Dr. S. May D-ITET, D-MATL Winter 2016

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 der Multiple-Choice-Aufgabe falls nicht explizit gefordert) werden nicht akzeptiert!

Name		Note
Vorname		
Studiengang		
Leginummer		
Prüfung	Numerische Methoden	
Datum	15.02.2016	

1	2	3	4	5	Bonuspunkte	Punkte
11.5	7	11	6.5	11		47

- Legen Sie Ihre Legi auf den Tisch. Schalten Sie Ihr Handy aus.
- Beginnen Sie jede Aufgabe auf einer neuen Seite, und schreiben Sie Ihren Namen und Ihre Leginummer auf alle Blätter.
- Bitte nicht mit Bleistift schreiben. Verwenden Sie einen Stift mit blauer oder schwarzer Farbe (keinesfalls rot oder grün).
- Versuchen Sie Ihren Lösungsweg möglichst klar darzustellen und arbeiten Sie sorgfältig!
- Beantworten Sie Aufgaben 1,4,5 auf diesem Papier,
  und Aufgaben 2,3 auf dem von Ihnen mitgebrachten Papier.
- **Schauen Sie das Prüfungsblatt erst an, wenn der Assistent das Signal dazu gibt!**

Viel Erfolg!

1. Kurze Fragen

(a) Wir betrachten das skalare Anfangswertproblem

$$\dot{y}(t) = -1000(y(t) - \sin(t)) + \cos(t), \quad y(0) = 0,$$

und lösen es mit der expliziten und impliziten Trapezregel für verschiedene Schrittweiten h. Wir erhalten die folgende Ergebnisse:

TAB 1

h	Fehler
2.0e-02	1.5309e+111
1.0e-02	1.8984e+159
5.0e-03	3.8261e+183
2.50e-03	5.4865e+81
1.25e-03	6.7384e-04
6.25e-04	3.3741e-04

TAB 2

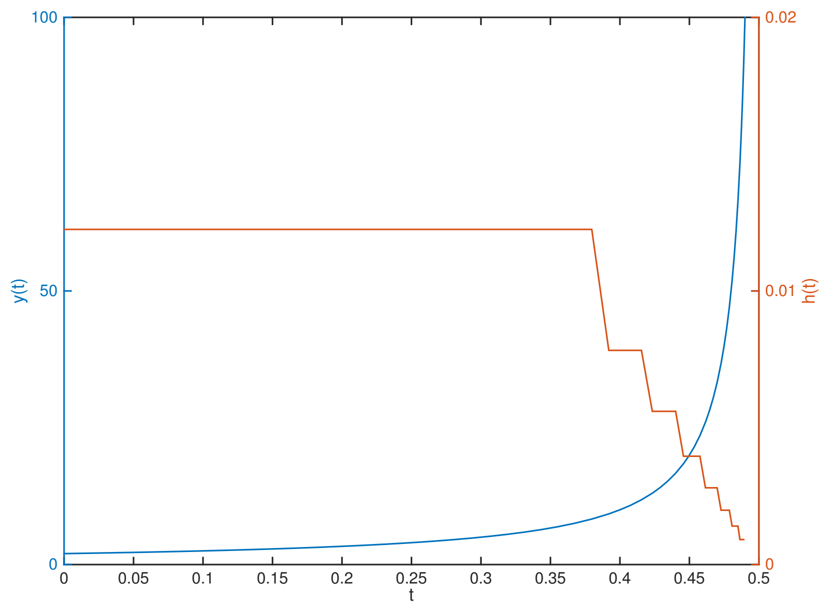
h	Fehler
2.0e-02	1.0636e-02
1.0e-02	5.3608e-03
5.0e-03	2.6909e-03
2.50e-03	1.3481e-03
1.25e-03	6.7472e-04
6.25e-04	3.3752e-04

Leider gerieten unsere Aufzeichnungen durch einen Luftstoß in Unordnung. Ordnen Sie die 2 Tabellen richtig der expliziten und impliziten Trapezregel zu. Entsprechen die Ergebnisse in den 2 Tabellen dem, was Sie aus der Theorie erwarten? *Begründen* Sie jeweils kurz Ihre Meinung.

