

Simulacija sistema opisanog matematičkim modelom

Modeliranje i simulacija sistema

Uvod

- Razmatra se: simulacija sistema opisanog matematičkim modelom
- Zadatak: računanje izlaza modela na osnovu poznatih ulaza
- Rešenje: analitičko i/ili numeričko (računarski program)
- Tipovi(opis) modela i numeričko rešavanje:
 - Linearan sistem algebarskih jednačina
 - Nelinearan sistem algebarskih jednačina
 - Sistem običnih diferencijalnih jednačina (nelinearan)
 - Sistem linearnih običnih diferencijalnih jednačina

Numeričko rešavanje sistema algebarskih jednačina

- Numeričko rešavanje sistema linearnih algebarskih jednačina
 - *Metoda najmanjih kvadrata*
- Numeričko rešavanje sistema nelinearnih algebarskih jednačina
 - *Gaus-Njutnov postupak*
 - *Gradijentni algoritam*
 - *Levenberg–Markartov algoritam*

Numeričko rešavanje sistema linearnih algebarskih jednačina

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2$$

...

$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m$$

- x_1, x_2, \dots, x_n – nepoznate veličine, a_{ij} – zadati koeficijenti, b_i – zadati slobodni članovi.

Numeričko rešavanje sistema linearnih algebarskih jednačina

- Matrični oblik $Ax = b$

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$$

1. jedno rešenje – regularan sistem tj. $\det(A) \neq 0$ - broj jednačina jednak broju nepoznatih ($n = m$)
2. Ako je ($m < n$) - rešenje obično nije jednoznačno
3. Ako je ($m > n$) - sistem ne mora da ima rešenje

Primer: Upotrebom
Julia-e rešiti sistem
jednačina

$$\begin{aligned}x_1 + 2x_2 - x_3 &= -2.5 \\ -x_1 + 3x_2 + 2x_3 &= 0.5 \\ -x_1 - x_2 + x_3 &= 1.5\end{aligned}$$

$$Ax = b$$

onda je

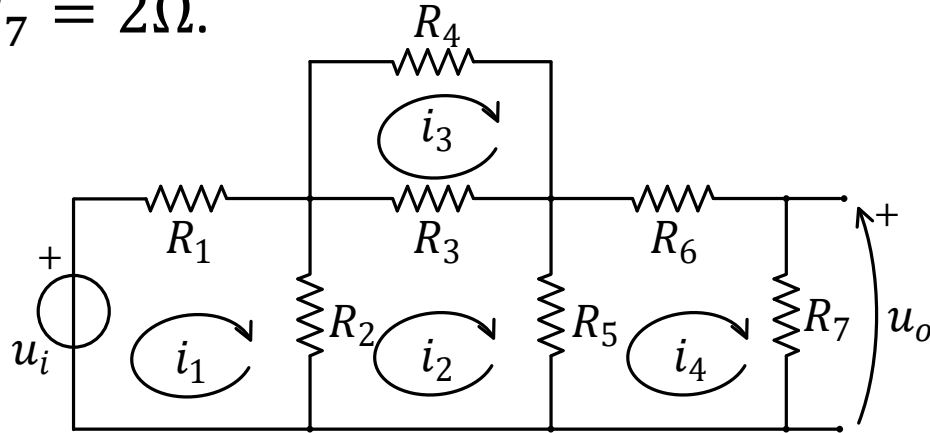
$$x = A^{-1}b$$

```
julia> A = [1 2 -1; -1 3 2; -1 -1 1];  
julia> b = [-2.5; 0.5; 1.5];
```

```
julia> x = A\b  
3-element Array{Float64,1}:  
 2.4999999999999996  
 -1.0  
 2.9999999999999996
```

```
julia> x= round.(x,digits=2)  
3-element Array{Float64,1}:  
 2.5  
 -1.0  
 3.0
```

Primer: Upotrebom Julia-e za električno kolo sa slike odrediti napon na izlazu u_o kada je ulazni napon $u_i = 12V$. Otpornosti otpornika su $R_1 = 0.5\Omega$, $R_2 = 5\Omega$, $R_3 = 3\Omega$, $R_4 = 6\Omega$, $R_5 = 4\Omega$, $R_6 = 10\Omega$, $R_7 = 2\Omega$.



