

## 1. Цель работы

Разработка скрипта на python, позволяющего рассчитывать радиусы кривизны произвольной полилинии (далее – скрипт)

## 2. Краткое описание входных данных

Исходными данными для обработки скриптом являются полилинии, т.е. последовательный набор точек (узлов) с известными координатами (заданы в WGS84 UTM34N, в метрах). Полилинии выполнены в формате ESRI Shape. В исходном файле может присутствовать как одна полилиния, так и более одной (чаще всего одна или две). Тестовый набор полилиний может быть предоставлен. Как были получены полилинии:

Полилинии были получены ручной оцифровкой снимков участка береговой линии (всего около 70) в разное время. Соответственно, точки (узлы) располагаются на различном расстоянии друг от друга, как линейно, так и вдоль полилинии. Узлы располагались таким образом, чтобы максимально описать форму береговой линии, в том числе небольшие выступы и вогнутости. (Пример на рисунке 1)

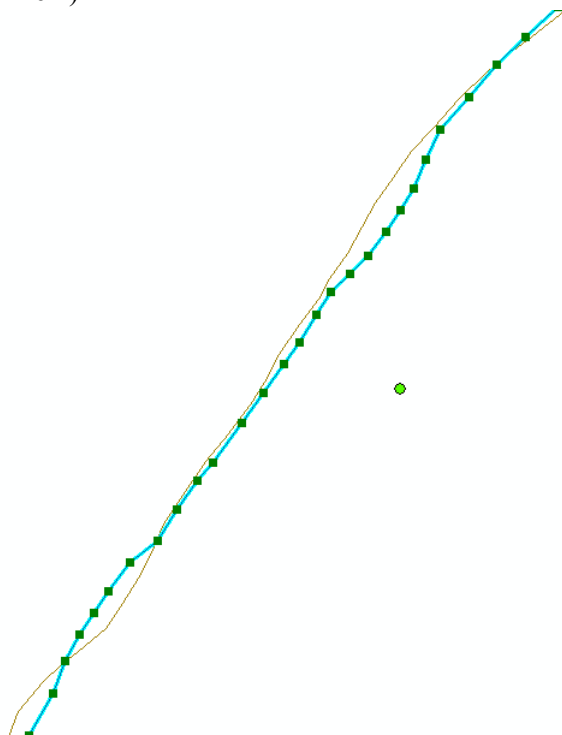


Рисунок 1 – Узлы на полилинии

На полилинии присутствуют вогнутости и выступы с очень различным радиусом кривизны, причем, на этих вогнутостях и выступах могут находиться вогнутости и выступы с меньшим радиусом. (Пример на рисунке 2)

