Пловдивски университет „Паисий Хилендарски”
Факултет по математика и информатика
Катедра “Приложна математика и моделиране”

“Компютърни числени методи”
ЛЕКЦИЯ 3

Проф. д-р Снежана Гочева-Илиева, snow@uni-plovdiv.bg

Он-лайн обучение: www.fmi-plovdiv.org/evlm
www.fmi-plovdiv.org/evlm/DBbg - числени методи

Литература:

1. Бояджиев Д., Гочева С., Макрелов И., Попова Л. – Ръководство по числени методи – част 1, Издания: 2003, 2006, 2010.
2. Семерджиев Х., Боянов Б., Числени методи, ПУ.
3. Гочева-Илиева С., Въведение в система Mathematica, ЕксПрес, Габрово, 2009.

Съдържание

Числени методи за решаване на системи уравнения

1. Системи линейни уравнения, детерминанти и обратни матрици

Постановка на задачата. Търси се решението на системата линейни алгебрични уравнения

$$Ax = b,$$

където A е матрица с реални коефициенти с размерност $n \times n$, x – вектор на неизвестните, b - дясна част:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ & & \cdots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{pmatrix} \neq \vec{0} \; .$$

В разгънат вид системата е:

$$
\left|\begin{array}{llll}
a_{11}x_1 & +a_{12}x_2 & \cdots & +a_{1n}x_n = b_1 \\
a_{21}x_1 & +a_{22}x_2 & & +a_{2n}x_n = b_2 \\
 & & \cdots & \\
 & & & \\
 & & & \\
a_{n1}x_1 & +a_{n2}x_2 & \cdots & +a_{nn}x_n = b_n
\end{array}\right. .
$$

Без ограничения на общността ще считаме, че задачата е коректна, т.е. съществува единствено решение на тази система. Както е добре известно от линейната алгебра, необходимо и достатъчно условие за това е детерминантата $\det A \neq 0$. Условието за устойчивост също подлежи на разглеждане.

В курса по Линейна алгебра са изучавани методът на Гаус и методът на Гаус-Жордан. Затова вместо тези методи в нашия курс ще покажем как се изчислява решението на задачата с помощта на система *Mathematica*.

Пример. Да се реши системата уравнения, да се намери детерминантата и обратната матрица.

$$\begin{cases} 2x_1 + x_2 + 2x_3 + 3x_4 = -1 \\ -2x_1 + 3x_2 + 2x_3 - 3x_4 = 2 \\ \quad + 4x_2 + 2x_3 + 3x_4 = 4 \\ x_1 + x_2 + x_3 + x_4 = 5 \end{cases}.$$

Записваме и изпълняваме следните няколко клетки в *Mathematica*:

```
Решаване на системи уравнения от вида A.x = b ,
намиране на детерминантата и обратната матрица на A.

a = {{2, 1, 2, 3}, {-2, 3, 2, -3}, {0, 4, 2, 3}, {1, 1, 1, 1}};
MatrixForm[a]
b = {-1, 2, 4, 5}; MatrixForm[b]
```

$$\begin{pmatrix} 2 & 1 & 2 & 3 \\ -2 & 3 & 2 & -3 \\ 0 & 4 & 2 & 3 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} -1 \\ 2 \\ 4 \\ 5 \end{pmatrix}$$

```
xx = LinearSolve[a, b]
```

$$\left\{\frac{169}{20}, \frac{73}{10}, -\frac{141}{20}, -\frac{37}{10}\right\}$$

```
(*  Проверка на решението със заместване *)
a.xx - b
```

```
{0, 0, 0, 0}
```

```
(* Детерминанта на матрицата A *)
Det[a]
```

```
20
```

```
(*  Обратна матрица *)
obr = Inverse[a]; MatrixForm[obr]
```

$$\begin{pmatrix} -\frac{11}{20} & -\frac{3}{20} & -\frac{1}{5} & \frac{9}{5} \\ -\frac{7}{10} & -\frac{1}{10} & \frac{1}{5} & \frac{6}{5} \\ \frac{19}{20} & \frac{7}{20} & -\frac{1}{5} & -\frac{6}{5} \\ \frac{3}{10} & -\frac{1}{10} & \frac{1}{5} & -\frac{4}{5} \end{pmatrix}$$

Отг.: Точното решение е: $x_1 = \frac{169}{20}$, $x_2 = \frac{73}{10}$, $x_3 = -\frac{141}{20}$, $x_2 = -\frac{37}{10}$,
детерминантата е =20, обратната матрица – получената по-горе.

