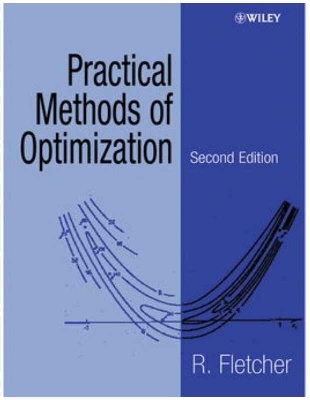
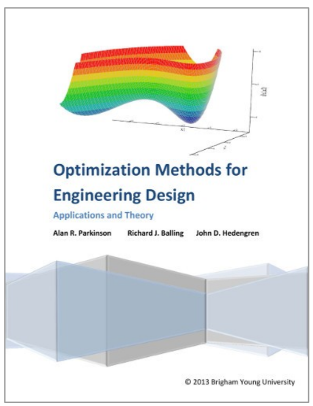
Outline

- Introduction
- **Part I: Basics of Mathematical Optimization**
  - Linear Least Squares
  - Nonlinear Optimization
- Part II: Basics of Computer Vision
  - Camera Model
  - Multi-Camera Model
  - Multi-Camera Calibration
- Part III: Depth Cameras
  - Passive Stereo
  - Structured Light Cameras
  - Time of Flight Cameras

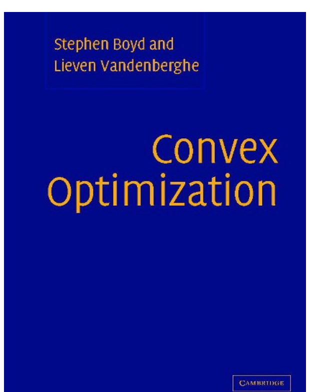
Literature on Mathematical Optimization



- Roger Fletcher:
  Practical Methods of Optimization.
  2nd Edition. Wiley, 2000.



- Alan Parkinson, Richard Balling, John Hedengren:
  Optimization Methods for Engineering Design. Applications and Theory.
  Brigham Young University, 2013.
- URL: http://apmonitor.com/me575/ (complete book)



- Stephen Boyd, Lieven Vandenberghe:
  Convex Optimization.
  Cambridge University Press, 2007.
- URL: http://web.stanford.edu/~boyd/cvxbook/
  (complete book, lecture slides, and exercise data)

Basics of Mathematical Optimization

General definition of optimization problem:

- Given is a cost function (objective function) $g : \mathbb{R}^n \to \mathbb{R}$
- **Aim:** Find parameters $\boldsymbol{x} \in \mathbb{R}^n$ that minimize g subject to constraints $h_1(\boldsymbol{x}) \geq 0, \ldots, h_l(\boldsymbol{x}) \geq 0$ for constraint functions $h_1, \ldots, h_l : \mathbb{R}^n \to \mathbb{R}$

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} g(\boldsymbol{x}) \;\; \text{s. t.} \;\; h_k(\boldsymbol{x}) \geq 0 \;\; \forall\, k = 1, \ldots, l$$

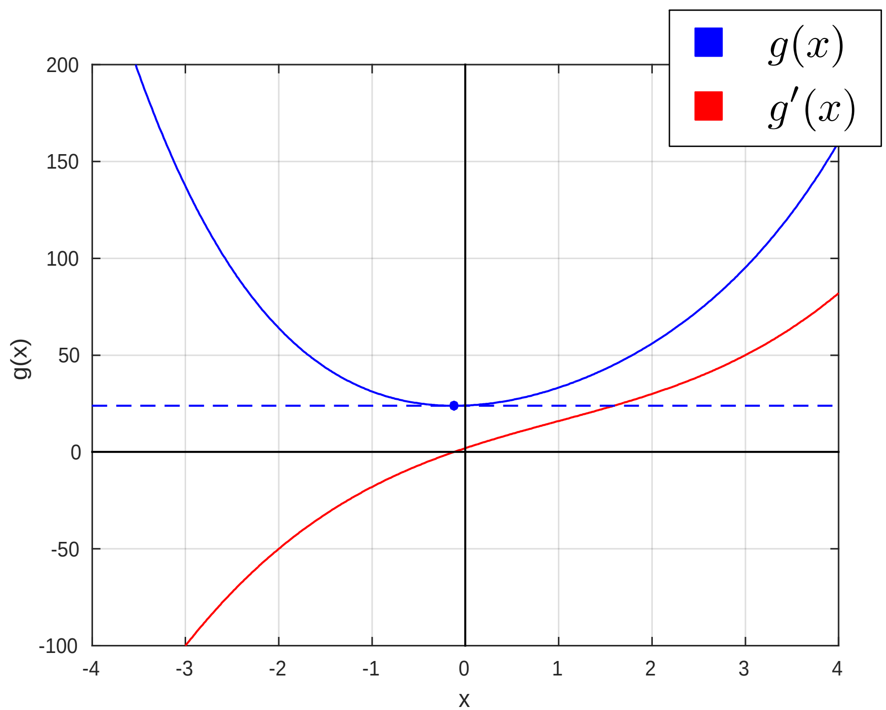
Note: g might have multiple local minima next to global minimum

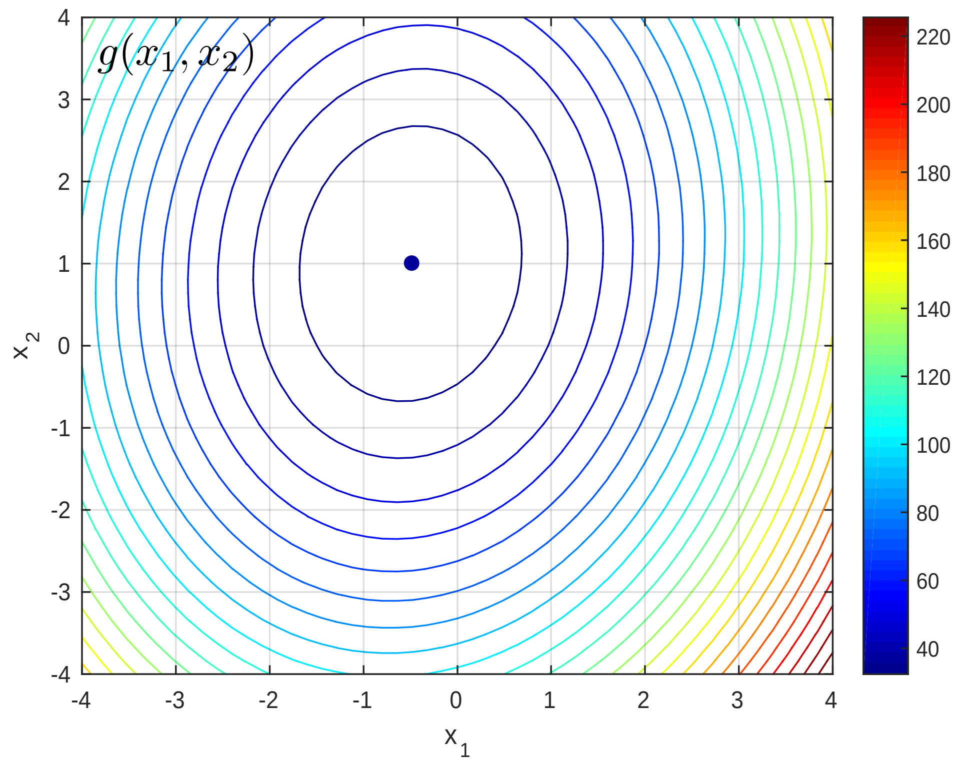
- Degree of freedom for parameters $\boldsymbol{x}$ is $n - l$
- Without constraints ($l = 0$): unconstrained optimization problem:

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} g(\boldsymbol{x})$$

Basics of Mathematical Optimization

- Necessary condition for (local) optimum of function $g : \mathbb{R}^n \to \mathbb{R}$





Single-parameter function:

$$g'(x) = \frac{\partial g(x)}{\partial x} = 0$$

i. e., slope at optimum is zero

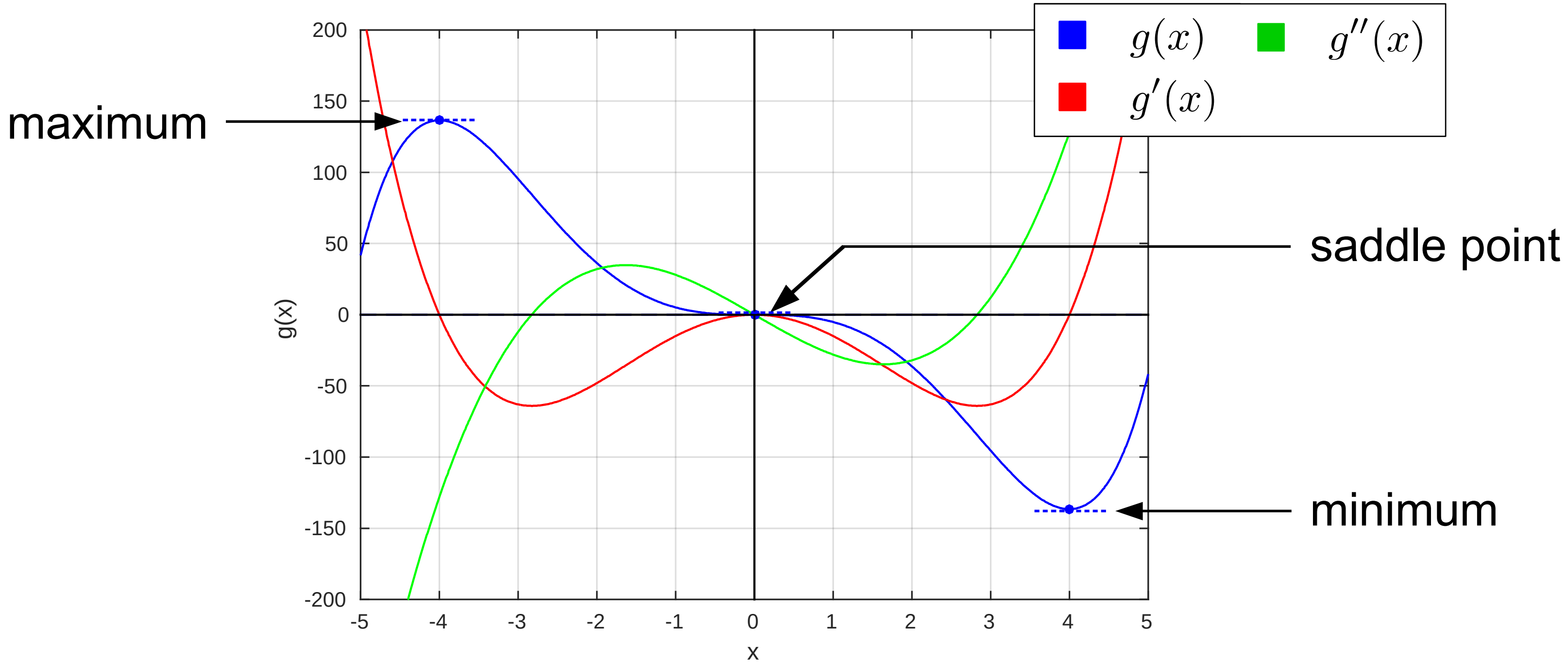
Multiple-parameter function:

$$\nabla g(\boldsymbol{x}) = \left(\frac{\partial g(\boldsymbol{x})}{\partial x_1}, \ldots, \frac{\partial g(\boldsymbol{x})}{\partial x_n} \right) = (0, \ldots, 0)$$

i. e., gradient is zero vector

Basics of Mathematical Optimization

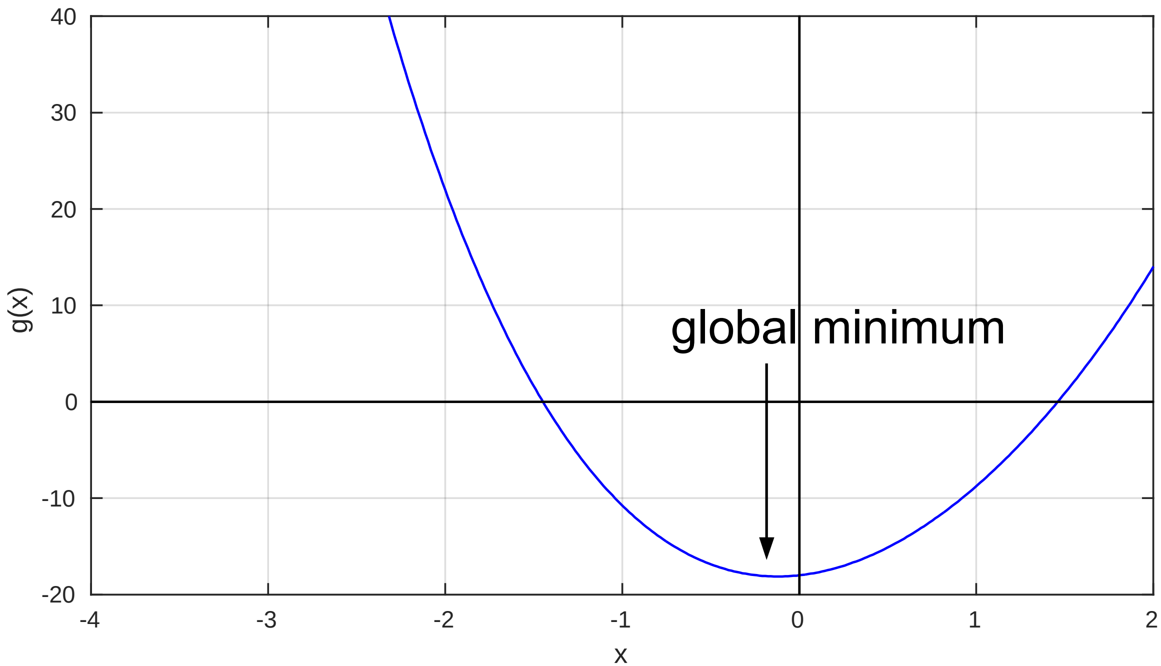
- Slope is zero for all critical points: Minima, maxima, and saddle points



