# ECE276A: Sensing & Estimation in Robotics
# Lecture 2: Unconstrained Optimization

Nikolay Atanasov

natanasov@ucsd.edu

UC San Diego
JACOBS SCHOOL OF ENGINEERING
Electrical and Computer Engineering

# Outline

# Field

- A **field** is a set $\mathcal{F}$ with two binary operations, $+ : \mathcal{F} \times \mathcal{F} \mapsto \mathcal{F}$ (addition) and $\cdot : \mathcal{F} \times \mathcal{F} \mapsto \mathcal{F}$ (multiplication), which satisfy the following axioms:
  - **Associativity**: $a + (b + c) = (a + b) + c$ and $a(bc) = (ab)c$, $\forall a, b, c \in \mathcal{F}$
  - **Commutativity**: $a + b = b + a$ and $ab = ba$, $\forall a, b \in \mathcal{F}$
  - **Identity**: $\exists 1, 0 \in F$ such that $a + 0 = a$ and $a1 = a$, $\forall a \in \mathcal{F}$
  - **Inverse**: $\forall a \in \mathcal{F}, \exists -a \in \mathcal{F}$ such that $a + (-a) = 0$
    $\forall a \in \mathcal{F} \setminus \{0\}, \exists a^{-1} \in \mathcal{F} \setminus \{0\}$ such that $aa^{-1} = 1$
  - **Distributivity**: $a(b + c) = (ab) + (ac)$, $\forall a, b, c \in \mathcal{F}$
- **Examples**: real numbers $\mathbb{R}$, complex numbers $\mathbb{C}$, rational numbers $\mathbb{Q}$

# Vector Space

- A **vector space** over a field $\mathcal{F}$ is a set $\mathcal{V}$ with two binary operations, $+ : \mathcal{V} \times \mathcal{V} \mapsto \mathcal{V}$ (addition) and $\cdot : \mathcal{F} \times \mathcal{V} \mapsto \mathcal{V}$ (scalar multiplication), which satisfy the following axioms:
  - **Associativity**: $\mathbf{x} + (\mathbf{y} + \mathbf{z}) = (\mathbf{x} + \mathbf{y}) + \mathbf{z}$, $\forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{V}$
  - **Compatibility**: $a(b\mathbf{x}) = (ab)\mathbf{x}$, $\forall a, b \in \mathcal{F}$ and $\forall \mathbf{x} \in \mathcal{V}$
  - **Commutativity**: $\mathbf{x} + \mathbf{y} = \mathbf{x} + \mathbf{y}$, $\forall \mathbf{x}, \mathbf{y} \in \mathcal{V}$
  - **Identity**: $\exists\ \mathbf{0} \in V$ and $1 \in \mathcal{F}$ such that $\mathbf{x} + \mathbf{0} = \mathbf{x}$ and $1\mathbf{x} = \mathbf{x}$, $\forall \mathbf{x} \in \mathcal{V}$
  - **Inverse**: $\forall \mathbf{x} \in \mathcal{V}, \exists {-\mathbf{x}} \in \mathcal{V}$ such that $\mathbf{x} + (-\mathbf{x}) = \mathbf{0}$
  - **Distributivity**: $a(\mathbf{x} + \mathbf{y}) = a\mathbf{x} + b\mathbf{y}$ and $(a + b)\mathbf{x} = a\mathbf{x} + b\mathbf{x}$, $\forall a, b \in \mathcal{F}$ and $\forall \mathbf{x}, \mathbf{y} \in \mathcal{V}$
- **Examples**: real vectors $\mathbb{R}^d$, complex vectors $\mathbb{C}^d$, rational vectors $\mathbb{Q}^d$, functions $\mathbb{R}^d \mapsto \mathbb{R}$

# Basis and Dimension

- A **basis** of a vector space $\mathcal{V}$ over a field $\mathcal{F}$ is a set $\mathcal{B} \subseteq \mathcal{V}$ that satisfies:
  - **linear independence**: for all finite $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\} \subseteq \mathcal{B}$, if $a_1\mathbf{x}_1 + \cdots + a_m\mathbf{x}_m = 0$ for some $a_1, \ldots, a_m \in \mathcal{F}$, then $a_1 = \cdots = a_m = 0$
  - $\mathcal{B}$ **spans** $\mathcal{V}$: $\forall \mathbf{x} \in \mathcal{V}$, $\exists\ \mathbf{x}_1, \ldots, \mathbf{x}_d \in \mathcal{B}$ and unique $a_1, \ldots, a_d \in \mathcal{F}$ such that $\mathbf{x} = a_1\mathbf{x}_1 + \cdots + a_d\mathbf{x}_d$
- The **dimension** $d$ of a vector space $\mathcal{V}$ is the cardinality of its bases