2. Метод на прогонването за тридиагонални системи линейни уравнения

Постановка на задачата. Търси се решението на системата линейни алгебрични уравнения

$$Ax = d\,, \tag{1}$$

където A е тридиагонална матрица, x – вектор на неизвестните, d - дясна част:

$$A = \begin{pmatrix} b_1 & c_1 & 0 & \dots & & & 0 \\ a_2 & b_2 & c_2 & 0 & \dots & & 0 \\ & & \dots & & & & \\ 0 & \dots & a_k & b_k & c_k & \dots & 0 \\ & & & & \dots & & \\ 0 & \dots & & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & \dots & & & o & a_n & b_n \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \\ x_k \\ \\ \dots \\ x_n \end{pmatrix}, \quad d = \begin{pmatrix} d_1 \\ d_2 \\ \\ d_k \\ \\ \dots \\ d_n \end{pmatrix} \neq 0 \quad . \tag{2}$$

В разгънат вид системата е:

$$
\left|\begin{array}{ccccccccc}
b_1x_1 & +c_1x_2 & & & & & & = & d_1 \\
a_2x_1 & +b_2x_2 & +c_2x_3 & & & & & = & d_2 \\
 & a_3x_2 & +b_3x_3 & +c_3x_4 & & & & = & d_3 \\
 & & & ..... & & & & & ..... \\
 & & & a_kx_{k-1} & +b_kx_k & +c_kx_{k+1} & & = & d_k \\
 & & & & ..... & & & & ..... \\
 & & & & a_{n-1}x_{n-2} & +b_{n-1}x_{n-1} & +c_{n-1}x_n & = & d_{n-1} \\
 & & & & & a_nx_{n-1} & +b_nx_n & = & d_n
\end{array}\right., \qquad (3)
$$

Тази система може да се реши по който и да е от съществуващите методи за системи линейни алгебрични уравнения: метод на Гаус, метод на Гаус-Жордан, проста итерация и др. Но тъй като тя е от специален вид, за нейното

решаване се прилагат специални ефективни методи, между които е методът на прогонването. Методът спада към групата на точните методи, т.е. теоретично, при работа с точни числа (без закръгляне), след краен брой аритметични операции се получава точното решение. В случая на реални системи обаче, като правило броят на уравненията n е много голям и неминуемо се работи със закръгляне на междинните резултати до определен знак след десетичната запетая. Това може да доведе до натрупване на грешка, независимо, че методът е точен.

3. Изчислителна схема на метода на прогонването

Целта е тридиагоналната система (3) да се преобразува в двудиагонална. Полагаме:

$$x_k = \alpha_k x_{k+1} + \beta_k \ , \quad k = \overline{1,n-1}, \tag{4}$$

където α_i, β_i са търсени прогонъчни коефициенти.

От първото уравнение на (3) имаме

$k=1$: $b_1 x_1 + c_1 x_2 = d_1$, откъдето сравнено с (4) намираме $x_1 = -\frac{c_1}{b_1} x_2 + \frac{d_1}{b_1}$, т.е. $\alpha_1 = -\frac{c_1}{b_1}$, $\beta_1 = \frac{d_1}{b_1}$.

По-нататък, ако сме получили формулата за някое *k-1*:

$x_{k-1} = \alpha_{k-1} x_k + \beta_{k-1}$, заместваме го в *k*-тото уравнение на (3)

$a_k x_{k-1} + b_k x_k + c_k x_{k+1} = d_k$ и получаваме

$a_k(\alpha_{k-1}x_k+\beta_{k-1})+b_kx_k+c_kx_{k+1}=d_k$, откъдето изразяваме x_k:

$x_k(a_k\alpha_{k-1}+b_k)+c_kx_{k+1}=d_k-a_k\beta_{k-1}$.

