

MONTANUNIVERSITÄT LEOBEN

Auslegung einer Festigkeitsprüfung von Dentalkeramik mittels des B3B-Tests



DIPLOMARBEIT
VON
DANIEL GERNOT MEVEC
AM
INSTITUT FÜR STRUKTUR- UND FUNKTIONSKERAMIK

Leoben, am 14.06.2016

Erklärung

Ich erkläre an Eides statt, dass ich diese Arbeit selbstständig verfasst, andere als die angegebenen Quellen nicht benützt und mich auch sonst keiner unerlaubten Hilfsmittel bedient habe.

Affidavit

I declare in lieu of oath, that I wrote this thesis and performed the associated research myself, using only literature cited in this volume.

Danksagung

Ich bedanke mich herzlich bei den Mitarbeitern des Instituts für Struktur- und Funktionskeramik, die mich während meiner dortigen Anstellung und der Zeit meiner Diplomarbeit immer unterstützt haben. Mein besonderer Dank gilt Herrn o. Univ. Prof. Dr. Robert Danzer für das Ermöglichen dieser Arbeit und sein Vertrauen.

Weiters bedanke ich mich bei meinen Betreuern ao. Univ. Prof. Mag. Dr. Peter Supancic und Ass. Prof. Dipl.-Ing. Dr. Tanja Lube, die mir mit Rat und Tat zur Seite standen und ohne deren Erfahrung diese Arbeit nie zustande gekommen wäre.

Ein Dank geht auch meine Eltern, die mir das Studium an der Montanuniversität ermöglichten und an meine Freunde und Kommilitonen, die mir dieses Studium und besonders meine letzten Semester zu einer unvergesslichen Zeit machten.

Kurzfassung

Diese Arbeit beschäftigt sich mit der Anwendung des 4-Kugelversuchs auf runde Scheiben und rechteckige Platten aus Dentalkeramik.

Ziel war es, eine Fit-Formel für die biaxiale Spannungsverteilung in Proben eines bestimmten Geometriebereiches aufzustellen. Weiters wurde ein Programm zur Berechnung von effektiven Volumina und Oberflächen entwickelt. Durch Finite-Elemente-Rechnungen wurden Spannungsverteilungen erstellt und eine empirische Formel entwickelt, welche die Maximalspannung unter der Lastkugel beschreibt. Diese ist abhängig von der Probengeometrie, Materialeigenschaften und der Kugelgröße im Messaufbau. Für die Spannungsberechnung wurde eine Vereinfachung von Plattengeometrien auf einen effektiven Scheibenradius vorgeschlagen.

Die effektiven Größen für konkrete Proben können durch Interpolation der numerischen Ergebnisse ermittelt werden.

Zusätzliche Berechnungen wurden durchgeführt, um die Grenzen der Anwendbarkeit der Formel für Werkstoffe mit extrem niedrigen E-Modul aufzuzeigen und eine Fehlereinschätzung zu ermöglichen.

Abstract

The topic of this thesis is the Ball-on-3-Balls test applied to discs and rectangular plates made of dental ceramics.

The goal was to prepare a fitted formula for the biaxial stress distribution in samples of a given range of sizes. A program was also developed to calculate the effective volume and surface of these samples. Stress values were calculated through Finite-Element-Analysis and an empirical formula for the maximum stress under the loading ball was fitted to them. This formula is dependent on the sample's material properties and geometry in relation to the size of the balls used. A simplification of the stress formula for plate geometries to an effective disc radius was proposed.

The effective volume and surface for a given sample can be interpolated from the numerical data.

Further studies were done concerning the limits of the B3B test towards materials with extremely low Young's Modulus.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Der 4-Kugelversuch für die Dentalkeramik	1
1.2. Problemstellung	2
2. Grundbegriffe zum Versagen von Keramiken	4
2.1. Versagensmodell Keramik	4
2.2. Versagenswahrscheinlichkeit Keramik	6
2.3. V_{eff} und S_{eff} und deren Einfluss	8
2.4. Wahl der Skalierungsspannung	10
3. Finite-Elemente-Methode und numerische Integration	12
3.1. Methode der Finiten Elemente	12
3.2. Numerische Berechnung effektiver Volumen und Oberflächen	13
3.2.1. Einfacher Fall — das 2D Element	13
3.2.2. Erweiterung auf 3 Dimensionen	16
3.3. Fortran Programm	19
3.3.1. Aufbau des Programms	19
3.3.2. ANSYS UPF	21
4. FEM-Berechnungen an Platten und Scheiben	23
4.1. Auflagergeometrie	23
4.2. Vereinfachen des Modells	23
4.3. Randbedingungen der Simulation	25
4.4. Simulationsablauf	27
4.5. Ergebnisse und Interpretation	30
5. Definition einer empirischen Formel	32
5.1. Formel für die Maximalspannung	32
5.1.1. Scheiben	32
5.1.2. Platten	34
5.1.3. Schaurechnung	35
5.1.4. Vernachlässigung des Überhangs	39
5.2. Interpolation der effektiven Größen	39
5.3. Fehlereinflüsse — Gültigkeitsbereiche für elastische Platten	41
5.3.1. Auswirkung der Durchbiegung	41
5.3.2. Auswirkung des Kontakts	43
5.3.3. Vereinen der beiden Einflüsse	45
5.4. Schlussfolgerungen	45
6. Zusammenfassung und Ausblick	47
Literatur	51

Anhang	52
A. ANSYS Syntax	53
A.1. multirun.inp	53
A.2. multirun_data-export.nb	55
A.3. main.inp	56
A.4. geom_Platte_Punkt.inp	60
A.5. geom_Scheibe_Punkt.inp	62
A.6. post_veff_6.inp	63
A.7. post_veff_full.inp	63
A.8. write_result.inp	64
A.9. x_readtime_i.mac	65
A.10.x_Effektives_Volumen.mac	66
B. Fortran Syntax	72
B.1. Effektives_Volumen.f90	72
B.2. overhead.f90	79

Abkürzungsverzeichnis

B3B	Ball-on-three-balls
FEM	Finite Elemente Methode
LEBM	Linear-elastische Bruchmechanik
APDL	ANSYS Parametric Design Language

Symbolverzeichnis

σ_m	Maximalspannung unter der Lastkugel
t	Probendicke
R	Probenradius (bei Scheiben)
R_B	Kugelradius
R_a	Auflagerradius $\frac{2}{\sqrt{3}} R_B$
\tilde{R}	Ersatzradius (bei Platten)
τ	Normierte Probendicke $\frac{t}{R_a}$
ρ	Normierter Scheibenradius $\frac{R}{R_a}$
$\tilde{\rho}$	Normierter Ersatzradius bei Platten $\frac{\tilde{R}}{R_a}$
E	E-Modul
ν	Querkontraktionszahl / Poissonzahl

1. Einleitung

1.1. Der 4–Kugelversuch für die Dentalkeramik

Wie in vielen medizinischen Gebieten gibt es in der Zahnmedizin für Prothesen und Implantatwerkstoffe höchste Anforderungen an die Festigkeit und Lebensdauer. Ein weiterer wesentlicher Aspekt in der Zahnprothetik ist die Ästhetik und das optische Erscheinungsbild der Werkstoffe. In dieser Hinsicht sind Keramiken durch ihrer Ähnlichkeiten mit natürlichem Zahnschmelz erste Wahl.

Um geforderte Festigkeit zu gewährleisten, gilt es Defekte in diesem spröden Werkstoffen auf ein Minimum zu reduzieren. Daher wurden die Schritte in der Herstellung von Zahnersatz aus keramischen Werkstoffen auf ein Minimum reduziert. Oft werden heute Blöcke oder Zylinder aus Dentalkeramik unter Bedingungen höchster Reinheit industriell in einer Produktionsstätte hergestellt. Somit wird das Risiko von Defekten im Material reduziert und eine hohe Qualität der Keramik kann garantiert werden. Diese sogenannten CAD/CAM–Blöcke werden dann an den Zahntechniker geliefert und müssen zum individuellen Zahnersatz nur computergestützt bearbeitet werden [1].

Um Festigkeiten und assoziierte Parameter zu erfassen und in Datenblättern zu spezifizieren werden standardisierte Messverfahren benötigt.



Abbildung 1.1.: Beispiele für CAD/CAM–Blöcke, wie sie zur Endbearbeitung verkauft werden. Darunter zu sehen sind die Proben die daraus geschnitten werden können. Nur sehr lange Blöcke (linker Block: $19 \times 19 \times 40 \text{ mm}^3$) eignen sich für die Herstellung von standardisierten 4–Punkt–Biegeproben. Die Abbildung wurde mit Erlaubnis aus [2] reproduziert.

Auf dem Gebiet der Werkstoffprüfung bieten Zahnkeramiken eine besondere Herausforderung. Sie dienen oft nur der Herstellung kleiner Teile wie Kronen und Inlays, die aus Blöcken mit $8 \times 8 \times 15 \text{ mm}^3$ (sog. I8 Blöcke) gefertigt werden. Diese Geometrien sind zur Prüfung durch uniaxiale 3– oder 4–Punkt–Biegung ungeeignet, da Proben für diese Versuche idealerweise 40 mm lang sein sollten. Das absolute Minimum für den Biegeversuch laut ISO 17565, ENV 843–1 und ASTM C 1161 ist 20 mm. Mit kleineren

Proben nimmt die Ungenauigkeit der Lasteinbringung zu und Festigkeiten werden oft grob verfälscht. Zusätzlich kommen noch Reibung und Kanteneffekte stärker zu tragen als bei standardisierten Größen.

Als Alternative werden daher biaxiale Festigkeitsprüfungen wie der piston-on-ring Test aus ASTM F 394-78 vorgeschlagen. Biaxiale Prüfungen haben einige Vorteile gegenüber dem Brechen von Biegebalken, dass zum Beispiel Orientierungen von Defekten in der Keramik nicht bevorzugt werden und eine biaxiale Beanspruchung unter Umständen die Belastung der Keramik im Einsatz besser beschreibt.

Biaxiale Versuchsanordnungen, bei denen eine scheibenförmige Probe auf einem Ring aufliegt und mittig mit einem Ring belastet wird, sind analytisch beschreibbar [3]. Es existiert dafür also eine mathematisch bewiesene Formel, die den Spannungszustand in der Probe beschreibt und dadurch die Umrechnung von Probenlast auf Maximalspannung einfach und korrekt liefert. Ein großer Nachteil ist aber, dass diese Versuche eine nahezu perfekt glatte und parallele Oberfläche der Probe erfordern. Ansonsten herrscht kein kreisförmiger Kontakt, sondern die Probe liegt nur auf wenigen Punkten am Umfang auf, was einen anderen Spannungszustand hervorruft, als die analytische Formulierung beschreibt. Ein weiteres Problem bei Biegeversuchen stellen fixierte Auflager dar. Zwischen einem solchen Auflager und Probe herrscht Reibung, was zusätzliche Spannungen einbringt [4].

Der von Börger *et al.* [5] entworfene 4-Kugelversuch (engl.: Ball-on-3-Balls (B3B) Test) bietet Lösungen für einige der Probleme anderer biaxialer Prüfverfahren, hat aber den Nachteil nicht analytisch beschreibbar zu sein. Seine Auswertung beruht auf der Berechnung des Spannungszustandes mithilfe von Finiten-Elementen-Methoden und der Definition einer empirischen Formel für die Maximalspannung. Dieser Rechenaufwand wird aber durch die Robustheit gegenüber messtechnischen Ungenauigkeiten entlohnt [6]. Es bietet sich also an, den 4-Kugelversuch auf Dentalkeramiken anzuwenden.

1.2. Problemstellung

Es gilt in dieser Arbeit, den 4-Kugelversuch für die Anwendung auf Dentalkeramiken zu definieren. Dazu muss eine Funktion für den Zusammenhang zwischen der aufgebrachtten Druckkraft und der Maximalspannung in der Probe entwickelt werden.

Die Probenserien die von den CAD/CAM-Blöcken stammen unterscheiden sich in einigen Punkten stark zu denen aus der Elektronik, für die der 4-Kugelversuch bisher berechnet wurde.

Der hauptsächliche Unterschied sind die Geometrien der Proben. Die Zahnkeramikblöcke können als rechteckige Prismen oder Zylinder gekauft werden. Proben werden als Platten von den Blöcken getrennt und sind daher mit rechteckigen und kreisförmigen Grundrissen vorhanden. Diese können von einer Vielzahl an Produkten verschiedener Hersteller stammen, die alle unterschiedliche Abmessungen haben können. Die gelieferten Proben haben Abmessungen bzw. Durchmesser von ca. 12 mm, können in der Testserie aber um Millimeter schwanken, wohingegen bei Bauteilen der Mikroelektronik Toleranzen für Werkstücke im Mikrometerbereich liegen! Das Testverfahren muss also auf Schwankungen der Probengeometrien angepasst werden, besonders auf den Zusammenhang zwischen runden und eckigen Proben.

Um Platten und Scheiben deutlich unterschiedlicher Größen vergleichen zu können, muss auch eine Applikation zur Bestimmung des beanspruchten Volumens bzw. der beanspruchten Fläche geschrieben werden. Zuerst gilt es eine solche Berechnung in der

Simulation mit ausreichender Genauigkeit und Geschwindigkeit durchzuführen. Dann müssen die punktwisen zu einer Interpolation zusammengefasst werden, die alle Geometrien und Materialwerte der Proben abdeckt.

Weiters handelt es sich bei den zu untersuchenden Keramiken um verschiedenste Werkstoffe, von Zirkonoxid über diverse Glaskeramiken bis hin zu Kunststoff–Keramik–Nanokomposit Werkstoffen (siehe Tabelle 1.1). Diese haben mitunter weitaus unterschiedliche elastische Eigenschaften. Es gilt deren Auswirkungen auf die Genauigkeit der Festigkeitsprüfung zu überprüfen und gegebenenfalls Schranken für die Prüfbarkeit extrem elastischer Proben zu bestimmen.

Tabelle 1.1.: Werkstoffe von CAD/CAM–Blöcken der Firmen VITA Zahnfabrik (Suprinity und Vitablocs), Ivoclar Vivadent (e.max und Empress), DeguDent (Celtra) und 3M (Lava)

e.max ZirCAD	Yttrium–stabilisiertes Zirkonoxid
e.max CAD	Lithium–Disilikat Glaskeramik
Suprinity	Lithium Silikat/Phosphat Glaskeramik
Celtra Duo	Lithium Silikat/Phosphat Glaskeramik
Empress CAD	Leucit basierendes Aluminosilikatglas
Vitablocs Mark II	Feldspat–verstärktes Aluminosilikatglas
Lava Ultimate	nanoparticuläres vorpolimerisierter Harzkomposit

2. Grundbegriffe zum Versagen von Keramiken

Keramiken sind im Allgemeinen spröde Werkstoffe, die spontan versagen und weitaus niedrigere Zug- als Druckfestigkeiten zeigen [7]. Der Grund für das schnelle Versagen unter Zug ist ein geringer Widerstand gegen Risswachstum. Plastisches Verformen (und damit das übliche Anzeichen eines unmittelbaren Bruches) ist hier energetisch ungünstiger als ein sofortiges Fortschreiten eines Risses durch den gesamten keramischen Körper. Solche Risse gehen von Fehlern im Material, wie Einschlüssen, Poren, Kratzern und Rissen, aus und können nur minimiert, aber nie ganz ausgeschlossen werden.

2.1. Versagensmodell Keramik

Die Linear-elastische Bruchmechanik (LEBM) bietet sich zur Beschreibung des Bruchvorgangs von spröden Werkstoffen an, da sie von vernachlässigbaren plastischen Verformungen ausgeht [8, S. 131].

Risse in der LEBM zeichnen sich durch eine Spannungsüberhöhung an der Rissspitze aus. Sie sind vergleichbar mit einer Kerbe mit nahezu unendlich kleinem Radius. Die Spannungsspitze ist proportional zur umgebenden Spannung und der Wurzel der Risstiefe, wobei der Proportionalitätsfaktor die Geometrie des Risses und seine Lage im Körper berücksichtigt. Der volle Zusammenhang wurde erstmals von Irwin [9] aufgestellt und lautet wie folgt:

$$K = Y \sigma \sqrt{\pi a} \quad (2.1)$$

K ... Spannungsintensitätsfaktor

Y ... Geometriefaktor (berechnet oder tabelliert)

σ ... Spannung

a ... Risstiefe

Da die Spannung im Allgemeinen aber ein Tensor ist, muss man verschiedene Idealfälle betrachten, um ein Verständnis der Auswirkungen dieser Belastungen zu gewinnen. Man geht von drei möglichen Belastungsfällen eines Risses aus, den sogenannten Riss-Moden (siehe Abb. 2.1):

Mode I: Zugspannung normal zu den Rissflanken

Mode II: Scherspannung parallel zu den Rissflanken und normal zur Rissfront

Mode III: Scherspannung parallel zu den Rissflanken und parallel zur Rissfront

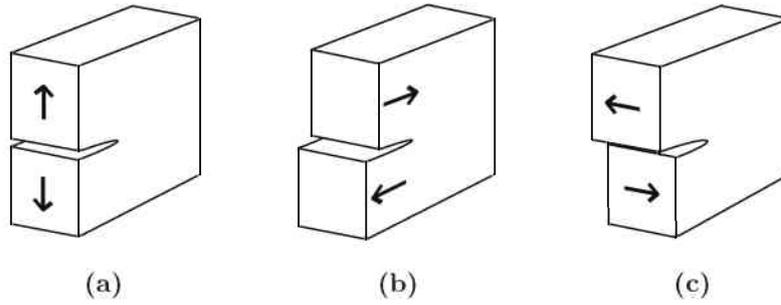


Abbildung 2.1.: Darstellung der Rissmoden: **a)** Mode I, **b)** Mode II, **c)** Mode III

Alle anderen Belastungsfälle haben entweder keine Auswirkung, wie Zugspannungen parallel zu den Rissflanken, oder setzen sich aus den drei Moden zusammen. Als Folge ergeben sich auch drei Spannungsintensitäten K_I , K_{II} und K_{III} :

$$K_I = Y_I \sigma_{zz} \sqrt{\pi a} \quad (2.2)$$

$$K_{II} = Y_{II} \sigma_{zx} \sqrt{\pi a} \quad (2.3)$$

$$K_{III} = Y_{III} \sigma_{zy} \sqrt{\pi a} \quad (2.4)$$

Weil Zugspannungen die Hauptursache für Versagen in Keramiken sind, ist Mode I auch die hauptsächlich behandelte Versagensart.

Das Versagenskriterium nach Griffith [10, 11] besagt, dass sich ein Riss ausbreitet, wenn die freiwerdende Rissausbreitungsenergie G größer ist als die Energie, die das Ausbreiten des Risses kostet, also einen kritischen Wert G_C überschreitet.

$$G \geq G_c \quad (2.5)$$

Diese Energie ist proportional zum Spannungsintensitätsfaktor K und führt so zum Griffith–Irwin Kriterium für Bruch (in diesem Falle dargestellt für Mode I — reine Zugbelastung):

$$G \propto \frac{K_I^2}{E}; \quad G_c \propto \frac{K_{Ic}^2}{E} \quad (2.6)$$

$$K_I \geq K_{Ic} \quad (2.7)$$

Hierbei sind der elastische Modul E und die kritische Spannungsintensität K_{Ic} beides Materialkonstanten!

Durch Einsetzen von (2.2) in (2.7) und Umformen auf σ erhält man:

$$\sigma \geq \frac{K_{Ic}}{Y_I \sqrt{\pi a}} \quad (2.8)$$

Dadurch, dass K_{Ic} eine Materialkonstante ist, ergibt sich eine kritische Spannung σ_c , bei der ein Defekt, der einer Risslänge a_c entspricht, aktiviert wird.

$$\sigma_c = \frac{K_{Ic}}{Y_I \sqrt{\pi a_c}} \quad (2.9)$$

Die Festigkeit einer Keramik hängt also von den Defektgrößen ab, die im Material vorkommen. Fehler im Material sind statistisch verteilt und bewirken so eine statistische Festigkeitsverteilung. Hauptannahme der folgenden statistischen Ansätze ist, dass

die Fehlerhäufigkeit exponentiell mit der Fehlergröße abnimmt. Dies gilt bei einem großen Teil aller untersuchten Keramiken, manchmal kommt es jedoch zu Abweichungen, die dann die Bruchwahrscheinlichkeit stark beeinflussen können [12]. Ausnahmen bilden zum Beispiel Werkstoffe mit Gradienten in der Fehlerverteilung [13] und extrem kleine Proben [14].

2.2. Versagenswahrscheinlichkeit Keramik

Die von Weibull [15] beschriebene Verteilungsfunktion macht die Versagenswahrscheinlichkeit F abhängig von der angelegten Spannung σ und dem Probenvolumen V :

$$F(\sigma) = 1 - \exp\left[-\frac{V}{V_0}\left(\frac{\sigma}{\sigma_0}\right)^m\right] \quad (2.10)$$

Die Weibullparameter σ_0 , V_0 und m leiten sich aus der angenommenen Fehlerverteilung ab [16]. In der Praxis sind sie statistisch bestimmte Werte, wobei die charakteristische Festigkeit σ_0 jene Spannung beschreibt, bei der der Exponent der Verteilung -1 wird, also $F(\sigma_0) = 1 - \exp[-1]$ eine Versagenswahrscheinlichkeit von 63,2% widerspiegelt. Das charakteristische Volumen V_0 gilt dann als das geprüfte Volumen bei dem σ_0 bestimmt wurde und wird durch die gleichartige Geometrie von Probenkörpern (bei denen also V/V_0 immer 1 ergibt) nur benötigt, wenn verschiedene Probengeometrien verglichen werden oder auf Bauteilgrößen umgerechnet wird.

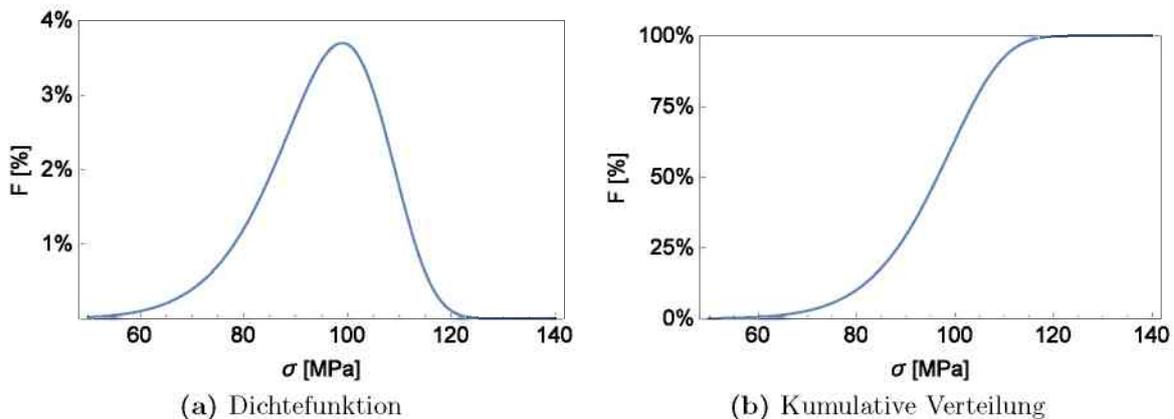


Abbildung 2.2.: Wahrscheinlichkeitsverteilungen einer Weibullverteilung mit $m = 10$ und $\sigma_0 = 100$ MPa.

Der Weibull-Modul m beschreibt die Schärfe der Verteilungsglocke in Abb. 2.2a bzw. die Steigung der Tangente von Abb. 2.2b bei σ_0 . Am anschaulichsten ist er als Steigung der Geraden im sog. Weibulldiagramm (Abb. 2.3). Um diese Gerade zu erlangen, wird zuerst V/V_0 als 1 angenommen und dann durch Umformen und Logarithmieren von (2.10) eine Gleichung (2.11) der Form $y = kx + d$ gewonnen. Aus deren Graphen kann man die Steigung m und der Nullpunkt σ_0 leicht ablesen:

$$\ln \ln\left(\frac{1}{1-F}\right) = m \ln \sigma - m \ln \sigma_0 \quad (2.11)$$

Die Bestimmung von σ_0 und m wird laut Norm [17] anhand von Bruchfestigkeiten einer Stichprobe mithilfe der Maximum-Likelihood Methode durchgeführt und liefert das Weibull Diagramm einer Keramik (ein Beispiel in Abb. 2.4).

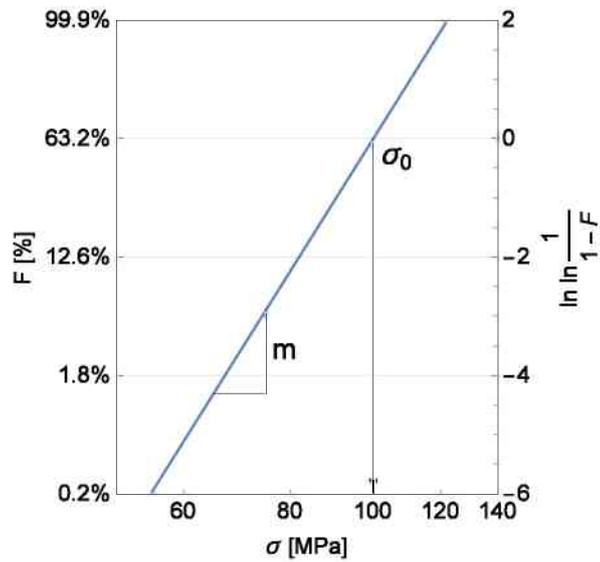


Abbildung 2.3.: Weibulldiagramm einer Verteilung mit $\sigma_0 = 100$ MPa und $m = 10$.

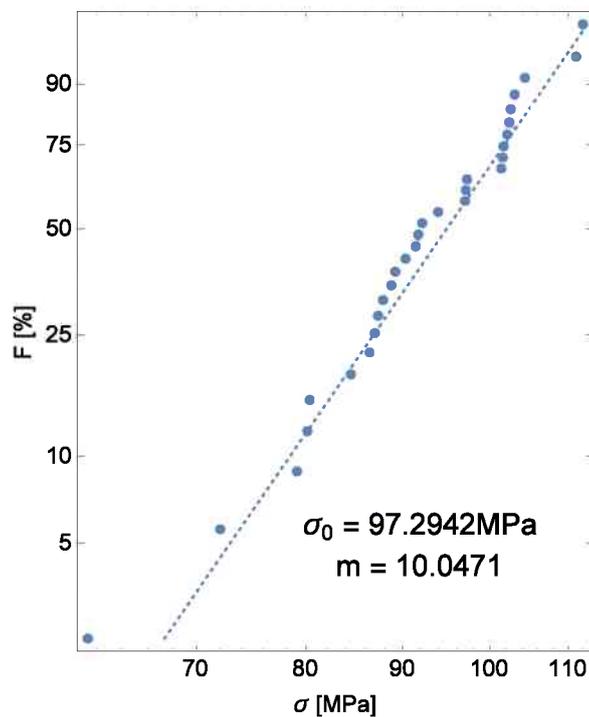


Abbildung 2.4.: Messwerte einer Stichprobe der weibullverteilten Keramik. Die aus der Stichprobe berechneten Parameter stehen unter den Punkten. Die wahren Werte der Weibullparameter sind wieder $\sigma_0 = 100$ MPa und $m = 10$.

2.3. V_{eff} und S_{eff} und deren Einfluss

Die zuvor bestimmten Weibullparameter gelten aber nur in den genauen Umständen der Prüfung. Die Versagenswahrscheinlichkeit F eines Werkstücks, das unter Spannung σ_0 steht, beträgt also nur dann 63,2%, wenn auch das beanspruchte Volumen dieses Werkstücks V_0 und der Spannungszustand gleich sind. Aus diesem Grund werden die Spannungszustände der Messung jenen im Anwendungsfall angeglichen, z.B. wird für einachsige Zugbeanspruchung ein 4-Punkt-Biegeversuch und für zweiachsigen Zug ein 4-Kugelversuch [5] gewählt.

Die beanspruchten Volumina haben aufgrund der statistischen Defektverteilung einen Einfluss auf die Festigkeit. Größere Proben haben ein größeres Volumen und somit eine größere Stichprobe an Defekten, von denen einer bei der gegebenen Spannung kritisch sein kann. Daher haben größere keramische Bauteile bei gleicher Defektverteilung eine niedrigere charakteristische Festigkeit (siehe Abb. 2.5). In der Formel der Versagenswahrscheinlichkeit (2.10) steht daher das Verhältnis des beanspruchten Volumens zu dem der Körper der Stichprobe. Für eine gleich bleibende Ausfallwahrscheinlichkeit, also gleicher Exponent in $F(\sigma)$, lässt sich also folgender Zusammenhang aufstellen:

$$-\frac{V_1}{V_0} \left(\frac{\sigma_1}{\sigma_0} \right)^m = -\frac{V_2}{V_0} \left(\frac{\sigma_2}{\sigma_0} \right)^m \quad (2.12)$$

$$V_1(\sigma_1)^m = V_2(\sigma_2)^m \quad (2.13)$$

$$\frac{\sigma_1}{\sigma_2} = \left(\frac{V_2}{V_1} \right)^{\frac{1}{m}} \quad (2.14)$$

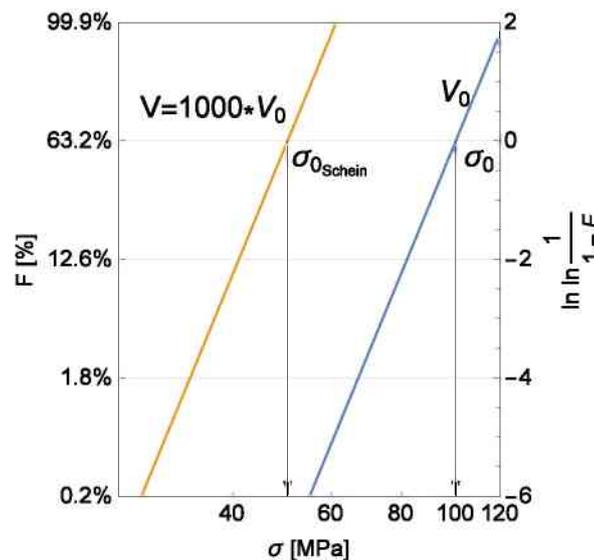


Abbildung 2.5.: Einfluss des Probenvolumens auf das Weibulldiagramm einer Keramik mit $m = 10$ und $\sigma_0 = 100$ MPa. Bei 1000-fachem Volumen zeigt σ_0 eine scheinbare Verringerung um die Hälfte, m bleibt unverändert.

Es gibt mehrere Ansätze mehrdimensionale Belastungsfälle zu behandeln. Bereits etablierte Postprozessoren wie STAU [18] und CARES [19] betrachten die Orientierung von Fehlern in der Keramik als anisotrop. STAU berechnet eine von der Fehlerverteilung und -orientierung abhängige charakteristische Festigkeit σ_0 . CARES berücksichtigt bei der Berechnung der Versagenswahrscheinlichkeit eine Wahrscheinlichkeit für die ungünstige Orientierung kritischer Fehler. Beide Verfahren führen zu erheblichen Aufwand in der Bestimmung von Festigkeitsdaten, weil orientierungsabhängig geprüft werden muss. Auch die Berechnung der effektiven Volumina und Oberflächen wird durch zusätzliche Integrale erschwert.

Eine einfachere Methode schafft einen Zusammenhang zwischen mehrachsigen Spannungszuständen und dem einfachsten Fall des einachsigen Zugs. Bei zylindrischen Proben unter einachsigem Zug wird das gesamte Volumen gleichmäßig belastet, $V(\sigma)^m$ ist also leicht anhand der Probenabmessungen und der Zuglast zu berechnen; im Allgemeinen muss dieser Ausdruck durch ein Integral ersetzt werden [20, S. 145]:

$$\int [\sigma(x, y, z)]^m dV \quad (2.15)$$

Wichtig ist hier zu beachten, dass nur Zugspannungen zum Integral beitragen, einerseits weil Keramiken hauptsächlich unter Zug versagen und andererseits weil bei Biegung sich Druck- und Zugspannung in der Rechnung aufheben würden.

$\sigma(x, y, z)$ kann von einer Skalierungsspannung σ^* abhängig gemacht werden, die aus der makroskopischen Last berechenbar ist (z.B. die Maximalzugspannung einer Biegeprobe):

$$\sigma(x, y, z) = \sigma^* g(x, y, z) \quad (2.16)$$

setzt man diese Gleichung in (2.15) ein, erhält man

$$\int [\sigma^* g(x, y, z)]^m dV, \quad (2.17)$$

welches wieder in der Funktion der Versagenswahrscheinlichkeit anstatt $V\sigma^m$ eingesetzt werden kann:

$$F(\sigma) = 1 - \exp \left[-\frac{1}{V_0} \left(\frac{\sigma^*}{\sigma_0} \right)^m \int [g(x, y, z)]^m dV \right] \quad (2.18)$$

Das Integral von g^m wird als effektives Volumen V_{eff} bezeichnet, mit dem Hochrechnungen auf die Versagenswahrscheinlichkeit größerer Bauteile gemacht werden können:

$$F(\sigma) = 1 - \exp \left[-\frac{V_{eff}}{V_0} \left(\frac{\sigma^*}{\sigma_0} \right)^m \right] \quad (2.19)$$

In der folgenden Arbeit steht uns jedoch nicht die Formfunktion g direkt zur Verfügung, sondern die Spannungsverteilung $\sigma(x, y, z)$ aus der FEM-Simulation. Durch (2.16) erhalten wir für das effektive Volumen:

$$V_{eff} = \int_V [g(x, y, z)]^m dV = \int_V \left[\frac{\sigma(x, y, z)}{\sigma^*} \right]^m dV \quad (2.20)$$

Oft wird auch eine effektive Oberfläche S_{eff} definiert, welche in Betracht zieht, dass viele Keramiken aufgrund von Oberflächenfehlern brechen, die deutlich größer sind als

jene im Volumen. Diese Fehler werden durch Oberflächenbehandlungen eingebracht und unterliegen so einer eigenen Ausfallwahrscheinlichkeit

$$F(\sigma) = 1 - \exp\left[-\frac{S_{eff}}{S_0} \left(\frac{\sigma^*}{\sigma_0}\right)^m\right], \quad (2.21)$$

wobei für S_{eff} wie schon für V_{eff} in (2.20) gilt:

$$S_{eff} = \int_A [g(x, y, z)]^m dS = \int_A \left[\frac{\sigma(x, y, z)}{\sigma^*}\right]^m dS \quad (2.22)$$

Bei der effektiven Oberfläche wird nur mehr die positive Spannung an der Oberfläche der Probe berücksichtigt.

Bei der Integration des effektiven Volumens durch die Spannungsverteilung gilt es immer die richtige Skalierungsspannung σ^* zu wählen. Bedingt durch die Definition ist $\sigma^{*m}V_{eff}$ immer konstant, wenn aber σ^* nicht der errechneten Spannung entspricht, mit der später ein Bauteil belastet wird, wird ein falsches Volumen bestimmt. Durch den Exponenten m , der in der Größenordnung 10 bis 40 liegt, eskalieren auch kleine Fehler schnell. Deswegen ist bei der Wahl der Skalierungsspannung (vor allem beim automatisierten Auslesen dieser in Programmen zur V_{eff} -Berechnung) große Vorsicht geboten!

