# Laboratorijske vaje iz Numeričnih metod

Andrej Perne

Fakulteta za elektrotehniko

2012/2013

# Naloga 1.1

Funkcijo $f(x) = e^{-x^2}$ razvijte v Taylorjevo vrsto in s pomočjo razvoja poiščite približno vrednost integrala $I = \int_0^1 e^{-x^2}\,dx$.

$$e^{-x^2} = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{n!} \Rightarrow \int_0^1 e^{-x^2}\,dx = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)n!}$$

Integral izračunajte še analitično z uvedbo nove spremenljivke $t = x^2$ in uporabo enakosti $\gamma(x, a) = \frac{1}{\Gamma(a)} \int_0^x t^{a-1} e^{-t}\,dt$.

$$t = x^2 \Rightarrow dt = 2x\,dx \Rightarrow dx = \frac{dt}{2\sqrt{t}} \Rightarrow$$

$$\int_0^1 e^{-x^2}\,dx = \tfrac{1}{2} \int_0^1 t^{-1/2} e^{-t}\,dt = \tfrac{\sqrt{\pi}}{2\Gamma(\frac{1}{2})} \int_0^1 t^{1/2-1} e^{-t}\,dt = \tfrac{\sqrt{\pi}}{2} \cdot \gamma(1, 1/2)$$

# Naloga 1.1

Seštejte prvih 11 členov vrste, kjer uporabite zvezo $\Gamma(n+1) = n!$ ter ukaze: `sum()`, .∧, .*, ./, : in `gamma()`.

```
n = 0:10;
s = (-1).^n./(2*n+1)./gamma(n+1);
I10 = sum(s);
fprintf('numintegral = %2.10f\n',I10);
```

Z uporabo funkcije `gammainc()` in zveze $\Gamma(\frac{1}{2}) = \sqrt{\pi}$ izračunajte točno vrednost integrala ter določite relativno napako: $\delta = \left|\frac{(I - I_{10})}{I}\right|$. Absolutna vrednost se izračuna z ukazom `abs()`.

```
I = gammainc(1,1/2)/2*sqrt(pi);
fprintf('integral = %2.10f\n',I);
delta = abs((I-I10)/I);
fprintf('delta = %2.10f\n',delta);
```

# Naloga 1.2

Z uporabo zveze

$$\frac{\sqrt{1+x}-1}{x} = \frac{(\sqrt{1+x}-1)(\sqrt{1+x}+1)}{x(\sqrt{1+x}+1)} = \frac{1}{\sqrt{1+x}+1}$$

izračunajte izraz $(\sqrt{1+x}-1)/x$ za $x = \frac{1}{100}$ na dva načina.
Za majhne vrednosti za $x$ je račun, kjer odštevamo dve približno enaki števili, slabo pogojen. Uporabimo funkcijo kvadratni koren `sqrt()`.
Kateri račun da natančnejši rezultat?

Izračun s programom `Mathematica`® na 50 točnih mest:
0.49875621120890270219264912759576186945023470026377

```
format long;
x = 0.01;
v1 = (sqrt(1+x)-1)/x;
v2 = 1/(sqrt(1+x)+1);
fprintf('prvi = %0.16f drugi = %0.16f\n',v1,v2);
```

# Naloga 1.3

Integral $I_n = \int_0^1 x^n/(x+10)\,dx$ izračunajte s pomočjo rekurzivne formule na dva načina za $n = 1, 2, \ldots, 17$.

$$I_n + 10I_{n-1} = \int_0^1 \frac{x^n + 10x^{n-1}}{x+10}\,dx = \int_0^1 x^{n-1}\,dx = \frac{1}{n}$$

1. rekurzivna formula:

$$I_0 = \int_0^1 \frac{dx}{x+10} = \log\frac{11}{10},\ \ I_n = \frac{1}{n} - 10I_{n-1},\ \ n = 1, 2, \ldots, 17$$

2. rekurzivna formula:

$$I_{30} = 0,\ \ I_{n-1} = \frac{1}{10}\left(\frac{1}{n} - I_n\right),\ \ n = 30, 29\ldots, 1$$

