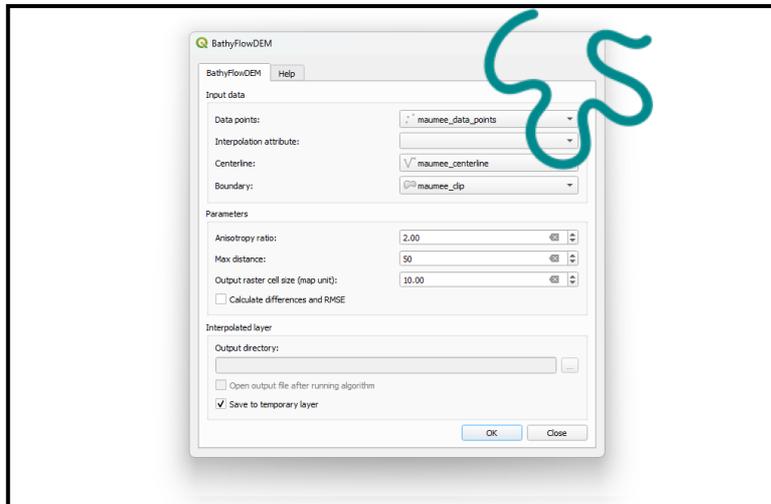


BathyFlowDEM: development of a QGIS plugin to interpolate bathymetric data in flow-oriented coordinate system

Melinda Femminis

Under the supervision of Dr. Christian Kaiser



BathyFlowDEM:
Development of a QGIS plugin to interpolate bathymetric
data in flow-oriented coordinate system

Melinda Femminis

Spring semester 2024

Abstract

BathyFlowDEM is an open-source QGIS plugin designed to interpolate bathymetric data with anisotropic considerations. It is fully integrated into QGIS and doesn't rely on external Python libraries.

The tool first converts input data points to a (s,n) flow-oriented coordinate system, also called curvilinear orthogonal coordinate system, optimizing the spatial context for interpolation. It then generates new data points whose depth values are interpolated using Inverse Distance Weighting (IDW) with directional anisotropy and creates a raster layer output representing the riverbed topography.

This approach aims to enhance the quality of the river surfaces generated from bathymetric input data points.

The parameters that are user-defined are the anisotropy ratio, final raster cell size, region of interest and IDW's search radius. Model validation functionalities are also available directly within the plugin.

Plugin's repository: <https://github.com/melindafemminis/BathyFlowDEM>

Keywords: Spatial interpolation, anisotropic interpolation, directional variability, riverbed modelling, geospatial analysis, bathymetry

Résumé

BathyFlowDEM est un plugin open-source pour QGIS conçu pour l'interpolation anisotrope de données bathymétriques. Il est entièrement intégré à QGIS et ne dépend pas de bibliothèques Python externes.

L'outil convertit tout d'abord les points connus dans un système de coordonnées (s,n) orienté dans le sens d'écoulement de la rivière, optimisant ainsi le contexte spatial pour l'interpolation. Puis il génère de nouveaux points dont la valeur à définir est interpolée à l'aide de la méthode de pondération inverse à la distance (IDW) avec une anisotropie directionnelle. Il crée ensuite une couche raster comme résultat représentant la topographie du lit de la rivière.

Cette approche vise à améliorer la qualité et précision du résultat final par rapport à d'autres méthodes d'interpolation ne prenant pas en compte l'anisotropie.

Le ratio d'anisotropie, la taille des cellules du raster, la région d'intérêt et le rayon de recherche de l'IDW sont définis par l'utilisateur. Des fonctionnalités de validation du modèle sont également disponibles directement dans le plugin.

Repository du plugin: <https://github.com/melindafemminis/BathyFlowDEM>

Mots-clés : Interpolation spatiale, interpolation anisotrope, variabilité directionnelle, modélisation de rivière, analyse géospatiale, bathymétrie

Acknowledgements

I would like to express my gratitude to my supervisor Christan Kaiser, Senior Lecturer at the Institute of geography and sustainability in the University of Lausanne for their guidance and support throughout this journey. Their insights and comments have contributed to shaping my thesis.

A special thank you to Eduardo my husband for his support, love and understanding during this challenging period. Your encouragements and your enthusiasm for listening to all my ramblings made it possible.

Thank you to Baptiste and Max for the lighthearted moments, the study sessions and in this work for your valuable feedback.

Contents

List of Figures

List of Tables

1	Introduction	1
1.1	What is bathymetry	1
1.2	Data acquisition with an echosounder	2
1.3	Bathymetric data and processing	3
1.4	Problem definition	4
1.5	Research question	4
2	Literature review	5
2.1	Interpolation methods	5
2.1.1	TIN interpolation	5
2.1.2	Nearest Neighbor and k-Nearest Neighbors	5
2.1.3	IDW	7
2.1.4	Elliptical Inverse Distance Weighting	8
2.1.5	Kriging model	9
2.1.6	Local Anisotropy Kriging	11
2.2	RMSE	11
2.3	Existing tools	11
2.3.1	ArcGIS Bathymetry Interpolation plugin	12
2.3.2	Tatooine Mesher	12
2.3.3	ANUDEM	13
3	Coordinate systems	14
3.1	Cartesian and flow-oriented coordinate systems	14
3.2	Coordinate transformation in BathyFlowDEM	15
4	Interpolation method	17
4.1	IDW with anisotropy	17
4.2	IDW with anisotropy in BathyFlowDEM	17
5	Tools	19
5.1	QGIS	19
5.2	Python and PyQGIS	20
5.3	Qt Designer	20
5.3.1	Widgets in Qt Designer	21
5.3.2	Signals and slots	22
5.4	Other useful tools for development	23
5.4.1	Plugin Builder	23
5.4.2	Plugin Reloader	23
6	Methodology	24
6.1	Definition of requirements and implementation of version 0.1	24
6.2	Definition of requirements and implementation of version 0.2	26
6.3	User tests	26
6.4	Git environment	27
7	BathyFlowDEM plugin	28
7.1	Plugin's functionalities	28
7.2	Modules and directory structure	30
7.3	Licence and version	30

8 Usage	32
8.1 Installation	32
8.2 Datasets	32
8.3 Using BathyFlowDEM: a practical example	34
9 Discussion	37
9.1 Evaluation of the plugin	37
9.1.1 Functionality	37
9.1.2 Usability	37
9.1.3 Integration to Quantum Geographic Information System (QGIS)	38
9.1.4 Performance	39
9.2 Impact and application	40
9.2.1 Limitations	41
9.2.2 Future developments	42
10 Conclusion	44

List of Figures

1	Example of a TIN surface made of triangles.	6
2	Two dimension illustration of the Nearest Neighbor and k-Nearest Neighbors interpolation algorithms. The new point X will take the value of the closest known point (circle) with NN and the average of the circle and three triangles with kNN where $k=4$	6
3	Two dimension Voronoi diagram illustrating the partitioning of the space based on the distance to the data points (black dots). Each color represents a point's region of influence.	7
4	Step-by-step of the IDW algorithm.	8
5	IWD with a circular neighborhood (A) and an elliptical neighborhood (B). Z is the unsampled point.	8
6	Modified distance of data points within the search radius	9
7	Example of a semivariogram	10
8	Theoretical bathymetric data represented in A) a Cartesian coordinate system where $p_i = (x_i, y_i)$ and B) in a flow-oriented coordinate system where $p_j = (s_j, n_j)$. The centerline and flow direction is represented by the blue line.	14
9	Two data points in a Cartesian coordinate system with corresponding flow-oriented coordinates. Includes the centerline and both banks.	15
10	The data points selected by the search radius before and after applying the anisotropy ratio to make points in the n-axis more distant.	18
11	A possible QGIS layout with the map canva in the middle, the processing toolbox on the right, and file browser on the top left and the layers loaded the project on the bottom left.	20
12	QgsMapLayerComboBox and QgsFieldComboBox widgets in the QGroupBox, with their respective label. Part of BathyFlowDEM's Graphical User Interface (GUI)	21
13	Example of a QgsDoubleSpinBox and QCheckBox widgets in the GUI of BathyFlowDEM.	22
14	The saving layout of "Clipped", a native QGIS algorithms.	22
15	The saving layout of the BathyFlowDEM plugin	22
16	The process of creating a new plugin boilerplate with the Plugin Builder	23
17	Main GUI of BathyFlowDEM's minimal working version.	25
18	The tooltip that appears when hovering over the "Search radius" spinbox.	27
19	BathyFlowDEM's issues in the Github repository.	27
20	Pipeline of BathyFlowDEM process with user input data in green, plugin processing steps in blue and outputs in purple. Dashed line means optional.	28
21	The interpolation algorithm will use all input data points that fall within the defined search radius - even when outside the boundary polygon.	29
22	The RMSE value displayed after processing in the GUI below the <i>Parameters</i> box.	29
23	The QGIS plugin manager.	32
24	Show the plugins folder location in explorer for the current user profile.	33
25	Accessing BathyFlowDEM through the menu	33
26	Accessing BathyFlowDEM through the toolbar icon	33
27	Bathymetric cross-sections and stream-wise data points of part of the Maumee River. Background Google Satellite 2024	34
28	Plugin's main tab	35
29	Plugin's help tab	35
30	Comparison of four algorithms used to compute a raster of 60x60m for the Maumee river.	38
31	A warning message shown in the QgsMessageBox within the plugin if the user tries to start processing with conflicting saving choices.	39
32	QGIS processing algorithms saving options	39
33	BathyFlowDEM's saving options	39
34	N is defined as the shortest distance to the centerline. Here d1 will be saved instead of the correct distance, d2.	42

List of Tables

1	Semivariogram models	10
2	The <code>eidw()</code> function's inputs and output	18
3	Minimal requirements of the first version	25
4	Complete requirements of the second version	26
5	Datasets available in BathyFlowDEM/data	34
6	Input data and parameters of BathyFlowDEM	36
7	Comparison of processing time and rmse with spatial index and without	40

Listings

- 1 Computing the s and n distances in the `eidw()` function 18
- 2 Connecting the signals and slots in the `self.initGUI()` method in `BathyFlowDEM` class 23

Acronyms

ANUDEM Australian National University Digital Elevation Model.

API Application Programming Interface.

CSV Comma Separated Values.

DEM Digital Elevation Model.

EIDW Elliptical Inverse Distance Weighting.

GDAL Geospatial Data Abstraction Library.

GIS Geographic Information System.

GNSS Global Navigation Satellite System.

GRASS Geographic Resources Analysis Support System.

GUI Graphical User Interface.

IDW Inverse Distance Weighting.

kNN k-Nearest Neighbors.

LAK Local Anisotropy Kriging.

LTR Long Term Release.

MBES Multi Beam Echo Sounder.

MVP Minimal Viable Product.

NN Nearest Neighbor.

QGIS Quantum Geographic Information System.

RMSE Root Mean Square Error.

ROI Region Of Interest.

SAGA System for Automated Geoscientific Analyses.

SBES Single Beam Echo Sounder.

TIN Triangulated Irregular Network.

UAV Unmanned Aerial Vehicules.

UI User Interface.

USV Unmanned Surface Vehicule.

1 Introduction

This thesis presents a novel QGIS plugin that brings an interpolation method developed specifically for creating gridded surfaces from bathymetric data.

The first section is about what bathymetry is, its methods and applications. Then it reviews some commonly used isotropic and anisotropic interpolation methods and existing tools used for bathymetric data interpolation.

The (s,n) flow-oriented coordinate system and the interpolation method used in BathyFlow-DEM are then presented, as well as the softwares and libraries used for its development.

Then the development workflow is explained as well as the the plugin's GUI, functionalities and user manual before a discussion reviewing the usefulness of the plugin and its potential future functionalities.

1.1 What is bathymetry

Bathymetry is the study of underwater topography of water bodies including oceans, lakes, reservoirs, rivers, streams and any other kind of water bodies. It provides detailed information about the underwater terrain that helps better understand the aquatic environments. Bathymetric studies include mapping of the seafloor or riverbed relief but also data about structures such as a bridge's foundations, a dam's water intake or a sunken boat. It can be considered to be the underwater equivalent of topography.

Bathymetric data is used by a wide range of people and fields. It is important to the maritime industry for safe navigation of ships. Marine scientists use it for oceanographic studies. Engineers rely on it to build and monitor underwater structures like pipelines and tunnels. Geologists can use it to understand the layers of sediments and types of rocks in a riverbed or in the seafloor. It is also used to understand how sediments move and evolve. Having access to this information is important for marine science, environmental monitoring and conservation, coastal management, dam monitoring and fishing industry (Legleiter & Kyriakidis, 2006; Merwade, 2009).

Numerical riverbed topography is especially important in hydrodynamic models, studies of sediment transport and documentation of channel changes (Legleiter & Kyriakidis, 2008).

The main goal of bathymetry is to produce detailed maps of the underwater floor. Different kind of equipment and techniques exist to survey the necessary data. The main ones are sonar, LIDAR and images.

Sonar

Sonar is a technique that uses waves (sound propagation) to measure a range. In bathymetry, it is used to measure depths and detect objects. The type of sonar used is called an echosounder, which is designed specifically to measure depths by transmitting sound pulses and analyzing their

echoes. They are two main types of echosounder: Single Beam Echo Sounder (SBES) and Multi Beam Echo Sounder (MBES). A single-beam echosounder emits a single pulse of sound vertically below the vessel while multi-beam echosounder emits multiple pulses in a fan shape and covers a wider area. Both types of echosounders can work in a range of frequencies. Low frequencies, typically below 100kHz, have longer wavelengths and can penetrate deeper in the water column but also in the floor. They are used to survey deep areas, and to investigate layers of sediments. High frequencies, above 100kHz, have shorter wavelengths and do not penetrate as deep as low frequencies. They do have a higher resolution, which make them more suitable for shallower waters where accurate mapping is important. The higher resolution is due to physical constraints of waves behaviours and sonar systems design; it is possible to create narrower beams with higher frequency than with a lower one. Both SBES and MBES can be mounted on ships or Unmanned Surface Vehicle (USV) (Li et al., 2023).

LIDAR

LIDAR (Light Detection and Ranging) uses laser pulses to measure distances. Airborne LIDAR is a common way to survey topography but also shallow waters and coastal areas (Li et al., 2023). It is a quick way to collect data but is limited to clear and shallow waters where the bottom can be seen. This is because LIDAR typically uses laser within the visible or near-infrared spectrum and these wavelengths are highly sensitive to absorption and scattering in the water. The strength and clarity of the signal diminishes with the depth.

Satellite-derived bathymetry

This is a technique used to map the underwater topography by analyzing data collected by satellites with active or passive sensors (Jawak, Somashekhar & Alvarinho, 2015). It applies reflectance analysis (clear water absorbs and scatters light differently depending of the depth) and/or machine learning algorithms to the data recorded by the satellites to estimate depths and create maps of the seafloor. This technique is efficient to cover large areas but is affected by water clarity, depth and other atmospheric factors.

Other techniques are available and constantly evolving. In the recent years Unmanned Aerial Vehicles (UAV) have become more technologically capable and widely available (Bergsma, Almar, de Almeida & Sall, 2019), facilitating their use in video-derived bathymetry and drone-tethered sonar system (Diaz et al., 2022).

This thesis concentrates on the sonar acquisition method with SBES, a technique widely used due to its simplicity, portability, cost-effectiveness and high accuracy (Li et al., 2023; Oliveira Ferreira & Coelho de Andrade, 2022). It is especially used in smaller water bodies such as lake, reservoirs and rivers where a high spatial resolution and extensive coverage isn't a critical requirement (Li et al., 2023).

1.2 Data acquisition with an echosounder

Using an echosounder to survey data is the traditional way (Jawak et al., 2015). It involves a field survey, as well as the necessary mission planning and post-mission data processing. While the specific steps can slightly differ depending on the mission's goal, field conditions and other requirements and constraints, the following general steps are applicable:

1. Mission planning and preparations

- Survey design: this steps involves planning the bathymetry survey lines (the path that the vessel will follow to record data) according to the purpose of the survey, the resolution required and the overall shape of the area. Access to the water body and overall safety is also considered at this point.
- Equipment and team preparations: ensuring that the echosounder and all related pieces of equipment such as computer, Global Navigation Satellite System (GNSS) antenna and power supply are functioning. Organize the team and roles.

