High Performance Scientific Computing

ECTS
3 crédits

Hourly volume
59h

Introducing

Objectives

At the end of this module, the student will have understood and be able to explain (main concepts):

· the principle of Krylov's methods to solve linear systems or compute eigenvalues and eigenvectors,
· the concept of preconditioning, the construction and use of preconditioners,
· theory and basic concepts of direct methods for sparse linear systems. Operating complexity and parallelism of direct methods,
· basic notions of parallel computer architecture, programming models for shared memory (OpenMP) and distributed memory (MPI) systems,
· basic concepts and methods for analyzing the performance of a parallel algorithm or code (Amdahl's law, cache hierarchy, principles of spatial and temporal locality, roofline model, critical path computation and high and low scalability).

At the end of this module, the student should be able to:

· evaluate the costs (flops/memory) of the different methods,
· analyze the influence of preconditioners,
· use high-level languages for the discretization of partial differential equations,
· program solvers, to parallelise simple codes according to the most adequate standard and to execute them on the appropriate resources,
· to analyse the efficiency of a method with regard to the operational complexity, the computing time and the memory footprint used in a high-performance computing perspective.

Necessary prerequisites

· Courses in Linear Algebra or Scientific Calculus, in particular the factorization methods LU or Cholesky
· Basics of computer architecture and imperative programming languages

Practical info

Location(s)

Toulouse