These are the lecture notes for CSC349A Numerical Analysis taught by Rich Little in the Spring of 2018. They roughly correspond to the material covered in each lecture in the classroom but the actual classroom presentation might deviate significantly from them depending on the flow of the course delivery. They are provide as a reference to the instructor as well as supporting material for students who miss the lectures. They are simply notes to support the lecture so the text is not detailed and they are not thoroughly checked. Use at your own risk. They are complimentary to the handouts. Many thanks to all the guidance and materials I received from Dale Olesky who has taught this course for many years and George Tzanetakis.

1 Gaussian Elimination with Partial Pivoting

- The Naive Gaussian elimination algorithm will fail if any of the pivots $a_{11}, a_{22}^{(1)}, a_{33}^{(2)}, ...$ is equal to 0.
- Mathematically, it works provided this does not occur.
- Algorithmically, it breaks down when the pivots are even close to 0 because of floating-point arithmetic.
- The problem occurs in the multiplier, it becomes far larger than the other entries.
- **Example:** Consider the $n = 2$ linear system with augmented matrix

$$\left[\begin{array}{cc|c} 0.003 & 59.14 & 59.17 \\ 5.291 & -6.13 & 46.78 \end{array}\right]$$

See Handout 16

Analysis of the above Example

- The source of the extremely inaccurate computed solution $\hat{x}$ is the **large magnitude of the multiplier**.
- Here, 1764 is much larger than the rest of the numbers in the system.
- This number is large because the pivot, $a_{11} = 0.003$, is much smaller than the other numbers in the system.

- Consequently, in the floating-point computations of $a_{22}^{(1)}$ and $b_2^{(1)}$, the numbers -6.13 and 46.78 are so small they are lost.
- The **partial pivoting strategy** is designed to avoid the selection of small pivots.

2 Partial Pivoting

- At step k of forward elimination, where $1 \le k \le n-1$, choose the pivot to be the **largest entry in absolute value**, from

$$\begin{bmatrix} a_{kk} \\ a_{k+1,k} \\ a_{k+2,k} \\ \vdots \\ a_{n,k} \end{bmatrix}$$

- If a_{pk} is the largest (that is, $|a_{pk}| = \max_{k \le i \le n} |a_{ik}|$), then switch row k with row p.
- Note that $|mult| \le 1$ for all multipliers since the denominator is always the largest value.
- Note also that switching rows does not change the final solution. It is an elementary row operation of type 3.

Algorithm 1 pseudocode for partial pivoting

for $k = 1$ to $n - 1$ **do**
    $p = k$
    for $i = k + 1$ to n **do**
        Find largest pivot
    end for
    if $p \neq k$ **then**
        for $j = k$ to n **do**
            swap a_{kj} and a_{pj}
        end for
        swap b_k and b_p
    end if
    do forward elimination
end for
do back susbstitution

3 Example with Pivoting

Example: Solve the following augmented matrix using Gaussian elimination with partial pivoting,

$$\left[\begin{array}{cccc|c} 1 & 1 & 0 & 3 & 4 \\ 2 & 1 & -1 & 1 & 1 \\ 3 & -1 & -1 & 2 & -3 \\ -1 & 2 & 3 & -1 & 4 \end{array}\right]$$

See Handout 16

4 Determinant of A

The reduction of A to upper triangular form by **Naive Gaussian elimination** uses only the type 2 elementary row operation

$$E_i = E_i - factor \times E_j.$$

This row operation does not change the value of the determinant of A. That is, if no rows are interchanged then,

$$\det A = a_{11} a_{22}^{(1)} a_{33}^{(2)} \cdots a_{nn}^{(n-1)}$$

since the determinant of a triangular matrix is equal to the product of its diagonal entries.

However, if **Gaussian elimination with partial pivoting** is used, then each row interchange causes the determinant to change signs (that is, determinant is multiplied by -1.)

Thus, if m row interchanges are done during the reduction of A to upper triangular form, then

$$\det A = (-1)^m a_{11} a_{22}^{(1)} a_{33}^{(2)} \cdots a_{nn}^{(n-1)}$$

As a consequence, Gaussian elimination provides us with a simple method of calculating the determinant of a matrix.

5 Gaussian Elimination with Scaling

- Section 9.4.3 on page 266 in the 7th edition of the text.
- Nothing in the Handouts on this topic.
- If the entries of maximum absolute value in different rows (equations) differ greatly, the computed solution (using floating point arithmetic and partial pivoting) can be very inaccurate.
- **Example** Using $k = 4$ precision, floating-point arithmetic with rounding, solve the following system by Gaussian Elimination with partial pivoting.

$$\left[\begin{array}{cc|c} 2 & 100,000 & 100,000 \\ 1 & 1 & 2 \end{array}\right]$$

Scaling

We look at two ways of using **scaling** to solve this problem: (1) Equilibration and (2) Scaled Factors.

(1) Equilibration:

- Multiply each row by a nonzero constant so that the <u>largest</u> entry in each row of A has magnitude of 1.
- Go through example again with scaling.

- Problem with this form of scaling:
  - Introduces another source of round-off error.

Try on the previous example.

$$\left[\begin{array}{cc|c} 2 & 100,000 & 100,000 \\ 1 & 1 & 2 \end{array}\right]$$

(2) Scaled Factors:

- Use the scaling factors to pick pivots but NOT actually scaling.
- Let $s_i = \max_{1 \leq j \leq n} |a_{ij}|$ for $i = 1, 2, \ldots, n$.
- Step $i = 1$: pivot is max of

$$\begin{bmatrix} |a_{11}/s_1| \\ |a_{21}/s_2| \\ \vdots \\ |a_{n1}/s_n| \end{bmatrix}$$

  If $|a_{p1}/s_p|$ is the max then interchange rows 1 and p then do forward elimination step.

- Step $i = 2$: pivot is max of

$$\begin{bmatrix} |a_{22}^{(1)}/s_2| \\ |a_{32}^{(1)}/s_3| \\ \vdots \\ |a_{n2}^{(1)}/s_n| \end{bmatrix}$$

  If $|a_{q2}/s_q|$ is the max then interchange rows 2 and q then do forward elimination step.

- etc.
- Finish with back susbstitution as usual.

Try also on the previous example.

$$\left[\begin{array}{cc|c} 2 & 100,000 & 100,000 \\ 1 & 1 & 2 \end{array}\right]$$