Xavier Granier

INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech

Inria informatiques mathématiques

-- Least Squares Fitting –

Finite-Dimensional Vector Spaces

THE 35TH ANNUAL CONFERENCE OF THE EUROPEAN ASSOCIATION FOR COMPUTER GRAPHICS

EUROGRAPHICS 2014

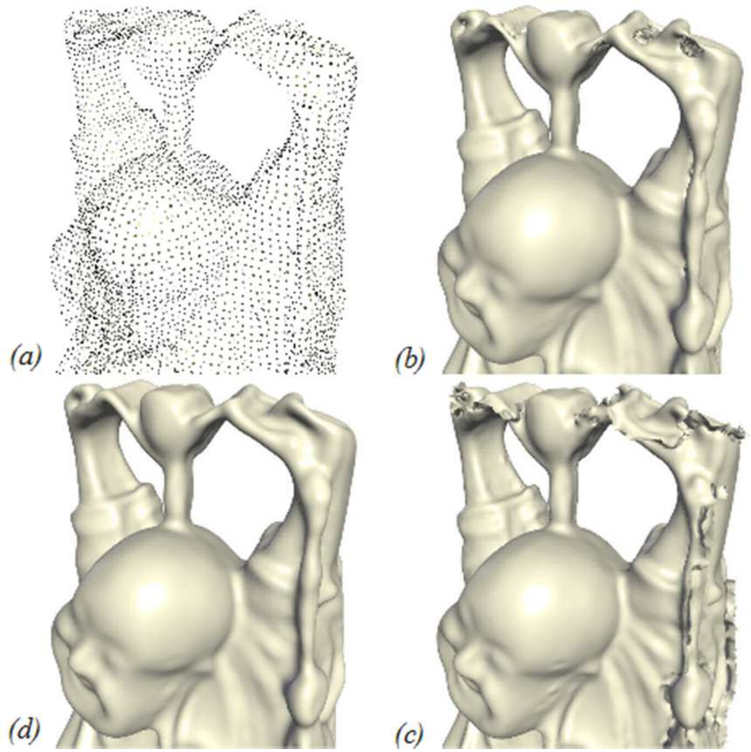
Strasbourg, France

CONFERENCE 7-11 APRIL 2014 STRASBOURG PALAIS DES CONGRÈS

General considerations on objectives

Data approximation and analysis

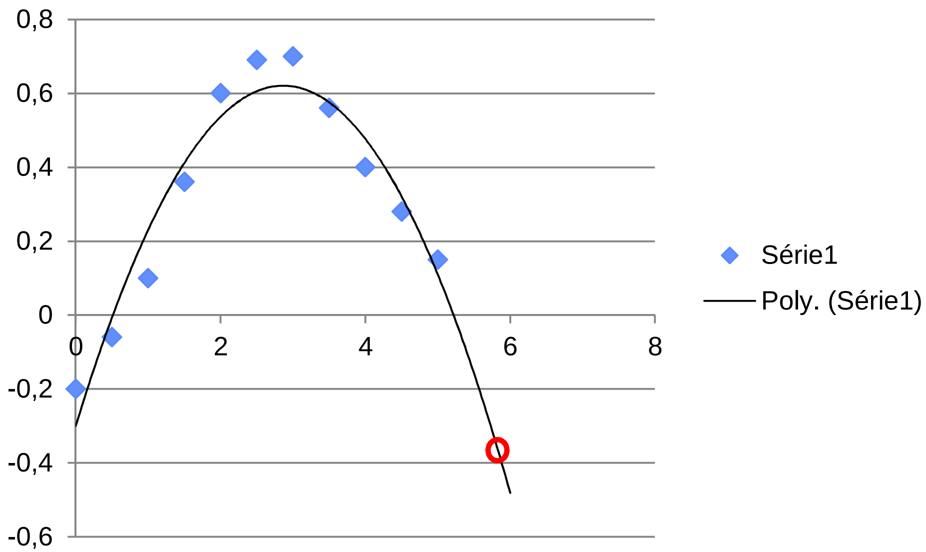
- Data from real measurements
  - How to use them in simulation / rendering ?
    - Ex: acquired point clouds for geometry



[Chen et al. – CGF 2013]

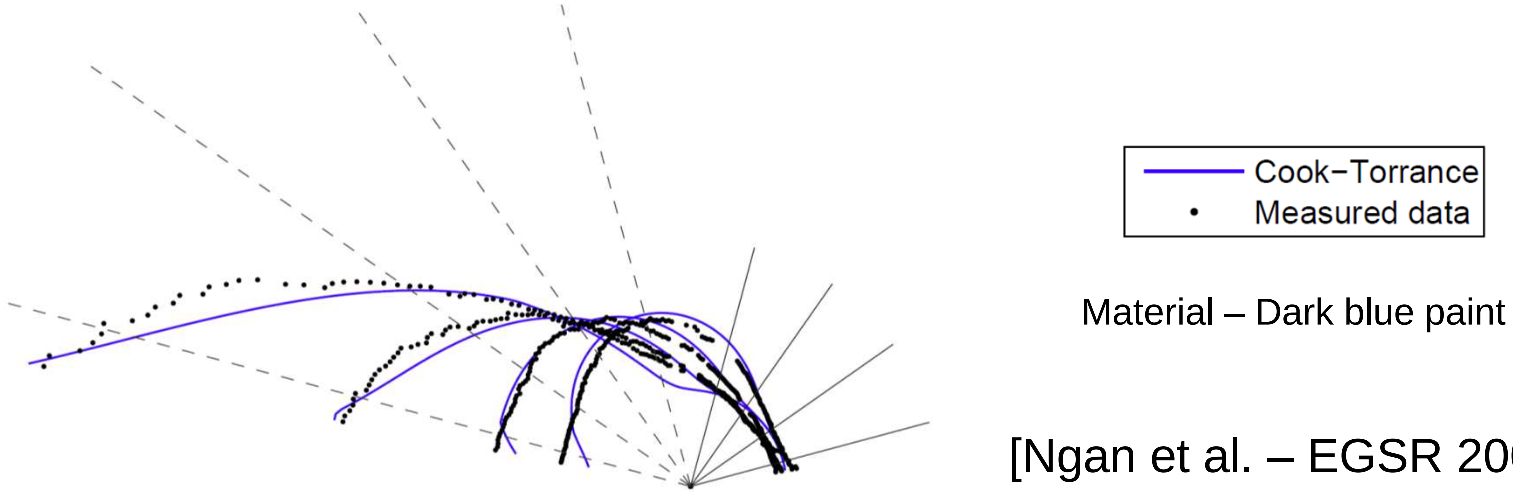
Data approximation and analysis

- Data from real measurements
  - How to use them in simulation / rendering ?
  - How to study the general behavior ?
    - Ex: data extrapolation in statistics



Data approximation and analysis

- Data from real measurements
  - How to use them in simulation / rendering ?
  - How to study the general behavior ?
  - How to remove the noise ?
    - Ex: BRDF measures at grazing angle

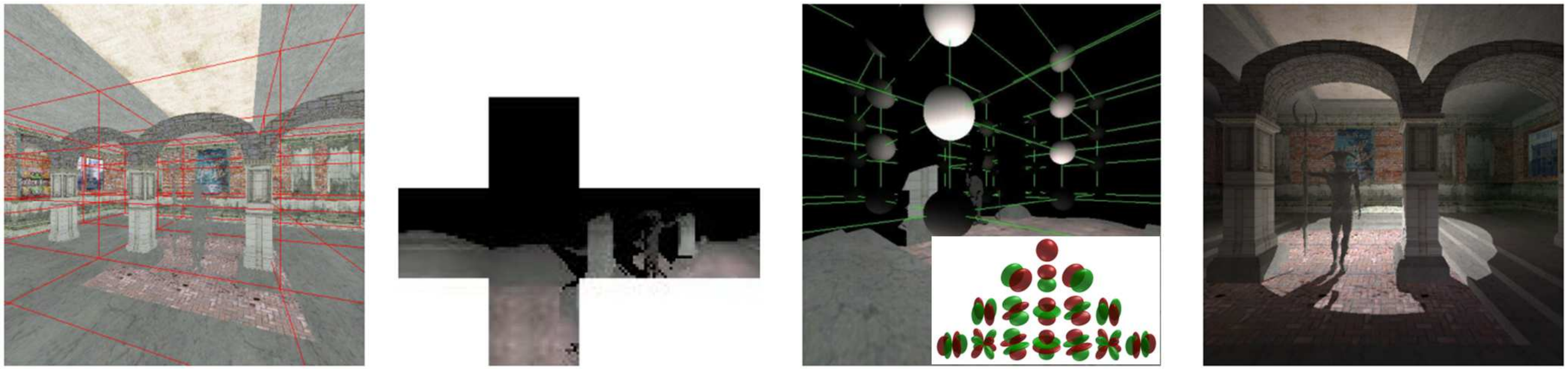


Material – Dark blue paint

[Ngan et al. – EGSR 2005]

