Information Technology
Lecture 1: Introduction to the course

Roman Putanowicz

Chair for Computational Engineering

6 marca 2020

KTIwI

In brief

- aim and scope
- organisation of the course
- motivation
- requirements
- overview of software tools
- sample problems

Aim and scope

The course will enable students to comprehend the concepts and practice of Computer Science, the main focus is however put on developing programming skills. The main environment to exercise programming concepts Matlab. The minor goal is to give the student wider perspective on operating systems and software tools for data processing, visualization, geometric modelling. The student should also develop appreciation of the implications of computer use in solving engineering and scientific problems.

Assessment method

Course grading

- The computer labs grade is the same time the course grade.

Computer lab grading rules

- Students prepare to computer lab in their own study time.
- It is allowed to miss only one computer lab.
- The grade for computer labs is based on student's engagement, activity, progress and the results of a test written during the last lab.
- Students who failed to get passing grade for computer labs have one additional resit term for the test. The data for the resit term will be depending on the circumstances (generally one date for all students).

Lectures' topics 1-5

- Lecture 1: Introduction to the course. Aim and scope. Method of working. Motivation. Requirements. Overview of software tools. Sample problems.
- Lecture 2: Introduction to programming. Processor and program. High level programming languages. Compilation versus interpretation. Overview of popular languages. Introduction to programming in Matlab/Octave.
- Lecture 3: Diving in Matlab/Octave programming Pseudocode. Variables and assignment operator. Matrix variables. Conditional statement. Loops. Functions. Programming paradigms: structural object oriented, functional.
- Lecture 4: Data structures and algorithms Basic data structures: arrays, lists, sets, graphs. Overview of selected algorithms for sequences. Sorting and searching. Matrix algorithms.
- Lecture 5: Data formats and data processing Binary code. Data types. Characters, ASCII, code pages, UTF-8. Data formats. HDF5, JSON, XML, CSV. Data visualization. ParaView, Matplotlib, plotting in Matlab and Octave.

Lectures' topics 6-8

- Lecture 6: Introduction to geometric modelling. Geometric models and their representations. B-Rep i CSG. Meshes as data structures. Basic geometric transformations. Selected tools: GMSH, OpenScad, Blender.
- Lecture 7: Overview of software engineering Repositories. Git, GitHub. Documenting code. Markdown. Linux environment. Automating repetitive tasks. Software licences.
- Lecture 8: Summary - the role of programming in engineering curriculum.

Computer labs' topics

1. Introduction to Matlab/Octave. IDE. Running scripts.
2. Assignment statement. Functions
3. Conditional statement. Functions (cont.)
4. Enumeration loops: `for`
5. Conditional loops: `while`
6. More complex programs. Exercises.
7. Final test.

Reading list

Primary readings

- Matlab(R) Primer, MathWorks, 2019 `https://www.mathworks.com/help/releases/R2019b/pdf_doc/matlab/getstart.pdf`
- David Houcque, Introduction to Matlab for Engineering Students, Northwestern University, 2005, `https://www.mccormick.northwestern.edu/documents/students/undergraduate/introduction-to-matlab.pdf`
- Tony Gaddis, Starting out with Python, Pearson, 2017 (4th edition)

Additional readings

- W. Gander, Learning MATLAB. A problem Solving Approach, Springer, 2015
- K.D Lee, Python Programming Fundamentals, Springer, 2011
- J. D. Foley et al, Computer Graphics: Principles and Practice, Addison Wesley; 3rd Edition, 2013
- D. M. Bourg, Physics for Game Developers, O'Reilly, 2001

The way we work

Required background knowledge:

- Vector and matrix algebra. Planimetry. Elementary functions. Kinematics and dynamics of material points.

Lectures:

- The scope of lectures is only sketched and generally is open. If there are some topics you would like hear about or if you have some specific questions you would like me to discuss during the lecture then pleas let me know and I will address them in the lectures.

Computer labs:

- The main purpose of computer labs is to make possible to ask very specific questions about your programs and problems you encounter. During the labs you have an opportunity to show your **skills and engagement**. The main mode of working is **discussion**. Reaching the goals of the course require **INTENSE and SYSTEMATIC** work in your self-study time.

Grading rule

		Engagement		
		small	medium	big
Effects	small	2.0	3	3.5
	medium	3.5	3.5	4
	big	4	4.5	5

- Generally I distinguish just two groups of students: the one who want to learn something and the others. Everything else are just less important details.
- The accuracy and objectivity of grading depends in fact on how much feedback on your engagement and achieved results you provide.
- **All the time it is possible to discuss the marks.**

Software tools : basic environment

Operating system:

- Windows

or

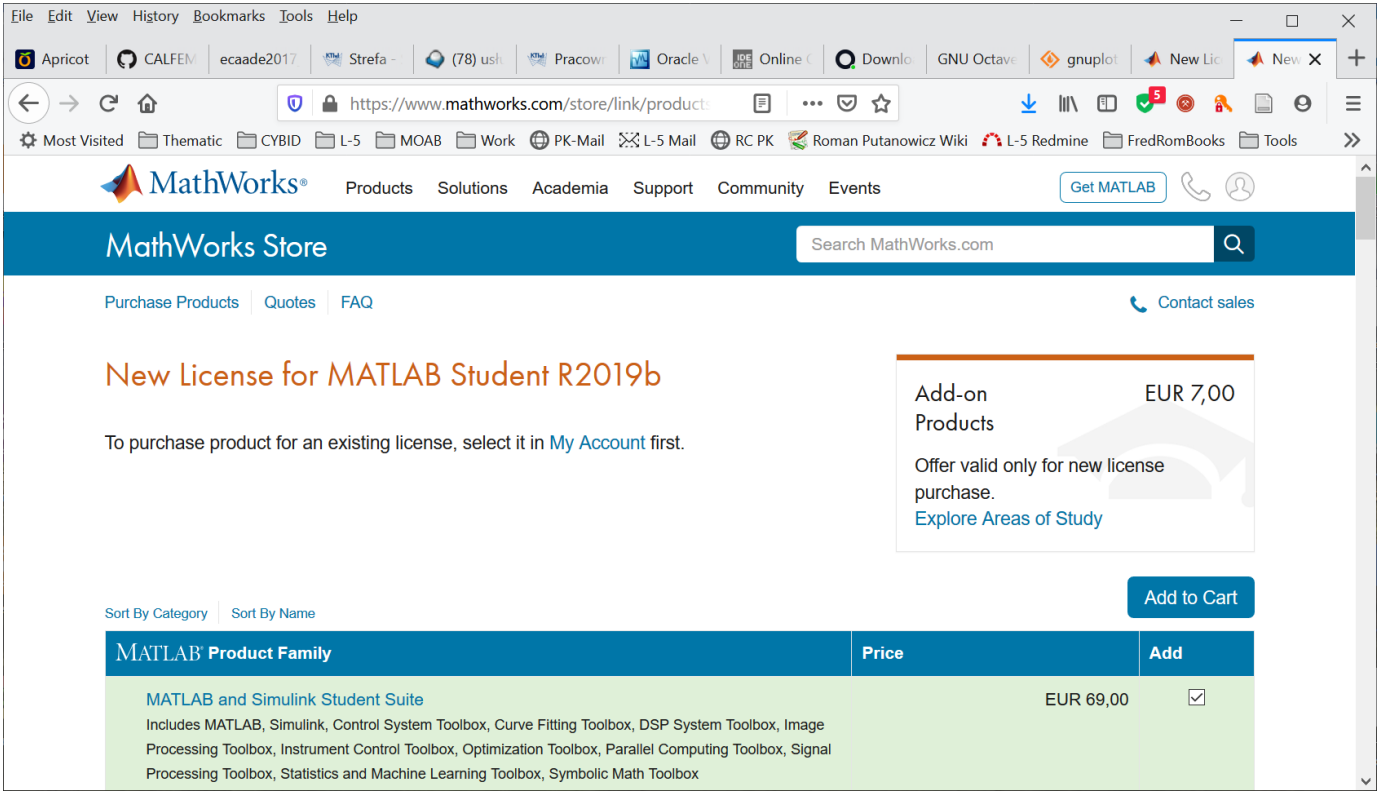
- Linux

Suggested solution:

- Windows + Linux via VirtualBox (`https://www.virtualbox.org/`)

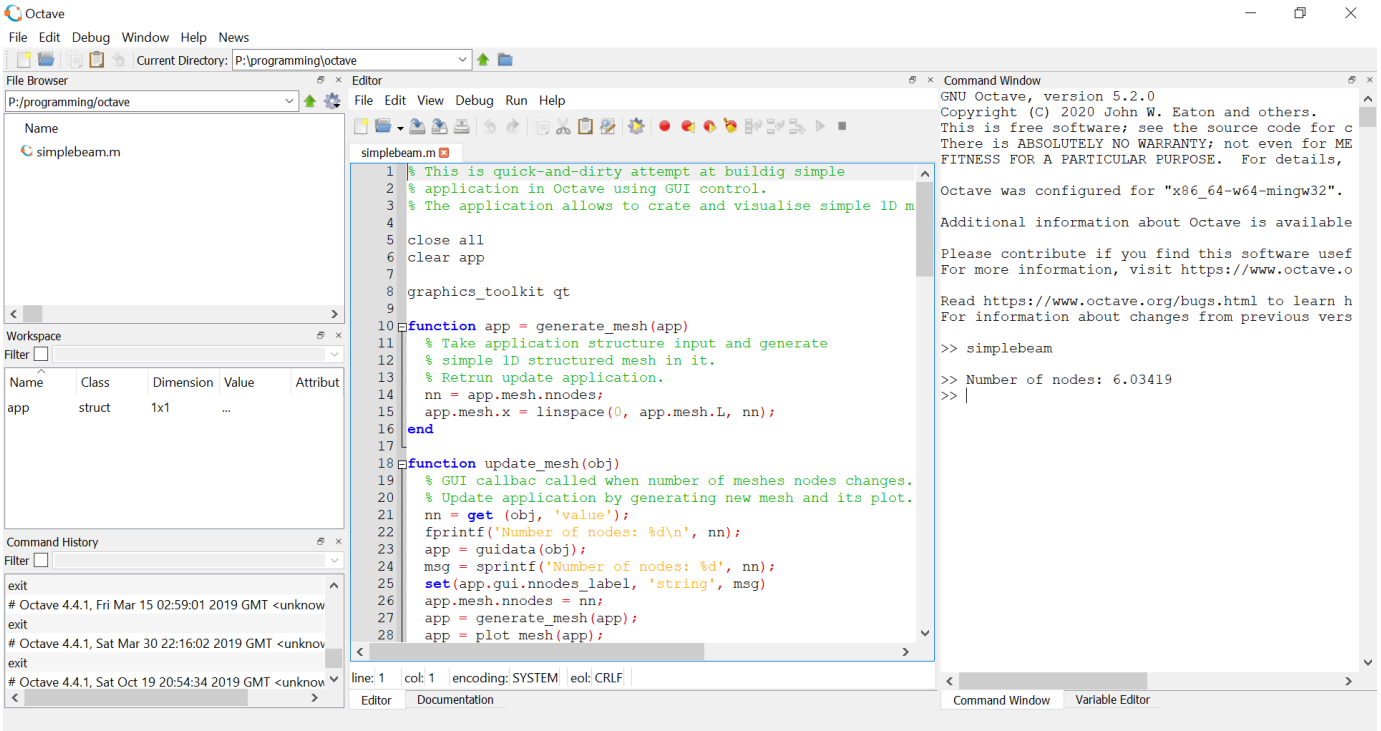
Absolute minimum: interpreter Octave/Matlab + editor