$$(R_1 + R_2)i_1 - R_2i_2 = u_i$$

$$-R_2i_1 + (R_2 + R_3 + R_5)i_2 - R_3i_3 - R_5i_4 = 0$$

$$-R_3i_2 + (R_3 + R_4)i_3 = 0$$

$$-R_5i_2 + (R_5 + R_6 + R_7)i_4 = 0$$

$$\begin{bmatrix} R_1 + R_2 & -R_2 & 0 & 0 \\ -R_2 & R_2 + R_3 + R_5 & -R_3 & -R_5 \\ 0 & -R_3 & R_3 + R_4 & 0 \\ 0 & -R_5 & 0 & R_5 + R_6 + R_7 \end{bmatrix} \cdot \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \end{bmatrix} = \begin{bmatrix} u_i \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

```

Ui=12;
R1=0.5; R2=5; R3=3; R4=6; R5=4; R6=10; R7=2;
R=[ R1+R2 -R2    0    0
    -R2  R2+R3+R5 -R3  -R5
      0   -R3    R3+R4  0
      0   -R5     0  R5+R6+R7];
U=[Ui;0;0;0];
format rational

I = R\U      # rešavanje sistema jednačina R*I=U
Uo = I[4]*R7  # izlazni napon

```

Izvršavanje koda daje

```

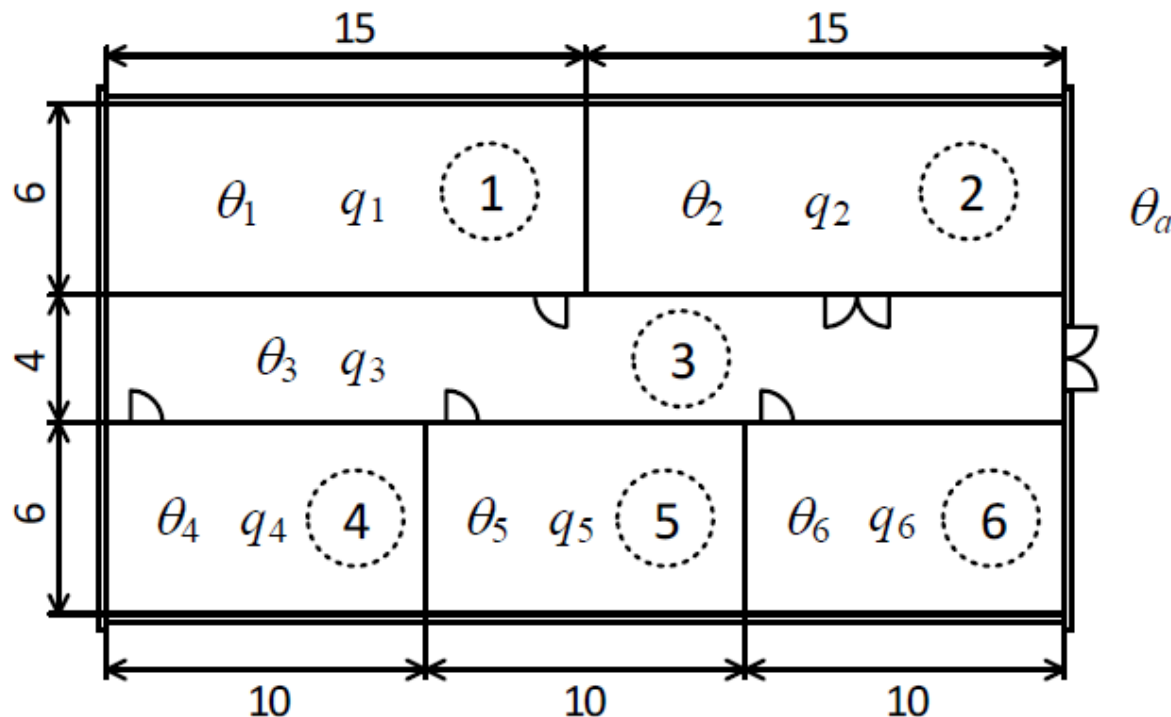
I =
      4
      2
     2/3
     1/2

Uo =
      1

```

Primer - Grejanje zgrade

- Napisati programski kod za izračunavanje potrebne snage grejača da bi u prostorijama bila temperatura od 25°C dok je spoljašnja temperatura 0°C

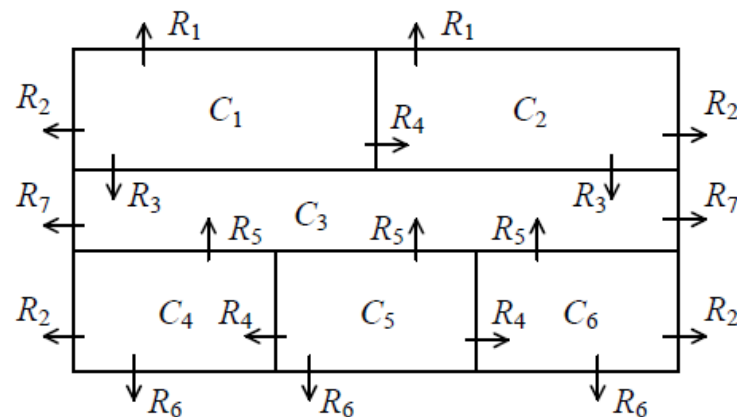


Matematički model

$$A\bar{\theta} + \bar{Q} + b\theta_a = 0$$

$$\bar{\theta} = \begin{bmatrix} \bar{\theta}_1 \\ \bar{\theta}_2 \\ \bar{\theta}_3 \\ \bar{\theta}_4 \\ \bar{\theta}_5 \\ \bar{\theta}_6 \end{bmatrix}$$