Data modeling and conversion

- Computed data
  - Conversion between representations
    - Ex: environment map to spherical harmonics



1(a) Uniform voxel grid 1(b) Sample cubemap 1(c) Spherical Harmonics 1(d) Result

[Nijasure et al. – JGT 2005]

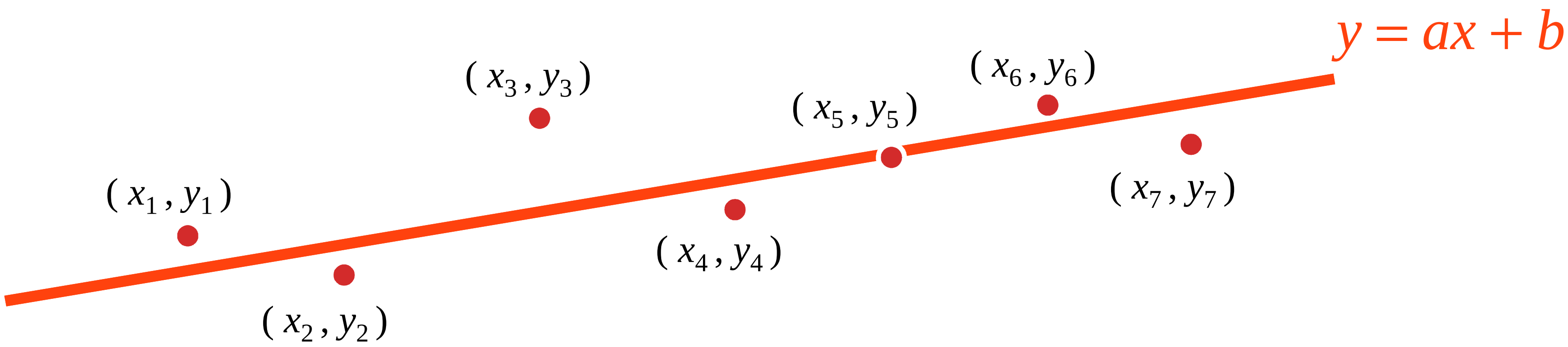
Data modeling and conversion

- Computed data
  - Conversion between representations
  - Objective-based modeling
    - Ex: anisotropic BRDF orientation field

[Raymond et al. – EG 2014]

Generalized Goal

- Finding the best approximation
  - Given a numerical model
  - Using a reduce set of parameters
- Ex: linear regression



Definition of “Best”

- **Maximize the quality**
  - Ex: expectation maximization

- Be as close as possible to the goal
  - Need a notion of **distance / norm**
  - To be **minimized**

Definitions

- Norm $\boldsymbol{x} = (x_1 \quad \dots \quad x_N)^{\mathrm{T}}$
  - Separate points

$$\| \boldsymbol{x} \| = 0 \Leftrightarrow \forall i = 1..N, x_i = 0$$

  - Absolute homogeneity

$$\| \lambda \boldsymbol{x} \| = | \lambda | \, \| \mathbf{x} \|$$

  - Triangle inequality

$$\| \boldsymbol{x} + \boldsymbol{y} \| \leq \| \boldsymbol{x} \| + \| \boldsymbol{y} \|$$

- Distance

$$\mathrm{d}(\boldsymbol{x}, \boldsymbol{y}) = \| \boldsymbol{x} - \boldsymbol{y} \|$$

Euclidian Norm for Least Squares

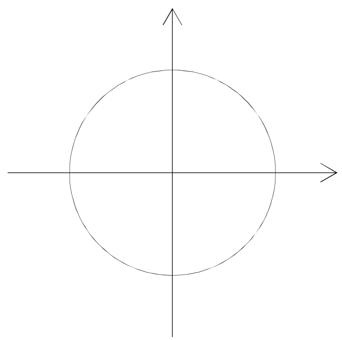
- Based on standard dot product

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^{N} x_i \, y_i = \boldsymbol{x}^{\mathrm{T}} \boldsymbol{y}$$

$$\| \boldsymbol{x} \|_2^2 = \langle \boldsymbol{x}, \boldsymbol{x} \rangle = \boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}$$

$$\| x_1 \dots x_N \|_2 = \left(\sum_{i=1}^{N} x_i^2 \right)^{\frac{1}{2}}$$

- Error ≈ average distance
  - Uniform weight for each dimension

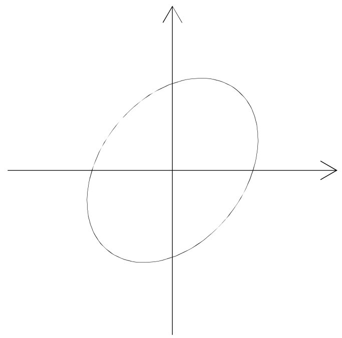


Euclidian Norm for Least Squares

- Generalized dot product
  - $\mathbf{W}$: symmetric positive definite matrix

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathbf{W}} = \boldsymbol{x}^{\mathrm{T}}\, \mathbf{W}\, \boldsymbol{y}$$

$$\| \boldsymbol{x} \|_{\mathbf{W}}^{2} = \langle \boldsymbol{x}, \boldsymbol{x} \rangle_{\mathbf{W}} = \boldsymbol{x}^{\mathrm{T}}\, \mathbf{W}\, \boldsymbol{x}$$

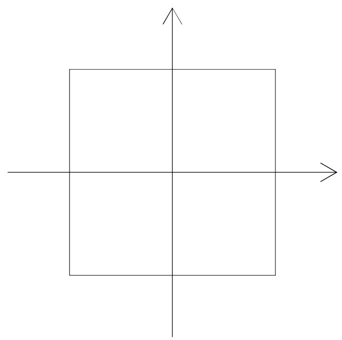


- Error ≈ weighted average distance
  - $\mathbf{W}$ is a diagonal matrix

Other: max norm

- Maximum of absolute values

$$\left\|(x_1 \quad \ldots \quad x_N)^{\mathrm{T}}\right\|_\infty = \max_{i=1..\mathrm{N}}\{|\ x\_i\ |\}$$

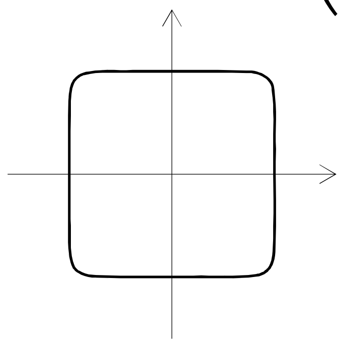


- Largest error

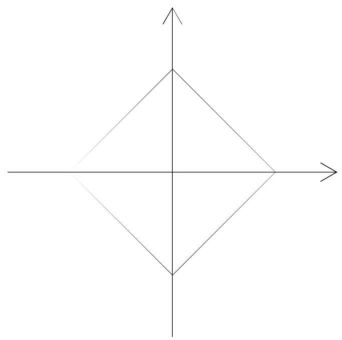
Other: p-norm

- Generalizing the Euclidian norm

$$\left\|\begin{pmatrix} x_1 & \dots & x_N \end{pmatrix}^{\mathrm{T}}\right\|_p = \left(\sum_{i=1}^{N} |x_i|^p\right)^{\frac{1}{p}}$$



$\| \|_{10}$



$\| \|_1$

Linear Optimization

Least Squares

Linear optimization

- M measured data $(x_m, y_m)_{m=1..M}$
- Linear approximating function
  – Parameters: $\boldsymbol{v}$ $\boldsymbol{v} = (v_1 \quad \dots \quad v_K)^{\mathrm{T}}$
  – Linear combination of basis function: $\mathbf{f}_k$

$$\boldsymbol{y} = \mathbf{f}_{\boldsymbol{v}}(\boldsymbol{x}) = \sum_{k=1}^{K} v_k \, \mathbf{f}_k(\boldsymbol{x})$$

  – 2D example 2D: line $y = ax + b$

$$y = \mathrm{f}_{(a,b)}(x) = ax + b$$

Least Squares

- Minimize Euclidian error = objective

$$E = \|(\boldsymbol{y}_m - \mathbf{f}_v(\boldsymbol{x}_\mathrm{m}))_m\|_2^2 = \sum_{m=1}^{M} \|\boldsymbol{y}_m - \mathbf{f}_v(\boldsymbol{x}_\mathrm{m})\|_2^2$$

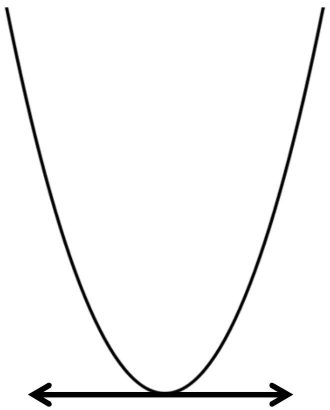
- Unique solution if well conditioned
  - Do not contain the trivial solution $\boldsymbol{v} = 0$
    - Example: implicit line

$$0 = \mathrm{f}_{(a,b,c)}(x, y) = ay + bx + c$$

  - Measures ≥ parameters: $M \geq K$
  - Measures are different

Solving Linear Least Squares

- Properties of the objective function
  - Positive
  - Quadratic
  - Parabola



- Minimum when gradient = 0

$$\forall k = 1..K, \qquad \frac{\partial}{\partial v_k} \sum_{m=1}^{M} \| \boldsymbol{y}_m - \mathbf{f}_{\boldsymbol{v}}(\boldsymbol{x}_{\mathrm{m}}) \|_2^2 = 0$$

