Diferenciálne rovnice

Na intervale [0, 10] je budeme riešiť diferenciálnu rovnicu:

y(0) = 2
y' = 3-7*sin(5*x)+1.4*y.^0.5.*cos(y)

Euler:
Začneme najjednoduchšou Eulerovou metódou s krokom 0.2 a postupne krokovanie zjemníme. Sme si vedomí toho, že počiatočný krok 0.2 je z hľadiska presnosti výpočtu neprimerane veľký,[1] len by sme radi videli aj na obrázku, čo to znamená.

Najprv si pripomenieme princíp výpočtu:

0 – Začíname v x = 0 a vtedy y = 2. Smernica v tomto bode je

3-7*sin(5*0)+1.4*2.^0.5.*cos(2) = 2.17607129974292

0,2 – V ďalšom bode je preto

y = 2 + 0.2*2.17607129974292 = 2.435214259948584

A takto pokračujeme až do bodu 10… a samozrejme, chceme to mať rýchlo:

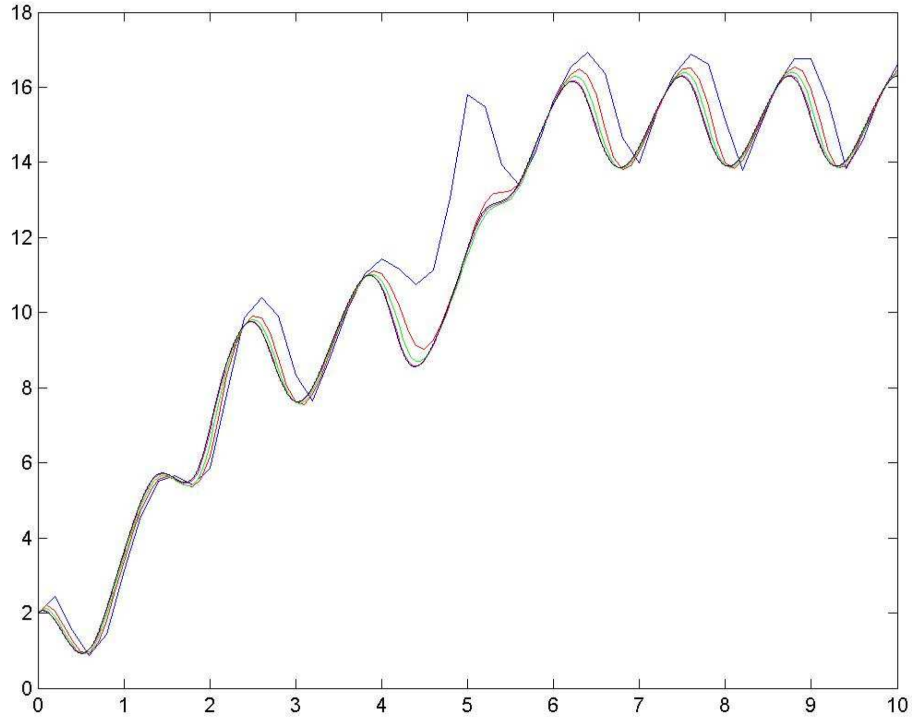
```
>> h=0.2; n=10/h+1; x=linspace(0,10,n);                    delenie intervalu 0-10
>> f=inline('3-7*sin(5*x)+1.4*y.^0.5.*cos(y)'); y=2; t=0;   začínam od [0.2]
   for k=2:n, y(k)=y(k-1)+h*f(t,y(k-1)); t=t+h; end, plot(x,y), hold on
```

v cykle robím po krokoch 0,2 výpočet y,
hodnoty y ukladáme do vektora y
a vykreslíme

