NUMERIC-ANALYTICAL CONSTRUCTION OF MATHIEU FUNCTIONS

Yu. A. Ryabov, Moskva

(Received April 16, 1997)

Abstract. In this paper we present an iterative algorithm for the construction of Mathieu functions of any order N in the form of Fourier series (practically, polynomials), and also the corresponding Quick-BASIC program for realization of this algorithm with numerical values of the parameter.

Keywords: Mathieu functions, iterative algorithm, Quick-BASIC program

MSC 2000: 65D15, 34A50, 34A25

The Mathieu functions are solutions (see [1]) of the so called Mathieu equation

$$z'' + (a + q\cos 2t)z = 0, \tag{1}$$

where a, q are constants. If the constant q (taken as a parameter) is fixed, then periodic solutions of this equation exist only for certain values of a depending on q and are called the eigenvalues of this equation or the corresponding Mathieu functions. These values form two sequences:

$$a_{cn},\ n = 0, 1, 2, \ldots \quad \text{and} \quad a_{sn},\ n = 1, 2, \ldots$$

The eigenvalues a_{cn} correspond to the functions $ce_n(t)$ called the Mathieu cosine-functions of order n and these functions tend to $\cos nt$ as $q \to 0$. The values a_{sn} correspond to the functions $se_n(t)$ called the Mathieu sine-functions of order n tending to $\sin nt$ as $q \to 0$.

In [2], [3] certain variants of analytical iterative algorithms for the construction of Mathieu functions in the form of trigonometric series (practically, polynomials) have been considered. Convergence of these algorithms has been proved if the parameter

q does not exceed certain bounds (different for different Mathieu functions); certain estimates for these bounds have been obtained. In [2] we have presented examples of realization of these algorithms with the aid of a computer in the cases of $ce_1(t)$, $se_1(t)$. In the present paper we offer a unified algorithm for the construction of Mathieu functions of any order n in the form of Fourier polynomials and also the corresponding Quick-BASIC program for realization of this algorithm.

1. Construction of algorithm

We will seek Mathieu cosine- and sine-functions $ce_n(t)$, $se_n(t)$ of order n as periodic solutions of the equation

$$z'' + (n^2 + qh + q\cos 2t)z = 0, \tag{2}$$

where q is a given numerical parameter (> 0), the constant h is to be found simultaneously with functions $ce_n(t)$ or $se_n(t)$. The eigenvalues of these functions equal to $A = n^2 + qh$. (For negative values of the parameter q we did not obtain new Mathieu functions, see [1].)

In the case of $se_n(t)$, $n = 1, 2, 3, \ldots$ we put

$$z = \sin nt + x, \tag{3}$$

$$x = \sum_{L=1}^{L0} E_L \sin(n - 2L)t + \sum_{K=1}^{K0} F_K \sin(n + 2K)t, \tag{4}$$

where E_L, F_K are unknown coefficients, $K0$ is a relatively large number (we set $K0 = 50$), $L0 = \frac{1}{2}(n-1)$ for odd n and $L0 = \frac{1}{2}n - 1$ for even n.

In the case of $ce_n(t)$, $n = 0, 1, 2, \ldots$ we put

$$z = \cos nt + x, \tag{5}$$

$$x = \sum_{L=1}^{L0} E_L \cos(n - 2L)t + \sum_{K=1}^{K0} F_K \cos(n + 2K)t, \tag{6}$$

where $L0 = \frac{1}{2}(n-1)$ for odd n, $L0 = \frac{1}{2}n$ for even n and $E_{n/2}$ is replaced for even n by $\frac{1}{2}E_{n/2}$.

The equations for x in the cases (3), (5) are

$$x'' + n^2 x = -q\big[h\sin nt + \cos 2t \sin nt + (h + \cos 2t)x\big], \tag{7}$$

$$x'' + n^2 x = -q\big[h\cos nt + \cos 2t \cos nt + (h + \cos 2t)x\big], \tag{8}$$

Substituting (4), (6) into these equations and equating on the left- and righthand sides the corresponding expressions we obtain algebraic recurrent relations for F_K, E_L, h. These expressions are as follows.

Coefficients F_K, $K \geqslant 1$ for all functions $ce_n(t)$, $se_n(t)$ satisfy one and the same relation

$$F_K = G_K(F_{K-1} + 2hF_K + F_{K+1}), \tag{9}$$

where $G_K = q/(8K(n+K))$, $F_0 = 1$ for $n > 0$ and $F_0 = 2$ for $n = 0$.

