REZOLVAREA APROXIMATIVĂ A ECUAȚIILOR ALGEBRICE ȘI TRANSCENDENTE – PARTEA 2

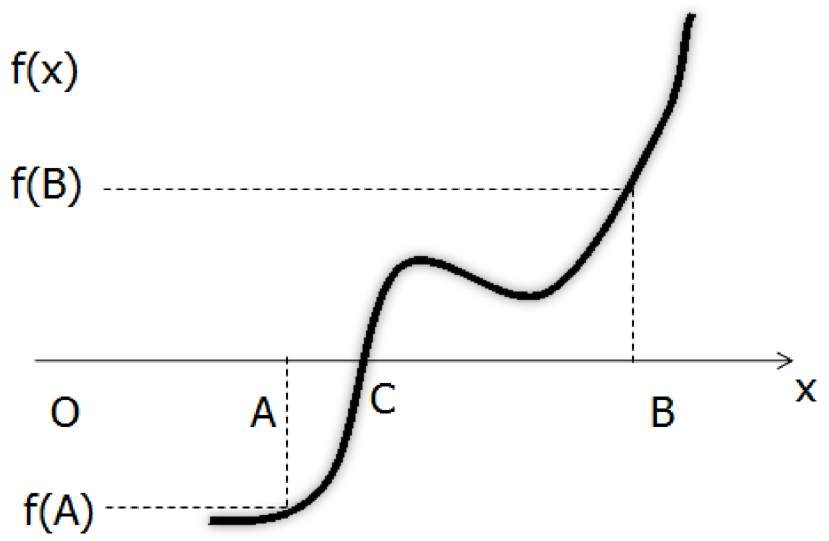
Ș.l.Dr.Ing. Levente CZUMBIL

UNIVERSITATEA TEHNICĂ
CLUJ-NAPOCA

Metoda bisecției (înjumătățirii intervalului)

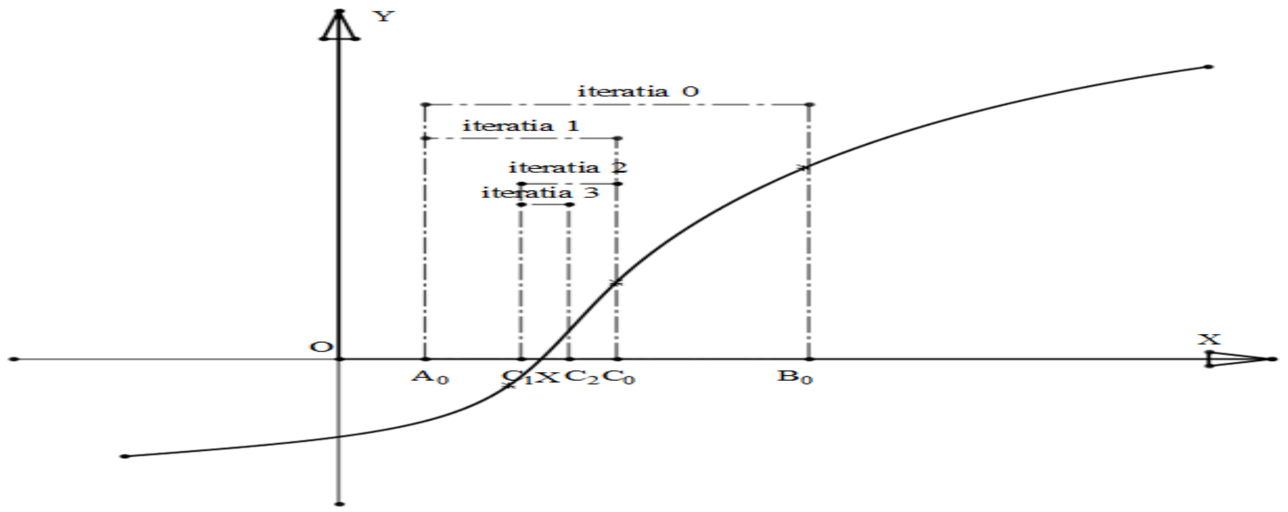
Metoda numerică constă în **despărțirea intervalului** în care se caută soluția ecuației **de un număr predefinit** de ori. Cu fiecare iterație se ajunge la o aproximare mai precisă a soluției, deci o **precizie mai bună**.

Dacă o funcție ia o valoare **negativă** într-un punct **A** și o valoare **pozitivă** într-un punct **B**, atunci funcția ia obligatoriu **valoarea 0** într-un punct **C**, care aparține intervalului **[A, B]**.