Matlab



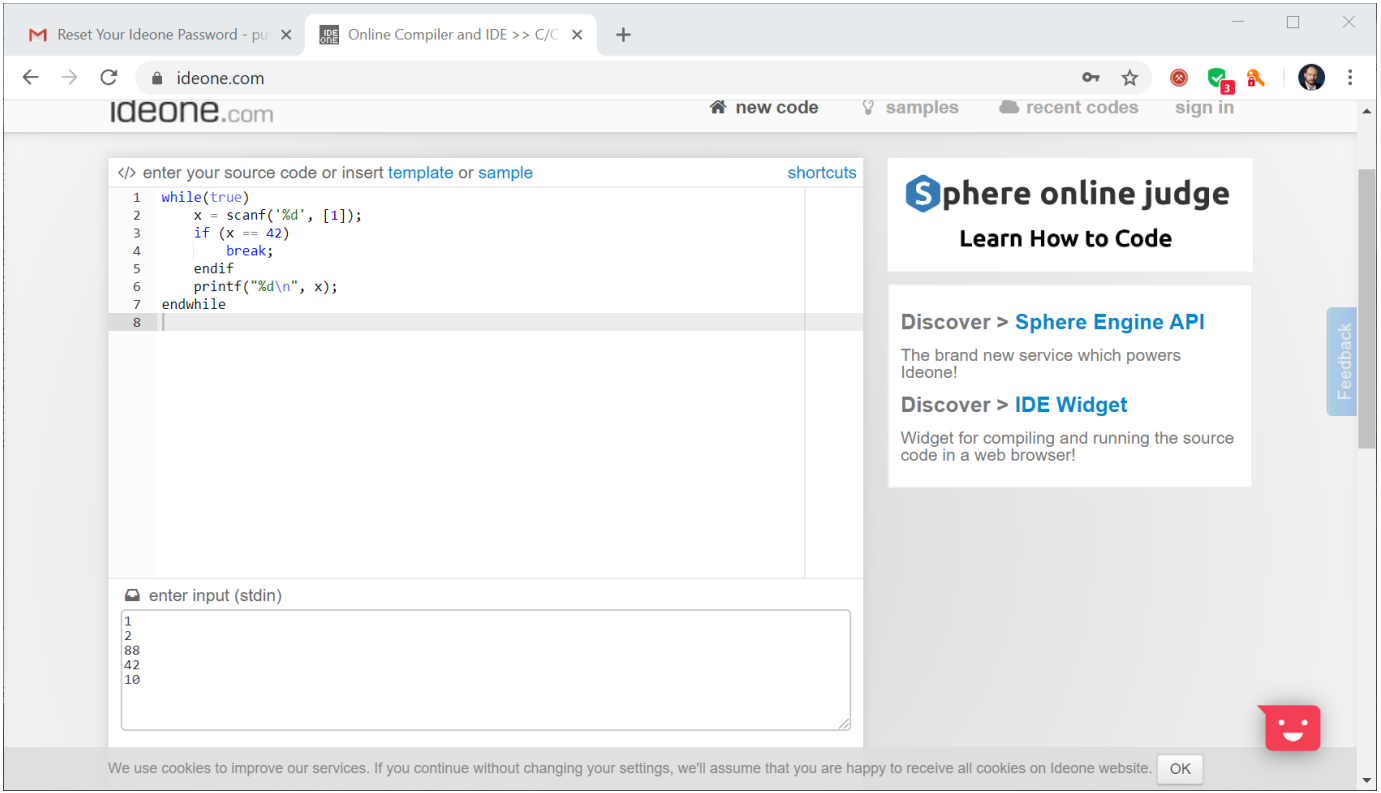
https://www.mathworks.com/products/matlab.html

