UTILIZAREA METODELOR NUMERICE ÎN SINTEZA MECANISMELOR

Studenți: COVALIU Leon-Dumitru, PANAIT Iulian-Mihail
Facultatea: IMST, Specializarea: Master MSSMM, Anul de studii: anul I, e-mail: leon.covaliu@yahoo.com
Conducători științifici: As.dr.ing. **Luciana DUDICI,** As.dr.ing. **Alexandra ROTARU**

REZUMAT: Dimensiunile liniare şi unghiulare ale elementelor mecanismelor se pot determina prin diverse metode numerice, cum ar fi: derivarea numerică, rezolvarea sistemelor de ecuaţii neliniare, rezolvarea sistemelor de ecuaţii liniare, inversarea matricelor, înmulţirea matricelor, transpusa unei matrici şi altele. În prezenta lucrare sunt prezentate pe scurt câteva din aceste metode şi se propune determinarea termenilor matricei funcţionale prin utilizarea unor formule. Ca exemplu este realizată sinteza mecanismului de săpat gropi pentru arbuşti.

CUVINTE CHEIE: sinteza mecanismelor, metode numerice, sistem de ecuaţii neliniare

1. Introducere

Pentru determinarea dimensiunilor liniare şi unghiulare ale elementelor mecanismelor se folosesc diverse metode numerice, cum ar fi: derivarea numerică, rezolvarea sistemelor de ecuaţii neliniare, rezolvarea sistemelor de ecuaţii liniare, inversarea matricelor, înmulţirea matricelor, transpusa unei matrice şi altele. În continuare se vor prezenta câteva din metodele numerice amintite.

2. Metode numerice folosite în sinteza mecanismelor

2.1. Rezolvarea sistemelor de ecuaţii neliniare

Pentru rezolvarea sistemelor de ecuaţii neliniare se pot folosi diverse metode numerice, cum ar fi: metoda Newton-Raphson, metoda gradientului, metoda Kani, etc. Fiecare din aceste metode cere matricea funcţională a sistemului. Determinarea pe cale analitică a termenilor matricei funcţionale este cu atât mai greoaie cu cât numărul ecuaţiilor sistemului devine din ce în ce mai mare. De exemplu, la sinteza mecanismului patrulater care aproximează o curbă dată, se pot impune maximum nouă puncte, ceea ce conduce la un sistem de 36 de ecuaţii neliniare cu 36 de necunoscute. În acest caz matricea funcţională are 1296 de elemente. Dacă se face sinteza unui mecanism Watt sau Stephenson pentru aproximarea unei curbe date, mecanisme ce au în componenţă numai cuple de rotaţie, apare un sistem de 90 de ecuaţii neliniare cu 90 de necunoscute, iar matricea funcţională are 8100 de termeni. Se observă din aceste două exemple cât de dificilă este obţinerea matricei funcţionale.

În lucrare se propune determinarea termenilor matricei funcţionale prin derivare numerică, folosind diverse formule.

2.1.1. Metoda Newton-Raphson pentru rezolvarea sistemelor de ecuaţii neliniare.

Fie sistemul de ecuaţii neliniare

$$\begin{cases} f_1(x_1, x_2, ..., x_n) = 0; \\ f_2(x_1, x_2, ..., x_n) = 0; \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ f_n(x_1, x_2, ..., x_n) = 0. \end{cases} \tag{1}$$

Vectorial sistemul poate fi scris sub forma: $f(x) = 0$, $f: \mathbf{X} \to \mathbf{Y}$, $\mathbf{X}, \mathbf{Y} \subset \mathbf{R}^n$.

Pe o vecinătate $\mathbf{V} \subset \mathbf{X}$ există o soluţie unică $x = \alpha$, adică $f(\alpha) = 0$.

