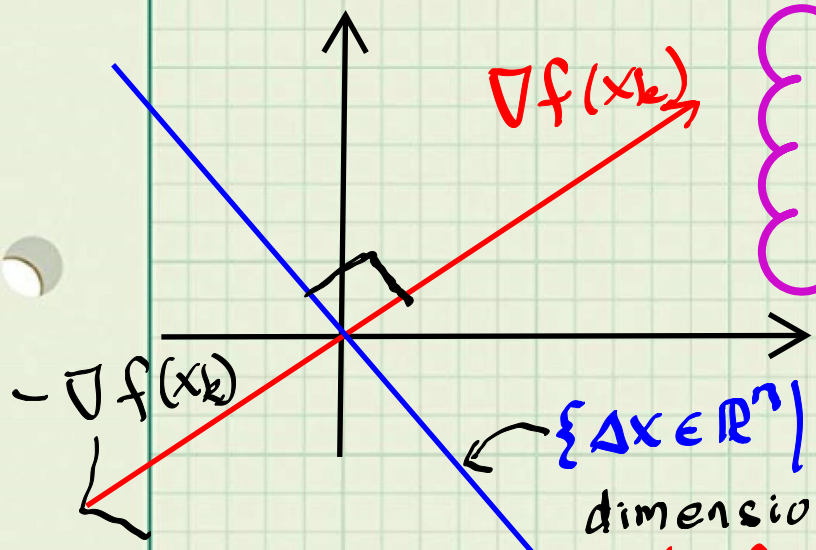
Summary Seek $x^* = \arg\min f(x)$. Let x_k be current value. How to choose Δx s.t. $f(x_k + \Delta x) < f(x_k)$? Use linearization

$$f(x_k + \Delta x) \cong f(x_k) + \nabla f(x_k) \cdot \Delta x$$

$$\Delta x = -s \nabla f(x_k) \Rightarrow f(x_{k+1}) \cong f(x_k) - s \|\nabla f(x_k)\|^2$$