# Inner Product and Norm

- An **inner product** on a vector space $\mathcal{V}$ over a field $\mathcal{F}$ is a function $\langle \cdot, \cdot \rangle : \mathcal{V} \times \mathcal{V} \mapsto \mathcal{F}$ such that for all $a \in \mathcal{F}$ and all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{V}$:
  - $\langle a\mathbf{x}, \mathbf{y} \rangle = a\langle \mathbf{x}, \mathbf{y} \rangle$ (homogeneity)
  - $\langle \mathbf{x} + \mathbf{y}, \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{y}, \mathbf{z} \rangle$ (additivity)
  - $\langle \mathbf{x}, \mathbf{y} \rangle = \overline{\langle \mathbf{y}, \mathbf{x} \rangle}$ (conjugate symmetry)
  - $\langle \mathbf{x}, \mathbf{x} \rangle \geq 0$ (non-negativity)
  - $\langle \mathbf{x}, \mathbf{x} \rangle = 0$ iff $\mathbf{x} = \mathbf{0}$ (definiteness)
- A **norm** on a vector space $\mathcal{V}$ over a field $\mathcal{F}$ is a function $\| \cdot \| : \mathcal{V} \to \mathbb{R}$ such that for all $a \in \mathcal{F}$ and all $\mathbf{x}, \mathbf{y} \in \mathcal{V}$:
  - $\|a\mathbf{x}\| = |a|\|\mathbf{x}\|$ (absolute homogeneity)
  - $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ (triangle inequality)
  - $\|\mathbf{x}\| \geq 0$ (non-negativity)
  - $\|\mathbf{x}\| = 0$ iff $\mathbf{x} = 0$ (definiteness)

# Euclidean Vector Space

- A **Euclidean vector space** $\mathbb{R}^d$ is a vector space with finite dimension $d$ over the real numbers $\mathbb{R}$
- A **Euclidean vector** $\mathbf{x} \in \mathbb{R}^d$ is a collection of scalars $x_i \in \mathbb{R}$ for $i = 1, \ldots, d$ organized as a column:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$$

- The **transpose** of $\mathbf{x} \in \mathbb{R}^d$ is organized as a row: $\mathbf{x}^\top = \begin{bmatrix} x_1 & \cdots & x_d \end{bmatrix}$
- The **Euclidean inner product** between two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ is:

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\top \mathbf{y} = \sum_{i=1}^{d} x_i y_i$$

- The **Euclidean norm** of a vector $\mathbf{x} \in \mathbb{R}^d$ is $\|\mathbf{x}\|_2 := \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle} = \sqrt{\mathbf{x}^\top \mathbf{x}}$

# Matrices

- A real $m \times n$ **matrix** $A$ is a rectangular array of scalars $A_{ij} \in \mathbb{R}$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$
- The set $\mathbb{R}^{m \times n}$ of real $m \times n$ matrices is a vector space
- The entries of the **transpose** $A^\top \in \mathbb{R}^{n \times m}$ of a matrix $A \in \mathbb{R}^{m \times n}$ are $A_{ij}^\top = A_{ji}$. The transpose satisfies: $(AB)^\top = B^\top A^\top$
- The **trace** of a matrix $A \in \mathbb{R}^{n \times n}$ is the sum of its diagonal entries:

$$\operatorname{tr}(A) := \sum_{i=1}^{n} A_{ii} \qquad \operatorname{tr}(ABC) = \operatorname{tr}(BCA) = \operatorname{tr}(CAB)$$

- The **Frobenius inner product** between two matrices $X, Y \in \mathbb{R}^{m \times n}$ is:

$$\langle X, Y \rangle = \operatorname{tr}(X^\top Y)$$

- The **Frobenius norm** of a matrix $X \in \mathbb{R}^{m \times n}$ is: $\|X\|_F := \sqrt{\operatorname{tr}(X^\top X)}$

# Matrix Determinant and Inverse

- The **determinant** of a matrix $A \in \mathbb{R}^{n \times n}$ is:

$$\det(A) := \sum_{j=1}^{n} A_{ij}\mathbf{cof}_{ij}(A) \qquad \det(AB) = \det(A)\det(B) = \det(BA)$$

  where $\mathbf{cof}_{ij}(A)$ is the **cofactor** of the entry $A_{ij}$ and is equal to $(-1)^{i+j}$ times the determinant of the $(n-1) \times (n-1)$ submatrix that results when the $i^{th}$-row and $j^{th}$-col of $A$ are removed. This recursive definition uses the fact that the determinant of a scalar is the scalar itself.

- The **adjugate** is the transpose of the cofactor matrix:

$$\mathbf{adj}(A) := \mathbf{cof}(A)^\top$$

- The **inverse** $A^{-1}$ of $A$ exists iff $\det(A) \neq 0$ and satisfies:

$$A^{-1}A = I \qquad A^{-1} = \frac{\mathbf{adj}(A)}{\det(A)} \qquad (AB)^{-1} = B^{-1}A^{-1}$$

## Eigenvalue Decomposition

- For any $A \in \mathbb{R}^{n \times n}$, if there exists $\mathbf{q} \in \mathbb{C}^n \setminus \{\mathbf{0}\}$ and $\lambda \in \mathbb{C}$ such that:

$$A\mathbf{q} = \lambda \mathbf{q}$$

  then $\mathbf{q}$ is an **eigenvector** corresponding to the **eigenvalue** $\lambda$.

- The $n$ eigenvalues of $A \in \mathbb{R}^{n \times n}$ are the $n$ roots of the **characteristic polynomial** $p(\lambda)$ of $A$:

$$p(\lambda) := \det(\lambda I - A)$$

- A real matrix can have complex eigenvalues and eigenvectors, which appear in conjugate pairs.

- Eigenvectors are not unique since for any $c \in \mathbb{C} \setminus \{0\}$, $c\mathbf{q}$ is an eigenvector corresponding to the same eigenvalue.

# Eigenvalue Decomposition

- **Diagonalizable matrix**: $n$ linearly independent eigenvectors $\mathbf{q}_i$ can be found for $A \in \mathbb{R}^{n \times n}$: $A\mathbf{q}_i = \lambda_i \mathbf{q}_i$ for $i = 1, \ldots, n$

- If the eigenvalues $\lambda_i$ of $A$ are distinct, then $A$ is diagonalizable

- **Eigen decomposition**: if $A$ is diagonalizable, we can stack all $n$ equations $A\mathbf{q}_i = \lambda_i \mathbf{q}_i$ to obtain an eigen decomposition of $A$:

$$A = Q \Lambda Q^{-1}$$

- **Jordan decomposition**: any $A$ can be decomposed using an invertible matrix $Q$ of generalized eigenvectors and an upper-triangular matrix $J$:

$$A = QJQ^{-1}$$

- **Jordan form $J$ of $A$**: an upper-triangular block-diagonal matrix:

$$J = \mathrm{diag}(B(\lambda_1, m_1), \ldots, B(\lambda_k, m_k))$$

where $\lambda_1, \ldots, \lambda_k$ are the eigenvalues of $A$ and $m_1 + \cdots + m_k = n$.

$$B(\lambda, m) = \begin{bmatrix} \lambda & 1 & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & \lambda & 1 \\ 0 & 0 & 0 & \lambda \end{bmatrix}$$

## Singular Value Decomposition

- An eigen-decomposition does not exist for $A \in \mathbb{R}^{m \times n}$
- $A \in \mathbb{R}^{m \times n}$ with rank $r \leq \min\{m, n\}$ can be diagonalized by two orthogonal matrices $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ via **singular value decomposition**:

$$A = U \Sigma V^\top \qquad \Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_r & \\ & & & \end{bmatrix} \in \mathbb{R}^{m \times n}$$

- $U$ contains the $m$ orthogonal eigenvectors of the symmetric matrix $AA^\top \in \mathbb{R}^{m \times m}$ and satisfies $U^\top U = UU^\top = I$
- $V$ contains the $n$ orthogonal eigenvectors of the symmetric matrix $A^\top A \in \mathbb{R}^{n \times n}$ and satisfies $V^\top V = VV^\top = I$
- $\Sigma$ contains the singular values $\sigma_i = \sqrt{\lambda_i}$, equal to the square roots of the $r$ non-zero eigenvalues $\lambda_i$ of $AA^\top$ or $A^\top A$, on its diagonal
- If $A$ is normal ($A^\top A = AA^\top$), its singular values are related to its eigenvalues via $\sigma_i = |\lambda_i|$

