MA3K7-15 Problem Solving with Python

23/24

Department
Warwick Mathematics Institute
Level
Undergraduate Level 3
Module leader
Siri Chongchitnan
Credit value
15
Module duration
10 weeks
Assessment
100% coursework
Study location
University of Warwick main campus, Coventry

Description

Introductory description

Mathematicians are used to solving problems, but few have really thought about what the problem-solving process really involves. The great mathematician George Pòlya was one of the pioneers who regarded problem solving as a subject worthy of studying on its own. He subsequently developed what he called heuristics (i.e. a recipe) for problem solving, and this was further refined by mathematicians over the past decades.
In this module, we will not only explore these problem-solving strategies in detail, but we will also learn how even a basic knowledge of Python can be used to greatly enhance these strategies and make problem solving an enjoyable and mathematically enriching experience.

Module aims

We will examine the process of mathematical problem solving, with the use of Python as an exploratory tool. Problem-solving skills will be developed through a system of rubrics, which allow each problem to be approached systematically in distinct phases. Python will used not just as a tool to perform elaborate calculations, but also for visualisation and simulation, hence allowing each problem to be explored more freely, deeply and efficiently than the pen-and-paper approach.
.

You will work both individually and in groups on mathematical problems that are challenging, unfamiliar and often open-ended. The problem-solving, critical-thinking and programming skills learnt in this course are highly desirable and transferrable, whether you go on to further study or into the job market.

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 phases of problem-solving according to Pólya, Mason and others
- Using Python to enhance the problem-solving phases
- Writing a problem-solving rubric
- Making conjectures and dealing with being stuck
- Justifying and convincing
- Asking question and extending a given mathematical problem

Learning outcomes

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

- Use an explicit scheme to organise your approach to solving mathematical problems.
- Use Python for calculations and visualisation of problems, aiding and enriching your solution.
- Critically evaluate your own problem-solving practice

Indicative reading list

- Mason, Burton and Stacey, Thinking mathematically, 2nd ed., Pearson (2010)
- Linge and Langtangen, Programming for Computations - Python, 2nd ed ., Springer (2019)
- Pólya, How to solve it, 2nd ed., (1990)
- Grieser, Exploring mathematics: problem-solving and proof, Springer (2018)

View reading list on Talis Aspire

Interdisciplinary

This module is a combination of the following broad fields:
-Mathematics
-Computing
-Education

Subject specific skills

-Ability to use rubrics to aid problem solving
-Ability to use Python to investigate challenging mathematical problems
-Ability reflect on one's problem-solving and critical-thinking skills

Transferable skills

-Programming in Python
-Group work
-Communication of scientific ideas to a wide range of audience
-Scientific writing skills
-Essay writing skills

Study

Study time

Type	Required
Lectures	10 sessions of 1 hour (7%)
Seminars	10 sessions of 1 hour (7%)
Online learning (scheduled sessions)	10 sessions of 1 hour (7%)
Private study	120 hours (80%)
Total	150 hours

Private study description

Homework problems.

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 A

Assessment component	Weighting	Study time	Eligible for self-certification
Problem sets	50%	3 hours	No

Five take-home problem sets involving rubric writing and Python coding

	Weighting	Study time	Eligible for self-certification

Reassessment component is the same

Assessment component

Mini project	50%	10 hours	No

A final substantial piece of work involving rubric writing and Python coding.

Reassessment component is the same

Feedback on assessment

Written feedback.

Availability

Pre-requisites

To take this module, you must have passed:

- All of
  - MA124-6 Mathematics by Computer

Anti-requisite modules

If you take this module, you cannot also take:

- MA3E7-15 Problem Solving

Courses

This module is Core option list B for:

- Year 3 of UMAA-GV17 Undergraduate Mathematics and Philosophy
- Year 3 of UMAA-GV19 Undergraduate Mathematics and Philosophy with Specialism in Logic and Foundations

This module is Core option list D for:

- Year 4 of UMAA-GV19 Undergraduate Mathematics and Philosophy with Specialism in Logic and Foundations

This module is Option list A for:

- Year 3 of UMAA-G100 Undergraduate Mathematics (BSc)
- Year 3 of UMAA-G103 Undergraduate Mathematics (MMath)
- Year 4 of UMAA-G101 Undergraduate Mathematics with Intercalated Year

This module is Option list B for:

- UMAA-G105 Undergraduate Master of Mathematics (with Intercalated Year)
    - Year 4 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 4 of G103 Mathematics (MMath)
- Year 4 of UMAA-G106 Undergraduate Mathematics (MMath) with Study in Europe