Inhalt

[1. Einleitung 2](#_Toc9439443)

[2. Erfassung der FEM-basierten Daten 3](#_Toc9439444)

[2.1 Allgemeines von FEM 3](#_Toc9439445)

[2.2 Vorgehensweise einer FEM-Analyse 3](#_Toc9439446)

[2.3 Automatisierung der Vorgehensweise durch SolidWorks-API mittels C# Programmierung 6](#_Toc9439447)

[2.3.1 Einführung der C# Programmierung 6](#_Toc9439448)

[2.3.2 Einführung der SolidWorks-API 7](#_Toc9439449)

[2.3.3 Automatisierte Vorgehensweise 8](#_Toc9439450)

[2.3.4 Ansammlung der Trainingsdaten 9](#_Toc9439451)

[3 Daten erkunden 10](#_Toc9439452)

# 1. Einleitung

Bis Mitte des 20. Jahrhunderts wurde zunächst die theoretische Grundlage der Finite-Elemente-Methode (FEM) gebildet. Dank der Fortschritte von Rechnern ist das Lösen das komplexe theoretische Gleichungssystem möglich. Heutzutage ist FEM in der festigkeitsmäßigen Auslegung von Bauteilen weit verbreitet. Die FEM basiert auf dem Lösen von unterschiedlichen Differenzialgleichungen mittels numerischen iterativen Verfahren. Ein Bauteil wird in endlich viele Elemente, die mit einfacher und nicht überlappender Form gekennzeichnet ist, aufgeteilt. Dazu spielen sowohl die Form der Elemente als auch die Größe der Elemente eine entscheidende Rolle. Deswegen erfordert FEM entsprechend qualifiziertes Fachpersonal. Das Lösen einer Variante von Bauteilen kann manchmal stundenlang dauern. Die benötigte Bearbeitungszeit ist abhängig von Geometrie und Rechenleistung. Vorteil der FEM liegt daran, dass nach einer Berechnung alle Spannungen und dazu entsprechende Verformungen herauskommen. Bei parametrisierbaren Geometriemodellen, welche individuell auf eine maximale Zielverformung hin, ausgelegt werden soll, ist die FEM aufwendig.

Das Interesse für künstliche neuronale Netze (KNN) setzte bereits in den frühen 1940er Jahren ein, also etwa gleichzeitig mit dem Einsatz programmierbarer Computer in angewandter Mathematik (David Kriesel, 2005, S. 27). Aber wegen die Beschränkung der Reichenleistung entwickelte sich die KNN sehr langsam. Bereits 1974 entwickelte Paul Werbos für seine Dissertation die Backpropagation bzw. die Fehlerrückführung (Paul Werbos, 1974). Das Modell war aber erst später von einer größeren Bedeutung. Neuronales Netz (NN) ist heutzutage ein sehr erfolgreicher Ansatz aus dem Bereich des maschinellen Lernens, welcher auf Basis von verfügbaren Wissen verallgemeinert. D.h. es lernt nicht das verfügbare Wissen auswendig, sondern erkennt. Auf Grund der Fortschritte die Rechenleistung von CPU und GPU entsteht die Möglichkeit, tieferes und tieferes Neuronales Netz zu antrainieren. Typische eingesetzte Beispiele von maschinellen Lernen sind Klassifikation von Bildern, Text- und Spracheübersetzung, und Empfehlungssystem. Dazu kommt spezielle Struktur des Netzes, z.B. Convolutional Neural Network (CNN) und Rekurrentes neuronales Netz (RNN).

Im Rahmen der Arbeit soll untersucht werden, ob wissensbasierte Methoden (Support Vector Maschine - SVM, Neuronale Netze - NN) ähnlich qualifizierte Aussagen treffen können, wie die numerische Simulation. Diese Arbeit konzentriert sich darauf, wie die maximale Verschiebung eines Bauteils unter bestimmten Lasten vorhergesagt werden kann. Dazu sind an mehreren parametrischen Bauteilen entsprechende SVM und NN aufzustellen, zu dimensionieren und zu validieren. Die benötigten Daten bzw. Wissen wird durch ein Addin-Modul „Simulation“ in einer CAD Software SolidWorks erzeugt. Um die Datengenerierung automatisch laufen zu können, ist die Anruf von SolidWorks durch „Application Programming Interface“ (API) in C# nötig. Die entsprechenden Wertebereiche für Bauteildimensionen und Lasten sind sinnvoll einzuschränken.

