|  |  |  |  |
| --- | --- | --- | --- |
|  | | logo_black.gif | |
|  | 201297726 |  | Total. 2500  Report-1005 |
|  | GEOG5990: Programming for Geographical Information Analysis: Core Skills |  |  |
|  | project |
|  |  |

|  |
| --- |
|  |
| 1. |
| 2. |
| 3. |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

**REPORT ON THE DESIGN OF A PROGRAM TO FIND THE MOST EXTREME GRADIENTS ON WHICH TO SET UP SNOWBOARD RUNS FOR AN EXTREME SPORTS HOLIDAY COMPANY USING PYTHON PROGRAMMING.**

1. **INTRODUCTION**

Snowboarding is among the common extreme sports which are also known as action sports that are of high speed and risk (Britannica, 2020). This means that a lot of care, design and planning needs to be put in consideration while designing snowboard runs in any extreme sports holiday company to ensure safety of the users (Laver et al., 2017). To achieve this, a number of methods can be used and python programming is among those that can effectively calculate slopes and gradients of an area (Lemenkova, 2019).

Therefore, this report intends to display a python program that finds the most extreme gradients on which to set up snowboard runs for an extreme sports holiday park. The program will pull in hillslope heights from a raster text file containing heights of the area, calculate the maximum slope in all the raster cells using D8 algorithm, build datasets with slope gradients rather than the heights, display the heights and gradients as an image and save them in similar formats.

1. **METHODOLOGY**
   1. **Data Sources**

The data used for this program is raster data in text format (.txt) containing heights in numbers of meters above sea level of the study hilly area. The data is a 300 by 300 raster grid of heights labeled snow.slop.txt.

* 1. **Technologies used**
     1. **Spyder**

The scientific python development environment commonly called spyder is a free integrated development environment (IDE) that automatically comes after installation of Anaconda and has capacity to edit, undertake interactive testing, debug codes and introspect them as well (Anaconda, 2021).

* 1. **General sources used**
     1. **Agent Based Modelling (ABM)**

The program was built basing on use of an Agent Based Model (ABM) that makes it easy to incorporate randomness especially when dealing with complex systems this making modelling of natural, ecological and social phenomena better too (Wilensky and Rand, 2015). The Agent based Model used to design this program relied on three basic elements namely the model code which acts as the main set up of the entire program that reads in and runs the data, there is also the agent framework code which builds the agents that initiate communications with other codes and lastly the environment code which is usually contains the key data that we are dealing with which other components will interact (Wilensky and Rand, 2015).

* 1. **Libraries used** 
     1. **NumPy**

This program relied on using NumPy to obtain gradients since this is the best python package used in performing mathematical calculations on arrays with capacity to convert NumPy arrays to images, python lists and so on being able to also calculate gradients using the numpy.gradient() method (NumPy, 2021).

* + 1. **Matplotlib**

Matplotlib is a key tool in any python program because of its capacity to visualise and display outputs in an understandable format such as bar graphs, scatters, images among others for example using matplotlib.imshow() method to show images (Hunter, 2007).

* + 1. **Random**

This is a key module in the program used to generate numbers pseudo-randomly for example for integer values, these are generated at a uniform range (Python, 2021).

* + 1. **csv**

To read in the data, the csv module is used making it the most widely used data reader in python programs with ability to import and export data in text files and spreadsheets (Python, 2021).

* + 1. **Agent framework**

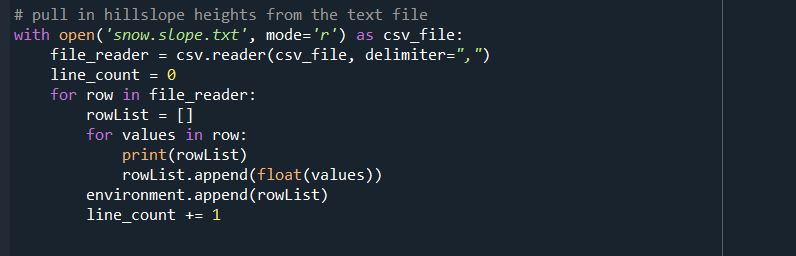
Since the program was built as an Agent Based Model, the agent framework code had to be connected to the main Model code through importing in the code to enable the communications and interactions between the two models.

