CMPT-439
Numerical Computation

Fall 2020

Solving Systems of Linear Equations
Directed Elimination Methods

Systems of Linear Equations

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots &= b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots &= b_2 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \cdots &= b_3 \\
&\vdots
\end{aligned}
$$

- In a matrix form:

$$
\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots \\ a_{21} & a_{22} & a_{23} & \cdots \\ a_{31} & a_{32} & a_{33} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \end{bmatrix}
=
\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \end{bmatrix}
$$

Method of Elimination

- The basic strategy is to successively solve one of the equations of the set for one of the unknowns and to eliminate that unknown from the remaining equations by substitution
- Trivial elimination becomes extremely tedious to solve by hand when the number of equations and unknowns is large
- Directed elimination is a sophisticated generalization of the trivial one for solving large systems of equations

Gaussian Elimination

- Extension of the method of trivial elimination to large sets of equations by developing a systematic algorithm to eliminate unknowns and to back substitute
- As in the case of the solution of two equations, the technique for n equations consists of two phases:
  - Forward elimination of unknowns
  - Back substitution

Gaussian Elimination

- The Gaussian elimination method is based on subtracting $\frac{a_{i1}}{a_{11}}$ times the 1[st] equation from the i^{th} equation to make the transformed coefficients in the first column equal to 0
- Then the 2[nd] equation is subtracted $\frac{a_{i2}}{a_{22}}$ times from the i^{th} equation $(i>2)\ldots$
- Finally, the n-1[st] equation is subtracted $\frac{a_{n,n-1}}{a_{n-1,n-1}}$ times from the last equation

- As a result, a matrix of the system becomes upper triangular

Gaussian Elimination

- Gaussian elimination is resulted in elimination of the unknown x_i from all equations subsequent to the i^{th} one
- As a result, a matrix of the system becomes <u>upper triangular</u>

Gaussian Elimination

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array}\right]$$

$$\Downarrow$$

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ & a'_{22} & a'_{23} & b'_2 \\ & & a''_{33} & b''_3 \end{array}\right]$$

Forward elimination

$$\Downarrow$$

$$x_3 = b''_3/a''_{33}$$

$$x_2 = (b'_2 - a'_{23}x_3)/a'_{22}$$

$$x_1 = (b_1 - a_{12}x_2 - a_{13}x_3)/a_{11}$$

Back substitution

Gaussian Elimination

- A diagonal element, which is used for normalization on the current step of Gaussian elimination is called a pivoting element.
- The first pivoting element is a_{11} , then a_{22}, … … … up to $a_{n-1,n-1}$
- If any of $a_{ii} = 0$, this means that either the system does not have a unique solution or some transformations should be done to resolve this issue

Techniques to avoid division by zero

- Use of more significant figures if any of a_{ii} is too small
- Pivoting. If a pivoting element is zero, normalization step leads to division by zero. The same problem may arise, when the pivoting element is close to zero. This can be resolved:
  - Partial pivoting. Switching the rows so that the largest element from the corresponding column becomes the pivoting element.
  - Complete pivoting. Searching for the largest element in all rows and columns then switching both rows and columns so that the largest element is the pivoting element.

Gaussian Elimination: Algorithm with Partial Pivoting

```
i = 1
j = 1
while (i ≤ n and j ≤ m) do
        Find pivot in column j, starting in row i:
        maxi = i
        for k = i+1 to n do
                if abs(A[k,j]) > abs(A[maxi,j]) then
                        maxi := k
                end if
        end for
        if A[maxi,j] ≠ 0
        then    swap rows i and maxi, but do not change the value of i;
                Now A[i,j] will contain the old value of A[maxi,j];
                Divide each entry in row i by A[ i, j ]
                Now A[ i, j ] will have the value 1.
                for u := i+1 to n do
                        subtract A[u ,j ] * (row i) from (row u)
                  Now A[u,j] will be 0, since A[u,j] - A[i,j] * A[u,j] = A[u,j] - 1 * A[u,j] = 0
                end for
                i := i + 1
        end if
        j := j + 1
end while
```

Gaussian Elimination

- After the elimination process is complete and a matrix of the system became upper triangular, then a solution can easily be assembled as follows:

➢ From the last, n^{th} equation $x_n = \dfrac{b'_n}{a'_{nn}}$

➢ Then by substitution of x_n into the n-1^{st} equation $x_{n\text{-}1}$ will be found, etc…

➢ … by substitution of $x_2, \ldots, x_n$ into the 1^{st} equation x_1 will be found

Drawbacks of Elimination Methods

- Division by zero. It is possible that during both elimination and back-substitution phases a division by zero may occur
- Round-off errors
- Ill-conditioned systems (Systems where small changes in coefficients result in large changes in the solution). Alternatively, it happens when two or more equations are nearly identical, resulting a wide ranges of answers to approximately satisfy the equations. Since round-off errors can induce small changes in the coefficients, these changes may then lead to major solution errors

Gauss Program Efficiency

The execution of Gauss elimination depends on the amount of *floating-point operations* (or flops). The flop count for an *n* x *n* system is:

	Forward elimination	$\frac{2n^3}{n} + O(n^2)$
+	Back substitution	$n^2 + O(n)$
	Total	$2n^3 + O(n^2)$

Conclusions:

- As the system gets larger, the computation time increases greatly.
- Most of the effort is incurred in the elimination step.

Gauss-Jordan Elimination

- It is a variation of Gaussian elimination.
- The idea is that the elements above the diagonal are made zero at the same time that zeros are created below the diagonal.
- When an unknown is eliminated, it is eliminated from all other equations rather than just the subsequent ones.
  - All rows are normalized by dividing them by their pivot elements.
  - Elimination step results in the identity matrix.
  - Consequently, it is not necessary to employ back substitution to obtain solution because the solution is resulted from the elimination procedure.

Gauss-Jordan Elimination

- Fundamental operations:
  1. Replace one equation with linear combination of other equations
  2. Interchange two equations
  3. Re-label two variables
- These operations should be combined to reduce a system to be solved to a trivial system

Gauss-Jordan Elimination

- Solve:

$$2x_1 + 3x_2 = 7$$
$$4x_1 + 5x_2 = 13$$

- Only care about numbers taken form an augmented matrix:

$$\left[\begin{array}{cc|c} 2 & 3 & 7 \\ 4 & 5 & 13 \end{array}\right]$$

Gauss-Jordan Elimination

- Given:

$$\left[\begin{array}{cc|c} 2 & 3 & 7 \\ 4 & 5 & 13 \end{array}\right]$$

- Goal: reduce this to trivial system

$$\left[\begin{array}{cc|c} 1 & 0 & a \\ 0 & 1 & b \end{array}\right]$$

and read off an answer from the right-most column

Gauss-Jordan Elimination

$$\left[\begin{array}{cc|c} 2 & 3 & 7 \\ 4 & 5 & 13 \end{array}\right]$$

- Basic operation 1: replace any row by linear combination with any other row
- Here, replace row 1 with $^1/_2$ * row1 + + 0 * row2

$$\left[\begin{array}{cc|c} 1 & 3/2 & 7/2 \\ 4 & 5 & 13 \end{array}\right]$$

Gauss-Jordan Elimination

$$\left[\begin{array}{cc|c} 1 & 3/2 & 7/2 \\ 4 & 5 & 13 \end{array}\right]$$

- Replace row2 with row2 – 4 * row1

$$\left[\begin{array}{cc|c} 1 & 3/2 & 7/2 \\ 0 & -1 & -1 \end{array}\right]$$

- Negate row2 (multiply row2 *(-1))

$$\left[\begin{array}{cc|c} 1 & 3/2 & 7/2 \\ 0 & 1 & 1 \end{array}\right]$$

Gauss-Jordan Elimination

$$\left[\begin{array}{cc|c} 1 & \tfrac{3}{2} & \tfrac{7}{2} \\ 0 & 1 & 1 \end{array}\right]$$

- Replace row1 with row1 – $^3/_2$ * row2

$$\left[\begin{array}{cc|c} 1 & 0 & 2 \\ 0 & 1 & 1 \end{array}\right]$$

- Read off a solution: $x_1 = 2$, $x_2 = 1$

Gauss-Jordan Elimination: the Rule

- For each row $\boldsymbol{i}$:
  - Multiply row $\boldsymbol{i}$ by $\mathbf{1/}\boldsymbol{a_{ii}}$
  - For every single other row j:
    - Add $\boldsymbol{-a_{ji}}$ times row $\boldsymbol{i}$ to row $\boldsymbol{j}$
- This process always results in an identity sub-matrix of the augmented matrix (after the last column of the augmented matrix is dropped)
- A <u>solution appears</u> in the **last (right-most)** column of the augmented matrix

Gauss-Jordan Elimination: Example

Example: The system of equations $\begin{cases} x+y+z=3 \\ 2x+3y+7z=0 \\ x+3y-2z=17 \end{cases}$ has augmented matrix

$$\left[\begin{array}{ccc|c} \boxed{1} & 1 & 1 & 3 \\ 2 & 3 & 7 & 0 \\ 1 & 3 & -2 & 17 \end{array}\right].$$

Row operations can be used to express the matrix in reduced row-echelon form.

