Tiesinių lygčių sistemų sprendimas (2)

http://oras.if.ktu.lt/moduliai/T170B115/Skaitiniai inzinerijos metodai su MATLAB (KP RG RB)2009

F7.pdf - 4.1-4.5 poskyriai

http://oras.if.ktu.lt/moduliai/T170B115 /Skaitiniai inzinerijos metodai(RB) 2006

I.1.2.pdf

Prisiminkime: kintamųjų eliminavimu paremti algoritmai

Gauso algoritmas

$$\begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} \begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \end{bmatrix} = \begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \end{bmatrix} \rightarrow \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{bmatrix} \begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \end{bmatrix} = \begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \end{bmatrix} \rightarrow x_{n,1:p},\ x_{n-1,1:p},\ldots,x_{1,1:p}$$

$$\mathbf{A} \quad \mathbf{x} = \mathbf{b}$$

Gauso-Žordano algoritmas

$$\begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} \begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \end{bmatrix} = \begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \end{bmatrix} = \begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \end{bmatrix}$$

$$\mathbf{A} \quad \mathbf{x} = \mathbf{b}$$

$$\mathbf{A}\,\mathbf{x} = \mathbf{b}\ ; \qquad \mathbf{A}^{-1}\mathbf{A}\,\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}; \qquad \mathbf{x} = \mathbf{A}^{-1}\mathbf{b};$$

Atvirkštinės matricos algoritmas

$$\begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A} \quad \mathbf{X} \quad \mathbf{E}$$

Taikome Gauso-Žordano algoritmą

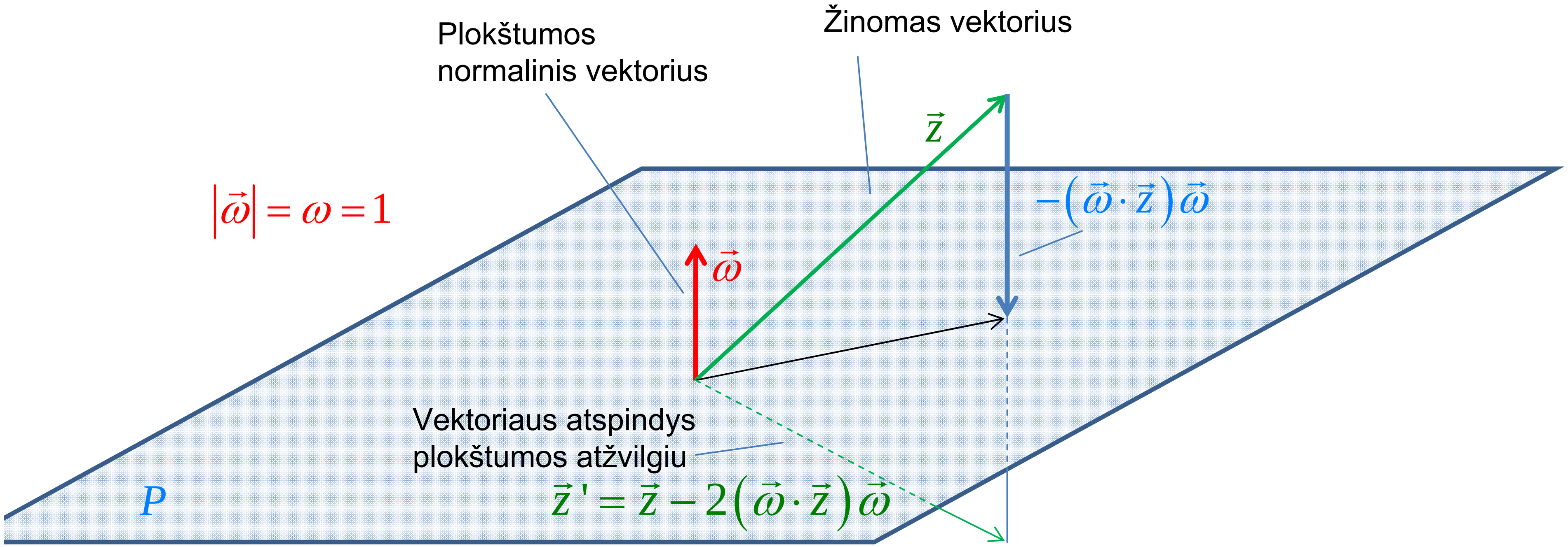
$$\Longrightarrow \qquad \mathbf{A}^{-1} = \mathbf{X}$$

Tiesioginiu kintamųjų eliminavimu paremti metodai nėra dažnai vartojami:

-Jeigu viename iš algoritmo vykdymo žingsnių vedančio elemento skaitinė reikšmė maža, galimos ženklios apvalinimo paklaidos. To išvengiama taikant *atspindžio algoritmą (QR algoritmas, Hausholderio algoritmas)*;

- Norint išspręsti lygčių sistemą su kitokiomis laisvųjų narių reikšmėmis, reikia iš naujo įvykdyti visą algoritmą. Iš kintamųjų eliminavimu pagrįstų metodų šio trūkumo neturi tik *atvirkštinės matricos algoritmas*, tačiau jis labai imlus skaičiavimams. Šią problemą sprendžia *skaidos metodai* (t.y. matrica išskaidoma trikampiais daugikliais)

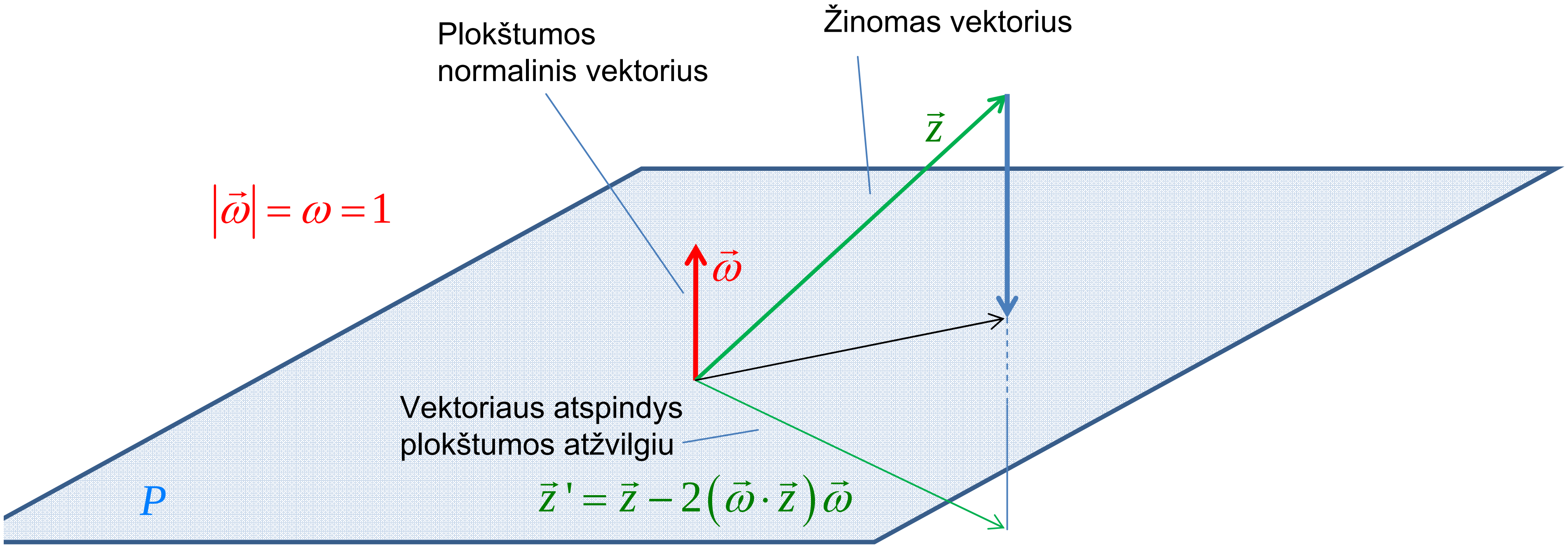
Atspindžio algoritmo idėja: vektoriaus atspindys plokštumos atžvilgiu 3D erdvėje (1)