2.4. Wahl der Skalierungsspannung

Es sollte im Allgemeinen die höchste auftretende Spannung herangezogen werden. Wenn also dieselbe Spannung im gesamten Volumen herrscht, ergibt $\frac{\sigma}{\sigma^*} = 1$, was das effektive Volumen dem tatsächlichen gleich macht.

Es stellt sich die Frage, wie denn die maximale Spannung eines womöglich kompliziert belasteten Bauteils definiert sein soll. Dahingehend gibt es verschiedene Ansätze, um aus dem Spannungstensor eine aussagekräftige Vergleichsspannung σ_V zu berechnen. Diese Vergleichsspannung muss den Einfluss verschiedener Spannungszustände auf das Materialversagen der zu untersuchenden Proben haben.

Bei duktilen Werkstoffen gibt es etwa die Gestaltänderungshypothese nach Huber, von Mises und Hencky [21], oder die Schubspannungshypothese nach Tresca [22]. Im Fall dieser Arbeit handelt es sich um Keramiken, für die unter anderem die folgenden Vergleichsspannungen zutreffen:

Hauptspannungshypothese

Das Versagenskriterium nach Rankine [23, S. 43] geht davon aus, dass Defekte in spröden Werkstoffen durch Zugspannungen aktiviert werden. Dadurch ist nur die größte, sich aus dem Spannungstensor ergebende Zugspannung von Bedeutung für den Bruch einer Keramik.

$$\begin{pmatrix} \sigma_x & \tau_{yx} & \tau_{zx} \\ \tau_{xy} & \sigma_y & \tau_{zy} \\ \tau_{xz} & \tau_{yz} & \sigma_z \end{pmatrix} \Rightarrow \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{pmatrix} \quad (2.23)$$

Der im Allgemeinen aus 6 Komponenten bestehende Spannungstensor $\underline{\underline{\sigma}}$ (Bem.: $\tau_{xy} = \tau_{yx}$) kann durch Rotation des Koordinatensystems auf drei sogenannte Hauptnormalspannungen reduziert werden. Diese stehen in rechten Winkel zueinander und sind

so orientiert, dass gilt: σ_1 entspricht der höchsten Spannung, σ_3 der niedrigsten Spannung (je nach Vorzeichen kann dies eine kleine Zug- oder eine große Druckspannung sein, und umgekehrt für σ_1). σ_2 liegt immer zwischen σ_1 und σ_3 und kann allenfalls gleich einer der beiden sein. Man spricht von *hydrostatischer Spannung*, wenn alle drei Hauptnormalspannungen gleich groß sind.

σ_1 ist also immer die größte Zugspannung, wenn Zugspannungen auftreten. Nach Rankine kann also σ_1 direkt zum Berechnen der Versagenswahrscheinlichkeit der Keramik herangezogen werden, als ob sie unter einachsigen Zug steht.

Principle of Independent Action

Was aber wenn σ_2 gleich groß oder ähnlich groß ist wie σ_1 ? In diesem Fall werden auch ungünstig zur 1. Hauptnormalspannung orientierte Defekte beansprucht, also insgesamt eine größere Anzahl. Dies führt zu einer erhöhten Versagenswahrscheinlichkeit und so zu einer niedrigeren Skalierungsspannung (oder anders: zu einer scheinbar höheren Belastung). σ_2 hat also auch einen Einfluss, der auch in die Vergleichsspannung einfließen muss.

Das Prinzip der unabhängigen Aktivität (PIA) [24, 25, 26] geht davon aus, dass in alle drei Raumrichtungen Spannungen wirken, deren Auswirkung sich überlagern lassen. Dies führt analog zu (2.10) zu einer Versagenswahrscheinlichkeit:

$$F(\sigma) = 1 - \exp \left[-\frac{V}{V_0} \left(\left(\frac{\sigma_1}{\sigma_0} \right)^m + \left(\frac{\sigma_2}{\sigma_0} \right)^m + \left(\frac{\sigma_3}{\sigma_0} \right)^m \right) \right] \quad (2.24)$$

Daraus lässt sich eine Vergleichsspannung ableiten, die in der regulären Weibullstatistik zu derselben Ausfallwahrscheinlichkeit führt:

$$\sigma_V = \sqrt[m]{\sigma_1^m + \sigma_2^m + \sigma_3^m} \quad (2.25)$$

Hierbei sollte beachtet werden, dass der Einfluss von Druckspannungen ein anderer ist als bei Zug. Es gibt verschiedene Ansätze, damit umzugehen. Barnett *et. al* [24] und Freudenthal [25] verwerfen alle Druckspannungen, Stanley [26] gewichtet sie mit dem empirischen Verhältnis zwischen Druck- und Zugfestigkeit.

Aufgrund dessen, dass in den folgenden Berechnungen keine Materialdaten wie etwa die Druck- und Zugfestigkeiten bekannt sind, wird die Vergleichsspannung mit Verwerfen aller Spannungen im Druckbereich kalkuliert.

3. Finite–Elemente–Methode und numerische Integration

Es gilt die Berechnung von V_{eff} aus Abschnitt 2.3 in eine computergestützte Simulation einzubauen, bei der die Spannungsverteilung einer B3B-Probe modelliert wird. Diese Spannungsverteilung dient im weiteren als Berechnungsgrundlage des effektiven Volumens.

3.1. Methode der Finiten Elemente

Analytische Integration ist im Allgemeinen für Computer eine sehr schwierige und in bestimmten Fällen unlösbare Aufgabe. Spannungsberechnungen aufgrund von Ausgangskräften basieren aber auf unbekanntem (und daher allgemeinen) Lastfunktionen in ebenso allgemeinen Geometrien. Integration wird daher oft numerisch implementiert, durch Unterteilen des zu integrierenden Bereichs und Aufsummieren der Funktionswerte zu Näherungslösungen (mehr dazu in Abschnitt 3.2.1).

Diese Unterteilungen heißen im Bezug auf Bauteilberechnungen *Finite Elemente* und geben dem Verfahren der Diskretisierung und Berechnung den Namen *Finite-Elemente-Methode* (FEM) [27].

Die Methoden der Diskretisierung, derer sich die FEM bedient, werden auch in der V_{eff} -Berechnung zur Anwendung kommen und in den folgenden Abschnitten im Detail erklärt. Die Kalkulation der Spannungsverteilung in den einzelnen Elementen beruht auf dem Lösen von Differenzialgleichungsmatrizen deren Abmessungen Millionen mal Millionen Werte betragen können (und in Zukunft mit stärkeren Rechnern noch weiter wachsen werden).

Um zu einer wirklichkeitsnahen Lösung zu kommen müssen außer den Randbedingungen (Krafteinwirkung und Lagerung) noch grundlegende Werkstoffparameter für jedes Element bekannt sein. Im Falle reiner Spannungsberechnung im linear-elastischen Bereich reichen hier der Elastizitätsmodul E und die Querkontraktionszahl ν . In der Regel sind diese über große Bereiche oder ganze Bauteile hinweg konstant, können aber für jedes Element einzeln definiert werden.

Das Endprodukt einer FEM-Rechnung sind folgende Daten:

- Eine Liste aller Elemente und der Berechnungspunkte, die sie ausmachen,
- und eine Liste aller Punkte und deren Koordinaten, Spannungen und Verschiebungen.

Diese Werte können im sogenannten *Postprocessing* abgefragt, gespeichert, weiterverarbeitet oder über graphische Verarbeitung anschaulich gemacht werden.

Im Laufe dieser Arbeit wird das FEM-Programm *ANSYS* [28] verwendet, um die Spannungsdaten zu generieren und Variation von Startparametern zu automatisieren. Die *APDL*-Oberfläche von *ANSYS* ermöglicht das Programmieren von Script-Files die nachvollziehbar sind und bei anderen Benutzern später identische Ergebnisse liefern.

3.2. Numerische Berechnung effektiver Volumen und Oberflächen

Die selben Probleme, die von abgeschlossenen FEM-Programmen gelöst werden, stellen sich auch in der Berechnung des effektiven Volumens bzw. der effektiven Oberfläche. Die Lösungsansätze der FEM sind größtenteils auch hier gültig und können daher übertragen werden.

3.2.1. Einfacher Fall — das 2D Element

2D Elemente kommen entweder bei der Berechnung der effektiven Oberfläche in 3D Modellen vor oder bei vereinfachten 2D Berechnungen von V_{eff} durch Multiplikation mit der Elementtiefe.

An ihnen lässt sich das Verfahren der Berechnung am anschaulichsten Vorführen, sodass es für 3D nur geringer Änderungen bedarf, auf die später eingegangen wird.

Geometrie und Transformation

Ausgegangen wird von viereckigen Elementen, die an jeder Ecke einen Knotenpunkt besitzen, der die Ausgangsinformationen der Berechnung trägt. Im Falle der Spannungsberechnung im FEM-Programm sind es die Raumkoordinaten des Knotens, aus denen dann Verschiebungen und im weiteren Spannungen ermittelt werden. Es gibt auch Elemente mit nur drei Ecken, diese machen Integration über das Element (im 2D Fall also Flächenintegration) aber schwieriger aufgrund von geometrischen Abhängigkeiten. Ebenfalls werden oft Zwischenknoten auf den Elementkanten eingebracht, die Verläufe höherer Ordnung ermöglichen. Diese werden hier ebenfalls vernachlässigt, da *ANSYS* im Zuge seiner Berechnungen Spannungsergebnisse nur an den Eckknoten eines jeden Elements speichert.

Numerische Integration von Vierecken ist am einfachsten über die sogenannte Gauß-Quadratur, welche aber nur im Raum $(-1, 1)$ beste Ergebnisse liefert, also für einheitliche Quadrate. Daher müssen die Elemente des Modells durch Transformation vom kartesischen Koordinatensystem (x, y) in ein natürliches Koordinatensystem (ξ, η) in ebensolche Quadrate verwandelt werden (siehe Abb. 3.1).

Das Element wird durch sogenannte Formfunktionen (3.1) beschrieben, die im zugehörigen Knoten 1 ergeben, in allem anderen Knoten 0.

$$h_i(\xi, \eta) = \frac{1}{4} \cdot (1 + \xi_i \xi)(1 + \eta_i \eta) \quad (3.1)$$

wobei hier gilt:

h_i ... Formfunktion des i ten Knotens

ξ_i, η_i ... natürliche Koordinaten des i ten Knotens, siehe Abb. 3.1b

ξ, η ... natürliche Koordinaten zum Interpolieren

Durch Überlagerung aller Formfunktionen der Elementknoten wird ein linearer Verlauf von Elementwerten bezweckt (in (3.2) und (3.3) sind diese Elementwerte die kartesischen Koordinaten x_i bzw. y_i , später gilt dasselbe für die Spannungswerte).

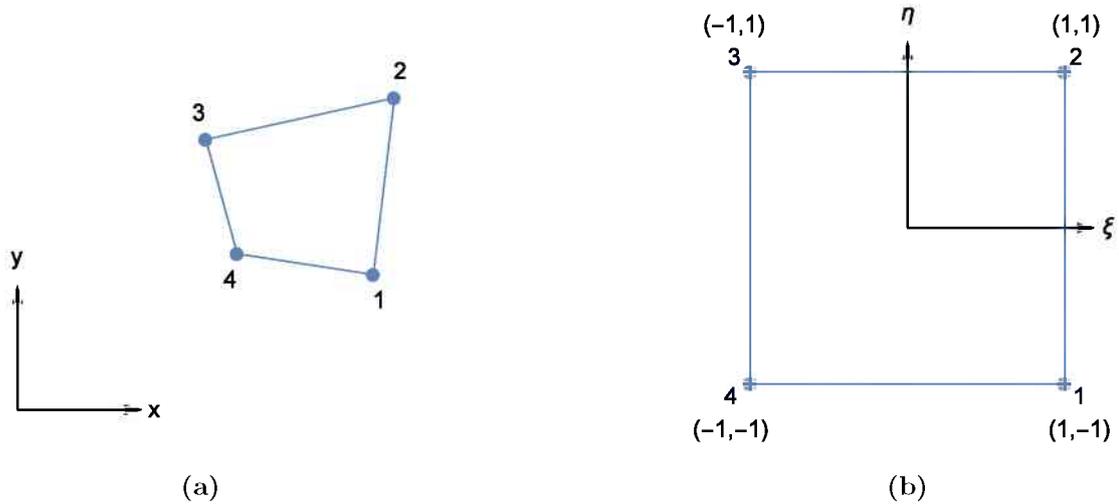


Abbildung 3.1.: Transformation eines viereckigen 2D Elements vom kartesischen (a) in das natürliche Koordinatensystem (b).

$$x = \frac{1}{4} \cdot [x_1(1 + \xi)(1 - \eta) + x_2(1 + \xi)(1 + \eta) + x_3(1 - \xi)(1 + \eta) + x_4(1 - \xi)(1 - \eta)] \quad (3.2)$$

$$y = \frac{1}{4} \cdot [y_1(1 + \xi)(1 - \eta) + y_2(1 + \xi)(1 + \eta) + y_3(1 - \xi)(1 + \eta) + y_4(1 - \xi)(1 - \eta)] \quad (3.3)$$

Um in Rechnungen die Transformation in das natürliche Koordinatensystem nutzen zu können, bedarf es der Jacobi-Determinante die im 2D-Fall wie folgt aussieht:

$$J^{(e)}(\xi, \eta) = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{vmatrix} \quad (3.4)$$

In einer Zeile wird J zu:

$$J^{(e)}(\xi, \eta) = \frac{\partial x}{\partial \xi} \cdot \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \cdot \frac{\partial y}{\partial \xi} \quad (3.5)$$

Gauß–Legendre–Quadratur

Um S_{eff} wie in (2.22) beschrieben integrieren zu können, muss zuerst ein Spannungsverlauf im Element definiert werden. Dazu werden wieder die Vergleichsspannungen und Formfunktionen der einzelnen Punkte überlagert und ergeben so die Elementspannung $f^{(e)}$:

$$f^{(e)}(\xi, \eta) = \frac{1}{4} \cdot [S_1(1 + \xi)(1 - \eta) + S_2(1 + \xi)(1 + \eta) + S_3(1 - \xi)(1 + \eta) + S_4(1 - \xi)(1 - \eta)] \quad (3.6)$$

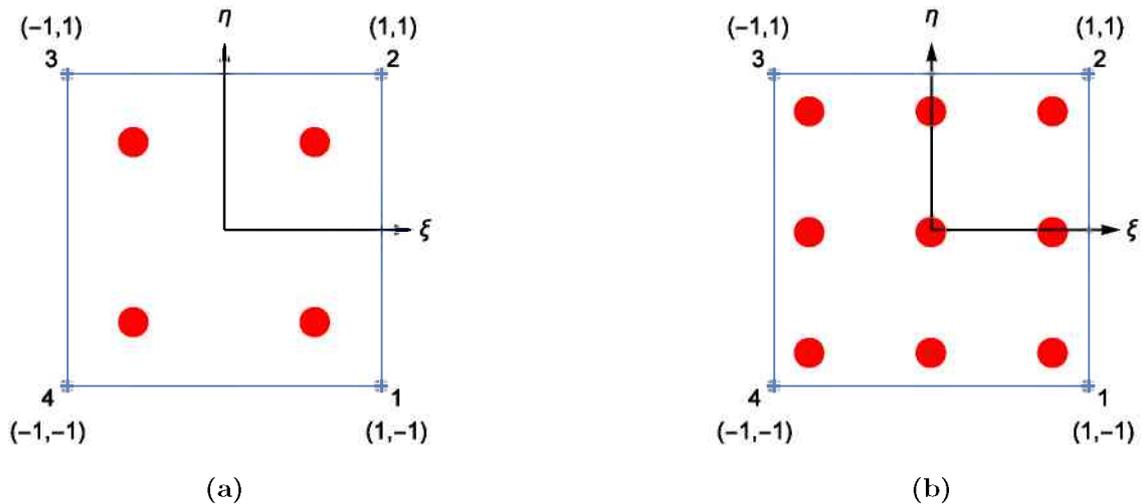


Abbildung 3.2.: Darstellung der Integrationspunkte der Gauß-Quadraturen vom Grad 2 (a) und 3 (b).

$$\begin{aligned}
 S_{eff}^{(e)} &= \int_A \left[\frac{f^{(e)}(\xi, \eta)}{\sigma^*} \right]^m dS = \iint_A \left[\frac{f^{(e)}(\xi, \eta)}{\sigma^*} \right]^m dx dy = \\
 &= \int_{-1}^1 \int_{-1}^1 \left[\frac{f^{(e)}(\xi, \eta)}{\sigma^*} \right]^m \cdot J^{(e)}(\xi, \eta) d\xi d\eta
 \end{aligned} \tag{3.7}$$

Zur Lösung dieses Integrals wird die Gauß-Legendre-Quadratur gewählt, die im Raum $(-1, 1)$ sehr gute Näherungslösungen liefert.

Die Integrationspunkte werden durch den Grad der Gauß-Quadratur vorgegeben und lassen sich aus den Nullpunkten der Legendre-Polynome errechnen.

Tabelle 3.1.: Stützpunkte der Gauß-Legendre-Quadratur mit Grad 2.

Index	Stützpunkte r_i	Gewichtungen w_i
1	-0.5773502691896	1.0
2	0.5773502691896	1.0

Tabelle 3.2.: Stützpunkte der Gauß-Legendre-Quadratur mit Grad 3.

Index	Stützpunkte r_i	Gewichtungen w_i
1	-0.7745966692414	0.55555555555555
2	0.0	0.88888888888888
3	0.7745966692414	0.55555555555555

Tabellen 3.1 und 3.2 zeigen Tabellenwerte für die Stützpunkte und Gewichtungen der Gauß-Quadraturen vom Grad 2 und 3. In mehrdimensionalen Fällen werden die

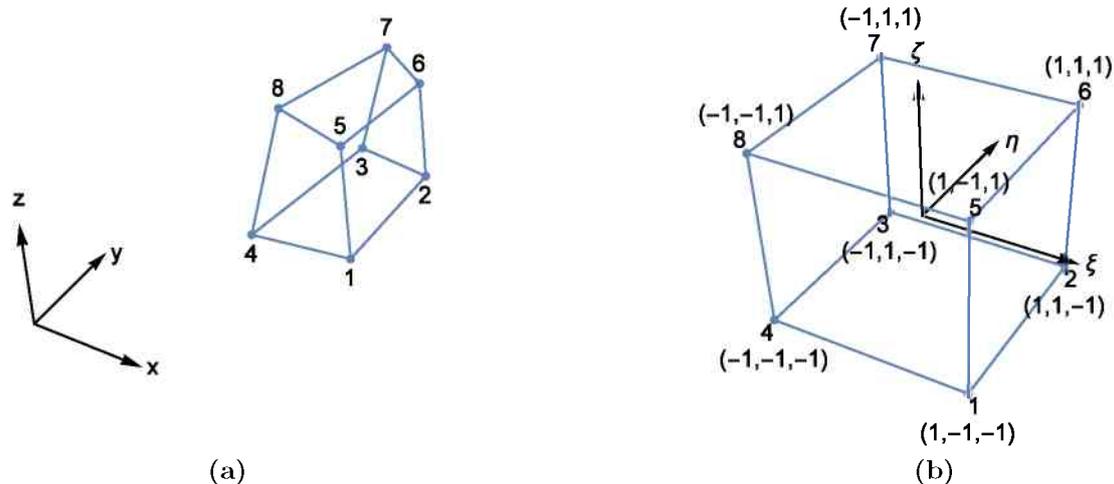


Abbildung 3.3.: Transformation eines achteckigen 3D Elements vom kartesischen (a) in das natürliche Koordinatensystem (b)

Punkte dann in alle Richtungen iteriert, die Gewichtungen sind dann das Produkt der Einzelgewichte der Indizes im Raum. Abb. 3.2 zeigt die Ausweitung der Stützpunkte in 2 Dimensionen für Grade 2 und 3.

Das numerische Integral wird dann zu einer Summe, die in 2D und für Gauß (3) so aussieht:

$$S_{eff}^{(e)} = \sum_{i=1}^3 \sum_{j=1}^3 \left[\frac{f^{(e)}(r_i, r_j)}{\sigma^*} \right]^m \cdot w_i \cdot w_j \cdot J^{(e)}(r_i, r_j) \quad (3.8)$$

Wird diese Berechnung für jedes Element durchgeführt und aufsummiert, erhält man die effektive Oberfläche S_{eff} .

3.2.2. Erweiterung auf 3 Dimensionen

Volumenelemente

Im 3D Fall ist für Volumenelemente der Ablauf völlig gleich, nur unter Zusatz einer weiteren Dimension. Elemente haben daher 8 Eckknoten und werden vom Raum (x, y, z) in den natürlichen Koordinatenraum (ξ, η, ζ) transformiert (siehe Abb. 3.3)

Dadurch wird auch die Formfunktion entsprechend erweitert:

$$h_i(\xi, \eta, \zeta) = \frac{1}{8} \cdot (1 + \xi_i \xi)(1 + \eta_i \eta)(1 + \zeta_i \zeta) \quad (3.9)$$

mit:

h_i ... Formfunktion des i ten Knotens

ξ_i, η_i, ζ_i ... natürliche Koordinaten des i ten Knotens

ξ, η, ζ ... natürliche Koordinaten zum interpolieren

Tabelle 3.3 listet die natürlichen Koordinaten der einzelnen Eckknoten auf.

Tabelle 3.3.: Natürliche Koordinaten der Eckknoten eines 3D Elements.

Index i	1	2	3	4	5	6	7	8
ξ_i	-1	1	1	-1	-1	1	1	-1
η_i	1	1	-1	-1	1	1	-1	-1
ζ_i	-1	-1	-1	-1	1	1	1	1

Auch die Transformationsregeln für Koordinaten werden dementsprechend größer, können aber vereinfacht dargestellt werden:

$$x = \sum_{i=1}^8 x_i h_i; \quad y = \sum_{i=1}^8 y_i h_i; \quad z = \sum_{i=1}^8 z_i h_i \quad (3.10)$$

Die Jacobi-Determinante wächst von 4 auf 9 Elemente:

$$J^{(e)}(\xi, \eta, \zeta) = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{vmatrix} \quad (3.11)$$

und lässt sich expandiert darstellen als:

$$J^{(e)}(\xi, \eta, \zeta) = \left[\frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} \frac{\partial z}{\partial \zeta} + \frac{\partial y}{\partial \xi} \frac{\partial z}{\partial \eta} \frac{\partial x}{\partial \zeta} + \frac{\partial z}{\partial \xi} \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \zeta} \right] - \left[\frac{\partial x}{\partial \zeta} \frac{\partial y}{\partial \eta} \frac{\partial z}{\partial \xi} + \frac{\partial y}{\partial \zeta} \frac{\partial z}{\partial \eta} \frac{\partial x}{\partial \xi} + \frac{\partial z}{\partial \zeta} \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi} \right] \quad (3.12)$$

Der Spannungsverlauf wird gleich den Koordinaten erweitert:

$$f^{(e)}(\xi, \eta, \zeta) = \sum_{i=1}^8 S_i h_i \quad (3.13)$$

Das Integral des effektiven Volumens wird von (2.20) übernommen und transformiert

$$\begin{aligned} V_{eff}^{(e)} &= \int_V \left[\frac{f^{(e)}(\xi, \eta, \zeta)}{\sigma^*} \right]^m dV = \iiint_V \left[\frac{f^{(e)}(\xi, \eta, \zeta)}{\sigma^*} \right]^m dx dy dz = \\ &= \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \left[\frac{f^{(e)}(\xi, \eta, \zeta)}{\sigma^*} \right]^m \cdot J^{(e)}(\xi, \eta) d\xi d\eta d\zeta \end{aligned} \quad (3.14)$$

Dieses wird dann genau so numerisch integriert wie schon beim 2D Element, nur mit einer zusätzlichen Dimension (hier wieder für eine Gauß-Quadratur 3. Grades):

$$V_{eff}^{(e)} = \sum_{i=1}^3 \sum_{j=1}^3 \sum_{k=1}^3 \left[\frac{f^{(e)}(r_i, r_j, r_k)}{\sigma^*} \right]^m \cdot w_i \cdot w_j \cdot w_k \cdot J^{(e)}(r_i, r_j, r_k) \quad (3.15)$$

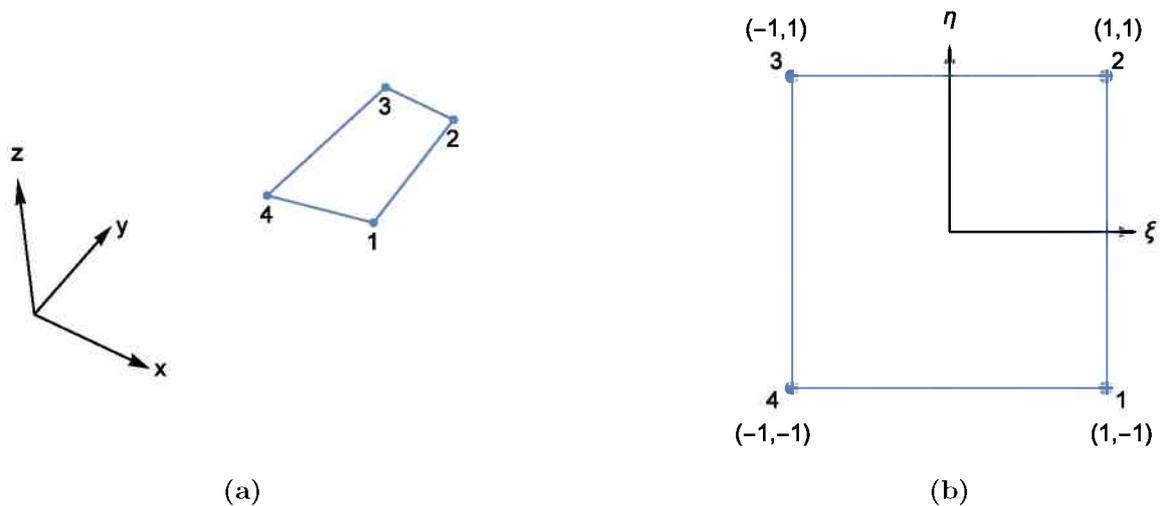


Abbildung 3.4.: Transformation eines viereckigen Flächen Elements von 3D (a) in natürliche 2D Koordinaten (b).

Flächenelemente

2D Elemente im 3D Raum bieten eine besondere Problemstellung (ebenso 1D Elemente, aber hier ist der Vorgang wieder identisch). Im natürlichen Koordinatensystem sind diese Elemente flach, haben also keine ζ -Achse, im karthesischen System haben sie aber sehr wohl alle drei x -, y - und z -Achsen (siehe Abb. 3.4). Dies führt zu den Transformationsregeln:

$$x = \frac{1}{4} \cdot [x_1(1 + \xi)(1 - \eta) + x_2(1 + \xi)(1 + \eta) + x_3(1 - \xi)(1 + \eta) + x_4(1 - \xi)(1 - \eta)] \quad (3.16)$$

$$y = \frac{1}{4} \cdot [y_1(1 + \xi)(1 - \eta) + y_2(1 + \xi)(1 + \eta) + y_3(1 - \xi)(1 + \eta) + y_4(1 - \xi)(1 - \eta)] \quad (3.17)$$

$$z = \frac{1}{4} \cdot [z_1(1 + \xi)(1 - \eta) + z_2(1 + \xi)(1 + \eta) + z_3(1 - \xi)(1 + \eta) + z_4(1 - \xi)(1 - \eta)] \quad (3.18)$$

was zu einer nicht-Quadratischen Jacobi-Matrix führt:

$$J^{(e)}(\xi, \eta) = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \end{pmatrix} \quad (3.19)$$

Von einer solchen Matrix kann man keine Determinante bilden, daher muss man sich der sogenannten Gram'schen Determinante bedienen.

$$\text{Gram}(A) = \text{Det}(A^T A) \quad (3.20)$$

Hierbei ist A eine Matrix mit Abmessungen $m \times n$, wobei $m > n$ gelten muss. Die Gram'sche Matrix $A^T A$ ist immer quadratisch und hat daher selbst eine Determinante.

Tabelle 3.4.: Berechnungszeit von je 4 Berechnungen (S_{eff} und V_{eff} auf Basis von σ_1 und σ_{PIA}) eines 4-Punkt-Biegebalken mit variierender Netzgröße. Verwendet wurden quadratische Hexaeder-Elemente.

Meshgröße [mm]	Elemente [-]	Fortran		APDL	
		wall time [s]	CPU time [s]	wall time [s]	CPU time [s]
1.0	552	0	1.1681	6	5.5123
0.5	4320	3	4.3403	12	21.8094
0.25	34560	23	22.6814	64	110.255
0.166667	116640	74	70.0645	265	370.763
0.125	276480	171	157.258	2060	1194.75
0.1	540000	330	303.004	5277	2213.41
0.083333	933120	567	517.891	10464	4078.72

Diese ist aber immer noch das Ergebnis einer Multiplikation der Jacobi-Matrix mit sich selbst, daher muss noch die Wurzel der Gram'schen Determinante gezogen werden, um ein Äquivalent einer nicht-quadratischen Jacobi-Determinante zu erhalten.

$$G(\xi, \eta) = \sqrt{\text{Gram}(J^{(e)}(\xi, \eta))} = \sqrt{\text{Det}(J^T J)} \quad (3.21)$$

$$S_{eff}^{(e)} = \sum_{i=1}^3 \sum_{j=1}^3 \left[\frac{f^{(e)}(r_i, r_j)}{\sigma^*} \right]^m \cdot w_i \cdot w_j \cdot G(r_i, r_j) \quad (3.22)$$

3.3. Fortran Programm

Die Programmiersprache Fortran eignet sich zum Implementieren der zuvor ausgearbeiteten Methoden weitaus besser als die ANSYS eigene Sprache APDL. Grund dafür sind unter anderem die Art wie die Programme Speicherzugriffe verarbeiten.

Fortran ermöglicht die Nutzung sogenannter *elemental intrinsic functions*, die auf Arrays angewandt, alle Speichereinträge sehr effizient bearbeiten. Zum Erstellen des Berechnungsprogramms wurde die Iteration Fortran90 als Kompromiss aus Einfachheit und Funktionalität gewählt. Tabelle 3.4 vergleicht die Berechnungszeit desselben Programms in Fortran90 und APDL.

Bemerkenswert ist hierbei einerseits die kürzere Zeit, die der Prozessor aufwenden musste (*CPU time*), um die Werte zu errechnen (hier ein Faktor 5 bis 8), andererseits die bessere Umsetzung von Prozessorzeit in wirklich verstrichene Zeit (*wall time*). Der geringe Faktor zwischen CPU time und wall time deutet auf die höhere Effizienz bei Speicherzugriffen hin.

3.3.1. Aufbau des Programms

Das Fortran-Programm besteht aus 2 Files (siehe Anhang B). Im Hauptprogramm `Effektives_Volumen.f90` werden die Unterscheidung von Geometrien und die Ausgabe der Ergebnistabelle veranlasst. Das Modul `overhead.f90` definiert globale Variablen und beinhaltet alle Funktionen des Hauptprogramms. Die verschiedenen Berechnungen für die effektiven Volumina in verschiedenen Dimensionen befinden sich hier. Die

beiden Quellcode Files werden beim Kompilieren zu einem Programm `Effektives_Volumen.exe` zusammengefügt (wird dem Programm auch auf Unix-Plattformen die Endung `.exe` gegeben, muss man den automatischen Programmaufruf bei einem Plattformwechsel nicht ändern).

Außerdem gibt es ein APDL-Macro `x.Effektives_Volumen.mac`, das die Schnittstelle zwischen `ANSYS` und dem Programm bildet. Das Macro bereitet die Element- und Knotentabellen vor und gibt diese (und die Berechnungsparameter) als Outputfiles aus, die vom Hauptprogramm eingelesen werden.

Ein Aufruf des Programms läuft wie folgt ab:

Vorbereiten des Modells

Im APDL Code der Berechnung müssen zwei Komponenten definiert werden. *Elemente_Veff* besteht aus allen Elementen, die für die Berechnung miteinbezogen werden sollen. Wichtig ist dabei das Wegschneiden von Krafteinleitungspunkten und anderen unrealistischen Singularitäten mit genug Abstand, dass deren Einflussgebiete ausklingen können. *Knoten_Seff* sind alle Knoten, die der effektiven Oberfläche angehören können. Auch hier werden Spannungspitzen wieder ausgeschnitten.

Aufruf des Macros

Nach der Lösung des FEM-Programms wird das APDL-Macro `x.Effektives_Volumen.mac` aufgerufen. Hier sind einige Optionen möglich, die im Code des Macros selbst erklärt werden (siehe Anhang A.10).

Nach Zuteilen von Programmparametern wie der Anzahl an Eckknoten und der Elemente im Modell wird eine Tabelle erschaffen, auf die dann die Knoten der jeweiligen Volumenelemente gespeichert werden. Dasselbe passiert auch für Knoten von Elementflächen, nur muss hier berücksichtigt werden, dass dann nicht jede Fläche selektiert ist. Daher wird bei jeder Fläche eine Abfrage über alle Eckknoten gemacht, die 1 wird wenn der Knoten ausgewählt ist und 0 wenn nicht. Durch Multiplikation der Abfragen aller Eckpunkte werden nur Flächen mit allen Eckpunkten garantiert.

Danach werden die Elementlisten reduziert. Zuvor war jede Zeile äquivalent zu der Nummer des beschriebenen Elements, was aber zu vielen Leerzeilen führt, vor allem bei den Flächen. Daher werden beide Tabellen nach Leerzeilen (in denen alle Werte 0 sind) durchsucht und ohne diese auf neue, kürzere Listen überspielt. Die Nummerierung der Elemente geht dabei verloren, war aber auch nie von Bedeutung.

Zuletzt wird noch eine Tabelle mit den Koordinaten und Spannungswerten eines jeden Knoten belegt. Hier kann nicht reduziert werden, da die Nummerierung für die Zuweisung von Elementen sehr wichtig ist. Daher wird die volle Liste, bei quadratischen Elementen auch mit Zwischenknoten, übergeben.

Diese Tabellen und ein File mit den Programmparametern werden abgespeichert und vom Fortran-Programm aufgegriffen. Das Fortran-Programm wird über einen Systembefehl aufgerufen und ausgeführt.

Initialisieren des Programms

Nach Definieren aller Variablen wird zuerst das Parameter-File eingelesen und alle berechnungsrelevanten Variablen belegt. Danach werden die Tabellen nacheinander

aufgerufen, deren Zeilen gezählt und entsprechende Tabellen im Programmspeicher angelegt.

Da das Programm immer von Quadraten und Hexaedern ausgeht, wird hier eine Vereinfachung eingebracht. Bei Tetraeder-Elementen werden die Tabellen um zusätzliche Wiederholungen bestimmter Eckknoten ergänzt, sodass die Elemente degenerierten Hexaedern entsprechen.

