



Bachelorarbeit

Optimierung einer Python-Routine zur automatischen Berechnung und Zuweisung von CT-basierten Materialeigenschaften auf FE-Netze

Themenstellende Hochschullehrerin: Prof. Dr. Beate Langer

Wissenschaftlicher Betreuer: M. Eng. Konrad Mehle

Betrieblicher Betreuer: Anantha Narayanan Ramakrishnan
Fraunhofer IMWS Halle

vorgelegt von: Immanuel Ries

Studiengang: BMMP

Matrikelnummer: 24966

Inhaltsverzeichnis

Abkürzungsverzeichnis	II
Abbildungsverzeichnis	III
Tabellenverzeichnis	VI
1. Einleitung	1
1.1 Problemstellung	2
1.2 Zielstellung	3
1.3 Programmablauf.....	4
2. Grundlagen CT und Materialkorrelationen	5
2.1 Computertomographie	5
2.2 DICOM-Standard.....	8
2.3 HU-Korrelationen	12
3. Material-Mapping.....	15
3.1 Rekonstruktion der Geometrie	15
3.2 ANSYS CDB-Datei	20
3.3 Mapping-Algorithmen	22
3.3.1 V1	22
3.3.2 V2	25
3.3.3 V3	31
3.3.4 Vergleich der Varianten.....	31
3.4 Programmaufbau und -ablauf	33
4. Änderungen und Benutzeroberfläche	37
5. Validierung anhand eines Beispiels	44
6. Ergebnisse und Diskussion.....	47
7. Zusammenfassung und Ausblick	56
8. Literaturverzeichnis.....	58
9. Anhang	62

Abkürzungsverzeichnis

BMD.....	body mineral density
CAD.....	computer aided design
CT.....	Computertomographie
FE.....	Finite Elemente
FEM.....	Finite Elemente Methode
IO.....	Image Orientation
IP.....	Image Position
K ₂ HP04.....	Dikaliumhydrogenphosphat
MRT.....	Magnetresonanztomographie
PS.....	Pixel Spacing
ROI.....	region of interest
S.....	Software
ST.....	Slice Thickness
VR.....	Value Representation
VM.....	Value Multiplicity

Abbildungsverzeichnis

Abbildung 1: gesunder Knochen (links) und osteoporotischer Knochen (mittig und rechts) [2]	2
Abbildung 2: Datenfluss der benötigten Dateien für die Mapping-Routine	4
Abbildung 3: Aufbau eines CT-Geräts mit den wichtigsten Bestandteilen [4].....	5
Abbildung 4: links: Schnittbild in X-Y bzw. transversaler Ebene mit 2 Röhre-Detektor-Einheiten; rechts: unterschiedliche Aufspannwinkel und Kollimierung des Röntgenstrahls in Z-X bzw. sagittaler Ebene [5]	6
Abbildung 5: Eingangspotonenzahl N_0 des Röntgenstrahls, der auf einen Absorber mit hoher (oben) und niedriger (unten) Dichte trifft und dabei unterschiedlich stark absorbiert. Durch die unterschiedliche Dichte durchdringen unterschiedlich viele Photonen N den Absorber	7
Abbildung 6: Hounsfield Skala mit gefensterem Bereich und typischen CT-Werten für unterschiedliche Gewebetypen [7]	8
Abbildung 7: Aufbau DICOM.....	9
Abbildung 8: Aufbau eines DICOM-Bildes mit Modulen und Attributen [9].....	10
Abbildung 9: DICOM Standard Browser mit geöffneten Informationen einer CT-Serie mit geöffnetem Modul „Patient“ und dessen Notwendigkeit „M“ und einigen zugehörigen Attributen mit Tag, Name und VR und Type	11
Abbildung 10: genauere Beschreibung des Attributs „Patient Sex" mit zugehörigem Tag, Type, VM, VR und Beispiel Inhalt (Attribut Value) [10]	11
Abbildung 11: durch Pydicom erstellte Liste der Attribute und deren Tags, Name, VR, und Value eines Bildes (Liste unvollständig).....	11
Abbildung 12: Selektiervorgang in Amira mit einzeltem Slice mit selektiertem rotem Bereich(links) und gefensterter 3D-Darstellung aller Bilder mit gesamt selektiertem Bereich (violett) und aktuell markiertem Bereich (rot)	16
Abbildung 13: Voxel-Rendering (links) und fehlerhaftes STL (rechts).....	16
Abbildung 14: Detailverlust gefährdete Regionen des Darmbeins mit transparenter alter Oberfläche und braune Oberfläche des geglätteten Körpers mit starker Glättung und maximaler Abweichung von 1,5mm (links) und Glättung mit mehr Detailerhalt und maximaler Abweichung von 0,4mm (rechts)	17
Abbildung 15: erzeugte Flächenkörper auf Basis der STL Dateien mit schwacher Annäherung und starker Glättung (links) und umgekehrt (rechts)	18

Abbildung 16: Leitlinien des Volumenkörper mit unterschiedlich großen Teilflächen	18
Abbildung 17: Vernetzung des linken Hüftbeins mit 10mm (links) und 5mm (rechts)	19
Abbildung 18: Ausschnitt Startblock der CDB-Datei	20
Abbildung 19: Ausschnitt Knotenblock mit Header	21
Abbildung 20: Ausschnitt Elementblock mit Header	21
Abbildung 21: Ausschnitt Materialblock mit zwei Materialien	22
Abbildung 22: Zusammensetzung der Einzelflächen, wenn sich der zusätzliche Punkt o im Gebiet befindet	23
Abbildung 23: Zusammensetzung der Einzelflächen, wenn sich der Punkt außerhalb des Gebiets befindet	24
Abbildung 24: Tetraeder mit Punkt o , mit dem die gerichteten Volumen der Teiltetraeder gebildet werden	24
Abbildung 25: allgemeiner Tetraeder im x, y, z -Raum (links) und der Einheitstetraeder im ξ, η, ζ – bzw. r, s, t , -Raum (rechts) [32]	26
Abbildung 26: Tetraeder mit originalen Eckpunkten u, v, w, t und den skalierten Eckpunkten u, v, w, t , welche den Integrationspunkten entsprechen	28
Abbildung 27: Knotenpunkt w im CT-Gitter (gestrichelte Linien) mit den 8 umgebenden Voxelmittelpunkten C1 bis C8	28
Abbildung 28: trilineare Interpolation in der „Voxelbox“, um ein HU-Wert für den Eckknoten w zu berechnen	29
Abbildung 29: FE-Element als globaler Tetraeder (links) mit Integrationsstellen (blau) und lokaler Referenztetraeder (rechts) mit Integrationspunkten und kleiner skaliertem Tetraeder (blau)	30
Abbildung 30: lineare und quadratische Ansatzfunktionen in 1D und die zugehörige Lösungsnäherung [25]	32
Abbildung 31: Aufbau und Verweisung der Python-Dateien des Programms	33
Abbildung 32: beispielhafte Veränderung der Materialeinträge vom individuellen Elementwert (Modul_integ) zum gruppierten Wert (Modul_grupp) mit dem Bereichsparameter gap	36
Abbildung 33: Import der gemappten CDB-Datei in ANSYS	36
Abbildung 34: Fehlermeldung beim Importieren der gemappten Datei des umgeänderten Programms	37
Abbildung 35: Funktion, welche den Header des Knotenblocks identifiziert und zurückgibt	37
Abbildung 36: falsche Zuordnung der Elementtypen (links) und original Geometrie und richtige Zuordnung der Elemente (rechts)	38

Abbildung 37: Ausschnitt original CDB-Datei, bei der Knoten auf einer Achse liegen.....	39
Abbildung 38: For-Schleife, welche die Koordinaten mit 0 auffüllt, falls diese nicht vorhanden sind	39
Abbildung 39: 3D-Plot der CT-Daten und der Netzgeometrie mit falschem Wert (links) und richtigem Wert (rechts) für den Voxelabstand in der z-Achse (Slice Thickness).....	40
Abbildung 40: sich öffnendes Fenster für die Kalibrierung mit gelbem Kreis als Auswahltool und markierten Voxel	41
Abbildung 41: neuer Startblock mit variabler Ansys-Version und NOPR Command...	42
Abbildung 42: Hauptbenutzeroberfläche bei Start des Programms, wobei die Flächen für die 3D-Darstellungen nicht mit abgebildet sind (s. Anhang 8).....	43
Abbildung 43: CT-Volumendarstellung des Gelatinezylinders mit unterschiedlichen Masseanteilen an Calcium und dem beigelegten Phantom	45
Abbildung 44: Schnittbilder der zwei Calciumabschnitte des Testzylinders mit Lufteinschlüssen und Randfehlern.....	46
Abbildung 45: Schnittbild des CT-Scans des Gelatineblocks mit untergelegtem Phantom mit den isolierten Zylindern 1 bis 5 (von links nach rechts) ..	46
Abbildung 46: rekonstruierter Gelatinezylinder (rot) und Ausgangszylinder (transparent)	47
Abbildung 47: Auswahlansicht der ROI im reinen Gelatineblock in der klinischen Software zur Ermittlung der BMD	48
Abbildung 48: E-Moduln durch Verwendung von verschiedenen Kalibrierungskorrelationen (blau kalibriert (C), grün nicht kalibriert (NC)) und die Differenz der Werte (rot).....	51
Abbildung 49: Einfluss der Elementgröße des FE-Netzes auf die Höhe des E-Moduls der Gruppierungen mit Ergebnissen aus der Python-Routine ..	52
Abbildung 50: linkes Hüftbein mit 2mm Vernetzung (rechts) und 10mm Vernetzung (links) mit gemapptem E-Modul und 200 MPa Gruppierung	53
Abbildung 51: E-Modul pro Gruppierung für 2mm Elementgröße	53
Abbildung 52: Gruppierungen mit Anzahl der Elemente pro Gruppierung für 2mm Elementgröße	54

Tabellenverzeichnis

Tabelle 1: Funktionen der Hauptoberfläche	43
Tabelle 2: K2HP04-Dichten der einzelnen Phantomzylinder	47
Tabelle 3: BMD und HU Werte der eigenen und der klinischen Kalibrierung	49
Tabelle 4: E-Moduln aller Elemente ohne Gruppierung	50
Tabelle 5: Differenzen aller Elemente	50
Tabelle 6: E-Moduln der gruppierten Elemente	52
Tabelle 7: benötigte Rechenzeit, um die gemappte CDB-Datei zu erstellen	54

1. Einleitung

Der demografische Wandel ist in den letzten Jahren und Jahrzehnten in Deutschland stetig gewachsen und wird auch in Zukunft weiter voranschreiten [1]. Dies bedeutet, dass mit einer stark alternden Bevölkerung zu rechnen ist, mit welcher auch ein Anstieg an Volkskrankheiten einhergeht. Zu einer solch verbreiteten Erkrankung zählt Osteoporose*. Die betroffenen Risikogruppen umfassen Frauen und Männer ab einem Alter von 50 Jahren, wobei weibliche Personen öfter betroffen sind.

Die daraus resultierende Abnahme der Knochendichte sowie der Knochenmasse erhöht das Risiko für Knochenbrüche erheblich, welche im Alter der Risikogruppen schlecht heilen. Eine Beckenringfraktur ist eine häufige Folge von Traumen (z. B. ein Sturz aus dem Stand). Insofern es sich um eine instabile Fraktur handelt, wird diese operativ mittels Osteosynthese** therapiert.

So kommt es beispielsweise bei Osteosynthese mittels Zugschrauben zur Kompression der Beckenfragmente. Die Knochen werden durch das Anziehen der Schrauben sowie durch die eigenen Bewegungen und das Körpergewicht beansprucht. Um eine Berechnung solcher Belastungen vorzunehmen, wird die Finite-Elemente-Methode (FEM) eingesetzt. Zur Anwendung dieser numerischen Berechnungsmethode müssen jedoch sämtliche Materialkennwerte bekannt sein. Da der Knochen kein gleichmäßiges Materialverhalten, sondern eine gewisse Dichteverteilung aufweist, lässt sich kein homogenes Materialmodell des Volumenkörpers für die Berechnungssoftware definieren.

Ziel der Arbeit ist es, eine bestehende Python-Routine weiterzuentwickeln, welche auf Basis von Computertomografie-Daten (CT-Daten) die Materialkennwerte Dichte und E-Modul den einzelnen Elementen des FE-Netzes zuordnet. Somit lässt sich ein realistischeres Modell in Berechnungssoftware implementieren, was wiederum zu genaueren Ergebnissen in darauffolgenden Festigkeitsberechnungen führt.

Die Ergebnisse können sowohl bei der Osteosynthesewahl, der Höhe der Anzugsmomente von Schrauben, als auch bei der Platzierung von Implantaten, in individueller Abhängigkeit zum Patienten, hilfreich sein. Darüber hinaus kann die ermittelte Knochendichteverteilungen Information zur Beurteilung der Knochenqualität, hinsichtlich der Erkrankung Osteoporose, liefern.

* Osteoporose bezeichnet eine Knochenkrankheit, bei der sich die Knochendichte im Alter verringert, umgangssprachlich auch Knochenschwund genannt. [31]

** Beschreibt die operative Versorgung von Frakturen mittels Fremdmaterial (Platten oder Schrauben)

1.1 Problemstellung

Der menschliche Knochen besteht aus unterschiedlich ausgebauten Phasen, unter anderem der Kortikalis und der Spongiosa, welche Grund für die Knochendichteverteilung des Knochens sind. Die Spongiosa ist ein schwammartiges Gewebe, welches sich innerhalb des Knochens befindet. Als Kortikalis wird die äußerste Knochenschicht bezeichnet, welche die Spongiosa umgibt und eine weitaus höhere Festigkeit als die Spongiosa aufweist. Bei Osteoporose verringert sich die Knochendichte in beiden Regionen, vor allem aber im spongiösen Bereich. Da im Alter der Risikopatienten ebenso Krankheitsbilder, exemplarisch Arthrose, vorkommen, werden häufig Implantate (z.B. künstliches Hüftgelenk) oder Osteosynthese-Operationen vorgenommen. Solch ein operativer Eingriff wird bei Osteoporose erschwert, da durch den weniger dichten Knochen der Halt der Befestigung (Schrauben oder Nägel) des Fremdmaterials im Körper verschlechtert wird (s. Abb. 1).

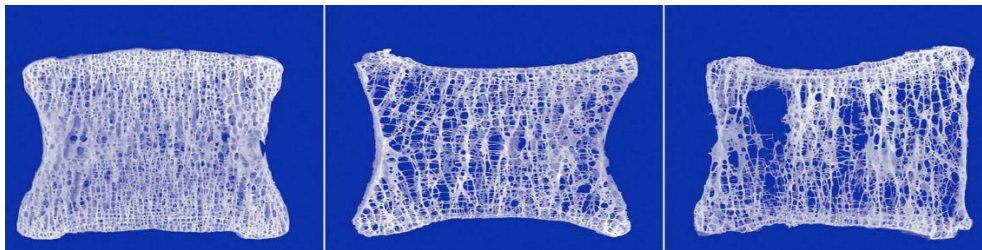


Abbildung 1: gesunder Knochen (links) und osteoporotischer Knochen (mittig und rechts) [2]

Die Befestigung einer Schraube im Knochen der rechten Abbildung 1 wäre nicht von langer Dauer, da zu wenig Knochengewebe vorhanden ist. Vor der Operation zu wissen, wie sich welche Region im Knochen verhält, wäre von Vorteil, wofür beispielsweise eine FE-Analyse genutzt werden kann.

Für Simulation in FEM-Software sind mindestens zwei Materialkennwerte nötig, der Elastizitätsmodul (E-Modul) und die Querkontraktionszahl. Der E-Modul beschreibt das Verhältnis von Spannung und Dehnung und die Querkontraktionszahl die Volumenänderung unter Zug- oder Druckbelastung.

Für den E-Modul existieren verschiedene Korrelationen, die auf empirisch ermittelten Daten basieren. Die Herleitung erfolgt über einen Zusammenhang von Skalarwerten aus dem CT und der Knochendichte und folgend ebenso aus Knochendichte und E-Modul. Jedem einzelnen Element wird ein konstanter Elastizitätsmodul und eine konstante Dichte zugeordnet, wobei die Querkontraktionszahl von 0,3 für alle Elemente gleich ist. Das gesamte Modell setzt sich somit aus vielen kleinen isotropen Elementen zusammen. Die konstante Querkontraktionszahl ergibt sich aus der isotropen Betrachtung und dem am meist verwendeten Wert in der Literatur. Darüber hinaus gibt es derzeit keine akkurate Korrelation, die die Querkontraktionszahl in Abhängigkeit von CT-Werten beschreibt.

Es bestehen bereits verschiedene Algorithmen zur Berechnung der Dichtewerte und des Elastizitätsmoduls von Knochengewebe und zur Zuordnung dieser Werte zu den einzelnen Elementen eines FE-Netzes. Das älteste und in der Literatur am häufigsten zitierte Programm, das die aufgeführten Zusammenhänge nutzt und auch als Open-Source zur Verfügung steht, ist „Bonemat“. Dieses Programm hat jedoch gewisse Nachteile, die unter anderem die Motivation für diese Arbeit darstellen:

- in Bonemat ist keine Kalibrierfunktion implementiert
- nicht alle vorliegenden CT-Datensätze können in Bonemat eingelesen werden, wobei der Fehler unbekannt ist

Ein CT-Scan mit gleichem Objekt auf unterschiedlichen CT-Geräten kann aufgrund ungleicher Einstellungen und Gerätehersteller zu verschiedenen Ergebnissen der CT-Werte führen. Um dies zu verhindern, gibt es die Möglichkeit des Kalibrierens.

Eine Kalibrierung der CT-Datensätze wird über Phantome durchgeführt, die zum scannenden Objekt dazu gelegt werden. Die Dichte der Phantome ist bekannt, wodurch über eine Regression eine Korrelation von CT-Daten und Knochendichte bestimmt werden kann. Der Zusammenhang der Dichte und der CT-Werte ohne Kalibrierfunktion kann verschiedener Literatur entnommen werden, führt aber zu einem ungenaueren Ergebnis. Aufgrund der genannten fehlenden Eigenschaften Bonemats schrieb A. Hanan eine Routine in Python [3], welche allerdings bis zum Ende seiner Masterarbeit nicht komplett fertiggestellt wurde.

1.2 Zielstellung

Das Hauptziel der Arbeit ist es, das von A. Hanan geschriebene Programm fertigzustellen, um eine Kalibrierfunktion zu erweitern und eine Benutzeroberfläche zu programmieren. Dabei werden folgende Punkte bearbeitet:

- Berechnungsmethoden und Korrelationen zwischen HU-Werten und Knocheneigenschaften ausarbeiten
- Vergleich verschiedener Mapping-Methoden und Zuordnung der Materialkennwerte auf einzelne Elemente eines FE-Netzes
- Evaluierung anhand eines Minimalbeispiels, welches bekannte Werte für die Dichte besitzt und Vergleich mit Ergebnissen von Bonemat
- Durchführen des gesamten Prozesses anhand des linken Hüftbeins
- Dokumentation des vorhandenen Programms, um das Einarbeiten in den Code zu erleichtern

1.3 Programmablauf

Der grobe Ablauf der benötigten Schritte, um die Berechnungen durchführen zu können, wird im Folgendem erklärt (s. Abb. 2).

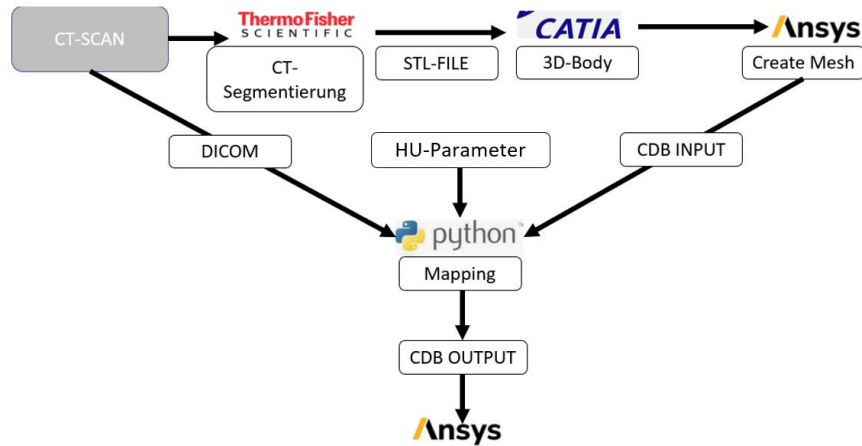


Abbildung 2: Datenfluss der benötigten Dateien für die Mapping-Routine

Die Basis aller benötigten Daten für das Programm sind CT-Daten. Für die Mapping-Routine selbst sind 2 Dateien und die Korrelationsparameter notwendig:

- CT-Datensatz
- Ansys CDB-Datei
- Parameter für Materialkorrelationen

Die Ansys CDB-Datei ist die Programm-Input-Datei, welche Geometrie (Netzeigenschaften) und Materialinformationen des 3D-Modells beinhaltet. Über das Programm Amira können die CT-Daten eingelesen sowie selektiert und als STL-Datei abgespeichert werden. Das STL kann in einer CAD-Software in einen Volumenkörper umgewandelt und anschließend in ANSYS importiert werden. In ANSYS kann durch einen Befehl in der eigenen Programmiersprache die CDB-Datei exportiert werden.

Die Korrelationsparameter können im Programm eigenständig festgelegt werden. Das Programm legt in den geometrischen Raum des CT-Datensatzes das vorher erstellte Geometriemodell und ordnet jedem Element CT-Werte, und daraus folgend einen Dichtewert und ein E-Modul, zu.

2. Grundlagen CT und Materialkorrelationen

2.1 Computertomographie

Die Basis aller Daten, welche in der Routine verarbeitet werden, ist der CT-Scan. Dementsprechend hängt das Ergebnis und dessen Qualität vom Scan ab, womit die Richtigkeit der Eingangsdaten von Bedeutung ist.

In der Medizin existieren verschiedene Verfahren zum Rekonstruieren (Erstellen von Schnittbildern des Körpers) von Körpergliedmaßen. Die meist angewandten sind die CT und die Magnetresonanztomographie (MRT). Die Hauptunterschiede dieser Verfahren liegen in der Funktionsweise und dem Anwendungsgebiet. Während die CT eine bessere Abbildung der Knochen hergibt, kann die MRT Weichteile (z.B. Tumore) genauer darstellen.

Die Computertomographie basiert auf der Röntgentechnik, liefert allerdings - im Vergleich zum klassischen Röntgenbild - mehrere transversale (Z-Richtung) Schnittbilder und ermöglicht somit eine dreidimensionale (3D) Darstellung. Das CT-Gerät besteht grundlegend aus folgenden Bauteilen: Gantry, Röntgenröhre, Patiententisch, Filter und Blenden sowie einem Detektor (s. Abb. 3)

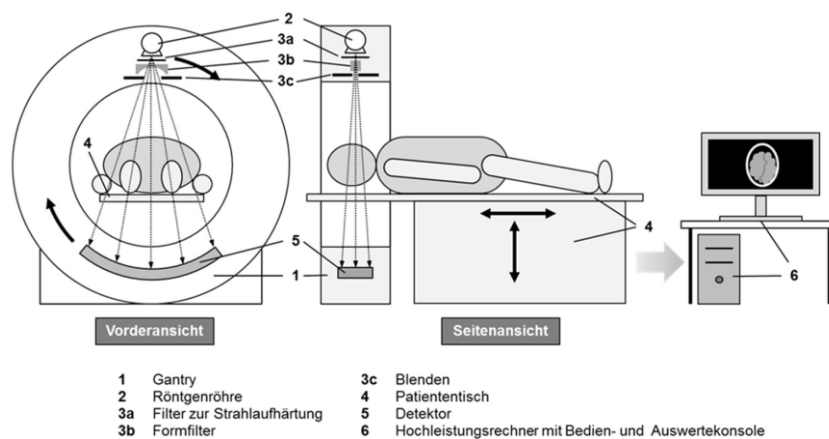


Abbildung 3: Aufbau eines CT-Geräts mit den wichtigsten Bestandteilen [4]

Die Gantry ist ein Gerüst, an dem hauptsächlich die Röntgenröhre und der Detektor angebracht sind und welches die Rotation dieser ermöglicht. Der Formfilter schwächt den Röntgenstrahl, um ihn an die seitliche Kontur des Körpers anzupassen. Mittels der Blende wird der Röntgenstrahl seitlich (X-Achse) und in Längsrichtung (Z-Achse) begrenzt. Ein verstellbarer Patiententisch verfährt während des Scans und der sich rotierenden Röntgenröhre und Detektor in Z-Richtung, damit Bilder in unterschiedlichen Schnittebenen aufgenommen werden können. Der Vorschub des Tisches erfolgt kontinuierlich oder

diskontinuierlich, wodurch eine Axiale- oder Spiralaufnahme entsteht. Der Detektor liegt der Röntgenröhre immer gegenüber und besitzt eine gebogene Form, wodurch der fächerförmige Röntgenstrahl (mit Aufspannwinkel Φ) an den Seiten die gleiche Wegstrecke durchläuft. Neuere Detektoren bestehen aus mehreren Zeilen und können so mehr als nur ein Schnittbild gleichzeitig aufnehmen, was die Aufnahmezeit verkürzt [4]. Der dabei in Z-Richtung aufgespannte Winkel wird Kegelwinkel (Γ) genannt und hängt von der Anzahl der Detektorzeilen ab. Die Anzahl der Röhre-Detektoreinheiten, der Kegelwinkel, der Aufspannwinkel und der entsprechenden Z-Abdeckung (Kollimierung in mm) variiert unter den Herstellern (s. Abb. 4).

