The objective of this module is to introduce model formulation for various chemical, biochemical and environmental processes, and numerical techniques in solving the associated equations. The module introduces a variety of numerical methods and their application to the solution of problems in the chemical engineering field. These problems include linear and nonlinear algebraic equations, root problems, numerical optimization, finite difference methods, interpolation, linear and nonlinear regression analysis, differentiation and integration, and ordinary differential equations (initial value and boundary value problems). MATLAB will be introduced and extensively used as the computing tool to solve all the above-mentioned problems. Students will learn both the object oriented programming and command line modes of MATLAB and apply them to the solution of a variety of problems involving optimization and dynamic simulation of engineering processes.

**Pre-requisites:**
Engineering Science 036a/b or Computer Science 026a/b or the former Computer Science 036a/b

Unless you have either the prerequisites for this course or written special permission from your Dean to enroll in it, you may be removed from this course and it will be deleted from your record. This decision may not be appealed. You will receive no adjustment to your fees in the event that you are dropped from a course for failing to have the necessary prerequisites.

**Co-requisites:** None

**Anti-requisites:** None

**Contact Hours:** 3 lecture hours (Wednesdays 9:30-11:30, SEB-1059, and Wednesdays 4:30-5:30 HSB-240), 2 tutorial hours (Mondays 3:30-5:30, HCB14/16), 0.5 course.

**Instructor(s):**
Lars Rehmann (TEB 459) lrehmann@uwo.ca

Undergraduate Assistant: (TEB 477) 519-661-2111 ext: 82131
Course Notes: Course Notes will be available on the course’s OWL site.

Reference Texts: There are many Numerical Analysis books available in the library. Some of them are listed here. Students may make use of any of the following books.


Laboratory: None

Units: SI and other units will be used.

General Learning Objectives

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>√</td>
<td></td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>√</td>
<td></td>
</tr>
</tbody>
</table>


**Detailed Course outline and Learning objectives**

**Introduction to numerical software packages**

Introduction to Matlab, Scilab, GNU Octave, Freemat and Sage. Course predominately uses Matlab, which will be introduced in detail. Introduction to MATLAB, introducing the theory of variable and matrices, basic operations and plotting. Introduction to process modeling and simulation, Examples of chemical, Biochemical and environmental engineering problems arising in fluid mechanics, thermodynamics, heat and mass transfer, separation processes, reaction engineering, process dynamics, and transport phenomena. Introducing advanced visualizations commands in MATLAB. Introduction to basic built-in MATLAB functions. Getting familiar with building user-defined functions, and understanding the importance of them for solving advanced chemical engineering models. Practicing basic flow-control in MATLAB by applying *if, else, elseif, for, while* loops in various problems.

**Roots and Optimization**

After this section students will:

- Have understanding of what roots problems are and where they occur in engineering and science.
- Know how to determine a root graphically, and though the incremental search method and be aware of shortcomings.
- Know how to solve a roots problem with the bisection method and how to estimate the error of bisection and why it differs from error estimates for other types of root location algorithms.
- Recognize the difference between bracketing and open methods for root location.
- Know how to solve a roots problem with the Newton-Raphson method and appreciating the concept of quadratic convergence.
- Understand why and where optimization occurs in engineering and scientific problem solving and recognize the difference between one-dimensional and multi-dimensional optimization.
- Distinguish between global and local optima.
- Be able to define the golden ratio and understand why it makes one-dimensional optimization efficient.
- Be able to locate the optimum of a single-variable function with the golden-section search, with parabolic interpolation and Matlab’s fminbnd function.
- Be able to develop MATLAB contours and surface plots to visualize two-dimensional functions.
- Know how to apply the fminsearch function to determine the minimum of a multidimensional function.