$$\vec{z}\,' = \vec{z} - 2\vec{\omega}(\vec{\omega}\cdot\vec{z}) = \begin{Bmatrix} z_x \\ z_y \\ z_z \end{Bmatrix} - 2\begin{Bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix}\left(\begin{Bmatrix} \omega_x & \omega_y & \omega_z \end{Bmatrix}\begin{Bmatrix} z_x \\ z_y \\ z_z \end{Bmatrix}\right) =$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{Bmatrix} z_x \\ z_y \\ z_z \end{Bmatrix} - 2\begin{Bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix}\begin{Bmatrix} \omega_x & \omega_y & \omega_z \end{Bmatrix}\begin{Bmatrix} z_x \\ z_y \\ z_z \end{Bmatrix} = [\mathbf{E}]\{\mathbf{z}\} - 2\{\boldsymbol{\omega}\}\{\boldsymbol{\omega}\}^T\{\mathbf{z}\} = \left([\mathbf{E}] - 2\{\boldsymbol{\omega}\}\{\boldsymbol{\omega}\}^T\right)\{\mathbf{z}\}$$

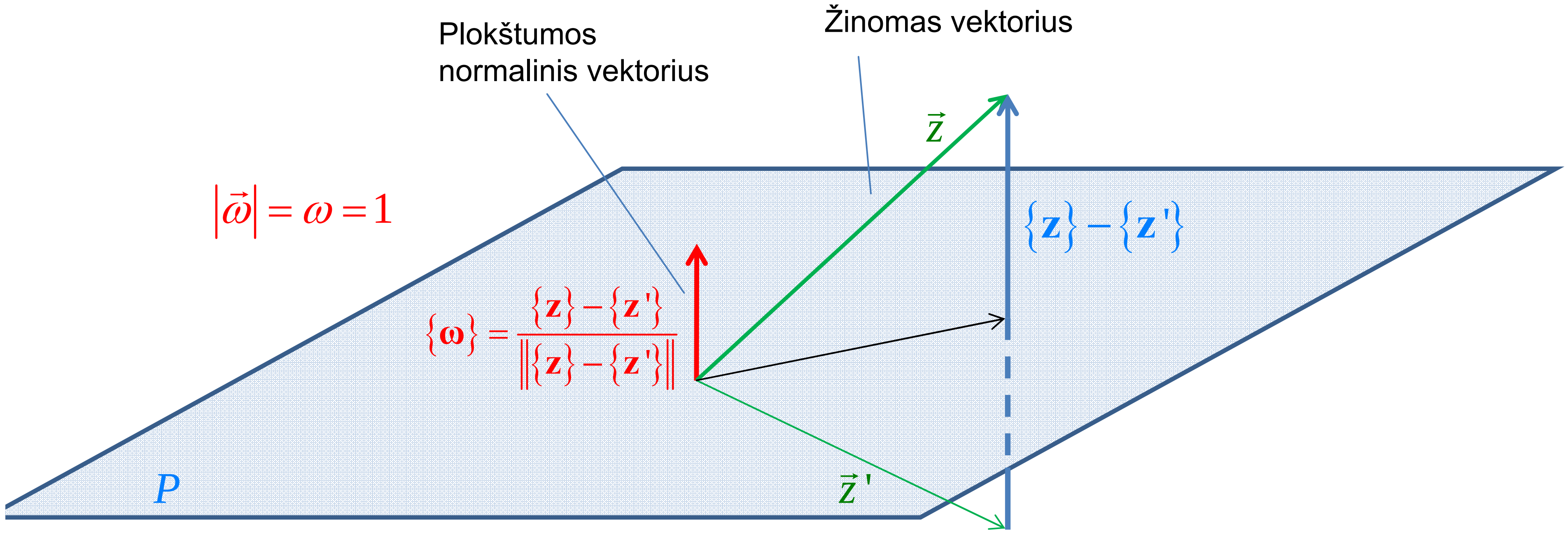
Atspindžio algoritmo idėja: vektoriaus atspindys plokštumos atžvilgiu 3D erdvėje (2)



Pagal formulę $\{\mathbf{z}'\} = [\mathbf{Q}]\{\mathbf{z}\}$, $[\mathbf{Q}] = [\mathbf{E}] - 2\{\boldsymbol{\omega}\}\{\boldsymbol{\omega}\}^T$

Vektorius $\{\mathbf{z}\}$ pertvarkomas į jo atspindį atžvilgiu plokštumos, kurios normalės vektorius yra $\{\boldsymbol{\omega}\}$

Atspindžio algoritmo idėja: vektoriaus atspindys plokštumos atžvilgiu 3D erdvėje (3)



Jeigu žinome $\{\mathbf{z}\}$ ir jo atspindį $\{\mathbf{z}'\}$, plokštumos normalę galime apskaičiuoti taip:

$$\{\boldsymbol{\omega}\} = \frac{\{\mathbf{z}\} - \{\mathbf{z}'\}}{\left\| \{\mathbf{z}\} - \{\mathbf{z}'\} \right\|}$$

$\left\| \{\mathbf{v}\} \right\| = \sqrt{\sum_{i=1}^{n} v_i^{\,2}}$ - vektoriaus Euklido norma (“ilgis”), n- erdvės matavimų skaičius

<u>Uždavinys:</u> koks turėtų būti $\{\boldsymbol{\omega}\}$, kad atspindėto vektoriaus $\{\mathbf{z}'\}$ visos koordinatės būtų =0, išskyrus pirmąją?

$$\{\mathbf{z}'\} = sign(z_1)\left\|\{\mathbf{z}\}\right\|\begin{Bmatrix}1\\0\\0\end{Bmatrix}$$

- atspindys turi būti tokio pat ilgio, kaip ir vektorius $\{\mathbf{z}\}$

$$\{\boldsymbol{\omega}\} = \frac{\{\mathbf{z}\} - \{\mathbf{z}'\}}{\left\|\{\mathbf{z}\} - \{\mathbf{z}'\}\right\|}$$

$$[\mathbf{Q}] = [\mathbf{E}] - 2\{\boldsymbol{\omega}\}\{\boldsymbol{\omega}\}^T$$

3D atveju suformuluotos ir geometriškai paaiškintos priklausomybės pritaikomos n-matėje tiesinėje erdvėje:

$$\{\mathbf{z}'\} = sign(z_1)\left\|\{\mathbf{z}\}\right\|\begin{Bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{Bmatrix} \qquad \{\boldsymbol{\omega}\} = \frac{\{\mathbf{z}\} - \{\mathbf{z}'\}}{\left\|\{\mathbf{z}\} - \{\mathbf{z}'\}\right\|} \qquad [\mathbf{Q}] = [\mathbf{E}] - 2\{\boldsymbol{\omega}\}\{\boldsymbol{\omega}\}^T$$

- Taikant formulę $\{\mathbf{z}'\} = [\mathbf{Q}]\{\mathbf{z}\}$ bet koks n-matis vektorius $\{\mathbf{z}\}$ gali būti pertvarkytas į tos pačios normos („ilgio") vektorių $\{\mathbf{z}'\}$, turinčio tik pirmąją nenulinę koordinatę;

- Kiekviena $\{\mathbf{z}'\}$ koordinatė yra vektoriaus $\{\mathbf{z}\}$ koordinačių tiesinė kombinacija. Tai seka iš matricų daugybos veiksmo apibrėžimo;

Kiekviena atspindžio matrica yra simetrinė ir ortogonalioji:

$$[\mathbf{Q}] = [\mathbf{E}] - 2\{\boldsymbol{\omega}\}\{\boldsymbol{\omega}\}^T;$$

$$[\mathbf{Q}]^T = \left([\mathbf{E}] - 2\{\boldsymbol{\omega}\}\{\boldsymbol{\omega}\}^T\right)^T = [\mathbf{E}]^T - 2\left(\{\boldsymbol{\omega}\}^T\right)^T\{\boldsymbol{\omega}\}^T = [\mathbf{E}] - 2\{\boldsymbol{\omega}\}\{\boldsymbol{\omega}\}^T;$$

$$[\mathbf{Q}]^T[\mathbf{Q}] = \left([\mathbf{E}] - 2\{\boldsymbol{\omega}\}\{\boldsymbol{\omega}\}^T\right)\left([\mathbf{E}] - 2\{\boldsymbol{\omega}\}\{\boldsymbol{\omega}\}^T\right) =$$

$$= [\mathbf{E}][\mathbf{E}] - 2\{\boldsymbol{\omega}\}\{\boldsymbol{\omega}\}^T[\mathbf{E}] - 2[\mathbf{E}]\{\boldsymbol{\omega}\}\{\boldsymbol{\omega}\}^T + 4\{\boldsymbol{\omega}\}\{\boldsymbol{\omega}\}^T\{\boldsymbol{\omega}\}\{\boldsymbol{\omega}\}^T = [\mathbf{E}];$$

$$\{\boldsymbol{\omega}\}^T\{\boldsymbol{\omega}\} = 1$$

$$[\mathbf{Q}]^{-1} = [\mathbf{Q}]^T$$

Atspindžio algoritmo taikymas lygčių sistemos pertvarkymui 1

- Duota lygčių sistemos koeficientų ir laisvųjų narių matrica:

$$[\mathbf{A}] = \left[\begin{array}{cccc|c} 1 & 1 & 1 & 1 & 2 \\ 1 & -1 & -1 & 1 & 0 \\ 2 & 1 & -1 & 2 & 9 \\ 3 & 1 & 2 & -1 & 7 \end{array}\right]$$

- Kiekvienas stulpelis laikomas vektoriumi, kuriam gali būti pritaikytas *atspindžio pakeitimas*;

Atspindžio algoritmo taikymas lygčių sistemos pertvarkymui 2

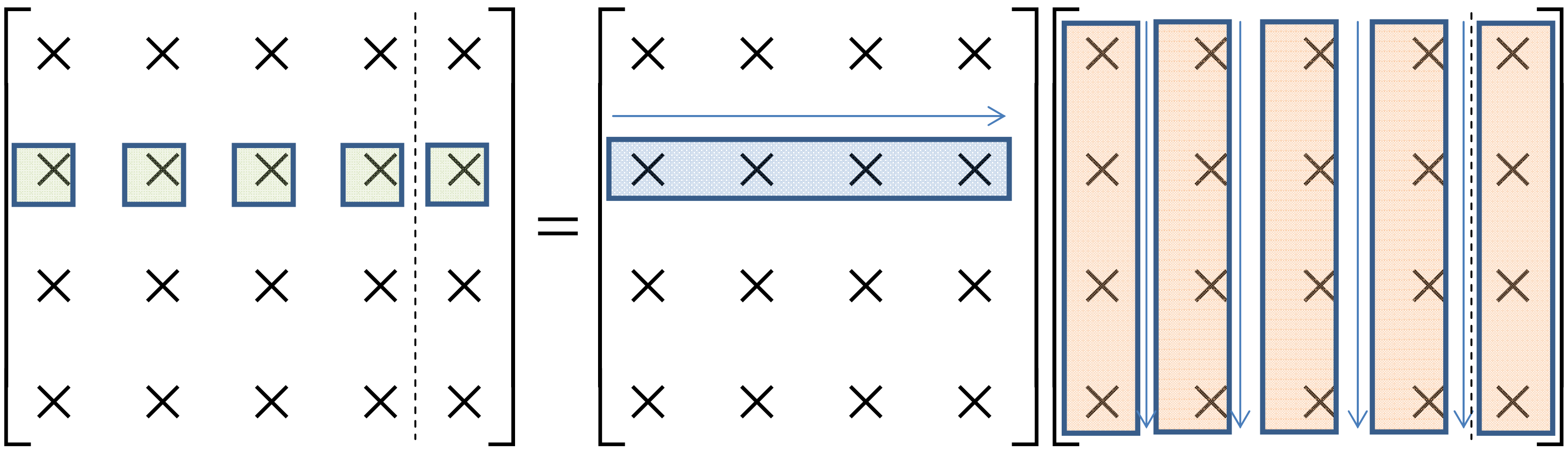
- **Taikant formulę [A’]= [Q] [A] , vienas ir tas pats atspindžio pakeitimas pritaikomas kiekvienam stulpeliui;**

$$[\mathbf{A}'] = [\mathbf{Q}][\mathbf{A}] = [\mathbf{Q}]\left[\left\{a_{1:n,1}\right\}, \quad \left\{a_{1:n,2}\right\}, \quad \dots \quad , \quad \left\{a_{1:n,n}\right\}\right] =$$

$$= \left[[\mathbf{Q}]\left\{a_{1:n,1}\right\}, \quad [\mathbf{Q}]\left\{a_{1:n,2}\right\}, \quad \dots \quad , \quad [\mathbf{Q}]\left\{a_{1:n,n}\right\}\right]$$

Atspindžio algoritmo taikymas lygčių sistemos pertvarkymui 2

- Matrica [A’] gaunama, tiesiškai kombinuojant matricos [A] eilutes (t.y. sprendžiamos sistemos lygtis)



$$[A'] = [Q][A]$$

- Tai reiškia, kad matricomis [A’] ir [A] aprašomų lygčių sistemų sprendiniai yra tokie patys

Atspindžio algoritmo taikymas lygčių sistemos pertvarkymui 3

vektorius $\{\mathbf{z}\} = \begin{Bmatrix} 1 \\ 1 \\ 2 \\ 3 \end{Bmatrix}$

$$[\mathbf{A}] = \left[\begin{array}{cccc:c} \boxed{1} & 1 & 1 & 1 & 2 \\ \boxed{1} & -1 & -1 & 1 & 0 \\ \boxed{2} & 1 & -1 & 2 & 9 \\ \boxed{3} & 1 & 2 & -1 & 7 \end{array}\right]$$

Pertvarkomi visi stulpeliai, tačiau nuliai gaunami tik tame, pagal kuri buvo apskaiciuota matrica Q. Toks stulpelių pertvarkymas reiškia, kad lygtys buvo tiesiškai kombinuojamos

atspindetas vektorius $\{\mathbf{z}'\} = sign(1)\sqrt{1^2+1^2+2^2+3^2}\begin{Bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 3.873 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$

$$\{\boldsymbol{\omega}\} = \frac{\{\mathbf{z}\} - \{\mathbf{z}'\}}{\left\| \{\mathbf{z}\} - \{\mathbf{z}'\} \right\|}$$

$$\left[\mathbf{Q}^{(1)}\right] = [\mathbf{E}] - 2\{\boldsymbol{\omega}\}\{\boldsymbol{\omega}\}^T =$$

