



## Masterarbeit

# Entwicklung einer gekoppelten FEM-DEM Simulation auf Basis ausgewählter Beispiele

Eingereicht von: Wenjie Huang  
Matrikel-Nr.: 56700

am 09.01.2017

Betreuer: M.-Sc. Thomas Falke

Prüfer: Prof. Dr.-Ing. Matthias Kröger  
Dr.-techn. Robert Szlosarek

---

## Erklärung

Hiermit versichere ich, die vorliegende Arbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Freiberg, den \_\_. \_\_.20\_\_

---

(Vorname Name)



# Inhaltsverzeichnis

<b>Inhaltsverzeichnis .....</b>	<b>i</b>
<b>Symbol- und Abkürzungsverzeichnis .....</b>	<b>iii</b>
<b>1 .....</b>	<b>1</b>
<b>Einleitung.....</b>	<b>1</b>
<b>2 .....</b>	<b>3</b>
<b>Theoretische Grundlagen .....</b>	<b>3</b>
2.1 .....	3
FEM .....	3
2.2 .....	3
DEM.....	3
2.3 .....	3
Anwendungsbeispiele .....	3
2.4 .....	5
Unterschied zwischen impliziten und expliziten Verfahren.....	5
<b>3 .....</b>	<b>9</b>
<b>Analytische Betrachtung .....</b>	<b>9</b>
<b>4 .....</b>	<b>13</b>
<b>Kopplungsmethode .....</b>	<b>13</b>
4.1 .....	13
Vorstellung der Methode .....	13
4.2 .....	14
Umsetzung.....	14
4.2.1 .....	14
Vorbereitung .....	14
4.2.2 .....	15
Zeiteinstellung.....	15
4.2.3 .....	15
Datenfluss.....	15
4.3 .....	17
Variantenvergleich .....	17
4.3.1 .....	17
Spannungszuwachs .....	17
4.3.2 .....	18
Auferlegung von Verschiebung auf Balken.....	18
4.3.3 .....	18
User Window Subroutine.....	18
4.4 .....	18
Einstellungen von LIGGGHTS.....	18
Dämpfungskonstante und Reibungskonstante .....	19
4.5 .....	20
Einstellungen von RADIOSS .....	20
<b>5 .....</b>	<b>21</b>
<b>Simulationsstudien .....</b>	<b>21</b>
5.1 .....	21
Ausgangsmodell .....	21
5.1.1 .....	21
Geometrie des Ausgangsmodells.....	21
5.1.2 .....	22
Abweichung zwischen Analyse und Simulation .....	22
5.1.3 .....	22
Durchlauf der Verschiebungen und Spannungszuwachs.....	22
5.2 .....	25
Parameterstudien.....	25
5.2.1 .....	25
Spannung in RADIOSS .....	25
5.2.2 .....	26
Zeitschritt .....	26
5.2.3 .....	27
Schrittweite .....	27
5.2.4 .....	29
Dicke des Balkens.....	29
5.2.5 .....	29
Radius der Kugel.....	29
5.2.6 .....	31
Anfangsgeschwindigkeit der Kugel.....	31
5.3 .....	31
Einheitssystem .....	31
5.4 .....	32
Diskussion über Gründe von Fehlern .....	32
<b>6 .....</b>	<b>35</b>
<b>Zusammenfassung .....</b>	<b>35</b>

<b>7.....</b>	<b>Quellenverzeichnis .....</b>	<b>37</b>
<b>Anhang A: Daten CD .....</b>		<b>A-1</b>

## Symbol- und Abkürzungsverzeichnis

### Lateinische Notation

a	m	Länge
b	m	Breite
c	-	Federkonstante
E	N/mm <sup>2</sup>	E-Modul
f	-	Fehler
F	N	Kraft
G	-	Funktion
h	m	Dicke
I	m <sup>4</sup>	Flächenträgheitsmoment
k	-	Konstante
l	m	Länge
m	kg	Masse
s	m <sup>2</sup>	Fläche
t	s	Zeit
u	m/s	Geschwindigkeit
v	m/s	Geschwindigkeit
w	m	Durchbiegung
Y	-	Funktion
y	-	Lösung

### Griechische Notation

$\zeta$	-	Verhältnis
$\sigma$		Spannung

## Indices

$()_0$	Anfangs-
$()_a$	Analytische
$()_b$	Balken
$()_i$	Zählvariable
$()_k$	Kugel
$()_K$	Lösung
$()_s$	Simulierte
$()_S$	Stoß
$O_{REL}$	Relative

# 1 Einleitung

Wenn Granulate auf einen dünnwandigen Metalsilo geschüttet werden, geben die Granulate dem elastischen Silo große Impulse. Um das Fließverhalten von Granulaten im Silo zu untersuchen, soll der Kontakt zwischen den beiden detailliert beschrieben werden. Für die Simulation von Granulaten ist die Diskrete-Elemente-Methode geeignet, wobei Granulate vereinfacht als Kugeln betrachtet werden können. Die elastische Verformung des Silos lässt sich durch die Finiten-Elemente-Methode berechnen.

Kopplung zwischen FEM und DEM ist ein neues Objekt der Forschung. Es gibt ein paar unterschiedliche Arten der Kopplung. Die Kopplung von dieser Arbeit gehört zu Oberflächenkopplung. FEM berechnet in ihrem eigenen Bereich, während DEM in einem anderen berechnet. Beide Bereiche sind innerhalb Simulation physikalisch voneinander getrennt [Str13].

Ein einfaches Beispiel wird zu Beginn überlegt, dass eine Kugel auf einen einseitig eingespannten und aufgelagerten schmalen elastischen Balken fällt, der Balken sich durchbiegt und die Kugel wieder hochdrückt. Das Beispiel lässt sich nicht nur simulativ nachstellen, sondern auch analytisch berechnen.

Ziel dieser Arbeit ist es, eine sinnvolle und logische Kopplung von DEM-Software LIGGGHTS und FEM-Solver Altair Hyperworks RADIOSS zu entwickeln, welche für beide Methoden eine gemeinsame Umgebung durch Programmiersprache Python schafft, um die Simulationsmodelle später in übertragenem Sinn zu Granulaten und Silos gebrauchen zu können. LIGGGHTS ist für den Zustand der Kugel verantwortlich, während RADIOSS für Verformung des Balkens zuständig ist.

Beide Softwares haben Beschränkungen. In LIGGGHTS ist die Geometrie des Balkens immer ein Starrkörper. Dadurch folgt, dass eine Kugel auf einen starren Balken trifft, wobei keine Durchbiegung an der Geometrie stattfinden wird. RADIOSS kann die Dynamik mehrerer Granulate auch nicht. Deshalb ist eine Kopplung für die beiden Softwares nötig.

Der Balken wird als ein Schalenmodell vernetzt und berechnet. Nach Kontakt zwischen der Kugel und dem Balken rechnet LIGGGHTS eine Spannung auf der Schale und die Kontaktstelle heraus, dabei speichert LIGGGHTS den Zustand der Kugel in Restart-Files. Das Element in dem vernetzten Balken, wo der Kontakt stattfindet, wird mit entsprechende Spannung und verformter Schale in die Inputdatei von RADIOSS angetragen, dabei wird die Schale für LIGGGHTS auch aktualisiert, um auf Ausführung der nächsten Schleife vorzubereiten. Mit Algorithmus lassen sich mehrere Schleifen in der Kopplung durchlaufen.

Dann lässt sich das Simulationsergebnis untersuchen, ob die kopplungsfähige Simulation physikalisch korrekt funktioniert, ob sie im dynamischen Prozess möglich ist, und wie groß die Abweichung zwischen Simulation und Analyse ist. In der Zukunft lassen sich auch mit mehreren Kugeln oder komplizierterer Geometrie simulieren.



## 2 Theoretische Grundlagen

### 2.1 FEM

FEM steht für Methode der finiten Elemente. Sie ist ein universelles und flexibles Verfahren zur numerischen Lösung von Randwertproblemen. Die schwache Form der Randwertprobleme wird mit der Galerkin-Methode räumlich diskretisiert. Meist als isoparametrisches Element mit einfachen Polynomen werden Ansatzfunktionen für primäre Feldgrößen stückweise eingesetzt, dadurch lässt sich das Randwertproblem in ein Gleichungssystem umwandeln, das numerisch lösbar ist [Hut16].

FEM-Solver Altair Hyperworks RADIOSS wird in dieser Arbeit angewendet. RADIOSS hat ein breites Anwendungsgebiet, wie Crash-Sicherheit, Drop-Testing, Explosion und hydrodynamische Auswirkungen, terminal ballistische Flüssigkeit strukturelle Interaktion, Formung und Composite-Mapping [Alt16]. RADIOSS berechnet mit impliziter und expliziter Integrationsverfahren. Übrigens wird das Modell durch CAE-Software HyperMesh vernetzt.

### 2.2 DEM

DEM wird als Diskrete-Elemente-Methode definiert. Die Methode wird für Berechnung von Dynamik zahlreicher Granulate angewendet. Jedes einzelnes Granulat wird betrachtet und in mathematischer Weise beschrieben. Ihr mechanisches Verhalten von diskret unterteilten Strukturen und Anordnungen wird untersucht. Im Vergleich zur FEM ist die DEM eine gitterlose Methode [Bör11].

DEM-Software LIGGGHTS wird in dieser Arbeit verwendet. LIGGGHTS steht für „LAMMPS improved for general granular and granular heat transfer simulations“. LAMMPS ist eine Abkürzung von „Large-scale Atomic/Molecular Massively Parallel Simulator“. LAMMPS als der Vorfahr von LIGGGHTS ist zur Verwendung bei der Dimension von Molekülen gedacht. .

LIGGGHTS bietet eine gute Plattform für DEM-Simulationen mit mehreren Vorteilen an, wie seine flexible Skriptsprache, effiziente parallele Berechnung sowie verfügbare Nachbearbeitungs- und Visualisierungstoolbox. LIGGGHTS berechnet intern nur mit Granulaten. Es sei denn, dass eine platte Geometrie wie ein riesengroßes Granulat ist. Nachteilig kann LIGGGHTS die Verformung der Granulate nicht ausüben, deshalb ist eine Kopplung mit einer anderen FEM-Software nötig.

### 2.3 Anwendungsbeispiele

Kopplung zwischen FEM und DEM ist ein neues Objekt der Forschung. Es gibt ein paar unterschiedliche Arten der Kopplungen, wie Oberflächenkopplung, Volumenkopplung, Multiskalenkopplung oder Kontaktkopplung. Durch verschiedene Forschungsarbeiten der letzten Jahre stellen viele Anwendungsbeispiele zur Verfügung. Die folgenden Darlegungen basieren auf Ausführungen von Stransky [Str13].

Die Kopplung von dieser Arbeit gehört zu Oberflächenkopplung. FEM berechnet in einem Bereich, während DEM in einem anderen berechnet. Beide Bereiche sind innerhalb Simulation physikalisch voneinander getrennt. In ähnlicher Weise hat Michael [Mic15] Wechselwirkungen zwischen einer Reifenlauffläche und körnigem Böden simuliert. In der Abbildung 2.1 ist das Beispiel dargestellt. Diese Kopplungsart findet auch große Anwendung auf Geomechanische Aufgaben.

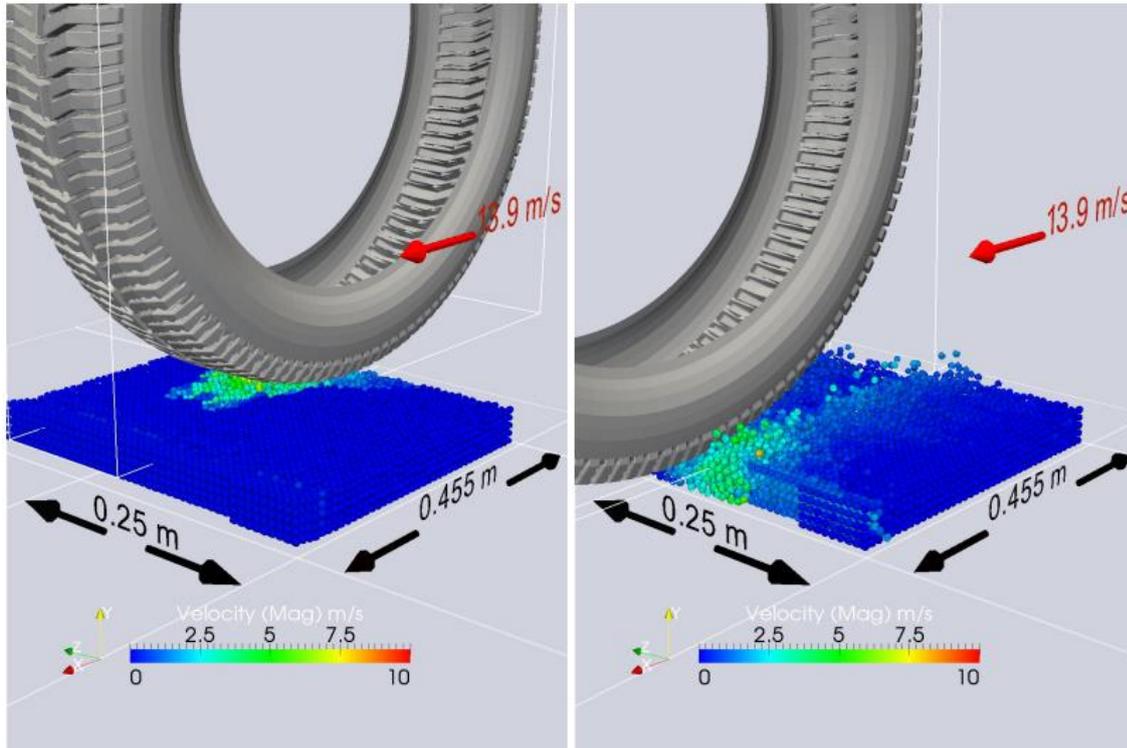


Abbildung 2.1: Simulation zwischen einer Reifenlauffläche und körnigem Böden [Mic15]

Volumenkopplung ist ähnlich mit Oberflächenkopplung, außer eine gewisse Überlappung zwischen FEM und DEM berücksichtigt wird. Beide Methoden sind verantwortlich für den überlappenden Bereich, um ein Kontinuum zu schaffen [Str13]. Volumenkopplung lässt sich auf dem Gebiet der Bruchmechanik in heterogenen Materialien einsetzen.

Wang [Wan14] hat das Dreikörper-Reibungsverhalten mit der FEM-DEM Kopplung untersucht. Sie gehört zu Multiskalenkopplung. Multiskalenkopplung löst Problem durch FEM (große Skala), außer Integrationspunkte durch DEM (kleine Skala) beschrieben werden, damit Rechenaufwand gering gehalten werden kann.