# Naloga 1.3

**Uporabimo zanko `for`, funkcijo `log()` in operacijo `[,]`.**

Vrednost $I_{17}$ izračunana s programom `Mathematica`® na 50 točnih mest: 0.00507489273026384323593898029218181489241671247733 44

```
format long;
% prva rekurzija
I1 = zeros(1,18);
I1(1) = log(11/10);
for i = 2:18
  I1(i) = 1/(i-1) - 10*I1(i-1);
end;
% druga rekurzija
I2 = zeros(1,31);
for i = 30:-1:1
  I2(i) = (1/i - I2(i+1))/10;
end;
fprintf('prvi = %0.16f drugi = %0.16f\n',I1(18),I2(18));
```

# Naloga 2.1

Z uporabo funkcijske m-datoteke definirajte novo funkcijo `fn`, ki je podana s predpisom:

$$\mathrm{fn}(x) = \begin{cases} \pi e^x, & x < 0 \\ \pi + \sin(\pi x), & 0 \leq x < \frac{1}{2} \\ \pi + 1 - (x - \frac{1}{2})^2, & \frac{1}{2} \leq x < \frac{1}{2} + \sqrt{\pi + 1} \\ 0, & \text{drugod} \end{cases}$$

Izračunajte funkcijski vrednosti v točkah $x = 1$ in $x = 2.4$.

Funkcijska m-datoteka se začne s stavkom `function`.

# Naloga 2.1

## Datoteka fn.m

```
function y = fn(x)
  % y = fn(x);
  y = zeros(1,length(x));
  for i = 1:length(x)
    if x(i) < 0, y(i) = pi*exp(x(i));
    elseif x(i) < 1/2, y(i) = pi+sin(pi*x(i));
    elseif x(i) < 1/2+sqrt(pi+1), y(i) = pi+1-(x(i)-1/2)^2;
    else y(i) = 0;
    end;
  end;
```

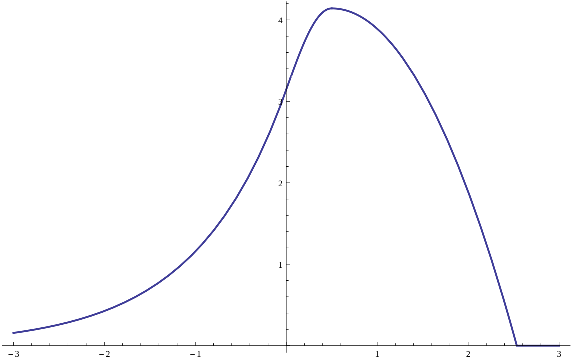
```
x1 = 1; y1 = fn(x1);
x2 = 2.4; y2 = fn(x2);
printf('y(%0.2f) = %0.5f y(%0.2f) = %0.5f\n',x1,y1,x2,y2);
ans:  y(1.00) = 3.89159 y(2.40) = 0.53159
```

# Naloga 2.1

Z uporabo ukaza `linspace` ali operatorja `:` in ukaza `plot` narišite graf funkcije `fn` na intervalu $[-3, 3]$.

Graf funkcije:

```
x = linspace(-3,3); y = fn(x); plot(x,y);
```



# Naloga 2.2

S pomočjo Eulerjeve transformacije določite približno vrednost vsote slabo konvergentne alternirajoče vrste $S = \sum_{k=0}^{\infty} \frac{(-1)^k}{\sqrt{k+1}}$. Členi vrste po absolutni vrednosti enakomerno padajo proti 0.

Delne vsote: $S_n = \sum_{k=0}^{n} \frac{(-1)^k}{\sqrt{k+1}}, \quad n = 0, 1, \ldots, N.$

Povprečenje delnih vsot:

$$
\begin{aligned}
S_n^{(0)} &= S_n, \quad n = 0, 1, \ldots, N, \\
S_n^{(1)} &= (S_{n+1}^{(0)} + S_n^{(0)})/2, \quad n = 0, 1, \ldots, N-1, \\
S_n^{(2)} &= (S_{n+1}^{(1)} + S_n^{(1)})/2, \quad n = 0, 1, \ldots, N-2, \\
&\ldots \\
S_n^{(m)} &= (S_{n+1}^{(m-1)} + S_n^{(m-1)})/2, \quad n = 0, 1, \ldots, N-m.
\end{aligned}
$$

# Naloga 2.2

Za povprečenje delnih vsot izberemo $N = 40$ in $m = 30$ ter uporabimo ukaz cumsum za izračun kumulativne vsote.

