Lec 15: Square Linear Systems – Analysis

Efficiency

Notation: Big-O and Asymptotic

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

- $f(n) = O\left(g(n)\right)$ ("f is *big-O* of g") as $n \to \infty$ if

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

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

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

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 $\sim cn^p$ 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*.

Cost of PLU Factorization

Note that we only need to count the number of *flops* required to zero out elements below the diagonal of each 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
  - $n-j+1$ multiplications to form cR_j
  - $n-j+1$ additions to form $R_i + cR_j$
- Since $i \in \mathbb{N}[j+1, n]$, the total number of *flops* needed to zero out all elements below the diagonal in the jth column is approximately $2(n-j+1)(n-j)$.
- Summing up over $j \in \mathbb{N}[1, n-1]$, we need about $(2/3)n^3$ *flops*:

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

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 $A\mathbf{x}_2 = \mathbf{b}_2$.

Method 1.

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

```
%% method 1
x1 = A \ b1;
x2 = A \ b2;
```

Method 2.

- 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 2
[L, U, P] = lu(A);
x1 = U \ (L \ (P*b1));
x2 = U \ (L \ (P*b2));
```

```
%% compact implementation
X = A \ [b1, b2];
x1 = X(:, 1);
x2 = X(:, 2);
```

Vector and Matrix Norms

Vector Norms

The "length" of a vector $\mathbf{v}$ can be measured by its **norm**.

Definition 1 (p-Norm of a Vector)

Let $p \in [1, \infty)$. The p-norm of $\mathbf{v} \in \mathbb{R}^m$ is denoted by $\|\mathbf{v}\|_p$ and is defined by

$$\|\mathbf{v}\|_p = \left(\sum_{i=1}^{m} |v_i|^p \right)^{1/p} .$$

When $p = \infty$,

$$\|\mathbf{v}\|_\infty = \max_{1 \leqslant i \leqslant m} |v_i| .$$

The most commonly used p values are 1, 2, and ∞:

$$\|\mathbf{v}\|_1 = \sum_{i=1}^{m} |v_i| , \quad \|\mathbf{v}\|_2 = \sqrt{\sum_{i=1}^{m} |v_i|^2}.$$

Vector Norms

In general, any function $\|\cdot\| : \mathbb{R}^m \to \mathbb{R}^+ \cup \{0\}$ is called a **vector norm** if it satisfies the following three properties:

1. $\|\mathbf{x}\| = 0$ if and only if $\mathbf{x} = 0$.
2. $\|\alpha\mathbf{x}\| = |\alpha| \, \|\mathbf{x}\|$ for any constant α and any $\mathbf{x} \in \mathbb{R}^m$.
3. $\|\mathbf{x} + \mathbf{y}\| \leqslant \|\mathbf{x}\| + \|\mathbf{y}\|$ for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$. This is called the *triangle inequality*.

Unit Vectors

- A vector $\mathbf{u}$ is called a **unit vector** if $\|\mathbf{u}\| = 1$.
- Depending on the norm used, unit vectors will be different.
- For instance:

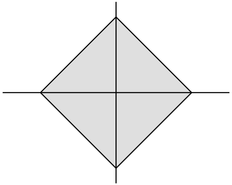

Figure 1: 1-norm

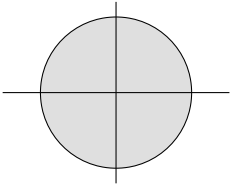

Figure 2: 2-norm

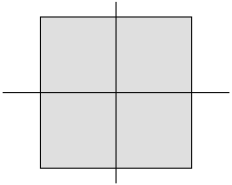

Figure 3: ∞-norm

Matrix Norms