Kontaktkopplung lässt sich als FEM betrachten, außer Kontakterkennung auf einigen Stellen durch DEM berechnet wird. In der Abbildung 2.2 sind die Anwendungsbeispiele der anderen drei Kopplungsarten dargestellt.

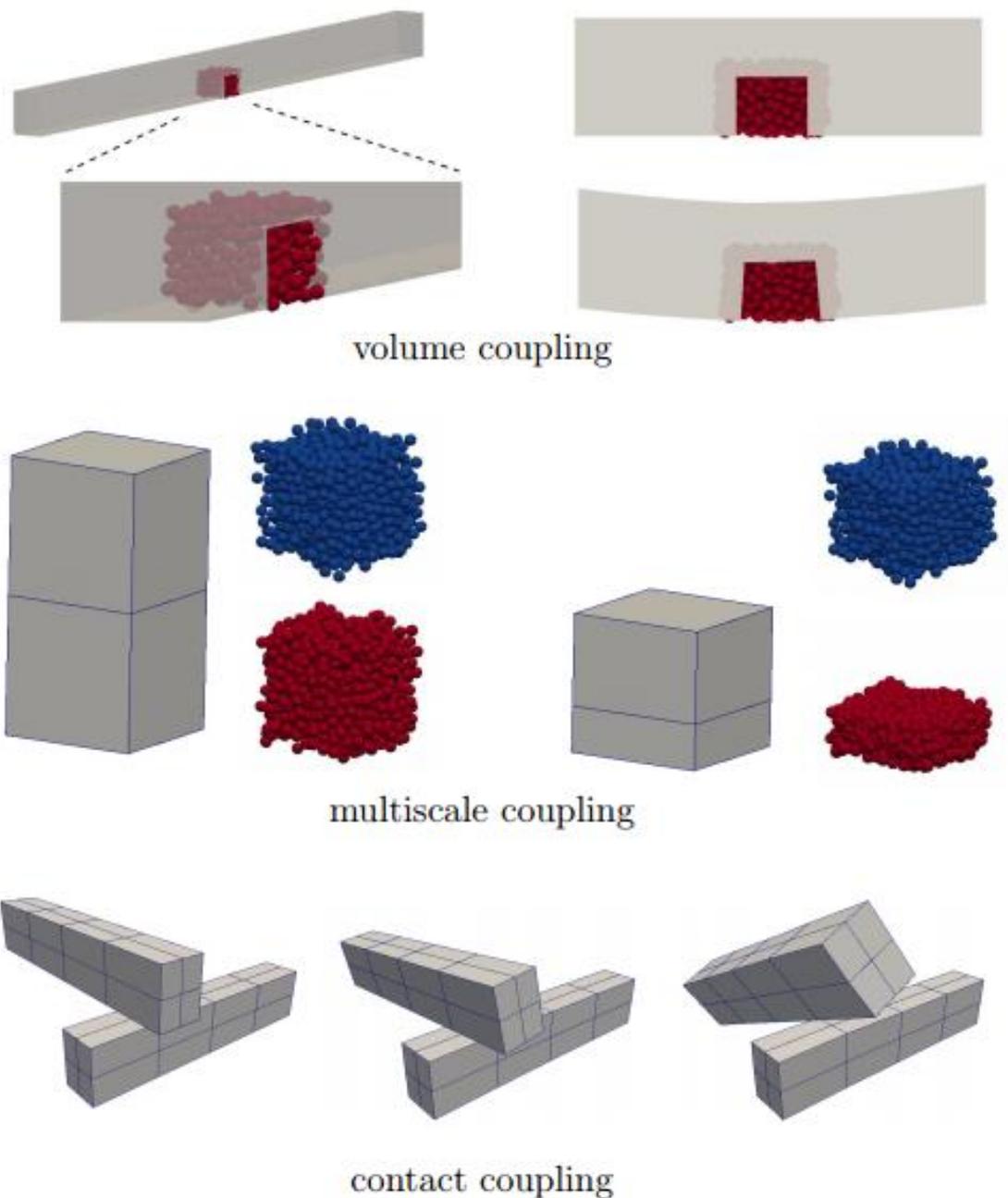


Abbildung 2.2: Anwendungsbeispiele der anderen drei Kopplungsarten [Str13]

## 2.4 Unterschied zwischen impliziten und expliziten Verfahren

Die Berechnung einer nichtlinearen Aufgabe kann explizit oder implizit erfolgen. Bei der Berechnung von hochdynamischen Vorgängen mit extrem kurzen Zeiträumen, bspw. Crashesimulationen, werden explizite Verfahren mehr eingesetzt; Implizite Verfahren hingegen werden maßgeblich im Bauingenieurwesen für statische Probleme angewendet [Spu12]. Ein sinnvoll gewählter Zeitschritt soll die Balance zwischen den Kosten für die Rechenzeit auf der einen und den Ansprüchen an Genauigkeit, Stabilität und Solver-Konvergenz auf der anderen Seite halten [Bor14].

In der Abbildung 2.3 ist ein deutlicher Vergleich der beiden Verfahren gezeigt. Grüne halbkreisförmigen Kurven sind die exakten Ergebnisse. Rote Kurven ergeben sich schrittweise durch näherungsweise Berechnung. Jeweils für einen Zeitschritt werden berechnet.

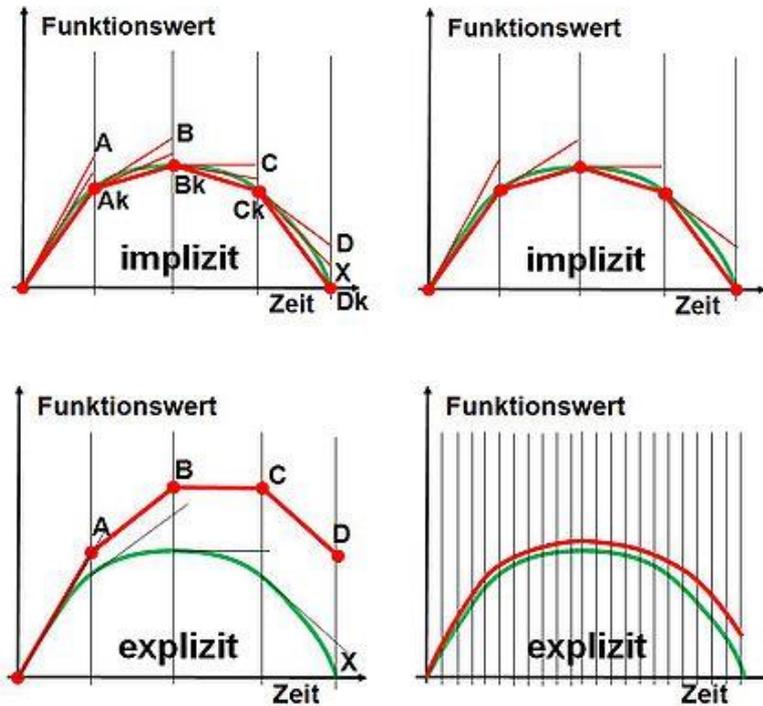


Abbildung 2.3: Vergleich der impliziten und expliziten Verfahren [cae16]

Die oberen gezeigten zwei Abbildungen stellen das implizite Verfahren dar. Durch eine Tangente (dünne rote Linie) im Startpunkt wird zu Startpunkt A geführt.  $A_k$  ist die Lösung, die mit Punkt A abgeschätzt wird. Dann wird A durch Berechnung auf  $A_1$  geführt. Danach wird  $A_1$  durch eine erneute Berechnung zu  $A_2$  geführt, usw. Dieser Vorgang wird nach Kriterium abgebrochen, wenn  $A_i$  gegen  $A_k$  läuft, beziehungsweise  $A_{i-1}-A_i$  gleich null oder ein Grenzwert ist. Für eine Lösung im Punkt  $A_k$  wird  $A_k$  selbst benötigt. Die nächsten Zeitschritte führen zu  $B_k$ ,  $C_k$  und  $D_k$ . Rechnerisch lässt sich der Funktionswert z.B. im Punkt  $B_k$  durch folgende Gleichung

$$F(y(t + dt), y(t)) = 0 \quad (2.1)$$

rechnen. In der Gleichung (2.1) ist  $y(t)$  der Funktionswert im Punkt  $A_k$ .  $y(t + dt)$  ist der Funktionswert im Punkt  $B_k$ . Und  $dt$  ist der Zeitschritt.

Bei der expliziten Berechnung führt der erste Schritt gleich zu Punkt A. Im Vergleich zu impliziten Verfahren wird Funktionswert von Punkt A parallel zur nächsten Tangente weiter zu Punkt B ausgeführt. Hier wird keine Rückrechnung durchgeführt. Die nächsten Schritte führen zu Punkt C und Punkt D. Rechnerisch lässt sich der Funktionswert im Punkt B durch

$$y(t + dt) = G(y(t)) \quad (2.2)$$

ausdrücken.

Die Tabelle 2.1 enthält eine Gegenüberstellung der expliziten und impliziten FEM-Verfahren. Im Rahmen dieser Arbeit berechnen die FEM-Software und die DEM-Software beide explizit, weil es sich um hochdynamische Vorgänge handelt.

Tabelle 2.1: Gegenüberstellung der expliziten und impliziten FEM-Verfahren [Sie15]

	<b>Implizite Verfahren</b>	<b>Explizite Verfahren</b>
Zeitschritt	Frei wählbar	Sehr klein
Initiale Lage- rung	Struktur muss durch Randbedingungen gelagert sein	Keine Fixierung erforderlich
Schnelle Kon- taktänderung	Schwierig, kann zu Konvergenzproble- men führen	Dank sehr kleinen Zeitschritten gut beherrschbar
Rechendauer	Sehr schwer abschätzbar	Gut abschätzbar
Parallelisier- barkeit	Aufwändig	Sehr einfach möglich



### 3 Analytische Betrachtung

Wie in der Abbildung 3.1 dargestellt, trifft eine starre Kugel von der Masse  $m_k$  mit einer Anfangsgeschwindigkeit  $v_0$  auf eine Stelle eines Balkens, dessen anderes Ende festgehalten ist. Um den hier nicht interessierenden Schwerkraft der Kugel in der Richtung der Durchbiegung zu vermeiden, wird der Balken senkrecht angeordnet.

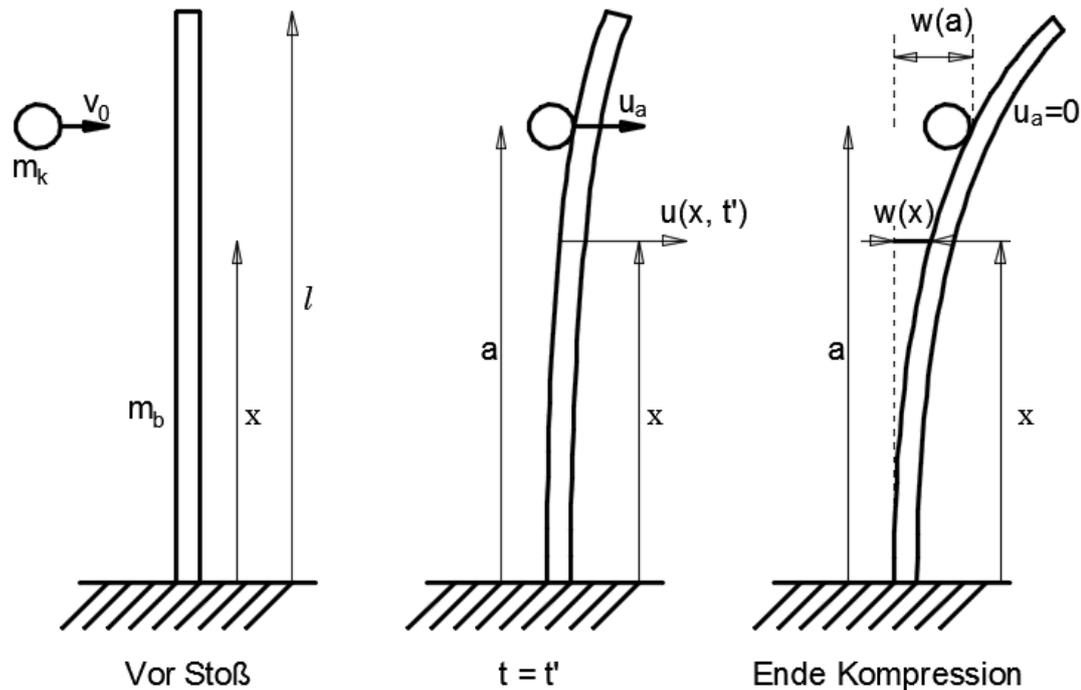


Abbildung 3.1: Kompressionsvorgang

Querstoß auf einen elastischen Balken wird mit einer Annahme [Pös13][Pös28] berechnet, dass sich die Geschwindigkeiten des nach dem Stoß ausschlagenden Stabes ebenso verteilen wie die Durchbiegungen infolge einer statischen Einzellast. Der ganze Stoßvorgang umfasst zwei Phasen: Kompression und Dekompression. Beim Kompressionsvorgang des Balkens werden ein Impulssatz und ein Energieerhaltungssatz verwendet. Die folgenden Darlegungen basieren auf Ausführungen von Gummert [Gum13].

Der Kompressionsvorgang ist wie folgt: Die Kugel trifft auf den Balken und dann verlangsamt; Inzwischen beschleunigt sich der ganze Balken bis die Kugel zu einem Zeitpunkt  $t'$  eine gleiche Geschwindigkeit  $u_a$  wie die Geschwindigkeit der Auftreffstelle des Balkens erreicht. Hierbei werden deswegen noch angenommen, dass die Kugel ein Starrkörper ist, und nicht vorher vom Balken zurückprallt, sondern sich mit ihm weiterbewegt. Nach der Kompression bleibt die Kugel Kontakt mit dem Balken bis sie ihn verlässt, und zwar der Dekompressionsvorgang.

Der Impuls des Systems vor dem Stoß soll gleich dem Impuls zu dem Zeitpunkt  $t'$  sein, als die Kugel mit dem Balken an der Auftreffstelle die gemeinsame Geschwindigkeit  $u_a$  hat:

$$m_k \cdot v_0 = m_k \cdot u_a + \int_{(l)} u(x, t') dm_b. \quad (3.1)$$

In der Gleichung (3.1) ist  $u(x, t')$  die Geschwindigkeit von allen Teilen des Balkens.  $\int_{(l)} u(x, t') dm_b$  ist der Impuls des ganzen Balkens, mit  $dm_b = \frac{m_b}{l} dx$ .