Metoda bisecției (înjumătățirii intervalului)

Se verifică valoarea funcției **la mijlocul intervalului** (punctul **C**). Dacă această valoare este pozitivă, înseamnă că funcția ia valoarea 0 în intervalul [A, C]. Dacă această valoare este exact 0, înseamnă că soluția ecuației, este exact C. Iar dacă valoarea funcției în punctul C este negativă, înseamnă că funcția ia valoarea 0 în intervalul [C, B]. Se repetă procesul pentru intervalul [A, C], respectiv [C, B], până când se ajunge până la un număr predefinit de itearații sau eroarea ajunge sub o anumită valoare impusă.



Pentru deducerea numărului minim de iterații necesare pentru a calcula soluția cu o eroare dorită se folește relația:

$$n \geq \frac{\log\left(B_0 - A_0\right) - \log\left(\varepsilon\right)}{\log\left(2\right)}$$

UNIVERSITATEA TEHNICĂ CLUJ-NAPOCA

Metoda bisecţiei (înjumătăţirii intervalului)

Repetarea procesului de căutare se realizează în contextul îndeplinirii unor condiţii şi a schimbării adaptate a capetelor intervalului. Ideea se transpune general după următorul **algoritm**:

Pasul 1: Se iniţializează intervalul: $[a;b]=[a_0;b_0]$

Pasul 2: Start iterativ: k=0

Pasul 3: Înjumătăţirea intervalului: $c_{k+1} = a_k + \frac{1}{2}\cdot(b_k - a_k)$

Pasul 4: Dacă $f(c_{k+1})\cdot f(a_k) < 0 \Rightarrow a_{k+1} = a_k; b_{k+1} = c_{k+1}$

Pasul 5: Dacă $f(c_{k+1})\cdot f(b_k) < 0 \Rightarrow a_{k+1} = c_{k+1}; b_{k+1} = b_k$

Pasul 6: Se incrementează: k=k+1; şi se reia **Pasul 3**.

Oricât de mult ar fi dusă restrângerea intervalului în jurul soluţiei, există posibilitatea ca valoarea considerată drept soluţie **să nu fie cea adevărată**.

Metoda bisecției (înjumătățirii intervalului)

Se consideră ecuația polinomială: $4x^5 - 8x^4 - 56x^3 + 23x^2 - 17x - 67 = 0$

Să se determine rădăcinile ecuației utilizând metoda bisecției, cu o eroare $\varepsilon \le 10^{-6}$

Pasul 1. Se introduce ecuația în Mathcad, folosind **egalul boolean**. (Ctrl =)

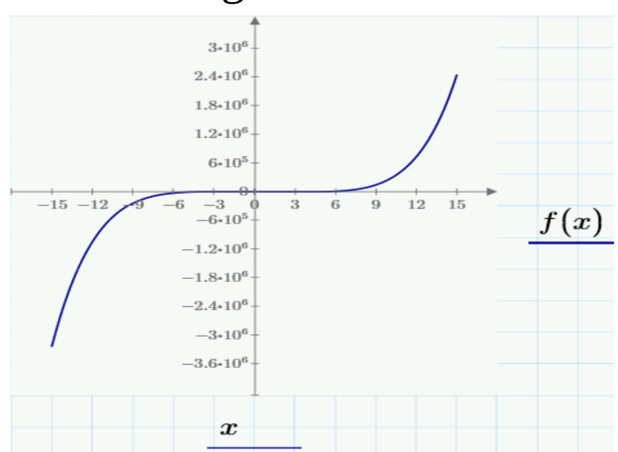
$$4x^5 - 8x^4 - 56x^3 + 23x^2 - 17x - 67 = 0$$

Pasul 2. Se determină **funcția atașată ecuației**.

$$f(x) := 4x^5 - 8x^4 - 56x^3 + 23x^2 - 17x - 67$$

Pasul 3. Se reprezintă grafic funcția f(x) de la -15 până la 15, cu o precizie de 0,1. Se setează afișarea grilei pe ambele axe, cu o culoare gri.

$x := -15, -14.99 .. 15$



UNIVERSITATEA TEHNICĂ CLUJ-NAPOCA

Metoda bisecției (înjumătățirii intervalului)

Pasul 4. Se aleg de pe grafic două valori A_0 și B_0 astfel încât funcția caracteristică atașată ecuației să fie **negativă** pentru valorarea A_0 și **pozitivă** pentru valoarea B_0. Se definește valoare erorii cu care se dorește determinarea soluției.

