Лабораторная работа №6

Тужилкина Н.Г., БПМ-18-2, вариант 23.

# Численное интегрирование

Требуется найти приближенное значение следующего интеграла

В качестве отрезка интегрирования взять отрезок . На этом отрезке подынтегральная функция не обращается в бесконечность.

Точное значение интеграла по формуле Ньютона-Лейбница:

## Цель работы:

Вычислить приближённое значение интеграла, используя

* составную формулу трапеций с 6-ью промежутками,
* составную формулу Симпсона с 6-ью промежутками,
* квадратурную формула Гаусса с 5-ью узлами.

## Формула трапеций

### Изложение метода

Простейшая формула трапеций применительно к интегрированию по отрезку может быть записана в виде точного равенства

где ‒ некоторая неизвестная точка интервала , а .

Выполняя разбиение исходного промежутка интегрирования на частей с шагом и применяя к каждой из частей, на которые по свойству аддитивности расчленяется исходный интеграл, будем иметь

Отсюда следует, что искомое значение интеграла можно приближенно найти по формуле

которую называют формулой трапеций.

### Текст программы

double trapezoid(const double& a, const double& b, const size\_t& n) {

double h((b - a) / n);

vector<double> x, y;

x.push\_back(a);

double sum\_y(0);

for (size\_t i(0); i <= n; ++i)

x.push\_back(h + x.back());

for (size\_t i(0); i <= n; ++i) {

y.push\_back(sqrt(x[i] \* x[i] + 2 \* x[i] + 5));

if ((i != 0) && (i != n))

sum\_y += y.back();

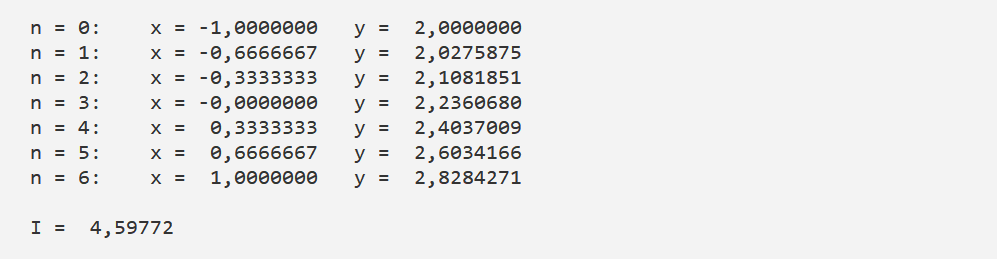
}

double I(h \* ((y[0] + y[n]) / 2 + sum\_y));

return I;

}

### Представление результатов



## Формула Симпсона

### Изложение метода

На основе простейшей формулы Симпсона записывается равенство

Выполнив разбиение так, чтобы число элементарных промежутков было четным, исходный интеграл представляется суммой интегралов вида

Отсюда получается формула численного интегрирования

где

которую называют формулой Симпсона.

### Текст программы

double simpson(const double& a, const double& b, const size\_t& n) {

double h((b - a) / n);

vector<double> x, y;

x.push\_back(a);

for (size\_t i(0); i <= n; ++i)

x.push\_back(h + x.back());

for (size\_t i(0); i <= n; ++i)

y.push\_back(sqrt(x[i] \* x[i] + 2 \* x[i] + 5));

double sigma1(0), sigma2(0);

for (size\_t i(1); i <= n - 1; i += 2)

sigma1 += y[i];

for (size\_t i(2); i <= n - 2; i += 2)

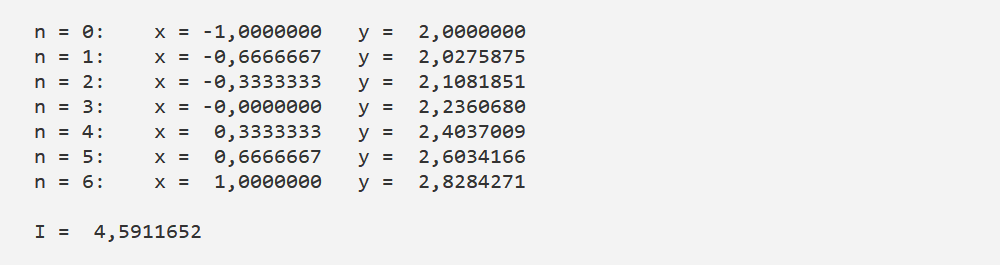
sigma2 += y[i];

double I((h / 3) \* (y[0] + y[n] + 4 \* sigma1 + 2 \* sigma2));

return I;

}

### Представление результатов



## Формула Гаусса

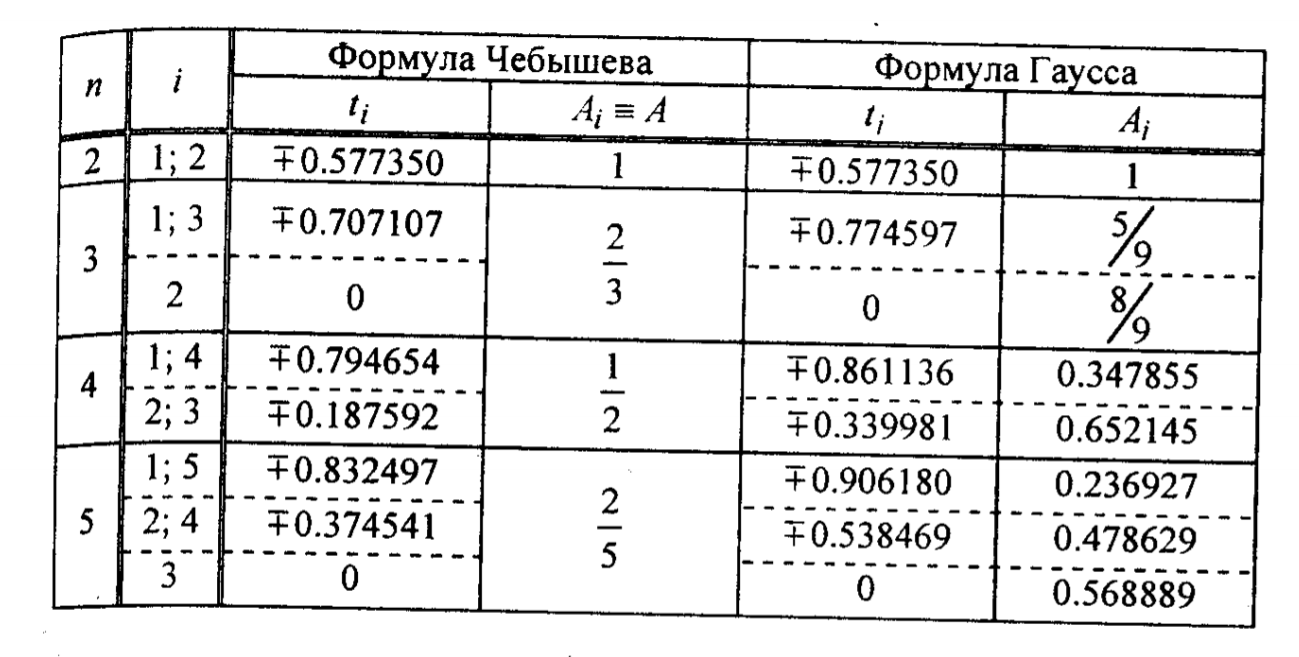
### Изложение метода

Общий вид линейной квадратурной формулы — это

где фиксированные аргументы называют узлами, а коэффициенты — весами (весовыми коэффициентами) квадратурной формулы.

Общая формула для квадратур Чебышева и Гаусса применительно к исходному интегралу по промежутку :

Значения узлов и весов , округленные до шести знаков после запятой, при приведены в следующей таблице:



### Текст программы

double gauss(const double& a, const double& b, const size\_t& n) {

vector<double> t{ -0.906180, -0.538469, 0, 0.538469, 0.906180 };

vector<double> A { 0.236927, 0.478629, 0.568889, 0.478629, 0.236927 };

double I(0);

vector<double> x;

for (size\_t i(1); i <= n; ++i) {

x.push\_back((a + b) / 2 + ((b - a) \* t[i - 1]) / 2);

double f(sqrt(x.back() \* x.back() + 2 \* x.back() + 5));

I += A[i - 1] \* f;

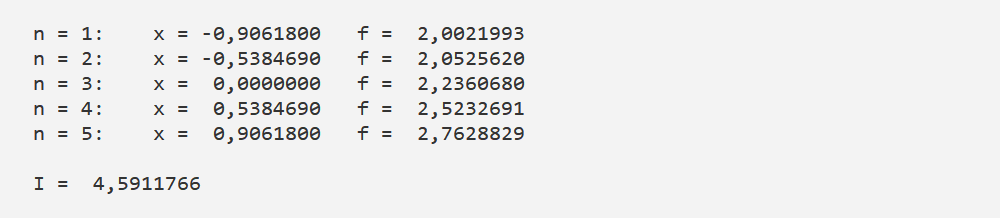
}

I = (b - a) \* I / 2;

return I;

}

### Представление результатов



## Вывод

|  |  |  |
| --- | --- | --- |
| Метод | *Значение* | *Погрешность* |
| Составная формула трапеций с 6-ью промежутками | 4.5977239 | 0,0065496 |
| Составная формула Симпсона с 6-ью промежутками | 4.5911652 | 0,0000091 |
| Квадратурная формула Гаусса с 5-ью узлами | 4,5911766 | 0,0000023 |