Lucrarea 7

INTERPOLAREA

1. SCOPUL LUCRĂRII

Prezentarea unor algoritmi de interpolare*, implementarea acestora în limbaje de nivel înalt (în particular, C) şi folosirea lor în rezolvarea unor probleme din electronică.*

2. PREZENTARE TEORETICĂ

Interpolarea este una dintre metodele de aproximare a funcțiilor.

Considerăm dată o funcție *sub formă de tabel*. Cunoscând valorile funcției în anumite puncte pentru care funcția este definită, se pune problema aflării valorilor funcției în alte puncte ale domeniului de definiție.

2.1. INTERPOLAREA POLINOMIALĂ

Se consideră funcția dată prin Tabelul 7.1:

Tabelul 7.1.

x	x_0	x_1	.	x_k	.	x_n
y	y_0	y_1	.	y_k	.	y_n

Cunoaştem valoarea funcției în *n+1* puncte. Prin *n+1* puncte se poate construi un polinom de gradul *n,* unic determinat. Pentru simplificarea calculului scriem polinomul sub forma următoare:

$$P_n(x) = x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0 \tag{7.1}$$

Considerăm următoarele polinoame:

$$\begin{aligned} &\Pi_0(x) = (x - x_1)(x - x_2)\ldots(x - x_n) \\ &\Pi_1(x) = (x - x_0)(x - x_2)\ldots(x - x_n) \\ &------------------- \\ &\Pi_n(x) = (x - x_0)(x - x_1)\ldots(x - x_{n-1}) \end{aligned} \tag{7.2}$$

Formăm polinomul $P_n(x)$ sub forma:

$$P_n(x) = b_0\Pi_0(x) + b_1\Pi_1(x) + \ldots + b_k\Pi_k + \ldots + b_n\Pi_n(x) \tag{7.3}$$

Trebuie să determinăm coeficienții $b_0, b_1, \ldots, b_n$.

$$b_0=\frac{P_n(x_0)}{\Pi_0(x_0)},\quad b_1=\frac{P_n(x_1)}{\Pi_1(x_1)},\ldots,\ b_k=\frac{P_n(x_k)}{\Pi_k(x_k)},\ldots,\ b_n=\frac{P_n(x_n)}{\Pi_n(x_n)} \tag{7.4}$$

Ca urmare,

$$P_n(x)=\sum_{i=0}^{n}P_n(x_i)\frac{\Pi_i(x)}{\Pi_i(x_i)}=\sum_{i=0}^{n}y_i\frac{\prod\limits_{j=0,j\neq i}^{n}\left(x-x_j\right)}{\prod\limits_{j=0,j\neq i}^{n}\left(x_i-x_j\right)}=\sum_{i=0}^{n}y_i\prod_{j=0,j\neq i}^{n}\frac{x-x_j}{x_i-x_j} \tag{7.5}$$

Polinomul de gradul n care trece prin n+1 puncte date, numit şi *polinomul de interpolare al lui Lagrange*, are forma:

$$P_n(x)=\sum_{i=0}^{n}y_i\prod_{j=0,j\neq i}^{n}\frac{x-x_j}{x_i-x_j}\,. \tag{7.6}$$

Expresia polinomului lui Lagrange este funcţie de coordonatele punctelor cunoscute şi de variabila x . Cu ajutorul formulei determinate a polinomului, care aproximează o funcţie, se poate calcula valoarea funcţiei în orice punct necunoscut cuprins între x_0 şi x_n , presupunând că punctele x_i sunt aranjate în ordine crescătoare.

Fiind un algoritm ce face parte din clasa metodelor ***cu pas variabil*** între abscise, precizia de calcul este la fel de bună pe întreg intervalul acoperit prin perechi (x_i, y_i), indiferent unde se alege abscisa intermediară în care se calculează ordonata necunoscută. Cu alte cuvinte, această metodă nu prezintă zone de precizie preferenţiale, aşa cum se întâmplă în cazul metodelor cu pas constant între abscise.

2.1.1. Algoritmul 7.1. *Polinomul lui Lagrange*

```
real Lagrange
(
   întreg n,    // gradul polinomului de interpolare
   real x[ ],   // vectorul absciselor punctelor cunoscute
   real y[ ],   // vectorul ordonatelor punctelor cunoscute
   real x̄ ,     // punctul în care se calculează interpolarea
)
{// declararea variabilelor locale functiei
întreg  i, j ; //contori pentru buclele 'for()' şi indexare.
real sum ;     // variabilă ce reţine sumele parţiale
real prod ;    // variabilă ce reţine produsul
// corpul de instrucţiuni al funcţiei
   sum=0;
   pentru i= 0 .. n
   {
   prod=1;
   pentru j= 0 .. n   dacă ( j ≠ i )   prod = prod * (x̄ - x[j]) / (x[i] - x[j]);
   sum = sum + y[i] * prod ;
   }
   returnează sum; //      valoarea în punctul cerut
}
```

2.2. POLINOMUL DE INTERPOLARE DE SPEȚA ÎNTÂI AL LUI NEWTON

Face parte din clasa metodelor ***cu pas constant*** între abscise. Aceasta se traduce prin a avea o anumită zonă preferată, în care precizia este cea mai bună pentru un interval acoperit prin punctele $x_0, .. ,x_n$.

Acest polinom de interpolare se exprimă funcție de ***diferențele finite***. Fie $f:[a,b] \to \mathbf{R}$ şi rețeaua $x_0, x_1, x_2, ..., x_n$ cu pasul constant *h*.

Definiția 7.1. Se numeşte *diferență finită* de *ordinul întâi* expresia:

$$\Delta f(x) = f(x+h) - f(x) \tag{7.7}$$

unde *h* este pasul constant.

