$$\nabla f+\sum_k \lambda_k \nabla g_k$$

Linear and Variational Problems

Justin Solomon

MIT, Spring 2019

MIT EECS

Some Announcements

- May change rooms
  Stay tuned!
- Diagram in homework revised
- Reading assignment posted today

Motivation

Extremely debatable perspective!

Part I:

Linear algebra $\subseteq$ Geometry

"Geometry of flat spaces"

Part II:

Geometry $\subseteq$ Optimization

Quick intro to variational calculus

Motivation

Part I:

Linear algebra $\subseteq$ Geometry

"Geometry of flat spaces"

Plus:

Intro to terrible notation.

#thankseinstein

Review and Notation

$$\text{(Column) vector: } \mathbf{x} \in \mathbb{R}^n$$

$$\text{Matrix: } A \in \mathbb{R}^{k \times \ell}$$

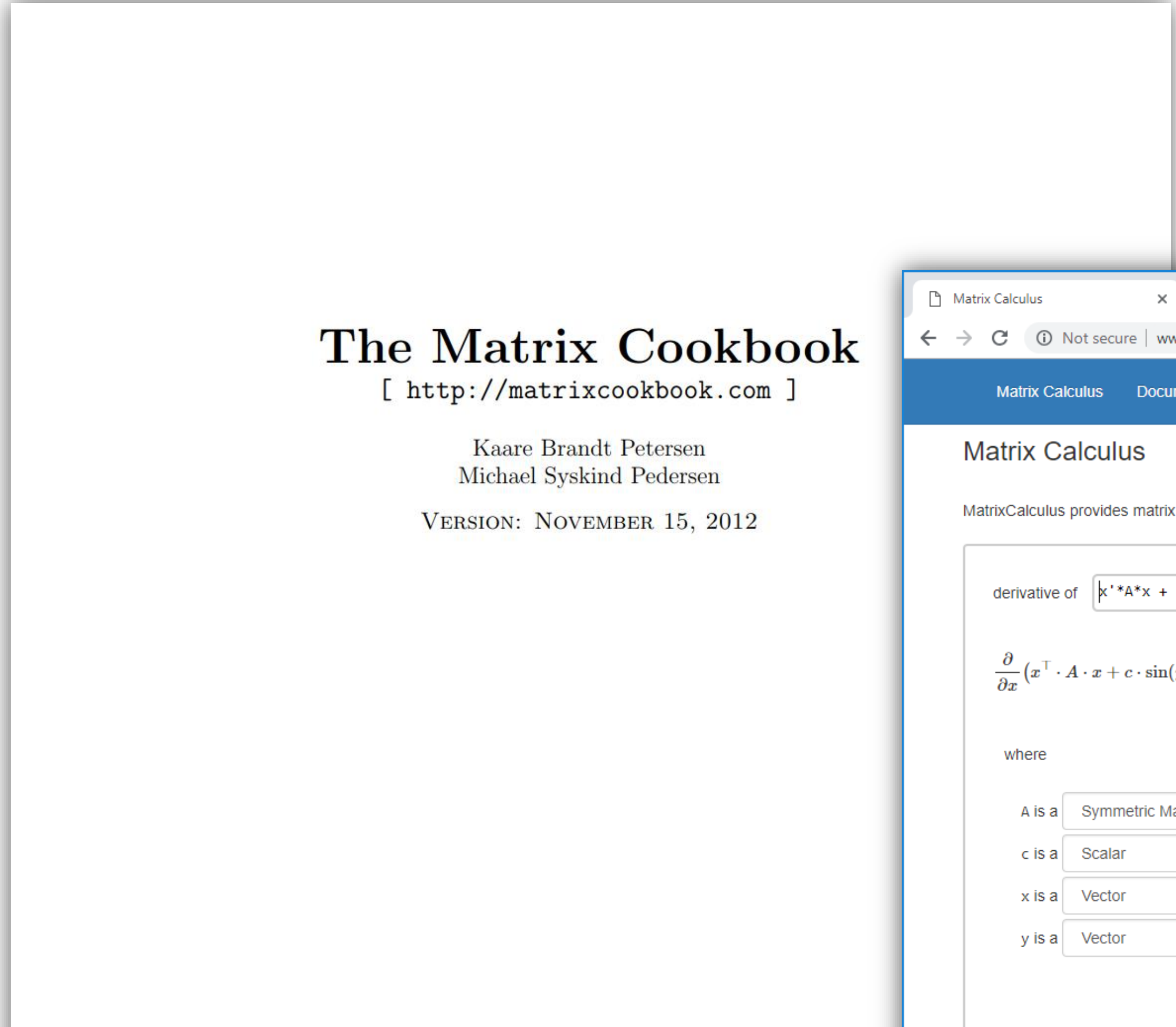
$$\text{Transpose: } \mathbf{x}^\top \in \mathbb{R}^{1 \times n}, A^\top \in \mathbb{R}^{\ell \times k}$$

Useful shorthand:

$$\text{Dot product: } \mathbf{x}^\top \mathbf{y}$$

$$\text{Quadratic form: } \mathbf{x}^\top A \mathbf{y}$$

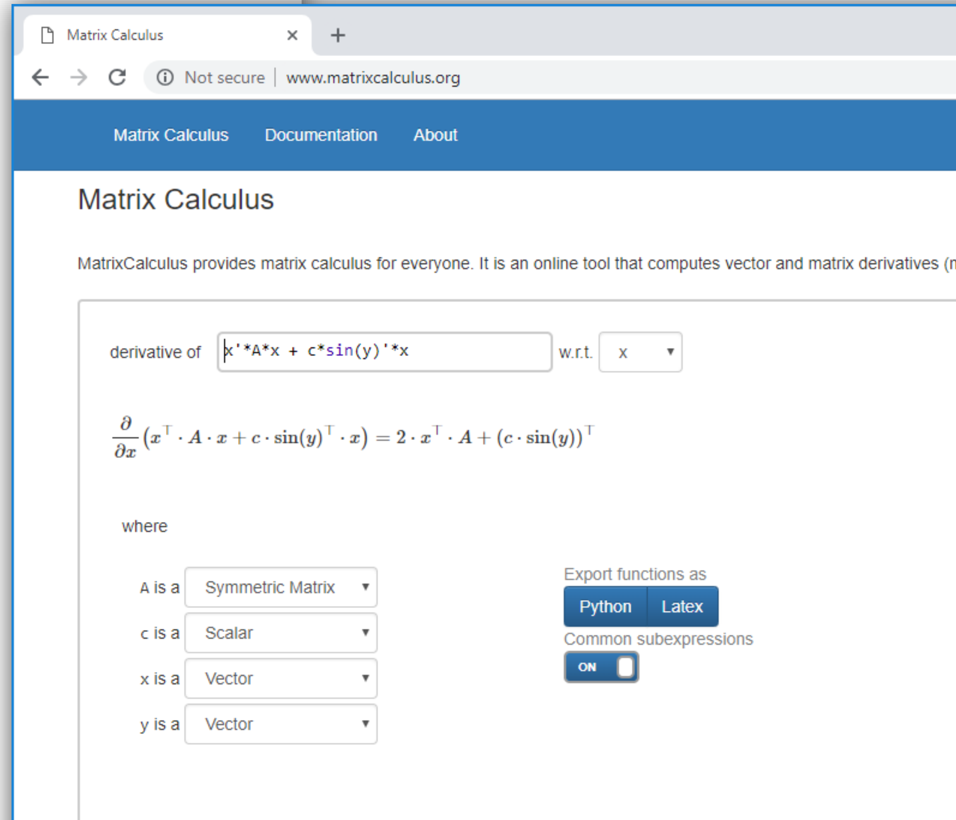
Aside: Matrix Calculus



The Matrix Cookbook
[http://matrixcookbook.com]

Kaare Brandt Petersen
Michael Syskind Pedersen

Version: November 15, 2012



Matrix Calculus

MatrixCalculus provides matrix calculus for everyone. It is an online tool that computes vector and matrix derivatives (m

derivative of `x'*A*x + c*sin(y)'*x` w.r.t. x

$$\frac{\partial}{\partial x}\left(x^\top \cdot A \cdot x + c \cdot \sin(y)^\top \cdot x\right) = 2 \cdot x^\top \cdot A + (c \cdot \sin(y))^\top$$

where

A is a Symmetric Matrix
c is a Scalar
x is a Vector
y is a Vector

Export functions as Python Latex
Common subexpressions ON

Two Roles for Matrices

Linear operator (map):

$$L[\mathbf{x}+\mathbf{y}] = L[\mathbf{x}] + L[\mathbf{y}]$$
$$L[c\mathbf{x}] = cL[\mathbf{x}]$$

$$L[\mathbf{x}] = A\mathbf{x}$$

Quadratic form (dot product):

$$g(\mathbf{u},\mathbf{v}) = g(\mathbf{v},\mathbf{u})$$
$$g(a\mathbf{u},\mathbf{v}) = ag(\mathbf{u},\mathbf{v})$$
$$g(\mathbf{u}+\mathbf{v},\mathbf{w}) = g(\mathbf{u},\mathbf{w}) + g(\mathbf{v},\mathbf{w})$$
$$g(\mathbf{u},\mathbf{u}) \geq 0$$

$$g(\mathbf{u},\mathbf{v}) = \mathbf{u}^\top B\mathbf{v}$$

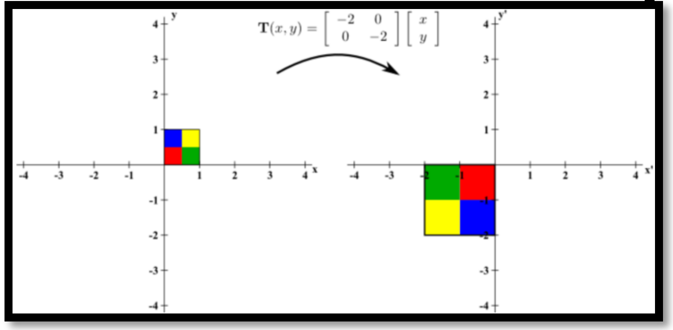
Q:

Are these the interchangeable?

$$L[\mathbf{x}] = A\mathbf{x} \qquad g(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\top B\mathbf{v}$$

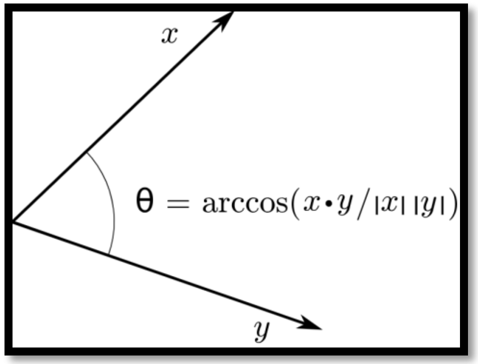
Same Data Structure, Two Uses

- Map between vector spaces


$\mathbf{T}(x,y) = \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$
x
y
x'
y'

$$L[\mathbf{x}] = A\mathbf{x}$$

- Inner product



$$g(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\top B \mathbf{v}$$

https://mathinsight.org/image/linear_transformation_2d_m2_0_0_m2

Protip:
Know your input and output

Matrices obscure geometry

Some More Notation

$$\mathbf{v} \text{ ``}=\text{''} \begin{pmatrix} v^1 \\ \vdots \\ v^n \end{pmatrix}$$

Standard basis: $\{\mathbf{e}_k\}_{k=1}^{n}$

$$\implies \mathbf{v} = \sum_k v^k \mathbf{e}_k$$

Einstein Notation

$$\mathbf{v} = v^k \mathbf{e}_k$$

🤢

Sum repeated upper/lower indices

Linear Map

$$\begin{pmatrix} L^1_1 & L^1_2 & \cdots & L^1_n \\ L^2_1 & L^2_2 & \cdots & L^2_n \\ \vdots & \vdots & \ddots & \vdots \\ L^m_1 & L^m_2 & \cdots & L^m_n \end{pmatrix} \begin{pmatrix} v^1 \\ v^2 \\ \vdots \\ v^n \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^n L^1_k v^k \\ \sum_{k=1}^n L^2_k v^k \\ \vdots \\ \sum_{k=1}^n L^m_k v^k \end{pmatrix} := \begin{pmatrix} w^1 \\ w^2 \\ \vdots \\ w^m \end{pmatrix}$$

Quadratic Form

$$\begin{aligned} g(\mathbf{u}, \mathbf{v}) &= g(u^k \mathbf{e}_k, v^\ell \mathbf{e}_\ell) \\ &= u^k v^\ell g(\mathbf{e}_k, \mathbf{e}_\ell) \\ &= u^k v^\ell g_{k\ell} \end{aligned}$$

Typechecking

$$\begin{pmatrix} L^1_1 & L^1_2 & \cdots & L^1_n \\ L^2_1 & L^2_2 & \cdots & L^2_n \\ \vdots & \vdots & \ddots & \vdots \\ L^m_1 & L^m_2 & \cdots & L^m_n \end{pmatrix} \begin{pmatrix} v^1 \\ v^2 \\ \vdots \\ v^n \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^n L^1_k v^k \\ \sum_{k=1}^n L^2_k v^k \\ \vdots \\ \sum_{k=1}^n L^m_k v^k \end{pmatrix} := \begin{pmatrix} w^1 \\ w^2 \\ \vdots \\ w^m \end{pmatrix}$$

$$\begin{aligned} g(\mathbf{u}, \mathbf{v}) &= g(u^k \mathbf{e}_k, v^\ell \mathbf{e}_\ell) \\ &= u^k v^\ell g(\mathbf{e}_k, \mathbf{e}_\ell) \\ &= u^k v^\ell g_{k\ell} \end{aligned}$$

Upper/lower indices matter

To Ponder At Home

Describe in Einstein notation:

$$\min_{\mathbf{x}} \left[\frac{1}{2}\mathbf{x}^\top A\mathbf{x} - \mathbf{x}^\top \mathbf{b}\right] \longrightarrow A\mathbf{x} = \mathbf{b}$$

What's up with *A*?

New Terminology

$$\underbrace{A}_{\text{matrix}} \; \underbrace{\mathbf{x}}_{\text{vector}}$$

$$\underbrace{\mathbf{x} \mapsto A\mathbf{x}}_{\text{linear operator}}$$

Abstract Example: Linear Algebra

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

$$\mathcal{L}[f] := -d^2 f / dx^2$$

Eigenvectors?
[“Eigenfunctions!”]

Back to reality:

Linear System of Equations

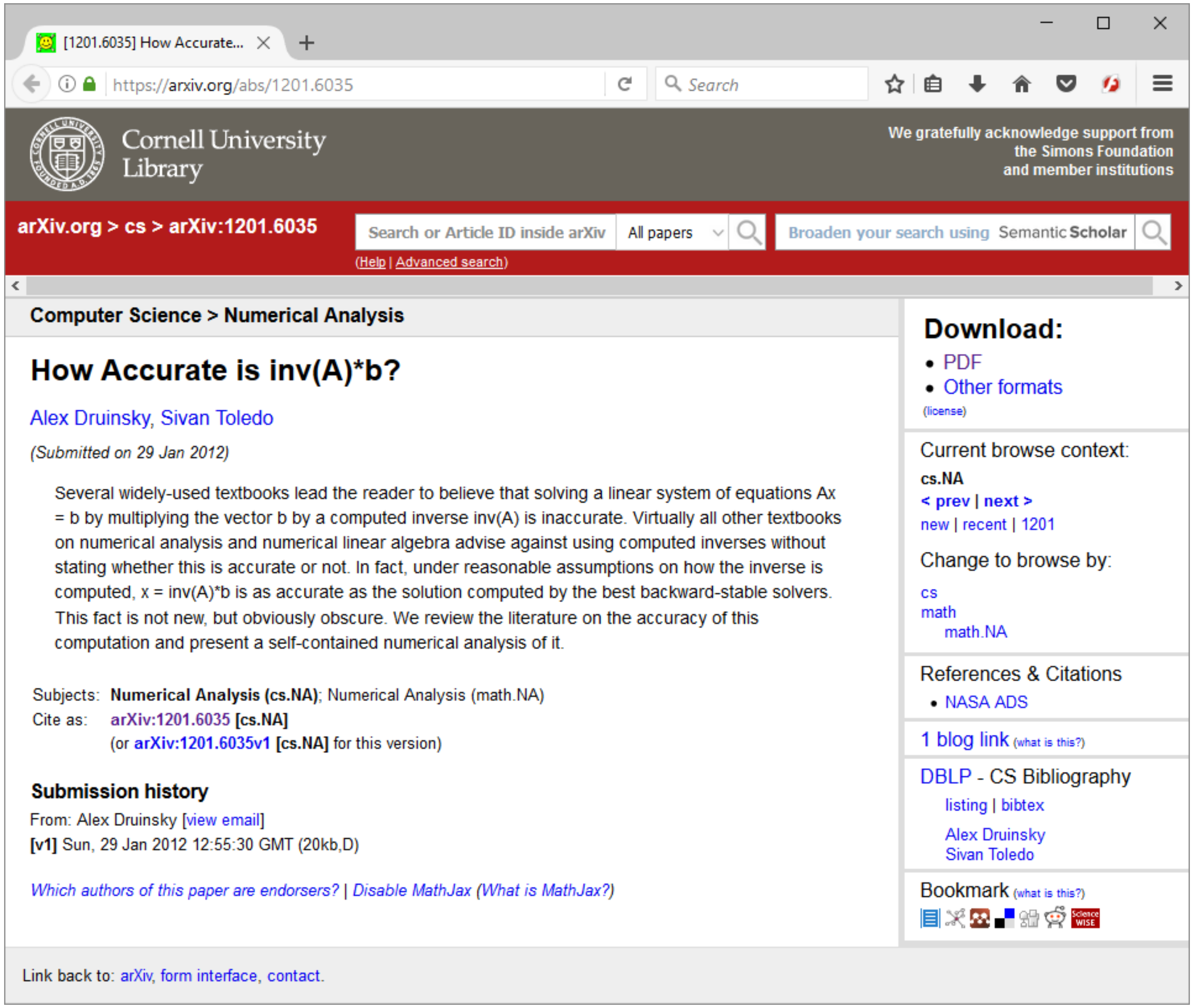
$$\begin{pmatrix} & & \\ & A & \\ & & \end{pmatrix} \begin{pmatrix} \\ \mathbf{x} \\ \\ \end{pmatrix} = \begin{pmatrix} \\ \mathbf{b} \\ \\ \end{pmatrix}$$

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

(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 A^{-1} explicitly**
  (if you can avoid it)
- **Added structure helps**
  Sparsity, symmetry, positive definiteness, bandedness