# Matrix Pseudo Inverse

- The **pseudo-inverse** $A^\dagger \in \mathbb{R}^{n\times m}$ of $A \in \mathbb{R}^{m\times n}$ can be obtained from its SVD $A = U\Sigma V^\top$:

$$A^\dagger = V\Sigma^\dagger U^T \qquad \Sigma^\dagger = \begin{bmatrix} 1/\sigma_1 & & & \\ & \ddots & & \\ & & 1/\sigma_r & \\ & & & \end{bmatrix} \in \mathbb{R}^{n\times m}$$

- The pseudo-inverse $A^\dagger \in \mathbb{R}^{n\times m}$ satisfies the Moore-Penrose conditions:
  - $AA^\dagger A = A$
  - $A^\dagger AA^\dagger = A^\dagger$
  - $\left(AA^\dagger\right)^\top = AA^\dagger$
  - $\left(A^\dagger A\right)^\top = A^\dagger A$

# Linear System of Equations

- Consider the linear system of equations $A\mathbf{x} = \mathbf{b}$ for $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$, and $A \in \mathbb{R}^{m \times n}$ with SVD $A = U\Sigma V^\top$ and rank $r$
- The **column space** or **image** of $A$ is $im(A) \subseteq \mathbb{R}^m$ and is spanned by the $r$ columns of $U$ corresponding to non-zero singular values
- The **null space** or **kernel** of $A$ is $ker(A) \subseteq \mathbb{R}^n$ and is spanned by the $n - r$ columns of $V$ corresponding to zero singular values
- The **row space** or **co-image** of $A$ is $im(A^\top) \subseteq \mathbb{R}^n$ and is spanned by the $r$ columns of $V$ corresponding to non-zero singular values
- The **left null space** or **co-kernel** of $A$ is $ker(A^\top) \subseteq \mathbb{R}^m$ and is spanned by the $m - r$ columns of $U$ corresponding to zero singular values
- The **domain** of $A$ is $\mathbb{R}^n = ker(A) \oplus im(A^\top)$
- The **co-domain** of $A$ is $\mathbb{R}^m = ker(A^\top) \oplus im(A)$

# Solution of Linear System of Equations

- Consider the linear system of equations $A\mathbf{x} = \mathbf{b}$ for $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$, and $A \in \mathbb{R}^{m \times n}$ with SVD $A = U\Sigma V^\top$ and rank $r$
- If $\mathbf{b} \in im(A)$, i.e., $\mathbf{b}^\top \mathbf{v} = 0$ for all $\mathbf{v} \in ker(A^\top)$, then $A\mathbf{x} = \mathbf{b}$ has **one or infinitely many solutions** $\mathbf{x} = A^\dagger \mathbf{b} + (I - A^\dagger A)\mathbf{y}$ for any $\mathbf{y} \in \mathbb{R}^n$
- If $\mathbf{b} \notin im(A)$, then **no solution exists** and $\mathbf{x} = A^\dagger \mathbf{b}$ is an approximate solution with minimum $\|\mathbf{x}\|$ and $\|A\mathbf{x} - \mathbf{b}\|$ norms
- If $m = n = r$, then $A\mathbf{x} = \mathbf{b}$ has a **unique solution** $\mathbf{x} = A^\dagger \mathbf{b} = A^{-1}\mathbf{b}$

## Positive Semidefinite Matrices

- The product $\mathbf{x}^\top A\mathbf{x}$ for $A \in \mathbb{R}^{n\times n}$ and $\mathbf{x} \in \mathbb{R}^n$ is called a **quadratic form** and $A$ can be assumed **symmetric**, $A = A^\top$, because:

$$\frac{1}{2}\mathbf{x}^\top(A + A^\top)\mathbf{x} = \mathbf{x}^\top A\mathbf{x}, \qquad \forall \mathbf{x} \in \mathbb{R}^n$$

- A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is **positive semidefinite** if $\mathbf{x}^\top A\mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$.
- A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is **positive definite** if it is positive semidefinite and if $\mathbf{x}^\top A\mathbf{x} = 0$ implies $\mathbf{x} = 0$.
- All eigenvalues of a symmetric positive semidefinite matrix are non-negative.
- All eigenvalues of a symmetric positive definite matrix are positive.