Um die vollständig Vorgehensweise zu klären, wird in der Arbeit 3 parametrischen Bauteile als Beispiele genommen, nämlich Balken, L-Profil und W-Profil (Siehe Abb. 1).

()

Die Arbeit ist anhand der Vorgehensweise der Erzeugung eines Vorhersagemodells eingegliedert. Zuerst müssen die benötigten Daten bzw. Wissen durch ein automatisiertes Skript erfassen werden. Danach wird die ansammelte Daten in einem bestimmten Regeln erkundet. Dann folgt das Erstellen eines Vorhersagemodells. Dazu sind mehre Modelle mit unterschiedlichen Strukturen und Parametern sich voneinander verglichen. Im letzten Teil ist die Vertiefungsrichtung dargestellt.

# 2. Erfassung der FEM-basierten Daten

## 2.1 Allgemeines von FEM

Die Methode der finiten Elemente (FEM) ist ein durch numerische Iterationen berechnetes Verfahren, das in vielen Anwendungsbereiche des Maschinenbaus und Elektromagnets zum Einsatz kommt. Die Grundgleichungen zur Beschreibung strukturmechanischer Probleme wie Deformationen, Spannungen, Geschwindigkeiten, Druck, Temperaturen usw., sind gewöhnlich oder partielle Differenzialgleichungen (DGLn) bzw. Differenzialgleichungssysteme (Friedrich U. Mathiak, 2010, S 1.1).

Der Grundgedanke von FEM besteht darin, dass die uns interessierende Bereiche in eine endliche Anzahl einfacher Teilbereiche. In FEM wird dieser Prozess Diskretisierung genannt, in der die großen Bereiche zu zerkleinern. Die Differenzialgleichungen, die das physikalische und mechanische System beschreibt, werden auf der endlichen Anzahl von Elementen gelöst.

## 2.2 Vorgehensweise einer FEM-Analyse

Im Ingenieurbereich stehen vielseitige Simulationssoftwaren zur Verfügung, die eine FEM-Analyse durchführen zu können. Z.B. ABAQUS, ANSYS und SolidWorks (mit Simulation-AddIn). Obwohl es viele Auswahl im Markt gibt, haben alle sehr ähnliche Vorgehensweise. Die beispielsweise in SolidWorks-Simulation können in 7 Schritte eingegliedert (Michael Brand, 2016, S 7).

● Erstellen einer Studie

In der SolidWorks-Simulation Zusatzanwendung stehen nicht nur konventionale lineare Mechanikanalyse, sondern auch mehre Einsatztype zur Verfügung, z.B. Thermische Analyse, Frequenzanalyse, Knickenanalyse usw. (siehe Abb. 2).

In der Arbeit wird auf Festigkeitsanalyse (Static in Abb. 2) konzentriert.



Abb. 2: Einsatztype in SolidWorks Simulation

● Anwenden des Materials

Einige am häufigsten eingesetzte Materialien sind in der Bibliothek vordefiniert und mit dazu entsprechend Name gekennzeichnet. Die benötigten Kennwerte von Werkstoff, z.B. Elastizitätsmodul und Schubmodul, sind auch dazu verbunden.

In der Arbeit wird nur ein Material „AISI 1020“ (DIN C22) als Beispiel untersucht.

● Einspannungen definieren

Ziel davon ist die Bewegungen eines Bauteils im Raum ein oder mehr Freiheitsgrad einzuschränken. Im Allgemeine heißt es auch Randbedingungen Definieren. Der Grund liegt daran, dass die Differenzialgleichungen der Mechanik die Randbedingungen benötigen, um gelöst werden zu ermöglichen. Die Type der Einspannung ist je nach dem einzelnen Szenario der Anwendung.

In der Arbeit wird eine Stirnfläche als fixiert betrachtet.

