Nonlinear estimation via least-squares approximation

- Motivation (slide of previous lecture)
- Introduction:
  - convex case: $\min_{x \in \mathbb{R}^n} \frac{1}{2} \| Ax - b \|^2$
    - Direct method
      - * Cholesky and QR decomposition for numerical stability
    - Indirect method:
      - * Gradient descent
  - nonconvex case (vector space): $\min_{x \in \mathbb{R}^n} \| r(x) \|^2$
    - Gauss-Newton method
    - Levenberg-Marquardt
- Nonconvex case (manifold): $\min_{x \in \mathcal{M}} \| r(x) \|^2$

Convex case

$$\min_{x \in \mathbb{R}^n} \frac{1}{2}\|Ax-b\|^2 \qquad (A, b \text{ are given})$$

- **Direct method:**

Observe that

$$\frac{1}{2}\|Ax-b\|^2 = \frac{1}{2}x^TA^TAx - x^TA^Tb + \|b\|^2$$

convex function ($\cup \to x^2$, 1-D case)

$\Downarrow$

minimized when gradient $= 0$

$\Uparrow$

gradient

$$\nabla\left\{\frac{1}{2}\|Ax-b\|^2\right\} = A^TAx_{OPT} - A^Tb = 0$$

$$\Rightarrow \quad A^TA\,x_{OPT} = A^Tb$$

Assuming A^TA is invertible:

$$x_{opt} = (A^TA)^{-1} A^T b.$$

CAUTION: Inverting $(A^TA)^{-1}$ is:

- computationally expensive
- possibly numerically unstable.

⇓

Instead, solve system of equations:

$$A^TA\, x_{opt} = A^T b$$

Efficient methods to this end are

- Cholesky
- QR

(Gaussian elimination is avoided, since it is

more computationally demanding, and numerically unstable)

Cholesky: The method is based on the observation that equations of the following form are solved easily:

$Ly = b$, where L, b are given

and, in particular:

$$L = \begin{bmatrix} l_{11} & 0 & 0 & 0 \cdots 0 \\ l_{21} & l_{22} & 0 & 0 \cdots 0 \\ l_{31} & l_{32} & l_{33} & 0 \cdots 0 \\ \vdots & \vdots & \vdots & \ddots \; 0 \\ l_{n1} & l_{n2} & l_{n3} & \cdots \; l_{nn} \end{bmatrix}$$: lower-triangular

Example: for $n=2$:

$$Ly = b \Rightarrow \begin{pmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

$\Rightarrow l_{11} y_1 = b_1 \Rightarrow y_1 = \frac{b_1}{l_{11}}$ (easy!)

$\Rightarrow$ given y_1, now y_2 is also easy!

So, given $\underbrace{A^T A}_{\triangleq \bar{A}} x = \underbrace{A^T b}_{\triangleq \bar{b}}$,

Cholesky method finds L such that $\bar{A} = L L^T$ → lower triang.

$\Rightarrow \bar{A} x = \bar{b}$

$\Rightarrow L(\underbrace{L^T x}_{\triangleq y}) = \bar{b}$

$\Rightarrow$ first solve $Ly = \bar{b}$,
then solve $L^T x = y$,
which is also easy for the same reason (L^T is upper triangular)

QR: This method has similar intuition: given

$$A^T A x = A^T b,$$

the QR method finds

- $Q \in \mathbb{R}^{m \times n}$ such that $Q^T Q = I_n$
- $R \in \mathbb{R}^{n \times n}$ is upper triangular

such that

$$A = QR.$$

$$\Rightarrow \quad A^T A x = A^T b$$

$$\Rightarrow \quad R^T \underbrace{Q^T Q}_{=I} R x = \underbrace{R^T Q^T b}_{\triangleq \tilde{b}}$$

$\Rightarrow$ $R^T R x = \bar{b}$ $\Rightarrow$ easy to find y!