- Lead to a linear system to solve

$$\mathbf{A}\,\boldsymbol{v} = \boldsymbol{b}$$

Demonstration 1D

$$\forall k = 1..K, \qquad \sum_{m=1}^{M} \frac{\partial}{\partial v_k} (y_m - \mathrm{f}_{\boldsymbol{v}}(x_\mathrm{m}))^2 = 0$$

$$\forall k = 1..K, \qquad \sum_{m=1}^{M} \cancel{2} \mathrm{f}_k(\boldsymbol{x}_m)(y_m - \mathrm{f}_{\boldsymbol{v}}(x_\mathrm{m})) = 0$$

$$\forall k = 1..K, \qquad \sum_{m=1}^{M} \mathrm{f}_k(x_m)\, \mathrm{f}_{\boldsymbol{v}}(x_\mathrm{m}) = \sum_{m=1}^{M} y_m \mathrm{f}_k(x_m)$$

$$\forall k = 1..K, \qquad \sum_{m=1}^{M} \mathrm{f}_k(x_m) \sum_{j=1}^{K} v_j\, \mathrm{f}_j(x_\mathrm{m}) = \sum_{m=1}^{M} y_m \mathrm{f}_k(x_m)$$

Corresponding Linear System

$$\forall k = 1..K, \qquad \sum_{j=1}^{K} v_j \underbrace{\sum_{m=1}^{M} f_k(x_m)\, f_j(x_m)}_{a_{kj}} = \underbrace{\sum_{m=1}^{M} y_m f_k(x_m)}_{b_k}$$

$$\mathbf{A} = \mathbf{F}\,\mathbf{F}^{\mathrm{T}} \qquad\qquad \boldsymbol{b} = \mathbf{F}\,\boldsymbol{y}$$

$$F_{km} = f_k(x_m)$$

$\mathbf{A}$ symmetric (positive-definite)

Demonstration ND

$$\forall k = 1..K, \qquad \sum_{m=1}^{M} \frac{\partial}{\partial v_k} \| \boldsymbol{y}_m - \mathbf{f}_v(\boldsymbol{x}_\mathrm{m})_2^2 = 0$$

$$\forall k = 1..K, \qquad \sum_{m=1}^{M} \cancel{2} \langle \mathbf{f}_k(\boldsymbol{x}_m), \boldsymbol{y}_m - \mathbf{f}_v(\boldsymbol{x}_\mathrm{m}) \rangle = 0$$

$$\forall k = 1..K, \qquad \sum_{m=1}^{M} \langle \mathbf{f}_k(\boldsymbol{x}_m), \mathbf{f}_v(\boldsymbol{x}_\mathrm{m}) \rangle = \sum_{m=1}^{M} \langle \mathbf{f}_k(\boldsymbol{x}_m), \boldsymbol{y}_m \rangle$$

$$\forall k = 1..K, \qquad \sum_{m=1}^{M} \left\langle \mathbf{f}_k(\boldsymbol{x}_m), \sum_{j=1}^{K} v_j \, \mathbf{f}_j(\boldsymbol{x}_\mathrm{m}) \right\rangle = \sum_{m=1}^{M} \langle \mathbf{f}_k(\boldsymbol{x}_m), \boldsymbol{y}_m \rangle$$

Corresponding Linear System

$$\forall k = 1..K, \qquad \sum_{j=1}^{K} v_j \underbrace{\sum_{m=1}^{M} \langle \mathbf{f}_k(x_{\mathrm{m}}), \mathbf{f}_j(\boldsymbol{x}_{\mathrm{m}}) \rangle}_{a_{kj}} = \underbrace{\sum_{m=1}^{M} \langle \boldsymbol{y}_m, \mathbf{f}_k(\boldsymbol{x}_m) \rangle}_{b_k}$$

$\mathbf{A}$ symmetric (positive-definite)

Equivalent Linear System

- Minimal least squares error
  - Equivalent linear system

$$\mathbf{A}\,\boldsymbol{v} = \boldsymbol{b}$$

  - $\mathbf{A}$ symmetric
  - If well conditioned, $\mathbf{A}$ positive-definite

- How to solve it ?
  - Use your favorite linear algebra solver
  - Ex: Cholesky factorization

Conditioning of a linear system

- **Conditioning = stability of a system**
  - Input: $\boldsymbol{d}$ (perturbation $\boldsymbol{d} + \delta\boldsymbol{d}$)
  - Output: $\boldsymbol{x}$ (perturbation $\boldsymbol{x} + \delta\boldsymbol{x}$)
  - Relative conditioning
    - Smaller is better

$$K_{\mathrm{rel}}(\boldsymbol{d}) = \lim_{\epsilon \to 0} \sup_{\|\delta \boldsymbol{d}\| < \epsilon} \left\{ \frac{\|\delta \boldsymbol{x}\| / \|\boldsymbol{x}\|}{\|\delta \boldsymbol{d}\| / \|\boldsymbol{d}\|} \right\}$$

- For a linear system
  - Conditioning of the matrix
  - Symmetric positive-definite matrix
    - Ratio of eigenvalues

$$K(\mathbf{A}) = \frac{\lambda_{\max}}{\lambda_{\min}}$$

Adding constraints

- For regularization
  - Improvement on conditioning
  - Removing trivial solution
    - Example of implicit line $\| \boldsymbol{v} \|_2^2 \neq 0$

$$\min_{\boldsymbol{v}} \sum_{m=1}^{M} \| \mathrm{f}_{(a,b,c)}(x_m, y_m) \|_2^2$$

$$\begin{bmatrix} \sum_{m=1}^{M} x_m^2 & \sum_{m=1}^{M} x_m y_m & \sum_{m=1}^{M} x_m \\ \sum_{m=1}^{M} x_m y_m & \sum_{m=1}^{M} y_m^2 & \sum_{m=1}^{M} y_m \\ \sum_{m=1}^{M} x_m & \sum_{m=1}^{M} y_m & M \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Adding constraints

- For regularization
  - Improvement on conditioning
  - Removing trivial solution
    - Example of implicit line $\quad \| \boldsymbol{v} \|_2^2 \neq 0$

$$\min_{\boldsymbol{v}} \sum_{m=1}^{M} \| \mathrm{f}_{(a,b,c)}(x_m, y_m) \|_2^2 \quad + \epsilon(a + b + c - 1)^2$$

$$\begin{bmatrix} \sum_{m=1}^{M} x_m^2 + \epsilon & \sum_{m=1}^{M} x_m y_m + \epsilon & \sum_{m=1}^{M} x_m + \epsilon \\ \sum_{m=1}^{M} x_m y_m + \epsilon & \sum_{m=1}^{M} y_m^2 + \epsilon & \sum_{m=1}^{M} y_m + \epsilon \\ \sum_{m=1}^{M} x_m + \epsilon & \sum_{m=1}^{M} y_m + \epsilon & M + \epsilon \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \epsilon \\ \epsilon \\ \epsilon \end{bmatrix}$$

Adding constraints

- For regularization
  - Improvement on conditioning
  - Removing trivial solution
    - Example of implicit line $\| \boldsymbol{v} \|_2^2 \neq 0$

$$\min_{\boldsymbol{v}} \sum_{m=1}^{M} \| \mathrm{f}_{(a,b,c)}(x_m, y_m) \|_2^2 \quad + \epsilon(a+b+c-1)^2$$

- Other linear constraints
  - Ex: continuity, … (cf. geometry part)

Lagrange Multipliers

- Approach
  - Original objective $\min_{\boldsymbol{v}} \mathbf{E}(\boldsymbol{v})$
  - A new constraint $g(\boldsymbol{v}) = c$
  - New objective $\min_{\boldsymbol{v},\lambda} E(\boldsymbol{v}) + \lambda(g(\boldsymbol{v}) - c)$

- Minimum is reached when

$$\frac{\partial}{\partial v_k} E(\boldsymbol{v}) + \lambda \frac{\partial}{\partial v_k} g(\boldsymbol{v}) = 0$$

$$\frac{\partial}{\partial \lambda} E(\boldsymbol{v}) + \lambda(g(\boldsymbol{v}) - c) = 0 = g(\boldsymbol{v}) - c$$

Equivalent Linear System

- If multiple linear constraints $\boldsymbol{g}_j^{\mathrm{T}} \boldsymbol{v} = c_j$
- New objective function

$$\min_{\boldsymbol{v}} \sum_{m=1}^{M} \| \boldsymbol{y}_m - \mathbf{f}_{\boldsymbol{v}}(\boldsymbol{x}_m) \|_2^2 + \sum_{j=1}^{J} \lambda_j \left(\boldsymbol{g}_j^{\mathrm{T}} \boldsymbol{v} - c_j \right)$$

- **Unique solution if it exists**
  - But matrix may not be symmetric
    - Cf. geometry part of the tutorial

Linear Least Squares - Summary

- Avantages
  - Euclidian norm : in average the best
    - Robust to noise
  - Linear system to solve : unique solution
  - Extensions
    - Non-uniform norm
    - Linear constraints as equalities
- But
  - Minimizing maximal error ?
  - Inequality linear constraints ?

