Linear and Variational Problems

Justin Solomon

6.8410: Shape Analysis
Spring 2023

Motivation

Extremely debatable perspective!

Part I:

Linear algebra ⊆ Geometry

"Geometry of flat spaces"

Part II:

Geometry ⊆ 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

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

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

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

Useful shorthand:

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

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

More Notation

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

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

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

Two Roles for Matrices in Finite-Dimensional Linear Algebra

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):

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

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

Einstein Notation

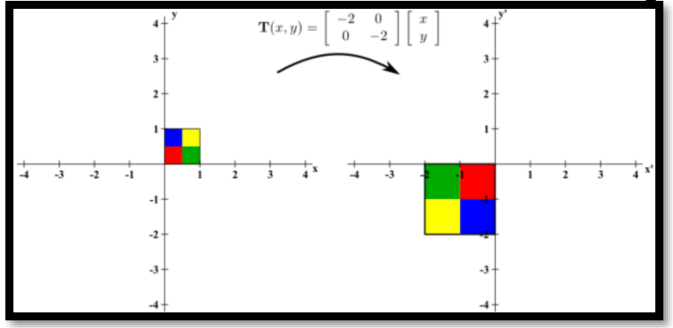
$$\mathbf{v} = v^k \mathbf{e}_k$$

Sum repeated upper/lower indices

Same Data Structure, Two Uses

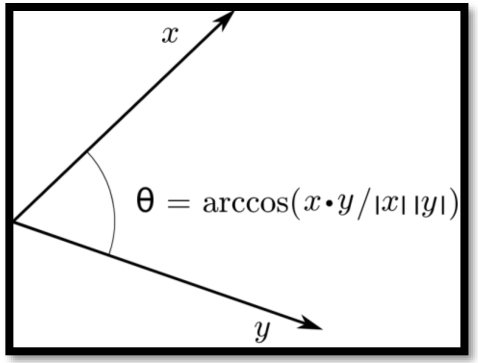
- **Map** between vector spaces

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



- 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

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

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

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.

Linear Solver Considerations

- **Never construct A^{-1} explicitly**
  (if you can avoid it)

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

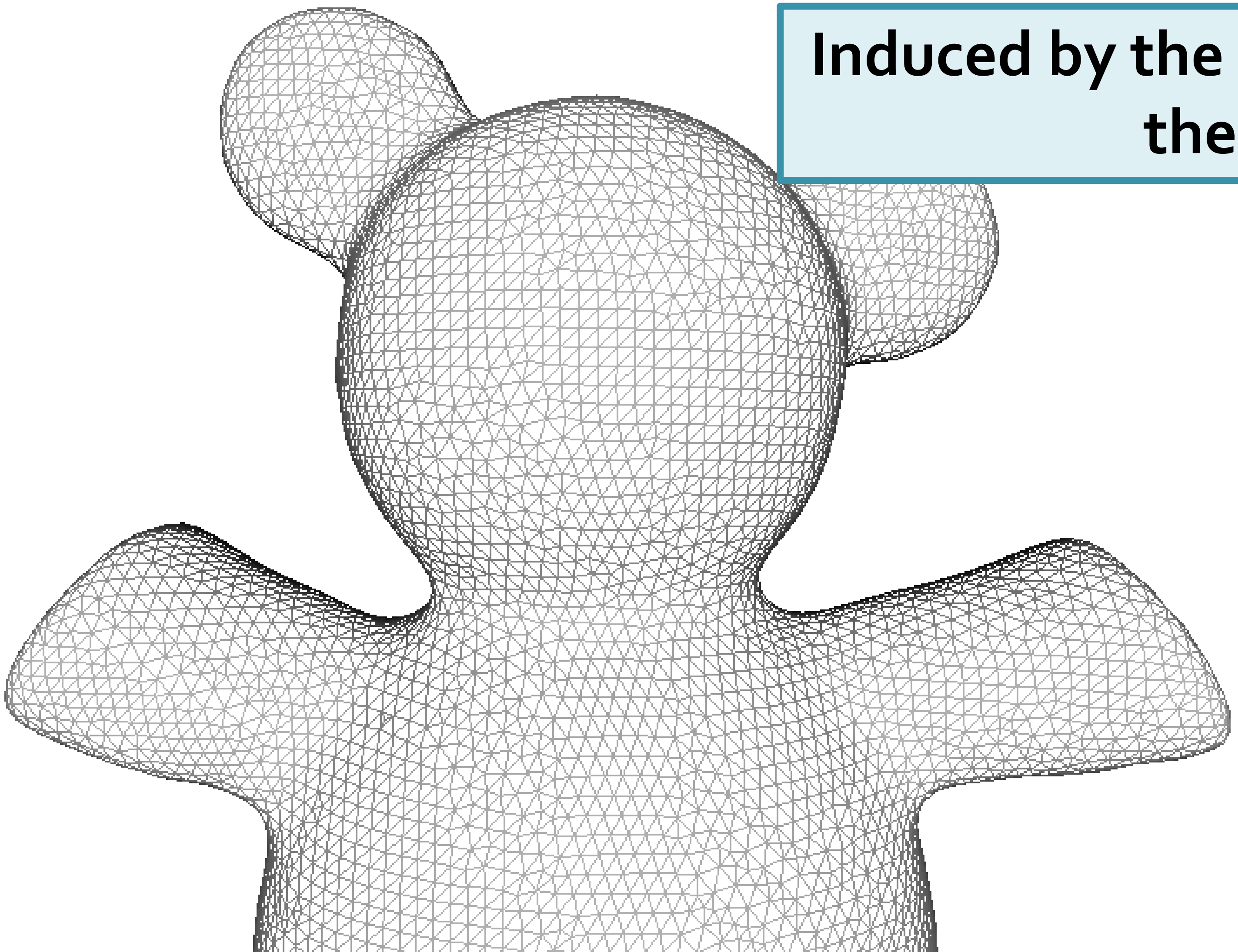
$$\texttt{inv(A)*b} \ll \texttt{(A'*A)\textbackslash(A'*b)} \ll \texttt{A\textbackslash b}$$

Example of a Structured Problem

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

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

Very Common: Sparsity



Induced by the connectivity of the triangle mesh.

