ALGORITHM 42
INVERT
T. C. WOOD
RCA Digital Computation and Simulation Group,
Moorestown, New Jersey

procedure Invert (A) order: (n) Singular: (s) Inverse: (A1):
array A, A1; integer n, s, value n;

begin
    comment augment matrix A with identity matrix;
    array a[1:n, 1:2 x n]; integer i, j;
    for i := 1 step 1 until n do
        for j := 1 step 1 until 2 x n do
            if j ≤ n then a[i, j] := A[i, j] else
                if i = n + 1 then a[i, j] := 1.0 else a[i, j] := 0.0;
            comment begin inversion;
    for i := 1 step 1 until n do
        begin
            integer k, l, ind; j := l := i; ind := s := 0;
            L1: if a[l, j] = 0 then
                begin
                    ind := 1; if l < n then begin l := l + 1; go to L1 end
                    else begin s := 1; go to L2 end;
                end;
            if ind = 1 then for k := 1 step 1 until 2 x n do
                begin
                    temp := a[l, k];
                    a[l, k] := a[i, k];
                    a[i, k] := temp;
                end;
            for k := j step 1 until 2 x n do
                begin
                    a[i, k] := a[i, k] / a[i, j];
                end;
            for l := 1 step 1 until n do
                if l ≠ i then for k := 1 step 1 until 2 x n do
                    begin
                        a[l, k] := a[l, k] - a[i, k] x a[l, j];
                    end;
            end loop;
            for j := 1 step 1 until n do
                begin
                    A1[i, j] := a[i, n + j];
                end;
        end;
    end;

(b) for k := j step 1 until 2 x n do
    a[i, k] := a[i, k] / a[i, j];
    should be
    for k := 2 x n step -1 until i do
        a[i, k] := a[i, k] / a[i, i];

(c) if i ≠ i then for k := 1 step 1 until 2 x n do
    a[i, k] := a[i, k] - a[i, k] x a[l, i];
    should be
    if i ≠ i then for k := 2 x n step -1 until i do
        a[i, k] := a[i, k] - a[i, k] x a[l, i];

Given these changes, j becomes superfluous in the second i loop,
and the other references to j may be changed to references to i.

INVERT obtained the following results:
The computer inverted a 17-by-17 matrix whose elements were
integers less than ten in absolute value. When the matrix and its
inverse were multiplied together, the largest nondiagonal element
in the product was ~.00003. Most nondiagonal elements were less
than .00001 in absolute value.

INVERT was tested using finite segments of the Hilbert matrix.
The following results were obtained in the 4 x 4 case:

<table>
<thead>
<tr>
<th></th>
<th>0.10</th>
<th>-1.20</th>
<th>2.40</th>
<th>-3.60</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>1.20</td>
<td>-2.40</td>
<td>3.60</td>
<td>-4.80</td>
</tr>
<tr>
<td>-1.20</td>
<td>2.40</td>
<td>-3.60</td>
<td>4.80</td>
<td>-6.00</td>
</tr>
<tr>
<td>2.40</td>
<td>-3.60</td>
<td>4.80</td>
<td>-6.00</td>
<td>7.20</td>
</tr>
<tr>
<td>-3.60</td>
<td>4.80</td>
<td>-6.00</td>
<td>7.20</td>
<td>-8.40</td>
</tr>
</tbody>
</table>

The exact inverse is:

<table>
<thead>
<tr>
<th></th>
<th>0.10</th>
<th>-1.20</th>
<th>2.40</th>
<th>-3.60</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>1.20</td>
<td>-2.40</td>
<td>3.60</td>
<td>-4.80</td>
</tr>
<tr>
<td>-1.20</td>
<td>2.40</td>
<td>-3.60</td>
<td>4.80</td>
<td>-6.00</td>
</tr>
<tr>
<td>2.40</td>
<td>-3.60</td>
<td>4.80</td>
<td>-6.00</td>
<td>7.20</td>
</tr>
<tr>
<td>-3.60</td>
<td>4.80</td>
<td>-6.00</td>
<td>7.20</td>
<td>-8.40</td>
</tr>
</tbody>
</table>

INVERT was also coded for the LGP-30 in machine language
and the 24.1 extended range interpretive system. This system,
which uses 30 significant bits for the fraction, obtained the follow-
ing as the inverse of the 4 x 4 Hilbert matrix:

<table>
<thead>
<tr>
<th></th>
<th>0.00</th>
<th>-1.00</th>
<th>2.00</th>
<th>-3.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>1.00</td>
<td>-2.00</td>
<td>3.00</td>
<td>-4.00</td>
</tr>
<tr>
<td>-1.00</td>
<td>2.00</td>
<td>-3.00</td>
<td>4.00</td>
<td>-5.00</td>
</tr>
<tr>
<td>2.00</td>
<td>-3.00</td>
<td>4.00</td>
<td>-5.00</td>
<td>6.00</td>
</tr>
<tr>
<td>-3.00</td>
<td>4.00</td>
<td>-5.00</td>
<td>6.00</td>
<td>-7.00</td>
</tr>
</tbody>
</table>

The program coded in the 24.0 interpretive system successfully
inverted a matrix consisting of ones on the minor diagonal and
zeros everywhere else.

REMARKS ON ALGORITHM 42

P. NAUR
Regnegentralen, Copenhagen, Denmark

INVERT cannot be recommended since it does not search for
pivot and therefore will give poor accuracy. This is confirmed by
the figures quoted by Knapp and Shaman in their certification
[Comm. ACM 4 (Nov. 1961), 408]. The results obtained by them
using 30 significant bits for the fraction may be compared directly
with those obtained using INVERSION II (Algorithm 120) and
gir with the GIER Algol system (see certification below). Inver-
ting the 4 x 4 segment of the Hilbert matrix, the largest error
in any element is found to be:

CERTIFICATION OF ALGORITHM 42
INVERT (T. C. Wood, Comm. ACM, Apr. 1961)

ANTHONY W. KNAPP AND PAUL SHAMAN
Dartmouth College, Hanover, N. H.

INVERT was hand-coded for the LGP-30 using machine lan-
guage and the 24.0 floating-point interpretive system, which car-
ries 24 bits of significance for the fractional part of a number and
five bits for the exponent. The following changes were found
necessary:

(a) if j = n + 1 then a[i, j] := 1.0 else a[i, j] := 0.0;
    should be
    if j = n + 1 then a[i, j] := 1.0 else a[i, j] := 0.0;
In view of this basic shortcoming of Algorithm 42, it is unnecessary to report on other features of it.

CORRECTION TO EARLIER REMARKS ON ALGORITHM 42 INVERT, ALG. 107 GAUSS'S METHOD, ALG. 120 INVERSION II, AND gjr [P. Naur, Comm. ACM, Jan. 1963, 38–40.]

P. NAUR
Regencentralen, Copenhagen, Denmark

George Forsythe, Stanford University, in a private communication has informed me of two major weaknesses in my remarks on the above algorithms:

1) The computed inverses of rounded Hilbert matrices are compared with the exact inverses of unrounded Hilbert matrices, instead of with very accurate inverses of the rounded Hilbert matrices.

2) In criticizing matrix inversion procedures for not searching for pivot, the errors in inverting positive definite matrices cannot be used since pivot searching seems to make little difference with such matrices.

It is therefore clear that although the figures quoted in the earlier certification are correct as they stand, they do not substantiate the claims I have made for them.

To obtain a more valid criterion, without going into the considerable trouble of obtaining the very accurate inverses of the rounded Hilbert matrices, I have multiplied the calculated inverses by the original rounded matrices and compared the results with the unit matrix. The largest deviation was found as follows:

<table>
<thead>
<tr>
<th>Order</th>
<th>INVERSION II</th>
<th>gjr</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-1.4910−8</td>
<td>-1.4910−8</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>-4.7710−7</td>
<td>-8.3410−7</td>
<td>0.57</td>
</tr>
<tr>
<td>4</td>
<td>-9.5410−6</td>
<td>-3.4310−5</td>
<td>0.28</td>
</tr>
<tr>
<td>5</td>
<td>-7.3210−4</td>
<td>-4.5810−4</td>
<td>1.6</td>
</tr>
<tr>
<td>6</td>
<td>-1.6110−2</td>
<td>-1.4210−2</td>
<td>1.1</td>
</tr>
<tr>
<td>7</td>
<td>-5.7810−1</td>
<td>-5.4710−1</td>
<td>1.1</td>
</tr>
<tr>
<td>8</td>
<td>-1.2010−2</td>
<td>-1.3810−1</td>
<td>8.7</td>
</tr>
<tr>
<td>9</td>
<td>-4.9110−1</td>
<td>-2.2210−1</td>
<td>2.2</td>
</tr>
</tbody>
</table>

This criterion supports Forsythe's criticism. In fact, on the basis of this criterion no preference of INVERSION II or gjr can be made.

The calculations were made in the GIERS ALGOL system, which has floating numbers of 20 significant bits.