Hillslope Morphology in Tectonically Active Settings

1. Introduction

In this practical, you will use a computer programming language called Python to perform data analysis on hillslope morphology data generated for you from a digital elevation model (DEM). You will do this by working through an interactive notebook exercise that combines the instructions for this class with snippets of computer code that you will executed in order as you work through the exercises.

This laboratory session is **not** designed to teach you how to be a computer programmer! That would take a little longer than a two hour class. Rather you will **use** code that is already written and tested in order to better understand the controls on hillslope morphology in a tectonically active setting. Hopefully you will also get an appreciation for how powerful python can be in helping us manipulate and analyse large datasets. You will only touch the surface of what is possible with Python in terms of data analysis, but use of such software is becoming more and more commonplace across academia and industry as our data gathering capabilities expand.

1.1. Intended Learning Outcomes

In the IPython notebook you will download (instructions below) are step-by-step instructions to introduce the IPython environment, perform numerical analysis and modelling, and to generate plots and figures that could be used in project work. By the end of this practical, each student will be able to:

- Follow and execute python code to analyse hillslope morphology extracted from a DEM.
- Further understand the concept of steady-state and transience in hillslope morphology
- Quantify metrics of dimensionless hillslope morphology
- Analyse hillslope morphology at a landscape scale to evaluate the distribution of erosion/uplift.
- Create and explain figures that show variation in hillslope form in a tectonically active landscape

1.2. Assessment

There is one assessment task highlighted in a blue box at the end of the IPython notebook, consisting of a mini essay on the results generated during the lab. You are required to **compile the figures and answer in a typed, two page A4 document** with font Calibri/Arial, font size 11 and page margins of 2 cm. The submission deadline for all practical assessments is 27th November 2017 and submission will be on Moodle. You should build a single document containing all four practical class assessment exercises for submission by this date, after the last practical class takes place on 15th November 2017.

1.3. Plagiarism

Feel free to work together and discuss your work openly in the class. However, your individual written work and final submission should be your own. Plagiarism is a very serious matter, with serious consequences. More information can be found on links provided in the course handbook (available on Moodle).

2. Download and Run the IPython Notebook

2.1. Download the notebook

The notebook file **Hillslope_Morphology_Tectonics_Lab.ipynb** can be found in the Lab 4 folder on Moodle. Download this file to a location on your personal space where you usually store your coursework and put it in a new folder that will contain all of your Lab 4 work (N.B. This **does not** have to be on the D: drive).

2.2. Open the notebook

You **cannot** open this file by double clicking on it. Instead we will open the notebook with some software called Jupyter Notebook. The Jupyter notebook is an extension of your internet browser so will open as a new tab inside Chrome. To open Jupyter Notebook:

- *Click Start and type "**cmd**" then press Enter*
- *At the command prompt type "**jupyter notebook**" then press Enter*
- *Navigate to the folder with the 1DHillslope.ipynb and click on it to open it.*

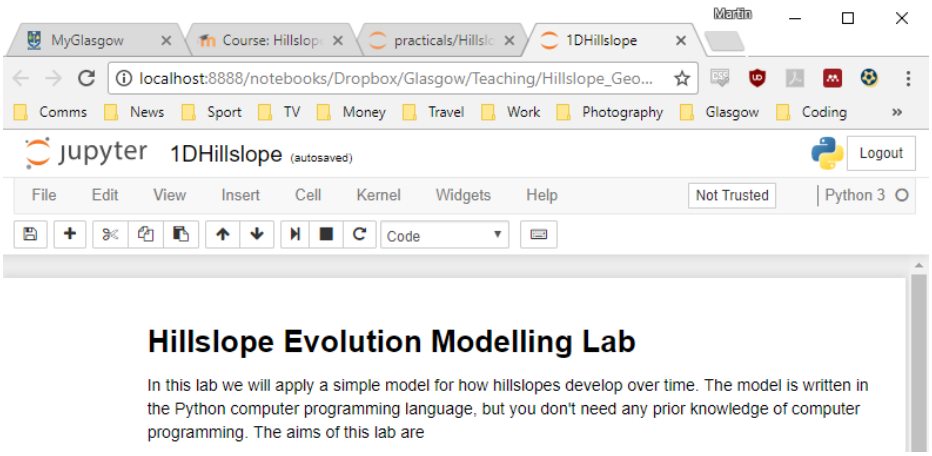

Figure 1: The Jupyter notebook environment with instructions for Hillslope Evolution Modelling practical class

3. Work through the exercises

The IPython Notebook (Figure 1) is a document that combines text and computer code. Your task is to **carefully** read the text boxes sequentially and run each of the blocks of Python code that have been provided in the IPython notebook.

To run a code block in the IPython notebook, click in a code cell, hold down **shift**, and press **enter**. An asterisk in square brackets `In [*]:` will appear while the code is being executed, and this will change to a number `In [1]:` when the code is finished. The order in which you execute the code blocks matters, they must be run in sequence. As you get more familiar with this environment, feel free to experiment with typing your own code into a cell and running that, but you will be given explicit instructions when you are required to change anything.