Министерство образования Российской Федерации
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПРОГРАММИРОВАНИЕ ВЫЧИСЛЕНИЙ

Методические указания к выполнению лабораторных работ
по курсу «Программирование вычислений»
для студентов 2 курса ФПМИ

Новосибирск
2006

Составители: канд.техн.наук, доц. *М.Э.Рояк*
канд.техн.наук, доц. *С.Х.Рояк*
канд.техн.наук, доц. *М.Г.Токарева*
канд.техн.наук, доц. *М.Г.Персова*

Рецензент: канд.техн.наук, доц. А.В. Чернышев

Работа подготовлена на кафедре прикладной математики НГТУ

ЛАБОРАТОРНАЯ РАБОТА № 1

Цель работы

Ознакомление с основными операторами языка ФОРТРАН, отслеживание возможностей переполнения и исчезновения порядка при вычислениях, знакомство с погрешностью вычислений.

Теоретическая часть

Одно из основных понятий вычислительной математики - понятие вычислительной погрешности. Если неправильно построить вычислительный процесс, то погрешность может появиться даже в тех задачах, которые имеют аналитическое решение. Однако большинство практических задач не имеют аналитического решения, и тогда возникает необходимость решать задачу численно с заданной точностью, не превышая определенной погрешности. При этом в численном анализе мы сталкиваемся с необходимостью работы с приближенными числами. Поэтому, чтобы правильно построить вычислительный процесс, уменьшить вычислительную погрешность и получить решение задачи с заданной точностью, необходимо знать источники погрешностей, правила работы с приближенными числами, правила сравнения чисел, правила распространения погрешности.

При работе с приближенными величинами важно уметь:

1) давать математические характеристики точности приближенных величин;

2) зная степень точности исходных данных, оценить степень точности результатов и требуемую точность промежуточных вычислений;

3) правильно построить вычислительный процесс, чтобы избавить его от тех выкладок, которые не окажут влияния на точные цифры результата.

Источники погрешностей

Погрешность результата решения задачи складывается из трех составных частей:

– неустранимой погрешности решения, обусловленной неточностью исходных данных;

– погрешности метода решения задачи;

– вычислительной погрешности, являющейся результатом округлений в процессе счета.

Неустранимая погрешность решения обусловлена неточностью исходных данных, которые возникают в результате неточности измерений (инструментальная ошибка) или из-за невозможности представить необходимую величину конечным числом значащих цифр (ошибка округления). Инструментальная ошибка всегда возникает при проведении физического измерения, поскольку оно не может быть выполнено абсолютно точно. Ошибки округления при задании исходных данных возникают в том случае, когда величину невозможно представить ограниченным числом значащих цифр, например число π, e.

Погрешности метода решения задач очень часто возникают при использовании численных методов. Действительно, многие математические задачи могут быть решены только приближенно, хотя и со сколь угодно большой точностью, так как любой численный метод предполагает использование конечного числа арифметических операций. Например, при решении задачи обычно производную заменяют разностью, интеграл – суммой конечного числа членов ряда, или бесконечный итерационный процесс обрывают после некоторого конечного числа итераций. Так же к погрешностям метода можно отнести неточность отображения реальных процессов, так как рассматривается не сам процесс, а его идеализированная математическая модель.

При решении численных задач на компьютере всегда возникают вычислительные погрешности, обусловленные ошибками округления в процессе счета (так как вычисления на ЭВМ выполняются с конечным числом значащих цифр, определенных конечностью разрядной сетки ЭВМ). Исключения составляют задачи, в которых операции над данными выполняются точно, например, в целочисленной арифметике. Однако в подавляющем большинстве вычислительных задач используются вещественные числа, операции над которыми выполняются с ошибками округления. В зависимости от реализованного в алгоритме метода решения, эти ошибки округления могут либо расти, либо уменьшаться.

При расчетах те или иные погрешности могут отсутствовать или их влияние может быть мало.

Абсолютная и относительная погрешности

Абсолютная погрешность (ошибка) Δ_X - это разница между истинным значением величины $\bar{X}$ (считая это истинное значение известным) и ее приближенным значением X.

Относительная погрешность (ошибка) $\delta_X = \frac{\Delta_X}{X}$ определяется, как отношение абсолютной погрешности к приближенному значению величины. Казалось бы, что более естественно определить ее, как отношение абсолютной погрешности к точному значению, но обычно точное значение нам неизвестно. Все, что обычно бывает известно, - это приближенное значение величины и *оценка* ошибки или *границы* максимально возможной величины ошибки. Если абсолютная погрешность мала, то разница в определениях не скажется на численной величине относительной погрешности.

Для величин, близких по значению к единице, абсолютная и относительная погрешности почти одинаковы. Для очень больших или очень малых величин относительная и абсолютная погрешности представляются совершенно разными числами.

Если из условия задачи или из контекста не ясно, какая ошибка имеется ввиду, то чаще считают, что ошибка относительная.

Порядок выполнения лабораторной работы

1. Написать программу, реализующую на языке ФОРТРАН вычисление заданного набора геометрических характеристик треугольников с использованием меню. Меню должно содержать следующие пункты:

1. Ввод нового треугольника.

2. Вычисление площади треугольника.
3. Вычисление минимального угла в градусах.
4. Вычисление косинуса минимального угла.
5. Окончание работы.

2. Написать программу с учетом следующих требований к реализации:

- программа должна быть модульной, каждое действие должно быть реализовано в виде подпрограммы;
- параметры треугольника хранить в общих блоках, остальные - по желанию;
- вещественные числа должны храниться с одинарной точностью, если в варианте не указано другое;
- вычисления должны производиться оптимальным образом;
- все вводимые данные должны проверяться на корректность;
- все углы должны измеряться в градусах (возможно, с дробной частью).

3. Протестировать разработанную программу.

4. Для разработанной программы провести исследования, позволяющие отследить: при каких входных данных возникает ситуация переполнения.

Для этого необходимо провести серию тестов, в которых для заданного треугольника (в каждом варианте согласно таблице 1 использовать свой тип треугольника: равнобедренный, равносторонний или прямоугольный) вычислить его площадь. При этом длина указанной стороны a треугольника должна увеличиваться, принимая следующие значения: 10, 10^2, 10^3, … Увеличение длины стороны выполнять до тех, пока при вычислениях не возникнет особая ситуация – ситуация переполнения. Внести в отчёт длину стороны a, при которой возникла особая ситуация. Отметить, как данная ситуация отображается на экране. Дать объяснение, почему при каких-то входных данных возникает ситуация переполнения.

Выполнить аналогичную серию тестов, уменьшая длину стороны a: 10^{-1}, 10^{-2}, 10^{-3}, …

5. Провести исследования, позволяющие отследить, при каких входных данных возникает исчезновение порядка при вычислениях.

В качестве исходного треугольника взять либо прямоугольный треугольник (рис. 1, 2), либо равнобедренный треугольник (рис. 3) (тип треугольника для каждого варианта указан в таблице 1). Значение длины стороны a изменять в диапазоне: $10^8 \div 10^{15}$.

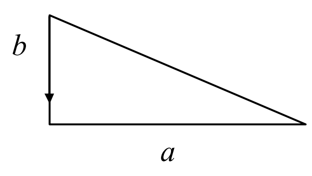


Рис. 1

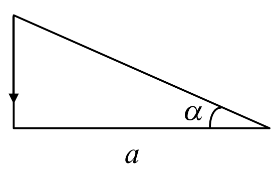


Рис. 2

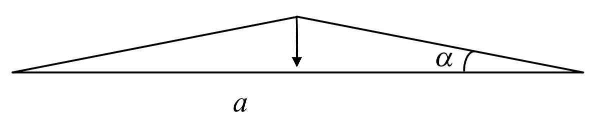


Рис. 3

Далее провести следующие действия:

Для прямоугольного треугольника (рис.1), зафиксировав значение длины стороны a (например, 10^8), вычислить геометрические характеристики (площадь – S, минимальный угол – α, косинус минимального угла – $\cos\alpha$) нескольких треугольников при уменьшении длины стороны b от значения $b = a$ на порядок.

Например: $a = 10^8$, $b = 10^8$, $S - ?$, $\alpha - ?$, $\cos\alpha - ?$

$a = 10^8$, $b = 10^7$, $S - ?$, $\alpha - ?$, $\cos\alpha - ?$

$a = 10^8$, $b = 10^6$, $S - ?$, $\alpha - ?$, $\cos\alpha - ?$

и т.д.

Уменьшать значение стороны b до тех пор, пока не возникнет нестандартная ситуация (например, несогласованные вычисленные характеристики треугольника). Далее выполнить аналогичную серию тестов для следующего значения длины стороны a (например 10^9).

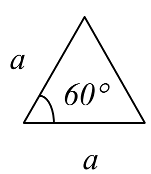
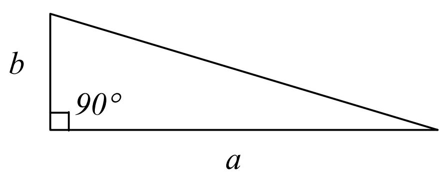
Для прямоугольного треугольника (рис.2) или равнобедренного треугольника (рис.3), зафиксировав сторону a (например 10^8), вычислить характеристики треугольника (площадь – S, косинус минимального угла – $\cos\alpha$) при уменьшении угла α. Уменьшать угол α до тех пор, пока не будет отмечена нестандартная ситуация (несогласованные вычисленные характеристики треугольника). Далее выполнить аналогичную серию тестов для следующего значения длины стороны a (например 10^9).

