Lecture #0

Rules and Procedures

Outline

• The Textbook
• Grading
• Homework
• Exams
• MATLAB Codes
• Final Project
The Textbook

There is no official textbook for the class because no such textbook exists.

The “textbook” is therefore the course website.

http://emlab.utep.edu/ee5390cem.htm

Grading

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
<th>Grade Range</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework</td>
<td>40%</td>
<td>90% – 100% → A</td>
<td></td>
</tr>
<tr>
<td>Midterm Exam #1</td>
<td>15%</td>
<td>80% – 89% → B</td>
<td></td>
</tr>
<tr>
<td>Midterm Exam #2</td>
<td>15%</td>
<td>70% – 79% → C</td>
<td></td>
</tr>
<tr>
<td>Final Project</td>
<td>20%</td>
<td>60% – 69% → D</td>
<td></td>
</tr>
<tr>
<td>Participation</td>
<td>10%</td>
<td>0% – 59% → F</td>
<td></td>
</tr>
</tbody>
</table>

HOMEWORK IS 40% OF YOUR FINAL GRADE!!!!!!!
Homework Rules

• Assigned on a weekly basis.
• Homework is very cumulative. It is not an option to miss a homework.
• Late Homework
  – -10% every day late.
  – Grade of zero after three days.
  – I need to distribute solutions as soon as possible.
• Homework is 40% of your final grade. The homework IS this class.
• Do your own work. Do not copy from other students.

Homework Format

• Must have a cover page.
  – Name, course information, assignment #, date, etc.
• Put problems in the proper order.
• Be neat and well organized.
• Providing computer codes is optional.
• **ALL CODES MUST GO INTO AN APPENDIX!**
• Construct homework as if you will need to relearn the material 10 years from now and have only your notes and homework.
• Stapled at upper-left corner with no additional binding.
Exams

• All exams are take-home.
• Exams follow the exact same format and rules as the homework.
• Cannot provide help on an exam.

Extra Credit?

No additional assignments will be given in this class for extra credit.

Extra credit is given in the following circumstances:

• You catch a mistake in the course materials.
• Your assignments go above and beyond what is asked.
Rules For Your MATLAB Codes

• You must use MATLAB for all homework and exams.
• Programs must follow the block diagrams in the class exactly.
• Codes must be neat, well organized, and well commented.
• Unless otherwise instructed, code must be a single program and NOT broken into separate functions.
• Try to use the same variable names as the notes and the instructor.
• Need help? If you are stuck and your codes follow ALL of the above rules, e-mail me your MATLAB code.
  – rcrumpf@utep.edu
  – Cannot provide help on exams.

Structure of the Ideal Code

Initialize MATLAB
- close all unnecessary windows
- clear memory
- open a figure window
- define units and constants

Dashboard
- Define what is to be learned
- Define source parameters
- Define device parameters
- Define method specific parameters

Rest of Code

Save/Show Results

Only numbers. No calculations!

Only calculations. No numbers!
The Final Project

• Purpose – to learn, practice, and share a topic outside of what was taught in class.
• Project should be summarized in Power Point.
  – Must be complete enough that another student from the class can reproduce your work if needed.
• Final Project = Final Exam
• Projects presented during the final exam period.
• May work alone or in teams, but teams must do proportionally more work.
• Must submit all electronic files to course instructor.
• No late projects will be accepted.
  • Get started early!!

Project Ideas

• Study a new device
  – Photonic crystals
  – Bragg gratings
  – Guided-mode resonance filters
  – Negative index metamaterials
  – Find band extreme away from key points of symmetry
  – Solve Shrödinger’s equation
  – Angle of reflection not equal to angle of incidence
• Implement a new method
  – Method of moments
  – Solve another differential equation (i.e. heat equation, etc.)
  – Finite element method
  – Waveguide analysis
  – Beam propagation method
  – Method of lines
  – Fourier-Space FDFD
  – Compare various methods on same device
  – Hybridize ROWA and MOL
• Add a feature to a method
  – FDFD for oblique coordinate system
  – Fix Gibbs’s phenomenon
  – Dielectric smoothing
  – Fast Fourier factorization
  – Incorporate model into optimization
  – Smart parameter sweep
  – Iterative solver (w/ fast seeded sweep)
  – Nonuniform or unstructured grid
  – Different language (Python, C, Fortran, etc.)
  – R/H/ETM matrices
  – SC-PML
  – Iterative FDFD
  – Parallelize a method
  – Optimize PML parameters
  – Higher-order accurate derivatives
  – Other boundary conditions
  – Analog layers in semi-analytical methods
  – Handle fully anisotropic materials
  – Calculate isofrequency contours
  – More efficient convmat()
  – Efficient convmat() for non-orthorhombic symmetries
  – Model a two-period device