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 x n finite segment of the Hilbert matrix and stores them
in the array S;
begin
real i, j, k;
S[1, 1] := n x n;
for i := 1 step 1 until n do
begin
S[i, i] := (n+i-1) x (n+i-1)/(i-1) x (i-1));
S[i, j] := S[i-1, i-1] x S[i, i] x S[i, j]
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] x (n+k) x (n-k)/(k x k)
end
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 x n, replaced by S[1, 1] := n x n;
2. S[j, j] := S[j, j]/(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 x 4 segment

\[
\begin{array}{cccc}
16 & -120 & 240.0002 & -140.0 \\
-120 & 1200 & -2700.0 & 1680.0019 \\
240 & -2700.0 & 6480.0 & -4200.0 \\
-140.0 & 1680.0019 & -4200.0 & 2900.0039 \\
\end{array}
\]

REMARKS ON AND CERTIFICATION OF ALGORITHM 50
P. NAUR
Regnecentralen, Copenhagen, Denmark

In addition to inserting the corrections indicated by B. Randell
[Comm. ACM 6 (Jan. 1963), 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 Explicit multiplications have been replaced by squaring.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 x 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) x
(n+i-1))/(i-1+1)/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] x (n+k) x (n-k)/k/2
end;
for i := 2 step 1 until i do for j := 1 step 1 until i do
S[i,j] := S[i,j]/(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 x 4 matrix, to be compared with the results
of Randell [loc. cit.]. Results:

\[
\begin{array}{cccc}
-16.00000 & -120.00000 & 240.00000 & -140.00000 \\
-120.00000 & 1200.00000 & -2700.00000 & 1680.0019 \\
240.00000 & -2700.00000 & 6480.00000 & -4200.00000 \\
-140.00000 & 1680.0019 & -4200.00000 & 2900.0039 \\
\end{array}
\]

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

\[
\begin{array}{cccc}
\text{n} & \text{time} \\
5 & 0.2 \text{ seconds} \\
10 & 0.6 \text{ "} \\
15 & 1.3 \text{ "}
\end{array}
\]