Optimization on manifolds
What's the worst that could happen?

Nicolas Boumal
Princeton University

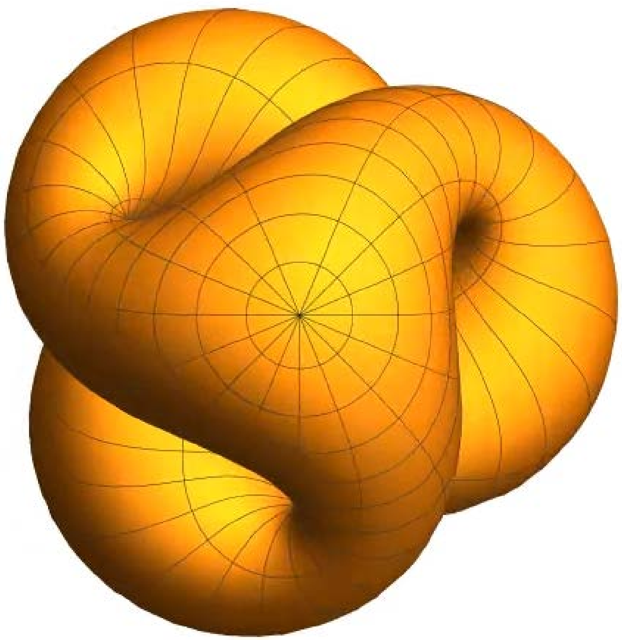
various parts with P.-A. Absil, N. Agarwal, B. Bullins, C. Cartis and C. Criscitiello

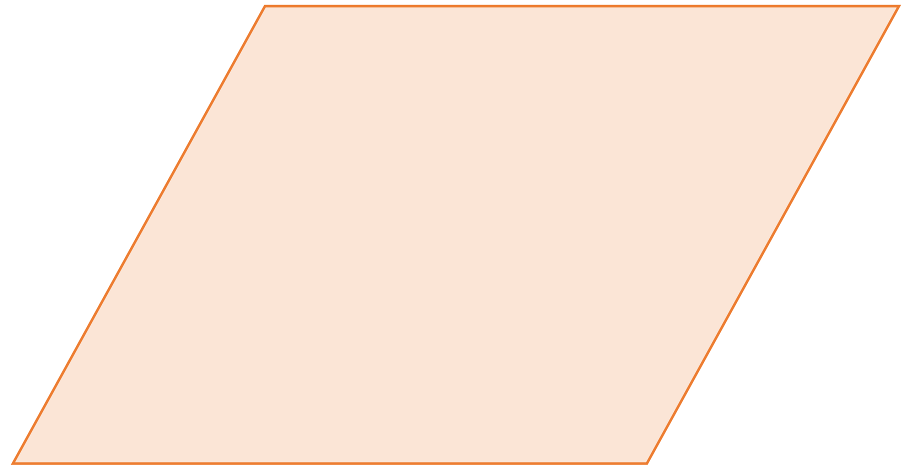
https://nypost.com/2019/02/19/youtube-is-helping-the-flat-earth-conspiracy-movement-grow/

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

“Apparently, some people believe the Earth is shaped like a donut.”

—Vice.com, Nov. 2018



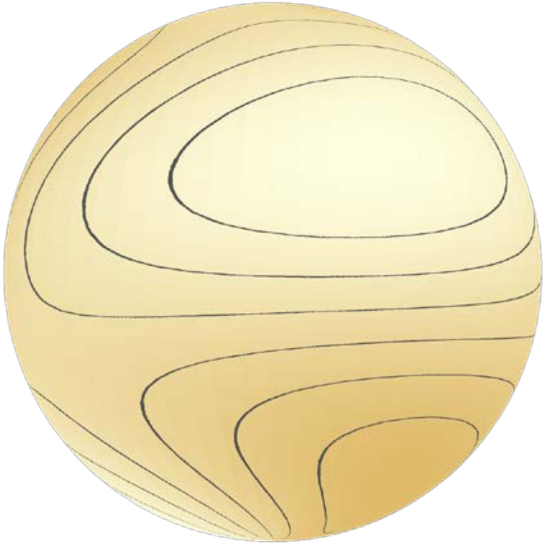


"How does curvature affect optimization?"