2. Data Collection

- Set up: the set up includes mounting the SBES on the survey vessel, connecting it to the computer and base station if used.
- Calibrations: configure the echosounder's setting according to the water conditions.
- Data recording: the vessel navigates following the pre-planned navigation lines with the SBES turned on. It records directly below the vessel. During this step, constant monitoring of the recorded data is good practice in order to catch inconsistent or erroneous soundings.

3. Data processing

- Data back up and storage: after each survey and before starting any processing, the raw data is backed up and stored.
- Data cleaning: check the recorded data and remove data points that are wrong. This can be done automatically with the help of filtering or noise-reduction algorithms or manually, or both.
- Corrections: apply corrections to the raw data. This usually includes corrections for the echosounder's depth below water level, adjustments of the recorded position, adjust the sound speed in the water according to the water temperature and corrections for the water level.

4. Post-processing

- Interpolation and gridding: this step is optional but is a useful way to present the data. From the processed bathymetric data points, interpolation and gridding methods are used to create a continuous surface that can be used in further analysis. The present thesis focus on this specific step of bathymetric data post-processing.
- Visualization: this is finding the best way to make the data talk. Cross-sections can be presented in 2D drawings and plans, data points and surfaces can be included in maps with other data such as topography.

5. Analysis and reporting

- With the relevant data on hand, the last step is to analyse and extract the relevant information based on the survey's objectives. This is usually presented in a survey report that also includes the methodology used, the processing and post-processing steps, the quality control and assessment.

1.3 Bathymetric data and processing

Bathymetric data from a SBES consists of a set of referenced points with at a minimum X , Y , *depth* and *water level* information. More attributes such as date, time, number of satellites seen by the GNSS antenna, survey vessel speed or positioning mode can also be part of it. Another type of data that some echosounder produce is the echogram, which is a graphical representation of the echosounder's data with the distance on the horizontal axis and the depth on the vertical axis. It has the specificity to display the echo's intensity in a color scale.

Data points are not always the most convenient format to use for flow-modeling or cartography. In these cases, it is possible to post-process it to create meshes, Digital Elevation Model (DEM), contour lines or 3D models. This enhances the data usability and visualization but also introduces several implications. Some of these are discussed below.

When going from data points to a surface or contour lines, there is a certain degree of data generalization that happens. This generalization can lead to smoothing out sharp features and masking local variability in the data. This generalization is something that inevitably happens when creating a grid where the cell size is bigger than the distance between the input data points.

Also, interpolation and models rely on assumptions to fill the gaps in the data and estimate the values in between measured points. It is therefore important to have some experience with the kind of data and area before choosing a method to best reflect the reality in the post-processed data. Finally, when using measured points as the ground truth to derive further information, potential errors will be intensified, leading to some possibly erroneous results. Overall the quality of the surface interpolated depends on the data points' error, placement, density and distribution (Glenn et al., 2016). Li et al. (2023) explain that the choice of the interpolation algorithm is of great importance when interpolating SBES data.

However, post-processing remains a critical step as it enhances data usability and integration with other datasets as well as visualization, ultimately creating and format and is most useful for decision support.

As discussed in this section, choosing the correct post-processing method for bathymetric data is an important step towards its usefulness. This thesis presents an interpolation method adapted to this specific context, that can be used to create DEM.

1.4 Problem definition

The main goal of this work is to **present a new plugin within the QGIS software that will allow the user to perform anisotropic interpolation on bathymetric data in an efficient and user-friendly way.**

At the same time, the plugin needs to provide an interpolation method that is more efficient than what is available within QGIS.

1.5 Research question

The following research questions have been defined. Each of those are part of the bigger goal presented above.

- How can QGIS be extended with a plugin that offers anisotropic spatial interpolation ?
- What is the balance between ease of access and complexity so that the plugin is useful to expert and non-expert users alike ?
- What anisotropic interpolation methods can implemented ?

2 Literature review

This section goes over the different spatial interpolation methods that can be used to interpolate bathymetry data points. It then presents existing tools available to perform flow-oriented anisotropic interpolation.

2.1 Interpolation methods

The interpolation methods presented here can all be used to predict unknown values at unsampled locations. In the context of bathymetry, they are used to create a continuous surface by interpolating scattered depth measurements taken at discrete points. Traditional methods of spatial interpolation for bathymetric data include Inverse Distance Weighting (IDW), Kriging, Nearest Neighbour, Triangulated Irregular Network (TIN) from Delaunay triangulation (Glenn et al., 2016).

These interpolate following a straight line between the data points, ignoring the meanders and direction of the river (Legleiter & Kyriakidis, 2008; Merwade et al., 2005). Merwade et al. (2006), who explored various methods used for the interpolation of river channel bathymetry, found that methods that take into account the inherent anisotropy of the river are significantly better at producing accurate surfaces.

Below are presented different common algorithms used in the context of bathymetry, some with and some without anisotropy considerations.

2.1.1 TIN interpolation

A triangulated irregular network is a representation of a surface from irregularly spaced data points. While DEM is used to model the terrain as a grid of regularly spaced cells, where each cell contains elevation data, TIN uses non-overlapping triangles to create a mesh. A TIN contains nodes representing the measured data points often irregularly spaced, edges that connect the nodes, and triangles that approximate the surface between three nodes (Figure 1). The algorithm used to partition the space is the Delaunay triangulation. It forms triangles from a point dataset in a way that no point is inside the circumference of any triangle formed by the points. When multiple solutions are possible, it maximises the minimum angle of all the triangles.

TIN have the advantage that it is adjusted to the data. Its resolution vary depending on the density of the input points and there is no loss of information. Because of this, TINs can become very data-heavy when working with large areas or areas with a lot of data points. The file size makes it complicated to store, manage and use. Due to its irregularity (unlike DEMs that have cells of a specific size, it can be complicated to work with this type of data in a Geographic Information System (GIS) environment.

2.1.2 Nearest Neighbor and k-Nearest Neighbors

Nearest Neighbor (NN) is a straightforward interpolation method that is used to estimate the value of new data points based on their closest neighbor. It can be applied in one or more dimensions

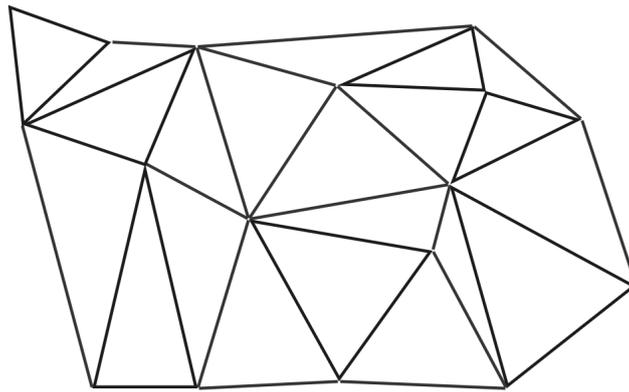


Figure 1: Example of a TIN surface made of triangles.

and commonly uses the Euclidean distance, especially when interpolating spatial data. It has the advantages of being non-negative, isotropic (all directions are treated equally) and directly representing the space. After identifying the new data point's position, the Nearest Neighbor algorithm computes the distance to all the known data points and assigns the value of the closest known point to the new data point.

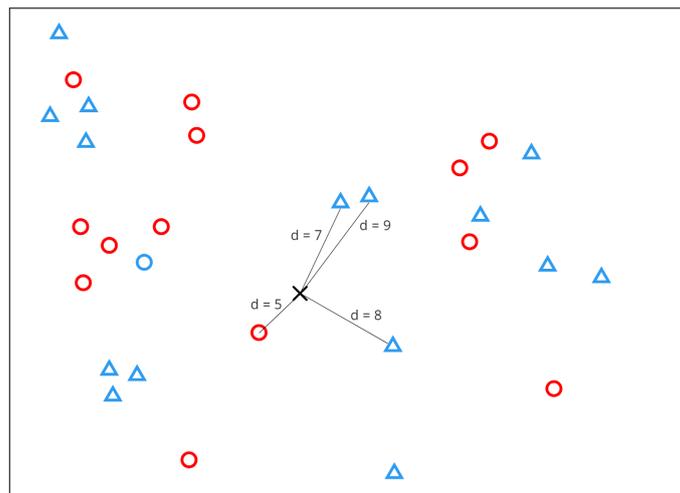


Figure 2: Two dimension illustration of the Nearest Neighbor and k-Nearest Neighbors interpolation algorithms. The new point X will take the value of the closest known point (circle) with NN and the average of the circle and three triangles with kNN where $k=4$

Another way to understand NN is through the Voronoi diagram, shown in Figure 3. The Voronoi diagram divides the space in regions based on distances. The boundaries between the regions are equidistant to the closest points. Each region around a data point is its region of influence. Using the NN interpolation method, any new data point falling into a known point's region of influence will have the same value. The Voronoi diagram can help understand and visualize the result of applying the nearest neighbor method in an area.

This algorithm can be generalized to the k-Nearest Neighbors (kNN), which considers the k-closest data points instead of a single closest one. Figure 2 shows the four closest neighbors to the target point. The value of the new point will be determined by the average of the values of the k-closest ones. Both methods are similar, and NN is a specific case of kNN method where $k=1$.

A few key differences are that NN is computationally less complex than kNN, whose complexity augments as k increases. NN is more sensitive to noise as it only takes a single point into account

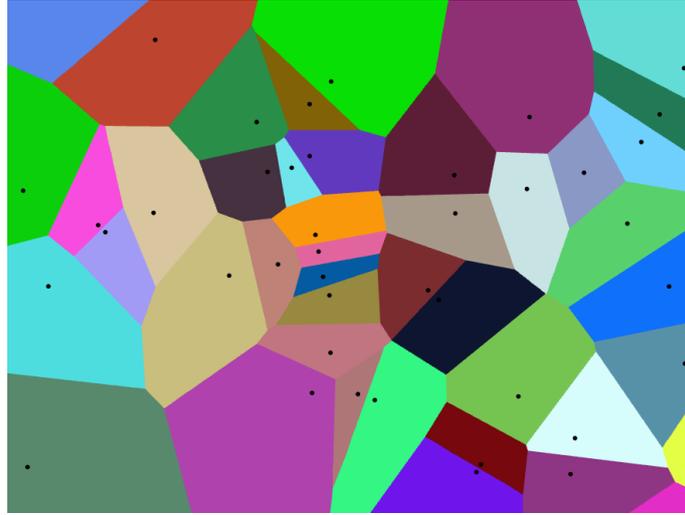


Figure 3: Two dimension Voronoi diagram illustrating the partitioning of the space based on the distance to the data points (black dots). Each color represents a point's region of influence.

compared to kNN.

2.1.3 IDW

Inverse Distance Weighting (IDW) is based on the distance between the unsampled data point and all the other sampled data points within a certain radius (Shepard, 1968). It assumes that the closer points will have a stronger influence and therefore assigns a weight that is inversely proportional to the power of distance. The new point's value is predicted as the weighted average of all the observed values within a defined radius. The main steps of the IDW algorithm are presented in Figure 4.

It is a method that is regarded as a standard interpolation method in GIS (Burrough, McDonnell & Lloyd, 2015).

To predict the value of an unknown data point (x, y) , first the distance d_i between that point and all the known points is calculated using Equation 1, where (x_i, y_i) is a known point.

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (1)$$

Then the value of the new point is determined by:

$$\hat{Z}(x) = \frac{\sum_{i=1}^N \frac{Z(x_i)}{d_i^p}}{\sum_{i=1}^N \frac{1}{d_i^p}} \quad (2)$$

where:

- $\hat{Z}(x)$ is the predicted value at location x .
- $Z(x_i)$ is the known value at location x_i .
- d is the distance between location x and location x_i .
- p is the power parameter that controls the influence of distance. A higher power means that the distant points will have less influence. Usually the value is 2 (Merwade et al., 2006).

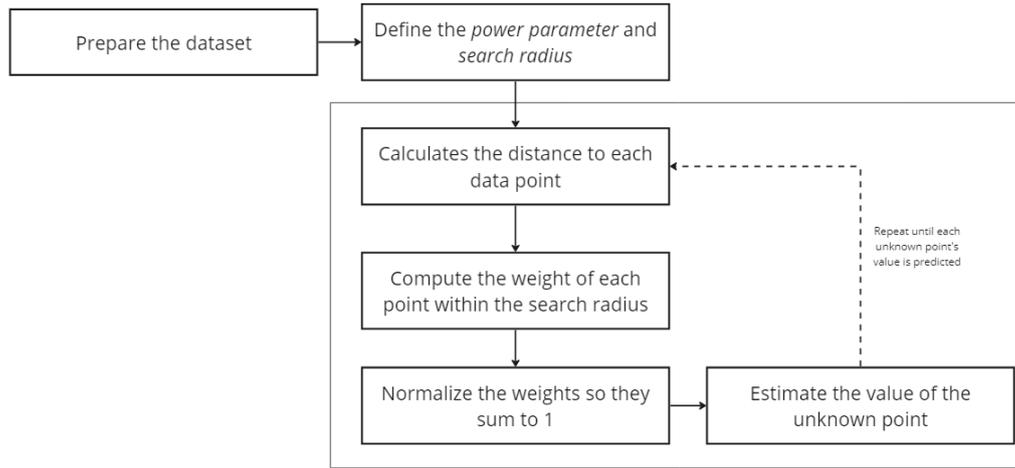


Figure 4: Step-by-step of the IDW algorithm.

2.1.4 Elliptical Inverse Distance Weighting

Elliptical Inverse Distance Weighting (EIDW) is a method developed by Merwade et al. (2006) to be applied to bathymetric data specifically. It incorporates the anisotropy of the riverbed by modifying the way that the search neighborhood is defined; instead of using a circular search as in IDW, it uses an ellipse that is elongated in the direction of the flow, allowing the use of more points in the flow direction. This is illustrated in Figure 5, that shows the circular neighborhood of standard IDW and elliptical neighborhood where the major axis is the direction of the anisotropy (B). Both neighborhoods are centered on the interpolation point.

This interpolation method takes advantage of the anisotropic nature of the data, which in bathymetric data is along the flow direction. Because of this, the data must be converted to a flow-oriented coordinate system before applying EIDW. One of the advantages of such a system is the constant flow direction. It is explained in details in Section 3.

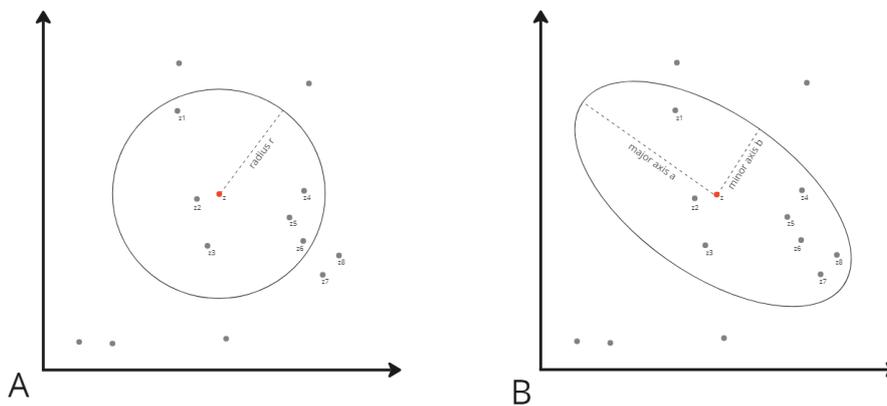


Figure 5: IWD with a circular neighborhood (A) and an elliptical neighborhood (B). Z is the unsampled point.

