MATH 3795
Lecture 5. Solving Linear Systems 3

Dmitriy Leykekhman

Fall 2008

Goals

- Positive definite and definite matrices.
- Cholesky decomposition.
- LU-Decomposition of Tridiagonal Systems
- Applications.

Symmetric Positive Definite Matrices

- $A \in \mathbb{R}^{n \times n}$ is called symmetric if $A = A^T$.
- $A \in \mathbb{R}^{n \times n}$ is called symmetric positive definite if $A = A^T$ and $v^T A v > 0$ for all $v \in \mathbb{R}^n$, $v \neq 0$.

Symmetric Positive Definite Matrices

- $A \in \mathbb{R}^{n\times n}$ is called symmetric if $A = A^T$.
- $A \in \mathbb{R}^{n\times n}$ is called symmetric positive definite if $A = A^T$ and $v^T A v > 0$ for all $v \in \mathbb{R}^n$, $v \neq 0$.
- $A \in \mathbb{R}^{n\times n}$ is called m- banded if $a_{ij} = 0$ for $|i - j| > m$.

Symmetric Positive Definite Matrices

- $A \in \mathbb{R}^{n\times n}$ is called symmetric if $A = A^T$.
- $A \in \mathbb{R}^{n\times n}$ is called symmetric positive definite if $A = A^T$ and $v^T A v > 0$ for all $v \in \mathbb{R}^n$, $v \neq 0$.
- $A \in \mathbb{R}^{n\times n}$ is called m- banded if $a_{ij} = 0$ for $|i - j| > m$.
- If $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite, then the LU decomposition can be computed in a stable way without permutation, i.e.,

$$A = LU$$

Symmetric Positive Definite Matrices

- $A \in \mathbb{R}^{n \times n}$ is called symmetric if $A = A^T$.
- $A \in \mathbb{R}^{n \times n}$ is called symmetric positive definite if $A = A^T$ and $v^T A v > 0$ for all $v \in \mathbb{R}^n$, $v \neq 0$.
- $A \in \mathbb{R}^{n \times n}$ is called m- banded if $a_{ij} = 0$ for $|i - j| > m$.
- If $A \in \mathbb{R}^{n \times n}$ is symmetric positive definite, then the LU decomposition can be computed in a stable way without permutation, i.e.,
$$A = LU$$
- Can we use the structure of A, i.e. symmetry and the positive definiteness of A or the property that A is m-banded to compute LU-decomposition more efficiently?

Symmetric Positive Definite Matrices

- **Fact 1:** If L is a unit lower triangular matrix and U is an upper triangular matrix such that $A = LU$, then L and U are unique, i.e., if there $\tilde{L}$ is a unit lower triangular matrix and $\tilde{U}$ is an upper triangular matrix such that $A = \tilde{L}\tilde{U}$, then $L = \tilde{L}$ and $U = \tilde{U}$.

Symmetric Positive Definite Matrices

- **Fact 1:** If L is a unit lower triangular matrix and U is an upper triangular matrix such that $A = LU$, then L and U are unique, i.e., if there $\tilde{L}$ is a unit lower triangular matrix and $\tilde{U}$ is an upper triangular matrix such that $A = \tilde{L}\tilde{U}$, then $L = \tilde{L}$ and $U = \tilde{U}$.
- **Fact 2:** If $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite and $A = LU$, then the diagonal entries of U are positive.

Symmetric Positive Definite Matrices

- **Fact 1:** If L is a unit lower triangular matrix and U is an upper triangular matrix such that $A = LU$, then L and U are unique, i.e., if there $\tilde{L}$ is a unit lower triangular matrix and $\tilde{U}$ is an upper triangular matrix such that $A = \tilde{L}\tilde{U}$, then $L = \tilde{L}$ and $U = \tilde{U}$.
- **Fact 2:** If $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite and $A = LU$, then the diagonal entries of U are positive.
- Any upper triangular matrix U we can write as $U = D\tilde{U}$, where $\tilde{U}$ is unit upper triangular and

$$D = diag(u_{11}, \ldots, u_{nn}).$$

Symmetric Positive Definite Matrices

Thus,

$$A = LU = LD\tilde{U}.$$

On the other hand

$$A^T = (LD\tilde{U})^T = \tilde{U}^T D L^T.$$

Symmetric Positive Definite Matrices

Thus,

$$A = LU = LD\tilde{U}.$$

On the other hand

$$A^T = (LD\tilde{U})^T = \tilde{U}^T DL^T.$$

Using that $A = A^T$ and LU decomposition is uniqie

$$A = LU = \tilde{U}^T DL^T = \tilde{U}^T (DL^T) = \text{(lower unit triangular)} \times \text{(upper triangular)}.$$