Dann wird, falls eine Maximalspannung angegeben wurde, überprüft ob es markant höhere Werte gibt. Diese würden auf übersehene Singularitäten hindeuten.

Zuletzt werden die Stützpunkte und Gewichtungen der Gauß-Quadratur aus einer internen Datenbank zugeteilt.

Nach diesen vorbereitenden Schritten werden anhand der Dimension der Elemente (2D oder 3D) die V_{eff} - und S_{eff} -Berechnungen aufgerufen.

Berechnung von V_{eff}

In den aufgerufenen Funktionen wird der Ablauf aus Abschnitt 3.2 durchgeführt. Einziger Unterschied zur Beschreibung ist, dass die Berechnung nicht für jedes Element einzeln durchgeführt wird, sondern eine Tabelle mit $n + 1$ Dimensionen für die Berechnung von n -dimensionalen Elementen erschaffen wird. Diese enthält für jedes Element Gaußgradⁿ Berechnungspunkte, die alle simultan belegt werden.

Es ist auch zu beachten, dass innerhalb der Funktionen alle großen Arrays als *allocatable* (zuteilbar) definiert werden. Wenn diese Arrays sofort in voller Größe in den Speicher geschrieben werden, passiert dies im sogenannten *Stack* (Stapel) des Speichers. Dieser ist aber begrenzt groß und Überschreitung des Stacks führt zum Programmabbruch. Deswegen werden die Arrays erst definiert, dann zugeteilt, was sie im *Heap* (Haufen) des Speichers platziert. Der Heap ist weitaus robuster gegenüber sehr großen Datenmengen.

Das Integral über das Volumen wird zur Zeitersparnis sofort für die Hauptspannungshypothese und die PIA-Vergleichsspannung durchgeführt und dann über alle Berechnungspunkte aller Elemente aufsummiert.

Weil das Integral von V_{eff} den unbekanntem Weibull-Modul beinhaltet, wird V_{eff} für alle Module bis zu einem vorgegebenen Höchstwert errechnet und als Tabelle abgespeichert.

Die Berechnung wird für V_{eff} und S_{eff} durchgeführt und ergibt so 4 Tabellen, die in ein Outputfile gespeichert werden. Der Name dieses Files wurde schon beim Aufruf des Macros festgelegt und als Parameter übergeben.

3.3.2. ANSYS UPF

Um eine engere Schnittstelle zwischen selbstgeschriebenen Programmen und *ANSYS* zu schaffen, können diese als sogenannte *User-Programmable Features* (UPFs) implementiert werden. UPFs ermöglichen die direkte Übergabe von Daten an ein Programm ohne Zwischenspeichern auf der Festplatte. Das UPF wird so zu einer Erweiterung von *ANSYS* und kann als *ANSYS* Routine immer zugegriffen werden.

Um ein UPF in *ANSYS* integrieren zu können, müssen aber strikte Bedingungen erfüllt werden. Der Compiler mit dem das Programm kompiliert werden kann, um als UPF in Frage zu kommen, wird für jede Version von *ANSYS* vorgegeben, und kann sich auch mit jeder neuen Version ändern. Auch die Namensgebung der Files ist festgelegt auf *USER01*, *USER02* etc.

Eine Implementierung der zuvor beschriebenen V_{eff} -Berechnung konnte in *ANSYS 16.2* nicht durchgeführt werden, da *ANSYS* trotz Unterstützung der Sprache Fortran90 Files mit der Endung *.f90 nicht compilieren konnte. Es wurde nur die Fortran-Endung *.F akzeptiert, die allerdings dem Compiler formatierten Fortran77 Code erzwingt, dem einige Funktionalitäten des moderneren Fortran90 fehlen. Daher konnte nur durch Aufruf einer externen Programmdatei gearbeitet werden.

Es kann aber auch argumentiert werden, dass eine solche externe Datei einfacher zu bedienen ist. Sie hängt nicht von bestimmten Compilern ab und kann auf jeder Plattform eingesetzt werden, ohne spezielle Voraussetzungen erfüllen zu müssen.

4. FEM–Berechnungen an Platten und Scheiben

Die Dentalkeramikproben wurden in zwei Geometrievarianten bereitgestellt. Es handelte sich um Platten mit kreisförmigen und rechteckigen Grundflächen. Die runden Proben werden im Folgenden als *Scheiben* bezeichnet und die eckigen als *Platten*.

An beiden wurden dieselben Untersuchungen durchgeführt, die unterschiedlichen Geometrien erforderten jedoch leichte Abweichungen in der Ausführung. Auf diese wird dort eingegangen, wo sie auftreten.

4.1. Auflagergeometrie

Zeitgleich mit der Implementierung der Simulation wurden zwei physische Auflager konstruiert, um die Ergebnisse der Berechnungen sofort für die Auswertung von Versuchen einsetzen zu können. Diese Experimente sind nicht Thema dieser Arbeit und werden in einem separaten Paper veröffentlicht [2].

Für Scheiben ist das Design des Auflagers bereits Routine: Einer Schablone aus Messing wurden zwei Vertiefungen gebohrt, die je Platz für die Probe und die Lagerkugeln bieten. Die Lastkugel befindet sich in einer durchgehenden Bohrung unter der Probe und wird durch einen Stahlstempel in Position gehalten. Beim Einsetzen der Probe und Kugeln befindet sich ein Keil unter der Schablone, sodass die passende Vertiefung die Probe fixiert. Sie sollte dabei aber nicht am Boden aufliegen, sondern nur von der Mantelfläche der Bohrung fixiert werden und auf der Lastkugel aufliegen (siehe Abb. 4.1). Nach Aufbringen einer Vorlast wird der Keil entfernt, die Schablone rutscht entlang des Stempels nach unten und die Kugeln und Scheibe werden nur durch den Anpressdruck in Position gehalten.

Für Platten wurde dieses System übernommen, mit dem Unterschied, dass Platten verschiedener Abmessungen eingepasst werden mussten. Aus den runden Bohrungen für die Probe und Lagerkugeln wurden so eine rechteckige Ausfräsung mit Ausnehmungen für die drei Lagerkugeln. So konnte eine Orientierung der Platte zum Auflager garantiert werden. Abb. 4.2 zeigt den Konstruktionsplan der Ausfräsung für die Schablone. Diese ist an die größte Platte in der Versuchsserie angepasst, aber so zentriert, dass die kleinste Platte der Serie mittig auf der Lastkugel liegt, wenn sie einer markierten Ecke anliegt. Abb. 4.3 zeigt den Konstruktionsplan der Ausfräsung für die Schablone. Alle anderen Platten werden somit exzentrisch belastet, was in die FEM–Berechnungen schon mit eingerechnet wurde.

4.2. Vereinfachen des Modells

FEM–Rechnungen können nicht die volle Wirklichkeit darstellen. Es lohnt sich auch nicht, ein möglichst umfangreiches Modell zu simulieren, wenn z.B. Reibung und Kontaktprobleme nicht zur Lösung beitragen. Solche Einflüsse verlängern die Rechenzeit von Simulationen um Größenordnungen.

Es ist daher empfehlenswert, die Proben durch ein möglichst einfaches Modell zu beschreiben. Dieses fängt bei der Geometrie an. Die Scheiben haben durch die drei

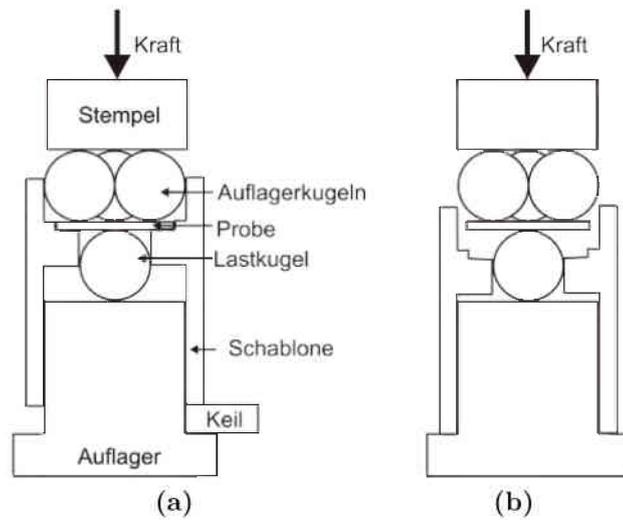


Abbildung 4.1.: Seitenschnitt eines Auflagers. **(a)** zeigt die Position der Schablone bei Vorbereitung der Probe und beim Aufbringen der Vorkraft. **(b)** nach Entfernen des Keils rutscht die Schablone nach unten weg und die Probe und Kugeln sind frei beweglich, nur die Vordruckkraft hält sie in Stellung. Daraufhin kann die Kraft bis zum Versagen der Probe erhöht werden.

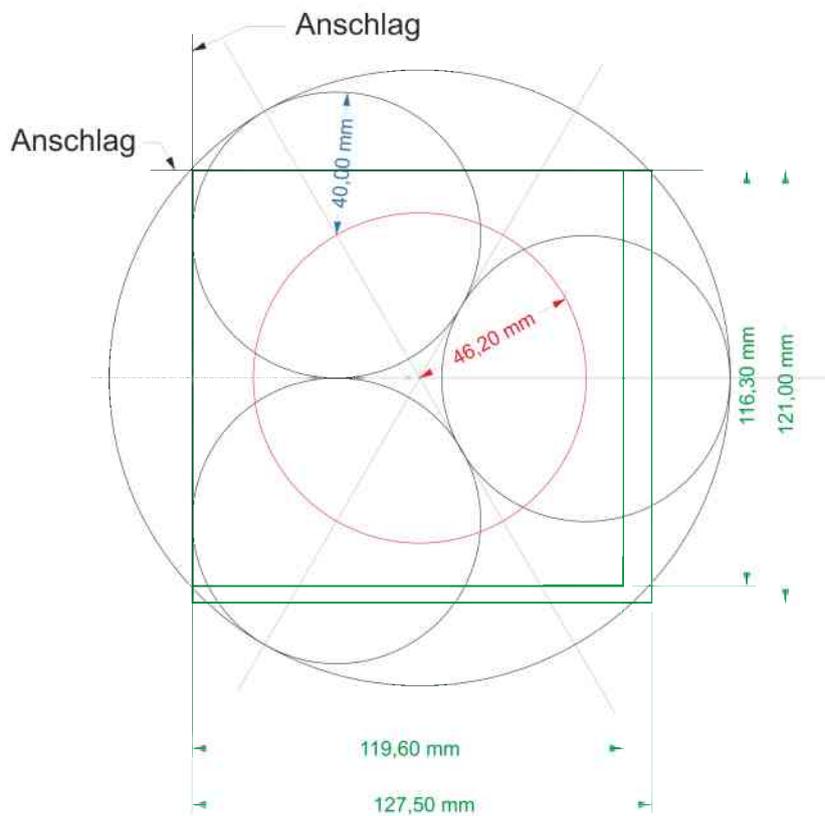


Abbildung 4.2.: Konstruktionsplan der Schablone. Die grünen Rechtecke geben die minimalen und maximalen Größen für Proben vor. Links oben ist der Anschlag eingezeichnet.

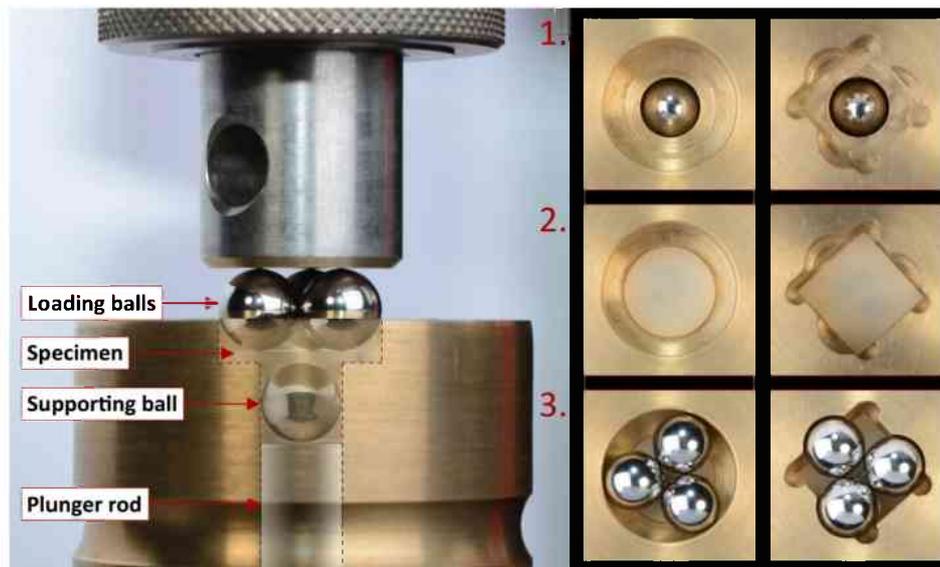


Abbildung 4.3.: Gefertigtes Auflager mit Schablone. Auf der rechten Seite wird der Ablauf des Einlegens einer Scheibe und einer Platte gezeigt. Für die rechteckige Probe befindet sich der Anschlag hier auf der untersten Ecke. Eine Kerbe außerhalb des Bildes markiert diese für einfache Handhabung. Bild mit Erlaubnis aus [2] reproduziert.

Lagerkugeln drei Spiegelebenen und eine dreifache Rotationssymmetrie. Dadurch lassen sie sich auf einen Kreissektor reduzieren, der $\frac{1}{6}$ der vollen Platte darstellt (siehe Abb. 4.4 und 4.5a).

Die Platten besitzen durch Kombination aus rechteckiger Geometrie und dreifacher Lagerung nur eine Spiegelebene. Die Konstruktion des Auflagers nimmt allerdings eine Ecke als Ausrichtungspunkt. Diese Ecke hat immer einen fixen Abstand zu den Kugeln, die gegenüber liegende Ecke hat einen variablen Überhang. Durch diese azentrische Belastung geht jegliche Symmetrie der Platten verloren und sie müssen voll vernetzt und simuliert werden.

In beiden Fällen wurde das Modell auf Punktlasten reduziert, sodass ein Simulieren des Kontakts mit den Kugeln wegfällt. Die Maximalspannung unter der Lasteinbringung wird durch diese Vereinfachung nicht beeinflusst [5]. Das Spannungsfeld ändert sich hauptsächlich in der unmittelbaren Umgebung der Fixierungspunkte, diese wird aber vor Evaluierung des effektiven Volumens ausgeschnitten und hat also auch keinen weiteren Einfluss.

4.3. Randbedingungen der Simulation

Als die Finiten Elemente der Berechnung wurden SOLID186-Elemente [29] gewählt. Diese würfelförmigen, 20-knotige Elemente Hexaeder ergeben durch die nichtlineare Berechnung von Knotenverschieben eine genauere Spannungsverteilung. Spannungswerte werden von ANSYS aber weiterhin nur auf die Eckpunkte gespeichert, für die V_{eff} -Berechnung ändert sich also nichts.

Beim Vernetzen wurden zuerst die Umgebungen der Kontaktpunkte im Grundriss mit einem sogenannten *mapped mesh* belegt. Dieses ist durch besondere Regelmäßigkeit definiert und liefert über viele Versionen Von ANSYS hinweg immer reproduzierbare

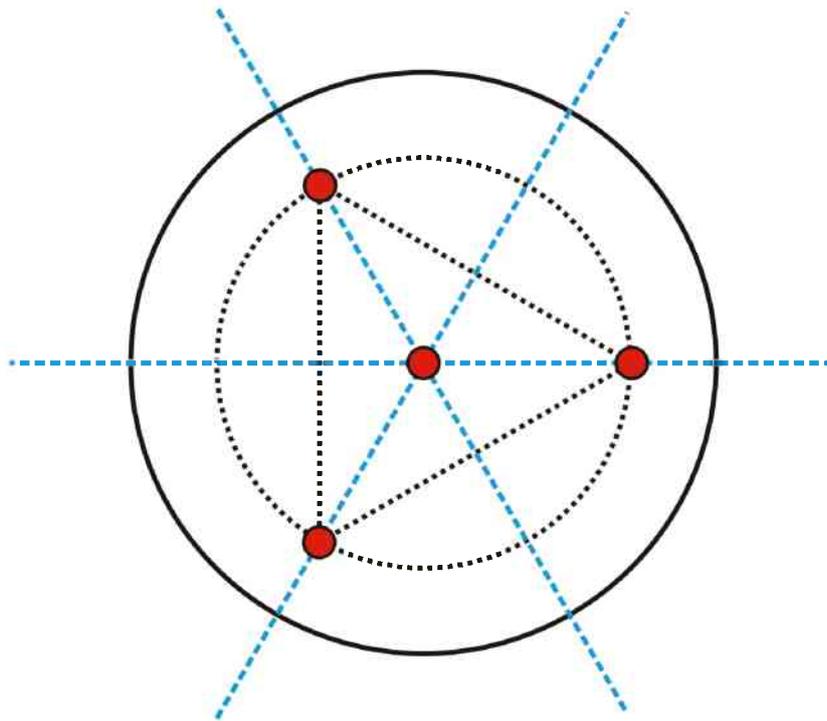


Abbildung 4.4.: Schematische Darstellung einer Scheibe im Auflager. Die roten Punkte symbolisieren die Kontaktpunkte der Kugeln, also die Lagerung und die Kräfteinleitung. Die blau gestrichelten Linien stellen die drei Spiegelebenen dar. Das schwarz gepunktete Dreieck und dessen Umkreis geben eine vereinfachte Darstellung der drei Kontaktpunkte und des Auflagerradius wieder.

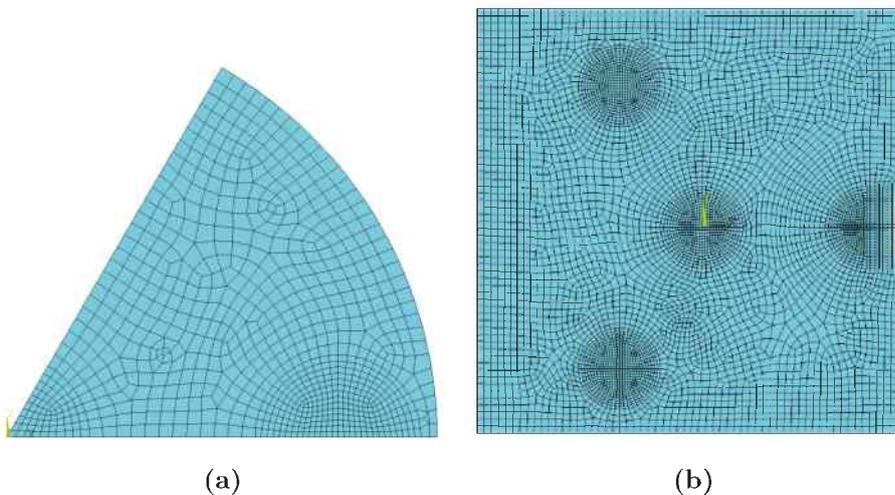


Abbildung 4.5.: Beispielhafte Vernetzungen einer Scheibe (a) und einer Platte (b).

Ergebnisse. Da aber nur 4-eckige und 3-eckige Flächen mit einem *mapped mesh* belegt werden können, wurde der übrige Zwischenraum mit einem 1.5 mal größeren freien Netz gefüllt.

Die Netzgröße wurde anhand der Unterteilung des Randgebietes einer Scheibe definiert und ein Äquivalent für Platten mit gleichem Inkreis errechnet. Wegen des Vollmodells wurde bei Platten generell ein 20% größeres Gitter eingesetzt, um die Elementzahl in der selben Größenordnung zu halten.

Zuletzt wurde dann das 2D Netz extrudiert. Die vertikale Unterteilung musste so gewählt werden, dass keine großen Spannungsgradienten in einzelnen Elementen vorkamen, da diese bei m-fachen Exponenten die Volumenintegration stark beeinflussen können. Im Allgemeinen bedeutete das mindestens 20 (besser 25) Unterteilungen in Z-Richtung. Bei Grenzfällen wurde ein höherer Grad für die Gaußquadratur eingestellt, um bessere Ergebnisse zu erzielen.

Die Lagerung bei Scheiben war eine Fixierung in Z-Richtung. weitere Bewegung wurde durch die Symmetriebedingungen verhindert. Diese wurden durch die eingebaute ANSYS-Funktion **DSYM** realisiert. Die erste Spiegelebene wurde durch die X- und Z-Achsen aufgespannt und konnte so im kartesischen Koordinatensystem definiert werden. Für die um 60°schräge zweite Spiegelebene wurde ein ebenso schräges lokales Koordinatensystem eingeführt und die Symmetrie in diesem festgelegt.

Bei Platten musste ein Knoten zusätzlich in X- und Y-Richtung und ein zweiter in eine der beiden Richtungen fixiert werden, um alle sechs Freiheitsgrade zu fixieren.

4.4. Simulationsablauf

Die volle Berechnung von der Geometrieerstellung bis hin zum Post Processing wurde für jede Kombination von Parametern durchgeführt. Ein File zur Steuerung dieser Kombinationen wurde zuvor mithilfe von *Mathematica* erstellt.

Je nach Ziel einer Simulationsserie wurde dieses Inputfile abgeändert und steuerte so die Berechnung.

Tabelle 4.1.: Durchschnittliche Materialwerte der Dentalkeramikproben aus [2] für die der Versuch ausgelegt wurde. Der E-Modul wird hier nur der Vollständigkeit halber aufgelistet. Er hat auf die Ergebnisse der Berechnungen keine Auswirkung.

Marke	Platten			Scheiben		Materialwerte	
	Länge [mm]	Breite [mm]	Dicke [mm]	Durchmesser [mm]	Dicke [mm]	E-Modul [GPa]	Poissonzahl [-]
Emax ZirCAD	12.06	11.96	1.23	12.03	1.24	207	0.25
Emax CAD	12.63	12.01	1.12	11.93	1.09	102	0.21
Vita Mark II	12.27	11.95	1.15	11.83	1.13	72	0.23
Vita Enamic	12.18	12.05	1.18	11.84	1.23	38	0.25
Vita Suprinity	12.43	12.03	1.22	11.92	1.21	104	0.21

Hauptdatenpool

Für die späteren Parametrisierungen und Interpolationen wurden sowohl die Abmessungen als auch die Querkontraktionszahl in ihrer vollen vorkommenden Bandbreite aus Tabelle 4.2 variiert. Die Dicken der Platten wurden Inkrementen von 0.14 mm (also in 3 Stufen) untersucht. Die Längen und Breiten der Platten wurden durch ein Inkrement von 1.2 mm ebenfalls in drei Größen geteilt. Bei Scheiben waren es fünf Unterteilungen durch einen 0.6 mm Schritt im Durchmesser. Die Poissonzahl wurde für Scheiben und Platten auf 0.2, 0.25 und 0.3 gesetzt. Der E-Modul wurde konstant als 100 GPa setzt, da er auf die idealisierten Berechnungen keinen Einfluss hat. Die Kraft, die auf die Proben wirkt, wurde ebenfalls konstant auf 1 N gehalten. Es wurde für Platten der Abstand der Lastkugel zur fixierten Ecke den Konstruktionszeichnungen des Auflagers entnommen. Dieser war in X-Richtung 6.309 mm und in Y-Richtung 5.815 mm.

Tabelle 4.2.: Parameterspektrum für die zu untersuchenden Proben. *Abmessung* steht hier für die Plattenabmessungen und den Scheibenradius. Die Werte in Klammern stammen von einem Material, dass aufgrund seiner extremen Materialwerte zu große Unsicherheiten aufwirft. Die Lagerkugeln wurden mit konstant 8 mm Durchmesser und aus Stahl ($E = 210$ GPa, $\nu = 0.33$) angenommen.

Abmessung	[mm]	11.7	–	14.1
Dicke	[mm]	0.99	–	1.27
E-Modul	[GPa]	(12) 38	–	207
Poissonzahl	[–]	0.20	–	0.29 (0.47)

Bei den Scheiben führte das zu 45, bei den Platten zu 81 Einzelrechnungen. Es wurden auch für jede dieser Simulationen je die effektive Oberfläche und das effektive Volumen basierend auf PIA- und σ_1 -Kriterium berechnet. Nach den Rechenschritten wurde von jedem Ergebnis ein Bild der Spannungsverteilung an der Zugseite der Probe abgespeichert.

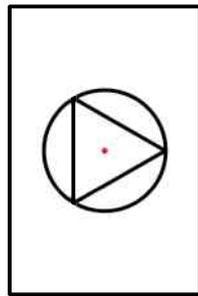
Weitere Plattenorientierungen

Zum Vergleich zwischen Platten und Scheiben wurden die Probenabmessungen laut Tabelle 4.2 abgerastert. Dazu wurden nicht nur die Plattenmaße wie im Hauptdatenpool variiert, es wurde auch zu jeder Platte die Exzentrizität des Krafteinflusses um $1.1 - 1.25 R_a$ gegen die Ecken verschoben. Probendicke und Materialwerte wurden konstant gehalten um nur den Einfluss des überhängenden Materials darzustellen ($t = 1$ mm, $E = 100$ GPa, $\nu = 0.25$).

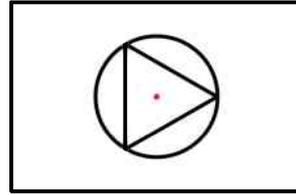
Bei den Platten wurden sechs verschiedene Anordnungen von Probe zu den Kugeln durchgerechnet. Abb. 4.6 zeigt die unterschiedlichen Arten auf denen die Platten belastet werden können. Auch das Ausmaß der Exzentrizität der Belastung wurde variiert, und zwar zwischen 10% und 25% des Auflagerradius.

Abschätzung nichtlinearer Einflüsse

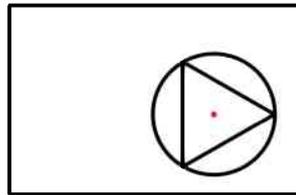
Für die Analyse des Fehlereinflusses von sehr elastischen Proben wurden eine Hertz'sche Lastverteilung im Zentrum und ein variabler Auflagerradius simuliert.



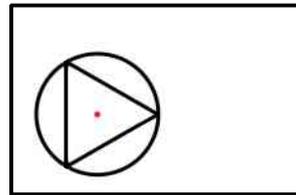
(a)



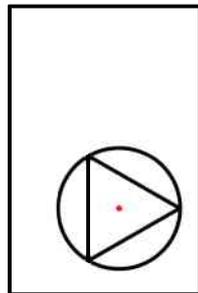
(b)



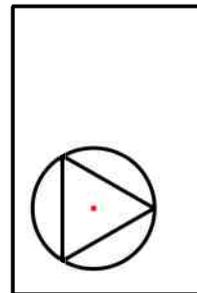
(c)



(d)



(e)



(f)

Abbildung 4.6.: Verschiedene mögliche Anordnungen von drei Auflagerkugeln auf einer rechteckigen Platte (Auflagerkreis und Dreieck als Symbol für die drei Berührungspunkte der Kugeln). Es wurde je zwischen einer Belastung im Hochformat und im Querformat unterschieden. (a) und (b) zeigen ein Belasten der Platte im Schwerpunkt, (c)–(f) nehmen an, dass die Platte an einem Eck angelegt wird und es so bei zwei der Kanten einen fixen Abstand zur Lastkugel gibt. Hier kann unterschieden werden, ob eine oder zwei der Auflagerkugeln näher zur Kante positioniert sein sollen.

Die Hertz'sche Last beschreibt die Kraftverteilung einer realen Kontaktsituation besser als eine Punktlast, da sie Kräfte flächig aufbringt. Dazu musste im Modell der Bereich des Kontakts ausgeschnitten und noch feiner vernetzt werden. Dies passierte mit einem freien Netz, um es schnell gröber werden zu lassen und so die Rechenzeit niedrig zu halten. Auf dieses feine Gitter wurde dann im radialen Koordinatensystem eine flächige Spannungsverteilung mit dem Befehl **SF** aufgebracht.

Die Gesamtkraft unter der Hertz'schen Kraftverteilung wurde weiterhin konstant auf 1 N gehalten, ebenso wie die elastischen Eigenschaften des simulierten Werkstoffs ($E = 100 \text{ GPa}$, $\nu = 0.2$), da der E-Modul in der (immer noch idealisierten) Simulation keinen Einfluss hatte. Der Scheibendurchmesser war konstant 12 mm, die Dicke war 1 mm. Verändert wurde nur der Radius des Kontaktbereichs, um die daraus resultierende Spannungsdifferenz zu plotten.

Die Änderung des Auflagerradius, stellte ein Abrollen der Auflagerkugeln dar. Alle anderen Variablen wurden wieder konstant gehalten.

Beide Radien wurden um 10% des eigentlichen Auflagerradiuses variiert. R_a wurde also bis auf 90% reduziert und a bis auf 0.1 R_a erhöht.

4.5. Ergebnisse und Interpretation

Für jeden Rechendurchgang wurde ein File **Ergebnis.out** erstellt, das die wichtigsten Randbedingungen der Einzelexperimente und die Maximalkraft unter der Lastkugel in einer Tabelle sammelte. Zusätzlich wurde vom Fortran-Programm für jede Iteration eine Tabelle angelegt, die effektive Oberflächen und Volumina für Hauptnormalspannungs- und PIA-Kriterium bis zu einem maximalen Weibullmodul verzeichnet.

Abb. 4.7 und 4.8 zeigt zwei beispielhafte Spannungsverteilungen einer Scheibe und einer Platte.

Durch die hohe Anzahl an Elementen in Dickenrichtung und eine dadurch erforderte Feinheit des Flächen-Netzes hatten Proben bis über 200 000 Elementen bzw. 800 000 Knoten.

Für Scheiben decken sich die Ergebnisse mit vorhergehenden Berechnungen [30] auf unter 1% Differenz. Auch bei Platten stimmten die erreichten Werte mit vorangegangenen Simulationen überein.

Durch die Bestätigung der Richtigkeit von einzelnen Rechnungen konnte die Gesamtmenge an Daten zur weiteren Einbindung in empirischen Formeln und Fehlerabschätzungen zugelassen werden.

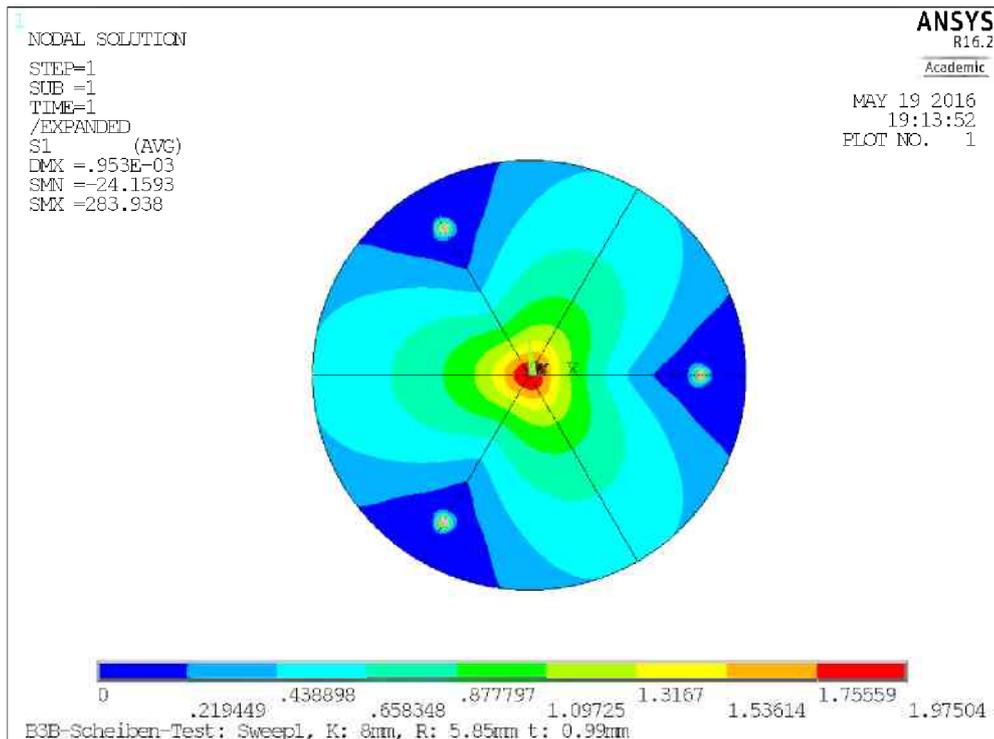


Abbildung 4.7.: Spannungsverteilung einer Scheibe mit Durchmesser 11.7 mm, Dicke 0.99 mm und Poissonzahl 0.2. Schwarze Linien durch die Platte stellen die Symmetrieebenen dar.

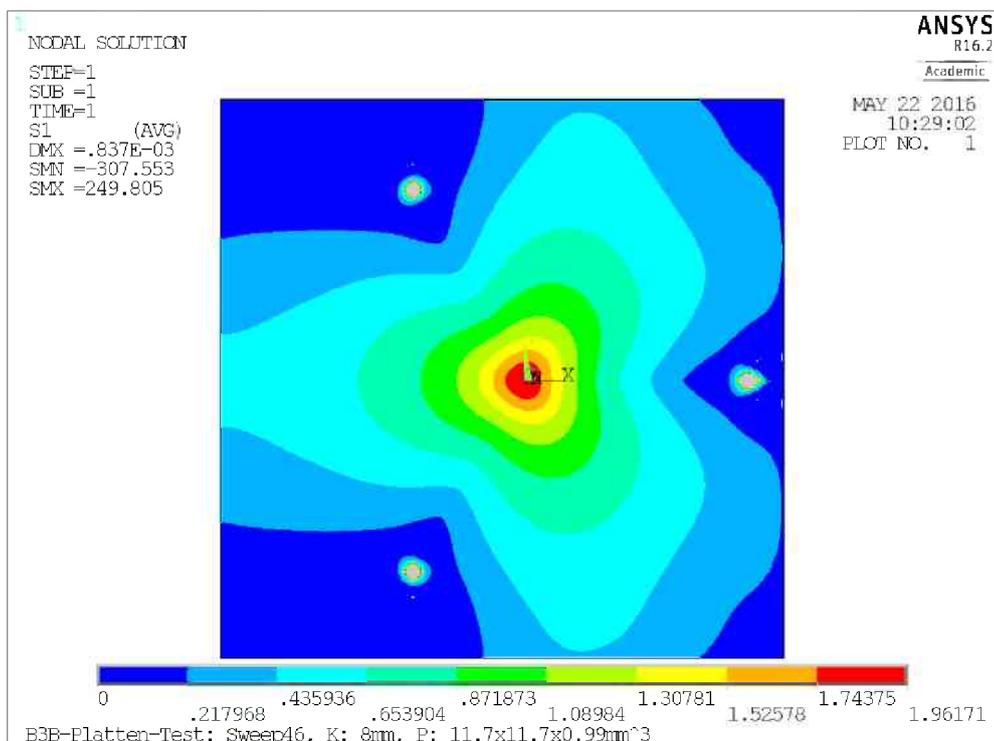


Abbildung 4.8.: Spannungsverteilung einer Platte mit Länge 11.7 mm, Breite 11.7 mm, Dicke 0.99 mm und Poissonzahl 0.2. Die Lastenbringung ist hier wie in Abb. 4.6b zentrisch.

