Prof. Dr. Stefan Funken
Dipl.- Math. Katharina Becker-Steinberger
Dipl.-Math. oec. Klaus Stolle
Institut für Numerische Mathematik
Universität Ulm

WS 2013/2014
Lösungsblatt 2
08.11.2013

Angewandte Numerik 2

Dieses Übungsblatt hat aufgrund des Feiertags Allerheiligen eine Bearbeitungszeit von zwei Wochen. Nicht alle Aufgaben können mit dem Inhalt der Vorlesung vom 23.10.2013 bearbeitet werden.

Aufgabe 4 *(Programmieraufgabe, Euler-Verfahren)* (8 Punkte)

Bestimmen Sie mit Matlab für das Anfangswertproblem

$$y'(t) = y(t) + e^t, \quad y(0) = 1$$

auf dem Intervall $[0, 0.2]$ eine Näherungslösung nach dem expliziten Euler-Verfahren

$$y_{k+1} = y_k + hf(t_k, y_k).$$

Wählen Sie als Schrittweite zunächst $h = \frac{1}{20}$ und lösen Sie ebenfalls für $h = \frac{1}{40}$. Vergleichen Sie die Ergebnisse mit der exakten Lösung $y(t) = (t + 1)e^t$.

Lösung:

```
function y = expl_Euler(a, b, N, y_0,f,y_exakt)
% explizites Euler-Verfahren mit gleichmaessiger Schrittweit zur
% Approximation einer ODE
%
% Input:  N        Anzahl der Schritte
%         a        Anfang des Intervalls I=[a,b]
%         b        Ende des Intervalls
%         y_0      Anfangswert
%         f        rechte Seite der ODE
%         y_exakt  exakte Loesung der ODE
% Output: y        Approximation der Losung
%          Tabelle

y = zeros(length(y_0), N+1);
y(:,1)= y_0(:)';
h=(b-a)/N;                              %Schrittweite
for k=1:N
   y(:,k+1)= y(:,k)+h*f(a+(k-1)*h, y(:,k)); % expliziter Euler
   err(:,k+1)= abs(y(:,k+1)-y_exakt(a+k*h, y(:,k+1)));      % Fehler
end

% Tabellarische Ausgabe
fprintf('\n\nSchrittweite h=%1.4f', h);
for k=1:N+1
    fprintf('\n k=%1f, t_k=%1.5f, y(t_k)=%1.5e, y_k=%1.5e, e_k=%1.5e',...
        k, a+(k-1)*h,y_exakt(a+(k-1)*h, y(:,k)),y(:,k),err(:,k) );
```