Picture: http://homepages.math.uic.edu/~ddumas/teaching/2017/fall/math549/boy/

Optimization on smooth manifolds

$$\min_x f(x) \text{ subject to } x \in \mathcal{M}$$



Linear spaces

Unconstrained; linear equality constraints

Low rank (matrices, tensors)

Recommender systems, large-scale Lyapunov equations, …

Orthonormality (Grassmann, Stiefel, rotations)

Dictionary learning, structure from motion, SLAM, PCA, ICA, SBM,…

Positivity (positive definiteness, positive orthant)

Metric learning, Gaussian mixtures, diffusion tensor imaging, …

Symmetry (quotient manifolds)

Invariance under group actions

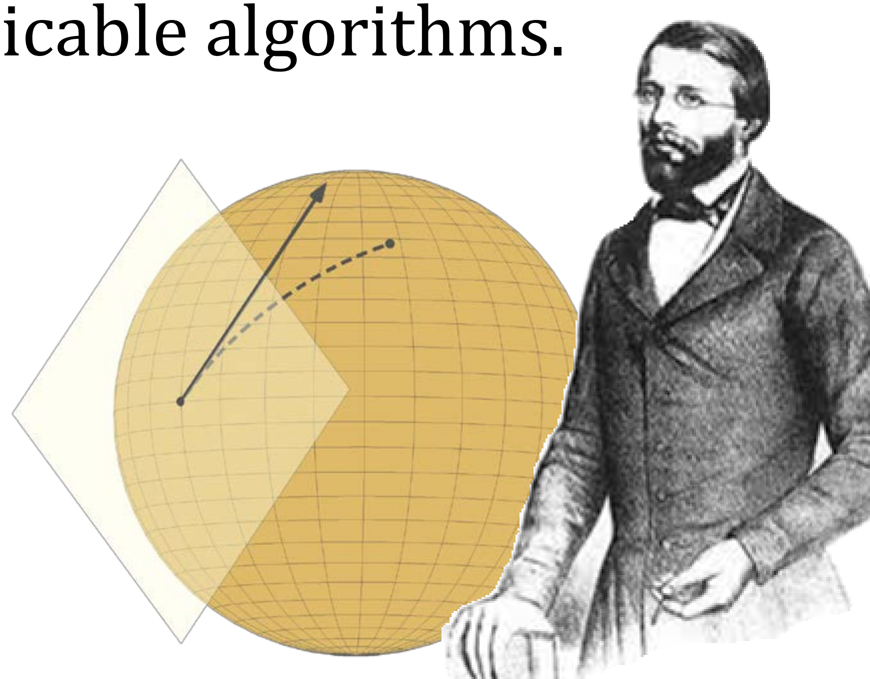
A Riemannian structure gives us gradients and Hessians

The essential tools of smooth optimization are defined generally on Riemannian manifolds.

Unified theory, broadly applicable algorithms.

First ideas from the '70s.

First practical in the '90s.



