|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
| NSTU | | |
| Кафедра вычислительных технологий | | |
|  | | |
| Лабораторная работа № 3 | | |
| по дисциплине «Численные методы» | | |
|  | | |
| **Численное интегрирование** | | |
|  | | |
|  | Факультет: | ПМИ |
| Группа: | ПМИ-81 |
| Бригада: | 4 |
| Студенты: | Кайда Даниил |
|  | Парышков Дмитрий |
|  |  |
| Преподаватели: | Иткина Наталья Борисовна |
|  | Марков Сергей Игоревич |
|  | | |
| Новосибирск | | |
| 2020 | | |

1. **Цель работы**

Изучить и реализовать алгоритмы построения квадратурных формул интерполяционного типа и квадратур Гаусса.

1. **Ход работы**
   1. Разработан класс, реализующий схемы численного интегрирования. Среди реализованных алгоритмов: схема средних прямоугольников (Gauss1), две схемы семейства Ньютона-Котеса (Trapezium и Simpson) и четыре квадратурные формулы Гаусса (Gauss1(схема средних прямоугольников), Gauss2, Gauss3, Gauss4, Gauss5). Далее проводились исследования на всех схемах, кроме: Gauss4, Gauss5

Так же код программы находится в архиве.

class Integration\_Scheme // Вспомогательный класс

{

protected:

vector<Point> Points;

vector<double> Weight;

public:

enum Integration\_Scheme\_Type

{

Gauss1,

Gauss2,

Gauss3,

Gauss4,

Gauss5,

Trapezium,

Simpson

};

};

Класс вычисления интеграла. Как можно увидеть, он имеет функцию с разными параметрами -ситуация перегрузки функции, второй вариант этой функции может быть использован для любого разбиения (равномерного или неравномерного), в неё подаётся вектор разбиений. Первый вариант может считать только для равномерного разбиения. Для наших исследований был выбран первый вариант, т.к. он удобнее.

class Integtal\_calculation : protected Integration\_Scheme

{

public:

Integtal\_calculation(Integration\_Scheme\_Type Type);

double Calculate\_Integral(const Point& Begin, const Point& End, int Number\_Segments,

const std::function<double(const Point & P)>& Func) const;

double Calculate\_Integral(vector<Point> MeshSplit,

const std::function<double(const Point & P)>& Func) const;

};

Integtal\_calculation::Integtal\_calculation(Integration\_Scheme\_Type Type)

{

switch (Type)

{

case Gauss1:

Weight = { 2 };

Points = { Point(0) };

break;

case Gauss2:

Weight = { 1, 1 };

Points = { Point(-1.0 / sqrt(3.0)),

Point(1.0 / sqrt(3.0)) };

break;

case Gauss3:

Weight = { 5.0 / 9.0, 8.0 / 9.0, 5.0 / 9.0 };

Points = { Point(-sqrt(3.0 / 5.0)),

Point(0),

Point(sqrt(3.0 / 5.0)) };

break;

case Gauss4:

Weight = { (18 - sqrt(30.0) / 36.0),

(18 + sqrt(30.0)) / 36.0,

((18 + sqrt(30.0)) / 36.0),

((18 - sqrt(30.0)) / 36.0) };

Points = { Point(-sqrt((3 + 2.0 \* sqrt(6.0 / 5.0) / 7.0))),

Point(-sqrt((3 - 2.0 \* sqrt(6.0 / 5.0) / 7.0))),

Point(sqrt((3 - 2.0 \* sqrt(6.0 / 5.0) / 7.0))),

Point(sqrt((3 + 2.0 \* sqrt(6.0 / 5.0) / 7.0))) };

break;

case Gauss5:

Weight = { (322 - 13.0 \* sqrt(17.0)) / 900.0,

(322 + 13.0 \* sqrt(17.0)) / 900.0,

128.0 / 255.0,

(322 - 13.0 \* sqrt(17.0)) / 900.0,

(322 + 13.0 \* sqrt(17.0)) / 900.0 };