Octave



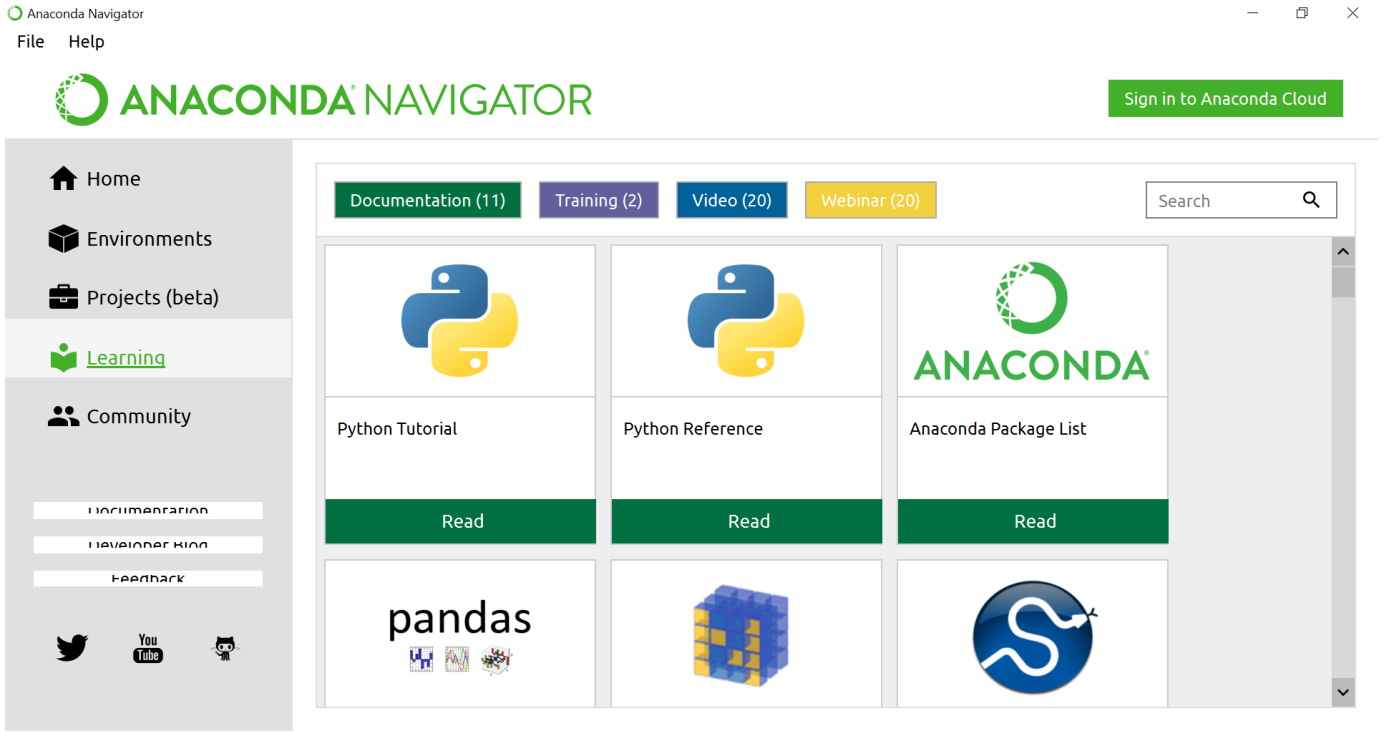
https://www.gnu.org/software/octave/

Ideone.com



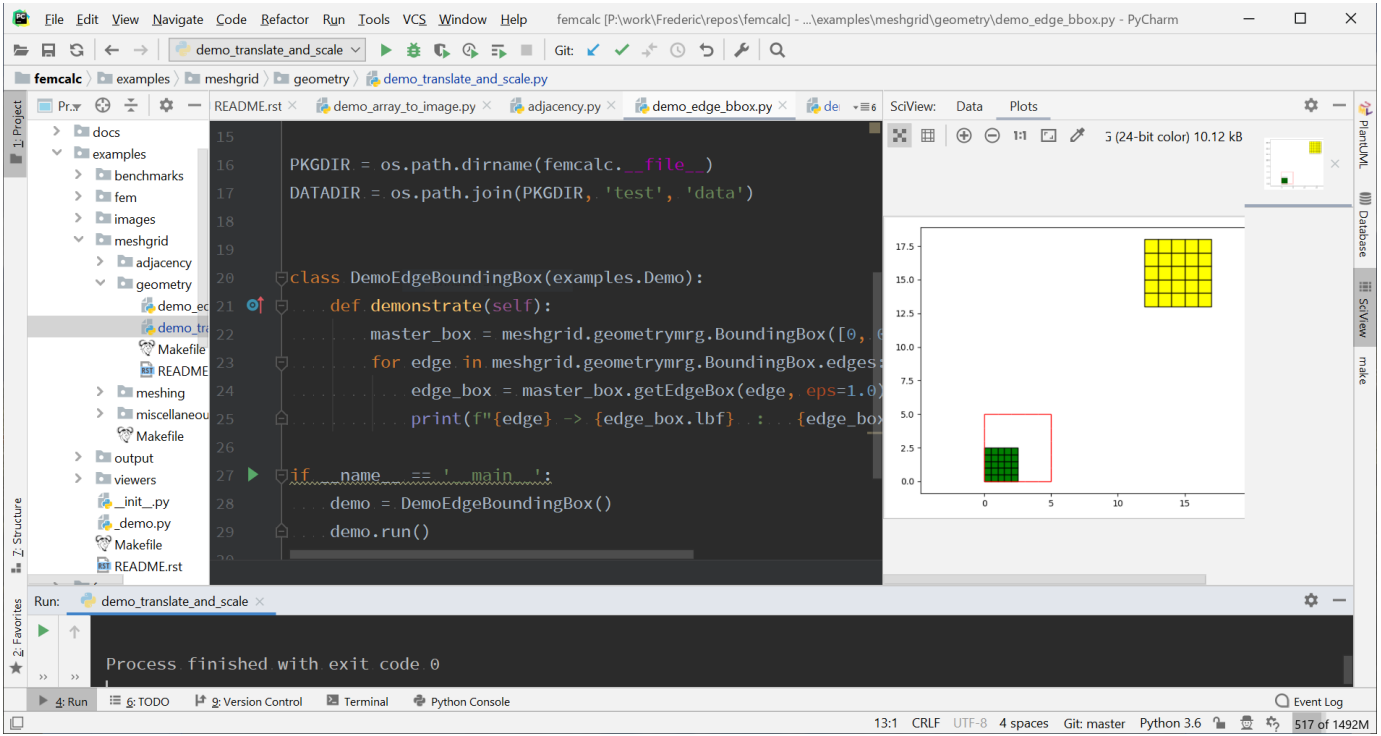
