MARILENA POPA ROMULUS MILITARU

METODE NUMERICE

Note de curs

1. REZOLVAREA NUMERICĂ A SISTEMELOR DE ECUAȚII LINIARE

Introducere. Rezolvarea sistemelor algebrice liniare şi operațiile de calcul matriceal (evaluarea determinanților, inversarea matriceală, calculul valorilor şi vectorilor proprii) sunt incluse în domeniul algebrei liniare – implicată în diverse probleme ştiințifice, de exemplu:

– problemele care depind de un număr finit de grade de libertate, reprezentate prin ecuații diferențiale ordinare sau cu derivate parțiale sunt transformate, cu ajutorul diferențelor finite, în sisteme de ecuații liniare;

– problemele neliniare sunt frecvent soluționate (aproximate) prin procese de liniarizare;

– programarea liniară implică rezolvarea unor sisteme de ecuații algebrice liniare;

– foarte multe probleme inginereşti din domeniul rețelelor electrice, analiza structurilor, proiectarea clădirilor, vapoarelor, avioanelor, transportul lichidelor şi gazelor prin conducte etc. necesită, pentru soluționare, rezolvarea unor sisteme liniare.

Formularea problemei. Fie $A \in R^{nxn}$ (o matrice reală cu n linii şi n coloane), $b \in R^n$ (un vector – matrice coloană – cu n componente reale) şi vectorul necunoscut $x \in R^n$. Atunci un sistem de ecuații liniare se scrie sub una din formele:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ ..... \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n = b_n \end{cases}$$

sau matriceal: Ax = b sub formă compactă şi

$$\begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{n1} & a_{n2} & ... & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \qquad (1)$$

sub formă dezvoltată.

O ultimă variantă de scriere a unui sistem liniar este:

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \ , \ i = 1, 2, ..., n$$

Observații.

1. Un sistem liniar (1) este **consistent** dacă are cel puțin o soluție şi **inconsistent** dacă nu are nici o soluție. Orice sistem liniar considerat în continuare poate avea soluție unică (compatibil determinat), nici o soluție (incompatibil) sau o infinitate de soluții (compatibil nedeterminat) – nu există alte posibilități. Bineînțeles că interesează primul caz.

2. Sistemele (1) se pot clasifica şi după vectorul b, în:

a) **sisteme omogene** – dacă b = 0. Orice sistem omogen de forma Ax = 0 este un sistem consistent (deoarece are soluția x = 0 – neinteresantă). Un sistem omogen are o soluție nebanală dacă şi numai dacă det A = 0 (adică A este singulară). În această situație soluția depinde de cel puțin un parametru.

b) **sisteme neomogene** – dacă b $\neq$ 0. Sistemul (1) este compatibil determinat pentru ($\forall$) b $\neq$ 0 dacă şi numai dacă sistemul omogen Ax=0 nu are decât soluția banală (adică A este nesingulară).

3. În sistemele obținute din aplicațiile fizice, numerele ce constituie matricea A şi vectorul b sunt afectate de erori inerente (provenite din măsurători) sau erori de rotunjire (1/7 nu poate fi reprezentat exact pe nici un calculator electronic – care are lungime fixă a cuvântului – deoarece reprezentarea lui în binar are o infinitate de biți). Dacă erorile mici în cadrul coeficienților lui A şi b, sau în procesul de calcul au un efect redus asupra vectorului, soluție, un astfel de sistem este **bine condiționat**, iar dacă efectul este considerabil, un astfel de sistem este **slab condiționat**.

Metodele numerice de rezolvare a sistemului liniar (1) sunt de două tipuri: directe şi iterative.

Rezolvarea directă a sistemelor de ecuații liniare

Metodele directe constau în reducerea sistemului (1), într-un număr finit de etape, la un sistem **echivalent**, (cu aceeaşi soluție) direct rezolvabil – prin substituție directă sau inversă.

Aceste metode furnizează soluția exactă x a sistemului (1) în cazurile (ideale) în care erorile de rotunjire sunt absente şi necesită, în

acest scop, efectuarea unui număr de operații aritmetice elementare de ordinul n^3.

Din acest motiv, metodele directe se utilizează pentru rezolvarea sistemelor „uzuale“, de dimensiune $n \leq 100$.

În continuare vom prezenta câteva metode directe de rezolvare a sistemelor de ecuații liniare.

1.1. Metoda Gauss. Tehnici de pivotare

Procedura de reducere a sistemului (1) la un sistem echivalent are la bază o schemă de eliminare succesivă a necunoscutelor, introdusă de Gauss în 1823, prin care se efectuează triangularizarea superioară a matricei sistemului prin transformări elementare.

Transformările elementare din metoda Gauss sunt de tipul:

- schimbarea a două linii între ele;
- adunarea unei linii, înmulțite cu un număr, la altă linie

şi se efectuează asupra matricei extinse (A | b) astfel încât în „n-1“ etape în locul matricei A să avem o matrice superior triunghiulară.

Observație. Transformările elementare de tipul celor de mai sus sunt permise numai asupra liniilor deoarece nu influențează obținerea soluției (se referă la ecuațiile sistemului care prin schimbare sau adunare – multiplicate cu un număr – nu influențează soluția).

Încercăm prezentarea algoritmică (pe paşi) a metodei Gauss:

- la primul pas se foloseşte prima ecuație a sistemului la eliminarea necunoscutei x_1 din celelalte n-1 ecuații obținându-se un nou sistem $A^{(2)}x = b^{(2)}$ (unde $A^{(1)} = A$ şi $b^{(1)} = b$);
- la al doilea pas se foloseşte a doua ecuație transformată (deci din sistemul anterior $A^{(2)}x = b^{(2)}$) pentru eliminarea necunoscutei x_2 din ultimele n-2 ecuații, obținându-se sistemul $A^{(3)}x = b^{(3)}$ ş.a.m.d.

Astfel, pentru k = 1, 2, ... , n-1, avem:

$$a_{ij}^{(k+1)} = \begin{cases} a_{ij}^{(k)}, 1 \le i \le k,\ i \le j \le n \\ 0, 1 \le j \le k,\ j+1 \le i \le n \\ a_{ij}^{(k)} - \dfrac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \cdot a_{kj}^{(k)},\ k+1 \le i, j \le n \end{cases}$$

$$b_i^{(k+1)} = \begin{cases} b_i^{(k)}, 1 \le i \le k \\ b_i^{(k)} - \dfrac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \cdot b_k^{(k)},\ k+1 \le i \le n \end{cases} \qquad (2)$$

Pentru k = n-1, înseamnă că necunoscuta x_{n-1} a fost eliminată din ultima ecuație, obținându-se un sistem sub formă triunghiulară.

$$\begin{cases} a_{11}^{(n)}x_1 + a_{12}^{(n)}x_2 + \ldots + a_{1k}^{(n)}x_k + \ldots + a_{1n}^{(n)}x_n = b_1^{(n)} \\ a_{22}^{(n)}x_2 + \ldots + a_{2k}^{(n)}x_k + \ldots + a_{2n}^{(n)}x_n = b_2^{(n)} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{kk}^{(n)}x_k + \ldots + a_{kn}^{(n)}x_n = b_k^{(n)} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ a_{nn}^{(n)}x_n = b_n^{(n)} \end{cases} \qquad (3)$$

sau

$A^{(n)}x = b^{(n)}$

Observații:

1. $A^{(n)}$ este o matrice superior triunghiulară.
2. Sistemul $A^{(n)}x = b^{(n)}$ este echivalent cu sistemul (1).

Sistemul triunghiular (3) este sistemul cel mai simplu din şirul de transformări; pentru rezolvarea lui se foloseşte procesul de eliminare (substituție) inversă, descris astfel:

$$\begin{cases} x_n = \dfrac{b_n^{(n)}}{a_{nn}^{(n)}} \\ x_i = \dfrac{b_i^{(n)} - \sum_{j=i+1}^{n} a_{ij}^{(n)} x_j}{a_{ii}^{(n)}}, \ i = n-1, n-2, ..., 1 \end{cases} \tag{4}$$

Observații.

1. Formulele (2) pot fi completate astfel încât sistemul (3) obținut să aibă forma:

$$\begin{cases} x_1 + a_{12}^{(n)} x_2 + ... + a_{1n}^{(n)} x_n = b_1^{(n)} \\ x_2 + ... + a_{2n}^{(n)} x_n = b_2^{(n)} \\ ..... \\ x_n = b_n^{(n)} \end{cases},$$

ceea ce înseamnă că în relațiile (4) dispar numitorii, sau formulele (2) pot fi modificate astfel încât sistemul (3) să aibă forma matriceală:

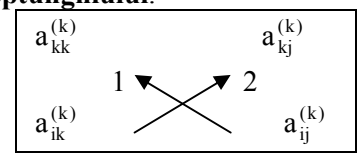
$$\begin{pmatrix} 1 & 0 & ... & 0 \\ 0 & 1 & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1^{(n)} \\ b_2^{(n)} \\ \vdots \\ b_n^{(n)} \end{pmatrix}$$

ceea ce înseamnă că relațiile (4) devin:

$x_i = b_i^{(n)}$, $1 \le i \le n$

Această modificare a metodei Gauss este cunoscută sub numele de metoda Gauss-Jordan.

2. Ultimele din formulele (2), pentru $a_{ij}^{(k+1)}$, $k+1 \le i, j \le n$ şi $b_i^{(k+1)}$, $k+1 \le i \le n$ sunt cunoscute şi sub numele de **regula dreptunghiului**.



$$a_{ij}^{(k+1)} = \frac{a_{ij}^{(k)} \cdot a_{kk}^{(k)} - a_{ik}^{(k)} \cdot a_{kj}^{(k)}}{a_{kk}^{(k)}}$$

1.2. Metoda factorizării LR

Definiția 1. Fie $A \in R^{nxn}$. O relație de forma:

$A = L \cdot R$ (5)

unde $L \in R^{nxn}$ este o matrice inferior triunghiulară iar R este o matrice superior triunghiulară, se numeşte factorizare LR a lui A.

