

Numerične metode

laboratorijske vaje
octave

Blaž Vizjak

Contents

1	Bisekcija	5
2	Regula falsi	5
3	Newtonova metoda	6
4	Newtonova metoda za sistem enačb	6
5	Iteracija	8
6	Nihajne kroglice	9
7	Strelanje v tarčo	9
8	Prehod žarka preko meje dveh snovi	10
9	Analiza vezja	11
10	Deltno pivotiranje	12
11	Poissonov problem	13
12	Robni problem v eni dimenziji	13
13	Jacobijeva in Gauss-Seidlova iteracija	14
14	Interpolacija polinoma	16
15	Aproksimacija	17
16	Robni problem in Simpsonova metoda	17

17	Koaksialen kabel	19
18	Večkrat uporabljene funkcije	20
19	Kubični zlepek	22
20	Trapezna in Simpsonova metoda	23
21	Rombergova metoda	24
22	Utež in vozle kvadrature formule	24
23	Gaussova kvadratura formula	25
24	Gaussova kvadratura formula za singularne integrale	25
25	Robni problem	26
26	Laplaceova enačba	28
27	Simpsonova metoda za parametrično krivuljo	28
28	Rombergova metoda za integral	29
29	Gaussova kvadratura formula in uteži	29
30	Dirichletov problem	30
31	Adams-Bashforth 2. reda	31
32	Runge-Kutta 4. reda	31
33	Funkcije	32

33.1	Trapezna metoda	32
33.2	Romberg	32
33.3	Runge-Kutta 4	33

1 Bisekcija

Z metodo bisekcije rešite nelinearno enačbo

$$x^3 - 0.536x^2 - 0.536 = 0.$$

Za začetni interval vzamite $[0, 2]$. Vpišite peto iteracijo.

```
1 % VHODNE SPREMENLJIVKE
2 f = inline('x^3-0.536*x^2-0.536','x'); % funkcija
3 iter = 5; % stevilo iteracij
4 a = 0; % leva robna točka
5 b = 2; % desna robna točka
6
7 % FUNKCIJA
8 fa=f(a);
9 fb=f(b);
10
11 for i=1:iter
12     c=(a+b)/2;
13     fc=f(c);
14
15     if sign(fa*fc)<0
16         b=c;
17         fb=fc;
18     else
19         a=c;
20         fa=fc;
21     end;
22 end;
23
24 printf('Rezultati metode bisekcije\n c = %0.10f\n f(c) = %0.10f\n stevilo ...
        iteracij: %d\n',c,fc,i);
```

2 Regula falsi

Z metodo regula falsi rešite nelinearno enačbo

$$x^3 + 3.70x^2 - 10 = 0.$$

Za začetni interval vzamite $[1, 2]$. Vpišite četrto iteracijo.

```
1 % VHODNE SPREMENLJIVKE
2 f = inline('x^3+3.7*x^2-10','x'); % funkcija
3 iter = 4; % stevilo iteracij
4 a = 1;
5 b = 2;
6
7 % FUNKCIJA
8 fa=f(a);
9 fb=f(b);
10
11 for i=1:iter
```

```

12     c=a-fa*(b-a)/(fb-fa);
13     fc=f(c);
14
15     if sign(fa*fc)<0
16         b=c;
17         fb=fc;
18     else
19         a=c;
20         fa=fc;
21     end;
22 end;
23
24 printf('Rezultati metode regula falsi\n c = %f\n f(c) = %f\n stevilo ...
        iteracij: %0.0f\n',c,fc,i);

```

3 Newtonova metoda

S pomočjo Newtonove metode poiščite rešitev enačbe:

$$0.70e^{-x} - x = 0.$$

Izberite začetni približek $x_0 = 0.03$. Iteracijo ponavljajte, dokler ni absolutna vrednost funkcije v trenutni iteraciji manjša kot $1e-8$. Koliko iteracij je bilo potrebnih?

```

1  % VHODNE SPREMENLJIVKE
2  f=inline('0.7*exp(-x)-x','x');
3  df=inline('0.7*(-1)*exp(-x)-1','x');
4  iter=100;
5  x0=0.03;
6  eps=1e-8;
7
8  % FUNKCIJA
9  for i=1:iter
10     x1=x0-f(x0)/df(x0);
11     x0=x1;
12
13     if eps>0
14         if abs(f(x0))<eps
15             break
16         end;
17     end;
18 end;
19
20 printf('Rezultati Newtonove metode\n x0 = %f\n stevilo iteracij: %0.0f\n',x1,i);

```

4 Newtonova metoda za sistem enačb

Z Newtonovo metodo rešite sistem nelinearnih enačb

$$x^2 + 2y^2 + z^2 = 4, 2x^2 + y^2 + 2z^2 = 5, x^2 + y^2 + z^2xy = 1.$$

Začetni približek je

$$[xyz] = [2.49, 2.87, 3.95].$$

Vpišite tretjo iteracijo na 6 decimalnih mest natančno.

```
1 % VHODNE SPREMENLJIVKE
2 f=inline(' [x^2+2*y^2+z^2-4; 2*x^2+y^2+2*z^2-5; x^2+y^2+z^2-x-y-1] ','x','y','z');
3 df=inline(' [2*x,4*y,2*z; 4*x,2*y,4*z; 2*x-1,2*y-1,2*z] ','x','y','z'); % v ...
    vsaki vrstici je odvod vsake enacbe, po stolpcih pa za vsako spremenljivko
4 x0=[2.49; 2.87; 3.95];
5 iter = 3;
6
7 % FUNKCIJA
8 for i=1:iter
9     x1=x0-df(x0(1),x0(2),x0(3))\f(x0(1),x0(2),x0(3));
10    x0=x1;
11 end;
12 printf(' x = %0.15f\n y = %0.15f\n z = %0.15f\n',x1);
```

Z Newtonovo metodo rešite sistem enačb

$$e^{-1.449y} - 1.209 - 1.964x = 0, \quad 1.076x + 1.016 - 1.002y = 0.$$

Začetni približek je $[x, y] = [1, 1]$.

```
1 % VHODNE SPREMENLJIVKE
2 % x = X(1), y = X(2)
3 F=inline(' [exp(-1.449*X(2))-1.209-1.964*X(1); ...
    1.076*X(1)+1.016-1.002*X(2)] ','X');
4
5 % matrika odvodov
6 % [dx/dx, dx/dy; dy/dx, dy/dy]
7 J=inline(' [-1.964, -1.449*exp(-1.449*X(2)); 1.076, -1.002] ','X');
8 % enacbi odvajani po x/X(1) in y/X(2)
9
10 % zacetni priblizek
11 X=[1;1];
12
13 % stevilo iteracij
14 iter = 5;
15
16 % FUNKCIJA
17 for i=1:iter
18     X1=X-J(X)\F(X);
19     X=X1;
20 end;
21
22 printf(' x = %0.15f\n y = %0.15f\n',X(1),X(2));
```

Z Newtonovo metodo rešite sistem nelinearnih enačb

$$-2 \cos x - 3x + 2y - 6 = 0, \quad y^2 + 4x - 2y + 1 = 0.$$

Začetni približek je $[x, y] = [0, 0]$. Vpišite rešitev ter število potrebnih iteracij, da neskončna norma razlike dveh zaporednih iteracij pade pod 10^{-12} .

```

1 % VHODNE SPREMENLJIVKE
2 F=inline('[-2*cos(X(1))-3*X(1)+2*X(2)-6;(X(2))^2+4*X(1)-2*X(2)+1'],'X');
3
4 % matrika odvodov
5 % [dx/dx, dx/dy; dy/dx, dy/dy]
6 J=inline('[2*sin(X(1))-3,2;4,2*X(2)-2'],'X');
7
8 % zacetni priblizek
9 X=[2.49;0];
10
11 % stevilo iteracij
12 iter = 5;
13
14 % epsilon
15 eps = 1e-12;
16
17 % FUNKCIJA
18 for i=1:iter
19     X1=X-J(X)\F(X);
20     norm1=norm(X);
21     norm2=norm(X1);
22     X=X1;
23
24     if i>1 && norm1-norm2<eps
25         break;
26     end;
27 end;
28
29 printf('x = %0.15f\nty = %0.15f\nstevilo iteracij: %d\nda norma razlike dveh ...
        zaporednih iteracij pade pod %.0e\n',X(1),X(2),i,eps);