Ако множителят пред x_k не е нула, то:

$$x_k=-\frac{c_k}{b_k+a_k\alpha_{k-1}}x_{k+1}+\frac{d_k-a_k\beta_{k-1}}{b_k+a_k\alpha_{k-1}},$$ което е във вида (4), където

$$\alpha_k=-\frac{c_k}{b_k+a_k\alpha_{k-1}},\qquad \beta_k=\frac{d_k-a_k\beta_{k-1}}{b_k+a_k\alpha_{k-1}}.$$

Накрая за $k=n$ от това представяне при $x_{n+1}=0$ формално намираме: $x_n=\beta_n$.

Окончателно изчислителната схема на прогонването е:

Прав ход: (изчисляване на прогонъчните коефициенти) по формулите:

$$\alpha_1 = -\frac{c_1}{b_1},\quad \beta_1 = \frac{d_1}{b_1},\quad \alpha_k = -\frac{c_k}{b_k + a_k\alpha_{k-1}},\qquad \beta_k = \frac{d_k - a_k\beta_{k-1}}{b_k + a_k\alpha_{k-1}},\qquad k = \overline{2,n} \tag{5}$$

Обратен ход: $x_n = \beta_n \quad x_k = \alpha_k x_{k+1} + \beta_k \quad k = n-1, n-2, \ldots, 1.$ (6)

Определение. Прогонъчните формули (5) (методът на прогонването) ще наричаме устойчиви, ако

$$|\alpha_k| \le 1, \ k = \overline{1,n} \qquad (7)$$

Забележка. Очевидно е, че ако това не е изпълнено, то от формула (6) напр. при някаква малка грешка ε да речем за $\tilde{x}_n = \beta_n + \varepsilon$ ще получаваме:

$$x_k = \alpha_k x_{k+1} + \beta_k = \alpha_k \alpha_{k+1} x_{k+2} + \ldots + \beta_k = \ldots = \alpha_k \alpha_{k+1} \ldots \alpha_n \tilde{x}_n + \ldots .$$

Вижда се, че водещият член $\alpha_k \alpha_{k+1} \ldots \alpha_n$ ще се умножи по грешката и ако не е по-малък от 1, то грешката на резултата е неуправляема.

Сега ще намерим условията, които могат да гарантират избягването на тази ситуация.

4. Теорема: Достатъчно условие за устойчивост на метода на прогонването

Нека за коефициентите на системата е изпълнено:

$|b_k| \geq |a_k| + |c_k|,\ c_k \neq 0,\ k = \overline{1,n}$. (8)

Тогава съществува единствено решение на системата (1), което е и устойчиво, т.е. малки грешки в началните данни водят до малки грешки в резултата.

Доказателство. Става по метода на непълната математическа индукция. Очевидно условието за устойчивост (7) е вярно при $k=1$:

$$|\alpha_1| = \left|-\frac{c_1}{b_1}\right| \leq 1 .$$

Допускаме, че твърдението е вярно за *k-1*. За следващото *k* имаме:

Знаменател от (5): $|b_k + a_k \alpha_{k-1}| \geq |b_k| - |a_k|.|\alpha_{k-1}| \geq |b_k| - |a_k| \geq |c_k| > 0$. (9)

Тогава: $|\alpha_k| = \left| -\frac{c_k}{b_k + a_k \alpha_{k-1}} \right| \le \frac{|-c_k|}{|b_k + a_k \alpha_{k-1}|} \le \frac{|c_k|}{|c_k|} = 1$, или $|\alpha_k| \le 1$. Доказахме условието за устойчивост и при k. Следователно по индукция условието е изпълнено за всяко k. В частност тъй като знаменателят от (9) е ненулев, то винаги е възможно делението, а оттам - винаги съществува решение по формулите на прогонването при поставените условия на теоремата.