Precizare: o matrice $T \in R^{nxn}$ se numeşte superior (inferior) triunghiulară dacă $t_{ij} = 0$ pentru $i > j$ ($t_{ij} = 0$ pentru $i < j$) unde $T = (t_{ij})$, $1 \le i, j \le n$.

În contextul definiției 1, sistemul liniar (1) devine:

$LRx = b$

care conduce la rezolvarea directă a două sisteme liniare cu matrice triunghiulare şi anume:

$Ly = b$ (6)

$Rx = y$ (7)

Observații:

1. Sistemul (6) are soluția „intermediară“ y ale cărei componente se obțin prin substituție directă (înainte) datorită formei matricei L – inferior triunghiulară.

2. Sistemul (7) are soluția „finală“ x ale cărei componente se obțin prin substituție inversă (înapoi) datorită formei matricei R – superior triunghiulară.

Pentru simplificarea referirilor ulterioare, notăm prin $A_k = (a_{ij}), 1 \le i, j \le k$ submatricele **lider principale** de ordinul k ale lui A, k = 1, 2, ..., n şi introducem condiția:

i) **submatricele A_k sunt nesingulare** (evident $A_1 = a_{11}$, iar $A_n = A$)

Observație. Condiția i) altfel exprimată este: **toți minorii diagonali principali ai matricei A sunt nenuli.**

$a_{11} \neq 0$, $\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \neq 0$, ş.a.m.d. $\det A \neq 0$

(de fapt toți **primii** minori diagonali de fiecare ordin sunt nenuli)

Este valabilă următoarea:

Teoremă. Dacă matricea A satisface condiția i), atunci există o factorizare LR a lui A, în care matricea L este nesingulară.

O factorizare LR a lui A poate fi calculată direct printr-o așa-numită **procedură compactă**, în care cele n^2 egalități scalare, din (5), se rezolvă succesiv în raport cu elementele necunoscute l_{ik}, $i \geq k$ și r_{kj}, $k \leq j$ ale matricilor L și respectiv R. Unicitatea procedurii este asigurată precizând apriori elementele diagonale ale matricei L sau elementele diagonale ale matricei R. Din acest punct de vedere, în practică se utilizează două tipuri de factorizări:

a) **Factorizarea Doolittle** – impune L cu diagonala unitate, adică $l_{kk} = 1$, $k = 1, 2, ..., n$.

b) **Factorizarea Crout** – impune R cu diagonala unitate, adică $r_{kk} = 1$, $k = 1, 2, ..., n$.

Ne vom ocupa de detalierea acestei factorizări, astfel:

Pasul 1. Fie $L = \begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{pmatrix}$ și $R = \begin{pmatrix} 1 & r_{12} & \dots & r_{1n} \\ & 1 & \dots & r_{2n} \\ & & \dots & \dots \\ & & & 1 \end{pmatrix}$

Explicitând egalitatea $A = L \cdot R$, sub forma:

$$a_{ij} = \sum_{k=1}^{\min(i,j)} l_{ik} r_{kj}, \qquad i, j = 1, 2, ..., n$$

elementele matricilor L și R se obțin astfel:

$$\begin{cases} l_{i1} = a_{i1}, \ 1 \leq i \leq n \rightarrow \text{prima coloană din L} \\ r_{1j} = \dfrac{a_{1j}}{l_{11}}, \ 2 \leq j \leq n \rightarrow \text{prima linie din R} \\ l_{ik} = a_{ik} - \sum_{h=1}^{k-1} l_{ih} \cdot r_{hk}, \ 2 \leq k \leq i \leq n \\ r_{kj} = \left(a_{kj} - \sum_{h=1}^{k-1} l_{kh} \cdot r_{hj} \right) \Big/ l_{kk}, \ 2 \leq k < j \leq n \end{cases} \tag{8}$$

Justificarea formulelor (8):

- a_{ik} se obține înmulțind linia i din matricea L cu coloana k din matricea R;

- dacă i > k, atunci $(l_{i1}\ l_{i2}\ \dots\ l_{i,k-1}\ l_{ik}\ \dots\ l_{ii}\ 0\ \dots\ 0) \cdot \begin{pmatrix} r_{1k} \\ r_{2k} \\ \vdots \\ r_{k-1,k} \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \to a_{ik}$

$$\Rightarrow a_{ik} = \sum_{h=1}^{k-1} l_{ih} r_{hk} + l_{ik} ;$$

- dacă i = k, atunci $(l_{i1}\ l_{i2}\ \dots\ l_{i,k-1}\ l_{ii}\ 0\ \dots\ 0) \cdot \begin{pmatrix} r_{1k} \\ r_{2k} \\ \vdots \\ r_{k-1,k} \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \to a_{ik}$

$\Rightarrow a_{ik} = \sum_{h=1}^{k-1} l_{ih} r_{hk} + l_{ii}$; deci am regăsit formulele (8) pentru $i \geq k$;

- dacă $k < j$, atunci $(l_{k1}\ l_{k2} \ldots l_{k,k-1}\ l_{kk}\ 0 \ldots 0) \cdot \begin{pmatrix} r_{1j} \\ r_{2j} \\ \vdots \\ r_{k-1,j} \\ r_{kj} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$

$\Rightarrow a_{kj} = \sum_{h=1}^{k-1} l_{kh} r_{hj} + l_{kk} r_{kj}$, deci am regăsit formulele (8) pentru $k < j$.

Pasul 2. Se rezolvă sistemul (6):

$$\begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \ldots & \ldots & \ldots & \ldots \\ l_{n1} & l_{n2} & \ldots & l_{nn} \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

cu formulele de **substituție directă** (înainte)

$$\begin{cases} y_1 = \dfrac{b_1}{l_{11}} \\ y_k = \left(b_k - \sum_{j=1}^{k-1} l_{kj} y_j \right) \Big/ l_{kk}, k = 2,\ldots,n \end{cases} \qquad (9)$$

Pasul 3. Se rezolvă sistemul (7):

$$\begin{pmatrix} 1 & r_{12} & \ldots & r_{1n} \\ & 1 & \ldots & r_{2n} \\ & & \ldots & \ldots \\ & & & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

cu formulele de **substituție inversă** (înapoi)

$$\begin{cases} x_n = y_n \\ x_k = y_k - \sum_{j=k+1}^{n} r_{kj} x_j, k = n-1, n-2, ..., 1 \end{cases} \quad (10)$$

1.4. Rezolvarea iterativă a sistemelor de ecuații liniare. Metodele Jacobi şi Seidel Gauss

Introducere

Metodele iterative constau în construcția unui şir $x^{(k)}$, (evident $x^{(k)} \in R^n$) k = 0, 1, ... , **convergent** către soluția **exactă** x a sistemului:

Ax = b, unde $A \in R^{nxn}$, $b \in R^n$

Oprirea **procesului iterativ**, adică trunchierea şirului $x^{(k)}$, are loc la un indice s, determinat pe parcursul calculului în funcție de precizia impusă, astfel încât termenul curent $x^{(s)}$ să constituie o **aproximație satisfăcătoare** a soluției căutate x.

Metoda Seidel-Gauss (metoda iterațiilor succesive) constă în construcția şirului $x^{(k+1)}$, k = 0, 1, ... astfel: $x^{(0)} \in R^n$ arbitrar, iar

$$\begin{cases} x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right), i = \overline{2, n-1}, \text{ unde} \\ x_1^{(k+1)} = \frac{1}{a_{1i}} \left(b_1 - \sum_{j=2}^{n} a_{1j} x_j^{(k)} \right) \\ x_n^{(k+1)} = \frac{1}{a_{nn}} \left(b_n - \sum_{j=1}^{n-1} a_{nj} x_j^{(k+1)} \right) \end{cases} \quad (27)$$

Se observă că iterația „nouă“ foloseşte iterații „vechi“, numai dacă nu există iterații „noi“ calculate.

*Observații: C*ondiția **suficientă** pentru ca şirul $x^{(k+1)}$ să conveargă la soluția x este ca A să fie **diagonal dominantă** pe linii (26) sau pe coloane $|a_{jj}| > \sum_{\substack{i=1 \\ i \neq j}}^{n} |a_{ij}|$, $j = \overline{1, n}$.

2. Pot exista sisteme pentru care condiția amintită nu este îndeplinită, dar algoritmul Seidel-Gauss să conveargă. Utilizând norme vectoriale adecvate se poate obține o condiție necesară şi suficientă de convergență, calculele depăşind cadrul prezentei lucrări. Pentru detalii vezi [19] sau [28].

3. Practic, iterațiile se opresc atunci când pentru un ε, impus de utilizator, avem îndeplinite condițiile:

$$\left|x_i^{(k+1)} - x_i^{(k)}\right| < \varepsilon,\ i = \overline{1,n} \qquad (28)$$

(sau trecând la maxim $d(x^{(k+1)}, x^{(k)}) < \varepsilon$). Am putea determina din teorema Banach, indicele la care ne putem opri, în funcție de q.

În acest caz, vom defini ca soluție vectorul $x^{(k+1)}$.

Metoda Jacobi (metoda iteraților simultane) constă în construcția şirului $x^{(k+1)}$, k = 0, 1, ... astfel

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}x_i^{(k)}\right),\ i = \overline{1,n}\ ;\ x^{(0)} \in R^n \text{ arbitrar} \qquad (29)$$

Observații:

1. Sunt valabile observațiile precedente.

2. Se observă că iterația „nouă“ foloseşte toate componentele iterației „vechi“ ceea ce face ca formula să fie mai simplă, dar metoda Jacobi este mai lent convergentă decât metoda Seidel-Gauss în aceleaşi condiții inițiale ($x^{(0)}$, ε).

2. ECUAȚII ŞI SISTEME DE ECUAȚII NELINIARE

2.1. Metoda bisecției

Fiind dată funcția f : [a, b] $\rightarrow$ R şi ecuația f(x) = 0 metoda bisecției se bazează pe proprietatea funcției f de a fi continuă pe [a, b] şi f(a)·f(b) $\leq$ 0. Atunci f are pe [a, b] un număr impar de zerouri.