https://ideone.com

Python 3 via Anaconda



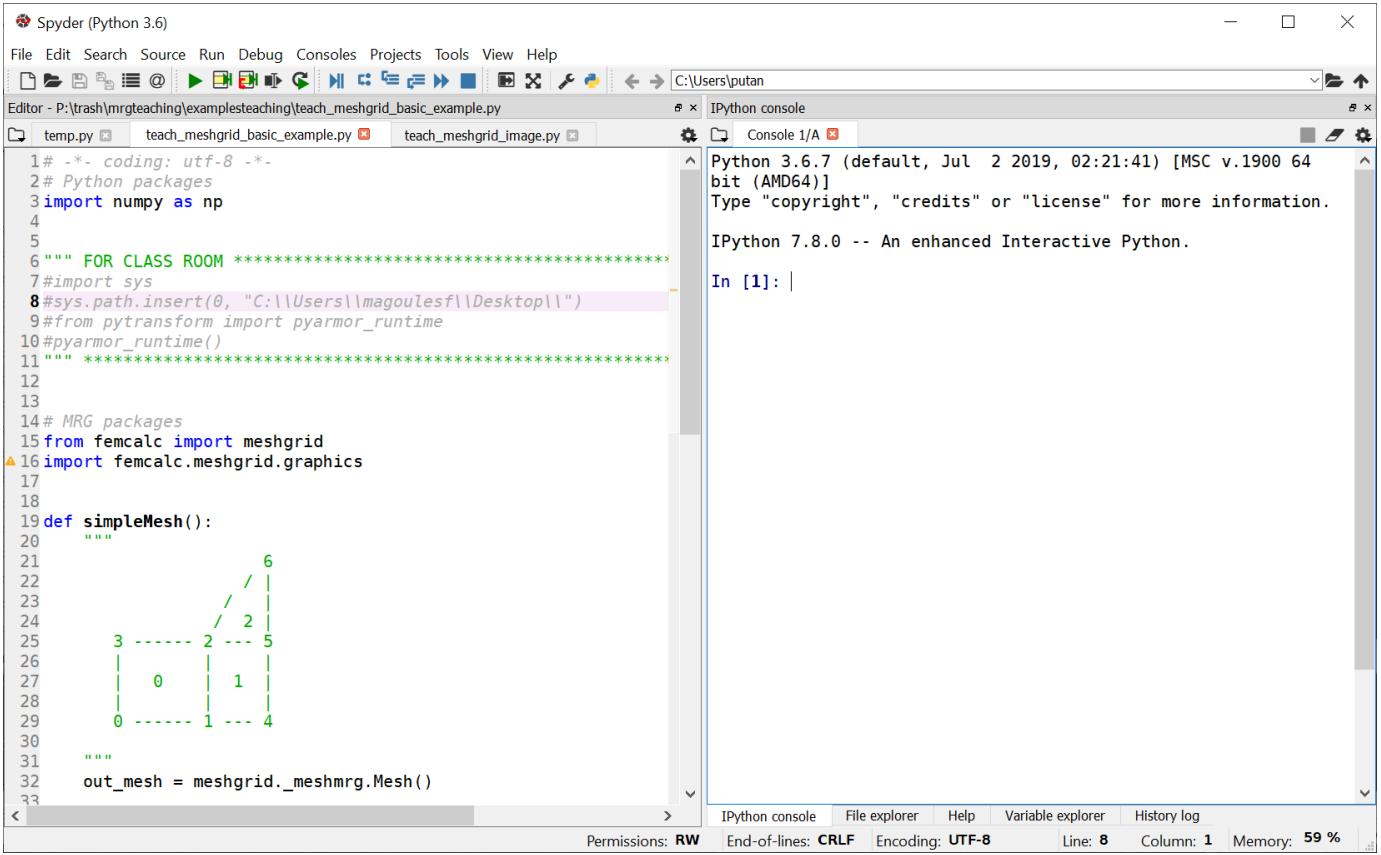
https://www.anaconda.com/

PyCharm



