Modelling Hillslope Evolution by Diffusion

1. Introduction

In this practical, you will use a computer programming language called Python to explore and simulate how hillslopes change through time. 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 hillslope evolution. Hopefully, you will also get an appreciation for how powerful computer models can be in helping us understand how landscapes change over longer timeframes than we can observe in a human lifetime. We will use quite a simple example of a one-dimensional hillslope today, but the principles involved are used across a wide range of modelling packages you might encounter in your professional lives, from climate modelling to flood risk prediction.

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:

- Run existing blocks of Python code in an IPython environment
- Recognise the importance of different sediment flux rules in governing hillslope form
- Execute a simple model of hillslope evolution
- Appreciate the concepts of topographic steady-state and response times on hillslopes
- Create and explain striking plots and figures that could be used in a project to show modelling results

1.2. Assessment

There are six assessment tasks highlighted in blue boxes in the IPython notebook. You are required to **compile the figures and answers in a typed, two page A4 document** with font Calibri/Arial, font size 11 and page margins of 2 cm. Each task weighting appears in the assessment box heading (%). 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 lab materials

Download the Lab 2 **folder** from Moodle. Unzip the contents and put the folder on your personal space in a new folder where you will conduct all of your Lab 2 work. (N.B. This does not have to be on the D: drive this time!).

2.2. Open the notebook file (1DHillslope.ipynb)

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.*

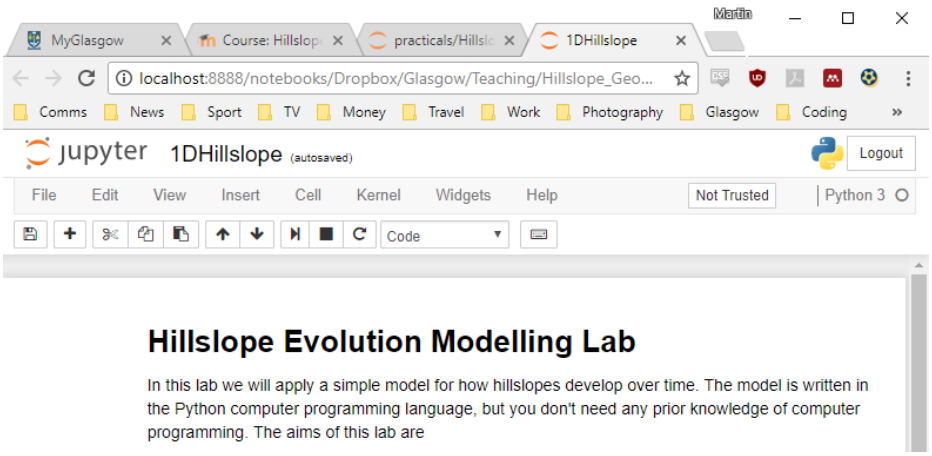

Figure 1: The Jupyter notebook environment with instructions for Hillslope Evolution Modelling practical class

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.