Scientific Computing Important information

September 29, 2021

Skoltech

Skolkovo Institute of Science and Technology

The structure of the course

- **Lectures:** Nikolay Koshev
- **Classes & Labs:** Dmitry Yarotsky
- **Homework grading:** Ekaterina Skidchenko
- **Personal research project**
- **Final written exam**

The grading criteria

- **Class Participation:** 5%
- **Homework assignments:** 30%
- **Final written exam:** 35%
- **Personal research project:** 30%

The Final written exam

The exam ticket consists of:

- 5 tasks consisting of 2 theoretical questions each.
- The theoretical questions can be found at the end of each lecture.
- 7 problems.

The Personal research project

The topics can be found in Canvas, or you can provide your own topic having nonempty intersection with the course topics. The presentation of the project should contain:

1. Informal description of the goal of the research and motivation.
2. The clear mathematical statement of the problem.
3. The State-Of-The-Art techniques on the problem.
4. Methods being used.
5. Results.
6. Discussion: conclusions, findings, comparison of used method with other methods.

The structure of the lectures

- Lec. 1. Introduction to Scientific Computing. A bit of history, a bit of preliminary information. Introduction to HPC: basic terminology, principles, programming strategies. Brief classification of the problems arising in Scientific Computing: Forward and Inverse, Well- and ill-posed, static and dynamic, etc.
- Lec. 2. Numerical approaches: reduction of the problem to linear system, meshing, approximation.
- Lec. 3. Basics of differential equations: problems classification and numerical solution strategies.
- Lec. 4. Integral transforms and integral equations.
- Lec. 5. Classification of second order partial differential equations, features of them, strategies of solution.
- Lec. 6. Basics of optimization/mathematical programming: concept, problem classification, some of numerical approaches.
- Lec. 7. Stochastic problems and some approaches to solution of them.

Introduction to Scientific Computing

Nikolay Koshev, Dmitry Yarotsky, Maxim Fedorov

September 29, 2021

Skoltech
Skolkovo Institute of Science and Technology

Tunnel Vision by Experts

- "I think there is a world market for maybe five computers." ***Thomas Watson, chairman of IBM, 1943.***.
- "There is no reason for any individual to have a computer in their home". ***Ken Olson, president and founder of Digital Equipment Corporation, 1977***.
- "640K [of memory] ought to be enough for anybody." ***Bill Gates, chairman of Microsoft,1981.***.
- "On several recent occasions, I have been asked whether parallel computing will soon be relegated to the trash heap reserved for promising technologies that never quite make it." ***Ken Kennedy, 1994***.

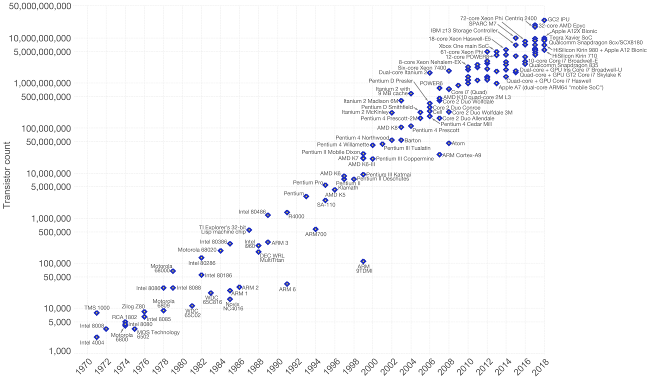
Slide source: Warfield et al.

2019: 50+ years of an exponential growth

Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

Our World in Data

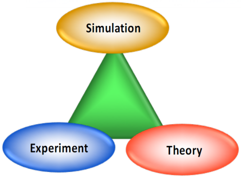
Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.



Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

Skoltech
Skolkovo Institute of Science and Technology

Simulation: The Third Pillar of Science

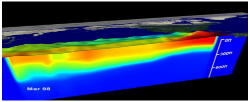


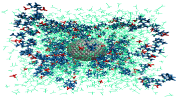
- **Traditional scientific and engineering method:**
  - Do theory of paper design;
  - Perform experiments or build system
- **Limitations**
  - Too difficult - build large wind tunnels;
  - Too expensive - build a throw-away passanger jet;
  - Too slow - wait for climate of galactic evolution;
  - Too dangerous - weapons, drug design, climate experimentation.
- **Computational science and engineering paradigm**:
  Use computers to **simulate and analyze** the phenomenon:
  - Based on known physical laws and efficient numerical methods;
  - Analyze simulation results with computational tools and methods beyond what is possible manually.

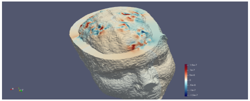
CompSci & Eng: Application areas

Application areas that benefit from usage of Computational Science and Engineering

- Computational Physics,
- Computational Fluid Dynamics,
- Bioinformatics,
- Modelling of Oil & Gas reservoirs,
- Weather forecasts,
- Visualisation,
- Large Data analysis,
- Renewable Energy,
- Large Data and Smart Cities applications
- and many others.



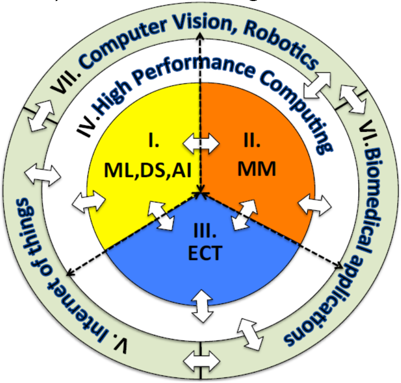




Skoltech
Skolkovo Institute of Science and Technology

Interconnections between research directions of the CDISE

HPC facilities: 5 installations (each focused on specific applications); up to 2.5 Pflops power (by the end of 2018); >2 Pb of storage.



Data Science (DS), Machine & Deep Learning (ML & DL) and Artificial Intelligence (AI).

Mathematical Models (MM) underpin future and emerging technologies related to Digital Agronomics & Pharmaceutical Design, Quantum Enhanced Simulation, Soft Matter & Complex Systems, Distributed Intelligent Systems and Biological & Medical Applications

Emerging Computing Technologies (ECT), such as Quantum Enhanced Computation & Simulations (QECS) focuses on the contemporary emerging and knowledge-generating computational technology

Scientific Computing
Introduction to HPC

NB: the slides are based on introductory courses to HPC & Parallel Computing from Berkley ParaLab, University of Reims, Boston University (Doug Sondak), and Lawrence Livermore National Laboratory

September 29, 2021

Skoltech

Skolkovo Institute of Science and Technology

Units of Measure

- High Performance Computing (HPC) units are:
  - Flop: floating point operation, usually double precision unless noted
  - Flop/s: floating point operations per second
  - Bytes: size of data (a double precision floating point number is 8)
- Typical sizes are millions, billions, trillions...
  Mega: Mflop/s = 10^6 flop/sec; Mbyte = 10^6 bytes (Mebibyte = 2^{20} bytes, MiB)
  Giga: Gflop/s = 10^9 flop/sec; Gbyte = 10^9 bytes
  Tera: Tflop/s = 10^{12} flop/sec; Tbyte = 10^{12} bytes
  Peta: Pflop/s = 10^{15} flop/sec; Pbyte = 10^{15} bytes (Pebibyte = 2^{50} bytes, PiB)
  Exa: Eflop/s = 10^{18} flop/sec; Ebyte = 10^{18} bytes
  Zetta: Zflop/s = 10^{21} flop/sec; Zbyte = 10^{21} bytes
  Yotta: Yflop/s = 10^{24} flop/sec; Ybyte = 10^{24} bytes
- Current fastest 514 Pflop/s (Supercomputer Fugaku, Fujitsu RIKEN Center for Computational Science, Japan.). Up-to-date list at www.top500.org;
- Current fastest machine in Russia is installed in Sberbank (SberCloud), app. 9 Pflop/s (see http://top50.supercomputers.ru/?page=rating)

The TOP500 Project

- Listing the 500 most powerful computers in the world
- Yardstick: Rmax of Linpack
  - Solve Ax=b, dense problem, matrix is random
- Update twice a year:
  - ISC in June in Germany
  - SC in November in the U.S.
- All information available from the TOP500 web site at: www.top500.org

Impact of Device Shrinkage

What happens when the feature size (transistor size) shrinks by a factor of x ?

- Clock rate goes up by x because wires are shorter - actually less than x, because of power consumption
- Transistors per unit area goes up by x^2
- Die size also tends to decrease - typically another factor of x
- Raw computing power of the chip goes up by x^4 ! - typically x^3 is devoted to either on-chip
- parallelism: hidden parallelism such as ILP
- So most programs x^3 times faster, without changing them

Parallelism nowadays

- These arguments are no longer theoretical
- All major processor vendors are producing multicore chips
  - Almost every machine is in fact is a parallel machine
  - To keep doubling performance, parallelism must double
- Which (commercial) applications can use this parallelism?
  - Do they have to be rewritten from scratch?
- Will all programmers have to be parallel programmers?
  - New software model needed
  - New compilers would hide complexity from most programmers - eventually
  - In the meantime, they would need to understand it
- Computer industry betting on this big change, but does not have all the answers

Memory is Not Keeping Pace

Technology trends against a constant or increasing memory per core

- Memory density is doubling every three years; processor logic is every two
- Storage costs (dollars/Mbyte) are dropping gradually compared to logic costs

- Question: Can you double concurrency without doubling memory?
- Strong scaling: fixed problem size, increase number of processors
- Weak scaling: grow problem size proportionally to number of processors

Parallel Speedup

Consider a system containing P processors.

- Parallel speedup: how much faster does my code run in parallel compared to serial?
- Max value is P, the number of processors

$$\text{Parallel Speedup} = \frac{\text{Serial execution time}}{\text{parallel execution time}}$$

Parallel Efficiency

Consider a system containing P processors.

- Parallel efficiency: how much faster does my code run in parallel compared to linear speedup from serial?
- Max value is 1

$$\text{Parallel efficiency} = \frac{\text{Parallel speedup}}{P}$$

Principles of Parallel Computing

- Finding enough parallelism (Amdahl's Law, Gustafson's Law)
- Granularity - how big should each parallel task be
- Locality - moving data costs more than arithmetic
- Load balance - don't want 1000 processors to wait for one slow one
- Coordination and synchronization - sharing data safely
- Performance modelling/debugging/tuning

All of these things make parallel programming more challenging than sequential programming

"Automatic" Parallelism in Modern Machines

- Bit level parallelism - within floating point operations, etc.
- Instruction level parallelism (ILP) - multiple instructions execute per clock cycle
- Memory system parallelism - overlap of memory operations with computation
- OS parallelism - multiple jobs run in parallel on commodity SMPs

Limits to all of these - for very high performance, need user to identify, schedule and coordinate parallel tasks

Finding Enough Parallelism

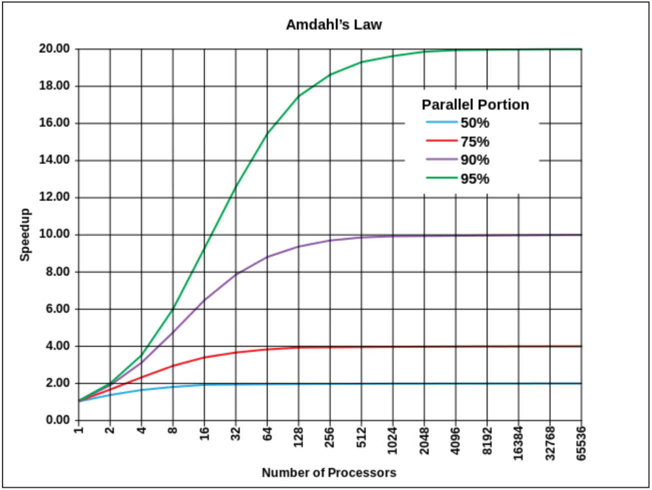
- Suppose only part of an application seems parallel
- Amdahl's law - let s be the fraction of work done sequentially, so (1-s) is fraction parallelizable
  - P = number of processors

$$\begin{aligned} \text{Speedup}(P) &= \text{Time}(1)/\text{Time(P)} \\ &<= 1/(s + (1-s)/P) \\ &<= 1/s \end{aligned}$$

- Even if the parallel part speeds up perfectly **performance is limited by the sequential part**
- Top500 list: currently fastest machine has P $10.7M$; 2^{nd} fastest has 3.1 M

