Lecture 1:

1) Traditional scientific and engineering approach includes only theory and experiment. Provide reasoning on disadvantages/limitations of such approach. How might the mathematical modelling help to manage it?

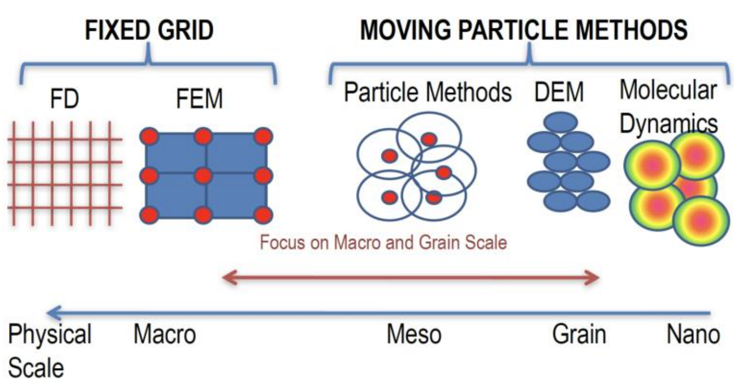
- **Limitations**
  - Too difficult - build large wind tunnels;
  - Too expensive - build a throw-away passanger jet;
  - Too slow - wait for climate of galactic evolution;
  - Too dangerous - weapons, drug design, climate experimentation.
- **Computational science and engineering paradigm**:
  Use computers 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.

2) Formulate the Moore's Law. Provide reasoning on its value for science, industry, and every-day life.

The number of transistors in an integrated circuit doubles every two years. Continued exponential increase in computational power and decrease of time in high-intensity computations.

3) List the computational approaches in terms of physical sizes scale (macro to nano). Explain the main idea meshed/meshfree approaches. Does the computational complexity of the problem depend on usage of concrete kind of mwthods with respect to the scale?

Computational methods Focus on Grain to Macro Scale Analysis



4) List the HPC units; explain the meaning of each one.

- 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)

5) What will happen when the transistor size shrinks by some factor x? Explain in terms of clock rate, unit area, raw computer power and shrinkage of performance of the program.

- 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 increase (typically another factor of x)
- Raw computing power of the chip goes up by x^4! (In practice, typically, proportionally to x^3)

6) Explain the terms of parallel speedup and parallel efficiency. Might the parallel efficiency metric be equal to 1?

$$\text{Parallel Speedup} = \frac{\text{Serial execution time}}{\text{parallel execution time}} \qquad \text{Parallel efficiency} = \frac{\text{Parallel speedup}}{P}$$

7) Formulate the Amdahl's law and provide reasoning on it: which overheads lead to limitation of the speedup?

- 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

$$\text{Speedup}(P) = \frac{\text{Time}(1)}{\text{Time(P)}} = \frac{1}{s + \frac{1-s}{P}} \qquad (1)$$

- Even if the parallel part speeds up perfectly **performance is limited by the sequential part**

8) List all overheads of Parallelism you know. Reason on the ways of overcoming some of them

- 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

9) List the limitations of serial computing. Explain each one in a few words.

- **Limits to serial computing** - both physical and practical reasons pose significant constraints to simply building ever faster serial computers.
- **Transmission speeds** - the speed of a serial computer is directly dependent upon how fast data can move through hardware. Absolute limits are the speed of light (30 cm/nanosecond) and the transmission limit of copper wire (9 cm/nanosecond). Increasing speeds necessitate increasing proximity of processing elements.
- **Limits to miniaturization** - processor technology is allowing an increasing number of transistors to be placed on a chip. However, even with molecular or atomic-level components, a limit will be reached on how small components can be.
- **Economic limitations** - it is increasingly expensive to make a single processor faster. Using a larger number of moderately fast commodity processors to achieve the same (or better) performance is less expensive.

10) Explain what do the terms ”load balance” and ”load imbalance” mean? By what can be the imbalance caused? Provide examples.

Load balance - don’t want 1000 processors to wait for one slow one. The amount of work each processor do in a parallel environment should be balanced, otherwise there will be a bottleneck caused by processors which are involved in a more complex work. Not every job is splittable into equal parts, but there are algorithms balancing the load between processes.

11) What kind of parallel computing resources do you know? List them and justify the answer: why the listed resources can be considered parallel?

The compute resources can include:

- A single computer with multiple processors;
- A single computer with (multiple) processor(s) and some specialized computer resources (GPU, FPGA etc)
- An arbitrary number of computers connected by a network;
- A combination of both.

12) List at least five applications demanding parallel computations. Justify your answer.

- Today, commercial applications are providing an equal or greater driving force in the development of faster computers. These applications require the processing of large amounts of data in sophisticated ways. Example applications include:
  - parallel databases, data mining
  - oil exploration
  - web search engines, web based business services
  - computer-aided diagnosis in medicine
  - management of national and multi-national corporations
  - advanced graphics and virtual reality, particularly in the entertainment industry
  - networked video and multi-media technologies
  - collaborative work environments
- Ultimately, parallel computing is an attempt to maximize the infinite but seemingly scarce commodity called time.

13) Describe the main idea of the von Neumann architecture

1) The Von Neumann architecture consists of a single, shared memory for both programs and data, a single bus for memory access, and a CPU.

2) Memory operations and program execution are done sequentially.

3) Data and programs are represented in binary.

14) Which kinds of software we have to use in order to write a solver for a complicated problem? Which ways (in relation to these kinds of software) can be used for optimization of your program?

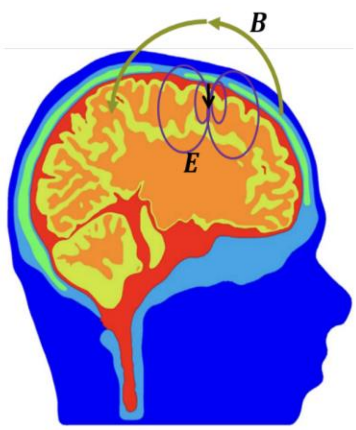
Libraries like LAPACK, ScaLAPACK, problems can be broken into chunks and parallelized. Compiler flags that leverage compiler level optimization should be applied.

15) Explain the meanings of shared and distributed memory. Explain the OpenMP and MPI paradigms. Compare them and analyze advantages and drawbacks/risks. Provide reasoning on when it is better to use one or another.

- Method used for parallelization may depend on *hardware*
- Distributed memory
  - each processor has own address space
  - if one processor needs data from another processor, must be explicitly passed
- Shared memory
  - common address space
  - no message passing required

Lecture 2:

16) What is the mathematical model of the phenomenon? How can it be represented in the operator form? Describe all parts of the latter in terms of possible representation.



- Neuronal currents occur inside the cortex.
- **Computational domain:** $\Omega \in \mathbb{R}^3$
- **Phenomenon:** generation of the electric potential by cortical currents.
- **Input:**
  - $\mathbf{J}(\mathbf{x})$ - current density
  - $\sigma(\mathbf{x}), \mathbf{x} \in$ Head - electric properties
- **Mathematical model:**

$$-div(\sigma \nabla U) = div\mathbf{J};$$

- **OUTPUT:** electric potential: $U(\mathbf{x})$.

17) What is the forward problem, and what is inverse problem? What is the difference between them? Reason on relativity of

The main equation: $A\mathbf{f} = \mathbf{b}$ or $L(\mathbf{f}, \mathbf{b}) = 0$.

- Forward problem (simulation): find the vector $\mathbf{b}$, if the input data $\mathbf{f}$ and the model A or L are known.
- Inverse problem:
  - Assuming the output $\mathbf{b}$ and the model A to be known, find the **input data f**;
  - Assuming the output $\mathbf{b}$ 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.

this classification. Explain with examples.

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'.

