Iterační metody řešení soustav lineárních rovnic

Michal Čihák

27. prosince 2011

Přímé metody řešení soustav lineárních rovnic

- V přednáškách z lineární algebry jste se seznámili s několika metodami řešení soustav lineárních rovnic (Gaussova eliminační metoda, Cramerovo pravidlo, atd.)
- Při praktické realizaci těchto metod na počítači vznikají drobné nepřesnosti při výpočtech způsobené zaokrouhlovacími chybami.
- Čím větší je počet rovnic a neznámých, tím je větší počet operací prováděných při výpočtech a tím větší mohou být vzniklé nepřesnosti.

Přímé metody řešení soustav lineárních rovnic

- V přednáškách z lineární algebry jste se seznámili s několika metodami řešení soustav lineárních rovnic (Gaussova eliminační metoda, Cramerovo pravidlo, atd.)
- Při praktické realizaci těchto metod na počítači vznikají drobné nepřesnosti při výpočtech způsobené zaokrouhlovacími chybami.
- Čím větší je počet rovnic a neznámých, tím je větší počet operací prováděných při výpočtech a tím větší mohou být vzniklé nepřesnosti.

Přímé metody řešení soustav lineárních rovnic

- V přednáškách z lineární algebry jste se seznámili s několika metodami řešení soustav lineárních rovnic (Gaussova eliminační metoda, Cramerovo pravidlo, atd.)
- Při praktické realizaci těchto metod na počítači vznikají drobné nepřesnosti při výpočtech způsobené zaokrouhlovacími chybami.
- Čím větší je počet rovnic a neznámých, tím je větší počet operací prováděných při výpočtech a tím větší mohou být vzniklé nepřesnosti.

Problém zaokrouhlovacích chyb

Příklad: Řešte soustavu rovnic Gaussovou eliminační metodou při použití přesnosti na čtyři platné číslice.

$$\begin{aligned} 0{,}003000x_1 + 59{,}14x_2 &= 59{,}17 \\ 5{,}291x_1 - 6{,}130x_2 &= 46{,}78 \end{aligned}$$

Problém zaokrouhlovacích chyb

Příklad: Řešte soustavu rovnic Gaussovou eliminační metodou při použití přesnosti na čtyři platné číslice.

$$\begin{aligned} 0{,}003000x_1 + 59{,}14x_2 &= 59{,}17 \\ 5{,}291x_1 - 6{,}130x_2 &= 46{,}78 \end{aligned}$$

Po úpravě na trojúhelníkový tvar obdržíme:

$$\begin{aligned} 0{,}003000x_1 + 59{,}14x_2 &= 59{,}17 \\ -104300x_2 &\approx -104400 \end{aligned}$$

Problém zaokrouhlovacích chyb

Příklad: Řešte soustavu rovnic Gaussovou eliminační metodou při použití přesnosti na čtyři platné číslice.

$$\begin{aligned} 0{,}003000x_1 + 59{,}14x_2 &= 59{,}17 \\ 5{,}291x_1 - 6{,}130x_2 &= 46{,}78 \end{aligned}$$

Po úpravě na trojúhelníkový tvar obdržíme:

$$\begin{aligned} 0{,}003000x_1 + 59{,}14x_2 &= 59{,}17 \\ -104300x_2 &\approx -104400 \end{aligned}$$

Přitom přesný tvar má být:

$$\begin{aligned} 0{,}003000x_1 + 59{,}14x_2 &= 59{,}17 \\ -104309{,}37\overline{6}x_2 &= -104309{,}37\overline{6} \end{aligned}$$

Problém zaokrouhlovacích chyb

Příklad: Řešte soustavu rovnic Gaussovou eliminační metodou při použití přesnosti na čtyři platné číslice.

$$
\begin{aligned}
0{,}003000x_1 + 59{,}14x_2 &= 59{,}17 \\
5{,}291x_1 - 6{,}130x_2 &= 46{,}78
\end{aligned}
$$

Po úpravě na trojúhelníkový tvar obdržíme:

$$
\begin{aligned}
0{,}003000x_1 + 59{,}14x_2 &= 59{,}17 \\
-104300x_2 &\approx -104400
\end{aligned}
$$

Přitom přesný tvar má být:

$$
\begin{aligned}
0{,}003000x_1 + 59{,}14x_2 &= 59{,}17 \\
-104309{,}37\overline{6}x_2 &= -104309{,}37\overline{6}
\end{aligned}
$$

Výše uvedená nepřesnost vede k hodnotě $x_2 \approx 1{,}001$ namísto přesné hodnoty $x_2 = 1{,}000$ (zatím nic tragického).