Skoltech
Skolkovo Institute of Science and Technology

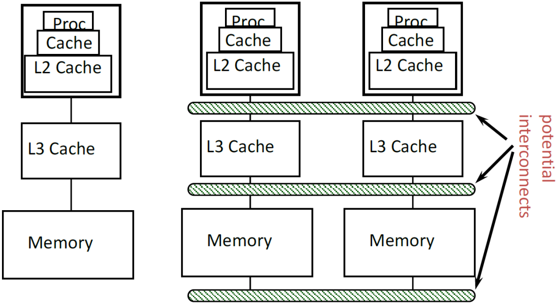
Amdahl's law illustration



Overhead of Parallelism

- Given enough parallel work, this is the biggest barrier to getting desired speedup
- Parallelism overheads include: - cost of starting a thread or process - cost of communicating shared data - cost of synchronizing - extra (redundant) computation
- Each of these can be in the range of milliseconds (=millions of flops) on some systems
- Tradeoff: Algorithm needs sufficiently large units of work to run fast in parallel (i.e. large granularity), but not so large that there is not enough parallel work

Conventional Storage Hierarchy



- Large memories are slow, fast memories are small
- Storage hierarchies are large and fast on average
- Parallel processors, collectively, have large, fast cache
  - the slow accesses to "remote" data we call "communication"
- Algorithm should do most work on local data

Load Imbalance

- Load imbalance is the time that some processors in the system are idle due to
  - insufficient parallelism (during that phase)
  - unequal size tasks
- Examples of the latter
  - adapting to "interesting parts of a domain"
  - tree-structured computations
  - fundamentally unstructured problems
- Algorithm needs to balance load
  - Sometimes can determine work load, divide up evenly, before starting: **"Static Load Balancing"**
  - Sometimes work load changes dynamically, need to rebalance dynamically: **"Dynamic Load Balancing"**

Parallel Software Eventually - a view from Berkley (ParLab)

- 2 types of programmers and 2 layers of software
- Efficiency Layer (10% of programmers)
  - Expert programmers build Libraries implementing kernels, "Frameworks", OS,....
  - Highest fraction of peak performance possible
- Productivity Layer (90% of programmers)
  - Domain experts / Non-expert programmers productively build parallel applications by composing frameworks & libraries
  - Hide as many details of machine, parallelism as possible
  - Willing to sacrifice some performance for productive programming
- Expect students may want to work at either level
  - In the meantime, we all need to understand enough of the efficiency layer to use parallelism effectively

Scientific Computing
Lecture 1

Problems Classification

Nikolay Koshev & Maxim Fedorov

September 29, 2021

Skolkovo Institute of Science and Technology

Or use a microscope to hammer in nails.

Classification allows:

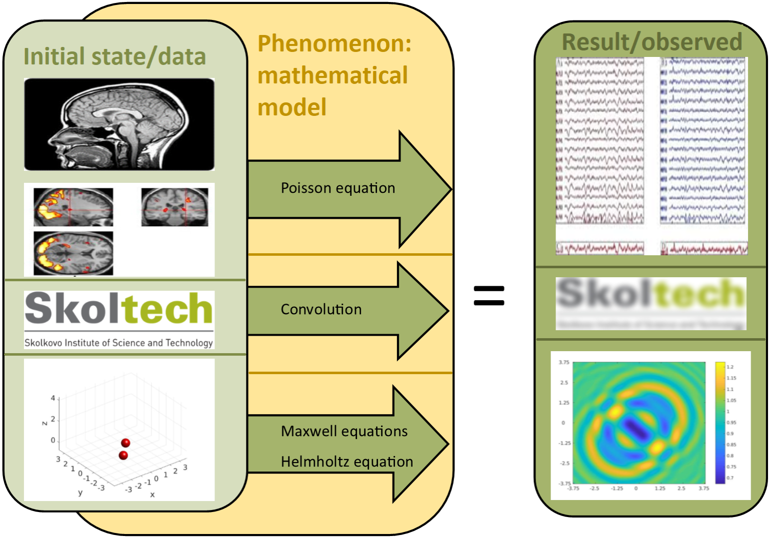
- Not to reinvent the wheel
- Use proper methodology
- Understand the effective way to solve
- Understand the problem itself

Skoltech
Skolkovo Institute of Science and Technology

General classification of Scientific Computing problems

- Forward and Inverse problems
- Well-posed and Ill-posed problems
- Static and Dynamic problems
- Deterministic and Stochastic problems
- Continuous and Discrete problems

Phenomena and data



Skoltech
Skolkovo Institute of Science and Technology

Phenomena and data

Phenomena and data

- Or the operator equation: $A\mathbf{f} = \mathbf{b}$.
- The **Residual**: $L(\mathbf{f}, \mathbf{b}) \equiv A\mathbf{f} - \mathbf{b}$.
- $L(\mathbf{f}, \mathbf{b}) = 0$ - another form of equation.

- $\mathbf{f}$ represents some **input** wrt. phenomenon, say, some initial state of the system, which is being affected during the process.
- A represents the **mathematical model** of the phenomenon under consideration. A affects the initial state $\mathbf{f}$ resulting at $\mathbf{b}$. May be known, unknown or partially known. May depend on the subject $\mathbf{f}$, or be independent.
- $\mathbf{b}$ represents the **output** wrt. phenomenon or the data observed after phenomenon occurred. Always depends on both **input data** $\mathbf{f}$ and **model** A or L.

The input data f could be:

- A vector or a matrix: **f** represents some distribution which defines some state of the phenomenon. Examples:
  - Spatial distribution of properties of the object of research: Electric activity (current density), intensity distribution, color distribution.
  - Spatial distribution of some parameters affecting the phenomenon output (density, solidity, conductivity etc).
  - Temporal distribution of boundary values for heat-transfer problem.
- A set of parameters of different nature, which are affecting on the model behavior. Examples:
  - Scanning electron microscopy: distribution of the electron probe density (function or matrix) comes together with its initial energy (value) and distribution of the density of the object under investigation.
  - Voltammetry: temporal dependence of the cathode current (a vector) comes together with parameters of the electrolyte.
- The input data/state may be changed by phenomenon, or become the same.

The Phenomenon

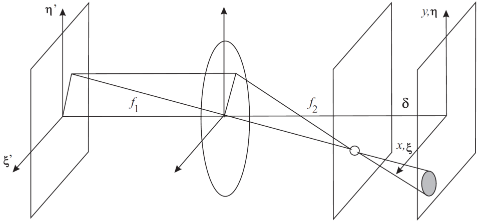
The mathematical description of the phenomenon is called the **Mathematical model**, or just **model**.

