Raabo.

A/S REGNECENTRALEN
SCANDINAVIAN INFORMATION PROCESSING SYSTEMS

SYSTEM LIBRARY

RCSL No: 53-M3

TYPE : Algol 5 Procedure

EDITION: September 1969.

AUTHOR : Tove Asmussen

RC 4000 SOFTWARE

MATHEMATICAL PROCEDURE LIBRARY

Bessel Functions - In(x) and Kn(x)

KEYWORDS
Mathematical, complete description, algol procedure, bessel.

ABSTRACT
The procedure besselik calculates by recurrence the values of the modified Bessel functions: I0(x),I1(x),...,In(x) and K0(x),K1(x),...,Kn(x).

1. Function and Parameters.

besselik calculates the modified Bessel functions:
I0(x),...,In(x) and K0(x),...,Kn(x).

Procedure heading:

```
procedure besselik (n,x,I,K);
value n,x; real x; integer n;
array I,K;
```

Call parameters:

- n : (real or integer) (real is rounded to nearest integer) maximum order of the Bessel functions. n must be >=0.
- x : (real or integer) the argument, must be > 0.

Return parameters:

- I : (real array I(0:n)) the values of the calculated functions: I(0)=I0(x),...,I(n)=In(x).
- K : (real array K(0:n)) the values of the calculated functions: K(0)=K0(x),...,K(n)=Kn(x).

2. Method.

First besselik calculates all the values of Ij(x), see [1].

The recurrence is performed from an upper bound nb. If x+14<=n+3 this integer is set equal n+4 otherwise x+14.
Then i(nb,x) is assigned the value of (x/2)××nb/(1×2×...×nb), while i(nb+1,x),...,i(n,x) is set to 0.
i(nb-1,x),...,i(0,x) are then computed from the recurrence formula

$$i(j-1,x) = 2\times j/x\times i(j,x) + i(j+1,x).$$

But since Ij(x)/i(j,x) is the same number for all j<=nb, Ij(x) can be calculated from the formula

$$\exp(x) = I0(x) + 2\times\sum_{k=1}^{nb} Ik(x)$$

by replacing x by abs x.

Then K0(x) and K1(x) are calculated by polynomial approximation see [2]:

```
if 0<x<2 then
  K0(x) := P1((x/2)××2)   -ln(x/2)×I0(x);
  K1(x) := P2((x/2)××2)/x +ln(x/2)×I1(x);
else
  K0(x) := P3(2/x)/exp(x)/sqrt(x);
  K1(x) := P4(2/x)/exp(x)/sqrt(x);
where Pi(x) is a polynomial of 6. degree.
```

Further values of Kj(x) are calculated by the recurrence formula:

K[i+1] = 2×i/x×K(i) + K(i-1).

3. Accuracy, Time and Storage Requirement.

Accuracy:	relative error $<_{10}-7$
Time:	approx. 10 + 0.5×n ms
	e.g. x=2 , n=10: 11 ms
	x=10, n=40: 29 ms
Storage requirement:	3 segments of program and 8 local real variables.
Typographical length:	87 lines incl. last comment.

4. Test and Discussion.

The algorithm is similar to the GIER procedure O.No. 179. [3].

Below program gives the following output:

```
begin

comment here the procedure is copied unless it is already
translated as an external;

integer i,n; real x;
  write(out,<:<12><10> n     x:>,false add 32,13,<:I(n):>,
            false add 32,18,<:K(n)<10>:>);
AGAIN:
  read(in,n); if n=-1 then goto END;
  begin array K,I(0:n);
  read(in,x);
  besselik(n,x,I,K);
  write(out,<:<10>:>,<<dd>,n,<<dd.dd>,x,
            <<    -d.ddddd ddddd₁₀-dd>,I(n),K(n) );
  goto AGAIN;
  end inner block;
END:
end

data:
0, 0.01
0, 0.5
0, 5
1, 5
10, 5
20, 5
-1,
```

n	x	I(n)	K(n)
0	0.01	1.00002 50003$_{10}$ 0	4.72124 47360$_{10}$ 0
0	0.50	1.06348 33708$_{10}$ 0	9.24419 07256$_{10}$ -1
0	5.00	2.72398 71829$_{10}$ 1	3.69109 83816$_{10}$ -3
1	5.00	2.43356 42146$_{10}$ 1	4.04461 33826$_{10}$ -3
10	5.00	4.58004 44196$_{10}$ -3	9.75856 28020$_{10}$ 0
20	5.00	5.02423 93598$_{10}$-11	4.82700 05078$_{10}$ 8

5. References.

[1] Goldstein and Thaler: Recurrence Techniques for the Calculation of Bessel Functions, MTAC 13 (1959), p. 102.