Linear Optimization

Linear/Quadratic Programming

Inequality constraints

- For regularization
  - Improvement on conditioning
  - Removing trivial solution
    - Example of implicit line $\| \boldsymbol{v} \|_2^2 \neq 0$

$$\min_{\boldsymbol{v}} \sum_{m=1}^{M} \| \mathrm{f}_{(a,b,c)}(x_m, y_m) \|_2^2 \quad + \epsilon(a+b+c-1)^2$$

- Other linear constraints
  - Ex: continuity, … (cf. geometry part)
  - Definition domain (see BRDF fitting)

Linear Programing

- Minimizing the max-norm

$$\min_{\boldsymbol{v}} \mathbf{max}_{m=1}^{M} \| y_m - \mathrm{f}_{\boldsymbol{v}}(\boldsymbol{x}_{\mathrm{m}}) \|_{\infty}$$

$$\Leftrightarrow \min_{\boldsymbol{v}} \| \boldsymbol{y} - \mathbf{f}_{\boldsymbol{v}}(\boldsymbol{x}) \|_{\infty}$$

- Towardlinear programing

$$\Leftrightarrow \begin{array}{c} \min_{\boldsymbol{v},\epsilon} \epsilon \\ \text{subject to} \begin{cases} \epsilon \geq 0 & \\ -y_m + \mathrm{f}_{\mathbf{v}}(\boldsymbol{x}_m) + \epsilon \geq 0 & \forall m \\ y_m - \mathrm{f}_{\mathbf{v}}(\boldsymbol{x}_m) - \epsilon \geq 0 & \forall m \end{cases} \end{array}$$

Linear programing

- Objective: dot product
- Constraints: linear equalities and inequalities

$$\begin{aligned} \min_{\boldsymbol{v}} \quad & \boldsymbol{a}^{\mathrm{T}}\boldsymbol{v} \\ \text{subject to} \quad & \boldsymbol{b}_m^{\mathrm{T}}\boldsymbol{v} \leq c_m \\ & \boldsymbol{d}_m^{\mathrm{T}}\boldsymbol{v} = e_m \end{aligned}$$

- **Unique solution if it exists**
- Solving
  - Simplex algorithm
  - #iterations ~ O(#constraints)

Simplex: Standard Form

$$\max_{\boldsymbol{v}} \boldsymbol{a}^{\mathrm{T}} \boldsymbol{v}$$

$$\text{subject to} \begin{cases} \boldsymbol{b}_m^{\mathrm{T}} \boldsymbol{v} + c_m = d_m & \forall m \\ v_k \geq 0 & \forall k \\ c_m \geq 0 & \forall m \end{cases}$$

$$\text{with} \begin{cases} \boldsymbol{v} = (v_1 \quad .. \quad v_k)^{\mathrm{T}} \\ \boldsymbol{a} = (a_1 \quad .. \quad a_k)^{\mathrm{T}} \\ \boldsymbol{b}_m = (b_{m1} \quad .. \quad b_{mk})^{\mathrm{T}} \end{cases}$$