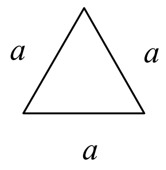
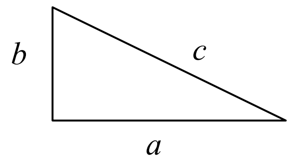
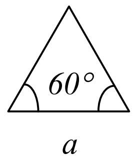
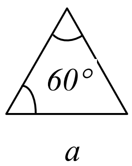
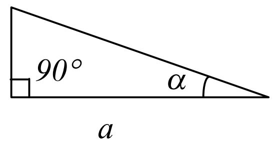
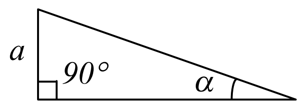
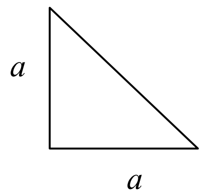
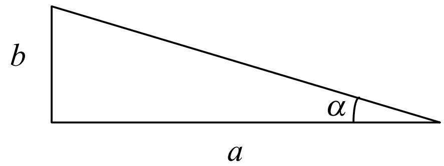
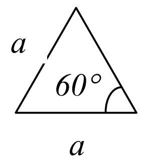
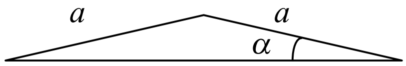
6. Для каждого проведенного исследования необходимо документировать результаты работы программы, возникающие ошибки, указать причину ошибки и методы ее устранения, сделать необходимые исправления в программе для того, чтобы программа выдавала корректные результаты.

7. Дополнительное исследование, позволяющее познакомиться с погрешностью вычислений, обусловленной неточностью исходных данных.

На примере точности задания числа π показать, как зависит погрешность результата решения задачи от неточности исходных данных. Провести серию тестов, иллюстрирующих данную ситуацию. Исследовать, изменяется ли результат вычисления в зависимости от количества значащих цифр в представлении числа π. Определить минимальное количество значащих цифр числа π, достаточное для решения задачи при хранении чисел с одинарной точностью.

Таблица 1

Вариант	1 исследование	2 исследование
1	Равносторонний треугольник 	Прямоугольный треугольник 

2	Равносторонний треугольник 	Прямоугольный треугольник  Организовать дополнительную процедуру вычисления с двойной точностью (real*8) третьей стороны *c* прямоугольного треугольника.
3, 4	Равносторонний треугольник Вариант 3 Вариант 4  	Прямоугольный треугольник Вариант 3 Вариант 4  
5, 6, 7	Равнобедренный прямоугольный треугольник 	Прямоугольный треугольник 
8, 9	Равносторонний треугольник 	Равнобедренный треугольник 

Варианты заданий

1. Треугольник задается двумя сторонами и углом между ними.
2. Треугольник задается тремя сторонами.
3. Треугольник задается стороной и двумя прилежащими углами.
4. Треугольник задается стороной, прилежащим и противолежащим углами.
5. Треугольник задается координатами вершин, использовать формулу Герона.
6. Треугольник задается координатами вершин, использовать формулу с определителем матрицы координат.

$$S_\Delta = \frac{1}{2}\begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{pmatrix}$$

7. Треугольник задается координатами вершин, использовать векторное произведение.
8. Треугольник задается двумя сторонами и углом между первой и оставшейся стороной.

9. Треугольник задается двумя сторонами и углом между второй и оставшейся стороной.

Контрольные вопросы

1. Основные операторы языка Fortran: выражения и операторы присваивания, операторы управления (оператор GOTO, конструкция IF, конструкция CASE), подпрограммы и функции (операторы SUBROUTINE и FUNCTION), операторы бесформатного ввода-вывода (операторы PRINT, READ, WRITE).
2. Типы данных. Явное описание (операторы INTEGER, REAL, LOGICAL, CHARACTER), неявное описание типов данных (описание по умолчанию, оператор IMPLICIT).
3. Именованные константы и константные выражения (оператор PARAMETER). Преимущества использования именованных констант.
4. Представление чисел в ЭВМ.
5. Переполнение и исчезновение порядка.
6. Понятие вычислительной погрешности, источники погрешностей.

ЛАБОРАТОРНАЯ РАБОТА № 2

Цель работы

Изучить операторы ввода-вывода и форматирования текста языка ФОРТРАН. Познакомиться со следующими понятиями вычислительной математики: конечная арифметика, значащие цифры, верные значащие цифры, абсолютная и относительная погрешность, накопление погрешности вычислений, округление.

Теоретическая часть

Десятичная запись приближенных чисел. Значащая цифра числа. Верная значащая цифра

Пусть приближенное число A задано в виде конечной позиционной записи:

$$A = \pm a_n a_{n-1} ... a_0, a_{-1} a_{-2} ... a_{-m}, \text{где } a_j - \text{десятичные цифры } \left(A = \sum_{j=-m}^{n} 10^j a_j \right). \quad (*)$$

Значащими цифрами числа называются все цифры в его десятичном изображении, отличные от нуля, и нули, если они содержатся между значащими цифрами или расположены в конце числа и указывают на сохранение разряда точности. Нули, стоящие левее первой отличной от нуля цифры, *не являются* значащими цифрами. (Например, числа 25.047 и 0.00250 имеют соответственно 5 и 3 значащих цифр.)

Любое число A может быть представлено в виде: $A = b \cdot q^p$, где p – некоторое целое число, называемое порядком числа A, q – основание системы счисления, b – число, называемое мантиссой числа A. Если выполнено условие $q^{-1} \le |b| < 1$, то говорят, что число A записано в нормализованном виде.

Пример. Рассмотрим представления числа $a = 2^{10}$.

Система счисления	Позиционная запись	Запись числа в нормализованном виде
Двоичная	10000000000	$.1*10^{1011}$
Десятичная	1024	$0.1024*10^4$

При решении задач очень часто ставится условие: вычислить результат с точностью до одной десятой, одной сотой и т.д. Создается впечатление, что точность вычислений определяется числом верных десятичных знаков после запятой. Это не так, число десятичных знаков зависит от единицы измерения. Остановимся на этом подробнее.

Цифра a_j в записи (*) называется *верной*, если абсолютная погрешность числа A не превосходит одной единицы соответствующего разряда десятичного числа.

Если приближенное число записывается без указания его предельной абсолютной погрешности, то выписываются только его верные цифры (знаки). При этом верные нули на правом конце числа не отбрасываются. Например, числа 0.0344 и 0.034400, как приближенные, различны. Относительно первого числа можно только утверждать, что его абсолютная погрешность не превосходит 10^{-4}, а из записи второго числа следует, что его абсолютная погрешность не больше чем 10^{-6}.

В том случае, когда у приближенного числа значащих цифр в целой части больше чем имеется верных знаков, то прибегают к записи в нормализованном виде. Например, $A = 0.390 \cdot 10^5$. Из этой записи понятно, что у числа A три верные значащие цифры. В данной ситуации запись вида $A = 39000$ недопустима. В нормализованном виде можно записать и рассмотренные выше приближенные числа: $0.344 \cdot 10^{-1}$, $0.34400 \cdot 10^{-1}$.

Можно говорить о числе верных значащих цифр у приближенного числа и о числе верных цифр после запятой. Как правило, при реальных вычислениях у приближенных чисел содержатся цифры после запятой, т.е. имеется дробная часть. Например, приближенное число $A = 25.030$ имеет 5 верных значащих цифр, и 3 верные цифры после запятой, а у числа $B = 0.00230$, наоборот, 3 верные значащие цифры и 5 верных цифр после запятой.

Вывод: Абсолютная погрешность приближенного числа характеризуется числом верных цифр данного приближенного числа, а относительная погрешность – числом верных значащих цифр.

Порядок выполнения лабораторной работы

1. Написать программу, реализующую на языке ФОРТРАН построение таблицы значений заданной функции двух переменных с учетом следующих требований:

- таблица должна содержать n+1 строк и m+1 столбцов, первая строка – значения y, первый столбец – значения x;
- определять n и m из заданных значений минимума и максимума каждого аргумента, а также шагов их изменения; независимо от того, делит шаг диапазон изменения нацело или нет, последним значением аргумента должен быть заданный максимум;
- строки и столбцы таблицы разделять приемлемыми символами (например, | и -);
- каждое представленное в таблице значение должно содержать ровно 4 значащих цифры мантиссы числа;
- результатом работы программы должен быть текстовый файл, содержащий таблицу;
- параметры вводить из файла с фиксированным именем, сообщения об ошибках выдавать на экран, в диалог не вступать; выдача таблицы на экран не обязательна;
- все вводимые данные должны проверятся на корректность, для любых данных программа должна давать корректный результат (сообщение об ошибке на русском языке – корректный результат, сообщение системы о делении на ноль - некорректный результат);
- все углы должны измеряться в градусах (возможно, с дробной частью).

2. Протестировать разработанную программу.

3. Провести исследования на наборах тестов, выданных преподавателем.

4. Для каждого проделанного исследования необходимо поместить в отчет фрагменты таблиц, отражающие особенности вычислительной математики, например, переход через особые точки функции, «нули» функции, «нули» аргументов, отобразить фрагменты таблицы, в которых замечено изменение значений функций при неизменных значениях аргументов x или y. Дать объяснения данным ситуациям, сделать необходимые исправления в программе для того, чтобы программа выдавала корректные результаты.

5. Поместить в отчет исправленные фрагменты программы, привести новые результаты всех исследований.

Варианты заданий

1. $\arcsin(x+y)$,

2. $\arccos\ (x+y)$,

3. $\ln(x+y)$,

4. $\sqrt{x+y}$,

5. $\dfrac{1}{x+y}$,

6. $\lg_{10}(x+y)$,

7. $\left|\mathrm{tg}\ (x+y)\right|$,

8. $\left|\mathrm{ctg}\ (x+y)\right|$,

9. $\left|\dfrac{\sin x}{\cos y}\right|$,

10. $\left|\dfrac{\cos x}{\sin y}\right|$,

11. $\left|\sec\ (x+y)\right|$,

12. $\left|\operatorname{cosec}\ (x+y)\right|$

13. $\operatorname{tg}(x+y)$,

14. $\operatorname{ctg}(x+y)$,

15. $\dfrac{\sin x}{\cos y}$,

16. $\dfrac{\cos x}{\sin y}$,

17. $\sec\ (x+y)$,

18. $\operatorname{cosec}\ (x+y)$,

19. $\dfrac{1}{\sin x+\cos y}$,

20. $\dfrac{\operatorname{tg} x}{\cos y}$,

21. $\dfrac{\operatorname{ctg} x}{\cos y}$,

22. $\dfrac{\sin y}{\cos x}$,

23. $\dfrac{\cos y}{\sin x}$

Контрольные вопросы

1. Организация циклов на Фортране (конструкция DO, число итераций конструкции DO, переменная цикла, значение переменной цикла, оператор DO WHILE, оператор EXIT, организация циклов с помощью условного логического оператора IF…THEN… и оператора безусловного перехода GOTO)
2. Форматы (задание формата в виде текстовой константы, с помощью оператора FORMAT). Дескрипторы формата данных (дескрипторы I, F, E, L, A, G), коэффициент повторения, дескриптор X, дескрипторы прямой и обратный слэш (/, \).
3. Форматный ввод-вывод с помощью операторов READ, PRINT, WRITE, спецификаторы операторов READ и WRITE.
4. Операторы открытия файла (OPEN), номер устройства, ввод из файла, вывод в файл, внутренний файл.
5. Понятие вычислительной погрешности.
6. Абсолютная и относительная погрешность.
7. Значащая цифра числа. Верная значащая цифра
8. Возникновение и распространение погрешностей вычислений.
9. Правила работы с приближенными числами
10. Вывод формул оценки абсолютной и относительной ошибок арифметических операций.
11. Сравнение вещественных чисел.

