

Linear and Variational Problems

Justin Solomon

6.8410: Shape Analysis

Fall 2026



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

(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

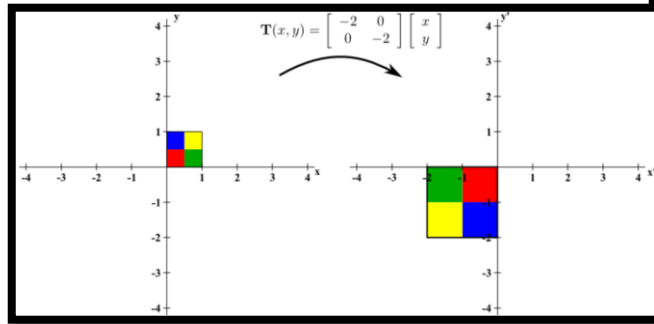
$$\mathbf{v} = v^k \mathbf{e}_k$$



Sum repeated upper/lower indices

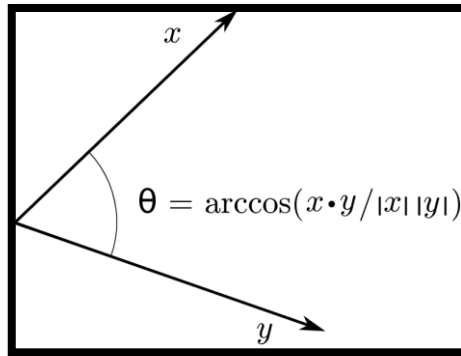
Same Data Structure, Two Uses

- **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_o_o_m2

Protip:
Know your input and output

Matrices obscure geometry

Linear Map

$$\begin{pmatrix} L_1^1 & L_2^1 & \cdots & L_n^1 \\ L_1^2 & L_2^2 & \cdots & L_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ L_1^m & L_2^m & \cdots & L_n^m \end{pmatrix} \begin{pmatrix} v^1 \\ v^2 \\ \vdots \\ v^n \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^n L_k^1 v^k \\ \sum_{k=1}^n L_k^2 v^k \\ \vdots \\ \sum_{k=1}^n L_k^m v^k \end{pmatrix} := \begin{pmatrix} w^1 \\ w^2 \\ \vdots \\ w^m \end{pmatrix}$$

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

$$\begin{pmatrix} L_1^1 & L_2^1 & \cdots & L_n^1 \\ L_1^2 & L_2^2 & \cdots & L_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ L_1^m & L_2^m & \cdots & L_n^m \end{pmatrix} \begin{pmatrix} v^1 \\ v^2 \\ \vdots \\ v^n \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^n L_k^1 v^k \\ \sum_{k=1}^n L_k^2 v^k \\ \vdots \\ \sum_{k=1}^n L_k^m v^k \end{pmatrix} := \begin{pmatrix} w^1 \\ w^2 \\ \vdots \\ w^m \end{pmatrix}$$

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

Upper/lower indices matter

New Terminology

A x
matrix vector

$x \mapsto Ax$
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

 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

Search

Broaden your search using Semantic Scholar

Search

(Help | Advanced search)

Computer Science > Numerical Analysis

How Accurate is $\text{inv}(A)*b$?

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 $\text{inv}(A)$ 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, $x = \text{inv}(A)*b$ 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

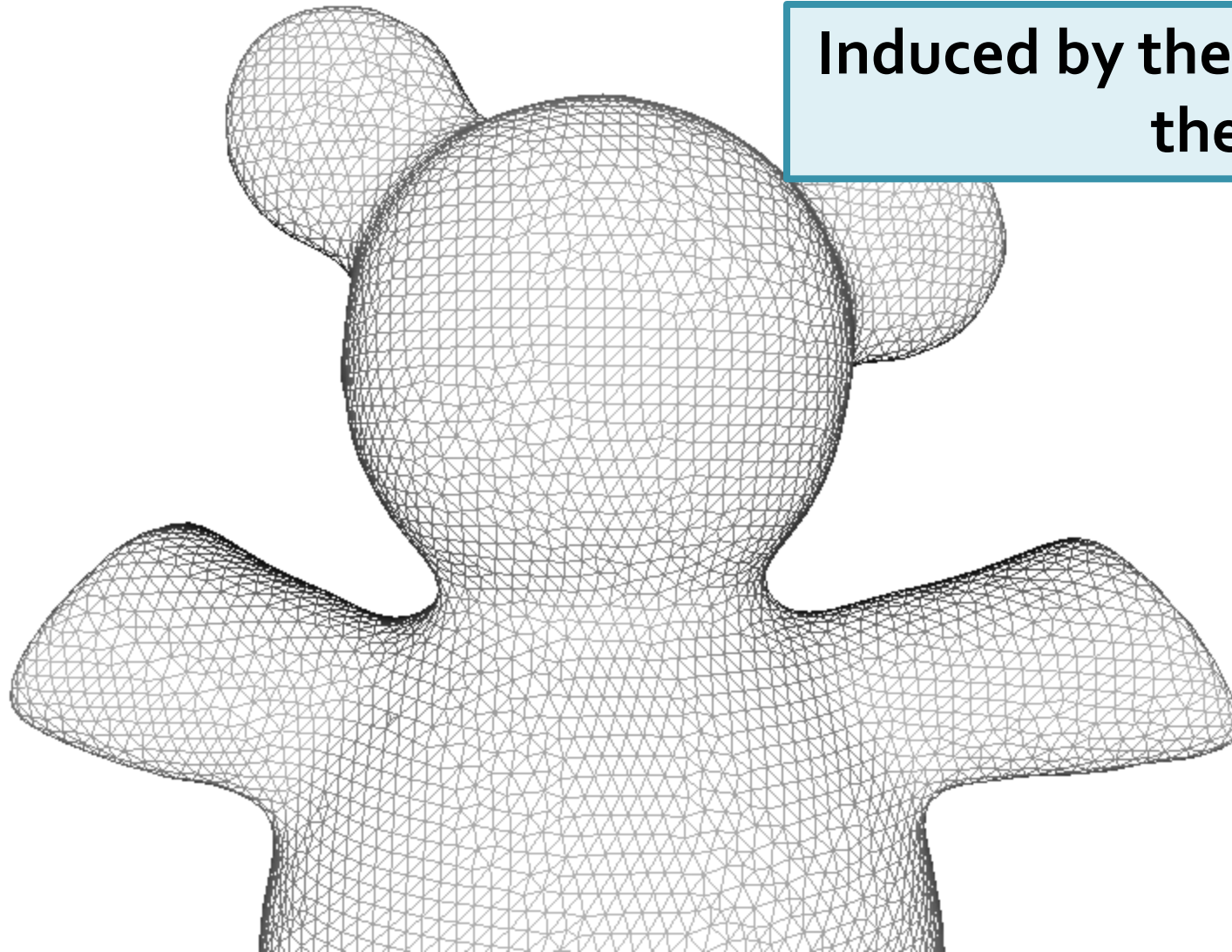
$$\text{inv}(A) * b \ll (A' * A) \setminus (A' * b) \ll A \setminus b$$

