Fachhochschule
Braunschweig/Wolfenbüttel

Fachbereich Informatik - Technische Informatik

Prof. Dr.-Ing. Jürgen Kreyßig

Diplomarbeit

Softwarelösung zur Materialklassifizierung in FEM-Input-Decks für simulierte PKW-Crash-Versuche

Christoph Borchers (20165077)
Wolfsburg, 23. August 2005

Volkswagen AG

EBKB - Pkw-Berechnung und Versuch Karosserie

Dipl.-Ing. Pierre Talkowsky
Dr. Heinrich Gades

Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt zu haben. Die verwendete Literatur und die sonstigen Hilfsmittel sind vollständig angegeben.

Wolfsburg, 23. August 2005

Danksagung

An dieser Stelle möchte ich allen danken, die durch Ihre Unterstützung zum Gelingen dieser Diplomarbeit beigetragen haben.
Zuerst möchte ich mich bei Herrn Prof. Dr.-Ing. Kreyßig bedanken, der mich seitens der Fachhochschule Braunschweig/Wolfenbüttel betreute, mich positiv unterstützte und jederzeit für Fragen zur Verfügung stand.
Weiterhin gilt mein besonderer Dank der Volkswagen AG und meinen beiden Betreuern, Herrn Talkowsky und Herrn Dr. Gades, sowie Herrn Dr. Liebertz, die mir die Möglichkeit zum Bearbeiten dieser interessanten Aufgabenstellung gaben. Sie haben mich bei allen auftretenden Fragen unterstützt und waren immer zu einem Fachgespräch bereit.

Inhaltsverzeichnis

Erklärung **II**

Danksagung **III**

Kurzfassung **VII**

1 Einleitung **1**

2 Aufgabenstellung **3**

3 Grundlagen **4**
3.1 Zugelastizität fester Körper 4
3.1.1 Mechanische Spannung 4
3.1.2 Das Hookesche Gesetz 6
3.1.3 Anelastisches Verhalten 7
3.2 Finite-Elemente-Methode 8
3.3 PAM-Crash 9
3.4 PAM-View 10
3.5 Die Programmiersprache C/C++ 11
3.6 HTML 12

4 Zustandsanalyse **13**
4.1 Ist-Zustand 13
4.2 Soll-Zustand 14

5 Theoretischer Lösungsansatz **15**
5.1 Zugversuch 15
5.2 Kurvenvergleich 17
5.2.1 Grundbedingungen 17
5.2.2 Vergleichsmethoden 17
5.2.2.1 Erste Vergleichsmethode 17
5.2.2.2 Zweite Vergleichsmethode 18
5.2.2.3 Dritte Vergleichsmethode 20
5.2.3 Mathematische Grundlagen des Kurvenvergleichs 21
5.2.3.1 Punkt-Punkt-Abstand 21
5.2.3.2 Punkt-Gerade-Abstand 22

5.2.4 Ablauf eines Kurvenvergleichs . . . 24
5.2.5 Gewichtung einer Kurve . . . 26

6 Planungs- und Definitionsphase 28
6.1 Lastenheft . . . 28
6.1.1 Zielbestimmungen . . . 28
6.1.2 Produkteinsatz . . . 29
6.1.3 Produktfunktionen . . . 29
6.1.4 Produktdaten . . . 31
6.1.5 Produktleistungen . . . 31
6.1.6 Qualitätsanforderungen . . . 32
6.2 Pflichtenheft . . . 32
6.2.1 Zielbestimmungen . . . 32
6.2.1.1 Musskriterien . . . 32
6.2.1.2 Wunschkriterien . . . 32
6.2.2 Produkteinsatz . . . 33
6.2.3 Produktschnittstellen . . . 33
6.2.4 Produktfunktionen . . . 33
6.2.5 Produktdaten . . . 40
6.2.6 Produktleistungen . . . 40
6.2.7 Qualitätsanforderungen . . . 41
6.2.8 Benutzeroberfläche . . . 41
6.2.9 Entwicklungs- und Arbeitsumgebung . . . 41

7 Entwicklungsphase 43
7.1 Implementierung . . . 43
7.2 Programmstruktur . . . 44
7.2.1 Beschreibung der Klassen . . . 45
7.2.2 Programmfunktionen . . . 48
7.3 Umsetzung des Kurvenvergleichs . . . 51
7.4 Programmtest . . . 53

8 Schlussbemerkung 55

Abkürzungsverzeichnis 57

Bildverzeichnis 58

Tabellenverzeichnis 59

Listings 60

Literatur 61

A Programmhandbuch **63**
A.1 Beschreibung 63
A.2 Funktionen 63
A.2.1 Klassifizierung 63
A.2.2 Materialdatenbank 64
A.3 Konfigurationsdatei 65
A.4 Hinweis 66

B CD-ROM **67**

Kurzfassung

In der Automobilentwicklung werden simulierte Crashtests von neuen Fahrzeug-Modellen durchgeführt. Dazu werden Simulationsmodelle verwendet, die aus der Geometrie des Fahrzeugs, den Verbindungen der einzelnen Bauteile und den physikalischen Materialeigenschaften bestehen. Die Materialeigenschaften sind wiederum als Materialkarten in den Modellen abgelegt und werden bei Neuentwicklungen aus einer VW-eigenen Datenbank bezogen. Leider wird die Datenbank von Fremdfirmen, die ebenfalls Simulationsmodelle entwerfen, nicht immer genutzt. Somit kommt es vor, dass in einem Modell auch bei Volkswagen unbekannte Materialien verwendet werden. Damit die Entwickler bei Volkswagen aber auch Änderungen an Modellen von Fremdfirmen vornehmen können, bietet es Vorteile, dass Materialien bekannt sind. Auf Grund der komplexen Struktur ist ein Vergleich unbekannter Materialkarten mit denen der VW-Datenbank u.U. extrem aufwendig. Um trotzdem unbekannte Materialien auf einfache Weise klassifizieren zu können, wird mit jedem Material ein standardisierter Zugversuch simuliert, dessen Ergebnisse mit Referenzzugversuchen verglichen werden können. Nachdem ein Material klassifiziert worden ist, kann die unbekannte Materialkarte durch eine bekannte ersetzt werden.

Inhalt dieser Arbeit ist es, eine Software zu entwerfen, die die Materialkarten extrahiert, die Zugversuche unter zuhilfenahme eines weiteren Programms automatisiert durchführt, die Ergebnisse mit einem geeigneten Verfahren miteinander vergleicht und abschließend die unbekannten Materialkarten durch bekannte Materialkarten ersetzt.

1 Einleitung

Die Entwicklung von Automobilen hat sich in den letzten Jahren tiefgreifend verändert. Wurden Autos früher am Reißbrett entworfen, so hat heute der Computer auch in diesem Bereich die klassischen Techniken verdrängt. Der Einsatz von CAD[1]-Software ermöglicht es, ein Fahrzeug am Computer zu entwerfen und Änderungen schnell und kosteneffizient durchzuführen.
Natürlich wird nicht nur die optische und funktionelle Seite eines Fahrzeugs mit Hilfe von Software gestaltet, sondern auch der sicherheitstechnische Aspekt. Auch hier spielen die Kosten eine große Rolle. Ein simulierter Fahrzeugcrash ist wesentlich günstiger als ein Prototypenversuch. Dabei ist die passive Sicherheit besonders wichtig. Zu ihr gehören hauptsächlich Elemente des Insassenschutzes, wie z.B. der Seitenaufprallschutz in den Türen. Aktive Sicherheitselemente sind hingegen Bremsen, Fahrwerk, elektronische Sicherheitskomponenten, usw.
Um das Crashverhalten eines Fahrzeugs zu simulieren, wird sich der FEM[2] bedient. Ein im CAD-Verfahren entwickeltes Fahrzeug wird in ein FEM-Modell überführt, indem die einzelnen Komponenten eines Fahrzeuges in finite Elemente mit einer möglichst gleichmäßigen Verteilung diskretisiert werden. Daraus ergibt sich ein Crashmodell, welches aus drei Teilen besteht:

- Geometrie (FEM-Netz)
- Verbindungen der Komponenten
- physikalische Materialeigenschaften der Komponenten

Die Beschreibung der Geometrie und der Verbindungen ist durch die Syntax der Simulationsprogramme festgelegt. Dies gilt auch für die Materialeigenschaften, die in Form von sogenannten Materialkarten abgelegt werden. Wegen der Vielzahl und komplexität der verfügbaren Materialkarten ist es jedoch möglich, dass unterschiedliche Materialkarten physikalisch gleichwertige Materialeigenschaften kodieren. Dies tritt häufig auf, wenn Simulationsmodelle

[1] Computer Aided Design
[2] Finite Elemente Methode

von Kooperationspartnern entworfen werden und von der Volkswagen AG weiterbearbeitet werden sollen. Zwar stellt Volkswagen allen Partnern eine Datenbank mit allen nötigen Materialmodellen zur Verfügung, aber in einigen Fällen wird diese nicht genutzt.
Leider ist es nicht so einfach, die Materialmodelle in einem Simulationsmodell gegen die aus der Datenbank zu tauschen, da eine einfache Identifizierung auf Grund der komplexen Strukturen der Materialkarten nicht möglich ist. An diesem Punkt setzt die hier vorliegende Arbeit an.
Damit ein Materialmodell trotzdem klassifiziert werden kann, muss ein anderes Vergleichsverfahren gefunden werden. Ein Vergleich von zwei Proben, hier ein unbekanntes und ein bekanntes Material, kann nur erfolgen, wenn beide die gleiche Vergleichsbasis besitzen. Diese wird durch einen Versuch mit identischen Bedingungen geschaffen. Für Stähle wird dazu ein Zugversuch durchgeführt, bei dem die Materialprobe mit einer bestimmten Geschwindigkeit gezogen wird. Als Ergebnis ergibt sich eine Kurve der entstandenen Spannung, aufgetragen über die relative Dehnung der Probe. Der Zugversuch lässt sich auch auf andere Materialien, wie z. B. Kunststoffe, anwenden.
Im Vorfeld wird also der Zugversuch mit allen Materialien aus der Datenbank, sowie mit allen unbekannten Materialien durchgeführt. Danach können die Ergbniskurven der unbekannten mit denen der bekannten Materialien verglichen werden. Über diesen Vergleich wird das unbekannte Material einem bekannten zugeordnen.
Der Vergleich soll durch den Einsatz einer Softwarelösung erfolgen, die in dieser Arbeit beschrieben wird. Dafür wird ein mathematisches Verfahren entwickelt und implementiert, welches es ermöglicht, zwei Kurven miteinander zu vergleichen.

2 Aufgabenstellung

Ziel der Diplomarbeit soll es sein, eine Softwarelösung zu entwickeln, die für alle Materialien eines FEM-Input-Decks einen automatischen Abgleich mit den in der VW-Datenbank abgelegten Standardmaterialien durchführt.
Die Materialien sind in Form von Materialkarten in den FEM-Input-Decks abgelegt. Die Klassifizierung soll durch einen simulierten dehnratenabhängigen Zugversuch erfolgen, der als Ergebnis ein Spannungs-Dehnungs-Diagramm liefert.
Das zu entwickelnde Programm soll folgende zwei Hauptaufgaben erfüllen:

1. Durchführen der Zugversuche für alle Materialien der VW-Datenbank und Ablegen der Spannungs-Dehnungs-Diagramme in einer Datenbank.

2. Extrahieren der Materialkarten aus den FEM-Input-Decks. Durchführen der Zugversuche mit den unbekannten Materialien mit anschließender Klassifizierung durch das Vergleichen der Spannungs-Dehnungs-Diagramme.

Das Programm soll in der Sprache C/C++ entwickelt werden. Im Rahmen der Entwicklung soll außerdem, ein Lastenheft, Pflichtenheft, usw. angefertigt werden.

3 Grundlagen

3.1 Zugelastizität fester Körper

Feste Körper werden durch eine einwirkende Kraft deformiert. Die Deformation wird in zwei Fälle aufgeteilt:

- Elastische Deformation:
  Der Körper geht nach dem Entfernen der einwirkenden Kraft wieder in seine Ursprungsform zurück.
- Plastische Deformation:
  Der Körper bleibt nach dem Entfernen der einwirkenden Kraft in seiner angenommen Form oder tritt in den Zustand des Fließens ein.

3.1.1 Mechanische Spannung

Eine Kraft F, die an einen Stab der Länge l und dem Querschnitt A zieht, führt zu einer Zugspannung im Material. Die Zugspannung wird durch die Normalspannung σ beschrieben:

$$\vec{F}_n = \sigma A \vec{n} \tag{3.1}$$

Dabei ist $\vec{n}$ die Flächennormale und $\vec{F}_n$ die Zug- oder Druckkraft. Wird der Stab auf Zug belastet, so ist $\sigma > 0$. Bei Durckbelastung wird $\sigma < 0$. Die Normalspannung σ trägt die Einheit $1Pa = 1N \cdot m^{-2}$.
Durch die Normalspannung wird eine Veränderung des Stabes in der Länge und im Querschnitt herbeigeführt (s. Bild 3.1).

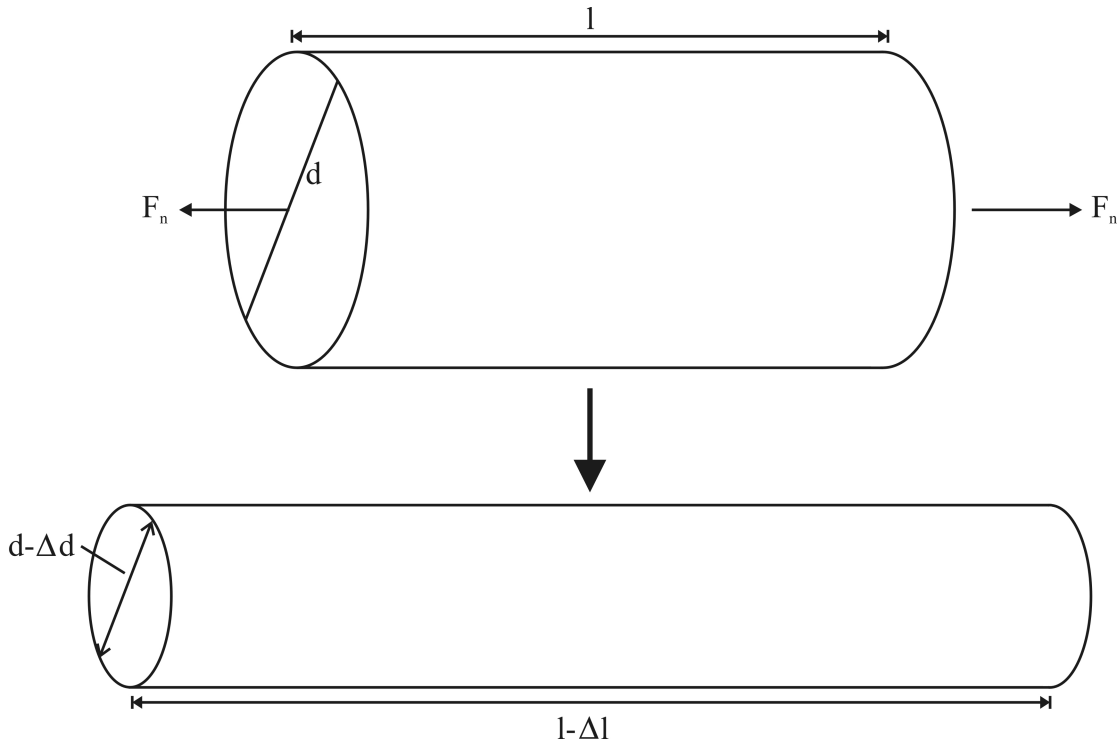

Bild 3.1: Stabdehnung

1. Die relative Längenänderung ϵ ist definiert als:

$$\epsilon = \frac{\Delta l}{l} \tag{3.2}$$

   wobei $\epsilon > 0$ eine Dehnung darstellt und $\epsilon < 0$ eine Stauchung. ϵ ist dimensionslos.

2. Die relative Querschnittsänderung wird beschrieben durch:

$$\frac{\Delta d}{d} = -\mu \cdot \epsilon \tag{3.3}$$

   $\Delta < 0$ entspricht dabei einer Querkontraktion und $\Delta > 0$ einer Querschnittsverbreiterung.

Wird (3.2) in (3.3) eingesetzt, so ist zu erkennen, dass die beiden Deformationen im Verhältnis zueinander stehen:

$$\frac{-\Delta d}{d}\frac{l}{\Delta l} = \mu \tag{3.4}$$

Das Verhältnis μ heißt Poisson-Zahl. Ihr Wert kann eingegrenzt werden, da das Volumen V eines metalischen Körpers in der Regel bei Zug konstant ist. Gemessene Werte für μ liegen

zwischen ca. 0,2 und ca. 0,5.[1] In Tabelle 3.1 sind Beispiele für die Poisson-Zahl aufgeführt.

Tabelle 3.1: Elastizitätsmodul E und Poission-Zahl μ [2]

Material	E/GPa	μ
Aluminium	71	0,34
Eisen	210	0,28
Quarzglas	75	0,17
Geräteglas	40...90	0,19...0,28
Gußeisen	110	0,22
Magnesium	44	0,28
Nickel	210	0,31
Plexiglas	3,2	0,35
Polystyrol	3,2	0,33
Schmiedeeisen	213	0,28
Stahl (1C)	210	0,28
Feder-	220	0,29
V2A-	190	
Titan	110	0,36

3.1.2 Das Hookesche Gesetz

Der Bereich der elastischen Deformation folgt dem Hookeschen Gesetz[3]. Dies besagt, dass sich ein Körper im Verhältnis zur Kraft, die auf ihn wirkt, dehnt. Hook postulierte daraus, dass zwischen der Längenänderung ϵ und der Normalspannung σ eine Proportionalität besteht:

$$\frac{F}{A} = E\frac{\Delta l}{l} \tag{3.5}$$

$$\sigma = E \cdot \epsilon \tag{3.6}$$

E bezeichnet den Elastizitätsmodul (s. Tabelle 3.1) und hat die Einheit $1Pa = 1\frac{N}{m^2}$.
Dabei ist zu beachten, dass das Hookesche Gesetz nur in einem bestimmten Bereich der relativen Längenänderung Gültigkeit besitzt. Darüber hinaus tritt ein anelastisches Verhalten auf.