Thus,

$$L = \tilde{U}^T \quad \text{and} \quad U = DL^T.$$

Symmetric Positive Definite Matrices

- Above we showed that if A is a symmetric positive definite matrix, then
$$A = LDL^T.$$

Symmetric Positive Definite Matrices

- Above we showed that if A is a symmetric positive definite matrix, then
$$A = LDL^T.$$
- Recall that
$$D = diag(u_{11}, \ldots, u_{nn})$$
has positive diagonal entries. So we can define
$$D^{1/2} = diag(\sqrt{u_{11}}, \ldots \sqrt{u_{nn}}).$$
Define $R := D^{1/2}L^T$, then
$$A = LDL^T = LD^{1/2}D^{1/2}L^T = R^T R : \qquad \text{Cholesky-decomposition}$$

Example

$$
\underbrace{\begin{pmatrix} 4 & -4 & 8 \\ -4 & 8 & -4 \\ 8 & -4 & 29 \end{pmatrix}}_{A} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & 1 & 1 \end{pmatrix}}_{L} \underbrace{\begin{pmatrix} 4 & -4 & 8 \\ 0 & 4 & 4 \\ 0 & 0 & 9 \end{pmatrix}}_{U}
$$

$$
= \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & 1 & 1 \end{pmatrix}}_{L} \underbrace{\begin{pmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 9 \end{pmatrix}}_{D} \underbrace{\begin{pmatrix} 1 & -1 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}}_{L^T}
$$

$$
= \underbrace{\begin{pmatrix} 2 & 0 & 0 \\ -2 & 2 & 0 \\ 4 & 2 & 3 \end{pmatrix}}_{R^T} \underbrace{\begin{pmatrix} 2 & -2 & 4 \\ 0 & 2 & 2 \\ 0 & 0 & 3 \end{pmatrix}}_{R}
$$

Symmetric Positive Definite Matrices

- A matrix $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite if and only if there exists a unit lower triangular matrix L and a positive definite diagonal matrix D such that $A = LDL^T$.
- A matrix $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite if and only if there exists an upper triangular matrix R with $r_{ii} > 0$, $i = 1, \ldots, n$, such that $A = R^T R$.

Symmetric Positive Definite Matrices

- A matrix $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite if and only if there exists a unit lower triangular matrix L and a positive definite diagonal matrix D such that $A = LDL^T$.
- A matrix $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite if and only if there exists an upper triangular matrix R with $r_{ii} > 0$, $i = 1, \ldots, n$, such that $A = R^T R$.
- So far we know how to compute LDL^T and $R^T R$ by first computing the LU-decomposition and then derive LDL^T or $R^T R$ from it. No savings. Not very useful.

Cholesky Decomposition

- If A is symmetric positive definite, then there exists an upper triangular matrix R with $r_{ii} > 0$, $i = 1, \ldots, n$, such that $A = R^T R$.
- From the matrix-matrix multiplication we have

$$a_{ij} = \sum_{k=1}^{n} r_{ki} r_{kj} = \sum_{k=1}^{\min\{i,j\}} r_{ki} r_{kj}.$$

Cholesky Decomposition

- If A is symmetric positive definite, then there exists an upper triangular matrix R with $r_{ii} > 0$, $i = 1, \dots, n$, such that $A = R^T R$.
- From the matrix-matrix multiplication we have

$$a_{ij} = \sum_{k=1}^{n} r_{ki} r_{kj} = \sum_{k=1}^{\min\{i,j\}} r_{ki} r_{kj}.$$

- Can easily derive the algorithm
  - Fix $i = 1$ and let $j = 1 : n$; $a_{1j} = r_{11} r_{1j}$. This implies
    $r_{11} = \sqrt{a_{11}}$, $j = 1$ and
    $r_{1j} = a_{1j}/r_{11}$, $j = 2 : n$.
  - Fix $i = 2$ and let $j = 2 : n$; $a_{2j} = r_{12} r_{1j} + r_{22} r_{2j}$. This implies
    $r22 = \sqrt{a_{22} - r_{12} r_{12}}$, $j = 2$, and
    $r2j = (a_{2j} - r_{12} r_{1j})/r_{22}$, $j = 3 : n$.
  - Continue this way to determine all rows of R successively.

