Академический университет

Сплайны

Николай Жидков

19 апреля 2018 г.

1 Структура программы

Функции из прошлых домашних работ, краткое напоминание, что делают (выписывал основные, вспомогательные и так очевидны).

- $uniform$ - выделяет равномерную подсетку из заданной
- $chebyshevX$ - выделяет чебышевскую подсетку из заданной
- $deviations$ - считает максимальное и среднее абсолютное отклонение
- $solve$ - решает СЛАУ
- $read$ - читает сетку из фалйа

Новые функции

- $process\_command\_line\_args$, 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 $grid$, индекс пропадающего узла ex, значение второй производной на правом конце $y2b$, флаг построения графика $plot$, тип сплайна $type\_$.
- $tridiagonal\_matrix\_solution(A, f)$, решение СЛАУ для матрицы из трех диагоналей (методом прогонки):

- Принимает матрицу A и столбец f
- Возвращает решение системы

Для работы со сплайнами был сделан базовый класс $Spline$, в нем есть методы:

- $der(self, k, x)$ - считает производную сплайна на k-ом отрезке в точке x
- $der2(self, k, x)$ - считает вторую производную сплайна на k-ом отрезке в точке x
- $test(self, full\_mode)$ - проверяет, что у построенного сплайна на каждом стыке отрезков равны первая и вторая производные
- $build\_b(self, n, h, Y)$ - строит столбец b из способа, описанного на паре, n - число узлов минус 1, h - столбец разностей соседних узлов, Y - значения в узлах.
- $build\_coef\_by\_m(self, n, m, Y, h)$ - восстанавливает сплайн по вектору вторых производных в узлах и сохраняет коэффициенты в $self.a$.
- $evaluate(self, x)$ - считает значение сплайна в заданной точке.

Сами сплайны строятся в конструкторе

- $CustomSpline(self, X, Y, ind, v, full\_mode)$ - строит кубический сплайн по сетке X, Y с пропадающим узлом ind и значением второй производной в правом конце v.
- $NaturalCubicSpline(self, X, Y)$ - строит естественный кубический сплайн

2 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 проблел 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.

3 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 $chebyshev$)
- $--type=$ для выбора типа сплайна (два варианта - естественный ($natural$) и с условиями ($custom$).

Дополнительные опции (по умолчанию выключены):

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

Дополнительный опции для сплайнов со специальными условиями (это опции нельзя выставлять в естественном)

- $--ex=$ для индекса пропадающего узла (целое число от 1 до $n-1$)
- $--y2b=$ для значения второй производной в правом конце (вещественной число)

Примеры запусков

- Строим полином 4-ой степени по точкам из файла $data$ с помощью равномерной сетки. Используем естественный кубический сплайн. Выводим дебаг информацию, строим график.

  python3 solve.py −−input=data −−deg=4 −−grid=uniform −−type=natural -f -p

- Строим полином 7-ой степени по точкам из файла $data$ с помощью чебышевской сетки. Используем кубический сплайн с выпадающим узлом под индексом 1 (индексы с 0), вторая производная в правом конце 0.5. Не выводим дебаг информацию, строим график.

  python3 solve.py −−input=data −−deg=7 −−grid=chebyshev −−type=custom −−ex=1 −−y2b=0.5 −−plot

4 Численный эксперимент

4.1 Сравнение естественных сплайнов с МНК на равномерной сетке

4.1.1 $m = 4$

Критерий анализа	МНК	естественный сплайн
Максимальная абсолютная ошибка	0.33220	0.31057
Средняя абсолютная ошибка	0.10554	0.07886

4.1.2 $m = 7$

Критерий анализа	МНК	естественный сплайн
Максимальная абсолютная ошибка	0.172415466746918	0.137542053987132
Средняя абсолютная ошибка	0.058579829907618	0.019950315980158

4.1.3 $m = 14$

Критерий анализа	МНК	естественный сплайн
Максимальная абсолютная ошибка	0.025300241322584	0.024776636494779
Средняя абсолютная ошибка	0.008317548076360	0.002456577103526

4.1.4 Вывод

4.2 Сравнение естественных сплайнов с интерполяционным многочленом на чебышевской сетке

4.2.1 $m = 4$

Критерий анализа	Ньютон	естественный сплайн
Максимальная абсолютная ошибка	0.472656658026733	0.458624244617529
Средняя абсолютная ошибка	0.090111472017959	0.084612078261414

4.2.2 $m = 7$

Критерий анализа	Ньютон	естественный сплайн
Максимальная абсолютная ошибка	0.289065939302974	0.299373710649463
Средняя абсолютная ошибка	0.055918049138215	0.053522488385622

4.2.3 $m = 14$

Критерий анализа	Ньютон	естественный сплайн
Максимальная абсолютная ошибка	0.042808199730915	0.031227324704524
Средняя абсолютная ошибка	0.008767324980551	0.003035806527237

4.2.4 Вывод

При степенях поменьше методы дают почти одинаковые результаты, но далее видно, что естественные сплайны немного выигрывают по обоим параметрам.

