CEE6501 — Lecture 7.3

Miscellaneous Extra Topics

Agenda

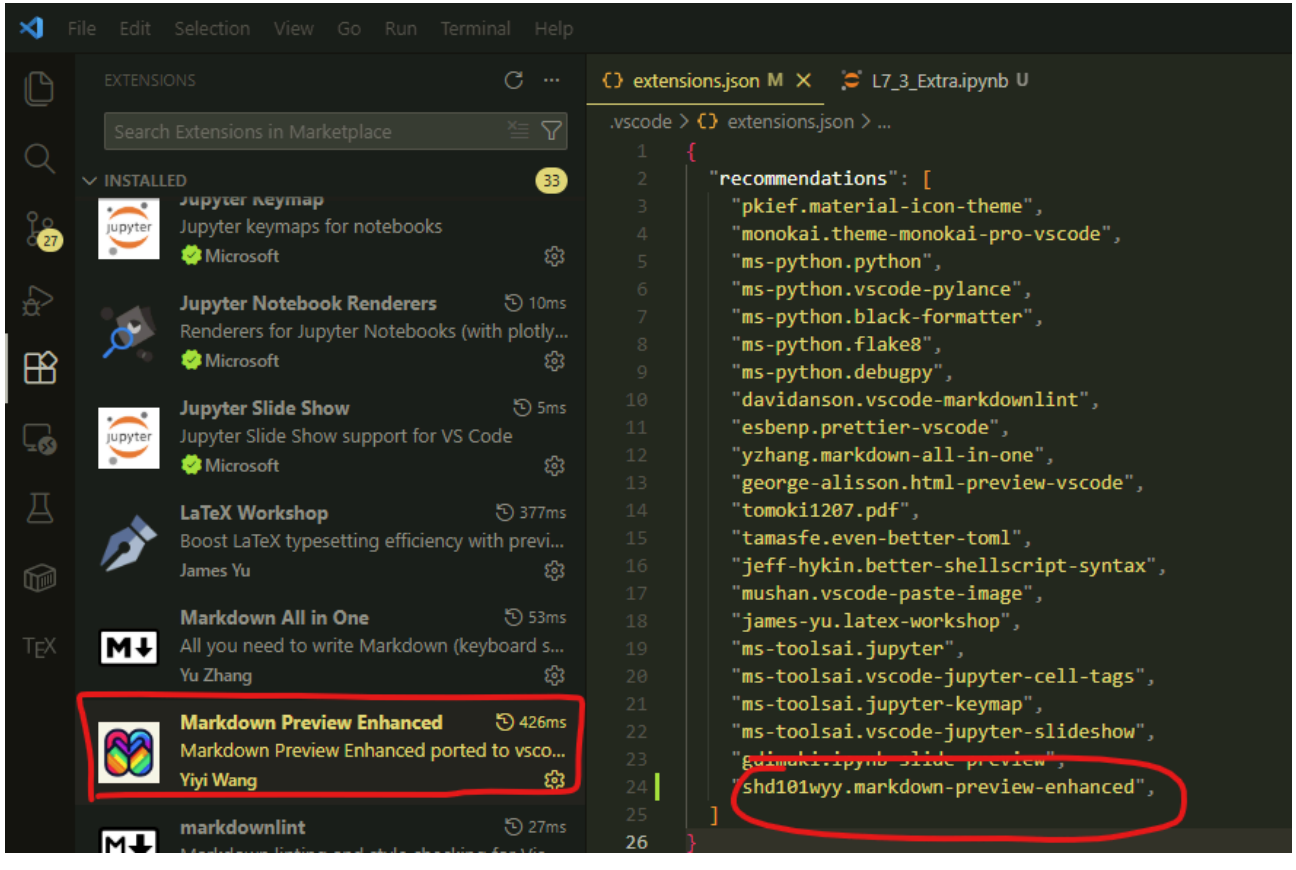
Part 1 — Discuss Week 7 Homework
Part 2 - Adding VS Code Extension
Part 3 - Midterm Results Overview
Part 4 - Discuss Feedback Quiz