- Necessary and sufficient condition for a minimal point:
  - Slope is zero: $g'(x) = 0$, curvature is positive: $g''(x) > 0$
  - Gradient is zero: $\nabla g(\boldsymbol{x}) = \mathbf{0}$, Hesse matrix $\mathbf{H}_g(\boldsymbol{x}) := \left(\frac{\partial^2 g(\boldsymbol{x})}{\partial x_i \partial x_j} \right)_{i,j=1,\dots,n}$ is positive-semidefinite (*i. e.:* $\boldsymbol{u}^T \mathbf{H}_g(\boldsymbol{x}) \boldsymbol{u} > 0 \;\; \forall \boldsymbol{u} \in \mathbb{R}^n, \boldsymbol{u} \neq \mathbf{0}$)

Basics of Mathematical Optimization

- **Analytic solution:** Find closed-form solution for $\nabla g(\boldsymbol{x}) = \mathbf{0}$ if possible, prune results with second derivative
- **Numeric solution:** Use iterative methods, *e. g.*, gradient descent or Newton methods (Gauss-Newton, Levenberg-Marquardt algorithm)
  - **But:** Which minimum is found depends on starting point here!
  - Convex functions (*e. g.*, quadratic functions) have unique minimum



Basics of Mathematical Optimization

Classes of optimization problems $\min_{\boldsymbol{x} \in \mathbb{R}^n} g(\boldsymbol{x}) \;\; \mathrm{s.\,t.} \;\; h_k(\boldsymbol{x}) \geq 0 \;\; \forall\, k = 1, \ldots, l$

- Linear optimization problem for linear functions $g, h_1, \ldots, h_l$,
  i. e., $g(a\boldsymbol{x} + b\boldsymbol{y}) = ag(\boldsymbol{x}) + bg(\boldsymbol{y})$ for all $a, b \in \mathbb{R}$, $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$
- Convex optimization problem for convex functions $g, h_1, \ldots, h_l$,
  i. e., $g(a\boldsymbol{x} + (1-a)\boldsymbol{y}) \leq ag(\boldsymbol{x}) + (1-a)g(\boldsymbol{y})$ for all $a \in [0,1]$, $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$
- Unconstrained problem for $l = 0$
- Least squares problem for $g(\boldsymbol{x}) = \|\boldsymbol{f}(\boldsymbol{x})\|^2$, $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ and $l = 0$
  with residual functions $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T$
- Linear least squares problem for $\boldsymbol{f}(\boldsymbol{x}) = \mathbf{A}\boldsymbol{x} - \boldsymbol{b}$, $\mathbf{A} \in \mathbb{R}^{m \times n}, \boldsymbol{b} \in \mathbb{R}^m$

Outline

- Introduction
- Part I: Basics of Mathematical Optimization
  - Linear Least Squares
  - Nonlinear Optimization
- Part II: Basics of Computer Vision
  - Camera Model
  - Multi-Camera Model
  - Multi-Camera Calibration
- Part III: Depth Cameras
  - Passive Stereo
  - Structured Light Cameras
  - Time of Flight Cameras

Least Squares

Common problem statement for least squares optimization:

- Given is a model function $b(\boldsymbol{a}, \boldsymbol{x})$ that maps input values $\boldsymbol{a} \in \mathbb{R}^k$ to output values $b \in \mathbb{R}$ which is parametrized by $\boldsymbol{x} \in \mathbb{R}^n$
- The residual functions $f_1, \ldots, f_m : \mathbb{R}^n \to \mathbb{R}$ describe the difference between measured output values and predicted values for given model parameters $\boldsymbol{x}$
- Given m measurements $b_1, \ldots, b_m \in \mathbb{R}$ for respective input vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m \in \mathbb{R}^k$, the i-th residual function is $f_i(\boldsymbol{x}) = b(\boldsymbol{a}_i, \boldsymbol{x}) - b_i$
- The objective function is given by $g(\boldsymbol{x}) = \sum_{i=1}^{m} f_i(\boldsymbol{x})^2 = \|\boldsymbol{f}(\boldsymbol{x})\|^2$

  i. e., the sum of squared residuals is minimized (“data fitting”):

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \sum_{i=1}^{m} (b(\boldsymbol{a}_i, \boldsymbol{x}) - b_i)^2$$

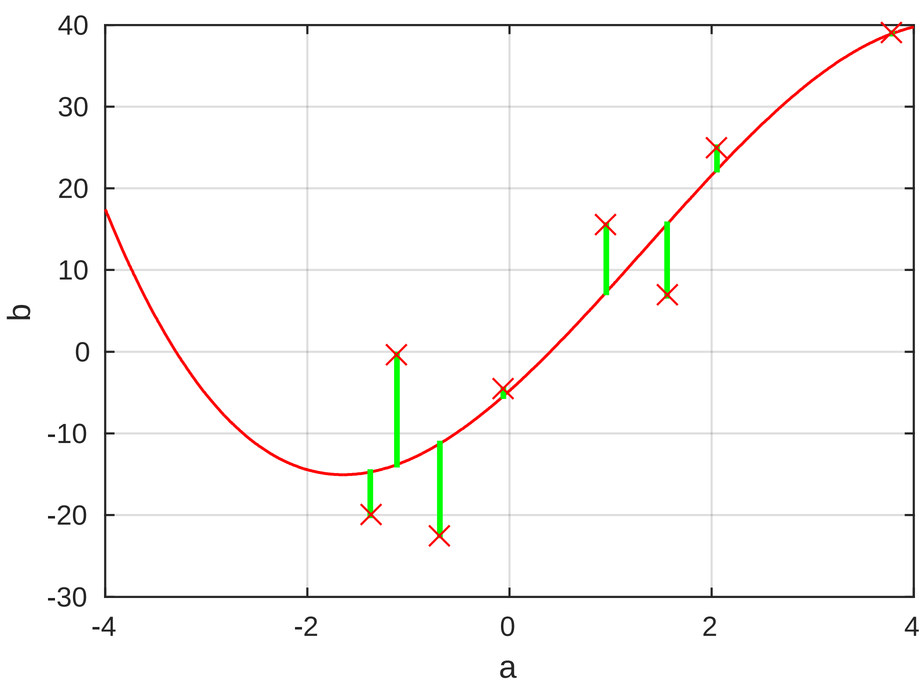
Examples for Least Squares Applications

- Line fitting, plane fitting (linear least squares), *e. g.:*

  Find 2D line parameters $\boldsymbol{x} \in \mathbb{R}^2$ to fit data points (a_i, b_i):
  $b(a_i, \boldsymbol{x}) = x_1 a_i + x_2 = b_i$
- Curve fitting, polynomial fitting (linear least squares), *e. g.:*

  Find polynomial coefficients $\boldsymbol{x} \in \mathbb{R}^n$ to fit data points (a_i, b_i):

  $$b(a_i, \boldsymbol{x}) = \sum_{j=1}^{n} a_i^{n-j} x_j = b_i$$
- Least squares problems are very common in Computer Vision



