Задание №5

Аппроксимация сплайнами и метод прогонки

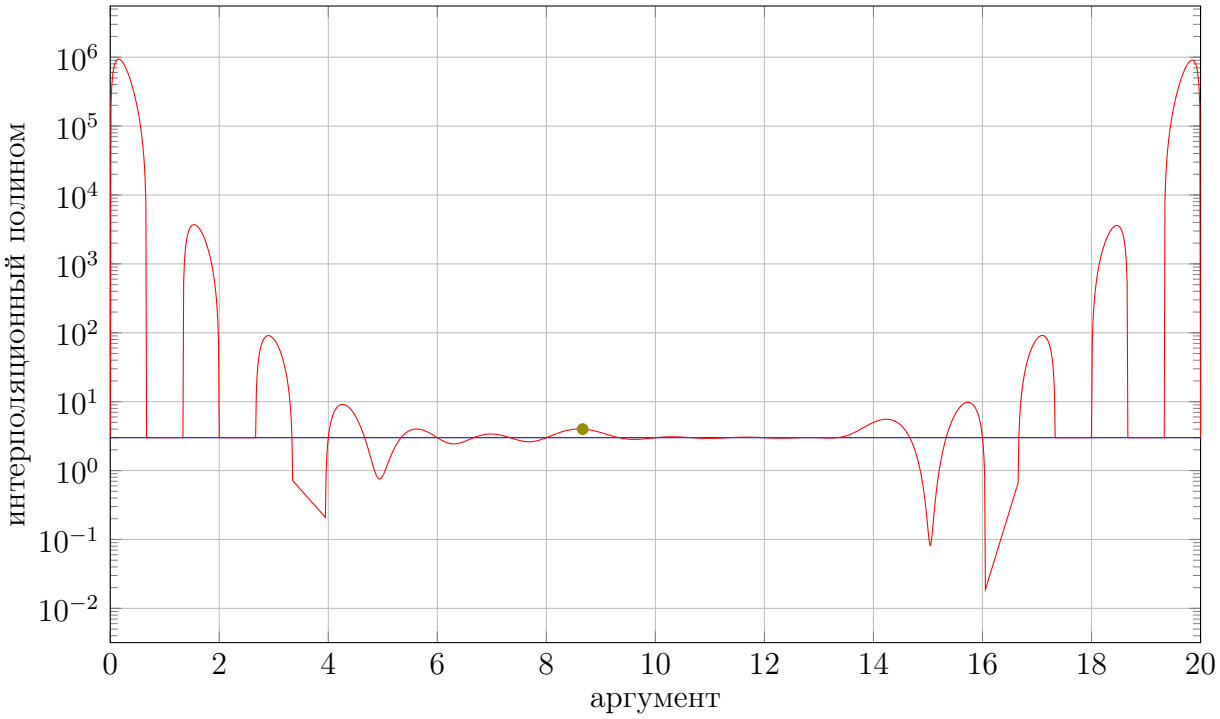
Постановка задачи

Интерполяция функций зачастую оказывается непригодной при решении задач, связанных с обработкой наблюдательных или экспериментальных данных. Причиной является наличие погрешностей в данных и сильнейшая чувствительность процедуры интерполяции к малому изменению условий.

Рассмотрим пример. Если провести интерполяцию на равномерной сетке (воспользовавшись результатами задания №2) с данными вида:

```
30
0.0000  20.000
3.0000
3.0000
......
3.0000
3.0000
```

то мы, естественно, получим в качестве результата горизонтальную прямую. Однако предположим, что одно (только одно!) значение в узлах из тридцати одного почему-то определено неправильно. Пусть в 14-м по счету узле случайно оказалось значение 4.0 (измерительный прибор дал сбой, вбивавший данные человек случайно нажал не ту кнопку и т.п.). Результат будет выглядеть как-то так:



Очевидно, что пользоваться таким результатом для оценки ожидаемых значений функции между узлами несколько неразумно. При этом использование чебышевской сетки не слишком улучшит ситуацию — отклонения от константы между узлами все равно останутся, хотя и станут меньше.