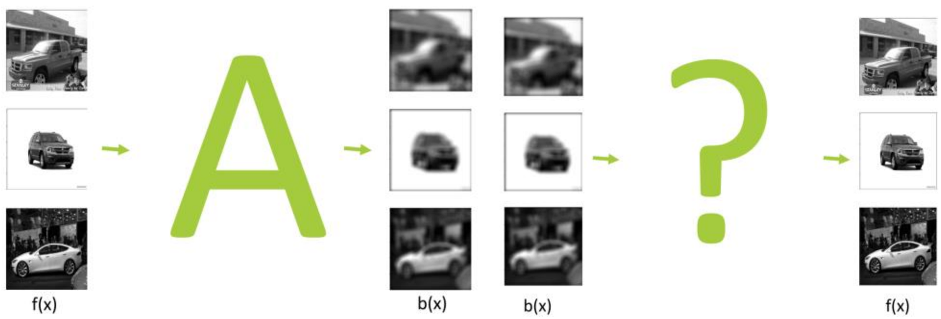
18) List several applications of both Forward and Inverse problems. Justify on why the concrete problem is Forward or Inverse.

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.**

Inverse problems: Applications

- **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.

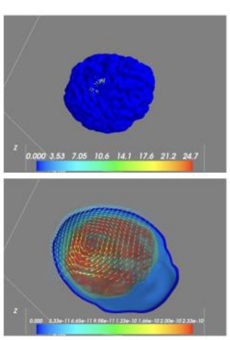


- Let $u(\mathbf{x})$ is a potential, and $\mathbf{J}(\mathbf{x})$ is the current density.
- **Forward Problem:** $\mathbf{J}(\mathbf{x}) \to u(\mathbf{x})$
- **Inverse Problem:** $u(\mathbf{x}) \to \mathbf{J}(\mathbf{x})$

19) Give the definitions and describe with examples the Top-Down and Bottom-Up approaches.

- **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: Top-Down approach.

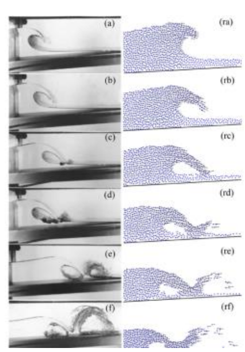


- **input data:** Current density $\mathbf{J}(\mathbf{x}), \mathbf{x} \in$ Cortex; Conductivity $\sigma(\mathbf{x}), \mathbf{x} \in$ Head.
- **The model:** Poisson equations

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

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

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.

20) Give the definitions and describe with examples direct and iterative approaches to solution of any problem.

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.

- 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.

21) List three points J.Hadamard proposed to consider in order to classify problems in terms of well- and ill-posedness. What do these points mean mathematically?

A **well posed** problem is "stable", as determined by whether it meets the three *Hadamard criteria*. These criteria tests whether or not the problem has:

1. **A solution:** a solution (*s*) exists for every data point (*d*), for every *d* relevant to the problem.
2. **A unique solution:** *s* is unique for all d; For every *d* there is at most one value of *s*.
3. **A stable solution:** *s* depends continuously on *d* (a tiny change in *d* will lead to a tiny change in *s*; and a large change in d will lead to a proportionally larger change in *s*).

The Hadamard criteria tells us how well a problem lends itself to mathematical analysis.

Examples of Well Posedness

The majority of problems we work with in calculus, engineering, and math are **well posed.** That includes problems like:

- $f(x) = x^2 + x$,
- $f(x) = 3x / 6$, and
- $f(x) = \sin(x) + 2x^2$.

For example, take $f(x) = x^2 + x$. For every real number x, $x^2 + x$ is also real and is well defined. There's no room for ambiguity; every input k will give exactly one solution; $k^2 + k$.

- If $x = 2$, $f(x) = 2^2 + 2$ or 6,
- If $x = -1$, $f(x) = (-1)^2 - 1 = 0$,

22) List the possible issues for the inverse operator in case of ill-posed problems.

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.

23) What does the continuous dependence of the solution on input means? Prove that derivative of noised function might depend discontinuously on the small error (noises) of the function to be differentiated.

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

$$||f(x) - f_\delta(x)||^2_{L^2[0,1]} = \delta^2.$$

On the other hand

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

and thus, $||\partial_x f(x) - \partial_x f_\delta(x)||^2_{L^2} = 2\pi^2\delta^2k^2$. We can see that
Assume the error $\delta = 0.01$ (1% error), and the frequency $k = 1000$.
Thus,

$$||\partial_x f_\delta - \partial_x f||^2 \approx 10^3, \quad \text{while} \quad ||f - f_\delta||^2 = 10^{-4}$$

24) What is ill-conditioned problem? Define it mathematically and give reasoning on behaviour of the solution with respect to changes in input data. Give any example of ill-conditioned problem. Explain it.

2. Ill-Posed Problem

An ill posed problem is one which **doesn't** meet the three *Hadamard criteria* for being well posed. These criteria are:

- Having a solution
- Having a unique solution
- Having a solution that depends continuously on the parameters or input data.

A problem which is not well posed is considered ill posed. Many first order differential equations and inverse problems are ill posed.

For example, consider the equation y' = (2 – y) / x. The solutions of the function are y = C/x + 2, where C is a constant. Since there are an infinite number of possible values of C, there are an infinite number of solutions, and the second Hadamard criteria is not met.

Examples of Ill Posed Problems

One simple example of an ill-posed problem is given by the equation
$y' = (3/2)y^{1/3}$ with $y(0) = 0$.
Since the solution is $y(t) = \pm t^{3/2}$, the solution is not unique (it could be plus $t^{3/2}$ or it could be minus $t^{3/2}$). As this violates rule 2 of the Hadamard criteria, the problem is ill posed.

Many inverse problems are ill-posed because either they don't have a solution everywhere, their solution is not unique, or their solution is not stable (continuous).

25) Which additional research and additional steps should or might be done in case of ill-posed problems? Reason on all possibilities you know.

- 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.

26) Define static and dynamic problems. Give at least two examples of each and explain these examples. List the cases in which a dynamic problem may be considered to be a static one. Explain these cases.

- **Static problem** is time-independent. In static problem, the observed data **b**, the model A, and the input **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 **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

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

- 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:

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

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.

Lecture 3:

27) Write down the forms of differential operator for Ordinary Differential Equation and for Partial Differential Equation. Explain the difference between them.

Differential operator

The operator equation for differential equations is often written as:

$$D\mathbf{u} = 0,$$

where D represents both mathematical model and the right-hand side; the function $\mathbf{u}$ is an unknown function to be found.

Examples

- Ordinary differential equation:

$$\frac{d\mathbf{u}}{dx} = f(x) \Leftrightarrow D \equiv \frac{d}{dx} - f(x).$$

- Partial differential equation (Poisson equation, 2D):

$$\Delta\mathbf{u} = -f(\mathbf{u}, x, y); \quad \Delta\mathbf{u} = \frac{\partial^2\mathbf{u}}{\partial x^2} + \frac{\partial^2\mathbf{u}}{\partial y^2}; \quad D\mathbf{u} \equiv \Delta\mathbf{u} + f(\mathbf{u}, x, y)$$

Differential operator

- **Ordinary differential equation (ODE):** Let $x \in \mathbb{R}^1$. ODE of the k-th order can be represented with the operator:

$$D\mathbf{u} = F\Big(x, u(x), u'(x), u''(x), ..., u^{(k)}(x)\Big).$$

- **Partial differential equation (PDE):** Let $\mathbf{x} \in \mathbb{R}^n \equiv (x_1, x_2, ..., x_n)$. PDE of the k-th order can be represented with the following operator:

$$D\mathbf{u} = F\Big(\mathbf{x}, \mathbf{u}(\mathbf{x}), \frac{\partial\mathbf{u}}{\partial x_1}, ..., \frac{\partial\mathbf{u}}{\partial x_n}, ... \frac{\partial^2\mathbf{u}}{\partial x_1^2}, ..., \frac{\partial^2\mathbf{u}}{\partial x_n^2}, ... \frac{\partial^{(k)}\mathbf{u}}{\partial x_1^{(k)}}, \frac{\partial^{(k)}\mathbf{u}}{\partial x_n^{(k)}}\Big).$$

28) Explain the meaning of terms: Linear differential equation, Quasi-linear differential equation, Non-linear differential equation. Provide examples of these equations.

