CMPSC/Math 451, Numerical Computation

Wen Shen

Department of Mathematics, Penn State University

Direct methods for systems of linear equations

The problem: n equations, n unknowns,

$$\left\{ \begin{array}{rcll} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n & = & b_1 & \qquad (1) \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n & = & b_2 & \qquad (2) \\ & \vdots & & \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n & = & b_n & \qquad (n) \end{array} \right.$$

In matrix-vector form:

$$A\vec{x} = \vec{b},$$

where $A \in \boldsymbol{R}^{n \times n}, \quad \vec{x} \in \boldsymbol{R}^n, \quad \vec{b} \in \boldsymbol{R}^n$

$$A = \{a_{ij}\} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}, \qquad \vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \qquad \vec{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}.$$

Naive Gaussian elimination

Step 1: Forward elimination.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n & = & b_1 & & (1) \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n & = & b_2 & & (2) \\ & \vdots & & & \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n & = & b_n & & (n) \end{cases}$$

Make an upper triangular system!
Algorithm:

for $k = 1, 2, 3, \cdots, n-1$

for $j = k+1, k+2, \cdots, n$

$(j) \leftarrow (j) - (k) \times \frac{a_{jk}}{a_{kk}}$,

end

end

Work count: $\#\text{flops}=\frac{1}{3}(n^3 - n) = \mathcal{O}(n^3)$

Step 2: Backward substitution

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \qquad (1) \\ a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \qquad (2) \\ &\vdots \\ a_{nn}x_n &= b_n \qquad (n) \end{cases}$$

Algorithm:

$x_n = \frac{b_n}{a_{nn}}$

for $i = n-1, n-2, \cdots, 1$

$\quad x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=i+1}^{n} a_{ij}x_j \right)$

end

Work count: #flops $= \frac{1}{2}(n^2 - n) = \mathcal{O}(n^2)$

Total work count: $= \mathcal{O}(n^3)$. Extremely slow.

Tridiagonal system

$$A = \begin{pmatrix} d_1 & c_1 & 0 & \cdots & 0 & 0 & 0 \\ a_1 & d_2 & c_2 & \cdots & 0 & 0 & 0 \\ 0 & a_2 & d_3 & \ddots & 0 & 0 & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & d_{n-2} & c_{n-2} & 0 \\ 0 & 0 & 0 & \cdots & a_{n-2} & d_{n-1} & c_{n-1} \\ 0 & 0 & 0 & \cdots & 0 & a_{n-1} & d_n \end{pmatrix}$$

Gaussian Elimination can be very efficiently:
Step 1: Forward Elimination:

for $i = 2, 3, \cdots, n$

$$d_i \leftarrow d_i - \tfrac{a_{i-1}}{d_{i-1}} c_{i-1}$$
$$b_i \leftarrow b_i - \tfrac{a_{i-1}}{d_{i-1}} b_{i-1}$$

end

Now the A matrix looks like

$$A = \begin{pmatrix} d_1 & c_1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & d_2 & c_2 & \cdots & 0 & 0 & 0 \\ 0 & 0 & d_3 & \ddots & 0 & 0 & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & d_{n-2} & c_{n-2} & 0 \\ 0 & 0 & 0 & \cdots & 0 & d_{n-1} & c_{n-1} \\ 0 & 0 & 0 & \cdots & 0 & 0 & d_n \end{pmatrix}$$

Step 2: Backward substitution:

$x_n \leftarrow b_n/d_n$

for $i = n-1, n-2, \cdots, 1$

$x_i \leftarrow \frac{1}{d_i}(b_i - c_i x_{i+1})$

end

Amount of work: $\mathcal{O}(n)$. Very efficient!

Review of linear algebra

Consider a square matrix $A = \{a_{ij}\}$. A is called *strictly diagonal dominant* if

$$|a_{ii}| > \sum_{j=1, j \neq i}^{n} |a_{ij}|, \quad i = 1, 2, \cdots, n$$

Properties:

- A is regular, invertible, A^{-1} exists, and $Ax = b$ has a unique solution.
- $Ax = b$ can be solved by Gaussian Elimination without pivoting.

One such example: the system from natural cubic spline.

Vector and matrix norms

A norm: measures the "size" of the vector and matrix.

General norm properties: Denote $\|x\|$ the norm of x. Then

1. $\|x\| \geq 0$, equal if and only if $x = 0$;
2. $\|ax\| = |a| \cdot \|x\|$, $\quad a$: is a constant;
3. $\|x + y\| \leq \|x\| + \|y\|$, triangle inequality.