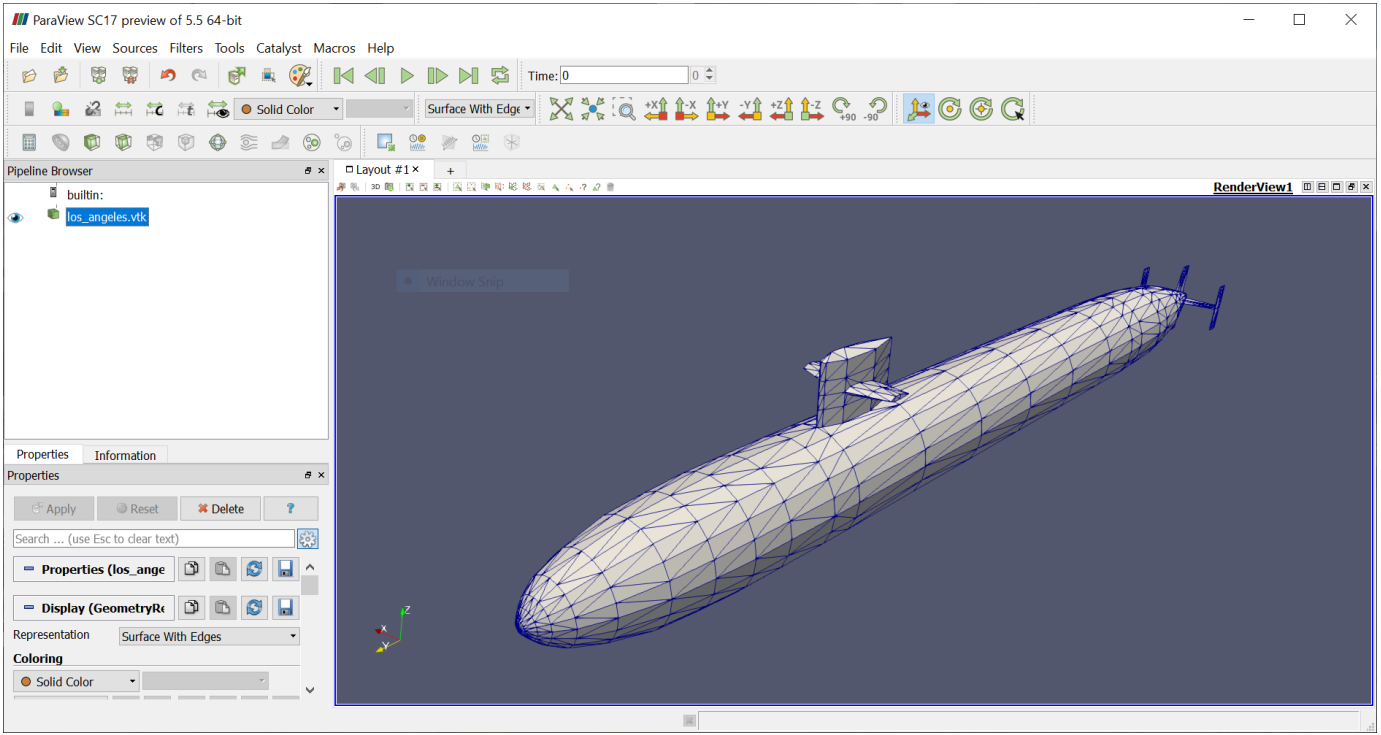
https://www.jetbrains.com/pycharm/

Spyder



https://www.spyder-ide.org/

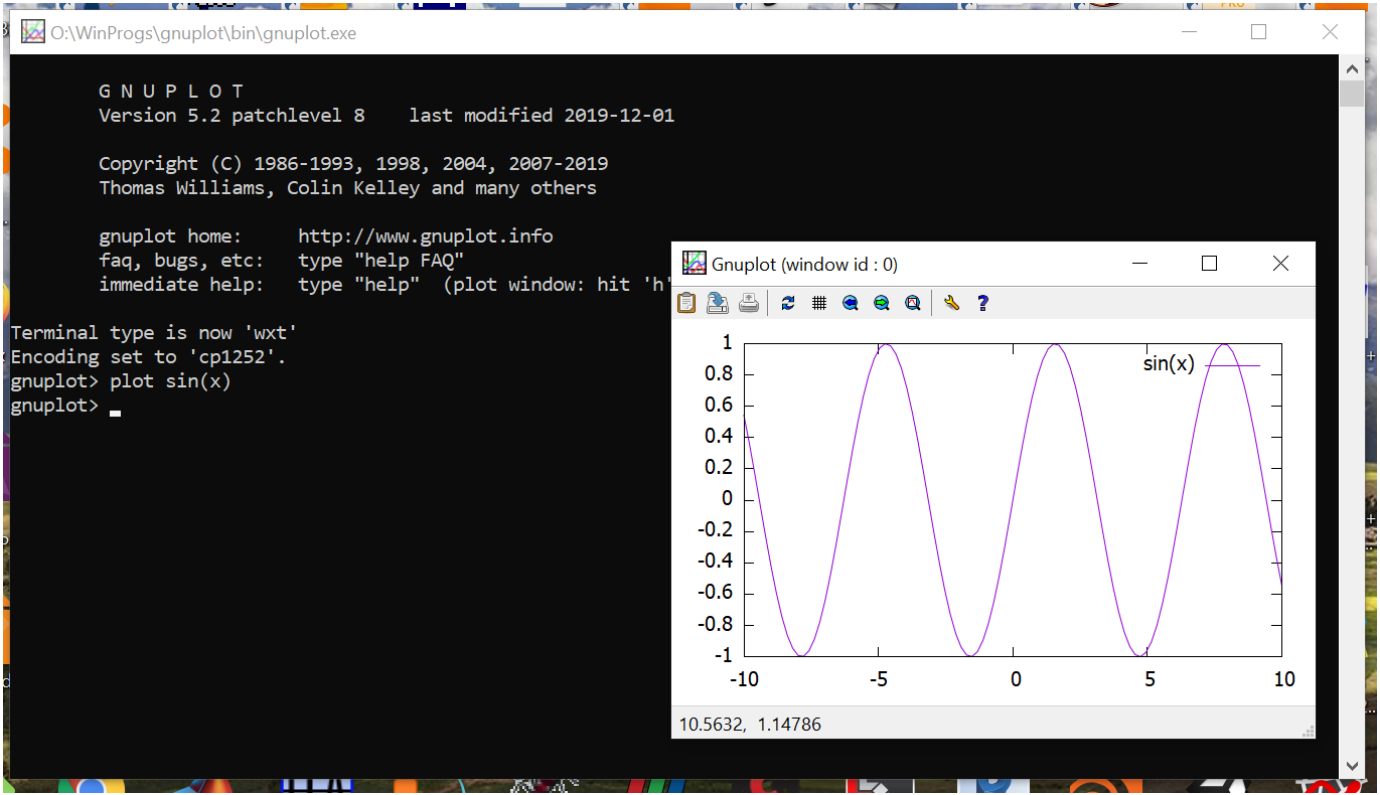