Example of a Structured Problem

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

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

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!**

The image shows two overlapping web browser windows. The background window is the Julia documentation page for Sparse Arrays, located at <https://docs.julialang.org/en/v0.7.0/stdlib/SparseArrays/>. It features the Julia logo, a search bar, and a sidebar with navigation links like Home, Manual, Getting Started, Variables, Integers and Floating-Point Numbers, Mathematical Operations and Elementary Functions, Complex and Rational Numbers, and Strings. The main content area is titled 'Sparse Arrays' and includes a link to 'Edit on GitHub'. Below the title, it states: 'Julia has support for sparse vectors and [sparse matrices](#) in the `SparseArrays` stdlib module. Sparse arrays are arrays that contain enough zeros that storing them in a special data structure can save execution time, compared to dense arrays.' The section 'Compressed Sparse Column (CSC) Sparse Matrices' is partially visible, explaining that in Julia, sparse matrices are stored in the Compressed Sparse Column format, with the type `SparseMatrixCSC{Tv, Ti}`, where `Tv` is the type of the stored values and `Ti` is the type of the stored column pointers and row indices. The foreground window is the MathWorks documentation page for Sparse Matrices, located at <https://www.mathworks.com/help/matlab/sparse-matrices.html>. It features the MathWorks logo, navigation links like Products, Solutions, Academia, Support, Community, and Events, and a search bar. The main content area is titled 'Sparse Matrices' and includes a link to 'R2018b'. Below the title, it states: 'Elementary sparse matrices, reordering algorithms, iterative methods, sparse linear algebra'. A sidebar on the left lists 'CONTENTS' with links to 'Documentation Home', 'MATLAB', 'Mathematics', and 'Elementary Math'.

