CS5321
Numerical Optimization

10 Least Squares Problem
Least-squares problems

- Linear least-squares problems
  - QR method

- Nonlinear least-squares problems
  - Gradient and Hessian of nonlinear LSP
  - Gauss--Newton method
  - Levenberg--Marquardt method
  - Methods for large residual problem
Example of linear least square

\( y = \beta_1 + \beta_2 x \) (from Wikipedia)

\[
\begin{align*}
\beta_1 + 1\beta_2 &= 6 \\
\beta_1 + 2\beta_2 &= 5 \\
\beta_1 + 3\beta_2 &= 7 \\
\beta_1 + 4\beta_2 &= 10
\end{align*}
\]

\[
[6 - (\beta_1 + 1\beta_2)]^2 + [5 - (\beta_1 + 2\beta_2)]^2 + [7 - (\beta_1 + 3\beta_2)]^2 + [10 - (\beta_1 + 4\beta_2)]^2
\]

\( \beta_1 = 3.5, \; \beta_2 = 1.4. \) The line is \( y = 3.5 + 1.4x \)
Linear least squares problems

- A linear least-squares problem is $f(x) = \frac{1}{2} ||Ax - y||^2$.
- It’s gradient is $\nabla f(x) = A^T(Ax - y)$
- The optimal solution is at $\nabla f(x) = 0$, $A^TAx = A^Ty$
  - $A^TAx = A^Ty$ is called the normal equation.
- Perform QR decomposition on matrix $A = QR$.
  
  $A^TAx = R^TQ^TQRx = R^TQ^Ty$
  - $R^T$ is invertible. The solution $x = R^{-1}Q^Ty$. 
Example of nonlinear LS

\[ \phi(x, t) = x_1 + tx_2 + t^2x_3 + x_4e^{-x_5t} \]

- Find \((x_1,x_2,x_3,x_4,x_5)\) to minimize \(\frac{1}{2} \sum_{j=1}^{m} [\phi(x, t_j) - y_j]^2\)

\textbf{Figure 10.1} Deviation between the model (10.7) (smooth curve) and the observed measurements are indicated by the vertical bars.
Gradient and Hessian of LSP

- The object function of least squares problem is
  \[ f(x) = \frac{1}{2} \sum_{j=1}^{m} r_j^2(x) \] where \( r_i \) are \( n \) variable functions.

- Define \( R(x) = \begin{pmatrix} r_1(x) \\ r_2(x) \\ \vdots \\ r_m(x) \end{pmatrix} \) The Jacobian \( J(x) = \begin{pmatrix} \nabla r_1(x)^T \\ \nabla r_2(x)^T \\ \vdots \\ \nabla r_m(x)^T \end{pmatrix} \)

- Gradient \( \nabla f(x) = \sum_{j=1}^{m} r_j(x) \nabla r_j(x) = J(x)^T R(x) \)

- Hessian \( \nabla^2 f(x) = J(x)^T J(x) + \sum_{j=1}^{m} r_j(x) \nabla^2 r_j(x)^T \)
Gauss-Newton method

\[ \nabla^2 f(x) = J(x)^T J(x) + \sum_{j=1}^{m} r_j(x) \nabla^2 r_j(x)^T \]

- Gauss-Newton uses the Hessian approximation
  \[ \nabla^2 f(x) \approx J(x)^T J(x) \]
  - It’s a good approximation if \( \|R\| \) is small.
  - This is the matrix of the normal equation
  - Usually with the line search technique
- Replace \( f(x) = \frac{1}{2} \sum_{j=1}^{m} r_j^2(x) \) with \( f(x) = \frac{1}{2} \|Jp + R\|^2 \)
Convergence of Gauss-Newton

- Suppose each $r_j$ is Lipschitz continuously differentiable in a neighborhood $\mathcal{N}$ of \{x | $f(x) \leq f(x_0)$\} and the Jacobians $J(x)$ satisfy $\|J(x)z\| \geq \gamma \|z\|$. Then the Gauss-Newton method, with $\alpha_k$ that satisfies the Wolfe conditions, has

$$\lim_{k \to \infty} J_k^T R_k = 0$$
Levenberg-Marquardt method

- Gauss-Newton + trust region

The problem becomes

\[ \min_p \frac{1}{2} \| Jp + R \|^2 \text{ subject to } \| p \| \leq \Delta_k \]

- Optimal condition: (recall that in chap 4)

\[
(J^T J + \lambda I)p = -J^T R \\
\lambda(\Delta - \| p \|) = 0
\]

- Equivalent linear least-square problem

\[
\min_p \frac{1}{2} \left\| \begin{pmatrix} J \\ \sqrt{\lambda} I \end{pmatrix} p + \begin{pmatrix} R \\ 0 \end{pmatrix} \right\|^2
\]
Convergence of Levenberg-Marquardt

- Suppose \( \mathcal{L} = \{x \mid f(x) \leq f(x_0)\} \) is bounded and each \( r_j \) is Lipschitz continuously differentiable in a neighborhood \( \mathcal{N} \) of \( \mathcal{L} \). Assume for each \( k \), the approximation solution \( p_k \) of the Levenberg-Marquardt method satisfies the inequality

\[
m_k(0) - m_k(p_k) \geq c_1 \| J_k^T r_k \| \min \left( \Delta_k, \frac{\| J_k^T r_k \|}{\| J_k^T J_k \|} \right)
\]

for some constant \( c_1 > 0 \), and \( \| p_k \| \leq \gamma \Delta_k \) for some \( \gamma > 1 \). Then \( \lim_{k \to \infty} J_k^T R_k = 0 \)
Large residual problem

$$\nabla^2 f(x) = J(x)^T J(x) + \sum_{j=1}^{m} r_j(x) \nabla^2 r_j(x)^T$$

- When the second term of the Hessian is large
  - Use quasi-Newton to approximate the second term
  - The secant equation of $\nabla^2 r_j(x)$ is
    $$\left(B_j\right)(x_{k+1} - x_k) = \nabla r_j(x_{k+1}) - \nabla r_j(x_k)$$
- The secant equation of the second term and the update formula (next slide)
\[ S_{k+1}(x_{k+1} - x_k) = \sum_{j=1}^{m} r_j(x_{k+1})(B_j)_{k+1}(x_{k+1} - x_k) \]

\[ = \sum_{j=1}^{m} r_j(x_{k+1})[\nabla r_j(x_{k+1}) - \nabla r_j(x_k)] \]

\[ = J_{k+1}^T R_{k+1} - J_k^T R_{k+1} \]

Dennis, Gay, Welsch update formula.

\[ S_{k+1} = S_k + \frac{(z - S_k s)y^T + y(z - S_k s)^T}{y^T s} - \frac{(z - S_k s)^T s}{(y^T s)^2} yy^T \]

\[ s = x_{k+1} - x_k \]

\[ y = J_{k+1}^T r_{k+1} - J_k^T r_k \]

\[ z = J_{k+1}^T r_{k+1} - J_k^T r_{k+1} \]