- The function $F(...)$ usually satisfies the regularity conditions: measurability, differentiability, continuity etc.
- **Linear equations:** If the function F depends linearly on $\mathbf{u}$ and its derivatives, then the ODE/PDE is linear.
- **Quasi-linear equations:** the function F linearly depends on higher-order derivatives.
- **Non-linear equations:** the most complicated case. The function F depends non-linearly on higher order (and, maybe, all other) derivatives.

29) How the ODE of k-th order can be replaced with a system of ODE of the first order?

Ordinary Differential Equation (ODE) of p^{th} order can be written in as follows:

$$F\Big(x, u'(x), u''(x), ..., u^{(p)}(x)\Big) \equiv D\mathbf{u} = 0.$$

Solve the equation above with respect to higher order derivative $u^{(p)}$ to express it via all other derivatives:

$$u^{(p)}(x) = f(x, u, u', u'', ..., u^{(p-1)}).$$

We refer to the fact that with $u^{(k)}(x) \equiv u_k(x)$ and $u(x) = u_0(x)$, the latter equation may be represented with a system of the first-order ODEs:

$$u_{k+1}(x) = u_k^{'}(x), \quad 0 \le k \le p-2,$$

$$u_p^{'}(x) = f(x, u_0, u_1, ..., u_{p-1}),$$

30) Give the definitions for three kinds of problems for differential equations (Cauchy problem, BV problem, eigenvalue problem). Explain each of them in a few words; provide a reasoning on how much initial/boundary conditions are needed to make each of these problems well-determined?

Let $\mathbf{u} = \{u_0, ..., u_p\}$, and $\mathbf{f} = \{f_1, ..., f_p\}$ By analogy, each system of differential equations of any order can be changed with equivalent system of the first-order ODE:

$$u_k^{'} = f(x, u_0, u_1, ..., u_p),$$

Or, the vector form of the system above:

$$\mathbf{u}'(x) = \mathbf{f}(x, \mathbf{u}(x)), \quad \mathbf{u} = \{u_0, ..., u_p\}, \mathbf{f} = \{f_1, ..., f_p\}.$$

There are three kinds of problem related to the ODEs

- The Cauchy problem
- Boundary value problem
- Eigenvalues problem

ODE: The Cauchy problem

- Vector form:

$$\mathbf{u}'(x) = \mathbf{f}(x, \mathbf{u}(x)), \quad x \in [\xi, X]$$
$$\mathbf{u}(\xi) = \eta, \quad \text{or } u_k(\xi) = \eta_k, 1 \le k \le p.$$

- Or common form:

$$F\Big(x, u(x), u'(x), ..., u^{(p)}(x)\Big) \equiv D\mathbf{u} = 0.$$
$$u^{(k)}(\xi) = \eta_k, \quad 0 \le k < p$$

- The conditions may be considered as the definition of some initial point $(\xi, \eta_0, ..., \eta_{p-1})$ for an integral curve in $(p+1$-dimensional space $(x, u_0, ..., u_p)$.

Augustin-Louis Cauchy
1789-1857

Boundary value problems

The system of ODE:

$$u_k^{'}(x) = f_k(x, u_1, u_2, ..., u_p), \quad 1 \le k \le p, \quad x \in [a, b]$$

In the Boundary Value problem, the conditions are being set both in points $x = a$ and $x = b$.

- Possible with $p \ge 2$.
- Approximate methods: Fourier-based methods, Ritz methods, Galerkin methods.
- Numerical methods: Shooting method, Finite Differences.

Eigenvalue problems

The Eigenvalue problem:

$$\mathbf{u}'(x) = \mathbf{f}(x, \mathbf{u}; \lambda_r, r = 1, ..., q);$$
$$\mathbf{u} = \{u_1, u_2, ..., u_p\}, \quad \mathbf{f} = \{f_1, f_2, ..., f_p\}.$$

- Need to find both u_k and λ_r.
- Additional (boundary) conditions needed: $p + q$.
- The solution: u_k and λ_r - eigenfunctions and eigenvalues.
- λ_k is also called **spectrum**.

31) What is the Cauchy problem? Describe it with an examples. When it has the solution, and when this solution is unique?

The simplest Cauchy problem

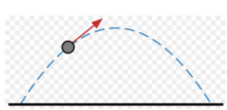
Consider a ball with mass m. Let this ball be thrown; we need to calculate its trajectory.

- m - the mass of the ball
- $\mathbf{x} \in L_2(\mathbb{R}^2) \times L_2(\mathbb{R}^2)$
- $\mathbf{v} \in L_2(\mathbb{R}^2) \times L_2(\mathbb{R}^2)$
- The governing equation:

$$\mathbf{x}''(t) = \mathbf{g},$$
$$\mathbf{x}'(t) = \mathbf{v}(t),$$

where $\mathbf{g}$ is a gravitational acceleration.

- Initial conditions: $\mathbf{x}(0) = \mathbf{x}_0; \mathbf{x}'(0) \equiv \mathbf{v}(0) = \mathbf{v}_0$



The solution of the problem can be easily found using integration:

$$\mathbf{v}(t) = Const + \int_0^t \mathbf{g}d\tau = \mathbf{v}_0 + \mathbf{g}t,$$

$$\mathbf{x}(t) = Const + \int_0^t \mathbf{v}(\tau)d\tau = \mathbf{x}_0 + \mathbf{v}_0 t + \mathbf{g}t^2$$

Since in our case $\mathbf{g}$ has no x-component ($\mathbf{g} = (0, -9.8)^T$), then, as we can see, the x component of the speed will be fixed. But...

- What if we have not only one body (ball)?
- What if we have air?
- What if we have other forces affecting the bodies?

32) What is the boundary value problem? Describe it with an example

Boundary value problem

- Simulation of loaded string (static deflection).

$$u''(x) = -f(x), \quad a \le x \le b, u(a) = u(b) = 0,$$

  where $f(x)$ is a deforming force.
- Ballistic trajectory in space:

$$\mathbf{x}''(t) = \mathbf{g}, \quad \mathbf{x}(a) = \mathbf{x}_a, \mathbf{x}(b) = \mathbf{x}_b, |\mathbf{x}'(t)| = v_0.$$

33) What is the eigenvalue problem? Describe it with an example.

Oscillations of the string

The wave (D'Alembert) equation:

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

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.

Ignoring the right boundary condition, we have the Cauchy problem

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

the solution of which $y(x;k)$ depends on the values of the parameter k.
The eigenvalues may be found then by changing the parameter k until the right boundary value be satisfied. Here the parameter k represents the proper (eigen) frequency of the string.

34) List three general approaches for solution of the Cauchy problem. Explain in few words advantages and disadvantages of them.

- **Exact methods:** the solution of the Cuachy problem may be represented with elementary functions. Rarely applicable even for simple equations. For example, the solution of the equation $u'(x) = x^2 + u^2(x)$ can not be presented with elementary functions (or just not reasonable).
- **Approximate methods:** the solution $\mathbf{u}(x)$ is representable as a limit of some sequence $y_n(x)$, members of which can be represented with elementary functions or with quadratures. Applicable only for relatively simple linear problems.
- **Numerical solution:** applicable for a wide class of Cauchy problems. The problem must be well-conditioned.

