Master in High Performance Computing

Student Handbook

2016-2017

info@mhpc.it

Joint Sissa/ICTP Master in High Performance Computing

Overview of the program

Industry and academia throughout the world are suffering from a severe shortage of skilled personnel able to bridge the gap between the dynamic environments in IT and research/science. The MHPC aims to create a group of experts who have the hands-on training and skills needed to fill this gap, by providing students with a solid background in scientific computing approaches, algorithms, and modeling. MHPC degree provides the access to instant career opportunities on an international scale.

- Industry Involvement

  The program features a strict connection with industry, both local and worldwide. The MHPC aims to fulfill the demand for professional experience in a production environment. HPC vendors, IDVs and local companies are involved in the financing of fellowships and in providing industry-based HPC problems for the final thesis. Some of the Master's lectures come from the industrial world to teach the latest applications of HPC technologies. This is also an excellent opportunity for students to get in touch with companies involved in HPC sectors and to explore potential career paths.

- Career Prospects

  MHPC provides skills useful both in the academic and in the industrial field, like technical and software development. The master fulfills the need of highly trained professionals in the HPC field. The strong connection with industry and highly reputed academic environment provide to an MHPC graduate a unique chance to to find the right place in the high technology market or academic scene.

The program is composed of two periods of courses followed by a six month research project. The first section provides the fundamentals of parallel computing and HPC, while the second period is focused on advanced computational methodology in science and technology. Students have access to high-profile HPC resources during the program.

General organization of the courses

Structure of the program

- The program is divided in three parts.
    - Part 1 provides the foundation of programming, parallel programming and high performance computing
    - Part 2 provides advanced techniques and methodology that are used in nowadays science and technology
    - Part 3 is a large-scale, high profile research project. (See “Thesis” section for more details)
- Each course will, with a few exceptions, span from 2 days to 2 weeks.
- The courses are extremely practical and exercise-based. To successfully complete a course, the student will have to hand in the solutions to all the exercises by a given deadline.
- External students are allowed to follow three courses maximum.

Contacts

- For everything related to the courses and the exercises please refer to the assistant coordinators, Nicola Cavallini and Alberto Sartori, nicola.cavallini@sissa.it alberto.sartori@sissa.it
- For the logistics in the Miramare campus please refer to Sabrina Visintin administration@mhpc.it
- For the logistics and secretariat at SISSA, please refer to the ILAS secretariat (ilas@sissa.it)

Facilities

- The official headquarter the master is at first floor of the so called “Old Sissa building” at the Miramare Campus
- Each student will share an office with other participants. Students offices are Room 101 110 109 108.
- Room 107 hosts the Master coordinator and external teacher/lecturers
- Room 106 hosts the assistant coordinators, Nicola Cavallini and Alberto Sartori.
- Lecture and HPC lab will be in ROOM E in the basement of the same building.
- Occasionally some lectures can be held in ROOM D, ROOM A or at the Sissa Main building
- For the logistic in the Miramare campus please refer to Sabrina Visintin administration@mhpc.it

Resources

- The student will have access to leading-edge HPC resources provided by Sissa and ICTP
- The student will have both a Sissa and an ICTP, with all the benefits those accounts bring
- The student is expected to use her own laptop for the exercise classes, through which will connect to the HPC resources. If the student does not have a laptop, he/she should communicate it to the program coordinator, Dr. S. Cozzini.

Evaluation

- The student is expected to successfully pass all the courses to obtain the title.
- Each course has a set of small assignments, to be completed in the afternoon of each lecture day, under the guidance of the lecturer and of the teaching assistants. This assignments will not be graded.
- The mark of each single course will be based on a longer assignment or exam that should encompass the relevant part of the course. This is the only graded part and will provide the final mark for the course.
- Individual courses will give a mark (from 18 to 30 as integer) only if passed, otherwise the mark will be FAIL.
- The minimum mark to pass a course is 18 out of 30
    - Mark 18 corresponds to a sufficient fulfillment of the assignment
    - Mark 27 corresponds to successfully complete the whole assignment
    - Mark 30 corresponds to successfully complete the whole assignment and to provide an original or outstanding solution.
- Every failed exam of the first part should be amended by February for the first part and by June for the second part.
- In case a student fails 3 courses in one of the teaching parts, the scientific council is entitled to reconsider his enrollment in the master and exclude him.
- The student excluded from the master is considered as an external student, and can therefore attend a maximum of three courses in the next section.
- The student is accepted to the second part of the master program if the average mark of the first part is at least 21/30.

