

# PX457-15 High Performance Computing and Machine Learning

**24/25**

**Department**

Physics

**Level**

Undergraduate Level 4

**Module leader**

Nicholas Hine

**Credit value**

15

**Module duration**

15 weeks

**Assessment**

100% coursework

**Study location**

University of Warwick main campus, Coventry

---

## Description

### Introductory description

The module will address the increased use of computer simulation and deep learning techniques on high performance computers in all fields of computational science. Computing skills, particularly parallel programming and machine learning, are greatly valued across science and beyond and we encourage students to go as far as they can to develop such skills.

[Module web page](#)

### Module aims

To explain the methods used in computer modelling, simulation and machine learning on high performance computing architectures, for research in computational physics and other sciences.

### 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.

The HPC Landscape: hardware and software. Compiled vs Interpreted Languages. Efficiency and

pipelining. Modern cache architectures and CPU pipelining. Avoiding expensive and repeated operations. Compiler optimisation techniques.

Introduction to parallel computing. Modern HPC hardware and parallelisation strategies. Analysing algorithms and codes to identify opportunities for parallelism.

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.

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

GPU programming. Low-level languages: CUDA vs OpenCL. Kernels and host-device communication. Shared and constant memory, synchronicity and performance. GPU coding restrictions. High-level languages: TensorFlow, PyTorch, JAX, etc. Automatic differentiation.

Introduction to Machine Learning. Key concepts: parameterisation, loss functions, regularisation. Types of ML: Supervised/Unsupervised. Key tasks: dimension reduction, clustering, regression, function approximation (universal approximators).

Deep Learning and Neural Networks. Backpropagation. Convolutional Neural Networks for computer vision; Residual NNs, Recurrent NNs, latent space, VAEs, Generative AI, Attention & Transformers.

Graph Neural Networks. Symmetry in Machine Learning. Message passing on graphs. Euclidean Neural Networks. Machine Learned Interatomic Potentials. Atomistic simulations with MLIPs.

## **Learning outcomes**

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

- Identify and correct common inefficiencies in serial scientific computer codes
- Write a parallel program using shared-memory or message-passing constructs in a physics context, write a simple GPU accelerated program, and write simple neural network models
- Choose an appropriate programming paradigm and identify sources of performance bottlenecks and parallelisation errors in parallel computer programs and understand how these relate to the computer architecture
- Design, retrain and use existing state-of-the-art machine learning models to extract efficiently functional forms describing complex data such as electron micrographs and ab initio potential energy surfaces

## **Indicative reading list**

R Chandra et. al., Parallel Programming in OpenMP , Morgan Kaufmann, P Pacheco, Parallel Programming with MPI, Morgan Kaufmann

M Quinn, Parallel Programming in C with MPI and OpenMP, McGraw-Hill

D Kirk and W Hwu, Programming Massively Parallel Processors, Elsevier

[View reading list on Talis Aspire](#)

## **Interdisciplinary**

High performance computing is behind almost all advances in AI, data analysis and modelling. It relies on efficient algorithms, parallel processing and machine learning, which are the techniques discussed in this module. Where an illustrative context is needed, it will usually be taken from physics but the universality of the techniques and ideas is emphasised throughout.

## **Subject specific skills**

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

## **Transferable skills**

Analytical, communication, problem-solving, self-study

---

## **Study**

### **Study time**

<b>Type</b>	<b>Required</b>
Lectures	30 sessions of 1 hour (20%)
Private study	120 hours (80%)
Total	150 hours

### **Private study description**

Working through lecture notes, formulating problems, writing and testing code, discussing with others taking the module, preparing and submitting coursework

### **Costs**

No further costs have been identified for this module.

---

## **Assessment**

You must pass all assessment components to pass the module.

### **Assessment group A1**

**Weighting****Study time**

Assessed Computing Assignments  
Submission of Computer Codes

100%

**Feedback on assessment**

Personal tutor, group feedback

---

**Availability****Courses**

This module is Optional for:

- UPXA-FG31 Undergraduate Mathematics and Physics (MMathPhys)
  - Year 4 of FG31 Mathematics and Physics (MMathPhys)
  - Year 4 of FG31 Mathematics and Physics (MMathPhys)
- Year 4 of UPXA-F303 Undergraduate Physics (MPhys)

This module is Core option list A for:

- Year 4 of UPXA-F3FA Undergraduate Physics with Astrophysics (MPhys)