```

5 Iteracija

S pomočjo iteracije $x_{n+1} = f(x_n)$, kjer je $f(x) = 0.70e^{-x}$ in $x_0 = 0.00$, poiščite rešitev enačbe $f(x) = x$. Zapišite vrednost x_6 . Ali je zaporedje konvergentno? Določite limito na 10 decimalnih mest natančno.

```

1 % VHODNE SPREMENLJIVKE
2 f=inline('0.7*exp(-x)','x');
3 x0=0;
4 eps=10e-8;
5 iter=6;
6
7 % FUNKCIJA
8 for i=1:iter
9     x1=f(x0);
10    if abs(x1-x0)<eps
11        break;
12    end;
13    x0=x1;
14 end;
15 printf(' Limita: x%d = %0.15f\n',i,x1);

```

6 Nihajne kroglice

Na dveh lahkih neprevodnih vrvicah dolžine $l = 12$ cm s skupnim prijemališčem visita kroglici enakih mas $m = 1$ g. Kroglici sta prevodni, majhnega polmera in se odklonita v odvisnosti od naboja Q , ki ga vsaka nosi. Kolikšen je kot α med vrvicama, če sta kroglici naelektreni s $Q = 2.90 \cdot 10^{-8}$ As naboja? Enačbo rešite z metodo bisekcije. Kot izrazite v kotnih stopinjah. Konstante: $\varepsilon_0 = 8.8542 \cdot 10^{-12}$ As/Vm, $g = 9.8067$ m/s².

```
1 % VHODNE SPREMENLJIVKE
2 m = 1e-3; % masa kroglice
3 l = 0.12; % dolzina vrvice
4 Q = 2.9*1e-8; % naboj kroglice [As]
5 eps = 1e-8; % epsilon okolica
6 kot = 20; % priblizhen kot v stopinjah
7
8 % KONSTANTE
9 epsilon0=8.854184817*1e-12;
10 g = 9.80665;
11 k = Q^2/(16*pi*epsilon0*m*g*l^2);
12
13 % FUNKCIJA
14 fs='u.^3-k*u.^2-k';
15 dfs='3*u.^2-2*k*u';
16 alpha=kot*pi/180;
17 u=tan(alpha/2);
18
19 for i=1:100
20     u1=u-eval(fs)/eval(dfs);
21     if abs(u1-u)<eps
22         break
23     end;
24     u=u1;
25 end;
26 alfa = 2*atan(u)/pi*180;
27
28 % RESITEV
29 printf(' kot alfa = %0.15f stopinj\n stevilo iteracij = %0d\n',alfa,i);
```

7 Strelanje v tarčo

Tarča se nahaja v točki (3.4, 0). Določi začetno hitrost $\vec{v}_0$ izstrelka (kot 45°), do bo ta zadel tarčo. Pot izstrelka je podana s sistemom diferencialnih enačb.

$$\ddot{x}(t) + \beta \dot{x}(t) = 0, \ddot{y}(t) + \beta \dot{y}(t) = -g,$$

z začetnimi pogoji $x(0) = y(0) = 0$, $\dot{x}(0) = \dot{y}(0) = v_0/\sqrt{2}$. Količnik upora $\beta = 2$.

```
1 % VHODNE SPREMENLJIVKE
2 zeljenatarca=[3.4,0]; % tocka tarce, vedno na abscisi!
3 tarca=zeljenatarca(1); % upostevanje samo abscise
4 t=[0.5,1];
5 v=4; % zacetni priblizek hitrosti
6 eps=1e-8;
```

```

7  beta=2; % definirana je v strell !!!
8
9  % FUNKCIJA
10 hitrost=[v,v+0.5];
11
12 for i=1:100
13     kiks=[strell(hitrost(1),t)-tarca, strell(hitrost(2),t)-tarca];
14     s=hitrost(2)-kiks(2)*diff(hitrost)/diff(kiks);
15     hitrost(1)=hitrost(2); hitrost(2)=s;
16     if abs(diff(hitrost))<eps
17         break;
18     end;
19 end;
20
21 speed=mean(hitrost);
22
23 printf(' v = %0.15f\n iter = %d\n zacetna hitrost strela v tarco (%0.1f,0): ...
        %0.3f m/s (pod kotom 45 stopinj)\n', speed, i, tarca, speed);

```

8 Prehod žarka preko meje dveh snovi

Dve snovi Ω_1 in Ω_2 različnih optičnih lastnosti, izpolnjujeta dovolj veliko področje v ravnini v okolici koordinatnega izhodišča in se stikata v paraboli

$$y = (x - 1.500)^2/10 - 0.300$$

Hitrost svetlobe v snovi Ω_1 je enaka $c_1 = 1$, medtem ko je v snovi Ω_2 enaka $c_2 = 1.541$. Snov Ω_1 prekriva področje nad parabolo, medtem ko snov Ω_2 prekriva področje pod njo. Kje preseka parabolo svetlobni žarek, ki potuje od točke $A = (0.000, 1.000)$ do točke $B = (3.000, -1.000)$. Svetlobni žarek izbere časovno najkrajšo pot. Vpiši absciso točke, kjer žarek prečka mejo.

```

1  % VHODNE SPREMENLJIVKE
2  f=inline('(x-1.5)^2/10 - 0.3','x'); % parabola sticisca snovi
3  A=[0,1]; % zacetna tocka
4  B=[3,-1]; % koncna tocka
5  c1=1; % hitrost svetlobe v snovi 1
6  c2=1.541; % hitrost svetlobe v snovi 1
7  eps=1e-8;
8  iter=100;
9
10 % FUNKCIJA
11 T=[A;B]; % matrika A in B
12 c=[c1,c2];
13 t=inline('norm(T(1,:)-[x,f(x)])/c(1)+norm([x,f(x)]-T(2,:))/c(2)','x');
14 dt=inline('(t(x+eps)-t(x-eps))/(2*eps)','x');
15 x1=T(1,1); x2=T(2,1);
16
17 for i=1:iter
18     xi=x2-dt(x2)*(x2-x1)/(dt(x2)-dt(x1));
19     x1=x2; x2=xi;
20
21     if abs(x1-x2)<eps
22         break
23     end;
24 end;
25

```

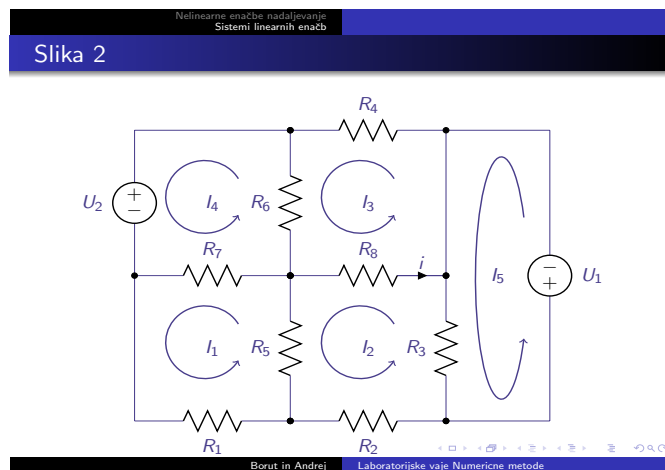
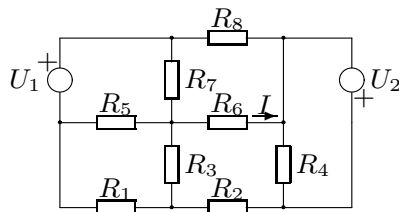
```

26 % RESITEV
27 printf(' Zarek seka mejo v tocki (Tx,Ty):\n Tx = %0.15f\n Ty = ...
    %0.15f\n',xi,f(xi));

```

9 Analiza vezja

V vezju na sliki,



določite tok I v A (amperih). Uporabite Gauss-Jordanovo metodo. Podatki: $U_1 = 10$ V, $U_2 = 20$ V in vektor uporov

$$R = [1.66, 10.00, 10.66, 3.59, 10.32, 9.09, 6.63, 1.53]$$

podanih v Ω .

```

1 % uporovni vektor
2 R=[1.66,10,10.66,3.59,10.32,9.09,6.63,1.53];
3
4 % pazi na vrstni red uporov glede na Sliko 2
5 r=[R(1),R(2),R(4),R(8),R(3),R(7),R(5),R(6)];
6
7 % napetostni generatorji
8 % [gen na zanki 1; gen za zanki 2; ...]
9 U=[0; 0; 0; -10; -20];
10
11 % uporovna matrika (glede na Sliko 2)
12 % [vpliv 1. zanke na 1. zanko, vpliv 2. zanke na 1. zanko...;
13 % vpliv z1 na z2, vpliv z2 na z2, vpliv z3 na z2, vpliv z4 na z2, vpliv z5 ...
    na z2...;...;...;...]
14
15 R=[r(1)+r(5)+r(7),-r(5),0,-r(7),0;
16     -r(5),r(2)+r(3)+r(5)+r(8),-r(8),0,-r(3);
17     0,-r(8),r(4)+r(6)+r(8),-r(6),0;
18     -r(7),0,-r(6),r(6)+r(7),0;
19     0,-r(3),0,0,r(3)];
20

