METODE NUMERICE: Laborator #6
Soluția ecuației neliniare $f(x) = 0$. Rădăcinile polinoamelor. Lucrul cu polinoame în Octave. Rezolvarea sistemelor neliniare.

Titulari curs: *Florin Pop, George-Pantelimon Popescu*

Responsabil Laborator: **Sorin N. Ciolofan**

Obiective Laborator

Insusirea notiunilor privitoare la determinarea aproximativa a solutiilor ecuatiei neliniare. Rezolvarea iterativa a unui sistem de ecuatii neliniare. Comenzi Octave pentru lucrul cu polinoame. Radacinile polinoamelor.

Solutia ecuatiei neliniare f(x)=0

Vom studia doua tipuri de metode:
a) Metode bazate pe interval
Se porneste de la observatia ca o functie isi schimba semnul in vecinatatea unei radacini, se "ghicesc" doua valori de o parte si cealalta a radacinii, apoi se micsoreaza acest interval care incadreaza radacina pina ce se ajunge la radacina, cu o anumita precizie. Metodele de acest tip sunt intodeauna convergente catre solutia dorita, deoarece aplicarea repetata a algoritmului duce la o estimare mai precisa a radacinii. Printre metode mentionam metoda bisectiei (Bolzano) si metoda pozitiei false (in latina "regula falsi").
b) Metode "deschise" in care e suficienta cunoasterea unei valori initale x_i care este folosita mai departe pentru estimarea valorii urmatoare x_{i+1}. Aceste metode, spre deosebire de primele, pot sa fie convergente sau pot fi divergente. Atunci cind ele converg, convergenta este mult mai rapida decit in cazul metodelor descrise la a). In aceasta categorie mentionam metoda tangentei (Newton-Raphson) si metoda contractiei.

a1) Metoda bisectiei
Daca o functie isi schimba semnul pe un interval, $f(a) \cdot f(b) < 0$, atunci se evalueaza valoarea functiei in punctul de mijloc al intervalului, $c = \frac{a+b}{2}$. Locatia radacinii se estimeaza ca fiind la mijlocul subintervalului pe care functia isi schimba semnul, noul subinterval avind la unul din capete pe c. Procedura se repeta pina ce se obtin estimari mai precise ale radacinii.
Criteriul de estimare a erorii si de oprire a calculului - se poate defini eroarea relativa procentuala cu relatia $\epsilon = \left| \frac{x_r^{new} - x_r^{old}}{x_r^{new}} \right|$
Cind $\epsilon < prag$, de ex. 0.1 %, atunci iteratiile sunt oprite iar x_r^{new}este decretata valoarea calculata a radacinii. Un alt avantaj notabil al acestei metode este ca, fiind data o eroare acceptata (toleranta) tol, se poate calcula numarul de iteratii care sunt necesarare sa se parcurga pentru a ajunge la aproximarea dorita a radacinii, conform formulei n=$\log_2(\frac{b-a}{tol})$

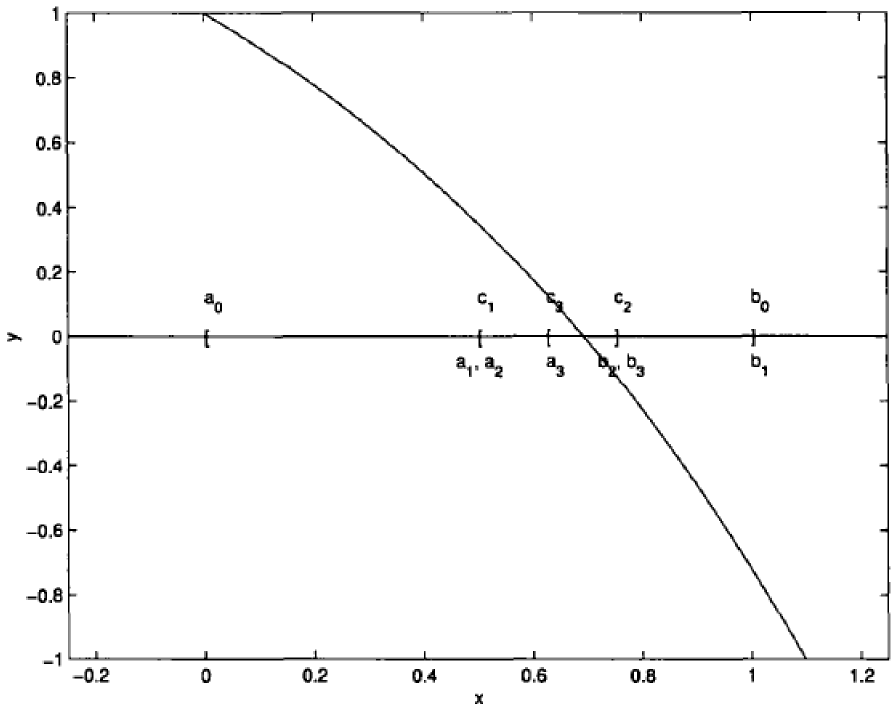

Figure 1: metoda bisectiei

In Figura1 de mai jos este reprezentata metoda bisectiei pentru f(x)=2-e^x

a2) Metoda pozitiei false (regula falsi)