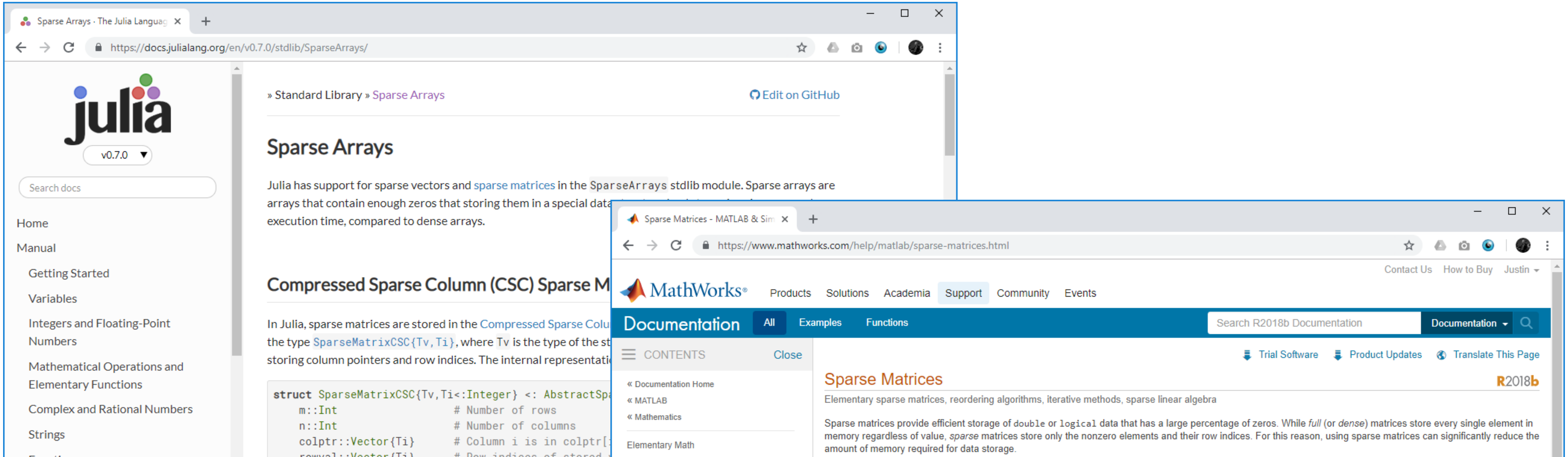
Two Classes of Solvers

- **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

For 6.8410

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



Motivation

Part I:

Linear algebra $\subseteq$ Geometry

"Geometry of flat spaces"

Part II:

Geometry $\subseteq$ Optimization

Quick intro to variational calculus

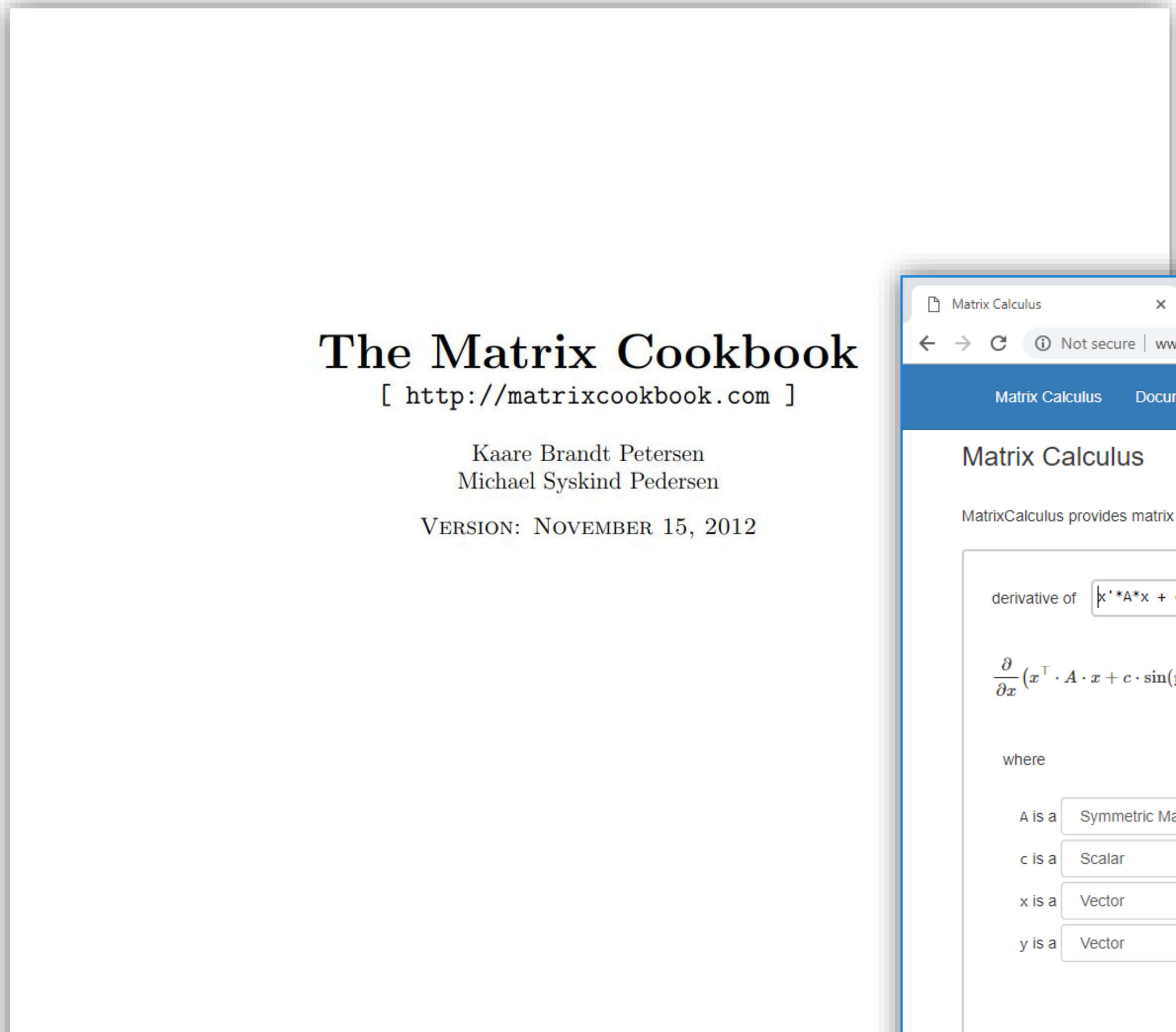
Motivation

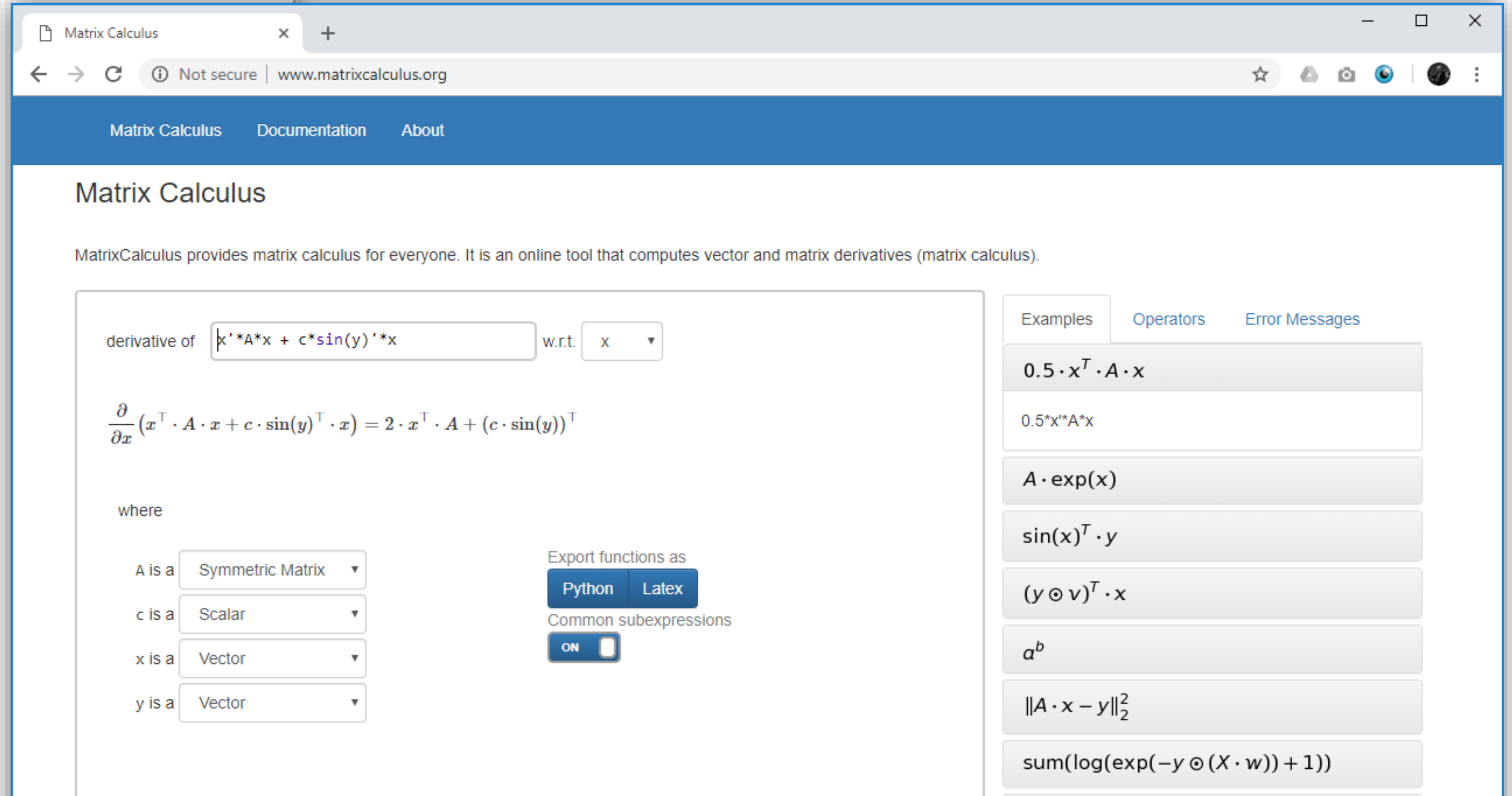
Part II:

Geometry $\subseteq$ Optimization

Quick intro to variational calculus

Aside: Matrix Calculus





Optimization Terminology

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

Objective ("Energy Function")

Optimization Terminology

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

Equality Constraints

Optimization Terminology

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

Inequality Constraints

Encapsulates Many Problems

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

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

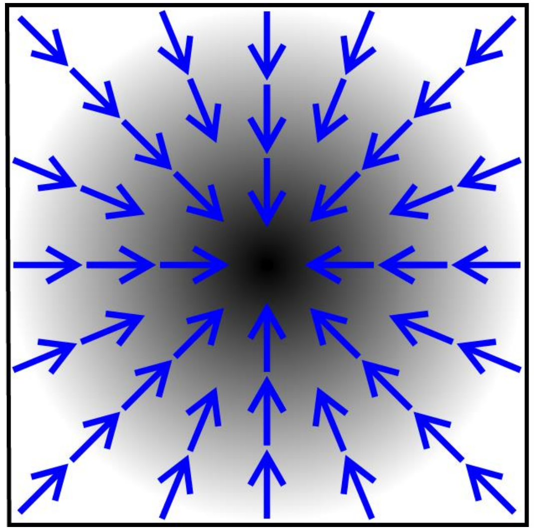
$$A\mathbf{x} = \lambda\mathbf{x} \leftrightarrow f(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x}, g(\mathbf{x}) = \|\mathbf{x}\|_2 - 1$$

