

# Non-linear Least-Square Problem

No .....  
Date .....

$$\min_x F(x) = \frac{1}{2} \|f(x)\|_2^2$$

$$f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$$

Objective: get  $x$  s.t.  $F(x)$  has min

if easy  $\rightarrow$  analytical form

$\downarrow$  through

$$\frac{dF}{dx} = 0$$

else hard.  $\frac{dF}{dx}$  not easily solved.

algo  $\leftarrow$

give initial value  $x_0$

while true

at  $i = k$

search  $\Delta x_k$

get  $\|f(x_k + \Delta x_k)\|_2^2$

if  $\Delta x_k < b$

return

else

continue

$$x_{k+1} = x_k + \Delta x_k$$

How to get this?

@  $x_k$ ,  $i = k$ , get  $\Delta x_k$

do Taylor Expansion:

Jacobian

Hessian

$$F(x_k + \Delta x_k) \approx F(x_k) + J(x_k)^T \Delta x_k + \frac{1}{2} \Delta x_k^T H(x_k) \Delta x_k$$

1st order method

$$F(x_k + \Delta x_k) \approx F(x_k) + J(x_k)^T \Delta x_k$$

$$\Delta x^* = -J(x_k)$$

length parameter

$$\Delta x = -J(x_k) \cdot \lambda$$

steepest descent method

2nd order method

$$F(x_k + \Delta x_k) \approx F(x_k) + J(x_k)^T \Delta x_k + \frac{1}{2} \Delta x_k^T H(x_k) \Delta x_k$$

$$\Delta x^* = \arg \min ( \underbrace{F(x) + J(x)^T \Delta x + \frac{1}{2} \Delta x^T H \Delta x}_{P(x)} )$$

$$\frac{dP(x)}{d\Delta x} = J + H\Delta x = 0$$

$$H\Delta x = -J$$

$$\Delta x = -H^{-1}J$$

$$x_{k+1} = x_k + \Delta x_k$$

1st

$$x_{k+1} = x_k - J(x_k) \cdot \lambda$$

2nd

$$x_{k+1} = x_k - H^{-1}J$$

# Gauss-Newton Method

No. ....

Date .....

$$f(x) = \frac{1}{2} \sum_{j=1}^m r_j(x)^2 = \frac{1}{2} \sum_{j=1}^m \|r(x)\|_2^2$$

$$\begin{cases} r_j(x) = \phi(x; t_j) - y_j \\ j = 1, 2, 3, \dots, m \\ r(x) = [r_1(x), r_2(x), \dots, r_m(x)]^T \end{cases}$$

$$\nabla f(x_1, x_2) = \begin{pmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{pmatrix}$$

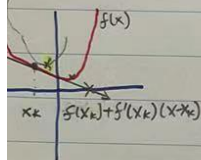
$$H(f(x_1, x_2)) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{pmatrix}$$

$x$  is the objective parameter (e.g. pose)  $x \in \mathbb{R}^n$   
 $r$  is the residual resuler calculated from  $x$  (e.g. reprojection  $u, v$ )  $m=1 \& 2$

recall Newton's method

$$X_{k+1} = X_k - [\nabla^2 f(X_k)]^{-1} \nabla f(X_k)$$

for non-linear



$$X_{k+1} = X_k - [H^{-1} J_r(X_k)^T r(X_k)]$$

to descent :

$$-\nabla f(x^*)^T [\nabla^2 f(x^*)]^{-1} \nabla f(x^*) < 0$$

thus Gauss-Newton Method  $f = \frac{1}{2} \sum_{j=1}^m r_j(x)^2$

$$\nabla f = J_r^T r \quad \frac{\partial f}{\partial x_i} = \sum_{j=1}^m \frac{\partial r_j}{\partial x_i} r_j$$

$$\nabla^2 f = J_r^T J_r + \sum_{i=1}^m r_i \nabla^2 r_i$$

$$= J_r^T J_r + Q$$

neglect this

$$\nabla^2 f \approx J_r^T J_r$$

$$\therefore X_{k+1} = X_k - \underbrace{[J_r(X_k)^T J_r(X_k)]^{-1} J_r(X_k)^T r(X_k)}_{\Delta X}$$

$$J(x) = \begin{bmatrix} \frac{\partial r}{\partial x_1} \\ \vdots \\ \frac{\partial r}{\partial x_n} \end{bmatrix}_{\substack{j=1 \dots m \\ i=1 \dots n}}$$

$$= \begin{bmatrix} \nabla r_1(x)^T \\ \nabla r_2(x)^T \\ \vdots \\ \nabla r_m(x)^T \end{bmatrix}$$