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

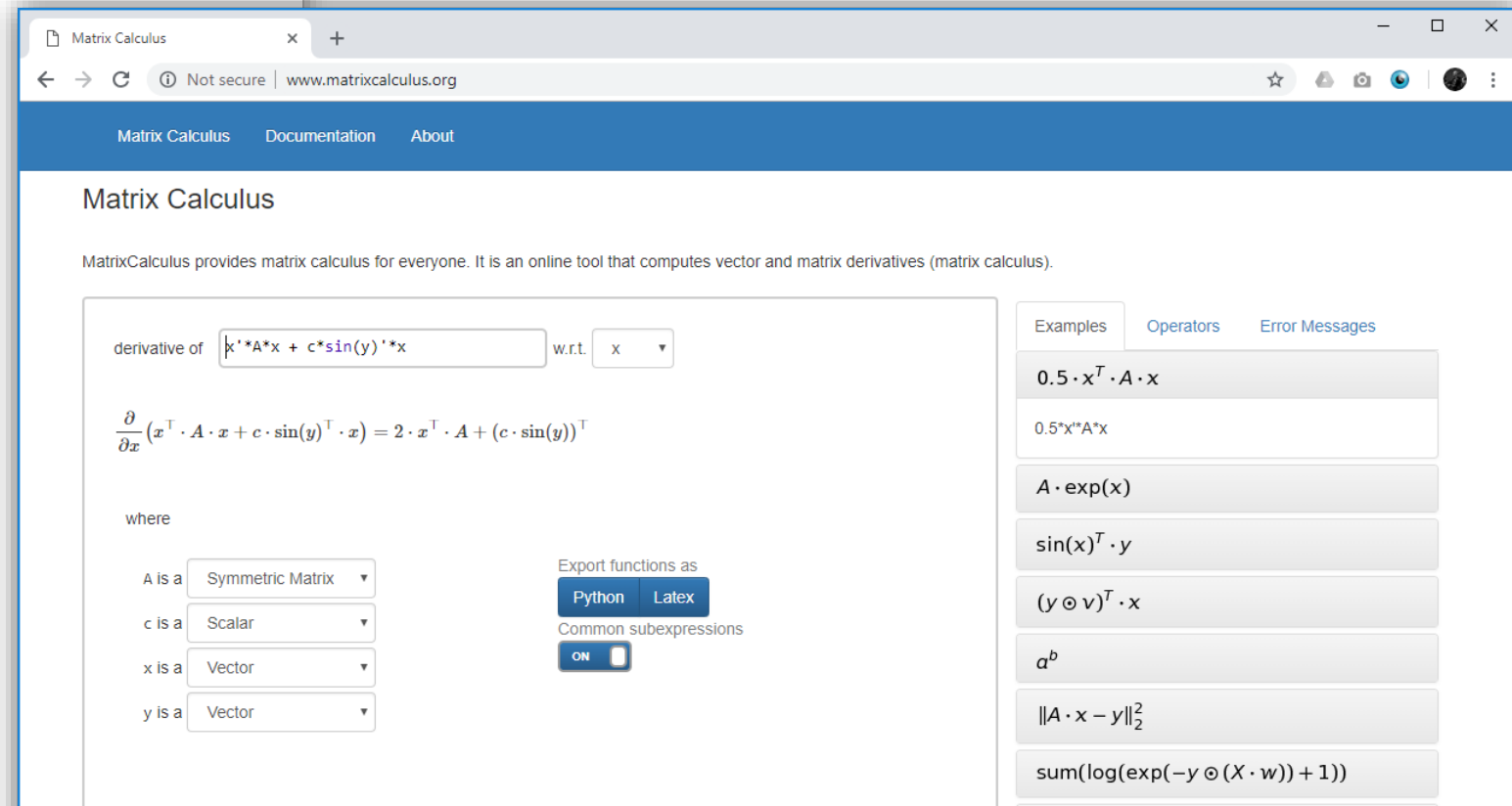
Aside: Matrix Calculus

The Matrix Cookbook

[<http://matrixcookbook.com>]

Kaare Brandt Petersen
Michael Syskind Pedersen

VERSION: NOVEMBER 15, 2012



The screenshot shows a web browser window with the URL www.matrixcalculus.org. The page title is "Matrix Calculus". The navigation bar includes links for "Matrix Calculus", "Documentation", and "About".

The main content area is titled "Matrix Calculus" and contains the text: "MatrixCalculus provides matrix calculus for everyone. It is an online tool that computes vector and matrix derivatives (matrix calculus)."

The interface shows a calculation of the derivative of the expression $x^T A x + c \sin(y)^T x$ with respect to x . The result is displayed as:

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

Below the result, it asks "where" and provides dropdown menus for the variables:

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

There are also buttons to "Export functions as" (Python, Latex) and a toggle for "Common subexpressions" (ON).

On the right side, there is a sidebar with tabs for "Examples", "Operators", and "Error Messages". The "Examples" tab is active, showing a list of mathematical expressions:

- $0.5 \cdot x^T \cdot A \cdot x$
- $0.5 x^T A x$
- $A \cdot \exp(x)$
- $\sin(x)^T \cdot y$
- $(y \odot v)^T \cdot x$
- a^b
- $\|A \cdot x - y\|_2^2$
- $\text{sum}(\log(\exp(-y \odot (X \cdot w)) + 1))$

Optimization Terminology

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

Objective (“Energy Function”)

Optimization Terminology

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

Equality Constraints

Optimization Terminology

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

Inequality Constraints

Encapsulates Many Problems

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

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

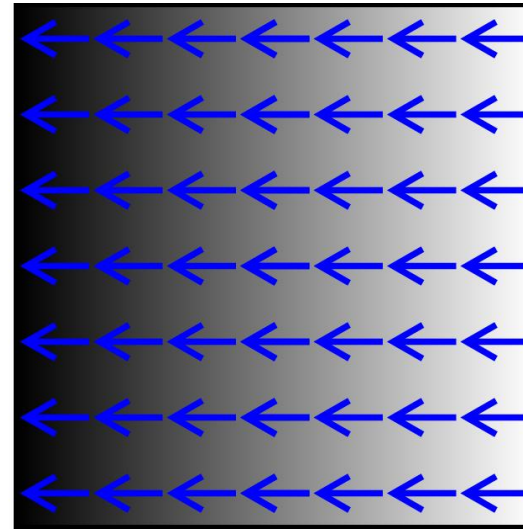
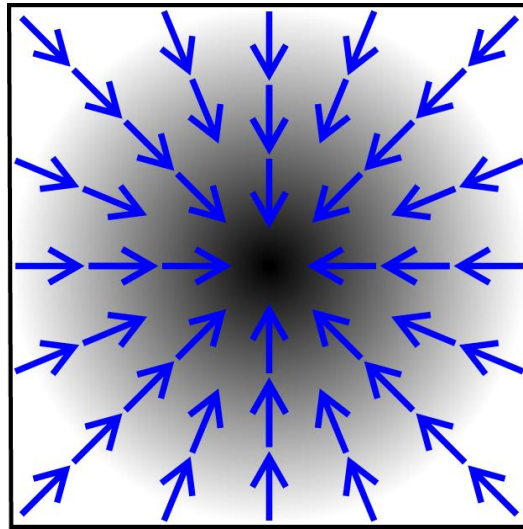
$$A\mathbf{x} = \lambda\mathbf{x} \Leftrightarrow f(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x}, g(\mathbf{x}) = \|\mathbf{x}\|_2 - 1$$

