.1563e+001, -3.1250e+000
30010906, 3.1875e+001, -3.1250e+000
30010907, 3.2188e+001, -3.1250e+000
30011001, 3.0625e+001, -3.7500e+000
30011002, 3.0938e+001, -3.7500e+000
30011003, 3.1250e+001, -3.7500e+000
30011004, 3.1563e+001, -3.7500e+000
30011005, 3.1875e+001, -3.7500e+000
30011101, 3.0938e+001, -4.3750e+000
30011102, 3.1250e+001, -4.3750e+000
30011103, 3.1563e+001, -4.3750e+000
30011201, 3.1250e+001, -5.0000e+000

*ELEMENT,TYPE=CPE4

80010501, 30010501, 30010601, 30010602, 30010502
80010502, 30010502, 30010602, 30010603, 30010503
80010503, 30010503, 30010603, 30010604, 30010504
80010504, 30010504, 30010604, 30010605, 30010505
80010505, 30010505, 30010605, 30010606, 30010506
80010506, 30010506, 30010606, 30010607, 30010507
80010507, 30010507, 30010607, 30010608, 30010508
80010508, 30010508, 30010608, 30010609, 30010509
80010601, 30010601, 30010701, 30010702, 30010602
80010602, 30010602, 30010702, 30010703, 30010603
80010603, 30010603, 30010703, 30010704, 30010604
80010604, 30010604, 30010704, 30010705, 30010605
80010605, 30010605, 30010705, 30010706, 30010606

80010606, 30010606, 30010706, 30010707, 30010607
80010607, 30010607, 30010707, 30010708, 30010608
80010608, 30010608, 30010708, 30010709, 30010609
80010701, 30010701, 30010801, 30010802, 30010702
80010702, 30010702, 30010802, 30010803, 30010703
80010703, 30010703, 30010803, 30010804, 30010704
80010704, 30010704, 30010804, 30010805, 30010705
80010705, 30010705, 30010805, 30010806, 30010706
80010706, 30010706, 30010806, 30010807, 30010707
80010707, 30010707, 30010807, 30010808, 30010708
80010708, 30010708, 30010808, 30010809, 30010709

*ELEMENT,TYPE=CPE3

80010801, 30010801, 30010901, 30010802
80010808, 30010808, 30010907, 30010809
80010901, 30010901, 30011001, 30010902
80010906, 30010906, 30011005, 30010907
80011001, 30011001, 30011101, 30011002
80011004, 30011004, 30011103, 30011005
80011101, 30011101, 30011201, 30011102
80011102, 30011102, 30011201, 30011103

*ELEMENT,TYPE=CPE4R

80010802, 30010802, 30010901, 30010902, 30010803
80010803, 30010803, 30010902, 30010903, 30010804
80010804, 30010804, 30010903, 30010904, 30010805
80010805, 30010805, 30010904, 30010905, 30010806
80010806, 30010806, 30010905, 30010906, 30010807
80010807, 30010807, 30010906, 30010907, 30010808
80010902, 30010902, 30011001, 30011002, 30010903
80010903, 30010903, 30011002, 30011003, 30010904
80010904, 30010904, 30011003, 30011004, 30010905
80010905, 30010905, 30011004, 30011005, 30010906
80011002, 30011002, 30011101, 30011102, 30011003
80011003, 30011003, 30011102, 30011103, 30011004

*Elset, elset=AlleE

80010501,
80010502,
80010503,
80010504,
80010505,
80010506,
80010507,
80010508,
80010601,
80010602,
80010603,
80010604,
80010605,
80010606,
80010607,
80010608,
80010701,
80010702,

```
80010703,
80010704,
80010705,
80010706,
80010707,
80010708,
80010801,
80010808,
80010901,
80010906,
80011001,
80011004,
80011101,
80011102,
80010802,
80010803,
80010804,
80010805,
80010806,
80010807,
80010902,
80010903,
80010904,
80010905,
80011002,
80011003,
*SOLID SECTION, ELSET=AlleE, MATERIAL=CollagenZ
1.,
*END Instance
** *****
** **   Collagen   **
** **   Zwickel   **
** **   NSETs für BC **
** *****
*NSET,NSET=CZUL001,Instance=CZ001
30010901,
30011001,
30011101,
*NSET,NSET=CZUM001,Instance=CZ001
30010904,
30011003,
30011102,
*NSET,NSET=CZUR001,Instance=CZ001
30010907,
30011005,
30011103,
*NSET,NSET=CZOQ001,Instance=CZ001
30010501,
30010502,
30010503,
30010504,
30010505,
```

```
30010506,
30010507,
30010508,
30010509,
*NSET,NSET=CZUQ001,Instance=CZ001
30010801,
30010802,
30010803,
30010804,
30010805,
30010806,
30010807,
30010808,
30010809,
*NSET,NSET=CZL001,Instance=CZ001
30010601,
30010701,
*NSET,NSET=CZM001,Instance=CZ001
30010505,
30010605,
30010705,
30010805,
*NSET,NSET=CZR001,Instance=CZ001
30010609,
30010709,
*NSET,NSET=CZEckOL001,Instance=CZ001
30010501,
*NSET,NSET=CZEckUL001,Instance=CZ001
30010801,
*NSET,NSET=CZEckOR001,Instance=CZ001
30010509,
*NSET,NSET=CZEckUR001,Instance=CZ001
30010809,
*NSET,NSET=CZEckUM001,Instance=CZ001
30011201,
** #####
** ## Bereich : Mineralplättchen 002 ##
** #####
.....
.....
.....
.....
.....

** #####
** ## Bereich : Mineralplättchen 044 ##
** #####
** *****
** ** MineralPlättchen **
** ** Knoten plus Elemente **
** *****
*Instance, name=MP044, part=Mineralplättchen
```

