

# PX925-10 High Performance Computing in Physics

**26/27**

**Department**

Physics

**Level**

Taught Postgraduate Level

**Module leader**

Albert Bartok-Partay

**Credit value**

10

**Module duration**

10 weeks

**Assessment**

100% coursework

**Study location**

University of Warwick main campus, Coventry

---

## Description

### Introductory description

The module will cover advanced aspects of computer simulation and data analysis on high performance computers in all fields of computational physics and other sciences. Computing skills are greatly valued across science and beyond and we encourage students to go as far as they can to develop such skills.

### Module aims

To continue your training in the use of scientific computing by exploring the use of parallel computing. The module teaches you how to write scalable, portable programs for parallel computer systems using MPI, OpenMP and GPU accelerators, and explores how large-scale physics problems are tackled.

### Outline syllabus

This is an indicative module outline only to give an indication of the sort of topics that may be covered. Actual sessions held may differ.

Programming for efficiency. Modern cache architectures and CPU pipelining. Avoiding expensive and repeated operations. Compiler optimisation flags. Profiling tools.

Introduction to parallel computing. Modern HPC hardware and parallelisation strategies. Applications in Physics, large scale problems need parallel computers.

Shared memory programming. The OpenMP standard. Parallelisation using compiler directives. Threading and variable types. Loop and sections constructs. Program correctness and reproducibility. Scheduling and false sharing as factors influencing performance.

Distributed memory programming. The MPI standard for message passing. Point-to-point and collective communication. Synchronous vs asynchronous communication. MPI communicators and topologies.

GPU programming. CUDA vs OpenCL. Kernels and host-device communication. Shared and constant memory, synchronicity and performance. GPU coding restrictions.

Limitations to parallel performance. Strong vs weak scaling. Amdahl's law. Network contention in modern many-core architectures. Mixed mode OpenMP+MPI programming.

## **Learning outcomes**

By the end of the module, students should be able to:

- Write a parallel program using shared-memory or message passing constructs, for physics applications
- Understand the concepts underlying parallel computing, and discuss their general areas of applicability.
- Be able to measure and optimise the performance of a serial or parallel application

## **Interdisciplinary**

Computational skills applicable to subjects across the STEM spectrum.

## **Subject specific skills**

Skills in computer-modelling, reasoning, thinking. Knowledge of mathematics and physics

## **Transferable skills**

Analytical, communication, computing and general IT, problem-solving, self-study

---

## **Study**

## **Study time**

Type	Required
Lectures	15 sessions of 1 hour (25%)
Private study	45 hours (75%)
Total	60 hours

## Private study description

The course requires an in-depth knowledge of several programming paradigms, which are likely to need extensive background reading and practical experience beyond the lectures.

## Costs

No further costs have been identified for this module.

## Assessment

You must pass all assessment components to pass the module.

### Assessment group A

Assessment component	Weighting	Study time	Eligible for self-certification
5 Programming Assignments	100%	40 hours	No

The course is examined by a series of programming assignments.

1. A time-limited programming test which must be completed and submitted during the first workshop
2. A 1-week programming assignment on code optimisation
3. A 2-week programming assignment on shared memory programming
4. A 2-week programming assignment on distributed memory programming
5. A longer programming and performance measurement assignment

Each assignment will involve preparing a short report (roughly 1-4 pages for the first few assignments, more for the final one) which will include comments on your submitted program, answers to questions in the script, validation tests, performance data and associated discussion.

Reassessment component is the same

## **Feedback on assessment**

Written feedback supplied by markers, a few paragraphs per assignment.

---

## **Availability**

### **Pre-requisites**

Knowledge of either C or Fortran Programming languages, eg via PX390 Scientific Computing module or other route.

There is currently no information about the courses for which this module is core or optional.