UC San Diego

Geometry Meets Machine Learning

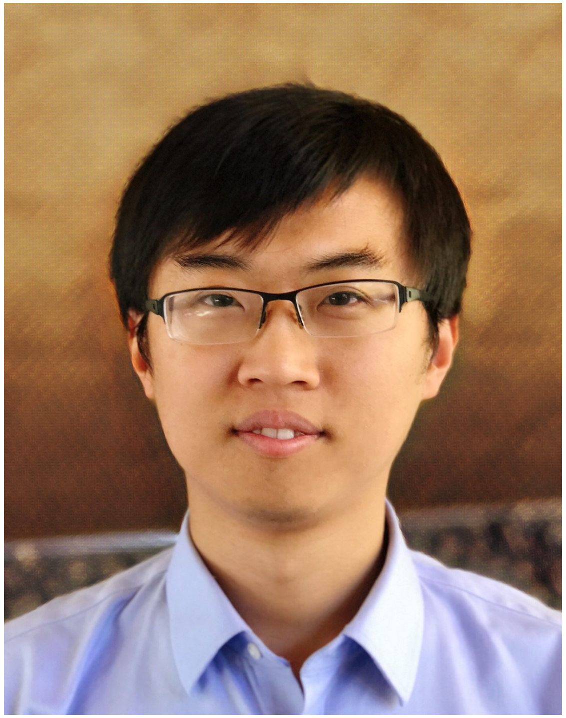
Instructor: Hao Su

Syllabus

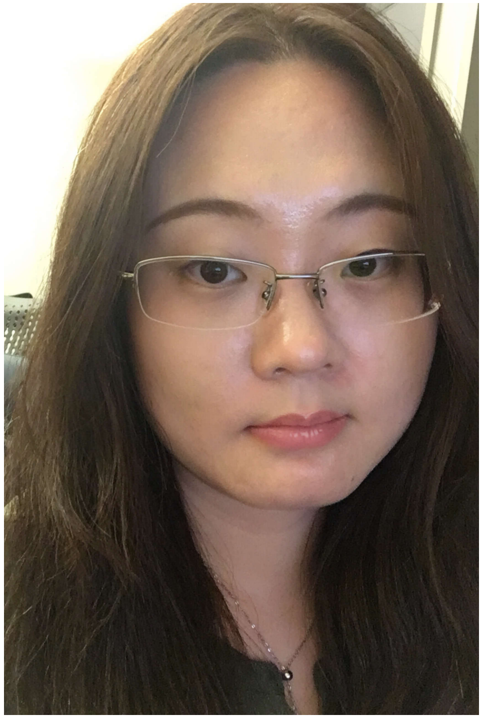
- **Course website**
  - http://geoml.github.io
- **Five units**
  - Geometry Basics
  - Laplacian Operator and Spectral Graph Theory
  - Data Embedding and Deep Learning
  - Map Networks
  - Deep Learning on 3D Data

Who we are?

Instructor: Hao Su



Teaching Assistant: Meng Song



Logistics

Grading (tentative)

- Quizzes 20%
- Course project presentation 40%
- Course project writeup 40%
- There will not be a final exam.

Pre-requisite

- Try to be as self-contained as possible
- Proficiency in Python and Matlab
- Calculus, Linear Algebra
- Machine learning
  - Classification
  - Optimization

Numerical Tools for Geometry

Credit: MIT 6.838, Justin Solomon

Motivation

Numerical problems abound

in modern geometry applications.

Quick summary!

Mostly for common ground: You may already know this material.
First half is important; remainder summarizes interesting recent tools.