Nach der am Anfang des Kapitels genannten Annahme: das Verhältnis der Geschwindigkeiten des ganzen Balkens und die Geschwindigkeit der Auftreffstelle  $x = a$  ist identisch mit dem Verhältnis der Durchbiegungen des ganzen Balkens und die Durchbiegung an der Auftreffstelle. Die zwei Verhältniszahlen lassen sich durch

$$\zeta(x) = \frac{u(x, t')}{u_a} = \frac{w(x)}{w(a)} \quad (3.2)$$

ausdrücken. Mit  $dm_b = \frac{m_b}{l} dx$  gilt aus der Gleichung (3.1) und der Gleichung (3.2) also der Impulssatz:

$$m_k \cdot v_0 = m_k \cdot u_a + \frac{m_b \cdot u_a}{l} \int_{x=0}^l \zeta(x) dx, \quad (3.3)$$

daraus kommt die gemeinsame Geschwindigkeit

$$u_a = \frac{m_k \cdot v_0}{m_k + \frac{m_b}{l} \int_{x=0}^l \zeta(x) dx} = \frac{m_k \cdot v_0}{m_k + k_1 \cdot m_b}, \quad (3.4)$$

worin die Konstante  $k_1$  durch

$$k_1 = \frac{1}{l} \int_{x=0}^l \zeta(x) dx. \quad (3.5)$$

berechnet werden kann.

Dann lässt sich der Energieerhaltungssatz für die Zeit  $t'$  und für Ende des Kompressionsvorgangs vom Balken ansetzen, wobei der Balken nach der oberen Abbildung 3.1 die größte Durchbiegung  $w(a)$  an der Auftreffstelle hat, als die Geschwindigkeit des Balkens und die Geschwindigkeit der Kugel beide null sind. Daraus kommt der Energieerhaltungssatz:

$$\frac{1}{2} m_k u_a^2 + \frac{1}{2} \int_{(l)} u^2(x, t') dm_b = \frac{1}{2} \cdot c \cdot w^2(a), \quad (3.6)$$

Gemäß der Gleichung (3.2) und  $dm_b = \frac{m_b}{l} dx$  lässt sich

$$\int_{(l)} u^2(x, t') dm_b = \frac{m_b \cdot u_a^2}{l} \int_{x=0}^l \zeta^2(x) dx = k_2 m_b u_a^2 \quad (3.7)$$

erhalten, in der die Konstante  $k_2$  durch

$$k_2 = \frac{1}{l} \int_{x=0}^l \zeta^2(x) dx. \quad (3.8)$$

auch bestimmt wird.

Die größte Durchbiegung an der Auftreffstelle  $x = a$  lässt sich mit den Gleichungen (3.6), (3.7) und (3.4)

$$w(a) = \sqrt{\frac{m_k}{c} \frac{\sqrt{1 + \frac{m_b k_2}{m_k}}}{1 + \frac{m_b k_1}{m_k}}} v_0 \quad (3.9)$$

erhalten, wobei die Federkonstante durch

$$c = \frac{3EI}{l^3} \quad (3.10)$$

gegeben wird [Ric08].

Für einen Kragbalken ist die Biegelinie des ganzen Balkens infolge einer statischen Einzelkraft  $F$  laut Romberg [Rom11] durch

$$w(x) = \frac{F}{6EI} lx^2 \left(3 - \frac{x}{l}\right) \quad (3.11)$$

gegeben. Die Durchbiegung an der Stelle  $x = a$  ist durch

$$w(a) = \frac{F}{6EI} la^2 \left(3 - \frac{a}{l}\right) \quad (3.12)$$

gegeben. Deswegen ist das Verhältnis also

$$\zeta(x) = \frac{w(x)}{w(a)} = \frac{1}{a^2(3l-a)} (3x^2l - x^3), \quad (3.13)$$

daraus mit der Gleichung (3.5) und (3.8) kommen

$$k_1 = \frac{3l^3}{4a^2(3l-a)}, \quad (3.14)$$

und

$$k_2 = \frac{33l^6}{35a^4(3l-a)^2}. \quad (3.15)$$

Falls die Masse der Kugel sehr groß im Vergleich zur Balkenmasse ist, kann ein Energieerhaltungssatz:

$$\frac{1}{2} m_k v_0^2 = \frac{1}{2} \cdot c \cdot w^2(a) \quad (3.16)$$

auch näherungsweise gesetzt werden, und daraus folgt eine angenäherte größte Durchbiegung an der Auftreffstelle:

$$w(a) = \sqrt{\frac{m_k}{c}} v_0. \quad (3.17)$$

Wenn die Auftreffstelle genau auf der Seite  $a = l$  ist, für  $m_b = m_k$  beträgt ein Fehler 19%; Für  $m_b = \frac{1}{10} m_k$  ist der Fehler nur 2,5%.

Der Grund des Fehlers liegt darin, dass der Energieerhaltungssatz nicht gut für Kompressionsvorgang geeignet ist. Der Fehler findet am Anfang des Kompressionsvorgangs bis die Kugel und die Auftreffstelle des Balkens beide die gemeinsame Geschwindigkeit  $v'$  hat statt. Der Impulserhaltungssatz wird so benutzt:

$$m_k \cdot v_0 = m_k \cdot u_a + \int_{(l)} u(x, t') dm_b. \quad (3.1)$$

Aber laut Energieerhaltungssatz:

$$\frac{1}{2} m_k v_0^2 = \frac{1}{2} (m_k + m_b) u_a^2 + \frac{1}{2} \cdot c \cdot w^2(a, t'), \quad (3.19)$$

lässt sich hier ein Unterschied beider Fälle sehen. Wenn  $m_b$  im Vergleich zu  $m_k$  sehr groß ist, wird die gemeinsame Geschwindigkeit  $u_a$  aus Impulssatz viel kleiner als die aus Energiesatz, denn die gemeinsame Geschwindigkeit  $u_a$  ins Quadrat erhoben wird.

## 4 Kopplungsmethode

LIGGGHTS bietet eine sehr gute Funktionalität, dass alle Informationen der Kugel nach Ausführung von LIGGGHTS für jede Ausführung in eine binäre Datei als Restart-File gespeichert werden kann. Für jedes Inkrement wird LIGGGHTS gestartet und entsprechende Restart-Datei erzeugt. LIGGGHTS liest das Restart-File aus und läuft für nächste Ausführung weiter ohne Informationen zu verlieren, wie Koordinaten, Rotationen, Geschwindigkeiten usw.

Darüber hinaus darf ihre Geometrie (.stl Datei) aktualisiert werden, während LIGGGHTS sich das Restart-File nimmt, was RADIOSS nicht kann. Deswegen muss die Geometrie in Starter-File für jedes Inkrement durch Pythonskript aktualisiert werden. Aber die Geschwindigkeit des Balkens wird nach Ausführung von RADIOSS verloren. Deswegen ist LIGGGHTS sehr gut für Kopplung geeignet. RADIOSS benötigt Programmierarbeit.

### 4.1 Vorstellung der Methode

Ziel der Kopplung ist Errichtung einer Umgebung für DEM-Software LIGGGHTS und FEM-Solver RADIOSS, wo die beide Softwares abwechselnd laufen können. In der Abbildung 4.1 ist ein Flussdiagramm der Kopplung dargestellt.

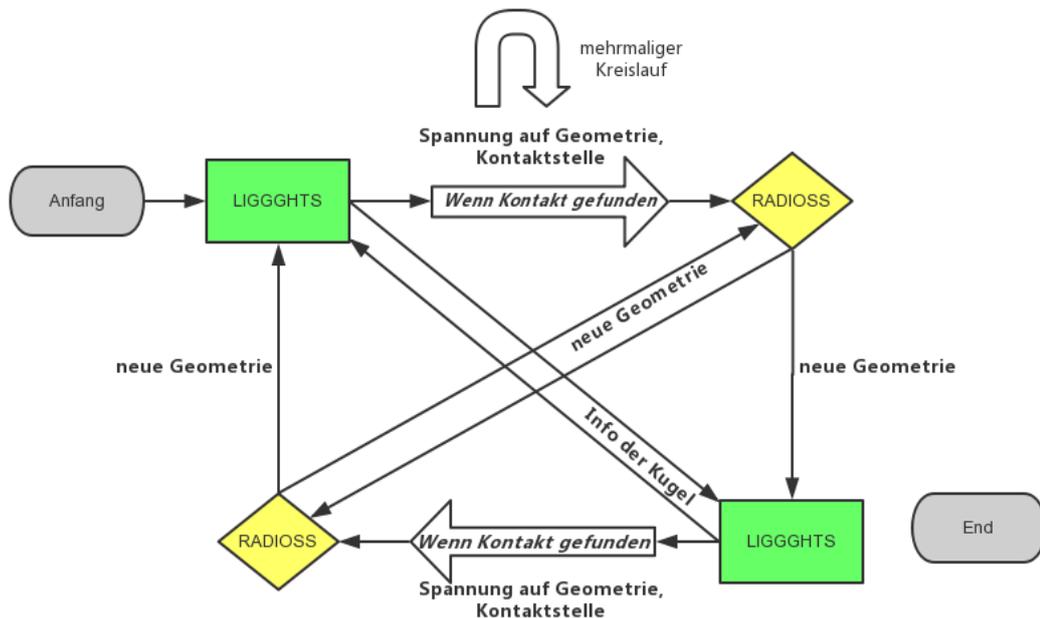


Abbildung 4.1: Flussdiagramm der Kopplung

Ein einfaches Beispiel wird simuliert, dass eine Kugel auf einen elastischen Balken fällt, der Balken sich durchbiegt und die Kugel wieder hochdrückt. Mit Altair HyperMesh wird die Geometrie des Balkens als 2D-Schalenmodell gezeichnet und vernetzt. Die Vernetzungsdaten brauchen LIGGGHTS und RADIOSS beide.

Die Kopplung fängt immer mit LIGGGHTS an. Informationen der Kugel werden in LIGGGHTS ständig gespeichert. Ohne Kontakt zwischen der Kugel und dem Balken rechnet immer nur LIGGGHTS. Nachdem ein Kontakt stattfindet, schreibt LIGGGHTS eine Spannung auf der vernetzten Geometrie und die Kontaktstelle der Spannung aus. RADIOSS liest die Spannung und die Kontaktstelle aus und beginnt zu rechnen. Nach Durchlauf von RADIOSS entsteht eine neue Geometrie des Balkens, weil er mit der ganzen Verschiebung aus Grund der Kugel verformt ist. RADIOSS aktualisiert die Geometrie nicht nur für LIGGGHTS, sondern auch für sie selbst.

Dann ist LIGGGHTS wieder dran mit der neuen Geometrie und den alten Informationen der Kugel zu rechnen. Die Informationen der Kugel bekommt LIGGGHTS von sich selbst. Bald findet Kontakt wieder statt. LIGGGHTS schreibt eine neue Spannung und gegebenenfalls eine neue Kontaktstelle aus. Danach folgt RADIOSS mit der neuen Geometrie zu rechnen, die durch die vorherige Runde von RADIOSS aktualisiert wird. Als Nächstes wiederholen beide Softwares und so weiter und so fort. Nach einer gewisser Zahl der Wiederholungen geht die Kopplung zu Ende.

## 4.2 Umsetzung

### 4.2.1 Vorbereitung

Die Kopplung basiert auf Programmiersprache Python. Das Pythonskript enthält viele Vorarbeiten eines Mitarbeiters vom Institut für Maschinenelemente und Konstruktionsfertigung. Es wurde manuell getestet, bisher funktionierte die Kopplung größtenteils händig, d.h., es existierte die Hauptstruktur mit den Funktionen des Ausschreibens von Ergebnisfiles, das Einlesen dieser, sowie dem Aufruf des Solvers, bevor ich diese Arbeit übernehme. Es fehlt dem Skript noch eine völlige Automatisierung des Skriptes und ein paar Schritte, wie z.B. Einschreiben der Kontaktstelle der Kugel auf den Balken, Aktualisierung der Vernetzungskoordinaten des Balkens, Spannungsanpassung und Anpassung der Dateiformate, sowie Fehlerbehebung. Übrigens werden die Befehle in Skripten in Kursivschrift in der vorliegenden Masterarbeit geschrieben.

Im Pythonskript wird erst Importieren von einigen hilfreichen Modulen vorgenommen, die später im Skript verwendet werden und die Programmierung viel erleichtern können. Innerhalb von Python werden insgesamt 6 Funktionen aufgerufen, was zum Block (LiggghtsRadioss\_func.py) ansteht. Dann kommen ein paar Einstellungen für die Kopplung, wie Zeitschritt von LIGGGHTS *dt\_dem*, Schrittweite *dt\_dump* (Anzahl an *dt\_dem*), Schleifenanzahl *cycles*. Im Kapitel 4.2.2 werden sie alle erklärt. Alle dürfen geändert werden, bevor das ganze Skript zu laufen beginnt. Die Einstellungen für das Simulationsobjekt, wie Dicke des Balkens, Radius der Kugel und Anfangsgeschwindigkeit der Kugel, werden in ihren Inputdaten manuell geändert.

Alle Programme werden auf Linux durchgeführt. Die Geometrie des Balkens kann in HyperMesh vernetzt werden und das Starter-File für RADIOSS und .stl Datei für LIGGGHTS erzeugt; RADIOSS Block 140 wird immer als Solver ausgewählt. Das Starter-File lässt sich auch

als ein Solver Deck in HyperMesh importieren. HyperView ist eine der verwendeten Nachbearbeitung und Visualisierungsumgebungen.

## 4.2.2 Zeiteinstellung

In der Abbildung 4.2 ist die Zeiteinstellung der Kopplung dargestellt. Der kleine Abschnitt ist Zeitschritt ( $dt_{dem}$ ). Der große Abschnitt ist Inkrement ( $dt_{dump}$ ). Hier lässt sich verschiedenen Zeitschritt eingeben. Der interne Zeitschritt innerhalb beider Softwares wird separat durch  $dt_{dem}$  und  $dt_{fem}$  im Python-Skript dargestellt. Die Schrittweite für eine Kopplung ist z.B. 10. Es ist keine Zeitspanne, sondern nur eine Zahl. Ein Inkrement ist in diesem Fall ein zehnfacher Zeitschritt, also eine Zeitspanne.