Points = { Point(-1.0 / 3.0 \* (sqrt(5 + 2.0 \* sqrt(10.0 / 7.0)))),

Point(-1.0 / 3.0 \* (sqrt(5 - 2.0 \* sqrt(10.0 / 7.0)))),

Point(0),

Point(1.0 / 3.0 \* (sqrt(5 - 2.0 \* sqrt(10.0 / 7.0)))),

Point(1.0 / 3.0 \* (sqrt(5 + 2.0 \* sqrt(10.0 / 7.0)))) };

break;

case Trapezium:

Weight = { 1, 1 };

Points = { Point(-1.0), Point(1.0) };

break;

case Simpson:

Weight = { 1.0 / 3.0, 4.0 / 3.0, 1.0 / 3.0 };

Points = { Point(-1), Point(0), Point(1) };

break;

}

}

double Integtal\_calculation::Calculate\_Integral(const Point& Begin, const Point& End, int Number\_Segments,

const function<double(const Point & P)>& Func) const

{

double Result = 0.0;

double X0;

double h = (End.x() - Begin.x()) / Number\_Segments;

for (int i = 0; i < Number\_Segments; i++)

{

X0 = Begin.x() + i \* h;

for (int i = 0; i < Points.size(); i++)

{

Point P = Point(X0 + (1 + Points[i].x()) \* h / 2.0);

Result += Weight[i] \* Func(P);

}

}

return Result \* (h / 2.0);

}

double Integtal\_calculation::Calculate\_Integral(vector<Point> MeshSplit, const function<double(const Point & P)>& Func) const

{

double Result = 0.0, Result\_2 = 0.0;

double X0;

for (int i = 1; i < MeshSplit.size(); i++)

{

X0 = MeshSplit[i].x();

double h = abs(MeshSplit[i].x() - MeshSplit[i-1].x());

for (int i = 0; i < Points.size(); i++)

{

Point P = Point(X0 + (1 + Points[i].x()) \* h / 2.0);

Result\_2 += Weight[i] \* Func(P);

}

Result += Result\_2 \* h;

Result\_2 = 0.0;

}

return Result / 2.0;

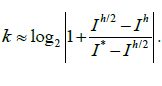
}

* 1. Исследуем две не полиномиальные функции: ex и sin(x) на отрезках [0,2] и [0,1], соответственно, с шагом h = 0.1 и h/2 = 0.05 на регулярной сетки.

I – вычисленное значение интеграла по заданной схеме с заданным шагом

I\* - точное значение интеграла

Порядок аппроксимации вычисляется по формуле:



* + 1. **Функция ex на отрезке [0,2]**

**Схема интегрирования: метод Гаусса 1 – метод прямоугольников**

Выходные данные программы:

h = 1.000000e-01

I = 6.386395e+00

I\* = 6.389056e+00

|I - I\*| = 2.661330e-03

h = 5.000000e-02

I = 6.388391e+00

I\* = 6.389056e+00

|I - I\*| = 6.654782e-04

Approximation order: 1.999684e+00

Порядок аппроксимации равен 1.999684e+00 ~ 2.

**Схема интегрирования: метод Гаусса 2**

Выходные данные программы:

h = 1.000000e-01

I = 6.389056e+00

I\* = 6.389056e+00

|I - I\*| = 1.478479e-07

h = 5.000000e-02

I = 6.389056e+00

I\* = 6.389056e+00

|I - I\*| = 9.242692e-09

Approximation order: 3.999657e+00

Порядок аппроксимации равен 3.999657e+00 ~ 4.

**Схема интегрирования: метод Гаусса 3**

Выходные данные программы:

h = 1.000000e-01

I = 6.389056e+00

I\* = 6.389056e+00

|I - I\*| = 3.167244e-12

h = 5.000000e-02

I = 6.389056e+00

I\* = 6.389056e+00

|I - I\*| = 4.884981e-14

Approximation order: 6.018731e+00

Порядок аппроксимации равен 6.018731e+00 ~ 6.

