B.Tech Project-II
(CE47006)

A Simulator for Surface Hydrology with Graphical User Interface using Python

Ashok Jangir
19CE10012

Spring Semester, 2022-23

Under the Supervision of

Prof. Dhrubajyoti Sen

Department of Civil Engineering
Indian Institute of Technology, Kharagpur
West Bengal, 721302

ACKNOWLEDGEMENT

Indian Institute of Technology, Kharagpur, offers a lot in terms of exploration and execution of various research topics. I express my awe and acknowledge the various facilities this institute has to offer.

With utmost pleasure, I extend my deepest appreciation to my esteemed guide, **Prof. Dhrubajyoti Sen** (Professor, Department of Civil Engineering, Indian Institute of Technology, Kharagpur), for their invaluable guidance, dedication, and unwavering support throughout my B.Tech thesis project. Their willingness to invest their precious time, expertise, and knowledge in training, educating, and assisting me has been remarkable. I am grateful for their proactive feedback and constant motivation, which have been instrumental in my progress and success during this project.

I would also like to express my heartfelt gratitude to **Mr. Pankaj Singh** (Research Scholar, Department of Civil Engineering, Indian Institute of Technology, Kharagpur) for their valuable suggestions and assistance, which have played a significant role in the completion of this project.

Furthermore, I am indebted to my beloved family members for their unconditional love, support, and understanding throughout my academic journey. Their constant encouragement, patience, and belief in my abilities have been a constant source of motivation for me. I deeply appreciate their sacrifices and the unwavering faith they have shown in me.

DECLARATION

I certify that

a) The work contained in this report has been done by me under the guidance of my supervisor.

b) The work has not been submitted to any other Institute for a degree or diploma.

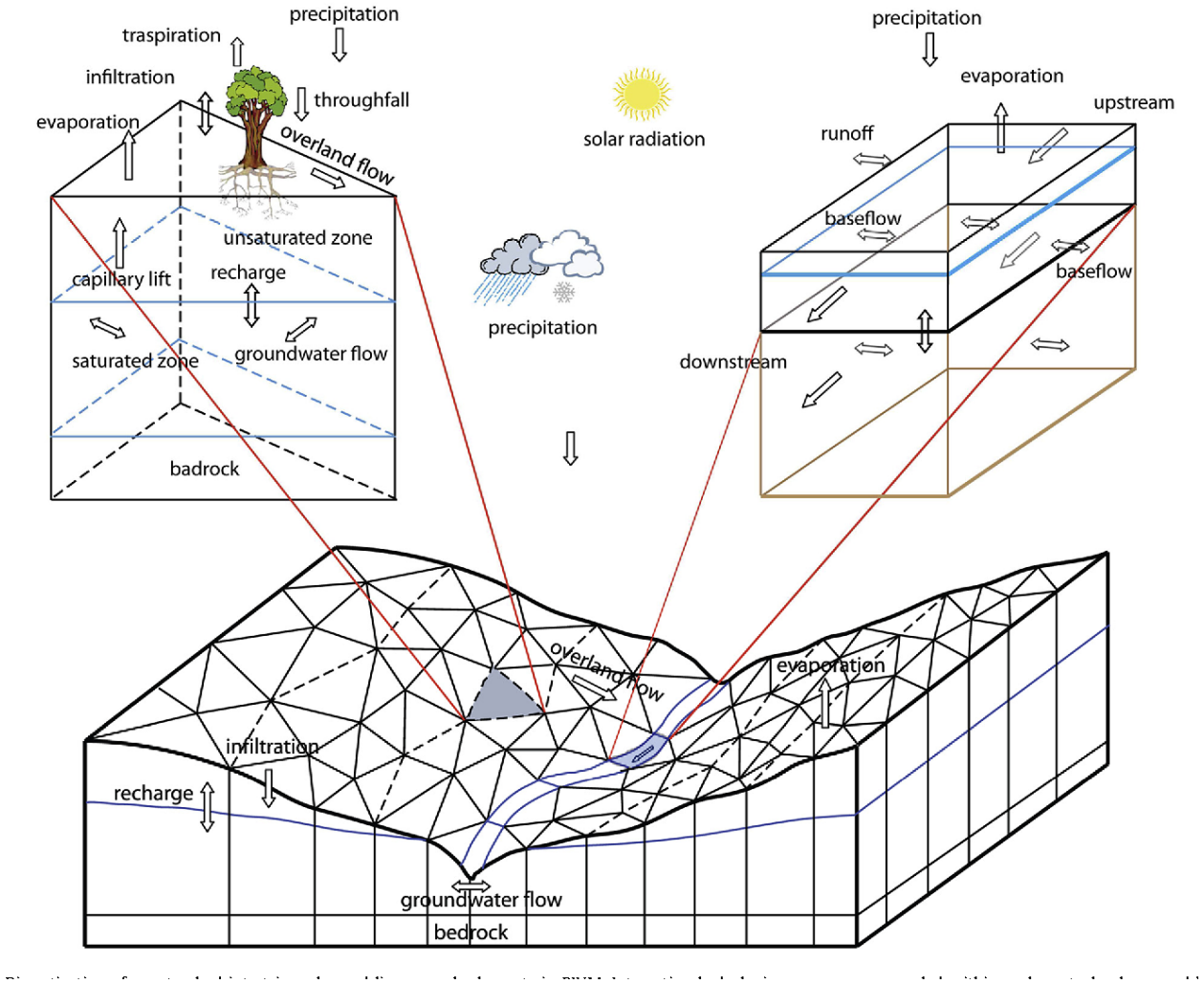
c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in references.

CONTENTS

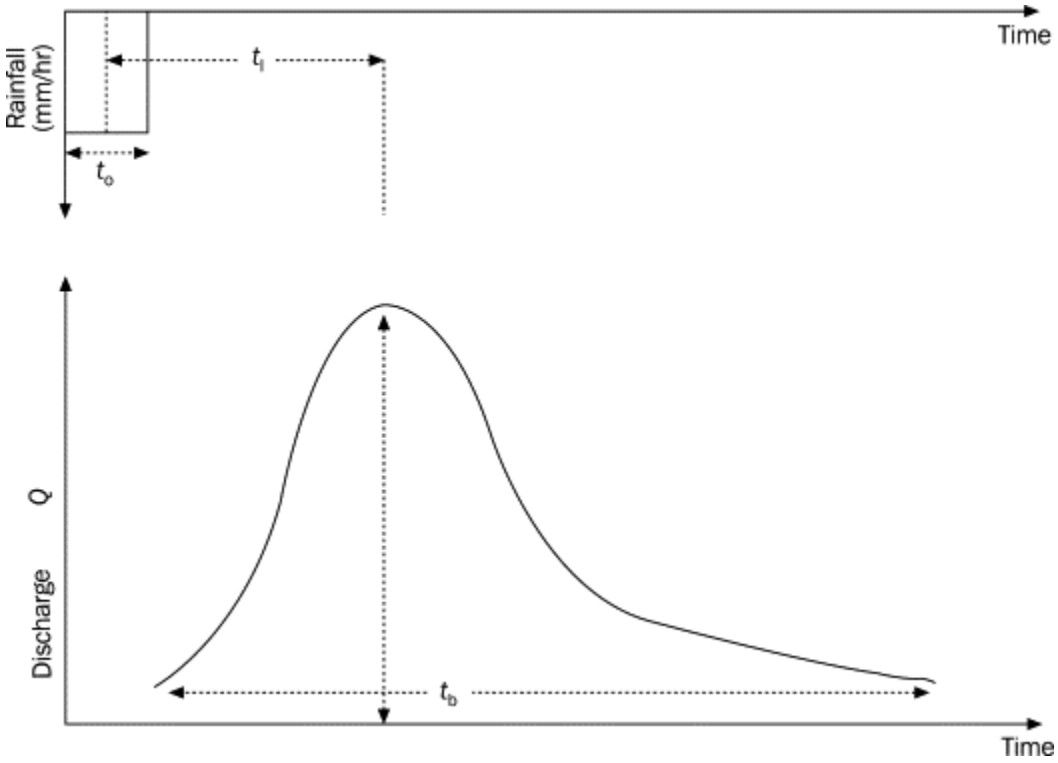
INTRODUCTION

In the realm of hydrological studies, accurate simulation of surface hydrology is of paramount importance for understanding and predicting the behavior of water on the Earth's surface. Hydrograph simulation, in particular, plays a crucial role in assessing the impacts of various land use practices, urban development, and climate change on water resources. The ability to simulate and visualize hydrological processes efficiently not only aids in decision-making for water resource management but also facilitates effective communication of scientific findings to diverse stakeholders.



[https://www.semanticscholar.org/paper/A-tightly-coupled-GIS-and-distributed-hydrologic-Bhatt-Kumar/715dd10d9c3a356ef61c576bf1afb8f704bfd0ca/figure/1]

In this project, I developed a hydrograph simulator with a user-friendly Graphical User Interface (GUI) using the Python programming language. The implementation of the simulator leverages the robust computational capabilities of Python, which has gained popularity in scientific computing due to its extensive libraries and packages. By harnessing Python's capabilities, we have created a flexible and customizable simulation framework that can model a wide range of hydrological phenomena. Furthermore, the inclusion of a GUI enables users to interact with the simulator effortlessly, allowing them to input parameters, visualize simulated hydrographs, and interpret results in real time.



[https://ars.els-cdn.com/content/image/3-s2.0-B9781855738348500088-f08-23-9781855738348.gif]

OBJECTIVE

The objectives of the hydrograph simulator, which takes topographic data as input and provides results for peak discharge and respective time, are as follows:

1. Develop a hydrograph simulation model that utilizes topographic data as input to accurately represent the surface hydrology processes, including rainfall-runoff transformations, within a given watershed.

2. Implement algorithms within the simulator to calculate peak discharge, which represents the maximum flow rate during a storm event, and determine the corresponding time at which it occurs. These results will provide valuable insights into the behavior of water flow within the watershed.

3. Design a user-friendly Graphical User Interface (GUI) that allows researchers, hydrologists, water resource managers, and students to easily input topographic data and visualize the simulated hydrographs with peak discharge and corresponding time. This interface will facilitate efficient analysis and interpretation of the simulation results.

4. Demonstrate the practical utility of the hydrograph simulator by showcasing its application in various scenarios, such as land-use change assessments, urban development planning, and climate change impact analysis. These case studies will highlight the simulator's potential to support decision-making processes related to water resource management.

5. Create an accessible and engaging platform for students to conduct fun experiments and learn about hydrograph simulation. The simulator will foster curiosity, exploration, and understanding of surface hydrology principles among students by providing a user-friendly interface and interactive visualization capabilities.

By achieving these objectives, the hydrograph simulator will serve as a valuable tool for researchers, hydrologists, water resource managers, and students. It will aid in conducting in-depth analyses of watershed behavior, supporting evidence-based decision-making processes, and fostering learning and experimentation in the field of surface hydrology.

METHODOLOGY

Methodology for Hydrograph Simulator:

1. Select Topographical Sheet from Hardware:
    - Obtain a topographical sheet of the desired study area from the available hardware or data sources.

2. Select Scale Points from the Topographical Sheet:
    - Identify key points on the topographical sheet that represent the desired scale for the simulation.
    - These points must be distinct so that they represent the finite segment's endpoints.

3. Set Corresponding Scale Value:
    - An appropriate scale value corresponds to selected scale points.
    - This will calculate the ratio between the distance on the topographical sheet and the corresponding distance on the ground.
    - Apply the scale value to represent the terrain in the simulation accurately.

4. Select Catchment Points:
    - Identify the points within the topographical sheet which encloses the catchment area where hydrological processes will be simulated.
    - These points should include areas of interest such as rivers, streams, and potential runoff areas.