Забележка 1. Условието (8) може да се прецизира. Не е трудно да се съобрази, че всъщност то означава преобладаващ главен диагонал на матрицата *A*.

Забележка 2. Възможни са и случаи, когато методът на прогонването работи и без да е изпълнено условието за устойчивост, т.к. то е само достатъчно, но не и необходимо.

5. Преимущества на метода на прогонването. Пример

- Бърз: Лесно се пресмята от формули (5), (6), че са необходими $3n$ операции събиране/изваждане и $5n$ операции умножение/деление (знаменателят се изчислява само веднъж). Или всичко $8n$ аритметични операции - $O(n)$.
- Малко памет: Ясно е, че са достатъчни само едномерни масиви за $a_k, b_k, c_k, d_k, \alpha_k, \beta_k, x_k$. При повторно ползване е нужна памет само $4n$.
- Лесно се програмира.
- Условията за устойчивост са прости за проверка.

Реално методът се използва за решаване на системи с n=1000000 и повече уравнения, т.е. изисква 1 секунда машинно време. Дори и за по-големи n работи в реално време.

Пример. Проверете дали метода на прогонването е устойчив (не натрупва грешки) за системата:

$$\left|\begin{array}{rrrrcr} -4x_1 & +2x_2 & & & = & 1 \\ x_1 & +3x_2 & -x_3 & & = & 3 \\ & x_2 & -7x_3 & -2x_4 & = & -1 \\ & & -9x_3 & +10x_4 & = & 0 \end{array}\right. .$$

Решение: Имаме по редове: $|-4| \geq |2|$, $|3| \geq |1| + |-1|$, $|-7| \geq |1| + |-2|$ и $|10| \geq |-9|$.

Етапи за решаване на тридиагонална система алгебрични уравнения по метода на прогонването:

Провежда се на три етапа:

1. Проверка на условието за устойчивост (8),
2. Изчисляване на прогонъчните коефициенти по формули (5),
3. Изчисляване на неизвестните в обратен ред по формули (6).

За да решим горната тридиагонална система, съставяме и изпълняваме следния код на *Mathematica*:

Зареждаме векторите: a, b, c, d, α, β и някаква стойност на x (за да има списък, всеки вектор с дължина n):

```
a = {0, 1, 1, -9};
b = {-4, 3, -7, 10};
c = {2, -1, -2, 0};
d = {1, 3, -1, 0};
α = a;
β = a;
x = a;
n = Length[a];
```

$$\alpha_{[1]} = -\frac{c_{[1]}}{b_{[1]}};\ \beta_{[1]} = \frac{d_{[1]}}{b_{[1]}};$$

$$\text{For}\left[i = 2,\ i \le n,\ i{+}{+},\ z = a_{[i]} * \alpha_{[i-1]} + b_{[i]};\right.$$

$$\left.\alpha_{[i]} = -\frac{c_{[i]}}{z};\ \beta_{[i]} = \frac{d_{[i]} - a_{[i]} * \beta_{[i-1]}}{z}\right]$$

```
Print["    α= ", α, "    β= ", β]
(*  Край прав ход *)
```

$$\alpha = \left\{\frac{1}{2}, \frac{2}{7}, -\frac{14}{47}, 0\right\} \quad \beta = \left\{-\frac{1}{4}, \frac{13}{14}, \frac{27}{94}, \frac{243}{1192}\right\}$$

$$x_{[n]} = \beta_{[n]};$$

$$\text{For}[i = n - 1,\ i \ge 1,\ i{-}{-},$$

$$x_{[i]} = \alpha_{[i]} * x_{[i+1]} + \beta_{[i]}]$$

```
Print["    x= ", x]    (*  Край обратен ход  *)
```

Отг.: $x = \left\{\frac{147}{596}, \frac{148}{149}, \frac{135}{596}, \frac{243}{1192}\right\}$.

