МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В. И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по практической работе №5

по дисциплине «Вычислительная математика»

Тема: Численное вычисление определённого интеграла

Студент гр. 0304 Крицын Д. Р.

Преподаватель Попова Е. В.

Санкт-Петербург

2021

Цель работы.

Целью работы изучение и сравнение различных методов численного интегрирования на примере составных формул прямоугольников, трапеций, Симпсона.

Основные теоретические положения.

Пусть требуется найти определенный интеграл

$$I=\int_a^b f(x)\,dx,$$

где функция f(x) непрерывна на отрезке [a,b]. Для приближенного вычисления интегралов чаще всего подынтегральную функцию заменяют «близкой» ей вспомогательной функцией, интеграл от которой вычисляется аналитически. За приближенное значение интеграла принимают значение интеграла от вспомогательной функции.

Заменим функцию на отрезке [a,b] её значением в середине отрезка. Искомый интеграл, равный площади криволинейной фигуры, заменяется на площадь прямоугольника. Из геометрических соображений нетрудно записать формулу прямоугольников:

$$\int_a^b f(x)dx \approx f\left(\frac{a+b}{2}\right)(b-a).$$

Приблизив f(x) линейной функцией и вычислив площадь соответствующей трапеции, получим формулу трапеций:

$$\int_a^b f(x)dx \approx 1/2(f(a)+f(b))(b-a).$$

Если же приблизить подынтегральную функцию параболой, проходящей через точки $(a, f(a))$, $(a+b/2\ , f((a+b)/2))$, $(b, f(b))$, то получим формулу Симпсона (парабол):

$$\int_a^b f(x)dx \approx \frac{1}{6}(f(a)+4f\left(\frac{a+b}{2}\right)+f(b))(b-a).$$

Для повышения точности интегрирования применяют составные формулы. Для этого разбивают отрезок $[a, b]$ на четное $n = 2m$ число отрезков длины $h = (b-a)/n$ и на каждом из отрезков длины 2h применяют

соответствующую формулу. Таким образом получают составные формулы прямоугольников, трапеций и Симпсона.

На сетке $x_i = a + ih$, $yi = f(x_i)$, $i = 0,2m$ составные формулы имеют следующий вид:

Формула прямоугольников:

$$\int_a^b f(x)dx = h\sum_{i=0}^{n-1} f\left(x_i + \frac{h}{2}\right) + R_1$$

$$R_1 \approx \frac{h^2}{24}\int_a^b f''(x)\,dx$$

Формула трапеций:

$$\int_a^b f(x)dx = h\sum_{i=0}^{n-1}\left(f(x_i) + f(x_{i+1})\right) + R_2$$

$$R_2 \approx -\frac{h^2}{12}\int_a^b f''(x)dx$$

Формула Симпсона:

$$\int_a^b f(x)dx = h/3\sum_{i=0}^{m-1}\left(f(x_{2i}) + 4f(x_{2i+2})\right) + R_3$$

$$R_3 \approx -\frac{h^4}{180}\int_a^b f^{IV}(x)\,dx$$

где R_1, R_2, R_3 – остаточные члены. Оценки остаточных членов получены в предположении, что соответствующие производные f(x) непрерывны на [a,b]. При $n \to \infty$ приближенные значения интегралов для всех трех формул (в предположении отсутствия погрешностей округления) стремятся к точному значению интеграла.

Для практической оценки погрешности квадратурной можно использовать правило Рунге. Для этого проводят вычисления на сетках с шагом h и h/2, получают приближенные значения интеграла I_h и $I_{h/2}$ за окончательные значения интеграла принимают величины:

- $I_{\frac{h}{2}} + \frac{I_{\frac{h}{2}} - I_h}{3}$ – для формулы прямоугольников;

- $I_{\frac{h}{2}} - \frac{I_{\frac{h}{2}} - I_h}{3}$ – для формулы трапеций;
- $I_{\frac{h}{2}} - \frac{I_{\frac{h}{2}} - I_h}{15}$ – для формулы Симпсона.

