Calcul Numeric – Tema #2

Ex. 1 Să se creeze în Python procedura **InvDet**(A), care returnează inversa şi determinantul matricei A conform metodei învăţate la curs. Să se rezolve în Python sistemul de mai jos, folosind inversa matricei asociate sistemului. Să se afişeze inversa, determinantul şi soluţia sistemului.

V1

$$\begin{cases} 10x_1 + 30x_2 + 16x_3 = 118 \\ 2x_1 + 15x_2 + 7x_3 = 53 \\ 2x_1 + 5x_2 + 3x_3 = 21 \end{cases}$$

V2

$$\begin{cases} 3x_1 + 8x_2 + 5x_3 = 18 \\ 3x_1 + 28x_2 + 23x_3 = 76 \\ 3x_1 + 3x_2 + x_3 = 4 \end{cases}$$

V3

$$\begin{cases} 6x_1 + 5x_2 + 6x_3 = 9 \\ 2x_1 + x_2 + x_3 = 0 \\ 6x_1 + 5x_2 + 7x_3 = 10 \end{cases}$$

V4

$$\begin{cases} x_1 + x_2 + 5x_3 = 4 \\ 2x_1 + 17x_2 + 20x_3 = 33 \\ 2x_1 + 7x_2 + 13x_3 = 16 \end{cases}$$

V5

$$\begin{cases} 12x_1 + 9x_2 + 17x_3 = 31 \\ 4x_1 + 2x_2 + 5x_3 = 12 \\ 20x_1 + 22x_2 + 38x_3 = 50 \end{cases}$$

V6

$$\begin{cases} 9x_1 + 18x_2 + 19x_3 = 84 \\ 15x_1 + 13x_2 + 12x_3 = 47 \\ 3x_1 + 2x_2 + 2x_3 = 7 \end{cases}$$

Ex. 2 Se consideră sistemul $Ax = b$, $A = (a_{ij})_{i,j=\overline{1,n}} \in \mathcal{M}_n(\mathbb{R})$ o matrice tridiagonală şi $b \in \mathbb{R}^n$. Pe diagonala principală are elementele $a_{ii} = a_i, i = \overline{1,n}$, deasupra diagonalei principale are elementele $a_{i,i+1} = b_i, i = \overline{1,n-1}$ şi sub diagonala principală are elementele $a_{i+1,i} = c_i, i = \overline{1,n-1}$. Presupunem că matricea poate fi descompusă în produs de două matrice $A = LR$, unde $L = (\ell_{ij})_{i,j=\overline{1,n}}$ este inferior triunghiulară şi $R = (r_{ij})_{i,j=\overline{1,n}}$ este superior triunghiulară având următoarele elemente:

- elementele diagonale ale matricei L sunt egale cu 1, i.e. $\ell_{ii} = 1, i = \overline{1,n}$
- sub diagonala principală se află elementele notate cu $\ell_i, i = \overline{1,n-1}$, i.e. $\ell_{i+1,i} = \ell_i$
- restul elementelor în matricea L sunt nule

- pe diagonala principală a matricei R se află numerele $r_i, i = \overline{1,n}$, i.e. $r_{i,i} = r_i$
- deasupra diagonalei principale se află numerele $s_i, i = \overline{1,n-1}$, i.e. $r_{i,i+1} = s_i$
- restul elementelor neprecizate din R sunt nule

(a) Să se deducă următoarea schemă numerică de determinare a matricelor L şi R

$$\begin{cases} r_1 = a_1 \\ for\ i = 1 : n-1\ do \\ \quad \ell_i = \dfrac{c_i}{r_i} \\ \quad s_i = b_i \\ \quad r_{i+1} = a_{i+1} - \ell_i s_i \\ endfor \end{cases} \tag{1}$$

Indicaţie: Produsul matriceal $A = LR$ poate fi scris pe componente $a_{ij} = \sum_{k=1}^{n} \ell_{ik} r_{kj}$.
Se vor calcula elementele $b_i := a_{i,i+1}, a_{i+1} := a_{i+1,i+1}, c_i := a_{i+1,i}$ folosind formula anterioară păstrând doar acei termeni din sumă care sunt nenuli.

(b) Sistemul $Ax = b$ se rezolvă folosind descompunerea $A = LR$ după cum urmează

- se rezolvă sistemul $Ly = b$
- cu y calculat anterior se rezolvă sistemul $Rx = y$ şi se calculează soluţia x

Se se deducă formulele de calcul pentru y_i şi x_i, $i = \overline{1,n}$. Nu se vor aplica metodele subsitituţiei descendente, respectiv ascendente.

