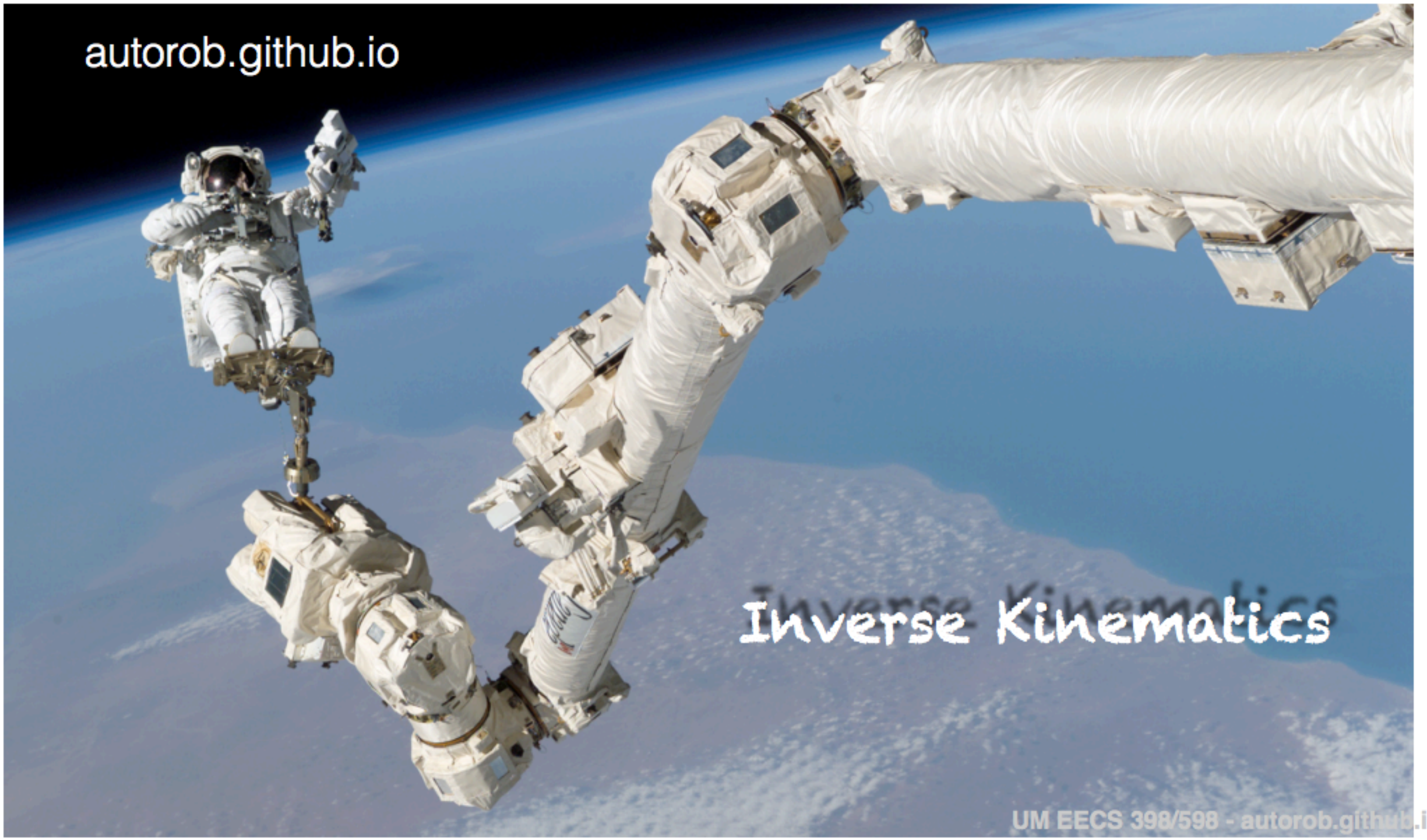
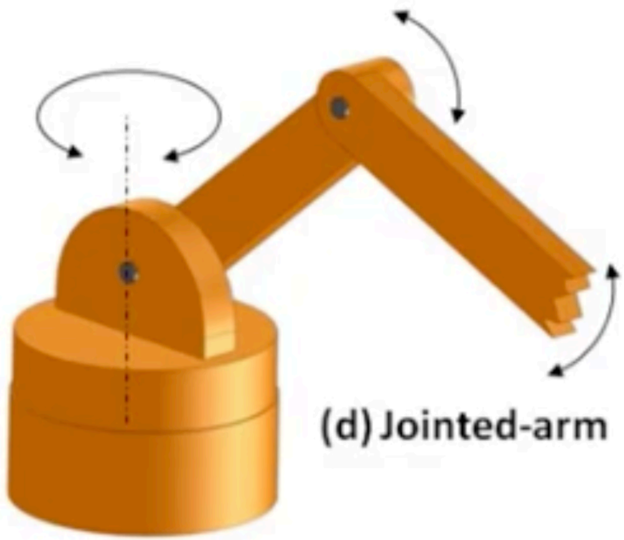