*NODE

440101, 1.5750e+002, -5.0000e+001
440102, 1.6500e+002, -5.0000e+001
440103, 1.7250e+002, -5.0000e+001
440104, 1.8000e+002, -5.0000e+001
440105, 1.8750e+002, -5.0000e+001
440106, 1.9500e+002, -5.0000e+001
440107, 2.0250e+002, -5.0000e+001
440108, 2.1000e+002, -5.0000e+001
440109, 2.1750e+002, -5.0000e+001
440201, 1.5750e+002, -5.0833e+001
440202, 1.6500e+002, -5.0833e+001
440203, 1.7250e+002, -5.0833e+001
440204, 1.8000e+002, -5.0833e+001
440205, 1.8750e+002, -5.0833e+001
440206, 1.9500e+002, -5.0833e+001
440207, 2.0250e+002, -5.0833e+001
440208, 2.1000e+002, -5.0833e+001
440209, 2.1750e+002, -5.0833e+001
440301, 1.5750e+002, -5.1667e+001
440302, 1.6500e+002, -5.1667e+001
440303, 1.7250e+002, -5.1667e+001
440304, 1.8000e+002, -5.1667e+001
440305, 1.8750e+002, -5.1667e+001
440306, 1.9500e+002, -5.1667e+001
440307, 2.0250e+002, -5.1667e+001
440308, 2.1000e+002, -5.1667e+001
440309, 2.1750e+002, -5.1667e+001
440401, 1.5750e+002, -5.2500e+001
440402, 1.6500e+002, -5.2500e+001
440403, 1.7250e+002, -5.2500e+001
440404, 1.8000e+002, -5.2500e+001
440405, 1.8750e+002, -5.2500e+001
440406, 1.9500e+002, -5.2500e+001
440407, 2.0250e+002, -5.2500e+001
440408, 2.1000e+002, -5.2500e+001
440409, 2.1750e+002, -5.2500e+001

*ELEMENT,TYPE=CPE4R

50440101,	440101,	440201,	440202,	440102
50440102,	440102,	440202,	440203,	440103
50440103,	440103,	440203,	440204,	440104
50440104,	440104,	440204,	440205,	440105
50440105,	440105,	440205,	440206,	440106
50440106,	440106,	440206,	440207,	440107
50440107,	440107,	440207,	440208,	440108
50440108,	440108,	440208,	440209,	440109
50440201,	440201,	440301,	440302,	440202
50440202,	440202,	440302,	440303,	440203
50440203,	440203,	440303,	440304,	440204
50440204,	440204,	440304,	440305,	440205
50440205,	440205,	440305,	440306,	440206
50440206,	440206,	440306,	440307,	440207

50440207, 440207, 440307, 440308, 440208
50440208, 440208, 440308, 440309, 440209
50440301, 440301, 440401, 440402, 440302
50440302, 440302, 440402, 440403, 440303
50440303, 440303, 440403, 440404, 440304
50440304, 440304, 440404, 440405, 440305
50440305, 440305, 440405, 440406, 440306
50440306, 440306, 440406, 440407, 440307
50440307, 440307, 440407, 440408, 440308
50440308, 440308, 440408, 440409, 440309

**** Region:*****Elset, elset=AlleE**

50440101,
50440102,
50440103,
50440104,
50440105,
50440106,
50440107,
50440108,
50440201,
50440202,
50440203,
50440204,
50440205,
50440206,
50440207,
50440208,
50440301,
50440302,
50440303,
50440304,
50440305,
50440306,
50440307,
50440308,

**** Section:*****SOLID SECTION, ELSET=AlleE, MATERIAL=Mineral**

1.,

END Instance*** *********** ** MineralPlättchen ******** ** NSETs für BC ******** **********NSET,NSET=MPOL044,Instance=MP044**

440102,
440103,
440104,

***NSET,NSET=MPUL044,Instance=MP044**

440402,
440403,
440404,

```
*NSET,NSET=MPOR044,Instance=MP044
  440106,
  440107,
  440108,
*NSET,NSET=MPUR044,Instance=MP044
  440406,
  440407,
  440408,
*NSET,NSET=MPR044,Instance=MP044
  440209,
  440309,
*NSET,NSET=MPL044,Instance=MP044
  440201,
  440301,
*NSET,NSET=MPEckOL044,Instance=MP044
  440101,
*NSET,NSET=MPEckUL044,Instance=MP044
  440401,
*NSET,NSET=MPEckOM044,Instance=MP044
  440105,
*NSET,NSET=MPEckUM044,Instance=MP044
  440405,
*NSET,NSET=MPEckOR044,Instance=MP044
  440109,
*NSET,NSET=MPEckUR044,Instance=MP044
  440409,
** #####
*MPC
** #####
** ## MPC-Bereich Mineralplättchen 001 ##
** #####
PIN,MPUR001,CRO001,
PIN,MPR001,CZL001,
PIN,CRL001,CZOR004,
PIN,CRR001,CZUL001,
PIN,MPEckOR001,CZEckOL001,
PIN,CZEckOM004,MPEckUM001,
PIN,CREckOL001,MPEckUM001,
PIN,CZEckUL001,MPEckUR001,
PIN,CREckOR001,MPEckUR001,
** #####
** ## MPC-Bereich Mineralplättchen 002 ##
** #####
.....
.....
.....
.....
.....
** #####
** ## MPC-Bereich Mineralplättchen 044 ##
** #####
PIN,MPOL044,CRU039,
```

```
PIN,MPOR044,CLU040,
PIN,MPL044,CZR043,
PIN,CZEckOR043,MPEckOL044,
PIN,CREckUL039,MPEckOL044,
PIN,CZEckUM039,MPEckOM044,
PIN,CLEckUL040,MPEckOM044,
PIN,CREckUR039,MPEckOM044,
PIN,CLEckUR040,MPEckOR044,
PIN,MPEckUL044,CZEckUR043,
** #####
** ##      INDENTER      ##
** ##      Knoten plus Elemente      ##
** #####
*Instance, name=IndenterTip, part=Indenter
** Region:
*Element, type=MASS, elset=TipElset
1,1
*Mass, elset=TipElset
1000.,
** Radienmittelpunkt: (0,100nm)
*NODE
1,0,100
*NSET, NSET=TipRP
1,
*SURFACE, TYPE=SEGMENTS, name=RigidSurfaceTip
START,470.71067812,429.28932188
LINE,70.71067812,29.28932188
CIRCL,0,0,0,100
*RIGID BODY,ref node=TipRP,analytical surface=RigidSurfaceTip
*END Instance
*ELSET,ELSET=MP001E,Instance=MP001
50010101, 10101, 10201, 10202, 10102
50010102, 10102, 10202, 10203, 10103
50010103, 10103, 10203, 10204, 10104
50010104, 10104, 10204, 10205, 10105
*ELSET,ELSET=CZ001E,Instance=CZ001
80010501, 30010501, 30010601, 30010602, 30010502
80010502, 30010502, 30010602, 30010603, 30010503
80010503, 30010503, 30010603, 30010604, 30010504
80010504, 30010504, 30010604, 30010605, 30010505
80010505, 30010505, 30010605, 30010606, 30010506
80010506, 30010506, 30010606, 30010607, 30010507
80010507, 30010507, 30010607, 30010608, 30010508
80010508, 30010508, 30010608, 30010609, 30010509
*ELSET,ELSET=MP002E,Instance=MP002
50020101, 20101, 20201, 20202, 20102
50020102, 20102, 20202, 20203, 20103
50020103, 20103, 20203, 20204, 20104
50020104, 20104, 20204, 20205, 20105
50020105, 20105, 20205, 20206, 20106
50020106, 20106, 20206, 20207, 20107
50020107, 20107, 20207, 20208, 20108
```

```
50020108, 20108, 20208, 20209, 20109
*ELSET,ELSET=CZ002E,Instance=CZ002
80020501, 30020501, 30020601, 30020602, 30020502
80020502, 30020502, 30020602, 30020603, 30020503
80020503, 30020503, 30020603, 30020604, 30020504
80020504, 30020504, 30020604, 30020605, 30020505
80020505, 30020505, 30020605, 30020606, 30020506
80020506, 30020506, 30020606, 30020607, 30020507
80020507, 30020507, 30020607, 30020608, 30020508
80020508, 30020508, 30020608, 30020609, 30020509
*ELSET,ELSET=MP003E,Instance=MP003
50030101, 30101, 30201, 30202, 30102
50030102, 30102, 30202, 30203, 30103
50030103, 30103, 30203, 30204, 30104
50030104, 30104, 30204, 30205, 30105
50030105, 30105, 30205, 30206, 30106
50030106, 30106, 30206, 30207, 30107
50030107, 30107, 30207, 30208, 30108
50030108, 30108, 30208, 30209, 30109
*ELSET,ELSET=CZ003E,Instance=CZ003
80030501, 30030501, 30030601, 30030602, 30030502
80030502, 30030502, 30030602, 30030603, 30030503
80030503, 30030503, 30030603, 30030604, 30030504
80030504, 30030504, 30030604, 30030605, 30030505
80030505, 30030505, 30030605, 30030606, 30030506
80030506, 30030506, 30030606, 30030607, 30030507
80030507, 30030507, 30030607, 30030608, 30030508
80030508, 30030508, 30030608, 30030609, 30030509
*ELSET,ELSET=MP004E,Instance=MP004
50040101, 40101, 40201, 40202, 40102
50040102, 40102, 40202, 40203, 40103
50040103, 40103, 40203, 40204, 40104
50040104, 40104, 40204, 40205, 40105
50040105, 40105, 40205, 40206, 40106
50040106, 40106, 40206, 40207, 40107
50040107, 40107, 40207, 40208, 40108
50040108, 40108, 40208, 40209, 40109
** #####
** ## GLOBALE ElementSet's ##
** #####
*ELSET,ELSET=OBEN
MP001E,
CZ001E,
MP002E,
CZ002E,
MP003E,
CZ003E,
MP004E,
*ELSET,ELSET=001R,Instance=CR001
AlleE,
*ELSET,ELSET=001Z,Instance=CZ001
AlleE,
```

