Module Specification

Computational Mathematics

Version: 2023-24, v2.0, 15 May 2023

Contents

Part 1: Information

Module title: Computational Mathematics

Module code: UFMFGH-30-M

Level: Level 7

For implementation from: 2023-24

UWE credit rating: 30

ECTS credit rating: 15

Faculty: Faculty of Environment & Technology

Department: FET Dept of Computer Sci & Creative Tech

Partner institutions: None

Field: Computer Science and Creative Technologies

Module type: Module

Pre-requisites: Numerical Analysis 2023-24

Excluded combinations: None

Co-requisites: None

Continuing professional development: No

Professional, statutory or regulatory body requirements: None

Part 2: Description

Overview: Not applicable

Features: Not applicable

Educational aims: In this module students will investigate problems, methods and techniques from the field of computational mathematics, specifically the areas of numerical linear algebra and numerical optimization. We will consider applications,

formulate algorithms and investigate properties of the algorithms such as convergence, stability and computational complexity.

Outline syllabus: The following syllabus is indicative:

Numerical Linear Algebra:

Systems of linear equations (LU factorisation, Cholesky factorisation)

Linear least-squares problems (normal equations, QR factorisation, modified Gram-Schmidt method, Householder and Givens transformations)

Eigenvalue and singular value problems (power iteration, inverse iteration, Rayleigh quotient, subspace iteration, Jacobi method, QR method, bidiagonalisation)

Iterative methods for systems of linear equations (Jacobi, Gauss-Seidel, SOR, Krylov, multigrid, domain decomposition, Schur complement)

Numerical Optimization:

Nonlinear equations (Newton method, secant method)

Unconstrained optimization (steepest descent, Newton method, conjugate gradients, Gauss-Newton, Levenberg-Marquardt, BFGS)

Constrained optimization (KKT conditions, linear programming, interior point methods)

Possible application areas: regression, numerical integration, solving differential equations, machine learning, image processing, signal processing.

Part 3: Teaching and learning methods

Teaching and learning methods: Typically the scheduled teaching hours take the form of: whole group lectures, used to deliver new material and to consolidate previous material, small group classroom tutorials with activities designed to reinforce and enhance students' understanding of the lecture material and small group computing laboratory sessions designed to develop the students' ability to generate and utilise software and to analyse software output.

Module Learning outcomes: On successful completion of this module students will achieve the following learning outcomes.

MO1 Describe, derive and interpret algorithms for problems in computational mathematics

MO2 For algorithms in computational mathematics analyse aspects such as error convergence, stability and computational complexity

MO3 Compare algorithms and choose and apply the appropriate algorithm or construct a combination of algorithms for a problem or application in computational mathematics

MO4 Implement algorithms and design and evaluate tests for these algorithms using a computer programming environment

Hours to be allocated: 300

Contact hours:

Independent study/self-guided study = 228 hours

Face-to-face learning = 72 hours

Total = 300

Reading list: The reading list for this module can be accessed at readinglists.uwe.ac.uk via the following link https://uwe.rl.talis.com/index.html

Part 4: Assessment

Assessment strategy: The assessment will involve two equally weighted components; a written examination and coursework that will require students to

demonstrate both practical skills and theoretical knowledge of a numerical analyst.

The first two learning outcomes will be assessed in both the exam and the coursework. The last two learning outcomes will be assessed primarily in the coursework.

Each coursework assignment will be a short piece of work designed to provide students the opportunity to investigate the implementation of computational methods and obtain feedback. Each assignment will result in a written report and the creation of computer code. The assignment tasks will involve implementation of algorithms, choosing and applying methods to an application and comparing methods. Due to the nature of the module the output will include a significant amount of graphical and tabular output which will potentially lead to a high page count. For some tasks, individualised data sets will be used to discourage plagiarism.

Assessment tasks:

Report (First Sit)
Description: Written report 1 (15 pages)
Weighting: 25 %
Final assessment: No
Group work: No
Learning outcomes tested: MO1, MO2, MO3, MO4

Report (First Sit)
Description: Written report 2 (15 pages)
Weighting: 25 %
Final assessment: No
Group work: No
Learning outcomes tested: MO1, MO2, MO3, MO4

Examination (First Sit)
Description: Written examination (3 hours)
Weighting: 50 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO2

Report (Resit)

Description: Written report 1 (15 pages)

Weighting: 25 %

Final assessment: No

Group work: No

Learning outcomes tested: MO1, MO2, MO3, MO4

Report (Resit)

Description: Written report 2 (15 pages)

Weighting: 25 %

Final assessment: No

Group work: No

Learning outcomes tested: MO1, MO2, MO3, MO4

Examination (Resit)

Description: Written examination (3 hours)

Weighting: 50 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO2

Part 5: Contributes towards

This module contributes towards the following programmes of study:

Mathematics [Sep][FT][Frenchay][4yrs] - Not Running MMath 2020-21

Mathematics [Sep][SW][Frenchay][5yrs] MMath 2019-20