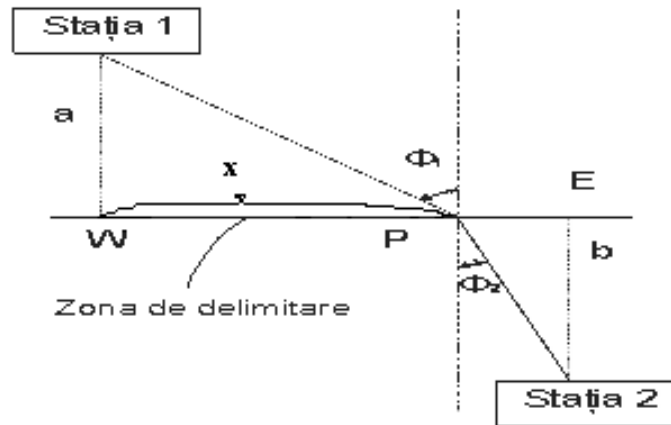


REZOLVAREA APROXIMATIVĂ A ECUAȚIILOR ALGEBRICE ȘI TRANSCENDENTE – PARTEA 2

Dr.Ing. Levente CZUMBIL

Aplicație

Societatea de transport a energiei electrice, S.C TRANSELECTRICA S.A, și-a stabilit că obiectiv de investiții pe anul 2010 construirea unei linii electrice de interconexiune între două stații electrice. Pe distanța dintre stații, datorită diferențelor de teren, amplasamentul se împarte în două zone, delimitate printr-o linie WE, astfel încât costul execuției liniei pe fiecare zonă se caracterizează prin indicii C_1 și C_2 .



Având la dispoziție indicii de cost, se poate stabili o relație de egalitate între sinusurile unghiurilor formate de direcțiile traseelor de linie din cele două zone și perpendiculara dusă prin punctul P la zona de delimitare:

$$C_1 \cdot \sin \theta_1 = C_2 \cdot \sin \theta_2$$

Aplicație

Prin ridicare la pătrat și exprimarea sinusurilor în raport cu distanțele se obține următoarea relație de echivalență:

$$C_1^2 \cdot \frac{x^2}{a^2 + x^2} = C_2^2 \cdot \frac{(L-x)^2}{b^2 + (L-x)^2}$$

după o rearanjare convenabilă, arată de forma:

$$C_1^2 \cdot x^2 \cdot [b^2 + (L-x)^2] = C_2^2 \cdot (L-x)^2 \cdot (a^2 + x^2)$$

și reprezintă o ecuație algebrică. Exprimarea devine mai sugestivă prin înlocuirea datelor cunoscute: $a=3$ km; $b=1$ km; $WE=L=12$ km; $C_1=87000$ lei/km; $C_2=93000$ lei/km.

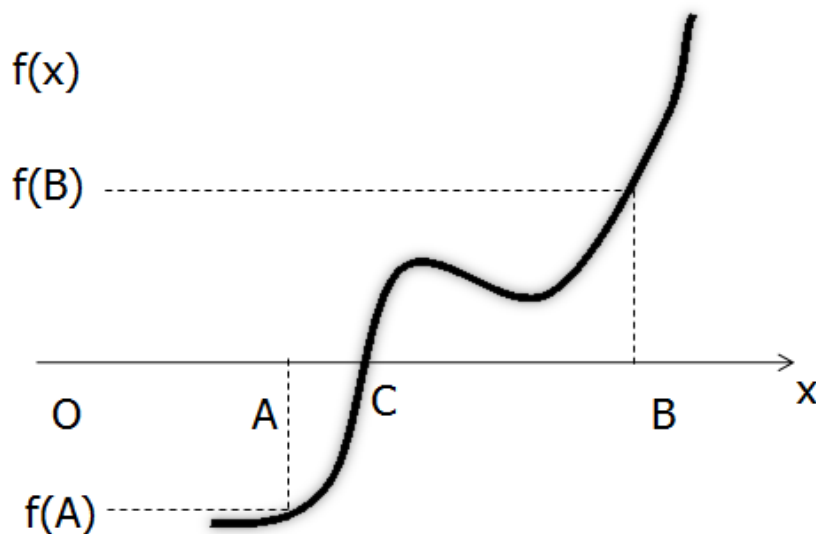
$$-1080 \cdot x^4 + 25920 \cdot x^3 - 225792 \cdot x^2 + 1868184 \cdot x - 11209104 = 0$$

