Кусочек итогового отчета

Каглинская Мария, БПМИ141

2016

1 Модуль численного интегрирования

Теория
Был реализован модуль, осуществляющий численное вычисление интеграла с помощью формулы трапеций. Простая формула трапеций:

$$\int_a^b f(x)dx \approx \frac{(b-a)}{2}\left(f(a)+f(b)\right)$$

Составная формула трапеций:

$$\int_a^b f(x)dx \approx \frac{h}{2}\left(f_0 + 2(f_1 + f_2 + ... + f_{N-1}) + f_N\right)$$

Предполагаем, что сетка равнмерная с шагом $h = \frac{b-a}{N}$ Для данного метода можно аналитически оценить порядок точности как:

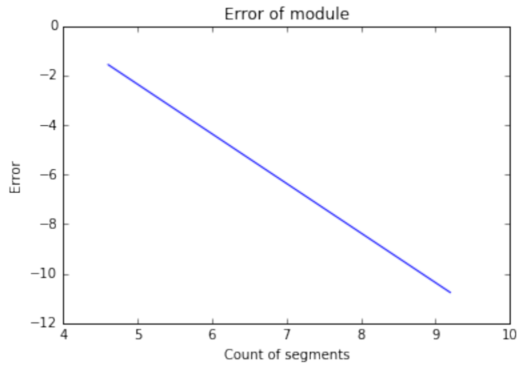
$$|Error| \leq \frac{b-a}{12} * \left\|f^{(2)}\right\|_{[a;b]}$$

Тестирование: Результаты сравнения метода с модулем интегрирования scipy.integrate.quad:

	x^2,	$100 * \frac{cos(x*3pi)}{x^2+1}$	$x = \begin{cases} 3 & \text{при x} < 1 \\ 1 & \text{Иначе} \end{cases}$
	$x \in [0,5], n = 50$	$x \in [-5,5], n = 50$	$x \in [-5,5], n = 50$
scipy.	41.666667	0.058502	22.000000
my module	41.675344	0.070983	22.040816
Difference	0.008677	0.012481	0.040816

Видим, что метод гораздо сильнее ошибается на осцилирующих и разрывных функциях.

Проверим порядок точности экспериментально, построив график зависимости погрешности от количества сегментов в сетке:



При использовании scipy.optimize.curve_fit получаем, что порядок аппроксимации k = 2.00, а оценка константы C = 7.64, то есть $Error \leq 7.64h^2$.

2 Модуль решения СЛАУ

Был реализован метод прогонки (так как он имет достаточно хорошее качество на трехдиагональных матрицах, а у нас как раз все матрицы будут такими). Этот метод также имеет достаточное условие диагонального преобладания.

Теория

Если матрица трехдиагональна, то система $M * x = F$ (обозначим элементы на ее значащих диагоналях A_i,B_i,C_i, где i-номер строки в матрице, на которой он расположен) имеет вид

$$A_i x_{i-1} + B_i x_i + C_i x_{i+1} = F_i$$

Метод основан также на предположении о том, что искомые неизвестные связаны рекуррентным соотношением: $x_i = x_{i+1}\alpha_{i+1} + \beta_{i+1}$, $i = n-1, ..1$. Можем тогда подставить это соотношение в уравнение, получим:

$$(A_i\alpha_i\alpha_{i+1} + C_i\alpha_{i+1} + B_i)\, x_{i+1} + A_i\alpha_i\beta_{i+1} + A_i\beta_i + C_i\beta_{i+1} - F_i = 0$$

Очевидно, что это соотношение будет выполняться вне зависимости от решения, если потребовать:

$$\begin{cases} A_i\alpha_i\alpha_{i+1} + C_i\alpha_{i+1} + B_i = 0 \\ A_i\alpha_i\beta_{i+1} + A_i\beta_i + C_i\beta_{i+1} - F_i = 0 \end{cases}$$

Можем выразить коэффициенты из этой системы:

$$\begin{cases} \alpha_{i+1} = \frac{-B_i}{A_i\alpha_i + C_i} \\ \beta_{i+1} = \frac{F_i - A_i\beta_i}{A_i\alpha_i + C_i} \end{cases}$$

Для первого уравнения получим:

$$\begin{cases} \alpha_2 = \frac{-B_1}{C_1} \\ \beta_2 = \frac{F_1}{C_1} \end{cases}$$

Когда все коэффициенты посчитаны, можем посчитать и сами решения с помощью рекуррентной формулы. Для получения n-ый элемент будем считать по формуле:

$$x_n = \frac{F_n - A_n\beta_n}{C_n + A_n\alpha_n} x_n = \frac{F_n - A_n\beta_n}{C_n + A_n\alpha_n}$$

Заметим, что из-за последовательного вычисления ошибка округления будет накапливается,то есть для матриц больших размерностей, она будет больше.

Тестирование

Возьмем для тестирования три матрицы, для всех из них посчитаем определитель, норму и число обусловленности. Далее с каждой из них решим по одной системе уравнений $M * x = F$ и для кажого решения посчитаем модуль невязки решения $E = |b - A\tilde{x}|$, где $\tilde{x}$– решение, полученное моим алгоритмом.

