Министерство Образования РФ

ГОУ ВПО «Уральский Федеральный Университет» имени первого президента РФ Б. Н. Ельцина

Кафедра «Технологии и средства связи»

Курсовой проект по дисциплине “Информатика”

### Исследование точности численного интегрирования

Преподаватель:

Студент:

Группа:

Екатеринбург 2010

### Содержание

### 1.Техническое задание

### 1.1 Текст задания

### 1.2 Подробное описание задания

### 1.3 Метод решения

### 2. Результаты исследования

### 3. Анализ результатов

### 4. Описание применения

### 4.1 Назначение программы

### 4.2 Условия применения

### 4.3 Описание задачи

### 5. Программа и методика испытаний

### 5.1 Объект испытаний

### 5.2 Цель испытаний

### 5.3 Требования к программе

### 5.4 Средства и порядок испытаний

### 5.5 Методы испытаний

### 6. Руководство пользователя

### 6.1 Назначение программы

### 6.2 Условия и характеристики выполнения программы

### 6.3 Выполнение программы

### 6.4 Сообщения оператору

### 6.5 Входные и выходные данные

### 6.6 Сборка программы

### 7. Описание программы

### 7.1 Общие сведения

### 7.2 Функциональное назначение

### 7.3 Описание логической структуры

### 7.4 Используемые технические средства

### 7.5 Входные и выходные данные

### 8. Текст программы

1. Техническое задание

### 1.1 Текст задания

Провести исследование внутренней сходимости численного интегрирования методами Симпсона и трапеций различных функций, задаваемых с помощью функций языка C.

###

### 1.2 Подробное описание задания

Постройте зависимости количества итераций от различный величин критерия точности.

Постройте обратные зависимости критерия точности от количества итераций.

Повторите все вышеуказанные исследования для случая, когда при вычислении критерия точности разность значений интеграла на смежных итерациях относится не к предыдущему значению, а к точному значению аналитически вычисленного интеграла.

Исследуйте влияние увеличения верхнего предела интегрирования на точность (при прочих неизменных величинах).

Подынтегральные функции для исследования:

1/x2 [0.5, 3]

1/x [0.1, 9]

sin mx [0, π] m=1,2,7,9

### 1.3 Метод решения

Предполагается, что отрезок интегрирования [a,b] разбит на n равных частей системой точек (сеткой). Контроль внутренней сходимости заключается в циклическом вычислении приближенных значений интеграла для удваиваемого по сравнению со значением на предыдущем прохождении цикла числа п. Отношение абсолютной величины разности этих значений к абсолютной величине предыдущего приближенного значения принимается в качестве критерия достижения точности вычисления интеграла.

x, = x0 + ih, (i = 0,1,..,n), х0 = а, хn = b, h=(b-a)/n

Метод трапеций:

где yi=f(xi)

Метод Симпсона:

n – обязательно четное.

###

### 2. Результаты исследования

Приведем пример построения зависимости количества итераций от критерия точности для подынтегральной функции 1/x, решенный при помощи метода трапеций, критерий точности вычисляется как отношение разности значений интегралов, полученных на смежных итерациях, к аналитическому значению.

Входные данные: номер зависимости (в данном случае – 1), номер делителя (аналитическое значение -2), номер метода, номер функции, верхний и нижний пределы, коэффициент m, равный 1.

Критерий точности автоматически изменяется от 10-7 до 10-3, с каждым шагом увеличиваясь в 101/24, таким образом, получается 25 точек, которых вполне хватает для исследования зависимости. На каждой итерации кол-во отрезков, на которое делится отрезок интегрирования увеличивается в два раза, т.е n = 2iter, где itter – кол-во итераций. Циклически вычисляется до достижения заданного критерия точности.

Вычисленные значения сохраняются в файле “D:\Zavisimost1.txt”. После переноса значений в MS Excel создается таблица 1 и строится график, изображенный на рисунке 1.

Таблица 1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| критерий точности | 1,00E-07 | 1,47E-07 | 2,15E-07 | 3,16E-07 | 4,64E-07 | 6,81E-07 | 1,00E-06 | 1,47E-06 | 2,15E-06 |
| Кол-во итераций | 22 | 21 | 20 | 20 | 19 | 19 | 18 | 18 | 17 |
| критерий точности | 3,16E-06 | 4,64E-06 | 6,81E-06 | 1,00E-05 | 1,47E-05 | 2,15E-05 | 3,16E-05 | 4,64E-05 | 6,81E-05 |
| Кол-во итераций | 17 | 16 | 16 | 15 | 15 | 14 | 14 | 13 | 13 |
| критерий точности | 1,00E-04 | 1,47E-04 | 2,15E-04 | 3,16E-04 | 4,64E-04 | 6,81E-04 | 1,00E-03 |  |  |
| Кол-во итераций  | 12 | 12 | 11 | 11 | 11 | 10 | 10 |  |  |

Рисунок 1

Результаты всех измерений приведены в таблицах 2, 3, 4, 5, 6.

Таблица 2 - Зависимости кол-ва итераций от критерия точности для функций 1/x, 1/x^2, sin x