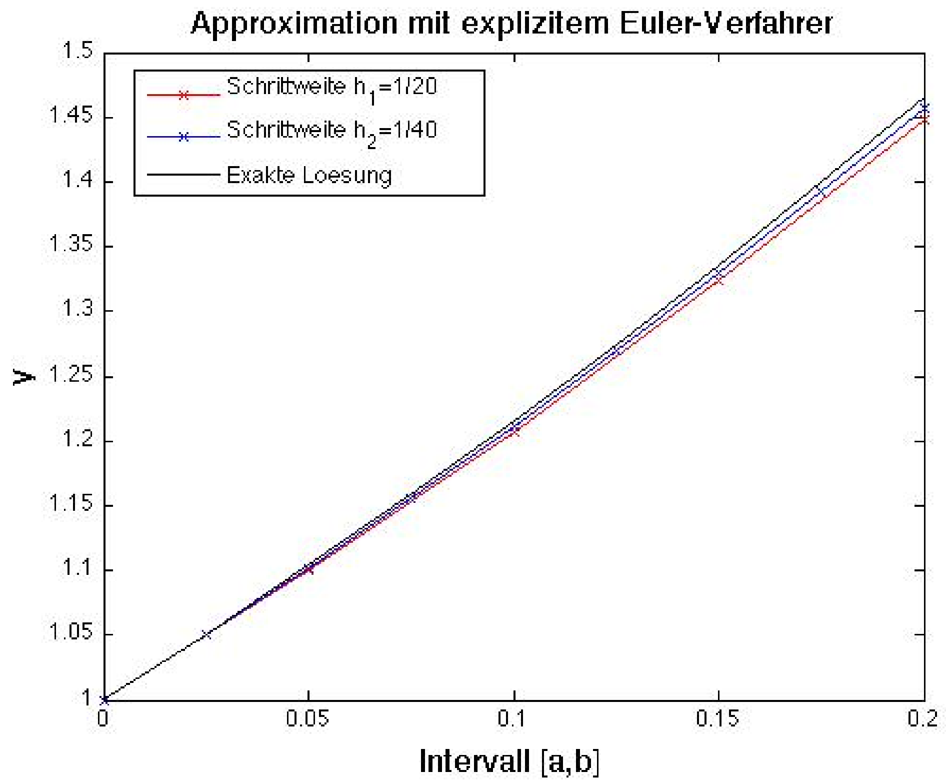
```
end

end
```

```
% Aufgabe 4: Test explizites Euler-Verfahren

clear all
close all

fprintf('\n*****************************************\n');
fprintf('\nAufgabe 4: Test explizites Euler-Verfahren\n');
fprintf('\n*****************************************\n');


y_0 = 1;
a = 0;
b = 0.2;
N1 = 4;
N2 = 8;
f=@(t,y) y+exp(t);
y_exakt = @(t,y) (t+1)*exp(t);
y1= expl_Euler(a, b, N1, y_0,f,y_exakt);
y2= expl_Euler(a, b, N2, y_0,f,y_exakt);

% Grafische Ausgabe
g = figure();
x1 = linspace(a,b, N1+1);
plot(x1, y1, 'x-r')
title('Approximation mit explizitem Euler-Verfahren','FontSize', 14)
xlabel('Intervall [a,b]', 'FontSize', 14)
ylabel('y', 'FontSize', 14)
hold on;
x2 = linspace(a,b, N2+1);
plot(x2, y2, 'x-b')
plot(x2, (x2+1).*exp(x2), '-k' )
legend ('Schrittweite h_1=1/20', 'Schrittweite h_2=1/40', ...
'Exakte Loesung', 'Location', 'Best')
```

```

*****************************************

Aufgabe 4: Test explizites Euler-Verfahren

*****************************************


Schrittweite h=0.0500
 k=1.000000, t_k=0.00000, y(t_k)=1.00000e+00, y_k=1.00000e+00, e_k=0.00000e+00
 k=2.000000, t_k=0.05000, y(t_k)=1.10383e+00, y_k=1.10000e+00, e_k=3.83465e-03
 k=3.000000, t_k=0.10000, y(t_k)=1.21569e+00, y_k=1.20756e+00, e_k=8.12446e-03
 k=4.000000, t_k=0.15000, y(t_k)=1.33611e+00, y_k=1.32320e+00, e_k=1.29091e-02
 k=5.000000, t_k=0.20000, y(t_k)=1.46568e+00, y_k=1.44745e+00, e_k=1.82313e-02

Schrittweite h=0.0250
```

```
k=1.000000, t_k=0.00000, y(t_k)=1.00000e+00, y_k=1.00000e+00, e_k=0.00000e+00
k=2.000000, t_k=0.02500, y(t_k)=1.05095e+00, y_k=1.05000e+00, e_k=9.47999e-04
k=3.000000, t_k=0.05000, y(t_k)=1.10383e+00, y_k=1.10188e+00, e_k=1.95177e-03
k=4.000000, t_k=0.07500, y(t_k)=1.15873e+00, y_k=1.15571e+00, e_k=3.01373e-03
k=5.000000, t_k=0.10000, y(t_k)=1.21569e+00, y_k=1.21155e+00, e_k=4.13639e-03
k=6.000000, t_k=0.12500, y(t_k)=1.27479e+00, y_k=1.26947e+00, e_k=5.32232e-03
k=7.000000, t_k=0.15000, y(t_k)=1.33611e+00, y_k=1.32954e+00, e_k=6.57424e-03
k=8.000000, t_k=0.17500, y(t_k)=1.39971e+00, y_k=1.39182e+00, e_k=7.89493e-03
k=9.000000, t_k=0.20000, y(t_k)=1.46568e+00, y_k=1.45640e+00, e_k=9.28729e-03
```

Man beachte, dass sich durch die Halbierung der Schrittweite, der Fehler $e_k = y(t_k) - y_k$ ungefähr halbiert.