**Схема интегрирования: Метод трапеций**

Выходные данные программы:

h = 1.000000e-01

I = 6.394379e+00

I\* = 6.389056e+00

|I - I\*| = 5.323326e-03

h = 5.000000e-02

I = 6.390387e+00

I\* = 6.389056e+00

|I - I\*| = 1.330998e-03

Approximation order: 1.999820e+00

Порядок аппроксимации равен 1.999820e+00 ~ 2.

**Схема интегрирования: Метод Симпсона**

Выходные данные программы:

h = 1.000000e-01

I = 6.389056e+00

I\* = 6.389056e+00

|I - I\*| = 2.217762e-07

h = 5.000000e-02

I = 6.389056e+00

I\* = 6.389056e+00

|I - I\*| = 1.386411e-08

Approximation order: 3.999678e+00

Порядок аппроксимации равен 3.999678e+00 ~ 4.

* + 1. **Функция sin(x)на отрезке [0,1]**

**Схема интегрирования: метод Гаусса 1 – метод прямоугольников**

Выходные данные программы:

h = 1.000000e-01

I = 4.598893e-01

I\* = 4.596977e-01

|I - I\*| = 1.915966e-04

h = 5.000000e-02

I = 4.597456e-01

I\* = 4.596977e-01

|I - I\*| = 4.788867e-05

Approximation order: 2.000316e+00

Порядок аппроксимации равен : 2.000316e+00 ~ 2.

**Схема интегрирования: метод Гаусса 2**

Выходные данные программы:

h = 1.000000e-01

I = 4.596977e-01

I\* = 4.596977e-01

|I - I\*| = 1.064453e-08

h = 5.000000e-02

I = 4.596977e-01

I\* = 4.596977e-01

|I - I\*| = 6.651246e-10

Approximation order: 4.000344e+00

Порядок аппроксимации равен 4.000344e+00 ~ 4.

**Схема интегрирования: метод Гаусса 3**

Выходные данные программы:

Порядок аппроксимации равен h = 1.000000e-01

I = 4.596977e-01

I\* = 4.596977e-01

|I - I\*| = 2.282063e-13

h = 5.000000e-02

I = 4.596977e-01

I\* = 4.596977e-01

|I - I\*| = 3.774758e-15

Approximation order: 5.917811e+00.

Порядок аппроксимации равен 5.917811e+00 ~ 6.

**Схема интегрирования: Метод трапеций**

Выходные данные программы:

h = 1.000000e-01

I = 4.593145e-01

I\* = 4.596977e-01

|I - I\*| = 3.831453e-04

h = 5.000000e-02

I = 4.596019e-01

I\* = 4.596977e-01

|I - I\*| = 9.577434e-05

Approximation order: 2.000180e+00

Порядок аппроксимации равен 2.000180e+00 ~ 2.

**Схема интегрирования: Метод Симпсона**

Выходные данные программы:

h = 1.000000e-01

I = 4.596977e-01

I\* = 4.596977e-01

|I - I\*| = 1.596648e-08

h = 5.000000e-02

I = 4.596977e-01

I\* = 4.596977e-01

|I - I\*| = 9.976820e-10

Approximation order: 4.000322e+00

Порядок аппроксимации равен 4.000322e+00 ~ 4.