35) What is explicit and implicit schemes for solving the Cauchy problem? Explain advantages and disadvantages for all of them.

- Numerical methods, being used with meshes.
- **Explicit methods** are commonly more accurate; there are some equations, however, which are not reasonable to solve with explicit schemes due to low stability/accuracy/performance. These equations called 'stiff equations'
  - Euler's method;
  - Runge-Kutta 4^{th} order method;
- **Implicit methods** are less accurate and somewhat slower some ODE; however, due to better stability, implicit methods are capable for wider class of problems.

36) Explain the idea of Euler's method.

The Euler's method

The Cauchy problem:

$$u'(x) = f(x, u(x)), \quad \xi \le x \le X, \quad u(\xi) = \eta.$$

Let $\{x_n, 0 \le n \le N\}$ be some mesh covering the interval $[\xi, N]$.
Using the Taylor's series for the interval $[x_n, x_{n+1}]$:

$$u_{n+1} = u_n + h_n u_n^{'} + \frac{1}{2} h_n^2 u_n^{''} + ..., \quad h_n = x_{n+1} - x_n$$

The equation states that $u'(x) = f(x, u)$. Assuming the steps h are small and ignoring higher order derivatives multiplied with $h_n^t, t > 1$, we obtain the Euler's method. The accuracy of the method is $O(max h_n)$.

37) Explain the idea of Shooting method.

The Shooting method

Ignoring the right boundary condition, we have the Cauchy problem

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

the solution of which $y(x; k)$ depends on the values of the parameter k.
The eigenvalues may be found then by changing the parameter k until the right boundary value be satisfied. Here the parameter k represents the proper (eigen) frequency of the string.

38) ~~Write at least three real-world examples of problems leading to the partial differential equations of the first order. Analyze your examples in terms of linearity.~~ DELETED

39) Can the Cauchy problem be stated for the partial differential equation? If yes, give an example of it.

The ill-posed Cauchy problem for heat transfer

The ill-posed Cauchy problem for heat transfer equation.

Suppose both initial condition $f(x)$ and the left boundary condition $g(t)$ to be unknown, and the function $q(t) = \partial u / \partial x(1, t)$ to be known:

$$u_t = u_{xx} + aS(u) + F(x, t), \quad (x, t) \in [0, 1]^2,$$
$$u(1, t) = p(t), \quad \frac{\partial u}{\partial x}(1, t) = q(t).$$

Find $u(x, t)$.

The problem is non-linear (quasi-linear), which leads that the cost functional has local minima

40) Describe the main idea of statement of N-body problem.

N-body problem

Consider a number of point-bodies $B_i, i = 1, ..., N$, defined with the coordinates $x_i(t)$ at the moment of time t. Each body interacts with other bodies via the gravitation:

$$m_i \ddot{x}_i(t) = \sum_{j \neq i}^{N} G m_j m_i \frac{x_j(t) - x_i(t)}{|x_j(t) - x_i(t)|^3}, \quad 1 \leq i, j \leq N.$$

In order to state the problem, we also need to define the initial body positions with known coordinates $\tilde{x}_i$:

$$x_i(t_0) = \tilde{x}_i,$$

and the initial velocities with known velocities:

$$v_i(t_0) \equiv \dot{x}_i(t_0) = \tilde{v}_i.$$

Thus, we have the Cauchy problem.

41) Describe the main idea of the Molecular dynamics method.

The Molecular Dynamics (MD) method

Microcanonical ensemble: the system is isolated from changes in moles (N), volume (V), and energy (E): adiabatic process with no heat exchange.

Newton's law of motion:

$$F(X) = -\nabla U(X) = M\dot{V}(t)$$
$$V(t) = \dot{X}(t).$$

Here:

- X are the particle coordinates.
- $U(X)$ is a potential energy of the system (comes from pair or many-body potentials).
- M - mass of particles.
- V - velocities of particles.

42) Describe the main idea and general pipeline for the Galerkin method.

The Galerkin method

Consider the equation:

$$A(u(x)) = f(x), \quad a \leq x \leq b, u(a) = u_a, u(b) = u_b.$$

- Approximate the function $u(x) \approx y_n(x) = \varphi_0(x) + \sum_{k=1}^{n} c_k \varphi_k(x)$, where the continuous function $\varphi_0(x)$ satisfies the boundary conditions: $\varphi_0(a) = u_a, \varphi_0(b) = u_b$; the functions $\varphi_k(x), 1 \leq k < \infty$ are linearly independent and vanish at points a and b: $\varphi_k(a) = \varphi_k(b) = 0$.
- The algebraic system of equations for the coefficients c_k:

$$\int_a^b \Big(A(y_n(x)) - f(x)\Big)\varphi_k(x)dx = 0, \quad 1 \leq k \leq n$$

- After definition of the coefficients c_k, we construct the approximate solution of the BV problem.

Lecture 4:

43) Give the definition of convolution. Explain its meaning in terms of point spread function. Provide real-world examples.

The convolution

- Let the functions $k(\mathbf{x})$ and $f(\mathbf{x})$ are both defined for $\mathbf{x} \in \mathbb{R}^n$.
- The function $g(\mathbf{x})$ is called convolution of k and f if:

$$g(\mathbf{x}) = k * f \equiv \int_{\mathbb{R}^n} k(\mathbf{x} - \xi) f(\xi) d\xi.$$

- The function $k(\mathbf{x})$ is the convolution kernel.
- The convolution means than each value $f(\mathbf{x})$ becomes some spot defined with $k(\mathbf{x})$. Thus, the function $k(\mathbf{x})$ is also sometimes called the **Point spread function**
- In the problems of image processing, the kernel k is often being defined by hardware function.
- If $\hat{\cdot}$ denotes the Fourier transform of ., then:

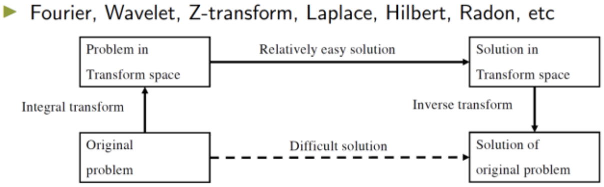
$$\widehat{k * f} = \hat{k}\hat{f}.$$

44) Write the generalized form of an integral transform; describe its parts. Why do we use the integral transforms? What are we expecting of usage of them?

Integral transforms

$$g(\mathbf{x}) = \int_{\Omega} K(\mathbf{x}, \xi) f(\xi) d\xi, \quad \mathbf{x}, \xi \in \Omega \subset \mathbb{R}^n$$

- $K(\mathbf{x}, \xi)$ - the Kernel;
- Mapping a function $f(\mathbf{x})$ in $\mathbf{x}$-space into another function $g(\xi)$ in ξ-space
- Fourier, Wavelet, Z-transform, Laplace, Hilbert, Radon, etc



45) Write the definition of the Fourier transform and inverse Fourier transform. Explain the differences of real domain and reciprocal (frequency) domain. Provide examples.

Fourier Transform

- Each function defined in spatial or time domain (**real** domain) by some function $h(t)$, can be presented also in **frequency** or **reciprocal** domain with a function $H(f)$ of frequency f.
- One goes back and forth between these two representations by means of the Fourier transform,

$$H(f) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} h(t) e^{-2\pi i f t} dt; \quad h(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} H(f) e^{2\pi i f t} df;$$

Or, using angular frequency $\omega = 2\pi f$:

$$H(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} h(t) e^{-i\omega t} dt; \quad h(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} H(\omega) e^{i\omega t} d\omega$$

46) What is the Fourier convolution theorem? Describe why, where and when this property is useful.

