



Metode Numerice – Lucrarea nr. 3

REZOLVAREA NUMERICĂ A ECUAȚIILOR ALGEBRICE ȘI TRANSCENDENTE – PARTEA II

Modelul matematic și metodele numerice utilizate

Metoda biseecției (a înjumătățirii intervalului)

Metoda biseecției este una din cele mai simple metode iterative de rezolvare a unei ecuații neliniare care se bazează pe observația că funcția $f(x)$ are semne contrare la capetele intervalul $[a, b]$ în interiorul căruia se află soluția, adică $f(a) \cdot f(b) < 0$. Se consideră deci ecuația de forma $f(x) = 0, x \in [a, b]$, pentru care s-a separat în prealabil o soluție în intervalul $[a, b]$, adică $f(a) \cdot f(b) < 0$. Știind că funcția este continuă pe intervalul $[a, b]$, se cere să se determine soluția în cauză. Caracteristic metodei biseecției este că, pornind de la intervalul $[a, b]$, la fiecare pas se restrânge domeniul în care se caută soluția, prin înjumătățirea intervalului de la pasul anterior, până la atingerea preciziei dorite. Aplicarea repetată a înjumătățirii intervalului de definiție a unei funcții, corespunzătoare unei ecuații a cărei soluție trebuie determinată, conduce la scăderea gradului de incertitudine al localizării soluției. Această repetare se realizează în contextul probării 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. Se pornește procesul 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.

Pașii enumerați anterior reprezintă un algoritm scris în pseudolimbaj, pe baza căruia poate fi implementat ușor un program de calcul. Limitele intervalului aflat în restrângere în jurul soluției sunt notate cu a_k respectiv b_k . Se observă că la fiecare valoare a lui k se schimbă una din limitele intervalului, după cum condiția de existență a soluției este verificată de prima sau a doua jumătate a intervalului partiționat. Vizualizarea schimbării acestor limite se arată în Fig. 1. Oricât de mult ar fi dusă restrângerea intervalului în jurul soluției, există posibilitatea ca valoarea considerată drept soluție să nu fie cea adevărată. Procesul iterativ de înjumătățire nu poate continua la infinit,

ci trebuie oprit după un anumit număr de partiționări determinat prin stabilirea unei erori limită între valoarea determinată ca și soluție și valoarea adevărată x .

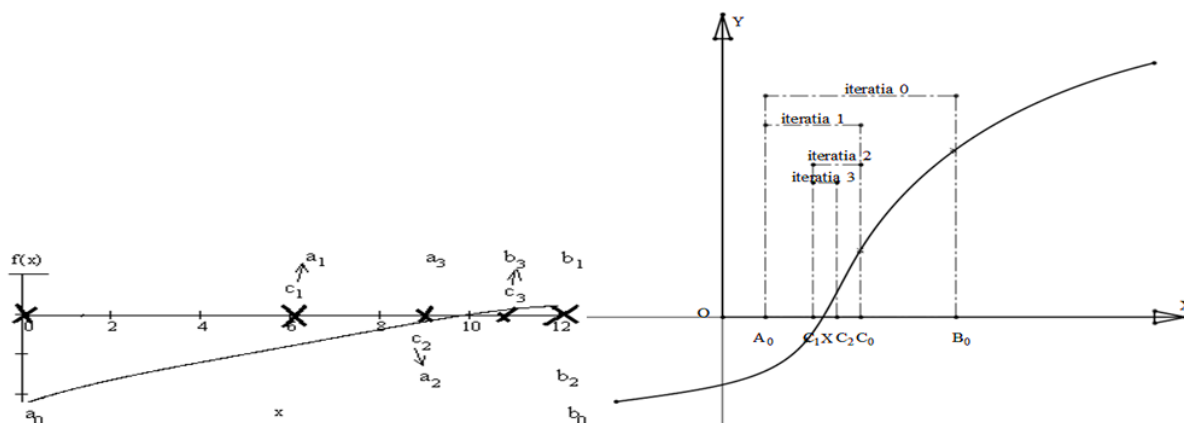


Fig. 1: Metoda Biseției

Eroarea de aproximare a soluției, la iterația k , este mai mică sau egală cu lungimea intervalului de căutare de la iterația k : $\varepsilon = |x - c_k| \leq (b_k - a_k)$, unde x este soluția exactă, a_k , b_k , c_k , sunt punctele de limită, respectiv de mijloc la iterația k . Lungimea intervalului de căutare se înjumătățește la fiecare iterație $(b_k - a_k) = \frac{1}{2}(b_{k-1} - a_{k-1})$, astfel, eroarea de aproximare a soluției la iterația k devine: $\varepsilon = (1/2)^k \cdot (b_0 - a_0)$.