Part 1 - Discuss Week 7 Homework

Part 2 - Markdown Preview Enhanced

Install VS-code Extension

Markdown Preview Enhanced has been added to the `extensions.json` file



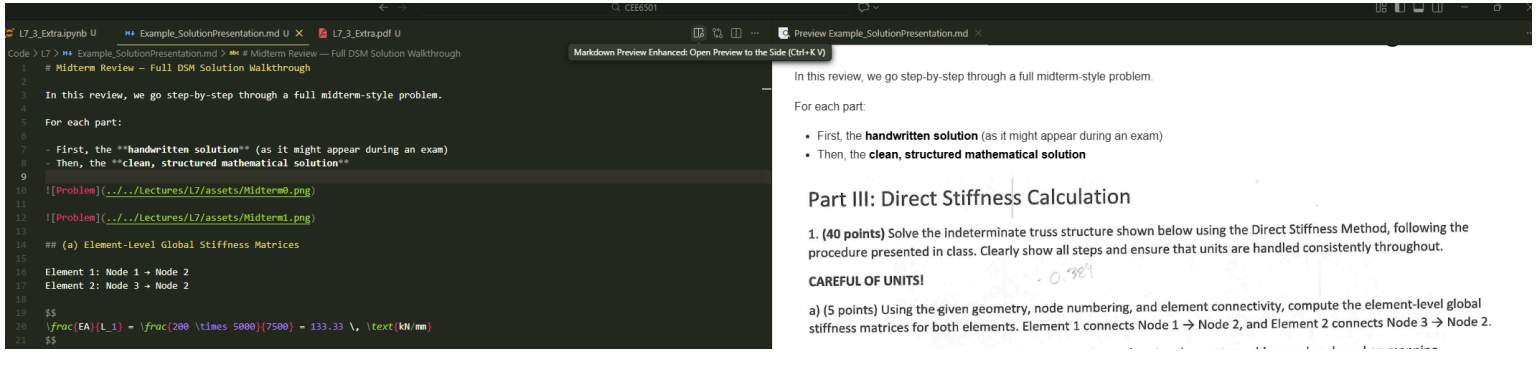
Open Enhanced Preview

Open a markdown file. For example, the `Example_SolutionPresentation.md` file in the `Code\L7` Folder.

Click the preview button top right

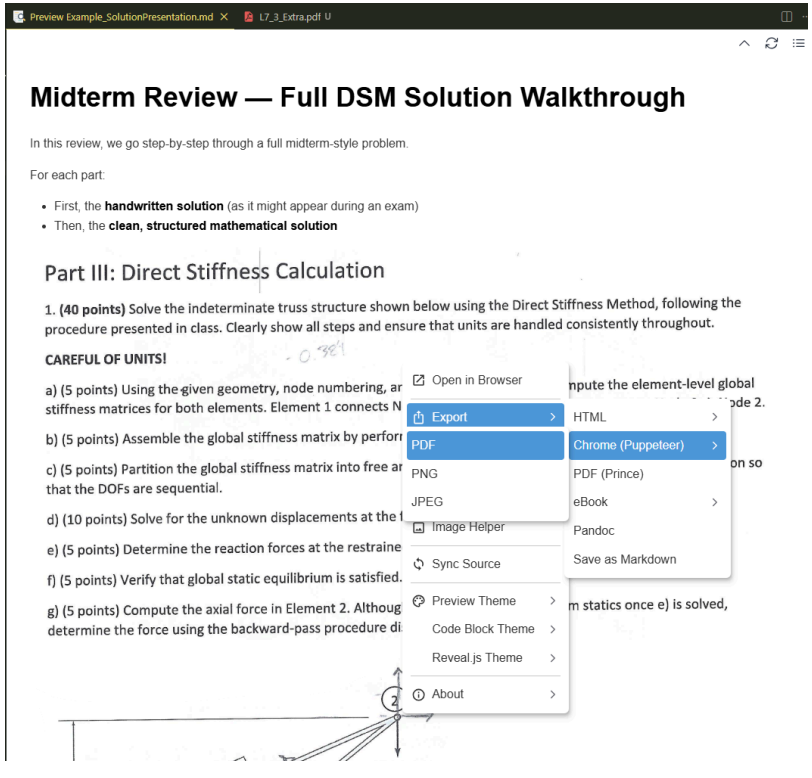
or

```
Ctrl + Shift + P
Markdown Preview Enhanced: Open Preview
```



Export to PDF

right click -> Export -> Chrome (Puppeteer) -> PDF



Find Exported PDF

The PDF should be saved in the same location as the .md file

Part 3 - Midterm Results Review

Summary

- **Highest score:** 87
- **Lowest score:** 41
- **Class average:** 67

After reviewing the distribution and overall difficulty of the exam:

> ***All students received +15 points added to their midterm score.***

This adjustment better aligns the results with course expectations.

