

ANEXA A

FIȘIERE FUNCȚIE EXISTENTE ÎN MATLAB

A.1. Aproximarea funcțiilor

- **vander** – pentru calculul matricei Vandermonde. Sintaxa de apelare a acestei funcții este următoarea:

$$V = \text{vander}(X)$$

unde:

X – vector cu n elemente ce reprezintă numărul de noduri de interpolare în care funcția $f(x)$ este definită;

V – matricea Vandermonde rezultată.

- **interp1** – pentru interpolarea datelor după o lege liniară, cubică sau spline cubică. Sintaxa de apelare a funcției este următoarea:

$$y_k = \text{interp1}(x, y, x_k, 'regula')$$

unde:

x și y – vectori care conțin abscisele și ordonatele datelor;

x_k – vector pentru care se dorește determinarea valorilor de interpolare y_k ;

regula este reprezentată de unul dintre cuvintele cheie:

- *linear* – pentru interpolarea liniară;
- *spline* – pentru interpolarea spline cubică;
- *cubic* – pentru interpolarea cubică.

Toate legile de interpolare necesită ca variabila x să fie ordonată monoton. În plus, metoda *cubic* cere ca toate punctele de pe axa x să fie echidistante.

Dacă y este o matrice cu numărul de linii, n , egal cu cel al vectorului x și cu m coloane, funcția **interp1** returnează o matrice cu un număr de linii egal cu cel al vectorului x_k și cu m coloane.

- **polyfit** – pentru aproximarea unei funcții $f(x)$, date sub formă tabelară prin puncte, cu un polinom de grad n . Sintaxa de apelare a funcției este următoarea:

$$c = \text{interp1}(x, y, n)$$

unde:

x și y – vectori care conțin abscisele și ordonatele datelor;

n – gradul polinomului de aproximare;

c – coeficienții polinomului de aproximare.

Funcția **polyfit** returnează coeficienții c_k ai polinomului de aproximare, care în punctele precizate de vectorul x are, în sensul celor mai mici pătrate, valorile date de vectorul y .

A.2. Rezolvarea numerică a ecuațiilor algebrice și transcendente

Ecuațiile polinomiale pot fi rezolvate prin utilizarea funcției **roots**. Această funcție determină rădăcinile unui polinom, sintaxa de apelare fiind:

$$r = \text{roots}(c)$$

unde:

c – este un vector linie care conține coeficienții polinomului, în ordine descrescătoare a puterilor variabilei;

r – reprezintă un vector coloană care conține rădăcinile polinomului.

Pentru verificarea rezultatelor obținute se poate folosi funcția Matlab **poly**. Sintaxa funcției este:

$$c = \text{poly}(r)$$

Funcțiile **roots** și **poly** sunt funcții reciproce, realizând trecerea de la coeficienți la rădăcini sau de la rădăcini la coeficienți.

A.3. Rezolvarea numerică a sistemelor de ecuații liniare

A.3.1. Analiza matriceală

Calculul determinantului unei matrice A se face cu funcția **det**. Se apelează cu sintaxa:

$$D = \text{det}(A)$$

unde A este o matrice pătratică.

Rangul unei matrice reprezintă numărul de linii sau coloane liniar independente ale acesteia și se determină cu funcția **rank**. Aceasta se apelează cu ajutorul sintaxei:

$$r = \text{rank}(A)$$

Urma unei matrice reprezintă suma elementelor de pe diagonala principală. Aceasta se calculează cu funcția Matlab **trace**, care se apelează cu sintaxa:

$$u = \text{trace}(A)$$

Numărul de condiționare al unei matrice este dat de funcția **rcond**. Funcția se apelează cu sintaxa:

$$c = \text{rcond}(A)$$

Dacă matricea A este bine condiționară, funcția **rcond** are valoarea aproximativ 1, iar dacă este slab condiționată, funcția are valoarea aproape de 0.

