Scientific Computing
Lecture 6

Optimization

Nikolay Koshev

October 19, 2023

Skoltech

Skolkovo Institute of Science and Technology

Contents

- Motivation.
- What is optimization?
- Background: definitions, criteria, notations.
- Pseudosolution
- Gradient methods
  - Gradient descent
  - Gradient descent with moment
  - Heavy ball method
  - Conjugated gradients
  - Genetic algorithms
- Zero-order methods
- Regularization and image processing: example (Jupyter notebook)

Skoltech
Skolkovo Institute of Science and Technology

Optimization

Motivation

October 19, 2023

Skoltech

Skolkovo Institute of Science and Technology

Deconvolution problem

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

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

- In case of finite and noisy functions u and K, the latter equation is ill-posed.
- The problem can be reduced to the optimization problem for the cost Tikhonov's functional

Skoltech

Skolkovo Institute of Science and Technology

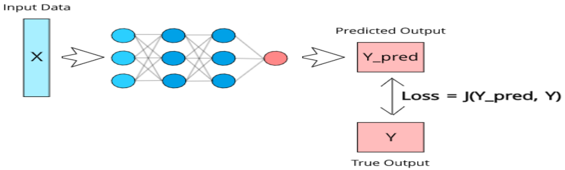
Deconvolution: regularization

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 $\mathbf{w}$

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!

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.

- 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.

Optimization

Basic concepts

October 19, 2023

Skoltech

Skolkovo Institute of Science and Technology

General problem statement

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

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

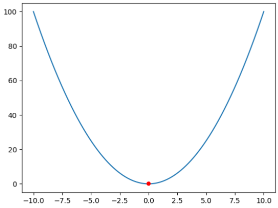
- **Linear Programming:** The function f is linear.
- **Nonlinear Programming:** The function f is non-linear.

Criteria and notations

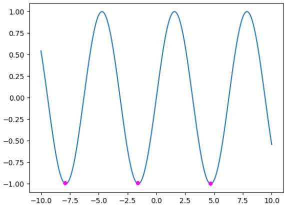
- $f(u) : H \to \mathbb{R}$ - the function to be minimized.
- $u \in U \subseteq H$.
- **Unconstrained programming:** $U = H$.
- **Constrained programming:** $U \subset H$.
- $u^* = \operatorname*{argmin}_{u \in U} f(u)$ - the minimizer.
- $f^* = f(u^*) = \min_{u \in U} f(u)$ - the minimum.
- $U^* \subset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

- 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 $\exists r > 0 : f(u) > f(u^*) \quad \forall u \in U \cap \bar{S}_r(u^*)$

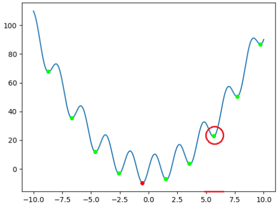
Local and Global minima



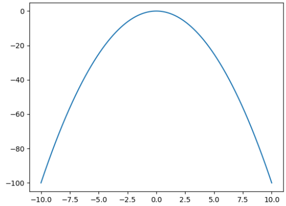
?



?



?



?

Differentiable functionals

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

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

- 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 $f'(u^*) = 0$.

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

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

- 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

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:

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

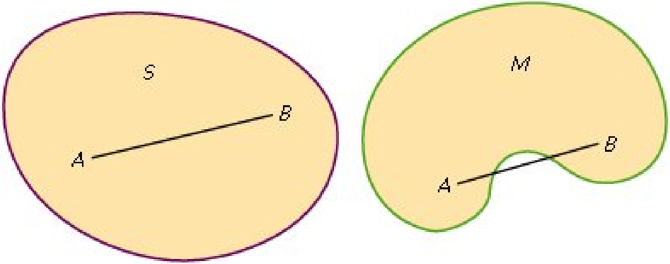
Examples:

- 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 $\bar{S}_r(u_0)$

Convex set



Convex functional

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

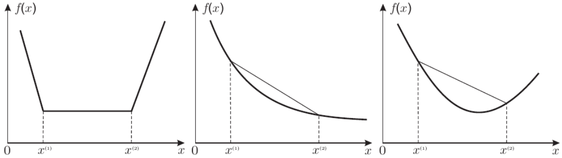
$$f(\lambda u_1 + (1-\lambda)u_2) \leq \lambda f(u_1) + (1-\lambda)f(u_2).$$

- 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

- $f(x) = x^2$ 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.

Optimization

Some aspects of will- and ill-posed problems

October 19, 2023

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:

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

$$\begin{aligned} \Phi'(x) &= 2(A^* Ax - A^* b), \\ \Phi''(x) &= 2A^* A \geq 0. \end{aligned}$$

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:**

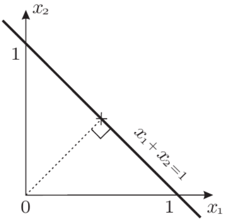
$$x = \min_{x:A^*Ax=A^*b} ||x||$$

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 normal pseudosolution is $x_n = (1/2, 1/2)^T$.



- 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: $x_n = (1/2, 1/2)^T$

Well-posed problems in terms of optimization

- Let X, B be some Hilbert spaces.

$$Ax = b, x \in X, b \in B, A : X \to B, \tag{1}$$

- Consider the residual:

$$M(x) = ||Ax - b||_B^2, \quad x_m = \operatorname{argmin} M(x) \tag{2}$$

- If the operator A is **linear**, then the problem (2) is a **quadratic** optimization problem and has only the unique global minimizer.
- If the operator A is not linear, the programming problem still might be convex.
- If the problem (6) has an exact solution x^*, then the exact minimizer $x_m = x^*$.
- If the operator A has low condition number, then the minimizer x_m is stable.

Ill-posed problems in terms of optimization: pseudosolutions

Let X, B be some Hilbert spaces.

$$Ax = b, x \in X, b \in B, A : X \to B, \tag{3}$$

- If the solution of the above problem does not exist, the residual functional still has a minimum!
- The minimizer is called the **normal pseudosolution** and **always** exists.
- If the operator A is non-linear, there might be some number (up to infinity) of minimizers. The solution with minimal norm is also a **normal pseudosolution**.
- If due to some restrictions the solution exists but is outside of the solution scope, then, regularizing the functional, we can also chose the **normal pseudosolution**

Regularization

For the equation $Ax = b$ we form the residual $\Phi(x) = ||Ax - b||_B^2$. The cost Tiknonov's functional can be formed as follow:

$$M_\alpha(x) = ||Ax - b||_B^2 + \alpha\Omega(x),$$

where the functional Ω, defined on the subset of possible solutions space $X_1 \subset X$ is called regularization functional (or stabilizer), and should satisfy the following conditions:

- $\Omega(x) \geq 0, \quad \forall x \in X_1.$
- The element x^* (i.e., the exact solution) belongs to the support of the functional Ω: $x^* \in X_1$
- $\forall d > 0$, the set $F_{1,d} = \{x \in F_1 : \Omega(x) \leq d\}$ is a compact.

The approximate regularized solution:

$$x_\alpha = \underset{x \in F_1}{\operatorname{argmin}}\, M_\alpha(x)$$

Ill-conditioned problems: the instability

Let X, B be some Hilbert spaces, and A be a linear bounded operator.

$$Ax = b, x \in X, b \in B, A : X \to B, \tag{4}$$

Consider the residual:

$$\Phi(x) = ||Ax - b||_B^2$$

The Frechet derivative and necessary (in some cases - sufficient) condition for the minima at some element x:

$$A^*Ax - A * b = 0, \quad A^*Ax = A^*b.$$

The condition number: $cond(A) = ||A||||A^*||$. Probably, $||A^*||$ is big...
OR EVEN MIGHT NOT EXIST!

Let X, B be some Hilbert spaces, and A be a linear bounded operator.

$$Ax = b, x \in X, b \in B, A : X \to B, \tag{5}$$

Consider the Tikhonov's functional with the simplest stabilizer $\Omega = ||x||^2_{X_1}, X_1 \subset X$. In this case:

$$M(x) = ||Ax - b||^2_B + \alpha ||x||^2_B, \quad x_\alpha = \underset{x \in X}{\operatorname{argmin}} \, M(x)$$

Then

- The solution is a **normal pseudosolution**
- The minimizer exists, and it is unique if A is bounded linear operator
- The minimizer converges to the exact solution if errors $\to 0$ (stability).

Non-unique solutions. Choise principle

Let X, B be some Hilbert spaces, and A be a non-linear operator.

$$Ax = b, x \in X, b \in B, A : X \to B, \quad (6)$$

- Let $Q_\mu : \Phi(x) \equiv ||Ax - b|| \leq \mu$ - set of possible solutions.
- Use physical/chemical/economical/etc laws in order to make a constrains for the solution
- Two ways:
  - Use the constrained optimization of the residual $\Phi(x)$.
  - Construct a stabilizer $\Omega(x)$ such that on 'better' solution it reaches lesser values.
  - Example: Functional space with bounded total variation allows us to reconstruct blurred piecewise-constant functions.

Methods

Zero order methods list

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

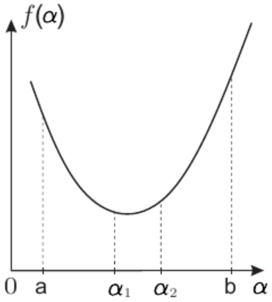
1D optimization: the Golden Section Method

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

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

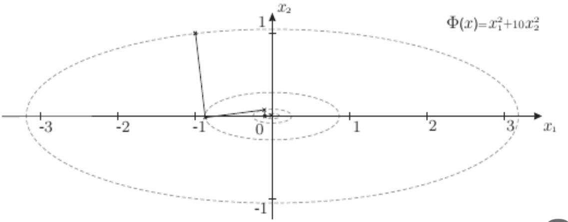
1D optimization: the quadratic approximation

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

- 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.

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

- Let the functional f be differentiable.
- Chose initial approximation x_0 and calculate initial gradient $g_0 = f'(x_0)$.
- $x_1 = x_0 - \alpha_0 g_0$, where α_0 is parameter (step) to chose.
- $g_n = f'(x_n); \alpha_n = \underset{\alpha\geq 0}{\operatorname{argmin}} f(x_n + \alpha g_n); x_{n+1} = x_n - \alpha_n g_n$



Constant descent rate

Let the function $f(x)$ be:

- Limited from below $(f(x) \geq f^* > -\infty)$
- Differentiable: $\exists f'(x) \forall U$
- The gradient is Lipshitz-continuous with constant L: $||f'(x) - f'(y)|| \leq L||x - y||$

Then, if $0 < \alpha < 2/L$, the following is true:

- The gradient tend to zero with growing k:

$$\lim_{k \to \infty} f'(x_k) = 0$$

- The function $f(x_k)$ decays with k:

$$f(x_{k+1}) \leq f(x_k)$$

Steepest descent

The optimization problem:

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

- Assume the function $f(x)$ be differentiable;
- Assume the function $f'(x)$ be continuous.

The **Steepest gradient descent**:

$$x_{n+1} = x_n - \alpha_n g_n$$

$$\alpha_n = \operatorname*{argmin}_{\alpha \geq 0} \varphi_k(\alpha), \quad \varphi_k(\alpha) = f\left(x_n - \alpha f'(x_n)\right)$$

- If $f(x)$ is quadratic function $f(x) = (Ax, x) + (b, x)$, then:

$$\alpha_n = \frac{||f'(x_n)||^2}{(Af'(x_n), f'(x_n))}$$

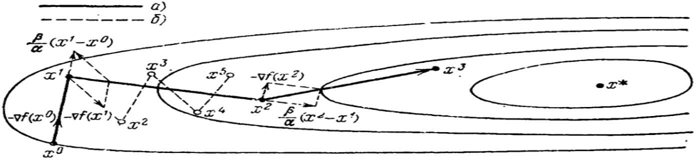
Heavy ball method

- Within the current iteration, the gradient descent does not use the information obtained in earlier iterations.
- **Multistep (s-step) methods**: $x_{n+1} = \varphi_n(x_n, x_{n-s-1})$

The heavy-ball method:

$$x_{n+1} = x_n - \alpha f'(x_n) + \beta(x_n - x_{n+1})$$

- Good for ill-conditioned problems.



Conjugate gradients method

One more two-step method:

$$x_{n+1} = x_n - \alpha_n f'(x_n) + \beta_n(x_n - x_{n-1}),$$

where

$$\{\alpha_n, \beta_n\} = \underset{(\alpha,\beta)}{\operatorname{argmin}}\, f\left(x_n - \alpha f'(x_n) + \beta(x_n - x_{n-1})\right)$$

In case of $f(x) = (Ax, x)/2 - (b, x)$ (quadratic):

$$\alpha_n = \frac{||r_n||^2(Ap_n, p_n) - (r_n, p_n)(Ar_n, p_n)}{(Ar_n, r_n)(Ap_n, p_n) - (Ar_n, p_n)^2}, \quad r_n = f'(x_n) = Ax_n - b,$$

$$\beta_n = \frac{||r_n||^2(Ar_n, p_n) - (r_n, p_n)(Ar_n, r_n)}{(Ar_n, r_n)(Ap_n, p_n) - (Ar_n, p_n)^2}, \quad p_n = x_n - x_{n-1}$$

Conjugate gradients method

- The method uses two previous approximations instead of only one. To initiate the CGM, the point x_1 can be obtained using any x_0 with the steepest descent method.
- The functions $r_n = f'(x_n)$ are pairwise orthogonal: $(r_n, r_k) = 0, \forall n \neq k$.
- The directions p_n are conjugate: $(Ap_n, p_k) = 0, \forall n \neq k$.
- In case of quadratic programming, the CGM allows to reach minimum at n iterations.
- Good for a 'big' quadratic tasks!

Another form of the CGM:

$$x_{n+1} = x_n + \alpha_n p_n, \quad \alpha_n = \operatorname*{argmin}_{\alpha} f(x_n + \alpha p_n),$$

$$p_n = r_n + \beta_n p_{n-1}, \quad \beta_n = \frac{||r_n||^2}{||r_{n-1}||^2},$$

$$r_n = f'(x_n), \beta_0 = 0$$

Optimization

Examples

October 19, 2023

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.

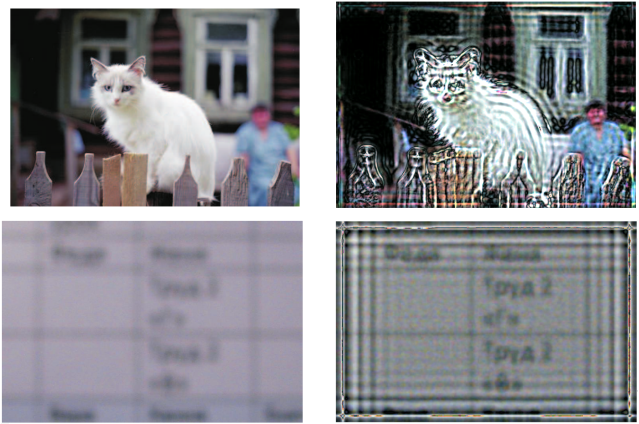
The cost Tikhonov's functional:

$$M_\alpha[z] = ||Az - u||^2 + \alpha\Omega[z],$$

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

1. $\Omega[z] = ||z||^2_{H^1}$. The solution of a quadratic functional can be found analytically.
2. $\Omega[z] = ||z||_{TV}$. In this case, the cost functional is not quadratic and requires iterative minimization. The minimization was provided using the method of Conjugated Gradients Projections (MCGP).

Quadratic optimization



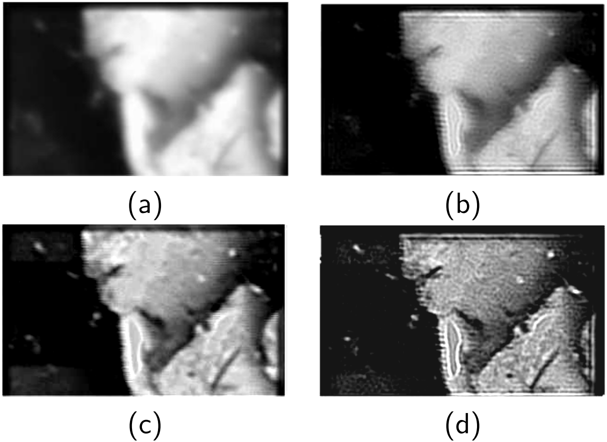
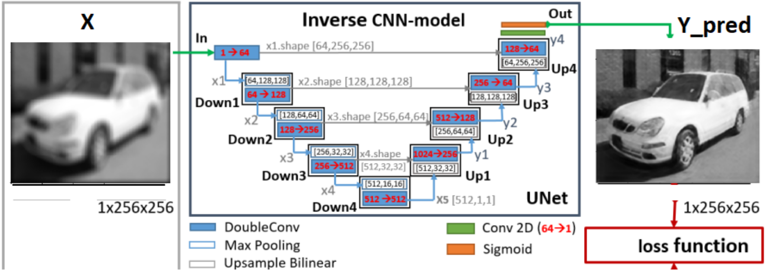


Figure: Results of image reconstruction in SEM BE: (a) - the enter data, (b) - reconstruction on H^1 Sobolev's space, analytical solution, (c) - reconstruction on TV Bounded Total Variations functional space, (MCGP, zero first approximation), (d) - reconstruction on TV with the result on H^1 taken as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

NN: Gradient Descent



L1 loss

BCE loss

MSE loss

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

Molecular dynamics

- Abalone
- AMBER
- Gromacs
- LAMMPS
- NAMD

FEM Meshing

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

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

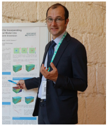
I.Zacharov: Introduction to Linux and Supercomputers

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

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

Thank you for your attention!