Ďalej budeme postupne zjemňovať krok. Každý výsledok vykreslíme inou farbou. Očakávame pochopiteľne, že čím jemnejšie delenie, tým bude výsledok presnejší.

```
>> h=0.1; n=10/h+1; x=linspace(0,10,n);
>> y=2; t=0; for k=2:n, y(k)=y(k-1)+h*f(t,y(k-1)); t=t+h; end, plot(x,y,'r'), hold on
>> h=0.05; n=10/h+1; x=linspace(0,10,n);
>>  y=2; t=0; for k=2:n, y(k)=y(k-1)+h*f(t,y(k-1)); t=t+h; end, plot(x,y,'g')
>> h=0.01; n=10/h+1; x=linspace(0,10,n);
>>  y=2; t=0; for k=2:n, y(k)=y(k-1)+h*f(t,y(k-1)); t=t+h; end, plot(x,y,'m')
>> h=0.001; n=10/h+1; x=linspace(0,10,n);
>>  y=2; t=0; for k=2:n, y(k)=y(k-1)+h*f(t,y(k-1)); t=t+h; end, plot(x,y,'k')
```

[1] V základnom predpise funkcie vystupuje 7*sin(5x). Sínus mení svoju hodnotu od -1 po 1 na intervale dĺžky *pí*, takže 7*sin(5x) zvládne -7 až 7 na *pí/5*, čo je orientačne 0.6. Delenie po 0.2 (tretina 0.6) teda nemá šancu zachytávať zmeny smernice dostatočne citlivo.



Nakoniec tá modrá čiara pri delení 0.2 nedopadla až tak katastrofálne, hoci o nejakej ohurujúcej presnosti nemôže byť reč. Červená čiara je na tom lepšie a zvyšné tri čiary sa takmer prekrývajú. Takže delenie s krokom 0.05 už môžeme považovať za primerané.

Euler – modifikácia:

Budeme počítať to isté modifikovanou Eulerovou metódou.
Opäť začneme pripomenutím si princípu a opäť začneme krokom 0.2, aby sme sa potom mohli tešiť z toho, ako nám všetko krásne konverguje.

0 – Začíname v x = 0 a vtedy y = 2. Smernica v tomto bode je

k1= 3-7*sin(5*0)+1.4*2.^0.5.*cos(2) = 2.17607129974292

0.1 – V smere danom smernicou urobíme POLkrok vpred a máme predbežné

y(0.1) = 2+0.1*2.17607129974292 = 2.217607129974292

Korigovaná smernica rátaná v 0.1:

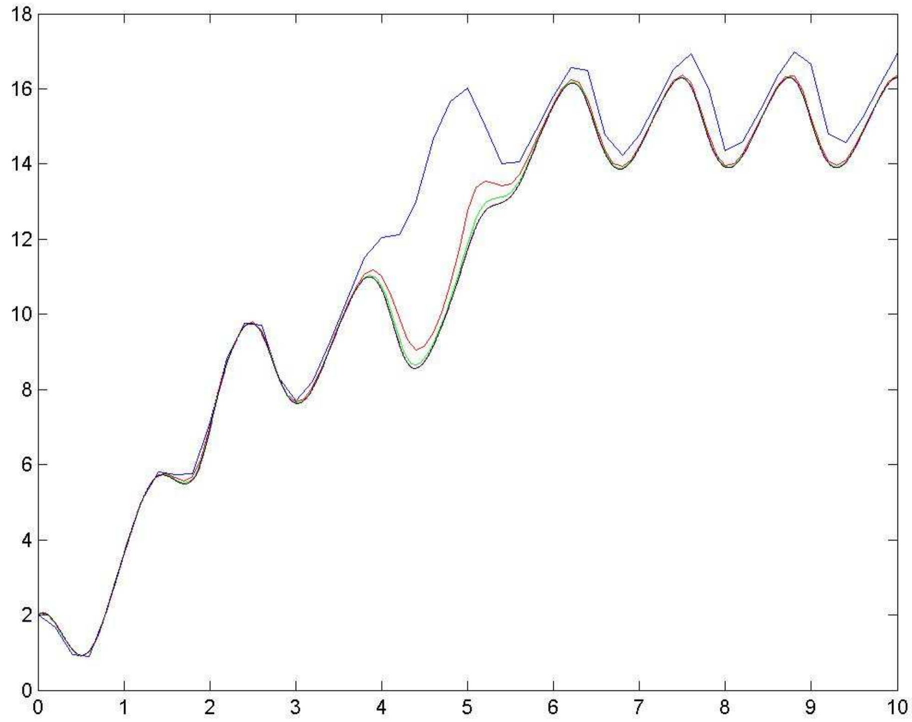
k2 = 3-7*sin(5*0.1)+1.4*2.217607129974292^0.5*cos(2.217607129974292)
= -1.612389139594002