In aceasta varianta, curba f(x) este aproximata cu dreapta care trece prin punctele f(a) si f(b). Din acest motiv mai poarta si numele de "metoda de interpolare liniara". Regula falsi apare mentionata in papirusurile din Egiptul antic (1650 i.Chr).

Aproximarea radacinii se considera a fi intersectia acestei drepte cu axa Ox (Figura 2), respectiv $c = b - \frac{f(b)(a-b)}{f(a)-f(b)}$. Apoi se considera subintervalul care are la unul din capete pe c, in maniera similara cu metoda bisectiei. Acelasi criteriu de oprire, eroarea relativa procentuala ϵ poate fi folosit. Se observa cum sirul de aproximari p_2, p_3, p_4 ,... converge spre radacina.

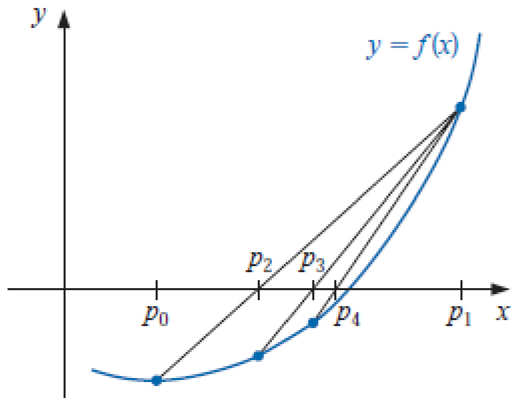

Figure 2: metoda pozitiei false

b1) Metoda tangentei

Dupa cum s-a mentionat anterior, se porneste cu o valoare de inceput x_i apoi se deduce o estimare imbunatatita, x_{i+1}. In cazul metodei tangentei (Newton-Raphson) se duce o tangenta la curba din punctul de coordonate [x_i,f(x_i)]. Punctul de intersectie a tangentei cu Ox se considera x_{i+1} (Figura 3).

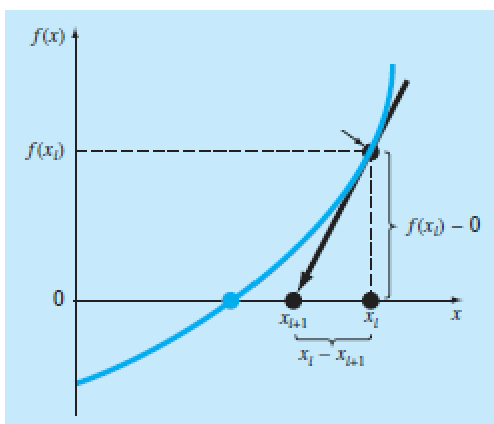

Figure 3: metoda tangentei

Relatia de recurenta este:

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

Se poate arata ca eroarea la iteratia curenta este proportionala cu patratul erorii la iteratia precedenta, ceea ce, aproximativ, s-ar traduce prin faptul ca la fiecare iteratie numarul de zecimale corect calculate din radacina se dubleaza.

b2) Metoda substitutiei succesive

In cadrul acestei metode, ecuatia f(x)=0 se rescrie ca x=g(x) ceea ce are avantajul de a furniza o formula prin care se poate calcula o noua valoare a lui x ca o functie (g()) aplicata vechii valori a lui x.

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

Aceasta metoda converge liniar (eorarea la pasul i este proportionala cu eroarea la pasul i-1 inmultita cu un factor subunitar, de ex. 0.5) daca $\left|g^{'}(x)\right| < 1; \forall x \in (a, b)$. Altfel, metoda este divergenta.