● Lasten definieren

Im Ingenieurbereich gibt’s unterschiedliche Lasten. Davon sind die äußeren Kräfte und Momente als allgemeine Lasten betrachtet.

In der Arbeit sind nur Kräfte als bestimmte Lasten berücksichtigt.

● Modell vernetzen

Im Rahmen der FEM ist die Vernetzung eines Bauteils entscheidend. Damit entstehen diese endlichen Elemente. Je nach Form, Größe und Anzahl der Elemente können die entsprechenden Ergebnisse sich voneinander abweichen.

Theoretisch stehen in der FEM vielseitige Elementtypklasse zur Verfügung, nämlich für Linienelemente gibt’s Stab und Balken, für Flächenelemente gibt’s Dreieck und Viereck, für Volumenelemente gibt’s Hexaeder und Tetraeder.

Bei SolidWorks Simulation sind drei grundlegende Vernetzungstype verfügbar: shell mesh, beam mesh und solid mesh (Glenn Whyte, 2018). Folgendes steht die Eigenschaft der typischen Vernetzungstype (siehe Tabelle 1).

Tabelle 1: Vergleich der Vernetzungstype bei SolidWorks Simulation

|  |  |  |  |
| --- | --- | --- | --- |
|  | Solid Mesh | Shell Mesh | Beam Mesh |
| Elementtype | Tetraeder | Dreieck | Balken |
| Einsätze | Für alles andere  (als standardmäßige Voreinstellung) | Dünne Bauteile | Bauteile mit konstanter Abschnitt |

Für Solid Mesh bei SolidWorks Simulation bittet der Lieferant keine andere Elementtype, sondern nur Tetraeder. D.h. in der Arbeit spielt der Elementtyp keine Rolle.

Die Größe und die Anzahl der Elemente sind sich voneinander abhängig. Je kleiner die Elemente sind, desto größere Anzahl der Elemente ist.

In der Abb. 3 wird Einfluss der Elementsgröße auf FEM Ergebnisse (maximale Verschiebung) gezeigt. Elementsgröße 0,2mm ist als angenommene Richtige bzw. Referenz betrachtet. Es weist eine Tendenz auf, dass mit absteigenden Elementsgröße neigt es dazu, größere Ergebnisse zu erzielen. Im Vergleich zu großen Elementsgröße besitzen die Vernetzungen, die mit kleinen Elementsgröße versehen sind, einen deutlichen Unterschied zur Referenz, dazu zählen sowohl die Verteilung als auch die Differenz.

In der Arbeit wird der Unterschied nicht weiter qualitativ untersucht, sondern nehmen wir eine quasi vernünftige Elementsgröße 0.2mm als Beispiel. Die ist nach der Abwägung zwischen Genauigkeit und Rechenzeit.



Abb. 3: Einfluss der Elementsgröße auf FEM Ergebnisse

(maximale Verschiebungen)

● Studie ausführen

Wenn alle benötigten Eingangsdaten (Geometrie, Material, Lasten, Vernetzung usw.) zugewiesen sind, dann kann das Berechnungsverfahren selbst automatisch durchlaufen.

● Ergebnisse analysieren

Beim Standard stehen 3 verformungsrelevante Ergebnisse zur Verfügung, nämlich vonMises Spannungen, resultierende Verschiebungen und äquivalente Dehnungen.

In der Arbeit uns interessieren die vonMises Spannungen und resultierende Verschiebungen.

## 2.3 Automatisierung der Vorgehensweise durch SolidWorks-API mittels C# Programmierung

2.3.1 Einführung der C# Programmierung

C# ist eine objektorientierte Programmierungssprache, die erste Version C# 1.0 war in 2002 von Microsoft veröffentlicht. Nach mehr als zehnjährige Entwicklungszeit hat C# große Fortschritte gemacht. Die mit der C# entwickelte Programme laufen auf der Microsoft .NET-Plattform. Dabei handelt es sich um eine virtuelle Maschine, die Programmen, die für diese Plattform entwickelt wurden, einen Prozessor vorgaukelt, der so in Form eines echten Computerchips gar nicht existiert.

In der Arbeit wird die Version C# 7.5.2 eingesetzt.