6. Итерационни методи: Метод на простата итерация

Постановка на задачата. Търси се приближеното решение на системата линейни алгебрични уравнения

$$Ax = b, \tag{10}$$

където A е матрица с реални коефициенти с размерност $n \times n$, x – вектор на неизвестните, b - дясна част:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ & & \cdots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \cdots \\ b_n \end{pmatrix} \neq \vec{0} \quad . \tag{11}$$

В разгънат вид системата е:

$$\left|\begin{array}{llll} a_{11}x_1 & +a_{12}x_2 & \cdots & +a_{1n}x_n = b_1 \\ a_{21}x_1 & +a_{22}x_2 & & +a_{2n}x_n = b_2 \\ & & \cdots & \\ & & & \\ a_{n1}x_1 & +a_{n2}x_2 & \cdots & +a_{nn}x_n = b_n \end{array}\right. . \qquad (12)$$

За да построим процеса на проста итерация, аналогично на итерационните методи за едно уравнение, изходният вид на системата трябва да е от тип $x = \varphi(x)$ с процес $x^{(k+1)} = \varphi(x^{(k)})$, $k = 0,1,...$

Най-често привеждането към такъв вид става така: първото уравнение се дели на a_{11} и всички останали членове се прехвърлят отдясно, второто уравнение се дели на a_{22} и т.н. Така от всяко уравнение се изразява неизвестното x_i от i-тия ред на системата.

Стигаме до вида:

$$x = Cx + d \tag{13}$$

и итерационен процес, наречен проста итерация или метод на Якоби:

$$x^{(k+1)} = Cx^{(k)} + d\,,\ k = 0,1,\ldots, \tag{14}$$

където $x^{(0)}$ е произволно начално приближение.

От (4) последователно изчисляваме редицата от приближения:

$$x^{(0)}, x^{(1)}, \ldots, x^{(k)}, \ldots \tag{15}$$

Както при итерационните методи за едно уравнение, формулата (14) определя явен процес, т.е. заместваме известна стойност $x^{(k)}$ в дясната част и изчисляваме следващото приближение $x^{(k+1)}$. Повтаряме процеса и получаваме последователно редицата (15).

7. Критерии за прекъсване на итерационния процес

за получаване на решение с желана точност ε (поне един от следните критерии):

1) $\left|x_i^{(k+1)} - x_i^{(k)}\right| < \varepsilon$, $i = 1, \ldots, n$ - по абсолютна грешка,

2) $\dfrac{\left|x_i^{(k+1)} - x_i^{(k)}\right|}{\left|x_i^{(k)}\right|} < \varepsilon$, $i = 1, \ldots, n$ - по относителна грешка.

Може и със заместване в уравненията.

8. Пример за метода на проста итерация

Пример. Да се реши по метода на проста итерация системата:

$$\left|\begin{array}{llll} 10x_1 & +x_2 & -3x_3 & =3 \\ x_1 & +5x_2 & -2x_3 & =5 \\ -x_1 & +x_2 & -5x_3 & =-14 \end{array}\right. . \qquad (16)$$

С директна проверка се вижда, че точното решение е: $x = (1,2,3)$.

Решение. 1) Привеждане към вид, удобен за итерация:

Делим първото уравнение на 10, второто на 5 и третото на (-5) и изразяваме неизвестните от главния диагонал:

$$\left|\begin{array}{llll} x_1 = & & -0.1x_2 & +0.3x_3 & +0.3 \\ x_2 = & -0.2x_1 & & +0.4x_3 & +1 \\ x_3 = & -0.2x_1 & +0.2x_2 & & +2.8 \end{array}\right. .$$

Тук матрицата C и дясната част в съответствие с (13)-(14) са:

$$C = \begin{pmatrix} 0 & -0.1 & 0.3 \\ -0.2 & 0 & 0.4 \\ -0.2 & 0.2 & 0 \end{pmatrix}, \quad d = \begin{pmatrix} 0.3 \\ 1 \\ 2.8 \end{pmatrix}. \tag{17}$$