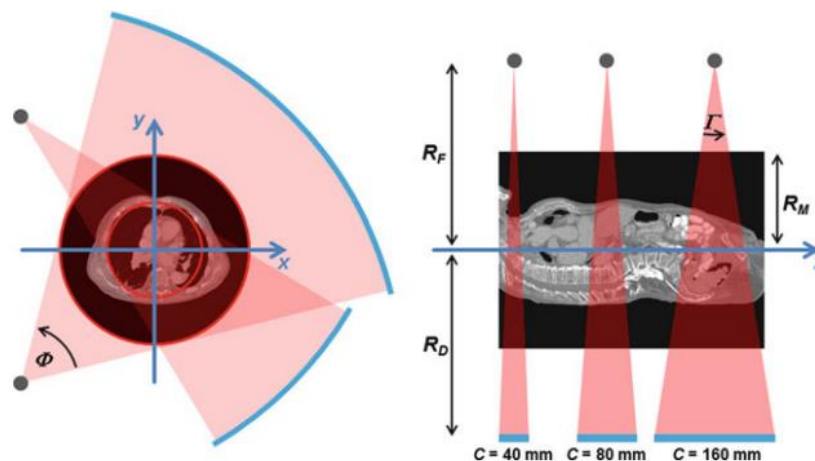


Abbildung 4: links: Schnittbild in X-Y bzw. transversaler Ebene mit 2 Röhre-Detektor-Einheiten; rechts: unterschiedliche Aufspannwinkel und Kollimierung des Röntgenstrahls in Z-X bzw. sagittaler Ebene [5]

Die Röntgenstrahlen werden durch den Körper abgeschwächt, was von den Detektoren erfasst wird. Die Schwächung der Strahlen wird durch das Lambert-Beer'sche Gesetz (Gl. 2.1) beschrieben:

$$N = N_0 e^{-\mu d} \quad (2.1)$$

Wenn eine bestimmte Photonenzahl N_0 (auch Intensität I) auf einen Absorber (Patient) mit der Dicke d und dem linearen Schwächungskoeffizient μ trifft, so durchdringt die Menge N an Photonen den Absorber. Die restlichen Photonen werden im Körper des Patienten absorbiert oder in andere Richtungen gestreut. Durch Messung der Photonenzahl N nach dem Absorber durch den Detektor, ist der Schwächungskoeffizient die einzige Unbekannte in der Gleichung. Der Schwächungskoeffizient ist eine Absorber-Eigenschaft. Da der Knochen keine konstante Dichte aufweist und der Schwächungskoeffizient von der Dichte abhängt, kann nach diesem umgestellt werden und es ergibt sich ein Wert, welcher repräsentativ für die Abschwächung der Intensität des Röntgenstrahls ist. Dabei

ist im Fall des CT-Scans der beeinflussende Parameter die Dichte des Absorbers. Je höher die Dichte des Absorbers, desto höher ist die Abschwächung der Intensität, desto größer ist μ (s. Abb. 5).

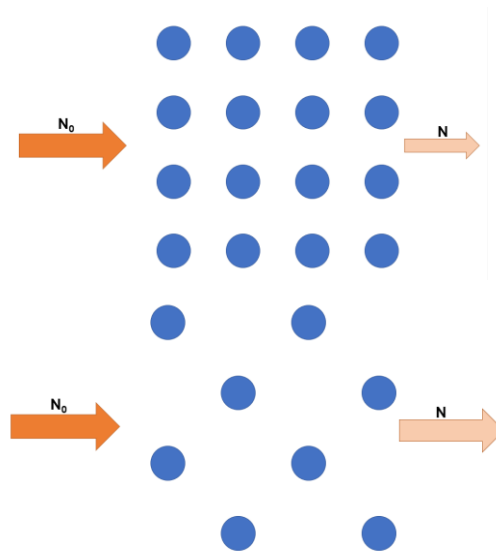


Abbildung 5: Eingangspotonenzahl N_0 des Röntgenstrahls, der auf einen Absorber mit hoher (oben) und niedriger (unten) Dichte trifft und dabei unterschiedlich stark absorbiert. Durch die unterschiedliche Dichte durchdringen unterschiedlich viele Photonen N den Absorber

Der Computertomograph misst durch die translatorische und rotatorische Bewegung die Schwächungseigenschaften des Patienten aus vielen verschiedenen Blickrichtungen und Winkeln, sodass aus den gesamten Daten die Verteilung $\mu(x, y, z)$, durch mathematische Rekonstruktion, errechnet wird. Bei der Rekonstruktion des Bildes wird das entstehende Bild in 512×512 Pixel (klinisches CT) mit variabler Pixelbreite und Pixelhöhe, auch *PixelSpacing* (PS), eingeteilt. Die dritte Dimension wird durch den Abstand der Schnittbilder erreicht und als *SliceThickness* (ST) bezeichnet. Dem entstandenen Voxel* mit den Dimensionen PS_Y , PS_X und ST kann nun ein μ -Wert mit den Koordinaten x , y und z zugeordnet werden. Da sich der μ -Wert der körpereigenen Substanzen sich allerdings nur gering vom μ -Wert des Wassers unterscheidet, wird dieser mittels der Formel (Gl. 2.2):

$$CT - Wert \text{ bzw. } HU^{**} = 1000 \times \frac{\mu_{Objekt} - \mu_{Wasser}}{\mu_{Wasser}} \quad (2.2)$$

auf eine einheitslose Skala transformiert, welche als Hounsfield-Skala (mit einheitslosem CT-Wert bzw. HU) bezeichnet wird. Diese stellt im Prinzip eine Grauwerttabelle dar, in

* 3D-Pixel

** HU steht für Hounsfield Unit

der unterschiedliche Substanzen unterschiedlichen Grautönen zugeordnet sind [6] (s. Abb.5).

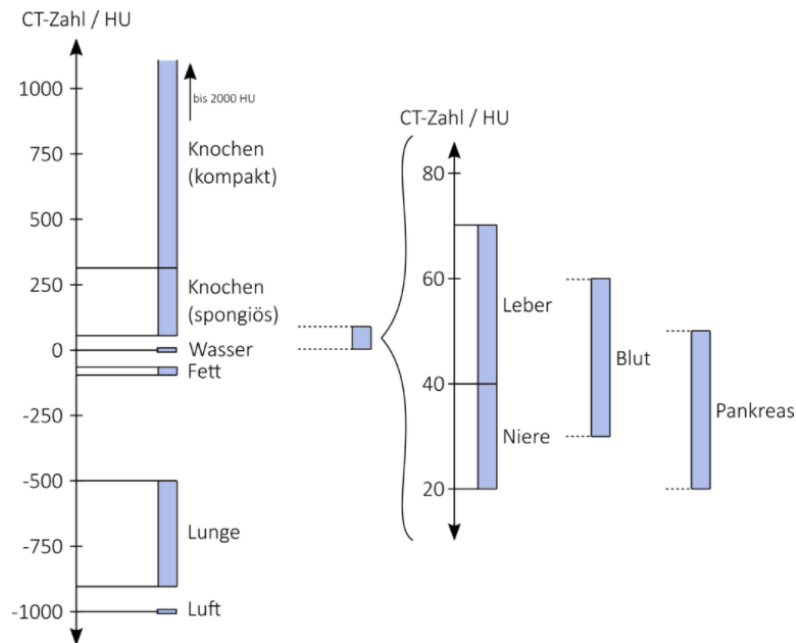


Abbildung 6: Hounsfield Skala mit gefensterstem Bereich und typischen CT-Werten für unterschiedliche Gewebetypen [7]

Per Definition wird Luft der Wert -1000 zugeordnet und Wasser der Wert 0. Die Visualisierung der Schnittbilder erfolgt demnach in Binärbildern (Bilder mit ausschließlich schwarzen oder weißen Pixeln bzw. aus Graustufen). Je höher bzw. niedriger der Wert, desto heller bzw. dunkler sind die Pixel. Da für das menschliche Auge nicht mehr als ca. 100 Graustufen zu unterscheiden sind, kann die Skala weiter in Fenster eingeteilt werden, um den für die Anwendung relevanten Bereich besser erkennbar zu machen. Definiert ist das Fenster durch Zentrum und Breite, welches im Fall des Fensters in Abb.2 einen Wert von 50 (Zentrum) bzw. 60 (Breite) hat.

Alle Pixel, die in der jeweiligen Schnittebene einen Wert kleiner als 20 aufweisen, würden schwarz dargestellt werden, alle Pixel mit einem Wert größer als 80, weiß.

2.2 DICOM-Standard

Die digitalen bildgebenden Verfahren, wie die CT, sind mitbestimmend für den hohen Gesundheitsstandard, der hierzulande geführt wird. Durch Archivierung der Daten und dem Bedürfnis der Bildverarbeitung im Krankenhaus musste ein Mittel gefunden werden, um die Daten geräte- und herstellerübergreifend nutzen zu können. Seit 1995 ist der DICOM-Standard (Digital Imaging and Communications in Medicine) in Europa als formaler Standard gesetzt, welcher eine offene Kommunikationsplattform für medizinische

Bilder und bildbezogene Informationen darstellt [8]. Dabei beschreibt DICOM nicht nur das abgespeicherte Dateiformat. Darüber hinaus gibt der Standard Auskunft über das Übertragen, Abrufen, Drucken, Verarbeiten und Anzeigen medizinischer Bilddaten und zugehöriger Informationen. Dementsprechend müssen alle Hersteller, die DICOM-fähige Geräte auf den Markt bringen, verschiedene Richtlinien befolgen, damit ein reibungsloser Informationsaustausch möglich ist.

Der Aufbau einer DICOM-Datei erfolgt entsprechend einer hierarchischen Struktur (s. Abb. 7).

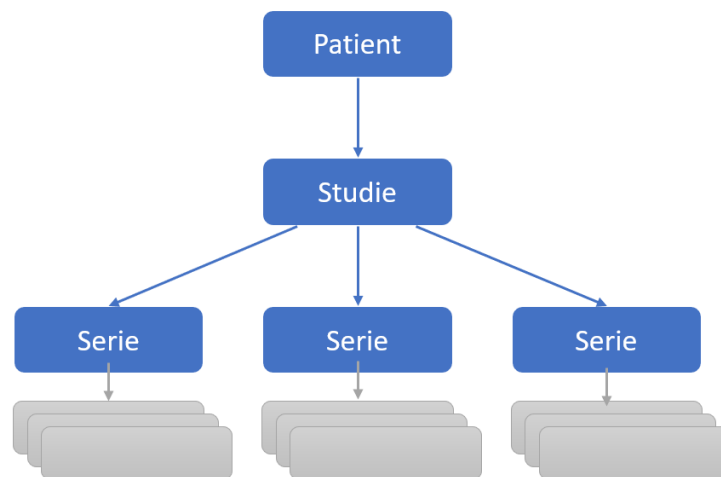


Abbildung 7: Aufbau DICOM

Einem Patienten wird im ersten Schritt eine bestimmte Studie zugeordnet. Diese bezieht sich beispielsweise auf verschiedenen Patientenbesuche oder auf krankheitsbedingte Untersuchungen. Jeder Studie können dann wiederum unterschiedliche Serien zugeordnet werden, welche exemplarisch unterschiedliche bildgebende Scans beinhalten. DICOM ermöglicht in solchen Fällen eine eindeutige Zuweisung und Abspeicherung der Daten über unterschiedliche Zeiträume und Befunde des Patienten hinweg.

Jedes Bild einer Serie wird als Data-Set bezeichnet, die jeweiligen Informationen im Bild werden Data-Element oder Attribut genannt. Die Attribute werden in Modulen gruppiert abgespeichert. So enthält das Modul *Patient* patienten-spezifische Informationen wie Alter, Patienten-ID, Geschlecht, Name etc. (s. Abb. 8).

SOP Common	SOP Class UID SOP Instance UID
Patient	Patient's Name Patient ID Patient's Sex
Study	Study UID Study Date Study Time Accession Number
Series	Series UID Series Number

Abbildung 8: Aufbau eines DICOM-Bildes mit Modulen und Attributen [9]

In jeder Definition eines Moduls ist aufgeführt, ob dieses zwingend notwendig ist und vorhanden sein muss (*M – Mandatory*), ob das Vorhandensein des Attributs an Bedingungen geknüpft ist (*C – Conditional*) oder ob es einer Benutzerentscheidung unterliegt (*U – User Option*).

Die einzelnen Attribute beinhalten wiederum die Informationen *Data Tag*, *Attribute Name*, *Value Representation (VR)*, *Value Multiplicity (VM)*, *Type*, *Attribute Length* und *Attribute Value*. Die Definition des Attributs erfolgt über den *Data Tag*, welcher zwei Zahlen enthält, die für die Gruppierung und das jeweilige Element stehen. So kann das Modul *Patient* die Attribute mit den Tags (0010,0010) und (0010, 0020) mit den Namen *Patient Name*, *Patient ID* beinhalten, dabei gibt es mehrere Gruppierungen innerhalb eines Moduls. Die Information *VR* beschreibt die Art, der im Attribut gespeicherten Daten. Das kann beispielsweise *DS* (Decimal String), *IS* (Integer String) oder *ST* (Short Text) sein. *VM* bezieht sich auf die Multiplizität eines Attributes. Ein Winkel hat beispielsweise den VM-Wert von 1, ein Koordinatenpunkt den VM-Wert 3, welcher für die 3 benötigten Koordinaten der 3 Raumachsen steht. Der *Type* steht für die Notwendigkeit des Attributs und kann die Werte 1, 2 oder 3 beinhalten. 1 steht für *zwingend notwendig und der Inhalt muss vorhanden sein*, 2 steht für *zwingend notwendig, kann aber leer sein* und 3 steht für *nicht zwingend notwendig bzw. optional*.

Attribute Length und *Attribute Value* beschreiben die Größe und den tatsächlichen Inhalt des Attributs. Alle Module und Attribute und deren Beschreibung können über den *DICOM Standard Browser* eingesehen werden (a. Abb. 9 und 10).

DICOM Standard Browser by Innolitics

▶ CR Image		CIOD
▼ CT Image		CIOD
▼ Patient	M	Module - Patient
▶ (0008,1120) Referenced Patient Sequence	3	Sequence
(0010,0010) Patient's Name	2	Person Name
(0010,0020) Patient ID	2	Long String
(0010,0021) Issuer of Patient ID	3	Long String

Abbildung 9: DICOM Standard Browser mit geöffneten Informationen einer CT-Serie mit geöffnetem Modul „Patient“ und dessen Notwendigkeit „M“ und einigen zugehörigen Attributen mit Tag, Name und VR und Type

Patient's Sex Attribute

Tag	(0010,0040)
Type	Required, Empty if Unknown (2)
Keyword	PatientSex
Value Multiplicity	1
Value Representation	Code String (CS)
Example Values	F M

Abbildung 10: genauere Beschreibung des Attributs „Patient Sex“ mit zugehörigem Tag, Type, VM, VR und Beispiel Inhalt (Attribut Value) [10]

In der Python Routine wird das Package *Pydicom* verwendet, um die DICOM Daten einzulesen. Mit dem Package wird ein Zugriff auf alle, im Scan erstellten Attribute, ermöglicht (s. Abb. 11).

```

(0008, 0005) Specific Character Set          CS: 'ISO_IR 100'
(0008, 0008) Image Type                     CS: ['ORIGINAL', 'PRIMARY', 'AXIAL']
(0008, 0016) SOP Class UID                   UI: CT Image Storage
(0008, 0018) SOP Instance UID               UI: 1.3.6.1.4.1.24930.03377529.5387.7017
(0008, 0020) Study Date                     DA: '20101002'
(0008, 0021) Series Date                   DA: '20101002'
(0008, 0022) Acquisition Date              DA: '20101002'
(0008, 0023) Content Date                  DA: '20101002'
(0008, 0030) Study Time                    TM: ''
(0008, 0033) Content Time                  TM: '092138'
(0008, 0060) Modality                      CS: 'CT'
(0008, 1030) Study Description              LO: 'TC COSCIA (HIPOP) D'
(0008, 103e) Series Description             LO: ''
(0010, 0010) Patient's Name                 PN: '-----^-----^'
(0010, 0020) Patient ID                    LO: '-----'
(0010, 0030) Patient's Birth Date          DA: ''
(0010, 0040) Patient's Sex                 CS: 'F'
(0018, 0022) Scan Options                  CS: 'HELICAL MODE'

```

Abbildung 11: durch *Pydicom* erstellte Liste der Attribute und deren Tags, Name, VR, und Value eines Bildes (Liste unvollständig)

2.3 HU-Korrelationen

Es bestehen verschiedene Korrelationen, um von HU-Werten auf die Knochendichte und den E-Modul zu schließen [11]. Da diese Zusammenhänge auf empirisch ermittelten Daten basieren, unterschieden sich die Korrelationen untereinander, abhängig von der Studie. Im Wesentlichen gilt es folgende Unterschiede zu beachten:

- a) Dichteart:
 - Knochenmineraldichte ρ_{QCT} oder **BMD** (body mineral density)
 - Aschedichte ρ_{ASH} (ash density)
 - Scheinbare Dichte ρ_{APP} (apparent density)
 - b) Region der gemessenen Dichte im Körper u.a.:
 - Wirbelsäule
 - Oberschenkelknochen (lat. Femur)
 - Schienbein (lat. Tibia)
 - Oberarmknochen
 - c) Region der gemessenen Dichte im Knochen:
 - Kompakta (äußerer Knochen)
 - Spongiosa (innerer Knochen)
 - d) isotrope und anisotrope E-Moduln
 - e) Zulässiges Dichtefenster:
 - Zulässige Werte der Dichte bzw. gültiger Bereich der HU-Werte für die Korrelation
 - f) Praktischer Versuchsaufbau:
 - 3-Punkt-Biegeversuch
 - Druckversuch
 - Kompletter Knochen
 - Proben aus dem Knochen
 - g) Regression zur Parameterermittlung:
 - Linear
 - Exponentiell
- a) Aufgrund der vielzähligen Definitionen der Knochendichte, ist es wichtig, die Art der Messung zu berücksichtigen. Die scheinbare Dichte des Knochens wird aus dem Quotienten des Nassgewichts und Gesamtvolumen des Knochens bestimmt. Für die Aschedichte wird der Quotient aus dem Aschegewicht und dem Gesamtvolumen des Knochens gebildet [12]. Dafür wird der Knochen mit Wasser gereinigt und das Knochenmark entfernt. Danach wird der Knochen rehydriert (in destilliertes Wasser getaucht und unter Vakuum entgast) und anschließend zentrifugiert, um Wasser von der Oberfläche zu entfernen. Die danach an der Luft

gewogene Masse, entspricht dem Nassgewicht. Für das Aschegewicht wird der Knochen anschließend getrocknet und eingeäschert. Das Verfahren ist dabei nicht in vivo (am lebenden Organismus) umsetzbar. Die quantitative Computertomographie bietet dagegen ein Verfahren, welches in vivo durchgeführt werden kann. Durch das Mitscannen eines Phantoms, bei welchem die Dichte bekannt ist, kann durch lineare Regression eine Korrelation zwischen HU und BMD hergestellt werden [13]. Jedoch werden fast ausschließlich Korrelationen verwendet, bei dem der E-Modul von der Aschedichte oder scheinbaren Dichte abgeleitet wird. Die Zusammenhänge in [14], [15], [11], [16] berechnen den E-Modul direkt aus der Asche- oder scheinbaren Dichte [14], [16] oder berechnen aus der BMD die Aschedichte [15]. Aus dieser kann dann direkt der E-Modul oder die scheinbare Dichte und daraus der E-Modul ermittelt werden.

- b) Die Knochen, welche für die Versuche eingesetzt wurden, sind unterschiedlicher Natur. Die am meist verwendeten Knochen sind das Femur und die Tibia, da diese sich am einfachsten als komplette Knochen in Versuchsaufbauten verwenden lassen. Ergebnisse der Studien zeigen, dass sich die Zusammenhänge von Dichte und E-Modul je nach verwendetem Knochen unterscheiden [14], [17].
- c) Auch die Differenzierung von Kompakta und Spongiosa führt zu unterschiedlichen Ergebnissen, da sich in den Experimenten deutlich andere Zusammenhänge von Dichte und E-Modul ergeben [18]. Die in der Literatur gesetzte Grenze liegt hier bei ca. $0,5 \frac{g}{cm^3}$ (ρ_{QCT}) [13], [19]. Aus den Studien [20], [21], in der drei bekannte Korrelationen für das Femur verglichen wurden, geht hervor, dass der Zusammenhang von Elise F. Morgan [14] das höchste Bestimmtheitsmaß aufweist. Die Korrelation (Gl. 2.3):

$$E = 6850 \rho_{APP}^{1,49} \quad (2.3)$$

scheint für das Femur (Spongiosa) am besten zu funktionieren, dabei ist E in MPa und ρ in $\frac{g}{cm^3}$. Der gültige Dichtebereich (ρ_{APP}) liegt zwischen $0,26 \frac{g}{cm^3}$ und $0,75 \frac{g}{cm^3}$. Allerdings wurden die Dichten nicht in vivo bestimmt, sodass der Zusammenhang zur Knochenmineraldichte fehlt. In der Studie [15] wird ein Zusammenhang, am Femur getestet und für Kompakta und Spongiosa geltend, von ρ_{APP} zu ρ_{ASH} und von ρ_{ASH} zu ρ_{QCT} mit einem Bestimmtheitsmaß $r^2 \geq 0,99$ hergestellt (Gl. 2.4 – 2.5):

$$\rho_{APP} = \frac{\rho_{ASH}}{0,6} \quad (2.4)$$

$$\rho_{ASH} = 0,87719 \rho_{QCT} + 0,07895 \quad (2.5)$$

Mit ρ_{QCT} aus der Kalibrierung mit dem Phantom in Form von $\rho_{QCT} = a HU + b$ und den Dichten in $\frac{g}{cm^3}$.

- d) Da jedem Element ein konstanter E-Modul und eine Querkontraktionszahl zugeordnet wird, handelt es sich um isotrope Materialeigenschaften. L. Peng [22] und V. Baca [23] vergleichen in den Studien die Ergebnisse verschiedener FE-Analysen mit isotropen und anisotropen Materialeigenschaften (E-Modul, Querkontraktionszahl) und kommen zu dem Fazit, dass die Unterschiede in großen Knochenmodellen so gering ist, dass eine isotrope Betrachtung ausreichend ist.

Allein in der Literatur [11] werden über 20 verschiedene Korrelationen gelistet, die sich in den oben genannten Eigenschaften a bis f unterscheiden und zu unterschiedlichen Ergebnissen führen würden. Diesbezüglich ist es wichtig, im Vorhinein festzulegen, welche Korrelationen verwendet werden. Insofern mit der Knochenmineraldichte gerechnet wird, ist eine Kalibrierung nötig, um akkurate Ergebnisse zu erzielen [15], [24], [25]. Für die verschiedenen Dichten untereinander können Zusammenhänge verwendet werden, die für den gesamten Dichtebereich gelten. Obwohl es auch Korrelationen für den E-Modul gibt, die den gesamten Dichtebereich abdecken, trägt eine Unterscheidung von Kompakta und Spongiosa zur Ermittlung von genaueren Ergebnissen bei.

Die Gleichungen (2.3) bis (2.5) sind als Ursprungswerte in Bonemat hinterlegt. Um eine Vergleichbarkeit der Programme herstellen zu können, werden diese Parameter auch in der Python-Routine als Standard-Werte angegeben, mit der Möglichkeit der Anpassung. Für die Unterscheidung von kompaktem und spongiösem Knochen kann eine zweite Korrelation eingestellt werden, welche ab einem einstellbaren Grenzwert gültig ist. Falls keine Unterscheidung angenommen wird, wird über den ganzen Dichtebereich mit den gleichen Parametern gerechnet.

Darüber hinaus ist anzumerken, dass bis dato keine Studien für den Bereich des Beckens (lat. Pelvis) existieren. In der Studie [21] wird versucht, die Korrelationen, die für das Femur erforscht sind, auf das Pelvis zu übertragen, mit dem Ergebnis, dass die für das Femur bestehenden Korrelationen nur bedingt für das Pelvis geeignet sind. Da die Parameter in der Python-Routine jedoch veränderbar sind, ist die Auswahl dem Anwender überlassen, welcher, je nach dem zu untersuchenden Objekt, die passende Korrelation wählen kann.

3. Material-Mapping

Das Material Mapping beschreibt den gesamten Prozess, der notwendig ist, um am Ende die Geometrie mit orts aufgelösten Materialeigenschaften im passenden Dateiformat für ANSYS vorliegen zu haben. Der Prozess startet mit der Rekonstruktion der Geometrie und anschließender Diskretisierung und folgender Materialzuordnung des diskretisierten Modells über die Python-Routine. Das erzeugte Modell kann nachfolgend wieder in ANSYS importiert werden.