Notând cu ε o limită de eroare absolută impusă și cu n numărul până la care a ajuns iterația k se poate determina o măsură destul de precisă a numărului minim de iterații necesare pentru calcularea unei soluții aproximative care să se încadreze într-o limită de eroare cunoscută:

$$n \geq \frac{\log(b-a) - \log(\varepsilon)}{\log(2)} \quad (1)$$

Astfel, în algoritmul prezentat mai sus, este posibilă adăugarea unei condiții de oprire a procesului când se atinge o anumită precizie, adică după ce se efectuează un număr de n iterații. Implementarea în limbaj de programare a acestei metode trebuie să se concretizeze în așa fel încât la fiecare iterație, în afară de prima, evaluarea funcției să fie făcută o singură dată.

Metoda lui Newton (tangentei)

Metoda Newton este una dintre cele mai eficiente metode iterative de rezolvare a ecuațiilor neliniare. Fie o ecuație de forma $f(x) = 0$, cu variabila $x \in [a, b]$ pentru care s-a separat în prealabil o soluție în intervalul $[a, b]$, adică $f(a) \cdot f(b) < 0$. Funcția este continuă și de două ori derivabilă pe intervalul dat și se dorește determinarea soluției în cauză. Dezvoltarea în serie Taylor în jurul unui punct x_k , în cazul în care se rețin doar primii doi termeni ai dezvoltării și restul, arată de forma:

$$f(x) = f(x_k) + (x - x_k) \cdot f'(x_k) + \frac{1}{2!} \cdot (x - x_k)^2 \cdot f''(\xi_k) \text{ cu } \xi_k \in [x; x_k] \quad (2)$$

Înlocuirea în expresia de mai sus, a unei aproximații x_{k+1} succesivă lui x_k , în locul lui x , pentru care presupunem că se anulează $f(x)$, neglijând restul (apare o eroare de metodă datorată

neglijării restului seriei Taylor care se bazează pe un calcul recursiv) și impunând (alegând) o aproximație inițială a soluției x_0 , ne conduce la o relație iterativă de determinare a soluției, care evidențiază faptul că la fiecare iterație graficul funcției este aproximat cu tangenta dusă în punctul $(x_k, f(x_k))$:

$$f(x) = 0 \Rightarrow x = x_k - \frac{f(x_k)}{f'(x_k)} - \frac{1}{2!} \cdot (x - x_k)^2 \cdot f''(\xi_k) / f'(x_k) \quad (3)$$

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (4)$$

Metoda lui Newton poate fi interpretată geometric ca o trasare repetată a tangentelor la funcție, prin punctele de aproximare a soluției, până când tangenta intersectează punctul în care funcția se anulează, ori un punct delimitat cu o anumită precizie de soluția exactă. Ecuația dreptei tangente în acest punct este:

$$y - f(x_k) = f'(x_k) \cdot (x - x_k) \quad (5)$$

Iar prin intersecția ei cu axa Ox, deci pentru $y = 0$ se obține relația care coincide cu relația (4):

$$x = x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (6)$$

Din acest motiv metoda lui Newton este cunoscută și sub numele de metoda tangentelor.