$s > 0$ step size; $\|\nabla f(x_k)\|^2 > 0$



$$x_{k+1} = x_k - s \nabla f(x_k)$$

$0 < s < 1$

$\{\Delta x \in \mathbb{R}^n \mid \nabla f(x_k) \cdot \Delta x = 0\}$

dimension $= n-1$

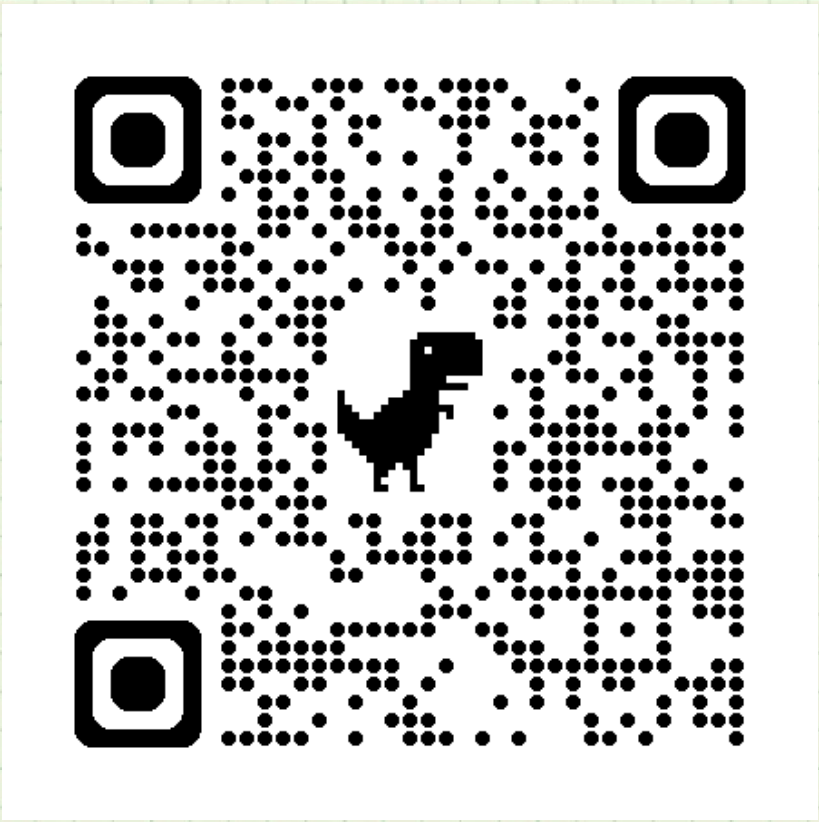
not fruitful to move in this direction

Next, seek $x^* = \arg\min f(x)$ s.t. $g(x) = 0$ How to choose Δx s.t. $g(x_k + \Delta x) = 0$ if $g(x_k) = 0$

$f(x_k + \Delta x) < f(x_k)$?

Quiz 3 is due Wed. HW5 Written + Julia HW5 due ~~Friday~~ Sunday 11:59 PM

Today:



Gradient Descent with One Equality Constraint

$f: \mathbb{R}^n \to \mathbb{R}$ cost function

$g: \mathbb{R}^n \to \mathbb{R}$ $g(x) = 0$ constraint

Assume x_k satisfies $g(x_k) = 0$.

Question: How to choose Δx such that

$f(x_k + \Delta x) < f(x_k)$ descending

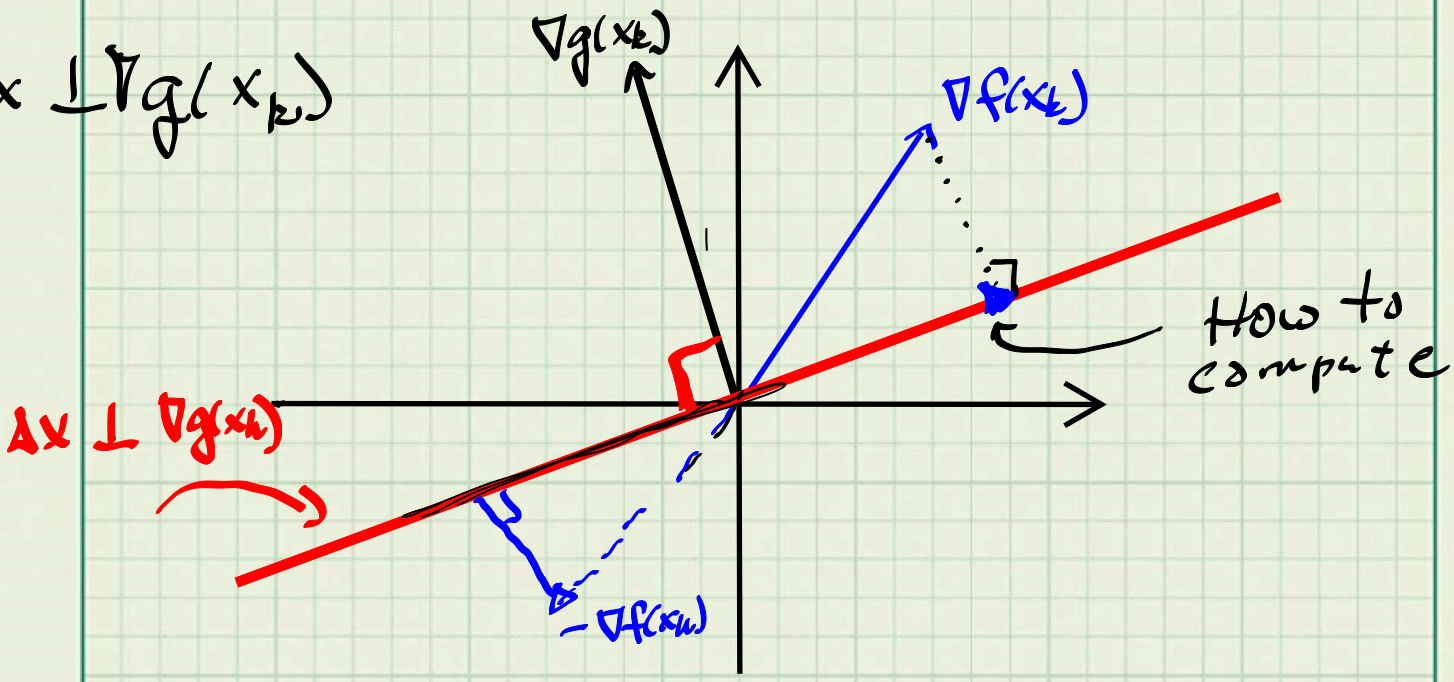
$g(x_k + \Delta x) = 0$ constraint satisfaction