5. Select Valley Nodes and Form Valley Segments:
    - Identify the nodes along the valley lines within the catchment area.
    - Connect the valley nodes to form valley segments, representing the path of water flow in the valleys.

6. Select Ridge Nodes and Form Ridge Segments:
    - Identify the nodes along the ridges within the catchment area.
    - These nodes will get connected one after another in the order they get selected from the topographical sheet.

7. Form Contour Lines from Joining Contour Points:
    - Identify contour points on the topographical sheet that represent changes in elevation.
    - Selected contour points should have the same elevation, which will get connected after entering the elevation value.

8. Triangulation from Collected Segments and Nodes:
    - Perform triangulation using the collected valley segments, ridge segments, contour lines, and all input nodes.
    - After Triangulation, the surface will be represented by appending the elevation values to the 2D triangulated points by a network of interconnected triangles.

9. Compute Surface Flow over Triangular Cells using Zero-Inertia Model:
    - Apply the zero-inertia model to simulate the flow of water over the triangulated surface.
    - Consider slope, roughness, and rainfall intensity to determine the flow rates and directions.

10. Compute Flow through Valley Segments using Kinematic Wave Equation:
    - Apply the kinematic wave equation to calculate the flow rates through the valley segments.
    - Parameters such as channel slope, rectangular cross-sectional area, and Manning's roughness coefficient were considered.

Following these steps, the hydrograph simulator can accurately represent the topography and simulate the hydrological processes within the selected catchment area.

Zero-Inertia Model:

The zero inertia model is a simplified approach for computing surface flow over triangular cells in hydrological and hydraulic models. It assumes that there is no inertia in the flow, neglecting the acceleration terms in the governing equations. The whole domain is closed within the catchment area. Each triangular cell receives the rainfall for the step timing of the simulation. Edges of triangular cell may either (1) be connected to a neighbor cell, or be (2) no flow boundary, or (3) have an outflow boundary. If a neighborhood cell is present, flow occurs due to the difference in the water elevation of cells, approximated by Manning's equation. The program jumps to the next edge if it encounters no flow edge(i.e.,

Catchment boundary or ridge segment). If the outflow boundary (i.e., valley segment) is there, the shifted water amount is calculated using the weir equation.

Kinematic Wave Approximation:

Below two equations (Chow et al., 1988) are used,

$$\frac{\partial Q}{\partial x} + \frac{\partial A}{\partial t} = q \quad \text{.....(1)}$$

$$S_o = S_f \quad \text{.....(2)}$$

Here, q = Lateral inflow entering the channel from its sides velocity
Q = Discharge passing through the section
A = Cross-sectional area of flow in the channel
S_o = Friction slope
S_f = Longitudinal slope of the channel bed

As we know the manning equation,

$$Q = \frac{S_o^{1/2}}{nP^{2/3}} \times A^{5/3} \quad \text{.....(3)}$$

Here, P = Wetted perimeter
n = Manning's channel friction coefficient

This can be re-write into,

$$A = \left(\frac{nP^{2/3}}{S_o^{1/2}}\right)^{3/5} \times Q^{3/5} \quad \text{.....(4)}$$

Solve eq.(1) and eq.(4), by using $\frac{\partial A}{\partial t} = \frac{\partial A}{\partial Q} \times \frac{\partial Q}{\partial t}$. We get,

$$\frac{\partial Q}{\partial x} + \alpha\beta Q^{\beta-1} \times \frac{\partial Q}{\partial t} = q \quad \text{.....(5)}$$

Here, $\alpha = \left(\frac{nb^{2/3}}{S_o^{1/2}}\right)^{3/5}$ (For Rectangular section of width b)

$\beta = 3/5$

For computing the discharge along the length of the channel, the finite difference numerical method can be applied. The length of the channel can be divided equally into a number of segments. The discharges at the computational nodes, assumed to be located at the interface of neighboring segments, are computed at specified intervals of time. The approximations of the derivative of the discharge Q can be expressed by:

$$\frac{\partial Q}{\partial x} = \frac{Q_i^{j+1} - Q_{i-1}^{j+1}}{\Delta x} \quad \text{.....(6)}$$

Here, Q_i^j = Discharge Q with the space variable x for the i-th time computational node

Δx = Length of a segment

Similarly, the derivative in time for the discharge variable can be expressed as:

$$\frac{\partial Q}{\partial t} \approx \frac{Q_i^{j+1} - Q_i^j}{\Delta t} \quad \text{.....(7)}$$

Here, Δt = time interval

Discharged value can be computed by averaging process of two values of discharge in time and space,

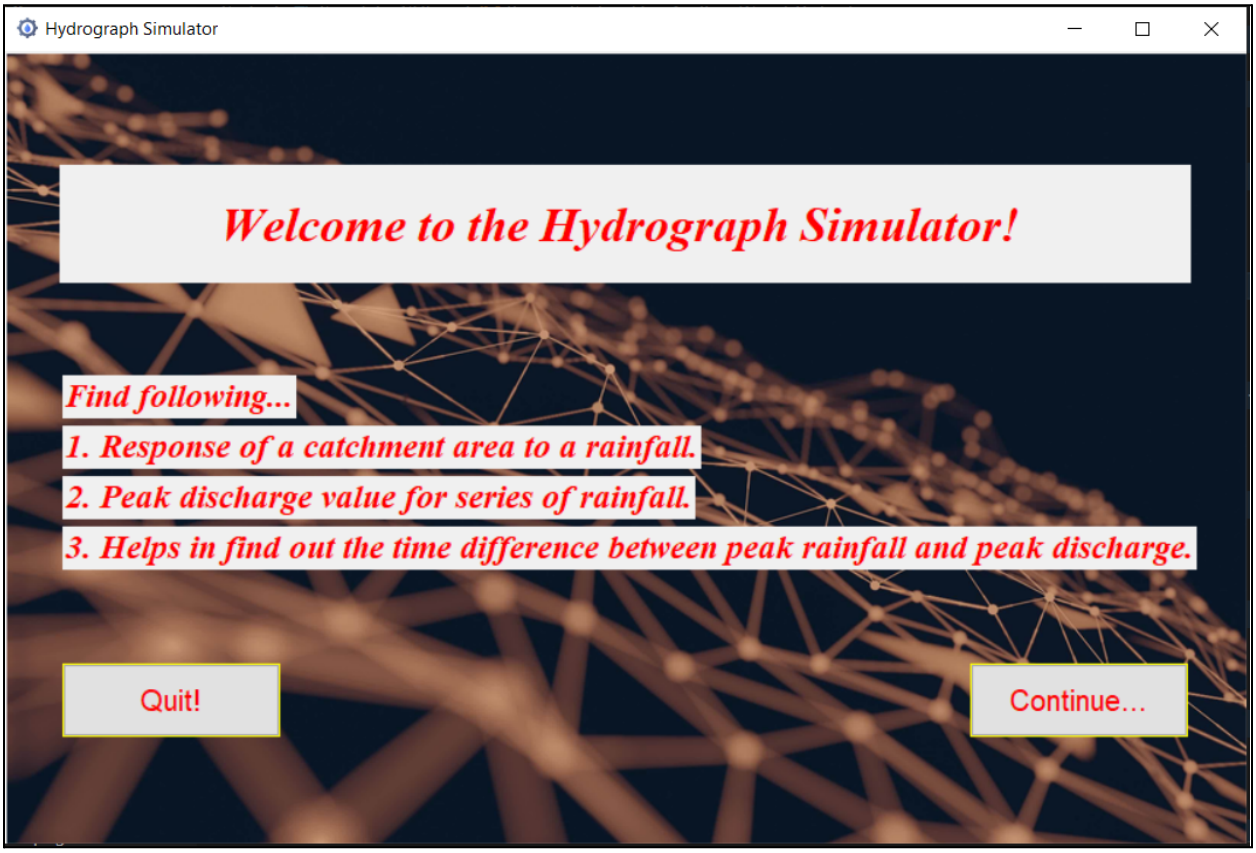
$$Q \approx \frac{Q_i^j + Q_{i-1}^{j+1}}{2} \quad \text{.....(8)}$$

Now, combine eq. (5), (6), (7) and (8), we get below final equation,

$$Q_i^{j+1} = \frac{\frac{\Delta t}{\Delta x} Q_{i-1}^{j+1} + \alpha\beta\left(\frac{Q_i^j + Q_{i-1}^{j+1}}{2}\right)^{\beta-1} Q_i^j + \Delta t.q}{\frac{\Delta t}{\Delta x} + \alpha\beta\left(\frac{Q_i^j + Q_{i-1}^{j+1}}{2}\right)^{\beta-1}}$$

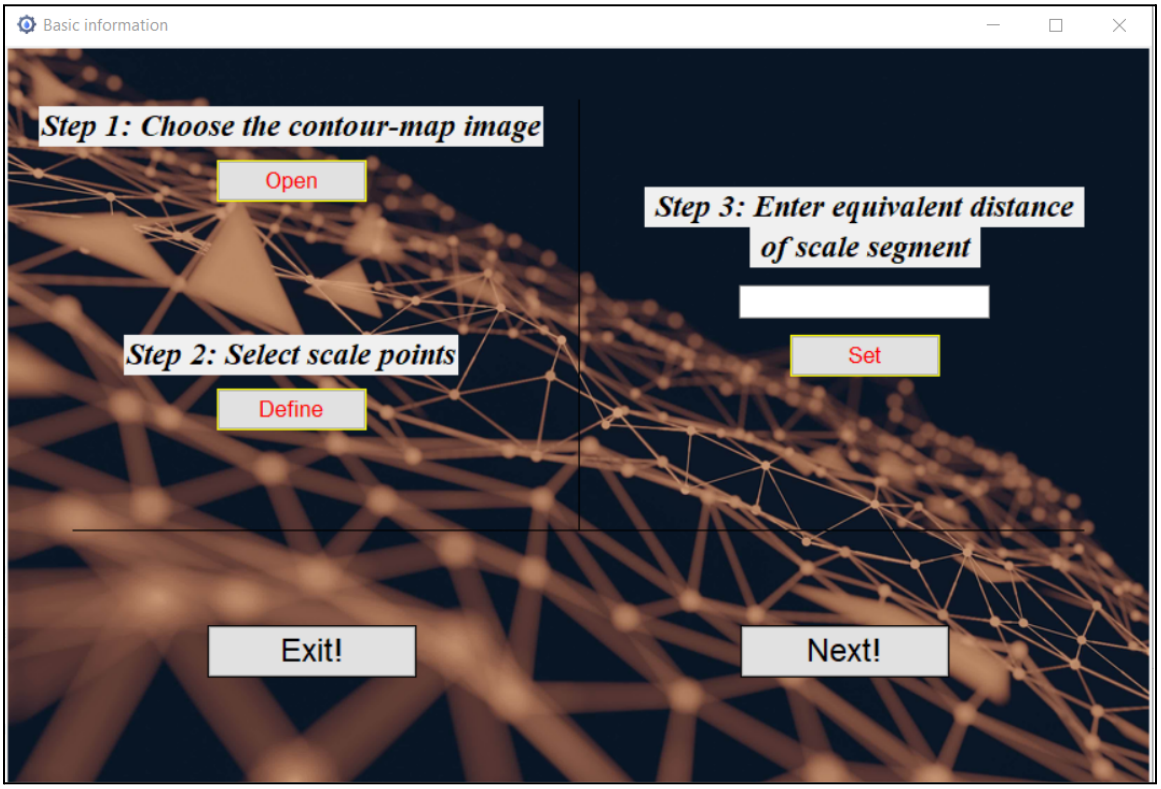
DEMONSTRATION OF GRAPHICAL USER INTERFACE