Diferența finită de *ordinul n* se poate defini recursiv, folosindu-se de diferența finită de ordin *n-1*, şi are expresia de definiție:

$$\Delta^n f(x) = \Delta\left(\Delta^{n-1} f(x)\right) \tag{7.8}$$

Diferențele finite au următoarele ***proprietăți***:

- Operatorul diferență finită este *liniar*:

$$\Delta(c_1 f_1 + c_2 f_2) = c_1 \Delta f_1 + c_2 \Delta f_2 \tag{7.9}$$

- Diferența finită de ordinul *n* se calculează cu formula:

$$\Delta^n f(x) = \sum_{k=0}^{n} (-1)^k C_n^k f\left(x + (n-k)h\right) \tag{7.10}$$

Diferențele finite mai pot fi obținute şi cu ajutorul tabelului 7.2.

Tabelul 7.2

x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$	$\Delta^5 y_i$	$\Delta^6 y_i$
x_0	y_0	Δy_0					
x_1	y_1	Δy_1	$\Delta^2 y_0$	$\Delta^3 y_0$			
x_2	y_2	Δy_2	$\Delta^2 y_1$	$\Delta^3 y_1$	$\Delta^4 y_0$	$\Delta^5 y_0$	
x_3	y_3	Δy_3	$\Delta^2 y_2$	$\Delta^3 y_2$	$\Delta^4 y_1$	$\Delta^5 y_1$	$\Delta^6 y_0$
x_4	y_4	Δy_4	$\Delta^2 y_3$	$\Delta^3 y_3$	$\Delta^4 y_2$		
x_5	y_5	Δy_5	$\Delta^2 y_4$				
x_6	y_6						

Definiția 7.2. Se numeşte *putere generalizată de ordinul n a lui x* expresia:

$$x^{[n]} = x(x-h)(x-2h)...(x-(n-1)h) \tag{7.11}$$

Pentru h=0 puterea generalizată coincide cu puterea obişnuită.

1. *Diferența finită a puterii generalizate* este:
$$\Delta x^{[n]} = nhx^{[n-1]} \tag{7.12}$$

2. Diferența finită de ordinul k a puterii generalizate este:
$$\Delta^k x^{[n]} = n(n-1)(n-2)\ldots(n-k+1)x^{[n-k]} \tag{7.13}$$

Fie funcția tabelată dată în *Tabelul 6.1*, unde rețeaua $x_0, x_1, x_2, x_3, \ldots, x_n$ este cu pasul constant h.

Prin cele $n+1$ puncte trece un polinom de gradul n pe care îl căutăm de forma:
$$P_n(x) = C_0 + C_1(x-x_0)^{[1]} + C_2(x-x_0)^{[2]} + \ldots + C_n(x-x_0)^{[n]} \tag{7.14}$$
unde
$$(x-x_0)^{[i]} = (x-x_0)(x-x_1)\ldots(x-x_{i-1}) \quad , \quad i = 1, 2, \ldots n$$
iar coeficienții $C_0, C_1, \ldots, C_n$ sunt necunoscute pe care le vom calcula. Se observă că:
$$P_n(x_0) = y_0 = C_0 \tag{7.15}$$
Calculăm diferența finită de ordinul întâi:
$$\Delta P_n(x) = C_1 h + 2C_2 h(x-x_0)^{[1]} + \ldots + nC_n h(x-x_0)^{[n-1]} \tag{7.16}$$
Făcând substituția $x = x_0$ rezultă $\Delta P_n(x_0) = C_1 1!h$

Se poate calcula: $$C_1 = \frac{\Delta P_n(x_0)}{1!h} \qquad \ldots\ldots\ldots(7.17)$$

Se continuă calculul diferențelor finite în punctul x_0 şi se observă că :
$$C_k = \frac{\Delta^{(k)} P_n(x_0)}{k!h^k}, \qquad \Delta^k P_n(x_0) = \Delta^k y_0, \; k = 0, 1, 2, \ldots \text{n}. \tag{7.18}$$

Ținând cont de formulele de calcul ale coeficienților, *polinomul lui Newton de interpolare de speța întâi* poate fi scris astfel:
$$P_n(x_0) = y_0 + \frac{\Delta y_0}{1!h}(x-x_0)^{[1]} + \frac{\Delta^2 y_0}{2!h^2}(x-x_0)^{[2]} + \cdots + \frac{\Delta^n y_0}{n!h^n}(x-x_0)^{[n]} \tag{7.19}$$

În calculul coeficienților s-au utilizat *diferențele finite la dreapta* perechii (x_0, y_0) (tabelul 7.2). Tot cu ajutorul acestui polinom (7.19) se poate face şi o *extrapolare*, adică tot o interpolare, dar văzută în afara intervalului acoperit prin abscisele cunoscute, $x_0, \ldots, x_n$.

Trebuie făcută o *observație* legată de precizia de calcul: pentru polinomul Newton de speța întâi, precizia cea mai bună se obține pentru abscise intermediare alese în apropierea perechii (x_0, y_0). Din acest motiv, extrapolarea amintită trebuie făcută plecând de la principiul că cu cât distanța punctului necunoscut dorit - față de abscisa x0 - este mai mare, cu atât precizia de calcul scade. Se respectă, deci, un alt principiu general în problemele de interpolare, şi anume numărul perechilor (x_i, y_i): cu cât acesta este mai mare, cu atât puterea polinomului de interpolare creşte.

