

Задание №8

Гауссова квадратура (метод численного интегрирования Гаусса)

Постановка задачи

Среди методов численного интегрирования функций, основанных на интерполяционных формулах, выделяется т.н. «гауссова квадратура» — метод, позволяющий добиться максимально возможного порядка точности для некоторого числа узлов.

Рассмотрим интеграл и его приближенное значение, вычисленное на интерполяционной сетке с n узлами:

$$\int_{-1}^1 f(t) dt \approx \sum_{i=1}^n A_i f(t_i)$$

Тогда аргументы t_i определяются как корни полинома Лежандра соответствующей степени

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n,$$

а коэффициенты A_i являются решениями линейной системы n уравнений вида

$$\sum_{i=1}^n A_i t_i^k = \begin{cases} \frac{2}{k+1} & \text{при четном } k \\ 0 & \text{при нечетном } k \end{cases}$$

где k меняется от 0 до $n - 1$.

Для получения полиномов Лежандра можно воспользоваться рекуррентным соотношением

$$nP_n(x) + (n - 1)P_{n-2}(x) - (2n - 1)xP_{n-1}(x) = 0.$$

Метод Бернулли

Для вычисления корней полиномов Лежандра можно воспользоваться алгоритмом получения корней произвольных многочленов, известным под названием «метода Бернулли».

Пусть дано алгебраическое уравнение

$$a_0 x^n + \dots + a_n = 0.$$

Построим для него т.н. «конечно-разностное уравнение» вида

$$a_0 y_{n+i} + a_1 y_{n+i-1} + \dots + a_n y_i = 0,$$

которое рекуррентно задает последовательность $\{y_k\}$. Требуется задать n начальных значений от y_0 до y_{n-1} , остальные определяются из соотношения. Стартовые значения можно сгенерировать с помощью датчика случайных чисел (достаточно, чтобы хотя бы часть из них не равнялась нулю).

Тогда

$$\lim_{k \rightarrow \infty} \frac{y_{k+1}}{y_k} = x_1,$$

где x_1 — наибольший по модулю корень исходного алгебраического уравнения.

Заметим, что если отношение $\frac{y_{k+1}}{y_k}$ колеблется, не стремясь к пределу, то наибольший по модулю корень является комплексным (или максимальных по модулю корней более одного).

Задание

1. Написать функцию (процедуру), реализующую деление многочлена $P_n(x)$ на $x - x_0$ методом Горнера (чуть более сложный вариант — реализовать деление многочлена на произвольный многочлен с остатком), для хранения многочленов использовать массивы (или списки) их коэффициентов.
2. Написать функцию, решающую уравнение вида $P_n(x) = 0$ методом Бернулли и возвращающую массив корней. Эта функция, естественно, может (и должна) использовать предыдущую, желательно оформить их в виде отдельного модуля.
3. Написать функцию, вычисляющую коэффициенты полинома Лежандра для заданного n и возвращающую массив коэффициентов.
4. Написать функцию, вычисляющую корни полинома Лежандра для заданного n и возвращающую массив корней. При этом надо воспользоваться функцией, реализующей метод Бернулли (предварительно подумав, как приспособить ее к данной задаче).
5. Написать программу, вычисляющую коэффициенты A_i квадратурной формулы Гаусса и корни t_i полинома Лежандра для заданного n и выводящую их в файл с именем вида *quad<номер>.dat* (например, *quad05.dat* или *quad31.dat*) в виде таблицы, каждая строка которой имеет вид «коэффициент — соответствующий корень». При этом требуется воспользоваться каким-либо из методов решения СЛАУ из заданий №3 или №4. Количество узлов n должно передаваться в программу как аргумент командной строки (т.е. вызов должен иметь вид вроде

`./progr 20`

6. Написать функцию, вычисляющую интеграл некоторой функции $f(x)$ с помощью гауссовой квадратуры степени n на отрезке $[a; b]$, а также вызывающую ее программу, выводящую на консоль вычисленный результат. Функция (или программа) должна считывать коэффициенты и корни для гауссовой квадратуры из файла с соответствующим названием. n, a, b и имя программной реализации функции $f(x)$ (или указатель на нее) передаются функции интегрирования как параметры. Не забудьте, что при переходе от отрезка $[a; b]$ к $[-1; 1]$ функцию $f(x)$ необходимо отмасштабировать.