Laboratorijske vaje Numerične metode
7. Vaja

B. Jurčič Zlobec[1], A. Perne[1]

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

Numerične metode FE, Ljubljana, 22. november 2012

Airyeva diferencialna enačba

Reši robni problem $y''(x) + x\,y(x) = 0$, $y(A) = y_A$ in $y(B) = y_B$, kjer je $A = 0$, $B = 2$, $y(A) = 0$ in $y(B) = 1$.

Poiskali bomo približno rešitev. Interval $[A, B]$ razdelimo na $n = 20$ enakih delov. Približne vrednosti funkcije $y_i = y(x_i)$ v vozliščih

$$x_i = i\,h, \quad i = 1, \ldots, n-1, \quad h = \frac{B-A}{n},$$

dobimo kot rešitev sistema linearnih enačb:

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + x_i y_i = 0$$

$$i = 1, \ldots, n-1, \quad y_0 = y_A, \text{ in } y_n = y_B.$$

Linearni sistem enačb

Tridiagonalni sistem $My = b$. Razširjena matrika sistema: $[M|b]$

$$
\left[\begin{array}{ccccc|c}
-2 + x_1 h^2 & 1 & 0 & \cdots & 0 & -y_0 \\
1 & -2 + x_2 h^2 & 1 & \cdots & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & \cdots & 1 & -2 + x_{n-2} h^2 & 1 & 0 \\
0 & \cdots & 0 & 1 & -2 + x_{n-1} h^2 & -y_n
\end{array}\right]
$$

Reševanje tridiagonalnega sistema

Glavna diagonala matrike M je podana z vektorjem d,

$$d_i = -2 + x_i h^2, \quad i = 1, \dots, n-1.$$

Zgornja in spodnja vzporednica l in u.

$$u = (\overbrace{1, \dots, 1}^{n-2 \ krat}), \quad l = u.$$

Desna stran sistema

$$b = (-y_0, \overbrace{0, \dots, 0}^{n-3 \ krat}, -y_n)$$

Zapiši program, ki bo sprejel štiri vektorje d, u, l, b in bo podal rešitev v vektoju y. Sistem reši s pomočjo Gaussove eliminacije ne da bi zapisal polno obliko matrike M sistema.

Algoritem (tridiagolani Gauss + vzvratno vstavljanje)

Gaussova eliminacija:

$$\left[\begin{array}{ccccc|c} d_1 & u_1 & 0 & \cdots & 0 & b_1 \\ l_1 & d_2 & u_2 & \cdots & 0 & b_2 \\ \multicolumn{6}{c}{\dotfill} \\ 0 & \cdots & 0 & l_{n-1} & d_n & b_n \end{array}\right] \rightarrow \left[\begin{array}{ccccc|c} \delta_1 & u_1 & 0 & \cdots & 0 & \beta_1 \\ 0 & \delta_2 & u_2 & \cdots & 0 & \beta_2 \\ \multicolumn{6}{c}{\dotfill} \\ 0 & \cdots & 0 & 0 & \delta_n & \beta_n \end{array}\right]$$

$$\delta_1 = d_1, \quad \delta_k = d_k - u_{k-1} l_{k-1}/\delta_{k-1}, \quad k = 2, \ldots n$$
$$\beta_1 = b_1, \quad \beta_k = b_k - u_{k-1} l_{k-1}/\delta_{k-1}, \quad k = 2, \ldots n$$

Vzvratno vstavljanje:

$$y_n = \beta_n/\delta_n, \quad y_k = (\beta_k - u_k y_{k+1})/\delta_k, \quad k = n-1, \ldots, 1$$

Algoritem (tridiagolani Gauss + vzvratno vstavljanje)

Gaussova eliminacija:

$$\left[\begin{array}{ccccc|c} d_1 & u_1 & 0 & \cdots & 0 & b_1 \\ l_1 & d_2 & u_2 & \cdots & 0 & b_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & l_{n-1} & d_n & b_n \end{array}\right] \rightarrow \left[\begin{array}{ccccc|c} \delta_1 & u_1 & 0 & \cdots & 0 & \beta_1 \\ 0 & \delta_2 & u_2 & \cdots & 0 & \beta_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 0 & \delta_n & \beta_n \end{array}\right]$$

$$\delta_1 = d_1, \quad \delta_k = d_k - u_{k-1} l_{k-1}/\delta_{k-1}, \quad k = 2, \ldots n$$
$$\beta_1 = b_1, \quad \beta_k = b_k - u_{k-1} l_{k-1}/\delta_{k-1}, \quad k = 2, \ldots n$$