1. **Landing interface:**

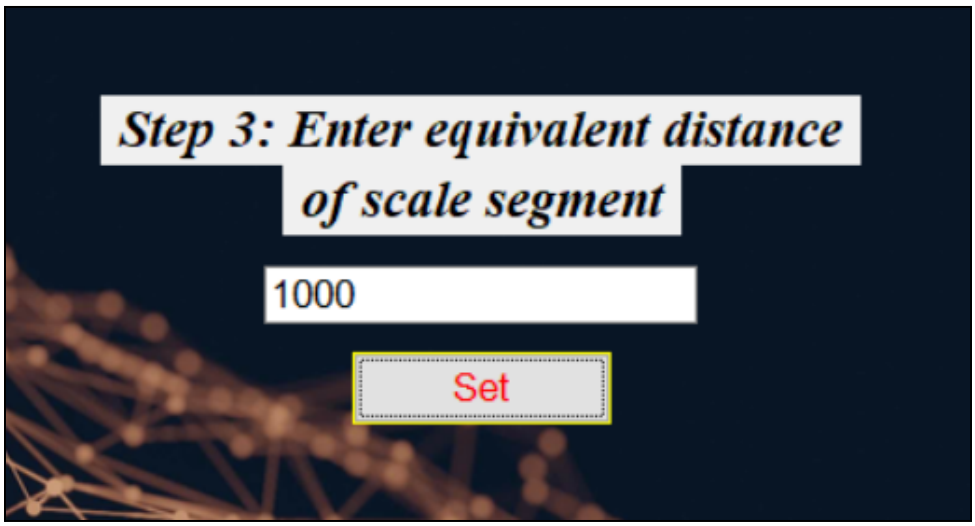


This contains general information and two navigation button. The "Quit!" button terminates the program in case the user wants to exit the application. The "Continue…" button will take the journey to the next page.

2. First page for Initial data collection:

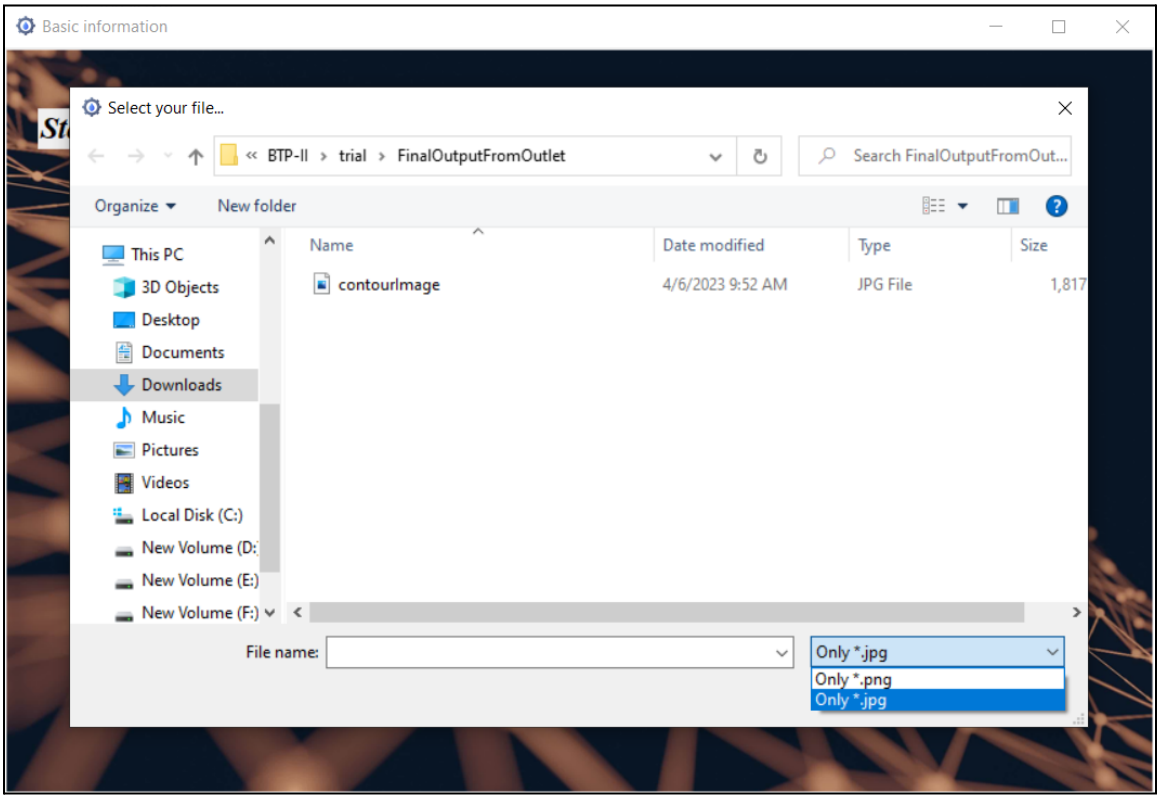


In Step 3, after entering a value of equivalent distance, the "Set" button should be pressed to save the current value input and reject the previous.



2(a). Choose a contour-map image:

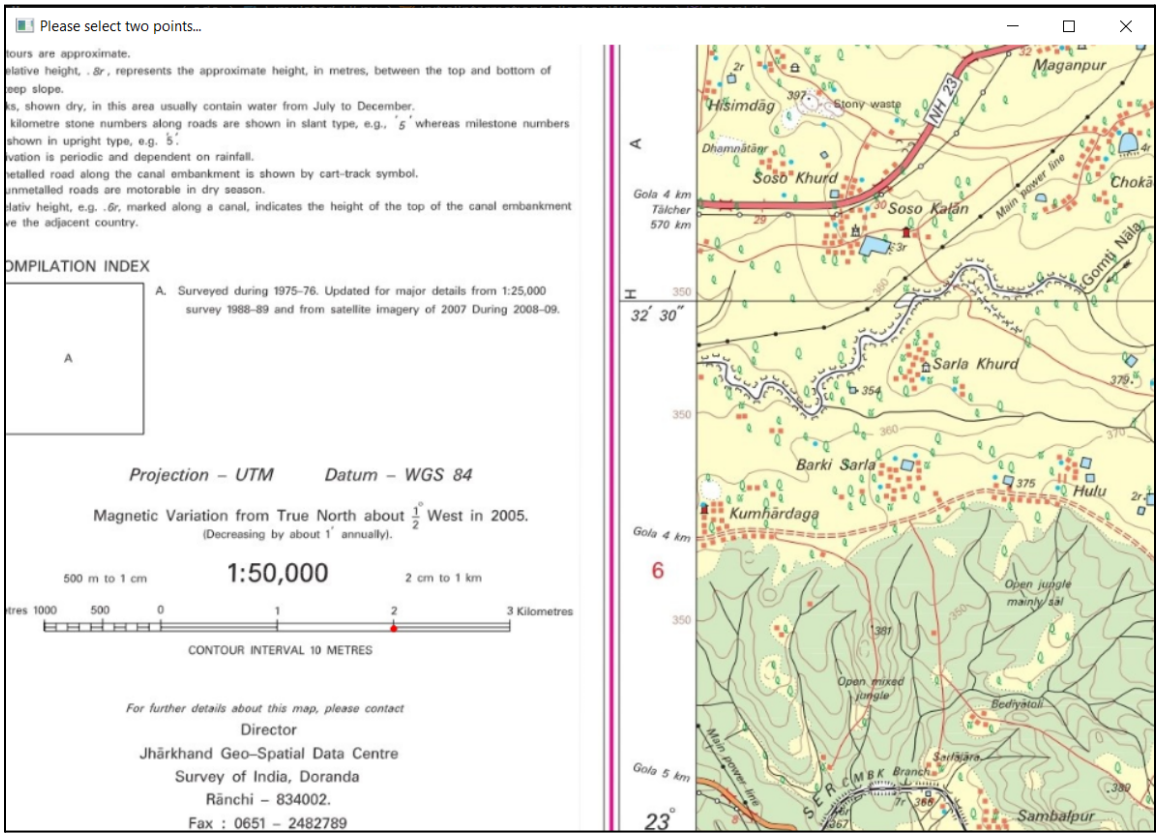
On clicking the "Open" button below Step 1, the following prompt will appear. Here two different image extensions are supported for the input image.



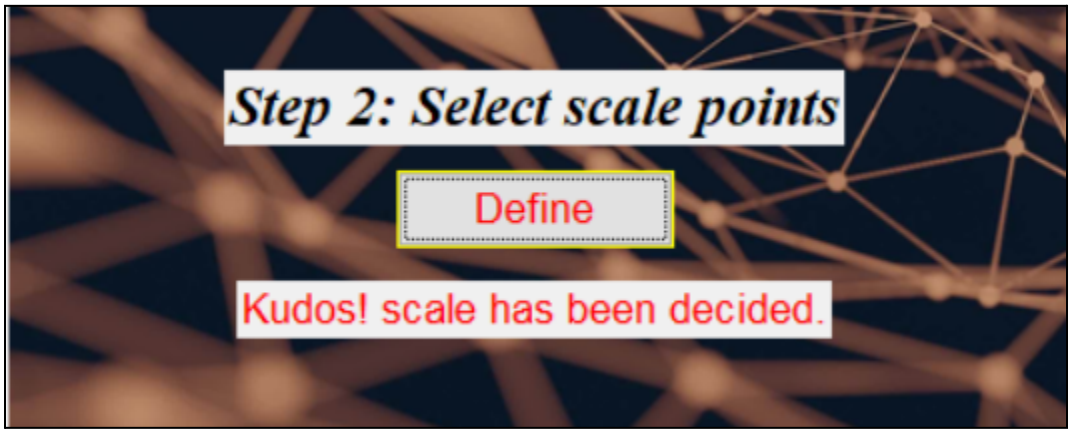
After selecting an image, the selected image's name will be shown,

2(b). Select scale points from the selected contour-map image:

After clicking on the "Define" button below Step 2, OpenCV's image viewer will show the image. The "ZoomInZoomOut" class makes this static window into a zoomable window for accuracy.

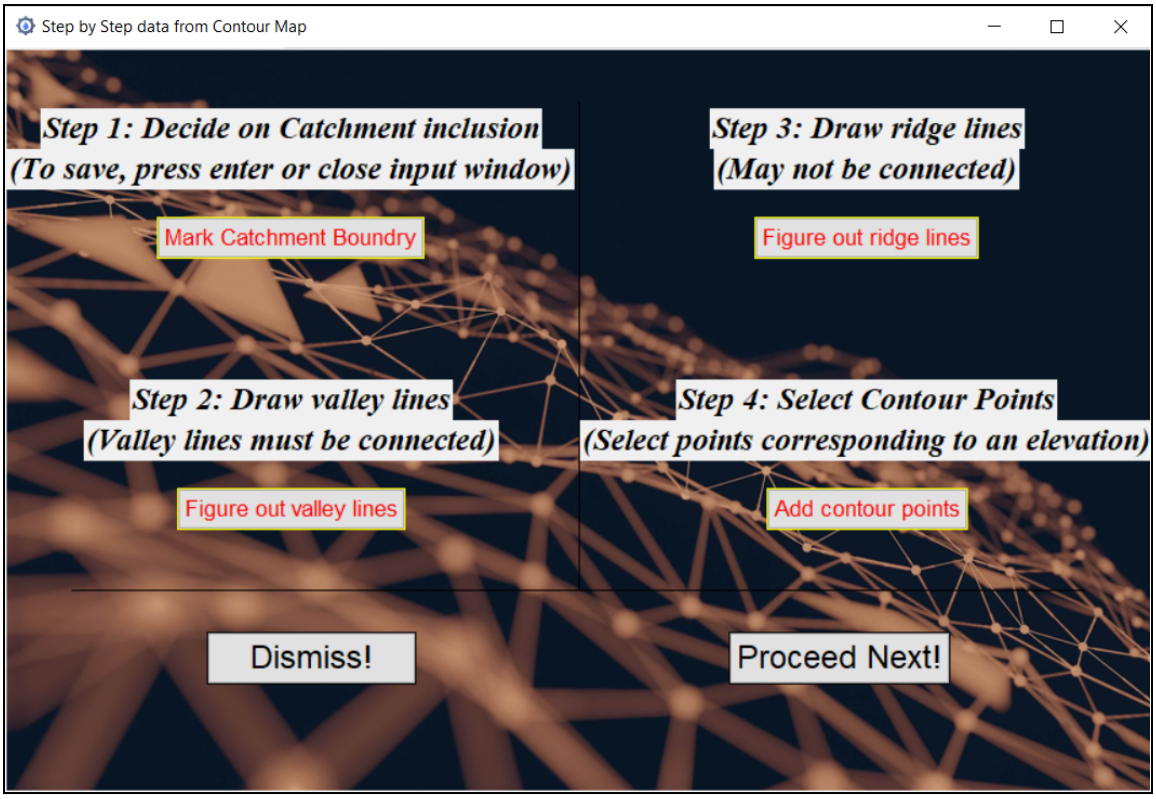


When two points get selected, the prompt will get automatically closed. And a label will be shown if our operation is completed.