ParaView



`https://www.paraview.org/`

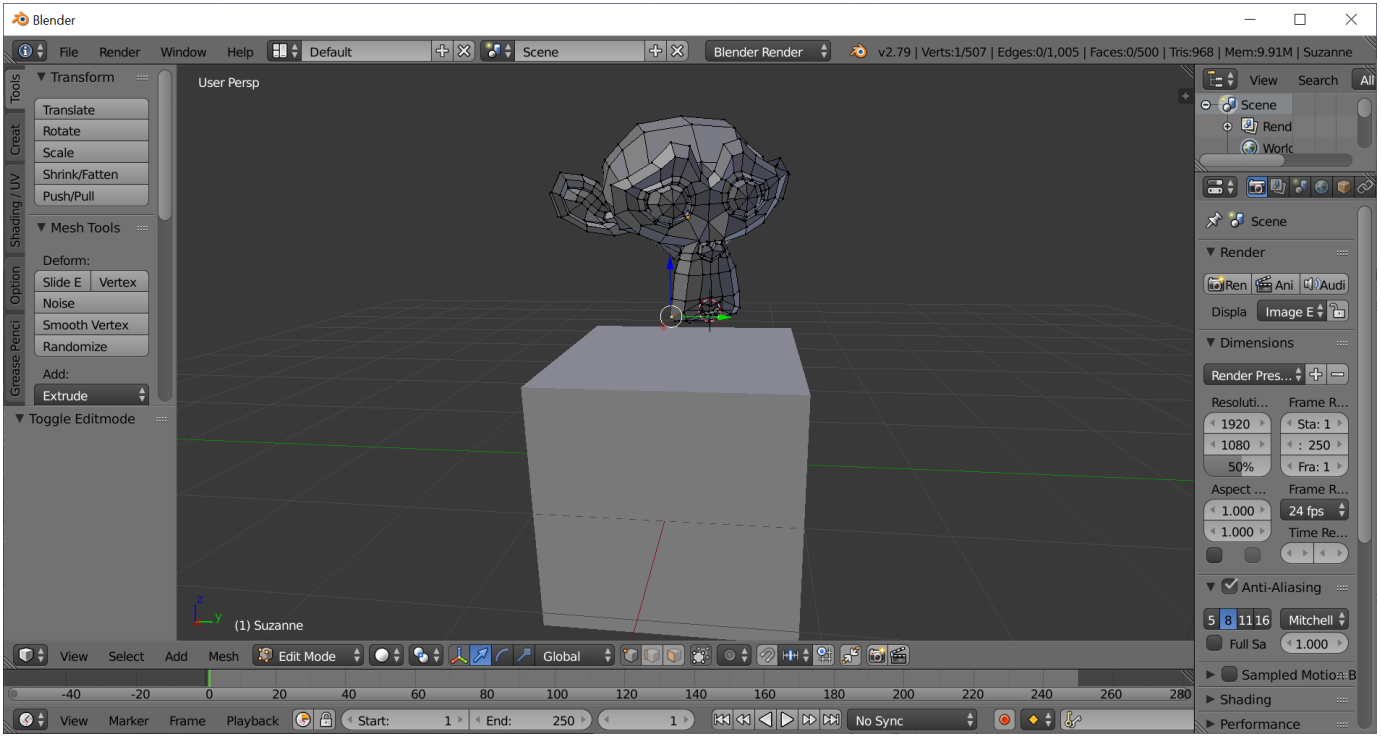
ParaView can be controlled and extended via Python scripts.