5. Definition und Parametrisierung einer empirischen Auswerteformel für den B3B–Versuch

5.1. Formel zur Abschätzung der Maximalspannung unter der Lastkugel

5.1.1. Scheiben

Die Maximalspannung unter der Lastkugel des B3B–Versuchs lässt sich vereinfacht darstellen durch

$$\sigma_m = \frac{F}{t^2} \cdot f \quad (5.1)$$

wobei f eine dimensionslose Funktion der Geometrie und Materialeigenschaften der Probe ist [5] (F ist die Druckkraft und t die Probendicke).

Die geometrischen Einflüsse auf f können durch Normierung der Größen nicht nur dimensionslos gemacht werden, es wird auch die Anzahl der Variablen der Funktion um eine reduziert. In vorhergehenden Arbeiten [5, 6, 31, 32, 33] wurden diese normierten Größen immer auf den Radius R der scheibenförmigen Proben bezogen, der für jede Versuchsfolge recht gleichmäßig war. Die Geometrie von rundgeschliffenen Proben schwankt hier jedoch stärker und macht so Werte von z.B. $\frac{t}{R}$ untereinander nicht vergleichbar. In dieser Arbeit wird daher vorgeschlagen eine immer konstante Normierungsgröße zu wählen, und zwar den Auflagerradius R_a der durch die in Kontakt zueinander stehenden Lagerkugeln (mit Radius R_K) immer gleich groß ist:

$$R_a = \frac{2R_K}{\sqrt{3}} \quad (5.2)$$

Für die Funktion f von Scheiben gibt es folgende dimensionslose Variablen:

$$\tau = \frac{t}{R_a} \quad (5.3)$$

$$\rho = \frac{R}{R_a} \quad (5.4)$$

und die Querkontraktionszahl ν , die von Natur aus keine Einheit hat.

Aufgrund der bisherigen relativ festen Verhältnisse zwischen R und R_a können die bekannten Ansätze für die Funktion f übernommen werden.

Börger *et al.* [5] stellten für f folgende empirische Form auf:

$$f\left(\frac{t}{R}, \frac{R_a}{R}, \nu\right) = c_0 + \frac{c_1 + c_2 \frac{t}{R} + c_3 \left(\frac{t}{R}\right)^2 + c_4 \left(\frac{t}{R}\right)^3}{1 + c_5 \frac{t}{R}} \left(1 + c_6 \frac{R_a}{R}\right) \quad (5.5)$$

Diese wurde für verschiedene ν errechnet, für Zwischenwerte wurde dann zwischen den einzelnen f linear interpoliert.

Für die hier gegebenen Werkstoffe ist die Schwankung von ν klein genug um einen linearen Term dafür direkt in die Fitformel einzubauen. Abb. 5.1 bestätigt, dass die FEM-Werte an einer Gerade liegen.

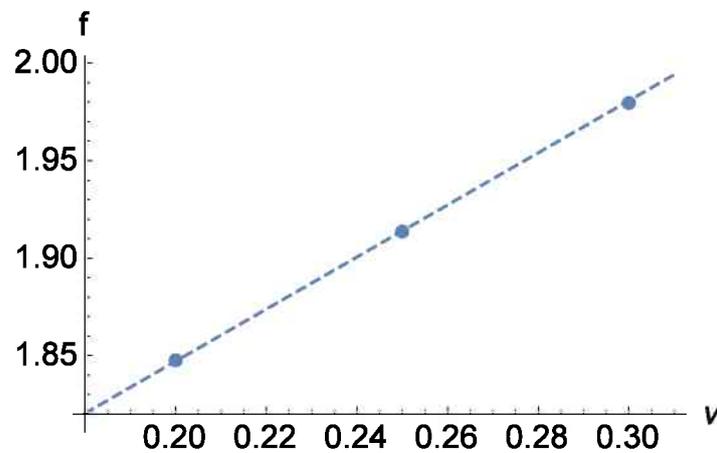


Abbildung 5.1.: Vergleich von drei beispielhaften Werten für f einer Platte mit Dicke 1.13 mm, Radius 6.15 mm. Eine lineare Abhängigkeit besteht mit ausreichender Genauigkeit.

Des weiteren wurde die Abhängigkeit von ρ weiter untersucht, da der Plattenüberhang bei den rechteckigen Proben beträchtlich werden konnte. Es wurde festgestellt, dass sich bei gleichem Auflagerradius für größere Scheiben f auf einen Plateau-Wert abfällt mit der Abhängigkeit:

$$f \propto c_0 + \frac{c_1}{\rho^2} \quad (5.6)$$

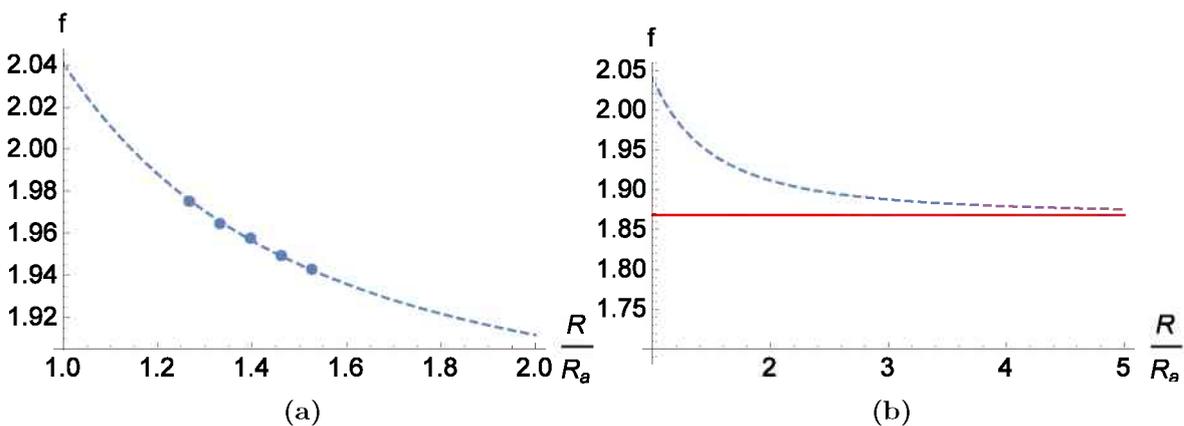


Abbildung 5.2.: Abfall von f mit wachsendem Scheibenradius. Die fünf Punkte in (a) entsprechen Proben mit einer Dicke von 1 mm und einer Poissonzahl von 0.25. Die Variation von R entspricht dem geometrischen Extremwerten aus Tabelle 4.2. (b) zeigt den Verlauf der Formel einen weiteren Bereich und in Rot den Limes für $\frac{R}{R_a} \rightarrow \infty$.

Abb. 5.2 zeigt, dass der Einfluss in den Größenordnungen der untersuchten Platten klein ist. Die Werte nähern sich sehr gut an die Kurve an und die Auswirkung einer sogar unendlich großen Platte wäre nur wenige Prozente.

Durch diese Abänderungen sieht der Ansatz für f folgendermaßen aus:

$$f(\tau, \rho, \nu) = c_0 + \frac{c_1 + c_2\tau + c_3\tau^2 + c_4\tau^3}{1 + c_5\tau} \cdot (1 + c_6\nu) \cdot \left(1 + \frac{c_7}{\rho^2}\right) \quad (5.7)$$

An die Ergebnisse der FEM-Rechnung gefittet — wo c_0 bis c_7 die Werten aus Tabelle 5.1 annehmen — gibt diese Form von f eine maximale Abweichung von unter 0.2% im untersuchten Parameterbereich (siehe Tab. 4.2). Die Beschreibung von Scheiben mit dieser Fitformel für f ist also ausreichend genau für die Festigkeitsprüfung.

Tabelle 5.1.: Die Konstanten der Gl. (5.7) angeglichen an die Ergebnisse der FEM-Rechnung. Die Zahlenwerte werden hier auf sechs signifikante Stellen gegeben.

c_0	−0.095 836 9
c_1	60.7238
c_2	1844.78
c_3	−1474.12
c_4	587.498
c_5	1104.47
c_6	0.797 097
c_7	0.090 529 5

5.1.2. Platten

Platten sind im Ausmaß ihrer Variablen um einiges komplizierter. Man braucht nicht nur zwei Maße statt einem um die Grundfläche zu beschreiben, die Platten werden in vielen Fällen außerhalb ihres Mittelpunkts belastet und weisen eine Orientierung zum Dreieck der Auflagerkugeln auf. Abb. 4.6 zeigt die Möglichkeiten, eine Platte sinnvoll in Relation zu den Auflagerkugeln zu orientieren.

Diese große Anzahl an Einflüssen auf f kann grundsätzlich in eine neue Fitformel miteinbezogen werden. Dafür sind aber um mehrere Größenordnungen mehr Berechnungen notwendig. Auch die Anwendbarkeit der Formel würde unter den vielen nötigen Konstanten und den zu messenden und aufzuzeichnenden Werten leiden.

Es wurde daher versucht, eine Verbindung zwischen den vielen möglichen Plattengeometrien und den einfacher zu beschreibenden Scheiben durch einen sogenannten Ersatzradius \tilde{R} zu finden.

Die Bedingung an diesen Radius ist es, eine möglichst kleine Abweichung von f für Platten zu ergeben, die mit der Formel für Scheiben behandelt wurden. Er beschreibt also eine Scheibe, die den selben Faktor f hat als die Platte, deren Geometrie er ersetzt.

Mehrere Ansätze zur Definition von \tilde{R} wurden in Erwägung gezogen:

1. Radius eines flächengleichen Kreises,
2. Kleinste Distanz zu einer Plattenkante (Abb. 5.3 stellt verschiedene Arten der Distanzmessung dar),
3. Mittlere Kantenlänge als Durchmesser eines Kreises,

4. Mittlere Distanz zu allen Kanten (wieder nach Abb. 5.3 definiert).

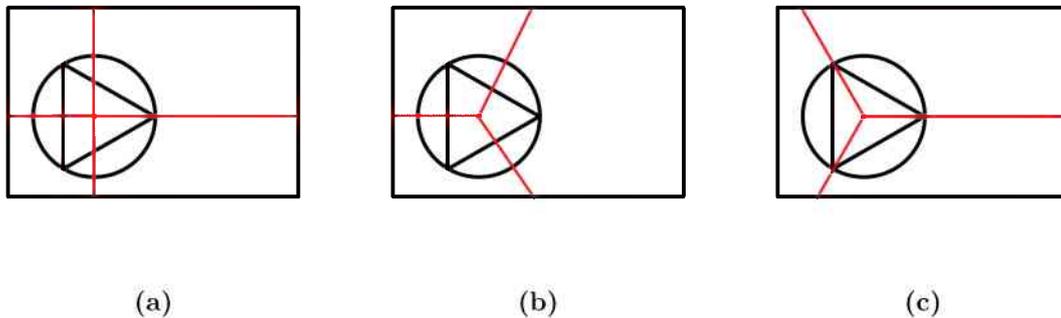


Abbildung 5.3.: Verschiedene Möglichkeiten eine Kantendistanz zu definieren: Die roten Linien stellen hier die gemessenen Distanzen dar. **(a)** stellt die Messung des Normalabstands zum Plattenrand dar, **(b)** misst von der Lastkugel durch die Seitenmitten des Auflagerdreiecks und **(c)** misst analog durch die Auflagerpunkte zur nächsten Kante.

Platten in allen Orientierungen und Größen wurden im FEM-Programm simuliert und deren Ersatzradien mit den verschiedenen Methoden berechnet. Mit diesen Ersatzradien wurden dann die errechneten Werte für f nach (5.1) mit der empirischen Formel (5.7) verglichen und untersucht, welche Ansätze die kleinste Abweichung liefern. Verändert wurde je nach Ansatz die Abszissen für die entsprechenden berechneten f -Werte. So ergab jede Version des Ersatzradius eine andere Punktwolke, die sich je nach Eignung verschieden von der Kurve der Formel (5.7) abhebt. Abb. 5.4 zeigt wie sich die FEM Werte für diese Ersatzradien entlang der Kurve der Fitformel anordnen.

Die geringsten Unterschiede lieferten die kleinsten Kantendistanzen nach Abb. 5.3a und 5.3b.

Anzuwenden sind sind beide Ersatzradien gleich: Je nach Konstruktion des Auflagers kann eine Platte entweder zentrisch oder exzentrisch belastet werden.

Wenn die Platten mittig belastet werden, gibt es aufgrund der Symmetrie nur zwei Werte zu vergleichen, diese ändern sich aber mit neuen Plattengeometrien ständig. Bei exzentrischer Last wird die Platte mit einer Ecke auf Anschlag gebracht, es gibt also zwei fixe Distanzen, die dann von freien Kanten nur bei sehr kleinen Proben unterschritten werden.

Abb. 5.4c zeigt noch, dass durch Mitteln der beiden Ersatzradien (Normalabstände und Distanzen durch die Auflagerseiten) ein weitaus besseres Anschmiegen an die Kurve erreicht wird. In einigen Fällen überschätzt ein Ansatz, was der andere unterschätzt. Der Durchschnitt der beiden gibt dann einen genaueren Ersatzradius.

Alle diese Berechnungen können einfach in einem Tabellen-Programm wie *Excel* oder *Mathematica* automatisiert werden und müssen für Prüfer keine zusätzlichen Aufwand bedeuten.

5.1.3. Schaurechnung

Um den Berechnungsablauf des Ersatzradius zu veranschaulichen, wird im Folgenden eine beispielhafte Platte berechnet. Tabelle 5.2 gibt die hierfür benötigten Abmessun-

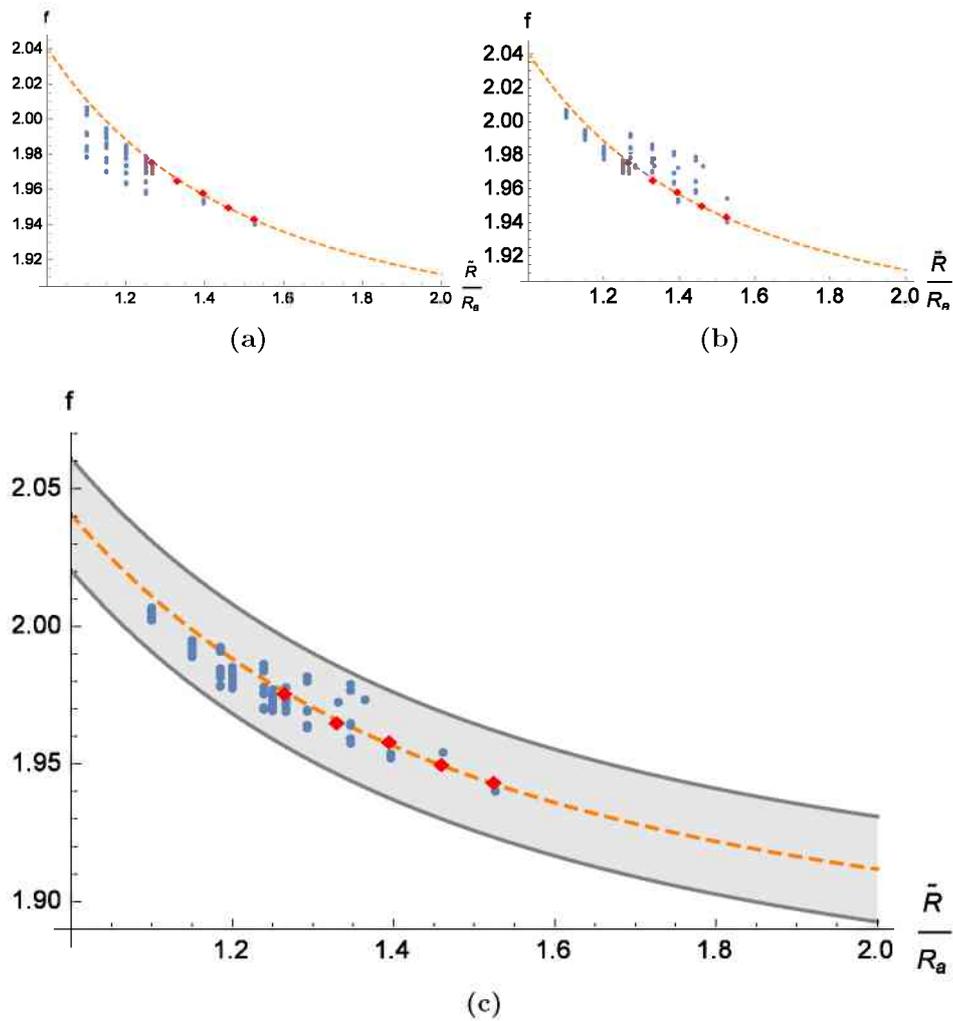


Abbildung 5.4.: \tilde{R} als die kleinste Kantendistanz, definiert durch **(a)** Abb. 5.3a und **(b)** Abb. 5.3b. Die gestrichelte Linie ist wieder der Zusammenhang $f \propto \rho^{-2}$ für Scheiben aus (5.6). Wie in Abb. 5.2 ist $t = 1$ mm und $\nu = 0.25$. Die maximale Abweichung vom Faktor f beträgt für **(a)** 1.62% und für **(b)** 1.41%. **(c)** zeigt den Durchschnitt aus **(a)** und **(b)**. Der graue Balken bezeichnet einen Fehler von 1%. Die größte Abweichung von der Formel beträgt hier nur 0.78%.

gen wieder, inklusive der Versetzungen des Auflagers aus Abschnitt 4.1. Es handelt sich hier um eine Orientierung nach Abb. 4.6d.

Tabelle 5.2.

Länge der Platte	12.12	mm
Breite der Platte	11.93	mm
Abstand Breitenkante	6.309	mm
Abstand Längskante	5.815	mm

Händisches Errechnen der Werte

Zuerst werden die Normalabstände der Krafteinleitung zu den Kanten errechnet. Zwei davon sind bereits durch das Auflager vorgegeben, nämlich 6.309 mm und 5.815 mm. Die beiden anderen Abstände erhält man durch die Differenzen aus Länge bzw. Breite zu den fixen Abständen.

$$12.12 - 6.309 = 5.811$$

$$11.93 - 5.815 = 6.115$$

Der kleinste dieser Werte ist 5.811 mm und stellt den Normalkantenabstand nach Abb. 5.4a dar.

Die Methode der Distanzen durch die Auflagerdreiecksseiten bedient sich der Zahlen auf der vorherigen Rechnung um die schrägen Geraden aus Abb. 5.4b zu errechnen.

Die erste Gerade ist parallel zur Längskante und im Falle dieses Auflagers bereits eine der fixen Abstände, nämlich 6.309 mm.

Die beiden anderen Linien gehen je 120° von dieser aus. Dabei gibt es zwei Möglichkeiten für deren Länge: Distanz zur Längskante oder zur Breitenkante (siehe Abb. 5.5). Je nach Größe der Platte sind beide Längen möglich. Bei Bearbeitung in Tabellenform ist es daher nötig, beide zu berechnen, wobei das kleinere der Ergebnisse immer die nähere Kante bezeichnet.

Durch Winkelfunktionen lassen sich so für beide Distanzen die Werte finden:

$$\left. \begin{array}{l} \frac{5.815}{\cos(30^\circ)} = 6.715 \\ \frac{5.811}{\cos(60^\circ)} = 11.62 \end{array} \right\} 6.715$$

$$\left. \begin{array}{l} \frac{6.115}{\cos(30^\circ)} = 7.061 \\ \frac{5.811}{\cos(60^\circ)} = 11.62 \end{array} \right\} 7.061$$

Das Minimum dieser Distanzfindung ist also 6.309 mm.

Wenn der Durchschnitt beider Methoden als Ersatzradius gewählt wird, muss dieser letztendlich noch berechnet werden:

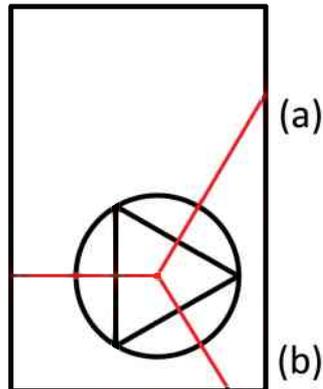


Abbildung 5.5.: Schräge Distanzen können sowohl an der vertikalen **(a)** als auch an der horizontalen **(b)** Kante enden. Daher werden die Abstände zu beiden (bzw. deren virtuellen Verlängerungen) berechnet. Der kleinere Wert ist immer der gewünschte.

$$\frac{5.811 + 6.309}{2} = 6.060$$

Der Ersatzradius für die Platte in diesem Beispiel ist also 6.06 mm. In die Funktion $f(\tau, \nu, \rho)$ ist der dieser als normierter Ersatzradius $\tilde{\rho}$ einzusetzen.

$$\tilde{\rho} = \frac{\tilde{R}}{R_a} \quad (5.8)$$

Die Maximalspannung in der Platte kann dann mit f aus (5.7) ausgerechnet werden, als ob es sich um eine Scheibe handeln würde mit dem Zusammenhang:

$$\sigma_m = \frac{F}{t^2} \cdot f(\tau, \nu, \tilde{\rho}) \quad (5.9)$$

Automatisierte Rechnung

All die vorhergegangenen Schritte werden in Praxis nicht einzeln für jede Probe durchgeführt sondern automatisiert in Tabellen durchgeführt.

Der Nachfolgende Code ist ein Beispiel für eine Implementierung in *Mathematica*:

Es werden zuerst die Abstände des Berührungspunktes der Lagerkugeln zum Anschlag angegeben:

```
DX = 6.309
DY = 5.815
```

Dann wird eine Liste mit den Abmessungen der Platten entweder eingegeben oder eingelesen:

```
platten = {{12.12, 11.93}, {11.96, 11.72}, {12.01, 11.81}, ...}
```

Auf diese Liste wird dann eine Funktion angewandt, die Für jede Platte vier Geradengleichungen für die vier Normalabstände aufstellt und daraus das Minimum nimmt:

```
R1 = Min[ DX, DY, #[[1]] - DX, #[[2]] - DY ]& /@ platten
```

Eine ähnliche Funktion wird für die Distanzen durch die schrägen Distanzen errechnet. Hier werden die Schrägen sowohl mit den horizontalen als auch mit der vertikalen Kante gekreuzt, um beide Fälle aus Abb. 5.5 zu behandeln.

```
R2 = Min[ DX, DY / Cos[30], (#[[2]] - DY) / Cos[30], (#[[1]] - DX) / Cos[60] ] & /@ platten
```

Zuletzt wird nun aus den beiden Ergebnislisten eine Liste mit dem Durchschnitt der beiden Methoden erstellt.

```
R3 = (R1 + R2) / 2
```

R3 enthält nun alle Ersatzradien für die in der Liste `platten` enthaltenen Maße.

```
R3 = {6.06, 5.98, 6.005, ...}
```

5.1.4. Vernachlässigung des Überhangs

Abb. 5.2 zeigt einen recht kleinen Unterschied von nur 10% für f zwischen Scheiben, die direkt an der Kante gelagert werden und unendlichen Scheiben. Im vollen untersuchten Parameterraum von 11.7 mm bis 14.1 mm Abmessung liegt die Differenz schon unter 4%.

Wenn man die Abmessungen der Proben in einem kleinen Bereich hält, wie bei den Proben aus Tabelle 4.1, wird der Einfluss des Überhängenden Materials vernachlässigbar. So lässt sich eine Fitformel definieren, bei welcher der Scheibenradius bzw. der Ersatzradius völlig wegfallen:

$$f(\tau, \rho, \nu) = c_0 + \frac{c_1 + c_2\tau + c_3\tau^2 + c_4\tau^3}{1 + c_5\tau} \cdot (1 + c_6\nu) \quad (5.10)$$

Einen solchen kleinen Geometriebereich gibt das Auflager für Platten aus Abb. 4.2 vor. Für Platten mit Länge zwischen 11.9 mm und 12.7 mm und Breite zwischen 11.6 mm und 12.1 mm beträgt der Fehler für f mit Konstanten aus Tabelle 5.3 hier $< 1\%$. Hierfür wurde eine Formel erneut an diesen kleineren Bereich gefittet, es ist aber auch möglich eine Applikation zu programmieren, die für gegebene Geometrien Gl. (5.10) aus Gl. (5.7) errechnet.

Durch die Anwendung dieser vereinfachten Formel entfällt der Bedarf für die Berechnung eines Ersatzradius. Allerdings muss darauf geachtet werden, dass die Proben im definierten Größenbereich bleiben, da sonst der Überhang einen merklichen Einfluss nehmen kann.

5.2. Interpolation der effektiven Größen

Für die effektiven Größen wurde für jede berechnete Probenkonfiguration eine Tabelle mit je vier Werten ($V_{eff}^{\sigma_1}$, V_{eff}^{PIA} , $S_{eff}^{\sigma_1}$ und S_{eff}^{PIA}) für jeden vollen Wert des Weibull Moduls bis hin zu m_{max} ausgegeben.

Mithilfe der *Interpolating Function* von *Mathematica* wurde eine Funktion definiert, die für alle möglichen Probengeometrien und Werkstoffe eine Tabelle effektiver Größen interpoliert.

Der Zusammenhang von V_{eff} über m ist nicht direkt zu interpolieren, da er eine stark exponentiell abfallende Funktion ist (siehe Abb. 5.6). Daher wurde zum Interpolieren

Tabelle 5.3.: Die Konstanten der Gl. (5.10) angeglichen an die Ergebnisse der FEM-Rechnung. Die Zahlenwerte werden hier auf sechs signifikante Stellen gegeben.

c_0	0.323 308
c_1	2.334 61
c_2	-4.126 11
c_3	8.739 27
c_4	-6.052 85
c_5	1.719 55
c_6	1.102 85

einer konkreten effektiven Größe für einen bestimmten Weibull Modul der Verlauf doppelt logarithmisch aufgetragen und linearisiert. Der Exponent dieser linearen Funktion modelliert den Verlauf der effektiven Größen sehr gut und ohne große Schwankungen zwischen den Stützpunkten.

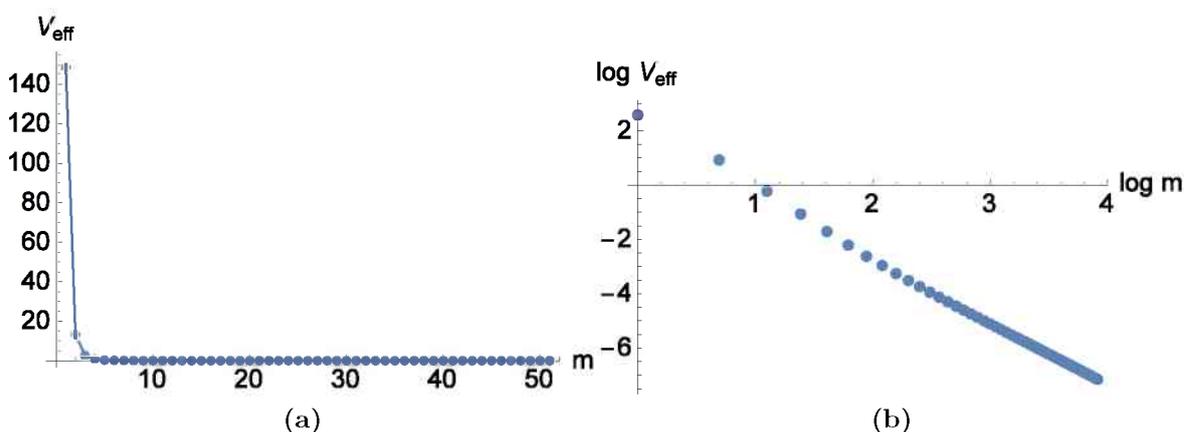


Abbildung 5.6.: Die effektiven Größen fallen durch die Potenzierung mit m sehr schnell gegen kleine Werte ab. **(a)** zeigt einen beispielhaften Verlauf von V_{eff} bis zu einem m_{max} von 50. Die effektiven Volumina können leicht fünf Größenordnungen überspannen, was eine Annäherung mit Polynomen sehr ungenau macht. **(b)** zeigt den selben Verlauf auf logarithmischen Axen. Hier lässt sich eine Gerade leicht anpassen.

Aufgrund der großen Menge an Ausgabeinformationen dieser gefitteten Funktionen konnten diese Funktionen nicht einfach dargestellt werden. Bei $m_{max} = 50$ gibt es 204 Werte in der Tabelle der Interpolation, was im Endeffekt 204 einzelne Polynome bedeutet. Für einen Verlauf einer effektiven Größe einer bestimmten Geometrie müssen 51 dieser Polynome ausgewertet und die Ergebnisse wieder logarithmiert und interpoliert werden.

Es musste daher eine für die Versuchsreihe eigene Tabellendatei mit den Interpolationsformeln angelegt werden, die jedes mal die vollen Stützpunktinformationen lädt.

5.3. Fehlereinflüsse — Gültigkeitsbereiche für elastische Platten

Börger *et al.* [6] haben die Auswirkungen von möglichen geometrischen und tribologischen Fehlern auf Scheiben schon weitgehend untersucht. Ihre Schlussfolgerung war, dass der 4-Kugelversuch gegenüber Fehlern in der Probenausrichtung und Messung des Probenradius sehr tolerant ist. Wichtige Einflüsse auf die Genauigkeit der Ergebnisse haben die Bestimmung der Probendicke und der Poissonzahl.

Nicht untersucht wurde der Einfluss eines niedrigen E-Moduls oder einer besonders hohen Last F auf die Gültigkeit der gefitteten Funktion f , da im untersuchten Parameterbereich keine Auswirkung gezeigt wurde [5].

Ref. [33] beschreibt die Auswirkung von hohen Kräften bzw. niedrigen E-Modul durch eine normierte Festigkeit $\frac{\sigma}{E}$. Mit dieser kann ein Bereich für definierte Fehler von f auf, der von der Probendicke abhängt. Je fester eine Probe also ist (also einer höheren Bruchkraft bedarf) bzw. je elastischer sie ist, desto mehr wird f überschätzt. Hierbei gibt es zwei zu beachtende Effekte. Sehr dünne Proben biegen sich unter Last mehr durch und können durch ihre Bewegung die Ausgangsgeometrie verändern. Dicke Proben brauchen so große Kräfte, um zu versagen, dass die Kontaktspannungen zu Problemen führen können.

Diese Überlegungen aus [33] wurden hier noch einmal aufgegriffen und ohne jene Vereinfachungen und Annahmen nachvollzogen, die nur für den im Buch behandelten Geometrien und Materialwerte gültig sind.

5.3.1. Auswirkung der Durchbiegung

Wenn sich eine Probe aufgrund der Belastung in der Probenmitte verformt, wölbt sie sich in der Mitte und zeigt gegen den Rand in einen Steigungswinkel θ . Diese Steigung bedeutet eine Änderung der Auflagerpunkte, da diese sich an den Lagerkugeln zur Lastkugel hin abrollen. Es verringert sich also der Auflagerradius R_a . Die kürzere Distanz zwischen Last und Lager führt zu einem verringerten Moment, das auf die Probe wirkt und so zu einer niedrigeren tatsächlichen Spannung. Die Festigkeit der Keramik wird also überschätzt.

Untersucht wurde diese Abweichung vorwiegend analytisch, basierend auf einer grundlegenden Annahme: Die Belastungssituation des 4-Kugelversuches kann ausreichend genau durch einen Doppelring-Versuch mit gegen 0 gehendem Belastungsradius beschreiben werden.

Eine Formel für den Winkel der Durchbiegung θ im Abstand r von der Krafteinleitung findet sich in Ref. [34, S. 432]:

$$\theta(r) = \frac{Fr}{4\pi D} \cdot \left(\frac{1}{1+\nu} + \ln \frac{R_a}{r} \right) \quad (5.11)$$

wobei

$$D = \frac{Et^3}{12(1-\nu^2)} \quad (5.12)$$

Der maximale Winkel ist derjenige im Auflagerpunkt ($r = R_a$), die Formel vereinfacht sich also zu:

$$\theta(R_a) = \frac{FR_a}{4\pi D} \cdot \frac{1}{1+\nu} \quad (5.13)$$

$$\theta(R_a) = \frac{FR_a}{Et^3} \cdot \frac{3(1-\nu^2)}{\pi(1+\nu)} \quad (5.14)$$

Man kann die Kugellast durch die Maximalspannung ausdrücken, da diese ja über den Faktor f und die Probendicke zusammenhängen.

$$F = \frac{\sigma_m t^2}{f} \quad (5.15)$$

Weiters kann man bei kleinen Winkeln vereinfachen:

$$\sin(\theta) = \frac{x}{R_K} \approx \theta \quad (5.16)$$

Hier ist R_K der Radius der Auflagerkugel und x die Verschiebung des Auflagerpunktes in Richtung des Scheibenmittelpunkts. Setzt man diese Annahmen ein, erhält man:

$$\frac{x}{R_K} = \frac{3(1-\nu)}{\pi} \cdot \frac{\sigma_m R_a}{fEt} \quad (5.17)$$

Formt man dann um auf eine normierte Festigkeit $\frac{\sigma}{E}$ bekommt man:

$$\frac{\sigma}{E} = \frac{\pi \cdot f}{3 \cdot (1-\nu)} \cdot \frac{x}{R_K} \cdot \frac{t}{R_a} \quad (5.18)$$

Zuletzt muss man noch den fixen Zusammenhang zwischen Kugel- und Auflagerradius beachten, der durch die Berührung der Kugeln festgelegt ist:

$$R_a = \frac{2}{\sqrt{3}} R_K \quad (5.19)$$

$$\frac{\sigma}{E} = \frac{2\pi \cdot f}{3^{3/2} \cdot (1-\nu)} \cdot x_{crit} \cdot \frac{t}{R_a^2} \quad (5.20)$$

Hierbei bezeichnet x_{crit} die Verschiebung des Auflagers, die nötig ist, um eine kritische Ungenauigkeit der Maximalspannung zu erhalten.