[1] vgl. Gerthsen, C.; Kneser, H.; Vogel, H. (1986), S. 119
[2] vgl. Kuchling, H. (2001), S. 614-615
[3] Robert Hooke (*1635, †1703), engl. Wissenschaftler

3.1.3 Anelastisches Verhalten

Wie schon unter 3.1.2 beschrieben besitzt das Hookesche Gesetz nur bis zu einer bestimmten Grenze Gültigkeit. Wird das Material stärker belastet, tritt anelastisches Verhalten auf. Die Spannungs-Dehnungs-Kurve $\sigma(\epsilon)$ (s. Bild 3.2) geht von dem elastischen in den plastischen Bereich über. Ist die Elastizitätsgrenze (auch Fließgrenze genannt) σ_E überschritten, bewirken Umlagerungen und Gefügeänderungen im Material eine dauerhafte Formveränderung. Die Festigkeitsgrenze σ_F gibt die höchste Spannung an, die das Material aushalten kann. Wird bis zur Bruchdehnung ϵ_B gezogen, führt eine starke Abnahme der Festigkeit zum Bruch des Materials. Bereits ab σ_E, aber hauptsächlich ab σ_F, tritt Fließen ein. Im Zustand des Fließens nimmt die Dehnung selbst bei Reduzierung der Belastung weiter zu.

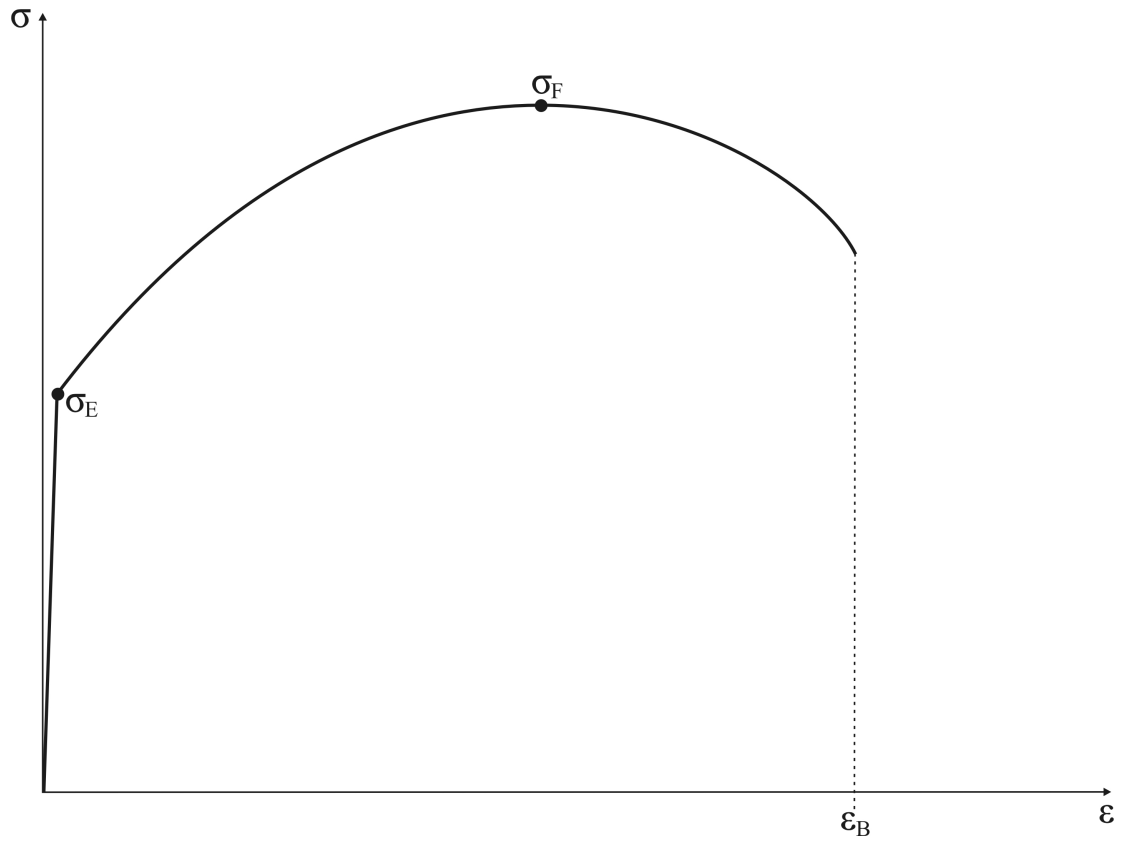

Bild 3.2: Spannungs-Dehnungs-Kurve $\sigma(\epsilon)$

3.2 Finite-Elemente-Methode

Die Finite-Elemente-Methode wurde in den 50er Jahren entwickelt. Ursprünglich diente sie dazu, dass Gesamtverhalten von komplexen Bauteilen zu beschreiben. Heute wird sie aber in vielen physikalischen Bereichen eingesetzt (z.B. Statik, Dynamik, Hydraulik, Wärmeleitung, ...). Der Begriff „Finite Elemente“ wurde allerdings erst später benutzt.
Bei der Finiten-Elemente-Methode handelt es sich um ein numerisches Berechnungsverfahren. Es dient der näherungsweisen Lösung partieller Differenzialgleichungen mit Randbedingungen. Die grundsätzliche Funktion liegt darin, komplexe Systeme auf kleinere lösbare Systeme aufzuteilen.
In der Elastizitätstheorie, die für die Crashsimulation von Bedeutung ist, wird die FEM genutzt, um die Verformung von Bauteilen unter Belastung, sowie deren mechanische Spannung zu berechnen. Dabei wird eine gegebenes System diskretisiert, indem sie in einfache Teilaufgaben zerlegt wird. Im Fall einer Crashsimulation würden die einzelnen Bauteile (Türen, Träger, usw.) eines Fahrzeuges durch eine Vielzahl von finiten Elementen ersetzt werden. Bild 3.3 zeigt Elemente für zweidimensionale und dreidimensionale Aufgaben.

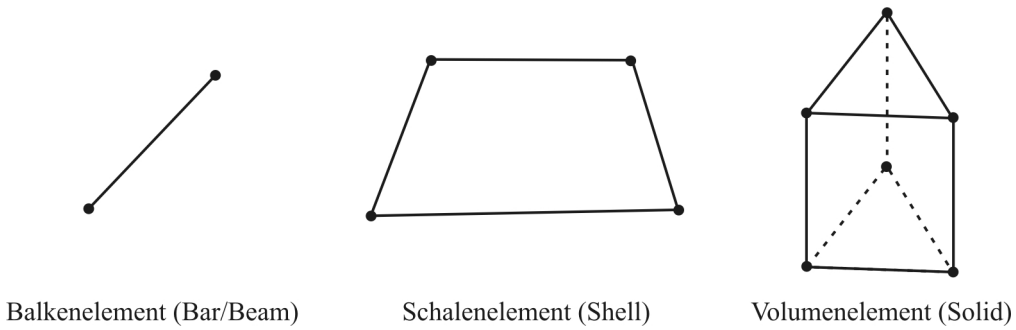

Bild 3.3: Beispiele von FEM-Elementen

Jedes Element besitzt an seinen Eckpunkten Knoten. Durch die Knoten wird die Position eines jeden Elements und die Verbindung mit dem Nachbarelement beschrieben. Die Summe der Elemente und Knoten bildet das FEM-Netz. (s. Bild 3.4)

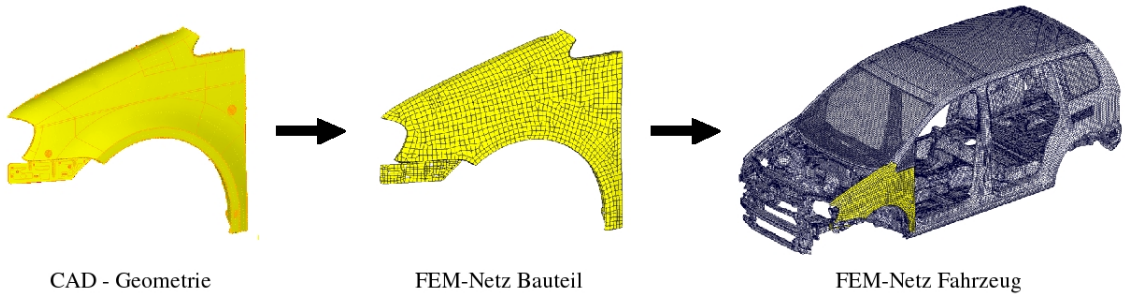

Bild 3.4: Entstehung des FEM-Netzes am Beispiel des VW Touran

Ein geometrisch komplexes und schwer zu berechnendes System wird in ein System aus kleinen lösbaren Teilaufgaben und deren Übertragungsfunktionen zerlegt, dass die Lösung nun aus der Ermittlung der einzelnen Teillösungen und deren Zusammenführung besteht.

3.3 PAM-Crash

Die französische ESI[4]-Group entwickelt seit Anfang der 80er Jahre das Simulationsprogramm PAM[5]-Crash. Dabei handelt es sich um einen nichtlinearen FEM-Solver. Es ermöglicht komplexe FEM-Modelle auf ihr Verhalten und ihre Auswirkungen nach verschiedenen Einwirkungen auf das Modell zu berechnen und wird hauptsächlich in der Automobilindustrie angewandt, um PKW-Unfälle zu simulieren. In PAM-Crash sind eine Vielzahl von Materialgesetzen und leistungsfähige Kontaktalgorithmen integriert, um eine realitätsnahe Simulation zu gewährleisten.
Als Eingabedatei für PAM-Crash dienen ASCII-Dateien[6], genannt Decks. In einem Input-Deck werden die einzelnen Bauteile (Parts), ihre Knoten (Nodes), sowie die Materialeigenschaften (MATER Cards) eines Modells beschrieben. Hinzu kommen weitere Daten (z.B. Anfangs- und Randbedingungen), die für die Simulation wichtig sind, hier aber nicht weiter ausgeführt werden sollen. Aus dem Deck errechnet PAM-Crash zwei Ergebnisdateien:

- DSY-Datei (Mesh-File): Ermöglicht die zeitlich aufgelöste grafische Darstellung der Simulation in z.B. dem Postprocessing Programm PAM-View (s. Kapitel 3.4).
- THP-Datei (Time-History-File): Wird u.a. zur Erzeugung von Ergebnis-Diagrammen einzelner Knoten oder anderer Größen in PAM-View verwendet.

[4]Engineering Systems International
[5]Program Applied Mechanics
[6]American Standard Code for Information-Interchange

3.4 PAM-View

PAM-View ist, wie auch PAM-Crash, ein Produkt der ESI-Group.
Es dient als Postprocessing-Werkzeug und ermöglicht dem Benutzer eine grafische Darstellung von Simulationsergebnissen in Diagrammform oder als zeitlich aufgelöstes 3D-Modell. Beide Darstellungsformen werden aus den zwei Ergebnisdateien (THP und DSY) von PAM-Crash erzeugt (s. Kapitel 3.3). Das 3D-Modell zeigt anschaulich wie sich, z.B. bei einem simulierten Fahrzeug-Crash, das Auto deformiert (s. Bild 3.5). Dabei kann der Benutzer u.a. einzelne Fahrzeugkomponenten ein- bzw. ausblenden.

Bild 3.5: Darstellung eines Fahrzeugs-Crash in PAM-View

Die Steuerung von PAM-View erfolgt auf verschiedene Arten. Neben einer GUI[7] steht dem Benutzer noch der Batch-Mode mit einer Session-Kommandodatei zur Verfügung. Im Batch-Mode erfolgt die Eingabe von Befehlen mittels einer Kommandosyntax. Soll eine Folge von Befehlen mehrmals wiederholt werden, wird die Session-Kommandodatei verwendet(s. Listing 3.1). In ihr wird die gewünschte Kommandofolge abgelegt und in PAM-View zur Ausführung importiert.

[7] Graphical User Interface

Listing 3.1: Beispiel einer PAM-View Session-Kommandodatei

```
PSCONSOLE:
6 955
END
DRAWSET:
OPENDSY: ./Unk_Mat/8161500/qs.THP
PSW3D:
506 245 "qs.THP"
END
CRVABSC:
CRVNODE:
...
...
...
```

3.5 Die Programmiersprache C/C++

Die Programmiersprache C verdankt seine Entstehung dem Betriebssystem UNIX. In den 70er Jahren wurde in den Bell Laboratories nach einer höheren Programmiersprache für das Betriebssystem UNIX gesucht. Bis dato wurden Programme und UNIX in der maschinennahen Sprache Assembler geschrieben.
Die neue Sprache C sollte effizient, gut zu portieren und leicht zu compilieren sein. Entwickelt wurde C schließlich von Brain W. Kernighan und Dennis M. Ritchie. Die Programmiersprache C verbreitete sich schnell in den folgenden Jahren.
Die Anforderungen für die Entwicklung von Software änderten sich mit der Zeit. Einfache Wartung und Wiederverwendung von Code traten in den Vordergrund und damit auch der Wunsch nach objektorientierten Konzepten.
Bjarne Stroustrup erweiterte C in den 80er Jahren um objektorientierte Konzepte. Aus C wurde die neue Sprache C++. C++ vereinte gegenüber C folgende Vorteile:

- hohe Wiederverwertbarkeit
- einfache Wartung
- gute Lesbarkeit

Trotz all der neuen Konzepte blieb C++ aber weiterhin voll abwärtskompatibel zu C.
Die Syntax und Semantik von C/C++ wurden in einer ISO/IEC-Norm festgelegt und sind als ANSI[8]-Standard bekannt. Der ANSI-Standard ermöglicht die Übertragbarkeit von einem nicht systemspezifischen Code auf eine andere Hardware und Compiler ohne eine Änderung am Code.

[8] American National Standards Institute

3.6 HTML

Die Sprache HTML[9] wurde 1989 von Tim Berners-Lee entworfen. HTML ist eine Auszeichnungssprache (Markup Language), die dazu dient, Informationen in Hypertexten im World Wide Web zu formatieren. Dabei steht neben klassischen Formatierungen (Schriftgröße, Schriftgrad, usw.), die Möglichkeit von Weiterleitungen (Links) im Vordergrund.
Die Formatierung einer Information wird mittels Tags vorgenommen, wobei zwei Tags immer eine Information einschließen und ein Element bilden (s. Listing 3.2).

Listing 3.2: Beispiel einer HTML-Datei

```
1 <html>
2 <body>
3 <b>Beispiel eines HTML-Tags üfr Fettdruck.</b>
4 </body>
5 </html>
```

Die Standardisierung von HTML ermöglicht es, eine Internetseite auf jedem System gleich darzustellen. Zur Zeit ist HTML als Version 4.01 verfügbar, wird jedoch nicht mehr weiterentwickelt. Aus einer Neuformulierung von HTML ist XHTML entstanden. XHTML ist abwärtskompatibel zu HTML und wurde zusätzlich um eine Modualisierung der Elemente erweitert.

[9] Hypertext Markup Language

4 Zustandsanalyse

Am Anfang jedes Projektes bzw. jeder Softwareentwicklung steht die Zustandsanalyse. Sie soll den momentanen und den gewünschten Arbeitszustand aufzeigen und grenzt das Projekt ein. Nach der Festlegung des gewünschten Zustandes, im folgenden Soll-Zustand genannt, sollte diese Grenze während der Projektabwicklung nicht verändert werden. Eine Änderung könnte im schlimmsten Fall zu einer Verwerfung oder einer enormen Veränderung des Programms führen. Aus diesen Gründen ist es sehr wichtig, den Soll-Zustand genau und nach reiflicher Überlegung zu definieren. Des Weiteren muss der Soll-Zustand mittels eines Lasten- und Pflichtenheftes exakt festgehalten werden (s. Kapitel 6).

4.1 Ist-Zustand

Zur Zeit werden Simulationsmodelle verwendet, die dem Fahrzeug entsprechende Materialmodelle beinhalten. Der Entwickler des Simulationsmodell weiß, welche Materialien er verwendet hat. Außerdem sind alle Materialien der VW Datenbank entnommen worden. So ist es dem Entwickler möglich, sollte es notwendig sein, gegebenenfalls ein Material durch ein anderes zu ersetzen.
Es kann aber vorkommen, das Simulationsmodelle von anderen Firmen entworfen werden, die dann von Volkswagen weiter bearbeitet werden müssen. In einem solchen Fall ist oft nicht bekannt, welche Materialien verwendet wurden. Die verwendeten Materialien an Hand ihrer Materialkarten zu identifizieren ist sehr aufwendig. Dies liegt daran, dass die Struktur einer Materialkarte sehr komplex ist und es zur Zeit ca. 140 unterschiedliche Materialmodelle gibt. So können zwei Karten mit sehr unterschiedlichen Parametern ein nahezu identisches Material beschreiben. Außerdem ist abzusehen, dass mit der Zeit neue Materialmodelle hinzukommen, die dann aber nicht berücksichtigt werden würden.

4.2 Soll-Zustand

Zur einfacheren Bearbeitung von Simulationsmodellen anderer Firmen, muss ein Weg gefunden werden, Materialien zu klassifizieren. Zu diesem Zweck soll das Programm „Idemat“ entstehen, welches automatisiert die unbekannten Materialkarten klassifiziert und durch bekannte aus der VW Datenbank ersetzt.
Da die Materialkarten, wie schon beschrieben, schwer zu vergleichen, soll mit jedem Material ein Zugversuch durchgeführt werden, der durch seine Ergebnisse eine einheitliche und vergleichbare Basis ermöglicht. Der Zugversuch wird mit dem Programm PAM-Crash, wie auch das eigentliche Fahrzeugmodell, simuliert. PAM-Crash legt als Ergebnis des Zugversuches eine THP-Datei ab. Diese Datei kann wiederum von dem Programm PAM-View eingelesen werden, welches daraus eine ASCII-Datei erzeugt, in der eine Spannungs-Dehnungs-Kurve als Tabelle abgelegt wird. Idemat liest die Tabellen ein und führt einen Vergleich der unbekannten Kurven mit denen der VW Datenbank durch. Wird ein passendes Material gefunden, wobei eine bestimmte Abweichung zu dem unbekannten Material nicht überschritten werden darf, wird die unbekannte Materialkarte durch die bekannte ersetzt. Als Ergebnis der gesamten Klassifizierung wird eine HTML-Datei angelegt, die alle ersetzten Materialien auflistet.

5 Theoretischer Lösungsansatz

In diesem Kapitel sollen für das zu entwickelnde Programm theoretische Lösungsansätze vorgestellt werden. Hier werden keine praktischen Umsetzungen innerhalb des Programms beschrieben, sondern die Grundlagen erläutert.
Um ein Material zu klassifizieren, muss für sämtliche Materialien, bekannt und unbekannt, eine einheitliche Vergleichsbasis geschaffen werden. Dies erfolgt durch die Simulation eines standardisierten Zugversuches. Auf der Grundlage der Simulationsergebnisse erfolgt dann der eigentliche Vergleich der Materialien
Nachstehend wird auf den Zugversuch und den Kurvenvergleich genauer eingegangen.

