Projekt 1: Implizite Euler und Trapezmethode

MND2

Andreas Bachmann
bachman0@students.zhaw.ch

5. November 2018

Einführung

Gesucht ist die Lösung des nichtlinearen Anfangswertproblems:

$$x \in (0, 4\pi]$$
$$y'(x) = \cos(y(x)) + \sin(x)$$
$$y(0) = -1$$

Aufgabe 1

Berechnen Sie mit Hilfe der Methode nach Runge numerisch eine Lösung der DGL.

Lösung

```
%% Projekt 1, Aufgabe 1
clear all;
close all;
clc;

% x = [0, 4*pi]
% y'(x) = cos(y(x)) + sin(x)
% y(x)  = -1

f       = @(xk, yk) cos(yk) + sin(xk);

h       = 0.1;
y0      = -1;
xEnd    = 4*pi;

[x, y] = explicitRunge(f, h, xEnd, y0);
plot(x, y);
grid on;
grid minor;
xlabel('x');
ylabel('y');
legend('y(x)');
```

Listing 1: `aufgabe1.m`

$$
\begin{aligned}
r_1 &= f\left(x_k,\ y_k\right)\\
r_2 &= f\left(x_k + \frac{h}{2},\ y_k + \frac{h}{2} \cdot r_1\right)\\
y_{k+1} &= y_k + h \cdot r_2
\end{aligned}
$$

Algorithmus

```
function [t, y] = explicitRunge(f, dt, Tend, y0)
    n        = round(Tend / dt);
    t        = zeros(n + 1, 1);
    y        = zeros(n + 1, length(y0));
    y(1,:)   = y0;
    for i = 1:n
        r1           = f(t(i), y(i,:));
        r2           = f(t(i) + dt/2,  y(i,:) + dt/2 * r1);
        y(i + 1, :)  = y(i,:) + dt * r2;
        t(i + 1)     = t(i) + dt;
    end
end
```

Listing 2: `explicitRunge.m`

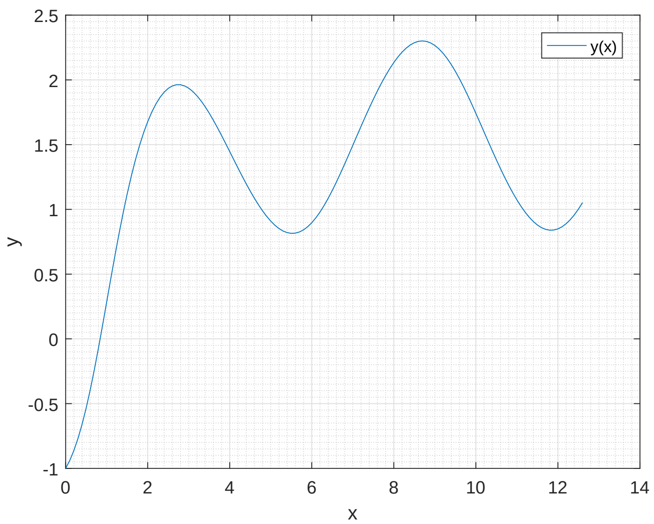

Abbildung 1: Explizites Runge-Verfahreb

Aufgabe 2

Notieren Sie die nichtlineare Gleichung für das **implizite Eulerverfahren**. Die Gleichung ist analytisch nicht nach y_{n+1} auflösbar.

Lösung

Nichtlineare Gleichung für das implizite Eulerverfahren

$$\begin{aligned} f\left(x,\ y\right) &= y'\left(x\right) = \cos\left(y\left(x\right)\right) + \sin\left(x\right) \\ y_{k+1} &= y_k + h \cdot f\left(x_{k+1},\ y_{k+1}\right) \end{aligned}$$