(R^T: lower triangular; $Rx = y$)

Once y is know, remains to solve:

$$Rx = y$$

which is also easy (R is upper triang)

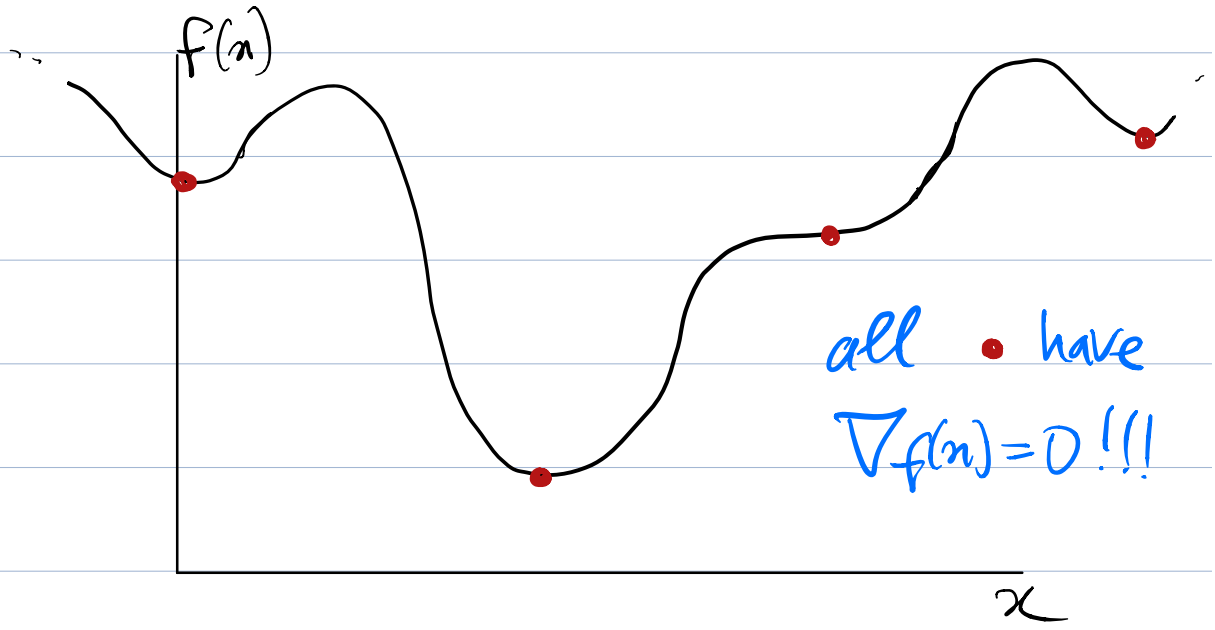
Cholesky vs QR

Cholesky:
- faster than QR
- valid only if A^TA is positive-DEFINITE

QR:
- slower than Cholesky
- valid even if A^TA is positive-SEMIdefinite

Indirect method: Gradient descent.

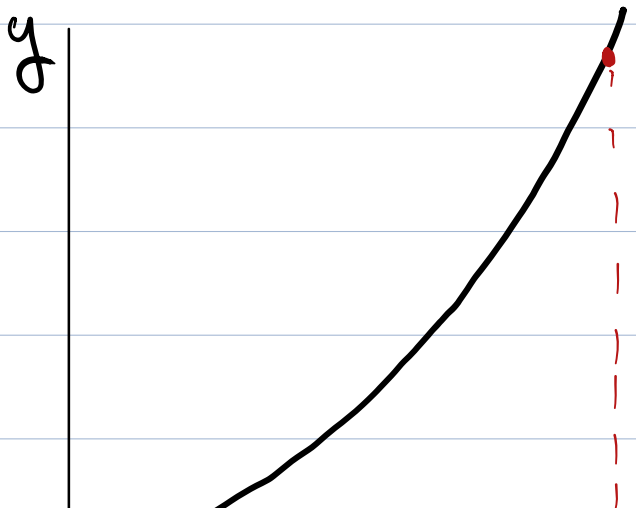
When our objective function is convex, as in the linear case above, we were able to find a closed form solution. But when the function will be nonconvex, that will not be easy.

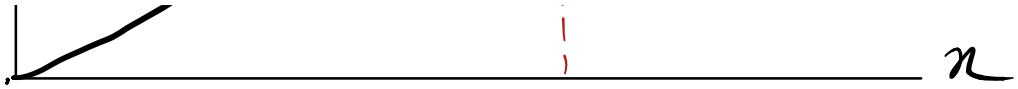


Therefore, we will need (as a potential approach) to develop iterative approximation methods.

Such a method is gradient descent, illustrated next, for simplicity, for a simple convex case.

Consider $r(x) = x^2$ (scalar case) and that want to $\min_x \|r(x)\|^2$





Question:

If I stand at x_0, can we decide at which x_1 should we stand next, such that $\|r(x_1)\|^2 \leq \|r(x_0)\|^2$? More generally, can we choose, sequentially x_i, $i = 1, 2, \ldots, k$ such that $\|r(x_i)\|^2 \to \min_x \|r(x)\|^2$ for $k \to \infty$?

Answer: Yes!

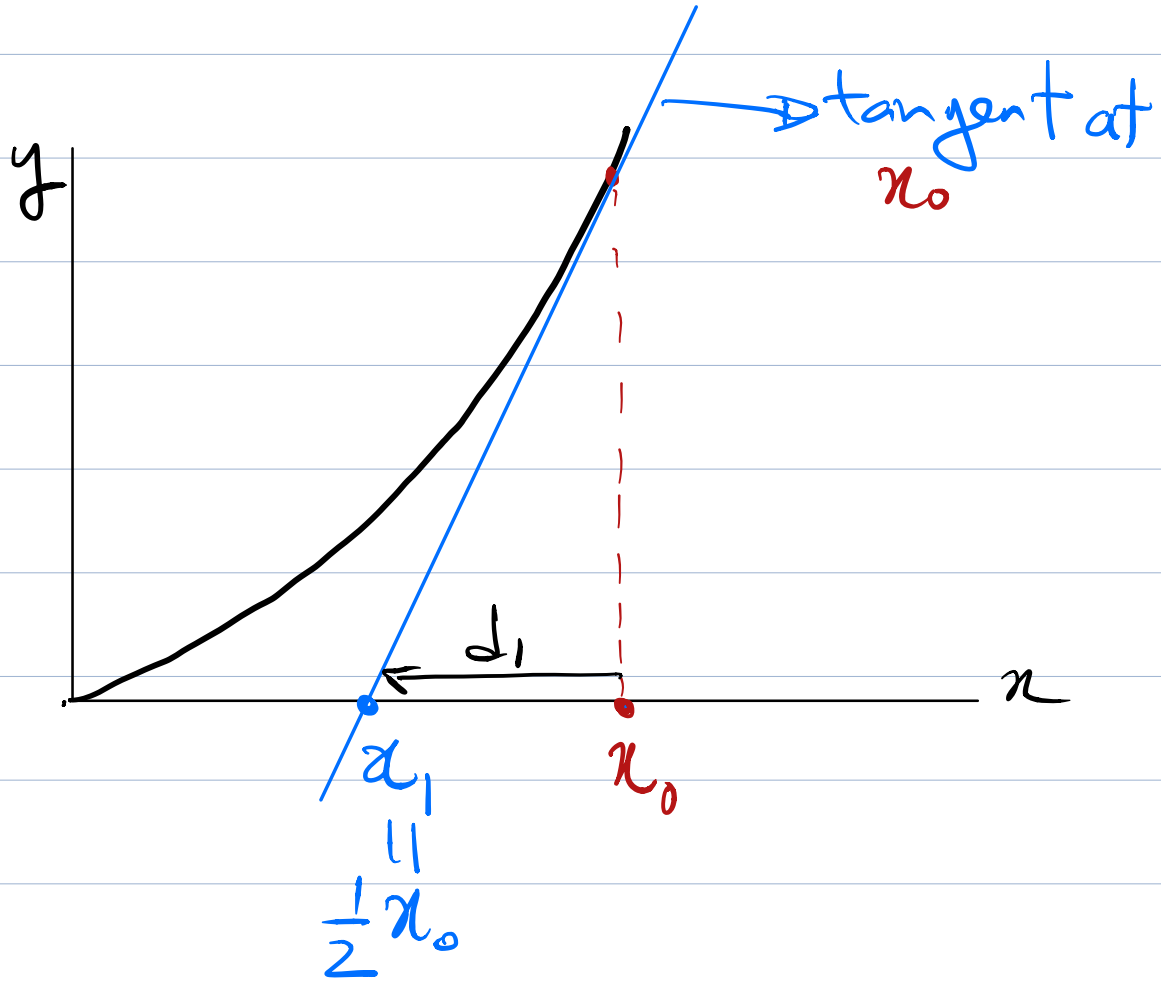
Instead of solving $\min_x \|r(x)\|^2$ approximate $r(x)$ with taylor expansion at $x = x_0$, i.e, $r(x_0 + d) \cong r(x_0) + r'(x_0)d$,