Diese kritische Verminderung wurde durch Variation des Auflagerradius in der analytischen Doppelringformel untersucht. Abb. 5.7 zeigt wie die biaxiale Spannung im Zentrum des Doppelring-Versuchs sich mit dem Auflagerradius ändert. Abb. 5.8 zeigt die Abhängigkeit der Spannungsänderung und der Auflagerverschiebung. Aus der Linearisierung dieses Zusammenhangs konnte x_{crit} für große R_a (nahe R) definiert werden als:

$$x_{crit} = R_a \cdot k_x \cdot \Delta\sigma \quad (5.21)$$

Hier beschreibt k_x die Steigung der Geraden in Abb. 5.8 und $\Delta\sigma$ die kritische Abweichung der Spannung in Prozent. Ein konkreter Fall wäre also:

$$x_{2\%} = R_a \cdot k_x \cdot 0.02 \quad (5.22)$$

Setzt man x_{crit} in (5.18) ein, lässt diese sich auf die normierte Dicke τ beziehen:

$$\frac{\sigma}{E} = \frac{2\pi \cdot f}{3^{3/2} \cdot (1 - \nu)} \cdot R_a \cdot k_x \cdot \Delta\sigma \cdot \tau \quad (5.23)$$

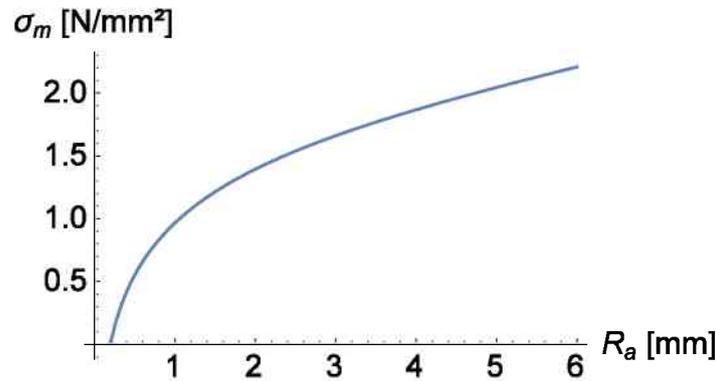


Abbildung 5.7.: Zunahme von σ_m eines Doppelring-Versuchs mit wachsendem R_a . Die Parameter dieser Rechnung waren arbiträrische Durchschnittswerte der Zahnkeramikproben: $t = 1$ mm, $\nu = 0.25$. Die Kraft war 100 N, aber sie skaliert linear und wird bei Vergleichen untereinander gekürzt. Der E-Modul kommt in der analytischen Formel für σ_m nicht vor.

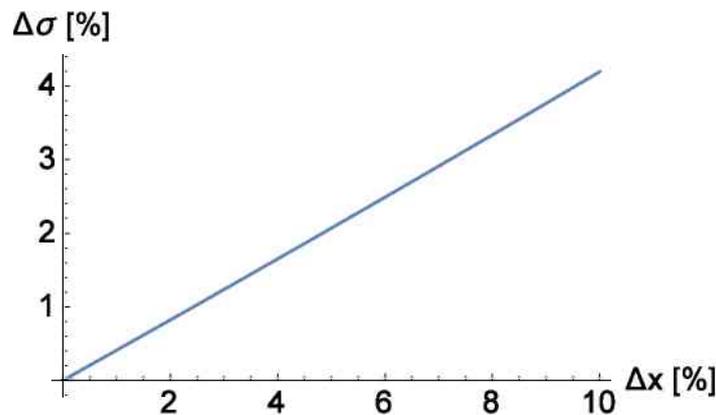


Abbildung 5.8.: Ungenauigkeit der Spannung in Prozent durch Verkleinern des Auflagerradius. Die selben Parameter aus Abb. 5.7 gelten, zusätzlich ist der Ausgangsradius als 6 mm gewählt.

5.3.2. Auswirkung des Kontakts

Nicht nur das Abrollen an Auflagerkugeln kann zu einer Verkleinerung des Moments an der Probe führen. Aufgrund der Größe der Kugeln im Versuch kann bei elastischen Werkstoffen der Hertz'sche Kontaktbereich dermaßen anwachsen, dass sich das Moment verringert.

Der Kontaktbereich ergibt sich aus der Last, die auf die Kugel wirkt, deren Radius und den kombinierten elastischen Eigenschaften von Kugel und Probe [35]:

$$a^3 = \frac{3FR_K}{4E^*} \quad (5.24)$$

$$E^* = \left(\frac{1 - \nu^2}{E} + \frac{1 - \nu_K^2}{E_K} \right)^{-1} \quad (5.25)$$

Wieder werden die Kraft durch die Spannung und der Kugelradius durch den des Auflagers beschrieben:

$$a^3 = \frac{3^{3/2} \sigma_m t^2 R_a}{8 f E^*} \quad (5.26)$$

E^* muss noch in Verhältnis zu E gebracht werden um einen Vergleich mit den Werten der Durchbiegung zu ermöglichen:

$$a^3 = \frac{3^{3/2} \sigma_m t^2 R_a}{8 f \frac{E^*}{E} E} \quad (5.27)$$

Schlussendlich wird auch hier auf die normierte Festigkeit $\frac{\sigma}{E}$ umgeformt und man erhält:

$$\frac{\sigma}{E} = \frac{8 \cdot f}{3^{3/2} \cdot (1 - \nu)} \cdot \frac{E}{E^*} \cdot a_{crit}^3 \cdot \frac{1}{t^2 R_a} \quad (5.28)$$

Leider kann der Hertz'sche Kontakt nicht analytisch in die Doppelringformel eingebunden werden, da diese für zu kleine Innenradien eine Singularität bildet. Es wurde daher auf FEM-Berechnungen zurückgegriffen, die auch gleichzeitig die Gültigkeit der Annahmen für die Variation von R_a bestätigen konnten.

Unter sonst konstanten Bedingungen wurde nur der Radius einer Hertz'schen Spannung (die ellipsenförmig zum Rand hin abfällt) geändert, bis hin zu 25% von R_a .

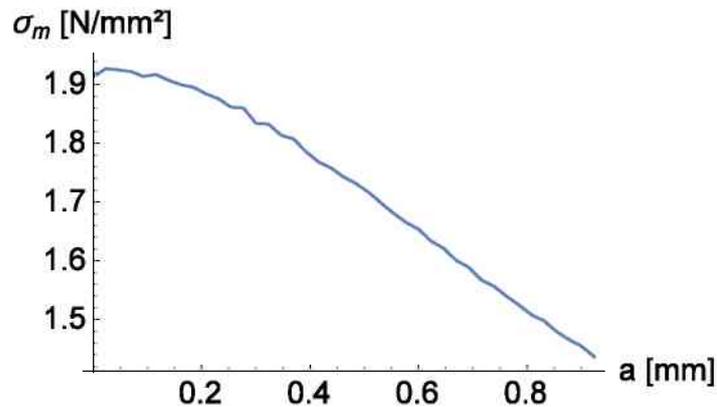


Abbildung 5.9.: Abnahme von σ_m einer B3B-Scheibe mit wachsendem a . Die selben Parameter aus Abb. 5.7 gelten, wieder ist der Auflagerradius als 6 mm gewählt. Der E-Modul kommt in der analytischen Formel für σ_m nicht vor. Die Kraft ist hier 1 N, und der E-Modul hat aufgrund der idealisierten Umstände der FEM-Rechnung keinen Einfluss.

Abb. 5.9 und 5.10 zeigen den Einfluss eines größer werdenden Kontakts. Hier ist kein linearer Zusammenhang gegeben, sondern eine Parabel.

$$a_{crit} = R_a \cdot k_a \cdot \sqrt{\Delta\sigma} \quad (5.29)$$

stellt eine gute Näherung bis ca. 10% von R_a dar.

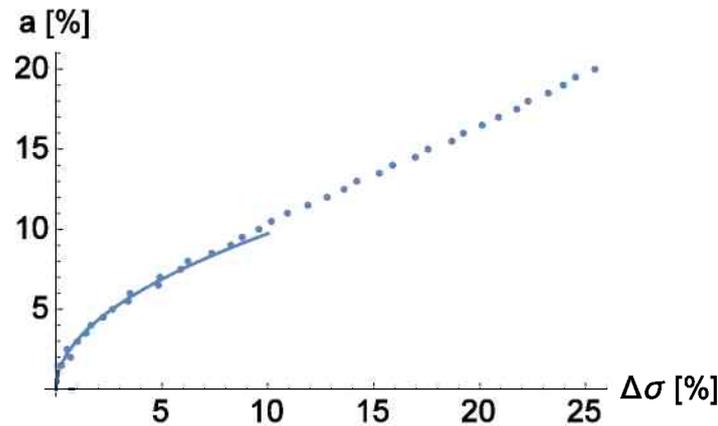


Abbildung 5.10.: Ungenauigkeit der Spannung in Prozent durch Wachsen des Hertz'schen Kontaktradius a . Die Punkte sind die berechneten Kontaktradien, die durchgehende Linie eine Parabel, die das Verhalten bis 10% Abweichung gut darstellt.

Setzt man dieses in (5.28) ein, wird $\frac{\sigma}{E}$ wieder von τ abhängig:

$$\frac{\sigma}{E} = \frac{8 \cdot f}{3^{3/2} \cdot (1 - \nu)} \cdot \frac{E}{E^*} \cdot (k_a \sqrt{\Delta\sigma})^3 \cdot \frac{1}{\tau^2} \quad (5.30)$$

5.3.3. Vereinen der beiden Einflüsse

Es gibt also zwei Kurven (5.23) und (5.30), die Grenzen für Genauigkeiten darstellen, eine proportional zu τ , eine zu τ^{-2} . Durch Einbeziehen der Funktion f wurde aus der Geraden eine Parabel, da f für dünne Scheiben stark zunimmt. Die Hyperbel wird durch diesen Bezug nicht so sichtlich verzerrt. Diese Kurven schließen eine Fläche ein, unter der eine gewählte Genauigkeit garantiert werden kann.

Abb. 5.11 zeigt verschiedene Genauigkeitsgrade, die auf eine Probenmenge gelten können und zusätzlich die Punkte der geprüften Keramiksorten aus Tabellen 1.1 und 4.1.

5.4. Schlussfolgerungen

Der Bereich in dem die Proben auf eine bestimmte Genauigkeit geprüft werden können (vgl. Abb. 5.11) zeigt ein klares Maximum im Dickenbereich $0.08 - 0.11 R_a$. Dieses Erkenntnis steht im Konflikt zu den vorhergehenden Berechnungen von Scheiben im 4-Kugelversuch [33]. Hier befand sich das Maximum bei $0.3 R$ (ca. $1.25 - 1.4 R_a$), was auch die enge Streuung um diesen Bereich bei den geprüften Proben verständlich macht.

Der Unterschied zwischen den idealen Dicken kommt durch verschiedene berechnete x_{crit} und a_{crit} zustande. Diese kritischen Verschiebungen der Lagerpunkte müssen noch genauer untersucht werden, da der 4-Kugelversuch durch seine sehr großen Kugeln sehr viel empfindlicher auf winkelabhängige Veränderungen ist.

Für die Konstruktion des Auflagers ist die ideale Probendicke jedoch von wenig Belang, da die Kugeln immer Kontakt haben.

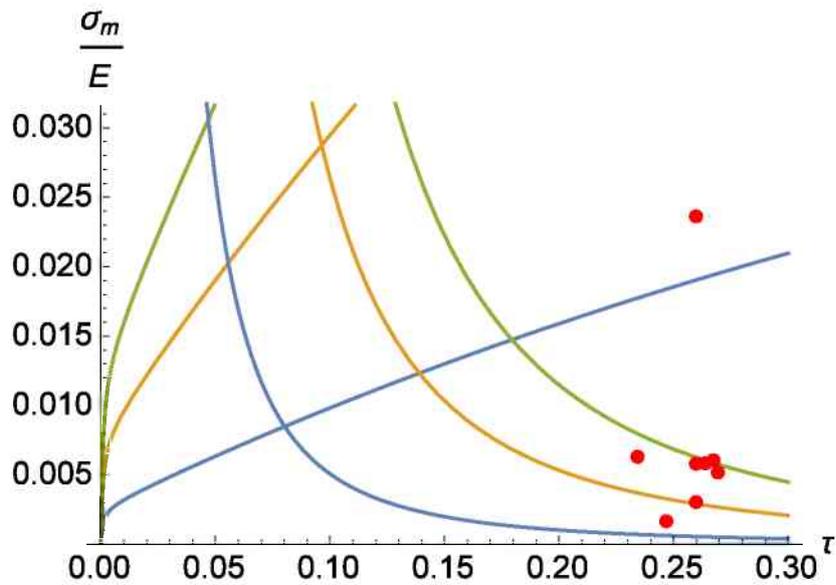


Abbildung 5.11.: Einfluss der normierten Festigkeit auf die Genauigkeit der Spannungsergebnisse durch die Fitformel. Die jeweils unterste Parabel und Hyperbel bezeichnen die Grenze von 1% Genauigkeit, die zweiten 3% und die obersten beiden 5%. Die Punkte sind geprüfte Scheiben aus Dentalkeramik [2], wobei der oberste Ausschlag einen E-Modul von 12 GPa besitzt und schon zuvor als ungeeignet eingestuft wurde.

Durch das Einführen eines Ersatzradius können Platten leicht als äquivalente Scheiben mit Gl. (5.7) mit einem akzeptablen Schätzfehler ausgewertet werden. Dadurch müssen Platten auch nicht zwangsläufig mittig belastet werden, sondern können an einer Ecke auf Anschlag liegen, was die Konstruktion des Auflagers und die genaue Bestimmung des Ersatzradius erheblich einfacher gestaltet.

6. Zusammenfassung und Ausblick

Ziel dieser Arbeit war, es den 4-Kugelversuch für das Prüfen von Dentalkeramiken anzupassen. Dentalkeramikproben können aus sogenannten CAD/CAM-Blöcken geschnitten werden. Diese Blöcke sind vorgefertigte Rohlinge, die direkt zur Endbearbeitung zu Zahnprothesen von zahntechnischen Laboratorien zugekauft werden. Daher hängt die genaue Geometrie der Proben von Hersteller und Produktgruppe ab. Proben könnten als kreisförmige Scheiben oder rechteckige Platten vorliegen. Für die vorliegende Arbeit gab es ein großes Spektrum an Parametern (Scheiben bzw. Plattengröße und -dicke, Poissonzahl), das für numerische Ergebnisse durchgearbeitet werden musste.

Eine dementsprechend große Anzahl an FEM-Rechnungen wurde durchgeführt, und eine empirische Formel für die Maximalspannung unter der Lastkugel bestimmt. Die Berechnungen der FEM-Ergebnisse erfolgte in drei Durchgängen. Der erste diente zum Erstellen eines Datenpools für die Berechnung der Fit-Formel für die Maximalspannung. Hier wurden alle relevanten Parameter vom Minimum der Probenwerte bis zu deren Maximum variiert. Der zweite und dritte Durchgang behandelten Auswirkungen von geometrischen bzw. elastischen Abweichungen vom idealisierten Modell.

Die Daten des Pools wurden verwendet, um eine Fit-Formel für den Faktor f in der Formel $\sigma_m = \frac{F}{t^2} \cdot f$ zu finden. In die Formel für f fließen wie in vorigen Parametrisierungen nur normierte Werte des Radius und der Dicke ein. Im Gegensatz zu früheren Arbeiten werden diese Werte aber nicht durch den Probenradius normiert. Dieser ist bei den gegebenen Proben zu variabel, was das Vergleichen z.B. von Dicken untereinander unmöglich macht. Daher wurde für die Normierung der Auflagerkreis herangezogen, der für ein Auflager immer konstant ist. Es wurde auch gezeigt, dass das Material, welches über den Auflagerkreis hinausragt nur geringen Einfluss auf den Spannungszustand unter der Lastkugel hat. Im untersuchten Parameterbereich $\nu = 0.2$ bis $\nu = 0.3$ hängt f linear von ν ab. Es muss daher, abweichend von den bekannten Ausdrücken für f , nicht mehr für konkrete Poissonzahlen interpoliert werden. Die Abhängigkeit des Faktors f von relativer Probendicke, relativem Probenradius und Poissonzahl lässt sich mit sieben Koeffizienten beschreiben. Für Scheiben mit Radien von 5.85 bis 7.05 ist die Abweichung zwischen FEM-Ergebnissen und denen, die mithilfe der neuen Fitformel berechnet wurden $< 0.2\%$.

Die weiteren Berechnungsergebnisse dienten zum Ermitteln eines optimalen Ersatzradius, mit dessen Hilfe man Platten als Scheiben behandeln kann. Somit können diese ebenso einfach mit nur drei normierten Variablen beschrieben werden. Der optimale Ersatzradius berechnet sich als Mittelwert von Normalabstand der Lastkugel und der minimalen Distanz der Lastkugel zu einer Kante, gemessen durch die Mitte zweier Lagerkugeln. Die Berechnung der Maximalspannung in einer Platte ist mit Hilfe dieses Ersatzradius damit innerhalb von $\pm 0.78\%$ (im Vergleich zu den FEM-Ergebnissen für Platten) möglich.

Untersucht wurde auch der Einfluss einer hohen Bruchkraft bzw. eines niedrigen E-Moduls auf den Faktor f . Sowohl eine sehr hohe Bruchkraft als auch eine niedriger E-Modul (bzw. die große Durchbiegung, die dünne Proben entsprechender Werkstoffe erfahren) führen zu Reduktion des Faktors f im Vergleich zur idealisierten Lösung. Es wurde einerseits der Einfluss der Durchbiegung einer Probe und andererseits die Auswirkungen der Lastaufbringung über eine Hertz'sche Kontaktspannungsverteilung (statt einer Punktlast) auf f untersucht. So konnten für gegebene Verhältnisse von

Probendicke t zu Auflagerradius R_a eine normierte Festigkeit $\frac{\sigma_m}{E}$ errechnet werden, bei der ein systematischer Fehler bestimmter Größe auftritt. Daraus konnte eine ideale Dicke für Proben bei $\frac{t}{R_a} \approx 0.08$ festgestellt werden. Diese unterscheidet sich allerdings merklich von den idealen Dicken aus vorhergegangenen Arbeiten, die bei 0.3 liegen. Das kann zum einen daran liegen, dass hier die Einfüsse „E-Modul“ und „hohe Prüfkraft“ getrennt untersucht wurden, in vorangegangenen Arbeiten aber in einer FEM-Berechnung kombiniert wurden. Die FEM-Rechnungen hier wurden weiters für Proben durchgeführt, die einen niedrigeren E-Modul haben als die Kugeln, im vorangegangenen Fall war es umgekehrt.

Für die Interpretation von Festigkeitswerten und den Vergleich von Probensätzen, die mit unterschiedlichen Methoden geprüft wurden, ist es unbedingt nötig, das effektive Volumen bzw. die effektive Oberfläche der Proben zu kennen. Auch diese Größen wurden anhand der Ergebnisse der FEM-Berechnungen ermittelt. Vorherige Versionen solcher Berechnungen wurden direkt in der Programmiersprache des FEM-Programms *ANSYS*, *APDL*, implementiert und dauerten oft länger als die Lösung des eigentlichen FEM-Problems. Eine neue Berechnung wurde daher in Fortran90 programmiert, welches für den Umgang mit großen Tabellen optimiert ist. So konnte die Rechenzeit für effektive Größen um Größenordnungen verkürzt werden. Das Programm bedient sich der Daten welche vom FEM-Programme tabelliert zwischengespeichert werden. Diese Tabellen beinhalten die Spannungsinformationen und Elementzugehörigkeiten aller Knotenpunkte der Simulation. Sie werden dann umgeformt und mithilfe von Fortran-eigenen Funktionen tabellenweise berechnet. Die Applikation kann zwischen tetraedrischen und hexaedrischen Elementen unterscheiden und berechnet auch 3D und 2D Fälle. Durch diese Vielfältigkeit und ein Makro zur Datenaufbereitung wurde eine Wiederverwendbarkeit für späteren FEM-Berechnung ermöglicht.

Die berechneten Tabellen für die effektiven Volumina und Oberflächen wurden in einem eigenen Mathematica Notebook interpoliert. So wurde ein Applet erschaffen, das aus den großen Datenmengen und Tabellen genaue Werte für konkrete Probengeometrien berechnen kann.

Durch diese Informationen über systematische Fehler bei elastischen Proben und der Interpolation effektiver Größen war es möglich, experimentelle Daten geprüfter Proben auszuwerten und zu vergleichen. Vor allem eine Untergrenze für den E-Modul von ca. 40 GPa wurde dabei festgelegt. Elastischere Proben müssen entweder bei extrem niedrigen Spannungen versagen, um gültige Werte zu liefern, oder mit anderen Methoden geprüft werden.

Die hier durchgeführten Arbeiten zeigen, dass es grundsätzlich möglich ist, eine einfach handhabbare Formel für die B3B-Spannungsberechnung zu finden, die sowohl für runde Scheiben als auch für rechteckige Platten anwendbar ist und einen breiten Bereich von ν abdeckt. Um eine solche Formel universeller Gültigkeit — analog zum bereits bekannten Ausdruck — zu entwickeln sind jedoch weitere Berechnungen nötig. Aufgrund der neuen Normierungsgrundlage kann die Formel für f aber nur in jenen geometrischen und elastischen Bereichen als richtig angenommen werden, für die auch die FEM-Daten erstellt wurden. Um eine allgemein gültige Formel definieren zu können, muss ein weit größeres Spektrum an Geometrien und elastischen Eigenschaften berechnet werden. Dieses wird dann auch zeigen, ob die Annahmen dieser Arbeit (wie der rein lineare Einfluss der Poissonzahl) sich für größere Bereiche bewahrheiten. Auch die Gültigkeit des hier vorgeschlagenen Ersatzradius muss in einer größeren Studie bestätigt werden. Hier gilt es zu beantworten, ob der Durchschnitt aus kleinstem

Normalabstand und kleinster Schwerlinie auch bei extrem geformten Rechtecken gute Ergebnisse liefert. Wenn es Abweichungen gibt, muss entweder ein besserer Ersatzradius gefunden werden oder eine Grenze für das Verhältnis von Probenlänge zu -breite festgelegt werden. Andere Geometrien wie Polygone und allgemeinere Formen müssen eventuell auch berücksichtigt werden.

Die Interpolation der effektiven Größen V_{eff} und S_{eff} bedarf hier immer noch großer Datenmengen zwischen denen interpoliert werden muss, um gute Werte zu erhalten. Eine parametrisierte Formel (wie sie hier schon für f definiert wurde) zu finden, ist für die Interpretation von Festigkeitskennwerten von großer Bedeutung. Die Interpolation durch *Mathematica* funktioniert zwar, ist aber nicht auf *Excel* oder andere Datenverarbeitungsprogramme übertragbar.

Für eine Anwendung des B3B-Versuches an Werkstoffen mit hohem $\frac{\sigma_m}{E}$ sollten auch die Effekte hoher Prüfkräfte und niedriger E-Moduli auf den Faktor f näher untersucht werden, um optimale Probengeometrien vorgeben zu können und systematische Abweichungen in der Spannungsberechnung evaluieren zu können.

Literatur

1. *VITABLOCS®RealLife®for CEREC®/ inLab®MC XL – Verarbeitungsanleitung* (retrieved 2016).
2. Wendler, M., Belli, R., Petschelt, A., Mevec, D., Harrer, W., Lube, T., Danzer, R. & Lohbauer, U. Characterization of dental CAD/CAM blocks: II. Flexural Strength. *To be Submitted* (2016).
3. *ASTM C 1499 – 08, Standard Test Method for Monotonic Equiaxial Flexural Strength of Advanced Ceramics at Ambient Temperature* 2009.
4. Baratta, F. I., Matthews, W. T. & Quinn, G. D. *Errors associated with flexure testing of brittle materials* Techn. Ber. (DTIC Document, 1987).
5. Börger, A., Supancic, P. & Danzer, R. The ball on three balls test for strength testing of brittle discs: stress distribution in the disc. *Journal of the European Ceramic Society* **22**, 1425–1436. ISSN: 0955-2219 (2002).
6. Börger, A., Supancic, P. & Danzer, R. The ball on three balls test for strength testing of brittle discs: Part II: analysis of possible errors in the strength determination. *Journal of the European Ceramic Society* **24**, 2917–2928. ISSN: 0955-2219 (2004).
7. Danzer, R., Lube, T., Supancic, P. & Damani, R. Fracture of Ceramics. *Advanced Engineering Materials* **10**, 275–298 (2008).
8. Rösler, J., Harders, H. & Bäker, M. *Mechanisches Verhalten der Werkstoffe 2*. (Teubner, 2006).
9. Irwin, G. R. Analysis of stresses and strains near the end of a crack traversing a plate. *Journal of Applied Mechanics* **24**, 361–364 (1957).
10. Griffith, A. A. The phenomena of rupture and flow in solids. *Philosophical Transactions of the Royal Society of London A* **221**, 163–198 (1921).
11. Griffith, A. A. *Theory of Rupture* in *First International Congress on Applied Mechanics* (1924).
12. Danzer, R., Supancic, P., Pascual, J. & Lube, T. Fracture statistics of ceramics – Weibull statistics and deviations from Weibull statistics. *Engineering Fracture Mechanics* **74**, 2919–2932. ISSN: 0013-7944 (2007).
13. Danzer, R., Reisner, G. & Schubert, H. Der Einfluß von Gradienten in der Defektdichte und Festigkeit auf die Bruchstatistik von spröden Werkstoffen. *Zeitschrift für Metallkunde* **83**, 508–517 (1992).
14. Danzer, R. Some notes on the correlation between fracture and defect statistics: Are Weibull statistics valid for very small specimens? *Journal of the European Ceramic Society* **26**, 3043–3049. ISSN: 0955-2219 (2006).
15. Weibull, W. A statistical distribution function with wide applicability. *Journal of Applied Mechanics* **18**, 293–297 (1951).
16. Danzer, R. A general strength distribution function for brittle materials. *Journal of the European Ceramic Society* **10**, 461–472. ISSN: 0955-2219 (1992).

17. *ENV 843-5, Advanced technical ceramics: Monolithic ceramics. Mechanical tests at room temperature, Part 5: Statistical analysis* 1995.
18. Brückner-Foit, A., Heger, A. & Munz, D. Evaluation of Failure Probabilities of Multiaxially Loaded Components Using the STAU Postprocessor. *Ceramic Engineering and Science Proceedings* **14**, 331–331 (1993).
19. Nemeth, N., Manderscheid, J. & Gyekenyesi, J. Designing Ceramic Components for Reliability. *Am. Ceram. Bull* **68**, 2064–2072 (1989).
20. Munz, D. & Fett, T. *Ceramics: Mechanical Properties, Failure Behaviour, Materials Selection* 2. Aufl. (Springer, 2001).
21. Von Mises, R. Mechanik der festen Körper im plastisch-deformablen Zustand. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* **1913**, 582–592 (1913).
22. Tresca, H. Memoire sur l'écoulement des solides à de forte pressions. *Acad. Sci. Paris* **2**, 59 (1864).
23. Gross, D. *Bruchmechanik* 4. Aufl. (Springer, 2007).
24. Barnett, R. L., Hermann, P. C., Wingfield, J. R. & Connors, C. L. *Fracture of Brittle Materials Under Transient Mechanical and Thermal Loading* Techn. Ber. AFFDL-TR-66-220 (Air Force Flight Dynamics Laboratory, 1967).
25. Freudenthal, A. M. Statistical approach to brittle fracture. *Fracture* **2**, 591–619 (1968).
26. Stanley, P., Fessler, H. & Sivill, A. D. *An engineer's approach to the prediction of failure probability of brittle components* in *Proc. Brit. Ceram. Soc.* **22** (1973), 453–487.
27. Mathiak, F. U. *Die Methode der finiten Elemente (FEM). Einführung und Grundlagen* <http://www.mechanik-info.de/dokumente/Skript_FEM.pdf> (FH Neubrandenburg, 2010).
28. *ANSYS® Academic Research, Release 16.2* 2015.
29. *ANSYS Documentation ¿ Mechanical APDL ¿ E ()*. <help/ans_elem/Hlp_E_ElemLib.html>.
30. *Ball on 3 Balls- Test (web-App)* 2011. <www.isfk.at/de/960/>.
31. Danzer, R., Harrer, W., Supancic, P., Lube, T., Wang, Z. & Börger, A. The ball on three balls test—Strength and failure analysis of different materials. *Journal of the European Ceramic Society* **27**, 1481–1485. ISSN: 0955-2219 (2007).
32. Harrer, W., Danzer, R., Supancic, P. & Lube, T. Influence of the Sample Size on the Results of B3B-Tests. *Key Engineering Materials* **409**, 176–184. ISSN: 1662-9795 (2009).
33. Danzer, R., Supancic, P. & Harrer, W. in (Hrsg. Kriegesmann, J.) 1–48 (HvB Verlag GbR., Ellerau, 2009).
34. Roark, R. J. *Roark's Formulas for Stress and Strain* 8th (Hrsg. Young, W. C. & Budynas, R. G.) (McGraw-Hill, 2002).
35. Hertz, H. Über die Berührung fester elastischer Körper. *Journal für die reine und angewandte Mathematik* **92**, 165–171 (1882).

Anhang

A.ANSYS Syntax

Der *ANSYS APDL* Code läuft folgendermaßen ab: Das *Mathematica*-Notebook `multirun_data-export.nb` erzeugt ein File, das alle Parameter der Versuchsserie beinhaltet. Dann wird in *ANSYS* das File `multirun.inp` aufgerufen, welches die Steuerung der Serie übernimmt.

Es werden für jede Simulation der Speicher gelöscht, neue Parameter geladen und das File `main.inp` aufgerufen. Dieses weist der Berechnung einen Namen zu und erstellt die Geometrie und das Netz mithilfe von entweder `geom_Platte_Punkt.inp` oder `geom_Scheibe_Punkt.inp`.

Dann werden Randbedingungen und Kräfte zugeteilt und das Modell berechnet.

Nach der fertigen Rechnung wird falls gewünscht ein Bild gespeichert. Falls die effektiven Größen gefragt sind, wird Geometrie-abhängig `post_veff_full.inp` oder `post_veff_6.inp` aufgerufen. Diese Postprocessing Files bestimmen die Knoten und Elemente für die Berechnung von V_{eff} und S_{eff} .

Das Macro `x_Effektives_Volumen.mac` erzeugt Tabellen für das sie Berechnung im Fortran Programm, wie in Abschnitt 3.3.1 erläutert.

Zu guter Letzt werden durch `write_result.inp` noch in einer Sammeltable die wichtigsten Eckparameter und Ergebnisse der Berechnung vermerkt, bevor die nächste begonnen wird. Unter diesen Eckdaten ist die Berechnungsdauer, die durch `x_readtime_i.mac` ermittelt wird.

A.1. multirun.inp

```
1 !#####
2 !#
3 !#           multirun.inp, dme, 2015-12-27
4 !#
5 !#           Befehl zum ausführen eines Berechnungsbatches zur Simulation
6 !#           eines B3B-Versuchs zur Prüfung einer rechteckigen Platte
7 !#
8 !#
9 !#
10 !#           /INPUT,multirun,inp
11 !#
12 !#   Abhängigkeiten:
13 !#           main.inp
14 !#           multirun_input.csv
15 !#           (multirun_data-export.nb)
16 !#           s_write.inp
17 !#           vwrite_info.inp
18 !#           x_readtime_i.mac
19 !#           geom_Platte_Punkt.inp
20 !#           geom_Scheibe_Punkt.inp
21 !#           post_veff_6.inp
22 !#           post_veff_full.inp
23 !#
24 !#           Stand: 2016-01-20
25 !#
26 !#####
27
28 /BATCH
29 FINISH
30 /CLEAR
31
32
33 !===== PARAMETER =====!
```

```

34
35 !~~~~~ Material ~~~~~!
36
37 db = 8 ! Lagerkugel Durchmesser [mm]
38
39 lp = 12 ! Plattenmaße in [mm]
40 bp = 12
41 tp = 1
42 rd = 6 ! Scheibenradius [mm]
43
44 rb=db/2 ! Kugelradius [mm]
45 ra=db/SQRT(3) ! Auflagerradius [mm]
46 rard=ra/rd ! Verhältnis Auflagerkreis zu Berührkreis mit Rand [-]
47
48 ep = 100E3 ! E-Modul Platte in [N/mm²]
49 np = 0.2 ! Poissonzahl Platte
50
51 eb = 200E3 ! E-Modul Kugeln in [N/mm²]
52 nb = 0.3 ! Poissonzahl Kugeln
53
54 fb = 1 ! Audruckkraft in [N]
55
56 !~~~~~ Logische Schalter ~~~~~!
57
58 L_Bilder = 1 ! Abfrage ob Bilder gespeichert werden sollen
59 L_Tabellen = 0 ! Abfrage ob Spannungsverläufe gespeichert werden sollen
60 L_Geom = 0 ! Abfrage für Kontaktproblem
61 ! 0:= Punktlast auf Platte,
62 ! 1:= Punktlast auf Scheibe,
63 ! 2:= Kontaktlast auf Platte,
64 ! 3:= Kontaktlast auf Scheibe,
65 L_ASYM = 0 ! Asymmetrischer Matrizensolver
66 ! (warnung, verdoppelt Rechenzeit!)
67
68 !~~~~~ Netz ~~~~~!
69
70 mcirc = 50 ! Meshunterteilung am Plattenrand als Größenunabhängige
    Meshbemaßung
71 mgra = (3.14159265/3)*(rd/mcirc)! Meshgröße in Ebene
72 mgrk = mgra*3 ! Meshgröße in Kugeln
73 ndivt = 10 ! Meshteilung in Dicke
74 mrad = ra/4 ! Mesh radius um Kontaktpunkt Musskleiner sein als ra/2!
75
76 !~~~~~ Fehler ~~~~~!
77
78 pphi_err = 0 ! Verdrehung der Platte zum Lager in [deg]
79 ! >0: gegen UZS
80 ! <0: in UZS
81 ! -60deg < PHI < 60deg
82
83 pdx_err = 0.0 ! Versatz Platte zum Lager in [mm]
84 ! >0: Kugeln weiter weg von Rand
85 ! <0: Kugeln näher Rand
86
87 pdy_err = 0.0 ! Versatz Platte zum Lager in [mm]
88 ! >0: Kugeln weiter weg von Rand
89 ! <0: Kugeln näher Rand
90
91 kd_err = 0.0 ! Versatz Einzelkugel gegen Lager in [mm]
92 ! Bitte nur >0 !
93
94 kphi_err = 0 ! Richtungswinkel des Versatzes Einzelkugel gegen Lager in
    [deg]
95 ! >0: gegen UZS
96 ! <0: in UZS
97 ! 0deg < PHI < 360deg
98
99
100 !===== OUTPUT VORBEREITEN =====!
101
102 *CFOPEN,Ergebnis, out,,
103 /INPUT,write_result,inp
104 *CFCLOSE

```

```

105                                     ! Dateipfade erstellen
106 /SYP,mkdir,'plots'
107 /SYP,mkdir,'tables'
108
109
110 !===== SIMULATION STARTEN =====!
111
112
113 /INPUT,multirun_input,inp
114 *DO,z_,1,y_
115
116 /INPUT,multirun_input,inp,,:P_DIM
117
118 db = pardata(1,z_)           ! Lagerkugel Durchmesser [mm]
119 lp = pardata(2,z_)           ! Plattenmaße in [mm]
120 bp = pardata(3,z_)
121 tp = pardata(4,z_)
122 rd = pardata(5,z_)           ! Verhältnis Auflagerkreis zu Berührkreis mit Rand
123 dx = pardata(6,z_)
124 dy = pardata(7,z_)
125 rb = db/2
126 ra = db/SQRT(3)
127 mrad = ra/4
128 ep = pardata(8,z_)           ! E-Modul Platte in [N/mm2]
129 np = pardata(9,z_)           ! Poissonzahl Platte
130 eb = pardata(10,z_)          ! E-Modul Kugeln in [N/mm2]
131 nb = pardata(11,z_)          ! Poissonzahl Kugeln
132 fb = pardata(12,z_)          ! Audruckkraft in [N]
133 L_Bilder = pardata(13,z_)     ! Abfrage ob Bilder gespeichert werden sollen
134 L_Tabellen = pardata(14,z_)  ! Abfrage ob Spannungsverläufe gespeichert werden
    sollen
135 L_Geom = pardata(15,z_)       ! Abfrage für Kontaktproblem
136 L_Veff = pardata(16,z_)       ! Asymmetrischer Matrizensolver
137 mcirc = pardata(17,z_)       ! Meshunterteilung am Plattenrand als Größenunabhä
    ngige Meshbemaßung
138 mgra = (3.14159265/3)*(rd/mcirc)! Meshgröße in Ebene
139 mgrk = mgra*3                 ! Meshgröße in Kugeln
140 mrad = ra/4
141 ndivt = pardata(18,z_)       ! Meshteilung in Dicke
142
143 /INPUT,main,inp
144 *ENDDO