Dacă $f\left(\frac{a+b}{2}\right) = 0$, $x_1 = \frac{a+b}{2}$ este soluție. Dacă nu, se notează $a_1 = a$ şi $b_1 = \frac{a+b}{2}$ şi $f(a)\cdot f\left(\frac{a+b}{2}\right) < 0$ sau $a_1 = \frac{a+b}{2}$ şi $b_1 = b$ în cazul $f\left(\frac{a+b}{2}\right) f(b) < 0$. Continuând, se obține succesiunea de intervale: $[a_1, b_1]$, $[a_2, b_2]$ ... $[a_n, b_n]$ cu proprietatea că:

$f(a_n) \cdot f(b_n) < 0$, n = 1, 2, 3, ...

cu $b_n - a_n = \frac{1}{2^n}(b-a)$.

Obținem astfel $(a_n)_{n\in N}$ şir crescător şi $(b_n)_{n\in N}$ şir descrescător cu: $x_1 = \lim_{n\to\infty} a_n = \lim_{n\to\infty} b_n$

În ipoteza că au fost separate mai multe zerouri, algoritmul expus poate fi aplicat pentru determinarea fiecărei soluții separate.

Metoda înjumătățirii intervalului este cea mai simplă (dar şi cea mai slab convergentă) metodă pentru determinarea unei soluții a ecuației f(x) = 0.

Algoritmul poate fi aplicat în două variante:

a) Au fost separate una sau mai multe soluții;

b) Nu a fost separată nici o soluție. În această situație se notează cu $a_1 = -\infty$, $b_1 = 0$ dacă $f(-\infty)\cdot f(0) < 0$ sau $a_1 = 0$, $b_1 = +\infty$

dacă $f(0)f(\infty) < 0$, în practică simbolul ∞ fiind înlocuit cu un număr foarte mare.

2.2. Metoda secantei

Considerăm din nou ecuația

$f(x) = 0, x \in [a, b]$ (1)

cu f continuă şi derivabilă de două ori pe [a, b] şi în plus $f(a)\cdot f(b) < 0$, adică a fost separată o soluție a ecuației în acest interval, iar $f''(x)$ păstrează semn constant pe [a, b].

Vom prezenta în continuare metoda secantei pentru aproximarea acestei soluții, notată cu $\bar{x}$.

Pentru a obține o primă aproximație x_1 pentru $\bar{x}$ vom scrie ecuația dreptei care trece prin punctele (a, f(a)) şi (b, f(b)):

$$\frac{x-a}{b-a} = \frac{y-f(a)}{f(b)-f(a)} \tag{2}$$

Aproximația x_1 va fi dată de abscisa punctului în care dreapta taie axa Ox, adică pentru y = 0, relația (2) se scrie sub forma:

$$x_1 = a - \frac{f(a)(b-a)}{f(b)-f(a)} \tag{3}$$

Vom construi din nou o dreaptă similară cu (2) folosind un punct: $(x_1, f(x_1))$ şi un al doilea punct în funcție de intervalul în care se află soluția, (a, x_1) sau (x_1, b).

Dacă $f(a) \cdot f(x_1) < 0$ atunci al doilea punct al dreptei va fi (a, f(a)) şi suntem conduşi la formula generală:

$$x_{n+1} = x_n - \frac{f(x_n)(x_n-a)}{f(x_n)-f(a)} \tag{4}$$

şir descrescător (din construcție) şi mărginit, deci convergent.

Dacă $f(x_1) \cdot f(b) < 0$, al doilea punct va fi (b, f(b)) şi formula iterativă va fi:

$$x_{n+1} = x_n - \frac{f(x_n)(b - x_n)}{f(b) - f(x_n)} \qquad (5)$$

obținând prin construcție un şir crescător şi mărginit.

În ambele cazuri şirul dat de (4) sau (5) va avea ca limită soluția căutată, adică:

$$\overline{x} = \lim_{n \to \infty} x_n$$

Trecând la limită în oricare din relația de recurență (4) sau (5) se obține:

$$f(\overline{x}) = 0$$

2.3. Metoda Newton (tangentei)

2.3.2. Metoda Newton pentru sisteme neliniare

Prin metoda lui Newton se construieşte un şir de aproximații $x^{(n)}$ ale soluției sistemului (6), cunoscând că acest sistem are o soluție $\overline{x}$ şi având o aproximație inițială $x^{(0)}$ a acestei soluții.

În definirea acestui şir de aproximații intervine derivata F'(x) a operatorului F.

Din relația (7) rezultă că numărătorul tinde la 0 mai repede decât h. Pentru ||h|| suficient de mic, putem face aproximarea

$$F(x+h) - F(x) - F'(x)h \cong 0$$

Să aplicăm această relație pentru o aproximație $x^{(n)}$ a soluției ecuației (1) şi pentru $h = \overline{x} - x^{(n)}$. Obținem:

$F(\overline{x}) - F(x^{(n)}) - F'(x^{(n)})(\overline{x} - x^{(n)}) \cong 0$, şi cum:

$F(\overline{x}) = 0$ – căci $\overline{x}$ este soluția ecuației (6), obținem:

$F(x^{(n)}) + F'(x^{(n)})(\overline{x} - x^{(n)}) \cong 0$

Datorită aproximării, rezolvând această ecuație nu vom obține soluția exactă ($\overline{x}$), ci o valoare $x^{(n+1)}$, care o definim ca aproximație de ordinul n+1.

$$F(x^{(n)}) + F'(x^{(n)})(x^{(n+1)} - x^{(n)}) = 0$$

Aproximația $x^{(n+1)}$, presupunând că există $[F'(x^{(n)})]^{-1}$, este de forma:

$x^{(n+1)} = x^{(n)} - [F'(x^{(n)})]^{-1} F(x^{(n)})$ (8)

Procesul iterativ definit de (8) poartă numele de metoda lui Newton.

Observație. În particular, metoda lui Newton, se poate aplica pentru rezolvarea ecuațiilor de forma:

$f(x) = 0$

unde $f : R \to R$ este o funcție derivabilă, cu derivata nenulă într-un anumit interval care conține o soluție $\bar{x}$ a acestei ecuații. Şirul (8) devine, în acest caz:

$$x^{(n+1)} = x^{(n)} - \frac{f(x^{(n)})}{f'(x^{(n)})}$$

şi metoda admite următoarea interpretare geometrică: „$x^{(n+1)}$ este abscisa punctului de intersecție cu axa Ox a tangentei la graficul funcției f(x) în punctul $(x^{(n)}, f(x^{(n)}))$“. Din acest motiv, metoda lui Newton se mai numeşte şi metoda tangentei.

Observație. Metoda Newton se poate aplica şi în cazul sistemelor neliniare:

$$\begin{cases} f_1(x_1, x_2, ..., x_m) = 0 \\ f_2(x_1, x_2, ..., x_m) = 0 \\ ..... \\ f_m(x_1, x_2, ..., x_m) = 0 \end{cases}$$

unde $F : R^m \to R^m$, cu $F = (f_1, f_2, ..., f_m)^t$. În exemplul dat am constatat că dacă f_i au derivate parțiale de ordinul I continue, atunci există derivata lui F şi $F'(x) = \left(\frac{\partial f_i(x)}{\partial x_j} \right)$, matricea iacobiană.

3. POLINOM CARACTERISTIC. VECTORI ŞI VALORI PROPRII

Preliminarii (noţiuni generale).

Definiţia 1. Dacă matricea $A \in R^{nxn}$, atunci polinomul definit prin:

$$p_A(\lambda) = \det(\lambda I - A) \qquad (1)$$

se numeşte **polinomul caracteristic** al matricei A.

Precizări:

1. Polinomul p_A este un polinom unitar, de grad n, cu coeficienţi reali şi ecuaţia caracteristică de forma:

$$p_A(\lambda) = 0 \Leftrightarrow \lambda^n + c_1\lambda^{n-1} + c_2\lambda^{n-2} + ...+ c_{n-1}\lambda + c_n = 0 \qquad (1')$$

are cel mult n rădăcini distincte (reale sau complexe).

2. Dacă λ este o rădăcină a lui p_A, atunci, din $\det(\lambda I - A) = 0$, înseamnă că sistemul liniar omogen definit prin

$$(A - \lambda I) \cdot x = \mathbf{0} \text{ (vectorul nul din } R^n) \qquad (2)$$

are şi soluţii nebanale.

În continuare vom studia rădăcinile lui p_A şi soluţiile nebanale ale sistemului (2).

Definiţia 2. Dacă p_A este polinomul caracteristic al matricei A, rădăcinile lui p_A se numesc **valori proprii** (sau valori caracteristice) ale matricei A. Dacă λ este o valoare proprie a lui A şi $x \neq 0$ verifică sistemul (2) atunci x se numeşte **vector propriu** (sau vector caracteristic) al lui A corespunzător valorii proprii λ.

Observaţie. Egalitatea (2) conduce la:

$$Ax = \lambda x \qquad (3)$$

Definiţia 3. Mulţimea $\sigma(A) = \{\lambda_1, \lambda_2, ..., \lambda_n\}$ se numeşte **spectrul** lui A, iar numărul $\rho(A) = \max_{1\le i\le n} |\lambda_i|$ se numeşte **raza spectrală** a lui A.

Consecinţă. Orice matrice $A \in R^{nxn}$ are cel puţin un vector propriu x, mai precis, fiecărei valori proprii $\lambda_i \in \sigma(A)$ îi corespunde cel puţin un vector propriu $x_i \in \text{Ker}(\lambda I_n - A)$.

Metodele numerice pentru calculul valorilor şi vectorilor proprii se împart în două clase astfel:

I. – Metode pentru calcularea polinomului caracteristic, în care determinarea coeficienților lui $p_A(\lambda)$ se face în mod direct. Deci o cale posibilă pentru a calcula valorile proprii este să obținem întâi coeficienții polinomului caracteristic:

$p_A(\lambda) = \lambda^n + c_1\lambda^{n-1} + ... + c_n$

şi apoi să rezolvăm ecuația caracteristică $p_A(\lambda) = 0$, folosind în acest scop orice metodă aproximativă de rezolvare a ecuațiilor algebrice neliniare.

