

SOUSTAVY LINEÁRNÍCH ALGEBRAICKÝCH ROVNIC

Pojmy:

- *Algebraická rovnice* ... rovnice obsahující pouze celé nezáporné mocniny neznámé x , tj. $a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0 = 0$, kde n je přirozené číslo.
- *Lineární algebraická rovnice* ... rovnice obsahující pouze první mocninu neznámé, resp. neznámých, tj. $ax + b = 0$, resp. např. $a_1 x_1 + a_2 x_2 + a_3 x_3 = b$.
- *Soustava lineárních algebraických rovnic*

$$\begin{aligned}a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\&\vdots \\a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n\end{aligned}$$

Příklad: Řešte soustavu 3 rovnic pro 3 neznámé x , y a z :

$$\begin{aligned}3x + 2y + z &= 10 \\2x + 4y + 5z &= 25 \\3x + 4y + 8z &= 35\end{aligned}$$

Řešení: Snažíme se postupně eliminovat jednotlivé neznámé. Docílíme to tím způsobem, že budeme sčítat vhodně přenásobené rovnice.

1. krok Opíšeme 1. rovnici

$$3x + 2y + z = 10$$

K 2. rovnici přičteme určitý násobek 1. rovnice tak, aby se eliminovala neznámá x (vhodný koeficient je $-\frac{2}{3}$).

$$\begin{aligned}2x + 4y + 5z &= 25 \\-\frac{2}{3} \cdot (3x + 2y + z) &= -\frac{2}{3} \cdot 10\end{aligned}$$

$$\begin{aligned}4y - \frac{4}{3}y + 5z - \frac{2}{3}z &= 25 - \frac{20}{3} \\ \frac{12-4}{3}y + \frac{15-2}{3}z &= \frac{75-20}{3} \\ \frac{8}{3}y + \frac{13}{3}z &= \frac{55}{3}\end{aligned}$$

Podobně eliminujeme neznámou x z 3. rovnice, tj. k 3. rovnici přičteme -1 násobek první rovnice.

$$\begin{array}{r} 3x + 4y + 8z = 35 \\ -1 \cdot (3x + 2y + z) = -1 \cdot 10 \\ \hline 4y - 2y + 8z - z = 35 - 10 \\ 2y + 7z = 25 \end{array}$$

Obdrželi jsme soustavu

$$\begin{array}{r} 3x + 2y + z = 10 \\ \frac{8}{3}y + \frac{13}{3}z = \frac{55}{3} \\ 2y + 7z = 25 \end{array}$$

2. krok Nyní opíšeme první dvě rovnice a ze třetí budeme eliminovat neznámou y , tj. ke 3. rovnici přičteme $-\frac{2}{\frac{8}{3}}$ násobek 2. rovnice.

$$\begin{array}{r} 2y + 7z = 25 \\ -\frac{2}{\frac{8}{3}} \cdot \left(\frac{8}{3}y + \frac{13}{3}z \right) = -\frac{2}{\frac{8}{3}} \cdot \frac{55}{3} \\ \hline 7z - \frac{2}{\frac{8}{3}} \cdot \frac{13}{3}z = 25 - \frac{2 \cdot 55}{8} \\ 7z - \frac{13}{4}z = 25 - \frac{55}{4} \\ \frac{28 - 13}{4}z = \frac{100 - 55}{4} \\ \frac{15}{4}z = \frac{45}{4} \end{array}$$

Obdrželi jsme soustavu (tzv. **trojúhelníkový tvar**)

$$\begin{array}{r} 3x + 2y + z = 10 \\ \frac{8}{3}y + \frac{13}{3}z = \frac{55}{3} \\ \frac{15}{4}z = \frac{45}{4} \end{array}$$

3. krok z 3. rovnice vypočteme $z = 3$

4. krok Dosadíme z do 2. rovnice a vypočteme y

$$\begin{array}{rcl} \frac{8}{3}y + \frac{13}{3} \cdot 3 & = & \frac{55}{3} \\ \frac{8}{3}y & = & \frac{55 - 39}{3} \\ y & = & 2 \end{array}$$

5. krok Dosadíme y, z do 1. rovnice a vypočteme x

$$\begin{array}{rcl} 3x + 2 \cdot 2 + 3 & = & 10 \\ 3x & = & 10 - 7 \\ x & = & 1 \end{array}$$

□

Uvedený postup lze samozřejmě aplikovat na obecnou soustavu n rovnic pro n neznámých a nazývá se **Gaussova eliminační metoda (GEM)** ... soustavu nejprve převedeme na trojúhelníkový tvar a potom zpětným chodem vyjadřujeme řešení.

Pro větší přehlednost můžeme použít maticový zápis:

$$\underbrace{\begin{bmatrix} 3 & 2 & 1 \\ 2 & 4 & 5 \\ 3 & 4 & 8 \end{bmatrix}}_{\mathbf{A}} \cdot \underbrace{\begin{bmatrix} x \\ y \\ z \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} 10 \\ 25 \\ 35 \end{bmatrix}}_{\mathbf{b}}, \quad \text{tj.} \quad \boxed{\mathbf{Ax} = \mathbf{b}}$$

A ... matice koeficientů u neznámých v jednotlivých rovnicích

x ... vektor neznámých x, y a z

b ... vektor pravé strany

Poznámka: (Násobení matice · vektor) Výsledek násobení matice typu $m|n$ (m řádek a n sloupců) a vektoru typu $n|1$ (sloupcový vektor o n prvcích) je vektor typu $m|1$ (sloupcový vektor o m prvcích), kde i -tá složka výsledného vektoru je skalárním součinem i -tého řádku matice a daného vektoru.

Mluvíme o numerické matematice, tzn. řešíme dané problémy numericky (nikoli analyticky). Pro řešení problémů odvozujeme postupy (algoritmy), které budou naprogramovány do počítače. Naší snahou je, aby odvozené metody byly nejen rychlé, potřebovaly co nejméně paměti, ale aby byly i použitelné pro ty úlohy, které jsou řešitelné.

Příklad: Řešte soustavu rovnic

$$\begin{array}{rcl} x + 2y + 3z & = & 14 \\ 2x + 4y + 5z & = & 25 \\ 7x + 8y + 9z & = & 50 \end{array}, \quad \text{tj.} \quad \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 7 & 8 & 9 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 14 \\ 25 \\ 50 \end{bmatrix}$$

Řešení: Pro zápis budeme z důvodu přehlednosti používat tvar *matice rozšířené*:

$$\left(\begin{array}{ccc|c} \textcircled{1} & 2 & 3 & 14 \\ 2 & 4 & 5 & 25 \\ 7 & 8 & 9 & 50 \end{array} \right) \begin{array}{l} / \cdot \left(-\frac{2}{1}\right) \leftarrow \\ \leftarrow \end{array} + \begin{array}{l} / \cdot \left(-\frac{7}{1}\right) \leftarrow \\ \leftarrow \end{array} +$$

$$\left(\begin{array}{ccc|c} 1 & 2 & 3 & 14 \\ 0 & \textcircled{0} & -1 & -3 \\ 0 & -6 & -12 & -48 \end{array} \right) \begin{array}{l} / \cdot \left(-\frac{6}{0}\right) \leftarrow \\ \leftarrow \end{array} + \quad \text{!!! dělíme 0}$$

\implies Algoritmus **Gaussovy eliminační metody** není pro tento příklad realizovatelný. Snadno se přesvědčíme, že má daná soustava řešení $x = 1$, $y = 2$ a $z = 3$, ale Gaussova eliminační metoda selhala. \square

Otázky:

1. Pro jaké matice \mathbf{A} má soustava $\mathbf{Ax} = \mathbf{b}$ právě jedno řešení?

\rightarrow Matice \mathbf{A} musí být **regulární**, tj. všechna vlastní čísla musí být různá od nuly.

Poznámka: Vlastní číslo matice \mathbf{A} je číslo λ splňující rovnici $\mathbf{Av} = \lambda\mathbf{v}$, kde \mathbf{v} je vlastní vektor odpovídající vlastnímu číslu λ . Číslo λ tedy určitým způsobem charakterizuje matici \mathbf{A} .

2. Pro jaké matice \mathbf{A} je algoritmus Gaussovy eliminační metody realizovatelný?

\rightarrow *Věta:* Je-li matice \mathbf{A} **ostře diagonálně dominantní**, pak je algoritmus Gaussovy eliminační metody realizovatelný.

Poznámka: Matice $\mathbf{A} = [a_{ij}]_{i,j=1,\dots,n}$ je ostře diagonálně dominantní, platí-li

$|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|$, tj. absolutní hodnota diagonálního prvku je větší než součet absolutních hodnot ostatních prvků v řádku.

\rightarrow *Věta:* Je-li matice \mathbf{A} **symetrická a pozitivně definitní**, pak je algoritmus Gaussovy eliminační metody realizovatelný.

Poznámka: Matice \mathbf{A} je symetrická, platí-li pro její prvky $a_{ij} = a_{ji} \forall i, j = 1, 2, \dots, n$.

Poznámka: Matice \mathbf{A} je pozitivně definitní, má-li všechna vlastní čísla kladná.

Poznámka: Pro soustavu s maticí, která splňuje předpoklady některé z uvedených vět, je možné dopředu říci, že půjde řešit pomocí Gaussovy eliminační metody. Obráceně to ovšem neplatí, tj. není-li např. matice soustavy ostře diagonálně dominantní, ještě to obecně neznamená, že nepůjde pomocí Gaussovy eliminační metody řešit.

Např. je-li $\mathbf{A} = \begin{bmatrix} \textcircled{7} & 1 & 2 \\ 1 & \textcircled{4} & 1 \\ 2 & 4 & \textcircled{9} \end{bmatrix}$, tj. diagonální prvky jsou vždy větší než součet zbylých prvků v řádku, pak půjde soustava s touto maticí řešit pomocí Gaussovy eliminační metody.

