STUDENT AND ACADEMIC SERVICES

**MODULE SPECIFICATION**

| Part 1:  Information | | | |
|---|---|---|---|
| Module Title | Computational Mathematics | | |
| Module Code | UFMFGH-30-M | Level | Level 7 |
| For implementation from | 2018-19 | | |
| UWE Credit Rating | 30 | ECTS Credit Rating | 15 |
| Faculty | Faculty of Environment & Technology | Field | Engineering, Design and Mathematics |
| Department | FET Dept of Engin Design & Mathematics | | |
| Contributes towards | | | |
| Module type: | Standard | | |
| Pre-requisites | Numerical Analysis 2018-19 | | |
| Excluded Combinations | None | | |
| Co- requisites | None | | |
| Module Entry requirements | None | | |

| Part 2: Description |
|---|
| **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.<br><br>**Outline Syllabus:** The following syllabus is indicative:<br><br>Numerical Linear Algebra:<br><br>Systems of linear equations (LU factorisation, Cholesky factorisation)<br><br>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.

**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.

---

| **Part 3: Assessment** |
|---|

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.

| First Sit Components | Final Assessment | Element weighting | Description |
|---|---|---|---|
| Report - Component B | | 25 % | Written report 1  (15 pages) |
| Report - Component B | | 25 % | Written report 2  (15 pages) |
| Examination - Component A | ✓ | 50 % | Written examination (3 hours) |
| Resit Components | Final Assessment | Element weighting | Description |
| Report - Component B | | 50 % | Written report (30 pages) |
| Examination - Component A | ✓ | 50 % | Written examination (3 hours) |

STUDENT AND ACADEMIC SERVICES

| Part 4:  Teaching and Learning Methods | | |
|---|---|---|
| Learning Outcomes | On successful completion of this module students will be able to: | |
| | **Module 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 |
| Contact Hours | **Contact Hours** | |
| | **Independent Study Hours:** | |
| | Independent study/self-guided study | 228 |
| | **Total Independent Study Hours:** | 228 |
| | **Scheduled Learning and Teaching Hours:** | |
| | Face-to-face learning | 72 |
| | **Total Scheduled Learning and Teaching Hours:** | 72 |
| | **Hours to be allocated** | 300 |
| | **Allocated Hours** | 300 |
| Reading List | *The reading list for this module can be accessed via the following link:* | |
| | https://uwe.rl.talis.com/index.html | |