Problém zaokrouhlovacích chyb

Příklad: Řešte soustavu rovnic Gaussovou eliminační metodou při použití přesnosti na čtyři platné číslice.

$$\begin{aligned} 0{,}003000x_1 + 59{,}14x_2 &= 59{,}17 \\ 5{,}291x_1 - 6{,}130x_2 &= 46{,}78 \end{aligned}$$

Po úpravě na trojúhelníkový tvar obdržíme:

$$\begin{aligned} 0{,}003000x_1 + 59{,}14x_2 &= 59{,}17 \\ -104300x_2 &\approx -104400 \end{aligned}$$

Přitom přesný tvar má být:

$$\begin{aligned} 0{,}003000x_1 + 59{,}14x_2 &= 59{,}17 \\ -104309{,}37\overline{6}x_2 &= -104309{,}37\overline{6} \end{aligned}$$

Výše uvedená nepřesnost vede k hodnotě $x_2 \approx 1{,}001$ namísto přesné hodnoty $x_2 = 1{,}000$ (zatím nic tragického).
Avšak $x_1 \approx -10{,}00$ namísto přesné hodnoty $x_1 = 10{,}00$ (což je průšvih!).

Iterační metody řešení soustav lineárních rovnic

- Iterační metody nachází své uplatnění tam, kde selhávají přímé metody.
- Používají se zejména v případě velkého počtu rovnic a neznámých – matice soustavy má velké rozměry a je *řídká* (obsahuje hodně nul).
- Je zajímavé, že takové matice se v praktických aplikacích vyskytují poměrně často, což ještě umocňuje význam iteračních metod.

Iterační metody řešení soustav lineárních rovnic

- Iterační metody nachází své uplatnění tam, kde selhávají přímé metody.
- Používají se zejména v případě velkého počtu rovnic a neznámých – matice soustavy má velké rozměry a je *řídká* (obsahuje hodně nul).
- Je zajímavé, že takové matice se v praktických aplikacích vyskytují poměrně často, což ještě umocňuje význam iteračních metod.

Iterační metody řešení soustav lineárních rovnic

- Iterační metody nachází své uplatnění tam, kde selhávají přímé metody.
- Používají se zejména v případě velkého počtu rovnic a neznámých – matice soustavy má velké rozměry a je *řídká* (obsahuje hodně nul).
- Je zajímavé, že takové matice se v praktických aplikacích vyskytují poměrně často, což ještě umocňuje význam iteračních metod.

Jacobiova iterační metoda – ukázka

Příklad: Je dána soustava

$$\begin{array}{rcrcrcrcl}
10x_1 & - & x_2 & + & 2x_3 & & & = & 6, \\
-x_1 & + & 11x_2 & - & x_3 & + & 3x_4 & = & 25, \\
2x_1 & - & x_2 & + & 10x_3 & - & x_4 & = & -11, \\
 & & 3x_2 & - & x_3 & + & 8x_4 & = & 15
\end{array}$$

Jacobiova iterační metoda – ukázka

Příklad: Je dána soustava

$$\begin{array}{rcrcrcrcr}
10x_1 & - & x_2 & + & 2x_3 & & & = & 6, \\
-x_1 & + & 11x_2 & - & x_3 & + & 3x_4 & = & 25, \\
2x_1 & - & x_2 & + & 10x_3 & - & x_4 & = & -11, \\
 & & 3x_2 & - & x_3 & + & 8x_4 & = & 15
\end{array}$$

Řešení začneme tím, že z j-té rovnice vyjádříme neznámou x_j pro $j = 1, 2, 3, 4$

$$\begin{array}{rcrcrcrcrcr}
x_1 & = & & & \frac{1}{10}x_2 & - & \frac{1}{5}x_3 & & & + & \frac{3}{5}, \\
x_2 & = & \frac{1}{11}x_1 & & & + & \frac{1}{11}x_3 & - & \frac{3}{11}x_4 & + & \frac{25}{11}, \\
x_3 & = & -\frac{1}{5}x_1 & + & \frac{1}{10}x_2 & & & + & \frac{1}{10}x_4 & - & \frac{11}{10}, \\
x_4 & = & & & -\frac{3}{8}x_2 & + & \frac{1}{8}x_3 & & & + & \frac{15}{8}.
\end{array}$$