**Linear Algebra**

After this section students will:

- Have and understanding matrix notation.
- Be able to identify the following types of matrices: identity, diagonal, symmetric, triangular, and tridiagonal.
- Know how to represent a system of linear equations in matrix form, and how to
solve linear algebraic equations with left division and matrix inversion in MATLAB

- Know how to solve small sets of linear equations with the graphical method and Cramer’s rule.
- Understand how to implement forward elimination and back substitution as in Gauss elimination.
- Understand the concepts of singularity and ill-condition.
- Understand how partial pivoting is implemented and how it differs from complete pivoting.
- Recognize how the banded structure of a tridiagonal system can be exploited to obtain extremely efficient solutions.
- Understand that LU factorization involves decomposing the coefficient matrix into two triangular matrices that can then be used to efficiently evaluate different right-hand-side vectors, and know how to express Gauss elimination as an LU factorization.
- Understand in general terms what happens when MATLAB’s backslash operator is used to solve linear systems.
- Know how to determine the matrix inverse in an efficient manner based on LU factorization, and how to use the matrix inverse to assess stimulus-response characteristics of engineering systems.
- Understand the meaning of matrix and vector norms and how they are computed, and how to use norms to compute the matrix condition number.
- Understand how the magnitude of the condition number can be used to estimate the precision of solutions of linear algebraic equations.
- Understand the difference between the Gauss-Seidel and Jacobi methods.
- Know how to assess diagonal dominance and knowing what it means.
- Recognize how relaxation can be used to improve convergence of iterative methods.
- Understand how to solve systems of nonlinear equations with successive substitution and Newton-Raphson.

Regression and Interpolation
After this section students will:

- Be familiar with some basic descriptive statistics and the normal distribution.
- Know how to compute the slope and intercept of a best-fit straight line with linear regression.
- Know how to compute and understand the meaning of the coefficient of determination and the standard error of the estimate.
- Understand how to use transformations to linearize nonlinear equations so that they can be fit with linear regression.
- Know how to implement linear regression with MATLAB.
- Know how to implement polynomial regression.
- Know how to implement multiple linear regression.
- Understand the formulation of the general linear least-squares model.
• Understand how the general linear least-squares model can be solved with MATLAB using either the normal equations or left division.
• Understand how to implement nonlinear regression with optimization techniques.
• Recognize that evaluating polynomial coefficients with simultaneous equations is an ill-conditioned problem.
• Know how to evaluate polynomial coefficients and interpolate with MATLAB’s polyfit and polyval functions.
• Know how to perform an interpolation with Newton’s polynomial and with a Lagrange polynomial.
• Know how to solve an inverse interpolation problem by recasting it as a roots problem.
• Appreciate the dangers of extrapolation by recognizing that higher-order polynomials can manifest large oscillations.
• Understand that splines minimize oscillations by fitting lower-order polynomials to data in a piecewise fashion.
• Recognize why cubic polynomials are preferable to quadratic and higher-order splines.
• Understand the differences between natural, clamped, and not-a-knot end conditions.
• Know how to fit a spline to data with MATLAB’s built-in functions.
• Understand how multidimensional interpolation is implemented with MATLAB.

Integration and Differentiation
After this section students will:
• Recognize that Newton-Cotes integration formulas are based on the strategy of replacing a complicated function or tabulated data with a polynomial that is easy to integrate.
• Knowing how to implement the following single application Newton-Cotes formulas: Trapezoidal rule (also as composite formula), Simpson’s 1/3 rule, and Simpson’s 3/8 rule (also as composite formula).
• Understand how Richardson extrapolation provides a means to create a more accurate integral estimate by combining two less accurate estimates.
• Understand how Gauss quadrature provides superior integral estimates by picking optimal abscissas at which to evaluate the function.
• Know how to use MATLAB’s built-in functions quad and quadl to integrate functions.
• Understand the application of high-accuracy numerical differentiation formulas for equispaced data, and know how to evaluate derivatives for unequally spaced data.
• Understand how Richardson extrapolation is applied for numerical differentiation.
• Recognize the sensitivity of numerical differentiation to data error.
• Know how to evaluate derivatives in MATLAB with the diff and gradient
functions.

• Know how to generate contour plots and vector fields with MATLAB.