Why? Frequency analysis, simplification of convolution, filtering

Where? Image and signal processing, Signal & Image Analysis

When? Solving differential equations and integrals

Fourier Transform: properties

- Scaling and shifting:

$$h(t) \Leftrightarrow H(f)\text{: Fourier Transform}$$
$$h(at) \Leftrightarrow \frac{1}{|a|}H\left(\frac{f}{a}\right)\text{: time scaling}$$
$$\frac{1}{|b|}h\left(\frac{t}{b}\right) \Leftrightarrow H(bf)\text{: frequency scaling}$$
$$h(t-t_0) \Leftrightarrow H(f)e^{2\pi i f t_0}\text{: time shifting}$$
$$h(t)e^{-2\pi i f_0 t} \Leftrightarrow H(f-f_0)\text{: frequency shifting}$$

- Convolution theorem:

$$g * h \equiv \int_{-\infty}^{\infty} g(x)h(x-\xi)d\xi \Leftrightarrow G(f)H(f).$$

47) Explain the pipeline of the spectral analysis with the Fourier transform. Which properties of Fourier Transform are meaningful in this context? Explain with examples.

48) How do we use Fourier transform for calculation of cross-correlation of two functions?

Fourier Transform: properties

Cross-correlation of two functions:

$$g \star h = \int_{-\infty}^{\infty} g(x+\xi)h(\xi)d\xi$$

- Correlation theorem; Autocorrelation, Wiener-Khinchin theorem::

$$g \star h \Leftrightarrow G(f)H^*(f), \quad \Rightarrow \quad g \star g = |G(f)|^2$$

49) How the derivatives of some function could be explained with its Fourier image? How can we apply this property to the differential equations? Explain with example.

Fourier Transform: properties

- Derivatives:

$$\mathcal{F}\left[h'(t)\right] = -i\omega H(f); \quad \mathcal{F}\left[h^{(n)}\right] = (-i\omega)^n H(f).$$

- Application: D'Alambert equation

$$\frac{\partial^2 y(x,t)}{\partial x^2} = \frac{1}{\nu^2}\frac{\partial^2 y(x,t)}{\partial t^2}, \quad y(x,0) = f(x)$$

- After application of FT to both sides of this equation:

$$(-i\omega)^2 Y(\omega,t) = \frac{1}{\nu^2}\frac{\partial^2 Y(\omega,t)}{\partial t^2}$$
$$F(\omega) = Y(\omega,0) = \mathcal{F}\left[f(x)\right]$$

50) What is Parseval's theorem? When is it important?

- Parseval's theorem:

$$||h(t)||_{L^2} = ||H(f)||_{L^2} \Rightarrow \int_{-\infty}^{\infty} |h(t)|^2 dt = \int_{-\infty}^{\infty} |H(f)|^2 df$$

51) Why FT is that important? List all the applications of FT you know. Provide at least two examples.

Main Applications of FFT

- Signal & Image Analysis (e.g. finding periodicities, feature extraction, correlation analysis etc)
- Signal & Image Processing (e.g. compression, de-noising etc)
- Convolution operations (mind the the convolution theorem and $O(NlogN)$ scaling)
- Numerical Solution of Integral Equations (discretization in the reciprocal domain)
- Treatment of long-range potential problem in N-body simulations (electrostatic and gravitational interactions)

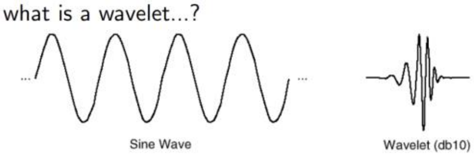
52) What is the Wavelet Transform? Give the definition and describe motivation.

Why wavelets?

Or: What's wrong with Fourier?

- By using Fourier Transform , we loose the time information : WHEN did a particular event take place ?
- FT can not locate drift, trends, abrupt changes, beginning and ends of events, etc.
- Calculating use complex numbers.

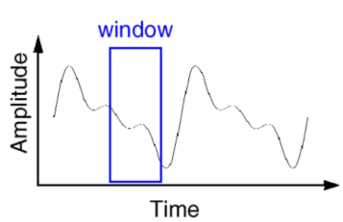
- And what is a wavelet...?



- A wavelet is a waveform of effectively limited duration that has an average value of zero.
- Short time localized waves with zero integral value.
- Possibility of time shifting.
- Flexibility.

53) Write the definition of the STFT. Write its advantages and disadvantages. Provide at least one possible application.

In order to analyze small section of a signal, Denis Gabor (1946), developed a technique, based on the FT and using windowing : STFT



- A compromise between time-based and frequency-based views of a signal.
- both time and frequency are represented in limited precision.
- The precision is determined by the size of the window.
- Once you choose a particular size for the time window - it will be the same for all frequencies.

54) Describe step-by-step the idea of wavelet analysis. What is similar between the Fourier and continuous wavelet transforms?

For easier understanding, consider the 1D wavelets in a time domain.

- $x(t)$ - time-evolving signal.
- The Mother Wavelet: $\Psi(t)$.
- Daughter Wavelets: $\Psi(\frac{t-b}{a})$ - shifted and scaled versions of the MW.
- Continuous Wavelet Transform (CWT):

$$X(a,b) = \frac{1}{|a|^{1/2}} \int_{\mathbb{R}} dt x(t) \Psi^*\left(\frac{t-b}{a}\right)$$

- Continuous Inverse Wavelet Transform:

$$x(t) = \frac{1}{C_\Psi} \iint_{\mathbb{R}^2} X(a,b) \frac{1}{|a|^{1/2}} \tilde{\Psi}\left(\frac{t-b}{a}\right) db \frac{da}{a^2}$$

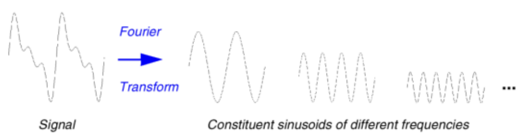
- C_Ψ - the admission constant; $\tilde{\Psi}$ - dual wavelet.

- A mathematical representation of the Fourier transform:

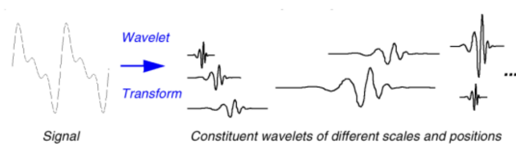
$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{i\omega t} dt$$

- Meaning: the sum over all time of the signal $f(t)$ multiplied by a complex exponential, and the result is the Fourier coefficients $F(\omega)$

- Those coefficients, when multiplied by a sinusoid of appropriate frequency ω yield the constituent sinusoidal component of the original signal:



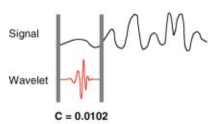
- And the result of the CWT are Wavelet coefficients .
- Multiplying each coefficient by the appropriately scaled and shifted wavelet yields the constituent wavelet of the original signal:



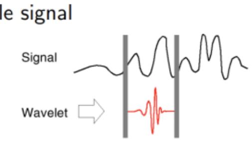
- Reminder: The CWT Is the sum over all time of the signal, multiplied by scaled and shifted versions of the wavelet function

Step 1: Take a Wavelet and compare it to a section at the start of the original signal.

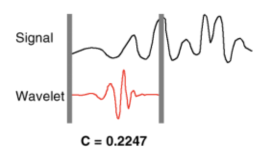
Step 2: Calculate a number, C, that represents how closely correlated the wavelet is with this section of the signal. The higher C is, the more the similarity.



Step 3: Shift the wavelet to the right and repeat steps 1-2 until you've covered the whole signal



Step 4: Scale (stretch) the wavelet and repeat steps 1-3



55) Write the general classification of integral equations. Which integral equations are well-posed, and which type of them are often ill-posed?

