УДК 530-1

ББК 22.311

Д

Печатается по рекомендации Ученого Совета

Института физики

Казанского федерального университета

Рецензент: кандидат физико-математических наук, Архипов Р.В.

Дулов Е.Н.

Д **Введение в численные методы:** Учебно-методическое пособие для студентов Института физики / Е.Н. Дулов. – Казань: Издательство Казанского федерального университета, 2012. – 62 с.: 1 илл.

Пособие предназначено для студентов Института физики дневного и вечернего отделения, к практической части курса «Численные методы и математическое моделирование».

УДК 530-1

ББК 22.311

Оглавление

Введение

Численные методы в настоящее время широко используются в различных областях знаний, их практическую значимость трудно переоценить. Прямые применения включают в себя разнообразные программные инструменты. Сюда относятся специализированные математические пакеты. Общим для них является то, что ими можно пользоваться, не задаваясь вопросом «как это работает в деталях». С одной стороны, это, несомненно, экономия времени, с другой – всегда ограниченный функциональный набор. Также как правило, пользователь максимально изолирован от ключевых понятий численных методов, таких как погрешность метода, вычислительная погрешность, устойчивость, аппроксимация, сходимость. Между тем, незнание этих понятий может приводить к курьезным результатам, один из которых описан в настоящем пособии.

Практическое применение численных методов можно обнаружить вокруг себя почти в любом современном цифровом устройстве. Так, устройство, работающее со звуковыми или видеоданными, как правило, использует быстрое преобразование Фурье для сжатия информации. Методы обработки изображений охватывают множество разделов численных методов, от интерполяции и численных методов линейной алгебры до решения уравнений в частных производных. Цифровая обработка сигнала находит все большее применение в датчиках физических величин, позволяя достичь значительных преимуществ в качестве измерительных приборов.

Интересно отметить, что основы современных численных методов были заложены еще в 19 веке, задолго до появления электронных вычислительных машин (ЭВМ) и их широкого распространения. Быстрое преобразование Фурье (БПФ) и некоторые численные методы линейной алгебры также появилось до «компьютерной эры», в середине 20 века (хотя ЭВМ на момент появления БПФ уже существовали, но в виде громоздких машин и с возможностями, близкими к программируемым калькуляторам). Несмотря на огромное число достижений, численные методы по сей день остаются одной из наиболее динамично развивающихся наук.

Настоящее пособие не претендует на полноту охвата и строгость изложения. Например, отсутствует строгое определение сходимости, опущен значительный раздел об интерполяции функций. Изложенный далее материал нацелен на первое знакомство с ключевыми понятиями численных методов, с подходами к анализу погрешностей вычислений и с применением полученных знаний на практике.

Материал рассчитан на семестровый практический курс и предполагает знание только двух разделов высшей математики – математического анализа и линейной алгебры.

1. Источники погрешности при численном решении задачи

Можно выделить три источника погрешности, которые возникают при численном решении задачи:

а) неустранимая погрешность

б) погрешность метода

в) вычислительная погрешность

Неустранимая погрешность связана с конечной точностью исходных данных, что отчетливо проявляется в физических задачах. Так, решая баллистическую задачу, о полете пули вблизи поверхности Земли, мы можем указать физические константы с точностью, как правило, на много порядков ниже точности представления вещественных чисел в компьютере.

Погрешность метода характеризует конкретный численный метод, который мы применяем к решению задачи. Так, интерполяцию функции по табличным значениям можно сделать разными способами, от линейной до полиномиальной, с высокой степенью интерполяционного полинома. Какова при этом будет погрешность интерполяции - зависит как от выбранного метода, так и от свойств интерполируемой функции. В отношении любого метода приближенного вычисления можно сказать, что он *аппроксимирует* точное решение задачи, при этом точность *аппроксимации* определяется как аппроксимирующим выражением, так и свойствами аппроксимируемого решения. Приведем простой пример – приближенное вычисление функций с помощью рядов Тейлора. Так, функция e^x раскладывается в сходящийся степенной ряд в окрестности точки $x = 0$, и разложение будет давать хорошие результаты при вычислении e^x вблизи нуля. Используемое разложение в ряд с ограничением конечным числом слагаемых и будет представлять собой *аппроксимацию* исходной функции. Для функции $\sqrt{x}$ такой подход аппроксимации рядом Тейлора неприменим, поскольку ее производные, начиная с первой, обращаются в бесконечность в точке $x = 0$. Также нужно отметить, что погрешность метода зачастую связана с неустранимой погрешностью. Так, наличие сравнительно небольшой погрешности при задании функции в узлах интерполяции может привести к тому, что интерполяция с

использованием полиномов высоких степеней для некоторого класса функций окажется хуже линейной интерполяции.

Вычислительная погрешность определяется точностью представления вещественных чисел. При численном решении задачи на компьютере вещественные числа всегда задаются с конечным числом знаком после запятой, равно как и при вычислениях ручкой на бумаге. Связано это с тем, что под представление вещественного числа отводится конечное число байт, которое, в свою очередь, может иметь конечное число различимых комбинаций битов. Таким образом, бесконечное множество вещественных чисел подменяется дискретным множеством. В качестве примера рассмотрим представление числа с плавающей точкой в формате IEEE-754 single precision (международный стандарт, впервые принятый в 1985 году). Запись числа выполняется с помощью 4 байт, как показано на рисунке 1.

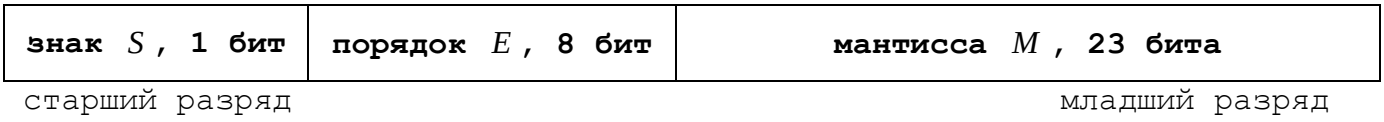

Рисунок 1. Представление вещественного числа в стандарте IEEE-754

Мантисса записывается в виде числа m в диапазоне $1 \le m < 2$, используется формат записи с фиксированной точкой. Целая часть мантиссы m всегда подразумевается равной 1, т.е. 23 бита M содержат только её дробную часть. Старший (левый на рис.1) бит имеет вес 1/2 , следующие 1/4, 1/8 и т.д. Таким образом, если представить себе число M как целое 23-битное число, значение мантиссы будет $m = 1 + M / 2^{23}$. Порядок величины E в двоичной системе счисления записан в виде целого числа, смещенного на 127. Бит знака выбирается равным 0 для положительных вещественных чисел и 1 для отрицательных. Таким образом, вещественное число R можно записать через целочисленные S, E, M:

$$R = (-1)^S \cdot 2^{E-127} \cdot (1 + M / 2^{23}) \qquad (1.1)$$

Выражение (1.1) охватывает диапазон вещественных чисел примерно от -3.4×10^{38} до 3.4×10^{38}. Внимательный читатель может заметить, что (1.1) должно описывать вдвое большие вещественные числа, т.е. динамический диапазон должен быть -3.4×10^{38} до 6.8×10^{38}. Дело в том, что числа с $E = 255$ рассматриваются как

специальные комбинации в IEEE-754, назначение которых – отслеживание выхода за динамический диапазон, т.е. обработка ошибок. Так, число с $M = 0$ и $E = 255$ служит для представления бесконечно большого числа и может быть использовано для оценки ошибок деления на ноль. Число с $E = 255$ и ненулевым M составляет комбинацию «Не Число» (NAN – Not a Number) и может быть использовано в тех случаях, когда результат арифметической операции не может быть определен однозначно, например, при выполнении операции умножения нуля на бесконечность. Также легко убедиться, что с помощью формулы (1.1) никогда не получить число, в точности равное нулю. Для операций с нулем используется правило: за ноль принимается наименьшее представимое (1.1) вещественное число, т.е. $E = 0$, $M = 0$. Количество специальных комбинаций, для которых вместо (1.1) используется особое соглашение, в IEEE-754 ограничено тремя вышеперечисленными – бесконечность, NAN и ноль.

Вернемся теперь к вопросу о вычислительной погрешности. Из (1.1) видно, что числа разного порядка будут иметь разную абсолютную точность представления. Так, если мы хотим записать число пи в формате IEEE-754, мы обнаружим, что сделать это можно только с точностью **7 знаков** после запятой. Значит, на этапе представления числа в формате с плавающей точкой, уже вносится погрешность, определяемая точностью формата.

Стандарт IEEE-754, с момента введения в 1985 году (иногда можно встретить обозначение этого первого варианта стандарта как IEEE-754-1985), дополняется возможностью работы с числами все большей точности. Последняя модификация IEEE-754-2008 включает в себя представление вещественного числа в формате quad precision с помощью 128 бит, 112 из которых отводятся под мантиссу. До 2008 года формат quad precision широко использовался в вычислениях и практически был стандартом де-факто.

Вычислительная погрешность может играть решающую роль в численных методах, в которых производится большое число однотипных арифметических операций.

Задание 1.1

Вычислить значение интеграла функции $f(x)=e^{-x}$ на отрезке [0,1] с помощью метода левых прямоугольников. Для этого разбить отрезок интегрирования на целое число N отрезков длиной $h=1/N$. В методе левых прямоугольников используется геометрический смысл интеграла как площади фигуры под функцией. *Аппроксимацией* в этом случае является приближенная замена каждого интеграла на отрезке h площадью прямоугольника с основанием h и высотой, равной значению функции на левой границе отрезка h. Сравнивая с точным решением $\int_0^1 e^{-x}dx = 1-e^{-1}$ определить погрешность ε метода левых прямоугольников как разницу между приближенным и точным значением. Результаты представить в виде таблицы:

h	ε
0.1	
0.01	
….	

Определить значение h, начиная с которого основной вклад в погрешность нахождения интеграла вносит вычислительная погрешность. Выполнить для двух различных значений точности представления вещественного числа.

2. Численное интегрирование

Численное интегрирование является тем разделом численных методов, на котором просто и ярко можно показать, что применение правильно выбранного метода вычислений может дать неожиданно большой выигрыш, как в точности, так и в скорости решения задачи. В этой главе будет введено понятие *порядка аппроксимации*, значимость которого также станет ясна после выполнения практических заданий.

Первое знакомство с численным интегрированием уже было в задаче предыдущей главы. Очевидный метод увеличения точности вычисления интеграла – уменьшение шага h – в итоге приводит к эффекту накопления вычислительной погрешности и ограничивает возможную точность вычислений. Применение более точных типов данных для вещественных чисел, конечно, позволит увеличивать точность численного интегрирования, но потребует увеличения времени вычислений. Причем, как было видно из задачи предыдущей главы, увеличение точности на один порядок потребует увеличения объема вычислений также на порядок. Интегрирование с числом отрезков $N>10^7$ занимает заметное время даже на современных персональных компьютерах.

Некоторые задачи требуют многократного вычисления интегралов. Так, задача математической обработки некоторого спектра с интегральным представлением одиночного спектрального контура, потребует 10^4-10^7 вычислений интегралов, в зависимости от числа точек спектра и выбранного метода обработки.

Прежде чем перейти к содержательной части главы, предлагается выполнить следующее задание.

Задание 2.1

Повторить Задание 1.1, заменив метод левых прямоугольников методом центральных прямоугольников. В методе центральных прямоугольников интеграл на каждом малом отрезке h заменятся площадью прямоугольника с высотой, равной значению функции на середине отрезка h. Таблицу из Задания 1 дополнить справа столбцом, в который следует записать точность нахождения интеграла методом центральных прямоугольников. Сравните результаты, полученные двумя методами.

Для того, чтобы объяснить различия в результатах метода центральных прямоугольников и метода левых прямоугольников, рассмотрим способ оценки погрешности, основанный на разложении в ряд Тейлора. Этот способ в дальнейших разделах курса встретится неоднократно.

2.1 Простейшие квадратурные формулы

Метод левых прямоугольников

Рассмотрим интеграл от функции $f(x)$ на отрезке $[x_0, x_0 + h]$:

$$I = \int_{x_0}^{x_0+h} f(x)dx \qquad (2.1.1)$$

Разложим $f(x)$ в ряд Тейлора в окрестности точки x_0, т.е. в левой границе отрезка h:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2} f''(x_0)(x - x_0)^2 + \ldots \qquad (2.1.2)$$

И подставим это разложение в (2.1.1):

$$I = \int_{x_0}^{x_0+h} \left[f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2} f''(x_0)(x - x_0)^2 + \ldots \right] dx \qquad (2.1.3)$$

Теперь любая функция, представленная в виде степенного ряда, может быть аналитически проинтегрирована, и мы получим результат также в виде степенного ряда:

$$I = f(x_0) \cdot h + f'(x_0)\frac{h^2}{2} + f''(x_0)\frac{h^3}{6} + \ldots \qquad (2.1.4)$$

Теперь заметим, что

$$I^* = f(x_0) \cdot h \qquad (2.1.5)$$

Представляет собой формулу метода левых прямоугольников. Тогда можно записать выражение, которое определяет, как соотносятся точное значение интеграла и приближенное, полученное методом левых прямоугольников:

$$I = f(x_0) \cdot h + f'(x_0)\frac{h^2}{2} + \ldots = I^* + O(h^2) \qquad (2.1.6)$$

Здесь $O(h^2)$ обозначает остаток разложения в степенной ряд. При малых h остаток $O(h^2)$ будет вести себя приближенно как величина, пропорциональная h^2.

Таким образом, определив погрешность вычислительного метода как:

$$\varepsilon = I^* - I \tag{2.1.7}$$

Можно утверждать, что погрешность будет приближенно уменьшаться вчетверо при уменьшении h вдвое, поскольку:

$$\varepsilon = I^* - I = -f'(x_0)\frac{h^2}{2} + ... = O(h^2) \tag{2.1.8}$$

Иначе говоря, формула (2.1.5) *аппроксимирует* точное значение интеграла (2.1.1) с порядком аппроксимации 2.

Из проделанных выкладок можно видеть, что конкретное положение x_0 отрезка h не влияет на полученный результат. Действительно, линейной заменой аргумента можно разместить отрезок h на оси x любым удобным нам образом, при этом вывод о втором порядке аппроксимации останется неизменным. Будем использовать это свойство в дальнейшем.

Формула (2.1.5) представляет собой простейшую квадратурную формулу, аппроксимирующую интеграл (2.1.1). Используя геометрический смысл интеграла, можно вообразить самые различные аппроксимации для (2.1.1). Например, метод трапеций или метод Симпсона. Все они будут иметь общий вид, который можно выразить следующим образом:

$$I^* = \sum_{k=1}^{n} A_k f(x_k) \tag{2.1.9}$$

Семейство формул (2.1.9) называется *квадратурными формулами*. В случае метода левых или центральных прямоугольников сумма в (2.1.9) состоит из одного единственного слагаемого.

