ALGORITHM 73
INCOMPLETE ELLIPTIC INTEGRALS

DAVID K. JEFFERSON
U. S. Naval Weapons Laboratory, Dahlgren, Virginia

```
procedure ellint (k, phi, E, F);
value k, phi;
real phi, F, k, E;
comment ellint computes the value of the incomplete elliptic
    integrals of the first and second kinds, F(phi, k) and E(phi, k),
    where phi is in radians. If | k | > 1 or | phi | > π/2, E and F
    will be set equal to 100,000,000, otherwise they will contain the
    computed integrals. For the formulation of this procedure, see
    DiDonato, A. R., and Hershey, A. V., "New Formulae for
    Computing Incomplete Elliptic Integrals of the First and
    Second Kind", J. ACM 6, 4 (Oct. 1959);
begin real kp, sinphi, n, cosphi;
    real array H [1:2], A [1:2], sigma [1:4], L [1:2], M [1:2],
      N [1:2], T [1:2], del [1:4];
    sigma [1] := sigma [2] := sigma [3] := sigma [4] := 0;
    H [1] := 1;
    n := 0;
    sinphi := sin(phi);
if abs (k × sinphi) ≤ tanh (1) then go to small else if abs (k) ≤
    1 ∧ abs(phi) ≤ π/2 then go to large;
    E := F := 100000000;
    go to stop;
small: A [1] := phi;
step 1: n := n + 1;
    cosphi := cos (phi);
    E := (2 × n−1) /(2 × N);
    H [2] := E × k↑2 × H [1];
    A [2] := E × A [1] − sinphi ↑ (2 × n−1) × cosphi /(2 × n);
    del [1] := H [2] × A [2];
    del [2] := −k↑2 × H [1] × A [2] /(2 × n);
    sigma [1] := sigma [1] + del [1];
    sigma [2] := sigma [2] + del [2];
    H [1] := H [2];
    A [1] := A [2];
if abs ((sigma [1] + del [1]) − sigma [1]) > 0 ∧ phi × sinphi
    ↑ (2 × n) ≥ A [2] then go to step 1;
    F := phi + sigma [1];
    E := phi + sigma [2];
    go to stop;
large: kp := sqrt (1−k↑2);
    A [1] := 1;
    L [1] := M [1] := N [1] := 0;
step 2: n := n+1;
    E := (2 × n−1) /(2 × n);
    F := abs (k) × sqrt (1−sinphi ↑2) × (1−k↑2 × sinphi
      ↑2)↑((2 × n−1) /(2 × n));
    H [2] := E × H [1];
    A [2] := E↑2 × kp↑2 × A [1];
    L [2] := L [1] + 1 /(n × 2 × n−1));
 M [2] := (M [1] − F × H [2]) × ((2 × n+1) /(2 × n+2))↑2 ×
    kp↑2;
N [2] := (N[1] − F × H [1]) × E × (2 × n+1) × kp↑2 /(2 ×
    n+2);
del [1] := M [2] − A [2] × L [2];
del [2] := N [2] − E × kp↑2 × A [1] × L [2] + kp↑2 × A [1]
    /((2 × n) ↑2);
del [3] := A [2];
del [4] := (2 × n+1) × A [2] /(2 × n+2);
sigma [1] := sigma [1] + del [1];
sigma [2] := sigma [2] + del [2];
sigma [3] := sigma [3] + del [3];
sigma [4] := sigma [4] + del [4];
H [1] := H [2];
A [1] := A [2];
L [1] := L [2];
M [1] := M [2];
N [1] := N [2];
if abs ((sigma [1] + del [1]) − sigma [1]) > 0 then go to step 2;
T [1] := ln (4 /(sqrt (1 − k↑2 × sinphi ↑2) + abs (k) × sqrt(1−
    sinphi ↑ 2)));
Γ [2] := abs (k) × sqrt ((1−sinphi ↑2) /(1−k↑2 × sinphi ↑2));
F := T [1] × (1+sigma [3]) + T [2] × ln (.5 + .5 × abs (k ×
    sinphi)) + sigma [1];
E := (.5 + sigma [4]) × kp↑2 × T [1] + 1−T [2] × (1−abs
    (k × sinphi)) + sigma [2];
stop: end
```

CERTIFICATION OF ALGORITHM 73
INCOMPLETE ELLIPTIC INTEGRALS (David K. Jefferson, *Comm. ACM*, Dec. 1961)

DEAN C. KRIEBEL
U. S. Naval Weapons Laboratory, Dahlgren, Virginia

This algorithm was originally coded in NORC machine language and K. Pearson's incomplete elliptic integral tables of the first and second kind generated. (See DiDonato, A. R., and Hershey, A. V., "New Formulae for Computing Incomplete Elliptic Integrals of the First and Second Kind", *J.ACM 6*, 4 (Oct. 1959)).

The algorithm was coded for the MAD Compiler exactly as written in ALGOL and run on an IBM 7090. Forty cases were computed with K ranging from 0° to 90° and PHI ranging from 0° to 90°. The results contained eight significant digits which agreed with the DiDonato and Hershey tables to within 0 to 2 units in the 8th digit. (This may be attributed to the decimal to binary, binary to decimal input-output conversion used with a binary computer as compared to straight decimal computation on the NORC.)

CERTIFICATION OF ALGORITHM 73
INCOMPLETE ELLIPTIC INTEGRALS [David K. Jefferson, *Comm. ACM 4*, Dec. 1961]

NOELLE A. MEYER
E. I. du Pont de Nemours & Co., Wilmington, Del.