```

```

21 I=R\U; % R*I=U
22 printf('I = %0.15f A\n',I(3)-I(2));

```

10 Deltno pivotiranje

Dan je sistem linearnih enačb $Ax = b$, kjer je

$$A = \begin{bmatrix} 7 & 3 & 1 & 1 & 7 \\ 4 & 3 & 8 & 0 & 5 \\ 8 & 0 & 9 & 5 & 8 \\ 0 & 5 & 5 & 2 & 8 \\ 0 & 4 & 8 & 8 & 0 \end{bmatrix}, b = \begin{bmatrix} 6 \\ 3 \\ 4 \\ 6 \\ 7 \end{bmatrix}.$$

Sistem rešite z Gauss - Jordanovo eliminacijo z delnim pivotiranjem in vpišite tretji pivot ter evklidsko normo rešitve sistema. Pivot je po absolutni vrednosti največje število v tekočem stolpcu od vključno diagonalnega elementa navzdol, ki ga pri delnem pivotiranju premaknemo na diagonalo.

```

1  % VHODNE SPREMENLJIVKE
2  A=[7 3 1 1 7; 4 3 8 0 5; 8 0 9 5 8; 0 5 5 2 8; 0 4 8 8 0];
3  b=[6; 3; 4; 6; 7];
4
5  n=length(b); %velikost vektorja b oz. visina matrike
6  m=size(A);
7  R=[A b]
8
9  printf('velikost matrike A je %0.0fx%0.0f\n\n',m(1),m(2));
10 for i=1:n
11     printf('—— nov krog ——\n \n');
12
13     [pivot k] = max(abs(R(i:end,i))); %trenutni pivot
14     printf('pivot%0.0f = %f\n',i,pivot);
15     k=k+i-1 %pravilen k za [A|b] matriko
16
17     t=R(i,:); %vrstica, kjer zelimo enico na mestu pivota
18     R(i,:) = R(k,:); %vrstica s pivotom
19     printf('\nmatrika s pivotom na pravem mestu\n');
20     R(k,:) = t %zamenjava vrstic
21
22     printf('pridelava enice na diagonalni\n');
23     R(i,:) = R(i,+)/R(i,i) %deljenje celotne vrstice s pivotom za pridelek enice
24
25     for j=(i+1):n
26         printf('urejanje drugih vrstic (vrstica %d)\n',j);
27         R(j,:) = R(j,:) - R(j,i)*R(i,:)
28     end;
29 end;

```

11 Poissonov problem

Poiščite približno rešitev Poissonovega robnega problema v eni dimenziji:

$$\Delta u = -1, \quad u(0) = 0.440, \quad u(1) = 0.400.$$

Interval $[0, 1]$ razdelite na 9 enako dolgih podintervalov. Vpišite največjo izračunano približno vrednost funkcije.

```
1 % VHODNE SPREMENLJIVKE
2 d = [];
3 n = 9;
4 x0 = 0; xend = 1; % interval
5 y0 = 0.44; yend = 0.4; % zacetna pogoja
6
7 % FUNKCIJA
8 h = (xend-x0)/n;
9 d = -2*ones(1,n-1);
10 l = ones(1,n-2);
11 u = 1;
12 b = -h^2*ones(1,n-1);
13 b(1) = b(1)-y0;
14 b(end) = b(end)-yend;
15
16 % Klic funkcije trisys
17 y = trisys(u,d,l,b);
18
19 % RESITEV
20 printf('max(u) = %0.8f\n',max(y))
```

12 Robni problem v eni dimenziji

Rešite robni problem v eni dimenziji:

$$y'' - y = x, \quad y'(0) = 1.95, \quad y'(1) = 3.70.$$

Interval $[0, 1]$ razdelite na $n = 5$ enako dolgih podintervalov. Drugi odvod aproksimirajte s končnimi razlikami

$$\frac{d^2 y}{dx^2}(x_i) \approx \frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1}))}{h^2},$$

prvega v levem krajišču z izrazom $y'(x_0) \approx \frac{1}{2h}(-3y_0 + 4y_1 - y_2)$, prvega v desnem krajišču pa z izrazom $y'(x_n) \approx \frac{1}{2h}(3y_n - 4y_{n-1} + y_{n-2})$, kjer je h dolžina podintervala, zapišite tridiagonalni sistem in ga rešite. Vpišite največjo izračunano vrednost funkcije.

```
1 n = 5;
2 x0 = 0;
3 xend = 1;
4 h = (xend-x0)/n; % interval
5
6 M=[-3 4 -1 0 0 0; 1 -2-h^2 1 0 0 0; 0 1 -2-h^2 1 0 0; 0 0 1 -2-h^2 1 0; 0 0 ...
    0 1 -2-h^2 1; 0 0 0 1 -4 3];
```

```

7 b=[1.95*2*h; h*h^2; 2*h*h^2; 3*h*h^2; 4*h*h^2; 3.7*2*h];
8
9 % RESITEV
10 printf('y_max = %0.8f\n',max(M\b));

```

Rešite robni problem v eni dimenziji:

$$y'' - 3y' + 2y = \cos x, \quad y(0) = 1.95, \quad y(1) = 3.70,$$

kjer interval $[0,1]$ razdelite na $n = 9$ enako dolgih podintervalov. Vpišite evklidsko normo rešitve (vključno z robnima vrednostima).

```

1 % VHODNE SPREMENLJIVKE
2 % interval [a,b]
3 a = 0;
4 b = 1;
5
6 % zacetna pogoja y(a), y(b)
7 ya = 1.95;
8 yb = 3.70;
9
10 % podintervali
11 n = 9;
12 n = n-1;
13
14 % robni problem
15 p = -3;
16 q = 2;
17 r = cos(x);
18
19 % FUNKCIJA
20 % robniproblem(p,q,r,a,ya,n)
21 % y'' + p*y' + q*y = r
22 % a - interval, ya - zacetna pogoja, n - podintervali
23 [y, x] = robniproblem(p, q, r, [a, b], [ya, yb], n);
24
25 norm = norm(y);
26
27 % robni vrednosti ne stejeta;
28 y = y(:, 2:end-1);
29
30 printf('Najvecja vrednost funkcije (robni vrednosti ne stejeta):\n y_max = ...
        %0.15f\nEvklidska norma resitve (vkljucno z robnima vrednostima):\n ...
        norma = %0.15f\n',max(y),norm);

```

13 Jacobijeva in Gauss-Seidlova iteracija

Dan je sistem $Ax = b$, kjer je

$$A = \begin{bmatrix} 2 & 1 \\ 2 & 2 \end{bmatrix}, b = \begin{bmatrix} 2 \\ 2 \end{bmatrix}.$$

Sistem rešite z Jacobijevo in Gauss-Seidlovo iteracijo z začetnim približkom $x_0 = \begin{bmatrix} 8 \\ 9 \\ 1 \\ 9 \end{bmatrix}$. Vpišite evklidsko normo druge iteracije za obe metodi.

```

1 % VHODNE SPREMENLJIVKE
2 A = [2 1; 2 2];
3 b = [2; 2];
4 x0 = [8/9; 1/9]; % zacetni priblizek
5 n = 2; % zeljena norma
6
7 % FUNKCIJA
8 % JACOBIJEVA METODA
9 D = diag(diag(A));
10 L = tril(A,-1);
11 U = triu(A,1);
12 S = D\(-L-U);
13 c = D\b;
14
15 x=x0;
16 for i=1:n
17     x = S*x + c;
18 end;
19
20 printf("Jacobijeva metoda\n");
21 x
22 printf("%d. norma: %0.15f\n",n,norm(x));
23
24
25 % GAUSS-SEIDLOVA METODA
26 S = (L+D)\(-U);
27 c = (L+D)\b;
28
29 x=x0;
30 for i=1:n
31     x = S*x + c;
32 end;
33
34 printf("\nGauss-Seidlova metoda\n");
35 x
36 printf("%d. norma: %0.15f\n",n,norm(x));