Denoting $M_L = -q/(8L(n-L))$, $L = 1, 2, 3, \dots$ we obtain for coefficients E_1, E_2 in the case of functions $ce_4(t)$, $ce_n(t)$, $se_n(t)$, $n \geqslant 5$ the relation

$$E_1 = M_L(1 + 2hE_1 + E_2). \tag{10}$$

In the cases of functions $ce_2(t)$ and $se_2(t)$ we have

$$E_1 = 2M_1(1 + hE_1) \quad \text{and} \quad E_1 = 0, \tag{11}$$

respectively. If $N = 3$ then

$$E_1 = M_1(1 + 2hE_1 \pm E_1), \tag{12}$$

where plus corresponds to $ce_3(t)$ and minus to $se_3(t)$. In the case of se_4 we have

$$E_1 = M_1(1 + 2hE_1). \tag{13}$$

The constant h is connected with F_1, E_1 by the relations

$$\begin{aligned}
&(14) & h + \tfrac{1}{2}F_1 &= 0 && \text{for } ce_0(t), se_2(t),\\
&(15) & h \pm \tfrac{1}{2} + \tfrac{1}{2}F_1 &= 0 && + \text{ for } ce_1(t), \ - \text{ for } se_1(t),\\
&(16) & h + \tfrac{1}{2}(F_1 + E_1) &= 0 && \text{for } ce_2(t) \text{ and for all } ce_n(t), se_n(t), \ n > 2.
\end{aligned}$$

Coefficients E_L, $2 < L < \frac{1}{2}n$ (for even n), $2 < L < \frac{1}{2}(n-1)$ (for odd n) for functions $ce_n(t), se_n(t)$ satisfy the relations

$$E_L = M_L(E_{L-1} + 2hE_L + E_{L+1}). \tag{17}$$

If $L = \frac{1}{2}n$ (for even n) then

$$E_L = 2M_L(E_{L-1} + hE_L) \tag{18}$$

in the case of $ce_n(t)$ and $E_L = 0$ in the case of $se_n(t)$.

If $L = \frac{1}{2}(n-1)$ (for odd n) then

$$E_L = M_L(E_{L-1} + 2hE_L \pm E_L), \tag{19}$$

where plus corresponds to $ce_n(t)$ and minus to $se_n(t)$.

These are the basic relations. The required iterative algorithm for the calculation of h, F_K, E_L is derived from them. We can use the method of the so called simple iterations (see [2, 3]). Then, for example, in the case of $ce_2(t)$ we have

$$\begin{aligned} (20)\quad & F_1^{(1)} = G_1,\ E_1^{(1)} = 2M_1,\ h_1 = -\tfrac{1}{2}(F_1^{(1)} + E_1^{(1)}),\ F_K^{(1)} = G_K F_{K-1}^{(1)},\ K \geqslant 2,\\ & F_1^{(2)} = G_1(1 + 2h_1F_1^{(1)} + F_2^{(1)}),\ E_1^{(2)} = 2M_1(1 + h_1E_1^{(1)}),\quad \text{etc.} \end{aligned}$$

If we leave q as a variable parameter then these iterations enable us to obtain the approximations $F_K^{(j)}$, $E_L^{(j)}$, h_j, $j = 1, 2, 3, \dots$ in analytical form as polynomials in q. But the realization of simple iterations for different numerical values of q shows that the convergence range of these iterations is relatively small. For example, the upper bound q_* of the convergence range in the case of $ce_0(t)$ equals approximately 4,7–4,8.

We use in the present paper another algorithm (of irrational structure) realized for a given numerical value of q and having a sufficiently large convergence range.

2. The scheme of calculations

1. The calculation at the first step (the loop counter $IR = 1$) of the first approximation $F_1^{(1)}$, $E_1^{(1)}$, h_1 for F_1, E_1, h according to (10)–(16), where we put $F_2 = E_2 = 0$ (subprogram T1).

2. The calculation at the same step of the first approximation $F_K^{(1)}$, $E_L^{(1)}$, $K \geqslant 2$, $L \geqslant 2$ according to (9), (17)–(19), where we put $F_{K+1} = E_{L+1} = 0$ (subprogram T2).

3. The calculation at the second step ($IR = 2$) of the second approximation $F_1^{(2)}$, $E_1^{(2)}$, h_2 according to (10)–(16), where we put $F_2 = F_2^{(1)}$, $E_2 = E_2^{(1)}$ (subprogram T1).

4. The calculation at the same step of the second approximation $F_K^{(2)}$, $E_L^{(2)}$, $K \geqslant 2$, $L \geqslant 2$ according to (9), (17)–(19), where we put $F_{K+1} = F_{K+1}^{(1)}$, $E_{L+1} = E_{L+1}^{(1)}$ (subprogram T3).

