REZOLVAREA NUMERICĂ A ECUAȚIILOR ŞI SISTEMELOR DE ECUAȚII ALGEBRICE NELINIARE

Forma generală a ecuației:

$$f(x)=0\ , \tag{1}$$

cu $f : I \subset R \rightarrow R$. În particular, f – polinom / adus la o formă polinomială, dar şi ecuațiile transcendente.

Rezolvarea ecuației (1) = găsirea zerourilor funcției f, adică a valorilor $x = c$ care satisfac (1).

Categorii de metode de rezolvare numerică a ecuațiilor algebrice neliniare:

a) metode de *separare* sau *localizare* a soluțiilor ecuației (1) – de izolare a unor subdomenii ale domeniului de definiție I, care să conțină câte unul din zerourile funcției f (a se vedea *şirul lui Rolle*);

b) metode de *determinare, cu o precizie a priori fixată, a unei soluții* care a fost izolată în prealabil, pornind de la o valoare aproximativă a acesteia;

c) metode de *determinare a tuturor soluțiilor* aplicabile, de regulă, în cazul în care f este un polinom.

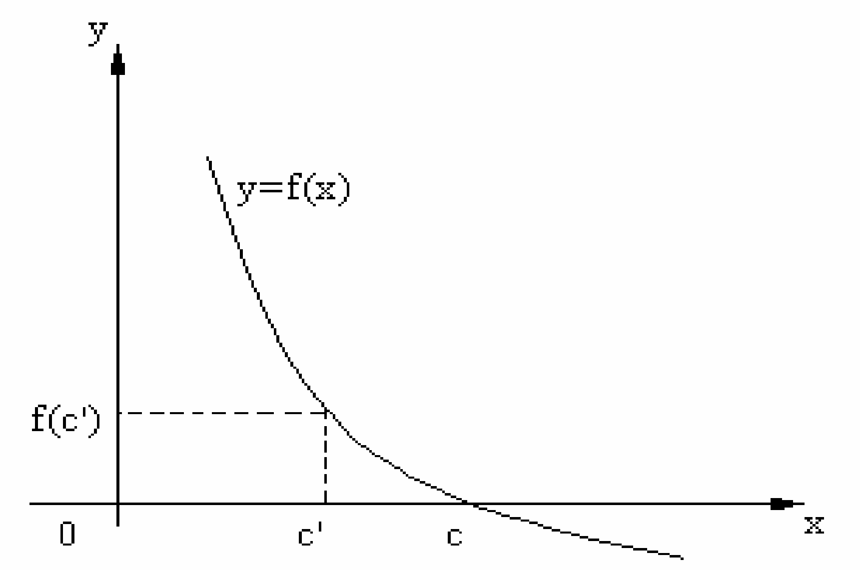
Soluție aproximativă

Se presupune că c este valoarea exactă a unei soluții a ecuației (1), iar c' o valoare aproximativă a acestei soluții. Soluția aproximativă se poate defini:

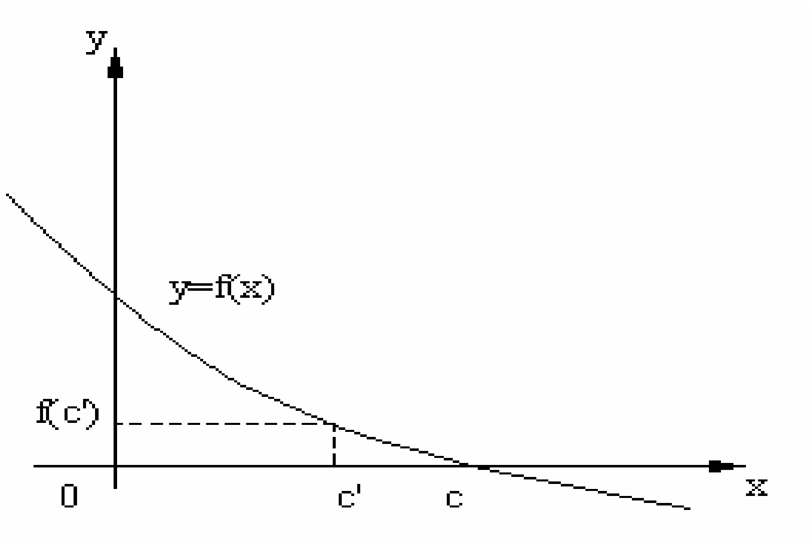
1) o valoare $x=c'$: $|c'-c| < \varepsilon_x$, cu $\varepsilon_x>0$ şi $f(c)=0$;

2) o valoare $x=c'$: $|f(c')| < \varepsilon_f$, cu $\varepsilon_f > 0$ şi $f(c)=0$.

Modul 1):



Modul 2):



1. Metode de calcul al unei soluții reale a unei ecuații algebrice neliniare

Soluția reală a ecuației (1) – separată în prealabil în intervalul [a, b]:

$$f(x) = 0,\ x \in [a, b]\ . \tag{1.1}$$

Două metode de partiționare a intervalului:

Metoda bisecției (înjumătățirii intervalului)

Este destinată rezolvării ecuației (1.1), pentru care s-a separat în prealabil o soluție în intervalul [a, b]:

$$f(a) \cdot f(b) < 0\ . \tag{1.2}$$

Se consideră f – continuă pe [a, b]; soluția va fi determinată cu erorile admise ε_x (pentru soluție) şi ε_f (pentru funcție).