Estimarea celui mai mic număr de condiționare se face cu funcția **condest**. Această funcție se apelează cu sintaxa:

$$cm = \text{condest}(A)$$

Pseudoinversa se utilizează, de obicei, pentru rezolvarea sistemelor de ecuații liniare, în sensul celor mai mici pătrate. Pseudoinversa unei matrice se calculează cu funcția Matlab **pinv**. Această funcție se apelează cu sintaxele:

$$X = \text{pinv}(A) \text{ sau } X = \text{pinv}(A, tol)$$

unde tol este toleranța, care are valoarea implicită (când lipsește la apelare):

$$tol = \max(\text{size}(A)) \cdot \text{norm}(A) \cdot \text{eps},$$

cu *eps*, depinzând de acuratețea calculelor în virgula mobilă.

A.3.2. Crearea matricelor diagonale și triunghiulare

Crearea unei matrice diagonale în Matlab se face cu funcția **diag**. Această funcție se apelează cu una dintre sintaxele:

$$Y = \mathbf{diag}(A) \text{ sau } Y = \mathbf{diag}(A, k)$$

unde:

A – este un vector sau o matrice asupra căreia se operează;

k – argument opțional care indică diagonală matricei/vectorului A , cu următoarea semnificație:

- $k = 0$, indică diagonală principală;
- $k > 0$, indică diagonală k de deasupra celei principale;
- $k < 0$, indică diagonală k de sub cea principală.

Pentru a putea crea o matrice superior sau inferior triunghiulară se pot folosi folosește funcțiile **tril** și **triu**, apelate cu una dintre sintaxele:

$$Y = \mathbf{tril}(A), \quad Y = \mathbf{tril}(A, k), \quad Y = \mathbf{triu}(A), \quad Y = \mathbf{triu}(A, k)$$

unde argumentele au aceeași semnificație ca mai sus.

Dacă A este un vector cu n componente, funcția **diag** apelată cu argumentele de intrare X și k generează o matrice pătrată de ordinul $n + \text{abs}(k)$, cu elementele lui A pe diagonală k . Prin folosirea sintaxei **diag(A)** se generează o matrice care are elementele lui A pe diagonală principală.

Dacă A este o matrice, funcția **diag** apelată cu parametrii de intrare A și k extrage un vector coloană format din elementele diagonalei k a matricei A . Apelată cu sintaxa **diag(A)**, funcția extrage elementele lui A de pe diagonală principală.

Funcția **triu(A)** extrage matricea superior triunghiulară din matricea A , în timp ce **triu(A, k)** înlocuiește cu zero toate elementele matricei A de sub diagonală k .

Funcția **tril(A)** extrage matricea inferior triunghiulară din matricea A , în timp ce **tril(A, k)** înlocuiește cu zero toate elementele matricei A de deasupra diagonalei k .

A.3.3. Factorizarea matricelor

Factorizarea LU

Factorizarea LU a unei matrice se face cu funcția Matlab **lu**. Aceasta se poate apela cu una sintaxa:

$$[L, U] = \text{lu}(A)$$

Astfel, în urma apelării se returnează o matrice superior triunghiulară U și o matrice inferior triunghiulară permutată L , astfel încât $[A] = [L] \cdot [R]$.

Factorizarea Cholesky

Această factorizare se calculează cu funcția Matlab **chol**. Se apelează cu sintaxele:

$$U = \text{chol}(A), \quad [U, p] = \text{chol}(A)$$

unde:

A – matrice pozitiv definită,

U – matrice superior triunghiulară, astfel încât $[A] = [U]^t \cdot [U]$;

p – scalar test, egal cu 0 dacă matricea A este pozitiv definită și un număr întreg pozitiv în caz contrar.

A.3.4. Inversarea matricelor

Inversa unei matrice se calculează cu funcția **inv**. Aceasta se apelează cu sintaxa:

$$Ai = \text{inv}(A)$$

Funcția se utilizează pentru rezolvarea sistemelor de ecuații lineare de tipul $[A] \cdot [X] = [b]$. Soluția sistemului este: $[X] = [A]^{-1} \cdot [b]$.

A.4. Derivarea numerică

A.4.1. Aproximarea derivatelor cu diferențe finite

Funcția **diff** evaluează diferența elementelor succesive ale unui vector sau ale coloanei unei matrice. Acesta se apelează cu una dintre sintaxele:

$$y = \mathbf{diff}(x) \text{ sau } y = \mathbf{diff}(x, n)$$

Dacă x este un vector linie sau coloană: $x = [x_1, x_2, \dots, x_n]$, funcția Matlab **diff**(x) returnează vectorul diferențelor elementelor adiacente, adică:

$$\Delta x = [x_2 - x_1, x_3 - x_2, \dots, x_n - x_{n-1}].$$

Vectorul rezultat Δx are cu un element mai puțin decât vectorul x .

