Scientific Computing
Lecture 6

Optimization

Nikolay Koshev

October 19, 2021

Skoltech

Skolkovo Institute of Science and Technology

Deconvolution problem

The 2D deconvolution problem is the problem of finding $u(\mathbf{x})$ from the equation:

$$Au(\mathbf{x}) \equiv \int_{\mathbb{R}^2} K(\mathbf{x} - \xi) u(\xi) d\xi = f(\mathbf{x}).$$

In case of finite u and K, the latter equation is ill-posed. Indeed, due to convolution theorem:

$$u(\mathbf{x}) = \int_{\mathbb{R}^2} e^{i\omega \cdot \mathbf{x}} \frac{\hat{u}(\omega)}{\hat{k}(\omega)}.$$

The integration in the latter equation is provided for the whole space; the frequencies due to noise, however, may be very high, and we are not able to state the integral will converge.

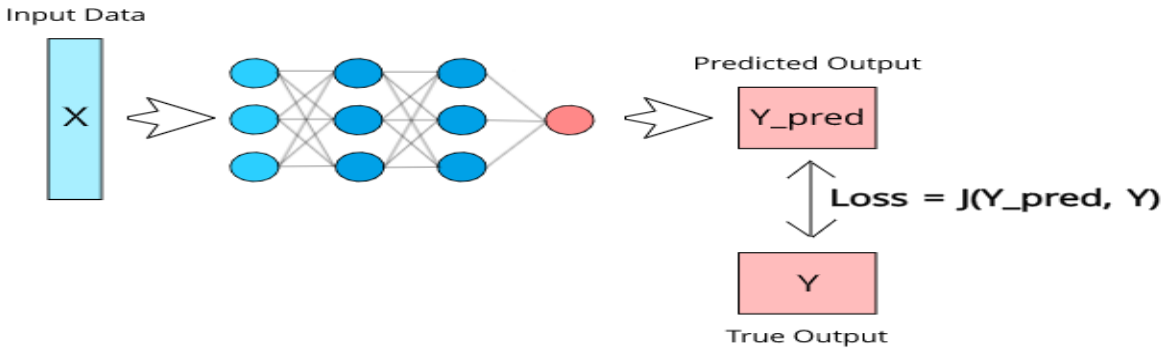
Deconvolution: regularization

The cost Tikhonov's functional:

$$M_\alpha[u] = ||Au - f||^2_{L_2(B)} + \alpha\Omega[u],$$

where the functional $\Omega[u]$ is a stabilizer.
To find an approximate solution, we need to **minimize** the functional above.

NN: Loss



- X - input
- Y - output (Ground True)
- $Y_{pred} \equiv Y_{pred}(\mathbf{w})$ - predicted output
- $\mathbf{w}$ - vector of all model weights

$$Loss(\mathbf{w}) = Error(Y_{pred}, Y); \quad \text{MSE Loss} = ||Y_{pred} - Y||^2_{L^2}$$

To restore the **the model** A, we need to **minimize** the loss function with respect to weights vector $\mathbf{w}$

Skoltech
Skolkovo Institute of Science and Technology

Galerkin approach

- Represent (or approximate) the solution u with some weighted sum of functions
- Substitute the approximation or representation into the original equations
- **Minimize** the residual with respect to weights
- After the weights are calculated, reconstruct the approximate (or, sometimes, exact) solution, substituting the weights into your representation (or approximation) of it
- Be careful: the residual rarely being equal to zero, but it should be small!

General problem statement

- H is some Hilbert space.
- Let $f : H \to \mathbb{R}$.
- **Optimization** or **Programming:**

$$u^* = \operatorname*{argmin}_{u \in U} f(u), \quad u^* = ?$$

- **Linear Programming:** The function f is linear.
- **Nonlinear Programming:** The function f is non-linear.

Applications

- Optimal transport regulation
- Time-based distribution of the products from warehouses to stores
- Spatial optimal distribution of electronics within spacecraft or submarines
- Industrial optimization
- Information flows
- Neural networks and machine learning.

Applications from the point of view of mathematics

- Solution of large systems of algebraic equations (especially, ill-conditioned systems)
- Galerkin: Research of integral and differential equations. Minimization of the residual after application of the original operator to the approximated solution.
- Cost functionals: Solving integral and differential equations. Construction of the cost functional for the problem and providing its constrained or unconstrained optimization.
- Ill-posed problems: Solution of complex problems using certain functional spaces.
- Minimization of the coefficients within an artificial neural network.

Criteria and notations

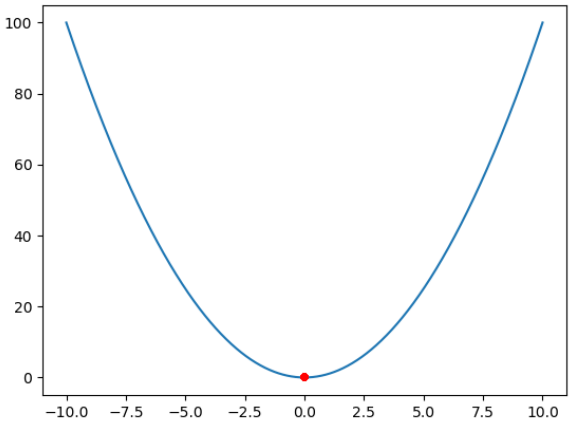
- $f(u) : H \to \mathbb{R}$ - the function to be minimized
- $u \in U \subseteq H$, U is the constrained set.
- **Unconstrained programming:** $U = H$.
- **Constrained programming:** $U \subset H$.
- $u^* = \underset{u \in U}{\operatorname{argmin}} f(u)$ - the minimizer.
- $f^* = f(u^*) = \min\limits_{u \in U} f(u)$ - the minimum.
- $U^* \subset U = \underset{u \in U}{\operatorname{Argmin}} f(u)$ - the set of minimizers.

Skoltech
Skolkovo Institute of Science and Technology

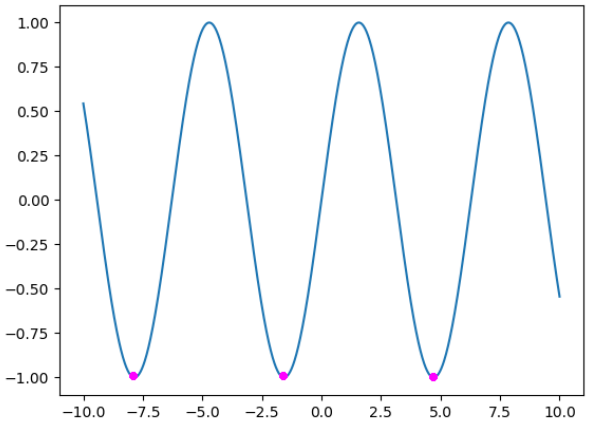
Local and Global minima

- u^* is a **Global Minimizer** if $\forall u \in U \quad f(u) \geq f(u^*)$.
- u^* is a **Strict Global Minimizer** if $\forall u \in u \quad f(u) > f(u^*)$.
- Set $\bar{S}_r(u^*) = \{u \in H : ||u - u^*|| \leq r$ - a ball with the radius r.
- u^* is a **Local Minimizer** if
  $\exists r > 0 : f(u) \geq f(u^*) \quad \forall u \in U \cap \bar{S}_r(u^*)$
- u^* is a **Strict Local Minimizer** if
  $\exists r > 0 : f(u) > f(u^*) \quad \forall u \in U \cap \bar{S}_r(u^*)$

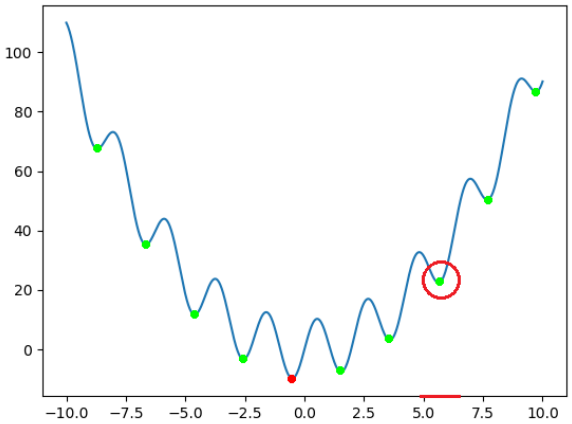
Local and Global minima



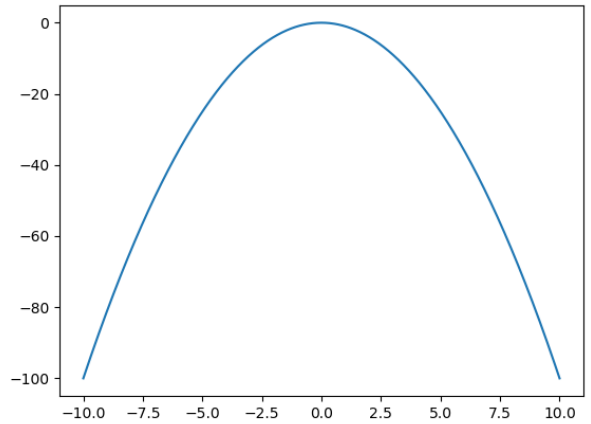
Global min.



Loc.min. only



Global & Local



No minima!

Skoltech
Skolkovo Institute of Science and Technology

Differentiable functionals

- The functional $f(u)$ is **differentiable** at the point u_0 if

$$f(u) = f(u_0) + \left(f'(u_0), u - u_0\right)_H + o\left(||u - u_0||_H\right)$$

- The operator f' is a **Strong derivative** (so-called the **Frechet derivative** or **The Gradient**).
- **Necessary condition of extremum:** Let the functional f to be differentiable at the point $u^* \in U$.
  If u^* is a local minimizer, than $f'(u^*) = 0$.

Skoltech
Skolkovo Institute of Science and Technology

Differentiable functionals

- If the functional $f : H \to \mathbb{R}$, then $f'(u) \in H$.
- For example, let

$$f(\mathbf{x}) = ax_1^2 + bx_2^3; \quad \mathbf{x} = (x_1, x_2)^T \in \mathbb{R}^2$$

- $f : \mathbb{R}^2 \to \mathbb{R}$
-

$$f'(\mathbf{x}) = \left(2ax_1, 3bx_2^2\right)^T \in \mathbb{R}^2$$

- We can say the same about the higher-order Frechet derivatives.

Differentiable functionals

- The functional $f(u)$ is **twice differentiable** at the point u_0 if

$$f(u) = f(u_0) + \left(f'(u_0), u - u_0\right)_H + \frac{1}{2}\left(f''(u_0)(u - u_0), u - u_0\right) + o(||u - u_0||_H^2)$$

- The linear functional $f''(u_0) : H \to H$ is a **Second Frechet Derivative (SFD)**
- In finite dimensional case, the SFD is being defined with the Hessian
- Linear bounded operator A is **positively definite** if $\forall h \in H : (Ah, h) > 0$.
- **Sufficient condition of extremum:** Let the functional f to be twice differentiable at the point $u^* \in U$.
  If $f'(u^*) = 0$ *and* $f''(u^*) > 0$ *(positively definite), then* u^* *is a local minimizer*

Skoltech
Skolkovo Institute of Science and Technology

Convex set

The set U is a **Convex set** if $\forall u_1, u_2 \in U$ the interval connecting these points belongs fully to the set U:

$$[u_1, u_2] = \{u \in U : u = u_1 + \lambda(u_2 - u_1), \lambda \in [0, 1]\}$$

Examples:

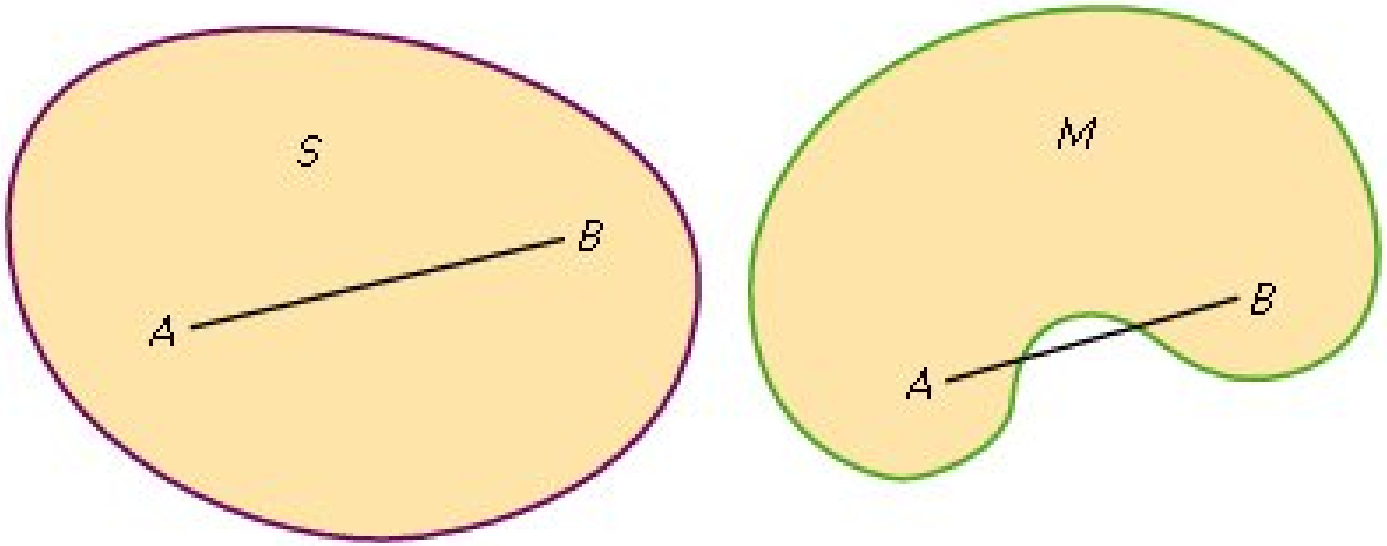
- The whole Hilbert space H
- Interval in $\mathbb{R}^1$
- Polyhedra in $\mathbb{R}^n$:

$$\{u \in \mathbb{R}^n : Au \leq b\} = \{u \in \mathbb{R}^n : (a_i, u) \leq b_i, i = 1, ..., n\}$$

  where a_i - vectors, b_i - numbers.
- The closed ball $\bar{S}_r(u_0)$

Skoltech
Skolkovo Institute of Science and Technology

Convex set



©1998 Encyclopaedia Britannica, Inc.

Convex functional

- The functional $f(u)$, defined on convex set is a **Convex** if $u_1, u_2 \in U$ and $\forall \lambda \in [0, 1]$:

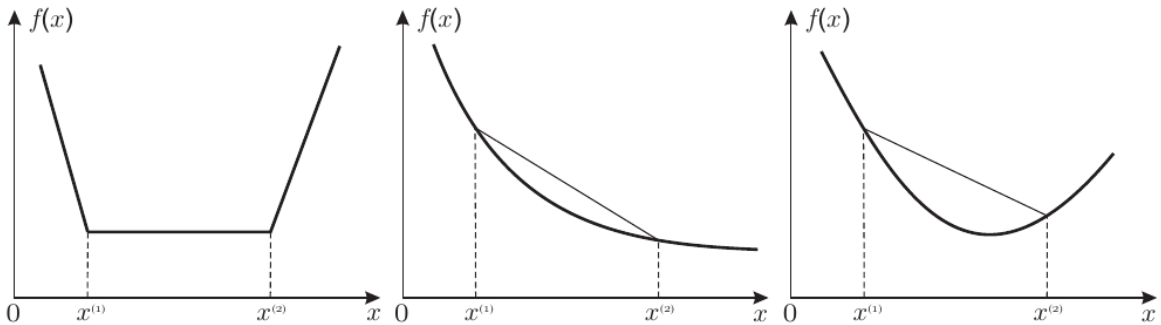
$$f(\lambda u_1 + (1 - \lambda)u_2) \leq \lambda f(u_1) + (1 - \lambda)f(u_2).$$

- If the inequality is strict, then the functional is **Strictly convex:**

$$f(\lambda u_1 + (1 - \lambda)u_2) < \lambda f(u_1) + (1 - \lambda)f(u_2).$$

- The functional is **Strongly Convex** if $\exists \theta > 0$ such that:

$$f(\lambda u_1 + (1 - \lambda)u_2) \leq \lambda f(u_1) + (1 - \lambda)f(u_2) - \theta\lambda(1 - \lambda)||u_1 - u_2||^2$$



(a) Convex functional, (b) - strictly convex functional, (c) - strongly convex functional

Skoltech
Skolkovo Institute of Science and Technology

Convex functionals

- $f(x) = x^2$ is strongly convex
- $f(x) = x^4$ is strictly convex but not strongly convex
- $f(x) = ||x||$ is convex but not strongly or strictly convex
- $f(x) = ||x||^2 = (x, x)$, defined on the whole Hilbert space, is strongly convex.
- **It will be an exam question**

Programming problems in terms of convexity

$$u^* = \operatorname*{argmin}_{u \in U} f(u) - ?$$

- **Convex programming problem:** U is a convex set, and f is a convex functional.
  In Convex Programming, any local minimizer is a global minimizer!
- **Linear programming problem:** $H = \mathbb{R}^n$, the functional is linear: $f(x) = (b, x) + c$
  Linear programming is much easier than nonlinear
- **Quadratic programming problem:** $H = \mathbb{R}^n$, and the function takes the form:

$$f(x) = \frac{1}{2}(Ax, x) + (b, x) + c,$$

where the matrix A is a symmetric matrix.
If the matrix A is nonnegative definite, then quadratic programming is a particular case of convex programming.

Skoltech
Skolkovo Institute of Science and Technology

Least Squares Fitting

System of Linear Algebraic Equations:

$$Ax = b, \quad x \in \mathbb{R}^n, b \in \mathbb{R}^m A : \mathbb{R}^n \to \mathbb{R}^m$$

The residual:

$$\Phi(x) = ||Ax - b||^2 = (A^*Ax, x) - 2(A^*b, x) + (b, b).$$

$$\Phi'(x) = 2(A^*Ax - A^*b),$$
$$\Phi''(x) = 2A^*A \geq 0.$$

Putting the gradient to zero, we obtain:

$$A^*Ax = A^*b$$

Pseudosolution

$$Ax = b \quad \Leftrightarrow \quad A^*Ax = A^*b$$

- In finite difference euclidian space the system $A^*Ax = A^*b$ have the solution for **any** b.
- The solution of this system is being called **Pseudosolution**
- If the original system has solution, then it coincides with the pseudosolution, and $\Phi(x) = \mu = 0$.
- If the original system is inconsistent ($\Phi(x) = \mu > 0$), then the pseudosolution with minimal norm is called **Normal Pseudosolution:**

$$x = \min_{x : A^*Ax = A^*b} ||x||$$

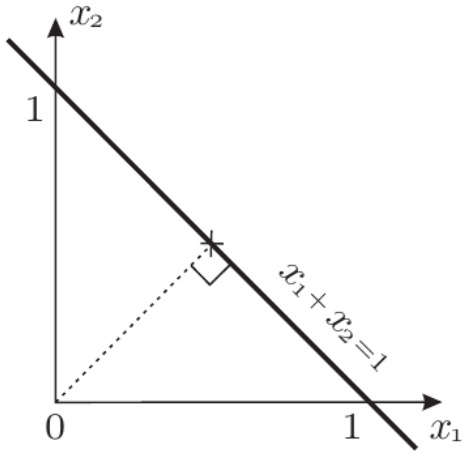
Skoltech
Skolkovo Institute of Science and Technology

Examples

- Consider the system:

$$\begin{cases} x_1 + x_2 = 1 \\ x_1 + x_2 = 1 \end{cases}$$

There are infinite number of solutions; the pseudonormal solution is $x_n = (1/2, 1/2)^T$. But it is unstable with respect to the error in matrix A.



- The system

$$\begin{cases} x_1 + x_2 = 1/2 \\ x_1 + x_2 = 3/2 \end{cases}$$

has no solutions at all; the pseudosolution, however, exists: $x_n = (1/2, 1/2)^T$ (unstable)

Skoltech
Skolkovo Institute of Science and Technology

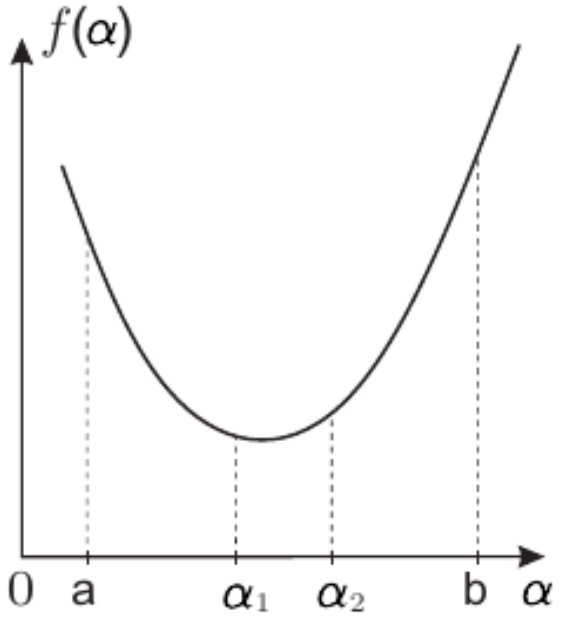
1D optimization: the Golden Section Method

$$x^* = \underset{x \in [a,b]}{\operatorname{argmin}} f(x).$$

1. Define the interval $[a, b]$ into three parts such that

$$\frac{b-\alpha_2}{b-a} = \frac{\alpha_1 - a}{b-a} = \xi \equiv \frac{2}{3+\sqrt{5}} \approx 0.38.$$

2. Compare $f(a), f(b), f(\alpha_1), f(\alpha_2)$. Obvious, that we can exclude the interval not being attached to a point where $f(x)$ is minimal.
3. Repeat steps 1 and 2 until the length of the interval will be bigger than certain $\varepsilon > 0$.



Skoltech
Skolkovo Institute of Science and Technology

1D optimization: the quadratic approximation

$$x^* = \underset{x \in [a,b]}{\operatorname{argmin}} f(x).$$

- Chose certain point $\alpha_1 \in [a, b]$ and calculate $f(a), f(b), f(\alpha_1)$
- Construct the parabola with these three points and find its minimizer α_{min}
- Denote $\alpha_2 = \alpha_{min}$.
- Throw out one of four points with, in which the value f is maximal
- Repeat the procedure.

Skoltech
Skolkovo Institute of Science and Technology

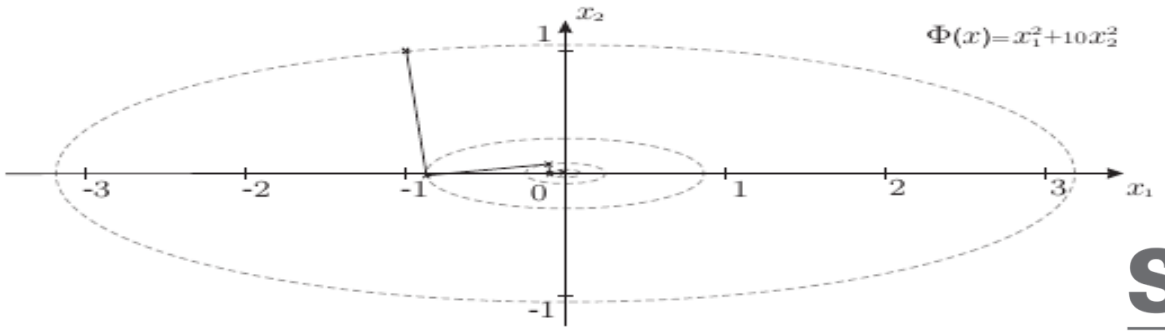
Zero order methods list

- Coordinate descent
- Hook & Jeeves method
- Nelder-Mead method
- Random-Walk methods
- Genetic Algorithms

Gradient descent

$$x^* = \underset{x \in \Omega}{\operatorname{argmin}} f(x), \quad \Omega \subseteq H$$

- Let the functional f be differentiable
- Chose initial approximation x_0 and calculate initial gradient $f'(x_0)$.
- The direction of optimization is the gradient: $g_0 = -f'(x_0)$
- $x_1 = x_0 + \alpha_0 g_0, \quad \alpha_0 = \underset{\alpha \geq 0}{\operatorname{argmin}} f(x_0 + \alpha g_n)$
- n^{th} iteration:
  $g_n = -f'(x_n); \alpha_n = \underset{\alpha \geq 0}{\operatorname{argmin}} f(x_n + \alpha g_n); x_{n+1} = x_n + \alpha_n g_n$
- Thus, the step α_n is the solution of 1D optimization problem.



Skoltech
Skolkovo Institute of Science and Technology

Image deblurring task

The problem:

$$Af \equiv K * f = u, \quad x \in B$$

The cost Tikhonov's functional:

$$M_\alpha[u] = ||Af - u||^2 + \alpha\Omega[f],$$

where the functional $\Omega[f]$ is a stabilizer.

Skoltech
Skolkovo Institute of Science and Technology

Quadratic optimization

Let $\Omega[f] = ||f||^2_{H^1}$. Then, due to Parseval's identity:

$$M[u] = ||\hat{K}\hat{f} - \hat{u}||^2_{L^2} + \alpha||\hat{f}||^2_{L^2} + \alpha||\frac{\partial \hat{f}}{\partial x}||^2_{L^2} + \alpha||\frac{\partial \hat{f}}{\partial x}||^2_{L^2}$$

Easy to see that this is a **quadratic programming problem**. The equation for minimizer:

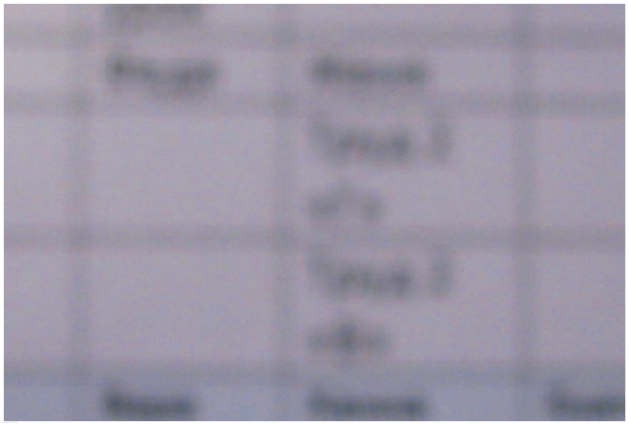
$$M'[u] \equiv 2(\hat{K}^*\hat{K}\hat{f} - \hat{K}^*\hat{u} + \alpha\hat{f}(1 + i\omega_1 + i\omega_2) = 0$$

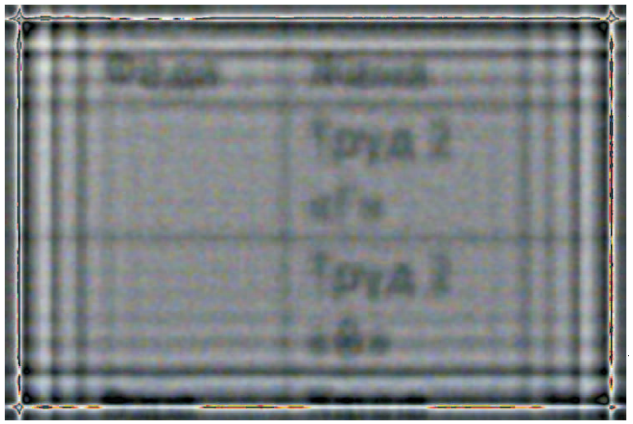
Thus,

$$f(\mathbf{x}) = \int\limits_{\mathbb{R}^2} d\omega_1 d\omega_2 e^{i\omega\cdot\mathbf{x}} \frac{\hat{K}^*\hat{u}}{|\hat{K}|^2 + \alpha(1 + i(\omega_1 + \omega_2))}$$

Skoltech
Skolkovo Institute of Science and Technology

Quadratic optimization





Conjugated gradients method

Let $\Omega[f] = ||f||_{HV}$ - the function with bounded total variation. The functional M is non-linear in this case. The norm in Total Variation functional space is given in the simplest case by:

$$VH(f) = \sup_S \sum_{i,j=0}^{N-2,M-2} |f_{i+1,j+1} - f_{ij+1} - f_{i+1j} + f_{ij}|.$$

The cost Tikhonov's functional with uniform $N \times N$ mesh takes the form:

$$M(\mathbf{f}) = \frac{1}{N^2} \sum_{i,j=0}^{N-1} \left(\frac{1}{N^2} \sum_{p,l=0}^{N-1} k_{i-p,j-l} f_{pl} - f_{ij} \right) + \alpha \sum_{i,j=0}^{N-2} |f_{i+1,j+1} - f_{ij+1} - f_{i+1j} + f_{ij}|.$$

The gradient of the functional can be obtained calculating the partial derivatives with respect to components of $\mathbf{f}$ array, after which we are able to use the conjugated gradients method in order to optimize the functional.

ConjGrad (SEM BE)

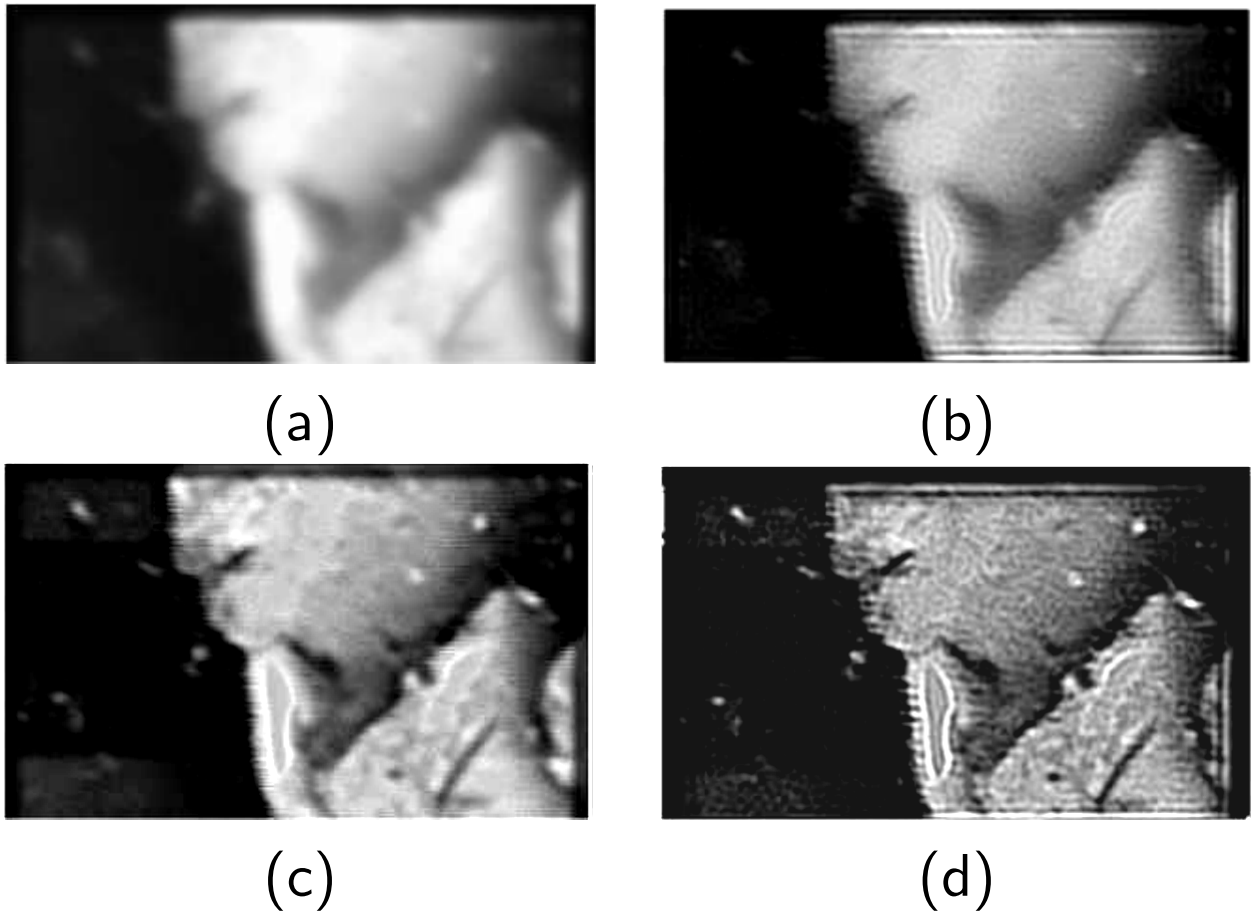
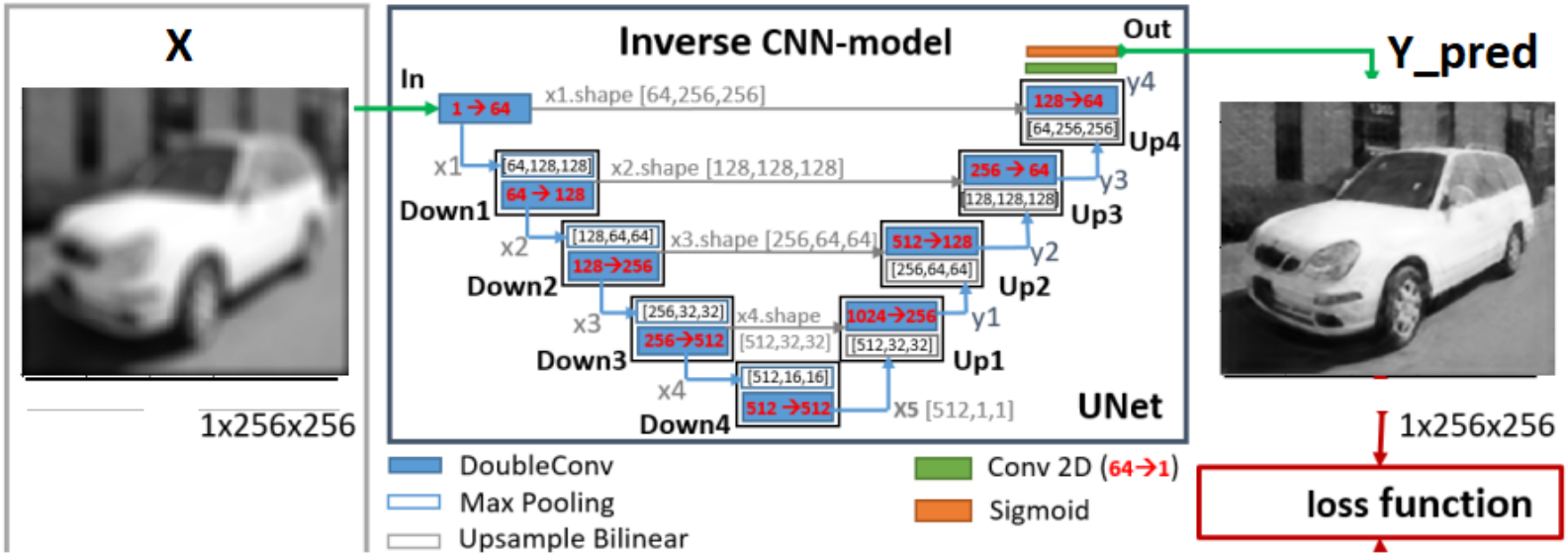


Figure: Results of image reconstruction in SEM BE: (a) - the enter data, (b) - reconstruction on H^1 Sobolev's space, analytical solution, (c) - reconstruction on TV Bounded Total Variations functional space, (MCGP, zero first approximation), (d) - reconstruction on TV with the result on H^1 taken as a first approximation for MCGP

Skoltech
Skolkovo Institute of Science and Technology

NN: Gradient Descent



L1 loss

BCE loss

MSE loss

Skoltech
Skolkovo Institute of Science and Technology

NN: Gradient Descent

Software

Optimization

- NLOPT (C++,Python)
- ALGIP (C++,C#)
- Dlib (C++)
- IPOPT (C++)
- SciPy (Python)

Differential equations solver

- FEniCS & DOLFIN (C++,Python)
- OpenFOAM

Software

Molecular dynamics

- Abalone
- AMBER
- Gromacs
- LAMMPS
- NAMD

FEM Meshing

- CGAL and CGAL-based products
- FEniCS
- HyperMesh
- Iso2Mesh (Matlab/Octave)
- FreeSurfer (Encephalography)

Skoltech courses: HPC

S.Rykovanov, A.Vishnyakov: Parallel Computing in Mathematical Modeling and Data-Intensive Applications

S.Rykovanov: High Performance Python Lab

S.Rykovanov: High Performance Computing and Modern Architectures

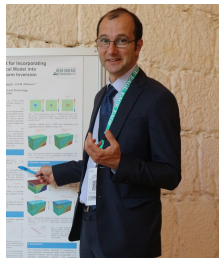
I.Zacharov: Introduction to Linux and Supercomputers

Numerical methods & Modeling

SLAE: I.Oseledets: Numerical Linear Algebra

Data Structures: G.Kucherov: Efficient Algorithms and Data Structures

PDE N.Yavich: Advanced Solvers for Numerical PDEs



Modeling: A.Shapeev: Numerical Modeling

Stochastic & Mesh-free

V.Palyulin: Stochastic Methods in Mathematical Modeling

N.Brilliantov: Foundations on Multiscale Modeling: Kinetics

ML & AI

M.Panov: Introduction to Data Science

E.Burnaev: Machine Learning

P.Popov: Machine Learning in Structural Bioinformatics and Chemoinformatics

D.Dylov: Computational Imaging

Skoltech
Skolkovo Institute of Science and Technology

Thank you for your attention!

Skoltech
Skolkovo Institute of Science and Technology