*ELSET,ELSET=002R,Instance=CR002
AlleE,
*ELSET,ELSET=002L,Instance=CL002
AlleE,
*ELSET,ELSET=002Z,Instance=CZ002
AlleE,
*ELSET,ELSET=003R,Instance=CR003
AlleE,
*ELSET,ELSET=003L,Instance=CL003
AlleE,
*ELSET,ELSET=003Z,Instance=CZ003
AlleE,
*ELSET,ELSET=004R,Instance=CR004
AlleE,
.....
.....
.....
.....
.....
*ELSET,ELSET=042Z,Instance=CZ042
AlleE,
*ELSET,ELSET=043Z,Instance=CZ043
AlleE,
*ELSET,ELSET=SET1
001R,001Z,
002R,002L,002Z,
003R,003L,003Z,
004R,004L,004Z,
005R,005L,005Z,
006R,006L,006Z,
007R,007L,007Z,
008L,
009R,009Z,
010R,010L,010Z,
011R,011L,011Z,
012R,012L,012Z,
013R,013L,013Z,
014R,014L,014Z,
015R,015L,015Z,
016L,
017R,017Z,
018R,018L,018Z,
019R,019L,019Z,
020R,020L,020Z,
021R,021L,021Z,
022R,022L,022Z,
023R,023L,023Z,
024L,
025R,025Z,
026R,026L,026Z,
027R,027L,027Z,
028R,028L,028Z,

029R,029L,029Z,
030R,030L,030Z,
031R,031L,031Z,
032L,
033R,033Z,
034R,034L,034Z,
035R,035L,035Z,
036R,036L,036Z,
037R,037L,037Z,
038R,038L,038Z,
039R,039L,039Z,
040L,
041Z,
042Z,
043Z,
** #####
** ## **GALE NodeSet's** ##
** #####
*NSET,NSET=**SYMMyMP**
AMP01,
AMP02,
AMP03,
AMP04,
AMP05,
AMP06,
*NSET,NSET=**SYMMyC**
AC01,
AC02,
AC03,
AC04,
AC05,
*NSET,NSET=**SYMMy**
SYMMyMP,SYMMyC,
*NSET,NSET=**XSMM**
SYMMy,
*NSET,NSET=**OBEN**
MPOR001,
CZOQ001,
MPOL002,
MPOR002,
CZOQ002,
MPOL003,
MPOR003,
CZOQ003,
MPOL004,
MPOR004,
*NSET,NSET=**UNTEN**
MPEckUM041,
MPUR041,
CZUQ041,
MPUL042,
MPUR042,

CZUQ042,
 MPUL043,
 MPUR043,
 CZUQ043,
 MPUL044,
 MPUR044,

c) ABAQUS/Standard Inputfile

Hier sei beispielhaft eine Inputdatei angeführt. In dieser Datei *Diplomarbeit.inp* hat der Autor um sich das Arbeiten zu erleichtern und Interessierten das Vorgehen zu erläutern zahlreiche Kommentare eingeführt. Diese sind durch ein vorangegangenes (**) und die grüne Farbe gekennzeichnet. Vom Autor definierte Namen sind rot und änderbare Variablen sind blau geschrieben. Der Text sollte somit selbsterklärend sein.

```

=====
** Verfasser: Florian Markus MAYER,                               2004
** Diplomarbeit : "Nanointendierung in eine 'brick and mortar'-Struktur"
**
** ABAQUS-Input-File :                                         Diplomarbeit.inp
=====
** HINWEIS zur Programmausführung :
**
** datacheck-Option
** Programmstart MIT datacheck und MIT Bildschirmausgabe :
**   abaqus job=<Name> datacheck interactive
** => startet eine interactive Überprüfung auf Fehler ohne tatsächliche
**   Analyseschritte! => Fehlerausgabe am Bildschirm und im *.dat-file
** => errechnet auch ungefähren Speicherbedarf, daher sollte immer ein
**   datacheck ausgeführt werden!
** zum Fortsetzen einer solch eingeleiteten Analyse :
**   abaqus job=<Name> continue
** Programmstart OHNE Datacheck aber MIT Bildschirmausgabe :
**   abaqus job=<Name> interactive
** Programmstart OHNE Datacheck und OHNE Bildschirmausgabe :
**   abaqus job=<Name>
**
** Erster Befehl (*Keyword line) muss im ABAQUS-Quellcode "*HEADING" sein
** Danach folgt die Möglichkeit einer Texteingabe, die den Sinn des Programms
** wiedergeben sollte => Kurzbeschreibung (ACHTUNG : NUR in diesem Bereich
** (und nach *STEP) sind auch leere Zeilen und Kommentarzeilen ohne (**)-Kenn-
** zeichnung zulässig!!)
**
*HEADING

SI Units (m, kg, s => N, Pa, J,...)