ЛАБОРАТОРНАЯ РАБОТА № 3

Цель работы

Изучение псевдодинамического распределения памяти на языке ФОРТРАН. Изучение форматов хранения матриц большой размерности. Оптимизация программ по точности, скорости, памяти. Изучение погрешности вычисления скалярного произведения. Изучение способов отладки. Изучение принципов формирования тестов для вычислительных программ. Изучение файлов прямого доступа.

Теоретическая часть

Ленточные форматы

Ленточные форматы хранения матриц используется, когда все ненулевые элементы матрицы расположены по диагоналям, плотно прилегающим к главной диагонали, то есть можно выделить ленту, состоящую из ненулевых элементов и имеющую определенную ширину – *m*. Схематично такая матрица изображена на рис. *4*.

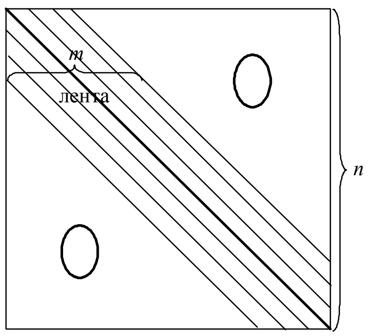


Рис. 4.

Существуют различные форматы хранения ленточных матриц. Рассмотрим некоторые из них.

Ленточный строчный формат. Для хранения исходной матрицы размерностью *n* и шириной ленты *m* используется прямоугольная матрица размером $n \times m$, которая в памяти хранится построчно. Для такого хранения необходимо доопределить нулями каждую побочную диагональ до общей размерности *n*, при этом для нижнего треугольника необходимое количество нулей записывается в начале диагоналей, а для верхнего треугольника - в конце диагоналей (см. рис. 5).

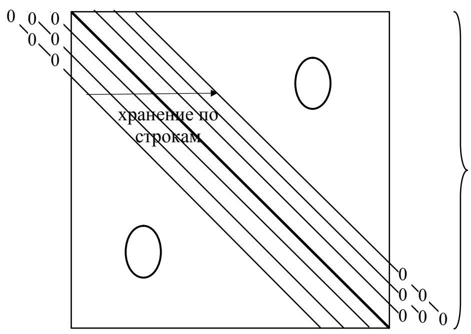


Рис. 5.

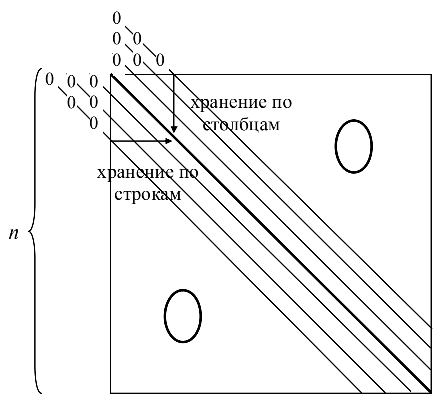


Рис.6.

Ленточный строчно-столбцовый формат. Для такого хранения ленточных матриц необходимо доопределить нулями побочные диагонали до общей размерности, записав необходимое количество нулей в начале диагоналей и для верхнего и для нижнего треугольника (см. рис. 6). Ленту возможно хранить в одном прямоугольном массиве размерностью $n \times m$ или в трех отдельных массивах: массив AL размерностью $n \times k$ (k-полуширина ленты) - для нижнего треугольника, массив AU размерностью $n \times k$ – для верхнего прямоугольника, массив Di – для хранения диагонали. При таком хранении ленточной матрицы (когда диагонали доопределяются нулями сверху) элементы нижнего треугольника хранятся по строкам, верхнего треугольника – по столбцам.

Если исходная матрица симметричная, то хранится построчно только её нижний треугольник (или постолбцово верхний, что в этом случае одно и то же). При этом формат хранения называют *симметричный ленточный строчный*.

<u>Пример</u>

Пусть дана ленточная матрица A размерностью 10×10, ширина ленты $m = 7$. Рассмотрим два способа хранения данной матрицы.

1 способ. Ленточный строчный формат.

A – исходная квадратная матрица, доопределенная нулями в начале диагоналей для нижнего треугольника, в конце диагоналей – для верхнего треугольника. A' – прямоугольная матрица размерностью 10×7, предназначенная для хранения исходной матрицы A.

$$A = \begin{array}{ccc} 0 & 0 & 0 \\ & 0 & 0 \\ & & 0 \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \left[\begin{array}{cccccccccc} \mathbf{1} & 12 & 13 & 14 & & & & & & \\ 2 & \mathbf{2} & 13 & 14 & 15 & & & & & \\ 3 & 3 & \mathbf{3} & 14 & 15 & 16 & & & & \\ 4 & 4 & 4 & \mathbf{4} & 15 & 16 & 17 & & & \\ & 5 & 5 & 5 & \mathbf{5} & 16 & 17 & 18 & & \\ & & 6 & 6 & 6 & \mathbf{6} & 17 & 18 & 19 & \\ & & & 7 & 7 & 7 & \mathbf{7} & 18 & 19 & 20 \\ & & & & 8 & 8 & 8 & \mathbf{8} & 19 & 20 \\ & & & & & 9 & 9 & 9 & \mathbf{9} & 20 \\ & & & & & & 10 & 10 & 10 & \mathbf{10} \end{array} \right] \begin{array}{ccc} \\ \\ \\ \\ \\ \\ \\ 0 & & \\ 0 & 0 & \\ 0 & 0 & 0 \end{array}, \quad A' = \left[\begin{array}{ccccccc} 0 & 0 & 0 & \mathbf{1} & 12 & 13 & 14 \\ 0 & 0 & 2 & \mathbf{2} & 13 & 14 & 15 \\ 0 & 3 & 3 & \mathbf{3} & 14 & 15 & 16 \\ 4 & 4 & 4 & \mathbf{4} & 15 & 16 & 17 \\ 5 & 5 & 5 & \mathbf{5} & 16 & 17 & 18 \\ 6 & 6 & 6 & \mathbf{6} & 17 & 18 & 19 \\ 7 & 7 & 7 & \mathbf{7} & 18 & 19 & 20 \\ 8 & 8 & 8 & \mathbf{8} & 19 & 20 & 0 \\ 9 & 9 & 9 & \mathbf{9} & 20 & 0 & 0 \\ 10 & 10 & 10 & \mathbf{10} & 0 & 0 & 0 \end{array} \right].$$

2 способ. Ленточный строчно-столбцовый формат.

A – исходная квадратная матрица, доопределенная нулями в начале диагоналей. AL и AU – прямоугольные матрицы размерностью 10×3, в которых хранятся нижний треугольник по строкам и верхний треугольник по столбцам соответственно. Di – вектор размерностью 10 хранит диагональные элементы.

$$A=\begin{array}{cc}
 & \begin{array}{cccccccccc}0&&&&&&&&&\\0&0&&&&&&&&\\0&0&0&&&&&&&\end{array}\\
\begin{array}{ccc}0&0&0\\&0&0\\&&0\\&&\\&&\\&&\\&&\\&&\\&&\\&&\end{array} & \begin{bmatrix}
\mathbf{1}&12&13&14&&&&&&\\
2&\mathbf{2}&13&14&15&&&&&\\
3&3&\mathbf{3}&14&15&16&&&&\\
4&4&4&\mathbf{4}&15&16&17&&&\\
&5&5&5&\mathbf{5}&16&17&18&&\\
&&6&6&6&\mathbf{6}&17&18&19&\\
&&&7&7&7&\mathbf{7}&18&19&20\\
&&&&8&8&8&\mathbf{8}&19&20\\
&&&&&9&9&9&\mathbf{9}&20\\
&&&&&&10&10&10&\mathbf{10}
\end{bmatrix}
\end{array},\quad
AAL=\begin{bmatrix}0&0&0\\0&0&2\\0&3&3\\4&4&4\\5&5&5\\6&6&6\\7&7&7\\8&8&8\\9&9&9\\10&10&10\end{bmatrix}\quad
AAU=\begin{bmatrix}0&0&0\\0&0&12\\0&13&13\\14&14&14\\15&15&15\\16&16&16\\17&17&17\\18&18&18\\19&19&19\\20&20&20\end{bmatrix}\quad
Di=\begin{bmatrix}1\\2\\3\\4\\5\\6\\7\\8\\9\\10\end{bmatrix}.$$

Диагональный формат

Диагональный формат хранения матриц используется, когда все ненулевые элементы матрицы расположены на различных диагоналях. Пример такой матрицы изображен на рис. 7. Индексы i_1, i_2, i_3, i_4, i_5 обозначают сдвиг диагоналей нижнего и верхнего треугольника от главной диагонали.

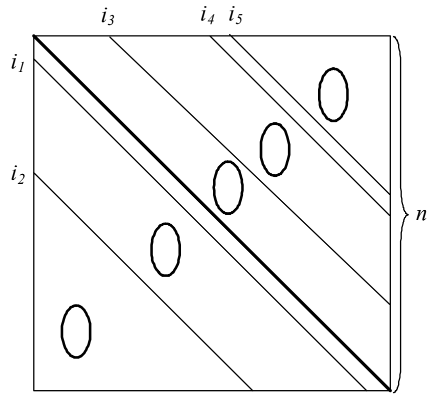
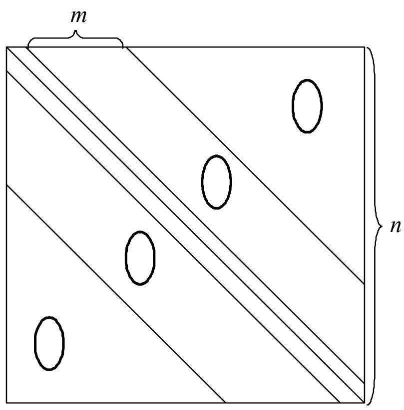


рис. 7.

