Course Syllabus

Skoltech

Course Title	Scientific Computing
Course Title (in Russian)	Научные вычисления
Lead Instructor	Koshev, Nikolay
Co-Instructor	Koshev, Nikolay; Yarotsky, Dmitry

1. Annotation

Course Description

This is an introductory course in Scientific Computing, with a focus on the mathematical and algorithmic aspects of real-world computations in various areas such as physics, mathematics, medicine, economics, and more. The course comprises both theoretical and practical components.
The theoretical branch introduces the fundamental concepts of various types of mathematical problems that arise in different applications. It also covers approaches to classifying these problems, reformulating them when necessary, and providing solutions.
The practical aspects involve the application of various computational techniques for solving scientific and engineering tasks. These techniques will be taught through practical demonstrations, and efforts will be made to integrate them as much as possible with the corresponding theoretical materials presented during lectures.
All topics in the course will be covered at an advanced introductory level. The goal is that, upon completing the course, students will have acquired enough knowledge to begin using scientific computing and high-performance computing (HPC) methods in their everyday research work.
Students should have a comfortable understanding of undergraduate mathematics, particularly in the basics of calculus, linear algebra, and probability theory. Some preliminary knowledge of Unix-like operating systems is a plus. While the course will provide an overview of some popular pieces of commercial software used in HPC, all the software used for practical tasks in this course is open source and freely available.

Course Description (in Russian)

Данный курс является введением в научные вычисления с акцентом на математические и алгоритмические аспекты численного решения широкого круга задач, возникающих в приложениях физики, биологии, медицины, экономики и пр. Курс состоит из теоретической и прикладной компонент. Теоретическая часть курса представляет для изучения базовые понятия и методы, применяемые в различных областях математики, необходимых для решения реальных задач в науке и индустрии, моделирования данных, их предобработки и обработки. В ходе лекций мы научимся классифицировать задачи, переформулировать их в случае необходимости, находить оптимальные подходы к их решению.
Обучение практическим аспектам использования различных вычислительных техник для решения научных и инженерных задач будет сопровождаться практическими демонстрациями и будет максимально интегрировано с соответствующим теоретическим лекционным материалом.
Все темы курса будут освещаться на продвинутом вводном уровне, с целью снабдить студентов достаточным числом навыков, необходимых для начала самостоятельного использования научных вычислений и методов HPC в повседневной исследовательской работе.
От студентов ожидается знание стандартных математических дисциплин, изучаемых на первых

курсах технических ВУЗов, в частности анализа, линейной алгебры и теории вероятности. Также будет полезно предварительное знакомство с Unix-подобными операционными системами. Хотя в курсе будет дан обзор некоторого популярного применяемого для HPC коммерческого ПО, для практических занятий будет использоваться лишь свободное общедоступное ПО.

2. Basic Information

Course Academic Level	Master-level course suitable for PhD students
Number of ECTS credits	6

Course Prerequisites / Recommendations

- Basic undergraduate courses in ODE, PDE, classical mechanics, basic linear algebra, probability
- Familiarity with the Python programming language

Type of Assessment	Graded

Mapping from grades to percentage:

A:	80
B:	70
C:	60
D:	50
E:	40
F:	0

Term	Term 1B (last four weeks)

Students of Which Programs do You Recommend to Consider this Course as an Elective?

Masters Programs	PhD Programs
Data Science Mathematical and Theoretical Physics All Master Programs	Computational and Data Science and Engineering Mathematics and Mechanics Physics All PhD Programs

Maximum Number of Students

	Maximum Number of Students
Overall:	30
Per Group (for seminars and labs):	30

Course Stream	Science, Technology and Engineering (STE)

3. Course Content

Lecture, lab and seminar hour distribution among topics

Topic	Summary of Topic	Lectures (# of hours)	Seminars (# of hours)	Labs (# of hours)
Introduction to Scientific Computing and HPC	History of scientific computing and HPC, terminology of HPC, main areas of applications of HPC, current trends in HPC architectures, power consumption	3	2	2
Programming strategies for HPC	Basic concepts of parallelization, introduction to OpenMP and MPI, compromise between scalability and convergence rate; memory greedy vs CPU/GPU greedy methods	3	2	2
Classification of mathematical problems arising in Scientific Computing	Definition of the operator equation and classification with respect to known and unknown (aim) data: forward and inverse problems, static and dynamic, deterministic and stochastic, continuous and discrete, well- and ill-posed problems. Data-driven and model-based approaches.	3	2	2

Topic	Summary of Topic	Lectures (# of hours)	Seminars (# of hours)	Labs (# of hours)
Numerical approaches in Scientific Computing, approximation of differentials and integrals.	Meshing (structured and unstructured grids), approximations (finite differences and finite elements), sources of errors (solvers, errors of the model, statistical errors), some popular methods of SLAE research and solution, balance between accuracy and computational efficiency.	3	2	2
Numerical approaches on solving the differential equations	Definition and classification of the differential operators, ordinary differential equations, partial differential equations. Initial value problems, boundary value problems (Cauchy, Dirichlet, Neumann). Applications and examples.	3	2	2

Topic	Summary of Topic	Lectures (# of hours)	Seminars (# of hours)	Labs (# of hours)
Integral transforms, integral equations, and their numerical applications in HPC	Main applications of integral transforms (image/video analysis, signal processing and solving of integral equations), different flavors of the Digital Fourier Transform (FFT, NFFT, STFT etc) and its applications, including CNNs. Parallel FFT, wavelets: tailored basis-sets as well as good digital filters, Radon transform and its applications in 3D imaging. Definition and classification of integral equations.	1.5	1	1
Numerical approaches to solution of multidimensional PDE-driven problems.	Classification of partial differential equations, description of elliptic, parabolic and hyperbolic partial differential equations, the general pipeline to numerical solution of multidimensional PDEs, Galerkin methods, Finite Elements method, Finite Volumes method. Applications and examples.	1.5	1	1

Topic	Summary of Topic	Lectures (# of hours)	Seminars (# of hours)	Labs (# of hours)
Stochastic problems	Classification, examples and applications, shared points with other kinds of problems, forward and inverse stochastic problems, methods of solution: Monte-Carlo, ML etc.	1.5	1	1
Optimization	Definition and classification of mathematical optimization problems, local and global minima, convex and non-convex programming, pseudosolutions. Overview of numerical optimization techniques. Optimization in Neural Networks; machine learning for ill-posed and well-posed problems.	1.5	1	1

4. Learning Outcomes

Please choose framework for learning outcomes Knowledge-Skill-Experience

Knowledge
Mathematical and algorithmic foundations of Computational Science & Engineering as well as main aspects of High Performance Computing (HPC).
Statements of all major computational problems in science and engineering as well as main approaches/techniques to solve them.
State of the art techniques of scientific computing including parallel programming and HPC.
Main methodological aspects of both, scientific research and application development in Computational Science and Engineering.
How to apply different scientific computing techniques and algorithms (including HPC) to real-world problems in natural and social sciences and different industrial sectors.

Skill
Identify, formalize, and solve outstanding computational problems in real-world applications.
Understand and formulate new models of complex systems arising in natural and social sciences and engineering.
Choose the most appropriate method to solve a particular computational problem.
Apply relevant software tools, computer languages, data models, and computational environments for modelling and visualization.

Experience
Contribute in the development of the next-generation scientific computing software competitive with or superior than the existing examples of software in critical and emerging application fields (such as oil and gas, material design, big data, aerospace, and pharma).
Integrate different components of computational tools and HPC hardware to produce computational solutions for real-world tasks.
Work with technical literature (e.g. perform an effective bibliographical research, read and critically analyze scientific articles, use scientific metrics and most important databases).
Communicate results of analysis effectively (visually and verbally) to different audiences (specialists, users, stakeholders etc).

5. Assignments and Grading

Assignment Types

Assignment Type	Assignment Summary	% of Final Course Grade
Class participation	Activity of students during lectures and seminars	5
Homework Assignments	Theoretical and computational tasks	30
Final Exam	Contains 10 theoretical questions and 14 practical tasks	35
Final Project	The research project with application of the course material	30

6. Assessment Criteria

Select Assignment 1 Type

Class participation

Input or Upload Sample of Assigment 1:

Input Sample of Assignment 1

Under assumptions of Amdahl's law, given a program in which exactly 60% of computations can be ideally parallelized, compute the efficiency of parallelizing the program to 3 processes.

