

# PX390-15 Scientific Computing

**21/22**

**Department**

Physics

**Level**

Undergraduate Level 3

**Module leader**

Bogdan Hnat

**Credit value**

15

**Module duration**

15 weeks

**Assessment**

100% coursework

**Study location**

University of Warwick main campus, Coventry

---

## Description

### Introductory description

This module should help you develop C programming skills. The module will consist of some lectures and a series of programming exercises designed to illustrate important aspects of program design. The module will also cover some important numerical techniques used in data processing in physics. Aspects relating to the reliability, accuracy and efficiency of these techniques will be discussed, as well as other issues such as making software user friendly, and data transfer between platforms. The module will be assessed on the basis of the exercises completed during the module and some project work.

[Module web page](#)

### Module aims

To develop proficiency in the solution of physics problems and to understand the fundamentals of numerical computation using the C programming language, which underpins modern software.

### 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.

1. The C Syntax and the development cycle. The underlying representation of data in hardware will be discussed, and the methods by which the C language reveals this low-level description through the use of pointers.
2. Tools for debugging and verifying C programs will be introduced, and coding standards and effective programming techniques introduced to reinforce the concepts of modularity and clarity in code writing.
3. Numerical representation of mathematical objects, and, in particular, multi-dimensional functions, will be introduced. Numerical integration will be used to illustrate concepts of numerical accuracy and the relationship of numerical methods to function properties.
4. Algorithms for solving chaotic and stochastic differential equations will be explored, and robustness and reproducibility will be quantified.
5. The finite difference method will be explained, along with concepts of numerical convergence and stability.
6. Minimisation techniques and the use of the weak form of equations will be used to introduce finite-element and variational approaches.
7. Data structures, and in particular lists and tree algorithms. Use of these in physics problems, in particular, n-body codes.

## Learning outcomes

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

- Program in the C language, and be familiar with the development cycle and debugging tools.
- Explain how computer hardware implements numerical operations and the implications for numerical software
- Select appropriate algorithms to solve physics problems, including those that involve partial differential equations
- Test their code and determine the size of numerical errors.

## Indicative reading list

Steve Oualliane, 'Practical C Programming', O'Reilly Media, 1997

William H Press, Saul Teukolsky, William Vetterling and Brian Flannery, 'Numerical Recipes in C', Cambridge University Press, 2007

[View reading list on Talis Aspire](#)

## 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	10 sessions of 1 hour (7%)
Other activity	30 hours (20%)
Private study	110 hours (73%)
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

### Other activity description

15 x 2 hours Workshops

## Costs

No further costs have been identified for this module.

---

## Assessment

You do not need to pass all assessment components to pass the module.

### Assessment group A1

	Weighting	Study time
Assessed computing assignments	100%	
Submission of computer codes		

### Feedback on assessment

Written feedback on assessments, group feedback

---

## Availability

## Courses

This module is Option list A for:

- UMAA-G100 Undergraduate Mathematics (BSc)
  - Year 3 of G100 Mathematics
  - Year 3 of G100 Mathematics
  - Year 3 of G100 Mathematics
- Year 3 of UMAA-G103 Undergraduate Mathematics (MMath)
- Year 4 of UMAA-G101 Undergraduate Mathematics with Intercalated Year
- UPXA-F300 Undergraduate Physics (BSc)
  - Year 3 of F300 Physics
  - Year 3 of F300 Physics
  - Year 3 of F300 Physics
- Year 3 of UPXA-F303 Undergraduate Physics (MPhys)
- Year 4 of UPXA-F301 Undergraduate Physics (with Intercalated Year)

This module is Option list B for:

- UMAA-G105 Undergraduate Master of Mathematics (with Intercalated Year)
  - Year 3 of G105 Mathematics (MMath) with Intercalated Year
  - Year 5 of G105 Mathematics (MMath) with Intercalated Year
- UMAA-G103 Undergraduate Mathematics (MMath)
  - Year 3 of G103 Mathematics (MMath)
  - Year 3 of G103 Mathematics (MMath)
  - Year 4 of G103 Mathematics (MMath)
  - Year 4 of G103 Mathematics (MMath)
- UMAA-G106 Undergraduate Mathematics (MMath) with Study in Europe
  - Year 3 of G106 Mathematics (MMath) with Study in Europe
  - Year 4 of G106 Mathematics (MMath) with Study in Europe
- Year 3 of UPXA-FG33 Undergraduate Mathematics and Physics (BSc MMathPhys)
- UPXA-GF13 Undergraduate Mathematics and Physics (BSc)
  - Year 3 of GF13 Mathematics and Physics
  - Year 3 of GF13 Mathematics and Physics
- UPXA-FG31 Undergraduate Mathematics and Physics (MMathPhys)
  - Year 3 of FG31 Mathematics and Physics (MMathPhys)
  - Year 3 of FG31 Mathematics and Physics (MMathPhys)
- Year 4 of UPXA-GF14 Undergraduate Mathematics and Physics (with Intercalated Year)
- Year 3 of UPXA-F303 Undergraduate Physics (MPhys)