- Every failed exam of the first part should be amended by February for the first part and by June for the second part.
- In case a student fails 3 courses in one the teaching parts, the scientific council is entitled to reconsider his enrollment in the master and exclude him.
- The student excluded from the master is considered as an external student, and can therefore attend a maximum of three courses in the next section.
- The student is accepted to the second part of the master program if the average mark of the first part is at least 21/30.

Other important notes:

- Plagiarism will not be tolerated. Any discovered plagiarism will be punished with the exclusion from the master program
- At the end of each course of the first part, the student will be able to fill out an anonymous evaluation questionnaire regarding the course
- At the end of the second part of the master, the student will be able to fill out an evaluation questionnaire regarding the courses taken.

Minimum attendance and absence

- Student should attend 70% of the lectures of each course and provide the solutions to all the proposed exercise to be able to enter the evaluation process
- Absence should be notified by mail to info@mhpc.it
- Absence that will go above the 30% of a specific course time, will have to be discussed before with the program coordinator Ralph Gebauer.

Thesis regulation

1. To obtain the MHPC diploma, the student is expected to pass a final exam, where he/she will present the thesis work to an exam committee proposed by the Scientific Council and nominated by the Director of the master.
2. The thesis project should be approved by the Thesis committee within two weeks from the end of the courses.
3. The thesis project will be carried out under the guidance of one or more supervisors, which will verify the progress.
4. One month before the final exam session the student should hand-in the registration form for the thesis. The form needs to filled out by the student and signed by both the student and the supervisor(s), and needs to be sent electronically to ilas@sissa.it
5. To be admitted to the final exam the student must have
    a. reached the minimum mark for each course
    b. discussed with the Thesis Committee the thesis topic in 2 meetings that will take place in May and October. The May meeting should verify the validity and objectives of the thesis and the feasibility in the given time. The October meeting should verify the progress of the work and determine whether the student can take the final exams in in December, or in February
    c. should have submitted the thesis (signed and approved by the supervisor) to the thesis committee at least two weeks before the exam date.
6. For each thesis project, the coordinator of the master will nominate a referee who will examine the thesis and write a report. The supervisor(s) should also write a report, and both the reports have to be sent to ilas@sissa.it before the final exam.
7. To pass the final exam the student should obtain a mark of at least 21/30. The mark is given on basis of the marks of courses of both parts and on

thesis and its defense. For each academic year, there will be 2 sessions for the final exam, one in December and one in February. The February session is the last of each academic year.

8. If the supervisor or the Thesis Committee expresses a negative opinion on the thesis work, the student is not admitted to the final exam in that session. but in the following one. If the student is not admitted to the second session, he/she will not obtain the master diploma.
9. To determine the final mark, the marks of the courses of the first and second part of master will count up to 50%, while the thesis work and its defense will count up to 50%.
10. A final mark of 27/30 or more will grant the student a grade of "Completed with merit"
11. The thesis committee is nominated by the "Consiglio del Laboratorio Interdisciplinare" within March of each academic year, upon indication of the MHPC scientific council, and approved by the proper authorities.

Best MHPC thesis

On each academic year, a prize will be awarded to the best MHPC thesis. The prize will be composed by a special diploma and the covering of expenses for an HPC international conference (Super Computing or International SuperComputing), including travel expenses, accommodation, and local expenses, up to 1500 euros.