Geometrical Analogy

Maximize $3\,x_1 + 5\,x_2$

Constraints

$$x_1 \leq 4$$

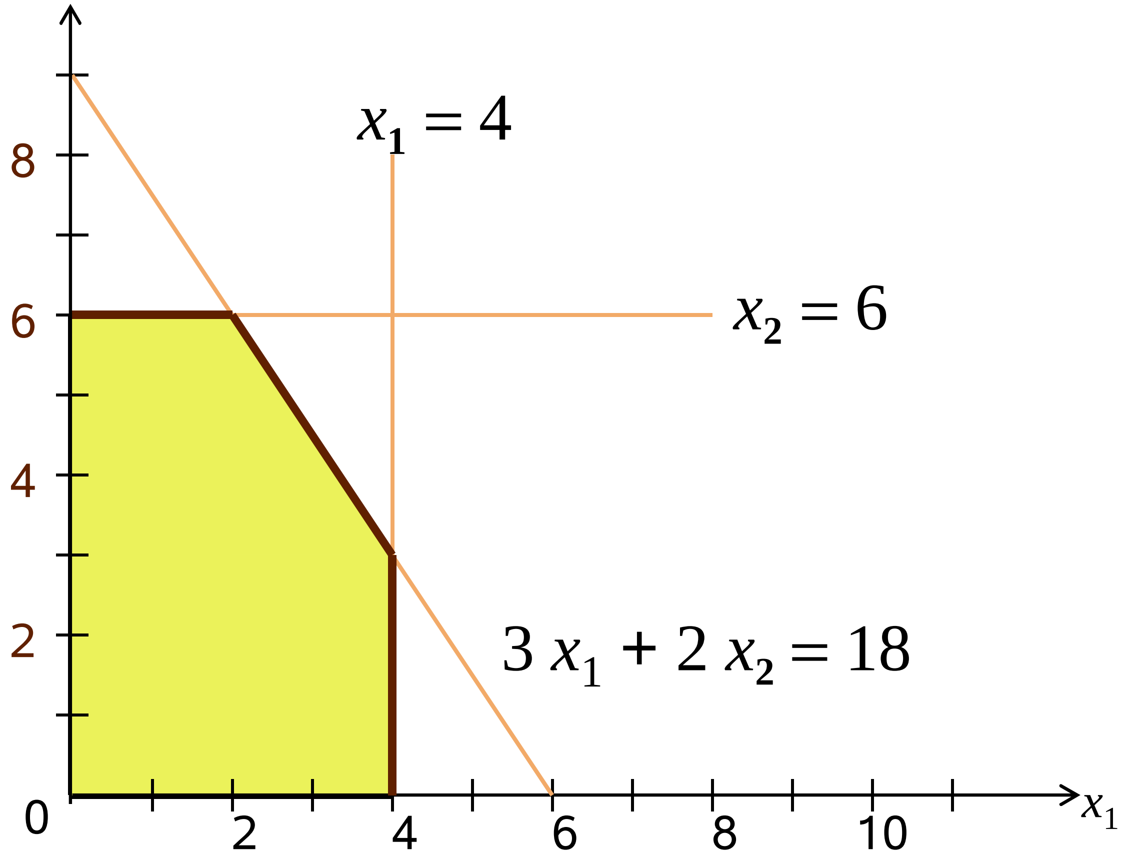
$$x_2 \leq 6$$

$$3\,x_1 + 2\,x_2 \leq 18$$

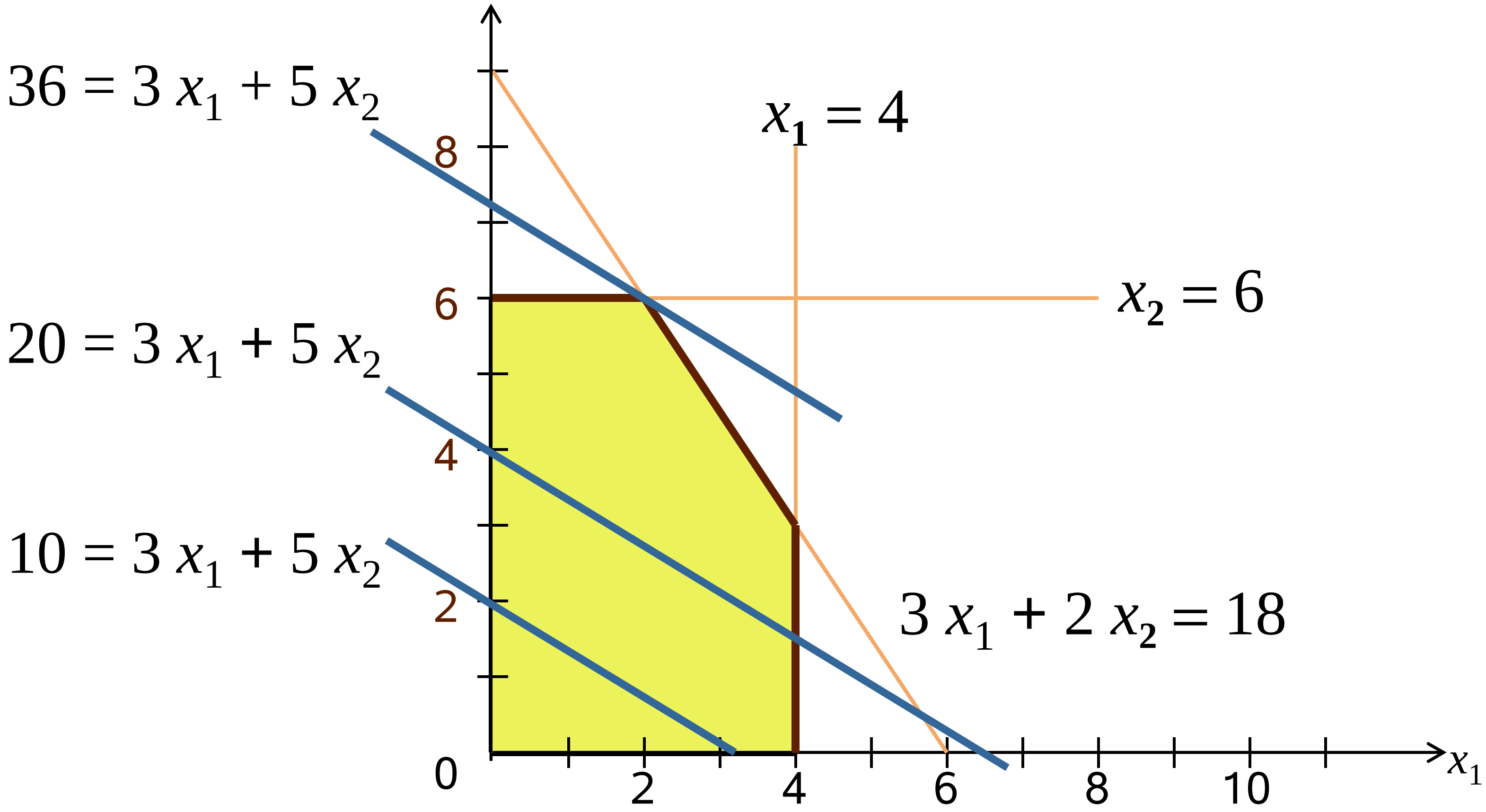
$$x_1 \geq 0$$

$$x_2 \geq 0$$

Geometrical Analogy



Geometrical Analogy



Quadratic Programing

- Term to minimize = quadratic form

$$\min_{\boldsymbol{v}} \boldsymbol{v}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\boldsymbol{v} + \boldsymbol{d}^{\mathrm{T}}\boldsymbol{v}$$

$$\text{subject to } \begin{cases} \boldsymbol{c}_j^{\mathrm{T}}\, v = b_j \\ \boldsymbol{c}_m^{\mathrm{T}}\, v \leq b_m \end{cases}$$

- Iterative solver
  - Classical least squares solver
    - Langrage multiplier for equalities
  - If some inequalities are not fulfilled
    - Take one and transform it into an equality
- **Unique solution if it exists**

Non-Linear Optimization

Non-linear Optimization

- When it is impossible to use
  - Linear combination of functions
  - Linear / quadratic objective function
  - Linear constraints
- Solvers are iterative
  - Step by step progression toward a solution
    - Still where gradient is null
  - Convergence toward a **local minima**
    - Not a unique solution
    - If a unique solution exists, it will be found

Finding $\mathrm{e}(x) = 0$ (Newton Method)

- 1st order Taylor expansion

$$\mathrm{e}\left(x^{(k)} + h\right) \simeq \mathrm{e}\left(x^{(k)}\right) + \partial_x e\left(x^{(k)}\right) h$$

- Look for 0-crossing

$$h = -\frac{\mathrm{e}\left(x^{(k)}\right)}{\partial_x \mathrm{e}\left(x^{(k)}\right)}$$

- Iterative scheme

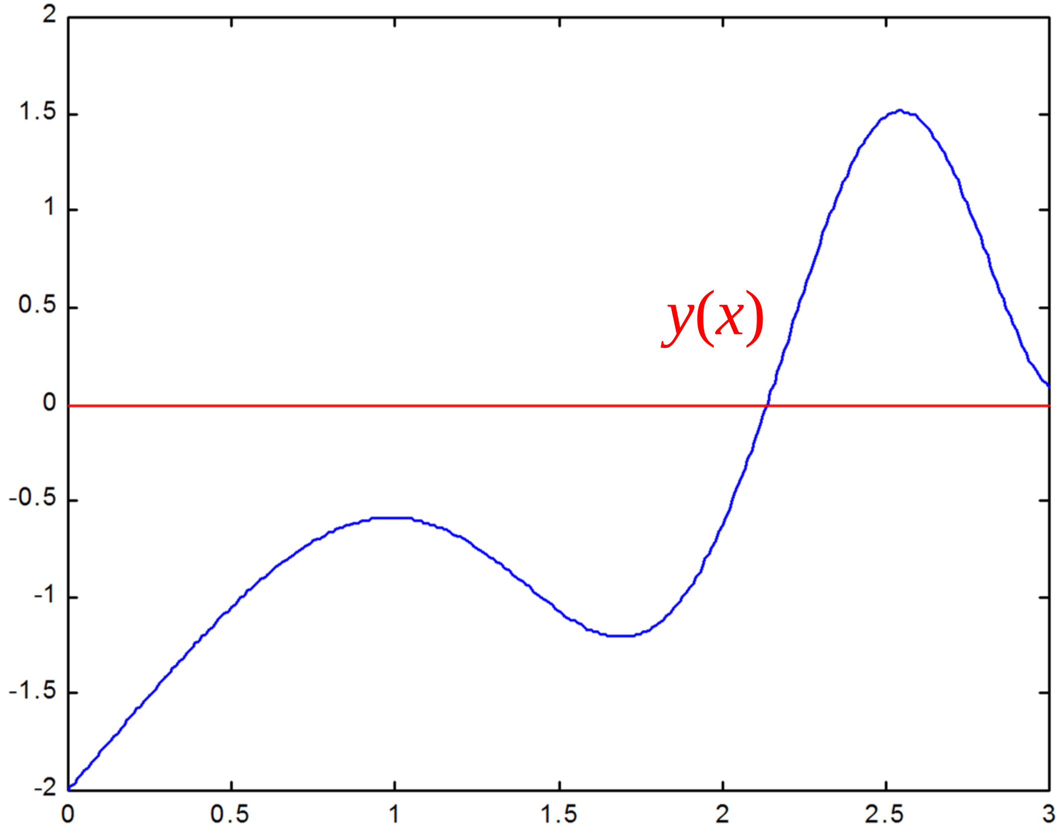
$$x^{(k+1)} = x^{(k)} - \frac{\mathrm{e}\left(x^{(k)}\right)}{\partial_x \mathrm{e}\left(x^{(k)}\right)}$$