Necunoscuta în această ecuație polinomială este distanța x de la marginea W la punctul P . Rămâne de rezolvat numeric această ecuație.

Metoda biseecției (înjumătățirii intervalului)

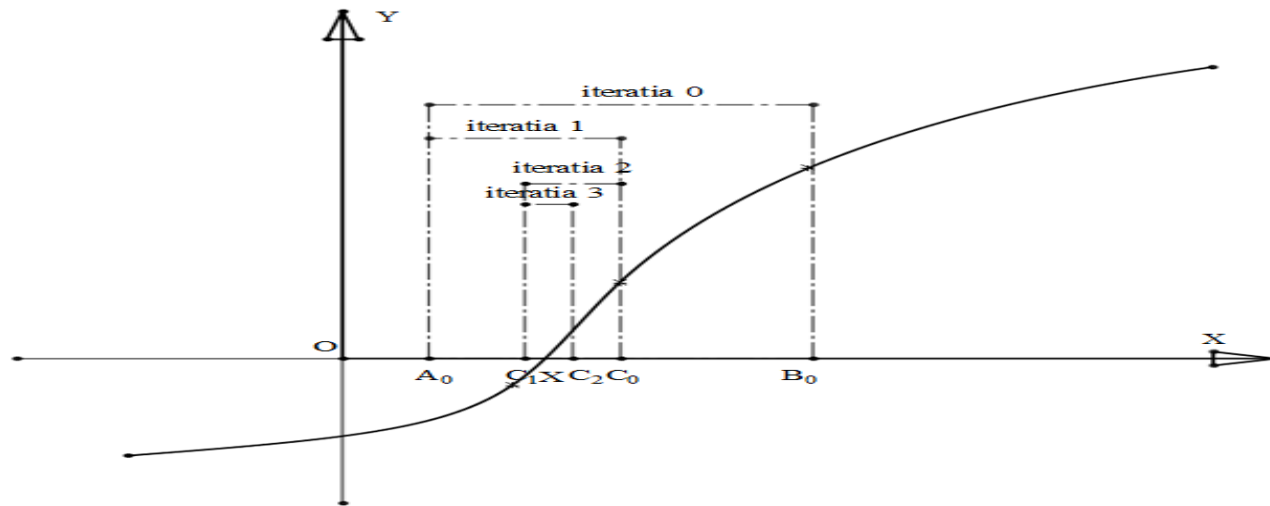
Metoda numerică constă în **despărțirea intervalului** în care se caută soluția ecuației **de un număr predefinit** de ori. Cu fiecare iterație se ajunge la o aproximare mai precisă a soluției, deci o **precizie mai bună**.

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 biseecției (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 iterații sau eroarea ajunge sub o anumită valoare impusă.



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

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

Metoda biseției (Înjumătățirii intervalului)

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

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

Pasul 2: Start iterativ: $k=0$

Pasul 3: Înjumătățirea 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: $k=k+1$; și se reia **Pasul 3**.