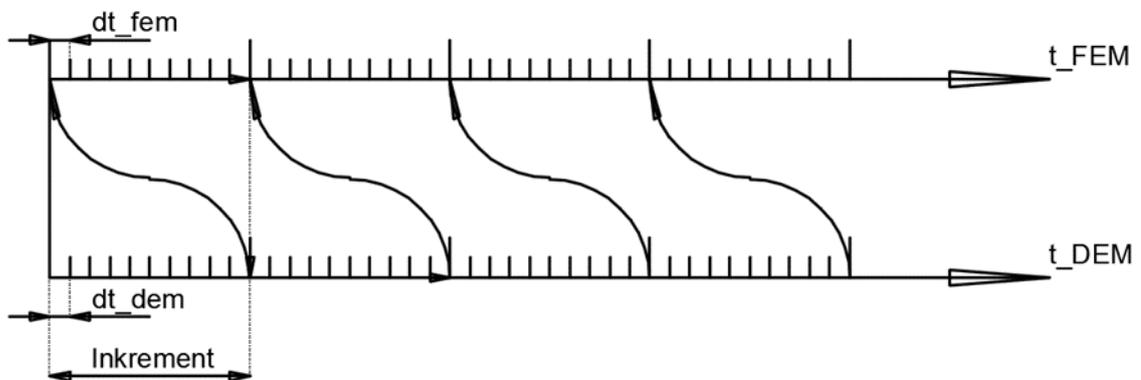


Abbildung 4.2: Zeiteinstellung der Kopplung

Es fängt bei dem Null auf der unteren Zeitschiene an. 10 Mal berechnet die DEM in kleinen Zeitschritten von vorne nach hinten. Nach jeweils 10 Mal Wiederholungen für die DEM, geht es dann nach oben zum Zeitpunkt Null in der FEM, und laufen 10 Mal Wiederholungen durch. Sobald alle 10 kleinen Schritte berechnet werden, geht zurück nach unten und läuft wieder 10 Mal auf der Zeitschiene für die DEM weiter. Das heißt, die DEM berechnet zuerst, dann berechnet die FEM und danach wieder unten zu der DEM dann wieder hoch usw.

Für jedes Inkrement werden Ergebnisse ausgeschrieben. Es wird in einem Inkrement zweimal berechnet, einmal DEM, und danach FEM. Und dann zurück zu der DEM, rechnet nochmal ein Inkrement. Die Anzahl aller gesamten Inkremente ist *cycles*. Hier in der Abbildung beträgt die Anzahl 4.

## 4.2.3 Datenfluss

In der Abbildung 4.3 ist ein Datenflussdiagramm dargestellt. Im Prinzip bearbeitet und schreibt das Pythonskript Daten in die Inputdatei von LIGGGHTS und RADIOSS. Die Inputdatei von LIGGGHTS heißt im Folgenden Liggghtsskript. Zur Ausführung einer Simulation braucht RADIOSS zwei Inputdaten: ein Starter-File und ein Engine-File. In dem Engine-File wird die Einstellung der Simulationsparameter vorgenommen. Nachdem die Kopplung gestartet wird, verändert sich das Engine-File nicht mehr, deshalb wird es in der Abbildung nicht erwähnt. Auf

Einschreiben der Daten in die zwei Files von RADIOSS muss geachtet werden, weil sie nur die Daten lesen können, die sich auf geregelten Ort im Skript befinden.

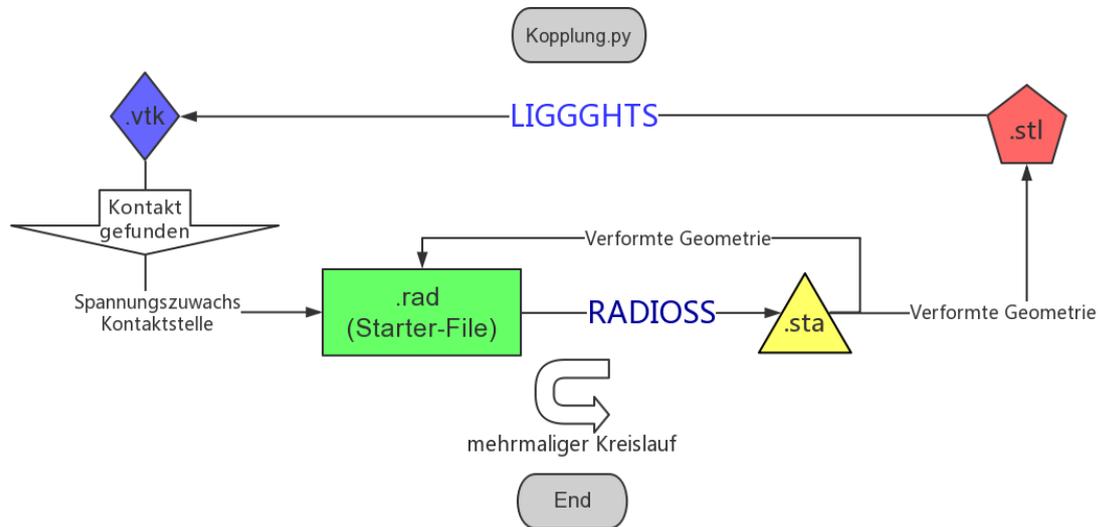


Abbildung 4.3: Datenflussdiagramm

LIGGGHTS ist immer der erste Start. LIGGGHTS wird für jedes Inkrement ( $dt_{dem}$  multipliziert mit  $dt_{dump}$ ) in der Schleife ausgeführt und bereitet Restart neu vor. LIGGGHTS läuft zuerst für ein Inkrement durch, damit die Kugel im Simulationsraum erzeugt wird. Nach Durchlauf von LIGGGHTS für jedes Inkrement wird eine .vtk Datei ausgeschrieben und in den Ordner /post/ gespeichert. Wenn kein Kontakt zwischen der Kugel und dem Balken gefunden wird, läuft immer nur LIGGGHTS für nächstes Inkrement weiter.

In der .vtk Datei lässt sich die Spannung wie eine Tabelle in jedem einzelnen Vernetzungselement auslesen. Vor Kontakt zwischen der Kugel und dem Balken ist die Spannung überall null. In einer Stelle gibt es erst nach Kontakt eine Spannung, an der Stelle die Partikel den Balken berührt hat, und eine Spannung erzeugt. Die aus Partikeln auf die Geometrie erzeugte Spannung wird ausgelesen und in einen Spannungszuwachs bearbeitet. Der Spannungszuwachs wird dann als eine Tabelle in eine .rad Datei (Starter-File von RADIOSS) übertragen. Die Bearbeitung der Spannung wird später im Kapitel 4.3.1 erklärt.

Starter-File speichert im Prinzip die Materialeigenschaft, die Geometrie, die Spannung und ihre wirkende Stelle sowie die Randbedingungen. Eine Funktion von Pythonskript schreibt den Spannungszuwachs und die Knoten einer dreieckigen Zelle in das Starter-File, an welche Stelle die Kugel senkrecht unterfällt und aufkommt. Jedes Inkrement sieht der Spannungszuwachs anders aus. RADIOSS nimmt sich den Spannungszuwachs, und wird gestartet. Für ein Inkrement berechnet RADIOSS die Verformung des Balkens, die durch den Spannungszuwachs verursacht wird, und eine .sta Datei (RADIOSS ASCII Output State-File) erzeugt.

Die .sta Datei stellt die Knoten der Vernetzung und ihre Koordinaten dar, und zwar eine neue vernetzte Geometrie. Am Anfang ist die Verformung mancher Knoten sehr klein, aber numerisch nicht ganz null, z.B.  $1e-43$  mm. Über eine Funktion innerhalb Pythonskript liest die neue Geometrie aus und erzeugt daraus noch eine Geometrie (.stl Datei) für LIGGGHTS.

Die .stl Datei ist hier eine reine Flächengeometriedatei in ASCII-Format. Sie beinhaltet die Beschreibung durch Dreiecksfacetten. Sie befindet sich im Ordner /data/. LIGGGHTS braucht immer eine .stl Datei in ASCII-Format zu rechnen. Über eine andere Funktion von Pythonskript wird die verformte Geometrie noch für Starter-File aktualisiert. Jetzt ist die erste Schleife beendet.

LIGGGHTS wird wieder gestartet, liest die verformte Geometrie aus, läuft für nächstes Inkrement, bereitet Restart neu vor und wartet bis nächster Kontakt im weiteren Inkrement stattfindet.

## 4.3 Variantenvergleich

### 4.3.1 Spannungszuwachs

RADIOSS verliert nach Durchführung nicht nur die Geschwindigkeit des Balkens, sondern auch die Elastizität zum Rückprall. Deshalb wird der Spannungszuwachs in das Starter-File übertragen. Der Spannungszuwachs erfolgt durch

$$\Delta\sigma = \sigma_S - \sigma_F, \quad (4.1)$$

worin  $\sigma_S$  eine Stoßspannung und  $\sigma_F$  eine Federspannung ist. Die Stoßspannung wird durch die Spannung auf der Kontaktstelle repräsentiert, die aus .vtk Daten von LIGGGHTS nach Kontakt zwischen der Kugel und dem Balken neu ermittelt wird. Und die Federspannung bildet die Spannung für vorheriges Inkrement ab. Das heißt, die Stoßspannung ist immer der Federspannung ein Inkrement voraus.

Der Gedanke ist wie Diskretisierung des Kompressionsvorgangs. Es wird angenommen, dass das Inkrement sehr klein ist. Sobald ein Kontakt zwischen der Kugel und dem Balken entsteht, erfolgt zuerst auf der Kontaktstelle eine Stoßspannung. Danach erzeugt der Balken erst die Federspannung durch seine Elastizität. Der Gedanke lässt sich mit .vtk Datei und Starter-File durch Pythonskript wahrnehmen. Der Spannungszuwachs ist die neue ausgelesene Spannung aus aktueller .vtk Datei minus die vorherige. Für jedes Inkrement nach Kontakt wird der in das Starter-File aufgetragene Spannungszuwachs geändert.

Außerdem muss der Spannungszuwachs im Starter-File als eine Funktion (PLOAD) definiert werden. Obwohl der Spannungszuwachs in der Praxis für ein Inkrement abfällt oder ansteigt, wird er als ein Wert angesehen, weil das Inkrement sehr klein ist. Voraussetzung des Gedankens ist, dass es um ein Modell mit linearem Materialgesetz geht.

Was umtreibt ist, dass RADIOSS in der Kopplung nicht nur die Geschwindigkeit des Balkens nach Kontakt, sondern auch die Elastizität zum Rückprall verliert. Im Vergleich zu LIGGGHTS fehlt RADIOSS die Funktionalität, dass die Spannung und die Kontaktstelle innerhalb binärer

Restart-Files aktualisierbar sind, trotzdem RADIOSS auch aufrufbare Restart-Files speichert. Sobald eine Spannung oder eine Kontaktstelle einmal in das Restart-File angetragen wird, darf die Spannung oder die Kontaktstelle nicht mehr für nächstes Inkrement verändert werden. Bevor der Gedanke auf Spannungszuwachs plus Aktualisierung der Geometrie von Starter-File gemacht wurde, hat die Simulation immer zu kleinere Durchbiegung als die analytische Lösung. Zwei andere Gedanken wurden damals wie folgt versucht, um das Problem zu lösen.

### 4.3.2 Auferlegung von Verschiebung auf Balken

Ohne Spannungszuwachs plus Aktualisierung der Geometrie in das Starter-File einzuschreiben, wurde ein anderer Gedanke versucht. Der Gedanke wird so abgebildet: die Verschiebung eines Knoten vom Vernetzungselement wird ausgelesen, wo Kontakt in einem Inkrement stattfindet; Dann wird diese Verschiebung dem Knoten als eine Randbedingung in RADIOSS (IMPDISP) für nächstes Inkrement auferlegt. Die Verschiebung vom ersten Inkrement soll sowieso in erster Hälfte der Kopplung kleiner als die vom weiteren Inkrement sein. Diese Verschiebung sollte als eine zurückhaltende Kraft wirken.

Verschiedene Anteile und Dauer sowie Zeitspanne der auferlegten Verschiebungen wurden probiert. Die simulierte Durchbiegung war immer viel größer als der analytische Wert. Übrigens sieht der Balken nach Simulation wie eine Welle aus. Tatsächlich wirkt diese Verschiebung als ein Stoß auf das Vernetzungselement, dadurch der Balken belastet wird. Die erwünschte zurückhaltende Kraft wirkt, aber nicht auffällig. Jedoch inspiriert dieser Gedanke zu dem Gedanke auf Spannungszuwachs.

### 4.3.3 User Window Subroutine

Der Gedanke auf User Window Subroutine ist eine schwierige Wahl. Zunächst soll eine Subroutine mit einer anderen Programmiersprache Fortran entwickelt werden, die die Spannung innerhalb binärer Restart-Files von RADIOSS ändern kann. Dann kompiliert GFortran die Subroutine. Schließlich wird RADIOSS mit einer installierten UserLibrary ausgeführt. Vorteil des Gedankens ist, dass RADIOSS nicht neu gestartet werden muss, damit sie immer das Restart-File auslesen kann. Keine Informationen über den Balken lassen sich dann verloren. In einem Monat wird aber der Gedanke wegen zu hohen Arbeitsaufwands aufgegeben.

## 4.4 Einstellungen von LIGGGHTS

Um eine Simulation durchzuführen benötigt LIGGGHTS ein Inputskript, in dem die Einstellung der Simulationsparameter vorgenommen wird. LIGGGHTS lässt sich genauer einstellen. Im Liggghtsskript wird die interne Genauigkeit von  $1e-20$  eingestellt. Aber aus Voruntersuchungen am Institut ist bekannt, dass der sinnvollste Wert des internen Zeitschritts in gewählter DEM-Umgebung zwischen  $1e-7$  und  $1e-6$  liegt.  $1e-20$  ist tatsächlich genauer. Im Folgenden werden ein paar Einstellungen von LIGGGHTS verdeutlicht, was mechanisch wichtig ist.

## Dämpfungskonstante und Reibungskonstante

Im Liggghtsskript stehen die Dämpfungskonstante und Reibungskonstante zur Einstellung. Weil es Kontaktmöglichkeit von zwei Materialien gibt, müssen vier Werte von Reibungskoeffizient oder Dämpfungskonstante wie eine zwei-mal-zwei Tabelle angegeben werden. Die Kugel ist aus PET hergestellt, und der Balken aus Stahl.

Wie in der Tabelle 4.1 dargestellt, sind der Wert oben rechts und der Wert unten links identisch, weil PET auf Stahl trifft und Stahl auf PET das gleiche ist. Der Reibungskoeffizient (unten rechts) von zwei gleichen Materialien lässt sich auch bestimmen und angeben, was ein relativ sinnloser und rein erfundener Wert ist, weil kein Kontakt zwischen Stahl und Stahl stattfinden wird.

