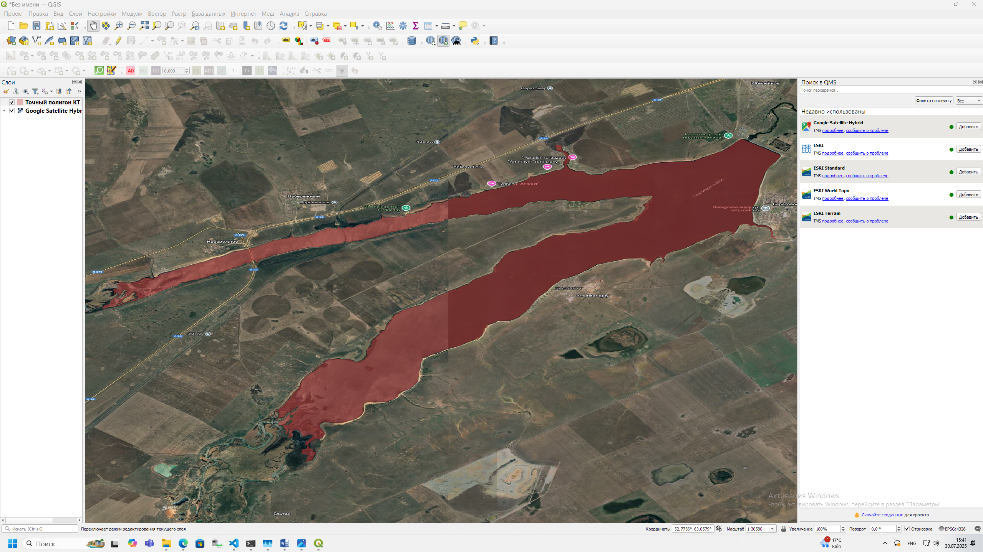
**Отчет по анализу глубины водохранилища**

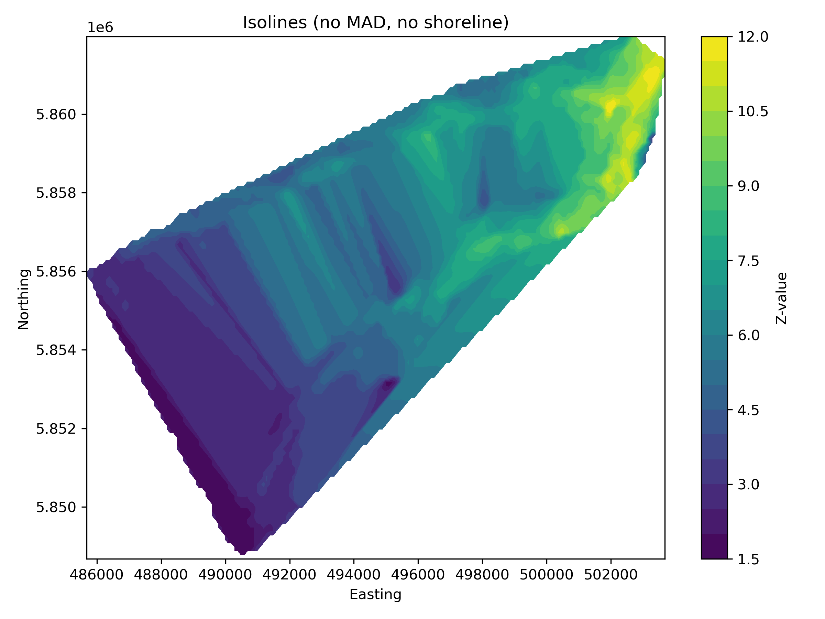
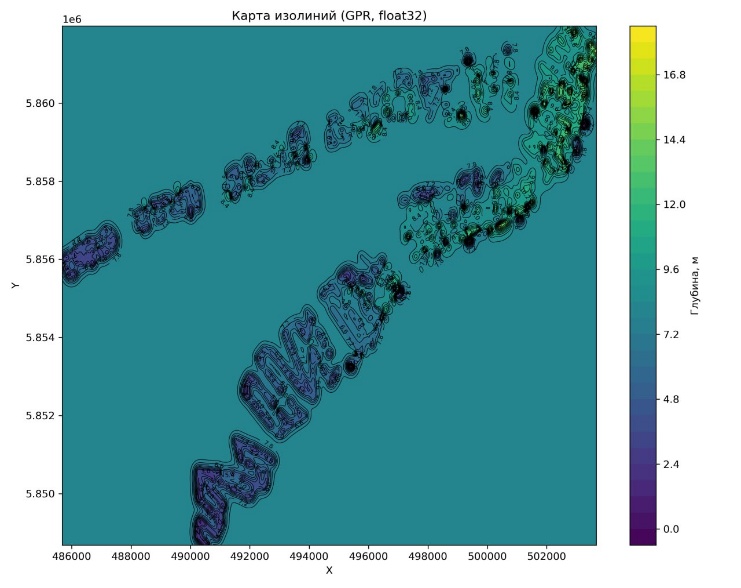
В ходе работы, на базе облака точек были разработаны методы построения изолиний, а так же написан автоматизированный код, что сам выполняет очистку данных от шума(Точек), и строит карты.  
  