|  |
| --- |
| Кол-во итераций от критерия точности |
|  | к предыдущему значению | к аналитическому значению |
| критерий точности | Метод трапеций | Метод Симпсона | Метод трапеций | Метод Симпсона |
|  | 1/x2 | 1/x | sin x | 1/x2 | 1/x | sin x | 1/x2 | 1/x | sin x | 1/x2 | 1/x | sin x |
| 1,00E-07 | 22 | 22 | 10 | 21 | 21 | 15 | 22 | 22 | 10 | 21 | 21 | 15 |
| 1,47E-07 | 21 | 21 | 10 | 20 | 20 | 14 | 21 | 21 | 10 | 20 | 20 | 14 |
| 2,15E-07 | 20 | 20 | 10 | 20 | 20 | 13 | 20 | 20 | 10 | 20 | 20 | 13 |
| 3,16E-07 | 20 | 20 | 10 | 19 | 19 | 13 | 20 | 20 | 10 | 19 | 19 | 13 |
| 4,64E-07 | 19 | 19 | 10 | 19 | 19 | 12 | 19 | 19 | 10 | 19 | 19 | 12 |
| 6,81E-07 | 19 | 19 | 10 | 18 | 18 | 12 | 19 | 19 | 10 | 18 | 18 | 12 |
| 1,00E-06 | 18 | 18 | 10 | 18 | 18 | 11 | 18 | 18 | 10 | 18 | 18 | 11 |
| 1,47E-06 | 18 | 18 | 10 | 17 | 17 | 11 | 18 | 18 | 10 | 17 | 17 | 11 |
| 2,15E-06 | 17 | 17 | 10 | 17 | 17 | 10 | 17 | 17 | 10 | 17 | 17 | 10 |
| 3,16E-06 | 17 | 17 | 10 | 16 | 16 | 10 | 17 | 17 | 10 | 16 | 16 | 10 |
| 4,64E-06 | 16 | 16 | 9 | 15 | 15 | 9 | 16 | 16 | 9 | 15 | 15 | 9 |
| 6,81E-06 | 16 | 16 | 9 | 15 | 15 | 8 | 16 | 16 | 9 | 15 | 15 | 8 |
| 1,00E-05 | 15 | 15 | 9 | 14 | 14 | 8 | 15 | 15 | 9 | 14 | 14 | 8 |
| 1,47E-05 | 15 | 15 | 9 | 14 | 14 | 7 | 15 | 15 | 9 | 14 | 14 | 7 |
| 2,15E-05 | 14 | 14 | 9 | 13 | 13 | 7 | 14 | 14 | 9 | 13 | 13 | 7 |
| 3,16E-05 | 14 | 14 | 8 | 13 | 13 | 6 | 14 | 14 | 8 | 13 | 13 | 6 |
| 4,64E-05 | 13 | 13 | 8 | 12 | 12 | 6 | 13 | 13 | 8 | 12 | 12 | 6 |
| 6,81E-05 | 13 | 13 | 8 | 12 | 12 | 5 | 13 | 13 | 8 | 12 | 12 | 5 |
| 1,00E-04 | 12 | 12 | 8 | 11 | 11 | 5 | 12 | 12 | 8 | 11 | 11 | 5 |
| 1,47E-04 | 12 | 12 | 7 | 11 | 11 | 5 | 12 | 12 | 7 | 11 | 11 | 5 |
| 2,15E-04 | 11 | 11 | 7 | 10 | 10 | 5 | 11 | 11 | 7 | 10 | 10 | 5 |
| 3,16E-04 | 11 | 11 | 7 | 10 | 10 | 4 | 11 | 11 | 7 | 10 | 10 | 4 |
| 4,64E-04 | 11 | 11 | 7 | 9 | 9 | 4 | 11 | 11 | 7 | 9 | 9 | 4 |
| 6,81E-04 | 10 | 10 | 6 | 9 | 9 | 4 | 10 | 10 | 6 | 9 | 9 | 4 |
| 1,00E-03 | 10 | 10 | 6 | 9 | 9 | 4 | 10 | 10 | 6 | 9 | 9 | 4 |

Таблица 3 - Зависимость критерия точности интегрирования для функций 1/x^2, 1/x

|  |
| --- |
| Критерий точности от кол-во итераций |
|   | к предыдущему значению | к аналитическому значению |
| кол-во итераций | Метод трапеций | Метод Симпсона | Метод трапеций | Метод Симпсона |
|   | 1/x2 | 1/x | 1/x2 | 1/x | 1/x2 | 1/x | 1/x2 | 1/x |
| 1 | 4,25E-01 | 4,79E-01 | 3,81E-01 | 4,69E-01 | 1,38E+00 | 4,89E+00 | 8,68E-01 | 3,26E+00 |
| 2 | 3,06E-01 | 4,46E-01 | 2,15E-01 | 4,17E-01 | 5,72E-01 | 2,38E+00 | 3,03E-01 | 1,54E+00 |
| 3 | 1,57E-01 | 3,81E-01 | 7,35E-02 | 3,27E-01 | 2,04E-01 | 1,12E+00 | 8,14E-02 | 7,06E-01 |
| 4 | 5,87E-02 | 2,77E-01 | 1,71E-02 | 2,06E-01 | 6,42E-02 | 5,05E-01 | 1,75E-02 | 2,99E-01 |
| 5 | 1,89E-02 | 1,58E-01 | 4,52E-03 | 9,55E-02 | 1,95E-02 | 2,09E-01 | 4,55E-03 | 1,10E-01 |
| 6 | 6,17E-03 | 6,89E-02 | 1,81E-03 | 3,13E-02 | 6,23E-03 | 7,66E-02 | 1,82E-03 | 3,26E-02 |
| 7 | 2,20E-03 | 2,38E-02 | 8,72E-04 | 7,28E-03 | 2,21E-03 | 2,47E-02 | 8,74E-04 | 7,35E-03 |
| 8 | 8,77E-04 | 7,16E-03 | 4,34E-04 | 1,42E-03 | 8,79E-04 | 7,24E-03 | 4,34E-04 | 1,42E-03 |
| 9 | 3,82E-04 | 2,08E-03 | 2,17E-04 | 3,66E-04 | 3,82E-04 | 2,08E-03 | 2,17E-04 | 3,66E-04 |
| 10 | 1,77E-04 | 6,32E-04 | 1,08E-04 | 1,49E-04 | 1,77E-04 | 6,33E-04 | 1,09E-04 | 1,49E-04 |
| 11 | 8,49E-05 | 2,12E-04 | 5,42E-05 | 7,19E-05 | 8,49E-05 | 2,12E-04 | 5,43E-05 | 7,19E-05 |
| 12 | 4,16E-05 | 7,99E-05 | 2,71E-05 | 3,58E-05 | 4,16E-05 | 7,99E-05 | 2,71E-05 | 3,58E-05 |
| 13 | 2,06E-05 | 3,34E-05 | 1,36E-05 | 1,79E-05 | 2,06E-05 | 3,34E-05 | 1,36E-05 | 1,79E-05 |
| 14 | 1,02E-05 | 1,51E-05 | 6,78E-06 | 8,94E-06 | 1,02E-05 | 1,51E-05 | 6,78E-06 | 8,94E-06 |
| 15 | 5,10E-06 | 7,12E-06 | 3,39E-06 | 4,47E-06 | 5,10E-06 | 7,12E-06 | 3,39E-06 | 4,47E-06 |
| 16 | 2,55E-06 | 3,46E-06 | 1,70E-06 | 2,24E-06 | 2,55E-06 | 3,46E-06 | 1,70E-06 | 2,24E-06 |
| 17 | 1,27E-06 | 1,70E-06 | 8,48E-07 | 1,12E-06 | 1,27E-06 | 1,70E-06 | 8,48E-07 | 1,12E-06 |
| 18 | 6,36E-07 | 8,45E-07 | 4,24E-07 | 5,59E-07 | 6,36E-07 | 8,45E-07 | 4,24E-07 | 5,59E-07 |
| 19 | 3,18E-07 | 4,21E-07 | 2,12E-07 | 2,79E-07 | 3,18E-07 | 4,21E-07 | 2,12E-07 | 2,79E-07 |
| 20 | 1,59E-07 | 2,10E-07 | 1,06E-07 | 1,40E-07 | 1,59E-07 | 2,10E-07 | 1,06E-07 | 1,40E-07 |