При сравнении результата выполнения задания 1.1 с проделанным анализом погрешности обязательно должен возникнуть вопрос, почему на практике получается первый порядок погрешности аппроксимации для квадратуры левых прямоугольников, тогда как теоретический расчет дает второй порядок аппроксимации. Дело в том, что применение метода левых прямоугольников было сделано в виде *составных квадратур*, т.е. в виде суммы большого числа результатов работы одиночной формулы (2.1.5). Число применений одиночной квадратуры

(2.1.5) было $N = 1/h$ и на каждом применении вносилась погрешность $O(h^2)$. Порядок аппроксимации составной квадратуры, таким образом, будет $NO(h^2) = O(h^2)/h = O(h)$, всегда на один порядок ниже, чем у одиночной квадратуры.

Метод центральных прямоугольников

Рассмотрим теперь аппроксимацию интегралов квадратурой центральных прямоугольников. Разместим отрезок интегрирования h на оси x так, что $x = 0$ будет приходиться точно на середину отрезка. Повторяем вышеописанный прием:

$$I = \int_{-h/2}^{+h/2} f(x)dx = \int_{-h/2}^{+h/2} \left[f(0) + f'(0)x + \frac{1}{2} f''(0)x^2 + \ldots \right] dx = f(0)h + \frac{1}{24} f''(0)h^3 + \frac{1}{1920} f^{(4)}(0)h^5 \ldots \tag{2.1.10}$$

Таким образом, одиночная квадратура центральных прямоугольников имеет третий порядок аппроксимации:

$$I = I^* + O(h^3) \tag{2.1.11}$$

Составная квадратура центральных прямоугольников будет иметь порядок аппроксимации 2, на единицу меньше чем у одиночной квадратуры. Что и должно быть видно из выполнения Задания 2.1 – увеличение объема вычислений в 10 раз увеличивало точность вычислений в 100 раз.

Метод трапеций

Разместим отрезок h так же, как и при анализе в методе центральных прямоугольников. В методе трапеций интеграл аппроксимируется площадью прямоугольной трапеции:

$$I^* = \left(f(-h/2) + f(h/2) \right) \cdot \frac{h}{2} \tag{2.1.12}$$

Точное значение интеграла при этом представляется степенным рядом (2.1.10):

$$I = f(0) \cdot h + f''(0)\frac{h^3}{24} + f^{(4)}(0)\frac{h^5}{1920} \ldots \tag{2.1.13}$$

Нам нужно сравнить (2.1.13) и (2.1.12). Для этого, опять же с помощью ряда Тейлора, найдем значения функции $f(x)$ в точках $x = \pm h/2$:

$$f(\pm h/2) = f(0) \pm f'(0)\frac{h}{2} + f''(0)\frac{h^2}{8} \pm \ldots \tag{2.1.14}$$

Подставляя (2.1.14) в (2.1.12), получим, что (2.1.12) также может быть представлено степенным рядом:

$$I^* = \left(2f(0) + f''(0)\frac{h^2}{4} + f^{(4)}(0)\frac{h^4}{192} + \ldots \right) \cdot \frac{h}{2} = f(0) \cdot h + f''(0)\frac{h^3}{8} + f^{(4)}(0)\frac{h^5}{384} + \ldots \qquad (2.1.15)$$

Найдем теперь погрешность метода трапеций, сравнивая (2.1.13) и (2.1.15):

$$\varepsilon = I^* - I = f''(0)\frac{h^3}{12} + f^{(4)}(0)\frac{h^5}{480} \ldots = O(h^3) \qquad (2.1.16)$$

Интересно отметить, что погрешности в методе трапеций и в методе центральных прямоугольников имеют схожий вид, но разный знак главного члена в соответствующем степенном ряде. При малых h погрешность метода трапеций будет с хорошей точностью по модулю вдвое больше погрешности метода центральных прямоугольников. Это соотношение погрешностей иногда используется для оценки значения интеграла сверху и снизу. Т.е. одновременное применение двух методов даст нам интервал, внутри которого точно находится истинное значение интеграла.

Метод Симпсона

Идею метода трапеций можно представить следующим образом. Функция на отрезке h интерполируется прямой, т.е. сама функция этой прямой линией аппроксимируется. Далее берется интеграл от аппроксимирующей функции, и мы получаем квадратуру трапеций (2.1.12). Этот подход можно развить, используя аппроксимацию функцией более высокого порядка, например, параболой. Именно так получается метод Симпсона. Через три соседних точки проводится интерполяционный полином второй степени, и интеграл берется уже от него. В итоге получается квадратурная формула, в справедливости которой нетрудно убедиться самостоятельно:

$$I^* = \frac{f(-h/2) + 4f(0) + f(h/2)}{6} \cdot h \qquad (2.1.17)$$

Также квадратура Симпсона может быть получена в виде взвешенной суммы квадратур трапеций и центральных прямоугольников. Обозначим вес квадратуры трапеций A. Тогда вес квадратуры центральных прямоугольников будет $1-A$, в

противном случае не будет выполняться условие *сходимости* полученной суммы к точному решению при $h \to 0$.

$$I^* = A\frac{f(-h/2)+f(h/2)}{2}\cdot h + (1-A)f(0)h \qquad (2.1.18)$$

Вспомним, что погрешность квадратуры трапеций примерно вдвое больше по модулю, чем в методе центральных прямоугольников и всегда имеет противоположный знак. Тогда, задавшись целью эту ошибку скомпенсировать (что значит исчезновение еще одного слагаемого в степенном ряде), нужно чтобы

$$2A = 1 - A \qquad (2.1.19)$$

Откуда получаем $A = 1/3$. Подставляя в (2.1.18) получаем квадратуру Симпсона.

Проанализируем подробно порядок аппроксимации этой квадратурной формулы.

Для метода трапеций, из (2.1.16):

$$I^* = I + f''(0)\frac{h^3}{12} + f^{(4)}(0)\frac{h^5}{480}... \qquad (2.1.20)$$

Для метода центральных прямоугольников, из (2.1.10):

$$I^* = I - \frac{1}{24}f''(0)h^3 - \frac{1}{1920}f^{(4)}(0)h^5... \qquad (2.1.21)$$

Подставляя степенные ряды для квадратур трапеций и прямоугольников в (2.1.18):

$$I^* = \frac{1}{3}\left[I + f''(0)\frac{h^3}{12} + f^{(4)}(0)\frac{h^5}{480}...\right] + \frac{2}{3}\left[I - \frac{1}{24}f''(0)h^3 - \frac{1}{1920}f^{(4)}(0)h^5...\right] \qquad (2.1.22)$$

Получаем:

$$I^* = I + f^{(4)}(0)\frac{h^5}{2880} + ...$$

$$\varepsilon = I^* - I = f^{(4)}(0)\frac{h^5}{2880} + ... = O(h^5) \qquad (2.1.22)$$

Задание 2.1.1

Дополнить решения Заданий 1.1 и 2.1 методами трапеций и Симпсона. Сравнить объем вычислений при использовании метода левых прямоугольников и при использовании метода Симпсона. Требуемая погрешность конечного результата 10^{-7}.

Еще один способ определения термина «порядок аппроксимации» состоит в следующем. Величина погрешности различных квадратур определяется производными различных порядков. В квадратуре Симпсона, например, это производные 4-го и более высокого порядка. Это значит, что применение квадратуры Симпсона к полиномам степени не выше 3 должно давать точный результат независимо от числа отрезков разбиения N в составной квадратуре. Значит, порядок аппроксимации можно определить максимальной степенью полинома, для которого аппроксимация точна.

Задание 2.1.2

Требуется найти на отрезке [0,1] значение интеграла от функции $f(x)=1+x+x^2+x^3$. Найти точное значение интеграла. Выполнить численное интегрирование с помощью методов трапеций и Симпсона. Записать погрешность вычислений в виде таблицы, такой же, как в предыдущих заданиях.

2.2 Квадратуры Гаусса

Все рассмотренные выше квадратурные формулы были построены с помощью геометрических аналогий (только для квадратуры Симпсона был предложен способ взвешенной суммы, но сама квадратура может быть получена и без него). При этом порядок аппроксимации получался «какой есть», то есть сам подход геометрических аналогий не содержит каких-либо начальных ограничений на порядок аппроксимации. Попробуем решить задачу в обратном порядке. Зададимся требованием получить максимально высокий порядок аппроксимации при заданном числе узлов квадратуры n. Сначала разберем случай $n=2$. Квадратуру для отрезка $[-h/2,+h/2]$ можно записать в виде, напоминающем формулу трапеций:

$$I^*=\sum_{k=1}^{2}A_k f(x_k)=A_1 f(x_1)+A_2 f(x_2)=A_1 f\left(-\alpha_1\frac{h}{2}\right)+A_2 f\left(+\alpha_2\frac{h}{2}\right) \qquad (2.2.1)$$

В отличие от формулы трапеций положение узлов здесь не зафиксировано, оно будет найдено из условия максимальной точности квадратуры.

Разложение в степенной ряд для точного значения интеграла (2.1.10) является нечетной функцией по h. Из (2.2.1) видно, что нечетную функцию со степенным рядом вида (2.1.10) можно получить только в случае

$$\alpha_1 = \alpha_2 = \alpha\,,\ A_1 = A_2 = \beta h \tag{2.2.2}$$

где α и β - некоторые постоянные

Условия (2.2.2) означают симметричность весов и расположения узлов квадратуры относительно точки $x = 0$. Такой симметрией будут обладать квадратуры Гаусса для любого числа узлов. Квадратурная формула (2.2.1) приобретает вид:

$$I^* = \beta h\left(f\left(-\alpha\frac{h}{2}\right) + f\left(+\alpha\frac{h}{2}\right)\right) \tag{2.2.3}$$

Запишем степенной ряд:

$$f(\pm\alpha h/2) = f(0) \pm f'(0)\frac{\alpha h}{2} + f''(0)\frac{(\alpha h)^2}{8} \pm f^{(3)}(0)\frac{(\alpha h)^3}{48} + f^{(4)}(0)\frac{(\alpha h)^4}{384} \pm \ldots \tag{2.2.4}$$

Тогда

$$I^* = \beta h\left(2f(0) + f''(0)\frac{(\alpha h)^2}{4} + f^{(4)}(0)\frac{(\alpha h)^4}{192} + \ldots\right) \tag{2.2.5}$$

Повторим здесь разложение в степенной ряд (2.1.10) для точного значения интеграла:

$$\begin{aligned} I &= \int_{-h/2}^{+h/2} f(x)dx = \int_{-h/2}^{+h/2}\left[f(0) + f'(0)x + \frac{1}{2}f''(0)x^2 + \frac{1}{6}f'''(0)x^3 + \frac{1}{24}f^{(4)}(0)x^4 \ldots\right]dx = \\ &= f(0)h + \frac{1}{24}f''(0)h^3 + \frac{1}{1920}f^{(4)}(0)h^5 + \ldots \end{aligned} \tag{2.2.6}$$

Сравнивая (2.2.5) и (2.2.6) видим, что выбором α и β можно обеспечить совпадение степенных рядов точного значения интеграла и квадратурной формулы до кубического слагаемого. При этом

$$\begin{cases} 2\beta = 1 \\ \dfrac{\beta}{4}\alpha^2 = \dfrac{1}{24} \end{cases} \tag{2.2.7}$$

Откуда получаем решение задачи об оптимальном выборе узлов в двухузловой квадратуре:

$$\beta = \frac{1}{2}\,,\ \alpha = \frac{1}{\sqrt{3}} \tag{2.2.8}$$

Степенной ряд для квадратуры приобретает вид:

$$I^* = f(0)h + f''(0)\frac{h^3}{24} + f^{(4)}(0)\frac{h^5}{3456} + \ldots \qquad (2.2.9)$$

Погрешность аппроксимации:

$$\varepsilon = I^* - I = f^{(4)}(0)\frac{h^5}{384}\left(\frac{1}{9} - \frac{1}{5}\right) + \ldots = -f^{(4)}(0)\frac{h^5}{864} + \ldots = O(h^5) \qquad (2.2.10)$$

Это и есть двухузловая квадратура Гаусса. Запишем ее окончательный вид и сравним с квадратурой трапеций (2.1.12).

$$I^* = \frac{h}{2}\left(f\left(-\frac{h}{2\sqrt{3}}\right) + f\left(+\frac{h}{2\sqrt{3}}\right)\right) \qquad (2.2.11)$$

Видно, что отличие заключается лишь в коэффициенте $1/\sqrt{3}$ в аргументах функции. Точки, на которых строится «трапеция» в 2-узловой квадратуре Гаусса оказываются сдвинуты к точке $x = 0$. Эта формула, если её использовать без знания о том, как она была получена, обычно вызывает интуитивное неприятие из-за отсутствия простых геометрических аналогов. Действительно, почему «неправильная» трапеция будет давать более точные результаты, в отличие от правильной? Однако это действительно так, в чем предлагается удостовериться в следующем задании:

Задание 2.2.1

Дополнить решения Заданий 1.1, 2.1, 2.1.1 методом 2-узловой квадратуры Гаусса. Найти в численном эксперименте порядок аппроксимации 2-узловой квадратуры и сравнить его с теоретическим. Проверить соотношение погрешностей в методах 2-узловой квадратуры Гаусса и в методе Симпсона.

Итак, 2-узловая квадратура имеет порядок аппроксимации одиночной квадратуры $O(h^5)$, такой же, как и квадратура Симпсона. Сравнивая выражения для степенных рядов погрешности в этих квадратурах, можно видеть, что для малых h знаки погрешности будут разными, а соотноситься они будут примерно как 3 к 10 (абсолютная точность квадратуры Симпсона выше).

В ходе вывода 2-узловой квадратуры Гаусса использовались два варьируемых параметра, при этом полученная квадратура дает совпадение первых двух слагаемых в степенном ряде с аналогичным рядом для точного значения интеграла. Попробуем предсказать, что получится в решении задачи об оптимальном построении 3-узловой

квадратуры. Расположение узлов также должно быть симметричным относительно центра отрезка, значит один узел должен приходиться на точку $x=0$. Тогда 3-узловая квадратура будет отличаться от 2-узловой наличием только одного дополнительного параметра – веса при узле $x=0$. Это значит, что можно будет добиться совпадения еще одного слагаемого в степенных рядах точного и приближенного значения интеграла. Подобные рассуждения можно продолжить и убедиться, что добавление одного узла приводит к «уточнению» степенного ряда приближенного выражения интеграла на одно слагаемое. В итоге N-узловая квадратура Гаусса должна иметь порядок аппроксимации $2N+1$.

Приведем без вывода выражение для 3-узловой квадратуры Гаусса на отрезке $[-h/2, h/2]$:

$$I^* = \frac{5f\left(-(h/2)\sqrt{3/5}\right)+8f(0)+5f\left((h/2)\sqrt{3/5}\right)}{18}\cdot h = I + O\left(h^7\right) \qquad (2.2.12)$$

Выглядит она похоже на квадратуру Симпсона, однако имеет порядок точности выше на 2, и, как и 2-узловая квадратура Гаусса не имеет простых геометрических аналогов.

Построение N-узловой квадратуры Гаусса сводится к решению системы нелинейных уравнений и для числа узлов не более 4 выполняется сравнительно просто аналитически. Чтобы построить квадратуры Гаусса с числом узлов 5 и более, обычно используют численные методы решения систем нелинейных уравнений или задач оптимизации – один из важных разделов численных методов, который будет подробно рассматриваться ближе к концу курса.