Dacă x este o matrice, diferențele se calculează pe fiecare coloană, prin scăderea elementului din linia imediat superioară, conform relației:

$$\mathbf{diff}(x) = x(2 : m, :) - x(1 : m-1, :)$$

unde m este numărul de linii.

Apelată cu argumentul n , funcția returnează diferența de ordinul n a vectorului x .

Funcția **diff** permite aproximarea derivatei numerice a unei funcții $f(x)$ cu relația:

$$df = \mathbf{diff}(f) ./ \mathbf{diff}(x).$$

A.4.2. Calculul derivatelor unui polinom

Derivata unui polinom de orice grad în Matlab utilizează funcția **polyder**, a cărei sintaxă generală de apelare este:

$$Dp = \mathbf{polyder}(Cp)$$

unde:

Cp – este vectorul linie al coeficienților polinomului, în ordinea descrescătoare a puterilor variabilei;

Dp – este vectorul linie al coeficienților polinomului derivat.

Apelată cu sintaxa:

$$Dp = \mathbf{polyder}(A, B)$$

funcția **polyder** returnează în vectorul Dp coeficienții derivatei polinomului produs $A \cdot B$:

$$Dp(x) = (A(x) \cdot B(x))' = A'(x) \cdot B(x) + A(x) \cdot B'(x)$$

Apelată cu sintaxa:

$$[Mp, Np] = \text{polyder}(A, B)$$

funcția **polyder** returnează în vectorii Mp și Np coeficienții numărătorului și numitorului derivatei raportului polinoamelor A/B :

$$\frac{Mp(x)}{Np(x)} = \left(\frac{A(x)}{B(x)} \right)' = \frac{A'(x) \cdot B(x) - A(x) \cdot B'(x)}{(B(x))^2}$$

A.5. Integrarea numerică

Funcția **quad** este o funcție de integrare numerică (cuadratură) care folosește metoda Simpson și se apelează cu una dintre sintaxele:

$$I = \text{quad}('f', a, b), \quad I = \text{quad}('f', a, b, \text{eps})$$

unde:

f – este numele fișierului funcție care descrie funcția $f(x)$;

a, b – sunt limitele de integrare;

eps – este eroarea relativă admisă dintre doi pași consecutivi;

I – valoarea estimată a integralei.

Dacă funcția **quad** este apelată cu parametrul eps , integrarea funcției continuă să se rafineze până când eroarea relativă a estimării este mai mică decât precizia de calcul. Dacă precizia de calcul este nu este introdusă, valoarea implicită este 10^{-3} .

Funcția **quad** este utilizată dacă funcția $f(x)$ este exprimată sub formă analitică. Dacă funcția este descrisă prin valori numerice, se aproximează mai întâi acestea printr-o funcție corespunzătoare și apoi aceasta va fi scrisă în Matlab, ca fișier cu extensia *.m.

Funcția **trapz** realizează calculul numeric al integralei prin metoda trapezelor. Se apelează cu una dintre sintaxele:

$$I = \text{trapz}(x, f); \quad I = \text{trapz}(f)$$

unde:

x – este un vector coloană care conține abscisele;

f – este un vector coloană sau o matrice, cu același număr de linii ca și x , care conține valorile funcției corespunzătoare acelor abscise;

l – este un scalar dacă f este un vector sau un vector linie dacă f este o matrice, în care sunt returnate ariile calculate pentru fiecare coloană a lui f .

Dacă funcția **trapz** este apelată numai cu argumentul f , distanța dintre două puncte succesive este considerată 1. În acest caz limitele de integrare sunt $a = 1$ și $b = n$, unde n este numărul de elemente ale vectorului f .

Funcția **trapz** presupune că funcția $f(x)$ este dată sub formă tabelară, prin valori numerice în noduri echidistante ale intervalului de integrare. Dacă funcția este descrisă sub formă analitică $y = f(x)$, se creează doi vectori care conțin valorile perechilor $(x - y)$. Acest lucru se realizează prin evaluarea funcției $f(x)$ cu un pas adecvat, pentru x , și apoi se aplică funcția **trapz**.