Jacobiova iterační metoda – ukázka

Pro zjednodušení dalšího zápisu budeme používat maticové zápisy. Vektor neznámých označíme $\mathbf{x} = (x_1, x_2, x_3, x_4)^t$.
Nyní zvolíme nějakou počáteční aproximaci vektoru neznámých $\mathbf{x}^{(0)} = (0, 0, 0, 0)^t$. Potom první iteraci $\mathbf{x}^{(1)}$ vektoru neznámých získáme takto

$$\begin{aligned}
x_1^{(1)} &= & & \tfrac{1}{10}x_2^{(0)} & - & \tfrac{1}{5}x_3^{(0)} & & & + & \tfrac{3}{5} & = & 0.6000,\\
x_2^{(1)} &= \tfrac{1}{11}x_1^{(0)} & & & + & \tfrac{1}{11}x_3^{(0)} & - & \tfrac{3}{11}x_4^{(0)} & + & \tfrac{25}{11} & = & 2.2727,\\
x_3^{(1)} &= -\tfrac{1}{5}x_1^{(0)} & + & \tfrac{1}{10}x_2^{(0)} & & & + & \tfrac{1}{10}x_4^{(0)} & - & \tfrac{11}{10} & = & -1.1000,\\
x_4^{(1)} &= & & -\tfrac{3}{8}x_2^{(0)} & + & \tfrac{1}{8}x_3^{(0)} & & & + & \tfrac{15}{8} & = & 1.8750.
\end{aligned}$$

Jacobiova iterační metoda – ukázka

Další iterace $\mathbf{x}^{(k)}$ získáme podobně a jsou uvedeny v tabulce

k	0	1	2	3	4	5	6	7	8	9	10
$x_1^{(k)}$	0.000	0.6000	1.0473	0.9326	1.0152	0.9890	1.0032	0.9981	1.0006	0.9997	1.0001
$x_2^{(k)}$	0.0000	2.2727	1.7159	2.053	1.9537	2.0114	1.9922	2.0023	1.9987	2.0004	1.9998
$x_3^{(k)}$	0.0000	-1.1000	-0.8052	-1.0493	-0.9681	-1.0103	-0.9945	-1.0020	-0.9990	-1.0004	-0.9998
$x_4^{(k)}$	0.0000	1.8750	0.8852	1.1309	0.9739	1.0214	0.9944	1.0036	0.9989	1.0006	0.9998

Jacobiova iterační metoda – ukázka

Další iterace $\mathbf{x}^{(k)}$ získáme podobně a jsou uvedeny v tabulce

k	0	1	2	3	4	5	6	7	8	9	10
$x_1^{(k)}$	0.000	0.6000	1.0473	0.9326	1.0152	0.9890	1.0032	0.9981	1.0006	0.9997	1.0001
$x_2^{(k)}$	0.0000	2.2727	1.7159	2.053	1.9537	2.0114	1.9922	2.0023	1.9987	2.0004	1.9998
$x_3^{(k)}$	0.0000	-1.1000	-0.8052	-1.0493	-0.9681	-1.0103	-0.9945	-1.0020	-0.9990	-1.0004	-0.9998
$x_4^{(k)}$	0.0000	1.8750	0.8852	1.1309	0.9739	1.0214	0.9944	1.0036	0.9989	1.0006	0.9998

Proces výpočtu dalších iterací byl ukončen ve chvíli, kdy hodnota $\max_{i=1,\ldots,n}\{|x_i^{(k)} - x_i^{(k-1)}|\}$ klesla pod předem danou toleranci $0{,}001$.

Jacobiova iterační metoda – ukázka

Další iterace $\mathbf{x}^{(k)}$ získáme podobně a jsou uvedeny v tabulce

k	0	1	2	3	4	5	6	7	8	9	10
$x_1^{(k)}$	0.000	0.6000	1.0473	0.9326	1.0152	0.9890	1.0032	0.9981	1.0006	0.9997	1.0001
$x_2^{(k)}$	0.0000	2.2727	1.7159	2.053	1.9537	2.0114	1.9922	2.0023	1.9987	2.0004	1.9998
$x_3^{(k)}$	0.0000	−1.1000	−0.8052	−1.0493	−0.9681	−1.0103	−0.9945	−1.0020	−0.9990	−1.0004	−0.9998
$x_4^{(k)}$	0.0000	1.8750	0.8852	1.1309	0.9739	1.0214	0.9944	1.0036	0.9989	1.0006	0.9998

Proces výpočtu dalších iterací byl ukončen ve chvíli, kdy hodnota $\max_{i=1,\dots,n}\{|x_i^{(k)} - x_i^{(k-1)}|\}$ klesla pod předem danou toleranci $0{,}001$. Přesné řešení zadané dané soustavy rovnic je $\mathbf{x} = (1, 2, -1, 1)^t$.