2) Записваме формулите за пресмятане по проста итерация:

$$\begin{cases} x_1^{(k+1)} = & -0.1x_2^{(k)} & +0.3x_3^{(k)} & +0.3 \\ x_2^{(k+1)} = & -0.2x_1^{(k)} & +0.4x_3^{(k)} & +1 \\ x_3^{(k+1)} = & -0.2x_1^{(k)} & +0.2x_2^{(k)} & +2.8 \end{cases}. \tag{18}$$

3) Избираме начално приближение $x^{(0)} = (0,0,0)^T$. Заместваме го отдясно (при k=0) и получаваме 1-во приближение:

$x_1^{(1)} = (-0.1).0 + (0.3).0 + 0.3 = 0.3$, $x_2^{(1)} = (-0.2).0 + (0.4).0 + 1 = 1$,

$x_3^{(1)} = (-0.2).0 + (0.2).0 + 2.8 = 2.8$, т.е. $x^{(1)} = (0.3,1,2.8)^T$.

За второ приближение заместваме полученото $x^{(1)}$ вдясно на (18):

$x_1^{(2)} = (-0.1).1 + (0.3).2.8 + 0.3 = 1.04$, $x_2^{(2)} = (-0.2).(0.3) + (0.4).(2.8) + 1 = 2.06$,

$x_3^{(2)} = (-0.2).(0.3) + (0.2).1 + 2.8 = 2.94$, т.е. $x^{(2)} = (1.04, 2.06, 2.94)^T$.

На следващата итерация заместваме тези стойности отдясно и т.н. Получаваме следната таблица от приближения:

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\left\|x_1^{(k+1)} - x_1^{(k)}\right\|$	$\left\|x_2^{(k+1)} - x_2^{(k)}\right\|$	$\left\|x_3^{(k+1)} - x_3^{(k)}\right\|$
0	0	0	0			
1	0.3	1	2.8	0.3	1	2.8
2	1.04	2.06	2.94	0.74	1.06	0.14
3	0.976	1.968	3.004	0.064	0.092	0.064
4	1.0044	2.0064	2.9984	0.0284	0.0384	0.0056
5	0.99888	1.99848	3.0004	0.00552	0.00792	0.002
6	1.00027	2.00038	2.99992	0.00139	0.00190	0.00048
7	0.99994	1.99991	3.00002	0.000334	0.000108	0.000102

Отговор с точност $\varepsilon = 0.001$, k=7 (закръглен): $x \approx (1.000, 2.000, 3.000)^T$.

Същата задача с *Mathematica* може да се реши така:

А) Зареждане на приведената матрица и дясна част по формула (17):

```
(* Проста итерация *)
c = { {0., -0.1, 0.3}, {-0.2, 0., 0.4}, {-0.2, 0.2, 0.} };
d = {0.3, 1., 2.8}; MatrixForm[c]
MatrixForm[d]
```

$$\begin{pmatrix} 0. & -0.1 & 0.3 \\ -0.2 & 0. & 0.4 \\ -0.2 & 0.2 & 0. \end{pmatrix}$$

$$\begin{pmatrix} 0.3 \\ 1. \\ 2.8 \end{pmatrix}$$

Б) Зареждаме клетка за точността 0.001 и начално приближение $x^{(0)}$, (стандартно =0), след което в цикъл уточняваме по формула (14):

```
eps = 0.001;
x = {0., 0., 0.}
n = Length[x]; r = ∞; k = 0;
While[r ≥ eps,
 {k++; xnew = c.x + d; r = Max[Abs[xnew - x]]; x = xnew;
  Print[" итерация k=", k, "  x= ", x, "  текуща грешка= ", r]}]
```

```
{0., 0., 0.}
итерация k=1  x= {0.3, 1., 2.8}  текуща грешка= 2.8
итерация k=2  x= {1.04, 2.06, 2.94}  текуща грешка= 1.06
итерация k=3  x= {0.976, 1.968, 3.004}  текуща грешка= 0.092
итерация k=4  x= {1.0044, 2.0064, 2.9984}  текуща грешка= 0.0384
итерация k=5  x= {0.99888, 1.99848, 3.0004}  текуща грешка= 0.00792
итерация k=6  x= {1.00027, 2.00038, 2.99992}  текуща грешка= 0.001904
итерация k=7  x= {0.999938, 1.99991, 3.00002}  текуща грешка= 0.0004704
```

Отг.: $x_1 \approx 0.999938 \approx 1$, $x_2 \approx 0.99991 \approx 2$, $x_2 \approx 3.00002 \approx 3$ с точност 0.001.