Radacinile polinoamelor

Fie p un polinom de grad n, p(x)= $a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + ... + a_1 x + a_0,\ cu\ a_n \neq 0.$
Conform cu Teorema fundamentala a Algebrei, p are n radacini reale sau complexe (numaraind si multiplicitatile). Daca coeficintii a_i sunt toti reali, atunci radacinile complexe apar conjugate (de forma c+di si c-di).
Folosind regula semnelor a lui Descartes putem numara cite radacini reale pozitive are p.
Fie v numarul variatiilor de semn ale coeficientilor $a_n, a_{n-1}, a_{n-2}, .., a_1, a_0$ ignorind coeficientii care sunt nuli.
Fie n_p numarul de radacini pozitive. Avem urmatoarele doua relatii:
1)$n_p \leq v$
2) v-n_p este numar par
Analog, numarul de radacini reale negative al lui p(x) se obtine folosind numarul de schimbari de semn ale coeficientilor lui p(−x). Pentru determinarea radacinilor se pot aplica metodele b1) si b2) descrise anterior, daca nu se cunoaste localizarea radacinilor pe intervale.

Lucrul cu polinoame in Octave

In Octave, un polinom e reprezentat prin coeficientii sai (in ordine descrescatoare) . Vectorul

```
octave:1> p = [-2, -1, 0, 1, 2];
```

reprezinta polinomul $-2x^4 - x^3 + x + 2$
Functia polyout genereaza o reprezentare functie de o variabila data (de ex. ‘x’)

```
octave:2> polyout(p,'x')
 -2*x^4 - 1*x^3 + 0*x^2 + 1*x^1 + 2
```

Evaluarea unui polinom se face cu:

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

Returneaza p(x). Daca x e vector sau matrice, polinomul e evaluat in fiecare din elementele lui x.
Inmultire:

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

Returneaza un vector r care contine coeficientii produsului dintre p si q
Impartire:

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

Returneaza ceoficientii polinoamelor b si r a.i y=ab+r unde b este citul si r este restul impartirii
Radacini:

```
roots(p)
```

Returneaza un vector ce contine toate radacinile lui p
Derivata:

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

q contine coeficientii derivatei lui p
Integrare:

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

Polinom de interpolare de gradul n care aproximeaza setul de date (x,y):

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

Exemplu: Adunarea polinoamelor
Presupunem ca dorim sa adunam p(x)=$x^2 - 1$ si q(x)=x+1.
Urmatoarea incercare va da eroare

```
octave:1> p = [1, 0, -1];
octave:2> q = [1, 1];
octave:3> "p + q error: operator +: nonconformant arguments (op1 is 1x3, op2 is 1x2)
error:evaluating binary operator `+' near line 22, column 3"
```

Operatia de adunare a doi vectori de dimensiuni diferite in Octave da eroare. Pentru a evita aceasta trebuie adaugate niste zerouri la q.

```
octave:4> q = [0, 1, 1];
octave:5> p + q
ans = 1 1 0
octave:6> polyout(ans, 'x')
1*x^2 + 1*x^1 + 0
```

Sisteme neliniare de ecuatii

Un sistem de ecuatii neliniare are forma:

$f_1(x_1, x_2, x_3, ..., x_{n-1}, x_n) = 0;$
$f_2(x_1, x_2, x_3, ..., x_{n-1}, x_n) = 0;$
...
$f_n(x_1, x_2, x_3, ..., x_{n-1}, x_n) = 0;$

unde f_i reprezinta functii cunoscute de n variabile $x_1, x_2, ..., x_n$, presupuse continue, impreuna cu derivatele lor partiale pana la un ordin convenabil (de obicei, pana la ordinul doi). Se va urmari gasirea solutiilor reale ale sistemului ıntr-un anumit domeniu de interes, domeniu ın care se considera valabile proprietatile de continuitate impuse functiilor f_i si derivatelor lor. Rezolvarea sistemului este un proces iterativ in care se porneste de la o aproximatie initiala pe care algoritmul o va imbunatati pana ce se va indeplini o conditie de convergenta. In cazul de fata localizarea apriori a solutiei nu mai este posibila (nu mai exista o meteoda analoaga metodei injumatatirii intervalelor in acest caz).

Metoda Newton

Pentru simplificarea notatiei consideram $F = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ ... \\ f_n \end{pmatrix}$ si $x = (x_1, x_2, ..., x_n)$. Sistemul il putem rescrie ca F(x)=0. Notam cu $x^{(k)}$ estimarea la pasul k a solutiei x^*, deci F(x^*)=0.
Se poate deduce relatia $x^{(k+1)} = x^{(k)} - J^{-1}F(x^{(k)})$ pt. k=0,1,2,... (I)
unde J este matricea Jacobiana (Carl Gustav Jacob Jacobi, 1804-1851)
$J = \begin{pmatrix} \frac{\partial F_1}{\partial x_1} & ... & \frac{\partial F_1}{\partial x_n} \\ & ... & \\ \frac{\partial F_n}{\partial x_1} & ... & \frac{\partial F_n}{\partial x_n} \end{pmatrix}$
Daca J este neinversabila atunci pasul este nedefinit. Vom presupune ca J(x*) este inversabila iar continuitatea lui J va asigura ca $J(x^{(k)})$ este inversabila pentru orice $x^{(k)}$ suficient de apropiat de x*. Secventa definita iterativ la (I) converge spre solutia x*. Conditia de oprire la iteratia k, $\left\|x^* - x^{(k)}\right\| < tol$ unde tol e o toleranta data, se poate arata ca revine la $\left\|x^{(k)} - x^{(k-1)}\right\| < tol$

