The Complete Pivoting Conjecture for Gaussian Elimination is False

*Alan Edelman**
Lawrence Berkeley Laboratory
& Department of Mathematics
University of California
Berkeley, California 94720
edelman@math.berkeley.edu

March 27, 1992

Abstract

A famous conjecture concerning Gaussian Elimination was recently "settled" as false, by a counterexample found on a Cray supercomputer. Mathematica did not yield the same conclusion when given identical data, reminding us of the care needed when proving mathematical statements using rounded arithmetic. Indeed, the conjecture is false, but a proper counterexample requires modifications of the data. In this note, we provide proper counterexamples by modifying numbers computed in rounded arithmetic by Nick Gould on a Cray.

1 Introduction

Gaussian elimination is the most basic numerical method for solving a dense linear system of equations $Ax = b$. There are many variations on how to organize the computations, but taken as a whole Gaussian elimination is probably one of the most widely known numerical algorithms. For decades, scientists have solved problems of ever increasing size using Gaussian elimination. By last year, the largest matrix solved was of size 55,000, and surely

*Supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy under Contract DE-AC03-76SF00098.

a matrix of size 100,000 will undergo Gaussian elimination very soon, if it has not already.

The algorithm may be old, but new and unanswered questions continue. Some relate to the practical details of implementing the algorithm on new and ever changing architectures. Others concern whether a different algorithm might be more suitable. This article focuses on a theoretical mystery associated with Gaussian elimination: the complete pivoting conjecture for the growth factor.

Associated with any matrix A is a *growth factor* $g(A)$ which describes the growth of matrix elements when A undergoes Gaussian elimination with complete pivoting. The conjecture states that $g(A) \leq n$ for an $n \times n$ matrix A. In the next two sections we explain this conjecture and present a Mathematica program to calculate the growth factor.

This article also focuses on the difference between exact and floating point arithmetic calculations. This distinction is not made often and clearly enough. Putting aside philosophical issues of whether or not one should trust a computer for mathematical proofs, one can not too hastily make inferences about exact arithmetic from rounded computations. Another step is needed: the justification of the approximation or a check in exact arithmetic.

Recently, Nick Gould reported on a counterexample to the complete pivoting conjecture [Gould 1991a]. He presented a 13×13 matrix "for which the growth is 13.0205", and said that "growth larger than n has also been observed for matrices of orders 14, 15, and 16". Gould found his matrix using floating point arithmetic on a Cray supercomputer.

To verify the results reported by Gould, I worked with two students, Miles Ohlrich and Su-Lin Wu, duplicating Gould's calculations in exact arithmetic with programs written in Mathematica and Maple.

Imagine our surprise when we observed a growth factor of under 7.34 for the matrix that was supposed to give growth of 13.0205! Initial attempts by one of the students failed to find a perturbation of Gould's matrix that would give large growth, and hence we began to wonder if the conjecture was indeed false. After all, the growth factor is only a piecewise continuous function of the matrix, and hence a small rounding could greatly change the result. Here we report that the conjecture is indeed false, and Gould's example can be modified in a small way so as to give a true counterexample.

2 Gaussian Elimination

In its simplest form, Gaussian elimination factors a matrix A into $L \times U$ where L is a lower triangular matrix with unit diagonal and U is upper triangular. Here is a 3×3 example.

$$\begin{pmatrix} 2 & 1 & 3 \\ 6 & 7 & 10 \\ -4 & 6 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ -2 & 2 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 3 \\ 0 & 4 & 1 \\ 0 & 0 & 8 \end{pmatrix}$$

Once A is in the form $L \times U$ it requires much less computational effort to solve first $Ly = b$ and then $Ux = y$ to get the solution x to $Ax = b$.

The matrix L is not needed for our purposes. U can be found by repeated *row operations*, adding multiples of one row to another to eliminate the nonzero entries below the diagonal. Algorithms for Gaussian elimination appear in many standard references, such as [Golub and van Loan 1989; Press et. al. 1986].[1] The "no-frills" method of computing U can be expressed in Mathematica as

```
NoFrills[ A_?MatrixQ ] :=
    Module[{a=A, U={}},
        Do[
            U = Append[U, a[[1]] ];
            r = Range[2,k];
            a = a[[r,r]] - Outer[Times, a[[r,1]] ,a[[1,r]] ]/a[[1,1]],
            {k, Length[A], 2, -1}
        ] ;
        U = Append[U, a[[1]] ]
    ]
```

This implementation erases a row and column of A after each pass through the loop and only stores the upper triangular part of U. The lower $(k-1) \times (k-1)$ part of the matrix is updated by the addition of a scaled outer product that is formed from the first column, the first row, and the upper left element as the scaling factor.