Trăsătură caracteristică: pornind de la [a, b], la fiecare pas se restrânge domeniul în care se caută soluția prin înjumătățirea intervalului de la pasul anterior, până la atingerea preciziei dorite.

Avantaj: simplă.

Dezavantaj: slab convergentă.

Algoritmul metodei bisecției – etape:

I) Se inițializează limitele intervalului de căutare, “*r*” şi “*s*”, cu valorile limitelor intervalului în care s-a separat soluția:

$$r^0 = a, \quad s^0 = b \tag{1.3}$$

(indicele superior – iterația curentă).

II) La pasul de calcul *k*, *k* = 1, 2, 3, …, se determină noua valoare a soluției:

$$x^k = \frac{r^{k-1} + s^{k-1}}{2} \quad . \tag{1.4}$$

III) La acelaşi pas *k* se calculează $f(x^k)$ şi $f(r^{k-1})$ $\Rightarrow$ noile limite ale intervalului de căutare:

$$\text{dacă } f(x^k) \cdot f(r^{k-1}) < 0 \Rightarrow r^k = r^{k-1} \text{ şi } s^k = x^k; \tag{1.5}$$

$$\text{dacă } f(x^k) \cdot f(r^{k-1}) > 0 \Rightarrow r^k = x^k \text{ şi } s^k = s^{k-1}; \tag{1.6}$$

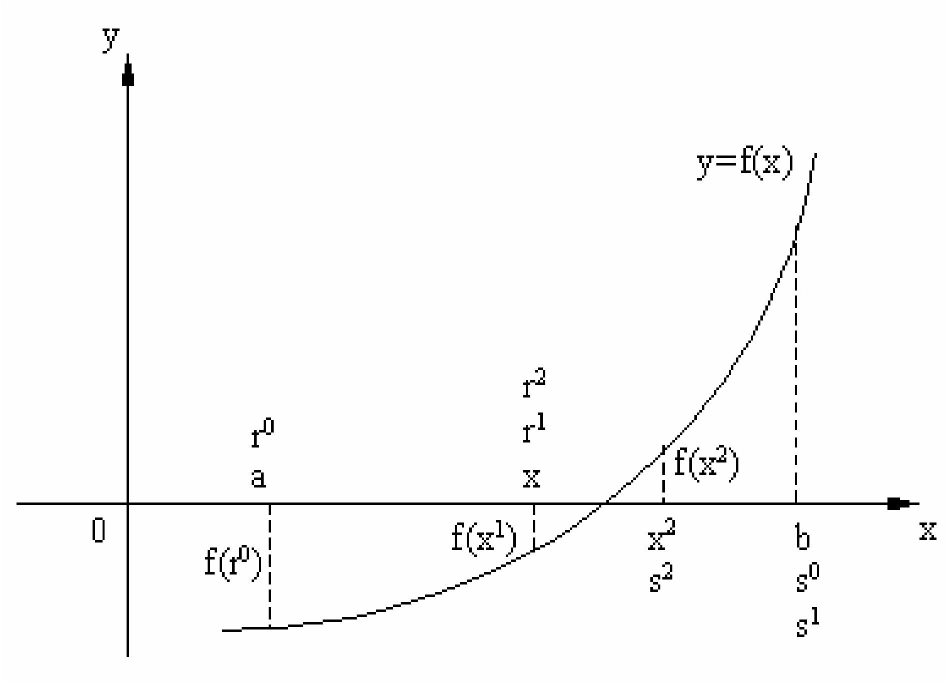
$$\text{dacă } f(x^k) \cdot f(r^{k-1}) = 0 \Rightarrow \text{calcul terminat şi } c = x^k; \tag{1.7}$$

IV) Procesul de calcul se consideră terminat când sunt îndeplinite condițiile (1.8) şi / sau (1.9):

$$|s^k - r^k| \le \varepsilon_x; \tag{1.8}$$

$$|f(x^k)| \le \varepsilon_f. \tag{1.9}$$

Interpretarea geometrică a metodei bisecției:



Exemplu: Se consideră ecuația:

f(x) = 0 ,   f(x) = 2·tg$x$ – 10·x + 3 ,

pentru care s-a separat o soluție în intervalul [–1, 1]. Să se determine soluția ecuației utilizând metoda bisecției, erorile admise fiind $\varepsilon_x = 10^{-3}$ şi $\varepsilon_f = 10^{-2}$.

Soluție: Se parcurg etapele metodei bisecției:

I) Inițializări:

$r^0 = -1$, $s^0 = 1$, $|r^0 - s^0| = 2$.

