Stability and pivoting

The book discusses the issue of pivoting in Gauss elimination. This issue occurs because the next equation not yet used as a pivot equation may have a zero coefficient for x_k when we might have wanted to use it as pivot equation for the elimination of the kth unknown.

The book deduces from this that there must already be some difficulty if this coefficient, though not zero, is quite small *compared to the coefficient for x_k in some other equation not yet used as pivot equation*, since that then leads to large multipliers, and large multipliers, according to the book, spell trouble.

But I think that the book's analysis is misleading.

For, if that were the trouble, then I could simply multiply the equation by a sufficiently large number to make the formerly small coefficient as big as I cared to, even bigger than any of the coefficients of x_k in any of the other equations, and that should then cure the trouble. However, changing the book's example, on pages 226-227, in this way, by multiplying the first equation by $1/\delta$, hence looking at the linear system

$$Ax = \begin{bmatrix} 1 & 1/\delta \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} (1+\delta)/\delta \\ 2 \end{bmatrix} = b,$$

does not cure the trouble at all, as running the corresponding variant of the script file `NoPivot` readily shows.

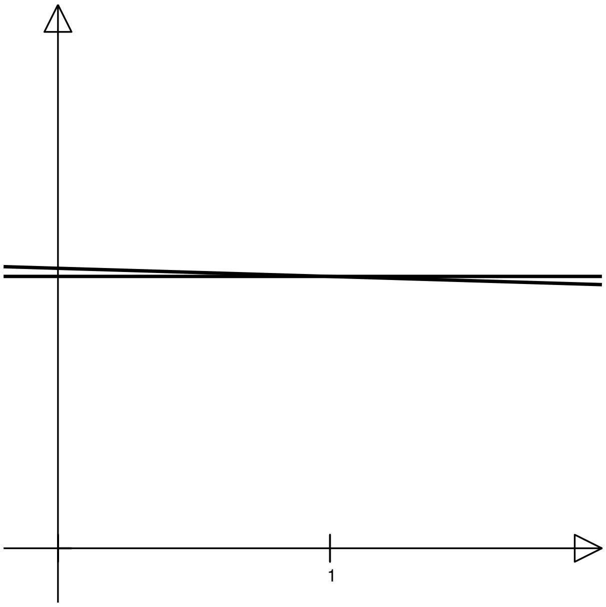
```
% Script File: NoPivot, modified
% Examines solution to
%
%        [ delta 1 ; 1 1][x1;x2] = [1+delta;2]
%
% for a sequence of diminishing delta values.
disp(' Delta            x(1)                   x(2)  ' )
disp('-----------------------------------------------------')
for delta = logspace(-2,-18,9)
   A = [delta 1; delta delta]; %%%%%% changed
   b = [1+delta; 2*delta];     %%%%%% changed
   L = [ 1 0; A(2,1)/A(1,1) 1];
   U = [ A(1,1) A(1,2) ; 0 A(2,2)-L(2,1)*A(1,2)];
   y(1) = b(1);
   y(2) = b(2) - L(2,1)*y(1);
   x(2) = y(2)/U(2,2);
   x(1) = (y(1) - U(1,2)*x(2))/U(1,1);
   disp(sprintf(' %5.0e   %20.15f  %20.15f',delta,x(1),x(2)))
end
```

Here is the output:

```
Delta             x(1)                     x(2)
-----------------------------------------------------
 1e-002       1.000000000000001      1.000000000000000
 1e-004       0.999999999999890      1.000000000000000
 1e-006       0.999999999917733      1.000000000000000
 1e-008       0.999999993922529      1.000000000000000
 1e-010       1.000000082740371      1.000000000000000
 1e-012       0.999866855977416      1.000000000000000
 1e-014       0.999200722162641      1.000000000000000
 1e-016       1.110223024625157      1.000000000000000
 1e-018       0.000000000000000      1.000000000000000
```

True, the numbers have changed some in the less significant part, but the trouble is just as bad as before.

I believe that the real difficulty (at least in this example) is due to the fact that, both times, we choose the first equation as pivot equation for x_1. This is a problem because of the following.



When we determine x_1 from the equation $a_1x_1+a_2x_2 = b_1$, using a computed value $\hat{x}_2$ for x_2, we are, in effect, determining the point at which the straight line $x_2 = \hat{x}_2$ intersects the straight line $a_1x_1 + a_2x_2 = b_1$. This is no problem unless $|a_1| \ll |a_2|$, i.e., unless the straight line of our equation is nearly parallel to the constant straight line $x_2 = \hat{x}_2$, in which case even very small changes in $\hat{x}_2$ (perhaps due to roundoff during the calculation of $\hat{x}_2$) may cause very large changes in the location of this intersection, as is evident from the Figure which shows that situation for the book's problem when δ equal .03.

This seems to say that the difficulty in the book's example really lies with the choice of the pivot equation for x_1. Choosing the first equation for that job is bad because, for small δ, that equation does not determine x_1 very well from a computed x_2.

To put it positively, we should choose as pivot equation for x_1 the one in which the coefficient of x_1 is absolutely large *compared to the other coefficients in its* **row** (rather than in its column). E.g., in the example, the straight line given by the second equation is at a nice 45° angle, and determining its intersection with a line $x_2 = \text{const}$ presents no difficulties.