3. Detailed inputs related to the geometry of the area of interest:

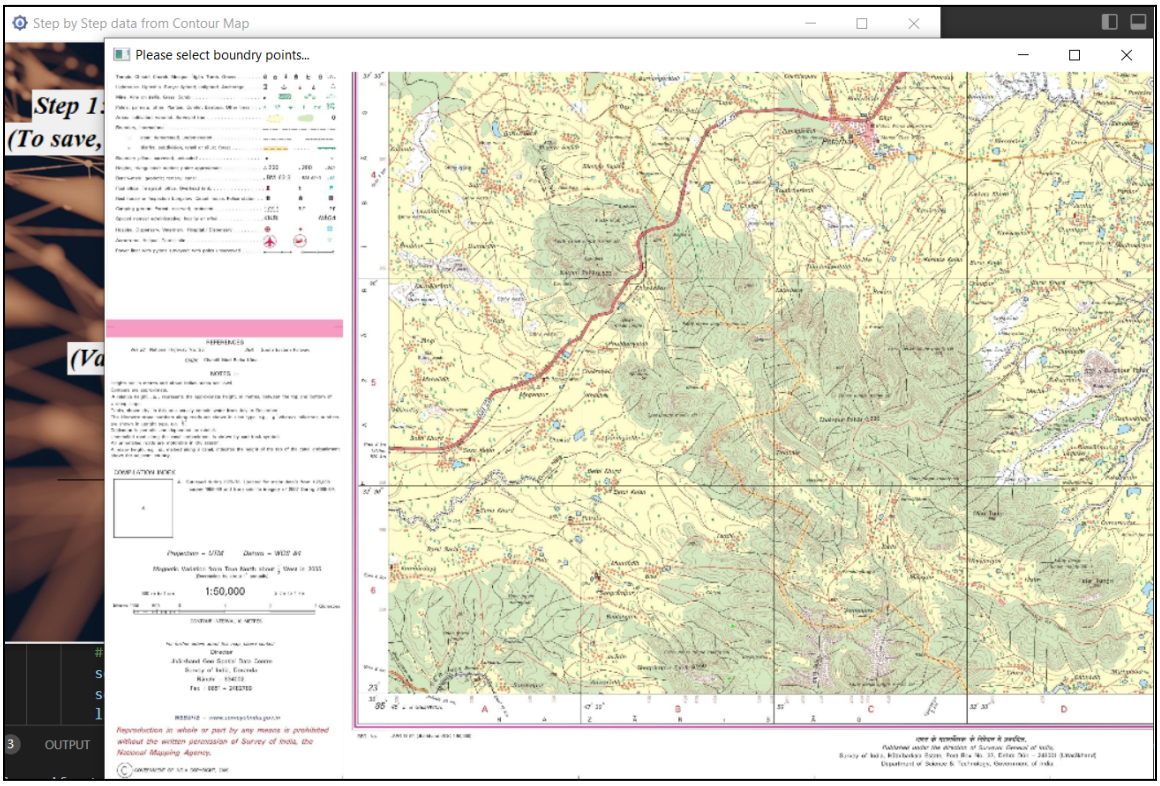
On clicking the “Next!” button on the first page, the following window will appear for taking in-depth data about the geometry of the catchment area.



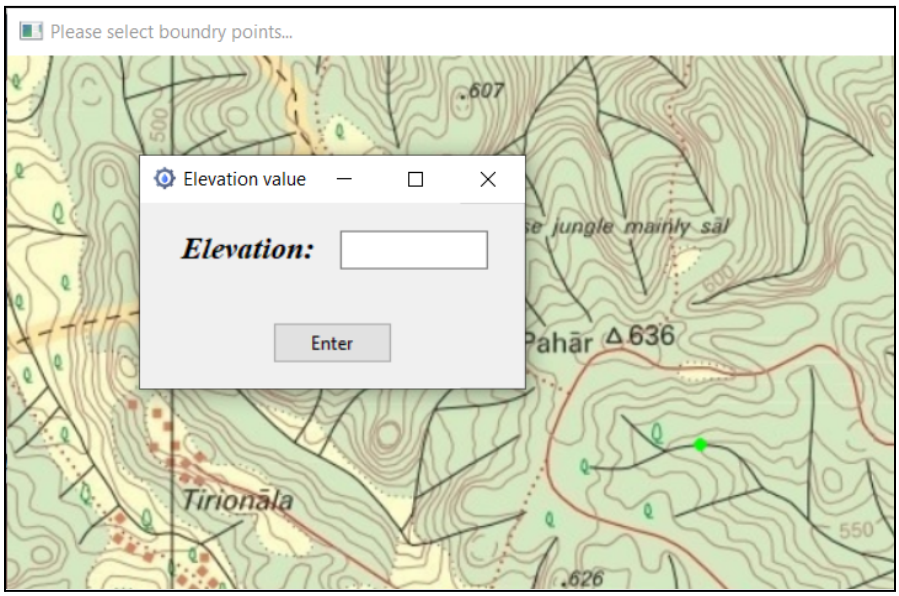
These data will be used for triangulation of the surface. Hence accuracy will highly depend on these data. The catchment boundary defines what area our simulation will work for. Valley lines will collect water from the surrounding area as per geometry. Ridge lines along with catchment boundaries define a boundary for valley segments to collect water from its surrounding. Contour lines formed from collected contour points will refine the geometry of the catchment area.

3(a). Catchment area selection:

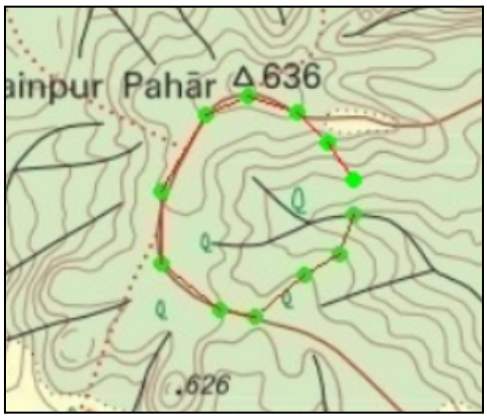
On clicking “Mark Catchment Boundry” button, a prompt will appear,

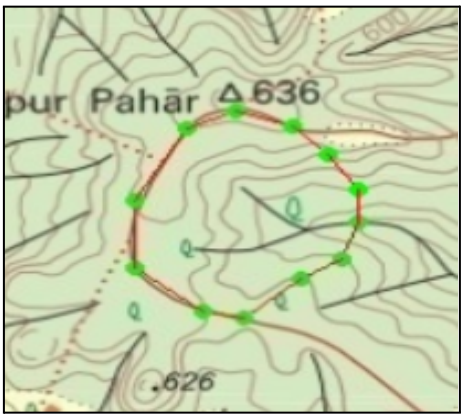


Here on selecting a point on the contour map, a sub-window will appear to take input of elevation value at that point.



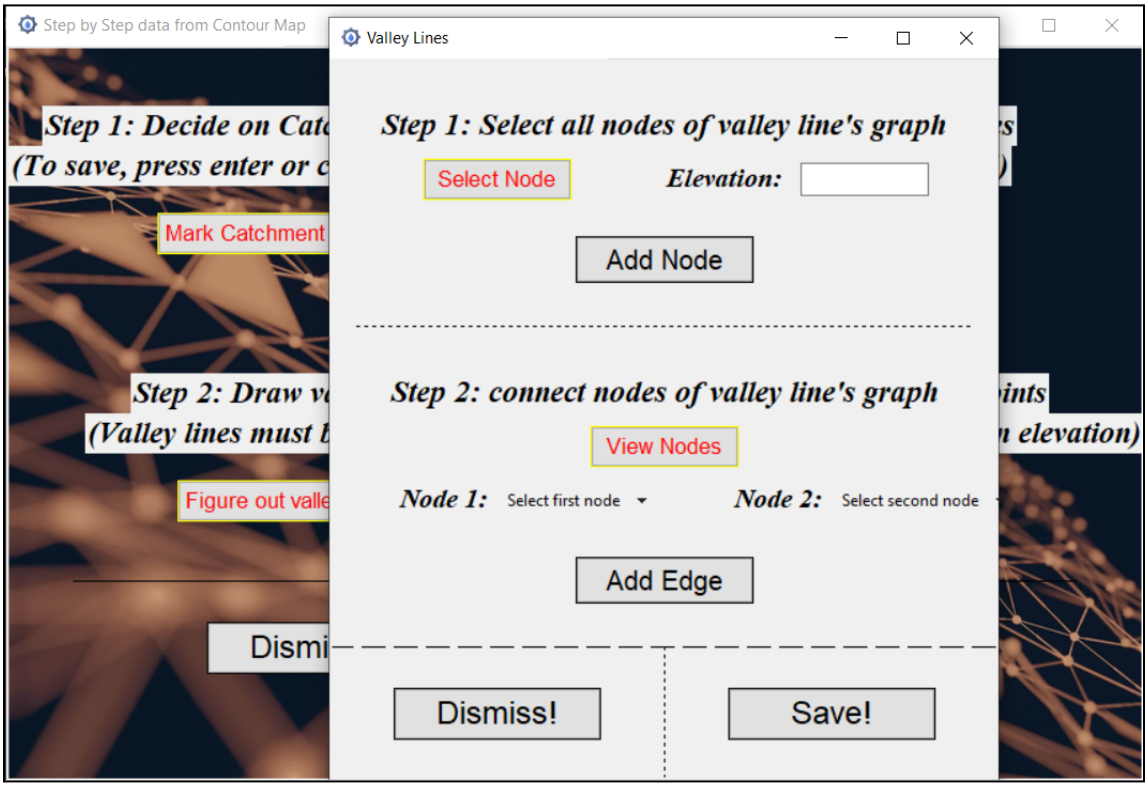
Boundary points should be selected in an ordered way, boundary segments will be generated in the way they got selected. The final segment will be generated between the first selected point and the last selected point. As shown below,