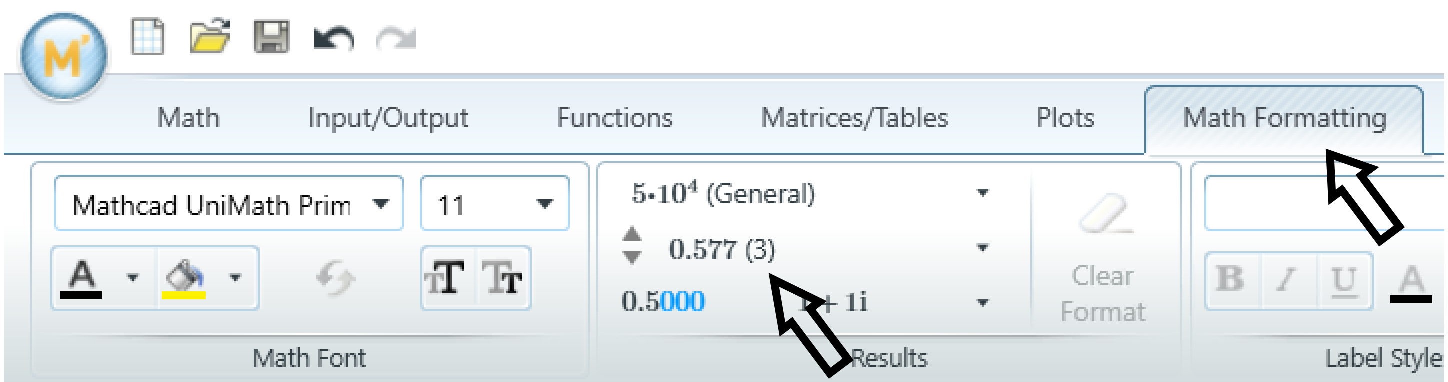
$$A_0 := -11 \qquad f(A_0) = -6.839\times 10^5$$

$$B_0 := 12 \qquad f(B_0) = 7.357\times 10^5$$

$$\varepsilon := 10^{-6}$$

Pasul 5. Se calcluează numărul minim de iterații necesare, conform relației .

$$n := \frac{\log(B_0 - A_0) - \log(\varepsilon)}{\log(2)} \qquad n = 24.455 \qquad n := 25$$

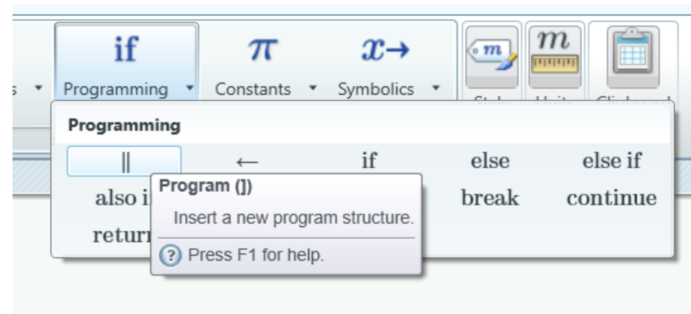


UNIVERSITATEA TEHNICĂ CLUJ-NAPOCA

Metoda bisecţiei (înjumătăţirii intervalului)

Pasul 6. Se trece la implementarea algoritmului de calcul a soluţiei sub <u>formă de linii de cod</u> prezentat mai jos. Se defineşte variabila *sol* la care se atribuie segvenţa de program ce urmează să fie introdusă introducând 4 linii de cod prin comanda || in toolbar-ul *Programing* (tasta “]”).

$$sol := \left| \begin{array}{l} \blacksquare \\ \blacksquare \\ \blacksquare \\ \blacksquare \end{array} \right.$$



Pasul 7. Se definesc valorile iniţiale ale vectorilor interni *a* şi b egale cu limitele de intervalului de căutare a soluţiei A_0 şi B_0.

$$sol := \left| \begin{array}{l} a_0 \leftarrow A_0 \\ b_0 \leftarrow B_0 \\ \blacksquare \\ \blacksquare \end{array} \right.$$

Pentru atribuiri în cadrul buclelor de programare se foloseşte săgeata la stânga ← din toolbar-ul *Programing*.

UNIVERSITATEA TEHNICĂ CLUJ-NAPOCA

Metoda bisecţiei (înjumătăţirii intervalului)

Pasul 8. Se introduce bucla repetitivă *for* în cadrul careia are loc căutarea soluţia ecuaţie studiate. Bucla repetitivă *for* se introduce din paleta *Programming*. Se introduc 3 linii de cod în cadrul bulei *for.*

$$
sol := \left|\begin{array}{l} a_0 \leftarrow A_0 \\ b_0 \leftarrow B_0 \\ \text{for } k \in 0..n \end{array}\right.
$$

$$
sol := \left|\begin{array}{l} a_0 \leftarrow A_{.0} \\ b_0 \leftarrow B_{.0} \\ \text{for } k \in 0..n \\ \quad \left| c_{k+1} \leftarrow \dfrac{a_k + b_k}{2} \right. \end{array}\right.
$$

Pasul 9. Se calculează poziţia de mijloc a intervalului curent.

Indicile la a_k, b_k, c_k se introduc utilizând paranteza pătratică ([ ).