Iterația k = 1:

II) $x^1 = \dfrac{r^0 + s^0}{2} = 0$;

III) $f(x^1) = f(0) = 3$, $f(r^0) = f(-1) = 9.885$,

$f(x^1) \cdot f(r^0) > 0 \quad \Rightarrow \quad r^1 = x^1 = 0$, $s^1 = s^0 = 1$.

Se verifică dacă sunt îndeplinite condițiile de terminare (1.8) şi (1.9):

$|r^1 - s^1| = 1 > \varepsilon_x$ şi $|f(x^1)| = 3 > \varepsilon_f$.

Nu sunt îndeplinite $\Rightarrow$ algoritmul se continuă cu:

Iterația k = 2:

II) $x^2 = \dfrac{r^1 + s^1}{2} = 0.5$;

III) $f(x^2) = f(0.5) = -0.9074$, $f(r^1) = f(0) = 3$,

$f(x^2) \cdot f(r^1) < 0 \Rightarrow r^2 = r^1 = 0$, $s^2 = x^2 = 0.5$.

$|r^2 - s^2| = 0.5 > \varepsilon_x$ şi $|f(x^2)| = 0.9074 > \varepsilon_f \Rightarrow$

Se trece la iterația următoare:

Iterația k = 3:

II) $x^3 = \dfrac{r^2 + s^2}{2} = 0.25$;

III) $f(x^3) = f(0.25) = 1.011$, $f(r^2) = f(0) = 3$,

$f(x^3) \cdot f(r^2) > 0 \Rightarrow r^3 = x^3 = 0.25$, $s^3 = s^2 = 0.5$.

$|r^3 - s^3| = 0.25 > \varepsilon_x$ şi $|f(x^3)| = 1.011 > \varepsilon_f \Rightarrow$

Algoritmul se continuă.

Iterația k = 4:

II) $x^4 = \dfrac{r^3 + s^3}{2} = 0.375$;

III) $f(x^4) = f(0.375) = 0.037$, $f(r^3) = f(0.25) = 1.011$,

$f(x^4) \cdot f(r^3) > 0 \Rightarrow r^4 = x^4 = 0.375$, $s^4 = s^3 = 0.5$.

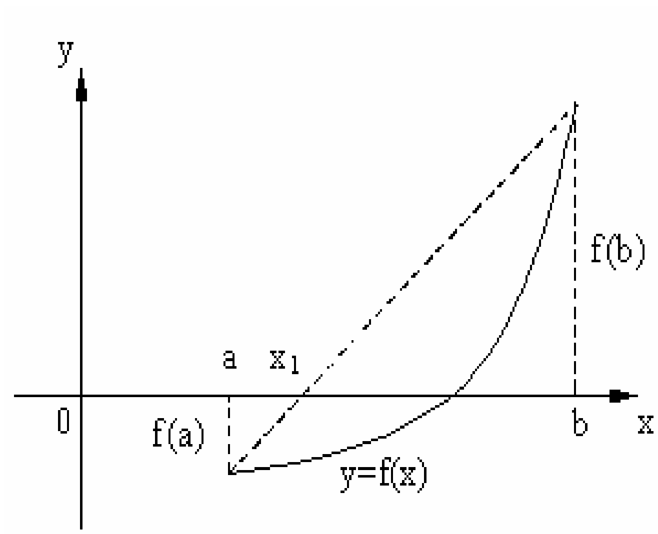
$|r^4 - s^4| = 0.125 > \varepsilon_x$ şi $|f(x^4)| = 0.375 > \varepsilon_f$.

Erorile au scăzut, însă nu suficient de mult pentru ca cele două condiții de terminare să fie îndeplinite $\Rightarrow$ alte iterații.

Metoda falsei poziții (metoda coardei, metoda secantei, metoda împărțirii intervalului în părți proporționale)

Avantaj: mai rapid convergentă.

Trăsătură caracteristică: pornind de la [a, b], la fiecare pas se restrânge domeniul de căutare a soluției, prin împărțirea intervalului de la pasul anterior în raportul valorilor funcției la capetele intervalului.



Interpretarea geometrică:

Coarda:

$$\frac{x-a}{b-a}=\frac{\mathrm{f}(x)-\mathrm{f}(a)}{\mathrm{f}(b)-\mathrm{f}(a)}\,, \tag{1.10}$$

$\Rightarrow$ abscisa punctului de intersecție cu Ox:

$$x_1=\frac{a\cdot \mathrm{f}(b)-b\cdot \mathrm{f}(a)}{\mathrm{f}(b)-\mathrm{f}(a)}\,. \tag{1.11}$$

Algoritmul metodei falsei poziții – etape:

I) Se inițializează limitele intervalului curent de căutare, “r” şi “s”:

$$r^0=a\,,\ s^0=b \tag{1.12}$$

şi se calculează f(r^0) şi f(s^0).

II) La un pas oarecare k, k = 1, 2, 3, …, al procesului iterativ de calcul, se calculează noua valoare a soluției:

$$x^k=\frac{r^{k-1}\cdot \mathrm{f}(s^{k-1})-s^{k-1}\cdot \mathrm{f}(r^{k-1})}{\mathrm{f}(s^{k-1})-\mathrm{f}(r^{k-1})}\,. \tag{1.13}$$

III) La acelaşi pas k se calculează f(x^k), rezultând noile limite ale intervalului de căutare, r^k şi s^k, conform (1.5) … (1.7), împreună cu f(r^k) şi f(s^k).