Ex. 3 Să se rezolve în Python următorul sistem tridiagonal, folosind metoda de factorizare LR. Să se verifice soluţia.

$$\begin{cases} dx_1 + fx_2 = 2 \\ cx_1 + dx_2 + fx_3 = 1 \\ \qquad cx_2 + dx_3 + fx_4 = 1 \\ \dots \\ \qquad\qquad cx_{n-2} + dx_{n-1} + fx_n = 1 \\ \qquad\qquad\qquad cx_{n-1} + dx_n = 2 \end{cases}$$

V1 $d = 12, f = -4, c = -4, n = 20$

V2 $d = 15, f = -5, c = -4, n = 15$

V3 $d = 18, f = -6, c = -6, n = 20$

V4 $d = 9, f = -3, c = -3, n = 15$

V5 $d = 21, f = -7, c = -7, n = 20$

V6 $d = 7, f = -3, c = -1, n = 20$

Ex. 4 Fie matricea simetrică $A = \begin{pmatrix} a_1 & a_2 & a_3 & ... & a_n \\ & a_1 & a_2 & ... & a_{n-1} \\ & & a_1 & ... & a_{n-2} \\ SYM & & & \ddots & \\ & & & & a_1 \end{pmatrix}$

1. Să se definească în Python vectorul a şi matricea simetrică A;
2. Să se construiască în Python procedura **FactCholesky** conform sintaxei **FactCholesky**(A, b) conform. Procedura **FactCholesky** returnează matricea L.
3. Să se afle factorizarea Cholesky a matricei A;
4. Să se rezolve sistemul $Ax = b$ conform metodei Cholesky.
5. Să se afişeze matricea L şi soluţia sistemului.

V1 $n = 5, b_i = i^2, i = \overline{1,n}, a = (2n, 2n-2, ..., 2)^T$

V2 $n = 8, b_i = i^3, i = \overline{1,n}, a = (3n, 3n-3, ..., 3)^T$

V3 $n = 10, b_i = i^3 + 2, i = \overline{1,n}, a = (n, n-1, ..., 1)^T$

V4 $n = 5, b_i = i^4, i = \overline{1,n}, a = (2^n, 2^{n-1}, ..., 2^1)^T$

V5 $n = 6, b_i = i^4, i = \overline{1,n}, a = (4^n, 4^{n-1}, ..., 4^1)^T$

V6 $n = 6, b_i = i^2, i = \overline{1,n}, a = (3^n, 3^{n-1}, ..., 3^1)^T$

Ex. 5 Fie $A = \begin{pmatrix} 2 & 3 & 1 \\ 4 & 8 & 5 \\ -4 & 0 & 10 \end{pmatrix}$

a) Să se determine matricele L, U conform metodei de factorizare LU, folosind pe rând metodei Gauss cu pivotare parţială.

b) Să se rezolve sistemul $Ax = b$, $b = (5, 18, 20)^T$ folosind factorizarea LU.

Ex. 6 Fie $A = \begin{pmatrix} 4 & 2 & 4 \\ 2 & 2 & 3 \\ 4 & 3 & 14 \end{pmatrix}$

a) Să se verifice dacă A este simetrică şi pozitiv definită.

b) În caz afirmativ, să se determine factorizarea Cholesky.

c) Să se rezolve sistemul $Ax = b, b = (10, 6, 11)^T$, folosind descompunerea LL^T.

Ex. 7 Să se afle polinomul de interpolare Lagrange $P_2(x)$ al funcţiei $f(x)$, conform metodelor Lagrange, Newton şi Newton cu diferenţe divizate, relativ la diviziunea $x = (x_1, x_2, x_3)$. Să se afle eroarea de trunchiere si o estimare a acesteia. Pentru discretizarea echidistantă să se studieze ce se întmplă cu eroarea de trunchiere când n creşte foarte mult.

a) $f(x) = lnx$, $x = (1, e, e^2)$

b) $f(x) = sin(\frac{\pi x}{2}), x = (-1, 0, 1)$

Ex. 8 Să se arate că polinomul Lagrange care interpolează datele $(-2,1),(-1,4)$, $(0,11),(1,16)$, $(2,13),(3,-4)$ este de gradul 3. Se va folosi metoda Newton DD.

Ex. 9 Fie $P_3(x)$ polinomul de interpolare Lagrange asociat setului de date $(0,0),(0.5,y),(1,3),(2,2)$. Să se afle y ştiind că, coeficientul lui x^3 în reprezentarea polinomului $P_3(x)$ este 6. Pentru reprezentarea polinomului $P_3(x)$ se va folosi metoda Lagrange.

Ex. 10 Următorul tabel se referă la polinomul de grad necunoscut, $P_n(x)$

x	0	1	2
$P_n(x)$	2	-1	4

Să se determine coeficientul lui x^2 în reprezentarea $P_n(x)$ dacă toate diferenţele divizate de ordinul 3 sunt 1.