5. The calculation at the third step ($IR = 3$) of the third approximation $F_1^{(3)}$, $E_1^{(3)}$, h_3 according to (10)–(16), where we put $F_2 = F_2^{(2)}$, $E_2 = E_2^{(2)}$ (subprogram T1), etc.

Subprogram T1 for the calculation of F_1, E_1, h consists of two parts. The first provides the calculation of F_1, h for

$$ce_0(t),\ ce_1(t),\ se_1(t),\ se_2(t) \quad \text{(then } E_1 = 0\text{)}.$$

For example, we have for $ce_1(t)$ according to (9), (16) and provided $F_2 = 0$

$$F_1^{(1)} = \tfrac{1}{2G}\big(\sqrt{1+4G^2}-1\big), \quad h_1 = -\tfrac{1}{2} - \tfrac{1}{2}F_1^{(1)},\ G = \tfrac{1}{4}q. \tag{21}$$

After calculating $F_K^{(1)}$, $K \geqslant 2$ following subprogram T2 we obtain at the second step $(IR = 2)$

$$F_1^{(2)} = \tfrac{1}{2G}\Big(\sqrt{1+4G^2(1+F_2^{(1)})}-1\Big), \qquad h_2 = -\tfrac{1}{2} - \tfrac{1}{2}F_1^{(2)}, \tag{22}$$

etc. Algorithms in the cases of $ce_0(t)$, $se_1(t)$, $se_2(t)$ are analogous.

In the cases of $ce_2(t)$, $ce_n(t)$, $se_n(t)$, $n \leqslant 3$ the relations for F_1, E_1, h are more complicated. It is possible to write these relations in the following unified form:

$$E_1 + F_1 = -2h, \tag{23}$$

$$E_1\,[1 + q_3 - q_1(E_1+F_1)] = -q_4(1+R), \tag{24}$$

$$F_1\,[1 + q_2(E_1+F_1)] = q_2(1+S), \tag{25}$$

where
$q_1 = q/(8n-8)$, $q_2 = q(8n+8)$, $S = F_2$,
$q_3 = 0$ for $n = 2$ and for all $n > 3$,
$q_3 = q_1$ for $ce_3(t)$ and $q_3 = -q_1$ for $se_3(t)$,
$q_4 = 2q_1$ for $n = 2$ and $q_4 = q_1$ in the other cases,
$R = 0$ for $ce_2(t)$, $ce_3(t)$, $se_3(t)$ and $R = E_2$ in the other cases.

The calculations of F_1, E_1, h provided R, S are known are realized by the second part of subprogram T1 (by subprogram T11 imbedded in T1).

At the first step $(IR = 1)$ we put $R = S = 0$ and consider relations (24)–(25) as a system of algebraic equations in E_1, F_1. If q is relatively small $(q < 2N)$, then we use the iterational Newton's algorithm with the initial approximation

$$E_1^0 = -q_4, \quad F_1^0 = q_2.$$

The maximal admissible number of Newton's iterations is taken to be $J1 = 150$. Iterations are stopped if adjacent approximations for F_1, E_1 coincide with accuracy

$\alpha = 10^{-10}$, and the values obtained are taken for the first approximation $F_1^{(1)}$, $E_1^{(1)}$. The first approximation

$$h_1 = -\tfrac{1}{2}\bigl(F_1^{(1)} + E_1^{(1)}\bigr)$$

is also calculated.

If the mentioned coincidence is not attained after 150 iterations, then the information "No convergence of T11" is displayed and printed out.

The initial values F_1^0, E_1^0 in the cases $q \geqslant 2N$ are selected more accurately (otherwise these iterations may converge to extraneous roots). For this purpose we reduce relations (23)–(25) provided R, S are known to a cubic equation in h:

$$c_0 h^3 + c_1 h^2 + c_2 h + c_3 = 0, \tag{26}$$

where $c_0, \dots, c_3$ are expressed in terms of $q_1, \dots, q_4$, R, S. For F_1, E_1 we obtain the expressions

$$F_1 = \frac{q_2(1+S)}{1-2hq_2}, \quad E_1 = -\frac{q_4(1+R)}{1+q_3+2hq_1} \tag{27}$$

The analysis and the corresponding calculations show that eq. (26) has for all considered values of n, q three real roots. The required root $h^0 = h^0(q)$ depends continuously on q and tends to zero as $q \to 0$. For the calculation of this root we use the Cardano-Hudde algorithm. Namely, substitution $h = y - c_1/3c_0$ reduces (26) to the equation

$$y^3 - b_1 y + b_2 = 0, \tag{28}$$

where b_1, b_2 are expressed in terms of $c_0, \dots, c_3$.

