

Prüfung Numerische Methoden**Wichtige Hinweise**

- Die Prüfung dauert 90 Minuten.
- Erlaubte Hilfsmittel: 5 A4-Blätter doppelseitig (=10 Seiten) eigenhändig und handschriftlich verfasste Zusammenfassung, nicht ausgedruckt, nicht kopiert. Sonst keine Hilfsmittel zugelassen.
- Begründen Sie jeweils Ihre Aussagen. Unbegründete Lösungen (außer bei Multiple-Choice-Aufgaben falls nicht explizit gefordert) werden nicht akzeptiert!

Name		Note
Vorname		
Studiengang		
Leginummer		
Prüfung	Numerische Methoden	
Datum	27.01.2017	

1	2	3	4	5	6	Bonuspunkte	Punkte
6	8	7	6	8	8		43

- Legen Sie Ihre Legi auf den Tisch. Schalten Sie Ihr Handy aus.
- Lösen Sie Aufgaben 1, 2 und 6 auf dem Angabenblatt.
Lösen Sie Aufgaben 3, 4 und 5 auf Extrablättern. Beginnen Sie hierfür für jede Aufgabe eine neue Seite, und schreiben Sie Ihren Namen und Ihre Leginummer auf alle Blätter.
- Schreiben Sie nicht mit Bleistift. Verwenden Sie einen Stift mit blauer oder schwarzer Farbe (keinesfalls rot oder grün).
- Versuchen Sie Ihren Lösungsweg möglichst klar darzustellen und arbeiten Sie sorgfältig!
- **Schauen Sie das Prüfungsblatt erst an, wenn der Assistent das Signal dazu gibt!**

Viel Erfolg!

1. Kurze Fragen

(a) Gegeben sei das Anfangswertproblem 3. Ordnung

$$\ddot{y}(t) = (\dot{y}(t) - y(t))^2 + 3 \sin(t)y(t),$$

mit Anfangswerten

$$y(0) = a, \dot{y}(0) = b, \ddot{y}(0) = c.$$

Schreiben Sie es in ein Anfangswertproblem 1. Ordnung um.

(b) Sie wollen die Gleichung

$$\frac{\partial u}{\partial t}(x, t) - \frac{\partial u}{\partial x}(x, t) = 0$$

mit finiten Differenzen im Ort diskretisieren und das daraus resultierende System von ODEs mit dem expliziten Eulerverfahren in der Zeit lösen. Welchen Differenzenquotienten verwenden Sie, um $\frac{\partial u}{\partial x}(x_i, t)$ zu approximieren, wobei x_i ein innerer Gitterpunkt eines äquidistanten Ortsgitters mit Gitterweite Δx ist?

$\frac{u(x_i + \Delta x, t) - u(x_i - \Delta x, t)}{2\Delta x}$

$\frac{u(x_i + \Delta x, t) - u(x_i, t)}{\Delta x}$

$\frac{u(x_i, t) - u(x_i - \Delta x, t)}{\Delta x}$

Keiner der genannten Differenzenquotienten führt zu einem stabilen Verfahren

Siehe nächstes Blatt!

(c) (i) Wir betrachten die folgenden Verfahren

- Explizites Eulerverfahren, explizite Trapezregel, RK4
- Implizites Eulerverfahren, implizite Trapezregel

Welches Verfahren würden Sie verwenden um

- Die eindimensionale Wärmeleitungsgleichung, die im Ort mit zentralen Differenzen für eine Gitterweite Δx diskretisiert wurde, unter Verwendung der Schrittweite $\Delta t = \Delta x$ *effizient* und stabil zu lösen?

Antwort:

(ii) Wir betrachten die folgenden Verfahren

- Explizites Eulerverfahren, explizite Trapezregel, RK4
- Implizites Eulerverfahren, implizite Trapezregel, BDF2, BDF3, BDF4

Welches Verfahren würden Sie verwenden um

- Das Problem

$$\begin{cases} \dot{\mathbf{y}}(t) = \begin{pmatrix} -10 & 20 \\ 0 & -500 \end{pmatrix} \mathbf{y}(t), & t \in (0, T), \\ \mathbf{y}(0) = \mathbf{y}_0, \end{cases}$$

effizient mit hoher Ordnung und stabil mit $\Delta t = 0.1$ zu lösen?