* 1. Порядок точности квадратурной формулы m = 4. Зададим отрезок [-1,1] и построим три вложенные сетки с равномерным шагом h = 0.2, h/2 = 0.1 и h/4 = 0.05. Была создана подпрограмма для вывода чисел в экспоненциальной форме, где каждое число содержит четыре значащих цифры мантиссы числа в нормализованном виде.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Схема интегрирования: метод Гаусс 1** | | | | | | |
| *f(x)* | Шаг  *h* | *I\*- Ih* |  |  | *IR* | *I\*- IR* |
| *x3* | *0.2* | -0.6661e-16 | 0.4615e+00 | -0.1443e-15 | 0.0000e+00 | 0.0000e+00 |
| *x4* | 0.1324e-01 | 0.3979e+01 | 0.3327e-02 | 0.4000e+00 | 0.0000e+00 |
| *x5* | 0.0000e+00 | 0.0000e+00 | -0.1443e-15 | 0.0000e+00 | 0.0000e+00 |
| *x3* | *0.1* | -0.1443e-15 | 0.6500e+00 | -0.2220e-15 | 0.0000e+00 | 0.0000e+00 |
| *x4* | 0.3327e-02 | 0.3995e+01 | 0.8330e-03 | 0.4000e+00 | 0.0000e+00 |
| *x5* | -0.1443e-15 | 0.8667e+00 | -0.1665e-15 | 0.0000e+00 | 0.0000e+00 |
| *x3* | *0.05* | -0.2220e-15 | 0.1333e+02 | -0.1665e-16 | 0.0000e+00 | 0.0000e+00 |
| *x4* | 0.8330e-03 | 0.3999e+01 | 0.2083e-03 | 0.4000e+00 | 0.0000e+00 |
| *x5* | -0.1665e-15 | 0.2000e+02 | -0.8327e-17 | -0.1541e-32 | 0.1541e-32 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Схема интегрирования: метод Гаусс 2** | | | | | | |
| *f(x)* | Шаг  *h* | *I\*- Ih* |  |  | *IR* | *I\*- IR* |
| *x3* | *0.2* | -0.1110e-16 | 0.1429e+00 | -0.7772e-16 | 0.0000e+00 | 0.0000e+00 |
| *x4* | 0.1778e-04 | 0.1600e+02 | 0.1111e-05 | 0.4000e+00 | 0.0000e+00 |
| *x5* | -0.2220e-16 | 0.4444e+00 | -0.4996e-16 | 0.6163e-32 | -0.6163e-32 |
| *x3* | *0.1* | -0.7772e-16 | 0.2667e+00 | -0.2914e-15 | 0.0000e+00 | 0.0000e+00 |
| *x4* | 0.1111e-05 | 0.1600e+02 | 0.6944e-07 | 0.4000e+00 | 0.0000e+00 |
| *x5* | -0.4996e-16 | 0.3673e+00 | -0.1360e-15 | 0.0000e+00 | 0.0000e+00 |
| *x3* | *0.05* | -0.2914e-15 | 0.3684e+01 | -0.7910e-16 | 0.0000e+00 | 0.0000e+00 |
| *x4* | 0.6944e-07 | 0.1600e+02 | 0.4340e-08 | 0.4000e+00 | 0.0000e+00 |
| *x5* | -0.1360e-15 | 0.6533e+00 | -0.2082e-15 | 0.2465e-31 | -0.2465e-31 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Схема интегрирования: метод Гаусс 3** | | | | | | |
| *f(x)* | Шаг  *h* | *I\*- Ih* |  |  | *IR* | *I\*- IR* |
| *x3* | *0.2* | -0.1221e-15 | 0.7857e+00 | -0.1554e-15 | 0.0000e+00 | 0.0000e+00 |
| *x4* | -0.1665e-15 | 0.1600e+01 | -0.76660e-02 | 0.4000e+00 | 0.0000e+00 |
| *x5* | -0.6661e-16 | 0.6000e+00 | -0.1110e-15 | 0.0000e+00 | 0.0000e+00 |
| *x3* | *0.1* | -0.1554e-15 | 0.1556e+01 | -0.9992e-16 | 0.2465e-31 | -0.2465e-31 |
| *x4* | -0.1665e-15 | 0.6000e+00 | -0.2776e-15 | 0.4000e+00 | 0.0000e+00 |
| *x5* | -0.1110e-15 | 0.1429e+01 | -0.7772e-16 | -0.1233e-31 | 0.1233e-31 |
| *x3* | *0.05* | -0.9992e-16 | 0.7579e+00 | -0.1318e-15 | 0.0000e+00 | 0.0000e+00 |
| *x4* | -0.2776e-15 | 0.1475e+01 | 0.4678e-07 | 0.4000e+00 | 0.0000e+00 |
| *x5* | -0.7772e-16 | 0.7089e+00 | -0.1096e-15 | 0.1233e-31 | -0.1233e-31 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Схема интегрирования: метод Трапеций** | | | | | | |
| *f(x)* | Шаг  *h* | *I\*- Ih* |  |  | *IR* | *I\*- IR* |
| *x3* | *0.2* | -0.1110e-16 | 0.1667e+00 | -0.6661e-16 | 0.0000e+00 | 0.0000e+00 |
| *x4* | -0.2656e-01 | 0.3988e+01 | -0.6660e-02 | 0.4000e+00 | 0.0000e+00 |
| *x5* | 0.0000e+00 | 0.0000e+00 | -0.1110e-15 | 0.0000e+00 | 0.0000e+00 |
| *x3* | *0.1* | -0.6661e-16 | 0.2667e+00 | -0.2498e-15 | 0.0000e+00 | 0.0000e+00 |
| *x4* | -0.6660e-02 | 0.3997e+01 | -0.1666e-02 | 0.4000e+00 | 0.0000e+00 |
| *x5* | -0.1110e-15 | 0.8696e+00 | -0.1277e-15 | 0.2465e-31 | -0.2465e-31 |
| *x3* | *0.05* | -0.2498e-15 | 0.2647e+01 | -0.9437e-16 | -0.1233e-31 | 0.1233e-31 |
| *x4* | -0.1666e-02 | 0.3999e+01 | -0.4166e-03 | 0.4000e+00 | 0.0000e+00 |
| *x5* | -0.1277e-15 | 0.9583e+00 | -0.1332e-15 | 0.1479e-30 | -0.1479e-30 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Схема интегрирования: метод Симпсона** | | | | | | |
| *f(x)* | Шаг  *h* | *I\*- Ih* |  |  | *IR* | *I\*- IR* |
| *x3* | *0.2* | -0.1665e-16 | 0.1224e+00 | -0.1360e-15 | 0.0000e+00 | 0.0000e+00 |
| *x4* | -0.2667e-04 | 0.1600e+02 | -0.1667e-05 | 0.4000e+00 | 0.0000e+00 |
| *x5* | -0.5551e-17 | 0.3571e-01 | -0.1554e-15 | 0.0000e+00 | 0.0000e+00 |
| *x3* | *0.1* | -0.1360e-15 | 0.1101e+01 | -0.1235e-15 | -0.2465e-31 | 0.2465e-31 |
| *x4* | -0.1667e-05 | 0.1600e+02 | -0.1042e-06 | 0.4000e+00 | 0.0000e+00 |
| *x5* | -0.1554e-15 | 0.1333e+01 | -0.1166e-15 | 0.2465e-31 | -0.2465e-31 |
| *x3* | *0.05* | -0.1235e-15 | 0.3956e+01 | -0.3123e-16 | 0.0000e+00 | 0.0000e+00 |
| *x4* | -0.1042e-06 | 0.1600e+02 | -0.6510e-08 | 0.4000e+00 | 0.0000e+00 |
| *x5* | -0.1166e-15 | 0.2754e+01 | -0.4233e-16 | -0.6163e-32 | 0.6163e-32 |

