|  |  |
| --- | --- |
| МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ | |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  «Пермский государственный национальный исследовательский университет» | |
| **ЧИСЛЕННЫЕ МЕТОДЫ** *Лабораторная работа №\_6\_*  **«**Численное интегрирование**»** | |
| Варианты: №3, №6 | |
|  | Работу выполнили студенты группы ПМИ-2-16  Мироненко Анастасия Олеговна,  Зимин Илья Владимирович |
| Оценка отчета   |  |  |  | | --- | --- | --- | |  | Баллы |  | | Конспект | 1 |  | | Опоздание с отчетом | -0.05 день |  | | Попытки | -0.5 попыт. |  | | Замечания к отчету | -0.25 зам. |  | | Программа | 1 |  | | Выводы  (Заключение) | 1 |  | | Защита | 1 |  | | ИТОГО: |  |  | | Проверил:  профессор, доктор физико-математических наук  С. В. Русаков  “\_\_\_\_” 2018 г. |
| Пермь 2018 | |

**Содержание**

[1. Задание. 3](#_Toc525161389)

[2. Исходные данные. 4](#_Toc525161390)

[3. Решение 5](#_Toc525161391)

[4. Краткие выводы. 6](#_Toc525161392)

[5. Текст программы. 7](#_Toc525161393)

1. **Задание**
2. Вычислить интеграл,



используя обобщенные квадратурные формулы и метод Рунге, с точностью (относительной погрешностью)  с помощью обобщенных формул:

* трапеций;
* трапеций (модифицированной с помощью сплайна);
* Симпсона;
* Гаусса (трехточечной).

Оценить порядок аппроксимации квадратурной формулы с помощью метода Рунге. Итоговый результат (значение интеграла) выдавать с максимально допустимой точностью.

1. Провести оценку эффективности рассмотренных формул по точности и вычислительным затратам.

Вычисления произвести на отрезке [1,2] для функций, использованных в лабораторной №5.

1. **Исходные данные**

|  |  |
| --- | --- |
| Вариант №3: | Вариант №6 |
|  |  |

1. **Решение**

В данном разделе будут приведены результаты, а также подробные шаги выполнения программы, на примере двух уравнений: №3 и №6.

На вход программе ничего подавать не нужно, указанные варианты статически внесены в код программы. Программа автоматически генерирует файл «output.txt» и выводит в него все необходимые данные.

# **Набор №3.**

# image003

Файл «output.txt»:

**ИСХОДНОЕ УРАВНЕНИЕ:**

f(x) = 5^x - 3

**ФОРМУЛА ТРАПЕЦИЙ**

N| h| Integral| Оценка погрешности| k|

1| 1.0000| 12.000000| | |

2| 0.5000| 10.090170| -6.3661001875018E-01| |

4| 0.2500| 9.593898| -1.6542402974852E-01| 1.9442|

8| 0.1250| 9.468583| -4.1771654233110E-02| 1.9856|

16| 0.0625| 9.437175| -1.0469285264243E-02| 1.9964|

32| 0.0313| 9.429318| -2.6189757997083E-03| 1.9991|

64| 0.0156| 9.427354| -6.5484745323729E-04| 1.9998|

128| 0.0078| 9.426862| -1.6371833379954E-04| 1.9999|

256| 0.0039| 9.426740| -4.0929987881929E-05| 2.0000|

512| 0.0020| 9.426709| -1.0232522246412E-05| 2.0000|

1024| 0.0010| 9.426701| -2.5581321404881E-06| 2.0000|

2048| 0.0005| 9.426699| -6.3953313282165E-07| 2.0000|

4096| 0.0002| 9.426699| -1.5988328409359E-07| 2.0000|

8192| 0.0001| 9.426699| -3.9970824280052E-08| 2.0000|

Результат: 9.426698731163075

Количество обращений: 8193

**ФОРМУЛА ТРАПЕЦИЙ (МОДИФ)**

N| h| Integral| Оценка погрешности| k|

1| 1.0000| 9.317603| | |

2| 0.5000| 9.419571| 3.3989111430700E-02| |

4| 0.2500| 9.426248| 2.2257527966942E-03| 3.9327|

8| 0.1250| 9.426670| 1.4079140319427E-04| 3.9827|

16| 0.0625| 9.426697| 8.8261448324782E-06| 3.9956|

32| 0.0313| 9.426699| 5.5205256129417E-07| 3.9989|

64| 0.0156| 9.426699| 3.4509831436935E-08| 3.9997|

Результат: 9.426698684289940

Количество обращений: 65

**ФОРМУЛА СИМПСОНА**

N| h| Integral| Оценка погрешности| k|

1| 1.0000| 9.453560| | |

2| 0.5000| 9.428474| -1.6724066829282E-03| |

4| 0.2500| 9.426811| -1.1083914559435E-04| 3.9154|

8| 0.1250| 9.426706| -7.0324549242902E-06| 3.9783|

16| 0.0625| 9.426699| -4.4119563931133E-07| 3.9945|

32| 0.0313| 9.426699| -2.7600882527660E-08| 3.9986|

Результат: 9.426698718799768

Количество обращений: 65

**ФОРМУЛА ГАУССА**

N| h| Integral| Оценка погрешности| k|

1| 1.0000| 9.426600| | |

2| 0.5000| 9.426697| 1.5344670483811E-06| |

4| 0.2500| 9.426699| 2.5581665998934E-08| 5.9065|

Результат: 9.426698665180934

Количество обращений: 21

# **Набор №6.**

# 

Файл «output.txt»:

**ИСХОДНОЕ УРАВНЕНИЕ:**

f(x) = e^(x-1)+2-x

**ФОРМУЛА ТРАПЕЦИЙ**

N| h| Integral| Оценка погрешности| k|

1| 1.0000| 2.359141| | |

2| 0.5000| 2.253931| -3.5069940588232E-02| |

4| 0.2500| 2.227222| -8.9030626357696E-03| 1.9779|

8| 0.1250| 2.220519| -2.2344374644049E-03| 1.9944|

16| 0.0625| 2.218841| -5.5915452810244E-04| 1.9986|

32| 0.0313| 2.218422| -1.3982275455578E-04| 1.9996|

64| 0.0156| 2.218317| -3.4957822077963E-05| 1.9999|

128| 0.0078| 2.218291| -8.7395888717090E-06| 2.0000|

256| 0.0039| 2.218284| -2.1849055524825E-06| 2.0000|

512| 0.0020| 2.218282| -5.4622690918530E-07| 2.0000|

1024| 0.0010| 2.218282| -1.3655675938177E-07| 2.0000|

2048| 0.0005| 2.218282| -3.4139191621800E-08| 2.0000|

Результат: 2.218281862598240

Количество обращений: 2049

**ФОРМУЛА ТРАПЕЦИЙ (МОДИФ)**

N| h| Integral| Оценка погрешности| k|

1| 1.0000| 2.215951| | |

2| 0.5000| 2.218134| 7.2759750466433E-04| |

4| 0.2500| 2.218273| 4.6321887454681E-05| 3.9734|

8| 0.1250| 2.218281| 2.9086664012172E-06| 3.9933|

16| 0.0625| 2.218282| 1.8200459879954E-07| 3.9983|

32| 0.0313| 2.218282| 1.1378619824143E-08| 3.9996|

Результат: 2.218281826183152

Количество обращений: 33

**ФОРМУЛА СИМПСОНА**

N| h| Integral| Оценка погрешности| k|

1| 1.0000| 2.218861| | |

2| 0.5000| 2.218319| -3.6153996989707E-05| |

4| 0.2500| 2.218284| -2.3124814566640E-06| 3.9666|

8| 0.1250| 2.218282| -1.4537653368526E-07| 3.9916|

16| 0.0625| 2.218282| -9.0993413396726E-09| 3.9979|

Результат: 2.218281837561772

Количество обращений: 33

**ФОРМУЛА ГАУССА**

N| h| Integral| Оценка погрешности| k|

1| 1.0000| 2.218281| | |

2| 0.5000| 2.218282| 1.2871135155926E-08| |

Результат: 2.218281815254037

Количество обращений: 9

1. **Краткие выводы**

**Вариант № 3**

|  |  |  |
| --- | --- | --- |
| Формула | Результат | Количество обращений |
| Трапеций | 9.426698731163075 | 8193 |
| Трапеций (модиф.) | 9.426698684289940 | 65 |
| Симпсона | 9.426698718799768 | 65 |
| Гаусса | 9.426698665180934 | 21 |



Более точный результат дала формула трапеций с модификацией, самый менее точный обычная формула трапеций.

**Вариант № 6**

|  |  |  |
| --- | --- | --- |
| Формула | Результат | Количество обращений |
| Трапеций | 2.218281862598240 | 2049 |
| Трапеций (модиф.) | 2.218281826183152 | 33 |
| Симпсона | 2.218281837561772 | 33 |
| Гаусса | 2.218281815254037 | 9 |



Опять же, более точный результат был проучен с помощью формулы трапеций с модификацией, самый менее точный с помощью обычной формулы трапеций.

* Исходя из полученных результатов можно сделать, что формула трапеций является самой медленной (и менее точной), а самой быстрой является формула Гаусса, самой более точной оказалась модифицированная при помощи сплайна формула трапеций.
* С точки зрения реализации - все формулы элементарно реализуются, поэтому трудно выделить наиболее простую, так как записи формул на языке программирования мало чем отличается для каждой.
* Модификация сплайнами усовершенствовала формулу трапеций в сторону быстродействия и точности, хотя присутствуют дополнительные затраты на подсчёт производных 1-го порядка.
* Формула Гаусса требует дополнительных затрат в ходе выполнения в виде вычисления в трёх точках, как и формула Симпсона. В случае сложных подынтегральных функций, это окажет негативное влияние.

