

A középponti módszer

Nézzünk most egy kétlépéses numerikus módszert: a középponti módszert. Ez úgy működik, hogy először a t_i -beli deriválttal ($s_1 = f(t_i, y_i)$) lépünk $h/2$ lépésközön, és megnézzük itt a meredekséget, ami nem más, mint $f(t_i + h/2, y_i + h/2s_1)$. Ezzel lépünk végül y_i -ből h távolságon. Tehát az i -edik részintervallumon elvégzett számítás főbb lépései:

$$t_{i+1} = t_i + h,$$

$$s_1 = f(t_i, y_i),$$

$$s_2 = f(t_i + h/2, y_i + h/2s_1),$$

$$y_{i+1} = y_i + hs_2.$$

Feladatok:

1. Programozzuk be a középponti módszert.
2. Oldjuk meg vele a múltkori első példát.
3. Készítsünk programot, amely mindkét módszerrel megoldja a feladatot, és egy ábrában kirajzolja a megoldásgörbéket.

A középponti módszer kódja:

```
% Kozeppeponti modszer
% A diffegy alfuggvény által megadott differencialegyenletet
% megoldja a t0 pontbol es y0 kezdeti ertekbol kiindulva
% h lepeskozzel es N db lepessele
% Hasznalata: pl. [tmid,ymid] = midpoint(@diffegy, 0, 1, 0.1, 10)
function [t,y] = midpoint(diffegy, t0, y0, h, N)
t=zeros(N+1,1); y = zeros(N+1,1);
t(1) = t0;
y(1) = y0;
for i=1:N
    s1=diffegy(t(i),y(i));
    s2=diffegy(t(i)+h/2,y(i)+h/2*s1);
    y(i+1) = y(i) + h * s2;
    t(i+1) = t(i) + h;
end
```

```

% Pontos megoldas
u=exp(-t)+t;
plot(t,u,'b',t,y,'r')
xlabel('t')
ylabel('y')
title('Kozeponti modszer')
legend('Pontos megoldas','Numerikus megoldas')

```

A két numerikus módszert összehasonlító programot nem függvényként készítettük el, hanem scriptként, a bemenő adatokat a program elején adjuk meg. Ha más bemenő adatokra akarjuk elvégezni az összehasonlítást, a program elején átírhatók.

```

% Az explicit Euler-modszer es a kozeponti modszer
% osszehasonlitasa
% A diffegy alfuggveny által megadott differencialegyenletet
% megoldja a t0 pontból es y0 kezdeti ertekekből kiindulva
% h lepeskozzel es N db lepesseel
clear all
close all
t0=0;
y0=1;
h=0.1;
N=10;
t=zeros(N+1,1);
y = zeros(N+1,1);
% Euler-modszer
t(1) = t0;
y(1) = y0;
for i=1:N
    t(i+1) = t(i) + h;
    y(i+1) = y(i) + h * diffegy(t(i),y(i));
end
y_ee = y;
clear t y; % Kozeponti modszer
t(1) = t0;
y(1) = y0;
for i=1:N

```

```

    s1=diffegy(t(i),y(i));
    s2=diffegy(t(i)+h/2,y(i)+h/2*s1);
    y(i+1) = y(i) + h*s2;
    t(i+1) = t(i) + h;
end y_mid = y;
% Pontos megoldas
u=exp(-t)+t;
plot(t,u,'b',t,y_ee,'r',t,y_mid,'g')
xlabel('t')
ylabel('y')
title('Explicit Euler es kozeponti modszer osszehasonlitasa')
legend('Pontos megoldas','Explixit Euler-modszer','Kozeponti modszer')

```

A fenti program egyetlen ábrába rajzolja bele mindkét módszer eredményét és a pontos megoldást. Az utolsó rész úgy is megírható, hogy két külön részábrában lássuk az eredményeket. Ehhez a `subplot` parancsot használhatjuk. A `subplot` után zárójelben megadjuk, hogy hány sornyi és oszlopnyi részábra lesz. A mi esetünkben egy sorban szeretnénk két ábrát, tehát 1,2. Utána vesszővel elválasztva az aktuális részábra sorszáma következik. Tehát ekkor a program vége így fest:

```

u=exp(-t)+t;
subplot(1,2,1) plot(t,u,'b',t,y_ee,'r')
legend('Pontos megoldas','Explixit Euler-modszer')
xlabel('t')
ylabel('y')
subplot(1,2,2) plot(t,u,'b',t,y_mid,'g')
xlabel('t')
ylabel('y')
legend('Pontos megoldas','Kozeponti modszer')

```