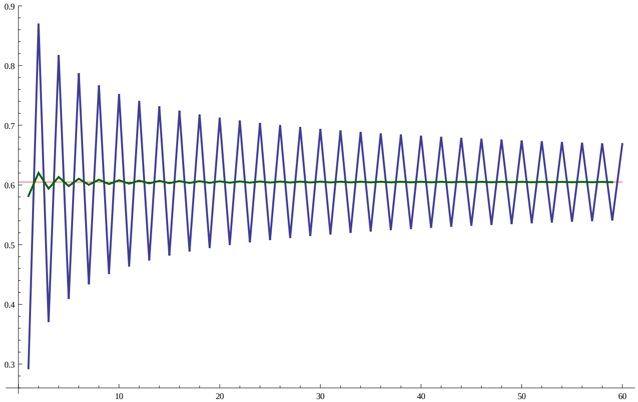
Rezultat primerjamo z vrednostjo, ki jo vrne program Mathematica® na 50 točnih mest:
0.60489864342163037024726591423595549975976254513025

```
n = 0:40;
s = (-1).^n./sqrt(n+1);
ss = cumsum(s);
se = ss;
for i = 1:30
  se = (se(1:end-1)+se(2:end))/2;
end;
fprintf('Vsota vrste z Euler.trans.S = %0.16f\n',se(end));
ans:  Vsota vrste z Euler.trans.S = 0.604898643421630
fprintf('Delna vsota vrste S40 = %0.16f\n',ss(end));
ans:  Delna vsota vrste S40 = 0.68250947328015б
```

# Naloga 2.2

## Graf delnih vsot:

```
z = ss; z = (z(1:end-1)+z(2:end))/2;
plot(n,ss,'b',n(1:end-1),z,'g',n,se(end)*ones(size(n)),'r')
```

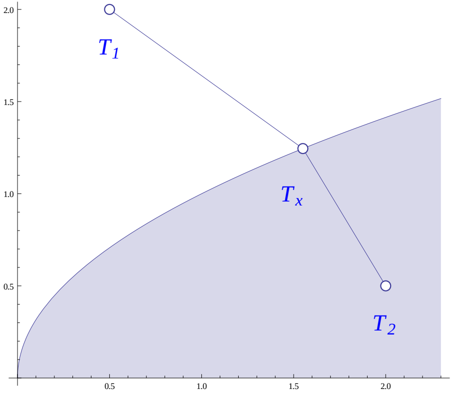



Modro so delne vsote, zeleno je prvo povprečenje in rdeče je limitna vrednost.

# Naloga 6.1

## Pot svetlobnega žarka

Svetlobni žarek začne pot v točki $T_1(1/2, 2)$ in jo konča v točki $T_2(2, 1/2)$. Svetlobna hitrost na svetlejšem delu območja je enaka $c_1 = 1$, na temnejšem delu pa $c_2 = 0.3$. Mejo območja določa funkcija $f(x) = \sqrt{x}$.



# Naloga 6.1

**Čas potovanja žarka od točke $T_1(x_1, y_1)$ do točke $T_2(x_2, y_2)$, kjer prečka mejo $y = f(x)$ v točki $T_x(x, f(x))$, je enak:**

$$t(x) = \frac{\sqrt{(x_1 - x)^2 + (y_1 - f(x))^2}}{c_1} + \frac{\sqrt{(x - x_2)^2 + (f(x) - y_2)^2}}{c_2}.$$

**Fermatov princip**

Žarek izbere med dvema točkama takšno pot, da je čas potovanja najkrajši:

$$t'(x) = 0.$$

# Naloga 6.1

## Program

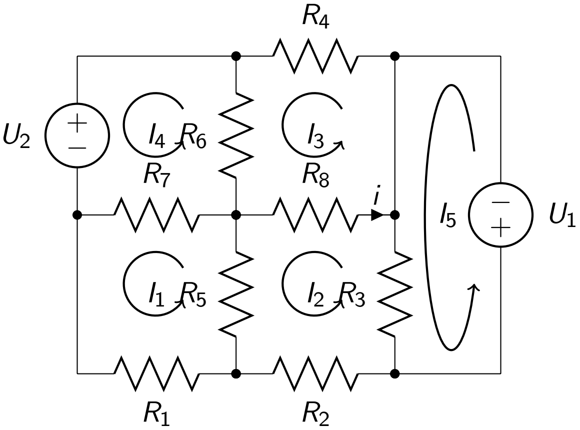
Za reševanje nelinearne enačbe uporabimo sekantno metodo.

```
f = inline('sqrt(x)','x');
T = [0.5,2;2,0.5]; c = [1,0.3]; eps = 1e-8;
t = inline('norm(T(1,:)-[x,f(x)])/c(1) +
norm([x,f(x)]-T(2,:))/c(2)','x');
dt = inline('(t(x+eps)-t(x-eps))/2/eps','x');
x1 = 1; x2 = 2;
for i = 1:100
  xi = x2-dt(x2)*(x2-x1)/(dt(x2)-dt(x1));
  x1 = x2; x2 = xi;
  if abs(x1-x2) < eps, break, end;
end;
printf('Tx(%0.6f,%0.6f)\n',xi,f(xi));
Tx(1.550027,1.245001)
```

# Naloga 6.2

## Električno vezje

Za vezje na sliki uporabite Kirchoffova zakona in zapišite sistem linearnih enačb. Koliko je $i$? Vektor uporov je $R = (20, 30, 40, 25, 10, 15, 34, 20)\Omega$, napetosti pa sta $U_1 = 40V$ in $U_2 = 35V$.



# Naloga 6.2

## Sistem linearnih enačb: $i = I_3 - I_2$

$$
\begin{aligned}
0 &= R_7(I_1 - I_4) + R_5(I_1 - I_2) + R_1 I_1 \\
0 &= R_5(I_2 - I_1) + R_2 I_2 + R_3(I_2 - I_5) + R_8(I_2 - I_3) \\
0 &= R_4 I_3 + R_6(I_3 - I_4) + R_8(I_3 - I_2) \\
-U_2 &= R_7(I_4 - I_1) + R_6(I_4 - I_3) \\
-U_1 &= R_3(I_5 - I_2)
\end{aligned}
$$

## Matrična oblika

$$
\begin{bmatrix}
R_1+R_5+R_7 & -R_5 & 0 & -R_7 & 0 \\
-R_5 & R_2+R_3-R_5+R_8 & -R_8 & 0 & -R_3 \\
0 & -R_8 & R_4+R_6+R_8 & -R_6 & 0 \\
-R_7 & 0 & -R_6 & R_6+R_7 & 0 \\
0 & -R_3 & 0 & 0 & R_3
\end{bmatrix}
\cdot
\begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ 0 \\ -U_2 \\ -U_1 \end{bmatrix}
$$

# Naloga 6.2

## Program

Za reševanje sistema linearnih enačb uporabimo levo deljenje.

```
r = [20,30,40,25,10,15,34,20];
U = [0; 0; 0; -35; -40];
R = [r(1)+r(5)+r(7),-r(5),0,-r(7),0;...
-r(5),r(2)+r(3)+r(5)+r(8),-r(8),0,-r(3);...
0,-r(8),r(4)+r(6)+r(8),-r(6),0;...
-r(7),0,-r(6),r(6)+r(7),0;...
0,-r(3),0,0,r(3)];
I = R\U;
printf('i=%0.6f\n',I(3)-I(2));
i = 0.314720
```

# Naloga 6.3

Poiščite rešitev linearnega sistema $Ax = b$, kjer je

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

Za reševanje sistema uporabite Gaussovo eliminacijo z delnim pivotiranjem ter določite vse tri pivotne elemente.

## Zapis razširjenega sistema

```
A = [1 4 -2; 2 1 -3; 3 5 -1];
b = [3; 0; 7];
R = [A b];
```

# Naloga 6.3

## Program

```
[pivot1,k] = max(abs(R(:,1)));
t = R(1,:); R(1,:)  = R(k,:); R(k,:)  = t;
R(1,:)  = R(1,:)/R(1,1);
for j = 2:3, R(j,:)  = R(j,:)-R(j,1)*R(1,:); end;
[pivot2,k] = max(abs(R(2:end,2)));
t = R(2,:); R(2,:)  = R(k+1,:); R(k+1,:)  = t;
R(2,:)  = R(2,:)/R(2,2);
R(3,:)  = R(3,:)-R(3,2)*R(2,:);
pivot3 = abs(R(3,3));
R(3,:)  = R(3,:)/R(3,3);
for j = 1:2, R(j,:)  = R(j,:)-R(j,3)*R(3,:); end;
R(1,:)  = R(1,:)-R(1,2)*R(2,:);
```

# Naloga 7.1 — Airyeva diferencialna enačba

Poiščite rešitev robnega problema

$$y''(x) + x\,y(x) = 0, \quad y(a) = A,\ y(b) = B,$$

kjer je $a = 0$, $b = 2$, $A = 0$ in $B = 1$.

Iščemo približno rešitev. Interval $[a, b]$ razdelimo na $n = 20$ enakih delov. Približne vrednosti funkcije $y_i \approx y(x_i)$ v vozlih

$$x_i = a + i\,h, \quad i = 1, \ldots, n-1, \quad h = (b-a)/n,$$

dobimo kot rešitev sistema linearnih enačb

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

kjer je $i = 1, \ldots, n-1$, $y_0 = A$ in $y_n = B$.

# Naloga 7.1 — tridiagonalni sistem

Sistem linearnih enačb lahko zapišemo v matrični obliki $My = b$, kjer je matrika $M$ tridiagonalna. Razširjena matrika sistema $[M|b]$:

$$\left[\begin{array}{cccccc|c} -2+x_1h^2 & 1 & 0 & \cdots & & 0 & -y_0 \\ 1 & -2+x_2h^2 & 1 & \cdots & & 0 & 0 \\ \multicolumn{7}{c}{\dotfill} \\ 0 & \cdots & 1 & -2+x_{n-2}h^2 & & 1 & 0 \\ 0 & \cdots & 0 & 1 & & -2+x_{n-1}h^2 & -y_n \end{array}\right]$$

Glavna diagonala matrike $M \in \mathbb{R}^{(n-1)\times(n-1)}$ je podana z vektorjem $d$, ki ima elemente

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

Prva nad- in poddiagonala sta podani z vektorjema $u$ in $l$, ki imata za elemente $n-2$ enojk. Desna stran sistema je podana z vektorjem $b$.

# Naloga 7.1

## Algoritem: tridiagonalna Gaussova eliminacija

$$\left[\begin{array}{ccccc|c} d_1 & u_1 & 0 & \cdots & 0 & b_1 \\ l_1 & d_2 & u_2 & \cdots & 0 & b_2 \\ \multicolumn{6}{c}{\dots\dots\dots\dots\dots\dots\dots\dots\dots} \\ 0 & \cdots & l_{n-2} & d_{n-1} & u_{n-1} & b_{n-1} \\ 0 & \cdots & 0 & l_{n-1} & d_n & b_n \end{array}\right] \to \left[\begin{array}{ccccc|c} \delta_1 & u_1 & 0 & \cdots & 0 & \beta_1 \\ 0 & \delta_2 & u_2 & \cdots & 0 & \beta_2 \\ \multicolumn{6}{c}{\dots\dots\dots\dots\dots\dots\dots\dots\dots} \\ 0 & \cdots & 0 & \delta_{n-1} & u_{n-1} & \beta_{n-1} \\ 0 & \cdots & 0 & 0 & \delta_n & \beta_n \end{array}\right]$$

$$\delta_1 = d_1, \quad \delta_k = d_k - u_{k-1} l_{k-1}/\delta_{k-1}, \quad k = 2, \dots, n$$

$$\beta_1 = b_1, \quad \beta_k = b_k - u_{k-1} l_{k-1}/\delta_{k-1}, \quad k = 2, \dots, n$$

## Algoritem: vzvratno vstavljanje

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

Zapišite program, ki dobi kot vhodne podatke štiri vektorje $d$, $u$, $l$ in $b$, ter poda rešitev kot vektor $y$. Sistem rešite z uporabo Gaussove eliminacije ne da bi zapisali polno obliko matrike $M$.