Linear Least Squares

Linear least squares optimization problem:

- Given is linear model function $b(\boldsymbol{a}, \boldsymbol{x}) = \boldsymbol{a}^T \boldsymbol{x}$
- Given are m residual functions $f_i = b(\boldsymbol{a}_i, \boldsymbol{x}) = \boldsymbol{a}_i^T \boldsymbol{x} - b_i$ defined by data points $(\boldsymbol{a}_i, b_i) \in \mathbb{R}^n \times \mathbb{R}$
- The objective function is given by $g(\boldsymbol{x}) = \sum_{i=1}^{m} f_i(\boldsymbol{x})^2 = \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2$ where $\mathbf{A} := (\boldsymbol{a}_1 \cdots \boldsymbol{a}_m)^T$

Solution:

- Necessary condition for minimum:
  $\nabla g(\boldsymbol{x}) = \mathbf{0} \;\Rightarrow\; \mathbf{A}^T \mathbf{A} \boldsymbol{x} = \mathbf{A}^T \boldsymbol{b} \;\Rightarrow\; \boldsymbol{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \boldsymbol{b}$
  derived from $g(\boldsymbol{x}) = \boldsymbol{x}^T \mathbf{A}^T \mathbf{A} \boldsymbol{x} - 2\boldsymbol{x}^T \mathbf{A}^T \boldsymbol{b} + \boldsymbol{b}^T \boldsymbol{b}$
- Minimum is unique because g is quadratic (= convex) function

Linear Least Squares

Error propagation:

- Assume ground truth values $\boldsymbol{b}^* = (b_1^*, \ldots, b_m^*)^T \in \mathbb{R}^m$ for input vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m \in \mathbb{R}^n$ and ground truth model parameter vector $\boldsymbol{x}^* \in \mathbb{R}^n$, *i. e.*, $\mathbf{A}\boldsymbol{x}^* = \boldsymbol{b}^*$
- Measured values are $\boldsymbol{b} = (b_1, \ldots, b_m)^T \in \mathbb{R}^m$ with measurement errors $\boldsymbol{\varepsilon}_b = (\varepsilon_{b_1}, \ldots, \varepsilon_{b_m})^T \in \mathbb{R}^m$, *i. e.*, $\boldsymbol{b} = \boldsymbol{b}^* + \boldsymbol{\varepsilon}_b$
- Linear error propagation:

$$\boldsymbol{x} = \underbrace{(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T}_{\mathbf{A}^\dagger} \boldsymbol{b} = \mathbf{A}^\dagger(\boldsymbol{b}^* + \boldsymbol{\varepsilon}_b) = \mathbf{A}^\dagger\mathbf{A}\boldsymbol{x}^* + \underbrace{\mathbf{A}^\dagger\boldsymbol{\varepsilon}_b}_{\boldsymbol{\varepsilon}_x} = \boldsymbol{x}^* + \boldsymbol{\varepsilon}_x$$

- For normal-distributed measurement errors $\boldsymbol{\varepsilon}_b \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_b)$:

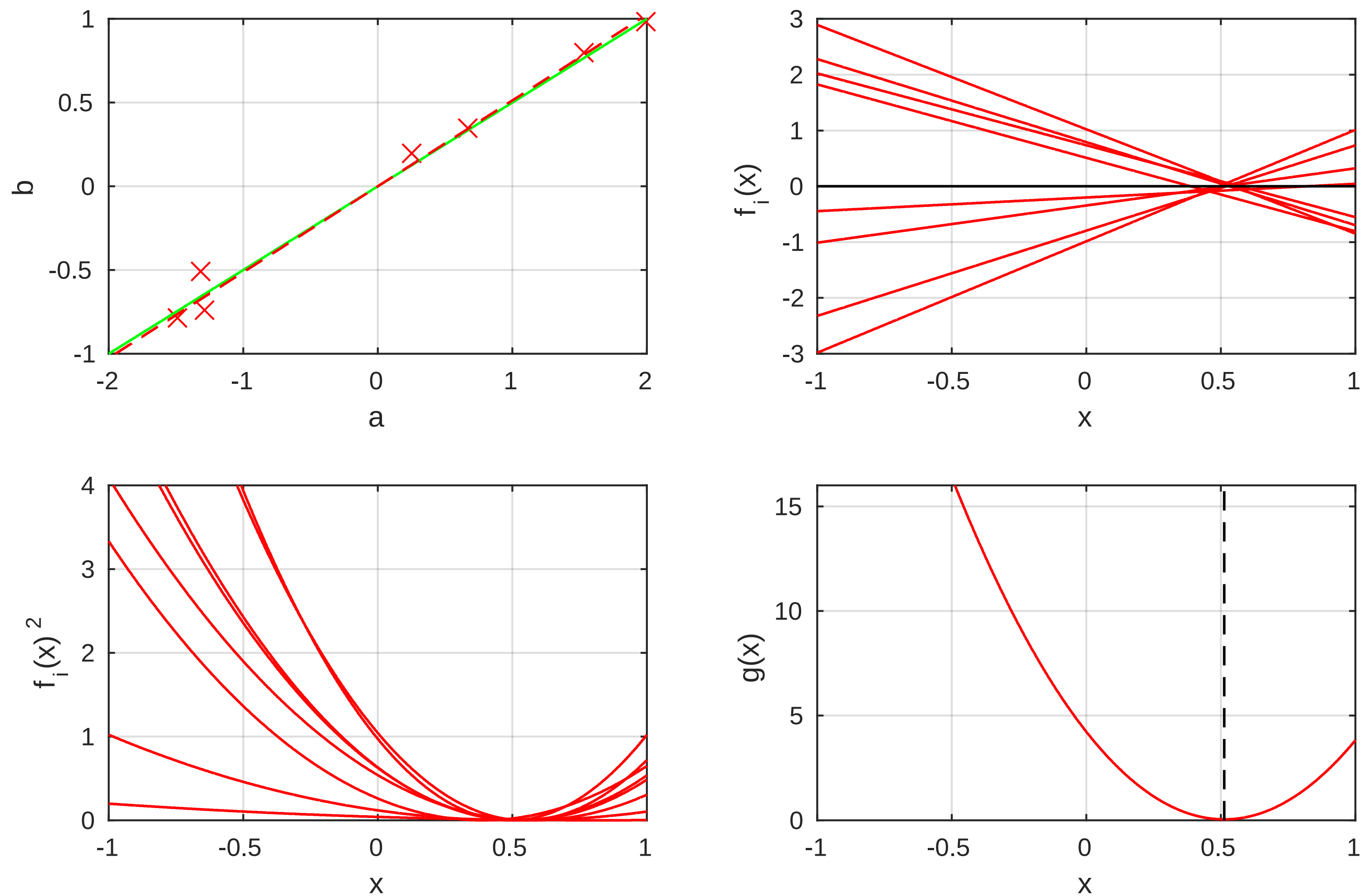
$$\Rightarrow \ \boldsymbol{\varepsilon}_x \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_x), \ \boldsymbol{\Sigma}_x = \mathbf{A}^\dagger\boldsymbol{\Sigma}_b(\mathbf{A}^\dagger)^T = (\mathbf{A}^T\mathbf{A})^{-1}(\mathbf{A}^T\boldsymbol{\Sigma}_b\mathbf{A})(\mathbf{A}^T\mathbf{A})^{-T}$$