При этом за погрешность приближённого значения интеграла принимается величина $\left|\frac{I_{\frac{h}{2}} - I_h}{2^k - 1}\right|$, причём для формул прямоугольников и трапеций k = 2, для формулы Симпсона k = 4.

Такую оценку погрешностей применяют обычно для построения адаптивных алгоритмов, т.е. таких алгоритмов, которые автоматически так определяют величину шага h, что результат удовлетворяет требуемой точности.

Порядок выполнения работы.

1. Составить подпрограмму-функцию F для вычисления подынтегральной функции.

2. Составить подпрограммы-функции для вычисления интегралов по формулам прямоугольников, трапеций и Симпсона соответственно.

3. Составить головную программу, содержащую оценку по Рунге погрешности каждой из перечисленных ранее квадратурных формул, удваивающих n до тех пор, пока погрешность не станет меньше ε, и осуществляющих печать результатов значения интеграла и значения n для каждой формулы.

4. Провести вычисления по программе, добиваясь, чтобы результат удовлетворял требуемой точности.

Выполнение работы.

Данный интеграл: $\int_0^1 \sin\left(x^4+2x^3+x^2\right)dx$.

1. Расчёт значения определённого интервала при помощи онлайн-калькулятора https://www.integral-calculator.ru/, получившегося равным **0,3629205713226523**, и принятие данного значения за точное в составленной для вычисления интеграла программе. Данная программа также считает получившуюся погрешность вычисления интеграла по Рунге для каждого из используемых методов. График интегрируемой функции приведён на Рисунке 1.

