Computer Methods in Civil Engineering

Instructor  Dr. G. Consolazio, Phone: 846-2220, E-mail: grc@ce.ufl.edu, Web: www.ce.ufl.edu/~grc

Classes  Lectures: MWF 3\textsuperscript{rd} Period (Tur. 2328), Lab: Mon. 10-11\textsuperscript{th} Periods (Weil 320 & computer labs)

Office Hrs  Weil Hall – Room 204-D, MTWF 4\textsuperscript{th} Period and by arrangement.

Web Site  The course web site is located at: www.ce.ufl.edu/~grc. Announcements, assignments, example Matlab code, and example MathCad worksheets will be posted to the web site regularly.


Software  Matlab programming environment, MathCad mathematical worksheet software

Goals  The goals of this course are to teach universal procedural programming skills, to teach problem decomposition algorithm development, and debugging, to teach numerical methods and their practical implementation for engineering problems, and to demonstrate the practical application of computers as an engineering tool. The material will be presented assuming the student has no prior programming experience. The primary vehicle for teaching programming skills will be the Matlab programming environment, however the skills learned will be readily adaptable to other languages that the student may work with in future. In addition, the use of the MathCad mathematical worksheet software will be covered. The first segment of the course will cover programming in Matlab and MathCad while the second segment will cover numerical methods and their implementation using Matlab and MathCad.

Objectives  The student is expected to gain proficiency in the process of developing computer programs and worksheets and implementing numerical methods for the purpose of solving practical engineering problems. This will be accomplished by developing several skills including:

1) Understanding universal procedural programming concepts (variables, control structures, modular programming, algorithm development).
2) Decomposing a given problem into a series of bottom level tasks.
3) Coding each bottom level task using programming modules.
4) Assembling these modules into an overall program/worksheet framework.
5) Debugging syntax errors and program logic errors.
6) Validating program correctness through simple examples confirmed by hand.
7) Understanding the theoretical concepts behind various numerical methods.
8) Implementing numerical methods in order to solve practical engineering problems.
9) Clearly communicating your procedures and results in written reports.

Outcomes  Successful completion of this course will provide the student with two essential engineering tools: computer programming and a basic knowledge of numerical methods. The course requires the student to apply math, science and engineering principles to solve practical engineering problems. Students completing the course will be able to identify, decompose, and solve complex problems and use modern engineering tools in the process. Students will be able to present their work in written report format and will be able to perform self-assessment.
Participation
The participation of students during class is expected. Students are encouraged to ask questions and contribute to discussions during class.

Grading
The grading scale for this course is as follows.

<table>
<thead>
<tr>
<th>Grade</th>
<th>A</th>
<th>B+</th>
<th>B</th>
<th>C+</th>
<th>C</th>
<th>D</th>
<th>D+</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Points</td>
<td>100-92</td>
<td>92-87</td>
<td>87-80</td>
<td>80-77</td>
<td>77-70</td>
<td>70-67</td>
<td>67-60</td>
<td>60-0</td>
</tr>
</tbody>
</table>

2-Credits
See the instructor immediately if you are registered for this course as a 2-credit course.

Grade Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework</td>
<td>17.5 %</td>
</tr>
<tr>
<td>Portfolio</td>
<td>7.5 %</td>
</tr>
<tr>
<td>Quizzes</td>
<td>10 %</td>
</tr>
<tr>
<td>Exam 1 – Programming</td>
<td>15 %</td>
</tr>
<tr>
<td>Exam 2 – Num. Methods &amp; Programming</td>
<td>15 %</td>
</tr>
<tr>
<td>Exam 3 – Num. Methods &amp; Programming</td>
<td>15 %</td>
</tr>
<tr>
<td>Project</td>
<td>20 %</td>
</tr>
</tbody>
</table>

Quizzes
Approximately two or three short announced quizzes will be given during the semester.

