Getting started: Installation & Signups

SM-2302 Software for Mathematicians

July 29, 2025

In this course, we will be making use of four main software:

1. **MATLAB**
2. **Python**
3. **Git & GitHub** (Desktop & online)
4. **Overleaf**

MATLAB

MATLAB is a high-level language and interactive environment for numerical computation, visualisation and programming.

You may use MATLAB in one of three ways listed below.

UBD Computers

If you are using any of UBD's computers, then MATLAB should already be installed. Locate the MATLAB app by scrolling through the start menu. Alternatively, type 'Matlab' in the Windows Run prompt.

Local MATLAB

By local, we mean that you are using your own laptop or computer. You need to ensure that you have done the following:

- Create a Mathworks account using your UBD e-mail
- Access the UBD campus-wide suite using your Mathworks account
- Download and install the latest version of MATLAB (R2024b or R2025a)

Please refer to the MATLAB individual CWL installation guide document.

Generally, the basic MATLAB installation require 2BG storage for MATLAB only. Upon installation, you will be offered other toolboxes which can typically take up 4-6GB each. For the purpose of this module, you can opt for MATLAB only.

MATLAB Online

- MATLAB Online allows access to MATLAB and Simulink from any device with a web browser, eliminating the need for local installation.
- Without access to the UBD CWL license, a basic version of MATLAB Online is free, offering 20 hours of usage per month and access to 10 commonly used products. Roughly speaking, this means you can only open or run projects in your workspace for a little over 4 hours per week. Maybe that's enough, maybe it isn't.

Python

Python is a powerful and versatile programming language widely used across mathematics, science, and engineering. It also plays a central role in the rapidly growing fields of data science and artificial intelligence. In this course, you will use Python to perform symbolic computation, numerical analysis, graph plotting, and explore basic concepts in data science and machine learning. There are several ways to run Python code, depending on your internet access, available tools, and personal preferences.

Recommended: Jupyter Notebook

If your internet connection is stable, we highly recommend using **Jupyter Notebook**. It provides an interactive environment where you can write code, see results, and add explanations in one place. Once installed, it works offline and is very convenient for mathematical tasks.

To get started, first install Python from `https://www.python.org` (version 3.9.5 or newer). During the installation, make sure to check the option Add Python to PATH. Then, open your terminal or command prompt and install the required packages:

```
pip install numpy matplotlib sympy scipy notebook
```

Once installation is complete, you can launch Jupyter by typing in your terminal:

```
jupyter notebook
```

A browser window will open where you can create and edit notebooks.

Alternative: Google Colab (No Installation)

If you prefer not to install anything, **Google Colab** is a great alternative. It is a free, browser-based platform provided by Google that supports Python notebooks. You only need a Google account.

To start using Colab, go to: https://colab.research.google.com. Click File New Notebook and you can immediately start writing and running Python code. Your notebooks will be automatically saved to your Google Drive.

Colab already includes most packages we need (such as `numpy`, `matplotlib`, and `sympy`), so you can use it without additional setup.

Full Local Installation (Optional)

If you want full control or prefer working entirely offline, you can install everything on your computer without Jupyter or Colab. After installing Python from the official website, you can write and run Python code using a text editor (like VS Code or Notepad++) and run it from the terminal or command prompt.

This method gives you the most flexibility, but it requires managing your files manually and doesnt offer the notebook interface.

Essential Python Packages

In all three options above, the following Python packages will be used frequently:

- `numpy` for numerical operations and arrays
- `matplotlib` for plotting graphs
- `sympy` for symbolic math (algebra, calculus)
- `scipy` for scientific computing

Final Notes

For most students, we strongly recommend using **Jupyter Notebook** unless internet limitations make it difficult. If you want a quick and easy start, **Google Colab** is also an excellent choice. Choose the method that suits your setup best, and make sure you are comfortable working with the terminal before our first python session.

Git & GitHub

There are two things needed here: a working git installation, and a GitHub account (and subsequently the Desktop app).

Luckily, if you are using macOS, git comes preinstalled, so you need not do anything. Especially if you tried installing Xcode from the above step, then you should be good to go.

For Windows, you need to download and install from here. Apparently, the 64-bit version is recommended when possible.

Next, sign up for a GitHub account from https://github.com.

You should use your UBD e-mail to sign up for GitHub, so that you can be eligible for student benefits. To apply, click on 'Get student benefits' from this link. This step is not really necessary for this course, but moving forward you might want these benefits–in particular it will allow you to host static websites for free (GitHub pages).

Important points

Some advice about selecting your account names (particularly for GitHub)[1]:

- Incorporate your actual name! People like to know who they're dealing with. Also makes your usename easier for people to guess or remember.
- Reuse your username from other contexts (e.g. social media platforms).
- Pick a username you will be comfortable revealing to your future boss.
- Shorter is better than longer, but be unique as possible.
- Make it timeless. Avoid highlighting your current university, employer, or place of residence.

At some point, you will be invited to the 'SM-2302' GitHub organisation. This is where all our code repositories will live. You can browse through some that we have created and you can explore GitHub features such as commits, issues, and GitHub Pages.

Assignments will also be distributed using GitHub (Classroom) – although there will still be a Canvas crossover so as to keep your academic activities within the Learning Management System (LMS).

Overleaf

We have decided that the path of least resistance to get up and running and typesetting LaTeX documents is by using the online editor called Overleaf. You will have to sign up for an account on `https://overleaf.com`, and to use Overleaf it obviously requires an internet connection.

For more advanced users, LaTeX can be installed on your local computers. This requires installing (and maintaining!) a TeX distribution, and selecting an IDE to work with. Now, going down this route is a little bit messy and things might not always work.

On the other hand, Overleaf has so many benefits for the new user that it allows us instructors to focus on actually teaching you to typeset rather than spending time debugging. A favourite of ours is the very forgiving way that it handles errors, which can literally drive you crazy otherwise.

General advice

- If you are using campus computers, then dedicate a folder where you will store all your work files. Note that these files are retained on the PC you are currently using (since it does not have individual logins). Perhaps a USB drive or some cloud storage would be useful as well.

[1] Taken from sta323 course at Duke

- Using your own device? It is recommended to work with either a laptop or desktop computer, and not a tablet. Generally speaking, tablets do not provide sufficient compute power and you may find the file system a bit clunkier to manage.

- Practice safe and secure logins. Keep your passwords secure and do not recycle them, especially given the multiple logins you need to create. I would recommend using a password manager, not just for this course, but beyond this. Also, two-factor authentications are a good idea too, so enabling that is highly recommended.