Review the DSM Question

- Full midterm solution posted on canvas.
- Today Go through the DSM problem step by step
- Please carefully review:
    - Element stiffness formulation
    - DOF ordering and mapping
    - Assembly into the global system
    - Application of boundary conditions
    - Solving the reduced system

Part 4 - Discuss Feedback Quiz

Q1: What aspects of the course have helped you learn the material most effectively? What do you like about the course?

- Strong integration of **theory and coding**, with examples and Jupyter Notebooks making abstract concepts more practical and clear.
- **Assignments and embedded practice code** effectively reinforce material and improve coding confidence.
- **Well-structured slides and logical course organization** help connect fundamentals to real-world engineering applications.

Q2: Is the mix of slides and code lecture format working for you? why or why not?

- The mix of **slides and live coding** is working well and makes computational concepts easier to understand and more engaging.
- Students appreciate learning coding alongside theory and value the **interactive Jupyter format**.
- Suggestions include slightly fewer slides and **more in-class time for hands-on coding practice**, especially with small examples.

Q3: Do you want there to be more or less python during lectures?

- There is interest in **slightly more Python during lectures**, particularly focused on challenging topics and common mistakes.
- The current balance between theory and coding is generally working well, especially with assignments reinforcing Python skills.
- Some preference remains for emphasizing core concepts during lecture and allowing coding practice to happen independently.

Q4: Are there any specific topics or concepts that you are struggling with?

- Some difficulty arose initially with distinguishing **local vs. global coordinates** and **element- vs. joint-level stiffness matrices**.
- Coding was initially a challenge for some, but comfort with Python is increasing over time.
- Several indicated no major conceptual struggles so far, noting that working through full examples—especially by hand once—helps clarify the overall workflow.

Q5: How could the lectures or instructional methods be improved to support your learning?

- The pace can feel dense in a single sitting;slightly slowing key sections could help.
- Adding **more in-depth, fully worked numerical examples**—especially ones similar to exam-style problems—would strengthen conceptual connections and exam preparation.
- Incorporating **more interactive, in-class problem-solving time** (e.g., working through a full example together or testing code live) would improve engagement and understanding.

Q6: What do you find most challenging about the assignments in this course?

- The **Python implementation** is the primary challenge, particularly translating clear theoretical steps into structured, logical code.
- The **time commitment and overall assignment length** can be significant, especially for coding-heavy tasks.
- The level of coding required can be demanding, particularly for those with limited prior programming experience.

Q7: How do you feel about the pace of the course?

- The overall pace is generally viewed as appropriate and well-balanced.
- Lectures can feel fast-paced at times, particularly given the volume of material covered.
- The pace may require additional effort outside of class, especially when combined with multiple assignment types.

Q8: How do you feel about the organization of the course and lectures? Are things clear to you? Why or why not?

- The course and lectures are viewed as **very well organized and clearly structured**, building step by step from fundamental concepts.
- Expectations are clear, including exam preparation, and the progression from basics to more advanced topics supports understanding.
- Assignments deepen comprehension, though there is some concern about overall workload and a desire for more numerical examples within lectures.

Q9: What additional resources or support would help you succeed in this class?

- Additional **Python coding tips and guidance** would be helpful, particularly for strengthening programming skills.
- **Practice exams, sample questions, and assignments aligned with exam style** would improve preparation and clarify expectations.
- A few would appreciate more examples connecting course material to **industry or real-world applications**, while many feel current resources are sufficient.

Q10: How much are you using AI tools to complete assignments?

- AI is primarily used for **debugging, troubleshooting errors, syntax help, and formatting**, often after attempting the work independently.
- Some use AI for **code suggestions, understanding specific lines of code, structuring solutions, plotting, or improving presentation**, while still ensuring they understand the logic.
- A few report moderate use (around ~50%) for coding support, particularly when lacking a strong programming background.
- Some complete assignments largely independently, using AI mainly for presentation or minor syntax corrections.
- There is general awareness that AI can be helpful but must be **carefully checked**, as it can produce incorrect assumptions or unit errors.