**1. Очистка данных.**

Для построения карты на базе облака точек, необходимо использовать методы поиска шума и их устранения. Это является основным и самым важным этапом построение карты высот.  
 Для этого использовался локальный подход через k-ближайших соседей. Для каждой точки берется k соседей (Это значение выбирается вручную, и его можно выбирать подбором). И оценивается то, насколько данная точка, является “Нормальной” относительно ее ближайших соседних k точек. Это важна в случае неравномерного распределения данных. Так например у нас может быть точка выше глубины –z, но все точки вокруг нее равны значению -z, что может говорить о том, что эхолот например снял просто шум в виде рыбы или какого-то мусора.  
 Для оценки высоты использовалась не средняя ближайших точек, а её медиана, что является более надежным методом в контексте оценке выбросов, и это не будет тянуть все значения к центру.  
 Так же использовался дополнительный порог для устранения выбросов. В данном случае все точки чье отношение Zj-med > 3MAD(Модуль отклонения медиан(median(∣zj​−med∣))) будут отсортированы и удалены.  
  
 **В коде это реализовано следующим образом:**  
def median\_mad\_filter(df: pd.DataFrame, k=10, t=3.0) -> pd.DataFrame:  
 coords = df[["X", "Y"]].to\_numpy()  
 z = df["Z"].to\_numpy()  
 tree = KDTree(coords)  
 keep = np.ones(len(df), bool)  
 for i in tqdm(range(len(df)), desc="MAD filter"):  
 \_, idx = tree.query(coords[i], k=k)  
 neigh = z[idx]  
 med = np.median(neigh)  
 mad = np.median(np.abs(neigh - med))  
 thr = 2 if mad == 0 else t \* mad  
 keep[i] = abs(z[i] - med) <= thr  
 return df[keep]  
  
Это гарантирует то, что в результате будут исключены выбросы.