Pomocné pojmy z lineární algebry

- Připomeňme pojem *vlastních čísel matice*.
- Pro čtvercovou $n \times n$ matici A definujeme *charakteristický polynom* matice A vztahem $p(\lambda) = \det(A - \lambda I)$, kde I je jednotková matice.
- Kořeny tohoto charakteristického polynomu se nazývají *vlastní čísla* matice A.
- *Spektrální poloměr* matice A je definován jako největší z absolutních hodnot vlastních čísel matice A a značíme jej $\rho(A)$.

Pomocné pojmy z lineární algebry

- Připomeňme pojem *vlastních čísel matice*.
- Pro čtvercovou $n \times n$ matici A definujeme *charakteristický polynom* matice A vztahem $p(\lambda) = \det(A - \lambda I)$, kde I je jednotková matice.
- Kořeny tohoto charakteristického polynomu se nazývají *vlastní čísla* matice A.
- *Spektrální poloměr* matice A je definován jako největší z absolutních hodnot vlastních čísel matice A a značíme jej $\rho(A)$.

Pomocné pojmy z lineární algebry

- Připomeňme pojem *vlastních čísel matice*.
- Pro čtvercovou $n \times n$ matici A definujeme *charakteristický polynom* matice A vztahem $p(\lambda) = \det(A - \lambda I)$, kde I je jednotková matice.
- Kořeny tohoto charakteristického polynomu se nazývají *vlastní čísla* matice A.
- *Spektrální poloměr* matice A je definován jako největší z absolutních hodnot vlastních čísel matice A a značíme jej $\rho(A)$.

Pomocné pojmy z lineární algebry

- Připomeňme pojem *vlastních čísel matice*.
- Pro čtvercovou $n \times n$ matici A definujeme *charakteristický polynom* matice A vztahem $p(\lambda) = \det(A - \lambda I)$, kde I je jednotková matice.
- Kořeny tohoto charakteristického polynomu se nazývají *vlastní čísla* matice A.
- *Spektrální poloměr* matice A je definován jako největší z absolutních hodnot vlastních čísel matice A a značíme jej $\rho(A)$.

Jacobiova iterační metoda – kritérium konvergence

Posloupnost iterací

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c},$$

kde $k = 1, 2, \ldots$, konverguje k jedinému řešení soustavy rovnic $\mathbf{x} = T\mathbf{x} + \mathbf{c}$ pro libovolné $\mathbf{x}^{(k)} \in \mathbb{R}^n$ právě tehdy, když $\rho(T) < 1$.

Gauss-Seidlova iterační metoda

- Při Jacobiově metodě se při výpočtu komponent vektoru k-tých iterací používají výhradně komponenty vektoru $(k-1)$-ních iterací.
- Přitom při výpočtu i-té komponenty $x_i^{(k)}$ vektoru k-tých iterací už máme spočítány komponenty $x_1^{(k)}, x_2^{(k)}, \ldots, x_{i-1}^{(k)}$ tohoto vektoru.
- Tyto čerstvě spočítané komponenty jsou pravděpodobně lepšími aproximacemi hodnot neznámých, než dříve spočítané komponenty $x_1^{(k-1)}, x_2^{(k-1)}, \ldots, x_{i-1}^{(k-1)}$.
- Základní myšlenkou Gauss-Seidlovy iterační metody je použít při výpočtu i-té komponenty $x_i^{(k)}$ vektoru k-tých iterací hodnoty $x_1^{(k)}, x_2^{(k)}, \ldots, x_{i-1}^{(k)}, x_i^{(k-1)}, \ldots, x_n^{(k-1)}$.