Linear Least Squares Problem

- **Example:** Consider 1D-LLS problem with single parameter $x \in \mathbb{R}$
  - linear model function $b(a, x) = ax$
  - input data $a_1, \ldots, a_m \in \mathbb{R}$
  - measurements $b_1, \ldots, b_m \in \mathbb{R}$
  - residual functions are $f_i(x) := b(a_i, x) - b_i,\ i = 1, \ldots, m$
- **Task:** $\min\limits_{x \in \mathbb{R}} g(x)$ with $g(x) = \sum\limits_{i=1}^{n} f_i(x)^2 = \|\boldsymbol{a}x - \boldsymbol{b}\|^2$
- **Solution:** $g'(x) = 0 \ \Rightarrow\ \boldsymbol{a}^T \boldsymbol{a} x = \boldsymbol{a}^T \boldsymbol{b} \ \Rightarrow\ x = \frac{\boldsymbol{a}^T \boldsymbol{b}}{\boldsymbol{a}^T \boldsymbol{a}}$

Linear Least Squares Problem

- **Example:** Ground truth value is $x^* = 0.5$, $m = 8$ measurements b_i for inputs a_i, measurement error from normal distribution $\varepsilon_b \sim \mathcal{N}(0, 0.1)$



Outline

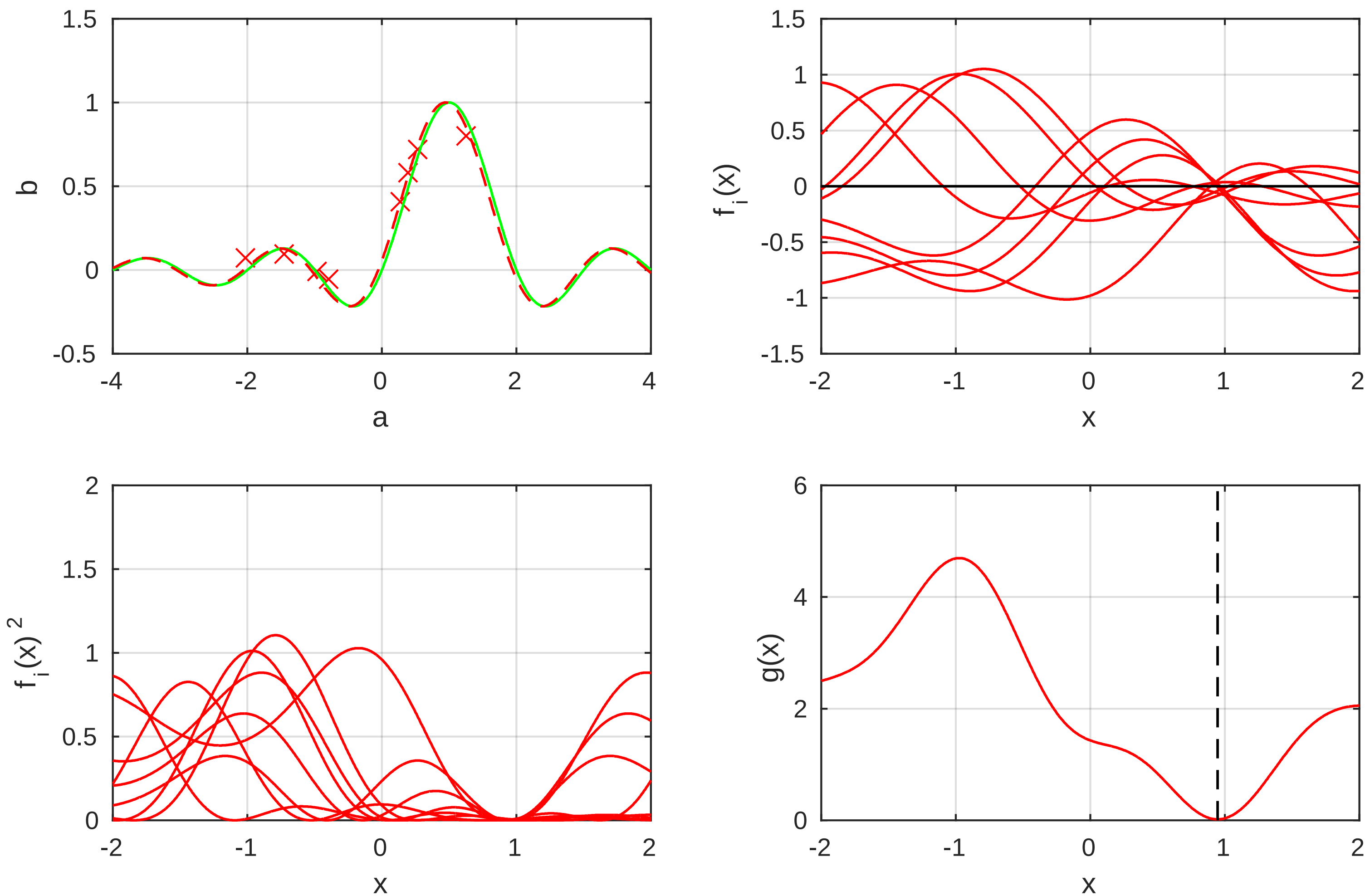
- Introduction
- Part I: Basics of Mathematical Optimization
  - Linear Least Squares
  - Nonlinear Optimization
- Part II: Basics of Computer Vision
  - Camera Model
  - Multi-Camera Model
  - Multi-Camera Calibration
- Part III: Depth Cameras
  - Passive Stereo
  - Structured Light Cameras
  - Time of Flight Cameras

Nonlinear Least Squares Problem

- **Example:** Consider 1D-NLS problem with single parameter $x \in \mathbb{R}$
  - nonlinear model function $b(a, x) = \mathrm{sinc}(a - x)$
  - input data $a_1, \ldots, a_m \in \mathbb{R}$
  - measurements $b_1, \ldots, b_m \in \mathbb{R}$
  - residual functions are $f_i(x) := b(a_i, x) - b_i, \; i = 1, \ldots, m$
- **Task:** $\min_{x \in \mathbb{R}} g(x)$ with $g(x) = \sum_{i=1}^{n} f_i(x)^2 = \sum_{i=1}^{n} (\mathrm{sinc}(a_i - x) - b_i)^2$
- **Solution:**
  - Analytic: Find closed-form solution for $g'(x) = \sum_{i=1}^{n} 2 f_i(x) f_i'(x) = 0$
  - Iterative methods, *e. g.*, gradient descent methods, Newton methods (Gauss-Newton, Levenberg-Marquardt algorithm)