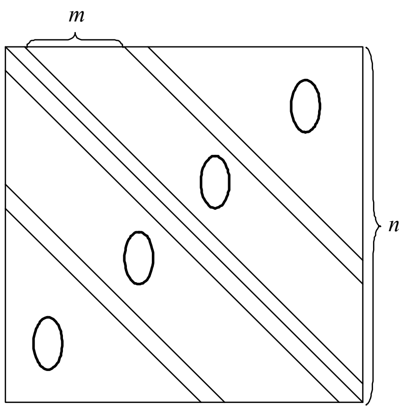
Хранение таких матриц аналогично хранению ленточных матриц, то есть для работы с матрицами диагонального вида используются прямоугольные матрицы размером $n \times m$, где n - размерность квадратной матрицы, m - количество ненулевых диагоналей. Прямоугольная матрица получается путем доопределения побочных диагоналей до общей размерностей нулями (либо в начале диагоналей нижнего треугольника и в конце диагоналей верхнего треугольника – тогда хранение построчное, либо только в начале диагоналей – тогда хранение нижнего треугольника исходной матрицы осуществляется по строкам, верхнего - по столбцам). Дополнительно необходимо хранить одномерный целый массив размерностью $m-1$, в котором будет указан сдвиг каждой побочной диагонали от главной: положительные индексы для верхнего треугольника, отрицательные для нижнего треугольника. Например, для матрицы, изображенной на рис. 7, это будет массив $I=\left[-i_2,-i_1,i_3,i_4,i_5\right]$.

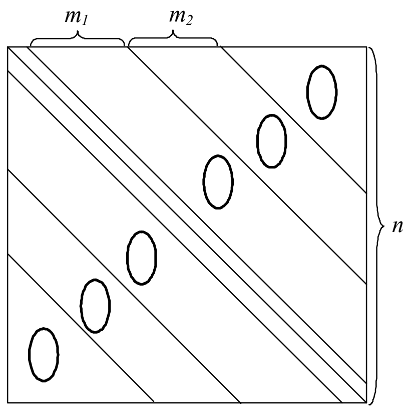
На рис. 7 изображена произвольная диагональная матрица. Однако, в данном курсе будут рассмотрены диагональные матрицы определенной структуры. Матрицы могут быть несимметричными, но расположение диагоналей матрицы относительно главной диагонали всегда симметрично:

1. 5-ти диагональная матрица с параметрами n – размерность матрицы, m – количество нулевых диагоналей.

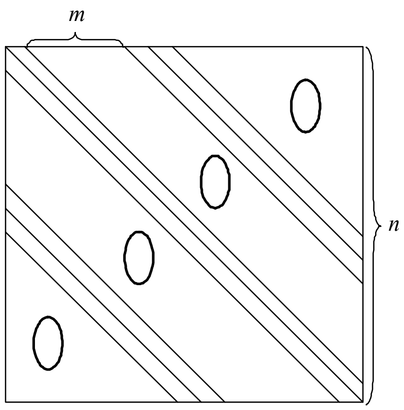


2. 7-ми диагональная матрица с параметрами n – размерность матрицы, m – количество нулевых диагоналей или 7-ми диагональная матрица с параметрами m_1, m_2 – количество нулевых диагоналей.





3. 9-ти диагональная матрица m – количество нулевых диагоналей, n – размерность матрицы.



Профильные форматы

Профильные форматы хранения матриц используется, когда матрица не обладает определенной структурой и ненулевые элементы расположены в произвольном порядке, но при этом они сосредоточены у главной диагонали, так что в строке можно выделить так называемый профиль – это часть строки от первого ненулевого элемента в строке до диагонального элемента. Причем, если между первым ненулевым элементом и элементом диагонали есть нулевые элементы, то они будут храниться в профиле. Если расположение элементов матрицы (портрет матрицы) является симметричным относительно главной диагонали, то такой формат называется *симметричный профильный*. Пример такой матрицы показан на рис. 8.

$$
\begin{array}{r l}
 & \\
A= & \left[\begin{array}{ccccccccc}
a_{11} & & & & & & & & \\
 & a_{22} & a_{23} & a_{24} & & & & 0 & \\
 & a_{32} & a_{33} & 0 & a_{35} & a_{36} & & & \\
 & a_{42} & 0 & a_{44} & a_{45} & 0 & a_{47} & & \\
 & & a_{53} & a_{54} & a_{55} & a_{56} & 0 & a_{58} & a_{59} \\
 & & a_{63} & 0 & a_{65} & a_{66} & 0 & a_{68} & 0 \\
 & & & a_{74} & 0 & 0 & a_{77} & 0 & a_{79} \\
 & 0 & & & a_{85} & a_{86} & 0 & a_{88} & 0 \\
 & & & & a_{95} & 0 & a_{97} & 0 & a_{99}
\end{array}\right] \\
\text{профиль 6-ой строки} \rightarrow a_{63} &
\end{array}
$$

Рис. 8.

Если портрет матрицы относительно главной диагонали несимметричный, то формат называется *несимметричный профильный*. Тогда в этом случае профиль – это часть строки от первого ненулевого элемента до последнего ненулевого элемента в строке. И формат в этом случае называется *строчный профильный.*

Рассмотрим подробнее симметричный профильный формат. Этот формат подразумевает, что элементы нижнего треугольника хранятся по строкам, элементы верхнего треугольника хранятся по столбцам.

Массивы, необходимые для хранения профильной квадратной матрицы, размерности n в симметричном профильном формате:

Массив $ia(n+1)$ - целый индексный массив. Массив ia содержит указатели начала строк (столбцов) нижнего (верхнего) треугольника в массиве al (au). Разность элементов $ia\left(i+1\right)-ia(i)$ показывает количество внедиагональных элементов в профиле i-ой строки (столбца) нижнего (верхнего) треугольника. Индексный массив ia формируется следующим образом: первые два элемента индексного массива ia всегда равны 1: $ia(1)=ia(2)=1$, так как в первой строке в нижнем треугольнике нет внедиагональных элементов. Далее к элементу $ia(i)$ добавляется количество элементов в профиле i-ой строки и получается элемент $ia(i+1)$. Количество внедиагональных элементов нижнего (верхнего) треугольника равно $k=ia(n+1)-1$.

Массивы $al(k)$ и $au(k)$ - вещественные массивы, которые хранят внедиагональные элементы нижнего (по строкам) и верхнего (по столбцам) треугольника матрицы. Индекс $ia(i)$ указывает номер элемента в массиве al (au), который является первым элементом i-ой строки (столбца).

Массив $di(n)$ - вещественный массив. Он хранит все диагональные элементы.

Если матрица симметричная, то массив $al=au$, и хранится только верхний или нижний треугольник, то есть только массив al или au.

<u>Пример</u>

Симметричный профильный формат матрицы A размерностью 9×9, изображенной на рис.8:

Индексный массив $ia(10)=\left(1,1,1,2,4,6,9,12,15,19\right)$ - указатели начала строк (столбцов) в массиве al (au).

Массивы al и au имеют размерность $k=ia(10)-1=18$. В массив al записываем последовательно профиль каждой строки, в массив au записываем по столбцам профиль каждого столбца:

$$al(18)=\left(\underbrace{a_{32}}_{3-я\ строка},\underbrace{a_{42},0}_{4-я\ строка},\underbrace{a_{53},a_{54}}_{5-я\ строка},\underbrace{a_{63},0,a_{65}}_{6-я\ строка},\underbrace{a_{74},0,0}_{7-я\ строка},\underbrace{a_{85},a_{86},0}_{8-я\ строка},\underbrace{a_{95},0,a_{97},0}_{9-я\ строка}\right),$$

$$au(18)=\left(\underbrace{a_{23}}_{3-й\ столбец},\underbrace{a_{24},0}_{4-й\ столбец},\underbrace{a_{35},a_{45}}_{5-й\ столбец},\underbrace{a_{36},0,a_{56}}_{6-й\ столбец},\underbrace{a_{47},0,0}_{7-й\ столбец},\underbrace{a_{58},a_{68},0}_{8-й\ столбец},\underbrace{a_{59},0,a_{79},0}_{9-й\ столбец}\right).$$

Рассмотрим, как получить элементы любой строки, например, шестой строки. Первый элемент шестой строки: $al\left(ia(6)\right)=al\left(6\right)=a_{63}$. Количество элементов шестой строки равно $ia\left(7\right)-ia(6)=9-6=3$, то есть это три элемента в массиве al, начиная с элемента $al\left(6\right)=a_{63}$ - a_{63}, 0, a_{65}.

Диагональные элементы хранятся в массиве

$di=\left(a_{11},a_{22},a_{33},a_{44},a_{55},a_{66},a_{77},a_{88},a_{99}\right)$.

Разреженные форматы

Разреженные форматы хранения матриц используется, когда в матрице ненулевые элементы расположены в произвольном порядке. Эти форматы аналогичны профильным форматам. Отличие в том, что хранятся только ненулевые элементы. Если элементы нижнего треугольника хранятся по строкам, а элементы верхнего треугольника – по столбцам, то такой формат называется *разреженным строчно-столбцовым*. Причем, если портрет матрицы симметричный (матрица может быть при этом несимметричной), то формат называется *симметричным*, если портрет несимметричный, то формат – *разреженный строчно-столбцовый несимметричный*. Если хранение ненулевых элементов матрицы осуществляется по строкам, то такой формат называется *разреженным строчным*.

Рассмотрим подробнее разреженный строчно-столбцовый симметричный формат хранения матриц. В отличие от симметричного профильного формата дополнительно необходим еще один массив, содержащий информацию о положении внедиагонального элемента в строке (для нижнего треугольника) или в столбце (для верхнего треугольника).

Итак, необходимые массивы для разреженного строчно-столбцового симметричного формата.

Одномерный целый массив $ia(n+1)$ - указатели начала строк и столбцов (для несимметричной матрицы) в массивах ja, al и au.