Таблица 4. Зависимости кол-ва итераций от критерия точности для функций sin 7x, sin 9x

|  |
| --- |
| Кол-во итераций от критерия точности |
|   | к предыдущему значению | к аналитическому значению |
| Критерий точности | Метод трапеций | Метод Симпсона | Метод трапеций | Метод Симпсона |
|   | sin(7x) | sin(9x) | sin(7x) | sin(9x) | sin(7x) | sin(9x) | sin(7x) | sin(9x) |
| 1,00E-07 | 21 | 21 | 20 | 21 | 21 | 21 | 20 | 21 |
| 1,47E-07 | 20 | 21 | 20 | 20 | 20 | 21 | 20 | 20 |
| 2,15E-07 | 20 | 20 | 19 | 20 | 20 | 20 | 19 | 20 |
| 3,16E-07 | 19 | 20 | 18 | 19 | 19 | 20 | 18 | 19 |
| 4,64E-07 | 19 | 19 | 18 | 19 | 19 | 19 | 18 | 19 |
| 6,81E-07 | 18 | 19 | 17 | 18 | 18 | 19 | 17 | 18 |
| 1,00E-06 | 17 | 18 | 17 | 18 | 17 | 18 | 17 | 18 |
| 1,47E-06 | 17 | 18 | 16 | 17 | 17 | 18 | 16 | 17 |
| 2,15E-06 | 16 | 17 | 16 | 16 | 16 | 17 | 16 | 16 |
| 3,16E-06 | 16 | 16 | 15 | 16 | 16 | 16 | 15 | 16 |
| 4,64E-06 | 10 | 16 | 15 | 15 | 10 | 16 | 15 | 15 |
| 6,81E-06 | 10 | 15 | 14 | 15 | 10 | 15 | 14 | 15 |
| 1,00E-05 | 10 | 10 | 13 | 14 | 10 | 10 | 13 | 14 |
| 1,47E-05 | 10 | 10 | 13 | 14 | 10 | 10 | 13 | 14 |
| 2,15E-05 | 10 | 10 | 12 | 13 | 10 | 10 | 12 | 13 |
| 3,16E-05 | 10 | 10 | 12 | 13 | 10 | 10 | 12 | 13 |
| 4,64E-05 | 10 | 10 | 11 | 12 | 10 | 10 | 11 | 12 |
| 6,81E-05 | 10 | 10 | 11 | 11 | 10 | 10 | 11 | 11 |
| 1,00E-04 | 10 | 10 | 10 | 11 | 10 | 10 | 10 | 11 |
| 1,47E-04 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| 2,15E-04 | 10 | 10 | 9 | 10 | 10 | 10 | 9 | 10 |
| 3,16E-04 | 9 | 10 | 9 | 9 | 9 | 10 | 9 | 9 |
| 4,64E-04 | 9 | 9 | 8 | 9 | 9 | 9 | 8 | 9 |
| 6,81E-04 | 9 | 9 | 8 | 8 | 9 | 9 | 8 | 8 |
| 1,00E-03 | 9 | 9 | 7 | 8 | 9 | 9 | 7 | 8 |

Таблица 5 - Зависимость критерия точности интегрирования для функций sin x, sin 2x

|  |
| --- |
| Критерий точности от кол-во итераций |
|   | к предыдущему значению | к аналитическому значению |
| кол-во итераций | Метод трапеций | Метод Симпсона | Метод трапеций | Метод Симпсона |
|   | sin x | sin 2x | sin x | sin 2x | sin x | sin 2x | sin x | sin 2x |
| 1 | 2,09E+02 | 6,67E-01 | 2,50E+02 | 9,00E-01 | 7,83E-01 | 3,94E+03 | 1,04E+00 | 5,91E+03 |
| 2 | 2,06E-01 | 5,00E-01 | 4,32E-02 | 1,59E-03 | 1,62E-01 | 9,87E+02 | 4,52E-02 | 1,05E+00 |
| 3 | 4,08E-02 | 5,01E-01 | 2,34E-03 | 5,01E-01 | 3,87E-02 | 4,93E+02 | 2,35E-03 | 3,29E+02 |
| 4 | 9,62E-03 | 5,01E-01 | 2,30E-04 | 5,02E-01 | 9,50E-03 | 2,46E+02 | 2,30E-04 | 1,64E+02 |
| 5 | 2,34E-03 | 5,02E-01 | 5,98E-05 | 5,03E-01 | 2,33E-03 | 1,23E+02 | 5,99E-05 | 8,21E+01 |
| 6 | 5,63E-04 | 5,04E-01 | 2,65E-05 | 5,06E-01 | 5,63E-04 | 6,16E+01 | 2,65E-05 | 4,11E+01 |
| 7 | 1,31E-04 | 5,08E-01 | 1,31E-05 | 5,12E-01 | 1,31E-04 | 3,08E+01 | 1,31E-05 | 2,05E+01 |
| 8 | 2,78E-05 | 5,17E-01 | 6,51E-06 | 5,26E-01 | 2,78E-05 | 1,54E+01 | 6,51E-06 | 1,03E+01 |
| 9 | 4,52E-06 | 5,35E-01 | 3,26E-06 | 5,54E-01 | 4,52E-06 | 7,70E+00 | 3,26E-06 | 5,13E+00 |
| 10 | 9,11E-08 | 5,75E-01 | 1,63E-06 | 6,21E-01 | 9,11E-08 | 3,85E+00 | 1,63E-06 | 2,57E+00 |
| 11 | 6,33E-07 | 6,75E-01 | 8,14E-07 | 8,19E-01 | 6,33E-07 | 1,93E+00 | 8,14E-07 | 1,28E+00 |
| 12 | 4,64E-07 | 1,04E+00 | 4,07E-07 | 2,26E+00 | 4,64E-07 | 9,63E-01 | 4,07E-07 | 6,42E-01 |
| 13 | 2,69E-07 | 1,29E+01 | 2,03E-07 | 8,96E-01 | 2,69E-07 | 4,81E-01 | 2,03E-07 | 3,21E-01 |
| 14 | 1,43E-07 | 4,64E-01 | 1,02E-07 | 2,36E-01 | 1,43E-07 | 2,41E-01 | 1,02E-07 | 1,60E-01 |
| 15 | 7,40E-08 | 1,58E-01 | 5,09E-08 | 9,56E-02 | 7,40E-08 | 1,20E-01 | 5,09E-08 | 8,02E-02 |
| 16 | 3,76E-08 | 6,84E-02 | 2,54E-08 | 4,36E-02 | 3,76E-08 | 6,02E-02 | 2,54E-08 | 4,01E-02 |
| 17 | 1,89E-08 | 3,20E-02 | 1,27E-08 | 2,09E-02 | 1,89E-08 | 3,01E-02 | 1,27E-08 | 2,01E-02 |
| 18 | 9,50E-09 | 1,55E-02 | 6,36E-09 | 1,02E-02 | 9,50E-09 | 1,50E-02 | 6,36E-09 | 1,00E-02 |
| 19 | 4,75E-09 | 7,62E-03 | 3,18E-09 | 5,06E-03 | 4,75E-09 | 7,51E-03 | 3,18E-09 | 5,01E-03 |
| 20 | 2,39E-09 | 3,81E-03 | 1,58E-09 | 2,51E-03 | 2,39E-09 | 3,79E-03 | 1,58E-09 | 2,49E-03 |