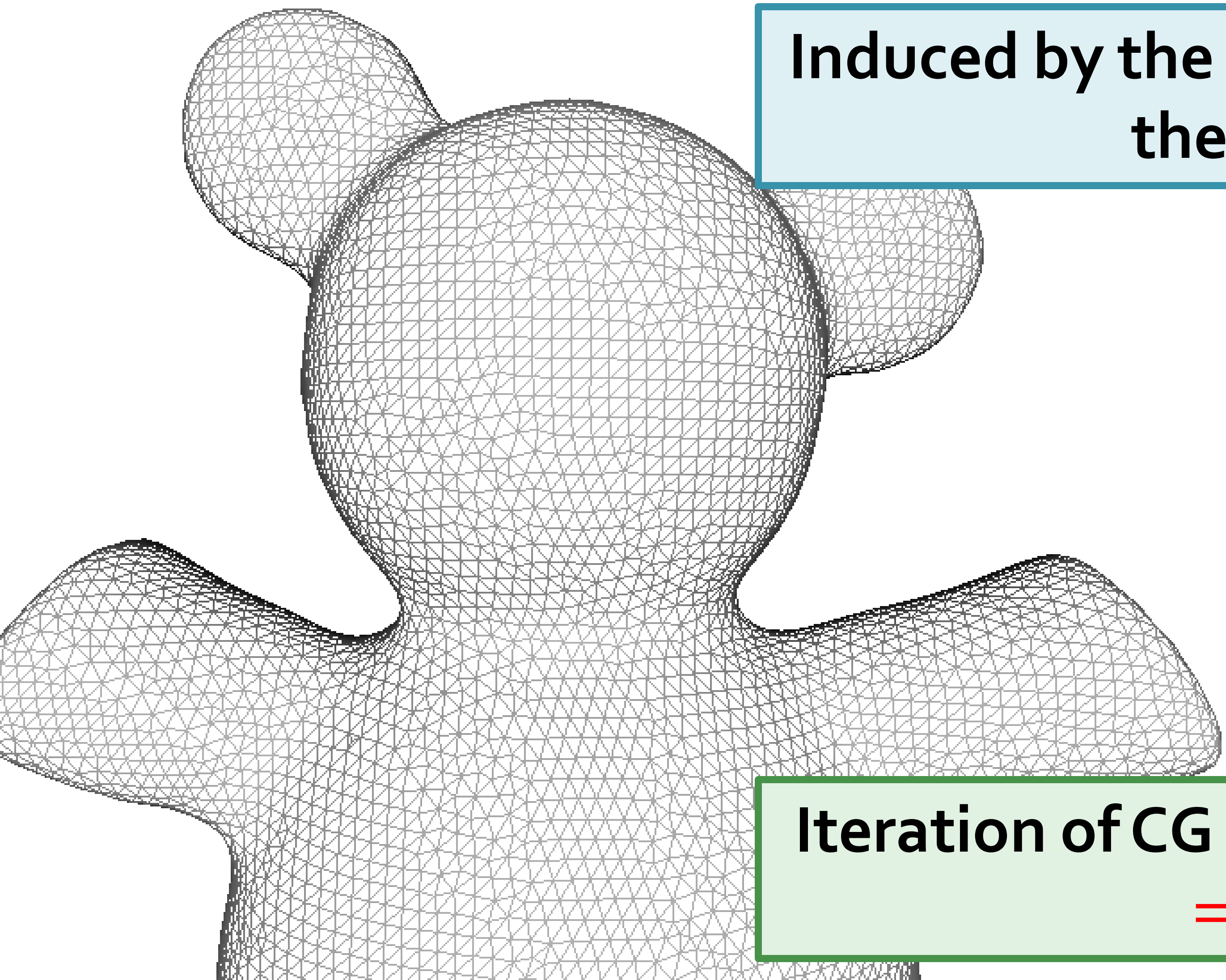
$$\texttt{inv(A)*b} \ll \texttt{(A'*A)}\backslash\texttt{(A'*b)} \ll \texttt{A}\backslash\texttt{b}$$

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

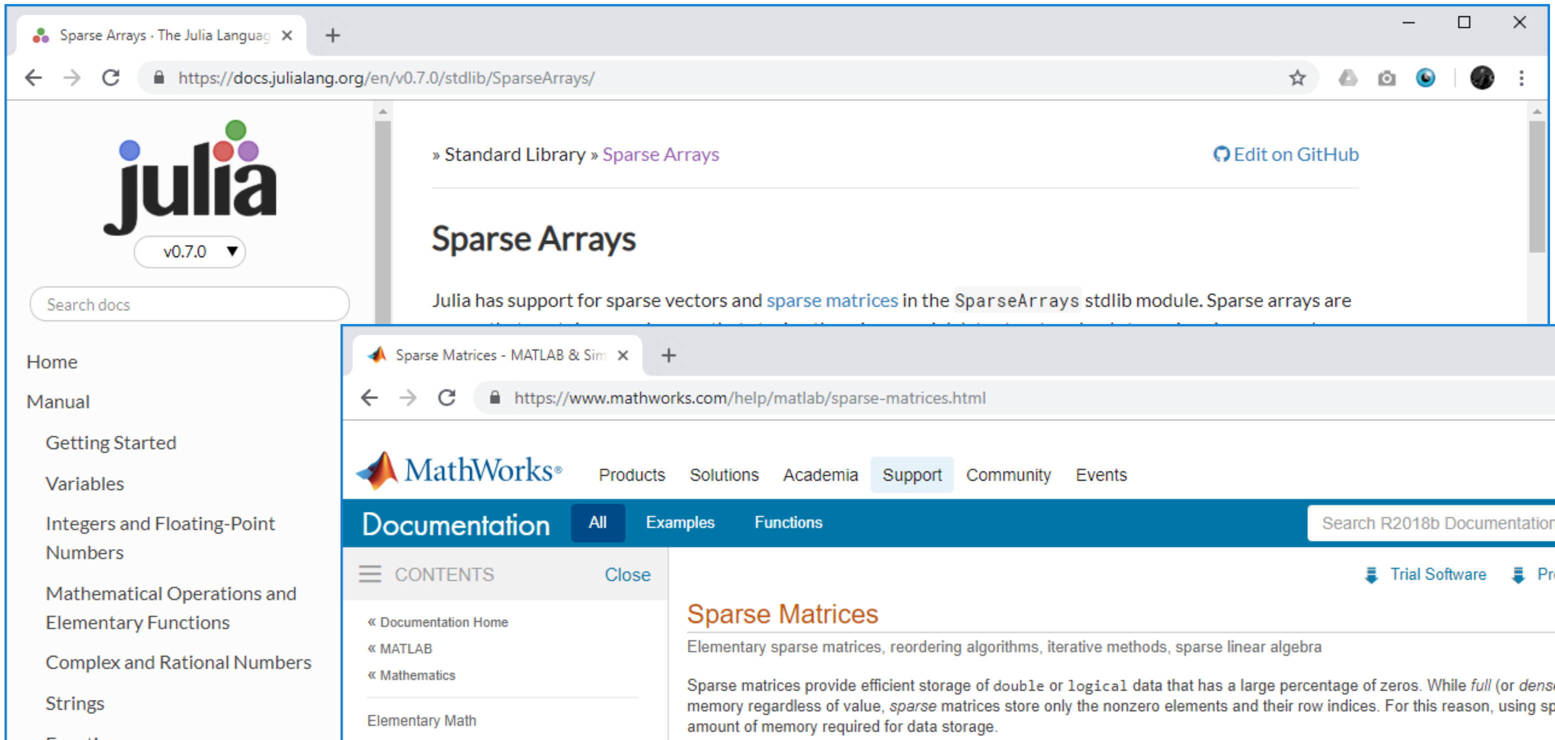


Induced by the connectivity of the triangle mesh.

Iteration of CG has local effect
$\Rightarrow$ Precondition!

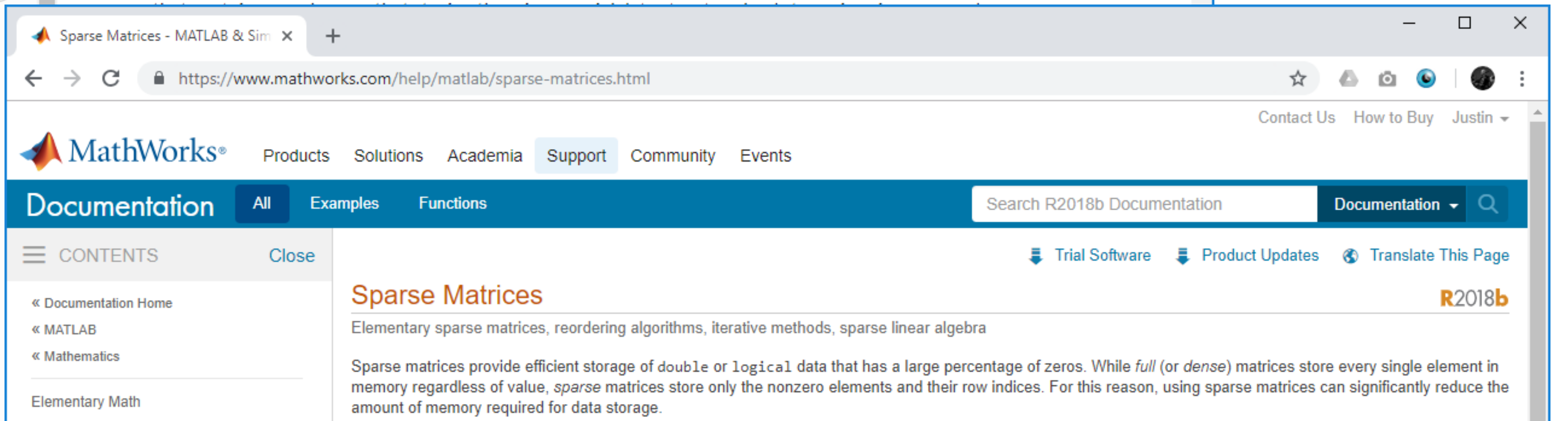
For 6.838

- **No need to implement** a linear solver
- If a matrix is sparse, your code should **store it as a sparse matrix!**


Sparse Arrays · The Julia Languag
https://docs.julialang.org/en/v0.7.0/stdlib/SparseArrays/
julia
v0.7.0
Search docs
Home
Manual
Getting Started
Variables
Integers and Floating-Point Numbers
Mathematical Operations and Elementary Functions
Complex and Rational Numbers
Strings
» Standard Library » Sparse Arrays
Edit on GitHub

Sparse Arrays

Julia has support for sparse vectors and sparse matrices in the `SparseArrays` stdlib module. Sparse arrays are



Sparse Matrices

Elementary sparse matrices, reordering algorithms, iterative methods, sparse linear algebra

Sparse matrices provide efficient storage of `double` or `logical` data that has a large percentage of zeros. While *full* (or *dense*) matrices store every single element in memory regardless of value, *sparse* matrices store only the nonzero elements and their row indices. For this reason, using sparse matrices can significantly reduce the amount of memory required for data storage.