Method: Exploit the linearization

$$f(x_k + \Delta x) \approx f(x_k) + \underbrace{\nabla f(x_k) \cdot \Delta x}_{<0} \quad \text{descend}$$

$$0 = g(x_k + \Delta x) \approx \cancel{g(x_k)}^{\,0} + \underbrace{\nabla g(x_k) \cdot \Delta x}_{=0} \quad \text{const. sat.}$$

$\Delta x \perp \nabla g(x_k)$



$$\Delta x = -\nabla f(x_k) + \alpha \nabla g(x_k)$$

and choose α such that $\Delta x \perp \nabla g(x_k)$

$$0 = (-\nabla f(x_k) + \alpha \nabla g(x_k)) \cdot \nabla g(x_k)$$

$$= -\nabla f(x_k) \cdot \nabla g(x_k) + \alpha \underbrace{\nabla g(x_k) \cdot \nabla g(x_k)}_{\neq 0 \text{ iff } \nabla g(x_k) \neq 0}$$

$$\therefore \alpha = \frac{\nabla f(x_k) \cdot \nabla g(x_k)}{\nabla g(x_k) \cdot \nabla g(x_k)}$$

Gradient Descent with one Equality Constraint

$$\underbrace{\Delta x}_{n\times 1} = -\underbrace{\nabla f(x_k)}_{n\times 1} + \underbrace{\left[\frac{\nabla f(x_k) \cdot \nabla g(x_k)}{\nabla g(x_k) \cdot \nabla g(x_k)}\right]}_{\alpha\text{ - scalar}} \cdot \underbrace{\nabla g(x_k)}_{n\times 1}$$

$$x_{k+1} = x_k + s\,\Delta x, \quad 0 < s < 1 \quad \text{step size}$$

Next Several Constraints

(mostly in Julia)

$$x^* = \arg\min f(x)$$
$$g_i(x) = 0 \quad 1 \le i \le m$$

$f: \mathbb{R}^n \to \mathbb{R}$

$g_i: \mathbb{R}^n \to \mathbb{R}$

Let x_k be such that $g_i(x_k) = 0$ $1 \le i \le m$.

Seek Δx such that < 0

$$f(x_k + \Delta x) \approx f(x_k) + \nabla f(x_k) \cdot \Delta x < f(x_k)$$

$$g(x_k + \Delta x) \approx \cancel{g(x_k)}^{\,0} + \underbrace{\nabla g_i(x_k) \cdot \Delta x}_{1 \le i \le m} = 0$$

$$\Delta x \perp \nabla g_i(x_k) \quad 1 \le i \le m$$

$$\begin{cases} \nabla f(x_k) \cdot \Delta x < 0 & \text{descent} \\ \nabla g_i(x_k) \cdot \Delta x = 0 & \text{constraint satisfaction} \end{cases}$$

