Berufsakademie Sachsen

Staatliche Studienakademie Leipzig

**Randwertprobleme für Differentialgleichungen 2. Ordnung**

Eingereicht von: Matthias Thurow

CS13-2

Sebastian Wolff

CS13-1

Inhalt

[Verbale Erklärung 3](#_Toc431315852)

[Aufteilung Programmierung 4](#_Toc431315853)

[Programmausgabe 4](#_Toc431315854)

# **Verbale Erklärung**

Zu Beginn werden zwei Objekte erzeugt. Das erste Objekt beinhaltet den Runge-Kutta Algorithmus und hat daher auch diesen Namen. Das zweite Objekt dient später dazu, um zwei AWP‘s in ein RWP umzuwandeln. Nach einigen Standardinformationen die das Programm ausgibt werden nun zwei Arrays vom Typ double erzeugt.

Dies geschieht durch die Methode *rungeKutta* welche vom gleichnamigen Objekt aufgerufen wird. Für den Methodenaufruf dieser Methode werden fünf Parameter benötigt. Der erste Parameter gibt den Y-Wert bei y(0) an, der zweite den Funktionswert für y‘(0). Der dritte Parameter ist ein Zielwert der mindestens erreicht werden soll. Hierbei muss der Wert jedoch größer sein als der Wert des ersten Parameters. Der vierte Parameter gibt die genutzte Schrittweite an, die benutzt wird, welche jedoch dann auch überall übereinstimmen muss. Der letzte Parameter gibt an welches AWP man zurückbekommen möchte. Da das Problem auf zwei AWP’s reduziert wird kann man zwischen den Werten 1 und 2 wählen. Dieser Parameter war notwendig, da wir die Funktionen fest codiert haben und entschieden werden muss für welche Funktion der Runge-Kutta Algorithmus ausgeführt werden soll.

In der Methode rungeKutta wird auch zunächst ein Array initialisiert um die Werte später wie gefordert als Wertetabelle ausgeben zu können. Danach werden die Anfangswerte von y und y‘ gesetzt, welche als Parameter übergeben wurden. X wird mit 0 initialisiert, da es sich jeweils um AWP’s handelt. Außerdem wird die Schrittweite durch den Parameter gesetzt.

Nun wird mithilfe des fünften Methodenparameters entschieden auf welche Funktion das Runge-Kutta Verfahren angewendet werden soll. Daraufhin wird dann solange das Verfahren durchgeführt bis der Zielwert erreicht wurde. Zum Schluss wird das gesamte Array für y zurückgegeben.

Die Methoden, welche mit f beginnen, beinhalten die genutzten Funktionen für das Runge-Kutta Verfahren. Diese wurden fest codiert da dynamische Anpassung von Formeln in Java nicht ohne weiteres möglich ist. Welche Funktion hierbei welche Aufgabe übernimmt ist in den Quelltext-Kommentaren genauer beschrieben.

Nachdem die beiden Arrays y1 und y2 durch das Runge-Kutta Verfahren mit Werten gefüllt worden sind, werden nun die beiden Anfangswertprobleme auf ein Randwertproblem reduziert. Dafür wird die als erstes die Methode calculateC an der Hilfsklasse AwpToRwp aufgerufen um die benötigte Formelkonstante zu berechnen.

Anschließend werden die Probleme reduziert indem die Methode toRwp aufgerufen wird.

Diese Methode erwartet wieder drei Parameter. Die ersten beiden Parameter sind die Arrays welche man vom Runge-Kutta Verfahren erhalten hat. Der letzte Parameter ist die eben genannte Konstante. Als Ergebnis erhält man die Lösung des Randwertproblems.

Zum Schluss wird noch die Betrachtung für h -> 0 vorgenommen. Die Werte dafür werden nach der oben genannten Lösung ausgegeben.

# **Aufteilung Programmierung**

Wer welchen Teil programmiert hat kann an den Kommentaren im Quelltext erkannt werden. Immer, wenn dort ein Name aufgeführt wird, wurde der Teil bis zum nächsten Namen von demjenigen programmiert.

# **Programmausgabe**

Loesung folgender DGL 2. Ordnung als Randwertproblem:

-y'' + y = t^3 , y(0) = 2, y(2) = 1

1. 2 Anfangswertprobleme mit Runge-Kutta

I. y'' = y - t^3 ; y(0)=2 ; y'(0)=0

II. y'' = y ; y(0)=0 ; y'(0)=1

t = 0.0 ; y = 2.0

t = 0.1 ; y = 2.0174539583333333

t = 0.2 ; y = 2.03483125

t = 0.3 ; y = 2.0518643750000005

t = 0.4 ; y = 2.068073333333334

t = 0.5 ; y = 2.0827656250000004

t = 0.6 ; y = 2.09503625

t = 0.7 ; y = 2.1037677083333337

t = 0.8 ; y = 2.1076300000000003

t = 0.9 ; y = 2.1050806250000003

t = 1.0 ; y = 2.0943645833333338

t = 1.1 ; y = 2.0735143750000002

t = 1.2 ; y = 2.04035

t = 1.3 ; y = 1.9924789583333338

t = 1.4 ; y = 1.9272962500000004

t = 1.5 ; y = 1.8419843750000005

t = 1.6 ; y = 1.733513333333334

t = 1.7 ; y = 1.5986406250000007

t = 1.8 ; y = 1.4339112500000004

t = 1.9 ; y = 1.2356577083333335

t = 2.0 ; y = 1.0

Betrachtung h -> 0:

h = 1.0 ; t = 1.0 ; y = 2.145833333333333

h = 0.5 ; t = 1.0 ; y = 2.12890625

h = 0.25 ; t = 1.0 ; y = 2.109212239583333

h = 0.125 ; t = 1.0 ; y = 2.09698486328125

h = 0.0625 ; t = 1.0 ; y = 2.0903294881184897

h = 0.03125 ; t = 1.0 ; y = 2.0868730545043945

h = 0.015625 ; t = 1.0 ; y = 2.0851134856541957

h = 0.0078125 ; t = 1.0 ; y = 2.084225967526436

h = 0.00390625 ; t = 1.0 ; y = 2.08378028807541

h = 0.001953125 ; t = 1.0 ; y = 2.0835569698829204

h = 9.765625E-4 ; t = 1.0 ; y = 2.083445191373672

h = 4.8828125E-4 ; t = 1.0 ; y = 2.0833892722912424

h = 2.44140625E-4 ; t = 1.0 ; y = 2.0833613052963043

h = 1.220703125E-4 ; t = 1.0 ; y = 2.0833473199357417

h = 6.103515625E-5 ; t = 1.0 ; y = 2.083340326789946

h = 3.0517578125E-5 ; t = 1.0 ; y = 2.083336830100359

h = 1.52587890625E-5 ; t = 1.0 ; y = 2.0833350817272773