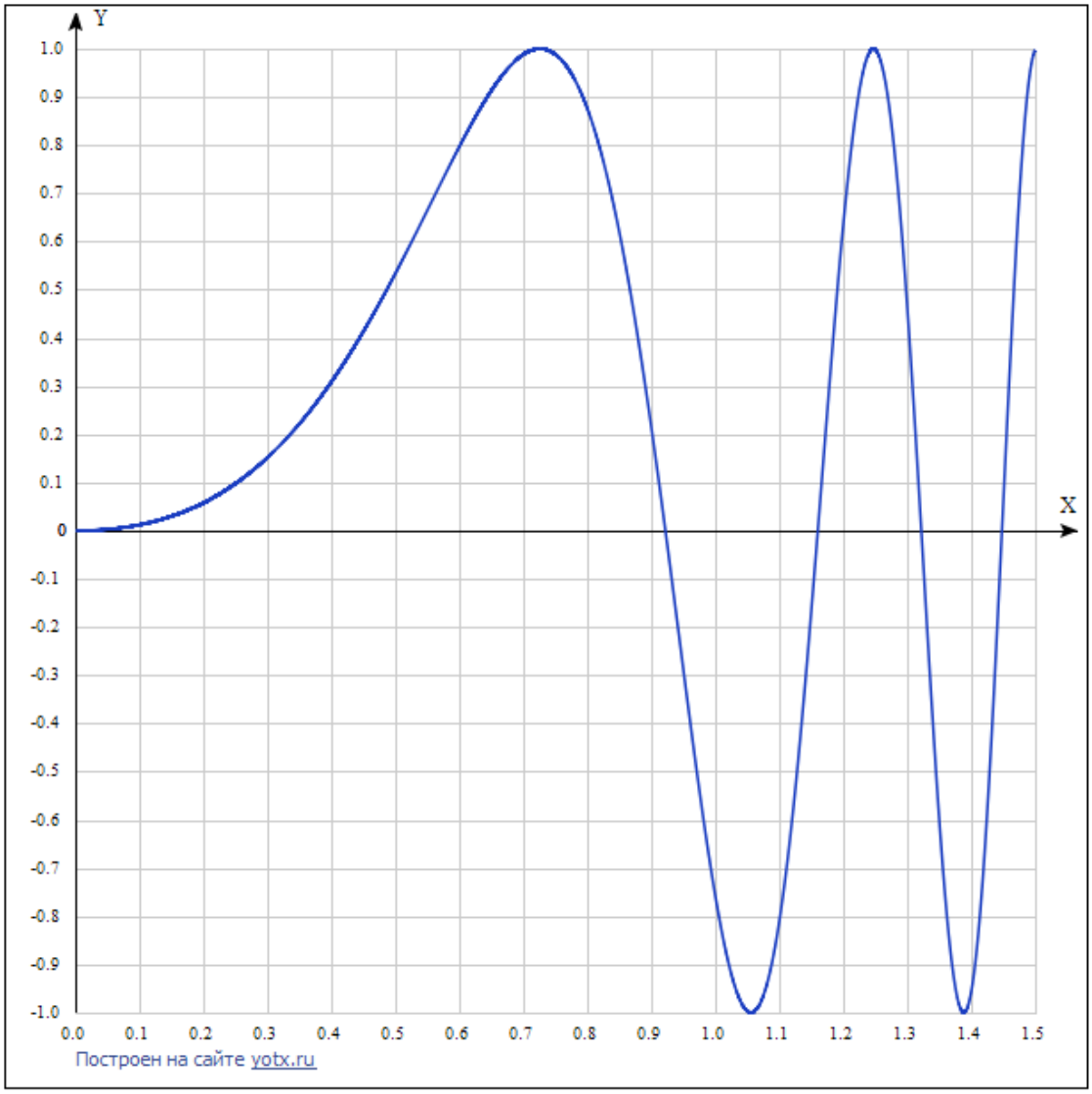


Рисунок 1 — график функции $\sin\left(x^4+2x^3+x^2\right)$

2. С помощью метода прямоугольников было вычислено приблизительное значение интеграла, а также необходимое число точек, на которые разбивается интегрируемый отрезок функции. Результаты данных вычислений при различных значниях точности ε приведены на Рисунке 2.

Метод прямоугольников:

ε	I	ΔI	Неточность по Рунге	N
0.00000100	0.36292182	0.00000125	0.000000311692	512
0.00001000	0.36294053	0.00001996	0.000004989456	128
0.00010000	0.36324183	0.00032126	0.000080446980	32
0.00100000	0.36423137	0.00131079	0.000329844211	16

Рисунок 2 — Результат вычисления интеграла методом прямоугольников

Видно, что формула метода прямоугольников значительно уменьшает минимально необходимое количество точек на интервале интегрирования [a; b] — видна зависимость $N \sim 2^{\frac{1}{\varepsilon}}$. Неточность по Рунге для данного метода на несколько порядков меньше, чем вычисленный интеграл. График зависимости числа точек разбиения от требуемой точности ε приведён на Рисунке 3.