0.2582	0.2582	0.5164	0.7746
0.2582	0.9101	-0.1797	-0.2696
0.5164	-0.1797	0.6405	-0.5392
0.7746	-0.2696	-0.5392	0.1912

$$\left[\mathbf{A}^{(1)}\right] = \left[\mathbf{Q}^{(1)}\right] \begin{bmatrix} 1 & 1 & 1 & 1 & 2 \\ 1 & -1 & -1 & 1 & 0 \\ 2 & 1 & -1 & 2 & 9 \\ 3 & 1 & 2 & -1 & 7 \end{bmatrix} =$$

3.8730	1.2910	1.0328	0.7746	10.5862
0.0000	-1.1013	-1.0114	1.0785	-2.9886
0.0000	0.7974	-1.0228	2.1569	3.0228
-0.0000	0.6961	1.9658	-0.7646	-1.9658

Atspindžio algoritmo taikymas lygčių sistemos pertvarkymui 4

vektorius $\{\mathbf{z}\}$

$$[\mathbf{A}^{(1)}] = \left[\begin{array}{rrrr:r} 3.8730 & 1.2910 & 1.0328 & 0.7746 & 10.5862 \\ 0.0000 & \boxed{-1.1013} & -1.0114 & 1.0785 & -2.9886 \\ 0.0000 & \boxed{0.7974} & -1.0228 & 2.1569 & 3.0228 \\ -0.0000 & \boxed{0.6961} & 1.9658 & -0.7646 & -1.9658 \end{array}\right]$$

atspindetas vektorius $\{\mathbf{z}'\} = sign(-1.103)\sqrt{1.103^2 + 0.7974^2 + 0.6961^2}\begin{Bmatrix}1\\0\\0\end{Bmatrix} = \begin{Bmatrix}-1.5275\\0\\0\end{Bmatrix}$

$$\{\boldsymbol{\omega}\} = \frac{\{\mathbf{z}\} - \{\mathbf{z}'\}}{\left\|\{\mathbf{z}\} - \{\mathbf{z}'\}\right\|}$$

$$[\mathbf{Q}^{(2)}] = [\mathbf{E}] - 2\{\boldsymbol{\omega}\}\{\boldsymbol{\omega}\}^T = \begin{matrix} 0.7210 & -0.5220 & -0.4557 \\ -0.5220 & 0.0233 & -0.8526 \\ -0.4557 & -0.8526 & 0.2557 \end{matrix}$$

Pertvarkomas tik matricos blokas (2:n,2:n+1) (t.y. pirmoji lygtis nebekinta). Todėl erdvės, kurioje atliekamas posūkis, išmatavimas sumažėja.

$$\begin{matrix} 3.8730 & 1.2910 & 1.0328 & 0.7746 & 10.5862 \\ 0.0000 & & & & \\ 0.0000 & [\mathbf{Q}^{(2)}] & \boxed{\begin{matrix} -1.1013 & -1.0114 & 1.0785 & -2.9886 \\ 0.7974 & -1.0228 & 2.1569 & 3.0228 \\ 0.6961 & 1.9658 & -0.7646 & -1.9658 \end{matrix}} & & \\ -0.0000 & & & & \end{matrix} = \begin{matrix} 3.8730 & 1.2910 & 1.0328 & 0.7746 & 10.5862 \\ 0.0000 & -1.5275 & -1.0911 & -0.0000 & -2.8368 \\ 0.0000 & 0 & -1.1719 & 0.1393 & 3.3067 \\ -0.0000 & -0.0000 & 1.8356 & -2.5260 & -1.7179 \end{matrix}$$

$[\mathbf{A}^{(2)}]$

Atspindžio algoritmo taikymas lygčių sistemos pertvarkymui 5

vektorius $\{\mathbf{z}\}$

$$\left[\mathbf{A}^{(2)}\right]= \begin{array}{cccc:c} 3.8730 & 1.2910 & 1.0328 & 0.7746 & 10.5862 \\ 0.0000 & -1.5275 & -1.0911 & -0.0000 & -2.8368 \\ 0.0000 & 0 & \boxed{-1.1719} & 0.1393 & 3.3067 \\ -0.0000 & -0.0000 & \boxed{1.8356} & -2.5260 & -1.7179 \end{array}$$

atspindetas vektorius $\{\mathbf{z}'\} = sign(-1.1719)\sqrt{1.1719^2+1.8356^2}\begin{Bmatrix}1\\0\end{Bmatrix}=\begin{Bmatrix}-2.1778\\0\end{Bmatrix}$

$$\{\boldsymbol{\omega}\} = \frac{\{\mathbf{z}\}-\{\mathbf{z}'\}}{\left\|\{\mathbf{z}\}-\{\mathbf{z}'\}\right\|}$$

$$\left[\mathbf{Q}^{(3)}\right]=[\mathbf{E}]-2\{\boldsymbol{\omega}\}\{\boldsymbol{\omega}\}^T = \begin{array}{cc} 0.5381 & -0.8429 \\ -0.8429 & -0.5381 \end{array}$$

$$\begin{array}{ccccc} 3.8730 & 1.2910 & 1.0328 & 0.7746 & 10.5862 \\ 0.0000 & -1.5275 & -1.0911 & -0.0000 & -2.8368 \\ 0.0000 & 0 & \left[\mathbf{Q}^{(3)}\right] \boxed{-1.1719} & \boxed{0.1393} & \boxed{3.3067} \\ -0.0000 & -0.0000 & \boxed{1.8356} & \boxed{-2.5260} & \boxed{-1.7179} \end{array} = \begin{array}{ccccc} 3.8730 & 1.2910 & 1.0328 & 0.7746 & 10.5862 \\ 0.0000 & -1.5275 & -1.0911 & -0.0000 & -2.8368 \\ 0.0000 & 0.0000 & -2.1778 & 2.2040 & 3.2274 \\ -0.0000 & 0.0000 & -0.0000 & 1.2418 & -1.8628 \end{array}$$

$$\left[\mathbf{A}^{(3)}\right]$$

Atspindžio algoritmo taikymas lygčių sistemos pertvarkymui 6

$$\left[\mathbf{A}^{(3)}\right] = \left[\begin{array}{cccc|c} 3.8730 & 1.2910 & 1.0328 & 0.7746 & 10.5862 \\ 0.0000 & -1.5275 & -1.0911 & -0.0000 & -2.8368 \\ 0.0000 & 0.0000 & -2.1778 & 2.2040 & 3.2274 \\ -0.0000 & 0.0000 & -0.0000 & 1.2418 & -1.8628 \end{array}\right]$$

Atliekamas Gauso algoritmo atvirkštinis etapas

x =

2.5000
4.0000
-3.0000
-1.5000

Atspindžio algoritmas

```
A1=[A,b]

for i=1:n-1
    z=A1(i:n,i);
    zp=zeros(size(z));
    zp(1)=sign(z(1))*norm(z);
    omega=(z-zp); omega=omega/norm(omega);
    Q=eye(n-i+1)-2*omega*omega‘;
    A1(i:n,:)=Q*A1(i:n,:)
end


% Atvirkstinis etapas toks pats, kaip ir Gauso algoritmo:

x=zeros(n,1);
x(n,:)=x(n,:)/A1(n,n);
for i=n-1:-1:1
    x(i,:)=(A1(i,n+1)-A1(i,i+1:n)*x(i+1:n,:))/A1(i,i);
end
```

Atspindžio algoritmas: apibendrinimai

- **Atspindžio algoritmas paremtas lygčių tiesiniu kombinavimu, siekiant gauti nulius žemiau matricos pagrindinės įstrižainės. Todėl *atspindžio algoritmas* yra *Gauso algoritmo* apibendrinimas;**
- **Gauso algoritme tiesinės kombinacijos gaunamos, pridedant ar atimant vedančią lygtį iš žemiau esančių. Atspindžio algoritme tiesinėse kombinacijose dalyvauja visos lygtys;**
- **Svarbiausia, kad po kiekvieno "atspindžio" matricos stulpelių normos nepakinta. Todėl tikėtina, kad lygčių tiesinės kombinacijos nesukurs stulpelių, kurių visi koeficientai absoliutiniu dydžiu yra labai maži**