# Matrix Derivatives (numerator layout)

- Derivatives of $\mathbf{y} \in \mathbb{R}^m$ and $Y \in \mathbb{R}^{m\times n}$ by scalar $x \in \mathbb{R}$:

$$\frac{d\mathbf{y}}{dx} = \begin{bmatrix} \frac{dy_1}{dx} \\ \vdots \\ \frac{dy_m}{dx} \end{bmatrix} \in \mathbb{R}^{m\times 1} \qquad \frac{dY}{dx} = \begin{bmatrix} \frac{dY_{11}}{dx} & \cdots & \frac{dY_{1n}}{dx} \\ \vdots & \ddots & \vdots \\ \frac{dY_{m1}}{dx} & \cdots & \frac{dY_{mn}}{dx} \end{bmatrix} \in \mathbb{R}^{m\times n}$$

- Derivatives of $y \in \mathbb{R}$ and $\mathbf{y} \in \mathbb{R}^m$ by vector $\mathbf{x} \in \mathbb{R}^p$:

$$\frac{dy}{d\mathbf{x}} = \underbrace{\begin{bmatrix} \frac{dy}{dx_1} & \cdots & \frac{dy}{dx_p} \end{bmatrix}}_{[\nabla_{\mathbf{x}} y]^\top \text{ (gradient transpose)}} \in \mathbb{R}^{1\times p} \qquad \frac{d\mathbf{y}}{d\mathbf{x}} = \underbrace{\begin{bmatrix} \frac{dy_1}{dx_1} & \cdots & \frac{dy_1}{dx_p} \\ \vdots & \ddots & \vdots \\ \frac{dy_m}{dx_1} & \cdots & \frac{dy_m}{dx_p} \end{bmatrix}}_{\text{Jacobian}} \in \mathbb{R}^{m\times p}$$

- Derivative of $y \in \mathbb{R}$ by matrix $X \in \mathbb{R}^{p\times q}$:

$$\frac{dy}{dX} = \begin{bmatrix} \frac{dy}{dX_{11}} & \cdots & \frac{dy}{dX_{p1}} \\ \vdots & \ddots & \vdots \\ \frac{dy}{dX_{1q}} & \cdots & \frac{dy}{dX_{pq}} \end{bmatrix} \in \mathbb{R}^{q\times p}$$

# Matrix Derivative Examples

- $\frac{d}{dX_{ij}} X = \mathbf{e}_i \mathbf{e}_j^\top$
- $\frac{d}{d\mathbf{x}} A\mathbf{x} = A$
- $\frac{d}{d\mathbf{x}} \mathbf{u}^\top \mathbf{v} = \mathbf{u}^\top \frac{d\mathbf{v}}{d\mathbf{x}} + \mathbf{v}^\top \frac{d\mathbf{u}}{d\mathbf{x}}$ (**product rule**)
- $\frac{d}{d\mathbf{x}} \mathbf{x}^\top A\mathbf{x} = \mathbf{x}^\top (A + A^\top)$
- $\frac{d}{dx} M^{-1}(x) = -M^{-1}(x) \frac{dM(x)}{dx} M^{-1}(x)$
- $\frac{d}{dX} \operatorname{tr}(AX^{-1}B) = -X^{-1}BAX^{-1}$
- $\frac{d}{dX} \log\det X = X^{-1}$