Newton Method

- Illustration

$$y = \tanh(x)\cos(x^2) + x - 2$$

$$y' = (1-\tanh^2(x))\cos(x^2) - 2\tanh(x)\sin(x^2)\,x + 1$$



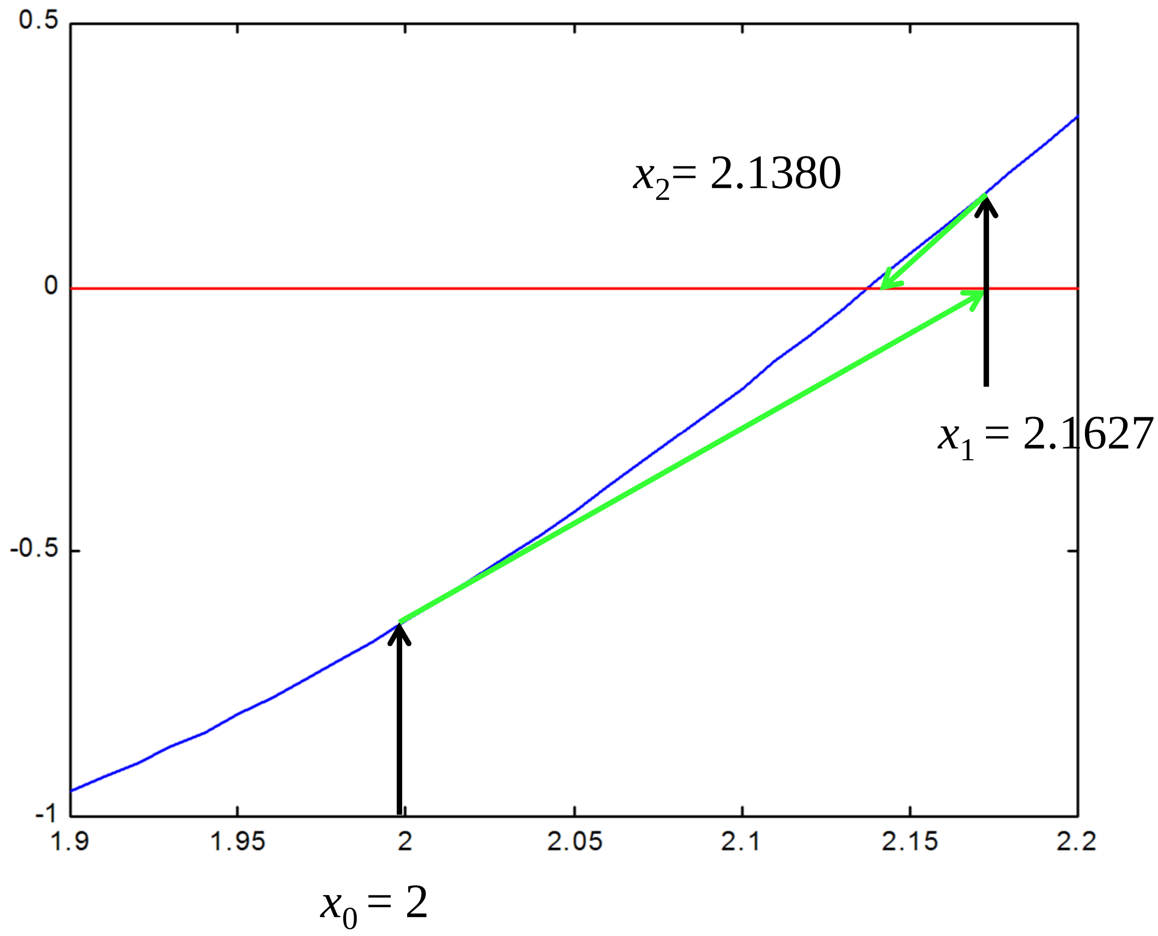
Newton Method

- Illustration

$$y = \tanh(x)\cos(x^2) + x - 2$$

$$y' = (1-\tanh^2(x))\cos(x^2) - 2\tanh(x)\sin(x^2)\,x + 1$$

$x_0 = 2$
$x_1 = 2.1627$
$x_2 = 2.1380$
$x_3 = 2.1378$
$x_4 = 2.1378$



©Insa Rouen

Newton Method: convergence

- Quadratic convergence

$$\left|x^{(k+1)} - x\right| \simeq \left|x^{(k)} - x\right|^2 \left|\frac{\partial^2_{xx}\mathrm{e}(x)}{2\partial_x \mathrm{e}(x)}\right|$$

- Conditions
  - Known analytic derivative
  - Tangent crosses 0-line in the definition domain.

1D Optimization – $e'(x) = 0$

- 2nd order Taylor expansion

$$e\left(x^{(k)} + h\right) \simeq e\left(x^{(k)}\right) + \partial_x e\left(x^{(k)}\right)h + \frac{1}{2}\partial_{xx}^2 e\left(x^{(k)}\right)h^2$$

- 0-crossing of derivative

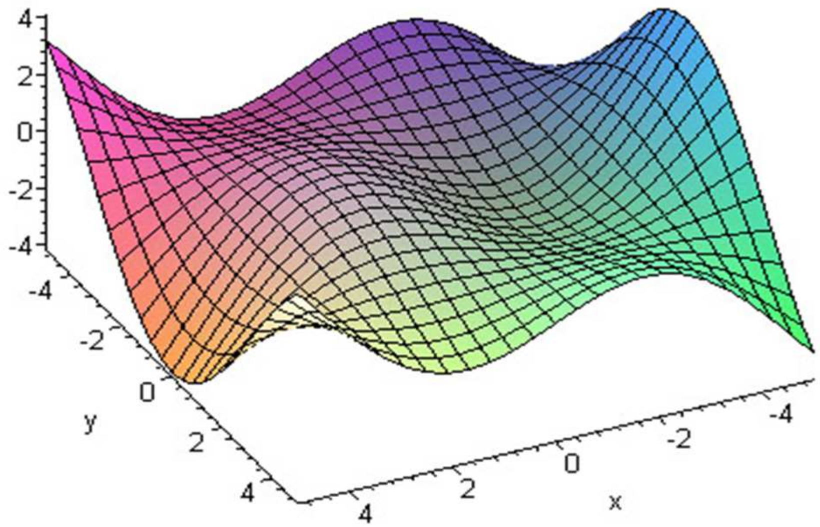
$$h = -\frac{\partial_x e\left(x^{(k)}\right)}{\partial_{xx}^2 e\left(x^{(k)}\right)}$$

- Similar iterative process

$$x^{(k+1)} = x^{(k)} - \frac{\partial_x e\left(x^{(k)}\right)}{\partial_{xx}^2 e\left(x^{(k)}\right)}$$

2D Taylor Expansion

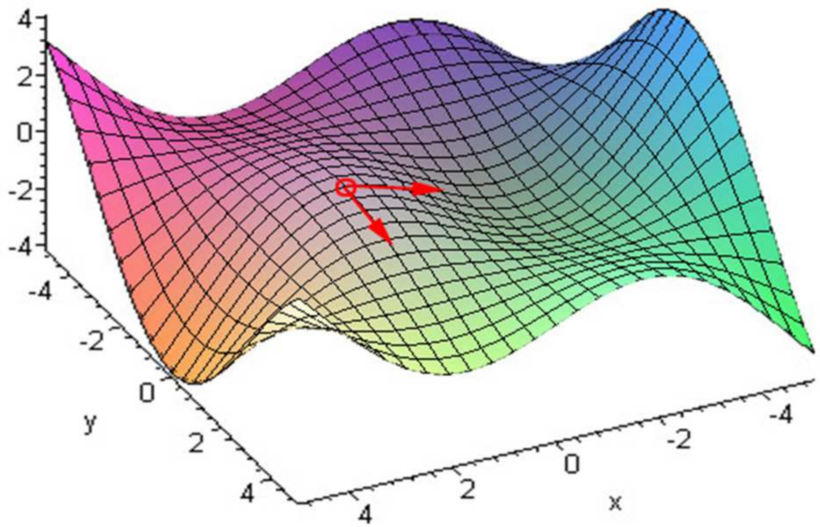
$$e(x, y) = x\cos\left(\frac{x}{2}\right)\sin\left(\frac{y}{2}\right)$$



2D Taylor Expansion

- Gradient $\nabla e(x) = \begin{bmatrix} \partial_x \mathrm{e} \\ \partial_y \mathrm{e} \end{bmatrix}$

$$\mathrm{e}(x, y) = x \cos\left(\frac{x}{2}\right) \sin\left(\frac{y}{2}\right)$$



$$\nabla e(x) = \begin{bmatrix} \sin\left(\frac{y}{2}\right)\left(\cos\left(\frac{x}{2}\right) - \frac{x}{2}\sin\left(\frac{x}{2}\right)\right) \\ \frac{x}{2}\cos\left(\frac{x}{2}\right)\cos\left(\frac{y}{2}\right) \end{bmatrix}$$

2D Taylor Expansion

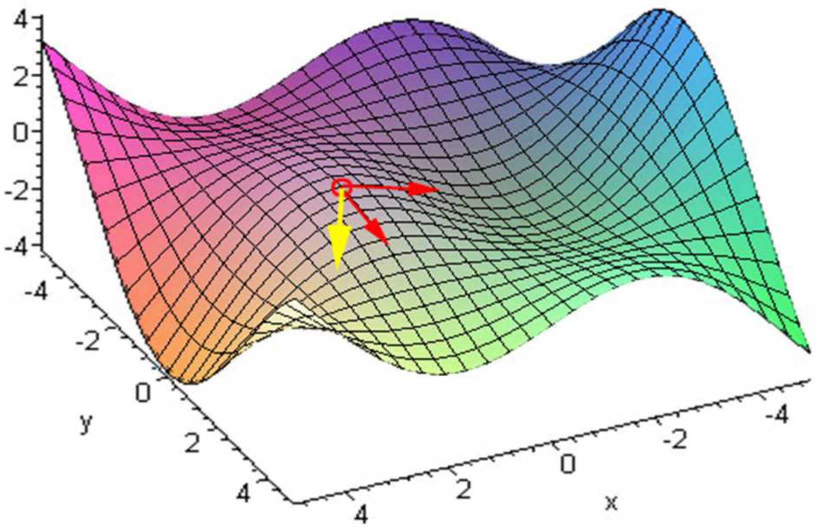
- Gradient $\boldsymbol{\nabla} e(x) = \begin{bmatrix} \partial_x \mathrm{e} \\ \partial_y \mathrm{e} \end{bmatrix}$
- 1st order derivative
  - Dot product with direction

$$\boldsymbol{d} = (d_x, d_y) \in \Re^2$$

$$\partial_{\boldsymbol{d}} \mathrm{e} = \langle \boldsymbol{d}, \boldsymbol{\nabla} \mathrm{e} \rangle$$

$$\partial_{\boldsymbol{d}} \mathrm{e} = d_x \partial_x \mathrm{e} + d_y \partial_y \mathrm{e}$$

$$\mathrm{e}(x,y) = x \cos\left(\frac{x}{2}\right) \sin\left(\frac{y}{2}\right)$$



N-dimensional Expansion

- 1 equation, N unknowns

$$\mathrm{e}(\boldsymbol{x}+\boldsymbol{h}) \simeq \mathrm{e}(\boldsymbol{x}) + \boldsymbol{h}^{\mathrm{T}}\boldsymbol{\nabla}\mathrm{e}(\boldsymbol{x}) + \frac{1}{2}\boldsymbol{h}^{\mathrm{T}}\mathbf{H}_{\mathbf{e}}(\boldsymbol{x})\boldsymbol{h} + \mathrm{o}(\|\,\boldsymbol{h}\,\|_2^2\,)$$

gradient

Hessian Matrix

$$\boldsymbol{\nabla} e = \begin{bmatrix} \partial_{x_1}\mathrm{e} \\ \vdots \\ \partial_{x_i}\mathrm{e} \\ \vdots \\ \partial_{x_N}\mathrm{e} \end{bmatrix} \quad \mathbf{H}_{\mathrm{e}}(\mathbf{x}) = \begin{bmatrix} \partial^2_{x_1 x_1}\mathrm{e} & \cdots & \partial^2_{x_1 x_i}\mathrm{e} & \cdots & \partial^2_{x_1 x_N}\mathrm{e} \\ \vdots & \ddots & \vdots & ⋰ & \vdots \\ \partial^2_{x_i x_1}\mathrm{e} & \cdots & \partial^2_{x_i x_i}\mathrm{e} & \cdots & \partial^2_{x_i x_N}\mathrm{e} \\ \vdots & ⋰ & \vdots & \ddots & \vdots \\ \partial^2_{x_N x_1}\mathrm{e} & \cdots & \partial^2_{x_N x_i}\mathrm{e} & \cdots & \partial^2_{x_N x_N}\mathrm{e} \end{bmatrix}$$

N-dimensional Expansion

- 1 equation, N unknowns

$$\mathrm{e}(\boldsymbol{x}+\boldsymbol{h}) \simeq \mathrm{e}(\boldsymbol{x}) + \boldsymbol{h}^{\mathrm{T}}\boldsymbol{\nabla}\mathrm{e}(\boldsymbol{x}) + \frac{1}{2}\boldsymbol{h}^{\mathrm{T}}\mathbf{H}_{\mathbf{e}}(\boldsymbol{x})\boldsymbol{h} + \mathrm{o}(\| \boldsymbol{h} \|_2^2)$$

gradient

Hessian Matrix = 2D Tensor

$$\boldsymbol{\nabla} e = \begin{bmatrix} \partial_{x_1}\mathrm{e} \\ \vdots \\ \partial_{x_i}\mathrm{e} \\ \vdots \\ \partial_{x_N}\mathrm{e} \end{bmatrix} \qquad \mathbf{H}_{\mathrm{e}}(\mathbf{x}) = \begin{bmatrix} \partial^2_{x_1x_1}\mathrm{e} & \cdots & \partial^2_{x_1x_i}\mathrm{e} & \cdots & \partial^2_{x_1x_N}\mathrm{e} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ \partial^2_{x_ix_1}\mathrm{e} & \cdots & \partial^2_{x_ix_i}\mathrm{e} & \cdots & \partial^2_{x_ix_N}\mathrm{e} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ \partial^2_{x_Nx_1}\mathrm{e} & \cdots & \partial^2_{x_Nx_i}\mathrm{e} & \cdots & \partial^2_{x_Nx_N}\mathrm{e} \end{bmatrix}$$

Hessian Matrix

- 2D Tensor
  - Associated to a quadratic form

$$\frac{1}{2}\boldsymbol{h}^{\mathrm{T}}\mathrm{H}_{\mathrm{e}}(\boldsymbol{x})\boldsymbol{h}$$

- Symmetric
  - Schwarz' theorem
  - *If a function has continuous n^{th}-order partial derivative, derivation order has no influence on the result.*

Derivatives in Dimension NxM

- M equations, N unknowns

$$\mathbf{e}(\boldsymbol{x}+\boldsymbol{h}) \simeq \mathbf{e}(\boldsymbol{x}) + \mathbf{J\_e}(\boldsymbol{x})\boldsymbol{h} + \mathrm{o}(\| \boldsymbol{h} \|_2)$$

Jacobian matrix

$$\mathbf{J}_e(\boldsymbol{x}) = \begin{bmatrix} \partial_{x_1} \mathrm{e}_1(\boldsymbol{x}) & \partial_{x_2} \mathrm{e}_1(\boldsymbol{x}) & \cdots & \partial_{x_N} \mathrm{e}_1(\boldsymbol{x}) \\ \partial_{x_1} \mathrm{e}_2(\boldsymbol{x}) & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \partial_{x_1} \mathrm{e}_N(\boldsymbol{x}) & \cdots & \cdots & \partial_{x_N} \mathrm{e}_N(\boldsymbol{x}) \end{bmatrix}$$

Jacobian Matrix

- Be careful: 1st order derivative only
  - Gradient for vector functions
  - Not a Hessian matrix
- Used for integration by substitution
  - $\mathbf{e}$ is a bijective vector function
  - $N = M$ (square matrix)

$$\iiint_D \mathrm{f}(\boldsymbol{x})\ \mathrm{d}x_1 \ldots \mathrm{d}x_N = \iiint_{\mathbf{e}^{-1}(D)} \mathrm{f}\big(\mathbf{e}(\boldsymbol{x})\big)\ |\det \mathbf{J}_{\mathbf{e}}(\boldsymbol{x})|\ \mathrm{d}y_1 \ldots \mathrm{d}y_N$$

Optimization: find $\nabla \mathrm{e}(x) = 0$

- 2nd order Taylor expansion

$$\mathrm{e}\left(\boldsymbol{x}^{(k)} + \boldsymbol{h}\right) \simeq \mathrm{e}\left(\boldsymbol{x}^{(k)}\right) + \boldsymbol{h}^{\mathrm{T}} \boldsymbol{\nabla} e\left(\boldsymbol{x}^{(k)}\right) + \frac{1}{2} \boldsymbol{h}^{\mathrm{T}} \mathbf{H}_{\mathrm{e}}\left(\boldsymbol{x}^{(k)}\right) \boldsymbol{h}$$

- Step estimation

$$\boldsymbol{h} = -\left(\mathbf{H}_{\mathrm{e}}\left(\boldsymbol{x}^{(k)}\right)\right)^{-1} \boldsymbol{\nabla} \mathrm{e}\left(\boldsymbol{x}^{(k)}\right)$$

- Iteration

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \left(\mathbf{H}_{\mathrm{e}}\left(\boldsymbol{x}^{(k)}\right)\right)^{-1} \boldsymbol{\nabla} \mathrm{e}\left(\mathbf{x}^{(k)}\right)$$

Limitation of Newton Method

- If the Hessian is not semi positive-definite
  - Each step increase the error !

Gradient Descent

- Follow the inclination of the function
  - Inclination = slope = gradient
- Compute how much in this direction

$$\boldsymbol{x}^{(k+1)} = \mathbf{x}^{(k)} + \rho\boldsymbol{\nabla}\mathrm{e}\left(\boldsymbol{x}^{(k)}\right) \text{ with } \rho \text{ such as } \mathrm{e}\left(\boldsymbol{x}^{(k+1)}\right) < \mathrm{e}\left(\boldsymbol{x}^{(k)}\right)$$

$$\min_{\rho} \mathrm{e}\left(\boldsymbol{x}^{(k)} + \rho\boldsymbol{\nabla}\mathrm{e}\left(\boldsymbol{x}^{(k)}\right)\right)$$

$$\partial_{\rho}\mathrm{e}\left(\boldsymbol{x}^{(k)} + \rho\boldsymbol{\nabla}\mathrm{e}\left(\boldsymbol{x}^{(k)}\right)\right) = 0 = \left(\boldsymbol{\nabla}\mathrm{e}\left(\boldsymbol{x}^{(k)}\right)\right)^{\mathrm{T}} \boldsymbol{\nabla}\mathrm{e}\left(\boldsymbol{x}^{(k)} + \rho\boldsymbol{\nabla}\mathrm{e}\left(\boldsymbol{x}^{(k)}\right)\right)$$