- The **model** represents the mathematical connection between the **input** and the **output**.
- The **model** may be both depend or do not depend on $\mathbf{x}$
  - *Example*: Heating of object, heat conductivity of which depends on the temperature. Input is initial heat distribution over the volume under study, model is heat equation, which depends on the initial heat distribution.
  - *Example*: Phenomenon of defocusing of flat screen obviously does not depend on the object it is affecting.
- The **model** may be an object of study; or both **model** and **input data** may be objects of study.
- The **model** may be represented as a differential, integral or matrix/tensor operator.

The output data

- The **output** is a result of application of a **model** on an **input data** .
- In simulation/modelling problems the **output** is the object of study.
- The **output** may be presented with vector of some properties.
- The **output** may be presented with some distribution.
- The **output** may be presented even with a number.
- The **output** may contain all of above.

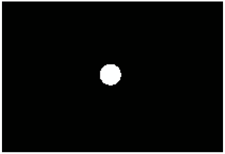
Example: defocused photographs



$$L(f, \mathbf{b}) \equiv K * f - \mathbf{b} = 0, \quad Af \equiv K * f.$$

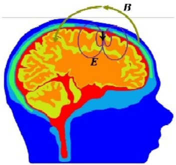
Input (enter) $f(\mathbf{x})$

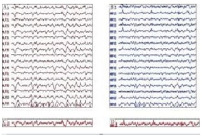
Output $b(\mathbf{x})$



The law (kernel) $K(\mathbf{x}, \xi)$

Source: Yagola A.G., Koshev N.A. Restoration of smeared and defocused color images, Numerical Methods and Programming, V.9, 207-212, 2008 (in Russian)





- **The input data:** Intracranial neuronal current distribution $\mathbf{J}(\mathbf{x})$, conductivity distribution $\sigma(\mathbf{x})$.
- **The Model:** MEG: Poisson equation:

$$L(\mathbf{J},\mathbf{B}) \equiv \nabla \times \mathbf{B} - \mu_0(\mathbf{J} - \sigma \nabla U) = 0,$$

$$\text{or} \quad \mathbf{B}(\mathbf{x}) = A\mathbf{J}(\mathbf{x}) \equiv \frac{\mu_0}{4\pi} \int_{Head} \frac{\nabla \times (\mathbf{J}(\xi) - \sigma \nabla U(\xi))}{|\mathbf{x} - \xi|} d\xi$$

- **The Observed Data:** magnetic induction $\mathbf{B}(\mathbf{x}_k)$, where x_k is a location of $k - th$ sensor

Forward problems, Inverse problems

Nikolay Koshev

September 29, 2021

Skoltech

Skolkovo Institute of Science and Technology

General classification of Scientific Computing problems

- Forward and Inverse problems
- Well-posed and Ill-posed problems
- Static and Dynamic problems
- Deterministic and Stochastic problems
- Continuous and Discrete problems

Forward and Inverse problems

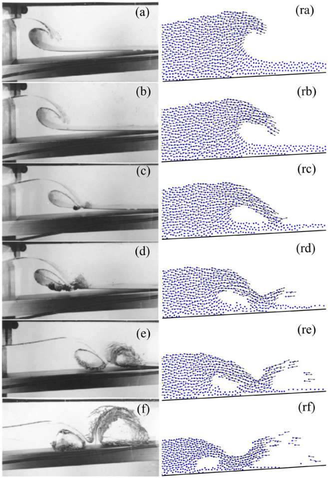
The main equation: $A\mathbf{f} = \mathbf{b}$

- Forward problem (simulation): find the vector $\mathbf{b}$, if the input data $\mathbf{f}$ and the model (operator) A are known.
- Inverse problem:
  - Assuming the observed data $\mathbf{b}$ and the model A to be known, find the **input data f**;
  - Assuming the observed data $\mathbf{b}$ and the **input data f** are known, define the model A;
  - Assuming the observed data $\mathbf{b}$ to be known, and the model A to be partially known, define $\mathbf{f}$ and complement the definition of the model A.

Forward problem: Applications

- **Studying the processes** Knowing basic principles and laws, it is possible to study more complicated processes, hard to understand from the analytical point of view.
- **Predictions** Knowing the basic rules and principles, it is possible to predict the further behaviour of the object of investigation.
- **Designing the processes:**
  - **Top-down approach (TD).** Knowing the objective process in general, it is possible simulate parts of it in order to optimize the process.
  - **Bottom-up approach (BU).** Knowing the simple interactions between the parts of a system, simulate the overall process.
- **Industrial design of objects: both top-down and bottom-up approaches.**

Forward problem: bottom-up approach



- **input data:** initial state of the system: $\mathbf{u}(t_0), \rho(t_0)$, fluid macroscopic parameters: ν.
- **The model:** Navier-Stokes (moving particles semi-implicit method, MPS):

$$\frac{1}{\rho}\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{u} = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{g}$$

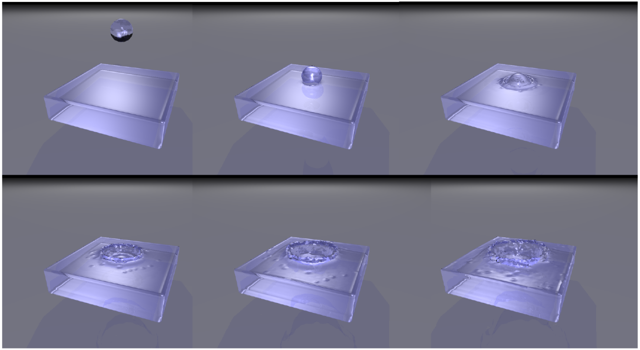
- **The output:** The state of the system at some moment T.
- Take a look at deep learning...

Source: Lizhu Wang et al., Improvement of moving particle semi-implicit method for simulation of progressive water waves. International Journal for Numerical Methods in Fluids, 2017.
Source: Jinning Li, Lianmin Zheng. DEEPWAVE: Deep Learning based Real-time Water Simulation, 2018

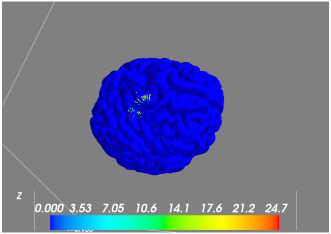
Skoltech
Skolkovo Institute of Science and Technology

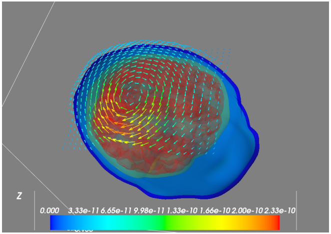
Forward problem: bottom-up approach

Falling drop example: 200000 particles, 10000 time steps



Forward problem: Top-Down approach.





- **input data:** Current density $\mathbf{J}(\mathbf{x}), \mathbf{x} \in \text{Cortex}$; Conductivity $\sigma(\mathbf{x}), \mathbf{x} \in \text{Head}$.
- **The model:** Poisson equations

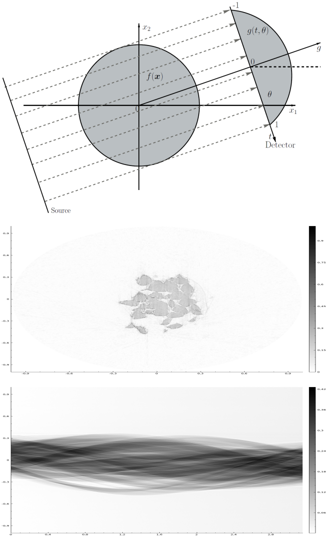
$$\Delta \mathbf{B} = \nabla \times (\mathbf{J} - \sigma \nabla U), \mathbf{x} \in \text{Head},$$
$$\nabla \cdot (\sigma \nabla U) = \nabla \cdot \mathbf{J}, \mathbf{x} \in \text{Head}.$$

- **The output:** Magnetic induction $\mathbf{B}(\mathbf{x})$ at positions of N_s sensors $\tilde{\mathbf{x}} = \{\mathbf{x}_k, k = 1, ..., N_s\}$.

Skoltech
Skolkovo Institute of Science and Technology

- **Studying the object or its properties.** On the base of observed data and known connection between the data and observed information, it is possible to restore the properties (input data) of some object.
- **Studying the model.** Knowing the object properties, and observing some data, we can study the connection between the object and observed data, which we call 'model'.
- **Studying both object and the model.** On the base of observed data and **incomplete** knowledge of 'model', it is sometimes possible to restore both 'model' and properties of the object under investigation.

Studying the object: tomography



- **input data:** Feature distribution: $f(\mathbf{x})$.
- **The model:** The Radon transform

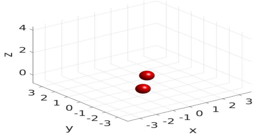
$$(Rf)(\theta, t) = \int_{\mathbb{R}^2} f(\mathbf{x})\delta(\mathbf{x}\cdot\xi - t)d\mathbf{x}$$

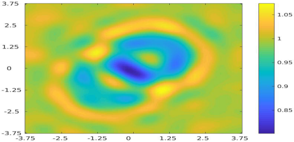
- **The output:** The sinogram

$$g(\theta, t) = Rf(\theta, t)$$

source: Eduardo Miqueles, Nikolay Koshev and Elias S. Helou. A Backprojection Slice Theorem for Tomographic Reconstruction. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 201

Studying both object and model: Coefficient inverse problems





- **input data:** Feature distribution (refractive index): $n(\mathbf{x})$
- **The model:** The Helmholtz

$$\Delta u(\mathbf{x}, k) + k^2 n^2(\mathbf{x}) u(\mathbf{x}, k) = 0.$$

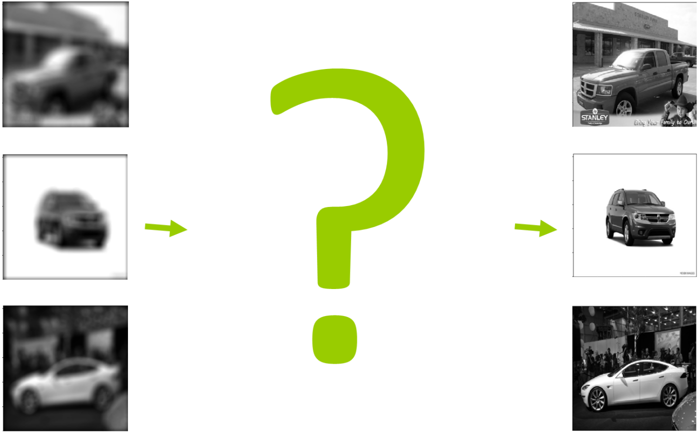
- **The output:** The cell-phone camera photo as a boundary condition in the frame P:

$$u(\mathbf{x}, k) = f(\mathbf{x}, k), \quad \mathbf{x} \in P$$

Source: M.Klibanov, N.Koshev et al., A Numerical Method to Solve a Phaseless Coefficient Inverse Problem from a Single Measurement of Experimental Data. *SIAM J. IMAGING SCIENCES*, 2018.

Example: defocused photographs & NN

Example: defocused photographs & NN

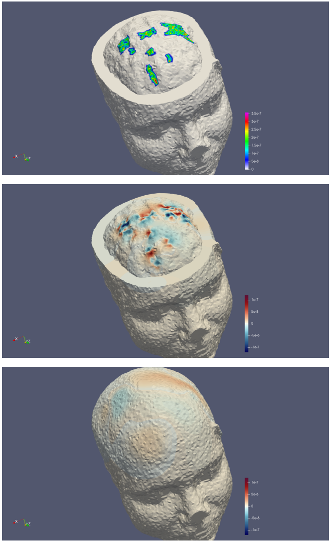


b(x) f(x)

The connection between forward and inverse problems

- The relativity of classification.
- Some inverse problems may be classified as forward (simulation) problems. Example: the Cauchy problem in EEG/MEG.
- The inverse problems are often being solved using the simulation. Iterative approaches are often based on comparison the simulation results on the base of the assumed object or partially known 'model'.

Forward or inverse?



- Let $u(\mathbf{x})$ is a potential, and $\mathbf{J}(\mathbf{x})$ is the current density.
- **Forward Problem:** $\mathbf{J}(\mathbf{x}) \to u(\mathbf{x})$
- **Inverse Problem:** $u(\mathbf{x}) \to \mathbf{J}(\mathbf{x})$
- **And what about** $u(\mathbf{x}), x \in Scalp$ on scalp to $\to u(\mathbf{x}), x \in Cortex$ on the cortex (brain surface)? :)
- **Finally,** $5x = 10$ or $\frac{10}{5} = x$?

Source: N.Koshev et al., FEM-based Scalp-to-Cortex data mapping via the solution of the Cauchy problem. *arXiv preprint arXiv:1907.01504*, 2019

Two approaches

Solving **ANY** problem, we can **try** to find:

- **Explicit solution.** $A\mathbf{f} = \mathbf{b} \implies \mathbf{f} = A^{-1}\mathbf{b}$ (of course, if the operator A^{-1} exists).

  PROS: This scheme is very fast and, in case of well-posed problems, most accurate.

  CONS: This scheme is capable for cases when the inverse operator exists, stable and easy to find.

  Mostly it is the case of SOME **well-posed** problems.

- **Iterative approach.**
  - Chose the first approximation $\mathbf{f}_0$;
  - Construct an appropriate functional $M(\mathbf{f}_0) = F(L(\mathbf{f}_0, \mathbf{b}))$;
  - Change $\mathbf{f}_0$ to obtain $\mathbf{f}_1$ such that $M(\mathbf{f}_1) < M(\mathbf{f}_0)$;
  - Iterate.

  PROS: With the right construction of the functional $M(\mathbf{f})$, the iterative is capable to solve most of problems.
  CONS: The scheme demands to solve the FORWARD problem many times in order to calculate the values $M(\mathbf{f}_k)$, which makes it much slower. The accuracy for some kinds of problems may be lesser.
  This approach is being applied to most of **ill-posed** problems.

Selection of an approach

- X-ray tomography: both approaches are being used.
- Blurred images reconstruction: both approaches can be used.
- Blurred images (AI): iterative approach only.
- EEG/MEG forward problem: both approaches on dependence on approximation.
- Coefficient Inverse Problems: iterative approach only.

Well- & Ill-posed problems

Nikolay Koshev

September 29, 2021

Skoltech
Skolkovo Institute of Science and Technology

Classification of Scientific Computing problems

- Forward and Inverse problems
- Well-posed and Ill-posed problems
- Static and Dynamic problems
- Deterministic and Stochastic problems
- Continuous and Discrete problems

The object and data are connected with a law

The operator equation: $A\mathbf{f} = b$.

- $\mathbf{f}$ represents input (wrt. phenomenon) data. Can be initial state of the system or can define its properties.
- $\mathbf{b}$ represents result (wrt. phenomenon) data. Can be e.g. observed data.
- A represents the model, i.e. the mathematical description of the phenomenon affecting the input to obtain the result. May be known, unknown or partially known.

Well-posed problems
(as defined by Hadamard 1902)
A problem is **well-posed** if:

- A solution exists.
- The solution is unique.
- The solution depends continuously on the **input data**.

- Problems which do not fulfill these criteria are **ill-posed**.
- Well-posed problems have a good chance to be solved numerically with a stable algorithm.

Ill-posed problems

- Ill-posed problems play an important role in some areas. Most of **inverse problems** are ill-posed. But not all of them.
- Problem needs to be reformulated for numerical treatment.
- Add additional constraints to chose the right solution from set of possible solutions. For example, smoothness or sharpness of the solution.
- **Input data** need to be regularized / preprocessed.
- The **model** should be researched and complemented. Sometimes, it should also be regularized.
- Ill-posed problems demand a development of **stable** algorithm.

Ill-posed inverse problems

$$A\mathbf{f} = b, \quad \mathbf{f} \in F, \mathbf{b} \in B.$$

Thus, theoretically:

$$\mathbf{f} = A^{-1}\mathbf{b}.$$

- The inverse operator A^{-1} does not exist.
- The inverse operator A^{-1} is not defined on the whole set $B : AF \neq B$.
- The inverse operator A^{-1} is not continuous.
- The inverse operator A^{-1} is not defined uniquely.

Ill-conditioned problems

- Ill-conditioned problems are the simplest example of ill-posed problems. Forward problems are sometimes also ill-conditioned and, therefore, ill-posed.
- $\Longrightarrow$ Small changes (errors, noise) in data lead to large errors in the solution.
- Can occur if continuous problems are solved approximately on a numerical grid.
  PDE $\Longrightarrow$ algebraic equation in a form $A\mathbf{f} = \mathbf{b}$
- Condition number of matrix A:

$$\kappa(A) = \left|\frac{\lambda_{max}(A)}{\lambda_{min}(A)}\right|,$$

  where $\lambda_{max}(A)$ and $\lambda_{min}(A)$ are maximal and minimal eigenvalues of A.
- Well conditioned problems have a low condition number.

Ill-conditioned problems

- Consider the system for a pair $\mathbf{f} = (x_1, x_2)^T$:

$$A\mathbf{f} = \mathbf{b}, \quad A = \begin{pmatrix} 1 & 1 \\ 1 & 1.001 \end{pmatrix}, \quad \mathbf{b} = (2,2)^T \Leftrightarrow \begin{cases} x_1 + x_2 = 2 \\ x_1 + 1.001 x_2 = 2. \end{cases}$$

- The system has a solution $(2, 0)^T$.
- Now let the $\mathbf{b} = (2, 2.001)^T$ instead of $\mathbf{b} = (2,2)^T$. The solution is $(1,1)^T \ldots$
- The condition number is $\kappa(A) = 4004.0010$.

Differentiation is ill-posed too

Let $f(x)$ be continuously differentiable, $x \in [0, 1]$. Consider $f_\delta(x) = f(x) + n_\delta(x)$, where $n_\delta(x) = \sqrt{2}\delta sin(2\pi kx)$ - a high frequency noise. It is easy to show that:

$$||f(x) - f_\delta(x)||^2_{L^2} = \delta^2.$$

On the other hand

$$\partial_x f_\delta(x) = \partial_x f(x) + 2\sqrt{2}\pi k\delta cos(2\pi kx),$$

and thus, $||\partial_x f(x) - \partial_x f_\delta(x)||^2_{L^2} = 4\pi\delta^2 k^2$. Assume the error $\delta = 0.01$ (1% error), and the frequency $k = 1000$. Even for high intensity signal ($||f||_{L^2} = 100$), we obtain the respective error

$$\frac{||\partial_x f_\delta - \partial_x f||}{||\partial_x f||} \approx 112\%...$$

Static and Dynamic problems

Nikolay Koshev

September 29, 2021

Skolkovo Institute of Science and Technology

Classification of Scientific Computing problems

- Forward and Inverse problems
- Well-posed and Ill-posed problems
- Static and Dynamic problems
- Deterministic and Stochastic problems
- Continuous and Discrete problems

- **Static problem** is time-independent. In static problem, the observed data $\mathbf{b}$, the model A, and the input $\mathbf{f}$ do not depend on time.
- **Dynamic problem** is the problem, evolving in time. It means at least the observed data $\mathbf{b} \equiv \mathbf{b}(..., t)$. The model A and input $\mathbf{f}$ also can be time-dependent.
- **Quasi-static problem** is a dynamic problem which can be considered static at every moment of time (changes slowly or in a way such that previous's state *behavior* do not affect current step). Mathematically, it often means the derivative of the function of interest on time is small.
- Some dynamic problems can be considered as static problem.
- Some static problems may be also considered as dynamic.

Dynamic problem: heat transfer

$$\begin{aligned} u_t(\mathbf{x},t) - a^2\Delta u(\mathbf{x},t) - f(\mathbf{x},t) = 0, \\ \mathbf{x} \in \Omega \subset \mathbb{R}^n, t > t_0 \geq 0, \\ u(\mathbf{x},t_0) = \varphi(\mathbf{x}). \end{aligned}$$

- u - the temperature.
- a^2 - kinematic heat conductivity.
- $f(\mathbf{x},t)$ - the heat source density.
- $\varphi(\mathbf{x})$ - initial state (temperature at the moment t_0.
- The heat transfer is definitely a dynamic problem.
- The initial (input) data $\varphi(x)$ does not depend on t;
- The model is time-dependent ($f(\mathbf{x},t)$ depends on t).

Static problem for evolving system

- Consider the equilibrium problem for the heat equation:

$$\begin{aligned} u_t(\mathbf{x}, t) - a^2 \Delta u(\mathbf{x}, t) - f(\mathbf{x}) = 0, \\ \mathbf{x} \in \Omega \subset \mathbb{R}^n, t > t_0 \geq 0, \\ u(\mathbf{x}, t_0) = \varphi(\mathbf{x}). \\ u_\Gamma = \mu(\mathbf{x}), \quad \Gamma = \partial\Omega. \end{aligned}$$

  Find $u(\mathbf{x}, \infty)$.
- It is proven that: $u(\mathbf{x}, t) \to w(\mathbf{x}), \quad t \to \infty, \quad \forall\varphi(\mathbf{x})$, where $w(\mathbf{x})$ is the solution of the following stationary problem:

$$\Delta w(\mathbf{x}) = -f(\mathbf{x}), \quad , w(\mathbf{x})|_\Gamma = \mu(\mathbf{x}).$$

- Thus, equilibrium problems could be considered both as static and dynamic problems.

- Consider Maxwell equations:

$$\nabla \cdot \mathbf{E} = \rho/\epsilon_0, \quad \nabla \cdot \mathbf{B} = 0 \tag{1}$$

$$\nabla \times \mathbf{B} = \mu_0(\mathbf{J} + \epsilon_0 \frac{\partial \mathbf{E}}{\partial t}), \quad \mathbf{J} = \sigma \mathbf{E} + (\epsilon - \epsilon_0)\frac{\partial \mathbf{E}}{\partial t}, \tag{2}$$

- Let $E = E_0(\mathbf{x})exp(i2\pi ft)$, where f is frequency.
- Then:
  $\nabla \times \mathbf{B} = \mu_0(\sigma \mathbf{E} + (\epsilon - \epsilon_0)\frac{\partial \mathbf{E}}{\partial t} + \epsilon_0 \frac{\partial \mathbf{E}}{\partial t}) = \mu_0(\sigma \mathbf{E} + i2\pi ft\epsilon \mathbf{E})$.
- Real values for the brain: $\sigma \approx 0.3, \epsilon = 10^5\epsilon_0, f \approx 100$. Then $2\pi f\epsilon/\sigma \approx 10^{-3} << 1$.
- Since the terms $\frac{\partial \mathbf{E}}{\partial t}$ are much lesser than ohmic current $\sigma \mathbf{E}$, we may ignore them, considering the problem as quasistationary.

Source: Hämäläinen M, Hari R, Ilmoniemi RJ, Knuutila J, Lounasmaa OV. Magnetoencephalography: theory, instrumentation, and applications to noninvasive studies of the working human brain. Rev Modern Phys. 1993

Static or dynamic? Oscillations of the string

The wave (D'Alembert) equation:

$$\left(\Delta - \frac{1}{v^2}\frac{\partial^2}{\partial t^2}\right)u = 0, \quad a \leq x \leq b, u(a) = 0; \quad u(b) = 0.$$

Assume the solution can be presented in a form $u(x,t) = y(x) \cdot exp(i\omega t)$. After substitution to the equation above, we obtain the following ODE:

$$\frac{d}{dx}\left(p(x)\frac{dy(x)}{dx}\right) = -k^2 q(x) y(x), \quad y(a) = 0, y(b) = 0.$$

where $k = \omega / v$ is a wavenumber, ω - its frequency, and v is the speed.

Just one more axis!

Determenistic & Stochastic

Nikolay Koshev

September 29, 2021

Skoltech

Skolkovo Institute of Science and Technology

Classification of Scientific Computing problems

- Forward and Inverse problems
- Well-posed and Ill-posed problems
- Static and Dynamic problems
- Deterministic and Stochastic problems
- Continuous and Discrete problems

Deterministic and Stochastic problems

- **Deterministic problems:** Model A, object $\mathbf{x}$ are not stochastic and do not include any randomness.
- **Stochastic problems:** At least, the object $\mathbf{x}$ is stochastic nature. The model A and observed data b can also be stochastic but unessential.
- Sometimes, stochastic nature caused by incomplete data on the model.
- The stochastic problems is sometimes a good way to study deterministic processes and its models.

Connection between deterministic and stochastic studies

- Knowing the stochastic observed data b, obtained with known stochastic object $\mathbf{x}$, it is possible to study the model A.
- Example: electron microscope probe diffraction: from differential scattering cross-section of electrons to the model A by deterministic methods.
- Example: electron microscope probe diffraction: Monte-Carlo simulation for model A determination.

Classification of Scientific Computing problems

- Forward and Inverse problems
- Well-posed and Ill-posed problems
- Static and Dynamic problems
- Deterministic and Stochastic problems
- Continuous and Discrete problems

Continuous and Discrete problems

- **Continuous problems:** all functions, distributions and models are continuous; "smooth" motion of object.
- **Discrete problems:** Events occur at discrete times, discrete spatial nodes.
- Complicated continuous problems are often being considered as discrete for possibility of numerical methods application.
- Some discrete nature problems may be reduced to continuous problems in order to find analytical solution.