```

A.2. multirun_data-export.nb

```

1 (*Kopfzeile, damit in Ausgabe verglichen werden kann*)
2 ausgabe = {'KugelD', "PlattenL", "PlattenB", "PlattenT", "RD", "dx", "dy", "PlattenE",
    "Plattenv", "KugelE", "Kugelv", "Kraft", "Bilder?", "Tabellen?", "Geometrie", "
    Veff", "MeshCirc", "nDiv", "Name"}, {'[mm]', "[mm]", "[mm]", "[mm]", "[mm]", "[mm]",
    "[-]", "[-]", "[-]", "[-]", "[-]", "[-]", "[-]", "[-]", "[-]", "[-]", "[-]",
    "[-]"};
3
4 (*Scheiben zum Vergleich*)
5 ausgabe = Join[ausgabe, {8, 0, 0, 1, #, 0, 0, 100*3, 0.25, 210*3, 0.33, 1, bld, tab,
    1, 1, 50, 30, 0} & /@ Table[r, {r, 117/20, 141/20, 6/20}]];
6
7
8 (*Platten, exzentrisch belastet*)
9 ausgabe = Join[ausgabe, {8, #[[1]], #[[2]], 1, 5.815, #[[3]], #[[4]], 100*3, 0.25,
    210*3, 0.33, 1, bld, tab, 0, 1, 40, 30, 0} & /@ Flatten[Table[{1, b, 1/2 - 6.309,
    b/2 - 5.815}, {b, 117/10, 141/10, 12/10}, {1, 117/10, 141/10, 12/10}],]];
10
11 (*Einfügen der einzelnen Namen der Rechnungen*)
12 ausgabe[[3 ;;, -1]] = Table["Sweep" <> ToString[n], {n, Length[ausgabe] - 2}];
13
14 (*Doppelte Aufträge löschen*)
15 data = DeleteDuplicates[ausgabe][[3 ;; -1, 1 ;; -2]]*1.;
16
17 SetDirectory[NotebookDirectory[]];
18
19 (*Werte in ANSYS APDL gerechte Formatierung bringen*)

```

```

20 datei = Join[{"y_ = " <> ToString[Length[data]], "", "/EOF", "", ":P_DIM", "", "*DEL,
    pardata(1,1),NOPR $ *DIM,pardata,ARRAY," <> ToString[Length[data[[1]]]] <> "," <>
    ToString[(Length[data])]}], Table["pardata(1," <> ToString[1] <> ")=" <>
    StringReplacePart[ToString[data[[1]]], "", {{1, 1}, {-1, -1}}, {1, 1, Length[data
    ]}]];
21
22 (*Auftragsnamen herausklauben*)
23 data2 = DeleteDuplicates[ausgabe][[3 ; -1, -1]];
24
25 (*Namen in ANSYS APDL gerechte Formatierung bringen*)
26 datei2 = Table["nametable(" <> ToString[n] <> ")=' ' <> data2[[n]] <> '", {n, Length[
    data2]}];
27
28 (*Seperator zwischen Zwei zuteilungen, damit APDL-Code einzeln zugreifen kann*)
29 datei2 = Join[{"", "", "/EOF", "", ":P_NAME", "", "*DEL,nametable(1),NOPR $ *DIM,
    nametable,CHAR," <> ToString[Length[data]], ""}, datei2];
30
31 (*Zusammenfügen*)
32 datei = Join[datei, datei2];
33 TableForm[datei];
34
35 (*Exportieren und Endung richtig einstellen*)
36 Export["multirun_input.txt", datei];
37 DeleteFile["multirun_input.inp"];
38 RenameFile["multirun_input.txt", "multirun_input.inp"];

```

A.3. main.inp

```

1 !#####
2 !#
3 !#           main.inp, dme, 2015-125-27
4 !#
5 !#           Hauproutine des B3B-Platten-Simulationsprogramms
6 !#
7 !#           /INPUT,main,inp
8 !#
9 !#           ACHTUNG:
10 !#           Wird in "Singlerun.inp" oder "Multirun.inp" aufgerufen.
11 !#           NICHT alleine starten, benötigt die Variablen
12 !#           und Ornder die diese Files erstellen
13 !#
14 !#           Abhängigkeiten: write_result.inp
15 !#                           x_readtime_i.mac
16 !#                           geom_Platte_Punkt.inp
17 !#                           geom_Scheibe_Punkt.inp
18 !#
19 !#
20 !#
21 !#           Stand: 2016-01-20
22 !#
23 !#####
24
25 PARSAV,,variablen,tmp
26 /CLEAR
27 PARRES,,variablen,tmp
28
29 /PNUM,KP,1
30 /PNUM,AREA,1
31 /PNUM,LINE,1
32 !===== OUTPUT VORBEREITEN #2 =====!
33
34 /inquire,wrkdir,directory           ! Working Directory bestimmen
35 *DIM,picdir,STRING,128              ! Bilderpfad-Variable erzeugen
36 *DIM,tabdir,STRING,128              ! Tabellenpfad-Variable erzeugen
37 *DIM,dataname,STRING,248            ! Outputdateinamen
38 picdir(1)='%wrkdir(1)/plots'        ! Bilderpfad festlegen
39 tabdir(1)='%wrkdir(1)/tables'        ! Tabellenpfad festlegen
40
41 /INPUT,multirun_input,inp,,:P_NAME
42 serienname=nametable(z_)
43
44 *IF,MOD(L_Geom,2),EQ,0,THEN

```

```

45     /TITLE, B3B-Platten-Test: %seriename%, K: %db%mm, P: %lp%x%bp%x%tp%mm^3
46     dataname(1)='B3B_%seriename%_square'
47     *ELSE
48     /TITLE, B3B-Scheiben-Test: %seriename%, K: %db%mm, R: %rd%mm t: %tp%mm
49     dataname(1)='B3B_%seriename%_round'
50     *ENDIF
51
52
53     !===== GEOMETRIE =====!
54     /PREP7
55     *AFUN,DEG                                ! Fixiere System auf Winkelgrad
56
57     xp0=-lp/2+dx
58     xp1=lp/2+dx
59     yp0=-bp/2+dy
60     yp1=bp/2+dy
61
62     *DIM,kpkt,ARRAY,4,2
63
64     kpkt(1,1)=0                                ! Koord. zentraler Kugelberührungspunkt
65     kpkt(1,2)=0
66
67     kpkt(2,1)=ra*COS(0)                        ! Koord. 1. Lagerkugelberührungspunkt
68     kpkt(2,2)=ra*SIN(0)
69
70     kpkt(3,1)=ra*COS(120)                     ! Koord. 2. Lagerkugelberührungspunkt
71     kpkt(3,2)=ra*SIN(120)
72
73     kpkt(4,1)=ra*COS(240)                     ! Koord. 3. Lagerkugelberührungspunkt
74     kpkt(4,2)=ra*SIN(240)
75
76
77     *IF,L_Geom,EQ,0,THEN
78     /INPUT,geom_Platte_Punkt,inp
79     *ELSE
80     /INPUT,geom_Scheibe_Punkt,inp
81     *ENDIF
82
83
84     !===== MESH =====!
85
86     !~~~~~ Materialeigenschaften ~~~~~!
87
88     MP,EX,1,ep                                ! E-Modul der Platte
89     MP,NUXY,1,np                              ! Nü der Platte
90     MP,MU,1,0.3                               ! Reibungskeffizient d. Platte
91
92     MP,EX,2,eb                                ! E-Modul der Kugeln
93     MP,NUXY,2,nb                              ! Nü der Kugeln
94     MP,MU,2,0.3                               ! Reibungskeffizient d. Stahls
95
96     TREF,0                                    ! Refenztemp, eigentlich egal
97
98
99     !~~~~~ Netzeigenschaften ~~~~~!
100
101     ET,1,SOLID186                             ! Quadratisches Würfel Netz
102
103     ET,2,SOLID187                             ! Quadratisches Tetraeder Netz
104
105     ET,3,MESH200                              ! Physikloses Mesh
106     KEYOPT,3,1,5                             ! quadratisches 3-Eck Netz
107
108     ET,4,MESH200                              ! Physikloses Mesh
109     KEYOPT,4,1,7                             ! quadratisches 4-Eck Netz
110
111
112     !~~~~~ Netz generieren ~~~~~!
113
114     *IF,L_Geom,EQ,0,THEN
115     /INPUT,geom_Platte_Punkt,inp,,P_MESHING
116     *ELSE
117     /INPUT,geom_Scheibe_Punkt,inp,,P_MESHING

```

```

118      *ENDIF
119
120
121      !===== RANDBEDINGUNGEN =====!
122
123
124      *IF ,L_Geom ,EQ ,0 , THEN
125
126          F ,NODE (kpkt (1 ,1) , kpkt (1 ,2) , -tp) , FZ , fb
127
128          D ,NODE (kpkt (2 ,1) , kpkt (2 ,2) , 0) , UX , 0
129          D ,NODE (kpkt (2 ,1) , kpkt (2 ,2) , 0) , UY , 0
130          D ,NODE (kpkt (2 ,1) , kpkt (2 ,2) , 0) , UZ , 0
131
132          D ,NODE (kpkt (3 ,1) , kpkt (3 ,2) , 0) , UY , 0
133          D ,NODE (kpkt (3 ,1) , kpkt (3 ,2) , 0) , UZ , 0
134
135          D ,NODE (kpkt (4 ,1) , kpkt (4 ,2) , 0) , UZ , 0
136
137      *ELSE
138
139          F ,NODE (0 ,0 , -tp) , FZ , fb/6
140
141          D ,NODE (ra ,0 ,0) , UZ , 0
142
143          NSEL , S , LOC , Y , 0
144          NSEL , U , LOC , X , 0
145          DSYM , SYMM , Y , 0
146
147          LOCAL , 15 , , , , 60
148          NSEL , S , LOC , Y , 0
149          NSEL , U , LOC , X , 0
150          DSYM , SYMM , Y , 15
151          CSYS , 0
152
153          NSEL , S , LOC , X , 0
154          NSEL , R , LOC , Y , 0
155          D , ALL , UX , 0
156          D , ALL , UY , 0
157
158          ALLSEL
159      *ENDIF
160
161      FINISH
162
163
164
165      !===== LÖSUNG =====!
166
167      /SOLU
168
169      EQSLV , PCG , 1E-8                ! PCG-Solver mit hoher Genauigkeit
170      !EQSLV , SPARSE                ! Sparse-Solver
171
172      SOLVE
173
174      FINISH
175
176
177      !===== POSTPROCESSING =====!
178
179      /POST1
180
181      CSYS , 1
182      NSEL , S , LOC , X , 0 , mrad
183      NSEL , R , LOC , Z , 0
184      NSORT , S , 1
185      *GET , smax , SORT , 0 , MAX
186      ALLSEL
187      CSYS , 0
188
189
190      !~~~~~ Bilder ~~~~~!

```

```

191
192 *IF,L_Bilder,EQ,1,THEN
193
194 /DSCALE,ALL,1
195 /RGB,INDEX,100,100,100,0 ! Schwarzen Hintergrund auf Weiß setzen
196 /RGB,INDEX,80,80,80,13 ! Grautöne tauschen
197 /RGB,INDEX,60,60,60,14 ! Grautöne tauschen
198 /RGB,INDEX,0,0,0,15 ! Weiße Linien schwarz machen
199 /GFILE,800 ! Bildhöhe festsetzen
200
201 VSEL,S,LOC,Z,0,-tp ! Plattenspezifische NODES auswählen
202 ESLV,S ! (für den Fall des Kontaktproblems)
203 /AUTO,1 ! Ausgewählte Objekte zentrieren
204
205 /CONTOUR,1,9,0,,smax ! Setze Farbkonturen von 0 bis zur M
206 /DSCALE,1,1 ! Verstärkungsfaktor für Verzerrungen ist 1
207 /TYPE,,4 ! PRECISE hidden line removal
208 /GRAPH,POWER ! Aktiviere PowerGraphics
209 /EFACET,4 ! Höhere Konturqualität
210
211 *IF,L_Geom,EQ,1,THEN
212 /EXPAND,6,POLAR,HALF,,60
213 *ENDIF
214
215 /CWD,picdir(1) ! Working Directory wechseln zum Bilderordner
216
217 /SHOW,png ! Folgenede Plot-Befehle an PNG Dateien
218 schicken
219 PLNSOL,S,1,,1 ! Nodale Lösung der ersten Hauptspannung
220 darstellen
221 /SHOW,CLOSE ! Bildexport abschließen
222 /INQUIRE,tempname_,JOBNAME ! Jobname erfragen
223 /RENAME,'%tempname_(1)%000',png,, 'PLOT_%dataname(1)%',png ! ausgegebene Bilder
224 umbenennen
225
226 /CWD,wrkdir(1) ! Working Directory zurück wechseln
227 ALLSEL
228
229 *ENDIF
230
231 !~~~~~ Veff bestimmen ~~~~~!
232
233 *IF,L_Veff,EQ,1,THEN
234
235 *IF,L_Geom,EQ,0,THEN
236 /INPUT,post_veff_full,inp
237 *USE,'x_Effektives_Volumen.mac',,smax,5,50,,34
238 *ELSE
239 /INPUT,post_veff_6,inp
240 *USE,'x_Effektives_Volumen.mac',,smax,5,50,6,34
241 *ENDIF
242
243 /RENAME,'effVol',out,, 'VEFF_%dataname(1)%',out
244 /SYP,mv,'VEFF_%dataname(1)%_out','tables'
245 *ENDIF
246
247 !===== OUTPUT SCHREIBEN =====!
248
249
250
251 *CFOPEN,Ergebnis,out,,APPEND
252 /INPUT,write_result,inp,,:P_write
253 *CFCLOSE
254
255 FINISH

```

A.4. geom_Platte_Punkt.inp

```
1  !#
2  !# geom_Platte_Punkt.inp, dme, 2016-01
3  !#
4  !# Subroutine zum Erstellen der Geometrie einer rechteckigen Platte im B3B-Test
5  !# gerechnet als idealisiertes Kontaktproblem --> Lagerkugeln werden als Punktlasten
   approximiert
6  !#
7  !# Stand: 2016-01-20
8  !#
9
10  !*IF,ra+mrاد/0.9,GE,rd,THEN
11  !   mrad=(rd-ra)*0.8
12  !*ENDIF
13
14  RECT, xp0, xp1, yp0, yp1          ! Platten Außengeometrie
15
16  ! Lokale Koordinatensysteme erstellen
17  ! für spätere Analysen
18  LOCAL, 100, CYLIN, kpkt(1,1), kpkt(1,2), -tp
19  LOCAL, 200, CYLIN, kpkt(2,1), kpkt(2,2), 0
20  LOCAL, 300, CYLIN, kpkt(3,1), kpkt(3,2), 0
21  LOCAL, 400, CYLIN, kpkt(4,1), kpkt(4,2), 0
22
23  *DO, j_, 1, 4
24
25  ! Rechtecke so anlegen dass Anlagepunkte als Keypoints vorliegen
26  CSYS, j_*100
27
28  ! Prüfen: ist der Berührungspunkt zu nahe am Rand?
29  ! --> wenn, dann Mapped Mesh Rechtecke zum Rand
30  ! --> wenn nicht Mapped Mesh Viertelkreise!!
31
32  s_fakt=1.1
33
34  *IF, kpkt(j_,1)+mrad*s_fakt, GT, xp1, AND, kpkt(j_,2)+mrad*s_fakt, GT, yp1, THEN
35  RECT, kpkt(j_,1)-, xp1, kpkt(j_,2), yp1
36  RECT, kpkt(j_,1)-, xp1, kpkt(j_,2), kpkt(j_,2)-mrad
37  RECT, kpkt(j_,1)-, kpkt(j_,1)-mrad, kpkt(j_,2), yp1
38  CYL4, kpkt(j_,1), kpkt(j_,2), 0, 180, mrad, 270
39
40  *ELSEIF, kpkt(j_,1)+mrad*s_fakt, GT, xp1, AND, kpkt(j_,2)-mrad*s_fakt, LT, yp0, THEN
41  RECT, kpkt(j_,1), xp1, kpkt(j_,2), yp0
42  RECT, kpkt(j_,1), xp1, kpkt(j_,2), kpkt(j_,2)+mrad
43  RECT, kpkt(j_,1), kpkt(j_,1)-mrad, kpkt(j_,2), yp0
44  CYL4, kpkt(j_,1), kpkt(j_,2), 0, 90, mrad, 180
45
46  *ELSEIF, kpkt(j_,1)-mrad*s_fakt, LT, xp0, AND, kpkt(j_,2)-mrad*s_fakt, LT, yp0, THEN
47  RECT, kpkt(j_,1), xp0, kpkt(j_,2), yp0
48  RECT, kpkt(j_,1), xp0, kpkt(j_,2), kpkt(j_,2)+mrad
49  RECT, kpkt(j_,1), kpkt(j_,1)+mrad, kpkt(j_,2), yp0
50  CYL4, kpkt(j_,1), kpkt(j_,2), 0, 0, mrad, 90
51
52  *ELSEIF, kpkt(j_,1)-mrad*s_fakt, LT, xp0, AND, kpkt(j_,2)+mrad*s_fakt, GT, yp0, THEN
53  RECT, kpkt(j_,1), xp0, kpkt(j_,2), yp1
54  RECT, kpkt(j_,1), xp0, kpkt(j_,2), kpkt(j_,2)-mrad
55  RECT, kpkt(j_,1), kpkt(j_,1)+mrad, kpkt(j_,2), yp1
56  CYL4, kpkt(j_,1), kpkt(j_,2), 0, 270, mrad, 360
57
58  *ELSEIF, kpkt(j_,1)-mrad*s_fakt, LT, xp0, THEN
59  RECT, kpkt(j_,1), xp0, kpkt(j_,2), kpkt(j_,2)-mrad
60  RECT, kpkt(j_,1), xp0, kpkt(j_,2), kpkt(j_,2)+mrad
61  CYL4, kpkt(j_,1), kpkt(j_,2), 0, 0, mrad, 90
62  CYL4, kpkt(j_,1), kpkt(j_,2), 0, 270, mrad, 360
63
64  *ELSEIF, kpkt(j_,1)+mrad*s_fakt, GT, xp1, THEN
65  RECT, kpkt(j_,1), xp1, kpkt(j_,2), kpkt(j_,2)-mrad
66  RECT, kpkt(j_,1), xp1, kpkt(j_,2), kpkt(j_,2)+mrad
67  CYL4, kpkt(j_,1), kpkt(j_,2), 0, 90, mrad, 180
68  CYL4, kpkt(j_,1), kpkt(j_,2), 0, 180, mrad, 270
69
```

```

70      *ELSEIF ,kpkt(j_,2)-mrad*s_fakt,LT,yp0,THEN
71          RECT,kpkt(j_,1),kpkt(j_,1)+mrad,kpkt(j_,2),yp0
72          RECT,kpkt(j_,1),kpkt(j_,1)-mrad,kpkt(j_,2),yp0
73          CYL4,kpkt(j_,1),kpkt(j_,2),0,0,mrad,90
74          CYL4,kpkt(j_,1),kpkt(j_,2),0,90,mrad,180
75
76      *ELSEIF ,kpkt(j_,2)+mrad*s_fakt,GT,yp1,THEN
77          RECT,kpkt(j_,1),kpkt(j_,1)+mrad,kpkt(j_,2),yp1
78          RECT,kpkt(j_,1),kpkt(j_,1)-mrad,kpkt(j_,2),yp1
79          CYL4,kpkt(j_,1),kpkt(j_,2),0,180,mrad,270
80          CYL4,kpkt(j_,1),kpkt(j_,2),0,270,mrad,360
81
82      *ELSE
83          CYL4,kpkt(j_,1),kpkt(j_,2),0,0,mrad,90
84          CYL4,kpkt(j_,1),kpkt(j_,2),0,90,mrad,180
85          CYL4,kpkt(j_,1),kpkt(j_,2),0,180,mrad,270
86          CYL4,kpkt(j_,1),kpkt(j_,2),0,270,mrad,360
87
88      *ENDIF
89
90      *ENDDO
91
92      CSYS,0
93      APTN,ALL          ! Flächen zusammenfügen
94      ALLSEL
95
96      ASEL,U,LOC,X,yp0,yp1          !
97      ASEL,U,LOC,Y,yp0,yp1          ! überstehende Flächen wählen und löschen
98      ADEL,ALL          !
99      ALLSEL
100
101 /EOF
102
103 :P_MESHING
104
105
106 !===== MESH =====!
107
108     REAL,1
109     MAT,1
110
111     TYPE,4          ! Rechteckiges MESH200
112     MSHAPE,0        ! Rechtecke bevorzugen
113     ESIZE,mgra/1.5  ! Elementgröße
114     MSHKEY,1        ! Mapped Mesh
115
116     *DO,j_,1,4      ! Flächen für Mapped Mesh wählen
117         CSYS,j_*100
118         !LSEL,S,LOC,X,mgra,mrad ! Linien im Umkreis zu Berührungspunkt
119         !ASLL,S,1      ! von Linien umschlossene Flächen
120
121         KSEL,S,,,KP(0,0,0)      ! Kontakt-KP wählen
122         LSLK,S          ! zu KP verbundene Linien
123         ASLL,S          ! zu Linien verbundene Flächen
124         AMESH,ALL      ! Flächen meshen
125         ALLSEL
126     *ENDDO
127
128     CSYS,0          ! Hauptkoordinatensystem
129
130     ESIZE,mgra      ! Elementgröße
131     MSHKEY,0        ! Free Mesh
132
133     AMESH,ALL      ! Rest meshen
134
135     TYPE,1          ! Qudaratishes Hexaeder Mesh
136     EXTOPT,ESIZE,ndivt,1      ! Anzahl Elemente in Extrusionsrichtung
137     EXTOPT,ACLEAR,1      ! Flächenmesh aus dem extrudiert wird nachher löschen
138
139     VEXT,ALL,,,,-tp      ! Plattenvolumen extrudieren
140
141     ALLSEL

```

A.5. geom_Scheibe_Punkt.inp

```

1  !#
2  !# geom_Platte_Punkt.inp, dme, 2016-01
3  !#
4  !# Subroutine zum Erstellen der Geometrie einer runden Scheibe im B3B-Test
5  !# gerechnet als idealisiertes Kontaktproblem --> Lagerkugeln werden als Punktlasten
    approximiert
6  !#
7  !# Stand: 2016-01-20
8  !#
9
10     *IF,ra+mrاد,EQ,rd,THEN
11         mrad=mrad*1.2
12     *ENDIF
13
14     CYL4,0,0,,rd,60          ! Scheibe Außengeometrie
15
16     ! Lokale Koordinatensysteme erstellen
17     ! für spätere Analysen
18     LOCAL,100,CYLIN,0,0,-tp
19     LOCAL,200,CYLIN,ra,0,0
20     CSYS,0
21
22     CYL4,0,0,0,0,mrad,60
23
24     CYL4,ra,0,0,90,mrad,180
25     CYL4,ra,0,0,0,mrad,90
26
27     CSYS,1
28     APTN,ALL                ! Flächen zusammenfügen
29     ALLSEL
30     ASEL,U,LOC,X,0,rd
31     ADEL,ALL
32     ALLSEL
33     CSYS,0
34
35 /EOF
36
37 :P_MESHING
38
39 !===== MESH =====!
40
41
42     REAL,1
43     MAT,1
44
45     TYPE,4                  ! Rechteckiges MESH200
46     MSHAPE,0                ! Rechtecke bevorzugen
47     ESIZE,mgra/1.5          ! Elementgröße
48     MSHKEY,1                ! Mapped Mesh
49
50
51     CSYS,100
52
53     KSEL,S,,KP(0,0,0)       ! Kontakt-KP wählen
54     LSLK,S                  ! zu KP verbundene Linien
55     ASLL,S                  ! zu Linien verbundene Flächen
56     AMESH,ALL               ! Flächen meshen
57     ALLSEL
58
59     CSYS,200
60
61     KSEL,S,,KP(0,0,0)       ! Kontakt-KP wählen
62     LSLK,S                  ! zu KP verbundene Linien
63     ASLL,S                  ! zu Linien verbundene Flächen
64     AMESH,ALL               ! Flächen meshen
65     ALLSEL
66
67     CSYS,0
68
69     ESIZE,mgra              ! Elementgröße

```

```

70     MSHKEY ,0                ! Free Mesh
71
72     AMESH ,ALL
73
74     TYPE ,1                  ! Qudaratisches Hexaeder Mesh
75     EXTOPT ,ESIZE ,ndivt ,1  ! Anzahl Elemente in Extrusionsrichtung
76     EXTOPT ,ACLEAR ,1       ! Flächenmesh aus dem extrudiert wird nachher löschen
77
78     VEXT ,ALL , , , , -tp    ! Plattenvolumen extrudieren
79
80     ALLSEL

```

A.6. post_veff_6.inp

```

1  ALLSEL
2
3  CSYS ,1
4  NSEL ,S,LOC ,X,0 ,mrad/2
5  NSEL ,R,LOC ,Z,-tp/2,-tp
6  ESLN ,S,1
7  CM ,Lastkugelbereich ,ELEM
8
9  CSYS ,200
10 NSEL ,S,LOC ,X,0 ,mrad/2
11 NSEL ,R,LOC ,Z,0,-tp/2
12 ESLN ,S,1
13 CM ,Lagerkugelbereich ,ELEM
14
15 CSYS ,0
16 ALLSEL
17 CMSEL ,U,Lastkugelbereich
18 CMSEL ,U,Lagerkugelbereich
19 CM ,Elemente_Veff ,ELEM
20
21 CSYS ,1
22 NSLE ,S
23 NSEL ,R,LOC ,X,rd
24 CM ,Mantel ,NODE
25
26 CSYS ,0
27 CMSEL ,S,Elemente_Veff
28 NSLE ,S
29 NSEL ,R,LOC ,Z,0
30 CM ,Zugflaeche ,NODE
31
32 CMSEL ,S,Elemente_Veff
33 NSLE ,S
34 NSEL ,R,LOC ,Z,-tp
35 CM ,Druckflaeche ,NODE
36
37 CMSEL ,S,Zugflaeche
38 CMSEL ,A,Mantel
39 CMSEL ,A,Druckflaeche
40 CM ,Knoten_Seif ,NODE

```

A.7. post_veff_full.inp

```

1  ALLSEL
2
3  CSYS ,1
4  NSEL ,S,LOC ,X,0 ,mrad/2
5  NSEL ,R,LOC ,Z,-tp/2,-tp
6  ESLN ,S,1
7  CM ,Lastkugelbereich ,ELEM
8
9  CSYS ,200
10 NSEL ,S,LOC ,X,0 ,mrad/2
11 NSEL ,R,LOC ,Z,0,-tp/2
12 ESLN ,S,1
13 CM ,Lagerkugelbereich1 ,ELEM

```

```

14
15 CSYS,300
16 NSEL,S,LOC,X,0,mrad/2
17 NSEL,R,LOC,Z,0,-tp/2
18 ESLN,S,1
19 CM,Lagerkugelbereich2,ELEM
20
21 CSYS,400
22 NSEL,S,LOC,X,0,mrad/2
23 NSEL,R,LOC,Z,0,-tp/2
24 ESLN,S,1
25 CM,Lagerkugelbereich3,ELEM
26
27 CSYS,0
28 ALLSEL
29 CMSEL,U,Lastkugelbereich
30 CMSEL,U,Lagerkugelbereich1
31 CMSEL,U,Lagerkugelbereich2
32 CMSEL,U,Lagerkugelbereich3
33 CM,Elemente_Veff,ELEM
34
35 ALLSEL
36 NSEL,S,LOC,X,xp0
37 NSEL,A,LOC,X,xp1
38 NSEL,A,LOC,Y,yp0
39 NSEL,A,LOC,Y,yp1
40 CM,Mantel,NODE
41
42
43 CMSEL,S,Elemente_Veff
44 NSLE,S
45 NSEL,R,LOC,Z,0
46 CM,Zugflaeche,NODE
47
48 CMSEL,S,Elemente_Veff
49 NSLE,S
50 NSEL,R,LOC,Z,-tp
51 CM,Druckflaeche,NODE
52
53 CMSEL,S,Zugflaeche
54 CMSEL,A,Mantel
55 CMSEL,A,Druckflaeche
56 CM,Knoten_Seff,NODE

```

A.8. write_result.inp

```

1  !#
2  !# vwrite_info.inp, dme, 2015-08
3  !# Subroutine zum ausgeben allgem. Informationen eines Simulationsdurchlaufs
4  !#
5  !# ACHTUNG: Muss umgeben von *CFOPEN und *CFCLOSE aufgerufen werden!
6  !#           Vor Einbau in Abläufe sind die Kopfzeile und die Ausgabeliste zu überprü-
7  !#           fen und anzupassen
8  !#
9  !# Abhängigkeiten: x_readtime_i.mac
10 !#
11 x_readtime_i,0
12
13 *VWRITE, 'SYSTEM: ',systemid,', System-PROCS: ', nprocmax      ! C-formatting
14 %-8C %-12C %19C %I
15 *VWRITE, ansysrev
16 ('ANSYS-Rev.: ',F4.1)
17 *IF,configfileexist,EQ,1,THEN
18     contentconfig=STRCAT(npr_str0(1),npr_str0(2))
19     *VWRITE, 'config-file exists:',strconfig,'.ans',contentconfig
20 %20C %10C%4C %16C
21 *ELSE
22     *VWRITE, 'config-file:', 'no'
23 %12C %2C
24 *ENDIF
25

```

```

26 !# Kopfzeile:
27 *VWRITE,'Elemente','Nodes','Dauer','Sigma_ZUG','Kugel-d','Laenge','Breite','Hoehe','
    RD','dx','dy','E-Platte','v-Platte','Kraft','Serie'
28 (15(TR3,A13))
29 *VWRITE,'[-] ','[-] ','[s] ','[N/mm2] ','[mm] ','[mm] ','[mm] ','[mm] ','[mm] ','[mm] ','[mm]
    ','[mm] ','[N/mm2] ','[-] ','[N] ','[-]'
30 (15(TR8,A8))
31
32
33 /eof
34
35 :P_write
36
37 *GET,nelem,ELEM,0,COUNT
38 *GET,nnode,NODE,0,COUNT
39
40 *GET,wallt1,ACTIVE,,TIME,WALL
41 deltwlls=(wallt1-wallt0)*3600
42 wallt0=wallt1
43
44 FSUM
45 *GET, freact ,FSUM,,ITEM,FY
46
47 !# Ausgabeliste:
48 *VWRITE,nelem,nnode,deltwlls,smax,db,lp,bp,tp,rd,dx,dy,ep,np,fb,serienname
49 %16.8f %16.8f
    %16.8f %16.8f %12C

```

A.9. x_readtime_i.mac

```

1 ! x_readtime_i.mac, phs, 15.05.2009
2 ! reads actual CPU- and WALL time, calc. delta time
3 ! Aufruf: *USE,'x_readtime_i.mac', index_
4 ! Input:
5 ! ARG1: index_ : Index der aktuellen time-Messung
6 !
7 ! --- programming history -----
8 ! 17.02.2007: Programmierung (Kopie von readtime1.inp), Adaption
9 ! 27.02.2007: Erweiterung, case arg1=0
10 ! 11.01.2009: nprocreq
11 ! 06.04.2009: Calc. integral time (updated at every call)
12 ! 15.05.2009: globaler Output_parameter: x_readtime_num
13 !
14
15 index_ = arg1 ! lokale Par: parameter names with trailing underbars
16 x_readtime_num = index_ ! aktuelle Indexnummer: global gespeichert
17
18 *IF, index_, EQ, 0, THEN
19 *DEL, tim_str0(1),,NOPR $ *DIM,tim_str0,CHAR,4 ! Def. des String-
    arrays
20 *GET,systemid,ACTIVE,,SYNAME ! ask for platform (SYNAME is Char*12, see syspar.inc)
21
22 *GET,nprocmax,ACTIVE,0,NPROC, ! getting max PROC number
23 *GET,nprocreq,COMMON,,anscom,,INT,11 ! number of processors to use (if available)
24 *GET,ansysrev,ACTIVE,0,REV, ! getting actual ANSYS Rev.
25
26 ansysrevmain=NINT(ansysrev)
27 ansysrevmin=(ansysrev-NINT(ansysrev))*10
28 str1conf =STRCAT('config',CHRVAL(ansysrevmain))
29 strconfig=STRCAT(str1conf,CHRVAL(ansysrevmin))
30
31 ! checking existence of config-file: configx.ans (variabel)
32 /INQUIRE,configfileexist,EXIST,%strconfig%,ans
33 ! *IF,configfileexist,EQ,1,THEN
34 ! *DEL,npr_str0(1),,NOPR $ *DIM,npr_str0,CHAR,2 ! Def. des
    String-arrays
35 ! *VREAD, npr_str0(1), %strconfig%,ans
36 ! (2A)
37 ! *ENDIF
38
39 *IF, systemid, EQ, 'WINDOWS 64', OR, systemid, EQ, 'WINDOWS x64', THEN !
    Win64-Platforms

```

```

40 /SYS, time /T > ansys_time0.txt
41 /SYS, date /T > ansys_date0.txt
42 *VREAD, tim_str0(1), ansys_time0,txt
43 (4A)
44 *VREAD, tim_str0(2), ansys_date0,txt
45 (3A)
46 *ELSEIF, systemid, EQ, 'INTEL NT', THEN ! Win32-Platforms
47 /SYS, time /T > ansys_time0.txt
48 /SYS, date /T > ansys_date0.txt
49 *VREAD, tim_str0(1), ansys_time0,txt
50 (4A)
51 *VREAD, tim_str0(2), ansys_date0,txt
52 (3A)
53 *ELSE ! for unix-platforms: 'SGI64','LINUX IA32','LINUX IA64'
54 /SYS, date > ansys_timedate0.txt
55 *VREAD, tim_str0(1), ansys_timedate0,txt
56 (4A)
57 /COM, ansys_timedate0.txt read
58 *ENDIF
59
60 /OUTPUT,dummy_config.txt $ /CONFIG,STAT $ /OUT ! writing /
    CONFIG status to file
61
62 *ENDIF
63
64 *GET,wallt%index%,ACTIVE,,TIME,WALL ! reading actual wall time (no reset at
    midnight for running job) [h]
65 *GET,cput%index%, ACTIVE,,TIME,CPU ! reading actual cpu time since Program start
    [s]
66
67 *IF, index_,GT,0,THEN
68 deltwllh%index%=wallt%index%-wallt%index_-1% ! [hr]
69 deltwlls%index%=(deltwllh%index%)*3600 ! [s]
70 deltcpus%index%=(cput%index%-cput%index_-1%) ! [s]
71 deltcpuh%index%=(deltcpus%index%)/3600 ! [hr]
72
73 ! integral time
74 deltwllh=wallt%index%-wallt0 ! [hr]
75 deltwlls=deltwllh*3600 ! [s]
76 deltcpus=(cput%index%-cput0) ! [s]
77 deltcpuh=deltcpus/3600 ! [hr]
78 *ENDIF

