Numerično reševanje diferencialnih enačb

14 vaja

Borut Jurčič Zlobec[1]

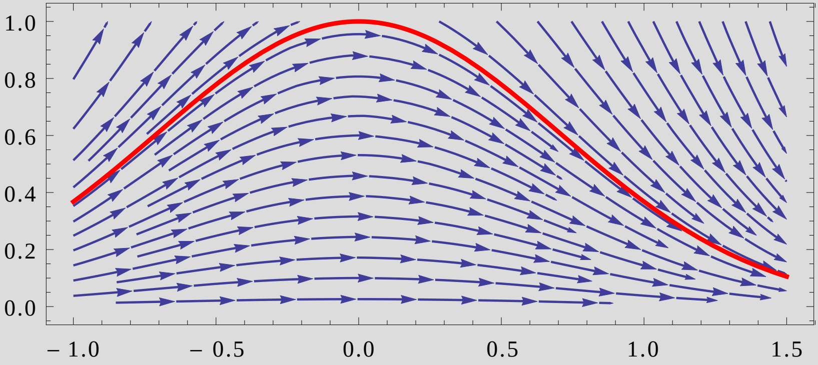
[1]Univerza v Ljubljani
Fakulteta za elektrotehniko,
Tržaška 25, 1000 Ljubljana

Numerične metode FE, Ljubljana, 14. januar 2013

Polje smeri

$$y' = f(x, y), \quad y(x_0) = y_0$$

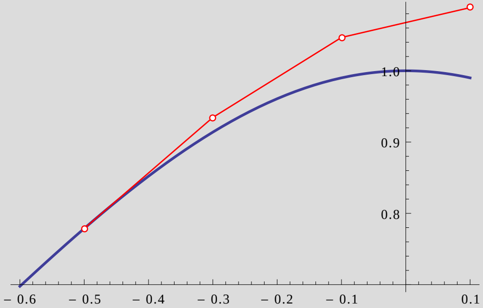
$$f(x, y) = -2xy, \quad y(0) = 1, \quad y = e^{-x^2}.$$



Eulerjeva metoda

$$x_0 = a, \quad y_0 = A, \quad h = x_{i+1} - x_i, \quad y_{i+1} = y_i + hf(x_i, y_i),$$

$$i = 0, \ldots, n, \quad n \in \mathbb{N}.$$



Program

```
% [y,x] = euler(f,I,ya,n); y'=f(x,y), y(a)=ya.
% Vhod: f:funkcija f(x,y), I:interval [a,b]
%   ya: zacetnih pogoji v a, n: stevilo podintervalov
% Izhod: x:delitev intervala I
%   y:  y(x) v notranjih tockah intervala in na robu
```

```
function [y,x] = euler(f,I,ya,n);
  h = (I(2)-I(1))/n; y = zeros(length(ya),n+1);
  y(:,1) = ya(:);
  x = linspace(I(1),I(2),n+1);
  for i = 2:n+1
    y(:,i) = y(:,i-1)+h*f(x(i-1),y(:,i-1));
  end;
```

Program

```
% [y,x] = euler(f,I,ya,n); y'=f(x,y), y(a)=ya.
% Vhod: f:funkcija f(x,y), I:interval [a,b]
%   ya: zacetnih pogoji v a, n: stevilo podintervalov
% Izhod: x:delitev intervala I
%   y:  y(x) v notranjih tockah intervala in na robu
```

```
function [y,x] = euler(f,I,ya,n);
  h = (I(2)-I(1))/n; y = zeros(length(ya),n+1);
  y(:,1) = ya(:);
  x = linspace(I(1),I(2),n+1);
  for i = 2:n+1
    y(:,i) = y(:,i-1)+h*f(x(i-1),y(:,i-1));
  end;
```

Program

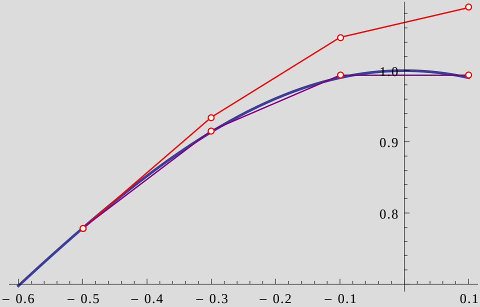
```
% [y,x] = euler(f,I,ya,n); y'=f(x,y), y(a)=ya.
% Vhod: f:funkcija f(x,y), I:interval [a,b]
%   ya: zacetnih pogoji v a, n: stevilo podintervalov
% Izhod: x:delitev intervala I
%   y:  y(x) v notranjih tockah intervala in na robu
```

```
function [y,x] = euler(f,I,ya,n);
  h = (I(2)-I(1))/n; y = zeros(length(ya),n+1);
  y(:,1) = ya(:);
  x = linspace(I(1),I(2),n+1);
  for i = 2:n+1
    y(:,i) = y(:,i-1)+h*f(x(i-1),y(:,i-1));
  end;
```