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

Notions from Calculus

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

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



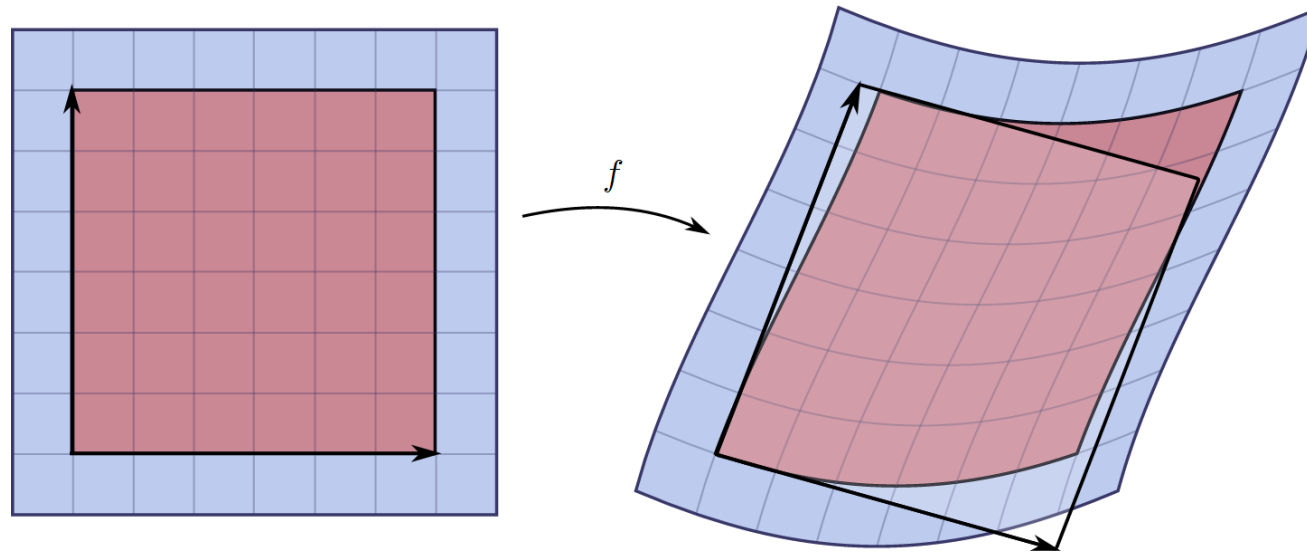
<https://en.wikipedia.org/?title=Gradient>

Gradient

Notions from Calculus

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

$$\rightarrow (Df)_j^i = \frac{\partial f^i}{\partial x^j}$$



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

Jacobian

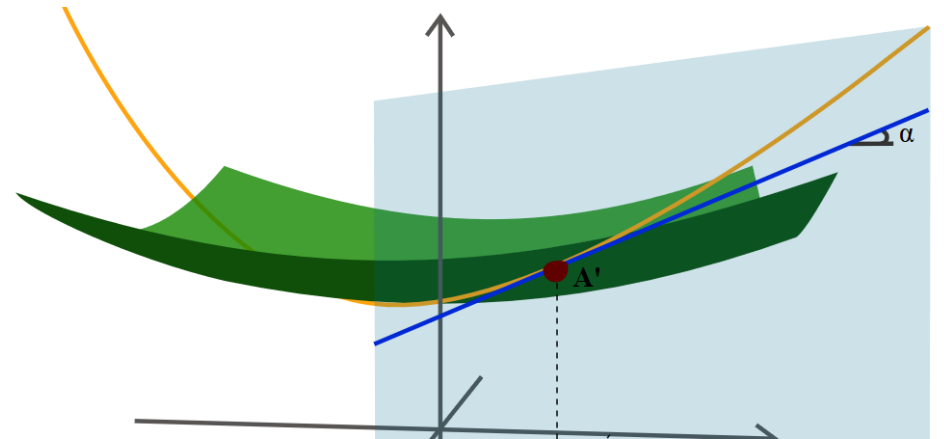
Differential

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

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

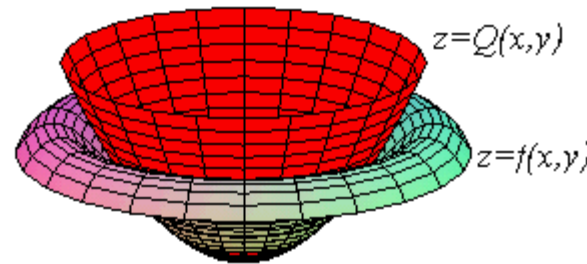
Proposition. df_{x_0} is a linear operator.