Legal definition of the title

- The title obtained at the end of the master is not equivalent to Master of Science (M.S.) or Laurea di secondo livello. It should be considered a specialization course ("Master universitario di I livello" in the italian system) and corresponds to level 7 in the *European Qualifications Framework*
- The recognition of the title by another institution is not automatic and it is subject to the decision of the evaluating institution.

People

MHPC Scientific Council

- Ralph Gebauer (ICTP)
- Stefano Cozzini (CNR/IOM)
- Luca Heltai (SISSA)
- Alessandro Laio (SISSA)
- Antonio Lanza (SISSA)
- Riccardo Valdarnini (SISSA)
- Ulrich Singe (ICTP)

MHPC Staff

- Ralph Gebauer (ICTP) - Program Coordinator - rgebauer@ictp.it
- Luca Heltai (SISSA) – MHPC Director – heltai@sissa.it
- Stefano Cozzini (CNR/IOM) - Vice Program Coordinator - cozzini@mhpc.it
- Nicola Cavallini (SISSA) - Assistant Program Coordinator - nicola.cavallini@sissa.it
- Alberto Sartori (SISSA) - Assistant Program Coordinator - alberto.sartori@sissa.it
- Mila Bottegal (SISSA) - Administration/Secretariat – ilas@sissa.it
- Claudia Parma (SISSA) - Administration/Secretariat – ilas@sissa.it
- Sabrina Visintin (ICTP) - Administration/Secretariat – administration@mhpc.it

Lecturers for Courses of the first part

- Stefano Cozzini (CNR/IOM)
- Nicola Cavallini (SISSA)
- Alberto Sartori (SISSA)
- David Grellscheid (ICTP)
- Ivan Girotto (ICTP)
- Fabio Affinito (CINECA)
- Filippo Mantovani (Barcelona Supercomputing Center)
- Andrea Bartolini (UniBO and ETH-Zurich)
- Axel Kohlmeyer (ICTP)
- Luca Heltai (SISSA)
- Gianluigi Rozza (SISSA)

- Moreno Baricevic (CNR/IOM)
- Massimo Bernaschi (CNR/IAC Rome)
- Chris Dahnken (INTEL)

Lecturers for Courses of the second part (to be confirmed)

- Axel Kohlmeyer (ICTP)
- Francesco Sanfilippo (University of Southampton)
- Riccardo Valdarnini (SISSA)
- Gianluigi Rozza (SISSA)
- Luca Heltai (SISSA)
- Olindo Zanotti (UNITN)
- Ralph Gebauer (ICTP)
- Ivan Girotto (ICTP)
- Giovanni Bussi (SISSA)
- Sandro Bottaro (SISSA)
- Stefano Baroni (SISSA)
- Sebastiano Pilati (ICTP)
- Nicola Seriani (ICTP)
- Alex Rodriguez (SISSA)
- Alessandro Laio (SISSA)
- Riccardo Zanella (UNIFE)

List of Courses

Part I

- 1.1 Scientific Programming Environment
- 1.2 Introduction to Computer Architectures for HPC
- 1.3 Parallel Programming
- 1.4 Introduction to Numerical Analysis
- 1.5 Object-Oriented Programming
- 1.6 Parallel Data Management and Data Exchange
- 1.7 Advanced Computer Architectures and Optimizations
- 1.8 High Performance Computing Technology
- 1.9 Best Practices in Scientific Computing

Part II

- 2.1 Data Structures and Sorting & Searching
- 2.2 Parallel Linear Algebra
- 2.3 Spatial locality algorithms
- 2.4 Reduced Basis Method
- 2.5 The Finite Element Method Using deal.II
- 2.6 Finite Volumes and Discontinuous Galerkin methods
- 2.7 Fast Fourier Transforms in Parallel and Multiple Dimensions
- 2.8 Molecular Dynamics
- 2.9 Monte Carlo methods
- 2.10 Approximation and interpolation of simple and complex functions
- 2.11 Cluster Analysis
- 2.12 Constrained Optimization

