D-ITET, D-MATL

Musterlösung Prüfung Numerische Methoden, Winter 2013

Dr. Lars Kielhorn

1. a) i) Wir definieren $f(t,y) = \lambda y$ und erhalten damit

$$y'(t) = \underbrace{\lambda y(t)}_{=f(t,y(t))}.$$

ii) Das implizite Eulerverfahren für eine Differentialgleichung der Form y'(t) = f(t, y(t)) ist

$$y_{i+1} = y_i + h f(t_{i+1}, y_{i+1}), \quad i = 0, 1, \dots$$

iii) Wir setzen f ein und erhalten

$$y_{i+1} = y_i + h\lambda y_{i+1}, \quad i = 0, 1, \dots$$

Das ist äquivalent zu

$$(1 - h\lambda)y_{i+1} = y_i, \quad i = 0, 1, \dots$$

und damit können wir (für $h\lambda \neq 1$) nach y_{i+1} auflösen:

$$y_{i+1} = \frac{y_i}{1 - h\lambda}, \quad i = 0, 1, \dots$$

b) Dies können wir direkt implementieren:

```
function [t,y] = implicitEuler(h, T, y0, lambda)
% gibt die Zeitpunkte t_i im Vektor t und
% die Naeherungen y_i im Vektor y zurueck
%
% h: Schrittweite
% T: Endzeitpunkt (Vielfaches von h)
% y0: Startwert
t=(0:h:T)';
y=zeros(size(t));
y(1)=y0;
for i = 1:length(t)-1;

y(i+1)=y(i)/(1-h*lambda);
```

c) Wir sehen direkt aus der Interationsvorschrift, dass für $G(z) = \frac{1}{1-z}$

$$y_{i+1} = \underbrace{\frac{1}{1 - h\lambda}}_{G(h\lambda)} y_i$$

end

gilt. Damit ist $G(z) = \frac{1}{1-z}$ die Stabilitätsfunktion des impliziten Eulerverfahrens.

d) Das (reelle) Stabilitätsgebiet ist

$$S = \{ z \in \mathbb{R} : |G(z)| < 1 \}.$$

Für
$$G(z) = 1 + z$$
 ist

$$|G(z)| < 1 \iff 1 + z < 1 \text{ und } -1 - z < 1$$

 $\iff z < 0 \text{ und } -2 < z$
 $\iff -2 < z < 0$

Also

$$S = \{ z \in \mathbb{R} : -2 < z < 0 \} = (-2, 0).$$

e) Das Verfahren ist stabil, wenn $z = h\lambda \in S$. Nach der vorherigen Aufgabe ist

$$z = h\lambda \in S \iff -2 < z < 0$$

Da h>0 und wir nur $\lambda<0$ betrachten ist $h\lambda=z<0$ automatisch erfüllt. Wegen

$$-2 < z = h\lambda \Longleftrightarrow \frac{-2}{\lambda} > h$$

ist das Verfahren V stabil, genau dann wenn die Schrittweite durch $h < \frac{-2}{\lambda}$ beschränkt ist. Das Verfahren V ist also nur bedingt stabil.

2. a) Anwenden von log ergibt

$$\log f(t) = \log \alpha + (-(t-\beta)^2) = \underbrace{\log \alpha - \beta^2}_{=:\alpha} + 2t\beta - t^2$$

wobei wir die neue Unbekannte a eingeführt haben.

Es soll also (möglichst genau) gelten, dass

$$\log f_i = a + 2t_i\beta - t_i^2, \quad i = 1, \dots m,$$

bzw.

$$\underbrace{(1,2t_i)}_{A_{i,1:2}}\underbrace{\begin{pmatrix} a \\ \beta \end{pmatrix}}_{x} = \underbrace{\log f_i + t_i^2}_{b_i}, \quad i = 1, \dots m.$$

Damit haben wir also ein überbestimmtes Gleichungssystem Ax = b mit $A = (A_{i,j}) \in \mathbb{R}^{m \times 2}$, $x \in \mathbb{R}^2$ und $b = (b_i) \in \mathbb{R}^m$.

