

Soluția ecuației neliniare $f(x) = 0$. Rădăcinile polinoamelor. Lucrul cu polinoame în Octave. Rezolvarea sistemelor de ecuații neliniare

Noțiuni teoretice

Soluția ecuației neliniare $f(x) = 0$

Vom studia două tipuri de metode:

a) Metode bazate pe interval

Se pornește de la observația că o funcție își schimbă semnul în vecinătatea unei rădăcini. Inițial, se consideră două valori de o parte și de cealaltă a rădăcinii, apoi se micșorează acest interval care încadrează rădăcina până ce se ajunge la rădăcină, cu o anumită precizie. Metodele de acest tip sunt întotdeauna convergente către soluție deoarece aplicarea repetată a algoritmului duce la o estimare mai precisă a rădăcinii.

În continuare vom detalia două metode bazate pe interval: metoda biseției și metoda secantei.

Metoda biseției

Dacă o funcție își schimbă semnul pe un interval, adică $f(a) \cdot f(b) < 0$, atunci se evaluează valoarea funcției în punctul de mijloc al intervalului, $c = \frac{a+b}{2}$. Localizarea rădăcinii se estimează ca fiind la mijlocul subintervalului în care funcția își schimbă semnul, noul subinterval având la unul dintre capete pe c . Procedura se repetă până ce se obțin estimări mai precise ale rădăcinii.

Se poate defini eroarea relativă procentuală folosind relația:

$$\epsilon = \left| \frac{x_r^{new} - x_r^{old}}{x_r^{new}} \right|$$

Când $\epsilon < prag$, algoritmul iterativ se termină, iar x_r^{new} este considerată valoarea calculată a rădăcinii.

Un alt avantaj al acestei metode este faptul că pentru o eroare acceptată tol , se poate calcula numărul de iterații ce trebuie parcurse pentru a se ajunge la aproximarea dorită a rădăcinii, conform formulei:

$$n = \log_2\left(\frac{b-a}{tol}\right).$$

Metoda secantei

În această metodă, aproximarea rădăcinii, în intervalul $[a_i, b_i]$, se consideră a fi intersecția dreptei care trece prin punctele $(a_i, f(a_i))$ și $(b_i, f(b_i))$ cu axa Ox , adică:

$$x_{i+1} = \frac{a_i f(b_i) - b_i f(a_i)}{f(b_i) - f(a_i)}.$$

Apoi, se consideră subintervalul care are la unul dintre capete pe x_{i+1} , în manieră similară cu metoda bisecției. Același criteriu de oprire: eroarea relativă procentuală ϵ , poate fi folosit.

b) Metode care nu se bazează pe un interval, în care este suficientă cunoașterea unei valori inițiale x_i care este folosită mai departe pentru estimarea valorii următoare x_{i+1} . Aceste metode, spre deosebire de primele, pot fi convergente sau pot fi divergente. Atunci când ele converg, convergența este mult mai rapidă decât în cazul metodelor bazate pe interval. Pentru această categorie, menționăm metoda tangentei (Newton-Raphson) și metoda aproximațiilor succesive (contractției).

Metoda tangentei

După cum s-a menționat anterior, se pornește cu o valoare de început x_i , apoi se deduce o estimare îmbunătățită x_{i+1} . În cazul metodei tangentei, se duce o tangentă la curba din punctul de coordonate $[x_i, f(x_i)]$. Punctul de intersecție a tangentei cu Ox se consideră x_{i+1} .

Relația de recurență este:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Se poate arăta că eroarea la iterația curentă este proporțională cu pătratul erorii la iterația precedentă, ceea ce, aproximativ, înseamnă că la fiecare iterație numărul de zecimale corect calculate din rădăcină se dublează.

Metoda aproximațiilor succesive

În cazul acestei metode, ecuația $f(x) = 0$ se rescrie ca $x = g(x)$, ceea ce are avantajul de a furniza o formulă prin care se poate calcula o nouă valoare a lui x ca o funcție g aplicată vechii valori a lui x , adică:

$$x_{i+1} = g(x_i)$$

Această metodă converge liniar (eroarea la pasul i este proporțională cu eroarea la pasul $i - 1$ înmulțită cu un factor subunitar) dacă $|g'(x)| < 1, \forall x \in (a, b)$. Altfel, metoda este divergentă.

