## (i) Vector valued functions.

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

Eg: $f(x,y) = [x^2, 2y]^T$

$f(t) = [\cos(t), \sin(t), t]^T$ (helix in 3d space)

Why are these important?

- Because there are situations in robotics where we taken a input as a vector of numbers (eg joint angles) and output a vector too (3D coordinates)

## (ii) Jacobians & Hessians.

- First order partial derivatives of a vector valued functions.

- $J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$

Eg: $f(x,y) = [x^2+y, xy]^T$

$J = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix} = \begin{bmatrix} 2x & 1 \\ y & x \end{bmatrix}$

## (iii) Hessians

Second order partial derivatives of a function $f: \mathbb{R}^m \to \mathbb{R}$

$H = \begin{bmatrix} \frac{\partial^2 f_1}{\partial x_1^2} & \frac{\partial^2 f_1}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f_1}{\partial x_1 \partial x_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial^2 f_n}{\partial x_n \partial x_1} & \frac{\partial^2 f_n}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$

Is $\frac{\partial^2 f}{\partial x_1 \partial x_2} = \frac{\partial^2 f}{\partial x_n \partial x_1}$ ?

only if $f$ is $C^2$.

- Necessary and Sufficient conditions.

* $J(x^*) = 0$ if $x^*$ is a local minima (necessary condition)

* $\boxed{v^T H(x^*) v > 0}$ $\forall\, v \in \mathbb{R}^n$ is the local minima (sufficient condition).

$\hookrightarrow$ PD

- Importance of Jacobian & Hessian.
  (i) Gradient based optimization.
  (ii) Tests for convexity.
  (iii)

## (iv) Vector derivatives.

Given a $f: \mathbb{R}^n \to \mathbb{R}$, how to find $\nabla_x f(x)$? Eg: $f(x) = x^T x$, $x \in \mathbb{R}^n$.

$f(x) = x^T A x$, $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$

$f(x) = \|x\|_2^2$, $x \in \mathbb{R}^n$

$f'(x) = \frac{f(x+dx) - f(x)}{dx}$

$\Rightarrow \boxed{f'(x)\, dx = f(x+dx) - f(x)}$

$$\nabla f(x) = [f'(x)]^T$$

Eg: $f(x) = x^T A x$

$$f(x+dx) - f(x) = (x+dx)^T A (x+dx) - x^T A x$$
$$= dx^T A x + \underbrace{dx^T A dx}_{\to 0} + \cancel{x^T A x} + x^T A dx - \cancel{x^T A x}$$
$$= x^T A dx + x^T A^T dx$$
$$= (x^T A + x^T A^T)\, dx$$
$$= x^T (A + A^T)\, dx.$$

$\therefore f(x+dx) - f(x) = f'(x)\, dx$

$$\boxed{\nabla f(x) = (A + A^T)x}$$

Eg: $f(x) = x^T x$

$$f(x+dx) - f(x) = (x+dx)^T (x+dx)$$
$$= \cancel{x^T x} + x^T dx + dx^T x + \underbrace{dx^T dx}_{\to 0} - \cancel{x^T x}$$
$$= x^T dx + x^T dx$$
$$= 2x^T dx$$

$\therefore \nabla f(x) = (2x)^T{}^T = 2x$

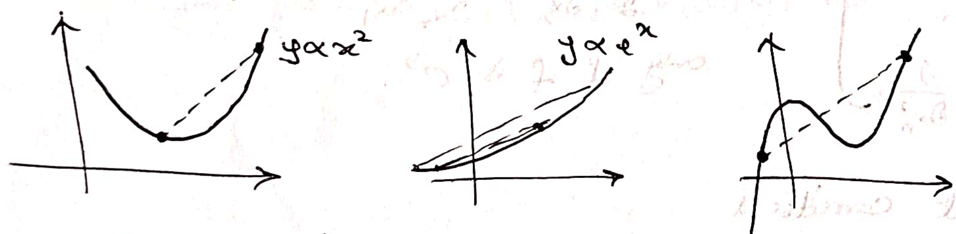
$\Rightarrow \boxed{\nabla f(x) = 2x}$

* - Convexity and convex functions.