```

Jacobijeva in Gauss-Seidlova iteracija konvergirata, če je matrika A strogo diagonalno dominantna. Preverite, ali dan sistem ustreza temu pogoju, rešite ga z obema iteracijama in vpišite evklidsko normo tretje iteracije. Začetni približek je vektor 0. Diagonalo d matrike sistema A , vzporednici u nad njo in l pod njo, ter desno stran sistema b dobite s spodnjim zaporedjem ukazov.

```

n=8;
d=1.93*ones(1,n);
u=ones(1,n-1);
l=ones(1,n-1);
b=2+sin(1:n);

```

```

1 % VHODNE SPREMENLJIVKE
2 n = 8;
3 d = 1.93*ones(1,n); % ones(vrstica,stolpec)
4 u = ones(1,n-1);
5 l = ones(1,n-1);
6 b = 2+sin(1:n);
7
8 % zacetni priblizek
9 x = zeros(size(d));

```

```

10
11 u1 = u./d(1:end-1);
12 l1 = l./d(2:end);
13 b1 = b./d;
14 x1 = x;
15
16 % stevilo iteracij
17 k = 3;
18
19 % JACOBIJEVA METODA
20 for j=1:k
21     x1(1) = b1(1) - u1(1)*x(2);
22
23     for i=2:n-1
24         x1(i) = b1(i) - u1(i)*x(i+1) - l1(i-1)*x(i-1);
25     end;
26
27     x1(n) = b1(n) - l1(n-1)*x(n-1);
28     x = x1;
29 end;
30
31 printf("Norma Jacobijeve metode: %0.15f\n",norm(x1));
32
33 % resetiranje x-a
34 x=zeros(size(d));
35
36 % GAUSS-SEIDLOVA METODA
37 for j=1:k
38     x(1) = b1(1) - u1(1)*x(2);
39
40     for i=2:n-1
41         x(i) = b1(i) - u1(i)*x(i+1) - l1(i-1)*x(i-1);
42     end;
43
44     x(n) = b1(n) - l1(n-1)*x(n-1);
45 end;
46
47 printf("Norma Gauss-Seidlove metode: %0.15f\n",norm(x));
48 printf('stevilo iteracij: %d\n',k);

```

14 Interpolacija polinoma

Izračunajte približno vrednost funkcije v točki $x_0 = 1.45$ tako, da skozi dane točke $(0, 1.82)$, $(1, 0.99)$, $(2, -0.75)$ in $(3, -1.79)$ interpolirate kubičen polinom

$$y = Ax^3 + Bx^2 + Cx + D.$$

```

1 % VHODNE SPREMENLJIVKE
2 x=[0 1 2 3]; % x komponente danih tock
3 y=[1.82 0.99 -0.75 -1.79]; % y komponente danih tock
4 x0=1.45; % zacetni priblizek
5
6 % FUNKCIJA
7 n=length(x); % velikost x-a
8 d=Newtons_Interp(x,y);
9 dx=x0-x;

```

```

10 y0=d(end);
11
12 for i=n-1:-1:1
13     y0=y0*dx(i)+d(i); % racunanje polinoma nazaj
14 end;
15
16 printf('y0 = %0.8f\n',y0);

```

15 Aproksimacija

Poiščite parametra a_1 in a_2 tako, da bo funkcija

$$f(x, a) = a(1)\sqrt{x} + a(2)/x$$

aproksimirala podatke v smislu najmanjših kvadratov.

$x = [1.00 \ 1.11 \ 1.22 \ 1.33 \ 1.44 \ 1.56 \ 1.67 \ 1.78 \ 1.89 \ 2.00];$

$y = [3.08 \ 3.17 \ 3.29 \ 3.36 \ 3.47 \ 3.53 \ 3.65 \ 3.71 \ 3.75 \ 3.86];$

Vpišite vsoto parametrov $a_1 + a_2$.

```

1 % VHODNE SPREMENLJIVKE
2 x=[1.00 1.11 1.22 1.33 1.44 1.56 1.67 1.78 1.89 2.00];
3 y=[3.08 3.17 3.29 3.36 3.47 3.53 3.65 3.71 3.75 3.86];
4
5 % FUNKCIJA
6 x=x(:); % pretvorba v vektor
7 y=y(:); % pretvorba v vektor
8 A=[sqrt(x), 1./x]; % y=a1*sqrt(x)+a2*1/x, pazi na piko!
9 a=(A'*A)\(A'*y);
10 %printf('a=(%f,%f)\n',a); % nekvadratna matrika, polnega ranga
11 a=A\y;
12 %printf('a=(%f,%f)\n',a); % pravilen nacin
13 d=polyfit(x,y,1);
14 %printf('d=(%f,%f)\n',d);
15 vsota=a(1)+a(2);
16 printf('a1 = %0.8f\n a2 = %0.8f\n',a(1),a(2));
17 printf('a1+a2 = %0.8f\n',vsota);

```

16 Robni problem in Simpsonova metoda

Rešite robni problem

$$y''(x) = \sin(x), y(x_0) = \frac{6}{19}, y'(x_n) = 0.$$

Drugi odvod nadomestite z izrazom

$$y''(x_i) \approx \frac{1}{h^2}(y_{i-1} - 2y_i + y_{i+1}),$$

prvega v desnem krajišču pa z izrazom

$$y'(x_n) \approx \frac{1}{2h}(3y_n 4y_{n1} + y_{n2}).$$

Vzemimo, da je korak $h = 1/2$. Izračunajte integral rešitve na intervalu $[0, 4]$ po Simpsonovi metodi.

Simpsonova metoda

$$\int_a^b f(x)dx \approx \frac{h}{3} \left[f(x_0) + 2 \sum_{j=1}^{n/2-1} f(x_{2j}) + 4 \sum_{j=1}^{n/2} f(x_{2j-1}) + f(x_n) \right]$$

```

1  x0=0;
2  xn=4;
3  h=1/2;
4  n=(xn-x0)/h;
5  y0=6/19;
6
7  M=[-2 1 0 0 0 0 0 0;
8      1 -2 1 0 0 0 0 0;
9      0 1 -2 1 0 0 0 0;
10     0 0 1 -2 1 0 0 0;
11     0 0 0 1 -2 1 0 0;
12     0 0 0 0 1 -2 1 0;
13     0 0 0 0 0 1 -2 1;
14     0 0 0 0 0 1 -4 3];
15  b=[h^2*sin(h)-y0;
16     h^2*sin(2*h);
17     h^2*sin(3*h);
18     h^2*sin(4*h);
19     h^2*sin(5*h);
20     h^2*sin(6*h);
21     h^2*sin(7*h);
22     0];
23
24  Y=M\b;
25
26  % Simpsonova metoda
27  S=0;
28
29  for i=1:2:(n-1)
30     x=4*Y(i);
31     S=S+x;
32  end;
33
34  for i=2:2:(n-2)
35     x=2*Y(i);
36     S=S+x;
37  end;
38
39  S=h/3*(y0+S+Y(8));
40  printf('S = %0.10f\n',S);

```

17 Koaksialen kabel

V koaksialnem kablu je polmer žile $a = 2$ mm, polmer plašča pa $b = 5$ mm. Plašč je ozemljen (na potencialu $V(b) = 0$ V), žila pa na potencialu $V(a) = 120$ V. Za potencial $V(\rho)$ v kablu velja enačba

$$\frac{\partial^2 V}{\partial \rho^2} + \frac{1}{\rho} \frac{\partial V}{\partial \rho} = 0.$$

Odvide aproksimiramo s končnimi razlikami

$$\frac{\partial^2 V}{\partial \rho^2} \approx \frac{V_{i+1} - 2V_i + V_{i-1}}{h^2}$$

in

$$\frac{\partial V}{\partial \rho} \approx \frac{V_{i+1} - V_{i-1}}{2h}$$

kjer je $i \in [1, N]$ in $h = \frac{ba}{N+1}$ (izberite $N = 10$). V takem zapisu sta $V_0 = V(a)$ in $V_{N+1} = V(b)$. Enačba pri $i = 2$ je

$$\frac{V_3 - 2V_2 + V_1}{h^2} + \frac{1}{\rho_i} \frac{V_3 - V_1}{2h} = 0.$$

Če zapišemo enačbo pri vsakem od polmerov ρ_i , $i \in [1, N]$ ter V_0 in V_{N+1} kot znani količini postavimo na desno stran sistema, dobimo tridiagonalni sistem enačb za neznane količine V_i . Izračunajte vrednost V_5 . Koliko je absolutna vrednost razlike med izračunano in točno vrednostjo, ki jo poznamo iz OE I, t.j.