Matricea funcţională a sistemului este:

$$W(x)=\begin{Vmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \ldots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \ldots & \frac{\partial f_2}{\partial x_n} \\ \ldots & \ldots & \ldots & \ldots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \ldots & \frac{\partial f_n}{\partial x_n} \end{Vmatrix}. \tag{2}$$

Dacă funcţiile $f_1, f_2, \ldots, f_n$, împreună cu derivatele lor parţiale $\frac{\partial f_i}{\partial x_j}$, $(i,j)=\overline{1,n}$ sunt continue pe vecinătatea **V** şi $\mathbf{det}\,W(x)\big|_{x=\alpha}\neq 0$, atunci matricea $\mathbf{W}(x)$ admite o inversă $\mathbf{W}^{-1}(x)$, iar soluţia sistemului este de forma, în cazul metodei Newton-Raphson este de forma:

$$x^{(k+1)}=x^{(k)}-\mathbf{W}^{-1}(x^{(k)})\cdot f(x^{(k)}), \tag{3}$$

unde $x^{(k+1)}$ este soluţia sistemului (1) la iteraţia (k+1), iar $\mathbf{W}^{-1}(x^{(k)})\cdot f(x^{(k)})$ este eroarea soluţiei de la iteraţia k la iteraţia k+1.

Procesul de calcul este iterativ şi se opreşte atunci când este realizată condiţia:

$$\left|x^{(k+1)}-x^{(k)}\right|\leq\varepsilon\ . \tag{4}$$

2.1.2. Metoda gradientului.

Dacă pentru rezolvarea sistemului de ecuaţii (1) se foloseşte metoda gradientului, soluţia este de forma:

$$x^{(k+1)}=x^{(k)}-\mu_k\mathbf{W}'(x^{(k)})\cdot f(x^{(k)}), \tag{5}$$

unde:

$$\mu_k=\frac{\left[f(x^{(k)}),\mathbf{W}(x^{(k)})\cdot\mathbf{W}'(x^{(k)})\cdot f(x^{(k)})\right]}{\left[\mathbf{W}(x^{(k)})\cdot\mathbf{W}'(x^{(k)})\cdot f(x^{(k)}),\mathbf{W}(x^{(k)})\cdot\mathbf{W}'(x^{(k)})\cdot f(x^{(k)})\right]}. \tag{6}$$

În ambele metode de rezolvare, a sistemului de ecuaţii neliniare, este necesară matricea funcţională a sistemului. Propunem determinarea termenilor matricei funcţionale a sistemului pe cale numerică, adică prin derivarea numerică a funcţiilor.

3. Derivarea numerică a funcţiilor date sub formă numerică şi sub formă analitică.

Dacă se consideră o funcţie $y=y(x)$, definită numeric prin coordonatele (x_i,y_i), $i=\overline{1,n}$, unde $x_i=x_{i-1}+h$ (h fiind pasul reţelei), pentru calculul derivatei $y'=\frac{dy}{dx}$ în punctul de abscisă x_i se pot folosi diverse formule de derivare numerică, după cum urmează:

$y_{\rm i}=\frac{y_{\rm i+1}-y_{\rm i-1}}{2h}$, $i=\overline{2,n-1}$, dacă se iau în considerare 3 puncte de pe curbă;

$y_{\rm i}=\frac{y_{\rm i-2}-8(y_{\rm i-1}-y_{\rm i+1})-y_{\rm i+2}}{12h}$, $i=\overline{3,n-2}$, dacă se iau în considerare 5 puncte de pe curbă;

$y_{\rm i}=\frac{58(y_{\rm i+1}-y_{\rm i-1})+67(y_{\rm i+2}-y_{\rm i-2})-22(y_{\rm i+3}-y_{\rm i-3})}{252h}$, $i=\overline{4,n-3}$, dacă se iau în considerare 7 puncte de pe curbă ş.a.m.d.

Pentru derivarea numerică a funcţiilor analitice corespunzătoare unui sistem de n ecuaţii neliniare în necunoscutele $x_1, x_2, ..., x_n$, se consideră un pas de derivare h şi se aplică una din relaţiile de derivare numerică. Astfel, dacă se consideră 3 puncte de pe curbă se obţin relaţiile:

$$\frac{\partial f_i}{\partial x_j} \cong \frac{f_i(x_1, x_2, \ldots, x_{j-1}, x_j + h, x_{j+1}, \ldots, x_n) - f_i(x_1, x_2, \ldots, x_{j-1}, x_j - h, x_{j+1}, \ldots, x_n)}{2h}, \ i = \overline{1, n}, \quad j = \overline{1, n}; \quad (7)$$

În mod similar se obţin şi celelalte formule de derivare.

4. Înmulţirea matricelor

Două matrice **A** şi **B** nu întotdeauna se pot înmulţi. Pentru a putea realiza înmulţirea a două matrice, trebuie ca numărul de coloane ale primei matrice să fie egal cu numărul de linii ale celei de a doua matrice.