Die ursprünglichen Parameter erhält man damit über $\beta = x_2$ und $\alpha = \exp(x_1 + \beta^2)$ (muss nicht angegeben werden).

b) Der Standard-Matlab-Befehl für lineare Ausgleichsprobleme ist "\":

```
function [x] = ausglStandard( A, b)
% loest das Ausgleichsproblem Ax=b mit dem Standardmatlabbefehl
    x= A\b;

function [x] = ausglQR( A, b)
% loest das Ausgleichsproblem Ax=b mit Hilfe der QR-Zerlegung, m>=n
    [m n]=size(A);

[Q,R]=qr(A);

c=Q'*b;

x= R(1:n,1:n)\c(1:n);
```

c) Durch Einsetzen und Ableiten sehen wir

$$F_{i} = f(t_{i}) - f_{i} = \alpha \exp(-(t_{i} - \beta)^{2}) - f_{i}$$

$$\frac{\partial}{\partial \alpha} F_{i} = \frac{\partial}{\partial \alpha} \left(\alpha \exp(-(t_{i} - \beta)^{2}) \right) = \exp(-(t_{i} - \beta)^{2})$$

$$\frac{\partial}{\partial \beta} F_{i} = \frac{\partial}{\partial \beta} \left(\alpha \exp(-(t_{i} - \beta)^{2}) \right) = \alpha \exp(-(t_{i} - \beta)^{2})(-2(t_{i} - \beta)(-1))$$

$$= \alpha \exp(-(t_{i} - \beta)^{2})2(t_{i} - \beta)$$

Die Jacobimatrix ist damit

$$J = (\frac{\partial}{\partial \alpha} F, \frac{\partial}{\partial \beta} F) = (\exp(-(t_i - \beta)^2), \alpha \exp(-(t_i - \beta)^2) 2(t_i - \beta))_{i=1}^m \in \mathbb{R}^{m \times 2}$$

```
d) function x=GaussNewton(x0, F, J, tol, maxit)
% loest das nichtlineare Ausgleichsproblem F(x)=0
% mit dem Gauss-Newton-Verfahren
%
% F(x): Residuumvektor F an der Stelle x
% x0: Startvektor
% J(x): Jacobi-Matrix von F an der Stelle x
% tol: Toleranz
% maxit: maximale Anzahl Iterationen

x=x0;
for i=1:maxit
%Gauss-Newton-Korrektur berechnen

s = J(x) \F(x);
```

%Korrektur anwenden

e) Es werden unterschiedliche Residuuen minimiert. Beim linearen Ausgleichsproblem aus a) wird

$$||Ax - b||_2 = ||(\log \alpha - \beta^2 + 2t_i\beta - t_i^2 - \log f_i)_{i=1}^m||_2$$

minimiert, während beim nichtlinearen Ausgleichsproblem c)

$$||F(x)||_2 = ||(f(t_i) - f_i)_{i=1}^m||_2 = ||(\alpha \exp(-(t_i - \beta)^2) - f_i)_{i=1}^m||_2$$

minimiert wird.

3. a)

$$x(\xi) = \frac{1}{2}(\xi + 1), dx = \frac{1}{2}d\xi \implies I[f] = \frac{1}{2}\int_{-1}^{1} f(\frac{1}{2}(\xi + 1)) d\xi$$
 (1)

- **b**) Gerade im Log-log Plot. Also Algebraische Konvergenz.
 - Gauss-Quadratur hat exponentielle Konvergenz, sofern der Integrand glatt ist. Aber: Singularität des Integranden bei $x_0 = 0$ und deswegen beobachten wir nur algebraische Konvergenz.
- c) i) Es werden nur Polynome vom Grad 1 betrachtet, d.h., es ist $\max(k) = 1$. Damit folgt n = 1. Es wird also ein Gewicht und eine Stützstelle benötigt.