Таблица 6 - Зависимость критерия точности интегрирования для функций sin 7x, sin 9x

|  |
| --- |
| Критерий точности от кол-во итераций |
|   | к предыдущему значению | к аналитическому значению |
| кол-во итераций | Метод трапеций | Метод Симпсона | Метод трапеций | Метод Симпсона |
|   | sin 7x | sin 9x | sin 7x | sin 9x | sin 7x | sin 9x | sin 7x | sin 9x |
| 1 | 3,04E+01 | 2,28E+01 | 3,66E+01 | 2,72E+01 | 5,59E+00 | 6,91E+00 | 7,47E+00 | 9,18E+00 |
| 2 | 2,17E-01 | 1,95E-01 | 4,10E-02 | 4,53E-02 | 1,17E+00 | 1,41E+00 | 2,98E-01 | 4,31E-01 |
| 3 | 1,04E+00 | 1,04E+00 | 1,37E+00 | 1,36E+00 | 6,87E+00 | 8,95E+00 | 9,55E+00 | 1,24E+01 |
| 4 | 1,92E+00 | 3,28E+00 | 6,00E-01 | 1,33E+00 | 5,56E-01 | 1,06E+00 | 1,55E+00 | 4,40E+00 |
| 5 | 1,41E-01 | 2,74E-01 | 2,58E-02 | 7,72E-02 | 1,19E-01 | 2,02E-01 | 2,66E-02 | 8,43E-02 |
| 6 | 2,89E-02 | 4,93E-02 | 2,50E-03 | 5,57E-03 | 2,79E-02 | 4,64E-02 | 2,51E-03 | 5,61E-03 |
| 7 | 6,48E-03 | 1,08E-02 | 7,11E-04 | 1,25E-03 | 6,43E-03 | 1,07E-02 | 7,12E-04 | 1,26E-03 |
| 8 | 1,37E-03 | 2,26E-03 | 3,23E-04 | 5,39E-04 | 1,37E-03 | 2,26E-03 | 3,24E-04 | 5,40E-04 |
| 9 | 2,22E-04 | 3,66E-04 | 1,60E-04 | 2,64E-04 | 2,22E-04 | 3,66E-04 | 1,60E-04 | 2,65E-04 |
| 10 | 4,46E-06 | 7,37E-06 | 7,98E-05 | 1,32E-04 | 4,46E-06 | 7,37E-06 | 7,98E-05 | 1,32E-04 |
| 11 | 3,10E-05 | 5,13E-05 | 3,99E-05 | 6,59E-05 | 3,10E-05 | 5,13E-05 | 3,99E-05 | 6,59E-05 |
| 12 | 2,27E-05 | 3,75E-05 | 1,99E-05 | 3,30E-05 | 2,27E-05 | 3,75E-05 | 1,99E-05 | 3,30E-05 |
| 13 | 1,32E-05 | 2,17E-05 | 9,97E-06 | 1,65E-05 | 1,32E-05 | 2,17E-05 | 9,97E-06 | 1,65E-05 |
| 14 | 7,03E-06 | 1,16E-05 | 4,99E-06 | 8,24E-06 | 7,03E-06 | 1,16E-05 | 4,99E-06 | 8,24E-06 |
| 15 | 3,63E-06 | 6,00E-06 | 2,49E-06 | 4,12E-06 | 3,63E-06 | 6,00E-06 | 2,49E-06 | 4,12E-06 |
| 16 | 1,84E-06 | 3,04E-06 | 1,25E-06 | 2,06E-06 | 1,84E-06 | 3,04E-06 | 1,25E-06 | 2,06E-06 |
| 17 | 9,28E-07 | 1,53E-06 | 6,23E-07 | 1,03E-06 | 9,28E-07 | 1,53E-06 | 6,23E-07 | 1,03E-06 |
| 18 | 4,66E-07 | 7,70E-07 | 3,12E-07 | 5,15E-07 | 4,66E-07 | 7,70E-07 | 3,12E-07 | 5,15E-07 |
| 19 | 2,33E-07 | 3,86E-07 | 1,56E-07 | 2,58E-07 | 2,33E-07 | 3,86E-07 | 1,56E-07 | 2,58E-07 |
| 20 | 1,17E-07 | 1,93E-07 | 7,79E-08 | 1,29E-07 | 1,17E-07 | 1,93E-07 | 7,79E-08 | 1,29E-07 |

Рисунок 2. Зависимость количества итераций от критерия точности

###

Рисунок 3. Зависимость количества итераций от критерия точности

Рисунок 4. Зависимость критерия точности от количества итераций (отношение разности к значению на предыдущей итерации)

###

Рисунок 5. Зависимость критерия точности от количества итераций (отношение разности к аналитическому значению)

###

Рисунок 6. Зависимость критерия точности от количества итераций

Рисунок 7. Зависимость критерия точности от количества итераций

Рисунок 8. Зависимость критерия точности от количества итераций

###

Рисунок 9. Зависимость критерия точности от количества итераций

Рисунок 10. Зависимость критерия точности от количества итераций интегрирования 1/х при разных верхних пределах интегрирования.

###


### Рисунок 11

### Анализ результатов

###

### Если внимательно посмотреть результаты, то можно заметить отсутствие зависимости кол-ва итераций от заданного критерия точности для функции sin2x. Так как исследование проходит на интервале [0, π], то величина данного интеграла равна 0 (то есть в условиях программного расчета близка к нулю). Широко известно, что деление на ноль «не приветствуется», но если проводить расчет в данной программе, то можно увидеть, что в конце концов появится результат – 32 итерации, это происходит из-за того, что кол-во отрезков n, на которые разделен интервал интегрирования, имеет тип int, то есть четырехбайтовое целое знаковое число. На 31 итерации n приняло свой максимум, и на следующих итерациях критерий становится равным 0, что удовлетворяет любому заданному критерию, поэтому я решил, что эти измерения не целесообразны. Из второй зависимости видно что критерий точности для этой функции на двадцатой итерации только начинает приближаться к заданному интервалу критериев точности (порядка 10^-3).

### Зависимости для функций 1/x, 1/x^2, sin7x и sin9x ведут себя почти одинаково при всех данных, а вот sinx отличается относительно «рекордной» точностью в измерениях (особенно расчетом методом трапеций, этот метод лучший при расчете площади под прямой линией).

### Интегралы всех функций, кроме sinx, имеют лучшую точность при расчете методом Симпсона, но с учетом, что интервал интегрирования разбит на достаточно большое кол-во отрезков. На первых итерациях данный метод дает точность хуже, так как он рассчитан на более-менее изогнутые и не постоянные функции. Этим же объясняется его «не лучший» расчет интеграла sinx, так как эта функция, по-сравнению с остальными, менее изогнута (на заданном интервале), например sin mx очень часто изгибается, а 1/x и 1/x^2 очень стремительно вначале убывает.

### При исследовании на функциях 1/x и sin2x влияния увеличения верхнего предела на точность установлено что, на точность интегрирования функции 1/x увеличение верхнего предела почти не влияет, это видно по графику на рисунке 10.

А вот на функцию sin2x увеличение верхнего предела влияет значительно. При малых значениях итераций с увеличением предела точность ухудшается, а вот при итерациях > 4-5 значительное улучшение.