Fie **A**, o matrice cu l linii şi m coloane şi **B**, o matrice cu m linii şi n coloane. Cele două matrice se mai pot scrie şi sub forma: $\mathbf{A}[l, m]$, $\mathbf{B}[m, n]$. Produsul celor două matrice este o altă matrice cu dimensiunile l şi n, adică $\mathbf{C}[l, n]$.

Elementele matricei **C** sunt de forma:

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \ldots + a_{in}b_{nj} = \sum_{k=1}^{m} a_{ik}b_{kj}, \ i = \overline{1, l}, \quad j = \overline{1, n}. \quad (8)$$

5. Sinteza mecanismului de săpat gropi pentru arbuşti

În continuare, se prezintă sinteza unui mecanism care aproximează o dreaptă

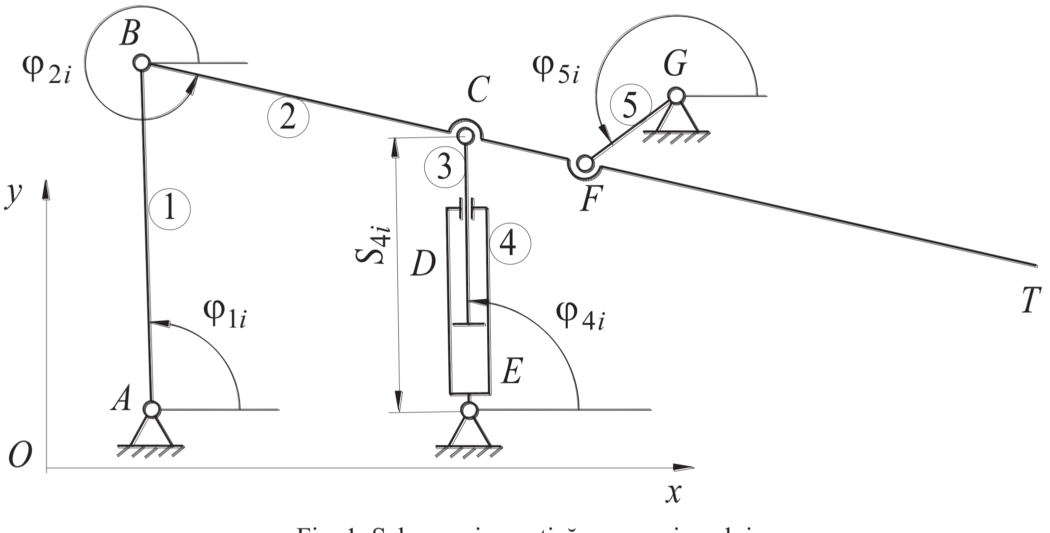

Fig. 1. Schema cinematică a mecanismului

Pentru scrierea ecuaţiilor necesare sintezei mecanismului, se foloseşte metoda contururilor [16]. Astfel, pe contururile independente OABCEO, OABFGO şi OABTO se scriu ecuaţiile vectoriale:

$$\begin{aligned} &\overline{OA} + \overline{AB} + \overline{BC} = \overline{OE} + \overline{EC}; \\ &\overline{OA} + \overline{AB} + \overline{BF} = \overline{OG} + \overline{GF}; \quad (9) \\ &\overline{OA} + \overline{AB} + \overline{BT} = \overline{OT}. \end{aligned}$$

Proiectând ecuaţiile (9) pe axele de coordonate ale sistemului xOy se obţin ecuaţiile scalare:

$$\begin{cases} XA + AB\cos\varphi_{1i} + BC\cos\varphi_{2i} - S_{4i}\cos\varphi_{4i} - XE = 0; \\ YA + AB\sin\varphi_{1i} + BC\cos\varphi_{2i} - S_{4i}\sin\varphi_{4i} - YE = 0; \\ XA + AB\cos\varphi_{1i} + BF\cos\varphi_{2i} - FG\cos\varphi_{5i} - XG = 0; \\ YA + AB\sin\varphi_{1i} + BF\sin\varphi_{2i} - FG\sin\varphi_{5i} - YG = 0; \\ XA + AB\cos\varphi_{1i} + BT\cos\varphi_{2i} - XT_i = 0; \\ YA + AB\sin\varphi_{1i} + BT\sin\varphi_{2i} - YT_i = 0; \\ i = \overline{1,n} \ . \end{cases} \tag{10}$$