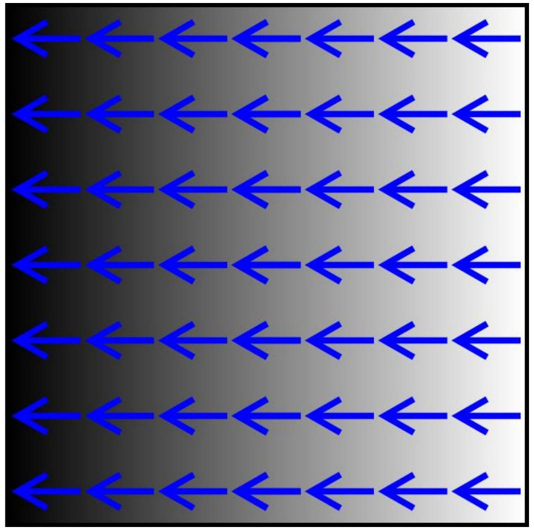
$$\text{Roots of } g(\mathbf{x}) \leftrightarrow f(\mathbf{x}) = 0$$

Notions from Calculus

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

$$\to \nabla f = \left(\frac{\partial f}{\partial x^1}, \frac{\partial f}{\partial x^2}, \ldots, \frac{\partial f}{\partial x^n} \right)$$





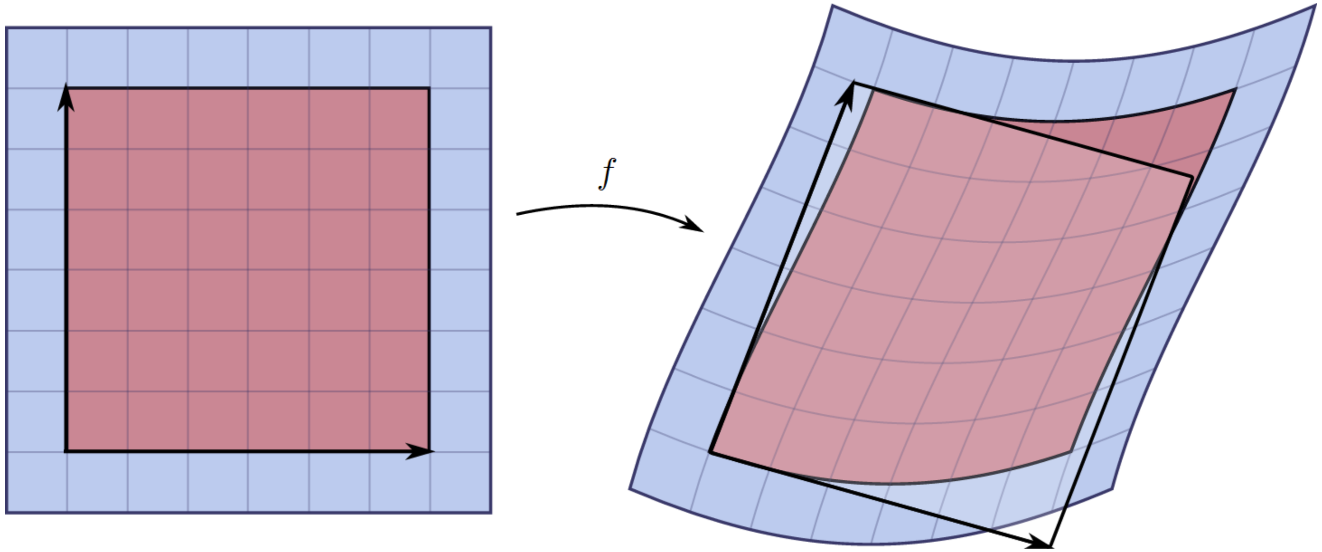
https://en.wikipedia.org/?title=Gradient

Gradient

Notions from Calculus

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

$$\to (Df)^i_j = \frac{\partial f^i}{\partial x^j}$$



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

Jacobian

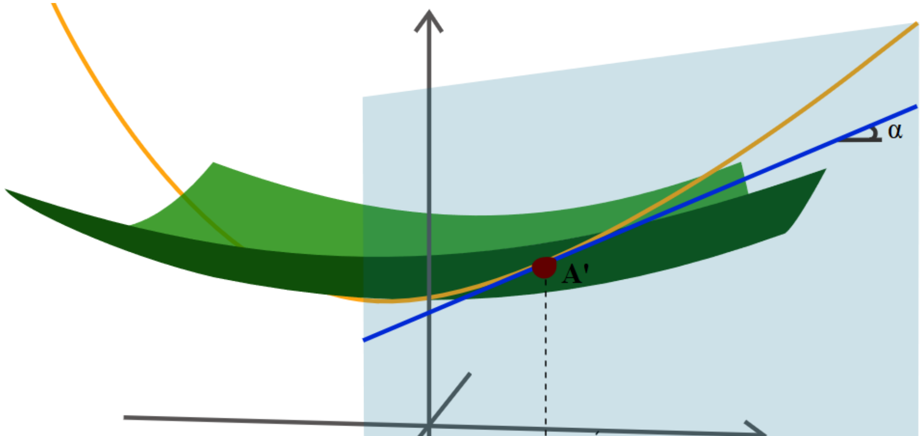
Differential

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

$$df_{\mathbf{x}_0}(\mathbf{v}) := \lim_{h \to 0} \frac{f(\mathbf{x}_0 + h\mathbf{v}) - f(\mathbf{x}_0)}{h}$$

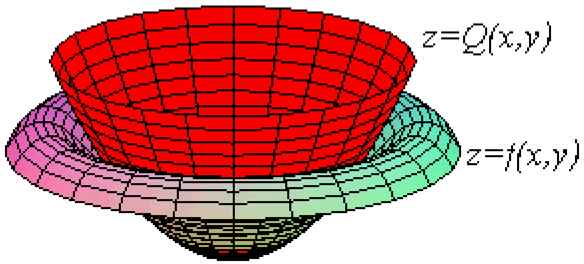
Proposition. df_{x_0} is a linear operator.