Gauss-Seidlova iterační metoda

- Při Jacobiově metodě se při výpočtu komponent vektoru k-tých iterací používají výhradně komponenty vektoru $(k-1)$-ních iterací.
- Přitom při výpočtu i-té komponenty $x_i^{(k)}$ vektoru k-tých iterací už máme spočítány komponenty $x_1^{(k)}, x_2^{(k)}, \ldots, x_{i-1}^{(k)}$ tohoto vektoru.
- Tyto čerstvě spočítané komponenty jsou pravděpodobně lepšími aproximacemi hodnot neznámých, než dříve spočítané komponenty $x_1^{(k-1)}, x_2^{(k-1)}, \ldots, x_{i-1}^{(k-1)}$.
- Základní myšlenkou Gauss-Seidlovy iterační metody je použít při výpočtu i-té komponenty $x_i^{(k)}$ vektoru k-tých iterací hodnoty $x_1^{(k)}, x_2^{(k)}, \ldots, x_{i-1}^{(k)}, x_i^{(k-1)}, \ldots, x_n^{(k-1)}$.

Gauss-Seidlova iterační metoda

- Při Jacobiově metodě se při výpočtu komponent vektoru k-tých iterací používají výhradně komponenty vektoru $(k-1)$-ních iterací.
- Přitom při výpočtu i-té komponenty $x_i^{(k)}$ vektoru k-tých iterací už máme spočítány komponenty $x_1^{(k)}, x_2^{(k)}, \dots, x_{i-1}^{(k)}$ tohoto vektoru.
- Tyto čerstvě spočítané komponenty jsou pravděpodobně lepšími aproximacemi hodnot neznámých, než dříve spočítané komponenty $x_1^{(k-1)}, x_2^{(k-1)}, \dots, x_{i-1}^{(k-1)}$.
- Základní myšlenkou Gauss-Seidlovy iterační metody je použít při výpočtu i-té komponenty $x_i^{(k)}$ vektoru k-tých iterací hodnoty $x_1^{(k)}, x_2^{(k)}, \dots, x_{i-1}^{(k)}, x_i^{(k-1)}, \dots, x_n^{(k-1)}$.

Gauss-Seidlova iterační metoda

- Při Jacobiově metodě se při výpočtu komponent vektoru k-tých iterací používají výhradně komponenty vektoru $(k-1)$-ních iterací.
- Přitom při výpočtu i-té komponenty $x_i^{(k)}$ vektoru k-tých iterací už máme spočítány komponenty $x_1^{(k)}, x_2^{(k)}, \dots, x_{i-1}^{(k)}$ tohoto vektoru.
- Tyto čerstvě spočítané komponenty jsou pravděpodobně lepšími aproximacemi hodnot neznámých, než dříve spočítané komponenty $x_1^{(k-1)}, x_2^{(k-1)}, \dots, x_{i-1}^{(k-1)}$.
- Základní myšlenkou Gauss-Seidlovy iterační metody je použít při výpočtu i-té komponenty $x_i^{(k)}$ vektoru k-tých iterací hodnoty $x_1^{(k)}, x_2^{(k)}, \dots, x_{i-1}^{(k)}, x_i^{(k-1)}, \dots, x_n^{(k-1)}$.

Gauss-Seidlova iterační metoda – ukázka

Příklad: Je dána soustava

$$\begin{array}{rcrcrcrcr}
10x_1 & - & x_2 & + & 2x_3 & & & = & 6, \\
-x_1 & + & 11x_2 & - & x_3 & + & 3x_4 & = & 25, \\
2x_1 & - & x_2 & + & 10x_3 & - & x_4 & = & -11, \\
 & & 3x_2 & - & x_3 & + & 8x_4 & = & 15
\end{array}$$

Gauss-Seidlova iterační metoda – ukázka

Příklad: Je dána soustava

$$
\begin{array}{rcrcrcrcl}
10x_1 & - & x_2 & + & 2x_3 & & & = & 6, \\
-x_1 & + & 11x_2 & - & x_3 & + & 3x_4 & = & 25, \\
2x_1 & - & x_2 & + & 10x_3 & - & x_4 & = & -11, \\
 & & 3x_2 & - & x_3 & + & 8x_4 & = & 15
\end{array}
$$