Examples of vector norms: $x \in \boldsymbol{R}^n$

1. $\|x\|_1 = \sum_{i=1}^{n} |x_i|,$ $\qquad$ l_1-norm
2. $\|x\|_2 = \left(\sum_{i=1}^{n} x_i^2\right)^{1/2},$ $\qquad$ l_2-norm
3. $\|x\|_\infty = \max_{1 \le i \le n} |x_i|,$ $\qquad$ l_∞-norm

Matrix norms

Matrix norm is defined in term of the corresponding vector norm:

$$\|A\| = \max_{\vec{x} \neq 0} \frac{\|Ax\|}{\|x\|}$$

Properties:

$$\|A\| \geq \frac{\|Ax\|}{\|x\|} \quad \Rightarrow \quad \|Ax\| \leq \|A\| \cdot \|x\|$$

$$\|I\| = 1, \qquad \|AB\| \leq \|A\| \cdot \|B\| \,.$$

Examples of matrix norms:

$$
\begin{aligned}
l_1 - \text{norm} &: \quad \|A\|_1 = \max_{1 \le j \le n} \sum_{i=1}^{n} |a_{ij}| \\
l_2 - \text{norm} &: \quad \|A\|_2 = \max_i |\lambda_i| , \qquad \lambda_i : \text{ eigenvalues of } A \\
l_\infty - \text{norm} &: \quad \|A\|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}|
\end{aligned}
$$

Eigenvalues λ_i for A

$$Av = \lambda v, \qquad \lambda: \text{ eigenvalue}, \qquad v: \text{ eigenvector}$$

$$(A - \lambda I)v = 0, \quad \Rightarrow \quad \det(A - \lambda I) = 0: \qquad \text{polynomial of degree } n$$

Property:

$$\lambda_i(A^{-1}) = \frac{1}{\lambda_i(A)}$$

This implies

$$\left\|A^{-1}\right\|_2 = \max_i \left|\lambda_i(A^{-1})\right| = \max_i \frac{1}{|\lambda_i(A)|} = \frac{1}{\min_i |\lambda_i(A)|}$$

Condition number of a matrix A

$$\text{Want to solve:} \qquad Ax = b$$

$$\text{Put some perturbation:} \qquad A\bar{x} = b + p$$

Relative errors:

$$e_b = \frac{\|p\|}{\|b\|}, \qquad e_x = \frac{\|\bar{x} - x\|}{\|x\|} \qquad \text{relation between them?}$$

We have

$$A(\bar{x} - x) = p, \quad \Rightarrow \quad \bar{x} - x = A^{-1}p$$

so

$$e_x = \frac{\|\bar{x} - x\|}{\|x\|} = \frac{\left\|A^{-1}p\right\|}{\|x\|} \leq \frac{\left\|A^{-1}\right\| \cdot \|p\|}{\|x\|}.$$

$$Ax = b \quad \Rightarrow \quad \|Ax\| = \|b\| \quad \Rightarrow \quad \|A\| \, \|x\| \geq \|b\| \quad \Rightarrow \quad \frac{1}{\|x\|} \leq \frac{\|A\|}{\|b\|}$$

we get

$$e_x \leq \frac{\left\|A^{-1}\right\| \cdot \|p\|}{\|x\|} \leq \left\|A^{-1}\right\| \cdot \|p\| \cdot \frac{\|A\|}{\|b\|} = \|A\| \cdot \left\|A^{-1}\right\| e_b = \kappa(A) \cdot e_b,$$

$$\kappa(A) = \|A\| \cdot \left\|A^{-1}\right\| : \qquad \text{the condition number of } A$$

$$\text{Using } l_2\text{-norm:} \quad \kappa(A) = \|A\|_2 \cdot \left\|A^{-1}\right\|_2 = \frac{\max_i |\lambda_i(A)|}{\min_i |\lambda_i(A)|}$$

Error in b propagates with a factor of $\kappa(A)$ into the solution.
If $\kappa(A)$ is very large, $Ax = b$ is very sensitive to perturbation, therefore difficult to solve. We call this *ill-conditioned system*.

Some Matlab commands:

```
norm(x);          % vector norm
eig(A);           % eigenvalue/eigen vector of a matrix
cond(A);          % condition number of A
```