$= x_0^2 + 2x_0 d$

and minimize:

$$\min_{d \in \mathbb{R}^n} \|x_0^2 + 2x_0 d\|^2$$

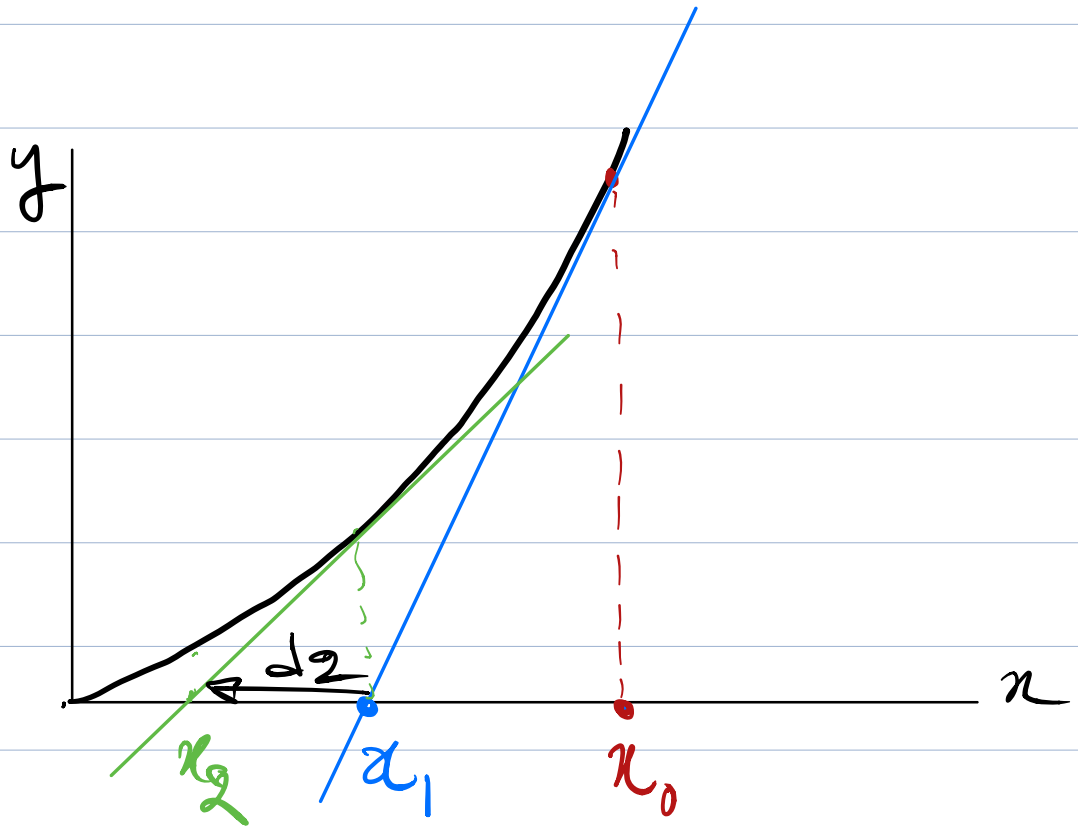
$$\Rightarrow d_1 = -\frac{1}{2} x_0$$



repeating from x_1

$$\Rightarrow d_2 = -\frac{1}{2} x_1 = -\frac{1}{4} x_0 \Rightarrow$$

$$x_2 = x_1 + d_2 = \frac{1}{4} x_0 = \frac{1}{2^2} x_0$$



$$\Rightarrow d_k = -\frac{1}{2^k} x_0, \; x_k = \frac{1}{2^k} x_0$$

and from the figure it's clear that $\|r(x_k)\|^2 \xrightarrow[k \to \infty]{} \min_x \|r(x)\|^2 = 0$

(since $x_k, d_k \to 0$).

All in all, gradient descent refers to the aforementioned iterative procedure. It becomes relevant next.