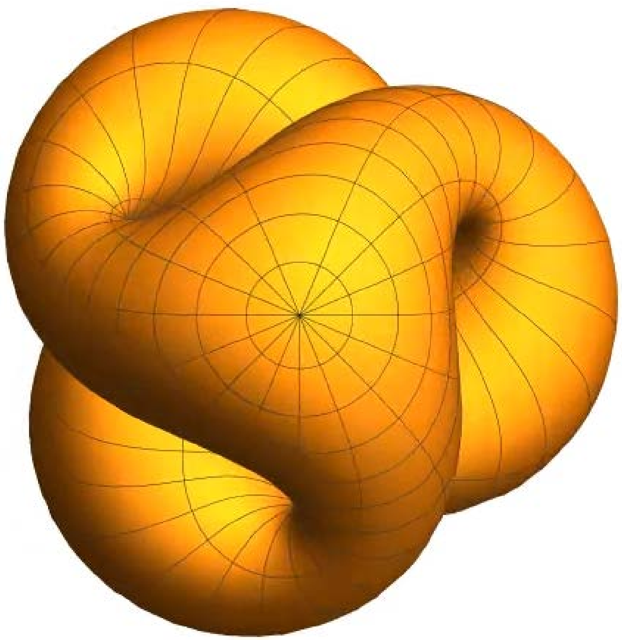
Non-convexity: reasonable targets

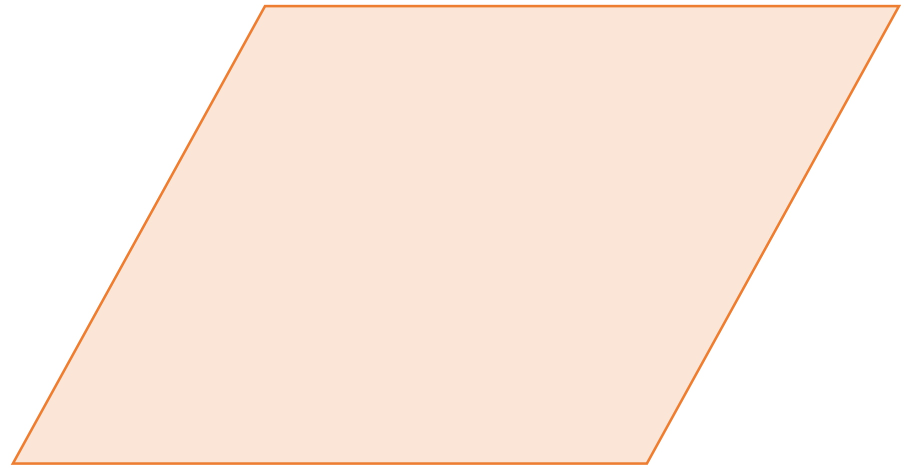
$$\|\mathrm{grad} f(x)\| \leq \varepsilon, \qquad \lambda_{\min}\big(\mathrm{Hess} f(x)\big) \geq -\sqrt{\varepsilon}$$

Want: worst-case iteration complexity

Particularly relevant for benign non-convexity

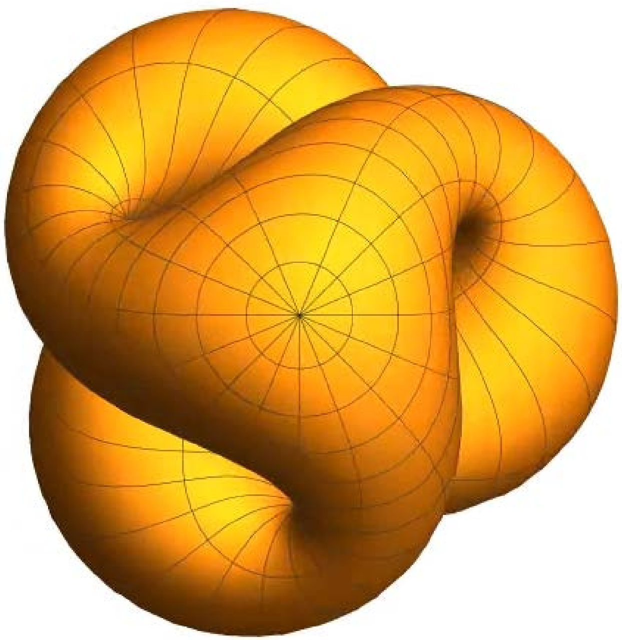
- Burer-Monteiro for SDPs under some conditions
- Dictionary learning / sparsest vector in a subspace
- Matrix / tensor completion
- Group synchronization (variants in SBM, SLAM, SfM, …)
- (And also geodesic convexity)

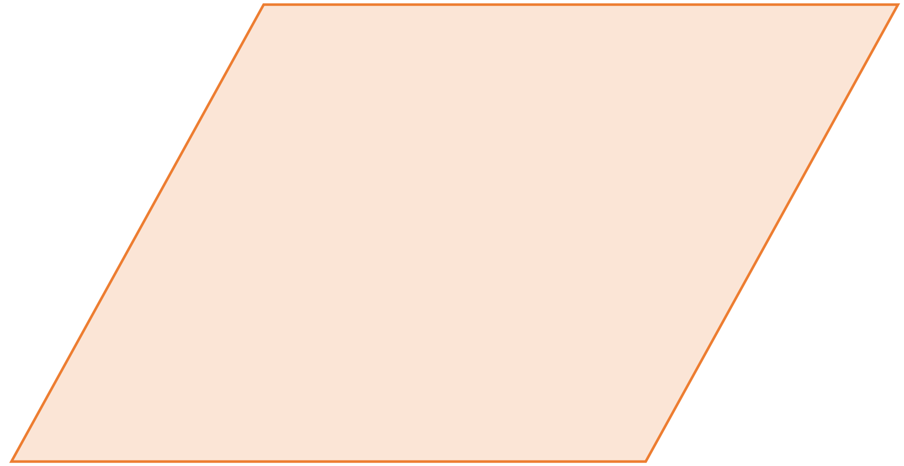




"All other things being equal, is it harder to optimize if the space is more curved?"