Tabelle 4.1: Definition der Reibungskoeffizienten von 2 Materialien in LIGGGHTS

0.55	0.175
0.175	0.32

Wenn es 3 Materialien gibt, ist es ein drei- mal-drei Matrix. Literaturwerte lassen sich bei realistischer experimentalen Fall verwenden. In dieser Arbeit dürfen die Reibungskoeffizienten alle null und die Dämpfungskonstante alle eins sein, weil die Reibung und die Dämpfung in der analytischen Lösung nicht berücksichtigt werden können.

## Kontaktmodell

Als Kontaktmodell in LIGGGHTS wird *gran model hooke* eingestellt, und zwar ein linearelastisches Materialgesetz. Andere Variante kann z.B. Hertzscher Kontaktmodell sein, das nicht linearelastisch ist und quadratischen Ansatz benutzt. LIGGGHTS berechnet intern nur mit Granulaten, deshalb ist die Geometrie des Balkens wie eine riesengroße Kugel. Die Geometrie besteht aus Vernetzungen, und jedes einzelnes Vernetzungselement wird repräsentiert durch eine Kugel mit einem unendlichen Radius.

## Gravitationskraft

Der Befehl *gravity* gibt die Richtung der Gravitationskraft an. Um den nicht interessierenden Schwerkraft der Kugel in der Richtung der Durchbiegung zu vermeiden, ist die Erdbeschleunigung sehr klein ( $0.0001 \text{ m/s}^2$ ) und in anderer Richtung (*minus X Richtung*) eingestellt. Denn Arbeit nicht an dem Balken in der Richtung der Durchbiegung verrichtet werden darf, um den Impulssatz in dieser Richtung zu einzusetzen, siehe Kapitel 3.

## Einfüge-Ort

Was wichtiger ist Definition des Einfüge-Ortes in LIGGGHTS. Das Granulat muss an einem Raum generiert werden, wo sie durch den Befehl *region* eingefügt werden. Eine Sphäre, ein Würfel, eine Pyramide oder ein Zylinder lassen sich alle als Einfüge-Ort definieren. Hier wird

das Granulat an einer kleineren Sphäre platziert, an der Stelle das Granulat landet. Das Granulat liegt drinnen bei der Mittelpunkt innerhalb der Sphäre fest. Vier Ziffern werden hier beim Befehl benötigt, die  $x$   $y$   $z$  (Sphärenzentrum) und  $r$  (Sphärenradius) repräsentieren. Das Granulat wird in dem Bereich eingefügt. Je feiner die Sphäre erwählt ist, desto genauer wird das Einfüge-Ort positioniert. Anderes Granulat lässt sich auch in zweitem Einfüge-Ort erzeugen. Das erzeugte Granulat fällt unter und landet auf den Balken und gibt eine gewisse Spannung.

### **LIGGGHTS Post-processing**

Mit Durchführung von `lpp dump*` im Terminal von den `/post/` Ordner, lassen sich `reib*.vtk` Daten mit einem kleinem Programm LIGGGHTS Post-processing LPP ausschreiben, die ParaView aufrufen kann. LPP braucht nur einige Sekunden durchzulaufen. Die ganzen Daten in dem Ordner lassen sich durch ParaView visualisieren. Wenn die Kugel unterfällt und auf die Geometrie drückt, lässt sich den Ort zeigen, wo die Spannung erzeugt wird. Übrigen lässt sich authentische Animationen des Stoßvorgangs sehen.

LIGGGHTS schreibt für jedes Inkrement noch `.stl` Daten aus und speichert sie in den Ordner `/post/` auf. Solche `.stl` Daten sind anders als die oben viel erwähnte Flächengeometriedatei des Balkens im Ordner `/data/`. Von diesen `.stl` Daten lassen sich Durchbiegungen einzelner Knoten auslesen. LIGGGHTS schreibt für jedes Inkrement noch eine `.reib` Datei aus, die den Zustand der Partikel beschreibt, was sich auch in dem Ordner `/post/` befindet. Kleine separaten Codes werden mit Python entwickelt, damit Informationen wie Durchbiegungen eines Knotens und Geschwindigkeiten der Kugel schnell ermittelt werden können. Im Anhang lassen sich die Codes auch finden.

## **4.5 Einstellungen von RADIOSS**

In RADIOSS gibt es Starter-File (`<prefix>_0000.rad`) und Engine-File (`<prefix>_0001.rad`). Das Engine-File (`v5_0001.rad`) ist geregelt, wie lange RADIOSS rechnen wird, welche Daten ausschreiben sollen usw. RADIOSS nutzt im Pythonskript dasselbe Engine-File zu rechnen. Im Prinzip ist das Engine-File eine Einstellungsdatei. Starter- und Engine- Files sind auch im Anhang beigefügt.

## 5 Simulationsstudien

Im Folgenden werden fünf Parameter jeweilig untersucht, welchen Einfluss jeder Parameter auf die Abweichung der Durchbiegung aus der analytischen Berechnung und der Simulation hat. Die Abweichung wird prozentual wie der relative Fehler angegeben. Die ergibt sich aus

$$f_{REL} = \frac{y_a - y_s}{y_a}, \quad (4.1)$$

wobei  $y_a$  die analytische Lösung und  $y_s$  das Simulationsergebnis der Durchbiegung ist.

Die 5 Parameter sind der Zeitschritt, die Schrittweite der Kopplung, die Dicke des Balkens, sowie der Radius und die Anfangsgeschwindigkeit der Kugel. Es handelt sich um 5 Liniendiagramme, als ein Parameter untersucht wird, mit andere vier festgehalten. Auf ihrer vertikalen Achse ist immer die Abweichung zwischen Analyse und Simulation in Prozent aufgetragen. Im Anhang sind die berechneten und simulierten Werte tabellarisch aufgeführt.

### 5.1 Ausgangsmodell

#### 5.1.1 Geometrie des Ausgangsmodells

Standardmäßig sind der Zeitschritt und die Schrittweite der Kopplung vorgegeben. Der Zeitschritt ist  $1e-7$  Sekunden, und die Schrittweite ist 10, was das Inkrement  $1e-6$  Sekunden versetzt wird. Das Ausgangsmodell wird genau wie das in der analytischen Berechnung umgesetzte Beispiel überlegt. Eine Kugel von dem Radius  $r=0.005$  m trifft mit der Anfangsgeschwindigkeit  $v_0=20$  m/s auf die Stelle  $a=1.86$  m eines Kragbalkens, dessen Querschnittabmessungen viel kleiner als seine Länge sind. Ein Ende von dem Kragbalken ist  $1.96$  m weit von der Lagerung auf der anderen Seite. Seine Zeichnung ist in der Abbildung 1 dargestellt. Die Kugel ist aus PET hergestellt, und der Balken aus Stahl. Der Durchmesser einer PET-Kugel beträgt 3 mm, und die Dicke des Balkens ist 5 mm.

Ein kleiner unabhängiger Code wird mit Python entwickelt, damit das analytische Ergebnis schnell ermittelt werden können. Andere Abmessungen, die in der analytischen Berechnung benötigt werden, lassen sich in dem Code im Anhang oder in der Abbildung 5.1 finden.

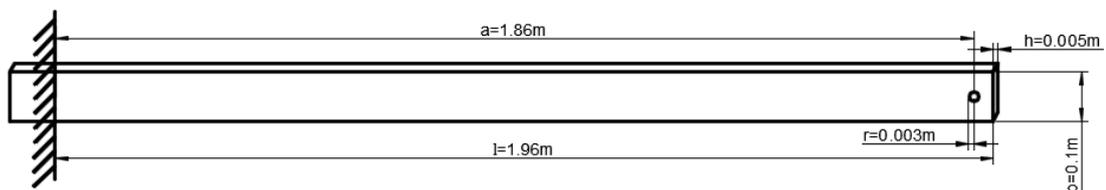


Abbildung 5.1: Zeichnung des Balkens und der Auftreffstelle

Der Balken wird als eine schmale Schale durch HyperMesh vernetzt. Er enthält insgesamt 308 Dreieckselemente. Die Auftreffstelle des Stoßes an den vernetzten Balken ist ein Dreieckselement (id 107), dessen Knoten jeweilig 181, 180 und 186 sind, siehe Abbildung 5.2. Näherungs-

weise wird nur die maximale Durchbiegung des Knotens 180 gesucht. Das Auftreffelement ändert sich im ganzen Stoßvorgang nicht. Die Durchbiegungen des Knotens 180 werden durch einen anderen kleinen Python-Code aus .stl Daten im Ordner /post/ ausgelesen und verglichen. Daraus kommt die maximale Durchbiegung des Knotens 180 im Stoßvorgang. Sie findet nicht am Ende des Stoßvorgangs statt, sondern in seiner Mittel, weil die Kugel noch Kontakt mit dem Balken hat, während die beiden zurückprallen. Die maximale Durchbiegung des Knotens 180 wird als das simulierte Ergebnis der größten Durchbiegung an der Auftreffstelle angesehen.

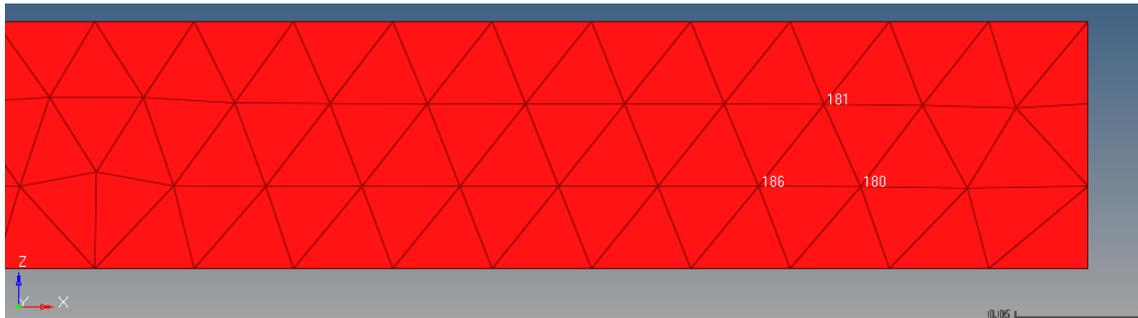


Abbildung 5.2: Kontaktstelle an den Balken

### 5.1.2 Abweichung zwischen Analyse und Simulation

Die analytische Lösung der größten Durchbiegung beträgt  $1.560699e-4$  m in dem Fall, während das simulierte Ergebnis  $1.11946e-4$  m beträgt. Die Abweichung von dem Ausgangsmodell nach der analytischen Berechnung beläuft sich auf rund 28.27%. Mit einer feineren Vernetzung lässt sich die Abweichung ein bisschen näher liegen, weil ein Knoten wahrscheinlich näher von Auftreffstelle gefunden werden kann.

Weil in jeder Kopplung eine Geschwindigkeit des Balkens vernachlässigt wird, ist das kleinere simulierte Ergebnis gerecht, was einer der unvermeidbaren Mängel bei der Kopplung ist. Je größer die Kopplungszahl bei einer Kopplung ist, desto mehr Geschwindigkeiten des Balkens vernachlässigt werden, z.B. bei einer kleineren Schrittweite oder einem dünneren Balken.

Der andere große Mangel bei der Kopplung ist Rückprall des Balkens, was Software-Fehler von LIGGGHTS verursachen kann. In manchen Inkrementen ist der in das Starterfile gegebene Spannungszuwachs viel größer als null, verformt sich der Balken so groß in die Richtung der fallende Kugel, sodass LIGGGHTS nicht das Verhalten des Balkens nicht verstehen. Manchmal kann die Kugel in den Balken einbetten und intensiv mit dem Balken schwingen, oder sogar den Balken durchdringen.

### 5.1.3 Durchlauf der Verschiebungen und Spannungszuwachs

Die Abbildung 5.3 zeigt den Verlauf der Verschiebung an der Auftreffstelle. Nach Kontakt steigt die Verschiebung stetig an bis ein maximaler Wert in der Mittel und dann fällt symmetrisch ab. Der Abstand zwischen jedem Punkt ist gleich das Inkrement, an der Stell  $1e-6$  Sekun-

den. Die X-Achse enthält 300 Inkremente. Nach erstem Kontakt wird in jedem Inkrement danach einmal Kontakt auftreten. Bei anderem Modell ist das nicht so.

Die Abbildung 5.4 verdeutlicht den Verlauf von Differenz zwischen nebeneinander stehenden Verschiebungen. Ihre Erscheinung ist ähnlich wie die Abbildung 5.3, außer die Verschiebungsdifferenz sich auf und ab wie ein Sägezahn bewegt. Das liegt darin, dass der Wert der Spannungszuwachs sich auch auf und ab ändert, weil die Verschiebung und der Spannungszuwachs ein fast lineares Verhältnis haben, wie in der Abbildung 5.5 dargestellt wird.