The linear integral equation generally is

$$g(\mathbf{x})y(\mathbf{x}) - Ay(\mathbf{x}) = f(\mathbf{x})$$

Here A is an integral operator, which can be expressed as follows:

$$Ay(\mathbf{x}) = \lambda \int_{\Omega} K(\mathbf{x},\xi) y(\xi) d\xi$$

- Ω is some domain (may represent the whole space).
- $K(\mathbf{x},\xi)$ is a Kernel.
- $y(\mathbf{x})$ is unknown function to be found.
- $f(\mathbf{x})$ is a right-hand side.

- **The integral equation of the first kind:** $g(\mathbf{x}) \equiv 0$.

$$Ay(\mathbf{x}) = f(\mathbf{x})$$

These equations are commonly ill-posed.

- **The integral equation of the second kind:**

$$y(\mathbf{x}) - \lambda Ay(\mathbf{x}) = f(\mathbf{x}).$$

These equations are commonly well-posed.

Lecture 5:

56) Describe four steps for computational solution of the scientific computing problem with usage of a mesh.

57) Suppose the computational domain is covered with a uniform mesh. How can you approximate the first and second derivatives of some function (defined in this domain) with FDM?

- Let the computational domain be $\Omega \subset \mathbb{R}^r$
- All r−dimensional points to be processed are therefore belong to this domain: $\mathbf{x} \in \Omega$
- **Nodes:** A finite set of points $\{x_n\}$ being considered in real computations. Here n is a multiindex.
- $x_n = (x^1_{n_1}, x^2_{n_2}, ..., x^r_{n_r})$; $n_i = 0, ..., N_i - 1$, where N_i is a number of nodes at i-th dimension.
- For example, for $r = 3$, the nodes will be denoted as: $x_{ijk}, i = 0, ..., N_x - 1, j = 0, ..., N_y - 1, k = 0, ..., N_k - 1$.
- The falue of some function depending on x will be denoted as: $f_n = f(x_n)$. For $r = 3$ it then will be f_{ijk}.

Approximation of the first derivatives. For short notations, consider the 2d case:

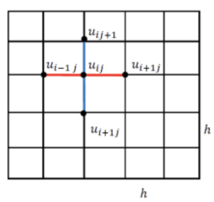
$$x_i = ih_x, \quad y_j = jh_y, \quad f_{ij} = u(x_i, y_j).$$

- Backward difference:
$$\frac{\partial u}{\partial x}(x_i, y_j) \equiv \left(\frac{\partial u}{\partial x}\right)_{ij} \approx \frac{u_{ij} - u_{i-1,j}}{h}.$$
- Forward difference:
$$\frac{\partial u}{\partial x}(x_i, y_j) \equiv \left(\frac{\partial u}{\partial x}\right)_{ij} \approx \frac{u_{i+1,j} - u_{ij}}{h}.$$
- Symmetric difference:
$$\frac{\partial u}{\partial x}(x_i, y_j) \equiv \left(\frac{\partial u}{\partial x}\right)_{ij} \approx \frac{u_{i+1,j} - u_{i-1,j}}{2h}.$$
- The same is for other variables;

FDM: Higher derivatives approximation illustration

$$u_{xx}(x_i, y_j) \approx \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h_x^2}.$$

$$u_{yy}(x_i, y_j) \approx \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{h_y^2}.$$



The Poisson equation in this approximation takes the form

$$-\left(\frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h_x^2} + \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{h_y^2}\right) = f_{ij}.$$

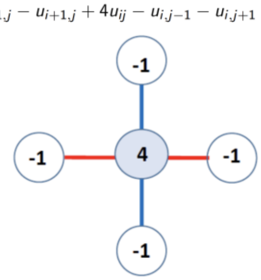
- The boundary conditions mean we can exclude variables u_{ij} for indices i, j belonging to the boundary, changing them with correspondent values φ_{ij}.
- The scheme has a square matrix: easy to see that indices vary such that $i = 1, ..., N_i - 2$, $j = 1, ..., N_j - 2$, which means $(N_i - 2) * (N_j - 2)$ equations. Since the boundary values are excluded, so we have the same number of unknown variables.

58) Describe the "cross" template for approximation of second derivatives; write the approximation of the boundary value problem for the Poisson equation using this template. Analyze the obtained SLAE with respect to number of equations and variables.

The Cross

The Cross template consists of five nodes. If the steps $h_x = h_y = h$ (the most often case):

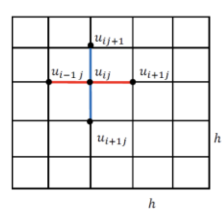
$$-u_{i-1,j} - u_{i+1,j} + 4u_{ij} - u_{i,j-1} - u_{i,j+1} = h^2 f_{ij}$$



- **The Poisson equation in this approximation takes the form**
$$-\left(\frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h_x^2} + \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{h_y^2}\right) = f_{ij}.$$
- The boundary conditions mean we can exclude variables u_{ij} for indices i, j belonging to the boundary, changing them with correspondent values φ_{ij}.
- The scheme has a square matrix: easy to see that indices vary such that $i = 1, ..., N_i - 2$, $j = 1, ..., N_j - 2$, which means $(N_i - 2) * (N_j - 2)$ equations. Since the boundary values are excluded, so we have the same number of unknown variables.

$$u_{xx}(x_i, y_j) \approx \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h_x^2}.$$

$$u_{yy}(x_i, y_j) \approx \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{h_y^2}.$$



The Poisson equation in this approximation takes the form

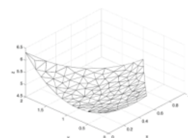
$$-\left(\frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h_x^2} + \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{h_y^2}\right) = f_{ij}.$$

60) Describe the procedure (steps) of reduction the problem for differential equation to SLAE using FDM.

The computational domain: $\Omega \subset \mathbb{R}^n$, $\quad n = 2$ or $n = 3$.
The boundary: $\partial\Omega$: the boundary of computational domain;

$$-\Delta u(\mathbf{x}) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad , u|_{\partial\Omega} = \varphi(\mathbf{x}).$$

- Cover the computational domain with some mesh.
- Define the locations of known and unknown variables.
- Construct the system of equations to solve numerically.
- Solve the SLAE:



$$u_h = A^{-1} g$$

61) Describe general properties of FDM matrix for the elliptic equations, obtained using the Cross template.

Finite-difference SLAE properties:

- The system contains a big number of unknown variables and equations
- The matrix is square, sparse, invertible, symmetric, positive definite
- Big condition number (often ill-conditioned):

$$\kappa(A) = O\left(\frac{1}{h^2}\right)$$

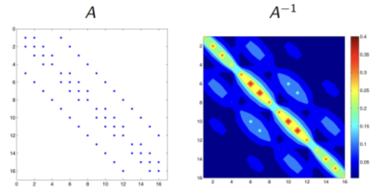
62) Why it is better to avoid usage of inverse matrices in real computations of Poisson equation with the Cross template?

SLAE: Inverse matrix

- $H = A^{-1}$ - approximates fundamental (Green's) function.
- H_{ij} - solution in the node i with the point source, located in the node j.
- The matrix A^{-1} is dense while the matrix A is sparse.

It's better to avoid usage of inverse matrices in real computations.

A $\qquad$ A^{-1}

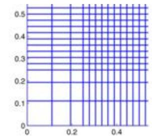


63) Explain the meaning of terms 'implicit scheme' and 'explicit scheme'. Which advantages and drawbacks has each of these schemes? Provide an example of implicit and explicit schemes for some differential equation.

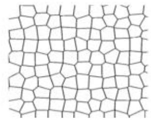
- Numerical methods, being used with meshes.
- **Explicit methods** are commonly more accurate; there are some equations, however, which are not reasonable to solve with explicit schemes due to low stability/accuracy/performance. These equations called 'stiff equations'
  - Euler's method;
  - Runge-Kutta 4^{th} order method;
- **Implicit methods** are less accurate and somewhat slower some ODE; however, due to better stability, implicit methods are capable for wider class of problems.