Ex. 11 Fiind dat tabelul diferenţelor divizate

x_i	DD ordin 0	DD ordin 1	DD ordin 2
$x_1=0$	$\boxed{f[x_1]=f(x_1)}$		
$x_2=0,4$	$f[x_2]=f(x_2)$	$\boxed{f[x_1,x_2]}$	
$x_3=0,7$	$f[x_3]=f(x_3)=6$	$f[x_2,x_3]=10$	$\boxed{f[x_1,x_2,x_3]=\frac{50}{7}}$

să se scrie polinomul $P_2(x)$.

Ex. 12
a) Să se construiască în Python procedurile **Directa**(X,Y,x), **Lagrange**(X,Y,x), **MetNDD**(X,Y,x) conform metodelor directă, Lagrange şi Newton DD. Vectorii X,Y reprezintă nodurile de interpolare, respectiv valorile funcţiei f în nodurile de interpolare. Procedurile **Directa**(X,Y,x), **Lagrange**(X,Y,x), **MetNDD**(X,Y,x) returnează valoarea polinomului de interpolare $y=P_n(x)$ conform metodelor precizate.

b) Să se construiască în Python în aceeaşi figură, graficele funcţiei f pe intervalul $[a,b]$, punctele $(X_i,Y_i), i=\overline{1,n+1}$ şi polinomul $P_n(x)$ obţinut în baza celor 3 metode de la a). Reprezentaţi grafic într-o altă figură eroarea $|e_t(x)|=|f(x)-P_n(x)|$.

Obs.: În cazul în care nu este dată funcţia nu se va reprezenta grafic şi nu se va calcula eroarea.

V1 $f(x)=sin3x, [a,b]=[1;\ 2,2], n=4$ (se consideră discretizare echidistantă)

V2 $X=(2,\ 3,\ 5,\ 8,\ 12), Y=(10,\ 15,\ 25,\ 40,\ 60)$

V3 $f(x)=e^x, [a,b]=[-1,1], n=4$ (se consideră discretizare echidistantă)

V4 $X=(0.4,\ 0.5\ 0.7\ 0.8), f(x)=lnx$

V5 $X=(0,\ 1,\ 3,\ 6\ ), Y=(18,\ 10,\ -18,90)$

V6 $f(x)=e^{3x}, [a,b]=[-1,1], n=5$ (se consideră discretizare Chebyshev)

Ex. 13
a) Să se scrie definiţia funcţiei de interpolare liniară.

b) Să se afle funcţia de interpolare spline liniară S pentru funcţia $f(x)=lnx$ relativ la diviziunea $(1,e,e^2)$.

Ex. 14 Fie f definită pe intervalul $[a, b]$ şi datele $a = x_1 < x_2 < x_3 = b$. Funcţia spline pătratică restricţionată pe fiecare interval în parte reprezintă un polinom de gradul doi; $S_1(x) = a_1+b_1(x-x_1)+c_1(x-x_1)^2, \quad x \in [x_1, x_2)$, $S_2(x) = a_2+b_2(x-x_2)+c_2(x-x_2)^2, \quad x \in [x_2, x_3]$. Să se afle coeficienţii $a_1, a_2, b_1, b_2, c_1, c_2$, dacă $S'(x_3) = f'(x_3)$.

Ex. 15 Să se scrie definiţia funcţiei de interpolare spline cubică. Să se determine funcţia spline cubică naturală S care interpolează datele: $f(0) = 0, f(1) = 1, f(2) = 2$.

Ex. 16 Să se determine funcţia spline cubică cu constrângeri S care interpolează datele:

a) $(1, 2), (2, 3), (3, 5),\ S'(1) = 2, S'(3) = 1$.

b) $(0, 0), (1, 1), (2, 2), S'(0) = 1, S'(2) = 1$.

(Se vor folosi direct formulele de calcul pentru coeficienţii a_j, b_j, c_j, d_j)

c) Să se verifice că funcţia spline cubică S calculată la punctele a) şi b) satisface condiţiile definiţiei unei funcţii spline cubice.

Ex. 17 Să se determine coeficienţii B, C, D, astfel încât funcţia

$$S(x) = \begin{cases} S_1(x) = 1 + 2x - x^3, x \in [0, 1) \\ S_2(x) = 2 + B(x-1) + C(x-1)^2 + D(x-1)^3, x \in [1, 2] \end{cases}$$

să reprezinte o funcţie spline naturală (i.e. $S''(a) = S''(b) = 0$, a, b fiind capetele intervalului de interpolare).

