



Syllabus

2022-2023



TRAINING THE EXCELLENCE

The Master in High Performance Computing (MHPC) is hosted and organized by **SISSA** (International School for Advanced Studies) and **ICTP** (Abdus Salam International Center for Theoretical Physics) in Trieste.

MHPC is an **innovative specialization program** devoted to training students in the booming field of HPC.

Students that complete the Master have a solid background in **advanced and parallel computing approaches, algorithms, and machine learning techniques**.

The program combines lectures with hands-on tutorials, that are held by **internationally renowned scientists**.



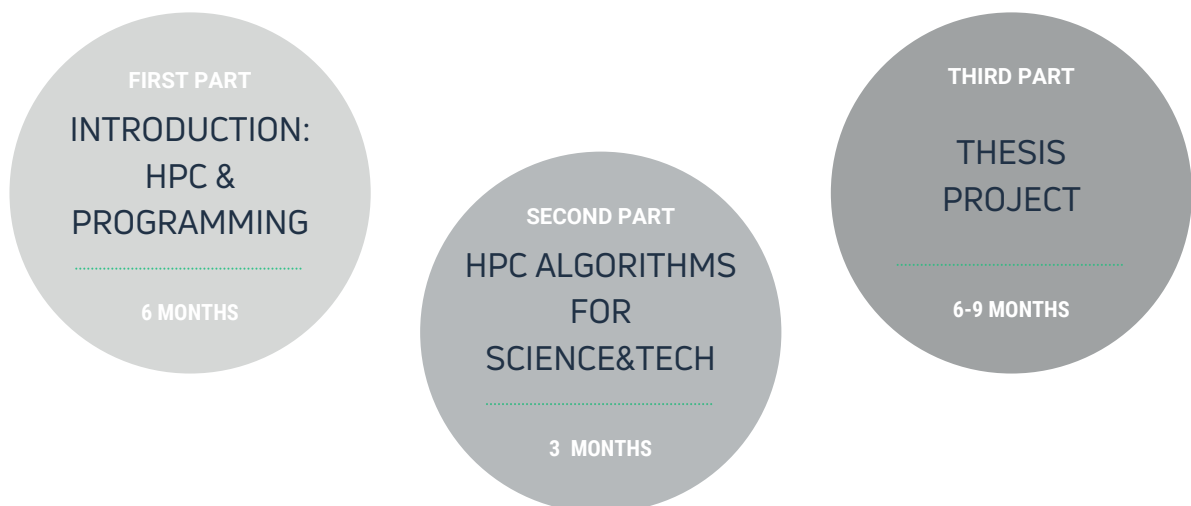
The Abdus Salam
International Centre
for Theoretical Physics





THE PROGRAM

During the **12-months-program**, students follow courses by international **lecturers from academia and from the industrial world**, who teach them the latest applications of HPC technologies and innovations.



Courses should include **practical and exercise-based sessions** and the majority of the first part ones spans several months.

Some courses of the second part could be optional, but a minimum of CFUs is needed to complete the program.



from September to June

COURSES

Part I - from September to February

- 1.1 Scientific Programming Environment*
- 1.2 Introduction to Computer Architectures for HPC*
- 1.3 Advanced Programming*
- 1.4 Introduction to Parallel Programming*
- 1.5 Introduction to PETSc library*
- 1.6 High Performance Computing Technology*
- 1.7 Introduction to Numerical Analysis*
- 1.8 The Finite Element Method Using deal.II*
- 1.9. Advanced Linear Algebra Libraries and Accelerators*
- 1.10 Unsupervised Machine Learning*

Part II - from February to June

- 2.1 Deep Learning
- 2.2 Advanced Deep: The Transformer Architecture Learning*
- 2.3 Molecular Dynamics
- 2.4 Electronic structure: from blackboard to source code
- 2.5 Reinforcement Learning
- 2.6 Data Structures & Sorting and Searching
- 2.7 Best Practices in Scientific Computing
- 2.8 Fast Fourier Transforms in Parallel and Multiple Dimensions
- 2.9 Approximation and interpolation of simple and complex functions
- 2.10 Reduced Order Modeling

* mandatory



from September to February

FIRST PART

Scientific Programming Environment

Irina Davydenkova

The course introduces Unix-like operating systems, and shows how to set up a scientific programming environment in such systems. The topics include bash scripting, introduction to best practices in code maintenance (version control, documentation), debugging techniques, build systems and basic rules of cluster usage. We also review the C language and use it to practice the topics mentioned above.

Introduction to Computer Architectures for HPC

Luca Tornatore

In this course we tackle three different key factors in High Performance Computing. As first, building on main elements of modern computer and CPU architecture, most important techniques for code optimization on single-core are discussed. As second, the fundamental traits of parallel computation are introduced and defined. Third, we establish the basis of parallel programming in both the shared-memory and the distributed-memory paradigm using OpenMP and MPI respectively.

Advanced Programming

Irina Davydenkova

This course mainly focuses on modern C++ and is designed to teach the latest features and best practices of the language up to C++17. The goal of the course is to present the topics relevant to writing scientific simulation codes using the advanced features of the language and to see the applications of the techniques learned during the introductory course.



from September to February

FIRST PART

Introduction to Parallel Programming

Ivan Girotto

Introduction to key topics in parallel computing. Main parallel programming paradigms: message passing (MPI) and multi-threading (OpenMP).