Dacă se consideră originea sistemului de axe xOy în punctul *A*, iar punctul *E* se află pe axa *Ox* a sistemului de coordonate, atunci necunoscutele sistemului de ecuaţii neliniare (10) sunt: *XE*, *XG*, *YG*, *AB*, *BC*, *BF*, *FG*, *BT*, φ_{1i}, φ_{2i}, φ_{4i}, φ_{5i}, S_{4i} $i = \overline{1,n}$. Se observă că în sistemul de ecuaţii (10) sunt 8 mărimi constante şi $5n$ mărimi variabile. Din condiţia de compatibilitate $8+5n=6n$, rezultă $n=8$, adică se pot impune 8 poziţii pentru punctul trasar T. Aşadar, se obţine un sistem de 48 de ecuaţii neliniare cu 48 de necunoscute. Numărul ecuaţiilor sistemului fiind foarte mare, determinarea expresiilor elementelor matricei funcţionale este deosebit de dificilă, ceea ce impune folosirea unei proceduri de derivare numerică.

În tabelul 1 se prezintă o procedură de rezolvare a sistemelor de ecuaţii neliniare prin metoda gradientului, folosind metoda derivării numerice a funcţiilor pentru obţinerea elementelor matricei funcţionale.

Tabelul 1

```
PROCEDURE GRADDERN(NUME:PROC6;MF:PROC7;N,ITER:INTEGER;
   H,EPS:REAL;VAR X:VECTOR1;VAR KOD:INTEGER);
 VAR I,J,N1,N2:INTEGER;AM,F11,F12,JESF:REAL;
  DX,F,F1,F2:VECTOR1;W,W1:VECTOR;
 BEGIN
  FOR I:=1 TO N DO
   BEGIN
    F1[I]:=0;
    F2[I]:=0;
   END;
  FOR I:=1 TO N*N DO
   BEGIN
    W[I]:=0;
    W1[I]:=0;
   END;
  N1:=N+1;
  I:=1;
  REPEAT
   MF(NUME,X,N,H,F,W,KOD);
   TRANSPUS(W,N,W1);
   PRODMAT(W1,F,N,N,1,F1);
   PRODMAT(W,F1,N,N,1,F2);
   F11:=0;
   F12:=0;
   FOR J:=1 TO N DO
    BEGIN
     F11:=F11+F[J]*F2[J];
     F12:=F12+F2[J]*F2[J];
    END;
    IF ABS(F12) < 1.E-15 THEN
     BEGIN
```

```
          KOD:=1;
          WRITELN(' KOD= ',KOD);
          READLN;
          EXIT;
         END
        ELSE
         AM:=F11/F12;
         FOR J:=1 TO N DO
          DX[J]:=AM*F1[J];
         JESF:=0;
         J:=1;
         REPEAT
          IF ABS(DX[J]) < EPS THEN
            J:=J+1
           ELSE
            BEGIN
             J:=N1;
             JESF:=1;
            END;
         UNTIL J > N;
         IF JESF = 0 THEN EXIT;
          FOR J:=1 TO N DO
          X[J]:=X[J]-DX[J];
          WRITELN(I:4,'  ',X[1]);
          I:=I+1;
     UNTIL I > ITER;
      WRITELN('SIST. NU CONVERGE');
      READLN;
      HALT;
     END;
```

În tabelul 2 se prezintă programul principal de calcul, pentru sinteza mecanismului menţionat.