```

Knoten, Knoten-Sets, Elemente & Element-Sets der Geometrie werden über Datei "KnotenElemente.i" eingelesen. (*INCLUDE)

ABAQUS-Inputfile für die Simulation des Verformungsverhaltens der aus den obigen Dateien eingelesenen Geometrie durch die im Bereich HISTORY DATA definierten Belastungsschritte (STEP)

HIER : Eindrückkraft in neg. y-Richtung (2er) mit einer Maximalgröße von 1kN

right-handed, global rectangular (Cartesian) system

1-Richtung rechtwinkelig zur Symmetrieachse entlang der Kontaktlinie
2-Richtung entlang der Symmetrieachse

```

** Bei großen Verformungen kann ein lokales Koordinatensystem mittels
** *ORIENTATION-Option eingeführt werden. Besonders auch bei Material-
** orientierung in Composite-Materialien nötig.
**
** Ein Stern vor der Zeile (*) bedeutet eine BEFEHLSZEILE (*Keyword lines)
** Zwei Sterne vor der Zeile (**) bedeutet eine KOMMENTARZEILE (**Comment =
** wird übergangen)
** Drei Sterne vor der Zeile (***) bedeutet eine MELDUNG DES PROGRAMMES (z.B. bei
** Fehlern!)
**
** Zur Hilfestellung wurden allgemeine Bemerkungen zum Programmaufbau bzw.
** zur gestellten Aufgabe in "*** Comment" -Zeilen angegeben.
** Die meisten "Option blocks" mit den "*Keyword lines" und den "Data lines" wurden
** bereits in der Datei "KnotenElemente.i" eingebaut.
**
** Knoten-Definition :
**   <KnotenNr>, <x1>, <x2>, <x3>
** Element-Definition :
**   <ElementNr>, <Knoten/Set1>, <Knoten/Set2>, ...
**
** Knotenset-Definition :
**   *NODE,NSET=<KnotenSetName>
**   <KnotenNr1>, ... (mit/ohne Koordinaten laut Knoten-Definition)
**   bzw.
**   <KnotenSetName1>, ...
** oder auch mit GENERATE Bereiche von Knotennummern zu Sets zusammenfassen
**   *NODE,NSET=<KnotenSetName>, GENERATE
**   <KnotenNr von2>, <KnotenNr bis2>, <Inkerement>, ...
** (1,101,2 würde beispielsweise bedeutet von 1 bis 101 mit Schrittweite 2 also
** 1,3,5, ..., 99,101)
**
** Elementset-Definition :
**   *ELEMENT,ELSET=<ElementSetName>,TYPE=<Elementtyp>
**   <ElementNr1>, ... (mit/ohne Koordinaten laut Element-Definition)
**   bzw.
**   <ElementSetName1>, ...
**

```

```

** Duplizieren von Knoten => erzeugen von neuen Knoten, deren Nummer um NN
** erhöht sind und die eine Verschiebung in 1, 2 und 3 erfahren:
**      *NCOPY,OLD SET=<QuellKnotenSetName>,
**      NEW SET=<ZielKnotenSetName>,CHANGE NUMBER=NN,SHIFT
**      <Verschiebung in 1>, <Verschiebung in 2>, <Verschiebung in 3>
**
** Achtung : in ABAQUS sind bis zu 256 Zeichen pro Zeile erlaubt
** => möglicherweise scrollen!!!
** #####
** PARTS
** #####
** Part, name=Mineralplättchen
** End Part
** Part, name=CollagenLinks
** End Part
** Part, name=CollagenRechts
** End Part
** Part, name=CollagenZwickel
** End Part
** Part, name=Indenter
** End Part
**
** RESTART, WRITE, FREQUENCY=10
** ABAQUS wird angewiesen das restart-File während der Analyse anzulegen
** Sollte ein vorheriger Versuch aus irgendwelchen Gründen abgebrochen worden sein, kann
** es hilfreich sein, nicht mehr die gesamte Analyse durchzuführen sondern die bisherigen
** Ergebnisse einzulesen *RESTART,READ und dann fortzusetzen (ACHTUNG, die
** Dateien können SEHR groß werden!!!
** Frequency=10 schreibt bei Increment 10,20,30,40,...
** Frequency=1 schreibt bei allen
** Frequency=0 unterdrückt (unterbricht) die Ausgabe. Sobald *RESTART angegeben wird
** werden auch die Ergebnisse des letztem Increment geschrieben.
**
** PREPRINT, ECHO=NO, MODEL=YES, HISTORY=YES
** PRE = vor, vorab => zuvor ausdrucken (z.B. Zeilennummern,...)
** ECHO erzeugt Zeilen-Zeichenlineale im *.dat-file (Voreinstellung : NO)
** MODEL gibt ALLE erzeugten Daten des MODEL-DATA-Bereiches im *.dat-file an,
** HISTORY gibt ALLE erzeugten Daten des HISTORY-DATA-Bereiches im *.dat-file
** an, da dies sehr umfangreich sein kann ist Voreinstellung BEIDER auf NO
**
** *****
**  M O D E L  D A T A
** *****
** -----
** #####
** GEOMETRIE - ASSEMBLY&INSTANCES
** #####
** Einlesen der Geometriedaten (Knotenkoordinaten und -Sets sowie Element-Sets)
** Richtlinien zur Erstellung der Datei : "KnotenElemente.i"
**
** MINERALPLÄTTCHEN : 4-node solid: CPS4
** Continuumelement, 2-dimensionaler 4-Knoter

