Facharbeit Informatik

Joel Mantik

4. März 2023

Inhaltsverzeichnis

1 Einleitung **2**

2 Theoretische Grundlagen **3**
2.1 Lineare Gleichungssysteme 3
2.2 Gaußsches Eliminationsverfahren 4

3 Implementierung des Algorithmus **5**
3.1 Schritte des Verfahrens 5
3.2 Erläuterung des Quellcodes 6
3.3 Beispielrechungen 6
3.4 Schwächen des Algorithmus 6

4 Anwendungen des Gaußschen-Eliminationsverfahrens **7**
4.1 Lösung von linearen Gleichungssystemen 7
4.2 Inversion von Matrizen 7
4.3 Beispielanwendungen 7

5 Abwägungen **8**
5.1 Vorteile im Vergleich zu anderen Methoden 8
5.2 Nachteile im Vergleich zu anderen Methoden 8

6 Fazit **9**

7 Anhang **10**

Kapitel 1

Einleitung

> "The simplest model in applied mathematics is a system of linear equations. It is also by far the most important."
> GILBERT STRANG

Der Gauß-Algorithmus ist eins der wichtigsten Lösungsverfahren zum Lösen lineare r Gleichungssysteme. Er spielt eine tragende Rolle in vielen Bereichen der Mathematik und ist dennoch recht unkompliziert. Aufgrund der Wichtigkeit, habe ich dazu entschieden, den Algorithmus in dieser Facharbeit zu implementieren, zu analysieren, und Anwendungsmöglichkeiten aufzuzeigen.

Kapitel 2

Theoretische Grundlagen

Der Gauß-Algorithmus ist ein Algorithmus, welcher beim Lösen von linearen Gleichungssystemen zum Einsatz kommt. Im folgenden Kapitel wird die mathematische Theorie von linearen Gleichungssystemen und dem Gauß-Algorithmus erläutert, welche die Grundlage für die spätere Implementierung sind. Die folgenden Definitionen sind sinngemäß aus [Gra21] entnommen.

2.1 Lineare Gleichungssysteme

Ein lineares Gleichungssystem ist eine Sammlung von Gleichungen, in denen jede Unbekannte mit höchstens dem ersten Grad vorkommt. Es kann in der Form $A_x = b$ geschrieben werden, wobei A eine $m * n$ Matrix ist, x ein n-dimensionaler Vektor von Unbekannten und b ein m-dimensionaler Vektor von Konstanten ist. Ein allgemeines lineares Gleichungssystem lässt sich wie folgt definieren. :

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m \end{aligned}$$

Das Ziel eines solchen linearen Gleichungssystems ist es, eine Lösung für x zu finden, die alle Gleichungen erfüllt. Hierbei gibt es drei Arten von Lösungen:

1. Das Gleichungssystem hat genau *eine* Lösung; es gibt genau eine Lösung, welche alle Gleichungen im System erfüllt. Die Lösungsmenge ist z. B. : $\mathbb{L} = \{(x, y, z)|(1, 2, 3)\}$.

2. Das Gleichungssystem hat *keine* Lösung, wenn es keine Lösung gibt, die alle Gleichungen erfüllt. Die Lösungsmenge ist eine leere Menge: $\mathbb{L} = \{\}$.

3. Das Gleichungssystem hat *unendlich viele* Lösungen, wenn es mehrere Lösungen gibt, die alle Gleichungen im System erfüllen. Hierbei sind die verschiedenen Variablen voneinander abhängig. Die Lösungsmenge sieht beispielsweise wie folgt aus:
   $\mathbb{L} = \{(x, y, z)|(x = y * z, y \in \mathbb{R}, z \in \mathbb{R})\}$.

2.2 Gaußsches Eliminationsverfahren

Kapitel 3

Implementierung des Algorithmus

Für die im Folgenden dargelegte Implementation und Analyse des Algorithmus wurde die Programmiersprache "Java"verwendet.