Ex. 18 Fie funcţia $S(x)$ o funcţie spline cubică cu constrângeri la capete (i.e. $S'(a) = f'(a), S'(b) = f'(b)$, unde a, b sunt capetele intervalului de interpolare):

$$S(x) = \begin{cases} S_1(x) = 3(x-1) + 2(x-1)^2 - (x-1)^3, x \in [1, 2) \\ S_2(x) = A + B(x-2) + C(x-2)^2 + D(x-2)^3, x \in [2, 3] \end{cases}$$

Să se afle A, B, C, D, dacă $f'(1) = f'(3)$.

Ex. 19 O funcţie spline cubică cu constrângeri este definită prin

$$S(x) = \begin{cases} S_1(x) = 1 + Bx + 2x^2 - 2x^3, x \in [0, 1) \\ S_2(x) = 1 + b(x-1) - 4(x-1)^2 + 7(x-1)^3, x \in [1, 2] \end{cases}$$

Să se afle $f'(0)$ şi $f'(2)$.

Ex. 20 O funcţie spline cubică naturală este definită prin

$$S(x) = \begin{cases} S_1(x) = 1 + B(x-1) - D(x-1)^3, x \in [1, 2) \\ S_2(x) = 1 + b(x-2) - \dfrac{3}{4}(x-2)^2 + d(x-2)^3, x \in [2, 3] \end{cases}$$

Să se afle B, D, b, d, ştiind că S interpolează datele $(1, 1), (2, 1)$ şi $(3, 0)$.

Ex. 21 Fie $f : [a, b] \to \mathbb{R}$ o funcţie continuă.

a) Să se construiască în Python procedura **SplineL** având sintaxa $y =$**SplineL**(X, Y, x), conform metodei de interpolare spline liniară. Datele de intrare: vectorul X, componentele căruia sunt nodurile de interpolare, i.e. $a = X_1 < X_2 < ... < X_{n+1} = b$;

vectorul Y definit prin $Y_i = f(X_i), i = \overline{1, n+1}$; variabila scalară $x \in [a, b]$. Datele de ieşire: Valoarea numerică y reprezentând valoarea funcţiei spline liniară $S(x)$ calculată conform metodei spline liniare.

b) Fie datele: $f(x) = sin(x),\ x \in [-\frac{\pi}{2}, \frac{\pi}{2}]$; $n = 2, 4, 10$; X - o diviziune echidistantă a intervalului $[-\frac{\pi}{2}, \frac{\pi}{2}]$ cu $n+1$ noduri; $Y = f(X)$. Să se construiască grafic funcţia f, punctele de interpolare (X, Y) şi un vector S calculat conform procedurii **SplineL**, corespunzător unei discretizări x a intervalului $[-\frac{\pi}{2}, \frac{\pi}{2}]$ cu 100 de noduri. Ind.: $S_i =$ **SplineL**$(X, Y, x_i), i = \overline{1, 100}$.

e) Să se modifice procedura $y =$**SplineL**(X, Y, x), astfel încât atât parametrul de intrare x, cât şi parametrul de ieşire y să fie vectori.

Ex. 22 Fie $f : [a, b] \to \mathbb{R}$ o funcţie continuă.

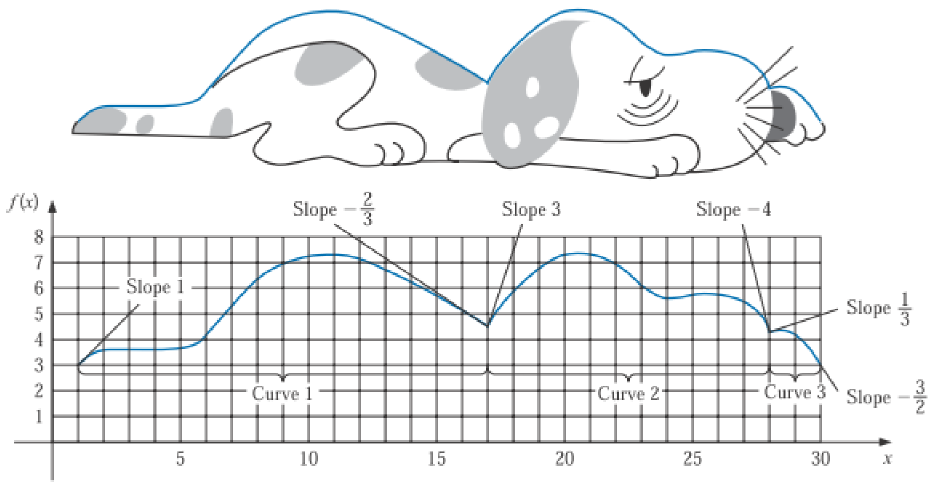
a) Să se construiască în Python procedura **SplineP** având sintaxa $[y, z] =$**SplineP**(X, Y, fpa, x), conform metodei de interpolare spline pătratică. Datele de intrare: vectorul X, componentele căruia sunt nodurile de interpolare, i.e. $a = X_1 < X_2 < ... < X_{n+1} = b$; vectorul Y definit prin $Y_i = f(X_i), i = \overline{1, n+1}$; derivata funcţiei f în capătul din stânga a intervalului, $fpa = f'(a)$; variabila scalară $x \in [a, b]$. Datele de ieşire: Valorile numerice y, z reprezentând valoarile funcţiei spline pătratică $S(x)$ şi derivatei $S'(x)$ calculate conform metodei spline pătratice. Indicaţie: $z = b_j + 2c_j(x - x_j)$.