UNIVERSITATEA TEHNICĂ CLUJ-NAPOCA

Pasul 10. Se determină noile capete ale intervalului de căutare a soluţiei folosindu-se instrucţiunea condiţională *if* introdusă de la tastatură. Actualizarea limitelor intervalului de căutare se face în funţie de valoarea funcţiei în punctul de mijloc c_{k+1}.

$$
sol := \left|
\begin{array}{l}
a_0 \leftarrow A_0 \\
b_0 \leftarrow B_0 \\
\text{for } k \in 0..n \\
\quad \left|
\begin{array}{l}
c_{k+1} \leftarrow \dfrac{a_k + b_k}{2} \\
a_{k+1} \leftarrow \text{if}\left(f\left(a_k\right) \cdot f\left(c_{k+1}\right) < 0, a_k, c_{k+1}\right) \\
b_{k+1} \leftarrow \text{if}\left(f\left(b_k\right) \cdot f\left(c_{k+1}\right) < 0, b_k, c_{k+1}\right)
\end{array}
\right. \\
\blacksquare
\end{array}
\right.
$$

$$
sol := \left|
\begin{array}{l}
a_0 \leftarrow A_0 \\
b_0 \leftarrow B_0 \\
\text{for } k \in 0..n \\
\quad \left|
\begin{array}{l}
c_{k+1} \leftarrow \dfrac{a_k + b_k}{2} \\
a_{k+1} \leftarrow \text{if}\left(f\left(a_k\right) \cdot f\left(c_{k+1}\right) < 0, a_k, c_{k+1}\right) \\
b_{k+1} \leftarrow \text{if}\left(f\left(b_k\right) \cdot f\left(c_{k+1}\right) < 0, b_k, c_{k+1}\right)
\end{array}
\right. \\
c
\end{array}
\right.
$$

Pasul 11. Se returnează vectorul de poziţii centrale *c* în variabila externă sol şi se afișează soluția: **sol =**

Începând cu cea de a 25-a iterație, primi 6 numere după virgulă rămân identici.

UNIVERSITATEA TEHNICĂ CLUJ-NAPOCA

Pasul 6. Se calculează soluția cu ajutorul algoritmului prezentat mai jos. Algoritmul folosește de instrucțiuniile de programare *for* și *if*.

Indicile la a_k, b_k, c_k se introduc utilizând paranteza pătratică ([ ).

$$
sol := \left|\begin{array}{l}
a_0 \leftarrow A_0 \\
b_0 \leftarrow B_0 \\
\text{for } k \in 0..n \\
\quad \left|\begin{array}{l}
c_{k+1} \leftarrow \dfrac{a_k + b_k}{2} \\
a_{k+1} \leftarrow \text{if}\left(f\left(a_k\right) \cdot f\left(c_{k+1}\right) < 0, a_k, c_{k+1}\right) \\
b_{k+1} \leftarrow \text{if}\left(f\left(b_k\right) \cdot f\left(c_{k+1}\right) < 0, b_k, c_{k+1}\right)
\end{array}\right. \\
c
\end{array}\right.
$$

Pasul 7. Se afișează soluția: **sol =**

Începând cu cea de a 25-a iterație, primi 6 numere după virgulă rămân identici.

UNIVERSITATEA TEHNICĂ CLUJ-NAPOCA

Metoda bisecției (înjumătățirii intervalului)

sol =

0	0
1	0.5
2	6.25
3	3.375
4	4.813
5	4.094
6	4.453
7	4.633
8	4.723
9	4.768
10	4.745
11	4.756
12	4.762
13	4.759
14	4.761
15	4.76
16	4.76
17	4.76
18	4.76
19	4.76
20	4.76
⋮	⋮
26	

sol =

	0
0	0
1	0.5
2	6.25
3	3.375
4	4.8125
5	4.09375
6	4.453125
7	4.6328125
8	4.7226563
9	4.7675781
10	4.7451172
11	4.7563477
12	4.7619629
13	4.7591553
14	4.7605591
15	4.7598572
16	4.7595062
17	4.7596817
18	4.759594
19	4.7596378
20	4.7596159
21	4.7596269
22	4.7596323
23	4.7596351
24	4.7596365
25	4.7596371
26	4.7596375

Metoda bisecției (înjumătățirii intervalului)

Pasul 12. Pentru a verifica soluția obținută ecuația se rezolvă și cu funcția *Symbolics – Solve*

$$4x^5 - 8x^4 - 56x^3 + 23x^2 - 17x - 67 = 0$$

$$\begin{pmatrix} 4.7596372057299709858 \\ 0.61532233984573568775 + 0.94212692320233033177i \\ 0.61532233984573568775 - 0.94212692320233033177i \\ -0.89909303721373502328 \\ -3.0911888482077073338 \end{pmatrix}$$

$$x_0 := 4.7596372057299709858$$

Pasul 13. Se compară cele două rezultate obținute.