Motivation

Part I:

Linear algebra $\subseteq$ Geometry

"Geometry of flat spaces"

Part II:

Geometry $\subseteq$ Optimization

Quick intro to variational calculus

Motivation

Part II:

Geometry $\subseteq$ Optimization

Quick intro to variational calculus

Optimization Terminology

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

Objective ("Energy Function")

Optimization Terminology

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

Equality Constraints

Optimization Terminology

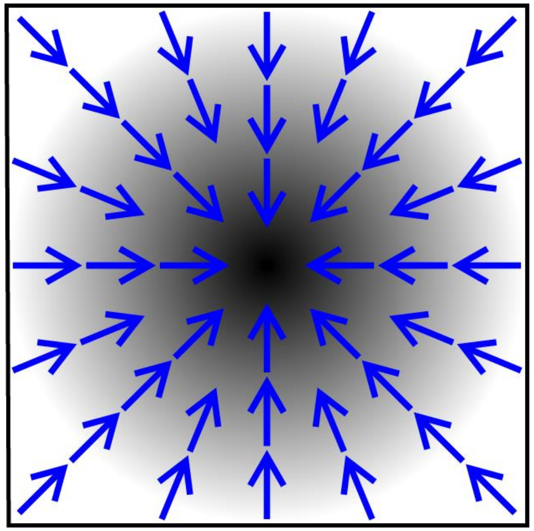
$$\begin{aligned} \min_{x \in \mathbb{R}^n} \; & f(x) \\ \text{s.t.}\; & g(x) = 0 \\ & h(x) \geq 0 \end{aligned}$$

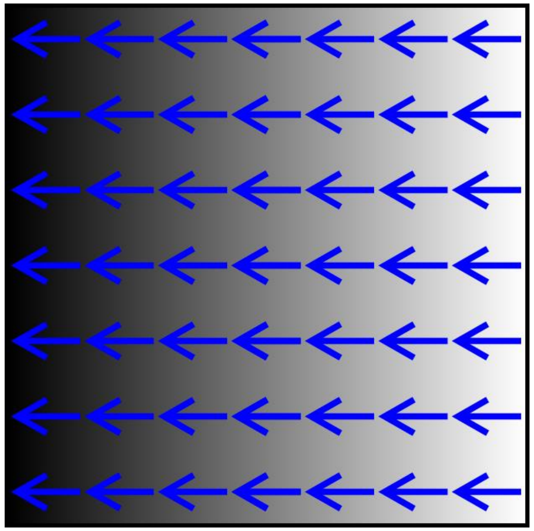
Inequality Constraints

Notions from Calculus

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

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





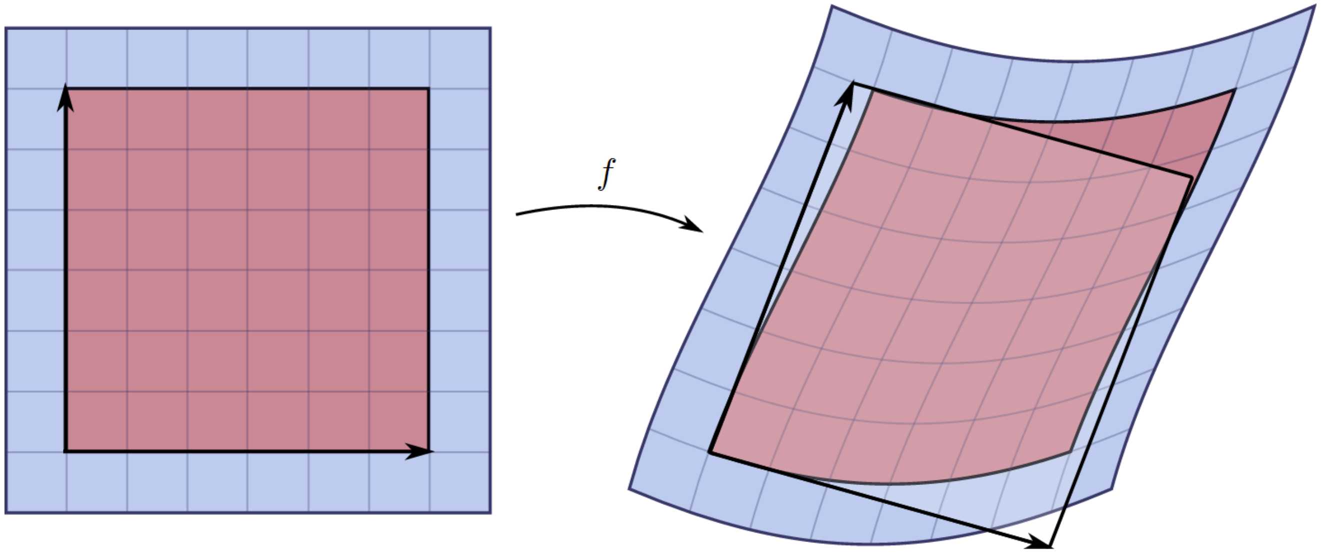
https://en.wikipedia.org/?title=Gradient

Gradient

Notions from Calculus

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

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

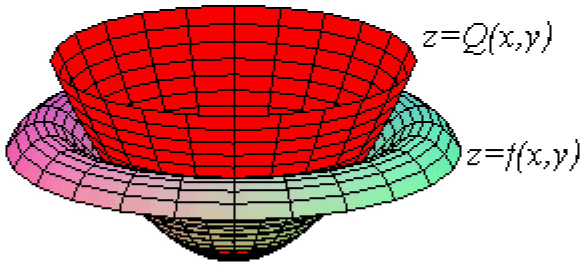


https://en.wikipedia.org/wiki/Jacobian_matrix_and_determinant

Jacobian

Notions from Calculus

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



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

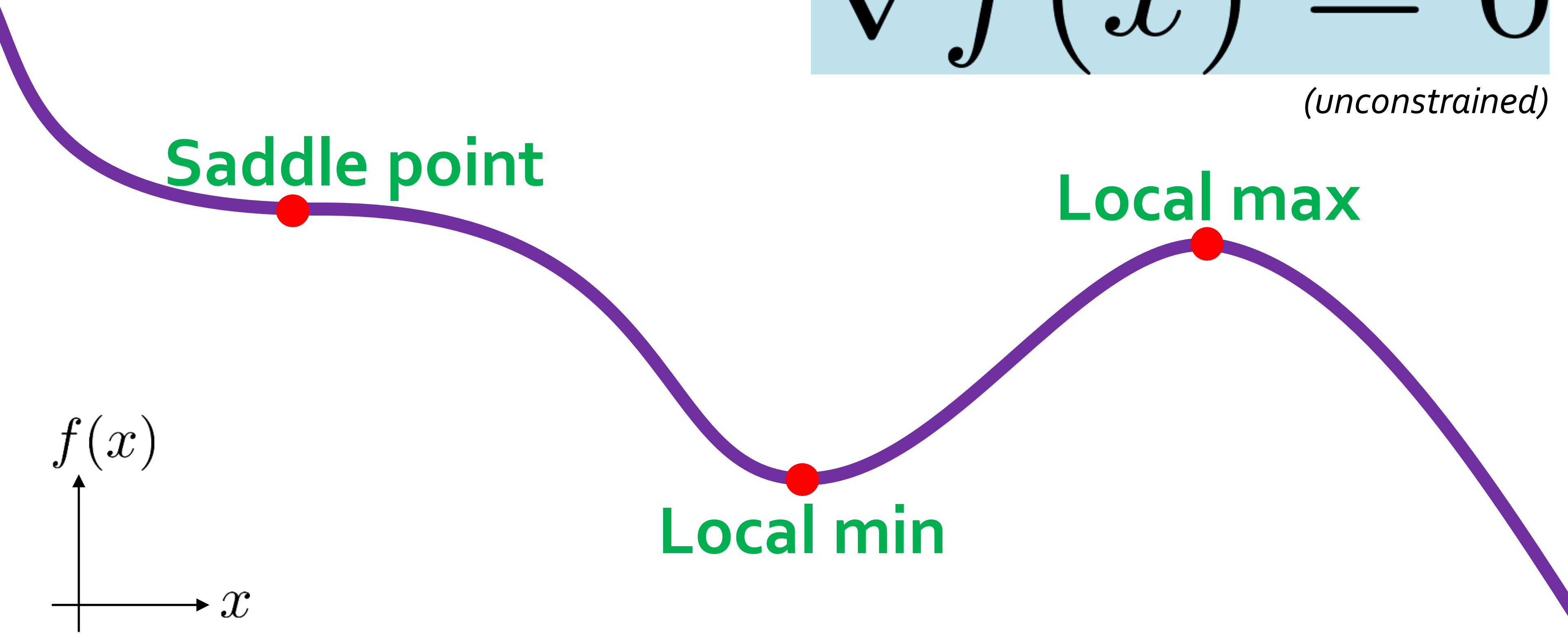
http://math.etsu.edu/multicalc/prealpha/Chap2/Chap2-5/10-3a-t3.gif

Hessian

Optimization to Root-Finding

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

(unconstrained)



Critical point

Encapsulates Many Problems

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

$$A\mathbf{x} = \mathbf{b} \leftrightarrow f(\mathbf{x}) = \|A\mathbf{x} - \mathbf{b}\|_2$$

$$A\mathbf{x} = \lambda\mathbf{x} \leftrightarrow f(\mathbf{x}) = \|A\mathbf{x}\|_2, g(\mathbf{x}) = \|\mathbf{x}\|_2 - 1$$

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

Q:

How effective are **generic** optimization tools?

Q:

How effective are **generic** optimization tools?

Not very!

Generic Advice

Try the

simplest method first.