5.1 Zugversuch

Der Zugversuch gilt als Standardverfahren zur Werkstoffprüfung und ermöglicht die Ermittlung von Werkstoffkennwerten durch Dehnen von länglichen Proben mit einer kleinen Querschnittsfläche. Er ist in der Norm DIN EN 10002 definiert. Die Proben werden bis zum Bruch gezogen. Aus dem Ergebnis eines Zugversuchs, dem Spannungs-Dehnungs-Diagramm, können verschiedene Kenngrößen ermittelt werden, z.B. E-Modul, Elastizitätsgrenze, usw. (vgl. 3.1).
Bei dem Zugversuch nach DIN-Norm handelt es sich um einen realen Versuch. Für diese Arbeit wird der Versuch jedoch mit dem Programm PAM-Crash (s. 3.3) simuliert. Die Probe wird auf eine Größe von 10×10 mm festgelegt und hat eine Dicke von 1 mm. Viele Materialien weisen eine Dehnratenabhänigkeit auf, d.h. sie reagieren unterschiedlich auf die Geschwindigkeit, mit der die Probe gezogen wird (s. Bild 5.1).

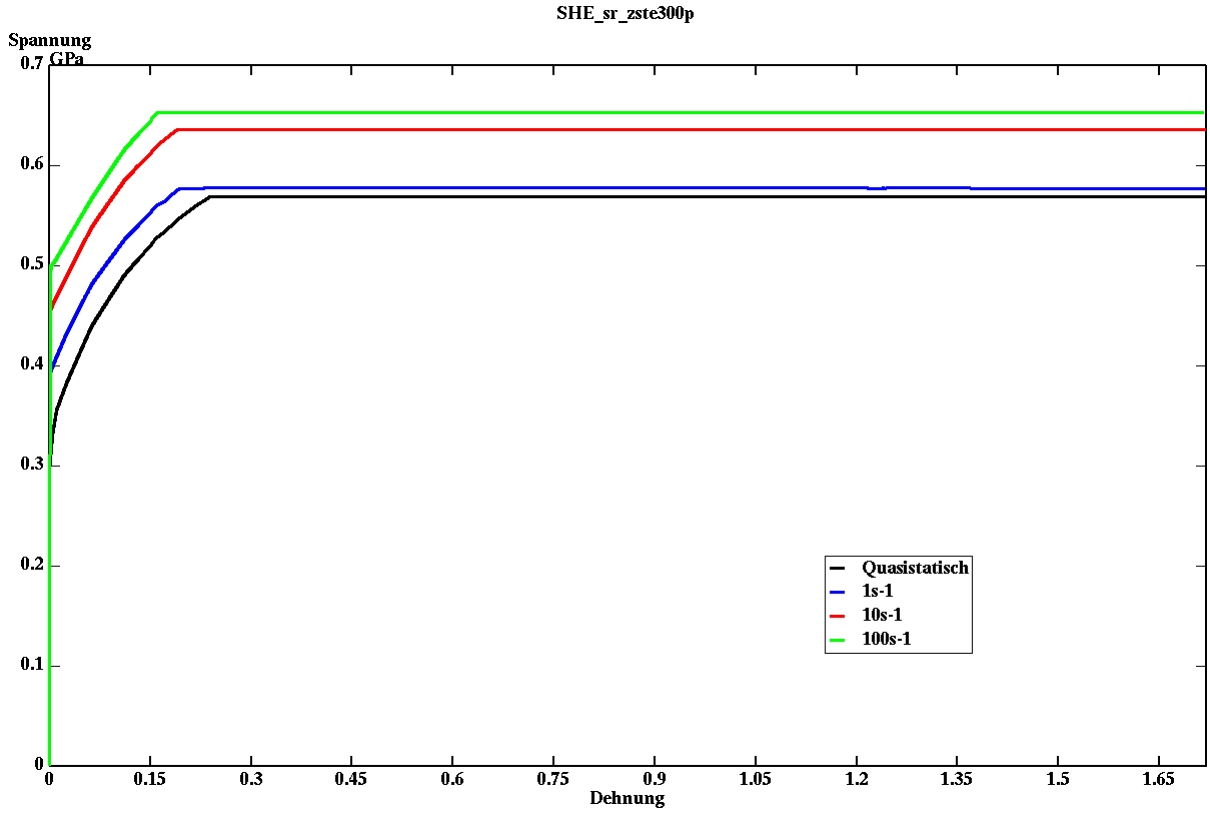

Bild 5.1: Auswirkung der Zuggeschwindigkeit auf das Spannungs-Dehnungs-Diagramm

Deshalb soll jede Probe mit vier unterschiedlichen Geschwindigkeiten gezogen werden, damit das unbekannte Material richtig klassifiziert werden kann:

- Quasistatisch
- $\frac{1}{sec}$
- $\frac{10}{sec}$
- $\frac{100}{sec}$

Die Ergebnisse der Zugversuche liegen nicht sofort als Spannungs-Dehnungs-Diagramme vor, sondern nur als THP-Dateien. Das Programm PAM-View (s. 3.4) liest die THP-Dateien ein und erzeugt daraus die Spannungs-Dehnungs-Diagramme und speichert sie als Datei in Form einer Tabelle ab. Diese sind auch die Grundlage für den im Folgenden vorgestellten Kurvenvergleich.

5.2 Kurvenvergleich

Ein Kernproblem dieser Arbeit ist es, zwei Kurven miteinander zu vergleichen. Selbstverständlich wird das unbekannte Material mit mehr als einem bekannten Material verglichen. Allerdings wird das folgende Verfahren immer nur auf zwei Kurven gleichzeitig angewendet, auf die Unbekannte $f_u(x)$ und die Bekannte $f_b(x)$.

5.2.1 Grundbedingungen

Damit zwei Kurven erfolgreich miteinander verglichen werden können, müssen die folgenden Grundbedingungen gelten.
Zum einen müssen beide Kurven das Ergebnis des selben Zugversuchs sein (s. Kapitel 5.1).
Zum anderen müssen die Funktionswerte für jede Kurve an den gleichen Stützstellen erhoben werden. Die Abtastrate muss also bei jedem Versuch übereinstimmen, was letztendlich gleiche Abzissen-Werte ergibt.
Für die im Folgenden vorgestellten Vergleichsmethoden liegen die Kurven als Tabelle mit Abzissen- und Ordinaten-Werten vor.

5.2.2 Vergleichsmethoden

Im Vorfeld gilt es zu klären, welche Vergleichsmethode die beste ist bzw. das beste Ergebnis liefert. Dazu wurden drei Methoden entwickelt, die hier kurz vorgestellt werden sollen.

5.2.2.1 Erste Vergleichsmethode

Die erste Methode ist ein sehr einfaches und schnelles Verfahren, da wenige mathematische Schritte erforderlich sind.
Ausgehend davon, dass die Werte der Abzisse von $f_u(x)$ und $f_b(x)$ identisch sind, bedeutet dies, dass $y_u = f_u(x_i)$ und $y_b = f_b(x_i)$ auf einer Senkrechten liegen. Folglich bildet die absolute Differenz aus y_u und y_b den Abstand a_i im Punkt i:

$$a_i = |f_u(x_i) - f_b(x_i)| \qquad \text{mit } i = 0, 1, 2, ..., n \tag{5.1}$$

Als Ergebnis eines Vergleichs wird das arithmetische Mittel gebildet, wobei n die Anzahl der Stützstellen ist:

$$\bar{a} = \frac{\sum_{i=0}^{n} a_i}{n} \tag{5.2}$$

Allerdings spiegelt $\bar{a}$ in der Regel nicht den kürzesten Abstand der beiden Kurven wieder. An dem Beispiel in Bild 5.2 lässt sich erkennen, dass der kürzeste Abstand b in beiden Fällen identisch ist, aber a_i sich stark unterscheidet.

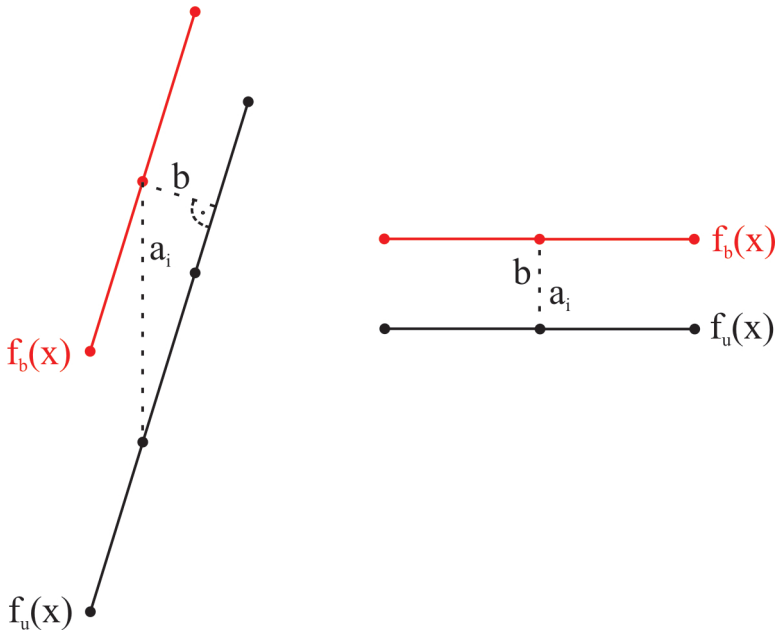

Bild 5.2: Kurvenabstand (Methode 1)

Fazit: Kurvenabschnitte mit großer Steigung haben einen überproportionalen Einfluss auf das Vergleichsergebnis.

5.2.2.2 Zweite Vergleichsmethode

Bei der zweiten Methode handelt es sich um das sogenannte Korridorverfahren. Das Korridorverfahren bietet die Möglichkeit, eine relative Bewertung der Abstände einer Kurve zu einer anderen vorzunehmen. Dabei wird eine Kurve, hier $f_u(x)$, als Bezugskurve definiert. Der Korridor wird nun mit einem Abstand a_k um die Bezugskurve gelegt (s. Bild 5.3).

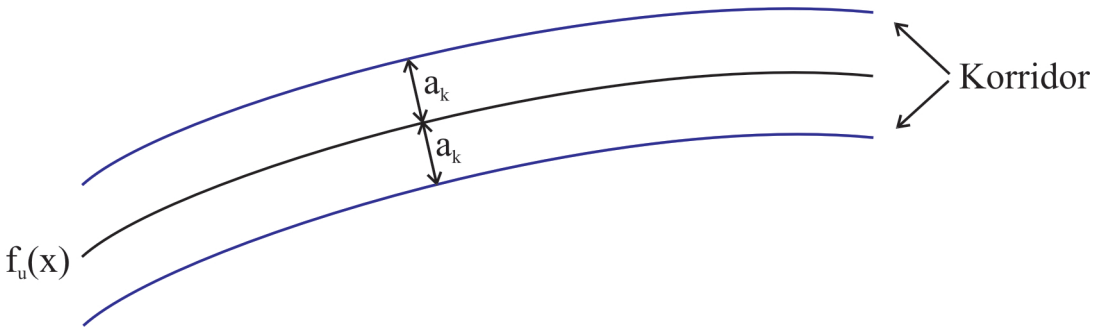

Bild 5.3: Korridor (Methode 2)

Die zu vergleichende Kurve $f_b(x)$ wird wie in Bild (5.4) über $f_u(x)$ gelegt.

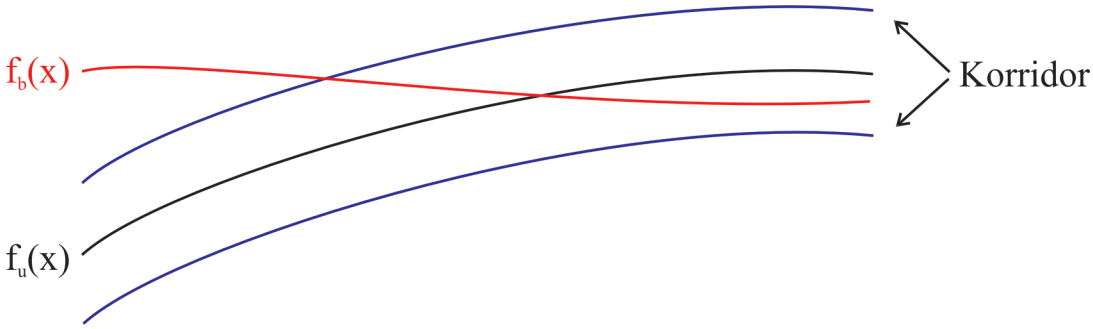

Bild 5.4: Überlagerung der Kurven $f_u(x)$ und $f_b(x)$ (Methode 2)

Im Bild ist ebenfalls zu erkennen, dass $f_b(x)$ nun teilweise innerhalb und außerhalb des Korridors liegt. Eine Bewertung wird wie folgt durchgeführt:

- Liegt $f_b(x)$ außerhalb des Korridors oder direkt auf dem Korridorrand, ist die Bewertung 0.
- Treffen $f_b(x)$ und $f_u(x)$ exakt zusammen, wird eine Bewertung mit 1 vorgenommen.
- Verläuft $f_b(x)$ zwischen Korridorrand und $f_u(x)$, erfolgt eine Bewertung zwischen 0 und 1. Die Bewertungsskala muss nicht unbedingt linear sein.

Wie aber schon in der ersten Methode tritt auch hier das Problem der Abstandsbestimmung auf. Zum einen um den Korridor zu erzeugen, zum anderen um die Lage von $f_b(x)$ zu ermitteln.

Fazit: Die 2. Methode ist sehr aufwendig. Sie bietet aber die Möglichkeit einer relativen Bewertung.

5.2.2.3 Dritte Vergleichsmethode

Die dritte und letzte vorgestellte Vergleichsmethode basiert auf dem Ansatz immer den kürzesten Abstand a von einem Punkt P_u auf der Kurve $f_u(x)$ zu ermitteln (s. Bild 5.5).

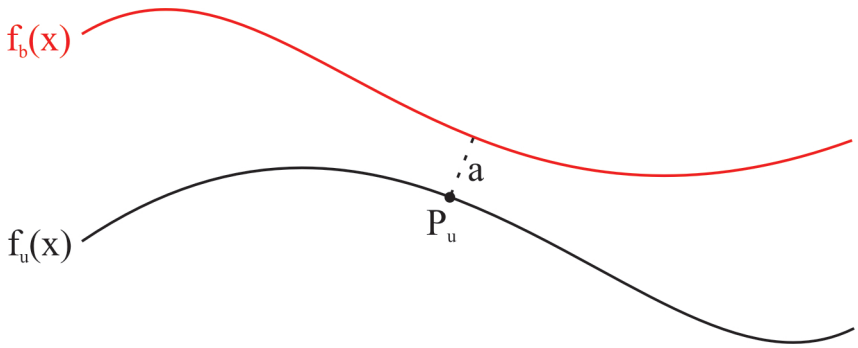

Bild 5.5: Abstand Punkt - Kurve (Methode 3)

Allerdings liegen beide Kurven, wie schon beschrieben, in Form von Tabellen mit Abzissen- und Ordinaten-Werten vor. Daraus ergibt sich eine Kurve, die nur als eine Vielzahl von Punkten, aber nicht als eine Funktion von x vorhanden ist. Bild 5.6 zeigt, dass es nur möglich ist, von einem Punkt P_u auf $f_u(x)$ zu einem Punkt P_b auf $f_b(x)$ den Abstand zu bestimmen.

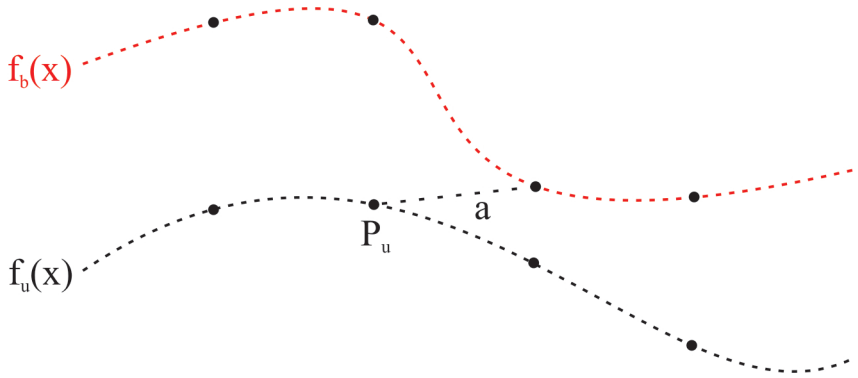

Bild 5.6: Abstand Punkt - Punkt (Methode 3)

Jedoch kann der fehlende Kurvenabschnitt zwischen den Punkten P_b an der Stelle i und $i+1$ interpoliert werden, indem durch beide Punkte eine Gerade g gelegt wird. In Bild 5.7 ist zu erkennen, dass nun auch ein kürzerer Abstand im rechten Winkel zu einem interpolierten

Punkt auf g zwischen den beiden bekannten Punkten ermittelt werden kann.

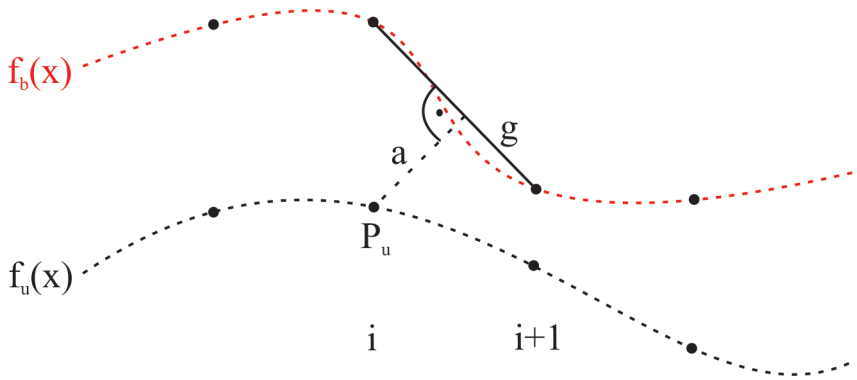

Bild 5.7: Abstand Punkt - Gerade (Methode 3)

Werden diese beiden Möglichkeiten in Betracht gezogen, kann für jeden Punkt auf $f_u(x)$ der kürzeste Abstand zu $f_b(x)$ berechnet werden.
Das arithmetische Mittel über alle kürzesten Abstände bildet ein Maß für die Übereinstimmung beider Kurven.

Fazit: Diese Vergleichsmethode ähnelt in gewisser Weise der 1. Methode. Ihre Ergebnisse werden aber nicht durch die Steigung der Kurven beeinflusst. Außerdem ist sie im Vergleich zu Methode Zwei nicht so aufwendig. Dementsprechend fällt die Wahl der Vergleichsmethode auf die vorgestellte Methode.

In den folgenden Unterkapiteln wird auf die Mathematik, sowie auf den funktionellen Ablauf von Methode Drei eingegangen.

5.2.3 Mathematische Grundlagen des Kurvenvergleichs

Als Verfahren für den Kurvenvergleich fiel die Wahl auf die dritte Methode.
Für die Berechnung des Abstandes bedarf es mathematischer Algorithmen, die im Weiteren definiert werden.