b) Fie datele: $f(x) = sin(x),\ x \in [-\frac{\pi}{2}, \frac{\pi}{2}]$; $n = 2, 4, 10$; X - o diviziune echidistantă a intervalului $[-\frac{\pi}{2}, \frac{\pi}{2}]$ cu $n+1$ noduri; $Y = f(X)$. Să se construiască grafic funcţia f, punctele de interpolare (X, Y) şi functia spline $S(x)$ calculată conform procedurii **SplineP**, corespunzător unei discretizări x a intervalului $[-\frac{\pi}{2}, \frac{\pi}{2}]$ cu 100 de noduri.

c) Într-o altă figură să se construiască grafic derivata funcţiei spline şi derivata funcţiei f.

d) Să se modifice procedura $[y, z] =$**SplineP**(X, Y, fpa, x), astfel încât parametrii de intrare/ieşire x şi respectiv y, z să poată fi vectori.

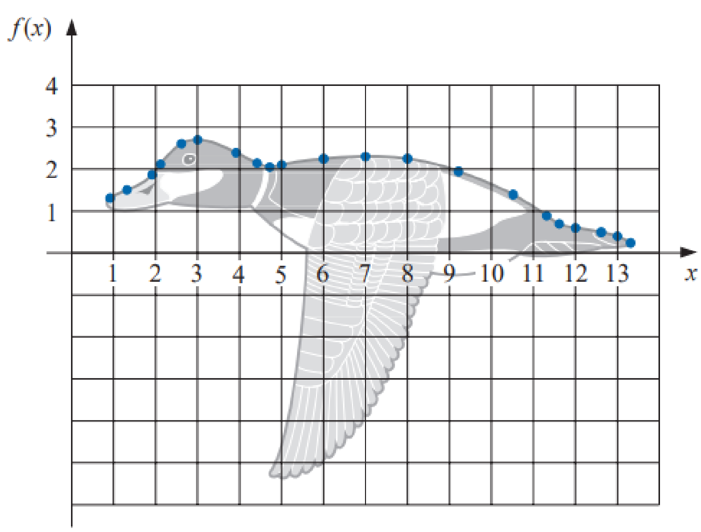
Ex. 23 Porţiunea de sus a acestui căţel este cel mai bine aproximată de funcţii spline cubice cu constrângeri. Folosiţi tabelul de mai jos, obţinut cu ajutorul graficului, pentru a construi cele trei curbe spline cubice cu constrângeri. Pentru reprezentarea grafică veţi considera aceeaşi scară pe cele două axe.



Curve 1				Curve 2				Curve 3			
i	x_i	$f(x_i)$	$f'(x_i)$	i	x_i	$f(x_i)$	$f'(x_i)$	i	x_i	$f(x_i)$	$f'(x_i)$
0	1	3.0	1.0	0	17	4.5	3.0	0	27.7	4.1	0.33
1	2	3.7		1	20	7.0		1	28	4.3	
2	5	3.9		2	23	6.1		2	29	4.1	
3	6	4.2		3	24	5.6		3	30	3.0	−1.5
4	7	5.7		4	25	5.8					
5	8	6.6		5	27	5.2					
6	10	7.1		6	27.7	4.1	−4.0				
7	13	6.7									
8	17	4.5	−0.67								

Ex. 24 Construiţi porţiunea de sus a raţei sălbatice folosind funcţii spline cubice cu constrângeri. Pentru reprezentarea grafică veţi considera aceeaşi scară pe cele două axe. Deasemenea, să se construiască aceeaşi curbă folosind însă de data aceasta una dintre metodele de interpolare Lagrange. Ce se observă conform graficelor? Care dintre metode aproximează mai bine?

x	0.9	1.3	1.9	2.1	2.6	3.0	3.9	4.4	4.7	5.0	6.0	7.0	8.0	9.2	10.5	11.3	11.6	12.0	12.6	13.0	13.3
$f(x)$	1.3	1.5	1.85	2.1	2.6	2.7	2.4	2.15	2.05	2.1	2.25	2.3	2.25	1.95	1.4	0.9	0.7	0.6	0.5	0.4	0.25



Ex. 25 Fie forma pătratică $f(x,y) = \dfrac{1}{2} < A\begin{pmatrix} x \\ y \end{pmatrix}, \begin{pmatrix} x \\ y \end{pmatrix} > - < \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, \begin{pmatrix} x \\ y \end{pmatrix} >$. Să se scrie matricea A şi vectorul b precizându-se natura matricei A şi natura punctului de extrem.