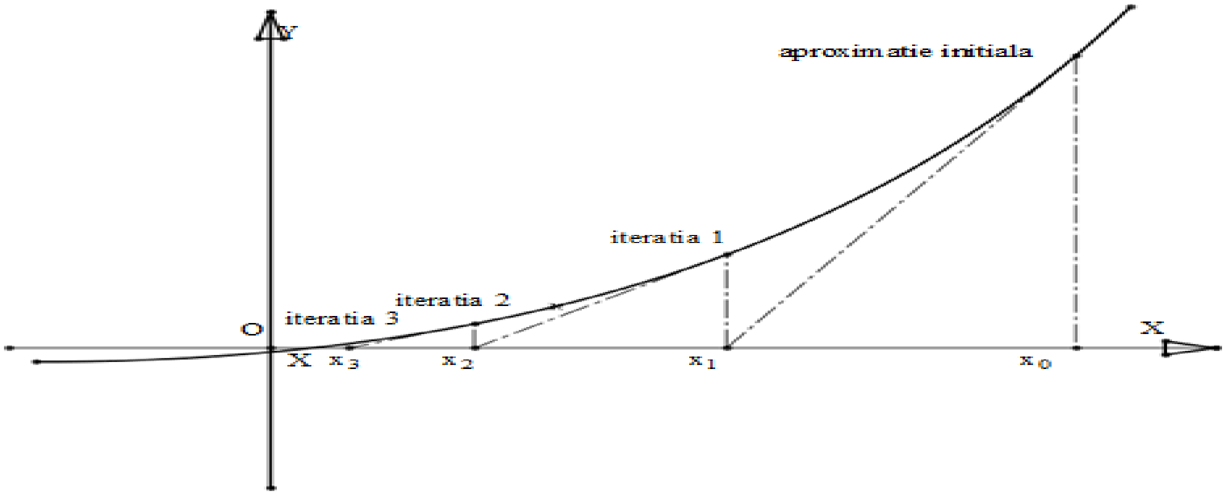
$$E_r := \left|sol_n - x_0\right| = 5.854 \times 10^{-8}$$

Pasul 14. Se salvează reultatul obținut în variabila denumită $x_{bisectie}$

$$x_{bisectie} := sol$$

Metoda lui Newton (tangente)

Metoda tangentei presupune **alegerea anterioară** a unei aproximații x_0 a soluției și trasarea unei **tangente** la graficul funcției în punctul x_0. Intersecția tangentei trase cu axa X, reprezintă următoarea aproximare a soluției căutate. Se repetă acest procedeu până se găsește soluția căutată sau până se determină o aproximare cu o anumită eroare impusă.



Algoritmul matematic care stă la baza **metodei lui Newton** se deduce din descompunerea în **serie Taylor** în jurul punctului x_0 a funcției atașate ecuației de rezolvat, scriind doar primii doi termeni ai seriei Taylor:

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}\cdot(x-x_0) + \frac{f''(x_0)}{2!}\cdot(x-x_0)^2 + \frac{f'''(x_0)}{3!}\cdot(x-x_0)^3 + \cdots + \frac{f^{(n)}(x_0)}{n!}\cdot(x-x_0)^n$$

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}\cdot(x-x_0)$$

UNIVERSITATEA TEHNICĂ CLUJ-NAPOCA

Metoda lui Newton (tangente)

Considerând $\mathbf{x_0}$ o primă aproximare, și înlocuind **x** cu $\mathbf{x_1}$, rezultă o nouă aporximare. Înlocuind în expresia seriei Taylor se poate deduce **formula de calcul iteratiiv**:

$$0 = f(x_0) + f'(x_0)\cdot(x_1 - x_0) \qquad x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

De aici se deduce **forma generală** de calcul a soluției ecuației la iterația **k+1** în funcție de aproximarea soluției obținute la iterația precedentă, **k**.

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Pentru a determina **numărul minim de iterații necesare** se folosește formula:

$$N \ge \log_2\left(\frac{\log(\varepsilon) + \log(M)}{\log(M\cdot\varepsilon_{admisibil})}\right) \qquad M = \frac{\max(f'(x))}{2\cdot\min(f'(x))}$$

unde: ε – eroarea de aproximare a soluției căutate

$\varepsilon_{admisibil}$ – eraorea maximă admisibilă a aproximării inițiale

UNIVERSITATEA TEHNICĂ CLUJ-NAPOCA

Metoda lui Newton (tangentei)

Se consideră ecuația polinomială: $4x^5 - 8x^4 - 56x^3 + 23x^2 - 17x - 67 = 0$

Să se determine rădăcinile ecuației utilizând metoda tangentei, cu o eroare $\varepsilon \leq 10^{-6}$

Pasul 1. Se introduce ecuația în Mathcad, folosind **egalul boolean**. (Ctrl =)

