ITMO

Level of education: **Master**
Field of study: **16.04.01 Technical Physics**

Advanced Quantum and Nanophotonic Systems

COMPUTATIONAL PHYSICS

Credits: 6 ECTS		
Semester		**Assessment**
2nd semester	6 ECTS	Exam

Course developers: **Dmitry R. Gulevich**

The research in modern physics relies heavily on numerical computations. Computers are used to calculate properties of molecules, atoms and materials, to simulate formation of the large structure of the Universe that we see today and model nuclear reactions in stellar interiors.
The Computational Physics is a constantly evolving field. Not that long time ago, a researcher had to write a programming code for routine operations such as calculation of matrix eigenvalues, interpolation and integration of scalar functions, optimization problems, finding roots and fitting functions, solving systems of linear equations and many others. Nowadays, most of these is done by calling a simple command in Python or highly optimized numerical libraries where all efforts to compete with those by writing its own code are in the majority of cases are doomed to failure. In this respect, the tools of the modern Computational Physics resemble a high-level programming language where many operations are reduced to the use of few commands. On the other hand, there are fields of Computational Physics which do not submit to this trend. One of these is partial differential equations. There are only few cases of partial differential equations where it is justified to use specialized numerical solvers, while many applications in physics require a custom-tailored approach by developing a performance-critical programming code. The MSc course in Computational Physics will provide the student those instruments and methods which, among the plethora of existing tools, may potentially be used in the current research in a Physics-related field. Upon completion of the course the student will be able to identify the situations where it is desirable to write his/her own custom-tailored programming code, and, to the opposite, the fields where competing with existing numerical libraries should be used. Also, the students will acquire the experience of working with Python numerical tools and libraries, will learn how to create custom-build Python modules and optimized libraries for performance-critical applications, will be able to propose solutions to realistic problems in modern Physics.

Requirements

The student should be familiar with the BSc level courses in Numerical Methods, Programming, Quantum Mechanics and Condensed Matter Physics read at the Faculty of Physics and Engineering or their equivalents.

Course structure

1. INTRODUCTION TO THE COMPUTATIONAL PHYSICS
   - 1.1. Programming with Python. Review of the existing Python interactive shells and online services.
   - 1.2. Postprocessing and visualization of scientific results. Plots and animation in Python. Python as an alternative to Gnuplot and Origin.
   - 1.3. Introduction to version control with git and GitHub. Acceleration and optimization of the Python code. Introduction to Numba.

2. MATHS BEHIND THE PYTHON TOOLS
   - 2.1. Solution to algebraic equations and linear system of equations in Python.
   - 2.2. Optimization problems. Fitting and interpolation. Numerical integration and quadrature formulas. Fourier analysis. Hankel transform. Special functions.

3. MATRIX COMPUTATIONS
   - 3.1. Matrix computations in Python. Dense and sparse matrices. Storage of sparse matrices in a computer. Linear algebra numerical libraries.

4. SOLUTION TO ORDINARY DIFFERENTIAL EQUATIONS
   - 4.1. Finite difference methods.
   - 4.2. Numerical schemes. Order and stability of a numerical scheme. Explicit and implicit schemes.

5. SOLUTION TO PARTIAL DIFFERENTIAL EQUATIONS
   - 5.1. Types of partial differential equations. Heat equation. Wave equation.
   - 5.2. Split step method. A review of finite element (FEM) solvers.
   - 5.3. Numerical stability.

6. COMPUTATIONAL QUANTUM CHEMISTRY
   - 6.1. Hydrogen atom. Schroedinger equation. Density functional theory and ab initio methods in Quantum Chemistry.

7. COMPUTATIONAL CONDENSED MATTER PHYSICS
   - 7.1. Solitons in the nonlinear Schroedinger equation (NSE).
   - 7.2. Exciton-polaritons in 2D quantum wells.
   - 7.3. 2D electron gas.
   - 7.4. Skyrmions in 2D magnetic materials.
   - 7.5. Sine-Gordon solitons in superconducting contacts.
   - 7.6. Edge modes in topological insulators.
   - 7.7. Density functional theory in Condensed Matter Physics.

8. COMPUTATIONAL ASTROPHYSICS
   - 8.1. N-body problem in Astrophysics. Simulation of spiral density waves of a spiral galaxy. Evaporation of a globular cluster.
   - 8.2. Numerical relativity. Gravitational waves.

9. THE MONTE CARLO METHOD
   - 9.1. Monte Carlo integration. Metropolis algorithm. Quantum Monte Carlo in Condensed Matter Physics.

10. PARALLEL COMPUTATIONS

10.1. Parallel computation on multiprocessor machines. Acceleration if computations using GPU.

Assessment

The final grade is calculated based on the student's progress in homework problem solving, oral examination grade, and score for in-class quizzes.

Faculty: **Faculty of Physics**

Contacts: **Alina Kozhevnikova, Academic Support Office manager**
Email address: a.kozhevnikova@metalab.ifmo.ru
Office location; phone: Lomonosova, 9, +7-900-630-16-43
Office hours: 10:00-17:00

Tags: **Numerical Computations in Physics, Scientific Python, Linear and Differential Equations, Matrix Computations, Parallel Computations**