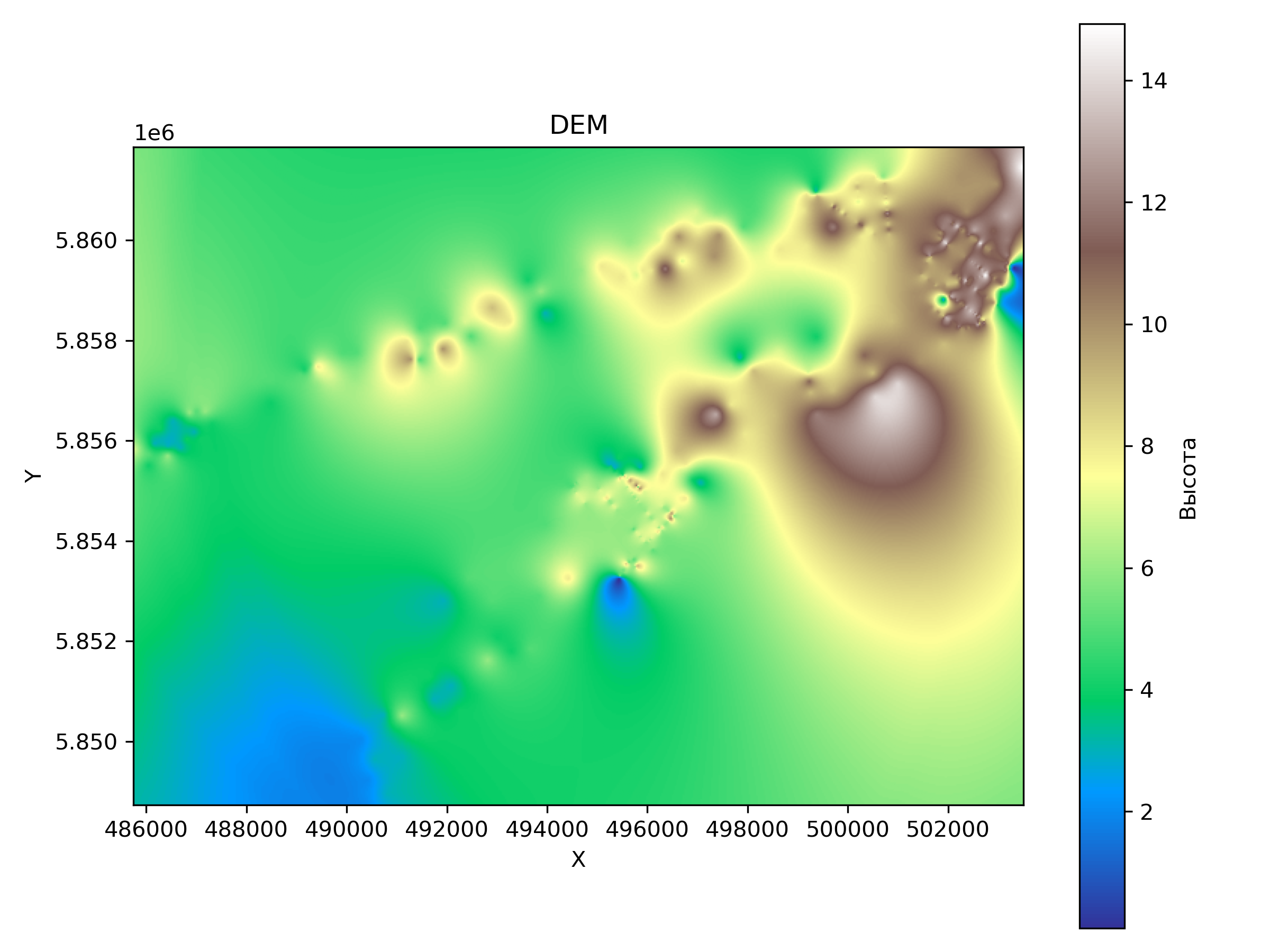
**2. Дополнение данных**

 Следующее к чему необходимо перейти это дополнение уже существующих данных. Для корректного построения карты, необходимо ограничить область постройки этой самой карты. Для этого необходимо добавить береговые точки, что станут зоной ограничения где Z будет равен нулю. Для этого необходимо создать маску области в каком-либо из редакторов, это может быть как QGIS, так и Google Earth Engine.