$$V(\rho) - V(b) = U \frac{\ln \frac{\rho}{b}}{\ln \frac{a}{b}}; \quad \rho \in [a, b],$$

kjer je $U = V(a) - V(b)$ napetost med žilo in plaščem?

```
1 % VHODNE SPREMENLJIVKE
2 % polmer in potencial zile
3 a = 2e-3; % a = 2mm
4 ya = 120; % V(a) = 120V
5
6 % polmer in potencial plasca
7 b = 5e-3; % b = 5mm
8 yb = 0; % V(b) = 0V
9
10 % interval [1:n]
11 n = 10;
12
13 % vrednost V_i
14 i = 5;
15
16 % FUNKCIJA
17 % robniproblem(p,q,r,a,ya,n)
18 % y'' + p*y' + q*y = r
19 % a - interval, ya - zacetna pogoja, n - podintervali
20 [y, x] = robniproblem('1/x', '0', '0', [a, b], [ya, yb], n);
21
22 % izracunana napetost po DE s priblizki
23 Vizr = y(i+1);
24 % r = ((b - a) / (N+1) * 5) + a;
25 ro = x(i+1); % ro_i
26
```

```

27 % točna napetost po formuli OE1
28 Vtocna = (ya - yb) * log(ro/b) / log(a/b);
29
30 % RESITEV
31 printf('Izracunana napetost med tocko V(%d) in plascem:\n U = %0.15f\nTocna ...
    napetost tocko V(%d) in plascem:\n U = %0.15f\nAbsolutna vrednost ...
    razlike med izracunano in točno vrednostjo med tocko V(%d) in plascem:\n ...
    |dU| = %0.15f\n',i,Vizr,i,Vtocna,i,abs(Vizr-Vtocna));

```

18 Večkrat uporabljene funkcije

robniproblem

```

1 function[y,x]=robniproblem(p,q,r,a,ya,n)
2 % y'' + p*y' + q*y = r
3
4 h=(a(2)-a(1))/(n+1); % velikost intervalov
5 x=linspace(a(1),a(2),n+2); % matrika x od a do b s podintervali
6 x=x(2:end-1); % samo podintervali
7
8 P=eval(vectorize(p)).*ones(size(x)); % matrika P
9 Q=eval(vectorize(q)).*ones(size(x)); % matrika Q
10 R=eval(vectorize(r)).*ones(size(x)); % matrika R
11 B=R; % vektor desnih strani
12
13 B(1)=B(1)+ya(1)*(-1/(h*h)+P(1)/(2*h)); % posodobitev prvega elementa z ...
    upostevanjem zacetnega pogoja y(a)
14 B(n)=B(n)+ya(2)*(-1/(h*h)-P(n)/(2*h)); % posodobitev zadnjega elementa z ...
    upostevanjem zacetnega pogoja y(b)
15
16 D=-2/(h*h)+Q; % diagonala
17 U=1/(h*h)+P(1:n-1)/(2*h); % naddiagonala
18 L=1/(h*h)-P(2:n)/(2*h); % poddiagonala
19
20 % trisys(u,d,l,b)
21 % naddiagonala, diagonala, poddiagonala, vektor desnih strani
22 y=trisys(U,D,L,B);
23
24 x=[a(1),x(:)',a(2)]; % matrika x od a do b s podintervali
25 y=[ya(1),y(:)',ya(2)]; % matrika y od y(a) do y(b) s podintervali
26 return;

```

trisys

```

1 % y = trisys(u,d,l,b)
2 % d, u, l : glavna diagonala ter prva nad- in poddiagonala
3 % b : vektor desnih strani
4 % y : resitev sistema
5
6 function y = trisys(u,d,l,b)
7 n = length(b); % velikost vektorja b
8

```

```

9  for k = 2:n
10     mult = 1/(k-1)/d(k-1); % multiplikator: razmerje med prejsno celico ...
        diagonale in spodnje diagonale
11     d(k) = d(k)-mult*u(k-1); % posodobitev trenutne celice diagonale
12     b(k) = b(k)-mult*b(k-1); % posodobitev trenutne celice vektorja
13 end;
14
15 y(n) = b(n)/d(n);
16
17 for k = (n-1):-1:1
18     y(k) = (b(k)-u(k)*y(k+1))/d(k); % izracun novih y
19 end;

```

Newtonova interpolacija

```

1  function D=Newtons_Interp(x,y);
2  x=x(:); y=y(:); % pretvorba v stolpce
3  n=length(x); % dolzina vektorja x
4  D=zeros(n); % matrika nicel n n
5  D(:,1)=y;
6
7  for i=1:n-1;
8      D(1:end-i,1+i)=diff(D(1:end-i+1,i))./(x(i+1:end)-x(1:end-i));
9      % za i=1
10     % D od prve do predzadnje vrstice, od drugega do zadnjega stolpca = ...
        medsebojni razliki vrstic deljeno vrednost drugega do zadnjega x-a ...
        manj prvega do predzadnjega x-a
11 end;
12
13 D=D(1,:); % izpis prve vrstice D

```

Interpolacija polinoma

Funkcija, za dana vektorja x in y , vrača vektor koeficientov interpolacijskega polinoma.

```

1  function a = Interpol(x,y);
2  n=length(x);
3  x=x(:); y=y(:);
4  A=[ones(size(x)),x];
5
6  for i=3:n
7      A=[A,A(:,end).*x]; % izdelava matrike A; mnozenje levega stolpca z x, da ...
        dobimo za eno potenco vec (1,x,x^2,x^3,...)
8  end;
9
10 a=A\y;

```

19 Kubični zlepek

Skozi točke $(0, 1.02)$, $(1, 1.39)$ in $(2, 1.24)$ interpolirajte kubični zlepek, ki je sestavljen iz dveh kubičnih polinomov

$$p(x) = a_0 + b_0(x - x_0) + c_0(x - x_0)^2 + d_0(x - x_0)^3$$

in

$$q(x) = a_1 + b_1(x - x_1) + c_1(x - x_1)^2 + d_1(x - x_1)^3.$$

Ker imamo 8 neznank, potrebujemo tudi 8 enačb: $p(x_0) = y_0$, $q(x_1) = y_1$, $q(x_2) = y_2$, $p(x_1) = q(x_1)$, $p'(x_1) = q'(x_1)$, $p''(x_1) = q''(x_1)$ ter naravna pogoja $p''(x_0) = 0$ in $q''(x_0) = 0$. Vpišite vrednost zleпка v točki $x_0 = \frac{1}{2}$.

```

1 % VHODNE SPREMENLJIVKE
2 X=[0, 1, 2];
3 Y=[1.02, 1.39, 1.24];
4 x0=1/2;
5
6 % FUNKCIJA
7 A=zeros(8);
8 B=[Y(1:3), 0, 0, 0, 0, 0]';
9 A(1,1:4)=[1, X(1)-x0, (X(1)-x0)^2, (X(1)-x0)^3];
10 A(2,5:8)=[1, X(2)-x0, (X(2)-x0)^2, (X(2)-x0)^3];
11 A(3,5:8)=[1, X(3)-x0, (X(3)-x0)^2, (X(3)-x0)^3];
12 A(4,:) = [1, X(2)-x0, (X(2)-x0)^2, (X(2)-x0)^3, -1, -(X(2)-x0), ...
            -(X(2)-x0)^2, -(X(2)-x0)^3];
13 A(5,:) = [0, 1, 2*(X(2)-x0), 3*(X(2)-x0)^2, 0, -1, -2*(X(2)-x0), ...
            -3*(X(2)-x0)^2];
14 A(6,3:4)=[2, 6*(X(2)-x0)];
15 A(6,7:8)=[-2, -6*(X(2)-x0)];
16 A(7,3:4)=[2, 6*(X(1)-x0)];
17 A(8,7:8)=[2, 6*(X(3)-x0)];
18
19 a=A\B;
20
21 % GRAF
22 x=linspace(0,4,200);
23 y=[polyval(a(4:-1:1),x(1:25)),polyval(a(8:-1:5),x(26:200))];
24 figure; plot(x,y);