Интересно отметить, что задача об оптимальном выборе узлов и весов квадратур была решена Гауссом в середине 19 века. Квадратура центральных прямоугольников представляет собой 1-узловую квадратуру Гаусса.

Задание 2.2.2

Дополнить решения Заданий 1.1, 2.1, 2.1.1 и 2.2.1 методом 3-узловой квадратуры Гаусса. Найти в численном эксперименте порядок аппроксимации 3-узловой квадратуры и сравнить его с теоретическим.

Дополнительное задание

Построить аналитические выражения для 3- и 4-узловой квадратуры Гаусса.

Выполненные задания ярко демонстрируют выигрыш от применения правильно выбранного численного метода. Время, необходимое для вычисления интеграла с заданной точностью, по мере выполнения заданий было сокращено более чем в миллион раз.

2.3 Метод Монте-Карло

Интегрирование с помощью квадратур удобно, когда подынтегральная функция имеет сходящийся степенной ряд на отрезке интегрирования, и когда интеграл берется по небольшому числу переменных. Если эти условия не выполняются, достижение требуемой точности интегрирования может потребовать большого объема вычислений для любых квадратур.

Допустим, интегрирование некоторой одномерной задачи требует использования составной квадратуры с числом отрезков разбиения 10. Эта задача может быть легко решена вручную на бумаге или на калькуляторе. Теперь представим себе аналогичную 10-мерную задачу. Вложенный интеграл потребует применения 10^{10} квадратур, что окажется трудной задачей даже для современных компьютеров. При этом обязательно возникнет проблема накопления вычислительной погрешности. Причем скорость накопления вычислительной погрешности может превышать скорость сходимости неудачно выбранной квадратуры.

Можно также привести примеры «трудноинтегрируемых» функций для метода квадратур. Это функции, не имеющие разложения в степенной ряд или имеющие малый радиус сходимости в некоторых точках отрезка интегрирования. К числу таких функций, при некоторых отрезках интегрирования, относятся $\sqrt{x}$, $1/x$, $tg(x)$, e^{x^2} и т.д.

Для описанных случаев выгодным может быть использование метода Монте-Карло. Этот метод не накладывает никаких требований на сходимость и существование степенного ряда в области интегрирования.

Рассмотрим применение метода Монте-Карло в простейшем случае. Допустим, нам известен диапазон значений функции $[f_{\min}, f_{\max}]$ на отрезке интегрирования $[x_{\min}, x_{\max}]$. Тогда можно использовать геометрический смысл интеграла как площади фигуры под функцией. Бросая случайным образом точку на прямоугольник $[f_{\min}, f_{\max}]$, $[x_{\min}, x_{\max}]$ будем подсчитывать число событий n, когда точка оказывается под функцией. Общее число бросаний обозначим N. Тогда вероятность попадания точки в фигуру под функцией будет найдена в численном эксперименте:

$$P = \frac{n}{N} \tag{2.3.1}$$

С другой стороны, эту вероятность мы можем найти аналитически:

$$P = \frac{1}{(x_{\max} - x_{\min})(f_{\max} - f_{\min})} \int_{x_{\min}}^{x_{\max}} f(x)dx \tag{2.3.2}$$

Понятно, что сделав, например, 1000 одинаковых бросаний, в разных случаях мы получим разное число n. Распределение случайной величины n будет биномиальным, с математическим ожиданием, которое определяется сравнением выражений (2.3.1) и (2.3.2). При большом числе бросаний биномиальное распределение n будет стремиться к пуассоновскому, с дисперсией n. Тогда стандартное отклонение $\sqrt{n}$ будет определять ошибку метода вычислений в методе Монте-Карло. Интеграл будет равен

$$\int_{x_{\min}}^{x_{\max}} f(x)dx = \lim_{n\to\infty}(x_{\max} - x_{\min})(f_{\max} - f_{\min})\frac{n}{N} \tag{2.3.3}$$

Относительная ошибка интегрирования составит $1/\sqrt{n}$. Это значит, что для достижения абсолютной точности 10^{-3} интегрирования функции e^{-x} из Задания 1 нам потребуется сделать несколько миллионов бросаний. Для увеличения точности интегрирования в 10 раз потребуется увеличение числа бросаний в 100 раз. В этом смысле метод Монте-Карло имеет порядок аппроксимации $O(h^{1/2})$. Это хуже, чем у простейшей квадратуры левых прямоугольников.

На первый взгляд, метод Монте-Карло дает гораздо более скромные результаты, по сравнению с методом квадратур. Однако в применении к многомерным задачам выигрыш может быть существенным, при удачном выборе области, в которую происходят бросания. Действительно, при фиксированном числе

бросаний относительную точность одного порядка мы получим и для одномерного и для десятимерного случая, т.е. размерность задачи слабо будет влиять на скорость ее решения.

Задание 2.3.1

Дополнить решения Заданий 1.1, 2.1, 2.1.1, 2.2.1 методом Монте-Карло. Найти в численном эксперименте порядок аппроксимации и сравнить его с теоретическим.

Задание 2.3.2

Вычислить значение интеграла функции $f(x) = \sqrt{x}$ на отрезке [0,1] с помощью составных квадратур центральных прямоугольников и с помощью метода Монте-Карло. Составить таблицу зависимости погрешности методов от объема вычислений, аналогичную таблице из Задания 1.1.

3. Численное дифференцирование

Численное нахождение производных, на первый взгляд, представляет собой тривиальную задачу. Используя определение первой производной, можно записать:

$$f'(x)=\lim_{x\to\infty}\frac{f(x+\Delta x)-f(x)}{\Delta x}\approx\frac{f(x+h)-f(x)}{h} \qquad (2.3.4)$$

И уменьшать h до тех пор, пока не будет достигнута необходимая точность. Такой «очевидный» подход таит в себе опасности. Для нахождения таким способом первой производной нам придется делить одну малую величину на другую. Обозначим точность представления числителя, как ε. Она практически всегда будет конечной, из-за вычислительной погрешности. Тогда погрешность вычисления первой производной, связанная с конечной точностью представления числителя будет:

$$f'(x)\approx\frac{f(x+h)-f(x)+\varepsilon}{h}=\frac{f(x+h)-f(x)}{h}+\frac{\varepsilon}{h} \qquad (2.3.5)$$

Таким образом, уменьшение h с целью увеличения точности результата, встретит на своем пути препятствие в виде вклада в погрешность ε/h. Допустим, у нас есть возможность задавать вещественные числа с точностью около 14 знаков после запятой и мы хотим взять первую производную от функции вида $f=e^x$ в точке $x=0$. При $h\approx10^{-7}$ вклад в погрешность вычисления первой производной уже будет 10^{-7}. Дальнейшее уменьшение h даст только ухудшение результата.

Задание 3.1

С помощью формулы (2.3.5) найти численно значения первых производных от функции $f=e^{-x}$ для различных значений h и для различной точности представления вещественных чисел. Определить погрешность численного решения, сравнивая с точным значением производной. Результаты представить в виде таблиц:

h	*e*
0.1	
0.01	
….	

Мы проанализировали только вклад вычислительной погрешности в численном нахождении производной. Кроме этого вклада будет также и конечная точность аппроксимации производной с помощью (2.3.5), определяемая малостью h. Выражение (2.3.5) представляет собой пример *конечной разности*, этот термин встретится в дальнейшем при численном решении дифференциальных уравнений.

Рассмотрим точность аппроксимации. Обозначим приближенное значение производной:

$$f'^{*}(x)=\frac{f(x+h)-f(x)}{h} \qquad (2.3.6)$$

И определим ошибку вычислений:

$$\varepsilon = f'^{*}(x)-f'(x) \qquad (2.3.7)$$

Представим $f(x+h)$ через разложение в ряд Тейлора в окрестности точки x. Тогда:

$$f'^{*}(x)=\frac{\left[f(x)+f'(x)h+\frac{1}{2}f''(x)h^2+\frac{1}{6}f'''(x)h^3+...\right]-f(x)}{h}=f'(x)+\frac{1}{2}f''(x)h+\frac{1}{6}f'''(x)h^2+...$$

$$f'^{*}(x)=f'(x)+O(h) \qquad (2.3.8)$$

То есть простейшая конечная разность (2.3.6) представляет собой аппроксимацию производной первого порядка точности по h. Обратимся к вышеописанному примеру с численным нахождением производной от $f=e^x$. С одной стороны, нам необходимо уменьшать h, чтобы повысить точность аппроксимации, с другой – нельзя делать h чересчур малым из-за вычислительной погрешности. Видно, что сочетание этих требований автоматически «загрубляет» вычисления, в которых присутствует конечная разность (2.3.6) для первой производной. При этом эффект получается таким, словно бы «потерялось» около половины значащих знаков.

Увеличить точность вычислений производных может помочь более точная аппроксимация. Рассмотрим конечно-разностное выражение, похожее на (2.3.6), но симметризованное относительно точки x:

$$f'^{*}(x)=\frac{f(x+h/2)-f(x-h/2)}{h} \qquad (2.3.9)$$

Заменим $f(x \pm h/2)$ через степенные ряды Тейлора в окрестности точки x:

$$f'^{*}(x) = \frac{\left[f(x) + f'(x)\frac{h}{2} + f''(x)\frac{h^2}{8} + f'''(x)\frac{h^3}{48} + \ldots \right] - \left[f(x) - f'(x)\frac{h}{2} + f''(x)\frac{h^2}{8} - f'''(x)\frac{h^3}{48} + \ldots \right]}{h}$$

$$f'^{*}(x) = f'(x) + f'''(x)\frac{h^2}{24} + \ldots = f'(x) + O(h^2) \qquad (2.3.10)$$

Точность аппроксимации здесь на один порядок лучше.

Задание 3.2

Дополнить решение задания 3.1 симметризованной конечной разностью (2.3.9). Определить предельную точность нахождения первой производной в этом методе, по сравнению с конечной разностью (2.3.6).

Из проделанных выкладок ясно, что для вычисления первой производной мы можем построить нечто похожее на квадратуры Гаусса в задачах численного интегрирования. Так, взяв две конечноразностные формулы вида (2.3.9), мы можем их скомбинировать с разными весами с целью исключения квадратичного слагаемого в степенном ряде для конечно-разностной формулы.

Задание 3.3

Из условия максимального порядка аппроксимации определить веса A, B в формуле приближенного вычисления первой производной

$$f'^{*}(x) = A\frac{f(x + h/2) - f(x - h/2)}{h} + B\frac{f(x + h) - f(x - h)}{2h}$$

Используя полученный результат, дополнить решение заданий 3.1, 3.2. Определить порядок аппроксимации.

Конечно-разностные выражения для производных более высоких порядков могут быть построены различным образом. Например, можно использовать геометрические аналогии. Через любые три точки всегда можно провести интерполирующую параболу, причем единственным образом. При этом, поскольку парабола является приближением к функции, следует ожидать, что и вторая производная от нее в центральной точке будет приближенно равна второй производной от параболы. Другой возможный способ состоит в двукратном

применении (2.3.9). Сначала с помощью (2.3.9) находятся производные в точках $x-h/2$ и $x+h/2$. Затем, полученная первая производная рассматривается как самостоятельная функция, и (2.3.9) применяется еще раз. Оба способа приведут к выражению:

$$f''^{*}(x)=\frac{f(x+h)-2f(x)+f(x-h)}{h^2}+O(h^2) \qquad (2.3.11)$$

Ситуация с нарастающим вкладом вычислительной погрешности при уменьшении h здесь будет драматичнее, поскольку деление на h происходит дважды. Максимальная достижимая точность результата будет характеризоваться лишь третьей частью значащих знаков от количества знаков в представлении вещественного числа.

Задание 3.4

Из условия максимального порядка аппроксимации определить веса A, B, C в формуле приближенного вычисления второй производной

$$f''^{*}(x)=\frac{Af(x+2h)+Bf(x+h)+Cf(x)+Bf(x-h)+Af(x-2h)}{h^2}$$

Определить порядок аппроксимации полученной формулы в численном эксперименте и сравнить его с полученным теоретическим выражением.

4. Практическое правило оценки погрешности Рунге и адаптивные вычисления

Предыдущие задания по численному интегрированию и дифференцированию функций включали в себя нахождение погрешности вычислений как разницы между известным точным и полученным численным решениями. На практике такой случай может встретиться, например, если разрабатывается новый, более быстрый алгоритм для некоторой задачи, точные решения которой уже известны. Но чаще возникают задачи с неизвестным заранее ответом. При этом возникает необходимость оценки погрешности полученного численного результата.

Наличие разложения для погрешности в степенной ряд и знакомство с понятием *порядка аппроксимации* позволяет просто понять правило Рунге для оценки погрешности. Рассмотрим пример вычисления интеграла с помощью составных квадратур центральных прямоугольников, имеющих второй порядок аппроксимации интеграла:

$$I = I^* + O\left(h^2\right) \tag{4.1}$$

Значит, погрешность интегрирования можно представить как:

$$\varepsilon = I^* - I \approx kh^2 \tag{4.2}$$

где k - некоторое число, не зависящее от h.

Получив два численных значения интеграла для длин отрезков h и $h/2$, обозначим их как I_h^* и $I_{h/2}^*$, соответственно. Тогда разность этих численных значений позволит нам найти k:

$$I_h^* - I_{h/2}^* \approx \frac{3}{4}kh^2 \tag{4.3}$$

Погрешность вычислений интегралов I_h^* и $I_{h/2}^*$ при этом составит:

$$\varepsilon\left(I_h^*\right) \approx kh^2 = \frac{4}{3}\left(I_h^* - I_{h/2}^*\right)$$

$$\varepsilon\left(I_{h/2}^*\right) \approx kh^2 = \frac{1}{3}\left(I_h^* - I_{h/2}^*\right) \tag{4.4}$$

Можно видеть, что для метода с порядком аппроксимации n с применением вышеописанного способа с шагами h и $h/2$:

$$\varepsilon\left(I_h^*\right) \approx \frac{1}{1-2^{-n}}\left(I_h^* - I_{h/2}^*\right)$$

$$\varepsilon\left(I^*_{h/2}\right) \approx \frac{1}{2^n - 1}\left(I^*_h - I^*_{h/2}\right) \qquad (4.5)$$

Полученная формула, конечно, будет пригодна только в случае, когда вклад вычислительной погрешности на фоне вклада погрешности численного метода пренебрежимо мал.

Имея инструмент для оценки погрешности в виде правила Рунге, можно реализовывать алгоритмы, в которых необходимое для достижения заданной точности ε значение h находится автоматически. Для этого начинают решать задачу с заведомо большим h и, уменьшают h вдвое, пока оценка Рунге не станет меньше заданной величины. Причем адаптивный метод может быть настроен как на достижение заданной абсолютной точности, так и на заданную относительную точность. Повторим, что необходимо учитывать эффекты, связанные с вычислительной погрешностью.

Задание 4.1

Реализовать адаптивные 3-узловые квадратуры Гаусса для вычисления интегралов от любых функций на отрезке от нуля до единицы. Абсолютная погрешность метода 10^{-8}.

5. Устойчивость

С понятием устойчивости мы уже познакомились в заданиях вычисления производных.

Устойчивость это свойство алгоритма решения задачи или самой задачи, при котором малое изменение входных данных приводит к малому изменению результата.