IV) Calculul se termină când sunt îndeplinite condițiile (1.8) şi / sau (1.9).

Exemplu: Se consideră ecuația de la exemplul anterior; să se rezolve utilizând metoda falsei poziții.

Soluție: Se aplică metoda falsei poziții:

I) Inițializări:

$r^0 = -1$, $s^0 = 1$

şi se calculează valorile funcției f în r^0 şi s^0:

$f(r^0) = f(-1) = 9.885$, $f(s^0) = f(1) = -3.885$.

Pentru k = 1, 2, 3, …, se repetă etapele II) … IV), până când condițiile etapei IV) sunt îndeplinite:

Iterația k = 1:

II) Se determină x^1 cu (1.13):

$$x^1 = \frac{r^0 \cdot f(s^0) - s^0 \cdot f(r^0)}{f(s^0) - f(r^0)} = \frac{-(-3.885) - 9.885}{-3.885 - 9.885} = 0.4357 \; .$$

III) Se determină valoarea funcției f în x^1:

$f(x^1) = f(0.4375) = -0.426$.

$f(x^1) \cdot f(r^0) < 0 \;\Rightarrow\;$ din (1.5): $r^1 = r^0 = -1$, $s^1 = x^1 = 0.4357$

şi valorile corespunzătoare ale funcției f:

$f(r^1) = f(-1) = 9.855$, $f(s^1) = f(0.4357) = -0.426$.

Se verifică condițiile de terminare a algoritmului:

$|r^1 - s^1| = 1.4357 > \varepsilon_x$, $|f(x^1)| = 0.426 > \varepsilon_f$.

(1.8) şi (1.9) nu sunt satisfăcute $\Rightarrow$ iterația următoare:

Iterația k = 2:

II) Se determină x^2:

$$x^2 = \frac{r^1 \cdot f(s^1) - s^1 \cdot f(r^1)}{f(s^1) - f(r^1)} = \frac{-(-0.426) - 0.4357 \cdot 9.885}{-0.426 - 9.885} = 0.3764$$

III) Se determină $f(x^2)$:

$f(x^2) = f(0.3764) = 0.265$.

$f(x^2) \cdot f(r^1) > 0 \Rightarrow$ din (1.6): $r^2 = x^2 = 0.3764$, $s^2 = s^1 = 0.4357$.

$f(r^2) = f(0.3764) = 0.265$, $f(s^2) = f(0.4357) = -0.426$.

Se verifică din nou condițiile de terminare a calculelor:

$|r^2 - s^2| = 0.0587 > \varepsilon_x$, $|f(x^2)| = 0.265 > \varepsilon_f$.

(1.8) şi (1.9) nu sunt satisfăcute $\Rightarrow$ iterația următoare:

Erorile au scăzut semnificativ, scăderea lor fiind mai rapidă decât în cazul metodei bisecției, însă încă nu s-a ajuns la îndeplinirea condițiilor de terminare a calculelor $\Rightarrow$ algoritmul se continuă.

2. Generalități privind soluționarea numerică a sistemelor de ecuații algebrice neliniare

Forna implicită a unnui sistem de ecuații algebrice neliniare de ordinul "*n*" – întodeauna posibilă:

$$\begin{cases} \mathrm{f}_1(x_1, x_2, \ldots, x_n) = 0 \\ \mathrm{f}_2(x_1, x_2, \ldots, x_n) = 0 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \mathrm{f}_n(x_1, x_2, \ldots, x_n) = 0 \end{cases}, \tag{2.1}$$

f_1, f_2, …, f_n, de variabile x_1, x_2, …, x_n – continue.

Notații matriceale: $$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \underline{\mathrm{f}} = \begin{bmatrix} \mathrm{f}_1 \\ \mathrm{f}_2 \\ \vdots \\ \mathrm{f}_n \end{bmatrix}, \tag{2.2}$$

$\Rightarrow$ forma compactă a sistemului:

$$\underline{\mathrm{f}}(\underline{x}) = \underline{0}\,, \ \ \underline{\mathrm{f}} : D \subset \boldsymbol{R}^n \to \boldsymbol{R}^n\,. \tag{2.3}$$

Forme intermediare:

$$\mathrm{f}_i(\underline{x}) = 0\,, \ \ i = 1 \ldots n\,. \tag{2.4}$$

Determinarea unei soluții a sistemului (2.1) =

găsirea unui set de valori: $$\underline{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}, \tag{2.5}$$

care satisfac (2.1) $\Leftrightarrow$ $\underline{\mathrm{f}}(\underline{c}) = \underline{0}$.