Non convex case (via iterative methods)

• Gauss-Newton method

$$f(x) = \frac{1}{2}\|r(x)\|^2,$$

where:

$\mathbf{r} : \mathbb{R}^n \to \mathbb{R}^m$ $(m \geq n)$

$\mathbf{r}$ is smooth, but not necessarily affine (i.e., $\mathbf{Ax} + \mathbf{b}$)

$\|\mathbf{r}(\mathbf{x})\|^2 = \sum_{i=1}^{m} r_i^2(\mathbf{x})$ where $r_i : \mathbb{R}^n \to \mathbb{R}$

first-order Taylor:

$$r_i(\mathbf{x}) \approx r_i(\mathbf{x}_0) + \nabla r_i(\mathbf{x}_0)^\top (\mathbf{x} - \mathbf{x}_0)$$

stack r_i's:

$$\mathbf{r}(\mathbf{x}) \approx \mathbf{r}(\mathbf{x}_0) + \underbrace{\mathbf{J}(\mathbf{x}_0)}_{\text{Jacobian}} (\mathbf{x} - \mathbf{x}_0)$$

$\Longrightarrow$

$$\mathbf{r}(\mathbf{x}_0 + \mathbf{d}) \approx \mathbf{r}(\mathbf{x}_0) + \mathbf{J}(\mathbf{x}_0)\mathbf{d}$$

where $\mathbf{J}(\mathbf{x}) \triangleq \dfrac{\partial \mathbf{r}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial r_1}{\partial x_1} & \frac{\partial r_1}{\partial x_2} & \cdots & \frac{\partial r_1}{\partial x_n} \\ \frac{\partial r_2}{\partial x_1} & \frac{\partial r_2}{\partial x_2} & \cdots & \frac{\partial r_2}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial r_m}{\partial x_1} & \frac{\partial r_m}{\partial x_2} & \cdots & \frac{\partial r_m}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{m \times n}$.

Gauss-Newton method

1. start from an initial guess $\mathbf{x}^0$

for $k = 0, 1, \cdots$ and until "convergence":

2. linearize the residual at the current guess $\mathbf{x}^k$:

$$\mathbf{r}(\mathbf{x}^k + \mathbf{d}) \approx \mathbf{r}(\mathbf{x}^k) + \mathbf{J}(\mathbf{x}^k)\mathbf{d}$$

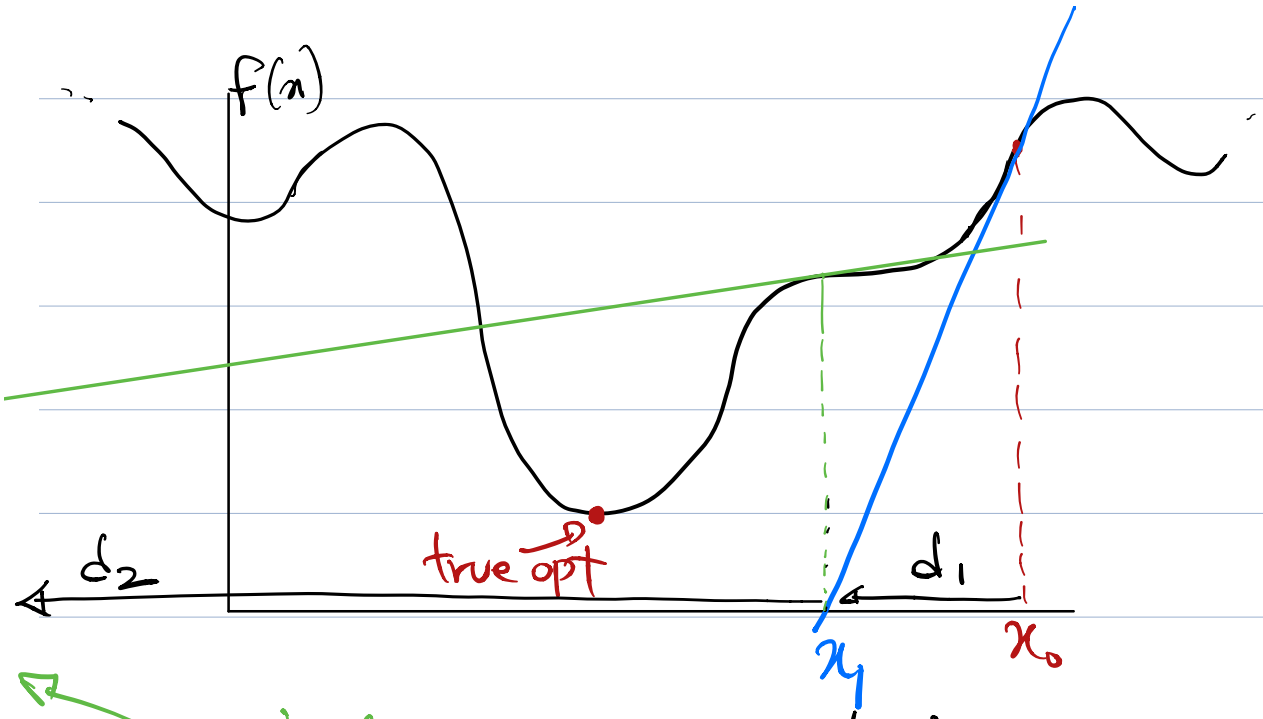
3. solve the resulting linear least squares to find the step $\mathbf{d}$:

$$\underset{\mathbf{d}}{\text{minimize}} \quad \|\mathbf{r}(\mathbf{x}^k) + \mathbf{J}(\mathbf{x}^k)\mathbf{d}\|^2$$

$$\Rightarrow \quad (\mathbf{J}_k^\top \mathbf{J}_k)\mathbf{d} = -\mathbf{J}_k^\top \mathbf{r}(\mathbf{x}^k)$$

4. $\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{d}$.

Issues with Gauss-Netwon method



x_2 is far away $\Rightarrow$ d_2 is very large!

idea: penalize $\|d\|^2$, so d doesn't explode.

$\Downarrow$

Levenberg - Marquardt method

Instead of (per Gauss-Netwon):

$$\min_d \; \|r(x^k) + J(x^k)d\|^2$$

do:

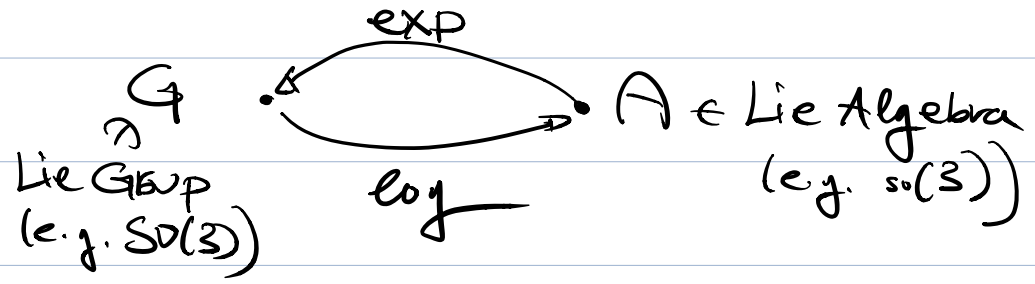
$$\min_d \; \|r(x^k) + J(x^k)d\|^2 + \lambda \|d\|^2$$

for a given $\lambda > 0$.

$$\Rightarrow \left[J^T(x^k)J(x^k) + \lambda I \right] d = -J^T(x^k) r(x^k)$$

Optimization over Manifolds

Review of Lie Groups / Algebra



where:

$$\exp(\mathbf{A}) \triangleq \mathbf{I} + \sum_{k=1}^{\infty} \frac{\mathbf{A}^k}{k!}$$

- $\exp(\mathbf{0}) = \mathbf{I}$
- in matrix Lie groups, $\exp$ maps Lie algebra (i.e., $\mathfrak{se}(3)$ and $\mathfrak{so}(3)$) to Lie group (i.e., $\mathrm{SO}(3)$ and $\mathrm{SE}(3)$)
- $\mathfrak{se}(3)$ and $\mathfrak{so}(3)$ are vector spaces $\rightarrow$ basis "vectors" (a.k.a. generators)

$$\widehat{\boldsymbol{\phi}} \in \mathfrak{so}(3) \Leftrightarrow \widehat{\boldsymbol{\phi}} = \phi_1 \mathbf{G}_1 + \phi_2 \mathbf{G}_2 + \phi_3 \mathbf{G}_3$$

where $\boldsymbol{\phi} \in \mathbb{R}^3$ and

$$\mathbf{G}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{G}_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad \mathbf{G}_3 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- $\widehat{\boldsymbol{\phi}} = [\boldsymbol{\phi}]_\times \Rightarrow \widehat{\boldsymbol{\phi}}\,\mathbf{a} = \boldsymbol{\phi} \times \mathbf{a}$

