Lecture 21 - *PIVOTING*

OBJECTIVE:
Pure Gaussian elimination is unstable.
But this instability can be controlled by permuting the rows of $\mathbf{A}$ as we proceed; this process is called *pivoting*.
Pivoting has been a standard feature of Gaussian elimination computations since the 1950s.

$\diamond$ PIVOTS

At step k of Gaussian elimination, multiples of row k are subtracted from rows $k+1, k+2, \ldots, m$ of the working matrix $\mathbf{X}$ in order to zero out the elements below the diagonal.

In this operation, row k, column k, and especially x_{kk} play special roles.

We call x_{kk} the *pivot*.

From every entry in the submatrix $\mathbf{X}(k+1:m, k:m)$ we subtract the product of a number in row k and a number in column k, divided by x_{kk}.

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{kk} & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{kk} & \times & \times & \times \\ & 0 & * & * & * \\ & 0 & * & * & * \\ & 0 & * & * & * \end{bmatrix}$$

But there is no inherent reason to eliminate against the k^{th} row.
$\rightarrow$ we could just as easily zero out other entries against row i $\quad (k < i \leq m)$.

In this case, x_{ik} would be the pivot.

e.g., $k = 2,\ i = 4$

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & x_{ik} & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ & 0 & * & * & * \\ & 0 & * & * & * \\ & x_{ik} & \times & \times & \times \\ & 0 & * & * & * \end{bmatrix}$$

Similarly, we could zero out the entries in column j instead of k $\quad (k < j \le m)$.

e.g., $k = 2,\ i = 4,\ j = 3$

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & x_{ij} & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ & * & 0 & * & * \\ & * & 0 & * & * \\ & \times & x_{ij} & \times & \times \\ & * & 0 & * & * \end{bmatrix}$$

Basically, we can choose any entry of $\mathbf{X}(k:m, k:m)$ as the pivot, as long as it is not zero.

This flexibility is good because $x_{kk} = 0$ is possible even in exact arithmetic!

In a floating-point number system, x_{kk} may be "numerically" zero.

For stability, we choose as pivot the element with the largest magnitude among the pivot candidates.

Pivoting in this crazy (but smart!) fashion can be confusing.

$\rightarrow$ It is easy to lose track of what has been zeroed and what still needs to be zeroed.

Instead of leaving x_{ij} in place after it is chosen as pivot (as illustrated above) we interchange rows and columns of the matrix so that x_{ij} takes the position of x_{kk}.

This interchange of rows and/or columns is what is commonly referred to as *pivoting*.

$\rightarrow$ we maintain the look of pure Gaussian elimination in this way.

Note 1. *Elements may or may not actually be swapped in practice!*
We may just keep a list of the positions of the swapped elements.

$\diamond$ PARTIAL PIVOTING

If we consider every element in $\mathbf{X}(k : m, k : m)$ as a possible pivot at step k, then we must examine $(m-k+1)^2$ elements to see which is largest.

$$\therefore \qquad \text{total number of elements examined} =$$

$$\sum_{k=1}^{m}(m-k+1)^2 = \mathcal{O}(m^3) \qquad \textit{(verify!)}$$

$\longrightarrow$ *this adds significantly to the overall cost of Gaussian elimination.*

(This strategy is called *complete pivoting*)

In practice, pivots of essentially the same quality can be found by searching far fewer entries
$\longrightarrow$ this is know as *partial pivoting*

We only allow row interchanges
$\longleftrightarrow$ we only search the elements in column k for the largest one.

So now we only search $(m-k+1)$ elements for a total number $=$

$$\sum_{k=1}^{m}(m-k+1) = \mathcal{O}(m^2) \qquad \textit{(verify!)}$$

$\longrightarrow$ this does not add significantly to the overall cost of

Gaussian elimination (why?).

The act of swapping rows can be viewed as left multiplication by a *permutation matrix* $\mathbf{P}$.

