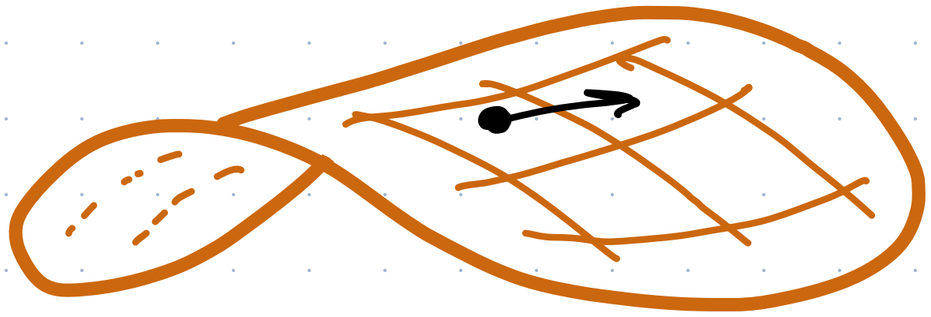
NTNU

Optimization on Riemannian Manifolds



HAJG JASA

Math meets industry, Trondheim, 14-15 August 2025

NTNU

Who cares anyways ?

NTNU

Who cares anyways ?

You should !

NTNU

Who cares anyways ?

You should !

if you work with

NTNU

Who cares anyways ?

You should !

if you work with

- shape analysis

NTNU

Who cares anyways ?

You should !

if you work with

- shape analysis
- signal / image processing

NTNU

Who cares anyways ?

You should !

if you work with

- shape analysis
- signal / image processing
- (sparse) PCA , matrix completion

NTNU

Who cares anyways ?

You should !

if you work with

- shape analysis
- signal / image processing
- (sparse) PCA, matrix completion
- tensor decompositon

NTNU

Who cares anyways ?

You should !

if you work with

- shape analysis
- signal / image processing
- (sparse) PCA, matrix completion
- tensor decompositon
- neural networks

NTNU

Who cares anyways ?

You should !

if you work with

- shape analysis
- signal / image processing
- (sparse) PCA, matrix completion
- tensor decompositon
- neural networks
- robotics and motion planning !

NTNU

Riemannian Optimization

→ algorithms to numerically solve

minimize $f(p)$, $p \in \mathcal{M}$

Riemannian Optimization

→ algorithms to numerically solve

minimize $f(p)$, $p \in \mathcal{M}$

with

- $\mathcal{M}$ a Riemannian manifold

Riemannian Optimization

→ algorithms to numerically solve

minimize $f(p)$, $p \in \mathcal{M}$

with

- $\mathcal{M}$ a Riemannian manifold
  ↳ possibly high-dimensional

Riemannian Optimization

→ algorithms to numerically solve

minimize $f(p)$, $p \in \mathcal{M}$

with

- $\mathcal{M}$ a Riemannian manifold
  - ↳ possibly high-dimensional
  - ↳ no "+" or "−" operations

Riemannian Optimization

→ algorithms to numerically solve

minimize $f(p)$, $p \in \mathcal{M}$

with

- $\mathcal{M}$ a Riemannian manifold
  - ↳ possibly high-dimensional
  - ↳ no "+" or "−" operations
- $f : \mathcal{M} \to \overline{\mathbb{R}}$ a function

NTNU

Riemannian Optimization

→ algorithms to numerically solve

minimize $f(p)$, $p \in \mathcal{M}$

with

- $\mathcal{M}$ a Riemannian manifold
  - ↳ possibly high-dimensional
  - ↳ no "+" or "−" operations
- $f : \mathcal{M} \to \overline{\mathbb{R}}$ a function
  - ↳ possibly nonsmooth (gradient?)

Riemannian Optimization

→ algorithms to numerically solve

minimize $f(p)$, $p \in \mathcal{M}$

with

- $\mathcal{M}$ a Riemannian manifold
  - ↳ possibly **high-dimensional**
  - ↳ **no "+" or "−"** operations
- $f : \mathcal{M} \to \overline{\mathbb{R}}$ a function
  - ↳ possibly **nonsmooth** and **nonconvex** (local minima)

NTNU

So, how do we solve 'em?

NTNU

So, how do we solve 'em?

Classicaly, for a smooth, convex function

NTNU

So, how do we solve 'em?

Classicaly, for a smooth, convex function

→ use gradient descent

$$x^{k+1} = x^k - \alpha^k \nabla f(x^k)$$

So, how do we solve 'em ?

Classicaly, for a smooth, convex function

→ use gradient descent

$$x^{k+1} = x^k - \alpha^k \nabla f(x^k)$$

- smoothness $\Rightarrow \exists \nabla f$

So, how do we solve 'em?

Classicaly, for a smooth, convex function

→ use gradient descent

$$x^{k+1} = x^k - \alpha^k \nabla f(x^k)$$

- smoothness $\Rightarrow \exists \nabla f$
- convexity $\Rightarrow$ optimality guarantees

NTNU

So, how do we solve 'em ?