QR skaida

Atspindžio algoritmas išskaido lygčių sistemos matricą į *ortogonalųjį* ir *trikampį* daugiklius:

$$\underbrace{\begin{bmatrix} \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} & \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} & \left[\mathbf{Q}^{(n-1)}\right] \end{bmatrix} \cdots \begin{bmatrix} 1 & \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} & \left[\mathbf{Q}^{(2)}\right] \end{bmatrix} \left[\mathbf{Q}^{(1)}\right]}_{[\mathbf{Q}]^T} [\mathbf{A}] = [\mathbf{R}]$$

Ortogonaliųjų matricų sandauga yra ortogonalioji matrica:

$$[\mathbf{Q}]^T [\mathbf{A}] = [\mathbf{R}]$$

$$[\mathbf{A}] = [\mathbf{Q}][\mathbf{R}]$$

QR skaida

Ortogonaliųjų matricų sandauga yra ortogonalioji matrica - įrodymas:

$$\left([\mathbf{C}_1][\mathbf{C}_2]\cdots[\mathbf{C}_k]\right)^T\left([\mathbf{C}_1][\mathbf{C}_2]\cdots[\mathbf{C}_k]\right)=$$

$$=\underbrace{[\mathbf{C}_k]^T\cdots\underbrace{[\mathbf{C}_2]^T\underbrace{[\mathbf{C}_1]^T[\mathbf{C}_1]}_{[\mathbf{E}]}[\mathbf{C}_2]}_{[\mathbf{E}]}\cdots[\mathbf{C}_k]}_{[\mathbf{E}]}=[\mathbf{E}]$$

Ortogonalūs daugikliai

Todėl matrica [Q] yra *ortogonalioji,* tačiau *nebe simetrinė(!)* matrica:

$$\underbrace{\begin{bmatrix}\begin{bmatrix}1&\cdots&0\\\vdots&\ddots&\vdots\\0&\cdots&1\end{bmatrix}&\begin{bmatrix}0&\cdots&0\\\vdots&\ddots&\vdots\\0&\cdots&0\end{bmatrix}\\\begin{bmatrix}0&\cdots&0\\\vdots&\ddots&\vdots\\0&\cdots&0\end{bmatrix}&\left[\mathbf{Q}^{(n-1)}\right]\end{bmatrix}\cdots\begin{bmatrix}1&\begin{bmatrix}0&\cdots&0\end{bmatrix}\\\begin{bmatrix}0\\\vdots\\0\end{bmatrix}&\left[\mathbf{Q}^{(2)}\right]\end{bmatrix}\left[\mathbf{Q}^{(1)}\right]}_{[\mathbf{Q}]^T}[\mathbf{A}]=[\mathbf{R}]$$

Skirtingų simetrinių matricų sandauga nėra simetrinė matrica

Matricos [Q] apskaičiavimas

$$[\mathbf{Q}]^T = \begin{bmatrix} \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} & \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} & \left[\mathbf{Q}^{(n-1)}\right] \end{bmatrix} \cdots \begin{bmatrix} 1 & \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} & \left[\mathbf{Q}^{(2)}\right] \end{bmatrix} \left[\mathbf{Q}^{(1)}\right]$$

$$[\mathbf{Q}] = \left[\mathbf{Q}^{(1)}\right] \begin{bmatrix} 1 & \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} & \left[\mathbf{Q}^{(2)}\right] \end{bmatrix} \cdots \begin{bmatrix} \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} & \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} & \left[\mathbf{Q}^{(n-1)}\right] \end{bmatrix}$$

Visi daugikliai yra simetrinės matricos:

$$\left[\mathbf{Q}^{(i)}\right] = \left[\mathbf{Q}^{(i)}\right]^T$$

QR skaidos taikymas tiesinių lygčių sistemų sprendimui

$$[\mathbf{A}]\{\mathbf{x}\} = \{\mathbf{b}\}; \quad [\mathbf{A}] = [\mathbf{Q}][\mathbf{R}]$$

$$[\mathbf{Q}][\mathbf{R}]\{\mathbf{x}\} = \{\mathbf{b}\}$$

$$[\mathbf{R}]\{\mathbf{x}\} = [\mathbf{Q}]^T \{\mathbf{b}\}$$

Trikampė matrica

- **Privalumas:** **Kartą apskaičiavę skaidos daugiklius, juos galime naudoti pakartotinai, esant vis kitokiam laisvųjų narių vektoriui;**
- **Privalumas:** **Trikampėje matricoje stulpelių normos lieka tokios pačios, kaip ir išeities matricoje**
- **Trūkumas:** **Daugikliai užima daugiau atminties, nei pradinė matrica**

QR skaidos algoritmas

$$\begin{bmatrix} \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} & \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} & \begin{bmatrix} \mathbf{Q}^{(i)} \end{bmatrix} \end{bmatrix}$$

```
disp(‘QR skaida:')
Q=eye(n);
for i=1:n-1
    z=A(i:n,i);
    zp=zeros(size(z));
    zp(1)=sign(z(1))*norm(z);
    omega=(z-zp); omega=omega/norm(omega);
    Qi=eye(n-i+1)-2*omega*omega‘;
    A(i:n,:)=Qi*A(i:n,:);
    Q=Q*[eye(i-1),zeros(i-1,n-i+1);zeros(n-i+1,i-1),Qi];
end

disp('Atvirkstinis zingsnis:')
b1=Q'*b
x=zeros(n,nb);
x(n,:)=b1(n,:)/A(n,n);
for i=n-1:-1:1
    x(i,:)=(b1(i,:)-A(i,i+1:n)*x(i+1:n,:))/A(i,i);
end
```

Kiti skaidos metodai: *LU algoritmas*

Skaidos metodais lygčių sistemos matrica *A* išskaidoma į dviejų trikampių matricų *L* ir *U* sandaugą

$$[\mathbf{A}]\{\mathbf{x}\} = \{\mathbf{b}\} \quad \Rightarrow \quad [\mathbf{L}]\,[\mathbf{U}]\,\mathbf{x} = \mathbf{b}$$

$$[\mathbf{L}] = \begin{bmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{21} & l_{22} & 0 & \cdots & 0 \\ l_{31} & l_{32} & l_{33} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{bmatrix} \qquad \{\mathbf{U}\} = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & u_{22} & u_{23} & \cdots & u_{2n} \\ 0 & 0 & u_{33} & \cdots & u_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & u_{nn} \end{bmatrix}$$

Išskaidyti galima, jeigu $\Delta_1 = |a_{11}| \neq 0 \qquad \Delta_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \neq 0 \qquad \cdots \qquad \Delta_n = \det(A) \neq 0$

Skaidinys nėra vienintelis, todėl galima priimti $l_{11}=\ldots=l_{nn}=1$

Taikant *LU algoritmą, lygčių sistema sprendžiama taip:*

$$[\mathbf{A}]\{\mathbf{x}\} = \{\mathbf{b}\}$$

Apskaičiuojami trikampiai daugikliai

$$[\mathbf{L}]\,[\mathbf{U}]\,\mathbf{x} = \mathbf{b} \qquad \begin{bmatrix} 1 & 0 & 0 & 0 \\ \otimes & 1 & 0 & 0 \\ \otimes & \otimes & 1 & 0 \\ \otimes & \otimes & \otimes & 1 \end{bmatrix} \begin{bmatrix} \otimes & \otimes & \otimes & \otimes \\ 0 & \otimes & \otimes & \otimes \\ 0 & 0 & \otimes & \otimes \\ 0 & 0 & 0 & \otimes \end{bmatrix} \begin{Bmatrix} \times \\ \times \\ \times \\ \times \end{Bmatrix} = \begin{Bmatrix} \times \\ \times \\ \times \\ \times \end{Bmatrix}$$