A permutation matrix is simply an identity matrix with its rows or columns permuted.

e.g.,

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

The permutations required to get from $\mathbf{I}$ to $\mathbf{P}$ tell you what permutations $\mathbf{P}$ performs.

e.g., with the above $\mathbf{P}$

$$\begin{aligned} \mathbf{PA} = &\quad (\mathbf{A} \text{ with 2nd and 3rd rows permuted}) \\ \mathbf{AP} = &\quad (\mathbf{A} \text{ with 2nd and 3rd columns permuted}) \\ &\quad \textit{(verify!)} \end{aligned}$$

We also saw in the last lecture that the elimination at step k corresponds to left-multiplication by an elementary lower-triangular matrix $\mathbf{L}_k$.

So the k^{th} step of Gaussian elimination with partial pivoting can be summed up as

$$\underset{\text{pivot-selection}}{\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & x_{ik} & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{\mathbf{P}_1} \underset{\text{row interchange}}{\begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{ik} & * & * & * \\ & \times & \times & \times & \times \\ & * & * & * & * \\ & \times & \times & \times & \times \end{bmatrix}}$$

$$\underset{\text{row interchange}}{\begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{ik} & * & * & * \\ & \times & \times & \times & \times \\ & * & * & * & * \\ & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{\mathbf{L}_1} \underset{\text{elimination}}{\begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{ik} & \times & \times & \times \\ & 0 & * & * & * \\ & 0 & * & * & * \\ & 0 & * & * & * \end{bmatrix}}$$

After $m-1$ steps, $\mathbf{A}$ is transformed into an upper-triangular matrix $\mathbf{U}$:

$$\mathbf{L}_{m-1}\mathbf{P}_{m-1}\ldots\mathbf{L}_2\mathbf{P}_2\mathbf{L}_1\mathbf{P}_1\mathbf{A} = \mathbf{U}$$

Example: Recall our friend

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}$$

To do Gaussian elimination with partial pivoting proceeds as follows:

Interchange the 1st and 3rd rows (left-multiplication by $\mathbf{P}_1$):

$$\begin{bmatrix} & & 1 & \\ & 1 & & \\ 1 & & & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 4 & 3 & 3 & 1 \\ 2 & 1 & 1 & 0 \\ 6 & 7 & 9 & 8 \end{bmatrix}$$

The first elimination step now looks like this (left-multiplication by $\mathbf{L}_1$):

$$\begin{bmatrix} 1 & & & \\ -\frac{1}{2} & 1 & & \\ -\frac{1}{4} & & 1 & \\ -\frac{3}{4} & & & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ 4 & 3 & 3 & 1 \\ 2 & 1 & 1 & 0 \\ 6 & 7 & 9 & 8 \end{bmatrix}$$

$$= \begin{bmatrix} 8 & 7 & 9 & 5 \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \end{bmatrix}$$

Now the 2nd and 4th rows are interchanged (multiplication by $\mathbf{P}_2$):

$$\begin{bmatrix} 1 & & & \\ & & & 1 \\ & & 1 & \\ & 1 & & \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \end{bmatrix}$$

The second elimination step then looks like this (multiplication by $\mathbf{L}_2$):

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & \frac{3}{7} & 1 & \\ & \frac{2}{7} & & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{2}{7} & \frac{4}{7} \\ & & -\frac{6}{7} & -\frac{2}{7} \end{bmatrix}$$

Now the 3rd and 4th rows are interchanged (multiplication by $\mathbf{P}_3$)

$$
\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & & 1 \\ & & 1 & \end{bmatrix}
\begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{2}{7} & \frac{4}{7} \\ & & -\frac{6}{7} & -\frac{2}{7} \end{bmatrix}
= \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{6}{7} & -\frac{2}{7} \\ & & -\frac{2}{7} & \frac{4}{7} \end{bmatrix}
$$