ii)

$$\int_0^1 x^0 \log(x) \, \mathrm{d}x = [x(\log(x) - 1)]_0^1 = -1 \stackrel{!}{=} \hat{\omega}_1$$

$$\int_0^1 x \log(x) \, \mathrm{d}x = \left[\frac{x^2}{2} \log(x)\right]_0^1 - \int_0^1 \frac{x}{2} \, \mathrm{d}x = \left[\frac{x^2}{2} (\log(x) - \frac{1}{2})\right]_0^1 = -\frac{1}{4} = \hat{\omega}_1 \hat{\xi}_1$$

$$\implies \hat{\omega}_1 = -1, \ \hat{\xi}_1 = \frac{1}{4}.$$

d)

$$\widetilde{Q}[2x+1] = \sum_{i=1}^{2} \widetilde{\omega}_{i}(2\widetilde{\xi}_{i}+1) = -\frac{3}{2} \cdot \underbrace{(2\frac{1}{2}+1)}_{2} + \underbrace{\frac{1}{2} \cdot \underbrace{(2\cdot 1+1)}_{3}}_{2} = -\frac{6}{2} + \frac{3}{2} = -\frac{3}{2}$$

4. a) Überprüfen von $|\phi_i'(x^*)| \leq 1$.

$$\phi_1'(x) = -\frac{1}{2} \frac{\exp(x)}{\sqrt{2 - \exp(x)}}$$
$$\phi_2'(x) = -\frac{2x}{2 - x^2}$$

Zu $\phi_1'(\frac{1}{2})$: Es ist $\left(\phi_1'(\frac{1}{2})\right)^2 = \frac{1}{4} \frac{\sqrt{e^2}}{2 - \sqrt{e}} \approx \frac{1}{4} \frac{1.6^2}{0.4} = 1.6 > 1$, keine lokale Konvergenz. $|\phi_2'(\frac{1}{2})| = \frac{4}{7} < 1$, lokale Konvergenz.

b) Eine Fixpunktiteration ist konsistent, wenn

$$f(x^*) = 0 \Leftrightarrow \phi(x^*) = x^*$$

gilt. Wegen

$$\phi_3(x) = x + \frac{f(x)}{2x + \exp(x)}$$

gilt $\phi_3(x^*) = x^* + \frac{f(x^*)}{2x^* + \exp(x^*)} = x^*$ für $f(x^*) = 0$. Dies zeigt also " \Rightarrow ".

Umgekehrt folgt aus $\phi_3(x^*) = x^*$, dass $\frac{f(x^*)}{2x^* + \exp(x^*)} = 0$, und damit $f(x^*) = 0$, da für den betrachteten Bereich $\{x \in \mathbb{R} | x \ge 0\}$ der Nenner $2x + \exp(x) > 0$ ist.

Es handelt sich um das Newtonverfahren, da $f'(x) = -2x - \exp(x)$ gilt und ϕ_3 damit von der Bauart

$$\phi_3(x) = x - \frac{f(x)}{f'(x)}$$

ist.

c) Es gilt:

$$x_{n+1} = x_n - Q_n^{-1} f(x_n), \ Q_n = \frac{\Delta f}{\Delta x} = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

Damit folgt:

$$x_{n+1} = x_n - \frac{1}{1 - \frac{f(x_{n-1})}{f(x_n)}} (x_n - x_{n-1}) = x_n - \frac{x_n - x_{n-1}}{1 - \frac{2 - x_{n-1}^2 - \exp(x_{n-1})}{2 - x_n^2 - \exp(x_n)}}$$

d) Für die Konvergenzordnung gilt: $p = \frac{\log(e^{(k+1)}) - \log(e^{(k)})}{\log(e^{(k)}) - \log(e^{(k-1)})}$

$$p_1 = \frac{-23.3 - (-14.2)}{-14.2 - (-8.6)} = \frac{9.1}{5.6} \approx \frac{9}{6} = 1.5$$
$$p_2 = \frac{-21.8 - (-10.7)}{-10.7 - (-5.1)} = \frac{11.1}{5.6} \approx 2$$

e) Das Verfahren ist gemäss b) das Newtonverfahren. Dieses konvergiert lokal quadratisch (d. h. die Konvergenzordnung ist zwei), sofern $f'(x^*) \neq 0$, was zu überprüfen ist:

$$f'(x^*) = -2x^* - \exp(x^*) \approx -2\frac{1}{2} - 1.6 = -2.6 \neq 0.$$