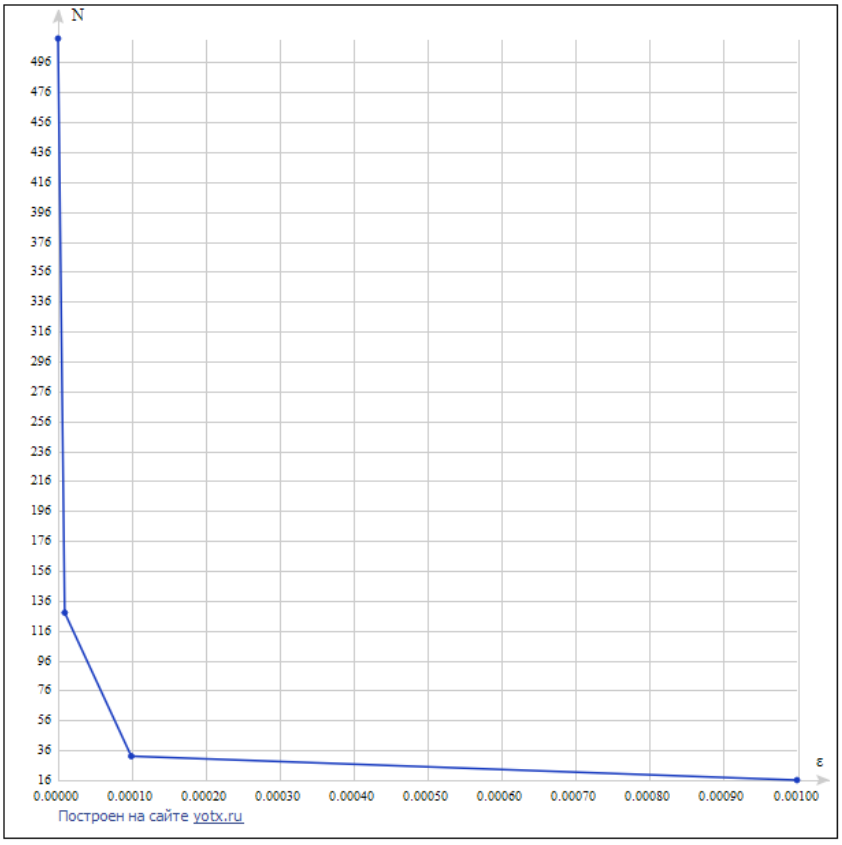


Рисунок 3 — график зависимости N от ε для метода прямоугольников

3. С помощью метода трапеций было вычислено приблизительное значение интеграла, а также нужное число точек, на которые разбивается отрезок интегрирования функции. Результаты данных вычислений при различных значениях точности ε приведены на Рисунке 4.

Метод трапеций:

ε	I	ΔI	Неточность по Рунге	N
0.00000100	0.36292057	0.00000000	0.000000240583	262144
0.00001000	0.36292057	0.00000000	0.000001924800	32768
0.00010000	0.36292065	0.00000008	0.000030833333	2048
0.00100000	0.36292556	0.00000499	0.000248848614	256

Рисунок 4 — Результат вычисления интеграла методом трапеций

Видно, что данный метод для данной функции требует куда более мелкого разбиения, нежели чем метод прямоугольников, однако количество точек в таком разбиении значительно убывает с уменьшением требований по точности, куда быстрее, чем метод прямоугольников, хотя примерно с такой же зависимостью ($N \sim 2^{\frac{1}{\varepsilon}}$). Погрешность по Рунге, так же как и в методе прямоугольников, значительно меньше, чем вычисленное значение интеграла. График зависимости плотности разбиения N от заданной точности ε приведён на рисунке 5.