Part I Courses

Module 1.1:	**Scientific Programming Environment**
Coordinator	Nicola Cavallini (SISSA) Alberto Sartori (SISSA)
Module Description	This course will introduce Unix-like operating systems, show how to setup the scientific programming environment in such operating systems. It will present the modern software tools required to provide such an environment and discuss important points like documentation and testing.
Main Topics	● Introduction to Unix-like operating systems (kernel vs. userspace, processes/threads, file system semantics) ● Shell scripting (bourne shell) ● Basics of python programming ● Review of basic concepts of C and Fortran programming language, and mixing thereof ● Compiler architecture (preprocessor, compiler, assembler, linker) and use of libraries ● Build automation tools (make and CMake) ● Source code management (git) ● Unit and regression testing ● Software documentation
Objectives	On successful completion of this module students should have their own software environment and tools prepared and configured for the rest of the activities of the Master Program. Students learn the workflows of software development and working collaboratively. These concepts are the basis on which to develop

	efficient scientific code for high-performance computing applications.

Module 1.2:	**Introduction to Computer Architectures for HPC**
Coordinator	Stefano Cozzini (CNR-IOM) Nicola Cavallini (SISSA) Alberto Sartori (SISSA)
Module Description	Introduction to key topics in computer architecture needed in HPC environment including a detailed overview on parallel architectures.
Main Topics	● Computer architecture ● Memory hierarchy ● Modern multi-core CPU systems ● Overview on massively parallel processors ● Parallel architectures for HPC and future trends ● Benchmarking and energy efficiency in HPC ● Profiling of a serial applications ● False sharing and memory efficiency on Multi-core systems ● Practical introduction to software optimization
Objectives	On successful completion of this module students should be able to understand basic concepts of modern computer architecture, and know the different kinds of parallel architectures commonly applied to scientific research.

Module 1.3:	**Parallel Programming**
Coordinator	Ivan Girotto (ICTP) Nicola Cavallini (SISSA) Alberto Sartori (SISSA)
Module Description	Introduction to key topics in parallel programming. Main parallel programming paradigms: message passing (MPI) and multi-threading (OpenMP).
Main Topics	● Introduction to parallel computing ● Principles of parallel algorithm design and multi-level parallelism ● Message-passing parallelization (MPI) ● Shared memory parallelization (OpenMP) ● Analysis of scalability and parallel performance metrics ● Overview on the debugging and the profiling of parallel applications ● Practical introduction to parallel Math libraries ● Further trends of parallel programming in HPC
Objectives	On successful completion of this module students should be able to write parallel programs and know fundamental techniques of parallel programming to develop parallel applications along with methods of parallel performance analysis and debugging.

Module 1.4:	**Introduction to Numerical Analysis**
Coordinator	Luca Heltai (SISSA) Gianluigi Rozza (SISSA) Nicola Cavallini (SISSA) Alberto Sartori (SISSA)
Module Description	Introduction to numerical analysis, with focus on linear algebra, polynomial approximation, numerical integration and numerical solution of ODEs
Main Topics	● The foundations of numerical analysis ● Polynomial Interpolation ● Numerical Integration ● Resolution of non-linear systems ● Resolution of large linear systems ● Eigenvalues approximation ● Numerical solution of ODEs ● Numerical solution of PDEs
Objectives	On successful completion of this module students should be able to understand and implement in computer program numerical integration, numerical derivation, functional interpolation and basic linear algebra operations.

Module 1.5:	**Object Oriented Programming**
Coordinator	David Grellscheid (University of Durham) Nicola Cavallini (SISSA) Alberto Sartori (SISSA)
Module Description	Introduction to fundamental concepts of programming from an object-oriented perspective.
Main Topics	• Classes and objects • Abstraction • Encapsulation • Calling methods and passing parameters • Exceptions • Inheritance • Polymorphic variables and methods • OOP in C++ • C++ Standard Template Library • Design issues • OOP in Python
Objectives	On successful completion of this module students should be able to understand modern software engineering and design principles and should develop fundamental programming skills in the context of a language that supports the object-oriented paradigm.