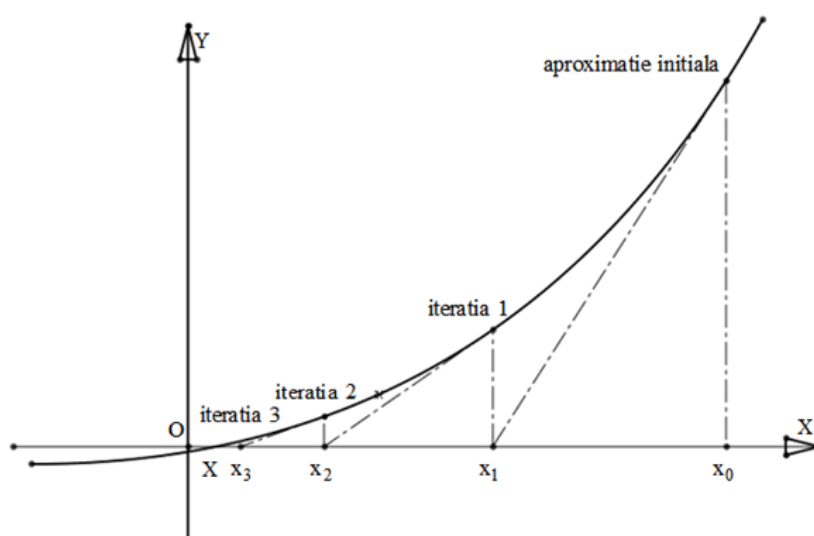


Fig. 2: Trasarea repetată a tangentelor la funcție

Procesul se continuă până când se ajunge, în limita unei precizii, la soluția considerată optimă. Algoritmul metodei se poate sintetiza astfel:

Pasul 1. Se inițializează soluția cu valoarea x_0 (aproximația inițială la iterația 0);

Pasul 2. La un pas oarecare k , $k = 1, 2, 3, \dots$, al procesului iterativ de calcul se determină $f(x_k)$ și $f'(x_k)$, noua valoare a soluției rezultând din relația $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$;

Pasul 3. Calculul se consideră terminat când la impunerea unei abateri maxim admisibilă a aproximației inițiale față de soluția adevărată, respectiv la impunerea unei erori de aproximare,



este îndeplinită condiția identificării unui număr minim de iterații necesare localizării soluției cu o anumită precizie impusă:

$$n \geq \log_2 \left[\frac{\log(\varepsilon) + \log(M)}{\log(M \cdot \varepsilon_{\text{admisibil}})} \right] \quad (7)$$

Valoarea acestui număr este condiționată de abaterea admisibilă impusă și de definirea numerică a rației M .

$$M = \frac{\max_{x \in I} (f''(x))}{2 \cdot \min_{x \in I} (f'(x))} < \infty \quad (8)$$

Forme îmbunătățite a Metodei lui Newton

Pentru a accelera convergența iterativă a soluției, pentru a crește precizia, sau pentru a reduce efortul de calcul din metoda inițială a lui Newton au fost dezvoltate o serie de forme îmbunătățite ale acesteia:

Metoda Newton – Kantorovici (metoda tangentelor paralele)

În cazul în care evaluarea derivatei funcției, în fiecare nou punct de aproximare, este costisitoare ca timp de calcul, formula lui Newton poate fi adaptată, în sensul reținerii valorii calculate a derivatei doar în primul punct de aproximare pe parcursul unui număr de iterații:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_0)} \quad (9)$$

Se reduce simțitor efortul computațional, cu dezavantajul reducerii concomitente a convergenței; sunt necesare mai multe iterații până la obținerea unei soluții dorite.

Metoda Newton discretă (metoda secantelor)

Dacă expresia derivatei funcției nu este cunoscută, atunci metoda lui Newton și varianta sa simplificată nu pot fi aplicate. În acest caz se face apel la aproximarea numerică a derivatei după care se deduce relația iterativă de determinare a soluției:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \approx x_k - f(x_k) \cdot \left[\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} \right] \quad (10)$$

Pentru aplicarea acestei metode trebuie cunoscute (alese) primele două aproximații inițiale x_0 și x_1 (pot fi alese de exemplu capetele intervalului $x_0 = a, x_1 = b$).

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

$$\begin{aligned} f(a) \cdot f(b) &< 0 \\ f(x_0) \cdot f''(x_0) &> 0 \\ f(x_0) \cdot f(x_1) &< 0 \end{aligned} \quad (11)$$

Observații: În funcție de tipul aplicației abordate trebuie să se realizeze un compromis între efortul de calcul implicat pentru atingerea unei precizii, convergență și precizie, dificultatea de evaluare repetată a funcției și a derivatelor ei. Trebuie să se facă apel pentru fiecare caracteristică aleasă la una din formulele expuse mai sus, ori la o formulă care să valorifice orientat sporirea preciziei, a convergenței, ori reducerea timpului de calcul.