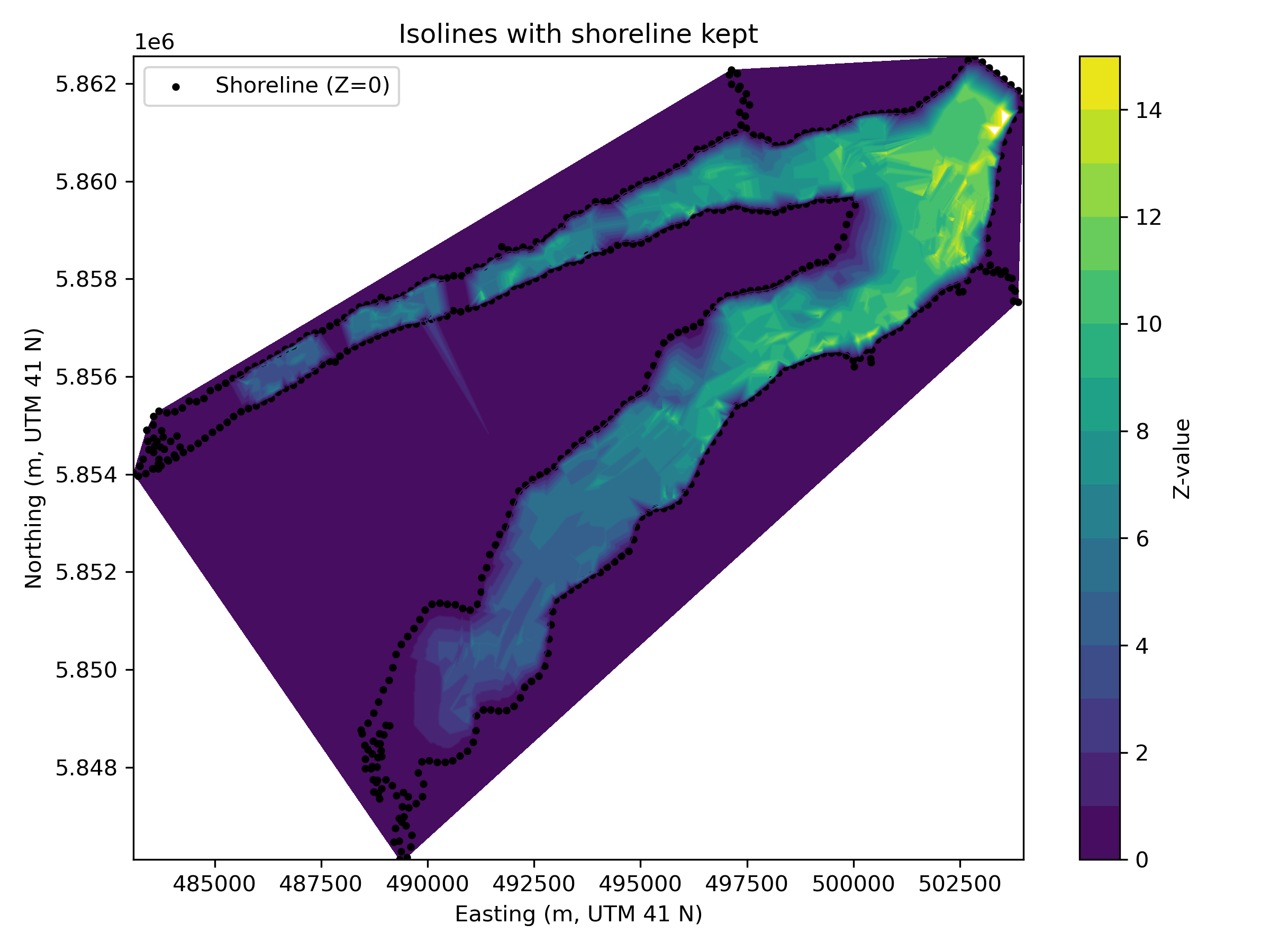
Изображение 1

 Эта маска необходима, чтобы наш результат, был ограничен, т.к. форма нашего объекта не линейна. В случае если это не выполнить, вы можете встретить подобные ошибки, как на изображении 2, 2.1  
  
 Изображение 2 Изображение 2.1

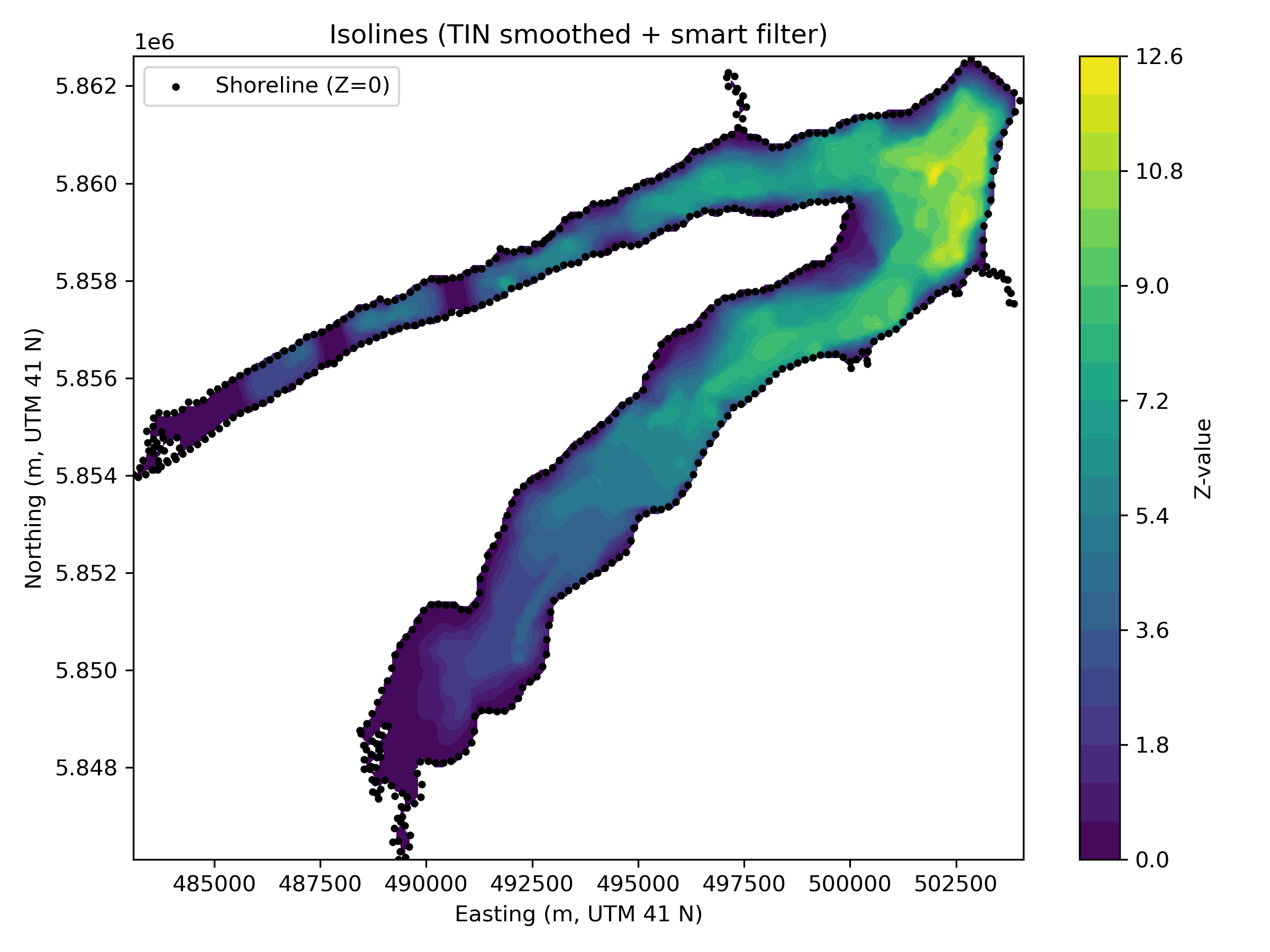
В случае, если не выполнить очистку данных, и не ограничить форму объекта, вы можете столкнуться со следующим графиком:



В данном случае пики являются более яркими, и не контролируемые зоной совершенно.  
 Результат который мы должны получить, будет выглядеть примерно вот так:

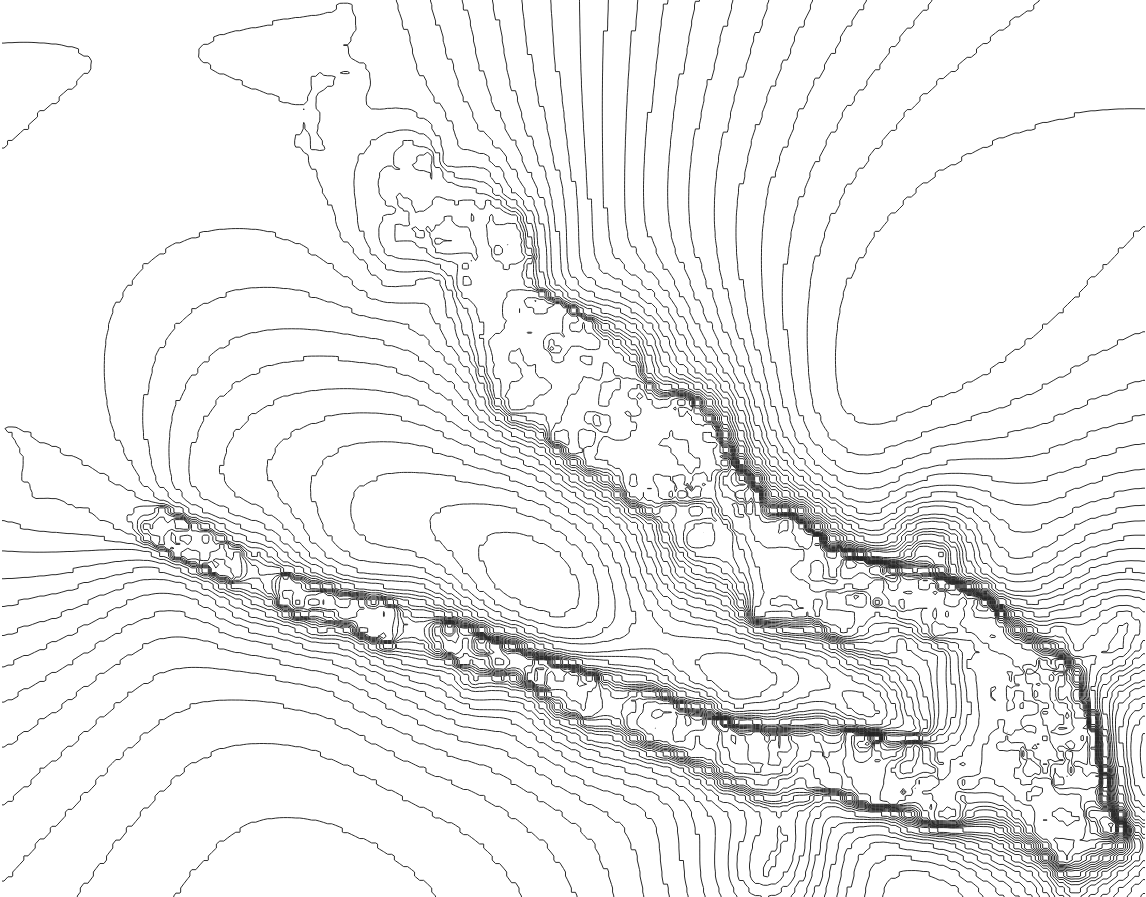


**3. Процесс построения карты** Далее для более наглядной демонстрации карты высот, нам стоит немного сгладить её, чтобы избавиться от артефактов присутствующих в моделе ранее. А так же мы можем задать гипер-параметры при создании карт. И как qgis, так и код позволяет их менять.   
K\_NEIGHBORS = 10 (Количество соседей)  
MAD\_THRESHOLD = 3.0 (Отклонение точек от медианы)  
SMOOTH\_ITERS = 2 (Количество сглаживаний)  
GRID\_STEP = 100.0 (Размер сетки)  
SIGMA\_FACTOR = 1.5 (Улучшение визуального качества и удаление артефактов(Например рваные линии, зубчатый рельеф), делая их более естественными).  
 И после запуска нашего кода, с гипер-параметрами выше, мы получаем следующую модель:  
 Так же алгоритм сам удаляет шумы из точек, и строит модель Интерполяции на базе TIN, но он так же предоставляет все очищенные точки и береговые в одном csv файле, благодаря чему, мы можем использовать эти точки, для построения других моделей, например на базе Интерполяции Кригера.