```

A.10. x_Effektives_Volumen.mac

```

1 !#
2 !# x_Effektives_Volumen.mac
3 !#
4 !# von: dme
5 !# am: 28.01.2016
6 !#
7 !# Wirkung: vorbereiten von Element und Node Tabellen für
8 !# das berechnen des effektiven Volumens durch
9 !# ein externes Fortran-Program.
10 !#
11 !# Achtung: um dieses Macro laufen lassen zu können
12 !# müssen ZWEI COMPONENTS definiert sein:
13 !# Elemente_Veff - ein Element - Component
14 !# der Gültigen Bereiche des Modells
15 !#
16 !# Knoten_Seff - ein Node - Component der
17 !# Gültigen Bereiche des der Oberfläche
18 !#
19 !# Des weiteren sollten nur Elemente einer Geometrie (siehe ELTYPE)
20 !# auf einmal berechnet werden. Für Modelle mit Teträdern und Hexaedern
21 !# bzw. Vierecken und Dreiecken sollte die Berechnung seperat mit beiden
22 !# Elementtypen durchgeführt werden. Die Ergebnistabellen können später
23 !# addiert werden.
24 !#
25 !# INPUT:
26 !# *USE,'Effektives_Volumen.mac', FNAME, SMAX, GAUß_FAKT ,MMAX, SYMM, ELTYPE, BREITE
27 !#

```

```

28  !# FNAME      :=      Name des Ausgabefiles (default: 'effVol')
29  !#
30  !#
31  !# SMAX       :=      Vergleichsspannung (kein default - muss gegeben werden!)
32  !#
33  !#
34  !#
35  !#
36  !#
37  !#
38  !# GAUß_FAKT:=      Grad der Gauß-Legendre Quadratur (default: 3)
39  !#
40  !#
41  !#
42  !# MMAX       :=      Weibull-Modul bis zu dem das Veff berechnet werden soll (default:
43  30)
44  !#
45  !#
46  !# SYMM       :=      Symmetriefaktor (default: 1)
47  !#
48  !#
49  !# ELTYPE     :=      Elementtyp des FEM Modells (default: 'HExaeDER')
50  !#
51  !#
52  !#
53  !#
54  !#
55  !# BREITE     :=      bei 2D Elementen die Breite der Geometrie (default: 0)
56  !#
57  !#
58  !#
59  !#
60  !#
61  !# CHANGELOG:
62  !#
63  !# 10.02. real-to-test für ARG6, Umordnen parameterfile
64  !#
65  !# 12.02. umstellung auf 1 NODE-file
66  !#
67  !# 20.02. Zusatz: PIA-OPTION
68  !#
69  !# 03.03. PIA wurde fix in Fortran eingebaut, gibt jetzt IMMER aus. Option wieder
70  !#
71  !# 15.03. Element- und Face-Tabellen werden auf die Selektierten Elemente/Flächen
72  !#
73  !#
74  !#####
75  !#####
76  *IF, '%ARG1%', EQ, '0', THEN
77      Veff_ausgabename='effVol'
78  *ELSE
79      Veff_ausgabename='%ARG1%'
80  *ENDIF
81
82  *IF, ARG3, LT, 2, OR, ARG3, GT, 9, THEN
83      Veff_gauss_faktor=3
84  *ELSE
85      Veff_gauss_faktor=NINT(ARG3)
86  *ENDIF
87
88  *IF, ARG4, EQ, 0, THEN
89      Veff_m_max=30
90  *ELSEIF, ARG4, LT, 0, THEN
91      Veff_m_max=0
92  *ELSE
93      Veff_m_max=NINT(ARG4)
94  *ENDIF
95
96  *IF, ARG5, LT, 1, THEN

```

```

97     Veff_symm_fakt=1
98 *ELSE
99     Veff_symm_fakt=NINT(ARG5)
100 *ENDIF
101
102 *IF, '%ARG6%', EQ, 'VIERECK', OR, '%ARG6%', EQ, '24', THEN
103     Veff_eckpunkte=4      ! Anzahl der Eckpunkte eines jeden Elements
104     Seff_facepoints=2    ! Anzahl der Punkte die eine Elementfläche ausmachen
105     Seff_elemfaces= 4    ! Anzahl der Elementflächen pro Element
106     Veff_dim=24
107 *ELSEIF, '%ARG6%', EQ, 'TETRÄDER', OR, '%ARG6%', EQ, '33', THEN
108     Veff_eckpunkte=4
109     Seff_facepoints=3
110     Seff_elemfaces= 4
111     Veff_dim=33
112 *ELSEIF, '%ARG6%', EQ, 'DREIECK', OR, '%ARG6%', EQ, '23', THEN
113     Veff_eckpunkte=3
114     Seff_facepoints=2
115     Seff_elemfaces= 3
116     Veff_dim=23
117 *ELSE
118     Veff_eckpunkte=8
119     Seff_facepoints=4
120     Seff_elemfaces= 6
121     Veff_dim=34
122 *ENDIF
123
124 *IF, ARG7, LT, 0, THEN
125     Veff_breite=-1
126 *ELSE
127     Veff_breite=ARG7
128 *ENDIF
129
130
131 ALLSEL
132
133 *GET, Veff_maxelem, ELEM, , NUM, MAX           ! Gibt max. Elementnummer
134 *GET, Veff_maxnode, NODE, , NUM, MAX         ! Gibt max. Knotennummer
135
136 CMSEL, S, Elemente_Veff                     ! Das zu berechnende Volumen wählen
137 NSLE, S, CORNER                             ! Die zu den gewählten Elementen gehörende
    Eckennodes
138
139 *GET, Veff_nesel, ELEM, , COUNT              ! Gibt Anzahl der selektierten Elemente
140
141 ! Definieren einer binären Liste die für jedes selektierte Element 1, sonst 0 ist
142 *DEL, Veff_E_maske(1), , NOPR $ *DIM, Veff_E_maske, , Veff_maxelem
143 ! Definieren eines Arrays für die Ecknodes jedes Elements bis zu höchsten
    Elementnummer
144 *DEL, Veff_Elemente(1,1), , NOPR $ *DIM, Veff_Elemente, , Veff_maxelem, Veff_eckpunkte
145
146 *VGET, Veff_E_maske(1), ELEM, 1, ESEL        ! Alle ESEL-Werte auf Array
147 *DO, node_, 1, Veff_eckpunkte                ! Nodes 1 - 'eckpunkte' auslesen (nur die
    Ecken-Nodes, an denen die Spannungen gespeichert werden, alle anderen nodes
    interessieren uns nicht!)
148     *VMASK, Veff_E_maske(1) $ *VGET, Veff_Elemente(1, node_), ELEM, 1, NODE, node_
149 *ENDDO
150
151 ! Definieren eines Arrays selektierter Elemente (also keine leeren oder mit 0 belegten
    Zeilen)
152 *DEL, Veff_SelElem(1,1), , NOPR $ *DIM, Veff_SelElem, , Veff_nesel, Veff_eckpunkte
153
154 ! selem_ zählt die Zeilen der selektierten Elemente, nelem_ die globalen.
155 ! wenn immer eine Zeile der globalen Elementliste nicht Null enthält,
156 ! wird diese zum Array der selektierten Elemente angefügt
157 selem_=0
158 *DO, nelem_, 1, Veff_maxelem
159     *IF, Veff_Elemente(nelem_, 1), NE, 0, THEN
160         selem_=selem_+1
161         *DO, node_, 1, Veff_eckpunkte
162             Veff_SelElem(selem_, node_)=Veff_Elemente(nelem_, node_)
163         *ENDDO
164 *ENDIF

```

```

165 *ENDDO
166
167
168
169 CMSEL,S,Knoten_Seiff           ! Zu berechnende Oberfläche auswählen
170 ESLN,S                          ! Die zu der Oberfläche gehörenden Elemente
171 NSLE,S,CORNER                  ! Die zu den Elementen gehörenden Eckennodes
172 CMSEL,R,Knoten_Seiff         ! Von diesen Eckennodes wieder nur die, die
    zur Oberfläche gehören
173
174 *GET,Seiff_maxelem,ELEM,,NUM,MAX      ! Gibt max. Elementnummer
175 Seiff_maxfaces=Seiff_maxelem*Seiff_elemfaces
176
177 *DEL,Seiff_Faces(1,1),,NOPR      $ *DIM,Seiff_Faces,,Seiff_maxfaces,Seiff_facepoints
178
179 *DO,el_,1,Seiff_maxelem          ! Elementweises auslesen der ersten selektierten
    Fläche
180 maske_ES=ESEL(el_)
181 *IF,maske_ES,EQ,1,THEN          ! Abfrage ob Element selektiert
182 *DO,face_,1,Seiff_elemfaces     ! Nummer der Elementfläche, zu der die gewählten nodes
    gehören
183     ! Nur Wenn alle vier Punkte an der selektierten Fläche liegen, soll diese
        miteinbezogen werden:
184     sel_=1
185     *DO,node_,1,Seiff_facepoints
186     ! NSEL(n) ist 1 wenn selektiert, 0 wenn undefiniert und -1 wenn unselektiert,
187     ! daher die (x>y) Funktion die den jeweils größeren Wert nimmt (hier 1 oder 0).
188     ! NDFACE(e,f,n) gibt die globale Nodenummer des n-ten Punktes des f-ten Face
        des e-ten Elements
189     ! Nur wenn ALLE NSEL Aufrufe 1 ergeben bleibt auch sel_ 1 --> nur vollständig
        selektierte Faces werden aufgerufen
190     sel_=sel_*(NSEL(NDFACE(el_,face_,node_))>0)
191     *ENDDO
192
193     *IF,sel_,GT,0,THEN
194     *DO,node_,1,Seiff_facepoints
195     Seiff_Faces((el_-1)*Seiff_elemfaces+face_,node_)=NDFACE(el_,face_,node_) !
        Eintragen der Nodes in Folge in Array
196
197     *ENDDO
198     *ENDIF
199 *ENDDO
200 *ENDIF
201
202 Seiff_nfaces=0
203 *DO,face_,1,Seiff_maxfaces
204     *IF,Seiff_Faces(face_,1),NE,0,THEN ! Zeilenzählen die nicht Null sind. Es gibt
        keinen *GET Befehl der die Anzahl der selektierten Flächen ausgibt
205     Seiff_nfaces=Seiff_nfaces+1
206     *ENDIF
207 *ENDDO
208
209 *DEL,Seiff_SelFaces(1,1),,NOPR      $ *DIM,Seiff_SelFaces,,Seiff_nfaces,
    Seiff_facepoints
210
211 selelem_=0
212 *DO,nelem_,1,Seiff_maxfaces
213     *IF,Seiff_Faces(nelem_,1),NE,0,THEN
214     selelem_=selelem_+1
215     *DO,node_,1,Seiff_facepoints
216     Seiff_SelFaces(selelem_,node_)=Seiff_Faces(nelem_,node_)
217     *ENDDO
218     *ENDIF
219 *ENDDO
220
221
222
223 ALLSEL
224 ESEL,S,ELEM,,1,Seiff_maxelem
225 NSLE,S,CORNER                  ! Die zu den gewählten Elementen gehörende
    Eckennodes
226
227

```

```

228 *DEL ,Veff_maske_N(1) , ,NOPR      $ *DIM ,Veff_maske_N , ,Veff_maxnode
229 *DEL ,Veff_Nodes_Gesamt(1,1) , ,NOPR $ *DIM ,Veff_Nodes_Gesamt , ,Veff_maxnode ,6
230 *VGET ,Veff_maske_N(1) ,NODE ,1 ,NSEL      ! Alle Koordinaten und Hauptspannungen der
      gewählten Eckennodes auf Array
231 *VMASK ,Veff_maske_N(1)              $ *VGET ,Veff_Nodes_Gesamt(1,1) ,NODE ,1 ,LOC ,X
232 *VMASK ,Veff_maske_N(1)              $ *VGET ,Veff_Nodes_Gesamt(1,2) ,NODE ,1 ,LOC ,Y
233 *VMASK ,Veff_maske_N(1)              $ *VGET ,Veff_Nodes_Gesamt(1,3) ,NODE ,1 ,LOC ,Z
234 *VMASK ,Veff_maske_N(1)              $ *VGET ,Veff_Nodes_Gesamt(1,4) ,NODE ,1 ,S ,1
235 *VMASK ,Veff_maske_N(1)              $ *VGET ,Veff_Nodes_Gesamt(1,5) ,NODE ,1 ,S ,2
236 *VMASK ,Veff_maske_N(1)              $ *VGET ,Veff_Nodes_Gesamt(1,6) ,NODE ,1 ,S ,3
237
238 ALLSEL
239
240 *IF ,Veff_dim ,EQ ,34 ,THEN
241 *MWRITE ,Veff_SelElem(1,1) ,'effVol_Elemente' ,out      ! 8 Eck-knoten
242 (F17.0 ,T17 ,F17.0 ,T33 ,F17.0 ,T49 ,F17.0 ,T65 ,F17.0 ,T81 ,F17.0 ,T97 ,F17.0 ,T113 ,F17.0 ,T129 ,' '
      )
243
244 *MWRITE ,Seff_SelFaces(1,1) ,'effVol_Faces' ,out      ! 4 Flächen-knoten
245 (F17.0 ,T17 ,F17.0 ,T33 ,F17.0 ,T49 ,F17.0 ,T65 ,' ')
246
247 *ELSEIF ,Veff_dim ,EQ ,33 ,THEN
248 *MWRITE ,Veff_SelElem(1,1) ,'effVol_Elemente' ,out      ! 4 Eck-knoten
249 (F17.0 ,T17 ,F17.0 ,T33 ,F17.0 ,T49 ,F17.0 ,T65 ,' ')
250
251 *MWRITE ,Seff_SelFaces(1,1) ,'effVol_Faces' ,out      ! 3 Flächenknoten
252 (F17.0 ,T17 ,F17.0 ,T33 ,F17.0 ,T49 ,' ')
253
254 *ELSEIF ,Veff_dim ,EQ ,24 ,THEN
255 *MWRITE ,Veff_SelElem(1,1) ,'effVol_Elemente' ,out      ! 4 Eck-knoten
256 (F17.0 ,T17 ,F17.0 ,T33 ,F17.0 ,T49 ,F17.0 ,T65 ,' ')
257
258 *MWRITE ,Seff_SelFaces(1,1) ,'effVol_Faces' ,out      ! 2 Linienknoten
259 (F17.0 ,T17 ,F17.0 ,T33 ,' ')
260
261 *ELSE
262 *MWRITE ,Veff_SelElem(1,1) ,'effVol_Elemente' ,out      ! 3 Eckknoten
263 (F17.0 ,T17 ,F17.0 ,T33 ,F17.0 ,T49 ,' ')
264
265 *MWRITE ,Seff_SelFaces(1,1) ,'effVol_Faces' ,out      ! 2 Linienknoten
266 (F17.0 ,T17 ,F17.0 ,T33 ,' ')
267 *ENDIF
268
269 *MWRITE ,Veff_Nodes_Gesamt(1,1) ,'effVol_Nodes' ,out
270 (6E16.8)
271
272
273 *CFOPEN ,'effVol_Parameter' ,out ,
274 *VWRITE ,'NAME: ' ,Veff_ausgabename
275 %-8c %32c
276 ! Lange Strings müssen mit C-Formatierung definiert werden - APDL-Fortran lässt nur
      max. 8 Char zu!
277 *VWRITE ,'smax: ' ,ARG2
278 (A8 ,ES16.8)
279 *VWRITE ,'Gauss: ' ,Veff_gauss_faktor
280 (A8 ,F17.0 ,T25 ,' ')
281 *VWRITE ,'Mmax: ' ,Veff_m_max
282 (A8 ,F17.0 ,T25 ,' ')
283 *VWRITE ,'Symm: ' ,Veff_symm_fakt
284 (A8 ,F17.0 ,T25 ,' ')
285 *VWRITE ,'Elem: ' ,Veff_dim
286 (A8 ,F17.0 ,T25 ,' ')
287 *VWRITE ,'Breite: ' ,Veff_breite
288 (A8 ,F16.8)
289 *CFCLOSE
290
291 !*GET ,OS_name ,ACTIVE , ,SYNAME
292 !
293 !*IF , OS_name , EQ , 'WINDOWS 64' , OR , OS_name , EQ , 'WINDOWS x64' , THEN      ! Win64-
      Platforms
294 ! /SYS ,./Effektives_Volumen.exe
295 !*ELSEIF , OS_name , EQ , 'INTEL NT' , THEN      ! Win32-Platforms
296 ! /SYS ,./Effektives_Volumen.exe

```

```
297 !*ELSE ! for unix-platforms: 'SGI64', 'LINUX IA32', 'LINUX IA64'
298 ! /SYS, ./Effektives_Volumen
299 !*ENDIF
300
301 ! Auch den Linux Build 'Effektives_Volumen.exe' nennen, dann spart man sich das ganze
Nach-dem-OS-fragen
302 /SYS, ./Effektives_Volumen.exe
```

B.Fortran Syntax

B.1. Effektives_Volumen.f90

```
1 PROGRAM effektivesVol
2 !
3 ! Berechnung des effektiven Volumens in ANSYS-FEM
4 ! ACHTUNG! Benötigt das ANSYS Macro "x_effektives_Volmen_3D.mac"
5 !
6 ! 06.04.2016
7 !
8 ! Abhaegigkeiten:
9 !   - overhead.f90
10 !
11 ! Vorsicht: Der Prozent-zaehler im Hauptteil verwendet eine Formatsynatx die nicht von
12 !           allen Compilern unterstuetzt wird!
13 !           Eine Alternative Schreibweise wurde als Kommentar zugefuegt, falls diese
14 !           auch nicht funktioniert, ist es am
15 !           einfachsten, den Prozentzaehler voellig auszukommentieren.
16 !
17 ! Versionen:
18 !
19 ! 1.0 : 24.02.2016 - Erste Lauffaehige Version
20 !
21 ! 1.1 : 03.03.2016 - PIA-Option geloeschst - berechnet jetzt immer beide (als Array!)
22 !
23 ! 2.0 : 09.03.2016 - von elementweiser Berechnung auf Arrayoperationen umgestiegen
24 !                   Viel speicherintensiver, dafuer _erheblich_ schneller!
25 !
26 ! 2.1 : 14.03.2016 - Fehlerbehebung in Gram'scher Determinante - Seff ist jetzt
27 !                   richtig!
28 !
29 ! 2.2 : 15.03.2016 - APDL Macro gibt nur mehr gewaehlte Elemente in listen aus
30 !                   face- und element-Arrays daher jetzt verschieden lang!
31 !                   Einlesen angepasst
32 !
33 ! 06.04.2016 - Weitere Kommentare zum Besseren Verstaendniss von Variablen und
34 !                   deren Bedeutungen
35 !
36 ! 14.04.2016 - Umlaute entfernt
37 !
38 USE overhead
39
40 IMPLICIT none
41
42 real, dimension (:,:), allocatable :: veff, seff
43 ! 2D Arrays fuer effektives Volumen und Oberflaeche. Laenge wird die Anzahl an
44 ! Weibull modulen, Tiefe wird 2. Also pro Modul einmal PIA-Kriterium und
45 ! einmal Hauptspannung.
46
47 logical, dimension (:), allocatable :: elem_nodes, surf_nodes
48 ! Masken Arrays fuer zum Sicherstellen der Skalierungsspannung
49
50 integer :: m,e, i, j, symm_fakt, element_type, m_max, error=0
51 ! Zaehler Variablen und Programm Parameter (siehe Einlesen unten)
52
53 integer :: nele, nface, nnode, spalten_elem, spalten_face ! Arraymasze. Werden
54 ! erst bestimmt, dann zugewiesen (siehe Einlesen unten)
55
56 character (len=256) :: ausgabenname
57 ! Name der Output-Files
58
59 character (len=10) :: error_num1, error_num2
60 ! Hilfsvariablen zum Ausgeben von Fehlermeldungen
61
62 character (len=32) :: error_loc
63 ! Hilfsvariable zum Ausgeben von Fehlermeldungen
64
65 character (len=256) :: error_msg
66 ! String zum Ausgeben von Fehlermeldungen
67
68 real, parameter :: pi=3.14159265358979
69
70 real :: smax_file, smax_elem, smax_surf, breite_2D, rad
71 ! Maximalspannungen zum (aus Parameter-file, Element-Liste und Oberflaechen-
72 ! Liste) Vergleich, Programm Parameter (siehe Einlesen unten)
73
74
75
76
77
78
```

```

59
60 !
#####

61 ! Vorbereitung: Einlesen und allokkieren von Arrays
62 !
#####

63
64 ! Einlesen des Parameter-Files
65
66 OPEN (101, FILE='effVol_Parameter.out', STATUS='old')
67 READ(101,'(8X,A256)') ausgabenname ! Name des Output-Files
68 READ(101,'(8X,ES16.8)') smax_file ! Bekannt Maximalspannung (zum ueberpruefen,
    Ob keine Singularitaeten sich eingeschlichen haben)
69 READ(101,'(8X,I16)') gausz_fakt ! Grad der Gausz-Legendre Quadratur
70 READ(101,'(8X,I16)') m_max ! Weibull Modul bis zu dem berechnet wird
71 READ(101,'(8X,I16)') symm_fakt ! Symmetriefaktor fuer gespiegelte Modelle,
    etc.
72 READ(101,'(8X,I16)') element_type ! Idikator fuer Elementgeometrie
73 READ(101,'(8X,F16.8)') breite_2D ! Breite von 2D Elementen, falls zutreffend
74 CLOSE(101)!, STATUS='delete')
75
76 ausgabenname = ADJUSTL(ausgabenname)
77
78 ! DEBUGGING
79 WRITE (*,'(A8,ES16.8)') 'SMAX = ', smax_file
80 WRITE (*,'(A8,I16)') 'GAUSS = ', gausz_fakt
81 WRITE (*,'(A8,I16)') 'MMAX = ', m_max
82 WRITE (*,'(A8,I16)') 'SYMM = ', symm_fakt
83 WRITE (*,'(A8,I16)') 'GEOM = ', element_type
84 WRITE (*,'(A8,F16.8)') 'BREIT = ', breite_2D
85
86 arraybreiten_zuweisung: SELECT CASE (element_type) ! die Anzahl an Nodes pro Element
    haengt von der bekannten Elementgeometrie ab.
87 CASE (23)
88     spalten_elem = 4!3 ! Nodes fuer Element-volumen
89     spalten_face = 2 ! Fodes fuer Element-flaechen
90 CASE (24)
91     spalten_elem = 4
92     spalten_face = 2
93 CASE (33)
94     spalten_elem = 8!4 - Hier wird statt reinen Tetraedern auf degeneierte
        Hexaeder vorbereitet!
95     spalten_face = 4!3
96 CASE (34)
97     spalten_elem = 8
98     spalten_face = 4
99 CASE DEFAULT
100     error_loc='arraybreiten_zuweisung'
101     error_msg='Fehler beim Zuweisen der Arraybreiten. ueberpruefen Sie die
        Elementtype im Parameter-File.'
102     CALL exit_error(error_loc,error_msg)
103     END SELECT arraybreiten_zuweisung
104
105 !DEBUGGING
106 WRITE (*,'(A15,I1)') 'spalten_elem = ', spalten_elem
107 WRITE (*,'(A15,I1)') 'spalten_face = ', spalten_face
108 PRINT *, 'Reading Element-File:'
109
110 ! File-laenge lesen, Anzahl Reihen bestimmen
111 OPEN (102, FILE='effVol_Elemente.out', STATUS='old')
112 nelelem = 0
113 DO ! Einzelnes Abzaehlen des Element-Files
114     READ(102,*, IOSTAT = error)
115     IF (error < 0) EXIT
116     nelelem = nelelem + 1
117 END DO
118 REWIND(102)
119
120 !DEBUGGING
121 WRITE (*,'(A28,I10)') '.....nelelem = ', nelelem
122 PRINT *, 'Allocating Elements'

```

```

123
124     ! Element- und Flaechen-Node-Arrays allokkieren
125     ALLOCATE (elements(spalten_elem,nelem))
126
127     !DEBUGGING
128     PRINT *, '..... Check'
129     PRINT *, 'Filling Element-Array'
130
131     READ(102,*) elements     ! Geht nur so einfach aufgrund der Umkehrung Spalten/
                             ! Zeilen in Element- und Face-Array. Bei (nelem,spalten_e) braucht man eine
                             ! implizite Schleife (faces(i,j),j=1,spalten_face) zum einlesen!
132     CLOSE(102)!, STATUS='delete')
133
134     !DEBUGGING
135     PRINT *, '..... Done'
136     PRINT *, 'Reading Face-File'
137
138
139     OPEN (103, FILE='effVol_Faces.out', STATUS='old')
140     nface = 0
141     DO ! Einzelnes Abzaehlen des Element-Files
142         READ(103,*, IOSTAT = error)
143         IF (error < 0) EXIT
144         nface = nface + 1
145     END DO
146     REWIND(103)
147
148     !DEBUGGING
149     WRITE (*,'(A28,I10)') '.....nface = ', nface
150     PRINT *, 'Allocating Faces'
151
152     ! Element- und Flaechen-Node-Arrays allokkieren
153     ALLOCATE (faces(spalten_face,nface))
154
155     !DEBUGGING
156     PRINT *, '..... Check'
157     PRINT *, 'Filling Face-Array'
158
159     READ(103,*) faces     ! Siehe einlesen elements, beachte die umgedrehten Indizes der
                             ! Arrays!
160     CLOSE(103) !, STATUS='delete')
161
162     !DEBUGGING
163     PRINT *, '..... Done'
164     PRINT *, 'Checking for Tets, Tris, rearranging'
165
166     SELECT CASE (element_type) ! reine Tetraeder und Dreiecke auf degenerierte
                             ! Hexaeder und Vierecke umwandeln. Zum besseren Verstaendis wird hier auf die
                             ! Element-Library der Ansys-Hilfe verwiesen: help/ans_elem/Hlp_E_SOLID186.html
167     CASE (23)
168         elements(4,:) = elements(3,:)
169         element_type=24
170     CASE (33)
171         elements(8,:) = elements(4,:)
172         elements(7,:) = elements(4,:)
173         elements(6,:) = elements(4,:)
174         elements(5,:) = elements(4,:)
175         elements(4,:) = elements(3,:)
176         faces(4,:) = faces(3,:)
177         element_type=34
178     END SELECT
179
180     !DEBUGGING
181     PRINT *, '..... Check'
182     PRINT *, 'Node-File Zeilen zaehlen'
183
184     ! Node-File einlesen, Zeilen zaehlen
185     OPEN (104, FILE='effVol_Nodes.out', STATUS='old')
186     nnode = 0
187     DO ! Einzelnes Abzaehlen des Node-Files
188         READ(104,*, IOSTAT = error)
189         IF (error < 0) EXIT ! Bei Lesefehler (= end-of-file) wird das Zeilen-zaehlen
                             ! abgebrochen

```

```

190     nnode = nnode + 1
191     END DO
192     REWIND(104)
193
194     !DEBUGGING
195     WRITE (*,'(A28,I10)') '.....nnode = ', nnode
196     PRINT *, 'Reading Node-File'
197
198     ! Node-Array-Groeszen zuteilen. Nodefile ist 6 Elemente tief, da es 3 Koordinaten
      und 3 Hauptspannungen gibt
199     ALLOCATE (nodes(nnode,6))
200
201     DO i=1,nnode
202         READ(104,*) (nodes(i,j),j=1,6)
203     END DO
204
205     CLOSE(104)!, STATUS='delete')
206
207     !DEBUGGING
208     PRINT *, '..... Done'
209     PRINT *, 'Priming node mask Arrays'
210
211     ! Die Maximalspannungen der Elemente bzw. Faces mit der gegebenen smax vergleichen
212
212     ALLOCATE (elem_nodes(nnode), surf_nodes(nnode))
213     elem_nodes = .FALSE.
214     surf_nodes = .FALSE.
215
216     !DEBUGGING
217     PRINT *, '..... Check'
218     PRINT *, 'Element SigMax'
219
219     singularity_check: IF (smax_file>0.) THEN
220
221     loop_smax_el1: DO i=1,spalten_elem
222     loop_smax_el2: DO j=1,nelem
223
224         ! Erstellen einer Node-Maske fuer alle zutreffenden Nodes
225         IF (elements(i,j)/=0) elem_nodes(elements(i,j))=.TRUE.
226
227     END DO loop_smax_el2
228     END DO loop_smax_el1
229
230     !Zuweisung gefundener Maximalhauptspannung
231     smax_elem=MAXVAL(nodes(:,4),elem_nodes)
232
233     ! Vergleich der gefundenen mit der gegebenen Maximalspannung. Abbruch bei zu
      grosser Diskrepanz. (inkludierte Singularitaet?)
234     IF ((smax_elem<smax_file*0.95) .OR. (smax_elem>smax_file*1.05)) THEN
235         error_loc='singularity_check'
236         WRITE (error_num1,'(ES10.2)') smax_file
237         WRITE (error_num2,'(ES10.2)') smax_elem
238         error_msg='Die im File gefundene maximale Element-Spannung ('
239         error_msg=TRIM(error_msg)//TRIM(error_num1)
240         error_msg=TRIM(error_msg)//') unterscheidet sich um mehr als 5% von der
      gegebenen Maximalspannung ('
241         error_msg=TRIM(error_msg)//TRIM(error_num2)
242         error_msg=TRIM(error_msg)//'). '
243         CALL exit_error(error_loc,error_msg)
244     END IF
245
246     !DEBUGGING
247     PRINT *, '..... Check'
248     PRINT *, 'Face SigMax'
249
250     loop_smax_sf1: DO i=1,spalten_face
251     loop_smax_sf2: DO j=1,nface
252
253         ! Erstellen einer Node-Maske fuer alle zutreffenden Nodes
254         IF (faces(i,j)/=0) surf_nodes(faces(i,j))=.TRUE.
255
256     END DO loop_smax_sf2
257     END DO loop_smax_sf1
258

```

```

259
260      !Zuweisung gefundener Maximalhauptspannung
261      smax_surf=MAXVAL(nodes(:,4),surf_nodes)
262
263      ! Vergleich der gefundenen mit der gegebenen Maximalspannung. Abbruch bei zu
      ! groeher Diskrepanz. (inkludierte Singularitaet?)
264      IF ((smax_surf<smax_file*0.95) .OR. (smax_surf>smax_file*1.05)) THEN
265          error_loc='singularity_check'
266          WRITE (error_num1,'(ES10.2)') smax_file
267          WRITE (error_num2,'(ES10.2)') smax_surf
268          error_msg='Die im File gefundene maximale Oberflaechen-Spannung ('
269          error_msg=TRIM(error_msg)//TRIM(error_num1)
270          error_msg=TRIM(error_msg)//') unterscheidet sich um mehr als 5% von der
      ! gegebenen Maximalspannung ('
271          error_msg=TRIM(error_msg)//TRIM(error_num2)//'). '
272          CALL exit_error(error_loc,error_msg)
273      END IF
274
275      !DEBUGGING
276      PRINT *, '..... Check'
277
278      ELSE
279
280          DO i=1,spalten_elem
281              DO j=1,nelem
282
283                  ! Erstellen einer Node-Maske fuer alle zutreffenden Nodes
284                  IF (elements(i,j)/=0) elem_nodes(elements(i,j))=.TRUE.
285
286              END DO
287          END DO
288
289      !Zuweisung gefundener Maximalhauptspannung
290      smax_file=MAXVAL(nodes(:,4),elem_nodes)
291
292      !DEBUGGING
293      PRINT *, '..... Used max stress from node-file'
294
295      END IF singularity_check
296
297      !DEBUGGING
298      PRINT *, 'Discarding negative stresses'
299
300      WHERE (nodes(:,4:6)<0) ! Es werden in allen Berechnungen nur positive
      ! Hauptnormalspannungen beruecksichtigt, negative (Druck-)Spannungen werden
      ! schaedigen das Material nicht!
301      nodes(:,4:6)=0.0
302      END WHERE
303
304      !DEBUGGING
305      PRINT *, '..... Check'
306      PRINT *, 'Scaling stresses'
307
308      nodes(:,4:6)=nodes(:,4:6)/smax_file ! Normieren der gesamten Spannungstabelle
309
310      !DEBUGGING
311      PRINT *, '..... Check'
312      PRINT *, 'Allocating Gauss-Arrays'
313
314      ! Gauss-Integrationspunkte zuteilen
315      ALLOCATE (gausz_r(gausz_fakt),gausz_w(gausz_fakt))
316
317      gausz_zuweisung: SELECT CASE (gausz_fakt) ! vorberechnete Gauss-Abszissen und
      ! Gewichte einspeichern
318      CASE (2)
319          gausz_r(1) = -0.5773502691896
320          gausz_r(2) = 0.5773502691896
321
322          gausz_w(1) = 1.0
323          gausz_w(2) = 1.0
324
325      CASE (3)
326          gausz_r(1) = -0.7745966692414