Ellint was hand-coded in FORTRAN for the IBM 7070. The following corrections were made

The statement

$$E := (2\times n-1)/(2\times N);$$

should be

E := (2×*n*−1)/(2×*n*);

The statement

F := *abs*(*k*) × *sqrt*(1−*sinphi* ↑ 2) × (1−*k* ↑ 2×*sinphi* ↑ 2) ↑ ((2×*n*−1)/(2×*n*));

should be

F := (*abs*(*k*)×*sqrt*(1−*sinphi* ↑ 2) × (1−*k* ↑ 2×*sinphi* ↑ 2) ↑ (*n*−.5))/(2×*n*) ;

The statement

L[2] := *L*[1] + 1/(*n*×2×*n*−1));

should be

L[2] := *L*[1] + (1/(*n*×(2×*n*−1));

In order to accommodate negative ϕ the following changes were made:

The statement

if *abs*((*sigma*[1]+*del*[1])−*sigma*[1]) > 0 ∧ *phi* × *sinphi* ↑ (2×*n*) ≥ *A*[2] **then go to** *step* 1;

was changed to

if *abs*((*sigma*[1]+*del*[1])−*sigma*[1])>0∧*abs*(*phi*×*sinphi* ↑ (2×*n*)) ≥ *abs*(*A*[2]) **then go to** *step* 1;

Also the following was inserted before the last statement (*stop*: **end**)

if *phi* < 0 **then go to** *wait* **else go to** *stop*;
wait: *F* := −*F*;
E := −*E*;

The revised algorithm yielded satisfactory answers when compared with the DiDonato and Hershey tables. Differences occurred in the eighth significant digit as shown in the following differencc tables.

DIFFERENCE TABLES

F-TABLE

ϕ (in degrees)	θ (in degrees) 0	30	60	90
0	0.	0.	0.	0.
30	-1×10^{-8}	-1×10^{-8}	-1×10^{-8}	-3×10^{-8}
60	1×10^{-8}	1×10^{-8}	2×10^{-8}	-3×10^{-8}
90	0.	2×10^{-8}	6×10^{-8}	0.

E-TABLE

0	0.	0.	0.	0.
30	-1×10^{-8}	-1×10^{-8}	-1×10^{-8}	-1×10^{-8}
60	1×10^{-8}	1×10^{-8}	-7×10^{-8}	3×10^{-8}
90	0.	0.	1×10^{-8}	0.

CERTIFICATION OF ALGORITHM 73 INCOMPLETE ELLIPTIC INTEGRALS [David K Jefferson, *Comm. ACM* Dec. 1961]

R. P. VAN DE RIET
Mathematical Centre, Amsterdam

The algorithm contained three misprints:
The 26th line of the procedure

E := (2 × *n*−1)/(2 × *N*);

should read

E := (2 × *n*−1)/(2 × *n*);

The 46th line of the procedure

↑ 2) ↑ ((2 × *n*−1)/(2 × *n*));

should read

↑ 2) ↑ ((2 × *n*−1)/2)/(2 × *n*);

The 49th line of the procedure

L [2] := *L* [4] + 1/(*n* × 2 × *n*−1));

should read

L [2] := *L* [1] + 1/(*n* × (2 × *n*−1));

The program was run on the X1 computer of the Mathematical Centre. For *phi* = 45°, *k* = *sin*(10°(10°)180°), *E* and *F* were calculated. The result contained 12 significant digits.

Comparison with a 12-decimal table of Legendre-Emde (1931) showed that the 12th digit was affected with an error, at most 4 units large. After about 10 minutes of calculation (i.e. more than 100 cycles) no results were obtained for *k* = *sin* 89°, *phi* = 1° and the calculation was discontinued.

REMARKS. As *phi* is unchanged during the calculation, we placed the statement *cos phi* := *cos* (*phi*) in the beginning of the program, to be certain that the cosine was not calculated 30 or more times. Moreover, in the expression for *T*[1] and *T*[2], *sqrt* (1-*sin phi* ↑ 2) was replaced by *cos phi*, so that loss of significant figures does not occur.

The expression 2 × *n* was changed in a new variable, to obtain a more rapid program.

REMARK ON ALGORITHM 73 INCOMPLETE ELLIPTIC INTEGRALS [David K. Jefferson, *Comm. ACM* (Dec. 1961)]

DAVID K. JEFFERSON
U. S. Naval Weapons Laboratory, Dahlgren, Virginia

In regard to Algorithm 73, two errors were found:
The 34th line of the procedure

F := *abs*(*k*) × *sqrt* (1−*sinphi* ↑ 2)
× (1−*k* ↑ 2 × *sinphi* ↑ 2) ↑ ((2 × *n*−1)/(2 × *n*));

should read

F := *abs*(*k*) × *sqrt* (1−*sinphi* ↑ 2)
× (1−k ↑ 2 × *sinphi* ↑ 2) ↑ ((2 × *n*−1)/2)/(2 × *n*);

The 37th line

L[2] := *L*[1] + 1/(*n* × 2 × *n*−1));

should read

L[2] := *L*[1] + 1/(*n* × (2 × *n*−1));

In addition, efficiency is improved by interchanging lines 13 and 14:

Step 1: *n* := *n*+1;
cosphi := *cos*(*phi*);

can be replaced by

cosphi:=*cos*(*phi*);
Step 1: *n* := *n* + 1;