https://nypost.com/2019/02/19/youtube-is-helping-the-flat-earth-conspiracy-movement-grow/

https://www.brookeandjubalradio.com/featured/brooke-and-jubal/content/2018-03-09-second-date-tommy-and-hope-why-on-earth/

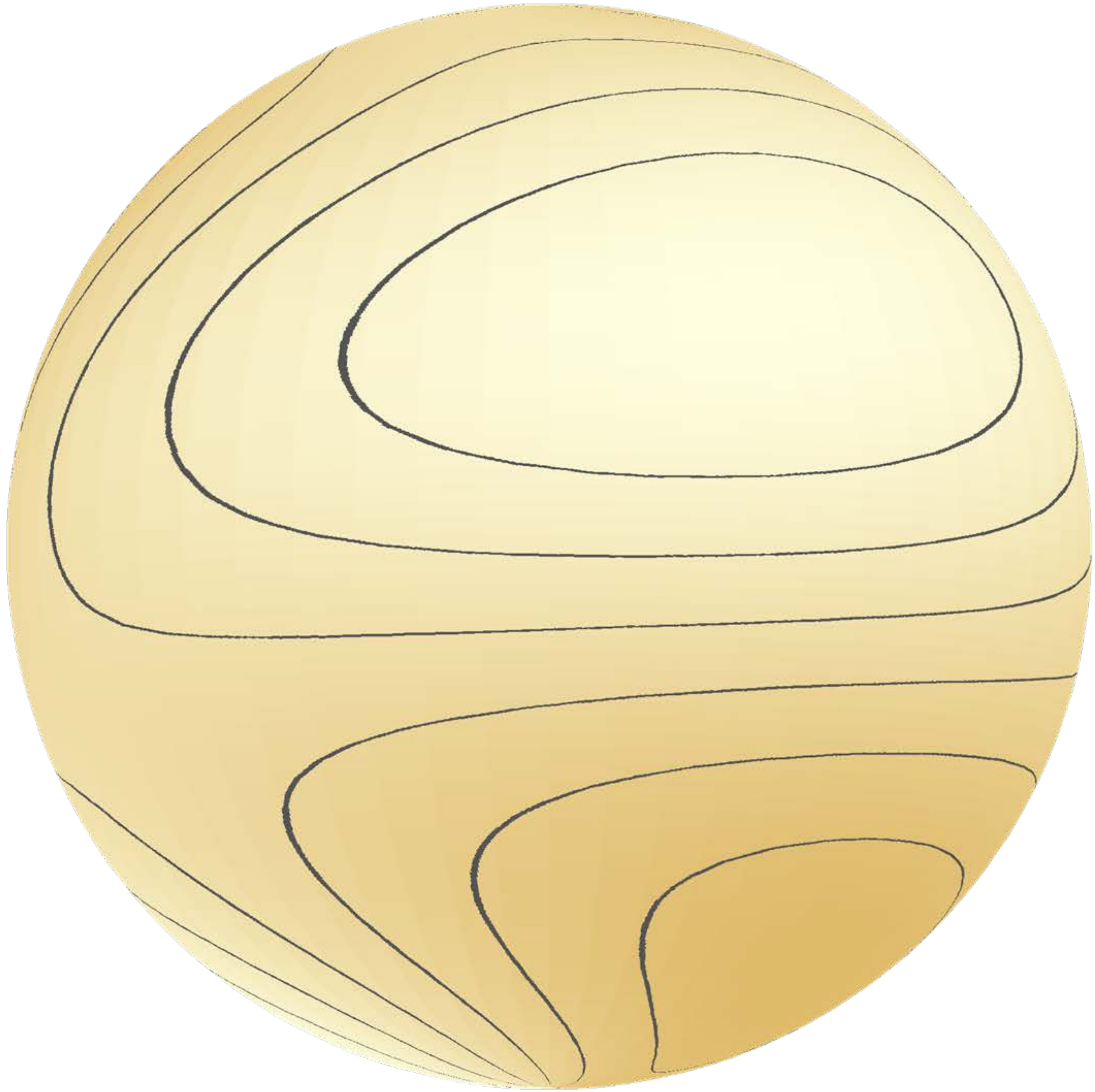
https://www.vice.com/en_us/article/mbyak8/apparently-some-people-believe-the-earth-is-shaped-like-a-donut-1