64) Compare FDM and FEM in terms of advantages and drawbacks of each method.

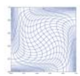
FDM: rectangular cells: fast and simple mesh construction; issues: discontinuous coefficients, complex geometries

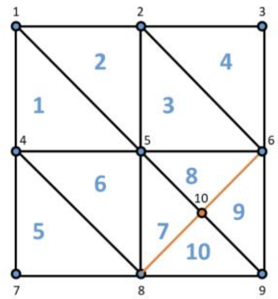


Finite Volumes Method: more flexible with respect to geometry



Finite Elements Method: Unstructured grids, complex geometries, stability

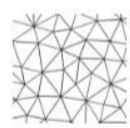


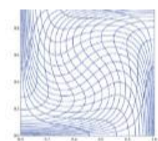


- The number of nodes here: $N = 10$;
- The number of elements here: $N_e = 10$;
- Sometimes, we are unable to enumerate nodes with appropriate order. Such grids called **unstructured grids.**

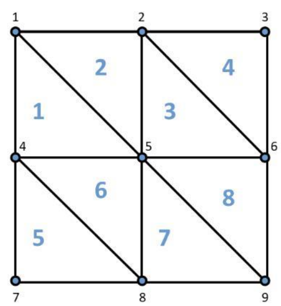
65) Define the FEM approximation with basis functions. List necessary properties of the basis functions. Write the approximations or derivatives and integrals.

Finite Elements Method: Unstructured grids, complex geometries, stability

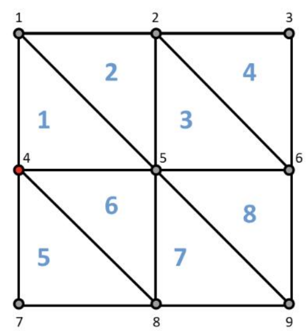




The FEM mesh notations



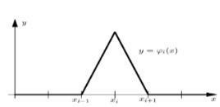
- The number of nodes: N (here $N = 9$);
- The nodes: $\mathbf{x}_i \in \mathbb{R}^d$, for $i = 0, ..., N-1$.
- The Mesh Cell (Triangle, tetrahedron, etc.): $K_i, 0 = 1, ..., N_e - 1$
- The number of mesh cells: N_e (here $N_e = 8$);

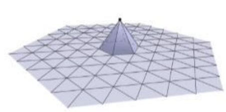


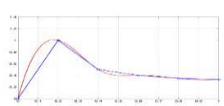
- Introduce **the basis:** the set of continuous functions $\varphi_i(\mathbf{x}), i = 0, ..., N-1$, associated with the nodes, such that:

$$\varphi_i(\mathbf{x}) = \begin{cases} 1, & \text{at } i\text{-th node} \\ 0, & \text{at other nodes} \end{cases}$$

- Each basis function is connected to the node: $\varphi_j(\mathbf{x}), j = 0, ..., N-1$.
- The basis functions being sometimes called **Elements**







- **The function $u(\mathbf{x})$ can be approximated as follows:**

$$u(\mathbf{x}) \approx \sum_{i=0}^{N-1} u_i \varphi_i(\mathbf{x})$$

- **The derivatives:** $\nabla u(\mathbf{x}) = \sum_{i=0}^{N-1} u_i \nabla \varphi_i(\mathbf{x})$
- **The integrals:**

$$\int_\Omega u(\mathbf{x}) d\mathbf{x} \approx \sum_{i=0}^{N-1} u_i \int_\Omega \varphi_i(\mathbf{x}) d\mathbf{x}, \quad \int_\Omega \nabla u(\mathbf{x}) d\mathbf{x} \approx \sum_{i=0}^{N-1} u_i \int_\Omega \nabla \varphi_i(\mathbf{x}) d\mathbf{x}$$

66) What is the weak formulation? Describe with the example.

The weak formulation is often used in the context of solving boundary value problems for elliptic Partial Differential Equations, such as the Poisson equation.

Strong form of the Poisson equation:

Poisson equation 2D example: solution steps

The computational domain: $\Omega \subset \mathbb{R}^n$, $n = 2$ or $n = 3$.
The boundary: $\partial\Omega$: the boundary of computational domain;

$$-\Delta u(\mathbf{x}) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad , u|_{\partial\Omega} = \varphi(\mathbf{x}).$$

Weak form of the Poisson equation:

Let F be a Hilbert space. The equation to be solved:

$$Au = f, \quad u, f \in F, A : F \to F.$$

The latter equation is an equivalent to finding $u \in F$ such that:

$$(Au, v)_F = (f, v)_F \quad \forall v \in F$$

The function v is called a test function. For Poisson equation it means, since $\Delta u \in L_2(\Omega)$, that:

$$(\Delta u, v)_{L_2(\Omega)} = -(f, v)_{L_2(\Omega)}.$$

67)Describe the procedure of reduction of the Poisson equation to the SLAE using FEM.

68) List the properties of the matrix of SLAE, obtained using the FEM for the Poisson equation.

0 **Original equation:** $Au = f, \quad u \in U, f \in F$

1 **Weak formulation:** $(Au, v)_F = (f, v)_F, \quad \forall v \in F$

2 **Approximation (assuming linearity of A):**

$$u(\mathbf{x}) \approx \sum_{i=0}^{N-1} u_i \varphi_i(\mathbf{x}) \quad \Longrightarrow \quad \sum_{i=0}^{N-1} u_i \left(A\varphi_i, v\right)_F = (f, v)_F$$

3 **The system:** since $v \in F$ is a certain function, we put $v = \varphi_j, \quad j = 0, 1, ..., N-1$ in order to obtain N equations with respect to unknown u_i:

$$\sum_{i=0}^{N-1} u_i \left(A\varphi_i, \varphi_j\right)_F = (f, \varphi_j)_F$$

4 Solve the system above.

5 Profit!

The system properties and methods

- The system contains $10^6 - 10^7$ equations and the same number of unknown variables;
- The matrix of the system is ill-conditioned;
- It's, however, symmetric and sparse;
- The suitable method to solve: generalized residual method with regularization.

Properties of the matrix

- Square matrix
- Sparse and symmetric
- $\kappa(A) = O\left(\frac{1}{h^2}\right)$

Lecture 6:

69) Write the general concept of Optimization problem. Define linear and non-linear optimization.

General problem statement

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

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

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

70) List and give explanations of the applications of the optimization from point of view of mathematical problems reduction.

Deconvolution problem

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

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

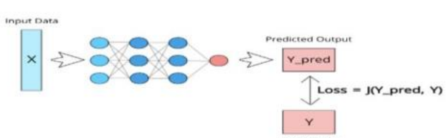
- In case of finite and noisy functions u and K, the latter equation is ill-posed.
- The problem can be reduced to the optimization problem for the cost Tikhonov's functional

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!

71) Define the unconstrained and constrained optimization. Give a simple examples and think on where it can be applied.

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

Example of Unconstrained Optimization:

Consider a simple case where you want to find the minimum of a quadratic function f(x) = ax^2 + bx+c. You want to find such value of x that will minimize the function f(x) - without any constrains for x.

Machine Learning: Training machine learning models often involves unconstrained optimization to find the best model parameters that minimize a loss function

Example of Constrained Optimization:

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:

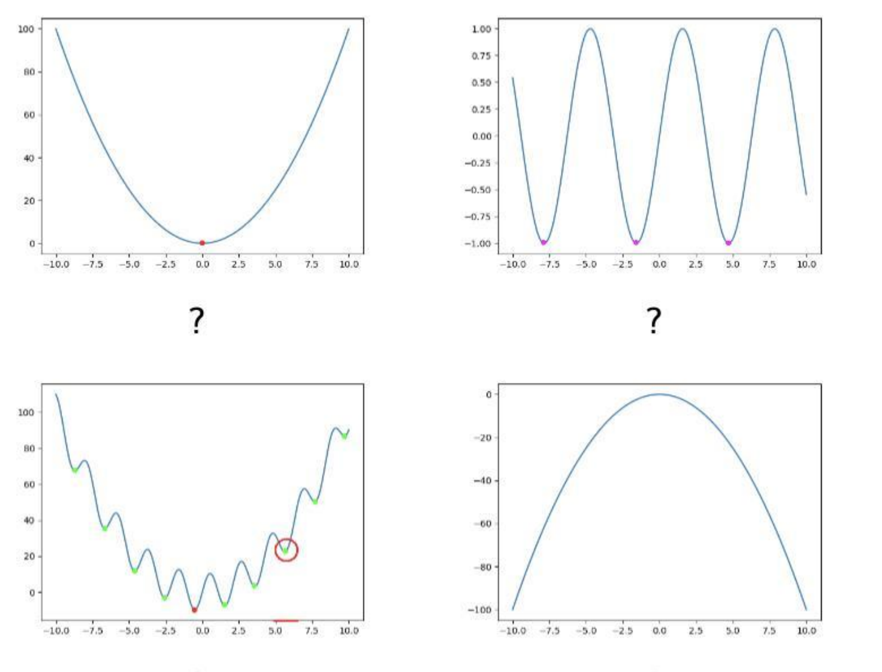
- $\Omega(x) \geq 0, \quad \forall x \in X_1$.
- The element x^* (i.e., the exact solution) belongs to the support of the functional Ω: $x^* \in X_1$
- $\forall d > 0$, the set $F_{1,d} = \{x \in F_1 : \Omega(x) \leq d\}$ is a compact.

The approximate regularized solution:

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

72) Write the definition of minimum and minimizer. Provide the classification of them and explain this classification with examples.

- $f(u) : H \to \mathbb{R}$ - the function to be minimized.
- $u \in U \subseteq H$.
- **Unconstrained programming:** $U = H$.
- **Constrained programming:** $U \subset H$.
- $u^* = \underset{u \in U}{\operatorname{argmin}} f(u)$ - 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.
- 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 $\exists r > 0 : f(u) > f(u^*) \quad \forall u \in U \cap \bar{S}_r(u^*)$



1) Strict Global minimizer
2) Strict local minimizer
3) Strict Global minimizer and strict local minimizer
4) ?

73) Write the definition of the differentiable function and the Frechet derivative.

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

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

- The operator f' is a **Strong derivative** (so-called the **Frechet derivative** or **The Gradient**).

74) Write the necessary and sufficient conditions for the functional f(u) to have a local minimum at the point u0. Give an example.

Necessary condition of extremum: Let the functional f to be differentiable at the point $u^* \in U$.
If u^* is a local minimizer, than $f'(u^*) = 0$.

Example:

f(x) = x^2, parabola with strict global minimizer x* = 0. f’(x) = 2x, f’(x*) = 0

75) What is a convex functional? Which kinds of convex functional do you know?

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:

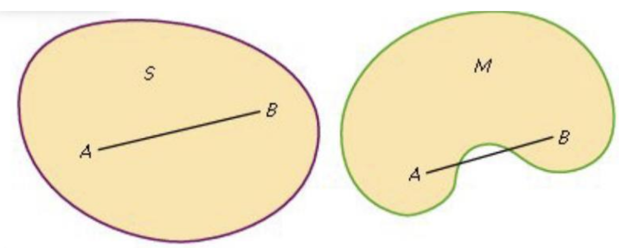
$$[u_1, u_2] = \{u \in U : u = u_1 + \lambda(u_2 - u_1), \lambda \in [0, 1]\}$$

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 $\bar{S}_r(u_0)$



©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 $\forall \lambda \in [0, 1]$:

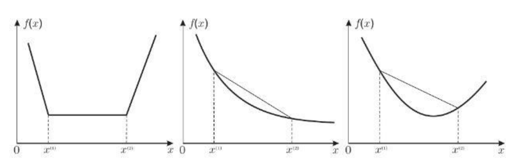
$$f(\lambda u_1 + (1 - \lambda)u_2) \leq \lambda f(u_1) + (1 - \lambda)f(u_2).$$

- 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

76) Prove that the functional f(x) =||x||^2 is strongly convex

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**

$$f(x) = \|x\|^2$$

Strong convex: if $\exists \theta > 0$:

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

$$f(\lambda x_1 + (1-\lambda)x_2) = \|\lambda x_1 + (1-\lambda)x_2\|^2 = \lambda^2\|x_1\|^2 + (1-\lambda)^2\|x_2\|^2 + 2\lambda(1-\lambda)\langle x_1, x_2\rangle$$

$$\lambda f(x_1) = \lambda\|x_1\|^2$$

$$(1-\lambda) f(x_2) = (1-\lambda)\|x_2\|^2$$

$$\lambda^2\|x_1\|^2 + (1-\lambda)^2\|x_2\|^2 + 2\lambda(1-\lambda)\langle x_1, x_2\rangle - \lambda\|x_1\|^2 - (1-\lambda)\|x_2\|^2 =$$

$$= \lambda\|x_1\|^2(\lambda - 1) + \|x_2\|^2(1-\lambda)(1-\lambda-1) + 2\lambda(1-\lambda)\langle x_1, x_2\rangle =$$

$$= \lambda\|x_1\|^2(\lambda - 1) + \lambda\|x_2\|^2(\lambda-1) - 2\lambda(-1+\lambda)\langle x_1, x_2\rangle =$$

$$= \lambda(\lambda-1)\cdot\{\|x_1\|^2 + \|x_2\|^2 - 2\langle x_1, x_2\rangle\} =$$

$$= \lambda(\lambda-1)\cdot\|x_1 - x_2\|^2$$

$\lambda \in [0; 1]$ $\quad -\lambda(1-\lambda)\|x_1 - x_2\|^2 \le -\underbrace{\frac{1}{2}}_{\theta}\lambda(1-\lambda)\|x_1 - x_2\|^2$

77) What is convex programming? What is the advantage of it?

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

- **Convex programming problem:** U is a convex set, and f is a convex functional.
  In Convex Programming, any local minimizer is a global minimizer!

78) Describe the Least Squares Method, construct the cost functional and discuss on its derivatives.

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:

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

79) What is the pseudosolution and normal pseudosolution?

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:**

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

80) Describe the main idea of the Golden Section Method.

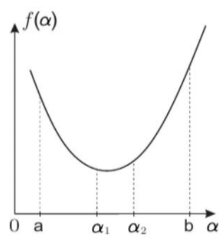
1D optimization: the Golden Section Method

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

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 $\varepsilon > 0$.

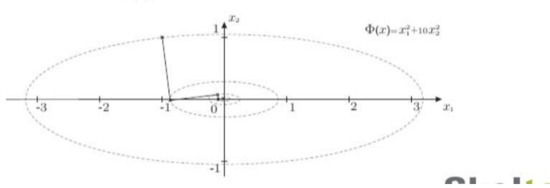


81) Describe the main idea of the Gradient Descent.

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 $g_0 = f'(x_0)$.
- $x_1 = x_0 - \alpha_0 g_0$, where α_0 is parameter (step) to chose.
- $g_n = f'(x_n)$; $\alpha_n = \underset{\alpha \geq 0}{\operatorname{argmin}} f(x_n + \alpha g_n)$; $x_{n+1} = x_n - \alpha_n g_n$



82) What is the k-step method? Which 2-steps methods do you know?

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:

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

83) Give the description of the heavy-ball method.

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.