$$df_{\mathbf{x}_0}(\mathbf{v}) = \nabla f(\mathbf{x}_0) \cdot \mathbf{v}$$



Notions from Calculus

$$f : \mathbb{R}^n \to \mathbb{R} \to H_{ij} = \frac{\partial^2 f}{\partial x^i \partial x^j}$$



$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)^\top (\mathbf{x} - \mathbf{x}_0) + (\mathbf{x} - \mathbf{x}_0)^\top H f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)$$

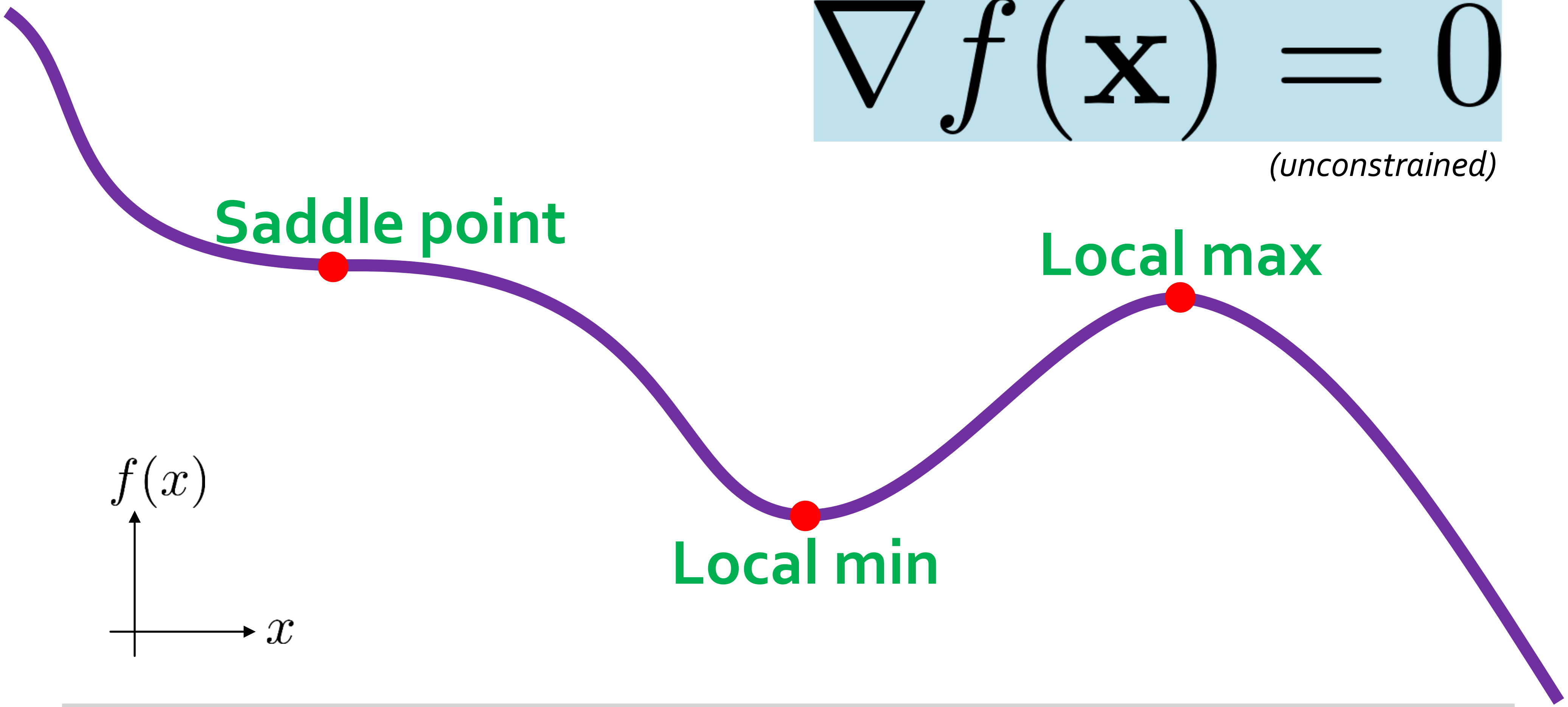
http://math.etsu.edu/multicalc/prealpha/Chap2/Chap2-5/10-3a-t3.gif

Hessian

From Optimization to Root-Finding

$$\nabla f(\mathbf{x}) = 0$$

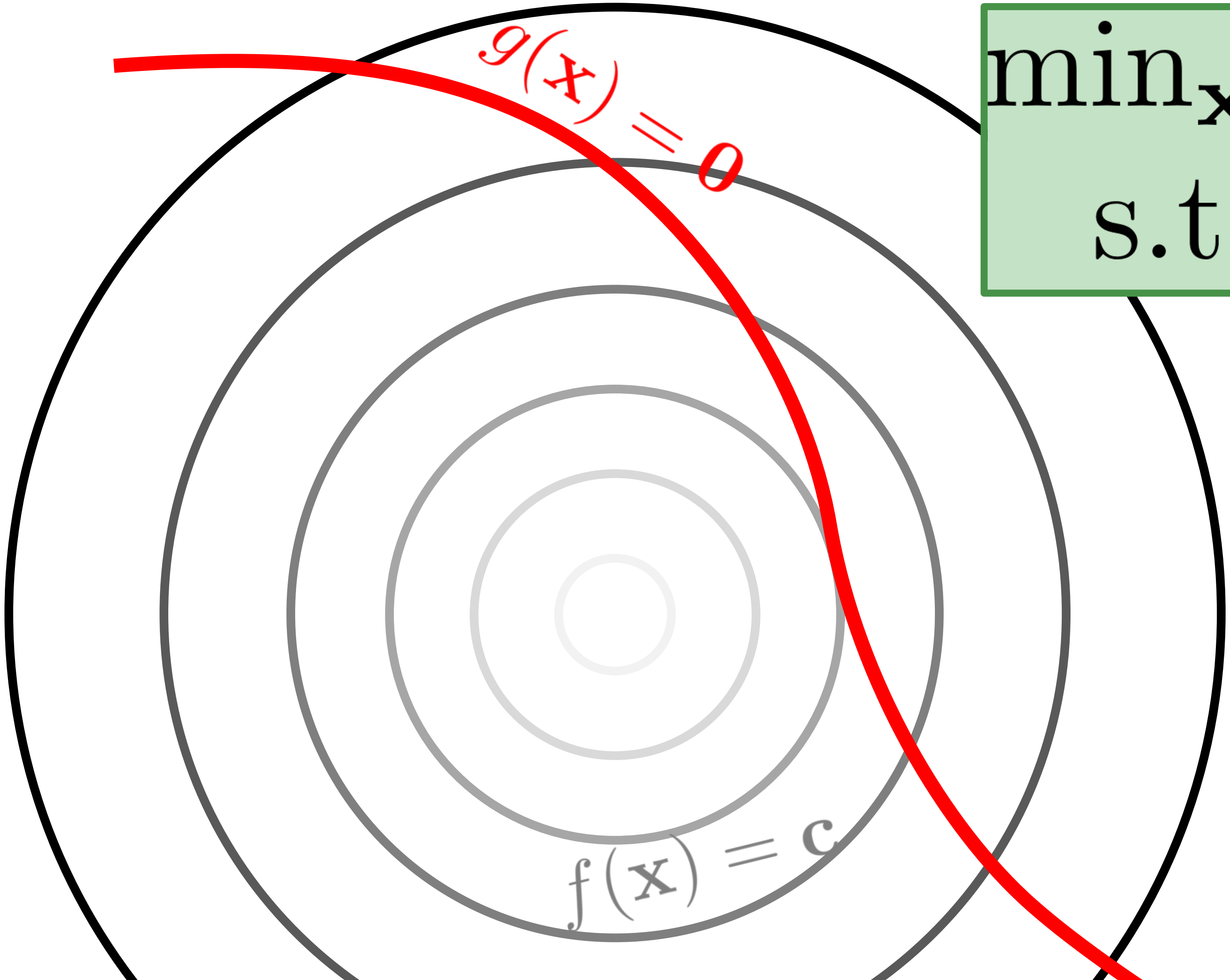
(unconstrained)



