Name:

D-ITET, D-MATL

Prüfung Numerische Methoden, Winter 2013

Dr. Lars Kielhorn

Wichtige Hinweise

- Prüfungsdauer: 90 Minuten.
- Nur begründete Resultate werden bewertet.
- Zugelassene Hilfsmittel: 10 A4-Seiten hand- oder computergeschriebene Notizen.
- Maximale Punktzahl: $40 = 8 + 12 + 8 + 12$ Punkte.

Viel Erfolg!

---

1. (8 Punkte)

Wir betrachten für ein gegebenes $\lambda \in \mathbb{R}$ und $y_0 \in \mathbb{R}$ das Anfangswertproblem

$$y'(t) = \lambda y(t), \quad t \geq 0, \tag{1}$$

$$y(0) = y_0. \tag{2}$$

a) **i)** Schreiben Sie die Gleichung (1) in der Form

$$y'(t) = f(t, y(t))$$

und geben sie insbesondere $f(t, y)$ an.

ii) Formulieren Sie das implizite Eulerverfahren, wobei y_i die Nährung an $y(t_i)$ sein soll. Dabei ist $t_i = hi$, $i = 0, 1, 2, \ldots$ und die Schrittweite $h > 0$.

iii) Vereinfachen Sie die Iterationsvorschrift soweit wie möglich und lösen Sie insbesondere nach y_{i+1} auf.

b) Ergänzen Sie den folgenden Matlab-Code, der das Anfangswertproblem mit dem impliziten Eulerverfahren löst.

```
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)=

end
```

Bitte wenden

c) Sei G die Stabilitätsfunktion, d. h. es gilt

$$y_{i+1} = G(h\lambda)y_i, \quad i = 0, 1, 2, \ldots$$

Bestimmen Sie $G(z)$ für das implizite Eulerverfahren ($z = h\lambda$).

d) Wir betrachten von nun an ein anderes Verfahren V. Dieses habe die Stabilitätsfunktion

$$G(z) = 1 + z$$

Das (reelle) Stabilitätsgebiet ist

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

Geben Sie S explizit für das Verfahren V an.

e) Was bedeutet dies nun für die Stabilität des Verfahrens V? Falls die Stabilität eine Beschränkung der Schrittweite erfordert, so geben Sie diese explizit an.
Hinweis: Für die Stabilität des Verfahrens betrachten wir hier nur $\lambda < 0$.

2. Ausgleichsrechnung (12 Punkte)

Gegeben seien Daten

$$(t_i, f_i), \quad i = 1, \ldots, m \quad (m > 2).$$

Durch diese Daten soll eine Funktion der Form

$$f(t) = \alpha \exp(-(t - \beta)^2)$$

gefittet werden, d. h. wir suchen $\alpha, \beta \in \mathbb{R}$ s. d. $f(t_i) \approx f_i$, $i = 1, \ldots, m$.

a) Stellen Sie ein geeignetes überbestimmtes lineares Gleichungssystem der Form

$$Ax = b \tag{3}$$

auf, aus dem mit Hilfe der linearen Ausgleichsrechnung α und β bestimmt werden können. Geben Sie insbesondere A, x und b an.
Hinweis: Verwenden sie $\log$ und führen Sie neue Unbekannte ein.

b) Ergänzen Sie die folgenden Matlab-Codes, die jeweils ein überbestimmtes lineares Gleichungssystem der Form (3) lösen.
Hinweis: Die Codes sind unabhängig davon, wie das Problem in **a)** konkret aussieht.
Zuerst soll das lineare Ausgleichsproblem mit dem Standard-Matlab-Befehl für solche Problem gelöst werden:

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

x= [ ]

Siehe nächstes Blatt!

Name:

Dann soll das Problem mit Hilfe der QR-Zerlegung gelöst werden:

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

x=

c) Nun soll mit Hilfe der nichtlinearen Ausgleichsrechnung für das gegebene Problem direkt nach α und β gelöst werden. Dazu definieren wir:

$$F_i = f(t_i) - f_i, \quad i = 1, \ldots, m$$

Um den Algorithmus anwenden zu können, benötigen wir die Jacobi-Matrix J von $F = (F_i)_{i=1}^m$. Bestimmen Sie also J.
Hinweis: Betrachten Sie $\frac{\partial}{\partial \alpha} F_i$ und $\frac{\partial}{\partial \beta} F_i$ und benutzen Sie, dass der Unbekanntenvektor $x = (\alpha, \beta)^T$ ist.

d) Ergänzen Sie die folgende Matlab-Funktion, die das Gauss-Newton-Verfahren auf dieses Ausgleichsproblem anwendet:

```
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 =

  %Korrektur anwenden

  x =

  if norm(s)<tol*norm(x)
    break;
```

Bitte wenden

```
  end
end
```

e) Worin besteht ein wesentlicher, konzeptioneller Unterschied zwischen der Lösung des linearen Ausgleichsproblems aus **a)** und des nichtlinearen Ausgleichsproblems **c)**?

