

Linear and Variational Problems

Justin Solomon

6.8410: Shape Analysis

Spring 2023



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

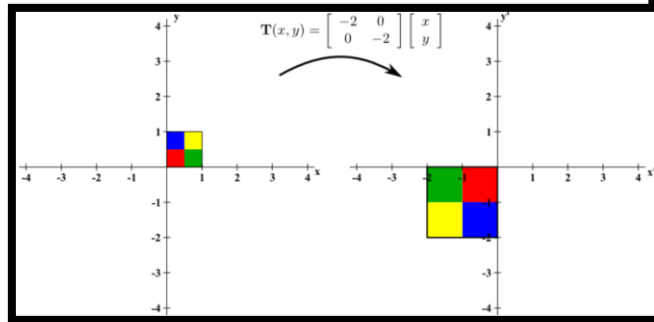
$$\mathbf{v} = v^k \mathbf{e}_k$$



Sum repeated upper/lower indices

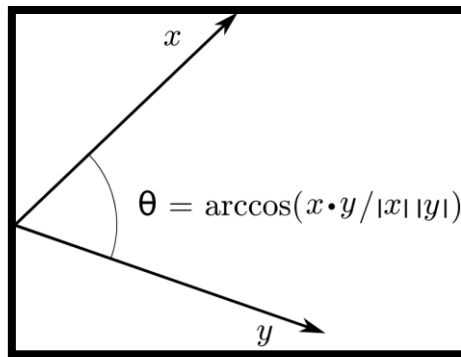
Same Data Structure, Two Uses

- **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

The screenshot shows a web browser window displaying the arXiv page for the paper "How Accurate is $\text{inv}(A)*b$?" by Alex Druinsky and Sivan Toledo. The browser's address bar shows the URL <https://arxiv.org/abs/1201.6035>. The page header includes the Cornell University Library logo and a search bar. The main content area features the paper title, authors, submission date (29 Jan 2012), and a summary paragraph. The summary states that 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. It notes that 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. The authors review the literature on the accuracy of this computation and present a self-contained numerical analysis of it.

The right sidebar contains several sections: "Download:" with links for PDF and other formats; "Current browse context:" showing the current context as cs.NA and navigation links; "Change to browse by:" with options for cs, math, and math.NA; "References & Citations" with a link to NASA ADS; "1 blog link" with a link to a blog post; "DBLP - CS Bibliography" with links for listing and bibtex; and "Bookmark" with various social media and search engine icons.

At the bottom of the page, there is a link back to the arXiv form interface and contact information.

Linear Solver Considerations

- **Never construct A^{-1} explicitly**
(if you can avoid it)
- **Added structure helps**
Sparsity, symmetry, positive definiteness,
bandedness

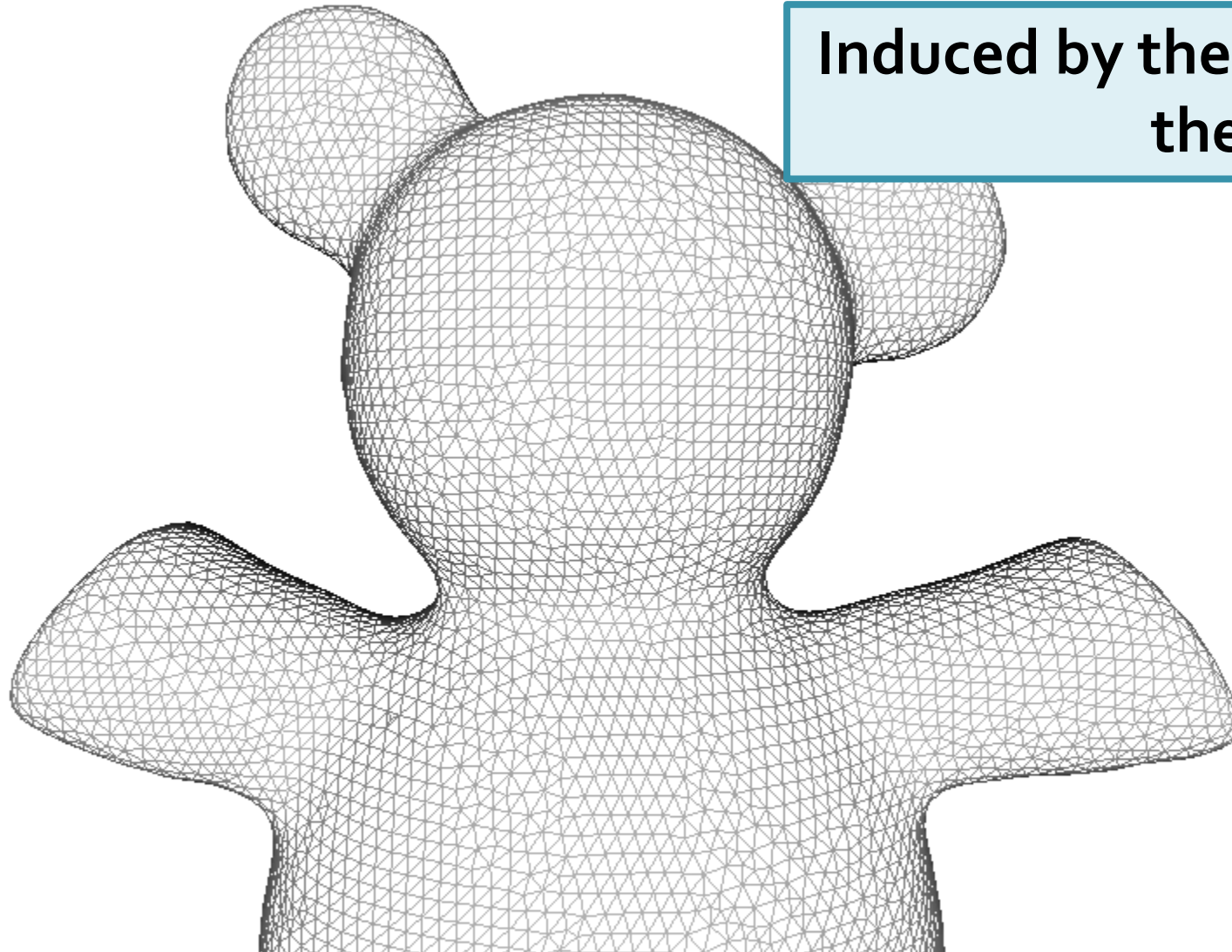
$$\text{inv}(A) * b \ll (A' * A) \setminus (A' * b) \ll A \setminus b$$

Example of a Structured Problem

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

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

Very Common: Sparsity



Induced by the **connectivity** of
the triangle mesh.

Two Classes of Solvers

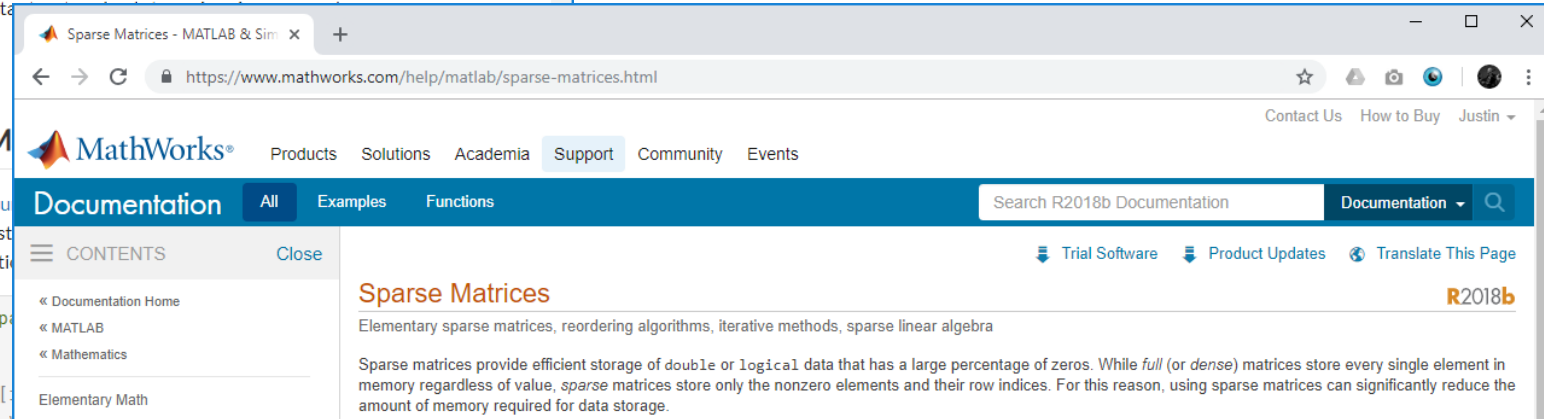
- **Direct** (*explicit matrix*)
 - **Dense:** Gaussian elimination/LU, QR for least-squares
 - **Sparse:** Reordering (SuiteSparse, Eigen)
- **Iterative** (*apply matrix repeatedly*)
 - **Positive definite:** Conjugate gradients
 - **Symmetric:** MINRES, GMRES
 - **Generic:** LSQR

For 6.8410

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



A screenshot of a web browser showing the Julia documentation page for Sparse Arrays. The browser's address bar displays the URL `https://docs.julialang.org/en/v0.7.0/stdlib/SparseArrays/`. The page features the Julia logo on the left and a navigation menu. The main content area is titled "Sparse Arrays" and includes a link to "Edit on GitHub". The text below the title 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."



A screenshot of the MathWorks documentation page for Sparse Matrices. The browser's address bar shows the URL `https://www.mathworks.com/help/matlab/sparse-matrices.html`. The page has a blue header with the MathWorks logo and navigation links for Products, Solutions, Academia, Support, Community, and Events. A search bar is present with the text "Search R2018b Documentation". The main content area is titled "Sparse Matrices" and includes a sub-heading "Elementary sparse matrices, reordering algorithms, iterative methods, sparse linear algebra". The text below explains: "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

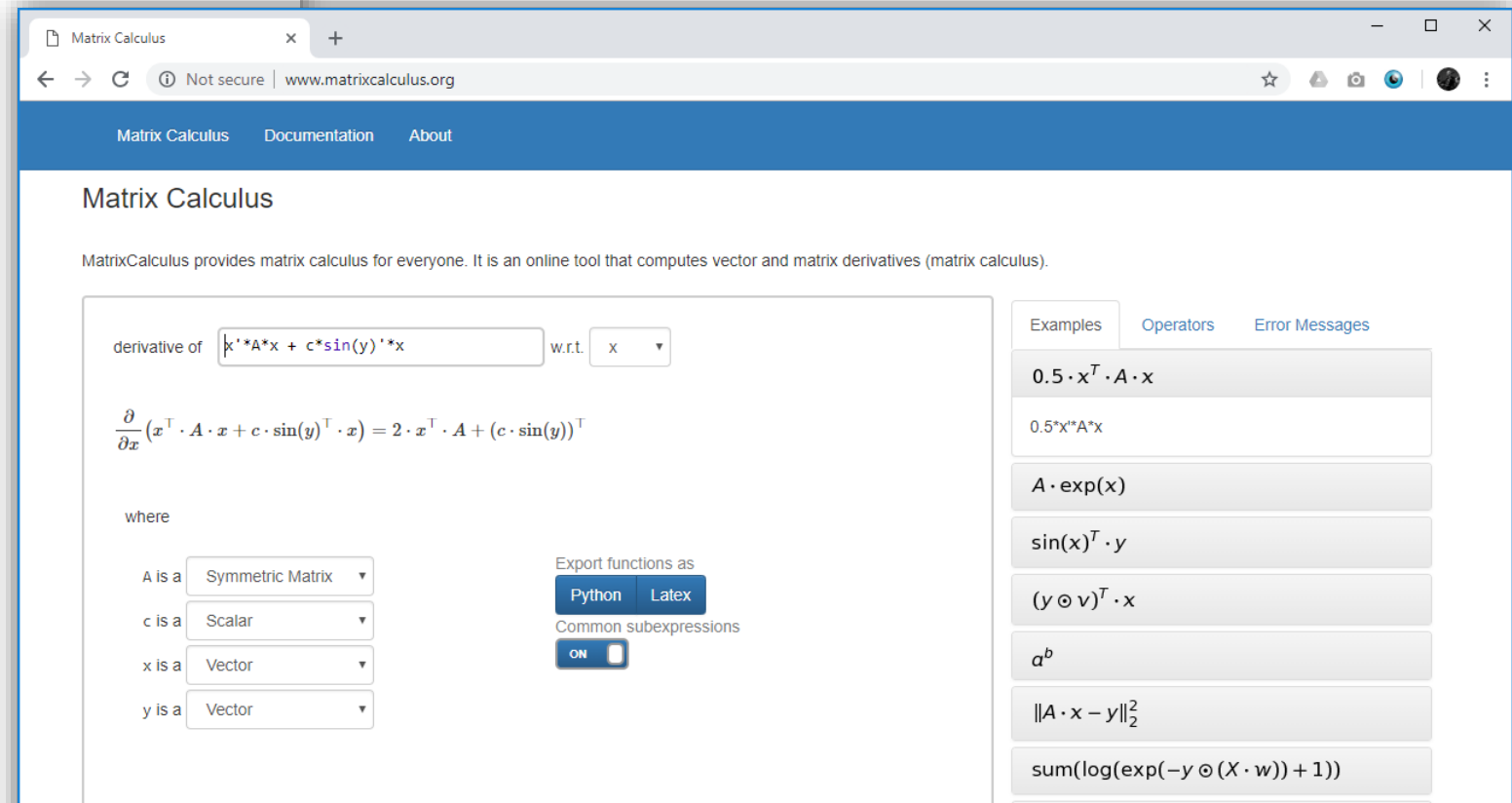
Aside: Matrix Calculus

The Matrix Cookbook

[<http://matrixcookbook.com>]

Kaare Brandt Petersen
Michael Syskind Pedersen

VERSION: NOVEMBER 15, 2012



Matrix Calculus

Not secure | www.matrixcalculus.org

Matrix Calculus Documentation About

Matrix Calculus

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

derivative of w.r.t.

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

where

A is a

c is a

x is a

y is a

Export functions as

Common subexpressions

Examples Operators Error Messages

-
-
-
-
-
-
-
-

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

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

Equality Constraints

Optimization Terminology

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

Inequality Constraints

Encapsulates Many Problems

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

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

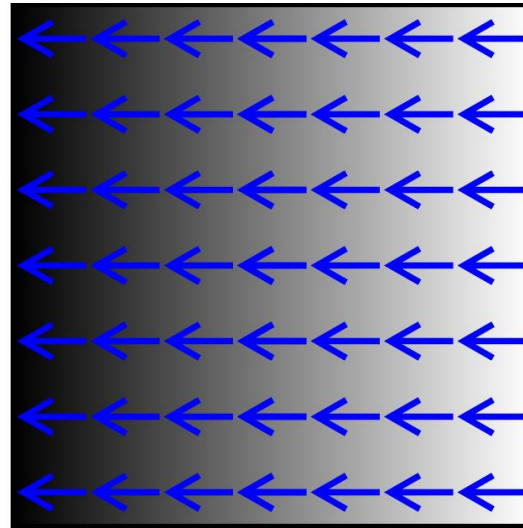
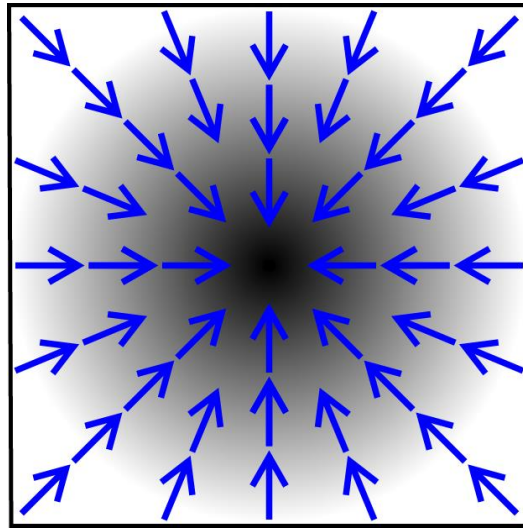
$$A\mathbf{x} = \lambda\mathbf{x} \leftrightarrow f(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x}, g(\mathbf{x}) = \|\mathbf{x}\|_2 - 1$$

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

Notions from Calculus

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

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



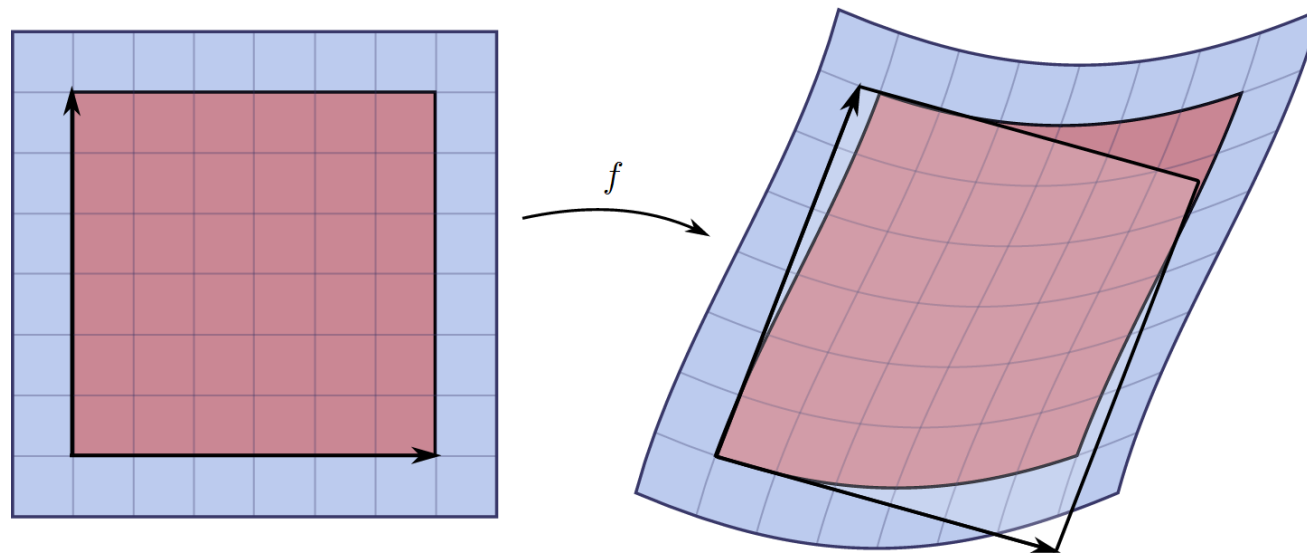
<https://en.wikipedia.org/?title=Gradient>

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

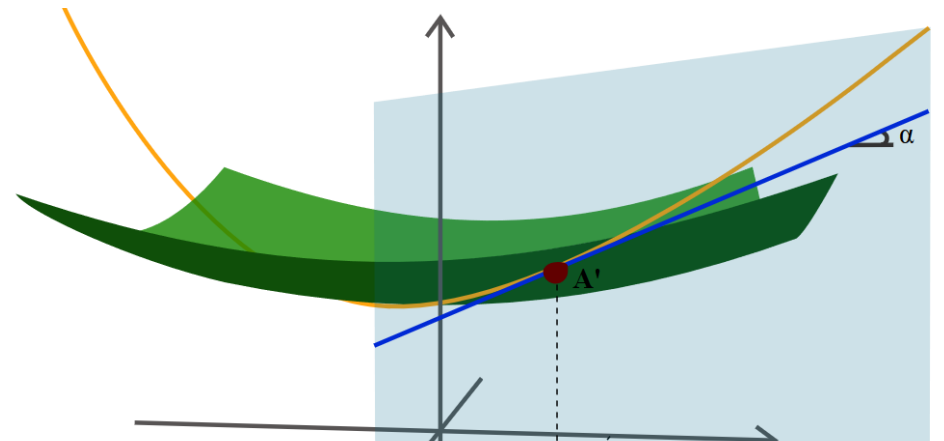
Differential

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

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

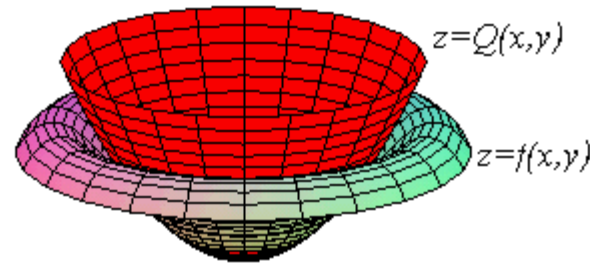
Proposition. df_{x_0} is a linear operator.

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



Notions from Calculus

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



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

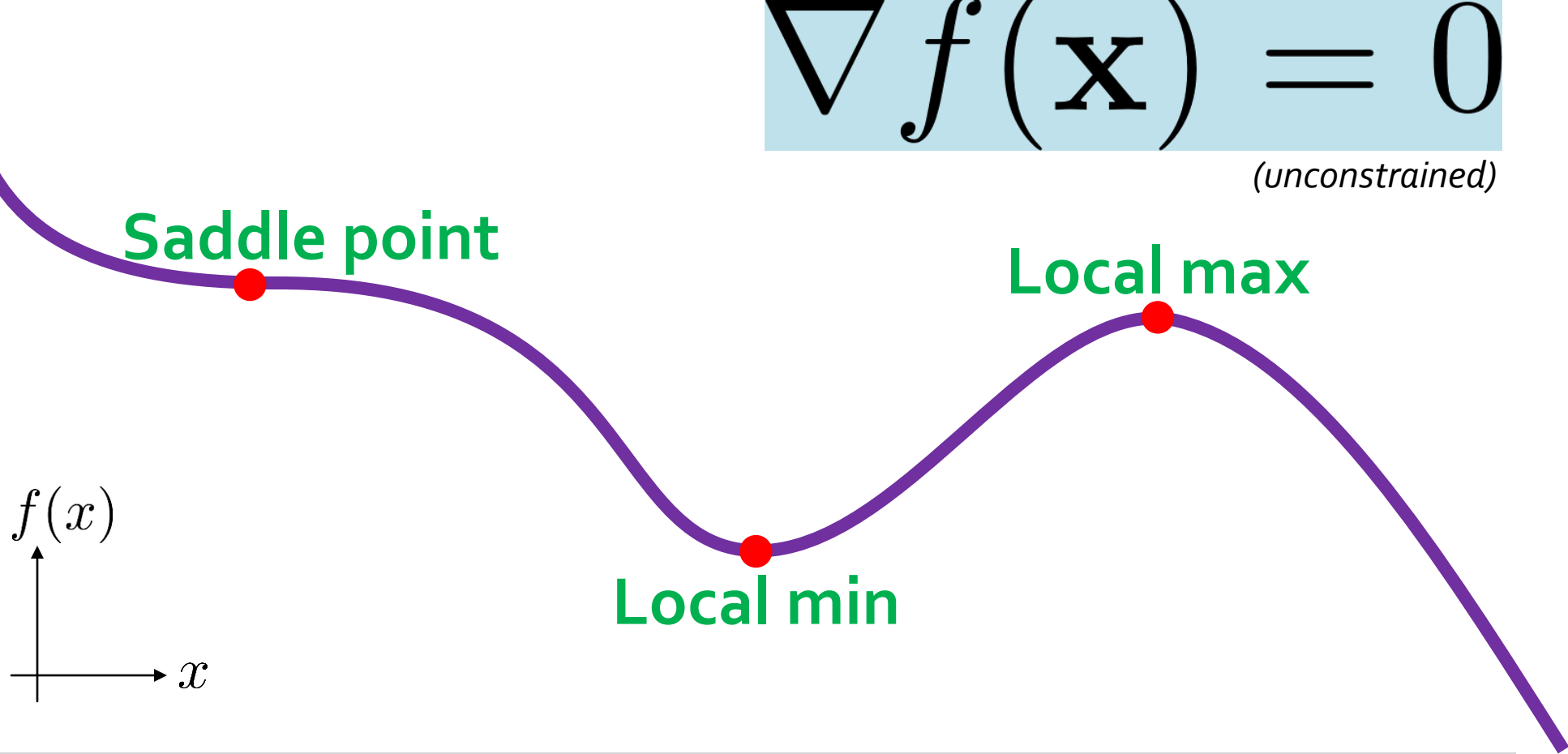
<http://math.etsu.edu/multicalc/prealpha/Chap2/Chap2-5/10-3a-t3.gif>

Hessian

From Optimization to Root-Finding

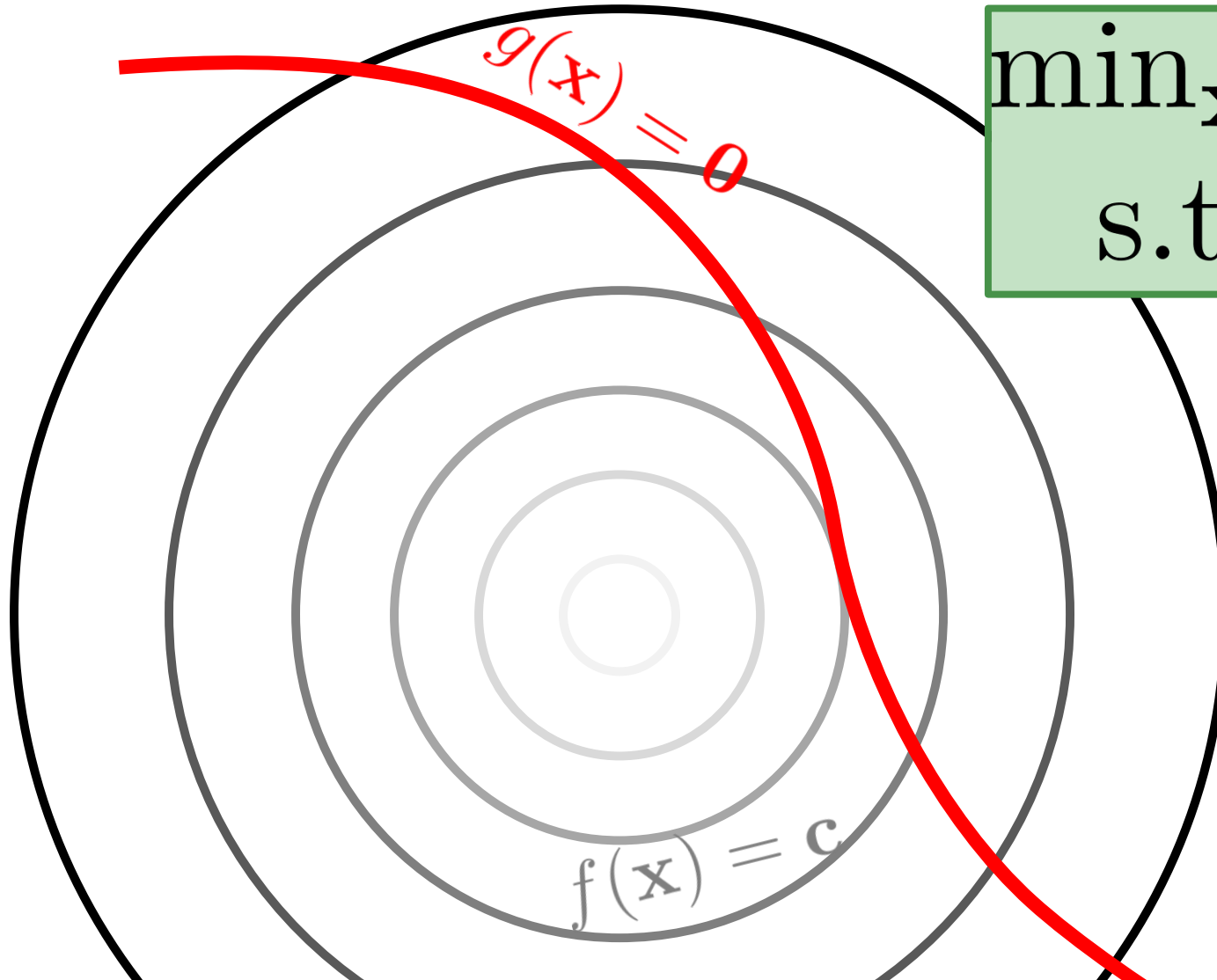
$$\nabla f(\mathbf{x}) = 0$$

(unconstrained)



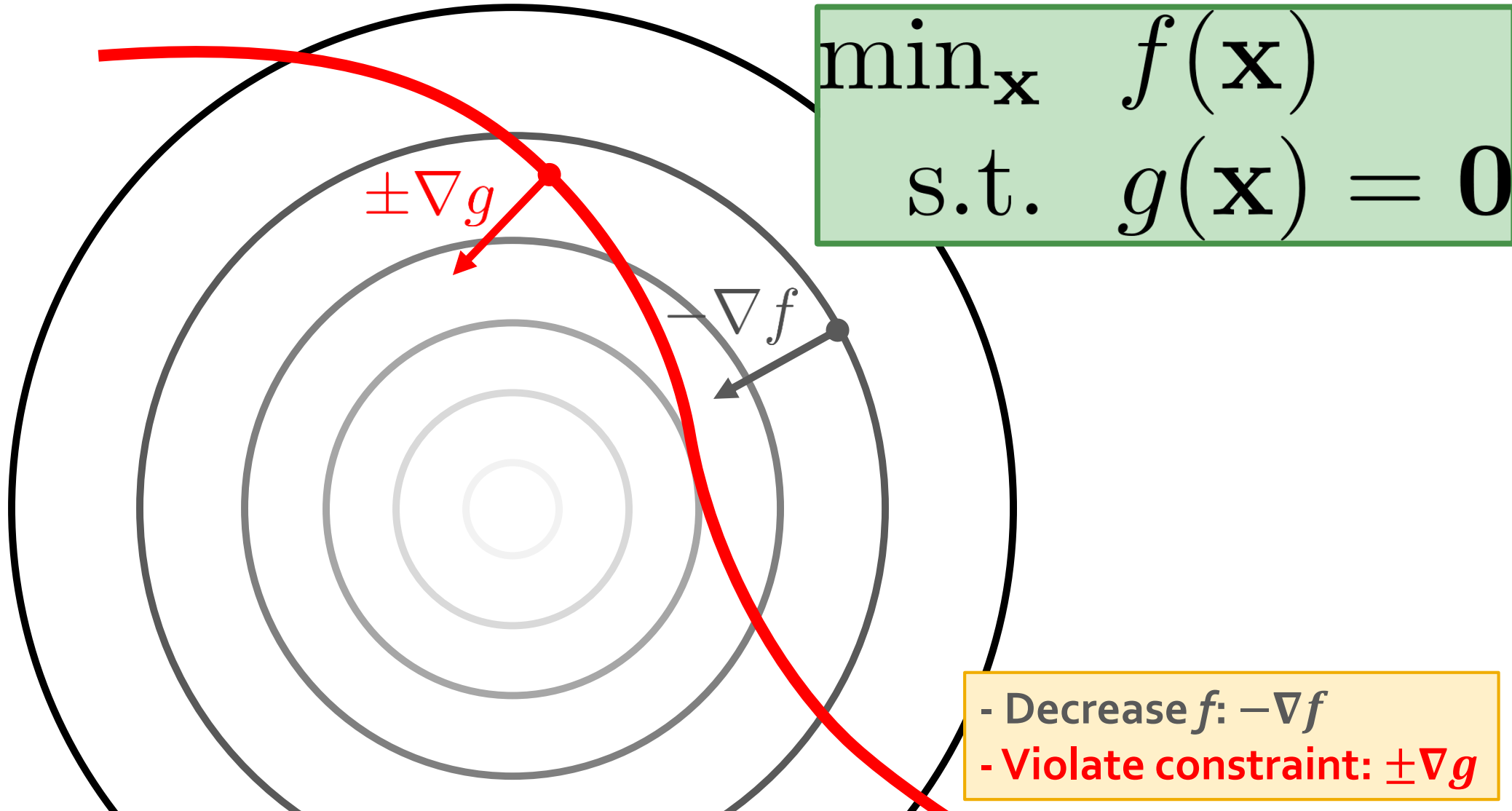
Critical point

Lagrange Multipliers: Idea



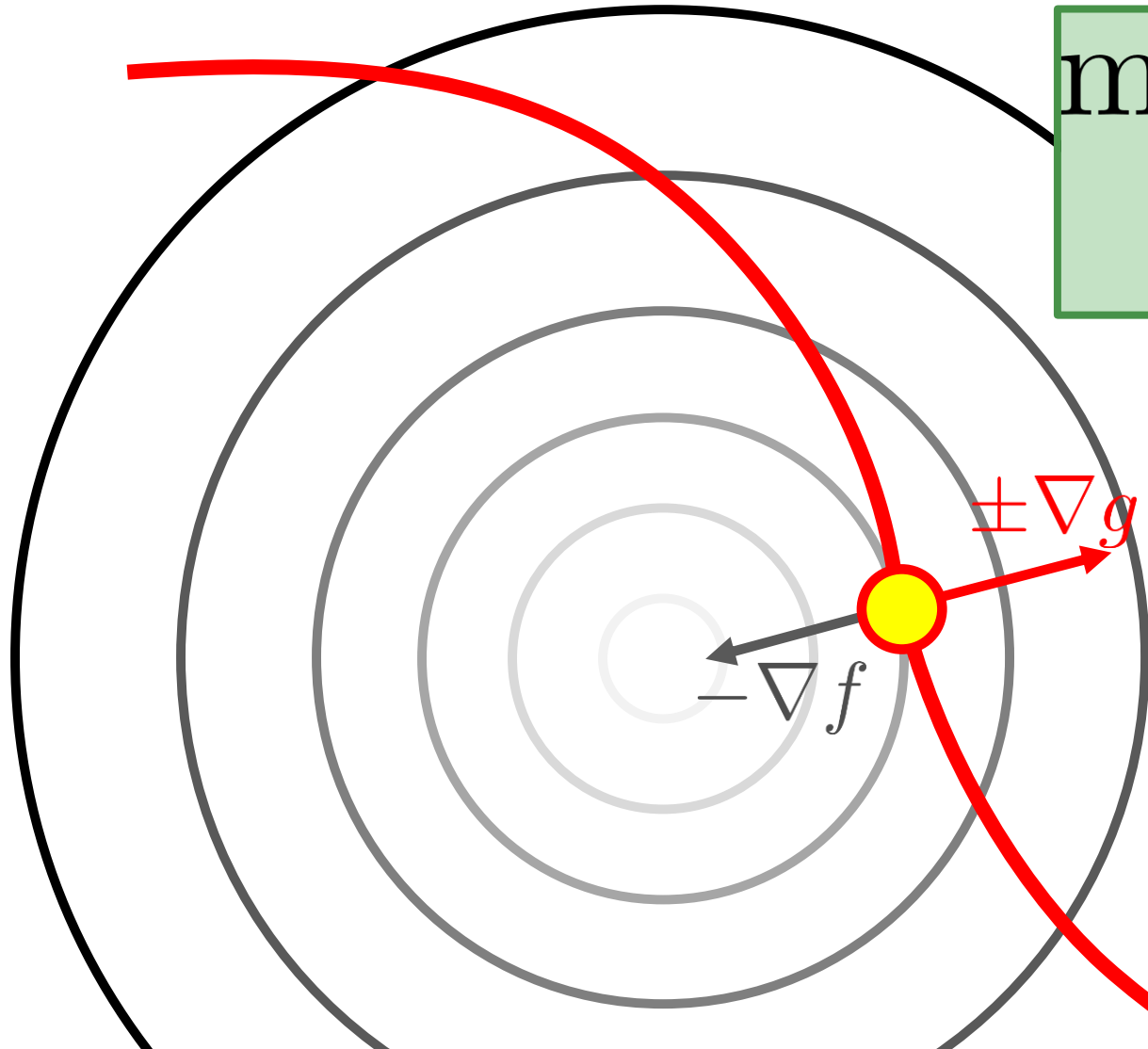
$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g(\mathbf{x}) = \mathbf{0} \end{aligned}$$

Lagrange Multipliers: Idea



Lagrange Multipliers: Idea

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



Want:

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

Use of Lagrange Multipliers

Turns constrained optimization into
unconstrained root-finding.