5.2.3.1 Punkt-Punkt-Abstand

Der Abstand eines Punktes zu einem anderen lässt sich im zweidimensionalen Raum einfach bestimmen. Wenn sämtliche Koordinaten beider Punkte bekannt sind, errechnet sich der

Abstand nach dem Satz des Pythagoras[1]:

$$c^2 = a^2 + b^2 \tag{5.3}$$

Wird dieses auf einen Punkt P_1 und P_2 mit ihren x- und y-Koordinaten angewandt, ergibt sich für den Abstand a:

$$a = \overline{P_1P_2} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{5.4}$$

5.2.3.2 Punkt-Gerade-Abstand

Um den Abstand zwischen einem Punkt P_1 und einer Geraden g zu bestimmen, muss zunächst g ermittelt werden. Dafür werden zwei Punkte, P_2 und P_3 benötigt. Durch Anwenden der Zwei-Punkt-Form

$$\vec{g}(\lambda) = \vec{r_2} + \lambda\overrightarrow{P_2P_3} \tag{5.5}$$

$$= \vec{r_2} + \lambda(\vec{r_3} - \vec{r_2} \tag{5.6}$$

$$= \vec{r_2} + \lambda\vec{z} \tag{5.7}$$

wird der Ortsvektor von g bestimmt (λ ist ein Parameter aus $\Re$) und $\vec{z}$ ist der Richtungsvektor der Geraden (vgl. Bild 5.8).

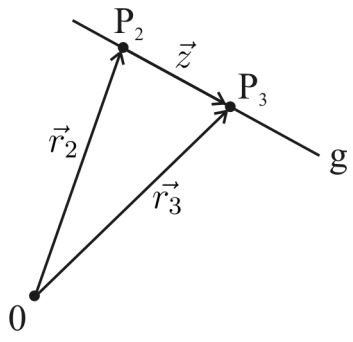

Bild 5.8: Zwei-Punkt-Form

Nachdem $\vec{g}(\lambda)$ ermittelt worden ist, kann der Abstand a des Punktes P_1 zu $\vec{g}$ berechnet werden. a ist immer die Strecke des von $\vec{g}$ fallenden Lots zu P_1. Es ist zugleich auch der

[1]Pythagoras von Samos (6. Jahrhundert v. Chr.), griechischer Mathematiker und Philosoph

kürzeste Abstand. Bild 5.9 soll dies veranschaulichen.

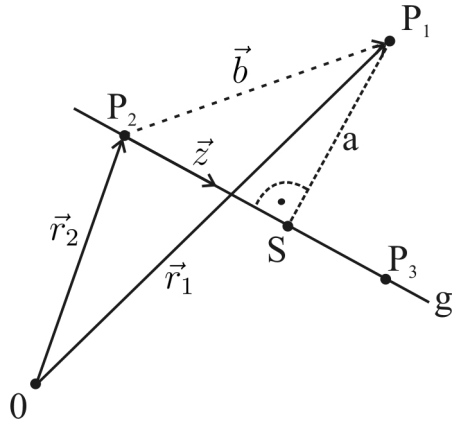

Bild 5.9: Abstand Punkt-Gerade

Aus dem Ortsvektor $\vec{r}_1$ des Punktes P_1 und der Geraden $\vec{g}$ lässt sich der Abstand a mit

$$a = \frac{|\vec{z} \times (\vec{r}_1 - \vec{r}_2)|}{|\vec{z}|} \tag{5.8}$$

bestimmen.
a ist nur dann ein verwendbarer Abstand, wenn S (Schnittpunkt von $\vec{g}$ mit dem Lot) zwischen P_2 und P_3 liegt.
Zur Bestimmung von S muss zunächst der Vektor $\vec{b}$ auf $\vec{z}$ projiziert werden:

$$\vec{b_z} = (\frac{\vec{z} \cdot \vec{b}}{|\vec{z}|^2}) \qquad \text{mit } \vec{b} = \vec{r}_1 - \vec{r}_2 \tag{5.9}$$

S ergibt sich damit wiederum aus

$$S = \vec{r}_2 + \vec{b_z} \tag{5.10}$$

Damit a verwertbar ist, muss also die Bedingung

$$x_2 \leq x_S \leq x_3 \tag{5.11}$$

erfüllt sein.

5.2.4 Ablauf eines Kurvenvergleichs

Für den kürzesten Abstand von einem Punkt P_u auf der Kurve $f_u(x)$ zu $f_b(x)$ könnte theoretisch jeder Punkt auf der Kurve $f_b(x)$ infrage kommen. Dies würde bedeuten, dass pro Punkt auf $f_u(x)$ die gesamte Kurve $f_b(x)$ betrachtet werden müsste, was wiederum zu einem enormen Rechenaufwand führen würde. Damit dieses vermieden wird, wurde ein dynamisches Verfahren entwickelt.
Der PAP[2] in Bild 5.11 sowie das Diagramm in Bild 5.10 sollen die weiteren Ausführungen veranschaulichen und sollten parallel dazu betrachtet werden.

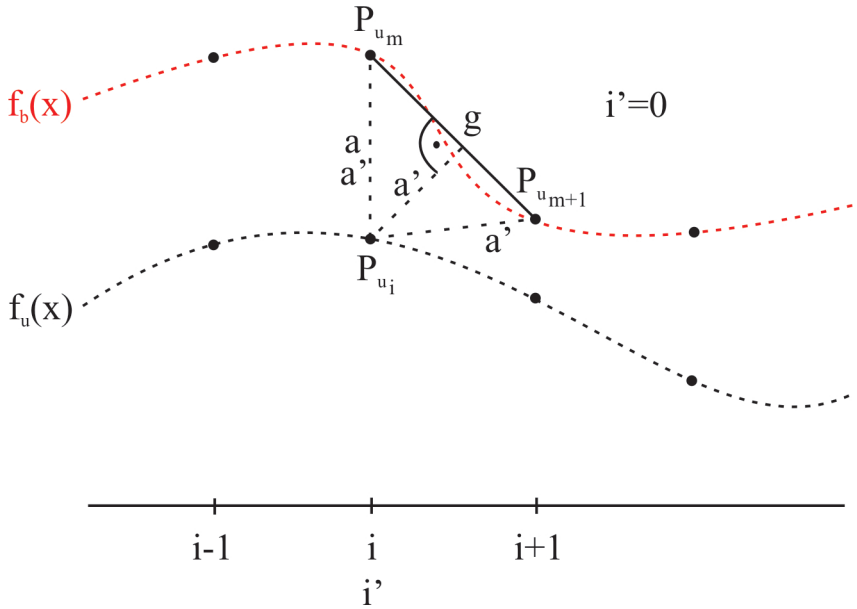

Bild 5.10: Kurvenvergleich - Suche in Rechtsrichtung

Die Dynamik der Suche nach dem kürzesten Abstand besteht darin, den Suchradius im Verlauf immer weiter zu verkleinern. Außerdem kann davon ausgegangen werden, dass der Punkt P_b auf $f_b(x)$, zu dem der kürzeste Abstand existiert, im näheren Umfeld von P_u liegt. Deshalb wird die Suche auch bei P_b an der Stelle i begonnen und nicht bei $i = 0$. Als Erstes wird also der Abstand

$$a = |f_b(x_i) - f_b(x_i)| \tag{5.12}$$

berechnet. a ist zugleich der Suchradius, da sich nur innerhalb dieses Radius um P_u noch ein kürzerer Abstand finden läßt. Aus a lassen sich die Suchgrenzen i_{min} und i_{max} bestimmen.

[2]Programm-Ablauf-Plan

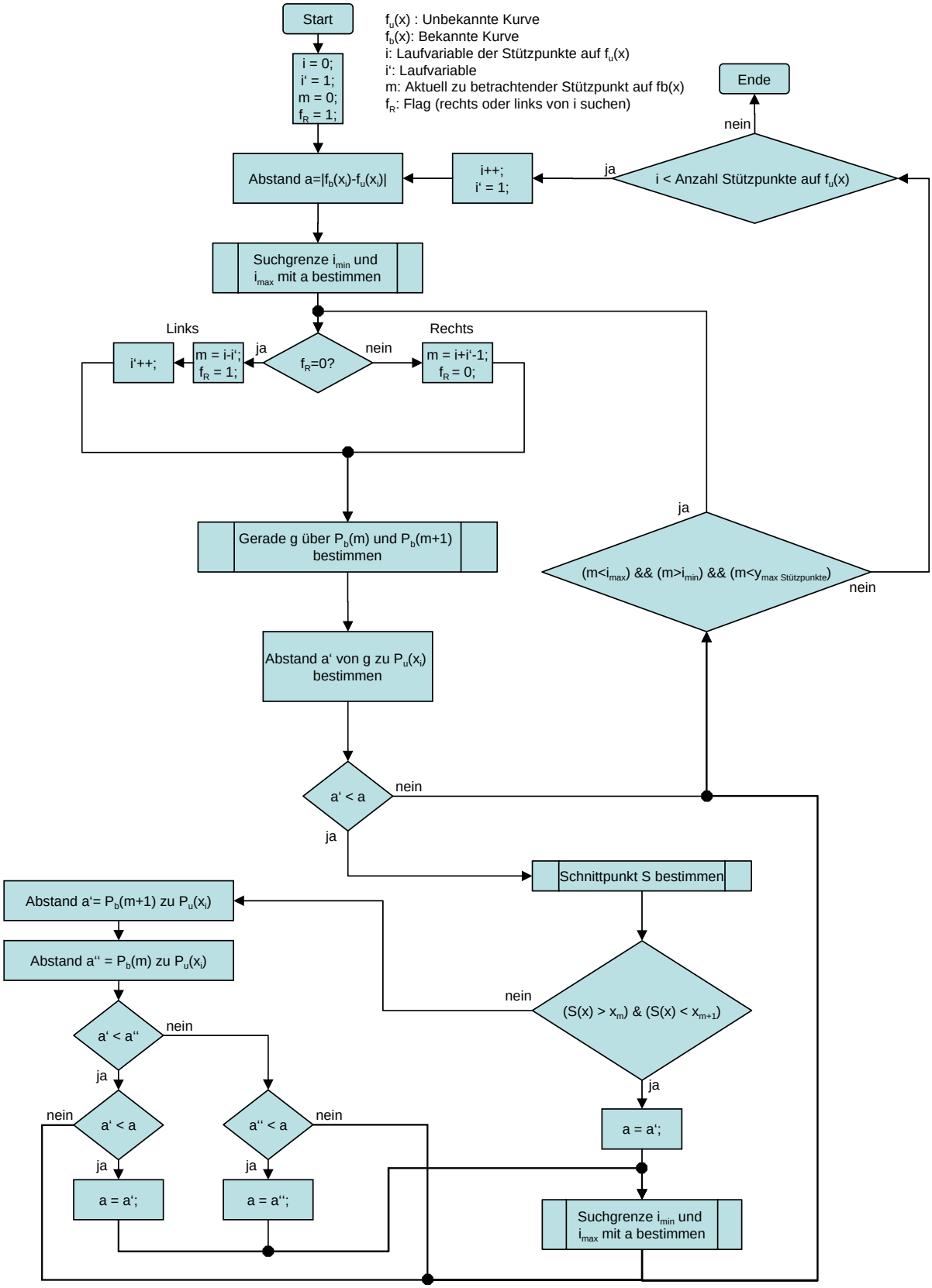

Bild 5.11: PAP - Kurvenvergleich

Sollte es einen noch kürzeren Abstand als a geben, wird weiterhin davon ausgegangen, dass dieser sich vermutlich näher an i als an i_{min} oder i_{max} befindet. Darum wird alternierend rechts und links von i gesucht.
Für die alternierende Suche müssen zwei weitere Variablen eingeführt werden:

- $i' = 1, 2, 3, ..., n$: Laufvariable innerhalb einer Suche. Diese wird bei jedem Suchwechsel in Linksrichtung inkrementiert.
- m : Der aktuell zu betrachtende Stützpunkt auf $f_u(x)$. Startbedingung $m = 0$.

m errechnet sich je nach Richtung unterschiedlich:

$$\text{Links: } m = i - i' \tag{5.13}$$

$$\text{Rechts: } m = i + i' - 1 \tag{5.14}$$

Nachdem m bestimmt wurde, wird immer erst die Gerade g über P_{b_m} und $P_{b_{m+1}}$ gebildet und der Abstand a' von P_{u_i} zu g bestimmt. Sollte a' kürzer sein als das aktuelle a, wird geprüft, ob der Abstand Gültigkeit besitzt (vgl. Kapitel 5.2.3.2). Ist Gültigkeit gegeben, wird

$$a = a' \tag{5.15}$$

und die Suchgrenzen neu festgelegt. Bei Ungültigkeit wird a' verworfen und geprüft, ob sich zu P_{b_m} oder $P_{b_{m+1}}$ ein kleinerer Abstand finden lässt. Sollte die Prüfung erfolgreich sein, werden auch hier die Grenzen neu bestimmt. Ist kein neuerer kleinerer Abstand gefunden worden, wird die Suchrichtung gewechselt und weiter gesucht, bis die beiden Suchgrenzen erreicht sind. Nach dem Erreichen der Grenzen wird i inkrementiert und eine neue Suche beginnt.
Sind alle Punkte auf $f_u(x)$ durchlaufen worden, wird der arithmetische Mittelwert $\overline{a}$ aus allen gefundenen a gebildet.

5.2.5 Gewichtung einer Kurve

Es gibt Bereiche des Spannungs-Dehnungs-Diagramms, die für einen Vergleich „interessanter“ sind als andere. So sind die Abschnitte im Bereich des E-Moduls sowie der Elastizitätsgrenze die wichtigsten Abschnitte für einen Vergleich, da dort die entscheidenden Merkmale eines Materials definiert sind. Zudem sind hier bereits Unterschiede zwischen den Materialien zu erkennen. Folglich ist es sinnvoll, den Anfangsabschnitt der Kurve genauer zu betrachten,

bzw. höher zu bewerten.
Zu diesem Zweck wurde das Diagramm in drei Bereiche unterteilt:

1. 1,25% der Kurve → Gewichtung: 100%
2. 11,25% der Kurve → Gewichtung: 10%
3. 87,5% der Kurve → Gewichtung: 1%

Dies bedeutet, dass der erste bzw. der linke Bereich der Kurve 1,25% der gesamten Kurve entspricht und jeder Punkt auf der Abzisse beim Vergleich betrachtet werden muss. Im zweiten Bereich ist es nur noch jeder zehnte.
Die Gewichtung soll auf das Vergleichsverfahren keinen Einfluss haben. Aus diesem Grund wird die tabellarische Darstellung der Kurve, unter Berücksichtigung der Gewichtung, gespeichert. Im Fall eines Diagramms mit 80000 Stützpunkten (entspricht der verwendeten Spannungs-Dehnungs-Diagramme in dieser Arbeit) bedeutet dies folgendes:

- Bereich 1: 1000 Stützpunkte
- Bereich 2: 900 Stützpunkte
- Bereich 3: 700 Stützpunkte

Es stehen also nur noch 2600 Punkte für den Vergleich zur Verfügung. Dies hat neben der Gewichtung außerdem den Vorteil, dass eine wesentlich kleinere Datenmenge zu verarbeiten ist und damit auch die Vergleichsgeschwindigkeit steigt.

6 Planungs- und Definitionsphase

Bevor ein Softwareprodukt implementiert wird, durchläuft es eine Analyse- und Defintionsphase. Sie dient dazu, ein Produkt zu beschreiben. Der Auftraggeber äußert seine Wünsche bzw. Anforderungen bezüglich der zu entwickelnden Software und schreibt sie in einem Lastenheft nieder. Es handelt sich dabei um eine vereinfachte Beschreibung. Oft wird in diesem Zusammenhang auch der Name „Grobes Pflichtenheft“ genannt. Das Lastenheft beinhaltet keine Details in Bezug auf die Entwicklung der einzelnen Funktionalitäten, sondern gibt nur an, welche Kernfunktionen das Produkt beinhalten soll.
Im darauf folgenden Schritt wird das Pflichtenheft definiert. Es ist eine Weiterführung des Lastenhefts und dient dazu, Funktionalitäten des Produkts im Detail zu klären. Außerdem werden die Zielbestimmungen, Produktdaten, Produktleistungen usw. genauer ausgeführt. Das Pflichtenheft ist ein Bestandteil des Vertrags zwischen Auftraggeber und Entwickler und ist Grundlage für die abschließende Überprüfung, ob das Produkt allen Anforderungen entspricht.
Im Folgenden wurde als Grundlage für die Erstellung dieses Lasten- und Pflichtenheftes die Ausführungen von Prof. Dr.-Ing. habil. Helmut Balzert herangezogen. [1] Der Umfang der Analyse- und Definitionsphase wurde dem zu entwickelnden Programm angepasst.

6.1 Lastenheft

6.1.1 Zielbestimmungen

Es soll ein Programm, im folgenden „Idemat“ genannt, entwickelt werden.
Die Hauptfunktion von Idemat besteht darin, unbekannte Materialien zu klassifizieren. Grundlage dafür ist, dass mit jedem bekannten und unbekannten Material ein Zugversuch simuliert wird. Als Ergebnis liefern die Simulation Spannungs-Dehnungs-Kurven. Mit den Kurven wird eine Basis für die Klassifizierung geschaffen. Bevor eine Klassifizierung vorgenommen werden kann, müssen für alle Materialien, die in der VW-Datenbank gespeichert sind, die Zugversuche durchgeführt werden, wobei die Ergebnisse für eine spätere Verwendung dauerhaft

gespeichert werden.
Der User soll Idemat nutzen können, um ein PAM-Crash-Deck mit unbekannten Materialien einzulesen. Dabei werden ebenfalls mit den unbekannten Materialien die Zugversuche durchgeführt. Da die Ergebnisse der bekannten und unbekannten Materialien auf den gleichen Versuchen basieren, können diese nun verglichen und das unbekannte Material klassifiziert werden. Wird in der VW-Datenbank ein Material gefunden, welches dem Unbekannten am ähnlichsten ist und innerhalb einer geforderten max. Abweichung liegt, so soll das Unbekannte im PAM-Crash-Deck durch das Bekannte ersetzt werden.
Die Klassifizierungsergebnisse sollen dem User am Ende des Programmdurchlaufs anschaulich dargestellt werden. Dazu ist eine tabellarische sowie grafische Darstellungsform zu wählen.
Das gesamte Programm soll möglichst automatisiert ablaufen.

6.1.2 Produkteinsatz

Anwendungsfeld: Pkw-Crashsimulation
Anwender: Entwicklungsingenieur/Crashberechner

6.1.3 Produktfunktionen

/LF 10/ Einlesen von Material-Dateien und PAM-Crash-Decks
PAM-Crash-Decks sollen nach Materialkarten, den dazugehörigen Funktionen und Partkarten durchsucht werden. Die gefundenen Abschnitte sollen für die Weiterverarbeitung extrahiert werden. Ebenfalls sollen die Materialnummern gelesen und für die spätere Nutzung gepuffert werden. Geschachtelte Decks, die aus mehreren Dateien bestehen, müssen ebenfalls durchsucht werden.