Categorii de metode numerice:

I. *metode de* ***separare*** a unei / unor soluții de interes;

II. *metode de* ***determinare***, cu o precizie fixată a priori, a unei soluții separate în prealabil.

În categoria II.:

a) metode bazate pe **exprimarea explicită echivalentă** a ecuațiilor sistemului (2.1) – ***metode de aproximații succesive***;

b) metode care utilizează derivatele parțiale ale funcțiilor f_i – ***metode de tip Newton***;

c) metode de ***descreştere*** sau de ***coborâre*** sau de ***gradient***.

Doar a) şi b).

3. Metode bazate pe exprimarea explicită echivalentă a ecuațiilor sistemului

Se cere să se determine o soluție $\underline{c}$ a sistemului (2.1), separată în prealabil în domeniul $D = \prod_{i=1}^{n} [a_i, b_i] \subset \boldsymbol{R}^n$, cu erorile maxim admise ε_x (pentru valorile variabilelor) şi ε_f (pentru valorile funcțiilor).

Trăsătură caracteristică: înlocuirea exprimărilor implicite (2.1) ale ecuațiilor sistemului cu exprimările explicite echivalente:

$$\begin{cases} x_1 = g_1(x_1, x_2, \ldots, x_n) \\ x_2 = g_2(x_1, x_2, \ldots, x_n) \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_n = g_n(x_1, x_2, \ldots, x_n) \end{cases}, \text{ cu } g_i,\ i = 1 \ldots n - \text{continue.} \qquad (3.1)$$

Notație: $\underline{g} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix}$, (3.2)

$\Rightarrow$ exprimarea matriceală: $\underline{x} = \underline{g}(\underline{x})$, $\underline{x} \in D \subset \boldsymbol{R}^n$, (3.3)

cu forma intermediară: $x_i = g_i(\underline{x})$, $i = 1 \ldots n$. (3.4)

Exprimările explicite sunt întotdeauna posibile şi, în plus, uneori sunt posibile mai multe variante !

Algoritmul metodei aproximațiilor succesive în versiunea Jacobi

I) Se inițializează $\underline{x}$ cu $\underline{x}^0 \in D$ (indicele superior – iterația curentă):

$$\underline{x}^0 = \begin{bmatrix} x_1^0 \\ x_2^0 \\ \vdots \\ x_n^0 \end{bmatrix}. \qquad (3.5)$$

II) La un pas oarecare k, k = 1, 2, 3, …, al procesului iterativ de calcul, se determină noile valori ale variabilelor:

$$x_i^k = \mathrm{g}_i(x_1^{k-1}, x_2^{k-1}, \ldots, x_n^{k-1}),\ i = \overline{1,n}\ . \tag{3.6}$$

III) Calculul este terminat atunci când sunt îndeplinite condițiile (3.7) şi / sau (3.8):

$$\left|x_i^k - x_i^{k-1}\right| \le \varepsilon_x,\ i = \overline{1,n}\ , \tag{3.7}$$

$$\left|\mathrm{f}_i(\underline{x}^k)\right| \le \varepsilon_f,\ i = \overline{1,n}\ . \tag{3.8}$$

Condițiile de convergență suficiente:

$$\left|\frac{\partial \mathrm{g}_i(\underline{x})}{\partial x_j}\right| < 1,\ i, j = \overline{1,n}\ . \tag{3.9}$$

Metoda aproximațiilor succesive în versiunea Gauss-Seidel

Diferență: relația (3.6), care devine:

$$x_i^k = \mathrm{g}_i(x_1^k, x_2^k, \ldots, x_{i-1}^k, x_i^{k-1}, \ldots, x_n^{k-1}),\ i = \overline{1,n}\ , \tag{3.10}$$

$\Leftrightarrow$ apar valorile "noi" ale variabilelor care au fost recalculate deja la iterația k.

Exemplu: Să se rezolve sistemul algebric neliniar:

$$\begin{cases} 2x_1^2 - x_2 x_3 - 5x_1 + 1 = 0 \\ x_2^2 - 2x_1 - \ln x_3 = 0 \\ x_3^2 - x_1 x_2 - 2x_3 - 8 = 0 \end{cases}$$

cu metoda Gauss-Seidel, cu erorile maxime admise $\varepsilon_x = 0{,}001$ şi $\varepsilon_f = 0{,}1$, cunoscând că s-a separat o soluție în domeniul $D = [0; 10] \times [0; 10] \times [1; 10]$.

Soluție: Rescrierea sistemului într-o formă cu exprimarea explicită a variabilelor, de forma (3.1):

$$\begin{cases} x_1 = \sqrt{0.5 \cdot (x_2 x_3 + 5x_1 - 1)} \\ x_2 = \sqrt{2x_1 + \ln x_3} \\ x_3 = \sqrt{x_1 x_2 + 2x_3 + 8} \end{cases} .$$

Iterația k = 0:

I) Se inițializează $\underline{x}$, de exemplu cu: $\underline{x}^0 = \begin{bmatrix} 10 \\ 10 \\ 10 \end{bmatrix}$.

