LUCRAREA Nr. 7

INTERPOLAREA ŞI EXTRAPOLAREA FUNCȚIILOR DE O VARIABILĂ

1. Obiective

Prezentarea şi dezvoltarea funcțiilor Matlab pentru interpolarea şi extrapolarea funcțiilor de o variabilă, cu scopul însuşirii de către utilizatori.

2. Noțiuni teoretice

2.1. Interpolarea funcțiilor de o variabilă

Problema aproximării unei funcții de o variabilă trebuie rezolvată în diverse situații, următoarele două fiind mai frecvente:

• funcția este cunoscută, dar are o formă complicată, dificil de aplicat în calcule;

• funcția nu este complet cunoscută, fiind date numai valorile ei pe o mulțime discretă şi finită de puncte.

În primul caz, aproximarea se poate face destul de exact, restricțiile fiind legate de condiția ca, funcția care aproximează să fie cât mai simplă.

În al doilea caz informațiile sunt reduse şi se completează cu presupuneri suplimentare, privind gradul de regularitate al funcției. Este cazul prezentat în continuare.

Fiind dată o funcție $\boldsymbol{f}$, cunoscută prin măsurători efectuate asupra ei, în punctele $\boldsymbol{x_1,\ldots,x_n}$, numite noduri de interpolare, $\boldsymbol{y_i=f(x_i)}$, se cere să se determine valoare aproximativă a acestei funcții în punctul $\mathbf{a \neq x_i}$, **(∀) i=1÷ n**. Altfel spus, există un set de date care reprezintă coordonatele (x_i, y_i) şi se pune problema să se estimeze valorile funcției $f(x)$, pentru orice punct $x \in [x_1, x_n]$. Curba de interpolare trece prin toate punctele care o definesc, legea de interpolare între puncte putând fi liniară, cubică sau polinomială.

În Matlab sunt definite funcții care rezolvă problema aproximării funcțiilor prin *interpolare liniară* sau *interpolare polinomială sau cubică*.

2.1.1. Interpolare liniară

Dacă se presupune că funcția dintre două noduri de interpolare (x_1, x_2) poate fi exprimată printr-o linie dreaptă, atunci valoarea funcției în orice punct x (dintre cele două valori) se deduce cu expresia (Fig.1):

$$f(x) = y_1 + \frac{x - x_1}{x_2 - x_1} \cdot \left(y_2 - y_1\right)$$

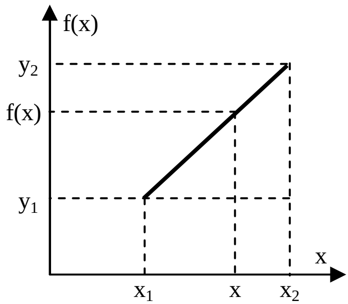


Fig. 1 *Interpolare liniară*

Pentru a realiza interpolarea liniară, în Matlab, se apelează funcția *interp1*, care determină valoarea de interpolare, *yi*, în punctul *xi* pentru vectorii de date *X* şi *Y*:

$yi = \text{interp1}(X, Y, xi)$ sau $yi = \text{interp1}(Y, xi)$

unde:

X – vector linie care conține abscisele punctelor inițiale ($x_1,...,x_n$), *în ordine strict crescătoare*;

Y – vector linie care conține ordonatele punctelor inițiale ($y_1,...,y_n$);

xi - vectorul care conține noile abscise ale punctelor în care se calculează funcția de interpolare ($xi \in \left[x_1, x_n\right]$);

y*i* - vectorul calculat de funcția de interpolare, asociat lui *xi*.

Sintaxa $yi = \text{interp1}(Y, xi)$ presupune că $x = 1 : n$, unde *n*=*length*(*Y*) este lungimea vectorului *Y* (dacă *Y* este vector), sau *n=size(Y,1)*, dacă *Y* este matrice.

2.1.2. Interpolare polinomială sau cubică

Pentru apelarea funcției de interpolare, alta decât cea liniară, se foloseşte sintaxa:

$$yi = \mathrm{interp1}(X, Y, xi, metoda)$$

unde *metoda* poate fi una din variantele:

'nearest'	Interpolarea celui mai apropiat vecin
'linear'	Interpolare liniară (implicit)
'spline'	Interpolare cubică spline
'pchip'	Interpolare cubică Hermite pe porțiuni
'cubic'	La fel ca 'pchip'
'v5cubic'	Interpolare cubică folosită in MATLAB 5

Observații:

- Pentru metodele 'nearest', 'linear' şi 'v5cubic', funcția $yi = \mathrm{interp1}(X, Y, xi, metoda)$ returnează NaN (nu este număr) pentru orice element *xi* care se află în afara intervalului definit de valorile vectorului *X*. Pentru toate celelalte metode, funcția *interp1* efectuează *extrapolarea* pentru valori *xi* în afara intervalului de valori ale lui *X*.
- Metodele 'nearest' şi 'linear' au implementări simple.
- Pentru metoda 'spline', funcția de interpolare *interp1* utilizează o funcție *spline* pentru a efectua interpolare spline cubică.
- Pentru metodele 'pchip' şi 'cubic', funcția *interp1* solicită o funcție care efectuează pe porțiuni interpolarea cubică în cadrul vectorilor X şi Y. Această metodă păstrează monotonia şi forma datelor.