/LF 20/ PAM-Crash Input-Decks für die Zugversuche vorbereiten
Für die Durchführung der Zugversuche werden vorbereitete PAM-Crash Input-Decks verwendet. Diese müssen für das jeweilige Material angepasst werden.

/LF 30/ Aufruf von PAM-Crash
Starten von PAM-Crash mit den vorbereiteten PAM-Crash-Decks für den simulierten Zugversuch.

/LF 40/ PAM-View Session-Kommandadateien bearbeiten
PAM-View nutzt Kommandodateien für einen automatisieren Programmablauf. Diese müssen je nach gewünschter Funktion angepasst werden.

/LF 50/ Aufruf von PAM-View
Ausführen von PAM-View mit den vorher bearbeiteten Kommandodateien zum Generieren der Spannungs-Dehnungs-Kurven und der jeweiligen Kurvenbilder.

/LF 60/ Materialdatenbank
In einer Datenbank sollen alle bekannten Materialen abgelegt werden. Dazu gehören die Spannungs-Dehnungs-Kurven und deren Bilder.

/LF 70/ Erstellen der Materialdatenbank
Unter Angabe eines Verzeichnisses soll mit allen darin enthaltenen Materialdateien eine Datenbank erstellt werden. Dabei werden die simulierten Zugversuche durchgeführt und die Spannungs-Dehnungs-Kurven, die dazugehörigen Bilder der Kurven, sowie die Materialliste erzeugt.

/LF 80/ Liste der vorhandenen Materialien in der Materialdatenbank
Alle vorhanden Materialien werden als aktuelle Liste in einer Datei gespeichert.

/LF 90/ Konfigurationsdatei des Programms
Verschiedene Einstellungen, die den Programmablauf betreffen, werden in einer Konfigurationsdatei mit vorgegebener Syntax gespeichert.

/LF 100/ Klassifizieren der Materialien
Das unbekannte Material wird mit den bekannten Materialien in der Datenbank verglichen. Die Klassifizierung wird entsprechend der in der Konfigurationsdatei vordefinierten Parameter durchgeführt.

/LF 110/ Ersetzen von Materialkarten
Die unbekannten Materialkarten sollen durch bekannte Materialkarten ersetzt werden, die die beste Übereinstimmung aufweisen. Über eine Variable in der Konfigurationsdatei soll die max. geforderte Abweichung gesetzt werden können. Die ersetzten Materialkarten sollen auskommentiert werden.

/LF 120/ Templatedatei erzeugen
Eine Templatedatei ist eine Steuerdatei für die Generierung eines Input-Decks mit einem Vernetzungstool. Sie beinhaltet Angaben über den Materialtyp und die Blechdicke der einzelnen Bauteile.

/LF 130/ Ausgabe der Klassifizierungsergebnisse
Nach dem Klassifizieren der Materialien sollen die Ergebnisse repräsentativ dargestellt werden. Dazu ist eine tabellarische sowie grafische Darstellungsform zu wählen.

/LF 140/ Logfile schreiben
Bei jedem Aufruf des Programms soll ein Logfile geschrieben werden. Das Logfile wird vor jedem Programmdurchlauf geleert und neu geschrieben.

6.1.4 Produktdaten

Während des Programmablaufs werden folgende Daten temporär oder dauerhaft gespeichert:

/LD 10/ Zugversuchergebnisse - THP-Dateien (Wahlweise temporär oder dauerhaft)

/LD 20/ Spannungs-Dehnungs-Kurven der Zugversuche

/LD 30/ Materialliste

/LD 40/ Konfigurationsdatei des Programms

/LD 50/ Klassifizierungsergebnisse

/LD 60/ Templatedatei

/LD 70/ Logfile

6.1.5 Produktleistungen

/LL 10/ Das Programm soll in der Konsole ausgeführt werden und eine einfache Bedienung vorweisen.

/LL 20/ Die Darstellung der Klassifizierungsergebnisse soll in einem systemunabhängigen Format erfolgen.

/LL 30/ Die PAM-Crash-Decks für die Zugversuche sollen durch andere ersetzt werden können.

/LL 40/ Das Ändern der Darstellungsform der Kurven soll möglich sein.

/LL 50/ Während des Programmablaufs sollen Fehler erkannt werden.

6.1.6 Qualitätsanforderungen

Tabelle 6.1: Qualitätsanforderungen Lastenheft

Produktqualität	+	o	-
Funktionalität	X		
Zuverlässigkeit	X		
Benutzbarkeit		X	
Effizienz	X		
Änderbarkeit		X	
Übertragbarkeit			X

6.2 Pflichtenheft

6.2.1 Zielbestimmungen

6.2.1.1 Musskriterien

- Auslesen von Materialdateien und PAM-Crash-Decks
- Durchführung der Zugversuche mittels PAM-Crash
- Erzeugung der Spannungs-Dehnungs-Kurven durch PAM-View
- Hinzufügen, Entfernen und Ändern von Materialien in der Datenbank
- Klassifizierung des unbekannten Materials
- Ausgabe der Klassifizierungsergebnisse
- Erzeugen einer Templatedatei

6.2.1.2 Wunschkriterien

- Darstellung der Fließgrenze eines Materials im Bild der Spannungs-Dehnungs-Kurve

6.2.2 Produkteinsatz

Anwendungsfeld: PKW-Crashsimulation
Anwender: Entwicklungsingenieur/Crashberechner

6.2.3 Produktschnittstellen

1. PAM-Crash Input-Decks
2. PAM-View Session-Kommandadateien
3. Konfigurationsdatei

6.2.4 Produktfunktionen

/F 10/ Einlesen von Material-Dateien und PAM-Crash Input-Decks
Material-Dateien und PAM-Crash Input-Decks sollen nach Materialkarten und den dazugehörigen Funktionen durchsucht werden. Die gefundenen Abschnitte sollen für die Weiterverarbeitung extrahiert werden. Ebenfalls sollen die Materialnummern gelesen und für die spätere Nutzung gepuffert werden. Geschachtelte Material-Dateien, die aus mehreren Dateien bestehen, müssen ebenfalls durchsucht werden. Das Listing 6.1 zeigt einen Auszug einer möglichen Materialdatei.

Listing 6.1: Beispiel einer Materialdatei bzw. eines PAM-Crash-Decks

```
MATER /  8133400     103     7.85000e-06       0       1       1       0
               0       0       0       0       0       0      1.       0
NAME ZSTE 340
      206.CURVE            0.3                 0.01      0.01      0.01  0.833333
   8133400   8133400   8133400   8133400   8133400   8133400         0         0
       0.0 1.0000e-4 5.0000e-4     0.001     0.005   100000.       0.0       0.0
       0.0       0.0       0.0
                           0.0       0.0       0.0       0.0       0.1       0.0
      EPMX      0.20
FUNCT /  8133400       7      1.      1.     0.0     0.0
NAME Curve_8133400
$ Units:  None / Stress
                             0.0          0.3336
                         0.00337            0.34
                        0.016277            0.38
                        0.035637            0.41
                        0.085492            0.44
                        0.139383           0.463
                         0.18932           0.476
SECURE/ FUNCT   9100003 v#8Fv1-
AM8QuN'w<9_b6}>xx3p>
,Wj+H^3F^UGB@0&y^^frlTrHtH^VlpO0vxhhhb?2\hFHz6>6nJhPjv~$
jEe]d_!x"_fRHbn^DlM
"hE&w~)R@BbkrynJv'JFQ\a
@LflvxhhhblTL|RHbn^DlxTPPjzfr*||LZR^@frXxVfzFxLK
`R`pB@Bt@LZR^@frXxVfzFHnJjzA$Bz.pB@Bt@LZR^@fr*|#
END_SECURE
INCLU /  Frontklappe.inc
```

Eine Materialkarte beginnt mit dem Schlüsselwort „MATER␣/␣“. Darauf folgt die Materialnummer. Die Materialnummer muss gespeichert werden, da sie später für die Zugversuche in die PAM-Crash-Decks eingefügt werden muss. Einige Materialkarten besitzen Funktionen. Funktionen beginnen mit „FUNCT␣/␣“, gefolgt von einer Nummer, die in der dazugehörigen Materialkarte referenziert wird. Ebenso ist es möglich, dass Materialkarten oder Funktionen verschlüsselt dargestellt werden. Ist dies der Fall, beginnen beide mit „SECURE/␣“, gefolgt von „MATER“ und Materialnummer oder „FUNCT“ und Nummer.
Ein weiteres Schlüsselwort lautet „INCLU␣/␣“ und gibt an, dass an dieser Stelle eine weitere Datei eingefügt werden soll. Diese kann ebenfalls Materialkarten enthalten und wird ebenfalls durchsucht.

/F 20/ PAM-Crash Input-Decks für Zugversuche vorbereiten
Für die Durchführung der simulierten Zugversuche werden vordefinierte PAM-Crash Input-Decks verwendet. Diese Decks werden für alle Materialien genutzt, die den Versuchen unterzogen werden sollen. Dabei müssen drei Punkte des Decks für das jeweilige Material angepasst werden. Im Listing 6.2 sind die drei Punkte als Aus-

schnitt dargestellt. „FILE“ gibt den Namen der Ergebnisdatei des Versuchs, sowie den Speicherpfad an. Unter „PART␣␣/␣“, ist die Materialnummer einzutragen und unter „INCLU␣/␣“ der Pfad, sowie die Datei, in der sich die Materialkarte und deren dazugehörige Funktion befindet.

Listing 6.2: Auszug eines eines PAM-Crash-Decks

```
...
FILE ./run/SHE_unbekannt
...
PART  /                 8133400
...
INCLU / ./mat/SHE_unbekannt/SHE_unbekannt.mat
...
```

/F 30/ Aufruf von PAM-Crash

Mit den unter /F 20/ vorbereiteten PAM-Crash-Decks werden die simulierten Zugversuche durchgeführt. PAM-Crash wird auf Konsolenebene in einem neuen Prozess gestartet und die Ausgabe in eine Logdatei umgeleitet. Der Name der Logdatei besteht aus dem Namen des durchgeführten Versuchs.

/F 40/ PAM-View Session-Kommandodatei bearbeiten

PAM-View nutzt eine Session-Kommandodatei für einen automatisieren Programmablauf. Diese Kommandodatei verhält sich ähnlich wie ein Makro und wird beim Programmstart mit übergeben.

So lassen sich aus den Time History Protokoll Dateien (*.THP), die als Ergebnis der simulierten Zugversuche geliefert werden, die Spannungs-Dehnungs-Kurven erzeugen. Sie werden die ASCII-Datei gespeichert.

Listing 6.3: Auszug einer PAM-View Session-Kommandodatei

```
...
OPENDSY: ./run/SHE_unbekannt.THP
...
CRVGENTITLE:  Material X
...
SAVECVFILE: ./pv/SHE_unbekannt.pv
...
```

Das Listing 6.3 zeigt einen Auszug aus einer PAM-View Session-Kommandodatei. Zu sehen sind die drei Punkte, die vor jedem Aufruf von PAM-View angepasst werden müssen. „OPENDSY:␣“ beschreibt die zu öffnende THP-Datei inkl. ihres relativen Pfades. Im Punkt „CRVGENTITLE:␣“ wird der Kurventitel angegeben. Er entspricht dem Namen des Material bzw. seiner Materialnummer. Unter

„SAVECVFILE:␣“ wird der Pfad, sowie der Name der zu speichernden Kurve angegeben. Der Name der Datei entspricht dem Namen der THP-Datei, mit dem Unterschied das sich der Dateityp in *.pv ändert.
Für die Erzeugung der Bilder im Format PNG[1] einer Kurve eine ähnliche Session-Kommandodatei verwendet. Allerdings mit dem Unterschied, dass der Punkt „PS-FILE:␣“ anstelle von „SAVECVFILE:␣“ vorhanden ist. Hier wird der gleiche Pfad und Name angegeben, nur ist der Dateityp diesmal *.png.

/F 50/ Aufruf von PAM-View
Mit den unter /F 40/ vorbereiteten Session-Kommandodateien wird PAM-View in einem neuen Prozess gestartet. Dabei werden aus den THP-Dateien automatisch die Spannungs-Dehnungs-Kurven erzeugt und als ASCII-Datei abgelegt.

/F 60/ Materialdatenbank
In der Materialdatenbank werden alle bekannten Materialien abgelegt. Dazu gehören die THP-Datein und die Spannungs-Dehnungs-Kurven (als ASCII-Datei und PNG-Bild). Die Datenbank selbst stellt ein Verzeichnis auf dem Server da. In ihm sind alle Materialien in Unterverzeichnissen abgelegt. Die Unterverzeichnisse tragen jeweils den Namen des Materials. Außerdem werden die Kurven als *.png Bilder gespeichert.

/F 70/ Erstellen der Materialdatenbank
Unter Angabe des Kommandos „-m“ gefolgt von einem Verzeichnisnamen, sollen alle in dem Verzeichnis enthaltenen Materialien der Datenbank hinzugefügt werden. Diese werden, wie unter /F 10/ beschrieben, eingelesen. Es werden die simulierten Zugversuche durchgeführt, die Spannungs-Dehungs-Kurven erzeugt und das Material in die Datenbank eingefügt.

/F 80/ Liste der vorhandenen Materialien in der Materialdatenbank
Alle vorhandenen Materialien werden als Liste in der Materialdatenbank in Form einer ASCII-Datei gespeichert. Die Liste enthält neben dem Materialnamen auch die Materialnummer.

/F 90/ Konfigurationsdatei des Programms
Damit einige Einstellungen nicht bei jedem Programmstart neu vorgenommen werden müssen, wird die Konfigurationsdatei „idemat.cfg“ verwendet. Dabei handelt es sich um eine ASCII-Datei. Im Folgenden sind die möglichen Einstellungen, sowie ihre Syntax aufgeführt:

[1] Portable Network Graphics

DB:	Pfad zur Materialdatenbank.
PAMViewExecute:	Prgrammname, mit dem PAM-View gestartet wird.
PAMCrashExecute:	Prgrammname, mit dem PAM-Crash gestartet wird.
PC:	Pfad zu den PAM-Crash Zugversuchs-Decks.
PC1: bis PC4:	Angabe der zu verwendenden PAM-Crash Input-Decks für die simulierten Zugversuche (1 bis 4).
PCG1: bis PCG4:	Gewichtung der Zugversuchs-Decks in Prozent.
PV:	Pfad zu den PAM-View Session-Kommandodateien.
PVC:	Dateiname der PAM-View Session-Kommandodatei zum Erzeugen der Spannungs-Dehnungs-Kurven.
PVPNG:	Dateiname der PAM-View Session-Kommandodatei zum Erzeugen der PNG-Bilder.
PVPNGDBBEST:	Dateiname der PAM-View Session-Kommandodatei zum Erzeugen der PNG-Bilder für die Darstellung der Abweichung zwischen bekannten und unbekannten Material.
B1: bis B3:	Aufteilung der Spannungs-Dehnungs-Kurve in drei Gewichtungsbereiche in Prozent.
BG1: bis BG3:	Gewichtung der drei Kurvenbereiche in Prozent.
AB:	Die max. Abweichung die eine Kurve aus der Datenbank zu der unbekannten Kurve haben darf um ersetzt zu werden.
THPDSY:	Auswahl: Löschen der THP und DSY Dateien.

/F 100/ Klassifizieren der Materialien

Die Klassifizierungsfunktion wird mit Kommando „-k“ gefolgt von einem PAM-Crash Deck gestartet. Die darin enthaltenen unbekannte Materialien werden mit den bekannten Materialien in der Datenbank verglichen. Der Vergleich wird entsprechend des unter 5.2.4 beschrieben Algorithmus durchgeführt.

/F 110/ Ersetzen von Materialkarten

Die unbekannten Materialkarten sollen durch bekannte Materialkarten ersetzt werden können, die die beste Übereinstimmung geliefert haben. Für den Bereich, der als beste Übereinstimmung gelten soll, ist die max. zulässige Abweichung aus der Konfigurationsdatei zu nehmen. Die Materialkarten und die Funktionen werden automatisch ersetzt. Die ersetzten Materialkarten und Funktionen werden auskommentiert.

/F 120/ Templateeintrag erzeugen

In einer ASCII-Datei, mit der Endung *.template wird eine Liste angelegt. Die Liste soll alle Parts enthalten, die in dem PAM-Crash Input-Deck gefunden worden

sind und deren Typ Shell ist. Anhand der Liste kann jedem Part eine unterschiedliche Blechdicke und Material zugewiesen werden. Wurde eines der unbekannten Materialien im PAM-Crash Input-Deck durch ein bekanntes ersetzt, so ist das bekannte Material dem entsprechenden Part zuzuordnen. Das unbekannte Material wird dann nicht mehr in die Liste eingetragen. Listing 6.4 und 6.5 zeigen eine Beispiel-Template-Datei. (Auf Grund der Größe wurde das Listing in zwei Teile unterteilt.)

Listing 6.4: Beispiel einer Templatedatei - Teil 1

```
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7- ...
$ Bauteilbezeichnung       Zeichnungsnummer TYP      PID     Blechdicke ...
$___________28_____________| |_____14_____| |_| |__8___| |_____14_____| ...
Anbindung Kuehlermodul                      SHE  2209300 r        5.000 ...
Manschette-Spurstange                       SHE  2812810 r        2.500 ...
Part 22334, Mater 2495440                   SHE  2495440 r        2.500 ...
```

Listing 6.5: Beispiel einer Templatedatei - Teil 2

```
... ---+----8----+----9----+----0----+----1----+----2----+----3----+----4----+|
...   Kontakt                       Material I         Gruppe Vis         Farbe
...  |__8___|  |___________28_____________| | |_____14_____| |_| |_____14_____|
...                                kunststoff 5 kuehler        she       darkcyan
...                            sr_st14_fep04 5 vorderachse
...                            nicht ersetzt 5 Abgas
```

Unter der Spalte Bauteilbezeichnug ist der Name des Bauteils einzutragen. Der Bauteilname ist in der Partkarte hinter „NAME␣“ zu finden. Ist kein Name in der Partkarte angeben, wird als Bauteilbezeichnung die Partnummer und die Materialnummer des Materials, aus der der Part besteht, eingetragen. In der Spalte „Typ“ wird ausgewiesen, um welchen Typ es sich bei dem Part handelt. Hier sind nur Parts vom Typ „SHE“ (Shell) einzutragen. Die Partnummer wird unter PID eingetragen. Der entscheidende Punkt der Liste ist die Blechdicke. Sie ist ebenfalls der Partkarte zu entnehmen und einzutragen. Die Spalte Material gibt den Namen des Materials an, aus dem der Part besteht. Wurde das unbekannte Material nicht ersetzt, so wird „Nicht ersetzt“ anstelle des Materialnamens eingetragen. Die Spalte „I“ (Integrationspunkte) erhält immer den Wert 5. In der Spalte „Gruppe“ wird ebenfalls der Materialname oder „Nicht ersetzt“ eingetragen. In der Spalte „vis“ wird immer der Eintrag „she“ vorgenommen. Alle nicht genannten Spalten bleiben leer.

