Calcul Numeric –Subiecte Examen 2019

~~I~~. Metoda poziţiei false.

~~II~~. Diferenţe finite progresive, regresive şi centrale pentru $f'(x)$.

III. Să se determine funcţia spline cubică S care interpolează datele $(1,2), (2,3), (3,5)$ şi $S'(1) = 2, S'(3) = 1$. Să se verifice că S satisface condițiile unei funcții spline cubice.

~~IV~~. Fiind dat următorul tabel conform metodei Neville prin care se aproximează $f(1.5)$ să se

x_i	$P_{m_1}(1,5)$	$P_{m_1,m_2}(1,5)$	$P_{m_1,m_2,m_3}(1,5)$
$x_1 = 1$	0		
$x_2 = 2$	1	0,5	
$x_3 = 3$			1/4

determine $f(3)$.

~~V~~. a) Să se construiască în Matlab procedura **SubsDesc** conform sintaxei x=**SubsDesc**(A, b).

b) Să se construiască în Matlab procedura **GaussPivPart** conform sintaxei $[x]$ =**GaussPivPart**(A, b).

c) Să se rezolve în Matlab sistemul

$$\begin{cases} x_2 + x_3 = 3 \\ 2x_1 + x_2 + 5x_3 = 5 \\ 4x_1 + 2x_2 + x_3 = 1 \end{cases} \tag{1}$$

apelând procedura **GaussPivPart**.

ALGORITM (Metoda substituţiei descendente)

Date de intrare: $A = (a_{ij})_{i,j=\overline{1,n}} \in \mathcal{M}_n(\mathbb{R})$; $b \in \mathbb{R}^n$; **Date de ieşire:** $x \in \mathbb{R}^n$;

$STEP1: \; x_n = \dfrac{1}{a_{nn}} b_n; \; k = n-1;$

$STEP2:$ `while` $k > 0$ `do`

$$x_k = \frac{1}{a_{kk}} \left(b_k - \sum_{j=k+1}^{n} a_{kj} x_j \right);$$

$k = k - 1;$

`endwhile`

$$P_{1,2} f(1.5) = \frac{(1.5-1)P_2(1.5) - (1.5-2)P_1(1.5)}{2-1} =$$

$$+0.5 \cdot 1 = 0.5$$

ALGORITM (Metoda Gauss cu pivotare parţială)

Date: $A = (a_{ij})_{i,j=\overline{1,n}} \in \mathcal{M}_n(\mathbb{R})$, $b \in \mathbb{R}^n$

```
STEP 1: A = (A | b) = (a_{i,j})_{i=1,n, j=1,n+1} (matricea extinsă)
STEP 2: for k = 1 : n − 1 do
            Determină primul indice p, (k ≤ p ≤ n)
            a.î. |a_pk| = max_{j=k,n} |a_jk|
            if a_pk = 0 then
                OUTPUT('Sist. incomp. sau comp. nedet.')
                STOP.
            endif
            if p ≠ k then
                L_p ↔ L_k (schimbă linia p cu linia k)
            endif
              for ℓ = k + 1 : n do
                m_ℓk = a_ℓk / a_kk;
                L_ℓ ← L_ℓ − m_ℓk L_k;
              endfor
        endfor
STEP 3: if a_nn = 0 then
            OUTPUT('Sist. incomp. sau comp. nedet.')
            STOP.
        endif
STEP 4: x = SubsDesc((a_ij)_{i,j=1,n}, (a_{i,n+1})_{i=1,n})
```

VI. Să se construiască în Matlab procedura **SplineC** având sintaxa $[y]$ =**SplineC** (X, Y, fpa, fpb, x), conform metodei de interpolaree spline cubice. Să se construiască grafic punctele de interpolare (X, Y) şi funcţia S calculată conform procedurii **SplineC**, corespunzătoare datelor de la EX.III. Pentru reprezentarea grafica a funcţiei S se va considera o discretizare a intervalului $[X_1, X_3]$ cu 20 de noduri.

ALGORITM (Interpolarea spline cubică)

Date de intrare: $X; Y; fpa; fpb; x$
Date de ieşire: y;

```
STEP 1: Determină n;
STEP 2: for j = 1 : n do
            Se determină coeficienţii a_j, b_j, c_j, d_j conform formulelor
        (2) - (4)
        endfor
STEP 3: for j = 1 : n do
            if x ∈ [X_j, X_{j+1}] do
                S = a_j + b_j(x - X_j) + c_j(x - X_j)^2 + d_j(x - X_j)^3;
                STOP
            endif
        endfor
        y = S;
```

$$a_j = Y_j, \tag{2}$$

$$\begin{cases} b_1 = S'(X_1) = fpa \\ b_{j-1} + 4b_j + b_{j+1} = \dfrac{3}{h}(Y_{j+1} - Y_{j-1}), \quad j = \overline{2,n} \\ b_{n+1} = S'(X_{n+1}) = fpb \end{cases} \tag{3}$$

$$\begin{cases} d_j = -\dfrac{2}{h^3}(Y_{j+1} - Y_j) + \dfrac{1}{h^2}(b_{j+1} + b_j), \quad j = \overline{1,n} \\ c_j = \dfrac{3}{h^2}(Y_{j+1} - Y_j) - \dfrac{b_{j+1} + 2b_j}{h}, j = \overline{1,n} \end{cases} \tag{4}$$

unde h este pasul discretizării echidistante $(X_i)_{i=\overline{1,n+1}}$.