* 1. **D8 Algorithm**

The program intended to calculate maximum slope for each cell in the raster file using the D8 algorithm therefore the neighborhood for a particular cell was set to 8. With this algorithm, the flow direction is computed whereby each cell is routed to one of the eight neighbouring cells relying on the steepest descent direction (Garbrecht and Martz, 1997).

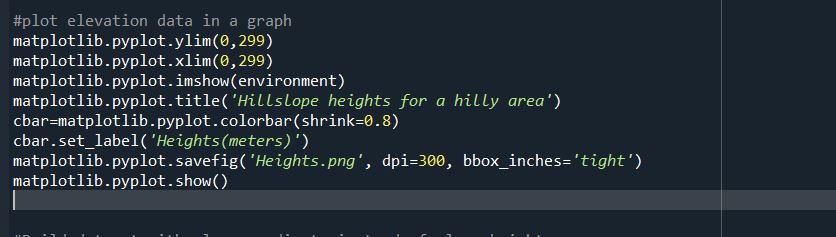
1. **RESULTS**

The program successfully read in the data using the csv module. To create the agents, a number of agents was set to 300 with 100 iterations and 8 neighbours as required to obtain the D8 algorithm. Creating the distance function between the agents as directed in the assignment and obtaining the square root, the maximum slope value for each cell was obtained and printed in the console.