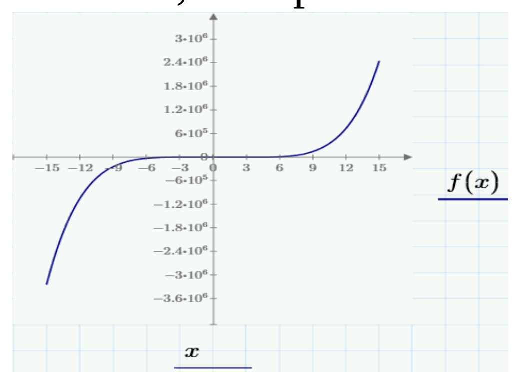
$$4x^5 - 8x^4 - 56x^3 + 23x^2 - 17x - 67 = 0$$

Pasul 2. Se determină **funcția atașată ecuației**.

$$f(x) := 4x^5 - 8x^4 - 56x^3 + 23x^2 - 17x - 67$$

Pasul 3. Se reprezintă grafic funcția f(x) de la -15 până la 15, cu o precizie de 0,1.

$$x := -15, -14.99 .. 15$$



Pasul 4. Se alege o aproximație a soluției. De exemplu x_0=12. Indicele se introduce utilizând paranteza pătratică ([ ).

UNIVERSITATEA TEHNICĂ CLUJ-NAPOCA

Metoda lui Newton (tangente)

Pasul 5. Se definește funcția *f ’(x),* derivata funcției atașate ecuație studiate.

În fereastra de comandă se introduce numele funcției (f de obicei) urmat de un **apostrof** (`) după care se introduce un operator de atribuire (:=).

Pe partea dreaptă a operatorului de atribuire se **introduce formula derivatei**, selectând *Derivative* din toolbar-ul *Calculus.*

Operators -> Calculus -> $\frac{d}{d\blacksquare}\blacksquare$

$$f'(x) := \frac{d}{dx}f(x)$$

Pasul 6. Se calculează iterativ soluția ecuația cu formula recursivă determinată din seria lui Taylor:

$$N := 20 \qquad k := 0..N \qquad f'(x) := \frac{d}{dx}f(x) \qquad x_{k+1} := x_k - \frac{f(x_k)}{f'(x_k)}$$

Pasul 7. Se compară rezultatul obținut cu rezultatul obținut prin metoda bisecției.

$$x = \begin{bmatrix} 12 \\ 9.809 \\ 8.097 \\ 6.787 \\ 5.827 \\ 5.19 \\ 4.86 \\ 4.767 \\ 4.76 \\ 4.76 \\ 4.76 \\ 4.76 \\ \vdots \end{bmatrix} \qquad sol = \begin{bmatrix} 0 \\ 0.5 \\ 6.25 \\ 3.375 \\ 4.813 \\ 4.094 \\ 4.453 \\ 4.633 \\ 4.723 \\ 4.768 \\ 4.745 \\ 4.756 \\ \vdots \end{bmatrix}$$

Se observă că încă de la iterația a 9-a se obține o aproximare cu o eroare de $\varepsilon \le 10^{-6}$, pe când la metoda bisecției, după 13 iterații se obține o eroare de $\varepsilon \le 10^{-3}$. De aici se poate observa eficiența acestei metode.

Pasul 8. Se salvează rezultatul obținut în variabila denumită x_{Newton}

$$x_{Newton} := x$$

Metoda coardei (secantei, Newton discretă)

Pentru a reduce efortul de calcul cerut de metoda tangentei s-a dezvoltat o nouă metodă, care în locul tangentei folosește o **secantă** trasă între două puncte aparținând graficului funcției, ne mai fiind nevoie de determinarea **derivatei de ordinul întâi.**

Dacă x_0 este o **aproximare inițială a soluției**, prin proces iterativ se poate defini algoritmul corespunzător metodei secantei:

$$x_0 = \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)} \qquad x_k = \frac{x_0 \cdot f(x_{k-1}) - x_{k-1} \cdot f(x_0)}{f(x_{k-1}) - f(x_0)}$$

Pentru ca metoda secantei **să convează** la o soluție, trebuie să se îndeplinească următoarele condiții:

$$f(a) \cdot f(b) < 0$$

$$f(x_0) \cdot f''(x_0) > 0$$

$$f(x_0) \cdot f(x_1) < 0$$

UNIVERSITATEA TEHNICĂ CLUJ-NAPOCA

Metoda coardei (secantei, Newton discretă)

Se consideră ecuația polinomială: $4x^5 - 8x^4 - 56x^3 + 23x^2 - 17x - 67 = 0$

Să se determine rădăcinile ecuației utilizând metoda tangentei, cu o eroare $\varepsilon \le 10^{-6}$

Pasul 1. Se introduce ecuația în Mathcad, folosind **egalul boolean**. (Ctrl =)