Module 1.6:	**Parallel Data Management & Data Exchange**
Coordinator	Stefano Cozzini (CNR) Luca Bortolussi (University of Trieste)
Module Description	The module introduces modern techniques to deal with the large amount of data in scientific and technical computing.
Main Topics	● Introduction to Big data issues ● Parallel file systems and parallel I/O ● Scientific data formats and libraries (NetCDF, HDF5) ● MPI-IO ● Data intensive computing (distributed file systems and MapReduce), Hadoop ● Web interface and protocols for data exchange (i.e. opendap) ● Workflows for data processing ● Benchmarking and profiling data intensive calculation
Objectives	On successful completion of this module students have an overview of the main techniques and tools to tackle data-intensive computational problems.

Module 1.7:	**Advanced Computer Architectures & Optimizations**
Coordinator	Stefano Cozzini (CNR) Massimo Bernaschi (CNR/IAC Rome) Chris Dahnken (INTEL)
Module Description	The course presents advanced topics in optimization techniques needed in HPC environment. In particular it will focus on the use of application accelerators in high-performance and scientific computing and issues that surround it.

Main Topics	● Advanced optimization techniques
	● Memory management and optimization
	● Introduction to novel accelerator processors, systems, and architectures
	● Introduction to GPU computing
	● Overview of accelerated architecture
	● Programming interfaces for accelerator
	    ○ CUDA
	    ○ OpenCL
	    ○ OpenACC
	● Specific libraries with accelerator support

Objectives	On successful completion of this module, students will have an overview of the advanced computational architectures and accelerators and how to use them.

Module 1.8:	**High Performance Computing Technology**
Coordinator	Moreno Baricevic (CNR-IOM) Stefano Cozzini (CNR-IOM)
Module Description	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.
Main Topics	● HPC system deployment ● Software Provisioning (modules) ● Managing hardware diversity ● Scheduling and resource management ● Usage accounting ● Data management (quotas, purging, archival) ● Sustainable HPC computing infrastructure ● Green computing ● Grid and Cloud Computing
Objectives	On successful completion of this module students should be able to understand common problems related to the installation and maintenance of a sustainable HPC infrastructure.

Module 1.9:	Best Practice in Scientific Computing
Coordinator	Axel Kohlmeyer (Temple U. & ICTP)
Module Description	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.
Main Topics	● Debug versus optimized mode ● Multilanguage programming ● Strategies for developing scientific codes ● Collaborative ways of developing scientific and technical packages. ● Tools for developing large software packages
Objectives	On successful completion of this module students should have a clear idea about the most successful practices that can be adopted in developing technical and scientific software and make them run efficiently on modern HPC platform.

Part II Courses

Module 2.1:	**Data Structures and Sorting & Searching**
Lecturer	Axel Kohlmeyer (Temple U. & ICTP)
Class Duration	10 hours
Laboratory Duration	24 hours
Module Description	Introduction to fundamental data structures and their impact on performance and memory consumption. Study parallelization issues.
Main Topics	● Implementation of data structures like arrays, linked lists, hash tables, trees, maps in different programming languages ● Comparison of performance for data access, data insert, data removal, data append, scaling with problem size, and need for auxiliary storage ● Accessing and updating data structures in parallel (using locks and lock-free) ● Data structures for numerical problems (example: sparse matrix storage) ● Some Considerations on Data Structure Design ● Implementation of selected popular sort algorithms ● Comparison of performance for unsorted and presorted data ● Searching in unsorted or presorted data
Objectives	On successful completion of this module students should be able to determine which data structure is best suited for a given task.

Module 2.2:	**Parallel Linear Algebra**
Lecturer	Francesco Sanfilippo (Southampton U.)
Class Duration	16 hours
Laboratory Duration	8 hours
Module Description	Introduction to the solution of large linear systems in parallel
Main Topics	● Overview of real-world occurrences of the problem ● Comparison of direct and iterative solvers ● Preconditioning of a problem ● Accuracy of the solution, efficiency of different solvers ● Extensive discussion on implementation of a solver in massive parallel architectures
Objectives	Upon completion students will be able to implement a parallel conjugate gradient with different level of parallelism

Module 2.3:	Spatial locality algorithms
Lecturer	Riccardo Valdarnini (SISSA)
Class Duration	4 hours
Laboratory Duration	4 hours
Module Description	Theory and applications of algorithms for spatial locality
Main Topics	• Space filling curves-theory • Morton and Peano-Hilbert orders • Tree codes, general framework • Quad tree, R- tree, Kd-tree • Nearest neighbor search using space filling curves.
Objectives	Upon completion students will be able to deal with spatial locality algorithms.