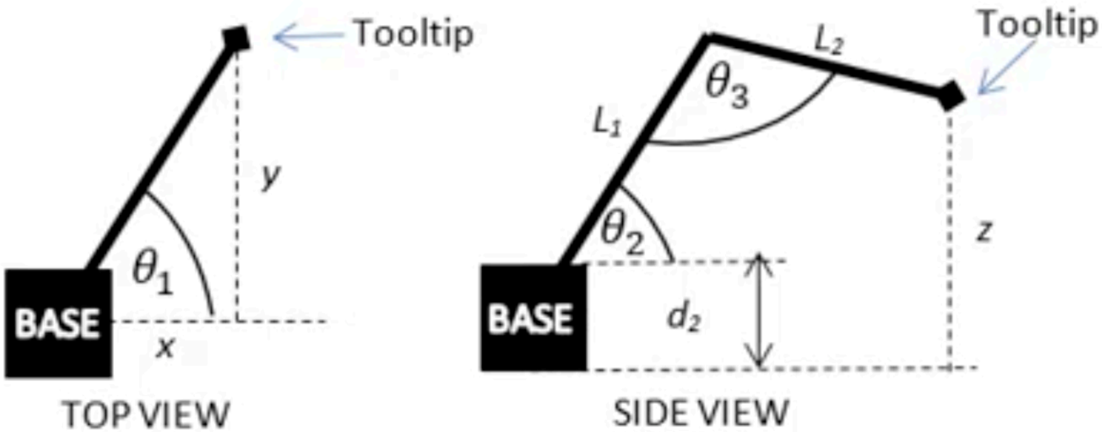
Nonlinear Equations

