# MA934-15 Numerical Algorithms and Optimisation

#### 23/24

Department Warwick Mathematics Institute Level Taught Postgraduate Level Module leader Radu Cimpeanu Credit value 15 Module duration 10 weeks Assessment 60% coursework, 40% exam Study location University of Warwick main campus, Coventry

## Description

## Introductory description

This module provides students with knowledge (and practice) of important numerical optimisation concepts at the intersection between mathematics and scientific computing. Algorithmic structures, data structures, numerical method construction and performance assessment will form key parts of the module, with applications and use cases concentrated on topics in linear algebra, signal processing and optimisation.

Module web page

#### Module aims

Numerical Algorithms and Optimisation teaches students the theory and implementation of a set of computational algorithms that provide the fundamental toolkit for advanced data analysis, simulation and optimisation. The syllabus will be drawn from the following list of topics: algorithmic structures (iteration, recursion, memorization) and computational complexity, data structures (linked lists, stacks and queues, binary indexed trees), sorting and search algorithms, Fast Fourier Transform, automatic differentiation, linear systems and the Conjugate Gradient algorithm, Singular Value Decomposition, convex and nonconvex optimisation, constrained optimisation,

linear programming, Dijkstra's algorithm and dynamic programming, discrete-event simulation.

# **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.

- algorithmic structures (iteration, recursion, memorization) and computational complexity
- data structures (linked lists, stacks and queues, binary indexed trees)
- sorting and search algorithms
- Fast Fourier Transform and its applications
- Topics in numerical linear algebra: solving linear systems, conjugate gradient algorithm, singular value decomposition
- unconstrained continuous optimisation: multivariate minimisation, Nelder-Mead algorithm, automatic differentiation, gradient descent
- constrained continuous optimisation: method of Lagrange multipliers, linear programming
- discrete optimisation: Dijkstra's algorithm, dynamic programming, combinatorial optimisation

# Learning outcomes

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

- Demonstrate a deep fundamental understanding of the most important computational algorithms used for advanced data analysis, mathematical modelling and optimisation of complex systems.
- Understand algorithmic structures like iteration, recursion and memorization and be able to apply them.
- Perform mathematical manipulation and implement computational problem-solving techniques.
- Identify the most appropriate approach for computational solution of a mathematical problem and understand problems that can arise such as numerical error, poor conditioning or instability.
- Design and apply both discrete and continuous approaches depending on the requirements of a particular problem.
- Develop efficient code for the computational solution and analysis and optimisation problems, select appropriate data structures and algorithms, present and visualise algorithm outputs and results of analyses in a clear and informative way.
- Distinguish the computational complexity of an algorithm and the practical constraints it imposes on problem solving.

# Indicative reading list

Lloyd N. Trefethen and David Bau, Numerical Linear Algebra, 3rd Edition, SIAM, 1997. William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. Numerical recipes: The art of scientific computing, 3rd Edition, Cambridge University Press, New York, NY, USA, 2007.

David Kincaid and Ward Cheney. Numerical analysis : mathematics of scientific computing, 3rd

Edition, American Mathematical Society, 2009. Research articles in the field will complement the textbooks above.

## Interdisciplinary

The module sits naturally between mathematics, computer science and elements from related disciplines such as physics, biology or image processing.

#### Subject specific skills

See learning outcomes: data structure recognition and usage, discrete mathematical analysis, algorithm construction, performance analysis, advanced linear algebra, optimisation techniques.

#### Transferable skills

Students will acquire key reasoning and problem solving skills which will empower them to address new problems with confidence. This will be grounded in enhanced software development knowledge (including transferable 'software carpentry' skills such as version control and continuous integration testing) and deployed for a wide variety of problems in interdisciplinary contexts.

# Study

## Study time

Туре	Required	
Lectures	30 sessions of 1 hour (20%)	
Supervised practical classes	10 sessions of 1 hour (7%)	
Private study	48 hours (32%)	
Assessment	62 hours (41%)	
Total	150 hours	

# Private study description

The students will engage with the material via exercises and assignments, which are constructed as an embodiment of theoretical material learned during the lectured components of the module. Lecture and practical class preparation (20h), engaging with online material (16h) and work on formative elements (12h, be they pen and paper based or programming) is anticipated to fill up the rest of the independent learning time. Checkpoints will be provided along the way, and time allowance for software installation and early steps, as well as exam revision is also considered.

# Costs

No further costs have been identified for this module.

#### Assessment

You must pass all assessment components to pass the module.

#### Assessment group D1

Weighting	Study time	Eligible for self- certification	
60%	50 hours	Yes (extension)	
Assignments with feedback provided within 2 weeks from submission, so as to enable feedforward into the following assessment and ultimately study. These will combine theoretical work (derivation, proofs, algorithmic formulation), implementation, data visualisation and interpretation. Discussions about the wider implications of the material will also be encouraged. Reassessment component is the same			
40%	12 hours	No	
	Weighting 60% k provided within 2 wee ving assessment and ult Igorithmic formulation), s about the wider implication me	WeightingStudy time60%50 hours60%50 hoursck provided within 2 weeks from submission ving assessment and ultimately study. These lgorithmic formulation), implementation, dat a about the wider implications of the materiame40%12 hours	

Reassessment component is the same

#### Feedback on assessment

Written feedback on assignments (annotated workbooks) plus informal/general oral feedback during classwork sessions. Written feedback from examiners on the oral examination.

Past exam papers for MA934

# Availability

# Courses

This module is Core for:

- Year 1 of RMAA-G1PG Postgraduate Research Mathematics of Systems
- TMAA-G1PF Postgraduate Taught Mathematics of Systems
  - Year 1 of G1PF Mathematics of Systems
  - Year 1 of G1PF Mathematics of Systems
- Year 1 of TESA-H1B1 Postgraduate Taught Predictive Modelling and Scientific Computing

This module is Optional for:

• Year 2 of TPXA-F345 Postgraduate Taught Modelling of Heterogeneous Systems (PGDip)

This module is Option list B for:

• Year 1 of TPXA-F345 Postgraduate Taught Modelling of Heterogeneous Systems (PGDip)