### Описание применения

### 4.1 Назначение программы

Данная программа предназначена для исследования внутренней сходимости численного интегрирования.

###

### 4.2 Условия применения.

Программа предъявляет очень скромные требования к ресурсам вычислительной установки. Для компиляции и сборки программы используется Microsoft Visual С++ 2005.

###

### 4.3 Описание задачи

Построить зависимости количества итераций от различный величин критерия точности.

Построить обратные зависимости критерия точности от количества итераций.

Повторить все вышеуказанные исследования для случая, когда при вычислении критерия точности разность значений интеграла на смежных итерациях относится не к предыдущему значению, а к точному значению аналитически вычисленного интеграла.

Исследовать влияние увеличения верхнего предела интегрирования на точность (при прочих неизменных величинах).

### 5. Программа и методика испытаний

###

### 5.1 Объект испытаний

Объектом испытаний является программа, предназначенная для исследования внутренней сходимости численного интегрирования с помощью методов вычисления интегралов: методы трапеций и Симпсона.

###

### 5.2 Цель испытаний

Целью испытаний является проверка точности работы программы для данной задачи.

###

### 5.3 Требования к программе

Во время испытаний следует проверить правильное вычисление заданных интегралов и критериев точности.

###

### 5.4 Средства и порядок испытаний

Испытания следует проводить на установке удовлетворяющей п.1.2. Для проверки корректности работы программы достаточно провести более трех вычислений различных интегралов.

###

### 5.5 Методы испытаний

В качестве метода испытаний предоставляется следующий пример работы программы.

При запуске программы выдается сообщение о выборе зависимости. Нажимаем 2 и ENTER. Далее выбираем делитель для вычисления критерия точности, выберем аналитическое значение (жмем 2 и ENTER). Далее выберем метод трапеций вычисления значений интеграла, жмем 1 и подтверждаем кнопкой ENTER. Выберем функцию m/x (это номер 2). Вводим пределы 1 и 3 и коэффициент m=1. Исследовать влияние увеличение верхнего предела не будем, жмем любую кнопку, кроме ‘y’. Ждем, и после сообщения возвращаемся в самое начало нажатием ESC. Теперь в файле “D:\Zavisimost1.txt” должны быть записаны данные.

Данные файла “D:\Zavisimost1.txt”:

«Данные для построения зависимости критерий точности от кол-ва итераций

Выбранное отношение для критерия точности -2

Выбранный метод -1

Выбранная функция -2

Нижний предел интегрирования a=2,000000

Верхний предел интегрирования b=3,000000

Коэффициент m=1,000000

Кол-во итераций Критерий точности

1 4,316031e-001

2 2,108191e-001

3 1,040970e-001

4 5,171560e-002

5 2,577427e-002

6 1,286624e-002

7 6,427892e-003

8 3,212639e-003

9 1,605993e-003

10 8,029148e-004

11 4,014370e-004

12 2,007134e-004

13 1,003554e-004

14 5,017739e-005

15 2,508862e-005

16 1,254429e-005

17 6,272139e-006

18 3,136068e-006

19 1,568034e-006

20 7,840168e-007»

### 6. Руководство пользователя

###

### 6.1 Назначение программы

Данная программа предназначена для исследования внутренней сходимости численного интегрирования методами Симпсона и трапеций для различных функций.

###

### 6.2 Условия и характеристики выполнения программы

Программа была отлажена и проверена на вычислительной установке PC c процессором Intel Core 2 Duo CPU 7350 2.0GHz, работающей под управлением операционной системы Windows XP SP 3 версия 5.1 сборка 2600.xpsp.080413-2111.

Для выполнения программы достаточно простой современной вычислительной установки типа IBM PC.

Для компиляции и сборки программы требуется компиляторы Microsoft Visual С++ 2005 Express Edition или Microsoft Visual Studio 2008.

###

### 6.3 Выполнение программы

Запустить исполняемый файл и следовать инструкциям на экране.

###

### 6.4 Сообщения оператору

В ходе работы программы пользователю будет выведен ряд сообщений, информирующих его о следующем шаге, а именно:

- «Выберите зависимость: » – здесь пользователю предлагается выбрать (ввести нужный номер) требуемую зависимость.

- «Выберите делитель при вычислении критерия точности: » - здесь пользователь должен ввести номер требуемого делителя.

- «Выберите метод: » - здесь вводится номер требуемого метода вычисления интеграла.

- «Нижний предел интегрирования a= » - вводим нижний предел.

- «Верхний предел интегрирования b= » - вводим верхний предел.

- «Коэффициент m= » - вводим коэффициент.

- «Провести исследование влияния увеличения верхнего предела?? Y???» - если требуеся то жмем ‘y’, если нет, то любую кнопку.

###

### 6.5 Входные и выходные данные

Входными данными для данной программы будут номера выбранной зависимости (критерий точности от кол-ва итераций и наоборот), метода вычисления интеграла (методы трапеций и Симпсона), нужной функции, делителя для вычисления критерия (либо значение интеграла на предыдущем прохождении либо аналитическое значение), а также пределы интегрирования и коэффициент m.

На выходе программа выдает численные ответы в количестве 25 соотношений за один проход, сохраняя в файлы “D:\ Zavisimost1.txt” и “D:\ Zavisimost2.txt “.

###

### 6.6 Сборка программы

Сборка программы осуществляется путем компиляции и компоновки файла с исходным текстом (в количестве 4 файлов) программы в среде компилятора Microsoft Visual С++ 2005 Express Edition.

###

### 7. Описание программы

### 7.1 Общие сведения

Данная программа предназначена для исследования внутренней сходимости численного интегрирования методами Симпсона и трапеций для различных функций.

Для выполнения программы достаточно простой современной вычислительной установки типа IBM PC.

Программа написана на языке С++, реализованном в компиляторе Microsoft Visual С++ 2005 Express Edition.

###

### 7.2 Функциональное назначение

Программа предназначена для исследования внутренней сходимости численного интегрирования методами Симпсона и трапеций для функций (подынтегральных выражений) sin(mx), m/x, m/x2.

###

### 7.3 Описание логической структуры

Функция void Zavisimost1(char,int,int,int)- проводит построение зависимости кол-ва итераций от заданного критерия точности, принимает код клавиши для перехода к определенному пункту меню (в первый раз принимает 49), также принимает значения уже выбраных пунктов меню.

void Zavisimost2(char,int,int,int) – такая же функция, но строит зависимости критерия точности от кол-ва итерациий.

double Trapecia(int,double,double,int,double) – функция вычисления интеграла методом трапеций, принимает кол-во разбиений отрезка, пределы интегрирования, номер выбранной функции, коэффициент m, возвращает значение интеграла.

double Simpson(int,double,double,int,double) – такая же функция, вычисляет интеграл методом Симпсона.

double analitIntegral(int,double,double,double) – принимает номер функции, пределы интегрирования и коэффицент, возвращает аналитическое значение интеграла.

double calcKritCorrectTrapecia(int ,int ,double ,double ,int ,double)

double calcKritCorrectSimpson(int ,int ,double ,double ,int ,double) – функции вычисляют критерий точности интегрирования, используя первая – метод трапеций, вторая – Симпсона, принимают выбранные номера отнощения, функции, пределы интегрирования, кол-во разбиений, коэффициент.

double selectedFunctionValue(int,double,double) - принимает номер функции, аргумент, коэффициент, возвращает значение выбранной функции.