Several ways to find Δx

$$\Delta x = -\nabla f(x_k) + \sum_{i=1}^{m} \alpha_i \nabla g_i(x_k)$$

Choose α_i such that

$$\Delta x \cdot \nabla g_i(x_k) = 0 \qquad 1 \le i \le m$$

Kicker: Can be solved by Gram Schmidt applied to

$\{\nabla g_1(x_k), \nabla g_2(x_k), \ldots, \nabla g_m(x_k), -\nabla f(x_k)\}$

Gram-Schmidt Process

Suppose that that the set of vectors $\{u_1, u_2, \ldots, u_m\}$ is linearly independent and you generate a new set of vectors by

$$\begin{aligned} v_1 &= u_1 \\ v_2 &= u_2 - \left(\frac{u_2 \bullet v_1}{v_1 \bullet v_1}\right) v_1 \\ v_3 &= u_3 - \left(\frac{u_3 \bullet v_1}{v_1 \bullet v_1}\right) v_1 - \left(\frac{u_3 \bullet v_2}{v_2 \bullet v_2}\right) v_2 \\ &\vdots \\ v_k &= u_k - \sum_{i=1}^{k-1} \left(\frac{u_k \bullet v_i}{v_i \bullet v_i}\right) v_i \quad \text{(General Step)} \end{aligned} \tag{9.16}$$

Then the set of vectors $\{v_1, v_2, \ldots, v_m\}$ is

- orthogonal, meaning, $i \neq j \implies v_i \bullet v_j = 0$
- span preserving, meaning that, for all $1 \leq k \leq m$,

$$\text{span}\{v_1, v_2, \ldots, v_k\} = \text{span}\{u_1, u_2, \ldots, u_k\}, \tag{9.17}$$

  and
- linearly independent.

Suggestion: If you do not see the pattern in the steps of the Gram-Schmidt Algorithm, please compare (9.16) to (9.11).

See Julia Demo

See textbook for the "Method of Lagrange Multipliers"

It also solves

$$x^* = \arg\min f(x)$$
$$g_i(x) = 0 \quad 1 \le i \le m$$

Many examples in the book.

- Less powerful than gradient descent
- More common in your engineering courses.

Learn what is a differential equation and how do they arise as models of mechanical systems.

Notation: $\frac{dx(t)}{dt} \longrightarrow \dot{x}(t)$

Third Notation reserved for the indep. variable t for time

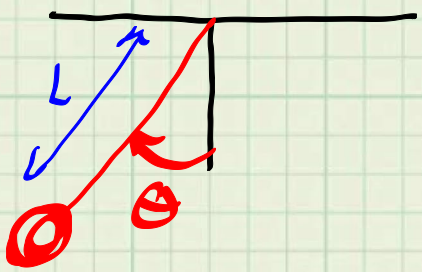
A differential equation is an equation that involves a derivative with respect to a single indep. variable.