[2] Allen, E.E.: Polynomial Approximations to some modified Bessel Functions, MTAC Vol. 10, 1956, p. 162-164.

[3] Zachariassen, J.: Bessel I and K, Algol procedure, Regnecentralen, April 1964, GSL O.No. 179.

[4] British Association Mathematical Tables, Vol. VI, Bessel Functions, zero and unity, Cambridge University Press, 1958.

[5] British Association Mathematical Tables, Vol. X Bessel Functions, order 2 to 20, Cambridge University Press, 1952.

6. Algorithm.

```
besselik=set 3
besselik=algol
external

  procedure besselik (n,x,I,K);
  value n,x; real x; integer n;
  array I,K;

  begin integer i,nb,m;
  real a,j0,j1,j2,sum,xhalf;
    m:=n;
    nb:=x+14;
    if nb<=n+3 then nb:=n+4;
    xhalf:=x/2;
    if x<=⏨-150 then nb:=0
    else
    begin
      j1:=1;
      i:=0;
      for i:=i+1 while j1>⏨-150∧i<=nb do j1:=j1×xhalf/i;
      nb:=i-1
    end;
    comment nb is the upper bound for recurrence;
    if nb<=n then
    begin
      for i:=nb+1 step 1 until n do I(i):=0;
      m:=nb
    end;
    sum:=j2:=0;
    for i:=nb step -1 until 1 do
    begin
      if i<=m then I(i):=j1;
      j0:=i/xhalf×j1 + j2;
      sum:=sum+j0;
      j2:=j1;  j1:=j0
    end recurrence loop;
```

```
    sum:=exp(x)/(2×sum-j0);
    a:=I(0):=j0×sum;
    j2:=j2×sum; if n>0 then I(1):=j2;
    for i:=m step -1 until 2 do I(i):=I(i)×sum;
    comment all values of I0(x),...,In(x) are calculated;
    if xhalf<1 then
    begin
      j0:=xhalf××2;
      j1:=ln(x)-.693147181;
      a:=K(0):=(((((  .00000740 ×j0 + .00010750)×j0
                    + .00262698)×j0 + .03488590)×j0
                    + .23069756)×j0 + .42278420)×j0
                    -.57721566      - j1×a;
      if n>0 then
      j2:=K(1):=((((((-.00004686 ×j0 - .00110404)×j0
                    -.01919402)×j0 - .18156897)×j0
                    -.67278579)×j0 + .15443144)×j0
                    + 1         )/x + j1×j2
    end
    else
    begin
      j0:=1/xhalf;
      j1:=sqrt(x)×exp(x);
      a:=K(0):=(((((( .00053208 ×j0 - .00251540)×j0
                   +.00587872)×j0 - .01062446)×j0
                   +.02189568)×j0 - .07832358)×j0
                  +1.25331414)/j1;
      if n>0 then
      j2:=K(1):=((((((-.00068245 ×j0 + .00325614)×j0
                    -.00780353)×j0 + .01504268)×j0
                    -.03655620)×j0 + .23498619)×j0
                   +1.25331414)/j1
    end calculating K0(x) and K1(x) by
        polynomial approximation;
    for i:=2 step 1 until n do
    begin
      sum:=K(i):=a+(i-1)/xhalf×j2;
      a:=j2; j2:=sum
    end recurrence loop
end besselik;
```

comment
    besselik calculates the modified bessel functions:
I0(x),...,In(x) and K0(x),...,Kn(x).

Call parameters:

- n : (real or integer)
  maximum order of the Bessel functions.
  n must be >=0.
- x : (real or integer) the argument, must be > 0.

Return parameters:

- I : (real array I(0:n))
  the values of the calculated functions:
  I(0)=I0(x),...,I(n)=In(x).
- K : (real array K(0:n))
  the values of the calculated functions:
  K(0)=K0(x),...,K(n)=Kn(x);