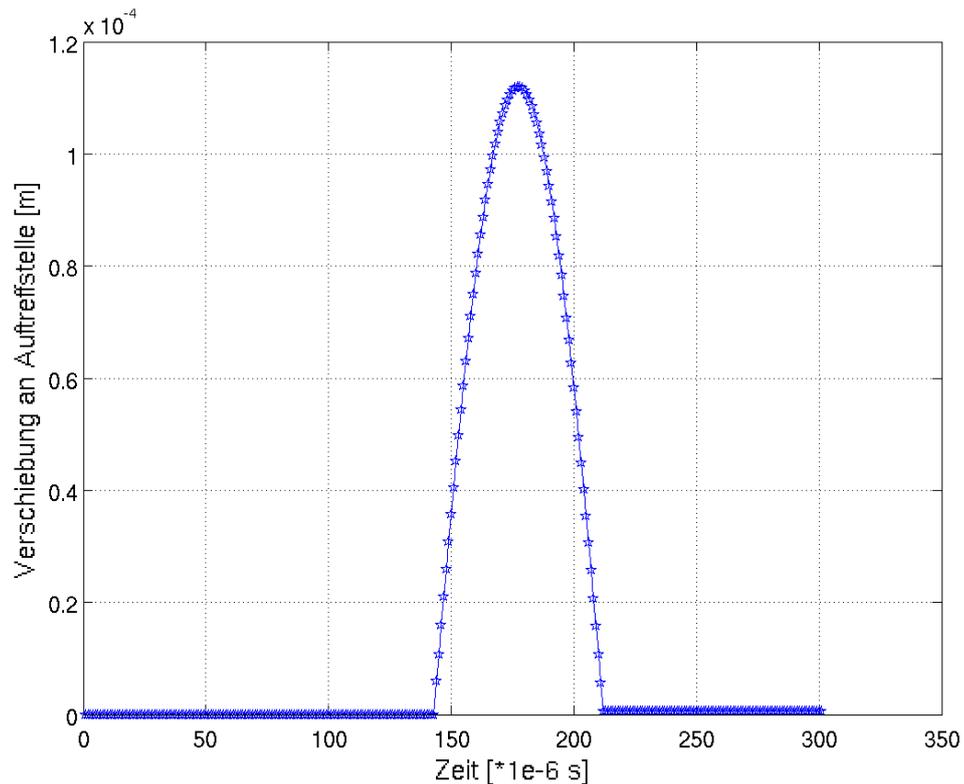


Abbildung 5.3: Durchlauf der Verschiebungen

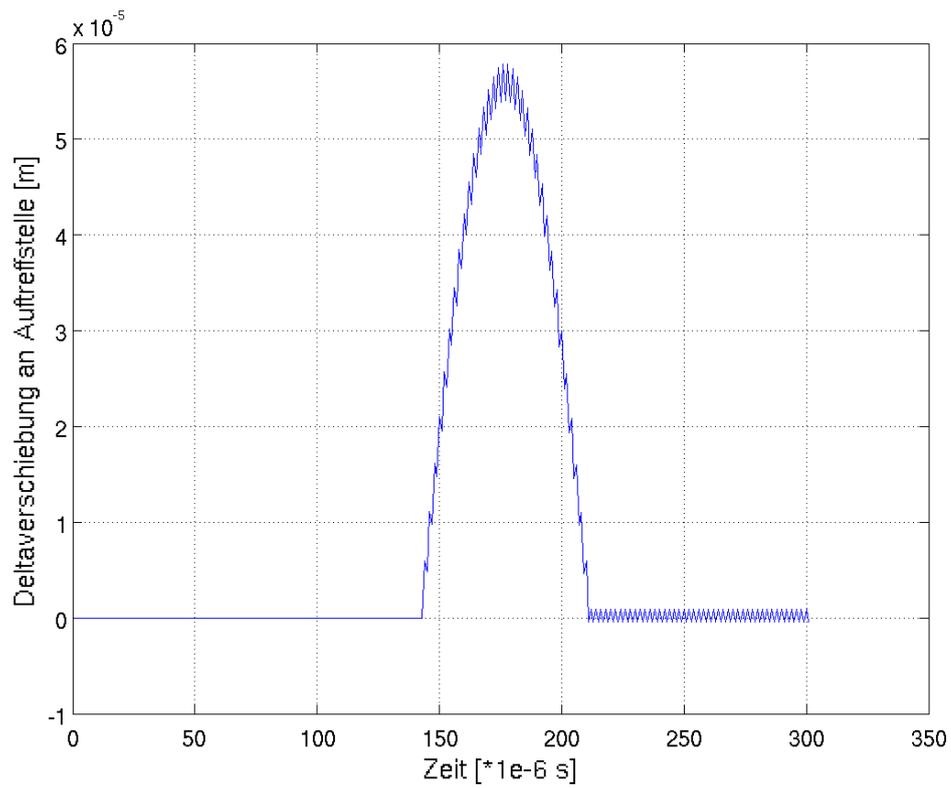


Abbildung 5.4: Verschiebungsdifferenz an der Auftreffstelle

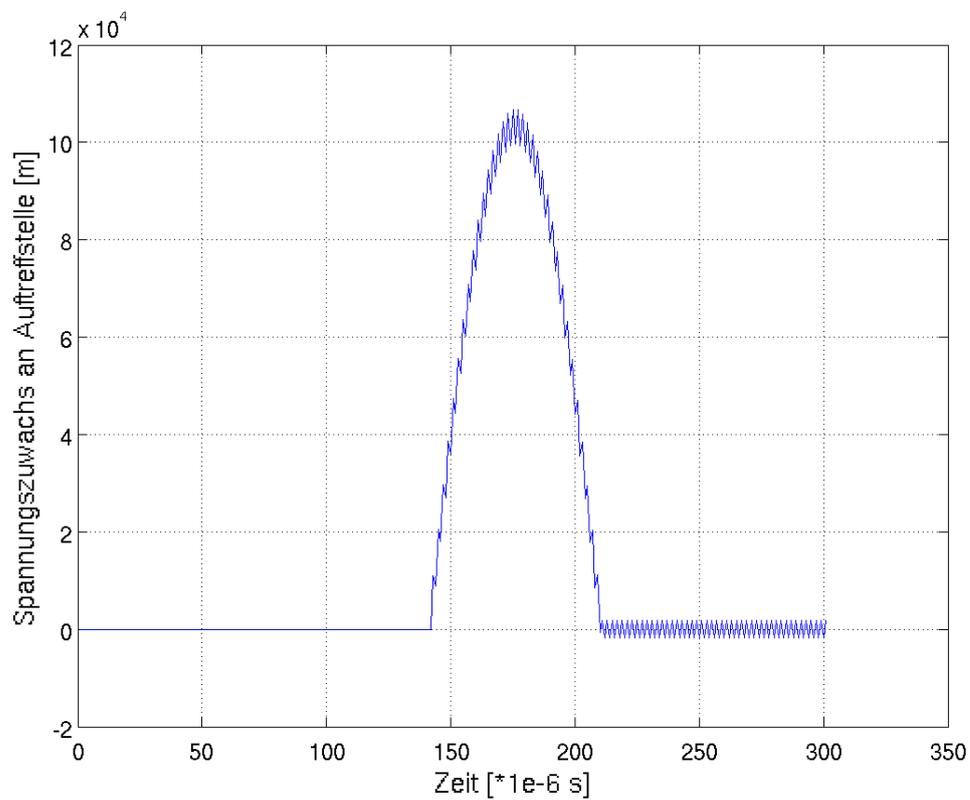


Abbildung 5.5: Spannungszuwachs an der Auftreffstelle

## 5.2 Parameterstudien

### 5.2.1 Spannung in RADIOSS

Ein Test über die in das Starter-File eingeschriebene Spannung (PLOAD) nur mit RADIOSS ohne Kopplung wurde gemacht, um ihre Funktionsweise zu untersuchen. Ein statischer Druck mit einer Größe wie der Luftdruck  $1.0e5$  Pascal wurde auf ein Element von derselben Balken gesetzt. Der Knoten (id 188) wird als die Kontaktstelle berücksichtigt.

Aus der Gleichung

$$w(x) = \frac{F}{6EI} lx^2 \left(3 - \frac{x}{l}\right) \quad (3.13)$$

ergibt sich die Durchbiegung eines Balkens unter einer statischen Kraft

$$F = \sigma \cdot s, \quad (5.1)$$

worin  $\sigma$  der Druck ist. Die Elementfläche lässt sich durch

$$s = \frac{s_b}{308} \quad (5.2)$$

berechnen, wobei  $s_b$  die gesamte Fläche der vernetzten Geometrie ist. 308 ist die Anzahl der Vernetzungselemente auf dem Balken. Die statische Kraft wurde auf den Balken ausgeübt. In der Abbildung 5.6 geht es um den Trend der Durchbiegung an den Knoten (id 188) durch diese Kraft innerhalb  $2e-5$  Sekunden. Aus der Abbildung geht hervor, dass die Durchbiegung auf rund  $0.00014$  m steigt. Es gibt ein Phänomen, dass die ermittelte Durchbiegung sich nicht ganz linear erhöht, sondern sich am Anfang leicht beschleunigt, was sich entnehmen lässt, die Genauigkeit am Anfang der Ausführungen von RADIOSS nicht sehr linear ist.

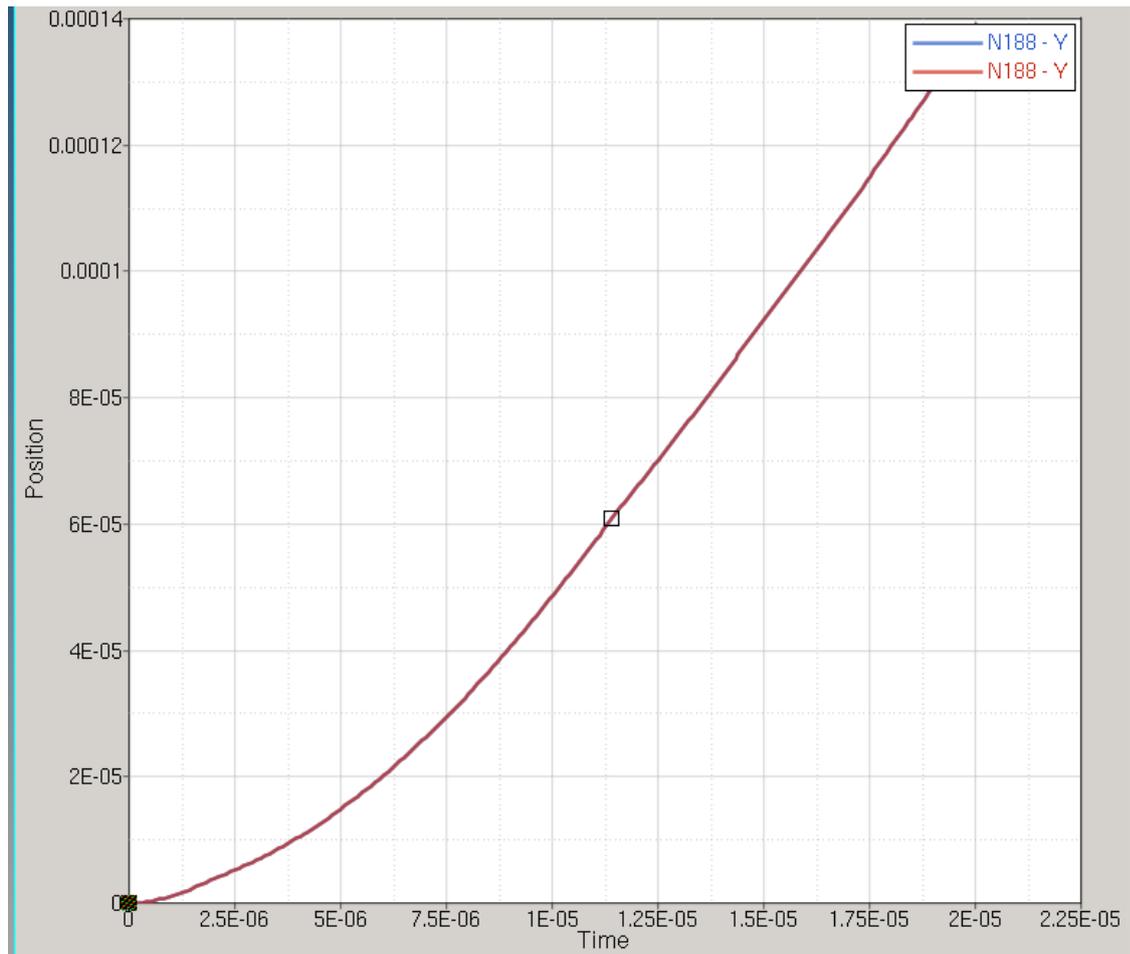


Abbildung 5.6: Trend der Durchbiegung durch eine statische Kraft

## 5.2.2 Zeitschritt

Die Abbildung 5.7 gibt Auskunft über den Einfluss von dem Zeitschritt in der Kopplung. Die Abweichung zeigt einen kontinuierlichen Rückgang im positiven Bereich und einen Zuwachs im negativen Bereich. Die Abweichung bei einem Zeitschritt von  $1.3e-7$  Sekunden beträgt circa null. Die Abbildung lässt sich entnehmen, dass sich der absolute Wert von der Abweichung mit entweder einem größeren Zeitschritt oder kleineren erhöht.

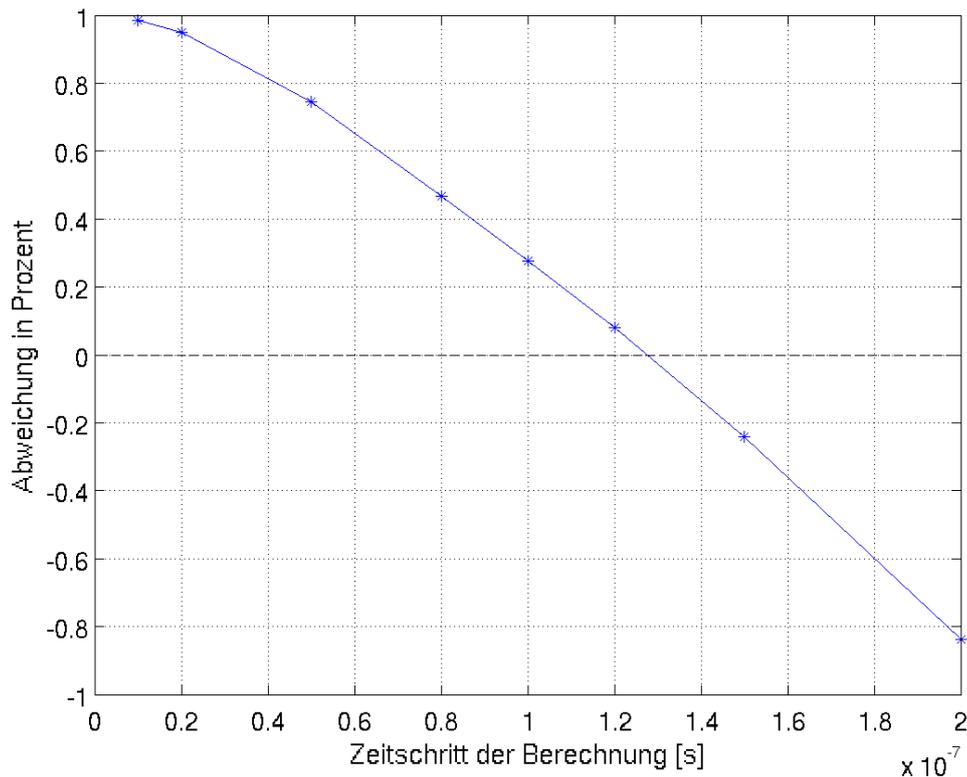


Abbildung 5.7: Einfluss von Zeitschritten auf Abweichung zwischen Analyse und Simulation

Der Grund für den Rückgang der Abweichung im positiven Bereich lässt sich durch vernachlässigte Geschwindigkeiten und Beschleunigungen des Balkens erklären, dabei sinkt die Abweichung stetig mit größeren Zeitschritten. Übrigens siehe Kapitel 5.2.1, am Anfang der Ausführungen von RADIOSS in der Kopplung ist die Genauigkeit nicht groß, weil die sehr kleine Durchbiegung sowie der Spannungszuwachs jeder Kopplung mit Verkleinern des Zeitschrittes mehr vernachlässigt werden könnte.