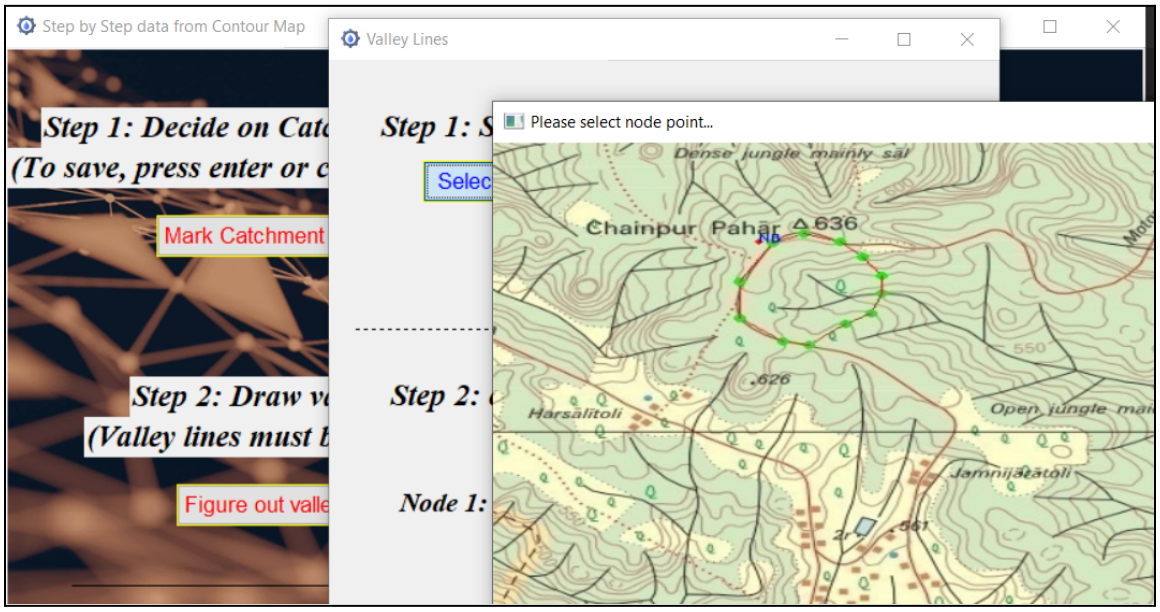


3(b). Mark valley lines from the contour map:

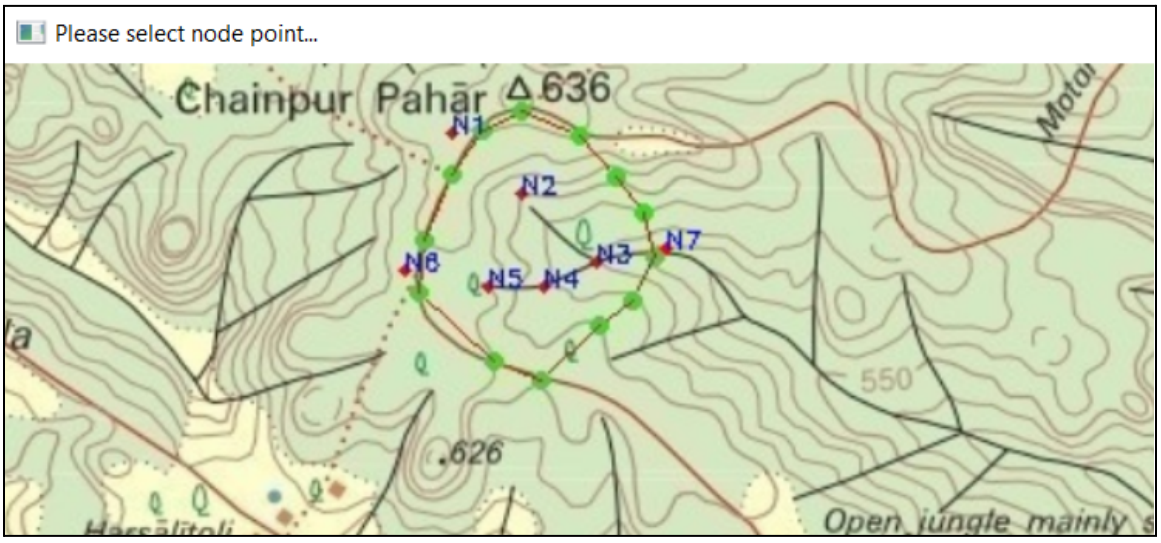
On clicking the "Figure out valley lines" button below Step 2, the following sub-window will appear for inputs,



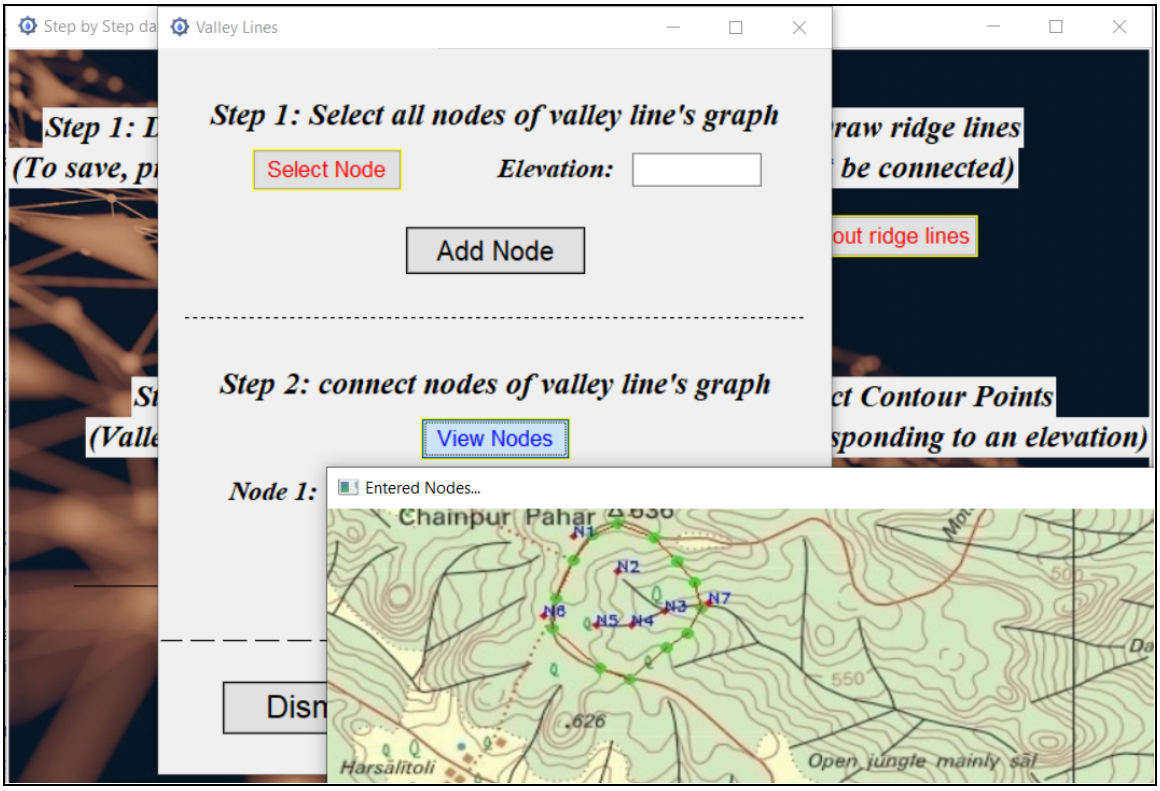
On clicking the “Select Node” button on the Valley Lines named sub-window, the following prompt will appear,



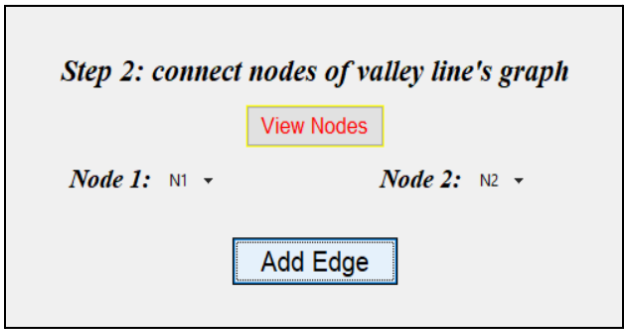
After selecting a point on the contour map, the prompt will automatically get closed. Provide the elevation value corresponding to the selected node point and press the “Add Node” button in Step 1. After this, that node point will appear in the contour map along with its name.

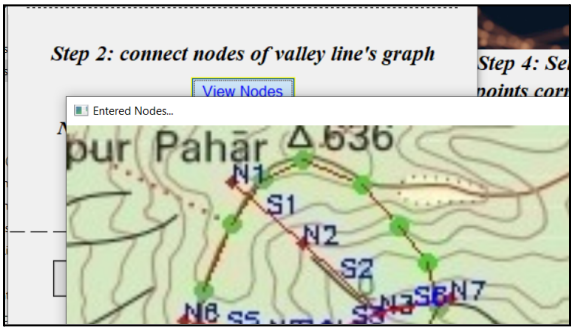


After selecting all valley nodes, Step 2 is for joining these nodes to form the valley segments. “View Nodes” functionality is provided to look at the previously selected nodes to join them accurately.



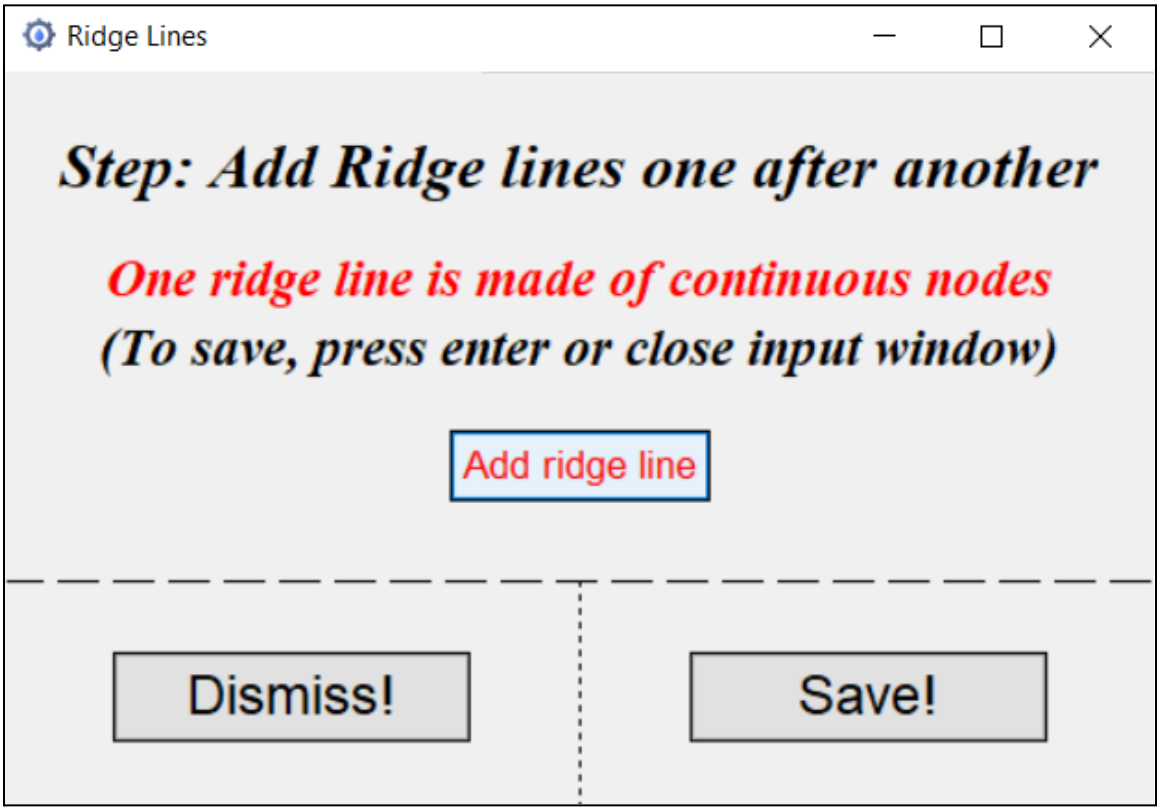
Select nodes from the list and press the “Add Edge” button to generate a valley segment. For identification purposes, the segment got its serial number on the line.



