Lecture 5
Gram-Schmidt Orthogonalization

MIT 18.335J / 6.337J
Introduction to Numerical Methods

Per-Olof Persson (persson@mit.edu)
September 19, 2007

The Classical Gram-Schmidt Algorithm

- The projection $P_j$ can equivalently be written as
  \[ P_j = P_{\perp q_{j-1}} \cdots P_{\perp q_2} P_{\perp q_1} \]
  where (last lecture)
  \[ P_{\perp q} = I - q q^T \]
- $P_{\perp q}$ projects orthogonally onto the space orthogonal to $q$, and
  \[ \operatorname{rank}(P_{\perp q}) = m - 1 \]
- The Classical Gram-Schmidt algorithm computes an orthogonal vector by
  \[ v_j = P_j a_j \]
  while the Modified Gram-Schmidt algorithm uses
  \[ v_j = P_{\perp q_{j-1}} \cdots P_{\perp q_2} P_{\perp q_1} a_j \]

Example: Classical vs. Modified Gram-Schmidt

- Small modification of classical G-S gives modified G-S (but see next slide)
- Modified G-S is numerically stable (less sensitive to rounding errors)

\[
\begin{align*}
\text{Classical/Modified Gram-Schmidt} \\
\text{for } j = 1 \text{ to } n, & \quad v_j = a_j \\
\text{for } i = 1 \text{ to } j - 1, & \quad r_{ij} = q_i^T a_j \\
& \quad v_j = v_j - r_{ij} q_i \\
& \quad r_{jj} = \|v_j\| \\
& \quad q_j = v_j / r_{jj}
\end{align*}
\]

\[
\begin{align*}
\text{Classical vs. Modified Gram-Schmidt} \\
\text{for } j = 1 \text{ to } n, & \quad v_j = a_j \\
\text{for } i = 1 \text{ to } n, & \quad r_{ij} = q_i^T a_j \\
& \quad v_j = v_j - r_{ij} q_i \\
& \quad r_{jj} = \|v_j\| \\
& \quad q_j = v_j / r_{jj}
\end{align*}
\]

Implementation of Modified Gram-Schmidt

- In modified G-S, $P_{\perp q_i}$ can be applied to all $v_j$ as soon as $q_i$ is known
- Makes the inner loop iterations independent (like in classical G-S)

<table>
<thead>
<tr>
<th>Classical Gram-Schmidt</th>
<th>Modified Gram-Schmidt</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{for } j = 1 \text{ to } n \hspace{1cm} v_j = a_j \hspace{1cm} r_{ij} = q_i^T a_j \hspace{1cm} r_{jj} = |v_j| \hspace{1cm} q_j = v_j / r_{jj}</td>
<td>\text{for } i = 1 \text{ to } n \hspace{1cm} v_i = a_i \hspace{1cm} r_{ii} = |v_i| \hspace{1cm} q_i = v_i / r_{ii} \hspace{1cm} \text{for } j = i + 1 \text{ to } n \hspace{1cm} r_{ij} = q_i^T v_j \hspace{1cm} v_j = v_j - r_{ij} q_i</td>
</tr>
</tbody>
</table>

Example: Classical vs. Modified Gram-Schmidt

- Compare classical and modified G-S for the vectors
  \[ a_1 = (1, \epsilon, 0, 0)^T, \quad a_2 = (1, 0, \epsilon, 0)^T, \quad a_3 = (1, 0, 0, \epsilon)^T \]
  making the approximation $1 + \epsilon^2 \approx 1$
- Classical:
  \[
  \begin{align*}
  v_1 & \leftarrow (1, \epsilon, 0, 0)^T, \quad r_{11} = \sqrt{1 + \epsilon^2} \approx 1, \quad q_1 = v_1 / 1 = (1, \epsilon, 0, 0)^T \\
  v_2 & \leftarrow (1, 0, \epsilon, 0)^T, \quad r_{12} = q_1^T a_2 = 1, \quad v_2 \leftarrow v_2 - 1 q_1 = (0, -\epsilon, 0, \epsilon)^T \\
  r_{22} & = \sqrt{2}, \quad q_2 = v_2 / r_{22} = (0, -1, 1, 0)^T / \sqrt{2} \\
  v_3 & \leftarrow (1, 0, 0, \epsilon)^T, \quad r_{13} = q_1^T a_3 = 1, \quad v_3 \leftarrow v_3 - 1 q_1 = (0, -\epsilon, 0, \epsilon)^T \\
  r_{33} & = \sqrt{2}, \quad q_3 = v_3 / r_{33} = (0, -1, 0, 1)^T / \sqrt{2}
  \end{align*}
\]
Example: Classical vs. Modified Gram-Schmidt