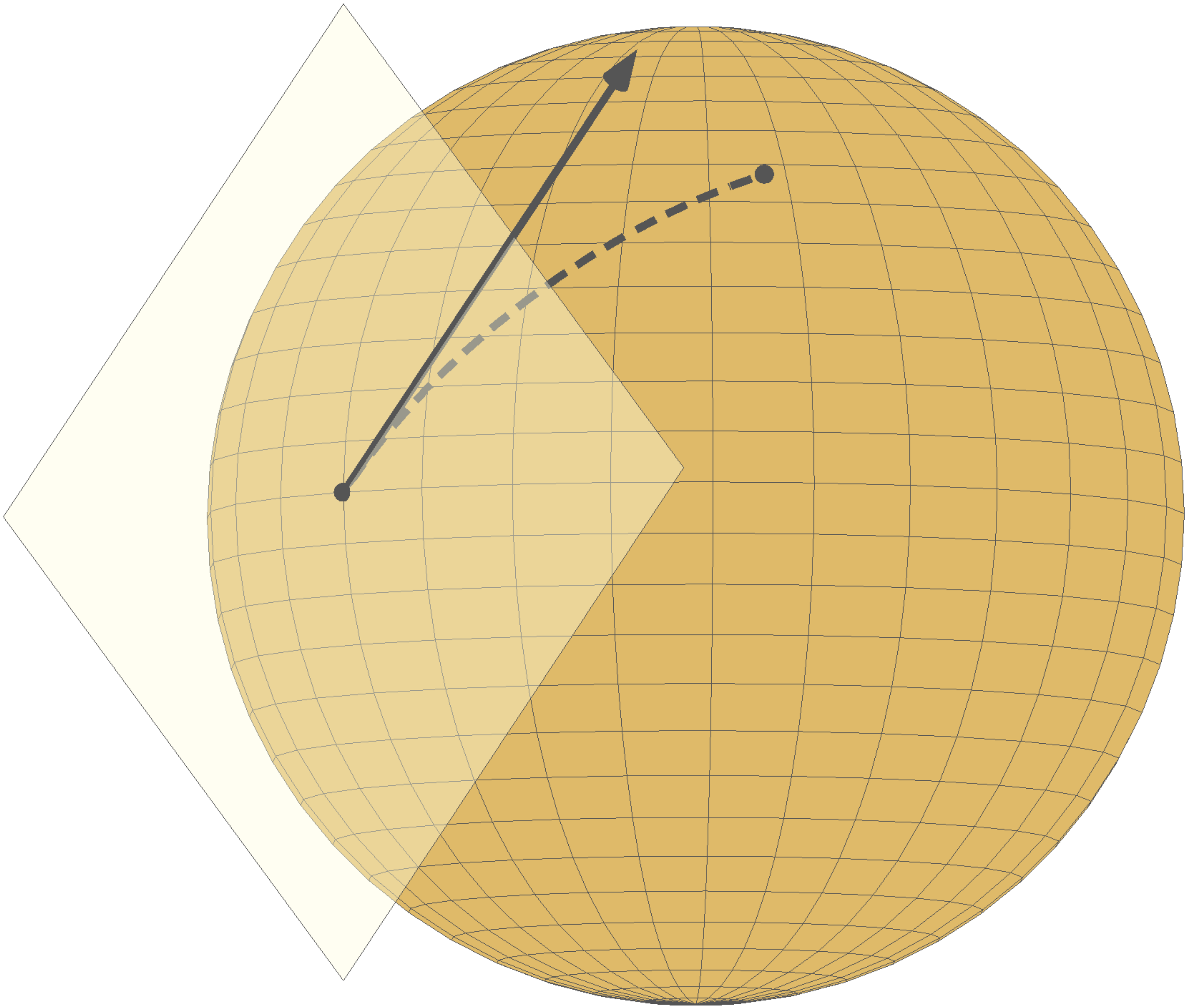
Worst-case complexity bounds for optimization on manifolds

Nicolas Boumal

Princeton University

with P.-A. Absil, N. Agarwal, B. Bullins, C. Cartis and C. Criscitiello





Target: approximate critical points

$$\|\mathrm{grad} f(x)\| \leq \varepsilon, \qquad \mathrm{Hess} f(x) \succcurlyeq -\sqrt{\varepsilon}$$

Iteration complexity?

1. Regularity assumptions?

2. Role of curvature?

Perturbed gradient descent (PGD)

Original analysis in $\mathbf{R}^d$ by Jin et al. arXiv:1902.04811

Target: approximate second-order critical point *without* Hessian query.

Algorithm: gradient descent; if gradient small, add random perturbation + make $\tilde{O}\left(\frac{1}{\sqrt{\varepsilon}}\right)$ gradient steps.

Complexity: $O\left(\frac{(\log d)^4}{\varepsilon^2}\right)$

PGD on manifolds: two approaches

By Sun, Flammarion & Fazel arXiv:1906:07355

Exponential steps on the manifold.
Bounds explicitly curvature dependent.

With Criscitiello arXiv:1906:04321

Retraction steps on the manifold,
perturbation steps in the tangent space.
No *explicit* curvature dependence.

Case study: Riemannian gradient descent

Classical analysis in $\mathbf{R}^n$.

Generalized in various ways to manifolds $\sim$2016:

Zhang & Sra,
First-order methods for geodesically convex optimization

Boumal, Absil & Cartis,
Global rates of convergence for nonconvex optimization on manifolds

Bento, Ferreira & Melo,
Iteration complexity of gradient, subgradient and proximal point methods on Riemannian manifolds

Gradient descent in $\mathbf{R}^n$

A1 $f(x) \geq f_{\text{low}}$ for all $x \in \mathbf{R}^n$

A2 ∇f is L-Lipschitz: $\|\nabla f(y) - \nabla f(x)\| \leq L\|y - x\|$

Algorithm: $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$

Complexity: $\|\nabla f(x_K)\| \leq \varepsilon$ for some $K \leq 2L(f(x_0) - f_{\text{low}})\frac{1}{\varepsilon^2}$

$$\mathbf{A2} \Rightarrow f(y) - f(x) - \langle y - x, \nabla f(x)\rangle \leq \frac{L}{2}\|y - x\|^2$$

$$\Rightarrow f(x_{k+1}) - f(x_k) + \frac{1}{L}\langle \nabla f(x_k), \nabla f(x_k)\rangle \leq \frac{1}{2L}\|\nabla f(x_k)\|^2$$

$$\Rightarrow f(x_k) - f(x_{k+1}) \geq \frac{1}{2L}\|\nabla f(x_k)\|^2$$

$$\mathbf{A1} \Rightarrow f(x_0) - f_{\text{low}} \geq \sum_{k=0}^{K} f(x_k) - f(x_{k+1}) > \frac{\varepsilon^2}{2L}(K + 1)$$

Lipschitz gradient on manifolds?

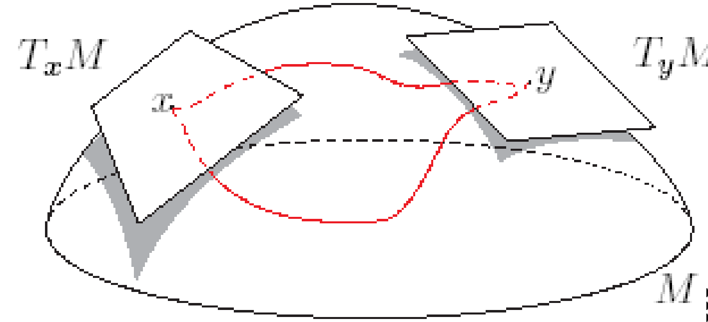


Image: D.K. Wise, IOP Science

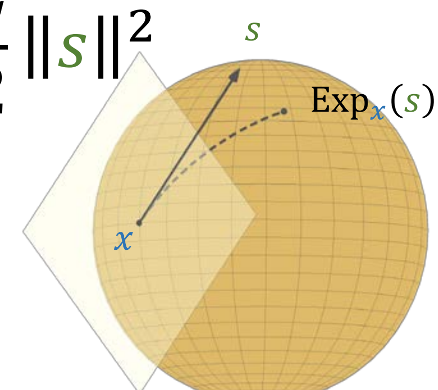
Using parallel transport and exponential map:

$$\|\mathrm{grad} f(y) - \mathrm{P}_{y \leftarrow x} \mathrm{grad} f(x)\| \leq L \cdot \mathrm{dist}(x, y),$$

$\mathrm{P}_{y \leftarrow x}$ is parallel transport along $\gamma(t) = \mathrm{Exp}_x(ts)$ from $x = \gamma(0)$ to $y = \gamma(1) = \mathrm{Exp}_x(s)$.