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



Notions from Calculus

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



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

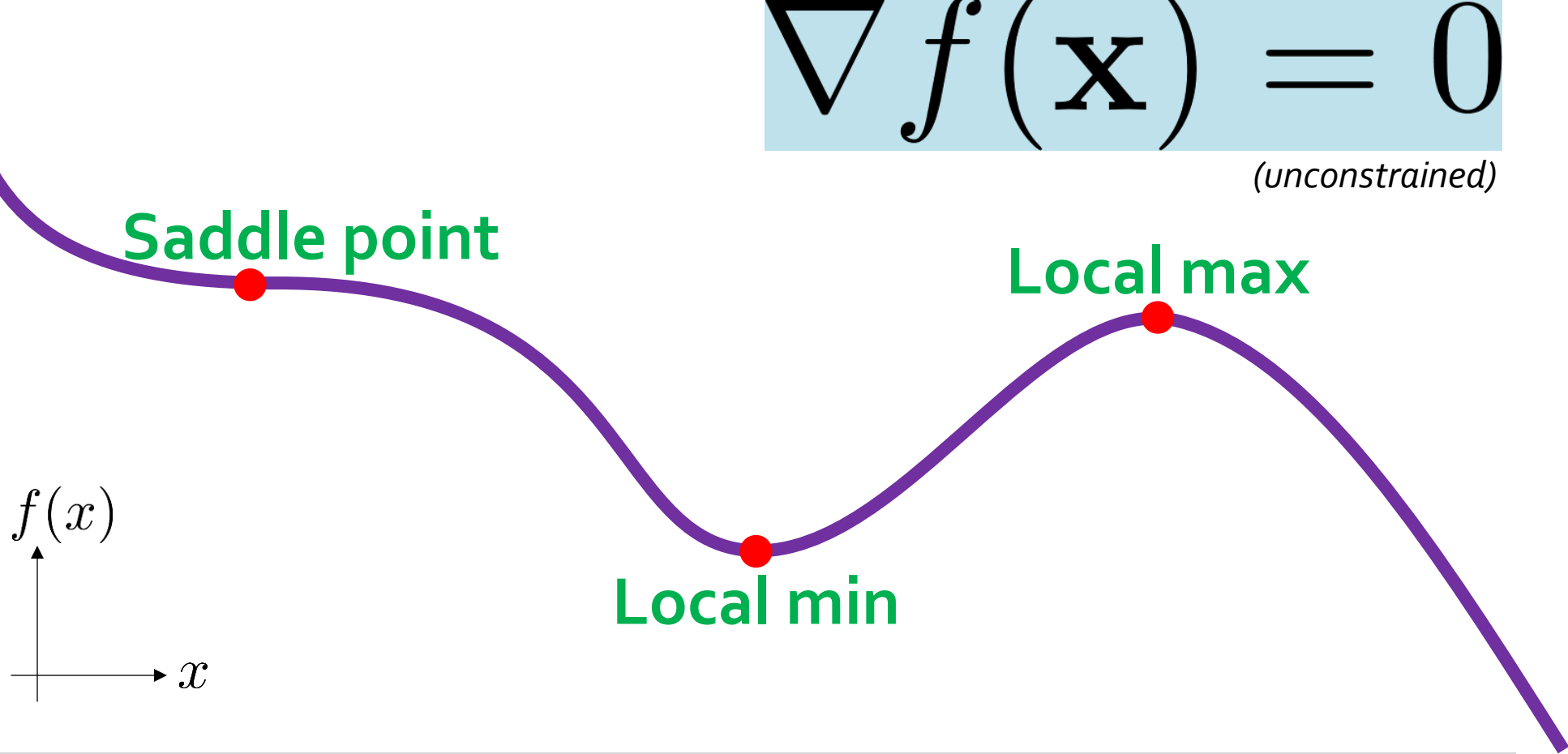
<http://math.etsu.edu/multicalc/prealpha/Chap2/Chap2-5/10-3a-t3.gif>