Vzvratno vstavljanje:

$$y_n = \beta_n/\delta_n, \quad y_k = (\beta_k - u_k\, y_{k+1})/\delta_k, \quad k = n-1, \ldots, 1$$

Algoritem (tridiagolani Gauss + vzvratno vstavljanje)

Gaussova eliminacija:

$$\left[\begin{array}{ccccc|c} d_1 & u_1 & 0 & \cdots & 0 & b_1 \\ l_1 & d_2 & u_2 & \cdots & 0 & b_2 \\ \multicolumn{6}{c}{\dotfill} \\ 0 & \cdots & 0 & l_{n-1} & d_n & b_n \end{array}\right] \rightarrow \left[\begin{array}{ccccc|c} \delta_1 & u_1 & 0 & \cdots & 0 & \beta_1 \\ 0 & \delta_2 & u_2 & \cdots & 0 & \beta_2 \\ \multicolumn{6}{c}{\dotfill} \\ 0 & \cdots & 0 & 0 & \delta_n & \beta_n \end{array}\right]$$

$$\delta_1 = d_1, \quad \delta_k = d_k - u_{k-1} l_{k-1}/\delta_{k-1}, \quad k = 2, \ldots n$$
$$\beta_1 = b_1, \quad \beta_k = b_k - u_{k-1} l_{k-1}/\delta_{k-1}, \quad k = 2, \ldots n$$

Vzvratno vstavljanje:

$$y_n = \beta_n/\delta_n, \quad y_k = (\beta_k - u_k\, y_{k+1})/\delta_k, \quad k = n-1, \ldots, 1$$

Program funkcija trisys

Definicija funkcije datoteka trisys.m

```
% x = trisys(u,d,l,b)
% u : nad, l : pod glavno diagonalo, dolzina je n-1
% d : na glavni diagonali, dolzina je n
% b : svobodni cleni, dolzina je n
% x : resitev sistema, dolzina je n
```

trisys

```
function x = trisys(u,d,l,b)
n = length(b);
for k=2:n,mult=l(k-1)/d(k-1);d(k)=d(k)-mult*u(k-1);
b(k)=b(k)-mult*b(k-1);end;
x(n) = b(n)/d(n);
for k=(n-1):-1:1,x(k)=(b(k)-u(k)*x(k+1))/d(k); end;
```

Program funkcija trisys

Definicija funkcije datoteka trisys.m

```
% x = trisys(u,d,l,b)
% u : nad, l : pod glavno diagonalo, dolzina je n-1
% d : na glavni diagonali, dolzina je n
% b : svobodni cleni, dolzina je n
% x : resitev sistema, dolzina je n
```

trisys

```
function x = trisys(u,d,l,b)
n = length(b);
for k=2:n,mult=l(k-1)/d(k-1);d(k)=d(k)-mult*u(k-1);
b(k)=b(k)-mult*b(k-1);end;
x(n) = b(n)/d(n);
for k=(n-1):-1:1,x(k)=(b(k)-u(k)*x(k+1))/d(k); end;
```

Program funkcija trisys

Definicija funkcije datoteka trisys.m

```
% x = trisys(u,d,l,b)
% u : nad, l : pod glavno diagonalo, dolzina je n-1
% d : na glavni diagonali, dolzina je n
% b : svobodni cleni, dolzina je n
% x : resitev sistema, dolzina je n
```

trisys

```
function x = trisys(u,d,l,b)
n = length(b);
for k=2:n,mult=l(k-1)/d(k-1);d(k)=d(k)-mult*u(k-1);
b(k)=b(k)-mult*b(k-1);end;
x(n) = b(n)/d(n);
for k=(n-1):-1:1,x(k)=(b(k)-u(k)*x(k+1))/d(k); end;
```

Gravni program

Priprava podatkov

```
n=20; x0=0; xend=2; y0=0; yend=1; h=(xend-x0)/n;
d=-2+(1:n-1)*h^3; l=ones(1,n-2); u=l;
b=zeros(size(d)); b(1)=-y0; b(end)=-yend;
```

Klic trisys

```
y=trisys(u,d,l,b);
```

Primerjava s točno rešitvijo

```
x=linspace(x0,xend,n+1); y=[y0,y,yend];
plot(x,y,'o',x,x.*hyperg_0F1(4/3,-x.^3/9)/...
hyperg_0F1(4/3,-8/9)/2);
```