Aufgabe 5 *(Programmieraufgabe, Einschrittverfahren: explizites Euler-Verfahren, Verfahren von Heun)*
(10 Punkte)

Schreiben Sie ein Matlab-Programm zur numerischen Lösung der Anfangswertaufgabe

$$y' = -200ty^2, \quad y(-1) = \frac{1}{101}$$

Verwenden Sie dazu die folgenden Einschrittverfahren mit konstanter Schrittweite h. Setze $n = 1/h$:

$$\begin{aligned} t_0 &= -1, \quad y_0 = \frac{1}{101}, \\ t_{j+1} &= t_j + h, \quad y_{j+1} = y_j + h\phi(t_j, y_j, h), \quad j = 0, \ldots, n-1 \end{aligned}$$

a) explizites Euler-Verfahren: $\phi(t, y, h) = f(t, y)$

b) Verfahren von Heun: $\phi(t, y, h) = 0.5\,(f(t, y) + f(t + h, y + hf(t, y)))$

Wählen Sie verschiedene konstante Schrittweiten h zwischen 10^{-1} und 10^{-4}. Tragen Sie in einer Grafik jeweils den Logarithmus des absoluten globalen Diskretisierungsfehlers $|y_n^n - y(0)|$ gegenüber dem Logarithmus der Schrittweite auf. Was besagt diese Grafik?

Hinweise:

- Berechnen Sie die exakte Lösung mit Trennung der Variablen.
- Zur Bezeichnung:
  Zu einer gegebenen Schrittweite h ergibt sich die Anzahl der Schritte im Zeitintervall $[-1, 0]$ durch $n = 1/h$. Den Näherungswert im k-ten Schritt des Einschrittverfahrens mit diesen Parametern h und n bezeichnen wir mit y_k^n.

Lösung: Exakte Lösung durch Trennung der Variablen:

$$
\begin{aligned}
& & y' &= -200ty^2 \\
&\iff & \frac{dy}{dt} &= -200ty^2 \\
&\iff & \frac{dy}{y^2} &= -200tdt \\
&\iff & \int \frac{dy}{y^2} &= \int -200t\,dt \\
&\iff & -y^{-1} + C_0 &= -100t^2 + C_1 \\
&\iff & -y^{-1} &= -100t^2 + C_2, \quad \text{mit } C_2 := C_1 - C_0 \\
&\iff & -\frac{1}{y} &= -\frac{100t^2 + C_2}{1} \\
&\iff & y &= \frac{1}{100t^2 + C_2}
\end{aligned}
$$

Da $t_0 = -1$ und $y_0 = 1/101$ gilt, bestimmen wir C_2 so, dass dies erfüllt ist:

$$
1/101 \overset{!}{=} \frac{1}{100(-1)^2 + C_2} \quad \Longrightarrow \quad C_2 = 1
$$

Somit folgt für den exakten Wert $y(0) = 1$.

