Row Reduction Errors

August 30, 2015

We examine how row reduction can lead to errors in computing the inverse of certain types of 2 × 2 matrix. This indicates the contexts in which we need to use more advanced pivoting techniques.

There are three types of pivoting that we consider:

- No pivoting. In other words, we always take the diagonal entry as the pivot.
- Partial Pivoting. At the i-th step we take the largest entry in the i-th column as the pivot.
- Scaled pivoting. At the i-th step we compare the ratios of the entry in the i-th column to the largest entry in the same row. We take as pivot the entry for which this ratio is the largest.

Let us assume that the matrix has entries as powers of 2 and that row interchange has been performed so that the pivot is at the top-left corner. This means that the matrix is:

$$\begin{pmatrix} 2^a & 2^b \\ 2^c & 2^c \end{pmatrix}$$

Moreover, the numbers satisfy the conditions as given below:

- No pivoting. No conditions.
- Partial Pivoting. In this case $a > c$.
- Scaled Pivoting. In this case $a - b > c - d$.

Applying row-reduction with the top-left corner as the pivot we obtain:

$$L = \begin{pmatrix} 1 & 0 \\ 2^{c-a} & 1 \end{pmatrix} \text{ and } U = \begin{pmatrix} 2^a & 2^b \\ 0 & 2^d - 2^{b+c-a} \end{pmatrix}$$

The "machine epsilon" is the biggest number ϵ such that $1 \pm \epsilon$ is the same as 1 when the arithmetic is done on the machine. In a typical modern computer this is 2^{-53} in double precision calculations.

Thus if $|(a+d)-(b+c)| \geq 53$, then some information is lost in U. There are two cases to consider:

- $d-(b+c-a) \leq -53$ in which case we have the computed value $U_1 = \begin{pmatrix} 2^a & 2^b \\ 0 & -2^{b+c-a} \end{pmatrix}$. In this case we have $LU_1 = \begin{pmatrix} 2^a & 2^b \\ 2^c & 0 \end{pmatrix}$
- $d-(b+c-a) \geq 53$ in which case we have the computed value $U_2 = \begin{pmatrix} 2^a & 2^b \\ 0 & 2^d \end{pmatrix}$. In this case we have $LU_1 = \begin{pmatrix} 2^a & 2^b \\ 2^c & 2^d + 2^{b+c-a} \end{pmatrix}$ which computes to $LU_1 = \begin{pmatrix} 2^a & 2^b \\ 2^c & 2^d \end{pmatrix}$.

In the first case, we see that the result of the computation is not the matrix we started with whereas in the second case we get what we started.

We note that if we use "scaled pivoting" then the second case cannot occur! Further, even in the case of "partial pivoting", we have $a > c$ so that the condition $a-c < (b-d)-53$ is only achieved if $b >> d+53$ which happens in fewer cases.

When the matrix is larger, we can produce examples where even "scaled pivoting" will lead to an accmulation of errors. Hence, the most stable method for pivoting is to use "full pivoting" where the pivot chosen is the largest entry of the matrix. However, this is a comparatively slower method and is only employed in rare cases.