Totuşi, nu trebuie să procedăm astfel decât în cazurile când matricea A este de ordin foarte mic şi soluțiile ecuației caracteristice sunt bine separate.

Motivul este că valorile proprii sunt foarte sensibile la perturbațiile coeficienților $c_1, c_2, ..., c_n$, datorate erorilor de rotunjire inerente.

II. – Metode pentru determinarea valorilor şi vectorilor proprii ai matricei A, fără a dezvolta în prealabil polinomul caracteristic. Aceste metode sunt în esență proceduri iterative de aducere a matricei A la anumite forme simple („canonice“) prin transformări de asemănare sau prin transformări de asemănare ortogonală (forma diagonală, forma tridiagonală, forma superior triunghiulară, forma superior Hessenberg).

3.1. Calculul polinomului caracteristic cu ajutorul minorilor diagonali

Dacă $A \in R^{nxn}$ este de forma $A = (a_{ij})$, $1 \le i, j \le n$, atunci polinomul ei caracteristic se poate scrie:

$p_A(\lambda) = \det(\lambda\delta_{ij} - a_{ij})$

Elementele fiecărei coloane a determinantului de mai sus sunt sume algebrice de câte două elemente. Deci $p_A(\lambda)$ se descompune într-o sumă de 2^n determinanți. Fie Δ unul dintre aceşti determinanți şi să notăm cu $j_1, j_2, ..., j_m$ coloanele lui Δ care conțin numai elemente ale matricei A. Toate celelalte coloane conțin λ la intersecția cu diagonala principală şi în rest 0.

Dezvoltând succesiv Δ după coloanele care conțin λ şi scoțând -1 factor comun pe celelalte coloane, se obține:

$$\Delta = (-1)^m \cdot \lambda^{n-m} \cdot M_{j_1 j_2 \dots j_m}$$

unde $M_{j_1 j_2 \dots j_m}$ este minorul diagonal al matricei A format cu elementele care se află la intersecția liniilor şi coloanelor de indici $j_1, j_2, \dots, j_m$.

Dacă în dezvoltarea lui $p_A(\lambda)$ se grupează termenii după puterile lui λ, se obține:

$$p_A(\lambda) = \lambda^n - \sigma_1\lambda^{n-1} + \sigma_2\lambda^{n-2} - \dots + (-1)^n\sigma_n,$$

unde:

$\sigma_1 = \sum_{i=1}^{n} a_{ii}$ (σ_1 este suma minorilor diagonali de ordinul unu);

$\sigma_2 = \sum_{1 \le i < j \le n} \begin{vmatrix} a_{ii} & a_{ij} \\ a_{ji} & a_{jj} \end{vmatrix}$ (σ_2 este suma minorilor diagonali de ordinul doi); ş.a.m.d.

$\sigma_n = \det A$ (σ_n reprezintă singurul minor diagonal de ordinul n);

3.2. Metoda Leverrier

Dacă $A \in R^{nxn}$, atunci polinomul său caracteristic este:

$p_A(\lambda) = \lambda^n - \sigma_1\lambda^{n-1} + \sigma_2\lambda^{n-2} - \dots + (-1)^n\sigma_n$

Pentru a-i determina coeficienții procedăm astfel:

1) Determinăm $s_k = Tr(A^k)$, $1 \le k \le n$.

2) $\sigma_1 = s_1$

$\sigma_k = (s_1\sigma_{k-1} - s_2\sigma_{k-2} + \dots + (-1)^{k+1}s_k)/k$, $2 \le k \le n$.

Algoritmul lui Leverrier se programează fără dificultate şi prezintă un grad mare de generalitate (nu există cazuri particulare).

Pentru n mare, calculele sunt dificile (trebuie calculate puterile matricei date) şi se măreşte timpul de răspuns.

3.3. Metoda Krylov

Este o metodă pentru calculul polinomului caracteristic bazată pe teorema lui Cayley-Hamilton. Fie $A \in R^{nxn}$ şi $p_A(\lambda) = \lambda^n + c_1\lambda^{n-1} + + ... + c_{n-1}\lambda + c_n$, polinomul caracteristic al matricei A. Trebuie să determinăm coeficienţii c_1, c_2, ..., c_n. Conform teoremei amintite, avem:

$$p_A(A) = A^n + c_1A^{n-1} + ... + c_{n-1}A + c_nI_n = O_n \quad (7)$$

Fie $y^{(0)} \in R^n$, oarecare, nenul. Din relaţia (7) rezultă:

$$A^ny^{(0)} + c_1A^{n-1}y^{(0)} + ... + c_{n-1}Ay^{(0)} + c_ny^{(0)} = \mathbf{0} \quad (8)$$

Dacă se notează

$$A^ky^{(0)} = y^{(k)},\ k = 1, 2, ..., n \quad (9)$$

atunci relaţia (8) se mai scrie:

$$c_1y^{(n-1)} + c_2y^{(n-2)} + ... + c_{n-1}y^{(1)} + c_ny^{(0)} = -y^{(n)} \quad (10)$$

Scrisă pe componente, relaţia (10) reprezintă un sistem de n ecuaţii liniare cu n necunoscute c_1, c_2, ..., c_n. Dacă determinantul sistemului (10) este nenul, atunci soluţia unică reprezintă coeficienţii polinomului caracteristic. Această soluţie se poate obţine prin una din metodele de rezolvare a sistemelor de ecuaţii liniare. Dacă determinantul sistemului (10) este nul, atunci trebuie reluate calculele cu un alt vector iniţial $y^{(0)}$.

Observaţie. Relaţia (9) pentru obţinerea vectorilor $y^{(1)}$, $y^{(2)}$, ..., $y^{(n)}$ conduce la calcule simplificate dacă este scrisă sub forma echivalentă:

$$y^{(k)} = Ay^{(k-1)},\ k = 1, 2, ..., n \quad (11)$$

Dacă polinomul caracteristic are rădăcini distincte, atunci vectorii $y^{(0)}$, $y^{(1)}$, ..., $y^{(n-1)}$ obţinuţi mai sus permit determinarea simplă şi a vectorilor proprii. Fie λ_1, λ_2, ..., λ_n valorile proprii distincte şi $x^{(1)}$, $x^{(2)}$, ... $x^{(n)}$ vectorii proprii corespunzători care formează o bază. Fie:

$$y^{(0)} = b_1x^{(1)} + b_2x^{(2)} + ... + b_nx^{(n)} \quad (12)$$

expresia vectorului iniţial în această bază.

Observaţie. Menţionăm faptul că vectorul iniţial a condus la obţinerea coeficienţilor c_1, c_2, ..., c_n ai polinomului caracteristic.

În continuare, din (9) sau (11) şi definiţia vectorilor proprii, obţinem:

$$\begin{cases} y^{(1)} = \sum_{k=1}^{n} b_k \lambda_k x^{(k)} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ y^{(n-1)} = \sum_{k=1}^{n} b_k \lambda_k^{n-1} x^{(k)} \end{cases} \quad (13)$$

Să notăm:

$$q_j(\lambda) = \frac{p_A(\lambda)}{\lambda - \lambda_j} = \lambda^{n-1} + q_{1j}\lambda^{n-2} + \ldots + q_{n-1,j},\ j = 1, 2, \ldots, n. \quad (14)$$

În ipoteza făcută (valori proprii distincte), avem.

$$\begin{cases} q_j(\lambda_k) = 0,\ k \neq j \\ q_j(\lambda_j) \neq 0 \end{cases} \quad (15)$$

şi din relaţiile de mai sus deducem:

$y^{(n-1)} + q_{1j}y^{(n-2)} + \ldots + q_{n-1,j}y^{(0)} = b_1q_j(\lambda_1)x^{(1)} + b_2q_j(\lambda_2)x^{(2)} + \ldots +$
$+ b_jq_j(\lambda_j)x^{(j)} + \ldots + b_nq_j(\lambda_n)x^{(n)} = b_jq_j(\lambda_j)x^{(j)}$

Dacă $b_j \neq 0$, atunci $b_jq_j(\lambda_j)x^{(j)}$ este de asemenea un vector propriu corespunzător valorii proprii λ_j, deci

$$y^{(n-1)} + q_{1j}y^{(n-2)} + \ldots + q_{n-1,j}y^{(0)} \quad (16)$$

este un vector propriu corespunzător valorii proprii λ_j.

3.4. Metoda Fadeev

Este o metodă, care pornind de la relaţia (17), cu un efort minim, permite obţinerea atât a coeficienţilor polinomului caracteristic al matricei date $A \in R^{nxn}$, cât şi a inversei acesteia A^{-1}.

Algoritmul metodei Fadeev este caracterizat de relaţiile:

$$\begin{cases} A_1 = A; & c_1 = -Tr(A_1); & B_1 = A_1 + c_1I_n \\ A_2 = AB_1; & c_2 = -\frac{1}{2}Tr(A_2); & B_2 = A_2 + c_2I_n \\ \ldots & \ldots & \ldots \\ A_n = AB_{n-1}; & c_n = -\frac{1}{n}Tr(A_n); & B_n = A_n + c_nI_n \end{cases} \quad (18)$$

Se demonstrează uşor următoarele afirmaţii:

$$\begin{cases} B_n = O_n; \\ A^{-1} = -\dfrac{1}{c_n} B_{n-1}, \text{ dacă } c_n \neq 0 \end{cases}$$

3.5. Metoda Danilevski

Fie $A \in R^{nxn}$. Ne propunem să obținem p_A.

Definiția 7. Spunem că matricea $F \in R^{nxn}$ are forma normală Frobenius, dacă:

$$F = \begin{pmatrix} f_1 & f_2 & \dots & f_{n-1} & f_n \\ 1 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix} \qquad (19)$$

Proprietatea 2.

$$p_F(\lambda) = \lambda^n - f_1\lambda^{n-1} - f_2\lambda^{n-2} - f_{n-1}\lambda - f_n \qquad (20)$$

(deci o matrice cu formă normală Frobenius are în prima linie opuşii coeficienților polinomului său caracteristic).