If the upper left entry is ever zero, the no-frills approach breaks in exact arithmetic. In finite precision arithmetic, the no-frills approach is numerically unstable, that is, roundoff errors tend to make the result unreliable. There are two fixes to this problem, *partial pivoting* and *complete pivoting*. In partial pivoting, a row interchange occurs to ensure that the upper left entry, the *pivot*, is the largest element (in magnitude) in the column. In complete pivoting, a row and column interchange occurs making the pivot the largest element in the submatrix. Partial pivoting is most common in applications. Complete pivoting is rarely used, because the improvement in numerical stability over partial pivoting does not justify the time spent searching for the largest element in the submatrix. Only in certain special cases can pivoting be avoided altogether.

Using Mathematica, pivoting can be implemented by defining functions to interchange (switch) rows or columns.

```
Attributes[RowSwitch]=HoldAll
Attributes[ColSwitch]=HoldAll
RowSwitch[m_,n_,a_] :=
   {a[[m]],a[[n]]}={a[[n]],a[[m]]}
ColSwitch[m_,n_,a_] :=
   (a=Transpose[a]; {a[[m]],a[[n]]}={a[[n]],a[[m]]}; a=Transpose[a])
```

[1] However, [Press et. al 1986] devotes undo attention to the Gauss-Jordan algorithm which is of little importance as a numerical recipe.

The row interchange of partial pivoting is then obtained by inserting the following steps into the loop:

```
m = First[Position[Abs[a], Max[Abs[#[[1]]]& /@ a]]];
RowSwitch[1,m[[1]],a]
```

Complete pivoting is given by a row interchange followed by a column interchange:

```
m = First[Position[Abs[a], Max[Abs[a]]]];
RowSwitch[1,m[[1]],a]
ColSwitch[1,m[[2]],a]
```

With either form of pivoting, the pivots will be the diagonal elements of the resulting upper triangular matrix U.

3 Growth Factors

The quantity that we wish to study is the *growth factor* of an $n \times n$ matrix A under complete pivoting, defined as

$$g_n(A) = \frac{\max_{i,j,k} |a_{ij}^{(k)}|}{\max_{i,j} |a_{ij}|},$$

where $a_{ij}^{(k)}$ is a matrix element at the k-th step of the elimination process. From the definition of complete pivoting it follows that the largest element at each step will be one of the pivots, so the growth factor can also be defined as

$$g_n(A) = \frac{\max_i |u_{ii}|}{\max_{i,j} |a_{ij}|}.$$

In the standard error analysis of Gaussian elimination, it is shown that the backward error (a measure of stability) in the numerical solution to $Ax = b$ is bounded by

$$8n^3 g_n(A)\mathrm{u},$$

where u denotes the "unit-roundoff" and the n^3 term is considered pessimistic in practice. Analysis of forward error (another measure of stability) also involves the growth factor. (See any textbook on numerical linear algebra for an explanation of the growth factor and error analysis of Gaussian elimination.)

It is natural to ask how large the growth factor can be for $n \times n$ matrices. Nobody has been able to answer this question. The only known bound is due to Wilkinson [Wilkinson 1961], who showed that

$$g_n(A) \leq n^{1/2}(2 \cdot 3^{1/2} \cdots n^{1/(n-1)})^{1/2}.$$

For $n = 100$, this bound is roughly 3500; however, nobody has ever observed growth bigger than 100 for a 100×100 matrix. Wilkinson observed that it was difficult to construct a matrix for which $g_n(A) > n$. Cryer published the statement which has become known as Wilkinson's conjecture[2] [Cryer 1968]:

Conjecture: If Gaussian elimination with complete pivoting is performed on a matrix A, then $g_n(A) \leq n$.