3.1 Rekonstruktion der Geometrie

Die benötigte 3D-Geometrie der Knochenfragmente basiert auf den CT-Daten. Das Programm Amira ermöglicht ein Lesen der Bilderserie im DICOM-Format und das selektieren einzelner Bildpunkte, ein Anpassen der Fenster bzw. des Kontrastbereichs sowie ein Abspeichern im Standard-Triangulation-Language-Format (STL), welches wiederum in ein CAD-Programm importiert werden kann. STL-Objekte bestehen grundlegend aus Koordinatenpunkten, welche in diesem Fall das CT-Bild liefern. Die Oberflächenpunkte werden zu Dreiecken verbunden (Triangulierung), wodurch eine Oberfläche aus kleinen dreieckigen Einzelflächen entsteht und die Konturen des gescannten Objekts dargestellt werden. Eine STL-Datei gibt demnach ausschließlich Information über die äußere Gestalt eines Körpers. Die Güte (Konturgenauigkeit) der erstellten Oberfläche hängt dabei von der Anzahl der Bildpunkte der CT-Datei ab. In der Medizin wird meist die klinische CT angewendet, welches eine Bildauflösung von 512x512 Pixel bietet. Aus der Anzahl der Bilder, auch Slices genannt, und deren Auflösung ergibt sich die Voxelgröße. Die in dieser Arbeit verwendeten Datensätze kommen vom Klinikum Bergmannstrost Halle und sind Aufnahmen des Beckens (lat. Pelvis). Die Maßgenauigkeit der Beckengeometrie hängt somit von der Qualität der Bilder und den einzelnen händisch selektierten Voxel ab. Um die Datenmengen zu kontingieren, wurde nicht das komplette Becken selektiert, sondern nur das linke Hüftbein.

In Abbildung 12 ist ein Slice zu sehen, in dem ein Knochenfragment selektiert und zum 3D-Voxel-Rendering hinzugefügt wurde.

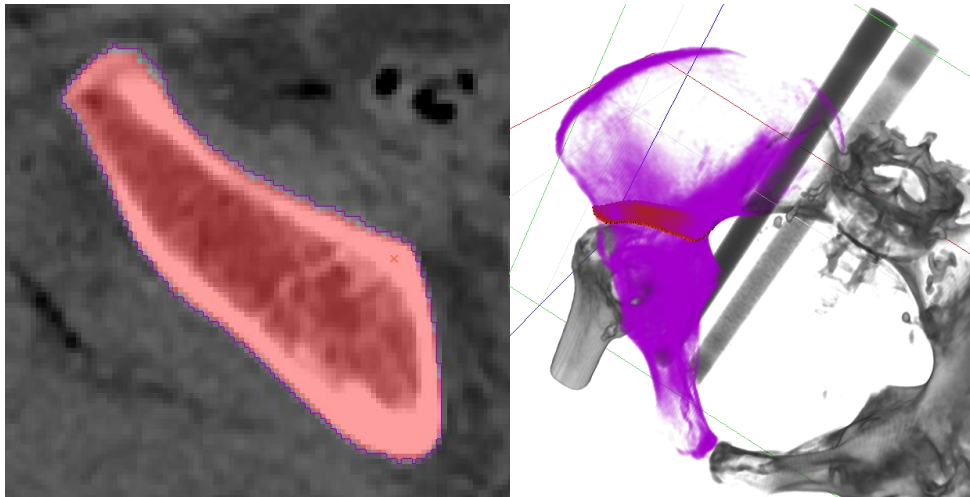


Abbildung 12: Selektiervorgang in Amira mit individuellem Slice mit selektiertem rotem Bereich (links) und gefensterter 3D-Darstellung aller Bilder mit dem gesamten selektierten Bereich (violett) und dem aktuell markierten Bereich (rot)

Die Segmentierung von Knochenfragmenten kann durch verschiedene Funktionen mit automatischer Erkennung gewählter HU-Bereiche erleichtert werden. Dies erfordert dennoch viel Bearbeitungszeit und ein gewisses Basiswissen zur Anatomie des Knochens. Aus dem Voxel-Rendering, welches lediglich die einzelnen selektierten Voxel im Raum darstellt, kann über eine Flächenfunktion das STL erstellt werden. Dieses ist jedoch in der Regel noch fehlerhaft, da zum Teil einzelne Voxel frei im Raum liegen, ohne dass eine Verbindung zu den benachbarten Voxel besteht. (s. Abb. 13).

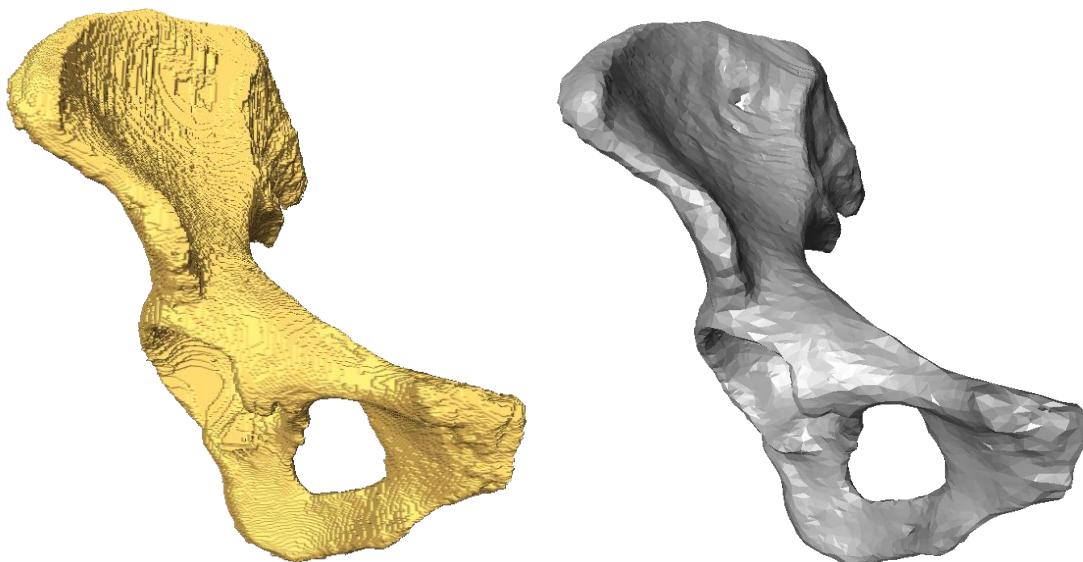


Abbildung 13: Voxel-Rendering (links) und fehlerhaftes STL (rechts)

Bei der Flächenerstellung kann die Größe der Triangulierung eingestellt werden, womit ein erstes Flächenglätten einhergeht. Ein gewisses Maß an Glättung ist jedoch erwünscht,

da dadurch Selektionsfehler wie fehlende Voxel an der Oberfläche oder zu viele hinzugefügte Voxel retuschiert werden.

Um das STL weiter zu verarbeiten, muss es in ein CAD-Programm importiert werden. Im Zuge dieser Arbeit wurde für diese Bearbeitungsschritte CATIA benutzt, mit dem Vorteil, dass eine direkte Anbindung zu ANSYS existiert. Die darauffolgenden Schritte umfassen die Reparatur des fehlerhaften STL und die Erstellung eines Volumenkörpers. Häufig auftretende Fehler in der STL-Oberfläche sind Löcher und Überlappungen der Oberfläche, welche händisch ermittelt und ausgebessert werden müssen. Folgend kann noch eine Glättung des reparierten STL durchgeführt werden, wobei die Stärke der Glättung über die maximale Abweichung der Oberflächen eingestellt wird. (s. Abb. 14).

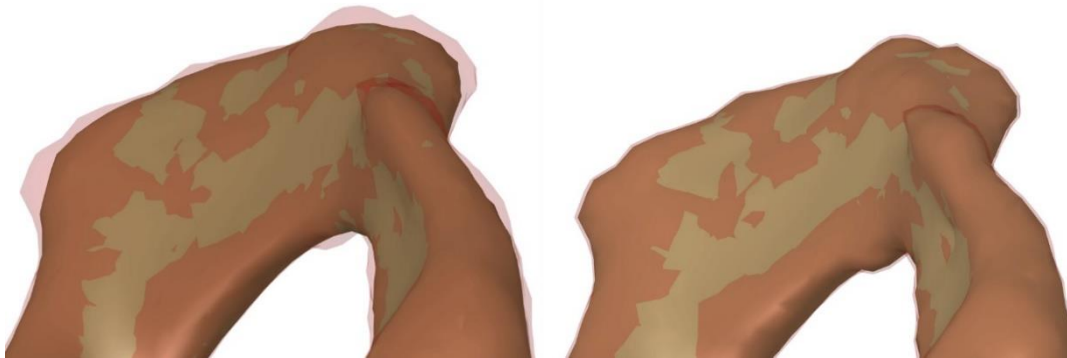


Abbildung 14: Detailverlust gefährdete Regionen des Darmbeins mit transparenter alter Oberfläche und braune Oberfläche des geglätteten Körpers mit starker Glättung und maximaler Abweichung von 1,5mm (links) und Glättung mit mehr Detailerhalt und maximaler Abweichung von 0,4mm (rechts)

Die vorherige Oberfläche ist rot-transparent und die neue geglättete Oberfläche braun dargestellt, wobei erkennbar ist, dass feine Regionen mit vielen Details durch eine zu grobe Glättung verloren gehen.

Der STL-Datensatz kann nach der ersten Glättung in ein Flächenkörper umgewandelt werden. Beim Erzeugen der Oberfläche müssen Parameter gewählt werden, die Einfluss darauf haben, wie genau die Details des STL auf die Oberfläche übertragen werden und wie stark die Abweichung ist (s. Abb. 15).

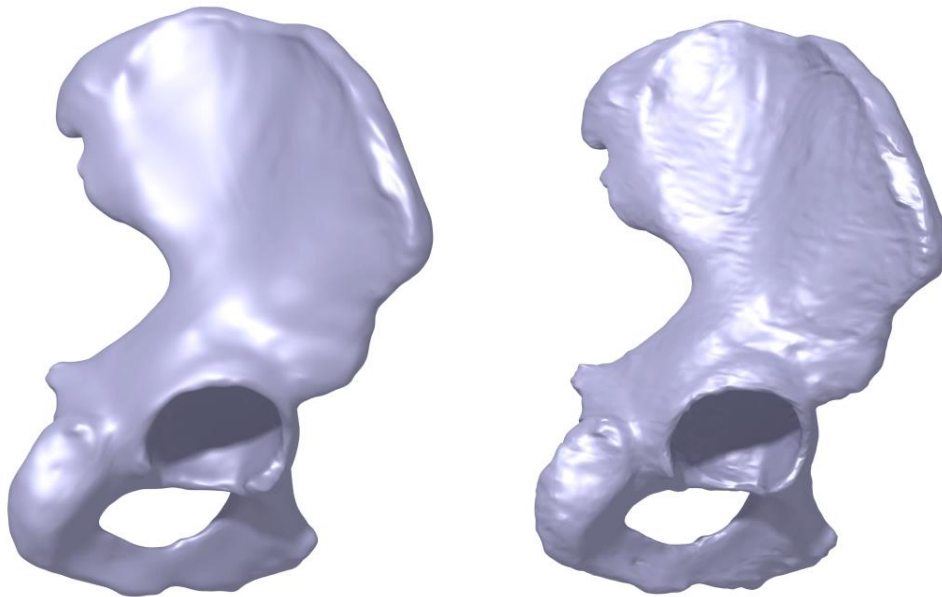


Abbildung 15: erzeugte Flächenkörper auf Basis der STL Dateien mit schwacher Annäherung und starker Glättung (links) und umgekehrt (rechts)

Links in Abbildung 15 wurde eine maximale Abweichung von 3,9mm und eine mittlere Abweichung von 0,08mm und rechts in Abbildung 15 eine maximale Abweichung von 0,63mm und eine mittlere Abweichung von 0,08mm gewählt.

Aus diesem Flächenkörper kann anschließend ein Volumenkörper mit der exakt gleichen Form erstellt werden.

Bei allen Glättungs- und Reparaturoptionen muss stets ein Mittelmaß an Genauigkeit und Glättung gewählt werden, da eine gewisse Glättung erwünscht ist, eine zu starke Glättung jedoch die Geometrie zu stark ändert. Für den Testknochen wurde final eine mittlere Abweichung von 0,08 mm und eine maximale Abweichung von 0,4 mm gewählt.

Durch das Erstellen des Flächen- und Volumenkörpers werden in CATIA automatisch Leitlinien erstellt, die die Flächenform definieren (s. Abb. 16)

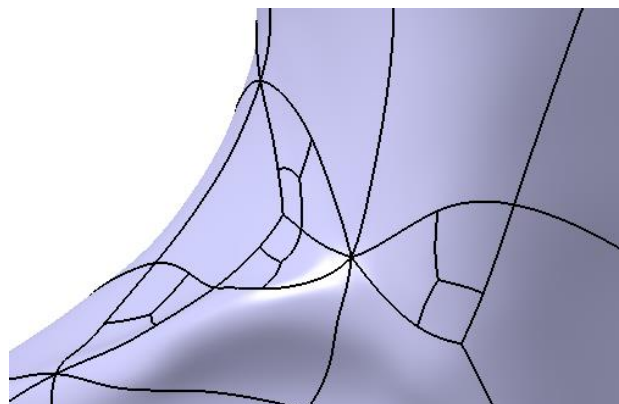


Abbildung 16: Leitlinien des Volumenkörper mit unterschiedlich großen Teilflächen

Diese Flächen werden später vom FE-Programm für die Positionierung der Elemente auf diesen verwendet, sodass gleichmäßig große Teilflächen von Vorteil sind. Wenn einzelne Teilflächen deutlich kleiner sind als der Durchschnitt, können keine gleichmäßig großen Elemente während des Vernetzens erstellt werden. Die Teilflächen können im CAD-Programm oder später im FE-Programm zu größeren Teilflächen zusammengefasst werden. Das CATIA-Part kann direkt in ANSYS geladen werden, wenn die Schnittstelle vorhanden ist, andernfalls muss die Geometrie als STP- oder IGS-Datei gespeichert werden. In ANSYS erfolgt daraufhin die Vernetzung des Bauteils. An dieser Stelle werden die verwendeten Ansatzfunktionen und der Elementtyp festgelegt. Auch die Qualität des erstellten Netzes kann teilweise gesteuert werden, wobei eine Qualität von 1 dem optimalen Element, in Form des Einheitselements, entspricht. Je geringer die Qualität der Elemente, desto stärker weicht die Form vom Einheitselement ab. Ein wichtiger Parameter bei der Diskretisierung ist die Elementgröße, denn je kleiner die Elemente, desto genauer kann die Oberfläche dargestellt werden und desto genauer können spätere Berechnungen erfolgen. Die entstehende Datenmenge steigt jedoch auch mit kleiner werdenden Elementen, was die Berechnungszeit der Python-Routine und folgender FE-Analysen erhöht. Das linke Hüftbein wurde folgend mit einer Elementgröße von 10mm und 5mm vernetzt (s. Abb. 17)

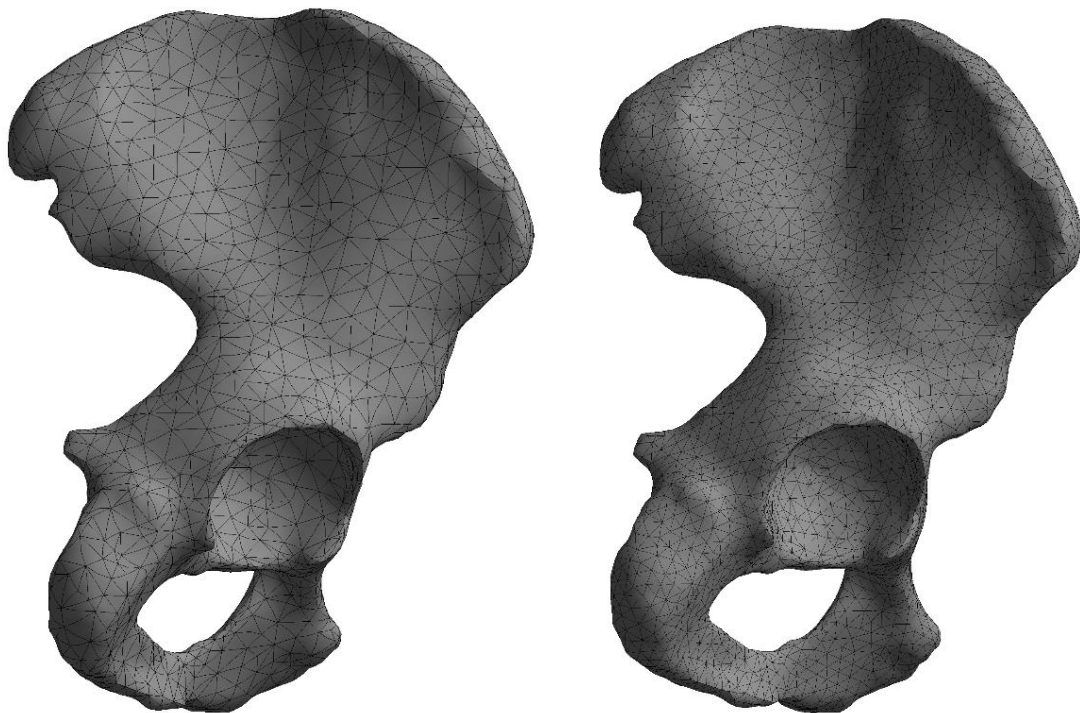


Abbildung 17: Vernetzung des linken Hüftbeins mit 10mm (links) und 5mm (rechts)

Bei zu grober Vernetzung wird die Geometrie nicht ausreichend dargestellt. Bei organischen Formen kommt hinzu, dass vermehrt Radien und Rundungen, welche in der Regel feiner vernetzt werden als gerade Flächen, existieren. Bei einer zu feinen Vernetzung

erhöht sich die Berechnungszeit in jedem Schritt unnötig. Eine gute Vernetzung, bei der modellspezifisch die Elementgröße und der Elementtyp angepasst sind, ist die Grundlage für eine akkurate und genaue FE-Analyse.

Als Schnittstelle der Python-Routine und ANSYS wird die CDB-Datei verwendet.

3.2 ANSYS CDB-Datei

Die ANSYS CDB-Datei ist eine Text-Datei, in der alle Informationen des vernetzten Modellkörpers enthalten sind. Sie kann im *Mechanical* (Ansys Benutzeroberfläche zum Definieren der Simulationsparameter) durch einen eingefügten APDL-Befehl erzeugt werden (s. Anhang 1). Die Datei ist in mehrere Blöcke unterteilt:

- Startblock
- Knotenblock
- Elementblock
- Materialblock

Der Start und das Ende der einzelnen Blöcke sind jeweils durch Header gekennzeichnet. Der Startblock enthält allgemeine Informationen wie den Namen des Modells, ANSYS Version, Anzahl Elemente und Knoten, Temperaturdefinition etc... Ebenfalls enthalten sind Programmbefehle in der internen ANSYS Sprache „ANSYS Parametric Design Language“ (APDL), damit die Datei richtig eingelesen und interpretiert werden kann (s. Abb. 18).

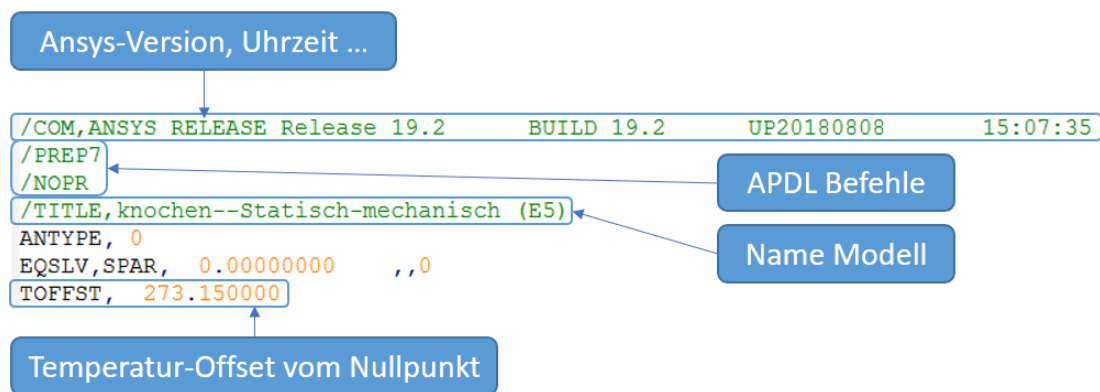


Abbildung 18: Ausschnitt Startblock der CDB-Datei

ANSYS APDL Befehle sind zu bestimmten Gruppen zusammengefasst. Unter anderem gibt es PREP7 und NOPR Befehle. Die Initialisierung der Commands ist in Zeilen 2 und 3 in Abbildung 15 zu sehen. Beispielsweise gehört der TOFFST Command zur PREP7 Gruppe und bestimmt den Temperaturoffset vom absoluten Nullpunkt. Dies bedeutet bei dem gezeigten Offset von 273,15, dass für das Modell die Celsius-Skala definiert ist.

Der Knotenblock enthält alle Knoten des Netzes mit den zugehörigen X, Y und Z-Koordinaten und einen Knotenindex. Der Header enthält u.a. die Anzahl der Knoten (s. Abb. 19)

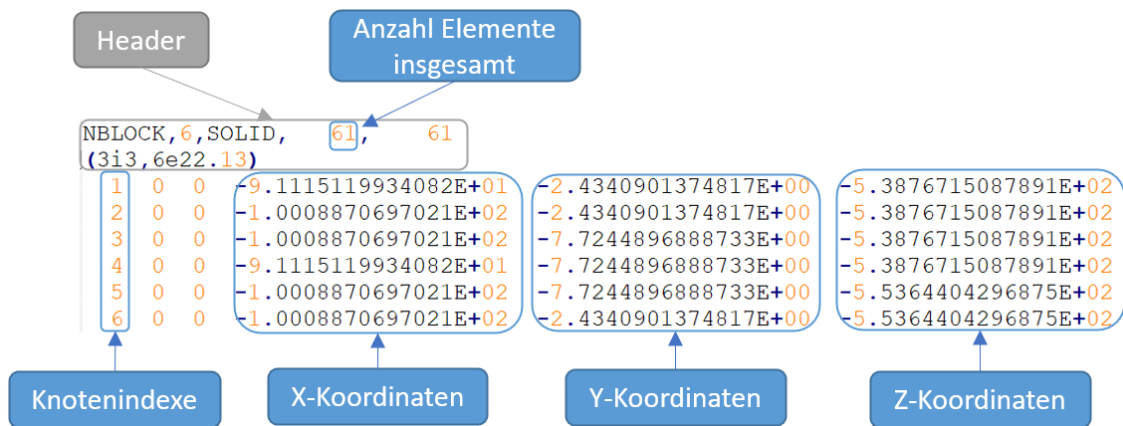


Abbildung 19: Ausschnitt Knotenblock mit Header

Der Elementblock enthält Informationen über die Elemente eines jeden Parts. Dies umfasst den Elementindex, Materialindex, Anzahl Knoten pro Element, Partindex, falls das Modell aus verschiedenen Einzelparts besteht und die Knotenindices, welche das jeweilige Element definieren. Der Partindex ist bei einem Einzelpart für jedes Element 1. Beim Becken, bestehend aus 3 Knochen, ergeben sich Partindices von 1 bis 3. Der Header enthält u.a. die Gesamtanzahl der Elemente (s. Abb. 20)



Abbildung 20: Ausschnitt Elementblock mit Header

Der Materialblock enthält Informationen über die Materialkennwerte, die im Original erstellt wurden. Dieser Block wird durch die Routine mit den individuellen Materialkennwerten, E-Modul und Dichte, erweitert (s. Abb. 21).

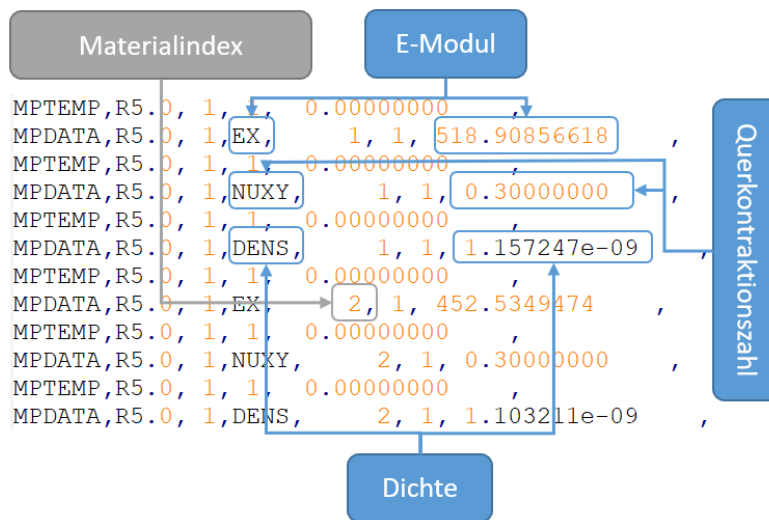


Abbildung 21: Ausschnitt Materialblock mit zwei Materialien

3.3 Mapping-Algorithmen

Die Mapping-Algorithmen beschreiben den Vorgang, bei dem jedem Element des FE-Netzes die Materialkennwerte Dichte und E-Modul zugeordnet werden. Die mathematische Herausforderung dabei ist, zu ermitteln, welche Voxel aus der CT welchem Element zugeordnet werden. Die Algorithmen, welche das Problem lösen, unterscheiden im Wesentlichen die Versionen.

In Bonemat stehen drei verschiedene Algorithmen zur Auswahl, welche nacheinander mit dem Entwicklungsstart 1999 implementiert wurden. Die Versionen werden folgend V1, V2 und V3 genannt, wobei eine abgeänderte Version von V3 in der Python-Routine verwendet wird.

Der Diskretisierungsschritt in der FEM kann unterschiedlich durchgeführt werden. Die entstehenden Elemente können durch verschiedene Formen definiert werden, welche unterschiedliche Vor- und Nachteile bieten. Die einfachste Form in einem dreidimensionalen Gebiet ist der Tetraeder mit 4 Knoten, welcher für die Folgenden Erklärungen der Varianten genutzt wird.

3.3.1 V1

Die mathematisch einfachste Methode ist es, das 3D-Modell des Knochens in den gleichen geometrischen Raum des CT-Scans zu legen und jedem Element ein gemittelten HU-Wert, jener Voxel, welche sich im Element befinden, zuzuweisen. Der gemittelte HU-Wert kann nachfolgend in Dichte und E-Modul umgerechnet werden, sodass jedes Element eigene Kennwerte erhält. Um zu entscheiden, welche Voxel sich innerhalb eines Elements befindet, kann das gerichtete Volumen des Elements mit dem jeweiligen Voxelmittelpunkt bestimmt werden. Sobald sich im Verlauf der Herleitung des Volumens

eine Vorzeichenänderung ergibt, befindet sich der Punkt außerhalb des Körpers. Die folgende mathematische Herleitung erfolgt erst in 2D und daraufhin in 3D.

Über die halbe Determinante einer Matrix $\in R^{2 \times 2}$, welche die Koordinaten zweier Vektoren (x_1, y_1) und (x_2, y_2) mit selbem Ursprung enthält, lässt sich der Flächeninhalt F_Δ des aufgespannten Dreiecks bestimmen:

$$F_\Delta = \frac{1}{2} \det \begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \end{bmatrix} \quad (3.1)$$

Entsprechend der Gleichung (3.1) kann das Ergebnis positiv wie negativ sein, je nach Reihenfolge der Vektoren in der Matrix. Somit ergibt sich der orientierte/gerichtete Flächeninhalt. Für ein beliebiges Dreieck mit den Eckpunkten $u, v, w \in R^2$ wird ein vierter Punkt o zur Berechnung des Flächeninhalts hinzugezogen. Dieser Punkt ist jener, der „überprüft“ werden soll, ob er sich innerhalb oder außerhalb des Gebiets befindet. Wenn sich der Punkt innerhalb befindet, setzt sich der Flächeninhalt aus den drei Einzelflächen zusammen, welche sich durch das Verbinden des Punktes o mit den anderen Punkten bilden (s. Abb. 22).

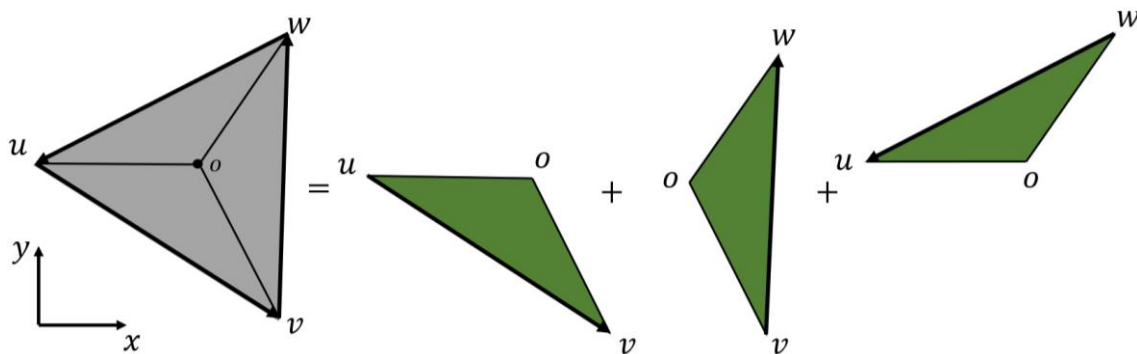


Abbildung 22: Zusammensetzung der Einzelflächen, wenn sich der zusätzliche Punkt o im Gebiet befindet

Es muss der Umlaufsinn eingehalten werden (Pfeile in Abb. 22), damit sich für die orientierten Flächeninhalte der Einzelflächen das richtige Vorzeichen ergibt. Wenn sich der Punkt außerhalb des Gebiets befindet, setzt sich der Flächeninhalt aus der grünen, positiven Gesamtfläche und den roten, negativen Teilflächen zusammen (s. Abb. 23).

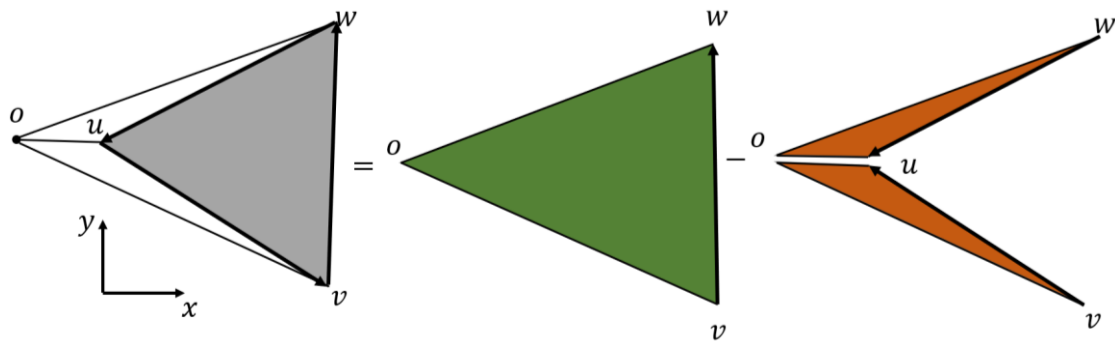


Abbildung 23: Zusammensetzung der Einzelflächen, wenn sich der Punkt außerhalb des Gebiets befindet

Auch hier ist der Umlaufsinn einzuhalten, da sich sonst die Vorzeichen ändern. Die Einzelflächen lassen sich auch in diesem Fall aus den Determinanten des Punktes o und der zwei jeweiligen anderen Punkte berechnen, somit ergibt sich für den Flächeninhalt des gesamten Dreiecks (Gl 3.2):

$$F_{\Delta} = \frac{1}{2} \det[o, u, v] + \frac{1}{2} \det[o, v, w] + \frac{1}{2} \det[o, w, u] \quad (3.2)$$

Sobald sich ein Vorzeichenwechsel in Gleichung (3.2) ergibt, befindet sich der zu prüfende Punkt außerhalb des Gebiets. Für den dreidimensionalen Raum lässt sich die Methode gut auf Tetraeder übertragen. Wo im 2D-Raum orientierte Flächeninhalte bestimmt werden, werden im 3D orientierte Volumen der Teiltetraeder ermittelt. Das Volumen eines Tetraeders im Allgemeinen berechnet sich aus $1/6$ der Determinante, welche die drei Vektoren beinhaltet, die ein Parallelepiped aufspannen. Das gerichtete Volumen eines Tetraeders lässt sich erneut über einen zusätzlichen Punkt o bestimmen, von dem aus die gerichteten Volumen der Teiltetraeder gebildet werden (s. Abb. 24).

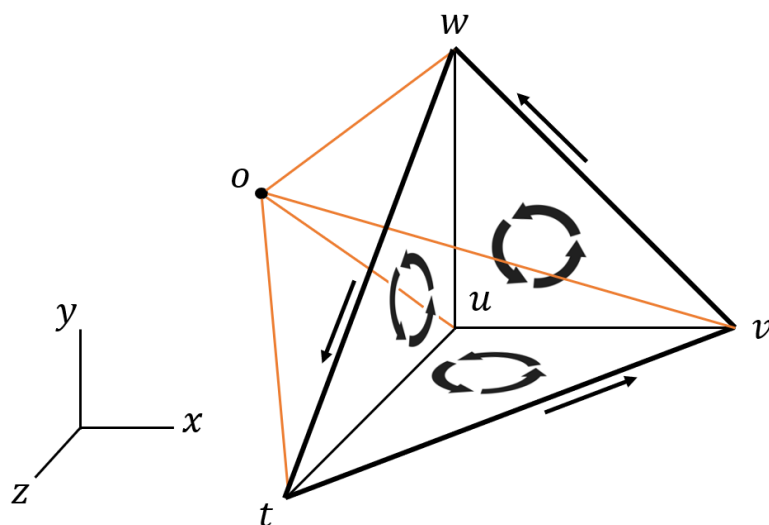


Abbildung 24: Tetraeder mit Punkt o , mit dem die gerichteten Volumen der Teiltetraeder gebildet werden

Die Summe der Teilvolumen ergibt demnach das Gesamtvolumen des Tetraeders, wobei der Umlaufsinn genau beachtet werden muss. Durch diesen erhalten die Teilvolumen automatisch das richtige Vorzeichen. Die Formel für das Volumen V_{Δ} lautet (Gl. 3.3):

$$V_{\Delta} = \frac{1}{6} (\det[o, v, w, t] + \det[o, w, t, u] + \det[o, t, v, u] + \det[o, v, w, u]) \quad (3.3)$$

Der zu untersuchende Punkt o befindet sich außerhalb des Tetraeders, wenn ein Vorzeichenwechsel in der Gleichung (3.3) vorkommt. Nachdem jeder Voxel auf die Elementzugehörigkeit untersucht wurde, kann dem Element der entsprechend gemittelte HU-Wert zugeordnet werden.

3.3.2 V2

In der Variante wird über das HU-Gebiet des Elements, für die Herleitung auch der einfache Tetraeder, integriert und durch das Volumen dessen geteilt, sodass sich der elementspezifische HU-Wert \overline{HU}_n ergibt (Gl. 3.4) [25]:

$$\overline{HU}_n = \frac{\int_{V_n} HU(x, y, z) dV}{\int_{V_n} dV} \quad (3.4)$$

Da die Integrale numerisch gelöst werden, bedarf es Stützstellen für die Integration, welche die HU-Werte an den Knoten u, v, w, t des Elements darstellen. Da die errechneten Werte an den Stützstellen repräsentativ für das ganze Element sind, müssen sich diese innerhalb dessen befinden, wofür das Element kleiner skaliert wird.

Die Skalierung des Elements wird über Formfunktionen realisiert. Formfunktionen finden Anwendung in der FEM, um das Verschiebungsfeld zu ermitteln. Darüber hinaus werden sie verwendet, um die Diskretisierung des Gebiets zu definieren. Dementsprechend wird auch die Geometrie der Elemente durch die Formfunktionen beschrieben. Die Formfunktionen sind Polynome, welche durch die Wahl des Elementtyps und der Ansatzfunktionen entstehen und sich durch diese unterscheiden. Die Ansatzfunktion entspricht dem allgemeinen Begriff für die Funktion der Beschreibung des Verschiebungsfeldes bzw. der Geometrie zwischen den bekannten Geometriepunkten. Die Ansatzfunktionen können mit unterschiedlicher Ordnung definiert werden. Ein linearer Ansatz verbindet die Punkte linear, ein quadratischer Ansatz quadratisch. Demzufolge hat ein Tetraederelement mit linearem Ansatz 4 Eckpunkte bzw. Knoten (wie in Abb. 24), wogegen ein Tetraederelement mit quadratischem Ansatz für die Definition zwischen den 4 Eckpunkten zusätzlich Mittelknoten benötigt, wodurch ein Element mit 10 Knoten entsteht. Für jeden Punkt im Element gibt es eine Formfunktion, welche allerdings in lokalen Koordinaten

definiert ist, damit eine universelle Anwendung auf alle im Raum befindlichen Elemente möglich ist. Anderweitig müsste für jedes Element im Raum eine eigene Formfunktionen definiert werden, was praktisch nicht umsetzbar ist. Die lokalen Koordinaten werden zur Unterscheidung als r, s, t oder ξ, η, ζ beziffert (s. Abb. 25).

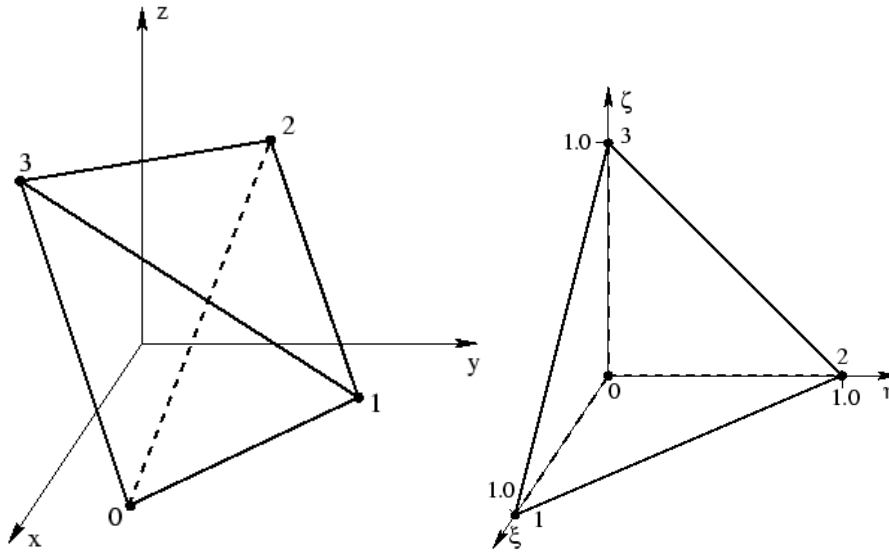


Abbildung 25: allgemeiner Tetraeder im x, y, z -Raum (links) und der Einheitstetraeder im ξ, η, ζ - bzw. r, s, t -Raum (rechts) [32]

Die Geometriedefinition der Elemente mittels der Formfunktionen in lokalen Koordinaten ergibt sich so zu (Gl. 3.5) [26] :

$$\tilde{x}_e = \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix}_e = \begin{bmatrix} N_{1(r)} & 0 & 0 & N_{2(r)} & 0 & 0 & \dots \\ 0 & N_{1(r)} & 0 & 0 & N_{2(r)} & 0 & \dots \\ 0 & 0 & N_{1(r)} & 0 & 0 & N_{2(r)} & \dots \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ x_2 \\ y_2 \\ z_2 \\ \vdots \end{bmatrix} = N_{(r)} x_e \quad (3.5)$$

Dabei ist \tilde{x}_e die Elementgeometrie, welche sich aus der Matrix-Vektormultiplikation der Matrix $N_{(r)}$ und dem Vektor x_e ergibt. $N_{(r)}$ beinhaltet die Formfunktionen des jeweiligen Punktes in lokalen Koordinaten und x_e beinhaltet die globalen Koordinaten der jeweiligen Punkte, wobei in Gleichung (3.5) $N_{(r)}$ und x_e für die ersten 2 Knoten angegeben ist, die Indizierung aber bis zur Anzahl der Knoten pro Element geht.

Für die Skalierung des Tetraeders bedeutet dies, dass sich die neuen skalierten Koordinaten des kleineren Tetraeders folgend berechnen lassen (Gl. 3.6 – 3.8):

$$\tilde{x} = \sum_{i=1}^n N_i * x_i \quad (3.6)$$

$$\tilde{y} = \sum_{i=1}^n N_i * y_i \quad (3.7)$$

$$\tilde{z} = \sum_{i=1}^n N_i * z_i \quad (3.8)$$

Die Koordinaten $\tilde{x}, \tilde{y}, \tilde{z}$ entsprechen den neuen skalierten, globalen Koordinaten der Elementknoten sind, n ist die die Anzahl der Knoten pro Element und x_i, y_i, z_i sind die jeweiligen globalen Koordinaten der Ausgangsknoten des Elements.

Durch die Transformation in lokale Koordinaten wird für die Ausgangsgleichung (3.4) noch ein Transformationsfaktor benötigt, welcher der Determinante der Jakobimatrix der Formfunktionen entspricht. Die Einträge der Matrix beinhalten die partiellen Ableitungen der globalen Koordinaten nach den lokalen Koordinaten bzw. die Ableitung der Gleichungen (3.6) bis (3.8) nach den lokalen Koordinaten (Gl. 3.9):

$$detJ = \det \begin{bmatrix} \frac{\partial \tilde{x}}{\partial r} & \frac{\partial \tilde{y}}{\partial r} & \frac{\partial \tilde{z}}{\partial r} \\ \frac{\partial \tilde{x}}{\partial s} & \frac{\partial \tilde{y}}{\partial s} & \frac{\partial \tilde{z}}{\partial s} \\ \frac{\partial \tilde{x}}{\partial t} & \frac{\partial \tilde{y}}{\partial t} & \frac{\partial \tilde{z}}{\partial t} \end{bmatrix} \quad (3.9)$$

Beispielhaft mit (Gl. 3.10):

$$J_{11} = \frac{\partial \tilde{x}}{\partial r} = \frac{\partial N_1}{\partial r} x_1 + \frac{\partial N_2}{\partial r} x_2 \dots = \sum_{i=1}^n \frac{\partial N_i}{\partial r} x_i \quad (3.10)$$

Die Gleichung (3.4) ändert sich somit zu (Gl. 3.11):

$$\widetilde{HU}_n = \frac{\int_{V_n} HU(r, s, t) detJ(r, s, t) dV}{\int_{V_n} dV} \quad (3.11)$$

Wenn die Integrationspunkte ermittelt wurden, müssen diesen noch entsprechende HU-Werte zugeordnet werden. Jedem berechneten Punkt $(\tilde{x}, \tilde{y}, \tilde{z})$ können 8 Gitterpunkte aus dem CT zugeordnet werden, welche durch die DICOM Eigenschaften *PixelSpacing* und *SliceThickness* definiert sind (s. Abb. 26)

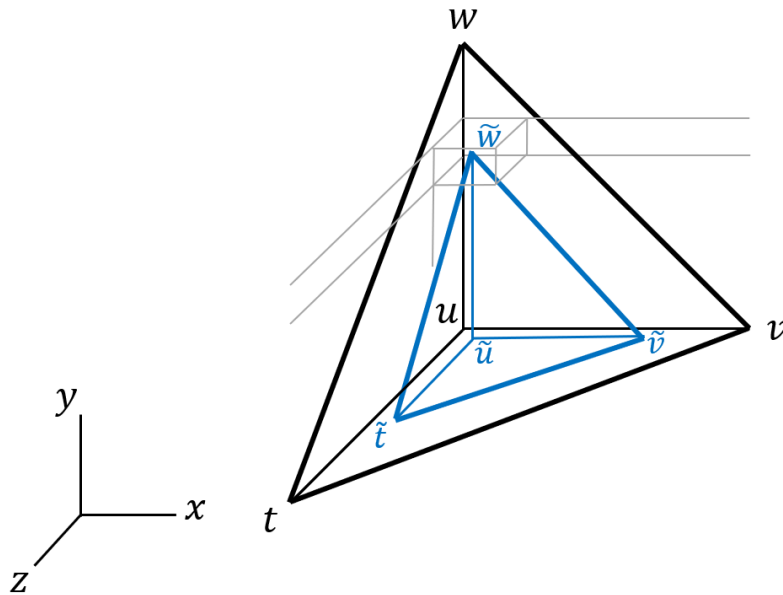


Abbildung 26: Tetraeder mit originalen Eckpunkten u, v, w, t und den skalierten Eckpunkten $\tilde{u}, \tilde{v}, \tilde{w}, \tilde{t}$, welche den Integrationspunkten entsprechen

Der Ausgangstetraeder mit dem skalierten Punkt \tilde{w} liegt im Gitter des CT-Scans (angedeutet mit den grauen Linien). Die umliegenden Gitterpunkte sind allerdings als Pixel- bzw. Voxelmittelpunkte definiert, was bedeutet, dass jeder der 8 Eckpunkte C_1 bis C_8 einen eigenen HU-Wert besitzt (s. Abb. 27).

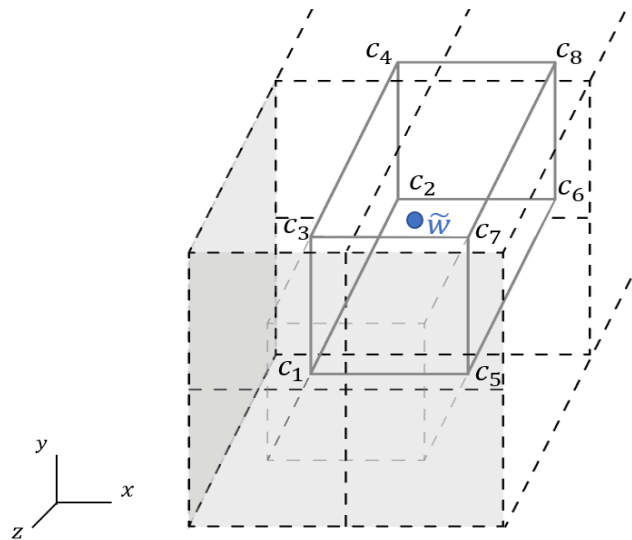


Abbildung 27: Knotenpunkt \tilde{w} im CT-Gitter (gestrichelte Linien) mit den 8 umgebenden Voxelmittelpunkten C_1 bis C_8

Für den Knotenpunkt \tilde{w} wird über trilineare Interpolation ein Wert, abhängig von der relativen Position $\Delta x, \Delta y, \Delta z$ in der „Voxelbox“ und den HU-Werten der 8 Voxel, bestimmt (s. Abb. 28)

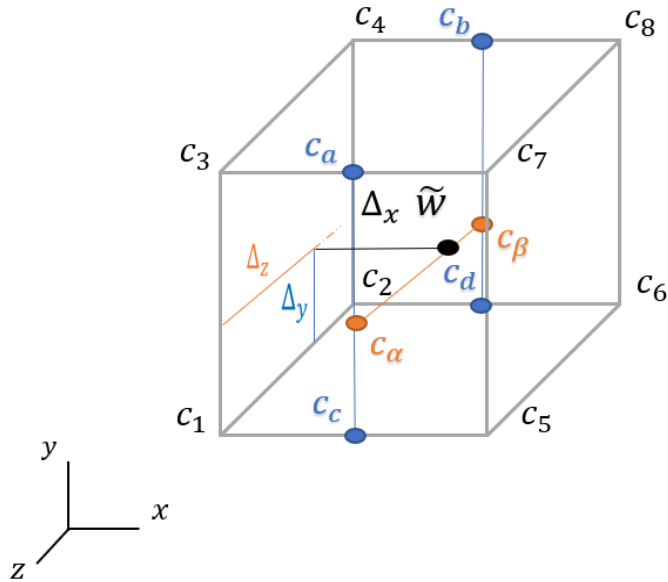


Abbildung 28: trilineare Interpolation in der „Voxelbox“, um ein HU-Wert für den Eckknoten \tilde{w} zu berechnen

Die relativen Positionen Δx , Δy , Δz in Bezug auf den unteren, linken, vorderen Gitterpunkt C_1 berechnen sich zu (Gl. 3.12 – 3.14):

$$\Delta x = \frac{x_{\tilde{w}} - x_{C_1}}{x_{C_5} - x_{C_1}} = \frac{x_{\tilde{w}} - x_{C_1}}{Lx} \quad (3.12)$$

$$\Delta y = \frac{y_{\tilde{w}} - y_{C_1}}{y_{C_3} - y_{C_1}} = \frac{y_{\tilde{w}} - y_{C_1}}{Ly} \quad (3.13)$$

$$\Delta z = \frac{z_{\tilde{w}} - z_{C_1}}{z_{C_2} - z_{C_1}} = \frac{z_{\tilde{w}} - z_{C_1}}{Lz} \quad (3.14)$$

Lx, Ly, Lz bilden die Breite, Höhe und Länge der „Voxelbox“, welche die gleichen Werte wie die DICOM-Informationen *PixelSpacing* und *SliceThickness* besitzen. Die Werte x_i, y_i, z_i sind die Koordinaten des Punktes \tilde{w} . Die Interpolation erfolgt zuerst in X-, dann in Y- und zum Schluss in Z-Richtung (Gl. 3.15):

$$c_a = \Delta x(c_7 - c_3) + c_3 \quad (3.15)$$

$$c_b = \Delta x(c_8 - c_4) + c_4$$

$$c_c = \Delta x(c_5 - c_1) + c_1$$

$$c_d = \Delta x(c_6 - c_2) + c_2$$

$$c_\alpha = \Delta y(c_a - c_c) + c_c$$

$$c_\beta = \Delta y(c_b - c_d) + c_d$$

$$HU\tilde{w} = \Delta z(c_\beta - c_\alpha) + c_\alpha$$

In diesem Fall stellen die Werte c_i die HU-Werte der Voxel C_1 bis C_8 bzw. die Zwischenwerte der Interpolationsschritte dar und $HU\tilde{w}$ entspricht dem schlussendlichen HU-Wert, der dem Knotenpunkt des Elements zugeordnet wird. Dies wird für alle Punkte des Elements durchgeführt, sodass 10 Stützstellen (quadratischer Tetraeder) für die numerische Lösung des Integrals im Zähler der Gleichung (3.11) vollständig bekannt sind.

Numerische Integrale werden in der Regel aus der Summe gewählter Funktionswerte multipliziert und mit einem Wichtungsfaktor berechnet. Im zweidimensionalen ist dies beispielsweise die Länge zwischen den gewählten Stützpunkten, welche vorzugsweis gleichmäßig lang sind, sodass ein konstanter Wichtungsfaktor entsteht. Bei der Integration über dem Gebiet, in Form eines FE-Elements, ist der Faktor jedoch zu jeder Stützstelle unterschiedlich. Er entspricht der Determinante der Jakobimatrix, welche repräsentativ dafür ist, wie sich der betrachtete Punkt des globalen Elements in Bezug auf das lokale Element verändert (s. Abb. 29).