Конечно, малость возмущения и отклика должна рассматриваться в контексте конкретной задачи. Так, в сравнении с точностью представления вещественных чисел в компьютере, результат работы конечноразностной формулы (2.3.9) может быть плохим. Однако полученная абсолютная точность может быть вполне приемлемой.

Устойчивыми или неустойчивыми могут быть как сами задачи, так и численные алгоритмы их решения. Приведем примеры неустойчивых задач:

1) Уравнение $(x-a)^n=\varepsilon$. При $n=10$ и нулевой правой части малое возмущение в правой части 10^{-10} приведет к изменению результата на величину примерно 10^{-1}.

2) Пример Уилкинсона. Требуется решить алгебраическое уравнение, записанное в виде

$$x^{20}-210x^{19}+...20!=0 \qquad (5.1)$$

которое сводится к

$$(x-1)(x-2)...(x-20)=0 \qquad (5.2)$$

и имеет набор корней $x=1,2,...20$

Предположим, что о сведении к виду (5.2) мы не знаем и решаем (5.1) численно. Ошибка в задании коэффициента 210 на величину 10^{-7} изменит не только значение корней но и тип решения. Около половины корней станут комплексными.

3) Решение системы линейных алгебраических уравнений (СЛАУ):

$$\begin{cases} x+10y=11 \\ 100x+1001y=1101 \end{cases} \qquad (5.3)$$

Эта СЛАУ имеет корни $x=1,\ y=1$.

Внеся небольшие изменения:

$$\begin{cases} x+10y=11.01 \\ 100x+1001y=1101 \end{cases} \qquad (5.4)$$

Получим решения $x = 11.01,\ y = 0$.

Это были примеры неустойчивых задач, которые показывают необходимость анализа задачи на устойчивость перед ее решением. Вне зависимости от методов решения малое возмущение во входных данных влечет большое изменение в результате. Численное решение таких задач сводится к использованию представлений вещественных чисел надлежащей точности.

Неустойчивым может быть и метод решения задачи. Рассмотрим исторический пример. До широкого распространения вычислительной техники имелись справочники, в которых печатались численные значения функций для разных значений аргумента. Справочники пользовались популярностью, в особенности при вычислениях специальных функций. Одной из таких спецфункций был встречающийся в математической статистике интеграл:

$$I_n = \frac{1}{e}\int_0^1 x^n e^x dx \qquad (5.5)$$

Интеграл в (5.5) может быть взят по частям:

$$\int_0^1 x^n e^x dx = x^n e^x\Big|_0^1 - n\int_0^1 x^{n-1} e^x dx = e - n\int_0^1 x^{n-1} e^x dx \qquad (5.6)$$

Что приводит к простой рекуррентной формуле:

$$I_n = 1 - nI_{n-1} \qquad (5.7)$$

При этом I_0 легко находится аналитически:

$$I_0 = \frac{1}{e}\int_0^1 e^x dx = 1 - \frac{1}{e} \qquad (5.8)$$

Известное значение I_0 и существование рекуррентной формулы представляли собой настолько большой соблазн, что многие независимые группы, составляющие табличные значения спецфункций, даже не подумали использовать, к примеру, квадратурные формулы. В результате различные опубликованные таблицы спецфункций содержали значения I_{14} в диапазоне от –148 до 5356. Результат был явно неверным – значение подынтегральной функции на отрезке интегрирования всегда положительно и не превышает e, то есть результат вычисления функции должен быть в интервале от единицы до нуля для любого n.

Задание 5.1

Вычислить значение интеграла (5.5) для n в диапазоне от 0 до 30 с помощью рекуррентной формулы и с помощью адаптивной 3-узловой составной квадратуры Гаусса с абсолютной точностью 10^{-8} (предыдущее задание). Найти n, начиная с которого ошибка вычислений с помощью рекуррентной формулы сравнима с самим значением интеграла.

Разберем подробно причины неустойчивости метода с рекуррентной формулой. Уже на этапе введения I_0 в компьютер вносится погрешность порядка вычислительной ε. Обозначим I_0^* значение, записанное в компьютер и будем отличать его от I_0.

$$I_0^* = I_0 + \varepsilon \qquad (5.9)$$

Посмотрим, что происходит с начальной погрешностью при работе рекуррентной формулы (5.7), допустив для простоты, что все арифметические операции выполняются с бесконечной точностью:

$$I_1^* = 1 - 1 \cdot (I_0 + \varepsilon) = I_1 - 1 \cdot \varepsilon$$

$$I_2^* = 1 - 2 \cdot (I_1 - 1 \cdot \varepsilon) = I_2 + 1 \cdot 2 \cdot \varepsilon$$

$$I_3^* = 1 - 3 \cdot (I_2 + 1 \cdot 2 \cdot \varepsilon) = I_3 - 1 \cdot 2 \cdot 3 \cdot \varepsilon$$

.

.

.

$$I_n^* = I_n + (-1)^n n! \cdot \varepsilon \qquad (5.10)$$

Таким образом, погрешность нарастает как $n!$, что гарантированно «испортит» решение для любого стандартного типа данных с плавающей точкой при $n = 30$. В решении Задания 13 можно видеть смену знака ошибки при увеличении n, как и прогнозируется формулой (5.10).

Устойчивость может играть важную роль в численных методах, которые содержат большое число однотипных вычислений.

6. Решение задачи Коши для обыкновенных дифференциальных уравнений

6.1 Метод Эйлера

Множество задач, как физических, так и вычислительных, могут быть сведены к решению дифференциального уравнения с начальными условиями – к задаче Коши. Рассмотрим ОДУ первого порядка:

$$\begin{cases} y'(x) = f(x, y(x)) \\ y(0) = y_0 \\ x \in [0, X] \end{cases} \qquad (6.1.1)$$

Здесь $f(x, y)$ произвольная функция, скомбинированная из x и y.

Численные схемы для решения этой задачи могут быть получены разными способами. Будем рассматривать только *сеточные методы*, то есть такие, в которых бесконечное множество значений непрерывного аргумента x подменяется конечным дискретным множеством значений $\{x_i\}$. Такая подмена имеет аналогию в виде представления вещественных чисел в компьютере и необходима, поскольку невозможно оперировать бесконечным числом значений. Расстояние между любыми двумя ближайшими точками x_i будем считать одинаковым и равным h. Такая сетка называется *однородной*. Кроме сеточных методов существуют еще квазинепрерывные методы, в которых решение $y(x)$ представляется в виде суммы конечного числа непрерывных функций. Например, используя условие задачи Коши, можно записать её решение в виде ряда Тейлора.

Сеточные методы имеют наибольшее практическое применение, из-за простоты их реализации и анализа.

Рассмотрим три способа построения вычислительной схемы для (6.1.1).

Во-первых, можно использовать конечно-разностную аппроксимацию первой производной:

$$\frac{y(x+h) - y(x)}{h} + O(h) = f(x, y) \qquad (6.1.2)$$

Тогда, используя начальное условие $y(0)=y_0$, можно последовательно найти значения функции во всех точках $\{x_i\}$:

$$y(x+h)=y(x)+hf(x,y)+O(h^2) \tag{6.1.3}$$

или

$$y_{i+1}=y_i+hf(x_i,y_i) \tag{6.1.4}$$

Во вторых, тот же результат можно получить, раскладывая $y(x)$ в ряд Тейлора в окрестности x:

$$\begin{aligned} &y(x+h)=y(x)+y'(x)h+\frac{1}{2}y''(x)h^2+\ldots=y(x)+hf(x,y)+\frac{1}{2}f_x'(x,y)h^2+\ldots= \\ &=y(x)+hf(x,y)+O(h^2) \end{aligned} \tag{6.1.5}$$

И в третьих, исходное уравнение можно проинтегрировать на малом отрезке h:

$$\int_x^{x+h} y'(x)dx=\int_x^{x+h} f(x,y(x))dx$$

$$y(x+h)-y(x)=\int_x^{x+h} f(x,y(x))dx \tag{6.1.6}$$

Для интеграла в правой части можно использовать какой-либо метод приближенного интегрирования, например, квадратурную формулу левых прямоугольников:

$$y(x+h)-y(x)=f(x,y(x))h+O(h^2) \tag{6.1.7}$$

Откуда мы вновь приходим к выражению (6.1.4).

Заметим, что использование квадратур для вычисления интеграла в правой части (6.1.6) позволяет строить методы любого порядка точности.

Мы получили тремя различными способами явный метод Эйлера для решения задачи Коши. Порядок аппроксимации одиночной формулы – второй, однако, по той же причине, что и при переходе от одиночных квадратур к составным, при решении методом Эйлера (6.1.7) погрешность аппроксимации $y(x)$ в точке $x=X$ будет $O(h)$. Далее, когда будет упоминаться порядок аппроксимации метода, будет подразумеваться именно погрешность аппроксимации $y(x)$ в точке $x=X$.

Вычислительная схема метода Эйлера реализует рекуррентные вычисления. Как было показано в примере Главы 5, рекуррентные вычисления могут приводить к

неустойчивым методам решения. Отметим без доказательства, что явный метод Эйлера устойчив для любых ОДУ вида (6.1.1), т.е. *безусловно устойчив*.

Задание 6.1.1

Численные методы решения ОДУ можно применить к вычислению функций. Для этого надо построить соответствующее вычисляемой функции ОДУ. Например, требуется вычислить функцию $y = e^{-x^2}$ на отрезке от 0 до 1. Дифференцируя, получим $y' = -2xe^{-x^2} = -2xy$. Значит $f(x, y) = -2xy$, начальное условие $y_0 = 1$. Метод Эйлера будет выглядеть:

$$y_0 = 1$$
$$y_{i+1} = y_i - 2x_i y_i h$$

В результате работы метода Эйлера мы получим значения функции $y = e^{-x^2}$, протабулированной на отрезке $[0,1]$ с шагом h.

Построить самостоятельно ОДУ для функций:

$y = tg(x)$

$y = \ln(1 + x)$

$y = erf(x)$

Пронаблюдать порядок аппроксимации метода, сравнить с теоретическим.

6.2 Предиктор-корректор и семейство методов Рунге-Кутты

При выводе формулы метода Эйлера уже упоминалось, что интегрирование дифференциального уравнение с использованием квадратур высокого порядка точности позволит получить более точные методы решения ОДУ. Рассмотрим применение квадратур трапеций

$$y(x+h) - y(x) = \int_x^{x+h} f(x, y(x))dx = \frac{h}{2}\left(f(x, y(x)) + f(x+h, y(x+h))\right) + O(h^3) \qquad (6.2.1)$$

Сразу заметен недостаток такого способа – искомое значение $y(x+h)$ задано неявно, требуется решать уравнение, чтобы его найти. Попробуем обойти это препятствие. Введем функцию «предиктор»:

$$y^*(x+h) = y(x) + hf(x, y) \qquad (6.2.2)$$

Поскольку вид предиктора совпадает с формулой Эйлера, можно утверждать:

$$y(x+h)-y^*(x+h)=O(h^2) \tag{6.2.3}$$

Посмотрим, на какую величину Δ изменится правая часть (6.2.1) при замене $y(x+h)$ на $y^*(x+h)$:

$$\Delta=\frac{h}{2}\left[f(x+h,y(x+h))-f(x+h,y^*(x+h))\right] \tag{6.2.4}$$

Правая часть равенства (6.2.4) может быть представлена через определенный интеграл:

$$\Delta=\frac{h}{2}\int\limits_{y^*(x+h)}^{y(x+h)} f'_y(x+h,y)dy \tag{6.2.5}$$

который, в свою очередь, может быть оценен, например, квадратурой центральных прямоугольников:

$$\Delta=\frac{h}{2}\left[f\left(x+h,\frac{y(x+h)+y^*(x+h)}{2}\right)(y(x+h)-y^*(x+h))+O(h^3)\right] \tag{6.2.6}$$

Откуда, учитывая (6.2.3), получаем:

$$\Delta=\frac{h}{2}\left[f\left(x+h,\frac{y(x+h)+y^*(x+h)}{2}\right)O(h^2)+O(h^3)\right]=O(h^3) \tag{6.2.7}$$

То есть замена $y(x+h)$ на $y^*(x+h)$ не ухудшает порядок аппроксимации формулы (6.2.1). Это значит, что для нахождения $y(x+h)$ по известному $y(x)$ можно использовать последовательность из двух формул:

$$\begin{cases} y^*(x+h)=y(x)+hf(x,y) \\ y(x+h)=y(x)+\dfrac{h}{2}\left(f(x,y(x))+f(x+h,y^*(x+h))\right)+O(h^3) \end{cases} \tag{6.2.8}$$

Вторая формула из этой последовательности называется корректором, а сам метод – методом прогноза и коррекции.

Переходя к сеточным обозначениям:

$$\begin{cases} y_{i+1}{}^*=y_i+hf(x_i,y_i) \\ y_{i+1}=y_i+\dfrac{h}{2}\left(f(x_i,y_i)+f(x_{i+1},y_{i+1})\right) \end{cases} \tag{6.2.9}$$

Конечно, предиктор и корректор может быть получен и с помощью замены интеграла в (6.2.1) квадратурой центральных прямоугольников. В этом случае получится последовательность формул:

$$\begin{cases} y_{i+1/2}* = y_i + \dfrac{h}{2} f(x_i, y_i) \\ y_{i+1} = y_i + hf(x_{i+1/2}, y_{i+1/2}*) \end{cases} \qquad (6.2.10)$$

Задание 6.2.1

Повторить задание 6.1.1 с использованием методов прогноза и коррекции.

Замена интеграла в (6.2.1) может быть сделана квадратурной формулой сколь угодно большого порядка аппроксимации, и, таким образом, могут быть получены последовательности формул сколь угодно высокого порядка точности. Все полученные таким образом методы решения ОДУ, а также метод Эйлера, составляют семейство методов Рунге-Кутты.

Общий вид методов Рунге-Кутты (без вывода):

$$\begin{cases} k_1(h) = hf(x, y) \\ k_2(h) = hf(x + \alpha_2 h, y + \beta_{2,1} k_1(h)) \\ . \\ . \\ k_q(h) = hf(x + \alpha_q h, y + \beta_{q,1} k_1(h) + .. + \beta_{q,q-1} k_{q-1}(h)) \\ y(x+h) = y(x) + \sum_{i=1}^{q} p_i k_i(h) + O(h^{s+1}) \end{cases} \qquad (6.2.11)$$

где $\alpha_2,..,\alpha_q$, $p_1,..,p_q$, $\beta_{i,j}$, $0 < j < i \le q$ - некоторые числовые параметры, не зависящие от решаемого уравнения. Они выбираются из соображений максимально высокого порядка аппроксимации $s+1$ одиночной формулы.

Сделаем выбор числовых параметров. Введем обозначение для приближенного решения:

$$z(h) = y(x) + \sum_{i=1}^{q} p_i k_i(h) \qquad (6.2.12)$$

Запишем погрешность аппроксимации φ как функцию от h:

$$\varphi(h) = y(x+h) - z(h)$$

Если потребовать от метода порядок точности s, это означает, что в ряде Тейлора:

$$\varphi(h) = \sum_{i=1}^{\infty} \frac{\varphi^{(i)}(0)}{i!} h^i \qquad (6.2.13)$$

все слагаемые со степенями до s включительно равны нулю. Тогда:

$$\varphi(0)=\varphi'_h(0)=\varphi''_{hh}(0)=...=\varphi^{(s)}_{h..}(0) \quad (6.2.14)$$

Значение $\varphi(0)$ должно быть равно нулю автоматически, этого требует условие сходимости метода.