This method also modifies the importance given to the data points within the search ellipse by assigning more weight to the points that are located along the s-axis (along the flow direction). The weight is given by Equation 3.

$$d_i = \sqrt{\left(\frac{1}{a_r}(x - x_i)\right)^2 + (y - y_i)^2} \tag{3}$$

The anisotropy ratio a_r is the ellipse's major axis divided by the ellipse's minor axis (Equation 4).

$$a_r = \frac{a}{b} \quad (4)$$

2.1.5 Kriging model

The Kriging model, formalized by Matheron (1963), influenced by the work of Krige (1951), estimates the value at a new point based on a linear combination of the known data points. It is a stochastic process, meaning that a value at a determined location is considered a realisation of a random function that is characterized by a specific probability distribution.

Unlike other spatial interpolation algorithms, it also adds the error by explicitly estimating the uncertainty of the predictions.

To define the weight of each known point, it assumes that the points that are closer to the new point will get a bigger weight, similar to IDW. It also considers autocorrelation of the data points: points that are close together are correlated and will not weight as much as a point that is further away. Figure 6 shows first A, where all the points have the same weight and B, where points 1 and 2 would each weight less because it is assumed that they share some redundant information.

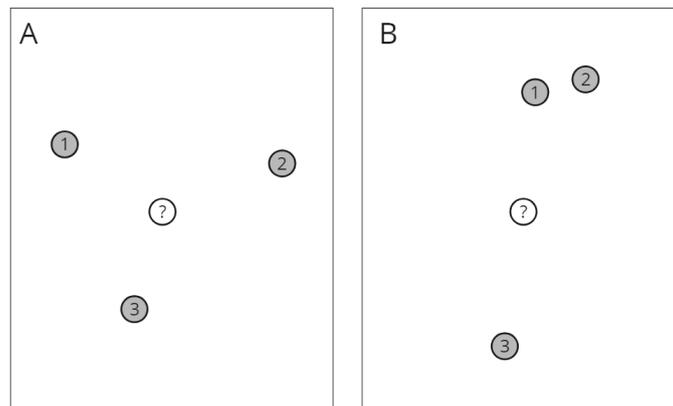


Figure 6: Modified distance of data points within the search radius

So the value of a new point is determined by equation 5:

$$\hat{Z}(x) = \sum_{j=1}^i w_j Z(x_j) + \epsilon \quad (5)$$

where:

- $\hat{Z}(x)$ is the predicted value at location x .
- x_j is the known value at location Z .
- w_i is the weight assigned to location Z .
- *epsilon* is the error term.

This relationship of distance and variance is defined in the semivariogram. The semivariogram is a graphical representation of the autocorrelation of a dataset. The x -axis is the distance between two given points and the y -axis is the semivariance, a measure of dependency between two points. In such a graph space, each point represents a pair of observed points. If the semivariance is high, it suggests that the two points have very different values. If it is low, that they are similar. The assumption is that points that are close together will have a low semivariance and the ones that are further away from each other a high semivariance. Figure 7 presents a semivariogram with

Table 1: Semivariogram models

Model's name	Pattern
Spherical model	Rapid increase of the semivariance up to a certain distance, then leveling off
Exponential model	Increases until it nearly levels off - which it never does as it is an exponential function
Gaussian model	Semivariance increases slowly near the origin, then sharply increases and slowly levels with the sill
Linear model	Assumes a linear increase of the semivariogram over the distance

theoretical data. It also shows three important concepts of the semivariogram. The *range*, representing the distance at which the semivariogram stops increasing, meaning that past this distance, points stop being autocorrelated. The *sill* is the corresponding semivariance at the range and the *nugget* is the y value where the graph begins. Theoretically, the semivariogram should start at 0, where two points that are at the same location have the same value. But this is not always the case because of measurement error for example. The higher the nugget, the noisier the data.

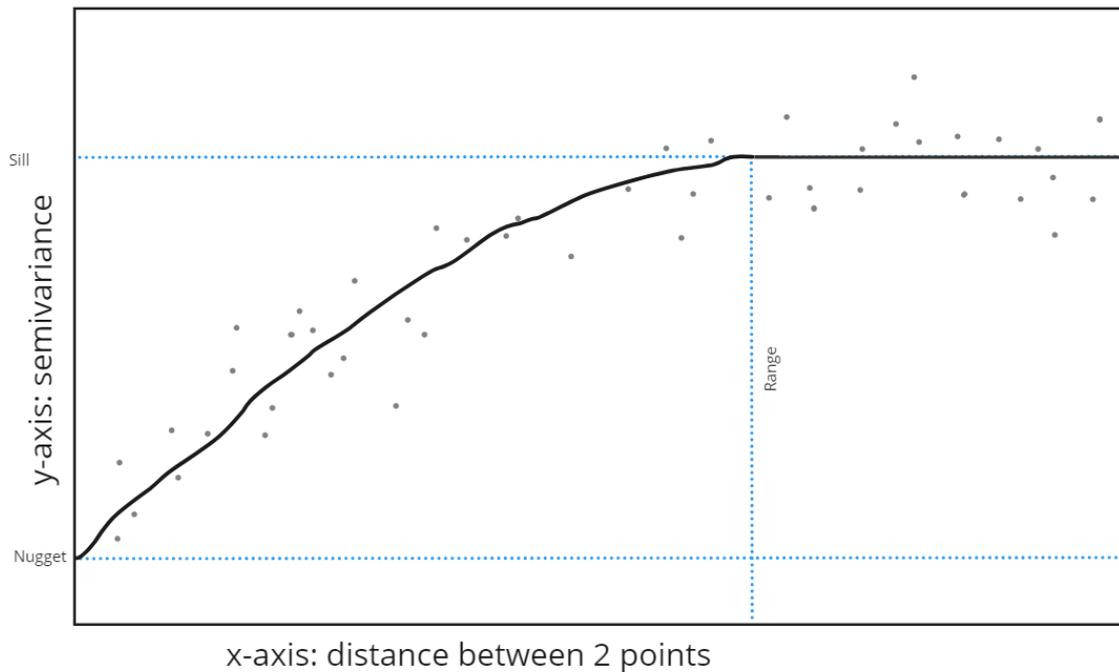


Figure 7: Example of a semivariogram

The first step to estimate new points' values using the Kriging methods is to choose a semivariogram model based on the dataset. The selected model will influence the estimations of the unknown values. Each model is distinct and leads for a different pattern. Common semivariogram models are listed in Table 1.

The parameters such as the nugget, sill and range are modified so the model fits the empirical semivariogram best. It is based on this semivariogram that the Kriging method will calculate the weight of each known point and use a linear combination to predict the new point's value. Also, the method calculates the variance for each point, which represent the uncertainty of the prediction.

While Kriging is a widely used geostatistical method for data interpolation, Krüger et al. (2018) explain that the Kriging method can be difficult to automate. It requires choosing a suitable

model and parameters according to the empirical variogram which can be an iterative process and presuppose some amount of user knowledge. It is also a sophisticated method compared to other interpolation methods such as NN or IDW (Merwade, 2009).

2.1.6 Local Anisotropy Kriging

While the Kriging method can use different models whose parameters can be adapted to the dataset, it doesn't take into account anisotropy. Stroet and Snepvangers (2005) present an interpolation method with Kriging that incorporate the changing local direction of anisotropy, Local Anisotropy Kriging (LAK). It combines image analysis to determine the direction of anisotropy for each cell and a Krigin method using the direction. This method calculates the anisotropy direction from input data points.

LAK works iteratively. After a first Kriging estimation with the sparse data, it applies the algorithm to calculate local anisotropy and use the Kriging method again with anisotropy. This cycle can continue multiple times until the results are appropriate.

One advantage of this method is that it doesn't need any other input than the dataset to work and no assumption about the structure have to be made in advance. But while it is a way to incorporate the varying direction of anisotropy and it could be used to interpolation bathymetry data, it is also fairly complicated to implement because of the iteration and requires a lot of user input and knowledge.

2.2 RMSE

The Root Mean Square Error (RMSE) is a metric used for assessing the accuracy of predictive models such as interpolation models. It measures the average of the errors between the predicted and observed values in the same metric as the origin data. Because it calculates the squared root of the average squared error, it penalizes larger errors more than smaller ones. By doing so, it makes sure that significant deviations are weighted more and smaller ones weight less proportionally. This is a good way to do it because in the real word, smaller errors and deviations are generally accepted while important deviations can have significant consequences.

The RMSE is given by:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (Z_i - \hat{Z}_i)^2} \quad (6)$$

where:

- Z_i is the observed value at location i .
- \hat{Z}_i is the interpolated value at location i using IDW.
- n is the number of validation points.

The result is a single numerical values that summarizes the accuracy of the models. This makes it easy to understand, compare an interpret: the smaller the RMSE the better the model.

2.3 Existing tools

Some tools and softwares were developed specifically to interpolate bathyemtric data with anisotropic considerations. This section presents some of them.

2.3.1 ArcGIS Bathymetry Interpolation plugin

Dr. W. Merwade created a custom tool for ArcGIS based of the work of Merwade et al., (2008). The most recent tutorial of the program is from 2018 (Merwade, 2018) and states that the tool is available to download as an add-in for ArcGIS 10.5. The program provides a semi-automatic way to interpolate the data with anisotropy in a flow-oriented coordinate system. The inputs are surveyed cross section, a centerline and the main channel boundaries.

First, the surveyed cross-sections are mapped in a flow-oriented coordinate system where s = streamwise direction is the distance along the centerline and n = normal direction is the direction perpendicular to the flow. The transformation is described in details in Merwade et al., (2005).

Secondly, new cross-sections are interpolated in the s,n coordinate system using linear interpolation. Then the cross-sections are joined to create a mesh that keeps the data of the original cross-sections. The resulting mesh is transformed back to s,n Cartesian coordinate system.

This method is based on the transformation from a Cartesian coordinate system to a flow-oriented coordinate system and linear interpolation. With the user's input data of centerline and channel boundaries, it can create a robust mesh even if the surveyed cross-sections are measured more that 10-15 times the channel width apart from one another. This method can work where other fail, such as EIDW (Merwade et al., 2008).

There are multiple limitations to this tool. The first one is that it was created for ArcGIS, a GIS software that is accessible only with a licence, limiting user access. Furthermore, the latest tutorial available online is from 2018 and applies to ArcGIS 10.5, a version that is no longer updated. This makes it difficult for potential users to find and use it.

Secondly, it was developed with SBES bathymetry in mind - it is interpolating new cross-sections, then creating a mesh. While the anisotropy component is there, that logic cannot be extended to a MBES survey as interpolating cross-sections wouldn't make sense in that context.

2.3.2 Tautoine Mesher

Tautoine Mesher is a tool developed by Duron, Cierco and Saad (2019) to interpolate bathymetry from discrete cross-sections. The software is built on the Python programming language and the source code is openly available. It has a high degree of flexibility; it integrate various 1D and 2D interpolation methods.

It uses cross-sections, constraint lines and the hydraulic axis (or centerline) as inputs. After converting the points to a flow-oriented coordinate system relative to the hydraulic axis, intermediate cross-sections are created and the whole area is meshed using TIN. Then the depth of the new nodes is interpolated, using either linear interpolation or a Spline interpolation.

The tool is based on several open-source scientific python libraries such as Numpy and SciPy and is available as command line interface scripts. This lets the end user have the highest degree of customization as the whole project's code is publicly available. At the same time, it restricts the type of users that use it because of the lack of GUI and complexity of set up.

From the interpolation of supplementary cross-section, to mesh generation and interpolation in a flow-oriented coordinate system, Tautoine Mesher is a multiple steps system. It is also a command-line system dedicated to transform data for further use in TELEMAC-MARCARET, a flow modeling software. The target demographic is very specific, excluding some users that might not have extended knowledge about geoprocessing or flow dynamics. Also, it requires a lot of input data to work, which again presuppose some knowledge from the end-user about a specific area.

2.3.3 ANUDEM

Australian National University Digital Elevation Model (ANUDEM) (Hutchinson, 2011) is a program developed at the Australian National University of Canberra by M. F. Hutchinson. It is specifically designed to produce hydrologically sound DEMs that accurately represent terrain features. It can take different kind of input data such as data points, contour lines, streamlines, sink points, lake boundaries, cliff lines and data mask polygons.

ANUDEM generates not only an output DEM but also error file for each type of input data, the derived stream line, cliff line, flow direction grid and output aspect grid.

The procedure is a morphological approach that can be adapted to the special nature of the terrain (Hutchinson, 1988, 1989). What differentiates it from other interpolation methods is the constrains of global drainage, which automatically removes invalid sinks. This global drainage condition can significantly improve the accuracy of DEMs interpolated from sparse datasets (Hutchinson, 2011).

The method used in the ANUDEM program has be implemented in ArcGIS Pro as the *Topo to Raster* tool. The source code however isn't available as ArcGIS Pro is a commercial GIS software.

Overall, this method aims to create accurate DEMs with an approach that implements an automatic drainage enforcement algorithm. One caveat is that it requires a lot of different input data and the resulting outputs needs to be carefully interpreted. Accessibility is the second problem of this method, as both the ANUDEM software and the tool in ArcGIS aren't available to the general public.

3 Coordinate systems

This section details the two different coordinate systems used in this work, namely the Cartesian coordinate system and flow-oriented coordinate system. It then explains the transformation method used in the BathyFlowDEM plugin to transform data points from the x,y Cartesian system to the s,n flow-oriented system.

3.1 Cartesian and flow-oriented coordinate systems

In a Cartesian coordinate system each data point P is defined by a pair of values, x and y . These values are the distance from a fixed origin. In geodesy, the x -axis is going from west to east and the y -axis from south to north and both are perpendicular to each other.

The flow-oriented coordinate system, also called curvilinear orthogonal coordinate system or channel oriented coordinate system is used here to describe the river environment. The s -axis is oriented along the flow direction (stream wise) and the n -axis is perpendicular to it (normal). The n values can be positive or negative while the s values cannot, the axis starting at 0. Figure 8 shows data in both systems.

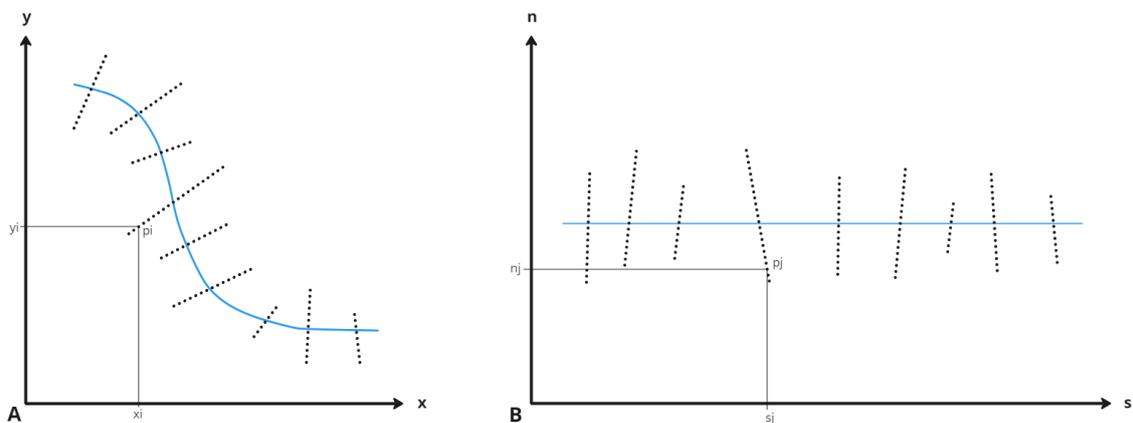


Figure 8: Theoretical bathymetric data represented in A) a Cartesian coordinate system where $p_i = (x_i, y_i)$ and B) in a flow-oriented coordinate system where $p_j = (s_j, n_j)$. The centerline and flow direction is represented by the blue line.

In both systems, the z -coordinate remain the same as the transformation is 2D.

The general direction of anisotropy in a river or channel is along the flow direction. This means that the spatial variability of the underwater topography is bigger perpendicular to the center line of the river than along the centerline. In a Cartesian coordinate system, the direction of anisotropy changes regularly as a river or channel is hardly ever completely straight. The fact that this direction is constantly changing makes it difficult to efficiently implement interpolation methods that

take anisotropy into account.