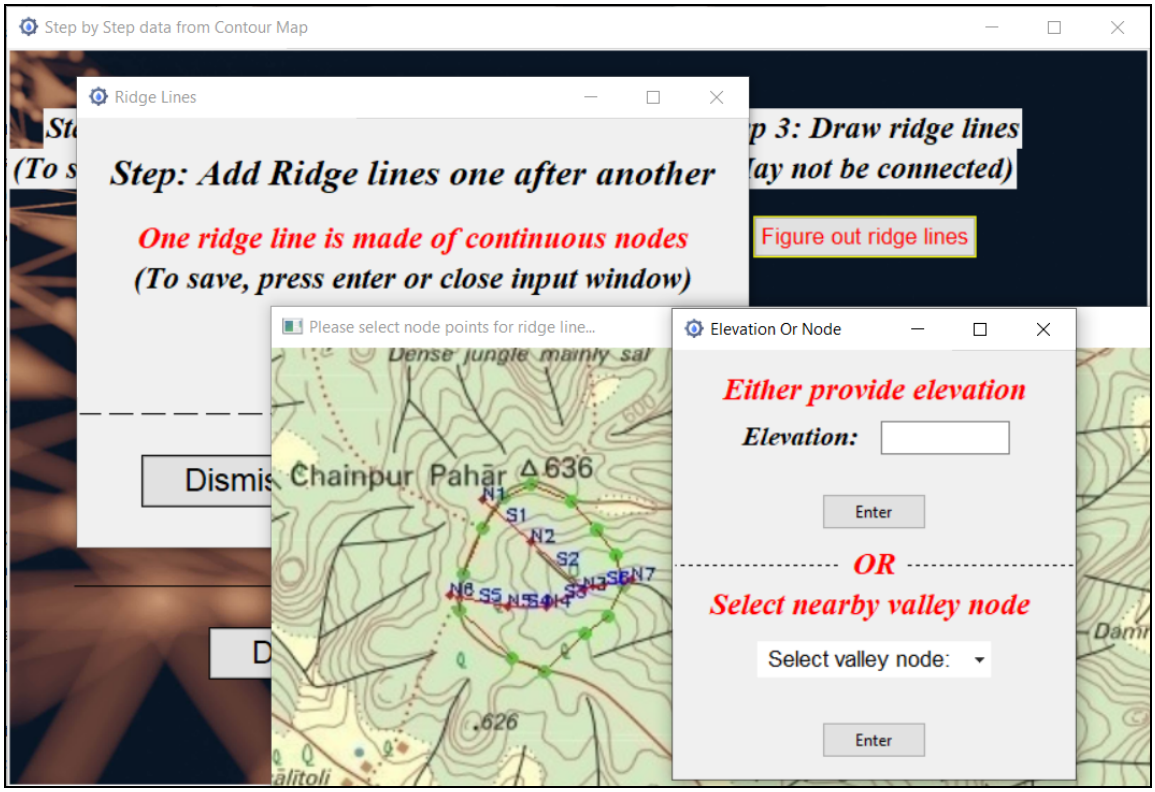


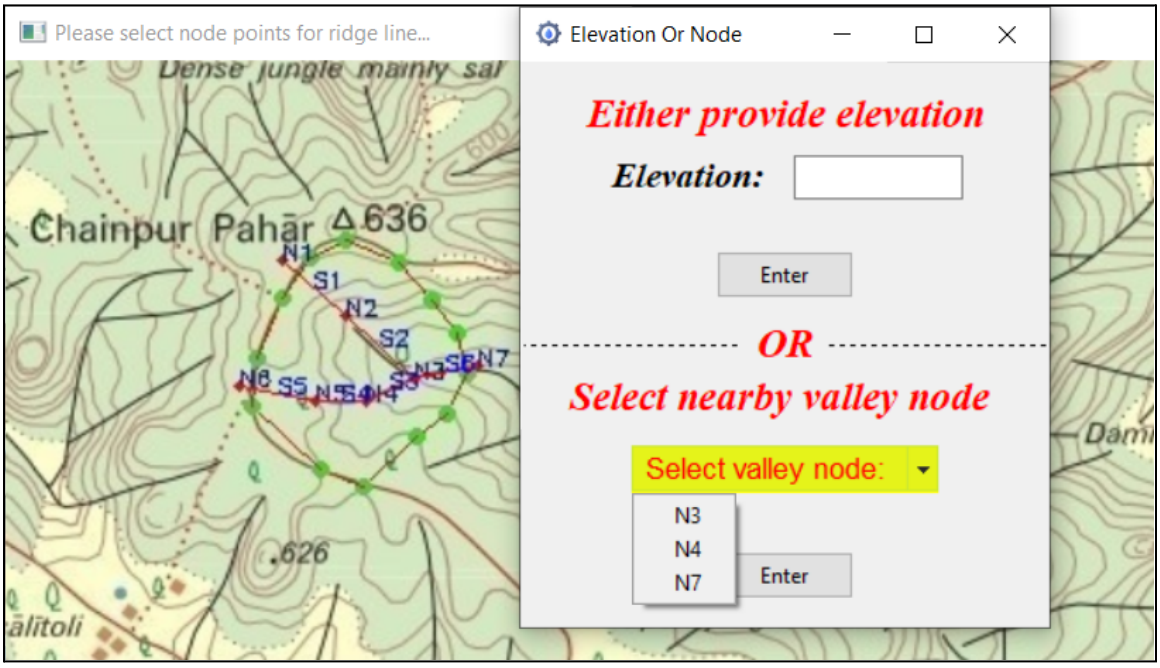
3(c). Mark ridge lines from the contour map:

After clicking the "Figure out ridge lines" button, the below sub-window will appear for the ridge line's node points,

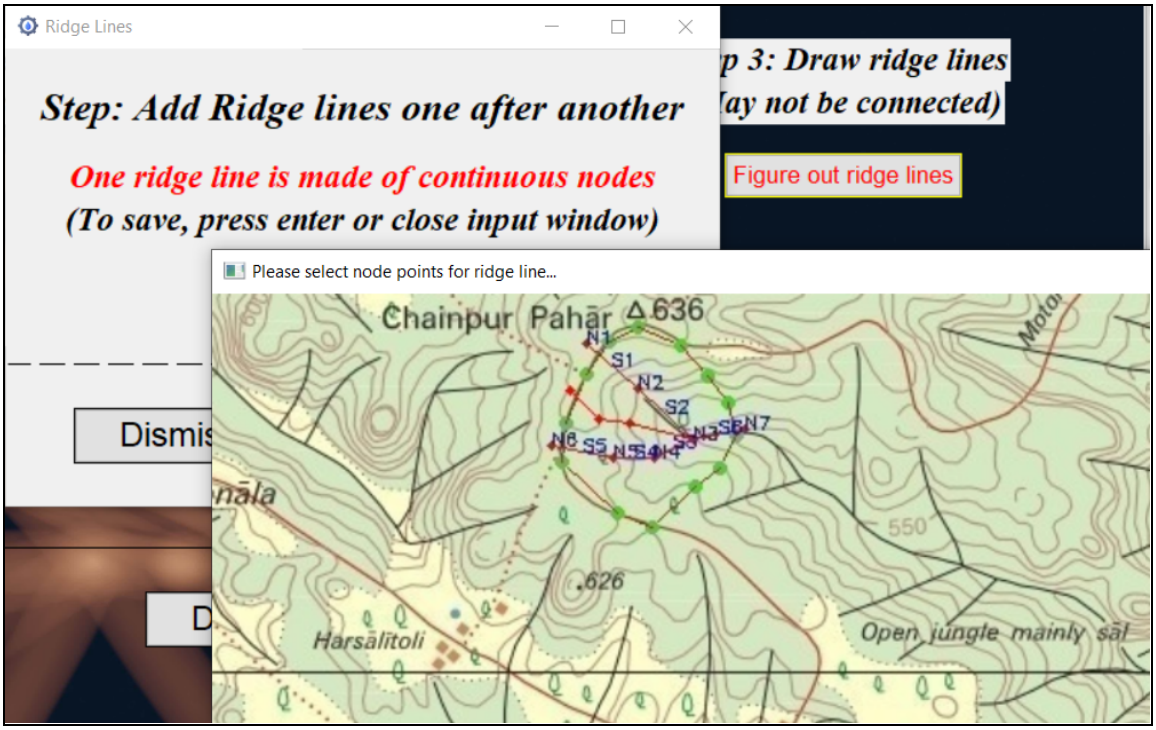


On clicking the "Add ridge line", the OpenCV window will appear for selecting ridge nodes lies on same ridge line.
On selecting a point on the OpenCV window, one more sub-window will appear, which demands either the elevation value Or selecting a valley node that exists on the contour from the list of three valley nodes. These three nodes will be selected on the basis of the closest distance between the selected point on the contour map and the valley nodes. In this case, it is not required to put an elevation value.


Step by Step data from Contour Map
Ridge Lines
Step: Add Ridge lines one after another
One ridge line is made of continuous nodes
(To save, press enter or close input window)
Dismis
p 3: Draw ridge lines
ay not be connected)
Figure out ridge lines
Please select node points for ridge line...
Chainpur Pahār Δ636
Elevation Or Node
Either provide elevation
Elevation:
Enter
OR
Select nearby valley node
Select valley node:
Enter

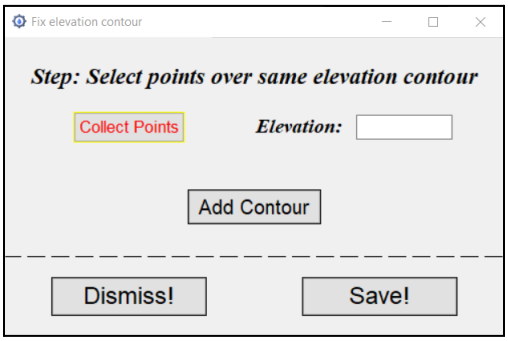

Please select node points for ridge line...
Chainpur Pahār Δ636
Elevation Or Node
Either provide elevation
Elevation:
Enter
OR
Select nearby valley node
Select valley node:
N3
N4
N7
Enter

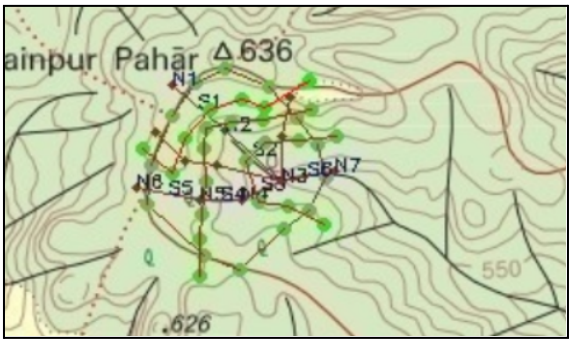
Valley segments are formed by joining the continuous nodes that are entered through the OpenCV window. Valley segments can be noticed as they got formed by joining the current node to the previous node.