```

$$\begin{aligned} p'(x) &= b_0 + 2c_0(x - x_0) + 3d_0(x - x_0)^2 & q'(x) &= b_1 + 2c_1(x - x_0) + 3d_1(x - x_0)^2 \\ p''(x) &= 2c_0 + 6d_0(x - x_0) & q''(x) &= 2c_1 + 6d_1(x - x_0) \end{aligned}$$

$$\begin{bmatrix} 1 & x_1 - x_0 & (x_1 - x_0)^2 & (x_1 - x_0)^3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x_2 - x_0 & (x_2 - x_0)^2 & (x_2 - x_0)^3 \\ 0 & 0 & 0 & 0 & 1 & x_3 - x_0 & (x_3 - x_0)^2 & (x_3 - x_0)^3 \\ 1 & x_2 - x_0 & (x_2 - x_0)^2 & (x_2 - x_0)^3 & -1 & -(x_2 - x_0) & -(x_2 - x_0)^2 & -(x_2 - x_0)^3 \\ 0 & 1 & 2(x_2 - x_0) & 3(x_2 - x_0)^2 & 0 & -1 & -2(x_2 - x_0) & -3(x_2 - x_0)^2 \\ 0 & 0 & 2 & 6(x_2 - x_0) & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 6(x_1 - x_0) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 6(x_3 - x_0) \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ b_0 \\ c_0 \\ d_0 \\ a_1 \\ b_1 \\ c_1 \\ d_1 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

20 Trapezna in Simpsonova metoda

Izračunajte integral

$$\int_0^1 x^3 e^{1.44x} dx$$

s trapeznim in Simpsonovim tretjinskim pravilom, kjer interval $[0, 1]$ razdelite na $n = 4$ podintervalov.

```
1 f=inline('x.^3.*exp(1.44*x)','x');
2 a=0;
3 b=1;
4 n=4; % stevilo intervalov
5 h=(b-a)/n; % dolzina intervala
6
7 % Trapezna metoda
8 x=linspace(a,b,n+1);
9 y=f(x);
10 wt=[1,2*ones(1,n-1),1]; % utezi [1 2 2 2 2 ... 2 1]
11 T=h/2*dot(y,wt);
12 printf('T = %0.16f\n',T);
13
14 % Simpsonova metoda
15 ws=[1,(4-(1+(-1).^(1:n-1))),1]; % utezi [1 4 2 4 ... 2 4 1]
16 S=h/3*dot(y,ws); % mnozenje celice y(i,j) s celico ws(i,j)
17 printf('S = %0.16f\n',S);
```

Izračunajte integral funkcije

$$f(x) = \frac{\sin(4x)}{1 - e^x}$$

po trapezni in 1/3 Simpsonovi formuli na intervalu $[0, 1]$.

Integracijski interval razdelite na 30 enako dolgih podintervalov. Funkcija v levem krajišču ni definirana, zato tam namesto funkcijske vrednosti vzamite limito.

```
1 f = inline('sin(4.*x)/(1-exp(1.*x))','x');
2 a = 0;
3 b = 1;
4 n = 30;
5 h = (b-a)/n;
6 x = linspace(a,b,n+1);
7 y = f(x);
8 y(1) = -4; % limita x -> 0
9
10 % Trapezna metoda
11 w = [1, 2*ones(1,n-1), 1]; % ute i so [1 2 2 2 .. 2 1]
12 T = h/2*dot(y,w);
13 printf('T = %0.10f\n',T);
14
15 % Simpsonova metoda
16 w = [1,(4-(1+(-1).^(1:n-1))),1];
17 S = h/3*dot(y,w);
18 printf('S = %0.10f\n',S);
```

21 Rombergova metoda

Izračunajte integral

$$\int_1^3 \frac{\sin 1.86x}{x} dx$$

z Rombergovo metodo. Vpišite vrednost $T_2(\frac{h}{4})$.

```
1 % VHODNE SPREMENLJIVKE
2 a = 1;
3 b = 3;
4 f = inline('sin(1.95.*x)./x','x');
5 Tn = 3;
6 n = 4;
7
8 % FUNKCIJA
9 I = zeros(n);
10
11 for i=1:n
12     I(i,1)=trapezna(f,a,b,2^(i-1));
13     printf('I(%d,1) = %f\n',i,I(i,1)); % izpis posameznega I-ja
14 end;
15
16 for k=2:n
17     for j=k:n
18         I(j,k)=(4^(k-1).*I(j,k-1)-I(j-1,k-1))/(4^(k-1)-1);
19         printf('I(%d,%d) = %f\n',j,k,I(j,k)); % izpis posameznega I-ja
20     end;
21 end;
22
23 printf('T_%d(h/4) = %0.15f\n',Tn,I(Tn+1,Tn+1));
```

22 Utež in vozeli kvadrature formule

Določite uteži in vozeli kvadrature formule

$$\int_0^h \frac{f(x)}{\sqrt{x}} dx = w_1 f(x_1)$$

tako, da bo točna za funkciji $f(x) = 1$ in $f(x) = x$ ter izračunajte integral

$$\int_0^{0.27} \frac{\sin x}{x\sqrt{x}} dx$$

s to kvadraturno formulo.

```
1 f = inline('sin(x)/x','x');
2 h = 0.27;
3 a = 2*sqrt(h)*f(h/3);
4
5 printf('I = %0.10f \n',a);
```

23 Gaussova kvadratura formula

Določi uteži w_1 in w_2 ter vozlišči x_1 in x_2 tako, da bo kvadratura formula

$$\int_0^1 f(x) dx \approx w_1 f(x_1) + w_2 f(x_2)$$

točna za $f(x) = x^n$, $n = 0, 1, 2, 3$.

Iz pogojev $\int_0^1 x^n dx = w_1 x_1^n + w_2 x_2^n$ dobimo enačbe:

$$\begin{aligned} 1 - w_1 - w_2 &= 0, \\ 1/2 - w_1 x_1 - w_2 x_2 &= 0, \\ 1/3 - w_1 x_1^2 - w_2 x_2^2 &= 0, \\ 1/4 - w_1 x_1^3 - w_2 x_2^3 &= 0 \end{aligned}$$

Sistem rešimo s pomočjo Newtonove metode.

```
1 f = inline('1-w1-w2; 1/2-w1*x1-w2*x2; 1/3-w1*x1^2-w2*x2^2; ...  
    1/4-w1*x1^3-w2*x2^3','x1','w1','x2','w2');  
2 df = inline('[0,-1,0,-1; -w1,-x1,-w2,-x2; -2*w1*x1,-x1^2,-2*w2*x2,-x2^2; ...  
    -3*w1*x1^2,-x1^3,-3*w2*x2^2,-x2^3]','x1','w1','x2','w2');  
3  
4 u = [0;1;1;1];  
5 eps = 1e-8;  
6  
7 for i=1:100  
8     x1=u(1); w1=u(2); x2=u(3); w2=u(4);  
9     u=[x1;w1;x2;w2]-df(x1,w1,x2,w2)\f(x1,w1,x2,w2);  
10    if abs(u-[x1;w1;x2;w2]) < eps  
11        break  
12    end;  
13 end;  
14  
15 w1=u(2); w2=u(4); x1=u(1); x2=u(3);  
16  
17 g = inline('exp(-x.^2)','x');  
18 n = 10; a = 0; b = 1;  
19 h = (b-a)/n;  
20 H = h*[(0:(n-1))+u(1)]; (0:(n-1))+u(3));  
21 I = h/2*sum(sum(g(H)));  
22 printf('I=%0.10f',I);
```

24 Gaussova kvadratura formula za singularne integrale

Izračunajte integral funkcije

$$f(x) = \frac{1 - e^{-x^2}}{x^2 \sqrt{x^{3/2}}} = \frac{1 - e^{-x^2}}{x^2 \cdot x^{3/4}}$$

na intervalu $[0, 2]$. Uporabite Gaussovo kvadraturno formulo za singularne integrale oblike

$$\int_a^b \frac{f(x)}{(x-a)^s} dx \approx w_1 f(x_1) + w_2 f(x_2),$$

kjer je funkcija f regularna na $[a, b]$. Vozla x_1 in x_2 ter koeficienta w_1 in w_2 dobite tako, da bo ta kvadratura formula točna za $f(x) \in \{1, x, x^2, x^3\}$.

