Отчет по дисциплине: «Численные методы»

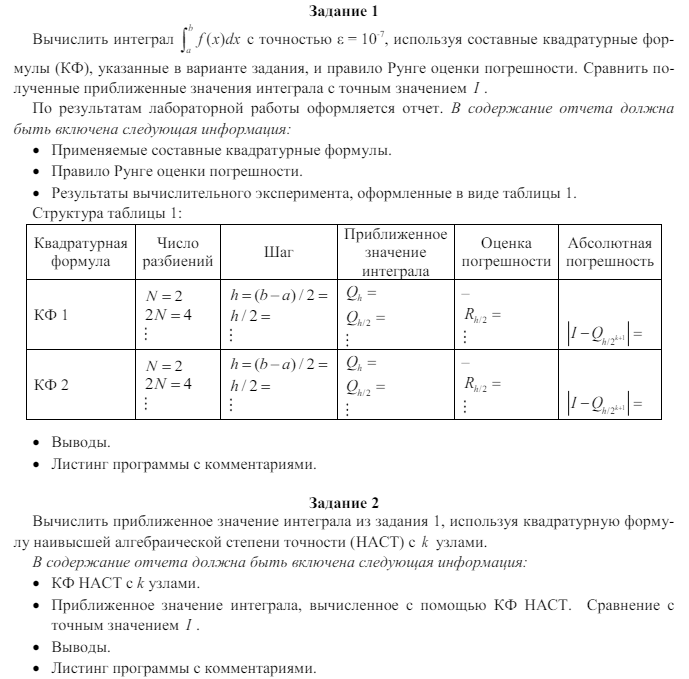
Лабораторная работа №3

## «Приближенное вычисление интегралов»

Подготовил студент 3 курса 4 группы

Кондратович Артём

Условие

  
Вариант 4:





Листинг программы:

namespace lab\_2\_3

{

internal class Program

{

delegate double QF(int n);

private const double A = 0.0;

private const double B = Math.PI / 4.0;

private const double \_epsilon = 0.0000001;

private static readonly double \_accurateValue = Math.PI / (3 \* Math.Sqrt(3));

private const int K = 4;

private static readonly Func<double, double> \_f = (x) => 1 / (1 + 2 \* Math.Pow(Math.Sin(x), 2));

static void Main()

{

var answer = Runge(QuadratureFormulaOfAverageRectangles, 2, 2, 1);

PrintTable(1, 0, double.NaN, double.NaN, double.NaN, Math.Abs(\_accurateValue - answer));

Console.WriteLine();

answer = Runge(SimpsonsQuadratureFormula, 4, 2, 2);

PrintTable(2, 0, double.NaN, double.NaN, double.NaN, Math.Abs(\_accurateValue - answer));

var a = new List<double>

{

0.34785484,

0.65214516,

0.65214516,

0.34785484

};

var t = new List<double>

{

-0.86113631,

-0.33998104,

0.33998104,

0.86113631

};

answer = Nast(a, t);

Console.WriteLine();

Console.WriteLine($"|I - Nast|: { Math.Abs(\_accurateValue - answer).ToString("0.0000000000")}");

}

//КФ средних прямоугольников

static double QuadratureFormulaOfAverageRectangles(int n)

{

var h = (A + B) / n;

var x = new List<double>();

for (var i = 0; i <= n; i++)

{

x.Add(A + h\*i);

}

var sum = 0.0;

for (var i = 0; i < n; i++)

{

sum += \_f((x[i + 1] + x[i]) / 2.0);

}

return sum \* h;

}

//КФ Симпсона

static double SimpsonsQuadratureFormula(int n)

{

var h = (A + B) / n;

var x = new List<double>();

for (var i = A; i <= B; i += h)

{

x.Add(i);

}

var sum = 0.0;

for (var i = 0; i < n; i++)

{

sum += \_f(x[i]) + 4 \* \_f((x[i] + x[i + 1]) / 2.0) + \_f(x[i + 1]);

}

return sum \* h / 6.0;

}

static double Nast(List<double> a, List<double> t)

{

var x = t.Select(t => (A + B) / 2.0 + (B - A) / 2.0 \* t).ToList();

var sum = 0.0;

for (var i = 0; i < K; i++)

{

sum += a[i] \* \_f(x[i]);

}

return sum \* (B - A) / 2.0;

}

static double Runge(QF qf, int m, int n, int type)

{

var q1 = qf(n);

var q2 = qf(2 \* n);

var r = (q2 - q1) / (Math.Pow(2, m) - 1);

if (Math.Abs(r) <= \_epsilon)

{

return q2 + (int)r;

}

else

{

PrintTable(type, n, (A + B) / n, q1, r, double.NaN);

return Runge(qf, m, 2 \* n, type);

}

}

public static void PrintTable(int type, int n, double h, double qh, double r, double error)

{

var kf = type switch

{

1 => "СКФСП",

2 => "СКФС"

};