Figure 9 below shows how the s, n coordinates are calculated in relation to the x, y Cartesian coordinate system.

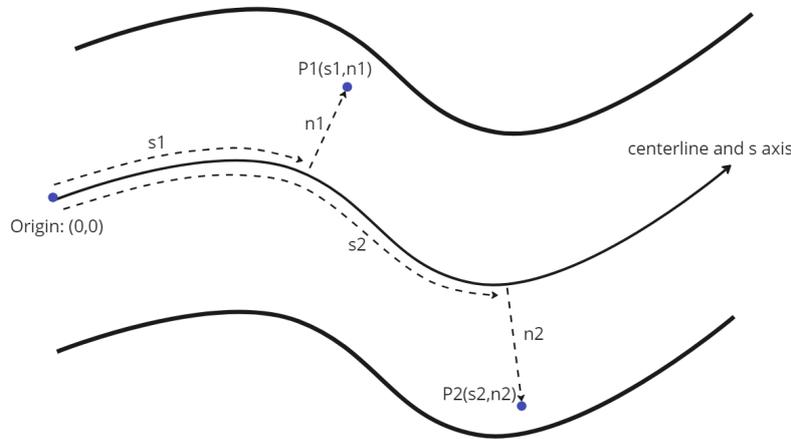


Figure 9: Two data points in a Cartesian coordinate system with corresponding flow-oriented coordinates. Includes the centerline and both banks.

Interpolation of bathymetric data with anisotropic methods gives better results when the anisotropy of the river or channel is regularized to a single direction (Merwade et al., 2006). In other words, anisotropic interpolation algorithms perform significantly better than anisotropic methods but only when the data is transformed in a flow-oriented coordinate system. This is the reason why the s, n channel-fitted coordinate system is used in BathyFlowDEM.

3.2 Coordinate transformation in BathyFlowDEM

Merwade et al. (2005), Goff and Nordfjord (2004) and Legleiter and Kyriakidis (2006) all present slightly different methods to transform data points from a Cartesian to a flow-oriented system. The transformation presented here differs as it was adapted to work best and most efficiently within the PyQGIS environment.

This method needs two different kinds of inputs. First the data points, in a Cartesian coordinate system. It is important that the dataset is in this system as the transformation doesn't take into account a geoid or ellipsoid. If the input data is in a geographic system, it needs to be projected first. The second input data is the river or channel centerline. The user is free to define it based on site knowledge and/or remotely sensed images. The centerline isn't necessarily the thalweg location. One constraint is that the input *.shp* layer containing the centerline must have a single line, and its direction has to be from upstream to downstream.

The steps are the following, for a data point $p(x, y)$ and centerline CL . The three methods are part of the QgsGeometry class (QGIS.org, 2024).

1. **lineLocatePoint:** It is a method that calculates how far along a line one has to go to get to the closest location that the line gets to a specific point. This is the s -coordinate.
2. **shortestLine:** This method returns the shortest line joining two geometries. It is effectively calculating the absolute $|n|$ -coordinate.
3. **closestSegmentWithContext:** closestSegmentWithContext takes a single point as input and returns a tuple with various information. The one that is used for the coordinate transformation is *LeftOf*: it returns

< 0 if the point is left of the centerline, > 0 if it is on the right side and 0 if the test is unsuccessful of the point is located exactly on the line.

Those methods are run on each of the input data point to get the s and n coordinates with reference to an input centerline. The program then creates a new point *.shp* file with the input points and attributes S and N and field to interpolate. This layer is loaded in the QGIS project.

4 Interpolation method

This section goes over the interpolation method used in BathyFlowDEM, IDW with anisotropy. It introduces the theory and explains how it is applied in the plugin.

4.1 IDW with anisotropy

The interpolation methods developed for this plugin is similar to Merwade et al. (2006) - described in more details in Section 2.1.4. While EIDW changes the s assigned to data points according to an anisotropy ratio derived from the ellipse's major and minor axis, BathyFlowDEM directly modifies the weight of the data points based on the user's anisotropy ratio.

The distance of each point is calculated by:

$$d_i = \sqrt{(s - s_i)^2 + a(n - n_i)^2} \quad (7)$$

where:

- d_i is the distance from the target to the known point.
- s_i and n_i are the coordinates of the known point.
- s and n are the coordinates of the target point.
- a is the anisotropy ratio.

BathyFlowDEM's IDW with anisotropy needs two parameters from the user: the search radius and the anisotropy ratio. When computing the distances between the target point and the known data point, the distance on the n -axis (transverse to flow) is multiplied by the anisotropy ratio (see Equation 7). This has the effect to exclude some points that might have been within the search radius if the distance calculations were done without the ratio. The search radius is used to select the points that will be taken into account to predict the new point's value. Figure 10 shows how the data points are moved after applying the anisotropy ratio. Some of them end outside of the search radius, here, $Z1$, $Z2$, $Z3$ and $Z4$.

4.2 IDW with anisotropy in BathyFlowDEM

A native IDW interpolation is available in QGIS. However, since the way to compute the points' distances is changed, BathyFlowDEM implements a custom function for it.

The function loops over all data points, computing the distance between the target point and the known point. Because the n coordinate can be positive or negative, it had to be adapted. The resulting value is then passed to the target point attribute.

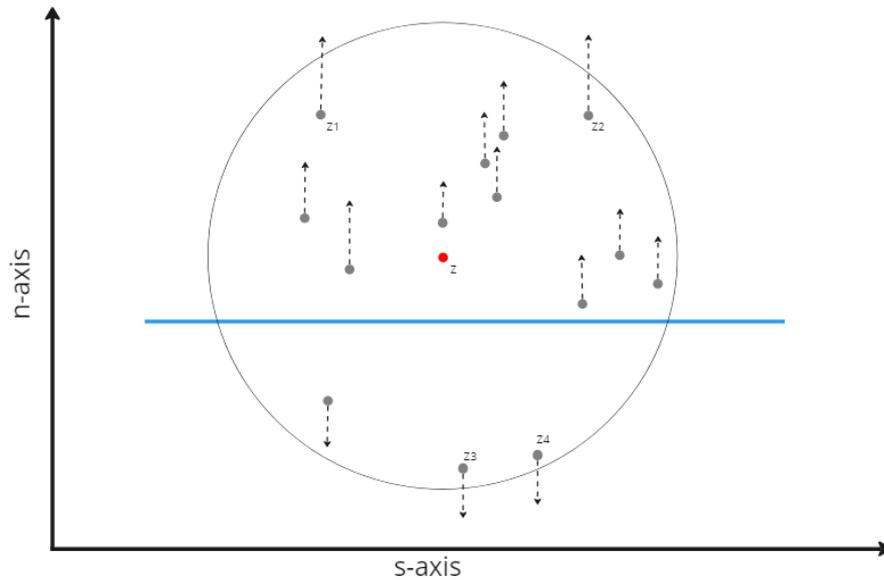


Figure 10: The data points selected by the search radius before and after applying the anisotropy ratio to make points in the n-axis more distant.

Table 2: The `eidw()` function's inputs and output

Function	<code>eidw()</code>
Inputs	<p>QgsVectorLayer with the data points with at least S, N and field to interpolate</p> <p>Value field (str): Name of the field with the values to interpolate</p> <p>Target_s (float): S coordinate of the unknown point</p> <p>Target_n (float): N coordinate of the unknown point</p> <p>Anisotropy ratio (float): Factor by which distances across the flow (N) are scaled</p> <p>Search radius (int): Radius to select the points to use for interpolation</p>
Output	A float, the interpolated value at target location.

```

1 # Calculate distances in the S direction
2 ds = s - target_s
3
4 # Same for n while taking negative numbers into account
5 if target_n and n >= 0:
6     dn1 = abs(target_n - n)
7 elif target_n and n < 0:
8     dn1 = abs(target_n - n)
9 else:
10    dn1 = abs(target_n) + abs(n)
11
12 # Applying the anisotropy ratio
13 dn = dn1 * anisotropy_ratio
14
15 # Calculate the anisotropic distance by modifying it on the N axis
16 distance = (ds**2 + dn**2) ** 0.5

```

Listing 1: Computing the s and n distances in the `eidw()` function

5 Tools

This section presents the softwares and tools used to create BathyFlowDEM.

5.1 QGIS

QGIS is an open-source geographic information software and is available on all platforms. It allows users to create, edit, visualize, analyse and publish geospatial information. It still offers a wide range of functionalities comparable to commercial GIS solutions. It is utilized across many industry fields due to its versatility and robustness such as urban planning, environmental monitoring, resources management, transportation, engineering and more.

A wide range of data type is handled by QGIS. It can read and use vector data including points, lines and polygons, different types of raster formats for DEM or satellite imagery, database formats with integration of GeoPackage and PostGIS. It can also handle Web Map Service and Web Map Tile Services. Working with table formats such as Comma Separated Values (CSV) or Excel spreadsheets is also possible. All these different kinds of data can be imported and visualized into a single QGIS project (Figure 11).

While it possesses some native functions, QGIS also integrates geospatial libraries. Geospatial Data Abstraction Library (GDAL) and the OGR Simple Feature Library are shipped with it, enabling more functionalities. Other libraries such as System for Automated Geoscientific Analyses (SAGA) and Geographic Resources Analysis Support System (GRASS) are also available but additional steps to be installed into the main QGIS environment.

Apart from the native functions and algorithms available within the software, QGIS also allows users to create their own workflows and algorithms:

- **Graphic Modeler:** the graphic modeler is a visual programming tool that allows users to create complex custom workflows. It chains together inputs, algorithms and outputs to ensure that the processing is reproducible. For users that want to go further, it also has the option to export the model to PyQGIS.
- **PyQGIS scripts:** PyQGIS is the Python Application Programming Interface (API) for QGIS. It allows users and developers to use existing functionalities and extend them by writing custom scripts. These scripts can be used directly into the main QGIS interface but also outside of the desktop application. This is especially useful to automate or schedule tasks.
- **Plugins:** another way that QGIS lets its users extend its functionalities is through the use of plugins. Plugins can easily be installed through the main plugin repository, accessible directly from the desktop application. They can also be installed locally.

The QGIS software and specifically the plugin route was chosen for this work for various reasons. First, because of its open-source nature encourages a high-degree of quality control through community collaboration (Flenniken, Stuglik & Iannone, 2020). The fact that it is not protected behind paywalls makes it accessible to everybody, users and developers alike. Second, because

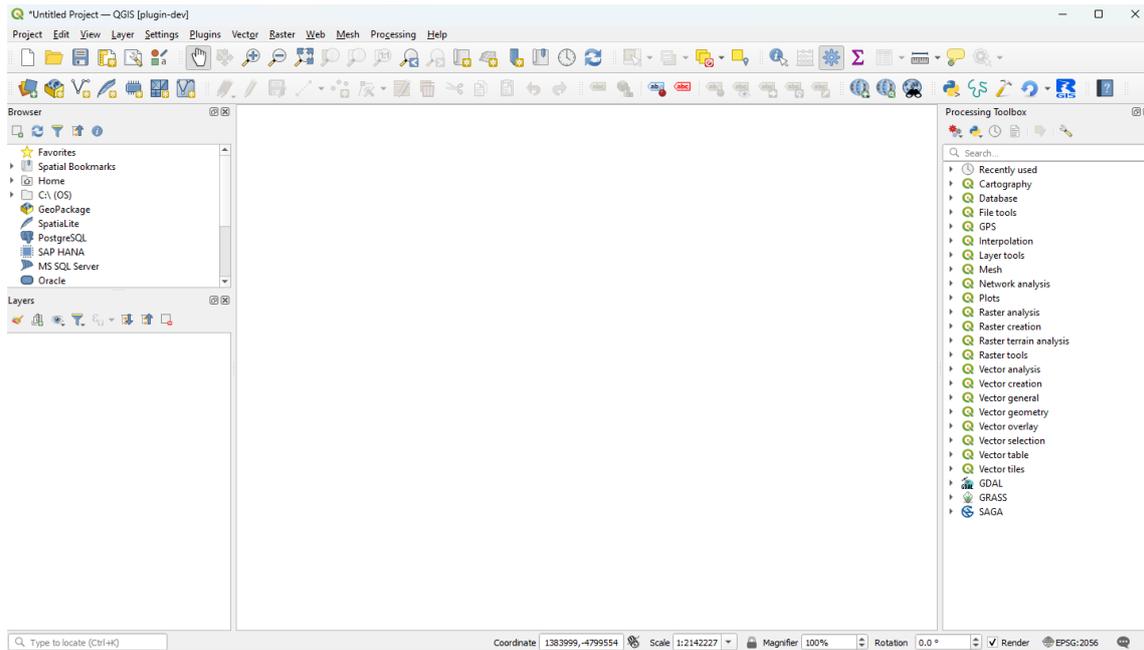


Figure 11: A possible QGIS layout with the map canva in the middle, the processing toolbox on the right, and file browser on the top left and the layers loaded the project on the bottom left.

it is widely used in different fields, including bathymetry data post-processing and topography. This allows the end-users to stay within the same environment for the data analysis and visualization. The new functionalities are easier to integrate in an existing workflow when they are in a known software/platform. Finally, PyQGIS' extensive documentation and community of users and developers were also a key factor in the decision to go with QGIS.

5.2 Python and PyQGIS

BathyFlowDEM is mostly written in the Python programming language. It uses PyQGIS's Python API to import the necessary libraries. PyQGIS allows developers to access QGIS functionalities programmatically and provides tools to manipulate components such as layers and features, to automate workflows and to create custom functions.

PyQGIS is divided into several libraries that can be imported independently and each serving a different purpose. Below are listed the ones used for this project.

- **Core** *qgis.core*: contains basic functionalities including managing vector and raster layers, spatial reference systems and feature manipulation.
- **Gui** *qgis.gui*: the Gui library is built on top of the Core library and contains classes and widget for creating graphical interfaces.
- **Processing** *qgis.processing*: the Processing library is built on top of the Core library and adds access to processing algorithms. It integrates with third-party tools such as GDAL, GRASS and SAGA.

5.3 Qt Designer

Qt Designer is the GUI design tool that is used to create and customize the plugin's interface. It offers a drag-and-drop interface to visually create the plugin's interface with GUI elements that display information or settings to be modified. Those are widgets such as buttons, text and spin-boxes that can be used in the canvas to design the interface. It is part of the Qt framework, the



Figure 12: QgsMapLayerComboBox and QgsFieldComboBox widgets in the QGroupBox, with their respective label. Part of BathyFlowDEM's GUI

one QGIS is built on and comes with the standard QGIS installation.

Some widgets come directly from Qt Designer while the version that ships with QGIS also has some QGIS-specific widgets. Those are related to layers and spatial data inputs.

Each widget has a set of widget-specific attributes that can be modified such as name, minimum or maximum values, some restrictions. The ones that are used to create the plugin's GUI are detailed in Section 5.3.1 below.

Once the interface is ready in Qt Designer, it is exported as *.ui* file and PyQt5 makes the links between Qt and the plugin. PyQt5 is a set of Python bindings for Qt libraries and is used to create cross-platform applications with GUI interfaces. It connects the *.ui* file (designed in Qt Designer and based on C++) to Python scripts. It creates a bridge between the GUI's elements and the plugin logic.

When the Qt GUI is initiated by opening the plugin in QGIS, it is constantly listening for "signals". If a signal is assigned to a "slot", a determine action will be executed. This behaviour is detailed in Section 5.3.2 - Signals and slots.

5.3.1 Widgets in Qt Designer

This sections goes over the different widgets used for BathyFlowDEM and is intended to better understand each widget's capabilities.

The "Tab" widget is the highest level widget used in BathyFlowDEM. Its role is to organize the plugin's GUI between the *Processing* part and the *Help* part. It has the advantage that when one is active, the other one is hidden, making for a cleaner and clearer interface. Within the "Tab" widget the content is divided into different boxes. The boxes only serve for the layout and organization and each have their own title.

