### **Cost of LU Factorization**

#### **Contents**

1 Cost of PLU Factorization Algorithm

# Cost of PLU Factorization Algorithm

Key result: In solving  $A\vec{x} = \vec{b}$  where  $A \in \mathbb{R}^{n \times n}$ :

- PLU factorization requires about  $\frac{2}{3}$  n<sup>3</sup> flops
- Forward & Backward substitutions

require only about no flops

## Notation: Big-O and Asymptotic

$$f(1), f(2), f(3), ---$$

Let f, q be positive functions defined on  $\mathbb{N}$ .

Let 
$$f,g$$
 be positive functions defined on  $\mathbb{N}$ .

•  $f(n) = O\left(g(n)\right)$  (" $f$  is  $b$ ig- $O$  of  $g$ ") as  $n \to \infty$  if

Complex simple  $\frac{f(n)}{g(n)} \leqslant C$ , for all sufficiently large  $n$ .

•  $f(n) \sim g(n)$  (" $f$  is asymptotic to  $g$ ") as  $n \to \infty$  if

•  $f(n) \sim g(n)$  (" $f$  is asymptotic to  $g$ ") as  $n \to \infty$  if

•  $f(n) \sim g(n)$  ("f is asymptotic to g") as  $n \to \infty$  if

tic to 
$$g''$$
) as  $n \to \infty$  if 
$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 1.$$

more accurate description

Examples Let 
$$f(n) = 3n^3 + 2n - 1$$

•  $f(n) = O(n^3)$  because  $\frac{3n^3 + 2n - 1}{n^3} = 3 + \frac{2}{n^2} - \frac{1}{n^3} \leqslant 3 + 1 + 1 = 5$ 

for all large  $n$ .

(  $f(n) = O(5n^3)$ ,  $f(n) = O(n^4)$ ,  $f(n) = O(n^6)$ , ----

•  $f(n) \sim 3n^3$  because  $\lim_{n \to \infty} \frac{3n^3 + 2n - 1}{3n^3} = 1$ .

### Timing Vector/Matrix Operations - FLOPS

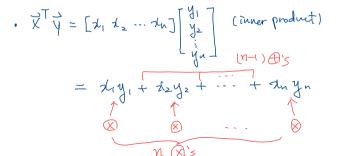
- One way to measure the "efficiency" of a numerical algorithm is to count the number of floating-point arithmetic operations (FLOPS) necessary for its execution.
- The number is usually represented by  $\underline{\sim cn^{[\!\![\!]\!\!]}}$  where c and p are given explicitly.
- We are interested in this formula when n is large.

### **FLOPS** for Major Operations

#### Vector/Matrix Operations

Let  $x, y \in \mathbb{R}^n$  and  $A, B \in \mathbb{R}^{n \times n}$ . Then

- (vector-vector)  $x^{\mathrm{T}}y$  requires  $\sim 2n$  flops.
- (matrix-vector) Ax requires  $\sim 2n^2$  flops.
- (matrix-matrix) AB requires  $\sim 2n^3$  flops.



 $= 2n-1 \sim 2n$ 

$$A \overrightarrow{x} = \begin{bmatrix} \overrightarrow{\alpha_1} \\ \vdots \\ \overrightarrow{\alpha_n} \\ \end{bmatrix} \overrightarrow{x} = \begin{bmatrix} \overrightarrow{\alpha_1} \\ \vdots \\ \overrightarrow{\alpha_n} \\ \end{array}$$
 ~ 2n \quad n \times

$$A = \begin{bmatrix} \overrightarrow{\lambda}_1^T \\ \vdots \\ \overrightarrow{\lambda}_n^T \end{bmatrix}$$

Total flops: 
$$(2n-1) \times n$$
  
=  $2n^2 - n \sim 2n^2$ 

#### Cost of PLU Factorization

of PLU Factorization | Privot: 
$$R_i \hookrightarrow R_j$$
 (No flops needed)

. Now replacement:  $R_i \rightarrow R_i + (-\frac{\alpha_{ij}}{\alpha_{ij}}) R_j$ 

Note that we only need to count the number of flops required to zero out elements below the diagonal of each column.