$$\bar{Q} = \begin{bmatrix} \bar{q}_1 \\ \bar{q}_2 \\ \bar{q}_3 \\ \bar{q}_4 \\ \bar{q}_5 \\ \bar{q}_6 \end{bmatrix}$$



$$A = \begin{bmatrix} a_1 & \frac{1}{R_4} & \frac{1}{R_3} & 0 & 0 & 0 \\ \frac{1}{R_4} & a_2 & \frac{1}{R_3} & 0 & 0 & 0 \\ \frac{1}{R_3} & \frac{1}{R_3} & a_3 & \frac{1}{R_5} & \frac{1}{R_5} & \frac{1}{R_5} \\ 0 & 0 & \frac{1}{R_5} & a_4 & \frac{1}{R_4} & 0 \\ 0 & 0 & \frac{1}{R_5} & \frac{1}{R_4} & a_5 & \frac{1}{R_4} \\ 0 & 0 & \frac{1}{R_5} & 0 & \frac{1}{R_4} & a_6 \end{bmatrix}$$

$$a_1 = -\left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{R_4}\right)$$

$$a_2 = -\left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{R_4}\right)$$

$$a_3 = -\left(\frac{2}{R_3} + \frac{3}{R_5} + \frac{2}{R_7}\right)$$

$$a_4 = -\left(\frac{1}{R_2} + \frac{1}{R_4} + \frac{1}{R_5} + \frac{1}{R_6}\right)$$

$$a_5 = -\left(\frac{2}{R_4} + \frac{1}{R_5} + \frac{1}{R_6}\right)$$

$$a_6 = -\left(\frac{1}{R_2} + \frac{1}{R_4} + \frac{1}{R_5} + \frac{1}{R_6}\right)$$

$$b = \begin{bmatrix} \frac{1}{R_1} + \frac{1}{R_2} \\ \frac{1}{R_1} + \frac{1}{R_2} \\ \frac{2}{R_7} \\ \frac{1}{R_2} + \frac{1}{R_6} \\ \frac{1}{R_6} \\ \frac{1}{R_2} + \frac{1}{R_6} \end{bmatrix}$$

```
function zgradaStatModel()
```

```
    Fb = 0.5/1.55; # betonski zid (spoljašnji)
```

```
    Fbt = 0.2/1.55; # betonski zid-tanak (unutrašnji)
```

```
    Fp = 0.01/0.8 + 0.02/0.026; # prozor: 2 x staklo 5mm + vazduh 2cm
```

```
    R1b = Fb/(15 * 1.5); R1p = Fp/(15 * 1.5);
```

```
    R1 = 1/(1/R1b+1/R1p); P1 = 1/R1;
```

```
    R2 = Fb/(6 * 3); P2 = 1/R2;
```

```
    R3 = Fbt/(15 * 3); P3 = 1/R3;
```

```
    R4 = Fbt/(6 * 3); P4 = 1/R4;
```

```
    R5 = Fbt/(10 * 3); P5 = 1/R5;
```

```
    R6b = Fb/(10 * 1.5); R6p = Fp/(10 * 1.5);
```

```
    R6 = 1/(1/R6b+1/R6p); P6 = 1/R6;
```

```
    R7 = Fp/(4 * 3); P7 = 1/R7;
```

```
    a1 = -(P1+P2+P3+P4);
```

```
    a2 = a1;
```

```
    a3 = -(2*P3+2*P7+3*P5);
```

```
    a4 = -(P2+P4+P5+P6);
```

```
    a5 = -(2*P4+P5+P6);
```

```
    a6 = a4;
```

```
    ...
```

```

...
A = [ a1 P4 P3 0 0 0
      P4 a2 P3 0 0 0
      P3 P3 a3 P5 P5 P5
      0 0 P5 a4 P4 0
      0 0 P5 P4 a5 P4
      0 0 P5 0 P4 a6 ];
b = [ P1+P2
      P1+P2
      2*P7
      P2+P6
      P6
      P2+P6 ];
return A,b;
end

```

```

julia> include("zgradaStatModel.jl")
zgradaStatModel (generic function with 1 method)

```

```

julia> A,b = zgradaStatModel();
julia> T = [25; 25; 25; 25; 25; 25];
julia> Ta = 0;
julia> Q=-A*(T.+273)-b*(Ta+273)