# Matrix Derivative Examples

- $\frac{d}{d\mathbf{x}} A\mathbf{x} = \begin{bmatrix} \frac{d}{dx_1}\sum_{j=1}^n A_{1j}x_j & \cdots & \frac{d}{dx_n}\sum_{j=1}^n A_{1j}x_j \\ \vdots & \ddots & \vdots \\ \frac{d}{dx_1}\sum_{j=1}^n A_{mj}x_j & \cdots & \frac{d}{dx_n}\sum_{j=1}^n A_{mj}x_j \end{bmatrix} = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mn} \end{bmatrix}$
- $\frac{d}{d\mathbf{x}} \mathbf{x}^\top A\mathbf{x} = \mathbf{x}^\top \frac{dA\mathbf{x}}{d\mathbf{x}} + \mathbf{x}^\top A^\top \frac{d\mathbf{x}}{d\mathbf{x}} = \mathbf{x}^\top (A + A^\top)$
- $M(x)M^{-1}(x) = I \quad \Rightarrow \quad 0 = \left[\frac{d}{dx} M(x)\right] M^{-1}(x) + M(x)\left[\frac{d}{dx} M^{-1}(x)\right]$
- $$\begin{aligned} \frac{d}{dX_{ij}} \operatorname{tr}(AX^{-1}B) &= \operatorname{tr}(A\frac{d}{dX_{ij}}X^{-1}B) = -\operatorname{tr}(AX^{-1}\mathbf{e}_i\mathbf{e}_j^\top X^{-1}B) \\ &= -\mathbf{e}_j^\top X^{-1}BAX^{-1}\mathbf{e}_i = -\mathbf{e}_i^\top \left(X^{-1}BAX^{-1}\right)^\top \mathbf{e}_j \end{aligned}$$
- $$\begin{aligned} \frac{d}{dX_{ij}} \log\det X &= \frac{1}{\det(X)} \frac{d}{dX_{ij}} \sum_{k=1}^n X_{ik}\mathbf{cof}_{ik}(X) \\ &= \frac{1}{\det(X)}\mathbf{cof}_{ij}(X) = \frac{1}{\det(X)}\mathbf{adj}_{ji}(X) = \mathbf{e}_j^\top X^{-1}\mathbf{e}_i \end{aligned}$$

# Outline

## Unconstrained Optimization

- **Unconstrained optimization problem** over 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 $f(\mathbf{x}_*) \leq f(\mathbf{x})$ 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 $f(\mathbf{x}_*) \leq f(\mathbf{x})$ 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 function $f : \mathbb{R}^d \mapsto \mathbb{R}$ is **differentiable** at $\mathbf{x} \in \mathbb{R}^d$ if its gradient exists:

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

- A **critical point** $\bar{\mathbf{x}} \in \mathbb{R}^d$ satisfies $\nabla f(\bar{\mathbf{x}}) = 0$ or $\nabla f(\bar{\mathbf{x}}) =$ undefined

- All minimizers are critical points but not all critical points are minimizers. A critical point is a local maximizer, a local minimizer, or neither (saddle point).

# 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 $\exists\ \delta\mathbf{x} \in \mathbb{R}^d$ 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 derive conditions for 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

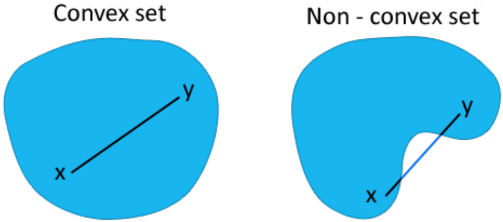
Suppose $f$ is twice-differentiable at $\bar{\mathbf{x}}$. If $\nabla f(\bar{\mathbf{x}}) = 0$ and $\nabla^2 f(\bar{\mathbf{x}}) \succ 0$, then $\bar{\mathbf{x}}$ is a local minimizer.

## 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** $\nabla f(\bar{\mathbf{x}}) = 0$.

# 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 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 obtain
- **Descent methods**: iterative methods to obtain a solution of $\nabla f(\mathbf{x}) = 0$
- Given initial guess $\mathbf{x}_k$, take step of size $\alpha_k > 0$ along 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

# Outline

Linear Algebra Review

Unconstrained Optimization

Gradient Descent

Newton's and Gauss-Newton's Methods

Example

## Gradient Descent (First-Order Method)

- **Idea**: $-\nabla f(\mathbf{x}_k)$ points in the direction of steepest 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 $\nabla f(\mathbf{x})$:

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

### Gradient Descent Convergence

Suppose $f$ is twice continuously differentiable with

$$mI \preceq \nabla^2 f(\mathbf{x}) \preceq LI, \qquad \forall \mathbf{x} \in \mathbb{R}^n.$$

The iterates $\mathbf{x}_k$ of gradient descent with step size $\alpha_k = \frac{1}{L}$ satisfy:

$$\|\nabla f(\mathbf{x}_k)\| \to 0 \qquad \text{and} \qquad \|\mathbf{x}_k - \mathbf{x}_*\| \to 0 \qquad \text{as } \; k \to \infty.$$