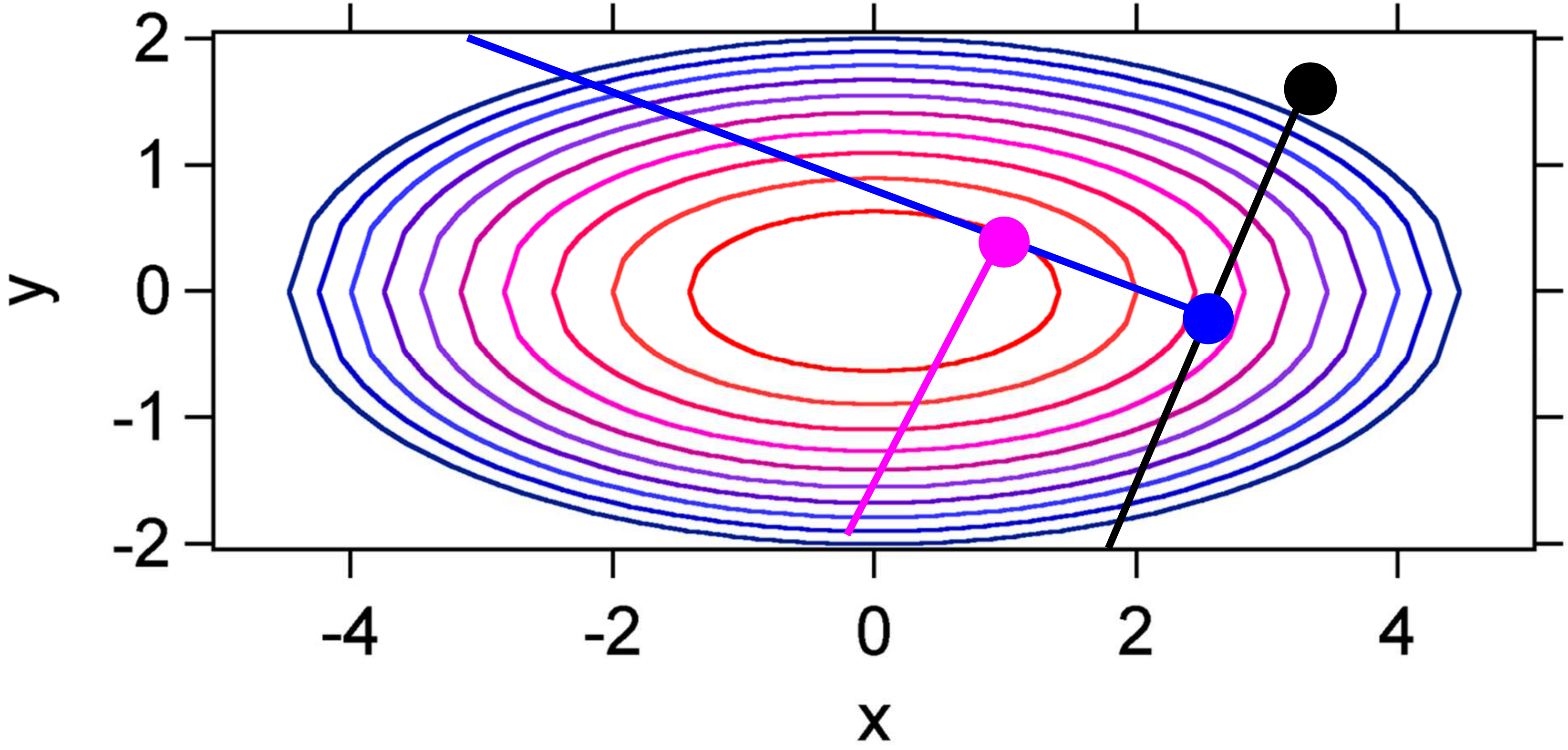
$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \frac{\left(\boldsymbol{\nabla}\mathrm{e}\left(\mathbf{x}^{(k)}\right)\right)^{\mathrm{T}} \boldsymbol{\nabla}\mathrm{e}\left(\boldsymbol{x}^{(k)}\right)}{\left(\boldsymbol{\nabla}\mathrm{e}\left(\boldsymbol{x}^{(k)}\right)\right)^{\mathrm{T}} \mathbf{H}_{\mathrm{e}}\left(\boldsymbol{x}^{(k)}\right)\boldsymbol{\nabla}\mathrm{e}\left(\boldsymbol{x}^{(k)}\right)} \boldsymbol{\nabla}\mathrm{e}\left(\boldsymbol{x}^{(k)}\right)$$

Gradient Descent

- Follow the inclination of the function
  - Inclination = slope = gradient
- Compute how much in this direction

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \frac{\left\|\boldsymbol{\nabla}\mathrm{e}\left(\boldsymbol{x}^{(k)}\right)\right\|_2^2}{\left\|\boldsymbol{\nabla}\mathrm{e}\left(\boldsymbol{x}^{(k)}\right)\right\|_{\mathbf{H}_\mathrm{e}\left(\boldsymbol{x}^{(k)}\right)}^2} \boldsymbol{\nabla}e\left(\boldsymbol{x}^{(k)}\right)$$

Gradient Descent



Gradient Descent : Convergence

- K:condition number of the Hessian
  - Convergence

$$\| \boldsymbol{x}^{(k)} - \boldsymbol{x} \|_{\mathbf{H}} \leq \left(\frac{K-1}{K+1} \right)^{k} \| \boldsymbol{x}^{(0)} - \mathbf{x} \|_{\mathbf{H}}$$

- **Starting point is very important**
  - As close as possible to the solution
- Remaining question
  - What is the best direction ?

Conjugate Gradient

- Basis
  - $\mathbf{H}$: symmetric positive-definite $\boldsymbol{x} \perp \boldsymbol{y} \Leftrightarrow \boldsymbol{x}^{\mathrm{T}}\mathbf{H}\boldsymbol{x} = 0$
  - Selecting pseudo-orthogonal direction
- For each step
  - Orthogonal direction (Gram-Schmidt)

$$\boldsymbol{d}_k = \boldsymbol{\nabla}_k - \sum_{i<k} \frac{\langle \boldsymbol{d}_i, \boldsymbol{\nabla}_k \rangle_{\mathbf{H}}}{\|\boldsymbol{d}_i\|_{\mathbf{H}}^2} \boldsymbol{d}_i$$

  - New step

$$h = \frac{\langle \boldsymbol{d}_k, \boldsymbol{\nabla}_k \rangle_{\mathbf{H}}}{\|\boldsymbol{d}_k\|_{\mathbf{H}}^2}$$

Convergences

- K:condition number of the Hessian
- Gradient descent

$$\| \boldsymbol{x}^{(k)} - \boldsymbol{x} \|_{\mathbf{H}} \leq \left(\frac{K-1}{K+1}\right)^{k} \| \boldsymbol{x}^{(0)} - \mathbf{x} \|_{\mathbf{H}}$$

- Conjugate gradient

$$\| \boldsymbol{x}^{(k)} - \boldsymbol{x} \|_{\mathbf{H}} \leq \left(\frac{\sqrt{K}-1}{\sqrt{K}+1}\right)^{k} \| \boldsymbol{x}^{(0)} - \mathbf{x} \|_{\mathbf{H}}$$

- Limitation: needs 2nd order derivatives

Residual-based form

- Residual $$\mathrm{r}_m(\boldsymbol{v}) = y_m - \mathrm{f}_{\boldsymbol{v}}(\boldsymbol{x}_m)$$
- Least-Square objective $$\mathrm{e}(\boldsymbol{v}) = \sum_{m=1}^{M} \mathrm{r}_m^2(\boldsymbol{v})$$
- Gradient $$\boldsymbol{\nabla}\mathrm{e} = 2\left(\mathbf{J}_{\boldsymbol{r}}^{\mathrm{T}}\boldsymbol{r}\right)$$

$$\mathbf{J}_{\boldsymbol{r}} = \begin{bmatrix} \partial_{v_1} r_1 & \partial_{v_2} r_1 & \cdots & \partial_{v_K} r_1 \\ \partial_{v_1} r_2 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \partial_{v_1} r_M & \cdots & \cdots & \partial_{v_K} r_N \end{bmatrix}$$

Residual-based form

- Residual $\mathrm{r}_m(\boldsymbol{v}) = y_m - \mathrm{f}_{\boldsymbol{v}}(\boldsymbol{x}_m)$
- Least-Square objective $\mathrm{e}(\boldsymbol{v}) = \sum_{m=1}^{M} \mathrm{r}_m^2(\boldsymbol{v})$
- Gradient $\boldsymbol{\nabla}\mathrm{e} = 2\left(\mathbf{J}_{\boldsymbol{r}}^{\mathrm{T}}\boldsymbol{r}\right)$
- Hessian matrix $\mathbf{H}\_\mathrm{e} = 2\left(\mathbf{J}_{\boldsymbol{r}}^{\mathrm{T}}\ \mathbf{J}_{\boldsymbol{r}} + \sum_{m=1}^{M} r_m \mathbf{H}_{r_m}\right)$

Gauss-Newton Method

- Idea: replacing the Hessian matrix

$$\mathbf{H}_{\mathrm{e}} \simeq 2\mathbf{J}_{\boldsymbol{r}}^{\mathrm{T}}\,\mathbf{J}_{\boldsymbol{r}} \qquad \boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \left(\mathbf{H}_{\mathrm{e}}\left(\boldsymbol{x}^{(k)}\right)\right)^{-1} \boldsymbol{\nabla}\mathrm{e}\left(\boldsymbol{x}^{(k)}\right)$$

- Advantages
  - No 2nd derivatives
  - Semi positive-definite matrix
- Limitations
  - Valid approximation for small residual
  - Carefull choice of initial values

Levenberg-Marquardt

- Most used method
- For each iteration, direction is

$$\left(\mathbf{J}_{\boldsymbol{r}}^{\mathrm{T}}\mathbf{J}_{\boldsymbol{r}} + \lambda\,\mathrm{diag}\,\left(\mathbf{J}_{\boldsymbol{r}}^{\mathrm{T}}\,\mathbf{J}_{\boldsymbol{r}}\right)\right)\boldsymbol{h} = \mathbf{J}_{\boldsymbol{r}}^{\mathrm{T}}\boldsymbol{r}$$

- Choice of λ is balancing the behavior
  - Small: Newton
  - Large: fast gradient descent

- More robust to noise