Wolfram Mathematica

```
1 Table[Integrate[x^n/x^s, {x, 0, 2}, Assumptions -> {0 < s < 1}], {n, 0, 3}] ...  
/. s -> 3/4
```

dobljeno tabelo vstavimo v vektor **a**

```
1 a = [4*(2)^(1/4), (8*(2)^(1/4))/5, (16*(2)^(1/4))/9, (32*(2)^(1/4))/13];  
2  
3 f = inline(' [a(1)-w1-w2; a(2)-w1*x1-w2*x2; a(3)-w1*x1^2-w2*x2^2; ...  
4           a(4)-w1*x1^3-w2*x2^3]', 'x1','w1','x2','w2');  
5 df = inline(' [0,-1,0,-1; -w1,-x1,-w2,-x2; -2*w1*x1,-x1^2,-2*w2*x2,-x2^2; ...  
6           -3*w1*x1^2,-x1^3,-3*w2*x2^2,-x2^3]', 'x1','w1','x2','w2');  
7  
8 u = [0,1,2,1];  
9 eps = 1e-8;  
10  
11 for i=1:100  
12     x1 = u(1); w1 = u(2); x2 = u(3); w2 = u(4);  
13     u = [x1;w1;x2;w2] - df(x1,w1,x2,w2)\f(x1,w1,x2,w2);  
14     if abs(u-[x1;w1;x2;w2])<eps  
15         break  
16     end;  
17 end;  
18 x1 = u(1);  
19 w1 = u(2);  
20 x2 = u(3);  
21 w2 = u(4);  
22  
23  
24 g = inline('(1-exp(-x.^2))/(x.^2)', 'x');  
25  
26 I = w1*g(x1) + w2*g(x2);  
27 printf('I = %0.10f\n',I);
```

25 Robni problem

Poišči približno rešitev Laplaceove enačbe

$$\frac{\partial^2 u(x, y)}{\partial x^2} + \frac{\partial^2 u(x, y)}{\partial y^2} = 0$$

na območju $\mathcal{D} = [0, 1] \times [0, 1]$ z robnim pogojem

$$u(x, y)|_{\partial\mathcal{D}} = f(x, y)$$

kjer je $f(x, y) = x^2 + y^2$.

Na območje $\mathcal{D}$ položiš kvadratno mrežo s korakom $h = 1/n$, $n = 10$.

Na vsakem notranjem vozlišču odvode nadomestiš z razlikami in zapišeš sistem enačb, ki ga rešiš z Gauss-Seidlovo iteracijo.

Približne vrednosti rešitve zapišemo v matriko $U_{i,j} \approx u(x_i, y_j)$, $i = 0, \dots, n$ in $j = 0, \dots, n$. Za $i = 0$, $i = n$, $j = 0$ in $j = n$ zapišemo ustrezne robne vrednosti.

Sistem enačb:

$$U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1} - 4U_{i,j} = 0,$$

$i, j = 1, \dots, n-1$.

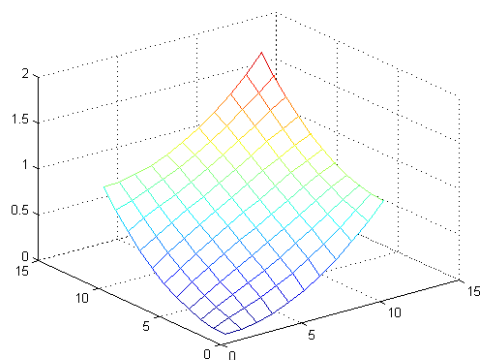
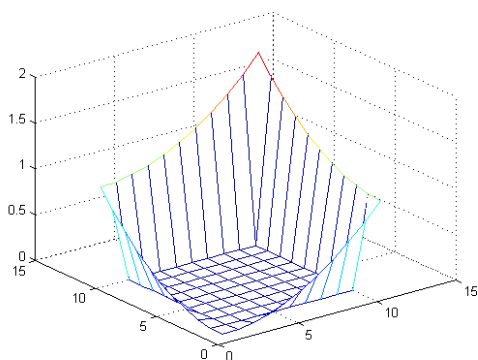
Inicializiramo vrednosti matrike, v notranjih vozliščih postavimo vrednost na 0, na robu pa zapišemo ustrezne robne vrednosti.

$$U_{i,j}^{k+1} = (U_{i+1,j}^k + U_{i-1,j}^k + U_{i,j+1}^k + U_{i,j-1}^k) / 4.$$

```

1 f=inline('x.^2+y.^2','x','y');
2 n=10; h=1/n; eps=1e-8; [x,y]=meshgrid(0:h:1,0:h:1);
3 U=f(x,y); U(2:end-1,2:end-1)=0; mesh(U);
4
5 for k=1:1000
6     V=U;
7
8     for i=2:n,
9         for j=2:n,
10             U(i,j)=(U(i+1,j)+U(i-1,j)+U(i,j-1)+U(i,j+1))/4;
11         end;
12     end;
13
14     if abs(U-V)<eps
15         break
16     end;
17 end;
18
19 figure; mesh(U);

```



26 Laplaceova enačba

Poiščite rešitev Laplaceove enačbe

$$u_{xx} + u_{yy} = 0$$

v štirih notranjih točkah območja, ki je podano na sliki. Na robu imate naslednje vrednosti: $u(1,0) = 10$, $u(2,0) = 100$, $u = 100$ na preostalih treh robovih. Na območje položite ekvidistančno mrežo in gornjo enačbo nadomestite z enačbo

$$\frac{u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + u_{i+1,j} - 4u_{i,j}}{h^2} = 0.$$

```
1 u10 = 10; % u(1,0)
2 u20 = 100; % u(2,0)
3 u = 100; % u na preostalih treh robovih
4
5 X = [1,0,-4,1; 0,1,1,-4; -4,1,1,0; 1,-4,0,1];
6 Y = [-(u10+u); -(u20+u); -(2*u); -(2*u)];
7
8 U = X\Y
```

27 Simpsonova metoda za parametrično krivuljo

Dana je krivulja v parametrični obliki

$$x = 4t - 4\sin(t), \quad y = 6 - 6\cos(t).$$

Izračunajte dolžino krivulje. Razdelite interval $[0, 2\pi]$ na $n = 26$ podintervalov in izračunajte integral s pomočjo Simpsonove formule. Formula za dolžino loka v parametrični obliki se glasi

$$s = \int_a^b \sqrt{\dot{x}^2 + \dot{y}^2} dt.$$

```
1 % VHODNE SPREMENLJIVKE
2 f = inline('sqrt((4-4*cos(t)).^2+(6.*sin(t)).^2)','t');
3
4 a = 0;
5 b = 2*pi;
6 n = 26;
7
8 % FUNKCIJA
9 h = (b-a)/n;
10 x=linspace(a,b,n+1);
11 y = f(x);
12 ws=[1, (4-(1+(-1).^(1:n-1))),1]; % utezi [1 4 2 4 ... 2 4 1]
13 S=h/3*dot(y,ws); % množenje celice y(i,j) s celico ws(i,j)
14 fprintf('S = %0.16f\n',S);
```

28 Rombergova metoda za integral

Izračunajte integral

$$\int_1^2 \frac{\cos 1.23x}{x} dx$$

z Rombergovo metodo. Integral najprej izračunamo s trapezno formulo, kjer za začetni korak h vzamemo kar dolžino intervala $[1, 2]$. Korak nato razpolavljamo, da dobimo vrednosti $T_{i,1}$. S formulama

$$T_{i,2} = \frac{4T_{i,1} - T_{i-1,1}}{3}, \quad T_{i,3} = \frac{16T_{i,2} - T_{i-1,2}}{15}$$

izračunajte vrednosti $T_{i,2}$ in $T_{i,3}$. Vpišite vrednosti $T_{3,1}$, $T_{3,2}$ in $T_{3,3}$.

```
1 % VHODNE SPREMENLJIVKE
2 f = inline('cos(1.23.*x)./x','x');
3
4 a = 1;
5 b = 2;
6 n = 3;
7
8 % FUNKCIJA
9 x = [a b];
10 I = rombergtab(f,x,n);
11 %{
12 h = (b-a)/3
13
14 % Trapezna formula
15 I11 = (b-a)/2 * (f(a)+f(b));
16 I21 = (b-a)/4 * (f(a) + 2*f((b-a)/2) + f(b));
17 I31 = (b-a)/6 * (f(a) + 2*f(a+h) + 2*f(a+2*h) + f(b));
18
19 I22 = (4*I21 - I11)/(4-1);
20 I32 = (4*I31 - I21)/(4-1);
21
22 I33 = (4^2*I32 - I22)/(4^2-1);
23 %}
24
25 printf('I31 = %0.10f \nI32 = %0.10f \nI33 = %0.10f\n',I(3,1), I(3,2), I(3,3));
```