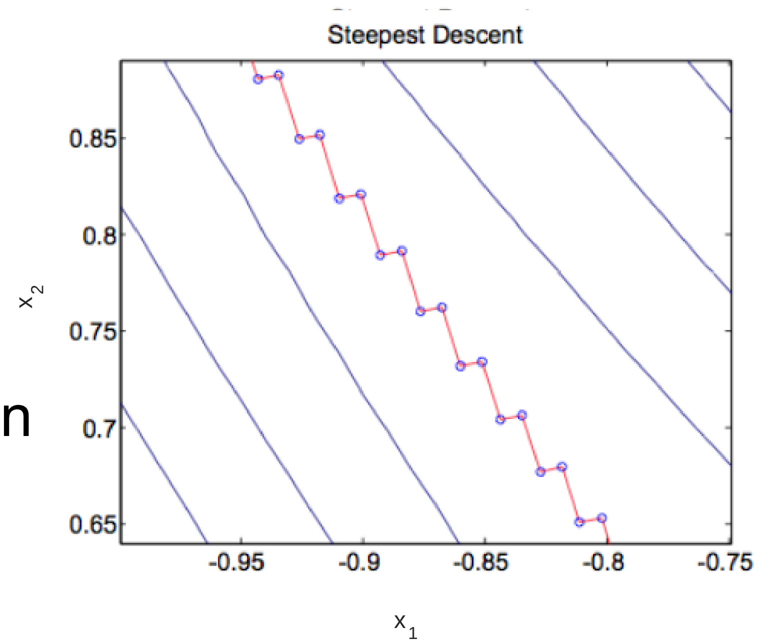
Nonlinear Least Squares Problem

- **Example:** Ground truth value is $x^* = 1$, $m = 8$ measurements b_i for inputs a_i, measurement error from normal distribution $\varepsilon_b \sim \mathcal{N}(0, 0.1)$



Gradient Descent Algorithm

- **Aim:** Find local minimum of nonlinear $g : \mathbb{R}^n \to \mathbb{R}$ starting from $\boldsymbol{x}_0 \in \mathbb{R}^n$
- In each step $k = 0, \dots, k_{\max}$:
  - Compute gradient at current $\boldsymbol{x}_k$: $\nabla g(\boldsymbol{x}_k) = \left(\frac{\partial g(\boldsymbol{x}_k)}{\partial x_1}, \dots, \frac{\partial g(\boldsymbol{x}_k)}{\partial x_n} \right)$
  - Move "downhill": $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k + \alpha_k \Delta \boldsymbol{x}, \ \Delta \boldsymbol{x} := -\nabla g(\boldsymbol{x}_k)^T$
  - Choose stepwidth α_k so that $g(\boldsymbol{x}_k + \alpha_k \Delta \boldsymbol{x}) < g(\boldsymbol{x}_k)$ (different strategies, *e. g.*, steepest descent: use line search $\alpha_k = \arg\min_{\alpha} g(\boldsymbol{x}_k + \alpha \Delta \boldsymbol{x})$)
  - Steps orthogonal to contour lines of g
  - Terminate if $\|\nabla g(\boldsymbol{x}_k)\| < \varepsilon_{\text{grad}}$
- **Convergence:** Stable, but slow
- **Example:** Rosenbrock's "banana" function



Newton Methods

- **Aim:** Find local minimum of nonlinear $g : \mathbb{R}^n \rightarrow \mathbb{R}$ starting from $\boldsymbol{x}_0 \in \mathbb{R}^n$
- In each step $k = 0, \ldots, k_{\max}$:
  - Approximate with Taylor expansion of 2nd order:
    $$g(\boldsymbol{x}_k + \Delta\boldsymbol{x}) \approx g(\boldsymbol{x}_k) + \nabla g(\boldsymbol{x}_k)\Delta\boldsymbol{x} + \frac{1}{2}\Delta\boldsymbol{x}^T \mathbf{H}_g(\boldsymbol{x}_k)\Delta\boldsymbol{x}$$
  - Solve $\nabla g(\boldsymbol{x}) = \mathbf{0}$ for approximation, solution is
    $$\nabla g(\boldsymbol{x}_k) + \mathbf{H}_g(\boldsymbol{x}_k)\Delta\boldsymbol{x} = \mathbf{0} \;\Rightarrow\; \Delta\boldsymbol{x} := -\mathbf{H}_g(\boldsymbol{x}_k)^{-1}\nabla g(\boldsymbol{x}_k)$$
  - Update x for next iteration: $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k + \Delta\boldsymbol{x}$
  - Terminate if $\|\nabla g(\boldsymbol{x}_k)\| < \varepsilon_{\text{grad}}$
- **Convergence:** Quadratic convergence, often combined with line search
- **Drawback:** Hessian $\mathbf{H}_g$ must be computed at each step

Gauss-Newton Algorithm

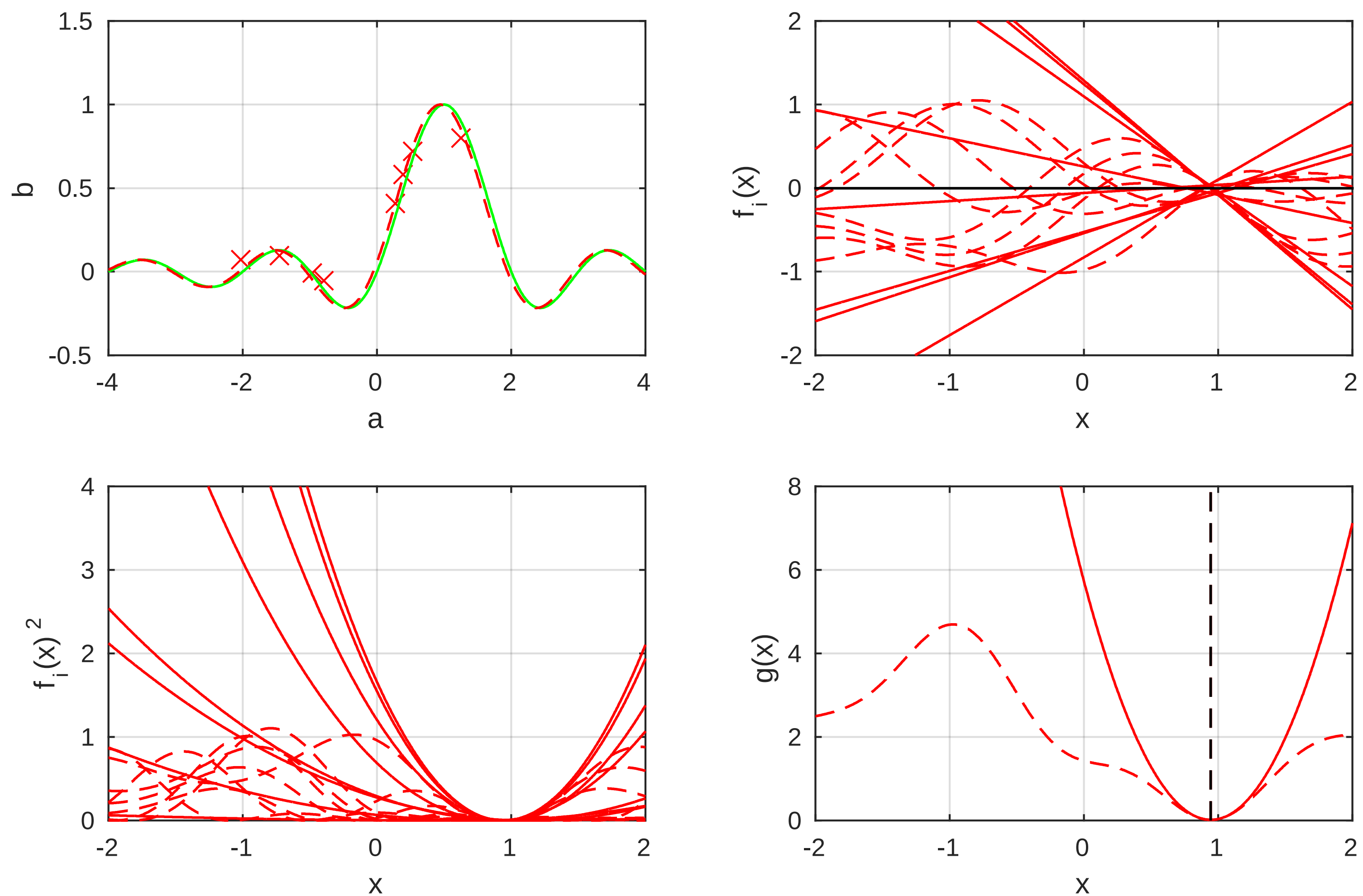
- **Aim:** Find local minimum of NLS problem near initial solution $\boldsymbol{x}_0 \in \mathbb{R}^n$