```

```

327   gausz_r (2) = 0.0
328   gausz_r (3) = 0.7745966692414
329
330   gausz_w (1) = 0.5555555555555
331   gausz_w (2) = 0.8888888888888
332   gausz_w (3) = 0.5555555555555
333
334   CASE (4)
335   gausz_r (1) = -0.8611363115940
336   gausz_r (2) = -0.3399810435848
337   gausz_r (3) = 0.3399810435848
338   gausz_r (4) = 0.8611363115940
339
340   gausz_w (1) = 0.3478548451374
341   gausz_w (2) = 0.6521451548625
342   gausz_w (3) = 0.6521451548625
343   gausz_w (4) = 0.3478548451374
344
345   CASE (5)
346   gausz_r (1) = -0.9061798459386
347   gausz_r (2) = -0.5384693101056
348   gausz_r (3) = 0.0
349   gausz_r (4) = 0.5384693101056
350   gausz_r (5) = 0.9061798459386
351
352   gausz_w (1) = 0.2369268850561
353   gausz_w (2) = 0.4786286704993
354   gausz_w (3) = 0.5688888888888
355   gausz_w (4) = 0.4786286704993
356   gausz_w (5) = 0.2369268850561
357
358   CASE (6)
359   gausz_r (1) = -0.9324695142031
360   gausz_r (2) = -0.6612093864662
361   gausz_r (3) = -0.2386191860831
362   gausz_r (4) = 0.2386191860831
363   gausz_r (5) = 0.6612093864662
364   gausz_r (6) = 0.9324695142031
365
366   gausz_w (1) = 0.1713244923791
367   gausz_w (2) = 0.3607615730481
368   gausz_w (3) = 0.4679139345726
369   gausz_w (4) = 0.4679139345726
370   gausz_w (5) = 0.3607615730481
371   gausz_w (6) = 0.1713244923791
372
373   CASE (7)
374   gausz_r (1) = -0.9491079123427
375   gausz_r (2) = -0.7415311855993
376   gausz_r (3) = -0.4058451513773
377   gausz_r (4) = 0.0
378   gausz_r (5) = 0.4058451513773
379   gausz_r (6) = 0.7415311855993
380   gausz_r (7) = 0.9491079123427
381
382   gausz_w (1) = 0.1294849661688
383   gausz_w (2) = 0.2797053914892
384   gausz_w (3) = 0.3818300505051
385   gausz_w (4) = 0.4179591836734
386   gausz_w (5) = 0.3818300505051
387   gausz_w (6) = 0.2797053914892
388   gausz_w (7) = 0.1294849661688
389
390   CASE (8)
391   gausz_r (1) = -0.9602898564975
392   gausz_r (2) = -0.7966664774136
393   gausz_r (3) = -0.5255324099163
394   gausz_r (4) = -0.1834346424956
395   gausz_r (5) = 0.1834346424956
396   gausz_r (6) = 0.5255324099163
397   gausz_r (7) = 0.7966664774136
398   gausz_r (8) = 0.9602898564975
399

```

```

400   gausz_w(1) = 0.1012285362903
401   gausz_w(2) = 0.2223810344533
402   gausz_w(3) = 0.3137066458778
403   gausz_w(4) = 0.3626837833783
404   gausz_w(5) = 0.3626837833783
405   gausz_w(6) = 0.3137066458778
406   gausz_w(7) = 0.2223810344533
407   gausz_w(8) = 0.1012285362903
408
409   CASE (9)
410   gausz_r(1) = -0.9681602395076
411   gausz_r(2) = -0.8360311073266
412   gausz_r(3) = -0.6133714327005
413   gausz_r(4) = -0.3242534234038
414   gausz_r(5) = 0.0
415   gausz_r(6) = 0.3242534234038
416   gausz_r(7) = 0.6133714327005
417   gausz_r(8) = 0.8360311073266
418   gausz_r(9) = 0.9681602395076
419
420   gausz_w(1) = 0.0812743883615
421   gausz_w(2) = 0.1806481606948
422   gausz_w(3) = 0.2606106964029
423   gausz_w(4) = 0.3123470770400
424   gausz_w(5) = 0.3302393550012
425   gausz_w(6) = 0.3123470770400
426   gausz_w(7) = 0.2606106964029
427   gausz_w(8) = 0.1806481606948
428   gausz_w(9) = 0.0812743883615
429
430   CASE DEFAULT
431     error_loc='gausz_zuweisung'
432     error_msg="Fehler beim Zuweisen der Gausz'schen Integrationspunkte und
Gewichtungsfaktoren. ueberpruefen Sie das Parameter-File."
433     CALL exit_error(error_loc,error_msg)
434   END SELECT gauusz_zuweisung
435
436   !DEBUGGING
437   PRINT *, '..... Check'
438   PRINT *, ''
439   PRINT *, '#####'
440   PRINT *, ''
441   PRINT *, 'MAIN ROUTINE:'
442   PRINT *, ''
443
444   ALLOCATE (veff(2,0:m_max),seff(2,0:m_max))
445
446   !
#####
447   ! Hauptteil: Effektive Volumen und Flaechen errechnen
448   !
#####
449
450   SELECT CASE (element_type) ! die Anzahl an Nodes pro Element haengt von der
bekannten Elementgeometrie ab.
451   CASE (24)
452     veff=seff_gausz(elements, nelelem, m_max, breite_2D)
453     seff=leff_gausz(faces, nface, m_max, breite_2D)
454   CASE (34)
455     veff=veff_gausz(elements, nelelem, m_max)
456     seff=seff_gausz(faces, nface, m_max, 1.)
457   END SELECT
458
459   WRITE (*, '(/,A8,4A16,/)' ) 'm', 'S1-V_eff', 'PIA-V_eff', 'S1-S_eff', 'PIA-S_eff'
460   DO m=0,m_max
461     WRITE (*, '(I8,4ES16.8)' ) m, veff(:,m), seff(:,m)
462   END DO
463
464   !
#####

```

```

465 ! Nachbereitung: Ausgabe des Ergebnis-files
466 !
#####
467
468   ausgabename=ADJUSTL(ausgabename)      ! vorlaufende Leerstellen nach hinten
      verschieben
469
470   OPEN (201, FILE=TRIM(ausgabename)//'.out', STATUS='replace') !'new'
471   WRITE (201,'(A8,ES16.8)') 'SMAX = ', smax_file
472   WRITE (201,'(A8,I16)') 'GAUSS = ', gausz_fakt
473   WRITE (201,'(A8,I16)') 'MMAX = ', m_max
474   WRITE (201,'(A8,I16)') 'SYMM = ', symm_fakt
475   WRITE (201,'(A8,I16)') 'GEOM = ', element_type
476   WRITE (201,'(A8,F16.8)') 'BREIT = ', breite_2D
477
478   WRITE (201,'(/,A8,4A16,/)' ) 'm', 'S1-V_eff', 'PIA-V_eff', 'S1-S_eff', 'PIA-S_eff'
479
480   DO i=0,m_max
481     WRITE (201, '(I8,4ES16.8)') i, veff(:,i), seff(:,i)
482   END DO
483   CLOSE(201)
484
485
486   DEALLOCATE (elements, faces, nodes, gausz_r, gausz_w, elem_nodes, surf_nodes)
487
488 END PROGRAM effektivesVol

```

B.2. overhead.f90

```

1  MODULE overhead
2  !
3  ! Stand: 03.03.2016
4  !
5  ! Versionen:
6  !
7  ! 1.0 : 24.02.2016 - Erste Lauffaehige Version
8  !
9  ! 1.1 : 03.03.2016 - PIA-Option geloescht - berechnet jetzt immer beide (als Array!)
10 !
11 ! 2.0 : 09.03.2016 - von elementweiser Berechnung auf Arrayoperationen umgestiegen
12 !
13 !
14 IMPLICIT none
15 SAVE
16
17 integer, dimension (:,:), allocatable :: elements, faces
18 ! 2D Arrays zum Verknuepfen von Knoten zu Elementen bzw. Flaechen
19 ! Laenge des Arrays ist die Anzahl der Nodes pro Element (bzw. Flaechen)
20 ! Tiefe ist die anzahl der Elemente (bzw. Flaechen), die zu berechnen sind.
21 ! (siehe Abschnitt 'Vorbereitung' des Hauptprogramms)
22 ! Diese Reihenfolge ermoeglicht leichteres einlesen und besseren zugriff auf
      einzelne Elemente,
23 ! da die Array Elemente mit gleichem 1. Index im Speicher gruppiert sind.
24 real, dimension (:,:), allocatable :: nodes
25 ! 2D Array zum halten der Knoteninformationen
26 ! Laenge ist die Anzahl der Knoten
27 ! Tiefe ist 6, zuerst 3 Raumkoordinaten, dann 3 Hauptnormalspannungen
28 ! Hier sind die Dimensionen verdreht, weil immer auf die 1. Hauptnormalspannung
      ALLER Knoten
29 ! zugleich zugegriffen werden soll.
30 real, dimension (:), allocatable :: gausz_r, gausz_w
31 ! Arrays mit den Abszissen und Gewichtungen der Gausz-Integration
32 integer :: gausz_fakt
33 ! Grad der Gauszintegration
34
35
36 CONTAINS
37
38 FUNCTION leff_gausz(nlist,nelem,m_max,breite)
39
40   integer, INTENT(IN) :: m_max

```

```

41      ! Maximaler Weibull-Modul bis zu dem hinberechnet werden soll
42      integer, parameter :: eckn=2
43      ! Anzahl der Ecknodes fuer ein Linienelement
44      integer, INTENT(IN) :: nelem
45      ! Tiefe der Element-Liste eingehenden Elementliste nlist
46      integer, dimension(eckn,nelem), INTENT(IN) :: nlist
47      ! Zu behandelnde Elementliste
48      ! Diese Eingabe ist nur noetig damit das Programm zwischen 2D "Volumens"-
49      ! Elementen und 3D Oberflaechen unterscheiden kann,
50      ! wurde der einheitlichkeit aber auf alle drei Funktionen ueberteagen
51      integer :: m,n,e,i
52      ! Diverse Zaehler
53      integer, dimension(eckn), parameter :: xin = (/ -1, 1 /)
54      ! Transformierte Koordinaten der Eckknoten
55      real, dimension (:,:), allocatable :: formFktH, dh_dxi, dh_deta, dh_dzeta
56      ! Array fuer die Werte der Formfunktion und deren Ableitungen nach den
57      ! Bildraum-koordinaten.
58      ! Dieses Array hat (1+n) Dimensionen, wobei n die Dimesion der zu
59      ! inetrierenden Elemente ist
60      ! Also 2D fuer Linien, 3D fuer Flaechen, und 4D fuer Volumen.
61      ! Die erste Dimension zaehlt die Anzahl der Knoten pro Element (also die
62      ! Anzahl der ueberlappenden Formfunktionen)
63      ! Die nachfolgenden zaehlen die Anzahl der Integrationspunkte in jede Dimesion
64      real, dimension (:,:,), allocatable :: element_nodes
65      ! 3D Array Mit Elementweisen Knoteninformationen
66      ! Die erste Dimension zaehlt alle Elemente, die zu berechnen sind
67      ! Die zweite Dimension zaehlt die Knoten, die jeses Element ausmachen
68      ! Die dritte Dimension zaehlt bis 7 und beinhaltet die 3 Raumkoordinaten, 3
69      ! Hauptnormalspannungen und 1 Vergleichsspannung eines jeden Knoten
70      real, dimension (:,:), allocatable :: gram, sigma1, sigmaV, intLen1, intLenV,
71      dx_dxi, dy_dxi, dz_dxi
72      ! Arrays fuer die Werte der Gram'sche Determinante (nicht-quadratisches
73      ! aquivalent der JacobiDet), Spannunsinterpolationen, effektive Integrale
74      ! und Ableitungen der Raumkoordinaten in den Bildraum fuer die
75      ! Determinantenberechnung. Werte fuer jeden Integrationspunkt in jedem
76      ! Element.
77      ! Diese Arrays haben auch (1+n) Dimensionen, wie oben.
78      ! Hier zaehlt die erste Dim aber die Anzahl aller Elemente, die zu berechnen
79      ! sind
80      ! Alle weiteren sind wieder die Integrationspunkte in den jeweiligen
81      ! Dimensionen
82      !real, dimension (eckn,gausz_fakt) :: formFktH, dh_dxi
83      !real, dimension (nelem,eckn,7) :: element_nodes
84      !real, dimension (nelem,gausz_fakt) :: gram, sigma1, sigmaV, intLen1, intLenV,
85      dx_dxi, dy_dxi, dz_dxi
86      real, dimension (2,0:m_max) :: leff_gausz
87      ! Rueckgabewert der Function. Muss als Array definiert werden um als Array
88      ! zurueckgegeben werden zu koennen.
89      real, INTENT(IN) :: breite
90      ! Breite der Elemente im 2D Fall
91
92      !DEBUGGING
93      PRINT *, '===== '
94      PRINT *, ''
95      PRINT *, 'Linien Gausz-Quadratur'
96      PRINT *, ''
97      PRINT *, 'Artray Allocation'
98
99      ALLOCATE(formFktH(eckn,gausz_fakt), dh_dxi(eckn,gausz_fakt), dh_deta(eckn,
100      gausz_fakt), dh_dzeta(eckn,gausz_fakt))
101      ALLOCATE(element_nodes(nelem,eckn,7))
102      ALLOCATE(gram(nelem,gausz_fakt), sigma1(nelem,gausz_fakt), sigmaV(nelem,gausz_fakt)
103      ), intLen1(nelem,gausz_fakt), intLenV(nelem,gausz_fakt))
104      ALLOCATE(dx_dxi(nelem,gausz_fakt), dy_dxi(nelem,gausz_fakt), dz_dxi(nelem,
105      gausz_fakt))
106
107      PRINT *, '..... Done'
108      PRINT *, 'Ansatzfunktion:'
109
110      DO n=1,eckn
111      DO i=1,gausz_fakt

```

```

98      formFktH(n,i) = (1 + xin(n)*gausz_r(i))/2.
99      dh_dxi(n,i) = ( xin(n) )/2.
100
101     END DO
102     END DO
103
104     !DEBUGGING
105     PRINT *, '..... Done'
106     PRINT *, 'Element-Node-Array:'
107
108     DO e=1,nelem
109     IF (nlist(1,e)/=0) THEN
110         DO n=1,eckn
111             element_nodes(e,n,:) = nodes(nlist(n,e),:)
112         END DO
113     ELSE
114         element_nodes(e,,:) = 0.
115     END IF
116     END DO
117
118     !DEBUGGING
119     PRINT *, '..... Done'
120     PRINT *, 'Ableitungen der Ansatzfunktion:'
121
122     dx_dxi = 0
123     dy_dxi = 0
124     dz_dxi = 0
125
126     DO i=1,gausz_fakt
127     DO n=1,eckn
128
129         dx_dxi(:,i) = dx_dxi(:,i) + element_nodes(:,n,1)*dh_dxi(n,i)
130         dy_dxi(:,i) = dy_dxi(:,i) + element_nodes(:,n,2)*dh_dxi(n,i)
131         dz_dxi(:,i) = dz_dxi(:,i) + element_nodes(:,n,3)*dh_dxi(n,i)
132
133     END DO
134     END DO
135
136     !DEBUGGING
137     PRINT *, '..... Done'
138     PRINT *, 'Gramsche Determinante:'
139
140     gram = dx_dxi**2 + dy_dxi**2 + dz_dxi**2
141     gram = SQRT(gram)
142
143     !DEBUGGING
144     PRINT *, '..... Done'
145     PRINT *, 'Spannung in Integrationspunkten:'
146
147     sigma1 = 0
148
149     DO i=1,gausz_fakt
150         gram(:,i) = gram(:,i) * gauz_w(i)
151         DO n=1,eckn
152             sigma1(:,i) = sigma1(:,i) + element_nodes(:,n,4)*formFktH(n,i)
153         END DO
154     END DO
155
156     !DEBUGGING
157     PRINT *, '..... Done'
158     PRINT *, '~~~~~',
159     PRINT *, 'M-Loop:'
160
161     DO m=0,m_max
162
163         WRITE(*,'(A28,I3,A12,I3,A1,A1,$)') 'Linien Gausz-Quadratur, m = ',m,',
164             Progress: ',(m*100)/m_max,'% ', achar(13)
165
166         element_nodes(:, :, 7) = ( element_nodes(:, :, 4)**m + element_nodes(:, :, 5)**m +
167             element_nodes(:, :, 6)**m )**(1./REAL(m))
168         sigmaV = 0
169     END DO
170
171     DO i=1,gausz_fakt

```

```

169         DO n=1,eckn
170             sigmaV(:,i) = sigmaV(:,i) + element_nodes(:,n,7)*formFktH(n,i)
171         END DO
172     END DO
173
174         intLen1 = sigma1**m * gram
175         intLenV = sigmaV**m * gram
176
177         leff_gausz(1,m) = SUM(intLen1)
178         leff_gausz(2,m) = SUM(intLenV)
179
180     END DO
181     !DEBUGGING
182     PRINT *, '..... Done'
183     PRINT *, 'deallocate Arrays:'
184     DEALLOCATE(formFktH, dh_dxi, dh_deta, dh_dzeta, element_nodes, gram, sigma1,
185               sigmaV, intLen1, intLenV, dx_dxi, dy_dxi, dz_dxi)
186     PRINT *, '..... Done'
187 END FUNCTION
188
189 FUNCTION seff_gausz(nlist,nelem,m_max,breite)
190
191     integer, INTENT(IN) :: m_max
192     integer, parameter :: eckn=4
193     integer, INTENT(IN) :: nelem
194     integer, dimension(eckn,nelem), INTENT(IN) :: nlist
195     integer :: m,n,e,i,j
196     integer, dimension(eckn), parameter :: xin = (/ -1, 1, 1, -1 /)
197     integer, dimension(eckn), parameter :: etan = (/ -1, -1, 1, 1 /)
198     real, dimension (:,:,), allocatable :: formFktH, dh_dxi, dh_deta, dh_dzeta
199     real, dimension (:,:,), allocatable :: element_nodes
200     real, dimension (:,:,), allocatable :: gram, sigma1, sigmaV, intSurf1, intSurfV
201     real, dimension (:,:,), allocatable :: dx_dxi, dx_deta, dy_dxi, dy_deta, dz_dxi,
202       dz_deta
203     !real, dimension (eckn,gausz_fakt,gausz_fakt) :: formFktH, dh_dxi, dh_deta
204     !real, dimension (nelem,eckn,7) :: element_nodes
205     !real, dimension (nelem,gausz_fakt,gausz_fakt) :: gram, sigma1, sigmaV, intSurf1,
206       intSurfV
207     !real, dimension (nelem,gausz_fakt,gausz_fakt) :: dx_dxi, dx_deta, dy_dxi, dy_deta
208       , dz_dxi, dz_deta
209     real, dimension (2,0:m_max) :: seff_gausz
210     real, INTENT(IN) :: breite
211
212     !DEBUGGING
213     PRINT *, '===== '
214     PRINT *, ''
215     PRINT *, 'Flaechen Gausz-Quadratur'
216     PRINT *, ''
217     PRINT *, 'Artray Allocation'
218
219     ALLOCATE(formFktH(eckn,gausz_fakt,gausz_fakt), dh_dxi(eckn,gausz_fakt,gausz_fakt),
220             dh_deta(eckn,gausz_fakt,gausz_fakt), dh_dzeta(eckn,gausz_fakt,gausz_fakt))
221     ALLOCATE(element_nodes(nelem,eckn,7))
222     ALLOCATE(gram(nelem,gausz_fakt,gausz_fakt), sigma1(nelem,gausz_fakt,gausz_fakt),
223             sigmaV(nelem,gausz_fakt,gausz_fakt), intSurf1(nelem,gausz_fakt,gausz_fakt),
224             intSurfV(nelem,gausz_fakt,gausz_fakt))
225     ALLOCATE(dx_dxi(nelem,gausz_fakt,gausz_fakt), dx_deta(nelem,gausz_fakt,gausz_fakt)
226             , dy_dxi(nelem,gausz_fakt,gausz_fakt), dy_deta(nelem,gausz_fakt,gausz_fakt))
227     ALLOCATE(dz_dxi(nelem,gausz_fakt,gausz_fakt), dz_deta(nelem,gausz_fakt,gausz_fakt)
228             )
229
230     PRINT *, '..... Done'
231     PRINT *, 'Ansatzfunktion:'
232
233     DO n=1,eckn
234     DO i=1,gausz_fakt
235     DO j=1,gausz_fakt
236
237         formFktH(n,i,j) = ( (1 + xin(n)*gausz_r(i)) * (1 + etan(n)*gausz_r(j)) ) /4.
238         dh_dxi(n,i,j) = ( xin(n) * (1 + etan(n)*gausz_r(j)) ) /4.

```

```

233         dh_deta(n,i,j) = ( (1 + xin(n)*gausz_r(i)) *          etan(n)          )/4.
234
235     END DO
236     END DO
237     END DO
238
239     !DEBUGGING
240     PRINT *, '..... Done'
241     PRINT *, 'Element-Node-Array:'
242
243     DO e=1,nelem
244     IF (nlist(1,e)/=0) THEN
245         DO n=1,eckn
246             element_nodes(e,n,:) = nodes(nlist(n,e),:)
247         END DO
248     ELSE
249         element_nodes(e,,:) = 0.
250     END IF
251     END DO
252
253     !DEBUGGING
254     PRINT *, '..... Done'
255     PRINT *, 'Ableitungen der Ansatzfunktion:'
256
257     dx_dxi = 0
258     dx_deta = 0
259     dy_dxi = 0
260     dy_deta = 0
261     dz_dxi = 0
262     dz_deta = 0
263
264     DO i=1,gausz_fakt
265     DO j=1,gausz_fakt
266     DO n=1,eckn
267
268         dx_dxi(:,i,j) = dx_dxi(:,i,j) + element_nodes(:,n,1)*dh_dxi(n,i,j)
269         dx_deta(:,i,j) = dx_deta(:,i,j) + element_nodes(:,n,1)*dh_deta(n,i,j)
270
271         dy_dxi(:,i,j) = dy_dxi(:,i,j) + element_nodes(:,n,2)*dh_dxi(n,i,j)
272         dy_deta(:,i,j) = dy_deta(:,i,j) + element_nodes(:,n,2)*dh_deta(n,i,j)
273
274         dz_dxi(:,i,j) = dz_dxi(:,i,j) + element_nodes(:,n,3)*dh_dxi(n,i,j)
275         dz_deta(:,i,j) = dz_deta(:,i,j) + element_nodes(:,n,3)*dh_deta(n,i,j)
276
277     END DO
278     END DO
279     END DO
280
281     !DEBUGGING
282     PRINT *, '..... Done'
283     PRINT *, 'Gramsche Determinante:'
284
285     gram = dx_dxi**2 * (dy_deta**2 + dz_deta**2)
286     gram = gram + (dy_dxi*dz_deta - dy_deta*dz_dxi)**2
287     gram = gram - 2*dx_dxi*dx_deta*(dy_dxi*dy_deta + dz_dxi*dz_deta)
288     gram = gram + dx_deta**2 * (dy_dxi**2 + dz_dxi**2)
289     gram = SQRT(gram)
290
291     !DEBUGGING
292     PRINT *, '..... Done'
293     PRINT *, 'Spannung in Integrationspunkten:'
294
295     sigma1 = 0
296
297     DO i=1,gausz_fakt
298     DO j=1,gausz_fakt
299         gram(:,i,j) = gram(:,i,j) * gausz_w(i) * gausz_w(j)
300         DO n=1,eckn
301             sigma1(:,i,j) = sigma1(:,i,j) + element_nodes(:,n,4)*formFktH(n,i,j)
302         END DO
303     END DO
304     END DO
305

```

```

306      !DEBUGGING
307      PRINT *, '..... Done'
308      PRINT *, '~~~~~',
309      PRINT *, 'M-Loop:'
310
311      DO m=0,m_max
312
313          WRITE(*,'(A28,I3,A12,I3,A1,A1,$)') 'Flaechen Gauss-Quadratur, m = ',m,'
          Progress: ',(m*100)/m_max,'% ',achar(13)
314
315          element_nodes(:, :, 7) = ( element_nodes(:, :, 4)**m + element_nodes(:, :, 5)**m +
          element_nodes(:, :, 6)**m )**(1./REAL(m))
316
317          sigmaV = 0
318
319          DO i=1,gausz_fakt
320          DO j=1,gausz_fakt
321          DO n=1,eckn
322          sigmaV(:,i,j) = sigmaV(:,i,j) + element_nodes(:,n,7)*formFktH(n,i,j)
323          END DO
324          END DO
325          END DO
326
327          intSurf1 = sigma1**m * gram
328          intSurfV = sigmaV**m * gram
329
330
331          seff_gausz(1,m) = SUM(intSurf1)
332          seff_gausz(2,m) = SUM(intSurfV)
333
334
335
336
337      END DO
338      !DEBUGGING
339      PRINT *, '..... Done'
340      PRINT *, 'deallocate Arrays:'
341      DEALLOCATE(formFktH, dh_dxi, dh_deta, dh_dzeta, element_nodes, gram, sigma1,
          sigmaV, intSurf1, intSurfV, dx_dxi, dx_deta, dy_dxi, dy_deta, dz_dxi, dz_deta)
342      PRINT *, '..... Done'
343
344  END FUNCTION
345
346  FUNCTION veff_gausz(nlist,nelem,m_max)
347
348      integer, INTENT(IN) :: m_max
349      integer, parameter :: eckn=8
350      integer, INTENT(IN) :: nelem
351      integer, dimension(:, :), INTENT(IN) :: nlist
352      integer :: m,n,e,i,j,k
353      integer, dimension(8), parameter :: xin = (/ -1, 1, 1, -1, -1, 1, 1, -1 /)
354      integer, dimension(8), parameter :: etan = (/ -1, -1, 1, 1, -1, -1, 1, 1 /)
355      integer, dimension(8), parameter :: zetan = (/ -1, -1, -1, -1, 1, 1, 1, 1 /)
356      real, dimension (:, :, :, :), allocatable :: formFktH, dh_dxi, dh_deta, dh_dzeta
357      real, dimension (:, :, :), allocatable :: element_nodes
358      real, dimension (:, :, :, :), allocatable :: jacobi, sigma1, sigmaV, intVol1, intVolV
359      real, dimension (:, :, :, :), allocatable :: dx_dxi, dx_deta, dx_dzeta, dy_dxi,
          dy_deta
360      real, dimension (:, :, :, :), allocatable :: dy_dzeta, dz_dxi, dz_deta, dz_dzeta
361      !real, dimension (eckn,gausz_fakt,gausz_fakt,gausz_fakt) :: formFktH, dh_dxi,
          dh_deta, dh_dzeta
362      !real, dimension (nelem,eckn,7) :: element_nodes
363      !real, dimension (nelem,gausz_fakt,gausz_fakt,gausz_fakt) :: jacobi, sigma1,
          sigmaV, intVol1, intVolV
364      !real, dimension (nelem,gausz_fakt,gausz_fakt,gausz_fakt) :: dx_dxi, dx_deta,
          dx_dzeta, dy_dxi, dy_deta
365      !real, dimension (nelem,gausz_fakt,gausz_fakt,gausz_fakt) :: dy_dzeta, dz_dxi,
          dz_deta, dz_dzeta
366      real, dimension (2,0:m_max) :: veff_gausz
367
368      !DEBUGGING
369      PRINT *, '===== '
370      PRINT *, ''

```

```

371 PRINT *, 'Volumen Gausz-Quadratur '
372 PRINT *, ''
373 PRINT *, 'Artray Allocation'
374
375
376 ALLOCATE(formFktH(eckn,gausz_fakt,gausz_fakt,gausz_fakt), dh_dxi(eckn,gausz_fakt,
    gausz_fakt,gausz_fakt), dh_deta(eckn,gausz_fakt,gausz_fakt,gausz_fakt),
    dh_dzeta(eckn,gausz_fakt,gausz_fakt,gausz_fakt))
377 ALLOCATE(element_nodes(nelem,eckn,7))
378 ALLOCATE(jacobi(nelem,gausz_fakt,gausz_fakt,gausz_fakt), sigma1(nelem,gausz_fakt,
    gausz_fakt,gausz_fakt), sigmaV(nelem,gausz_fakt,gausz_fakt,gausz_fakt),
    intVol1(nelem,gausz_fakt,gausz_fakt,gausz_fakt), intVolV(nelem,gausz_fakt,
    gausz_fakt,gausz_fakt))
379 ALLOCATE(dx_dxi(nelem,gausz_fakt,gausz_fakt,gausz_fakt), dx_deta(nelem,gausz_fakt,
    gausz_fakt,gausz_fakt), dx_dzeta(nelem,gausz_fakt,gausz_fakt,gausz_fakt),
    dy_dxi(nelem,gausz_fakt,gausz_fakt,gausz_fakt), dy_deta(nelem,gausz_fakt,
    gausz_fakt,gausz_fakt))
380 ALLOCATE(dy_dzeta(nelem,gausz_fakt,gausz_fakt,gausz_fakt), dz_dxi(nelem,gausz_fakt
    ,gausz_fakt,gausz_fakt), dz_deta(nelem,gausz_fakt,gausz_fakt,gausz_fakt),
    dz_dzeta(nelem,gausz_fakt,gausz_fakt,gausz_fakt))
381
382
383 PRINT *, '..... Done'
384 PRINT *, 'Ansatzfunktion:'
385
386 DO n=1,eckn
387 DO i=1,gausz_fakt
388 DO j=1,gausz_fakt
389 DO k=1,gausz_fakt
390
391     formFktH(n,i,j,k) = ( (1 + xin(n)*gausz_r(i)) * (1 + etan(n)*gausz_r(j)) * (1
    + zetan(n)*gausz_r(k)) )/8.
392     dh_dxi(n,i,j,k) = ( xin(n) * (1 + etan(n)*gausz_r(j)) * (1
    + zetan(n)*gausz_r(k)) )/8.
393     dh_deta(n,i,j,k) = ( (1 + xin(n)*gausz_r(i)) * etan(n) * (1
    + zetan(n)*gausz_r(k)) )/8.
394     dh_dzeta(n,i,j,k) = ( (1 + xin(n)*gausz_r(i)) * (1 + etan(n)*gausz_r(j)) *
    zetan(n) )/8.
395
396 END DO
397 END DO
398 END DO
399 END DO
400
401 !DEBUGGING
402 PRINT *, '..... Done'
403 PRINT *, 'Element-Node-Array:'
404
405 DO e=1,nelem
406 IF (nlist(1,e)/=0) THEN
407 DO n=1,eckn
408     element_nodes(e,n,:) = nodes(nlist(n,e),:)
409 END DO
410 ELSE
411     element_nodes(e,:,:) = 0.
412 END IF
413 END DO
414
415 !DEBUGGING
416 PRINT *, '..... Done'
417 PRINT *, 'Ableitungen der Ansatzfunktion:'
418
419 dx_dxi = 0
420 dx_deta = 0
421 dx_dzeta = 0
422 dy_dxi = 0
423 dy_deta = 0
424 dy_dzeta = 0
425 dz_dxi = 0
426 dz_deta = 0
427 dz_dzeta = 0
428
429 DO i=1,gausz_fakt

```

```

430 DO j=1,gausz_fakt
431 DO k=1,gausz_fakt
432 DO n=1,eckn
433
434 dx_dxi(:,i,j,k) =dx_dxi(:,i,j,k) + element_nodes(:,n,1)*dh_dxi(n,i,j,k)
435 dx_deta(:,i,j,k) =dx_deta(:,i,j,k) + element_nodes(:,n,1)*dh_deta(n,i,j,k)
436 dx_dzeta(:,i,j,k) =dx_dzeta(:,i,j,k)+ element_nodes(:,n,1)*dh_dzeta(n,i,j,k)
437
438 dy_dxi(:,i,j,k) =dy_dxi(:,i,j,k) + element_nodes(:,n,2)*dh_dxi(n,i,j,k)
439 dy_deta(:,i,j,k) =dy_deta(:,i,j,k) + element_nodes(:,n,2)*dh_deta(n,i,j,k)
440 dy_dzeta(:,i,j,k) =dy_dzeta(:,i,j,k)+ element_nodes(:,n,2)*dh_dzeta(n,i,j,k)
441
442 dz_dxi(:,i,j,k) =dz_dxi(:,i,j,k) + element_nodes(:,n,3)*dh_dxi(n,i,j,k)
443 dz_deta(:,i,j,k) =dz_deta(:,i,j,k) + element_nodes(:,n,3)*dh_deta(n,i,j,k)
444 dz_dzeta(:,i,j,k) =dz_dzeta(:,i,j,k)+ element_nodes(:,n,3)*dh_dzeta(n,i,j,k)
445
446 END DO
447 END DO
448 END DO
449 END DO
450
451 !DEBUGGING
452 PRINT *, '..... Done'
453 PRINT *, 'Jacobi Determinante:'
454
455 jacobi = dx_dxi*dy_deta*dz_dzeta + dx_dzeta*dy_dxi*dz_deta + dx_deta*
dy_dzeta*dz_dxi
456 jacobi = jacobi - dx_dzeta*dy_deta*dz_dxi - dx_dxi*dy_dzeta*dz_deta - dx_deta*
dy_dxi*dz_dzeta
457
458 !DEBUGGING
459 PRINT *, '..... Done'
460 PRINT *, 'Spannung in Integrationspunkten:'
461
462 sigma1 = 0
463
464 DO i=1,gausz_fakt
465 DO j=1,gausz_fakt
466 DO k=1,gausz_fakt
467 jacobi(:,i,j,k) = jacobi(:,i,j,k) * gausz_w(i) * gausz_w(j) * gausz_w(k)
468 DO n=1,eckn
469 sigma1(:,i,j,k) = sigma1(:,i,j,k) + element_nodes(:,n,4)*formFktH(n,i,j,k)
470 END DO
471 END DO
472 END DO
473 END DO
474
475
476 !DEBUGGING
477 PRINT *, '..... Done'
478 PRINT *, '~~~~~',
479 PRINT *, 'M-Loop:'
480
481 DO m=0,m_max
482
483 WRITE(*,'(A28,I3,A12,I3,A1,A1,$)') 'Volumen Gausz-Quadratur, m = ',m,',
Progress: ',(m*100)/m_max,'% ',achar(13)
484
485 element_nodes(:, :,7) = (element_nodes(:, :,4)**m + element_nodes(:, :,5)**m +
element_nodes(:, :,6)**m )**(1./REAL(m))
486 sigmaV = 0
487
488 DO i=1,gausz_fakt
489 DO j=1,gausz_fakt
490 DO k=1,gausz_fakt
491 DO n=1,eckn
492 sigmaV(:,i,j,k) = sigmaV(:,i,j,k) + element_nodes(:,n,7)*formFktH(n,i,j,k)
493 END DO
494 END DO
495 END DO
496 END DO
497
498 intVol1 = sigma1**m * jacobi

```

```

499     intVolV = sigmaV**m * jacobi
500
501
502     veff_gausz(1,m) = SUM(intVol1)
503     veff_gausz(2,m) = SUM(intVolV)
504
505
506 END DO
507 !DEBUGGING
508 PRINT *, '..... Done '
509 PRINT *, 'deallocate Arrays:'
510 DEALLOCATE(formFktH, dh_dxi, dh_deta, dh_dzeta ,element_nodes ,jacobi, sigma1,
           sigmaV, intVol1, intVolV,dx_dxi, dx_deta, dx_dzeta, dy_dxi, dy_deta,dy_dzeta,
           dz_dxi, dz_deta, dz_dzeta)
511 PRINT *, '..... Done'
512
513 END FUNCTION
514
515 SUBROUTINE exit_error(location,message)
516
517 ! character :: confirm='n'
518 character (len=32), INTENT(IN) :: location
519 character (len=256), INTENT(IN) :: message
520
521 ! DO
522     WRITE (*,'(A72)') 'Ein Fehler in folgender Routine hat zum Abbrechen des
           Programms gefuehrt:'
523     WRITE (*,'(X)')
524     WRITE (*,'(A32)') location
525     WRITE (*,'(A256)') message
526 !     WRITE (*,'(A25)') 'Bitte bestaetigen mit "y":'
527 !     READ (*,*) confirm
528 !     IF (confirm=='y') STOP
529 ! END DO
530 END SUBROUTINE
531
532 END MODULE overhead

```