Assessment Criteria for Assignment 1

Student activity and correctness of solutions

Select Assignment 2 Type Homework Assignments

Input or Upload Sample of Assigment 2:

Input Sample of Assignment 2

Under assumptions of Amdahl's law, given a program in which exactly 60% of computations can be ideally parallelized, compute the efficiency of parallelizing the program to 3 processes.

Assessment Criteria for Assignment 2

Each home assignment will include several problems, theoretical as well as computational (requiring students to perform simulations). In the case of computational tasks the solution should be accompanied by the working programming code.

Select Assignment 3 Type Final Exam

Input or Upload Sample of Assigment 3:

Input Sample of Assignment 3

Under assumptions of Amdahl's law, given a program in which exactly 60% of computations can be ideally parallelized, compute the efficiency of parallelizing the program to 3 processes.

Assessment Criteria for Assignment 3

Correctness and completeness of solutions and answers

Select Assignment 4 Type Final Project

Input or Upload Sample of Assigment 4:

Input Sample of Assignment 4

For the final project, the student must perform a research on the chosen topic and make a short presentation in class.

Typical structure of the presentation:
a. statement of the problem;
b. description of the current situation in this field;
c. problem solution;
d. numerical simulations;
e. conclusion.

Assessment Criteria for Assignment 4

Baseline grade (8/10) requirements:
- Provide background material (motivation, theory, existing methods of solution)
- Demonstrate your numerical implementation
- Analyze results (validate your numerical solution, compare with theory and other existing methods of solution)
- Answer questions

Higher grades (9,10) are usually given for some of the following:
- An original, interesting project topic (understandable by a non-expert)
- An original approach/solution
- An extensive analysis of the topic / substantial amount of work / implementation involving advanced technologies
- An outstanding presentation

Input or Upload Sample of Assigment 5:

Input or Upload Sample of Assigment 6:

Input or Upload Sample of Assigment 7:

Input or Upload Sample of Assigment 8:

Input or Upload Sample of Assigment 9:

In the next question we ask you to define general categories of the course. What does your course teaches in broad terms?

7. Textbooks and Internet Resources

You can request at most two required textbooks. Additionaly, you can suggest up to nine recommended textbooks.

Required Textbooks	ISBN-13 (or ISBN-10)
-	

Recommended Textbooks	ISBN-13 (or ISBN-10)
Computation and its limits, by Paul Cockshott, Lewis M. Mackenzie and Greg Michaelson, Oxford University Press, 2015.	9780198729129
Introduction to High Performance Computing for Scientists and Engineers, Georg Hager, Gerhard Wellein, CRC Press, Taylor & Francis Group, 2011.	9781439811924
B. Polyak, Introduction to optimization	0-911575-14-6

Web-resources (links)	Description
http://www.ee.surrey.ac.uk/Teaching/Unix/	Linux/Unix tutorial
http://www.gromacs.org/Documentation	Gromacs Manual
http://www.openfoam.com/documentation/user-guide/index.php	OpenFoam User Guide
http://www.john-von-neumann-institut.de/nic/EN/Publications/NICSeries/_node.html	IC Lecture series
http://www.cs.nyu.edu/courses/spring09/G22.2112-001/book/book.pdf	Principles of Scientific Computing, by David Bindel and Jonathan Goodman
http://pages.tacc.utexas.edu/~eijkhout/istc/html/index.html	Introduction to High-Performance Scientific Computing, by Victor Eijkhout
http://jupyter.org/	Jupyter notebooks
https://colab.research.google.com	Google Colab

Web-resources (links)	Description
https://mpi4py.readthedocs.io/en/stable/tutorial.html	An mpi4py tutorial
https://web.stanford.edu/class/cs97si/suffix-array.pdf	Suffix arrays

8. Facilities

Software
IPhython, Numpy, Gromacs, OpenFoam (all open source)

Equipment
Audience (till 30 persons): 421, 408.
Access to the Internet through a computer class and Wi-Fi network of the institute

9. Additional Notes