/F 130/ Ausgabe der Klassifizierungsrgebnisse

Nach der Klassifizierung der Materialien sollen die Ergebnisse repräsentativ dargestellt werden. Dazu wird eine HTML-Datei erzeugt, die folgendes beinhalten soll (s. Bild 6.1):

- Datum, Uhrzeit des Vergleichs
- Anzahl der gefundenen Materialien
- Max. zulässige Abweichung
- Tabelle aller Vergleichsergebnisse dieses Durchlaufs:
  - Unbekanntes Material
  - Bekanntes Material mit bester Übereinstimmung unter der zulässigen Abweichung
  - Abweichung in Promille
  - Bekanntes Material mit bester Übereinstimmung über der zulässigen Abweichung
  - Graphische Darstellung der Kurven über Hyperlinks

Klassifizierungsergebnis

Stand: 2005-8-23 15:34:43
Unbekannte Materialien: 4
Max. zulaessige Abweichung: 2 Promille

Unbekanntes Material	Materials aus DB	Abweichung in Promille	Bestes Material über der max. zulaessigen Abweichung
Guertel PID: 300	SHE_stahl_elas	0.09481	
H220B = ZSTE220, mit Kaltverfestigung	SHE_bl_zste220	0.00000	
Luftansaugung	Kein passendes Material gefunden.	13.11184	SHE_RT_PP4_Stamylan_PHC
CP800, mit Kaltverfestigung	SHE_sr_CP800_0	0.72253	

Bild 6.1: Darstellung eines Klassifizierungergebnisses

/F 140/ Logfile schreiben

Bei jedem Aufruf des Programms wird ein Logfile geschrieben. Bei dem Logfile handelt es sich um eine ASCII-Datei. Der Name entspricht dem aktuellen Datum und der Zeit, zu der das Programm gestartet wurde. Das Logfile hat folgenden Aufbau:

<Datum> <Uhrzeit> <Meldung>

Die zu schreibenden Meldungen sind:

- Programmstart/-ende
- Fehlermeldungen
- ...

6.2.5 Produktdaten

Während des Programmablaufs werden folgende Daten temporär oder dauerhaft gespeichert:

/D 10/ Zugversuchergebnisse - THP-Dateien (Wahlweise temporär oder dauerhaft)
→ 1-n Time-History-Dateien

/D 20/ Spannungs-Dehnungs-Kurven der Zugversuchsergebnisse
→ 1-n ASCII-Dateien

/D 30/ Materialliste
→ 1 ASCII-Datei

/D 40/ Konfigurationsdatei des Programms
→ 1 ASCII-Datei

/D 50/ Klassifizierungsergebnisse
→ 1 HTML-Dokument

/D 60/ Templatedatei
→ 1 ASCII-Datei

/D 70/ Logfile
→ 1 ASCII-Datei

6.2.6 Produktleistungen

/L 10/ Das Programm wird über die Konsole gestartet. Dabei werden den Programmen Kommandos übergeben. Fehlerhafte Eingaben müssen abgefangen werden. Die Ausgabe eines Hilfetextes soll dem Benutzer die Möglichkeit geben, seine Eingabe zu korrigieren.

/L 20/ Die Darstellung der Klassifizierungsergebnisse erfolgt im HTML-Format. Da das HTML-Format standardisiert ist, ist es vom Browser und System unabhängig. So ist es möglich die Ergebnisse, unabhängig davon, welcher Browser auf dem System installiert ist, korrekt darzustellen.

/L 30/ Durch die Verwendung von PAM-Crash Input-Decks für die Zugversuche, ist es möglich, die Parameter der Versuche zu ändern, oder die kompletten Versuche durch andere zu ersetzen.

/L 40/ Der Einsatz von PAM-View Kommandodateien gestattet es, die zu erzeugenden Kurvenbilder zu formatieren (Beschriftung, Farbe, ...).

/L 50/ Es besteht die Möglichkeit, dass die Dateien für die Konfiguration falsch editiert worden sind. Da eine fehlerhafte Datei den Programmablauf entscheidend stören kann, ist diese zu prüfen und das Programm mit einer Fehlermeldung zu beenden.

6.2.7 Qualitätsanforderungen

Tabelle 6.2: Qualitätsanforderungen Pflichtenheft

Produktqualität	+	o	-
Funktionalität	X		
Zuverlässigkeit	X		
Benutzbarkeit		X	
Effizienz	X		
Änderbarkeit		X	
Übertragbarkeit			X

6.2.8 Benutzeroberfläche

Die Programmausführung des Programms erfolgt in einer UNIX Shell. Die UNIX Shell ist ein Kommandozeileninterpreter und stellt eine Schnittstelle zwischen Benutzer und System dar. Eine grafische Benutzeroberfläche ist nicht notwendig, da nur wenig Interaktion mit dem Benutzer erfolgt. Lediglich die Darstellung der Ergebnisse erfolgt mittels eines HTML Browsers.

6.2.9 Entwicklungs- und Arbeitsumgebung

Als Entwicklungs- und Arbeitsumgebung steht folgendes System zur Verfügung:

LINUX I686 PC

CPU: 2x AMD Athlon 2000+ 1,67GHz
Main memory size: 3GB
Graphics board: GeForce4 Ti4600
Operating System: Red Hat Enterprise Linux WS Release 3 (Kernel 2.4.21-20.ELsmp)

C++-Compiler: GCC 3.2.3
Quellcodebearbeitung: KDevelop 2.1.5
Debugger: gdb 6.1
Leistungsanalyse: gprof 2.14.90.0.4

7 Entwicklungsphase

Nachdem das Lasten- und Pflichtenheft definiert wurden und alle theoretischen Lösungsansätze ausgearbeitet sind, erfolgt die praktische Umsetzung des Programms.
Zuerst soll kurz beschrieben werden, welche Art von Implementierungsform für das Projekt gewählt worden ist. Danach folgt eine Veranschaulichung der Struktur des Programms Idemat. Abschließend wird auf die durchgeführten Tests und deren Ergebnisse eingegangen.

7.1 Implementierung

Für dieses Softwareprojekt wurde die Programmiersprache C++ gewählt, da sie für den Umfang des Programms Idemat eine elegantere und anschaulichere Programmstruktur ermöglicht, als es zum Beispiel C tun würde. Als Entwicklungswerkzeug wurde KDevelop verwendet, welches zum Kompilieren wiederum den gcc-Compiler nutzt. Zusätzlich zum KDevelop eigenen Debugger, wurde der Debugger gdb benutzt. Für die Leistungsanalyse von Idemat wurde das Werkzeug gprof verwendet.
Während der Entwickslungsphase wurde der Quellcode immer mit Debug- und Profilinginformationen kompiliert, um bei unerwarteten Programmabbrüchen Informationen über den aufgetretenen Fehler zu erhalten. Die Profilinginformationen halfen dabei, die Ablaufzeit des Programms zu optimieren. Dies ist besonders für den Kurvenvergleich wichtig, da dieser sehr viel Rechenzeit benötigt. Um die Rechenzeit weiter zu verkürzen, wurde außerdem die Quellcodeoptimierung der Compilers mit der Stufe 3 aktiviert.
Als Betriebssystemumgebung wird Linux verwendet, da für die Programmausführung notwendigen externen Programme z. Z. nur unter Linux bzw. Unix verfügbar sind und die Anwender auch nur unter diesem System arbeiten.
Der Quellcode wurde dem ANSI-Standard entsprechend geschrieben. Weiterhin wurden Elemente der STL[1] genutzt.

[1] Standard Template Library

7.2 Programmstruktur

Die Festlegung einer Programmstruktur durch ihre Klassen und Unterklassen (auch Child-Klassen genannt) erfordert eine genaue Betrachtung des gesamten Projektes aus verschiedenen Blickwinkeln. Eine Klassenstruktur sollte nach Kriterien, wie Wiederverwendung, Übersichtlichkeit und Kernfunktionen des Produktes gewählt werden.
Die Untersuchung des Projekts zeigte vier wesentliche Klassentypen für eine Struktur auf:

1. Dateioperationen:
   Während des Programmablaufes wird auf eine Vielzahl verschiedener Dateien zugegriffen. Dabei werden die Dateien immer vom Typ ASCII sein und dem UNIX-Format entsprechen, aber ganz unterschiedliche Inhalte aufweisen, welches wiederum unterschiedliche Lese- und Schreibalgorithmen erfordert.

2. Materialkurven:
   Für den Vergleich der einzelnen Kurven ist es erforderlich, alle Kurveninformationen zur Laufzeit zu speichern und gegebenenfalls Methoden für ihre Bearbeitung zur Verfügung zu stellen.

3. Klassifizierungsalgorithmen:
   Zur Durchführung der eigentlichen Klassifizierung wird eine Klasse benötigt, die alle für einen Kurvenvergleich wichtigen mathematischen Funktionen kapselt.

4. Programmsteuerung:
   Um die Zugversuche mit den einzelnen Materialien durchzuführen und um die Spannungs-Dehnungs-Kurven aus den Ergebnisdateien zu erzeugen, werden die beiden Programme PAM-Crash und PAM-View verwendet. Die beiden Programme sollen zur Laufzeit von Idemat gestartet werden. Für das Starten der Programme mit den nötigen Parametern werden zwei Klassen verwendet; eine für jedes Programm.

Zu den vier oben genannten Klassenarten wird es noch eine weitere Klasse geben. Sie wird die gesamte Programmflußsteuerung übernehmen und ermöglicht es, sich komplett von einer funktionsorientierten Programmierung zu lösen und nur noch objektorientierte Strukturen zu nutzen.
Auf ein Klassendiagramm muss hier verzichtet werden, da eine verständliche Darstellung auf Grund der Größe nicht möglich ist. Das komplette Diagramm ist aber in anschaulicher Form auf der CD-ROM (Anhang B) gespeichert.

7.2.1 Beschreibung der Klassen

Im Folgenden sollen die verwendeten Klassen vorgestellt werden. Hierbei wird nicht auf den geschrieben Quellcode eingegangen, sondern lediglich eine kurze Einführung in die Aufgaben der Klassen gegeben.

cClassifi::cStraight_Line

In dieser Klasse wird hauptsächlich eine Methode bereitgestellt, die den Vergleich zweier Kurven mit den unter Kapitel 5.2.4 vorgestellten Algorithmus ermöglicht. Da nicht der gesamte Vergleichsalgorithmus, aus Gründen der Übersicht, in einer Methode implementiert werden sollte, wurden weitere Protected-Methoden geschaffen, die einzelne kleinere Funktionen übernehmen (z.B. das Bestimmen eines Abstandes von einer Geraden zu einem Punkt). Zusätzlich existiert eine Methode, die es ermöglicht, den E-Modul, sowie die Fließgrenze eines Materials zu bestimmen. Diese wird benötigt, um bei der Klassifizierung eine Vorauswahl der Materialien zu treffen, die für einen Vergleich am ehesten in Frage kommen.
Innerhalb von cClassifi befindet sich außerdem die Klasse cStraight_Line. In ihr wird immer die aktuell zu betrachtende Gerade gespeichert.

cConfig

cConifg dient dazu, die Konfigurationsdatei „idemat.cfg“ des Programms auszulesen. Dabei wird auch überprüft, ob die Eingaben in der Konfigurationsdatei korrekt vorgenommen wurden. Während des Programmablaufs stellt cConfig den anderen Klassen evtl. benötigte Werte aus idemat.cfg zur Verfügung.

cDir

Die Klasse cDir handelt alle Funktionen ab, die Verzeichnisse betreffen. Dazu gehört u.a. das Auslesen oder Erzeugen eines Vezeichnisses. Außerdem stellt cDir eine Methode zur Verfügung, die überprüft, ob ein Verzeichnis vorhanden ist. Diese wird z.B. von cConfig genutzt.

cFile

Die einzige Parent-Klasse in dem Programm ist cFile. Von ihr werden alle Child-Klassen abgeleitet, die in irgendeiner Weise lesend und/oder schreibend auf Dateien zugreifen. Die

Hauptmethoden von cFile sind das Öffnen, Schließen oder Kopieren einer Datei. Weiterhin ist auch, wie unter cDir, eine Prüf-Methode vorhanden. Im Unterschied zu cDir überprüft sie, ob eine Datei vorhanden ist. Auch diese wird z.B. wieder von cConfig genutzt.
Im Laufe der Entwicklung wurden in cFile zusätzliche Methoden abgelegt, die nicht direkt etwas mit Dateien zu tun haben. Trotzdem wurden sie in cFile untergebracht, da sie dort von den meisten Klassen ohne Umwege aufgerufen werden können und im häufigsten Fall im Zusammenhang mit Dateinamen verwendet werden können. Bei diesen Methoden handelt es sich um String-Operationen, um z.B. Freizeichen, vor oder hinter einem Dateinamen oder Dateiendungen zu entfernen.

cGraph

Die Klasse cGraph beinhaltet eine Public-Methode. Sie dient dazu, die Kurven in den von PAM-View erzeugten ASCII-Dateien einzulesen. Über verschiedene Variablen kann dabei ausgewählt werden, ob eine Kurve nur gelesen oder auch gleichzeitig gewichtet (s. Kapitel 5.2.5) werden soll. Die Gewichtungsparameter werden cConfig entnommen. Wird eine Kurve gewichtet, wird sie auch wieder gewichtet auf der Festplatte gespeichert, so dass beim nächsten Einlesen keine neue Gewichtung vorgenommen werden muss.

cHTML

Die Ergebnisdatei einer Klassifizierung wird mit der Klasse cHTML geschrieben. Zusätzlich existiert in cHTML auch eine Methode zum Darstellen einer Zusammenfassung für das Erzeugen einer Materialdatenbank.

cidemat

Die Steuerung des Programmablaufs wird in cidemat vorgenommen. Von hier aus werden fast alle anderen Klassen angesprochen und ihre Methoden aufgerufen. cidemat besitzt somit auch keine Methoden, die von einer anderen Methode aufgerufen werden kann. Daraus resultiert auch der Zustand, dass fast keine Algorithmen in cidemat vorhanden sind.

cLogfile

Idemat erzeugt bei jedem Aufruf eine Logdatei, in der Meldungen abgelegt werden. Die Klasse cLogfile stellt dafür verschiedene Methoden bereit, die es ermöglichen, Meldungen in die

Logdatei zu schreiben. Alle Meldungen, die nicht variabel sind, wurden in der Headerdatei const.h als Konstante abgelegt.

cMatfile

Materialdateien oder PAM-Crash-Decks werden mit den Methoden in der Klasse cMatfile eingelesen. Zusätzlich existieren Methoden, die es ermöglichen, die unbekannten Materialien nach einer Klassifizierung auszukommentieren und die äquivalenten bekannten Materialien in eine neue Datei zu kopieren.

cMatlist::cMat

cMatlist ist eine Klasse, in der eine Liste mit allen gefundenen Materialien angelegt wird. Wird eine Materialdatenbank erzeugt, kann die Materialliste mit einer Methode als Datei abgelegt werden, die für einer spätere Klassifizierung wieder eingelesen werden kann. Wie auch cClassifi hat cMatlist eine interne Protected-Klasse. Der Name ist cMat und sie stellt die eigentliche Materialliste dar, die als Form eines Vektors von cMat erstellt wird.

cPC

Die Aufgaben von der Klasse cPC sind sehr gering. Sie dient lediglich dazu, die PAM-Crash-Decks für die Zugversuche zu editieren und wird ausschließlich von cPamC verwendet.

cPVcmdf

cPVcmdf ist ähnlich der Klasse cPC. Ihr obliegt es, die Session-Kommandodateien für PAM-View zu editieren. Auch sie wird nur von den Methoden einer Klasse genutzt. Hier ist es cPamV.

cPamC

Der externe Aufruf des Programm PAM-Crash wird mit der Klasse cPamC durchgeführt. Bevor PAM-Crash allerdings gestartet werden kann, müssen mit den Methoden von cPC die Decks für die Zugversuche dem Material entsprechend geändert werden.

cPamV

Die Klasse cPamV erfüllt in etwa die gleichen Aufgaben wie cPamC, nur das anstatt PAM-Crash das Programm PAM-View gestartet wird. Im Vorfeld werden die PAM-View Session-Kommandodateien mit den Methoden der Klasse cPVcmdf geändert.

cPart

In der Klasse cPart werden die wichtigen Punkte einer eingelesenen Partkarte gespeichert. Sie dient nur dem Speichern dieser Daten und hat keine nennenswerten Methoden.

cTemplate

cTemplate hat nur die Aufgabe, die Templatedatei zu erzeugen. Dabei werden die in cPart gespeicherten Daten einem ersetzten Material zugeordnet und in die Datei geschrieben.

cTime

Die Berechnung der aktuellen Zeit übernimmt die Klasse cTime. Sie wird nur von cLofile genutzt, um alle geschriebenen Meldungen mit einem Zeitstempel zu versehen. Über eine Methode können die Zeitwerte in den Variablen von cTime aktualisiert werden.

7.2.2 Programmfunktionen

Idemat stellt dem Benutzer zwei Hauptfunktionen zur Verfügung. Zum einen die Erzeugung der Materialdatenbank, zum anderen das Klassifizieren von Materialien in PAM-Crash Input-Decks. Eine weitere dritte Funktion gibt einen Hilfetext auf dem Bildschirm aus.
Die folgenden zwei PAPs (s. Bild 7.1 und 7.2) sollen den Ablauf beider Programmfunktionen anschaulich darstellen. Auf die Darstellung von Unterprogrammen (Methoden) wurde verzichtet, da dies den Rahmen dieser Arbeit überschreiten würde. Um zu verdeutlichen, welche Klasse an dem jeweiligen Programmschritt beteiligt ist, wird diese immer in Klammern mit angegeben.