The recent claim by Gould that he found a 13×13 matrix with growth 13.0205 is the first published "counterexample" to the conjecture [Gould 1991a]. We will show that Gould's floating point calculation is not quite correct, in that it does not give the same result in exact arithmetic. We will demonstrate rigorously that the conjecture is indeed false by modifying Gould's counterexample ever so slightly. To do this we need a Mathematica program to calculate the absolute pivots $|u_{ii}|$. Here is such a program:

```
Options[Pivots] = {Pivoting -> True}

Pivots[A_, opt___Rule] :=
    Module[{a=A, m, p={}, piv},
        piv = Pivoting /. {opt} /. Options[Pivots];
        Do[
            m = First[Position[Abs[a],Max[Abs[a]]]];
            If[ piv,
                RowSwitch[1,m[[1]],a];
                ColSwitch[1,m[[2]],a]];
            p = Append[p, {m+Length[A]-k,N[Abs[a[[1,1]]],40]}];
            r = Range[2,k];
            a = a[[r,r]] - Outer[
                Times, a[[r,1]] ,a[[1,r]] ]/a[[1,1]],
            {k, Length[A], 1, -1}
        ] ;
     p ]
```

The program returns a list of pivots and the locations of the largest element in magnitude at each step of the Gaussian elimination. The elimination is performed either with no pivoting or with complete pivoting, depending on how the option is set. In Section 5, we describe briefly our modifications to some of Gould's examples. The location of the maximums were essential for finding these modifications.

4 Gould's Floating Point Counterexample

Gould's purported counterexample to the growth conjecture is a 13×13 matrix which we represent in Mathematica as follows:

```
a= {1, -1, -1, 660848918578853640, 350768677240296530, 139130936348087710,
   1, -1, 945463095088536990, -64358761317393848, -47259056539260776,
```

[2] though Wilkinson never published this explicitly as a conjecture

```
    981447528786957180, l, l, l, -l, -l, -882625441488454570,
    -793497892195840220, -l, -700496337540687080, l, l, -l, l,
    -651498589419302720, l, 493218479970826740, l, 523219868894640230, l,
     931478025815019150, -l, -l, -l, 906340171404097510, l, 196359942450215320,
     520200438016106050, -852377236166545040, l, -799595937286409320, l,
    -613950298735988050, -l, -l, l, l, l, l, -l, l, -641979766159483270, l,
    -823477739209516720, -l, l, -l, -l, l, -l, -l, -980475145622109130,
     l, l, -757461144210523130, 876253886818607830, -l, -l,
    -814104693902053870, l, l, -l, -l, l,
    -l, -l, l, l, l, l, 588225298469760790, l, -l, 117806934515049340,
    -l, l, -l, -l, -l, l, l, l, -123654398954411060, -l, -l, l, l, l, l, -l, l,
     l, l, -l, 167280198905618540, -l, -l, l, 670377079454039460, -l,
    -l, l, -l, l, -l, -l, -l, l, 734512344136362240,
     774209922789794840, l, l, l, l, l, l, -l, l,
    -l, -l, -322948030097235110, l, -l, 59471427088948606, -l, l,
    -773051215153670920, l, l, l, l, l, -l, -170078579523277070, l, l,
    -l, l, -l, -l, 918980310122519350, -l, -l, 250493402326499640, l,
     961431109359263460, -l, 724092990184259320, -l, l, l, -l, l, l,
     l, -l, -l, l};

gould = Partition[a/l /. l -> 10^18, 13]
```

We computed `Pivots[gould]` and `Pivots[gould, Pivoting->False]` and found the pivots listed in Table 1. With complete pivoting the matrix yields a growth factor of around 7.355 in exact arithmetic, considerably smaller than the 13+ needed to be a counterexample. It does, however, yield 13.0205 in double precision floating point arithmetic. When we ran the elimination without pivoting, we found that there was a near tie in the sixth pivot. The proper winner of this near tie would not be resolvable by the finite precision arithmetic in the hardware of most computers.

To speed up the computation, we can replace the matrix `A` with `N[A,100]` in the call to `Pivots`. Of course, there is no guarantee that this will give the correct answer in general, but it does for this example.

5 Finite Precision, Exact Arithmetic, and True Counterexamples

We found that a true counterexample could be obtained from Gould's matrix simply by changing the $(11, 10)$ entry from 1 to $1 - 10^{-7}$.