Two Roles

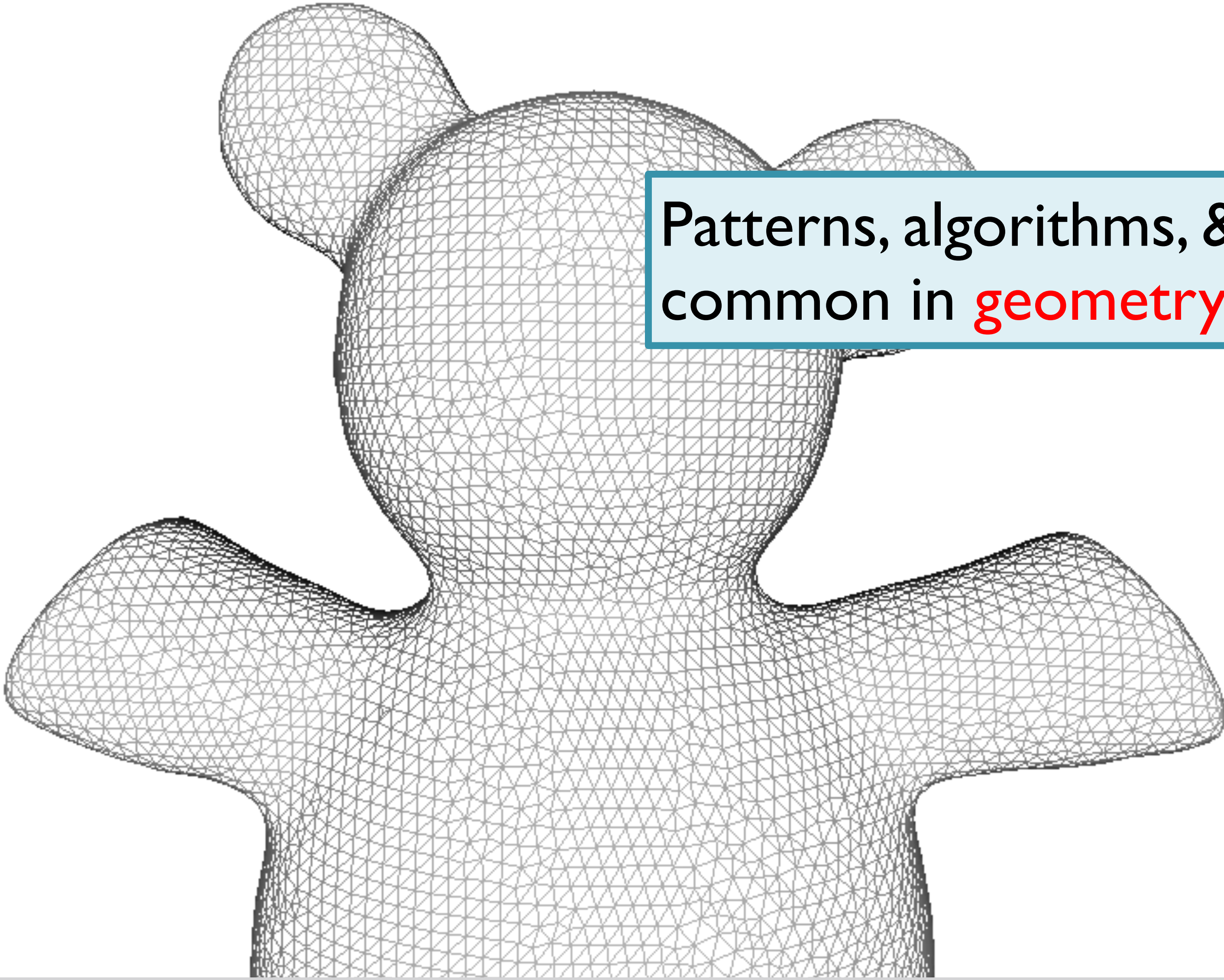
- Client

  Which optimization tool is relevant?

- Designer

  Can I design an algorithm for this problem?

Our Bias



Patterns, algorithms, & examples common in geometry.

Numerical analysis is a huge field.

Rough Plan

- Linear problems
- Unconstrained optimization
- Equality-constrained optimization

Rough Plan

- Linear problems
- Unconstrained optimization
- Equality-constrained optimization