Nonlinear system of equations



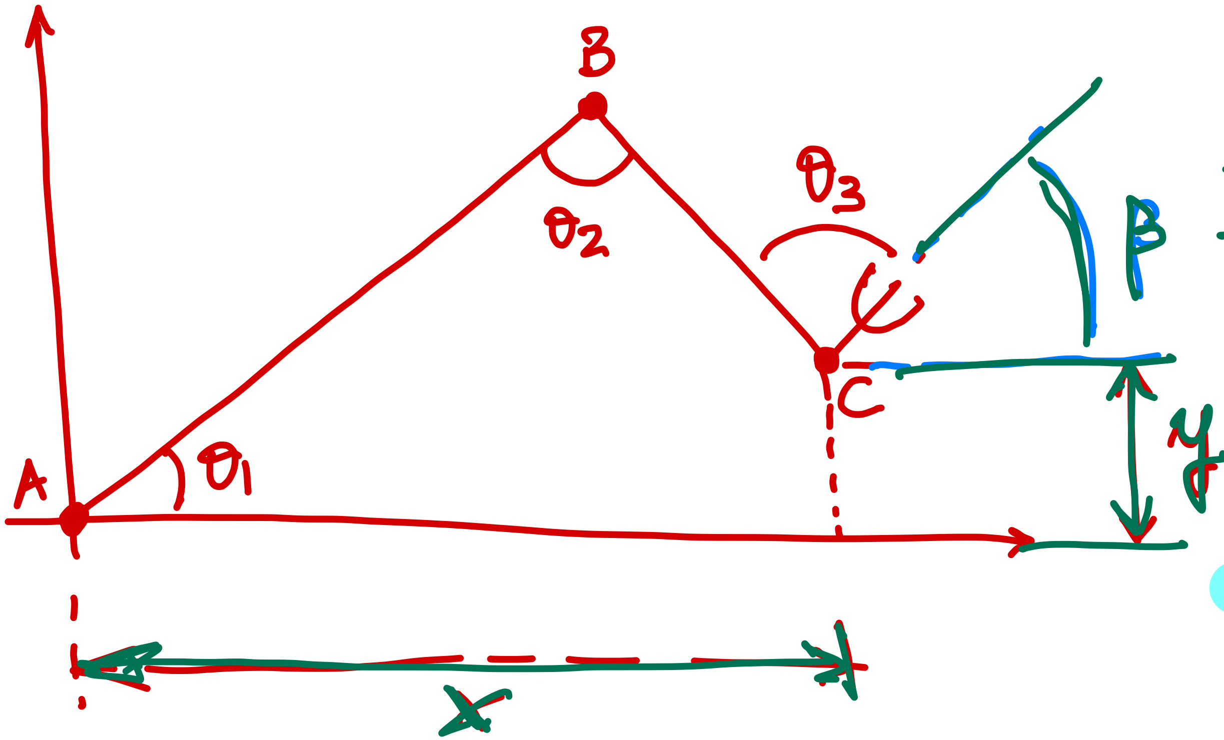
Robotic arms





https://www.youtube.com/watch?v=NRgNDlVtmz0 (Robotic arm 1)
https://www.youtube.com/watch?v=9DqRkLQ5Sv8 (Robotic arm 2)
https://www.youtube.com/watch?v=DZ_ocmY8xEI (Blender)

Inverse Kinematics



Given:

x, y, β

Find:

$\theta_1, \theta_2, \theta_3$

$$\boxed{\underset{\sim}{f}(\underset{\sim}{\theta}) = \underset{\sim}{0}}$$

$$f_1(\theta_1, \theta_2, \theta_3) = 0$$
$$f_2(\theta_1, \theta_2, \theta_3) = 0$$
$$f_3(\theta_1, \theta_2, \theta_3) = 0$$

3 eq.
+
3 unknowns

→ solve

↓

iterative

Nonlinear system of equations

Goal: Solve $\boldsymbol{f}(\boldsymbol{x}) = \mathbf{0}$ for $\boldsymbol{f}: \mathcal{R}^n \to \mathcal{R}^n$

$$\underset{\sim}{f}(\underset{\sim}{x}) = \underset{\sim}{0}$$

$$\underset{\sim}{f}(\underset{\sim}{x}) = \begin{bmatrix} f_1(x_1, x_2, x_3, \ldots, x_n) \\ f_2(x_1, x_2, x_3, \ldots, x_n) \\ \vdots \\ f_n(x_1, x_2, x_3, \ldots, x_n) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Suppose

$$\begin{cases} x_1^2 + 2x_1x_2 + x_2^3 = 4 \\ 2x_1 + 3x_2 = 5 \end{cases} \Longrightarrow \begin{aligned} f_1 &= -x_1^2 - 2x_1x_2 - x_2^3 + 4 = 0 \\ f_2 &= -2x_1 - 3x_2 + 5 = 0 \end{aligned}$$

Newton's method (ND)

Approximate the nonlinear function $\boldsymbol{f}(\boldsymbol{x})$ by a linear function using Taylor expansion:

$$\underbrace{\underbrace{\underset{\sim}{f}(\underset{\sim}{x}+\underset{\sim}{s})}_{\text{vector}}}_{\text{nonlinear}} \cong \underbrace{\underbrace{\underset{\sim}{f}(\underset{\sim}{x})}_{\text{vector}} + \underbrace{\underbrace{\underset{\sim}{J}(\underset{\sim}{x})}_{\text{matrix}}\ \underset{\sim}{s}}_{\text{vector}}}_{\text{linear approximation}}$$