The fix: `gould[[11,10]] = 1 - 10^(-7);`

This small perturbation of the matrix jumps over a discontinuity in the growth factor function, yielding the growth of 13.02, even in exact arithmetic. The fact that we were able to find a counterexample by a small perturbation leads us to consider the

Perturbation Question for the Growth Factor: If $\hat{g}_n(A)$ denotes the growth factor of a matrix computed in finite precision, must there exist a small perturbation E such that $g_n(A + E) = \hat{g}_n(A)$ in exact arithmetic?

Gould has very recently informed me of matrices with observed growth factors below [Gould 1991b]:

n	$\hat{g}_n$
18	20.45
20	24.25
25	32.99

Of course, I quickly tried the Mathematica program on the biggest matrix, and found a growth factor of 9.4. Again, it turned out to be possible, though laborious, to find a perturbation of this 25×25 matrix that gives growth of nearly 32.99 in exact arithmetic. We list in the table below the fix to the 25×25 matrix to give the reader an idea of what changes need to be made. The matrix appears in the electronic supplement.

Entry	Gould's matrix	our fix
10,10	`.99998703567977021`	`.999987035679771`
18,18	`.99997583082741470`	`.999975830827420`
20,20	`.99996637588164239`	`.999966375881650`
21,21	`.99997417725485349`	`.999974177254860`
23,23	`.99995075834718583`	`.999950758347190`

With these fixes Gould's matrix gives a 25×25 matrix with growth factor 32.986341.

We do not elaborate on how we found the fixes, but we invite the reader to find it for himself. Very roughly, the idea is that if two elements are nearly tied, but the "wrong" element is ever so slightly larger in magnitude, exact arithmetic picks a different pivot than does floating point arithmetic. Thus, by knowing the location of the false maximum, which is returned by the function `Pivots`, we can reduce the corresponding element in the original matrix in the hope of forcing the near tie to have the desired outcome. We must, however, be careful not to change the element too much or we run the risk of destroying the delicate structure that gives large growth.

6 Computers and Mathematical Proofs

Ever since the proof of the four color-color conjecture there has been a lively debate over the applicability of computer-aided proofs. Such questions have even appeared in the lay press (see [Kolata 1991] for one recent article). We take the pragmatic view that people will (and even should) use whatever tools are available, provided that such tools can be verified for correctness.

In our case, we were not satisfied with Mathematica's verification of the counterexample so we wrote a program for another symbolic system, Maple. We found that Maple's results for the 13×13 matrix agreed perfectly with those from Mathematica. One might still legitimately philosophize about whether this confirmation constitutes a proof. However, from our pragmatic point of view, for two completely different software systems to give precisely the same answer to the same question is an overwhelming verification of its correctness. While one software system could have a bug, it is almost certainly impossible for two different widely used systems to lead to the same erroneous conclusion on correct programs. At least, I would argue, it is more likely that humans would err.

7 Appendix: Maple program

Here is a Maple program to verify results found with Mathematica:

```
with(linalg): Digits := 100:
# Begin by defining the Gould matrix
l := 10^18:
A := matrix(13, 13,
 [l, -l, -l, 660848918578853640, 350768677240296530, 139130936348087710,
   l, -l, 945463095088536990, -64358761317393848, -47259056539260776,
   981447528786957180, l, l, l, -l, -l, -882625441488454570,
   -793497892195840220, -l, -700496337540687080, l, l, -l, l,
   -651498589419302720, l, 493218479970826740, l, 523219868894640230, l,
    931478025815019150, -l, -l, -l, 906340171404097510, l, 196359942450215320,
    520200438016106050, -852377236166545040, l, -799595937286409320, l,
   -613950298735988050, -l, -l, l, l, l, l, -l, l, -641979766159483270, l,
   -823477739209516720, -l, l, -l, -l, l, -l, -l, -980475145622109130,
    l, l, -757461144210523130, 876253886818607830, -l, -l,
   -814104693902053870, l, l, -l, -l, l,
   -l, -l, l, l, l, l, 588225298469760790, l, -l, 117806934515049340,
   -l, l, -l, -l, -l, l, l, l, -123654398954411060, -l, -l, l, l, l, l, -l, l,
    l, l, -l, 167280198905618540, -l, -l, l, 670377079454039460, -l,
   -l, l, -l, l, -l, -l, -l, l, 734512344136362240,
    774209922789794840, l, l, l, l, l, l, -l, l,
   -l, -l, -322948030097235110, l, -l, 59471427088948606, -l, l,
   -773051215153670920, l, l, l, l, l, -l, -170078579523277070, l, l,
   -l, l, -l, -l, 918980310122519350, -l, -l, 250493402326499640, l,
    961431109359263460, -l, 724092990184259320, -l, l, l, -l, l, l,
    l, -l, -l, l]):
A := evalm(A/l):

# Choice 1 -- The line below if uncommented will give a matrix with large
# growth in exact arithmetic.
 A[11,10]:=1-10^(-7):

# Define a maxindex function
maxindex := proc(A, i0, j0)
   local  m;
   m := abs(A[1,1]);i0:=1;j0:=1;
   for i to n do for j to n do
          if abs(A[i,j]) > m then m := abs(A[i,j]); i0:=i; j0:=j; fi;
    od;od;
end:

# The basic elimination step
elim := proc(A)
  local i,j;
# Choice 2--Perform Pivoting (comment out the next three lines for no pivoting)
  maxindex(A, i, j);
```

```
  A := swaprow(A, 1, i);
  A := swapcol(A, 1, j);
  D := submatrix(A, 2..n, 1..1) &* submatrix(A, 1..1, 2..n);
  print(convert(A[1,1],float));
  A := evalm(submatrix(A, 2..n, 2..n) - D/A[1,1]);
end:

# Main program
for n from rowdim(A) by -1 to 2  do
    elim(A);
  od:
convert(A[1,1],float);
```