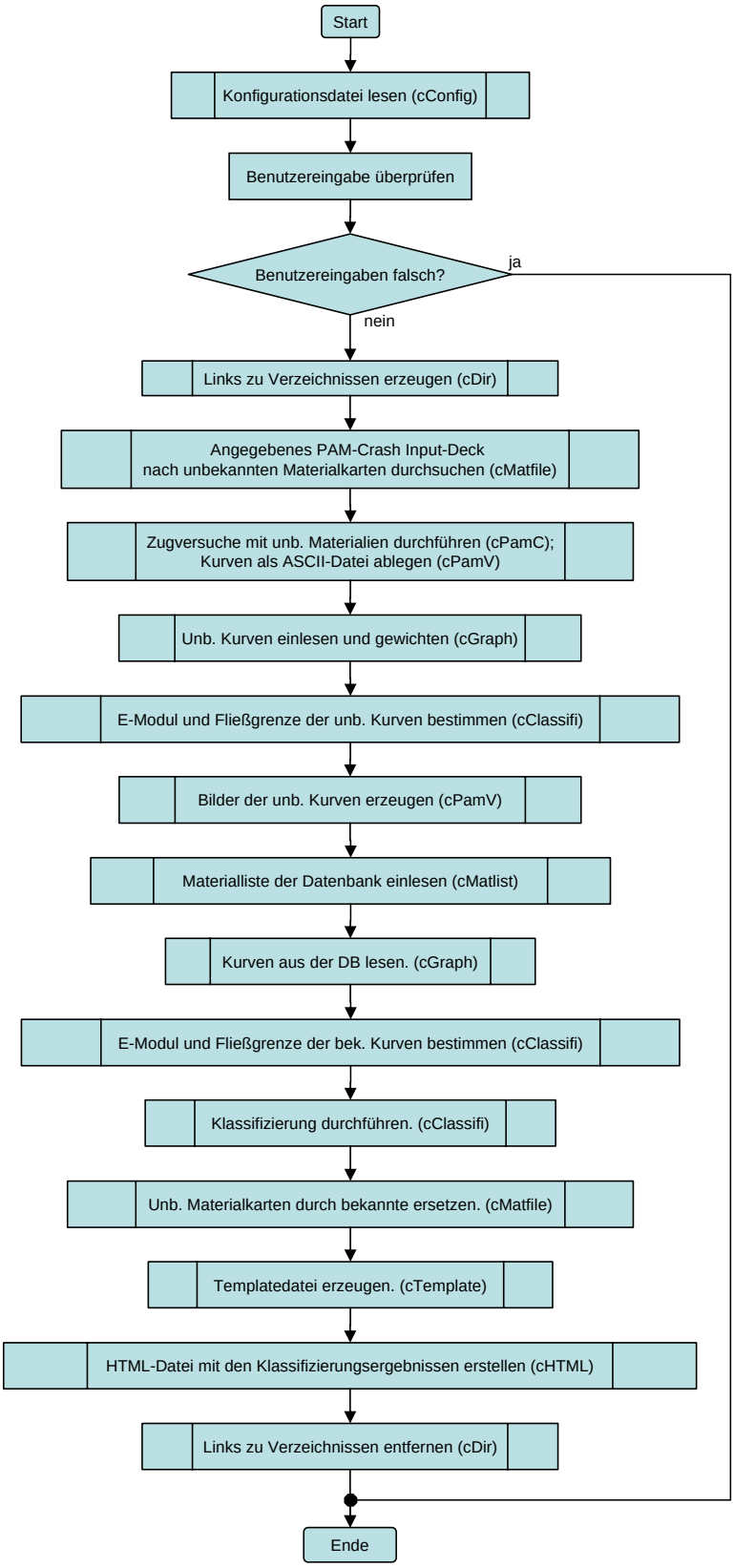

Bild 7.1: PAP - Materialklassifizierung

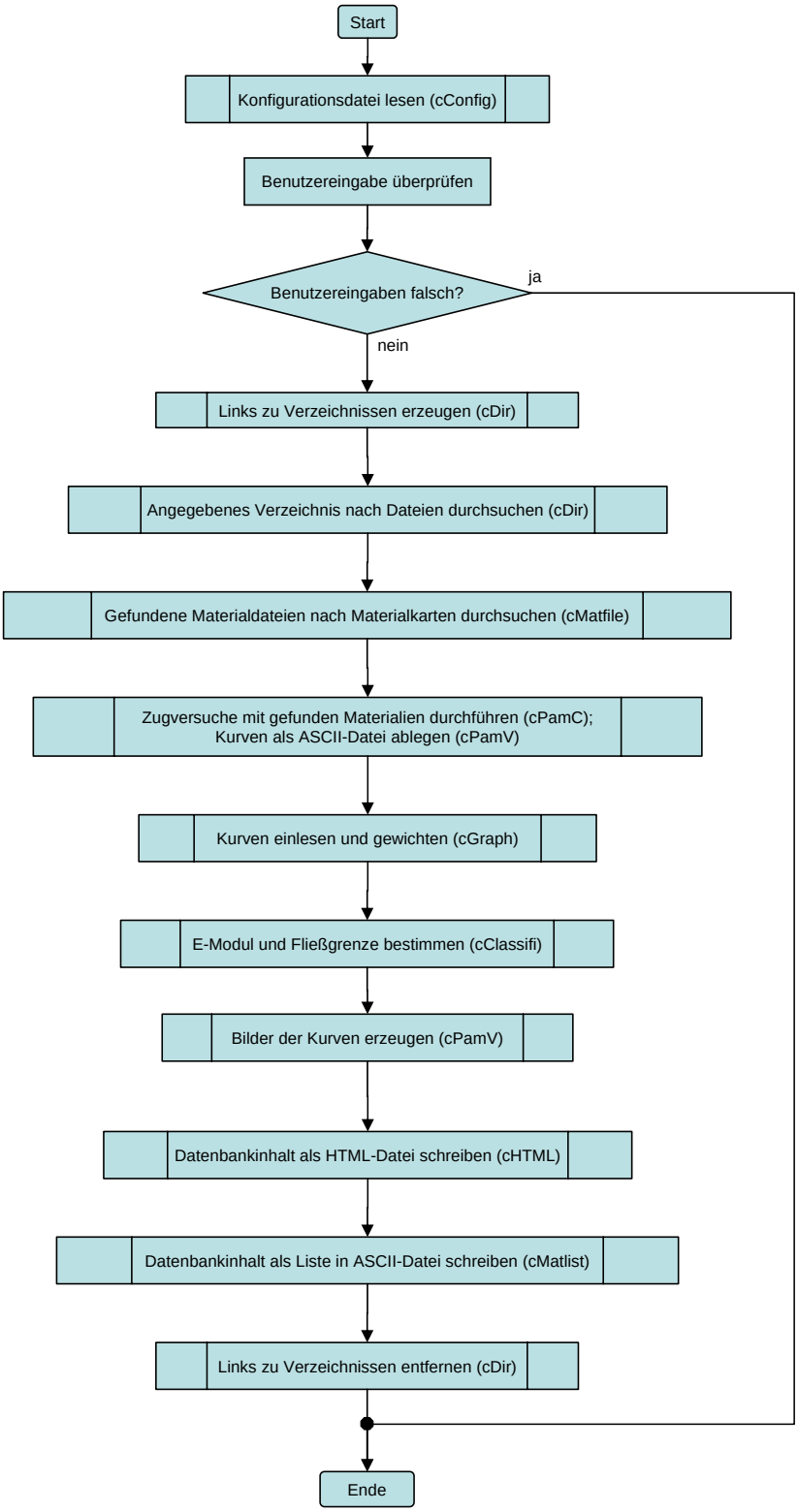

Bild 7.2: PAP - Materialdatenbank

7.3 Umsetzung des Kurvenvergleichs

Der Schwerpunkt von Idemat liegt auf dem Vergleich von zwei Kurven. Das Verfahren dafür wurde bereits in Kapitel 5 beschrieben. Für die Umsetzung des Verfahrens in Idemat wurde die Klasse cClassifi geschrieben. Der Kurvenvergleich selber wurde nach dem PAP in Bild 5.11 implimentiert.
Da die Durchführung der Zugversuche, sowie der Kurvenvergleich, je nach verwendeter System-Hardware sehr viel Zeit in Anspruch nehmen kann, wird der Kurvenvergleich mit einer Vorauswahl und einer Dynamik versehen. Dabei wird im Vorfeld anhand des Vergleichs der E-Module entschieden, welches Material in der Materialdatenbank überhaupt für einen Vergleich in Frage kommt. Weicht der E-Modul mehr als 10% von dem der unbekannten Kurve ab, wird sie aussortiert. So kann sich die Anzahl der zu vergleichenden Kurven evtl. auf eine kleine Menge reduzieren.
Weiterhin kann der eigentliche Kurvenvergleich vorzeitig abgebrochen werden, wenn während des Vergleichs festgestellt wird, dass die geforderte max. Abweichung AB bereits überschritten wurde. Für einen Abbruch muss folgende Bedingung erfüllt werden:

$$a > AB \cdot y_{sum} \tag{7.1}$$

AB wird in der Konfigurationsdatei festgelegt und y_{sum} ergibt sich aus:

$$y_{sum} = \sum_{i=0}^{n} f_u(x_i) \qquad \text{mit } i = 0, 1, 2, ..., n \tag{7.2}$$

a ist immer die Summe aller ermittelten Abstände bis zum aktuell betrachteten Stützpunkt. Wird a also größer als das Produkt aus AB und y_{sum}, wird der weitere Vergleich mit dieser bekannten Kurve abgebrochen.
Da die Zugversuche auch sehr viel Rechenzeit benötigen, wird zu Beginn nur der erste der vier Zugversuche durchgeführt. Finden sich mehrere bekannte Materialien, die einen kleinere Abweichung als AB haben, werden auch die restlichen Zugversuche mit dem unbekannten Material durchgeführt und die Ergebnisse ebenfalls verglichen.
Am Ende wird das Ergebnis von jedem der vier Versuche mit dem Gewichtungsfaktor aus der Konfigurationsdatei multipliziert und die Summe gebildet. Das Material mit der kleinsten Summe entspricht dem unbekannten am ehesten.
Wurde kein Material gefunden, das unterhalb von AB liegt, wird dem unbekannten Material kein bekanntes zugewiesen.
Der PAP in Bild 7.3 zeigt das Klassifizierungsverfahren für ein unbekanntes Material. Das

Abbruchverfahren ist in dem eigentlichen Kurvenvergleich integriert und nicht im PAP dargestellt.

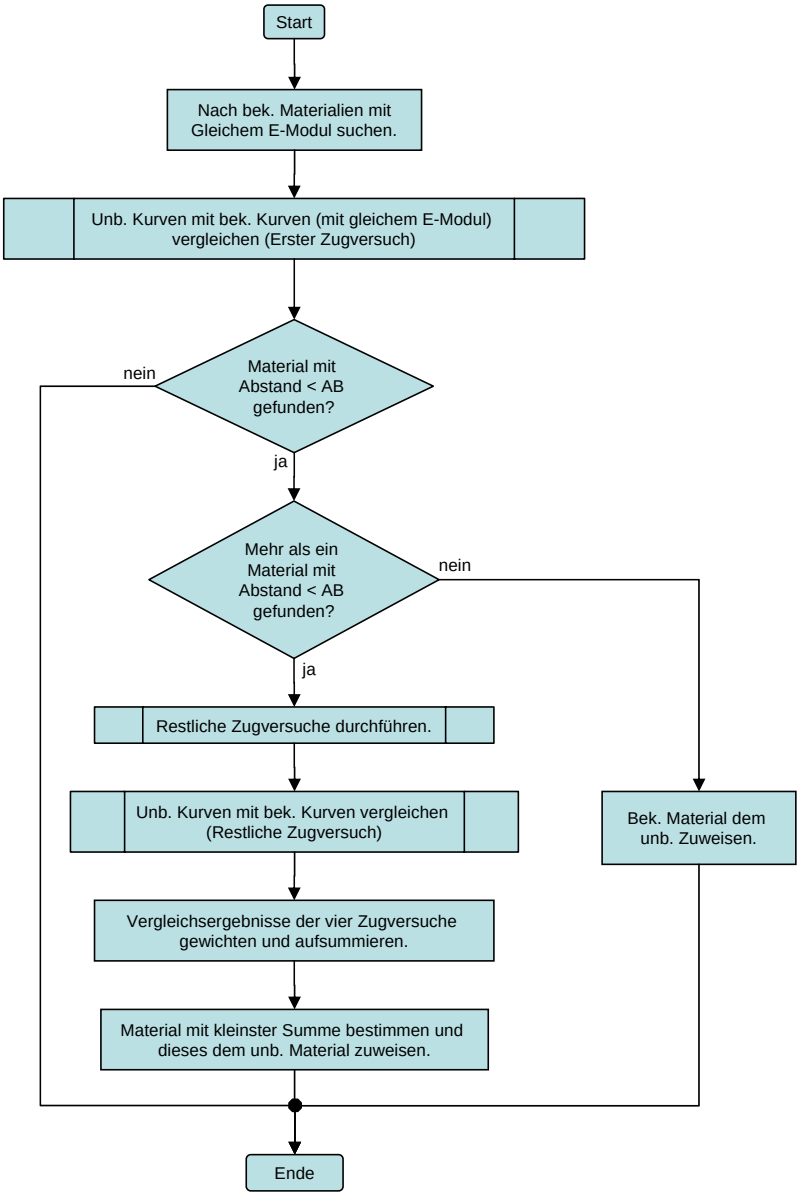

Bild 7.3: PAP - Vergleichsdurchlauf für eine unbekannte Kurve

7.4 Programmtest

Der Test von Idemat wurde in zwei Phasen aufgeteilt. Die erste Phase bestand aus kontinuierlichen Funktionsprüfungen. Dies bedeutet, dass während der gesamten Entwicklungsphase einzelne Teilfunktionen nach ihrer Fertigstellung oder Änderung immer wieder getestet wurden. So wurde sichergestellt, dass die einzelnen Funktionen in sich fehlerfrei arbeiten.
Zum Überprüfen der Gesamtfunktionalität wurden verschiedene PAM-Crash Input-Decks verwendet. Dazu gehörten zu Beginn der Testphase kleine einfache Decks, deren Inhalt aus ein paar Materialien (max. 5) bestanden. Diese Materialien stammten aus der VW-Datenbank, was eine eindeutige Überprüfung der Klassifizierungsergebnisse erlaubte. Zu den endgültigen Test-Decks gehörten verschiedene Simulationsmodelle von unterschiedlichen Fahrzeugen. Dabei erstreckte sich die Bandbreite der Materialien in einem Deck von ca. 20 bis zu über 2000 Materialien. Da die verwendeten Materialien in den Decks bekannt waren, konnten die Ergebnisse der Klassifizierung verifiziert werden. In Bild 7.4 ist das Klassifizierungsergebnis eines herkömmlichen PAM-Crash Input-Decks zu sehen. In der linken Spalte ist das unbekannte Material zu sehen. Der Name wurde der Materialkarte entnommen. Die mittlere Spalte zeigt das Material, welches dem unbekannten aus der Materialdatenbank zugeordnet worden ist. In der dritten Spalte ist die Abweichung der Spannungs-Dehnungs-Kurven beider Materialien zu sehen. Da die Materialien des Decks aus der VW-Datenbank stammen, ist der Abstand der Kurven 0. Wurde eine Abweichung größer 0 ermittelt, ist das Material nicht exakt in der gleichen Form in der Datenbank abgelegt. Wurde der Vermerk „Kein passendes Material gefunden“ vorgenommen, wurde kein entsprechendes Material gefunden, das unterhalb des geforderten max. Abweichung liegt. Ist dies der Fall, wird in der vierten Spalte das Material angegeben, was mit dem unbekannten am besten übereinstimmt.
Während des Programmtests auf verschiedenen Systemen fiel auf, dass die errechnete Abweichung bei identischen Materialien nicht immer exakt 0 war. Nach der Untersuchung des Problems wurde festgestellt, dass die Ergebnisse des Zugversuchs von dem verwendeten Prozessor abhängen. Wird die Klassifizierung mit einem anderen System durchgeführt als die Erstellung der Materialdatenbank, kann es zu kleinen Differenzen in den Materialkurven kommen. Im schlimmsten Fall könnte dies zu einer falschen Klassifizierung führen. In den bisher durchgeführten Test war das Klassifizierungsergebnis aber immer korrekt. Die Rücksprache mit den Entwicklungsingenieuren ergab, dass dieses Verhalten, bekannt und nicht weiter bedenklich ist. In Crashsimulationen wird es sich sogar zunutze gemacht, um die Robustheit einer Fahrzeugstruktur zu ermitteln.
In weiteren Prüfungen wurden u.a. die Fehleranfälligkeit gegenüber dem Benutzer getestet. Dafür wurden unterschiedliche Möglichkeiten der fehlerhaften Eingabe beim Starten des

Klassifizierungsergebnis

Stand: 2005-8-23 15:34:43
Unbekannte Materialien: 28
Max. zulaessige Abweichung: 2 Promille

Unbekanntes Material	Materials aus DB	Abweichung in Promille	Bestes Material über der max. zulaessigen Abweichung
Guertel PID: 300	SHE_stahl_elas	0.09481	
Guertel PID: 300	SHE_stahl_elas	0.09481	
Guertel PID: 300	SHE_stahl_elas	0.09481	
Guertel PID: 300	SHE_stahl_elas	0.09481	
Bleibrucken-Verb.zuPlatten	Kein passendes Material gefunden.	8599.05544	SHE_alu_ELAS
Bleibrucken	Kein passendes Material gefunden.	10328.55377	SHE_alu_ELAS
Bleibrucken-Verb.zuPlatten	Kein passendes Material gefunden.	8599.05544	SHE_alu_ELAS
Bleibrucken	Kein passendes Material gefunden.	10328.55377	SHE_alu_ELAS
Stamylan PHC PP4	Kein passendes Material gefunden.	32.60410	SHE_RT_PP4_Stamylan_PHC
STAHL -- IDEAL ELASTISCH	SHE_stahl_elas	0.00000	
ST1203	SHE_pl_st1203	0.00000	
St1203, mit Kaltverfestigung	SHE_bl_st1203	0.00000	
St1203, mit Kaltverfestigung	SHE_bl_st1203	0.00000	
St1203, mit Kaltverfestigung	SHE_bl_st1203	0.00000	
BTR 165hard, mit Kaltverfestigung	Kein passendes Material gefunden.	1.00557	SHE_sr_CP1000
ZSTE220, mit Kaltverfestigung	SHE_bl_zste220	0.00045	
ZSTE260, mit Kaltverfestigung	SHE_bl_zste260	0.0654	
ZSTE300, mit Kaltverfestigung	SHE_bl_zste300	0.0023	
ZSTE340, mit Kaltverfestigung	SHE_bl_zste340	0.0987	
ZSTE420, mit Kaltverfestigung	SHE_bl_zste420	0.0234	
QSTE500TM	SHE_sr_qste500tm	0.0345	
BENTELER ROHR BTR150	SHE_pl_btr150	0.00046	
TRIP800	SHE_pl_trip800	0.00000	
*Auflageprofil schwellers. ----------	SHE_bl_zste340	0.00007	
*Auflageprofil tunnelseit. ----------	SHE_bl_zste340	0.00007	
SECURE MATER 9000003	Kein passendes Material gefunden.	176763.26674	SHE_alu_ELAS
SECURE MATER 9000022	Kein passendes Material gefunden.	34.84917	SHE_RT_SMA-GF15_Dylark378P15
SECURE MATER 9000035	SHE_stahl_elas	0.09481	

Bild 7.4: Klassifizierungsergebnis eines herkömmlichen PAM-Crash Input-Decks

Programms und den Einträgen in der Konfigurationsdatei überprüft. Einige Falscheingaben können von Idemat selber korrigiert werden. Schwerwiegende Falscheingaben sollten jedoch zu einem gewollten Abbruch des Programms führen.