Vector Spaces and Linear Operators

$$\mathcal{L}[\vec{x} + \vec{y}] = \mathcal{L}[\vec{x}] + \mathcal{L}[\vec{y}]$$

$$\mathcal{L}[c\vec{x}] = c\mathcal{L}[\vec{x}]$$

Abstract Example

$$C^\infty(\mathbb{R})$$

$$\mathcal{L}[f] := {}^{df}\!/_{dx}$$

In Finite Dimensions

$$\underbrace{A}_{\text{matrix}} \underbrace{\vec{x}}_{\text{vector}}$$

$$\underbrace{\vec{x} \longmapsto A\vec{x}}_{\text{linear operator}}$$

Linear System of Equations

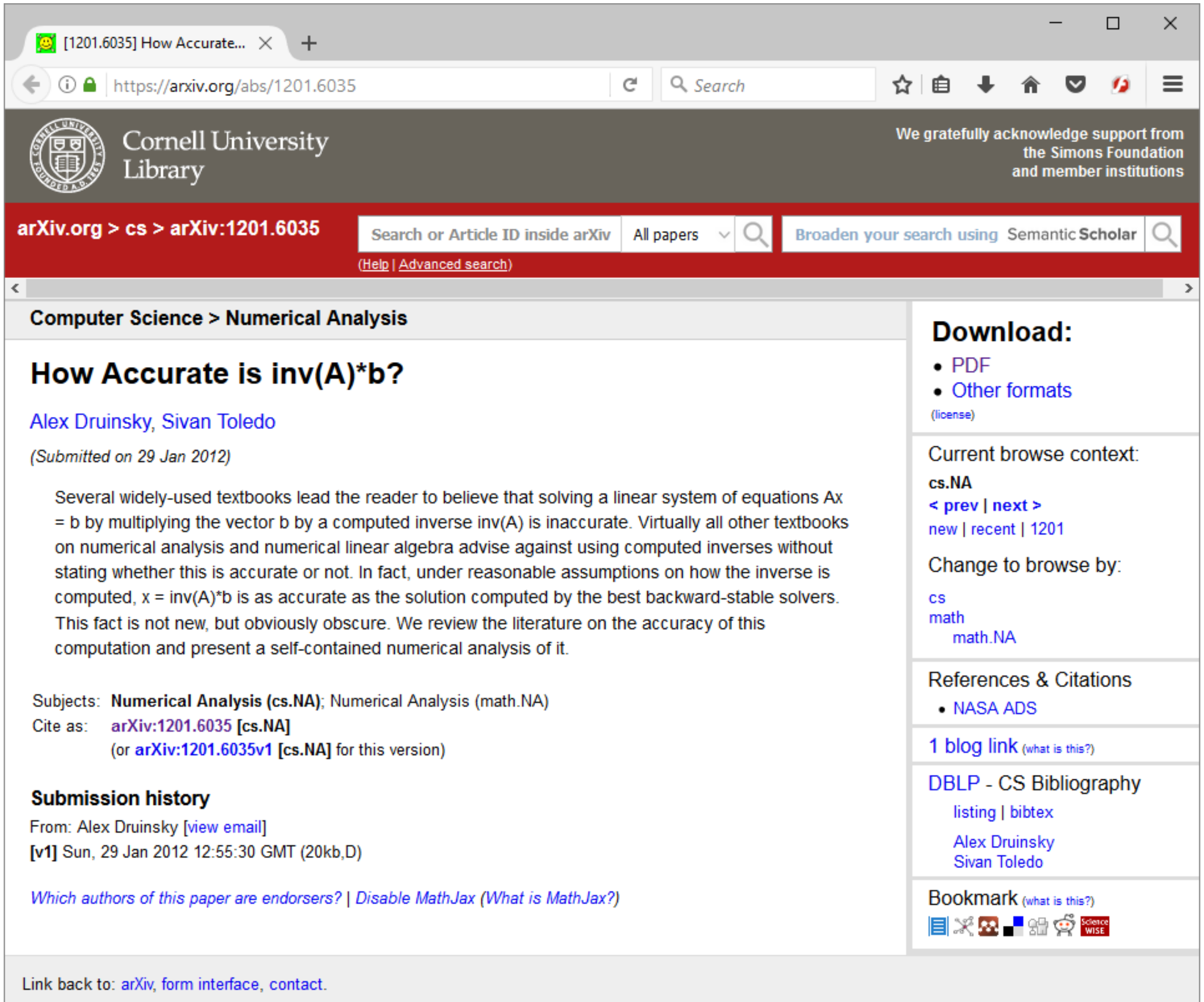
$$\begin{pmatrix} & & \\ & A & \\ & & \end{pmatrix} \begin{pmatrix} \\ \vec{x} \\ \\ \end{pmatrix} = \begin{pmatrix} \\ \vec{b} \\ \\ \end{pmatrix}$$

Simple “inverse problem”

Common Strategies

- Gaussian elimination
  - $O(n^3)$ time to solve $Ax=b$ or to invert
- **But:** Inversion is unstable and slower!
- Never ever compute A^{-1} if you can avoid it.

Interesting Perspective



[1201.6035] How Accurate...

https://arxiv.org/abs/1201.6035

Cornell University Library

We gratefully acknowledge support from the Simons Foundation and member institutions

arXiv.org > cs > arXiv:1201.6035

Search or Article ID inside arXiv | All papers | Broaden your search using Semantic Scholar

(Help | Advanced search)

Computer Science > Numerical Analysis

How Accurate is inv(A)*b?

Alex Druinsky, Sivan Toledo

(Submitted on 29 Jan 2012)

Several widely-used textbooks lead the reader to believe that solving a linear system of equations Ax = b by multiplying the vector b by a computed inverse inv(A) is inaccurate. Virtually all other textbooks on numerical analysis and numerical linear algebra advise against using computed inverses without stating whether this is accurate or not. In fact, under reasonable assumptions on how the inverse is computed, x = inv(A)*b is as accurate as the solution computed by the best backward-stable solvers. This fact is not new, but obviously obscure. We review the literature on the accuracy of this computation and present a self-contained numerical analysis of it.

