A NOTE ON PAIRWISE PIVOTING

Given a non-singular matrix

$$A = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \end{bmatrix},$$

we want to perform Gaussian elimination to A on a machine with limited memory. Assuming that we are only allowed to store two rows in the working memory at a time. Then one way to include some form of pivoting will be as follows.

STEP 1. Compare the magnitude of the leading entries in the rows 1 and 2,

$$A^{(0)} = A = \begin{bmatrix} \boxtimes & \times & \times & \times & \times & \times \\ \boxtimes & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \end{bmatrix}$$

and choose the larger entry as the pivot.

STEP 2. Permute rows 1 and 2 accordingly to bring the larger entry into the pivoting position.

STEP 3. Eliminate the entry below the pivot

$$A^{(1)} = \begin{bmatrix} \boxtimes & \times & \times & \times & \times & \times \\ 0 & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \end{bmatrix}.$$

Notice that the elements in bold are now changed because of the row operation (unless, of course, if the entry below the pivot is 0 to begin with, then we don't do anything and proceed to the next stage).

STEP 4. Repeat STEP 1 through STEP 4 with rows 1 and 3 of the matrix

$$A^{(1)} = \begin{bmatrix} \boxtimes & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ \boxtimes & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \end{bmatrix}$$

Date: November 3, 2005, version 1.0, bug reports: `lekheng@stanford.edu`.

to get

$$A^{(2)} = \begin{bmatrix} \boxtimes & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ \boxtimes & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \end{bmatrix};$$

repeat until we have

$$A^{(n)} = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \end{bmatrix}.$$

STEP 5. Now repeat STEPS 1 through STEPS 5 to the $(n-1) \times (n-1)$ principal block $A^{(n)}[2{:}n; 2{:}n]$ to get

$$B^{(n)} = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \end{bmatrix}.$$

STEP 6. Repeat until you get

$$U^{(n)} = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times \end{bmatrix}.$$

In the homework, you are required to do this for *Gauss-Jordan elimination* instead of Gaussian elimination. The idea is similar but now you will need to eliminate the entry above the pivot in STEP 5:

$$B^{(n)} = \begin{bmatrix} \times & \boxtimes & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \end{bmatrix}.$$

Comparing Gaussian elimination with pairwise pivoting and partial pivoting, you will quickly see that pairwise pivoting offers the following advantages:

- Pairwise pivoting avoids a search for a largest pivot down a column.
- As in partial pivoting, pairwise pivoting has the property that the multipliers are always bounded by 1.

- Pairwise pivoting permits greater parallelism. Observe that the following entries (labelled as $n = ①, \ldots, ⑨$) can be eliminated simultaneously in the nth stage:

$$\begin{bmatrix} \times & \times & \times & \times & \times & \times \\ ① & \times & \times & \times & \times & \times \\ ② & ③ & \times & \times & \times & \times \\ ③ & ④ & ⑤ & \times & \times & \times \\ ④ & ⑤ & ⑥ & ⑦ & \times & \times \\ ⑤ & ⑥ & ⑦ & ⑧ & ⑨ & \times \end{bmatrix}.$$

  Your program does not need to exhibit this parallel feature.

Pairwise pivoting has some serious disadvantages too — most notably a 4^n factor in the error bound. Another point is that while partial pivoting requires 1 row interchange to zero out all entries below a pivot, pairwise pivoting may take up to n row interchanges (STEP 2).