# Naloga 7.1

## Funkcijska m-datoteka trisys.m

```
% y = trisys(u,d,l,b)
% d, u, l :  glavna diagonala ter prva nad- in poddiagonala
% b :  vektor desnih strani
% y :  resitev sistema
function y = 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;
y(n) = b(n)/d(n);
for k = (n-1):-1:1
  y(k) = (b(k)-u(k)*y(k+1))/d(k);
end;
```

# Naloga 7.1

## Glavni program

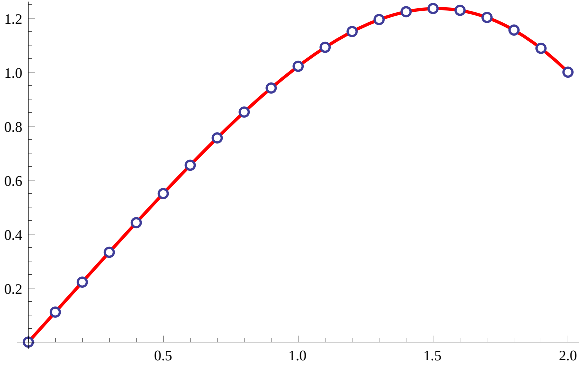
```
% Priprava podatkov
d = [];
n = 20;
x0 = 0; xend = 2;
y0 = 0; yend = 1;
h = (xend-x0)/n;
d = -2+(1:n-1)*h^3;
l = ones(1,n-2); u = l;
b = zeros(size(d)); b(1) = -y0; b(end)= -yend;

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

# Naloga 7.1

## Primerjava numerične in točne rešitve

```
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);
```



# Naloga 7.1

Nalogo rešite še z levim deljenjem, tako da z ukazom `M = sparse(I,J,V)` sestavite redko matriko sistema $M$.

```
% Konstrukcija redke matrike
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);

% Resevanje sistema z levim deljenjem
b = zeros(n-1,1);
b(1) = -y0;
b(end) = -yend;
Y = M\b;
Y = [y0,Y',yend];
```

# Naloga 8.1

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

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

Sistem rešite z Jacobijevo in Gauss-Seidlovo iteracijo, kjer najprej preverite, da je maksimalna lastna vrednost po absolutni vrednosti manjša od 1 za obe iteracijski matriki. Uporabite začetni približek $x^{(0)} = \begin{bmatrix} 2/3 \\ 1/3 \end{bmatrix}$.

Matriko $A$ razcepimo takole:

$$A = D - L - U,$$

kjer je $D$ diagonalni del matrike, $-U$ zgornji in $-L$ spodnji trikotnik matrike $A$.

# Naloga 8.1

## Jacobijeva iteracija:

$$
\begin{aligned}
(D-L-U)x &= b \\
Dx &= (L+U)x + b \\
x^{(k+1)} &= \underbrace{D^{-1}(L+U)}_{R_J}\, x^{(k)} + \underbrace{D^{-1}b}_{c_J}
\end{aligned}
$$

## Gauss-Seidlova iteracija:

$$
\begin{aligned}
(D-L-U)x &= b \\
(D-L)x &= Ux + b \\
x^{(k+1)} &= \underbrace{(D-L)^{-1}U}_{R_{GS}}\, x^{(k)} + \underbrace{(D-L)^{-1}b}_{c_{GS}}
\end{aligned}
$$

# Naloga 8.1

Jacobijeva in Gauss-Seidlova iteracija konvergirata k rešitvi enačbe $Ax = b$ pri poljubnem začetnem približku $x^{(0)}$ in pri poljubnem stolpcu desnih strani $b$ natanko tedaj, ko je po absolutni vrednosti največja lastna vrednost iteracijske matrike $R_J$ oz. $R_{GS}$ manjša od 1.

$$\max_{\lambda \in \Lambda(R_J)} |\lambda| < 1 \qquad \max_{\lambda \in \Lambda(R_{GS})} |\lambda| < 1$$

Iteracijo zaključimo, ko je $\|x^{(k+1)} - x^{(x)}\| < \varepsilon$.