3(d). Add contour lines from the contour map:

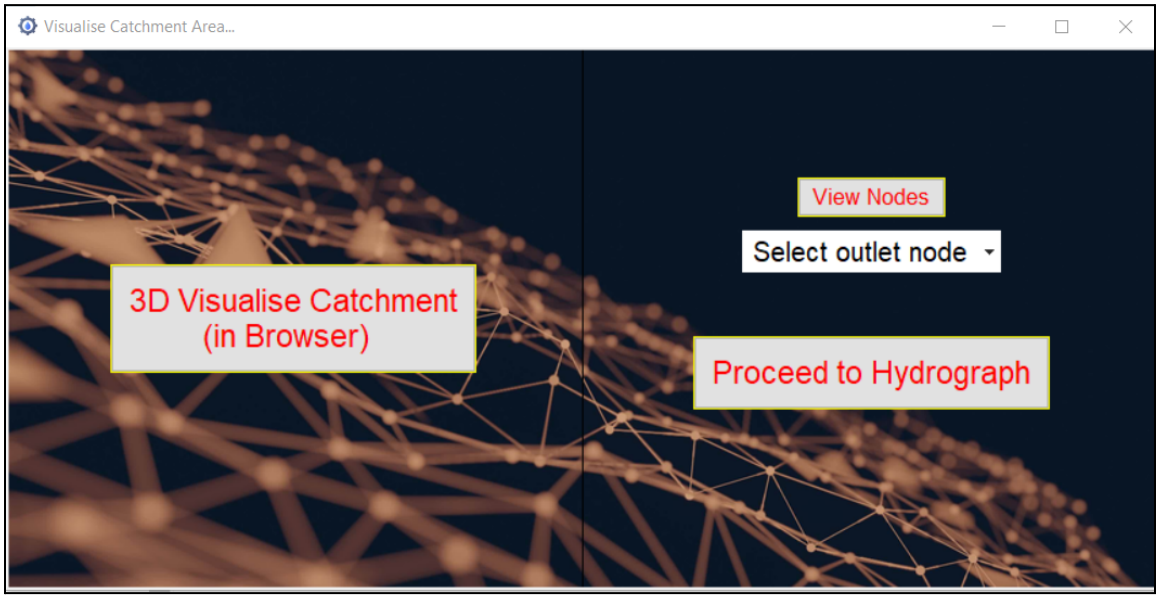
After clicking “Add contour points” button, the following sub-window will appear for taking points on the contour map corresponding to a fixed contour line. Then corresponding elevation value should be provided, and press “Add Contour” button on the sub-window.



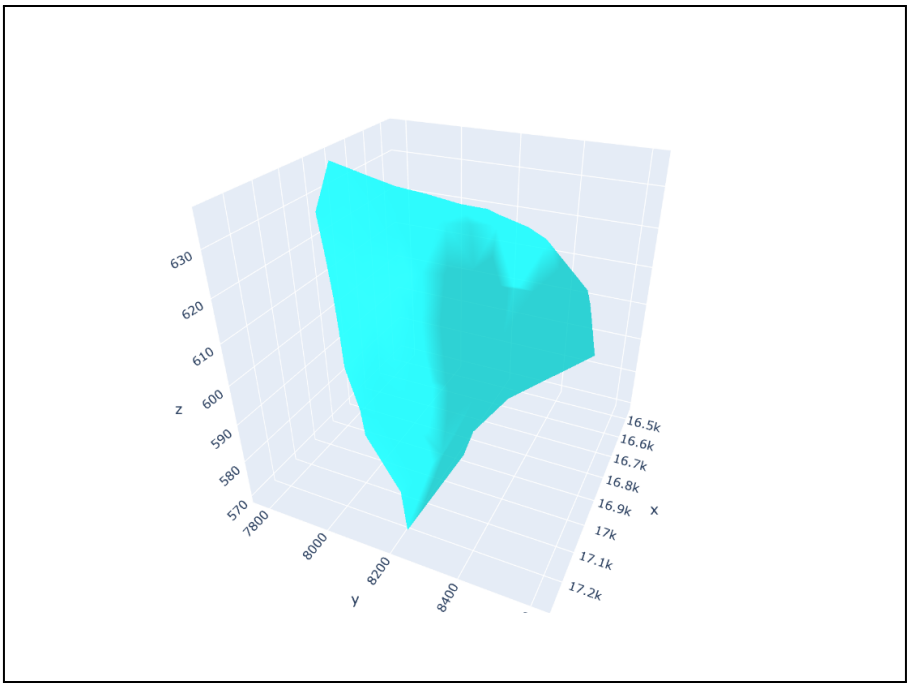


4. Visualise catchment geometry and Select outlet node:

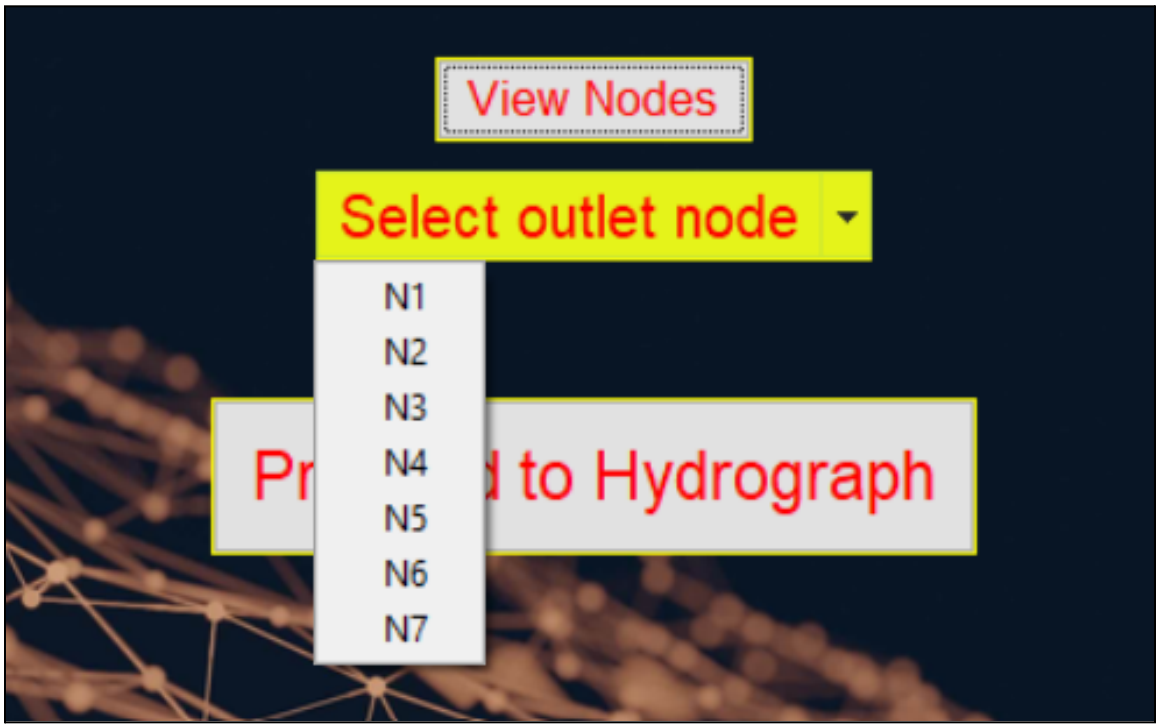
On clicking “Proceed Next!” button, the following window will appear.



After triangulation of nodes and segments inputs, the three-dimensional geometry of the catchment can be visualized in the browser after clicking “3D Visualised Catchment (in Browser)”.

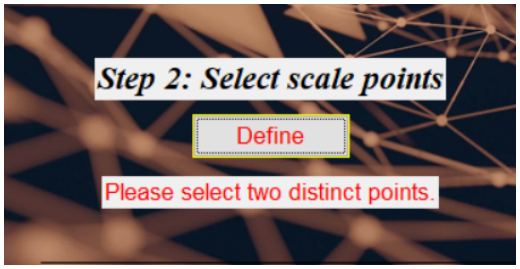


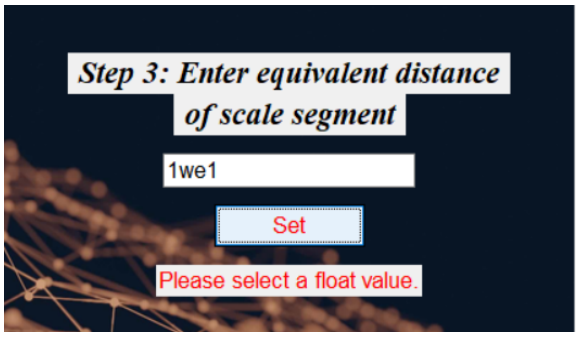
"View Nodes" can be used for recalling the potential valley nodes which are candidates for an outlet node. Select an outlet node from a list of valley nodes. And click on "Proceed to Hydrograph" to land on the final page where the peak discharge value and corresponding time when it occurred could be found along with the hydrograph.

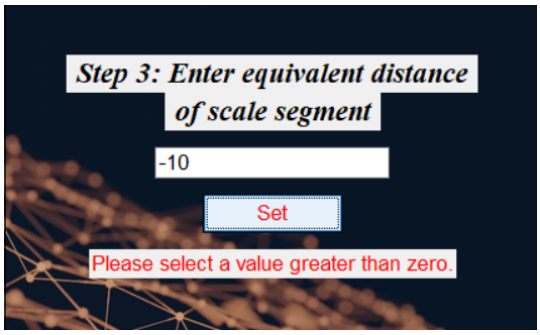


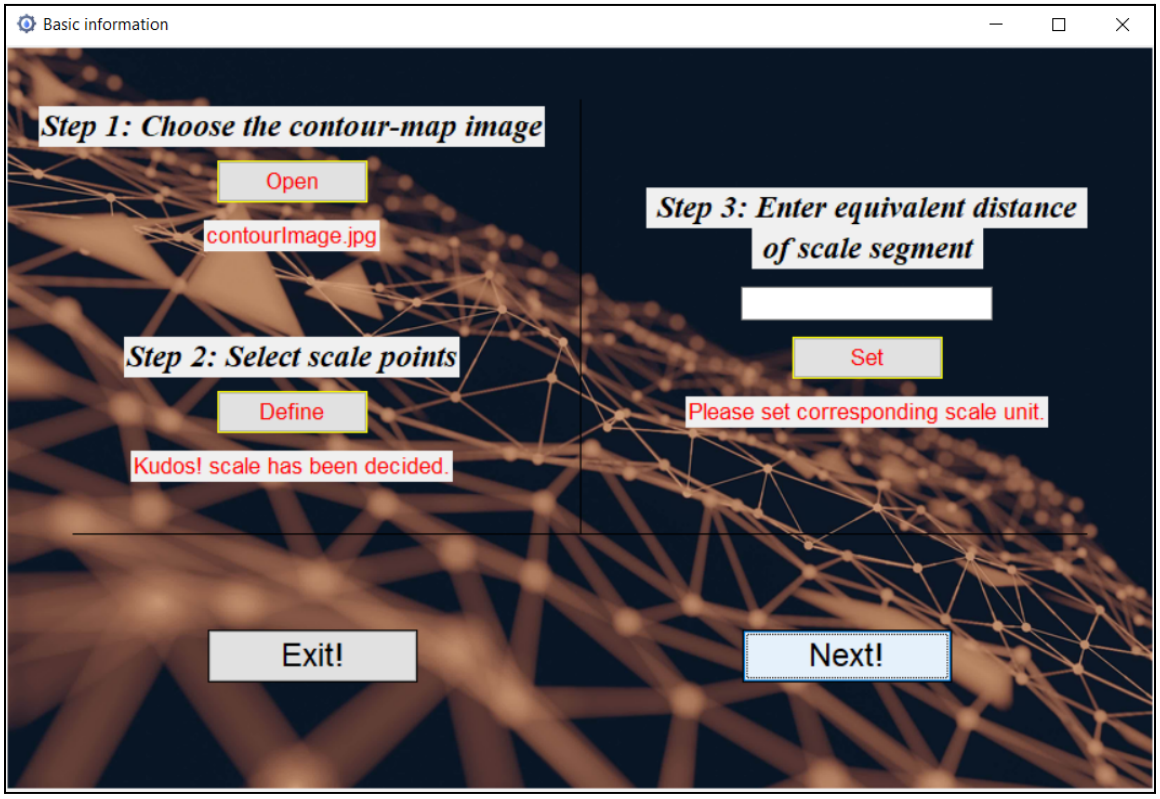
5. Common Errors handled during implementation:

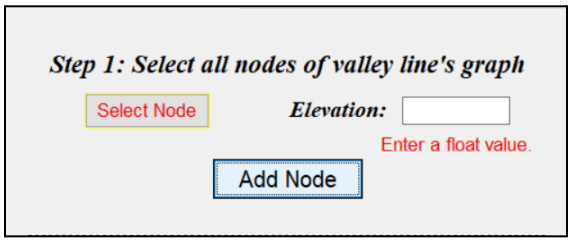
The label shows a particular error that the user knowingly or unknowingly has made while giving inputs to the simulator.