Oricât de mult ar fi dusă restrângerea intervalului în jurul soluției, există posibilitatea ca valoarea considerată drept soluție să **nu fie cea adevărată**. Pentru deducerea numărului minim de iterații necesare pentru a calcula soluția cu o eroare dorită se folosește relația:

Metoda biseției (înjumătățirii intervalului)

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

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

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

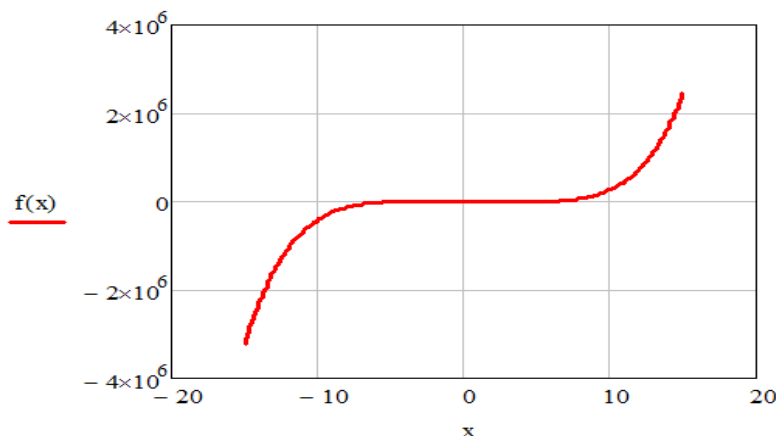
$$4x^5 - 8x^4 - 56x^3 + 23x^2 - 17x - 67 = 0$$

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. Se setează afișarea grilei pe ambele axe, cu o culoare gri.

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



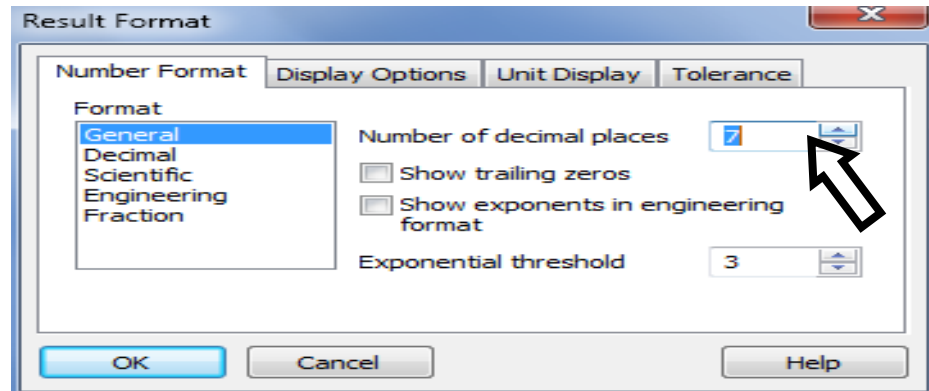
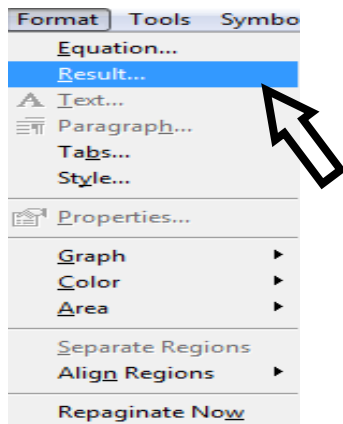
Metoda biseției (înjumătățirii intervalului)

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

$$\begin{aligned} A_0 &:= -11 & f(A_0) &= -6.839 \times 10^5 \\ B_0 &:= 12 & f(B_0) &= 7.357 \times 10^5 \end{aligned} \quad \varepsilon := 10^{-6}$$

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

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



Metoda biseției (înjumătățirii intervalului)

Pasul 6. Se trece la implementarea algoritmului de calcul a soluției sub formă de linii de cod prezentat mai jos. Se definește variabila *sol* la care se atribuie secvența de program ce urmează să fie introdusă introducând 4 linii de cod prin comanda *Add Line* în toolbar-ul *Programing* (tasta “]”).

```
sol := |  
      |  
      |  
      |  
      |
```

Pasul 7. Se definesc valorile inițiale ale vectorilor interni *a* și *b* egale cu limitele de intervalului de căutare a soluției A_0 și B_0 .

```
sol := | a0 ← A0  
      | b0 ← B0  
      |  
      |  
      |
```

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

Metoda biseției (înjumătățirii intervalului)

Pasul 10. Se determină noile capete ale intervalului de căutare a soluției folosindu-se instrucțiunea condițională *if* introdusă de la tastatură. Actualizarea limitelor intervalului de căutare se face în funcție de valoarea funcției în punctul de mijloc c_{k+1} .