9. Метод на Гаус-Зайдел. Пример

Постановка на задачата – както в метода на простата итерация.

Аналогично системата трябва да бъде предварително приведена във вида (13) - $x = Cx + d$.

Дотук приликата между методите завършва, тъй като схемата на итерациите при метода на Гаус-Зайдел е друга. Сега за изчисляване на всяко следващо приближение се използват най-новите, току-що получени приближения. По-точно формулите са

$$
\begin{array}{llllll}
x_1^{(k+1)} = & c_{11}x_1^{(k)} & +c_{12}x_2^{(k)} \ldots & +c_{1n}x_n^{(k)} & +d_1 \\
x_2^{(k+1)} = & c_{21}x_1^{(k+1)} & +c_{22}x_2^{(k)} \ldots & +c_{2n}x_n^{(k)} & +d_2 \\
x_3^{(k+1)} = & c_{31}x_1^{(k+1)} & +c_{32}x_2^{(k+1)} \ldots & +c_{3n}x_n^{(k)} & +d_3 \\
 & & & \ldots & \\
x_n^{(k+1)} = & c_{n2}x_1^{(k+1)} & +c_{n2}x_2^{(k+1)} \ldots & +c_{3n}x_n^{(k)} & +d_n
\end{array}
.
$$

Пример. Да се реши системата (16) по метода на Зайдел.

Решение. 1) е същото.

2) Записваме формулите по Зайдел:

$$\left|\begin{array}{llll} x_1^{(k+1)} = & & -0.1x_2^{(k)} & +0.3x_3^{(k)} & +0.3 \\ x_2^{(k+1)} = & -0.2x_1^{(k+1)} & & +0.4x_3^{(k)} & +1 \\ x_3^{(k+1)} = & -0.2x_1^{(k+1)} & +0.2x_2^{(k+1)} & & +2.8 \end{array}\right. . \qquad (19)$$

3) Избираме $x^{(0)} = (0,0,0)^T$. Заместваме го отдясно (при k=0) и получаваме 1-во приближение: $x_1^{(1)} = (-0.1).0 + (0.3).0 + 0.3 = 0.3$. За $x_2^{(1)}$ на мястото на x_1 заместваме вече получената стойноср 0.3 и намираме $x_2^{(1)} = (-0.2).(0.3) + (0.4).0 + 1 = 0.94$. Тези две нови стойности веднага заместваме в третото уравнение и получаваме $x_3^{(1)} = (-0.2).(0.3) + (0.2).(0.94) + 2.8 = 2.929$. Така $x^{(1)} = (0.3, 0.94, 2.928)^T$. Аналогично работим нататък по (19).

Резултатите нанасяме в следната таблица от приближения:

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\left\|x_1^{(k+1)} - x_1^{(k)}\right\|$	$\left\|x_1^{(k+1)} - x_1^{(k)}\right\|$	$\left\|x_1^{(k+1)} - x_1^{(k)}\right\|$
0	0	0	0			
1	0.3	0.94	2.928	0.3	0.94	2.928
2	1.0844	1.95432	2.97398	0.7844	1.01432	0.04598
3	0.99968	1.99902	2.99870	0.08472	0.04470	0.02472
4	1.00058	1.99936	2.99976	0.00090	0.00034	0.00106

Отговор за $\varepsilon = 0.001$, k=4, (със закръгляне): $x \approx (1.000, 1.999, 3.000)^T$.

Решени примери може да намерите на:
http://www.fmi-plovdiv.org/evlm/DBbg/numanmenu/programs_list.htm