0,2 – V ďalšom bode je potom hodnota

y = 2 + 0.2*-1.612389139594002 = 1.677522172081200

Takto to urobíme pre celý interval 0 až 10, a potom aj pre jemnejšie delenia:

```
>> h=0.2; n=10/h+1; x=linspace(0,10,n);
>> f=inline('3-7*sin(5*x)+1.4*y.^0.5.*cos(y)'); y=2; t=0; for k=2:n,
        s=y(k-1)+0.5*h*f(t,y(k-1)); y(k)=y(k-1)+h*f(t+0.5*h,s); t=t+h; end, plot(x,y), hold on
>> h=0.1; n=10/h+1; x=linspace(0,10,n);  y=2; t=0;
        for k=2:n, s=y(k-1)+0.5*h*f(t,y(k-1)); y(k)=y(k-1)+h*f(t+0.5*h,s); t=t+h; end, plot(x,y,'r')
>> h=0.05; n=10/h+1; x=linspace(0,10,n);  y=2; t=0;
        for k=2:n, s=y(k-1)+0.5*h*f(t,y(k-1)); y(k)=y(k-1)+h*f(t+0.5*h,s); t=t+h; end, plot(x,y,'g')
>> h=0.01; n=10/h+1; x=linspace(0,10,n); y=2; t=0;
        for k=2:n, s=y(k-1)+0.5*h*f(t,y(k-1)); y(k)=y(k-1)+h*f(t+0.5*h,s); t=t+h; end, plot(x,y,'m')
>> h=0.001; n=10/h+1; x=linspace(0,10,n); y=2; t=0;
        for k=2:n, s=y(k-1)+0.5*h*f(t,y(k-1)); y(k)=y(k-1)+h*f(t+0.5*h,s); t=t+h; end, plot(x,y,'k')
```



Komentár k obrázku by bol podobný ako v predošlom prípade, s dodatkom, že pri tom trápení sa navyše by človek čakal rýchlejšiu konvergenciu a razantnejšie blíženie sa k správnemu výsledku.

<u>Heunova metóda:</u>

Pripomenieme si princíp:

0 – Začíname v x = 0 a y = 2. Smernica v tomto bode je k1= 2.17607129974292

(0.2) – V smere danom smernicou urobíme predbežný krok vpred a máme

ypom(0.2) = 2 + 0.2*2.17607129974292 = 2.435214259948584

Korigovaná smernica rátaná v 0.2:

k2 = 3-7*sin(5*0.2)+1.4*2.435214259948584^0.5*cos(2.435214259948584)
= -4.552255026992040