Subjects: **Numerical Analysis (cs.NA)**; Numerical Analysis (math.NA)
Cite as: arXiv:1201.6035 **[cs.NA]**
(or **arXiv:1201.6035v1 [cs.NA]** for this version)

Submission history

From: Alex Druinsky [view email]
[v1] Sun, 29 Jan 2012 12:55:30 GMT (20kb,D)

Which authors of this paper are endorsers? | *Disable MathJax* (*What is MathJax?*)

Download:
- PDF
- Other formats

(license)

Current browse context:
cs.NA
< prev | next >
new | recent | 1201

Change to browse by:
cs
math
math.NA

References & Citations
- NASA ADS

1 blog link (what is this?)

DBLP - CS Bibliography
listing | bibtex
Alex Druinsky
Sivan Toledo

Bookmark (what is this?)

Link back to: arXiv, form interface, contact.

Simple Example

$$\frac{d^2 f}{dx^2} = g, f(0) = f(1) = 0$$

$$\begin{pmatrix} -2 & 1 & & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & & \ddots & & & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{pmatrix}$$

Structure?

$$
\begin{pmatrix}
-2 & 1 & & & & \\
1 & -2 & 1 & & & \\
 & 1 & -2 & 1 & & \\
 & & & \ddots & & \\
 & & & 1 & -2 & 1 \\
 & & & & 1 & -2
\end{pmatrix}
$$

Linear Solver Considerations

- Never construct explicitly
  (if you can avoid it)

- Added structure helps
  Sparsity, symmetry, positive definiteness, bandedness

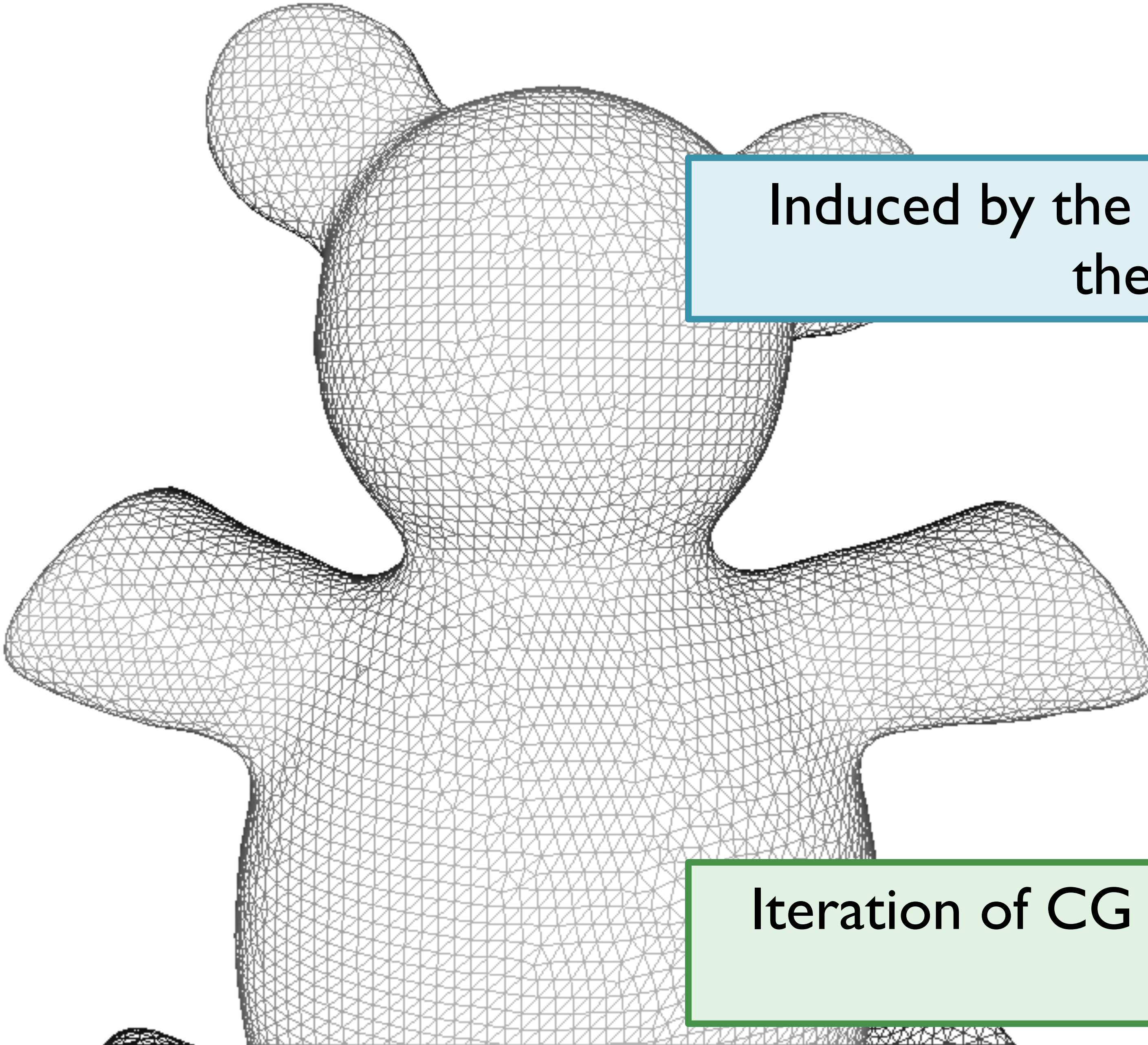
$$\texttt{inv(A)*b} \ll \texttt{(A'*A)}\backslash\texttt{(A'*b)} \ll \texttt{A}\backslash\texttt{b}$$

