Lecture 8: Fast Linear Solvers (Part 3)
Cholesky Factorization

• Matrix $A$ is **symmetric** if $A = A^T$.

• Matrix $A$ is **positive definite** if for all $x \neq 0$, $x^T A x > 0$.

• A symmetric positive definite matrix $A$ has Cholesky factorization $A = LL^T$, where $L$ is a lower triangular matrix with positive diagonal entries.

• Example.
  
  $- A = A^T$ with $a_{ii} > 0$ and $a_{ii} > \sum_{j \neq i} |a_{ij}|$ is positive definite.
\[ A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{12} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & \cdots & l_{n1} \\ 0 & l_{22} & \cdots & l_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & l_{nn} \end{bmatrix} \]
In $2 \times 2$ matrix size case

\[
\begin{bmatrix}
a_{11} & a_{21} \\
a_{21} & a_{22}
\end{bmatrix}
= \begin{bmatrix}
l_{11} & 0 \\
l_{21} & l_{22}
\end{bmatrix}
\begin{bmatrix}
l_{11} & l_{21} \\
obzero & l_{22}
\end{bmatrix}
\]

\[l_{11} = \sqrt{a_{11}}; \quad l_{21} = a_{21}/l_{11}; \quad l_{22} = \sqrt{a_{22} - l_{21}^2}\]
Submatrix Cholesky Factorization Algorithm

\[
\text{for } k = 1 \text{ to } n \\
\quad a_{kk} = \sqrt{a_{kk}} \\
\quad \text{for } i = k + 1 \text{ to } n \\
\quad \quad a_{ik} = a_{ik}/a_{kk} \\
\quad \text{end} \\
\quad \text{for } j = k + 1 \text{ to } n \quad \quad \text{\(/ reduce submatrix} \\
\quad \quad \text{for } i = j \text{ to } n \\
\quad \quad \quad a_{ij} = a_{ij} - a_{ik}a_{jk} \\
\quad \quad \text{end} \\
\quad \text{end} \\
\text{end}
\]
Remark:
1. This is a variation of Gaussian Elimination algorithm.
2. Storage of matrix $A$ is used to hold matrix $L$.
3. Only lower triangle of $A$ is used (See $a_{ij} = a_{ij} - a_{ik}a_{jk}$).
4. Pivoting is not needed for stability.
5. About $n^3/6$ multiplications and about $n^3/6$ additions are required.

Data Access Pattern

- Read only
- Read and write
Column Cholesky Factorization Algorithm

\[
\begin{align*}
\text{for } j &= 1 \text{ to } n \\
\text{for } k &= 1 \text{ to } j - 1 \\
\text{for } i &= j \text{ to } n \\
\quad a_{ij} &= a_{ij} - a_{ik}a_{jk} \\
\end{align*}
\]

end

\[
\begin{align*}
\text{end} \\
\text{end} \\
\text{end} \\
a_{jj} &= \sqrt{a_{jj}} \\
\text{for } i &= j + 1 \text{ to } n \\
\quad a_{ij} &= a_{ij}/a_{jj} \\
\end{align*}
\]

end
Data Access Pattern

Read only

Read and write
Parallel Algorithm

- Parallel algorithms are similar to those for LU factorization.

References
QR Factorization and HouseHolder Transformation

**Theorem** Suppose that matrix $A$ is an $m \times n$ matrix with linearly independent columns, then $A$ can be factored as $A = QR$
where $Q$ is an $m \times n$ matrix with orthonormal columns and $R$ is an invertible $n \times n$ upper triangular matrix.
QR Factorization by Gram-Schmidt Process

Consider matrix \( A = [a_1 | a_2 | ... | a_n] \)

Then,

\[
\begin{align*}
    u_1 &= a_1, \\
    q_1 &= \frac{u_1}{||u_1||}, \\
    u_2 &= a_2 - (a_2 \cdot q_1)q_1, \\
    q_2 &= \frac{u_2}{||u_2||}, \\
    &\vdots \\
    u_k &= a_k - (a_k \cdot q_1)q_1 - \cdots - (a_k \cdot q_{k-1})q_{k-1}, \\
    q_k &= \frac{u_k}{||u_k||}
\end{align*}
\]

\[
A = [a_1 | a_2 | ... | a_n] = [q_1 | q_2 | ... | q_n]
\]

\[
\begin{bmatrix}
    a_1 \cdot q_1 & a_2 \cdot q_1 & \cdots & a_n \cdot q_1 \\
    0 & a_2 \cdot q_2 & \cdots & a_n \cdot q_2 \\
    \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & \cdots & a_n \cdot q_n
\end{bmatrix} = QR
\]
for $j = 1$ to $n$

$$u_j = a_j$$

for $i = 1$ to $j-1$

$$
\begin{align*}
    r_{ij} &= q_i^* a_j \quad \text{(Classical)} \\
    r_{ij} &= q_i^* u_j \quad \text{(Modified)}
\end{align*}
$$

$$u_j = u_j - r_{ij} q_i$$

endfor

$$r_{jj} = \|u_j\|_2$$

$$q_j = u_j / r_{jj}$$

endfor
Householder Transformation

Let \( v \in \mathbb{R}^n \) be a nonzero vector, the \( n \times n \) matrix
\[
H = I - 2 \frac{vv^T}{v^Tv}
\]
is called a Householder transformation (or reflector).