Example on electronic microscopy

Electronic Microscopy contains all described above

Scheme of the experimental setup

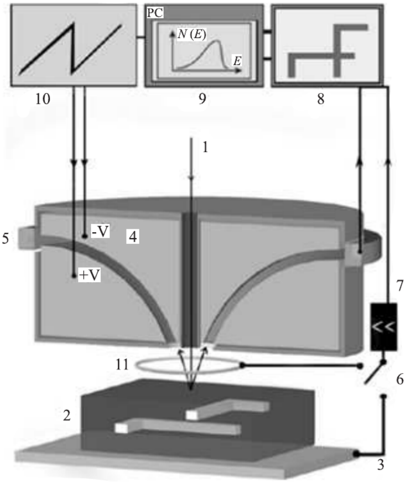


Figure: 1. Electron beam.
2. Object of research.
3. Metal base.
4. Spectrometer/energy filter.
5. Electron detectors.
6. Video controller.
7. Current supply.
8. Video out.
9. Computer.
10. Saw tooth voltage supply.
11. Electron detector.

The advantages of SEM BE

- Very high resolution images of the surface of the object.
- Possibility of the reconstruction of inner structure (tomography).
- Possibility of precise spectrographic research.
- Comparably low price.
- Additional reconstruction makes all advantages much more valuable.

Two kinds of signal:

- Image of the layer, located under the surface. The signal is being distorted due to the finite radius of electron probe, which depends on the depth of penetration into the surface:

$$r^2 = r_0^2 + 0.625\left(\frac{Z}{E_0}\right)\left(\frac{\rho}{A}\right)^{0.5} t^{1.5}.$$

- The spectrum of electrons backscattered from a target is naturally subject to distortion by a spectrometer with unusual instrument characteristics.
- Both signals are the being affected by rather high noises.

Variety of problems

- We do not consider state of the system change: **Static Problem**.
- We do consider state change (e.g. electric charge): **Dynamic Problem**.
- Modeling the image or spectrum: **Forward Problem**, **Well-posed**.
  - Modeling with Monte-Carlo: **Stochastic Problem**.
  - Modeling with convolution: **Determenistic Problem**.
- Reconstruction of spectra and images: **Inverse Problem**, **Ill-posed**.
- The problem may be considered both naturally discrete or continuous...

The problem of image restoration in SEM BE

Mathematical description:

$$u(x,y) = z(x,y) \star k(x,y) \equiv \int_B z(\xi,\eta)k(x-\xi, y-\eta)d\xi d\eta,$$

where B represents the frame, $u(x,y)$ is detected picture (signal), $z(x,y)$ real picture of the layer, and $k(x,y)$ is a hardware function:

$$k(x,y) = Az \equiv \frac{1}{2\pi r^2} exp\left(-\frac{x^2+y^2}{2r^2}\right)$$

Skoltech
Skolkovo Institute of Science and Technology

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).

Results on image restoration in SEM BE

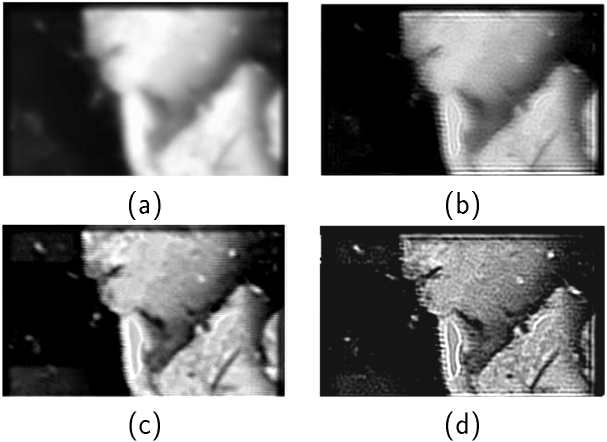


Figure: Results of image reconstruction in SEM BE: (a) - the enter data, (b) - reconstruction on H^1 Sobolev's space, analytical solution, (c) - reconstruction on TV Bounded Total Variations functional space, (MCGP, zero first approximation), (d) - reconstruction on TV with the result on H^1 taken as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

SEM BE example: sources

1. N. Koshev and L. Beilina, A posteriori error estimates for Fredholm integral equations of the first kind, *Applied Inverse Problems, Springer Proceedings in Mathematics & Statistics*, V. 48, 75-93, 2013.
   DOI 10.1007/978-1-4614-7816-4,
2. N. Koshev and L. Beilina, An adaptive finite element method for Fredholm integral equations of the first kind and its verification on experimental data, *Central European Journal of Mathematics*, V. 11, 1489-1509, 2013.
   DOI: 10.2478/s11533-013-0247-3,
3. Koshev N.A., Orlikovsky N.A., Rau E.I., Yagola A.G. Solution of the inverse problem of restoring the signals from an electronic microscope in the backscattered electron mode on the class of bounded variation functions, *Numerical Methods and Programming*, V.11, 362-367, 2011 (in Russian).
4. N.A. Koshev, F.A. Luk'anov, E.I. Rau, R.A. Sennov, and A.G. Yagola. Increasing Spatial Resolution in the Backscattered Electron Mode of Scanning Electron Microscopy, *Bulletin of the Russian Academy of Sciences. Physics*, V. 75, 1181-1184, Allerton Press, 2011.
   DOI: 10.3103/S1062873811090139,
5. S.I. Zaycev, N.A. Koshev, F.A. Lukyanov, E.I. Rau, E.B. Yakimov. Straight measurement of diameter and current density arrangement in crossover of the electron probe, *Abstracts for XXIII Rissian Conference in Electronic Microscopy*, Chernogolovka, 2010.
6. N. A. Koshev, N. A. Orlikovskii, E. I. Rau, and A. G. Yagola. Reconstructing the Energy Spectra of Backscattered Electrons with Allowance for the Spectrometer's Spread Function, *Bulletin of the Russian Academy of Sciences. Physics*, V. 76, 959-964, Allerton Press, Inc., 2012.
   DOI: 10.3103/S1062873812090109
7. A.V. Gostev, N. A. Koshev, N.A. Orlikovsky, E.I. Rau. Hardware function of the toroidal spectrometer and its influence on detected electrons spectra. *XXIV Russian electronic microscopy conference*, Chernogolovka, Russia, p. 177, 2012 (In Russian).
8. AA Borzunov, VY Karaulov, NA Koshev, DV Lukyanenko, EI Rau, AG Yagola, SV Zaitsev, 3D surface topography imaging in SEM with improved backscattered electron detector: Arrangement and reconstruction algorithm. *Ultramicroscopy, V.207*, 2019
   DOI: 10.1016/j.ultramic.2019.112830

Skoltech
Skolkovo Institute of Science and Technology

Thank you for your attention!