(b) Welches/welche der folgenden Verfahren ist/sind A-stabil?

☐ Explizite Trapezregel
☐ Implizite Mittelpunktsregel
☐ RK4
☐ BDF2
☐ BDF4

(c) Ein skalares AWP wird mit dem MATLAB-Löser `ode45` gelöst. Im Bild sind die approximative Lösung $y(t)$ in blau und die verwendeten Schrittweiten $h(t)$ in orange dargestellt. Interpretieren Sie das Verhalten von $h(t)$ in Abhängigkeit von $y(t)$.



(d) Sie haben eine neue Runge-Kutta Methode der Ordnung 3 zur Lösung des Anfangswertproblems

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

implementiert. Was ist eine gute Möglichkeit, um zu überprüfen ob der Code korrekt ist?

Bitte wenden!

(e) Gegeben sei das Anfangswertproblem 2. Ordnung

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

mit Anfangswerten

$$y(0) = y_0,\ \dot{y}(0) = y_1.$$

Schreiben Sie es in ein Anfangswertproblem 1. Ordnung um.

(f) Gegeben sei das Anfangswertproblem

$$\begin{cases} \dot{y}(t) = & -4t^2 y(t), \\ y(1/2) = & 2. \end{cases}$$

Führen Sie einen Schritt der impliziten Trapezregel mit $h = 1/2$ aus.

Bitte wenden!

2. Quadratur mit Newton-Cotes Formeln

(a) Es bezeichne $Q^T(f;a,b)$ die Trapezregel zur Approximation von $\int_a^b f(x)\,dx$. Zeigen Sie, dass die Trapezregel *mindestens* die Ordnung 2 besitzt.

(b) Es bezeichne $Q_N^T(f;a,b)$ die zusammengesetzte Trapezregel mit N Intervallen gleicher Länge $h = \frac{b-a}{N}$ zur Approximation von $\int_a^b f(x)\,dx$. Werten Sie $Q_3^T(g;0,1)$ mit $g(x) = x^3$ aus.

(c) Es sei

$$E(h) = \left| \int_0^1 g(x)\,dx - Q_N^T(g;0,1) \right|,$$

mit g aus (b) und $h = \frac{1}{N}$.

(i) Welches Verhalten/Konvergenzordnung erwarten wir für $E(h)$ für $h \to 0$?

(ii) Welches Verhalten/Konvergenzordnung erwarten wir für $E(h)$ für $h \to 0$, falls wir statt der zusammengesetzten Trapezregel die zusammengesetzte Simpsonregel verwenden?

Geben Sie jeweils eine kurze Begründung.

Siehe nächstes Blatt!

3. Für $\gamma \in [0,1]$, sei das Runge-Kutta-Einschrittverfahren mit dem folgenden Butcher-Schema gegeben:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \gamma & \gamma & 0 \\ \hline & 1-\frac{1}{2\gamma} & \frac{1}{2\gamma} \end{array} \tag{1}$$

(a) Schreiben Sie für das Anfangswertproblem

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

das in (1) gegebene Verfahren in die Stufenform eines Runge-Kutta-Verfahrens um. Verwenden Sie hierbei den Schritt von y_n zu y_{n+1}.

(b) Für welche Werte von γ ist das Verfahren implizit?

(c) Zeigen Sie, dass das Verfahren für $\gamma = \frac{1}{2}$ die Konsistenzordnung 2 besitzt. Führen Sie dazu eine Analyse mittels Taylorentwicklung durch!

Hinweis: Sie können annehmen, dass das Verfahren autonomisierungsinvariant ist.

(d) Berechnen Sie das Stabilitätsintervall des Verfahrens für $\gamma = \frac{1}{2}$.

4. Newton Methode

Wir wollen die Lösung des nichtlinearen Gleichungssystems

$$\begin{cases} \exp(x_1^2 + x_2^2) = 1, \\ \exp(x_1^2 - x_2^2) = 1, \end{cases} \tag{2}$$

mithilfe des Newton-Verfahrens berechnen.

(a) Geben Sie die Funktion $\mathbf{F}(x_1, x_2)$ und Jacobi-Matrix $\mathbf{DF}(x_1, x_2)$ an, die im Newton-Verfahren verwendet werden.