Picture: http://homepages.math.uic.edu/~ddumas/teaching/2017/fall/math549/boy/





Whatever that means…

“All other things being equal, is it harder to optimize if the space is more curved?”

Picture: http://homepages.math.uic.edu/~ddumas/teaching/2017/fall/math549/boy/

Does curvature impede optimization?

Message 1

**Under natural Lipschitz assumptions,
for some optimal algorithms, it does not hurt.**

Message 2

Unclear for more sophisticated algorithms.

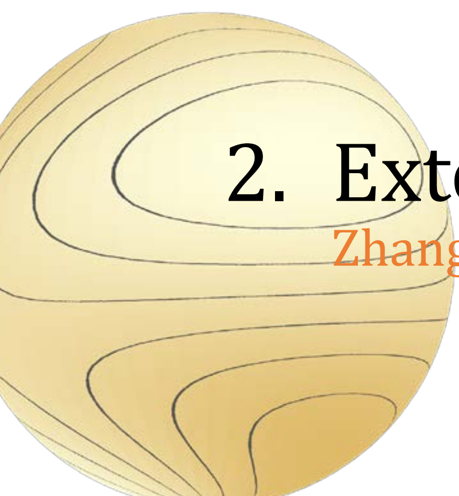
Target: approximate critical points

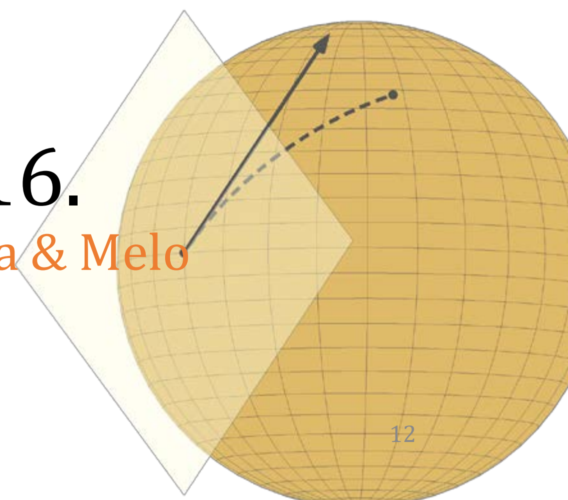
$$\|\mathrm{grad} f(x)\| \leq \varepsilon$$

Iteration complexity of gradient descent?

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

2. Extended to manifolds ~2016.
   Zhang & Sra; B., Absil & Cartis; Bento, Ferreira & Melo





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 gradients on *complete* manifolds

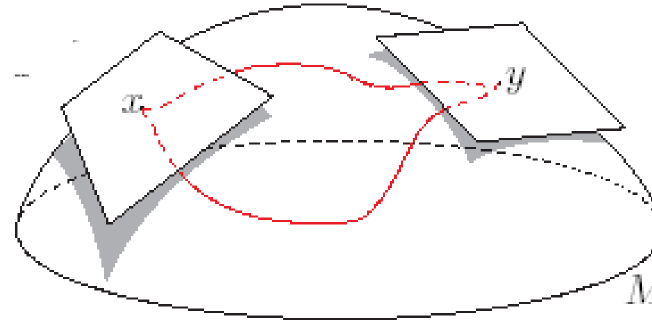


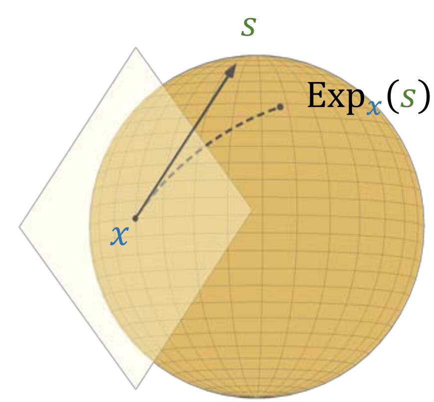
Image: D.K. Wise, IOP Science

Using parallel transport and exponential map:

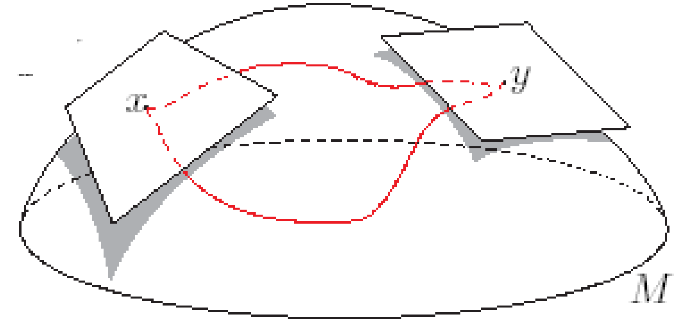
$$\left\| \mathrm{grad} f(y) - \mathrm{P}_{y \leftarrow x} \mathrm{grad} f(x) \right\| \leq L \cdot \mathrm{dist}(x, y),$$

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

Already used for optimization in 1998 (da Cruz de Neto)



Lipschitz gradients on complete manifolds



Using parallel transport and exponential map:

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

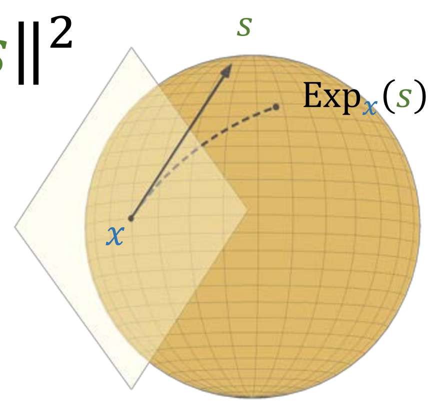
$\mathrm{P}_{y\leftarrow x}$ is parallel transport along $\gamma(t) = \mathrm{Exp}_x(ts)$ from $x = \gamma(0)$ to $y = \gamma(1) = \mathrm{Exp}_x(s)$.

Image: D.K. Wise, IOP Science

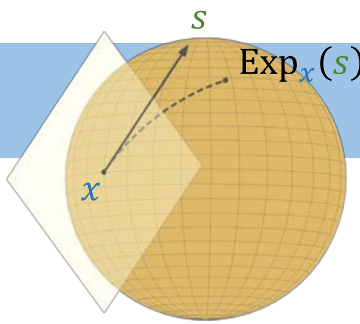
Implies the key quadratic bound:

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

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



Gradient descent on $\mathcal{M}$



A1 $f(x) \geq f_{\mathrm{low}}$ 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: $x_{k+1} = \mathrm{Exp}_{x_k}\left(-\frac{1}{L}\mathrm{grad} f(x_k)\right)$

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

$$\textbf{A2} \Rightarrow f(x_{k+1}) - f(x_k) + \frac{1}{L}\|\mathrm{grad} f(x_k)\|^2 \leq \frac{1}{2L}\|\mathrm{grad} f(x_k)\|^2$$

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

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

Gradient descent on $\mathcal{M}$

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

Same as in $\mathbf{R}^n$, where it is tight and optimal.

In particular, it is dimension free and curvature free!

Second-order target

$$\|\mathrm{grad} f(x)\| \leq \varepsilon, \qquad \lambda_{\min}\left(\mathrm{Hess} f(x)\right) \geq -\sqrt{\varepsilon}$$

Assume Lipschitz continuous Riemannian Hessian.

Implies Riemannian versions of the usual inequalities.

Riemannian trust regions: $O\left(\varepsilon^{-2.5}\right)$

With Absil and Cartis, arXiv:1605.08101

Riemannian cubic regularization: $O\left(\varepsilon^{-1.5}\right)$

With Agarwal, Bullins and Cartis, arXiv:1806.00065; See also Zhang and Zhang, arXiv:1805.05565

These complexities also dimension and curvature free.

Cubic regularization is also optimal in $\mathbf{R}^n$.

What is the role of curvature so far?

In $\mathbf{R}^n$, GD and ARC are optimal under Lipschitz.

Same upper bounds on manifolds.

Thus, curvature does not hurt in those cases.

Might it help? What about other classes/algos?

Do Lipschitz constants hide curvature?

For more sophisticated algorithms, known bounds suffer from curvature

Several recent papers study advanced algorithms for, e.g., Hessian-free saddle escapes and acceleration.