Rădăcinile polinoamelor

Fie p un polinom de grad n , $p(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \dots + a_1 x + a_0$, cu $a_n \neq 0$. Conform cu *teorema fundamentală a algebrei*, polinomul p are n rădăcini reale sau complexe (numărând și multiplicitățile). În cazul în care coeficienții a_i sunt toți reali, rădăcinile complexe apar conjugate (de forma $c + id$ și $c - id$).

Folosind regula semnelor a lui Descartes, putem număra câte rădăcini reale pozitive are polinomul p . Fie v numărul variațiilor de semn ale coeficienților

$$a_n, a_{n-1}, a_{n-2}, \dots, a_1, a_0,$$

ignorând coeficienții care sunt nuli. Fie n_p numărul de rădăcini pozitive. Avem următoarele două relații:

- 1) $n_p \leq v$;
- 2) $v - n_p$ este un număr par.

Analog, numărul de rădăcini reale negative ale lui $p(x)$ se obține folosind numărul de schimbări de semn ale coeficienților polinomului $p(-x)$. Pentru determinarea rădăcinilor se pot aplica metodele descrise anterior la punctul b), dacă nu se cunoaște localizarea rădăcinilor pe intervale.

Lucrul cu polinoame în Octave

În Octave, un polinom este reprezentat prin coeficienți (în ordine descrescătoare). Vectorul

```
> p = [-2, -1, 0, 1, 2]
```

reprezintă polinomul $-2x^4 - x^3 + x + 2$.

Funcția *polyout* generează o reprezentare a funcției de o variabilă dată (de exemplu x)

```
> polyout(p, 'x')
```

are ca rezultat $-2 * x^4 - 1 * x^3 + 0 * x^2 + 1 * x^1 + 2$.

Evaluarea unui polinom:

```
> y = polyval(p, x)
```

returnează $p(x)$. Dacă x este vector sau matrice, polinomul este evaluat în fiecare dintre elementele lui x .

Înmulțirea a două polinoame:

```
> r = conv(p, q)
```

returnează un vector r care conține coeficienții produsului dintre p și q .

Împărțire a două polinoame:

```
> [b, r] = deconv(y, a)
```

returnează coeficienții polinoamelor b și r a.î. $y = ab + r$, unde b este câtul și r este restul împărțirii.

Rădăcinile unui polinom:

```
> roots(p)
```

Soluția ecuației neliniare $f(x) = 0$. Rădăcinile polinoamelor. Lucrul cu polinoame în Octave. Rezolvarea sistemelor de ecuații neliniare

returnează un vector ce conține toate rădăcinile polinomului p .

Derivata unui polinom:

```
> q = polyder(p)
```

returnează un vector ce conține coeficienții derivatei polinomului p .

Integrarea unui polinom:

```
> q = polyint(p)
```

returnează un vector ce conține coeficienții rezultați din integrarea polinomului p .

Polinomul de interpolare de gradul n care aproximează setul de date (x, y) :

```
> p = polyfit(x, y, n)
```

Exemplu: Adunarea polinoamelor

Presupunem că dorim să adunăm polinoamele $p(x) = x^2 - 1$ și $q(x) = x + 1$. Următoarea încercare va da eroare:

```
> p = [1, 0, -1];
> q = [1, 1];
> "p + q error: operator +: nonconformant arguments
(op1 is 1x3, op2 is 1x2)
error: evaluating binary operator '+' near line 22, column 3"
```

Operația de adunare a doi vectori de dimensiuni diferite în Octave dă eroare. Pentru a evita aceasta, trebuie adăugate zerouri la q .

```
> q = [0, 1, 1];
> p + q
ans = 1 1 0
> polyout(ans, 'x')
1*x^2 + 1*x^1 + 0
```