Newton-Verfahren

$$\begin{aligned} y_{k+1} &= y_k + h \cdot f\left(x_{k+1}, y_{k+1}\right) \\ s &= y_k + h \cdot f\left(x_{k+1}, s\right) \\ 0 &= s - y_k - h \cdot f\left(x_{k+1}, s\right) \\ G\left(s\right) &= s - y_k - h \cdot f\left(x_{k+1}, s\right) = 0 \\ s^{(n+1)} &= s^{(n)} - \frac{G\left(s^{(n)}\right)}{G'\left(s^{(n)}\right)} \\ s^{(n)} &\rightarrow s^{(\text{Iterationsschritt})} \end{aligned}$$

Einsetzen der Funktion

$$\begin{aligned} f\left(x, y\right) &= \cos\left(y\right) + \sin\left(x\right) \\ G\left(s\right) &= s - y_k - h \cdot f\left(x_{k+1}, s\right) \\ G\left(s\right) &= s - y_k - h \cdot \left[\cos\left(s\right) + \sin\left(x_{k+1}\right)\right] \\ G\left(s\right) &= s - y_k - h \cdot \cos\left(s\right) - h \cdot \sin\left(x_{k+1}\right) \end{aligned}$$

$G'\left(s\right)$ ableiten

```
syms s yk h xkp1
g = s - yk - h*cos(s) - h*sin(xkp1)
diff(g,s)
% h*sin(s) + 1
```

Einsetzen der Funktion

$$\begin{aligned} G\left(s\right) &= s - y_k - h \cdot \cos\left(s\right) - h \cdot \sin\left(x_{k+1}\right) \\ G'\left(s\right) &= \frac{dG}{ds} = h \cdot \sin\left(s\right) + 1 \\ s^{(n+1)} &= s^{(n)} - \frac{G\left(s^{(n)}\right)}{G'\left(s^{(n)}\right)} \\ s^{(n+1)} &= s^{(n)} - \frac{s^{(n)} - y_k - h \cdot \cos\left(s^{(n)}\right) - h \cdot \sin\left(x_{k+1}\right)}{h \cdot \sin\left(s^{(n)}\right) + 1} \end{aligned}$$

Aufgabe 3

Setzen Sie den folgenden Algorithmus um

```
%% Projekt 1, Aufgabe 3
clear all;
close all;
clc;

% x = [0, 4*pi]
% y'(x) = cos(y(x)) + sin(x)
% y(x)  = -1

G            = @(h, s, yk, xkp1)    s - yk - h*cos(s) - h*sin(xkp1);
dG           = @(h, s, yk, xkp1)    h*sin(s) + 1;

h            = 0.5;
y0           = -1;
xEnd         = 4*pi;
tolerance    = 10e-8;
maxIter      = 300;

[x, y] = implizitEulerNewton(G, dG, h, xEnd, y0, tolerance, maxIter);

plot(x, y);
grid on,
grid minor;
xlabel('x');
ylabel('y');
legend('y(x)');
```

Listing 3: `aufgabe3.m`

```
function [x, y] = implizitEulerNewton(G, dG, h, Xend, y0, tolerance, maxIter)
    n        = round(Xend / h);
    x        = zeros(n + 1, 1);
    y        = zeros(n + 1, 1);
    y(1)   = y0;
    for i = 1:n
        x(i + 1) = x(i) + h;
        y(i + 1) = newton(G, dG, h, y(i), y(i), x(i + 1), tolerance, maxIter);
    end
end
```

Listing 4: `implizitEulerNewton.m`

```

function [s] = newton(G, dG, h, s0, yk, xkp1, tolerance, maxIter)
    iter   = 0;
    err    = 1;
    s      = s0;
    while err > tolerance
        snp1 = s - (G(h, s, yk, xkp1) / dG(h, s, yk, xkp1));
        %err    = abs(snp1 - s);
        err    = abs(G(h, s, yk, xkp1));
        iter   = iter + 1;

        if (iter >= maxIter)
            error('Maximale Iteration erreicht!');
        end

```

```
        s = snp1;
    end
    fprintf('s0=%5.3f s=%5.3f iter=%d\n', s0, s, iter);
end
```