Cholesky Decomposition

Input: $A \in \mathbb{R}^{n\times n}$, symmetric positive definite.
Output:$R \in \mathbb{R}^{n\times n}$

1. For $i = 1 : n$
2. If $a_{ii} - \sum_{k=1}^{i-1} r_{ik}^2 \leq 0$, return with an error message: A is not positive definite.
3. $r_{ii} = \left(a_{ii} - \sum_{k=1}^{i-1} r_{ik}^2\right)^{1/2}$
4. For $j = i + 1 : n$
5. $r_{ij} = \frac{1}{r_{ii}} \left(a_{ij} - \sum_{k=1}^{i-1} r_{ik} r_{kj}\right)^{1/2}$
6. End
7. End

Total number of flops: $n^3/3$ (compare to $2n^3/3$ for LU).
Matlab command syntax: $[R] = chol(A)$.

Tridiagonal Systems

- A matrix $A \in \mathbb{R}^{n\times n}$ is called a tridiagonal matrix if
$$a_{ij} = 0 \quad \text{for all} \quad i,j \quad \text{with} \quad |i-j| > 1.$$
-
$$A = \begin{pmatrix} d_1 & e_1 & & & \\ c_1 & d_2 & e_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{n-2} & d_{n-1} & e_{n-1} \\ & & & c_{n-1} & d_n \end{pmatrix}$$
- There are only about $3n$ entries in matrix A.

Tridiagonal Systems

- A matrix $A \in \mathbb{R}^{n\times n}$ is called a tridiagonal matrix if

$$a_{ij} = 0 \quad \text{for all} \quad i,j \quad \text{with} \quad |i-j| > 1.$$

- $$A = \begin{pmatrix} d_1 & e_1 & & & \\ c_1 & d_2 & e_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{n-2} & d_{n-1} & e_{n-1} \\ & & & c_{n-1} & d_n \end{pmatrix}$$

- There are only about $3n$ entries in matrix A.

- **Question:** How can we compute the LU-decomposition of A efficiently?

LU-Decomposition of Tridiagonal Systems

$$\underbrace{\begin{pmatrix} 1 & & & & \\ -\frac{c_1}{d_1} & 1 & & & \\ 0 & 0 & \ddots & & \\ & & \ddots & 1 & \\ & & & 0 & 1 \end{pmatrix}}_{=M_1}, \quad \underbrace{\begin{pmatrix} d_1 & e_1 & & & & \\ & \tilde{d}_2 & e_2 & & & \\ & c_2 & d_3 & e_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & c_{n-2} & d_{n-1} & e_{n-1} \\ & & & & c_{n-1} & d_n \end{pmatrix}}_{=M_1 A, \quad \text{where} \quad \tilde{d}_2 = d_2 - \frac{c_1}{d_1} e_1}$$

LU-Decomposition of Tridiagonal Systems

$$
\underbrace{\begin{pmatrix}
1 & & & & & \\
0 & 1 & & & & \\
 & -\frac{c_2}{\tilde{d}_2} & 1 & & & \\
 & 0 & 0 & \ddots & & \\
 & & & \ddots & 1 & \\
 & & & & 0 & 1
\end{pmatrix}}_{=M_2},
\underbrace{\begin{pmatrix}
d_1 & e_1 & & & & & \\
 & \tilde{d}_2 & e_2 & & & & \\
 & & \tilde{d}_3 & e_3 & & & \\
 & & c_3 & d_4 & e_4 & & \\
 & & & \ddots & \ddots & \ddots & \\
 & & & & c_{n-2} & d_{n-1} & e_{n-1} \\
 & & & & & c_{n-1} & d_n
\end{pmatrix}}_{=M_2 M_1 A, \quad \text{where} \quad \tilde{d}_3 = d_3 - \frac{c_2}{\tilde{d}_2} e_2}
$$

LU-Decomposition of Tridiagonal Systems

After $n-1$ steps we obtain

$$A = \begin{pmatrix} 1 & & & & \\ \frac{c_1}{d_1} & 1 & & & \\ & \frac{c_2}{d_2} & 1 & & \\ & & \ddots & \ddots & \\ & & & \frac{c_{n-1}}{d_{n-1}} & 1 \end{pmatrix} \begin{pmatrix} d_1 & e_1 & & & \\ & \tilde{d}_2 & e_2 & & \\ & & \ddots & \ddots & \\ & & & \tilde{d}_{n-1} & e_{n-1} \\ & & & & \tilde{d}_n \end{pmatrix},$$

where $\tilde{d_{i+1}} = d_{i+1} - \frac{c_i}{\tilde{d}_i} e_i$, for $i = 1 : n-1$.