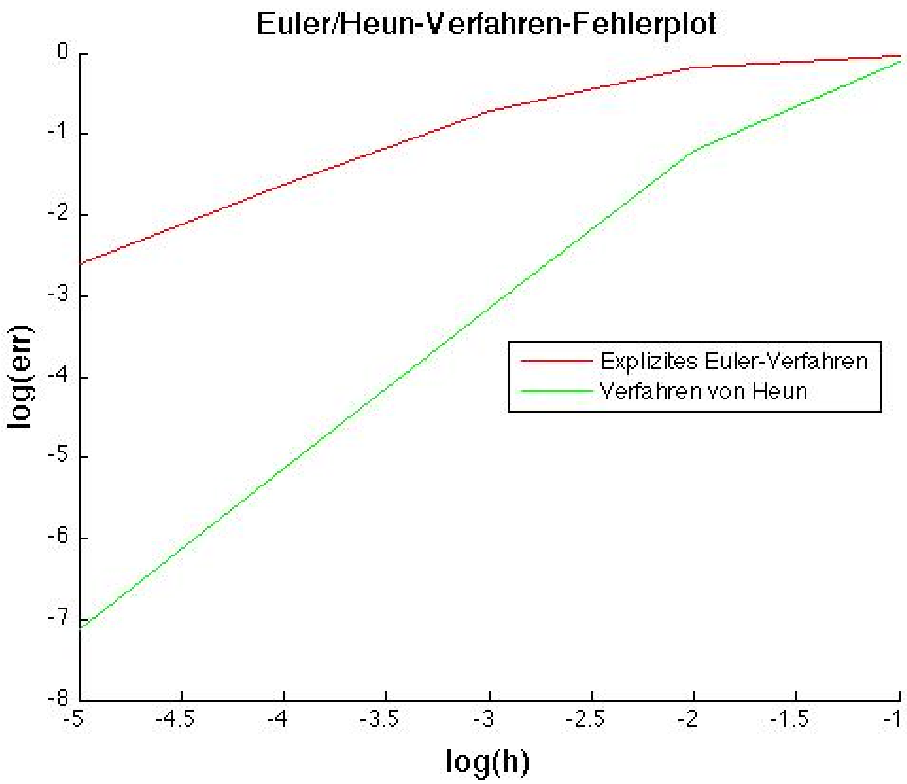
```
function y = einschritt(phi, f, N, t_0, y_0)

% Funktion zur Implementierung von Einschrittverfahren zur Loesung der ODE
%       y'=f(t,y)
% mit Anfangswert y(t_0)= y_0 mittels der Verfahrensfunktion phi und
% der Schrittweite h=1/N

h = 1/N;
t = t_0;

y = zeros(length(y_0), N+1);
y(:,1)= y_0(:)';

for k=1:N
    y(:,k+1)= y(:,k)+h*feval(phi, f, t, y(:,k),h);
    t = t + h;
end

end
```

```
function y = euler (f, t, y, h)
% Verfahrensfunktion Euler-Verfahren
y = feval (f, t, y);
```

```
function y = heun (f,t, y, h)
% Verfahrensfunktion Verfahren von Heun
fty = feval(f, t, y);
y = 0.5*(fty + feval(f, t+h, y+h*fty));
```

```
function dy = fun(t, y)
dy = -200 * t * y* y;
```

```
% Skript zum Erstellen der Grafik

clear all;
close all;

t_0 = -1;
y_0 = 1/101;

%N1 = logspace(1,5,5);
N1 = [logspace(1,5,5), 1000000,  10000000 ]

for k = 1:length(N1)
    y_euler  = einschritt (@euler, @fun, N1(k), t_0, y_0);
    y_heun  = einschritt (@heun , @fun, N1(k), t_0, y_0);
    err_euler(k) = abs( y_euler(end) -1);
    err_heun (k) = abs( y_heun(end) -1);
end

hold on;
%plot (log10(1./N1), log10 (err_euler(:)),'r');
%plot (log10(1./N1), log10 (err_heun(:)),'g');
%plot (log10(1./N1), log10(1./N1),'m');
%plot (log10(1./N1), log10((1./N1).^2), 'k');

plot (log(1./N1), log (err_euler(:)),'r');
plot (log(1./N1), log (err_heun(:)),'g');
plot (log(1./N1), log(1./N1),'m');
plot (log(1./N1), log((1./N1).^2), 'k');



title('Euler/Heun-Verfahren-Fehlerplot','FontSize', 14)
xlabel('log(h)', 'FontSize', 14)
ylabel('log(err)', 'FontSize', 14)
legend ('Explizites Euler-Verfahren', 'Verfahren von Heun', 'h','h^2',...
        'Location','Best')
hold off;
```


Euler/Heun-Verfahren-Fehlerplot
log(err)
log(h)
Explizites Euler-Verfahren
Verfahren von Heun
0
-1
-2
-3
-4
-5
-6
-7
-8
-5
-4.5
-4
-3.5
-3
-2.5
-2
-1.5
-1

Aufgabe 6 *(lokaler Diskretisierungsfehler, Konsistenzordnung)* (15 Punkte)

Zur Lösung der Anfangswertaufgabe $y' = f(t,y), y(t_0) = y_0$ betrachten wir die folgenden Einschrittverfahren zu einem äquidistanten Gitter mit der Schrittweite $h = (b-a)/n$:

a) Euler-Schritte mit Schrittweite h: $v_{j+1} = v_j + hf(t_j, v_j)$

b) Zwei Euler-Schritte mit halber Schrittweite:

$$w_{j+\frac{1}{2}} = w_j + \frac{h}{2}f(t_j, w_j), \quad w_{j+1} = w_{j+\frac{1}{2}} + \frac{h}{2}f(t_j + \frac{h}{2}, w_{j+\frac{1}{2}})$$

c) Extrapoliertes Verfahren: $u_{j+1} = \alpha v_{j+1} + (1-\alpha)w_{j+1}$

Bestimmen Sie jeweils den führenden Term des lokalen Diskretisierungsfehlers. Kann α so gewählt werden, dass der lokale Diskretisierungsfehler des extrapolierten Verfahrens von der Ordnung 3 ist?

Lösung: Allgemein gilt :

$$\begin{aligned}
z(t+h) - z(t) &\overset{Taylor}{=} \left(z(t) + hz'(t) + \frac{h^2}{2}z''(t) + \mathcal{O}(h^3) - z(t)\right) \\
&= hz'(t) + \frac{h^2}{2}z''(t) + \mathcal{O}(h^3) \\
&\overset{z'=f(t,z)}{=} hf(t,z) + \frac{h^2}{2}(f_t(t,z) + f_y(t,z)f(t,z)) + \mathcal{O}(h^3) \\
&= hf + \frac{h^2}{2}(f_t + f_y f) + \mathcal{O}(h^3)
\end{aligned}$$

a) Es gilt $\phi_a(t,y;h) = f(t,y) = f$. Da wir ein äquidistantes Gitter betrachten, reicht es d_{j+1} für ein beliebiges $j+1$ zu betrachten.

$$\begin{aligned}
d_{j+1} &= y(t_{j+1}) - y(t_j) - h\phi_a(t_j, y(t_j), h) \\
&= y(t_j + h) - y(t_j) - h\phi_a(t_j, y(t_j), h) \\
&= hf + \frac{h^2}{2}(f_t + f_y f) + \mathcal{O}(h^3) - hf \\
&= \frac{h^2}{2}(f_t + f_y f) + \mathcal{O}(h^3) = \mathcal{O}(h^2)
\end{aligned}$$

Das Verfahren hat somit die Ordnung $p = 1$.

b) Einsetzen liefert

$$\begin{aligned}
w_{j+1} &= w_{j+\frac{1}{2}} + \frac{h}{2}f(t_j + \frac{h}{2}, w_{j+\frac{1}{2}}) \\
&= w_j + \frac{h}{2}f(t_j, w_j) + \frac{h}{2}f(t_j + \frac{h}{2}, w_{j+\frac{1}{2}}) \\
&= w_j + \frac{h}{2}f(t_j, w_j) + \frac{h}{2}f(t_j + \frac{h}{2}, w_j + \frac{h}{2}f(t_j, w_j)) \\
&= w_j + h\underbrace{\frac{1}{2}\left(f(t_j, w_j) + f(t_j + \frac{h}{2}, w_j + \frac{h}{2}f(t_j, w_j))\right)}_{=:\phi_b}
\end{aligned}$$

damit folgt

$$\begin{aligned}
\phi_b &= \frac{1}{2}\left(f(t_j, w_j) + f(t_j + \frac{h}{2}, w_j + \frac{h}{2}f(t_j, w_j))\right) \\
&\overset{Taylor}{=} \frac{1}{2}\left(f + f + \frac{h}{2}f_t + \frac{h}{2}f_y f + \mathcal{O}(h^2)\right) \\
&= f + \frac{h}{4}(f_t + f_y f) + \mathcal{O}(h^2).
\end{aligned}$$