2.2.1. Algoritmul 7.2. Newton 1

```
real Newton_1
(
întreg n,       // gradul polinomului de interpolare
real x0,  // abscisa primului punct cunoscut
real h,  // pasul constant între abscisele cunoscute
real y[ ],     // vectorul ordonatelor punctelor cunoscute
real xp, // abscisa punctului în care se face interpolarea
)
{ // declararea şi definirea variabilelor locale
real sum ;     // variabila ce reţine sumele parţiale
real prod ;    // variabila ce reţine produsele
integ i, j ;   //  contoare  în  buclele  'for()'  şi  indecşi
               //pentru vectori.

// corpul de instrucţiuni al funcţiei
sum = y0; // pornesc cu suma de la y0 (vezi formula //(7.19))
prod = 1 ;
 pentru i= 1… n
   {           //calculul diferenţelor finite
   pentru j= 0… n-i   y[j] = y[j+1] - y[j];
```

$$prod = prod * \left(x_p - \left(x_0 + (i-1)*h\right)\right) * \frac{1}{h} * \frac{1}{i};$$

```
   sum = sum + y0*prod ;
   }
returnează sum ;          // valoarea interpolată
}
```

2.3. POLINOMUL DE INTERPOLARE DE SPEŢA A DOUA AL LUI NEWTON

Pentru funcţia dată în *Tabelul 6.1* se caută un polinom de gradul n care trece prin cele $n+1$ puncte, sub forma:

$$P_n(x) = C_0 + C_1(x - x_n) + C_2(x - x_n)(x - x_{n-1}) + ... + C_n(x - x_n)(x - x_{n-1})...(x - x_1) \quad (7.20)$$

Polinomul de gradul n poate fi scris sub forma:

$$P_n\left(x\right) = y_n + \frac{\Delta y_n}{1!h}\left(x - x_n\right)^{[1]} + \frac{\Delta^2 y_{n-1}}{2!h^2}\left(x - x_{n-1}\right)^{[2]} + \cdots + \frac{\Delta^n y_1}{n!h^n}\left(x - x_1\right)^{[n]} \quad (7.21)$$

Acest polinom este numit *polinomul lui Newton de interpolare de speţa a doua* deoarece s-au utilizat diferenţele finite la stânga perechii (x_n, y_n). Dacă punctul de aproximare a funcţiei se găseşte în apropierea lui x_n, se recomandă utilizarea metodei se speţa a doua deoarece dă erori mai mici.

Trebuie făcută şi aici o *observaţie* legată de precizia de calcul: pentru polinomul Newton de speţa a doua, precizia cea mai bună se obţine pentru abscise intermediare alese în apropierea ultimei perechi, *(x_n, y_n)*. Din acest motiv, o eventuală extrapolare trebuie făcută, cu aceeaşi atenţie, ca şi în cazul polinomului Newton de speţa întâi, anume trebuie mai înainte să mărim suficiente de mult numărul de perechi, pentru a reuşi o extrapolare cu erori minime.

2.3.1. Algoritmul 7.3. Newton 2

```
real Newton_2
(
   întreg n,   // gradul polinomului de interpolare
   real xn ,   // abscisa maximă a punctelor cunoscute
   real h,     // pasul constant între abscisele cunoscute
   real y[ ],  // vectorul ordonatelor punctelor cunoscute
   real xp     // abscisa punctului de interpolare
)
{ // declararea şi definirea variabilelor locale
real sum ;     // variabila ce reţine sumele parţiale
real prod ;    // variabila ce reţine produsele
întreg i, j ;  // contoare

// corpul de instrucţiuni al funcţiei
 sum = yn ; // valoarea interpolată porneşte de la yn
            //(vedeţi relaţia (6.21))
prod = 1;
pentru i= 1.. n
{       // calculul diferenţelor finite
   pentru j= n.. i    y[j] = y[j] - y[j-1] ;
```

$$prod = prod * \left(x_p - \left(x_n - (i-1) * h\right)\right) * \frac{1}{h} * \frac{1}{i};$$

```
   sum = sum + yn*prod ;
}
 returnează sum;    // valoarea polinomului de interpolare
                    //în punctul cerut.
}
```

2.4. POLINOMUL LUI NEWTON DE INTERPOLARE CU DIFERENŢE DIVIZATE

Fie funcţia $f(x)$ dată sub forma celei prezentate în tabelul (7.1) unde reţeaua $x_0, x_1, x_2, ..., x_n$ din domeniul de definiţie al funcţiei nu are pas constant. Fiind o metodă cu pas variabil între abscise, precizia sa nu este preferenţială.

Definiţia 7.3. Se numeşte *diferenţă divizată de ordinul k+i* a funcţiei f expresia :

$$f(x_{i-1},x_i,x_{i+1},\ldots,x_{i+k}) = \frac{f(x_i,x_{i+1},\ldots,x_{i+k}) - f(x_{i-1},x_i,\ldots,x_{i+k-1})}{x_{i+k} - x_{i-1}} \quad (7.22)$$

Se determină polinomul de gradul n, de forma :

$$P_n(x) = C_0 + C_1(x-x_0) + C_2(x-x_0)(x-x_1) + \ldots + C_n(x-x_0)(x-x_1)\ldots(x-x_{n-1}) \quad (7.23)$$

cunoscând $n+1$ puncte (x_i, y_i), i=0, ...,n , care verifică polinomul.

Se observă că $P_n(x_0) = y_0 = C_0$.