Tabelul 2

```
 PROGRAM SINTEZA_MECANISM;
  USES CRT;
   {$M 65520,0,655360}
      TYPE
        VECTOR1=ARRAY[1..48]OF REAL;
        VECTOR=ARRAY[1..2304]OF REAL;
      PROC6=PROCEDURE(N:INTEGER;X:VECTOR1;VAR F:VECTOR1;VAR KOD:INTEGER);
      PROC7=PROCEDURE(PROC:PROC6;X:VECTOR1;N:INTEGER;EPS:REAL;VAR
             F:VECTOR1;VAR W:VECTOR;VAR KOD:INTEGER);
   VAR X,XT,YT:VECTOR1;
       H,EPS:REAL;
       I,N,ITER,KOD:INTEGER;
       F1:TEXT;
{F+}
 PROCEDURE FUNCT(N:INTEGER;X:VECTOR1;VAR F:VECTOR1;
             VAR KOD:INTEGER);
   VAR C1,S1,C2,S2,C4,S4,C5,S5:REAL;
       XT,YT:VECTOR1;
       I:INTEGER;
    BEGIN
     FOR I:=1 TO 8 DO
      BEGIN
       XT[I]:=2.352;
       YT[I]:=0.223+(I-1)*0.14;
      END;
```

```
 FOR I:=1 TO 8 DO
  BEGIN
   C1:=COS(X[I+8]);C2:=COS(X[I+16]);C4:=COS(X[I+24]);C5:=COS(X[I+32]);
   S1:=SIN(X[I+8]);S2:=SIN(X[I+16]);S4:=SIN(X[I+24]);S5:=SIN(X[I+32]);
   F[I]:=X[4]*C1+X[5]*C2-X[40+I]*C4-X[1];
   F[I+8]:=X[4]*S1+X[5]*S2-X[40+I]*S4;
   F[I+16]:=X[4]*C1+X[6]*C2-X[7]*C5-X[2];
   F[I+24]:=X[4]*S1+X[6]*S2-X[7]*S5-X[3];
   F[I+32]:=X[4]*C1+X[8]*C2-XT[I];
   F[I+40]:=X[4]*S1+X[8]*S2-YT[I];
  END;
 KOD:=0;
END;
{F-}
  PROCEDURE PRODMAT(A:VECTOR;B:VECTOR1;L,M,N:INTEGER;
                VAR R:VECTOR1);
   VAR I,J,K,IJ,IK,KJ:INTEGER;
    BEGIN
     FOR I:=1 TO L DO
      BEGIN
       FOR J:=1 TO N DO
        BEGIN
         IJ:=I+(J-1)*L;
         R[IJ]:=0.;
         FOR K:=1 TO M DO
          BEGIN
           IK:=I+(K-1)*L;
           KJ:=K+(J-1)*M;
           R[IJ]:=R[IJ]+A[IK]*B[KJ];
          END;
        END;
      END;
    END;
 PROCEDURE TRANSPUS(A:VECTOR;N:INTEGER;VAR B:VECTOR);
   VAR I,J:INTEGER;
    BEGIN
     FOR J:=1 TO N DO
      BEGIN
       FOR I:=1 TO N DO
       B[I+(J-1)*N]:=A[J+(I-1)*N];
      END;
    END;
PROCEDURE MF(PROC:PROC6;X:VECTOR1;N:INTEGER;H:REAL;
                VAR F:VECTOR1;VAR W:VECTOR;VAR KOD:INTEGER);
  VAR I,K:INTEGER;
      F1,F2,X1:VECTOR1;
   BEGIN
    FOR I:=1 TO N*N DO
     W[I]:=0;
    FOR K:=1 TO N DO
     BEGIN
      FOR I:=1 TO N DO
       X1[I]:=X[I];
      X1[K]:=X[K]-H;
      PROC(N,X1,F1,KOD);
      X1[K]:=X[K]+H;
      PROC(N,X1,F2,KOD);
```

```
   FOR I:=1 TO N DO
    W[I+(K-1)*N]:=(F2[I]-F1[I])/(2*H);
  END;
   PROC(N,X,F,KOD);
 END;
PROCEDURE GRADDERN(NUME:PROC6;MF:PROC7;N,ITER:INTEGER;
      H,EPS:REAL;VAR X:VECTOR1;VAR KOD:INTEGER);
 VAR I,J,N1,N2:INTEGER;AM,F11,F12,JESF:REAL;
    DX,F,F1,F2:VECTOR1;W,W1:VECTOR;
 BEGIN
  FOR I:=1 TO N DO
   BEGIN
    F1[I]:=0;
    F2[I]:=0;
   END;
  FOR I:=1 TO N*N DO
   BEGIN
    W[I]:=0;
    W1[I]:=0;
   END;
  N1:=N+1;
  I:=1;
  REPEAT
   MF(NUME,X,N,H,F,W,KOD);
   TRANSPUS(W,N,W1);
   PRODMAT(W1,F,N,N,1,F1);
   PRODMAT(W,F1,N,N,1,F2);
   F11:=0;
   F12:=0;
   FOR J:=1 TO N DO
    BEGIN
     F11:=F11+F[J]*F2[J];
     F12:=F12+F2[J]*F2[J];
    END;
     IF ABS(F12) < 1.E-15 THEN
      BEGIN
       KOD:=1;
       WRITELN(' KOD= ',KOD);
       READLN;
       EXIT;
      END
     ELSE
      AM:=F11/F12;
      FOR J:=1 TO N DO
       DX[J]:=AM*F1[J];
      JESF:=0;
      J:=1;
      REPEAT
       IF ABS(DX[J]) < EPS THEN
         J:=J+1
        ELSE
         BEGIN
          J:=N1;
          JESF:=1;
         END;
      UNTIL J > N;
      IF JESF = 0 THEN EXIT;
```

```
          FOR J:=1 TO N DO
          X[J]:=X[J]-DX[J];
          WRITELN(I:4,'  ',X[1]);
          I:=I+1;
      UNTIL I > ITER;
       WRITELN('SIST. NU CONVERGE');
       READLN;
       HALT;
      END;
BEGIN
 CLRSCR;
 ASSIGN(F1,'FUNCT.DAT');
 REWRITE(F1);
  KOD:=0;
 X[1]:=0.6; X[2]:=0.75; X[3]:=0.7; X[4]:=0.83; X[5]:=0.58;
 X[6]:=0.66; X[7]:=0.27; X[8]:=2.35;
 X[9]:=1.58; X[10]:=1.62; X[11]:=1.64; X[12]:=1.66; X[13]:=1.67;
 X[14]:=1.67; X[15]:=1.67; X[16]:=1.66;
 X[17]:=-0.25; X[18]:=-0.2; X[19]:=-0.15; X[20]:=-0.09; X[21]:=-0.043;
 X[22]:=0.0084; X[23]:=0.06; X[24]:=0.111;
 X[25]:=1.65; X[26]:=1.68; X[27]:=1.69; X[28]:=1.7; X[29]:=1.71;
 X[30]:=1.7; X[31]:=1.7; X[32]:=1.68;
 X[33]:=-2.5; X[34]:=-2.64; X[35]:=-2.77; X[36]:=-2.9; X[37]:=-3.;
 X[38]:=-3.14; X[39]:=3.; X[40]:=2.9;
 FOR I:=1 TO 8 DO
  BEGIN
   XT[I]:=2.352;
   YT[I]:=0.223+(I-1)*0.14;
   X[I+40]:=0.563+0.03;
  END;
ITER:=10000;
H:=0.0015;
EPS:=1.E-05;
N:=48;
   GRADDERN(FUNCT,MF,N,ITER,H,EPS,X,KOD);
      IF KOD = 0 THEN
       BEGIN
        FOR I:=1 TO N DO
        BEGIN
         WRITELN('X[',I,']:=',X[I]:8:5);
         WRITELN(F1,'X[',I,']:=',X[I]:8:5);
         READLN;
        END;
        CLOSE(F1);
        HALT;
       END
      ELSE
       BEGIN
        WRITELN('KOD= ',KOD);
        READLN;
        HALT;
       END;
END.
```