Critical point

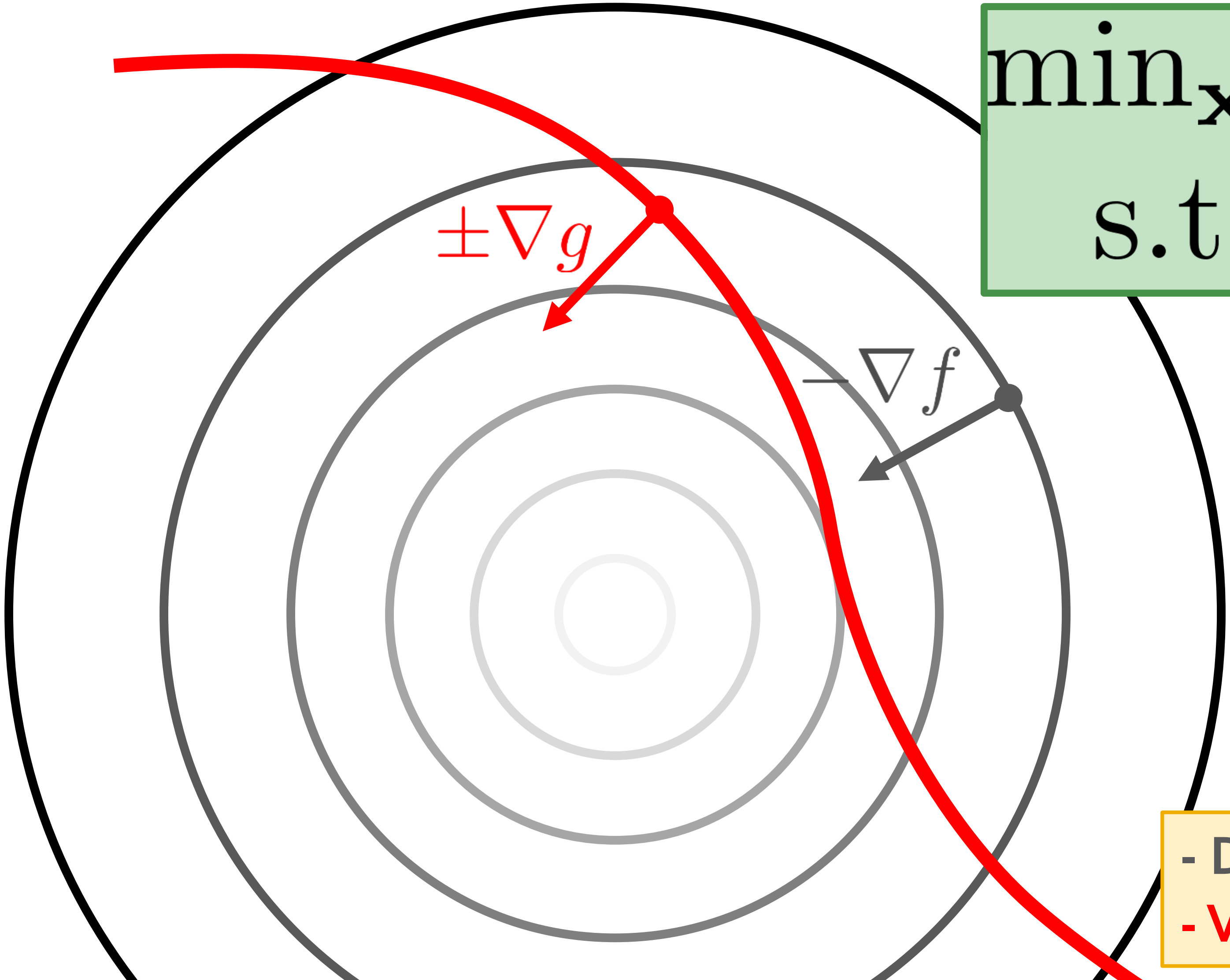
Lagrange Multipliers: Idea

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



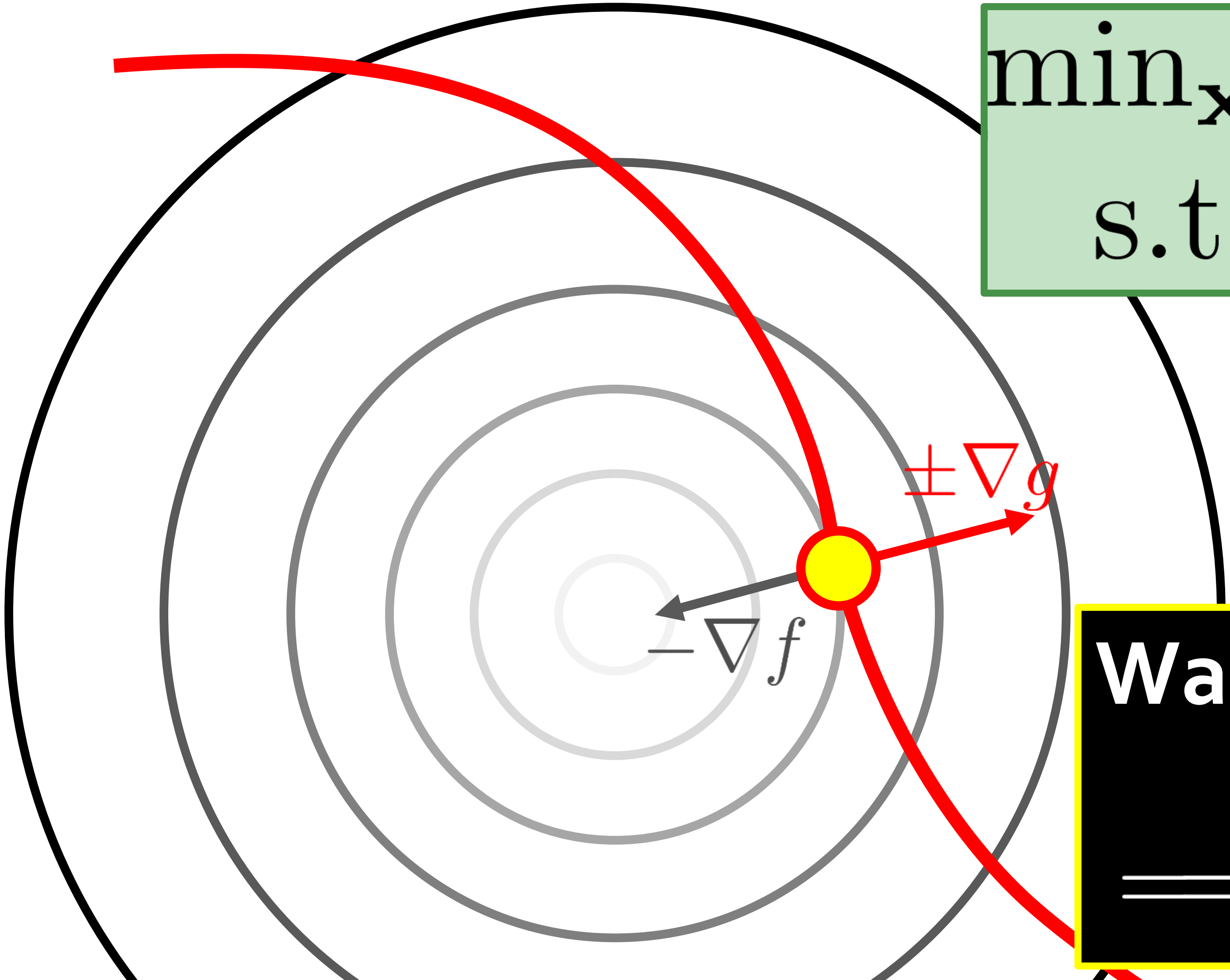
Lagrange Multipliers: Idea

$$\min_{\mathbf{x}} \quad f(\mathbf{x})$$
$$\text{s.t.} \quad g(\mathbf{x}) = \mathbf{0}$$



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

Lagrange Multipliers: Idea



$$\min_{\mathbf{x}} \quad f(\mathbf{x})$$
$$\text{s.t.} \quad g(\mathbf{x}) = \mathbf{0}$$

Want:

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

$$\implies \nabla f = \lambda \nabla g$$

Use of Lagrange Multipliers

Turns constrained optimization into

unconstrained root-finding.

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

$$g(x) = 0$$

Example: Symmetric Eigenvectors

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

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

$$\implies Ax = \lambda x$$

(New for 2023!) See Course Notes For Details

- Lagrange multipliers
- KKT conditions
- Special cases:
  - Linear problems
  - Eigenvalue problems