Observație. Demonstrarea relației (20) se face dezvoltând determinantul matricei $(\lambda I - F)$ după prima linie.

Metoda lui Danilevski, pentru aflarea polinomului caracteristic al matricei date, A, constă în aducerea matricei A, la forma normală Frobenius prin procedee de asemănare. Acest lucru se realizează în n-1 etape, la fiecare etapă, obținând câte o linie din matricea F, dată de (19), de la ultima linie până la prima.

Astfel:

Etapa 1. Presupunem $a_{n,n-1} \neq 0$ pentru a realiza ultima linie din matricea F.

Observație. Cazul $a_{n,n-1} = 0$ (posibil practic) va fi tratat la cazuri particulare.

Cu presupunerea $a_{n,n-1} \neq 0$, vom prelucra matricea A astfel:

- împărțim elementele coloanei n-1 prin $a_{n,n-1}$

- apoi, din fiecare coloană j, $1 \le j \le n$, $j \ne n-1$ vom scădea coloana n–1 (obținută anterior) înmulțită cu a_{nj}.

Adică:

$$\begin{cases} a'_{i,n-1} = a_{i,n-1} / a_{n,n-1}, & 1 \le i \le n, \\ a'_{ij} = a_{ij} - a'_{i,n-1} \cdot a_{nj}, & 1 \le j \le n,\ j \ne n-1,\ 1 \le i \le n. \end{cases} \tag{21}$$

Observație. Linia n a matricei A se înlocuieşte cu ultima linie din F, adică 0 0 ... 1 0 (vezi formulele (21) pentru i = n).

Observație. Transformările (21) sunt echivalente cu înmulțirea la dreapta a matricei A cu matricea:

$$M_{n-1} = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ -\dfrac{a_{n1}}{a_{n,n-1}} & -\dfrac{a_{n2}}{a_{n,n-1}} & \dots & \dfrac{1}{a_{n,n-1}} & -\dfrac{a_{nn}}{a_{n,n-1}} \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \text{ linia „n–1“} \tag{22}$$

Deci, matricea M_{n-1} diferă de matricea unitate de ordinul n, doar în linia „n–1“, cu componența din (22).

Observație. Dacă notăm cu $B_{n-1} = A \cdot M_{n-1}$, atunci ultima linie a matricei B_{n-1} este 0 0 ... 1 0 şi elementele celorlalte linii i, $1 \le i \le n-1$ se calculează cu ajutorul relațiilor (21).

Observație. Matricea M_{n-1} este inversabilă şi

$$M_{n-1}^{-1} = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{n,n-1} & a_{nn} \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \text{ linia „n–1“} \tag{23}$$

Deci, matricea M_{n-1}^{-1} diferă de matricea unitate de ordinul n doar în linia n-1 care este ultima linie din matricea A.

Cu toate aceste precizări făcute, în prima etapă se obține matricea $C_{n-1} = M_{n-1}^{-1} \cdot B_{n-1}$, adică $C_{n-1} = M_{n-1}^{-1} \cdot A \cdot M_{n-1}$, care este asemenea cu matricea $A (A \sim C_{n-1})$.

Observaţie. Cu excepţia liniei „n–1“, matricele C_{n-1} şi B_{n-1} au aceleaşi elemente. Elementele liniei „n–1“ din matricea C_{n-1} se obţin înmulţind elementele liniei „n–1“ din matricea M_{n-1}^{-1}, (23), cu elementele fiecărei coloane din B_{n-1}.

Obţinem, astfel, relaţiile:

$$a'_{n-1,j} = \sum_{i=1}^{n} a_{ni} \cdot a'_{ij}\ , \quad 1 \le j \le n \qquad (24)$$

Relaţiile (24) se referă la linia „n–1“ din matricea C_{n-1}, restul liniilor matricei C_{n-1} fiind preluate din B_{n-1}, fără nici-o modificare.

Etapa 2. În etapa a doua se reiau calculele bazate pe relaţiile (21) + (24), presupunând $a'_{n-1,n-2} \neq 0$ ($a'_{n-1,n-2}$ este element din C_{n-1}). Cu alte cuvinte se construieşte matricea:

$$C_{n-2} = M_{n-2}^{-1} \cdot C_{n-1} \cdot M_{n-2},$$

asemenea cu matricea C_{n-1}, deci asemenea cu A (relaţia de asemănare fiind relaţie de echivalenţă). În plus, ultimele două linii din C_{n-2} sunt identice cu ultimele două linii din F.

Observaţie. Putem scrie $C_{n-2} = M_{n-2}^{-1}(M_{n-1}^{-1} \cdot A \cdot M_{n-1}) \cdot M_{n-2} =$ $= (M_{n-2}^{-1} \cdot M_{n-1}^{-1}) \cdot A \cdot (M_{n-1} \cdot M_{n-2})$ – folosind asociativitatea produsului matricelor. Deci $A \sim C_{n-2}$.

Observaţie. Matricele M_{n-2}, M_{n-2}^{-1} se calculează asemănător matricelor M_{n-1}, M_{n-1}^{-1} din (22) respectiv (23), folosind linia „n–1“ din C_{n-1} pentru a obţine linia „n-2“ din M_{n-2}, respectiv M_{n-2}^{-1}. Adică, linia „n–2“ din M_{n-2} este:

$$-\frac{a'_{n-1,1}}{a'_{n-1,n-2}} \; -\frac{a'_{n-1,2}}{a'_{n-1,n-2}} \; \cdots \; \frac{1}{a'_{n-1,n-2}} \; -\frac{a'_{n-1,n-1}}{a'_{n-1,n-2}} \; -\frac{a'_{n-1,n}}{a'_{n-1,n-2}}$$

Analog, linia „n–2“ din M_{n-2}^{-1} este:

$$a'_{n-1,1} \quad a'_{n-1,2} \quad \ldots \quad a'_{n-1,n-2} \quad a'_{n-1,n-1} \quad a'_{n-1,n} \quad \text{ş.a.m.d.}$$

Etapa n-1. În etapa n-1 se obţine matricea $C_1 = M_1^{-1} \cdot C_2 \cdot M_1$ (în ipoteza $c_{21} \neq 0$, c_{21} fiind element din C_2), $A \sim C_1$, C_1 având forma (19) deoarece se obţin ultimele n-1 linii din (19). Relaţia dintre C_1 şi A este dată de:

$C_1 = (M_1^{-1} M_2^{-1} ... M_{n-1}^{-1}) A \cdot (M_{n-1} \cdot M_{n-2} ... M_1)$

Notând $S = M_{n-1} M_{n-2} ... M_1 \Rightarrow C_1 = S^{-1} \cdot A \cdot S$.

4. APROXIMAREA FUNCȚIILOR

Introducere

Vom încerca în cele ce urmează să prezentăm pe scurt, procesul de aproximare a unei funcții cunoscută printr-un număr finit de valori.

Printr-un anumit procedeu, utilizarea unor aparate de măsură de exemplu, putem cunoaşte valorile unei funcții (temperatură, tensiuni sau intensitățile electrice, magnetism, etc.) cu expresie necunoscută pentru anumite valori ale variabilei.

Vom nota cu $x_0, x_1, ..., x_n$ valorile pentru care au avut loc măsurătorile şi cu $f_0, f_1, ..., f_n$ valorile măsurate, adică: $f_i = f(x_i)$, $i = \overline{0,n}$.

Există situații în care este necesară cunoaşterea valorii funcției f într-un punct $\overline{x}$, $\overline{x} \in [x_0, x_n]$, $\overline{x} \neq x_i$, $i = \overline{0,n}$, care nu a făcut deci parte din setul de măsurători inițiale.

Dacă reluarea măsurătorilor pentru determinarea valorii $f(\overline{x})$ nu mai este posibilă din diverse motive, de exemplu acela că fenomenul nu se mai poate repeta, suntem nevoiți ca pe baza datelor acumulate să căutăm o aproximare pentru $f(\overline{x})$.

Având în vedere faptul că pot exista mai multe valori de genul $\overline{x}$, vom căuta un aproximant pentru funcția f pe intervalul studiat $[x_0, x_n]$, care să aproximeze de fiecare dată valorile $f(\overline{x})$, pentru $\overline{x}$ dat.

Vom expune în cadrul acestui capitol două metode de construcție a aproximantului:

a) ca polinom de interpolare (când n este mic!);

b) ca element de cea mai bună aproximare obținut prin metoda celor mai mici pătrate (când n este mare!).

Aproximanții construiți pot de asemenea înlocui funcția în operații de integrare şi derivare aproximativă aşa cum se va prezenta în capitolele următoare.

Aproximarea funcțiilor prin interpolare

Fie „n+1“ puncte **distincte**, x_i, $i = \overline{0,n}$ (numite noduri) pe un interval $I \subset R$, şi $f(x_i) \overset{not}{=} f_i$, $i = \overline{0,n}$ valorile **date** ale funcției reale f în aceste puncte ($f : R \to R$).

Să determinăm un polinom de grad **cel mult** egal cu n, notat P_n care să treacă prin punctele (x_i, f_i), $i = \overline{0,n}$.

Un asemenea polinom îl numim polinom de interpolare, iar despre f spunem că se aproximează polinomial pe I.

Se pun următoarele întrebări:

a) dacă există, care este expresia polinomului de interpolare în funcție de x_i şi f_i, $i = \overline{0,n}$?

b) cât de bine aproximează polinomul de interpolare funcția f pe I?

Să încercăm să răspundem la aceste întrebări.

4.1. Polinomul de interpolare Lagrange

4.1.1. Construirea polinomului

Calea directă de determinare a polinomului de interpolare ar fi folosirea expresiei generale a polinomului de gradul n, coeficienții lui (în număr de n+1) urmând să rezulte din cele n+1 condiții.

$$P_n(x_i) = f_i,\ i = \overline{0,n} \qquad (1)$$

Adică, fie $P_n(x) = a_0x^n + a_1x^{n-1} + ... + a_n$. Astfel (1) reprezintă un sistem liniar cu n+1 ecuații (deci cele n+1 condiții (1)) şi n+1 necunoscute a_0, a_1, ..., a_n (coeficienții polinomului de interpolare). Determinantul matricei acestui sistem este determinantul Vandermonde $V(x_0, x_1, ..., x_n) \neq 0$, deoarece nodurile sunt distincte. Rezultă soluția unică a_0, a_1, ..., a_n.