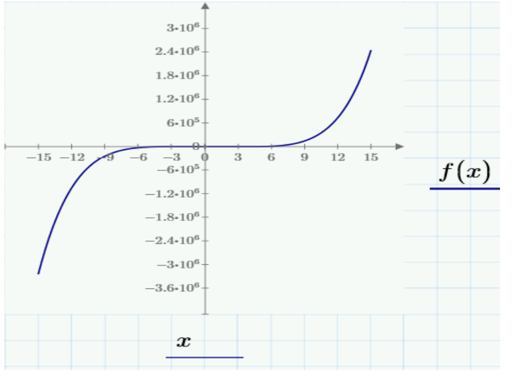
$$4x^5 - 8x^4 - 56x^3 + 23x^2 - 17x - 67 = 0$$

Pasul 2. Se determină **funcția atașată ecuației**.

$$f(x) := 4x^5 - 8x^4 - 56x^3 + 23x^2 - 17x - 67$$

Pasul 3. Se reprezintă grafic funcția f(x) de la -15 până la 15, cu o precizie de 0,1.

$$x := -15, -14.99 .. 15$$

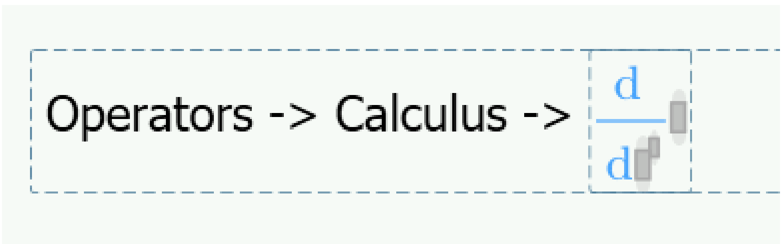


Pasul 4. Se alege o aproximație a soluției. De exemplu x_0=12. Indicele se introduce utilizând paranteza pătratică ([ ).

Metoda coardei (secantei, Newton discretă)

Pasul 5. Se definește o funție *f''(x),* derivata de ordinul 2 a funcției atașate ecuației studiate necesare verificării convergenței metodei.

Derivata de odinul 2 se introduce utilizând operatorul *Nth Derivative* din toolbar-ul *Calculus,* la exponent se introducându-se valoarea 2.



$$f''(x) := \frac{d^2}{dx^2} f(x)$$

Pasul 6. Se alege de pe grafic intervalul [a,b] în care se caută soluţia ecuaţiei, astfel încât funcția caracteristică atașată ecuației să fie **negativă** pentru valorarea **a** și **pozitivă** pentru valoarea **b**, și o **aproximare inițială** a soluției, $\mathbf{x_0}$.

Toţi indicii se introduc utilizând paranteza pătrată ([ ).

$a := -11 \qquad b := 12 \qquad x_0 := 5$

Metoda coardei (secantei, Newton discretă)

Pasul 7. Se verifică dacă **condițiile de convergență** a metodei coardei sunt îndeplinite în cazul ecuației prezente.

$$x_1 := \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)} \qquad x_1 = 0.0802145$$

$$f(x_0) \cdot f'(x_0) > 0 = 1 \qquad f(x_0) \cdot f(x_1) < 0 = 1 \qquad f(a)\, f(b) < 0 = 1$$

Pasul 8. Se calculează soluția cu ajutorul algoritmului corespunzător metodei secantei.

$$N := 20 \qquad k := 2..N \qquad x_k := \frac{x_0 \cdot f(x_{k-1}) - x_{k-1} \cdot f(x_0)}{f(x_{k-1}) - f(x_0)}$$

Metoda coardei (secantei, Newton discretă)

Pasul 9. Se salvează reultatul obținut în variabila denumită x_{coarda} şi se afişează rezultatul.

$$x_{coarda} := x$$

$$x_{coarda} = \begin{bmatrix} 5 \\ 0.080215 \\ 0.41893 \\ 0.760639 \\ 1.148417 \\ 1.677349 \\ 2.503204 \\ 3.634568 \\ 4.451022 \\ 4.701952 \\ 4.749943 \\ 4.75804 \\ 4.759375 \\ 4.759594 \\ \vdots \end{bmatrix}$$

UNIVERSITATEA TEHNICĂ CLUJ-NAPOCA

Metoda coardei (secantei, Newton discretă)

Pasul 10. Se compară soluția cu soluțiile obținute cu celelate două metode.