At the first step ($IR = 1$) we put $R = S = 0$ in the expressions of $c_0, \dots, c_3$. The required root y^0 is calculated in this case by the formula

$$y^0 = \sqrt{\frac{b_1}{3}}\bigl(-\cos D + \sqrt{3}\sin D\bigr), \tag{30}$$

where

$D = \frac{1}{3}D_1$ if $b_2 < 0$, and $D = D_1 + \pi$ if $b_2 > 0$,
$D_1 = \arctan D_0$, $-\frac{1}{2}\pi < D_1 < \frac{1}{2}\pi$, $D_0 = -2\sqrt{D_{00}}/b_2$,
$D_{00} = \frac{1}{27}b_1^3 - \frac{1}{4}b_2^2$.

After the calculation of y^0 we determine $h^0 = y^0 = -c_1/3c_0$, and also F_1^0, E_1^0 by (27) provided $R = S = 0$. These values F_1^0, E_1^0 are taken as initial values for the above mentioned Newton's iterations. As the result of a relatively small number of these iterations we obtain the required values $F_1^{(1)}$, $E_1^{(1)}$ with accuracy up to $\alpha = 10^{-10}$. Thereafter we calculate

$$h_1 = -\tfrac{1}{2}\bigl(F_1^{(1)} + E_1^{(1)}\bigr)$$

and $F_K^{(1)}$, $E_L^{(1)}$, $K \geqslant 2$, $L \geqslant 2$ according to subprogram T2.

At the second step ($IR = 2$) we put $R = E_2^{(1)}$, $S = F_2^{(1)}$ and use the same scheme for the calculation of $F_1^{(2)}$, $E_1^{(2)}$, h_2. If $q < 2n$, then we obtain these quantities from (24)–(25) using Newton's algorithm with the initial approximation $F_1^0 = F_1^{(1)}$, $E_1^0 = E_1^{(1)}$. If $q \geqslant 2n$, then the initial F_1^0, E_1^0 are calculated again with the aid of the cubic equation (26) etc.

The maximal amount of coefficients F_K calculated is restricted by the number $K0 = 50$. The amount of all coefficients E_L equals $L0$ and depends on the order n. The calculations show that the modules of the coefficients F_K, E_L decrease sufficiently quickly as their indices grow. It is natural to take into account only the first coefficients satisfying the estimates

$$|F_K| > \alpha = 10^{-10}, \qquad |E_L| > \alpha = 10^{-10}.$$

The amount of such coefficients obtained by calculations is denoted by $K1$, $L1$, respectively. As a rule, $K1 < K0$ and $L1 < L0$ for large N.

The maximal number IR of iterations taken is $Imax = 200$. The adjacent approximations for F_K, E_L ($K \leqslant K1$, $L \leqslant L1$), h corresponding to the steps IR and $IR+1$ are compared with each other. If they coincide with accuracy up to $eps = 10^{-8}$ at the step $IR \leqslant Imax$, then we take the iterational process for a convergent one (to say correctly, practically convergent), calculations are stopped and the values F_K, E_L, h obtained at the last step are taken for the final result. If the required coincidence is not attained for $IR \leqslant Imax$, then the calculations are stopped and the information "No convergence" for given n, ICS, q and also the results obtained at the last step $IR = 200$ are displayed and printed out.

