



Lecture 05 Homogenous Linear System & Least Squares Problem

EE382-Visual localization & Perception

Danping Zou,

Associate Professor

Institute for Navigation and Sensing



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY



Outline

- Homogenous linear system
 - Basics of Linear algebra
 - Range
 - Null space
 - Rank
 - Singular Value Decomposition (SVD)
- Nonlinear least squares problem
 - Vector derivative, Jacobian matrix, Gradient
 - Taylor series expansion
 - Linear least squares problem
 - Gauss-Newton solver
 - Levinberg-Marquardt solver



Homogeneous linear system



- A homogeneous system is equivalent to a matrix equation of the form :

$$\mathbf{A}\mathbf{x} = \mathbf{0}$$

- Every homogeneous system has at least one solution, known as the **zero solution**
- If \mathbf{A} is a singular matrix, there are infinite non-trivial solutions, they span the **null space** of \mathbf{A} .



About the null space & SVD



- Fundamentals
 - **Range (column space)** : $\text{range}(\mathbf{A})$ is the space spanned by the columns of the matrix \mathbf{A} .

$$\mathbf{Ax} = x_1 \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} + x_2 \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} + \cdots + x_n \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}$$

- **Null space** : $\text{null}(\mathbf{A})$ is a space where all the vectors satisfy

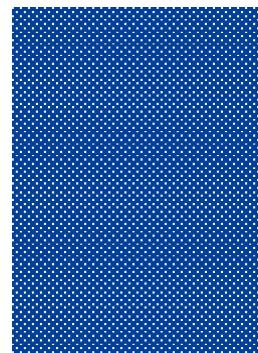
$$\mathbf{Ax} = 0$$

- **Rank** : The rank is the dimension of the column space



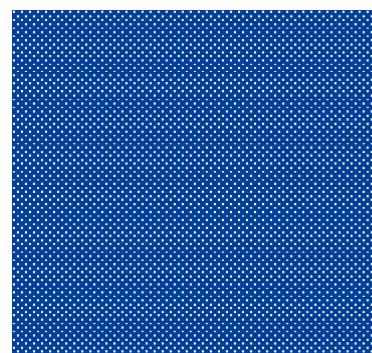
About the null space & SVD

- Use SVD to compute the null space of a matrix.
- The **Singular Value Decomposition (SVD)** is a basic tool to analysis a matrix.
- Many problems of linear algebra can be better understood if we first ask the question: **what if we take the SVD ?**

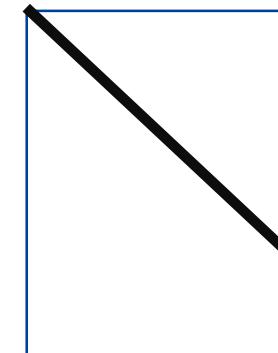


M

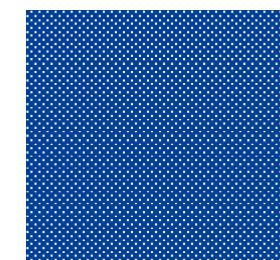
=



U



Σ

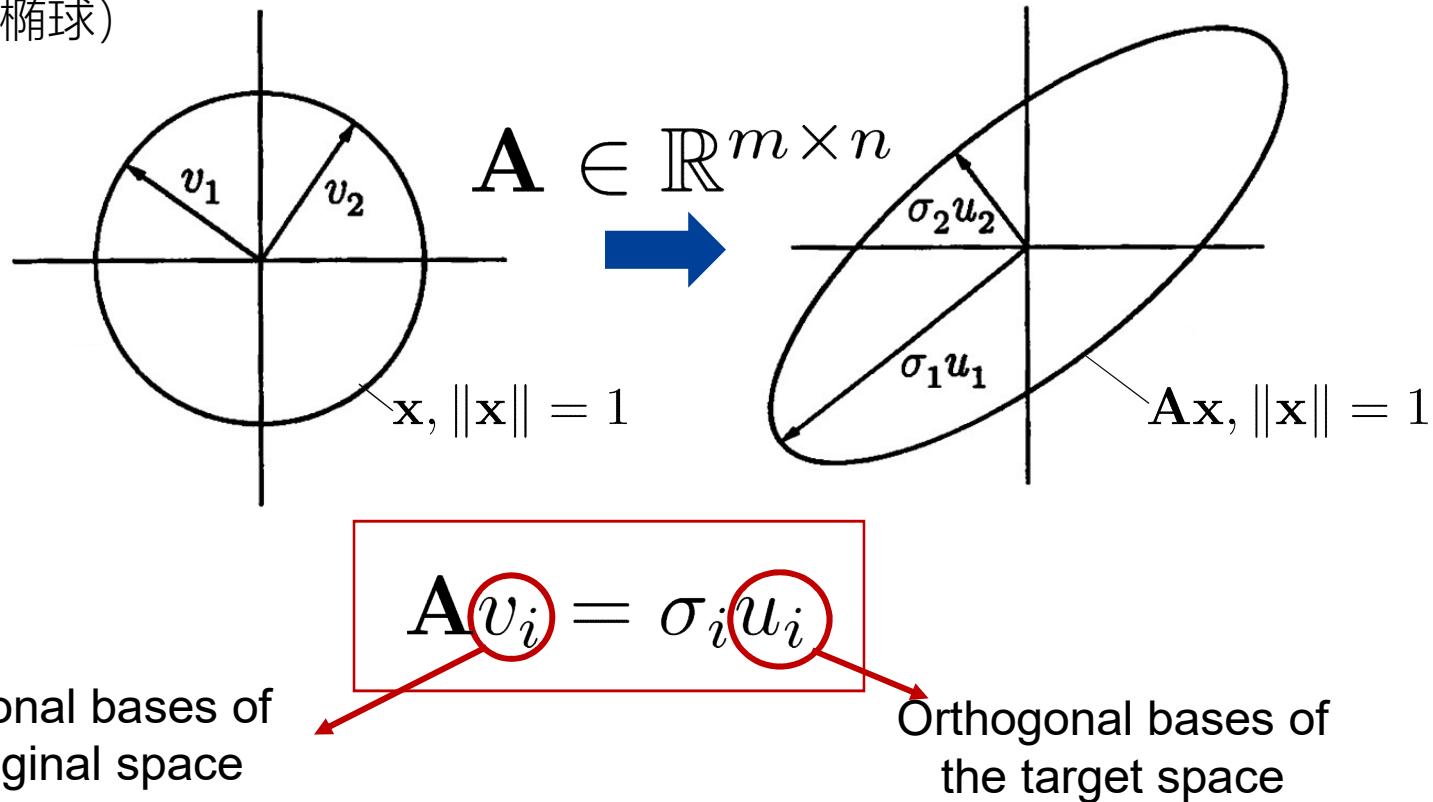


V^T



SVD

- The SVD is motivated by the following geometric fact:
 - The image of the unit sphere under **any** $m \times n$ matrix is a hyper-ellipse (超椭球)





SVD



- From the geometric interpretation, we have the following equations

$$\mathbf{A}v_i = \sigma_i u_i$$

- By collecting all those equations together, we have

$$[\mathbf{A}] \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} = [\mathbf{u}_1 \mathbf{u}_2 \cdots \mathbf{u}_n] \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix}$$

$$\rightarrow \mathbf{AV} = \mathbf{U}\Sigma$$

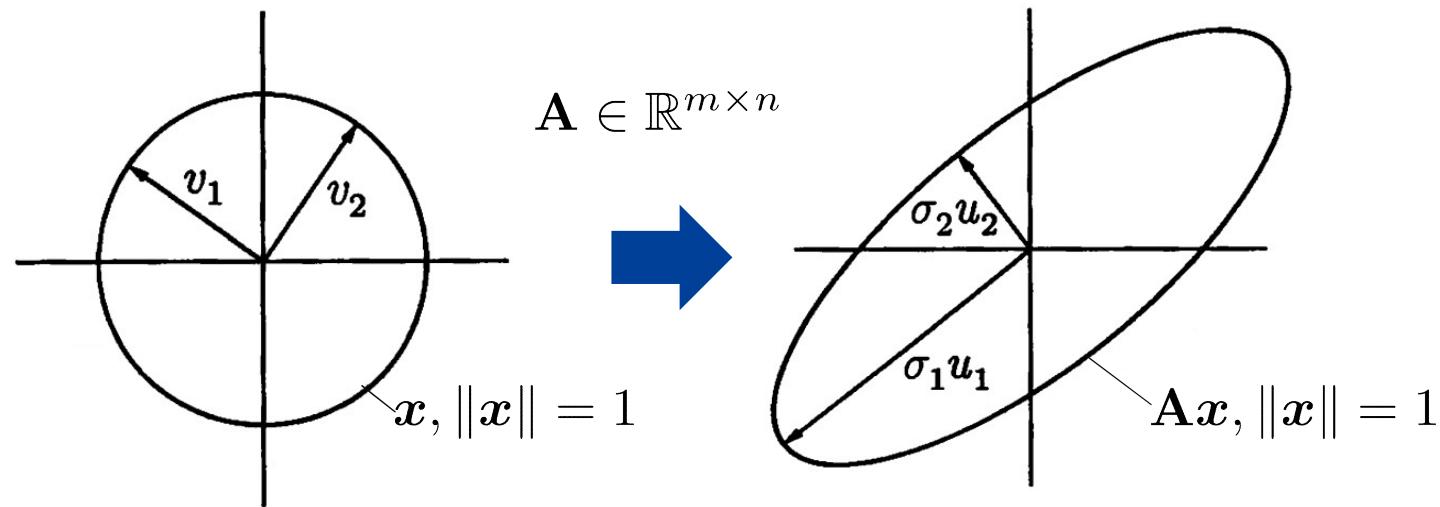
Since \mathbf{V} has orthogonal columns, we get

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$$



SVD

- Back to the 2D case, we have



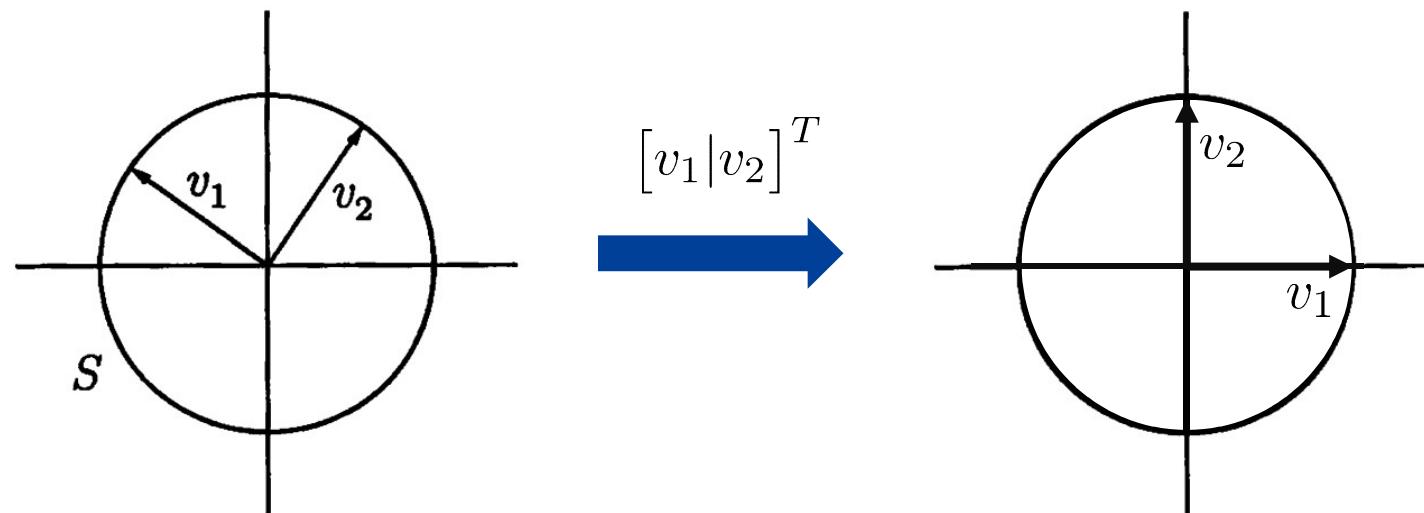
$$A = [u_1 \mid u_2] \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} [v_1 \mid v_2]^T$$

Left singular vectors Singular values Right singular vectors



SVD

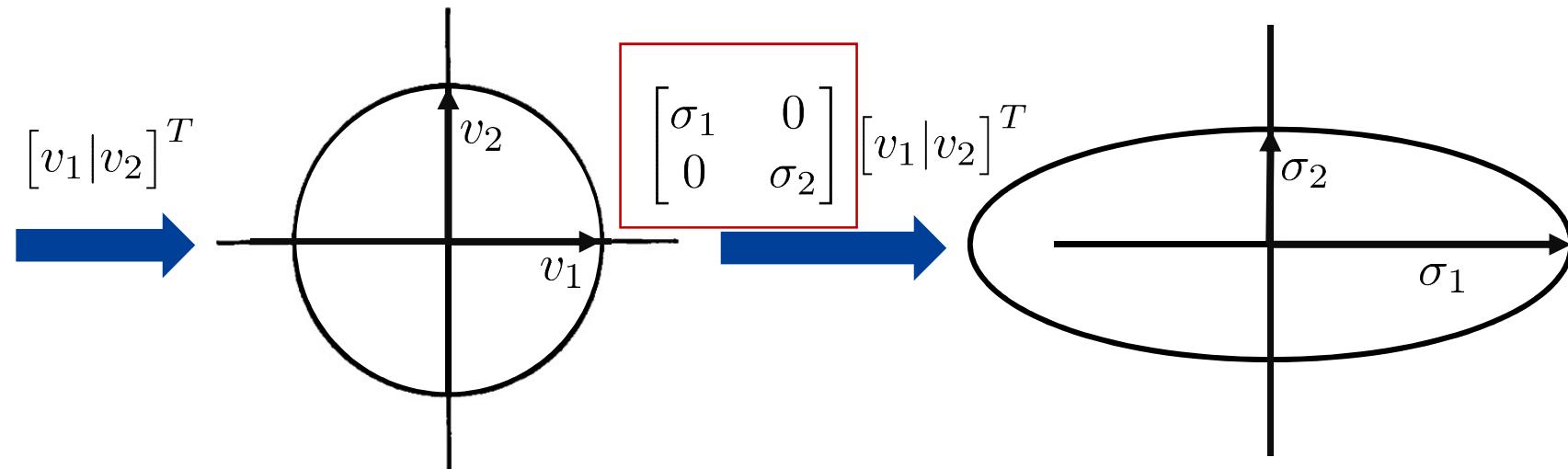
- Step by step interpretation
 - Step 1. map (rotation + with/without reflection) the bases $\{v_i\}$ to the **canonical frame**.