Gnuplot



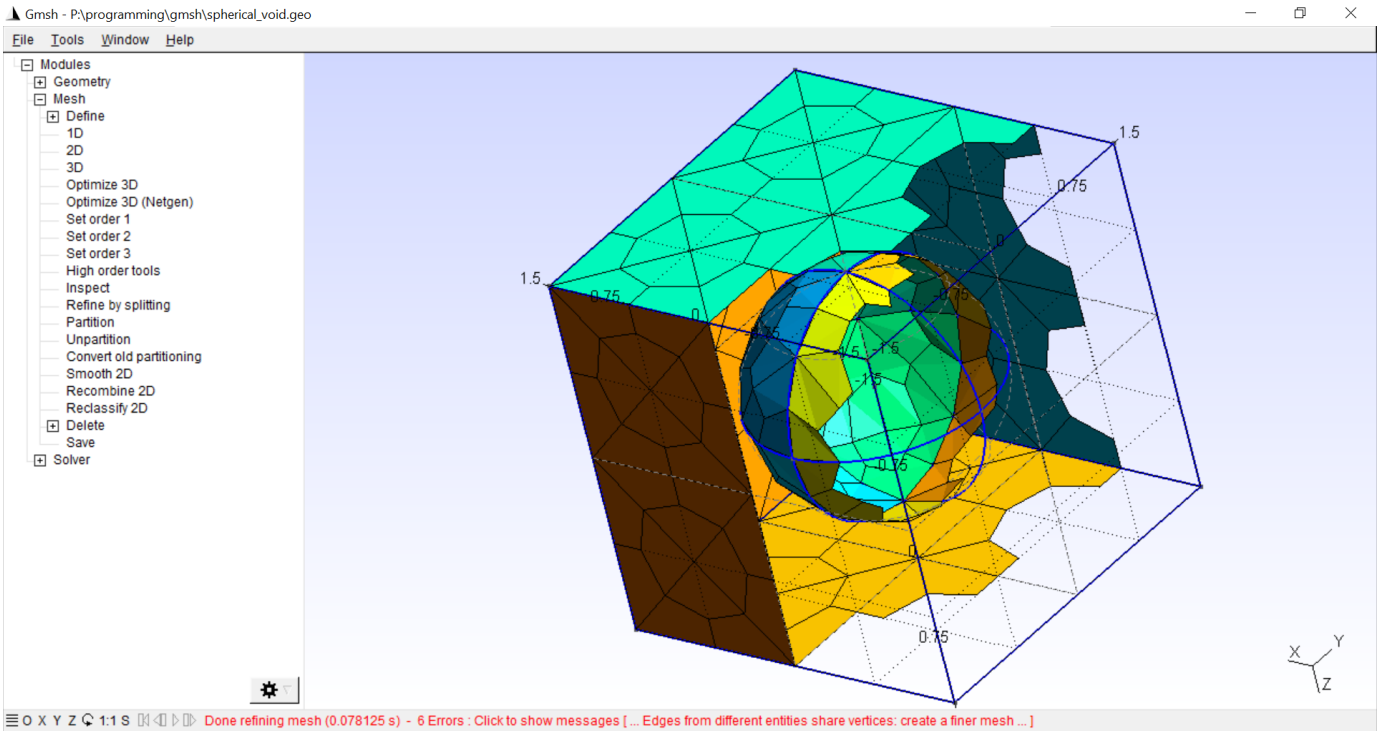
https://www.gnuplot.info

Blender



http://www.blender.org

GMSH



https://gmsh.info

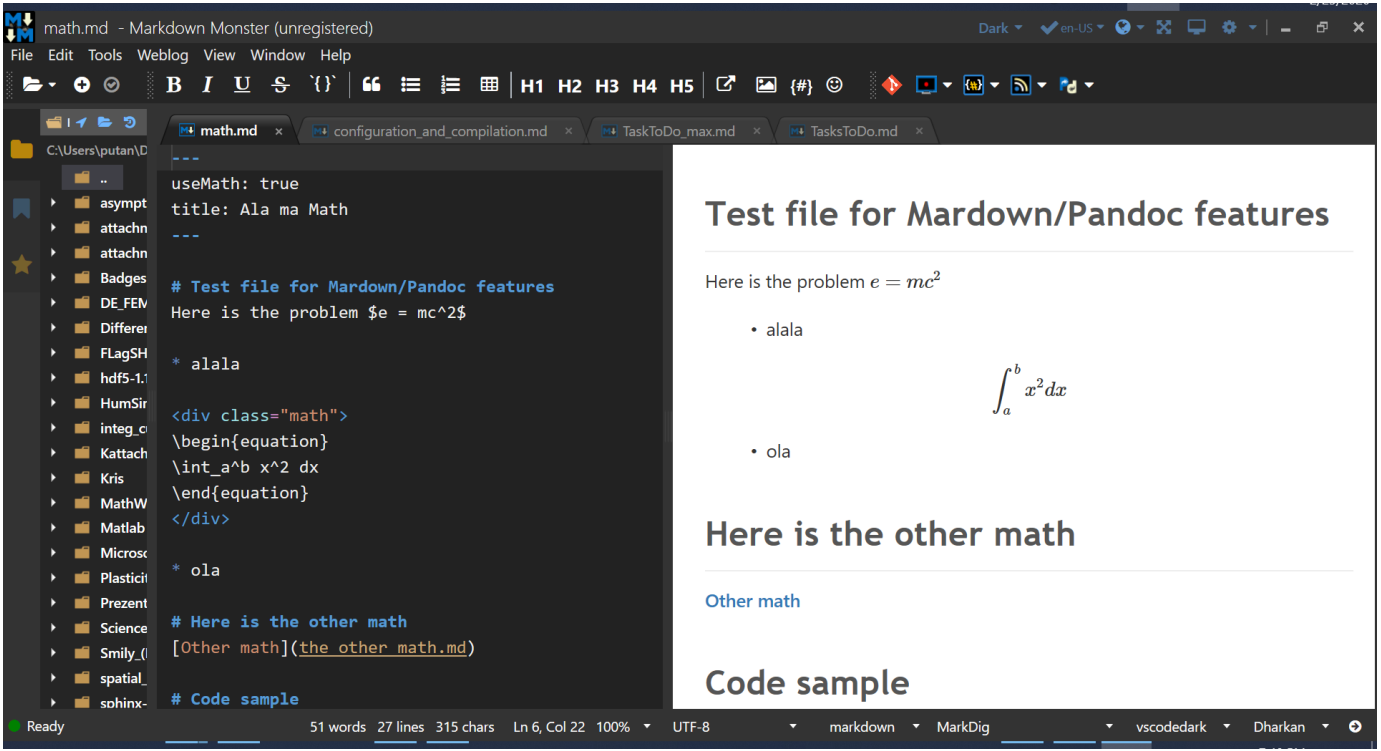
Git

Git is a specific revision control system. Such systems are nowadays the basic tool underlying any but trivial software project. Git (`https://git-scm.com/`) is a *distribute* revision control system and one of the most popular in its category.

The minimal set of Git's commands: `clone`, `add`, `commit`, `status`, `diff`, `pull`, `push`

The best approach to learn Git is to start with its intrinsic command line interface and only after mastering the basic to use one of many Git's graphical user interfaces.

Markdown + Pandoc



https://markdownmonster.west-wind.com/ https://pandoc.org

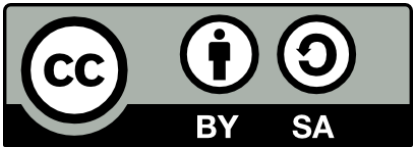
LaTeX(be careful with pronounciation) is a document preparation system that uses special markup language to define document structure and the appearance of rendered final form. The document source is saved as a plain text file what makes it very suitable for storing in all kinds of revision control systems, easy to edit and share.

Sample problems

- Tabulating scalar function of a scalar variable
- Plotting a function of one variable
- Dot product and cross product of vectors
- Rotating points in plane

Thank you for your attention

Copyrights



Roman.Putanowicz@L5.pk.edu.pl