4.3 Изучеие влияния индекса пропадающего узла, везде $y2b = 0$

4.3.1 $m = 4$

Индекс пропадающего узла	Максимальная абсолютная ошибка	Средняя абсолютная ошибка
1	0.414801535633325	0.100508348089877
2	0.629735705283550	0.146770611096861
3	0.599053884451811	0.140540707629912

4.3.2 $m = 7$

Индекс пропадающего узла	Максимальная абсолютная ошибка	Средняя абсолютная ошибка
1	0.192134778456213	0.035204547756872
2	0.735740802092888	0.097795207656577
3	1.492264068660650	0.186859865814244
4	1.057952309369165	0.135720620135911
5	0.153353961118412	0.026476226031434
6	0.567341106285179	0.077983863354257

4.3.3 $m = 14$

Индекс пропадающего узла	Максимальная абсолютная ошибка	Средняя абсолютная ошибка
1	0.024779356497023	0.002457631667002
4	2.846133072095893	0.136514243348274
7	14.904737830359689	0.699634911210021
10	5.457541124367491	0.259522887751026
13	443.543848124345686	20.895287837851132

4.3.4 Выводы

Наблюдается увеличение ошибки при приближении к середине, далее в случае 4 и 7 степенй идет уменьшение ошибок, в случае 14 получаюится какие-то странные пики в противоположном от проподающего узла конце, что очень резко увеличивает ошибку.

4.4 Изучения влияние значения, заданного на фиксированном конце, везде $ex = 1$

4.4.1 $m = 4$

Значение второй производной на правом конце	Максимальная абсолютная ошибка	Средняя абсолютная ошибка
-4	0.410914266582467	0.108145516300492
-1	0.413829718370610	0.102337583043834
0	0.414801535633325	0.100508348089877
1	0.415773352896040	0.102269351231221
4	0.418688804684183	0.110151127920161

4.4.2 $m = 7$

Значение второй производной на правом конце	Максимальная абсолютная ошибка	Средняя абсолютная ошибка
-4	0.192108614149205	0.036943570113528
-1	0.192128237379461	0.035639303346036
0	0.192134778456213	0.035204547756872
1	0.192141319532965	0.035432225843398
4	0.192160942763222	0.036603914971144

4.4.3 $m = 14$

Значение второй производной на правом конце	Максимальная абсолютная ошибка	Средняя абсолютная ошибка
-10	0.024779422984575	0.002897460393552
-4	0.024779383092024	0.002633563157585
-1	0.024779363145780	0.002501614539660
0	0.024779356497023	0.002457631667002
1	0.024779349848261	0.002493051548017
4	0.024779329901997	0.002623025830546
10	0.024779290009419	0.002884202323742

4.4.4 Выводы

Разные значения второй производной практически никак не влияют на максимальную/среднюю абсолютною ошибку.

4.5 Сравнение двух предыдущих пунктов

Как мы уже поняли из предыдущих пунктов, выбор значения второй производной на конце практически не влияет на ошибку, в то время как выбор пропадающего узла наоборот оказывает очень сильное влияние. Поэтому можно считать, что оптимизация по этим двум параметрам более или менее сводится к оптимизации индекса пропадающего узла.

4.6 Сравнение естественного и специального сплайнов на равномерной сетке

4.6.1 $m = 4$

Сплайн	Визуальное сравнение	Максимальная абсолютная ошибка	Средняя абсолютная ошибка
естественный	конец полностью совпадает, начало графика повторяет форму, но довольно сильно сдвинуто в сторону	0.310576934890638	0.078863802225813
специальный ($ex = 1, y2b = 0$)	конец полностью совпадает, начало графика повторяет форму, но довольно сильно сдвинуто в сторону	0.414801535633325	0.100508348089877

4.6.2 $m = 7$

Сплайн	Визуальное сравнение	Максимальная абсолютная ошибка	Средняя абсолютная ошибка
естественный	конец полностью совпадает, середина и начало довольно близки к графику	0.137542053987132	0.019950315980158
специальный ($ex = 1, y2b = 0$)	конец полностью совпадает, середина довольно близка, в начале есть довольно сильно проседает	0.153353961118412	0.026476226031434

4.6.3 $m = 14$

Сплайн	Визуальное сравнение	Максимальная абсолютная ошибка	Средняя абсолютная ошибка
естественный	график почти полностью совпадает, в середине есть небольшое отклонение	0.024776636494779	0.002456577103526
специальный ($ex = 1, y2b = 0$)	график почти полностью совпадает, в середине есть небольшое отклонение	0.024779356497023	0.002457631667002

4.6.4 Выводы

В целом естественные сплайны дают меньшие ошибки. По форме графики довольно похожи при любых степенях, но естественные сплайны чуть меньше отличаются от графика (например, при $m = 7$ оба графика провисают в начале, но естественные чуть меньше). При увеличении степени эти отличия становятся почти незаметны (так как слайны почти одинаковые) и различий в ошибках почти нет.