Similarly, for $\mathfrak{se}(3)$ consider $\boldsymbol{\phi} \in \mathbb{R}^3$ and $\boldsymbol{\rho} \in \mathbb{R}^3$ and the overloaded hat operator:

$$\widehat{\begin{bmatrix}\boldsymbol{\phi}\\ \boldsymbol{\rho}\end{bmatrix}} \in \mathfrak{se}(3) \Leftrightarrow \widehat{\begin{bmatrix}\boldsymbol{\phi}\\ \boldsymbol{\rho}\end{bmatrix}} = \phi_1\mathbf{G}_1 + \phi_2\mathbf{G}_2 + \phi_3\mathbf{G}_3 + \rho_1\mathbf{G}_4 + \rho_2\mathbf{G}_5 + \rho_3\mathbf{G}_6$$

where

$$\mathbf{G}_1 = \begin{bmatrix}0&0&0&0\\0&0&-1&0\\0&1&0&0\\0&0&0&0\end{bmatrix} \quad \mathbf{G}_2 = \begin{bmatrix}0&0&1&0\\0&0&0&0\\-1&0&0&0\\0&0&0&0\end{bmatrix} \quad \mathbf{G}_3 = \begin{bmatrix}0&-1&0&0\\1&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$$

$$\mathbf{G}_4 = \begin{bmatrix}0&0&0&1\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix} \quad \mathbf{G}_5 = \begin{bmatrix}0&0&0&0\\0&0&0&1\\0&0&0&0\\0&0&0&0\end{bmatrix} \quad \mathbf{G}_6 = \begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&1\\0&0&0&0\end{bmatrix}$$

Optimization over Manifolds

$$\min_{x \in \mathcal{M}} \frac{1}{2}\|r(x)\|^2$$

Example. Bundle adjustment:

$$\min_{\substack{(\boldsymbol{R}^{\mathrm{w}}_{\mathrm{c}_i}, \boldsymbol{t}^{\mathrm{w}}_{\mathrm{c}_i}), i=1,2,3 \\ \boldsymbol{p}^{\mathrm{w}}_k, k=1,\ldots,N}} \sum_{k=1}^{N}\sum_{i=1}^{3} \|\boldsymbol{x}_{k,i} - \pi(\boldsymbol{R}^{\mathrm{w}}_{\mathrm{c}_i}, \boldsymbol{t}^{\mathrm{w}}_{\mathrm{c}_i}, \boldsymbol{p}^{\mathrm{w}}_k)\|^2$$

where $p_k^w \in \mathbb{R}^3$ but $(R_{c_i}^w, t_{c_i}^w) \in SE(3)$.

Question Is Gauss-Newton (or any of the iterative methods we explored) valid anymore?

Answer NO!

$$x^{k+1} = x^k + d$$

since adding rotation doesn't make sense; multiplying does.

Problem to solve: How to define gradient descent steps "d" over manifolds?

Solution: Define "d_{11}" in the Lie Algebra and then map it to the group!

$$x^{k+1} = x^k \exp(\hat{d}),$$

where: d in Lie Algebra

- Gauss-Newton over $\mathbb{R}^n$

$$\mathbf{r}(\mathbf{x}^k + \mathbf{d}) \approx \mathbf{r}(\mathbf{x}^k) + \mathbf{J}_k \mathbf{d}$$

$$\mathbf{J}_k = \left.\frac{\partial \mathbf{r}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^k} = \left.\frac{\partial \mathbf{r}(\mathbf{x}^k + \mathbf{d})}{\partial \mathbf{d}}\right|_{\mathbf{d}=\mathbf{0}}$$

- Gauss-Newton over $\mathrm{SO}(3)$ — $\boxed{\mathbf{d} \in \mathbb{R}^3}$

$$\mathbf{r}(\mathbf{x}^k \exp(\widehat{\mathbf{d}})) \approx \mathbf{r}(\mathbf{x}^k) + \mathfrak{J}_k \mathbf{d}$$

$$\mathfrak{J}_k \triangleq \left.\frac{\partial \mathbf{r}(\mathbf{x}^k \exp(\widehat{\mathbf{d}}))}{\partial \mathbf{d}}\right|_{\mathbf{d}=\mathbf{0}}$$

- Gauss-Newton over $\mathrm{SE}(3)$ — $\boxed{\mathbf{d} \in \mathbb{R}^6}$

$$\mathbf{r}(\mathbf{x}^k \exp(\widehat{\mathbf{d}})) \approx \mathbf{r}(\mathbf{x}^k) + \mathfrak{J}_k \mathbf{d}$$

$$\mathfrak{J}_k \triangleq \left.\frac{\partial \mathbf{r}(\mathbf{x}^k \exp(\widehat{\mathbf{d}}))}{\partial \mathbf{d}}\right|_{\mathbf{d}=\mathbf{0}}$$