Für die weitere Entwicklung im negativen Bereich der Abweichung lässt sich durch zu wenige Kopplungszahl erklären. Während der Zeitschritt sich vergrößert, verringert sich die Kopplungszahl. Das heißt, die Anzahl der abwechselnden Berechnungen zwischen LIGGGHTS und RADIOSS wird reduziert. Die Abweichung soll bei dem expliziten Verfahren durch Vergrößern des Zeitschrittes divergierend sein. In der Praxis, Je länger rechnen beide Programme in einem Inkrement, desto schlagartiger ist die Durchbiegung im Inkrement. Ebenfalls verformt sich der Balken auch mehr.

### 5.2.3 Schrittweite

Die Abbildung 5.8 stellt eine fast gleiche Linie wie die von der Abbildung 5.7 dar. So sieht die Tabelle im Anhang 5.1 aus: die simulierten Werte der 2 Reihen sind sogar oft identisch bei gleichen Inkrementen. Mit Vergleich der zwei Abbildungen und der Tabelle lassen sich entnehmen, dass Zeitschritte und Schrittweiten denselben Einfluss auf die Abweichung der Durchbiegung

haben. Genauer gesagt hängt die Abweichung von dem Inkrement (Zeitschritt Mal Schrittweite).

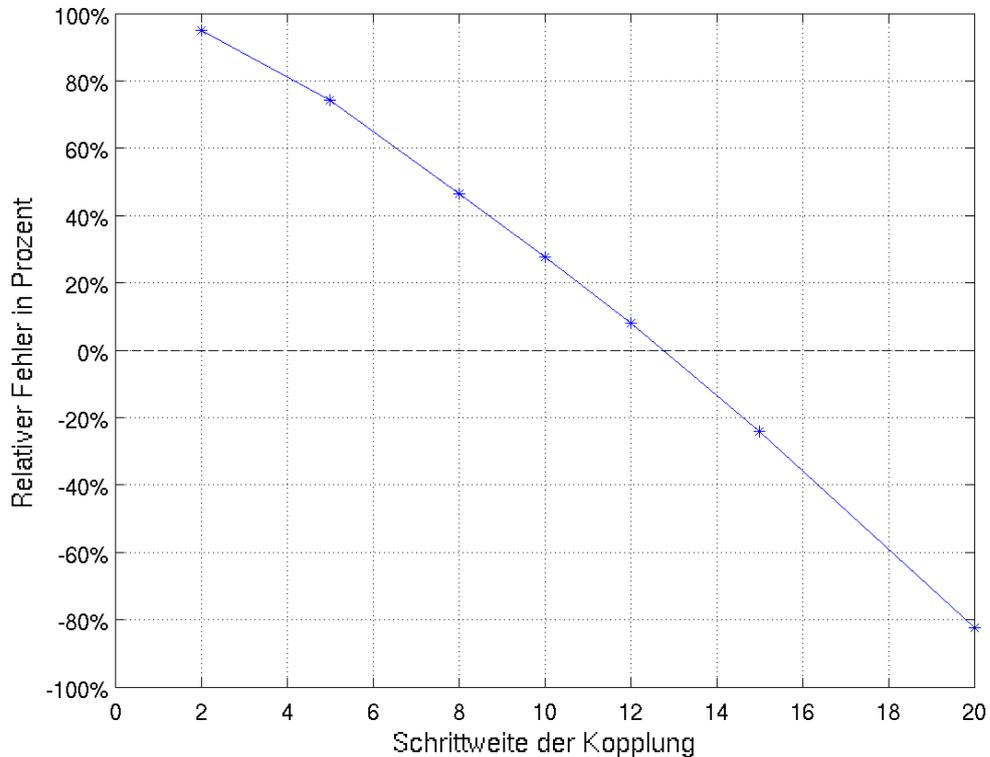


Abbildung 5.8: Einfluss von Schrittweiten auf Abweichung zwischen Analyse und Simulation

Tabelle 5.1: Vergleich der simulierten Werte bei gleichem Inkrement

Zeitschritt	Simulierter Wert	Schrittweite	Simulierter Wert
8	8.2927e-5	8E-8	8.21795e-5
10	0.00011946	1e-7	0.000111946
12	0.000142279	1.2e-7	0.000142283
15	0.000191625	1.5e-7	0.000191625

In dem Fall weist die Abweichung den geringsten absoluten Wert knapp bei einem Inkrement von  $1.3e-7$  Sekunden auf. Dieser Inkrementwert ist aber nicht global. Er verhält sich nur für gewählten Radius oder Dicke so. Bei anderen Radius oder Dicke liegt die geringste Abweichung auf anderem Inkrement. Übrigens darf die Schrittweite nicht 1 sein, sonst tritt Fehler in LIGGGHTS auf.

### 5.2.4 Dicke des Balkens

Die Abbildung 5.9 gibt einen Überblick über den Einfluss von dem Balkendicken auf die Abweichung zwischen Analyse und Simulation. Mit Vermehrung der Balkendicken nimmt der Absolutwert der Abweichung im positiven Bereich ab und steigt wieder im negativen Bereich an. Im Unterschied zu dem Zeitschritt hat die Balkendicke eine Grenze.

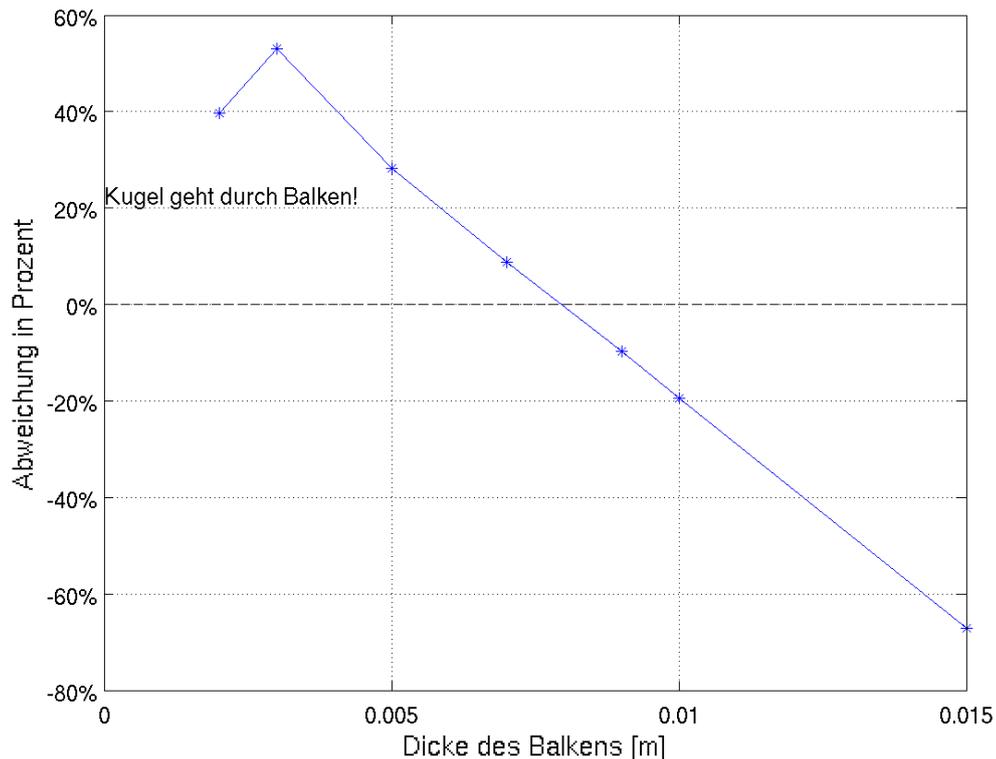


Abbildung 5.9: Einfluss von Dicke des Balkens auf Abweichung zwischen Analyse und Simulation

Der Balken darf nicht unendlich dünn sein, weil die Kugel den Balken bei einem Dicken von 0.001 m durchdringen kann. Bei einem Dicken von 0.002 m findet ein großer Rückprall des Balkens vermutlich schon statt, was intern von LIGGGHTS abhängig ist (siehe Kapitel 5.1.2). deswegen befindet sich der Punkt ganz links außerhalb der ganzen Linie.

Andererseits wird die Durchbiegung kleiner mit Verdickung des Balkens. Die Linie kann noch sehr stetig abbilden und kann sehr weit nach rechts unten erreichen. Für diese Entwicklung im negativen Bereich der Abweichung lässt sich keine genaue Erklärung finden. Vermutlich lässt sich das Problem unter die Grenze der Berechnung durch LIGGGHTS betrachten.

### 5.2.5 Radius der Kugel

Aus Abbildung 5.10 geht hervor, dass die Kurve der Abweichung über Kugelradius wie eine Parabel verläuft. Je mehr der Radius sich verringert oder vermehrt, desto vermehrt sich immer

die Abweichung zwischen Analyse und Simulation. Die Kurve sieht anders als bisher diskutierte Linien aus. Der Radius hat auch Grenzen, dass sie nicht größer als 0.012 m oder kleiner als 0.00001 m sein soll. Beim Radius im Bereich von 0.02 m bis 0.12 m ist die Abweichung kontrollierbar, darin die Abweichung zweimal bei null liegt.

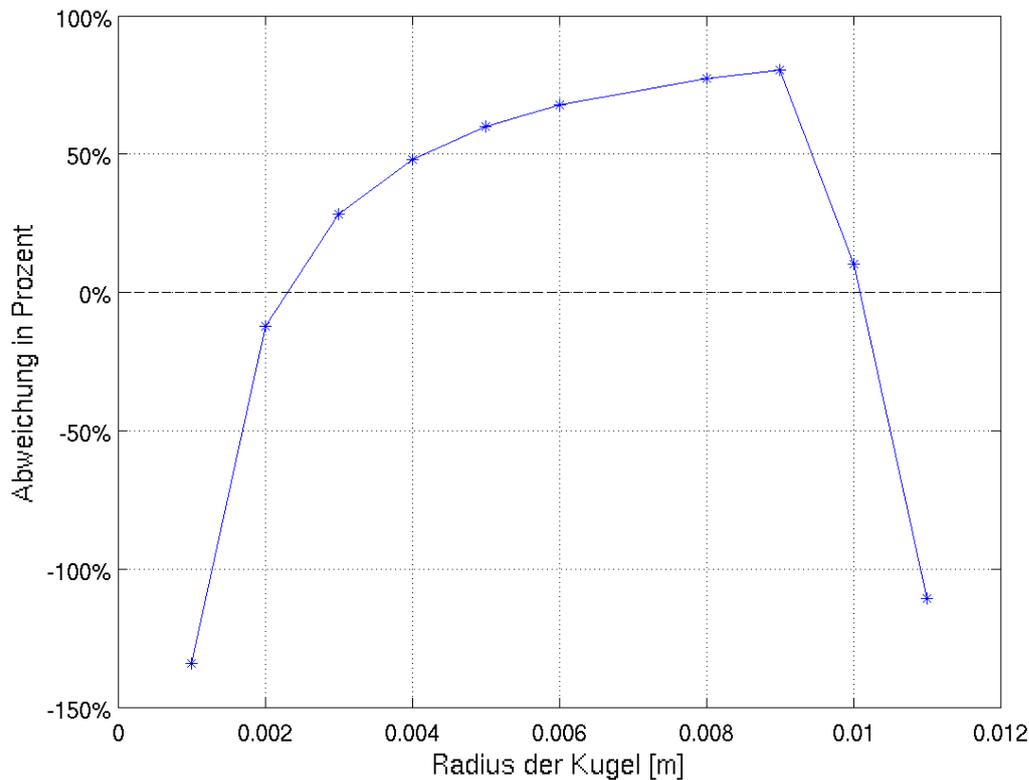


Abbildung 5.10: Einfluss von Radius der Kugel auf Abweichung zwischen Analyse und Simulation

Wenn die Kugel zu groß ist, geht sie nicht durch den Balken, sondern bettet sie in den Balken ein und schwingt intensiv mit dem Balken. Das passiert schon beim Radius von 0.012 m. Beim Radius ungefähr von 0.009 m bis 0.011 m findet Stoß der Kugel auf den Balken wieder zu lokal statt, deshalb illustriert die Abbildung in dem Bereich wie eine Linie nach unten und vermehrt sich der simulierter Wert zu schnell.

Aber wenn die Kugel zu klein ist, beginnt LIGGGHTS nicht zu rechnen, weil sein Limit auf Berechnung erreicht wird. Während der Radius sich jedes Mal verringert, rechnet LIGGGHTS schwerer und dauert es länger.

Wenn ein Experiment über das Modell später durchgeführt wird, lässt sich z.B. eine Kugel mit dem Radius von 0.01 m einsetzen. Mit allen Parameter unverändert soll die analytische Durchbiegung bei  $5.778e-3$  m liegt, was gut messbar ist, damit die Analytische Lösung, das Simulationsergebnis und das Versuchsergebnis miteinander verglichen werden können. Das Ausgangsmodell ist aber schwer messbar, da seine analytische Durchbiegung nur  $1.561e-4$  m beträgt.

## 5.2.6 Anfangsgeschwindigkeit der Kugel

Die Abbildung 5.11 bildet den Einfluss von Anfangsgeschwindigkeit der Kugel auf Abweichung zwischen Analyse und Simulation ab. Die Abweichung bleibt auf etwa demselben Niveau zwischen 28.25% und 28.3%. Die Grenze der Anfangsgeschwindigkeit liegt ungefähr bei 190 m/s, weil die Kugel mit dieser Anfangsgeschwindigkeit wieder durch den Balken eindringen wird.