Their analyses in $\mathbf{R}^n$ use Lipschitzness in more ways than the simple inequalities we used earlier.

Proof techniques often involve triangles on manifolds to track iterates: curvature comes up.

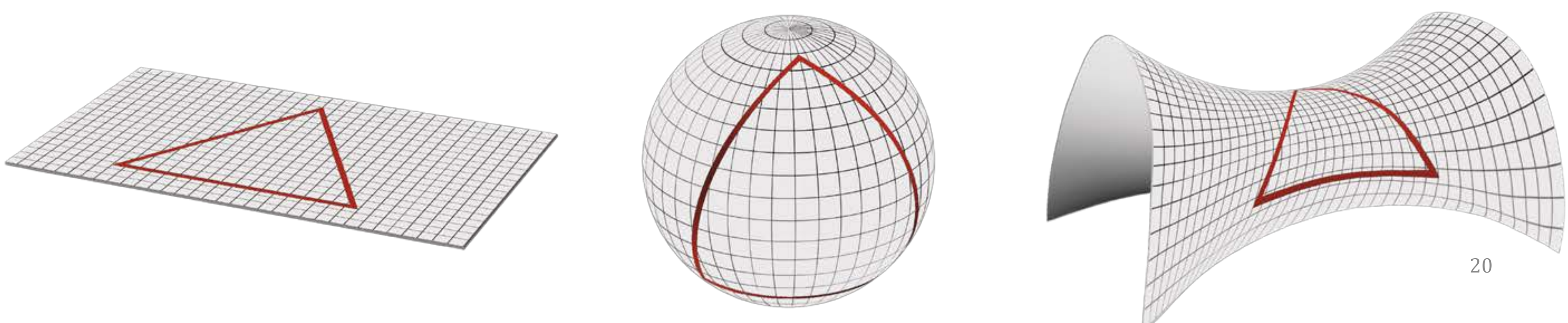


Image: https://januscosmologicalmodel.com/negativemass

Does curvature affect Lipschitz cnsts?

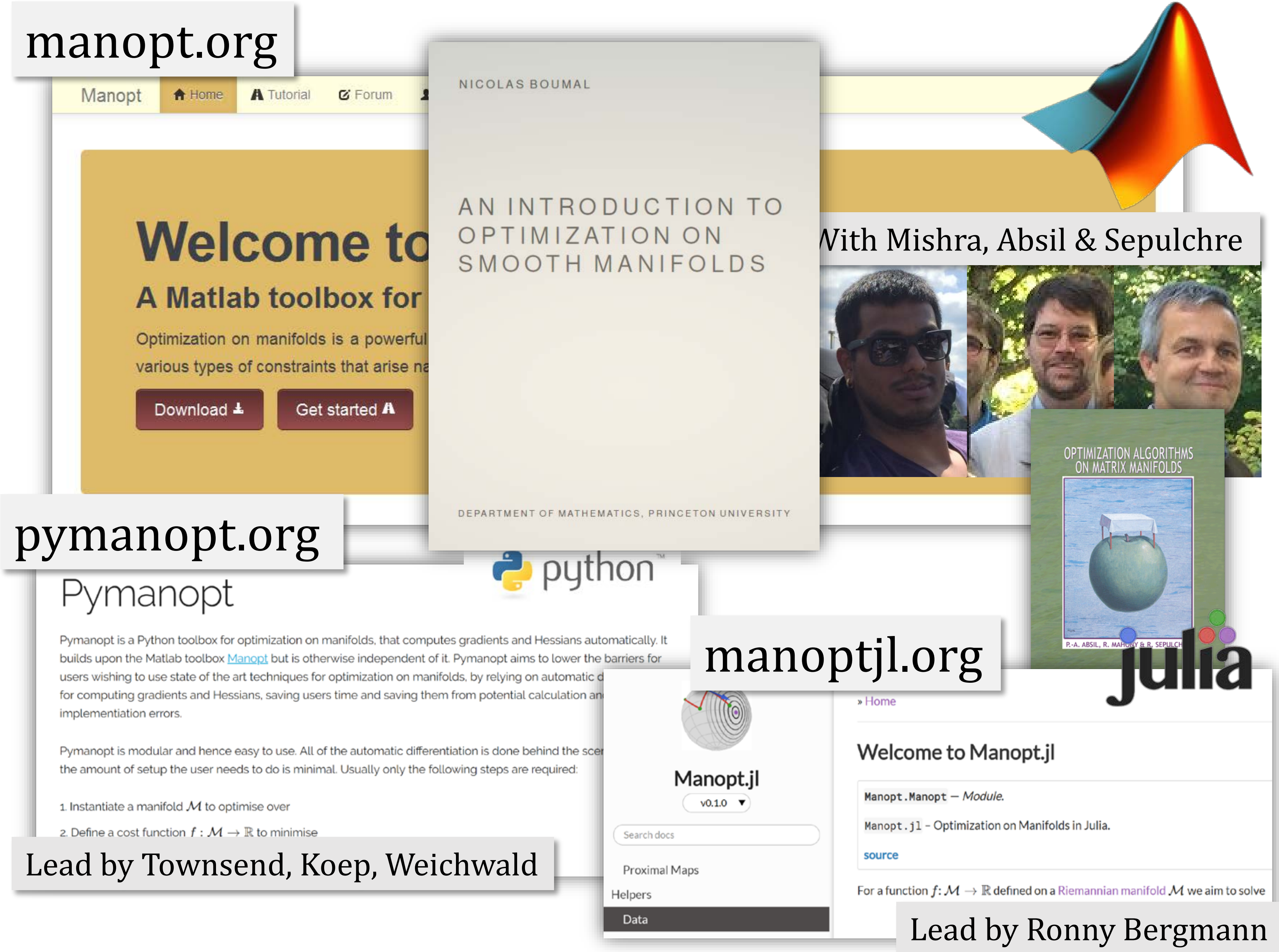
Here are two possible ways to address this.