## Proof: Gradient Descent Convergence

- By the Mean Value Theorem for some $\mathbf{c}_k$ between $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$:

$$\nabla f(\mathbf{x}_{k+1}) = \nabla f(\mathbf{x}_k) + \nabla^2 f(\mathbf{c}_k)(\mathbf{x}_{k+1} - \mathbf{x}_k) = \nabla f(\mathbf{x}_k) - \alpha_k \nabla^2 f(\mathbf{c}_k)\nabla f(\mathbf{x}_k)$$

- Let $\lambda_i$ be the eigenvalues of $\nabla^2 f(\mathbf{c}_k)$ so that:

$$0 \leq 1 - \alpha_k L \leq 1 - \alpha_k \lambda_i \leq 1 - \alpha_k m$$

- This is sufficient to show that $\|\nabla f(\mathbf{x}_k)\| \to 0$ linearly:

$$\|\nabla f(\mathbf{x}_{k+1})\| \leq (1 - m/L)\|\nabla f(\mathbf{x}_k)\| \leq (1 - m/L)^{k+1}\|\nabla f(\mathbf{x}_0)\|$$

- By the Mean Value Theorem for some $\tilde{\mathbf{c}}_k$ between $\mathbf{x}_k$ and $\mathbf{x}_*$:

$$\mathbf{x}_{k+1} - \mathbf{x}_* = (\mathbf{x}_k - \mathbf{x}_*) - \alpha_k(\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_*)) = (\mathbf{x}_k - \mathbf{x}_*) - \alpha_k \nabla^2 f(\tilde{\mathbf{c}}_k)(\mathbf{x}_k - \mathbf{x}_*)$$

- Since $mI \preceq \nabla^2 f(\tilde{\mathbf{c}}_k) \preceq LI$:

$$\|\mathbf{x}_{k+1} - \mathbf{x}_*\| \leq (1 - m/L)\|\mathbf{x}_k - \mathbf{x}_*\| \leq (1 - m/L)^{k+1}\|\mathbf{x}_0 - \mathbf{x}_*\|$$

## Projected Gradient Descent

- **Constrained optimization problem** over a closed convex set $\mathcal{C} \subseteq \mathbb{R}^n$:

$$\min_{\mathbf{x} \in \mathcal{C}} f(\mathbf{x})$$

- **Constrained optimality condition**: for differentiable convex function $f$:

$$\mathbf{x}_* \in \arg\min_{\mathbf{x} \in \mathcal{C}} f(\mathbf{x}) \qquad \Leftrightarrow \qquad \langle \nabla f(\mathbf{x}_*), \mathbf{y} - \mathbf{x}_* \rangle \geq 0, \quad \forall \mathbf{y} \in \mathcal{C}$$

- **Euclidean projection onto $\mathcal{C}$**:

$$\Pi_{\mathcal{C}}(\mathbf{x}) := \arg\min_{\mathbf{y} \in \mathcal{C}} \|\mathbf{y} - \mathbf{x}\|$$

- **Projected gradient descent**:

$$\mathbf{x}_{k+1} = \Pi_{\mathcal{C}}(\mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)), \qquad \alpha > 0$$

# Projected Gradient Descent

## Projected Gradient Descent Convergence

Suppose $f$ is twice continuously differentiable with

$$mI \preceq \nabla^2 f(\mathbf{x}) \preceq LI, \qquad \forall \mathbf{x} \in \mathbb{R}^n.$$

The iterates $\mathbf{x}_k$ of projected gradient descent with step size $\alpha = \frac{1}{L}$ satisfy:

$$\|\mathbf{x}_{k+1} - \mathbf{x}_*\| \leq (1 - m/L)^{k+1} \|\mathbf{x}_0 - \mathbf{x}_*\|.$$

- The proof is based on:
  - Euclidean projection is non-expansive:

$$\|\Pi_{\mathcal{C}}(\mathbf{x}) - \Pi_{\mathcal{C}}(\mathbf{y})\| \leq \|\mathbf{x} - \mathbf{y}\|, \qquad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$$

  - Constrained optimizers are fixed points of the projected gradient descent operator with $\alpha > 0$:

$$\mathbf{x}_* \in \arg\min_{\mathbf{x} \in \mathcal{C}} f(\mathbf{x}) \qquad \Leftrightarrow \qquad \mathbf{x}_* = \Pi_{\mathcal{C}}(\mathbf{x}_* - \alpha \nabla f(\mathbf{x}_*))$$