Abychom zaručili, že soustava půjde vyřešit pro libovolnou regulární matici, musíme algoritmus Gaussovy eliminační metody upravit. Zavedeme tzv. **výběr hlavního prvku (pivotaci)**.

Poznámka: pivot (hlavní prvek) ... první nenulový prvek v daném řádku matice.

Příklad: Pomocí **GEM se sloupcovou pivotací** vyřešte soustavu rovnic z předcházejícího příkladu, kde selhala klasická GEM, tj. řešíme soustavu $\mathbf{Ax} = \mathbf{b}$, kde

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 7 & 8 & 9 \end{bmatrix} \quad \text{a} \quad \mathbf{b} = \begin{bmatrix} 14 \\ 25 \\ 50 \end{bmatrix}.$$

Řešení:

1. sloupec
↓

vyměň $\left[\begin{array}{ccc|c} 1 & 2 & 3 & 14 \\ 2 & 4 & 5 & 25 \\ \textcircled{7} & 8 & 9 & 50 \end{array} \right] \rightsquigarrow \left(\begin{array}{ccc|c} 7 & 8 & 9 & 50 \\ 2 & 4 & 5 & 25 \\ 1 & 2 & 3 & 14 \end{array} \right) \begin{array}{l} / \cdot (-\frac{2}{7}) \leftarrow + \\ / \cdot (-\frac{1}{7}) \leftarrow + \end{array}$

2. sloupec
↓

není třeba měnit $\rightarrow \left(\begin{array}{ccc|c} 7 & 8 & 9 & 50 \\ 0 & \frac{12}{7} & \frac{17}{7} & \frac{75}{7} \\ 0 & \frac{6}{7} & \frac{12}{7} & \frac{48}{7} \end{array} \right) / \cdot (-\frac{6}{\frac{12}{7}}) = -\frac{1}{2} \leftarrow +$

$$\left(\begin{array}{ccc|c} 7 & 8 & 9 & 50 \\ 0 & \frac{12}{7} & \frac{17}{7} & \frac{75}{7} \\ 0 & 0 & \frac{1}{2} & \frac{3}{2} \end{array} \right) \Rightarrow \mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

□

Poznámky:

- Při sloupcové pivotaci jsme postupně v každém sloupci (resp. jeho části pod diagonálou včetně) vybírali číslo, které bylo maximální v absolutní hodnotě a v případě, že toto číslo neleželo na diagonále, vyměnili jsme příslušné 2 rovnice. Dále jsme pokračovali jako v GEM bez pivotace, tj. nulovali jsme koeficienty pod diagonálou.
- Sloupcová pivotace není jediná možnost. Podobně můžeme vybírat i maximální prvek v absolutní hodnotě z příslušného řádku (resp. jeho části) a poté vyměnit příslušné sloupce. Pozor! Je ovšem třeba zaměnit i příslušné složky řešení \mathbf{x} . V tomto případě hovoříme o **řádkové pivotaci**.
- Další možností je vybírat maximální prvek v absolutní hodnotě z celé matice \mathbf{A} (resp. příslušné podmatice). V tomto případě hovoříme o **úplné pivotaci**. Opět je třeba mít na paměti, že je třeba zaměnit složky ve vektoru řešení. Nevýhodou úplné pivotace je pomalejší výpočet neboť hlavní prvek vyhledáváme z celé dosud neupravené části.

Gaussova eliminační metoda není jedinou metodou pro řešení soustav lineárních algebraických rovnic. Další metodou je např. **metoda LU-rozkladu**.

Opět uvažujeme regulární matici \mathbf{A} řádu n . Matici \mathbf{A} lze rozložit na součin $\mathbf{A} = \mathbf{L} \cdot \mathbf{U}$, kde \mathbf{L} je dolní trojúhelníková matice řádu n a \mathbf{U} je horní trojúhelníková matice řádu n .

Poznámka: Aby byl rozklad jednoznačný, předpokládáme na diagonále matice \mathbf{L} jedničky.

Příklad LU-rozkladu matice \mathbf{A}

$$\underbrace{\begin{bmatrix} 1 & 2 & 3 \\ 2 & 8 & 11 \\ 3 & 22 & 35 \end{bmatrix}}_{\mathbf{A}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 4 & 1 \end{bmatrix}}_{\mathbf{L}} \cdot \underbrace{\begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{bmatrix}}_{\mathbf{U}}$$

□

Řešení soustavy $\mathbf{Ax} = \mathbf{b}$ metodou LU-rozkladu:

$$\left. \begin{array}{l} 1) \text{ Realizace LU-rozkladu} \\ 2) \text{ Řešení soustavy (zpětný chod)} \\ 3) \text{ Řešení soustavy (zpětný chod)} \end{array} \right\} \begin{array}{l} \mathbf{A} = \mathbf{L} \cdot \mathbf{U} \\ \mathbf{Ly} = \mathbf{b} \\ \mathbf{Ux} = \mathbf{y} \end{array} \quad \mathbf{Ax} = \mathbf{L} \underbrace{\mathbf{Ux}}_{\mathbf{y}} = \mathbf{b}$$

Příklad: Řešte soustavu $\mathbf{Ax} = \mathbf{b}$ metodou LU-rozkladu, kde

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 8 & 11 \\ 3 & 22 & 35 \end{bmatrix} \quad \text{a} \quad \mathbf{b} = \begin{bmatrix} 13 \\ 45 \\ 133 \end{bmatrix}.$$

Řešení:

1. Z předchozího textu známe LU-rozklad matice \mathbf{A} .

2. Řešíme soustavu $\mathbf{L}\mathbf{y} = \mathbf{b}$

$$\left(\begin{array}{ccc|c} 1 & 0 & 0 & 13 \\ 2 & 1 & 0 & 45 \\ 3 & 4 & 1 & 133 \end{array} \right) \Rightarrow \mathbf{y} = \begin{bmatrix} 13 \\ 19 \\ 18 \end{bmatrix}.$$

3. Řešíme soustavu $\mathbf{U}\mathbf{x} = \mathbf{y}$

$$\left(\begin{array}{ccc|c} 1 & 2 & 3 & 13 \\ 0 & 4 & 5 & 19 \\ 0 & 0 & 6 & 18 \end{array} \right) \Rightarrow \mathbf{x} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}.$$

□

Poznámka: Metoda LU-rozkladu je vhodná v případech, kdy řešíme více soustav se stejnou maticí \mathbf{A} , ale různými pravými stranami \mathbf{b} . (Důvod je ten, že jsme investovali práci do samotného LU-rozkladu. Vyřešení dvou soustav s trojúhelníkovou maticí je pouze věc dosazování.)

Poznámka: I v algoritmu LU-rozkladu lze využít pivotaci podobně jako v GEM.

Dosud jsme hovořili o tzv. **přímých metodách** (GEM, metoda LU-rozkladu), tzn. o metodách, které naleznou přesné řešení po konečném počtu kroků. Další skupinou metod pro řešení soustav rovnic jsou tzv. **iterační metody**, které *teoreticky* najdou přesné řešení až po nekonečně mnoha krocích.

Pamatujme si, že v numerické praxi používáme pro řešení soustav s plnou maticí *přímé metody*, zatímco pro speciální (řídké) matice používáme *iterační metody*.

Toto rozdělení je dáno **výpočetní složitostí** těchto metod, tj. počtem matematických operací sčítání, odčítání, násobení a dělení nutných k získání výsledku.

Poznámka: V případě plné matice je výpočetní cena v každé iteraci řádu n^2 , srovnáme-li toto s celkovou výpočetní cenou přímých metod, tj. řádově $\frac{2}{3}n^3$, vidíme, že má-li být výpočetní složitost iterační metody stejná jako u přímé metody, musela by iterační metoda najít řešení (s předem zadanou přesností) řádově po n iteracích. Na druhou stranu v případě speciální (řídké) matice je výhodné použít iterační metodu.

V následujícím příkladu si ukážeme princip iteračních metod pro řešení soustav lineárních algebraických rovnic.

Příklad: Pomocí iterační metody řešte soustavu

$$\begin{array}{r} 11x + 2y + z = 15 \\ x + 10y + 2z = 16 \\ 2x + 3y - 8z = 1 \end{array}$$

Princip iteračních metod je založen na iterační formuli, tj. předpisu, kde se neznámé (resp. jedna z nich) vyskytují na obou stranách. Nejjednodušším způsobem jak získat iterační formuli je z i -té rovnice vyjádřit i -tou složku řešení, tzn. v našem případě z první rovnice vyjádříme x , z druhé rovnice vyjádříme y a ze třetí rovnice vyjádříme z . Dostaneme

$$\begin{array}{r} x = \frac{1}{11} \cdot (15 - 2y - z) \\ y = \frac{1}{10} \cdot (16 - x - 2z) \\ z = -\frac{1}{8} \cdot (1 - 2x - 3y) \end{array}$$

Tyto iterační formule můžeme zapsat jednodušším způsobem pomocí matice a vektorů

$$\underbrace{\begin{bmatrix} x \\ y \\ z \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} 0 & -\frac{2}{11} & -\frac{1}{11} \\ -\frac{1}{10} & 0 & -\frac{1}{5} \\ \frac{1}{4} & \frac{3}{8} & 0 \end{bmatrix}}_{\mathbf{H}} \cdot \underbrace{\begin{bmatrix} x \\ y \\ z \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} \frac{15}{11} \\ \frac{8}{5} \\ -\frac{1}{8} \end{bmatrix}}_{\mathbf{g}} \quad \text{tj.} \quad \mathbf{x} = \mathbf{H} \cdot \mathbf{x} + \mathbf{g}$$