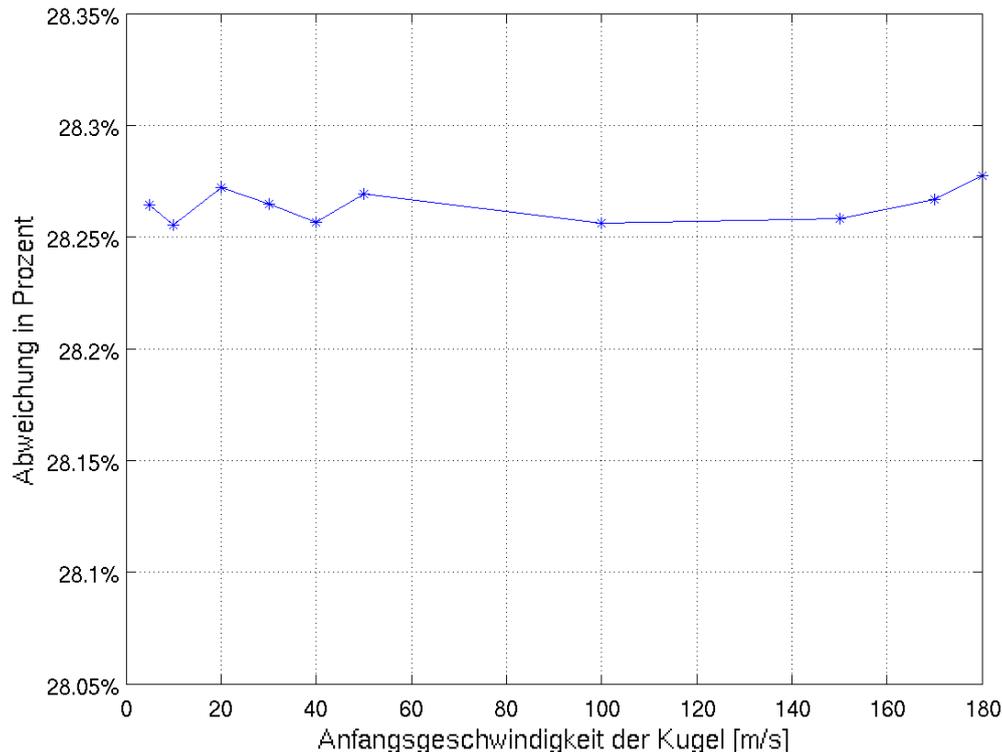


Abbildung 5.11: Einfluss von Anfangsgeschwindigkeit der Kugel auf Abweichung zwischen Analyse und Simulation

Die Anfangsgeschwindigkeit der Kugel spielt keine große Rolle in der Abweichung zwischen Analyse und Simulation. Die Abweichungen sind in einem kleinen Bereich stabilisiert. Und die Genauigkeit der Simulation hängt nicht von der Anfangsgeschwindigkeit der Kugel ab.

## 5.3 Einheitssystem

Eigentlich gibt LIGGGHTS Spannung aus und RADIOSS bildet Vernetzung ab. Was ganz wichtig ist, LIGGGHTS verwendet das Internationale Einheitensystem (SI), während RADIOSS in einem anderen Einheitssystem rechnet. Deswegen muss der Spannungszuwachs für jedes Inkrement skaliert werden. In dem Fall wird er immer mit einem Faktor von  $-1e-6$  multipliziert, weil die Spannung in RADIOSS nicht mit Pascal wie in LIGGGHTS, sondern mit  $N/mm^2$  rechnet wird, und die Richtung der Durchbiegung des Balkens nach unten sein soll.

Alle Abmessungen und Koordinaten können jedoch einheitslos zwischen LIGGGHTS und RADIOSS aufgetragen werden. Denn LIGGGHTS denkt, dass RADIOSS auch im Internationalen Einheitensystem rechnet. Tatsächlich läuft LIGGGHTS in einem Einheitssystem mit Längeneinheit von Millimeter. Das Einheitssystem nennt sich im Folgenden Millimetersystem. Auf der anderen Seite bekommt RADIOSS immer die skalierte durch Pythonskript von LIGGGHTS ausgegebene Spannung.

RADIOSS ist doch einheitslos, aber jedes Modell in RADIOSS hat seine eigene Grenze. Wenn RADIOSS für dies vernetzte Balkenmodell im Internationalen Einheitensystem berechnet, leitet der Stoß langsamer als im Millimetersystem. Wenn das Inkrement zu klein eingestellt ist, findet der Stoß zu lokal statt, was so wie in der Abbildung 5.12 aussieht. Im Millimetersystem passiert das nur mit noch kleinerem Inkrement.

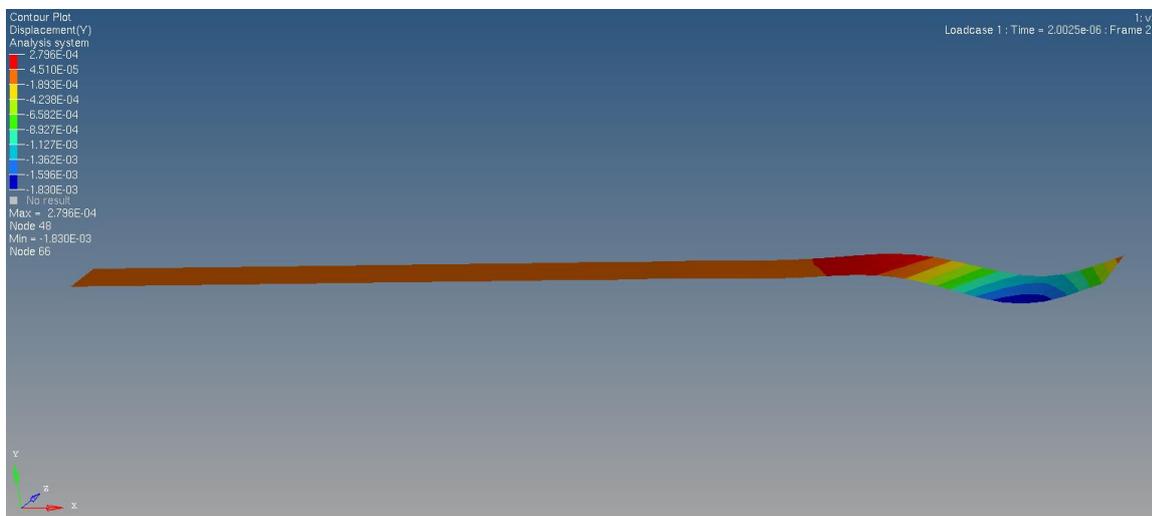


Abbildung 5.12: Stoß findet zu lokal statt

## 5.4 Diskussion über Gründe von Fehlern

Es gibt bei Simulation drei Arten von Fehlern, siehe Tabelle 5.2. Eine ist die Ungenauigkeit der Kopplung. Zweite ist die internen Softwarefehler und Softwaregrenzen von den beiden gekoppelten Softwares. Dritte ist die Annäherung der analytischen Lösung. Die verschiedenen Probleme und Fehler, die im Zusammenhang mit der Kopplung auftreten können sowie deren Problembehandlung, sind in der Tabelle beschrieben.

Tabelle 5.2: Übersicht der Fehlerarten

Fehlerarten	Phänomen	Einfluss auf Abweichung	Lösung
Kopplungsfehler	Vernachlässigte Geschwindigkeiten und Beschleunigungen	Größere Abweichung mit Abnahme und Zunahme des Inkrements	Begrenzung des Inkrements

Softwarefehler	Rückprall des Balkens in RADIOSS	Lokale Umformung	Zunahme des Inkrements
	Ungenauigkeit am Anfang der Simulation von RADIOSS	Größere Abweichung mit Abnahme des Inkrements (vielleicht)	Zunahme des Inkrements
	Grenzen von LIGGGHTS	Läuft nicht	Änderung des Einheitssystems
Analytische Fehler	Kugel nicht als Starrkörper betrachtet; Viele Annahmen	Verkleinerung der analytischen Lösung	Unklar

Laut Simulation ist die ermittelte Durchbiegung immer noch kleiner als die durch die analytische Berechnung von Gummert, weil die Geschwindigkeit und Beschleunigungen des Balkens für jedes Inkrement immer vernachlässigt werden muss. Das heißt, die Massenträgheit des Balkens wird überschätzt. Die Ungenauigkeit lässt sich durch Begrenzung des Inkrements vermeiden.

Die analytische Lösung ist unter viele Annahmen, z.B. die Kugel als Starrkörper, was nicht realisierbar durch LIGGGHTS ist, die realistische Durchbiegung soll kleiner als der analytische Wert sein.



## 6 Zusammenfassung

Ziel der vorliegenden Masterarbeit war es, eine Umgebung für DEM-Software LIGGGHTS und FEM-Solver RADIOSS durch Programmiersprache Python zu errichten, wo die beide Softwares abwechselnd laufen können, damit die Simulationsmodelle später in übertragenem Sinn zu Granulaten und Silos gebraucht werden können.

Im Kapitel 2 wurden die theoretischen Grundlagen über DEM und FEM, den aktuellen technischen Stand der FEM-DEM Kopplung und das Unterschied zwischen impliziten und expliziten Verfahren dargestellt. Im Kapitel 3 wurde die analytische Betrachtung der dynamischen Durchbiegung bei Querstoß auf einen elastischen Balken erklärt. Im Kapitel 4 wurden die Kopplungsmethode vorgestellt. Ein Ausgangsmodell wird genau wie das in der analytischen Berechnung umgesetzte Beispiel überlegt.

Das Kapitel 5 gab Information über die Simulationsstudien und Auswertungen. Die analytische Lösung der größten Durchbiegung beim Ausgangsmodell beträgt  $1.560699e-4$  m in dem Fall, während das simulierte Ergebnis  $1.11946e-4$  m beträgt. Die Abweichung von dem Ausgangsmodell zwischen Analyse und Simulation beläuft sich auf rund 28.27%. Das Ziel dieser Arbeit wurde somit erreicht, da die FEM-DEM Kopplung realisiert wurde.

Die Kopplung kann zur Simulation des Kontakts zwischen Kugel und Schalenmodell verwendet werden. LIGGGHTS ist für den Zustand der Kugel verantwortlich, während RADIOSS für Verformung des Balkens zuständig ist. In einem kleinen Inkrement gibt LIGGGHTS die Spannung auf die Kontaktstelle für RADIOSS aus und RADIOSS bildet neue Vernetzung für LIGGGHTS ab. In einem kurzen Stoßvorgang berechnen beide Softwares abwechselnd mehrmals. Durch ParaView lässt sich die authentische Animationen des Stoßvorgangs sehen, einschließlich des Vorformungsvorgangs von Schalenmodell und Kugel. Bei verschiedenen Kugelarten sind nur einige Parameter im Liggghtsskript zu ändern. Beim Einsatz eines neuen Schalenmodells muss das Pythonskript auch verändert werden.

Beide Softwares haben ihre Grenzen. Ihre Einheitssysteme lassen sich ändern, weil sie beide einheitslos berechnen. Es ist möglich, den Stoßvorgang in verschiedener Zahlendimension zu untersuchen. Die Kopplung ist noch nicht perfekt. Ihr Umfang ist nicht sehr weit. Die Geschwindigkeit und Beschleunigung nach Ausführung von RADIOSS sind verlorengegangen. Deshalb lässt sich das Inkrement nicht beliebig klein einstellen. Vielleicht kann LIGGGHTS mit anderem FEM-Solver gekoppelt werden, dessen Restart-File für jedes Inkrement aktualisieren darf. In der Zukunft lassen sich auch mit mehreren Kugeln oder komplizierterer Geometrie des Schalenmodells versuchen.



## 7 Quellenverzeichnis

- [Alt16] Altair Hyperworks: Internet:  
<http://www.altairhyperworks.com/product/RADIOSS> (Zugriff: 04.01.2017)
- [Bör11] Börner, M.: Einführung in die Diskrete Element, Vorlesungsunterlagen.  
Internet:  
[http://www.ovgu.de/ivt/tvt/media/6a3a9decbbafe2b/dem\\_skript\\_deutsch.pdf](http://www.ovgu.de/ivt/tvt/media/6a3a9decbbafe2b/dem_skript_deutsch.pdf)  
(Zugriff: 04.01.2017)
- [Bor14] Borst, R.D.; Crisfield, M.A.; Remmers, J.J.C.; Verhoosel, C.V.: Nichtlineare Finite-Elemente-Analyse von Festkörpern und Strukturen. John Wiley & Sons.
- [cae16] Explizit-implizit anschaulich. Internet: [http://www.cae-wiki.info/wikiplus/index.php/Explizit-implizit\\_anschaulich](http://www.cae-wiki.info/wikiplus/index.php/Explizit-implizit_anschaulich)
- [Hut16] Hütter, G.: Ausgewählte Kapitel der FEM, Vorlesungsunterlagen. Institut für Mechanik und Fluidodynamik. Technische Universität Freiberg
- [LIG16] CFDEMresearch GmbH: LIGGGHTS Open Source Discrete Element Method Particle Simulation Code. Internet: <http://www.cfdem.com/liggghts-open-source-discrete-element-method-particle-simulation-code> (Zugriff: 04.01.2017)
- [Mic15] Michael, Mark; Vogel, Frank; Peter, Bernhard: DEM–FEM coupling simulations of the interactions between a tire tread and granular terrain. Computer Methods in Applied Mechanics and Engineering: Volume 289, 1 June 2015, Pages 227–248
- [Pös13] Pöschl, Theodor: Elementare Festigkeitslehre: Zum Gebrauche bei Vorlesungen und zum Selbststudium. Springer-Verlag. S.212
- [Pös28] Pöschl, Theodor; Angenheister, G.; Föppl, O.: Mechanik der Elastischen Körper. Springer-Verlag. Springer-Verlag. S.552
- [Pös28] Gummert P.R.W; Reckling K.A.: Mechanik. Springer-Verlag. Springer-Verlag. S.571 - 572
- [Sie15] Siegert, Klaus: Blechumformung: Verfahren, Werkzeuge und Maschinen. S.286
- [Spu12] Spur, Günter; Neugebauer, Reimund; Hoffman, Hartmut: Handbuch Umformen. Carl Hanser Verlag. S.57
- [Str13] Stransky, J.: Open Souce DEM-FEM Coupling. Internet:  
[https://www.researchgate.net/profile/Jan\\_Stransky/publication/278410338\\_OPEN\\_SOURCE\\_DEM-FEM\\_COUPLING/links/55807c6708aed40dd8cd27ae.pdf](https://www.researchgate.net/profile/Jan_Stransky/publication/278410338_OPEN_SOURCE_DEM-FEM_COUPLING/links/55807c6708aed40dd8cd27ae.pdf) (Zugriff: 04.01.2017)
- [Wan14] Wang, Wei; Liu, Yong; Liu, Kun: Using FEM–DEM coupling method to study three-body friction behaviour. Computer Methods in Applied Mechan-

ics and Engineering. Wear: Volume 318, Issues 1–2, 15 October 2014, Pages 114–123

## Anhang A: Daten CD

- 📁 CD Stammverzeichnis
  - 📄 Arbeit als maschinenlesbare PDF inkl. gescanntem Deckblatt
  - 📁 Ausgangsmodell
    - 📄 analytische berechnung.py
    - 📄 biggest displacement.py
    - 📄 Kopplung.py
    - 📄 LiggghtsRadioss\_func.py
    - 📄 LiggghtsRadioss\_func.pyc
    - 📁 data
      - 📄 in.kopplung0
      - 📄 mesh\_fein
      - 📄 Sim
      - 📄 v5\_0000.rad
      - 📄 v5\_0001.rad
    - 📁 liggghts
      - 📄 in.kopplung
      - 📄 in.kopplung\_restart
      - 📄 in.kopplung0
      - 📄 log.liggghts
        - data
        - post
        - restart\_files
    - 📁 radioss
      - 📄 run\_0000.rad
      - 📄 run\_0001.rad
      - 📄 ...