Se calculează *diferența divizată* de ordinul întâi pentru polinomul $P_n(x)$ şi se face $x = x_1$. Rezultă:

$$P_n(x_0,x_1) = C_1.$$

Calculând în continuare, se determină diferența divizată de ordinul k şi luând pe $x = x_k$ se obține valoarea coeficientului C_k :

$$C_k = P_n(x_0,x_1,x_2,\ldots,x_k), \qquad k=0,\ 1,\ \ldots,n. \quad (7.24)$$

Polinomul se scrie acum sub forma:

$$\begin{aligned} P_n(x) = {} & y_0 + P_n(x_0,x_1)(x-x_0) + P_n(x_0,x_1,x_2)(x-x_0)(x-x_1) + \ldots \\ & + P_n(x_0,\ldots,x_n)(x-x_0)(x-x_1)\ldots(x-x_{n-1}) \end{aligned} \quad (7.25)$$

unde fiecare diferență divizată se calculează cu ajutorul formulei (7.22). Polinomul obținut (7.25) poartă numele de *polinomul lui Newton de interpolare cu diferențe divizate.*

2.4.1. Algoritmul 7.4. Newton 3

```
real Newton_3
(
   întreg n,    // numărul de puncte date ale funcției
   real x[ ],   // vectorul absciselor punctelor date
   real y[ ],   // vectorul ordonatelor punctelor date
   real x̄ ,     // punctul în care se interpolează funcția
)
{ // declararea şi definirea variabilelor locale
întreg i, j ;  // contoare
real sum ;     // variabilă ce reține sumele parțiale
real prod ;    // variabilă ce reține produsele parțiale
// corpul de instrucțiuni al funcției
   sum = y[0] ;
   prod = 1 ;
   pentru i= 1.. n
   {
      // calculul diferențelor divizate
         pentru j= 0.. n-i    y_j = (y_{j+1} - y_j) / (x_{j+i} - x_j);
         prod = prod*( x̄ - x_{i-1}) ;
         sum = sum + y[0]*prod ;
   }
         returnează  sum ;
}
```

2.5. METODA LUI AITKEN DE INTERPOLARE

Este tot o metodă *cu pas variabil* între abscise. Metoda lui Aitken de interpolare dă acelaşi rezultat ca şi metoda lui Lagrange de interpolare.

Aici se realizează **mai multe interpolări liniare**. Cu fiecare interpolare, numărul punctelor rămase se micşorează cu o unitate, iar funcția ce trece prin două puncte de la etapa curentă creşte în grad, astfel că în final, dacă sunt date $n+1$ puncte, se obține o funcție de gradul n.

Tehnica este următoarea:

- la primul pas, se scrie ecuația unei drepte folosind drept perechi de puncte pe prima (x_0, y_0), neschimbată, şi pe fiecare pereche de sub ea, succesiv (x_i, y_i), i=1,..,n ; se obțin ordonatele y_{0i} ;
- la pasul al doilea, perechile sunt (x_1, y_{01}) fixă împreună cu toate perechile de sub ea (i=2,..,n), folosindu-ne însă de ordonatele y_{0i} calculate la pasul întâi; se obțin ordonatele y_{01i} ;
- la pasul al treilea perechea fixă este $(x_2,\ y_{012})$ şi sunt folosite perechile $(x_i,\ y_{01i})$, cu i=3,…,n; se generează perechile y_{012i};

Se continuă în acelaşi mod, până ce se ajunge la o interpolare în care sunt folosite doar două perechi, $(x_{n-1},\ y_{012,\dots n-2,n-1})$ şi $(x_n,\ y_{012,\dots n-2,n})$. Se generează la acest ultim pas ordonata $y_{012,3,4,\dots,n-1,n}$, care este de fapt polinomul căutat. Gradul acesteia este n.

Considerăm funcția dată în *Tabelul 6.3*.

Tabelul 7.3

x	y					
x_0	y_0					
x_1	y_1	y_{01}				
x_2	y_2	y_{02}	y_{012}			
x_3	y_3	y_{03}	y_{013}	y_{0123}		
x_4	y_4	y_{04}	y_{014}	y_{0124}	----	
-----	----	----	----	----	----	
x_n	y_n	y_{0n}	y_{01n}	y_{012n}	----	$y_{0123\dots n}$

Pentru două puncte, ecuația unei drepte ne oferă ordonata următoare:

$$y_{0j} = \frac{x_j - \bar{x}}{x_j - x_0} y_0 + \frac{\bar{x} - x_0}{x_j - x_0} y_j \qquad j=1, 2, 3, ..., n$$

Pentru pasul al doilea, urmând exact acelaşi model, adică ecuaţia unei drepte, avem:

$$y_{01j} = \frac{x_j - \bar{x}}{x_j - x_1} I_{01} + \frac{\bar{x} - x_1}{x_j - x_1} I_{0j} \qquad j=2, 3, ..., n \qquad (7.26)$$

- -

După n-1 paşi de interpolări liniare rezultă:

$$y_{012...n} = \frac{x_n - \bar{x}}{x_n - x_{n-1}} I_{012...(n-2)(n-1)} + \frac{\bar{x} - x_{n-1}}{x_n - x_{n-1}} I_{012...(n-2)n} \qquad (7.27)$$

Valoarea interpolată a funcţiei tabelate în punctul $\bar{x}$ este $y_{012...n}$.

2.5.1. Algoritmul 7.5. Metoda lui Aitken

```
real Aitken
(
   întreg n,    // gradul polinomului de interpolare
   real x[ ],   // vectorul absciselor punctelor date
   real y[ ],   // ordonatele punctelor date
   real xp      // punctul în care se interpolează funcţia
)
{
 // declararea şi definirea variabilelor locale
întreg i, j ;  // contoare şi indecşi de vectori

// corpul de instrucţiuni al funcţiei
   pentru i= 1.. n
   pentru j= i.. n execută
```

$$y[j] = y[i-1] * \frac{xp - x[j]}{x[i-1] - x[j]} + y[j] * \frac{xp - x[i-1]}{x[j] - x[i-1]};$$

```
   returnează  y[n] ;
}
```

2.6. INTERPOLAREA CU FUNCŢII SPLINE

Cuvântul '*spline*' provine din engleză şi înseamnă o riglă elastică de care dacă se agaţă greutăţi poate fi făcută să treacă prin diferite puncte dorite, cuprinse între capetele riglei.