```

```

** bilinear, plain stress elements
** ebener Spannungszustand (PS)=>S33=0
** => nur DOF 1-2 (ux, uy) in jedem Knoten
** oder auch CPE4
**
** COLLAGEN-RAUTEN : 4-node solid: CPS4
** Continuumelement, 2-dimensionaler 4-Knoter
** bilinear, plain stress elements
** ebener Spannungszustand (PS)=>S33=0
** => nur DOF 1-2 (ux, uy) in jedem Knoten
**
** COLLAGENZWICKEL :
** Vierecke : 4-node solid: CPS4
** Continuumelement, 2-dimensionaler 4-Knoter
** bilinear, plain stress elements
** ebener Spannungszustand (PS)=>S33=0
** => nur DOF 1-2 (ux, uy) in jedem Knoten
** Dreiecke : 3-node solid: CPS3
** Continuumelement, 2-dimensionaler 3-Knoter
** linear, plain stress elements
** ebener Spannungszustand (PS)=>S33=0
** => nur DOF 1-2 (ux, uy) in jedem Knoten
**
** Mehrpunktabhängigkeiten (MultiPointConstrains) ermöglicht es, Knotenkopplungen ganzer
** Sets durchzuführen selbst wenn diese unterschiedlichen DOF besitzen
** *MPC
** <MPC-Typ>, <Knoten/Set1>, <Knoten/Set2>, ...
** als MPC-Typ steht beispielsweise der Befehl PIN (nur Translationen werden übertragen)
** oder TIE (auch Momente werden übergeben) zur Verfügung
** Verschiedenste Knoten können mittels UNSORTED in einem Set zusammengefasst
** werden und mit einem weiteren UNSORTED Set mittels MPC verknüpft werden
** Weitere „Abhängigkeits-Definitionen“ zwischen bestimmten Knoten wären die
** *EQUATION-Option und die Definition der Randbedingungen *BOUNDARY
** *EQUATION erstellt die gleiche DOF-Beschränkung für mehrere Knoten/Sets
**  $A_1u_1 + A_2u_2 + \dots + A_nu_n = 0$  mit Koeffizienten  $A_i$  der Verschiebungen  $u_i$  und mit
** Nummer der Ausdrücke (number of terms) n
** *EQUATION
** <n>
** <KnotenNr1>, <DOF1>, <A1>, <KnotenNr2>, <DOF2>, <A2>, ... <DOFn>, <An>
** z.B. die Verschiebung in 3-Richtung im Knoten 10 ist doppelt so groß, wie
** die Verschiebung des Knoten 501 in 1-Richtung =>  $2u_1(501) = u_3(10)$  =>
**  $2u_3(501) - u_1(10) = 0$ 
** *EQUATION
** 2
** 501,3,2,10,1,-1
** ** ACHTUNG!!
** Knoten, welche bereits einmal als ERSTER Knoten in einer *MPC oder *EQUATION-
** Option definiert wurden, DÜRFEN NICHT erneut in einer ANDEREN solchen Option
** verwendet werden, da bei diesen Befehlen die Freiheitsgrade des Knotens aus der
** Elementmatrix gelöscht werden und daher bei einem erneuten Befehlsaufruf nicht mehr
** zur Verfügung stehen. => Zirkelbezug
** #####

```

```

** ASSEMBLY
** #####
*Assembly, name=Assembly
** #####
** INSTANCES
** #####
** =====> Erfolgt in Datei "KnotenElemente.i"
*INCLUDE,INPUT=KnotenElemente.i
** #####
** NSETs & ELSETs
** #####
** =====> Erfolgt grÖßtenteils in Datei "KnotenElemente.i"
*NSET, NSET=RPu, Instance=MP001
00010101,
** der hier definierte Referenzpunkt zwwyyxx ist der tiefste Punkt des Indenter (unten)
** und entspricht einem Mineralplättchenknoten (z=0) Und zwar ist es der erste Knoten in x-
** Richtung (xx=01) sowie y-Richtung (yy=01) des Mineralplättchens 1 (www=1)
*NSET, NSET=TipRP, Instance=IndenterTip
1,
** dieser Knoten 1 ist in der Datei KnotenElemente.i definiert als Indenterradiusmittelpunkt
*NSET, NSET=Fuehrung, Instance=IndenterTip
452,
453,
** diese Knoten repräsentieren zwei in der Datei KnotenElemente.i definierte Indenterpunkte
** welche sich während der Indentation parallel zur Symmetrieebene bewegen
** #####
** SURFACES
** #####
** MASTERSURFACE
** Definiert durch Elemente (ELSET : TipE) und Seite der Elemente, wobei
** SNEG die Seite der Elemente angibt, welche die Mastersurface bilden
*SURFACE, TYPE=ELEMENT, NAME=IndenterSurf
TipE, SNEG
** SLAVESURFACE
** Definiert durch Elemente oder hier Seite der Elemente (NODESET : OBEN)
** 1 gibt die "Dicke" der Slavesurface an!!
*SURFACE, TYPE=NODE, NAME=MP_C_Surf
OBEN, 1.
** Definiert die MASTERSURFACE als Starrkörper (keine Verformung) und dies mit Hilfe
** des Referenzpunktes (NODESET : TIPRP) des Instances "IndenterTip" und
** der Elementdefinition (ELSET : IndenterTip) des Instances "IndenterTip"
*Rigid Body, ref node=IndenterTip.TipRP, elset=IndenterTip.IndenterTip
** #####
** ABSCHNITTSZUORDNUNG
** #####
** Zuordnen von Eigenschaften wie Material, Querschnitt,... an die Elemente
** Für 3D- und AX- Elemente sind keine weiteren Geometrieangaben erforderlich,
** bei PE- und PS- Elementen muss die Dicke angegeben werden (data line)!!
** *SOLID SECTION, ELSET=MPG, MATERIAL=Mineral
** 1.,
** =====> Erfolgt in Datei "KnotenElemente.i"
*End Assembly

```

```

** -----
** #####
** MATERIALDATEN
** #####
** -----
** MINERAL
** -----
** rein elastisches Materialverhalten (E-Modul : 114GPa = 1.14e+11Pa)
** MATERIAL, NAME=Mineral
** ELASTIC
  1.14e+11, 0.3
** E-Modul, Poissonzahl (angenommen), Temperatur(optional),...
** -----
** COLLAGEN-LinksRechts
** -----
** rein elastisches Materialverhalten (E-Modul : 1.5GPa = 1.5e+09Pa)
** linearer Spannungsanstieg bis 3% Dehnung, danach rein plastisches
** Verhalten des Collagens
** MATERIAL, NAME=CollagenLR
** ELASTIC
  1.5e+09, 0.3
** E-Modul, Poissonzahl (angenommen)
** PLASTIC
  4.5e+07, 0.0
** Spannung, plastische Dehnung
** daraus weis ABAQUS, dass sich der Übergang bei (3% = 0.03) befindet
** -----
** COLLAGEN-Zwickel
** -----
** rein elastisches Materialverhalten (E-Modul : 1.5GPa = 1.5e+09Pa)
** linearer Spannungsanstieg bis 3% Dehnung, danach rein plastisches
** Verhalten des Collagens
** MATERIAL, NAME=CollagenZ
** ELASTIC
  1.5e+09, 0.3
** E-Modul, Poissonzahl (angenommen)
** PLASTIC
  4.5e+07, 0.0
** Spannung, Dehnung (3% = 0.03)
** -----
** #####
** KRAFTVERLAUF
** #####
** Möglichkeit, einen Kraft-Zeit-Verlauf (Zeit[s/100],%/100) anzugeben!
** *AMPLITUDE, NAME=INTEND, VALUE=RELATIVE
** 0.0,0.0, 0.1,1.0, 1.1,1.0, 1.2,0.0
** Zeitpunkt1, Amplitude1, Zeitpunkt2, Amplitude2,...
** RELATIVE normiert die Amplitude auf die Maximallast
** Wenn dynamische Belastung (*dynamic) ist nun REELLE, TATSÄCHLICHE Zeit und
** -inkrement gefragt
** z.B. : *DYNAMIC, ALPHA=0
**      dt, Δt