Při výpočtu hodnoty k-té iterace i-té neznámé $x_i^{(k)}$ použijeme hodnoty $x_1^{(k)}, x_2^{(k)}, \ldots, x_{i-1}^{(k)}, x_i^{(k-1)}, \ldots, x_4^{(k-1)}$

$$
\begin{array}{rcrcrcrcrcl}
x_1^{(k)} & = & & & \frac{1}{10}x_2^{(k-1)} & - & \frac{1}{5}x_3^{(k-1)} & & & + & \frac{3}{5}, \\
x_2^{(k)} & = & \frac{1}{11}x_1^{(k)} & & & + & \frac{1}{11}x_3^{(k-1)} & - & \frac{3}{11}x_4^{(k-1)} & + & \frac{25}{11}, \\
x_3^{(k)} & = & -\frac{1}{5}x_1^{(k)} & + & \frac{1}{10}x_2^{(k)} & & & + & \frac{1}{10}x_4^{(k-1)} & - & \frac{11}{10}, \\
x_4^{(k)} & = & & & -\frac{3}{8}x_2^{(k)} & + & \frac{1}{8}x_3^{(k)} & & & + & \frac{15}{8}.
\end{array}
$$

Gauss-Seidlova iterační metoda – ukázka

V tabulce jsou uvedeny jednotlivé iterace postupně vypočtené Gauss-Seidlovou iterační metodou

k	0	1	2	3	4	5
$x_1^{(k)}$	0.0000	0.6000	1.030	1.0065	1.0009	1.0001
$x_2^{(k)}$	0.0000	2.3272	2.037	2.0036	2.0003	2.0000
$x_3^{(k)}$	0.0000	−0.9873	−1.014	−1.0025	−1.0003	−1.0000
$x_4^{(k)}$	0.0000	0.8789	0.9844	0.9983	0.9999	1.0000

Gauss-Seidlova iterační metoda – ukázka

V tabulce jsou uvedeny jednotlivé iterace postupně vypočtené Gauss-Seidlovou iterační metodou

k	0	1	2	3	4	5
$x_1^{(k)}$	0.0000	0.6000	1.030	1.0065	1.0009	1.0001
$x_2^{(k)}$	0.0000	2.3272	2.037	2.0036	2.0003	2.0000
$x_3^{(k)}$	0.0000	-0.9873	-1.014	-1.0025	-1.0003	-1.0000
$x_4^{(k)}$	0.0000	0.8789	0.9844	0.9983	0.9999	1.0000

Proces výpočtu dalších iterací byl ukončen ve chvíli, kdy hodnota $\max_{i=1,\dots,n}\{|x_i^{(k)} - x_i^{(k-1)}|\}$ klesla pod předem danou toleranci $0{,}001$.

Gauss-Seidlova iterační metoda – ukázka

V tabulce jsou uvedeny jednotlivé iterace postupně vypočtené Gauss-Seidlovou iterační metodou

k	0	1	2	3	4	5
$x_1^{(k)}$	0.0000	0.6000	1.030	1.0065	1.0009	1.0001
$x_2^{(k)}$	0.0000	2.3272	2.037	2.0036	2.0003	2.0000
$x_3^{(k)}$	0.0000	-0.9873	-1.014	-1.0025	-1.0003	-1.0000
$x_4^{(k)}$	0.0000	0.8789	0.9844	0.9983	0.9999	1.0000

Proces výpočtu dalších iterací byl ukončen ve chvíli, kdy hodnota $\max_{i=1,\ldots,n}\{|x_i^{(k)} - x_i^{(k-1)}|\}$ klesla pod předem danou toleranci $0{,}001$. Přesné řešení zadané dané soustavy rovnic je $\mathbf{x} = (1, 2, -1, 1)^t$.

Gauss-Seidlova iterační metoda – ukázka

V tabulce jsou uvedeny jednotlivé iterace postupně vypočtené Gauss-Seidlovou iterační metodou

k	0	1	2	3	4	5
$x_1^{(k)}$	0.0000	0.6000	1.030	1.0065	1.0009	1.0001
$x_2^{(k)}$	0.0000	2.3272	2.037	2.0036	2.0003	2.0000
$x_3^{(k)}$	0.0000	-0.9873	-1.014	-1.0025	-1.0003	-1.0000
$x_4^{(k)}$	0.0000	0.8789	0.9844	0.9983	0.9999	1.0000

Proces výpočtu dalších iterací byl ukončen ve chvíli, kdy hodnota $\max_{i=1,\ldots,n}\{|x_i^{(k)} - x_i^{(k-1)}|\}$ klesla pod předem danou toleranci $0{,}001$. Přesné řešení zadané dané soustavy rovnic je $\mathbf{x} = (1, 2, -1, 1)^t$. Všimněte si, že Gauss-Seidlovou metodou jsme dospěli k předepsané přesnosti řešení po 5 iteracích, zatímco při použití Jacobiovy metody jsme k dosažení podobné přesnosti potřebovali 10 iterací.