A.6. Determinarea vectorilor și valorilor proprii

Valorile și vectorii proprii corespunzătoare unei matrice pătratice se determină cu ajutorul funcției **eig**. Aceasta se poate apela cu una dintre sintaxele:

$$V = \text{eig}(A) \quad (a)$$

$$[V, W] = \text{eig}(A) \quad (b)$$

$$[V, W] = \text{eig}(A, 'nobalance') \quad (c)$$

$$V = \text{eig}(A, B) \quad (d)$$

$$[V, W] = \text{eig}(A, B) \quad (e)$$

care conduc la următoarele rezultate:

- (a) returnează un vector ce conține valorile proprii ale matricei A ;
- (b) returnează o matrice diagonală W care conține valorile proprii ale matricei A și o matrice V ale cărei coloane sunt vectorii proprii corespunzători valorilor proprii, astfel încât:

$$A \cdot V = V \cdot W$$

- (c) returnează valorile și vectorii proprii fără a executa în prealabil o balansare. Dacă matricea A conține elemente mici, comparabile cu erorile de rotunjire, balansarea le scalează, făcându-le la fel de semnificative ca celelalte elemente ale matricei originale. Acest lucru conduce în final la vectori proprii incorecți. În astfel de situații se utilizează opțiunea "*nobalance*". Dacă matricea are numai valori proprii de ordinul întâi, atunci matricea originală este neregulată. Chiar dacă o matrice este neregulată, soluția funcției **eig** satisface ecuația:

$$A \cdot X = X \cdot W$$

Valorile și vectorii proprii generalizați determină soluțiile nebanale ale ecuației:

$$A \cdot X = \lambda \cdot B \cdot X$$

unde A și B sunt matrice pătratice de ordinul n , iar λ este un scalar. Valorile lui λ care satisfac ecuația se numesc valori proprii generalizate și valorile X corespunzătoare sunt vectorii proprii generalizați. Dacă B este o matrice nesingulară, adică determinantul acesteia este diferit de zero, problema determinării valorilor și vectorilor proprii se reduce de la o problemă standard de valori proprii prin înlocuirea lui A cu $B^{-1} \cdot A$, deoarece ecuația este echivalentă cu:

$$B^{-1} \cdot A \cdot X = \lambda \cdot X$$

- (d) returnează un vector care conține valorile proprii generalizate, dacă A și B sunt matrice pătratice.
- (e) returnează o matrice diagonală W ce conține valorile proprii generalizate și o matrice V ale cărei coloane sunt vectorii proprii corespunzători, astfel încât:

$$A \cdot V = B \cdot V \cdot W$$

A.7. Funcții Matlab generale

- **sin** – calculează sinusul argumentului;
- **cos** – calculează cosinusul argumentului;

- **asin** – calculează arcsinusul argumentului;
- **acos** – calculează arccosinusul argumentului;
- **sign** – returnează semnul argumentului;
- **exp** – calculează exponențiala e^x ;
- **log** – calculează logaritmul natural ($\ln$);
- **sqrt** – calculează radicalul de ordin 2;
- **abs** – calculează valoarea absolută (modulul) argumentului;
- **conj** – calculează conjugata complexă a argumentului;
- **real** – extrage partea reală a argumentului;
- **imag** – extrage partea imaginară a argumentului;
- **round** – returnează un număr întreg, rotunjit la cel mai apropiat întreg;
- **floor** – returnează un număr întreg, rotunjit la cel mai apropiat întreg spre $-\infty$;
- **ceil** – returnează un număr întreg, rotunjit la cel mai apropiat întreg spre $+\infty$;
- **rem** – returnează restul împărțirii argumentelor;
- **disp** – afișează o matrice sau un text;
- **norm** – calculează norma unui vector sau a unei matrice;
- **mean** – calculează valoarea medie;
- **median** – calculează valoarea mediană;
- **min** – determină cea mai mică componentă dintr-un vector sau matrice;
- **max** – determină cea mai mare componentă dintr-un vector sau matrice;
- **prod** – calculează produsul;
- **std** – calculează abaterea standard;
- **sum** – calculează suma;
- **sort** – sortează elementele în ordine crescătoare.
- **feval** – evaluează funcțiile Matlab.