

Интерполяционные полиномы

Николай Жидков

10 апреля 2018 г.

1 Краткое описание

В этом задании я строил интерполяционные полиномы с помощью метода Ньютона. В качестве сеток использовалась либо равномерная (*uniform*), либо корни многочлена Чебышева, округленные до ближайших узлов из входной сетки (*chebyshev*).

2 Структура программы

Программа разделена на функции, записанные в файле `solve.py`.

- *process_command_line_args*, разбор аргументов командной строки:
 - Ничего не принимает
 - Возвращает файл для считывания данных *filename*, флаг полного дебаг вывода *full_mode*, степень полинома *m*, способ выбора узлов сетки *grid*, флаг построения графика *plot*.
- *read(filename)*, чтение данных:
 - Принимает имя файла
 - Возвращает число точек в сетке *n*, массив точек *X*, массив значений в точках *Y*
- *uniform(n, a_, b_, m)*, рассчитывает равномерную сетку:
 - Принимает количество узлов в сетке минус один *n*, начало и конец отрезка *a_*, *b_* и размер той сетки, которую надо построить *m*.

- Возвращает индексы узлов из начальной сетки, которые мы возьмем в качестве новой подсетки.
- *chebyshev*(*n*, *a_*, *b_*, *m*), рассчитывает сетку из корней многочлена Чебышева:
 - Принимает количество узлов в сетке минус один *n*, начало и конец отрезка *a_*, *b_* и размер той сетки, которую надо построить *m*.
 - Возвращает индексы узлов из начальной сетки, которые мы возьмем в качестве новой подсетки.
- *deviations*(*X*, *Y*, *P*), считает отклонения:
 - Принимает массив точек *X*, массив значений функции в точках *Y*, массив значений в точках полинома в точках *P*.
 - Возвращает максимальные абсолютное и относительное отклонение.
- *calc_newton*(*xes*, *a*, *X*), считает значение интерполяционного многочлена, построенного по методу Ньютона, в указанных точках
 - Принимает массив точек для подсчета значения *xes*, коэффициенты многочлена *a* и массив узлов, по которым он строился *X*.
 - Возвращает массив значений многочлена в точках *xes*.
- *newton*(*n*, *X*, *Y*), вычисляет коэффициенты интерполяционного многочлена методом Ньютона.
 - Принимает степень многочлена *n*, узлы *X*, значения *Y*.
 - Возвращает значения коэффициентов в разложении $P(x) = \sum a_i(x - x_0)(x - x_1) \dots (x - x_i)$

3 Структура файлов исходных данных

Во входном файле ожидаются некоторые числа, формат которых описан дальше, при этом наличие пробелов и переводов строк между ними не важен (можно все данные задать в строку через пробел или по одному на строке, это не имеет значения).

Сначала ожидается число *n* - число узлов. Дальше идут *n* чисел - узлы сетки, потом еще *n* чисел - значения функции в узлах.

Пример входных данных

3

0.01 0.02 0.03

1 12 3.343

В результате программе примет функцию, заданную в трех точках 0.01, 0.02, 0.03 со значениями 1, 12, 3.343.

4 Примеры вызова из командной строки

Обязательные флаги (для каждого должно быть обязательно указано какое-то значение):

- `--input` = для указания входного файла (произвольная строка)
- `--deg` = для указания степени полинома (натуральное число)
- `--grid` = для выбора сетки (два варианта - *uniform* и *chebyshev*)

Дополнительные опции (по умолчанию выключены):

- `--full` или `-f` для вывода подробной информации
- `--plot` или `-p` для вывода графика (синим выводится функция, оранжевым полином)

Примеры запусков

- Строим полином 4-ой степени по точкам из файла *data* с помощью равномерной сетки. Дебаг информация не выводится.

```
python3 solve.py --input=data --deg=5 --grid=uniform -p
```

5 Численный эксперимент

5.1 Сравнение полиномов по равномерной сетке

Критерий анализа	Степень $m = 4$	Степень $m = 7$	Степень $m = 14$
Визуальное сравнение	По структуре довольно похож на данную функцию, но слишком неточен в начале	Большие отклонения в начале, начиная с середины довольно близок к функции	В середине почти везде совпадает с функцией, в начале и конце странные отклонения
Максимальная абсолютная ошибка	0.440	0.568	0.57
Максимальная относительная ошибка	0.436	0.617	0.63

5.2 Построение с помощью корней многочлена Чебышева степени 14

Максимальная абсолютная ошибка **0.04**.

Максимальная относительная ошибка **0.04**.

График очень близок к данной функции, в начале и конце имеются незначительные изгибы.

5.3 Сравнение с МНК

Перебором вариантов было выяснено, что наименьшая абсолютная ошибка достигается при построении с помощью Чебышева многочлена степени 14 (результаты выше).

Сравнивая с результатом прошлого задания, можно заметить, что при степени многочлена 14 МНК давал результат в два раза лучше (там была ошибка порядка 0.02).

5.4 Выводы

- Использование равномерной сетки не приводит к хорошему результату для данной функции. В целом график довольно похож на функцию, но есть странные пики, природа которых мне не ясна (возможно, это из-за специфики функции).
- Использование сетки по узлам Чебышева дает куда лучший результат, график очень похож на функцию, отклонения есть по краям, но не очень большие. Для данной функции немного проигрывает методу МНК той же степени (0.04 вместо 0.02).