Somit erhalten wir

$$
\begin{aligned}
d_{j+1} &= y(t_{j+1}) - y(t_j) - h\phi_b(t_j, y(t_j), h) \\
&= y(t_j + h) - y(t_j) - h\phi_b(t_j, y(t_j), h) \\
&= hf + \frac{h^2}{2}(f_t + f_y f) + \mathcal{O}(h^3) - h(f + \frac{1}{4}h(f_t + f_y f) + \mathcal{O}(h^2)) \\
&= \frac{h^2}{2}(f_t + f_y f) - \frac{h^2}{4}(f_t + f_y f) + \mathcal{O}(h^3) \\
&= \frac{h^2}{4}(f_t + f_y f) + \mathcal{O}(h^3) = \mathcal{O}(h^2).
\end{aligned}
$$

Das Verfahren hat Ordnung $p = 1$.

c) Wähle $v_0 = w_0 = y_0$, dann gilt mit b)

$$
\begin{aligned}
u_{j+1} &= \alpha v_{j+1} + (1-\alpha) w_{j+1} \\
&= \alpha(v_j + hf(t_j, v_j)) + (1-\alpha)\left(w_j + \frac{h}{2}f(t_j, w_j) + \frac{h}{2}f(t_j + \frac{h}{2}, w_j + \frac{h}{2}f(t_j, w_j))\right) \\
&= \underbrace{\alpha v_j + (1-\alpha) w_j}_{=:u_j} + \alpha h \underbrace{f(t_j, v_j)}_{=\phi_a} + (1-\alpha)\frac{h}{2} \underbrace{\left(f(t_j, w_j) + f(t_j + \frac{h}{2}, w_j + \frac{h}{2}f(t_j, w_j))\right)}_{=2\phi_b} \\
&= u_j + \alpha h \phi_a + (1-\alpha) h \phi_b \\
&= u_j + h \underbrace{(\alpha\phi_a + (1-\alpha)\phi_b)}_{=:\phi_b}
\end{aligned}
$$

Daraus folgt

$$
\begin{aligned}
\phi_c &= \alpha\phi_a + (1-\alpha)\phi_b \\
&\overset{Taylor}{=} \alpha f + (1-\alpha)\left(f + \frac{h}{4}(f_t + f_y f) + \mathcal{O}(h^2)\right) \\
&= f + (1-\alpha)\frac{h}{4}(f_t + f_y f) + \mathcal{O}(h^2)
\end{aligned}
$$

Somit ergibt sich für den lokalen Diskretisierungsfehler

$$
\begin{aligned}
d_{j+1} &= y(t_{j+1}) - y(t_j) - h\phi_c(t_j, y(t_j), h) = y(t_j + h) - y(t_j) - h\phi_c(t_j, y(t_j), h) \\
&= hf + \frac{h^2}{2}(f_t + f_y f) + \mathcal{O}(h^3) + h\left(f + (1-\alpha)\frac{h}{4}(f_t + f_y f) + \mathcal{O}(h^2)\right) \\
&= \frac{h^2}{2}(f_t + f_y f) - (1-\alpha)\frac{h^2}{4}(f_t + f_y f) + \mathcal{O}(h^3)
\end{aligned}
$$

Wenn

$$\frac{1-\alpha}{4} = \frac{1}{2}$$

gilt, dann gilt $d_{j+1} = \mathcal{O}(h^3)$. Somit muss $\alpha = -1$ sein, damit das Verfahren die Ordnung $p = 2$ hat.