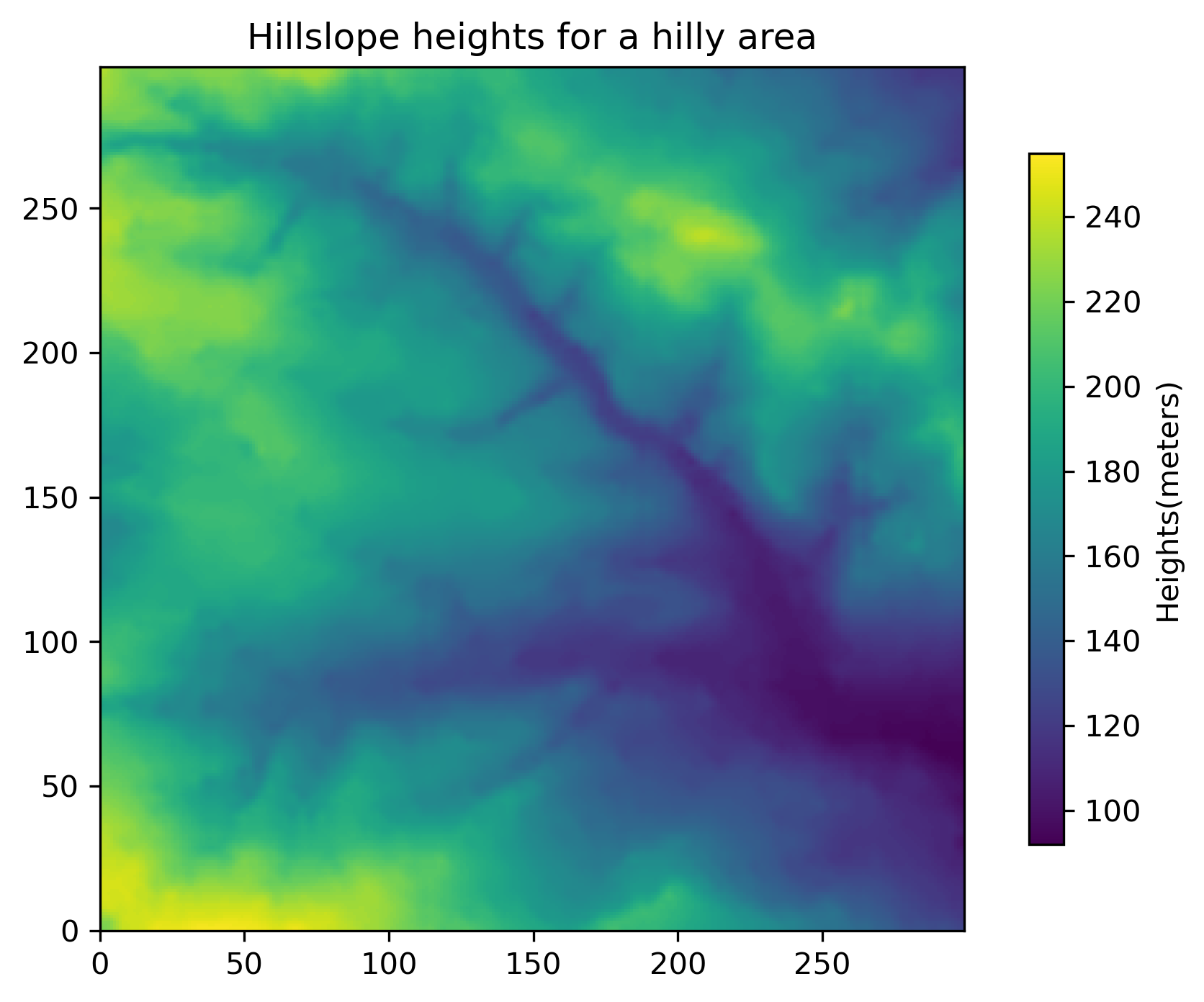


***Figure 1: Showing a code line in the Model reading in the data***

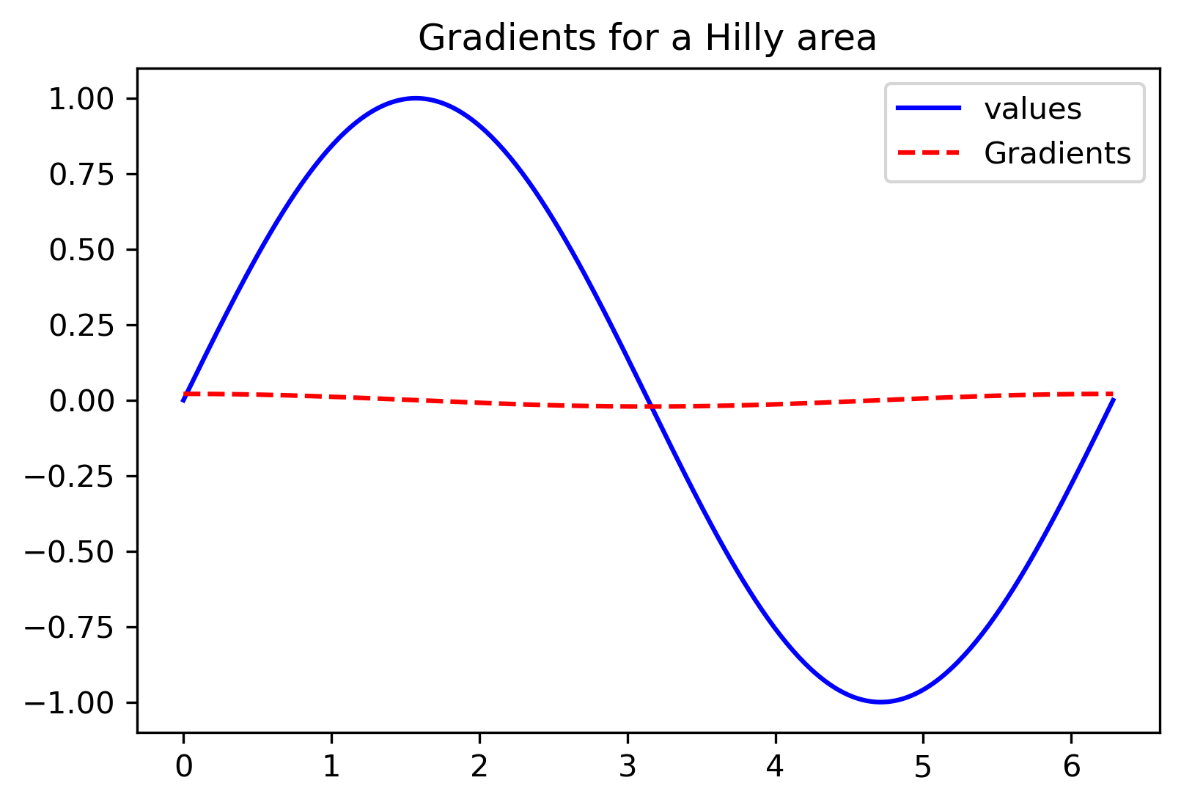
Uisng Matplotlib, the that was read in and processed was able to be visualised in the console depicting the slope heights in meters and the gradient was calculated using the NumPy.gradient() function to enable developers understand areas with positive and negative gradients in the area for snowboard runs. The code was able to show the products and also save them as images to the working directory.