- Convex sets: A set $S \subseteq \mathbb{R}^n$ is convex iff, $\forall x, y \in S$ $\exists \lambda \in [0,1]$ st. $\lambda x + (1-\lambda) y \in S$.

geometric intuition: all points in the line joining two point $(x,y)$ also belong to the set $S$.

- Convex functions: $f: \mathbb{R}^n \to \mathbb{R}$ is convex (i) its domain is convex.
(ii) $\forall x, y \in dom(f)$, $\exists \lambda \in [0,1]$ st. $f(\lambda x + (1-\lambda) y) \le \lambda f(x) + (1-\lambda) f(y)$




- The line joining the two points is always ~~the~~ above the function.

- Why do we care about convex functions?
For a convex function on a convex set, any local minima is a global minima. For strict convex function, there is atmost one global minimum.

- Tests for convexity:
(i) $f(\lambda x + (1-\lambda) y) \le \lambda f(x) + (1-\lambda) f(y)$
(ii) First order test; $\forall x, y \in dom(f)$, $f(y) \geq f(x) + \nabla f(x)^T (y - x)$.
(iii) $H$ is psd: $v^T H v \geq 0$ $\forall v$ and $\forall x \in dom(f)$.
↪ eigenvalues are non-negative.

Pg-3

Eg: $f(x,y) = x^2 + xy + y^2$

$\nabla^2 f(x,y) = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} = H$

$Tr(H) = 4 \geq 0$

$|H| = 3 \geq 0$

$\therefore$ $f(x,y)$ is convex.

How to check if eigenvalues are $\geq 0$:

$Tr(H) \geq 0$ and $|H| \geq 0$ then eigenvalues are $\geq 0$.

* Linear & Non-linear functions.

$f(x) = a^T x$

(i) Additivity: $f(x+y) = f(x) + f(y)$ $\forall x, y \in \mathbb{R}^n$

(ii) Homogeneity: $f(\alpha x) = \alpha f(x)$, $\forall \alpha \in \mathbb{R}^n$, $x \in \mathbb{R}^n$

Anything that is not linear is non-linear.

---

* Error Modelling.

Objective function most of time in our case would be a loss function.

somehow quantify the difference between model's predictions and the actual observations.

* Constrained optimization.

$\min f(x)$

s.t. $h_i(x) \leq 0$, $i = 1, 2 \ldots n$

$g_j(x) = 0$, $i = 1, 2 \ldots m$

feasible region, $F = \{x \mid x \in \mathbb{R}^n, h_i(x) \leq 0 \text{ and } g_j(x) = 0\}$

We look into the case of,

$\min f(x)$

s.t. $g_j(x) = 0$, $j = 1, 2 \ldots m$. (~~should be~~

$$\boxed{\mathcal{L}(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x)} \rightarrow \nabla \mathcal{L}(x, \lambda) = \nabla f(x) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x) = 0$$

(Lagrangian)

Lagrangian multipliers.

$x^*$ (candidate for optimum)

- $x^*$ can be minima, maxima or saddle points we dont know?
- When is $x^*$ guaranteed to be a minima point?
  - When $f(x)$ is convex and $g_i(x)$ are affine.
  - Lagrangian becomes sufficient in this case.
- otherwise Lagrangian is just necessary.

Eg: $\min x^2+y^2$
st. $x+y-1=0$.

$\mathcal{L}(x,\lambda) = x^2+y^2 + \lambda(x+y-1)$

$\therefore \nabla_x \mathcal{L}(x,y) = 2x+\lambda = 0$
$\nabla_y \mathcal{L}(x,y) = 2y+\lambda = 0$ } $\boxed{x=y}$

and, $x+y-1=0$

$\therefore 2x-1=0$
$\Rightarrow [x=y=\frac{1}{2}], \lambda=-1$

$\therefore \boxed{x^* = \frac{1}{2}, y^* = \frac{1}{2}}$

H/w: $\left[\begin{array}{l} \max f(x,y) = x+y \\ \text{st.: } x^2+y^2=4. \end{array}\right]$