Deci polinomul de interpolare există şi este unic, expresia sa fiind dependentă de x_i, f_i, $i = \overline{0,n}$.

Fiind destul de laborioasă această cale, vom prefera o altă cale care conduce direct la expresia polinomului de interpolare.

Să notăm cu l_k, $k = \overline{0,n}$, un polinom de grad n pentru care $l_k(x_j) = \delta_{kj}$ (simbolul lui Kronecker), $(\forall) j = \overline{0,n}$.

Aşadar

$$l_k(x) = \prod_{\substack{j=0 \\ j\neq k}}^{n} \frac{x - x_j}{x_k - x_j}, \qquad k = \overline{0,n} \tag{2}$$

adică $l_k(x) = \dfrac{(x-x_0)...(x-k_{k-1})(x-x_{k+1})...(x-x_n)}{(x_k-x_0)...(x_k-x_{k-1})(x_k-x_{k+1})...(x_k-x_n)}$, $k= \overline{0,n}$

Observaţie. Polinoamele de grad n, definite de (2) se numesc polinoame fundamentale Lagrange (justificarea notaţiei).

Este valabilă următoarea proprietate:

Proprietatea 1. Polinomul $P_n(x) = \sum_{k=0}^{n} f_k\, l_k(x)$ este polinom de interpolare (numit polinom de interpolare Lagrange).

Aşadar, polinomul de interpolare Lagrange:

$$P_n(x) = \sum_{k=0}^{n} f_k \prod_{\substack{j=0 \\ j\neq k}}^{n} \frac{x - x_j}{x_k - x_j} \tag{3}$$

este un polinom de interpolare pentru datele (x_i, f_i), $i = \overline{0,n}$.

Proprietatea 2. Polinomul de interpolare corespunzător la „n+1“ puncte distincte, există şi este unic (sau polinomul de interpolare Lagrange constituie **singurul** polinom de grad cel mult n ce interpolează datele (x_i, f_i), $i = \overline{0,n}$).

4.2. Polinomul de interpolare Newton

4.2.1. Diferenţe divizate

Fie „n+1“ puncte **distincte** x_i, $i = \overline{0,n}$ (numite noduri) pe un interval $I \subset R$ şi $f(x_i) \overset{not}{=} f_i$, $i = \overline{0,n}$ valorile date ale funcției reale f în aceste puncte.

Definiția 1. Definim diferențele divizate de ordin k, $k = \overline{0,n}$, recursiv astfel:

– diferențele divizate de ordin **zero** coincid cu valorile f_i, ale funcției f în nodurile x_i, $i = \overline{0,n}$ (deci sunt „n+1“ diferențe divizate de ordin 0). Le notăm $f(x_i)$;

– diferențele divizate de ordin **unu** se definesc prin egalitatea

$$f(x_i; x_{i+1}) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i},\ 0 \le i \le n\text{-}1,$$

adică $f(x_0; x_1)$, $f(x_1; x_2)$, ..., $f(x_{n-1}, x_n)$ (deci sunt „n“ diferențe divizate de ordin 1);

– diferențele divizate de ordin k, se obțin din diferențe divizate de ordin k-1 după formula:

$$f(x_0; x_1; ...; x_k) = \frac{f(x_1; x_2; ...; x_k) - f(x_0; x_1; ...; x_{k-1})}{x_k - x_0} \qquad (7)$$

Observație. Există o singură diferență divizată de ordinul n şi anume $f(x_0; x_1; ..., x_n) = \dfrac{f(x_1; x_2; ...; x_n) - f(x_0; x_1; ...; x_{n-1})}{x_n - x_0}$.

Observație. Din definiția 1 rezultă că putem trece diferențele divizate (de toate ordinele) într-un tablou astfel:

x_i	dif.div. de ord.0	dif.div. de ord.1	ord.2	...	ord. n
x_0	$f(x_0) = f_0$	$f(x_0; x_1)$	$f(x_0; x_1; x_2)$		$f(x_0; x_1; ...; x_n)$
x_1	$f(x_1) = f_1$	$f(x_1; x_2)$	$f(x_1; x_2; x_3)$		–
x_2	$f(x_2) = f_2$	$f(x_2; x_3)$	$f(x_2; x_3; x_4)$		–
⋮	⋮	⋮	⋮		⋮
x_{n-2}	$f(x_{n-2}) = f_{n-2}$	$f(x_{n-2}; x_{n-1})$	$f(x_{n-2}; x_{n-1}; x_n)$		–
x_{n-1}	$f(x_{n-1}) = f_{n-1}$	$f(x_{n-1}; x_n)$	–		–
x_n	$f(x_n) = f_n$	–	–		–

folosind formulele (7).

Deci notând d_{ij}, elementele acestui tablou

$\Rightarrow d_{i0} = f_i$, $i = 0, n$

$$d_{ij} = \frac{d_{i+1,j-1} - d_{i,j-1}}{x_{j+i} - x_i}\ , j = \overline{1,n}\ ; i = \overline{0,n-j}$$

Observație. Se poate demonstra, prin inducție după m, că

$$f(x_0; ...; x_m) = \sum_{j=0}^{m} \frac{f(x_j)}{\prod_{\substack{i=0 \\ i \neq j}}^{m} (x_j - x_i)} \qquad (8)$$

ceea ce arată că diferențele divizate sunt simetrice în argumente.

Polinomul notat

$$N_n(x) = f(x_0) + \sum_{k=0}^{n-1} f(x_0;...;x_{k+1}) \prod_{j=0}^{k} (x - x_j) \qquad (12)$$

se numeşte **polinomul lui Newton de interpolare cu diferențe divizate**. Din (9), combinând cu (5), deducem:

$$f(x; x_0; ...; x_n) = \frac{f^{(n+1)}(\xi)}{(n+1)!},$$

ξ este intermediar punctelor x, x_0, ..., x_n.

Observații:

1. Polinomul de interpolare Lagrange, cu exprimarea (3) este identic cu polinomul de interpolare al lui Newton cu exprimarea (12).

2. Din tabloul diferențelor divizate, se observă că ne interesează doar prima linie.

4.4. Aproximarea cu funcții spline cubice

4.4.1. Construcția aproximantului spline cubic

Dacă gradul polinomului este 3 vom spune că aproximăm cu **spline cubic**, definit prin condițiile de mai jos:

1. $S(x_i) = f_i$, $0 \leq i \leq n$;
2. S, S', S'' continue pe [a, b];

3. S polinom de gradul trei pe fiecare subinterval $[x_{i-1},x_i]$, $1 \le i \le n$. Notăm:

S_i – restricția lui S pe $[x_{i-1}, x_i]$ adică $S_i(x) = S(x)$, $x \in [x_{i-1}, x_i]$.

$u_i = S''(x_i)$, $0 \le i \le n$, deci $u_i = S''_i(x_i) \overset{cont}{=} S''_{i+1}(x_i)$.

Forma generala a ramurii „i" a spline-ului cubic este

$$S_i(x) = \frac{u_i(x-x_{i-1})^3 + u_{i-1}(x_i-x)^3}{6h_i} + \left(f_i - u_i\frac{h_i^2}{6}\right)\frac{x-x_{i-1}}{h_i} +$$

$$+ \left(f_{i-1} - u_{i-1}\frac{h_i^2}{6}\right)\frac{x_i-x}{h_i}, \qquad (22)$$

Se observă continuitatea lui S.

În continuare, vom impune condițiile de continuitate şi pentru S', adică

$$S'_i(x_i) = S'_{i+1}(x_i),\ 1 \le i \le n-1$$

de unde avem:

$$u_{i-1}\frac{h_i}{6} + u_i\frac{h_i+h_{i+1}}{3} + u_{i+1}\frac{h_{i+1}}{6} = \frac{f_{i+1}-f_i}{h_{i+1}} - \frac{f_i-f_{i-1}}{h_i} \qquad (23)$$

$1 \le i \le n-1$

Relațiile (23) reprezintă un sistem de n-1 ecuații cu n+1 necunoscute u_0, u_1, ..., u_n, pentru a căror determinare există două cazuri practice:

I. S''(a) = S''(b) = 0, adică $u_0 = u_n = 0$. Rezultă n – 1 ecuații cu n – 1 necunoscute: u_1, u_2, ..., u_{n-1}, adică am redus numărul de necunoscute.

II. $\left.\begin{aligned} S'(a) = f'(a) &\overset{not}{=} f'_0 \\ S'(b) = f'(b) &\overset{not}{=} f'_n \end{aligned}\right\} \Rightarrow u_0\frac{h_1}{3} + u_1\frac{h_1}{6} = \frac{f_1-f_0}{h_1} - f'_0 \qquad (24)$

şi

$$u_{n-1}\frac{h_n}{6} + u_n\frac{h_n}{3} = f'_n - \frac{f_n-f_{n-1}}{h_n}, \qquad (25)$$

adică mărim numărul de ecuaţii şi deci (24) + (23) + (25) reprezintă un sistem de n+1 ecuaţii cu n+1 necunoscute: u_0, u_1, ..., u_n.

În ambele cazuri I şi II sistemele obţinute sunt tridiagonale simetrice cu diagonala principală dominantă.

4.5. Metoda celor mai mici pătrate – cazul funcţiilor tabelate

4.5.1. Construcţia elementului de cea mai bună aproximare

Fiind dată o funcţie f : [a, b] $\rightarrow$ R vom spune că este tabelată atunci când se cunosc valorile $f(x_i) = f_i$ ale funcţiei f pe un sistem de noduri echidistante sau nu: $a = x_1 < x_2 < ... < x_n = b$.

Metoda celor mai mici pătrate (m.c.m.m.p.) îşi propune ca pentru funcţiile tabelate să determine un aproximant care să aproximeze suficient de bine funcţia în orice punct din intervalul [a, b], fără să coincidă cu funcţia în mod expres în vreun punct.