- Alternatively, let \( u = v / \| v \| \), \( H \) can be rewritten as
\[
H = I - 2uu^T.
\]

**Theorem.** A Householder transformation \( H \) is symmetric and orthogonal, so \( H = H^T = H^{-1} \).
1. Let vector $\mathbf{z}$ be perpendicular to $\mathbf{v}$.

\[
\left( I - 2 \frac{\mathbf{v} \mathbf{v}^T}{\mathbf{v}^T \mathbf{v}} \right) \mathbf{z} = \mathbf{z} - 2 \frac{\mathbf{v} (\mathbf{v}^T \mathbf{z})}{\mathbf{v}^T \mathbf{v}} = \mathbf{z}
\]

2. Let $\mathbf{u} = \frac{\mathbf{v}}{||\mathbf{v}||}$. Any vector $\mathbf{x}$ can be written as

\[
\mathbf{x} = \mathbf{z} + (\mathbf{u}^T \mathbf{x}) \mathbf{u}.
\]

\[
(I - 2 \mathbf{u} \mathbf{u}^T) \mathbf{x} = (I - 2 \mathbf{u} \mathbf{u}^T)(\mathbf{z} + (\mathbf{u}^T \mathbf{x}) \mathbf{u}) = \mathbf{z} - (\mathbf{u}^T \mathbf{x}) \mathbf{u}
\]
• Householder transformation is used to selectively zero out blocks of entries in vectors or columns of matrices.
• Householder is stable with respect to round-off error.
For a given a vector $\mathbf{x}$, find a Householder transformation, $H$, such that $H\mathbf{x} = a \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = a\mathbf{e}_1$

- Previous vector reflection (case 2) implies that vector $\mathbf{u}$ is in parallel with $\mathbf{x} - H\mathbf{x}$.
- Let $\mathbf{v} = \mathbf{x} - H\mathbf{x} = \mathbf{x} - a\mathbf{e}_1$, where $a = \pm||\mathbf{x}||$ (when the arithmetic is exact, sign does not matter).
- $\mathbf{u} = \mathbf{v}/||\mathbf{v}||$
- In the presence of round-off error, use $\mathbf{v} = \mathbf{x} + sign(x_1)||\mathbf{x}||\mathbf{e}_1$ to avoid catastrophic cancellation. Here $x_1$ is the first entry in the vector $\mathbf{x}$.
Thus in practice, $a = -sign(x_1)||\mathbf{x}||$
Algorithm for Computing the Householder Transformation

\[
x_m = \max\{|x_1|, |x_2|, \ldots, |x_n|\}
\]

\[
\text{for } k = 1 \text{ to } n
\]

\[
v_k = x_k / x_m
\]

end

\[
\alpha = \text{sign}(v_1)[v_1^2 + v_2^2 + \cdots + v_n^2]^{1/2}
\]

\[
v_1 = v_1 + \alpha
\]

\[
\alpha = -\alpha x_m
\]

\[
u = v / \|v\|
\]

Remark:
1. The computational complexity is \(O(n)\).
2. \(Hx\) is an \(O(n)\) operation compared to \(O(n^2)\) for a general matrix-vector product. \(Hx = x - 2u(u^T x)\).
QR Factorization by HouseHolder Transformation

Key idea:

Apply Householder transformation successively to columns of matrix to zero out sub-diagonal entries of each column.
• **Stage 1**

- Consider the first column of matrix $A$ and determine a Householder matrix $H_1$ so that

$$H_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

- Overwrite $A$ by $A_1$, where

$$A_1 = \begin{bmatrix} \alpha_1 & a_{12}^* & \cdots & a_{1n}^* \\ 0 & a_{22}^* & \vdots & a_{2n}^* \\ \vdots & \vdots & \vdots & \vdots \\ 0 & a_{n2}^* & \cdots & a_{nn}^* \end{bmatrix} = H_1 A$$

Denote vector which defines $H_1$ vector $v_1$. $u_1 = v_1 / ||v_1||$

Remark: Column vectors

$$\begin{bmatrix} a_{12}^* \\ a_{22}^* \\ \vdots \\ a_{n2}^* \end{bmatrix} \ldots \begin{bmatrix} a_{1n}^* \\ a_{2n}^* \\ \vdots \\ a_{nn}^* \end{bmatrix}$$

should be computed by

$$Hx = x - 2u(u^T x)$$
• **Stage 2**
  
  – Consider the second column of the updated matrix $A \equiv A_1 = H_1A$ and take the part below the diagonal to determine a Householder matrix $H_2^*$ so that

\[
H_2^* \begin{bmatrix}
a_{22}^* \\
a_{32}^* \\
\vdots \\
a_{n2}^*
\end{bmatrix} = \begin{bmatrix}
\alpha_2 \\
0 \\
\vdots \\
0
\end{bmatrix}
\]

Remark: size of $H_2^*$ is $(n - 1) \times (n - 1)$.

Denote vector which defines $H_2^*$ vector $v_2$. $u_2 = v_2/||v_2||$

– Inflate $H_2^*$ to $H_2$ where

\[
H_2 = \begin{bmatrix}
1 & 0 \\
0 & H_2^*
\end{bmatrix}
\]

Overwrite $A$ by $A_2 \equiv H_2 A_1$
• Stage $k$

- Consider the $k$th column of the updated matrix $A$ and take the part below the diagonal to determine a Householder matrix $H_k^*$ so that

$$H_k^* \begin{bmatrix} a_{kk}^* \\ a_{k+1,k}^* \\ \vdots \\ a_{n,n}^* \end{bmatrix} = \begin{bmatrix} \alpha_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Remark: size of $H_k^*$ is $(n - k + 1) \times (n - k + 1)$.

Denote vector which defines $H_k^*$ vector $v_k$. $u_k = v_k/||v_k||$

- Inflate $H_k^*$ to $H_k$ where

$$H_k = \begin{bmatrix} I_{k-1} & 0 \\ 0 & H_k^* \end{bmatrix}, I_{k-1} \text{ is } (k - 1) \times (k - 1) \text{ identity matrix.}$$

Overwrite $A$ by $A_k \equiv H_k A_{k-1}$
• After \((n - 1)\) stages, matrix \(A\) is transformed into an upper triangular matrix \(R\).

\[
R = A_{n-1} = H_{n-1}A_{n-2} = \cdots = H_{n-1}H_{n-2} \cdots H_1A
\]

• Set \(Q^T = H_{n-1}H_{n-2} \cdots H_1\). Then \(Q^{-1} = Q^T\). Thus \(Q = H_1^T H_2^T \cdots H_{n-1}^T\)

• \(A = QR\)
Example. $A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}$.

Find $H_1$ such that $H_1 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ 0 \\ 0 \end{bmatrix}$.

By $a = -\text{sign}(x_1)||x||$, 
$\alpha_1 = -\sqrt{3} = -1.721$.

$u_1 = 0.8881, u_2 = u_3 = 0.3250$.

By $Hx = x - 2u(u^T x)$, $H_1 \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} -3.4641 \\ 0.3661 \\ 1.3661 \end{bmatrix}$

$H_1A = \begin{bmatrix} -1.721 & -3.4641 \\ 0 & 0.3661 \\ 0 & 1.3661 \end{bmatrix}$
Find $H_2^*$ and $H_2$, where

$$H_2^* \begin{bmatrix} 0.3661 \\ 1.3661 \end{bmatrix} = \begin{bmatrix} \alpha_2 \\ 0 \end{bmatrix}$$

So $\alpha_2 = -1.4143$, $u_2 = \begin{bmatrix} 0.7922 \\ 0.6096 \end{bmatrix}$

So $R = H_2 H_1 A = \begin{bmatrix} -1.7321 & -3.4641 \\ 0 & -1.4143 \\ 0 & 0 \end{bmatrix}$

$$Q = (H_2 H_1)^T = H_1^T H_2^T = \begin{bmatrix} -0.5774 & 0.7071 & -0.4082 \\ -0.5774 & 0 & 0.8165 \\ -0.5774 & -0.7071 & -0.4082 \end{bmatrix}$$
\[ \begin{align*}
A x &= b \\
\Rightarrow Q R x &= b \\
\Rightarrow Q^T Q R x &= Q^T b \\
\Rightarrow R x &= Q^T b
\end{align*} \]

Remark: Q rarely needs explicit construction
Parallel Householder QR

• Householder QR factorization is similar to Gaussian elimination for LU factorization.
  – Additions + multiplications: $O \left( \frac{4}{3} n^3 \right)$ for QR versus $O \left( \frac{2}{3} n^3 \right)$ for LU
• Forming Householder vector $\mathbf{v}_k$ is similar to computing multipliers in Gaussian elimination.
• Subsequent updating of remaining unreduced portion of matrix is similar to Gaussian elimination.
• Parallel Householder QR is similar to parallel LU. Householder vectors need to broadcast among columns of matrix.
References