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

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

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

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



Metoda biseției (Înjumătățirii intervalului)

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

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

```
sol := | a0 ← A0  
      | b0 ← B0  
      | for k ∈ 0..n  
      | | ck+1 ←  $\frac{a_k + b_k}{2}$   
      | | ak+1 ← if(f(ak) · f(ck+1) < 0, ak, ck+1)  
      | | bk+1 ← if(f(bk) · f(ck+1) < 0, bk, ck+1)  
      | c
```

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

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



Metoda biseției (înjumătățirii intervalului)

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

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 biseției (Înjumătățirii intervalului)

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

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

$$\left(\begin{array}{c} 4.7596372057299709858 \\ 0.61532233984573568775 + 0.94212692320233033177i \\ 0.61532233984573568775 - 0.94212692320233033177i \\ -0.89909303721373502328 \\ -3.091188848207707338 \end{array} \right)$$

$$x_0 := 4.7596372057299709858$$

Pasul 13. Se compară cele două rezultate obținute.

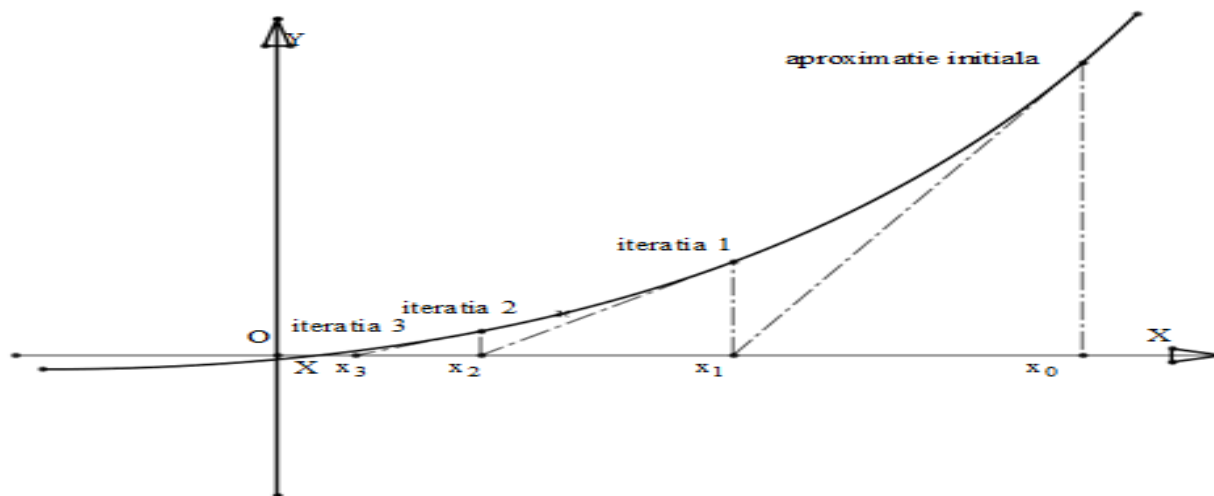
$$E_r := |sol_n - x_0| = 5.854 \times 10^{-8}$$

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

$$x_{\text{bisectie}} := sol$$

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 aproximație a soluției căutate. Se repetă acest procedeu până se găsește soluția căutată sau până se determină o aproximație 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:

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

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



Metoda lui Newton (tangente)

Considerând x_0 o primă aproximare, și înlocuind x cu x_1 , rezultă o nouă aproximare. Înlocuind în expresia seriei Taylor se poate deduce **formula de calcul iterativ**:

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

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:

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

unde: ε – eroarea de aproximare a soluției căutate

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

Metoda lui Newton (tangente)

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 =)