Alegerea aproximantului se va face dintr-o familie de funcţii liniar independentă, numită sistem de generatori sau funcţii standard şi evident forma lui va depinde de familia din care face parte.

Cea mai comodă alegere pentru aproximant va fi de formă polinomială.

Elementul de cea mai bună aproximare în sensul celor mai mici pătrate se va determina din condiţia de minimizare a abaterii dintre funcţia f şi aproximantul g:

$$\|f - g\|^2 = \sum_{i=1}^{n} |f(x_i) - g(x_i)|^2 \qquad (33)$$

unde:

$$\|f\|^2 = \sum_{i=1}^{n} f_i^2 \ .$$

Aproximantul g va fi generat de o familie finită de funcţii liniar independente: $\varphi_1, \varphi_2, ..., \varphi_m$ adică g va avea forma:

$$g(x) = \sum_{i=1}^{m} c_i \varphi_i(x) \tag{34}$$

Coeficienţii $c_i \in R$ îi vom determina din condiţia de minimizare a erorii de aproximare:

$$\min R(c_1, c_2, ..., c_m) = \min \sum_{j=1}^{n} \left(f(x_j) - \sum_{i=1}^{m} c_i \varphi_i(x_j) \right)^2 \tag{35}$$

echivalentă cu rezolvarea sistemului liniar:

$$\frac{\partial R}{\partial c_1} = 0 ; \quad \frac{\partial R}{\partial c_2} = 0 ; \quad .... \quad \frac{\partial R}{\partial c_m} = 0 \tag{36}$$

sau

$$\sum_{i=1}^{m} c_i \sum_{j=1}^{n} \varphi_k(x_j) \varphi_i(x_j) = \sum_{j=1}^{n} \varphi_k(x_j) f(x_j) , \quad k = \overline{1,m} \tag{37}$$

elementului de cea mai bună aproximare pentru o funcţie dată.

4.5.2. Aproximarea funcţiilor prin metoda celor mai mici pătrate utilizând polinoame algebrice

Vom alege funcţiile standard ca polinoame algebrice, o primă posibilitate fiind ca:

$$\varphi_1(x) = 1; \quad \varphi_i(x) = x^{i-1}, \quad i \in \overline{2,m} \tag{41}$$

Ecuaţiile sistemului (37) (care se mai numesc şi ecuaţii normale) vor deveni:

$$\begin{cases} nc_1 + \sum_{i=2}^{m} c_i \sum_{j=1}^{n} x_j^{i-1} = \sum_{j=1}^{n} f_j \\ \sum_{i=1}^{m} c_i \sum_{j=1}^{n} x_j^{i} = \sum_{j=1}^{n} x_j f_j \\ ..... \\ \sum_{i=1}^{m} c_i \sum_{j=1}^{n} x_j^{m+i-2} = \sum_{j=1}^{n} x_j^{m-1} f_j \end{cases} \tag{42}$$

unde x_1, x_2, ..., x_n sunt punctele în care se cunoaşte fenomenul fizic a cărui modelare se studiază.

5. EVALUAREA NUMERICĂ A INTEGRALELOR

Introducere

Vom studia în continuare metode de evaluare aproximativă a integralelor definite $\int_a^b f(x)dx$ pentru o funcție f integrabilă a cărei primitivă este dificil de evaluat.

Aproximarea integralei definite se va face cu formule de forma:

$$\int_a^b f(x)dx = \sum_{i=0}^{n} c_i f(x_i) + E(f), \quad (1)$$

unde: c_i sunt coeficienți, x_i sunt noduri echidistante alcătuind o diviziune a intervalului [a, b] iar E(f) este restul metodei de aproximare numerică.

Pentru fiecare metodă restul (eroarea) este specific, iar coeficienții c_i şi nodurile x_i se aleg astfel încât restul să fie nul atunci când funcția se înlocuieşte cu un polinom de un anumit grad m.

Există o varietate de metode de aproximare a integralelor cunoscute şi sub numele de formule de cvadratură, clasificarea acestora făcându-se în funcție de modul de obținere.

a) Dacă pentru obținerea formulei de cvadratură (1) nodurile echidistante x_0, x_1, ..., x_n sunt fixate şi se determină coeficienții c_i, astfel încât E(f) să fie nul pentru orice f – polinom cu gradul cel mult egal cu m, ordinul de exactitate va fi m.

Câteva exemple de metode pe care le vom prezenta în continuare din cadrul acestei familii sunt: metoda trapezului, metoda Simpson, metoda lui Newton, cunoscute în general sub denumirea de formule de tip Newton-Côtes.

Aceste metode permit şi evaluarea integralelor din funcţii a căror expresie nu este cunoscută dar avem acces prin măsurători la valorile care ne interesează.

b) Dacă în formula (1) toţi coeficienţii c_i sunt egali: $c_i = c$, $i = \overline{0,n}$ şi se determină nodurile x_i astfel încât restul E(f) să fie nul pentru orice f – polinom de gradul $\leq m$, atunci se obţine o formulă de tip Cebîşev:

$$\int_a^b f(x)dx = c\sum_{i=0}^{n} f(x_i) + E(f)$$

de ordinul m.

c) Dacă în formula (1) se determină atât coeficienţii c_i cât şi valorile x_i iar restul va fi nul pentru orice f – polinom de grad maxim 2m obţinem formule de cvadratură de tip Gauss având ordinul de exactitate 2m.

Formulele de tip Newton-Côtes vor fi expuse în continuare iar cele de tip Cebîşev şi Gauss vor fi expuse în capitolul 7.

5.1. Metoda trapezului

Se obţine aproximând funcţia de integrat cu un polinom de interpolare Lagrange construit pe nodurile a şi b.

$$\text{Deci } \int_a^b f(x)dx \approx \frac{b-a}{2}\left[f(a)+f(b)\right]$$

Geometric, membrul drept reprezintă aria trapezului cu bazele $f(a)$, $f(b)$ şi înălţimea b – a – care se obţine aproximând curba dată pe [a, b] printr-o dreaptă (de aici denumirea metodei).

De la interpolarea Lagrange ştim că

$f(x) = L(x) + \frac{f''(\xi)}{2}(x-a)(x-b)$, în condiţia $f \in C^2[a, b]$, $\xi \in (a,b)$

$$\text{Deci } \int_a^b f(x)dx = \frac{b-a}{2}\left[f(a)+f(b)\right] + E_T(f),$$

cu $$E_T(f) = \frac{1}{2}\int_a^b f''(\xi)\,(x-a)(x-b)\,dx$$

eroarea de aproximare în formula trapezului.

De aceea vom considera o diviziune a intervalului [a, b], prin punctele echidistante (pentru comoditatea formulei pe care o vom stabili) $x_i = x_o + ih$, $i = \overline{0,n}$, unde $x_0 = a$, $x_n = b$, iar pasul $h = \frac{b-a}{n}$.

Obtinem: $\int_a^b f(x)dx = \frac{h}{2}\left[f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n) \right] + E_T(f)$,

unde $E_T(f) = -\frac{h^3}{12}\left[f''(\xi_1) + ... + f''(\xi_n)\right]$.

Cum f'' este continuă rezultă că $(\exists)\xi \in (a,b)$ astfel încât

$f''(\xi) = \frac{f''(\xi_1) + ... + f''(\xi_n)}{n}$ şi rezultă că

$E_T(f) = -\frac{h^3}{12} n f''(\xi) = -\frac{h^2}{12}(b-a) f''(\xi) = -\frac{(b-a)^3}{12n^2} f''(\xi)$

Notând cu $M_2 = \max_{x\in[a,b]} |f''(x)|$, obţinem:

$| E_T(f) | \le \frac{(b-a)^3 M_2}{12n^2}$,

deci putem mărgini eroarea în formula trapezului.

6. APROXIMAREA NUMERICĂ A SOLUŢIILOR ECUAŢIILOR DIFERENŢIALE

Introducere

Ecuaţiile diferenţiale ordinare sau cu derivate parţiale constituie modelele matematice pentru majoritatea problemelor inginereşti: studiul eforturilor la care sunt supuse elementele de rezistenţă: bare, grinzi, plăci subţiri, groase, conducte; studiul problemelor de câmp electric în dielectrici, câmp magnetic, câmp termic, propagarea undelor de toate felurile şi lista poate continua.

Odată stabilit fenomenul fizico-tehnic şi ecuaţiile diferenţiale care îl guvernează, ca formă, coeficienţi, condiţii la limită (pe frontieră) rămâne de rezolvat ultima problemă: rezolvarea acestui

model matematic. Din diverse motive: neomogenitățile fizice, frontiere cu geometrie dificilă, număr de necunoscute, etc., rezolvarea o vom face căutând o soluție aproximativă cu ajutorul unui calculator.

Vom expune în cadrul acestui capitol principalele metode aproximative care se pretează la implementarea pe calculator pentru rezolvarea ecuațiilor diferențiale.

Pentru ecuații diferențiale ordinare acestea se pot clasifica în două mari tipuri:

- metode unipas (Euler, Runge-Kutta) în care determinarea soluției în fiecare punct se va obține direct;

- metode multipas, sau predictor-corector în care valoarea soluției în fiecare punct se va „predicta“ şi apoi se va corecta iterativ.

Evident este vorba de soluții aproximative pe care nu avem cum să le comparăm cu o soluție exactă, deoarece practic aceasta este imposibil de găsit.

De aceea în practică trebuie să procedăm cu atenție pentru alegerea algoritmilor cei mai potriviți pentru problema concretă de rezolvat.

6.1. METODE DE TP EULER

6.1.1. Metoda Euler

Se consideră ecuația diferențială:

$$y' = f(x, y) \qquad (1)$$

cu condiția inițială:

$$y(x_0) = y_0, \qquad (2)$$

unde funcția f este definită într-un domeniu D din planul xOy.

Metoda lui Euler propune aproximarea soluției printr-o linie poligonală în care fiecare segment are direcția data de capatul sau stang. Astfel se consideră nodurile echidistante $x_i = x_0 + ih,\ i = \overline{0,n}$.