******

***Figure 2: Showing use of Matplotlib to visualise the environment***

****

***Figure 3: Showing an image of the hillslope heights***

******

***Figure 4. Showing gradients (positive and negative) of the snowboarding run hilly area.***

1. **LIMITATIONS DURING DEVELOPMENT**

The biggest challenge faced with developing this program was with the application of the D8 algorithm as this is understood to deal mainly with flow direction and now had to relate it with distances between the cells(agents) for a snowboarding run. However, following my understanding of what the explanation in question says, I set the neighbourhood to 8 and ensured that the greatest value among the neighbours sets the maximum slope for the cell.

Another limitation with the code appeared with the use of numpy.gradient () to calculate the gradient as this varied a lot from relevant sources each applying it differently and mostly with Digital Elevation Data in geotiff, tiff or ASC II grid format.

1. **CONCLUSION**

The program set out to display hillslope heights, gradients and maximum slope for each of the cells using D8 algorithm in a 300 by 300 raster text file which was successfully achieved. This shows the power python programming has in solving complex calculations and modelling in the natural, ecological and social phenomena.

1. **REFERENCES**

Anaconda. 2021. Spyder. [Online]. [Accessed 10 May 2021]. Available from: <https://docs.anaconda.com/anaconda/user-guide/tasks/integration/spyder/>

Britannica. 2020. The Editors of Encyclopaedia. "Extreme sports". Encyclopedia Britannica. [Online]. [Accessed 8 May 2021]. Available from: [https://www.britannica.com/sports/extreme-sports. Accessed 16 May 2021](https://www.britannica.com/sports/extreme-sports.%20Accessed%2016%20May%202021).

Garbrecht, J. and Martz, L. W. 1997. The Assignment of Drainage Direction Over Flat Surfaces in Raster Digital Elevation Models*. Journal of Hydrology*. **193**, pp.204-213.

Hunter, J.D. 2007. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*. **9**(3), pp. 90-95.

Laver, L., Pengas, I.P. and Mei-Dan, O. 2017. Injuries in extreme sports. *Journal of Orthopaedic Surgery and Research*. **12**(59).

Lemenkova, P. 2019. Calculating slope gradient variations in the submarine landforms by R and Python statistical libraries. *MANAS Journal of Engineering*. **7**(2), pp.99-113.

NumPy. 2021. What is NumPy? [Online]. [Accessed 11 May 2021]. Available from: <https://numpy.org/doc/stable/user/whatisnumpy.html>

Python. 2021. csv- csv file reading and writing. [Online]. [Accessed 6 May 2021]. Available from: <https://docs.python.org/3/library/csv.html>

Python. 2021. Random-Generate pseudo-random numbers. [Online]. [Accessed 6 May 2021]. Available from: <https://docs.python.org/3/library/random.html>

Wilensky, U. and Rand, W. 2015. An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo. MIT Press.