IFT 6113 BACKGROUND: LINEAR ALGEBRA AND OPTIMIZATION

http://tiny.cc/ift6113



Mikhail Bessmeltsev

Some slides from Justin Solomon

ROUGH PLAN

- (intro) Matrices and Eigenvalues
- Linear problems
- Unconstrained optimization
- Equality-constrained optimization
- Variational problems

ON THE BOARD

- Vectors
- inner products and norms
- affine and vector spaces
- function spaces

How are those matrices special?

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 5 \end{pmatrix} \qquad \qquad \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 5 \end{pmatrix} \qquad \qquad \begin{pmatrix} 0 & -1 & -2 \\ 1 & 0 & 4 \\ 2 & -4 & 0 \end{pmatrix}$$

$$\begin{pmatrix} \cos(0.4) & -\sin(0.4) & 0\\ \sin(0.4) & \cos(0.4) & 0\\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 3\\ 0 & 1 & 4\\ 0 & 0 & 1 \end{pmatrix}$$

How are those matrices special?



What is the geometric meaning of those?

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 5 \end{pmatrix}$$
Diagonal

$$\begin{pmatrix} \cos(0.4) & -\sin(0.4) & 0\\ \sin(0.4) & \cos(0.4) & 0\\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 3\\ 0 & 1 & 4\\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{array}{c} \text{Orthogonal} \\ A^{-1} = A^T \end{array} \qquad \text{Upper triangular}$$

INTRO TO INTRO

What is the geometric meaning of those?

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 5 \end{pmatrix}$$

Non-uniform scaling

$$\begin{pmatrix} \cos(0.4) & -\sin(0.4) & 0\\ \sin(0.4) & \cos(0.4) & 0\\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 3\\ 0 & 1 & 4\\ 0 & 0 & 1 \end{pmatrix}$$
Rotation Translation in 2D*

NO IDEA WHAT THOSE ARE?

FOUNDATIONS OF





https://umontreal.on.worldcat.org/oclc/829434522 Free online at UdeM library Chapters **I.3, I.4**

EIGENVALUES AND EIGENVECTORS

 $Ax = \lambda x$

EIGENVECTORS/VALUES

Geometric meaning? $Ax = \lambda x$ Vectors which **only** get scaled



DIAGONAL MATRICES

- Eigenvalues on the diagonal
- Scaling along coordinate axes = eigenvectors
- $\Theta(n)$ space to store
- O(n) time to invert

$$\begin{pmatrix} 1 & & \\ & -2 & \\ & & 5 \end{pmatrix}$$
Diagonal

SYMMETRIC MATRICES

- All eigenvalues are real
- All eigenvectors are orthogonal

Can be diagonalized by an orthogonal matrix: $D = Q^T A Q$

$$\begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 5 \end{pmatrix}$$

Symmetric
$$A = A^{T}$$

ORTHOGONAL MATRICES

- $AA^T = I$
- Columns have zero dot product
- Doesn't change length of vectors or angles between them (*isometry*)
- Determinant is ± 1

$$\begin{pmatrix} \cos(0.4) & -\sin(0.4) & 0\\ \sin(0.4) & \cos(0.4) & 0\\ 0 & 0 & 1 \end{pmatrix}$$

Orthogonal
 $A^{-1} = A^T$

- Geometric transformations
- Linear operators on (discrete) functions

MATRIX IS POSITIVE SEMIDEFINITE

 $A \geq 0$

All eigenvalues are nonnegative

Geometric meaning?

ROUGH PLAN

- (intro) Matrices and Eigenvalues
- Linear problems
- Unconstrained optimization
- Equality-constrained optimization
- Variational problems



MOTIVATION

Numerical problems are everywhere in geometric modeling!

ROUGH PLAN

- (intro) Matrices and Eigenvalues
- Linear problems
- Unconstrained optimization
- Equality-constrained optimization
- Variational problems

OUR BIAS



EXAMPLES

How to flatten a mesh?



"Mesh Parameterization: Theory and Practice" by Kai Hormann, Bruno Lévy, Alla Sheffer



EXAMPLES

How to flatten a mesh?





"Boundary First Flattening" by Rohan Sawhney and Keenan Crane

EXAMPLES

How to animate a character?



"Linear Subspace Design for Real-Time Shape Deformation" by Yu Wang, Alec Jacobson, Jernej Barbič, Ladislav Kavan

EXAMPLES

How to animate a character?

 $\min_{x_i, y_i \in \mathbb{R}^2} f(x_1, y_1, \dots, x_n, y_n)$ s.t. $x_i - x'_i = 0$ $j \in J$ $y_j - y_i' = 0$ Known positions Triangle Some distortion? smoothness?



Objective function (**Energy**)

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

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



Inequality constraints

EXAMPLES

$Ax = b \quad \leftrightarrow \quad \min_{x \in \mathbb{R}^n} ||Ax - b||_2^2$

EXAMPLES $\begin{array}{l} \underset{x \in \mathbb{R}^{n}}{\min} \|Ax\|_{2}^{2} \\ Ax = \lambda x \quad \leftrightarrow \\ \text{ s.t. } \|x\| - 1 = 0 \end{array}$

VECTOR SPACES AND LINEAR OPERATORS

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

EIGENVALUES AND EIGENVECTORS

$$Ax = \lambda x$$

Discrete (matrix)

or

 $\mathcal{L}[f(x)] = \lambda f(x)$

Continuous (operator)

ABSTRACT EXAMPLE

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

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

Eigenvectors?

IN FINITE DIMENSIONS




LINEAR SYSTEM OF EQUATIONS



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.

SIMPLE EXAMPLE

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

Discretization?

On the board

SIMPLE EXAMPLE

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



STRUCTURE?



LINEAR SOLVER CONSIDERATIONS

- Never construct A⁻¹ explicitly (if you can avoid it)
- Added structure helps <u>Sparsity</u>, symmetry, positive definiteness, bandedness

$inv(A)*b \ll (A'*A) \setminus (A'*b) \ll A \setminus b$

LINEAR SYSTEMS: SOLVERS

• **Direct** (explicit matrix)

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

GENERIC ADVICE

Generic tools are often not too effective!

GENERIC ADVICE

Try the simplest solver first.

VERY COMMON: SPARSITY



FOR IFT 6113

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

Eigen 3.3.5			(Q: Search
Overview Getting started Chapters	Sparse matrix manipulations Sparse linear algebra		
 Dense matrix and array manipulation Dense linear problems and decompositio Sparse linear algebra 	Manipulating and solving sparse problems involves various modules which are summarized below:		
Sparse matrix manipulations	Module	Header file	Contents
Solving Sparse Linear Systems Matrix-free solvers	SparseCore	#include ≮Eigen/SparseCore>	SparseMatrix and SparseVector classes, matrix assembly, basic sparse linear algebra (including sparse triangular solvers)
Quick reference guide for sparse matr	SparseCholesky	<pre>#include <eigen sparsecholesky=""></eigen></pre>	Direct sparse LLT and LDLT Cholesky factorization to solve sparse self-adjoint positive definite problems
Geometry	SparseLU	<pre>#include<eigen sparselu=""></eigen></pre>	Sparse LU factorization to solve general square sparse systems
 Extending/Customizing Eigen Concernent tension 	SparseQR	<pre>#include<eigen sparseqr=""></eigen></pre>	Sparse QR factorization for solving sparse linear least-squares problems
 Class List 	IterativeLinearSolvers	<pre>#include <eigen iterativelinearsolvers=""></eigen></pre>	Iterative solvers to solve large general linear square problems (including self-adjoint positive definite problems)
	Sparse	<pre>#include <eigen sparse=""></eigen></pre>	Includes all the above modules
Table of contents ψ Sparse matrix format ψ First example ψ The SparseMatrix class	Sparse matrix format In many applications (e.g., finite element methods) it is common to deal with very large matrices where only a few coefficients are different from zero. In such cases, memory consumption can be reduced and performance increased by using a specialized representation storing only the nonzero coefficients. Such a matrix is called a sparse matrix. The SnarseMatrix class		
 ↓ Filling a sparse matrix ↓ Supported operators and functions ↓ Basic operations ↓ Matrix products ↓ Elock operations 	The class SparseMatrix is the main sparse matrix representation of Eigen's sparse module; it offers high performance and low memory usage. It implements a more versatile variant of the widely-used Compressed Column (or Row) Storage scheme. It consists of four compact arrays:		
mangular anu senaujumi views	- Values Leteres the	coefficient values of the non-zeros	Constrated on Man Aug 27 2019 05 50-20 for Elegan by CONTRACT 1 9 12

https://eigen.tuxfamily.org/dox/group__TutorialSparse.html

ROUGH PLAN

- (intro) Matrices and Eigenvalues
- Linear problems
- Unconstrained optimization
- Equality-constrained optimization
- Variational problems

UNCONSTRAINED OPTIMIZATION



Trivial when f(x) is linear

Easy when f(x) is quadratic

Hard in case of generic non-linear



$$f(x, y) = 3x^{2} - y$$
$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right) = (6x, -1)$$

Geometric meaning?





Gradient

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

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



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





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

Jacobian

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



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

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



OPTIMIZATION TO ROOT-FINDING $\nabla f(x) = 0$ (unconstrained) Saddle point Local max f(x)Local min ► X

Critical point



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

CONVEX FUNCTIONS $H(x) \ge 0$



Aside: Convex Optimization Tools



Try lightweight options

SPECIAL CASE: LEAST-SQUARES

 $\min_{x} \frac{1}{2} \|Ax - b\|_{2}^{2}$

 $\rightarrow \min_{x} \frac{1}{2} x^{\top} A^{\top} A x - b^{\top} A x + \|b\|_{2}^{2}$

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

Normal equations (better solvers for this case!)

USEFUL DOCUMENT

The Matrix Cookbook Petersen and Pedersen

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

UNCONSTRAINED OPTIMIZATION

Unstructured.

BASIC ALGORITHMS



Gradient descent

BASIC ALGORITHMS

$$egin{aligned} &\lambda_0 = 0, \lambda_s = rac{1}{2}(1 + \sqrt{1 + 4\lambda_{s-1}^2}), \gamma_s = rac{1-\lambda_2}{\lambda_{s+1}} \ &y_{s+1} = x_s - rac{1}{eta}
abla f(x_s) \ &x_{s+1} = (1 - \gamma_s) y_{s+1} + \gamma_s y_s \end{aligned}$$

Quadratic convergence on convex problems! (Nesterov 1983)

A very cool intro: https://distill.pub/2017/momentum/

Accelerated gradient descent

BASIC ALGORITHMS

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



Newton's Method

BASIC ALGORITHMS $x_{k+1} = x_k - M_k^{-1} \nabla f(x_k)$ Hessian approximation

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

Quasi-Newton: BFGS and

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.



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.

 x_1

Sum

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

SOFTWARE

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

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



SOME TRICKS

Lots of small elements: $||x||_2^2 = \sum_i x_i^2$ Lots of zeros: $||x||_1 = \sum_i |x_i|$ Uniform norm: $||x||_{\infty} = \max_i |x_i|$ Low rank: $||X||_* = \sum_i \sigma_i$ Mostly zero columns: $||X||_{2,1} = \sum_j \sqrt{\sum_i x_{ij}^2}$ Smooth: $\int \|\nabla f\|_2^2$ Piecewise constant: $\int \|\nabla f\|_2$???: Early stopping Regularization

SOME TRICKS





Original

Blurred

Multiscale/graduated optimization

ROUGH PLAN

- (intro) Matrices and Eigenvalues
- Linear problems
- Unconstrained optimization
- Equality-constrained optimization
- Variational problems
LAGRANGE MULTIPLIERS: IDEA



LAGRANGE MULTIPLIERS: IDEA



LAGRANGE MULTIPLIERS: IDEA



USE OF LAGRANGE MULTIPLIERS

Turns constrained optimization into unconstrained root-finding.

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

Find the maximum and minimum of f(x, y) = 5x - 3y subject to the constraint $x^2 + y^2 = 136$.

Find the maximum and minimum of f(x, y) = 5x - 3y subject to the constraint $x^2 + y^2 = 136$.

$$\nabla f = (5, -3) = \lambda \nabla g = \lambda (2x, 2y)$$
$$x^2 + y^2 = 136$$

Problem credit: https://tutorial.math.lamar.edu/classes/calciii/lagrangemultipliers.aspx

Find the maximum and minimum of f(x, y) = 5x - 3y subject to the constraint $x^2 + y^2 = 136$.

$$\nabla f = (5, -3) = \lambda \nabla g = \lambda (2x, 2y)$$
$$x^{2} + y^{2} = 136$$
(arithmetic)

$$\lambda = \pm \frac{1}{4}, x = \pm 10, y = \pm 6$$

Problem credit: https://tutorial.math.lamar.edu/classes/calciii/lagrangemultipliers.aspx

LAGRANGIAN

Equivalently, you can form a Lagrangian: $\mathcal{L}(x,\lambda) = f + \lambda g$ Then find its stationary points: $\frac{\partial \mathcal{L}}{\partial x} = \nabla f + \lambda \nabla g = 0$ $\frac{\partial \mathcal{L}}{\partial \lambda} = g = 0$

Find the maximum and minimum of f(x, y) = 5x - 3ysubject to the constraint $x^2 + y^2 = 136$.

$$\mathcal{L}(x,\lambda) = 5x - 3y + \lambda(x^2 + y^2 - 136)$$

Then find its stationary points => solve a system of eqns:

$$\frac{\partial \mathcal{L}}{\partial x} = 5 + 2\lambda x = 0$$
$$\frac{\partial \mathcal{L}}{\partial y} = -3 + 2\lambda y = 0$$
$$\frac{\partial \mathcal{L}}{\partial \lambda} = x^2 + y^2 - 136 = 0$$



QUADRATIC WITH LINEAR EQUALITY

$$\min_{x} \quad \frac{1}{2}x^{\top}Ax - b^{\top}x + c \\ \text{s.t.} \quad Mx = v$$

(assume A is symmetric and positive definite)

QUADRATIC WITH LINEAR EQUALITY





MANY OPTIONS

• Reparameterization Eliminate constraints to reduce to unconstrained case

Newton's method

Approximation: quadratic function with linear constraint

Penalty method

Augment objective with barrier term, e.g. $f(x) + \rho |g(x)|$

EXAMPLE: SYMMETRIC EIGENVECTORS

 $f(x) = x^{\top}Ax \implies \nabla f(x) = 2Ax$ $g(x) = \|x\|_2^2 \implies \nabla g(x) = 2x$ $\implies Ax = \lambda x$

EXAMPLE: MESH EMBEDDING



G. Peyré, mesh processing course slides

LINEAR SOLVE FOR EMBEDDING

$$x_i \in \mathbb{R}^2$$

 $\min_{\substack{x_1, \dots, x_{|V|} \\ \text{s.t.} }} \sum_{\substack{(i,j) \in E \\ v \in V_0}} w_{ij} \|x_i - x_j\|_2^2$

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

Assumption: w symmetric.

$$\min_{\substack{x_1, \dots, x_{|V|} \\ \text{s.t.} \quad x_v \text{ fixed } \forall v \in V_0}} \sum_{ij \in E} w_{ij} \|x_i - x_j\|_2^2$$

NONTRIVIALITY CONSTRAINT

$$\left\{\begin{array}{cc} \min_{x} & \|Ax\|_{2} \\ \text{s.t.} & \|x\|_{2} = 1 \end{array}\right\} \mapsto A^{\top}Ax = \lambda x$$

Prevents trivial solution $x \equiv 0$.

Extract the smallest eigenvalue.



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

$$\min_{\substack{u\\ u^{\top}Be=0 \\ u^{\top}Bu=1}} u^{\top}L_{C}u \quad \longleftrightarrow \quad L_{c}u = \lambda Bu$$

BASIC IDEA OF EIGENALGORITHMS

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

TRUST REGION METHODS



Example: Levenberg-Marquardt

EXAMPLE: POLYCUBE MAPS



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

$$\begin{aligned} & \underset{X \in \mathcal{A}}{\text{Mign with coordinate axes}} \\ & \underset{X \in \mathcal{A}}{\min_{X} \sum_{b_{i}}} \quad \mathcal{A}(b_{i};X) \| n(b_{i};X) \|_{1} \\ & \text{s.t.} \quad \sum_{b_{i}} \mathcal{A}(b_{i};X) = \sum_{b_{i}} \mathcal{A}(b_{i};X_{0}) \end{aligned}$$

Note: Final method includes more terms!

Preserve area

ITERATIVELY REWEIGHTED LEAST SQUARES

$$\min_{x} \sum_{i} \phi(x^{\top}a_i + b_i) \leftrightarrow \begin{cases} \min_{x, y_i} & \sum_{i} y_i (x^{\top}a_i + b_i)^2 \\ \text{s.t.} & y_i = \phi(x^{\top}a_i + b_i) (x^{\top}a_i + b_i)^{-2} \end{cases}$$



Repeatedly solve linear systems

ALTERNATING PROJECTION



AUGMENTED LAGRANGIANS

$$\begin{array}{ccc} \min_{x} & f(x) \\ \text{s.t.} & g(x) = 0 \\ & \downarrow \\ \min_{x} & f(x) + \frac{\rho}{2} \|g(x)\|_{2}^{2} \end{array} & \xrightarrow{\text{Does nothing when } \\ \text{constraint is satisfield} \\ \text{s.t.} & g(x) = 0 \end{array}$$

Add constraint to objective

nstraint is satisfied

ALTERNATING DIRECTION METHOD OF MULTIPLIERS (ADMM) $\min_{x,z} f(x) + g(z)$ s.t. Ax + Bz = c

 $\Lambda_{\rho}(x,z;\lambda) = f(x) + g(z) + \lambda^{\top}(Ax + Bz - c) + \frac{\rho}{2} \|Ax + Bz - c\|_{2}^{2}$

$$\begin{aligned} x \leftarrow \arg \min_{x} \Lambda_{\rho}(x, z, \lambda) \\ z \leftarrow \arg \min_{z} \Lambda_{\rho}(x, z, \lambda) \\ \lambda \leftarrow \lambda + \rho(Ax + Bz - c) \end{aligned}$$

https://web.stanford.edu/~boyd/papers/pdf/admm_slides.pdf

FRANK-WOLFE





https://en.wikipedia.org/wiki/Frank%E2%80%93Wolfe_algorithm

Linearize objective, preserve constraints

ROUGH PLAN

- (intro) Matrices and Eigenvalues
- Linear problems
- Unconstrained optimization
- Equality-constrained optimization
- Variational problems

VARIATIONAL CALCULUS: BIG IDEA

Sometimes your unknowns are not numbers!

Can we use calculus to optimize anyway?

ON THE BOARD

 $\min_{f} \int_{\Omega} \|\vec{v}(x) - \nabla f(x)\|_{2}^{2} d\vec{x}$

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

GATEAUX 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



"I got a great reputation for doing integrals, only because my box of tools was different from everybody else's, and they had tried all their tools on it before giving the problem to me."

Richard P. Feynman