**3.1 Модель на базе Интерполяции Кригинга**

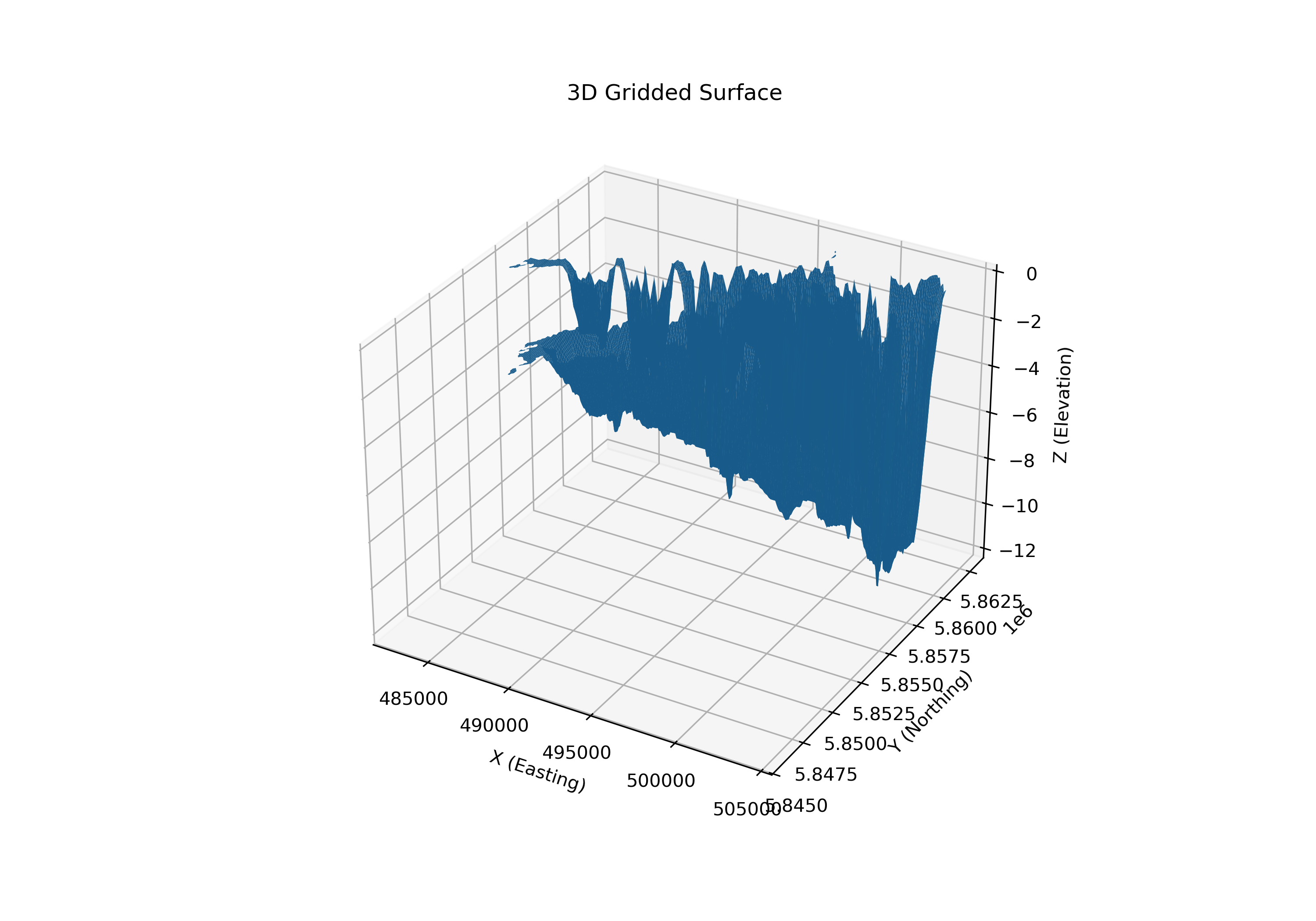
В данном модель карты изолиний, на базе Интерполяции Кригера будет выглядеть следующим образом:

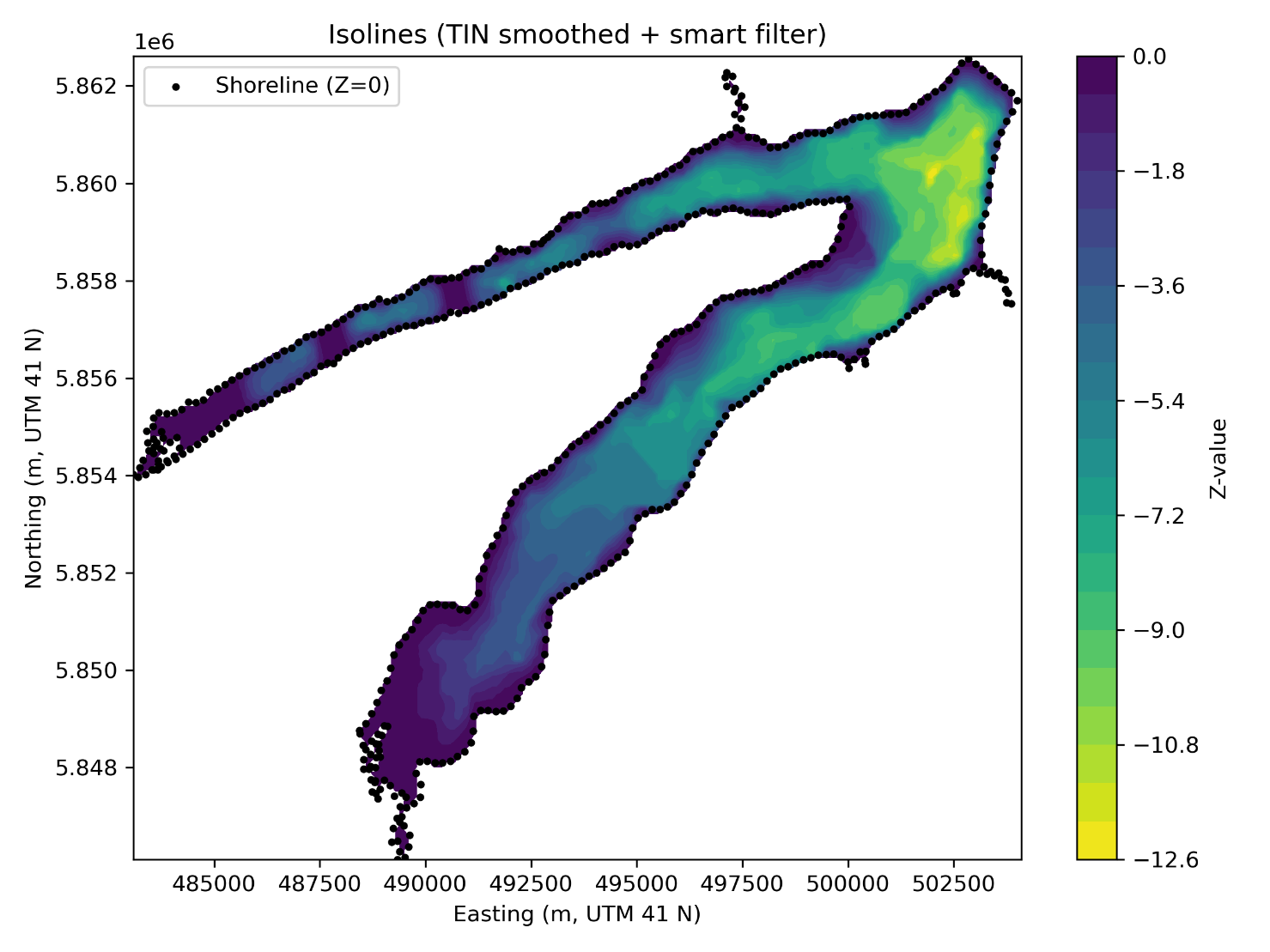


Данный метод, достаточно хорошо отражает глубину объектов, но достаточно сложен для понимания в визуализации того, насколько одно место более глубокое относительно уже глубоких, и в какую сторону идет спад высоты.  
 Для данной модели использовался другой алгоритм на базе pykriging – Python. Ordinary Kriging с грид сеткой в 100 метров. Код так же доступе в приложенном zip файле, с названием Kriging.py. Шаг изолиний модели равен 1

3.2 3D-Model на базе Каратамарского Водохранилища

И в данном случае, мы можем попробовать смоделировать 3Д модель на базе нашего облака точек, чтобы понять тенденцию глубины водоема, что соотвествует тому, что мы видим на карте изолиний. Алгоритм использованный для построения 2д карты изолиний, так же автоматизировано строит 3д модель, для более наглядного мониторинга ошибок модели, путем наблюдения пик на 3д модели, что позволяет корректировать гипер-параметры до более оптимальных.



Разработанный алгоритм, позволяет почти автоматизированно выполнить построение карты изолиний. Все что нам необходимо, это наличие полигона, из которого мы будем формировать береговые линии, а так же сам список точек. После чего все что нам нужно это запустить скрипт, указав путь к shp file, и csv file. И алгоритм автоматизированно проанализирует csv на MAD(Медианную ошибку отклонения), проверит все ли координаты в UTM формате. После чего построит карту изолиний, и при постройке будет учитывать насколько ячейка отклоняется от других ячеек рядом с собой по медиане. Код так же доступе в приложенном zip файле, с названием Analysis\_Isobars\_Full.py