- Modified:
  \[ v_1 \leftarrow (1, \epsilon, 0, 0)^T, \quad r_{11} = \sqrt{1 + \epsilon^2} \approx 1, \quad q_1 = v_1 / 1 = (1, \epsilon, 0, 0)^T \]
  \[ v_2 \leftarrow (1, 0, \epsilon, 0)^T, \quad r_{12} = q_1^T v_2 = 1, \quad v_2 \leftarrow v_2 - q_1 (0, -\epsilon, 0, 0)^T \]
  \[ r_{22} = \sqrt{2}\epsilon, \quad q_2 = v_2 / r_{22} = (0, -1, 1, 0)^T / \sqrt{2} \]
  \[ v_3 \leftarrow (1, 0, 0, \epsilon)^T, \quad r_{13} = q_1^T v_3 = 1, \quad v_3 \leftarrow v_3 - q_1 (0, -\epsilon, 0, 0)^T \]
  \[ r_{23} = q_2^T v_3 = \epsilon / \sqrt{2}, \quad v_4 \leftarrow v_3 - r_{23} q_2 = (0, -\epsilon / 2, -\epsilon / 2, \epsilon)^T \]
  \[ r_{33} = \sqrt{6}\epsilon / 2, \quad q_3 = v_3 / r_{33} = (0, -1, -1, 2)^T / \sqrt{6} \]

- Check Orthogonality:
  - Classical: \( q_2^T q_3 = (0, -1, 1, 0)(0, -1, 0, 1)^T / 2 = 1/2 \)
  - Modified: \( q_2^T q_3 = (0, -1, 1, 0)(0, -1, -1, 2)^T / \sqrt{12} = 0 \)

Operation Count - Modified G-S

- Example: Count all +, −, *, / in the Modified Gram-Schmidt algorithm (not just the leading term)

  (1) \( \text{for } i = 1 \text{ to } n \quad v_i = a_i \)
  
  (3) \( \text{for } i = 1 \text{ to } n \quad r_{ii} = ||v_i|| \quad m \text{ multiplications, } m - 1 \text{ additions} \)
  
  (4) \( q_i = v_i / r_{ii} \quad m \text{ divisions} \)
  
  (6) \( \text{for } j = i + 1 \text{ to } n \quad r_{ij} = q_i^* v_j \quad m \text{ multiplications, } m - 1 \text{ additions} \)
  
  (8) \( v_j = v_j - r_{ij} q_i \quad m \text{ multiplications, } m \text{ subtractions} \)

Operation Count - Modified G-S

- The total for each operation is
  \[
  #A = \sum_{i=1}^{n} \left( m - 1 + \sum_{j=i+1}^{n} m - 1 \right) = n(m - 1) + \sum_{i=1}^{n} (m - 1)(n - i) = \\
  = n(m - 1) + \frac{n(n - 1)(m - 1)}{2} = \frac{1}{2} n(n + 1)(m - 1) 
  \]
  
  \[
  #S = \sum_{i=1}^{n} \sum_{j=i+1}^{n} m = \sum_{i=1}^{n} m(n - i) = \frac{1}{2} mn(n - 1) 
  \]
  
  \[
  #M = \sum_{i=1}^{n} \frac{m + \sum_{j=i+1}^{n} 2m}{2} = mn + \sum_{i=1}^{n} 2m(n - i) = \\
  = mn + \frac{2mn(n - 1)}{2} = mn^2 
  \]
  
  \[
  #D = \sum_{i=1}^{n} m = mn 
  \]

Operation Count - Modified G-S

- and the total flop count is
  \[
  \frac{1}{2} n(n + 1)(m - 1) + \frac{1}{2} mn(n - 1) + mn^2 + mn = \\
  2mn^2 + mn - \frac{1}{2} n^2 - \frac{1}{2} n \sim 2mn^2 
  \]

- The symbol \( \sim \) indicates asymptotic value as \( m, n \to \infty \) (leading term)

- Easier to find just the leading term:
  - Most work done in lines (7) and (8), with \( 4m \) flops per iteration
  - Including the loops, the total becomes
    \[
    \sum_{i=1}^{n} \sum_{j=i+1}^{n} 4m = 4m \sum_{i=1}^{n} (n - i) \sim 4m \sum_{i=1}^{n} i = 2mn^2 
    \]