To enforce this choice turns out to be rather costly since it involves computing, after each step of Gauss elimination, for each equation not yet used as a pivot equation the absolute maximum of each coefficients. A less costly but still somewhat effective alternative is to carry out this calculation just once, at the beginning of the process, and then using the resulting maxima $s_i := \max_j |A(i,j)|$, $i = 1{:}n$, throughout the elimination process to compare the coefficient of x_k in equation i against, picking from among the rows not yet used as pivot rows the row i for which $|A(i,k)|/s_i$ is largest. This way of picking pivot rows is called **scaled partial pivoting**.

when no pivoting is needed: 1. diagonal dominance

A particular happy circumstance occurs when, by our discussion, no pivoting is needed, namely when the coefficient matrix $A = (a_{ij})$ is **diagonally dominant**. This means that, in each row, the diagonal entry is not only the absolutely biggest, it is, in absolute value, bigger than the sum of the absolute values of all the other entries in that row. In formulæ,

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|, \qquad i = 1{:}n.$$

In particular, the first row is an excellent choice for pivot row for the first unknown. But, more than that. After you have used the first row as pivot row to eliminate the first unknown from all other rows, the resulting matrix $\hat{A}$ is just as diagonally dominant (in fact, in a certain sense, it is even more so). A proof of that is given below, for the record. In any case, it says that now the second row is an excellent choice for pivot row for the second unknown. Etc.

Not having to pivot can be very useful. For example, if the coefficient matrix A is tridiagonal, then, for each k, we would like to choose the kth equation as pivot equation for the kth unknown, since that makes it possible to work with just three bands, the diagonal, the subdiagonal, and the superdiagonal, saving both memory and computation time.

As a particular example, take the linear system (3.3) (page 124 of our textbook) for finding the slope s_j at x_j, $j = 1{:}n$, of the complete cubic spline interpolant to data (x_j, y_j), $j = 1{:}n$. We obtain these slopes as the solution to the $n-2$ equations

$$\Delta x_i s_{i-1} + 2(\Delta x_{i-1} + \Delta x_i) s_i + \Delta x_{i-1} s_{i+1} = b_i, \quad i = 2{:}n{-}1,$$

(see (3.3)), with s_1 and s_n assumed given. The resulting linear system is clearly diagonally dominant and tridiagonal, hence knowing that it can be solved stably without pivoting is a great boon.

Here is the proof that elimination without pivoting applied to a diagonally dominant matrix preserves diagonal dominance. (I am merely recording here for my own benefit :-).

I only have to show that, after elimination of the first unknown, using row 1 as its pivot row, the resulting row

$$(0, \hat{a}_{k2}, \ldots, \hat{a}_{kn})$$

of the resulting matrix $\hat{A}$ is still diagonally dominant. Here,

$$\hat{a}_{ki} = a_{ki} - a_{k1}a_{1i}/a_{11}.$$

Therefore,

$$\begin{aligned}
\sum_{i \neq k} |\hat{a}_{ki}| &= \sum_{i \neq k,1} |\hat{a}_{ki}| \\
&\le \sum_{i \neq k,1} |a_{ki}| + \sum_{i \neq k,1} |a_{k1}a_{1i}/a_{11}| \\
&= \sum_{i \neq k} |a_{ki}| - |a_{k1}| + |a_{k1}| \sum_{i \neq 1} |a_{1i}/a_{11}| - |a_{k1}||a_{1i}/a_{11}| \\
&< |a_{kk}| - |a_{k1}| + |a_{k1}| - |a_{k1}||a_{1k}/a_{11}| \\
&\le |a_{k1} - a_{k1}a_{1k}/a_{11}| \;=\; |\hat{a}_{kk}|
\end{aligned}$$

The first equality takes account of the fact that $\hat{a}_{k1} = 0$ by construction. The crucial step is the strict inequality. It uses the fact that (i) $\sum_{i \neq k} |a_{ki}| < |a_{kk}|$; and that also (ii) $\sum_{i \neq 1} |a_{1i}| < |a_{11}|$, hence $\sum_{i \neq 1} |a_{1i}/a_{11}| < 1$, therefore $|a_{k1}| \sum_{i \neq 1} |a_{1i}/a_{11}| < |a_{k1}|$.

when no pivoting is needed: 2. symmetric positive definite

The book rightfully stresses another situation when pivoting is not needed, namely when A is **symmetric positive definite**, or **SPD**. This means that A is (i) **symmetric**, i.e., $A^T = A$; and (ii) **positive definite**, i.e., $x^T A x > 0$ for all $x \neq 0$.

If A is SPD, then necessarily all its diagonal entries are positive, and, for all i and j,

$$a_{ii} + a_{jj} \ge 2|a_{ij}|.$$

More than that, we can write such A as the product

$$A = GG^T$$

of a lower triangular matrix and its transpose, the so-called **Cholesky factorization** for A. The book describes how one can compute the entries of G column by column from the requirement that $A = GG^T$, with the positive definiteness guaranteeing that the pivot element in the kth row, i.e., the entry $G(k, k) = G^T(k, k)$ is positive and not too small compared to the other entries, $G^T(k, j)$, $j > k$, in that row.