```

```

** Gesamtanalysezeit z.B.  $\Delta t = 2s$  (Belastungszeit laut *AMPLITUDE Anweisung 1.2s,
** danach ** Kraft = 0)) und Inkrement-Schrittweite z.B.  $dt = 0.01 s$  ergeben dann 200
** Inkremente => Abstimmung mit *FREQUENCY-Anweisung => FREQUENCY=5 ergäbe
** dann eine Ausgabe von 40 Ergebnissen.
** ALPHA ist der numerische Dämpfungsfaktor (hier 0 = keine Dämpfung, daher wäre  $\Delta t =$ 
** 1.2s empfehlenswert)
** -----
** #####
** INTERACTION PROPERTIES
** #####
** Reibungsbedingungen bei Oberflächenkontakt
** hier "surface to surface" Angaben mit KEINER Reibung zwischen den Kontaktflächen
**SURFACE INTERACTION,NAME=NoFRIC
1.,
*FRICTION
0.,
** #####
** INTERACTIONS
** #####
** Interaction: Kontakt
** In Kontakt stehen die im Punkt SURFACES definierten Oberflächen
** und zwar mit dem im Punkt INTERACTION PROPERTIES definierten
** Kontaktbedingungen!!
**CONTACT PAIR, INTERACTION=NoFric
MP_C_Surf, IndenterSurf
** -----
** #####
** BOUNDARY CONDITIONS
** #####
** Name: UNTEN Type: Displacement/Rotation ( $u_y = 2$ )
** entspricht Materialauflage und verhindert Absinken (entspricht Widerlager!)
**BOUNDARY
UNTEN, 2, 2
** <NSET>, <DOF von>, <DOF bis>
** Name: Fuehrung Type: Displacement/Rotation ( $u_x = 1$ )
** führt den Indenter und verhindert dessen Verdrehen
**BOUNDARY
Fuehrung, 1, 1
** Name: Symmetry Type: Symmetry/Antisymmetry/Encastre ( $u_x = r_z = 0$ )
** Materialfixierung definiert Symmetrie um y-Achse, da ja nur halbes
** Kontinuum (eine Seite) simuliert!
**BOUNDARY
SYMy, XSYMM
** -----
*****
** HISTORY DATA
*****
** Mit *STEP beginnt erste Befehlszeile (*Keyword line) des History-Data-Blocks
** PERTURBATION besagt, dass es sich um eine LINEARE Analyse handelt
** Wenn es sich um ein statisches Problem handelt gibt es keine data-lines nach der
** *STATIC-Option
** -----

```

```

** *****
** STEP 1 : KONTAKT
** *****
*STEP, NAME=KONTAKT

```

Stellt Kontakt zwischen Indenter und Werkstoff sicher!!

```

** wie im *HEADING-Block hier Möglichkeit einer Texteingabe, z.B. der Belastungs-
** schrittbeschreibung
** statische Analyse – Langzeitbetrachtung unter konstanter Belastung
** Möglichkeit der Inkrementanzahleinschränkung, wenn man nicht weis, wie viele die
** Simulation benötigt => INC=<max.Anzahl>
** Einbau einer generellen, nichtlinearen Simulation (z.B. bei Material-Nichtlinearität =>
** Plastizität ab Fließgrenze,..)=> Entfernen von PERTURBATION
** Statisch linear => Zeit spielt überhaupt keine Rolle
** Dynamisch => Zeit spielt eine Rolle
** Statisch mit Nichtlinearität => "Zeit" spielt eine Rolle => Vorgabe des ersten
** Belastungsschrittes, ab wann die Analyse begonnen werden soll! HIER : 20% der
** Gesamtzeit (1.0=100% => Belastungsschritt abgeschlossen => KEINE TATSÄCHLICHE
** (reale) Zeit!!))
** Gesucht wird natürlich das Ergebnis bei 100% (dieses kann möglicherweise nicht,
** innerhalb der vorgegebenen Incrementanzahl ermittelt werden)
** STEP, INC=25
** *STATIC
** 0.2,1.0
** Würde geometrische Nichtlinearität hier eine Rolle spielen (z.B. große Verschiebungen)
** würde in der STEP-Zeile mit der NLGEOM-Option darauf hingewiesen werden!!
** *STATIC da KEINE Nichtlinearitäten => sonst *DYNAMIC
** -----
** Belastungstyp
** -----
*STATIC
1., 1., 1e-05, 1.
** Initialisierungszeitincrement, STEP-Zeit, minimal, maximal erlaubtes Zeitincrement
** -----
** Randbedingungen & Belastung
** -----
** Definition : welche DOF sind in welchen Knoten gesperrt bzw. beschränkt?
** Randbedingungen – degrees of freedom (DOF)
** bei kartesischen Koordinaten
** 1-3 Translation in 1(x),2(y),3(z)
** 4-6 Rotation um 1(x),2(y),3(z)
** ...elektr.Potentiale (9), Temperatur (10),...
** bzw. bei achsensymmetrischen Elementen (Zylinderkoordinaten)
** 1 Transformation in r-Richtung
** 2 Transformation in z-Richtung
** 6 Rotation in der r-z-Ebene
**
** Angabe
** <KnotenNr>, <erster DOF>, <letzter DOF>, <maximale Verschiebung>
** oder
** <KnotenNr>, <DOF1>, <DOF2>,... (voreingestellte Verschiebung = 0.0)