"QgsMapLayerComboBox" (Figure 12) and "QgsFieldComboBox" widgets are used for the user input part of the dialog. The first one is to select a layer from the current open QGIS project. This widget is set up so that the kind of layer it can accept is restricted: only point vector layers for the input data points field and only a line vector layer for the centerline. This reduces the need to check for the layers' type in the algorithm before starting the processing and reduces user error and potential frustration. The "QgsFieldComboBox" is a type of widget that is directly related to "QgsMapLayerComboBox". It lets the user select which attribute of the selected layer to use for interpolation. Both widgets are linked together using signals and slots described below.

"QgsDoubleSpinBox" (Figure 13) and "QgsSpinBox" both take all the features and attributes of the respective "QDoubleSpinBox" and "QSpinBox" (the formers for QGIS and the latter native Qt widgets). They all allow the users to input numbers. The difference is in the design, the QGIS widgets are made so that they are coherent with the rest of the GUI. For consistency, QGIS-specific



Figure 13: Example of a QgsDoubleSpinBox and QCheckBox widgets in the GUI of BathyFlowDEM.

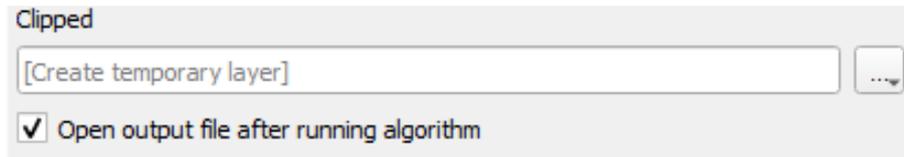


Figure 14: The saving layout of "Clipped", a native QGIS algorithms.

widgets are always used when available, even when not using the extra options. Like other widgets they can have restrictions such as a maximum or minimum value, or a defined step. In BathyFlowDEM, a "QgsDoubleSpinBox" is used for the anisotropy ratio value and the output raster's cell size and a "QgsSpinBox" is used for the search radius.

The "QCheckBox" widget (shown in Figure 13) is a boolean type input. It can be checked or unchecked and is used to let the user decide if they want data validation or not.

The last widget to be presented here is the "QgsFileWidget", (Figures 14 and 15) used to get the directory path of where the user wants to save the results of the processing. This widget has four different modes: "GetFile", "GetDirectory", "GetMultipleFiles" or "SaveFile". It is the "GetDirectory" that is used in BathyFlowDEM.

When the design is ready, it is saved as a `.ui` file containing the XML description of the plugin's layout.

5.3.2 Signals and slots

Signals and slots are the way that the different elements of the interface communicate. When the internal state of an object has changed, a signal is emitted and if it is linked to a slot, the action is executed immediately. In BathyFlowDEM, four signals that can be triggered by the user are linked to slots. First, the change of the data points layer. When it is changed by the user, the "QgsMapLayerComboBox" emits a signal that is caught by the corresponding slot and the action is to populate the "QgsFieldComboBox" with the attribute from the new layer. The second one is linked to the temporary layer "QCheckBox": when it is checked, the "QgsFileWidget" is enabled and when it isn't checked, it is not. The slots are defined in the main BathyFlowDEM class and are connected to a signal in the `self.initGUI()` function.

The two other slots are linked to the "OK" and "Close" push buttons. Unlike other "QPushButton" widgets, those have predefined behaviours: the "OK" button closes the plugin window, starts the main algorithm and resets the GUI and the "Close" button closes the window and resets the GUI. This worked at first, but needed to be changed when adding the data validation



Figure 15: The saving layout of the BathyFlowDEM plugin

function. The RMSE needs to be displayed in the plugin's window and the default "OK" button behaviour wasn't the best option anymore. So the default buttons have been replaced by standard "QPushButton" widgets with "OK" and "Close" text on them. The two have slots connected to the **clicked** event - when the user click on the button. Listing 2 shows how the slots are connected to the signals.

```

1 # Connect dialog signals and slots for user interaction with GUI
2 self.dlg.cbInputPointLayer.layerChanged.connect(self.
   oncbInputPointLayerWidget_layerChanged)
3 self.dlg.cbTempLayer.stateChanged.connect(self.oncbTempLayer_stateChanged)
4 self.dlg.buttonOK.clicked.connect(self.onStart)
5 self.dlg.buttonCancel.clicked.connect(self.onCancel)

```

Listing 2: Connecting the signals and slots in the self.initGUI() method in BathyFlowDEM class

5.4 Other useful tools for development

This section goes over some more tools that were used for the development of BathyFlowDEM. While not as critical as QGIS or Qt Designer, they made the process efficient.

5.4.1 Plugin Builder

The Plugin Builder is a plugin available in the official QGIS plugin repository. It is a useful tool for developers made to simplify the initial set up of new plugins. It creates the boilerplate code and directory structure through a user-friendly interface. This allows to save time in the first step of development. The version used for BathyFlowDEM is v3.2.1 (Figure 16).

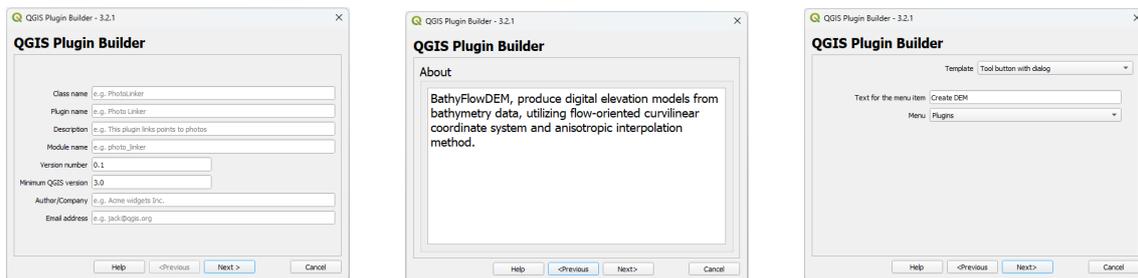


Figure 16: The process of creating a new plugin boilerplate with the Plugin Builder

5.4.2 Plugin Reloader

The Plugin Reloader is another plugin designed to help the development of other QGIS plugins. When installed, it adds a button to the toolbar that allow the plugin that is being developed to be restarted instead of having to close and open QGIS again. This is very useful as normally for any change in the source code of the plugin, QGIS needs to be closed and opened again for them to be applied. The Plugin Reloader makes testing changes and new functionalities quicker.

6 Methodology

This sections presents the development of BathyFlowDEM. The implementation followed an iterative process that was done twice. This is to ensure the plugin's robustness and viability as well as to make sure that the most important functionalities were developed first.

First, a basic version of the plugin was conceptualized and developed to implement the core functionalities. The process involved defining the necessary requirements for a Minimal Viable Product (MVP). The goal is to have a MVP that can already be used, tested and reviewed. After this initial first version, the requirements for the second one were defined and selected based on their impact, feasibility and feedback from the MVP tests. This involved refining some of the existing functionalities and adding new features.

The two version, described in details below, are:

- **Version 0.1:** this version implements data import, the main interpolation algorithm, creates a new raster layer with the results and saves it to a temporary layer. It is already integrated into QGIS and can use layers from an open project.
- **Version 0.2:** integrated more functionalities: data validation method, added user inputs and more complete GUI.

6.1 Definition of requirements and implementation of version 0.1

The first stage was to decide on the requirements of the minimal working version (listed in Table 3). The focus here is to have a plugin that is well-integrated into QGIS, that can perform a coordinate transformation from (x,y) to (s,n) and do the interpolation with anisotropy. This includes creating a raster output and loading is as a temporary layer to the project. User's inputs and choices are kept minimal. Figure 17 shows the GUI of that MVP.

Because there is no attribute selection at this point, the input data points' layer must have one attribute named exactly 'Z' that BathyFlowDEM will use. The resulting layer is loaded to the current project as a temporary layer.

The QGIS integration was handled with the Plugin Builder (Section 5.4.1). This tool creates all the necessary boilerplate code and more to start plugin development. It was decided to make BathyFlowDEM accessible from the toolbar as well as from the *Plugin* menu.

The core functionalities are the coordinate transformation and interpolation algorithm, described in details in Section 3 and 4. The coordinate transformation is using native QGIS algorithm while the interpolation function is done using Python.

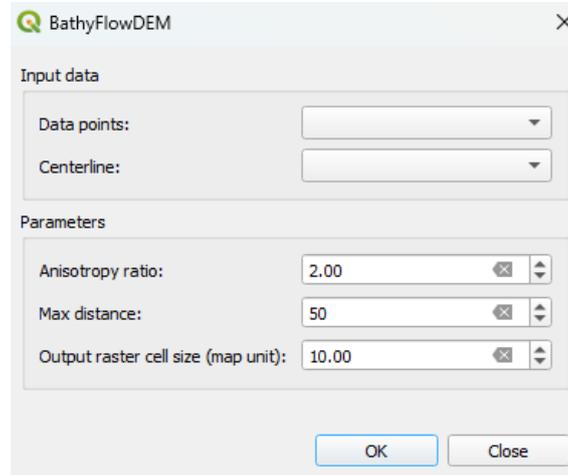


Figure 17: Main GUI of BathyFlowDEM's minimal working version.

Table 3: Minimal requirements of the first version

Requirement	Type	Description	State
Qgis integration	Set up	BathyFlowDEM should be seamlessly integrated into QGIS. This includes being accessible from the Plugin menu and from a button in the tool bar.	Done
Coordinate transformation	Main algorithm	A core function that transforms points from a (x,y) coordinate system to a flow-oriented (s,n) coordinate system.	Done
Interpolation algorithm	Main algorithm	A main interpolation algorithm that uses anisotropy.	Done
User defined cell size	User input	Let the user decide the cell size of the result raster	Done
Create an output raster	Utilities	Creates a DEM surface with the results of the interpolation	Done
Save to temporary layer	Utilities	Save results to a temporary layer in the Qgis project by default	Done

Table 4: Complete requirements of the second version

Requirement	Type	Description	State
Integrated Help tab in GUI	UX	Some information about the BathyFlowDEM functionalities should be shown within the plugin's GUI.	Done
Choice of interpolated attribute	UX	Let the user select the field to interpolate.	Done
Communication	UX	Communicate with the user through the QgsMessageBox for Errors, Warnings and Success messages.	Done
Progress bar	UX	Integrate a progress bar.	Not done
Different interpolation methods	Main algorithm	Add more interpolation methods using anisotropy so that the user can test and compare.	Not done
ROI	Refining interpolation	Add boundary layer input, to restrict the interpolation to a specific area.	Done
Data validation with RMSE	Data validation	Return the RMSE as a data validation tool	Done
Unit testing	Plugin stability	Implement unit testing for code stability and longevity	Not done
Save options	Utilities	Add standard QGIS saving options: save as temporary layer or save to folder.	Done

6.2 Definition of requirements and implementation of version 0.2

The second version's requirements are listed in Table 4. This includes some GUI improvements and adds user inputs such as a boundary layer for the interpolation and a choice to save the results to a temporary layer or to a folder. Moreover, this version should integrate different interpolation methods that includes anisotropy. Some data validation method such as calculating and returning the RMSE should also be implemented. Lastly, in order to improve the program's stability, code management and development, unit testing and error management should be implemented.

As seen in the right-side column of table 4, not everything could be done. What could not be implemented was added to the Github repository as an issue with the tag "*Enhancement*".

6.3 User tests

BathyFlowDEM has been used and tested to different extent during the development. The goal is to get user feedback about the plugin functionalities, GUI, ease of use and to gather any other comments that can arise. Testers:

- User 1, MacOS, 13th May 2024
- User 2, MacOS, 15th May 2024

The tests were done on the version available on the main branch of the Github repository. The main feedback was that the plugin didn't offer any help on what it does, what kind of data to input and especially how to choose the appropriate parameters for the interpolation. This was taken into considerations and BathyFlowDEM was adapted:

- **Help tab:** a *Help* tab was added. This contains a brief presentation of what BathyFlowDEM does, the distance calculations during the interpolation and the explains the different inputs and outputs types. It also links to the Github repository.
- **Tooltips:** tooltips appear in the main plugin window when the mouse is hovering over a widget (Figure 18).

The purpose of the Help tab and tooltips is to provide information to the users that might need it without affecting the user experience of those who don't.

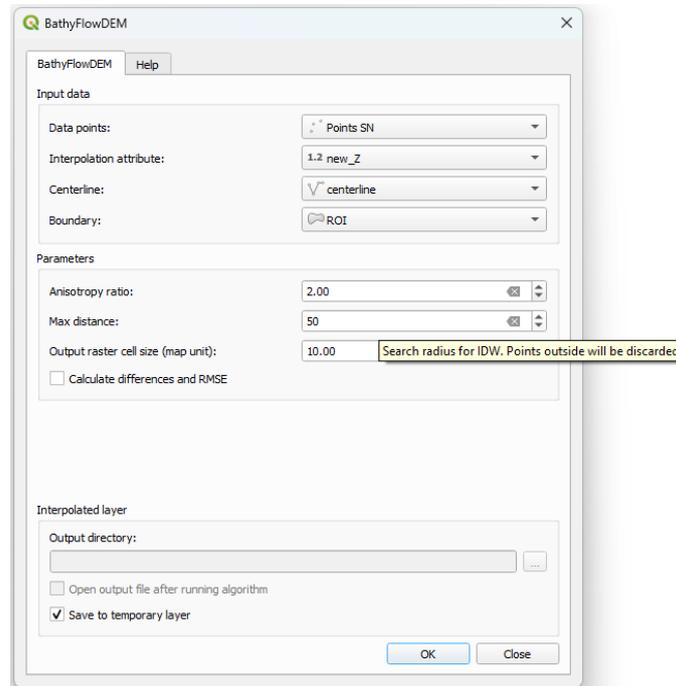


Figure 18: The tooltip that appears when hovering over the "Search radius" spinbox.

6.4 Git environment

A git environment is used to keep track of changes and manage the code. Github serves as a central repository for version control. It also sets up an efficient way to collaborate without conflicts. Two branches were used for the process, a **main** branch with the most up-to-date and tested version of BathyFlowDEM and a **dev** branch where new functions are pushed before making it to the main branch. This way, the main branch remains clean with the latest up-to-date code and new changes are first tested on the dev branch.

The repository also contains a README file where BathyFlowDEM is presented. It explains the plugin's use and functionalities, how to install, the limitations and how to use it. The purpose is that a potential user finds all the necessary information either within the plugin itself or in the Github repository.

A bug tracker is also part of the repository. This allows developers and users alike to report bugs, ask questions or request an enhancement (Figure 19).

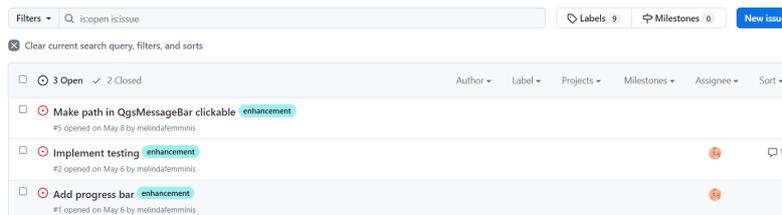


Figure 19: BathyFlowDEM's issues in the Github repository.

7 BathyFlowDEM plugin

This section introduces the plugin workflow, explains how the code is structured into different modules and cites the licences.

