ALGORITHM 43
CROUT WITH PIVOTING II
HENRY C. THACHER, JR.*
Argonne National Laboratory, Argonne, Illinois

real procedure INNERPRODUCT (u,v) index : (k) start : (s)
finish : (f);
value s, f; integer k, s, f; real u, v;
comment INNERPRODUCT forms the sum of u(k) x v(k) for
k = s, s+1, . . . , f. If k > f, the value of INNERPRODUCT is
zero. The substitution of a very accurate inner product procedure
would make CROUT more accurate;
comment INNERPRODUCT may be declared in the head of any
block which includes the block in which CROUT is de-
clared. It may be used independently for forming the inner
product of vectors;
begin
    real h;
    h := 0; for k := s step 1 until f do h := h + u x v;
    INNERPRODUCT := h
end INNERPRODUCT;

procedure CROUT II (A, b, n, y, pivot, det, repeat)
comment This procedure is a revision of Algorithm 16, CROUT
With Pivoting by George E. Forsythe, Comm. ACM 3, (1960)
507-8. In addition to modifications to improve the running
of the program, and to conform to proper usage, it provides for
the computation of the determinant, det, of the matrix A. The
solution is obtained by CROUT's method with row interchanges,
as formulated in reference [1], for solving Ay = b and transform-
ing the augmented matrix [A b] into its triangular decomposi-
tion LU with all L(k,k) = 1. If A is singular we exit to 'singular,'
a nonlocal label; pivot (k) becomes the current row index of the
pivot element in the k-th column. Thus enough information
is preserved for the procedure to process a new right-hand
side without repeating the triangularization, if the boolean pa-
rameter repeat is true. The accuracy obtainable from CROUT
would be much increased by calling CROUT with a more accu-
rate inner product procedure than INNERPRODUCT.

The contributions of Michael F. Lipp and George E. Forsythe
by prepublisher review and pointing out several errors are
gratefully acknowledged;
comment Nonlocal identifiers appearing in this procedure are:
(1) The nonlocal label 'singular', to which the procedure exits
if det A=0, and (2) the real procedure 'INNERPRODUCT'
given above;
begin
    integer i, j, imax, p; real det; Boolean repeat;

begin
    integer k, i, j, imax, p; real TEMP, det;
    det := 1; if repeat then go to 6;
    for k := s step 1 until n do

begin
    TEMP := 0;
    for i := k step 1 until n do

begin
p, 1, k-1);
    if abs(A[i,k]) > TEMP then
begin
    TEMP := abs(A[i, k]); imax := i
end 3
end 2;
pivot [k] := imax;
comment We have found that A[imax, k] is the largest pivot in
column k. Now we interchange rows k and imax;
if imax /= k then
begin
    det := - det; for j := 1 step 1 until n do
begin
end 5;
TEMP := b[k]; b[k] := b[imax]; b[imax] := TEMP
end 4;
comment The row interchange is done. We proceed to the
elimination;
if A[k,k] = 0 then go to singular;
quot := 1.0/A[k,k];
for i := k+1 step 1 until n do
Ai,k] := quot * A[i,k];
for j := k+1 step 1 until n do
A[p,j], p, 1, k-1);
end 1; go to 7;
comment The triangular decomposition is now finished,
and we skip to the back substitution;
begin
comment This section is used when the formal parameter repeat is true,
indicating that the matrix A has previously been decomposed into a triangular
form by CROUT II, with row interchanges specified by pivot,
and that it is desired to solve the linear system with a
new vector b, without repeating the triangularization;
for k := s step 1 until n do
begin
    TEMP := b[pivot[k]]; b[pivot[k]] := b[k]; b[k] :=
TEMP;
    b[k] := b[k] - INNERPRODUCT (A[k,p],
b[p], p, 1, k-1)
end 6;
for k := n step -1 until 1 do
begin
    y[k] := (b[k] - INNERPRODUCT (A[k,p], y[p], p,
k+1, n)/A[k,k]
end 8;
end CROUT II;

REFERENCE:
(1) J. H. Wilkinson, Theory and practice in linear systems. In
John W. Carr III (editor), Application of Advanced Nu-
merical Analysis to Digital Computers, pp. 43-100 (Lectures
given at the University of Michigan, Summer 1958, College
of Engineering, Engineering Summer Conferences, Ann
Arbor, Michigan [1959]).