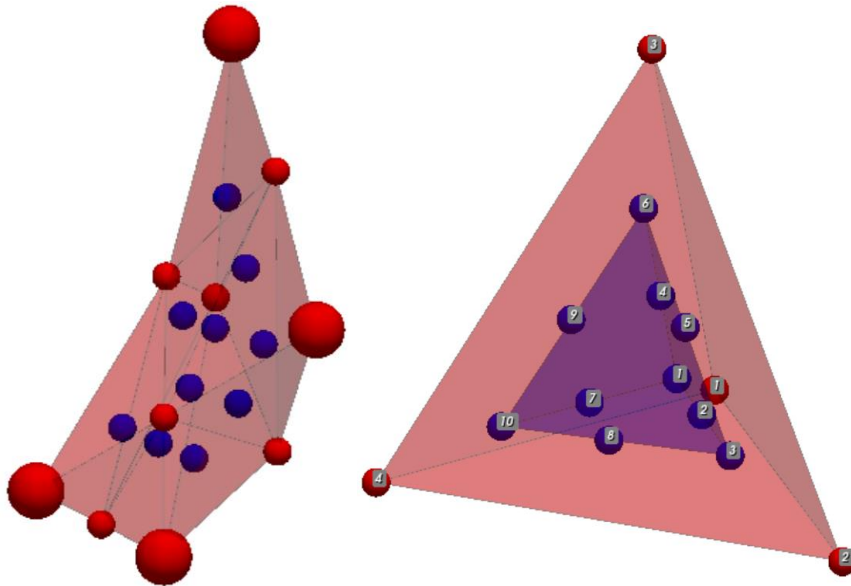


Abbildung 29: FE-Element als globaler Tetraeder (links) mit Integrationsstellen (blau) und lokaler Referenztetraeder (rechts) mit Integrationspunkten und kleiner skaliertem Tetraeder (blau)

Wobei die lokalen Koordinaten so gewählt sind, dass der skalierte Tetraeder $1/8$ des Volumens des Originaltetraeders einnimmt. Die numerische Notation der Gleichung (3.4) lautet demnach (Gl. 3.16):

$$\widetilde{HU} = \frac{\sum_{k=1}^{10} HU_{(k)} \det J_{(k)} \left(\frac{1}{6}\right)}{\sum_{k=1}^{10} \det J_{(k)} \left(\frac{1}{6}\right)} \quad (3.16)$$

Hierbei steht k für den jeweiligen Punkt des Elements. Das noch fehlende Volumen des Tetraeders im Nenner wird numerisch über das Volumenintegral berechnet, welches sich aus der Summe der Determinanten der Matrix J und dem Faktor $1/6$ (Tetraeder-Element) ermitteln lässt.

3.3.3 V3

Diese Variante entspricht von der Vorgehensweise der Variante 2, jedoch wird nicht über dem HU-Feld integriert, sondern über dem Skalar-Feld des E-Moduls. Dafür wird erst jeder HU-Wert der Voxel über die Dichte-Korrelationen aus Kapitel 2 in einen E-Modul-Wert umgewandelt und folgend über den Algorithmus von V2 für jedes Element ein Skalar berechnet.

3.3.4 Vergleich der Varianten

In Anbetracht der sich unterscheidenden Ergebnisse der aufgeführten Varianten gilt es, die beste Variante zu ermitteln. Variante 1 stellt zwar die am wenigsten aufwendige Variante dar, hat jedoch einen bedeutenden Nachteil.

In der FEM werden verschieden Arten von Elementen verwendet, welche unterschiedliche Eigenschaften aufweisen. Eine mögliche Unterteilung ist die Anzahl der Eckknoten, ob es sich beispielsweise um einen Tetraeder (4 Knoten) oder Hexaeder (8 Knoten) handelt. Eine andere Unterscheidung ist die Wahl der Ansatzfunktionen, welche maßgeblich für die Qualität der approximierten Geometrie und dem zu ermittelnden Verschiebungsfeld verantwortlich sind. Die Herleitungen in Kapitel 3 erfolgten am Beispiel eines hier sogenannten einfachen Tetraeders, welcher lineare Ansatzfunktionen verwendet. Jedoch kann die Wahl von Ansatzfunktionen höheren Grades zu genaueren Ergebnissen führen, da die Geometrie des Gebiets besser approximiert wird und das zu ermittelnde Verschiebungsfeld nicht mehr linear, sondern quadratisch (bei quadratischer Ansatzfunktion) oder höher angenähert wird (s. Abb. 30).

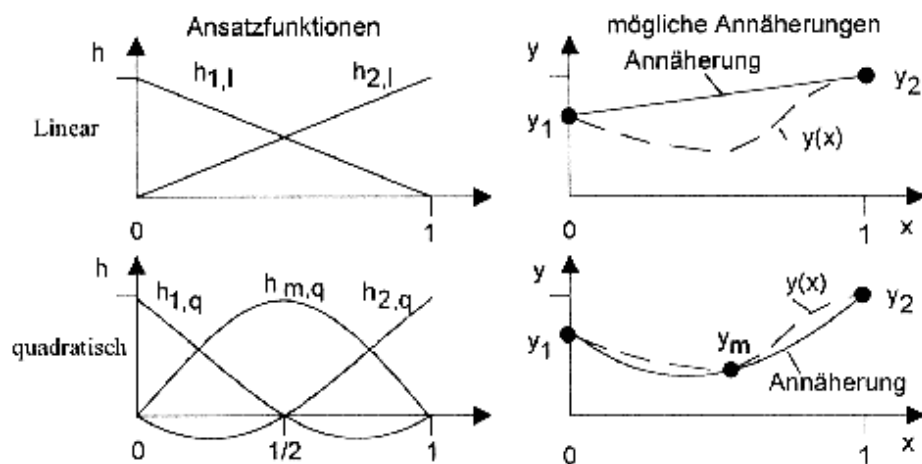


Abbildung 30: lineare und quadratische Ansatzfunktionen in 1D und die zugehörige Lösungsnaherung [25]

Für Ansatzfunktionen höherer Ordnung werden zusätzliche Knoten auf dem Element benötigt, was die Anzahl der Knoten bei einem quadratischen Tetraeder auf 10 erhöht. Da die Diskretisierung, und damit die Wahl des Elementtyps und der Ansatzfunktionen vor dem Materialmapping durchgeführt wird, muss der Algorithmus elementspezifisch funktionieren oder er ist auf einen Elementtyp begrenzt. Bei Variante 1 ist es mathematisch schwer umsetzbar, herauszufinden, welche Elemente sich innerhalb des Gebiets befinden, wenn es sich nicht um lineare Elemente handelt. Jede Elementseite müsste zusätzlich trianguliert werden, wobei die vorher definierten zusätzlichen Mittelknoten verwendet werden müssten. Somit begrenzt sich die Variante 1 auf lineare Elemente. Da in Variante 3 und 4 die Ansatzfunktionen verwendet werden, ist der Algorithmus für alle Elemente der gleiche, es müssen lediglich die jeweiligen Formfunktionen implementiert sein, womit in diesen Algorithmen keine Beschränkung des Elementtyps vorliegt. F. Taddei [25] untersuchte den Einfluss der Mapping-Algorithmen von V1 und V2 auf die Unterschiede in den Ergebnissen der zugeordneten Materialien und FEM-Analysen, in denen die Verformung und die Spannung ausgewertet wurden. V2 wurde als die aufwendigere Methode eingestuft, welche jedoch akkuratere Ergebnisse, explizit bei feiner Vernetzung (Elemente ggf. kleiner als CT-Voxel), liefert. In der Studie [27] werden des Weiteren V2 und V3 miteinander verglichen und anhand von mechanischen Tests des Femurs validiert, wobei als Vergleichsgröße die experimentellen und numerisch berechneten Spannungen und Dehnungen ermittelt wurden. Das Bestimmtheitsmaß der Korrelationen von experimentellen und numerisch bestimmten Spannungen sind bei beiden Varianten größer als 0.9 und weisen nur geringe Unterschiede auf, während V3 Ergebnisse liefert, die näher an den Experimentellen liegen. Bei dem Vergleich der Dehnung ist ein stärkerer Unterschied der Versionen zu beobachten, wobei auch V3 die genaueren Ergebnisse liefert. Aufgrund der flexibleren Auswahl an Elementtypen und den genaueren gelieferten Ergebnissen wurde V3 als Algorithmus zur Materialzuweisung gewählt.

3.4 Programmaufbau und -ablauf

Die Vorlage für das Programm, welches im Zuge dieser Arbeit entsteht, ist der Code von H. Anan [3], welcher das Programm von Elise C. Pegg [28] so umschrieb, dass eine ANSYS-Datei erzeugt wurde, der grundlegende Datenfluss des Berechnungsalgorithmus aber unverändert blieb.

Das Programm ist in unterschiedliche Module bzw. Dateien aufgeteilt (Namen der Python-Dateien des Programmaufbaus folgend in **fett** geschrieben), in denen die unterschiedlichen Abschnitte, wie Datenimport, Berechnung, Datenexport etc. sortiert sind (s. Abb. 31).

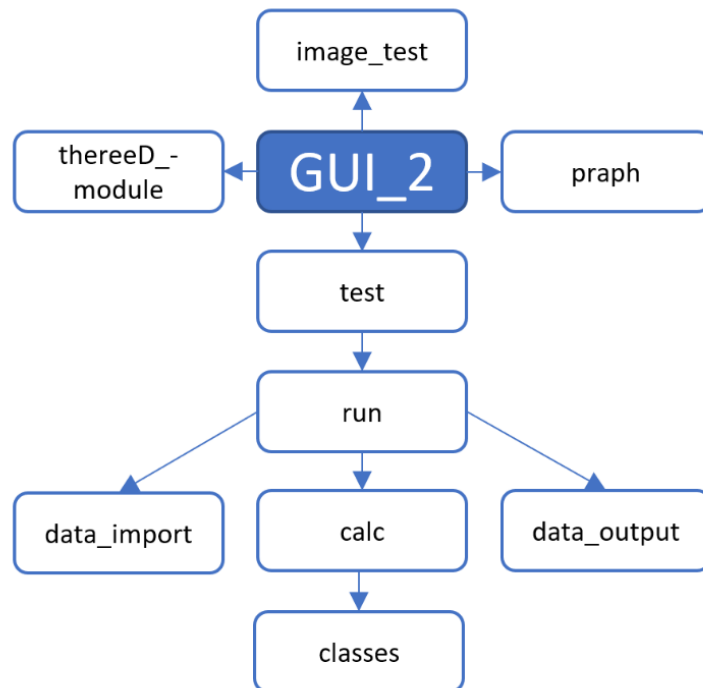


Abbildung 31: Aufbau und Verweisung der Python-Dateien des Programms

GUI_2

In diesem Modul befindet sich der Code für die Benutzeroberfläche, wenn das Programm gestartet wird. Von dieser Datei aus werden die Benutzereingaben entgegengenommen und die Information weitergeleitet bzw. abgespeichert und sobald alle benötigten Eingaben getätigt sind, das Programm gestartet.

image_test

In der Datei werden alle nötigen Daten extrahiert, um einen 3D-Plot des CT-Scans und der CDB-Datei zu erstellen.

praph

Die sich öffnende Kalibrierungsoberfläche und die Berechnungen der Kalibrierung ist in dieser Datei hinterlegt.

thereeD_module

In der Datei ist der DICOM-Slicer hinterlegt.

test

Die vorher abgespeicherten Pfade für die CT-Daten, die CDB-Datei und die Benutzereingaben (HU-Parameter für Korrelationen) werden hier zusammen in einer Funktion abgerufen. Für Testzwecke kann durch Ausführen der Datei das Programm gestartet werden, ohne die Eingaben in der Benutzeroberfläche zu tätigen.

run

In dem Modul wird der eigentliche Datenfluss gesteuert. Nacheinander werden die Speicherpfade abgerufen und die Dateien verarbeitet und anschließend die Output-Datei geschrieben. Zuerst erfolgt der Datenimport der CDB-Datei, dann der Import der CT-Daten, anschließend die Berechnung der E-Moduln und der Dichten für die Elemente und das Erstellen der neuen CDB-Datei mit den Materialwerten. Zudem wird die Zeitmessung an dieser Stelle gestartet und eine Überprüfung der eingegebenen und verlinkten Daten durchgeführt. Für die Auswertung der berechneten Daten werden zudem mehrere Text-Dateien geschrieben, die im selbigen Speicherort abrufbar sind und Rohdaten wie die ermittelten E-Moduln, Dichtewerte, Werte der Kalibrierung etc. beinhalten.

data_import

Die importierten Daten (CT, CDB, Parameter) werden in diesem Modul bearbeitet und in die jeweils benötigte Form gebracht. Die DICOM-Daten können mit dem *Pydicom*-Package eingelesen und ausgewertet werden, die CDB-Datei dagegen wird als Textdatei verarbeitet.

calc

Die Zuordnung der einzelnen Werte des E-Moduls und der Dichte für jedes Element erfolgt in dieser Datei, dabei wird auch kontrolliert, ob sich die Geometrie der CDB-Datei innerhalb des CT-Raums befindet. Außerdem werden die Korrelationen für die Berechnung der Materialkennwerte, anhand der Eingabeparameter, definiert.

classes

In der Datei sind die Klassen hinterlegt, mit denen das Programm arbeitet. Die CT-Daten und jede einzelne Geometrie der CDB-Datei sind mit den jeweiligen Eigenschaften als Objekt gespeichert. In dem Modul sind zudem die Notationen für die Gleichungen aus

dem Kapitel *HU-Korrelationen* hinterlegt und es erfolgt der eigentliche Berechnungsschritt, welcher den jeweiligen E-Modul und die Dichte für jedes Element berechnet.

data_output

Die neue CDB-Datei mit den Materialwerten wird an dieser Stelle erstellt und am hinterlegten Pfad abgespeichert. Zudem erfolgt an der Stelle die Gruppierung der Materialwerte. Jedes Element erhält individuelle Werte für E-Modul und Dichte, was bei einer großen Anzahl an Elementen auch zur gleichen Anzahl an Materialwerten führt. Für folgende Festigkeitsberechnungen ist dies jedoch nicht zielführend und würde die Berechnungsdauer und Dateigröße erhöhen. Die Materialwerte werden aufgrund dessen mit einem definierten Wert, der die Bandbreite der Gruppierung beschreibt, zusammengefasst. Der Bereich, in welchem der Materialmittelwert erstellt werden soll, kann vom Benutzer in der Startoberfläche eingegeben werden. Vom höchsten Wert aus wird dieser Wert abgezogen und alle Materialwerte innerhalb werden zu einem zusammengefasst, womit auch alle Elemente in diesem Bereich die gleichen Materialwerte erhalten.

Programmablauf

Zuerst werden die Benutzereingaben und Parameter in einem Dictionary gespeichert, folgend werden die CT-Daten und die CDB-Datei ausgewertet. Die CT-Daten werden nach der Z-Koordinate sortiert und die benötigten Daten jedes Schnittbildes in einer Liste gesichert. Diese beinhaltet die Anzahl der Pixel in x und in y, den Pixelabstand in x, y und z (DICOM Tag *PixelSpacing* und *SliceThickness*), die Koordinaten des ersten Pixels (DICOM Tag *ImagePosition*), die Orientierung des Bildes in Bezug auf x und y Ebene (DICOM Tag *ImageOrientation*), und die linearen Koeffizienten, um die Pixeldaten in HU-Werte umzurechnen (DICOM Tag *RescaleSlope* und *RescaleIntercept*). Die Daten werden in passende Strukturen umgewandelt und in einem Objekt gespeichert, welches die Information der Voxelkoordinaten des CT-Gitters, die zugehörigen HU-Werte und die Dimensionen des CT-Gitters als Attribute beinhaltet.

Als nächstes wird die CDB-Datei verarbeitet, indem für jeden verwendeten Elementtyp ein Objekt angelegt wird, das die Information Partindex, Elementtyp und die ANSYS spezifische Elementnummer beinhaltet. In der Liste *elements*, welche ein Attribut des jeweiligen Elementobjekts darstellt, wird für jedes einzelne Element ein Objekt angelegt, das die Informationen der Elementindizes, Knotenindizes und Koordinaten der Knoten speichert.

Nachdem die benötigten Daten importiert und in die richtigen Strukturen umgewandelt sind, wird der Berechnungsschritt für jedes Element durchgeführt. Das Ergebnis ist eine Liste, in der für jedes Element der E-Modul und die Röntgendichte gespeichert ist, wobei die Reihenfolge der Listeneinträge der Reihenfolge der Elemente im Elementblock der CDB-Datei entspricht.

Nachfolgend wird die Gruppierung der Materialwerte durchgeführt. Die Liste der Materialeinträge wird der Größe nach sortiert, und der definierte zusammenzufassende Bereich wird vom größten Wert aus gemessen. Aus den in diesem Bereich liegenden Werten wird der Mittelwert gebildet und eine neue Liste der gruppierten Werte mit der gleichen Länge der Einzelwertliste erstellt. (s. Abb. 32).

```
gap = 2000
Modul_integ = [1000,2000,3000,2500,4000]
Modull_sortiert = [4000,3000,2500,2000,1000]
Modul_mittel = [2857,1000]
Modul_grupp = [2857,2857,2857,2875,1000]
```

Abbildung 32: beispielhafte Veränderung der Materialeinträge vom individuellen Elementwert (*Modul_integ*) zum gruppierten Wert (*Modul_grupp*) mit dem Bereichsparameter *gap*

Diese Vorgehensweise wird für den E-Modul und für die Dichte angewandt. Die Werte aus der Liste *Modul_mittel* werden in einem Dictionary mit neu erstelltem Materialindex als Key gespeichert. In einer If-Schleife wird anschließend für jeden Wert der Liste *Modul_grupp* der Materialwert aus dem Dictionary abgefragt, um den neuen Materialindex zu erhalten. Mit den gruppierten Materialwerten und den neuen Materialindizes wird folgend die neue CDB-Datei erstellt und am angegebenen Speicherort abgelegt. Die Datei muss in ANSYS über ein externes Modell eingelesen werden und mit dem statisch-mechanischen Analysesmodul verknüpft werden (s. Abb. 33).

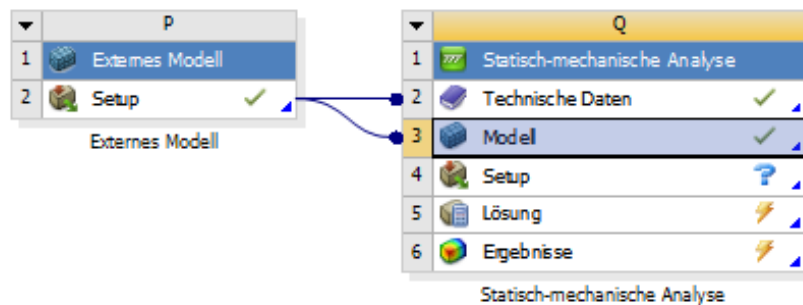


Abbildung 33: Import der gemappten CDB-Datei in ANSYS

Dabei müssen zwingend die eingestellten Einheiten beachtet werden. Die Einheiten des E-Moduls und der Dichte werden in der Output-Datei in MPa und in g/cm³ angegeben, somit muss beim Import in ANSYS die Einheitsskala auf *t, mm* gestellt werden.

Um lange Ladezeiten von über 24h zu verhindern, muss in den Importeinstellungen der Button *Geometrie erstellen* deaktiviert werden. Bei Aktivierung wird für jedes einzelne Element ein eigener Körper erstellt, was bei einer hohen Anzahl an Elementen zu extremen Ladezeiten führt.

4. Änderungen und Benutzeroberfläche

Für die Verbesserung und Funktionstüchtigkeit des Programms wurden einige Stellen im Code geändert oder neu erstellt.

Ansysis Output

Die Output-Datei des unveränderten Programms war nicht vollständig, sodass ANSYS diese nicht einlesen konnte (s. Abb. 34).

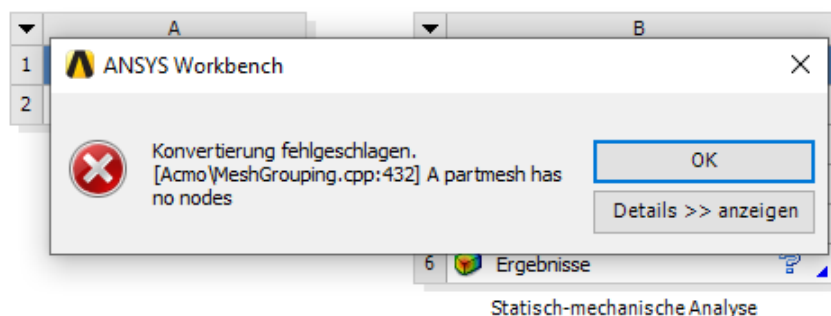


Abbildung 34: Fehlermeldung beim Importieren der gemappten Datei des ungeänderten Programms

Grund der Fehlermeldung war der nicht vorhandene Header des Knotenblocks, welcher durch das Erstellen einer Funktion behoben wurde (s. Abb. 35).

```
def _get_Nblock():
    #get the NBLOCK head
    os.chdir(sys.path[0])
    with open("pathcdb.txt", "r") as f:
        pathcdb = f.read()
    with open(pathcdb, "r") as f:
        for row in f:
            if "NBLOCK" in row:
                nline = row
                break
    return nline
```

Abbildung 35: Funktion, welche den Header des Knotenblocks identifiziert und zurückgibt

Zuordnung Elementtyp

Für jeden verwendeten Elementtyp, der bei der Vernetzung der Geometrie festgelegt wird, wird im Programm ein Objekt angelegt, welches die Informationen aus der CDB-Datei speichert. Bei der Verwendung unterschiedlicher Elementtypen innerhalb einer Geometrie, wurde die gemappte CDB-Datei mit unvollständigen und falschen Informationen erstellt, was auf einen Fehler im Datenfluss beim Extrahieren der zugehörigen Elemente je Elementtyp zurückzuführen ist. In ANSYS wird jedem Element eine

Typnummer zugeordnet, welche in APDL verwendet wird. Quadratische Tetraederelemente besitzen die Nummer 187 und lineare Hexaeder die 185. Im Programm wurde ein Elementtyp überschrieben und nur dieser gespeichert, was beim Einlesen der gemappten Datei zur fehlerhaften Zuordnung der Elemente führte (s. Abb. 36).

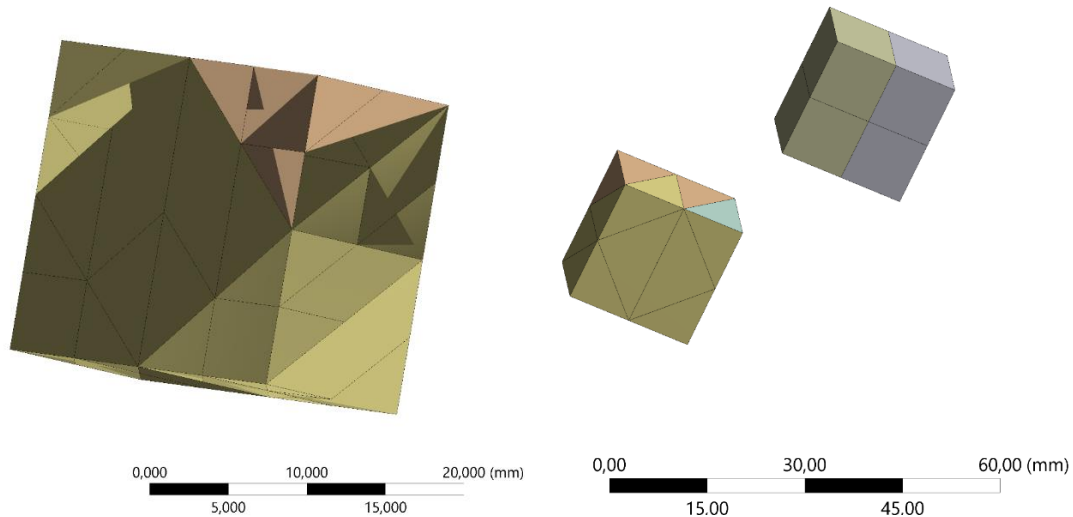


Abbildung 36: falsche Zuordnung der Elementtypen (links) und original Geometrie und richtige Zuordnung der Elemente (rechts)

Durch Ändern des Datenflusses in den Modulen *data_input* und *data_output* erfolgt im überarbeiteten Code die richtige Zuordnung der Elementtypen, sodass unterschiedliche Elementtypen verwendet werden können.

Die falsche Zuordnung der Elementtypen ist in der CDB-Datei an zwei Stellen bemerkbar. Die Initialisierung der verwendeten Elementtypen mithilfe der spezifischen Elementnummer erfolgt bei der originalen CDB-Datei direkt vor dem Knotenblock. In der gemappten CDB-Datei wurde diese zwischen Knoten- und Elementblock eingefügt, mit dem zusätzlichen Fehler, dass nicht alle verwendeten Elementnummern vorhanden waren. Durch eine zusätzliche For-Schleife und einen zusätzlichen Rückgabewert verschiedener Funktionen im *data_import* Modul werden im geänderten Code die richtigen Elementtypen ermittelt und gespeichert. Die Initialisierung der Elementtypen an der richtigen Stelle in der CDB-Datei wird durch Neuordnung der Elementnummern in der CDB-Datei realisiert (s. Anhang 2).

Es ist anzumerken, dass ein Vergleich solch einer Geometrie mit Bonemat nicht möglich ist, da dort anscheinend ein ähnliches Problem vorliegt und das Programm abstürzt, sobald unterschiedliche Elementtypen in einer Geometrie verwendet werden.