Для того, чтобы не сталкиваться с подобными проблемами, используется аппроксимация — построение функции заданного типа, в том или ином смысле наилучшим образом описывающей имеющиеся данные. Важно, что аппроксимирующая функция не обязательно проходит через заданные точки. Как правило, для нахождения такой функции используется метод наименьших квадратов — берется функция с одним или несколькими неизвестными параметрами, после чего параметры определяются из условия, что сумма квадратов расстояний между аппроксимируемыми точками и функцией должна быть минимальной.

Однако встречаются случаи, когда и тип аппроксимирующей функции неизвестен. Вся информация о нем сводится к тому, что соответствующая зависимость является достаточно гладкой. Тогда для аппроксимации используют т.н. «сплайны» — кусочно-заданные функции, совпадающие с какими-либо различными простыми функциями (чаще всего полиномами) на каждом отрезке области определения. Простейшим примером сплайна является обычная ломаная, построенная по точкам.

Но если мы хотим, чтобы получившаяся зависимость оказалась еще и гладкой, следует придумать что-то лучше, чем ломаная линия. Чаще всего для решения такой задачи используют т.н. «кубические» сплайны, на каждом отрезке области определения совпадающие с полиномами 3-й степени, коэффициенты которых подобраны таким образом, чтобы на краях отрезков сплайн оставался не только непрерывным, но и имел непрерывные первую и вторую производные.

Именно построением аппроксимирующего кубического сплайна мы и займемся. Заметим, что сплайны можно применять и для решения задачи интерполяции, но это сравнительно более простая задача.

Пусть даны вектора X_i, Y_i и P_i (абсциссы, ординаты и веса), где i меняется от 0 до n. Требуется построить кубический сплайн, аппроксимирующий точки (X_i, Y_i) с учетом весов P_i (чем больше вес, тем ближе к соответствующей точке пройдет сплайн). Будем строить так называемый «естественный» сплайн (с нулевыми вторыми производными на концах).

Описание метода

Строим трехдиагональную симметричную матрицу:

$$A = \begin{pmatrix} 2(X_1-X_0) & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 2(X_2-X_0) & X_2-X_1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & X_2-X_1 & 2(X_3-X_1) & X_3-X_2 & 0 & \dots & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & X_{n-2}-X_{n-3} & 2(X_{n-1}-X_{n-3}) & X_{n-1}-X_{n-2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & X_{n-1}-X_{n-2} & 2(X_n-X_{n-2}) & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 2(X_n-X_{n-1}) \end{pmatrix}.$$

И еще одну трехдиагональную и «почти симметричную»:

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \frac{1}{X_1-X_0} & -\frac{1}{X_1-X_0}-\frac{1}{X_2-X_1} & \frac{1}{X_2-X_1} & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & \frac{1}{X_2-X_1} & -\frac{1}{X_2-X_1}-\frac{1}{X_3-X_2} & \frac{1}{X_3-X_2} & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & \frac{1}{X_{n-1}-X_{n-2}} & -\frac{1}{X_{n-1}-X_{n-2}}-\frac{1}{X_n-X_{n-1}} & \frac{1}{X_n-X_{n-1}} \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix}.$$

Заметим, что при реализации метода обе матрицы целесообразно заполнять синхронно (они во многом похожи).

Также строим матрицу Q, у которой $Q_{ii} = \frac{1}{P_i}$, а все остальные элементы нулевые. Формируем и решаем относительно вектора S систему:

$$(A + 6 \cdot BQB^T)\, S = 6 \cdot BY$$

Затем формируем вектор результатов

$$R = Y - QB^T S$$

Собственно значения сплайна в точках вычисляются так. Для $\chi \in [X_i, X_{i+1})$ вводим

$$h = X_{i+1} - X_i,$$

$$t = \frac{\chi - X_i}{h}.$$

Тогда искомый результат — это функция

$$f(\chi) = R_i(1-t) + R_{i+1}t - h^2 \cdot \frac{t(1-t)}{6} \cdot \Big((2-t)\,S_i + (1+t)\,S_{i+1}\Big),$$

а ее производная (соответствующее выражение само по себе для выполнения задания не нужно, но может пригодиться для контроля правильности вычисления сплайна, а также полезно для общего развития):

$$f'(\chi) = \frac{R_{i+1} - R_i}{h} - \frac{h}{6} \cdot \Big((2 - 6t + 3t^2)\,S_i + (1 - 3t^2)\,S_{i+1}\Big).$$