* Work supported by the U. S. Atomic Energy Commission.
CERTIFICATION OF ALGORITHM 43
CROUT II (Henry C. Thacher, Jr., Comm. ACM, 1960)
HENRY C. THACHER, JR.*
Argonne National Laboratory, Argonne, Illinois

CROUT II was coded by hand for the Royal Precision LGP-30 computer, using a 28-bit mantissa floating point interpretive system (24.2 modified).

The program was tested against the linear system:

\[
\begin{bmatrix}
12.1719 & 27.3941 & 1.9827 & 7.3757 \\
8.1163 & 23.3385 & 9.8397 & 4.9474 \\
3.0706 & 13.5494 & 15.5973 & 7.5172 \\
\end{bmatrix}
\begin{bmatrix}
6.6359 \\
6.0563 \\
7.4686 \\
6.0580 \\
\end{bmatrix}
= \begin{bmatrix}
6.6355 \\
6.1304 \\
4.6921 \\
2.5393 \\
\end{bmatrix}
\]

with the following results:

\[
\begin{bmatrix}
12.171900 & 27.394100 & 1.982700 & 7.375999 \\
0.25220957 & 6.6327021 & 15.097125 & 5.656532 \\
0.25124262 & -0.56260167 & 14.979620 & 14.527683 \\
0.66680633 & 0.76468965 & -0.20207312 & -1.3606142 \\
\end{bmatrix}
\begin{bmatrix}
6.6354999 \\
3.0818633 \\
2.5702026 \\
-0.082780734 \\
\end{bmatrix}
\begin{bmatrix}
pivot = 1 & 3 & 4 & \end{bmatrix}
= \begin{bmatrix}
1.1599210 \\
0.1469177 \\
0.1125748 \\
0.060840712 \\
\end{bmatrix}
\]

\[\det = -1645.4499\]

All elements of \(Ab-y\) were less than \(10^{-7}\) in magnitude. Identical results were obtained with the same \(b\), and repeat true. With the same \(b\) and the last row vector of \(A\) replaced by \(\langle 19.1927, 33.4409, 25.1288, 5.2811 \rangle\), i.e., \(A, \vec{j} = A 1 j\), and 2A 2, \(j = 3\)A 3, \(j\), the results were:

\[\det = 0.10924352 \times 10^4\]

\[\begin{bmatrix}
0.29214425 \times 10^6, -0.12531172 \times 10^6, 0.72411923 \times 10^7, -0.50183892 \times 10^7 \\
\end{bmatrix}\]

Failure to recognize this singular matrix is due to roundoff, either in the data input or in the calculation.

* Work supported by the U.S. Atomic Energy Commission.

CERTIFICATION OF ALGORITHM 43
CROUT II [Henry Thacher, Jr., Comm. ACM (1960), 176]
C. DOMINGO AND F. RODRIGUEZ-GIL
Universidad Central, Caracas, Venezuela

CROUT II was coded in PUC-R2 and tested in the IBM-1620. Two types of INNERPRODUCT subroutines were used. The first one finds the scalar product in fixed-point arithmetic to increase accuracy, using an accumulator of 32 digits. The second one uses ordinary floating-point with eight significative figures.

Using a unit matrix as right-hand side, a 6 \times 6 segment of Hilbert matrix was inverted. The inverse was inverted again.

The maximum difference between this result and the original segment of Hilbert matrix was:

Using fixed-point INNERPRODUCT \ldots \ldots \ldots \ldots 8.2426 \times 10^{-4}
(Value of determinant) \ldots \ldots \ldots \ldots 4.7737088 \times 10^{-18}

Using floating-point INNERPRODUCT \ldots \ldots \ldots \ldots 3.014016 \times 10^{-2}
(Value of determinant) \ldots \ldots \ldots \ldots 4.4950721 \times 10^{-18}

Two typographical errors were observed in the algorithm: