ECE276A: Sensing & Estimation in Robotics
Lecture 4: Unconstrained Optimization

Instructor:

Nikolay Atanasov: natanasov@ucsd.edu

Teaching Assistants:

Qiaojun Feng: qjfeng@ucsd.edu

Arash Asgharivaskasi: aasghari@eng.ucsd.edu

Ehsan Zobeidi: ezobeidi@ucsd.edu

Rishabh Jangir: rjangir@ucsd.edu

UC San Diego

JACOBS SCHOOL OF ENGINEERING

Electrical and Computer Engineering

Unconstrained Optimization

- Many problems we encounter in this course, lead to an **unconstrained optimization problem** over the Euclidean vector space $\mathbb{R}^d$:

$$\min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x})$$

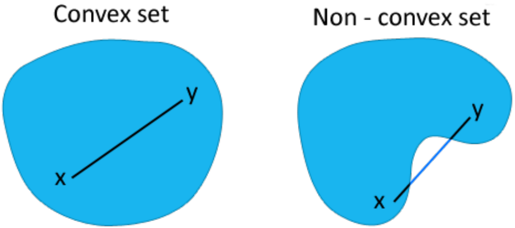
- A **global minimizer** $\mathbf{x}^* \in \mathbb{R}^d$ satisfies $f(\mathbf{x}^*) \le f(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^d$. The value $f(\mathbf{x}^*)$ is called **global minimum**.
- A **local minimizer** $\mathbf{x}^* \in \mathbb{R}^d$ satisfies $f(\mathbf{x}^*) \le f(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{N}(\mathbf{x}^*)$, where $\mathcal{N}(\mathbf{x}^*) \subset \mathbb{R}^d$ is a neighborhood around $\mathbf{x}^*$ (e.g., an open ball with small radius centered at $\mathbf{x}^*$). The value $f(\mathbf{x}^*)$ is called **local minimum**.
- The objective function $f : \mathbb{R}^d \mapsto \mathbb{R}$ is **differentiable** if the gradient:

$$\nabla f(\mathbf{x}) := \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f(\mathbf{x})}{\partial x_d} \end{bmatrix}^\top \in \mathbb{R}^d$$

  exists at each $\mathbf{x} \in \mathbb{R}^d$
- A **critical point** $\bar{\mathbf{x}} \in \mathbb{R}^d$ satisfies $\nabla f(\bar{\mathbf{x}}) = 0$ or $\nabla f(\bar{\mathbf{x}}) = \text{undefined}$
- All minimizers are critical points but not all critical points are minimizers. A critical point is either a local maximizer, a local minimizer, or neither (saddle point).

Convexity

- A set $\mathcal{D} \subseteq \mathbb{R}^d$ is **convex** if $\lambda \mathbf{x} + (1-\lambda)\mathbf{y} \in \mathcal{D}$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{D}$, $\lambda \in [0,1]$
- A convex set contains the line segment between any two points in it



- A function $f : \mathcal{D} \mapsto \mathbb{R}$ with $\mathcal{D} \subseteq \mathbb{R}^d$ is **convex** if:
  - $\mathcal{D}$ is a convex set
  - $f(\lambda \mathbf{x} + (1-\lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{D}$, $\lambda \in [0,1]$
- **First-order convexity condition**: a differentiable $f : \mathcal{D} \mapsto \mathbb{R}$ with convex $\mathcal{D}$ is convex iff $f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x})$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{D}$
- **Second-order convexity condition**: a twice-differentiable $f : \mathcal{D} \mapsto \mathbb{R}$ with convex $\mathcal{D}$ is convex iff $\nabla^2 f(\mathbf{x}) \succeq 0$ for all $\mathbf{x} \in \mathcal{D}$

Descent Direction

- Consider the **unconstrained optimization problem**:

$$\min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x})$$

Descent Direction Theorem

Suppose f is differentiable at $\bar{\mathbf{x}}$. If $\exists\ \delta\mathbf{x}$ such that $\nabla f(\bar{\mathbf{x}})^\top \delta\mathbf{x} < 0$, then $\exists\ \epsilon > 0$ such that $f(\bar{\mathbf{x}} + \alpha\delta\mathbf{x}) < f(\bar{\mathbf{x}})$ for all $\alpha \in (0, \epsilon)$.

- The vector $\delta\mathbf{x}$ is called a **descent direction**
- The theorem states that if a descent direction exists at $\bar{\mathbf{x}}$, then it is possible to move to a new point that has a lower f value
- **Steepest descent direction**: $\delta\mathbf{x} := -\frac{\nabla f(\bar{\mathbf{x}})}{\|\nabla f(\bar{\mathbf{x}})\|}$
- Based on this theorem, we can derive conditions for determining the optimality of $\bar{\mathbf{x}}$

Optimality Conditions

First-order Necessary Condition

Suppose f is differentiable at $\bar{\mathbf{x}}$. If $\bar{\mathbf{x}}$ is a local minimizer, then $\nabla f(\bar{\mathbf{x}}) = 0$.

Second-order Necessary Condition

Suppose f is twice-differentiable at $\bar{\mathbf{x}}$. If $\bar{\mathbf{x}}$ is a local minimizer, then $\nabla f(\bar{\mathbf{x}}) = 0$ and $\nabla^2 f(\bar{\mathbf{x}}) \succeq 0$.

Second-order Sufficient Condition

Suppose f is twice-differentiable at $\bar{\mathbf{x}}$. If $\nabla f(\bar{\mathbf{x}}) = 0$ and $\nabla^2 f(\bar{\mathbf{x}}) \succ 0$, then $\bar{\mathbf{x}}$ is a local minimizer.

Necessary and Sufficient Condition

Suppose f is differentiable at $\bar{\mathbf{x}}$. If f is **convex**, then $\bar{\mathbf{x}}$ is a global minimizer **if and only if** $\nabla f(\bar{\mathbf{x}}) = 0$.

Descent Optimization Methods

- A critical point of f can be obtained by solving $\nabla f(\mathbf{x}) = 0$ but an explicit solution may be difficult to derive

- **Descent methods**: iterative methods to obtain a solution of $\nabla f(\mathbf{x}) = 0$

- Given an initial guess $\mathbf{x}^{(k)}$, take a step of size $\alpha^{(k)} > 0$ along a descent direction $\delta\mathbf{x}^{(k)}$:
$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\delta\mathbf{x}^{(k)}$$

- Different methods differ in the way $\delta\mathbf{x}^{(k)}$ and $\alpha^{(k)}$ are chosen

- $\delta\mathbf{x}^{(k)}$ needs to be a descent direction: $\nabla f(\mathbf{x}^{(k)})^\top \delta\mathbf{x}^{(k)} < 0$, $\forall \mathbf{x}^{(k)} \neq \mathbf{x}^*$

- $\alpha^{(k)}$ needs to ensure sufficient decrease in f to guarantee convergence:
  - The best step size choice is $\alpha^{(k)} \in \arg\min_{\alpha>0} f(\mathbf{x}^{(k)} + \alpha\delta\mathbf{x}^{(k)})$
  - In practice, $\alpha^{(k)}$ is obtained via approximate **line search** methods

Gradient Descent (First-Order Method)

- **Idea**: $-\nabla f(\mathbf{x}^{(k)})$ points in the direction of steepest local descent
- **Gradient descent**: let $\delta\mathbf{x}^{(k)} := -\nabla f(\mathbf{x}^{(k)})$ and iterate:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha^{(k)}\nabla f(\mathbf{x}^{(k)})$$

- **Step size**: a good choice for $\alpha^{(k)}$ is $\frac{1}{L}$, where $L > 0$ is the Lipschitz constant of $\nabla f(\mathbf{x})$:

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}')\| \leq L\|\mathbf{x} - \mathbf{x}'\| \qquad \forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$$

Newton's Method (Second-Order Method)

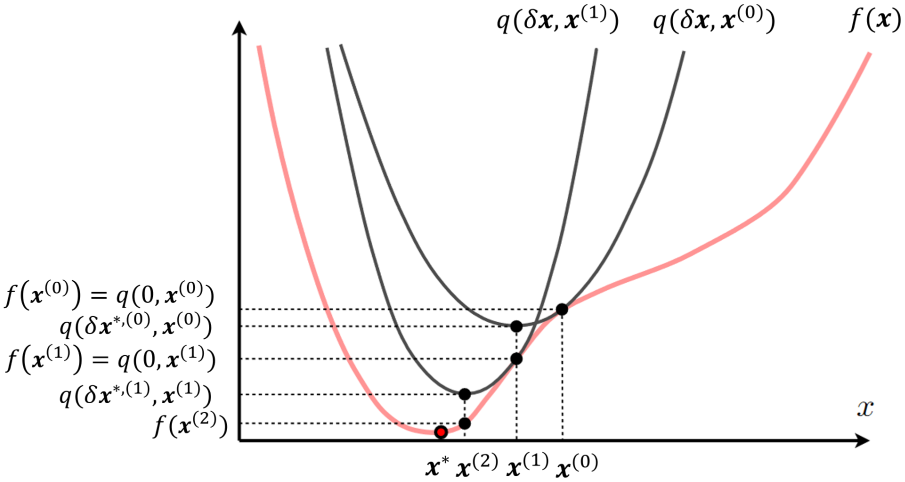
- **Newton's method**: iteratively approximates f by a quadratic function
- Since $\delta\mathbf{x}$ is a 'small' change to the initial guess $\mathbf{x}^{(k)}$, we can approximate f using a Taylor-series expansion:

$$f(\mathbf{x}^{(k)} + \delta\mathbf{x}) \approx f(\mathbf{x}^{(k)}) + \underbrace{\left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \bigg|_{\mathbf{x}=\mathbf{x}^{(k)}} \right)}_{\text{Gradient Transpose}} \delta\mathbf{x} + \frac{1}{2}\delta\mathbf{x}^\top \underbrace{\left(\frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^\top} \bigg|_{\mathbf{x}=\mathbf{x}^{(k)}} \right)}_{\text{Hessian}} \delta\mathbf{x}$$

$$=: \underbrace{q(\delta\mathbf{x}, \mathbf{x}^{(k)})}_{\text{quadratic function in } \delta\mathbf{x}}$$

- The symmetric Hessian matrix $\nabla^2 f(\mathbf{x}^{(k)})$ needs to be positive-definite for this method to work.

Newton's Method (Second-Order Method)



Newton's Method (Second-Order Method)

- Find $\delta\mathbf{x}$ that minimizes the quadratic approximation to $f(\mathbf{x}^{(k)} + \delta\mathbf{x})$
- Since this is an unconstrained optimization problem, $\delta\mathbf{x}$ can be determined by setting the derivative with respect to $\delta\mathbf{x}$ to zero:

$$0 = \frac{\partial q(\delta\mathbf{x}, \mathbf{x}^{(k)})}{\partial \delta\mathbf{x}} = \left(\left.\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right) + \delta\mathbf{x}^\top \left(\left.\frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x}\partial \mathbf{x}^\top}\right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right)$$

$$\Rightarrow \quad \left(\left.\frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x}\partial \mathbf{x}^\top}\right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right) \delta\mathbf{x} = -\left(\left.\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right)^\top$$

- The above is a linear system of equations and can be solved when the Hessian is invertible, i.e., $\nabla^2 f(\mathbf{x}^{(k)}) \succ 0$:

$$\delta\mathbf{x} = -\left[\nabla^2 f(\mathbf{x}^{(k)})\right]^{-1} \nabla f(\mathbf{x}^{(k)})$$

- **Newton's method**:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha^{(k)} \left[\nabla^2 f(\mathbf{x}^{(k)})\right]^{-1} \nabla f(\mathbf{x}^{(k)})$$

Newton's Method (Comments)

- Newton's method, like any other descent method, converges only to a **local** minimum
- **Damped Newton phase**: when the iterates are "far away" from the optimal point, the function value is decreased sublinearly, i.e., the step sizes $\alpha^{(k)}$ are small
- **Quadratic convergence phase**: when the iterates are "sufficiently close" to the optimum, full Newton steps are taken, i.e., $\alpha^{(k)} = 1$, and the function value converges quadratically to the optimum
- A **disadvantage** of Newton's method is the need to form the Hessian, which can be numerically ill-conditioned or very computationally expensive in high-dimensional problems

Gauss-Newton's Method

- **Gauss-Newton** is an approximation to Newton's method that avoids computing the Hessian. It is applicable when the objective function has the following quadratic form:

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{e}(\mathbf{x})^\top \mathbf{e}(\mathbf{x}) \qquad \mathbf{e}(\mathbf{x}) \in \mathbb{R}^m$$

- The Jacobian and Hessian matrices are:

$$\text{Jacobian:} \quad \left.\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(k)}} = \mathbf{e}(\mathbf{x}^{(k)})^\top \left(\left.\frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right)$$

$$\text{Hessian:} \quad \left.\frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^\top}\right|_{\mathbf{x}=\mathbf{x}^{(k)}} = \left(\left.\frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right)^\top \left(\left.\frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right) + \sum_{i=1}^{m} e_i(\mathbf{x}^{(k)}) \left(\left.\frac{\partial^2 e_i(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^\top}\right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right)$$

Gauss-Newton's Method

- Near the minimum of f, the second term in the Hessian is small relative to the first and the Hessian can be approximated according to:

$$\left.\frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^\top}\right|_{\mathbf{x}=\mathbf{x}^{(k)}} \approx \left(\left.\frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right)^\top \left(\left.\frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right)$$

- The above does not involve any second derivatives

- Setting the gradient of this new quadratic approximation of f with respect to $\delta\mathbf{x}$ to zero, leads to the system:

$$\left(\left.\frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right)^\top \left(\left.\frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right) \delta\mathbf{x} = - \left(\left.\frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right)^\top \mathbf{e}(\mathbf{x}^{(k)})$$

- **Gauss-Newton's method**:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \delta\mathbf{x}$$

Gauss-Newton's Method (Alternative Derivation)

- Another way to think about the Gauss-Newton method is to start with a Taylor expansion of $\mathbf{e}(\mathbf{x})$ instead of $f(\mathbf{x})$:

$$\mathbf{e}(\mathbf{x}^{(k)} + \delta\mathbf{x}) \approx \mathbf{e}(\mathbf{x}^{(k)}) + \left(\left. \frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right) \delta\mathbf{x}$$

- Substituting into f leads to:

$$f(\mathbf{x}^{(k)} + \delta\mathbf{x}) \approx \frac{1}{2} \left(\mathbf{e}(\mathbf{x}^{(k)}) + \left(\left. \frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right) \delta\mathbf{x} \right)^{\top} \left(\mathbf{e}(\mathbf{x}^{(k)}) + \left(\left. \frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right) \delta\mathbf{x} \right)$$

- Minimizing this with respect to $\delta\mathbf{x}$ leads to the same system as before:

$$\left(\left. \frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right)^{\top} \left(\left. \frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right) \delta\mathbf{x} = - \left(\left. \frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right)^{\top} \mathbf{e}(\mathbf{x}^{(k)})$$

Levenberg-Marquardt's Method

- The **Levenberg-Marquardt** modification to the Gauss-Newton method uses a positive diagonal matrix D to condition the Hessian approximation:

$$\left(\left(\left.\frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right)^\top \left(\left.\frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right) + \lambda D \right) \delta\mathbf{x} = - \left(\left.\frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{(k)}} \right)^\top \mathbf{e}(\mathbf{x}^{(k)})$$

- When $\lambda \geq 0$ is large, the descent vector $\delta\mathbf{x}$ corresponds to a very small step in the direction of steepest descent. This helps when the Hessian approximation is poor or poorly conditioned by providing a meaningful direction.

Levenberg-Marquardt's Method (Summary)

- An iterative optimization approach for the unconstrained problem:

$$\min_{\mathbf{x}} f(\mathbf{x}) := \frac{1}{2} \sum_j \mathbf{e}_j(\mathbf{x})^\top \mathbf{e}_j(\mathbf{x}) \qquad \mathbf{e}_j(\mathbf{x}) \in \mathbb{R}^{m_j},\ \mathbf{x} \in \mathbb{R}^n$$

- Given an initial guess $\mathbf{x}^{(k)}$, determine a descent direction $\delta\mathbf{x}$ by solving:

$$\left(\sum_j J_j(\mathbf{x}^{(k)})^\top J_j(\mathbf{x}^{(k)}) + \lambda D \right) \delta\mathbf{x} = - \left(\sum_j J_j(\mathbf{x}^{(k)})^\top \mathbf{e}_j(\mathbf{x}^{(k)}) \right)$$

  where $J_j(\mathbf{x}) := \frac{\partial \mathbf{e}_j(\mathbf{x})}{\partial \mathbf{x}} \in \mathbb{R}^{m_j \times n}$, $\lambda \geq 0$, $D \in \mathbb{R}^{n \times n}$ is a positive diagonal matrix, e.g., $D = \mathbf{diag}\left(\sum_j J_j(\mathbf{x}^{(k)})^\top J_j(\mathbf{x}^{(k)}) \right)$

- Obtain an updated estimate according to:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \delta\mathbf{x}$$

Unconstrained Optimization Example

- Let $f(\mathbf{x}) := \frac{1}{2}\sum_{j=1}^{n} \|A_j\mathbf{x} + b_j\|_2^2$ for $\mathbf{x} \in \mathbb{R}^d$ and assume $\sum_{j=1}^{n} A_j^\top A_j \succ 0$
- Solve the unconstrained optimization problem $\min_{\mathbf{x}} f(\mathbf{x})$ using:
  - The necessary and sufficient optimality condition for convex function f
  - Gradient descent
  - Newton's method
  - Gauss-Newton's method
- We will need $\nabla f(\mathbf{x})$ and $\nabla^2 f(\mathbf{x})$:

$$\frac{df(\mathbf{x})}{d\mathbf{x}} = \frac{1}{2}\sum_{j=1}^{n} \frac{d}{d\mathbf{x}}\|A_j\mathbf{x} + b_j\|_2^2 = \sum_{j=1}^{n} (A_j\mathbf{x} + b_j)^\top A_j$$

$$\nabla f(\mathbf{x}) = \frac{df(\mathbf{x})}{d\mathbf{x}}^\top = \left(\sum_{j=1}^{n} A_j^\top A_j\right)\mathbf{x} + \left(\sum_{j=1}^{n} A_j^\top b_j\right)$$

$$\nabla^2 f(\mathbf{x}) = \frac{d}{d\mathbf{x}}\nabla f(\mathbf{x}) = \sum_{j=1}^{n} A_j^\top A_j \succ 0$$

Necessary and Sufficient Optimality Condition

- Solve $\nabla f(\mathbf{x}) = 0$ for $\mathbf{x}$:

$$0 = \nabla f(\mathbf{x}) = \left(\sum_{j=1}^{n} A_j^\top A_j \right) \mathbf{x} + \left(\sum_{j=1}^{n} A_j^\top b_j \right)$$

$$\mathbf{x} = -\left(\sum_{j=1}^{n} A_j^\top A_j \right)^{-1} \left(\sum_{j=1}^{n} A_j^\top b_j \right)$$

- The solution above is unique since we assumed that $\sum_{j=1}^{n} A_j^\top A_j \succ 0$

Gradient Descent

- Start with an initial guess $\mathbf{x}^{(0)} = \mathbf{0}$
- At iteration k, gradient descent uses the descent direction $\delta\mathbf{x}^{(k)} = -\nabla f(\mathbf{x}^{(k)})$
- Given arbitary $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$, determine the Lipschitz constant of $\nabla f(\mathbf{x})$:

$$\|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\| = \left\| \left(\sum_{j=1}^{n} A_j^\top A_j \right) (\mathbf{x}_1 - \mathbf{x}_2) \right\| \leq \underbrace{\left\| \sum_{j=1}^{n} A_j^\top A_j \right\|}_{L} \|\mathbf{x}_1 - \mathbf{x}_2\|$$

- Choose step size $\alpha^{(k)} = \frac{1}{L}$ and iterate:

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \alpha^{(k)} \delta\mathbf{x}^{(k)} \\ &= \mathbf{x}^{(k)} - \frac{1}{L} \left(\sum_{j=1}^{n} A_j^\top A_j \right) \mathbf{x}^{(k)} - \frac{1}{L} \left(\sum_{j=1}^{n} A_j^\top b_j \right) \end{aligned}$$

Newton's Method

- Start with an initial guess $\mathbf{x}^{(0)} = \mathbf{0}$
- At iteration k, Newton's method uses the descent direction:

$$\begin{aligned}\delta\mathbf{x}^{(k)} &= -\left[\nabla^2 f(\mathbf{x}^{(k)})\right]^{-1} \nabla f(\mathbf{x}^{(k)}) \\ &= -\mathbf{x}^{(k)} - \left(\sum_{j=1}^{n} A_j^\top A_j\right)^{-1} \left(\sum_{j=1}^{n} A_j^\top b_j\right)\end{aligned}$$

  and updates the solution estimate via:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \delta\mathbf{x}^{(k)} = -\left(\sum_{j=1}^{n} A_j^\top A_j\right)^{-1} \left(\sum_{j=1}^{n} A_j^\top b_j\right)$$

- Note that for this problem, Newton's method converges in one iteration!

Gauss-Newton's Method

- $f(\mathbf{x})$ is of the form $\frac{1}{2}\sum_{j=1}^{n} \mathbf{e}_j(\mathbf{x})^\top \mathbf{e}_j(\mathbf{x})$ for $\mathbf{e}_j(\mathbf{x}) := A_j\mathbf{x} + b_j$
- The Jacobian of $\mathbf{e}_j(\mathbf{x})$ is $J_j(\mathbf{x}) = A_j$
- Start with an initial guess $\mathbf{x}^{(0)} = \mathbf{0}$
- At iteration k, Gauss-Newton's method uses the descent direction:

$$
\begin{aligned}
\delta\mathbf{x}^{(k)} &= -\left(\sum_{j=1}^{n} J_j(\mathbf{x}^{(k)})^\top J_j(\mathbf{x}^{(k)})\right)^{-1} \left(\sum_{j=1}^{n} J_j(\mathbf{x}^{(k)})^\top \mathbf{e}_j(\mathbf{x}^{(k)})\right) \\
&= -\left(\sum_{j=1}^{n} A_j^\top A_j\right)^{-1} \left(\sum_{j=1}^{n} A_j^\top (A_j\mathbf{x}^{(k)} + b_j)\right) \\
&= -\mathbf{x}^{(k)} - \left(\sum_{j=1}^{n} A_j^\top A_j\right)^{-1} \left(\sum_{j=1}^{n} A_j^\top b_j\right)
\end{aligned}
$$

- If $\alpha^{(k)} = 1$, in this problem, Gauss-Newton's method behaves exactly like Newton's method and coverges in one iteration!