2.3.2 Einführung der SolidWorks-API

API heißt auch Programmierschnittstelle, mit der von einem Softwaresystem die anderen Programme zur Anbindung an das System zur Verfügung gestellt wird. Es gibt zwei Type von API (A.T. Mathew, 2010). Eine ist mit Abhängigkeit von Programmierungssprache, damit können sowohl die Syntax als auch die Elemente von dieser bestimmten Programmierungssprache zugegriffen werden. Im Gegensatz ist die andere ohne Abhängigkeit von Programmierungssprache. D.h. die kann von mehreren Programmierungssprachen angerufen werden.

SolidWorks ist ein kommerzielles CAD System von Dassault Systemes. Das umfassen vollständige 3D-Modellierung und umfangreiche zusätzliche Anwendungen (auf Englisch Add-In-Module), SolidWorks Simulation ist eine davon. Neben dem bietet Dassault Systemes SolidWorks-APIs in C#, VB und C++ für die Automatisierung und Anpassung der Vorgehensweise. In der Arbeit wird C# als API-Programmierungssprache eingesetzt.

Im SolidWorks gibt’s drei grundlegende Dokumente, nämlich Bauteil, Komponente und Zeichnungen. Entsprechend stehen Drei dazugehörten Objekte in SolidWorks-API, nämlich PartDoc, AssemblyDoc und DrawingDoc. Die Struktur der Objekte in SolidWorks API ist in Abb. 4 dargestellt. SldWorks funktioniert als Wurzel und alle sind davon deriviert. Jede Objekte hat seine eigenen Funktionen, damit die Manipulationen in hinterlegten Applikationen erledigt werden. Darauffolgende ist ein Beispiel.

„*swModel.SketchManager.CreateLine(0, 0, 0, 1, 0, 0);*“

Eine Instanz von ModelDoc2 heißt „*swModel*“. SketechManager ist eine von dazugehörten Objekten. Die fasst alle Manipulationen um, damit der Sketch erstellt und/oder ergänzt werden können. CreateLine ist eine Funktion davon, die eine Linie anhand der vorgegebenen Koordinaten im Sketch hinzufügen kann.

In der Arbeit wird SolidWorks 2016 und dazu entsprechende API verwendet.

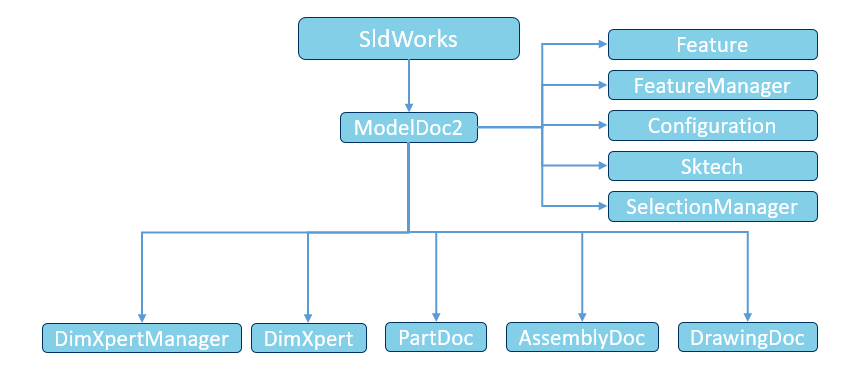


Abb. 4: Struktur der Objekte in SolidWorks API

2.3.3 Automatisierte Vorgehensweise

Für das Training eines wissensbasierten Modells (z.B. Neuronales Netz) ist i.d.R. eine hohe Anzahl von Daten unerlässlich. Bei manchen Anwendungen wie Bilderarbeitung und Spracheerkennung können die benötigten Daten im Internet herausgefunden und heruntergeladen werden. Aber bei dieser Arbeit steht keine bestimmten bereiten Daten zur Verfügung. Trotzdem ist es tausende Daten manuell anzusammeln unrealistisch. Deswegen spielt die Automatisierung der Vorgehensweise eine große Rolle.

Je nach der Geometrie und Lasten unterscheidet die Anzahl des Eingangsfeatures (siehe xxx) sich. Hauptsache ist es mit dieser Eingangsfeature der Bauteil und dessen äußeren Lasten vollständig definiert werden zu können. Tabelle 2 zeiget die Beispiele (siehe Abb. 1) und dazugehörte Anzahl des Eingangsfeatures.

