# SLAM: Nonlinear Least-Squares

# Robot Localization and Mapping
# 16-833

Michael Kaess

October 7+9, 2024

# Nonlinear Least-Squares

- On the board:
  - Linearization
  - Gradient descent
  - Gauss-Newton
  - Levenberg-Marquardt
  - Powell’s Dog-Leg

# Nonlinear -> Linear Least Squares

Taylor series expansion:

$$h_i(X_i) = h_i(X_i^0 + \Delta_i) \approx h_i(X_i^0) + H_i \Delta_i$$

Measurement Jacobian: $H_i \triangleq \left.\frac{\partial h_i(X_i)}{\partial X_i}\right|_{X_i^0}$

State update vector: $\Delta_i \triangleq X_i - X_i^0$

Linear least-squares problem:

$$\begin{aligned} \Delta^* &= \underset{\Delta}{\operatorname{argmin}} \sum_i \left\| h_i(X_i^0) + H_i \Delta_i - z_i \right\|_{\Sigma_i}^2 \\ &= \underset{\Delta}{\operatorname{argmin}} \sum_i \left\| H_i \Delta_i - \underbrace{\left\{ z_i - h_i(X_i^0) \right\}}_{\text{Prediction error}} \right\|_{\Sigma_i}^2 \end{aligned}$$

# Simplifying to Quadratic Form

Original term with Mahalanobis Distance:

$$\begin{aligned}\Delta^* &= \underset{\Delta}{\operatorname{argmin}} \sum_i \left\| h_i(X_i^0) + H_i\Delta_i - z_i \right\|_{\Sigma_i}^2 \\ &= \underset{\Delta}{\operatorname{argmin}} \sum_i \left\| H_i\Delta_i - \left\{ z_i - h_i(X_i^0) \right\} \right\|_{\Sigma_i}^2\end{aligned}$$

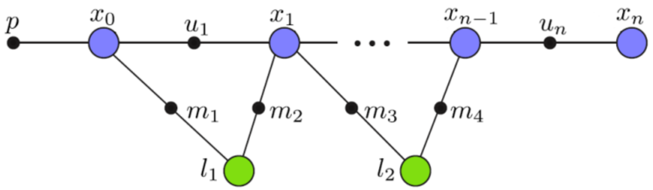
Simplification:

$$\begin{aligned}A_i &= \Sigma_i^{-1/2} H_i \\ b_i &= \Sigma_i^{-1/2} \left( z_i - h_i(X_i^0) \right)\end{aligned}$$

Quadratic form:

$$\begin{aligned}\Delta^* &= \underset{\Delta}{\operatorname{argmin}} \sum_i \| A_i\Delta_i - b_i \|_2^2 \\ &= \underset{\Delta}{\operatorname{argmin}} \| A\Delta - b \|_2^2\end{aligned}$$

# SLAM as a Sparse Least-Squares Problem



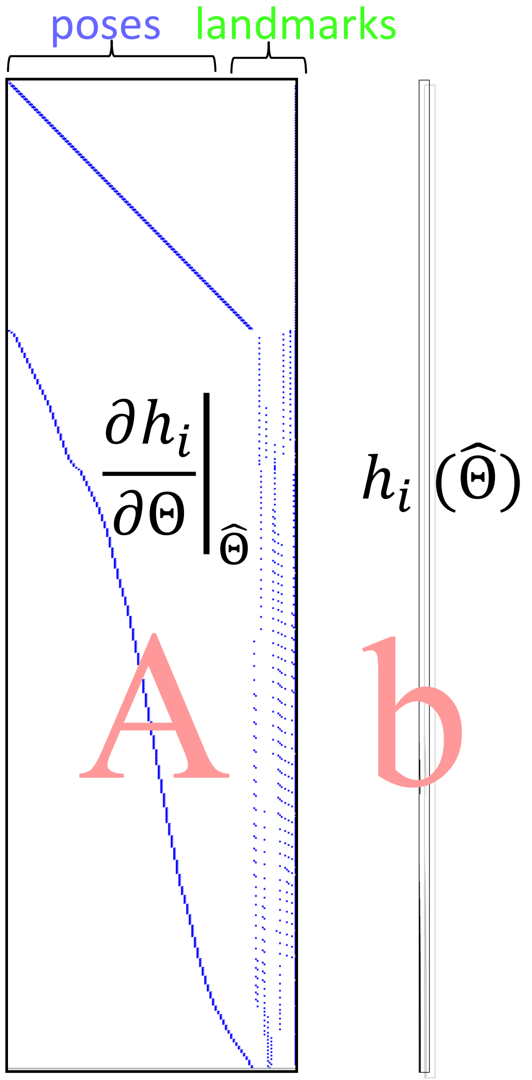

$$\mathrm{argmax}_{\Theta} \prod_{z \in Z} p(z|\Theta)$$

Gaussian noise

$$\mathrm{argmin}_{\Theta} \sum_i \|h_i(\Theta)\|^2$$

$$\mathrm{argmin}_{\theta} \|A\theta - b\|^2$$

Sparse measurement Jacobian



# Steepest Descent

Cost function:

$$g(X) \triangleq \sum_i \|h_i(X_i) - z_i\|^2_{\Sigma_i}$$

$$g(X) \approx \|A(X - X^t) - b\|_2^2$$

Steepest descent step:

$$\Delta_{sd} = -\alpha \, \nabla g\,(X)|_{X=X^t}$$

gradient: $\nabla g\,(X)|_{X=X^t} = -2A^T b$

# Gauss-Newton

Cost function:

$$g(X) \approx \|A(X - X^t) - b\|_2^2$$

Gauss-Newton step:

$$A^T A \Delta_{gn} = A^T b$$

# Levenberg-Marquardt

Levenberg:

$$(A^T A + \lambda I)\Delta_{lb} = A^T b$$

Levenberg-Marquardt:

$$\left(A^T A + \lambda \operatorname{diag}(A^T A)\right) \Delta_{lm} = A^T b$$

# Levenberg-Marquardt

**Algorithm 2.1** The Levenberg-Marquardt algorithm

**function** $\mathrm{LM}(g(), X^0)$ $\quad\triangleright$ quadratic cost function $g()$, $\triangleright$ initial estimate $X^0$

$\quad\lambda = 10^{-4}$

$\quad t = 0$

$\quad$**repeat**

$\qquad A, b \leftarrow$ linearize $g(X)$ at $X^t$

$\qquad \Delta \leftarrow \text{solve } \left(A^T A + \lambda\,\mathrm{diag}(A^T A)\right)\Delta = A^T b$

$\qquad$**if** $\mathrm{g}(X^t + \Delta) < \mathrm{g}(X^t)$ **then**

$\qquad\quad X^{t+1} = X^t + \Delta$ $\quad\triangleright$ accept update

$\qquad\quad \lambda \leftarrow \lambda / 10$

$\qquad$**else**

$\qquad\quad X^{t+1} = X^t$ $\quad\triangleright$ reject update

$\qquad\quad \lambda \leftarrow \lambda * 10$

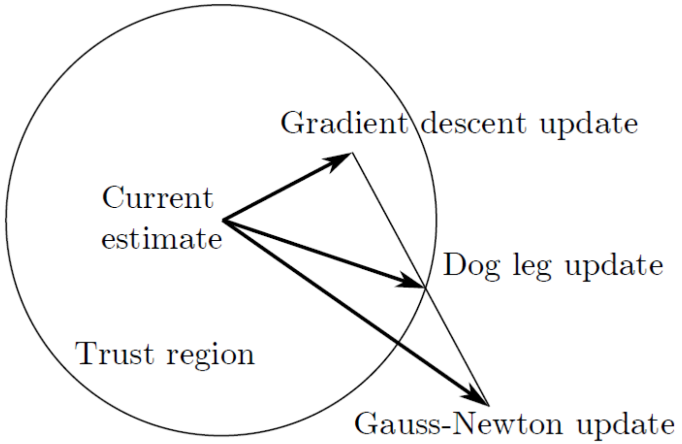
$\qquad t \leftarrow t + 1$

$\quad$**until** convergence

$\quad$**return** $X^t$ $\quad\triangleright$ return latest estimate

# Powell's Dog-Leg Algorithm

Key idea: Explicitly maintain a trust region




Gain ratio:

$$\rho = \frac{g(X^t) - g(X^t + \Delta)}{L(0) - L(\Delta)}$$

where $$L(\Delta) \;=\; A^T A \Delta \;-\; A^T b$$

# Summary

- Nonlinear similar to EKF, but:
  - Batch processing of all measurements and control inputs
  - Need initial estimate (EKF: prior state estimate)
  - Need to iterate to convergence
- Gradient descent (slow)
- Gauss Newton (need line search, otherwise brittle)
- Levenberg-Marquardt, Powell's Dog-Leg

- Next: How to deal with rotations in SLAM
  Factor Graphs for Robot Perception: 6 and appendix B