Considerăm dată o funcţie tabelată, reprezentată în tabelul 7.4.

Tabelul 7.4

x	x_1	x_2	x_3	----	x_n
y	y_1	y_2	y_3	----	y_n

Considerăm $x_1 = a$ şi $x_n = b$, $[a,b]$ fiind intervalul de definiţie a funcţiei f, iar abscisele $x_1, x_2, \ldots, x_n$ o diviziune Δ a intervalului de definiţie. Valorile funcţiei sunt

$$y_i = f(x_i), \quad i = 1,2,\ldots,n \tag{7.28}$$

Definiţia 7.4.

Se numeşte *funcţie spline de ordinul n* relativ la diviziunea Δ a intervalului $[a,b]$ o funcţie $S: [a,b] \to \boldsymbol{R}$ de clasă $C^{m-1}[a,b]$ ale cărei restricţii $S_i(x)$ pe fiecare interval $[x_i, x_{i+1}]$ al diviziunii sunt polinoame de ordinul m, adică:

$$S_i(x) = P_m^i(x) \quad \text{dacă} \quad x \in [x_i, x_{i+1}], \quad i = 1,2,\ldots,(n-1) \tag{7.29}$$

Funcţia $S(x)$ este netedă pe porţiuni deoarece are primele (m-1) derivate continue pe $[a,b]$, iar derivata de ordinul m este discontinuă în x_i, $i = 1,2,\ldots,n$. Gradul de netezire al funcţiei este m. Restricţiile funcţiei sunt polinoamele:

$$S_i(x) = A_i x^m + B_i x^{m-1} + C_i x^{m-2} + E_i x^{m-3} + \ldots + R_i \tag{7.30}$$

dacă, $x \in [x_i, x_{i+1}]$, $i = 1,2,\ldots,(n-1)$

Aceste funcţii sunt derivabile până la (m-1) şi sunt continue împreună cu derivatele. Derivata de ordinul (m-1) a lui $S_i(x)$ pe intervalul $[x_i, x_{i+1}]$ este o funcţie liniară şi trece prin punctele (x_i, D_i) şi (x_{i+1}, D_{i+1}) unde, $D_i = S_i^{(m-1)}(x)$ $i = 1,2,\ldots,n$.

Rezultă ecuaţia liniară:

$$S_i^{(m-1)}(x) = \frac{D_{i+1}(x - x_i) + D_i(x_{i+1} - x)}{h_i} \tag{7.31}$$

unde $h_i = x_{i+1} - x_i$, $i = 1,2,\ldots,(n-1)$

Integrând de m-1 ori relaţia (7.31) se obţine:

$$S_i^{(m-2)}(x) = \frac{D_{i+1}(x - x_i)^2 - D_i(x_{i+1} - x)^2}{2h_i} + C_{1i} \tag{7.32}$$

$$S^{(m-3)}(x) = \frac{D_{i+1}(x - x_i)^3 + D_i(x_{i+1} - x)^3}{6h_i} + C_{1i}x + C_{2i} \tag{7.33}$$

$$S_i'(x) = \frac{D_{i+1}(x - x_i)^{m-1} + (-1)^{m-2}(x_{i+1} - x)^{m-1} D_i}{(n-1)!\,h_i} + \frac{C_{1i}x^{m-3}}{(m-3)!} + \frac{C_{2i}x^{m-4}}{(m-4)!} + \ldots + C_{m-1,i} \tag{7.34}$$

$$S_i(x) = \frac{D_{i+1}(x - x_i)^m + (-1)^{m-1}(x_{i+1} - x)^{m-1} D_i}{n!\,h_i} + \frac{C_{1i}x^{m-2}}{(m-2)!} + \frac{C_{2i}x^{m-3}}{(m-3)!} + \ldots + C_{m-2,i}x + C_{m-1,i} \tag{7.35}$$

Pentru întreg intervalul $[a,b]$ rezultă un sistem liniar punând condiţia ca $S_i(x_i) = y_i$, $i = 1,2,\ldots,n$ şi continuitatea celor (m-1) ecuaţii în toate punctele x_i. În extremele x_1 şi x_2 se scrie polinomul lui Lagrange, îl derivăm până la m-1 şi aflăm corespunzător valorile derivatelor în x_1 şi x_n. Rezultă necunoscutele:

$$D_i,\ D_{i+1},\ C_{1i}, \ldots,\ C_{m-1,i}$$

pentru fiecare interval. În acest caz sunt $n+(m-1)+(n-1)$ necunoscute şi $n+(m-1)+(n-1)$ ecuaţii. Sistemul fiind liniar se rezolvă cu una din metodele din ***Lucrarea 3 - Metode pentru rezolvarea sistemelor de ecuaţii***.

În lucrare se consideră restricţiile S_i *polinoame de gradul 3*. Din acest motiv interpolarea este denumită *spline cubică*.

În acest caz:

$$S_i(x)=A_ix^3 + B_ix^2 + C_ix + E_i \tag{7.36}$$

$$S_i''(x)=\frac{D_{i+1}(x-x_i)+D_i(x_{i+1}-x)}{h_i}$$

$$S'(x)=\frac{D_{i+1}(x-x_i)^2-D_i(x_{i+1}-x)^2}{2h_i}+C_{1i}$$

$$S_i(x)=\frac{D_{i+1}(x-x_i)^3-D_i(x_{i+1}-x)^3}{6h_i}+C_{1i}x+C_{2i} \tag{7.37}$$