The final elimination step looks like this (multiplication by $\mathbf{L}_3$):

$$
\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & -\frac{1}{3} & 1 \end{bmatrix}
\begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{6}{7} & -\frac{2}{7} \\ & & -\frac{2}{7} & \frac{4}{7} \end{bmatrix}
= \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{6}{7} & -\frac{2}{7} \\ & & & \frac{2}{3} \end{bmatrix}
$$

$\diamond$ $\mathbf{PA} = \mathbf{LU}$ AND A THIRD STROKE OF LUCK

Notice that if you form $\mathbf{L}$ and then $\mathbf{LU}$ from the previous example, you don't get $\mathbf{LU} = \mathbf{A}$!

But you will notice that you <u>almost</u> get $\mathbf{A}$...you get a permuted version of $\mathbf{A}$.
(This should be understandable since we have been messing around with pivots.)

In fact,

$$\mathbf{LU} = \mathbf{PA}$$

where

$$\begin{bmatrix} 1 & & & \\ \frac{3}{4} & 1 & & \\ \frac{1}{2} & -\frac{3}{7} & 1 & \\ \frac{1}{4} & -\frac{3}{7} & \frac{1}{3} & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{6}{7} & -\frac{2}{7} \\ & & & \frac{2}{3} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}$$

Note 2. ***IMPORTANT***
The entries of $\mathbf{L}$ *all satisfy* $|l_{ij}| \leq 1$*.*
This is a consequence of pivoting ($\leftrightarrow$ *eliminating against* $|x_{kk}| = \max_j |x_{jk}|$*).*

But given the way the permutations were introduced, it is not obvious as to why all of them can be lumped into one big $\mathbf{P}$.

i.e.,

$$\mathbf{L}_3\mathbf{P}_3\mathbf{L}_2\mathbf{P}_2\mathbf{L}_1\mathbf{P}_1\mathbf{A} = \mathbf{U}$$

Here is where we use a third stroke of good luck:

These (elementary) operations can be reordered as follows:

$$\mathbf{L}_3\mathbf{P}_3\mathbf{L}_2\mathbf{P}_2\mathbf{L}_1\mathbf{P}_1 = \mathbf{L}_3'\mathbf{L}_2'\mathbf{L}_1'\mathbf{P}_3\mathbf{P}_2\mathbf{P}_1$$

where $\mathbf{L}_k' =$($\mathbf{L}_k$ with subdiagonal entries permuted).

To be precise,

$$
\begin{aligned}
\mathbf{L}_3' &= \mathbf{L}_3 \\
\mathbf{L}_2' &= \mathbf{P}_3\mathbf{L}_2\mathbf{P}_3^{-1} \\
\mathbf{L}_1' &= \mathbf{P}_3\mathbf{P}_2\mathbf{L}_1\mathbf{P}_2^{-1}\mathbf{P}_3^{-1}
\end{aligned}
$$

Note 3. *Each $\mathbf{P}_j$ has $j > k$ for $\mathbf{L}_k$*
$\rightarrow \mathbf{L}_k'$ *has the same structure of $\mathbf{L}_k$* (verify!)

Thus

$$
\begin{aligned}
\mathbf{L}_3'\mathbf{L}_2'\mathbf{L}_1'\mathbf{P}_3\mathbf{P}_2\mathbf{P}_1 &= \mathbf{L}_3(\mathbf{P}_3\mathbf{L}_2\mathbf{P}_3^{-1})(\mathbf{P}_3\mathbf{P}_2\mathbf{L}_1\mathbf{P}_2^{-1}\mathbf{P}_3^{-1})\mathbf{P}_3\mathbf{P}_2\mathbf{P}_1 \\
&= \mathbf{L}_3\mathbf{P}_3\mathbf{L}_2\mathbf{P}_2\mathbf{L}_1\mathbf{P}_1
\end{aligned}
$$