Количество внедиагональных элементов нижнего и верхнего треугольников $k=ia(n+1)-1$.

Одномерный целый массив $ja(k)$ - номера столбцов (строк) внедиагональных элементов нижнего (верхнего) треугольника матрицы. Записывается аналогично массиву al - последовательно по строкам номер столбца каждого элемента в строке нижнего треугольника (или по столбцам номер строки каждого элемента в столбце верхнего треугольника).

Одномерные вещественные массивы $al(k)$ и $au(k)$ – внедиагональные элементы нижнего (хранение по строкам) и верхнего (хранение по столбцам) треугольника матрицы.

Одномерный вещественный массив $di(n)$ - диагональные элементы матрицы.

Пример

Разреженный строчно-столбцовый симметричный формат матрицы, представленной на рис. 8: индексный массив $ia(10) = \left(1,1,1,2,3,5,7,8,10,12\right)$. Количество внедиагональных элементов $k = ia(10) - 1 = 11$. Массив номеров столбцов (строк) $ja(11) = (2,2,3,4,3,5,4,5,6,5,7)$.

Массивы внедиагональных ненулевых элементов по строкам для массива al и по столбцам для массива au:

$al(11) = \left(a_{32}, a_{42}, a_{53}, a_{54}, a_{63}, a_{65}, a_{74}, a_{85}, a_{86}, a_{95}, a_{97}\right)$,

$au(11) = \left(a_{23}, a_{24}, a_{35}, a_{45}, a_{36}, a_{56}, a_{47}, a_{58}, a_{68}, a_{59}, a_{79}\right)$.

Найдем внедиагональные элементы, например, для шестой строки: начало шестой строки в массивах ja и al: $ia(6) = 5$. Количество элементов в шестой строке $ia(7) - ia(6) = 7 - 5 = 2$. Начиная с пятого элемента в массиве al, последовательно находятся два элемента шестой строки a_{63} и a_{65}, которые располагаются в столбцах $ja\left(ia\left(6\right)\right) = 3$ и $ja\left(ia\left(6\right) + 1\right) = 5$.

Порядок выполнения лабораторной работы №3

1. Написать программу, реализующую на языке ФОРТРАН требуемые действия над матрицами произвольной (задаваемой пользователем) размерности с учетом следующих требований:

- для распределения памяти под необходимые массивы использовать идеи псевдодинамической памяти; память выделять в головной программе, описав 1 одномерный массив максимально возможной для данного компьютера размерности; вся оставшаяся после распределения память должна находиться в конце этого массива; при недостатке памяти выдавать соответствующее сообщение;
- в подпрограммах, где это необходимо, матрицы описывать как двумерный массив;
- каждое действие должно быть реализовано в виде подпрограммы; для тех вариантов, в которых, возможно реализовать перемножение строки на столбец с помощью перемножения одномерных векторов, обязательно наличие такой подпрограммы;
- вычисления должны производиться оптимальным образом;
- все вводимые данные должны проверятся на корректность, для любых данных программа должна давать корректный результат (сообщение об ошибке

на русском языке – корректный результат, сообщение системы о делении на ноль - некорректный результат);

- все входные данные должны вводиться из файлов (матрица и вектор в разных файлах); в первой строке файла хранятся размерности объекта (например, 2 целых числа для прямоугольной плотной матрицы, одно – для вектора).
- матрица в файле хранится в заданном формате.

2. Протестировать разработанную программу. Для тестирования использовать матрицы небольшой размерности.

3. Реализовать задание лабораторной работы с использованием написанных ранее подпрограмм, при условии, что матрицы и вектора хранятся в файлах прямого доступа.

4. Для тестирования программы написать программу перевода текстовых файлов, содержащих матрицу в заданном формате, в файлы прямого доступа.

5. Разработать программу генерации тестов большой размерности, при представлении матриц в заданном формате. Программа должна создавать файлы прямого доступа.

6. Разработать программу, распечатывающую матрицы, представленные по формату, в виде плотных матриц.

7. Протестировать разработанную программу. Для тестирования использовать матрицы большой и максимально возможной для данного компьютера размерности.

8. Дополнительное исследование. Провести исследование блочного ввода-вывода.

Для данного исследования использовать функцию Gettim(I1,I2,I3,I4), возвращающую системное время. В переменные I1,I2,I3,I4 типа integer*2 сохраняются часы, минуты, секунды, миллисекунды соответственно.

Для матриц большой размерности N с помощью данной функции вычислить время перемножения матрицы на вектор и время считывания матрицы из файла прямого доступа.

Исследовать, изменяется ли время считывания матрицы в зависимости от размера записи файла прямого доступа (длина записи равна 4 байта, 8 байт, 16 байт, 32 байта, 64 байта) и от количества переменных, считываемых за один раз. Сделать необходимые корректировки в программе.

Полученные результаты представить в таблице:

N	Время умножения матрицы на вектор	Время считывания				
		4 байта	8 байт	16 бай	32 байта	64 байта
20000						
50000						
…						
Максимально возможная размерность						

Проанализировать полученные результаты. Сделать соответствующие выводы об изменении время считывания из файла прямого доступа в зависимости от длины записи и количества переменных, считываемых за один раз.

Варианты заданий для 3 лабораторной работы

1. Умножение прямоугольной матрицы на вектор.
2. Умножение квадратной матрицы на вектор.
3. Произведение прямоугольных матриц.
4. Степень квадратной матрицы.
5. Произведение двух квадратных матриц.
6. Симметризация Гаусса (умножение матрицы на транспонированную к ней) для прямоугольных матриц.
7. Симметризация Гаусса для квадратных матриц.
8. Умножение симметричной квадратной матрицы на вектор. Хранить только нижний треугольник (в файле - вся матрица).
9. Умножение квадратной ленточной несимметричной матрицы на вектор. В файле задается полуширина ленты и лента. Лента в памяти хранится построчно.
   9.1 С выделением диагонали **9.2** Без выделения диагонали
10. Умножение ленточной симметричной матрицы на вектор. Хранится только половина ленты. В файле задается полуширина ленты (для диагональной матрицы полуширина равна 1) и матрица.
   10.1 С выделением диагонали **10.2** Без выделения диагонали
11. Умножение квадратной ленточной несимметричной матрицы на вектор. Лента в памяти хранится в трех массивах: отдельно диагональ, нижний треугольник по строкам, верхний треугольник по столбцам. На вход поступает матрица в трёх файлах. Размерность и полуширина ленты (для диагональной матрицы полуширина равна 1) хранятся в четвёртом файле
12. Умножение 5-диагональной матрицы на вектор.
   12.1 Симметричная матрица. **12.2** Несимметричная матрица.
13. Умножение 7-диагональной матрицы на вектор.
   13.1 Симметричная матрица. **13.2** Несимметричная матрица.
14. Умножение 9-диагональной матрицы на вектор.
   14.1 Симметричная матрица. **14.2** Несимметричная матрица.
15. Умножение профильной матрицы на вектор.
   15.1 Симметричная матрица. **15.2** Несимметричная матрица.
16. Умножение матрицы в разреженном строчно-столбцовом формате на вектор.
   16.1 Симметричная матрица. **16.2** Несимметричная матрица.

Контрольные вопросы

1. Массивы: оператор DIMENSION, положение элемента массива в памяти, способы передачи массивов в подпрограммы.

2. Передача данных с помощью параметров процедур, передача данных с помощью общего блока памяти: оператор COMMON.
3. Псевдодинамическое распределение памяти.
4. Файлы прямого доступа. Спецификаторы ACCESS='DIRECT' и RECL оператора OPEN. Организация ввода-вывода из файлов прямого доступа с помощью операторов READ и WRITE.
5. Форматы матриц: ленточный, диагональный, симметричный и несимметричный профильный, разреженный строчно-столбцовый.
6. Преобразование матрицы из одного формата в другой.

ЛАБОРАТОРНАЯ РАБОТА № 4

Цель работы

Изучение методов численного интегрирования, оценки порядка точности, оценки погрешности по правилу Рунге, уточнения значений по Ричардсону.

Теоретическая часть

Основные определения

Общий принцип численного интегрирования следует из определения определенного интеграла: интеграл – предел последовательности частичных сумм функции.

Введем понятие **квадратурной формулы**. Пусть необходимо вычислить определенный интеграл

$$I = \int_a^b f(x)\,dx \tag{1}$$

от непрерывной на $[a,b]$ функции f. Приближенное равенство

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} q_i f(x_i) \tag{2}$$

где q_i – некоторые числа, x_i – некоторые точки отрезка $[a,b]$, называется **квадратурной формулой,** определяемой весами q_i и узлами x_i.

Говорят, что квадратурная формула *точна* для многочленов степени m, если для произвольного алгебраического многочлена степени m приближенное равенство (2) является точным.

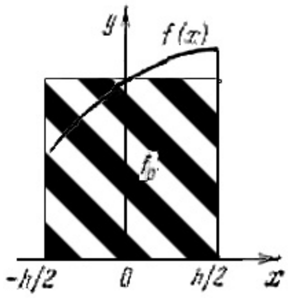
Порядок точности определяется порядком малости остаточного члена относительно шага разбиения h. Порядок точности показывает, во сколько раз уменьшается погрешность формулы метода при вычислении интеграла на сетке с шагом $h/2$ относительно погрешности, полученной при вычислениях на сетке с шагом h.

Каноническая формула прямоугольников

Пусть $f \in C^2\left[-\frac{h}{2},\frac{h}{2}\right]$, $h > 0$. Положим приближенно

$$\int_{-\frac{h}{2}}^{\frac{h}{2}} f(x)\,dx \approx hf(0), \tag{3}$$

то есть площадь криволинейной трапеции, ограниченной сверху графиком функции, аппроксимируется площадью заштрихованного прямоугольника, высота которого равна значению f в средней точке основания трапеции.



Оценим погрешность формулы (3). Для этого определим первообразную $F(x)$ функции $f(x)$ как

$$F(x) = \int_0^x f(t)\,dt. \tag{4}$$

Т.к. $F(0) = 0$, $F'(x) = f(x)$, $F''(x) = f'(x)$, $F'''(x) = f''(x)$ и т.д., то согласно формуле Тейлора с остаточным членом в форме Лагранжа имеем:

$$F\left(\pm\frac{h}{2}\right) = \pm\frac{h}{2}f(0) + \frac{h^2}{8}f'(0) \pm \frac{h^3}{48}f''(\xi_\pm), \tag{5}$$

где $\xi_- \in \left[-\frac{h}{2},0\right]$, $\xi_+ \in \left[0,\frac{h}{2}\right]$.