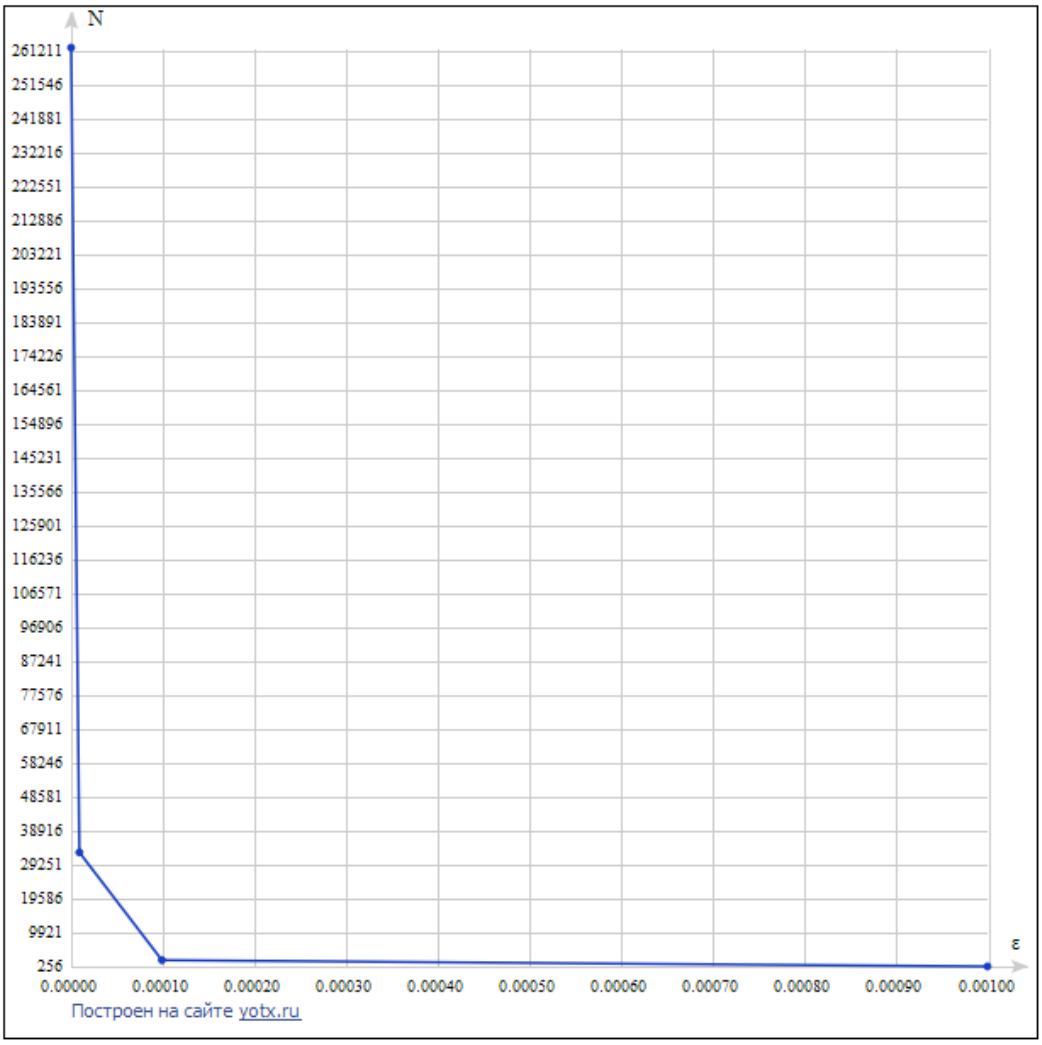


Рисунок 5 — график зависимости N от ε для метода трапеций

4. С помощью метода Симпсона было вычислено значение определённого интеграла, а также количество отдельно взятых точек на отрезке интегрирования функции. Результаты вычислений этим методом при различных значениях необходимой точности ε приведены на Рисунке 6.

Метод Симпсона:

ε	I	ΔI	Неточность по Рунге	N
0.00000100	0.36423137	0.00131079	0.000000612170	16
0.00001000	0.36860706	0.00568649	0.000010503523	8
0.00010000	0.36860706	0.00568649	0.000010503523	8
0.00100000	0.39387571	0.03095514	0.000193582152	4

Рисунок 6 — Результат вычисления интеграла методом Симпсона

Видно, что метод Симпсона для данной функции достаточно менее требователен к количеству точек на отрезке разбиения, и даже достаточно строгие требования точности по ε требуют 16 точек, и требования убывают с увеличением ε примерно в такой же зависимости, что и у предыдущих двух

методов. При этом неточность по Рунге остаётся настолько же меньшей вычисленного значения интеграла. График зависимости числа точек на отрезке интегрирования от заданной точности приведён на Рисунке 7.