Antwort:

- Das Problem

$$\begin{cases} \dot{\mathbf{y}}(t) = \begin{pmatrix} -1 & 80 & 3i \\ 0 & -10 & 12 \\ 0 & 0 & -10 + i \end{pmatrix} \mathbf{y}(t), & t \in (0, T), \\ \mathbf{y}(0) = \mathbf{y}_0. \end{cases}$$

effizient mit hoher Ordnung und stabil mit $\Delta t = 0.1$ zu lösen?

Antwort:

Bitte wenden!

2. Quadratur

a) Berechnen Sie das Integral

$$\int_{-1}^0 (x^4 - x^2 + 2) dx$$

mithilfe der Simpson-Regel.

b) Gegeben sei die folgende MATLAB-Funktion.

```
1 function val = welche_quad(f, a, b, N)
2 val = f(a) + f(b);
3 for ii=1:N-1
4     val = val + 2*f(a+ii*(b-a)/N);
5 end
6 val = ((b-a)/(2*N))*val;
```

Welche der folgenden Quadraturformel wird im obigen Code implementiert?

- Summierte Mittelpunkstregel
- Summierte Gauss Quadraturformel mit 2 Stützstellen pro Intervall
- Summierte Trapezregel
- Summierte Simpson-Regel

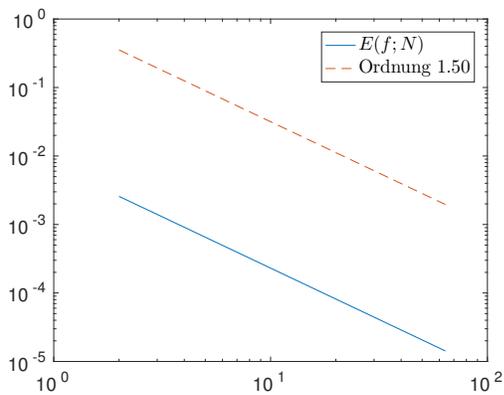
Siehe nächstes Blatt!

c) Für diese Aufgabe stehen die folgenden Funktionen $f_i : \mathbb{R} \rightarrow \mathbb{R}$ zur Auswahl:

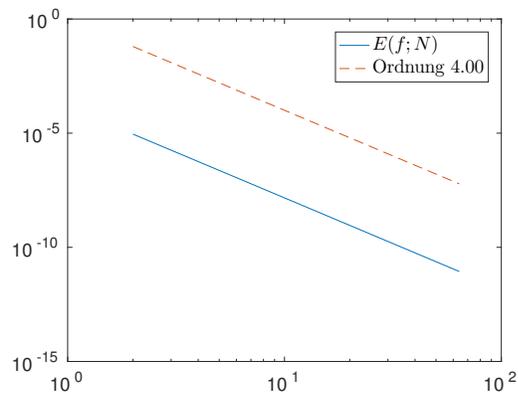
- $f_1(x) = x^5 - 1$
- $f_2(x) = x^2 - 3x + 2$
- $f_3(x) = e^{-x}$
- $f_4(x) = \sqrt{x}$

Geben Sie für alle Graphiken dieser Aufgabe an, welche der Funktionen f_1, \dots, f_4 zu diesen Graphiken geführt hat. (Pro Bild kann es eine oder mehrere Möglichkeiten geben. Wird mindestens eine falsche Funktion für ein Bild angegeben, so gibt es für dieses Bild 0 Punkte. Nicht alle Funktionen f_1, \dots, f_4 müssen in dieser Aufgabe verwendet werden. Die Graphiken verwenden teilweise eine log-Skalierung der x - und/oder y -Achse.)

(i) Für eine gegebene Funktion $f : [0, 1] \rightarrow \mathbb{R}$ bezeichnen wir mit $Q_N^{G,S}(f; 0, 1)$ die *zusammengesetzte Simpsonregel* mit N Subintervallen der Länge $\frac{1}{N}$. Die folgende Graphiken zeigen das Verhalten von $E(f; N) := \left| \int_0^1 f(x) dx - Q_N^{G,S}(f; 0, 1) \right|$ bezüglich N für verschiedene Funktionen.