Homework
Homework will be assigned on a regular basis and will be due at the beginning of class on the designated date. Homework will generally consist of solving problems using Matlab programming or using MathCad spreadsheets. There is a strict format for homework submissions. Failure to submit homework in the format indicated below will result in the assignment being returned without a grade.

1) The first page shall consist of a typed cover sheet. This cover sheet shall include your name, the course name, the assignment title, and a one to two paragraph assignment summary. The summary shall include a description of the problem, an overview of the solution strategy used, methods employed, and any appropriate observations regarding the assignment. This sheet must not exceed one page in length.

2) The entire assignment must be submitted on 8½ x 11 inch paper. All computer input, output, listings, and plots shall be printed on 8½ x 11 inch paper. The entire homework package shall be stapled together.

3) All data plots must be computer generated plots (no hand plots). Plots generated in Matlab must include the full name of the student at the top of each plot (see the Matlab ‘title’ command).

4) An outline (pseudocode) of Matlab programs and complex MathCad worksheets shall be included.

5) You must submit the program listing you used (.m files or spreadsheet printouts). The source code (listing) or spreadsheet must contain comments explaining what the program is doing. The top of all function m-files shall have a standard header block (an example will be given in class).

6) Input and output must be included. Labels must be included in all output and must be computer generated (do not hand-write the labels on printed output).

7) Input data read into Matlab programs shall be clearly printed to the output data written by the program. This “echoing” of data is always necessary so that the user can verify that your program correctly read the data from the input source.

8) Validation of the output is required to show that your program finds the correct answer. This can be done either by hand, using some higher level programming options that we will discuss, or using an independent software package.
Academic Honesty

Students are encouraged to work together on the homework assignments and to study in groups to help each other learn the material. HOWEVER, students are expected to submit only work that represents their own effort. Any two assignments submitted with sufficient resemblance will be returned without a grade. Consider this carefully before giving your programs or spreadsheets to another student to “help them out”. Any cases of cheating on homeworks or exams will have severe consequences. Cheating on a homework would include obtaining another student’s work and submitting it as your own without their knowledge.

Fair Warning : Simply doing a search and replace on variable names in Matlab code or MathCad worksheets will not be sufficient to hide the fact that two assignments are essentially one in the same. Programming style tends to be very individualistic; similarities in style and formatting will always be apparent even if superficial changes are made.

Late Policy

Late homework will be accepted, with a penalty of 30%, at the beginning of the class that follows the original due date for the assignment. After that time, no late homework will be accepted. Exceptions to this rule include cases where the student has spoken to the instructor prior to the due date of the homework or cases where there is a valid excuse (e.g. medical emergency with written proof).

Portfolio

Students will keep a portfolio of their work as part of this course. Portfolios will be collected, evaluated, and returned several times during the semester. Portfolios will be kept in a 3-ring binder, will be clearly organized, and will contain the following sections:

1. Table of contents
   - Typed table of contents
   - Include page numbers or easily accessible section numbers
2. Topic summaries
   - Typed topic summaries
   - Summarize the most important points for the topic
   - Points that need special attention regarding the topic
   - Figures, either computer drawn or neatly hand drawn using a straight edge, should be included in the summaries wherever appropriate.
3. Homework
   - Copies of the cover sheets from every homework assignment (only the cover sheets!).
   - Copies of your best Matlab and/or MathCad assignments. The number of assignments that can be included will be specified before the portfolios are submitted. However, you should choose the assignments that you feel best demonstrate the skills you have learned during the class.
4. Quizzes
   - Your original graded quizzes.
5. Self-Assessments
   - Periodic self-evaluations of your strengths and weaknesses

The portfolio will serve several purposes. It will serve as a mechanism by which the student demonstrates proficiency in the various areas covered by the course. It will serve as an assessment method by which each student can assess their progress during the course and can assess the overall utility and relevance of the material covered in the course. It will also serve as a representative collection of the student’s abilities in a packaged format that can be easily reviewed by potential employers during job interviews. In summary, the portfolio is intended to be as useful to the student as it is to the instructor.