Sisteme de ecuații neliniare

Un sistem de ecuații neliniare are forma:

$$\begin{cases} f_1(x_1, x_2, x_3, \dots, x_{n-1}, x_n) = 0 \\ f_2(x_1, x_2, x_3, \dots, x_{n-1}, x_n) = 0 \\ \dots \\ f_n(x_1, x_2, x_3, \dots, x_{n-1}, x_n) = 0 \end{cases}$$

unde f_i reprezintă funcții cunoscute de n variabile x_1, x_2, \dots, x_n , presupuse continue, împreună cu derivatele lor parțiale până la un ordin convenabil (de obicei, până la ordinul doi). Se va urmări găsirea soluțiilor reale ale sistemului într-un anumit domeniu de interes, domeniu în care se consideră valabile proprietățile de continuitate impuse funcțiilor f_i și derivatelor lor. Rezolvarea sistemului este un proces iterativ în care se pornește de la o aproximație inițială pe care algoritmul o va îmbunătăți până ce se va îndeplini o condiție de convergență. În cazul de față, localizarea apriori a soluției nu mai este posibilă (nu există o metodă analoagă metodei înjumătățirii intervalului).

Metoda Newton

Pentru simplificarea notației, considerăm $F = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \dots \\ f_n \end{pmatrix}$ și $x = (x_1, x_2, \dots, x_n)$.

Sistemul îl putem rescrie ca $F(x) = 0$. Notăm cu $x^{(k)}$ estimarea la pasul k a soluției x^* , deci $F(x^*) = 0$.

Se poate deduce relația:

$$x^{(k+1)} = x^{(k)} - J^{-1}F(x^{(k)}), k = 0, 1, 2, \dots$$

unde J este matricea *Jacobiană*

$$J = \begin{pmatrix} \frac{\partial F_1}{\partial x_1} & \dots & \frac{\partial F_1}{\partial x_n} \\ \dots & \dots & \dots \\ \frac{\partial F_n}{\partial x_1} & \dots & \frac{\partial F_n}{\partial x_n} \end{pmatrix}$$

Dacă matricea J este neinvertibilă, atunci pasul este nedefinit. Vom presupune că $J(x^*)$ este inversabilă, iar continuitatea lui J va asigura că $J(x^{(k)})$ este inversabilă pentru orice $x^{(k)}$ suficient de apropiat de x^* . Secvența definită iterativ converge spre soluția x^* . Condiția de oprire la iterația k , $\|x^* - x^{(k)}\| < tol$, unde tol este o toleranță dată, se poate arăta că revine la $\|x^{(k)} - x^{(k-1)}\| < tol$.

Probleme propuse

Problema 1

Scrieți o funcție Octave care să rezolve o ecuație neliniară folosind metoda biseției. Parametrii pe care îi primește funcția sunt: a , b (capetele intervalului în care se află soluția reală), tol (toleranța), respectiv un handler la funcția care definește ecuația neliniară de rezolvat.

Problema 2

Scrieți o funcție Octave care să rezolve o ecuație neliniară folosind metoda secantei. Parametrii pe care îi primește funcția sunt: a , b (capetele intervalului în care se află soluția reală), tol (toleranța), respectiv un handler la funcția care definește ecuația de rezolvat.

Problema 3

Scrieți o funcție Octave care să rezolve o ecuație neliniară prin metoda tangentei. Parametrii pe care îi primește funcția sunt: x_0 (aproximația inițială a soluției), tol (toleranța), respectiv un handler la funcția care definește ecuația de rezolvat și un handler la derivata acesteia.

Problema 4

Scrieți o funcție care să rezolve o ecuație neliniară prin metoda aproximațiilor succesive. Parametrii pe care îi primește funcția sunt: x_0 (aproximația inițială a soluției), tol (toleranța), respectiv un handler la funcția care definește ecuația de rezolvat.

Problema 5

Să se determine tipul (dacă sunt reale pozitive sau negative, complexe) rădăcinilor polinomului $p(x) = 3x^6 + x^4 - 2x^3 - 5$.

Problema 6

Să se scrie în Octave un program care rezolvă un sistem de ecuații neliniare prin metoda Newton. Ca intrare, se consideră un vector coloană care reprezintă $x^{(0)}$, un pointer (handler) la o funcție care evaluează F într-un vector generic x , un pointer la o funcție care calculează Jacobiana într-un vector generic x , o toleranță dată ϵ . Metoda se oprește atunci când $\|x^{(k)} - x^{(k-1)}\| < \epsilon$ și returnează vectorul soluție x^* și numărul de iterații n care au fost necesare pentru producerea soluției.