2.2. Extrapolarea funcțiilor de o variabilă

Pentru extrapolarea funcțiilor de o variabilă se foloseşte funcția definită prin una din sintaxele:

$$yi = \mathrm{interp1}(X, Y, xi, metoda, 'extrap')$$

unde:

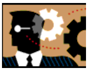
X – vector linie care conține abscisele punctelor inițiale ($x_1,...,x_n$), *în ordine strict crescătoare*;

Y – vector linie care conține ordonatele punctelor inițiale ($y_1,...,y_n$);

xi - vectorul care conține noile abscise ale punctelor în care se calculează funcția de extrapolare în afara domeniului de valori ale vectorului *X* ($xi \notin [x_1, x_n]$);

y*i* - vectorul calculat de funcția de extrapolare, asociat lui *xi*.

Sintaxa $yi = \mathrm{interp1}(X, Y, xi, metoda, 'extrapval')$ determină valorile extrapolate aflate în afara intervalului vectorului *X*. Răspunsurile NaN (nu este număr) şi 0 (zero) sunt deseori utilizate pentru această sintaxă.



3. Probleme de rezolvat

3.1. Interpolare liniară şi interpolare cubică

1. Să se aproximeze prin interpolare liniară şi apoi prin interpolare cubică şi să se reprezinte grafic vectorii de date **m** şi **n**, cu valorile de mai jos:

$m = 0 \ldots 10;\ n = [2\ 5\ \text{-}7\ 3\ 1\ 4\ \text{-}9\ 10\ 11\ 14\ 1];$

2. Să se aproximeze prin interpolare liniară şi apoi prin interpolare cubică şi să se reprezinte grafic vectorii de date **m** şi **n**, cu valorile de mai jos:

$m_1 := 1;\ m_2 := 1.7;\ m_3 := 2.1;\ m_4 := 2.5;\ m_5 := 3;$

$n_1 := 1.5;\ n_2 := 1.3;\ n_3 := 1.7;\ n_4 := 1.4;\ n_5 := 1.9.$

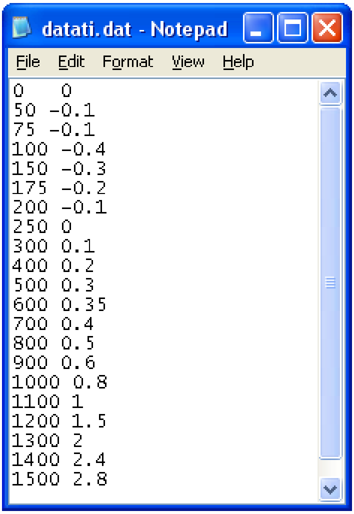
3. Să se aproximeze prin interpolare liniară şi apoi prin interpolare cubică şi să se reprezinte grafic vectorii de date **m** şi **n**, ale căror valorile sunt conținute în coloana 1, respectiv coloana 2 din matricea **A**:

$$A := \begin{pmatrix} 1 & 10 \\ 2 & 20 \\ 3 & 23 \\ 4 & 4 \\ 5 & 35 \\ 6 & 24 \\ 7 & 2 \\ 8 & 7 \end{pmatrix}.$$

4. Să se aproximeze prin interpolare liniară şi apoi prin interpolare cubică şi să se reprezinte grafic vectorii de date **m** şi **n**, ale căror valori sunt conţinute în prima coloană, respectiv a doua coloană din fişierul exterior de date cu numele **datati.dat**.

Pentru citirea datelor dintr-un fişier de date exterior (cu numele *datati.dat*) se va utiliza instrucţiunea Matlab, *load*, cu sintaxa:

```
load datati.dat
B=datati
m=B(:,1)
n=B(:,2)
```



3.2. Extrapolarea funcţiilor de o variabilă

Să se aproximeze prin extrapolare şi să se reprezinte grafic următoarele 30 de valori ale vectorului **v** care are expresia:

$$v_k := e^{-\frac{k}{100}} \cdot \sin\left(\frac{k}{10}\right), \quad k := 1..50.$$

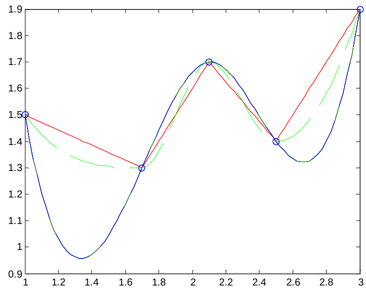
4. Probleme rezolvate

4.1. Se vor realiza cerinţele impuse ţinându-se cont de noţiunile teoretice prezentate în §2.1.

O variantă de program pentru cerințele de la punctul §3.1:

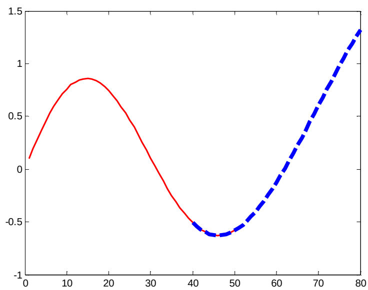
```
clear
x = 0:10;
y = [2 5 -7 3 1 4 -9 10 11 14 1];
xi = 0:.25:10;
yi = interp1(x,y,xi);
yi1 = interp1(x,y,xi,'spline');
plot(x,y,'o',xi,yi,'-r', xi,yi1,'--b');
hold on
yi1_s=spline(x,y,xi);
plot(x,y,'o',xi,yi1_s,'-k','LineWidth',1);
hold on
plot(x,y,'o',xi,yi1);hold on
yi2 = interp1(x,y,xi,'cubic');
plot(x,y,'o',xi,yi2,'--g',xi,yi3,'-.b');
hold on
plot(x,y,'o',xi,yi2,':g')
```

O variantă posibilă de răspuns poate fi cea prezentată în figură.



4.2. Se vor realiza cerințele impuse ținându-se cont de noțiunile teoretice prezentate în §2.2.

O variantă posibilă de răspuns poate fi cea prezentată în figură.



5. Probleme propuse