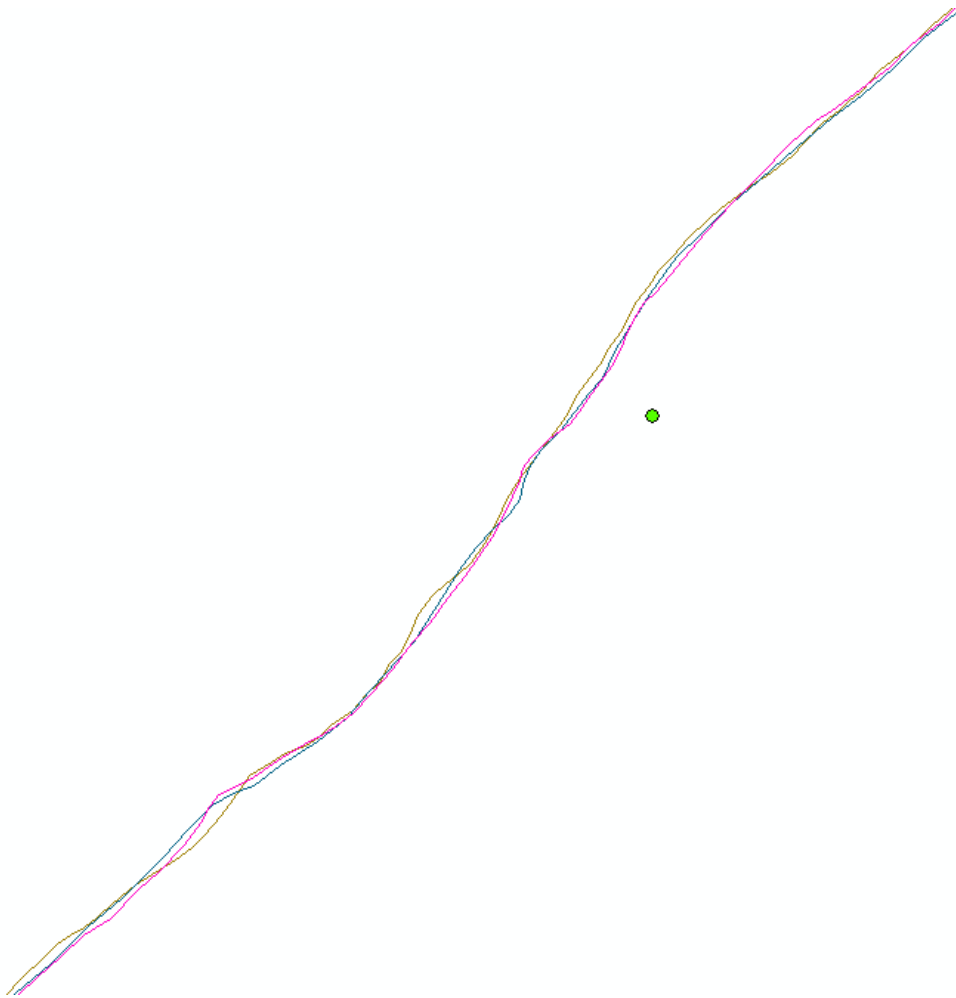


Рисунок 2 – Пример вогнутостей и выступов на трех полилиниях – береговых линиях

### 3. Основные положения

С использованием скрипта необходимо получить возможность рассчитать кривизну как больших вогнутостей и выступов, так и малых, которые могут быть вложены в более крупные. Таким образом:

3.1 Расчет локальных радиусов кривизны для полилинии. Необходимо получить радиусы кривизны для участков полилинии (как больших вогнутостей и выступов, так и малых, которые могут быть вложены в более крупные).

Приведу несколько уже существующих обсуждений проблемы:

Для расчета двигаются вдоль полилинии с определенным шагом и находят кривизну (такой подход уже был реализован в

[https://repository.nced.umn.edu/browser.php?current=keyword&keyword=5&dataset\\_id=15&folder=237802](https://repository.nced.umn.edu/browser.php?current=keyword&keyword=5&dataset_id=15&folder=237802) (данные файлы могу предоставить для изучения)

Обсуждение 2.

<https://gis.stackexchange.com/questions/127169/how-can-i-calculate-the-radius-or-curvature-of-a-meandering-river>

3.2 Дать возможность изменять количество окрестных точек, по которым производится расчет кривизны и ширину зоны расчета. Поясню: при расчете для центральной точки скользящего окна грид обычно используются соседние 2 точки – одна справа, другая слева.

Необходима возможность варьировать количество используемых точек, чтобы захватывать большее или меньшее количество соседних точек, а также варьировать расстояние поиска этих точек. Например:

- расчет по максимум 5 соседним точкам, расположенным в радиусе 100 метров;

- расчет по 4 соседним точкам; ширина захвата не указана
- расчет по точкам расположенным в радиусе 120 метров, количество точек не указано.

Очевидно, что иногда при определённой ширине зоны поиска не будет хватать необходимого числа точек. В таком случае приоритет отдается ширине зоны поиска, используется тот минимум точек, что она содержит. Предусмотреть возможность что точек будет не хватать в таком радиусе.

3.3 Реализовать расчет вписывания в точки вдоль полилинии окружностей с различным радиусом (задавать радиусы будет пользователь заранее списком в тексте скрипта или в отдельном файле) с расчетом RMSE и доли (как мер отклонения участков полилинии от окружностей заданного радиуса). И возможность включить\отключить этот расчет

3.4 Давать комментарии в коде скрипта, чтобы иметь возможность самостоятельно настраивать параметры

3.5 Использовать ту же систему координат в выходных данных, что была во входных (а не новую, условную или расстояние от начала линии). Поясню: расчеты будут проводиться для большого количества полилиний, затем будет проводиться сравнение изменений кривизны во времени и с другими показателями, для чего нужно одно координатное пространство

3.6. Необходима прозрачность используемой математики (матрицы, используемый в скрипте должен быть известен, либо описан. (

Выходные данные можно представлять в текстовой табличной форме с разделителями (например «;» или табуляцией). По структуре выходных данных такое предложение:

1. Координата x точки для которой выполнялся расчет (например x)
2. Координата y (например y)
3. Кривизна в точке (например Cu)
4. Размер ширины зоны расчета (например Bandwidth)
5. Количество точек, используемых для расчета (например Count)
- 6.7.8 и тд. – результаты вписывания окружностей (Можно назвать их R\_%указанный радиус%\_RMSE и R\_%указанный радиус%\_b , например)

#### **Дополнительные пожелания**

1. Желательна (но не обязательна) возможность загрузки полилиний непосредственно из формата Shp (думаю есть готовые решения для импорта esri shapefile). В принципе полилинию можно задавать и в текстовом виде как список узлов (x, y), но вручную переводить shp в список довольно времязатратная процедура.
2. Желательна (но не обязательна) возможность пакетной обработки (например, возможность указать директорию, содержащую файлы полилиний)
3. Желательно использовать python, поскольку взаимодействие с большинством гис (QGIS; ArcGIS идет через него и у них есть уже готовые блоки кода и библиотеки)

Спасибо. Каждый пункт готов обсуждать, методические предложения математика категорически приветствуются.

Выбор используемых математических формул – за исполнителем.

Может быть предоставлена литература, где описаны формулы и расчет кривизны на поверхности грид, если это необходимо.