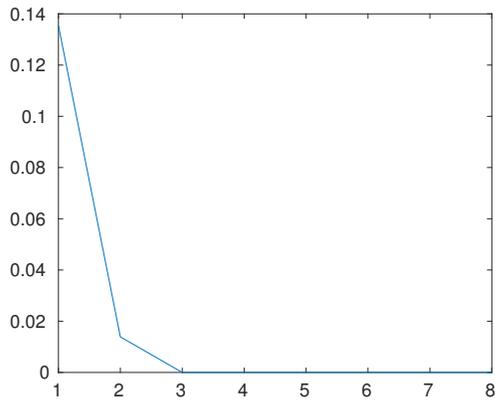
Funktion/en:



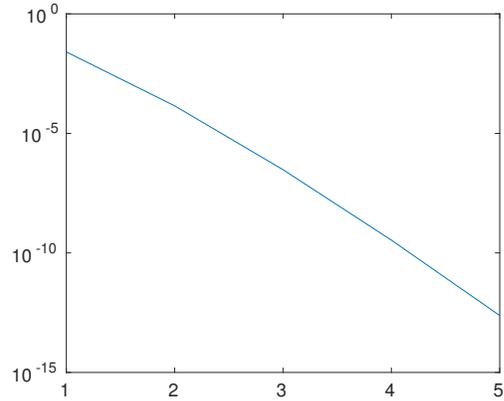
Funktion/en:

Bitte wenden!

(ii) Für eine gegebene Funktion $f : [0, 1] \rightarrow \mathbb{R}$ bezeichnen wir mit $Q^{G,n}(f; 0, 1)$ die *Gauss Quadraturformel mit n Stützstellen*, angewendet auf das Intervall $[0, 1]$. Die folgende Graphiken zeigen das Verhalten von $E(f; n) := \left| \int_0^1 f(x) dx - Q^{G,n}(f; 0, 1) \right|$ bezüglich n für verschiedene Funktionen.



Funktion/en:



Funktion/en:

(Eine Begründung ist nicht notwendig.)

Siehe nächstes Blatt!

3. Wir betrachten das skalare Anfangswertproblem

$$\dot{y}(t) = f(t, y(t)), \quad y(0) = y_0,$$

und das Runge-Kutta Einschrittverfahren, das durch das folgende Butcher-Schema gegeben ist:

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{2}{3} & \frac{2}{3} & 0 & 0 \\ \hline & 0 & \frac{1}{2} & \frac{1}{2} \end{array}$$

Zeigen Sie, dass das Verfahren genau die Konsistenzordnung 2 (und nicht die Ordnung 3) besitzt.

4. Wir betrachten das skalare Anfangswertproblem

$$\dot{y}(t) = f(t, y(t)), \quad y(0) = y_0.$$

a) Wir betrachten für $\alpha \in [0, 1]$ das Verfahren, das durch das folgende Butcher-Schema gegeben ist

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \alpha & \alpha & 0 \\ \hline & 1 - \alpha & \alpha \end{array}$$

Berechnen Sie die Stabilitätsfunktion des Verfahrens in Abhängigkeit des Parameters α .

b) Zeigen Sie, dass das implizite Eulerverfahren L-stabil ist.

Bitte wenden!

5. Wir wollen das folgende Optimierungsproblem lösen

$$\min_{(x_1, x_2) \in \mathbb{R}^2} f(x_1, x_2) := \min_{(x_1, x_2) \in \mathbb{R}^2} (x_1^2 + 2x_2^2 - 2x_2 + 3x_1 + 7). \quad (1)$$

a) Finden Sie eine Funktion $\mathbf{F} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ so dass

$$\mathbf{F}(\mathbf{x}^*) = \mathbf{0} \Leftrightarrow \mathbf{x}^* = \arg \min_{(x_1, x_2) \in \mathbb{R}^2} f(x_1, x_2).$$

(Das heißt: Schreiben Sie das Optimierungsproblem in ein Nullstellenproblem um und begründen Sie, dass die Lösung des Nullstellenproblems auch wirklich eine Lösung des Optimierungsproblems ist.)

b) Wir wollen \mathbf{x}^* mithilfe des Newton Verfahrens approximieren. Führen Sie einen Schritt des Newton Verfahrens für \mathbf{F} aus Teilaufgabe a) und den Startwert $\mathbf{x}^{(0)} = (0, 0)^T$ aus.
Hinweis: Falls, Sie a) nicht lösen konnten, verwenden Sie

$$\mathbf{F}(\mathbf{x}) = \begin{pmatrix} 9x_1 + 1 \\ 5x_2 - 7 \end{pmatrix}.$$

- c) Begründen Sie, dass das Newton Verfahren für die betrachtete Funktion \mathbf{F} in einem Schritt konvergiert. (Somit ist $\mathbf{x}^{(1)}$ aus Teil b) die Lösung des Nullstellenproblems aus Teilaufgabe a).)
- d) Lösen Sie das Optimierungsproblem (1) numerisch, indem Sie das aus a) resultierende Nullstellenproblem mit dem Newton Verfahren lösen. Vervollständigen Sie dazu den folgenden Code, der speziell auf das Optimierungsproblem in dieser Aufgabe ausgelegt ist.