Valorile funcțiilor f_1, f_2 şi f_3 pentru valorile inițiale ale variabilelor:

$f_1(\underline{x}^0) = 2 \cdot 10^2 - 10 \cdot 10 - 5 \cdot 10 + 1 = 51$,

$f_2(\underline{x}^0) = 10^2 - 2 \cdot 10 - \ln 10 = 77.7$,

$f_3(\underline{x}^0) = 10^2 - 10 \cdot 10 - 2 \cdot 10 - 8 = -28$.

Observație: Inițializarea se poate face şi cu alte valori şi se pot urmări efectele asupra evoluției convergenței procesului de calcul în funcție de aceste valori inițiale.

Iterația k = 1:

II) Se utilizează (3.10):

$$x_1^1 = \sqrt{0.5 \cdot (x_2^0 \cdot x_3^0 + 5 \cdot x_1^0 - 1)} = \sqrt{0.5 \cdot (10 \cdot 10 + 5 \cdot 10 - 1)} = 8.631$$

$$x_2^1 = \sqrt{2 \cdot x_1^1 + \ln x_3^0} = \sqrt{2 \cdot 8.631 + \ln 10} = 4.423 \ ,$$

$$x_3^1 = \sqrt{x_1^1 \cdot x_2^1 + 2 \cdot x_3^0 + 8} = \sqrt{8.631 \cdot 4.423 + 2 \cdot 10 + 8} = 8.135 \ .$$

Se calculează erorile:

$|x_1^1 - x_1^0| = |8.631 - 10| = 1.369 > \varepsilon_x$,

$|x_2^1 - x_2^0| = |4.423 - 10| = 5.577 > \varepsilon_x$,

$|x_3^1 - x_3^0| = |8.135 - 10| = 1.865 > \varepsilon_x$.

$|f_1(\underline{x}^1)| = 70.86 > \varepsilon_f$,

$|f_2(\underline{x}^1)| = 0.206 > \varepsilon_x$,

$|f_3(\underline{x}^1)| = 3.73 > \varepsilon_x$.

Condițiile de terminare a calculelor nu sunt îndeplinite $\Rightarrow$ necesară continuarea algoritmului cu iterația următoare:

Iterația k = 2:

II) Se calculează $\underline{x}^2$ utilizând (3.10):

$x_1^2 = \sqrt{0.5 \cdot (x_2^1 \cdot x_3^1 + 5 \cdot x_1^1 - 1)} = \sqrt{0.5 \cdot (4.423 \cdot 8.135 + 5 \cdot 8.631 - 1)} = 6.251$,

$x_2^2 = \sqrt{2 \cdot x_1^2 + \ln x_3^1} = \sqrt{2 \cdot 6.251 + \ln 8.135} = 3.821$,

$x_3^2 = \sqrt{x_1^2 \cdot x_2^2 + 2 \cdot x_3^1 + 8} = \sqrt{6.251 \cdot 3.821 + 2 \cdot 8.135 + 8} = 6.939$.

Se determină erorile:

$|x_1^2 - x_1^1| = |6.251 - 8.631| = 2.38 > \varepsilon_x$,

$|x_2^2 - x_2^1| = |3.821 - 4.423| = 0.602 > \varepsilon_x$,

$|x_3^2 - x_3^1| = |6.939 - 8.135| = 1.196 > \varepsilon_x$.

$|f_1(\underline{x}^2)| = 21.4 > \varepsilon_f$,

$|f_2(\underline{x}^2)| = 0.16 > \varepsilon_x$,

$|f_3(\underline{x}^2)| = 2.39 > \varepsilon_x$.

Erorile au scăzut, dar nu sunt încă îndeplinite condițiile de terminare a procesului de calcul $\Rightarrow$ etapele II) şi III) ale algoritmului se repetă pentru $k = 3, 4, \ldots, 12$.

Iterația k = 13:

II) Se calculează $\underline{x}^{13}$:

$x_1^{13} = \sqrt{0.5 \cdot (x_2^{12} \cdot x_3^{12} + 5 \cdot x_1^{12} - 1)} = \sqrt{0.5 \cdot (3.292 \cdot 5.89 + 5 \cdot 4.531 - 1)} = 4.53$,

$x_2^{13} = \sqrt{2 \cdot x_1^{13} + \ln x_3^{12}} = \sqrt{2 \cdot 4.53 + \ln 5.89} = 3.291$,

$x_3^{13} = \sqrt{x_1^{13} \cdot x_2^{13} + 2 \cdot x_3^{12} + 8} = \sqrt{4.53 \cdot 3.291 + 2 \cdot 5.89 + 8} = 5.89$.

Se determină erorile:

$|x_1^{13} - x_1^{12}| = |4.53 - 4.531| = 0.001 \le \varepsilon_x$,

$|x_2^{13} - x_2^{12}| = |3.291 - 3.292| = 0.001 \le \varepsilon_x$,

$|x_3^{13} - x_3^{12}| = |5.89 - 5.89| = 0 \le \varepsilon_x$.

$|f_1(\underline{x}^2)| = 0.008 \le \varepsilon_f$,

$|f_2(\underline{x}^2)| = 0.003 \le \varepsilon_f$,

$|f_3(\underline{x}^2)| = 0.004 \le \varepsilon_f$.