$$x_{bisectie} = \begin{bmatrix} 0 \\ 0.5 \\ 6.25 \\ 3.375 \\ 4.8125 \\ 4.09375 \\ 4.453125 \\ 4.632813 \\ 4.722656 \\ 4.767578 \\ 4.745117 \\ 4.756348 \\ 4.761963 \\ 4.759155 \\ 4.760559 \\ 4.759857 \\ 4.759506 \\ 4.759682 \\ 4.759594 \\ 4.759638 \\ \vdots \end{bmatrix} \quad x_{Newton} = \begin{bmatrix} 12 \\ 9.808859 \\ 8.097237 \\ 6.787119 \\ 5.826719 \\ 5.189606 \\ 4.859593 \\ 4.766595 \\ 4.759674 \\ 4.759637 \\ 4.759637 \\ 4.759637 \\ 4.759637 \\ 4.759637 \\ 4.759637 \\ 4.759637 \\ 4.759637 \\ 4.759637 \\ 4.759637 \\ 4.759637 \\ 4.759637 \\ 4.759637 \end{bmatrix} \quad x_{coarda} = \begin{bmatrix} 5 \\ 0.080215 \\ 0.41893 \\ 0.760639 \\ 1.148417 \\ 1.677349 \\ 2.503204 \\ 3.634568 \\ 4.451022 \\ 4.701952 \\ 4.749943 \\ 4.75804 \\ 4.759375 \\ 4.759594 \\ 4.75963 \\ 4.759636 \\ 4.759637 \\ 4.759637 \\ 4.759637 \\ 4.759637 \\ 4.759637 \end{bmatrix}$$

Metoda lui Halley

Pentru a obține o **convergență mai rapidă** decât cea dată de metoda lui Newton se folosește o dezvoltare în serie Taylor a funcției atașate ecuației studiate, folosind **primele 3 componente** din dezvoltare.

$$f(x) = f(x_0) + f'(x_0)\cdot(x - x_0) + \frac{1}{2}\cdot f''(x_0)\cdot(x - x_0)^2$$

De aici se poate ajunge la formula iterativă de calcul:

$$x_{k+1} = x_k - \frac{2\cdot f(x_k)}{f'(x_k) - \sqrt{f'(x_k)^2 - 2\cdot f(x_k)\cdot f''(x_k)}}$$

Se rezolvă aceeaşi ecuaţie polinomială ca şi la metodele anterioare:

$$4x^5 - 8x^4 - 56x^3 + 23x^2 - 17x - 67 = 0$$

Pasul 1. Se introduce ecuația în Mathcad, folosind **egalul boolean**. (Ctrl =)

$$4x^5 - 8x^4 - 56x^3 + 23x^2 - 17x - 67 = 0$$

UNIVERSITATEA TEHNICĂ CLUJ-NAPOCA

Pasul 2. Se determină **funcția atașată ecuației**.

$$f(x) := 4x^5 - 8x^4 - 56x^3 + 23x^2 - 17x - 67$$

Pasul 3. Se definesc operatori de derivare de ordinul 1, respectiv ordinul 2, necesare algoritmului.

$$f'(x) := \frac{d}{dx} f(x) \qquad f''(x) := \frac{d^2}{dx^2} f(x)$$

Pasul 4. Se calculează soluția cu ajutorul algoritmului corespunzător metodei lui Halley.

$$N := 20 \qquad k := 0..N \qquad x_{k+1} := x_k - \frac{2 f(x_k)}{f'(x_k) + \sqrt{f'(x_k)^2 - 2 f(x_k) \cdot f''(x_k)}}$$

Pasul 5. Se salvează rezultatul obținut în variabila denumită x_{Halley}

$$x_{Halley} := x$$

3.4. Rezolvare

$x_{Halley} := x$

$x_{Halley} =$

0	12
1	9.2119322+2.1082647i
2	7.6239293+0.0116156i
3	6.0201259-1.106818i
4	5.2890236-0.0374716i
5	4.7080627+0.0158223i
6	$4.7596584+2.6012945i\cdot 10^{-5}$
7	4.7596372
8	4.7596372
9	4.7596372
10	4.7596372
11	4.7596372
12	4.7596372

Metoda lui Halley

Pasul 6. Se vizualizează şi se compară soluția cu soluția obținută prin metoda lui Newton.

$x_{Newton} =$

0	12
1	9.8088585
2	8.097237
3	6.7871189
4	5.8267193
5	5.1896056
6	4.8595929
7	4.7665952
8	4.7596739
9	4.7596372
10	4.7596372
11	4.7596372
12	4.7596372
13	4.7596372

$x_{Halley} =$

0	12
1	9.2119322+2.1082647i
2	7.6239293+0.0116156i
3	6.0201259-1.106818i
4	5.2890236-0.0374716i
5	4.7080627+0.0158223i
6	$4.7596584-2.6012945i\cdot 10^{-5}$
7	4.7596372
8	4.7596372
9	4.7596372
10	4.7596372
11	4.7596372
12	4.7596372
13	4.7596372

REZOLVAREA APROXIMATIVĂ A ECUAȚIILOR ALGEBRICE ȘI TRANSCENDENTE – PARTEA 2

Ș.l.Dr.Ing. Levente CZUMBIL

UNIVERSITATEA TEHNICĂ CLUJ-NAPOCA