Exemplu:
Consideram sistemul neliniar cu doua necunoscute
$x_1^2 + x_2^2 - 1 = 0$
$x_2 - x_1^2 = 0$

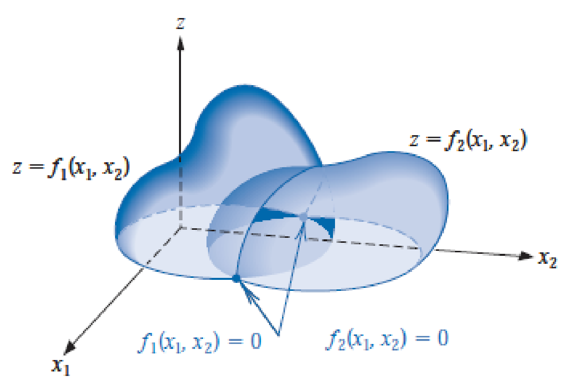

Figure 4: interpretarea geometrica a unui sistem neliniar cu 2 necunoscute

a carui interpretare geometrica este prezentata in Figura4.
O prima solutie este x*= $(\sqrt{\frac{\sqrt{5}-1}{2}}, \frac{\sqrt{5}-1}{2})$ iar a doua este simetrica lui x* fata de axa Ox_2. Aplicind algoritmul Newton pentru k=5 iteratii si pornind cu $x^{(0)} = (0.5, 0.5)$ obtinem valorile din tabelul de mai jos (calculele au fost efectuate in aritmetica IEEE cu dubla precizie, ceea ce corespunde la 16 zecimale precizie).

k	$\|x^* - x^{(k)}\|$	$\|F(x^{(k)})\|$
0	$3.0954 \cdot 10^{-1}$	$5.5902 \cdot 10^{-1}$
1	$8.9121 \cdot 10^{-2}$	$2.1021 \cdot 10^{-1}$
2	$4.5233 \cdot 10^{-3}$	$1.0090 \cdot 10^{-2}$
3	$1.2938 \cdot 10^{-5}$	$2.8769 \cdot 10^{-5}$
4	$1.0646 \cdot 10^{-10}$	$2.3673 \cdot 10^{-10}$
5	0.0000	$1.1102 \cdot 10^{-16}$

Figure 5: convergenta metodei Newton

Exercitii propuse

1) Sa se rezolve ecuatia tg(x)-2x=0 pe intervalul [0.1,1.5] utilizind metoda bisectiei.
a) Creati un fisier f.m care calculeaza f(x) intr-un punct x dat ca parametru. Creati apoi sample.m care intoarce un vector cu valori ale functiei calculate in suficient de multe puncte a.i sa se traseze un grafic pe intervalul considerat folosind functia plot (pentru detalii, *help plot* in linia de comanda Octave).
b) Creati un fisier bisect.m care sa contina o functie ce identifica o radacina prin metoda bisectiei. Metoda primeste ca parametru pe linga capetele intervalului, o toleranta acceptata, tol (se poate testa cu tol=0.000001).

2) Sa se gaseasca manual toate solutiile ecuatiei 5x+ln(x)=10000 cu minimum 4 zecimale exacte folosind metoda Newton-Raphson. (Se poate folosi un calculator stiintific pentru ajutor).

3) Sa se determine tipul (daca sunt reale pozitive sau negative, complexe) radacinilor polinomului p(x)=$x^4 + 2x^2 - x - 1$.

4) Sa se scrie in Octave un program care rezolva un sistem neliniar de ecuatii prin metoda Newton (descrisa la pag.5). Ca intrare se considera un vector coloana care reprezinta $x^{(0)}$, un pointer (handler) la o functie care evalueaza F intr-un vector generic x, un pointer la o functie care calculeaza Jacobiana intr-un vector generic x, o toleranta data ϵ. Metoda se opreste atunci cind $\left\|x^{(k)} - x^{(k-1)}\right\| < \epsilon$ si returneaza vectorul solutie x^* si numarul de iteratii n care au fost necesare pentru producerea solutiei.