Ordinary Differential Equations
After this section students will:

• Understand the meaning of local and global truncation errors and their relationship to step size for one-step methods for solving ODEs.
• Know how to implement the following Runge-Kutta (RK) methods for a single ODE: Euler, Heun, Midpoint, and Fourth-Order RK
• Know how to iterate the corrector of Heun’s method.
• Know how to implement the following Runge-Kutta methods for systems of ODEs: Euler, Fourth-order RK
• Understand how the Runge-Kutta Fehlberg methods use RK methods of different orders to provide error estimates that are used to adjust step size.
• Be familiar with the built-in MATLAB function for solving ODEs.
• Learn how to adjust options for MATLAB’s ODE solvers.
• Learn how to pass parameters to MATLAB’s ODE solvers.
• Understand what is meant by stiffness and its implications for solving ODEs.
• Understand the difference between initial-value and boundary-value problems.
• Know how to implement the shooting method for linear ODEs by using linear interpolation to generate accurate “shots.”
• Understand how derivative boundary conditions are incorporated into the shooting method.
• Know how to solve nonlinear ODEs with the shooting method by using root location to generate accurate “shots.”

Evaluation:
The final course mark will be determined as follows:
  Assignments 20%
  Quizzes 20%
  Midterm 20%
  Final Examination 40%

Examination will be conducted on a computer and will be open book. Only notes and access to OWL are allowed. Use of the internet for anything other than OWL is not allowed.

Note: Students must pass the final examination to pass this course. Students who fail the final examination will be assigned 48% if the aggregate mark is higher than 50%, or the aggregate mark. Assignments are to be handed in electronically through OWL on the specified due date provided by the Instructor, unless otherwise directed.

Use of English:
In accordance with Senate and Faculty Policy, students may be penalized up to 10% of the
marks on all assignments, tests, and examinations for the improper use of English. Additionally, poorly written work with the exception of the final examination may be returned without grading. If resubmission of the work is permitted, it may be graded with marks deducted for poor English and/or late submission.

**Attendance**

*Attendance in all lectures, tutorials and laboratories is mandatory.* Any student who, in the opinion of the instructor, is absent too frequently from class or laboratory periods in any course, will be reported to the Dean (after due warning has been given). On the recommendation of the Department concerned, and with the permission of the Dean, the student will be debarred from taking the regular examination in the course.

**Cheating**

University policy states that cheating is a scholastic offence. The commission of a scholastic offence is attended by academic penalties, which might include expulsion from the program. If you are caught cheating, there will be no second warning (see Scholastic Offence Policy in the Western Academic Calendar).

**Plagiarism**

Students must write their essays and assignments in their own words. Whenever students take an idea, or a passage from another author, they must acknowledge their debt both by using quotation marks where appropriate and by proper referencing such as footnotes or citations. Plagiarism is a major academic offence (see Scholastic Offence Policy in the Western Academic Calendar).

The University of Western Ontario has software for plagiarism checking. Students may be required to submit their work in electronic form for plagiarism checking.

**Conduct:**

Students are expected to arrive at lectures on time, and to conduct themselves during class in a professional and respectful manner that is not disruptive to others.

**Sickness and Other Problems:**

Students should immediately consult with the Undergraduate Services if they have any problems that could affect their performance in the course. Where appropriate, the problems should be documented (see attached). The student should seek advice from Undergraduate Services regarding how best to deal with the problem. Failure to notify Undergraduate Services immediately (or as soon as possible thereafter) will have a negative effect on any appeal.

**Notice:**

Students are responsible for regularly checking their email and notices posted on the dedicated OWL site.
Consultation:
Students are encouraged to discuss problems with their teaching assistant and/or instructor in tutorial sessions. Office hours will be arranged for the students to see the instructor and teaching assistants. Other individual consultation can be arranged by appointment with the appropriate instructor.

Accreditation (AU) Breakdown
Math = 50%
Engineering Science = 30%
Engineering Design = 20%

January 09, 2018/LR