МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ» ИМ. УЛЬЯНОВА (ЛЕНИНА)

Кафедра МО ЭВМ

ОТЧЕТ

по практической работе №3

по дисциплине «Вычислительная математика»

Тема: Численное интегрирование

Студент гр. 8301	____________________	Готовский К.В.
Преподаватель	____________________	Сучков А.И
	____________________	Колоницкий С.Б.

Санкт-Петербург

2020

Цель работы.

Изучение и сравнение различных методов численного интегрирования на примере составных формул прямоугольников, трапеций и Симпсона.

Основные теоретические положения.

Для повышения точности интегрирования применяют составные формулы. Для этого разбивают отрезок $[a, b]$ на четное $n = 2m$ число отрезков длины $h = \frac{(b-a)}{n}$ и на каждом из отрезков длины $2h$ применяют соответствующую формулу. Таким образом получают составные формулы прямоугольников, трапеций и Симпсона.

На сетке $x_i = a + i * h, y_i = f(x_i), i = \overline{0{,}2m}$ составные формулы имеют следующий вид:

- формула прямоугольников: $\int_a^b f(x)dx = h\sum_{i=0}^{n-1} f\left(x_i + \frac{h}{2}\right) + R_1$, где $R_1 \approx \frac{h^2}{24}\int_a^b f''(x)dx = o(h^2)$;
- формула трапеций: $\int_a^b f(x)dx = \frac{h}{2}\sum_{i=0}^{n-1}(f(x_i) + f(x_{i+1})) + R_2$, где $R_2 \approx -\frac{h^2}{12}\int_a^b f''(x)dx = o(h^2)$;
- формула Симпсона: $\int_a^b f(x)dx = \frac{h}{3}\sum_{i=0}^{m-1}(f(x_{2i}) + 4f(x_{2i+1}) + f(x_{2i+2})) + R_3$, где $R_3 \approx -\frac{h^2}{180}\int_a^b f^{IV}(x)dx = o(h^4)$;

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

- для формулы прямоугольников;
- для формулы трапеций;
- для формулы Симпсона;

Постановка задачи.

Используя квадратурные формулы прямоугольников, трапеций и Симпсона, вычислить значения заданного интеграла и, применив правило Рунге, найти наименьшее значение n (наибольшее значение шага h), при котором каждая из указанных формул дает приближенное значение интеграла с погрешностью ε, не превышающей заданную.

1. Составить подпрограмму-функцию `F` для вычисления подынтегральной функции.
2. Составить подпрограммы-функции `RECT`, `TRAP`, `SIMPS` 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 ($\varepsilon = 0.01, 0.001, 0.0001$), и осуществляющих печать результатов значения интеграла и значения n для каждой формулы.
4. Провести вычисления по программе, добиваясь, чтобы результат удовлетворял требуемой точности.

Таблицы для оценки погрешности.

Таблица 1 – Таблица значений для равномерных узлов.

Значение ε	0,01	0,001	0,0001
Значение n для формулы прямоугольников	8	512	4096
Значение интеграла для формулы прямоугольников	0.23994	0.36194	0.36280

Значение n для формулы трапеций	16	32	128
Значение интеграла для формулы трапеций	0.35706	0.36157	0.36284
Значение n для формулы Симпсона	8	16	512
Значение интеграла для формулы Симпсона	0.37463	0.39161	0.36448

Сравнительная оценка применяемых для вычисления формул.

При повышенных точностях быстрее всего сближает формула трапеции. При любых приближениях формула треугольника требует большего разбиения. При низкой точности формула Симпсона быстрее всего достигает нужного результата.

Выводы.

Я изучил и сравнил различные методы численного интегрирования на примере составных формул прямоугольников, трапеций и Симпсона.

ПРИЛОЖЕНИЕ А

Программа в режиме для точности 0.01

```
function [intRECT, n1, intTRAP, n2, intSIMPS, n3] = Yrok_3

  function [y] = F(x)
    y = sin(x^2 + 2*x^3 + x^4);
  endfunction

  function [s1] = RECT(h1, n1, x)
    s1 = 0;
    for i = 1:n1
      s1 = s1 + F(x(i) + h1/2);
    endfor
    s1 = s1*h1;
  endfunction

  function [s2] = TRAP(h2, n2, x)
    s2 = 0;
    for i = 1:(n2-1)
      s2 = s2 + F(x(i)) + F(x(i+1));
    endfor
    s2 = s2*h2/2;
  endfunction

    function [s3] = SIMPS(h3, n3, x)
    s3 = 0;
    for i = 1:(n3/2 - 0.5)
      s3 = s3 + F(x(2*i-1)) + 4*F(x(2*i)) + F(x(2*i+1));
    endfor
    s3 = s3*h3/3;
  endfunction

  %%%%%%%%%%%%%%%%%%%%%
  a = 0; b = 1; n = 2; h = (b - a)/n; eps = 0.01;
  n1 = n; n2 = n; n3 = n; h1 = h; h2 = h; h3 = h;
  rect1 = a; rect2 = b; trap2 = a; trap1 = b; simps1 = a; simps2 =
b;
  for i = 1:n
    x(i) = a + i*h;
  endfor

  while (abs(rect1 - rect2)/3) > eps
```

```
    rect1 = RECT(h1, n1, x);
    n1 = n1*2;
    h1 = (b - a)/n1;
    for i = 1:n1
      x(i) = a + i*h1;
    endfor
    rect2 = RECT(h1, n1, x);
  endwhile
  intRECT = rect2 + (rect2 - rect1)/3;

while (abs(trap2 - trap1)/3) > eps
    trap1 = TRAP(h2, n2, x);
    n2 = n2*2;
    h2 = (b - a)/n2;
    for i = 1:n2
      x(i) = a + i*h2;
    endfor
    trap2 = TRAP(h2, n2, x);
  endwhile
  intTRAP = trap2 - (trap2 - trap1)/3;

while (abs(simps2 - simps1)/15) > eps
    simps1 = SIMPS(h3, n3, x);
    n3 = n3*2;
    h3 = (b - a)/n3;
    for i = 1:n3
      x(i) = a + i*h3;
    endfor
    simps2 = SIMPS(h3, n3, x);
  endwhile
  intSIMPS = simps2 - (simps2 - simps1)/15;
endfunction
```