V1 $f(x,y) = x^2 + xy - 2x + y^2 - 3y$

V2 $f(x,y) = \dfrac{3}{2}x^2 + 2xy - x + \dfrac{3}{2}y^2 - 3y$

V3 $f(x,y) = 2x^2 + xy - x + 2y^2 - 2y$

V4 $f(x,y) = 2x^2 + 2xy + x + 2y^2 - 2y$

V5 $f(x,y) = 2x^2 + 3xy + x + 2y^2 - y$

V6 $f(x,y) = \dfrac{5}{2}x^2 + xy + x + \dfrac{5}{2}y^2 - y$

Ex. 26 Pentru datele de la Exerciţiul de mai sus şi domeniul $(x,y) \in [a,b]\times[c,d]$ să se implementeze în Python următoarele cerinţe:

- Să se construiască suprafaţa $z = f(x,y)$ pe domeniul dat;
- Să se afle conform Gradientului Conjugat $x^{(1)}, x^{(2)}$, dacă $x^{(0)} = (a,c)^T$.
- Să se reprezinte pe suprafaţă punctul de minim.
- Să se construiască într-o altă figură curbele de nivel care trec prin punctele $x^{(k)}, k = 0,1,2$, precum şi traseul deplasărilor până la punctul de minim.

Ex. 27 Folosind aceleaşi date ca la metoda gradientului conjugat să se afle punctul de minim local conform metodei pasului descendent. Să se construiască punctul de minim pe suprafaţa $z = f(x,y)$. Într-o altă figură să se reprezinte traseul format din punctele calculate la toate iteraţiile, liniile de nivel care trec prin punctele respective şi direcţiile gradientului de la fiecare iteraţie.

Ex. 28 Să se deducă o formulă de aproximare a derivatei $f'(x)$ care foloseşte termenii $f(x-h), f(x+h), f(x+2h), f(x+3h)$. Care este ordinul de aproximarea a formulei obţinute? Ind.: Se scrie pentru fiecare termen dezvoltarea în serie Taylor până la ordinul 4. Se consideră combinatia $Af(x-h)+Bf(x+h)+Cf(x+2h)+Df(x+3h)$ şi se scriu 4 relaţii în raport cu A,B,C,D astfel încât coeficienţii de pe lângă $f''(x), f'''(x), f^{(IV)}(x)$ să se anuleze, iar coeficientul termenului $f'(x)$ să fie egal cu 1. Se rezolvă sistemul în A,B,C,D şi se extrage expresia derivatei $f'(x)$.

Ex. 29 Să se deducă o formulă care aproximează $f'''(x)$ cu eroarea de aproximare $\mathcal{O}(h^2)$. Ind.:Se rezolvă similar cu exerciţiul anterior.

Ex. 30 Presupunem că $\varphi(h)$ este aproximarea derivatei $f'(x_0)$ de ordinul $\mathcal{O}(h)$ şi $f'(x_0) = \varphi(h) + a_1h+a_2h^2+a_3h^3+...$, unde a_1, a_2, a_3 sunt constante. Să se folosească valorile $\varphi(h), \varphi(h/3), \varphi(h/9)$ pentru a se construi o formulă de aproximare de ordinul $\mathcal{O}(h^3)$.

Ex. 31 Dacă se aplică formula trapezului calculului integralei $\int_0^2 f(x)dx$ se obţine valoarea 5, iar dacă se aplică formula dreptunghiului, se obţine 4. Ce valoare va da formula Simpson pentru calculul integralei.

Ex. 32 Formula trapezului aplicată $\int_0^2 f(x)dx$ dă valoarea 4, iar formula lui Simpson dă voloarea 2. Care este valoarea $f(1)$.

Ex. 33 Fiind date $f(0) = 1, f(0.5) = 2.5, f(1) = 2, f(0.25) = f(0.75) = \alpha$. Să se afle α dacă, folosind formula trapezului sumată cu $m = 4$, se obţine $\int_0^1 f(x)dx \approx 1.75$.

Ex. 34 Să se deducă formula cuadraturii Newton-Cotes închisă ($n = 3$). Această formulă se mai numeşte şi formula de cuadratură Newton. Să se deducă formula de cuadratură sumată Newton. **Obs.:** Pentru calculul coeficienţilor w_k folosiţi calculul simbolic din **SymPy** pentru evaluarea simbolică a integralelelor.