Argumente für die Funktion `newton_minimierung`:

- `f`: Funktion für das Minimierungsproblem aus (1)
- `df, d2f`: Gradient und Hessematrix von `f`
- `x0`: Startwert des Algorithmus

```
1 function x = newton_minimierung(f, df, d2f, x0)
2 % Startwert
3 x = ...
4
5 % Newton update
6 delta_x = ...
7
8 % Neue Iterierte
9 x = ...
```

Siehe nächstes Blatt!

6. Wir wollen das Integral

$$\int_a^b f(x)dx$$

mithilfe eines adaptiven Quadraturverfahrens, basierend auf einem Quadraturverfahren der Ordnung p , numerisch berechnen. Dafür müssen wir einen geeigneten Fehlerschätzer für den lokalen Fehler auf dem Teilintervall $[x_j, x_{j+1}]$ definieren. Diesen leiten wir folgendermaßen her: für eine gegebene Quadraturformel Q der Ordnung p , berechnen wir

- einerseits $Q^1(f; x_i, x_{i+1}) := Q(f; x_i, x_{i+1})$
- und andererseits $Q^2(f; x_i, x_{i+1}) := Q\left(f; x_i, \frac{x_i+x_{i+1}}{2}\right) + Q\left(f; \frac{x_i+x_{i+1}}{2}, x_{i+1}\right)$.

Daraus ergibt sich dann für das Intervall $[x_i, x_{i+1}]$ der Fehlerschätzer

$$EST_{[x_i, x_{i+1}]} := \left| \frac{Q^1(f; x_i, x_{i+1}) - Q^2(f; x_i, x_{i+1})}{1 - 2^{-p}} \right|.$$

Als Quadraturformel Q verwenden wir die Trapezregel. Wir haben versucht, dies in der folgenden MATLAB-Funktion umzusetzen. Leider erreicht h für alle Funktionen den unteren Grenzwert h_{min} . Wir vermuten, dass sich einige Fehler in unseren Code eingeschlichen haben. *Finden* Sie diese Fehler und *korrigieren* Sie sie.

Hinweise:

- Die Anzahl der Fehler liegt zwischen 5 und 10.
- Korrigieren Sie fälschlicherweise korrekte Stellen, führt dies zu Punktabzug.
- Es ist *NICHT* erlaubt, Zeilen hinzuzufügen, zu streichen oder zu vertauschen.

Erklärungen zum Code:

- `f`: Funktion, die integriert wird
- `a,b`: Intervall für das Integral
- `tol`: Toleranz
- `hmin`: minimale Schrittweite
- `Trapez_quad(f, a, b)`: berechnet eine Approximation des Integrals $\int_a^b f(x)dx$ mithilfe der Trapezregel

Bitte wenden!

```

1 function apprintegral = adaptiveQuadratur(f,a,b,tol,hmin)
2 p = 3;
3 xstart = a;
4 xend = b;
5 htoosmall = false;           % Gehe aus while-loop fuer h zu klein
6 apprintegral = 0;
7
8 while xstart < b
9     ytemp1 = Trapez_quad(f,xstart,xend);
10    yhalf = Trapez_quad(f,xstart,(xend+xstart)/2);
11    ytemp2 = Trapez_quad(f,(xend+xstart)/2,xend);
12    est = abs(ytemp1 - ytemp2);
13
14    if est > tol
15        xend = (xend+xstart)/2;
16        if xend-xstart < hmin
17            htoosmall = true;
18            break
19        end
20        ytemp1 = Trapez_quad(f,xstart,xend);
21        yhalf = Trapez_quad(f,xstart,(xend+xstart)/2);
22        ytemp2 = Trapez_quad(f,(xend+xstart)/2,xend);
23        est = abs(ytemp1 - ytemp2);
24    end
25
26    apprintegral = ytemp1;
27    xstart = xend;
28    xend = (xend+xstart)/2;
29    if (xend-xstart < hmin) && (xstart < b-1e-15)
30        htoosmall = true;
31    end
32
33    if htoosmall
34        apprintegral = NaN;
35        fprintf('h ist zu klein \n')
36        break
37    end
38 end

```