Two Classes of Solvers

- Direct (*explicit* matrix)
  - **Dense:** Gaussian elimination/LU, QR for least-squares
  - **Sparse:** Reordering (SuiteSparse, Eigen)

- Iterative (*apply* matrix repeatedly)
  - **Positive definite:** Conjugate gradients
  - **Symmetric:** MINRES, GMRES
  - **Generic:** LSQR

Very Common: Sparsity



Induced by the connectivity of the triangle mesh.

Iteration of CG has local effect

Precondition!

Rough Plan

- Linear problems
- Unconstrained optimization
- Equality-constrained optimization

Optimization Terminology

$$\min_{x \in \mathbb{R}^n} \boxed{f(x)}$$

$$\text{s.t. } g(x) = 0$$

$$h(x) \geq 0$$

Objective ("Energy Function")

Optimization Terminology

$$\min_{x \in \mathbb{R}^n} f(x)$$
$$\text{s.t. } \boxed{g(x) = 0}$$
$$h(x) \geq 0$$

Equality Constraints

Optimization Terminology

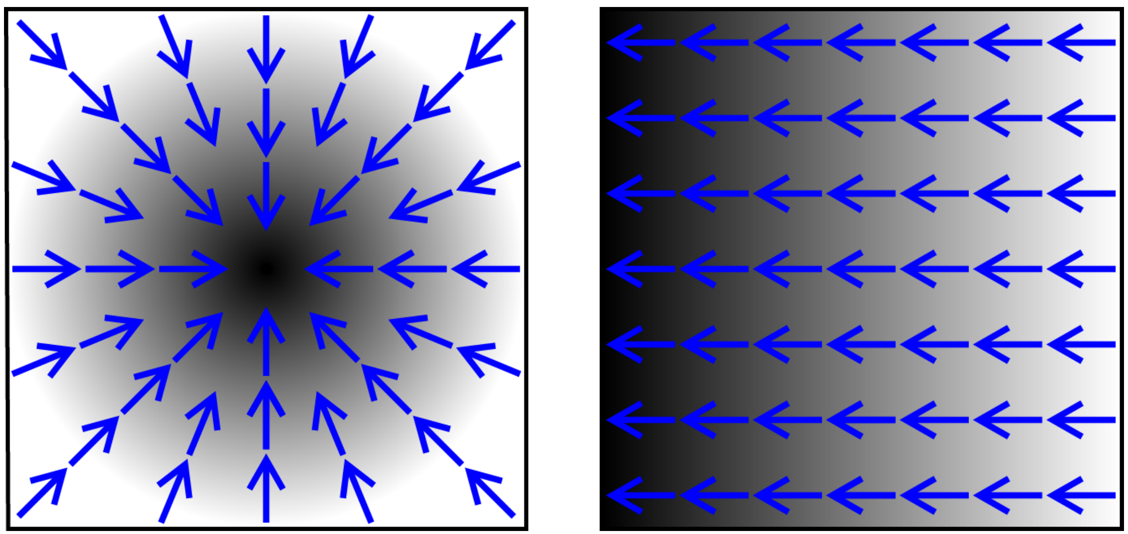
$$\min_{x\in\mathbb{R}^n} f(x)$$
$$\text{s.t. } g(x) = 0$$
$$h(x) \geq 0$$