Listing 5: `newton.m`

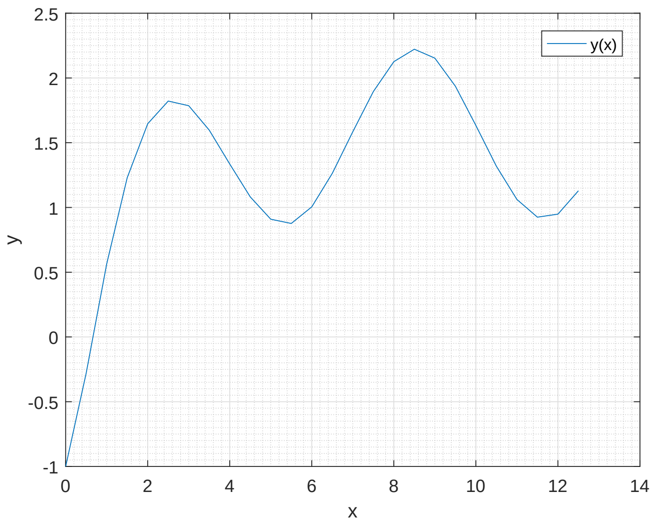

Abbildung 2: Implizites Euler-Verfahren

Aufgabe 4

Wenden Sie die Methode auf die **implizite Trapezmethode** an.

Lösung

```
%% Projekt 1, Aufgabe 4
clear all;
close all;
clc;

f            = @(xk, yk) cos(yk) + sin(xk);
G            = @(h, s, yk, xkp1)    s - yk - h*cos(s) - h*sin(xkp1);
dG           = @(h, s, yk, xkp1)    h*sin(s) + 1;

h            = 0.5;
y0           = -1;
xEnd         = 4*pi;
tolerance    = 10e-8;
maxIter      = 300;

[x, y] = implicitTrapez(f, G, dG, h, xEnd, y0, tolerance, maxIter);

plot(x, y);
grid on,
grid minor;
xlabel('x');
ylabel('y');
legend('y(x)');
```

Listing 6: `aufgabe4.m`

```
function [x, y] = implicitTrapez(f, G, dG, h, Xend, y0, tolerance, maxIter)
    n        = round(Xend / h);
    x        = zeros(n + 1, 1);
    y        = zeros(n + 1, 1);
    y(1)   = y0;

    for i = 1:n
        x(i + 1) = x(i) + h;
        k1         = newton(G, dG, h, y(i), y(i), x(i + 1), tolerance, maxIter);
        y(i+1)     = y(i) + h *(f(x(i), y(i)) + f(x(i) + h,k1)) / 2;
    end;
end
```

Listing 7: `implicitTrapez.m`

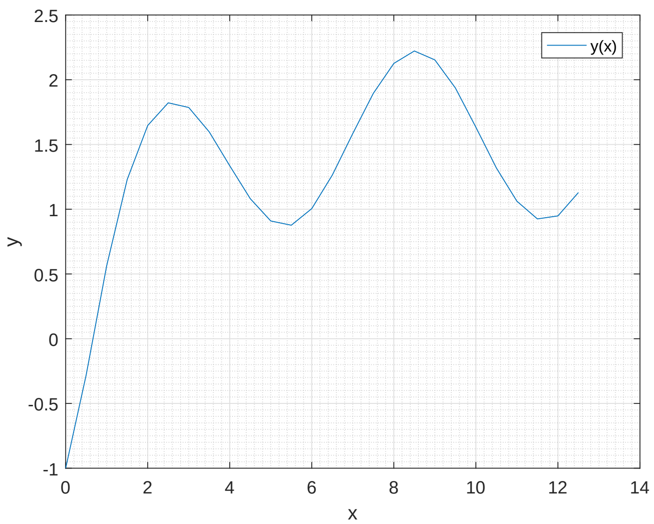

Abbildung 3: Implizite Trapezmethode

Aufgabe 5

Vergleichen Sie die drei verschiedenen Lösungen mit Hilfe einer Referenzlösung, für welche Sie eine 10x kleinere Schrittweite h verwenden.