Tabelle 2: Anzahl des Eingangsfeatures bei unterschiedlichen Szenen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Balken | L-Profil | W-Profil | W-Profil  (Planare Deformation) |
| Anzahl des Eingangsfeatures | 6 | 7 | 9 | 8 |

Für eine gute Verallgemeinerung eines Vorhersagesmodells sollen die gefütterte Traningsdaten möglichst zufällig sein. D.h. die Daten dürfen keine subjektiven Präferenzen enthalten. In C# ist Random-Funktion eine der Möglichkeiten, Zufallszahl zu erzeugen. Man muss darauf beachten, dass Random-Funktion in C# keine mathematisch echte Zufallszahl erzeugen kann, sondern Pseudozufallszahl.

In der Arbeit wird die Pseudozufallszahl verwendet. Der Unterschied der Wirkung auf Vorhersagesmodell zwischen Pseudozufallszahl und mathematisch echten Zufallszahl wurde nicht tiefer untergesucht.

Das Wertebereich von Eingangsfeature beschränkt sich innerhalb einer sinnvollen Grenz. Der Grund liegt daran, dass die Komplexität des Vorhersagesmodells sich auch in einem gewissen Niveau befindet. Dazu entsprechen es ein relativ kleiner Umfang von Trainingsdatensatz und eine quasi niedrige Rechenleistung. In Tabelle 3 steht das Wertbereich des Eingangsfeatures von bespielweise W-Profil-Bauteile.

Tabelle 3: Wertbereich des Eingangsfeatures von W-Profil-Bauteile

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | m | width | thickness | L1 | L2 | L3 | F1 | F2 | F3 |
| Wertbereich | [2, 4] | [3, 10] | [1, 10] | [24, 32] | [15, 21] | [14, 22] | [0, 50] | [0, 50] | [0, 50] |

Die manuelle Vorgehensweise einer FEM-Studie ist schon in Kapitel 2.2 gezeigt. Bevor jeder räumlichen FEM-Studie muss 3D-Modell vorhanden sein. D.h. zuerst muss man ein geometrisches 3D-Modell anhand der durch die Random-Funktion generierten Eingangsdaten erstellen. Danach läuft eine entsprechende FEM-Studie durch. Um vollständige automatische Durchführungen zu realisieren, wird eine for-Schleife im Programm geschrieben. Der konkrete C# Quellecode wird im Anhang beigefügt.

2.3.4 Ansammlung der Trainingsdaten

Am Ende jeder for-Schleife können alle FEM-Ergebnisse durch dazu entsprechende API-Funktion abgelesen werden. Ziel der Arbeit ist es die maximale Verschiebung eines Bauteils unter äußeren Lasten vorherzusagen, deswegen ist die maximale Verschiebung im Trainingsdatensatz unerlässlich. Darüber hinaus funktioniert die entsprechende maximale Spannung als ein zusätzlicher Faktor, der sich um die Streckgrenze behandelt (siehe Kapitel xxx). Nach erfolgreicher Ablesen von FEM-Ergebnissen wird die Zielgröße durch Microsoft-Excel-API in einer Tabelle (.xlsx) exportiert.

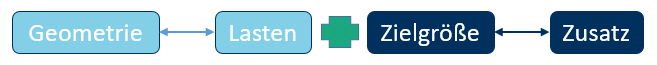


Abb.5: Blockdiagramm der Datenstruktur

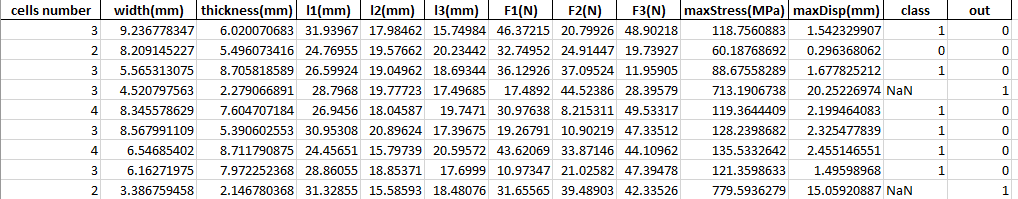


Abb. 6: Aufbau der Datenstrukturen von W-Profil

(räumliche Deformation)