3. Gauß-Quadratur (8 Punkte)

Gegeben sei die Funktion $f(x)$. Das Integral

$$I[f] = \int_0^1 f(x)\ \mathrm{d}x \tag{4}$$

soll durch eine Gauß-Quadratur $Q_n[f] \approx I[f]$ approximiert werden.

a) Für die Gauß-Quadratur gilt

$$Q_n[f] = \sum_{i=1}^{n} \omega_i f(\xi_i) \approx \int_{-1}^{1} f(x)\ \mathrm{d}x\ . \tag{5}$$

Geben Sie eine geeignete Variablentransformation an, so dass die Gauß-Quadratur auf (4) angewendet werden kann. Geben Sie das transformierte Integral $I[f]$ an.

b) Wir nehmen nun an, dass die Funktion $f(x)$ das Produkt eines Polynoms erster Ordnung $p(x)$ mit dem Logarithmus $\log(x)$ ist, d. h. $f(x) = p(x)\log(x)$. Konkret sei $p(x) = 2x + 1$. In Abb. 1 ist der Approximationsfehler

$$\varepsilon_n[f] := |Q_n[f] - I[f]| \tag{6}$$

für die Quadraturregeln $n = 1, \ldots, 30$ aufgetragen.

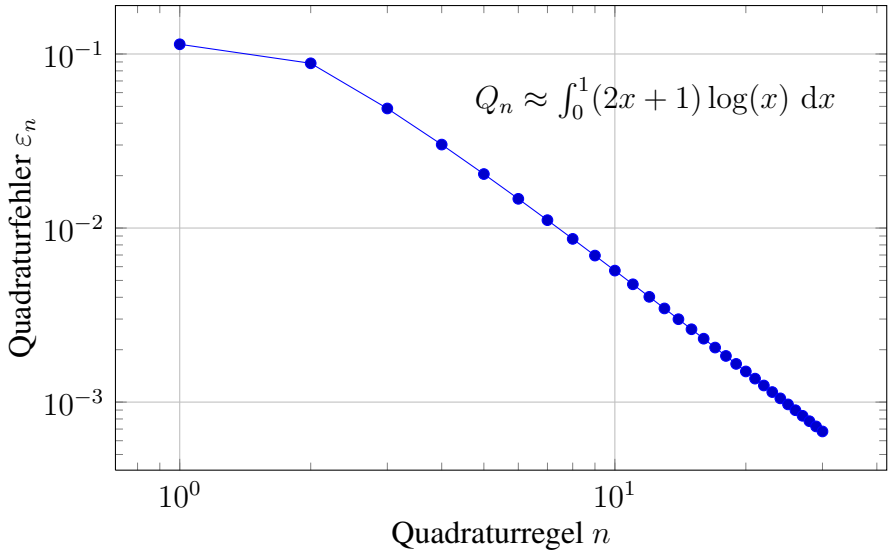

Abbildung 1: Quadraturfehler für die Quadraturregeln $n = 1, \ldots, 30$

Ist die Konvergenz exponentieller oder algebraischer Art? Weshalb beobachten wir hier diese Art von Konvergenz? *(Begründung!!!)*

Siehe nächstes Blatt!

Name:

c) Im Folgenden soll eine spezielle "Logarithmus-Quadratur" $\widehat{Q}_n$ der Form

$$\widehat{Q}_n[p] = \sum_{i=1}^{n} \hat{\omega}_i p(\hat{\xi}_i) \approx \int_0^1 p(x)\log(x)\ \mathrm{d}x = \widehat{I}[p] \tag{7}$$

entwickelt werden, die Polynome vom Grad 1 exakt integriert. Die Gewichte $\hat{\omega}_i$ und Stützstellen $\hat{\xi}_i$ können equivalent zur Gauß-Quadratur mit folgender Formel berechnet werden

$$\widehat{I}[x^k] = \widehat{Q}_n[x^k], \quad 0 \le k \le 2n-1, \tag{8}$$

d. h.