LU-Decomposition of Tridiagonal Systems

Input: $A \in \mathbb{R}^{n\times n}$, symmetric positive definite.
Output:$R \in \mathbb{R}^{n\times n}$

1. For $k = 1 : n-1$
2. If $d_k = 0$, stop
3. $c_k = c_k/d_k$
4. $d_{k+1} = d_{k+1} - c_k e_k$
5. End

This algorithm requires about $3n$ flops.

Application: Boundary Value Problems

- We want to compute a function $y(x)$ that satisfies the differential equation

$$\begin{aligned} -\alpha y''(x) + \beta y'(x) + \gamma y(x) &= f(x), \quad x \in (a,b), \\ y'(a) &= y_a, \\ y(b) &= y_b. \end{aligned}$$

  The system is called a two-point boundary value problem BVP.
- The slope of the unknown function y is specified at the left end-point a and the value of the unknown function y is specified at the right end-point b. Other boundary conditions are possible and will be considered later.
- The function f and the scalars α, β, γ, y_a, and y_b are given. We assume that $\alpha > 0$ and $\gamma \geq 0$.

Application: Boundary Value Problems

- The solution y of a BVP is rarely known analytically, especially if α, β, and γ are nonconstant. In most application, the solution has to be approximated by numerical methods. We use the finite difference method.

Application: Boundary Value Problems

- The solution y of a BVP is rarely known analytically, especially if α, β, and γ are nonconstant. In most application, the solution has to be approximated by numerical methods. We use the finite difference method.
- We select a grid

$$a = x_0 < x_1 < \cdots < x_n = b$$

with equidistant points

$$x_i = a + i\frac{b-a}{n} = a + i\Delta x$$

where the mesh size

$$\Delta x = \frac{b-a}{n}.$$

Application: Boundary Value Problems

- **Finite differences** At a point x we can approximate the derivative of a function $g : \mathbb{R} \to \mathbb{R}$ by

$$g'(x) \approx \frac{g(x+h) - g(x-h)}{2h}, \tag{1}$$

Application: Boundary Value Problems

- **Finite differences** At a point x we can approximate the derivative of a function $g : \mathbb{R} \to \mathbb{R}$ by

$$g'(x) \approx \frac{g(x+h) - g(x-h)}{2h}, \tag{1}$$

- If we apply the approximation (1) with $h = \frac{\Delta x}{2}$ to $g(x) = y'(x)$ we obtain

$$-y''(x_i) = -(y'(x_i))' \approx -\frac{y'(x_i + \Delta x/2) - y'(x_i - \Delta x/2)}{\Delta x}.$$

  We approximate the derivatives $y'(x_i \pm \Delta x/2)$ using (1) with $h = \Delta x/2$ and $g(x) = y(x)$. This gives

Application: Boundary Value Problems

This gives

$$-y''(x) \approx -\frac{\frac{y(x_{i+1})-y(x_i)}{\Delta x} - \frac{y(x_i)-y(x_{i-1})}{\Delta x}}{\Delta x} = \frac{-y(x_{i+1}) + 2y(x_i) - y(x_{i-1})}{(\Delta x)^2}. \tag{2}$$

Similarly,

$$y'(x) \approx \frac{y(x_{i+1}) - y(x_{i-1})}{2\Delta x}. \tag{3}$$

Application: Boundary Value Problems

Thus in the BVP

$$
\begin{aligned}
-\alpha y''(x) + \beta y'(x) + \gamma y(x) = f(x), \quad x \in (a,b), \\
y'(a) = y_a, \\
y(b) = y_b.
\end{aligned}
$$

we can approximate at points x_i, $i = 0, 1, \ldots, n$ by

$$
\frac{-(\alpha + \frac{\Delta x}{2}\beta)y_{i-1} + (2\alpha + (\Delta x)^2\gamma)y_i - (\alpha - \frac{\Delta x}{2})\beta y_{i+1}}{(\Delta x)^2} = f(x_i) \quad (4)
$$