Consider $f: \mathbf{R}^n \to \mathbf{R}$ with Lipschitz gradient:

1. Restrict to a Riemannian submanifold $\mathcal{M} \subset \mathbf{R}^n$.
   Constant L is affected by *extrinsic* curvature.

2. Deform $\mathbf{R}^n$ into a Riemannian manifold.
   Derivative of metric does affect L,
   but link with curvature is indirect.

Case in point: one-dimensional manifolds have *no* intrinsic curvature, yet see both effects.



manopt.org
Manopt
Home
Tutorial
Forum
Welcome to
A Matlab toolbox for
Optimization on manifolds is a powerful
various types of constraints that arise na
Download
Get started
NICOLAS BOUMAL
AN INTRODUCTION TO OPTIMIZATION ON SMOOTH MANIFOLDS
DEPARTMENT OF MATHEMATICS, PRINCETON UNIVERSITY
With Mishra, Absil & Sepulchre
OPTIMIZATION ALGORITHMS ON MATRIX MANIFOLDS
pymanopt.org
python
Pymanopt
Pymanopt is a Python toolbox for optimization on manifolds, that computes gradients and Hessians automatically. It builds upon the Matlab toolbox Manopt but is otherwise independent of it. Pymanopt aims to lower the barriers for users wishing to use state of the art techniques for optimization on manifolds, by relying on automatic d
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 sce
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
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

Riemannian Lipschitz, **with** Riemannian curvature in bounds

RSVRG (Zhang, Reddi & Sra 2016)
SGD with averaging (Tripuraneni, Flammarion, Bach & Jordan 2018)
Perturbed gradient descent (Sun, Flammarion & Fazel 2019)
More stochastic methods (Kasai, Sato & Mishra 2016/17; Zhang et al. 2016)
Geodesically convex optimization (Zhang & Sra 2016)
  Also with (steps toward) acceleration (Zhang & Sra; Alimisis et al. 2020)

Riemannian Lipschitz, **no** Riemannian curvature in bounds

Gradient descent (Bento, Ferreira & Melo 2017)
Trust-regions (B., Absil & Cartis 2018)
Adaptive regularization with cubics (Agarwal, B., Bullins & Cartis 2019)
R-Spider (stochastic) (Zhang, Zhang & Sra 2018)
Frank-Wolfe (Weber & Sra 2017)

Pullback Lipschitz, no curvature in bounds, but maybe hidden

Gradient descent (B., Absil & Cartis 2018)
Trust-regions
Adaptive regulization with cubics
Perturbed gradient descent (Criscitiello & Boumal, 2019)