7.1 Plugin's functionalities

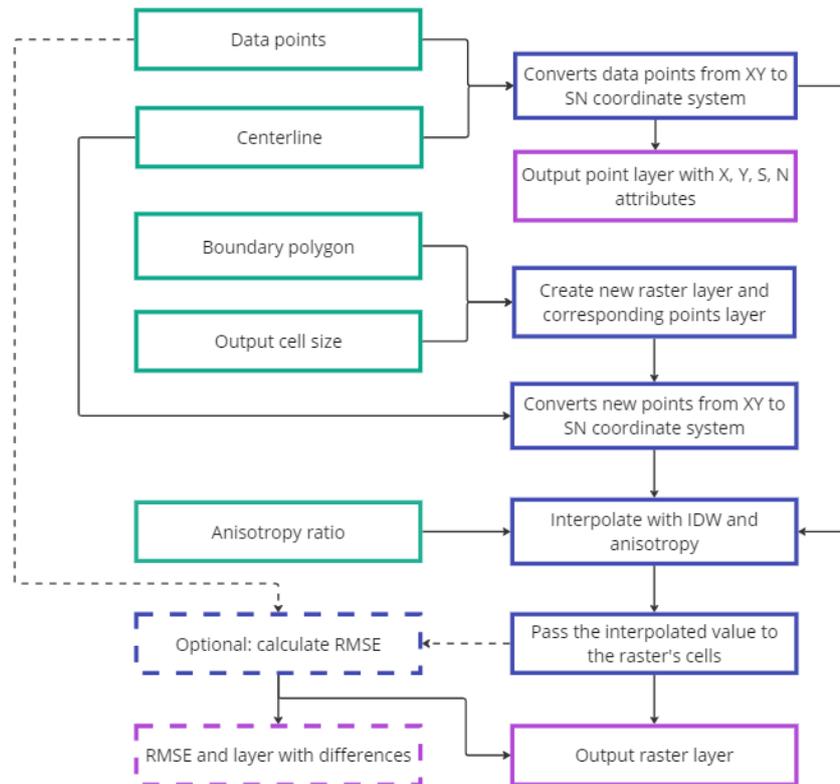


Figure 20: Pipeline of BathyFlowDEM process with user input data in green, plugin processing steps in blue and outputs in purple. Dashed line means optional.

The Figure 20 highlights the main steps of BathyFlowDEM. All the inputs are mandatory. The RMSE calculation is optional and can be deactivated via a checkbox in the plugin's GUI.

First, the input data points are assigned (s,n) coordinates. The detailed procedure is explained in Section 3 - Flow-oriented (s,n) coordinate system. A temporary layer with the input data point's s and n coordinates is loaded in the QGIS project. This is so the user can see each point's new

values and potentially use them for other processes. Then, a new raster layer is created with the input polygon's extent and the user-defined output cell size. A new vector point layer is created with points that are in the center of each raster cells. This new point layer is converted to s,n coordinate system using the same center line as the input data points.

The boundary layer is used to create the final raster layer and the points to interpolate. Those new points will all fall within the Region Of Interest (ROI). However, all input data points within the chosen radius for the interpolation will be used, even when outside of the ROI (Figure 21).

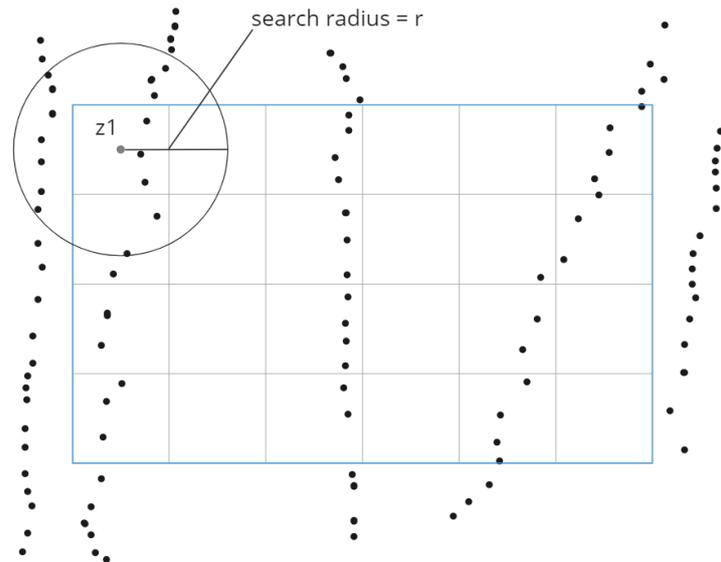


Figure 21: The interpolation algorithm will use all input data points that fall within the defined search radius - even when outside the boundary polygon.

The next step is interpolation. For each point representing the center of the raster cells, the IDW interpolation method with anisotropy is applied as detailed in Section 4 - IDW with anisotropy. Then the interpolated value is passed to the corresponding cell and the final DEM returned to the user either as a temporary layer or saved to a folder.

Optionally, the plugin can calculate data validation values. If desired, it creates a vector point layer containing all the data points located within the boundary polygon with their known value, interpolated value (retrieved from the final raster). It then loads the layer to the project as a temporary layer. With these values, the RMSE is calculated and returned to the user within the plugin window (See Figure 22). The RMSE is one way to assess the quality of the interpolation; it tells the average difference between the predicted values and the actual values from the input data points.

The screenshot shows a software interface with the following elements:

- Max distance:** A text input field containing the value '250'.
- Output raster cell size (map unit):** A text input field containing the value '60.00'.
- Calculate differences and RMSE:** A checkbox that is checked.
- Validation results:** A section containing a text box that displays 'Final RMSE: 0.7080067324534018'.

Figure 22: The RMSE value displayed after processing in the GUI below the *Parameters* box.

7.2 Modules and directory structure

The plugin's functionalities are split into different modules. This modular approach ensures that each module focuses on a single functionality with one of a few related functions. When the code is segmented into defined modules, it makes it easier to locate a specific section and enhances readability by avoiding having a single monolithic file.

BathyFlowDEM.py

This is the main file, where the BathyFlowDEM class is instantiated. It is the entry point for the plugin and takes care of a lot of aspects:

- When QGIS starts or the plugin is loaded, it initializes and shows in the toolbar and in the main menu so that the user can use it.
- It handles the loading and unloading of the plugin when it is activated or deactivated in the QGIS project.
- It connects the User Interface (UI) elements to the correct functionalities. It handles event with signals and slots by continuously listening for user actions and triggers the appropriate response. It will for example populate the "Attribute" field with the selected input point layer's attributes and dynamically change the content of the field when the user changes the point layer. It is also in the *BathyFlowDEM.py* file that the OK button is linked to the correct method that starts the processing.

interpolation.py

This module contains the core part of the plugin, the interpolation function.

coordinates.py

This is where the functions used to transform points from a (x,y) to a (s,n) flow-oriented coordinates are. To further enhance modularity, the file contains three functions, each taking care of a specific step of the process.

validation.py

The validation module takes care of calculating the differences between the input points' value and the final raster value at those points. It also contains a function that calculates and returns the final RMSE.

layers_helpers.py

While this module doesn't do any of the central calculations, it is used to create new layers. First the new raster that will be output as the final result, and the sample point layer with the data points whose value will be interpolated.

This way of structuring the project ensures readability as mentioned earlier, but also promotes reusability of the code. This way, it is possible to understand and reuse part of the code without having to read the whole thing. It is also easier to modify a module, which encourages collaboration. This modular structure keeps the code organized and defined, each developer could work on a specific part of the project without interfering with other's work. Lastly, having a modular structure facilitates the debugging and maintenance process. It is easier to isolate specific modules where the problem is or is likely to occur.

7.3 Licence and version

BathyFlowDEM is free and open-source. It is released under the GNU General Public License version 3 (GPL-3.0).

The source code is available at <https://github.com/melindafemminis/BathyFlowDEM>.

The plugin was developed in QGIS Long Term Release (LTR) v28. The minimum QGIS version required is v3.X. No external libraries need to be installed as BathyFlowDEM only uses PyQGIS and GDAL, which is shipped with QGIS.

8 Usage

This section provides comprehensive instructions on installing the plugin, followed by a step-by-step guide on how to effectively use it with real-world data examples. The guide will cover configuration settings and demonstrate the plugin's features using actual datasets to better understand how it works.

8.1 Installation

The plugin can be installed through the main QGIS plugin manager (Figure 23) that can be accessed through the menu **Plugin - Manage and Install Plugins...**

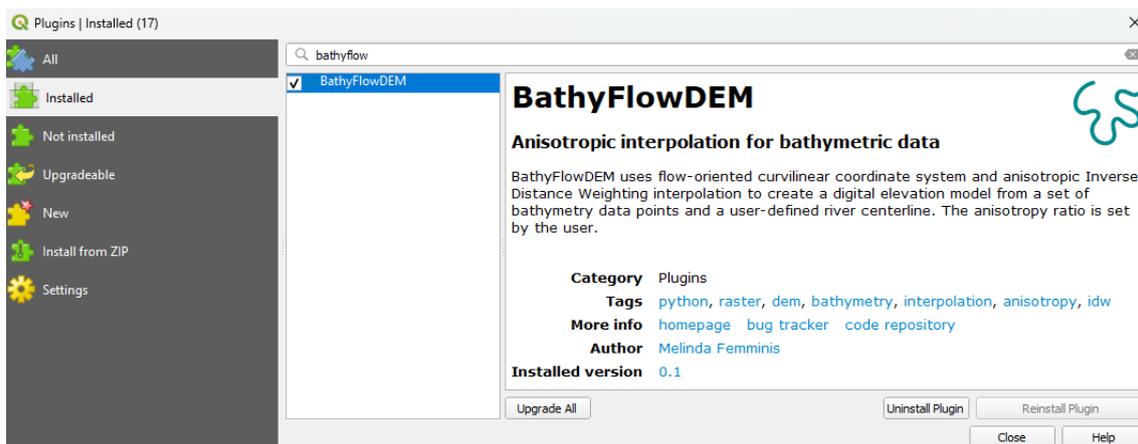


Figure 23: The QGIS plugin manager.

Alternatively, download the master branch of the repository and move it to the current active QGIS user profile folder located at:

```
.../AppData/Roaming/QGIS/QGIS3/profiles/PROFILE_NAME/python/plugins
```

To open the active profile's folder from QGIS, go to **Settings - User Profiles - Open Active User Profile** - see Figure 24).

BathyFlowDEM is now available in the toolbar and in the QGIS plugin menu (Figures 25 and 26).

8.2 Datasets

Two datasets are available in the Github repository, in the *data* folder. They are also hosted by the USGS ScienceBase and are originally in *.csv* format but were converted to *.shp* format for use in QGIS. For both dataset, only part of the data is used. For each one, a centerline and a clip

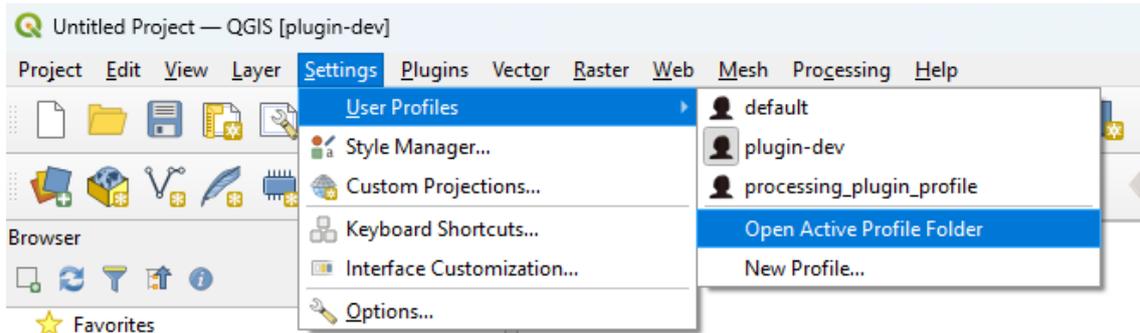


Figure 24: Show the plugins folder location in explorer for the current user profile.

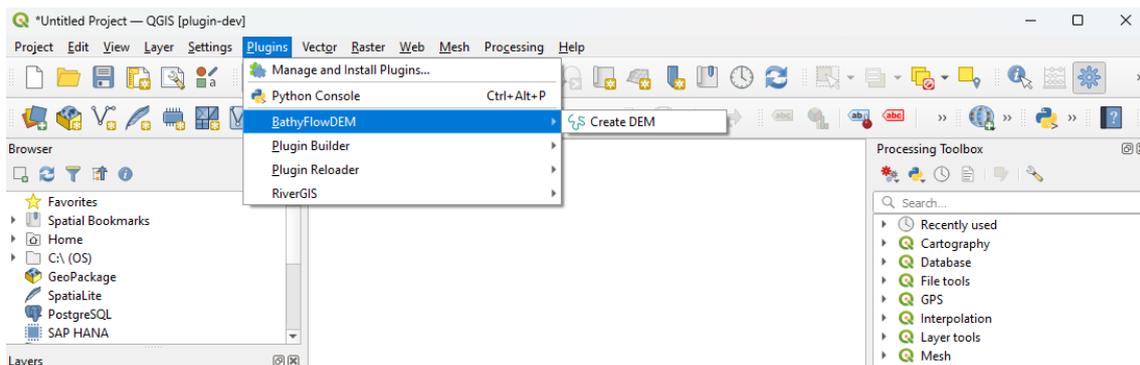


Figure 25: Accessing BathyFlowDEM through the menu

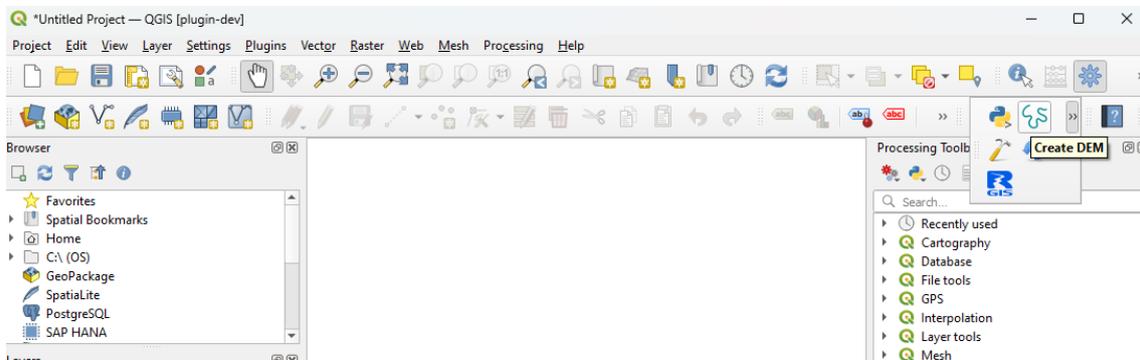


Figure 26: Accessing BathyFlowDEM through the toolbar icon

layer was created.

One is from the Sacramento river (Legleiter & Harrison, 2019) and the other from the Maumee river (Jackson & Vonins, 2022) (Table 5).

Table 5: Datasets available in BathyFlowDEM/data

Name	Bathymetric data for Maumee River between Defian and Toledo, Ohio	Depth measurements from the upper Sacramento River in northern California
Survey date	Summer 2019	September 2017
Publishing date	2022	2019
CRS	WGS84 lat-long, NAVD88	UTM Zone 10 N and NAD83
Cross-sections	251, with additional stream-wise sections in-between	10 cross-sections surveyed multiple times each
Equipment	Collected with an acoustic Doppler current profiler (ADCP)	ADCP-based depth measurements
Purpose	Develop a one-dimensional hydraulic model	Support research on salmon habitat and geomorphic changes along the river

For the rest of this guide, the Maumee dataset (Jackson & Vonins, 2022) is used, with the corresponding centerline and clip layers. In the subset of the data that will be used for this example, the cross-sections are surveyed 270 to 330 meters apart from each other and the width of the river is between 250 and 700m wide. This specific stretch of river, shown in Figure 27, has three curves and the water flows north-east to Lake Erie.