$\underset{\sim}{s}$ ← vector

$$\underset{\sim}{f} = \begin{bmatrix} f_1(x_1, \dots, x_n) \\ f_2 \\ \vdots \\ f_n(x_1, \dots, x_n) \end{bmatrix} \qquad \underset{\sim}{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & & & \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{n \times n}$$

Jacobian

$$J_{ij} = \frac{\partial f_i}{\partial x_j}$$

$$\underbrace{\underset{\sim}{f}(\underset{\sim}{x}+\underset{\sim}{s})}_{=0} \simeq \underbrace{\underset{\sim}{f}(\underset{\sim}{x}) + \underset{\approx}{J}(\underset{\sim}{x})\,\underset{\sim}{s}}_{=0}$$

$$\underset{\sim}{f}(\underset{\sim}{x}) + \underset{\approx}{J}(\underset{\sim}{x})\,\underset{\sim}{s} = \underset{\sim}{0}$$

$$\boxed{\underset{\approx}{J}(\underset{\sim}{x})\,\underset{\sim}{s} = -\underset{\sim}{f}(\underset{\sim}{x})} \rightarrow$$

solve for $\underset{\sim}{s}$

Linear system of equation

$\underset{\sim}{x}_0$ = initial vector

$$\underset{\sim}{x}_{k+1} = \underset{\sim}{x}_k + \underset{\sim}{s}$$

Newton's method

$f(\underset{\sim}{x}) = \underset{\sim}{0}$

Algorithm:

$\underset{\sim}{x}_0$: initial guess

for $i = 1, 2, \cdots$

$n^2 \leftarrow$ evaluate $J(x_k) = J$

evaluate $f(x_k) = f$

$O(n^3) \leftarrow$ solve $\underset{\sim}{J}\underset{\sim}{s} = -\underset{\sim}{f} \rightarrow$ Find $\underset{\sim}{s}$

update $\underset{\sim}{x}_{k+1} = \underset{\sim}{x}_k + \underset{\sim}{s}$

iterative process sequence of linear system of equation

Convergence:

- Typically has quadratic convergence
- Drawback: Still only locally convergent

n^2

Cost:

- Main cost associated with computing the Jacobian matrix and solving the Newton step.

n^3

Example

$$\underset{\sim}{f} = \begin{bmatrix} 2y + x - 2 \\ x^2 + 4y^2 - 4 \end{bmatrix}$$

Consider solving the nonlinear system of equations

$$2 = 2y + x$$
$$4 = x^2 + 4y^2$$

What is the result of applying one iteration of Newton's method with the following initial guess?

$$\boldsymbol{x}_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\underset{\sim}{J} = \begin{bmatrix} 1 & 2 \\ 2x & 8y \end{bmatrix}$$

$$\underset{\sim}{x}_1 = \underline{x}_0 + \underset{\sim}{s} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$$

$J(x_0)$, $f(x_0)$

$$\underline{f}(\underline{x}_0) = \begin{bmatrix} -1 \\ -3 \end{bmatrix} \qquad \underline{J}(\underline{x}_0) = \begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix}$$

$$\underset{\sim}{x}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1.5 \\ -0.25 \end{bmatrix} = \begin{bmatrix} 2.5 \\ -0.25 \end{bmatrix}$$

$$\underline{J}\,\underline{s} = -\underline{f}$$

$$\begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

$s_1 + 2s_2 = 1$

$2s_1 = 3 \longrightarrow s_1 = 1.5$

$2s_2 = 1 - 1.5 = -0.5$

$s_2 = -0.25$

Newton's method

$\boldsymbol{x}_0 = initial\ guess$

$For\ k = 1,2,\ldots$

Evaluate $\mathbf{J} = \boldsymbol{J}(\boldsymbol{x}_k)$

Evaluate $\boldsymbol{f}(\boldsymbol{x}_k)$

la.solve

Factorization of Jacobian (for example $\mathbf{LU} = \boldsymbol{J}$)