Rezultatele calculelor sunt prezentate în tabelul 3:

Tabelul 3

X[1]:= 0.56992
X[2]:= 0.78725
X[3]:= 0.78762
X[4]:= 0.78556
X[5]:= 0.55477
X[6]:= 0.61275
X[7]:= 0.25571
X[8]:= 2.43556
X[9]:= 1.59491
X[10]:= 1.63084
X[11]:= 1.65576
X[12]:= 1.67060
X[13]:= 1.67581
X[14]:= 1.67150
X[15]:= 1.65747
X[16]:= 1.63322
X[17]:=-0.23295
X[18]:=-0.17377
X[19]:=-0.11510
X[20]:=-0.05696
X[21]:= 0.00072
X[22]:= 0.05809
X[23]:= 0.11535
X[24]:= 0.17284
X[25]:= 1.64535
X[26]:= 1.67310
X[27]:= 1.68914
X[28]:= 1.69589
X[29]:= 1.69489
X[30]:= 1.68709
X[31]:= 1.67303
X[32]:= 1.65297
X[33]:=-2.54222
X[34]:=-2.69956
X[35]:=-2.84400
X[36]:=-2.98187
X[37]:=-3.11803
X[38]:=-3.25636
X[39]:= 2.88296
X[40]:= 2.73053
X[41]:= 0.65907
X[42]:= 0.69182
X[43]:= 0.72405
X[45]:= 0.78767
X[46]:= 0.81929
X[47]:= 0.85088
X[48]:= 0.88239