Порядок аппроксимации одиночной формулы $s+1$ на единицу больше порядка аппроксимации s в правой точке решения, он не имеет простой зависимости от числа выражений q. Для порядка аппроксимации до 4 включительно $s=q$, а для построения метода пятого порядка точности потребуется уже $q=6$.

Рассмотрим построение метода Рунге Кутты для случая $q=2$.

Конечное выражение метода Рунге-Кутты:

$$y(x+h)=y(x)+p_1hf(x,y)+p_2hf(x_2,y_2)+O(h^{s+1}) \quad (6.2.15)$$

где

$$\begin{aligned} x_2 &= x+\alpha_2 h \\ y_2 &= y+\beta_{2,1}hf(x,y) \end{aligned} \quad (6.2.16)$$

Заметим, что при $h=0$:

$$\begin{aligned} x_2 &= x \\ y_2 &= y \end{aligned} \quad (6.2.17)$$

А также что:

$$y'_h(x+h)=y'_x(x+h) \quad (6.2.18)$$

При $h=0$:

$$y'_x(x+h)=y'_x(x)=f(x,y) \quad (6.2.19)$$

Функция ошибки:

$$\varphi(h)=y(x+h)-y(x)-h(p_1f(x,y)+p_2f(x_2,y_2)) \quad (6.2.20)$$

Находим производные:

$$\begin{aligned} \varphi'_h(h) &= y'_x(x+h)-(p_1f(x,y)+p_2f(x_2,y_2))-h(p_1f(x,y)+p_2f(x_2,y_2))'_h \\ \varphi'_h(0) &= f(x,y)-f(x,y)(p_1+p_2)=0 \end{aligned} \quad (6.2.21)$$

Получаем первое условие на числовые параметры:

$$1-p_1-p_2=0 \quad (6.2.22)$$

Вторая производная:

$$\varphi''_{hh}(h) = y''_{hh}(x+h) - 2\left(p_1 f(x,y) + p_2 f(x_2,y_2)\right)'_h - $$
$$- h\left(p_1 f(x,y) + p_2 f(x_2,y_2)\right)''_h = $$
$$= f'_h(x+h, y(x+h)) - 2\left(p_2\alpha_2 f'_{x2}(x_2,y_2) + p_2 (y_2)'_h f'_{y2}(x_2,y_2)\right) - \quad (6.2.23)$$
$$- h\left(p_1 f(x,y) + p_2 f(x_2,y_2)\right)''_{hh}$$

Найдем $(y_2)'_h$

$$(y_2)'_h = \beta_{2,1}(k_1)'_h = \beta_{2,1} f(x,y) \quad (6.2.24)$$

Подставим (6.2.24) в (6.2.23):

$$\varphi''_{hh}(h) = f'_h(x+h, y(x+h)) - 2\left(p_2\alpha_2 f'_{x2}(x_2,y_2) + p_2\beta_{2,1} f(x,y) f'_{y2}(x_2,y_2)\right) - h(...)''_{hh} =$$
$$= f'_x(x+h, y(x+h)) + y'_h f'_y(x+h, y(x+h)) - 2\left(p_2\alpha_2 f'_{x2}(x_2,y_2) + p_2\beta_{2,1} f(x,y) f'_{y2}(x_2,y_2)\right) - h(...)''_{hh} =$$
$$= f'_x(x+h, y(x+h)) + f(x+h, y(x+h)) f'_y(x+h, y(x+h)) -$$
$$- 2p_2\alpha_2 f'_{x2}(x_2,y_2) - 2p_2\beta_{2,1} f(x,y) f'_{y2}(x_2,y_2) - h(...)''_{hh} \quad (6.2.25)$$

Рассмотрим условие $\varphi''_h(0) = 0$:

$$\varphi''_{hh}(0) = f'_x + f f'_y - 2p_2\alpha_2 f'_x - 2p_2\beta_{2,1} f f'_y = f'_x(1 - 2p_2\alpha_2) + f f'_y(1 - 2p_2\beta_{2,1}) = 0 \quad (6.2.26)$$

Понятно, что в общем случае оно выполняется, если:

$$1 - 2p_2\alpha_2 = 0$$
$$1 - 2p_2\beta_{2,1} = 0 \quad (6.2.27)$$

В итоге, для построения метода Рунге-Кутты второго порядка аппроксимации требуется решить систему нелинейных уравнений:

$$\begin{cases} 1 - p_1 - p_2 = 0 \\ 1 - 2p_2\alpha_2 = 0 \\ 1 - 2p_2\beta_{2,1} = 0 \end{cases} \quad (6.2.28)$$

Эта система имеет множество решений. Так, для $p_1 = p_2 = 1/2$, $\alpha_2 = 1$, $\beta_{2,1} = 1$ получится метод (6.2.9), а для $p_1 = 0$, $p_2 = 1$, $\alpha_2 = 1/2$, $\beta_{2,1} = 1/2$ получим (6.2.10). Оба метода имеют второй порядок аппроксимации, по системе уравнений (6.2.28) можно получить и другие варианты метода.

Видно, что даже в случае $q = 2$ вывод последовательности формул методов Рунге-Кутты получается довольно громоздкий. Описанная выше схема, сводящая задачу построения методов Рунге-Кутты к решению систем нелинейных уравнений,

позволяет, в принципе, получить метод сколь угодно высокого порядка аппроксимации.

Приведем здесь одну из возможных последовательностей формул для метода Рунге-Кутты третьего порядка аппроксимации:

$$\begin{aligned} k_1 &= hf(x, y) \\ k_2 &= hf(x + h/2, y + k_1/2) \\ k_3 &= hf(x + h, y - k_1 + 2k_2) \\ y(x+h) &= y(h) + \frac{1}{6}(k_1 + 4k_2 + k_3) + O(h^4) \end{aligned} \qquad (6.2.29)$$

И четвертого порядка аппроксимации:

$$\begin{aligned} k_1 &= hf(x, y) \\ k_2 &= hf(x + h/2, y + k_1/2) \\ k_3 &= hf(x + h/2, y + k_2/2) \\ k_4 &= hf(x + h, y + k_3) \\ y(x+h) &= y(h) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) + O(h^5) \end{aligned} \qquad (6.2.30)$$

Задание 6.2.2

Повторить задание 6.1.1, 6.2.1 с использованием методов третьего и четвертого порядка аппроксимации.

7. Численные методы линейной алгебры: обратные матрицы

Решение систем линейных алгебраических уравнений (СЛАУ) или эквивалентная задача нахождения обратной матрицы – ключевая в численных методах линейной алгебры. Матричные уравнения могут встретиться в любой численной задаче, где возникает упорядоченная совокупность значений, и эту совокупность можно рассматривать как вектор.

Умение находить обратную матрицу фактически дополняет операцией деления набор таких легко реализуемых операций над матрицами, как сложение, вычитание и умножение. После этого, решение задач с векторами и матрицами становится не намного сложнее решения обычных скалярных линейных уравнений. Нужно лишь учитывать некоммутативность операции умножения матриц.

Почему нахождение обратной матрицы может представлять трудность? Хорошо известный из линейной алгебры метод Крамера позволяет просто находить обратную матрицу через ее детерминант и алгебраические дополнения. Однако число слагаемых в детерминанте матрицы зависит от размера матрицы n как $n!$ и уже при $n=10$ метод Крамера будет занимать слишком много времени. Такие алгоритмы, в которых объем вычислений растет как $O(n!)$ или пропорционален $O(e^n)$ называются неполиномиальными. Задачей численных методов нахождения обратных матриц является нахождение полиномиального алгоритма, т.е. такого, в котором число арифметических операций будет зависеть от размерности задачи как $O(n^k)$, k - некоторое число, постоянное для алгоритма. Так, операция сложения матриц подразумевает *сложность алгоритма* $O(n^2)$ а умножения - $O(n^3)$.

7.1 Метод Гаусса

Метод Гаусса представляет собой метод последовательного исключения переменных, схожие действия обычно используются при решении СЛАУ вручную. Именно поэтому алгоритм Гаусса очень легко запоминается. Чтобы найти обратную матрицу последовательным исключением переменных, нужно ввести вспомогательную единичную матрицу. После чего операциями умножения строк на число и сложения строк добиваются сведения оборачиваемой матрицы к

верхнетреугольной, затем к диагональной, и, как завершающий этап, к единичной. Все операции, производимые над оборачиваемой матрицей, в точности повторяются со вспомогательной единичной матрицей. В итоге, вспомогательная матрица будет содержать обратную матрицу, а на месте исходной будет единичная.

Задание 7.1.1

Реализовать алгоритм Гаусса для нахождения обратной матрицы $\hat{A}^{-1}$ произвольной размерности n. Строки и столбцы матрицы считать пронумерованными от 1 до n. Элементы исходной матрицы $A_{i,j} = 1/(i+j-1)$. Оценить число арифметических операций как функцию от n.

7.2 LU-разложение

Алгоритм Гаусса можно разбить на два этапа. Матрицу $\hat{A}$ представим как произведение двух матриц частного вида:

$$\hat{A} = \hat{L}\hat{U} \tag{7.2.1}$$

где $\hat{L}$ - нижнедиагональная матрица, т.е. такая матрица, у которой все элементы выше главной диагонали равны нулю, а $\hat{U}$ - верхнедиагональная матрица с единичной главной диагональю.

Для матриц $\hat{L}$ и $\hat{U}$ легко найти соответствующие обратные матрицы с помощью метода Крамера, поскольку детерминанты и все алгебраические дополнения будут содержать лишь по одному слагаемому, составленному из произведения элементов главной диагонали. Обратная матрица $\hat{A}^{-1}$ тогда будет равна:

$$\hat{A}^{-1} = \hat{U}^{-1}\hat{L}^{-1} \tag{7.2.2}$$

Рассмотрим детально произведение $\hat{L}\hat{U}$ на частном примере матрицы $n = 4$:

$$\begin{pmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{pmatrix} \begin{pmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \tag{7.2.3}$$

$$= \begin{pmatrix} l_{11} & l_{11}u_{12} & l_{11}u_{13} & l_{11}u_{14} \\ l_{21} & l_{21}u_{12} + l_{22} & l_{21}u_{13} + l_{22}u_{23} & l_{21}u_{14} + l_{22}u_{24} \\ l_{31} & l_{31}u_{12} + l_{32} & l_{31}u_{13} + l_{32}u_{23} + l_{33} & l_{31}u_{14} + l_{32}u_{24} + l_{33}u_{34} \\ l_{41} & l_{41}u_{12} + l_{42} & l_{41}u_{13} + l_{42}u_{23} + l_{43} & l_{41}u_{14} + l_{42}u_{24} + l_{43}u_{34} + l_{44} \end{pmatrix}$$

Откуда видно способ нахождения матриц $\hat{L}$ и $\hat{U}$. Находим первую строку матрицы $\hat{U}$. Первый столбец матрицы $\hat{L}$ известен сразу:

$$l_{k,1} = a_{k,1}$$

$$k = 1..n \qquad (7.2.4)$$

Зная его, можно найти первую строку матрицы $\hat{U}$:

$$u_{1,k} = \frac{a_{1,k}}{l_{1,1}}$$

$$k = 2..n \qquad (7.2.5)$$

Теперь, зная первый столбец $\hat{L}$ и первую строку $\hat{U}$, можно найти вторые строку и столбец.

Представим алгоритм LU-разложения следующим образом. Будем осуществлять операции над матрицей $\hat{a}$, начальными элементами которой являются элементы оборачиваемой матрицы $\hat{A}$. Будем производить арифметические действия над элементами a_{ij} задавшись целью получить в конечном итоге на месте матрицы $\hat{a}$ матрицу, содержащую одновременно $\hat{L}$ и $\hat{U}$. Для этого осуществим следующую последовательность действий:

1) обработка k-й строки, $k = 1,..,n-1$

$$a_{ik} = a_{ik} / a_{kk},\ i = k+1,..,n \qquad (7.2.6)$$

2) обработка элементов $i, j = k+1,..,n$

$$a_{i,j} = a_{i,j} - a_{k,j} a_{i,k} \qquad (7.2.7)$$

В результате этих действий на месте матрицы $\hat{a}$ получим матрицу $\hat{L} + \hat{U} - \hat{E}$.

Найдем теперь обратную матрицу $\hat{A}^{-1}$, элементы которой обозначим r_{ij}. Умножим (7.2.2) на $\hat{L}$ справа и на $\hat{U}$ слева. Получим:

$$\hat{A}^{-1}\hat{L} = \hat{U}^{-1}$$

$$\hat{U}\hat{A}^{-1} = \hat{L}^{-1} \qquad (7.2.8)$$

Учтем, что обратные к треугольным матрицы имеют ту же структуру, что и исходные матрицы. Тогда:

$$\begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{pmatrix} \begin{pmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{pmatrix} = \begin{pmatrix} 1 & * & * & * \\ 0 & 1 & * & * \\ 0 & 0 & 1 & * \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{pmatrix} = \begin{pmatrix} * & 0 & 0 & 0 \\ * & * & 0 & 0 \\ * & * & * & 0 \\ * & * & * & * \end{pmatrix} \quad (7.2.9)$$

Звездочками обозначены числа, не играющие роли в нахождении обратной матрицы.

Из структуры перемножаемых матриц можно видеть способ нахождения элементов обратной матрицы. Посмотрим на произведение четвертой строки и четвертого столбца в верхнем уравнении (7.2.9):

$$x_{44}l_{44} = 1,\ x_{44} = 1/l_{44} \quad (7.2.10)$$

Далее на произведение четвертой строки и третьего столбца:

$$x_{43}l_{33} + x_{44}l_{43} = 0,\ x_{43} = -x_{44}l_{43}/l_{33} \quad (7.2.11)$$

Четвертой строки и второго столбца:

$$x_{42}l_{22} + x_{43}l_{32} + x_{44}l_{42} = 0,\ x_{42} = -\left(x_{43}l_{32} + x_{44}l_{42}\right)/l_{22} \quad (7.2.12)$$

Четвертой строки и первого столбца:

$$x_{41}l_{11} + x_{42}l_{21} + x_{43}l_{31} + x_{44}l_{41} = 0,\ x_{41} = -\left(x_{42}l_{21} + x_{43}l_{31} + x_{44}l_{41}\right)/l_{11} \quad (7.2.13)$$

Теперь нам известна 4-я строка обратной матрицы.