Hessian

From Optimization to Root-Finding

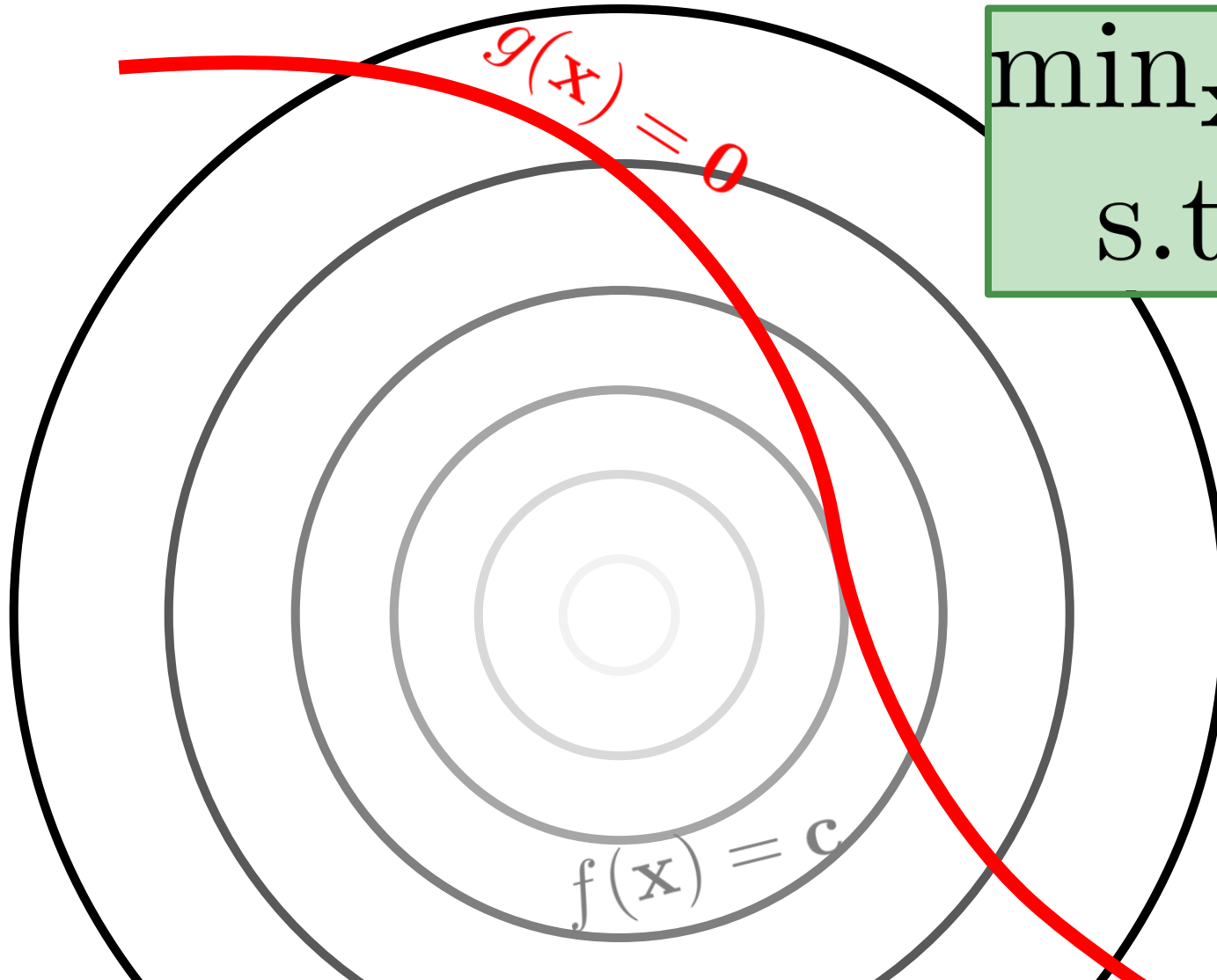
$$\nabla f(\mathbf{x}) = 0$$

(unconstrained)