3.1 Schritte des Verfahrens

Grundlegend lässt sich der Algorithmus durch folgende Schritte implementieren, es gibt bei gewissen Implementationen (wie der Brute-Force Methode) zwar kleine Abweichungen jedoch folgen auch diesen demselben Prinzip.

1. Eingabe: Ein lineares Gleichungssystem $A_x = b$ mit einer $n * n$ Matrix A und einem n-dimensionalem Vektor b.
2. Initialisierung:
3. Pivotisierung: Finden des größten Elements a_{ki} in der Spalte von A unterhalb der Diagonalen und tauschen der Zeilen k und i von A und B.
4. Elimination:
5. Rückwärtssubstitution:

Der Aufbau des Algorithmus wird in folgendem Diagramm deutlich:

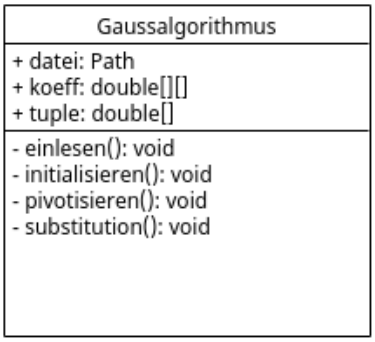

Abbildung 3.1: Implementationsdiagramm der Klasse Gauß-Algorithmus

3.2 Erläuterung des Quellcodes

3.3 Beispielrechungen

3.4 Schwächen des Algorithmus

Kapitel 4

Anwendungen des Gaußschen-Eliminationsverfahrens

4.1 Lösung von linearen Gleichungssystemen

4.2 Inversion von Matrizen

4.3 Beispielanwendungen

Kapitel 5

Abwägungen

5.1 Vorteile im Vergleich zu anderen Methoden

5.2 Nachteile im Vergleich zu anderen Methoden

Kapitel 6

Fazit

Kapitel 7

Anhang

```
1
2  import java.nio.charset.StandardCharsets;
3  import java.nio.file.Files;
4  import java.nio.file.Path;
5  import java.nio.file.Paths;
6  import java.io.IOException;
7  import java.util.List;
8
9  public class Gauss {
10     static Path datei = Paths.get("./input1.txt");
11     static double[][] koeff = new double[3][3];
12     static double[] tuple = new double[3];
13
14     public static void main(String[] args) throws IOException
       {
15         einlesen();
16     }
17
18     private static void einlesen() throws IOException {
19         List<String> f = Files.readAllLines(datei,
       StandardCharsets.UTF_8);
20         int zeilenIndex = 0;
21         for (int i = 0; i < f.size(); i++) {
22             String zeile = f.get(i);
23             if (zeile.isEmpty() || zeile.charAt(0) == '#') {
24                 continue;
25             }
26
27             String[] zeilenElemente = zeile.replace(',', '.')
       .split("\\s+");
28             for (int spaltenIndex = 0; spaltenIndex <
       zeilenElemente.length; spaltenIndex++) {
29                 String element = zeilenElemente[spaltenIndex
```

```
    ];
                double wert = Double.parseDouble(element);
                if (spaltenIndex < 3) {
                    koeff[zeilenIndex][spaltenIndex] = wert;
                } else {
                    tuple[zeilenIndex] = wert;
                }
            }
            zeilenIndex++;
        }
        System.out.println("Koeffizienten:");
    for (int i = 0; i < koeff.length; i++) {
        for (int j = 0; j < koeff[i].length; j++) {
            System.out.print(koeff[i][j] + " ");
        }
        System.out.println();
    }

    System.out.println("Tupel:");
    for (int i = 0; i < tuple.length; i++) {
        System.out.print(tuple[i] + " ");
    }
    System.out.println();
    }
}
```

Listing 7.1: Quellcode des Gauß-Algorithmus

Literatur

[Gra21] Günther Gramlich. *Lineare Algebra: Eine Einführung*. Carl Hanser Verlag GmbH Co KG, 2021.