Analysis

Algorithm *GaussElimination* ...

```
for i ← 1 to n do  A[i, n+1] ← b[i]
for i ← 1 to n − 1 do
    for j ← i + 1 to n do
        multiplier ← A[j, i]/A[i, i]    ▷ compute row multiplier once
        for k ← i to n + 1 do
            A[j, k] ← A[j, k] − A[i, k] ∗ multiplier
```

What's the efficiency?

- ✎ Back substitution
- ✎ Solution, system of linear equations

Quiz
What is the efficiency if the coefficients matrix happens to be **upper triangular**.

Gaussian Elimination Corrections

- Scaling factor
- Partial pivoting

http://www.hashimi.ws/cs223/figs/ieee__addition2.pdf

Important to realize that first pseudocode doesn't describe a correct algorithm.

$$\left(\begin{array}{ccc:c} 2 & -1 & 1 & 1 \\ 4 & 1 & -1 & 5 \\ 1 & 1 & 1 & 0 \end{array}\right)$$

Algorithm *BetterElimination*

…

```
for i ← 1 to n do  A[i, n+1] ← b[i]
for i ← 1 to n − 1 do
    find row j with largest value in column i
    swap rows i, j (make j pivotrow)
    for j ← i + 1 to n do
        m ← A[j, i]/A[i, i]                ▷ row multiplier
        for k ← i to n + 1 do
            A[j, k] ← A[j, k] − A[i, k] ∗ m
```

Gaussian Elimination
An Algorithm

Exercise
Find the simpler instance after partial pivoting.

$$\begin{pmatrix} 2 & -1 & 1 \\ 4 & 1 & -1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 5 \\ 0 \end{pmatrix}$$

Exercise
Swap columns 1,2 in addition to partial pivoting in *WolframApha*, verify that solution stays the same.

Algorithm *BetterElimination*
Input Coefficients matrix $A[1..n, 1..n]$, verctor $b[1..n]$
Output Reduced augmented A, equivalent upper matrix in-place

1: **for** $i \leftarrow 1$ **to** n **do** $A[i, n+1] \leftarrow b[i]$
2: **for** $i \leftarrow 1$ **to** $n-1$ **do**
3: $\quad$ $pivotrow \leftarrow i$ $\quad$ ▷ `Lines 3-6 replace default by row with max value under i`
4: $\quad$ **for** $j \leftarrow i+1$ **to** n **do**
5: $\quad\quad$ **if** $\left|A[j,i]\right| > \left|A[pivotrow, i]\right|$ **then** $pivotrow \leftarrow j$
6: $\quad$ **for** $k \leftarrow i$ **to** $n+1$ **do** swap $A[i,k], A[pivotrow, k]$
7: $\quad$ **for** $j \leftarrow i+1$ **to** n **do**
8: $\quad\quad$ $m \leftarrow A[j,i]/A[i,i]$
9: $\quad\quad$ **for** $k \leftarrow i$ **to** $n+1$ **do**
10: $\quad\quad\quad$ $A[j,k] \leftarrow A[j,k] - A[i,k] * m$

Gaussian Elimination Applications

- Solution system of linear equations
- *LU* decomposition of matrix
- Computing matrix inverse
- Computing the determinant

Gaussian elimination is fundamental to matrix processing (just like sorting for lists).

Gaussian Elimination Applications
LU Decomposition

- Lower matrix
- Amortized efficiency

$$\begin{pmatrix} 2 & -1 & 1 \\ 4 & 1 & -1 \\ 1 & 1 & 1 \end{pmatrix}$$

Quiz
Where did elements of *L*,*U* come from?

Exercise
Find decomposition after partial pivoting, write the re-arranged matrix *A*.

$$U = \begin{pmatrix} 2 & -1 & 1 \\ 0 & 3 & -3 \\ 0 & 0 & 2 \end{pmatrix} \quad L = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{pmatrix}$$

$$\mathbf{A}x = \mathbf{b} \qquad \overbrace{L\underbrace{Ux}_{y}}^{\mathbf{A}} = \mathbf{b}$$

Quiz
What's the (overall) efficiency of solving a system given an *LU* decomposition of coefficients matrix?

Quiz
What about a sequence of systems with different right-hand side vectors and the same coefficients?

$$Ly = \mathbf{b} \qquad Ux = y$$

Gaussian Elimination Applications
Matrix Inverse

- Singular matrix
- Identity matrix

$A^{-1}A = I = A\,A^{-1}$

Quiz
How can Gaussian elimination be used to test if matrix is **singular**?

Exercise
Use ***WolframAlpha*** to find A^{-1} (from Slide 3), verify solution to the system of equations.

Exercise
Compare the cost of computing the inverse via Gaussian elimination for each of the n systems of equations, or using the *LU* decomposition of *A* then solving for each column in I.

Importance

$$A^{-1}Ax = A^{-1}b$$

$$x = A^{-1}b$$

Computing the inverse

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_{11} & \boldsymbol{x_{12}} & \cdots & x_{1n} \\ x_{21} & \boldsymbol{x_{22}} & \cdots & x_{2n} \\ \vdots & & & \\ x_{n1} & \boldsymbol{x_{n2}} & \cdots & x_{nn} \end{pmatrix} = \begin{pmatrix} 1 & \mathbf{0} & \cdots & 0 \\ 0 & \mathbf{1} & \cdots & 0 \\ \vdots & & & \\ 0 & \mathbf{0} & \cdots & 1 \end{pmatrix}$$

$A\,A^{-1}$

Gaussian Elimination
Take Home

Basic methods and strategies to compute the solution of a system of linear equations and related matrix operations, interesting scenarios, and resulting efficiencies.

- **Computational effort**
- **Scenarios, efficiencies**
- **Use *WolframAlpha* to study**
- **Applications of matrix and linear algebra**

Exercise
Look up info about the test workload suite **Linpack**.