# Outline

# Newton's Method (Second-Order Method)

- Consider the unconstrained optimization problem:

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

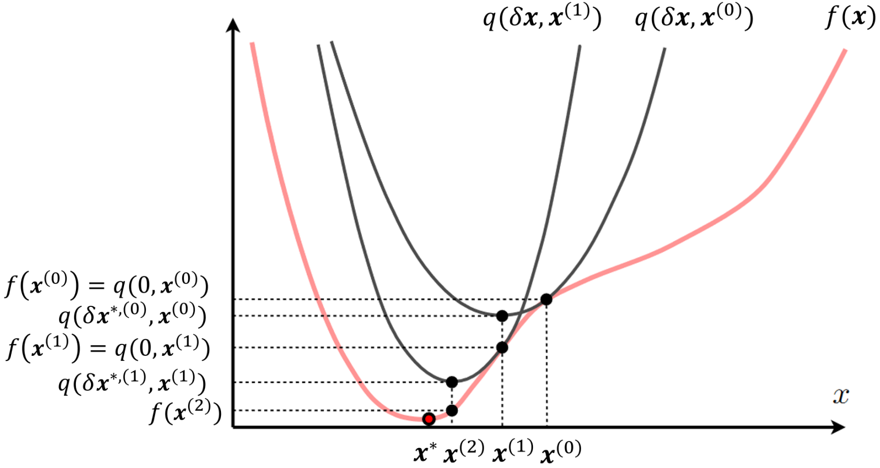
- **Newton's method** iteratively approximates $f$ by a quadratic function
- For a small change $\delta\mathbf{x}$ to $\mathbf{x}_k$, we can approximate $f$ using Taylor series:

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

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

- 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 $f(\mathbf{x}_k + \delta\mathbf{x})$:

$$\min_{\delta\mathbf{x}\in\mathbb{R}^d} q(\delta\mathbf{x}, \mathbf{x}_k)$$

- Since this is an unconstrained optimization problem, $\delta\mathbf{x}$ can be determined by setting the derivative of $q$ with respect to $\delta\mathbf{x}$ to zero:

$$0 = \frac{\partial q(\delta\mathbf{x}, \mathbf{x}_k)}{\partial \delta\mathbf{x}} = \nabla f(\mathbf{x}_k)^\top + \delta\mathbf{x}^\top \nabla^2 f(\mathbf{x}_k)$$

- This is a linear system of equations in $\delta\mathbf{x}$ and can be solved uniquely 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**:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \left[\nabla^2 f(\mathbf{x}_k)\right]^{-1} \nabla f(\mathbf{x}_k), \qquad \alpha_k > 0$$

# Newton's Method (Second-Order Method)

- Like other descent methods, Newton's method converges to a local minimum
- **Damped Newton phase**: when the iterates are "far away" from the optimum, 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 $\nabla^2 f(\mathbf{x}_k)$, which can be numerically ill-conditioned or 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:

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

- Derivative and Hessian:

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

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

## Gauss-Newton's Method

- Near the minimum of $f$, the second term in the Hessian is small relative to the first. The Hessian can be approximated without second derivatives:

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

- Approximation of $f(\mathbf{x}_k + \delta\mathbf{x})$:

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

- 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**:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \delta\mathbf{x}, \qquad \alpha_k > 0$$

## Levenberg-Marquardt's Method

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

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

- $\lambda D$ compensates for the missing Hessian term $\sum_{i=1}^{m} e_i(\mathbf{x}_k)\left(\left.\frac{\partial^2 e_i(\mathbf{x})}{\partial \mathbf{x}\partial \mathbf{x}^\top}\right|_{\mathbf{x}=\mathbf{x}_k}\right)$

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

## Gauss-Newton's Method

- 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:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \delta\mathbf{x}, \qquad \alpha_k > 0$$

# Outline

# 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 $\delta\mathbf{x}_k = -\nabla f(\mathbf{x}_k)$
- 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:

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

- With $\alpha_k = 1$, Newton's method converges in one iteration:

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

## 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:

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

- With $\alpha_k = 1$, in this problem, Gauss-Newton's method behaves like Newton's method and converges in one iteration