Inequality Constraints

Notions from Calculus

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

$$\to \nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right)$$



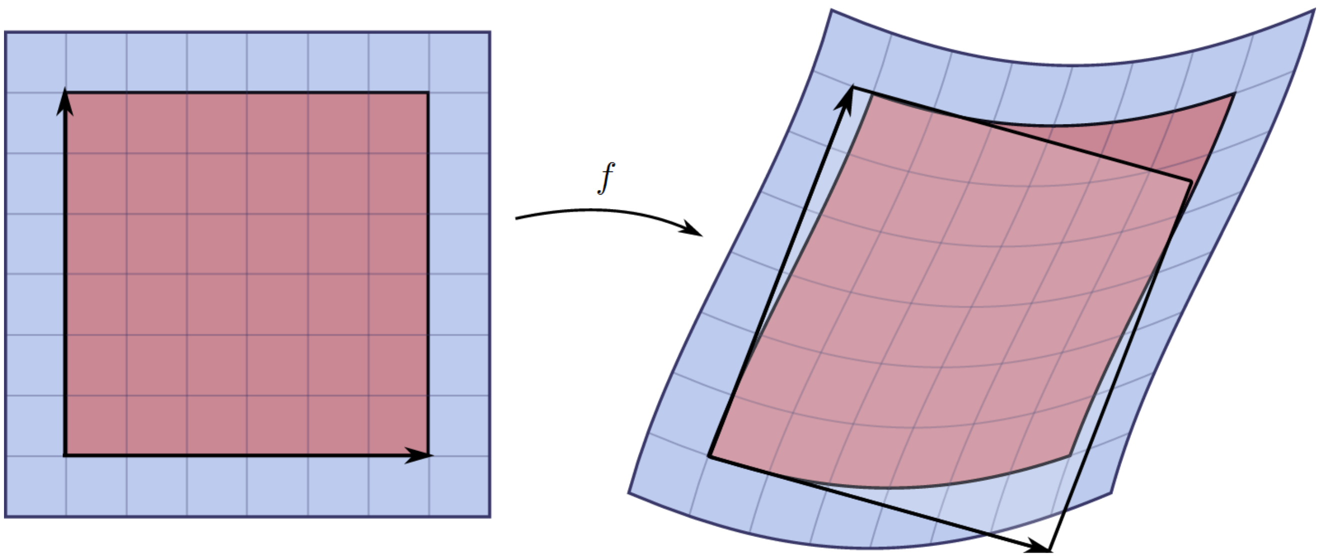
https://en.wikipedia.org/?title=Gradient

Gradient

Notions from Calculus

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

$$\to (Df)_{ij} = \frac{\partial f_i}{\partial x_j}$$

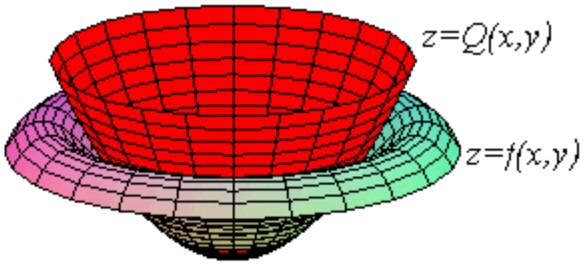


https://en.wikipedia.org/wiki/Jacobian_matrix_and_determinant

Jacobian

Notions from Calculus

$$f : \mathbb{R}^n \to \mathbb{R} \to H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$



$$f(x) \approx f(x_0) + \nabla f(x_0)^\top (x - x_0) + (x - x_0)^\top H f(x_0)(x - x_0)$$

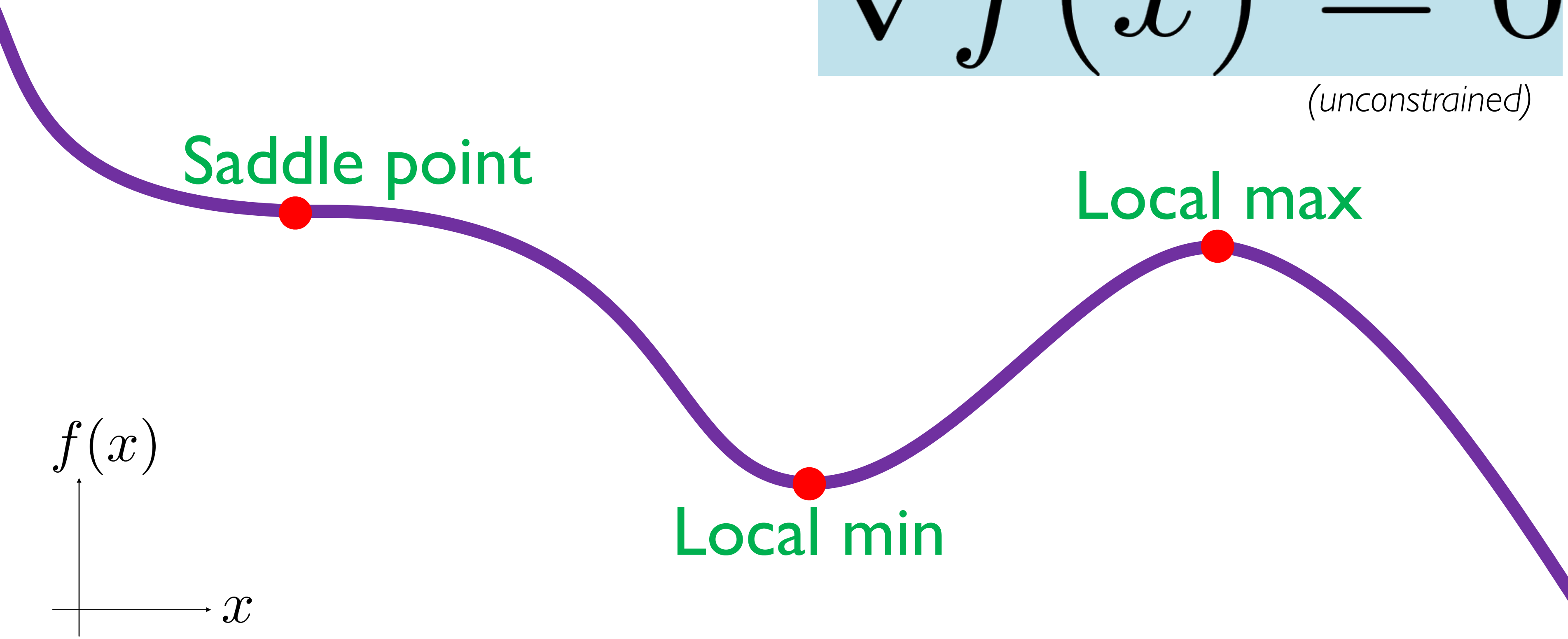
http://math.etsu.edu/multicalc/prealpha/Chap2/Chap2-5/10-3a-t3.gif