8 Schlussbemerkung

In dieser Arbeit wurde die Entwicklung des Programms Idemat vorgestellt. Die Aufgabe von Idemat ist es, unbekannte Materialien in PAM-Crash Input-Decks mit bekannten Materialien in einer Datenbank zu vergleichen und bei entsprechender Übereinstimmung zu ersetzen. Dafür war es nötig, ein Vergleichsverfahren zu entwickeln.
Bevor ein Vergleich durchgeführt werden kann, müssen alle unbekannten Materialen aus den PAM-Crash Input-Decks extrahiert werden und einem simulierten Zugversuch unterzogen werden. Dies erfolgt mit dem Programm PAM-Crash. Die Ergebnisse der Zugversuche müssen mit einem weiteren Programm aufbereitet werden. Dafür wurde PAM-View verwendet, welches Spannungs-Dehnungs-Kurven in ASCII-Dateien ablegt. Diese Kurven können miteinander verglichen werden.
Für die Umsetzung von Idemat wurde bewusst die Sprache C++ gewählt. Die objektorientierte Auslegung von C++ ermöglicht eine übersichtliche und strukturierte Programmierung. Außerdem spart die Wiederverwendbarkeit von Programmmodulen sehr viel Zeit, da einzelne Funktionalitäten nicht immer wieder neu geschrieben werden müssen. Weiterhin lässt sich dadurch das Programm einfach um weitere Funktionen erweitern.
Große Aufmerksamkeit wurde der Umsetzung des Kurvenvergleichs gewidmet. Da dieser darüber entscheidet, durch welches bekannte Material das unbekannte ersetzt wird, muss das Verfahren sehr genau und zuverlässig sein. Der Vergleich erfordert eine Vielzahl von mathematischen Operationen, was dazu führt, dass ein Vergleich von zwei Kurven viel Zeit in Anspruch nimmt. Dieser Abschnitt wurde mit einem Leistungsanalysetool stetig analysiert. Dabei wurden Verbesserungen durch Zusammenfassen von Rechenoperationen und Optimierung des Speicherzugriffs vorgenommen.
Ein weiteres Problem stellen die Zugversuche selbst da, die einen großen Teil der Ausführungszeit beanspruchen. Abhilfe würde schnellere lokale Hardware oder die Verlagerung der Zugversuche auf eine Grossrechenanlage schaffen, mit der auch die Crashsimulationen eines PKWs berechnet werden.
Die Verwendung eines HTML-Dokuments zur Wiedergabe der Ergebnisse ermöglicht eine optisch anschauliche Darstellung, ohne selbst auf eine Grafikschnittstelle zurückgreifen zu müssen. Zudem ist ein HTML-Dokument einfach zu erstellen, und kann durch die weite Ver-

breitung von Internet-Browsern auf fast jedem System dargestellt werden.
Bei der Benutzung von Idemat wurde auf eine einfache Handhabung wert gelegt. So kann das Programm mit nur zwei unterschiedlichen Parametern gestartet werden. Alle weiteren Einstellungen werden in einer Konfigurationsdatei vorgenommen. Diese muss aber nur in seltenen Fällen verändert werden.
Für die Zukunft von Idemat wäre die Erweiterung um eine grafische Oberfläche sehr interessant. Dies würde weitere Funktionen ermöglichen, die in einer Konsole nur umständlich zu verwirklichen sind. Eine dieser Funktionen wäre es, dem Benutzer die Wahl zu ermöglichen, welche Materialien er klassifizieren möchte. Des Weiteren würde dies eine einfachere Verwaltung der Materialdatenbank ermöglichen. Einzelne Materialien könnten hinzugefügt bzw. entfernt werden.
Abschließend ist zu erwähnen, dass Idemat die Analyse eines PAM-Crash Input-Decks in Bezug auf die verwendeten Materialien erheblich vereinfacht und dem Entwicklungsingenieur bei seiner Arbeit eine hilfreiche Unterstützung bietet.

Abkürzungsverzeichnis

ANSI	American National Standards Institute
ASCII	American Standard Code for Information-Interchange
CAD	Computer Aided Design
ESI	Engineering Systems International
FEM	Finite Elemente Methode
GUI	Graphical User Interface
HTML	Hypertext Markup Language
PAM	Program Applied Mechanics
PAP	Programm-Ablauf-Plan
PNG	Portable Network Graphics
STL	Standard Template Library
THP	Time History Protokoll

Bildverzeichnis

3.1 Stabdehnung 5
3.2 Spannungs-Dehnungs-Kurve $\sigma(\epsilon)$ 7
3.3 Beispiele von FEM-Elementen 8
3.4 Entstehung des FEM-Netzes am Beispiel des VW Touran 9
3.5 Darstellung eines Fahrzeugs-Crash in PAM-View 10

5.1 Auswirkung der Zuggeschwindigkeit auf das Spannungs-Dehnungs-Diagramm 16
5.2 Kurvenabstand (Methode 1) 18
5.3 Korridor (Methode 2) 19
5.4 Überlagerung der Kurven $f_u(x)$ und $f_b(x)$ (Methode 2) 19
5.5 Abstand Punkt - Kurve (Methode 3) 20
5.6 Abstand Punkt - Punkt (Methode 3) 20
5.7 Abstand Punkt - Gerade (Methode 3) 21
5.8 Zwei-Punkt-Form 22
5.9 Abstand Punkt-Gerade 23
5.10 Kurvenvergleich - Suche in Rechtsrichtung 24
5.11 PAP - Kurvenvergleich 25

6.1 Darstellung eines Klassifizierungergebnisses 39

7.1 PAP - Materialklassifizierung 49
7.2 PAP - Materialdatenbank 50
7.3 PAP - Vergleichsdurchlauf für eine unbekannte Kurve 52
7.4 Klassifizierungsergebnis eines herkömmlichen PAM-Crash Input-Decks 54

A.1 Einteilung einer Kurve in drei Gewichtungsbereiche 66

Tabellenverzeichnis

3.1 Elastizitätsmodul E und Poission-Zahl μ 6

6.1 Qualitätsanforderungen Lastenheft 32
6.2 Qualitätsanforderungen Pflichtenheft 41

Listings

3.1 Beispiel einer PAM-View Session-Kommandodatei 11
3.2 Beispiel einer HTML-Datei 12
6.1 Beispiel einer Materialdatei bzw. eines PAM-Crash-Decks 34
6.2 Auszug eines eines PAM-Crash-Decks 35
6.3 Auszug einer PAM-View Session-Kommandodatei 35
6.4 Beispiel einer Templatedatei - Teil 1 38
6.5 Beispiel einer Templatedatei - Teil 2 38

Literatur

[1] Balzert, Helmut: *Lehrbuch der Software-Technik: Software-Entwicklung.* Heidelberg; Berlin; Oxford: Spektrum Akademischer Verlag, 1996.

[2] ESI-Group: *ESI-Group Homepage.* http://www.esi-group.com. Aktuallisierungsdatum: 2004.

[3] Fowler, Martin; Scott, Kendall: *UML - konzentriert.* 1. Aufl. Bonn: Addison-Wesley-Longman, 1998.

[4] Gerthsen, Christian; Kneser, Hans; Vogel, Helmut: *Physik* 15. Aufl. Berlin; Heidelberg; New York; Tokio: Springer, 1986.

[5] Kuchling, Horst: *Taschenbuch der Physik.* 17. Aufl. München: Carl Hanser Verlag, 2001.

[6] Louis, Dirk: *C/C++ - New Reference.* München: Markt+Technik, 2001.

[7] *PAM-CRASH/SAFE 2000 - PAM-VIEW Reference Manual.* Rungis Cadex - France: PAM System International, 2000.

[8] *PAM-CRASH/SAFE 2002 - Solver Reference Manual.* Rungis Cadex - France: ESI Software, 2002.

[9] Papula, Lothar: *Mathematische Formelsammlung - Für Ingenieure und Naturwissenschaftler.* 8. Aufl. Wiesbaden: Vieweg, 2003.

[10] Sieber, Helmut: *Mathematische Begriffe und Formeln.* Leipzig: Ernst Klett Schulbuchverlag, 2000.

[11] Stroustrup, Bjarne: *Die C++ Programmiersprache.* 4. Aufl. München: Addison-Wesley, 2000.

[12] *The C++ Resources Network.* http://www.cplusplus.com. Aktualisierungsdatum: 2003.

[13] Wenz, Tobias; Hauser, Tobias: *HTML.* München: Markt+Technik: 2001.

[14] Wielsch, Michael: *Linux.* 4. Aufl. Düsseldorf: DATA BECKER, 1999.

[15] Willemer, Arnold: *Willemers Informatik-Ecke.* http://www.willemer.de/informatik/index.htm. Aktualisierungsdatum: 2003.

[16] Wikimedia Foundation Inc.: *Wikipedia - Die freie Enzyklopädie.* http://de.wikipedia.org. Aktualisierungsdatum: 2005-06-15.

A Programmhandbuch

In diesem Handbuch wird die Benutzung, sowie Konfiguration von Idemat beschrieben.

A.1 Beschreibung

Idemat ist ein Programm zum Klassifizieren von Materialien in PAM-Crash Input-Decks. Damit eine Klassifizierung (s. Kapitel A.2.1)durchgeführt werden kann, muss eine Materialdatenbank vorhanden sein. Ist dies nicht der Fall, kann mit Idemat eine Materialdatenbank erstellt werden (s. Kapitel A.2.2).

A.2 Funktionen

Dem Benutzer stehen zwei Programmfunktionen zur Verfügung, die im folgenden beschrieben werden.

A.2.1 Klassifizierung

Die Klassifizierung wird mit Kommando

-k [PAM-Crash Input-Deck]

gestartet.
Beispiel:

idemat -k ./beispieldeck/bspdeck.pc

Bei einer Klassifizierung, wird zunächst das angegebene PAM-Crash Deck, sowie alle über ein Include eingebundenen Dateien, nach Shell-Materialkarten durchsucht. Danach wird mit jedem Material ein Zugversuch durchgeführt und die Ergebnisse mit den Materialien in der Materialdatenbank verglichen. Wurde ein äquivalentes Material in der Datenbank gefunden, so wird die originale Materialkarte inkl. ihrer Funktionen auskommentiert und das Material aus der Datenbank in eine neue Includedatei geschrieben. Die Partkarte, die mit der originalen Materialkarte zusammenhängt, wird entsprechend der neuen Materialnummer geändert. Die neue Includedatei wird automatisch in das PAM-Crash Deck eingebunden.
Desweiteren wird eine *.template Datei erzeugt, in der alle Shell-Partkarten eingetragen werden. Unter der Spalte Material wird entweder ein „Nicht ersetzt“ oder das neu zugeordnete Material eingetragen.
Als Ergebnis der Klassifizierung wird eine *.html Datei geschrieben, in der das unbekannte und neue Material aufgelistet ist. In einer weiteren Spalte wird der relative Abstand zwischen den beiden Materialien angegeben.
Bei der Programmausführung werden die originalen PAM-Crash Input-Deck nur lesend angesprochen. Für alle beschriebenen Veränderungen wird eine Kopie erzeugt, die in einem Ordner mit dem Namen des PAM-Crash Decks plus der Endung „_new“ abgelegt wird.

A.2.2 Materialdatenbank

Das Erzeugen der Materialdatenbank wird dem Kommando

-m [Materialverzeichnis]

gestartet.
Beispiel:

idemat -m ./material/

Die Materialdatenbank wird aus den Materialien erzeugt, die in dem angegebenen Verzeichnis liegen. Dabei muss jede Materialkarte in einer eigenen Datei liegen. Unterverzeichnisse oder Include-Anweisungen werden nicht berücksichtigt. Der Name, der dem Material zugewiesen wird, entspricht dem Namen der Materialdatei. Die Materialdatenbank wird in dem Verzeichnis abgelegt, das in der Konfigurationsdatei angegeben ist.
Nach dem Erzeugen der Datenbank steht dem Benutzer eine HTML Datei zur Verfügung, in der alle Materialien aufgelistet werden. Weiterhin wird eine Liste der Materialien in der Datei matlist.txt abgelegt. Diese Datei wird für die Klassifizierung benötigt.

A.3 Konfigurationsdatei

Zur Steuerung von Idemat wird eine Konfigurationsdatei „idemat.cfg“ verwendet. Diese muss sich entweder in dem selben Verzeichnis wie Idemat befinden oder über die Umgebungsvariable „idemat_configfile“ gesetzt werden.
Folgende Einstellungen sind in der Konfigurationsdatei möglich:

- *DB:* Pfad zur Materialdatenbank. (z.B. ./db/)
- *PAMViewExecute:* Programmname, mit dem PAM-View gestartet wird.
- *PAMCrashExecute:* Programmname, mit dem PAM-Crash gestartet wird.
- *PC:* Pfad zu den PAM-Crash Zugversuchs-Decks. (z.B. ./pc/)
- *PC1:* bis *PC4:* Dateinamen der zu verwendenen PAM-Crash Zugversuchs-Decks. (z.B. qs.pc)
- *PCG1:* bis *PCG4:* Gewichtung der Zugversuchs-Decks in Prozent. Summe muss 100% ergeben. Angabe ohne Prozentzeichen. (z.B. 60)
- *PV:* Pfad zu den PAM-View Session-Kommandodateien. (z.B. ./pc/)
- *PVC:* Dateiname der PAM-View Session-Kommandodatei zum Erzeugen der Spannungs-Dehnungs-Kurven. (z.B. pv.cmdf_curve)
- *PVPNG:* Dateiname der PAM-View Session-Kommandodatei zum Erzeugen der PNG-Bilder. (z.B. pv.cmdf_png)
- *PVPNGDBBEST:* Dateiname der PAM-View Session-Kommandodatei zum Erzeugen der PNG-Bilder für die Darstellung der Abweichung zwischen bekannten und unbekannten Material. (z.B. pv.cmdf_png_db_best)
- *B1:* bis *B3:* Aufteilung der Spannungs-Dehnungs-Kurve in drei Gewichtungsbereiche in Prozent. Angabe ohne Prozentzeichen.

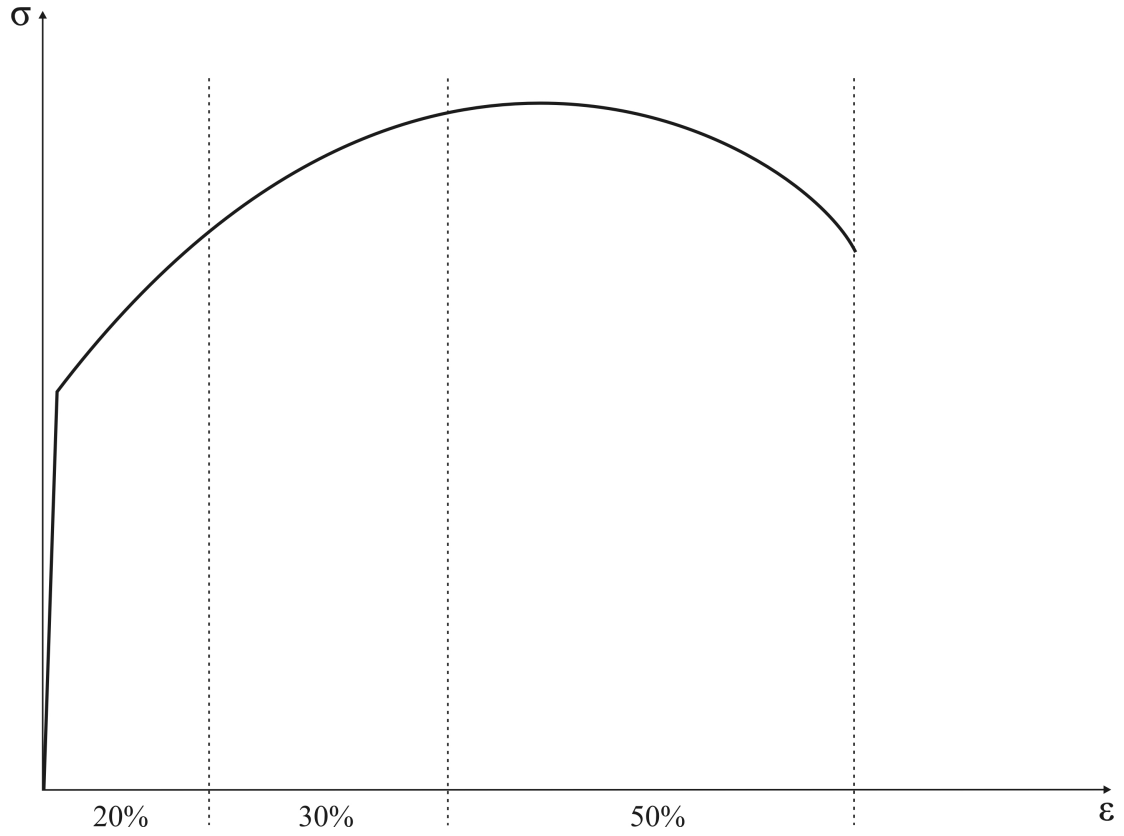

Bild A.1: Einteilung einer Kurve in drei Gewichtungsbereiche

- *BG1:* bis *BG3:* Gewichtung der drei Kurvenbereiche in Prozent. 100% enspricht der Betrachtung jeden Stützpunktes in dem Bereich, 10% entspricht jeden zehnten Stützpunkt. Angabe ohne Prozentzeichen.
- *AB:* Die max. Abweichung die eine Kurve aus der Datenbank zu der unbekannten Kurve haben darf um ersetzt zu werden. (z.B. 0.002)
- *THPDSY:* Löschen der THP und DSY Dateien. 1 = löschen, 0 = nicht löschen.

A.4 Hinweis

Während des Programmablaufs werden mit PAM-View Bilder für die Ergebnisdarstellung erzeugt. Dazu verwendet PAM-View eine Screenshot-Funktion. Damit keine falschen bzw. fehlerhaften Bilder entstehen, sollten während des Programmablaufs der Bildschirmschoner und andere Programme die eine Bilddarstellung beeinflussen deaktiviert werden.

B CD-ROM

Verzeichnis	**Inhalt**
Diplomarbeit	Diplomarbeit als PDF-Dokument
Idemat	
↪ Source	Quellcode von Idemat
↪ Compiled	Idemat als kompilierte Version
↪ Dokumentation	Programmdokumentation
	Klassendiagramm