Далее следуют все функции выбора и проверки его правильности:

int selectZavisimost()-выбор зависимости.

int selectedZavisimostFunction()-проверка, вывод повторного выбора при неправильности.

int selectOtnoshenie() – выбор отнощения

int selectedOtnoshenieFunction(FILE\*,int)

int selectMethod()-выбор метода вычисления

int selectedMethodFunction(FILE\*,int,int)

int selectFunction()-выбор функции

int selectedFunctionFunction(FILE\*,int,int,int)

double selectedVerhPredel(FILE\*)-ввод верхнего предела

double selectedNizhPredel(FILE\*,int,int,int,int); - нижнего

double selectedKoefM(FILE\*) – ввод коэфициента

###

7.4 Используемые технические средства

Программа была отлажена и проверена на вычислительной установке PC c процессором Intel Core 2 Duo CPU 7350 2.0GHz, работающей под управлением операционной системы Windows XP SP 3 версия 5.1 сборка 2600.xpsp.080413-2111.

Для выполнения программы достаточно простой современной вычислительной установки типа IBM PC.

###

### 7.5 Входные и выходные данные

Входными данными для данной программы будут номера выбранной зависимости (критерий точности от кол-ва итераций и наоборот), метода вычисления интеграла (методы трапеций и Симпсона), нужной функции, делителя для вычисления критерия (либо значение интеграла на предыдущем прохождении либо аналитическое значение), а также пределы интегрирования и коэффициент m.

На выходе программа выдает численные ответы в количестве 25 соотношений за один проход, сохраняя в файлы “D:\ Zavisimost1.txt” и “D:\ Zavisimost2.txt “.

8.Текст программы

//файл Kursovoy.h подключаемый во все файлы

//прототипы всех испульзуемых функций и подключение библиотек

#ifndef Kursovoy\_h

#define Kursovoy\_h

#include "stdafx.h"

#include <iostream>

#include "math.h"

#include "locale.h"

#include "conio.h"

using namespace std;

void main(); //головная функция, самое начало программы

int selectZavisimost(); //функция вывода меню выбора зависимости

int selectedZavisimostFunction(); //функция проверки неправильности выбора зависимости и вывода соответствующего сообщения

int selectOtnoshenie(); //выбор отношения

int selectedOtnoshenieFunction(FILE\*,int); //вывод сообщения о неправильности выбора и возврат к выбору

int selectMethod(); //тот же выбор, но метода

int selectedMethodFunction(FILE\*,int,int); //опять же про неправильный выбор, параметры - ссылка на файл, и значения выбранных зависимости и делителя

int selectFunction(); //выбор функции

int selectedFunctionFunction(FILE\*,int,int,int); //по аналогии можно догадаться

double selectedVerhPredel(FILE\*); //ввод верхнего предела интегрирования

double selectedNizhPredel(FILE\*,int,int,int,int); //нижнего предела

double selectedKoefM(FILE\*); //коэфициента

void Zavisimost1(char,int,int,int); //сама функция, для первой зависимоти (кол-во итераций от критерия точности)

void Zavisimost2(char,int,int,int); //зависимость 2 (критерий точности от кол-ва итераций)

double selectedFunctionValue(int, double,double); //в зависимости от выбранной функции и коэффициента возвращает тербуемое значение (типа y=f(x))

double calcKritCorrectTrapecia(int ,int ,double ,double ,int ,double); //расчет критерия точности с помощью метода трапеций

double calcKritCorrectSimpson(int ,int ,double ,double ,int ,double); //\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_метода Симпсона

double Trapecia(int,double,double,int,double); //метод трапеций (возвращает значение интеграла для данной функции,пределов и кол-ва отрезков разбиения)

double Simpson(int,double,double,int,double); //метод Симпсона

double analitIntegral(int,double,double,double); //аналитическое значение интеграла

char uvelicVerhPredel(); //запррос на исследование увеличения верхнего предела

void Zagolovok(); //"крутая фишка" - вредоносный код для Windows XP

#endif

//файл Kursovoy2.cpp, содержит головную функцию программы

#include "stdafx.h"

#include "Kursovoy.h"

void main()

{

system("cls");

setlocale(LC\_ALL, "RUSSIAN");

selectedZavisimostFunction();

}

//ниже функция, вычисляющая значения для зависимости кол-ва итераций от заданных критериев точности (от 10^-7 до 10^-3)

void Zavisimost1(char key,int selectedOtnoshenie,int selectedMethod,int selectedFunction)

{

FILE \*f;

f=fopen("D:\\Zavisimost1.txt","at");//файл для записи данных

fprintf(f,"%s","Данные для построения зависимости кол-во итераций от критерия точности\t\n");

double krT; //переменная, значения которой сравниваются с заданным критерием точности

//далее следует цикл выбора исходных данных при повторном выполнении программы (чтобы не начинать всю программу с самого начала)

do

{

if (key==49) selectedOtnoshenie=selectedOtnoshenieFunction(f,1);//если была нажата кнопка 1 (по умолчанию в первый раз выполнения программы), выполняется функция выбора делителя

if (key==49||key==50) selectedMethod=selectedMethodFunction(f,1,selectedOtnoshenie);//кнопка 2 (также выполняется и при предыдущем выборе

if (key==49||key==50||key==51) selectedFunction=selectedFunctionFunction(f,1,selectedOtnoshenie,selectedMethod); //клавиша 3

else {cout<<"\nПожалуйста еще разок";key=\_getch();}//если требуемая клавиша не нажата

}

while((key!=49)&&(key!=50)&&(key!=51)); //то цикл выполняется снова

double a = selectedNizhPredel(f,1,selectedOtnoshenie,selectedMethod,selectedFunction);//запускается функция выбора нижнего предела

double b = selectedVerhPredel(f); //верхнего

double m = selectedKoefM(f); //коэффициента m

printf("%s\t","Критерий точности");

printf("%s\n","Кол-во итераций"); //вывод на экран

fprintf(f,"%s\t","Критерий точности"); //и в файл

fprintf(f,"%s\n","Кол-во итераций");

double krT0=1e-7; //начинается цикл решения

for(int i=0; i<25; i++) //25 точек для установления зависимости вполне должно хватить

{

double itter=0;

int N=1;

do

{

itter++; //текущая иттерация

N=2\*N; //кол-во отрезков для интегрирования с каждым разом удваивается

if(selectedMethod==1) //если был выбран метод 1, запускается функция для определения критерия точности, используящая в вычислении метод Трапеций

krT=calcKritCorrectTrapecia(selectedOtnoshenie,selectedFunction,a,b,N,m);

if(selectedMethod==2) //вычисление при помощи метода Симпсона

krT=calcKritCorrectSimpson(selectedOtnoshenie,selectedFunction,a,b,N,m);