Modificirana Eulerjeva metoda

$$x_0 = a, \quad y_0 = A, \quad h = x_{i+1} - x_i, \quad i = 0, \dots, n, \quad n \in \mathbb{N},$$

$$y_{i+1} = y_i + hf(x_i + \frac{h}{2}, y_i + \frac{h}{2} f(x_i, y_i)).$$



Program

```
function [y,x] = modeuler(f,a,ya,n)
  h = (I(2)-I(1))/n;
  y = zeros(length(ya),n+1);
  y(:,1) = ya(:);
  x = linspace(I(1),I(2),n+1);
  for i = 2:n+1
    k1 = f(x(i-1),y(:,i-1));
    k2 = f(x(i-1)+h/2,y(:,i-1)+h/2*k1(:));
    y(:,i) = y(:,i-1)+h*k2(:);
  end;
```

Crank-Nicholson in leapfrog

$$y' = f(x, y)$$

$$I = [a, b], \quad x_0 = a, \quad y_0 = y(x_0), \quad h = \frac{b-a}{n}, \quad x_i = x_0 + ih$$

Crank-Nicholson

$$y_{i+1} = y_i + \frac{h}{2}\left(f(x_i, y_i) + f(x_{i+1}, y_{i+1})\right).$$

Leapfrog

$$y_{i+1} = y_{i-1} + 2hf(x_i, y_i).$$

Crank-Nicholson in leapfrog

$$y' = f(x, y)$$

$$I = [a, b], \quad x_0 = a, \quad y_0 = y(x_0), \quad h = \frac{b-a}{n}, \quad x_i = x_0 + ih$$

Crank-Nicholson

$$y_{i+1} = y_i + \frac{h}{2}\left(f(x_i, y_i) + f(x_{i+1}, y_{i+1})\right).$$

Leapfrog

$$y_{i+1} = y_{i-1} + 2hf(x_i, y_i).$$

Crank-Nicholson in leapfrog

$$y' = f(x, y)$$

$$I = [a, b], \quad x_0 = a, \quad y_0 = y(x_0), \quad h = \frac{b-a}{n}, \quad x_i = x_0 + ih$$

Crank-Nicholson

$$y_{i+1} = y_i + \frac{h}{2}\left(f(x_i, y_i) + f(x_{i+1}, y_{i+1})\right).$$

Leapfrog

$$y_{i+1} = y_{i-1} + 2hf(x_i, y_i).$$

Crank-Nicholson in leapfrog

$$y' = f(x, y)$$

$$I = [a, b], \quad x_0 = a, \quad y_0 = y(x_0), \quad h = \frac{b-a}{n}, \quad x_i = x_0 + ih$$

Crank-Nicholson

$$y_{i+1} = y_i + \frac{h}{2}\left(f(x_i, y_i) + f(x_{i+1}, y_{i+1})\right).$$

Leapfrog

$$y_{i+1} = y_{i-1} + 2hf(x_i, y_i).$$

Program Crank-Nicholson

```
function [y,x] = cranknicholson(f,I,ya,n);
  m = 20; e = 1e-10; h = (I(2)-I(1))/n;
  y = zeros(length(ya),n+1);
  y(:,1) = ya(:);
  x = linspace(I(1),I(2),n+1);
  for i = 2:n+1
    ff = f(x(i-1),y(:,i-1)); yy = y(:,i-1);
    for j = 1:m
      yit = y(:,i-1)+h/2*(ff+f(x(i),yy));
      if abs(yy-yit) < e break; end; yy = yit;
    end;
    y(:,i) = yit;
  end;
```

Program leapfrog

```
function [y,x] = leapfrog(f,a,ya,n)
  h = (a(2)-a(1))/n;
  y = zeros(length(ya),n+1);
  y(:,1) = ya(:);
  x = linspace(a(1),a(2),n+1);
  k1 = f(x(1),y(:,1));
  k2 = f(x(1)+h/2,y(:,1)+h/2*k1(:));
  y(:,2) = y(:,1)+h*k2(:);
  for i=3:n+1
        y(:,i) = y(:,i-2)+2*h*f(x(i-1),y(:,i-1));
  end;
```

Primeri

Reši diferencialno enačbo $y' = -2xy$, interval $I = [-1, 3]$, število podintervalov $n = 40$, začetni pogoj $y(-1) = \frac{1}{e}$, na vse štiri načine. Izpiši vrednosti za posamezno metodo v $x = 0$.

```
f=inline('-2*x*y','x','y');
a=[-1,3]; y0=exp(-1); n=40;
[y,x]=euler(f,a,y0,n); figure; plot(x,y);
[y,x]=modeuler(f,a,y0,n); figure; plot(x,y);
[y,x]=cranknicholson(f,a,y0,n); figure; plot(x,y);
[y,x]=leapfrog(f,a,y0,n); figure; plot(x,y);
```

```
0.00000, 1.03064; 0.00000, 1.00144;
0.00000, 0.99667; 0.00000, 1.00000;
```

Rezultati

Primerjava rezultatov: modra: Euler, zelena: Crank-Nicholson, rdeča: leapfrog in rumena: točna rešitev.