Hessian

Optimization to Root-Finding

$$\nabla f(x) = 0$$

(unconstrained)



Critical point

Encapsulates Many Problems

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \ f(x) \\ \text{s.t.}\, g(x) = 0 \\ h(x) \geq 0 \end{aligned}$$

$$Ax = b \leftrightarrow f(x) = \|Ax - b\|_2$$

$$Ax = \lambda x \leftrightarrow f(x) = \|Ax\|_2, g(x) = \|x\|_2 - 1$$

$$\text{Roots of } g(x) \leftrightarrow f(x) = 0$$

Q:

How effective are
generic
optimization tools?

Generic Advice

Try the

simplest solver first.

Quadratic with Linear Equality

$$\begin{aligned} \min_x \quad & \frac{1}{2} x^\top A x - b^\top x + c \\ \text{s.t.} \quad & Mx = v \end{aligned}$$

(assume A is symmetric and positive definite)

$$\begin{pmatrix} A & M^\top \\ M & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} b \\ v \end{pmatrix}$$

Useful Document

The Matrix Cookbook

Petersen and Pedersen

http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/3274/pdf/imm3274.pdf

Special Case: Least-Squares

$$\min_x \frac{1}{2}\|Ax - b\|_2^2$$

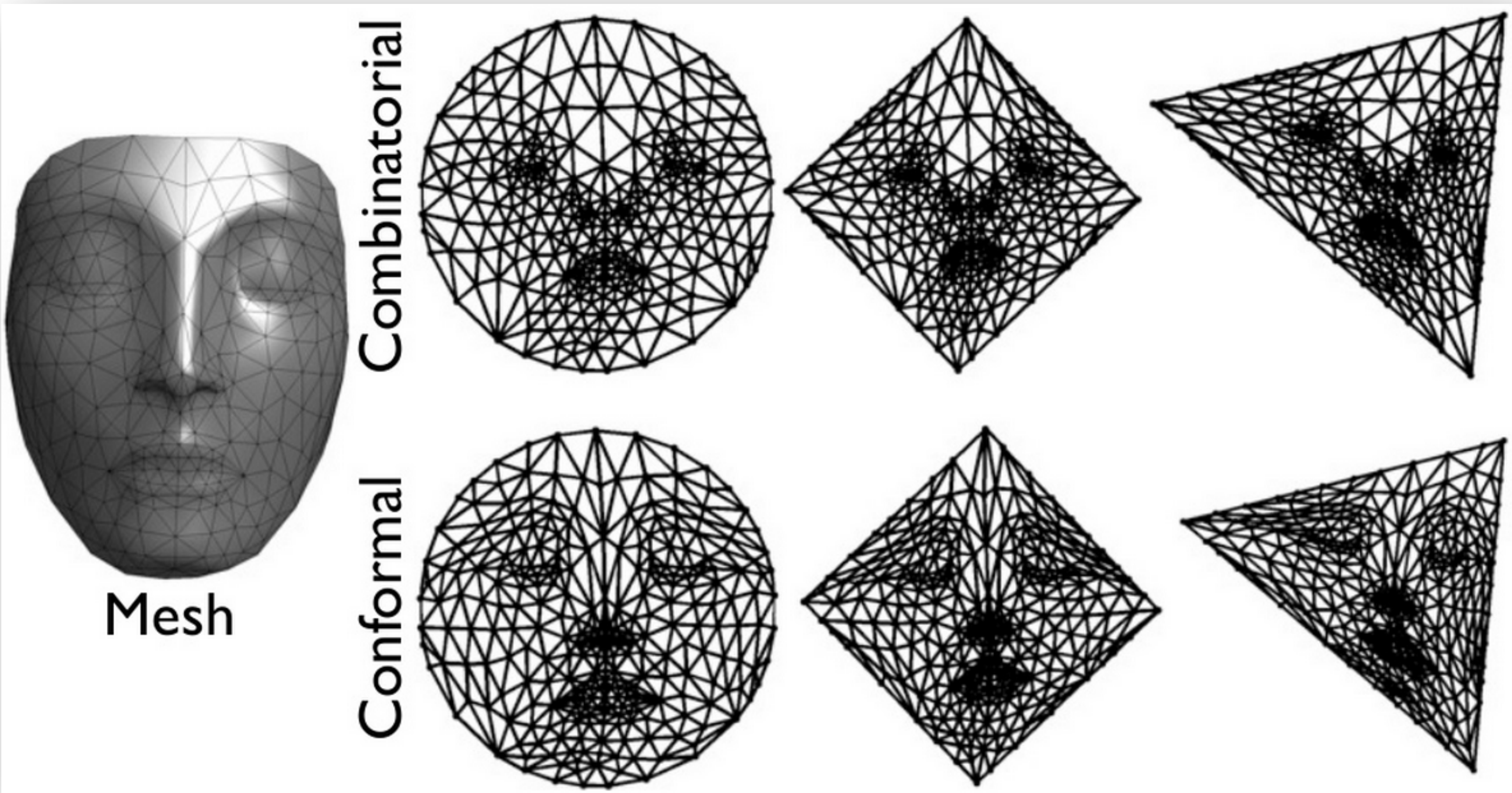
$$\to \min_x \frac{1}{2}x^\top A^\top Ax - b^\top Ax + \|b\|_2^2$$

$$\implies A^\top Ax = A^\top b$$

Normal equations

(better solvers for this case!)

Example: Mesh Embedding



G. Peyré, mesh processing course slides

Linear Solve for Embedding