Examples

a) $F = ma \iff F = m\frac{d^2p(t)}{dt^2} \iff F = m\ddot{x}(t)$

$$\ddot{x}(t) = \frac{1}{m}F(t)$$

b) Pendulum



$$\ddot{\theta}(t) = -\frac{g}{L}\sin(\theta)$$

$g = 9.8\ m/s^2$

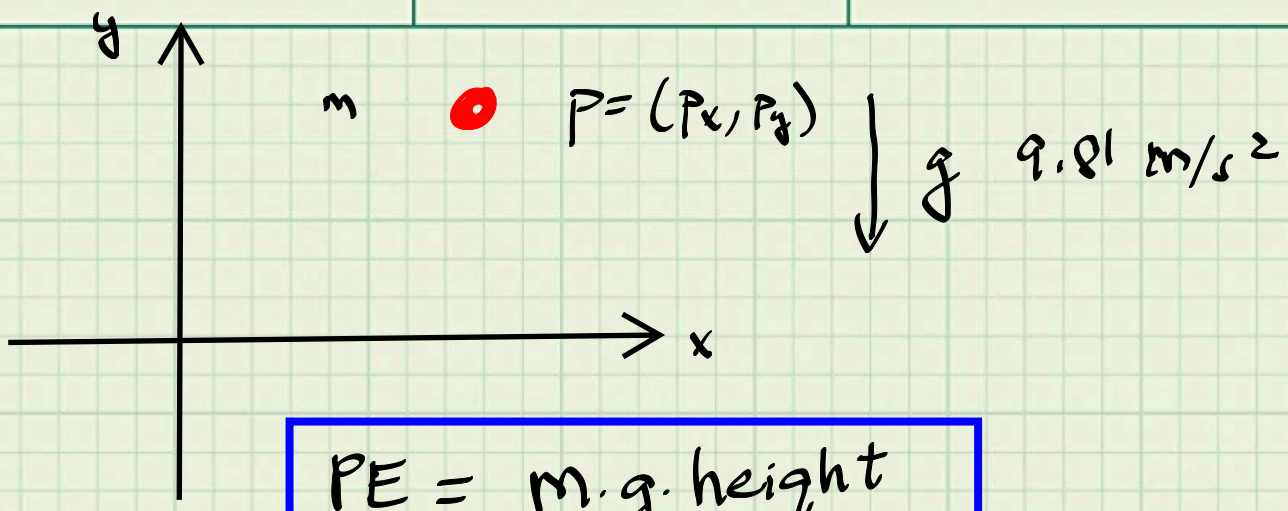
Our Goal: Is to be able to derive diff. eqn's for more complicated models.

Lagrange invented an energy-based approach to deriving diff. eqn's for a broad class of mechanical systems, such as robots.

Key Ideas

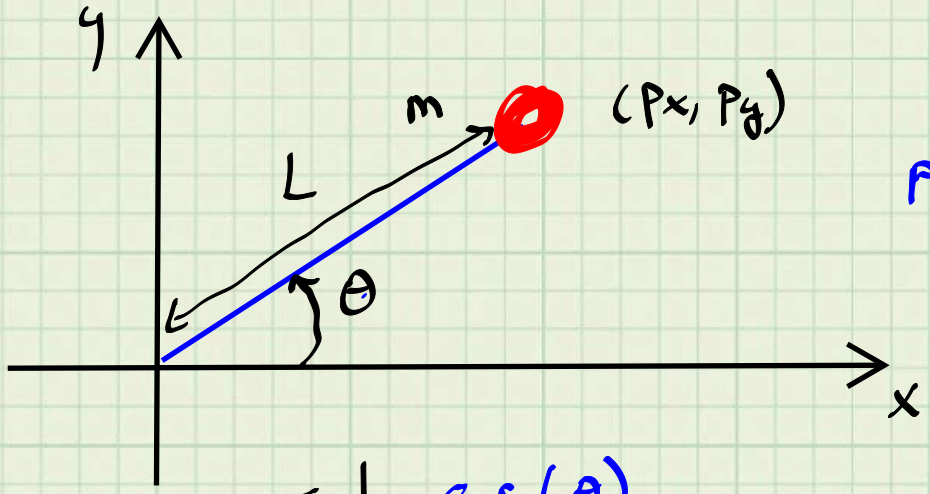
- PE = potential energy = energy of position
- KE = kinetic energy = energy of motion
- $\mathcal{L}$: KE - PE = the Lagrangian
- Secret Sauce : Lagrange's Eqn.