In the general $m \times m$ case, Gaussian elimination with partial pivoting can be written as

$$
(\mathbf{L}_{m-1}' \ldots \mathbf{L}_2'\mathbf{L}_1')(\mathbf{P}_{m-1} \ldots \mathbf{P}_2\mathbf{P}_1)\mathbf{A} = \mathbf{U}
$$

where

$$
\mathbf{L}_k' = \mathbf{P}_{m-1} \ldots \mathbf{P}_{k+1}\mathbf{L}_k\mathbf{P}_{k+1}^{-1} \ldots \mathbf{P}_{m-1}^{-1}
$$

Now we write

$$\mathbf{L} = (\mathbf{L}'_{m-1} \ldots \mathbf{L}'_2\mathbf{L}'_1)^{-1}$$

and

$$\mathbf{P} = (\mathbf{P}_{m-1} \ldots \mathbf{P}_2\mathbf{P}_1)$$

to get

$$\mathbf{PA} = \mathbf{LU}.$$

Note 4. *In general, any square matrix (whether non-singular or not) has a factorization* $\mathbf{PA} = \mathbf{LU}$*, where* $\mathbf{P}$ *is a permutation matrix,* $\mathbf{L}$ *is unit lower-triangular and whose elements satisfy* $|l_{ij}| \leq 1$*, and* $\mathbf{U}$ *is upper-triangular.*

Despite this being a bit of an abuse of notation, this factorization is really what is meant by the LU factorization of $\mathbf{A}$.

The formula $\mathbf{PA} = \mathbf{LU}$ has another great interpretation:

If you could permute $\mathbf{A}$ *ahead of time* using matrix $\mathbf{P}$, then you could apply Gaussian elimination to $\mathbf{A}$ and you would not need pivoting!

Of course, this cannot be done in practice because $\mathbf{P}$ is not known a priori.

ALGORITHM 21.1: GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING

$\mathbf{U} = \mathbf{A}$
$\mathbf{L} = \mathbf{I}$
$\mathbf{P} = \mathbf{I}$
for $k = 1$ to $m - 1$ **do**
  select $i \geq k$ to maximize $|u_{ik}|$
  $u(k, k : m) \longleftrightarrow u(i, k : m)$
  $l(k, 1 : k - 1) \longleftrightarrow l(i, 1 : k - 1)$
  $p(k, :) \longleftrightarrow p(i, :)$
  for $j = k + 1$ to m **do**
    $l(j, k) = u(j, k)/u(k, k)$
    $u(j, k : m) = u(j, k : m) - l(j, k) * u(k, k : m)$
  end for
end for

To leading order, this has complexity $\sim \frac{2}{3}m^3$, just like pure Gaussian elimination.

In the above algorithm, storage could be saved by overwriting $\mathbf{L}$ and $\mathbf{U}$ into $\mathbf{A}$.

$\mathbf{P}$ is never stored as a full matrix, but rather as a vector of indices.

For the example in this lecture, $\mathbf{p} = (3, 4, 2, 1)$.

◇ COMPLETE PIVOTING

In theory, this is more stable than partial pivoting, but the improvements realized in practice are usually marginal.

Formally, the algorithm proceeds as follows

$$\mathbf{L}_{m-1}\mathbf{P}_{m-1} \dots \mathbf{L}_2\mathbf{P}_2\mathbf{L}_1\mathbf{P}_1\mathbf{A}\mathbf{Q}_1\mathbf{Q}_2 \dots \mathbf{Q}_{m-1} = \mathbf{U}$$

where the matrices $\mathbf{Q}_k$ (potentially) permute the columns.

Once again we write this as

$$(\mathbf{L}'_{m-1} \dots \mathbf{L}'_2\mathbf{L}'_1)(\mathbf{P}_{m-1} \dots \mathbf{P}_2\mathbf{P}_1)\mathbf{A}(\mathbf{Q}_1\mathbf{Q}_2 \dots \mathbf{Q}_{m-1}) = \mathbf{U}$$

leading to

$$\mathbf{PAQ} = \mathbf{LU}.$$