SOR metoda

Metoda SOR je podobná Gauss-Seidlově metodě, ale využívá vhodný součinový koeficient pro rychlejší dosažení požadované přesnosti aproximace řešení soustavy.

SOR metoda

Metoda SOR je podobná Gauss-Seidlově metodě, ale využívá vhodný součinový koeficient pro rychlejší dosažení požadované přesnosti aproximace řešení soustavy.
Pro výpočet hodnoty k-té iterace i-té neznámé $x_i^{(k)}$ použijeme vztah

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

kde ω je zmíněný součinový koeficient.

- Pro $\omega = 1$ je metoda totožná Gauss-Seidlovu metodu.
- Pro $\omega > 1$ se metoda nazývá *superrelaxační metoda (metoda SOR)*.

SOR metoda

Metoda SOR je podobná Gauss-Seidlově metodě, ale využívá vhodný součinový koeficient pro rychlejší dosažení požadované přesnosti aproximace řešení soustavy.
Pro výpočet hodnoty k-té iterace i-té neznámé $x_i^{(k)}$ použijeme vztah

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

kde ω je zmíněný součinový koeficient.

- Pro $\omega = 1$ je metoda totožná Gauss-Seidlovu metodu.
- Pro $\omega > 1$ se metoda nazývá *superrelaxační metoda (metoda SOR)*.

SOR metoda

Metoda SOR je podobná Gauss-Seidlově metodě, ale využívá vhodný součinový koeficient pro rychlejší dosažení požadované přesnosti aproximace řešení soustavy.
Pro výpočet hodnoty k-té iterace i-té neznámé $x_i^{(k)}$ použijeme vztah

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

kde ω je zmíněný součinový koeficient.

- Pro $\omega = 1$ je metoda totožná Gauss-Seidlovu metodu.
- Pro $\omega > 1$ se metoda nazývá *superrelaxační metoda (metoda SOR)*.

SOR metoda – ukázka

Příklad: Je dána soustava

$$\begin{array}{rcrcrcr} 4x_1 & + & 3x_2 & & & = & 24, \\ 3x_1 & + & 4x_2 & - & x_3 & = & 30, \\ & & -x_2 & + & 4x_3 & = & -24 \end{array}$$

SOR metoda – ukázka

Příklad: Je dána soustava

$$
\begin{array}{rcrcrcr}
4x_1 & + & 3x_2 & & & = & 24, \\
3x_1 & + & 4x_2 & - & x_3 & = & 30, \\
 & & -x_2 & + & 4x_3 & = & -24
\end{array}
$$

Při použití Gauss-Seidlovy metody vypočteme k-tou iteraci hodnot neznámých pomocí vztahů

$$
\begin{aligned}
x_1^{(k)} &= -0.75x_2^{(k-1)} + 6, \\
x_2^{(k)} &= -0.75x_1^{(k)} + 0.25x_3^{(k-1)} + 7.5, \\
x_3^{(k)} &= 0.25x_2^{(k)} - 6,
\end{aligned}
$$

SOR metoda – ukázka

Příklad: Je dána soustava

$$\begin{array}{rcrcrcl} 4x_1 & + & 3x_2 & & & = & 24, \\ 3x_1 & + & 4x_2 & - & x_3 & = & 30, \\ & & -x_2 & + & 4x_3 & = & -24 \end{array}$$

Při použití Gauss-Seidlovy metody vypočteme k-tou iteraci hodnot neznámých pomocí vztahů

$$\begin{aligned} x_1^{(k)} &= -0.75x_2^{(k-1)} + 6, \\ x_2^{(k)} &= -0.75x_1^{(k)} + 0.25x_3^{(k-1)} + 7.5, \\ x_3^{(k)} &= 0.25x_2^{(k)} - 6, \end{aligned}$$