29 Gaussova kvadratura formula in uteži

Določite uteži Gaussove kvadrature formule

$$\int_0^h \sqrt{x} f(x) dx = w_1 f(h/5) + w_2 f(4h/5)$$

tako, da bo formula točna za funkcije $f(x) = 1$ in $f(x) = x$ ter izračunajte integral

$$\int_0^{2.95} x^{7/2} dx$$

s to kvadraturno formulo. Rezultat primerjajte s točno rešitvijo.

```

1 % VHODNE SPREMENLJIVKE
2 h = 2.95;
3 A = [1 1; h/5 4*h/5];
4 b1 = 2/3 * (h)^(3/2);
5 b2 = 2/5 * (h)^(5/2);
6
7 % FUNKCIJA
8 b = [b1 b2]';
9 x = A\b;
10 f = inline('x^3','x');
11 int = x(1)*f(h/5) + x(2)*f(4*h/5);
12 tocna = 2/9 * h^(9/2);
13 printf('w1 = %0.10f\nw2 = %0.10f \nI = %0.10f\ntocna = ...
        %0.10f\n',x(1),x(2),int,tocna);

```

30 Dirichletov problem

Rešujete Laplaceovo enačbo z Dirichletovim robnim pogojem, na kvadratnem območju s stranico 1:

$$\Delta u(x, y) = 0, \quad (x, y) \in \mathcal{D} = (0, 1) \times (0, 1),$$

$$u(x, y) = x^2 + 1.17y^3, \quad (x, y) \in \mathcal{D}.$$

Na območje položite ekvidistantno pravokotno mrežo z $n \times n$ vozlišči znotraj območja, kjer je $n = 27$, in nadomestite gornjo diferencialno enačbo z diferenčno enačbo:

$$\frac{1}{h^2}(U_{i+1,j} - 2U_{i,j} + U_{i-1,j} + U_{i,j+1} - 2U_{i,j} + U_{i,j-1}) = 0,$$

$$i = 1, \dots, n, \quad j = 1, \dots, n, \quad h = x_{i+1} - x_i = y_{j+1} - y_j.$$

Sistem rešite z Jacobijevo iteracijo po naslednji shemi:

$$U_{i,j}^{(k+1)} = \frac{1}{4}(U_{i+1,j}^{(k)} + U_{i-1,j}^{(k)} + U_{i,j+1}^{(k)} + U_{i,j-1}^{(k)}).$$

Postopek končate, ko absolutna vrednost največje razlike med dvema sosednjima iteracijama pade pod $\varepsilon = 10^{-4}$. Začetni približek je enak 0. Koliko je $U_{10,10}$?

```

1 % VHODNE SPREMENLJIVKE
2 f = inline('x.^2+1.17.*y.^3','x','y');
3 n = 27;
4 h = 1/n;
5 eps=1e-4;
6
7 % FUNKCIJA
8 n = n+1;
9 [x,y] = meshgrid(0:h:1,0:h:1); % mreza
10 U=f(x,y); % robne vrednosti
11 U(2:end-1,2:end-1) = 0; % sredina na nic
12
13 figure; mesh(U); % graf
14
15 for k=1:1000
16     V=U;
17     for i=2:n,

```

```

18         for j=2:n,
19             U(i,j)=(V(i+1,j)+V(i-1,j)+V(i,j-1)+V(i,j+1))/4;
20         end;
21     end;
22     if abs(U-V)<eps
23         break
24     end;
25 end;
26
27 figure; mesh(U); % graf
28 printf('U(10,10)= %0.10f\n',U(11,11)); % U(+1,+1)
29 printf('k = %d\n',k);

```

31 Adams-Bashforth 2. reda

Rešite diferencialno enačbo

$$y_0 = -x^2y, \quad y_0 = y(0) = 2.51$$

z Adams-Bashforthovo metodo 2. reda:

$$y_{i+2} = y_{i+1} + h \left(\frac{3}{2} f(x_{i+1}, y_{i+1}) - \frac{1}{2} f(x_i, y_i) \right)$$

na intervalu $[0, 1/2]$. Integracijski interval razdelite na $n = 10$ enako dolgih podintervalov. Ker potrebujete za računanje y_{i+2} pri tej metodi tako y_{i+1} kot tudi y_i , izračunajte y_1 z Eulerjevo metodo $y_{i+1} = y_i + hf(x_i, y_i)$. Vpišite vrednost rešitve v točki $\frac{1}{2}$.

```

1  % VHDONE SPREMENLJIVKE
2  f = inline('-x.^2.*y','x','y');
3  n = 10;
4  a = [0 1/2];
5  y0 = 2.51;
6
7  % FUNKCIJA
8  [y1, x1] = euler(f,a,y0,n);
9  y = [y0, y1(2)];
10 h = (a(2)-a(1))/n;
11 x = linspace(0,1/2,n+1);
12
13 for i = 3:n+1
14     y(i) = y(i-1)+h*(3/2*f(x(i-1),y(i-1))-1/2*f(x(i-2),y(i-2)));
15 end;
16
17 printf('res = %0.10f\n',y(end));

```

32 Runge-Kutta 4. reda

Rešite diferencialno enačbo

$$y_0 = -2xy, \quad y(0) = 1.82$$

po Runge-Kutta metodi četrtega reda:

$$\begin{aligned}k_1 &= hf(x_n, y_n), \\k_2 &= hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right), \\k_3 &= hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2\right), \\k_4 &= hf(x_n + h, y_n + k_3), \\y_{n+1} &= y_n + \frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4,\end{aligned}$$

na intervalu $[0, 2]$. Integracijski interval razdelite na $n = 24$ enako dolgih podintervalov. Primerjajte vrednost numerične rešitve v desnem krajišču po tej metodi s točno rešitvijo.

```
1 % VHODNE SPREMENLJIVKE
2 f = inline('-2*x*y', 'x', 'y');
3 n = 24;
4 a = [0 2];
5 y0 = 1.82;
6
7 % FUNKCIJA
8 [y,x] = rk4(f,a,y0,n);
9 y1 = y0*exp(-a(end)^2);
10 printf('rk4 = %0.10f\ntocna = %0.10f\n', y(end), y1);
```

33 Funkcije

33.1 Trapezna metoda

```
1 function T = trapezna(f,a,b,n);
2 % funkcija, meji, n
3 h=(b-a)/n;
4 x=linspace(a,b,n+1);
5 y=f(x);
6 w=[1,2*ones(1,n-1),1];
7 T=h/2*dot(y,w);
```

33.2 Romberg

```
1 % I = rombergtab(f,x,n)
2 % Izracun tabele vrednosti integrala z Rombergovo metodo.
3 % Vhod:
4 % f : analiticno podana funkcija
5 % x : integracijski interval [a,b]
6 % n : dimenzija tabele vrednosti
7 % Izhod:
8 % I : tabela približnih vrednosti integrala
9 function I = rombergtab(f,x,n)
```

```

10 I = zeros(n);
11 h = diff(x);
12 a = x;
13 y = f(x);
14 I(1,1) = h*mean(y);
15 for i = 2:n
16     h = h/2;
17     x = a(1)+h:2*h:a(2);
18     s = sum(f(x));
19     I(i,1) = I(i-1,1)/2+h*s;
20     q = 1;
21     for j = 2:i
22         q = q/4;
23         I(i,j) = (I(i,j-1)-q*I(i-1,j-1))/(1-q);
24     end;
25 end;

```

33.3 Runge-Kutta 4

```

1 % [y,x] = rk4(f,a,ya,n);
2 % Eksplicitna Runge-Kutta metoda cetrttega reda za reševanje začetnega problema
3 % y' = f(x,y), y(x0) = y0
4 % Vhod:
5 % f: funkcija f(x,y) — desna stran diferencialne enačbe y' = f(x,y)
6 % a: interval [a,b] na katerem iscemo numerično rešitev
7 % ya: vektor začetnih pogojev v levem krajiscu
8 % n: stevilo podintervalov danega intervala
9 % Izhod:
10 % x: delitev intervala [a, b]
11 % y: matrika vrednosti funkcije y(x) v notranjih točkah intervala in na robu
12 function [y,x] = rk4(f,a,ya,n)
13     h = (a(2)-a(1))/n;
14     y = zeros(length(ya),n+1);
15     y(:,1) = ya(:);
16     x = linspace(a(1),a(2),n+1);
17
18     for i = 2:n+1
19         k1 = h*f(x(i-1), y(:,i-1));
20         k2 = h*f(x(i-1)+h/2, y(:,i-1)+k1(:)/2);
21         k3 = h*f(x(i-1)+h/2, y(:,i-1)+k2(:)/2);
22         k4 = h*f(x(i-1)+h, y(:,i-1)+k3(:));
23         y(:,i) = y(:,i-1) + k1(:)/6 + k2(:)/3 + k3(:)/3 + k4(:)/6;
24     end;

```