Din $S(x_i)=y_i$ *şi* $S(x_{i+1})=y_{i+1}$ rezultă:

$$C_{1i}=\frac{y_{i+1}-y_i}{h_i}-\frac{D_{i+1}-D_i}{6}h_i$$

$$C_{2i}=\frac{x_{i+1}y_i-x_iy_{i+1}}{h_i}-\frac{D_ix_{i+1}-D_{i+1}x_i}{6}h_i$$

Din identificarea relaţiilor (7.36) şi (7.37) rezultă :

$$A_i=\frac{D_{i+1}+D_i}{6h_i} \quad ; \quad B_i=\frac{D_ix_{i+1}-D_{i+1}x_i}{2h_i} \tag{7.38}$$

$$C_i=\frac{D_{i+1}x_i^2-D_ix_{i+1}^2}{2h_i}+\frac{y_{i+1}-y_i}{h_i}-\frac{D_{i+1}-D_i}{6}h_i$$

$$E_i=\frac{x_{i+1}^3D_i-x_i^3D_{i+1}}{6h_i}+\frac{y_ix_{i+1}-y_{i+1}x_i}{h_i}-\frac{D_ix_{i+1}-D_{i+1}x_i}{6}h_i \tag{7.39}$$

Din continuitatea primei derivate în punctul x_i, $S'_{i-1}(x_i)=S'_i(x_i)$ rezultă:

Considerând derivata de ordinul întâi în punctul x_1 şi x_n egală cu:

$$h_{i-1}D_{i-1}+2(h_i+h_{i-1})D_i+h_iD_{i+1}=6\cdot\left(\frac{y_{i+1}-y_i}{h_i}-\frac{y_i-y_{i-1}}{h_{i-1}}\right) \tag{7.40}$$

$$y_1'=\frac{y_2-y_1}{x_2-x_1}-\frac{y_3-y_2}{x_3-x_2}+\frac{y_3-y_1}{x_3-x_1} \tag{7.41}$$

$$y_n'=-\frac{y_{n-1}-y_{n-2}}{x_{n-1}-x_{n-2}}+\frac{y_n-y_{n-1}}{x_n-x_{n-1}}+\frac{y_n-y_{n-2}}{x_n-x_{n-2}} \tag{7.42}$$

respectiv rezultă sistemul tridiagonal în D_i $\quad i=1,2,3,\ldots,n$

$$\begin{cases} 2h_1D_1 + h_1D_2 = 6\left(\dfrac{y_2 - y_1}{h_1} - y_1'\right) \\ h_1D_1 + 2(h_1 + h_2)D_2 + h_2D_3 = 6\left(\dfrac{y_4 - y_3}{h_3} - \dfrac{y_3 - y_2}{h_3}\right) \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ h_{i-1}D_{i-1} + 2(h_{i-1} + h_i)D_i + h_iD_{i+1} = 6\left(\dfrac{y_{i+1} - y_i}{h_i} - \dfrac{y_i - y_{i+1}}{h_i}\right) \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ h_{n-2}D_{n-2} + 2(h_{n-2} + h_{n-1})D_{n-1} + h_{n-1}D_n = 6\left(\dfrac{y_n - y_{n-1}}{h_{n-1}} - \dfrac{y_{n-1} - y_{n-2}}{h_{n-1}}\right) \\ h_{n-1}D_{n-1} + 2h_{n-1}D_n = 6\left(y_n' - \dfrac{y_n - y_{n-1}}{h_i}\right) \end{cases} \tag{7.43}$$

unde y_1' şi y_n' sunt date de expresiile (7.41), respectiv (7.42) .

Din sistemul (7.43) rezultă valorile lui D_i , i =1,2,...,n. Din (7.38) rezultă coeficienţii restricţiilor pe fiecare interval, restricţii care aproximează funcţia dată. Dacă se dă $\bar{x}$ în care trebuie calculată funcţia, se stabileşte intervalul în care se găseşte $\bar{x}$ şi se calculează valoarea restricţiei funcţiei pe acest interval în punctul $\bar{x}$.

2.6.1. Algoritmul 7.6. Funcţii ‘spline’

```
real Spline
(
   întreg n,    // gradul polinomului de interpolare
   real x[ ],   // abscisele punctelor date
   real y[ ],   // ordonatele punctelor date
   real xp,     // punctul în care se interpolează funcţia
)
{ // declararea şi definirea variabilelor locale
// vectorii elementelor diagonale ale sistemului tridiagonal
real A[N-1], B[N], C[N-1] ;
// vectorul termenilor liberi al sistemului tridiagonal;
real TL[N];
// vectorul paşilor punctelor pe abscisă;
real h[N];
// vectorul soluţiilor sistemului tridiagonal;
real D[N];
real derx1;     // derivata în punctul  x1, real;
real derxn;     // derivata în punctul  xn, real;
întreg num;     //  variabila ce dă numărul intervalului în
                //care se gaseste xp
real valint;    //   variabilă   reală   ce   reţine   valoarea
                //polinomului de gradul trei în punctul xp .
întreg i, j;    // contoare şi indecşi de vectori.

// corpul de instrucţiuni al funcţiei
   pentru  i= 1.. n-1     h[i] = x[i+1] − x[i];
```

$$derx1 = \frac{y_2 - y_1}{x_2 - x_1} - \frac{y_3 - y_2}{x_3 - x_2} + \frac{y_3 - y_1}{x_3 - x_1} \;;$$

$$derxn = -\frac{y_{n-1} - y_{n-2}}{x_{n-1} - x_{n-2}} + \frac{y_n - y_{n-1}}{x_n - x_{n-1}} + \frac{y_n - y_{n-2}}{x_n - x_{n-2}} \;;$$

```
// construieşte sistemul tridiagonal
```

$$A_1 = 0 \;;\; \mathrm{B}_1 = 2h_1 \;;\; \mathrm{C}_1 = h_1 \;;\; \mathrm{TL}_1 = 6 \cdot \left(\frac{y_2 - y_1}{h_1} - y_1' \right)$$