Advanced Topic: Variational Calculus

Sometimes your unknowns

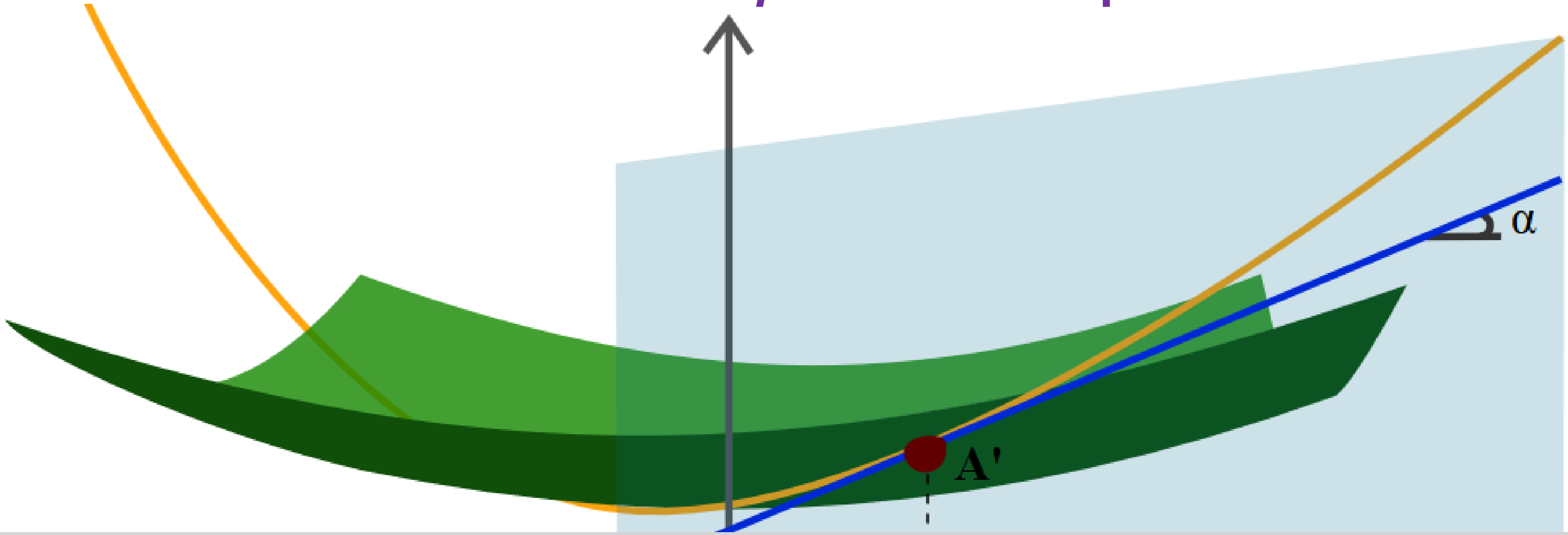
are not numbers!

Can we use calculus to optimize anyway?

Gâteaux Derivative

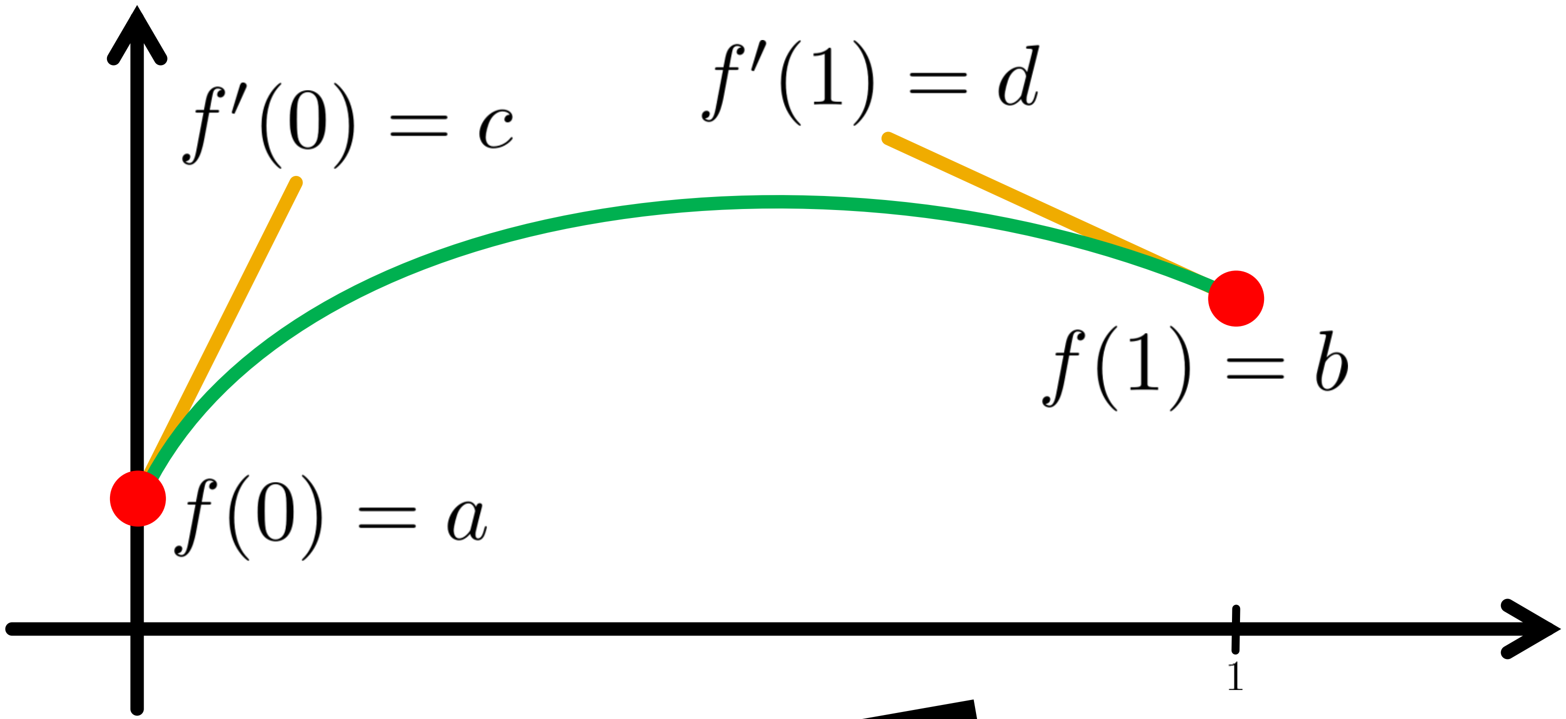
$$dF[u;\psi] := \frac{d}{dh}\mathcal{F}[u + h\psi]|_{h=0}$$

Vanishes for all ψ at a critical point!



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

Example: Cubic Splines



- Given
  - $f(0) = a$
  - $f(1) = b$
  - $f'(0) = c$
  - $f'(1) = d$
- Compute
  - "Reasonable" interpolant $f(t)$
  - Smooth, near-linear: $f''(t) \approx 0$

On the board:

$$E[f] := \int_0^1 |f''(t)|^2 \, dt$$

Linear and Variational Problems

Justin Solomon

6.8410: Shape Analysis
Spring 2023