Atliekamas pakeitimas $[\mathbf{U}]\,\{\mathbf{x}\} = \{\mathbf{y}\}$

$$[\mathbf{L}]\,\{\mathbf{y}\} = \{\mathbf{b}\} \qquad \begin{bmatrix} 1 & 0 & 0 & 0 \\ \otimes & 1 & 0 & 0 \\ \otimes & \otimes & 1 & 0 \\ \otimes & \otimes & \otimes & 1 \end{bmatrix} \begin{Bmatrix} \times \\ \times \\ \times \\ \times \end{Bmatrix} = \begin{Bmatrix} \times \\ \times \\ \times \\ \times \end{Bmatrix}$$

Pagal trikampę koeficientų matricą apskaičiuojamas **{y}**

$$[\mathbf{U}]\,\{\mathbf{x}\} = \{\mathbf{y}\} \qquad \begin{bmatrix} \otimes & \otimes & \otimes & \otimes \\ 0 & \otimes & \otimes & \otimes \\ 0 & 0 & \otimes & \otimes \\ 0 & 0 & 0 & \otimes \end{bmatrix} \begin{Bmatrix} \times \\ \times \\ \times \\ \times \end{Bmatrix} = \begin{Bmatrix} \times \\ \times \\ \times \\ \times \end{Bmatrix}$$

Pagal trikampę koeficientų matricą apskaičiuojamas **{x}**

$$\{\mathbf{x}\}$$

Analogiškai, kaip ir Gauso algoritmo atvirkštiniame etape

LU skaidos apskaičiavimas (1)

U apskaičiuojamas atliekant Gauso algoritmo tiesioginį etapą:

$$\begin{cases} x_1 + x_2 + x_3 + x_4 = 2 \\ x_1 - x_2 - x_3 + x_4 = 0 \\ 2x_1 + x_2 - x_3 + 2x_4 = 9 \\ 3x_1 + x_2 + 2x_3 - x_4 = 7 \end{cases}$$

$$\begin{bmatrix} 1^* & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 2 & 1 & -1 & 2 \\ 3 & 1 & 2 & -1 \end{bmatrix} \begin{array}{l} \downarrow \\ \leftarrow \times(-1) \\ \leftarrow \times(-2) \\ \leftarrow \times(-3) \end{array} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -2^* & -2 & 0 \\ 0 & -1 & -3 & 0 \\ 0 & -2 & -1 & -4 \end{bmatrix} \begin{array}{l} \\ \\ \leftarrow \times\left(-\tfrac{1}{2}\right) \\ \leftarrow \times(-1) \end{array} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -2 & -2 & 0 \\ 0 & 0 & -2^* & 0 \\ 0 & 0 & 1 & -4 \end{bmatrix} \begin{array}{l} \\ \\ \\ \leftarrow \times \tfrac{1}{2} \end{array} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -2 & -2 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & -4 \end{bmatrix}$$

$$\begin{bmatrix} 1^* & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 2 & 1 & -1 & 2 \\ 3 & 1 & 2 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -2^* & -2 & 0 \\ 0 & -1 & -3 & 0 \\ 0 & -2 & -1 & -4 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -2^* & -2 & 0 \\ 0 & -1 & -3 & 0 \\ 0 & -2 & -1 & -4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \tfrac{1}{2} & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -2 & -2 & 0 \\ 0 & 0 & -2^* & 0 \\ 0 & 0 & 1 & -4 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -2 & -2 & 0 \\ 0 & 0 & -2^* & 0 \\ 0 & 0 & 1 & -4 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\tfrac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -2 & -2 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & -4 \end{bmatrix}$$

LU skaidos apskaičiavimas (2)

$$\begin{bmatrix}1^* & 1 & 1 & 1\\ 1 & -1 & -1 & 1\\ 2 & 1 & -1 & 2\\ 3 & 1 & 2 & -1\end{bmatrix}\begin{matrix}\downarrow\\ \leftarrow\times(-1)\\ \leftarrow\times(-2)\\ \leftarrow\times(-3)\end{matrix}\Rightarrow\begin{bmatrix}1 & 1 & 1 & 1\\ 0 & -2^* & -2 & 0\\ 0 & -1 & -3 & 0\\ 0 & -2 & -1 & -4\end{bmatrix}\begin{matrix}\leftarrow\times\left(-\tfrac{1}{2}\right)\\ \leftarrow\times(-1)\end{matrix}\Rightarrow\begin{bmatrix}1 & 1 & 1 & 1\\ 0 & -2 & -2 & 0\\ 0 & 0 & -2^* & 0\\ 0 & 0 & 1 & -4\end{bmatrix}\leftarrow\times\tfrac{1}{2}\Rightarrow\begin{bmatrix}1 & 1 & 1 & 1\\ 0 & -2 & -2 & 0\\ 0 & 0 & -2 & 0\\ 0 & 0 & 0 & -4\end{bmatrix}$$

$$\begin{bmatrix}1^* & 1 & 1 & 1\\ 1 & -1 & -1 & 1\\ 2 & 1 & -1 & 2\\ 3 & 1 & 2 & -1\end{bmatrix}=\begin{bmatrix}1 & 0 & 0 & 0\\ 1 & 1 & 0 & 0\\ 2 & 0 & 1 & 0\\ 3 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 1 & 1 & 1\\ 0 & -2^* & -2 & 0\\ 0 & -1 & -3 & 0\\ 0 & -2 & -1 & -4\end{bmatrix}$$

$$\begin{bmatrix}1 & 1 & 1 & 1\\ 0 & -2^* & -2 & 0\\ 0 & -1 & -3 & 0\\ 0 & -2 & -1 & -4\end{bmatrix}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & \tfrac{1}{2} & 1 & 0\\ 0 & 1 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 1 & 1 & 1\\ 0 & -2 & -2 & 0\\ 0 & 0 & -2^* & 0\\ 0 & 0 & 1 & -4\end{bmatrix}$$

$$\begin{bmatrix}1 & 1 & 1 & 1\\ 0 & -2 & -2 & 0\\ 0 & 0 & -2^* & 0\\ 0 & 0 & 1 & -4\end{bmatrix}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & -\tfrac{1}{2} & 1\end{bmatrix}\begin{bmatrix}1 & 1 & 1 & 1\\ 0 & -2 & -2 & 0\\ 0 & 0 & -2 & 0\\ 0 & 0 & 0 & -4\end{bmatrix}$$

$$\begin{bmatrix}1^* & 1 & 1 & 1\\ 1 & -1 & -1 & 1\\ 2 & 1 & -1 & 2\\ 3 & 1 & 2 & -1\end{bmatrix}=\begin{bmatrix}1 & 0 & 0 & 0\\ 1 & 1 & 0 & 0\\ 2 & 0 & 1 & 0\\ 3 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & \tfrac{1}{2} & 1 & 0\\ 0 & 1 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & -\tfrac{1}{2} & 1\end{bmatrix}\begin{bmatrix}1 & 1 & 1 & 1\\ 0 & -2 & -2 & 0\\ 0 & 0 & -2 & 0\\ 0 & 0 & 0 & -4\end{bmatrix}$$

LU skaidos apskaičiavimas (3)

$$\begin{bmatrix} 1^* & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 2 & 1 & -1 & 2 \\ 3 & 1 & 2 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1/2 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -2 & -2 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & -4 \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} 1^* & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 2 & 1 & -1 & 2 \\ 3 & 1 & 2 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 2 & 1/2 & 1 & 0 \\ 3 & 1 & -1/2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -2 & -2 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & -4 \end{bmatrix}$$

$$[A] = [L] \cdot [U]$$

Pradinė koeficientų matrica

Koeficientai (su minuso ženklu), kurie buvo panaudoti atliekant Gauso algoritmo tiesioginį etapą