Erorile calculate sunt mai mici sau cel mult egale cu erorile maxim admisibile $\Rightarrow$ algoritmul se opreşte.

$\Rightarrow$ soluţia aproximativă: $\underline{x} = \begin{bmatrix} 4.43 \\ 3.291 \\ 5.89 \end{bmatrix}$.

4. Metode de tip Newton

Versiunea clasică a ***metodei lui Newton*** utilizează explicit derivatele parţiale de ordinul I ale funcţiilor $f_i(\underline{x})$, $i = 1 \ldots n$.

Se presupune că s-a ajuns la pasul k al procesului iterativ de calcul, ultima valoare aproximativă a soluţiei fiind $\underline{x}^{k-1}$. Se doreşte determinarea unei corecţii $\underline{h}^{k-1}$ care, adăugată la $\underline{x}^{k-1}$, să conducă la soluţia exactă $\underline{c}$:

$$\underline{c} = \underline{x}^{k-1} + \underline{h}^{k-1}. \tag{4.1}$$

Dezvoltând în serie Taylor funcțiile $f_i(\underline{x})$, $i = 1 \ldots n$, în vecinătatea lui $\underline{x}^{k-1} \Rightarrow$

$f_i(\underline{c}) = 0 \;\Leftrightarrow\; f_i(\underline{x}^{k-1} + \underline{h}^{k-1}) = 0 \;\Leftrightarrow$

$$f_i(\underline{x}^{k-1}) + \frac{\partial f_i(\underline{x}^{k-1})}{\partial x_1} h_1^{k-1} + \frac{\partial f_i(\underline{x}^{k-1})}{\partial x_2} h_2^{k-1} + \cdots + \frac{\partial f_i(\underline{x}^{k-1})}{\partial x_n} h_n^{k-1} + \cdots = 0,\ i = \overline{1,n}. \tag{4.2}$$

Dacă din această dezvoltare se rețin doar termenii care conțin derivatele de ordinul I (restul termenilor se neglijează) $\Rightarrow$ se poate aproxima acea valoare a lui $\underline{h}^{k-1}$ care nu va mai conduce la soluția exactă $\underline{c}$, ci la noua valoare aproximativă $\underline{x}^k$ a soluției (evident, mai bună decât $\underline{x}^{k-1}$, în cazul convergenței).

$\Rightarrow$ relațiile (4.2) conduc la sistemul liniar de ordinul n în necunoscutele $h_1^{k-1}, h_2^{k-1}, \ldots, h_n^{k-1}$:

$$\begin{cases} \dfrac{\partial f_1}{\partial x_1} h_1^{k-1} + \dfrac{\partial f_1}{\partial x_2} h_2^{k-1} + \cdots + \dfrac{\partial f_1}{\partial x_n} h_n^{k-1} = -f_1(\underline{x}^{k-1}) \\ \dfrac{\partial f_2}{\partial x_1} h_1^{k-1} + \dfrac{\partial f_2}{\partial x_2} h_2^{k-1} + \cdots + \dfrac{\partial f_2}{\partial x_n} h_n^{k-1} = -f_2(\underline{x}^{k-1}) \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \dfrac{\partial f_n}{\partial x_1} h_1^{k-1} + \dfrac{\partial f_n}{\partial x_2} h_2^{k-1} + \cdots + \dfrac{\partial f_n}{\partial x_n} h_n^{k-1} = -f_n(\underline{x}^{k-1}) \end{cases}, \tag{4.3}$$

unde toate derivatele sunt calculate în $\underline{x}^{k-1}$.

Matricea Jacobian:
$$\underline{J}^{k-1} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}, \qquad (4.4)$$

$\Rightarrow$ sistemul (4.3) se poate rescrie sub formă restrânsă:

$$\underline{J}^{k-1} \cdot \underline{h}^{k-1} = -\underline{f}^{k-1} \ . \qquad (4.5)$$

Algoritmul versiunii clasice a metodei lui Newton:

I) Se inițializează $\underline{x}$ cu $\underline{x}^0 \in$ D (indicele superior – iterația curentă).

II) La un pas oarecare k, k = 1, 2, …, al procesului iterativ de calcul, se calculează elementele vectorului $\underline{f}^{k-1}$ şi matricea $\underline{J}^{k-1}$ pentru $\underline{x} = \underline{x}^{k-1}$.

III) La acelaşi pas se rezolvă sistemul (4.5) $\Rightarrow$ noile valori ale variabilelor:

$$\underline{x}^k = \underline{x}^{k-1} + \underline{h}^{k-1} \ . \qquad (4.6)$$

Calculul este terminat când sunt îndeplinite condițiile (4.7) şi / sau (4.8):

$$\left|h_i^{k-1}\right| \le \varepsilon_x,\ i=\overline{1,n}\ , \tag{4.7}$$

$$\left|\mathrm{f}_i(\underline{x}^k)\right| \le \varepsilon_f,\ i=\overline{1,n}\ . \tag{4.8}$$

Exemplu: Să se rezolve sistemul de ecuații din exemplul anterior utilizând metoda clasică a lui Newton, cu erorile maxim admise $\varepsilon_x = 0{,}01$ şi $\varepsilon_f = 0{,}1$, cunoscând că s-a separat o soluție în domeniul $D = [0;\ 10] \times [0;\ 10] \times [1;\ 10]$.