* $\min_x x^TAx$ , $A=A^T$
st. $\|x\|_2^2 = 1$

$\mathcal{L}(x,\lambda) = x^TAx + \lambda(x^Tx-1)$
$\nabla\mathcal{L}(x,\lambda) = (A+A^T)x + 2\lambda x$
$= 2Ax + 2\lambda x = 0$
$\Rightarrow \boxed{Ax = -\lambda x}$

$x^Tx=1$ or $[Ax=\lambda x]$

$\therefore x^TAx = x^T\lambda x$
$\Rightarrow x^TAx = \lambda x^Tx = \lambda$

$\therefore \boxed{\lambda = x^TAx}$.
eigenvalues.

$\left[x^* = \text{eigenvectors}(A)\right.$ corresponding to smallest $\lambda$.$\left.\right]$

---

* · Nonlinear optimization (Gradient based).

$\min_x f(x)$. → NP hard, can never guarantee the global minima
be happy with a local minima.

$x_{k+1} = x_k + (\text{something})$ → decide on the direction
→ and the magnitude.

— take $-\nabla f(x)$ as the direction.

$\therefore x_{k+1} = x_k + [-\nabla f(x)]$

To control the magnitude, put a user-defined scalar.

$\therefore \boxed{x_{k+1} = x_k - \alpha \nabla f(x)}$ → Gradient Descent.

— All of non-linear optimization is just changing $\alpha$ and $\nabla f(x)$ to something better.

(i) Newton's Method:

$f(x) \approx f(x_k) + \nabla f(x_k)^T(x-x_k) + \frac{1}{2}(x-x_k)^T\nabla^2 f(x_k)(x-x_k)$

$\Rightarrow \nabla f(x) = \cancel{\nabla f(x_k)}^{0} + \nabla f(x_k)^T + \frac{1}{2}\times\left(2\nabla^2 f(x_k)^T x - 2\nabla^2 f(x_k)\, x_k\right)$

$= \nabla f(x_k) + \nabla^2 f(x_k)(x-x_k) = 0$

$\Rightarrow x - x_k = -[\nabla^2 f(x_k)]^{-1}\nabla f(x_k)$

$\Rightarrow \boxed{x = x_k - [\nabla^2 f(x_k)]^{-1}\nabla f(x_k)}$ → Newton's Method.

(iv) LM - method.


$f(x)^2$

$\cancel{f(x)\cdot\nabla f}$

$$\min_x f(x) = \frac{1}{2}\sum_{i=1}^{n} r_i(x)^2$$

$$= \frac{1}{2}\|r(x)\|_2^2$$

$r_i(x)$ are residuals.

$$f(x) = \frac{1}{2} r(x)^T r(x).$$

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

$$\Rightarrow \nabla f(x) = \frac{1}{2}\times 2 r(x)^T \nabla r(x)$$

$$= r(x)^T \nabla r(x)$$

$$\therefore \nabla f(x) = \sum_{i=1}^{m} r_i(x)\nabla r_i(x) = J(x)^T r(x)$$

$$\Rightarrow \nabla^2 f(x) = \cancel{H(x)^T r(x)} + J(x)^T J(x)$$

$$\sum_{i=1}^{n} r_i(x)\nabla^2 r_i(x)$$

Now,

$$\nabla^2 f(x) \approx J(x)^T J(x).$$

Now,

In newton's method: $x_{k+1} = x_k - [\nabla^2 f(x)]^{-1}\nabla f(x)$.

$$\therefore x_{k+1} = x_k - [J(x)^T J(x)]^{-1} \nabla f(x) \rightarrow J(x_k)^T r(x_k)$$

issues is when $J^T J$ is singular, so the inverse doesn't exist.

- GD uses,

$$x_{k+1} = x_k - \alpha_k J(x_k)^T r(x_k)$$

Combining both of them,

$$\boxed{x_{k+1} = x_k - [J(x)^T J(x) + \mu_k I]^{-1} J(x_k)^T r(x_k)}$$

(i) if $\mu_k$ is small: algo becomes ~~Newton's~~ Gauss Newton.

(ii) if $\mu_k$ is large: gradient descent.