Siehe nächstes Blatt!

(b) Ergänzen Sie die MATLAB-Funktion `x = newton2D`, die eine approximative Lösung des nichtlinearen Gleichungssystems (2) mit dem Newton-Verfahren berechnet.

```
function x = newton2D

%Initialisierung
x = 0.1*[1;1];

F = ...


DF = ...







%Toleranz
tol = 1e-8;
Nmax = 100;

for i=1:Nmax

    x_old = ...

    %Update
    x = ...

    %Abbruchkriterium
    if norm(DF(x_old)\F(x_old)) < tol;
        break
    end
end
end
```

Bitte wenden!

5. Wir wollen das Anfangswertproblem

$$\begin{cases} \dot{y}(t) = & f(y(t)), \\ y(a) = & y_0, \end{cases}$$

mit $y(t) : \mathbb{R} \to \mathbb{R}^n$ mithilfe eines adaptiven Einschrittverfahrens lösen. Um den Fehler zu schätzen, vergleichen wir die Ergebnisse für je einen Schritt mit der *expliziten Trapezregel* und dem *klassischen RK4 Verfahren*. Als Vorschlag für die neue Schrittweite verwenden wir die optimale Schrittweite für den gerade akzeptierten Schritt, den Ideen aus der Vorlesung folgend.

Wir haben versucht, dies in der folgenden MATLAB-Funktion umzusetzen. Die Funktionen `ynew = Explizit_Trapez(f,yn,tn,h)` und `ynew= RKF4(f,yn,tn,h)` berechnen einen Schritt der expliziten Trapezregel und des klassischen RK4 Verfahren mit Schrittweite h mit `ynew` und `yn` Zeilenvektoren.

Leider erhalten wir schlechte Ergebnisse und 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 12.
- Korrigieren Sie fälschlicherweise korrekte Stellen, führt dies zu Punktabzug.
- Es ist *NICHT* erlaubt, Zeilen hinzuzufügen oder Zeilen zu vertauschen.
- Es ist erlaubt, Zeilen zu streichen oder zu ersetzen.

Argumente für Funktion `adaptive_fehler`:

- `f`: rechte Seite der ODE
- `a`,`b`: ODE soll für Zeitintervall [a,b] gelöst werden
- `y0`: Zeilenvektor mit Startwerten für y
- `h`: Startwert für Schrittweite
- `tol`: Toleranz
- `hmin`: minimale Schrittweite
- `maxstep`: maximale Anzahl an Schritten

Siehe nächstes Blatt!

```
function [y tt] = adaptive_fehler(f,a,b,y0,h,tol,hmin,maxstep)
p = 4;      % Konvergenzordnung
t = a;      % t: aktuelle Zeit
tt(1) = a;
y(1,:) = y0;
i = 1;    % Zaehler fuer akzeptierte Schritte

for nstep = 1:maxstep
    ytemp1 = Explizit_Trapez(f,y(nstep,:),t,h);
    yhalf  = Explizit_Trapez(f,y,t,h/2);
    ytemp2 = Explizit_Trapez(f,yhalf,t+h/2,h/2);
    e = norm(ytemp2 - ytemp1);

    if e <= tol
        t = t + h;
        i = i + 1;    % i: Zaehler fuer akzeptierte Schritte
        tt(i) = t;
        y(i,:) = ytemp1;

        h = min(hmin, min(2*h,0.8*h*(tol/e)^(1/(p+1))));
        if (t + h > b - 1.E-12)
           h = b - t;
        end;

        if (t > b - 1.E-12)
            fprintf(1,'Schrittweitensteuerung erfolgreich.\n');
            break;
        end
    else
        h = 2*h;
    end;
    if (h < hmin - 1.E-12)
        fprintf(1,'h zu klein.\n'); break;
    end
end;

if (nstep >= maxstep)
    fprintf(1,'Haben maximale Anzahl an Schritten erreicht.\n');
    fprintf(1,'Brechen Rechnung jetzt ab.\n');
    fprintf(1,'Haben die Zeit T=%f erreicht.\n',b);
end
```