$$\int_0^1 x^k \log(x)\ \mathrm{d}x = \sum_{i=1}^{n} \hat{\omega}_i \hat{\xi}_i^k, \quad 0 \le k \le 2n-1\ . \tag{9}$$

i) Wie viele Gewichte und Stützstellen werden benötigt?

ii) Berechnen Sie die notwendigen Gewichte $\hat{\omega}_i$ und die Stützstellen $\hat{\xi}_i$ mittels Gl. (9).
Hinweis: Benutzen Sie partielle Integration.

d) Gegeben sei die "Logarithmus-Quadratur" $\widetilde{Q}$ der Form

$$\widetilde{Q}[p] = \sum_{i=1}^{2} \tilde{\omega}_i p(\tilde{\xi}_i) \approx \int_0^1 p(x)\log(x)\ \mathrm{d}x = \widehat{I}[p] \tag{10}$$

mit den Stützstellen $\tilde{\xi}_1 = 1/2$ und $\tilde{\xi}_2 = 1$ sowie den Gewichten $\tilde{\omega}_1 = -3/2$ und $\tilde{\omega}_2 = 1/2$. Werten Sie die Quadratur für das Integral

$$\widehat{I}[2x+1] = \int_0^1 (2x+1)\log(x)\ \mathrm{d}x \tag{11}$$

aus.

4. Fixpunktiteration (12 Punkte)

Gesucht sei eine Lösung der Gleichung

$$f(x) = 2 - x^2 - \exp(x) = 0\ . \tag{12}$$

Hierzu seien zwei Iterationsfunktionen ϕ_1, ϕ_2 wie folgt gegeben

$$\begin{aligned} \phi_1(x) &= \sqrt{2-\exp(x)} \\ \phi_2(x) &= \log(2-x^2)\ . \end{aligned} \tag{13}$$

a) Wir betrachten im Folgenden die Lösung x^* von Gleichung (12), die bei $\frac{1}{2}$ liegt, und nehmen an, dass $x^* = \frac{1}{2}$ exakt gilt, um die Rechnungen zu vereinfachen.
Bestimmen Sie, ob für die beiden gegebenen Iterationsfunktionen ϕ_1 und ϕ_2 die jeweilige Fixpunktiteration lokal konvergiert (d. h. bezüglich des Fixpunktes $x^* = \frac{1}{2}$ anziehend ist).
Hinweis: Nehmen Sie $\exp(\frac{1}{2}) = \sqrt{\mathrm{e}} \approx 1.6$ an.

Bitte wenden

b) Zu (12) sei eine dritte Fixpunktiteration gegeben

$$\phi_3(x) = x + \frac{2 - x^2 - \exp(x)}{2x + \exp(x)} \ . \tag{14}$$

Zeigen Sie, dass es sich bei ϕ_3 um eine konsistente Fixpunktiteration handelt. Um welches Iterationsverfahren handelt es sich hier *(Begründung!)*?
Hinweis: Wir betrachten hier nur den Bereich $\{x \in \mathbb{R} | x \geq 0\}$.

c) Die Gl. (12) soll mit dem Sekantenverfahren gelöst werden. Geben Sie die zugehörige Iterationsvorschrift explizit an.

d) Für die gesuchte Nullstelle von Gl. (12) betrachten wir zwei verschiedene iterative Verfahren V_1 und V_2. Der absolute Fehler $e^{(k)}$ in jedem Iterationsschritt k ist in der Tabelle 1 dargestellt.

	V_1		V_2	
k	$e_1^{(k)}$	$\log(e_1^{(k)})$	$e_2^{(k)}$	$\log(e_2^{(k)})$
0	1.7E-1	-1.8	5.4E-1	-0.6
1	4.5E-2	-3.1	4.6E-1	-0.8
2	5.7E-3	-5.2	9.9E-2	-2.3
3	1.8E-4	-8.6	5.9E-3	-5.1
4	6.7E-7	-14.2	2.3E-5	-10.7
5	7.9E-11	-23.3	3.5E-10	-21.8

Tabelle 1: Absoluter Fehler und dessen Logarithmus für die beiden Verfahren V_1, V_2.

Schätzen Sie die jeweiligen Konvergenzordnungen p_1 und p_2 der beiden Verfahren V_1, V_2.

e) Welche Konvergenzordnung erwarten Sie für das Verfahren aus **b)**, wenn der Startwert genügend nahe bei x^* ist (Begründung!)?