Soluție: Se parcurg etapele algoritmului:

I) Se face inițializarea: $\underline{x}^0 = \begin{bmatrix} 10 \\ 10 \\ 10 \end{bmatrix}$.

Iterația k = 1:

II) Se calculează elementele vectorului $\underline{\mathrm{f}}^0 = \underline{\mathrm{f}}(\underline{x}^0)$:

$$\underline{\mathrm{f}}^0 = \begin{bmatrix} 2(x_1^0)^2 - x_2^0 x_3^0 - 5x_1^0 + 1 \\ (x_2^0)^2 - 2x_1^0 - \ln x_3^0 \\ (x_3^0)^2 - x_1^0 x_2^0 - 2x_3^0 - 8 \end{bmatrix} = \begin{bmatrix} 51 \\ 77.697 \\ -28 \end{bmatrix}$$

şi elementele matricei Jacobian:

$$\underline{J}^0 = \begin{bmatrix} 4x_1^0 - 5 & -x_3^0 & -x_2^0 \\ -2 & 2x_2^0 & -\dfrac{1}{x_3^0} \\ -x_2^0 & -x_1^0 & 2x_3^0 - 2 \end{bmatrix} = \begin{bmatrix} 35 & -10 & -10 \\ -2 & 20 & -0.1 \\ -10 & -10 & 18 \end{bmatrix}.$$

$\Rightarrow$ sistemul (4.5) devine: $\begin{cases} 35h_1^0 - 10h_2^0 - 10h_3^0 = -51 \\ -2h_1^0 + 20h_2^0 - 0{,}1h_3^0 = -77.697 \\ -10h_1^0 - 10h_2^0 + 18h_3^0 = 28 \end{cases}$.

III) Se rezolvă sistemul $\Rightarrow$ $\underline{h}^0 = \begin{bmatrix} -3.445 \\ -4.243 \\ -2.716 \end{bmatrix}$.

$\Rightarrow$ noile valori ale lui $\underline{x}$: $\underline{x}^1 = \underline{x}^0 + \underline{h}^0 = \begin{bmatrix} 6.555 \\ 5.757 \\ 7.284 \end{bmatrix}$.

Însă $|h_1^0| = |3.445| > \varepsilon_x$, $|h_2^0| > \varepsilon_x$, $|h_3^0| > \varepsilon_x$ $\Rightarrow$ nu sunt îndeplinite condițiile de terminare a calculelor $\Rightarrow$ algoritmul se continuă cu iterația următoare:

Iterația $k = 2$:

II) Se calculează elementele vectorului $\underline{\mathrm{f}}^1 = \underline{\mathrm{f}}(\underline{x}^1)$ şi ale matricei Jacobian:

$$\underline{\mathrm{f}}^1 = \begin{bmatrix} 2(x_1^1)^2 - x_2^1 x_3^1 - 5x_1^1 + 1 \\ (x_2^1)^2 - 2x_1^1 - \ln x_3^1 \\ (x_3^1)^2 - x_1^1 x_2^1 - 2x_3^1 - 8 \end{bmatrix} = \begin{bmatrix} 12.227 \\ 18.047 \\ -7.248 \end{bmatrix}$$

$$\underline{J}^1 = \begin{bmatrix} 4x_1^1 - 5 & -x_3^1 & -x_2^1 \\ -2 & 2x_2^1 & -\dfrac{1}{x_3^1} \\ -x_2^1 & -x_1^1 & 2x_3^1 - 2 \end{bmatrix} = \begin{bmatrix} 21.22 & -7.284 & -5.757 \\ -2 & 11.514 & -0.137 \\ -5.757 & -6.555 & 12.568 \end{bmatrix}.$$

$\Rightarrow$ sistemul (4.5):

$$\begin{cases} 21.22h_1^1 - 7.284h_2^1 - 5.757h_3^1 = -12.227 \\ -2h_1^1 + 11.514h_2^1 - 0.137h_3^1 = -18.047 \\ -5.757h_1^1 - 6.555h_2^1 + 12.568h_3^1 = 7.248 \end{cases}.$$

III) Rezolvarea sistemului $\Rightarrow$ $\underline{h}^1 = \begin{bmatrix} -1.498 \\ -1.84 \\ -1.069 \end{bmatrix}$.

$\Rightarrow$ noile valori ale lui $\underline{x}$: $\underline{x}^2 = \underline{x}^1 + \underline{h}^1 = \begin{bmatrix} 5.057 \\ 3.917 \\ 6.215 \end{bmatrix}$.

Însă, din nou $|h_1^1| = |1.498| > \varepsilon_x$, $|h_2^1| > \varepsilon_x$, $|h_3^1| > \varepsilon_x$ $\Rightarrow$ calculele se continuă cu iterația următoare.

Alte variante ale metodei lui Newton: eliminarea calculului derivatei.