Console.WriteLine("{0,-5}{1,-5}{2,-10}{3,-10}{4,-10}{5,-10}{6,-10}{7,-10}{8,-10}{9,-10}{10,-10}",

kf,

" n:", n,

" h:", h.ToString("0.00000000"),

" qh:", qh.ToString("0.00000000"),

" r:", r.ToString("0.00000000"),

" |I - Qh|:", error.ToString("0.00000000"));

}

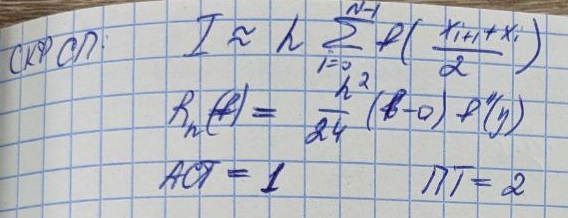
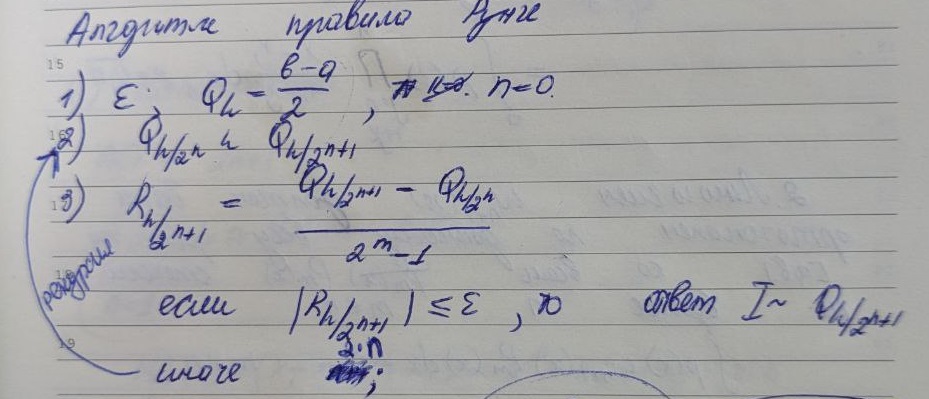
}

}

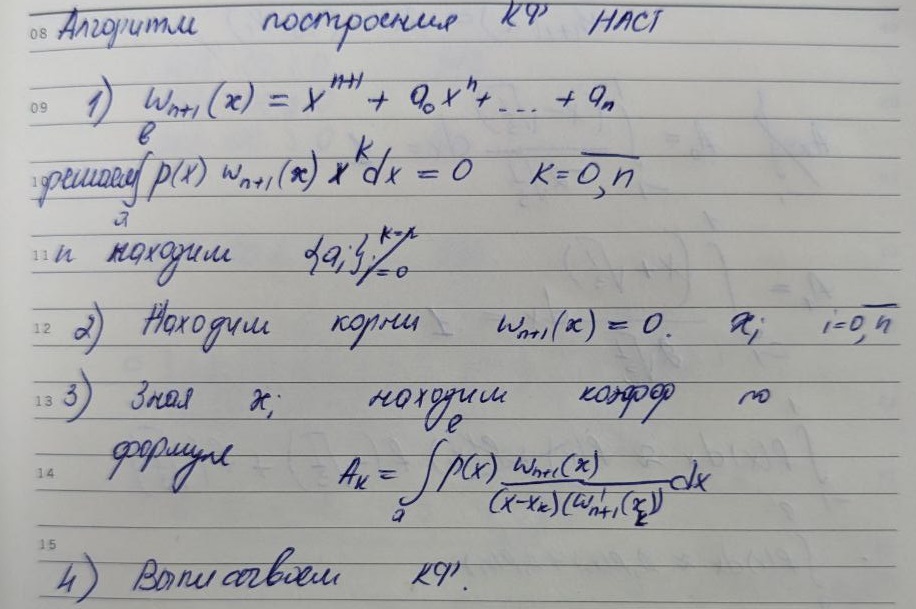
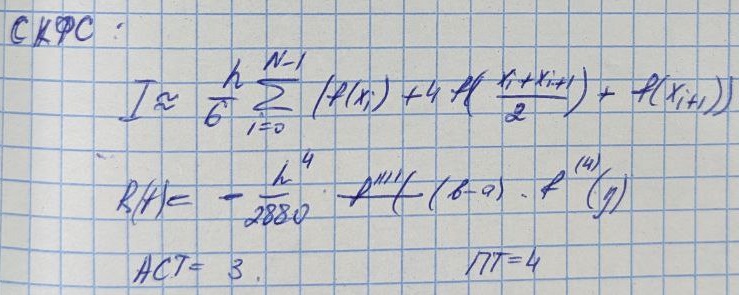
Теория:

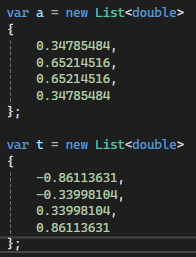
Алгоритм правила Рунге:

СКФ СП:



СКФ С:  
  
Алгоритм построения НАСТ:  
  
Полученные коэффициенты и узлы на отрезке [-1, 1]:



  
Переход к отрезку [0, ]:  
  
  
Результаты:

