ALGORITHM 50
INVERSE OF A FINITE SEGMENT OF THE HILBERT MATRIX

JOHN R. HERNDON
Stanford Research Institute, Menlo Park, California

```
procedure  INVHILBERT (n,S);  value n;  real n;
           real array S;
comment  This procedure computes the elements of the inverse
  of an n × n finite segment of the Hilbert matrix and stores them
  in the array S;
begin      real i, j, k;
           S[1, 1] = n × n;
           for i := 2 step 1 until n do
             begin
               S[i, i] := (n+i−1) × (n−i+1)/((i−1) × (i−1));
               S[i, i] := S[i−1, i−1] × S[i, i] × S[i, i]
             end;
           for i := 1 step 1 until n−1 do
             begin
               for j := i+1 step 1 until n do
                 begin
                   k := j−1;
                   S[i, j] := −S[i, k] × (n+k) × (n−k)/(k × k)
                 end
             end;
           for i := 2 step 1 until n do
             begin S[i, i] := S[i, i]/(i+i−1);
               for j := 1 step 1 until i−1 do
             begin S[j, i] := S(j, 1]/(i+j−1);
                   S[i, j] := S[j, i]
             end
           end
end INVHILBERT;
```

CERTIFICATION OF ALGORITHM 50
INVERSE OF A FINITE SEGMENT OF THE HILBERT MATRIX [J. R. Herndon, *Comm. ACM 4* (Apr. 1961)]

B. RANDELL
Atomic Power Division, The English Electric Co., Whetstone, England

INVHILBERT was translated using the DEUCE ALGOL compiler and the following corrections being needed.

1. S[1, 1] = n × n, replaced by S[1, 1] := n × n;
2. S[j, i] := S(j, 1]/(i + j − 1)
replaced by S[j, i] := S[j, i]/(i + j − 1)

The compiled program, which used a 20 bit mantissa floating point notation then produced the following 4 × 4 segment

16.0	−120.0	240.0002	−140.0
−120.0	1200.0	−2700.0	1680.0019
240.0	−2700.0	6480.0	−4200.0
−140.0	1680.0019	−4200.0	2800.0039

REMARKS ON AND CERTIFICATION OF ALGORTHM 50
INVERSE OF A FINITE SEGMENT OF THE HILBERT MATRIX [J. R. Herndon, *Comm. ACM*, Apr. 1961]

P. NAUR
Regnecentralen, Copenhagen, Denmark

In addition to inserting the corrections indicated by B. Randell [*Comm. ACM 5* (Jan. 1962), 50], we have modified and simplified the algorithm as follows:

1. The types of n, i, j and k have been changed to **integer.** This saves roundoff operations in subscripts.
2. Explicit multiplications have been replaced by squaring. This saves code length and execution time, at least in a compiler like ours for the GIER.
3. Repeated references to subscripted variables have been eliminated, partly with the aid of an additional simple working variable, w, partly by using simultaneous assignments.
4. An unnecessary **begin end** pair has been removed.

In total, these changes, in addition to reducing the code length, have increased speed by a factor of 1.6.

The resulting algorithm is as follows:

```
procedure INVHILBERT(n,S);
value n;  integer n;  real array S;
comment  ALG. 50: This procedure computes the elements of
  the inverse of an n × n finite segment of the Hilbert matrix and
  stores them in the array S. The Hilbert matrix has the elements
  HILBERT[i,j] = 1/(i+j−1). The segments of this are known
  to be increasingly ill-conditioned with increasing size;
begin integer i, j, k;  real w;
w := S[1,1] := n↑2;
for i := 2 step 1 until n do w := S[i,i] := w × ((n+i−1) ×
    (n−i+1)/(i−1)↑2)↑2;
for i := 1 step 1 until n−1 do for j := i+1 step 1 until n do
  begin
  k := j−1;
  S[i,j] := −S[i,k] × (n+k) × (n−k)/k↑2
  end;
for i := 2 step 1 until n do for j := 1 step 1 until i do
  S[i,j] := S[j,i] := S[j,i]/(i+j−1)
end INVHILBERT;
```

Both the original version and the above improved one have been run successfully on the GIER ALGOL system (30-bit mantissa). The test program included:

(a) Output of the 4 × 4 matrix, to be compared with the results of Randell [loc. cit.]. Results:

16.000000	−120.000000	240.000000	−140.000000
−120.000000	1200.000000	−2700.000000	1680.000000
240.000000	−2700.000000	6480.000000	−4200.000000
−140.000000	1680.000000	−4200.000000	2799.999977

(b) For n := 1 **step** 1 **until** 15, the inverse of the segment was calculated by INVHILBERT and multiplied by the segment of the Hilbert matrix, and the result was compared with the unit matrix. The maximum error was divided by the largest element of the inverse to form a relative error. Some of the results, which were entirely satisfactory throughout, are given below:

Order	*Element of max error*	*abs (max error)*	*Largest element of INVHILBERT*	*Relative error*
3	$S[3,3]$	$2.38_{10}-7$	$1.92_{10}2$	$1.24_{10}-9$
6	$S[2,4]$	$4.39_{10}-3$	$4.41_{10}6$	$9.96_{10}-10$
9	$S[2,8]$	$1.24_{10}2$	$1.22_{10}11$	$1.01_{10}-9$
12	$S[5,9]$	$1.54_{10}6$	$3.66_{10}15$	$4.21_{10}-10$
15	$S[1,12]$	$1.06_{10}11$	$1.15_{10}20$	$9.22_{10}-10$

(c) The time for a call of the revised INVHILBERT was found as follows:

n	
5	0.2 seconds
10	0.6 "
15	1.3 "