Поскольку функция (4) является первообразной для $f(x)$, для интеграла, стоящего в левой части приближенного равенства (3) из формулы Ньютона-Лейбница с учетом (5) следует:

$$\int_{-\frac{h}{2}}^{\frac{h}{2}} f(x)\,dx = F\left(\frac{h}{2}\right) - F\left(-\frac{h}{2}\right) = hf(0) + \frac{h^3}{24}\frac{f''(\xi_-) + f''(\xi_+)}{2}, \tag{6}$$

откуда

$$\int_{-\frac{h}{2}}^{\frac{h}{2}} f(x)\,dx = hf(0) + \frac{h^3}{24} f''(\xi), \quad |\xi| \le \frac{h}{2} \tag{7}$$

Формула (7) называется **канонической квадратурной формулой метода прямоугольников с остаточным членом.**

Усложненная квадратурная формула прямоугольников

Пусть требуется вычислить приближенно интеграл (1) от функции $f \in C^2[a,b]$. Разобьём область интегрирования $[a,b]$ на N отрезков произвольным образом. Обозначим частичные отрезки через $\left[x_i, x_{i+1}\right]$, $i = \overline{1,N}$, $x_1 = a$, $x_{N+1} = b$, $h_i = x_{i+1} - x_i$ - длина частичного отрезка. На каждом частичном отрезке применим каноническую квадратурную формулу и просуммируем полученные результаты. Построенная таким образом квадратурная формула называется **усложненной**.

В соответствии с (7) интеграл на одном частичном отрезке вычисляется

$$\int_{x_i}^{x_{i+1}} f(x)\,dx = h_i f_{i+1/2} + \frac{h_i^3}{24} f''(\xi_i), \tag{8}$$

где $f_{i+1/2} = f\left(\dfrac{x_i + x_{i+1}}{2}\right)$ - значение функции в середине частичного отрезка $\left[x_i, x_{i+1}\right]$, $\xi_i \in \left[x_i, x_{i+1}\right]$.

Суммирование по всем отрезкам равенства (8) приводит к **усложненной квадратурной формуле прямоугольников:**

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{N} h_i f_{i+1/2}\,. \tag{9}$$

для которой остаточный член ε_h имеет вид

$$\varepsilon_h = \frac{1}{24} \sum_{i=1}^{N} h_i^3 f''(\xi_i), \tag{10}$$

Оценим порядок малости остаточного члена усложненной квадратурной формулы относительно длины наибольшего частичного отрезка $\left[x_i, x_{i+1}\right]$ - $h_{\max} = \max\limits_{i=\overline{1,N}} h_i$.

С учётом того, что сумма $\sum_{i=1}^{N} h_i f''(\xi_i) \le M(b-a)$, где $M = \max_{x\in[a,b]} f''(x)$, остаточный член можно оценить следующим образом

$$\varepsilon_h = \frac{1}{24}\sum_{i=1}^{N} h_i^3 f''(\xi_i) \le \frac{h_{\max}^2}{24}\sum_{i=1}^{N} h_i f''(\xi_i) \le \frac{h_{\max}^2}{24} M(b-a), \tag{11}$$

и так как $\frac{M(b-a)}{24}$ - константа, не зависящая от шагов разбиения h_i, то можно утверждать, что остаточный член усложненной квадратурной формулы прямоугольников имеет второй порядок относительно $h_{\max}$ и $\varepsilon_h = O(h_{\max}^2)$.

Если отрезок интегрирования разбит на N равных отрезков, то усложненная квадратурная формула с остаточным членом примет вид

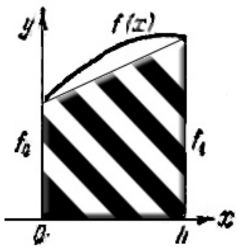
$$\int_a^b f(x)\,dx = h\sum_{i=1}^{N} f_{i+1/2} + h^2\frac{b-a}{24} f''(\xi). \tag{12}$$

Каноническая квадратурная формула трапеций

Пусть $f \in C^2[0,h]$, $h > 0$. Положим приближенно

$$\int_0^h f(x)\,dx \approx h\frac{f(0)+f(h)}{2}, \tag{13}$$

то есть площадь криволинейной трапеции, ограниченной сверху графиком функции, приближенно заменяется площадью заштрихованной трапеции.



Получим остаточный член для формулы (13). Как и ранее, определим первообразную $F(x)$ функции $f(x)$ выражением (4). По формуле Тейлора с остаточным членом в интегральной форме для функций $f(x)$ и $F(x)$ в точке $x = h$ имеем:

$$f(h) = f(0) + hf'(0) + \int_0^h (h-t) f''(t)\,dt, \tag{14}$$

$$\begin{aligned} F(h) &= F(0) + hF'(0) + \frac{h^2}{2}F''(0) + \frac{1}{2}\int_0^h (h-t)^2 F'''(t)\,dt = \\ &= hf(0) + \frac{h^2}{2} f'(0) + \frac{1}{2}\int_0^h (h-t)^2 f''(t)\,dt\,. \end{aligned} \tag{15}$$

Из выражений (14) и (15), с учетом того, что $F(x)$ – первообразная $f(x)$, получаем:

$$\int_0^h f(x)\,dx = h\frac{f(0)+f(h)}{2} - \frac{1}{2}\int_0^h (h-t)tf''(t)\,dt,$$

и, используя теорему о среднем (см. приложение), получаем

$$\frac{1}{2}\int_0^h (h-t)tf''(t)\,dt = \frac{1}{2}f''(\xi)\int_0^h (h-t)t\,dt = \frac{h^3}{12}f''(\xi).$$

Таким образом, мы пришли к **канонической формуле трапеций** с остаточным членом:

$$\int_0^h f(x)\,dx = h\frac{f(0)+f(h)}{2} - \frac{h^3}{12}f''(\xi), \quad \xi \in [0,h]. \tag{16}$$

Усложненная квадратурная формула трапеций

Как и в методе прямоугольников, разобьем промежуток интегрирования на N произвольных отрезков $\left[x_i, x_{i+1}\right]$, $i=\overline{1,N}$, $x_1=a$, $x_{N+1}=b$, $h_i = x_{i+1}-x_i$. Просуммируем все значения интегралов, найденные на каждом из частичных отрезков в соответствии с (16), и получим **усложненную квадратурную формулу трапеций**:

$$\int_a^b f(x)\,dx \approx \frac{1}{2}\left[f(a)\cdot h_1 + \sum_{i=2}^{N} f\left(x_i\right)\cdot\left(h_{i-1}+h_i\right) + f(b)\cdot h_N\right] \tag{17}$$

с остаточным членом ε_h, имеющим второй порядок относительно $h_{\max}$ $\varepsilon_h = O(h_{\max}^2)$.

Если промежуток разбит на N равных частей, то усложненная квадратурная формула трапеций с остаточным членом примет вид

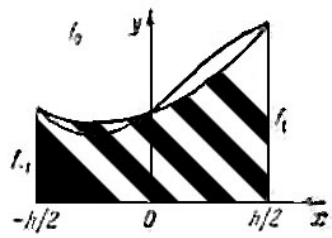
$$\int_a^b f(x)\,dx = h\left[\frac{f(a)}{2} + \sum_{i=1}^{N-1} f\left(x_i\right) + \frac{f(b)}{2}\right] - h^2\frac{b-a}{12}f''(\xi), \ \ \xi \in [a,b] \tag{18}$$

Каноническая формула Симпсона

Можно ещё больше увеличить порядок точности метода, если увеличить на интервале степень полинома, для которого вычисляется площадь криволинейной трапеции. Пусть $f \in C^4[-h/2, h/2]$, $h > 0$. Интеграл $\int\limits_{-h/2}^{h/2} f(x)dx$ приближенно заменим площадью заштрихованной криволинейной трапеции, ограниченной сверху параболой, проходящей через точки $(-h/2, f(-h/2))$, $(0, f(0))$ и $(h/2, f(h/2))$. Указанная парабола задается уравнением $y = f(0) + \dfrac{f\left(\frac{h}{2}\right) - f\left(-\frac{h}{2}\right)}{h} x + \dfrac{2\left[f\left(-\frac{h}{2}\right) - 2f(0) + f\left(\frac{h}{2}\right)\right]}{h^2} x^2$.

Таким образом, получаем **каноническую формулу Симпсона:**

$$\int\limits_{-h/2}^{h/2} f(x)\,dx \approx \int\limits_{-h/2}^{h/2} y\,dx = \frac{h}{6}\left(f\left(-\frac{h}{2}\right) + 4f(0) + f\left(\frac{h}{2}\right)\right). \qquad (19)$$



Формула Симпсона называется также формулой парабол.