Koordinaten im Knotenblock

Im Knotenblock der originalen CDB-Datei sind u.a. die Koordinaten abgespeichert. Wenn eine Koordinate genau auf einer Achse liegt, wird diese nicht mit Null aufgefüllt, sondern weggelassen (s. Abb. 37).

```
NBLOCK,6,SOLID,      87,      87
(3i9,6e21.13e3)
1      0      0 0.000000000000E+000 1.000000000000E+001
2      0      0 0.000000000000E+000 1.000000000000E+001 5.000000000000E+000
3      0      0 0.000000000000E+000
4      0      0 0.000000000000E+000 5.000000000000E+000
```

Abbildung 37: Ausschnitt original CDB-Datei, bei der Knoten auf einer Achse liegen

Das Programm benötigt für die Berechnung jedoch immer exakt 3 Koordinaten, weshalb diese mit einer For-Schleife hinzugefügt wurden, wenn ein Knoten auf einer Achse liegt (s. Abb. 38).

```
for n in list(nodes.keys()):
    if len(nodes[n]) == 1:
        nodes[n].append(0.0)
        nodes[n].append(0.0)
    elif len(nodes[n]) == 2:
        nodes[n].append(0.0)
```

Abbildung 38: For-Schleife, welche die Koordinaten mit 0 auffüllt, falls diese nicht vorhanden sind

Partindizes

Im Elementblock der gemappten CDB-Datei wurden die Partindizes falsch gespeichert (s. Anhang 3). Der Code wurde so geändert, dass der Elementblock der gemappten CDB-Datei dem Originalen entspricht (bis auf Materialindizes).

Slice Thickness

Für die Berechnung der Koordinaten des CT-Gitters wurden die DICOM-Informationen *PixelSpacing* und *SliceThickness* genutzt, welche die Abstände der Voxelmittelpunkte im Raum beschreiben. Es kann jedoch durch ein Eingabefehler beim Erstellen des Scans vorkommen, dass die DICOM-Daten nicht mit den tatsächlich verwendeten Abständen übereinstimmen. Da dadurch das CT-Bild verzerrt wird, werden für die Elemente der Netzgeometrie falsche Werte ermittelt (s. Abb. 39).

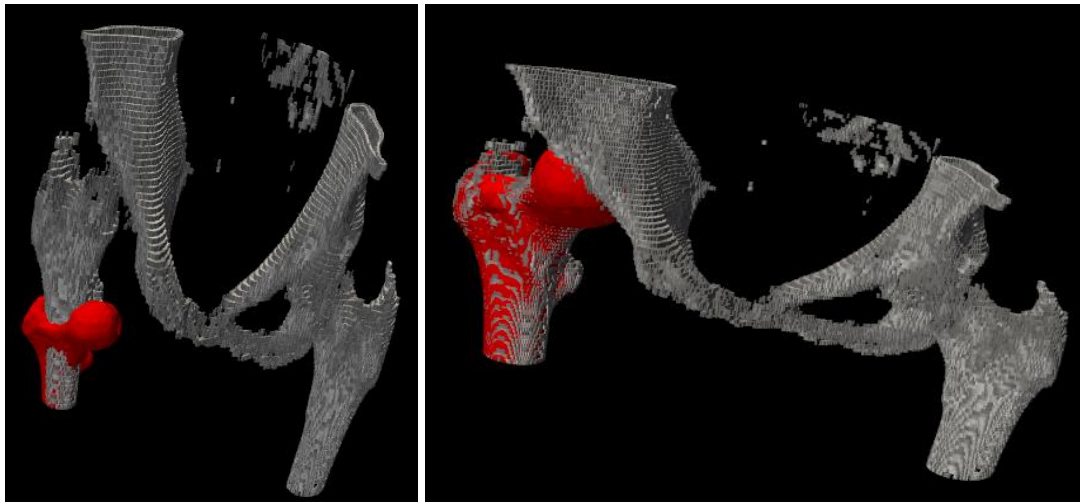


Abbildung 39: 3D-Plot der CT-Daten und der Netzgeometrie mit falschem Wert (links) und richtigem Wert (rechts) für den Voxelabstand in der z-Achse (Slice Thickness)

Die Quelle für den Wert der Voxelabstände in Z-Richtung wurde geändert, indem nicht mehr auf die mitgelieferte DICOM-Information zurückgegriffen wird, sondern auf die Differenz der Z-Koordinaten zweier Schnittbilder (s. Anhang 4).

XML-Parameter

Für das Programm im Ausgangszustand musste zusätzlich zu den CT-Daten und der CDB-Datei eine XML-Datei importiert werden, in denen die Parameter für die HU-Korrelationen und für die Kalibrierung hinterlegt waren. Der Hintergrund dafür ist, dass der Code eine Anpassung des Codes aus [28] ist und in diesem mit dem XML-Format gearbeitet wurde. Jedoch ist es einfacher, die verwendeten Korrelationsparameter direkt in der Benutzeroberfläche einzugeben, um diese nach Bedarf direkt ändern zu können. Zudem fällt durch das direkte Benutzen der Parameter eine zusätzliche Datei weg. Das Programm wurde so angepasst, dass alle nötigen Daten als Ausgangsdaten im Programm hinterlegt sind und bei Bedarf geändert werden können (s. Anhang 5).

Kalibrierfunktion

Eine der Hauptaufgaben war es, eine Kalibrierfunktion zu implementieren, wodurch die HU-Werte der CT-Daten mit der Dichte der Phantome abgeglichen werden. Die Korrelation der Röntgendichte ρ_{QCT} und der HU-Werte wird über lineare Regression ermittelt. In Bonemat sind standardgemäß lineare Parameter hinterlegt, welche auch geändert werden können, jedoch müssen die Parameter dafür bekannt sein. Da es unterschiedliche Phantome auf dem Markt gibt, entsteht für jedes einzelne eine individuelle Korrelation, welche Einfluss auf die weitere Berechnung des E-Moduls nimmt. Das Ermitteln der phantomspezifischen Korrelation von HU-Werten und Röntgendichte trägt somit zu einem genaueren Ergebnis bei und verhindert einen möglichen Drift der Dichtewerte, wofür die Dichte- und HU-Werte des Phantoms bekannt sein muss.

In der Benutzeroberfläche ist dafür ein Button angelegt, welcher die Kalibrieroberfläche öffnet, in der die Phantomdichten eingegeben und die Phantome für die Ermittlung der HU-Werte markiert werden können (s. Abb. 40).

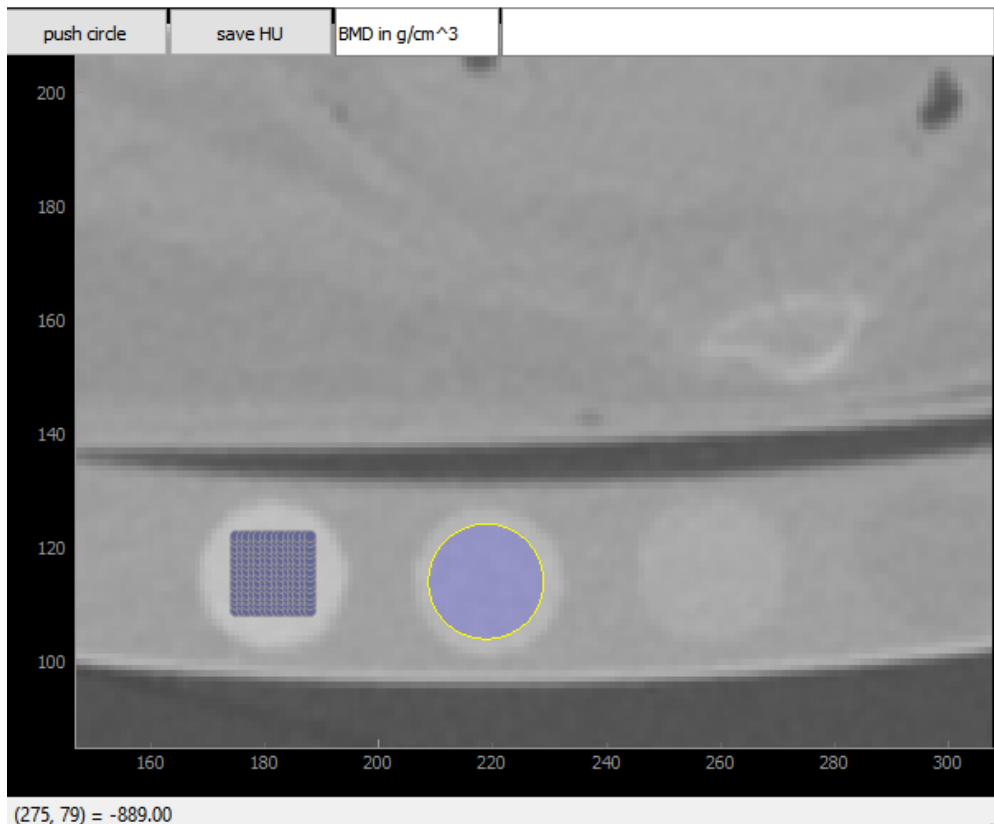


Abbildung 40: sich öffnendes Fenster für die Kalibrierung mit gelbem Kreis als Auswahltool und markierten Voxel

Im Fenster erscheint das Schnittbild, welches sich in der Mitte aller Aufnahmen befindet. Ein gelber Kreis kann als Auswahltool in der Größe geändert und verschoben werden. Wenn sich dieser auf dem Phantom befindet, muss der Knopf *push circle* gedrückt werden und die sich innerhalb des Kreises befindenden Pixel werden markiert. Für die Berechnung des mittleren HU-Wertes eines jeden Zylinders werden insgesamt je 7 Schnittbilder im gleichen Abstand, abhängig von der Gesamtanzahl der Slices, herangezogen. Das verwendete Phantom besteht aus 5 Zylindern mit definierter Dikaliumhydrogenphosphat-Dichte, welche äquivalent zur Röntgendichte ist. In der Reihenfolge, in der die Zylinder ausgewählt wurden, muss in dem Eingabefeld *BMD* diese Dichte in g/cm^3 und mit einem Komma getrennt eingegeben werden. Durch Drücken des *save HU*-Knopfes wird die Berechnung der Korrelation gestartet und als Diagramm dargestellt (s. Anhang 6). Der lineare Zusammenhang und das Bestimmtheitsmaß r^2 wird zudem im oberen rechten Feld der Oberfläche angezeigt.

Ausgabe der Röntgendichte

Im Programm wurde für die Berechnung des E-Moduls zwar die Röntgendichte benutzt, in der Ausgabe ist diese jedoch in die scheinbare Dichte umgerechnet worden. Da allerdings in der Medizin mit der Röntgendichte mehr interpretiert werden kann (z.B. Erkennen von Osteoporose), wurde als Ausgabe die Röntgendichte gewählt.

Unterscheidung Kortikalis Spongiosa

Die Korrelationen für die Umrechnung der verschiedenen Knochendichten sind für den gesamten HU-bzw. Dichtebereich gültig, wodurch keine Unterscheidung in kortikalem und spongiösem Knochen vorgenommen werden muss. Die Korrelationen, die für den E-Modul erforscht sind, gelten meistens nur für eine Region des Knochens. Für den Fall, dass Korrelationen in der Form $E = a * \rho_{Ash}^b$ für spongiösen und kortikalen Knochen existieren, ist eine Unterscheidung mit einem einstellbaren Grenzwert implementiert (s. Anhang 7).

ANSYS Versionen

Ansys hat mit fortlaufenden Softwareupdates auch den Aufbau der CDB-Datei geändert. Die bestehende Routine war auf Ansys 19.2 ausgelegt, was ein Einlesen der gemappten CDB-Datei in neuere ANSYS Versionen nicht zuließ. Damit auch mit neueren Versionen gearbeitet werden kann, muss der Startblock angepasst werden. Die Datei der bisherigen Routine enthielt lediglich die PREP7 Initialisierung. Für Ansys 19.2 hat dieser ausgereicht, jedoch muss für neuere Versionen der NOPR Command und die verwendete Ansys-Version initialisiert werden. Die Routine wurde geändert, sodass die Output-Datei den Command und die Version enthält, womit die erzeugte Datei mit Ansys 19.2, 2021 R1 und 2022 R2 kompatibel ist (s. Abb. 41).

```
/COM,ANSYS RELEASE 19.2      2023-01-19 11:51:22.996261  
/PREP7  
/NOPR
```

Abbildung 41: neuer Startblock mit variabler Ansys-Version und NOPR Command

Benutzeroberfläche

Die Benutzeroberfläche ist im Allgemeinen so aufgebaut, dass alle nötigen Funktionen in der Hauptebene vorhanden sind, um das Materialmapping durchzuführen. Zudem kann ein 3D-Plot der CT-Daten und der Netzgeometrie in der ersten Ansichtsebene erzeugt werden. Für den DICOM-Slicer, die Kalibrierung und für eine extra Ansicht der Netzgeometrie öffnen sich neue Fenster.

Die Funktionen hinter den grünen *Browse* und dem roten *Save* Button müssen durchgeführt werden, die Funktionen hinter den hellgrauen Buttons sind zusätzlich (s. Abb. 42).

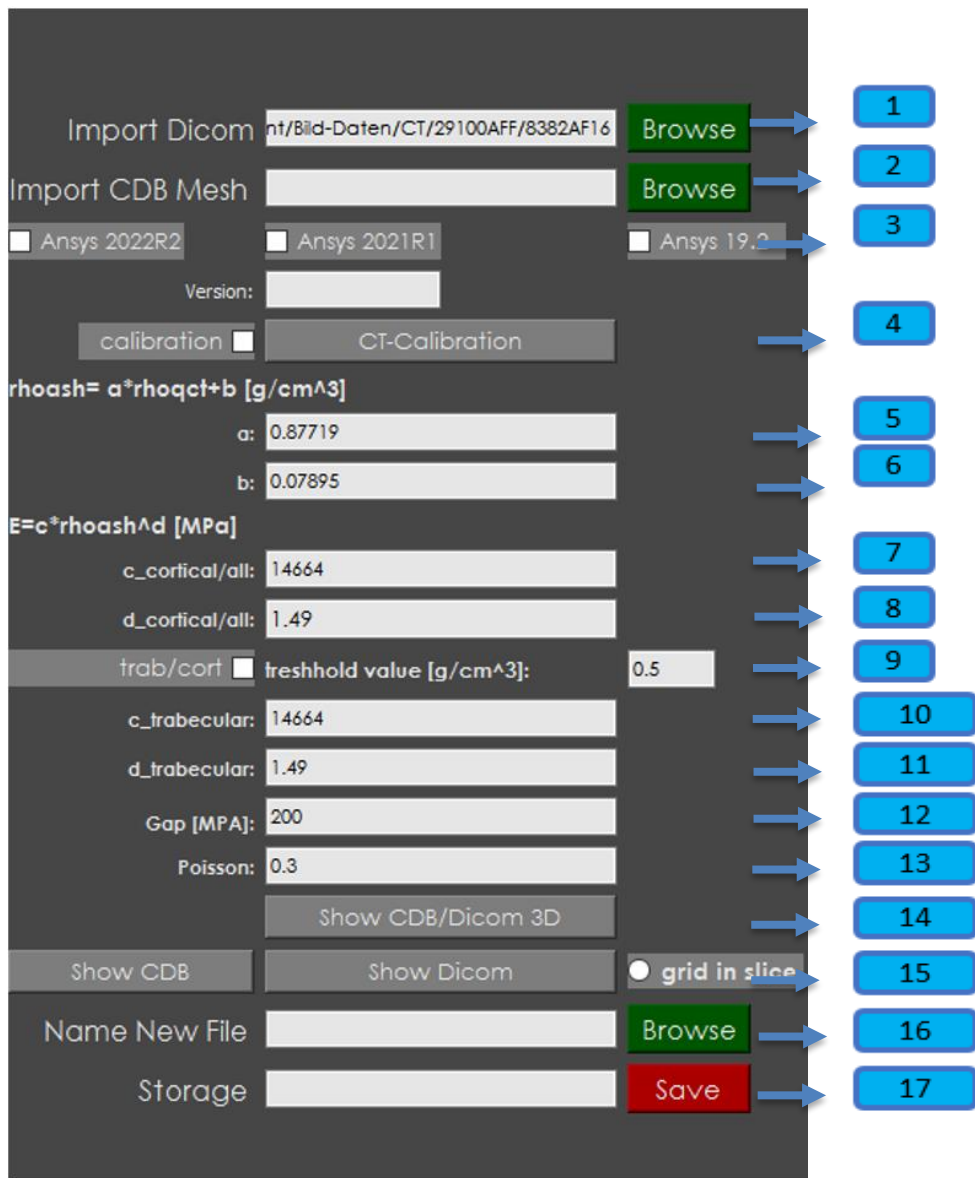


Abbildung 42: Hauptbenutzeroberfläche bei Start des Programms, wobei die Flächen für die 3D-Darstellungen nicht mit abgebildet sind (s. Anhang 8)

Die schon ausgefüllten Zeilen beinhalten die Standardwerte, welche genutzt werden, falls keine Änderungen vorgenommen wird. Dabei sind in den Zeilen der Oberfläche folgende Funktionen hinterlegt (s. Tab. 1):

Tabelle 1: Funktionen der Hauptoberfläche

Zeile	Funktion
1	Auswahl der CT-Bilder
2	Auswahl der CDB-Datei
3	Auswahl der ANSYS-Version
4	Kalibrierungsaktivierung und Starten der Oberfläche

5 und 6	Einstellen der Parameter für die Korrelation der Aschedichte
7 und 8	Einstellen der Parameter für die Korrelation des E-Moduls (kompletter oder kortikaler Bereich)
9	Aktivierung und Einstellen der Unterscheidung der Berechnung des E-Moduls bezüglich kortikalem und spongiösem Bereich
10 und 11	Einstellen der Parameter für die Korrelation des E-Moduls (bei Aktivierung spongiöser Bereich)
12	Einstellen der Größe der Gruppierung des E-Moduls
13	Einstellen der Querkontraktionszahl
14	Erzeugen der 3D-Ansichten
15	Aktivierung der Oberfläche für den DICOM-Slicer und der CDB-Datei
16	Name der neuen CDVB-Datei und Eingabe des Speicherorts
17	Anzeige des Speicherorts und Start der Berechnung

Sobald die Berechnung gestartet wird, erscheint ein Popup-Fenster, bei dem der Start nochmals bestätigt werden muss. Wenn die Berechnung fertig ist, erscheint erneut ein Popup-Fenster in dem der Speicherort und die benötigte Rechenzeit angezeigt wird.

5. Validierung anhand eines Beispiels

Um die Funktion der Routine zu überprüfen, wurde ein Prüfkörper erstellt, der dem gesamten Material-Mapping unterzogen wurde. Ziel war es, die geometrische Genauigkeit des Prozesses zu untersuchen und die Ergebnisse der berechneten Knochenmineraldichte mit den vorher bekannten Werten und mit den Werten einer Dichtemessung, wie sie im Krankenhaus zur Osteoporose-Untersuchung durchgeführt wird, zu vergleichen.

Als Prüfkörper wurde ein Zylinder hergestellt, welcher als Basismaterial 10-prozentige Gelatine beinhaltet. In 2 Stufen wurde Calcium hinzugefügt, sodass im CT erkennbare Stufen entstehen, welche die Dichten voneinander abgrenzen (s. Abb. 43).

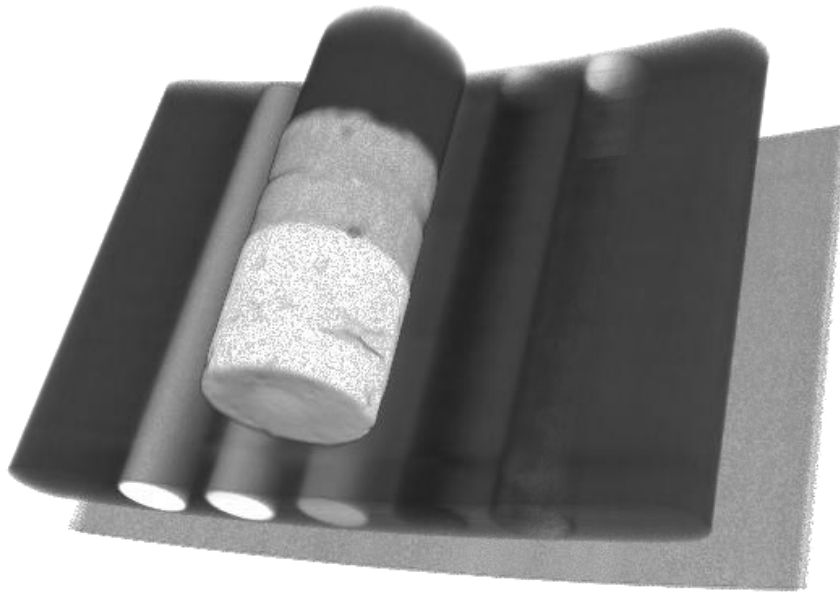


Abbildung 43: CT-Volumendarstellung des Gelatinezylinders mit unterschiedlichen Masseanteilen an Calcium und dem beigelegten Phantom

Während des Prozesses der Geometrierekonstruktion gibt es, aufgrund der individuell selektierten Voxel und der Glättungen der Oberfläche, Abweichungen des 3D-Körpers zum realen Körper. Es soll untersucht werden, wie groß eine solche Abweichung ist und ob diese einen Einfluss auf das Ergebnis des Material-Mappings hat.

Die Masseanteile des Calciums mussten so gewählt werden, dass sich die Masse noch verrühren und in die Zylinderform gießen ließ, sich jedoch eine möglichst hohe Dichte ergab, um einen großen Bereich abzudecken. Für die Gelatineschichten mit Calcium wurde ein Masseverhältnis von 2:1 und 1,5:1 Gelatine zu Calcium gewählt. Die theoretischen Dichten ergeben sich nach der Formel:

$$\rho_{\text{misch}} = \frac{m_1 + m_2}{\frac{m_1}{\rho_1} + \frac{m_2}{\rho_2}} \quad (5.1)$$

zu $1,05 \text{ g/cm}^3$ für die reine Gelatinelösung, $1,36 \text{ g/cm}^3$ für den Zylinderabschnitt mit dem Verhältnis 2:1 und $1,45 \text{ g/cm}^3$ für den Zylinderabschnitt mit dem Masseverhältnis 1,5:1, wobei das reine Calcium eine Dichte von $3,4 \text{ g/cm}^3$ aufweist.

Da die mit Calcium gemischten Gelatine nicht homogen gemischt werden konnte und es zudem zu geometrischen Fehlern kam (s. Abb. 44), wurde für den Vergleich der Geometrie ausschließlich der reine Gelatinezylinder genommen, da das Ausgangsvolumen bei diesem am besten bestimmt werden konnte.

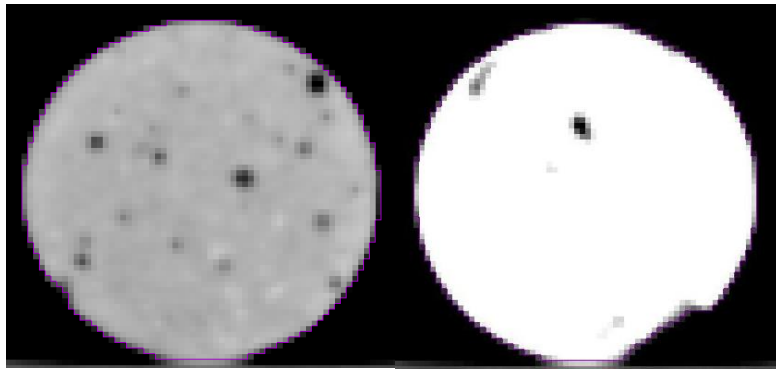


Abbildung 44: Schnittbilder der zwei Calciumabschnitte des Testzylinders mit Lufteinschlüssen und Randfehlern

Für den Vergleich der ermittelten E-Moduln für jedes Element und für die Gruppierung dieser wurde zudem das linke Hüftbein, welches als Beispiel in den vorherigen Kapiteln Anwendung fand, mit 3 unterschiedlichen Elementgrößen vernetzt und die Ergebnisse der Python-Routine mit den kalibrierten und unkalibrierten Ergebnissen von Bonemat verglichen. Dafür wurde der Volumenkörper mit 10mm, 5mm und 2mm, unter Verwendung quadratischer Tetraederelemente, diskretisiert.

Die Kalibrierung konnte mit der eigenen implementierten Funktion durchgeführt werden, die Dichtewerte der einzelnen Phantomzylinder waren diesbezüglich vom Klinikum vorgegeben. Diese werden in Dikaliumhydrogenphosphat äquivalenter Dichte angegeben, was allgemein der Knochenmineraldichte entspricht (K₂HPO₄-Dichte, BMD, Röntgendichte, ρ_{QCT} oder Knochenmineraldichte beschreiben die gleiche Dichteart), mit der Einheit g/cm^3 K₂HPO₄. Für die Kalibrierung werden Phantome aus in Wasser gelöstem K₂HPO₄ verwendet, da dies ähnliche Röntgenschwächungseigenschaften aufweist, die dem echten Knochenmineral (Calciumhydroxylapatit) ähnlich ist.

Das Phantom besteht aus 5 isolierten Zylindern, die unterschiedliche K₂HPO₄-Dichten aufweisen (s. Abb. 45 und Tab. 2).