3. Quick-BASIC program

```
PRINT "Construction of Mathieu functions CE(t),SE(t) of order N(<=100)"
PRINT "for diff.eq. Z'' + (N^2 + QH + Qcos2t) Z = 0 in form of polynomials"
PRINT "...+E(2)cos(N-4)t+E(1)cos(N-2)t+cosNt+F(1)cos(N+2)t+F(2)cos(N+4)t+..."
PRINT "or"
PRINT "...+E(2)sin(N-4)t+E(1)sin(N-2)t+sinNt+F(1)sin(N+2)t+F(2)sin(N+4)t+..."
PRINT "for given numerical value of parameter Q. Coefficients E(1),...,"
PRINT " E(L0),F(1),...,F(K0) and constant H are required quantities"
```

```
PRINT "Maximal number of coeff.E(L) equals L0,maximal number of"
PRINT "coeff.F(K0) is taken K0.The indicator ICS=1 corresponds to CE(t)"
PRINT "and indicator ICS=0 corresponds to SE(t)."
DEFDBL B-H, M, Q-S, X-Z
DIM F(2, 51), E(2, 51), G(50), H(2), M(50), U(50), V(50)
K0 = 50: IMAX = 200: EPS = 1E-08: ALPH = 1E-10
100 :
INPUT ; "N="; N: INPUT ; "  ICS="; ICS: INPUT "   Q="; Q
IF N MOD 2 = 0 THEN
        L0 = N / 2
ELSE
         L0 = (N - 1) / 2
END IF
FOR K = 2 TO K0
        G(K) = Q / (8 * K * (N + K))
NEXT K
FOR L = 2 TO L0
       M(L) = -Q / (8 * L * (N - L))
NEXT L
FOR J = 1 TO 2
FOR I = 1 TO 51
F(J, I) = 0: E(J, I) = 0
NEXT I
NEXT J
IR = 1
R = 0: S = 0
GOSUB T1 'Calculation of X,Y,Z for F(1,1),E(1,1), H(1)
F(1, 1) = Y: E(1, 1) = X: H(1) = Z
GOSUB T2 'Calculation of F(1,K), E(1,L), K>1, L>1
NextIteration:
IR = IR + 1
R = E(1, 2): S = F(1, 2)
GOSUB T1 'Calculation of X,Y,Z for F(2,1), E(2,1), H(2)
F(2, 1) = Y: E(2, 1) = X: H(2) = Z
GOSUB T3 'Calculation of F(2,K), E(2,L), K1, L1
W0 = H(2) - H(1)
IF N + ICS > 2 THEN
U1 = F(2, 1) - Q2 * (1 + 2 * H(2) * F(2, 1) + F(2, 2))
END IF
IF N = 2 AND ICS = 1 THEN
   V1 = E(2, 1) + 2 * Q1 * (1 + H(2) * E(2, 1))
ELSEIF N = 3 AND ICS = 1 THEN
   V1 = E(2, 1) + Q1 * (1 + 2 * H(2) * E(2, 1) + E(2, 1))
ELSEIF N = 3 AND ICS = 0 THEN
   V1 = E(2, 1) + Q1 * (1 + 2 * H(2) * E(2, 1) - E(2, 1))
ELSEIF N = 4 AND ICS = 1 THEN
   V1 = E(2, 1) + Q1 * (1 + 2 * H(2) * E(2, 1) + E(2, 2))
ELSEIF N = 4 AND ICS = 0 THEN
   V1 = E(2, 1) + Q1 * (1 + 2 * H(2) * E(2, 1))
ELSEIF N - 4 > 0 THEN
```

```
  V1 = E(2, 1) + Q1 * (1 + 2 * H(2) * E(2, 1) + E(2, 2))
END IF
FOR J = 1 TO K1
   U(J) = F(2, J) - F(1, J)
NEXT J
FOR J = 1 TO L1
   V(J) = E(2, J) - E(1, J)
NEXT J
IF K1 < L1 THEN J1 = L1 ELSE J1 = K1
FOR J = 1 TO J1
     IF ABS(U(J)) > EPS OR ABS(V(J)) > EPS GOTO 200
NEXT J
IF ABS(W0) > EPS GOTO 200
IF ABS(U1) > EPS OR ABS(V1) > EPS GOTO 200
GOTO PrintRes
200 :
IF IR = IMAX GOTO 300
FOR J = 1 TO J1
    F(1, J) = F(2, J): E(1, J) = E(2, J): H(1) = H(2)
NEXT J
GOTO NextIteration
300 :
PRINT "No convergence for Q="; Q; SPC(3); "N="; N; SPC(3); "ICS="; ICS
LPRINT "No convergence for Q="; Q; SPC(3); "N="; N; SPC(3); "ICS="; ICS
PrintRes:
A = N ^ 2 + Q * H(2)
PRINT "N="; N; SPC(3); "ICS="; ICS; SPC(3); "Q="; Q
PRINT "IR="; IR; SPC(3); "K1="; K1; SPC(3); "L1="; L1
LPRINT "N="; N; SPC(3); "ICS="; ICS; SPC(3); "Q="; Q
LPRINT "IR="; IR; SPC(3); "K1="; K1; SPC(3); "L1="; L1
IF N + ICS > 2 THEN
  PRINT "U1="; U1; SPC(1); "V1="; V1; SPC(1); "D00="; D00
LPRINT "U1="; U1; SPC(1); "V1="; V1; SPC(1); "D00="; D00
END IF
PRINT "H="; H(2); SPC(1); "W0="; W0; SPC(1); "A="; A
LPRINT "H="; H(2); SPC(1); "W0="; W0; SPC(1); "A="; A
FOR J = 1 TO K1
PRINT "F("; J; ")="; F(2, J); : PRINT TAB(35); "U("; J; ")="; U(J)
LPRINT "F("; J; ")="; F(2, J); : LPRINT TAB(35); "U("; J; ")="; U(J)
IF J MOD 17 = 0 THEN
   PRINT "press any key"
   AA$ = "":  WHILE AA$ = "": AA$ = INKEY$: WEND
END IF
NEXT J
PRINT "press any key"
AA$ = "": WHILE AA$ = "": AA$ = INKEY$: WEND
FOR J = 1 TO L1
PRINT "E("; J; ")="; E(2, J); : PRINT TAB(35); "V("; J; ")="; V(J)
LPRINT "E("; J; ")="; E(2, J); : LPRINT TAB(35); "V("; J; ")="; V(J)
IF J MOD 17 = 0 THEN
```

```
   PRINT "press any key"
   AA$ = "":  WHILE AA$ = "": AA$ = INKEY$: WEND
END IF
NEXT J
400 :
PRINT "If you want to repeat calculations with others or with the same"
PRINT "parameters N,ICS,Q, then enter 1 else 0"
INPUT "W="; W
IF W = 1 GOTO 100
PRINT "Calculations are ended": END
T1: 'Calculation of X,Y,Z  for F1, E1, H
IF N = 0 THEN
   G = Q / 4
   Y = (SQR(1 + (2 + S) * G ^ 2) - 1) / G: X = 0: Z = -Y / 2
   RETURN
ELSEIF N = 1 AND ICS = 1 THEN
   G = Q / (16 + Q): X = 0
   Y = (SQR(1 + 4 * (1 + S) * G ^ 2) - 1) / (2 * G):
   Z = -(1 + Y) / 2
   RETURN
ELSEIF N = 1 AND ICS = 0 THEN
   G = Q / 16: X = 0
   Y = (G - 1 + SQR((G - 1) ^ 2 + 4 * (1 + S) * G ^ 2)) / (2 * G)
   Z = (1 - Y) / 2
RETURN
END IF
IF N > 1 THEN
      Q1 = Q / (8 * (N - 1)): Q2 = Q / (8 * (N + 1))
END IF
IF N = 2 AND ICS = 1 THEN
      Q4 = 2 * Q1: Q3 = 0
ELSEIF N = 2 AND ICS = 0 THEN
      G = Q / 24: X = 0
      Y = (SQR(1 + 4 * (1 + S) * G ^ 2) - 1) / (2 * G): Z = -Y / 2
      RETURN
END IF
IF N > 2 THEN Q4 = Q1
IF N = 3 AND ICS = 1 THEN
     Q3 = Q1
ELSEIF N = 3 AND ICS = 0 THEN
     Q3 = -Q1
ELSEIF N > 3 THEN
     Q3 = 0
END IF
GOSUB T11
RETURN
T2: 'Calculation of F(1,K),E(1,L), K>1,L>1
FOR K = 2 TO K0
      F(1, K) = G(K) * F(1, K - 1) / (1 - 2 * H(1) * G(K))
NEXT K
```

```
IF N = 4 AND ICS = 1 THEN
      E(1, 2) = -Q * E(1, 1) / (16 + H(1) * Q)
ELSEIF N = 4 AND ICS = 0 THEN
      E(1, 2) = 0
END IF
IF N = 5 THEN
     IF ICS = 1 THEN E1 = 1 ELSE E1 = -1
     E(1, 2) = -Q * E(1, 1) / (48 + Q * (2 * H(1) + E1))
END IF
IF N > 5 THEN
     L00 = L0 - 1
     FOR J = 2 TO L00
           E(1, J) = M(J) * E(1, J - 1) / (1 - 2 * H(1) * M(J))
     NEXT J
     J = L0
     IF N MOD 2 = 0 THEN
        IF ICS = 1 THEN