```

```
julia> b
```

```
6-element Array{Float64,1}:
```

```
154.33228782287824  
154.33228782287824  
30.701107011070107  
121.48819188191882  
65.68819188191883  
121.48819188191882
```

```
julia> A
```

```
6×6 Array{Float64,2}:
```

```
-642.582  139.5   348.75   0.0   0.0   0.0  
139.5   -642.582  348.75   0.0   0.0   0.0  
348.75   348.75  -1425.7  232.5  232.5  232.5  
0.0      0.0    232.5 -493.488 139.5   0.0  
0.0      0.0    232.5 139.5  -577.188 139.5  
0.0      0.0    232.5 0.0    139.5  -493.488
```

```
julia> round.(Q,digits=2)
```

```
6-element Array{Float64,1}:
```

```
3858.31  
3858.31  
767.53  
3037.2  
1642.2  
3037.2
```

Metoda najmanjih kvadrata

- Polazi se od sistema

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2$$

...

$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m$$

pri čemu je

$$m > n$$

Metoda najmanjih kvadrata

- Ako se sistem jednačina prepíše u oblik

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n - b_1 = e_1$$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n - b_2 = e_2$$

...

$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n - b_m = e_m$$

- Rešenje se dobija minimizacijom kriterijuma

$$J = \frac{1}{2} \sum_{k=1}^m e_k^2$$

$$\min_x \mathcal{J} = \min_x \frac{1}{2} \sum_{k=1}^m e_k^2 = \min_x \frac{1}{2} \sum_{k=1}^m (a_{k1}x_1 + a_{k2}x_2 + \cdots + a_{kn}x_n - b_k)^2$$

Potreban uslov $\nabla \mathcal{J} = 0$, tj.

$$\frac{\partial \mathcal{J}}{\partial x_i} = 0, \quad i = 1, 2, \dots, n.$$

Na primer, za $i = 2$

$$\frac{\partial \mathcal{J}}{\partial x_2} = \frac{1}{2} \cdot 2 \sum_{k=1}^m (a_{k1}x_1 + a_{k2}x_2 + \cdots + a_{kn}x_n - b_k) a_{k2}$$

pa se dobija

$$\frac{\partial \mathcal{J}}{\partial x_i} = \sum_{k=1}^m a_{ki} e_k = 0$$

$$\frac{\partial \mathcal{J}}{\partial x_i} = \sum_{k=1}^m a_{ki} e_k = 0$$

$$\frac{\partial \mathcal{J}}{\partial x_i} = [a_{1i} \quad a_{2i} \quad \dots \quad a_{mi}] \begin{bmatrix} e_1 \\ e_2 \\ \dots \\ e_m \end{bmatrix} = 0$$

$$\nabla \mathcal{J} = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \dots & \dots & \dots & \dots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ \dots \\ e_m \end{bmatrix} = \mathbf{A}^T (\mathbf{A}\mathbf{x} - \mathbf{b}) = \mathbf{0}$$

Sređivanjem dobijamo

$$\mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{A}^T \mathbf{b}$$

množenjem $(\mathbf{A}^T \mathbf{A})^{-1}$ sa leve strane dobija se

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}.$$

Opisani postupak se naziva **METODA NAJMANJIH KVADRATA**.

Primer: Upotrebom Julia-e rešiti sistem jednačina

$$x_1 + 2x_2 - x_3 = -2.5$$

$$-x_1 + 3x_2 + 2x_3 = 0.5$$

$$-x_1 - x_2 + x_3 = 1.5$$

$$x_1 + x_2 + x_3 = 4$$

4 jednačine sa 3 nepoznate i ne može se odrediti rešenje tako da su sve postavljene jednačine ispunjene

Cilj – da levi delovi jednačina budu što je moguće bliži zadatim slobodnim članovima

```
julia> A = [1 2 -1; -1 3 2; -1 -1 1; 1 2 3];
```

```
julia> b = [-2.5; 0.5; 1.5; 4];
```

```
julia> x = A\b
```

```
3-element Array{Float64,1}:
```

```
0.6238317757009348
```

```
-0.7001557632398757
```

```
1.5950155763239875
```

```
julia> x=round.(x,digits=4)
```

```
3-element Array{Float64,1}:
```

```
0.6238
```

```
-0.7002
```

```
1.595
```

```
julia> A*x
```

```
4-element Array{Float64,1}:
```

```
-2.3716
```

```
0.46559999999999998
```

```
1.6714
```

```
4.0084
```

Do istog rezultata $x = \text{inv}(A' * A) * A' * b;$

- Ako se pretpostavi neko drugo rešenje

```
julia> e=A*x-b
```

```
julia> J=e'*e
```

```
julia> x2 = [0.62;-0.7;1.59];
```

```
julia> e2 = A*x2 - b;
```

```
julia> J2 = e2'*e2;
```

```
julia> J2-J>0
```

```
true
```

- rezultat potvrđuje da je dobijeno odstupanje veće (ans=true).