Ex. 35

a) Să se construiască în Python procedura **Integrare**, având sintaxa

$I =$**Integrare**$(f, a, b, m, metoda)$, care calculează valoarea aproximativă a integralei $I(f) = \displaystyle\int_a^b f(x)dx$ conform formulelor de cuadratură sumate a dreptunghiului, trapezului, Simpson şi Newton. Variabila *metoda* este un şir de caractere din mulţimea { 'dreptunghi', 'trapez', 'Simpson','Newton'}.

b) Să se afle valoarea exactă a integralei folosind calculul simbolic din **SymPy**.

c) Să se calculeze erorile absolute $|I(f) - I_{dreptunghi}|$, $|I(f) - I_{trapez}|$, $|I(f) - I_{Simpson}|$, $|I(f) - I_{Newton}|$, unde $I_{dreptunghi}, I_{trapez}, I_{Simpson}, I_{Newton}$ sunt valorile aproximative ale integralei calculate conform metodelor menţionate la punctul a).

Obs.: Datele problemei le veţi alege individual.

Ex. 36 Se consideră trei funcţii: o funcţie liniară, una pătratică si respectiv, una exponenţială. Fiecare student îşi va alege în mod individual câte o funcţie din fiecare categorie. În baza acestora se vor construi trei tabele de date $(x_i, y_i), i = \overline{1, n}$, câte unul pentru fiecare funcţie. Datele y_i vor fi perturbate, pentru fiecare caz în parte, cu valori intre 0 şi 1, calculate aleator cu funcţia **random.rand** din **NumPy**. Se vor calcula, pentru fiecare caz în parte, curbele de regresie care aproximează datele perturbate, conform metodei celor mai mici pătrate. Se vor reprezinta grafic atât punctele perturbate, cât şi curbele de regresie liniară, pătratică şi respectiv exponenţială care s-au obţinut. Fiecare tabel de date va conţine 20 de puncte.

Câteva probleme de pregătire Python

Problemele de mai jos nu sunt obligatorii, ci sunt recomandate pentru exersarea programului Python. Pot constitui unul dintre subiectele probei practice.

Problema #1 Se consideră ecuaţia

$$ax^2 + bx + c = 0, a \neq 0$$

Să se definească o procedură în Python care rezolvă problema ecuaţiei de gradul 2. Funcţia va avea drept parametrii de intrare coeficienţii a, b, c, iar parametrii de ieşire vor fi soluţiile reale sau complexe x_1, x_2 ale ecuaţiei. Să se testeze pentru datele:

V1. $a = 2, b = 5, c = 2$

$a = 2, b = 4, c = 2$

$a = 2, b = 4, c = 3$

V2. $a = 1, b = -3, c = 2$

$a = 1, b = -2, c = 2$

$a = 1, b = -2, c = 1$

V3. $a = 1, b = -5, c = 6$

$a = 1, b = -4, c = 4$

$a = 1, b = -4, c = 5$

V4. $a = 1, b = -9, c = 20$

$a = 1, b = -8, c = 16$

$a = 1, b = 2, c = 2$

V5. $a = 3, b = 4, c = 1$

$a = 3, b = 6, c = 3$

$a = 3, b = 4, c = 3$

V6. $a = 1, b = 4, c = 2$

$a = 2, b = 4, c = 2$

$a = 1, b = -1, c = 2$

Problema #2 Se consideră funcţia $f : \mathbb{R} \to \mathbb{R}, y = f(x)$. Să se definească o procedură care evaluează funcţia într-un singur punct. Să se verifice pentru cazurile $x = a, x = b$. Să se construiască grafic funcţia $f(x)$ pe intervalul `[xmin,xmax]`, folosind comanda `plot` cu 100 de puncte. Să se formateze graficul funcţiei executând următoarele acţiuni:

- Afişarea titlului figurii: `Funcţie definită pe ramuri.`
- Etichetarea axei Ox: `x`
- Etichetarea axei Oy: `y`
- Afişarea liniilor de grilă
- Afişarea legendei

V1. $f(x) = \begin{cases} e^x, x > 0 \\ e, x = 0 \\ e^{-x}, x < 0 \end{cases}$

$a = 2, b = 0,$ `[xmin,xmax]` $= [-1, 1]$.

V2. $f(x) = \begin{cases} 2^{\frac{1}{x-1}}, x < 1 \\ 0, x = 1 \\ ln(x^2 - 2x + 2), x > 1 \end{cases}$

$a = -2, b = 2,$ `[xmin,xmax]` $= [-1, 3]$.

V3. $f(x) = \begin{cases} x^2, x \in [-2,3) \\ \dfrac{1}{8}x^3 + 1, x \in [3,4) \\ 8sin(x), x \in [4,5] \\ 0, \text{în rest} \end{cases}$

$a = -2, b = 3.5,$ `[xmin,xmax]` $= [-3, 6]$.