Lösung

```
%% Projekt 1, Aufgabe 5
clear all;
close all;
clc;

f            = @(xk, yk) cos(yk) + sin(xk);
G            = @(h, s, yk, xkp1)    s - yk - h*cos(s) - h*sin(xkp1);
dG           = @(h, s, yk, xkp1)    h*sin(s) + 1;

h            = 1;
y0           = -1;
xEnd         = 4*pi;
tolerance    = 10e-8;
maxIter      = 300;

[x_r, y_r] = explicitRunge(f, h, xEnd, y0);
[x_e, y_e] = implizitEulerNewton(G, dG, h, xEnd, y0, tolerance, maxIter);
[x_t, y_t] = implicitTrapez(f, G, dG, h, xEnd, y0, tolerance, maxIter);

plot(x_r, y_r, x_e, y_e, x_t, y_t);
grid on,
grid minor;

xlabel('x');
ylabel('y');
legend('y(x) explicitRunge', 'y(x) implizitEulerNewton', 'y(x) implicitTrapez');
legend('Location','southeast')
```

Listing 8: `aufgabe5.m`

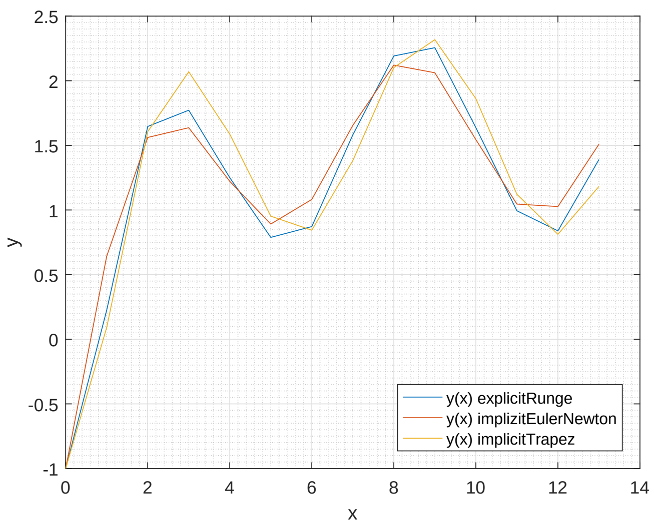

Abbildung 4: Vergleich

Aufgabe 6

Leiten Sie die linearisierte DGL her und lösen Sie diese mit Hilfe der impliziten Trapezmethode. Vergleichen Sie die Lösung mit der Lösung aus Schritt 4.

Lösung

```
%% Projekt 1, Aufgabe 6
clear all;
close all;
clc;

f            = @(xk, yk) cos(yk) + sin(xk);
f_lin        = @(xk, yk) 1 + sin(xk);
G            = @(h, s, yk, xkp1)    s - yk - h*cos(s) - h*sin(xkp1);
dG           = @(h, s, yk, xkp1)    h*sin(s) + 1;

h            = 0.1;
y0           = -1;
xEnd         = 4*pi;
tolerance    = 10e-8;
maxIter      = 300;

[x_t,   y_t]   = implicitTrapez(f,     G, dG, h, xEnd, y0, tolerance,
    maxIter);
[x_lin, y_lin] = implicitTrapez(f_lin, G, dG, h, xEnd, y0, tolerance,
    maxIter);

plot(x_t, y_t, x_lin, y_lin);
grid on,
grid minor;

xlabel('x');
ylabel('y');
legend('y(x) implicitTrapez', 'y(x) implicitTrapez mit linearer Funktiuon
    ');
legend('Location','southeast')
```