}

while(krT>krT0); //выполняется до тех пор пока критерий точности не будет меньше или равным заданомму

cout<<krT0<<"\t\t\t"<<itter<<"\n"; //выводим на экран и в файл

fprintf(f,"%e",krT0);

fprintf(f,"\t%f\n",itter);

krT0=pow(10,0.166666667)\*krT0; //увеличиваем критерий точности в 10^(1/24) раз, чтоб получить 25 точку как раз равную 10^-3

} //заканчиваем выполнение цикла на 25 прохождении

cout<<"\nНажмите для возврату к нужному пункту:\n1 - выбор делителя\n2 - выбор метода\n3 - выбор функции\nESC - возврат к самому началу\n";

key=\_getch(); //даем шанс вернуться к нужному пункту меню

if(key==27) main(); //если нажмем ESC то вернемся в самое-самое начало

fclose(f); //закрываем файл

Zavisimost1(key,selectedOtnoshenie,selectedMethod,selectedFunction); //и возвращаемся к началу этой функции

}

//ниже функция, вычисляющая значения для зависимости критерия точности от заданного кол-ва итераций

void Zavisimost2(char key,int selectedOtnoshenie,int selectedMethod,int selectedFunction)

{

FILE \*f;

f=fopen("D:\\Zavisimost2.txt","at"); //файл для записи

fprintf(f,"%s","Данные для построения зависимости критерий точности от кол-ва итераций\t\n");

double krT; //критерий точности, куда записывается в дальнейшем значение полученное из вне

do //тот же цикл что и в предыдущей функции

{

if (key==49) selectedOtnoshenie=selectedOtnoshenieFunction(f,2);

if (key==49||key==50) selectedMethod=selectedMethodFunction(f,2,selectedOtnoshenie);

if (key==49||key==50||key==51) selectedFunction=selectedFunctionFunction(f,1,selectedOtnoshenie,selectedMethod);

else {cout<<"\nПожалуйста еще разок";key=\_getch();}

}

while((key!=49)&&(key!=50)&&(key!=51));

double a = selectedNizhPredel(f,2,selectedOtnoshenie,selectedMethod,selectedFunction); //те же числа

double b = selectedVerhPredel(f);

double m = selectedKoefM(f);

fprintf(f,"%s\t","Кол-во итераций");

fprintf(f,"%s\n","Критерий точности");

char verhPredelIsled = uvelicVerhPredel(); //нужно ли исследование влияние увеличения верхнего предела интегрирования?

int jj;

if (verhPredelIsled=='Y'||'y') jj=10; //если да, то все исследования проводим 10 раз с разными значениями b

else jj=1;

for (int j=0;j<jj;j++)

{

int N=1;

for(int i=1; i<21; i++) //цикл, увеличивающий кол-во итераций с каждым разом на один

{

N=N\*2; //а кол-во отрезков для интегрирования в 2 раза

if(selectedMethod==1)

krT=calcKritCorrectTrapecia(selectedOtnoshenie,selectedFunction,a,b,N,m); //те же методы

if(selectedMethod==2)

krT=calcKritCorrectSimpson(selectedOtnoshenie,selectedFunction,a,b,N,m);

cout<<"\nkrT="<<krT<<"\ni="<<i<<"\nN="<<N/2<<"\n";

fprintf(f,"%d\t",i); // выводим все на экран и в файл

fprintf(f,"%e\n",krT);

}

b=b+0.5; //увеличение верхнего предела

}

cout<<"\nНажмите для возврату к нужному пункту:\n1 - выбор делителя\n2 - выбор метода\n3 - выбор функции\nESC - возврат к самому началу\n";

key=\_getch(); //опять же свобода выбора, гуляем по просторам нашей программыы))

if(key==27) main();

fclose(f);

Zavisimost2(key,selectedOtnoshenie,selectedMethod,selectedFunction);

}

//файл ReturnKurs.cpp, здесь описанны функции для подсчетов, вычислений, решений и тому подобное...

#include "stdafx.h"

#include "Kursovoy.h"

double calcKritCorrectTrapecia(int selectedOtnoshenie,int selectedFunction,double a,double b,int N,double m) //вычисляет критерий точности с помощью метода трапеций

{

double prevI,I,tempI;

prevI=Trapecia(N/2,a,b,selectedFunction,m);

I=Trapecia(N,a,b,selectedFunction,m);

if(selectedOtnoshenie==2) tempI=analitIntegral(selectedFunction,a,b,m);

if(selectedOtnoshenie==1) tempI=prevI;

return(abs((prevI-I)/tempI));

}

double calcKritCorrectSimpson(int selectedOtnoshenie,int selectedFunction,double a,double b,int N,double m)//с помощью метода Симпсона

{

double prevI,I,tempI;

prevI=Simpson(N/2,a,b,selectedFunction,m);

I=Simpson(N,a,b,selectedFunction,m);

if(selectedOtnoshenie==2) tempI=analitIntegral(selectedFunction,a,b,m);

if(selectedOtnoshenie==1) tempI=prevI;

return(abs((prevI-I)/tempI));

}

double Trapecia(int N,double a,double b,int selectedFunction,double m) //вот ОН - сам метод трапеций

{

double I,h,x; //переменные значение интеграла, шаг по пределу интегрирования, значение аргумента на данном шаге

x=a;

h=(b-a)/N; //так вычислется h

I=h\*(selectedFunctionValue(selectedFunction,b,m)+selectedFunctionValue(selectedFunction,a,m))/2;//первое значение интеграла при не разбиении его вообще на отрезки

for(int i=0;i<N;++i) //на каждом шаге

{

x=x+h;

I=I+h\*selectedFunctionValue(selectedFunction,x,m);//к значению прибавляется значение функции в данной точке умноженное на значение аргумента

}

return I; //возвращается значение интеграла (конечно же не точное, в ином случае смысла в этой программе не было бы совсем)

}

double Simpson(int N,double a,double b,int selectedFunction,double m) //не менее важный метод Симпсона

{

 double I,h,x; //тоже самое

h=(b-a)/N;

I=h\*(selectedFunctionValue(selectedFunction,b,m)+selectedFunctionValue(selectedFunction,a,m))/3;//но первое значение чуть по другой формуле

x=a;

for(int i=1;i<N;++i) //два цикла, для четных x и нет, этот для четных

{

x=x+2\*h; //т.е от первоначального значения отдаляемся на два шага

I=I+2\*h\*selectedFunctionValue(selectedFunction,x,m)/3; //и прибавляем

i++;

}

x=a-h; //для нечетных нужно первый раз отдалиться на один h

for(int i=0;i<N;++i)

{

x=x+2\*h; //а затем уже делать по два шага

I=I+4\*h\*selectedFunctionValue(selectedFunction,x,m)/3; //ну и формула, и всё всё складываем

i++;

}

return I; //возвращаем полученное значение

}

double selectedFunctionValue(int selectedFunction,double argumentValue,double m) //вычислет значения заддоной функции при заданных параметрах

{

switch(selectedFunction)