Classicaly, for a smooth, convex function

→ use gradient descent

$$x^{k+1} = x^k - \alpha^k \nabla f(x^k)$$

- ~~smoothness~~ $\Rightarrow \exists \nabla f$?

- convexity $\Rightarrow$ optimality guarantees

NTNU

So, how do we solve 'em ?

Classicaly, for a smooth, convex function

→ use subgradient descent

$$x^{k+1} = x^k - \alpha^k g(x^k), \quad g(x^k) \in \partial f(x^k)$$

- ~~smoothness~~ $\Rightarrow \exists \nabla f$?

we can define a generalization: subdifferential ∂f

- convexity $\Rightarrow$ optimality guarantees

So, how do we solve 'em ?

Classicaly, for a smooth, convex function

→ use subgradient descent

$$x^{k+1} = x^k - \alpha^k g(x^k), \quad g(x^k) \in \partial f(x^k)$$

- ~~smoothness~~ $\Rightarrow \exists \nabla f$?

we can define a generalization : subdifferential ∂f

- ~~convexity~~ $\Rightarrow$ optimality guarantees ?

NTNU

So, how do we solve 'em ?

Classicaly, for a smooth, convex function

→ use subgradient descent

$$x^{k+1} = x^k - \alpha^k g(x^k), \quad g(x^k) \in \partial f(x^k)$$

- ~~smoothness~~ $\Rightarrow \exists \nabla f$?

  we can define a generalization : subdifferential ∂f

- ~~convexity~~ $\Rightarrow$ optimality guarantees ?

  ε - optimality : when $\|g(x^k)\| \leq \varepsilon$, for $g(x^k) \in \partial f(x^k)$

NTNU

So, how do we solve 'em ?

On manifolds, for a smooth, convex function

→ use subgradient descent

$$p^{k+1} = p^k - \alpha^k g(p^k), \quad g(p^k) \in \partial f(p^k)$$

- ~~smoothness~~ $\Rightarrow \exists \partial f$?

we can define a generalization: subdifferential ∂f

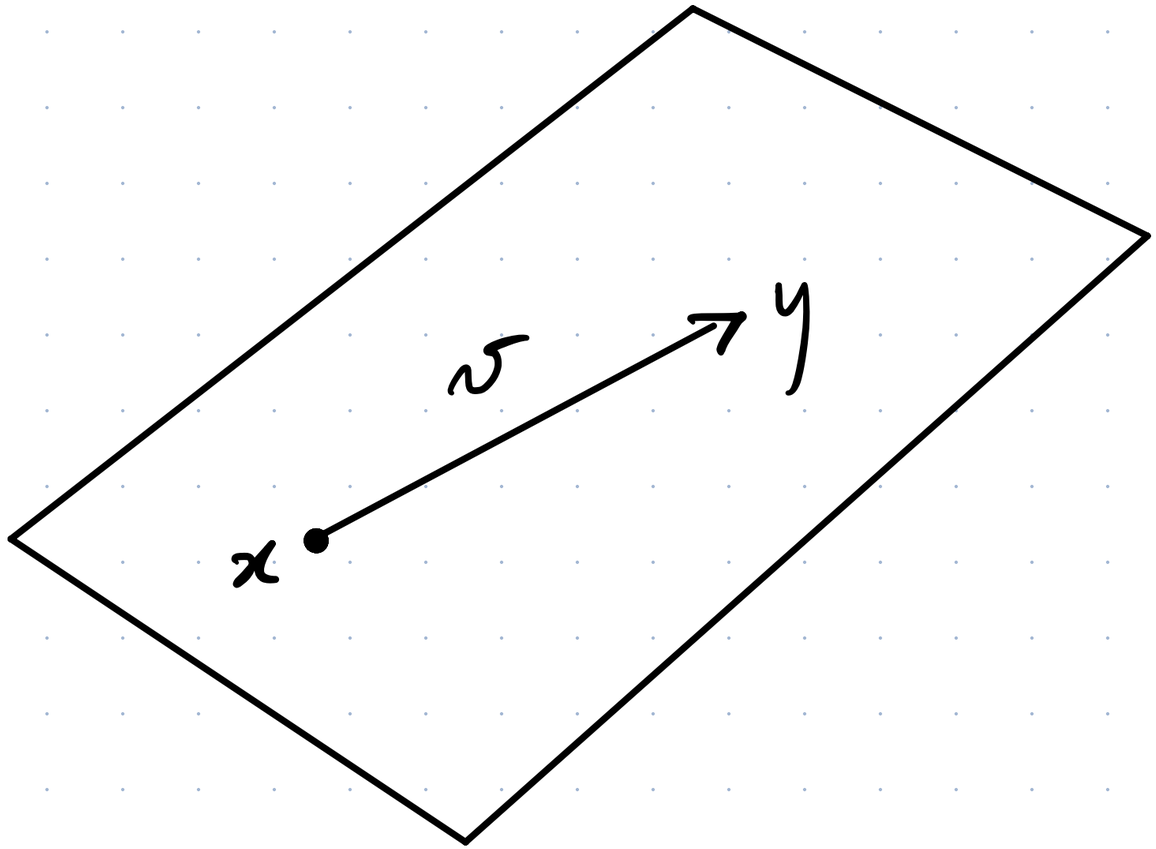
- ~~convexity~~ $\Rightarrow$ optimality guarantees ?