The “size” of a matrix $A \in \mathbb{R}^{m\times n}$ can be measured by its **norm** as well. As above, we say that a function $\|\cdot\| : \mathbb{R}^{m\times n} \to \mathbb{R}^{+} \cup \{0\}$ is a **matrix norm** if it satisfies the following three properties:

1. $\|A\| = 0$ if and only if $A = 0$.
2. $\|\alpha A\| = |\alpha| \, \|A\|$ for any constant α and any $A \in \mathbb{R}^{m\times n}$.
3. $\|A + B\| \leqslant \|A\| + \|B\|$ for any $A, B \in \mathbb{R}^{m\times n}$. This is called the *triangle inequality*.

Matrix Norms (Cont')

- If, in addition to satisfying the three conditions, it satisfies

$$\|AB\| \leqslant \|A\| \, \|B\| \quad \text{for all } A \in \mathbb{R}^{m\times n} \text{ and all } B \in \mathbb{R}^{n\times p},$$

  it is said to be **consistent**.
- If, in addition to satisfying the three conditions, it satisfies

$$\|A\mathbf{x}\| \leqslant \|A\| \, \|\mathbf{x}\| \quad \text{for all } A \in \mathbb{R}^{m\times n} \text{ and all } \mathbf{x} \in \mathbb{R}^{n},$$

  then we say that it is **compatible** with a vector norm.

Induced Matrix Norms

Definition 2 (p-Norm of a Matrix)

Let $p \in [1, \infty]$. The p-norm of $A \in \mathbb{R}^{m \times n}$ is given by

$$\|A\|_p = \max_{\mathbf{x} \neq 0} \frac{\|A\mathbf{x}\|_p}{\|\mathbf{x}\|_p} = \max_{\|\mathbf{x}\|_p = 1} \|A\mathbf{x}\|_p \ .$$

- The definition of this particular matrix norm is **induced** from the vector p-norm.
- By construction, matrix p-norm is a compatible norm.
- Induced norms describe how the matrix stretches unit vectors with respect to the vector norm.

Induced Matrix Norms

The commonly used p-norms (for $p = 1, 2, \infty$) can also be calculated by

$$
\begin{aligned}
\|A\|_1 &= \max_{1 \leqslant j \leqslant n} \sum_{i=1}^{m} |a_{ij}|, \\
\|A\|_2 &= \sqrt{\lambda_{\max}(A^{\mathrm{T}} A)} = \sigma_{\max}(A), \\
\|A\|_\infty &= \max_{1 \leqslant i \leqslant m} \sum_{j=1}^{n} |a_{ij}|.
\end{aligned}
$$

In words,

- The 1-norm of A is the maximum of the 1-norms of all column vectors.
- The 2-norm of A is the square root of the largest eigenvalue of $A^{\mathrm{T}} A$.
- The ∞-norm of A is the maximum of the 1-norms of all row vectors.

Non-Induced Matrix Norm – Frobenius Norm

Definition 3 (Frobenius Norm of a Matrix)

The Frobenius norm of $A \in \mathbb{R}^{m \times n}$ is given by

$$\|A\|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{1/2}.$$

- This is not induced from a vector p-norm.
- However, both p-norm and the Frobenius norm are consistent and compatible.
- For compatibility of the Frobenius norm, the vector norm must be the 2-norm, that is, $\|A\mathbf{x}\|_2 \leqslant \|A\|_F \|\mathbf{x}\|_2$.

Norms in MATLAB

- Vector p-norms can be easily computed:

```
norm(v, 1)       % = sum(abs(v))
norm(v, 2)       % = sqrt(v'*v)  if v is a column
norm(v, 'inf')   % = max(abs(v))
```

- The same function `norm` is used to calculate matrix p-norms:

```
norm(A, 1)        % = max(sum(abs(A), 1))
norm(A, 2)        % = max(sqrt(eig(A'*A)))
norm(A, Inf)      % = max(sum(abs(A), 2))
```

- To calculate the Frobenius norm:

```
norm(A, 'fro')    % = sqrt(A(:)'*A(:))
                  % = norm(A(:), 2)
```