Quadratic with Linear Equality

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

(assume A is symmetric and positive definite)

$$\downarrow$$

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

Special Case: Least-Squares

$$\min_{\mathbf{x}} \frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|_2^2$$

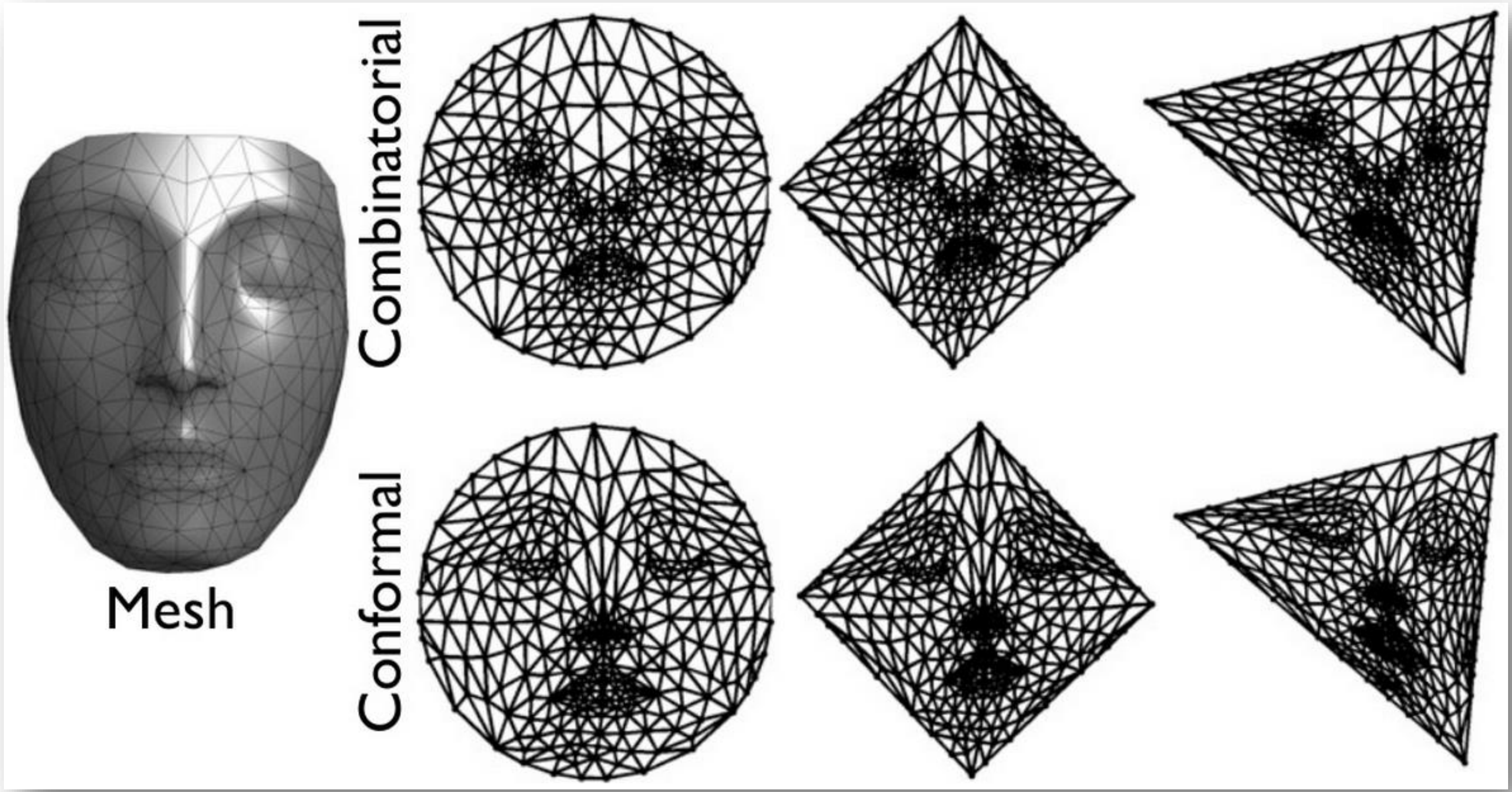
$$\to \min_{\mathbf{x}} \frac{1}{2}\mathbf{x}^\top A^\top A\mathbf{x} - \mathbf{b}^\top A\mathbf{x} + \|\mathbf{b}\|_2^2$$

$$\implies A^\top A\mathbf{x} = A^\top \mathbf{b}$$

Normal equations

(better solvers for this case!)

Example: Mesh Embedding



G. Peyré, mesh processing course slides

Linear Solve for Embedding

$$\begin{array}{rl}\min_{\mathbf{x}_1,\ldots,\mathbf{x}_{|V|}} & \sum_{(i,j)\in E} w_{ij}\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \\ \text{s.t.} & \mathbf{x}_v \text{ fixed } \forall v \in V_0\end{array}$$

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

Assumption: w symmetric.

Returning to Parameterization

$$\begin{array}{rl} \min_{\mathbf{x}_1,\ldots,\mathbf{x}_{|V|}} & \sum_{(i,j)\in E} w_{ij}\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \\ \text{s.t.} & \mathbf{x}_v \text{ fixed } \forall v \in V_0 \end{array}$$

What if $V_0 = \{\}$?

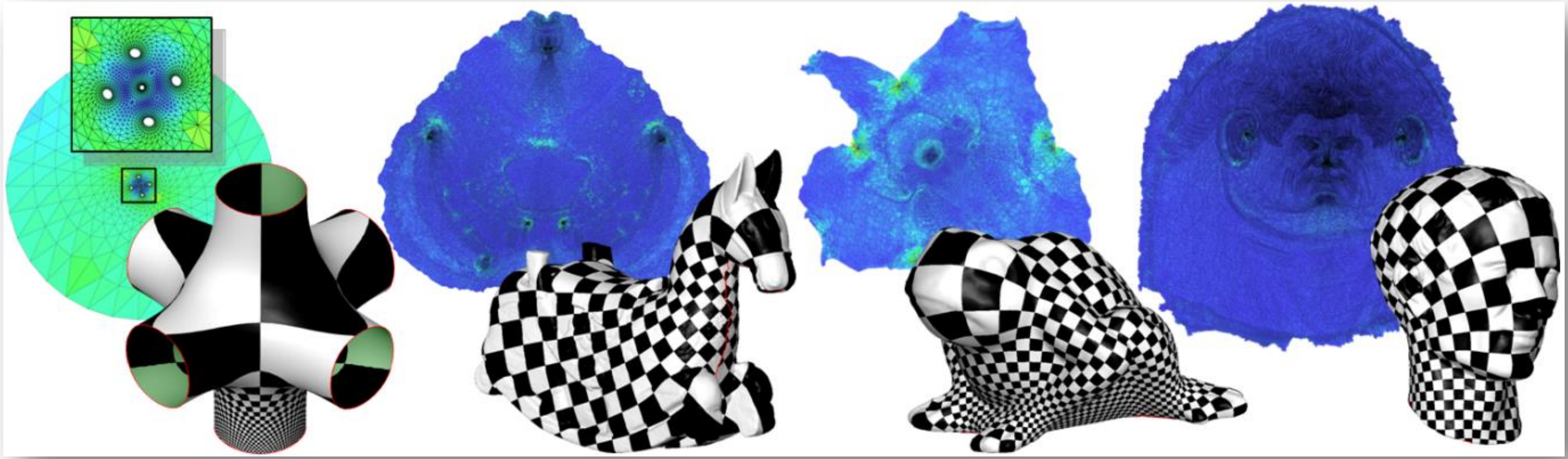
Nontriviality Constraint

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

Prevents trivial solution $x \equiv \mathbf{0}$.

Extract the smallest eigenvalue.

Back to Parameterization



Mullen et al. "Spectral Conformal Parameterization." SGP 2008.

$$\min_{\mathbf{u}} \quad \mathbf{u}^\top L_C \mathbf{u} \quad \longleftrightarrow \quad L_c\mathbf{u} = \lambda B\mathbf{u}$$

$\mathbf{u}^\top B\mathbf{e}=0$ ← *Easy fix*

$\mathbf{u}^\top B\mathbf{u}=1$

Combining Tools So Far

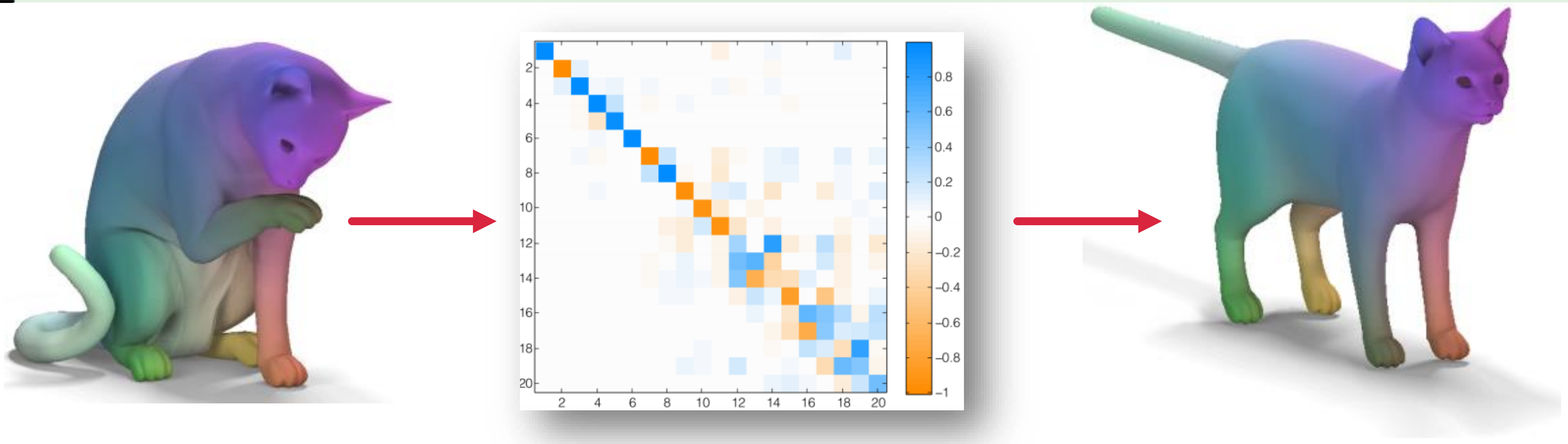
Roughly:

1. **Extract Laplace-Beltrami eigenfunctions:**

$$L\phi_i = \lambda_i A \phi_i$$

2. **Find mapping matrix (linear solve!):**

$$\min_{A \in \mathbb{R}^{n \times n}} \|AF_0 - F\|_{\mathrm{Fro}}^2 + \alpha \|A\Delta_0 - \Delta A\|_{\mathrm{Fro}}^2$$

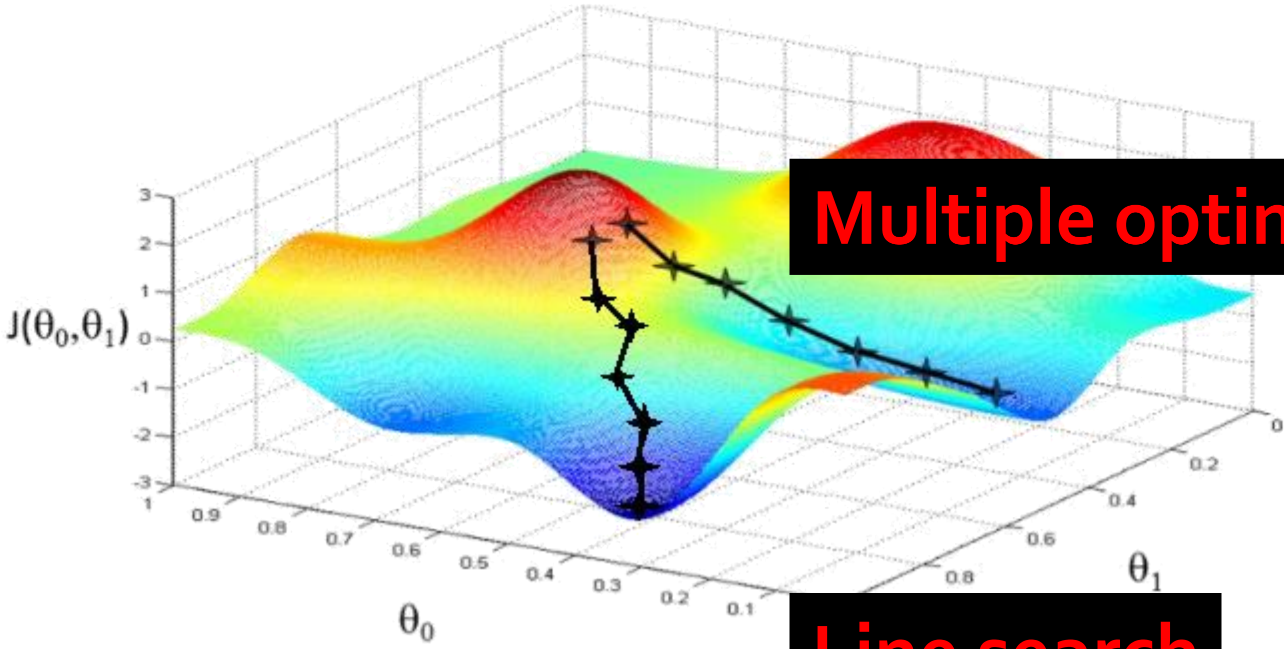


Ovsjanikov et al. "Functional Maps." SIGGRAPH 2012.

Unconstrained Optimization

$$\min_{x} \underset{\substack{\uparrow \\ \text{Unstructured.}}}{f}(x)$$

Basic Algorithms



Multiple optima!

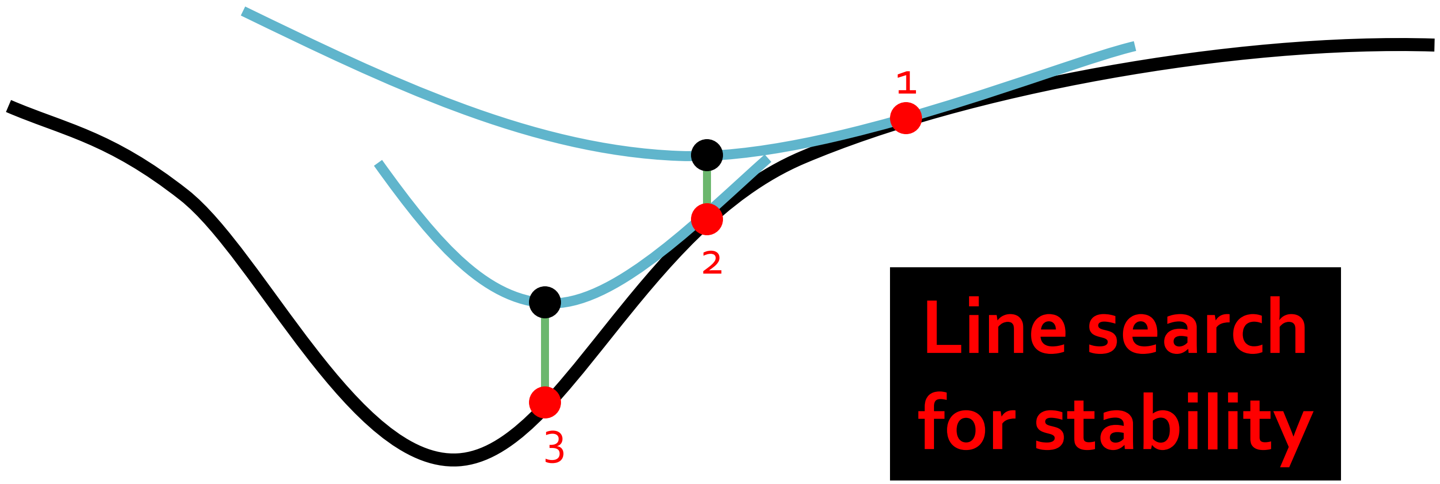
Line search

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

Gradient descent

Basic Algorithms

$$x_{k+1} = x_k - \left[Hf(x_k)\right]^{-1} \nabla f(x_k)$$



Line search for stability

Newton's Method

Basic Algorithms

$$x_{k+1} = x_k - \underbrace{M_k^{-1}}_{\text{Hessian approximation}} \nabla f(x_k)$$

- (Often sparse) approximation from previous samples and gradients
- Inverse in closed form!

Quasi-Newton: BFGS and friends

Example: Shape Interpolation

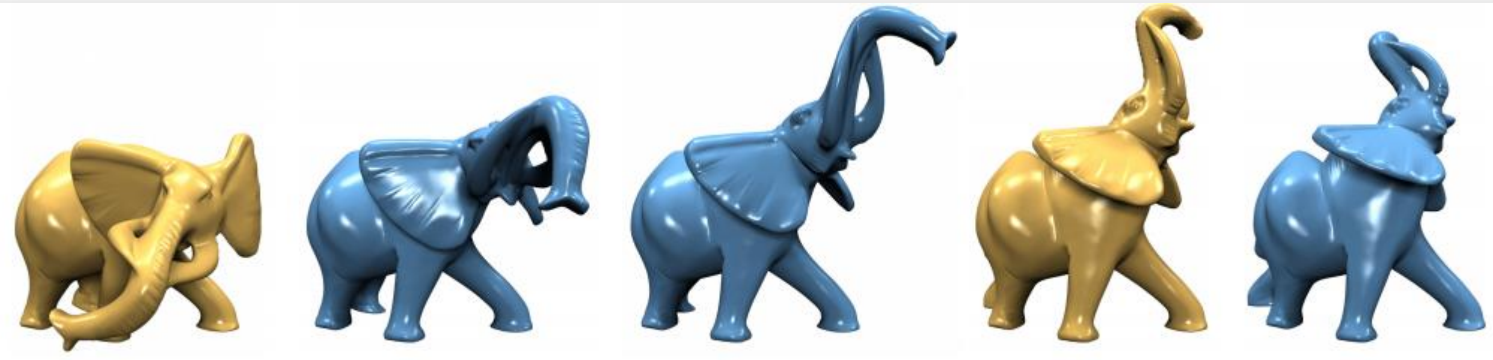


Figure 5: *Interpolation and extrapolation of the yellow example poses. The blending weights are* 0, 0.35, 0.65, 1.0, *and* 1.25.

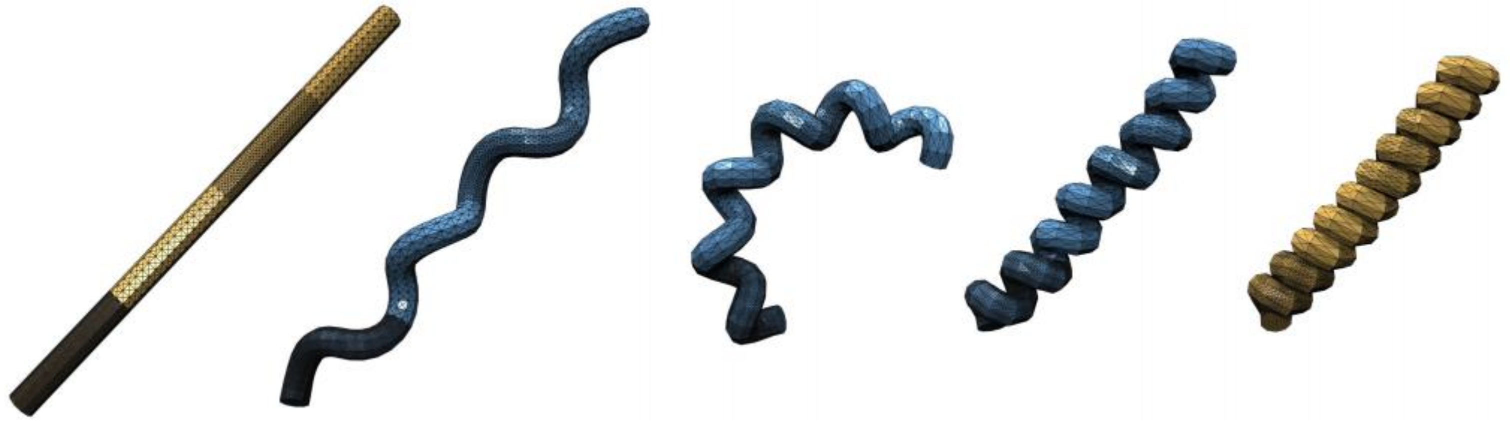


Figure 6: *Interpolation of an adaptively meshed and strongly twisted helix with blending weights* 0, 0.25, 0.5, 0.75, 1.0.