Considerând cunoscută aproximarea y_i a soluției problemei (1)+(2) în x_i, procedeul de aproximare Euler, poate fi acum rezumat astfel:

$$\begin{cases} f_i = f(x_i, y_i); \\ x_{i+1} = x_i + h; \\ y_{i+1} = y_i + hf_i; \end{cases} \tag{4}$$

Neglijarea termenilor de ordin superior în (4) face ca metoda să fie comodă în calcul, dar puțin precisă, erorile cumulându-se la fiecare pas.

Metoda se poate aplica şi dacă nodurile x_i nu sunt echidistante, având la fiecare iterație alt pas, h în acest caz.

6.1.2. Metoda Euler modificată

Considerăm pe $[x_i,\ x_{i+1}]$ ca direcție a segmentului M_iM_{i+1} direcția definită de punctul de la mijlocul segmentului (nu de extremitatea stângă ca în formula inițială) se obține metoda Euler modificată. Dacă x_i, y_i sunt valori calculate, procesul iterativ este următorul:

$$\begin{cases} x_i = x_0 + ih;\ f_i = f(x_i, y_i);\ x_{i+\frac{1}{2}} = x_i + \dfrac{h}{2}; \\ y_{i+\frac{1}{2}} = y_i + \dfrac{h}{2} f_i; f_{i+\frac{1}{2}} = f\left(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}} \right);\ y_{i+1} = y_i + hf_{i+\frac{1}{2}}; \end{cases}$$

6.1.4. Metoda Euler–Heun

Vom încheia trecerea în revistă a variantelor Euler pentru rezolvarea ecuațiilor de tipul (1) prezentând o ultimă variantă de ordinul doi a metodei lui Euler şi anume aceea propusă de Heun, [26].

Presupunând calculată valoarea y_i la pasul x_i, se propune pentru calcularea soluției la pasul x_{i+1} expresia:

$$y_{i+1} = y_i + \frac{h}{4}\left[f(x_i, y_i) + 3f\left(x_i + \frac{2}{3}h, y_i + \frac{2}{3}hf(x_i, y_i) \right) \right].$$

6.2. Metode de tip Runge-Kutta (R-K) pentru problema Cauchy

Introducere

Dacă metodele de tip Euler prezentate anterior au mai mult un caracter didactic pentru familiarizarea cititorului cu problematica, metodele de tip Runge Kutta pe care le vom expune în continuare pot fi utilizate cu succes în rezolvarea problemelor concrete din practica inginerească.

Metodele Runge-Kutta au trei proprietăți distincte [30]:

1. Sunt metode directe, adică pentru determinarea aproximării soluției la pasul i+1 avem nevoie de informațiile existente în punctul precedent x_i, y_i.

2. Sunt identice cu seriile Taylor până la termenii h^n, unde h este pasul curent iar n este diferit pentru metode diferite din această familie şi defineşte ordinul metodei.

Metodele de tip Euler pot fi şi ele incluse în familia Runge-Kutta şi putem astfel observa că metoda Euler este o metodă R-K de ordinul întâi iar metodele Euler-Cauchy şi Euler-Heun sunt metode R-K de ordinul 2.

3. Metodele de tip R-K necesită în procesul de calcul doar evaluarea funcției din membrul drept în diferite puncte nu şi a derivatelor ei. Aceasta este, alături de precizia bună a metodelor de ordin 3, 4, 5 motivul principal al utilizării lor frecvente în practică.

Oricum şi pentru metodele din această familie, formulele de evaluare a erorii păstrează doar un caracter teoretic neputând fi aplicate în practică decât pentru exemple simple cu caracter didactic.

O formulă Runge-Kutta de ordinul 2:

$$\begin{cases} y(x+h) = y(x) + \dfrac{1}{4}(k_1 + 3k_2) \\ k_1 = hf(x,y) \\ k_2 = hf\left(x + \dfrac{2}{3}h, y + \dfrac{2}{3}k_1\right) \end{cases} \qquad (10'')$$

cunoscută ca formula Euler-Heun.

O formulă de ordinul 3 este următoarea:

$$y(x+h) = y(x) + \frac{1}{4}(k_1 + 3k_3)$$

$$k_1 = hf(x,y)$$

$$k_2 = hf\left(x + \frac{h}{3}, y + \frac{k_1}{3}\right)$$

$$k_3 = hf\left(x + \frac{2h}{3}, y + \frac{2k_2}{3}\right)$$

Formule Runge-Kutta de ordinul 4:

$$y(x+h) = y(x) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = hf(x,y)$$

$$k_2 = hf\left(x + \frac{h}{2}, y + \frac{k_1}{2}\right)$$

$$k_3 = hf\left(x + \frac{h}{2}, y + \frac{k_2}{2}\right)$$

$$k_4 = hf(x + h, y + k_3)$$

formula *propriu-zisă Runge-Kutta* şi

$$y(x+h) = y(x) + \frac{1}{8}(k_1 + 3k_2 + 3k_3 + k_4)$$

$$k_1 = hf(x,y)$$

$$k_2 = hf\left(x + \frac{h}{3}, y + \frac{k_1}{3}\right)$$

$$k_3 = hf\left(x + \frac{2h}{3}, y - \frac{k_1}{3} + k_2\right)$$

$k_4 = hf(x + h, y + k_1 - k_2 + k_3)$

formula Kutta-Simpson

Rezolvarea sistemelor de ecuații diferențiale cu metoda Runge-Kutta

Metodele Runge-Kutta, prezentate pentru ecuații diferențiale, se pot extinde cu uşurinţă şi la sisteme de ecuații diferențiale.

Fie sistemul de ecuații diferențiale:

$$\begin{cases} y_1' = f_1(x, y_1, ..., y_n), \\ ..... \\ y_n' = f_n(x, y_1 ..., y_n) \end{cases} \qquad (12)$$

şi urmărim să determinăm soluția care satisface condițiile inițiale:

$y_i(x_0) = y_{i,0}$, $i = \overline{1,n}$ (13)

Presupunând că dispunem de soluția problemei (12) + (13) la pasul i: $y_{1,i}$, ..., $y_{n,i}$ metoda Runge-Kutta de ordinul 4 calculează soluția în pasul i+1 cu formulele:

$$\begin{aligned} &y_{1,i+1} = y_{1,i} + \Delta y_{1,i} \\ &..... \\ &y_{n,i+1} = y_{n,i} + \Delta y_{n,i} \end{aligned} \qquad (14)$$

Corecțiile: $\Delta y_{1,i}$, $\Delta y_{2,i}$, ... , $\Delta y_{n,i}$ care intervin în formulele (14) se calculează cu ajutorul relațiilor:

$$\Delta y_{1,i} = \frac{1}{6}k_1^1 + \frac{1}{3}k_2^1 + \frac{1}{3}k_3^1 + \frac{1}{6}k_4^1;$$

.....

$$\Delta y_{n,i} = \frac{1}{6}k_1^n + \frac{1}{3}k_2^n + \frac{1}{3}k_3^n + \frac{1}{6}k_4^n;$$

iar coeficienții $k_1^1,...,k_4^1,...,k_1^n,...,k_4^n$ au următoarea formă:

$k_1^j = hf_j(x_i, y_{1,i} ...., y_{n,i})$, $j = \overline{1,n}$;

$$k_2^j = hf_j\left(x_i + \frac{h}{2}, y_{1,i} + \frac{k_1^1}{2}, ..., y_{n,i} + \frac{k_1^n}{2}\right), j = \overline{1,n};$$

$$k_3^j = hf_j\left(x_i + \frac{h}{2}, y_{1,i} + \frac{k_2^1}{2}, ..., y_{n,i} + \frac{k_2^n}{2}\right), j = \overline{1,n};$$

$$k_4^j = hf_j\left(x_i + h, y_{1,i} + k_3^1, ..., y_{n,i} + k_3^n\right), j = \overline{1,n};$$

6.3. Metoda diferențelor finite pentru probleme Sturm-Liouville

Considerăm problema bilocală (Sturm-Liouville) dată de ecuația:

$u(x)y''(x) + v(x)y'(x) + w(x)y(x) = f(x)$ (24)

şi condițiile de limită

$$\begin{cases} y(a) = \alpha \\ y(b) = \beta \end{cases} \quad (25)$$

unde $x \in [a, b]$ şi presupunem că u, v, w, f sunt continue pe [a, b], $u(x) > 0$, $w(x) < 0$ $(\forall)$ $x \in [a, b]$ iar (24)+(25) are soluție unică pe [a,b].

Fie Δ o diviziune echidistantă de pas h a lui [a, b], $x_0 = a$, $x_i = a + ih$, $0 \le i \le n$ cu $x_n = b$.

Dacă în ecuația (24) facem $x = x_i$, $1 \le i \le n - 1$ şi folosim diferentele finite centrate pentru aproximarea operatorilor diferentiali avem

$$u(x_i)\frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{h^2} + v(x_i)\frac{y(x_{i+1}) - y(x_{i-1})}{2h} +$$

$+w(x_i)y(x_i) = f(x_i)$

Iar condițiile (25) devin $\begin{cases} y(x_0) = \alpha \\ y(x_n) = \beta \end{cases}$

Notând simplificat $u_i = u(x_i)$, $v_i = v(x_i)$, $w_i = w(x_i)$, $f_i = f(x_i)$, $y_i = y(x_i)$, $0 \le i \le n$, avem

$y_0 = \alpha$, $y_n = \beta$, iar

$$u_i \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + v_i \frac{y_{i+1} - y_{i-1}}{2h} + w_i y_i = f_i, \quad 1 \le i \le n\text{-}1$$

sau

$$\left(\frac{u_i}{h^2} - \frac{v_i}{2h}\right) y_{i-1} + \left(w_i - \frac{2u_i}{h^2}\right) y_i + \left(\frac{u_i}{h^2} + \frac{v_i}{2h}\right) y_{i+1} = f_i, \qquad (28)$$

$1 \le i \le n\text{-}1$, care reprezintă un sistem liniar de n-1 ecuaţii şi n-1 necunoscute y_1, y_2, ..., y_{n-1}, matricea sistemului fiind tridiagonală dominant diagonală.