To Keep things Simple, we will cover only point masses.



$$PE = m \cdot g \cdot \text{height}$$

$$= m \cdot g \cdot P_y$$

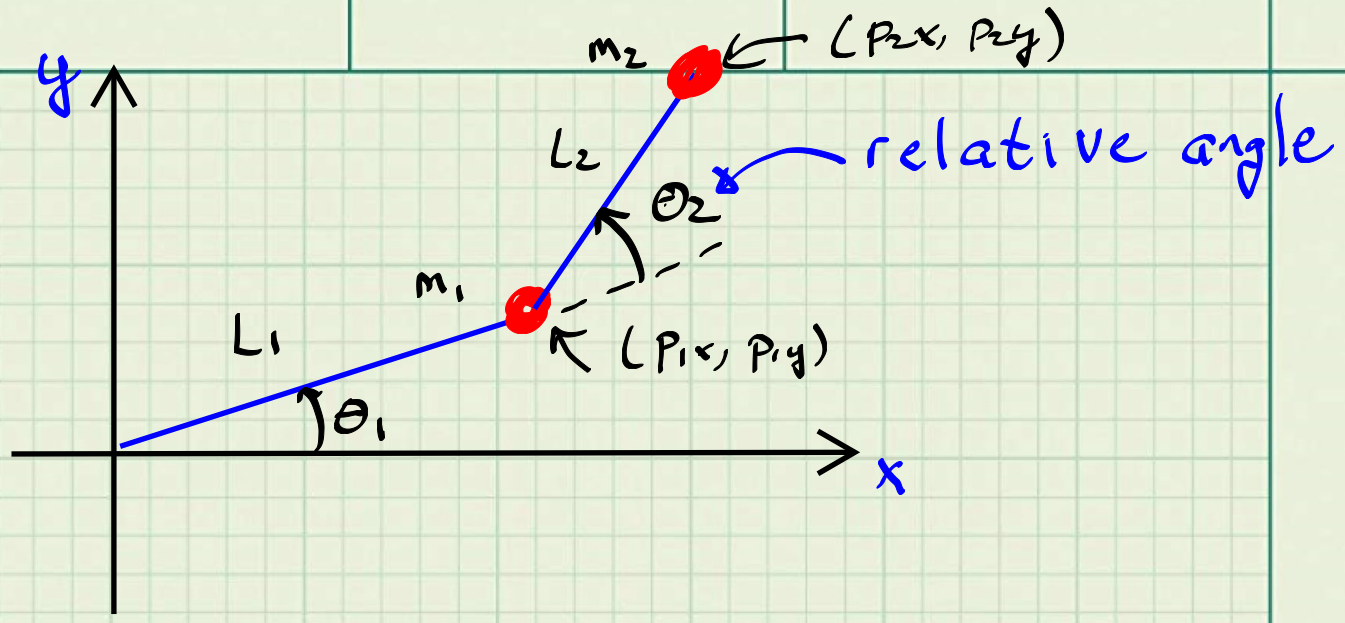


$PE = m \cdot g \cdot P_y$

$P_x = L \cdot \cos(\theta)$

$P_y = L \cdot \sin(\theta)$

$$PE = m \cdot g \cdot L \sin(\theta)$$



$$PE_{total} = m_1 g p_{1y} + m_2 g p_{2y}$$

$$p_{1y} = L_1 \sin(\theta_1)$$

$$p_{2y} = p_{1y} + L_2 \sin(\theta_1 + \theta_2)$$

$\theta_1 + \theta_2$: absolute angle of the second link

Bonus: Why Lagrange Multipliers Work (aka, proof)

Seek to maximize or minimize $f: \mathbb{R}^n \to \mathbb{R}$ such that $g: \mathbb{R}^n \to \mathbb{R}^m$ is equal to zero. Write

$$g(x) = \begin{bmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_m(x) \end{bmatrix}$$

Suppose x^* is a candidate stationary point (aka, extremum) and look at the linearizations

$$f(x^* + \Delta x) = f(x^*) + \nabla f(x^*) \cdot \Delta x$$

$$1 \le i \le m \quad g_i(x^* + \Delta x) = g(x^*) + \nabla g_i(x^*) \cdot \Delta x$$

For x^* to be an extremum (aka, stationary point) it must be true that

$$\star \begin{cases} \nabla f(x^*) \cdot \Delta x = 0 & \text{for all } \Delta x \text{ satisfying} \\ \nabla g_i(x^*) \cdot \Delta x = 0 & 1 \le i \le m. \end{cases}$$

Because, if not, choose ∇x s.t.