$$\begin{cases} A_i = h_{i-1} \;;\; \mathrm{B_i} = 2 \cdot \left(h_{i-1} - h_i \right) \;;\; \mathrm{C_i} = h_i \;; \\ \mathrm{TL_i} = 6 \cdot \left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}} \right) \;;\; \mathrm{i} = 2,3\;,\ldots \mathrm{n}-1\;; \end{cases}$$

$$A_n = h_{n-1} \;;\; \mathrm{B_n} = 2 \cdot h_{n-1} \;;\; \mathrm{C_n} = 0 \;;\; \mathrm{TL_n} = 6 \cdot \left(y_n' - \frac{y_n - y_{n-1}}{h_{n-1}} \right)$$

```
// Rezolvă sistemul conform algoritmului (3.5) din Lucrarea 3.
   Tridiagonal(A,B,C,TL,D) ; // această este funcţia din L3.
// calculează coeficienţii polinomului de interpolare
   pentru i= 1.. n
   {
```

$$A_i = \frac{D_{i+1} + D_i}{6h_i} \;;$$

$$B_i = \frac{D_i x_{i+1} - D_{i+1} x_i}{2h_i} \;;$$

$$C_i = \frac{D_{i+1} x_i^2 - D_i x_{i+1}^2}{2h_i} + \frac{y_{i+1} - y_i}{h_i} - \frac{D_{i+1} - D_i}{6} h_i \;;$$

$$E_i = \frac{x_{i+1}^3 D_i - x_i^3 D_{i+1}}{6h_i} + \frac{y_i x_{i+1} - y_{i+1} x_i}{h_i} - \frac{D_i x_{i+1} - D_{i+1} x_i}{6} h_i \;;$$

```
   }
// Se caută intervalul care conţine punctul de interpolare
   i =1 ;
   execută
   {
   i = i + 1 ;!!!aici e ceva in neregula!!!
   }
   cât timp ( xp>=x[i]) ;
```

$$val\,\mathrm{int} = A_i.xp^3 + B_i.xp^2 + C_i.xp + E_i \;;$$

```
   returnează  valint ;
}
```

2.7. INTERPOLAREA FUNCȚIILOR PERIODICE

Se consideră funcția periodică $f:[a,b] \to \mathbf{R}$ cu proprietatea că $f(a) = f(b)$ unde $T = b - a$ este perioada funcției.

Experimental s-au obținut n valori ale funcției pe perioada $[a,b]$ prezentate în tabelul 7.5 şi se cere o valoare a funcției f într-un punct $x_p \in [a,b]$, dar $x_p \neq x_i$, $i = 1,2,...,n$.

Tabelul 7.5

x	$x_1 = a$	x_2	x_3	----	$x_{2n} = b$
y	y_1	y_2	y_3	----	y_{3n}

Pentru determinarea acestei valori se determină un polinom de interpolare a funcției periodice.

2.7.1. Algoritmul 7.7. Funcții periodice

```
real Inte_per
(
   întreg n,   // numărul de valori cunoscute ale funcției
   real x[ ],  // vectorul absciselor punctelor cunoscute
   real y[ ],  // vectorul ordonatelor punctelor cunoscute
   real T,     // perioada
   real x̄      // punctul de interpolare a funcției
)
{ // declararea şi definirea variabilelor locale
întreg i, j ;  // contoare
real sum ;     // variabilă pentru a reține sumele
real prod ;    // variabilă pentru a reține produsele

// corpul de instrucțiuni al funcției
sum = 0 ;
pentru  i= 0.. n
{
   prod = 1;
   pentru  j= 0.. n
```

$$\text{dacă} \quad (j \neq i) \quad prod = prod * \frac{\sin\left[\dfrac{\pi * (\bar{x} - x[j])}{T}\right]}{\sin\left[\dfrac{\pi * (x[i] - x[j])}{T}\right]} \; ;$$

```
   sum = sum + y[i]*prod;
}

returnează  sum ;
}
```

2.8. INTERPOLAREA FUNCȚIILOR DE MAI MULTE VARIABILE

Vom considera o funcție de două variabile: $f : \mathbf{E} \to \mathbf{R}$ unde $\mathbf{E} \subset \mathbf{R}^2$ de forma $f = g(x, y)$. Se consideră următoarele puncte cunoscute $(x_i, y_i, f(x_i, y_i))$ pentru $i = 0,1,2,\ldots,n$ și $j = 0,1,\ldots,n$.

Se pune problema determinării unui polinom de gradul n care să satisfacă condițiile $P_n(x_i, y_i) = f(x_i, y_i)$ cu $i, j = 0,1,2,\ldots,n$.

Polinomul lui Lagrange pentru două variabile are forma:

$$L(x,y) = \sum_{i=0}^{n} \sum_{j=0}^{m} f(x_i, y_j) \prod_{\substack{k=0 \\ k \neq i}}^{n} \frac{x - x_k}{x_i - x_k} \prod_{\substack{k=0 \\ k \neq j}}^{m} \frac{y - y_k}{y_j - y_k} \tag{7.47}$$

2.8.1. Algoritmul 7.8. Funcții de două variabile

```
real Lagrange_2
(
   întreg n,   // numărul de puncte pe Ox
   întreg m,   // numărul de puncte pe Oy
   real x[ ],  // vectorul coordonatelor punctelor pe Ox
   real y[ ],  // coordonatele punctelor pe Oy
   real z[ ][N], // matricea valorilor funcției
   real pcx,      //  coordonata  pe  Ox  a  punctului  de
                  //interpolare, reală
   real pcy       //  coordonata  pe  Oy  a  punctului  de
                  //interpolare, reală
)

{ // declararea și definirea variabilelor locale
întreg i, j, k ; // contoare
real prodx, prody ;  // produse parțiale
real sum ;           // suma parțială

   // corpul de instrucțiuni al funcției
   sum = 0 ;
   pentru i= 0 .. n
   pentru j= 0 .. m
   {
   prodx = 1 ;
   prody = 1 ;
   pentru k= 0 to n
    dacă  ( k ≠ i)   prodx = ( pcx - x[k])/( x[i] - x[k]) ;
   pentru k= 0 to m
    dacă  ( k ≠ j)   prody = ( pcy - y[k])/( y[j] - y[k]) ;
   sum = z[i][j]*prodx*prody ;
   }
returnează sum;
}
```