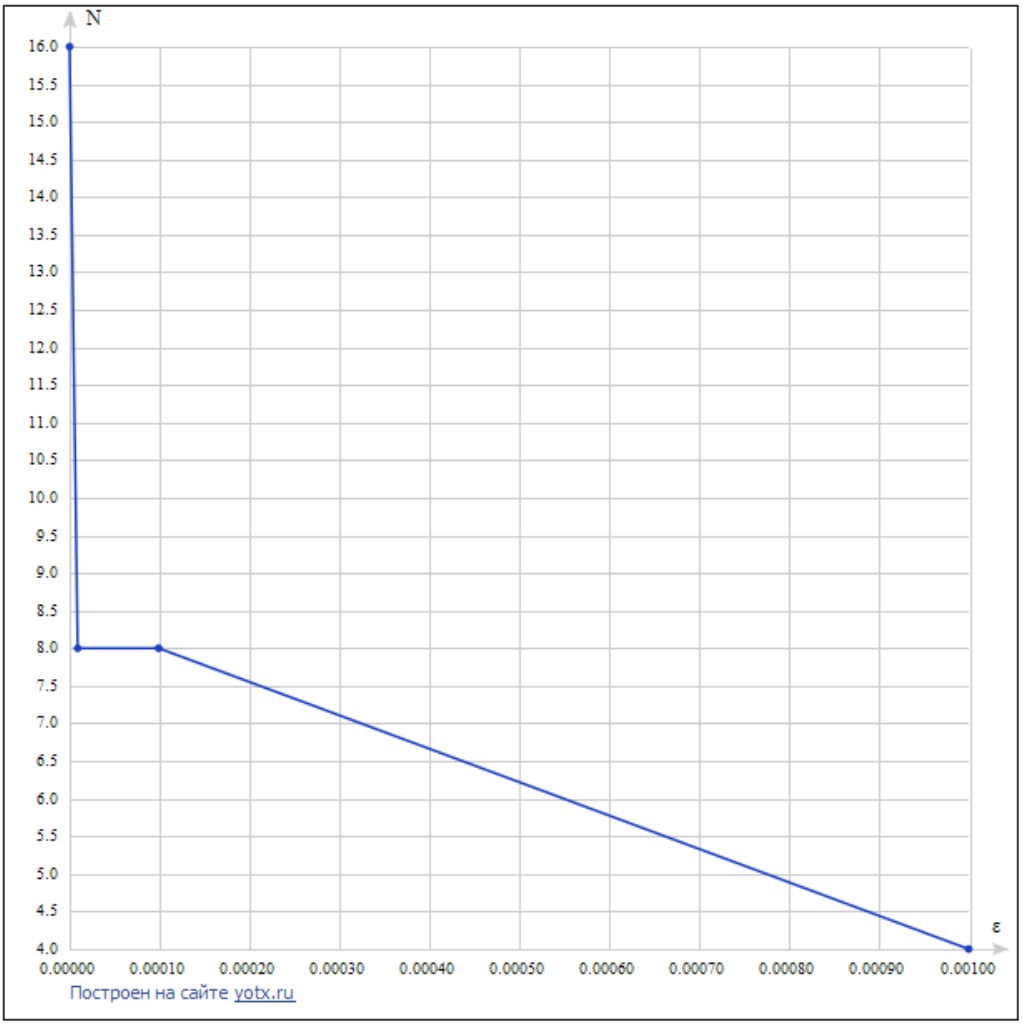