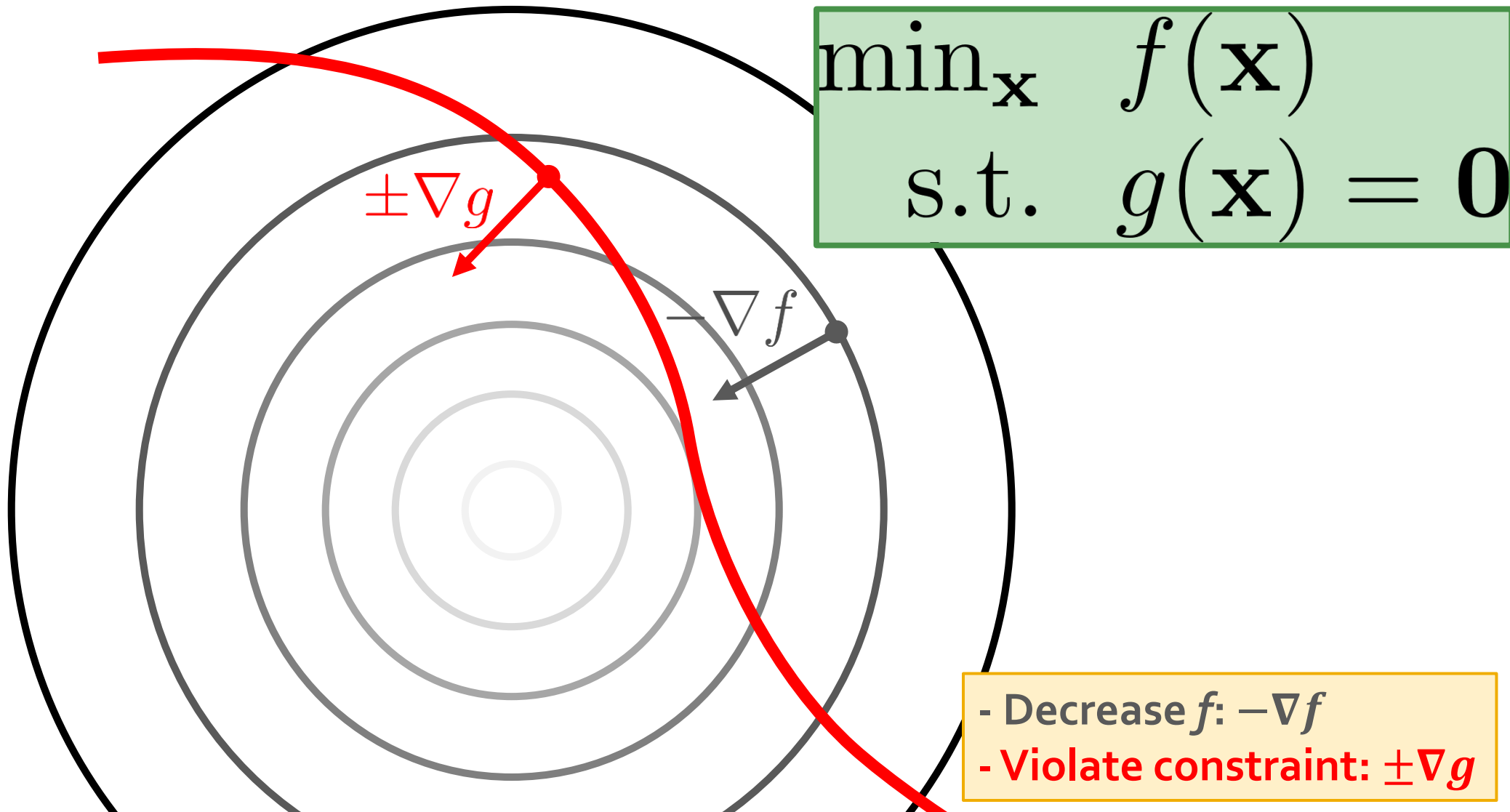
Critical point

Lagrange Multipliers: Idea

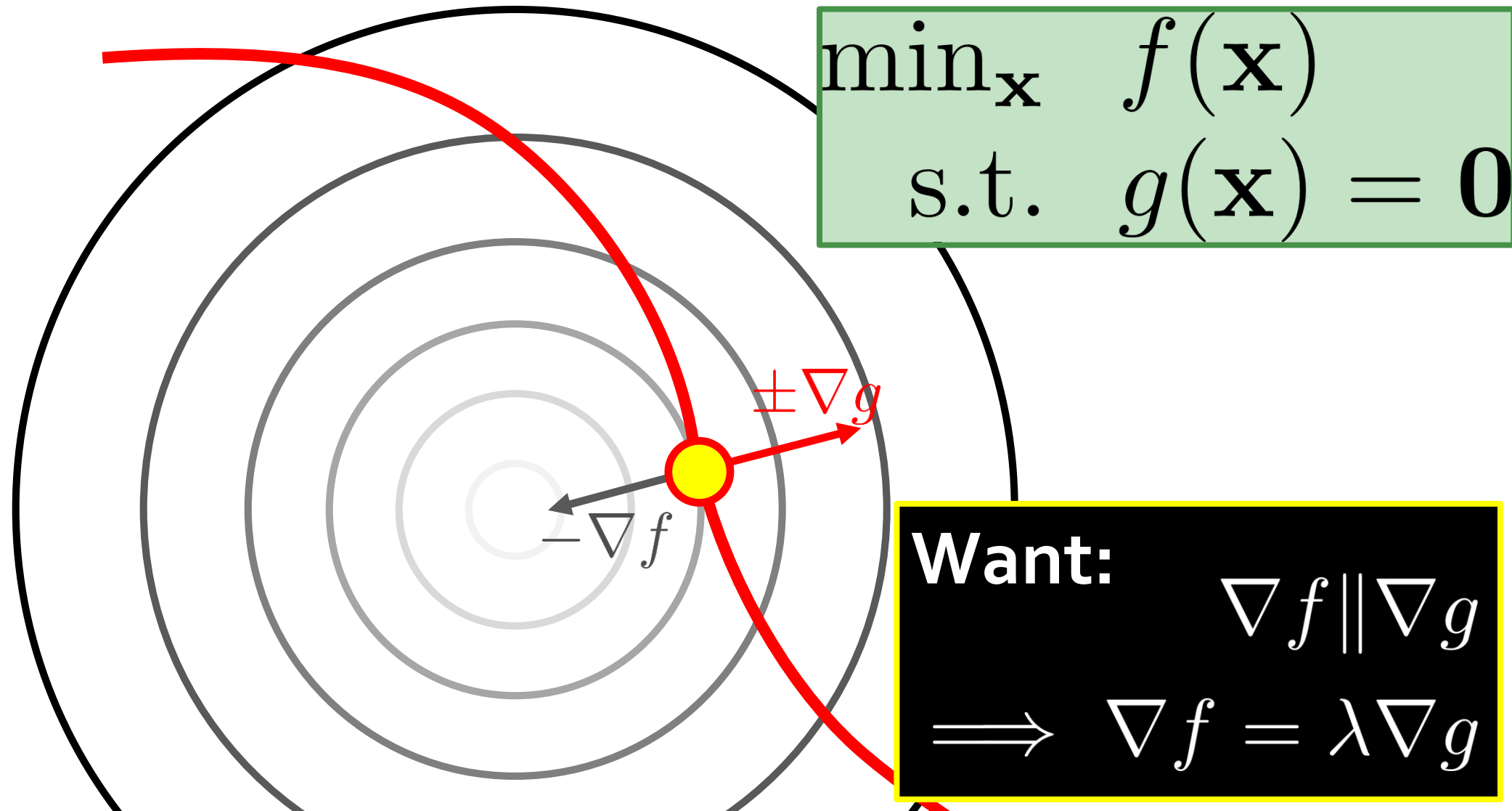


$$\begin{array}{ll} \min_{\mathbf{x}} & f(\mathbf{x}) \\ \text{s.t.} & g(\mathbf{x}) = 0 \end{array}$$

Lagrange Multipliers: Idea



Lagrange Multipliers: Idea



Use of Lagrange Multipliers

Turns constrained optimization into
unconstrained root-finding.

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

$$g(x) = 0$$

Example: Symmetric Eigenvectors

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

$$g(x) = \|x\|_2^2 - 1 \implies \nabla g(x) = 2x$$

$$\implies Ax = \lambda x$$

See Course Notes For Details

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

Advanced Topic: Variational Calculus

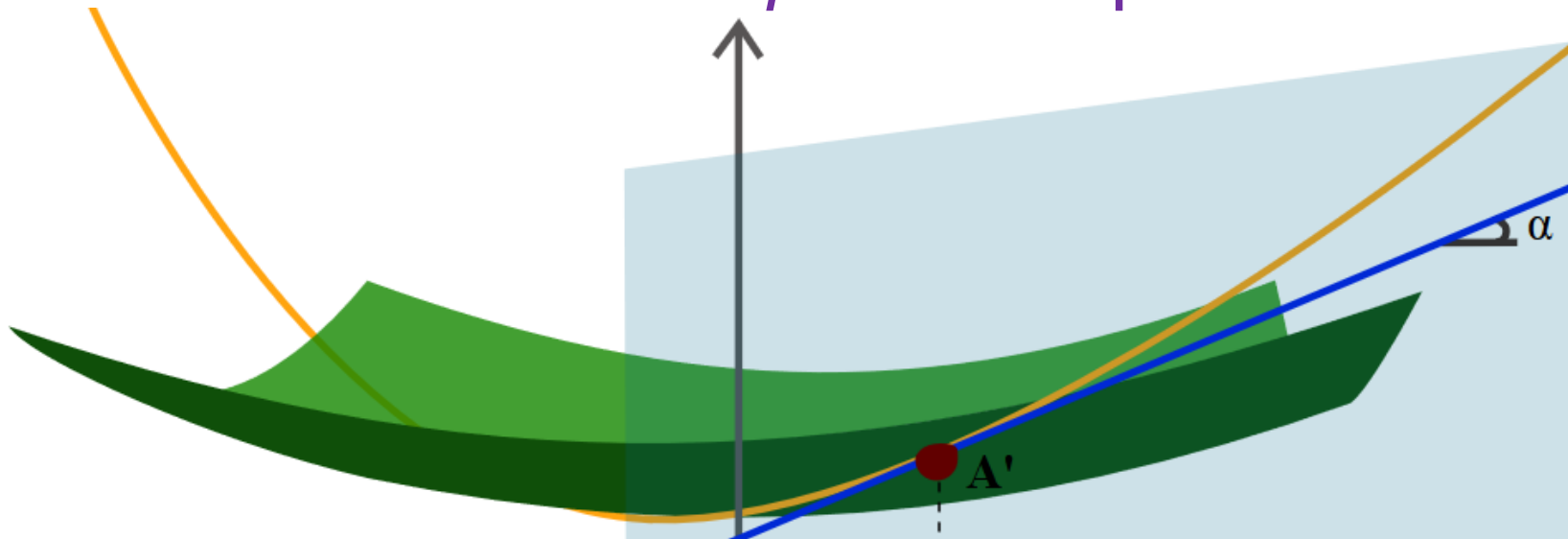
Sometimes your unknowns
are not numbers!

Can we use calculus to optimize anyway?

Gâteaux Derivative

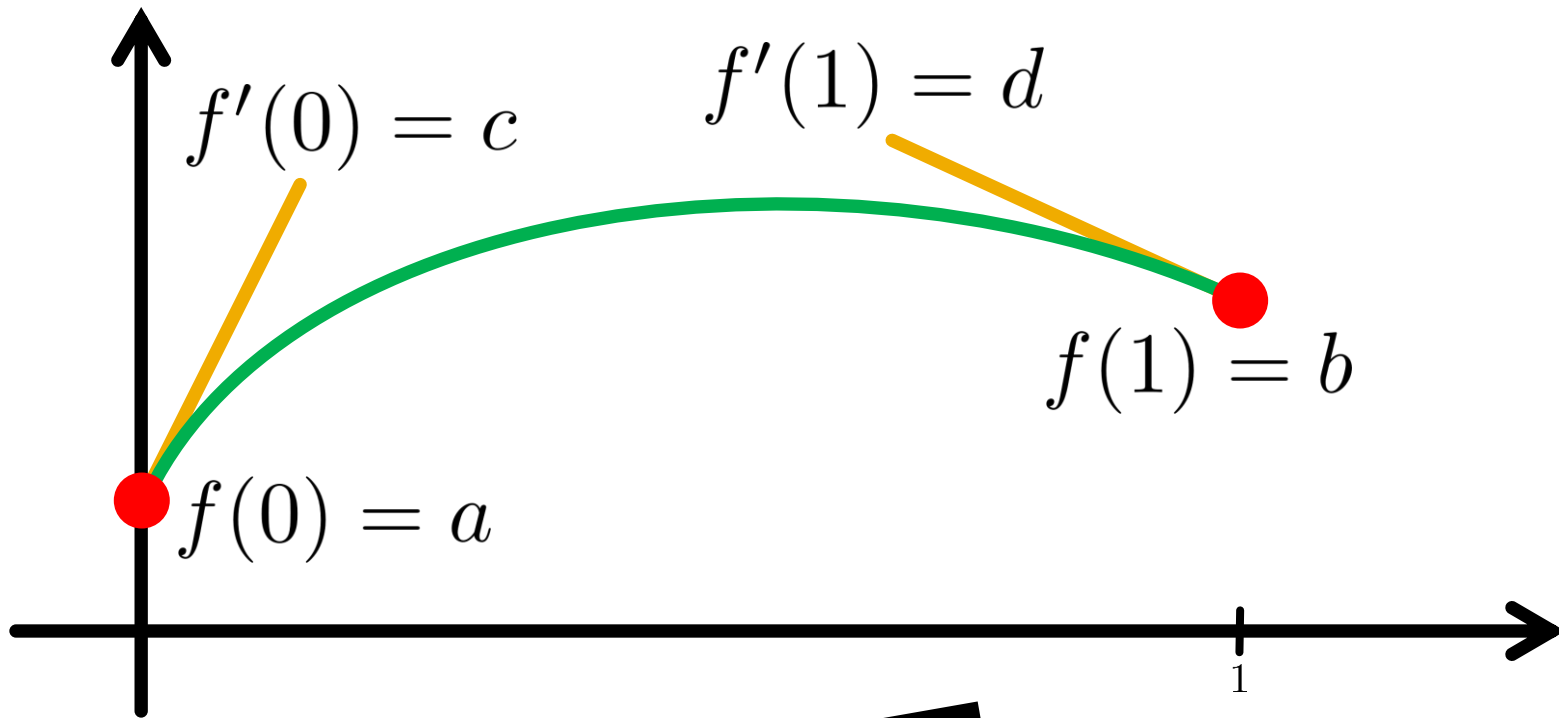
$$d\mathcal{F}[u; \psi] := \frac{d}{dh} \mathcal{F}[u + h\psi] \big|_{h=0}$$

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

Fall 2026