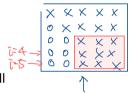
- Spose you are working out the column. For each i>j, we replace  $R_i$  by  $R_i+cR_j$  where  $c=-a_{i,j}/a_{j,j}$ . This requires approximately 2(n-j+1) flops:

  - 1 division to form *c*
  - 1 division to form c• n-j+1 multiplications to form  $cR_j$  n-j+1 additions to form  $R_j$   $\sim 2(n-j+1)$
- n-j+1 additions to form  $R_i+cR_i$ • Since  $i \in \mathbb{N}[j+1,n]$ , the total number of flops needed to zero out all elements below the diagonal in the *i*th column is approximately
  - 2(n-i+1)(n-i).
- Summing up over  $j \in \mathbb{N}[1, n-1]$ , we need about  $(2/3)n^3$  flops:

$$\sum_{j=1}^{n-1} 2(\underbrace{n-j+1}_{\text{N-J}})(n-j) \sim 2 \sum_{j=1}^{n-1} (n-j)^2 = 2 \sum_{j=1}^{n-1} j^2 \sim \frac{2}{3} n^3$$
 reversing the term

( change of indices)

(flop counting needed)



$$3 = 5 - 3 + 1$$
 $= n - 5 + 1$ 

Why 
$$2 \int_{5=1}^{n-1} \int_{5}^{2} \sim \frac{2}{3} n^{3}$$
?

$$\int_{1}^{\infty} \int_{1}^{\infty} \int_{1$$

One way Recall 
$$\sum_{j=1}^{n} j^2 = 1^2 + 2^2 + \dots + n^2 = \frac{n(n+1)(2n+1)}{6}$$
.

$$\int_{0}^{\infty} \int_{0}^{\infty} \int_{0$$

$$\int_{S=1}^{\infty} \int_{S=1}^{\infty} \int_{S$$

$$\int_{3}^{2} \int_{3}^{2} \int_{3$$

$$\frac{1}{\sqrt{3}}$$

$$\frac{0}{2} = \frac{1}{2} \cdot \frac{1}$$

$$\int_{-1}^{2} \int_{-1}^{2} \int_{-1}^{2$$

 $\sim \frac{2n^3}{6} = \frac{n^3}{2}$ 

- $\sum_{n=1}^{N-1} j^2 = \frac{(N-1)[(N-1)+1][2(N-1)+1]}{\sqrt{(N-1)+1}}$ 

  - $= \frac{(n-1) n (2n-1)}{6} = \frac{2n^3 + (boven order terms)}{6}$

Another Using the following result.

$$\frac{\sum_{j=1}^{N-1} \tilde{j}^{p}}{\tilde{j}^{p}} \sim \frac{\tilde{j}^{p+1}}{p+1}$$

Think 
$$\int d^{2} dd = \frac{d^{2}}{p+1} + C$$

$$\sum_{j=1}^{n-1} j^{2} \sim \frac{j^{2+1}}{2+1} = \frac{j^{3}}{3}$$

#### Cost of Forward Elimination and Backward Substitution

#### **Forward Elimination**

- The calculation of  $y_i = \beta_i \sum_{j=1}^{i-1} \ell_{ij} y_j$  for i > 1 requires approximately 2i flops:
  - 1 subtraction
  - i-1 multiplications
  - i-2 additions
- Summing over all  $i \in \mathbb{N}[2, n]$ , we need about  $n^2$  flops:

$$\sum_{i=2}^{n} 2i \sim 2\frac{n^2}{2} = n^2.$$

#### **Backward Substitution**

• The cost of backward substitution is also approximately  $n^2$  flops, which can be shown in the same manner.

### Cost of G.E. with Partial Pivoting

Gaussian elimination with partial pivoting involves three steps:

- PLU factorization:  $\sim (2/3)n^3$  flops
- Forward elimination:  $\sim n^2$  flops
- Backward substitution:  $\sim n^2$  flops

#### Summary

The total cost of Gaussian elimination with partial pivoting is approximately

$$\frac{2}{3}n^3 + n^2 + n^2 \sim \frac{2}{3}n^3$$

flops for large n.

# Application: Solving Multiple Square Systems Simultaneously

To solve two systems  $A\mathbf{x}_1 = \mathbf{b}_1$  and  $A\mathbf{x}_2 = \mathbf{b}_2$ . (Note both involve the same matrix.)

#### Method 1. (mefficient)

- Use G.E. for both.
- It takes  $\sim (4/3)n^3$  flops.

### Method 2. ( efficient )

- Do it in two steps:
  - 1 Do PLU factorization PA = LU.
  - 2 Then solve  $LU\mathbf{x}_1 = P\mathbf{b}_1$  and  $LU\mathbf{x}_2 = P\mathbf{b}_2$ .
- It takes  $\sim (2/3)n^3$  flops.

%% method 1  

$$x1 = A \setminus b1; \sim \frac{2}{3} n^3$$
  
 $x2 = A \setminus b2; \sim \frac{1}{3} n^3$ 

```
%% method 2
[L, U, P] = lu(A); \sim \frac{2}{3}n^3
\times 1 = U \setminus (L \setminus (P*b1)); \sim 2n^2
\times 2 = U \setminus (L \setminus (P*b2)); \sim 2n^2
```

```
%% compact implementation

X = A \ [b1, b2];

x1 = X(:, 1);

x2 = X(:, 2);
```

$$A \begin{bmatrix} \vec{x}, \vec{x}_2 \end{bmatrix} = \begin{bmatrix} \vec{b}, \vec{b}, \end{bmatrix}$$