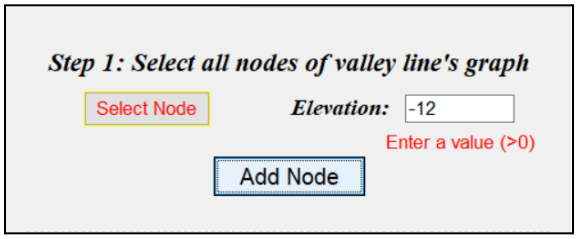


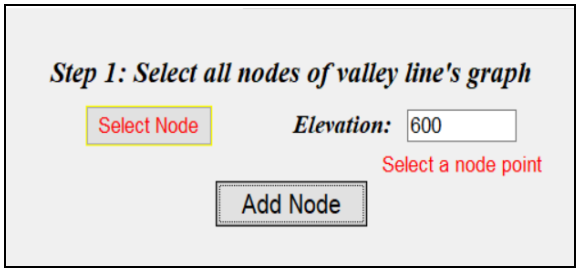

Step 3: Enter equivalent distance of scale segment
1we1
Set
Please select a float value.

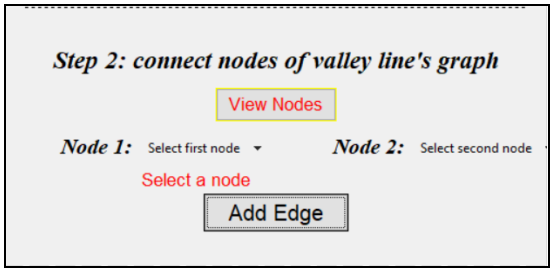

Step 3: Enter equivalent distance of scale segment
-10
Set
Please select a value greater than zero.

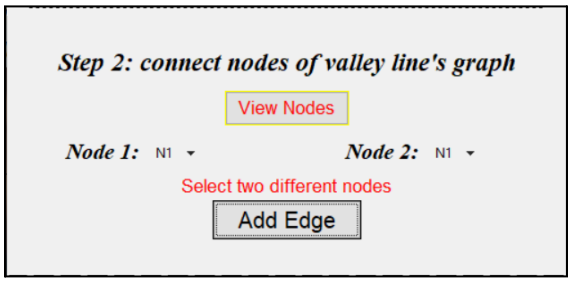

Basic information
Step 1: Choose the contour-map image
Open
contourImage.jpg
Step 2: Select scale points
Define
Kudos! scale has been decided.
Step 3: Enter equivalent distance of scale segment
Set
Please set corresponding scale unit.
Exit!
Next!

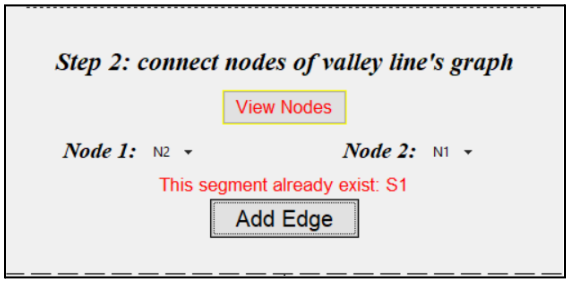

Step 1: Select all nodes of valley line's graph
Select Node
Elevation:
Enter a float value.
Add Node

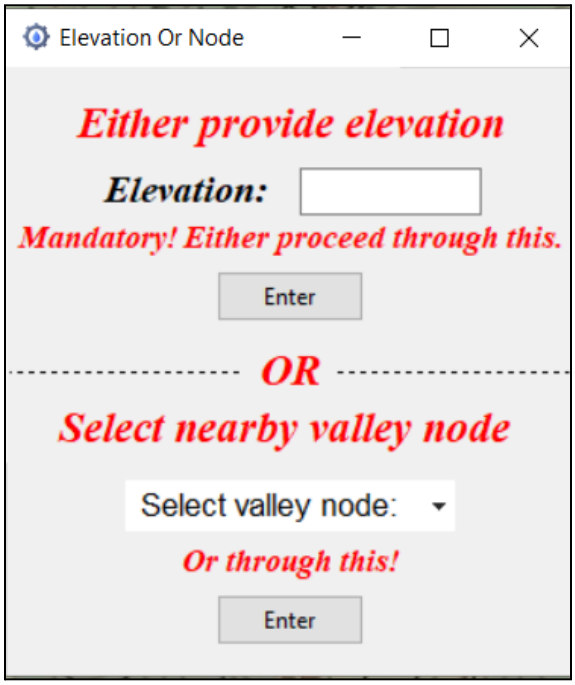

Step 1: Select all nodes of valley line's graph
Select Node
Elevation:
-12
Enter a value (>0)
Add Node

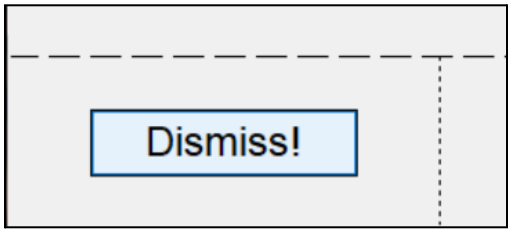








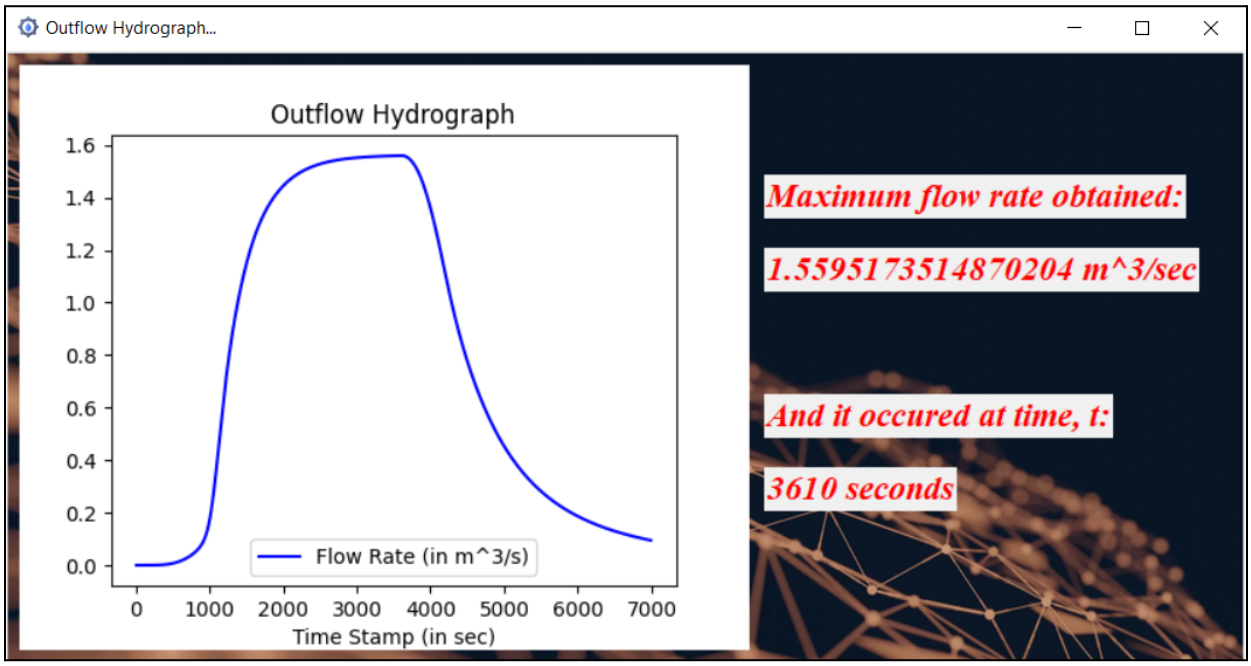




The “Dismiss!” button is there in catchment area selection, valley segments, ridge segments, and contour segments. In case a mistake is made during input, that portion could be rejected. So it is advised to save work after small tasks so that the error portion can be dismissed.

RESULTS

Below is the final output hydrograph representing the final outflow from the outlet point:



Theoretical calculations:

Area of the vertical projection of the catchment area = 592700 m^2

Rainfall intensity = 1 cm/hour

= 1 / (100x60x60) meter/sec

= 2.778 x 10^{-6} m / sec

Outflow during the direct runoff = Area x Rainfall intensity

= 0.59727 x 10^6 x 2.778 x 10^{-6} m^3 / sec

= **1.646** m^3 / sec

Peak occurred on 3610 seconds after the start of rainfall which is reasonable matching theoretical value because unit rainfall stops at 3600 seconds.
Hence, the results are matched with good accuracy.

DISCUSSION AND CONCLUSIONS

In-accuracy in the calculation may be due to factors like less refinement in the cross-section during kinematic wave approximation; during triangulation, there may be the possibility that some triangles didn't get properly formed, i.e., all three nodes of a triangle got on same contour line or valley line or ridge line; there may be the possibility that during entry of elevation value, the incorrect value provided or point location was not accurate, etc. So if human errors could be eliminated, there would be more accurate results. Refinement of the cross-section could be made in proportion to the corresponding length of the valley segment. Triangulation errors could be handled efficiently by selecting a more number of the points along the contour line and placing ridge lines accurately in combination with valley lines.

Future prospects for enhancement of the current version are the following:

1. Current model assumes that the surface is impervious and there is no evaporation during runoff. But in a real scenario, there would be infiltration, evaporation, base flow, etc. Hence appropriate factors of reduction or addition should be implemented.
2. Data could be saved in a given location and then could be reused for a given geometry and outflow from different outlets. This will save computational power. Hence, at last, there could be an option for data storage, and on the front page, there could be an option for reusing previous data.
3. Flow through valley segments can be improved by implementing more accurate methods like the preissmann scheme, which includes the velocity of the flow hence if the flow has some velocity, it would pass through a cross-section where it has no slope.

REFERENCES

1. Daniel Caviedes-Voullième, Javier Fernández-Pato, Christoph Hinz. 2020. "Performance assessment of 2D Zero-Inertia and Shallow Water models for simulating rainfall-runoff processes." Journal of Hydrology. 584 (2020) 124663
2. B.C Yen, C.W.-S Tsai. 2001. "On noninertia wave versus diffusion wave in flood routing." Journal of Hydrology. 244 (2001) 97-104
3. Svein Linge, Hans Petter. 2016. Programming for Computations - A Gentle Introduction to Numerical Simulations with Python. Southeast Norway
4. M. Hanif Chaudhry. 2008. Open-Channel Flow. Main Street Columbia.

APPENDIX

- Link to full code: https://github.com/jangidashok/Hydrograph-Simulator.git

- Working presentation: https://drive.google.com/drive/folders/1VX98eFx112T9eA3aFvxe3PAjqn_KJBjs?usp=share_link