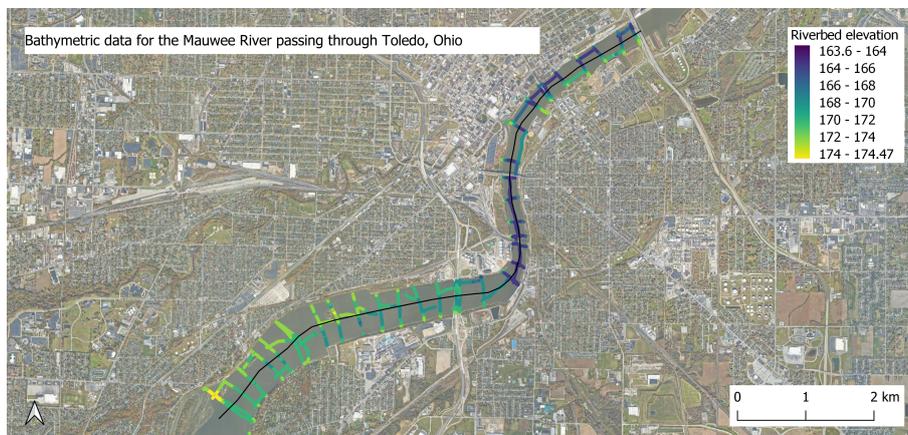


Figure 27: Bathymetric cross-sections and stream-wise data points of part of the Maumee River. Background Google Satellite 2024

8.3 Using BathyFlowDEM: a practical example

Open the plugin through the menu or the toolbar icon. BathyFlowDEM will open on the main tab (Figure 28) and automatically pre-select layers if some are already loaded in the project.

Note: if no layer can be selected in the field, it might be because the input layer's type isn't correctly set up. The "Data points" field can only accept point vector layers, the "Centerline" only accepts line vector layers and the "Boundary" field only accepts polygon vector layers. The "Interpolation attribute" field will update automatically to show the fields from the "Data points" layer.

In the "Parameters" box, some placeholder parameters are already set and can be changed to best fit the data.

- **Anisotropy ratio:** this will define the influence of the distance along the n-axis. Neutral is 1, which means no anisotropy. Less than 1 and the distance on the n-axis will be reduced, more than 1 it will be increased.
- **Max distance:** this is the search radius. All points that are closer than this maximum distance *after* applying the anisotropy will be taken into account for the interpolation.
- **Output raster cell size (map unit):** this is the pixel size of the output raster.

Those parameters work together and need to be fine-tuned according to the data. There is no exact method to determine the best ones. In the plugin, since the anisotropy ratio is used to augment the distance between the target and some of the points, it is good practice to have a big search ratio. Another thing to consider is the distance between cross-sections. If they are 200m apart, the search radius should be at least 100. If it is too small, there is a risk that there will be no points within the radius and the value cannot be interpolated. A dataset with a lot of points such as data surveyed with a MBES can have a much smaller cell size and search radius than cross-sections surveyed with a SBES 100m apart from each other. For the Maumee dataset, the chosen parameters are **Anisotropy ratio = 3, Max distance = 250, Cell size = 60**.

Open the Help tab (Figure 29) to get some inputs on what kind of data each field accepts and an explanation on the anisotropy ratio. See Table 6 for a complete list of the inputs.

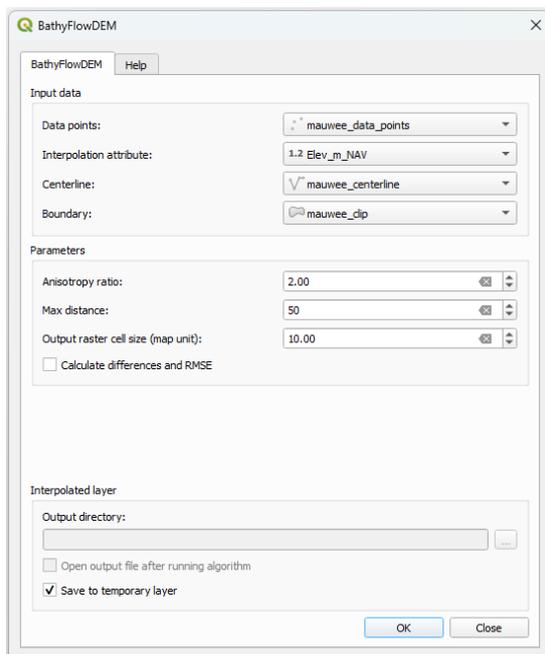


Figure 28: Plugin's main tab

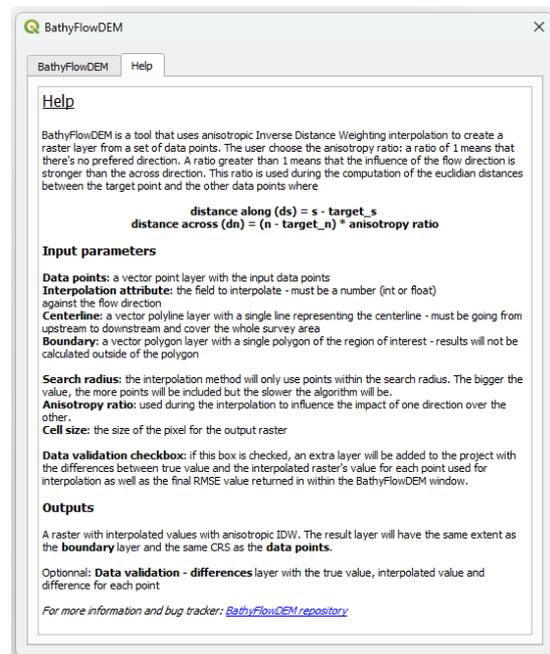


Figure 29: Plugin's help tab

Figures 28 and 29 show the GUI of BathyFlowDEM. The first tab is the main one, where the user can add the input layers for processing and decide of the different parameters. At the bottom, different options to save the final raster layer. The help tab is intended as a quick reference and reminder to explain what the plugin does. This is mostly to help set the plugin specific parameters that are the anisotropy ratio, search radius and output cell size. For further explanations, it redirects the user to the main repository.

With those parameters and data, the final RMSE is 0,7. RMSE is a way to determine the best parameters for the interpolation: after creating a surface multiple times with different anisotropy ratio, max distance and cell size parameters, the most appropriate set of inputs is the one that returns the lowest RMSE value.

Table 6: Input data and parameters of BathyFlowDEM

Data	File format	Comment
Bathymetry points	shp	Surveyed bathymetry points with at the least the attribute to interpolate
Field to interpolate		Choose from list
Centerline	shp	Line vector layer containing a single line going from upstream to downstream
Boundary	shp	Polygon vector layer with a single polygon delimitating the area to interpolate
Anisotropy ratio	int	Used to set the impact of the anisotropy
Search radius	int	Search radius to be used by the IDW
Cell size	int	The output raster's cell size in map units

9 Discussion

This section will evaluate and discuss BathyFlowDEM's functionalities, capabilities, usability and integration but also go over the limitations that as well as some possible future developments.

9.1 Evaluation of the plugin

9.1.1 Functionality

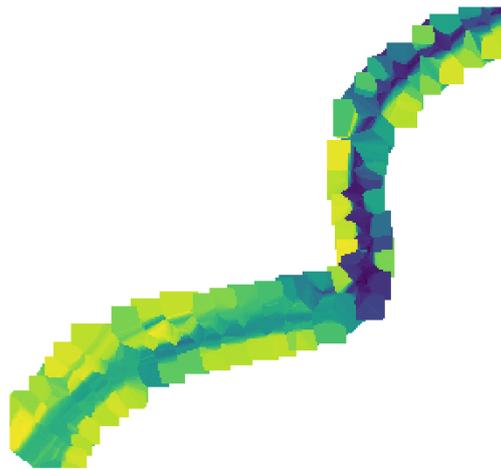
The plugin offers interpolation capabilities using anisotropic IDW, allowing users to generate raster surfaces from point data. This is the core functionality of BathyFlowDEM and can be used in various applications such as bathymetry, sediment transport analysis, and river channel morphology studies.

The efficiency of this method can be assessed with the returned RMSE. This data validation option returns the performance over a specific dataset. Another way is to look at the interpolated surfaces for discrepancies and errors. Figure 30 shows the visual results of different interpolation methods. In (a), the one created with NN, the search radii are visible and it wouldn't make sense as an underwater topography. Each cross-section is visible in the IDW surface. The results of TIN interpolation then rasterized look more plausible, but we can see the BathyFlowDEM (d) is overall smoother. This is especially visible in the first curve. This is where a cross-section seems to be missing; there is less data points and IDW with anisotropy creates a smoother surface even there.

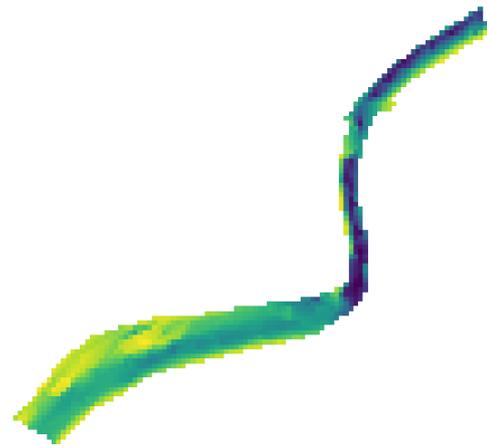
The project would benefit to have more interpolation methods added to it. This way, users can try and test the different algorithms available and decide which one is the most efficient for the present dataset. Having those methods in a same plugin will make it unique and useful as it will be a way to quickly compare them. At the moment, while a new anisotropic interpolation method available within QGIS is a great addition to the ecosystem, it is not possible to directly compare with other methods as different workflows and tools will need to be used.

9.1.2 Usability

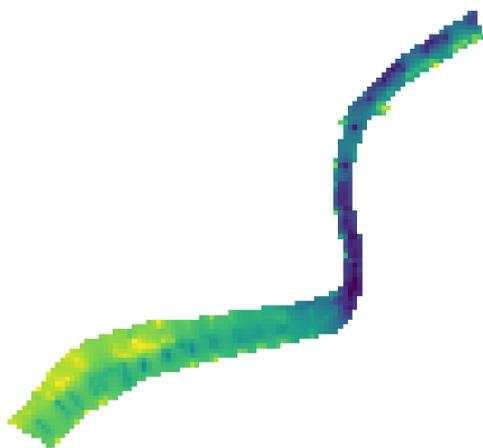
In the usability evaluation of the plugin, different aspects are taken into account to ensure that it efficiently meets the needs of the users. First the GUI was designed to be intuitive and user-friendly. It incorporates familiar elements and looks like other QGIS's tools. See Section 9.1.3 for more details. The interface is built so that the workflow is clear without having to add too much information that can get in the way. This is made possible by grouping similar inputs and restricting the kind of data that can be chosen, ie the input data points layer field is set to only accept point vector layers. It prioritizes preventing user errors before they happen, rather than having to handle them afterwards. This approach enhances efficiency and usability by guiding the users to make the correct choices. A help tab, tooltips and documentation is still available if needed and can be accessed directly from within the tool. Overall, the plugin is easy enough to understand for a user that might not have extended knowledge of geospatial data, and at the same time provides some customization options with the input parameters.



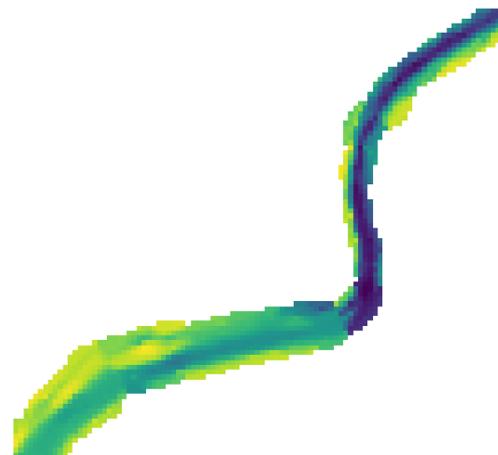
(a) Surface computed by Nearest Neighbor interpolation



(b) Surface computed by TIN interpolation, then gridded



(c) Surface computed by IDW



(d) Surface computed by BathymetryFlowDEM

Figure 30: Comparison of four algorithms used to compute a raster of 60x60m for the Maumee river.

9.1.3 Integration to QGIS

BathyFlowDEM integrates well with QGIS, offering accessibility via the main plugin menu as well as from the toolbar, like other tools within the software. With PyQt5, the GUI maintains consistency with QGIS algorithms and functions, ensuring a familiar user experience. In instances where specific widgets and methods were not directly accessible, they were created in BathyFlowDEM to have the same look and functionality than the original ones. This was done twice.

- **Save options for the interpolated raster:** while the exact saving options used by processing algorithms (Figure 32) could not be replicated in BathyFlowDEM (Figure 33), different widgets were used together to create a similar feel. Both offer three possibilities: save to a temporary layer, save to folder and load to project, save to folder and don't load to project. In both cases, the result is saved as a temporary layer if the user doesn't change the base settings. To ensure that a folder path for the results cannot be chosen while the "Save to temporary layer" option is checked, method was implemented that inform the user and prevents the processing from starting until the user makes the necessary changes.
- **Communication with the user:** the plugin uses the `QgsMessageBar` to display *Success*, *Warning* and *Error* messages to the user (QGIS.org, 2024). This was used in BathyFlowDEM

within the plugin to inform the user of input data changes when needed and to inform that the results have been exported. QLabels were added to it with a text message and a link to make it similar to the QGIS messages in the main window (Figure 31).

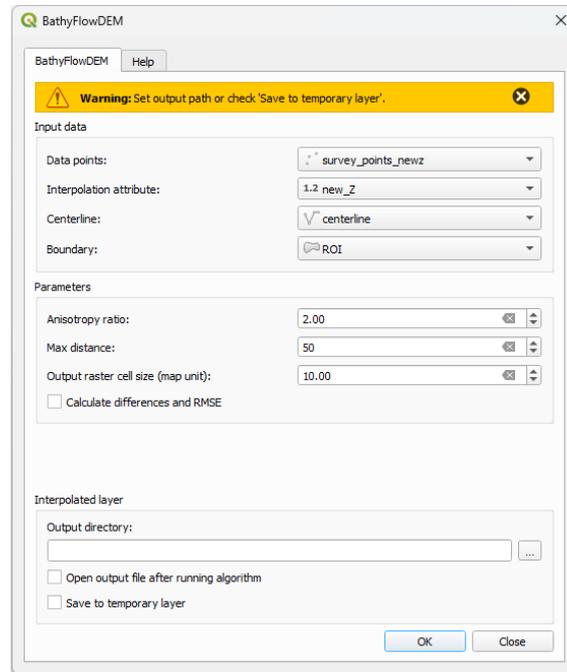


Figure 31: A warning message shown in the QgsMessageBox within the plugin if the user tries to start processing with conflicting saving choices.

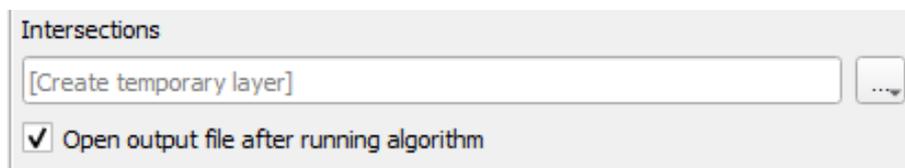


Figure 32: QGIS processing algorithms saving options

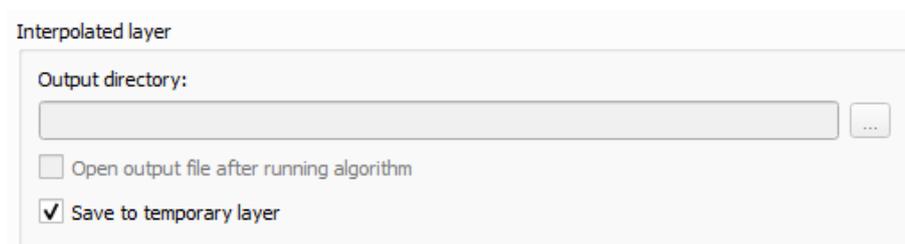


Figure 33: BathyFlowDEM’s saving options

9.1.4 Performance

Performance isn’t optimal in BathyFlowDEM; it can take minutes to process and create a DEM, which can be more time than the user agrees to wait. This is especially true if multiple surfaces need to be created in order to test different parameters for a specific dataset.

Spatial indexes is a way to significantly improve the performance when doing of queries to a vector layer - such as interpolation (QGIS.org, 2024). In BathyFlowDEM, the interpolation function first compute the distance from the target point to all other data points, then select the ones that fall into the search radius. This works well for a dataset of a few hundreds points but becomes increasingly time-consuming when dealing with bigger datasets. Spatial indexes are a way to "preorder" points and save time during the distance calculations.