Abychom mohli podle této formule počítat, je třeba zvolit počáteční vektor $\mathbf{x}^{(0)}$, který dosadíme na pravou stranu formule. Získáme novou iteraci $\mathbf{x}^{(1)}$, tu opět dosadíme na pravou stranu formule a získáme $\mathbf{x}^{(2)}$, atd. Tento postup je pro zadané $\mathbf{x}^{(0)}$ popsán rekurentní formulí

$$\mathbf{x}^{(k+1)} = \mathbf{H} \cdot \mathbf{x}^{(k)} + \mathbf{g}.$$

Za vektor $\mathbf{x}^{(0)}$ můžeme volit libovolný vektor, nejjednodušší možností je tedy volit nulový vektor. Konkrétně v našem případě získáme následující iterace (pro jednoduchost zapisujeme na 4 desetinná místa)

$$\begin{array}{l} \mathbf{x}^{(0)} = [0.0000, 0.0000, 0.0000]^T \\ \mathbf{x}^{(1)} = [1.3636, 1.6000, -0.1250]^T \\ \mathbf{x}^{(2)} = [1.0841, 1.4886, 0.8159]^T \\ \mathbf{x}^{(3)} = [1.0188, 1.3284, 0.7043]^T \\ \mathbf{x}^{(4)} = [1.0581, 1.3573, 0.6279]^T \\ \mathbf{x}^{(5)} = [1.0598, 1.3686, 0.6485]^T \end{array}$$

$$\begin{aligned}
\mathbf{x}^{(6)} &= [1.0558, 1.3643, 0.6532]^T \\
\mathbf{x}^{(7)} &= [1.0562, 1.3638, 0.6506]^T \\
\mathbf{x}^{(8)} &= [1.0565, 1.3643, 0.6505]^T \\
\mathbf{x}^{(9)} &= [1.0565, 1.3643, 0.6507]^T \\
\mathbf{x}^{(10)} &= [1.0564, 1.3642, 0.6507]^T \\
\mathbf{x}^{(11)} &= [1.0564, 1.3642, 0.6507]^T
\end{aligned}$$

Máme-li iterační předpis a první iteraci, můžeme začít počítat. Samozřejmě je třeba se zamyslet nad podmínkou pro zastavení výpočtu. Pro ukončení výpočtu budeme používat podmínku pro rozdíl dvou po sobě jdoucích iterací, kde tento rozdíl (resp. jeho norma) musí být menší než předem zadaná tolerance. V našem příkladu je z posloupnosti jednotlivých iterací zřejmé, že se 10-tá a 11-tá iterace shodují na 4 desetinná místa ve všech složkách, proto jsme výpočet ukončili a řekneme, že řešení soustavy $\mathbf{x} \approx \mathbf{x}^{(11)}$.

□

Metoda, kterou jsme odvodili v předchozím příkladu, se nazývá **Jacobiova metoda**.

Poznámka:

Uvědomme si souvislost s řešením obecné soustavy nelineárních rovnic $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ pomocí metody prosté iterace. Rovnici jsme si vektorově přepsali na tvar $\mathbf{x} = \Phi(\mathbf{x})$. Uvažujeme-li soustavu lineárních algebraických rovnic, tj. funkce \mathbf{F} je lineární

$$\underbrace{\mathbf{Ax} - \mathbf{b}}_{\mathbf{F}(\mathbf{x})} = \mathbf{0}$$

můžeme potom najít lineární předpis pro funkci Φ

$$\mathbf{x} = \underbrace{\mathbf{Hx} + \mathbf{g}}_{\Phi(\mathbf{x})}$$

Všechny iterační metody pro řešení soustavy lineárních algebraických rovnic budou používat iterační formuli ve tvaru

$$\mathbf{x}^{(k+1)} = \mathbf{H} \cdot \mathbf{x}^{(k)} + \mathbf{g},$$

samozřejmě s různou iterační maticí \mathbf{H} a vektorem \mathbf{g} a je zřejmé, že o kvalitě metody budou rozhodovat právě vlastnosti matice \mathbf{H} .

Nyní si stručně zformulujeme principy některých iteračních metod.

Jacobiova metoda

Princip: Z i -té rovnice vyjádříme i -tou složku vektoru \mathbf{x}

$$i\text{-tá rovnice: } a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = b_i$$

$$\text{pro } a_{ii} \neq 0: \quad x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1, j \neq i}^n a_{ij}x_j \right)$$

Iterační formule:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1, j \neq i}^n a_{ij}x_j^{(k)} \right)$$

Gaussova-Seidelova metoda

Princip: Stejný jako u Jacobiovy metody s tím rozdílem, že jestliže při výpočtu $(k+1)$ -iterace již známe $(k+1)$ -iteraci některých složek, tak ji použijeme.

Iterační formule:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right)$$

Relaxační metoda SOR

Princip: Vyjdeme z Gaussovy-Seidelovy metody jejíž iterační formulu lze psát takto:

$$x_i^{(k+1)} = x_i^{(k)} + r_i^{(k)} \quad \text{kde } r_i^{(k)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^n a_{ij}x_j^{(k)} \right)$$

Abychom urychlili výpočet, nebudeme přičítat $r_i^{(k)}$, ale $\omega r_i^{(k)}$, tj. $x_i^{(k+1)} = x_i^{(k)} + \omega r_i^{(k)}$

Iterační formule:

$$x_i^{(k+1)} = x_i^{(k)} + \omega \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^n a_{ij}x_j^{(k)} \right)$$

Poznámka: $(k+1)$ -iterace metody SOR je lineární kombinací $(k+1)$ -iterace získané Gauss-Seidelovou metodou a předchozí k -té iterace.

$$x_i^{(k+1)} = \omega \underbrace{\frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right)}_{x_i^{(k+1)} \text{ z Gaussovy-Seidelovy metody}} + (1 - \omega)x_i^{(k)}$$

Uvedené iterační formule lze přepsat do maticové podoby. Nejprve rozložíme matici \mathbf{A} :

$$\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U},$$

kde \mathbf{L} je dolní trojúhelníková část matice \mathbf{A} s nulami na diagonále, \mathbf{D} je diagonální matice a \mathbf{U} je horní trojúhelníková část matice \mathbf{A} s nulami na diagonále.

Jacobiova metoda:

$$\begin{aligned} \mathbf{Ax} &= \mathbf{b} \\ (\mathbf{L} + \mathbf{D} + \mathbf{U})\mathbf{x} &= \mathbf{b} \\ \mathbf{Dx} + (\mathbf{L} + \mathbf{U})\mathbf{x} &= \mathbf{b} \\ \mathbf{Dx} &= \mathbf{b} - (\mathbf{L} + \mathbf{U})\mathbf{x} \\ \hline \mathbf{x} &= \underbrace{-\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{x}}_{\mathbf{H}_J} + \underbrace{\mathbf{D}^{-1}\mathbf{b}}_{\mathbf{g}_J} \end{aligned}$$

Gauss-Seidlova metoda:

$$\begin{aligned} \mathbf{Ax} &= \mathbf{b} \\ (\mathbf{L} + \mathbf{D} + \mathbf{U})\mathbf{x} &= \mathbf{b} \\ (\mathbf{L} + \mathbf{D})\mathbf{x} + \mathbf{Ux} &= \mathbf{b} \\ (\mathbf{L} + \mathbf{D})\mathbf{x} &= \mathbf{b} - \mathbf{Ux} \\ \hline \mathbf{x} &= \underbrace{-(\mathbf{L} + \mathbf{D})^{-1}\mathbf{U}\mathbf{x}}_{\mathbf{H}_{GS}} + \underbrace{(\mathbf{L} + \mathbf{D})^{-1}\mathbf{b}}_{\mathbf{g}_{GS}} \end{aligned}$$

Relaxační metoda SOR:

$$\begin{aligned} \mathbf{Ax} &= \mathbf{b} \\ \omega\mathbf{Ax} &= \omega\mathbf{b} && / + \mathbf{Dx} \\ (\omega\mathbf{A} + \mathbf{D})\mathbf{x} &= \omega\mathbf{b} + \mathbf{Dx} \\ [\omega(\mathbf{L} + \mathbf{D} + \mathbf{U}) + \mathbf{D}]\mathbf{x} &= \omega\mathbf{b} + \mathbf{Dx} \\ (\omega\mathbf{L} + \mathbf{D})\mathbf{x} &= \omega\mathbf{b} + \mathbf{Dx} - \omega\mathbf{Dx} - \omega\mathbf{Ux} \\ (\omega\mathbf{L} + \mathbf{D})\mathbf{x} &= [(1 - \omega)\mathbf{D} - \omega\mathbf{U}]\mathbf{x} + \omega\mathbf{b} \\ \hline \mathbf{x} &= \underbrace{(\omega\mathbf{L} + \mathbf{D})^{-1}[(1 - \omega)\mathbf{D} - \omega\mathbf{U}]\mathbf{x}}_{\mathbf{H}_{SOR}} + \underbrace{(\omega\mathbf{L} + \mathbf{D})^{-1}\omega\mathbf{b}}_{\mathbf{g}_{SOR}} \end{aligned}$$

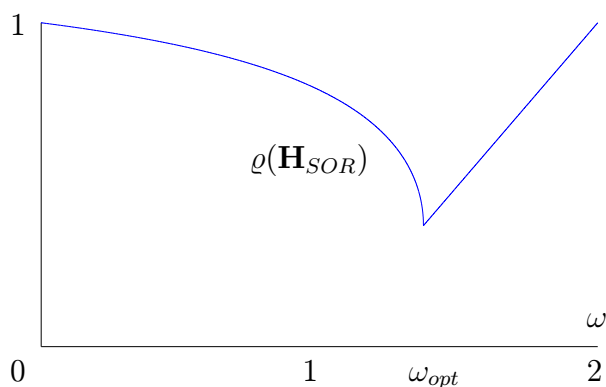
Poznámka:

Parametr ω v relaxační metodě SOR volíme z intervalu $(0, 2)$. Pro $\omega = 1$ přejde relaxační metoda na Gauss-Seidlovu metodu. Volba parametru ω samozřejmě ovlivní rychlost konvergence iteračního procesu metody SOR. Lze ukázat, že existuje optimální hodnota parametru omega

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \varrho^2}}, \quad \text{kde } \varrho \text{ je spektrální poloměr Jacobiovy iterační matice } \mathbf{H}_J.$$

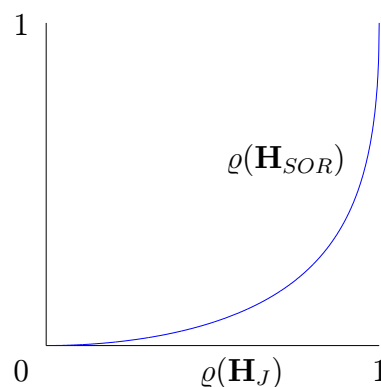
(spektrální poloměr = maximální vlastní číslo v absolutní hodnotě)

Pro spektrální poloměr iterační matice \mathbf{H}_{SOR} relaxační metody lze odvodit následující závislosti:



Obr. 1

Závislost spektrálního poloměru iterační matice metody SOR na relaxačním parametru ω



Obr. 2

Závislost spektrálního poloměru matice metody SOR na spektrálním poloměru iterační matice Jacobiovy metody

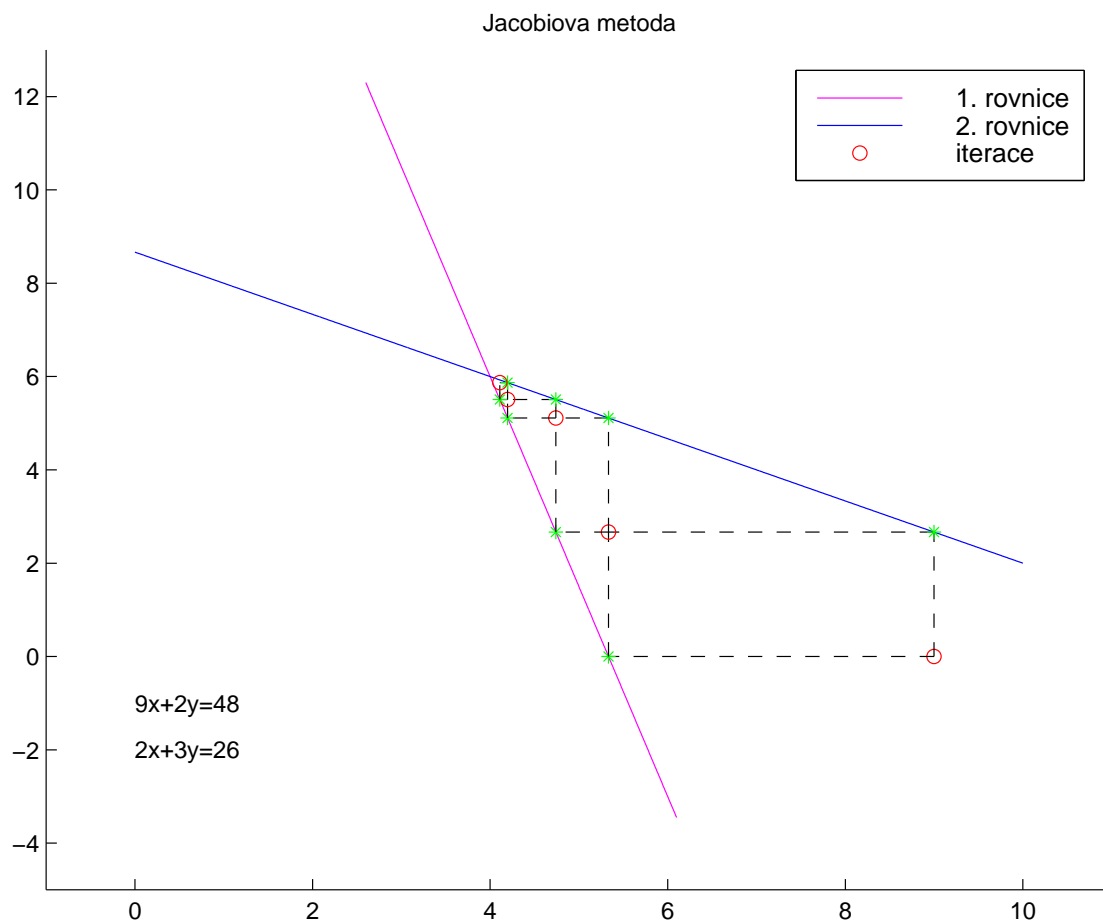
Věta: Konzistentní iterační metoda daná předpisem $\mathbf{x}^{(k+1)} = \mathbf{H}\mathbf{x}^{(k)} + \mathbf{g}$ konverguje pro libovolnou počáteční iteraci právě tehdy, když $\varrho(\mathbf{H}) < 1$.

Poznámka: Konzistentní metoda splňuje podmínku $\mathbf{A}^{-1}\mathbf{b} = \mathbf{H}\mathbf{A}^{-1}\mathbf{b} + \mathbf{g}$.

Poznámka: Čím je menší spektrální poloměr iterační matice \mathbf{H} , tím rychleji metoda konverguje.

Pro soustavu 2 rovnic pro 2 neznámé si lze princip uvedených iteračních metod znázornit geometricky.

GEOMETRICKÝ VÝZNAM JACOBOVY METODY

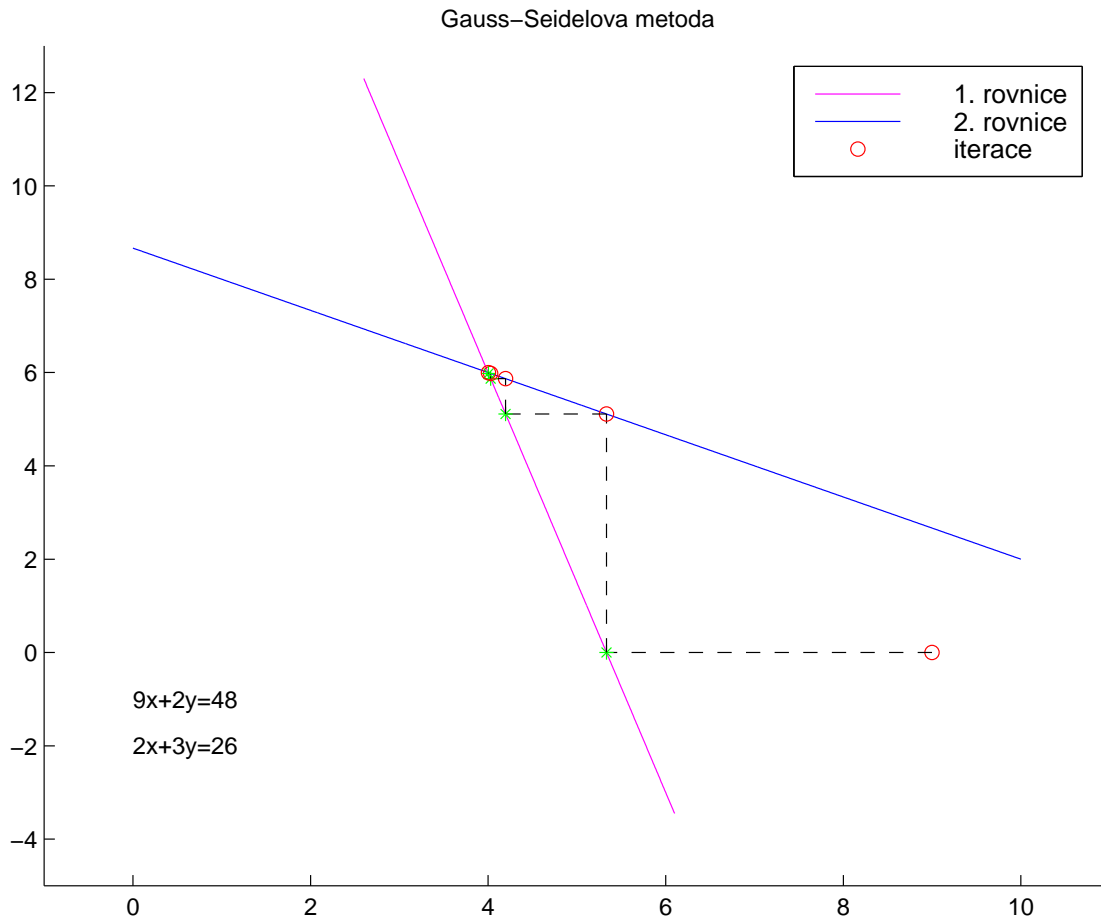


$$x^{(k+1)} = \frac{1}{9}(48 - 2y^{(k)})$$

$$y^{(k+1)} = \frac{1}{3}(26 - 2x^{(k)})$$

k	$x^{(k)}$	$y^{(k)}$
0	9.0000	0
1	5.3333	2.6667
2	4.7407	5.1111
3	4.1975	5.5062
4	4.1097	5.8683
5	4.0293	5.9268