SVD

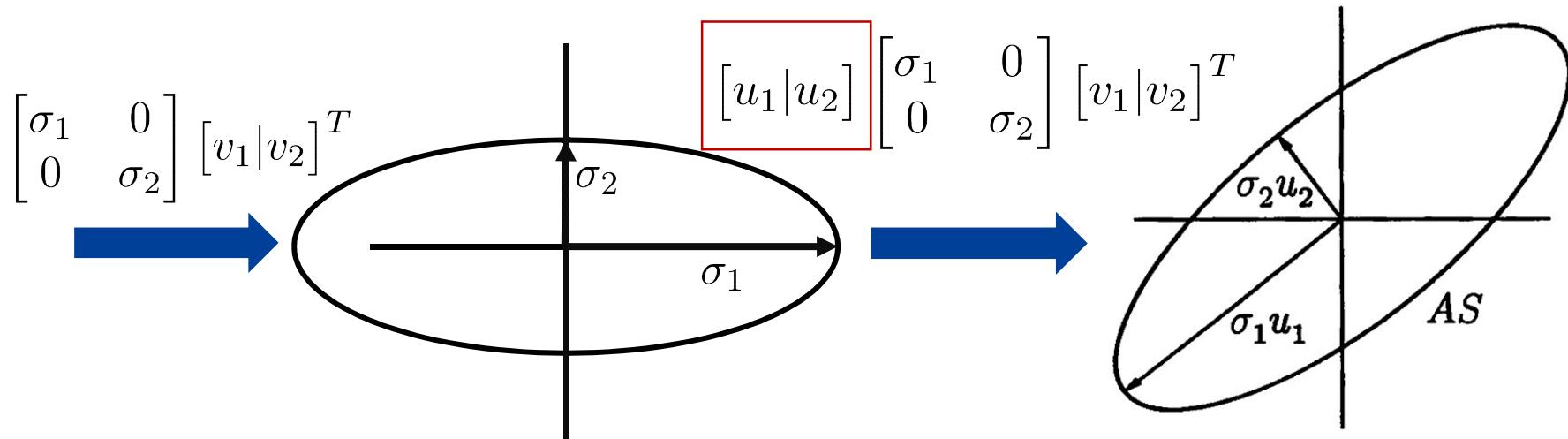
- Step by step interpretation
 - Step 2. Stretch the axes





SVD

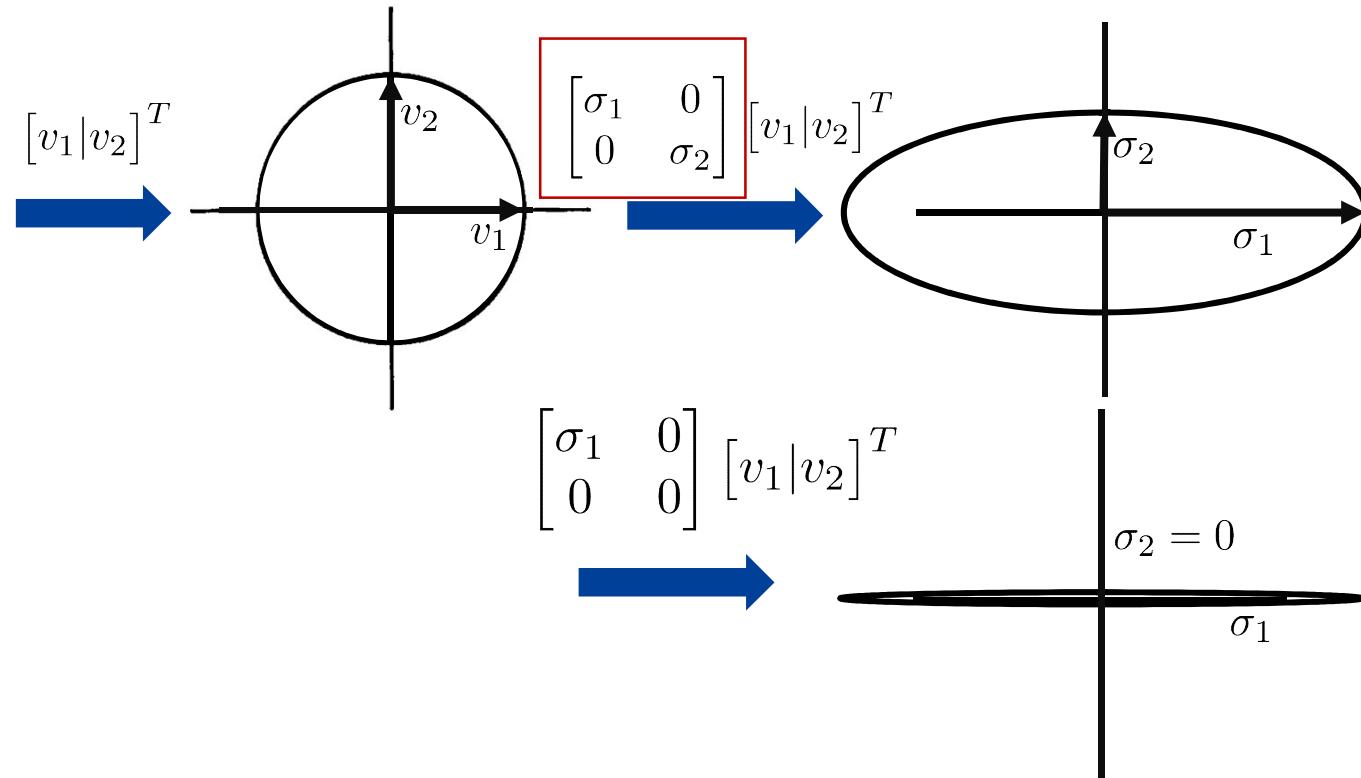
- Step by step interpretation
 - Step 3. rotate the canonical axes





SVD

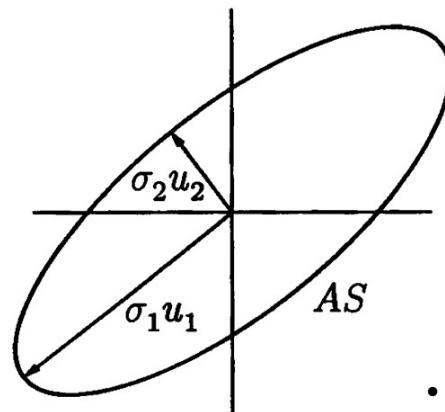
- In the step 2, what if we set $\sigma_2 = 0$?





SVD

- After transformation into the target space



$$\begin{bmatrix} \sigma_1 & 0 \\ 0 & 0 \end{bmatrix} [v_1 | v_2]^T \rightarrow \begin{array}{c} \text{---} \\ \sigma_2 = 0 \\ \text{---} \\ \sigma_1 \end{array}$$

$$\boxed{[u_1 | u_2]} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} [v_1 | v_2]^T \rightarrow \begin{array}{c} \text{---} \\ \sigma_2 = 0 \\ \text{---} \\ \sigma_1 \end{array}$$

- We have : $\mathbf{A}v_2 = 0$
 $\text{span}\{v_2\}$ is the null space of \mathbf{A}



Null space from SVD

- So the null space of a matrix can be obtained by SVD decomposition.
- The bases of the null space are the **right singular vectors that corresponds to the zero singular values**.

$$\mathbf{A} = [u_1 | u_2 | \cdots | u_m] \begin{bmatrix} \sigma_1 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} [v_1 | v_2 | \cdots | v_n]^T$$

A red arrow points from the right singular vector $[v_1 | v_2 | \cdots | v_n]^T$ to the zero singular value row in the diagonal matrix.



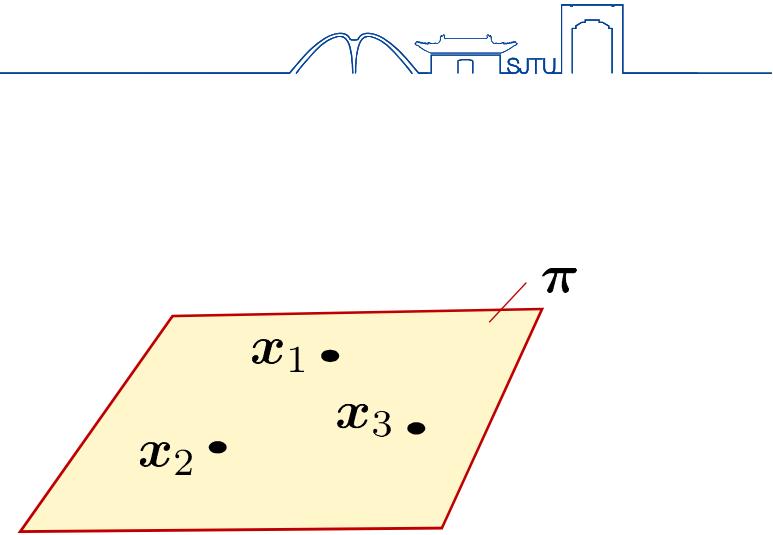
Examples

- A plane from three points

$$\begin{aligned}\pi^T x_1 &= 0 \\ \pi^T x_2 &= 0 \\ \pi^T x_3 &= 0\end{aligned} \quad \rightarrow \quad \begin{bmatrix}x_1^T \\ x_2^T \\ x_3^T\end{bmatrix} \quad \begin{matrix}\pi = 0 \\ \mathbb{R}^{4 \times 1}\end{matrix}$$

\downarrow

$$\mathbb{R}^{3 \times 4}$$





Example

- Back to the plane problem: Three points define a plane :

$$\begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \mathbf{x}_3^T \end{bmatrix} \boldsymbol{\pi} = 0 \quad \rightarrow \quad \mathbf{A}\boldsymbol{\pi} = 0$$

- Its null space can be solved by SVD

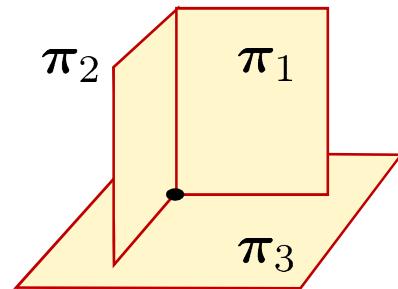
$$\mathbf{M} = [\mathbf{u}_1 | \mathbf{u}_2 | \mathbf{u}_3] \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} [\mathbf{v}_1 | \mathbf{v}_2 | \mathbf{v}_3 | \mathbf{v}_4]^T$$

→ $\boldsymbol{\pi} \sim \mathbf{v}_4$



Example

- Three planes define a point



$$\begin{aligned}\pi_1 x &= 0 \\ \pi_2 x &= 0 \\ \pi_3 x &= 0\end{aligned}$$



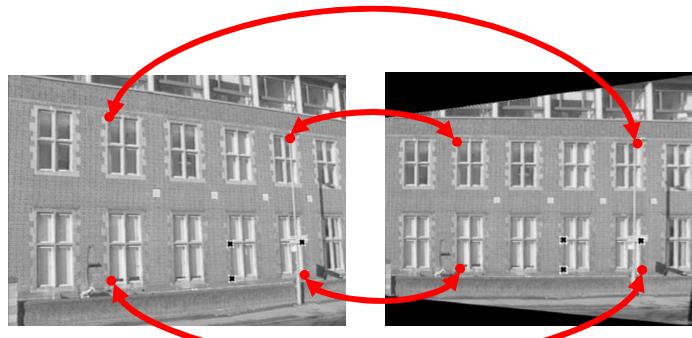
$$\begin{bmatrix} \pi_1^T \\ \pi_2^T \\ \pi_3^T \end{bmatrix} x = 0 \quad \begin{matrix} \downarrow \\ \mathbb{R}^{4 \times 1} \end{matrix}$$
$$\mathbb{R}^{3 \times 4}$$





Example

- Homography transformation



$$\mathbf{H} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$$

$$x'_1 \sim \mathbf{H}x_1$$

$$x'_2 \sim \mathbf{H}x_2$$

$$x'_3 \sim \mathbf{H}x_3$$

$$x'_4 \sim \mathbf{H}x_4$$

$$x'_1 \times \mathbf{H}x_1 = 0$$

$$x'_2 \times \mathbf{H}x_2 = 0$$

$$x'_3 \times \mathbf{H}x_3 = 0$$

$$x'_4 \times \mathbf{H}x_4 = 0$$



$$A$$

$$\begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = 0$$



$$\mathbf{H}$$



Summary



- Singular Value Decomposition $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$

- Geometric interpretation of SVD : Describes the behavior of the transformation $\mathbf{A} : \mathbf{V} \rightarrow \mathbf{U}$

- How to solve a homogenous equation
 - Find the null space of a matrix



Nonlinear least squares problem

- Find a minimizer x^* of the nonlinear objective function $f(x)$:

$$f(x) = \frac{1}{2} \|r(x)\|^2 = \frac{1}{2} \sum_{i=1}^m r_i(x)^2$$

- where $x \in \mathbb{R}^n$,

$$r(x) = \begin{pmatrix} r_1(x) \\ \vdots \\ r_m(x) \end{pmatrix} \in \mathbb{R}^m$$

$$r_i(x) = y_i - h_i(x), \quad i = 1, \dots, m$$

- Here y_i are the measured data. The nonlinearity arises from the **observation function** $h_i(x)$.



Nonlinear least squares problem

- The objective function can be also written as

$$f(\mathbf{x}) = \frac{1}{2}(\mathbf{y} - \mathbf{h}(\mathbf{x}))^T(\mathbf{y} - \mathbf{h}(\mathbf{x}))$$

where $\mathbf{y} = [y_1, \dots, y_i, \dots]^T$ and $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \dots, h_i(\mathbf{x}), \dots]^T$

- The basic idea of solving a nonlinear least squares problem is to
 - approximate the cost function as a linear function locally
 - and minimize the linear cost function by solving the **linear least squares problem**;
 - repeat above steps until the optimum value is reached.



Derivative & Jacobian matrix



- **Vector derivative** – Let $\mathbf{x} = (x_1 \cdots x_n)^T \in \mathbb{R}^n$, The $1 \times n$ vector derivative operator is denoted by:

$$\frac{\partial}{\partial \mathbf{x}} = \left(\frac{\partial}{\partial x_1} \cdots \frac{\partial}{\partial x_n} \right)$$

- For a general m -dimensional nonlinear **vector function** $\mathbf{h}(\mathbf{x}) \in \mathbb{R}^m$, we define :

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{h}(\mathbf{x}) \triangleq \begin{pmatrix} \frac{\partial}{\partial \mathbf{x}} h_1(\mathbf{x}) \\ \vdots \\ \frac{\partial}{\partial \mathbf{x}} h_m(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial x_1} h_1(\mathbf{x}) & \cdots & \frac{\partial}{\partial x_n} h_1(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial x_1} h_m(\mathbf{x}) & \cdots & \frac{\partial}{\partial x_n} h_m(\mathbf{x}) \end{pmatrix} \in \mathbb{R}^{m \times n}$$

This is known as the **Jacobian matrix**.



Computation of Jacobian matrix

- The following results are very useful :

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{a}^T \mathbf{x} = \frac{\partial}{\partial \mathbf{x}} \mathbf{x}^T \mathbf{a} = \mathbf{a}^T$$

(Inner product)

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{A} \mathbf{x} = \mathbf{A}$$

(Matrix-vector product)

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{x}^T \Omega \mathbf{x} = 2\mathbf{x}^T \Omega = 2\Omega \mathbf{x}^T$$

$(\Omega = \Omega^T)$

(Weighted 2-norm squared)

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{h}(\mathbf{g}(\mathbf{x})) = \frac{\partial \mathbf{h}}{\partial \mathbf{g}} \frac{\partial \mathbf{g}}{\partial \mathbf{x}}$$

(Chain rule)



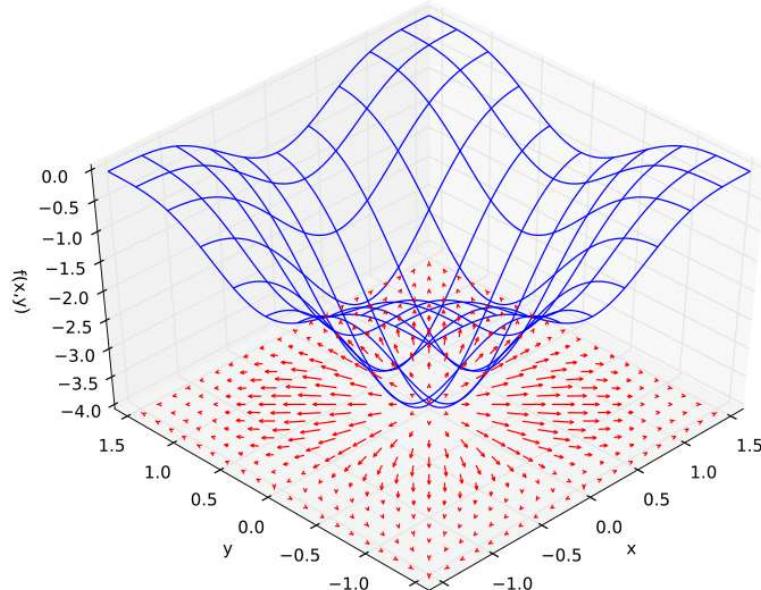
Gradient



- The **gradient** of a **scalar-valued function** $f(x)$ is defined by

$$\nabla_x f(x) \triangleq \left(\frac{\partial}{\partial x} f(x) \right)^T$$

- The **gradient** gives that local ***direction of steepest ascent*** (increase in value) in the domain of definition.



The gradients of the function

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

are indicated by the vector fields on the bottom plane [from Wikipedia]



Taylor Series Expansion



- Recall the Taylor series expansion of a function $f(x)$ about a point x_0 as

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \frac{f'''(x_0)}{3!}(x - x_0)^3 + \dots$$

- The first-order expansion is

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0)$$

- The second-order expansion is

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2$$



Taylor Series Expansion



- The Taylor series expansion of a vector-valued function $\mathbf{h}(x)$ about a point x_0 as (First-order approximation)

$$\mathbf{h}(x) = \mathbf{h}(x_0 + \Delta x) \approx \mathbf{h}(x_0) + \frac{\partial \mathbf{h}(x_0)}{\partial x} \Delta x$$

- Let $\Delta \mathbf{y} = \mathbf{h}(x) - \mathbf{h}(x_0)$, the Jacobian matrix $\mathbf{H}(x) \triangleq \frac{\partial \mathbf{h}(x)}{\partial x}$ provides the *linearization* of $\mathbf{h}(x)$,

$$\Delta \mathbf{y} \approx \mathbf{H}(x_0) \Delta x$$



Taylor Series Expansion



- The **Taylor series expansion** of a scalar-valued function $f(\mathbf{x})$ is written as (second order)

$$f(\mathbf{x}_0 + \Delta\mathbf{x}) \approx f(\mathbf{x}_0) + \frac{\partial f(\mathbf{x}_0)}{\partial \mathbf{x}} \Delta\mathbf{x} + \frac{1}{2} \Delta\mathbf{x}^T \mathcal{H}(\mathbf{x}_0) \Delta\mathbf{x}$$

- $\mathcal{H}(\mathbf{x}_0)$ denotes the **Hessian matrix** of second-order partial derivatives,

$$\mathcal{H}(\mathbf{x}_0) \triangleq \nabla^2 f(\mathbf{x}) = \frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x}^2} = \left[\frac{\partial f(\mathbf{x})}{\partial x_i \partial x_j} \right]$$



Summary

- Jacobian matrix, for a vector function $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{h}(\mathbf{x}) \triangleq \begin{pmatrix} \frac{\partial}{\partial x_1} h_1(\mathbf{x}) & \cdots & \frac{\partial}{\partial x_n} h_1(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial x_1} h_m(\mathbf{x}) & \cdots & \frac{\partial}{\partial x_n} h_m(\mathbf{x}) \end{pmatrix} \in \mathbb{R}^{m \times n}$$

- Gradient, for a scalar function $f : \mathbb{R}^n \rightarrow \mathbb{R}$

$$\nabla_{\mathbf{x}} f(\mathbf{x}) \triangleq \left(\frac{\partial}{\partial \mathbf{x}} f(\mathbf{x}) \right)^T$$

- Tayler Series Expansion

$$f(\mathbf{x}_0 + \Delta \mathbf{x}) \approx f(\mathbf{x}_0) + \frac{\partial f(\mathbf{x}_0)}{\partial \mathbf{x}} \Delta \mathbf{x} + \frac{1}{2} \Delta \mathbf{x}^T \mathcal{H}(\mathbf{x}_0) \Delta \mathbf{x}$$

- Hessian matrix

$$\mathcal{H}(\mathbf{x}_0) \triangleq \nabla^2 f(\mathbf{x}) = \left[\frac{\partial f(\mathbf{x})}{\partial x_i \partial x_j} \right]$$



Linear Least squares problem



- A linear least squares problem is to solve

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (\mathbf{A} \in \mathbb{R}^{m \times n})$$

where the $m > n$.

- Let the sum of the differences between \mathbf{b} and $\mathbf{A}\mathbf{x}$ be as small as possible. Let $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$. We want to minimize the sum of squares

$$\mathbf{r}^T \mathbf{r} = \sum_i r_i^2$$

- That is, to minimize

$$f(\mathbf{x}) = \frac{1}{2}(\mathbf{b} - \mathbf{A}\mathbf{x})^T(\mathbf{b} - \mathbf{A}\mathbf{x})$$



Linear Least squares problem



- The cost function of a linear least squares problem is defined as

$$f(\mathbf{x}) = \frac{1}{2}(\mathbf{b} - \mathbf{Ax})^T(\mathbf{b} - \mathbf{Ax})$$

- The optimum is obtained by let

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \mathbf{0}$$

$$\frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{A}^T \mathbf{Ax} - 2\mathbf{b}^T \mathbf{Ax} + \mathbf{b}^T \mathbf{b}) = \mathbf{0}$$

$$2\mathbf{A}^T \mathbf{Ax} - 2\mathbf{b}^T \mathbf{A} = \mathbf{0}$$

$$\mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}$$

$$\boxed{\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}}$$



Linear Least squares problem

- The solution to the linear least squares problem

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (\mathbf{A} \in \mathbb{R}^{m \times n})$$

$$\mathbf{A}^T \mathbf{A}\mathbf{x} = \mathbf{A}^T \mathbf{b} \quad (\text{normal equation})$$

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$



Nonlinear least squares problem

- The objective function can be also written as

$$f(\mathbf{x}) = \frac{1}{2}(\mathbf{y} - \mathbf{h}(\mathbf{x}))^T(\mathbf{y} - \mathbf{h}(\mathbf{x}))$$

where $\mathbf{y} = [y_1, \dots, y_i, \dots]^T$ and $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \dots, h_i(\mathbf{x}), \dots]^T$

- The basic idea of solving a nonlinear least squares problem is to
 - approximate the cost function as a linear function locally
 - and minimize the linear cost function by solving the **linear least squares problem**;
 - repeat above steps until the optimum value is reached.



Gauss-Newton method



- The cost function of the non-linear least squares problem:

$$\begin{aligned}f(\mathbf{x}) &= \frac{1}{2} \|\mathbf{r}(\mathbf{x})\|^2 = \frac{1}{2} \mathbf{r}(\mathbf{x})^T \mathbf{r}(\mathbf{x}) \\&= \frac{1}{2} (\mathbf{y} - \mathbf{h}(\mathbf{x}))^T (\mathbf{y} - \mathbf{h}(\mathbf{x}))\end{aligned}$$

- If we use the first-order Taylor expansion to approximate $\mathbf{h}(\mathbf{x})$ around \mathbf{x}_0

$$\mathbf{h}(\mathbf{x}_0 + \Delta\mathbf{x}) \approx \mathbf{h}(\mathbf{x}_0) + \mathbf{H}(\mathbf{x}_0)\Delta\mathbf{x}$$

Step1 : approximate the cost function as a linear function locally



Gauss-Newton method



- So the objective function can be approximated by

$$\begin{aligned} f(\mathbf{x}_0 + \Delta\mathbf{x}) &= \frac{1}{2}(\mathbf{y} - \mathbf{h}(\mathbf{x}_0 + \Delta\mathbf{x}))^T(\mathbf{y} - \mathbf{h}(\mathbf{x}_0 + \Delta\mathbf{x})) \\ &\approx \frac{1}{2}(\mathbf{r}(\mathbf{x}_0) - \mathbf{H}(\mathbf{x}_0)\Delta\mathbf{x})^T(\mathbf{r}(\mathbf{x}_0) - \mathbf{H}(\mathbf{x}_0)\Delta\mathbf{x}) \\ &\quad (\mathbf{r}(\mathbf{x}_0) = \mathbf{y} - \mathbf{h}(\mathbf{x}_0)) \end{aligned}$$

- We want to find a $\Delta\mathbf{x}$ that minimize the objective function locally
 - This is a standard linear least squares problem:

$$\frac{\partial}{\partial \Delta\mathbf{x}} f(\mathbf{x} + \Delta\mathbf{x}) = -\mathbf{H}^T \mathbf{r} + \mathbf{H}^T \mathbf{H} \Delta\mathbf{x} = \mathbf{0}$$

$$\Delta\mathbf{x} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{r}$$

Step2 : minimize the linear cost function by solving the **linear least squares problem**



Gauss-Newton method



- Step 1 : Start from an initial point x_0
- Step 2 : solve an incremental step Δx ,

$$\Delta x = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{r}$$

- Step 3 : Update the solution $x \leftarrow x + \Delta x$
- *Repeat Step 2~Step 3 until convergence.*
- Practical issues:
 - The initial guess x_0 should be close to the real solution.
 - $(\mathbf{H}^T \mathbf{H})$ may be ill-conditioned or not positive-definite.



Levenberg-Marquardt method



- Add a damping term to limit the length of step at each iteration.

$$\Delta\boldsymbol{x}^{LM} \leftarrow \underbrace{\min f(\boldsymbol{x} + \Delta\boldsymbol{x}) + \lambda \|\Delta\boldsymbol{x}\|^2}_{f'(\boldsymbol{x} + \Delta\boldsymbol{x})}$$

- Solve $\Delta\boldsymbol{x}$ by minimizing the new local cost function:

$$\frac{\partial}{\Delta\boldsymbol{x}} f'(\boldsymbol{x} + \Delta\boldsymbol{x}) = 0$$

$$\Delta\boldsymbol{x}^{LM} = (\mathbf{H}^T \mathbf{H} + \lambda \mathbf{I})^{-1} \mathbf{H}^T \mathbf{r}$$



About the damping parameter

- The damping parameter λ has several effects:
 1. it ensures the positive definite and avoids bad condition of the coefficient matrix: $\mathbf{A} = (\mathbf{H}^T \mathbf{H} + \lambda \mathbf{I})$
 2. For large values of λ we get $\Delta \mathbf{x}^{LM} \approx \mathbf{H}^T \mathbf{r} / \lambda = -\nabla f / \lambda$
It is a short step in the steepest descent direction.
 3. If λ is small, the step is close to the one obtained Gauss-Newton method, which is good in the final stages of iteration, which $\mathbf{x} \rightarrow \mathbf{x}^*$



Choose the damping parameter

- The incremental of the objective function predicted by the linear model is given by $l(\Delta x) = (r - \mathbf{H}\Delta x)^T(r - \mathbf{H}\Delta x) - r^T r$

$$= \Delta x^T \mathbf{H}^T \mathbf{H} \Delta x - 2\Delta x^T \mathbf{H}^T r$$

- The incremental predicted by the LM step is computed as

$$l(\Delta x^{LM}) = (\Delta x^{LM})^T (\mathbf{H}^T \mathbf{H} \Delta x^{LM} - 2\mathbf{H}^T r)$$



$$(\mathbf{H}^T \mathbf{H} + \lambda \mathbf{I}) \Delta x^{LM} = \mathbf{H}^T r$$

$$= (\Delta x^{LM})^T (\lambda \Delta x^{LM} + \mathbf{H}^T r)$$



Choose the damping parameter



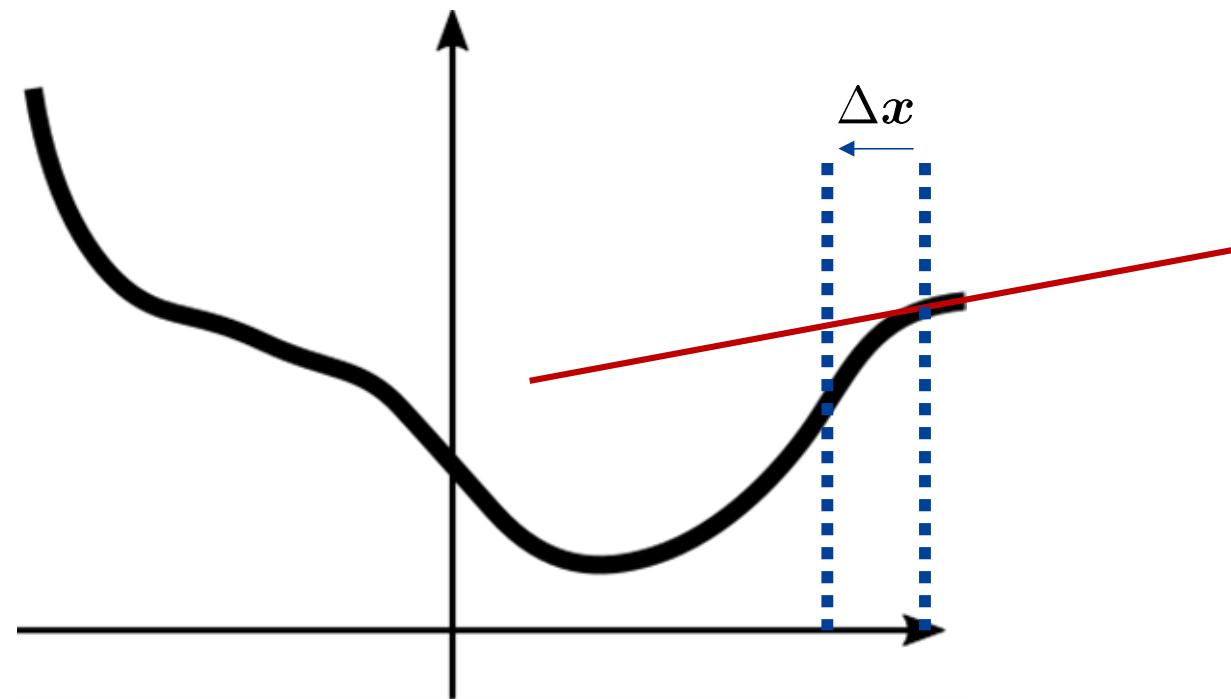
- The updating is controlled by the *gain ratio*

$$\rho = \frac{f(\mathbf{x} + \Delta\mathbf{x}^{LM}) - f(\mathbf{x})}{l(\Delta\mathbf{x}^{LM})}$$

- A large value of gain ratio indicates the current linear model is a good approximation to the objective function
 - **decrease** λ at the next LM step to get closer to the Gauss-Newton direction.
- A small value indicates a poor approximation
 - **increase** λ to reduce the step length and get closer to the steepest descent direction.



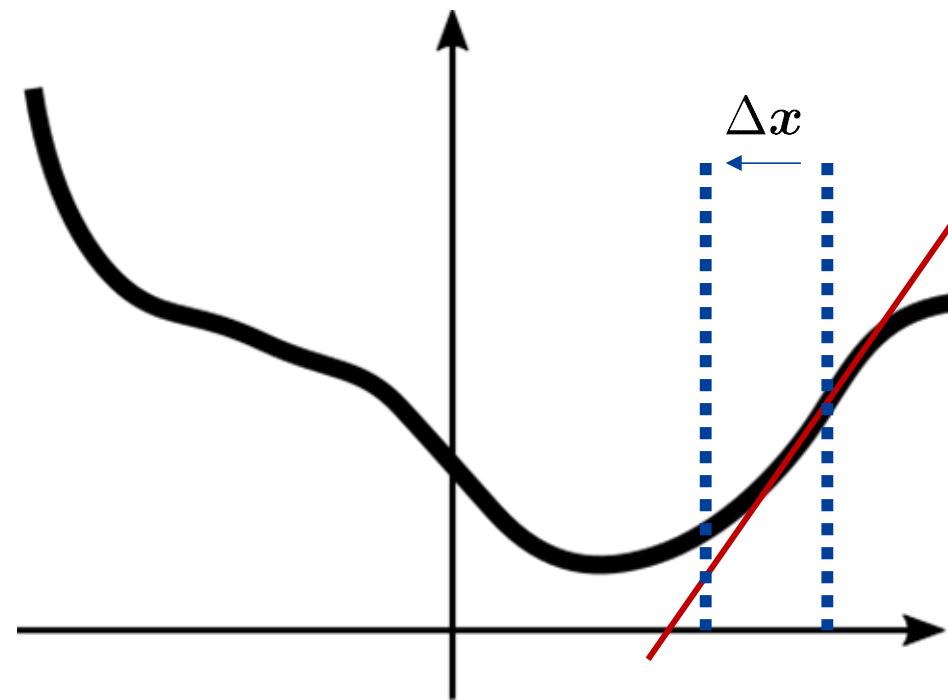
Choose the damping parameter



$\rho > 1$ Significantly increase the step length = decrease λ largely



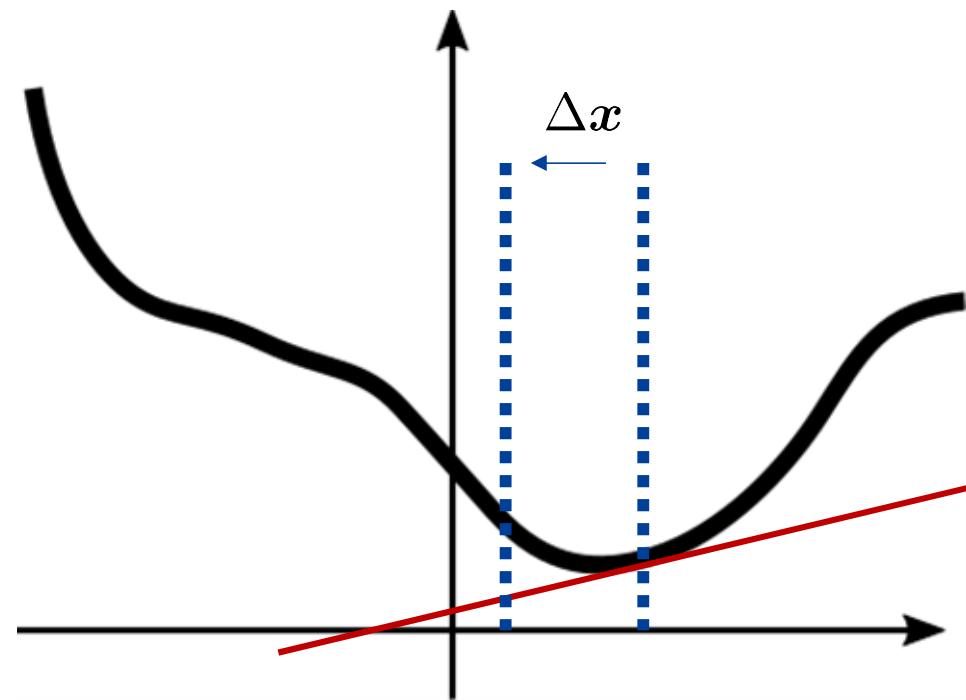
Choose the damping parameter



$1 > \rho > 0$ Increase step length = decrease λ



Choose the damping parameter



$0 > \rho$ decrease step length = increase λ



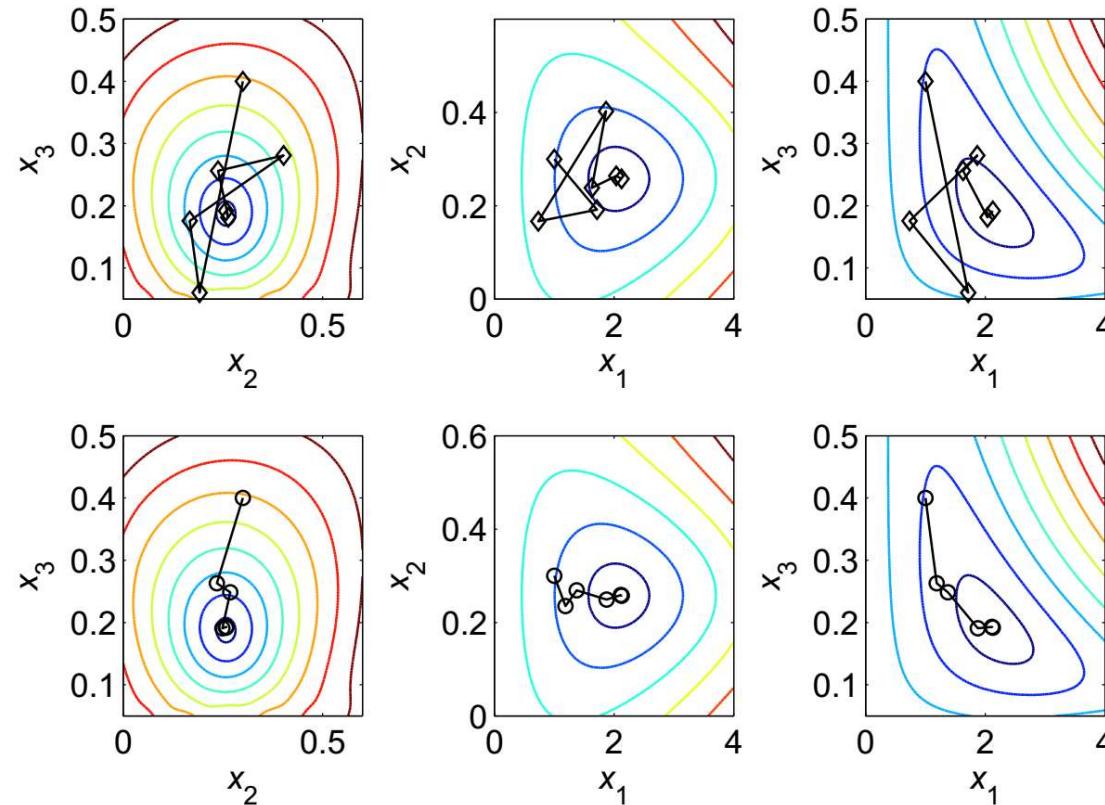
Levenberg-Marquardt algorithm

- Initialization: $\mathbf{A} = \mathbf{H}^T \mathbf{H}$, $\lambda = \max\{a_{ii}\}$
- Repeat until the step length vanishes, $\|\Delta \mathbf{x}^{LM}\| \rightarrow 0$, or the gradient of $f(\mathbf{x})$ vanishes, $\nabla f = -\mathbf{H}^T \mathbf{r} \rightarrow 0$:
 - a) Solve $(\mathbf{A} + \lambda \mathbf{I})\Delta \mathbf{x} = \mathbf{H}^T \mathbf{r}$ to get $\Delta \mathbf{x}^{LM}$
 - b) $\mathbf{x} \leftarrow \mathbf{x} + \Delta \mathbf{x}^{LM}$
 - c) Adjust the damping parameter by checking the *gain ratio*
 1. $\rho > 0$: Good approximation, **decrease** the damping parameter
 2. $\rho \leq 0$: Bad approximation, **increase** the damping parameter



Gauss-Newton v.s. Levenberg-Marquardt

- Top row (Gauss-Newton) and Bottom row (Levenberg-Marquardt)





Summary

- Linear least squares problem: $\mathbf{A}\mathbf{x} = \mathbf{b}$

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$

- Gauss-Newton method $\mathbf{x} \leftarrow \mathbf{x} + \Delta\mathbf{x}$

$$\Delta\mathbf{x} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{r}$$

- Levenberg-Marquardet method $\mathbf{x} \leftarrow \mathbf{x} + \Delta\mathbf{x}$

$$\Delta\mathbf{x} \leftarrow (\mathbf{H}^T \mathbf{H} + \lambda \mathbf{I})^{-1} \mathbf{H}^T \mathbf{r}$$



About parameter perturbations



- If the parameter is not in a vector space, it cannot be updated by a simple vector addition :

$$\boldsymbol{x} \leftarrow \boldsymbol{x} + \Delta \boldsymbol{x}$$

- Sometimes, the dimension of the parameter space may not be the same as the dimension of the perturbation space:

$$\boldsymbol{x} \in \mathbb{R}^m; \quad \Delta \boldsymbol{x} \in \mathbb{R}^n$$

$$m \neq n$$



Parameter perturbations

- We define the operator \boxplus to represent “**adding**” a perturbation to our parameters.
- Suppose the parameter space is denoted by \mathcal{X} and the perturbation space is \mathbb{R}^n . The operator is a mapping defined as :

$$\boxplus : \mathcal{X} \times \mathbb{R}^n \rightarrow \mathcal{X}$$



Parameter perturbations



- If the parameter space is a vector space, $\mathcal{X} = \mathbf{R}^n$,

$$\boldsymbol{x} \boxplus \Delta\boldsymbol{x} = \boldsymbol{x} + \Delta\boldsymbol{x}$$

- If the parameter is a Lie group (like rotation).

$$\begin{aligned}\boldsymbol{x} \boxplus \Delta\boldsymbol{x} &= \boldsymbol{x} \otimes \exp(\Delta\boldsymbol{x}^\wedge) \quad (\text{Right multiplication}) \\ &= \exp(\Delta\boldsymbol{x}^\wedge) \otimes \boldsymbol{x} \quad (\text{Left multiplication})\end{aligned}$$

- $\wedge : \mathbb{R}^n \rightarrow so(n)$ is an one-to-one mapping from the perturbation vector space to *lie algebra*. For a rotation, it is defined as

$$\boldsymbol{a}^\wedge = [\boldsymbol{a}]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$



Parameter perturbation

- About lie algebra associated with rotation SO3

parameter

vectorspace : $\phi \in \mathbb{R}^3$

field : \mathbb{R}

Cross product : $\phi_1 \times \phi_2$

Lie algebra

vectorspace: $\mathfrak{so}(3) = \{\Phi = \phi^\wedge \in \mathbb{R}^{3 \times 3} \mid \phi \in \mathbb{R}^3, \}$,

field: \mathbb{R} ,

Lie bracket: $[\Phi_1, \Phi_2] = \Phi_1 \Phi_2 - \Phi_2 \Phi_1,$

Lie group

$SO(3) = \{\mathbf{C} \in \mathbb{R}^{3 \times 3} \mid \mathbf{C}\mathbf{C}^T = \mathbf{1}, \det \mathbf{C} = 1\}$

. \wedge . \vee

$\exp(\cdot)$ $\log(\cdot)$





A few words on group/algebra



- Both group and algebra are concepts from abstract algebra. They are both algebraic structures.
- A **algebraic structure** is a set with one or more finitary **operations** defined on it that satisfies a list of axioms.
- Examples:
 - Matrices : matrix group
 - 3D Rotation matrices : SO(3) Lie group $\mathbf{R}^T \mathbf{R} = \mathbf{I}$, $\det(\mathbf{R}) = 1$
 - 3x3 Skew-symmetric matrices : so(3) Lie algebra

$$\mathbf{A} = -\mathbf{A}^T \in \mathbb{R}^{3 \times 3} \quad \mathbf{A} = \begin{bmatrix} 0 & a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} = [\mathbf{a}]_\times$$



Rotation perturbations



- When the parameter is represented by a rotation matrix :

$$x \boxplus \Delta x \leftrightarrow \mathbf{R} \cdot \delta \mathbf{R}$$

- Matrix exponential : $A \in \mathbb{R}^{n \times n}$

$$\exp(\mathbf{A}) = \mathbf{I} + \mathbf{A} + \frac{1}{2!}\mathbf{A}^2 + \frac{1}{3!}\mathbf{A}^3 + \dots$$

- The incremental rotation is computed from the perturbation vector by :

$$\begin{aligned}\delta \mathbf{R} &= \exp([\Delta x]_{\times}) = \mathbf{I} + [\Delta x]_{\times} + [\Delta x]_{\times}[\Delta x]_{\times} + \dots \\ &\approx \mathbf{I} + [\Delta x]_{\times}\end{aligned}$$

$$x \boxplus \Delta x \approx \mathbf{R}(\mathbf{I} + [\Delta x]_{\times})$$



Rotation perturbations



- When the parameter is represented by a unit quaternion:

$$x \boxplus \Delta x \leftrightarrow q \otimes \delta q$$

- Definition of a unit quaternion:

$$q = \begin{bmatrix} \cos(\theta/2) \\ \mathbf{u} \sin(\theta/2) \end{bmatrix} \xrightarrow{\theta \rightarrow 0} q \approx \begin{bmatrix} 1 \\ u\theta/2 \end{bmatrix}$$

- Let the perturbation vector $\Delta x \triangleq u\theta$, we have

$$x \boxplus \Delta x \approx q\{x\} \otimes \begin{bmatrix} 1 \\ \Delta x/2 \end{bmatrix}$$



Jacobian matrix with respect to non-vector parameters



- How do we compute the Jacobian matrix if the parameter is not a vector?
- The change of the function value after a small perturbation is described by

$$\frac{\partial}{\partial \Delta x} h = \frac{h(x \oplus \Delta x) - h(x)}{\Delta x} \mid_{\Delta x \rightarrow 0}$$



Numerical differentiation

- The Jacobian matrix can be evaluated by numerical differentiation if the analytic approach is too complicated.
- At each time, the perturbation is only enabled in a single dimension.

$$\Delta \boldsymbol{x}^{(j)} = [0 \cdots \delta \cdots 0]^T \in \mathbb{R}^n$$

δ is a small value: $\max(|10^{-4}x_i|, 10^{-6})$

$$\mathbf{H}(:, j) \approx \frac{\mathbf{h}(\boldsymbol{x} \boxplus \Delta \boldsymbol{x}^{(j)}) - \mathbf{h}(\boldsymbol{x})}{\delta}$$



Summary



- Non-vector parameters, like rotations
- Perturbation operator $\boxplus : \mathcal{X} \times \mathbb{R}^n \rightarrow \mathcal{X}$,
- Rotation perturbation :

$$x \boxplus \Delta x \leftrightarrow \mathbf{R} \cdot \delta \mathbf{R}$$

$$x \boxplus \Delta x \approx \mathbf{R}(\mathbf{I} + [\Delta x]_{\times})$$

$$x \boxplus \Delta x \leftrightarrow \mathbf{q} \otimes \delta \mathbf{q}$$

$$x \boxplus \Delta x \approx \mathbf{q}\{x\} \otimes \begin{bmatrix} 1 \\ \Delta x/2 \end{bmatrix}$$

- Jacobian matrix with respect to non-vector parameters:

$$\frac{\partial}{\partial \Delta x} \mathbf{h} = \frac{\mathbf{h}(x \boxplus \Delta x) - \mathbf{h}(x)}{\Delta x} \mid_{\Delta x \rightarrow 0}$$

- Numerical differentiation