Metoda lui Halley

Pentru a obține o convergență mai rapidă decât cea a metodei lui Newton se folosește o dezvoltare în serie Taylor a funcției atașate ecuației studiate, folosind primele 3 componente:

$$f(x) = f(x_0) + f'(x_0) \cdot (x - x_0) + \frac{1}{2} \cdot f''(x_0) \cdot (x - x_0)^2 \quad (12)$$

De aici se poate ajunge la formula iterativă de calcul:

$$x_{k+1} = x_k - \frac{2 \cdot f(x_k)}{f'(x_k) - \sqrt{f'(x_k)^2 - 2 \cdot f(x_k) \cdot f''(x_k)}} \quad (13)$$

Instrumente folosite

Algoritmul general al metodei biseției (înjumătățirii intervalului)

Relația (14) reprezintă algoritmul general de calcul al metodei biseției. Se observă că algoritmul se folosește de instrucțiunile de programare *for* și *if*, prezentate în **Lucrarea 2**. Indicii k , $k+1$ și 0 se introduc utilizând paranteza pătrată („[”], deoarece variabilele a , b , respectiv c sunt variabile de tip vector.

$\text{sol} :=$	$\text{for } k \in 0..n$	$\backslash\backslash$ se apeșază ciclul "for"
	$a_0 \leftarrow A_0$	$\backslash\backslash$ se initializează marginile
	$b_0 \leftarrow B_0$	primului interval în care se afla
		soluția
	$c_{k+1} \leftarrow \frac{a_k + b_k}{2}$	$\backslash\backslash$ formula de aproximare
		a soluției
	$a_{k+1} \leftarrow \text{if}(f(a_k) \cdot f(c_{k+1}) < 0, a_k, c_{k+1})$	$\backslash\backslash$ modificarea la fiecare iterație
		a capetelor intervalului aflat în
	$b_{k+1} \leftarrow \text{if}(f(b_k) \cdot f(c_{k+1}) < 0, b_k, c_{k+1})$	restrângere
		$\backslash\backslash$ memorarea în vectori
	$A_0 \wedge B_0$	a acestor margini
	c	$\backslash\backslash$ algoritmul returnează
		vectorul aproximațiilor soluției,
		care sunt de fapt înjumătățirile
		intervalului.

(14)

Operatorul de derivare

Algoritmul metodei Newton presupune utilizarea operației de derivare. Pentru utilizarea operatorului de derivare din *Mathcad*, este necesară apelarea toolbar-ului *Calculus*. Dacă toolbar-ul *Calculus* nu este vizibil, din meniul principal se bifează *View – Toolbars – Calculus* (a se vedea 3). Din toolbar-ul *Calculus* se selectează *Derivative*.

În fereastra de comandă se introduce numele funcției urmat de un apostrof (') după care se introduce un operator de atribuire (:=). În dreapta operatorului de derivare se introduce funcția de derivat.

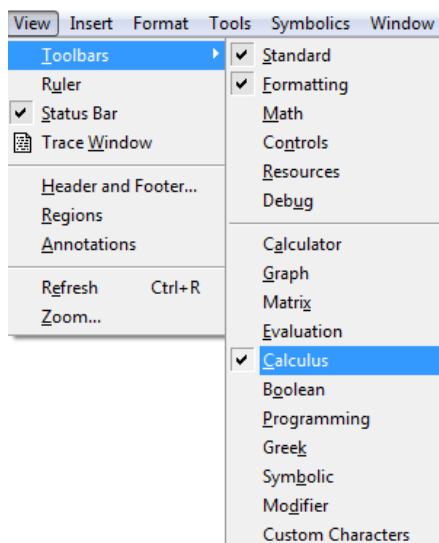


Fig. 3: Activarea toolbar-ului Calculus

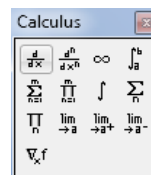


Fig. 4: Inserarea operatorului de derivare

$$f'(x) := \frac{d}{dx} f(x) \quad (15)$$

Operatorul de derivare de ordin superior

Algoritmul metodei secantei presupune utilizarea operației de derivare de ordinul 2. Din toolbar-ul *Calculus* se selectează *Nth Derivative* (a se vedea Fig. 5)

În fereastra de comandă se introduce numele funcției urmat de doi apostrofi (``) după care se introduce un operator de atribuire (:=).

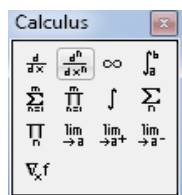


Fig. 5: Inserarea operatorului de derivare de ordin superior

$$f''(x) := \frac{d^2}{dx^2} f(x) \quad (16)$$