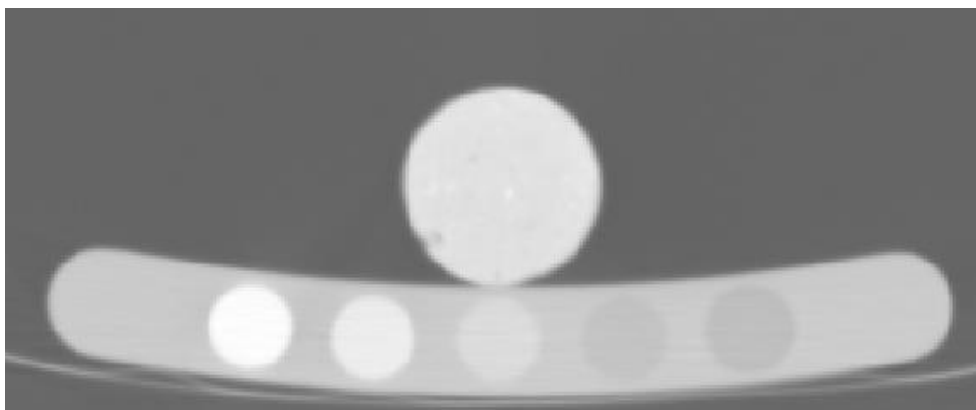


Abbildung 45: Schnittbild des CT-Scans des Gelatineblocks mit untergelegtem Phantom mit den isolierten Zylindern 1 bis 5 (von links nach rechts)

Tabelle 2: K2HP04-Dichten der einzelnen Phantomzylinder

Zylinder	BMD [g/cm ³]
1	0,37588
2	0,15705
3	0,05888
4	-0,05340
5	-0,05183

Die linearen Kalibrierungsparameter werden für die Umrechnung von den HU-Werten in die BMD verwendet. In Bonemat sind Ausgangsparameter hinterlegt, welche abgeändert werden können. Für den Vergleich werden die Berechnungen mit den Ausgangsparametern als unkalibriert deklariert und die Berechnungen mit den eigens ermittelten Parametern als kalibriert.

6. Ergebnisse und Diskussion

Goemetriegenauigkeit

Die Gießform der Zylinder hatte ein Durchmesser von 46mm und eine Höhe von 40mm. Der Gelatinezylinder hat durch Schrumpfung nach dem Entformen einen Durchmesser von ca. 45,3mm und eine Höhe von 39,5mm, wodurch das Volumen des reinen Gelatinezylinders 63,7cm³ beträgt. Das Volumen des rekonstruierten 3D-Modells beträgt hingegen 59,6g/cm³, was einem Volumenschwund von 6,4 Prozent entspricht (s. Abb. 46). Der Effekt wird jedoch geringer, je größer der Körper ist, da die Flächenabweichung nicht von der Größe des Körpers, sondern von den gewählten Glättungsoptionen abhängig ist.

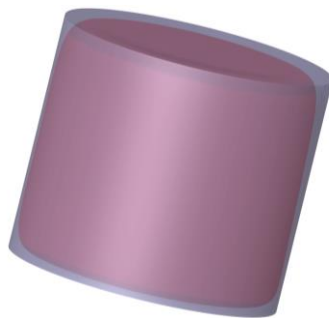


Abbildung 46: rekonstruierter Gelatinezylinder (rot) und Ausgangszylinder (transparent)

Der Einfluss der Geometrieabweichung hat bei dem gewählten Beispiel keinen großen Einfluss, da die HU-Grenze der Gelatine zur umliegenden Luft eher gering ist.

Bei der Untersuchung von Knochen kann es jedoch zu größeren Einflüssen in den Randbereichen und der kortikalen Schicht kommen. Dabei spielen das geringere Volumen und die Auflösung des Scans eine Rolle. Der verwendete Scan hat eine Auflösung von 512x512x48 mit einer Voxelgröße von 0,73x0,73x3mm. Bei einer zu groben Auflösung kann es zu ungenauen und unterschätzten Werten in der Kortikalis kommen, da diese, vor allem bei Osteoporose, wenige Millimeter dick sein kann. Wenn der Scan an kortikalen Stellen lediglich 3mm Voxellänge aufweist und die Kortikalis zusätzlich durch den Volumenschumpf dünner wird, werden die ermittelten HU-Werte und E-Moduln geringer als bei höherer Auflösung und dickerer Kortikalis, da die HU-Werte zum Ermitteln der Dichte und des E-Moduls geringer werden.

BMD

Die im Klinikum ermittelten Dichten basieren auf einer automatisch implementierten Kalibrierung in der Software, wobei die Korrelation nicht angegeben wird. Es muss lediglich die ROI (region of interest) ausgewählt werden (s. Abb. 47).

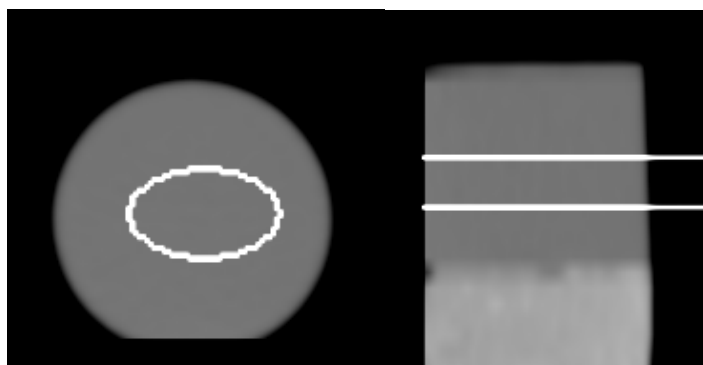


Abbildung 47: Auswahlansicht der ROI im reinen Gelatineblock in der klinischen Software zur Ermittlung der BMD

Aus 3 Schnittbildern und der ROI wurden die HU-Werte für die Korrelation ermittelt, welche folgend die MDB für das selektierte Volumen liefert.

Die eigene Kalibrierung des CT-Datensatzes des Gelatine-Calciumzylinders ergab mit den Daten aus Tabelle 2 die folgende lineare Korrelation (Gl. 6.1):

$$BMD \left[\frac{g}{cm^3} \right] = 0,0061982356 * HU - 0,04111 \quad (6.1)$$

In Tabelle 3 sind die resultierenden BMD-Werte der eigenen Korrelation und die der klinischen Software mit zugehörigen HU-Werten der ROI und der Abweichung aufgeführt:

Tabelle 3: BMD und HU-Werte der eigenen und der klinischen Kalibrierung

Zylinderabschnitt	eigens kalibrierte BMD [mg/cm ³]	Klinische BMD [mg/cm ³]	HU der ROI	Abweichung in %
Gelatinelösung	-12,6	23,4	46	153,8
Masseverhältnis 2:1	141,7	200,7	295	29,4
Masseverhältnis 1,5:1	423,6	522,7	750	18,9

Es ist kein Zusammenhang der eigens ermittelten Daten und der klinisch ermittelten Daten erkennbar. Im niedrigen Bereich ergab sich eine negative Dichte bei der eigenen Korrelation und eine Abweichung von ca. 153%. Bei höheren HU- und Dichtebereichen näherten sich die BMD Werte, weisen jedoch trotzdem eine Abweichung von bis zu ca. 19% auf.

Da die genaue Vorgehensweise der Kalibrierung der klinischen Software nicht bekannt ist, kann nicht nachempfunden werden, wie die BMD-Werte genau ermittelt wurden. Ein Grund für die Unterschiede kann eine gänzlich andere Berechnungsmethode der BMD sein. Dies würde jedoch der aktuellen Literatur widersprechen, in welcher ausschließliche Korrelationen über lineare Regression hergeleitet werden. Ein weiterer Grund könnte sein, dass nicht alle Phantomzylinder zum Kalibrieren genommen wurden. Bei Nutzung der Phantomzylinder mit niedrigstem und höchstem Dichtewert konnten BMD-Werte über eigene Kalibrierung errechnet werden, die eine Abweichung von rund 24%, 7% und 5% für die jeweiligen Gelatineabschnitte, gegenüber den klinischen Werten, aufweisen. Diese Werte kommen zwar näher an die klinisch ermittelten Daten heran, aber es erscheint nicht sinnvoll, nur 2 der 5 gegebenen Phantomzylinder zum Kalibrieren zu verwenden. Auch die Verwendung einer quadratischen Regression erzielte keine Verbesserung, sondern sorgte für eine Erhöhung der Abweichungen.

Eine andere Ursache könnten falsche Ausgangswerte der K₂HPO₄-Dichten der Phantomzylinder sein, die für die eigene Korrelation verwendet wurden. Die Richtigkeit dieser Daten lassen sich nur mit erneuter Absprache der verantwortlichen Personen im Klinikum überprüfen.

E-Modul

Für den Vergleich der E-Moduln der Python-Routine und der kalibrierten und nicht kalibrierten Software Bonemat sind in Tabelle 4 die Werte für alle Elemente (Gruppierung von 0 MPa) und in Tabelle 5 die Unterschiede der Ergebnisse gelistet. In Tabelle 6 sind die E-Modul-Werte der gruppierten Elemente (immer 200 MPa Gruppierung) und deren Differenzen zwischen den Programmen aufgeführt. BM-C steht dabei für die kalibrierten

Ergebnisse Bonemats und BM für die Unkalibrierten. Die resultierenden E-Moduln der Python-Routine und der kalibrierten Bonemat Software S. basieren auf der HU-BMD Korrelation aus Gleichung (6.1). Die Ergebnisse der unkalibrierten Bonemat Software basieren auf dem in Bonemat hinterlegten Zusammenhang:

$$BMD = 0.000791701 * HU - 0.00393573 \quad (6.2)$$

Die Berechnung der Aschedichte und des E-Moduls erfolgt für alle Berechnungsmethoden auf dieselbe Weise nach den Gleichungen (2.3) bis (2.5).

Tabelle 4: E-Moduln aller Elemente ohne Gruppierung

E-Modul	Elementgröße[mm]	Python	BM-C	BM
Maximum [MPa]	10	11585,3	11581,7	14166,5
	5	12465,9	12454,3	15194,1
	2	13889,3	13879,1	16871,4
Minimum [MPa]	10	333,7	333,7	337,8
	5	333,7	333,7	333,9
	2	333,7	333,7	333,9
Mittel [MPa]	10	2765,7	2764,7	3643
	5	2513,8	2512,9	3335,7
	2	2343,2	2342,3	3126,3

Im Allgemeinen lässt sich feststellen, dass die maximalen E-Moduln, bezüglich jeder Berechnungsmethode, kleiner werdenden Elementen größer werden. Der Trend ist nachvollziehbar, da pro Element weniger Einflüsse umliegender Voxel einbezogen werden bzw. das zu betrachtende HU-Feld innerhalb eines Elements kleiner wird und die HU-Differenzen untereinander geringer werden.

Die Minimalwerte liegen bei der Berechnungsmethode Python und BM-C beim Minimalwert, der durch die Begrenzung in der Routine definiert wird, wogegen bei BM dieser nicht erreicht wird. Die Differenzen der Werte aus Tabelle 4 sind in Tabelle 5 aufgeführt.

Tabelle 5: Differenzen aller Elemente

E-Modul	Elementgröße[mm]	Python-BM-C	Python-BM
Maximum [MPa]	10	3,6	2581,2
	5	11,6	2728,2
	2	10,2	2982,1

Mini- mum [MPa]	10	0	4,1
	5	0	0,2
	2	0	0,2
Mittel [MPa]	10	1	877,3
	5	0,9	821,9
	2	0,8	783,1

Die Unterschiede der Moduln in der Python-Routine und der kalibrierten Bonemat S. fallen so gering aus, dass sie mit Rundungsunterschieden in der Berechnung begründet werden können. Die Differenz von knapp 3000 MPa zu der unkalibrierten Bonemat S. ist auf die unterschiedliche Korrelation von HU zu BMD zurückzuführen (s. Abb. 48).

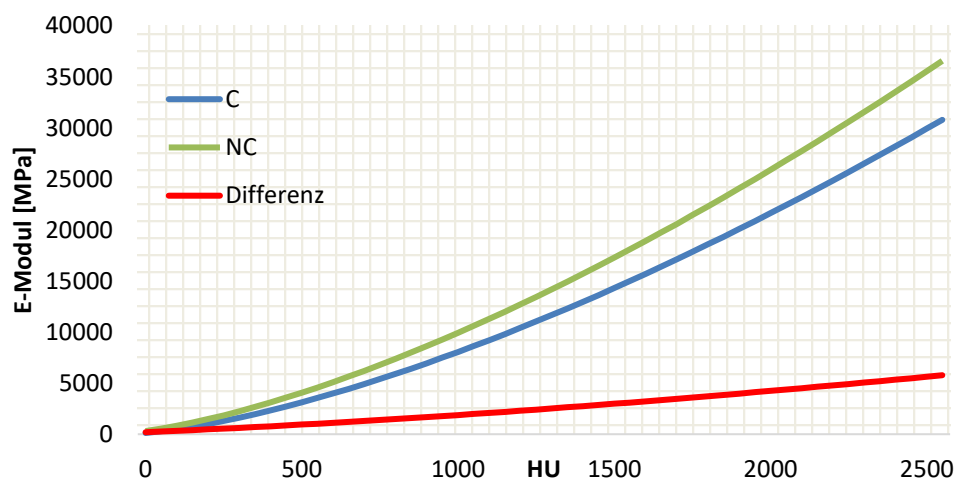


Abbildung 48: E-Moduln durch Verwendung von verschiedenen Kalibrierungskorrelationen (blau kalibriert (C), grün nicht kalibriert (NC)) und die Differenz der Werte (rot)

E-Modul Gruppierung

Bei der Gruppierung der Elementwerte wurde für alle Berechnungen ein Gap von 200 MPa gewählt. Auch bei den gruppierten Elementen werden die Maximalwerte des E-Moduls mit kleiner werdenden Elementzahl größer. Diesbezüglich ist die gleiche Begründung wie bei den ungruppierten Ergebnissen aufzuführen (s. Abb. 49, 50 und Tab. 6).

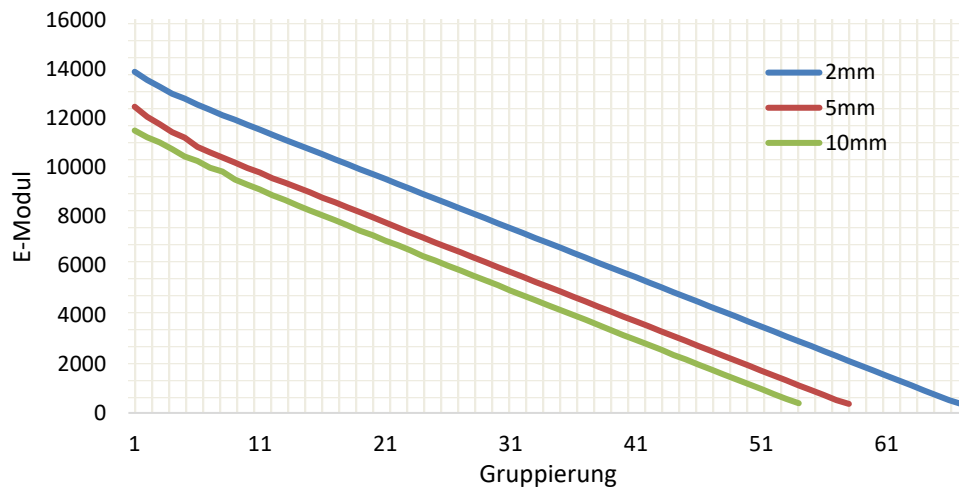


Abbildung 49: Einfluss der Elementgröße des FE-Netzes auf die Höhe des E-Moduls der Gruppierungen mit Ergebnissen aus der Python-Routine

Tabelle 6: E-Moduln der gruppierten Elemente

E-Modul	Elementgröße[mm]	Python	BM-C	BM
Maximum [MPa]	10	11497,72	11581,7	14166,5
	5	12465,9	12454,3	15194,1
	2	13889,3	13879,1	16871,4
Minimum [MPa]	10	393,7	479,0	337,8
	5	368,1	413,1	462,8
	2	359,1	423,3	445,7
Mittel [MPa]	10	2765,7	2863,9	3742,4
	5	2513,8	2611,3	3435,5
	2	2434,2	2440,9	3226,1
Anzahl Gruppierungen	10	54	54	66
	5	58	58	71
	2	67	67	81

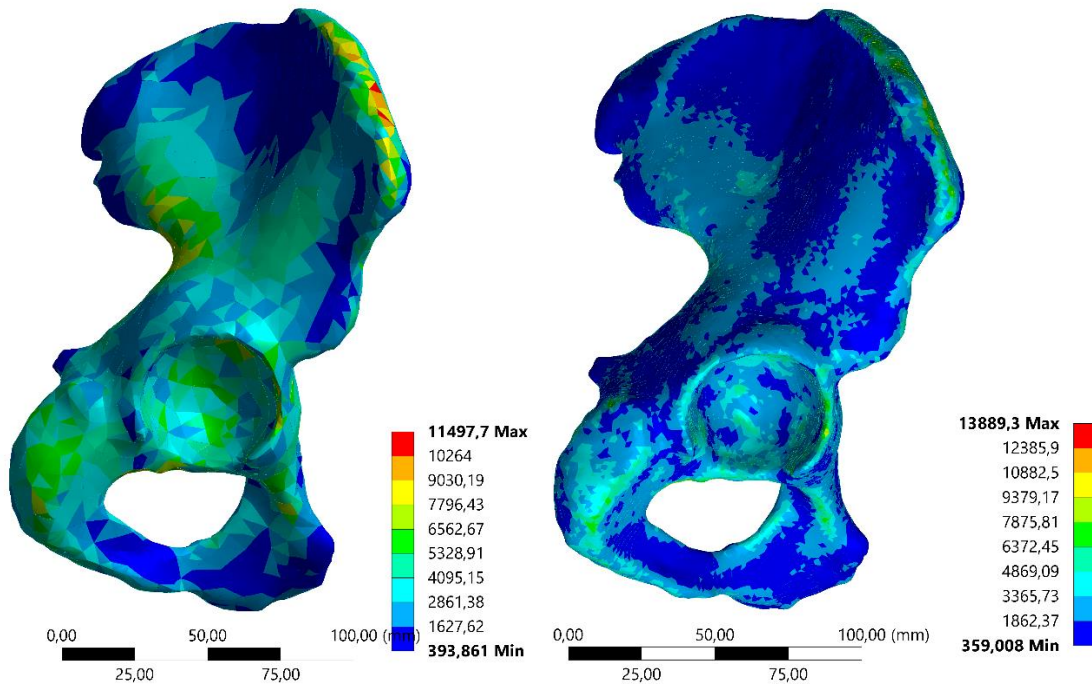


Abbildung 50: linkes Hüftbein mit 2mm Vernetzung (rechts) und 10mm Vernetzung (links) mit gemapptem E-Modul und 200 MPa Gruppierung

In der Python-Routine wurden für die in einem gruppierten Bereich liegenden Elemente der Mittelwert gebildet und jedem dieser Elemente zugeordnet. Diese Option ist auch in Bonemat aktiviert. Trotz dessen wurde jedoch stets der Maximalwert des ersten Elements der Gruppierung als Wert gewählt, sodass dieser auch immer mit den Maximalwerten aus Tabelle 4 übereinstimmt (s. Tabelle 6) und leicht höher ist als die äquivalente Gruppierung der Python-Routine (s. Abb. 51).

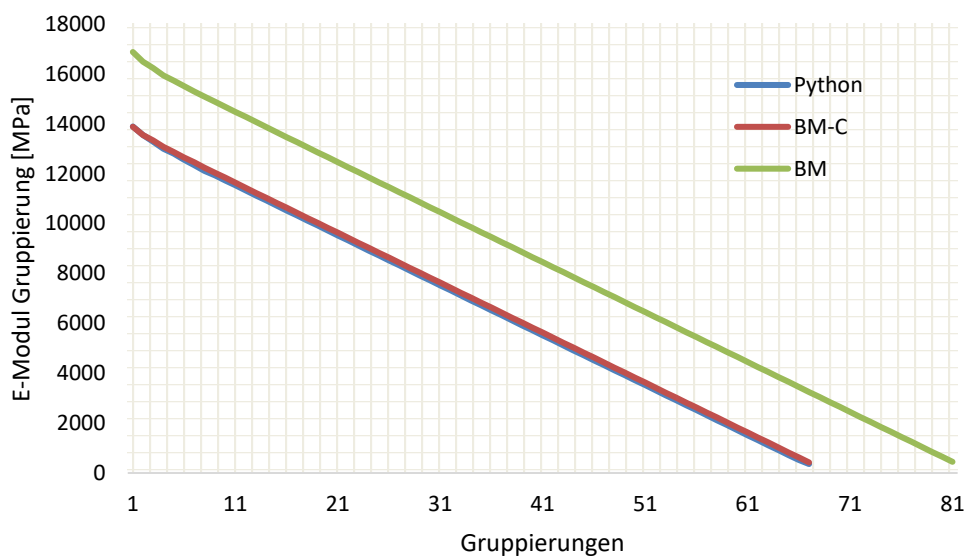


Abbildung 51: E-Modul pro Gruppierung für 2mm Elementgröße

Die gruppierten Werte der unkalibrierten Bonemat S. liegen deutlich höher als die Werte der anderen beiden Methoden, was der Verwendung der anderen Kalibrierungskorrelation geschuldet ist.

Die Anzahl der berechneten Gruppierungen ist bei den kalibrierten Methoden (Python und MB-C) exakt gleich, wogegen die unkalibrierte Bonemate S. durch die höheren E-Moduln mehr Gruppierungen generiert (s. Abb. 52).

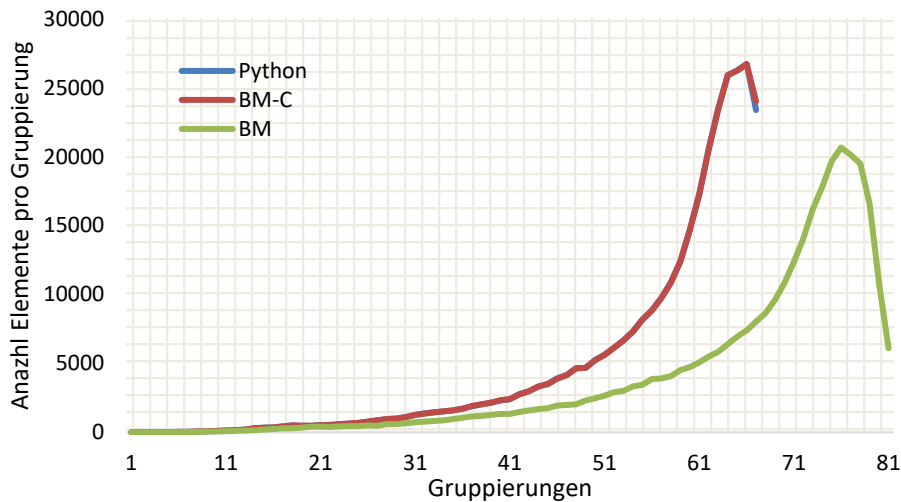


Abbildung 52: Gruppierungen mit Anzahl der Elemente pro Gruppierung für 2mm Elementgröße

Die Diagramme in Abbildung 51 und 52 stellen die Ergebnisse der unterschiedlichen Programme anhand des 2mm vernetzten linken Hüftbeins dar. Die anderen Elementgrößen (5mm und 10mm) ergeben äquivalente Diagramme.

Die Ergebnisse zeigen, dass bei der Verwendung gleicher Kalibrierkorrelationen, der Berechnungsalgorithmus, der Python-Routine und der von Bonemat, vergleichbare Ergebnisse liefert und ein Drift verhindert werden kann. Die Unterschiede liegen in der Gruppierung, wobei in Bonemat die eingestellte Variante nicht berücksichtigt wurde.

Rechenzeit

Um ein Vergleich der Alltagstauglichkeit der Python-Routine zu ziehen, wurden die Berechnungszeiten mit denen von Bonemat verglichen, wobei immer der gleiche Computer verwendet wurde (s. Tab. 7).

Tabelle 7: benötigte Rechenzeit, um die gemappte CDB-Datei zu erstellen

Elementanzahl	Anzahl Körper	Python-Routine [min]	Bonemat [min]
546105	6	156	5,4
320502	1	73,7	1,9

26186	1	16,4	1,1
8281	1	11,6	0,8
107 (1 Part)	1	0,07 (4,3s)	0,05 (3s)

Die Rechenzeiten der Python-Routine sind bei hohen Anzahlen von Elementen um ein Vielfaches höher als die von Bonemat. Das liegt einerseits daran, dass ein in C geschriebenes Programm immer schneller ist, als ein in Python geschriebenes. Die Hauptursache liegt jedoch in der Parallelisierung. Die Python-Routine ist dahingehend nicht optimiert, alle Datenflüsse werden nacheinander abgearbeitet, sobald die Berechnung gestartet wird. Darüber hinaus sind nicht alle Schleifen optimal implementiert und es könnten schnellere Varianten gefunden werden. So werden bei jeder Berechnung des E-Moduls für ein Element alle Parameter (aus der Benutzeroberfläche) abgefragt, was nicht nötig ist, aber die Rechenzeit stark erhöht.

Bonemat ist über Jahrzehnte im Entwicklungsprozess und wurde über die bezüglich der Effizienz verbessert und parallelisiert. Die Bilddaten des CT-Scans werden beispielsweise direkt beim Angeben des Pfades eingelesen und verarbeitet, wohingegen in der Python-Routine zuerst der Speicherpfad des Ordners abgelegt wird und erst bei Berechnungsstart die Bilddaten verarbeitet werden. Das Programm hat diesbezüglich noch Potential, um die Rechengeschwindigkeiten zu reduzieren. Es ist jedoch zu beachten, dass die Zeit, die benötigt wird um die Dateien so vorzubereiten, dass sie im Programm verwendet werden können, die Berechnungszeit übersteigt. Dies bedeutet, dass das händische Selektieren der Geometrie und das Umwandeln in die CDB-Datei die meiste Zeit in Anspruch nimmt.