Multiply the 1st row by **1/1**

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 3 \\ 2 & 3 & 7 & 0 \\ 1 & 3 & -2 & 17 \end{array}\right] \to \left[\begin{array}{ccc|c} 1 & 1 & 1 & 3 \\ 0 & \boxed{1} & 5 & -6 \\ 0 & 2 & -3 & 14 \end{array}\right]$$

Subtract the 1st row multiplied by 2 from the 2nd row

Subtract the 1st row multiplied by 1 from the 3rd row

Multiply the 2nd row by **1/1**

$$\to \left[\begin{array}{ccc|c} 1 & 0 & -4 & 9 \\ 0 & 1 & 5 & -6 \\ 0 & 0 & \boxed{-13} & 26 \end{array}\right]$$

Subtract the 2nd row multiplied by 1 from the 1st row

Subtract the 2nd row multiplied by 2 from the 3rd row

Multiply the 3rd row by **1/13**

$$\to \left[\begin{array}{ccc|c} 1 & 0 & -4 & 9 \\ 0 & 1 & 5 & -6 \\ 0 & 0 & 1 & -2 \end{array}\right]$$

Subtract the 3rd row multiplied by -4 from the 1st row

Subtract the 3rd row multiplied by 5 from the 2nd row

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

The augmented matrix now says that $x=1$, $y=4$, and $z=-2$.

Pivoting

- Consider this system:

$$\left[\begin{array}{cc|c} 0 & 1 & 2 \\ 2 & 3 & 8 \end{array}\right]$$

- Immediately run into problem: algorithm wants us to divide by zero!
- Slightly better case:

$$\left[\begin{array}{cc|c} 0.001 & 1 & 2 \\ 2 & 3 & 8 \end{array}\right]$$

Partial Pivoting

$$\left[\begin{array}{cc|c} 0 & 1 & 2 \\ 2 & 3 & 8 \end{array}\right]$$

- Swap rows 1 and 2:

$$\left[\begin{array}{cc|c} 2 & 3 & 8 \\ 0 & 1 & 2 \end{array}\right]$$

- Now continue: (multiply the 1st row by ½)

$$\left[\begin{array}{cc|c} 2 & 3 & 8 \\ 0 & 1 & 2 \end{array}\right] \Rightarrow \left[\begin{array}{cc|c} 1 & 3/2 & 4 \\ 0 & 1 & 2 \end{array}\right]$$

Partial Pivoting

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

- Now continue: (subtract the 2nd row multiplied by 3/2 from the first one:

$$\left[\begin{array}{cc|c} 1 & 3/2 & 4 \\ 0 & 1 & 2 \end{array}\right] \Rightarrow \left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 2 \end{array}\right]$$

Full Pivoting

$$\left[\begin{array}{cc|c} 0 & 1 & 2 \\ 2 & 3 & 8 \end{array}\right]$$

- Swap largest element onto diagonal by swapping rows 1 and 2 and columns 1 and 2:

$$\left[\begin{array}{cc|c} 0 & 1 & 2 \\ 2 & 3 & 8 \end{array}\right] \Rightarrow \left[\begin{array}{cc|c} 2 & 3 & 8 \\ 0 & 1 & 2 \end{array}\right] \Rightarrow \left[\begin{array}{cc|c} 3 & 2 & 8 \\ 1 & 0 & 2 \end{array}\right]$$

- <u>Critical</u>: when swapping rows of the augmented matrix, must remember to swap elements in its last column (biases)!

Full Pivoting

- Multiply the 1st row by 3 and subtract it from the 2nd one:

$$\left[\begin{array}{cc|c} 3 & 2 & 8 \\ 1 & 0 & 2 \end{array}\right] \qquad \left[\begin{array}{cc|c} 1 & \frac{2}{3} & \frac{8}{3} \\ 0 & -\frac{2}{3} & -\frac{2}{3} \end{array}\right]$$

- Add the 2nd row to the 1st one:

$$\left[\begin{array}{cc|c} 1 & \frac{2}{3} & \frac{8}{3} \\ 0 & -\frac{2}{3} & -\frac{2}{3} \end{array}\right] \Rightarrow \left[\begin{array}{cc|c} 1 & 0 & 2 \\ 0 & 1 & 1 \end{array}\right]$$