Implies the key quadratic bound:

$$f\big(\mathrm{Exp}_x(s)\big) - f(x) - \langle s, \mathrm{grad} f(x) \rangle \leq \frac{L}{2} \|s\|^2$$



RGD: $x_{k+1} = \mathrm{Exp}_{x_k}\left(-\frac{1}{L} \mathrm{grad} f(x_k)\right)$

A1 $f(x) \geq f_{\text{low}}$ for all $x \in \mathcal{M}$

A2 $\|\text{grad} f(y) - \text{P}_{y \leftarrow x} \text{grad} f(x)\| \leq L \cdot \text{dist}(x, y)$

Algorithm: $x_{k+1} = \text{Exp}_{x_k}\left(-\frac{1}{L} \text{grad} f(x_k)\right)$

$\Rightarrow \|\text{grad} f(x_K)\| \leq \varepsilon$ with $K \leq 2L(f(x_0) - f_{\text{low}}) \frac{1}{\varepsilon^2}$

1. Curvature-free complexity!

2. Retractions instead of exponential map?

A1 $f(x) \geq f_{\text{low}}$ for all $x \in \mathcal{M}$

A2 $f\big(\mathrm{Exp}_x(s)\big) - f(x) - \langle s, \mathrm{grad} f(x) \rangle \leq \frac{L}{2} \|s\|^2$

Algorithm: $x_{k+1} = \mathrm{Exp}_{x_k}\left(-\frac{1}{L}\mathrm{grad} f(x_k)\right)$

$\Rightarrow \|\mathrm{grad} f(x_K)\| \leq \varepsilon$ with $K \leq 2L(f(x_0) - f_{\text{low}})\frac{1}{\varepsilon^2}$

1. Curvature-free complexity.

2. Retractions instead of exponential map?

A1 $f(x) \geq f_{\text{low}}$ for all $x \in \mathcal{M}$

A2 $f\big(\mathrm{Retr}_x(s)\big) - f(x) - \langle s, \mathrm{grad} f(x) \rangle \leq \frac{L}{2} \|s\|^2$

Algorithm: $x_{k+1} = \mathrm{Retr}_{x_k}\left(-\frac{1}{L} \mathrm{grad} f(x_k)\right)$

$\Rightarrow \|\mathrm{grad} f(x_K)\| \leq \varepsilon$ with $K \leq 2L(f(x_0) - f_{\text{low}}) \frac{1}{\varepsilon^2}$

1. Curvature-free complexity.

2. Retractions instead of exponential map.

Other algorithms on manifolds

Trust regions: curvature free

With Absil & Cartis, arXiv:1605.08101

R-SPIDER (var. red. stochastic): curvature free

Zhang, Zhang & Sra, arXiv:1811.04194

Adaptive regularization with cubics: unclear

With Agarwal, Bullins & Cartis, arXiv:1806.00065; see also Zhang & Zhang, arXiv:1805.05565

Perturbed gradient descent: unclear (trickier)

manopt.org

Manopt | Home | Tutorial | Forum | About | Contact

Welcome to Manopt!

A Matlab toolbox for optimization on manifolds

Optimization on manifolds is a powerful paradigm to address nonlinear optimization probl
various types of constraints that arise naturally in applications, such as orthonormality or lo

Download | Get started

With Mishra, Absil & Sepulchre

pymanopt.github.io

python

Pymanopt

Pymanopt is a Python toolbox for optimization on manifolds, that computes gradients and Hessia
builds upon the Matlab toolbox Manopt but is otherwise independent of it. Pymanopt aims to low
users wishing to use state of the art techniques for optimization on manifolds, by relying on autor
for computing gradients and Hessians, saving users time and saving them from potential calculation an
implementiation errors.

Pymanopt is modular and hence easy to use. All of the automatic differentiation is done behind the scer
the amount of setup the user needs to do is minimal. Usually only the following steps are required:

1. Instantiate a manifold $\mathcal{M}$ to optimise over
2. Define a cost function $f : \mathcal{M} \to \mathbb{R}$ to minimise

Lead by Townsend, Koep, Weichwald

(Book in progress.)

manoptjl.org

julia

Manopt.jl

v0.1.0

Search docs

Proximal Maps

Helpers

Data

» Home

Welcome to Manopt.jl

`Manopt.Manopt` — *Module.*

`Manopt.jl` – Optimization on Manifolds in Julia.

source

For a function $f\colon \mathcal{M} \to \mathbb{R}$ defined on a Riemannian manifold $\mathcal{M}$ we aim to solve

Lead by Ronny Bergmann