Here are the steps when using a spatial index in QGIS:

1. Load features to a index using QgsSpatialIndex or QgsSpatialIndexKDBush class: `index = QgsSpatialIndex(point_layer.getFeatures())`
2. Query the spatial index. QgsSpatialIndex has a `nearestNeighbor()` method to get the X closest points and a `intersects()` method to get all points within a polygon. The QgsSpatialIndexKDBush has a `within()` method to get all points within a circle defined by its radius.
3. Use only a subset of points when calling the main interpolation method.

The downside with spatial indexes in the scope of this thesis is that the methods used to select a subset of points are made to work in a defined coordinate system. So first it will select the points within a certain radius in the (x,y) coordinate system, then compute the new value with `eidw()` in a (s,n) coordinate system. With these steps, it is possible that some points that should be included in the interpolation will not be included. This downside can slightly be balanced by choosing a value (search radius for the `within()` method and number of neighbors for the `nearestNeighbors()` method) that is high, and therefore less likely to miss data points. Table 7 compares the performance of using QgsSpatial index with `nearestNeighbor=2000`, QgsSpatialIndexKDBush with `within=max_distance*3` and no spatial index at all. The anisotropy ratio is kept the same as it will not impact the time of execution.

Table 7: Comparison of processing time and rmse with spatial index and without

SpatialIndex	Dataset	Aniso.	Max dist.	Cell size	RMSE	Time (min)
None	Maumee	3	250	60	0.7	27.1
None	Maumee	3	200	80	0.89	15.8
None	Maumee	3	100	120	1.29	7.4
QgsSpatialIndex	Maumee	3	250	60	0.7	7.9
QgsSpatialIndex	Maumee	3	200	80	0.89	4.5
QgsSpatialIndex	Maumee	3	100	120	1.66	2.13
QgsSpatialIndeKDBush	Maumee	3	250	60	0.5	0.6
QgsSpatialIndeKDBush	Maumee	3	200	80	0.79	0.25
QgsSpatialIndeKDBush	Maumee	3	100	120	1.66	0.22

While the results clearly show that using a spatial index is much more time-efficient, it is needed to be able to use them in a (s,n) coordinate system before implementing it in BathyFlowDEM.

9.2 Impact and application

With BathyFlowDEM, more methods are available to GIS analysts or environmental scientist to create surfaces from point data. Below are a couple example use cases where the plugin brings some added value to a workflow.

Sediment management in dam reservoirs

A hydroelectric power company responsible for the exploitation of a dam is facing a critical issue with sediment accumulation in the reservoir. Over time, the sediments carried down by the river from upstream areas have been settling in the reservoir and gradually affecting its capacity. With

more sediments there is less water for the same water level, impacting operational efficiency and safety. To address this problem, a flushing operation is planned where the diversion tunnels will be fully open for a few hours in order to release large volumes of water and wash away the accumulated sediments. To effectively measure the success of this operation, the company needs accurate data on the sediment volume before and after flushing.

To get this, two bathymetric surveys are commissioned: one before and one after the flushing. These involves collecting data with an SBES following predefined cross and longitudinal sections and process it. The resulting data points are then post-processed to create a surface raster map. It can be created by using BathyFlowDEM developed as part of this master thesis project. It has the advantages to be a reproducible interpolation method and to be quick and easy to install. While other interpolation methods are available within QGIS, BathyFlowDEM is particularly appropriate as it takes the anisotropy inherent to the geospatial data into account, helping to create accurate results without making the post-processing step longer. In the end, this will help the company assess the situation and take appropriate decisions.

River navigation safety assessment

A river corridor might soon be open to commercial and recreational vessel traffic. In order to assess the conditions and ensure safe navigation, the authority that oversees the project need to know more about the underwater topography. This will help to assess navigation hazard, prevent accident as well as safeguard the environment.

BathyFlowDEM can be used to create a surface from the bathymetric data collected on site. This surface can then be used to better understand the area, to derive contour lines that will be used to create navigation maps or to create visualization of the area to inform the public. Using an interpolation method can give more accurate results, enhancing the final map accuracy.

The plugin was developed with river bathymetry in mind but can also be applied to any data that has some anisotropy. This includes creating topographic maps of glaciers areas where the melting water creates surface anisotropy, mapping the soil nutrient distribution that might be affected by irrigation patterns and winds or geological mapping. As long as the anisotropy can be defined by the user with a vector line layer, BathyFlowDEM's interpolation will work.

One of the goal of the project was to develop a plugin that strikes a balance between ease of use and enhanced accuracy. The intuitive design and seamless integration into QGIS makes it easy for all to test it and maybe keep using it. The impact is not only on the data results but on the new choices it offers to users that need to integrate anisotropy into their data processing workflow.

9.2.1 Limitations

When interpolating cross-sections to create a raster surface, the sections might be far from each other. Merwade et al. (2008) define that 10-15 times the channel width apart is too much. This leads to unrealistic results as the search neighborhood needs to be stretched a lot. For better results with BathFlowDEM, the input data should be from MBES or cross-sections that are not more than 10 channel width apart.

The plugin might give erroneous results if the river is very sinuous. As the algorithm calculates the N (across the flow) coordinate as the closest distance from each to the centerline, it can become a problem in cases where a river has meandres (Figure 34).

Another limitation inherent to the way the algorithm is defined is that BathyFlowDEM can only be applied to single channel or river. When working with braided rivers and multiple centerlines, the area needs to be split and processed in steps.

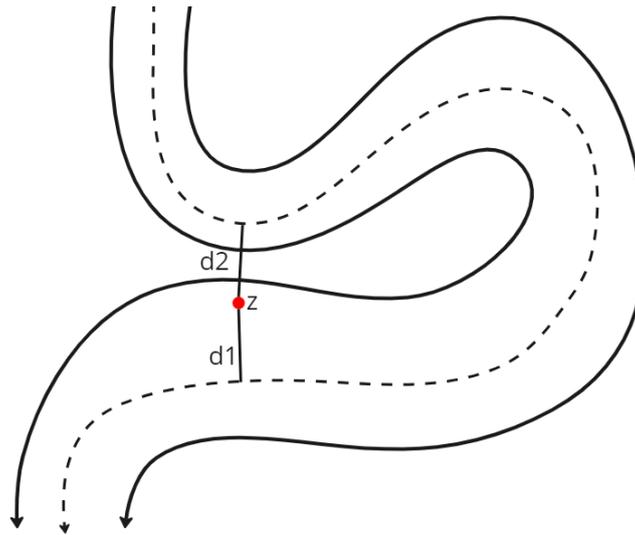


Figure 34: N is defined as the shortest distance to the centerline. Here $d1$ will be saved instead of the correct distance, $d2$.

9.2.2 Future developments

This subsection lists some possible future developments for the plugin. Some of them aim to answer the different limitations discussed earlier while others bring new options and inputs. The first part is about some new functionalities to develop or major changes. Below that are listed some improvements, GUI design modifications and smaller changes that could be implemented more easily.

- **Centerline definition:** at the moment, the centerline is an input that is fully user-defined. This can be a good thing when the user has some prior knowledge of the area Krüger et al. (2008) use. Another solution is to derive the centerline from the input data. As Merwade et al. (2005) present, the centerline is derived from the lowest point of the bathymetry data points as the thalweg. This method is interesting as it strictly only uses the surveyed data and potentially eliminate bias. Having both options available in BathyFlowDEM will allow the user to select the best method.
- **Linear interpolation of cross-sections:** to respond to the problem of having cross-sections that are too far apart (see Section 9.2.1), one solution is to create more cross-sections before starting the interpolation. Merwade et al. (2008) and Duron, Cierco and Saad (2019), both propose methods to generate intermediate cross-sections that can be implemented in the plugin.
- **Different interpolation methods:** add new interpolation methods with anisotropy
- **Adding channel banks:** this aims to fix the problem of very sinuous rivers. Having an input of the channel boundaries can restrict the algorithm's possibilities when converting Cartesian (x, y) coordinates to a (s, n) coordinates.
- **Testing:** a proper testing method should be implemented to improve code quality, consistency and detect error early.

While not as big of a change than the ones listed above, some areas of the project can be improved.

- **Customize the anisotropy ratio:** the way that the anisotropy ratio is applied could be adjusted. Currently, the user-defined value is used to modify the way the distance is calculated on the n -axis. It will be even more interesting to have to separate inputs to impact the n -axis and the s -axis. This will leave more freedom to the user to adjust and modify the

weights of each data point in the interpolation as needed.

The distance of each point would be calculated by:

$$d_i = \sqrt{a(s - s_i)^2 + b(n - n_i)^2} \quad (8)$$

where:

- d_i is the distance from the target to the known point.
 - s_i and n_i are the coordinates of the known point.
 - s and n are the coordinates of the target point.
 - a is the anisotropy ratio applied to the s -axis.
 - b is the anisotropy ratio applied to the n -axis.
- **Spatial index:** using spatial indexes can significantly improve the performance of the interpolating within PyQgis (QGIS.org, 2024) by optimizing the process of querying and getting the data. This becomes especially interesting when working with big datasets.
 - **Progress bar:** adding a progress bar to give a live feedback about the advancement of the processing will enhance the user experience. This could be implemented as a simple "Busy" state, or a loading bar that is advancing until completion.

Other improvements include adding an input to have a user-defined p-value, make the boundary layer input optional for added flexibility, adding a graph that plots the residuals and letting the user choose to interpolate the data either in the Cartesian (x,y) coordinate system or in the (s,n) flow-oriented coordinate system.

Those enhancements and potential developments are listed as issues with the "Enhancement" tag in the project's Github repository.

10 Conclusion

The development and implementation of the BathyFlowDEM plugin has added new functionalities in the QGIS software and provided some advancement to the field of geospatial analysis, specifically bathymetry and hydrological studies. By using Python, PyQGIS and PyQt5, the plugin integrates well into the QGIS environment and provide an easy to use tool for performing anisotropic interpolation with IDW.

The main motivation behind this work was to address the limitations in existing tools when dealing with anisotropic spatial data especially in the context of underwater topography. The plugin's ability to incorporate directional influences into the interpolation algorithm in a user-friendly way improves the quality of the final output data. It is beneficial in real-world applications such as river management, sediment transport analysis or any field where understanding and visualizing anisotropy is important. The evaluation of the plugin's usability and functionality confirms its practical benefits.

The method implemented in the plugin is a modified version of IDW with anisotropy. It was chosen for ease of use and implementation.. In further versions of the plugin, more interpolation algorithms can be added such as local anisotropy kriging or nearest neighbor.

BathyFlowDEM is a finished product. Nevertheless, improvements and extra functionalities are always possible and listed in the section above. These include small improvements and bigger changes. As this work is fully based on open-source tools and is open-source itself, collaboration and community modifications is always welcome.

BathyFlowDEM answers the research questions as it seamlessly combines with QGIS while extending its functionalities. It is simple enough that non-expert users can use it. That way, better results interpolating bathymetric data are accessible to all. Nevertheless, as the anisotropy value can be manually set, it also gives more experienced users some freedom.

The final version of the plugin answers the research questions (Section 1.5).

1. **How can QGIS be extended with a plugin that offers anisotropic interpolation ?** BathyFlowDEM could be developed with Python, PyQGIS and PyQt5 as well as some existing plugins (Plugin Builder and Plugin Reloader) and integrates to the QGIS ecosystem.
2. **What is the balance between ease of access and complexity so that the plugin is useful to experts and non-experts alike ?** With a familiar design, intuitive use and clear documentation, BathyFlowDEM is simple and accessible to all QGIS users and scientists, students and engineers in the geospatial fields. The different customizable inputs and available code, it is also interesting to more experienced people.
3. **What anisotropic interpolation method can be implemented ?** A method adapted from Merwade et al. (2008) was used in BathyFlowDEM as the main algorithm for interpolation. However, different ones could be implemented.

To conclude, BathyFlowDEM brings a new way to use anisotropy when interpolating bathymetry data within QGIS. While it has its limitations (see Section 9), it is also open to extensions and further developments thanks to its open-source nature.

References

- Diaz, A. L., Ortega, A. E., Tingle, H., Pulido, A., Cordero, O., Nelson, M., Cocoves, N. E., Shin, J., Carthy, R. R., Wilkinson, B. E., et al. (2022). The bathy-drone: An autonomous uncrewed drone-tethered sonar system. *Drones*, *6*(10), 294.
- Glenn, J., Tonina, D., Morehead, M. D., Fiedler, F., & Benjankar, R. (2016). Effect of transect location, transect spacing and interpolation methods on river bathymetry accuracy. *Earth Surface Processes and Landforms*, *41*(9), 1185–1198.
- Hutchinson, M. (1988). Calculation of hydrologically sound digital elevation models. *Proceedings of the third international symposium on spatial data handling*, *133*, 117–133.
- Hutchinson, M. (1989). A new procedure for gridding elevation and stream line data with automatic removal of spurious pits. *Journal of hydrology*, *106*(3-4), 211–232.
- Hutchinson, M. (2011). Anudem version 5.3, user guide.
- Jackson, P., & Vonins, B. (2022). Bathymetric data for the maumee river between defiance and toledo, ohio. *U.S. Geological Survey data release*. <https://www.sciencebase.gov/catalog/item/5abaad29e4b081f61abb4d81>
- Jawak, S. D., Vadlamani, S. S., & Luis, A. J. (2015). A synoptic review on deriving bathymetry information using remote sensing technologies: Models, methods and comparisons. *Advances in remote Sensing*, *4*(2), 147–162.
- Legleiter, C. J., & Harrison, L. (2019). Remotely sensed data and field measurements used for bathymetric mapping of the upper sacramento river in northern california. *U.S. Geological Survey data release*. <https://www.sciencebase.gov/catalog/item/5abaad29e4b081f61abb4d81>
- Legleiter, C. J., & Kyriakidis, P. C. (2006). Forward and inverse transformations between cartesian and channel-fitted coordinate systems for meandering rivers. *Mathematical Geology*, *38*, 927–958.
- Legleiter, C. J., & Kyriakidis, P. C. (2008). Spatial prediction of river channel topography by kriging. *Earth Surface Processes and Landforms: The Journal of the British Geomorphological Research Group*, *33*(6), 841–867.
- Li, Z., Peng, Z., Zhang, Z., Chu, Y., Xu, C., Yao, S., García-Fernández, Á. F., Zhu, X., Yue, Y., Levers, A., et al. (2023). Exploring modern bathymetry: A comprehensive review of data acquisition devices, model accuracy, and interpolation techniques for enhanced underwater mapping. *Frontiers in Marine Science*, *10*, 1178845.
- Merwade, V. M. (2009). Effect of spatial trends on interpolation of river bathymetry. *Journal of Hydrology*, *371*, 169–181.
- Merwade, V. M. (2018). Creating river bathymetry mesh from cross-sections - arcgis plugin.
- Merwade, V. M., Cook, A., & Coonrod, J. (2008). Gis techniques for creating river terrain models for hydrodynamic modeling and flood inundation mapping. *Environmental Modelling & Software*, *23*(10-11), 1300–1311.
- Merwade, V. M., Maidment, D. R., & Goff, J. A. (2006). Anisotropic considerations while interpolating river channel bathymetry. *Journal of Hydrology*, *331*(3-4), 731–741.
- Merwade, V. M., Maidment, D. R., & Hodges, B. R. (2005). Geospatial representation of river channels. *Journal of Hydrologic Engineering*, *10*(3), 243–251.
- Oliveira Ferreira, I., & Coelho de Andrade, L. (2022). Bathymetric surveys: Improvements and barriers. *Hydro International*. Retrieved May 12, 2024, from <https://www.hydro-international.com/content/ARTICLE/bathymetric-surveys-improvements-and-barriers>
- QGIS.org. (2024). Qgis 3.34. geographic information system api documentation [Electronic document]. *QGIS Association*.
- Shepard, D. (1968). A two-dimensional interpolation function for irregularly-spaced data. *Proceedings of the 1968 23rd ACM national conference*, 517–524.