```

```
** oder
** <KnotenNr/Set>, <vordefinierte Randbedingung> (ENCASTRE, PINNED, TIED...)
**
** *TRANSFORM-Option zur Erzeugung eines lokalen Koordinatensystems für
** Randbedingungs-Anwendung möglich
** -----
** Ausgabe
** -----
** Ausgabe - *OUTEPUT => *.odb
** Ausgabe vordefinierter OUTPUT-variables (U Verschiebungen, U2 Versch. in 2,
** RF Reaktionskräfte, S Spannungen,...) in das *.dat-file (datacheck) durch
** PRINT-Option, in das *.odb-file (database for postprocessing) durch
** OUTPUT-Option an den Knoten bzw. in den Elementen.
** OUTPUT,<Name>,HISTORY,FREQUENCY=<Wert>,VARIABLE=ALL,
** FIELD,OP=NEW
** *ELEMENT OUTPUT oder *NODE OUTPUT
** den Zusatz HISTORY würde man benötigen, wenn man im *.odb-File History Daten
** auszugeben wünscht, sollen die für solche Analysen "üblichen" Variablen ausgegeben
** werden, verwendet man die voreingestellte Option VARIABLE=PRESELECT anstelle
** von ALL
** der Zusatz FIELD definiert, dass im Output-File die Variablen eine Felddefinition
** aufweisen und OP=NEW gibt an, dass alle bisherigen Belastungsschritte im Output-File
** vernachlässigt werden können, da es sich um eine neue Belastung handelt ( => vgl.
** MultipleStepAnalysis) im Gegensatz zu wäre z.B. OP=ADD.
** Gleiches gilt auch für die *PRINT Anweisung!
** *NODE PRINT, NSET=<Name>, TOTAL=YES, SUMMARY=NO
** *ELEMENT OUTPUT,NSET=<Name>, TOTAL=YES, SUMMARY=NO
** Ohne <Name> Ausgabe ALLER Knotenergebnisse, TOTAL=YES summiert die
** jeweiligen Ergebnisse bei einem Knoten (z.B. alle RF, damit einfache Überprüfung der
** Ergebnisse => Newton'sches Axiom = Summe alle Kräfte = 0) und gibt dies aus,
** SUMMARY=NO verhindert Ausgabe des höchsten und niedrigsten Wertes der Tabelle
** ALLGEMEINE BEFEHLSSYNTAX :
** *PRINT,<◇>,<◇>,... ; *OUTPUT,<◇>,<◇>,...
** *NODE PRINT,<◇>,... ; *EL OUTPUT,<◇>,...
** Welche Variablen (U,A,V,.....) wovon (Knoten-Elementen) werden in welche Datei
** ausgegeben
** *PRINT, RESIDUAL=NO,FREQUENCY=5
** Voreinstellung ist RESIDUAL=YES, was bedeutet, dass die Gleichgewichtsbedingungen
** ausgegeben werden, HIER wird dies unterdrückt und durch FREQUENCY-Option nur
** jeder 5te Wert ausgegeben (in die *.dat-Datei) =>
** bei DYNAMISCHEN (bzw. nichtlinearen) Analysen kann angegeben werden, wie viele
** Werte an das *.dat- (PRINT-Anweisung) bzw. *.odb-File (OUTPUT-Anweisung)
** ausgegeben werden, da diese Dateien ansonst, durch alle Iterationsergebnisse über-
** proportional wachsen (>GB möglich) => 0 bedeutet KEINE Increment-Ergebnisse
** (voreingestellt ist =1, also ALLE) (wenn nur das Endergebnis ausgegeben werden soll
** (nur plastische Verformung bei multi-Step-Analyse), muss man diesen Wert sehr groß
** wählen (z.B. 9999)
** *NODE PRINT, FREQUENCY=5
** U, RF, V3, A3
** HIER : auch Ausgabe von Geschwindigkeit(V) und Beschleunigung(A) in 3-Richtung
** *EL FILE, FREQUENCY=25
** würde nachfolgende Variablen <Var1>,<Var2>,... in eine Ergebnisdatei *.fil
```

```

** ABAQUS/Standard, *.sel ABAQUS/Explicit) ausgeben
** <Var1>,<Var2>,...
** gleiches würde für *NODE FILE gelten
** Ausgabe - *PRINT => *.dat
** -----
** Ausgabe
** -----
** ++++++
** OUTPUT REQUESTS
** ++++++
** *RESTART, WRITE, FREQUENCY=1
** Auf Ausgabe wird hier verzichtet, da ja im MODEL-DATA angegeben!(Größe!!)
** ++++++
** FIELD OUTPUT
** ++++++
*OUTPUT, FIELD, VARIABLE=PRESELECT
** ++++++
** HISTORY OUTPUT
** ++++++
*OUTPUT, HISTORY, VARIABLE=PRESELECT
*EL PRINT, FREQ=999999
*NODE PRINT, FREQ=999999
*END STEP
** Ende des STEP (Belastungsschrittes)
** -----
** *****
** STEP 2 : BELASTUNG
** *****
*STEP, NAME=BELASTUNG, NLGEOM, INC=3000

```

Eindringen des Indenters in den Werkstoff kraftgesteuert mit Hilfe einer Einzelkraft (CLOAD) am Indenter in Höhe von $1e+8$ (100MN) senkrecht von OBEN (negativ) oder weggesteuert durch Vorgabe einer Eindringtiefe in negativer y-Richtung

```

** -----
** Belastungstyp
** -----
*STATIC
1e-11, 2., 1e-40, 2.
** -----
** Randbedingungen & Belastung
** -----
** Name: Load Type: Concentrated force
** Punktlast mit Betrag -1e+8 in Richtung 2 im Referenzpunkt "TipRP"
*CLOAD
TipRP, 2, -1.0e+8
** <ElementNr/Set>, <DOF>, <Betrag>
** Punktlast durch Befehl *CLOAD, (Druck durch *DLOAD,... mit möglicher Angabe der
** Fläche <load ID> auf welche der Druck ausgeübt wird => P6 (auf Fläche 6))
** Würde man die Punktlast mit Belastungsverlauf laut *AMPLITUDE-Option definiert
** ergäbe sich
** *CLOAD, AMPLITUDE=STOSS

```

```

** <KnotenNr>, <DOF>, <Betrag>
** TipRP,2,-1.0e+8
** -----
** Ausgabe
** -----
** ++++++
** OUTPUT REQUESTS
** ++++++
** ++++++
** FIELD OUTPUT
** ++++++
*OUTPUT, FIELD, VARIABLE=PRESELECT, FREQUENCY=2
** ++++++
** HISTORY OUTPUT
** ++++++
*OUTPUT, HISTORY, VARIABLE=PRESELECT, FREQUENCY=2
*END STEP
** -----
** *****
** STEP 3 : ENTLASTUNG
** *****