3. DESFĂŞURAREA LUCRĂRII

3.1. ÎNTREBĂRI

1. Având 7 valori `(xi, yi)`, i=0..6, ce aparțin unei funcții de tip exponențial, precizați care din metodele:
   a) Lagrange si Newton1
   b) Newton1 si Newton2
   c) Newton1 si Newton3

   vor determina valoarea funcției in punctul **ics** din intervalul [x_0, x_6] cu o precizie mai bună? Explicați.

2. Să se găsească polinoamele de interpolare *Newton 1* şi *Newton 3* pentru următoarele valori:

X	0.1	0.15	0.2	0.25	0.3
y	0.11	0.1725	0.24	0.3125	0.39

3.2. PROBLEME LABORATOR

1. Să se implementeze algoritmii 7.1-7.8 în limbajul ANSI C.

2. Se consideră datele experimentale obținute pentru caracteristica rezistenței termice joncțiune termică $R_{jt}[^\circ C/W]$ funcție de lungimea terminalului l [$inch$], pentru o diodă Zener prezentate în tabelul următor:

Tabelul 6.6

l [inch]	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Rjt [°C/w]	70	140	175	200	225	250	265	280	290	300

Se cere să se determine puncte între valorile date pentru o reprezentare grafică cât mai precisă.

3. Se consideră datele experimentale:

x	0	3.14	4.71	6.28
y	0	0	-1	0

Să se calculeze `y(1.57)` cu ajutorul metodei de interpolare a funcțiilor periodice.

3.2. TEME DE CASĂ

1. Într-unul din terminalele unei diode de mică putere se injectează succesiv o serie de curenți *i*, rezultând respectiv valorile tensiunii *v*, conform tabelului:

v (V)	.2	.28	.36	.44	.52	.6
i (A)	10^{-9}	10^{-8}	10^{-7}	10^{-6}	10^{-5}	10^{-4}

Determinați curentul corespunzător următoarelor tensiuni *v*:

```
.253 V            // stânga intervalului
.395 V            // ~ mijlocul intervalului
.559 V            // dreapta intervalului.
```

Se aplică două dintre metodele de interpolare prezentate la laborator:

- cu pas constant între abscise;
- cu pas variabil între abscise.

Metode cu pas constant: Newton, variantele 1 şi 2;
Metode cu pas variabil: Lagrange, Newton-varianta 3 şi Aitken.

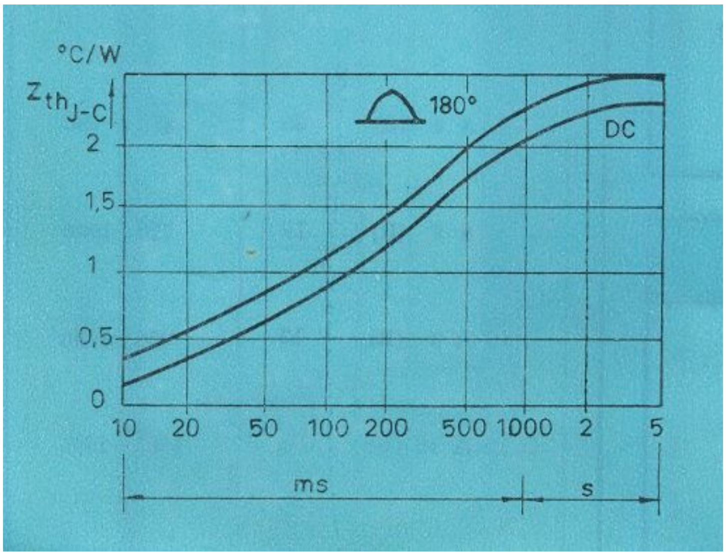
Indicație: În situația în care rezultatele metodelor de interpolare nu sunt concludente (poate chiar eronate, datorită ordinului de mărime), încercați să logaritmați valorile curentului.

2. În **figura de mai jos** se prezintă impedanța termică tranzitorie joncțiune-capsulă $Z_{thJ\text{-}C}$ pentru o *diodă cu avalanşă controlată*. Realizați minim 7 măsurări cât mai precise de pe această caracteristică.

Exemplu: $Z_{thJ\text{-}C}$ se măsoară pentru:

```
t=10, 20, 50, 200, 500, 1000, 2000 ms
```

şi apoi se verifică valoarea obținută prin interpolare (să zicem pentru `1500ms`) cu valoarea măsurată corespunzător pe caracteristică.



Caracteristica impedanței termice tranzitorii joncțiune-capsulă pentru dioda cu avalanşă controlată.

BIBLIOGRAFIE

1. I. Rusu, "*Metode numerice în electronică. Aplicații în limbaj C*", Editura Tehnică, Bucureşti, 1997
2. E.A. Volkov, „*Numerical Methods*", Mir Publisher Moscow, 1986.
3. George-Daniel Mateescu, Ileana-Carmen Mateescu, „*Analiză Numerică*", manual pentru cls. XII, profil Informatică, Editura Petrion, Bucureşti, 1995.
4. B. Demidovici, I.A. Maron, „*Computational Mathematics*", Mir Publisher Moscow, 1981.