            E(1, J) = 2 * M(J) * E(1, J - 1) / (1 - 2 * H(1) * M(J))
        ELSE
            E(1, J) = 0
        END IF
     ELSE
        IF ICS = 1 THEN E1 = 1 ELSE E1 = -1
        E(1, J) = M(J) * E(1, J - 1) / (1 - M(J) * (2 * H(1) + E1))
     END IF
END IF
RETURN
T11: 'Calculation  of X,Y,Z for F1,E1,H
C0 = Q1 * Q2
C1 = (Q2 * (1 + Q3) - Q1) / 2
C2 = (-1 - Q3 - Q2 * Q4 * (1 + R) - Q1 * Q2 * (1 + S)) / 4
C3 = (Q4 * (1 + R) - Q2 * (1 + Q3) * (1 + S)) / 8
B1 = C1 ^ 2 / (3 * C0 ^ 2) - C2 / C0
B2 = -C1 * C2 / (3 * C0 ^ 2) + (2 * C1 ^ 3) / (27 * C0 ^ 3) + C3 / C0
D00 = B1 ^ 3 / 27 - B2 ^ 2 / 4
N1 = 2 * N
IF Q < N1 GOTO 500
D0 = -SQR(D00) * 2 / B2
D1 = ATN(D0)
IF B2 < 0 THEN
     D = D1 / 3
ELSE
     D = (D1 + 4 * ATN(1)) / 3
END IF
Z0 = SQR(B1 / 3) * (-COS(D) + SQR(3) * SIN(D)) - C1 / (3 * C0)
X1 = -Q4 * (1 + R) / (1 + Q3 + 2 * Q1 * Z0)
Y1 = Q2 * (1 + S) / (1 - 2 * Q2 * Z0)
JT = 1: J1 = 150
GOTO 600
500 :
```

```
IF IR = 1 THEN
     X1 = -Q4: Y1 = Q2
ELSE
   X1 = E(1, 1): Y1 = F(1, 1)
END IF
JT = 1: J1 = 150
600 :
F1 = X1 * (1 + Q3) - Q1 * (X1 + Y1) * X1 + Q4 * (1 + R)
F2 = Y1 + Q2 * (X1 + Y1) * Y1 - Q2 * (1 + S)
DF1 = (1 + Q3) * (1 + Q2 * (X1 + 2 * Y1))
DF = DF1 - Q1 * (2 * X1 + Y1) - 2 * Q1 * Q2 * (X1 + Y1) ^ 2
DF2 = (1 + Q2 * (X1 + 2 * Y1)) * F1 + Q1 * X1 * F2
DF3 = (1 + Q3 - Q1 * (2 * X1 + Y1)) * F2 - Q2 * Y1 * F1
X2 = X1 - (DF2 / DF): Y2 = Y1 - (DF3 / DF)
IF ABS(X2 - X1) > ALPH OR ABS(Y2 - Y1) > ALPH GOTO 700
X = X2: Y = Y2: Z = -(X + Y) / 2
RETURN
700 :
IF JT > J1 GOTO 800
X1 = X2: Y1 = Y2
JT = JT + 1
GOTO 600
800 :
PRINT "No convergence of T11 for IR="; IR
PRINT "Press any key"
AA$ = "": WHILE AA$ = "": AA$ = INKEY$: WEND
GOTO 400
T3: 'Calculation of F(2,K),E(2,L),K=2 to K1,L=2 to L1
FOR K = 2 TO K0
F(2, K) = G(K) * (F(2, K - 1) + F(1, K + 1)) / (1 - 2 * H(2) * G(K))
IF ABS(F(2, K)) < ALPH THEN EXIT FOR
NEXT K
K1 = K - 1
IF N = 4 AND ICS = 1 THEN
    E(2, 2) = -Q * E(2, 1) / (16 + H(2) * Q)
ELSEIF N = 4 AND ICS = 0 THEN
    E(2, 2) = 0
END IF
IF N = 5 THEN
    IF ICS = 1 THEN E1 = 1 ELSE E1 = -1
    E(2, 2) = -Q * E(2, 1) / (48 + Q * (2 * H(2) + E1))
END IF
IF N > 5 THEN
    L00 = L0 - 1
    FOR J = 2 TO L00
      E(2, J) = M(J) * (E(2, J - 1) + E(1, J + 1)) / (1 - 2 * H(2) * M(J))
    NEXT J
    J = L0
    IF N MOD 2 = 0 THEN
         IF ICS = 1 THEN
```

```
            E(2, J) = 2 * M(J) * E(2, J - 1) / (1 - 2 * M(J) * H(2))
         ELSE
            E(2, J) = 0
         END IF
    ELSE
         IF ICS = 1 THEN E1 = 1 ELSE E1 = -1
         E(2, J) = M(J) * E(2, J - 1) / (1 - M(J) * (2 * H(2) + E1))
    END IF
END IF
FOR J = 1 TO L0
    I = L0 + 1 - J
    IF ABS(E(2, I)) > ALPH THEN EXIT FOR
NEXT J
L1 = L0 + 1 - J
RETURN
STOP
```

