CMPT-439 Numerical Computation
Course Projects

Each course project will cover one of the main topics considered in this course. The following general requirements are common for all projects:

1) Each project is a **team effort** (3-4 students in each team). It shall result in a **software system** designed in any high-level language or Matlab.

2) This software system shall consist of

   a) A **main function** with a **graphical user interface**, which can be used for entering parameters and displaying the results.
   b) **Functions** where numerical methods covered by a certain project are utilized. These functions shall be called from the main program.

3) All team members working on the project shall contribute to the design of the main function.

   a) **The most important requirement is** that **every single team member must incorporate his/her own functions where respective numerical methods are utilized**, in a software system.
   b) Thus, a main function shall be designed with a standardized system of calling and returning parameters for all functions utilizing numerical methods. This shall ensure that every single team member will be able to run a main program with his/her own functions utilizing respective numerical methods.

4) The following results are expected from each team and its members:

   a) **Each team** will make a **10 min oral presentation** with slides showing an architectural structure of a software system designed and with the demonstration of how it works. The source **code of a main function is due the day of a final presentation**.
   b) **Every single team member** shall prepare **a written report** describing his/her contributions to the project (Section 1) and showing the results of test runs for all numerical methods utilized in the project and obtained **using his/her own functions** utilizing the respective numerical methods and incorporated in a software system (Section 2). All **reports are due the day of a final presentation**.
   c) **Every single team member shall turn in sources of his/her functions utilizing the respective numerical methods used in the project**. **Sources are due the day of a final presentation**.

5) **Grading criteria**:

- Functions utilizing numerical methods – **50 points** (<u>if you do not use your own functions utilizing numerical methods and rely on functions designed by your teammate, you will not be able to get more than 20 points for this part of the project</u>.
- A report presenting own contributions and demonstrating how a system works with his/her functions – **20 points**
- A main function with GUI – **15 points** (for each team member)
- Final presentation – **15 points** (for each team member).

Project 1

Solving nonlinear equations

1) A software system for solving nonlinear equations shall be designed.

2) The system shall utilize the following methods:

   a) Bisection
   b) False position
   c) Secant
   d) Newton

3) The user shall be able to

   a) Enter an equation to be solved
   b) Choose a method to be used
   c) Choose a stopping criterion
   d) Enter a threshold parameter for a stopping criterion
   e) Enter a starting approximation or starting brackets or ends of the starting interval.

4) The following output should be created

   a) A root of an equation, which was found (or a message that a root cannot be found)
   b) A true error (the difference between 0 and a left hand side of an equation after a root was plugged there).

Project 2

Solving systems of linear algebraic equations using directed elimination

1) A software system for solving systems of ***n*** linear algebraic equations in ***n*** unknowns shall be designed.

2) The system shall utilize the following methods:

    a) Gaussian directed elimination
    b) Gauss-Jordan directed elimination

3) The user shall be able to

    a) Enter an augmented matrix of the system to be solved or load it from a text file
    b) Choose a method to be used

4) The following output should be created

    a) The roots of a system, which were found (or a message that a root cannot be found – the program shall check whether the matrix of a system is singular; the program also shall warn the user if this matrix is not diagonally dominant).
    b) b) A true mean absolute error (mean absolute difference between biases and left-hand side of all equations after the roots were plugged there).

Project 3

Solving systems of linear algebraic equations using iterative methods

1) A software system for solving systems of ***n*** linear algebraic equations in ***n*** unknowns shall be designed.

2) The system shall utilize the following methods:

   a) Gauss-Seidel iterative method
   b) Jacobi iterative method.

3) The user shall be able to

   a) Enter an augmented matrix of the system to be solved or load it from a text file
   b) Choose a method to be used
   c) Choose a stopping criterion for an iterative method
   d) Enter a threshold parameter for a stopping criterion
   e) Enter a starting approximation for an iterative method or use a starting approximation created by default in the program.

4) The following output should be created

   a) The roots of a system, which were found (or a message that a root cannot be found – the program shall check whether the matrix of a system is singular; the program also shall warn the user if this matrix is not diagonally dominant).
   b) A true mean absolute error (mean absolute difference between biases and left-hand side of all equations after the roots were plugged there).

Project 4

Numerical Differentiation

1) A software system for numerical differentiation shall be designed

2) The system shall utilize the following methods:

   a) Lagrange interpolation for finding missing values of a function (if necessary)
   b) Two point forward difference formula
   c) Three point centered difference formula
   d) Three point forward difference formula

3) The user shall be able to

   a) Enter the known values of a function to be differentiated or load them from a text file
   b) Enter a point where a derivative shall be found
   c) Choose a method to be used

4) The following output should be created: a value of the derivative found

Project 5

Numerical Integration

1) A software system for numerical integration shall be designed

2) The system shall utilize the following methods:

    a) Lagrange interpolation for finding missing values of a function
    b) A function implementing multiple application of the trapezoidal rule
    c) A function implementing the composite Simpson's rule

3) The user shall be able to

    a) Enter the known values of a function to be integrated or load them from a text file
    b) Enter a value of the parameter h and n (the number of intervals)
    c) Choose a method to be used

4) The following output should be created: a value of the integral found

Project 6

Nonlinear Optimization

This project shall be designed in Matlab or Python

1) A software system for finding min and max of a nonlinear function shall be designed.

2) The system shall utilize the following methods:

   a) The Golden Section method
   b) The Newton's method (with an automatic evaluation of a corresponding derivative)

3) The user shall be able to

   a) Enter a function whose min/max shall be found
   b) Enter a starting point (Newton) or the ends of the starting interval (Golden Section)
   c) Choose what shall be found (min or max) in the Golden Section method

4) The following output should be created:

   a) A function shall be plotted to make it possible for the user to choose starting point(s) properly
   b) A min/max point found and a value of the function there.

Project 7

Combination of Projects 2 and 3 (merging them in a single project) – **15 extra credit points**

Project 8

Combination of Projects 4 and 5 (merging them in a single project) – **10 extra credit points**