Při použití metody SOR s volbou součinového koeficientu $\omega = 1{,}25$ vypočteme k-tou iteraci hodnot neznámých pomocí vztahů

$$\begin{aligned} x_1^{(k)} &= -0.25x_1^{(k-1)} - 0.9375x_2^{(k-1)} + 7.5, \\ x_2^{(k)} &= -0.9375x_1^{(k)} - 0.25x_2^{(k-1)} + 0.3125x_3^{(k-1)} + 9.375, \\ x_3^{(k)} &= 0.3125x_2^{(k)} - 0.25x_3^{(k-1)} - 7.5. \end{aligned}$$

SOR metoda – ukázka

V první tabulce jsou uvedeny jednotlivé iterace postupně vypočtené Gauss-Seidlovou metodou pro počáteční aproximaci $\mathbf{x}^{(0)} = (1, 1, 1)^t$:

k	0	1	2	3	4	5	6	7
$x_1^{(k)}$	1	5.250000	3.1406250	3.0878906	3.0549316	3.0343323	3.0214577	3.0134110
$x_1^{(k)}$	1	3.812500	3.8828125	3.9267578	3.9542236	3.9713898	3.9821186	3.9888241
$x_1^{(k)}$	1	-5.046875	-5.0292969	-5.0183105	-5.0114441	-5.0071526	-5.0044703	-5.0027940

SOR metoda – ukázka

V první tabulce jsou uvedeny jednotlivé iterace postupně vypočtené Gauss-Seidlovou metodou pro počáteční aproximaci $\mathbf{x}^{(0)} = (1, 1, 1)^t$:

k	0	1	2	3	4	5	6	7
$x_1^{(k)}$	1	5.250000	3.1406250	3.0878906	3.0549316	3.0343323	3.0214577	3.0134110
$x_1^{(k)}$	1	3.812500	3.8828125	3.9267578	3.9542236	3.9713898	3.9821186	3.9888241
$x_1^{(k)}$	1	−5.046875	−5.0292969	−5.0183105	−5.0114441	−5.0071526	−5.0044703	−5.0027940

Ve druhé tabulce jsou uvedeny jednotlivé iterace postupně vypočtené SOR metodou pro počáteční aproximaci $\mathbf{x}^{(0)} = (1, 1, 1)^t$ a součinový koeficient $\omega = 1{,}25$:

k	0	1	2	3	4	5	6	7
$x_1^{(k)}$	1	6.312500	2.6223145	3.1333027	2.9570512	3.0037211	2.9963276	3.0000498
$x_2^{(k)}$	1	3.5195313	3.9585266	4.0102646	4.0074838	4.0029250	4.0009262	4.0002586
$x_3^{(k)}$	1	−6.6501465	−4.6004238	−5.0966863	−4.9734897	−5.0057135	−4.9982822	−5.0003486

SOR metoda – ukázka

V první tabulce jsou uvedeny jednotlivé iterace postupně vypočtené Gauss-Seidlovou metodou pro počáteční aproximaci $\mathbf{x}^{(0)} = (1, 1, 1)^t$:

k	0	1	2	3	4	5	6	7
$x_1^{(k)}$	1	5.250000	3.1406250	3.0878906	3.0549316	3.0343323	3.0214577	3.0134110
$x_1^{(k)}$	1	3.812500	3.8828125	3.9267578	3.9542236	3.9713898	3.9821186	3.9888241
$x_1^{(k)}$	1	−5.046875	−5.0292969	−5.0183105	−5.0114441	−5.0071526	−5.0044703	−5.0027940

Ve druhé tabulce jsou uvedeny jednotlivé iterace postupně vypočtené SOR metodou pro počáteční aproximaci $\mathbf{x}^{(0)} = (1, 1, 1)^t$ a součinový koeficient $\omega = 1{,}25$:

k	0	1	2	3	4	5	6	7
$x_1^{(k)}$	1	6.312500	2.6223145	3.1333027	2.9570512	3.0037211	2.9963276	3.0000498
$x_2^{(k)}$	1	3.5195313	3.9585266	4.0102646	4.0074838	4.0029250	4.0009262	4.0002586
$x_3^{(k)}$	1	−6.6501465	−4.6004238	−5.0966863	−4.9734897	−5.0057135	−4.9982822	−5.0003486

Přesné řešení zadané dané soustavy rovnic je $\mathbf{x} = (3, 4, -5)^t$.

SOR metoda – dokončení ukázky

Pro ilustraci: K dosažení přesnosti dané například tolerancí 10^{-7} bychom potřebovali 34 iterací Gauss-Seidlovy metody, ale pouze 14 iterací metody SOR.