Introduction to PETSc library

Stefano Zampini

This course introduces the Portable and Extensible Toolkit for Scientific Computations (PETSc) as a tool for the design and implementation of high-performant, distributed memory software for the solution, of nonlinear equations and eigenvalue problems. It covers topics ranging from concepts about library design to advanced solver capabilities, alternating class lectures and hands-on sessions. The exploitation of Graphical Processing Units (GPUs) computing capabilities will also be discussed.

Upon completing this course, students will be able to write scalable, performance portable software projects and design robust solvers strategies using PETSc.

High-Performance Computing Technologies

Fernando Posada

This module introduces state-of-the-art technologies and innovation in High Performance Computing. Main components of computing infrastructure are analyzed and discussed. Students will install and configure a HPC Linux Cluster and will also be exposed to the use of Cloud and Grid Infrastructures.



from September to February

FIRST PART

Introduction to Numerical Analysis

Luca Heltai

Introduction to numerical analysis, with focus on linear algebra, polynomial approximation, numerical integration and numerical solution of ODEs.

The Finite Element Method Using deal.II

Luca Heltai

The courses introduces deal.II, a modern C++ library for solving differential equations with the Finite Element Method.

Advanced Linear Algebra Libraries and Accelerators

Piotr Luszczek

Large supercomputing installations and scientific clusters will be discussed with the emphasis on their architectural features that are essential for good performance and scalability. Although majority of the presented code will be explained, familiarity with programming is welcome and will be helpful in following along.

Unsupervised Machine Learning

Alex Rodriguez

During this module, we will follow the whole procedure of exploratory data analysis with this technique, starting from the raw data and finishing in the validation of the results. This path will lead us to introduce some concepts of other unsupervised machine learning techniques like dimensional reduction. The duration of the module is for three days, with both theoretical and practical lessons.



from February to June

SECOND PART

Deep Learning

Alessio Ansuini, Alberto Cazzaniga

This course covers the basics of deep learning. Starting from the simple artificial neuron (perceptron), we will introduce artificial neural networks and their most common architectures, such as fully connected and convolutional models. Then, we will see how these networks learn to accomplish specific tasks from data. Back propagation and optimization of the loss function will be explained together with the standard pipeline to train, validate, and test these models. We will also introduce some basic concepts of unsupervised deep learning.

Advanced Deep: The Transformer Architecture Learning

Cristiano De Nobili

During the course, we will see advanced DL algorithms, such as the Transformer. We will code it from scratch in PyTorch. Then we will learn how to leverage more high-level API to train and fine-tune transformer-based models to solve specific tasks in NLP and not only. We will also taste some Interpretability/Explainability methods and apply them to the tasks we previously solved.

Molecular Dynamics

Giovanni Bussi

Analyzing the physical movements of atoms and molecules through computer simulation.



from February to June

SECOND PART

Electronic structure: from blackboard to source code

Stefano De Gironcoli

Material science and condensed matter theorists extensively employ in their research ab initio atomistic simulations as implemented in a number of widely available software codes. Most often these tools are used as 'black boxes' with little or only partial knowledge of the practical implementation of the general theoretical ideas they are based on. This is a particularly severe drawback when new theoretical developments, analysis tools or computational experiments can not be pursued due to the lack of insight on the internal structure of the employed research software.

The aim of this course is to address the above mentioned issue providing the students with a detailed knowledge of the internal design of state of the art electronic structure codes, filling the gap that exists between the knowledge of the general principles underlying modern atomistic simulations and their practical implementation in actual codes. Tools and codes available in the Open Source Quantum ESPRESSO software distribution will be used as working examples.

Reinforcement Learning

Emanuele Panizon

An introduction to the basic concepts and algorithms that stand at the foundations of Reinforcement Learning.



from February to June

SECOND PART

Data Structures & Sorting and Searching

Axel Kohlmeyer

Introduction to fundamental data structures and their impact on performance and memory consumption. Study parallelization issues.

Best Practices in Scientific Computing

Axel Kohlmeyer

A module where students are introduced to best practices in scientific computing from different perspective: software development with modern software engineering techniques, optimal exploitation of different HPC platforms, usage and maintenance of large scientific software packages.

Fast Fourier Transforms in Parallel and Multiple Dimensions

Ivan Giroto - Ralph Gebauer

Introduction to the Discrete Fourier Transform (DFT) and its application to real problems. From the Discrete to the "Fast" version (FFT). Analysis of a most common algorithm for the solution of a multi-dimensional FFT on parallel systems.

Approximation and interpolation of simple and complex functions

Nicola Seriani

Introduction to several techniques for efficient approximation of numerical functions to varying degrees of accuracy.



Reduced Order Modeling

Gianluigi Rozza

In this course we present reduced basis (RB) approximation and associated a posteriori error estimation for rapid and reliable solution of parametrized partial differential equations (PDEs).

OTHER COURSES FROM THE PAST EDITIONS:

- Big Data Processing with MapReduce
- Lattice Boltzmann
- Object Oriented Programming
- HPC Applications in Science and Technology
- Lookup Tables, Cell and Neighbor Lists
- Domain Decomposition Methods
- Advanced Optimization Techniques
- Build system tools and unit-testing
- Deep Generative Models with TensorFlow 2
- Foundation of HPC
- Spatial Locality Algorithms
- Supervised Machine Learning
- Unsupervised learning: from the foundations to advanced deep models

CONTACT US

SISSA

Via Bonomea, 265
34136 Trieste
ITALY

mhpc.it
info@mhpc.it

ICTP

Strada Costiera, 11
34151 Trieste
ITALY