Gravni program

Priprava podatkov

```
n=20; x0=0; xend=2; y0=0; yend=1; h=(xend-x0)/n;
d=-2+(1:n-1)*h^3; l=ones(1,n-2); u=l;
b=zeros(size(d)); b(1)=-y0; b(end)=-yend;
```

Klic trisys

```
y=trisys(u,d,l,b);
```

Primerjava s točno rešitvijo

```
x=linspace(x0,xend,n+1); y=[y0,y,yend];
plot(x,y,'o',x,x.*hyperg_0F1(4/3,-x.^3/9)/...
hyperg_0F1(4/3,-8/9)/2);
```

Gravni program

Priprava podatkov

```
n=20; x0=0; xend=2; y0=0; yend=1; h=(xend-x0)/n;
d=-2+(1:n-1)*h^3; l=ones(1,n-2); u=l;
b=zeros(size(d)); b(1)=-y0; b(end)=-yend;
```

Klic trisys

```
y=trisys(u,d,l,b);
```

Primerjava s točno rešitvijo

```
x=linspace(x0,xend,n+1); y=[y0,y,yend];
plot(x,y,'o',x,x.*hyperg_0F1(4/3,-x.^3/9)/...
hyperg_0F1(4/3,-8/9)/2);
```

Gravni program

Priprava podatkov

```
n=20; x0=0; xend=2; y0=0; yend=1; h=(xend-x0)/n;
d=-2+(1:n-1)*h^3; l=ones(1,n-2); u=l;
b=zeros(size(d)); b(1)=-y0; b(end)=-yend;
```

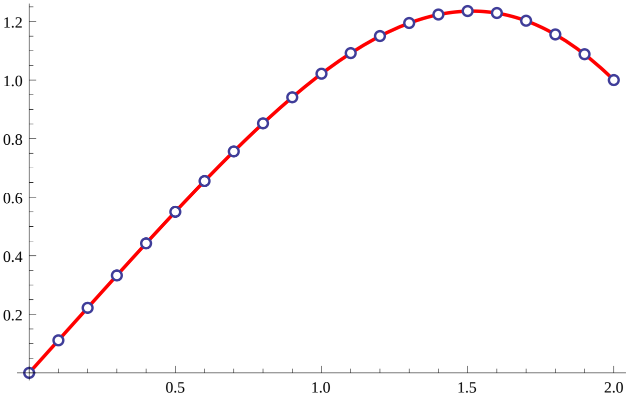
Klic trisys

```
y=trisys(u,d,l,b);
```

Primerjava s točno rešitvijo

```
x=linspace(x0,xend,n+1); y=[y0,y,yend];
plot(x,y,'o',x,x.*hyperg_0F1(4/3,-x.^3/9)/...
hyperg_0F1(4/3,-8/9)/2);
```

Slika 1



Slika: Točna in približna rešitev

Rešimo gornji sistem z levim deljenjem

Matrika sistema je redka. Sestavimo jo ukazom `M=sparse(I,J,V)`.

```
I=[1:n-1,2:n-1,1:n-2];
J=[1:n-1,1:n-2,2:n-1];
V=[-2+(1:n-1)*h^3,ones(1,2*n-4)];
M=sparse(I,J,V);
```

Rešimo sistem z levim deljenjem

```
b=zeros(n-1,1);
b(1)=-y0;
b(end)=-yend;
Y=M\b;
Y=[y0,Y',yend];
```

Rešimo gornji sistem z levim deljenjem

Matrika sistema je redka. Sestavimo jo ukazom M=sparse(I,J,V).

```
I=[1:n-1,2:n-1,1:n-2];
J=[1:n-1,1:n-2,2:n-1];
V=[-2+(1:n-1)*h^3,ones(1,2*n-4)];
M=sparse(I,J,V);
```

Rešimo sistem z levim deljenjem

```
b=zeros(n-1,1);
b(1)=-y0;
b(end)=-yend;
Y=M\b;
Y=[y0,Y',yend];
```

Rešimo gornji sistem z levim deljenjem

Matrika sistema je redka. Sestavimo jo ukazom M=sparse(I,J,V).

```
I=[1:n-1,2:n-1,1:n-2];
J=[1:n-1,1:n-2,2:n-1];
V=[-2+(1:n-1)*h^3,ones(1,2*n-4)];
M=sparse(I,J,V);
```

Rešimo sistem z levim deljenjem

```
b=zeros(n-1,1);
b(1)=-y0;
b(end)=-yend;
Y=M\b;
Y=[y0,Y',yend];
```