	$A = \begin{pmatrix} 1000 & 1 & 0 \\ 1 & 1000 & 1 \\ 0 & 1 & 1000 \end{pmatrix}$	$B_{8*8} = \begin{pmatrix} 1000 & 1 & ..0 \\ .. & .. & .. \\ 0.. & 1 & 1000 \end{pmatrix}$	$C_{8*8} = \begin{pmatrix} 1 & 1000 & ..0 \\ .. & .. & .. \\ 0.. & 1000 & 1 \end{pmatrix}$
norm	1732	2828	3741
cond	1	1	5
E	1.0	3.33e-16	8.65e-13

Известно, что метод прогонки работает лучше на выражениях с сильной главной диагональю (из-за достаточного условия). Это и видим на тестах: ошибка на матрице B меньше, чем на C. Также видим, что на матрице меньшего размера ошибка значительно меньше.

3 Модуль интерполяци

Интерполяция осуществляется с помощью кубических сплайнов дефекта 1.
Теория
Этот метод основан на том, что на кажом отрезке $[x_{i-1}, x_i]$ будем интерполировать нашу функцию полиномом степени не выше третьей, который: имеет непрерывные первую и вторую производные на всём отрезке, в точках x_i выполняется равенство $S(x_i) = f(x_i)$ (где S_i-сплайн). Коэффициенты этих полиномов и хотим найти.
На кажом отрезке разбиения $[x_{i-1}, x_i]$ сплайн задается таким соотношением:

$$S_i(x) = \frac{1}{6\delta_i}\left(m_i(x_{i+1} - x)^3 + m_{i+1}(x - x_i)^3\right) + A_i x + B_i$$

где m – вектор значений вторых производных. Значения m с 1 по n-1 можно найти, решив СЛАУ такого вида (также необходимы краевые условия на m_0, m_n, как правило их принимают равными 0): $A * x = F$, где

$$A = \begin{pmatrix} \frac{\delta_0+\delta_1}{3} & \frac{\delta_1}{6} & ... & ... & 0 \\ \frac{\delta_1}{6} & \frac{\delta_1+\delta_2}{3} & ... & ... & 0 \\ ... & ... & ... & ... & \frac{\delta_{i-2}}{6} \\ 0 & ... & ... & \frac{\delta_{i-2}}{6} & \frac{\delta_{i-2}+\delta_{i-1}}{3} \end{pmatrix}$$

$F = \begin{pmatrix} \frac{f_2-f_1}{\delta_1} - \frac{f_1-f_0}{\delta_0} \\ ... \\ \frac{f_{i+1}-f_i}{\delta_i} - \frac{f_i-f_{i-1}}{\delta_{i-1}} \end{pmatrix}$ Когда СЛАУ решена, коэффициенты A_i,B_i можно найти по формулам:

$A_i = \frac{f_{i+1}-f_i}{\delta_i} - \frac{\delta_i}{6}(m_{i+1} - m_i)$

$B_i = f_i - \frac{1}{6}m_i\delta^2 - A_i x_i$

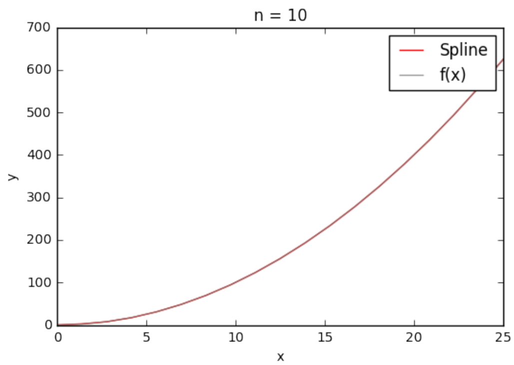
Таким образом находятся все коэффициенты всех сплайнов.

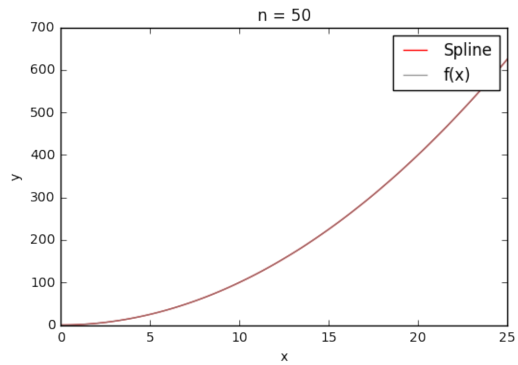
Тестирование
Возьмем три различные функции и для каждой два разных разбиения, одно мельче другого:

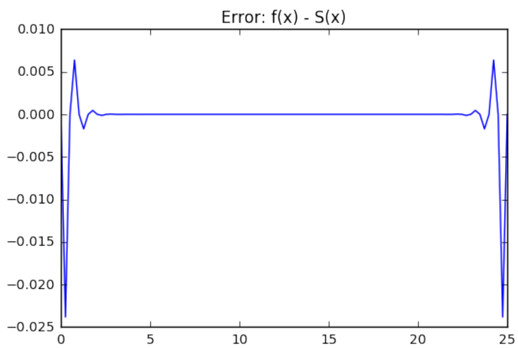
x^2,	$100 * \frac{cos(x*3pi)}{x^2+1}$	$x = \begin{cases} 3 & \text{при x} < 1 \\ 1 & \text{Иначе} \end{cases}$
$x \in [0, 5], n = 10, 50$	$x \in [-5, 5], n = 50, 100$	$x \in [-5, 5], n = 15, 50$

Для каждого теста нарисуем график функциии сплайна, построенного на двух сетках, и графики разности E(x) =f(x)S(x), гдеf– тестовая функция, а S– её интерполяция, построенная по более густой сетке из двух.