Метод пятиточечной прогонки

Можно заметить, что матрица системы линейных алгебраических уравнений, которую необходимо решить, является пятидиагональной и симметричной. Это обстоятельство позволяет воспользоваться для ее решения методом намного более быстрым, чем различные модификации метода Гаусса. Метод этот называется «методом пятиточечной прогонки», и представляет собой частный случай метода прогонки, используемого при решении систем линейных алгебраических уравнений с ленточными матрицами (а также задач, сводящихся к решению таких систем — например, краевых задач для дифференциальных уравнений). Достаточным условием применимости метода является диагональное преобладание у матрицы системы (почти всегда получающееся в тех случаях, когда в приложениях возникают ленточные матрицы).

Пусть есть система вида

$$\begin{pmatrix}
a_1 & b_1 & c_1 & 0 & 0 & \dots & 0 & 0 & 0 \\
b_1 & a_2 & b_2 & c_2 & 0 & \dots & 0 & 0 & 0 \\
c_1 & b_2 & a_3 & b_3 & c_3 & \dots & 0 & 0 & 0 \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
0 & 0 & 0 & 0 & 0 & \dots & a_{n-2} & b_{n-2} & c_{n-2} \\
0 & 0 & 0 & 0 & 0 & \dots & b_{n-2} & a_{n-1} & b_{n-1} \\
0 & 0 & 0 & 0 & 0 & \dots & c_{n-2} & b_{n-1} & a_n
\end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ \dots \\ d_{n-2} \\ d_{n-1} \\ d_n \end{pmatrix}$$

Тогда такая система приводится к верхнему треугольному виду с помощью рекуррентных соотношений

$$\begin{cases}
\beta_i = b_{i-1} - p_{i-2}c_{i-2} \\
\alpha_i = a_i - p_{i-1}\beta_i - q_{i-2}c_{i-2} \\
p_i = \frac{b_i - q_{i-1}\beta_i}{\alpha_i} \\
q_i = \frac{c_i}{\alpha_i} \\
r_i = \frac{d_i - r_{i-1}\beta_i - r_{i-2}c_{i-2}}{\alpha_i}
\end{cases}$$

После этого решение ищется «обратным ходом»:

$$\begin{cases} x_i = r_i - p_i x_{i+1} - q_i x_{i+2}, & i = n, \ldots, 1 \end{cases}$$

Стартовые значения во всех случаях можно получить, считая, что все величины с индексами $i < 1$ или $i > n$ равны нулю.

Представление данных и результатов

Имеется файл *data.dat*, первая строка которого — символ ♯, пробел и значение n, а последующие строчки (n+1 строка) содержат тройки чисел X_i, Y_i, P_i (разделенных одним или несколькими пробелами). Можно считать (и проверять это не требуется), что точки уже отсортированы по возрастанию значений X_i, причем среди них нет повторяющихся. Требуется построить по этим данным кубический аппроксимационный сплайн и вывести в файл *result.dat* зависимость X, Y (в каждой строчке пара чисел «абсцисса–ордината», разделенная пробелами) на равномерной сетке с числом узлов, равным $q \cdot n + 1$, где q достаточно велико (например, $q = 100$).

Решение СЛАУ для построения сплайна должно использовать отдельную функцию (или процедуру), реализующую метод пятиточечной прогонки. Поскольку все используемые в задаче матрицы не более чем пятидиагональны, следует предусмотреть возможность не хранить многочисленные нули в далеких от главной диагонали элементах. Для операций с такими матрицами нужно воспользоваться кодом, написанным Вами при решении задания №1 (надеюсь, что этот код написан в таком виде, что Вы сможете им воспользоваться).

Обратите внимание, что в двух частях задания величина n используется в чуть разных смыслах, поэтому надо сделать так, чтобы реализации метода прогонки и построения сплайна не использовали n как глобальную переменную. Лучше всего сделать так, чтобы размеры матриц и т.п. не передавались в соответствующие процедуры/функции явно, а определялись по размеру поступивших аргументов-массивов, для чего можно использовать встроенные функции Фортрана SIZE, LBOUND и UBOUND.