```

Elastische Rückverformung durch Wegnahme der Last

```

*STEP, NAME=ENTLASTUNG, NLGEOM, INC=2000
** -----
** Belastungstyp
** -----
*Static
0.01, 2., 1e-20, 2.
** -----
** Randbedingungen & Belastung
** -----
** Name: Load  Type: Concentrated force
** Punktlast OHNE Betrag in Richtung -2 (=y) im Referenzpunkt "TipRP"
** wobei der bisherige Kraftverlauf UNBERÜCKSICHTIGT bleibt (OP=NEW)
*CLOAD, OP=NEW
TipRP, 2, 0
** -----
** Ausgabe
** -----
** ++++++
** OUTPUT REQUESTS
** ++++++
** ++++++
** FIELD OUTPUT
** ++++++
*OUTPUT, FIELD, VARIABLE=PRESELECT, FREQUENCY=2
** ++++++
** HISTORY OUTPUT
** ++++++
*OUTPUT, HISTORY, VARIABLE=PRESELECT, FREQUENCY=2

```

```
*END STEP
** dieser Befehl ist IMMER die letzte Befehlszeile
** -----
** *****
** ALLGEMEINES zur Programmierung (Theoretischer Hintergrund)
**
** CONTINUUM-Elements
**
** Spannungs-Verformungs-Kopplung mittels
** Lagrang'schen oder Material-Verhaltens
** => Volumenskonstanz; Material bleibt während Verformung dem jeweiligen
** Element zugeordnet; kein Materialfluss zwischen Elementen
** Euler'schen oder räumlichen Verhaltens
** => Element ist im Raum fixiert, das Material fließt durch (Strömungslehre,CFD)
**
** Integration mittels Gauß Quadratur(eindimensionale Integration)
** Volle oder reduzierte Integration (in C – elementen)
**
** Voll bezieht sich auf die Anzahl der Gauß-Stützstellen die für die exakte
** Integration der Polynomial-Ausdrücke in der Steifigkeitsmatrix eines
** Elementes mit regelmäßiger Form benötigt werden.
** Voll integrierte, lineare Elemente haben 2 Stützstellen in jeder Richtung,
** voll integrierte, quadratische Elemente nützen 3 Stützstellen je Richtung.
** Problem bei voller Integration von linearen (1-ster Ordnung) Solid-Elementen
** unter Biegebelastung => SHEAR LOCKING (gleichgültig, wie viele Elemente über
** die Dicke des Biegeträgers, Netzfeinung hat KEINEN Effekt mehr!)
** Ergebnis: Träger erscheint zu steif bei Biegung, da die Kanten nicht fähig
** sind, sich zu biegen!
** Abhilfe : bei Biegung nur quadratische Elemente verwenden
** incompatible mode elemente (optimierter Verformungsgradient)
** => können Rechenaufwand reduzieren wenn passend eingesetzt
**
** Reduzierte Integration funktioniert nur bei Vierecken und Sechseckern.
** Man nutzt dabei eine Stützstelle weniger als bei der vollen Integration.
** Reduziert integrierte, lineare Elemente haben nur 1 Stützstelle (im Zentrum
** des Elementes)
** Problem bei reduzierter Integration von linearen (1-ster Ordnung) Solid-
** Elementen => HOURGLASSING (Null-Energie-Verformung, da sich die Linien durch
** die Stützstellen nicht ändern => kein Widerstand gegen eine solche Verformung
** quadratische Elementen zeigen bei reduzierter Integration auch hourglassing,
** jedoch in vernachlässigbarer Größe
** Ergebnis: Träger erscheint zu flexibel wenn Netz zu grob
** Abhilfe: Netzfeinung (mindestens 4 Elemente in der Dicke!)
**
** Allgemein : Netzverzerrungen sollen so gering wie möglich gehalten werden!
** Bei komplexen Geometrien (Aufgaben) mit ungewissen Ergebnissen sollte mit
** reduzierten, quadratischen Elementen gearbeitet werden, da deren Netz
** unempfindlicher gegen Verzerrungen ist und daher bessere Ergebnisse zu
** erwarten sind.
**
** Netz-Konvergenz : liegt vor, wenn eine Netzfeinung nur mehr eine
** vernachlässigbare Verbesserung bringt.(Netzfeinung ist sinnvoll, um Fehler
```

```
** durch stark schwankende Ergebnisse aufzudecken!)
**
**
** BEAM-Elements
**
** allgemein erlauben BEAM-Elemente axiale, Biege- und Torsionsdeformationen
** => Timoshenko-Beams (berücksichtigt auch transversale Scher-Deformation)
** Scherdeformationen sind wichtig bei dicken Trägern, unwesentlich bei
** schlanken Trägern (Timoshenko simuliert beide Arten)
** lineare und quadratische (Timoshenko) Beams bieten die Möglichkeit der
** Querschnittsänderung während der Belastung => Nichtlinearität
** während kubische Beams (Euler-Bernoulli) keine Scherflexibilität ausweisen
** => beste Eignung für schlanke Träger
** lineare Träger für Kontakt-Probleme
** Hybrid-Elemente für sehr flexible oder starre Strukturen bei geometrischer
** Nichtlinearität
** dynamische Vibrationen mit kubischen Trägern simulieren
** wenn allerdings auch transversale Deformation eine Rolle spielt =>
** quadratische Elemente
** dünnwandige Querschnittsflächen sollten mit Spezialtypen simuliert werden =>
** open section warping => B31OS
**
** Nicht-Linearitäten
**
** kein linearer Bezug zwischen Belastung und Verformung, gegeben durch
** Geometrie (Blechausbäulung „snap through“,...)
** Material (Orientierungen, Composit,...)
** Randbedingungen (freier Weg bis Auflager,...)
**
** Hier wird davon ausgegangen, dass, trotz Kompositwerkstoff, keinerlei Nichtlinearitäten
** vorliegen!!!
**
** Hier wird von IDEAL ELASTISCHEM MATERIALVERHALTEN
** (&SYSTEMVERHALTEN) ausgegangen
** Wir gehen von GROSZEN Verschiebungen aus!
**
** ITERATIONEN, INCREMENTE, ...
**
** ABAQUS versucht sich einem korrekten Ergebnis zu nähern (Newton-Raphson)(z.B.
** Verformung bei angelegter Kraft, nichtlinear => Versuch der Kraftaufbringung in
** Teilschritten (Incrementen))
** Diese (den ersten kann man auch vorgeben) wählt ABAQUS selbst, erhöht sie, wenn das
** Programm merkt, dass das dann erzielte Ergebnis ebenfalls konvergiert und verringert
** es mit einem "Rückwertsschritt" (nimmt Teilbelastung zurück => cut-back), wenn dem
** nicht so ist.
```