- x^2

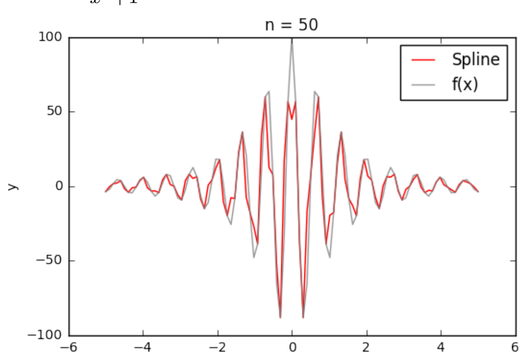


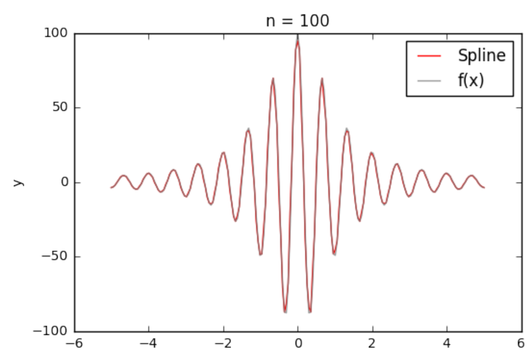


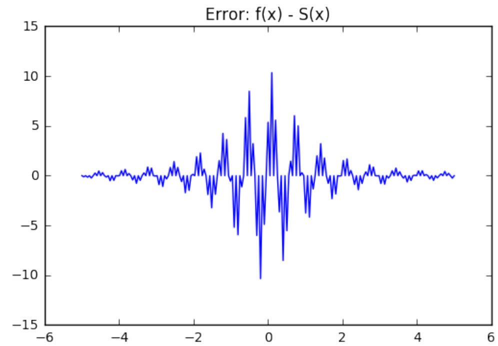


Видим, что ошибка равна 0 везде (так как квадратичную функцию приблизить многочленом третьей степени возможно), кроме краев отрезка, где краевые условия не дают сплайнам верно оценить нашу функцию.

- $100 * \frac{cos(x*3pi)}{x^2+1}$

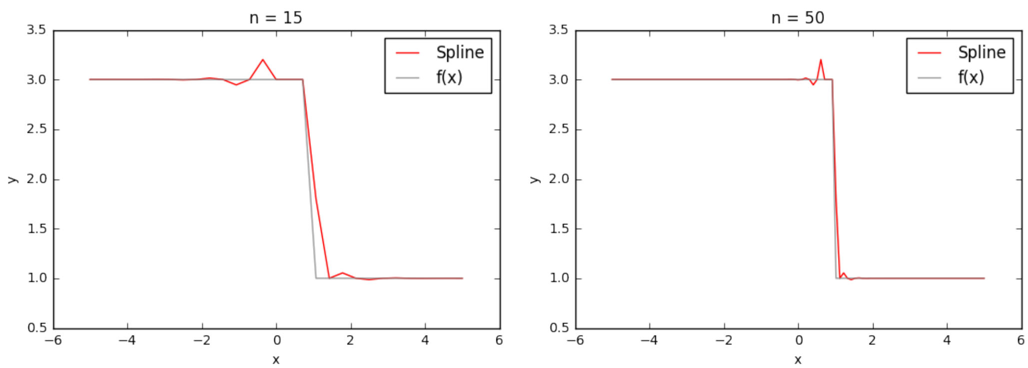


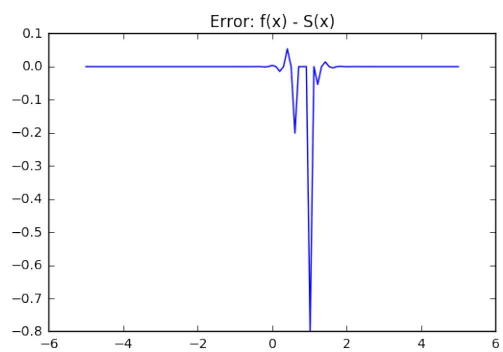




Для осцилирующей функции график ошибки также осцилирует около нуля и ошибка тем больше, чем больше колебания самой функции (что логично), то есть она может быть очень большой.

- $x = \begin{cases} 3 & \text{при x} < 1 \\ 1 & \text{Иначе} \end{cases}$





Для разрывной функции видим, что на непрерывных участках ошибка 0 (так как там функция- константа, то есть легко выражается многочленом третьей степени), а вот в точке разрыва даже на довольно большом количестве итераций ошибка присутствует.