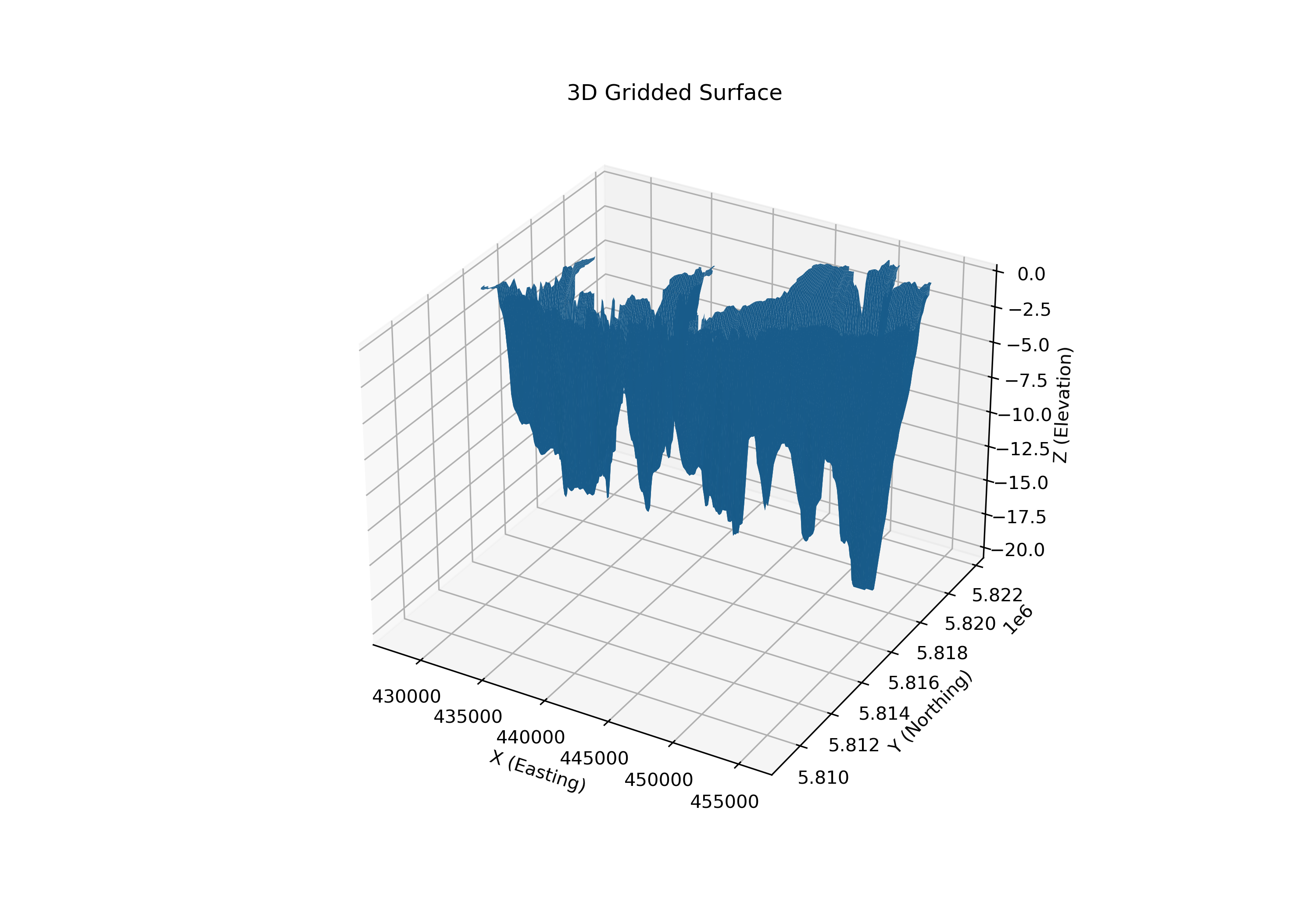
Для построения данной модели Каратамарского водохранилища использовались следующие параметры. Grid = 100, это позволяет наблюдать плавную карту без сверх-детализации, k-Neighbors – 30, MAD\_Treshhold – 3.0, Sigma\_factor – 1.5, Smooth\_Iters – 3, Количество точек для построения этой модели 80 000.

**4. Построения карты Верхне-Тобольское Водохранилище и их параметры.**

Для построения карты, использовался прежний скрипт интерполяции, а так же был создан и использован дополнительный скрипт, что объединяет все точки из 1 директории в один файл, что далее используется для построения карты далее. Это связано с тем, что запись происходит в множество разных разбитых файлов, что зависят от включения батиметра.   
  
**Зона покрытия точек в данный момент не достаточно для построения визуально приятной и корректной карты Верхне-Тобольского водохранилища !**  


Для построения использовались следующие параметры:   
Grid = 100, это позволяет наблюдать плавную карту без сверх-детализации, k-Neighbors – 30, MAD\_Treshhold – 3.0, Sigma\_factor – 1.5, Smooth\_Iters – 3, Количество точек для построения этой модели 30 000.

**3D-Model на базе Каратамарского Водохранилища**



**Модель на базе Интерполяции Кригинга**

Как и другие дополнительные материалы.  
При необходимости возможна доработка данных алгоритмов, добавление графического интерфейса для удобства пользователей, и сшитие нескольких csv в один csv путем итерации в папке, а так же проверка на дубликацию точек при сшитии. Т.к. зачастую съемка с использованием ботиметрии имеет один файл за один запуск, что создает около 8 файлов в течении дня, которые необходимо сшивать.