for $i = 0 : n-1$. The boundary conditions and lead to

$$
\frac{y_1 - y_{-1}}{2\Delta x} = y_a, \quad (5)
$$

i.e., $y..1 = y_{-1} - 2\Delta x y_a$. Finally, the other boundary condition yields

$$
y_n = y_b. \quad (6)
$$

Application: Boundary Value Problems

The equations (4)-(6) lead to tridiagonal system $A\vec{y} = \vec{f}$ with the unknowns $\vec{y} = (y_0, \dots, y_{n-1})^T$, where matrix A is

$$\begin{pmatrix} 2\alpha + (\Delta x)^2\gamma & -2\alpha & & & \\ -(\alpha + \frac{\Delta x}{2}\beta) & 2\alpha + (\Delta x)^2\gamma & -(\alpha - \frac{\Delta x}{2}\beta) & & \\ \ddots & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & -(\alpha + \frac{\Delta x}{2}\beta) & 2\alpha + (\Delta x)^2\gamma & -(\alpha - \frac{\Delta x}{2}\beta) \\ & & & -(\alpha + \frac{\Delta x}{2}\beta) & 2\alpha + (\Delta x)^2\gamma \end{pmatrix}$$

and vector

$$\vec{f} = \begin{pmatrix} (\Delta x)^2 f(x_0) - 2(\alpha + \frac{\Delta x}{2}\beta)\Delta x y_a \\ (\Delta x)^2 f(x_1) \\ \vdots \\ (\Delta x)^2 f(x_{n-2}) \\ (\Delta x)^2 f(x_{n-1}) + (\alpha - \frac{\Delta x}{2}\beta) y_b \end{pmatrix}$$

Matlab's spdiags

Given three arrays of length n, $c = [c_1; \ldots; c_n]$, $d = [d_1; \ldots; d_n]$; $e = [e_1; \ldots; e_n]$.
The Matlab command

$$A = spdiags([c, d, e], -1 : 1, n, n);$$

generates in a sparse matrix format a matrix

$$A = \begin{pmatrix} d_1 & e_2 & & & \\ c_1 & d_2 & e_3 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{n-2} & d_{n-1} & e_n \\ & & & c_{n-1} & d_n \end{pmatrix}$$

```
Example
>> c = [1; 2; 3]; d = [4; 5; 6]; e = [7; 8; 9];
>> A = spdiags([c, d, e], -1:1, 3,3);
>> full(A)
ans =
4 8 0
1 5 9
0 2 6
```

Summary. Strategies for solving $Ax = b$.

- If we need to solve the linear system once, just use backslash i.e. $x = A\backslash b$.

Summary. Strategies for solving $Ax = b$.

- If we need to solve the linear system once, just use backslash i.e. $x = A\backslash b$.
- If we need to solve the linear system many time, use LU decomposition, $[L, U] = lu(A)$. Then

$$x = U\backslash(L\backslash b).$$

Summary. Strategies for solving $Ax = b$.

- If we need to solve the linear system once, just use backslash i.e. $x = A\backslash b$.
- If we need to solve the linear system many time, use LU decomposition, $[L, U] = lu(A)$. Then

$$x = U\backslash(L\backslash b).$$

- If matrix A is symmetric positive definite use Cholesky decomposition instead, $R = chol(A)$. Then,

$$x = R\backslash(R'\backslash b).$$

Summary. Strategies for solving $Ax = b$.

- If we need to solve the linear system once, just use backslash i.e. $x = A\backslash b$.
- If we need to solve the linear system many time, use LU decomposition, $[L, U] = lu(A)$. Then
$$x = U\backslash(L\backslash b).$$
- If matrix A is symmetric positive definite use Cholesky decomposition instead, $R = chol(A)$. Then,
$$x = R\backslash(R'\backslash b).$$
- To find other variations of LU and Cholesky decompositions type $help\ lu$ and $help\ chol$

Summary. Strategies for solving $Ax = b$.

- If we need to solve the linear system once, just use backslash i.e. $x = A\backslash b$.
- If we need to solve the linear system many time, use LU decomposition, $[L, U] = lu(A)$. Then
$$x = U\backslash(L\backslash b).$$
- If matrix A is symmetric positive definite use Cholesky decomposition instead, $R = chol(A)$. Then,
$$x = R\backslash(R'\backslash b).$$
- To find other variations of LU and Cholesky decompositions type $help\ lu$ and $help\ chol$
- There are better algorithms available if matrix A has a special structure, like *tridiagonal, banded, etc.*