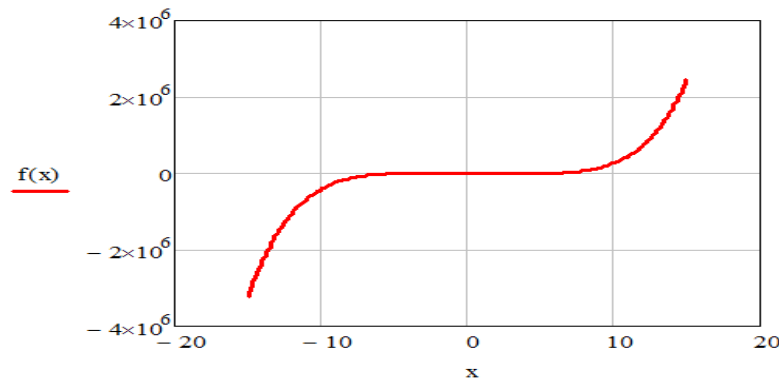
$$4x^5 - 8x^4 - 56x^3 + 23x^2 - 17x - 67 = 0$$

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.

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



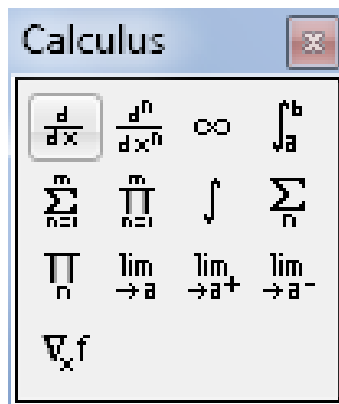
Pasul 4. Se alege o aproximație a soluției. De exemplu $x_0=12$. Indicele se introduce utilizând paranteza pătratică ([).

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*, prezentat pe figura 3.10.



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

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

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

Metoda lui Newton (tangente)

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

7	4.7665952
8	4.7596739
9	4.7596372
10	4.7596372
11	4.7596372
12	4.7596372
13	4.7596372
14	4.7596372

7	4.6328125
8	4.7226563
9	4.7675781
10	4.7451172
11	4.7563477
12	4.7619629
13	4.7591553
14	4.7605591

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

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

$$x_{\text{Newton}} := x$$

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:

$$x_0 = \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)}$$

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

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

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

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

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

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 $\varepsilon \leq 10^{-6}$

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

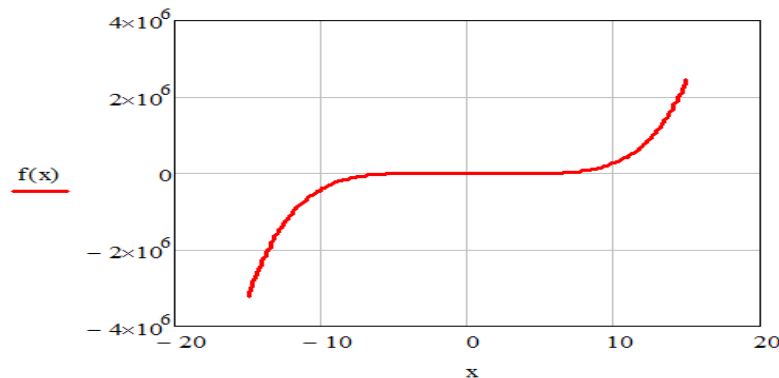
$$4x^5 - 8x^4 - 56x^3 + 23x^2 - 17x - 67 = 0$$

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.

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

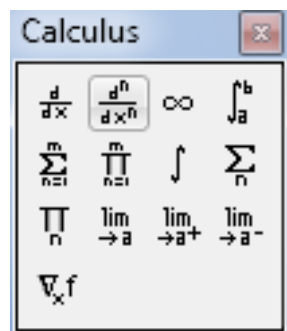


Pasul 4. Se alege o aproximație a soluției. De exemplu $x_0=12$. Indicele se introduce utilizând paranteza pătratică ([).