Пусть F – первообразная функции $f(x)$, определяемая соотношением (4). Тогда согласно формуле Тейлора с остаточным членом в интегральной форме имеем:

$$F\left(\pm\frac{h}{2}\right) = \pm\frac{h}{2} f(0) + \frac{h^2}{8} f'(0) \pm \frac{h^3}{48} f''(0) + \frac{h^4}{384} f'''(0) \pm$$
$$\pm \frac{1}{384} \int\limits_0^{h/2} (h/2 - t)^4 f^{(4)}(\pm t)dt,$$

$$f\left(\pm\frac{h}{2}\right) = f(0) \pm \frac{h}{2} f'(0) + \frac{h^2}{8} f''(0) \pm \frac{h^3}{48} f'''(0) + \frac{1}{48} \int\limits_0^{\frac{h}{2}} \left(\frac{h}{2} - t\right)^3 f^{(4)}(\pm t)dt,$$

откуда

$$\int\limits_{-\frac{h}{2}}^{\frac{h}{2}} f(x)\,dx = F\left(\frac{h}{2}\right) - F\left(-\frac{h}{2}\right) = \frac{h}{6}\left[f\left(-\frac{h}{2}\right) + 4f(0) + f\left(\frac{h}{2}\right)\right] -$$
$$- \frac{1}{384} \int\limits_0^{\frac{h}{2}} \left(\frac{h}{2} - t\right)^3 \left(\frac{h}{6} + t\right)\left(f^{(4)}(t) + f^{(4)}(-t)\right)dt. \qquad (20)$$

Применив теорему о среднем (37) и преобразовав полученное выражение, с учётом того, что

$$\frac{1}{384}\int_{0}^{\frac{h}{2}}\left(\frac{h}{2}-t\right)^{3}\left(\frac{h}{6}+t\right)dt=\frac{h^{5}}{9216}, \tag{21}$$

получим каноническую формулу Симпсона с остаточным членом:

$$\int_{-\frac{h}{2}}^{\frac{h}{2}}f(x)dx=\frac{h}{6}\left(f\left(-\frac{h}{2}\right)+4f(0)+f\left(\frac{h}{2}\right)\right)-\frac{h^{5}}{9216}f^{(4)}(\xi). \tag{22}$$

Усложненная квадратурная формула Симпсона

Аналогично методам, рассмотренным ранее, разобьём отрезок интегрирования на N произвольных сегментов $\left[x_i,x_{i+1}\right]$, $i=\overline{1,N}$, $x_1=a$, $x_{N+1}=b$, $h_i=x_{i+1}-x_i$. Запишем каноническую формулу Симпсона применительно к отрезку $[x_i,x_{i+1}]$:

$$\int_{x_i}^{x_{i+1}}f(x)dx\approx\frac{h_i}{6}\left[f(x_i)+4f\left(x_i+\frac{h_i}{2}\right)+f(x_{i+1})\right].$$

Суммируя левую и правую части этого соотношения по i от 1 до N, получаем **усложненную квадратурную формулу Симпсона**

$$\int_{a}^{b}f(x)dx\approx\frac{1}{6}\left(f(a)h_1+f(b)h_N+4\sum_{i=1}^{N}f\left(x_i+\frac{h_i}{2}\right)h_i+\sum_{i=2}^{N}f(x_i)(h_{i-1}+h_i)\right). \tag{23}$$

Остаточный член ε_h усложненной квадратурной формулы Симпсона имеет четвертый порядок относительно $h_{\max}$ $\varepsilon_h=O(h_{\max}^4)$.

Если отрезок интегрирования разбит на N равных сегментов, то усложненная квадратурная формула с остаточным членом имеет вид:

$$\int_{a}^{b}f(x)dx=\frac{h}{6}\left(f(a)+f(b)+4\sum_{i=1}^{N}f\left(x_{i-1}+\frac{h}{2}\right)+2\sum_{i=1}^{N-1}f(x_i)\right)-$$
$$-h^4\frac{b-a}{9216}f^{(4)}(\xi). \tag{24}$$

Каноническая квадратурная формула Гаусса

Очевидно, что для дальнейшего уменьшения ошибки формул численного интегрирования требуется ещё повысить порядок точности метода.

Поставим следующую задачу: найти среди всех квадратурных формул с n узлами вида (2) квадратурную формулу с таким расположением узлов x_i на отрезке $[a,b]$ и с такими весами q_i, при которых она точна для алгебраических многочленов максимально возможной степени. Заметим, что эта степень обязательно меньше, чем $2n$, так как при любом выборе узлов x_i и весов q_i, $i=\overline{1,n}$, можно построить многочлен степени $2n$ вида $P_{2n}=\left(x-x_1\right)^2\left(x-x_2\right)^2\ldots\left(x-x_N\right)^2$, который обладает тем свойством, что $\int\limits_a^b P_{2n}(x)\,dx>0$, а $\sum\limits_{i=1}^{n} q_i P_{2n}\left(x_i\right)=0$, то есть приближенное равенство (2) не может быть выполнено точно.

Построим формулы интегрирования для отрезка $[-1,1]$. Пусть $x_j\in[-1,1], j=\overline{1,n}$ – произвольные попарно непересекающиеся узлы. Тогда, если взять веса

$$q_j=\int\limits_{-1}^{1} p_j(x)\,dx\,,\ \ j=\overline{1,n}, \tag{25}$$

где $p_j(x)=\dfrac{\left(x-x_1\right)\ldots\left(x-x_{j-1}\right)\left(x-x_{j+1}\right)\ldots\left(x-x_n\right)}{\left(x_j-x_1\right)\ldots\left(x_j-x_{j-1}\right)\left(x_j-x_{j+1}\right)\ldots\left(x_j-x_n\right)}$ – **лагранжевы коэффициенты** для интерполяционного многочлена степени $n-1$ (см. Приложение), то квадратурная формула будет точна, во всяком случае, для многочленов степени $n-1$. Действительно, любой алгебраический многочлен $P_{n-1}(x)$ степени $n-1$ точно восстанавливается интерполяционным многочленом $L_{n-1}(x)$ той же степени. Поэтому $\int\limits_{-1}^{1} P_{n-1}(x)\,dx=\int\limits_{-1}^{1} L_{n-1}(x)\,dx$, и по определению многочлена Лагранжа

$$\int\limits_{-1}^{1} L_{n-1}(x)\,dx=\int\limits_{-1}^{1}\sum_{j=1}^{n} p_j(x)P_{n-1}\left(x_j\right)dx=$$

$$=\sum_{j=1}^{n}\left(\int\limits_{-1}^{1} p_j(x)\,dx\right)P_{n-1}\left(x_j\right)=\sum_{j=1}^{n} q_j P_{n-1}\left(x_j\right).$$

А это и означает, что квадратурная формула с такими параметрами точна для многочленов $n-1$ степени.

Возьмем теперь в качестве узлов $x_j, j=\overline{1,n}$ корни многочлена Лежандра $X_n(x)$ (см. Приложение), которые согласно свойству 3 (см. Приложение) расположены в интервале $[-1,1]$, а веса q_j найдем по формуле (25).

Квадратурная формула (2) с выбранными узлами и весами точна для многочленов степени $2n-1$ и тем самым для многочленов максимальной степени.

Таким образом, **каноническая квадратурная формула Гаусса** имеет вид

$$\int_{-1}^{1} f(x)\,dx \approx \sum_{j=1}^{n} q_j f\left(x_j\right), \tag{26}$$

где веса q_j определяются соотношением (25), а x_j – корни многочлена Лежандра $X_n(x)$.

Узлы x_j расположены симметрично относительной точки $x=0$, а веса q_j положительны и в симметричных уздах совпадают при любом n.

Приведем численные выражения неотрицательных узлов x_j и весов q_j формула Гаусса для *n*=1, 2, 3, 4 с пятнадцатью десятичными знаками после запятой.

n	x_1 q_1	*n*	x_1 q_1	x_2 q_2
1	0,000000000000000 2,000000000000000	3	0,000000000000000 0,888888888888889	0,774596669241483 0,555555555555556
2	0,577350269189625 1,000000000000000	4	0,339981043584856 0,652145154862546	0,861136311594053 0,347854845137454

Примечание: можно проверить, что при $n=1$ формула Гаусса совпадает с канонической формулой прямоугольников.

Усложненная квадратурная формула Гаусса

Разобьем отрезок $[a,b]$ на N произвольных частичных отрезков $[x_{k-1},x_k], k=\overline{1,N}$, $x_1=a$, $x_{N+1}=b$, $h_k=x_{k+1}-x_k$. На каждом частичном отрезке зададим по n узлов:

$$x_{k,j}=\frac{x_k+x_{k+1}}{2}+\tilde{x}_j\frac{h_k}{2},\ \ j=\overline{1,n}, \tag{27}$$

где $\tilde{x}_j$ – узлы канонической формулы Гаусса (26). Расположение узлов на частичном отрезке геометрически подобно расположению узлов канонической формулы. В силу этого, квадратурная формула:

$$\int_{x_k}^{x_{k+1}} f(x)\,dx \approx \frac{h_k}{2}\sum_{j=1}^{n} q_j f\left(x_{k,j}\right) \tag{28}$$

также точна для многочленов степени $2n-1$.

Суммируя соотношение (28) по k, получаем **усложненную квадратурную формулу Гаусса:**

$$\int_a^b f(x)\,dx \approx \frac{1}{2}\sum_{j=1}^{n} q_j \sum_{k=1}^{N} h_k \cdot f\left(x_{k,j}\right) \tag{29}$$

Остаточный член ε_h данной формулы имеет $2n$-ый порядок относительно $h_{\max}$ и $\varepsilon_{h_i} = O(h_{\max}^{2n})$.

Остаточный член усложненной квадратурной формулы Гаусса при разбиении отрезка интегрирования $[a,b]$ на N равных отрезков имеет вид:

$$R_n = \frac{(b-a)^{2n+1}}{N^{2n}} \frac{(n!)^4}{((2n)!)^3(2n+1)} f^{2n}(\xi) \tag{30}$$

Правило Рунге практической оценки погрешности численного интегрирования

Пусть I^* – неизвестное точное значение некоторого интеграла. Обозначим за I_{h_i} - приближенное значение интеграла, полученное при разбиении отрезка $[a,b]$ на N произвольных сегментов $[x_{i-1}, x_i]$, $i=\overline{1,N}$, $h_i = x_{i+1} - x_i$, $h_{\max} = \max\limits_{i=\overline{1,N}} h_i$, $I_{\frac{h_i}{2}}$ - приближенное значение интеграла, полученное на сетке, в которой каждый частичный отрезок разбит пополам.

Применяя формулы численного интегрирования, можно получить следующее соотношение между точным значением I^* и приближенным значением I_{h_i}

$$I^* = I_{h_i} + C(h_{\max})^k + O\left((h_{\max})^{k+m}\right) \tag{31}$$

где C – некоторая не зависящая от $h_{\max}$ постоянная, k – порядок малости остаточного члена квадратурной формулы, $m \geq 1$ – некоторое целое число, определяемое остаточным членом квадратурной формулы.