Lift-Solve-Retract

perturbation:

$$\mathbf{x}^{k+1} = \mathbf{x}^k \exp(\widehat{\mathbf{d}})$$

1. lift:

$$g : \mathbb{R}^{n_d} \to \mathbb{R}^m : \mathbf{d} \mapsto \mathbf{r}(\mathbf{x}^k \exp(\widehat{\mathbf{d}}))$$

e.g., $n_d = 3$ in $\mathrm{SO}(3)$ and $n_d = 6$ in $\mathrm{SE}(3)$

$$g(\mathbf{d}) \approx g(\mathbf{0}) + \left.\frac{\partial g(\mathbf{d})}{\partial \mathbf{d}}\right|_{\mathbf{d}=\mathbf{0}} \mathbf{d} \qquad \text{Taylor at } \mathbf{d} = \mathbf{0}$$

$$\mathbf{r}(\mathbf{x}^k \exp(\widehat{\mathbf{d}})) \approx \mathbf{r}(\mathbf{x}^k) + \mathfrak{J}_k \mathbf{d}$$

2. solve:

$$\underset{\mathbf{d}}{\text{minimize}} \qquad \frac{1}{2}\|\mathbf{r}(\mathbf{x}^k \exp(\widehat{\mathbf{d}}))\|^2 \approx \frac{1}{2}\|\mathbf{r}(\mathbf{x}^k) + \mathfrak{J}_k \mathbf{d}\|^2$$

linear least squares $\Rightarrow$ normal equations

$$\mathbf{d} = -(\mathfrak{J}_k^\top \mathfrak{J}_k)^{-1} \mathfrak{J}_k^\top \mathbf{r}(\mathbf{x}^k)$$

3. retract:

$$\mathbf{x}^{k+1} = \mathbf{x}^k \exp(\widehat{\mathbf{d}})$$

Tips to compute $\mathfrak{J}_k$: in $\mathbf{r}(\mathbf{x}^k \exp(\widehat{\mathbf{d}}))$

observe that:

- For $\|\mathbf{d}\| \approx 0$:

$$\exp(\widehat{\mathbf{d}}) \approx \mathbf{I} + \widehat{\mathbf{d}}$$

- express $\widehat{\mathbf{d}} = \sum_i d_i \mathbf{G}_i$ and take derivatives w.r.t. each d_i (i.e., columns of $\mathfrak{J}_k$)

Example

- Consider $\|\mathbf{r}(\mathbf{x}_1,\mathbf{x}_2)\|^2$ where $\mathbf{x}_1 \in \mathbb{R}^3$ and $\mathbf{x}_2 \in \mathrm{SO}(3)$

$$\|\mathbf{r}(\mathbf{x}_1^k + \mathbf{d}_1, \mathbf{x}_2^k \exp(\widehat{\mathbf{d}_2}))\|^2 \approx \|\mathbf{r}(\mathbf{x}_1^k,\mathbf{x}_2^k) + \mathbf{J}_{1,k}\mathbf{d}_1 + \mathfrak{J}_{2,k}\mathbf{d}_2\|^2$$

$$\mathbf{J}_{1,k} \triangleq \left.\frac{\partial \mathbf{r}(\mathbf{x})}{\partial \mathbf{x}_1}\right|_{\mathbf{x}=(\mathbf{x}_1^k,\mathbf{x}_2^k)} = \left.\frac{\partial \mathbf{r}(\mathbf{x}_1^k + \mathbf{d}_1, \mathbf{x}_2^k)}{\partial \mathbf{d}_1}\right|_{\mathbf{d}_1=\mathbf{0}}$$

$$\mathfrak{J}_{2,k} \triangleq \left.\frac{\partial \mathbf{r}(\mathbf{x}_1^k, \mathbf{x}_2^k \exp(\widehat{\mathbf{d}_2}))}{\partial \mathbf{d}_2}\right|_{\mathbf{d}_2=\mathbf{0}}$$

- solve the resulting linear least squares
- retract: $\mathbf{x}_1^{k+1} = \mathbf{x}_1^k + \mathbf{d}_1$ and $\mathbf{x}_2^{k+1} = \mathbf{x}_2^k \exp(\widehat{\mathbf{d}_2})$