$$\begin{aligned} \min_{x_1,\ldots,x_{|V|}} \quad & \sum_{(i,j)\in E} w_{ij} \|x_i - x_j\|_2^2 \\ \text{s.t.} \quad & x_v \text{ fixed } \forall v \in V_0 \end{aligned}$$

- $w_{ij} \equiv 1$: Tutte embedding
- w_{ij} **from mesh:** Harmonic embedding

Assumption: symmetric.

Returning to Parameterization

$$\begin{array}{rl} \min_{x_1,\ldots,x_{|V|}} & \sum_{(i,j)\in E} w_{ij}\|x_i - x_j\|_2^2 \\ \text{s.t.} & x_v \text{ fixed } \forall v \in V_0 \end{array}$$

What if

$V_0 = \{\}$?

Nontriviality Constraint

$$\left\{ \begin{array}{rl} \min_x & \|Ax\|_2 \\ \text{s.t.} & \|x\|_2 = 1 \end{array} \right\} \mapsto A^\top Ax = \lambda x$$

Prevents trivial solution $x \equiv 0$.

Extract the smallest eigenvalue.

Basic Idea of Eigenalgorithms

$$
\begin{aligned}
A\vec{v} &= c_1 A\vec{x}_1 + \cdots + c_n A\vec{x}_n \\
&= c_1\lambda_1\vec{x}_1 + \cdots + c_n\lambda_n\vec{x}_n \text{ since } A\vec{x}_i = \lambda_i\vec{x}_i \\
&= \lambda_1 \left(c_1\vec{x}_1 + \frac{\lambda_2}{\lambda_1} c_2\vec{x}_2 + \cdots + \frac{\lambda_n}{\lambda_1} c_n\vec{x}_n \right) \\
A^2\vec{v} &= \lambda_1^2 \left(c_1\vec{x}_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^2 c_2\vec{x}_2 + \cdots + \left(\frac{\lambda_n}{\lambda_1}\right)^2 c_n\vec{x}_n \right) \\
&\vdots \\
A^k\vec{v} &= \lambda_1^k \left(c_1\vec{x}_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^k c_2\vec{x}_2 + \cdots + \left(\frac{\lambda_n}{\lambda_1}\right)^k c_n\vec{x}_n \right).
\end{aligned}
$$