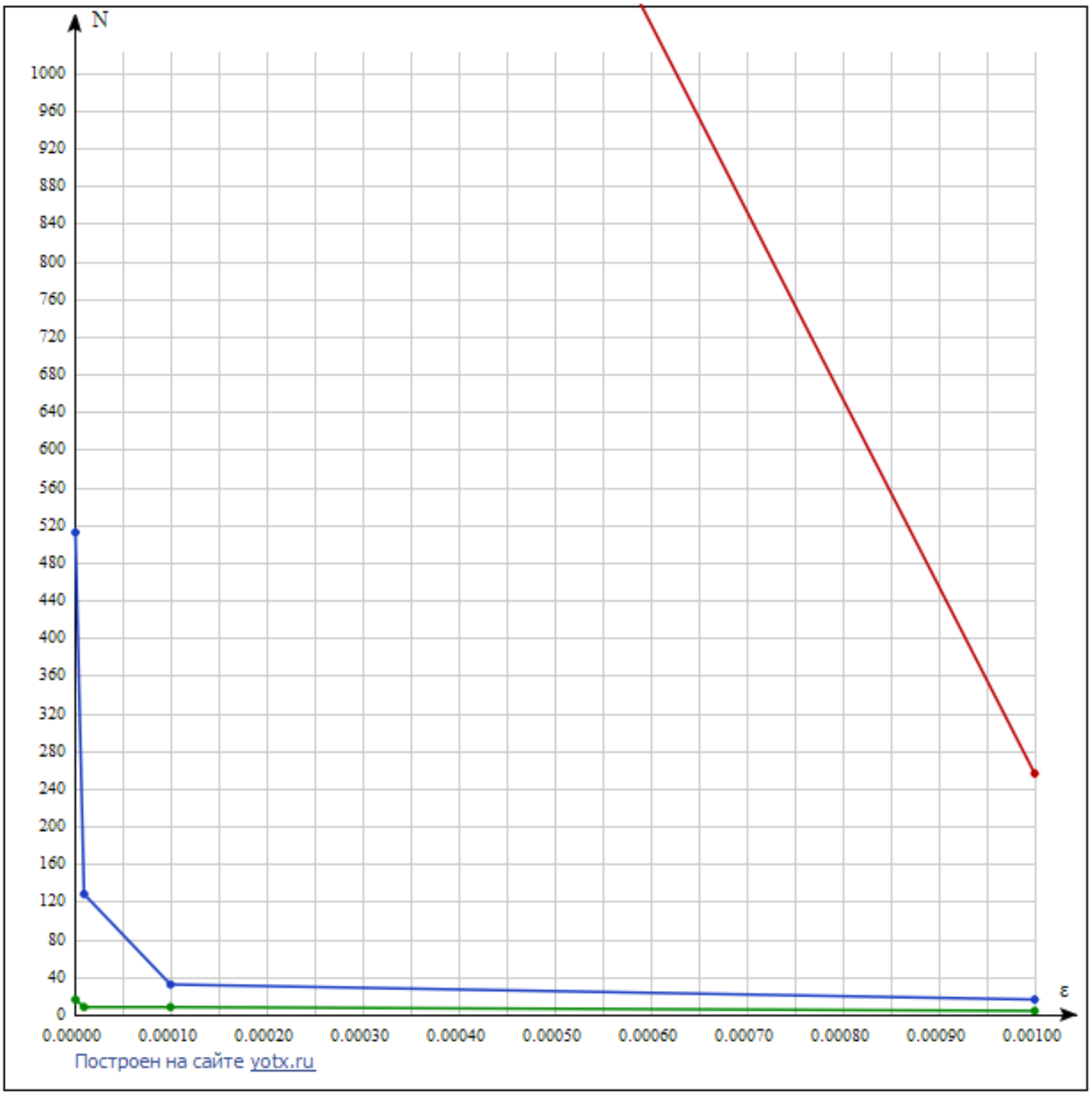