V4. $f(x) = \begin{cases} e^x + x - 1, x \in [-2,1] \\ x^{\frac{1}{x-1}}, x \in (1,4] \\ 0, \text{în rest} \end{cases}$

$a = -2, b = 2,$ `[xmin,xmax]` $= [-3, 6]$.

V5. $f(x) = \begin{cases} e^x, x \in [0,1) \\ sinx, x \in [1,2) \\ lnx, x \in [2,3] \\ 0, \text{în rest} \end{cases}$

$a = 0.5, b = 2.5,$ `[xmin,xmax]` $= [-2, 6]$.

V6. $f(x) = \begin{cases} x^2 + 1, x \geq 1 \\ \dfrac{1}{x^2+1}, x < 1 \end{cases}$

$a = -2, b = 2,$ `[xmin,xmax]` $= [-4, 5]$.

Problema #3 Se consideră un vector cu n elemente $a = [a_1 \;\; a_2 \;\; ... \;\; a_n]$. Să se scrie o procedură care determină elementul maximal al vectorului a. Să se determine în plus numărul k de elemente maxime, precum şi poziţiile p ale acestora salvate într-un vector. Să se verifice pentru:

V1. $a = [-3 \;\; 4 \;\; 7 \;\; 4 \;\; 7 \;\; -6 \;\; 0 \;\; 3 \;\; 9 \;\; 0]$

V2. $a = [-5 \;\; -4 \;\; 8 \;\; 3 \;\; 8 \;\; -6 \;\; 0 \;\; 8 \;\; 2 \;\; 0]$

V3. $a = [3 \;\; -2 \;\; -9 \;\; 4 \;\; 5 \;\; 5 \;\; 2 \;\; -3 \;\; 4 \;\; 0]$

V4. $a = [-1 \;\; 2 \;\; 3 \;\; 4 \;\; -2 \;\; -6 \;\; 0 \;\; 6 \;\; 6 \;\; 6]$

V5. $a = [-1 \;\; 4 \;\; 2 \;\; 4 \;\; 4 \;\; -1 \;\; 0 \;\; 2 \;\; 3 \;\; 0]$

V6. $a = [-1 \;\; 2 \;\; 9 \;\; 4 \;\; -5 \;\; -6 \;\; 0 \;\; 8 \;\; 9 \;\; 0]$

Problema #4 Se consideră un vector cu n elemente $a = [a_1 \;\; a_2 \;\; ... \;\; a_{n-1} \;\; a_n]$. Să se construiască în baza elementelor vectorului a, vectorul $b = [a_n \;\; a_{n-1} \;\; ... \;\; a_2 \;\; a_1]$. Să se verifice pentru:

V1. $a = [2 \;\; 3 \;\; 4 \;\; 5 \;\; 6]$

V2. $a = [3 \;\; 4 \;\; 5 \;\; 6 \;\; 7]$

V3. $a = [4 \;\; 5 \;\; 6 \;\; 7 \;\; 8]$

V4. $a = [5 \;\; 6 \;\; 7 \;\; 8 \;\; 9]$

V5. $a = [6 \;\; 7 \;\; 8 \;\; 9 \;\; 10]$

V6. $a = [1\ \ 2\ \ 3\ \ 4\ \ 5]$

Problema #5 Se consideră seria numerică $\sum_{k=1}^{\infty} a_k$ şi valoarea exactă a sumei seriei $S = \sum_{k=1}^{\infty} a_k$.

a. Să se determine sumele parţiale S_{10}, S_{20}, S_{30}, unde $S_n = \sum_{k=1}^{n} a_k$.

b. Să se afle erorile absolute

$$e_a = |S - S_n|, n = 10, 20, 30.$$

c. Să se afle erorile relative reprezentate în procente

$$e_r = \frac{|S - S_n|}{|S|} \cdot 100\%, n = 10, 20, 30.$$

d. Să se determine numărul minim de termeni n_{min} astfel încât eroarea relativă să nu depăşească valoarea critică 3%.

V1. $\sum_{k=1}^{n} \frac{1}{2^k},\ S = 1$

V2. $\sum_{k=1}^{n} \frac{(-1)^k 2^{3k}}{9^k},\ S = \frac{-8}{17}$

V3. $\sum_{k=1}^{n} \frac{1}{(k+1)(k+2)},\ S = \frac{1}{2}$

V4. $\sum_{k=1}^{n} \frac{1}{k!},\ S = e - 1$

V5. $\sum_{k=1}^{n} \frac{k}{3^k},\ S = \frac{3}{4}$

V6. $\sum_{k=1}^{n} \frac{1}{k^2},\ S = \frac{\pi^2}{6}$