Listing 9: `aufgabe6.m`

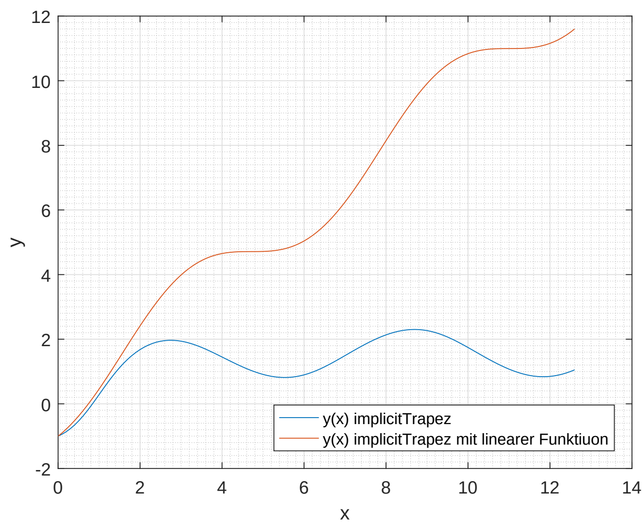

Abbildung 5: Vergleich mit Linear

Aufgabe 7

Leiten Sie die DGL mit einer Reihenentwicklung bis zum quadratischen Term her und lösen Sie diese mit Hilfe der impliziten Trapezmethode. Vergleichen Sie die Lösung wiederum mit der Lösung aus Schritt 4. (Die Verfahrensformel kann analytisch berechnet werden.)

Lösung

$$
\begin{aligned}
T_{\infty,\,x_0} f(x) &= \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k \\
f(y) &= \cos(y) \\
T_{2,\,0} f(y) &= \cos(0) - \sin(0) \cdot y - \frac{\cos(0)}{2!} \cdot y^2 \\
&= 1 - \frac{1}{2} \cdot y^2 \\
y'(x) &= f(x,\ y) \\
y'(x) &= 1 - \frac{1}{2} \cdot y^2 + \sin(x)
\end{aligned}
$$

Algorithmus

```
%% Projekt 1, Aufgabe 7
clear all;
close all;
clc;

f           = @(xk, yk) cos(yk) + sin(xk);
f_quad      = @(xk, yk) 1 - 1/2*yk^2 + sin(xk);
G           = @(h, s, yk, xkp1)    s - yk - h*cos(s) - h*sin(xkp1);
dG          = @(h, s, yk, xkp1)    h*sin(s) + 1;

h           = 0.1;
y0          = -1;
xEnd        = 4*pi;
tolerance   = 10e-8;
maxIter     = 300;

[x_t,   y_t]   = implicitTrapez(f,      G, dG, h, xEnd, y0, tolerance,
    maxIter);
[x_lin, y_lin] = implicitTrapez(f_quad, G, dG, h, xEnd, y0, tolerance,
    maxIter);

plot(x_t, y_t, x_lin, y_lin);
grid on,
grid minor;

xlabel('x');
ylabel('y');
legend('y(x) implicitTrapez', 'y(x) implicitTrapez mit Reihenentwicklung 
    bis zu quadratischem Term');
legend('Location','southeast')
```

Listing 10: `aufgabe7.m`

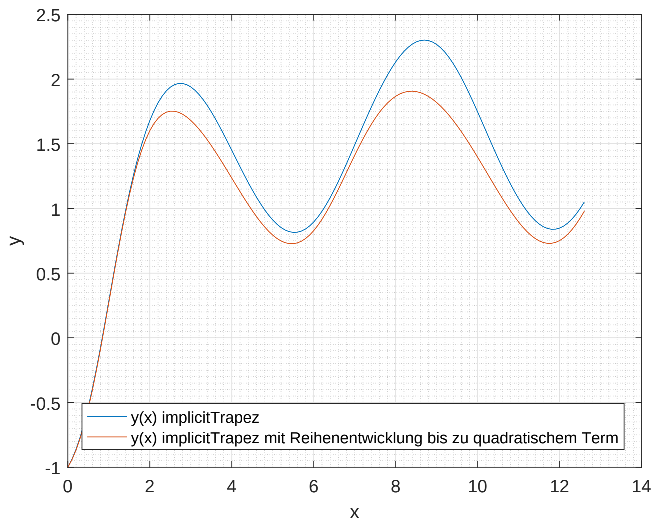

Abbildung 6: Vergleich mit Reihenentwicklung