Далее рассмотрим произведение третьей строки и четвертого столбца в нижнем уравнении (7.2.9):

$$x_{34} + u_{34}x_{44} = 0,\ x_{34} = -u_{34}x_{44} \quad (7.2.14)$$

Второй строки на четвертый столбец:

$$x_{24} + u_{23}x_{34} + u_{24}x_{44} = 0,\ x_{24} = -\left(u_{23}x_{34} + u_{24}x_{44}\right) \quad (7.2.15)$$

Первой строки на четвертый столбец:

$$x_{14} + u_{12}x_{24} + u_{13}x_{34} + u_{14}x_{44} = 0,\ x_{14} = -\left(u_{12}x_{24} + u_{13}x_{34} + u_{14}x_{44}\right) \quad (7.2.16)$$

В итоге мы нашли четвертую строку и четвертый столбец обратной матрицы. Далее процесс можно повторять, пока не будет найдена вся обратная матрица. В итоге, для построения обратной матрицы по известным матрицам $\hat{L}$ и $\hat{U}$ нужно выполнить следующую рекурсивную последовательность действий:

Для $i = n,..,1$:

$$x_{ii} = \frac{1}{l_{ii}}\left(1 - \sum_{k=i+1}^{n} x_{ik}l_{ki}\right)$$

$$x_{ij} = -\frac{1}{l_{jj}} \sum_{k=j+1}^{n} x_{ik} l_{kj}\ ,\ \ j = i-1,..,1$$

$$x_{ji} = -\sum_{k=j+1}^{n} u_{jk} x_{ki}\ ,\ \ j = i-1,..,1 \qquad (7.2.17)$$

Подчеркнем еще раз, что последовательность действий рекурсивная, следующая строка (7.2.17) использует результат работы предыдущей.

Задание 7.2.1

Реализовать алгоритм нахождения обратной матрицы $\hat{A}^{-1}$ произвольной размерности n с использованием LU-разложения. Строки и столбцы матрицы считать пронумерованными от 1 до n. Элементы исходной матрицы $A_{i,j} = 1/(i+j-1)$. Оценить число арифметических операций как функцию от n.

Повторим, что по существу, нахождение обратной матрицы с помощью LU-разложения представляет собой модифицированный алгоритм Гаусса, более структурированный и легкий в отладке.

Несмотря на небольшой объем получившийся программы, при выполнении заданий 7.1.1 и 7.2.1 можно было видеть, что малейшая ошибка в реализации алгоритмов приведет к неработоспособности всей программы, а само написание программы требует большой концентрации внимания.

Рассмотрим далее более простые для реализации способы нахождения обратных матриц.

7.3 Итерационные алгоритмы

Среди всех алгоритмов нахождения обратных матриц итерационные алгоритмы, наверное, самые красивые и простые, как в изложении, так и в практической реализации. Единственный их недостаток следует из названия – алгоритмы представляют собой рекуррентные соотношения, осуществляя которые мы получаем все более точное приближение к обратной матрице. В алгоритмах Гаусса и LU-разложения обратная матрица получается после фиксированного числа

арифметических операций. Впрочем, этот недостаток может оказаться не таким уж значительным, поскольку в любом случае мы получаем матрицу с элементами, точность которых ограничена вычислительной погрешностью.

Рассмотрим *метод простой итерации*. В его основе лежит идея о том, что алгебра матриц почти не отличается от алгебры вещественных чисел (кроме некоммутативности умножения). Операцию обращения матрицы можно представить посредством операции деления.

$$\hat{A}^{-1} = \frac{1}{\hat{A}} = \frac{1}{\hat{E} - \left(\hat{E} - \hat{A}\right)} \tag{7.3.1}$$

Вводя матрицу $\hat{B} = \hat{E} - \hat{A}$, используем разложение в ряд Тейлора:

$$\hat{A}^{-1} = \frac{1}{\hat{E} - \hat{B}} = \hat{E} + \hat{B} + \hat{B}^2 + \hat{B}^3 + ... = \hat{E}\left(\hat{E} + \hat{B}\left(\hat{E} + \hat{B}\left(\hat{E} + \hat{B}(...)\right)\right)\right) \tag{7.3.2}$$

аналогичное разложению в ряд Тейлора функции:

$$\frac{1}{1-x} = \sum_{k=1}^{\infty} x^k \tag{7.3.3}$$

в окрестности нуля.

В частном случае матриц с размерностью единица мы имеем именно ряд (7.3.3).

То есть операцию обращения матрицы можно свести к бесконечной последовательности операций сложения и умножения матриц.

Вводя начальное приближение $\hat{R}_0$ для обратной матрицы $\hat{R} = \hat{A}^{-1}$ получаем формулу метода простой итерации:

$$\hat{R}_{k+1} = \hat{E} + \hat{B}\hat{R}_k \tag{7.3.4}$$

Задание 7.3.1

Реализовать алгоритм нахождения обратной матрицы $\hat{A}^{-1}$ произвольной размерности n с использованием метода простой итерации. Строки и столбцы матрицы считать пронумерованными от 1 до n. Элементы исходной матрицы $A_{i,j} = 1/(i + j - 1)$. Оценить число итераций, которые требуются для нахождения элементов обратной матрицы с точностью 3-х знаков после запятой. Проделать эту оценку для разных n.

В методе простой итерации может потребоваться большое число итераций и, соответственно, много времени на достижение результата. Сходимость ряда (7.3.2)

будет полностью определяться свойствами матрицы $\hat{B}$, и, в общем случае, не каждая матрица $\hat{B}$ будет давать быстро сходящийся ряд (7.3.2). Развитием метода простой итерации является метод минимальных невязок или *метод Шульца*.

Идея метода Шульца состоит в получении степенного ряда, аналогичного (7.3.2), но по такой матрице $\hat{\psi}$, для которой ряд быстро сходится.

Для этого в методе Шульца, на каждой итерации k вводится невязка $\hat{\psi}_k$:

$$\hat{\psi}_k = \hat{E} - \hat{A}\hat{R}_k \qquad (7.3.5)$$

Невязкой она называется потому, что обращается в ноль при $\hat{R}_k = \hat{A}^{-1}$. По мере приближения $\hat{R}_k$ к обратной матрице невязка будет приближаться к нулевой матрице.

Рассмотрим выражение:

$$\frac{1}{\hat{E} - \hat{\psi}_k} = \hat{E} + \hat{\psi}_k + \hat{\psi}_k^2 + ... = \frac{1}{\hat{A}\hat{R}_k} = \hat{R}_k{}^{-1}\hat{A}^{-1} \qquad (7.3.6)$$

Умножим обе части слева на $\hat{R}_k$ и получим тождество:

$$\hat{A}^{-1} = \hat{R}_k\left(\hat{E} + \hat{\psi}_k + \hat{\psi}_k^2 + ...\right) \qquad (7.3.7)$$

То есть, получено разложение в ряд по уменьшающемуся параметру $\hat{\psi}_k$.

Обычно, ограничиваются только линейным слагаемым в ряде (7.3.7). В результате формулы метода Шульца выглядят:

$$\hat{\psi}_k = \hat{E} - \hat{A}\hat{R}_k$$

$$\hat{R}_{k+1} = \hat{R}_k\left(\hat{E} + \hat{\psi}_k\right) \qquad (7.3.8)$$

Почему в подавляющем большинстве случаев достаточно ограничиваться линейным слагаемым в ряде (7.3.7), предлагается выяснить в следующем задании.

Задание 7.3.2

Реализовать алгоритм нахождения обратной матрицы $\hat{A}^{-1}$ произвольной размерности n с использованием метода Шульца. Элементы исходной матрицы $A_{i,j} = 1/(i + j - 1)$. Оценить число итераций, которые требуются для нахождения элементов обратной матрицы с точностью 3-х знаков после запятой. Проделать эту оценку для разных n. Сравнить с методом простой итерации.

8. Задачи оптимизации

Задачи оптимизации возникают во многих областях науки и техники. Математическая формулировка задачи оптимизации заключается в поиске экстремума функции многих переменных. Поскольку условием экстремума является равенство нулю всех частных производных, эквивалентной является задача поиска решения системы нелинейных, в общем случае, уравнений.

Без потери общности, будем рассматривать задачи оптимизации на примере приближения некоторой функции другой функцией, что в точности является задачей обработки экспериментальных данных некоторой модельной функцией.

В качестве критерия близости экспериментальной и модельной функций обычно используется сумма квадратов отклонений s^2 или статистический критерий χ^2. Такой выбор обусловлен статистической природой отклонения эксперимента от модели, связанного с вкладом разных случайных факторов. Для решения задачи требуется найти минимум s^2 или χ^2.

Модель, как правило, определяется конечным числом n переменных параметров $\{\alpha_k\}$, а саму функцию, для которой ищется экстремум, можно представить как:

$$s^2 = s^2(\alpha_1, \alpha_2, ..., \alpha_n) = s^2(\overline{\alpha}) \qquad (8.1)$$

Здесь упорядоченная совокупность параметров $\{\alpha_k\}$ обозначена через вектор $\overline{\alpha}$.

8.1. Поиск экстремума функции и задача поиска корней уравнений

Рассмотрим одномерную задачу. Требуется найти минимум функции $f(\alpha)$ на заданном отрезке $[\alpha^L, \alpha^R]$.

Самый очевидный способ поиска экстремума – ввести на интервале $[\alpha^L, \alpha^R]$ сетку с шагом h и, перебирая значения функции в узлах сетки, найти узел, в котором функция минимальна.

Задание 8.1.1

Реализовать простейший алгоритм поиска экстремума перебором значения функции $F(\alpha)=\left(1+(\alpha-0.5)^2\right)^{-1}$ в узлах сетки. Оценить среднее время работы алгоритма для различных h.

Гораздо более эффективные методы получаются из рассмотрения эквивалентной задачи решения нелинейного уравнения:

$$\frac{\partial F}{\partial \alpha}=f(\alpha)=0 \qquad (8.1.1)$$

Рассмотрим метод половинного деления (дихотомии).

Если на участке поиска экстремума содержится только один экстремум, то $f(\alpha^L)$ и $f(\alpha^R)$ должны иметь разные знаки, а точка $f(\alpha)=0$ должна быть единственной. В этом случае мы можем разделить отрезок $[\alpha^L,\alpha^R]$ пополам и из двух получившихся отрезков выбрать тот, на границах которого функция $f(\alpha)$ снова имеет разные знаки. Далее процесс половинного деления можно повторять до тех пор, пока ширина получившегося отрезка (ему будет принадлежать корень уравнения (8.1.1)) не станет меньше наперед заданной точности.

Задание 8.1.2

Реализовать алгоритм дихотомии для поиска экстремума функции $F(\alpha)=\left(1+(\alpha-0.5)^2\right)^{-1}$. Оценить среднее время работы алгоритма для различных значений погрешности.

Рассмотрим метод Ньютона (касательных).

В этом методе выбирается начальное приближение α_0 к корню уравнения (8.1.1), затем функция $f(\alpha)$ аппроксимируется в окрестности точки α_0 прямой линией:

$$f(\alpha)\approx f(\alpha_0)+f'(\alpha_0)(\alpha-\alpha_0) \qquad (8.1.2)$$

Эта прямая будет являться касательной к графику функции $f(\alpha)$ в точке α_0.

Далее находится следующее приближение α_1 как корень уравнения:

$$f(\alpha_0)+f'(\alpha_0)(\alpha_1-\alpha_0)=0$$

$$\alpha_1=\alpha_0-\frac{f(\alpha_0)}{f'(\alpha_0)} \qquad (8.1.3)$$

Процесс построения касательной и нахождение корня аппроксимирующего уравнения повторяется. В результате поиск корня сводится к рекуррентному соотношению:

$$\alpha_{k+1} = \alpha_k - \frac{f(\alpha_k)}{f'(\alpha_k)} \qquad (8.1.4)$$

Задание 8.1.3

Реализовать алгоритм Ньютона для поиска экстремума функции $F(\alpha) = \left(1 + (\alpha - 0.5)^2\right)^{-1}$. Оценить среднее время работы алгоритма для различных значений погрешности.

Метод Ньютона играет важную роль при построении методов поиска экстремума функции многих переменных.

8.2. Метод наименьших квадратов для произвольной функции

Рассмотрим задачу обработки экспериментальных данных в случае нескольких модельных параметров. Из условия экстремума суммы квадратов отклонений строим систему нелинейных уравнений:

$$\begin{cases} \dfrac{\partial s^2}{\partial \alpha_1} = f_1(\overline{\alpha}) = 0 \\ \dfrac{\partial s^2}{\partial \alpha_2} = f_2(\overline{\alpha}) = 0 \\ . \\ . \\ \dfrac{\partial s^2}{\partial \alpha_n} = f_n(\overline{\alpha}) = 0 \end{cases} \qquad (8.2.1)$$

Здесь $f_k(\overline{\alpha})$ введены для удобства записи.

Из задания 8.1.1 была видна низкая эффективность метода прямого перебора значений функции. Попробуем применить методы решения одномерной задачи к решению системы нелинейных уравнений.

Будем находить α_1 из уравнения $\frac{\partial s^2}{\partial \alpha_1} = 0$, затем α_2 из $\frac{\partial s^2}{\partial \alpha_2} = 0$ и т.д. Перебрав все переменные, снова вернемся к α_1. Будем повторять этот процесс циклически, до тех пор, пока $|\overline{\alpha}| > \varepsilon$ (условие точности нахождения экстремума). Эта последовательность действий составляет метод *покоординатного спуска*. Для решения одномерных нелинейных уравнений можно использовать как метод Ньютона, так и метод дихотомии.

Задание 8.2.1

Реализовать алгоритм покоординатного спуска для задачи аппроксимации параболой $P(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2$ функции e^{-x} на отрезке $x \in [0,1]$. Использовать методы Ньютона и дихотомии для поиска экстремума функции. Оценить среднее время работы алгоритма для различных методов и различных значений погрешности.

Метод покоординатного спуска дает очень плохие результаты в случае коррелирующих параметров модели.

Метод Ньютона можно применить к системе уравнений (8.2.1) иначе. Подобно одномерному случаю, рассмотрим линеаризацию уравнений в окрестности некоторой точки $\overline{\alpha}^0$:

$$\begin{cases} f_1(\overline{\alpha}^0) + \sum_{k=0}^{n} \frac{\partial f_1(\alpha_k^0)}{\partial \alpha_k^0}(\alpha_k^1 - \alpha_k^0) = 0 \\ f_2(\overline{\alpha}^0) + \sum_{k=0}^{n} \frac{\partial f_2(\alpha_k^0)}{\partial \alpha_k^0}(\alpha_k^1 - \alpha_k^0) = 0 \\ . \\ . \\ f_n(\overline{\alpha}^0) + \sum_{k=0}^{n} \frac{\partial f_n(\alpha_k^0)}{\partial \alpha_k^0}(\alpha_k^1 - \alpha_k^0) = 0 \end{cases} \quad (8.2.2)$$

Это уже СЛАУ, аппроксимирующая (8.2.1), и ее можно записать в матричном виде:

$$\overline{V}^0 + \hat{A}^k(\overline{a}^1 - \overline{a}^0) = 0 \quad (8.2.3)$$

Или, переходя к итерации с номером k:

$$\overline{a}^{k+1} = \overline{a}^k - (\hat{A}^k)^{-1}\overline{V}^k \quad (8.2.4)$$

Это выражение составляет метод *наискорейшего спуска*, нечувствительно к корреляции параметров модели.

Вектор $\overline{V}$ составлен из первых производных от функции, экстремум которой ищется.