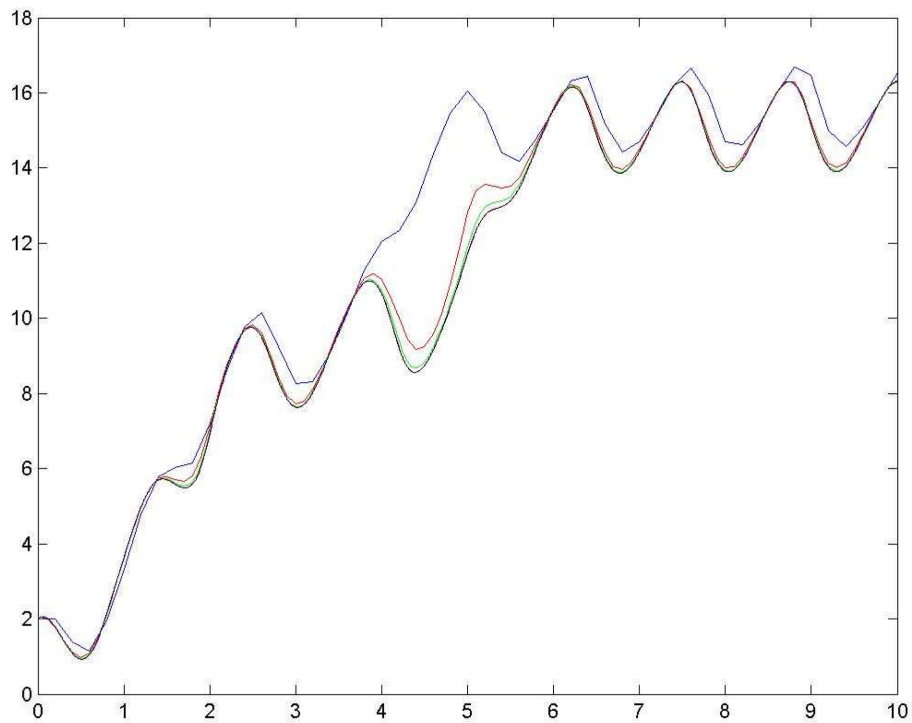
0,2 – V ďalšom bode je potom hodnota y výsledkom kroku so smernicou

k=(k1+k2)/2= -1.18809186362456

y = 2 + 0.2*-1.18809186362456 = 1.762381627275088

Necháme si vyrátať všetko naraz a pre ďalšie delenia.

```
>> clear all
>> h=0.2; n=10/h+1; x=linspace(0,10,n);
>> f=inline('3-7*sin(5*x)+1.4*y.^0.5.*cos(y)'); y=2; t=0;
>> for k=2:n, k1=f(t,y(k-1)); yp=y(k-1)+h*k1; k2=f(t+h,yp); y(k)=y(k-1)+h*(k1+k2)/2; t=t+h; end,
        plot(x,y), hold on
>> h=0.1; n=10/h+1; x=linspace(0,10,n); y=2; t=0;
>> for k=2:n, k1=f(t,y(k-1)); yp=y(k-1)+h*k1; k2=f(t+h,yp); y(k)=y(k-1)+h*(k1+k2)/2; t=t+h; end, plot(x,y,'r')
>> h=0.05; n=10/h+1; x=linspace(0,10,n); y=2; t=0;
>> for k=2:n, k1=f(t,y(k-1)); yp=y(k-1)+h*k1; k2=f(t+h,yp); y(k)=y(k-1)+h*(k1+k2)/2; t=t+h; end, plot(x,y,'g')
>> h=0.01; n=10/h+1; x=linspace(0,10,n); y=2; t=0;
>> for k=2:n, k1=f(t,y(k-1)); yp=y(k-1)+h*k1; k2=f(t+h,yp); y(k)=y(k-1)+h*(k1+k2)/2; t=t+h; end, plot(x,y,'m')
>> h=0.001; n=10/h+1; x=linspace(0,10,n); y=2; t=0;
>> for k=2:n, k1=f(t,y(k-1)); yp=y(k-1)+h*k1; k2=f(t+h,yp); y(k)=y(k-1)+h*(k1+k2)/2; t=t+h; end, plot(x,y,'k')
```



Obrázok sa skoro nelíši od predošlého.

Runge-Kutta 4. rádu:

Princíp metódy je podobný ako v modifikácii Eulera, akurát výpočet potiahneme ďalej a vhodne výsledky spriemerujeme.