A d d i t i o n a l r e m a r k s. The above presented Quick-BASIC program realizes the described algorithm with double precision. The program runs as an ordinary Quick-BASIC program. The realization of the algorithm begins after introducing from keyboard (as the answer to inquiry) of the order N, of the indicator $ICS = 1$ or $ICS = 0$ for the functions $ce_N(t)$, $se_N(t)$, respectively, and the numerical value of parameter q. If the iterational process converges, then the following data are displayed and printed:

1) given values of N, ICS, q;

2) number IR of steps providing required accuracy $eps = 10^{-8}$ of coefficients F_K, E_L;

3) numbers $K1$, $L1$ of coefficients F_K, E_L exceeding the modulus $\alpha = 10^{-10}$;

4) residual errors $U1$, $V1$ obtained after substituting the final values of F_1, F_2, E_1, E_2, h into relations (24)–(25);

5) values of h, of coefficients F_K, E_L, $1 \leqslant K \leqslant K1$, $1 \leqslant L \leqslant L1$ (with double precision), differences $W0$, $U(K)$, $V(K)$ between two last approximations for h, F_K, E_L, respectively, and also the quantity $D00$ being proportional to the discriminant of the cubic equation (26).

If $K > 17$, then the first 17 coefficients F_K are displayed and printed and after pressing any key the next 17 etc. The output mode of coefficients E_L is the same.

Maximal order N of Mathieu functions provided by the program is equal to 100.

Note that the users of this program can change such parameters of the program as $K0$, $Imax$, α, eps, and also DIMENSION for calculated coefficients F_K, E_L and quantities G_K, M_L, $U(K)$, $V(L)$.

The calculations realized with this program show very large convergence range of the proposed algorithm in the cases of $ce_0(t)$, $ce_1(t)$, $se_1(t)$, $se_2(t)$. The iterations converge, at least, for $0 < q < 7000$. In the cases of other Mathieu functions the convergence range is much smaller, but nevertheless sufficiently large and besides, this range widens for Mathieu functions of large order N.

The values of q near to the upper bound of the practical convergence range (corresponding values of IR are near to 200) in the cases of different functions $ce_N(t)$, $se_N(t)$ are presented in the following table:

ce_N	ce_2	ce_3	ce_4	ce_5	ce_6	ce_7	ce_8	ce_9	ce_{10}	ce_{20}	ce_{30}	ce_{40}	ce_{50}	ce_{100}
q_*	40,5	160	33	59	48	65	67	80	88	182	273	365	458	915
se_N		se_3	se_4	se_5	se_6	se_7	se_8	se_9	se_{10}	se_{20}	se_{30}	se_{40}	se_{50}	se_{100}
q_*		30,3	70	40	60	58	72	78	88	182	270	360	458	915

The example of printout (in the case of $ce_2(t)$, $q = 32$):

```
N= 2    ICS= 1    Q= 32
IR= 36    K1= 11    L1= 1
U1=-6.795683E-09  V1= 4.56666E-16  D00= .2469067098022291
H= 1.160101914937828D-02  W0=-1.631753E-10  A= 4.371233
F( 1 )= 7.297387560717799         U( 1 )= 9.071099E-09
F( 2 )= 4.303726409970468         U( 2 )= 5.096762E-09
F( 3 )= 1.210210035599587         U( 3 )= 1.367956E-09
F( 4 )= .2064818841575251         U( 4 )= 2.250614E-10
F( 5 )= 2.389046878665688D-02     U( 5 )= 2.531559E-11
F( 6 )= 2.00541015603754D-03      U( 6 )= 2.078218E-12
F( 7 )= 1.27923482678711D-04      U( 7 )= 1.302111E-13
F( 8 )= 6.416610614172347D-06     U( 8 )= 6.435982E-15
F( 9 )= 2.598511597751983D-07     U( 9 )= 2.5746E-16
F( 10 )= 8.67651371123749D-09     U( 10 )= 8.508039E-18
F( 11 )= 2.429387587345403D-10    U( 11 )= 2.359225E-19
press F5
E( 1 )=-7.320589599016555         V( 1 )=-8.744748E-09
If you want to repeat calculations with others or with the same
parameters N,ICS,Q, then enter 1 else 0
W=?
```

References

[1] *Ch. Hayashi*: Nonlinear Oscillations in Physical Systems. McGraw-Hill, New York, 1964.

[2] *Yu. A. Ryabov*: The estimations of convergence radius for series in power of parameter and algorithms for construction of Mathieu functions. Differencialnye uravnenija *15* (1979), no. 11, 1993–2003. (In Russian.)

[3] *E. A. Grebenikov, Yu. A. Ryabov*: Constructive methods in the analysis of nonlinear systems. Mir Publishers, Moscow, 1983, translated from Russian.

Author's address: *Yu. A. Ryabov*, Auto and Road Construction Engineering University, Leningradsky pr. 64, Moskva, 125829, Russia, e-mail: `vmath@madi.msk.su`.