Матрица:

$$\hat{A}=\begin{pmatrix} \frac{\partial^2 s^2}{\partial\alpha_1\partial\alpha_1} & \frac{\partial^2 s^2}{\partial\alpha_1\partial\alpha_2} & \cdot & \cdot \\ \frac{\partial^2 s^2}{\partial\alpha_2\partial\alpha_1} & \frac{\partial^2 s^2}{\partial\alpha_2\partial\alpha_2} & & \\ \cdot & & \cdot & \\ \cdot & & & \cdot \end{pmatrix} \qquad (8.2.5)$$

играет важную роль в оценке доверительных интервалов определяемых параметров, чего мы в данном курсе не касаемся.

Задание 8.2.2

Реализовать алгоритм наискорейшего спуска для задачи аппроксимации параболой $P(x)=\alpha_0+\alpha_1 x+\alpha_2 x^2$ функции e^{-x} на отрезке $x\in[0,1]$. Оценить среднее время работы алгоритма для различных значений погрешности.

8.3. Метод наименьших квадратов для полиномов

Из вывода метода наискорейшего спуска можно было заметить, что для аппроксимирующих функций $\Phi(x)$ некоторого вида, а именно:

$$\Phi(x)=\sum_{k=0}^{n}\alpha_k\varphi_k(x) \qquad (8.3.1)$$

метод наискорейшего спуска должен на первой же итерации давать точный ответ.

Рассмотрим подробнее. Введем сетку на отрезке аппроксимации $x\in[0,1]$, содержащую N точек. Совокупность значений аппроксимируемой функции $\{\Phi(x_i)\}$ и аппроксимируемой $\{f(x_i)\}$ в узлах сетки $\{x_i\}$обозначим через вектора $\overline{\Phi}$ и $\bar{f}$ в N-мерном евклидовом пространстве. Тогда сумма квадратов отклонений может быть записана:

$$s^2 = \left(\overline{\Phi} - \bar{f}, \overline{\Phi} - \bar{f}\right) = \left(\bar{f}, \bar{f}\right) - 2\left(\overline{\Phi}, \bar{f}\right) + \left(\overline{\Phi}, \overline{\Phi}\right) \tag{8.3.2}$$

Наложим условие экстремума как равенство нулю всех частных производных:

$$\frac{\partial s^2}{\partial \alpha_j} = -2\frac{\partial\left(\overline{\Phi}, \bar{f}\right)}{\partial \alpha_j} + \frac{\partial\left(\overline{\Phi}, \overline{\Phi}\right)}{\partial \alpha_j} = 0, \ j = 0,..,n \tag{8.3.3}$$

В свою очередь $\overline{\Phi}$ можно представить как:

$$\overline{\Phi} = \sum_{k=0}^{n} \alpha_k \overline{\varphi}_k \tag{8.3.4}$$

Тогда:

$$\frac{\partial\left(\overline{\Phi}, \bar{f}\right)}{\partial \alpha_j} = \left(\bar{f}, \overline{\varphi}_j\right) \tag{8.3.5}$$

Чтобы найти $\frac{\partial\left(\overline{\Phi}, \overline{\Phi}\right)}{\partial \alpha_j}$, удобно представить $\left(\overline{\Phi}, \overline{\Phi}\right) = \sum_{j=0}^{n}\sum_{i=0}^{n} \alpha_i \alpha_j \left(\overline{\varphi}_i, \overline{\varphi}_j\right)$ в виде таблицы:

$$\begin{pmatrix} \alpha_0\alpha_0\left(\overline{\varphi}_0, \overline{\varphi}_0\right) & \alpha_0\alpha_1\left(\overline{\varphi}_0, \overline{\varphi}_1\right) & . & . \\ \alpha_0\alpha_1\left(\overline{\varphi}_0, \overline{\varphi}_1\right) & \alpha_1\alpha_1\left(\overline{\varphi}_1, \overline{\varphi}_1\right) & & \\ . & & . & \\ . & & & . \end{pmatrix} \tag{8.3.6}$$

Тогда, дифференцируя, к примеру, по α_1, можно видеть, что результатом будет нулевой вклад всех столбцов и строк, кроме 1-х:

$$\frac{\partial\left(\overline{\Phi}, \overline{\Phi}\right)}{\partial \alpha_j} = 2\sum_{i=0}^{n} \alpha_i \left(\overline{\varphi}_i, \overline{\varphi}_j\right) \tag{8.3.7}$$

Подставляя (8.3.5) и (8.3.7) в (8.3.3), получаем СЛАУ:

$$\sum_{i=0}^{n} \alpha_i \left(\overline{\varphi}_i, \overline{\varphi}_j\right) = \left(\bar{f}, \overline{\varphi}_j\right), \ j = 0,..,n \tag{8.3.8}$$

В матричном виде:

$$\begin{pmatrix} \left(\overline{\varphi}_0, \overline{\varphi}_0\right) & \left(\overline{\varphi}_0, \overline{\varphi}_1\right) & . & . \\ \left(\overline{\varphi}_0, \overline{\varphi}_1\right) & \left(\overline{\varphi}_1, \overline{\varphi}_1\right) & & \\ . & & . & \\ . & & & . \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ . \\ . \end{pmatrix} = \begin{pmatrix} \left(\bar{f}, \overline{\varphi}_0\right) \\ \left(\bar{f}, \overline{\varphi}_1\right) \\ . \\ . \end{pmatrix}$$

$$\hat{\Gamma}\overline{\alpha} = \overline{F}$$

$$\overline{\alpha} = \hat{\Gamma}^{-1}\overline{F} \tag{8.3.9}$$

Матрица $\hat{\Gamma}$ называется матрицей Грама.

В итоге задача аппроксимации для полиномов сводится к решению СЛАУ или к нахождению обратной матрицы и ее умножению на вектор.

Частным случаем обобщенного многочлена (8.3.1) являются полиномы:

$$\varphi_k(x) = x^k \tag{8.3.10}$$

В предельном переходе $N \to \infty$ матрица $\hat{\Gamma}$ для полиномов равна:

$$\hat{\Gamma} = \begin{pmatrix} 1 & 1/2 & 1/3 & . \\ 1/2 & 1/3 & 1/4 & \\ 1/3 & 1/4 & . & \\ . & & & . \end{pmatrix} \tag{8.3.11}$$

Эта частная матрица носит название матрицы Гильберта.

При этом ($N \to \infty$) вектор $\overline{F}$ равен:

$$\overline{F} = \begin{pmatrix} \int_0^1 x^0 f(x)dx \\ \int_0^1 x^1 f(x)dx \\ . \\ . \end{pmatrix} \tag{8.3.12}$$

Задание 8.3.1

Реализовать алгоритм полиномиального метода наименьших квадратов для задачи аппроксимации полиномом степени до 6 включительно функции e^{-x} на отрезке $x \in [0,1]$.

9. Быстрое преобразование Фурье

Рассмотрим сначала дискретное преобразование Фурье (ДПФ), происходящее из непрерывного преобразования Фурье функции $f(t)$ на конечном отрезке $t \in [0,1]$:

$$F(\omega_n) = \int_0^1 f(t) e^{i\omega_n t} dt\,,\ \omega = 2\pi n\,,\ n = 0,..,\infty \qquad (9.1)$$

Этот интеграл представляет собой ортогональное преобразование в бесконечномерном (гильбертовом) пространстве. От разложения вектора в ортогональном базисе привычного 2- или 3-мерного евклидового пространства преобразование Фурье отличается лишь числом измерений и комплексным характером функций. Интеграл соответствует скалярному произведению в гильбертовом пространстве и описывает проецирование вектора $\bar{f}$ на n-ю орту $\bar{e}_n = e^{i\omega_n t}$, т.е.

$$F(\omega_n) = (\bar{f}, \bar{e}_n) \qquad (9.2)$$

Сами же орты удовлетворяют условию

$$(\bar{e}_m, \bar{e}_n) = \int_0^1 e^{-i\omega_m t} e^{i\omega_n t} dt = \delta_{mn} \qquad (9.3)$$

δ_{mn} - символ Кронекера.

Здесь учитывается комплексный характер базисных функций, из-за которого длина вектора определяется как скалярное произведение вектора на свое комплексное сопряжение.

Заметим, что преобразование Фурье может быть формально получено через метод наименьших квадратов с линейно входящими параметрами (предыдущая глава), причем матрица Грама будет диагональной.

При численном нахождении (9.1) бесконечное множество точек $t \in [0,1]$ заменяется дискретным $\{t_k\}$, $k = 0,..,N-1$, $t_k = k/N$. Тогда вместо пространства Гильберта получаем обычное N-мерное евклидовое пространство, а формулы (9.1), (9.2) приобретают вид:

$$F(\omega_n) = \frac{1}{N}\sum_{k=0}^{N-1} f(t_k) e^{i\omega_n t_k} = \frac{1}{N}\sum_{k=0}^{N-1} f(t_k) e^{2\pi i \frac{nk}{N}} \qquad (9.4)$$

При этом свойство ортогональности базиса сохраняется (без доказательства):

$$\frac{1}{N}\sum_{k=0}^{N-1} e^{-2\pi i\frac{mk}{N}} e^{2\pi i\frac{nk}{N}} = \delta_{mn} \qquad (9.5)$$

Число базисных векторов, которые полностью описывает вектор $\bar{f}$, будет равно размерности пространства N.

В итоге получаем формулы дискретного преобразования Фурье (ДПФ):

$$F_n = \frac{1}{N}\sum_{k=0}^{N-1} f_k e^{2\pi i\frac{nk}{N}}, \; n = 0,..,N-1 \qquad (9.6)$$

Соответствующие формулы обратного ДПФ:

$$f_n = \frac{1}{N}\sum_{k=0}^{N-1} F_k e^{-2\pi i\frac{nk}{N}}, \; n = 0,..,N-1 \qquad (9.7)$$

Здесь $f_k = f(x_k),\; F_k = F(x_k)$

Эти формулы могут быть представлены в матричном виде:

$$\overline{F} = \hat{A}\bar{f}$$

$$\bar{f} = \hat{A}^{-1}\overline{F} \qquad (9.8)$$

Для получения ДПФ требуется $O(N^2)$ операций, столько же, сколько для операции умножения матрицы на вектор.

Задание 9.1

Реализовать ДПФ для функции $f(t) = e^{-0.1t}\sin(\alpha t) + 2\cos(9t)$, $t \in [0,1]$, где α пробегает значения от 0 до 1000 с шагом 1. Число точек $N = 1024$. Результаты представить в виде анимации амплитудного спектра, т.е. ДПФ выполняется для α, выводится на экран график амплитудного спектра (амплитуда – модуль комплексного вектора в каждой точке Фурье-образа), затем α увеличивается на 0.0001 и процесс повторяется. По достижению $\alpha = 1000$ параметр α сбрасывается в 0. Измерить время одного цикла анимации, при котором α пробегает значения от нуля до единицы.

Идея быстрого преобразования Фурье состоит в том, что при N не являющимся простым числом, размерность задачи может быть уменьшена. Предположим, что N четное. Тогда сумму (9.6) можно разбить на две, содержащие четные и нечетные узлы.

$$F_n = \frac{1}{N}\sum_{k=0}^{N/2-1} f_{2k} e^{2\pi i\frac{n(2k)}{N}} + \frac{1}{N}\sum_{k=0}^{N/2-1} f_{2k+1} e^{2\pi i\frac{n(2k+1)}{N}}, \; n = 0,..,N-1 \qquad (9.9)$$

После небольших преобразований:

$$F_n = \frac{1}{2}\left(\frac{1}{N/2} \sum_{k=0}^{N/2-1} f_{2k} e^{2\pi i \frac{nk}{N/2}} \right) + \frac{e^{2\pi i \frac{n}{N}}}{2} \left(\frac{1}{N/2} \sum_{k=0}^{N/2-1} f_{2k+1} e^{2\pi i \frac{nk}{N/2}} \right) \qquad (9.10)$$

Каждая из сумм теперь соответствует самостоятельному ДПФ с вдвое меньшим числом точек. Общее число арифметических операций при таком представлении уменьшилось примерно вдвое.

Если $N/2$ тоже четное число, то операцию разбиения каждой суммы на две можно повторить. Наиболее выгодным для практической реализации будет случай $N = 2^p$, обычно именно он и подразумевается, когда говорят о быстром преобразовании Фурье (БПФ). Рассмотрим практическую реализацию БПФ для $N = 16$. Посмотрим как будут группироваться точки при каждом разбиении $\{f_k\}$ на четные и нечетные:

Таблица 1. Группировка узлов для быстрого преобразования Фурье.

f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
f_0	f_2	f_4	f_6	f_8	f_{10}	f_{12}	f_{14}	f_1	f_3	f_5	f_7	f_9	f_{11}	f_{13}	f_{15}
f_0	f_4	f_8	f_{12}	f_2	f_6	f_{10}	f_{14}	f_1	f_5	f_9	f_{13}	f_3	f_7	f_{11}	f_{15}
f_0	f_8	f_4	f_{12}	f_2	f_{10}	f_6	f_{14}	f_1	f_9	f_5	f_{13}	f_3	f_{11}	f_7	f_{15}

Введем r - новый индекс точки, который нумерует слева направо точки из нижней строки таблицы. Последовательность $r = 0,1,2,3,4,...$ будет соответствовать последовательности $k = 0,8,4,12,2,10,...$. Заметим, что в бинарном представлении числа r и k будут «зеркальными», т.е. одно получается из другого перестановкой битов в обратном порядке.

В итоге ДПФ по $N = 16$ точкам сводится к сумме восьми ДПФ по $N = 2$ точкам. Двухточечное ДПФ:

$$F_n = \frac{1}{2} \sum_{k=0}^{1} f_k e^{2\pi i \frac{nk}{2}} = \frac{1}{2} \left(f_0 + (-1)^n f_1 \right), \; n = 0,1 \qquad (9.11)$$

Или

$$F_0 = \frac{1}{2}(f_0 + f_1), \; F_1 = \frac{1}{2}(f_0 - f_1) \qquad (9.12)$$

Приведем один из возможных вариантов реализации алгоритма БПФ для числа точек, являющегося степенью двойки. Алгоритм предполагает, что операции с комплексными числами уже реализованы.

1) Выборка из $N = 2^q$ дискретных значений подвергается пересортировке, как показано в Таблице 1. Значения записываются в массив A, всего N элементов в массиве. Массив A должен быть предназначен для хранения комплексных чисел. Диапазон индексов массива $0,..,N-1$.

2) Вводится целочисленная переменная m, пробегающая значения от 1 до q. Начальное значение $m = 1$.

3) Производится обработка элементов массива по следующей схеме:

$$A[i] = \frac{1}{2}\left(A[i_0] + \exp\left(\frac{2\pi i}{N}\left(i - \left(i \setminus 2^m\right)\cdot 2^m\right)\cdot 2^{q-m} \right) A[i_1] \right) \qquad (9.13)$$

i_0, i_1 получаются заменой $(m-1)$-го бита в двоичной записи числа i на 0 и 1, соответственно.

$i \setminus 2^m$ означает целочисленное деление, когда вещественная часть результата деления отбрасывается. Выражение $i - \left(i \setminus 2^m\right)\cdot 2^m$ представляет остаток от деления i на 2^m, что практически в любом языке программирования записывается как одна операция. Отметим также, что операции умножения и деления на числа, являющие степенью двойки, в двоичной системе удобно представлять через сдвиги битов числа влево и вправо, соответственно.