GEOMETRICKÝ VÝZNAM GAUSSOVY-SEIDELOVY METODY

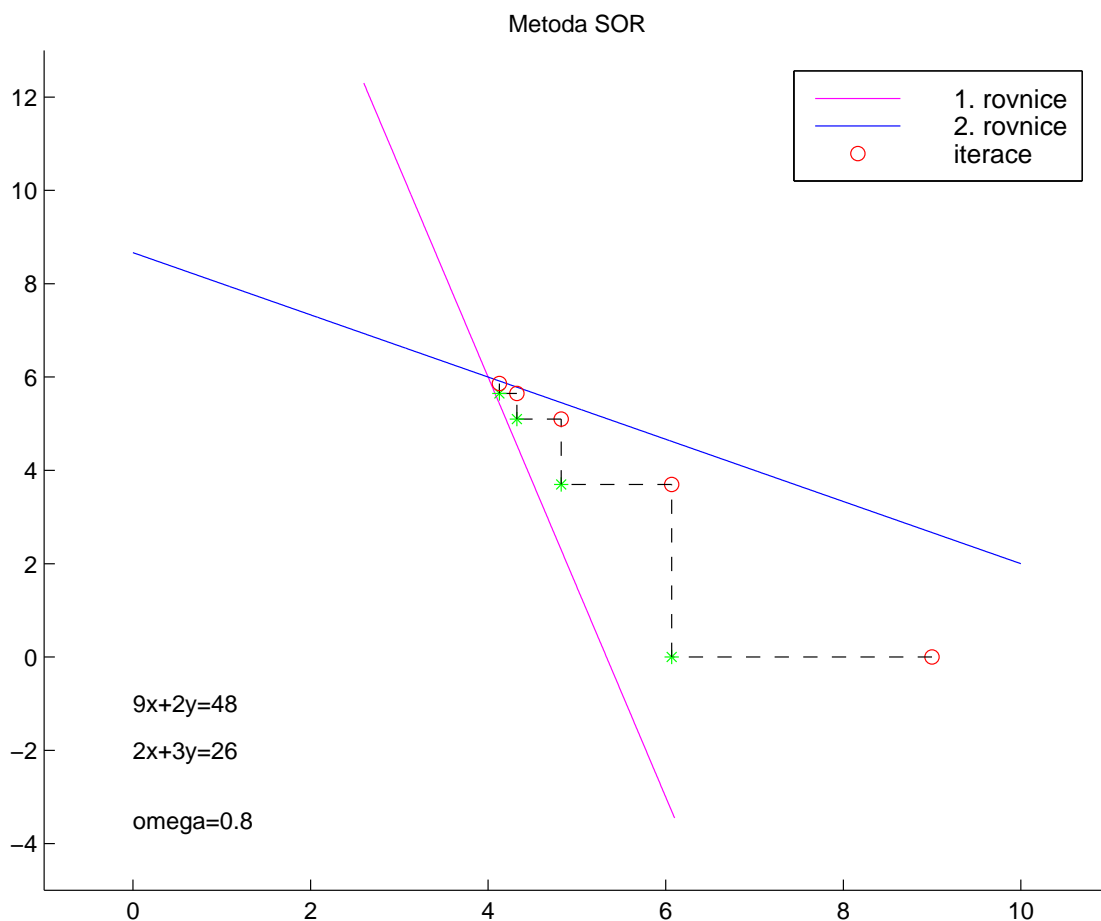


$$x^{(k+1)} = \frac{1}{9}(48 - 2y^{(k)})$$

$$y^{(k+1)} = \frac{1}{3}(26 - 2x^{(k+1)})$$

k	$x^{(k)}$	$y^{(k)}$
0	9.0000	0
1	5.3333	5.1111
2	4.1975	5.8683
3	4.0293	5.9805
4	4.0043	5.9971
5	4.0006	5.9996

GEOMETRICKÝ VÝZNAM METODY SOR ($\omega = 0.8$)

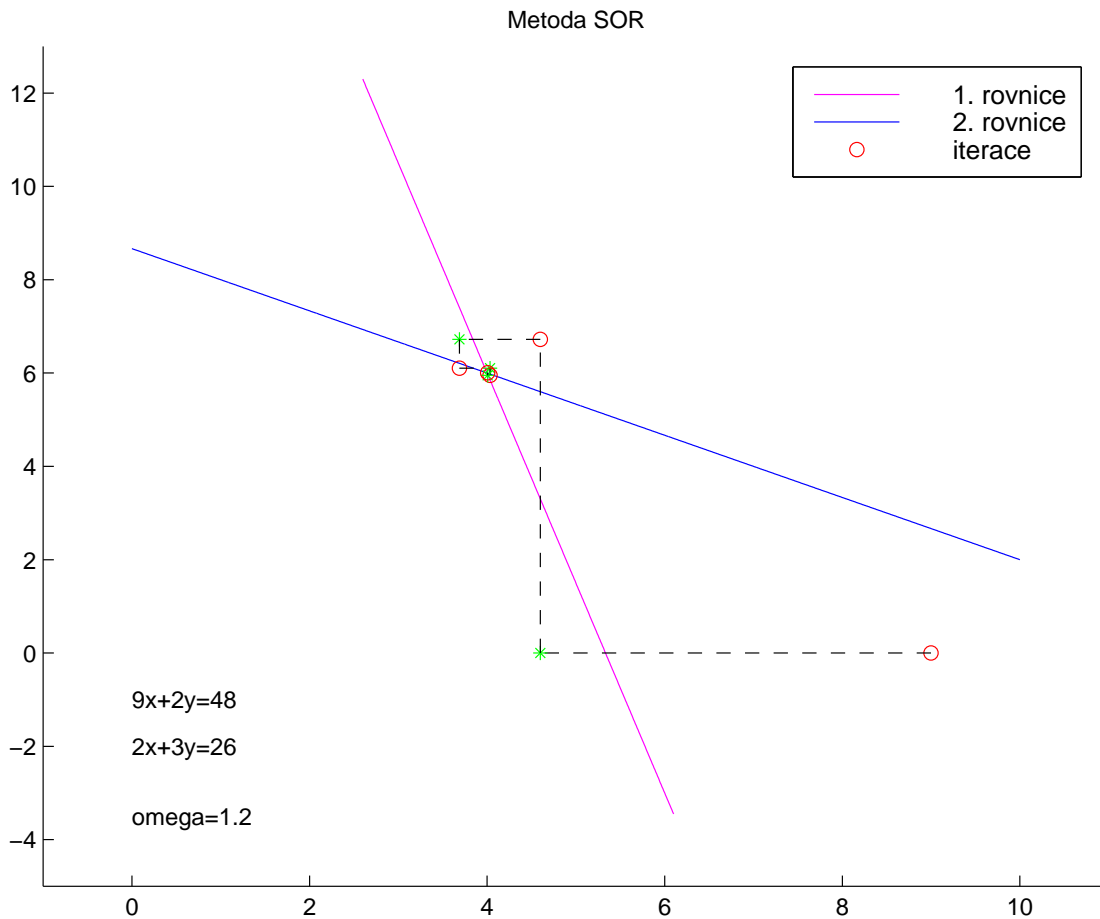


$$x^{(k+1)} = \omega \frac{1}{9}(48 - 2y^{(k)}) + (1 - \omega)x^{(k)}$$

$$y^{(k+1)} = \omega \frac{1}{3}(26 - 2x^{(k+1)}) + (1 - \omega)y^{(k)}$$

k	$x^{(k)}$	$y^{(k)}$
0	9.0000	0
1	6.0667	3.6978
2	4.8226	5.1008
3	4.3244	5.6472
4	4.1276	5.8614
5	4.0502	5.9455

GEOMETRICKÝ VÝZNAM METODY SOR ($\omega = 1.2$)



$$x^{(k+1)} = \omega \frac{1}{9}(48 - 2y^{(k)}) + (1 - \omega)x^{(k)}$$

$$y^{(k+1)} = \omega \frac{1}{3}(26 - 2x^{(k+1)}) + (1 - \omega)y^{(k)}$$

k	$x^{(k)}$	$y^{(k)}$
0	9.0000	0
1	4.6000	6.7200
2	3.6880	6.1056
3	4.0342	5.9515
4	4.0061	6.0048
5	3.9975	6.0010

Uvedme nyní speciální typ iteračních metod pro řešení soustav lineárních algebraických rovnic, tzv. **gradientní metody**. Jako motivace nám může posloužit příklad pro funkci jedné reálné proměnné. Uvažujme kvadratickou funkci

$$f(x) = \frac{1}{2}ax^2 - bx + c, \quad a > 0.$$

Nutná a postačující podmínka minima má tvar

$$ax = b.$$

To znamená, že místo řešení rovnice můžeme řešit úlohu najít minimum konvexní kvadratické funkce $f(x)$ (obě úlohy mají stejné řešení). Uvědomme si, že v případě funkce více proměnných je třeba splnit další podmínky kladené na matici soustavy \mathbf{A} , abychom zaručili konvexnost příslušné kvadratické funkce.

Uvažujme soustavu

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

kde matice \mathbf{A} je symetrická, pozitivně definitní.

Dále uvažujeme kvadratickou formu, tzv. **energetický funkcionál**

$$\mathbf{F}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{b}^T \mathbf{x}.$$

Platí

$$\text{grad } \mathbf{F}(\mathbf{x}) = \mathbf{A} \mathbf{x} - \mathbf{b}.$$

Funkce $\mathbf{F}(\mathbf{x})$ je konvexní a kvadratická $\Rightarrow \mathbf{F}(\mathbf{x})$ má globální minimum. Pro bod minima $\tilde{\mathbf{x}}$ platí

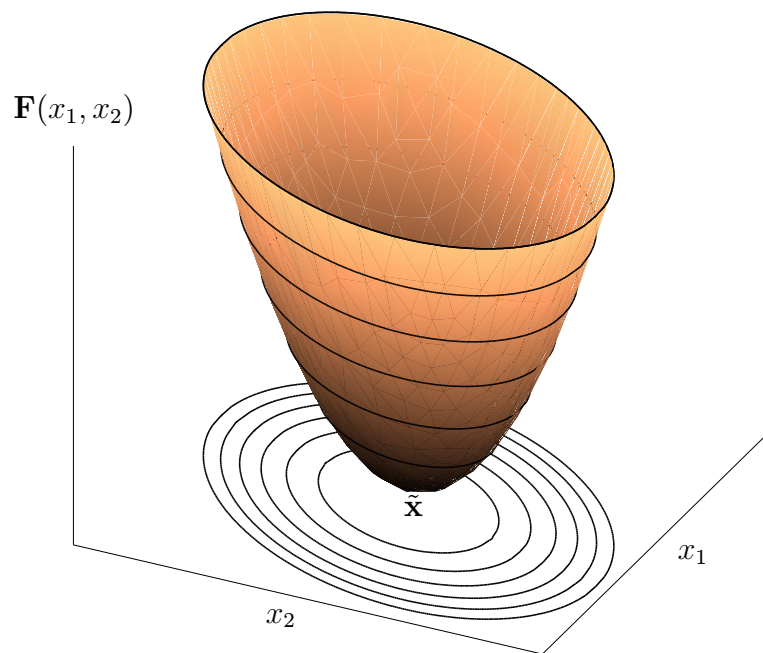
$$\text{grad } \mathbf{F}(\tilde{\mathbf{x}}) = \mathbf{A}\tilde{\mathbf{x}} - \mathbf{b} = \mathbf{0}.$$

Bod minima $\tilde{\mathbf{x}}$ je tedy řešením soustavy $\mathbf{A}\mathbf{x} = \mathbf{b}$.