Rezultatele din tabelul 7, obţinute în urma rulării programului de calcul din tabelul 6, au următoarele semnificaţii:

- valorile introduse în locaţiile 1-8, ale tabloului X, reprezintă dimensiunile elementelor şi poziţiile articulaţiilor la bază, şi anume:

$XE = X[1]$; $XG = X[2]$; $YG = X[3]$; $AB = X[4]$; $BC = X[5]$; $BF = X[6]$; $FG = X[7]$; $BT = X[8]$;

- în locaţiile 9-40, ale tabloului X, sunt introduse valorile unghiurilor φ_{1i}, $i = \overline{1,8}$; φ_{2i}, $i = \overline{1,8}$;

φ_{4i}, $i=\overline{1,8}$; φ_{5i}, $i=\overline{1,8}$;

- în locaţiile 41-48, ale tabloului X, sunt introduse valorile distanţelor S_{4i}, $i=\overline{1,8}$, dintre articulaţiile *C* şi *E* ale cilindrului hidraulic.

6. CONCLUZII

Folosind metodele de calcul numeric, se poate realiza cu uşurinţă sinteza oricărui mecanism. Operatorul trebuie să aibă minime cunoştinţe de matematică, de mecanisme şi de programare.

O mare atenție trebuie acordată alegerii soluției inițiale, pentru rezolvarea sistemelor de ecuații neliniare.

Pentru evitarea calculului laborios de obținere analitică a elementelor matricei funcționale a sistemelor de ecuații neliniare, se foloseşte cu succes metoda de derivare numerică a funcțiilor de mai multe variabile.

7. BIBLIOGRAFIE

[1]. Artobolevski, I.I. (1977), *Theorie des mecanismes et des machines*, Editions Mir, Moscou.
[2]. Demidovitch, B., Maron, I. (1976), *Elements de calcul numeriques*, Editions Mir, Moscou.
[3]. Duca, C., Buium, Fl., Părăoanu, G. (2003), *Mecanisme*, Editura Gh. Asachi, Iaşi.
[4]. Ghinea, M., Fireţeanu, V. (1995), *MATLAB. Calcul numeric+Grafică+Aplicaţii*, Editura Teora, Bucureşti.
[5]. Moise, V. (1986), *Cinematica şi controlul mecanismelor manipulatoare cu 4 grade de mobilitate*, Teză de doctorat, Bucureşti.
[6]. Moise, V. (2018), *Sinteza mecanismelor plane cu bare articulate*, Editura Printech, Bucureşti.
[7]. Moise, V., Maican, E., Moise, Şt. I. (2003), *Metode numerice în inginerie*, Editura Bren, Bucureşti.
[8]. Moise, V., Simionescu, I., Ene, M., Neacşa, M., Tabără, I.A. (2007), *Analiza mecanismelor aplicate*, Editura Printech, Bucureşti.
[9]. Pelecudi, Chr., Simionescu, I., Moise, V., Ene, M. (1981), *Proiectarea mecanismelor*, Institutul Politehnic Bucureşti.
[10]. Pelecudi, Chr., Simionescu, I., Ene, M., Moise, V., Candrea, A. (1982), *Probleme de mecanisme*, Editura Didactică şi Pedagogică, Bucureşti.
[11]. Pelecudi, Chr., Maroş, D., Merticaru, V., Pandrea, N., Simionescu, I. (1985), *Mecanisme*, Editura Didactică şi Pedagogică, Bucureşti.
[12]. Simionescu, I., Dranga, M., Moise, V. (1995), *Metode numerice în tehnică*, Ed. Tehnică, Bucureşti.
[13]. Simionescu, I., Moise, V. (1999), *Mecanisme*, Editura Tehnică, Bucureşti.
[14]. Voinea, R., Voiculescu, D., Ceauşu, V. (1985), *Mecanica*, Editura Didactică şi Pedagogică, Bucureşti.