7. Zusammenfassung und Ausblick

Das Ziel, den vorhandenen Code zu optimieren, sodass ein funktionstüchtiges Programm entsteht, wurde erreicht. Das Python-Programm berechnet in gleicher Weise wie Bonemat die Materialdaten und liefert, bis auf die Gruppierung und Rundungsunterschiede, gleiche Ergebnisse. Es konnten zudem Funktionen hinzugefügt werden, die in Bonemat nicht vorhanden sind. So können verschiedene Elementtypen in verschiedenen Parts verwendet werden, die Kalibrierung des CT-Scans kann für die Korrelation der HU-Werte und der BMD direkt im Programm durchgeführt werden und es ist möglich, zwei verschiedenen Berechnungsvorschriften für den E-Modul bei Verwendung eines Dichtegrenzwertes zu verwenden. Außerdem können CT-Daten eingelesen werden, bei denen Bonemat aus unerklärlichen Gründen abstürzt.

Es ist jedoch anzumerken, dass die ermittelten Kalibrierungsparameter des gegebenen Scans nicht mit denen einer klinisch verwendeten Software übereinstimmen, die Kalibrierung selbst jedoch essenziell für weitere Berechnungen ist.

Die Ergebnisse der Dichte und des E-Moduls, auf Basis der CT-Daten, können auf 5 Arten beeinflusst werden: Durch die Korrelationen, die die Werte umrechnen; durch den Zuweisungsalgorithmus der Materialwerte auf die Elemente des FE-Netzes; durch die Wahl der verwendeten Elementgröße, der Auflösung des CT-Scans und durch den selektier-Vorgang der CT-Voxel, um die 3D-Geometrie aus den CT-Daten abzuleiten.

Mapping-Algorithmus

Durch Vergleiche der unterschiedlichen Algorithmen konnte der im Python-Programm verwendete Algorithmus, als der Genauste identifiziert werden.

Elementgröße, CT-Auflösung und Geometrierekonstruktion

Die Elementgröße und die Auflösung des CT-Scans sollten möglichst gering sein, um akkurate Ergebnisse zu erzielen.

Die Geometrieveränderung durch den Vorgang der Rekonstruktion hat einen Einfluss auf die ermittelten Ergebnisse. Durch den Volumenschrumpfung wird der Effekt der unterschätzten E-Moduln der Kortikalis durch geringe CT-Auflösung verstärkt. Dieser wurde in verschiedenen Studien aufgegriffen und verschiedene Lösungsansätze ausgearbeitet [29] [30]. Eine Verbesserung der Ergebnisse wurde durch die separate Betrachtung der Kortikalis als eigener Körper erzielt, was sich auch ohne Weiteres auf die Python-Routine anwenden lassen würde.

Die händische Rekonstruktion der Geometrie benötigt anatomisches Wissen und viel Zeit, welche bei Verwendung eines automatischen Selektionsalgorithmus verringert werden könnte.

Materialkorrelationen

Die Materialkorrelationen können frei gewählt werden und haben somit den größten Einfluss auf die berechneten Materialkennwerte. Es ist also von Bedeutung, die passende Korrelation für das zu untersuchende Objekt zu wählen oder gar eigens herzuleiten. Dabei ist zu erwähnen, dass für das in der Arbeit genutzte Beispiel des Hüftbeins keine E-Modul-Korrelationen existieren. Es wurde die Korrelation implementiert, die in der Literatur am meisten verwendet wurde und welche für das Femur gültig ist. Durch die Wahl der Korrelation wäre es allerdings auch denkbar, andere Materialien als Knochen zu untersuchen.

Der E-Modul wird, historisch bedingt, aus der Dichte berechnet. Im Laufe der Zeit und der Forschung wurden verschiedene Korrelationen zwischen E-Modul und den verschiedenen Dichtearten sowie HU-Werten hergeleitet. Es wäre jedoch eine mögliche und als sinnvoll eingeschätzte Lösung, einen direkten Zusammenhang von E-Modul und HU-Werten oder E-Modul und BMD zu verwenden. In einigen Studien wurde dies schon praktiziert, jedoch basieren die Herleitungen der Korrelationen auf neuronalen Netzwerke-gestützten Berechnungen und sind so nicht vollständig nachvollziehbar, ergaben jedoch eine Verbesserung der berechneten Materialwerte [19].

Anisotropie

Die Verwendung von anisotropen Materialeigenschaften wird in den betrachteten Studien, aufgrund geringer Unterschiede in den Ergebnissen unterschiedlicher FE-Analysen, als nicht notwendig empfunden.

Durch die vielen verschiedenen Schritte, die für den Mapping-Prozess notwendig sind, ist es wichtig, an jeder Stelle möglichst geringe Abweichungen zu erzielen. Demnach kann die Verwendung von anisotropen Materialkennwerten trotzdem zu genaueren Ergebnissen führen und könnte als eine weitere Funktion in die Python-Routine implementiert werden.

Programmfluss

Für das gesamte Material-Mapping sind mindestens 3 Programme notwendig: Zum einen das Selektionsprogramm zur Geometrierekonstruktion, zum anderen ein CAD-Programm, um dieses zu verarbeiten, zu verbessern und um es in das passende Dateiformat umzuwandeln. Und zuletzt ein FE-Programm, um den Körper zu vernetzen und um spätere Festigkeitsberechnungen durchzuführen. Diese Funktionen innerhalb eines Programms verwenden zu können, würde Zeit sparen und die allgemeine Verfügbarkeit verbessern.

8. Literaturverzeichnis

- [1] S. Bundesamt, „Bundeszentrale für politische Bildung,“ 19 09 2019. [Online]. Available: <https://www.bpb.de/kurz-knapp/zahlen-und-fakten/soziale-situation-in-deutschland/61541/bevoelkerungsentwicklung-und-altersstruktur/>. [Zugriff am 01 03 2022].
- [2] „Osteoporose Selbsthilfegruppen Dachverband e.V,“ 06 01 2023. [Online]. Available: <https://www.osd-ev.org/osteoporose/>. [Zugriff am 07 02 2023].
- [3] A. Hanan, „Data management of high-resolution microCt data sets of osteoporotic bone and Finite Element (FE) meshes,“ Hochschule Reihn-Waal, 2022.
- [4] THM, „THM Mittelhessen Campus Giessen,“ [Online]. Available: <https://www.thm.de/mni/forschung/institute-gruppen/imps/imps-strahlung-information/radiologie/modalitaeten/computertomographie.html#:~:text=Funktionweise%3A,ein%20Signal%20am%20Detektor%20aufgenommen..> [Zugriff am 30 Dezember 2022].
- [5] W. Schlegel, Medizinische Physik, Berlin: Springer-Verlag GmbH Deutschland, 2018.
- [6] S.-W. HA, Medizintechnik, München: Springer, 2009.
- [7] R. Medizinphysik, „Medizinphysik.wiki,“ [Online]. Available: <https://medizinphysik.wiki/computertomographie/>. [Zugriff am 15 03 2022].
- [8] DICOM@OFFIS, „dicom.offis.de,“ 05 04 2022. [Online]. Available: <https://dicom.offis.de/dcmintro.php.de>. [Zugriff am 30 09 2022].
- [9] wikiwand, „www.wikiwand.com,“ 07 03 019. [Online]. Available: https://www.wikiwand.com/de/Digital_Imaging_and_Communications_in_Medicine. [Zugriff am 05 12 2022].

- [10] Innolitics, „innolitics.com,“ [Online]. Available: <https://dicom.innolitics.com/ciods/ct-image/patient/00081120>. [Zugriff am 05 12 2022].
- [11] B. Helgason, „Mathematical relationships between bone density and mechanical properties: A literature review,“ ScienceDirect, 2007.
- [12] J. Galante, „Physical Properties of Trabecular Bone,“ Springer-Verlag, 1970.
- [13] N. Ashjaee, „QCT-FE modeling of the proximal tibia: Effect of mapping strategy on convergence time and model accuracy,“ Medical Engineering and Physics, Saskatoon, 2021.
- [14] E. F. Morgan, „Trabecular bone modulus density relationships depend on anatomic site,“ Elsevier, Berkeley, 2003.
- [15] E. Schileo, „An accurate estimation of bone density improves the accuracy of subject-specific finite element models,“ Elsevier, Bologna, 2008.
- [16] T. Keller, „Predicting the compressive mechanical behavior of bone,“ Journal of Biomechanics, 1994.
- [17] J.Y. Rho, „Relations of mechanical properties to density and CT numbers in human bone,“ Butterworth Heinemann, Dallas, 1995.
- [18] D. C. Wirtz, „Critical evaluation of known bone material properties to realize anisotropic FE-simulation of the proximal femur,“ Journal of Biomechanics, Aachen, 2000.
- [19] S. M. Nazemi, „Optimizing finite element predictions of local subchondral bone,“ Clinical Biomechanics, Vancouver, 2017.
- [20] E. Schileo, „Subject-specific finite element models can accurately predict strain levels in long bones,“ Journal of Biomechanics, Bologna, 2007.
- [21] R. Scholz, „Validation of density-elasticity relationships for finite element modeling of human pelvic bone by modal analysis,“ Journal of Biomechanics, Leipzig, 2013.

- [22] V. Baca, „Comparison of isotropic and orthotropic material property assignments on femoral finite element models under two loading conditions,“ Medical Engineering and Physics, Beijing, 2005.
- [23] V. Baca, „Comparison of an inhomogeneous orthotropic and isotropic material model used for FE analyses,“ Medical Engineering and Physics, Prague, 2007.
- [24] C. Les, „Estimation of Material Properties in the Equine Metacarpus with use of Quantitative Computed Tomography,“ Journal of Orthopaedic Research, San Francisco, 1994.
- [25] T. Fulvia, „An improved method for the automatic mapping of computed,“ Medical Engineering and Physics, Bologna, 2004.
- [26] W. Marcus, Lineare und nichtlineare FEM, Wiesbaden: Springer Vieweg, 2017.
- [27] F. Taddei, „The material mapping strategy influences the accuracy of,“ Medical Engineering and Physics, Bologna, 2007.
- [28] E. C. Pegg, „github,“ 09 2016. [Online]. Available: https://github.com/elisepegg/py_bonemat_abaqus. [Zugriff am 01 11 2022].
- [29] M. Asmaa, „The Impact of Voxel Size-Based Inaccuracies on the Mechanical Behavior,“ Annals of Biomedical Engineering, Toronto, 2010.
- [30] E. Shileo, „Cortical bone mapping improves finite element strain prediction,“ Elsevier, Bologna, 2020.
- [31] I. f. Q. u. W. i. G. (IQWiG), „gesundheitsinformation.de,“ 04 10 2018. [Online]. Available: <https://www.gesundheitsinformation.de/osteoporose.html>. [Zugriff am 2022 03 01].
- [32] wikibooks, „wikibooks.org,“ 30 11 2018. [Online]. Available: https://de.wikibooks.org/wiki/Physikalische_Grundlagen_der_Nuklearmedizin/_D%C3%A4mpfung_von_Gammastrahlen. [Zugriff am 2022 12 01].
- [33] hebala, „www.hebala.de,“ 2022. [Online]. Available: <https://www.hebala.de/shop/bewegliches-becken.html>. [Zugriff am 06 12 2022].
- [34] C. Gebhardt, Praxisbuch FEM mit ANSYS Workbench, München: Carl Hanser Verlag, 2018.

- [35] A. C. S. d. Alcantara, „Implementierung verschiedener Algorithmen zur automatischen Berechnung und Zuweisung von Materialgesetzen von CT-Daten auf FE-Netzte,“ Hochschule Merseburg, Merseburg, 2017.
- [36] T. U. München, „tum,“ [Online]. Available: https://www5.in.tum.de/lehre/seminare/semsoft/unterlagen_02/FE/website/. [Zugriff am 10 01 2023].
- [37] *Osteoporose: Symptome früh erkennen und behandeln*. [Film]. Deutschland: NDR, 2023.
- [38] S. R., „www.iue.tuwien,“ [Online]. Available: <https://www.iue.tuwien.ac.at/phd/sabelka/node28.html>. [Zugriff am 05 02 2023].
- [39] S. M. Nazemi, „Optimizing finite element predictions of local subchondral bone,“ Clinical Biomechanics, Saskatoon, 2016.

9. Anhang

Abbildungen Anhang

A.1 APDL-Code CDB-Datei

A.2 Zuordnung der Elementtypen

A.3 Partindizes

A.4 Voxelabstände in Z-Achse

A.5 Parameter der GUI

A.6 Korrelation der Kalibrierung

A.7 spongiöser und kortikaler Knochen

A.8 3D-Ansichten der CDB- Datei und CT-Datei (a) und des Schnittbildes (b)

A.9 DICOM-Slicer

Abbildungen Anhang

Abbildung A-1: falsche Initialisierung der Elementtypen der gemappten CDB-Datei zwischen Knoten- und Elementblock

Abbildung A-2: richtige Initialisierung der Elementtypen der originalen CDB-Datei im Startblock

Abbildung A-3: zusätzliche For-Schleife in der get_etype Funktion und zusätzlichen Rückgabewert t der _get_lh_elem Funktion

Abbildung A-4: falsche Partindizes der gemappten CDB-Datei

Abbildung A-5: richtige Partindizes der originalen CDB-Datei

Abbildung A- 6: Zuordnung der Partindizes mit neuer Codezeile (oben) und alter Codezeile (unten)

Abbildung A-7: Voxelgröße in Z-Richtung durch Differenz der Z-Koordinaten zweier Slices

Abbildung A-8: die in der Benutzeroberfläche und Standard-Parameter

Abbildung A-9: Korrelation der HU-Werte und der Röntgengichte durch Kalibrierung

Abbildung A-10: Berechnung des E-Moduls mit 2 Korrelationen und einen Grenzwert

A.1 APDL-Code CDB-Datei

```
!Length unitmm
Prep7
cmsel,s,Becken
cdwrite,db,Beckencdb, cdb
ALLSEL
SOLUTION
EOFS
```

A.2 Zuordnung der Elementtypen

Falsche und richtige Initialisierung der Elementtypen in der originalen und geänderten CDB-Datei (s. Abb. A-1 und Abb. A-2) und im geänderten Code (s. Abb. A-3).

```
N,R5.3,LOC,      -1,
ET,      1,185
EBLOCK,19,SOLID, 30,      30
(19i10)
```

Abbildung A-1: falsche Initialisierung der Elementtypen der gemappten CDB-Datei zwischen Knoten- und Elementblock

```
ET,      1,185
ET,      2,187
NBLOCK,6,SOLID, 91,      91
(3i9,6e21.13e3)
```

Abbildung A-2: richtige Initialisierung der Elementtypen der originalen CDB-Datei im Startblock

```
def _get_lh_elem(elements):
    eles = [i for i in elements if len(i) == 190]
    x = []
    t = []
    for i in eles:
        x.append(i[100:])
        t.append(i[10:21])
    ele = [" ".join(v.split()).split(" ") for v in x]
    print(t)
    return ele,t
```

```

def get_etype(parts):
    typ = []
    for num in range(len(parts)):
        types = parts[num].numtype #types = dic. {1 : 187}
        typ.append(types)
        print(typ)
        etype = ""
    for i in typ:
        for k, v in i.items():
            etype += 'ET,' + str(k) + ',' + str(v) + '\n'
        print("etype :", etype)
    return etype

```

Abbildung A-3: zusätzliche For-Schleife in der get_etype Funktion und zusätzlichen Rückgabewert t der _get_lh_elem Funktion

A.3 Partindizes

Die Zeilen des Elementblocks wurden mit den falschen Partindizes gefüllt, da diese an der falschen Stelle im *data_import* Modul ermittelt wurden. Im alten Code wurden die Zeilen des Elementblocks mit der Zahl 1 aufgefüllt, was jedoch nur bei der Verwendung eines einzelnen Parts richtig ist. Der Partindex jedes einzelnen Elements wird im Code in der Liste *elindex* im *data_output* Modul gespeichert. Für die Füllung der Zeilen im Elementblock wird auf diese zurückgegriffen (s. A-4 und A-5).

```

EBLOCK,19,SOLID,      30,      30
(19i18)
   3      2      1      1
  35     31
   3      2      1      1

```

Abbildung A-4: falsche Partindizes der gemappten CDB-Datei

```

EBLOCK,19,SOLID,      30,      30
(19i10)
   2      2      2      2
  62     68
   2      2      2      2

```

Abbildung A-5: richtige Partindizes der originalen CDB-Datei

Durch Ändern des Codes werden die richtigen Partindizes verwendet (s. Abb. A-6).

```

if len(finals[i]) == 210:
    x = indexes[i] + elindex[i] + elindex[i] + elindex[i]

    x = indexes[i] + elindex[i] + '      1      1'

```

Abbildung A-6: Zuordnung der Partindizes mit neuer Codezeile (oben) und alter Codezeile (unten)

A.4 Voxellabstände in Z-Achse

Die Abstände der einzelnen Slices bzw. die Größe der Voxel in Z-Richtung wurde durch den Code in Abb. A-7 berechnet und basieren nicht mehr auf den Informationen des DICOM Tags *SliceThickness*

```
s_s = []
for p in range(2):
    d = pydicom.dcmread(path + "/" + dicom_order[p])
    s_s.append(d)
ss = abs(round(float(s_s[0].ImagePositionPatient[2]) - float(s_s[1].ImagePositionPatient[2]), 4))
```

Abbildung A-7: Voxelgröße in Z-Richtung durch Differenz der Z-Koordinaten zweier Slices

A.5 Parameter der GUI

Die in der Benutzeroberfläche eingegebenen Parameter werden im Dictionary *gui_param* gespeichert. (s. Abb. A-8)

```
param['rhoQCTa'] = "-0.003935729"
param['rhoQCTb'] = "0.000791701"
param["ansys"] = gui_param["ansys_version"]
param["poisson"] = str(gui_param["poisson"])
param["Ethresh1"] = str(gui_param["treshhold_value"])
param['rhoAshall'] = str(gui_param['b'])
param['rhoAshb11'] = str(gui_param['a'])
param['Eb11'] = str(gui_param['c_cort'])
param['Ec11'] = str(gui_param['d_cort'])
param['numEparam'] = str(gui_param["treshhold_check"])
param['Eb1'] = str(gui_param['c_trab'])
param['Ec1'] = str(gui_param['d_trab'])
param['gapValue'] = str(gui_param["gap"])
param['minVal'] = '1e-06'
param['groupingDensity'] = 'Mean'
param['integration'] = 'E'
```

Abbildung A-8: die in der Benutzeroberfläche und Standard-Parameter

A.6 Korrelation der Kalibrierung

Die bei der Kalibrierung entstehende Korrelation zwischen den HU-Werten und der Röntgendichte wird in einem eigenen Fenster angezeigt (s. Abb. A-9). Die lineare Gleichung wird in der Oberfläche der Kalibrierfunktion angezeigt.

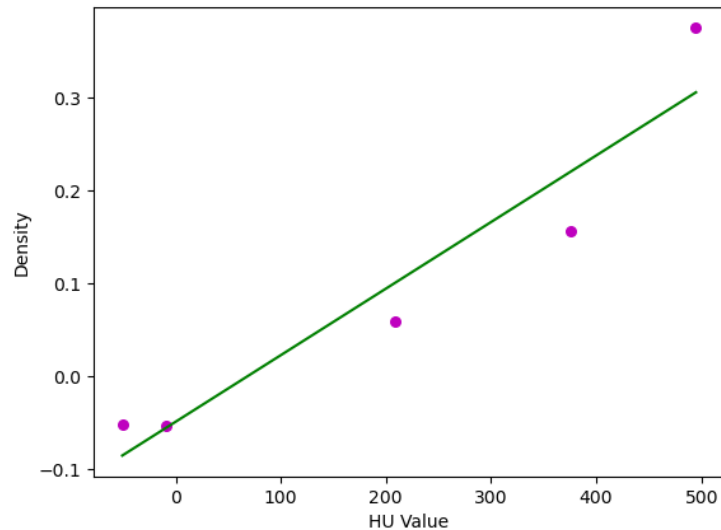


Abbildung A-9: Korrelation der HU-Werte und der Röntgendichte durch Kalibrierung

A.7 spongioser und kortikaler Knochen

Die Unterscheidung, ob mit unterschiedlichen Korrelationen gerechnet werden soll, muss in der Benutzeroberfläche aktiviert werden und es erfolgt die Berechnung des E-Moduls, abhängig vom eingestellten Grenzwert (s. Abb. A-10).

```

if param['numEparam'] == 'single':
    def modulus(Ash):
        res = param['Eb11'] * (Ash ** param['Ec11'])
        if res < 0:
            return 0.000001
        else:
            return res

elif param['numEparam'] == 'double':
    def modulus(Ash):
        if Ash < param['Ethresh1']:
            res = param['Eb1'] * (Ash ** param['Ec1'])
            if res < 0:
                res = 0.000001

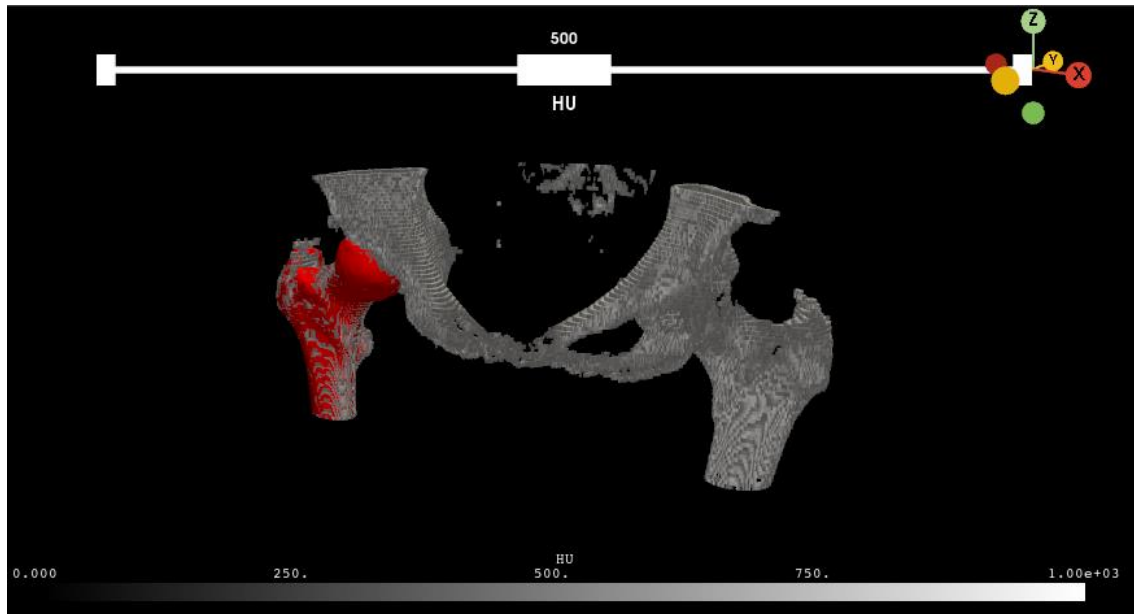
        elif Ash >= param['Ethresh1']:
            res = param['Eb11'] * (Ash ** param['Ec11'])
            if res < 0:
                res = 0.000001
        return res

```

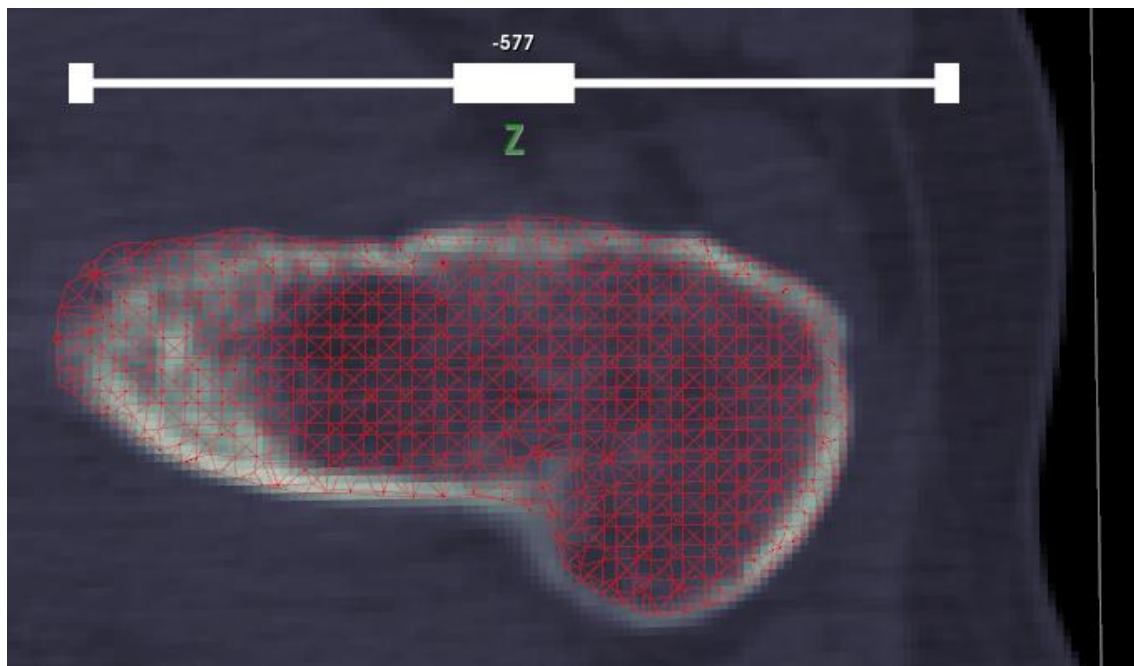
Abbildung A-10: Berechnung des E-Moduls mit 2 Korrelationen und einen Grenzwert

A.8 3D-Ansichten der CDB- Datei und CT-Datei (a) und des Schnittbildes (b)

a)



b)



A.9 DICOM-Slicer