References

[1] A.M.Cohen, A note on pivot size in Gaussian elimination, *Lin. Alg. Appl. 8* (1974), 361–368.

[2] C.W.Cryer, Pivot size in Gaussian elimination, *Num. Math. 12* (1968), 335–345.

[3] J.Day and B.Peterson, Growth in Gaussian elimination, *Amer. Math. Monthly* (June 1988), 489–513.

[4] A.Edelman, Note to the editor, *SIAM J. Matrix Anal. Appl., 12* (1991).

[5] G.H. Golub and C.F. van Loan, *Matrix Computations, Second Edition*, Johns Hopkins University Press, Baltimore, 1989.

[6] N.Gould, On growth in Gaussian elimination with complete pivoting, *SIAM J. Matrix Anal. Appl., 12* (1991), 354–361.

[7] N.Gould, private communication, 1991.

[8] M.Hall, *Combinatorial Theory, 2nd ed.*, John Wiley, New York, 1986.

[9] N.J.Higham and D.J.Higham, Large growth factors in Gaussian elimination with pivoting, *SIAM J. Matrix Anal. Appl., 10* (1989), 155–164.

[10] N.Higham and N.Trefethen, Complete pivoting conjecture is disproved, *SIAM News 24* (January 1991), 9.

[11] G.Kolata, Computers still can't do beautiful mathematics, *New York Times*, (July 14, 1991), Section 4, p. 4.

[12] W.H.Press, B.P.Flannery, S.A.Teukolsky, and W.T.Vetterling, *Numerical Recipes*, Cambridge University Press, 1986.

[13] L.Tornheim, Pivot size in Gauss reduction, Tech Paper, Chevron Research Co., Richmond CA., 1964.

[14] L.Tornheim, Maximum pivot size in Gaussian elimination with complete pivoting, Tech Report, Chevron Research Co., Richmond CA., 1970.

[15] J.H.Wilkinson, Error analysis of direct methods of matrix inversion, *J. Assoc. Comp. Machinery 8* (1961), 281–330.

[16] J.H.Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford Univ. Press, London, 1965.

Pivot	Complete Pivoting	No Pivoting
1	1	=
2	2	=
3	2	=
4	2.5964300000000003429555442	=
5	2.3776999999999999370719964	=
6	2.3038700000000000410065317	2.3038699999999999397929678
7	4.4163321272515367596933451	2.9587400000000000482926402
8	3.8552767929754123988079185	3.5890399999999999939571234
9	4.0185942254812673817312640	4.1163800000000001153146980
10	5.0293415272986442173884056	3.3550400000000000307580270
11	5.7152002451610692471571681	6.5102699999999996365037028
12	5.5949255626529073239265750	6.5102699999999998941614154
13	7.3552186391545473364176438	13.020500001372419493320065 2

Table 1: Gaussian elimination in exact arithmetic on Gould's 13 by 13 example