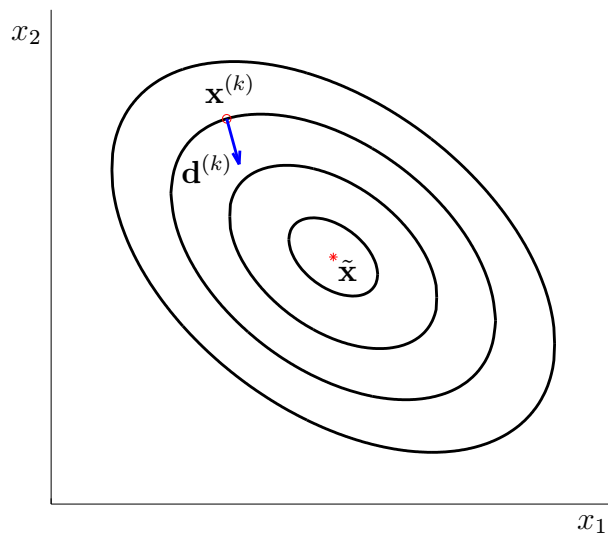
Poznámka:

Úlohy najít bod minima funkce \mathbf{F} a řešit soustavu $\mathbf{A}\mathbf{x} = \mathbf{b}$ jsou ekvivalentní.

Poznámka: V případě soustavy 2 rovnic si lze udělat geometrickou představu, neboť pro $\mathbf{x} \in \mathbb{R}^2$ je grafem funkce $\mathbf{F}(\mathbf{x})$ eliptický paraboloid, jehož vrstevnice jsou elipsy. Minima $\mathbf{F}(\mathbf{x})$ se nabývá ve vrcholu paraboloidu.



Stejně jako u každé iterační metody nejprve zvolíme počáteční aproximaci řešení $\mathbf{x}^{(0)}$. Princip gradientních metod spočívá v tom, že zvolíme směr a v tomto směru se budeme chtít co nejvíce přiblížit k přesnému řešení. Gradientní metoda je tedy dána volbou směrů, ve kterých minimalizujeme funkci \mathbf{F} .



V případě soustavy dvou rovnic získáme promítnutím grafu funkce $\mathbf{F}(\mathbf{x})$ do roviny proměnných x_1, x_2 systém soustředných elips - hladin (vrstevnic).

První možností je za směrový vektor volit směr největšího spádu, tj. vektor

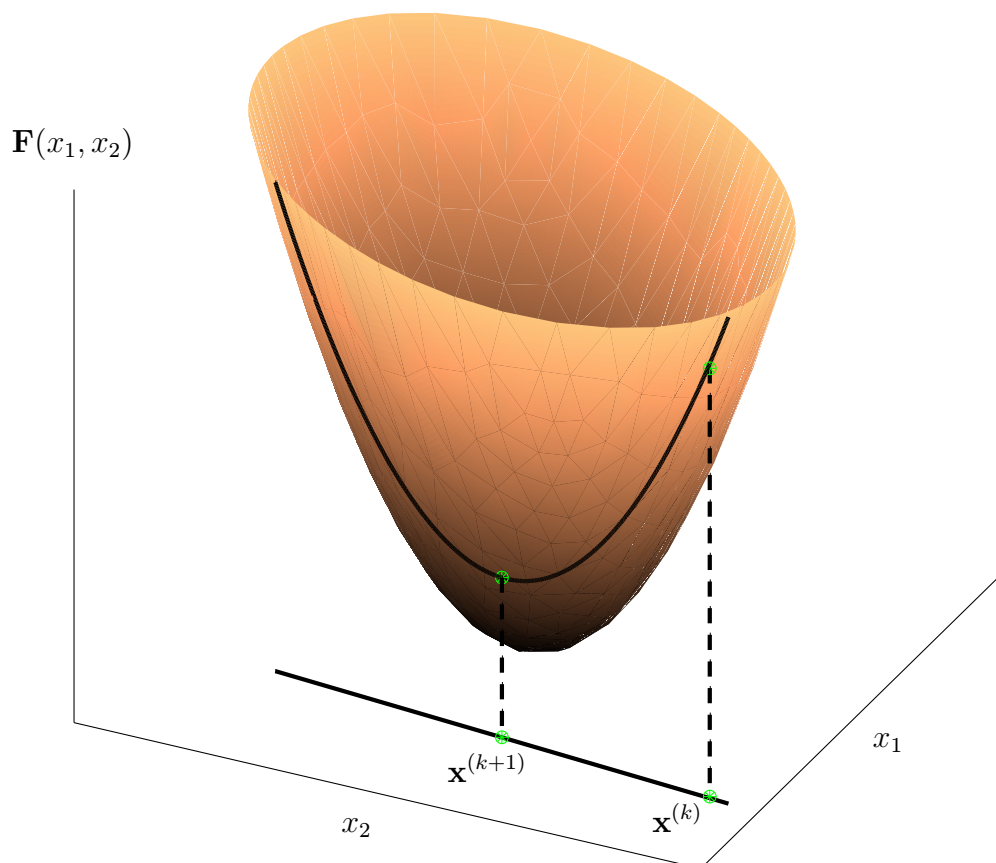
$$\mathbf{d}^{(k)} = -\text{grad } \mathbf{F}(\mathbf{x}^{(k)}) = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}.$$

Získáme tzv. **metodu největšího spádu**. Iterační formuli volíme ve tvaru

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)} \cdot \mathbf{d}^{(k)},$$

v každém kroku metody určíme směr největšího spádu $\mathbf{d}^{(k)}$ a provedeme jednorozměrnou minimalizaci v tomto směru, tj.

$$\min_{t>0} \mathbf{F}(\mathbf{x}^{(k)} + t \mathbf{d}^{(k)}).$$



Minimalizovanou funkci proměnné t označíme $\Psi(t)$. Platí:

$$\Psi(t) = \mathbf{F}(\mathbf{x}^{(k)} + t \mathbf{d}^{(k)}) = \frac{1}{2} (\mathbf{x}^{(k)} + t \mathbf{d}^{(k)})^T \mathbf{A} (\mathbf{x}^{(k)} + t \mathbf{d}^{(k)}) - \mathbf{b}^T (\mathbf{x}^{(k)} + t \mathbf{d}^{(k)}).$$

$$\frac{d\Psi(t)}{dt} = t \mathbf{d}^{(k)T} \mathbf{A} \mathbf{d}^{(k)} + \underbrace{\mathbf{x}^{(k)T} \mathbf{A} \mathbf{d}^{(k)} - \mathbf{b}^T \mathbf{d}^{(k)}}_{\underbrace{(\mathbf{x}^{(k)T} \mathbf{A} - \mathbf{b}^T) \mathbf{d}^{(k)}}_{-\mathbf{d}^{(k)T}}}$$

$$\frac{d\Psi(t)}{dt} = t \mathbf{d}^{(k)T} \mathbf{A} \mathbf{d}^{(k)} - \mathbf{d}^{(k)T} \mathbf{d}^{(k)} = 0$$