1. **Текст программы**

Программа состоит из одного файла «Main.cpp» - исходный код программы

#define \_USE\_MATH\_DEFINES

#include <fstream>

#include <iomanip>

#include <iostream>

using namespace std;

const int variant = 6;

const double EPS = 10E-8;

// Вычисление значения функции в точке

double f(double x)

{

switch (variant)

{

case 3: return pow(5, x) - 3; // 3-й вариант

case 6: return pow(M\_E, x - 1) + 2 - x; // 6-й вариант

default: cout << "No function!"; return 0;

}

}

// Вычисление производной n-го порядка числа x

double Diff(int n, double x)

{

switch (variant)

{

case 3:

switch (n)

{

default: return pow(5, x)\*pow(log(5), n);

}

case 6:

switch (n)

{

case 1: return pow(M\_E, x - 1) - 1;

default: return pow(M\_E, x - 1);

}

default: cout << "No function!"; return 0;

}

}

// Вывод шапки таблиц в файл

void PrintHeadTable(ofstream & fout)

{

fout << setw(7) << right << "N|" << setw(10) << right << "h|" << setw(13) << right << "Integral|"<<setw(24)<<right<<"Оценка погрешности|"<< setw(10) << right<<"k|"<< endl;

}

// Вычисление интеграла на определенном шаге

double SolveIteration(int N, double a, double b, double h, int numMethod, int &count)

{

double integral(0);

if (numMethod == 0) // Формула трапеций

{

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

{

double xj = a + j \* h, xj1 = a + (j + 1) \* h;

integral += (xj1 - xj)\*(f(xj) + f(xj1)) / 2.;

}

count += N / 2;

}

else if (numMethod == 1) // Формула трапеций (модиф.)

{

for (int j = 1; j < N; j++)

integral += f(a + j \* h);

integral = h \* (0.5 \* (f(a) + f(b)) + integral)

+ (h\*h / 12)\*(Diff(1, a) - Diff(1, b));

count += N / 2;

}

else if (numMethod == 2) // Формула Симпсона

{

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

{

double xk = a + k \* h, xk1 = a + (k + 1) \* h;

integral += ((xk1 - xk) / 6)\*(f(xk) + 4 \* f((xk + xk1) / 2) + f(xk1));

}

count += N;

}

else // Формула Гаусса

{

double m = 0.;

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

{

m = a + (2 \* i + 1)\*h / 2.;

integral += h / 2. \* ((5. / 9.) \* f(m - h \* sqrt(0.6) / 2.) +

(8. / 9.) \* f(m) + (5. / 9.) \* f(m + h \* sqrt(0.6) / 2.));

count += 3;

}

}

return integral;

}

// Пошаговое вычисление интеграла

void Integral(double a, double b, int numMethod, ofstream & fout)

{

PrintHeadTable(fout);

int N = 1, count(0);

double h(1), integral(0), integralPrev, pogr(1), k, teta = (numMethod == 2) ? 1 / 15. : (numMethod == 3) ? 1/63. : 1 / 3.,

S0(SolveIteration(N, a, b, h, numMethod, count)), // Вычисляем первое значение

S1(SolveIteration(N \* 2, a, b, h / 2, numMethod, count));// Вычисляем второе значение

count = 0;

while(abs(pogr) > EPS)

{

integralPrev = integral;

h = (b - a) / N;

integral = SolveIteration(N, a, b, h, numMethod, count);

pogr = (integral - integralPrev) \* teta;

k = log((integral - S0) / (S1 - S0) - 1) / log(0.5);

S0 = S1;

S1 = integral;

fout << fixed << setw(6) << right << setprecision(0) << N << "|" << setw(9)

<< right << setprecision(4) << h << "|" << setw(12) << right

<< setprecision(6) << integral << "|" << setw(23) << right << uppercase

<< setprecision(13) << scientific << pogr << "|" << setw(9) << right << fixed << setprecision(4) << k << "|" << endl;

N \*= 2;

}

fout << "\nРезультат: " << setprecision(15) << integral << "\nКоличество обращений: "

<< setprecision(0) << ((numMethod == 3) ? count : (count + 2)) << endl;

}

// Основная функция

int main()

{

const double a = 1, b = 2;

ofstream fout("output.txt");

fout << "\n\nФОРМУЛА ТРАПЕЦИЙ\n\n";

Integral(a, b, 0, fout);

fout << "\n\nФОРМУЛА ТРАПЕЦИЙ (МОДИФ)\n\n";

Integral(a, b, 1, fout);

fout << "\n\nФОРМУЛА СИМПСОНА \n\n";

Integral(a, b, 2, fout);

fout << "\n\nФОРМУЛА ГАУССА \n\n";

Integral(a, b, 3, fout);

return 0;

}