In sum : Gauss-Newton Approximate  $H \approx J^T J$

1<sup>st</sup> order

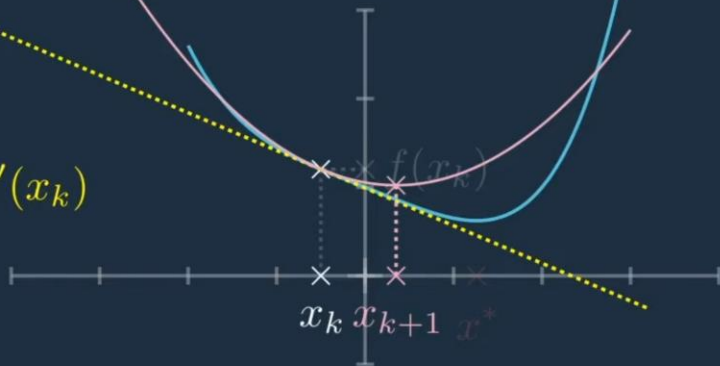
$$\begin{aligned} \nabla f(x) &= \nabla \frac{1}{2} \sum_{j=1}^m (r_j(x))^2 \\ &= \sum_{j=1}^m r_j(x) \nabla r_j(x) \\ &= J(x)^T r(x) \end{aligned}$$

2<sup>nd</sup> order

$$\begin{aligned} \nabla^2 f(x) &= \nabla \sum_{j=1}^m r_j(x) \nabla r_j(x) \\ &= \sum_{j=1}^m \nabla r_j(x)^T \nabla r_j(x) + \sum_{j=1}^m r_j(x) \nabla^2 r_j(x) \\ &= J(x)^T J(x) + \sum_{j=1}^m r_j(x) \nabla^2 r_j(x) \\ &= J(x)^T J(x) + Q(x) \end{aligned}$$

$$f(x) = \frac{1}{20}x^4 - \frac{2}{5}x + 1$$

$$x_{k+1} = x_k - \alpha \quad f'(x_k)$$



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

## 2 Gauss-Newton method

The Gauss-Newton method is a **simplification or approximation** of the Newton method that applies to functions  $f$  of the form (1). Differentiating (1) with respect to  $x_j$  gives

$$\frac{\partial f}{\partial x_j} = \sum_{i=1}^m \frac{\partial r_i}{\partial x_j} r_i,$$

and so the gradient of  $f$  is

$$\nabla f = J_r^T \mathbf{r},$$

where  $\mathbf{r} = [r_1, \dots, r_m]^T$  and  $J_r \in \mathbb{R}^{m,n}$  is the Jacobian of  $\mathbf{r}$ ,

$$J_r = \left[ \frac{\partial r_i}{\partial x_j} \right]_{i=1, \dots, m, j=1, \dots, n}.$$

Differentiating again, with respect to  $x_k$ , gives

$$\frac{\partial^2 f}{\partial x_j \partial x_k} = \sum_{i=1}^m \left( \frac{\partial r_i}{\partial x_j} \frac{\partial r_i}{\partial x_k} + r_i \frac{\partial^2 r_i}{\partial x_j \partial x_k} \right),$$

and so the Hessian of  $f$  is

$$\nabla^2 f = J_r^T J_r + Q,$$

where

$$Q = \sum_{i=1}^m r_i \nabla^2 r_i.$$

The Gauss-Newton method is the result of **neglecting the term  $Q$** , i.e., making the approximation

$$\nabla^2 f \approx J_r^T J_r. \quad (3)$$

Thus the Gauss-Newton iteration is

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (J_r(\mathbf{x}^{(k)})^T J_r(\mathbf{x}^{(k)}))^{-1} J_r(\mathbf{x}^{(k)})^T \mathbf{r}(\mathbf{x}^{(k)}).$$

In general the Gauss-Newton method will not converge quadratically but if the elements of  $Q$  are small as we approach a minimum, we can expect fast convergence. This will be the case if either the  $r_i$  or their second order partial derivatives

$$\frac{\partial^2 r_i}{\partial x_j \partial x_k}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta \mathbf{x}_k$$

$$\Delta \mathbf{x} = - \left[ J(\mathbf{x}_k)^T J(\mathbf{x}_k) \right]^{-1} J(\mathbf{x}_k)^T \mathbf{r}(\mathbf{x}_k)$$

$$\begin{matrix} [J(\mathbf{x}_k)^T J(\mathbf{x}_k)] & \Delta \mathbf{x} & = & J(\mathbf{x}_k)^T \cdot \mathbf{r}(\mathbf{x}_k) \\ A & \times & = & b \end{matrix}$$

$$\begin{aligned} f & \rightarrow \begin{matrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{matrix} \\ \nabla f &= \begin{bmatrix} \nabla r_1(\mathbf{x})^T \\ \nabla r_2(\mathbf{x})^T \\ \vdots \\ \nabla r_m(\mathbf{x})^T \end{bmatrix} = \begin{bmatrix} \frac{\partial r_1}{\partial x_i} \\ \vdots \\ \frac{\partial r_m}{\partial x_i} \end{bmatrix} \\ & \quad \begin{matrix} j=1, \dots, m \\ i=1, \dots, n \end{matrix} \\ &= J_r^T \mathbf{r}(\mathbf{x}) \end{aligned}$$