Yves Radunz
509242



# 1 Aufgabenstellung

Ich habe mich entschieden, die erste Aufgabe zu bearbeiten. Die Aufgabenstellung lautet:

Ein einfaches Modell für ein Fahrzeug, das über Straßenunebenheinten fährt, kann durch ein Feder-Dämpfungssystem beschrieben werden, dass gemäß dem NEWTON'schen Kraftgesetz durch die Differentialgleichung

$$my''(x) = -mg - k(y(x) - u(x)) - d(y'(x) - u'(x))$$

gegeben ist. Die Funktion $u(x)$ modelliert die Straßenunebenheiten.
Sei $u(x) = sin(x),\ k = 4,\ d = 2,\ m = 10,\ g = 9.81$. Als Randwerte sind gegeben: $y(0) = 10,\ y(2) = 1$.
Die Diskretisierung von Randwertaufgaben der Art

$$\begin{aligned} y''(x) &= r(x)y(x) + s(x)y'(x) + q(x) \qquad x \in [0, b] \\ y(0) &= y_0 \\ y(b) &= y_b \end{aligned}$$

mittels

$$0 = x_0 < x_1 < ... < x_N = b$$

$$x_i = \frac{i}{N}b, \quad h = \frac{b}{N}, \quad i = 0, 1, 2, ..., N$$

$$y''(x_i) \approx \frac{1}{h^2}(y(x_i + h) - 2y(x_i) + y(x_i - h))$$

$$y'(x_i) \approx \frac{1}{2h}(y(x_i + h) - y(x_i - h))$$

$$y(x_i) = y_i,\ f(x_i) = f_i,\ r(x_i) = r_i,\ s(x_i) = s_i,\ q(x_i) = q_i$$

führt auf ein lineares Gleichungssystem mit Tridiagonalgestalt:

$$\begin{aligned} y_{i-1}(-1 - \tfrac{h}{2}s_i) + y_i(2 + h^2 r_i) + y_{i+1}(-1 + \tfrac{h}{2}s_i) &= -h^2 q_i \qquad i = 1, 2, ..., N-1 \\ y_0 &= y_0 \\ y_N &= y_b \end{aligned}$$

Implementieren Sie einen Algorithmus zur Lösung von Gleichungssystemen mit Tridiagonalgestalt!
Berechnen Sie je eine Näherungslösung für $N \in \{2, 5, 10\}$ Diskretisierungspunkte!

# 2 Vorbemerkungen

Zunächst lässt sich die Gleichung $my''(x) = -mg - k(y(x) - u(x)) - d(y'(x) - u'(x))$ äquivalent umformen in $y''(x_i) \approx \frac{1}{h^2}(y(x_i + h) - 2y(x_i) + y(x_i - h))$ mit $r(x) = -\frac{k}{m}$, $s(x) = -\frac{d}{m}$ und $q(x) = -g + \frac{k}{m}u(x) + \frac{d}{m}u'(x)$. Damit gilt die in dem Programm verwendeten Definitionen von `func_r` und `func_s`. Bei `func_q` gilt offensichtlich auch $q(x) = -g - r(x)u(x) - s(x)u'(x)$.

Zur Bestimmung der Einträge 0 bis $N-1$ des Lösungsvektors $x$ wird die Gleichung $a_{j,k}x_k + a_{j,k+1}x_{k+1} = b_k$ verwendet, welche äquivalent zu $x_k = (b_k - a_{j,k+1}x_{k+1})(a_{j,k})^{-1}$ ist.

# 3 Programmablauf

Das Programm erzeugt nach dem Start (`javac Main.java && java Main`) folgende Ausgabe:

```
Wissenschaftliches Rechnen II
Loesung: Serie 4 - Aufgabe 1
Copyright (C) 2006 Yves Radunz


+-------- N =  2 --------+
|                        |
|  10.0                  |
|  11.878096884994504    |
|  1.0                   |
|                        |
+------------------------+


+-------- N =  5 --------+
|                        |
|  10.0                  |
|  8.713622032147486     |
|  5.622387063344147     |
|  3.8559821163266417    |
|  2.519666840145476     |
|  1.0                   |
|                        |
+------------------------+


+-------- N = 10 --------+
|                        |
|  10.0                  |
|  7.083160355226845     |
|  3.858376242155094     |
|  2.2814448994248364    |
|  1.5111077489412814    |
|  1.1367778790235494    |
|  0.9590792628183569    |
|  0.883154215568302     |
|  0.8676643272785113    |
|  0.9020988607543501    |
|  1.0                   |
|                        |
+------------------------+
```

Die Ausgabe enthält die Werte von $y_0$ bis $y_N$ in eben dieser Reihenfolge für $N \in \{2, 5, 10\}$.

# 4 Auswertung

Für $N = 2$ erhält man eine Näherung, die auf ein Maximum in der Nähe von $x = 1$ hindeutet.
Für $N = 5$ scheint $y(x)$ fast linear zu fallen, während für $N = 10$ die Funktion erst schnell fällt und bei $x \approx 1.7$ ein lokales Minimum annimmt.
Diese Unterschiede liegen jedoch nur zu einem kleinen Teil durch die Fehlerfortpflanzung bei der Berechnung der Lösung der linearen Gleichungssysteme für größere Werte von $N$ begründet.
Vielmehr wirkt sich hier aus, dass die verwendete Näherungen für $y'(x)$ und $y''(x)$ besser werden, je kleiner der Abstand $h$ zwischen zwei benachbarten Diskretiserungspunkten ist. Für sehr kleine $N$ werden die Fehler auf Grund der ungenauen Abschätzung der Ableitungen so groß, dass sie das Ergebnis verfälschen.
Wenn $N$ größer gewählt wird (z.B. $N \geq 10$) wird der Betrag von $h$ kleiner, was die Fehler

$$\left|y'(x_i) - \tfrac{1}{2h}(y(x_i + h) - y(x_i - h))\right|$$
und
$$\left|y''(x_i) - \tfrac{1}{h^2}(y(x_i + h) - 2y(x_i) + y(x_i - h))\right|$$

wegen

$$\lim_{h\to 0} \tfrac{1}{2h}(y(x_i + h) - y(x_i - h)) = y'(x_i)$$
und
$$\lim_{h\to 0} \tfrac{1}{h^2}(y(x_i + h) - 2y(x_i) + y(x_i - h)) = y''(x_i)$$

ebenfalls verkleinert.
Auf Grund der Rechenungenauigkeit lassen sie sich zwar nicht beliebig verkleinern, indem man $N$ vergrößert, aber zumindest anfangs verbessert sich die Qualität des Ergebnisses.