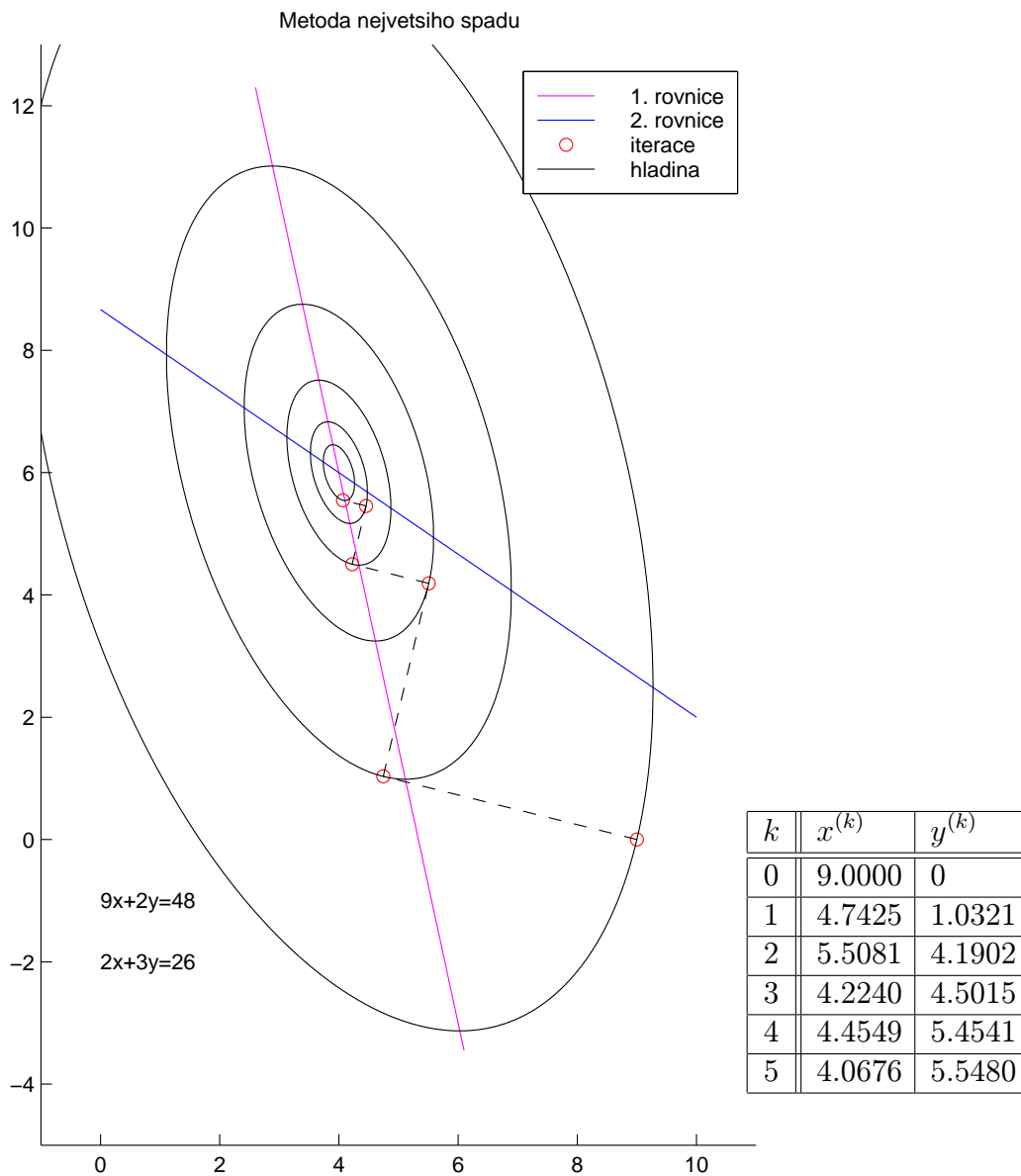
$$t^{(k)} = \frac{\mathbf{d}^{(k)T} \mathbf{d}^{(k)}}{\mathbf{d}^{(k)T} \mathbf{A} \mathbf{d}^{(k)}}$$

Poznámka: Ve výrazu pro derivaci $\Psi(t)$ jsme využili symetrii matice \mathbf{A} .

Algoritmus metody největšího spádu potom můžeme zapsat takto:

- 1) volba $\mathbf{x}^{(0)}, \varepsilon$
- 2) výpočet směru spádu $\mathbf{d}^{(k)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}$
- 3) výpočet koeficientu $t^{(k)} = \frac{\mathbf{d}^{(k)T} \mathbf{d}^{(k)}}{\mathbf{d}^{(k)T} \mathbf{A} \mathbf{d}^{(k)}}$
- 4) výpočet nové iterace $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)} \mathbf{d}^{(k)}$
- 5) $k = k + 1$ a zpět na 2) pokud $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| > \varepsilon$

GEOMETRICKÝ VÝZNAM METODY NEJVĚTŠÍHO SPÁDU



Pro soustavu dvou rovnic si opět lze udělat geometrickou představu. Všimněme si faktu, že vždy po sobě jdoucí iterace směru spádu, tj. $\mathbf{d}^{(k)}$ a $\mathbf{d}^{(k+1)}$ jsou na sebe kolmé.

Cvičení: Dokažte, že platí $\mathbf{d}^{(k)T} \mathbf{d}^{(k+1)} = 0$

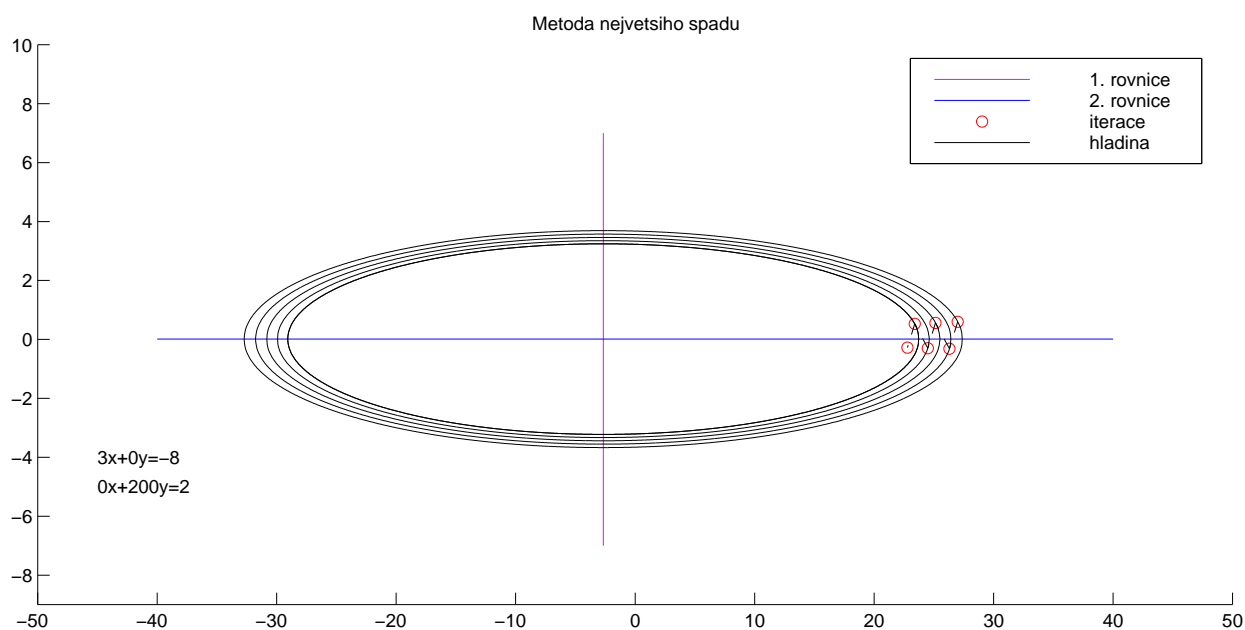
$$\begin{aligned}
 \mathbf{d}^{(k)T} \mathbf{d}^{(k+1)} &= \mathbf{d}^{(k)T} (\mathbf{b} - \mathbf{A}\mathbf{x}^{(k+1)}) = \\
 &= \mathbf{d}^{(k)T} (\mathbf{b} - \mathbf{A}(\mathbf{x}^{(k)} + t^{(k)} \mathbf{d}^{(k)})) = \\
 &= \mathbf{d}^{(k)T} (\mathbf{b} - \mathbf{A}\mathbf{x}^{(k)} - t^{(k)} \mathbf{A} \mathbf{d}^{(k)}) = \\
 &= \mathbf{d}^{(k)T} (\mathbf{d}^{(k)} - t^{(k)} \mathbf{A} \mathbf{d}^{(k)}) = \\
 &= \mathbf{d}^{(k)T} \mathbf{d}^{(k)} - t^{(k)} \mathbf{d}^{(k)T} \mathbf{A} \mathbf{d}^{(k)} = \\
 &= \mathbf{d}^{(k)T} \mathbf{d}^{(k)} - \frac{\mathbf{d}^{(k)T} \mathbf{d}^{(k)}}{\mathbf{d}^{(k)T} \mathbf{A} \mathbf{d}^{(k)}} \mathbf{d}^{(k)T} \mathbf{A} \mathbf{d}^{(k)} = \\
 &= \mathbf{d}^{(k)T} \mathbf{d}^{(k)} - \mathbf{d}^{(k)T} \mathbf{d}^{(k)} = 0
 \end{aligned}$$

Poznámka: V případě, že budou hladiny (elipsy) „velmi protáhlé“, bude metoda největšího spádu konvergovat velmi pomalu, nastane tzv. *cik-cak efekt*. Na druhou stranu, pokud budou hladiny (elipsy) „skoro kružnice“, bude metoda největšího spádu konvergovat velmi rychle. Nevýhodu cik-cak efektu odstraňuje nová metoda, tzv. **metoda sdružených gradientů**, která využívá důmyslnější volby směrů minimalizace, a sice tak, aby se neopakovali, jak k tomu docházelo u metody největšího spádu.

Příklad 1. Pomocí metody největšího spádu řešte soustavu

$$\begin{aligned} 3x + 0y &= -8 \\ 0x + 200y &= 2 \end{aligned}$$

Jako počáteční aproximaci volte $x^{(0)} = 27, y^{(0)} = 0.6$.



k	$x^{(k)}$	$y^{(k)}$
0	27.000000	0.600000
1	26.307758	-0.317804
2	25.139940	0.563008
3	24.491101	-0.297251
4	23.396504	0.528335
5	22.788346	-0.277987
\vdots	\vdots	\vdots
250	-2.657606	0.010180

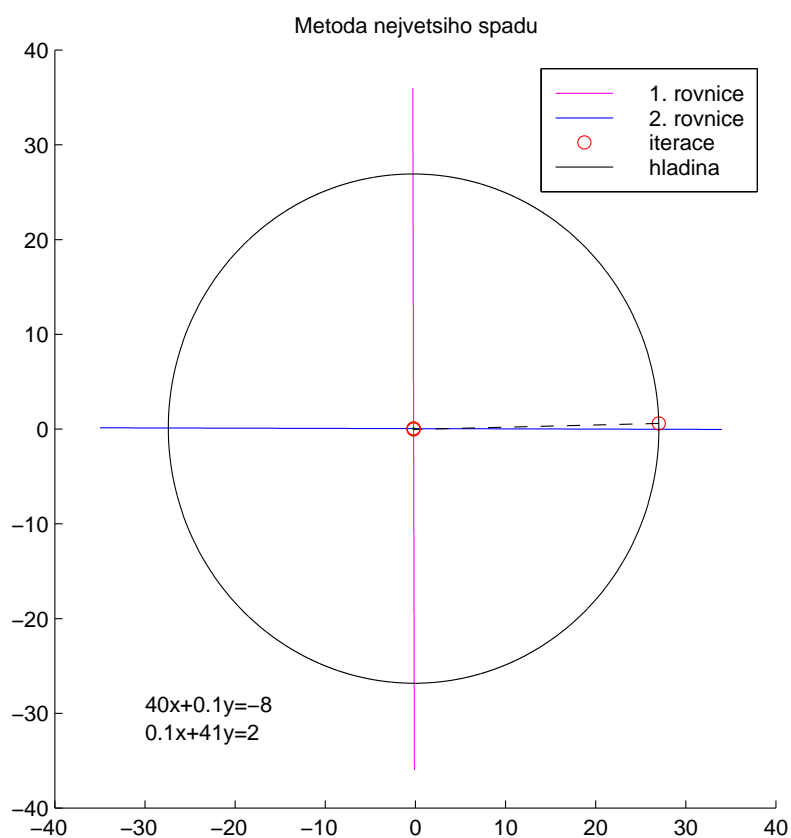
(Přesné řešení soustavy je $[\tilde{x}, \tilde{y}] = \left[-\frac{8}{3}, \frac{1}{100}\right]$.)