Fröhlich and Botsch. "Example-Driven Deformations Based on Discrete Shells." CGF 2011.

Interpolation Pipeline

Roughly:

1. **Linearly interpolate** edge lengths and dihedral angles.

$$\ell_e^* = (1-t)\ell_e^0 + t\ell_e^1$$

$$\theta_e^* = (1-t)\theta_e^0 + t\theta_e^1$$

2. **Nonlinear** optimization for vertex positions.

$$\min_{x_1,\ldots,x_m} \lambda \sum_e w_e(\ell_e(x) - \ell_e^*)^2 + \mu \sum_e w_b(\theta_e(x) - \theta_e^*)^2$$

Sum of squares: Gauss-Newton

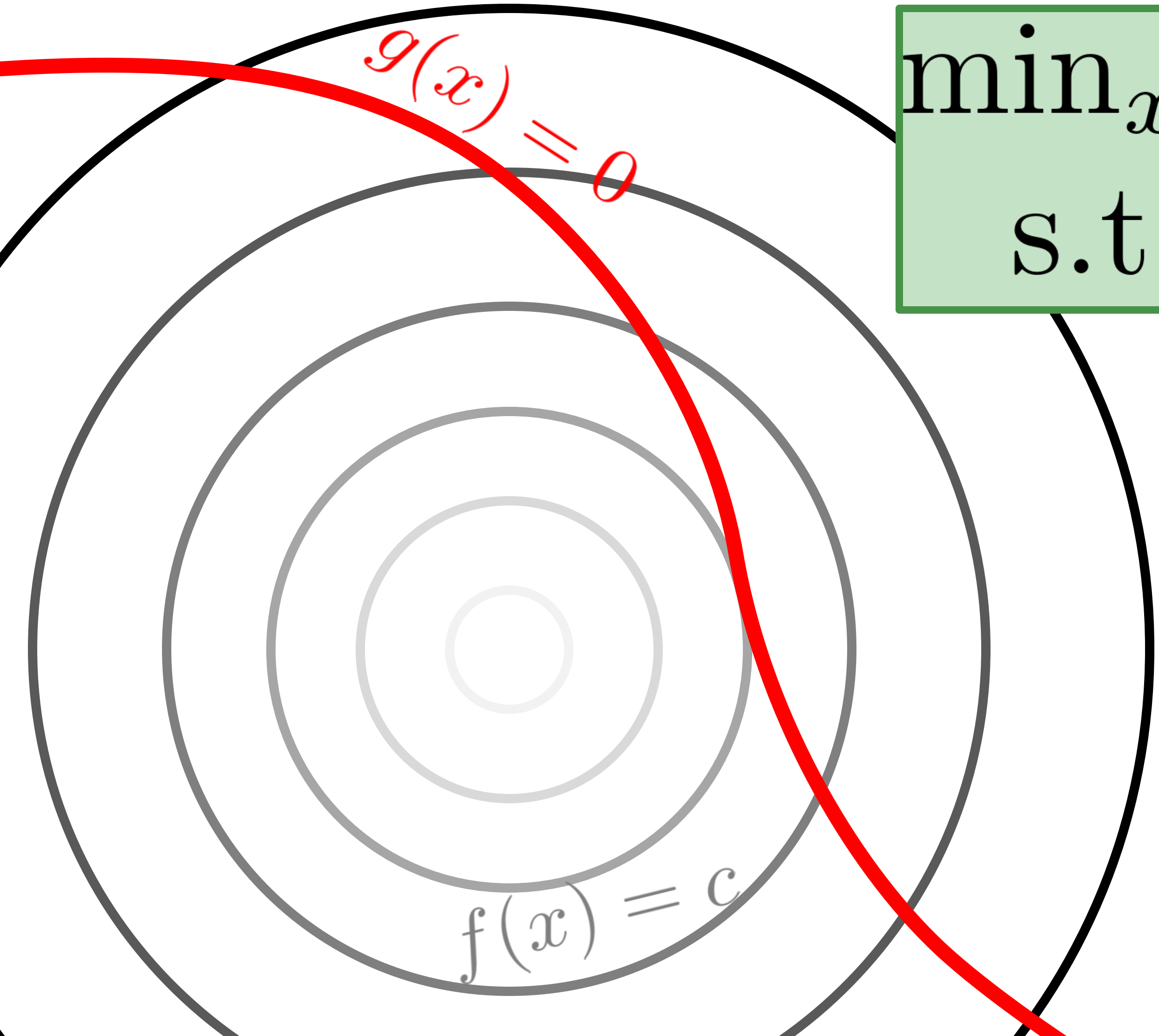
Software

- Matlab: `fminunc` or `minfunc`
- C++: `libLBFGS`, `dlib`, others

Typically provide functions for function and gradient (and optionally, Hessian).

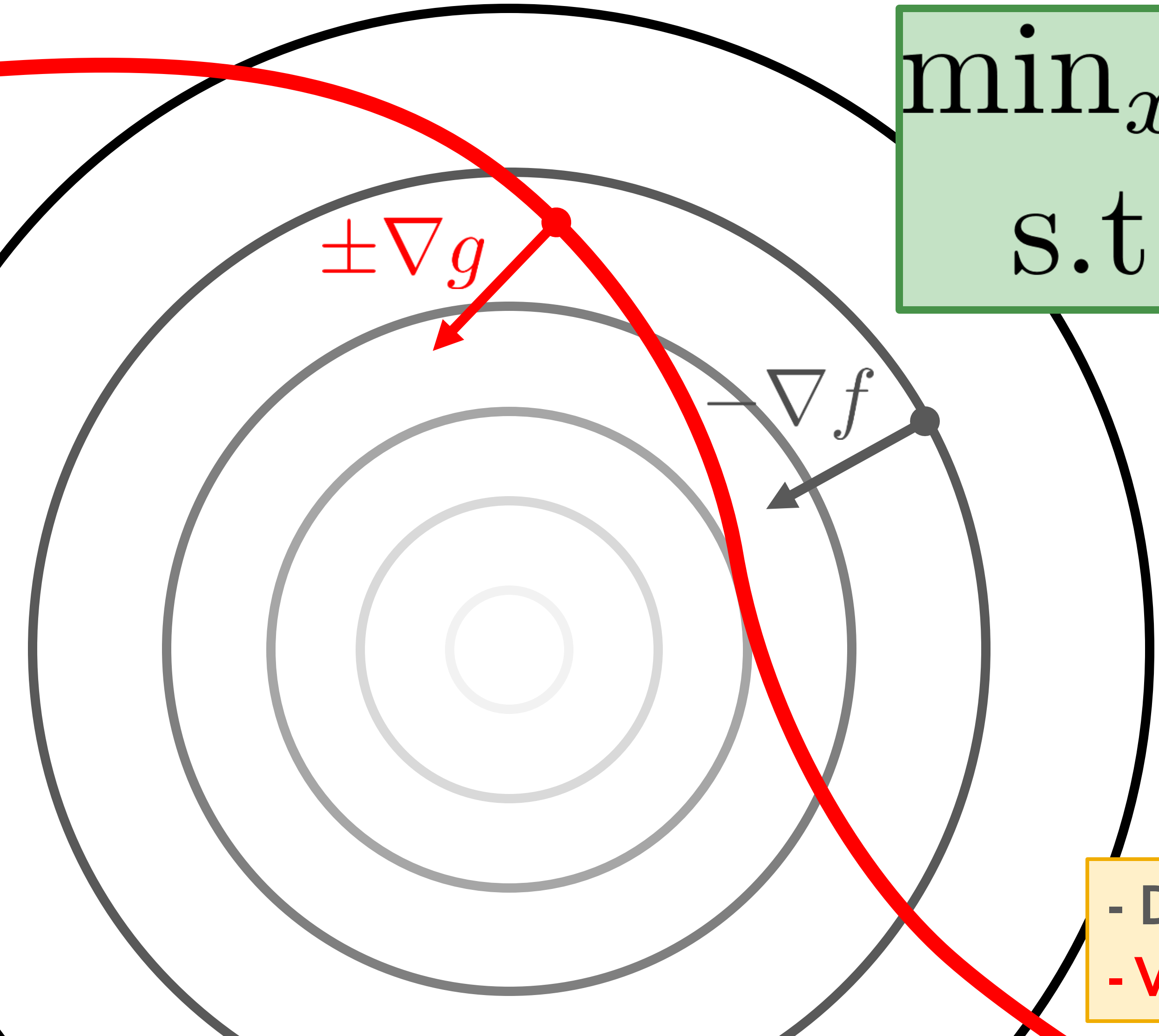
Try several!