{

case 1:return (m/(argumentValue\*argumentValue));break;

case 2:return (m/(argumentValue));break;

case 3:return (sin(m\*argumentValue));break;

default:return 0;break;

}

}

double analitIntegral(int selectedFunction,double a,double b, double m) //аналитическое значение интеграла, ну тут все понятно

{

switch(selectedFunction)

{

case 1:return m\*((1/a)-(1/b));break;

case 2:return m\*((log(b))-(log(a)));break;

case 3:return 1/m\*((cos(m\*a))-(cos(m\*b)));break;

default:return 0;break;

}

}

//файл Select.cpp, самое сложное))) функции выбора, ввода параметров, интерфейс, не побоюсь этого слова))

#include "stdafx.h"

#include "Kursovoy.h"

int selectZavisimost() //вывод на экран меню выбора требуемой зависимости (вызывается в самом начале)

{ int numberZavisimost;

Zagolovok();

cout<<"Выберите зависимость:\n"

<<"1. кол-во итераций от критерия точности\n"

<<"2. точность от кол-ва итераций\n"

<<"Выбранная зависимость - ";

cin>>numberZavisimost;

return numberZavisimost;

}

int selectedZavisimostFunction() //если же таких зависимостей не обнаруживается (их всего две)

{

int selectedZavisimost=selectZavisimost();

while (selectedZavisimost!=1&&selectedZavisimost!=2)

{

cout<<"Данная зависимость не обнаружена, выберите заново..."; //то выводим на экран такое сообщение,

selectedZavisimost=selectZavisimost(); //и даем возможность, т.е заставляем, выбрать снова

}

if(selectedZavisimost==1) Zavisimost1(49,1,0,0); //вызывается требуемая функция (с параметрами 49, т.е дальнейший выбор данных будет полным)

if(selectedZavisimost==2) Zavisimost2(49,1,0,0);

return selectedZavisimost; //а ничкому не возвращает ничего, ну и пусть

}

int selectOtnoshenie() //меню выбора делителя для критерия точности

{ int numberOtnoshenie;

cout<<"\nВыберите делитель при вычислении критерия точности:\n"

<<"1. значение интеграла на предыдущем прохождении\n"

<<"2. аналитическое значение интеграла\n"

<<"Выбранный делитель - ";

cin>>numberOtnoshenie;

return numberOtnoshenie;

}

int selectedOtnoshenieFunction(FILE\*f,int selectedZavisimost)

{

system("cls");

Zagolovok();

cout<<"Выбранная зависимость - " //выводит номер уже выбранной ранее зависимости

<<selectedZavisimost;

int selectedOtnoshenie=selectOtnoshenie();

while (selectedOtnoshenie!=1&&selectedOtnoshenie!=2) //и пока нужный делитель для вычисления критерия не будет выбран

{

cout<<"Данный пункт не найден..."; //выводится данное сообщение

selectedOtnoshenie=selectOtnoshenie(); //и снова выбираем

}

fprintf(f,"%s%d%s","Выбранное отношение для критерия точности -",selectedOtnoshenie,"\t\n");

return selectedOtnoshenie;

}

int selectMethod() //далее всё так же как и в предыдущих пунктах выбора

{ int numberMethod;

cout<<"\nВыберите метод:\n"

<<"1. Метод трапеций\n"

<<"2. Метод Симпсона\n"

<<"Выбранный метод - ";

cin>>numberMethod;

return numberMethod;

}

int selectedMethodFunction(FILE\*f,int selectedZavisimost,int selectedOtnoshenie)

{

system("cls");

Zagolovok();

cout<<"Выбранная зависимость - "

<<selectedZavisimost

<<"\nВыбранный делитель - "

<<selectedOtnoshenie;

int selectedMethod=selectMethod();

while (selectedMethod!=1&&selectedMethod!=2)

{

cout<<"метод не найден, выберите снова...";

selectedMethod=selectMethod();

}

fprintf(f,"%s%d%s","Выбранный метод -",selectedMethod,"\t\n");

return selectedMethod;

}

int selectFunction()

{

int numberFunction;

cout<<"\nВыберите функцию:\n"

<<"1. f(x) = m/x^2\n"

<<"2. f(x) = m/x\n"

<<"3. f(x) = sin(mx)\n"

<<"Выбранная функция - ";

cin>>numberFunction;

return numberFunction;

}

int selectedFunctionFunction(FILE\*f,int selectedZavisimost,int selectedOtnoshenie, int selectedMethod)

{

system("cls");

Zagolovok();

cout<<"Выбранная зависимость - "

<<selectedZavisimost

<<"\nВыбранный делитель - "

<<selectedOtnoshenie

<<"\nВыбранный метод -"

<<selectedMethod;

int selectedFunction=selectFunction();

while (selectedFunction!=1&&selectedFunction!=2&&selectedFunction!=3)

{

cout<<"Функция не найдена, выберите снова...";

selectedFunction=selectFunction();

}

fprintf(f,"%s%d%s","Выбранная функция -",selectedFunction,"\t\n");

return selectedFunction;

}

double selectedNizhPredel(FILE\*f,int selectedZavisimost,int selectedOtnoshenie, int selectedMethod, int selectedFunction) //вводим нижний предел

{

system("cls");

Zagolovok();

cout<<"Выбранная зависимость - "

<<selectedZavisimost

<<"\nВыбранный делитель - "

<<selectedOtnoshenie

<<"\nВыбранный метод -"

<<selectedMethod

<<"\nВыбранная функция - "

<<selectedFunction;

double a;

cout<<"\nНижний предел интегрирования a=";

cin>>a;

fprintf(f,"%s%f%s","Нижний предел интегрирования a=",a,"\t\n");

return a;

}

double selectedVerhPredel(FILE\*f) //и верхний

{

double b;

cout<<"Верхний предел интегрирования b=";

cin>>b;

fprintf(f,"%s%f%s","Верхний предел интегрирования b=",b,"\t\n");

return b;

}

double selectedKoefM(FILE\*f)

{

double m; //ну и почти не нужный коэффициент (используем в нашей задаче только для синуса)

cout<<"Коэффициент m=";

cin>>m;

cout<<"\n\nИдет вычисление...\n в зависимости от параметров процесс может занять некоторое время\n";

fprintf(f,"%s%f%s","Коэффициент m=",m,"\t\n");

return m;

}

char uvelicVerhPredel()

{

char a;

cout<<"провести исследование при увелении верхнего предела?? Y???";

cin>>a;

return a;

}

void Zagolovok() //а это самая главная часть программы, без которой ничего работь не будет

{ //выводит на экран КУРСОВАЯ, для красоты, ну или для подобия красоты

cout<<"\* \* \* \* \*\* \* \* \* \* \* \* \* \* \* \* \n"

<<"\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \n"

<<"\*\* \* \*\* \* \* \* \* \* \* \* \* \* \* \n"

<<"\* \* \* \* \* \* \* \* \* \* \* \* \* \* \n"

<<"\* \* \* \* \* \* \* \* \* \* \* \* \* \* \n"

<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \n\n";

}