Solve using factorized J (for example $\mathbf{LU}\ \boldsymbol{s}_k = -\boldsymbol{f}(\boldsymbol{x}_k)$)

Update $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$

Newton's method - summary

- Typically quadratic convergence (local convergence)

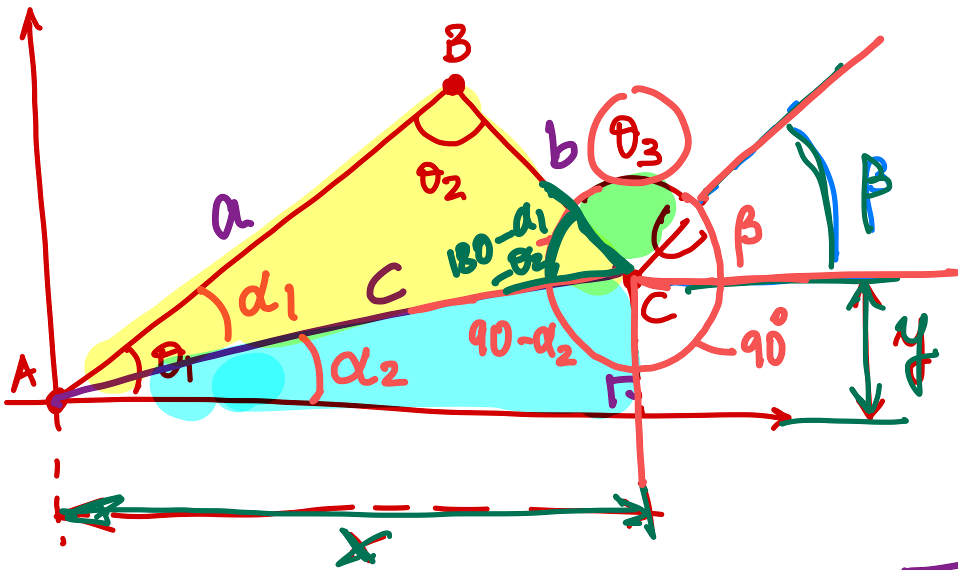
- Computing the Jacobian matrix requires the equivalent of n^2 function evaluations for a dense problem (where every function of $\boldsymbol{f}(\boldsymbol{x})$ depends on every component of $\boldsymbol{x}$).

- Computation of the Jacobian may be cheaper if the matrix is sparse.

- The cost of calculating the step $\boldsymbol{s}$ is $O(n^3)$ for a dense Jacobian matrix (Factorization + Solve)

- If the same Jacobian matrix $\boldsymbol{J}(\boldsymbol{x}_k)$ is reused for several consecutive iterations, the convergence rate will suffer accordingly (trade-off between cost per iteration and number of iterations needed for convergence)

Inverse Kinematics



$x, y, \beta \longrightarrow \theta_1, \theta_2, \theta_3$

$c = \sqrt{x^2+y^2}$ ✓

a, b given ✓

$\boxed{\theta_1 = \alpha_1 + \alpha_2} \rightarrow \alpha_1 = \theta_1 - \alpha_2$

$\alpha_2 = \tan^{-1}(y/x)$ ✓

$c^2 = a^2+b^2-2ab\cos\theta_2 \rightarrow \boxed{f_1 = c^2-a^2-b^2+2ab\cos\theta_2 = 0}$

$b^2 = a^2+c^2-2ac\cos\alpha_1 \rightarrow \boxed{f_2 = b^2-a^2-c^2+2ac\cos(\theta_1-\alpha_2)=0}$

$(\cancel{180}-\alpha_1-\theta_2)+\theta_3+\beta+\cancel{90}+(\cancel{90}-\alpha_2) = \cancel{360}$

$-\alpha_1-\theta_2+\theta_3+\beta-\alpha_2 = 0$

$-(\theta_1-\cancel{\alpha_2})-\theta_2+\theta_3+\beta\cancel{-\alpha_2}=0$

$\boxed{f_3 = -\theta_1-\theta_2+\theta_3+\beta=0}$