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} g(\boldsymbol{x}) \quad \text{with} \quad g(\boldsymbol{x}) = \|\boldsymbol{f}(\boldsymbol{x})\|^2 = \sum_{i=1}^{m} f_i(\boldsymbol{x})^2$$

- In each step $k = 0, \ldots, k_{\max}$:
  - Approximate $\boldsymbol{f}(\boldsymbol{x}_k + \Delta\boldsymbol{x}) \approx \underbrace{\boldsymbol{f}(\boldsymbol{x}_k)}_{\boldsymbol{f}_k} + \underbrace{\frac{\partial \boldsymbol{f}(\boldsymbol{x}_k)}{\partial \boldsymbol{x}}}_{\mathbf{J}_k} \Delta\boldsymbol{x} = \mathbf{J}_k \Delta\boldsymbol{x} + \boldsymbol{f}_k$
  - Solve $\min_{\Delta\boldsymbol{x}\in\mathbb{R}^n} \|\mathbf{J}_k \Delta\boldsymbol{x} + \boldsymbol{f}_k\|^2$, solution is $\Delta\boldsymbol{x} := -(\mathbf{J}_k^T \mathbf{J}_k)^{-1} \mathbf{J}_k^T \boldsymbol{f}_k$
  - Update x for next iteration: $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k + \Delta\boldsymbol{x}$
  - Terminate if $\|\boldsymbol{f}_k\| < \varepsilon_{\text{error}}$, $\|\Delta\boldsymbol{x}\| < \varepsilon_{\text{param}}$ or $\|\mathbf{J}_k^T \boldsymbol{f}_k\| < \varepsilon_{\text{grad}}$
- **Convergence:** Unstable, but fast

Gauss-Newton Algorithm

- **Example:** Ground truth value is $x^* = 1, m = 8$ measurements b_i for inputs a_i , measurement error from normal distribution $\varepsilon_b \sim \mathcal{N}(0, 0.1)$



Levenberg-Marquardt Algorithm

- Gauss-Newton approximation is not good when away from the minimum in regions where curvature is negative:
  - Better use steepest descent step in such cases.
- Steppest descent can progress slowly when close to the minimum ("zig-zagging"):
  - Better use Gauss-Newton step in such cases.
- The Levenberg-Marquardt algorithm provides mechanism for changing between steepest descent and Gauss-Newton steps depending on how good the approximation is locally.

Levenberg-Marquardt Algorithm

- **Aim:** Find local minimum of NLS problem near initial solution $\boldsymbol{x}_0 \in \mathbb{R}^n$

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} g(\boldsymbol{x}) \quad \text{with} \quad g(\boldsymbol{x}) = \|\boldsymbol{f}(\boldsymbol{x})\|^2 = \sum_{i=1}^{m} f_i(\boldsymbol{x})^2$$

- In each step $k = 0, \ldots, k_{\max}$:
  - Approximate $\boldsymbol{f}(\boldsymbol{x}_k + \Delta\boldsymbol{x}) \approx \underbrace{\boldsymbol{f}(\boldsymbol{x}_k)}_{\boldsymbol{f}_k} + \underbrace{\frac{\partial \boldsymbol{f}(\boldsymbol{x}_k)}{\partial \boldsymbol{x}}}_{\mathbf{J}_k} \Delta\boldsymbol{x} = \mathbf{J}_k \Delta\boldsymbol{x} + \boldsymbol{f}_k$
  - Solve $\min_{\Delta\boldsymbol{x}\in\mathbb{R}^n} \|\mathbf{J}_k\Delta\boldsymbol{x} + \boldsymbol{f}_k\|^2$ with damping factor $\mu_k \geq 0$
    solution is $\Delta\boldsymbol{x} := -(\mathbf{J}_k^T\mathbf{J}_k + \mu_k \operatorname{diag}(\mathbf{J}_k^T\mathbf{J}_k))^{-1}\mathbf{J}_k^T\boldsymbol{f}_k$
  - Update x for next iteration: $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k + \Delta\boldsymbol{x}$
  - Update μ for next iteration to improve convergence
  - Terminate if $\|\boldsymbol{f}_k\| < \varepsilon_{\text{error}}$, $\|\Delta\boldsymbol{x}\| < \varepsilon_{\text{param}}$ or $\|\mathbf{J}_k^T\boldsymbol{f}_k\| < \varepsilon_{\text{grad}}$

Constrained Optimization

Consider optimization problem with equality constraint:

- Given is a cost function $g : \mathbb{R}^n \to \mathbb{R}$
  and constraint function $h : \mathbb{R}^n \to \mathbb{R}$
- **Aim:** Find parameters $\boldsymbol{x} \in \mathbb{R}^n$ that minimize g subject to $h(\boldsymbol{x}) = 0$

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} g(\boldsymbol{x}) \;\; \text{s. t.} \;\; h(\boldsymbol{x}) = 0$$

Solutions:

- Add penalty term to cost function (with heuristic weight μ):