Рисунок 7 — график зависимости N от ε для метода Симпсона

Вывод.

Были рассмотренны различные методы вычисления определённого интеграла на примере интеграла $\int_0^1 \sin\left(x^4+2x^3+x^2\right)dx$. Было эксперементально подтверждено, что метод Симпсона требует куда менее плотного разбиения, нежели чем методы прямоугольника и трапеций, и это требование растёт куда медленее с устрожением требования к точности: это можно увидеть на графике зависимости N от ε для трёх функций, на Рисунке 8. Кроме того, оценка неточности по Рунге у метода Симпсона совпадает с таковыми у методов

прямоугольника и трапеций, и на несколько порядков меньше, чем вычисленное значение интеграла, что подтверждает корректность программной реализации данных методов.



метод прямоугольников

метод трапеций

метод Симпсона

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

```
#include <math.h>
#include <stdio.h>

double int_rect(double (*f)(double), double a, double b,
            unsigned n)
{
      double h = (b - a) / n;
      double xi = a;

      double I = 0;
      for(unsigned i = 1; i <= n; ++i, xi += h)
            I += f(xi + h/2);
      I *= h;
      return I;
}

double int_trap(double (*f)(double), double a, double b,
              unsigned n)
{
      double h = (b - a) / n;
      double xi = a;

      double I = (f(a) + f(b)) / 2;
      for(unsigned i = 1; i <= n; ++i, xi += h)
            I += (f(xi) + f(xi + h)) / 2;
      I *= h;
      return I;
}

double int_simps(double (*f)(double), double a, double b,
              unsigned n)
{
      double h = (b - a) / n;

      double I = 0;
      for(unsigned i = 0, xi = a; i < n; ++i)
            I += f(a + i*h) + 4*f(a + (i + 0.5)*h) + f(a + (i + 1)*h);
      I *= h / 6;
      return I;
```

```
}

double f(double x) { return sin(pow(x,4) + 2*pow(x,3) + pow(x,2)); }

double achieve_eps_int(double(*int_mt)(double (*)(double), double, double,
unsigned), double (*f)(double),
                  double a, double b, unsigned* n,
                  double inacc_divider, double target_val, double target_eps)
{
      double Ih = int_mt(f, a, b, *n);
      if(fabs(Ih - target_val) <= target_eps){
            double Ih2 = int_mt(f, a, b, (*n) * 2);
            return fabs(Ih2 - Ih) / inacc_divider;
      }
      for(;; (*n) *= 2){
            double Ih2 = int_mt(f, a, b, (*n) * 2);
            double eps = fabs(Ih2 - target_val);
            if(eps <= target_eps)
                  return fabs(Ih2 - Ih) / inacc_divider;
            Ih = Ih2;
      };
}


int main()
{
      double a = 0, b = 1;
      double real_int_val = 0.3629205713226523; // "точное" значение интеграла
      double teps = 1e-6; // искомое ε (разница между "точным" и вычисленным
приблизительным значением

      printf("Метод прямоугольников:\n");
      printf("%13s%13s%13s%40s%8s\n", "ε", "I", "ΔI", "Неточность по Рунге",
"N");
      for(double eps = 1e-6; eps < 0.01; eps *= 10)
      {
            unsigned n = 2;
            double inacc = achieve_eps_int(&int_rect, &f,
                              a, b, &n,
                              3, real_int_val, eps);
            double I = int_rect(&f, a, b, n);
            printf("%12.8lf%13.8lf%12.8lf%23.12lf%8u\n", eps, I,
fabs(real_int_val - I), inacc, n);
```

```
	}

	printf("\nМетод трапеций:\n");
	printf("%13s%13s%13s%40s%8s\n", "ε", "I", "ΔI", "Неточность по Рунге",
"N");
	for(double eps = 1e-6; eps < 0.01; eps *= 10)
	{
		unsigned n = 2;
		double inacc = achieve_eps_int(&int_trap, &f,
					a, b, &n,
					3, real_int_val, eps);
		double I = int_rect(&f, a, b, n);
		printf("%12.8lf%13.8lf%12.8lf%23.12lf%8u\n", eps, I,
fabs(real_int_val - I), inacc, n);
	}

	printf("\nМетод Симпсона:\n");
	printf("%13s%13s%13s%40s%8s\n", "ε", "I", "ΔI", "Неточность по Рунге",
"N");
	for(double eps = 1e-6; eps < 0.01; eps *= 10)
	{
		unsigned n = 2;
		double inacc = achieve_eps_int(&int_simps, &f,
					a, b, &n,
					15, real_int_val, eps);
		double I = int_rect(&f, a, b, n);
		printf("%12.8lf%13.8lf%12.8lf%23.12lf%8u\n", eps, I,
fabs(real_int_val - I), inacc, n);
	}
}
```