0 – Začíname v x = 0 a vtedy y = 2. Smernica v tomto bode je

k1= 3-7*sin(5*0)+1.4*2.^0.5.*cos(2) = 2.17607129974292

(0.1) – V smere danom smernicou urobíme POLkrok vpred a máme predbežné

yp(0.1) = 2+0.1*2.17607129974292 = 2.217607129974292

Korigovaná smernica rátaná v 0.1:

k2 = 3-7*sin(5*0.1)+1.4*2.217607129974292^0.5*cos(2.217607129974292)
= -1.612389139594002

{0.1} – V smere danom k2 urobíme zase polkrok:

yq(0.1) = 2+0.1*-1.612389139594002 = 1.8387610860406

Ďalšia korekcia smernice:

k3 = 3-7*sin(5*0.1)+1.4*1.8387610860406^0.5*cos(1.8387610860406)
= -0.8586203896160678

(0.2) – V smere k3 urobíme celý krok vpred a zistíme si smernicu:

yr(0.2) = 2 + 0.2*-0.8586203896160678 = 1.828275922076786

k4 = 3-7*sin(5*0.2)+1.4*1.828275922076786^0.5*cos(1.828275922076786)
= -3.372336172843774

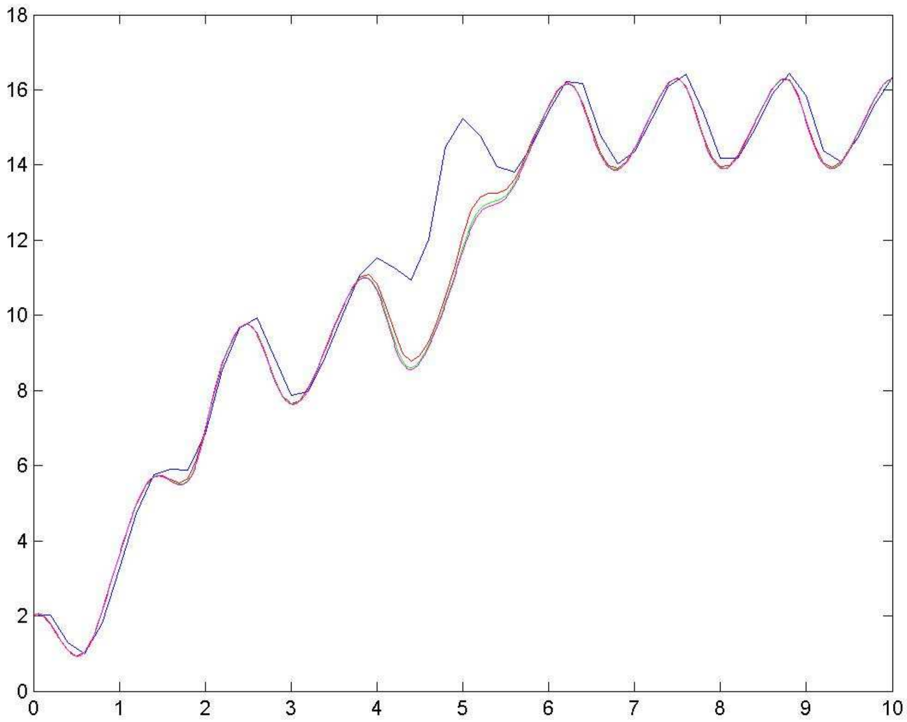
Konečná verzia smernice je potom k=(k1+2*k2+2*k3+k4)/6 = -1.023047321920166

Týmto smerom urobíme krok vpred:

y(0.2) = 2 + 0.2*-1.023047321920166 = 1.795390535615967

Výpočet naraz na celom intervale už bude komplikovanejší.

```
>> clear all
>> h=0.2; n=10/h+1; x=linspace(0,10,n);
>> f=inline('3-7*sin(5*x)+1.4*y.^0.5.*cos(y)'); y=2; t=0;
>> for k=2:n, k1=f(t,y(k-1)); yp=y(k-1)+0.5*h*k1; k2=f(t+0.5*h,yp);
yq=y(k-1)+0.5*h*k2; k3=f(t+0.5*h,yp); yr=y(k-1)+h*k3; k4=f(t+h,yr);
y(k)=y(k-1)+h*(k1+2*k2+2*k3+k4)/6; t=t+h; end, plot(x,y), hold on
>> h=0.1; n=10/h+1; x=linspace(0,10,n); y=2; t=0;
>> for k=2:n, k1=f(t,y(k-1)); yp=y(k-1)+0.5*h*k1; k2=f(t+0.5*h,yp);
yq=y(k-1)+0.5*h*k2; k3=f(t+0.5*h,yp); yr=y(k-1)+h*k3; k4=f(t+h,yr);
y(k)=y(k-1)+h*(k1+2*k2+2*k3+k4)/6; t=t+h; end, plot(x,y,'r')
>> h=0.05; n=10/h+1; x=linspace(0,10,n); y=2; t=0;
>> for k=2:n, k1=f(t,y(k-1)); yp=y(k-1)+0.5*h*k1; k2=f(t+0.5*h,yp);
yq=y(k-1)+0.5*h*k2; k3=f(t+0.5*h,yp); yr=y(k-1)+h*k3; k4=f(t+h,yr);
y(k)=y(k-1)+h*(k1+2*k2+2*k3+k4)/6; t=t+h; end, plot(x,y,'g')
>> h=0.01; n=10/h+1; x=linspace(0,10,n); y=2; t=0;
>> for k=2:n, k1=f(t,y(k-1)); yp=y(k-1)+0.5*h*k1; k2=f(t+0.5*h,yp);
yq=y(k-1)+0.5*h*k2; k3=f(t+0.5*h,yp); yr=y(k-1)+h*k3; k4=f(t+h,yr);
y(k)=y(k-1)+h*(k1+2*k2+2*k3+k4)/6; t=t+h; end, plot(x,y,'m')
```



Až na modrú čiaru, ktorej nepomôže nijaká metóda, to konverguje rýchlejšie.

Úlohy:

1. Kreslite do jedného obrázku štyri grafy získané všetkými uvedenými štyrmi metódami vždy pri jednom (pre všetkých rovnakom) delení intervalu x.

2. Počítajte rôznymi metódami pri rôznych hustotách delenia intervalu [1, 5] dif. rovnicu:

$$y' = (1+\sin(y))^{\cos(y)} + x-1 \;; \qquad y(1) = 2$$

3. Predošlú úlohu riešte pomocou matlabovskej ode45 alebo iných solverov, ktorých zoznam a stručnú charakteristiku získate cez help.

Návod:

Rovnicu y'= f(x,y) so zač. podm. y(a)=y0 riešime na intervale [a,b] takto:

Zadáme f ako inline objekt a pokračujeme:

```
[t,y] = ode45(f , [a, b], y0)
```

Výsledkom je vektor t, v ktorom sa nachádzajú body intervalu [a,b] – matlab si ho sám podelil na menšie časti, a hodnoty neznámej funkcie y v bodoch t.
Výslednú funkciu si môžeme dať vykresliť cez príkaz plot(t,y).

Ak nám ide len o obrázok a samotné hodnoty t,y priamo na nič nepotrebujeme, môžeme získať obrázok hneď, stačí iba nepýtať uloženie do t,y:

```
ode45(f , [a, b], y0)
```

Presnosť výsledku je štandardne 1e-6 (ohraničenie absolútnej chyby). Ak máme vzhľadom na presnosť vyššie ambície, musíme sa s nimi zdôveriť aj Matlabu.

Štandardné nastavenia matlabovských funkcií ode** zistíme príkazom odeset:

```
>> odeset
    AbsTol: [ positive scalar or vector {1e-6} ]
    RelTol: [ positive scalar {1e-3} ]
     ........
```

Zmenu absolútnej presnosti napr. na 1e-9 dosiahneme tým istým príkazom:

```
>> odeset('AbsTol', 1e-9)
ans =
        AbsTol: 1.000000000000000e-009
          ....
```