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

$$g(x) = 0$$

Example: Symmetric Eigenvectors

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

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

$$\implies Ax = \lambda x$$

(New for 2023!) See Course Notes For Details

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

Advanced Topic: Variational Calculus

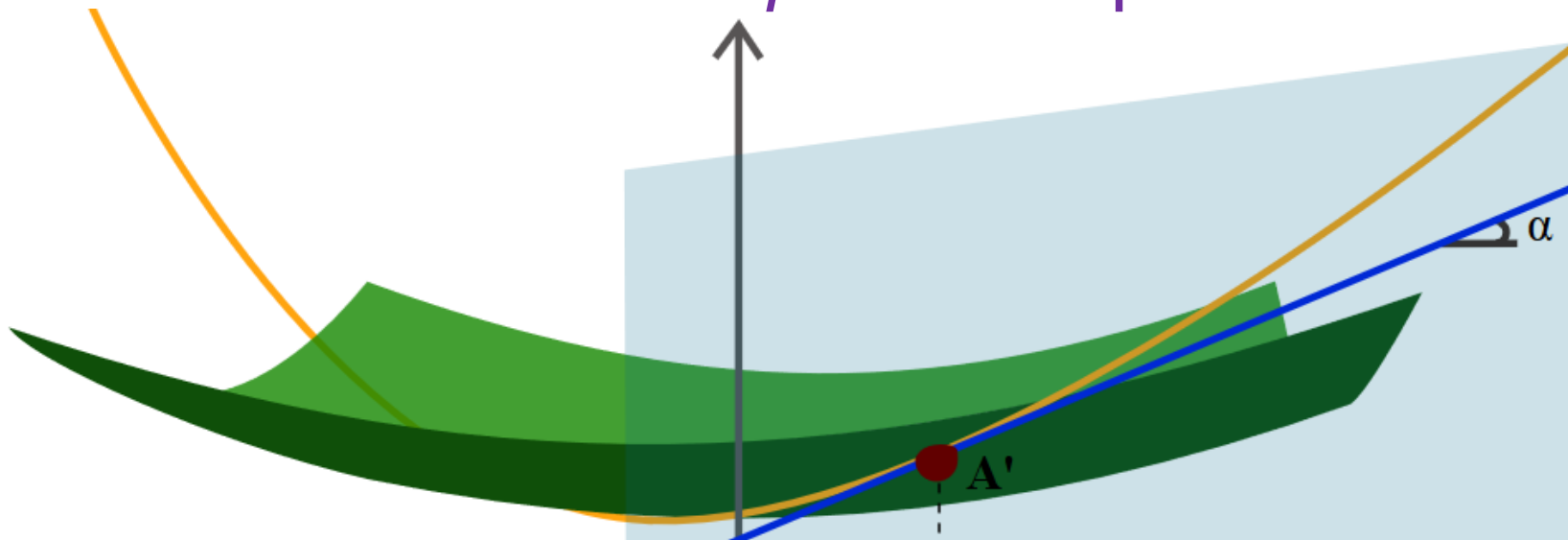
Sometimes your unknowns
are not numbers!

Can we use calculus to optimize anyway?

Gâteaux Derivative

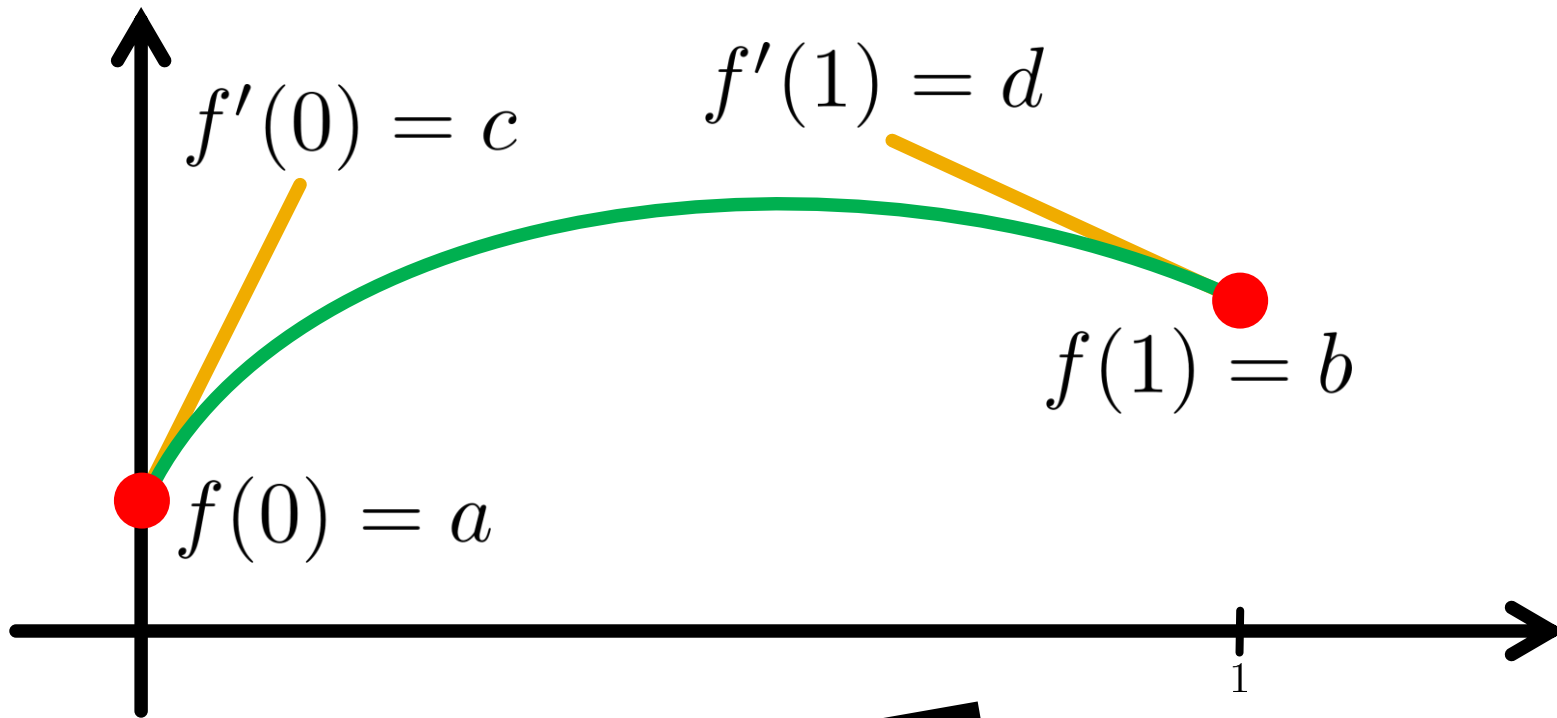
$$d\mathcal{F}[u; \psi] := \frac{d}{dh} \mathcal{F}[u + h\psi] \Big|_{h=0}$$

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

Spring 2023