Module 2.4:	Reduced Basis Method
Lecturer	Gianluigi Rozza (SISSA)
Class Duration	6 hours
Laboratory Duration	12 hours
Module Description	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).
Main Topics	● Introduction to RB methods, offline-online computing, elliptic coercive affine problems ● Sampling, greedy algorithm, POD ● A posteriori error bounds ● Primal-Dual Approximation ● Time dependent problems: POD-greedy sampling ● Non-coercive problems ● Approximation of coercivity and inf-sup parametrized constants ● Geometrical parametrization ● Reference worked problems ● Examples of Applications in CFD
Objectives	On successful completion of this module students should know the basic aspects of numerical approximation and efficient solution of parametrized PDEs for computational mechanics problems (heat and mass transfer, linear elasticity, viscous and potential flows).

Module 2.5:	**The Finite Element Method Using deal.II**
Lecturer	Luca Heltai (SISSA)
Class Duration	12 hours
Laboratory Duration	16 hours
Module Description	Hands-on module that guides the students to solve a simple poisson problem
Main Topics	● Generation simple meshes ● Degrees of Freedom - Matrix Sparsities ● A Laplace Solver in 2D ● Dimension independent Laplace Solver ● Adaptively refined meshes ● Hanging nodes and other constraints ● A Parallel Laplace Solver in 2D
Objectives	On successful completion of this module students should be able to understand existing codes for the solution of PDEs, and to develop efficient HPC enabled scientific codes dedicated to the solution of PDEs using existing parallel libraries and tools, dealII in particular.

Module 2.6:	**Finite Volumes and Discontinuous Galerkin methods**
Lecturer	Olindo Zanotti (UniTN)
Class Duration	9 hours
Laboratory Duration	12 hours
Module Description	An introduction to to Finite Volume Method for solving PDEs.
Main Topics	● Non-conservative finite Differences ● Finite Volumes Method ● Discontinuous Galerkin Methods
Objectives	On successful completion of this module students will be able to solve a class of PDEs using FVM.

Module 2.7:	**Fast Fourier Transforms in Parallel and Multiple Dimensions**
Lecturer	Ralph Gebauer (ICTP) Ivan Girotto (ICTP)
Class Duration	4 hours
Laboratory Duration	8 hours
Module Description	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
Main Topics	● Discrete and Continuous Fourier Transform ● The Cooley-Tukey FFT algorithm ● Implementation of an MPI parallel multi-dimensional FFT based on a 1d FFT ● Example use in a diffusion problem
Objectives	On successful completion of this module students should be able to integrate parallel FFTs into applications.

Module 2.8:	Molecular Dynamics
Lecturer	Giovanni Bussi (SISSA) Sandro Bottaro (SISSA)
Class Duration	6 hours
Laboratory Duration	10 hours
Module Description	Theory and applications of molecular dynamics simulations."Hands-on exercises on neighbor-list, linked cells, and parallelization strategies
Main Topics	● Introduction to molecular dynamics ● Neighbor lists and linked cells ● Parallelization using MPI ● Implementation of multi-replica algorithms (parallel tempering)
Objectives	Upon completion students will be able to run molecular dynamics simulations of a Lennard-Jones system and to optimize and parallelize a molecular dynamics code. Additionally, they will be able to implement parallel tempering algorithms.