Gauso algoritmo tiesioginio etapo rezultatas

LU skaidos apskaičiavimas (4)

$$\begin{cases} x_1 + x_2 + x_3 + x_4 = 2 \\ x_1 - x_2 - x_3 + x_4 = 0 \\ 2x_1 + x_2 - x_3 + 2x_4 = 9 \\ 3x_1 + x_2 + 2x_3 - x_4 = 7 \end{cases}$$

$$[\mathbf{A}] = \begin{bmatrix} 1^* & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 2 & 1 & -1 & 2 \\ 3 & 1 & 2 & -1 \end{bmatrix} \begin{matrix} \downarrow \\ \leftarrow\times(-1) \\ \leftarrow\times(-2) \\ \leftarrow\times(-3) \end{matrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -2^* & -2 & 0 \\ 0 & -1 & -3 & 0 \\ 0 & -2 & -1 & -4 \end{bmatrix} \begin{matrix} \\ \\ \leftarrow\times\left(-\tfrac{1}{2}\right) \\ \leftarrow\times(-1) \end{matrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -2 & -2 & 0 \\ 0 & 0 & -2^* & 0 \\ 0 & 0 & 1 & -4 \end{bmatrix} \begin{matrix} \\ \\ \\ \leftarrow\times\tfrac{1}{2} \end{matrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -2 & -2 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & -4 \end{bmatrix} = [\mathbf{U}]$$

$$[\mathbf{L}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 2 & \tfrac{1}{2} & 1 & 0 \\ 3 & 1 & -\tfrac{1}{2} & 1 \end{bmatrix}$$

Įstrižainėje priimamos vienetinės reikšmės

LU skaidos apskaičiavimas (5)

Taupome kompiuterio atmintį:

$$\begin{bmatrix} 1^* & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 2 & 1 & -1 & 2 \\ 3 & 1 & 2 & -1 \end{bmatrix} \begin{matrix} \downarrow \\ \leftarrow\times(-1) \\ \leftarrow\times(-2) \\ \leftarrow\times(-3) \end{matrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -2^* & -2 & 0 \\ 2 & -1 & -3 & 0 \\ 3 & -2 & -1 & -4 \end{bmatrix} \begin{matrix} \\ \\ \leftarrow\times\left(-\tfrac{1}{2}\right) \\ \leftarrow\times(-1) \end{matrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -2 & -2 & 0 \\ 2 & \tfrac{1}{2} & -2^* & 0 \\ 3 & 1 & 1 & -4 \end{bmatrix} \begin{matrix} \\ \\ \\ \leftarrow\times\tfrac{1}{2} \end{matrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -2 & -2 & 0 \\ 2 & \tfrac{1}{2} & -2 & 0 \\ 3 & 1 & -\tfrac{1}{2} & -4 \end{bmatrix}$$

Atlikus algoritmą, daugikliai [L] ir [U] užima pradinės matricos [A] vietą:

$$[U] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -2 & -2 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & -4 \end{bmatrix}$$

$$[L] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 2 & \tfrac{1}{2} & 1 & 0 \\ 3 & 1 & -\tfrac{1}{2} & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -2 & -2 & 0 \\ 2 & \tfrac{1}{2} & -2 & 0 \\ 3 & 1 & -\tfrac{1}{2} & -4 \end{bmatrix} \Rightarrow [A]$$

Įstrižainės vienetus saugoti netikslinga. Ir taip žinome, kad [L] pagrindinėje įstrižainėje turi būti vienetai

LU algoritmo atvirkštinis etapas

$$[\mathbf{L}]\{\mathbf{y}\} = \{\mathbf{b}\} \qquad \begin{bmatrix} 1 & 0 & 0 & 0 \\ \times & 1 & 0 & 0 \\ \times & \times & 1 & 0 \\ \times & \times & \times & 1 \end{bmatrix} \begin{Bmatrix} \times \\ \times \\ \times \\ \times \end{Bmatrix} = \begin{Bmatrix} \times \\ \times \\ \times \\ \times \end{Bmatrix}$$

apskaičiuojamas y

$$[\mathbf{U}]\{\mathbf{x}\} = \{\mathbf{y}\} \qquad \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix} \begin{Bmatrix} \times \\ \times \\ \times \\ \times \end{Bmatrix} = \begin{Bmatrix} \times \\ \times \\ \times \\ \times \end{Bmatrix}$$

apskaičiuojamas x

$$\{\mathbf{x}\}$$

Kartą apskaičiavus daugiklius L ir U, juos galima naudoti pakartotinai, kai lygtis sprendžiama su kitu laisvųjų narių vektoriumi.

LU skaidos algoritmas

```
A=[1  1  1  1;
   1 -1 -1  1;
   2  1 -1  2;
   3  1  2 -1]
b=[2;0;9;7]
n=size(A,1)

% LU skaida
for i=1:n-1
    for j=i+1:n
        r=A(j,i)/A(i,i);
        A(j,i:n)=A(j,i+1:n)-A(i,i+1:n)*r;
        A(j,i)=r;
    end
end
% 1-as atvirkstinis etapas, sprendziama  Ly=b, y->b
for i=2:n
    b(i,:)=b(i,:)-A(i,1:i-1)*b(1:i-1);
end
% 2-as atvirkstinis etapas , sprendziama Ux=b, x->b
b(n)=b(n)/A(n,n);
for i=n-1:-1:1
    b(i)=(b(i)-A(i,i+1:n)*b(i+1:n))/A(i,i);
end
```

Choleckio skaida

- Kai lygčių sistemos matrica *A* yra simetrinė ir teigiamai apibrėžta, taikomas kitas, dvigubai spartesnis, skaidos metodas, vadinamas *Choleckio metodu;*

- Simetrinė matrica yra teigiamai apibrėžta, kai

$$\Delta_1 = |a_{11}| > 0 \quad \Delta_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{vmatrix} > 0 \quad \cdots \qquad \Delta_n = \det(A) > 0$$

Choleckio skaidos taikymas lygčių sistemai išspręsti

$$[\mathbf{A}]\{\mathbf{x}\} = \{\mathbf{b}\};$$

$$[\mathbf{L}]^T [\mathbf{L}]\{\mathbf{x}\} = \{\mathbf{b}\};$$

$$[\mathbf{L}]^T \{\mathbf{y}\} = \{\mathbf{b}\}; \qquad \Rightarrow \{\mathbf{y}\}$$

$$[\mathbf{L}]\{\mathbf{x}\} = \{\mathbf{y}\}; \qquad \Rightarrow \{\mathbf{x}\}$$

$$\begin{array}{ccccc} \mathbf{A} & = & [\mathbf{L}]^T & & [\mathbf{L}] \\ \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} & = & \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{12} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{1n} & l_{2n} & \cdots & l_{nn} \end{bmatrix} & & \begin{bmatrix} l_{11} & l_{12} & \cdots & l_{1n} \\ 0 & l_{22} & \cdots & l_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & l_{nn} \end{bmatrix} \end{array}$$

Choleckio skaida galėtų būti laikoma atskiru LU skaidos atveju:

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & 0 \\ 0 & -1 & 3 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -0.3333 & 1 & 0 \\ 0 & 0 & -0.3750 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 3 & -1 & 0 \\ 0 & 0 & 2.667 & -1 \\ 0 & 0 & 0 & 1.625 \end{bmatrix}$$

$$[\mathbf{A}] \quad = \quad [\mathbf{L}] \quad [\mathbf{U}]$$

$$\alpha = \mathrm{diag}(\mathrm{sqrt}(\mathrm{diag}(\mathrm{U})))$$

$$[\boldsymbol{\alpha}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1.7321 & 0 & 0 \\ 0 & 0 & 1.6330 & 0 \\ 0 & 0 & 0 & 1.2748 \end{bmatrix}$$

Choleckio skaida galėtų būti laikoma atskiru LU skaidos atveju:

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & 0 \\ 0 & -1 & 3 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -0.3333 & 1 & 0 \\ 0 & 0 & -0.3750 & 1 \end{bmatrix}[\alpha]\right)\left([\alpha]^{-1}\begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 3 & -1 & 0 \\ 0 & 0 & 2.667 & -1 \\ 0 & 0 & 0 & 1.625 \end{bmatrix}\right)$$

$$[A] = [L][\alpha] \quad [\alpha]^{-1}[U]$$

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & 0 \\ 0 & -1 & 3 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1.7321 & 0 & 0 \\ 0 & -0.5774 & 1.6330 & 0 \\ 0 & 0 & -0.6124 & 1.2748 \end{bmatrix}\begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1.7321 & -0.5774 & 0 \\ 0 & 0 & 1.6330 & -0.6124 \\ 0 & 0 & 0 & 1.2748 \end{bmatrix}$$

$$[A] = [L_C]^T \quad [L_C]$$

Choleckio(kvadratinės šaknies) skaidos algoritmas

Žinomi koeficientai

Ieškomi koeficientai

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{12} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{13} & a_{23} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & a_{3n} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{12} & l_{22} & 0 & \cdots & 0 \\ l_{13} & l_{23} & l_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{1n} & l_{2n} & l_{3n} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & l_{12} & l_{13} & \cdots & l_{1n} \\ 0 & l_{22} & l_{23} & \cdots & l_{2n} \\ 0 & 0 & l_{33} & \cdots & l_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & l_{nn} \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{12} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{13} & a_{23} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & a_{3n} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11}^{2} & l_{12}l_{11} & l_{13}l_{11} & \cdots & l_{1n}l_{11} \\ \times & \times & \times & \cdots & \times \\ \times & \times & \times & \cdots & \times \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \times & \times & \times & \cdots & \times \end{bmatrix}$$

$$l_{11} = \sqrt{a_{11}};$$

$$l_{1,2:n} = {a_{1,2:n}}/{l_{11}}$$

apskaičiuoti koeficientai

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{12} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{13} & a_{23} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & a_{3n} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{12} & l_{22} & 0 & \cdots & 0 \\ l_{13} & l_{23} & l_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{1n} & l_{2n} & l_{3n} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & l_{12} & l_{13} & \cdots & l_{1n} \\ 0 & l_{22} & l_{23} & \cdots & l_{2n} \\ 0 & 0 & l_{33} & \cdots & l_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & l_{nn} \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{12} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{13} & a_{23} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & a_{3n} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{12} & l_{22} & 0 & \cdots & 0 \\ l_{13} & l_{23} & l_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{1n} & l_{2n} & l_{3n} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & l_{12} & l_{13} & \cdots & l_{1n} \\ 0 & l_{22} & l_{23} & \cdots & l_{2n} \\ 0 & 0 & l_{33} & \cdots & l_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & l_{nn} \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{12} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{13} & a_{23} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & a_{3n} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} \times & \times & \times & \cdots & \times \\ \times & l_{12}^2 + l_{22}^2 & l_{12}l_{13} + l_{22}l_{23} & \cdots & l_{12}l_{1n} + l_{22}l_{2n} \\ \times & \times & \times & \cdots & \times \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \times & \times & \times & \cdots & \times \end{bmatrix}$$

$$l_{22} = \sqrt{a_{22} - l_{12}^2}; \quad l_{2,3:n} = \left(a_{2,3:n} - l_{12}l_{1,3:n}\right) \big/ l_{22}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{12} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{13} & a_{23} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & a_{3n} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{12} & l_{22} & 0 & \cdots & 0 \\ l_{13} & l_{23} & l_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{1n} & l_{2n} & l_{3n} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & l_{12} & l_{13} & \cdots & l_{1n} \\ 0 & l_{22} & l_{23} & \cdots & l_{2n} \\ 0 & 0 & l_{33} & \cdots & l_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & l_{nn} \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{12} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{13} & a_{23} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & a_{3n} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{12} & l_{22} & 0 & \cdots & 0 \\ l_{13} & l_{23} & l_{33} & \cdots & l_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{1n} & l_{2n} & l_{3n} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & l_{12} & l_{13} & \cdots & l_{1n} \\ 0 & l_{22} & l_{23} & \cdots & l_{2n} \\ 0 & 0 & l_{33} & \cdots & l_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & l_{nn} \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{12} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{13} & a_{23} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & a_{3n} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} \times & \times & \times & \cdots & \times \\ \times & \times & \times & \cdots & \times \\ \times & \times & l_{13}^{\ 2} + l_{23}^{\ 2} + l_{33}^{\ 2} & \cdots & l_{13}l_{1n} + l_{23}l_{2n} + l_{33}l_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \times & \times & \times & \cdots & \times \end{bmatrix}$$

$$l_{33} = \sqrt{a_{33} - l_{13}^{\ 2} - l_{23}^{\ 2}}; \quad l_{3,4:n} = \left(a_{3,4:n} - l_{13}l_{1,4:n} - l_{23}l_{2,4:n}\right) \Big/ l_{33}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{12} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{13} & a_{23} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & a_{3n} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{12} & l_{22} & 0 & \cdots & 0 \\ l_{13} & l_{23} & l_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{1n} & l_{2n} & l_{3n} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & l_{12} & l_{13} & \cdots & l_{1n} \\ 0 & l_{22} & l_{23} & \cdots & l_{2n} \\ 0 & 0 & l_{33} & \cdots & l_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & l_{nn} \end{bmatrix}$$

Choleckio L’*L skaidos algoritmas

```
A=[ 1 -1  0  0;
   -1  2 -1  0;
   0  -1  2 -1;
   0   0 -1  2]
b=[2;0;0;0]
n=size(A,1)

% Choleckio L'*L skaida. L saugoma virsutiniame A trikampyje.
for i=1:n
    if i>1, s=sum(A(1:i-1,i).^2); else, s=0; end
    A(i,i)=sqrt(A(i,i)-s);
    for j=i+1:n
        if i>1, s=A(1:i-1,i)’*A(1:i-1,j); else, s=0; end
        A(i,j)=(A(i,j)-s)/A(i,i);
    end
end
% 1-as atvirkstinis zingsnis, sprendziama  L'y=b, y->b
b(1)=b(1)/A(1,1);
for i=2:n
    b(i,:)=(b(i,:)-A(1:i-1,i)'*b(1:i-1))/A(i,i);
End
% 2-as atvirkstinis zingsnis   Lx=b,  x->b
b(n)=b(n)/A(n,n);
for i=n-1:-1:1
    b(i)=(b(i)-A(i,i+1:n)*b(i+1:n))/A(i,i);
end
```

MATLAB funkcijos algoritmams, paremtiems koeficientų matricos pertvarkymu

x=A\b   % “atvirkštinė dalyba”. Algoritmas paremtas LU skaida

[L,U]=lu(A)   % LU skaida

L = chol(A) % Choleckio skaida

[Q,R] = qr(A) % QR skaida

x = linsolve(A,B) % kombinuotas metodas, taiko LU skaida ir
                  % atspindzius

Y = inv(X) % atvirkstines matricos apskaiciavimas.
           % Neekonomiska naudoti x=inv(A)*b,   geriau x=A\b