Metoda coardei (secantei, Newton discretă)

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

Derivata de ordinul 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 soluția ecuației, astfel încât funcția caracteristică atașată ecuației să fie **negativă** pentru valoarea **a** și **pozitivă** pentru valoarea **b**, și o **aproximare inițială** a soluției, x_0 .

Toți indicii se introduc utilizând paranteza pătrată ([]).

$$a := -11$$

$$b := 12$$

$$x_0 := 5$$

Metoda coardei (secantei, Newton discretă)

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

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

$$x_1 = 0.0802145$$

$$f(x_0) \cdot f'(x_0) > 0 = 1$$

$$f(x_0) \cdot f(x_1) < 0 = 1$$

$$f(a) f(b) < 0 = 1$$

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

$$N := 20$$

$$k := 2..N$$

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

Metoda coardei (secantei, Newton discretă)

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

$x_{\text{bisectie}} =$

	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	...

$x_{\text{Newton}} =$

	0
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
14	4.7596372
15	...

$x_{\text{coarda}} =$

	0
0	5
1	0.0802145
2	0.4189302
3	0.760639
4	1.1484174
5	1.6773492
6	2.5032044
7	3.6345678
8	4.4510224
9	4.7019524
10	4.7499434
11	4.75804
12	4.7593749
13	4.7595942
14	4.7596301
15	...

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 aceeași ecuație polinomială ca și 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 =)

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

Metoda lui Halley

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.

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

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

$$\begin{array}{l} N := 20 \\ \text{---} \\ k := 0..N \end{array} \quad x_{k+1} := x_k - \frac{2f(x_k)}{f'(x_k) + \sqrt{f'(x_k)^2 - 2f(x_k) \cdot f''(x_k)}}$$

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

$$x_{\text{Halley}} := x$$

3.4. Rezolvare

$x_{\text{Halley}} := x$

$x_{\text{Halley}} =$

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 · 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 și se compară soluția cu soluția obținută prin metoda lui Newton.

$$x_{\text{Newton}} =$$

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

$$x_{\text{Halley}} =$$

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 · 10 ⁻⁵
7	4.7596372
8	4.7596372
9	4.7596372
10	4.7596372
11	4.7596372
12	4.7596372
13	4.7596372

Probleme propuse

Să se determine rădăcinile polinoamelor de mai jos folosindu-se metoda lui Newton, respectiv funcțiile predefinite `root` și `polyroots`:

Set 1.

a)	$P(x) = 5x^4 - 6x^3 + 4x - 8$	b)	$P(x) = 4x^4 - 3x^2 + 6x - 11$
c)	$P(x) = 6x^4 - 7x^2 + 8x - 15$	d)	$P(x) = 3x^6 - 4x^3 - 5x^2 + 9$
e)	$P(x) = 6x^5 + 5x^4 - 8x^2 + 24$	f)	$P(x) = 9x^3 - 7x^2 + 3x - 15$
g)	$P(x) = 6x^5 - 7x^3 + 8x - 2$	h)	$P(x) = 3x^7 - 4x^6 + 5x^5 - 25$

Set 2.

a)	$P(x) = 7x^5 - 6x^2 + 4x - 18$	b)	$P(x) = 13x^4 - 42x^3 + 25x^2 + 9$
c)	$P(x) = 12x^4 - 31x^2 + 6x - 11$	d)	$P(x) = 16x^5 - 27x^2 + 18x - 15$
e)	$P(x) = 19x^3 - 27x^4 + 13x - 5$	f)	$P(x) = 26x^5 + 5x^3 - 18x^2 + 24$
g)	$P(x) = 3x^5 - 14x^4 + 5x^3 - 41$	h)	$P(x) = 4x^5 - 17x^3 + 8x - 22$



REZOLVAREA APROXIMATIVĂ A ECUAȚIILOR ALGEBRICE ȘI TRANSCENDENTE – PARTEA 2

Dr.Ing. Levente CZUMBIL