Module 2.9:	**Monte Carlo method**
Lecturer	Sebastiano Pilati (ICTP) Stefano Baroni (SISSA)
Class Duration	10 hours
Laboratory Duration	6 hours
Module Description	Theory and applications of the Monte Carlo methods. Hands-on with examples, analysis of simulations and parallelization
Main Topics	● Review of Probability theory ● Pseudo-random numbers generators, Parallel RNG (SPRNG) ● Parallel Libraries for random generators ● Crude Monte Carlo Integration ● Markov Chain MC: Metropolis-Hastings Algorithm
Objectives	Upon completion students will be able to optimize software for the general application of the Monte Carlo Method on parallel systems having a good knowledge of the most advanced library for RNG.

Module 2.10:	**Approximation and interpolation of simple and complex functions**
Lecturer	Nicola Seriani (ICTP) Axel Kohlmeyer (Temple U. & ICTP)
Class Duration	12 hours
Laboratory Duration	12 hours
Module Description	Introduction to several techniques for efficient approximation of numerical functions to varying degrees of accuracy
Main Topics	● Interpolation (linear, spline) and errors (number and spacing of interpolation points) ● Cost of interpolation vs. explicit function evaluation of complex functions ● Approximation (taylor/maclaurin, pade). ● Lookup plus newton-raphson ● Range reduction as an efficient way to approximate exp(), log(),sin(), cos() and others in combination with a spline table or pade approximation for a small interval. ● Floating point math tricks
Objectives	Upon completion of this module students will be able to implement efficient approximations and tabulations of numerical functions and determine the accuracy of the approximations.

Module 2.11:	**Cluster Analysis**
Lecturer	Alex Rodriguez (SISSA) Alessandro Laio (SISSA)
Class Duration	9 hours
Laboratory Duration	12 hours
Module Description	Theory and applications of Clustering algorithms.
Main Topics	● Motivation for Clustering ● Similarities and Distances ● Flat, fuzzy and Hierarchical clustering methods ● Clustering methods examples ● External and Internal Validation ● Clustering applications
Objectives	The students should be able to implement a clustering algorithm method and to choose the one that fits better the problem that they want to solve.

Module 2.12:	Constrained Optimization
Lecturer	Riccardo Zanella (UNIFE - Math. and Comp. Sci. Department)
Class Duration	9 hours
Laboratory Duration	12 hours
Module Description	Advanced Constrained Optimisation Applications with CUDA
Main Topics	● Stationary point characterization for constrained optimization problems. ● Definition of projection problem, examples using L1 norm. ● Algorithms for L2 minimization with sparsity constraints.
Objectives	On successful completion of this module students should be able to provide a parallel CUDA code for the L2 norm minimization with sparsity constraints.

Useful links and resources

- **MPI**
  - http://www.mpi-forum.org/

    (MPI documents, including standard specifications with examples, forum for discussing standard specifications)

- **OPENMP**
- http://openmp.org/wp/

  (OpenMP specifications, examples, news on conferences and schools on parallel programming, forum for discussing standard specifications)

- **OPENACC**

  (OpenACC standards for accelerator directives, forums and tutorials)

- **GPGPU**
  - http://hgpu.org/
  - http://gpgpu.org/

  (These are two very frequently updated blogs with software release news, specifications, papers, thesis and codes on GPGPU programming as well as conferences, workshops and schools on HPC world)

  - http://developer.nvidia.com/cuda-zone

  (Commercial NVIDIA CUDA website with news and tutorials)

  - http://www.khronos.org/opencl/

  (OpenCL standards, reference cards and information on commercial and open source implementations)

- **OTHER MATERIALS**

  - **Intel Parallel Universe Magazine**

  (This is a quarterly publication devoted to exploring innovations in the world of software development with stories and tutorials, especially from Intel software enginners and updates on new Intel products and features)

  - **HPCWIRE**

  http://www.hpcwire.com/

  (News from HPC world and vendors)

- **SOME INTERESTING MATERIAL ON EXASCALE**

  - **http://www.cse.nd.edu/Reports/2008/TR-2008-13.pdf**
  - **http://science.energy.gov/~/media/ascr/pdf/research/am/docs/EMWGreport.pdf**