

1. Man berechne die Lösung des AWP

$$y' = \cos(x)y^2, \quad y(0) = -\frac{1}{2}.$$

2. Man löse das AWP aus Aufgabe 1 im Intervall $[0, 10]$ mit dem expliziten Euler-Verfahren. Dazu zerlege man das Intervall nacheinander in 10, 20, 40, 80, 160 gleichlange Teilintervalle und gebe jeweils den Fehler zwischen der berechneten Lösung und der analytischen Lösung im Punkt $x = 10$ an.

3. Die Aufgabenstellung ist wie in Aufgabe 2, wobei man jedoch jetzt das Verfahren von Heun und das Verfahren von Kutta verwenden soll.

Lösung

Aufgabe 1: Trennung der Veränderlichen ergibt

$$\begin{aligned} \int \frac{dy}{y} &= \int \cos(x) dx \implies \\ -y^{-1} &= \sin(x) + c \implies \\ y &= \frac{-1}{\sin(x) + c} \end{aligned}$$

Einsetzen der Anfangsbedingung ergibt

$$y(0) = -\frac{1}{c} = -\frac{1}{2} \implies c = 2$$

Ergebnis:

$$y = \frac{-1}{\sin(x) + 2}$$

Aufgaben 2 und 3:

$n =$	10	20	40	80	160
Euler	0.467468	0.348097	0.240644	0.15147	0.0876569
Heun	0.0989086	0.000822415	0.000389916	0.000245064	8.18657e-05
Kutta	0.188106	0.0343729	0.0059849	0.0012124	0.00026938

SCILAB

```
function r = expl_rkv_seriel2(b,n)

// expl_rkv_seriel2 ist die Implementierung dreier expl. RKV
// zur L"osung der Aufgaben aus Serie 12
// Aufruf r = expl_rkv_seriel2(b,n,fnc)
// b - rechte Intervallgrenze
// n - Anzahl der Zerlegungsintervalle

h = b/n;

x(1) = 0;
y(1) = -0.5;

// Explizites Euler-Verfahren
for i=1:n
    x(i+1) = i*h;
    y(i+1) = y(i) + h * cos(x(i))*y(i)*y(i);
end

K_1 = -1/(sin(x(n+1))+2)
s = sprintf('Explizit Euler: Fehler im Endpunkt %g',...
    abs(y(n+1) - K_1));
disp(s);

// Verfahren von Heun
c2 = 0.5;
b1 = 1-0.5/c2;
b2 = 0.5/c2;

for i=1:n
    x(i+1) = i*h;
    K_1 = cos(x(i))*y(i)*y(i);
    x1 = x(i)+c2*h;
    y1 = y(i)+h*c2*K_1;
    K_2 = cos(x1)*y1*y1;
    y(i+1) = y(i) + h*(b1*K_1 + b2*K_2);
end

K_1 = -1/(sin(x(n+1))+2)
s = sprintf('Verfahren von Heun: Fehler im Endpunkt %g',...
    abs(y(n+1) - K_1));
disp(s);

// Verfahren von Kutta
c2 = 1;
b1 = 1-0.5/c2;
b2 = 0.5/c2;

for i=1:n
    x(i+1) = i*h;
    K_1 = cos(x(i))*y(i)*y(i);
    x1 = x(i)+c2*h;
```

```
    y1 = y(i)+h*c2*K_1;
    K_2 = cos(x1)*y1*y1;
    y(i+1) = y(i) + h*(b1*K_1 + b2*K_2);
end

K_1 = -1/(sin(x(n+1))+2)
s = sprintf('Verfahren von Kutta: Fehler im Endpunkt %g',...
    abs(y(n+1) - K_1));
disp(s);

r = 0;

endfunction
```

MATLAB

```
function r = expl_rkv_seriel2(b,n,fnc)

% expl_rkv_seriel2 ist die Implementierung dreier expl. RKV
% zur L"osung der Aufgaben aus Serie 12
% Aufruf r = expl_rkv_seriel2(b,n,fnc)
% b - rechte Intervallgrenze
% n - Anzahl der Zerlegungsintervalle
% fnc - rechte Seite der Dgl.

h = b/n;

x(1) = 0;
y(1) = -0.5;

% Explizites Euler-Verfahren
for i=1:n
    x(i+1) = i*h;
    t = [x(i),y(i)]';
    K_1 = feval(fnc,t);
    y(i+1) = y(i) + h * K_1(1);
end

t = [x(n+1),y(n+1)]';
K_1 = feval(fnc,t);
s = sprintf('Explizit Euler: Fehler im Endpunkt %g',...
    abs(y(n+1) - K_1(2)));
disp(s);

clf
plot(x,y,'-or')
hold on

% Verfahren von Heun
c2 = 0.5;
b1 = 1-0.5/c2;
b2 = 0.5/c2;

for i=1:n
    x(i+1) = i*h;
    t = [x(i),y(i)]';
    K_1 = feval(fnc,t);
    t = [x(i)+c2*h,y(i)+h*c2*K_1(1)]';
    K_2 = feval(fnc,t);
    y(i+1) = y(i) + h*(b1*K_1(1) + b2*K_2(1));
end

t = [x(i+1),y(i+1)]';
K1 = feval(fnc,t);
s = sprintf('Verfahren von Heun: Fehler im Endpunkt %g',...
    abs(y(n+1) - K1(2)));
disp(s);
```

```

plot(x,y,'-sg')
hold on

% Verfahren von Kutta
c2 = 1;
b1 = 1-0.5/c2;
b2 = 0.5/c2;

for i=1:n
    x(i+1) = i*h;
    t = [x(i),y(i)]';
    K_1 = feval(fnc,t);
    t = [x(i)+c2*h,y(i)+h*c2*K_1(1)]';
    K_2 = feval(fnc,t);
    y(i+1) = y(i) + h*(b1*K_1(1) + b2*K_2(1));
end

t = [x(i+1),y(i+1)]';
K1 = feval(fnc,t);
s = sprintf('Verfahren von Kutta: Fehler im Endpunkt %g',...
    abs(y(n+1) - K1(2)));
disp(s);

plot(x,y,'-db')
hold on

h = b/1000;
for i=1:1001
    xf(i) = (i-1)*h;
    t = [xf(i),0]';
    K1 = feval(fnc,t);
    ff(i) = K1(2);
end

plot(xf,ff,':k')
legend('Euler','Heun','Kutta');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [val] = ode_seriel2_fnc(t)

% rechte Seite
val(1) = cos(t(1))*t(2)*t(2);

% L"osung zum AWP y(0) = -1/2
val(2) = -1/(sin(t(1))+2);

```