$\nabla f(x^*) \cdot \Delta x > 0$ while $\nabla g_i(x^*) \cdot \Delta x = 0$ $1 \le i \le m$. Then

$f(x^* + \Delta x) > f(x^*)$ and $f^*(x^* - \Delta x) < f(x^*)$

It is a (little known) FACT in Linear Algebra that the following two conditions

are equivalent when $\{\nabla g_1(x^*), \ldots, \nabla g_m(x^*)\}$ is linearly independent:

(a)

$$\star \begin{cases} \nabla f(x^*) \cdot \Delta x = 0 & \text{for all } \Delta x \text{ satisfying} \\ \nabla g_i(x^*) \cdot \Delta x = 0 & 1 \le i \le m. \end{cases}$$

(b)

$\nabla f(x^*)$ is a linear combination of $\{\nabla g_1(x^*), \nabla g_2(x^*), \ldots, \nabla g_m(x^*)\}$.

□

Therefore, by $\star$, there exist $\lambda_1^*, \lambda_2^*, \ldots, \lambda_m^*$ such that

$$0 = \nabla f(x^*) + \lambda_1^* \nabla g_1(x^*) + \lambda_2^* \nabla g_2(x^*) + \cdots + \lambda_m^* \nabla g_m(x^*)$$

that is

$$0_{n\times 1} = \nabla_x L(x^*, \lambda^*) = \begin{bmatrix} \frac{\partial L}{\partial x_1}(x^*, \lambda^*) \\ \vdots \\ \frac{\partial L}{\partial x_n}(x^*, \lambda^*) \end{bmatrix} \quad \text{for}$$

$$L(x,\lambda) := f(x) + \lambda_1 g_1(x) + \cdots + \lambda_m g_m(x)$$

which is Lagrange's Famous result, once you realize that

$$0 = \frac{\partial L}{\partial \lambda_1} = g_1(x)$$
$$\vdots$$
$$0 = \frac{\partial L}{\partial \lambda_m} = g_m(x)$$

} the constraints.

All of these famous results in optimization follow from clever analyses of linearized equations

(Optional Read)

Second Method to Solve For Δx

Using matrix-vector notation,

$$\Delta x = -\underbrace{\nabla f(x_k)}_{\substack{\nabla f \\ n\times 1}} + \underbrace{[\nabla g_1(x_k)\ \nabla g_2(x_k)\ \cdots\ \nabla g_m(x_k)]}_{\substack{\nabla G \\ n\times m}} \underbrace{\begin{bmatrix}\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_m\end{bmatrix}}_{\substack{\alpha \\ m\times 1}}$$

Then $\Delta x \perp \nabla g_i \quad 1 \le i \le m \iff \underbrace{(\nabla G)^T}_{m\times n} * \underbrace{\Delta x}_{n\times 1} = 0_{m\times 1}$

$$\iff -(\nabla G)^T * \nabla f + (\nabla G)^T * (\nabla G)\cdot \alpha = 0_{m\times 1}$$

If the columns of ∇G are linearly indep. (aka, rank = m), then

$$\alpha = (\nabla G)^T * \nabla G \backslash [(\nabla G)^T * \nabla f]$$

backslash in Julia

This gives the same Δx as Gram Schmidt applied to

$$\{\nabla g_1, \nabla g_2, \ldots, \nabla g_m, -\nabla f\}$$

Fact: $\Delta x = 0_{n\times 1} \iff \nabla f$ is a linear combination of $\{\nabla g_1, \nabla g_2, \ldots, \nabla g_m\}$. This happens at Convergence.