Příklad 2. Pomocí metody největšího spádu řešte soustavu

$$40x + 0.1y = -8$$

$$0.1x + 41y = 2$$

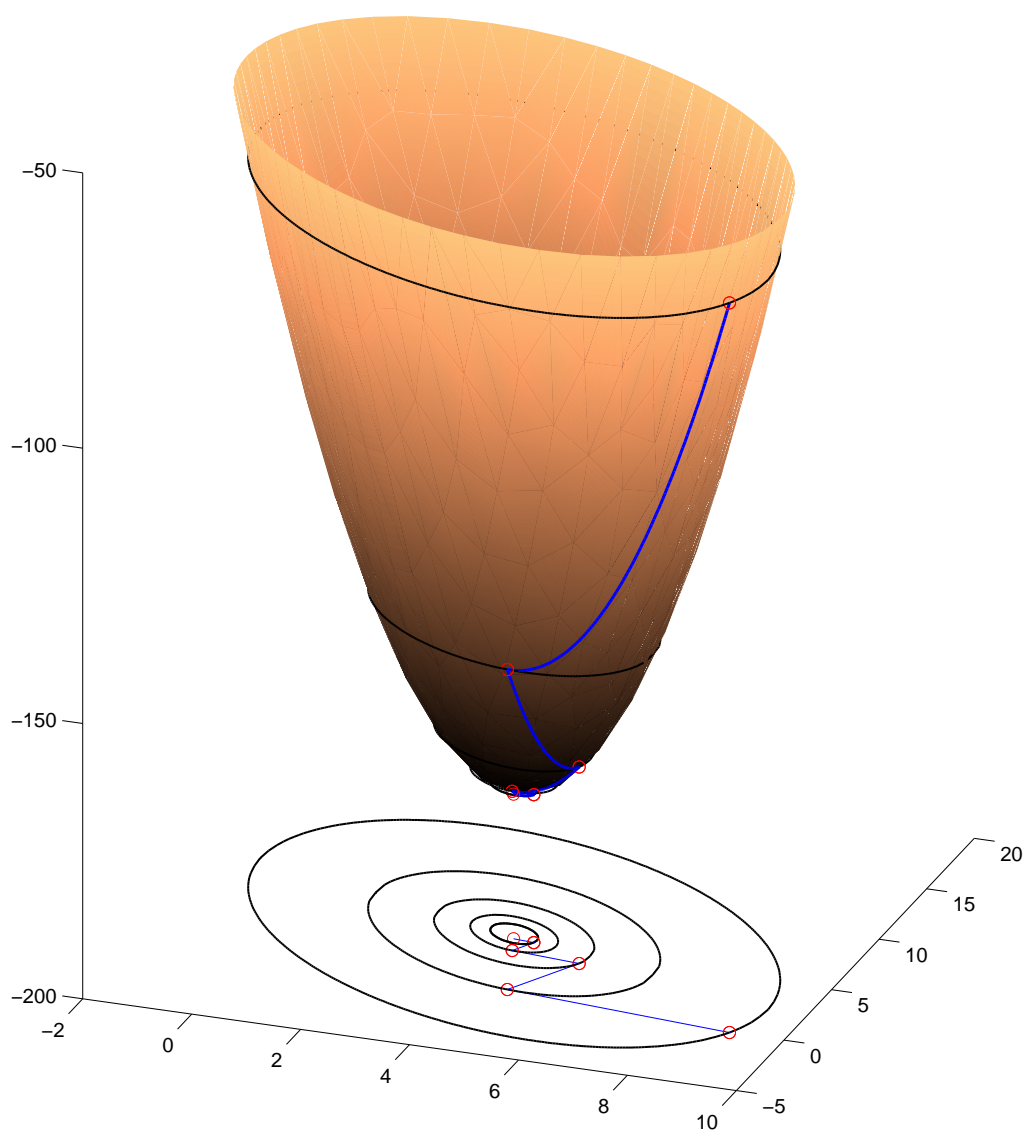
Jako počáteční aproximaci volte $x^{(0)} = 27, y^{(0)} = 0.6$.



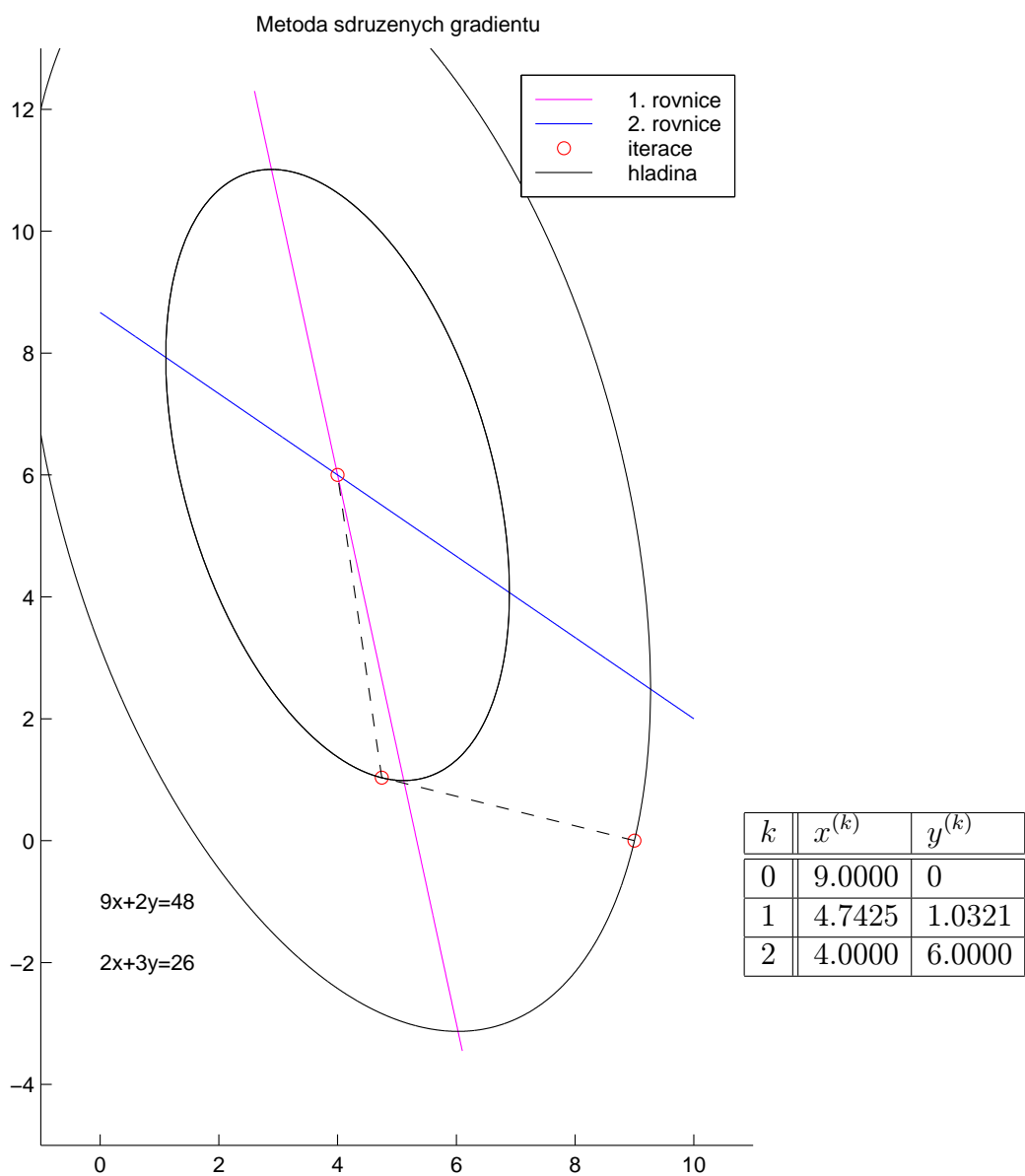
k	$x^{(k)}$	$y^{(k)}$
0	27.000000	0.600000
1	-0.197972	-0.032418
2	-0.199872	0.049274
3	-0.200123	0.049268

(Přesné řešení soustavy se na 6 desetinných míst shoduje s $[x^{(3)}, y^{(3)}]$.)

GEOMETRICKÝ VÝZNAM METODY NEJVĚTŠÍHO SPÁDU VE 3D



GEOMETRICKÝ VÝZNAM METODY SDRUŽENÝCH GRADIENTŮ



GEOMETRICKÝ VÝZNAM METODY SDRUŽENÝCH GRADIENTŮ VE 3D