Lagrange Multipliers: Idea



$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0 \end{aligned}$$

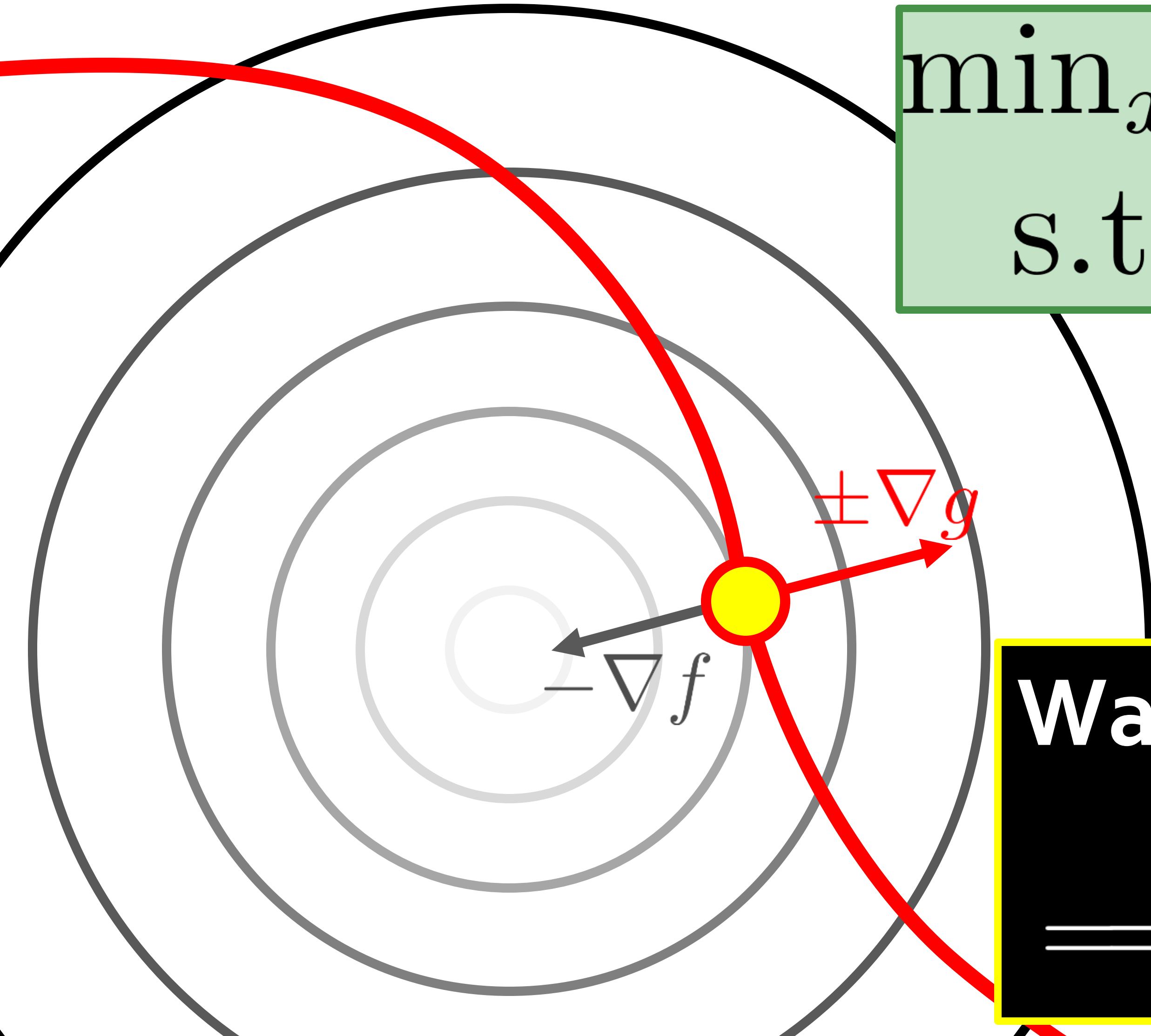
Lagrange Multipliers: Idea



$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0 \end{aligned}$$

- Decrease f: $-\nabla f$
- Violate constraint: $\pm\nabla g$

Lagrange Multipliers: Idea



$$\begin{aligned}\min_x \quad & f(x)\\ \text{s.t.} \quad & g(x)=0\end{aligned}$$

Want:

$$\nabla f \| \nabla g$$

$$\Longrightarrow \quad \nabla f = \lambda \nabla g$$

Example: Symmetric Eigenvectors

$$f(x) = x^\top A x \implies \nabla f(x) = 2Ax$$

$$g(x) = \|x\|_2^2 \implies \nabla g(x) = 2x$$

$$\implies Ax = \lambda x$$

Use of Lagrange Multipliers

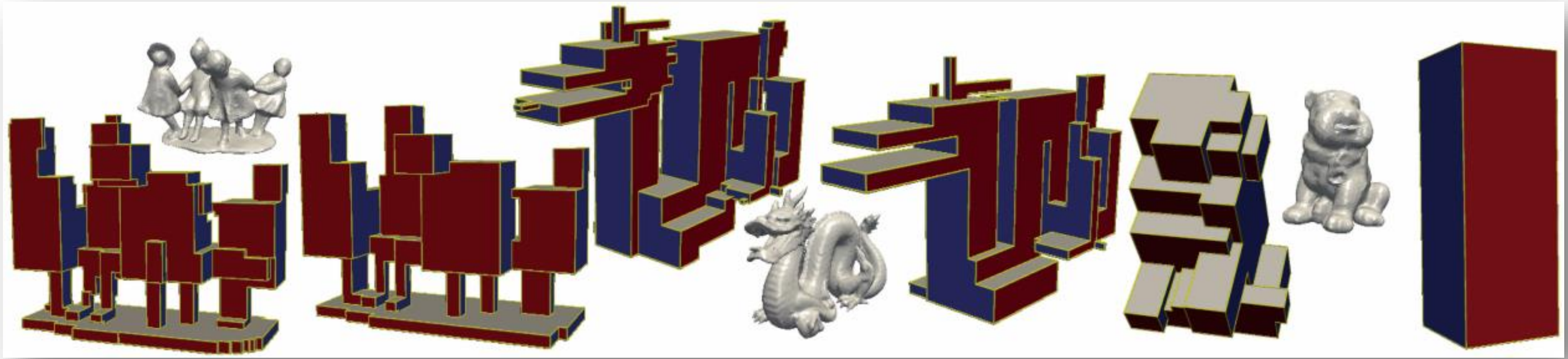
Turns constrained optimization into

unconstrained root-finding.

$$\nabla f(x) = \lambda \nabla g(x)$$

$$g(x) = 0$$

Example: Polycube Maps



Huang et al. "L1-Based Construction of Polycube Maps from Complex Shapes." TOG 2014.

Align with coordinate axes

$$\begin{aligned} \min_X \sum_{b_i} \quad & A(b_i;X)\|n(b_i;X)\|_1 \\ \text{s.t.} \quad & \sum_{b_i} A(b_i;X) = \sum_{b_i} A(b_i;X_0) \end{aligned}$$

Preserve area

Note: Final method includes more terms!

Variational Calculus: Big Idea

Sometimes your unknowns

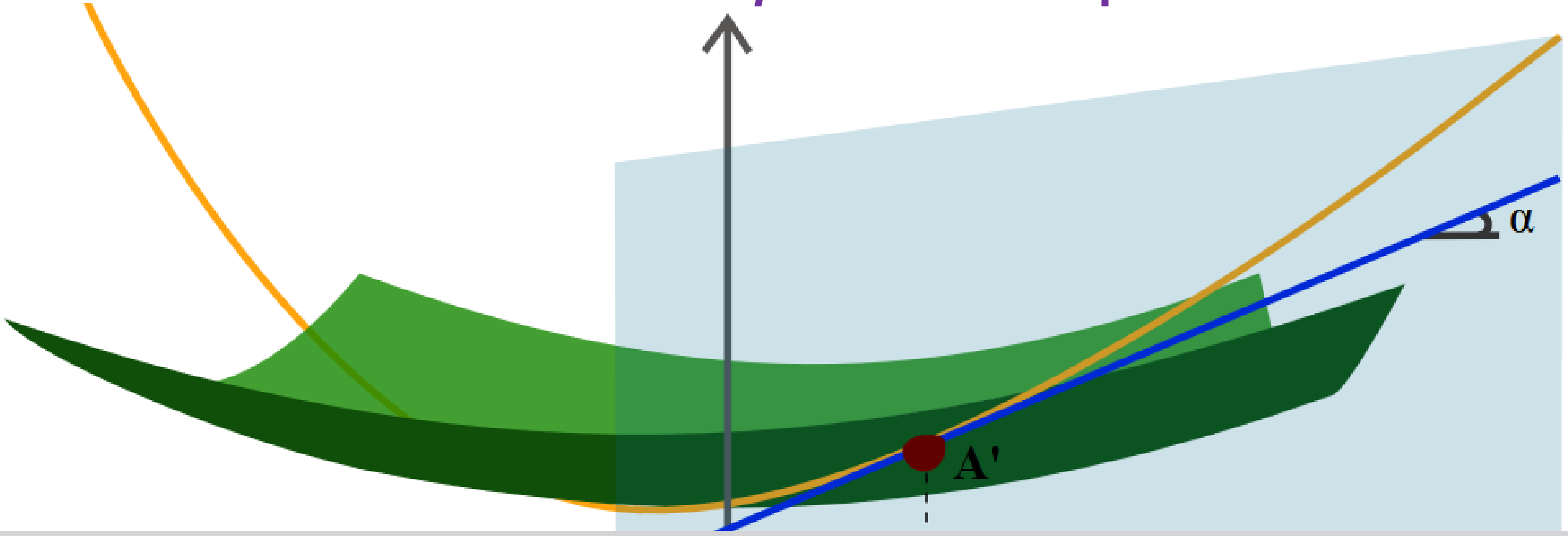
are not numbers!

Can we use calculus to optimize anyway?

Gâteaux Derivative

$$d\mathcal{F}[u;\psi] := \frac{d}{dh}\mathcal{F}[u+h\psi]|_{h=0}$$

Vanishes for all *ψ* at a critical point!



Analog of derivative at *u* in *ψ* direction

On the Board

$$\min_f \int_\Omega \|\mathbf{v}(\mathbf{x}) - \nabla f(\mathbf{x})\|_2^2 \, d\mathbf{x}$$

$$\min_{\int_\Omega f(\mathbf{x})^2 \, d\mathbf{x}=1} \int_\Omega \|\nabla f(\mathbf{x})\|_2^2 \, d\mathbf{x}$$

$$\nabla f + \sum_k \lambda_k \nabla g_k$$

Linear and Variational Problems

Justin Solomon

MIT, Spring 2019

MIT EECS