Хотя выражение (9.13) описывает операцию над одним массивом, удобно вводить вспомогательный массив, в котором хранится результат операций над A. По окончанию обработки всего массива A, в него переписывается вспомогательный массив.

4) Увеличивается на единицу m. Если $m > q$, то основная работа алгоритма закончена, массив A содержит Фурье-образ исходной выборки, умноженной на N. В противном случае выполняется переход на п.3.

5) Выполняется деление всех элементов на N.

Нулевой элемент массива A по окончанию описанной процедуры БПФ соответствует частотной компоненте с нулевой частотой (постоянная составляющая), $N-1$ - максимальной частоте, равной частоте дискретизации.

Задание 9.2

Реализовать БПФ для функции $f(t)=e^{-0.1t}\sin(\alpha t)+2\cos(9t)$, $t\in[0,1]$, где α пробегает значения от 0 до 1000 с шагом 1. Число точек $N=1024$. Экспоненциальные множители предварительно вычислить и использовать в виде массива. Результаты представить в виде анимации амплитудного спектра, т.е. БПФ выполняется для α, выводится на экран график амплитудного спектра (амплитуда – модуль комплексного вектора в каждой точке Фурье-образа), затем α увеличивается на 1 и процесс повторяется. По достижению $\alpha=1000$ параметр α сбрасывается в 0. Измерить время одного цикла анимации, при котором α пробегает значения от нуля до единицы. Сравнить со случаем ДПФ.

При реализации БПФ и ДПФ можно было видеть, что в амплитудном спектре появляются «зеркальные» частоты, симметрично расположенные относительно половинной частоты дискретизации. Объясняется это дискретностью входного сигнала, дискретизация «зеркальных» компонент приведет к одной и той же последовательности точек, т.е. они принципиально неотличимы.

Приведем здесь без подробного обсуждения теорему Найквиста-Котельникова, проявление которой наблюдалось в заданиях 9.1 и 9.2. Эта теорема гласит следующее. Если $f(t)$ имеет ограниченный спектр, то этот спектр может быть однозначным образом восстановлен по дискретному аналогу функции, если частота дискретизации вдвое выше верхней граничной частоты спектра сигнала $f(t)$.

10. Решение дифференциальных уравнений в частных производных

Уравнения в частных производных охватывают широчайший круг практических задач, возникающих в аэро- и гидродинамике, сопротивлении материалов, физике твердого тела, квантовой механике и т.д.

В рамках настоящего курса эта часть численных методов представлена очень кратко, на примере явного и неявного конечно-разностных методов решения одномерного уравнения теплопроводности без источников тепла и учета фазовых переходов, с упрощенными граничными условиями.

10.1. Явная и неявная конечно-разностные схемы для уравнения теплопроводности

Запишем уравнение теплопроводности (которое также является уравнением диффузии):

$$\frac{\partial T(x,t)}{\partial t} = A\frac{\partial^2 T(x,t)}{\partial x^2}$$

$$T(x,0) = T_0(x)$$

$$x \in [0, X] \qquad (10.1)$$

Здесь A представляет собой единственную числовую величину, которая определяет эволюцию температурного профиля. В физических задачах A равняется коэффициенту теплопроводности, деленному на произведение теплоемкости и плотности.

Граничные условия будем использовать упрощенные, «нефизичные».

$$T(0,t) = T(X,t) = T_b = const \qquad (10.2)$$

Наша цель – показать ключевые этапы в построении численного метода решения уравнения теплопроводности, граничные условия здесь не играют роли.

Самый простой способ численного решения (10.1) – конечно-разностная аппроксимация производных. Для этого введем двумерную сетку на области

решения, с шагами h и τ по координатам x и t, соответственно. Узлы сетки по оси x будем нумеровать числами $i = 0,..,N$, а по оси t - числами $j = 0,..,\infty$. Тогда:

$$\frac{\partial T(x,t)}{\partial t} = \frac{T_i^{j+1} - T_i^j}{\tau} + O(\tau)$$

$$\frac{\partial^2 T(x,t)}{\partial x^2} = \frac{T_{i+1}^j - 2T_i^j + T_{i-1}^j}{h^2} + O(h^2) \qquad (10.3)$$

Конечно-разностный аналог исходного уравнения (10.1):

$$\frac{T_i^{j+1} - T_i^j}{\tau} = A\frac{T_{i+1}^j - 2T_i^j + T_{i-1}^j}{h^2} \qquad (10.4)$$

Или:

$$T_i^{j+1} = T_i^j + \lambda\left(T_{i+1}^j - 2T_i^j + T_{i-1}^j\right)$$

$$\lambda = A\frac{\tau}{h^2} \qquad (10.5)$$

Это и есть формула явного метода решения уравнения теплопроводности. По известному температурному профилю рассчитывается температурный профиль в следующий момент времени. Этот процесс можно повторять сколь угодно долго, однако необходимо учитывать, что на каждом применении формулы (10.5) вносится погрешность, связанная с конечно-разностной аппроксимацией производных и равная $O(h^2)+O(\tau)$, или $O(h^2+\tau)$.

Задание 10.1

Решить с помощью явного конечно-разностного метода уравнение теплопроводности для длинного железного стержня с числом узлов по координатной оси 500. В качестве начального профиля взять такое распределение температур, при котором в одном узле сетки, примерно посередине стержня температура равна 400К, тогда как в остальных узлах она равна 300К. Определить в численном эксперименте максимальное λ, при котором явный метод остается устойчивым.

Итак, явный метод не всегда устойчив или, как говорят, *условно устойчив*.

Исходное уравнение (10.1) можно аппроксимировать иначе:

$$T_i^{j+1} = T_i^j + \lambda\left(T_{i+1}^{j+1} - 2T_i^{j+1} + T_{i-1}^{j+1}\right)$$

$$\lambda = A\frac{\tau}{h^2} \qquad (10.6)$$

На первый взгляд, такой подход выглядит нерациональным, потому что T^{j+1} не выражается явно через T^j. Это и есть неявная конечно-разностная схема для уравнения теплопроводности, с тем же порядком аппроксимации $O(h^2+\tau)$, что и у явной. Для нахождения следующего температурного профиля придется решать систему линейных уравнений:

$$-\lambda T_{i+1}^{j+1}+(1+2\lambda)T_i^{j+1}-\lambda T_{i-1}^{j+1}=T_i^j,\ i=1,..,N-1$$

$$T_0^{j+1}=T_N^{j+1}=T_0^j=T_N^j=T_b \qquad (10.7)$$

Или в матричном виде:

$$\begin{pmatrix} 1 & 0 & 0 & . \\ -\lambda & 1+2\lambda & -\lambda & . \\ 0 & -\lambda & 1+2\lambda & -\lambda \\ . & . & . & . \end{pmatrix}\begin{pmatrix} T_0^{j+1} \\ T_1^{j+1} \\ T_2^{j+1} \\ . \end{pmatrix}=\begin{pmatrix} T_0^j \\ T_1^j \\ T_2^j \\ . \end{pmatrix}$$

$$\hat{Y}\overline{T}^{j+1}=\overline{T}^j$$

$$\overline{T}^{j+1}=\hat{Y}^{-1}\overline{T}^j \qquad (10.8)$$

То есть необходимо до решения задачи один раз (в нашем случае) найти матрицу, обратную к $\hat{Y}$. Матрица $\hat{Y}$ имеет частный вид, она называется трехдиагональной. Для оборачивания таких, частного вида матриц, которые также возникают в задачах интерполяции функций, существует *метод прогонки*. Фактически, это алгоритм Гаусса, в котором исключена большая часть заведомо ненужных операций и сложность алгоритма сведена к $O(N+1)$. Кроме этого, такое сокращение числа операций для матриц большого размера значительно уменьшает эффект накопления вычислительной погрешности.

Рассмотрим метод прогонки применительно к разностной схеме (10.6), (10.7).

Сделаем подстановку:

$$T_{i-1}^{j+1}=c_{i-1}T_i^{j+1}+\varphi_{i-1},\ i=1..N$$

$$T_N=T_b \qquad (10.9)$$

Тогда (10.7) приобретает вид:

$$-\lambda T_{i+1}^{j+1}+(1+2\lambda-\lambda c_{i-1})T_i^{j+1}=T_i^j+\lambda\varphi_{i-1} \qquad (10.10)$$

Или:

$$T_i^{j+1}=\frac{\lambda}{1+2\lambda-\lambda c_{i-1}}T_{i+1}^{j+1}+\frac{T_i^j+\lambda\varphi_{i-1}}{1+2\lambda-\lambda c_{i-1}} \qquad (10.11)$$

Но из исходной подстановки:

$$T_i^{j+1} = c_i T_{i+1}^{j+1} + \varphi_i \qquad (10.12)$$

Откуда, получаем рекуррентные соотношения:

$$c_i = \frac{\lambda}{1+2\lambda-\lambda c_{i-1}},\ i=1..N$$

$$\varphi_i = \frac{T_i^j + \lambda\varphi_{i-1}}{1+2\lambda-\lambda c_{i-1}},\ i=1..N$$

$$c_0 = 0,\ \varphi_0 = T_b \qquad (10.13)$$

В результате решение СЛАУ (10.7) свелось к двум циклам. В первом находятся коэффициенты линейной подстановки, при этом двигаемся от элементов с меньшими номерами к элементам с большими номерами. Во втором цикле, двигаясь в обратном порядке, находим решение СЛАУ. Из-за такого движения вверх и вниз по индексам метод прогонки и получил свое название.

Задание 10.2

Решить с помощью неявного конечно-разностного метода уравнение теплопроводности для длинного железного стержня с числом узлов по координатной оси 500. Использовать метод прогонки. В качестве начального профиля взять такое распределение температур, при котором в одном узле сетки, примерно посередине стержня температура равна 400К, тогда как в остальных узлах она равна 300К. Определить в численном эксперименте максимальное λ, при котором неявный метод остается устойчивым.

При правильном решении Задания 10.2 должен получиться метод, устойчивый для любых λ или *безусловно устойчивый* метод. В этом и заключается смысл использования неявной схемы, других преимуществ перед явной схемой она не имеет.

10.2. Спектральный метод анализа устойчивости

Полученные в численном эксперименте результаты по устойчивости явной и неявной конечно-разностной схемы уравнения теплопроводности могут быть

получены также аналитическим путем. Рассмотрим метод, применение которого выходит далеко за рамки частной численной задачи об уравнении теплопроводности – спектральный метод (СМ) анализа устойчивости.

Идея СМ очень проста. Любая функция, являющаяся решением (10.1) может быть разложена в ряд Фурье. Если конечно-разностная схема устойчива, то она должна быть устойчива по отношению к каждой компоненте ряда Фурье. При этом действие конечной разности (10.5), (10.6) на одну компоненту ряда Фурье не меняет её вид, а лишь сводится к умножению на число, равно как и действие оператора второй производной в (10.1). Рассмотрим подробнее, сначала для явной схемы (10.5):

$$T_i^j = e^{i\omega x_i},\ T_i^{j+1} = Ce^{i\omega x_i} \tag{10.2.1}$$

Подставим (10.2.1) в (10.5):

$$\begin{aligned} &Ce^{i\omega x_i} = e^{i\omega x_i} + \lambda\left(e^{i\omega x_{i+1}} - 2e^{i\omega x_i} + e^{i\omega x_{i-1}}\right) = e^{i\omega x_i} + \lambda\left(e^{i\omega(x_i+h)} - 2e^{i\omega x_i} + e^{i\omega(x_i-h)}\right) = \\ &= e^{i\omega x_i} + \lambda e^{i\omega x_i}\left(e^{i\omega h} - 2 + e^{-i\omega h}\right) \end{aligned} \tag{10.2.2}$$

Сокращая $e^{i\omega x_i}$, получаем C:

$$C = 1 + \lambda\left(e^{i\omega h} - 2 + e^{-i\omega h}\right) = 1 + \lambda\left(2\cos\omega h - 2\right) = 1 - 4\lambda\sin^2\frac{\omega h}{2} \tag{10.2.3}$$

Чтобы разностная схема была устойчивой, должно выполняться $|C| \le 1$ для любого ω. В противном случае вес соответствующей компоненты будет возрастать экспоненциально с числом итераций, а само решение станет неустойчивым.

Тогда:

$$-1 \le \left(1 - 4\lambda\sin^2\frac{\omega h}{2}\right) \le 1 \tag{10.2.4}$$

Или:

$$\begin{cases} 4\lambda\sin^2\dfrac{\omega h}{2} \ge 0 \\ 4\lambda\sin^2\dfrac{\omega h}{2} \le 2 \end{cases} \tag{10.2.5}$$

Первое неравенство в (10.2.5) выполняется всегда. Второе будет выполняться для любого ω только в случае $\lambda \le 1/2$. Это и есть условие устойчивости для явной конечноразностной схемы уравнения теплопроводности, полученное нами в численном эксперименте при выполнении задания из предыдущей главы.

Для неявной схемы получим похожее на (10.2.3) уравнение:

$$C = 1 + C\lambda\left(e^{i\omega h} - 2 + e^{-i\omega h}\right) = 1 + C\lambda\left(2\cos\omega h - 2\right) = 1 - 4C\lambda\sin^2\frac{\omega h}{2} \qquad (10.2.6)$$

Решая относительно C:

$$C = \frac{1}{1 + 4\lambda\sin^2\frac{\omega h}{2}} \qquad (10.2.7)$$

Здесь, поскольку знаменатель (10.2.7) всегда больше единицы, условие устойчивости $|C| \le 1$ будет выполняться для любых ω и λ.

Заключение

Конечные формулы, описывающие тот или иной численный метод, зачастую выглядят странно, не вызывая ощущения их связанности с исходной задачей. У непосвященного человека они могут вызвать недоумение, непонимание и реакцию «Зачем решать задачу так сложно? Я сделаю по-своему, намного проще и очевиднее». И решение несложной задачи, например, взятие определенного интеграла, которое надлежащим выбором метода вычислений делается за доли секунды, растягивается на часы из-за метода левых прямоугольников. Если Вы ощутили на практике магию конечного результата и простоту построения методов вычислений, подобных ошибок у Вас не будет ни при самостоятельной реализации численных методов, ни при использовании готовых решений в виде математических пакетов. В этом случае цель настоящего пособия достигнута.

Литература

1. Вержбицкий В.М. Основы численных методов: Учебник для ВУЗов. – М.:Высш.шк., 2002. – 840с.
2. Самарский А.А. Введение в численные методы – М.: Наука, 1997. – 234 с
3. Бахвалов Н.С. Жидков Н.П. Кобельков Г.М. Численные методы. – М: Лаборатория Базовых Знаний, 2002. – 632 с
4. Самарский А.А., Гулин А.В. Численные методы. – М.: Наука, 1989. – 430 с.
5. Демидович Б.П., Марон И.А. Основы вычислительной математики. – М.: ГИФМЛ, 1960. – 659 с.
6. Hairer E. A Runge-Kutta Method of Order 10 J.Inst.Math.Applic. 1978. V. 21. P. 47 – 59.