I.d.R. ist ein Datensatz wie Abb. 5 von vier Teilen aufgebaut, nämlich Geometrie, Lasten, Zielgröße und Zusatz (wenn nötig). In der Arbeit wird eine Kombination von die 4 (o. 3) Eigenschaften eine Datei genannt. Z.B. eine Zeile in Abb.6 heißt eine Datei.

In Abb. 6 wird Aufbau der Datenstrukturen von W-Profil mit räumlicher Deformation dargestellt. Davon sind die zwei „maxStress(MPa)“ und „maxDisp(mm)“ die obere genannte zwei Zielgröße. Rechts liegen die durch Zielgröße abgeleitet Eigenschaften, „class“ und „out“, die als Zusatz betrachtet und in Kapitel xxx besprecht werden.

# 3 Daten erkunden

Nach der automatisierten Datenerfassung steht eine Tabelle zur Verfügung, die alle Eingangsparameter und Zielgröße enthaltet. Beim Erkunden von Daten geht es um Statistik. D.h. in diesem Schritt werden alle Eigenschaften, die nicht direkt entdeckt werden können, mit Hilfe der statistischen Beschreibungen einfach und explizit interpoliert. Dazu zählen sowohl die Verteilung als auch die sich miteinander verbundenen Korrelationsindex. Ziel davon ist es, Universalität von Daten zu überprüfen, die bei der Verallgemeinerung des Vorhersagesmodells eine entscheidende Rolle spielt.

## 3.1 Korrelation innerhalb des Datensatz

Eine Korrelation beschreibt eine Beziehung zwischen zwei oder mehreren Merkmalen, Ereignissen, Zuständen oder Funktionen. Die Maßzahlen der Korrelation liegen betragsmäßig meist in einem Bereich von -1 (ein vollständig negativer linearer Zusammenhang) bis 1 (ein vollständig positiver linearer Zusammenhang). Für zwei quadratisch integrierbare Zufallsvariablen *X* und *Y* mit jeweils positiver Standardabweichung bzw. und Kovarianz Cov(*X*, *Y*) ist der Korrelationskoeffizient (Pearsonscher Maßkorrelationskoeffizient) definiert durch:

In Abb. 7 wird die Korrelation des Eingangsfestures punktweise dargestellt. Die Diagonale steht die sogenannte Kerndichteschätzung, die wird im Kapitel 3.2 erläutert. Außer der Diagonale sind die andere Zelle paarweise Diagramm, in dem wird ein Feature zu alle anderen punktweise aufgezeichnet. Die entsprechenden pearsonscher Korrelationskoeffizienten befinden sich in Abb. 8. Davon liegen alle Werte (außer der Diagonale) in der Nähe von Null, deswegen weißt das Eingangsfeature kein Zusammenhang sich miteinander auf. D.h. bei solchem Datensatz importiert man keine objektive Präferenz, die eine wichtige Voraussetzung beim Trainieren des Vorhersagesmodells ist. Trotzdem bei manchen Situationen würden einige richtige Vorwissen zum Datensatz geführt (Siehe Kapitel xxx).



Abb. 7: Scatter-Matrix des Eingangsfeatures



Abb. 8: Korrelationskoeffizienten des Eingangsfeatures

Literaturverzeichnis

David Kriesel, Ein kleiner Überblick über Neuronale Netze, www.dkriesel.com, 2005

Paul Werbos, Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, Dissertation, Harvard University, 1975

Michael Brand, FEM-Praxis mit SolidWorks, Springer Vieweg, 2016

Glenn Whyte, Ins and outs on meshing elements for SOLIDWORKS SIMULATION, https://hawkridgesys.com/blog/ins-and-outs-on-meshing-in-solidworks-simulation, 2018

A.T. Mathew, C.S.P. Rao, A Novel Method of Using API to Generate Liaison Relationships from an Assembly, J. Software Engineering & Applications, 2010