(Very Optional Read) Grad-level view

Suppose you don't believe that the best solution to

$$\nabla f \cdot \Delta x < 0$$

$$\nabla g_i \cdot \Delta x = 0 \quad 1 \le i \le m$$

is $\Delta x = -\nabla f + \sum_{i=1}^{m} \alpha_i \nabla g_i$. Can one use optimization to find a "best direction", Δx?

Yes! We note that

- $\nabla f \cdot \Delta x = J_f * \Delta x$
- $\nabla g_i \cdot \Delta x = 0 \quad 1 \le i \le m \iff J_G * \Delta x = 0$

where $G = \begin{bmatrix} g_1 \\ \vdots \\ g_m \end{bmatrix}$ and $J_G = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & & \\ \frac{\partial g_m}{\partial x_m} & \cdots & \frac{\partial g_m}{\partial x_n} \end{bmatrix}$

We seek Δx such that

$$\Delta x^* = \underset{\substack{J_G * \Delta x = 0 \\ \|\Delta x\|^2 = 1}}{\operatorname{argmin}} \; J_f * \Delta x$$

We impose $\|\Delta x\|^2 = 1$, because, otherwise, we can make $J_f * \Delta x = -\infty$!

Solution via Lagrange Multipliers

$$L = J_f * \Delta x + \lambda_1 (\Delta x^T * \Delta x - 1) + \lambda_2^T \cdot J_G * \Delta x$$

$$* \quad \frac{\partial L}{\partial x} = (J_f)^T + \lambda_1 \Delta x + (J_G)^T \cdot \lambda_2 = 0$$

Applying the constraint $J_G * \Delta x = 0$, yields

$$0 = J_G * (J_f)^T + \lambda_1 \underbrace{J_G * \Delta x}_{\to 0} + J_G * (J_G)^T \cdot \lambda_2$$

and thus, the Lagrange multiplier λ_2 satisfies

$$J_G * (J_G)^T \lambda_2 = -J_G * (J_f)^T$$

Recognizing that

$$(J_f)^T = \nabla f$$

$$(J_G)^T = [\nabla g_1 \ \nabla g_2 \ \cdots \ \nabla g_m] =: \nabla G$$

we have that λ_2 satisfies

$$(\nabla G)^T * \nabla G \ \lambda_2 = -(\nabla G)^T * \nabla f$$

the same equation we had for α in the previous derivation, except for a minus sign

From (*), we obtain

$$\lambda_1 \Delta x = -\nabla f - \nabla G * \lambda_2$$

$$\text{or} \quad \Delta x = \frac{-\nabla f - \nabla G \lambda_2}{\lambda_1}$$

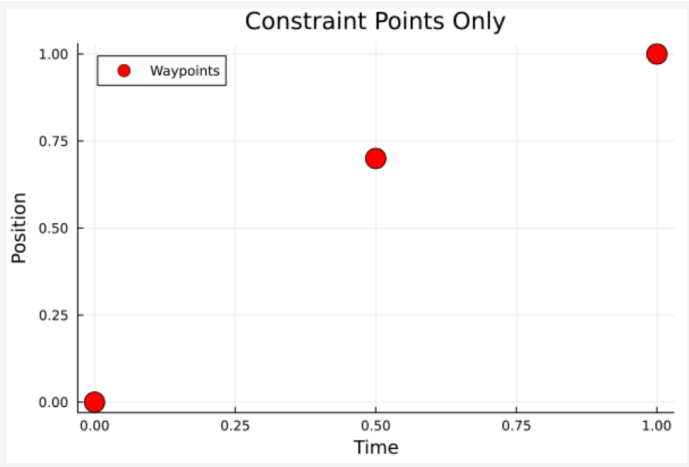
$$-\lambda_2 = (\nabla G^T * \nabla G) \backslash [(\nabla G)^T * \nabla f]$$

where λ_1 can be chosen to impose $\|\Delta x\|^2 = 1$.

But since we are using a step size $\alpha < s$, there is no need to perform the final normalization.

ROB 201 Path Planning Examples

#1



#2