Тогда для сетки с уменьшенными вдвое частичными отрезками справедливо

$$I^* = I_{\frac{h_i}{2}} + C\left(\frac{h_{\max}}{2}\right)^k + O\left((h_{\max})^{k+m}\right) \tag{32}$$

Вычитая (31) из (32), получим

$$I_{\frac{h_i}{2}} - I_{h_i} = C\left(\frac{h_{\max}}{2}\right)^k \left(2^k - 1\right) + O\left((h_{\max})^{k+m}\right),$$

откуда

$$C\left(\frac{h_{\max}}{2}\right)^k = \frac{I_{\frac{h_i}{2}} - I_{h_i}}{2^k - 1} + O\Big(\left(h_{\max}\right)^{k+m}\Big). \tag{33}$$

Подставив (33) в (32), получим **правило Рунге** оценки погрешности численного интегрирования:

$$I^* - I_{\frac{h_i}{2}} = \frac{I_{\frac{h_i}{2}} - I_{h_i}}{2^k - 1} + O\Big(\left(h_{\max}\right)^{k+m}\Big). \quad (34).$$

Уточнение приближенного решения по Ричардсону

Квадратурная формула

$$I^* = I_{\frac{h_i}{2}} + \frac{I_{\frac{h_i}{2}} - I_{h_i}}{2^k - 1} + O\Big((h_{\max})^{k+m}\Big) = \frac{2^k I_{\frac{h_i}{2}} - I_{h_i}}{2^k - 1} + O\Big((h_{\max})^{k+m}\Big) \quad (35)$$

имеет более высокий порядок малости остаточного члена, чем исходная формула (31). Число $I^R = \dfrac{2^k I_{\frac{h_i}{2}} - I_{h_i}}{2^k - 1}$ называется **уточненным (или экстраполированным) по Ричардсону** приближенным значением величины I^*.

Особенности практического использования правила Рунге и поправки Ричардсона

Необходимо отметить, что при практическом использовании правила (34) необходимо учитывать два обстоятельства.

Во-первых, при достаточно больших h_i влияние остаточного члена (который мы записали в виде $O\Big((h_{\max})^{k+m}\Big)$) может оказаться существенным, и оценка (34) окажется практически неверной.

Во-вторых, при слишком малом шаге h_i начинает влиять конечность используемой для вычислений арифметики, и оценка (34) вследствие этого может дать погрешность численного интегрирования существенно меньшую, чем есть на самом деле (просто значения I_{h_i} и $I_{\frac{h_i}{2}}$ практически не стали отличаться с точки зрения представления чисел в ЭВМ).

В-третьих, возможен случай, когда для интегрируемой функции константа C в (31) равна нулю (как видно из приведённых в этом пособии квадратурных формул, константа C определяется именно интегрируемой функцией). В этом случае правило Рунге даст неверную оценку просто потому, что порядок малости остаточного члена k определён неверно для конкретной интегрируемой функции.

На практике для оценки возможности применения правила Рунге и поправки Ричардсона часто используют соотношение

$$\left| 2^k \frac{I_{h_i} - I_{\frac{h_i}{2}}}{I_{2h_i} - I_{h_i}} - 1 \right| < 0.1 \tag{36}$$

Порядок выполнения лабораторной работы

1. Запрограммировать указанный метод, используя его усложненную квадратурную формулу.

2. Провести практическое исследование порядка точности указанного метода для многочленов степени m. Данные занести в таблицу следующего вида:

Название метода

Аналитическая оценка порядка точности k

Отрезок интегрирования

Степень полинома	Полином	Точное значение	Число отрезков	Шаг	Численное значение интеграла	Отношение погрешностей	Погрешность	Оценка погрешности по правилу Рунге	Уточнение по Ричардсону	Погрешность уточненного решения
m	$f(x)$	I^*	N	h	I_h	$\dfrac{I^* - I_{2h}}{I^* - I_h}$	$I^* - I_h$	$\dfrac{I_h - I_{2h}}{2^k - 1}$	I^R	$I^* - I^R$
$m = k-1$			1							
$m = k$			1							
			2							
$m = k+1$			1							
			2							
$m = k+2$			1							
			2							

3. Провести исследование порядка малости остаточного члена для нескольких функций. Среди них обязательно должна быть сильно осциллирующая (колеблющаяся) функция (4-5 колебаний на отрезке интегрирования). Заполнить для каждой функции следующую таблицу.

Название метода

Отрезок интегрирования

Порядок малости остаточного члена относительно h

Интегрируемая функция

Аналитическое значение интеграла

Число отрезков	Шаг	Численное значение интеграла на сетке с шагом h	Отношение погрешностей	Погрешность	Оценка погрешности по правилу Рунге	Уточнение по Ричардсону	Погрешность уточненного решения
N	h	I_h	$\dfrac{I^*-I_{2h}}{I^*-I_h}$	I^*-I_h	$\dfrac{I_h-I_{2h}}{2^k-1}$	I^R	I^*-I^R
1							
2							
4							
…							

4. Исследовать влияние длины слова при реализации численного интегрирования, для чего сравнить реализации методов с одинарной и двойной точностью на нескольких функциях. Замечание: При задании констант двойной точности на языке Фортран необходимо задавать их в виде PI=3,14159265358979D00, а также все переменные объявлять как REAL*8.
5. Провести исследование на неравномерных сетках. Выяснить, в каких промежутках сгущение сетки дает наилучшие результаты: в окрестности экстремумов, на местах осцилляции или в местах постоянства. Результаты сравнить по точности и по вычислительным затратам, например, при различном разбиении отрезка, но с равным количеством узлов, получены существенно разные по точности результаты; или при практически равной точности необходимо существенно разное количество узлов для различных разбиений. Выяснить, в каких промежутках (ближе к точкам экстремума или в местах быстрого роста функции) требуется увеличение числа отрезков интегрирования.

Замечания:

1) В пунктах 2 и 3 все вычисления выполнять с одинарной точностью. Все переменные, которые используются для накопления, и переменные результатов объявлять, как REAL*8.

2) При разработке тестов использовать дифференцирование, то есть подынтегральное выражение – это дифференциал функции, от которой считается интеграл.

Варианты заданий

1. Методы прямоугольников и трапеций
2. Методы прямоугольников и Симпсона
3. Методы трапеций и Симпсона
4. Методы прямоугольников и Гаусса-2
5. Методы трапеций и Гаусса-2
6. Методы прямоугольников и Гаусса-3
7. Методы трапеций и Гаусса-3
8. Методы прямоугольников и Гаусса-4
9. Методы трапеций и Гаусса-4
10. Методы Симпсона и Гаусса-2

11.Методы Симпсона и Гаусса-3
12.Методы Симпсона и Гаусса-4
13.Методы прямоугольников, Гаусса-2 и Гаусса-4
14.Методы трапеций, Гаусса-2 и Гаусса-4
15.Методы Симпсона, Гаусса-2 и Гаусса-4
16.Методы прямоугольников, Гаусса-3 и Гаусса-4
17.Методы трапеций, Гаусса-3 и Гаусса-4
18.Методы Симпсона, Гаусса-3 и Гаусса-4

Контрольные вопросы

1. Понятия порядка точности.
2. Численное интегрирование. Канонические квадратурные формулы методов прямоугольников, трапеций, Симпсона, Гаусса.
3. Вывод усложненных квадратурных формул методов прямоугольника, трапеций, Симпсона с остаточным членом для сеток с постоянным шагом разбиения h и для сеток с переменным шагом h_i.
4. Правило Рунге. Вывод правила Рунге для сеток с переменным шагом h_i.
5. Экстраполяция по Ричардсону.

Приложение. Перечень необходимых определений, теорем и свойств

Обобщенная теорема о среднем

Пусть $f, g \in C[a,b]$, причем $g(x) \geq 0$ на $[a,b]$. Тогда существует такая точка $\xi \in [a,b]$, что

$$\int_a^b f(x)g(x)dx = f(\xi)\int_a^b g(x)dx. \tag{37}$$

Многочлены Лежандра

Многочлены, задаваемые выражением

$$X_n(x) = \frac{1}{n!\cdot 2^n}\frac{d^n}{dx^n}(x^2 - 1)^n \tag{38}$$

называются многочленами Лежандра.

Свойства многочленов Лежандра

1) Многочлен Лежандра $X_n(x)$ степени n ортогонален любому алгебраическому многочлену $P_k(x)$ степени $k < n$.

2) Многочлен $X_n(x)$ является алгебраическим многочленом *n*-ой степени со старшим коэффициентом, не равным нулю. Для многочленов Лежандра справедлива рекуррентная формула $(n+1)X_{n+1}(x) - (2n+1)xX_n(x) + nX_{n-1}(x) = 0$, по

которой с учетом того, что $X_0 = 1$, $X_1 = x$ может быть найден многочлен Лежандра любой степени. В частности:

$$X_2(x) = \frac{1}{2}(3x^2 - 1), \;\; X_3(x) = \frac{1}{2}(5x^3 - 3x), \;\; X_4(x) = \frac{1}{8}(35x^4 - 30x^2 + 3).$$

3) Все корни многочлена Лежандра действительные, простые (однократные) и расположены в интервале $(-1;1)$

Интерполяционный многочлен Лагранжа

Пусть известны значения некоторой функции f в $n+1$ различных точках $x_0, x_1, ..., x_n$, которые обозначим следующим образом: $f_i = f(x_i)$, $i = 0,1,...,n$.

Для решения задачи приближенного восстановления функции f в произвольной точке x строится алгебраический многочлен $L_n(x)$ степени n, который в точках x_i принимает заданные значения, т.е. полином

$$L_n(x_i) = f_i, \;\; i = 0,1,...,n \tag{39}$$

и называется интерполяционным. Точки x_i называются узлами интерполяции.

Приближенное восстановление функции f по формуле $f(x) \approx L_n(x)$ называется интерполяцией функции f. Если же x расположен вне минимального отрезка, содержащего все узлы интерполяции $x_0, x_1, ..., x_n$, то замену функции f по этой формуле называют экстраполяцией.

Существует единственный интерполяционный многочлен n-ой степени, удовлетворяющий условиям (39), который ищется в виде:

$$L_n(x) = \sum_{i=0}^{n} p_{ni}(x) f_i, \tag{40}$$

где

$$p_{ni}(x) = \frac{(x - x_0)...(x - x_{i-1})(x - x_{i+1})...(x - x_n)}{(x_i - x_0)...(x_i - x_{i-1})(x_i - x_{i+1})...(x_i - x_n)}, \;\; i = 0, \; 1, \; ..., \; n \text{ (41).}$$

Интерполяционный многочлен, представленный в виде (40), называется интерполяционным многочленом Лагранжа, а функции (многочлены) вида (41) – лагранжевыми коэффициентами.