ε-optimality: when $\|g(p^k)\| \leq \varepsilon$, for $g(p^k) \in \partial f(p^k)$

NTNU

Making sense of "±"

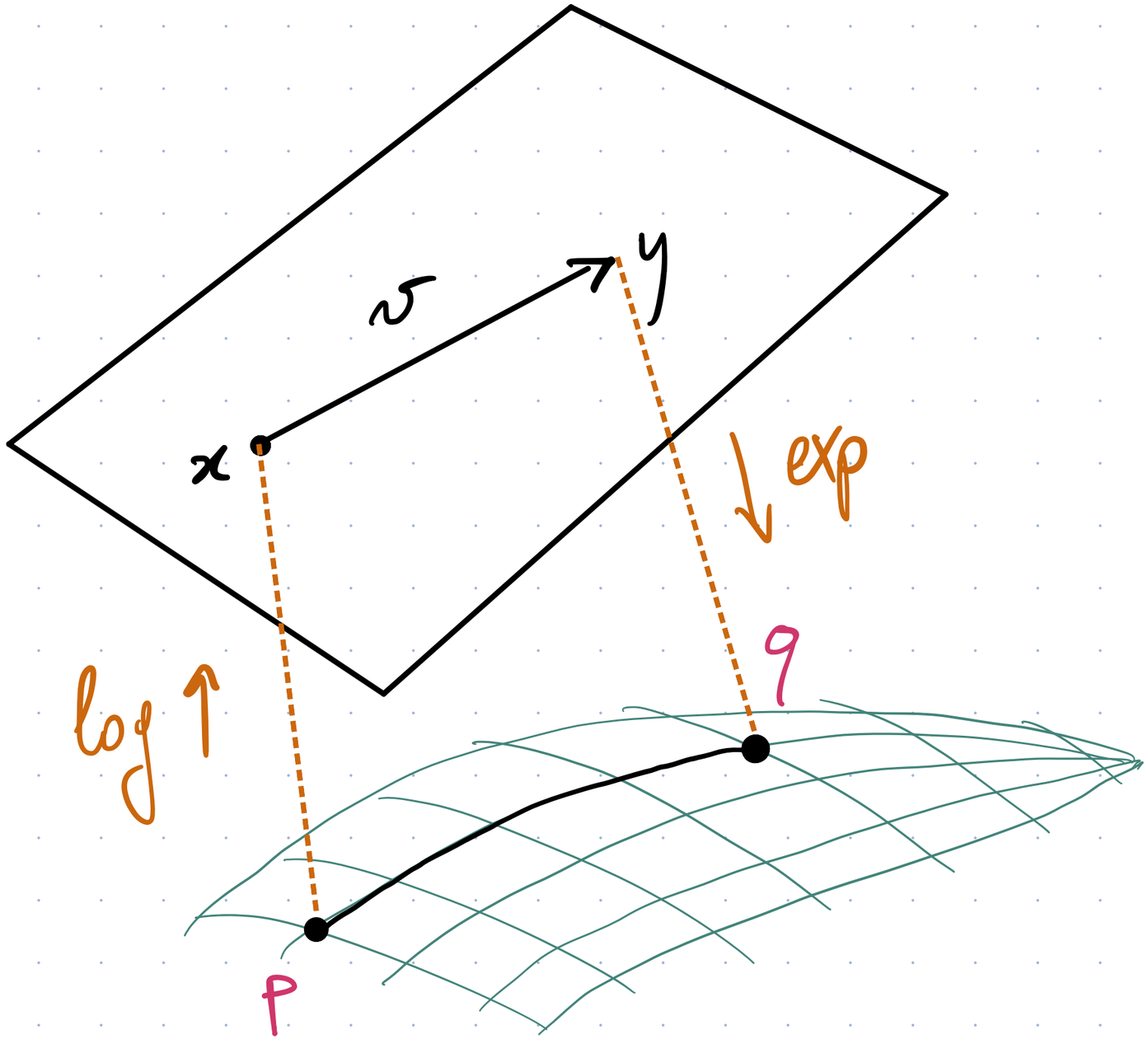
Making sense of "$\pm$"



$$v = y - x$$

$$y = x + v$$

Making sense of "$\pm$"



$$v = y - x$$

$$y = x + v$$

$$x = \log_p p$$

$$v = \log_p q$$

$$q = \exp_p v$$

NTNU

Software

Optimization on manifolds: manopt family

JULIA : Manopt.jl

PYTHON : pymanopt

MATLAB : Manopt

NTNU

Software

Optimization on manifolds: manopt family

JULIA : Manopt.jl

PYTHON : pymanopt

MATLAB : Manopt

in Julia, separate manifolds library in Manifolds.jl

NTNU

Summary

- motivation : real-world applications
- solutions : from Euclidean to Riemannian algorithms
- for the user : software

Thank you

for your

attention !