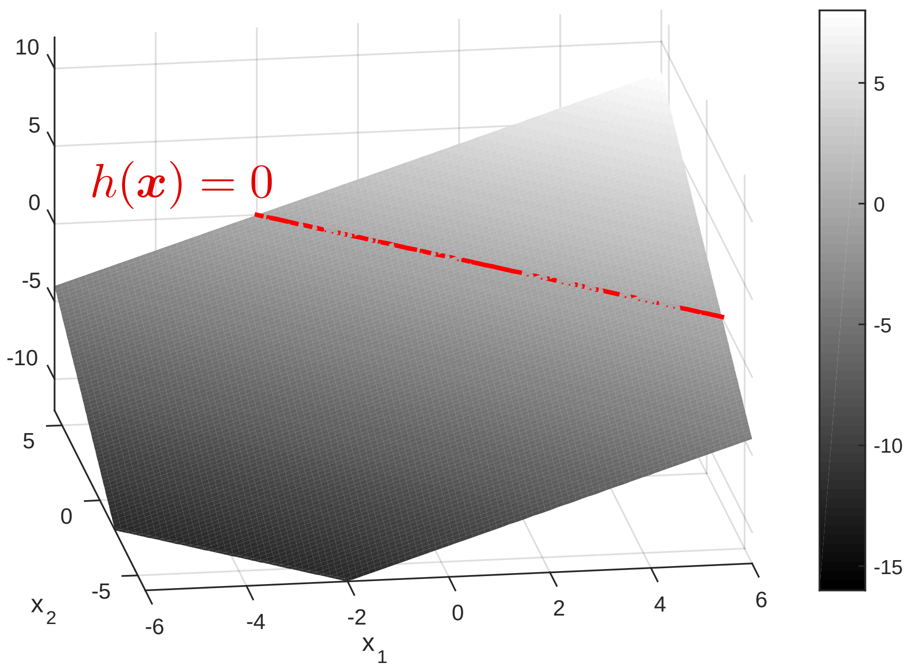
$$\min_{\boldsymbol{x} \in \mathbb{R}^n} g(\boldsymbol{x}) + \mu h(\boldsymbol{x})^2$$

  - **Pro:** Can be solved with default methods, *e. g.*, Levenberg-Marquardt
  - **Contra:** Result depends on choice of μ
- Solve with Lagrange multiplier method

Constrained Optimization

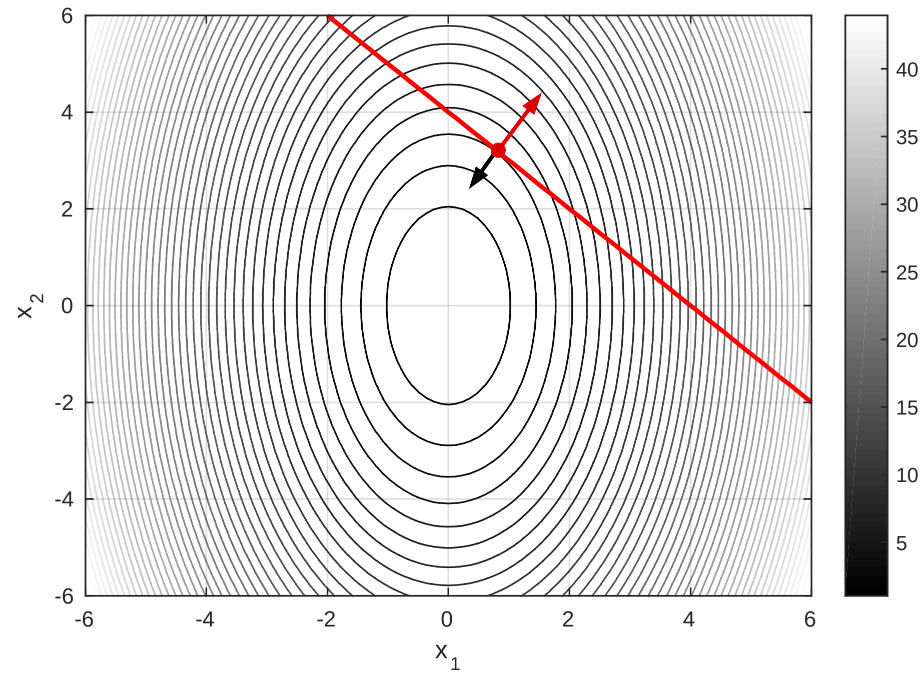
Lagrange multiplier:

- **Note:** Gradient of g is parallel to gradient of h at a constrained minimum



Plot of constraint function

$$h(\boldsymbol{x}) = x_1 + x_2 - 4$$



Plot of objective function

$$g(\boldsymbol{x}) = x_1^2 + (\frac{x_2}{2})^2$$

Constrained Optimization

Lagrange multiplier:

- **Note:** Gradient of g is parallel to gradient of h at a constrained minimum
- This is described by critical points of Lagrange function L, *i. e.*, extension of g by h scaled with an additional parameter λ (Lagrange multiplier):

  $$L(\boldsymbol{x}, \lambda) := g(\boldsymbol{x}) - \lambda h(\boldsymbol{x})$$

- Critical point conditions:

  $$\frac{\partial}{\partial \boldsymbol{x}} L(\boldsymbol{x}, \lambda) = \nabla g(\boldsymbol{x}) - \lambda \nabla h(\boldsymbol{x}) = \mathbf{0} \quad \rightarrow \text{gradients are parallel}$$

  $$\frac{\partial}{\partial \lambda} L(\boldsymbol{x}, \lambda) = h(\boldsymbol{x}) = 0 \quad \rightarrow \text{constraint is satisfied}$$

- Solve $\nabla L(\boldsymbol{x}, \lambda) = \mathbf{0}$ to obtain constrained minima of g

Constrained Optimization

- **Example:** Solve underconstrained linear least squares problem for unit length parameter vector $\boldsymbol{x}$:

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\mathbf{A}\boldsymbol{x}\|^2 \ \text{ s. t. } \ \|\boldsymbol{x}\|^2 = 1$$

- Lagrange function is:

$$L(\boldsymbol{x}, \lambda) := \boldsymbol{x}^T \mathbf{A}^T \mathbf{A}\boldsymbol{x} - \lambda(\boldsymbol{x}^T \boldsymbol{x} - 1)$$

- Critical point (*i. e.*, constrained minimum of g) satisfies:

$$\frac{\partial}{\partial \boldsymbol{x}} L(\boldsymbol{x}, \lambda) = 2\mathbf{A}^T \mathbf{A}\boldsymbol{x} - 2\lambda \boldsymbol{x} = \mathbf{0}$$

$$\Rightarrow \ \mathbf{A}^T \mathbf{A}\boldsymbol{x} = \lambda \boldsymbol{x}$$

- Solution $\boldsymbol{x}$ is unit length eigenvector of matrix $\mathbf{A}^T \mathbf{A}$
- Can be solved via matrix decomposition (*e. g.*, via SVD)

Optimization Problems in Computer Vision

- **Relative pose estimation:** Estimate rotation and translation between two cameras from 2D/2D point correspondences
- **Absolute pose estimation:** Estimate camera rotation and translation from 2D/3D point correspondences
- **Absolute orientation:** Estimate rotation and translation between two point sets from 3D/3D point correspondences
- **Camera calibration:** Estimate camera function from 2D/3D point correspondences
- **Stereo calibration:** Estimate rotation and translation between two cameras from 2D/3D point correspondences
- **Stereo reconstruction:** Estimate 3D point from 2D projections in two camera images with known stereo calibration