* 1. Выполнили исследования из пункта 3 для не полиномиальной функции

*x\**sin(10.000*x*) на отрезке [0,1] . Исследовали влияние измельчения шага сетки к концам отрезка [0,1] на относительную и абсолютную погрешности численного интегрирования.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *f(x) = x*\*sin(10000*x*) | | | | | | |
| Схема интегрирования | Шаг  *h* | *I\*- Ih* |  |  | *IR* | *I\*- IR* |
| *Метод Гаусс 1* | *0,1* | 0.9878e-01 | 0.4004e+01 | 0.2467e-01 | 0.9521e-04 | -0.1014e-16 |
| *0,05* | 0.2467e-01 | 0.1264e+01 | 0.1951e-01 | 0.9521e-04 | 0.2711e-18 |
| *0,025* | 0.1951e-01 | 0.1057e+01 | 0.1847e-01 | 0.9521e-04 | 0.2711e-18 |
| *Метод Гаусс 2* | *0,1* | 0.2711e-18 | 0.3828e+01 | 0.2431e-01 | 0.9521e-04 | 0.2711e-18 |
| *0,05* | 0.2431e-01 | -0.1263e+01 | 0.1658e+00 | 0.1851e+00 | -0.1850e+00 |
| *0,025* | -0.1925e-01 | 0.2789e+02 | -0.6904e-03 | 0.9521e-04 | 0.1626e-18 |
| *Метод Гаусс 3* | *0,1* | 0.9670e-02 | 0.5713e+00 | 0.1692e-01 | 0.9521e-04 | 0.2711e-18 |
| *0,05* | 0.1692e-01 | -0.4180e+02 | 0.4248e-03 | 0.9249e-03 | -0.8297e-03 |
| *0,025* | -0.4049e-03 | -0.7455e-01 | 0.6307e-02 | 0.9702e-03 | -0.8750e-03 |
| *Метод Трапеций* | *0,1* | -0.8637e-01 | -0.1391e+02 | -0.7170e-02 | -0.1328e-01 | 0.1338e-01 |
| *0,05* | 0.6209e-02 | 0.4021e+00 | 0.1544e-01 | 0.9521e-04 | 0.2711e-18 |
| *0,025* | 0.1544e-01 | 0.8834e+00 | 0.1748e-01 | 0.9521e-04 | 0.3740e-17 |
| *Метод Симпсона* | *0,1* | 0.3707e-01 | 0.2002e+01 | 0.1852e-01 | 0.9521e-04 | -0.6668e-17 |
| *0,05* | 0.1852e-01 | 0.1020e+01 | 0.1816e-01 | 0.9521e-04 | 0.2711e-18 |
| *0,025* | 0.1816e-01 | 0.1001e+01 | 0.1814e-01 | 0.9521e-04 | 0.3497e-16 |

|  |  |  |  |
| --- | --- | --- | --- |
| Схема интегрирования | Шаг  *h* | ∆ | δ |
| *Метод Гаусс 1* | *0,1* | 0.9878e-01 | 0.1038e+04 |
| *0,05* | 0.2467e-01 | 0.2591e+03 |
| *0,025* | 0.1951e-01 | 0.2050e+03 |
| *Метод Гаусс 2* | *0,1* | 0.9308e-01 | 0.9776e+03 |
| *0,05* | 0.2431e-01 | 0.2553e+03 |
| *0,025* | 0.1925e-01 | 0.2022e+03 |
| *Метод Гаусс 3* | *0,1* | 0.9670e-02 | 0.1016e+03 |
| *0,05* | 0.1692e-01 | 0.1778e+03 |
| *0,025* | 0.4049e-03 | 0.4253e+01 |
| *Метод Трапеций* | *0,1* | 0.8637e-01 | 0.9071e+03 |
| *0,05* | 0.6209e-02 | 0.6521e+02 |
| *0,025* | 0.1544e-01 | 0.1622e+03 |
| *Метод Симпсона* | *0,1* | 0.3707e-01 | 0.3893e+03 |
| *0,05* | 0.1852e-01 | 0.1945e+03 |
| *0,025* | 0.1816e-01 | 0.1907e+03 |

1. **Вывод**

Были изучены и реализованы алгоритмы построения квадратурных формул интерполяционного типа и квадратур Гаусса. Для них выполнили оценку порядка аппроксимации относительно шага равномерного сеточного разбиения, результаты оказались равны теоретическим, с небольшой погрешностью. Из пункта 2.3 видно, что уточнение по Ричардсону даёт почти верный результат с точностью до e-30 для полиномов. Из пункта 2.4 (первая таблица) видно, что при уменьшении шага сетки, результат вычисления интеграла приближается к реальным значениям, следовательно, при уменьшении шага сетки, вычисления становятся точнее. Из пункта 2.4 (Таблица 2) видно, что измельчение шага сетки к концам отрезка [0,1] уменьшает абсолютную и относительную погрешность.