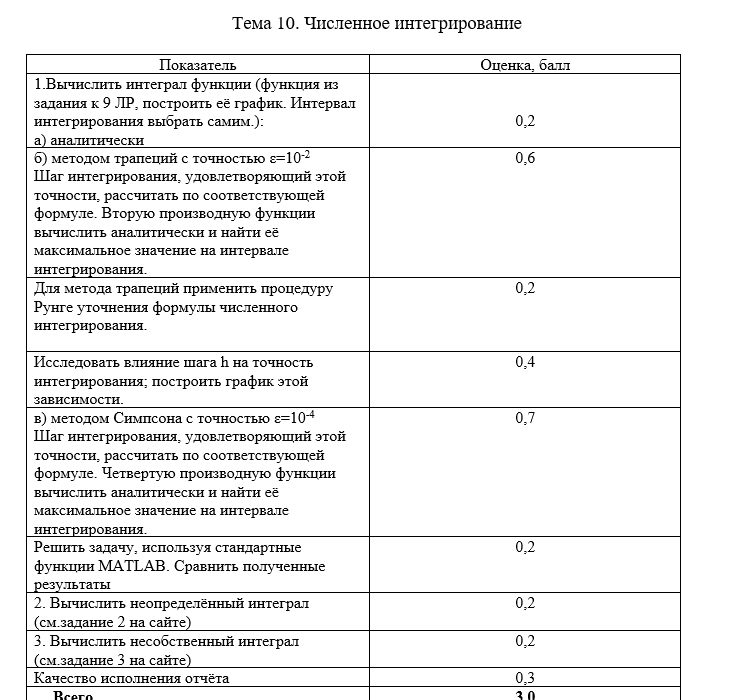
*Дубкова Валерия КС-26*

**10-ая лабораторная работа**

**Задание**



**Дополнительное задание (из 9 работы)**

**Код Lab10.m**

clc; clear all;

a = 0;

b = 1.4;

h = 0.001;

x = [a:h:b];

anAns = 1.753031574;

y = f(x);

fprintf('Ответ аналитически: %f', anAns);

figure;

plot(x, y, 'r');

grid on

xlabel('x'); ylabel('y');

title('График функции');

legend('f(x)');

% Вычисление интеграла методом трапеций с определённой точностью

e = 1e-2;%точность

% m2 = max(ddy);

% h = sqrt((12\*e) / ((b-a)\*m2));

h = 0.2;

n = 0; %счетчик итераций

trpzAns = 0;

oldAns = 0;

while 1

for i = a:h:(b-h)

r = -((h^3)/12)\*funcddy(i);

tempS = (h/2)\*(f(i) + f(i + h)) + r;

trpzAns = trpzAns + tempS;

n = n + 1;

end

if abs(oldAns - trpzAns)< e

break;

else

h = h/2;

oldAns = trpzAns;

trpzAns = 0;

n = 0;

end

end

fprintf('\nОтвет методом Трапеций с точностью %.2f: %f\n\tn = %d; h=%.3f',...

e, trpzAns, n, h);

h = 0.4;

for j = 1:1:10

newTrpz = 0;

for i = a:h:(b-h)

r = -((h^3)/12)\*funcddy(i);

tempS = (h/2)\*(f(i) + f(i + h)) + r;

newTrpz = newTrpz + tempS;

end

eList(j) = abs(anAns - newTrpz);

hs(j) = h;

h = h/2;

end

fprintf('\nМассив погрешностей зависимости от величины шага:\n');

fprintf('%f\n', eList);

figure;

plot(hs, eList, '-\*r');

axis([-0.05, 0.45, -0.1, 0.35]);

grid on

hold off

legend('Погрешность');

legend('Location', 'northwest');

title('График зависимости погрешности от вечиличны шага');

xlabel('Величина шага'); ylabel('Погрешность');

% Решение с помощью метода Симпсона

e = 1e-4;

h = 0.2;

n = 0;

oldAns = 0;

sympAns = 0;

while 1

for i = (a+h):2\*h:(b-h)

r = -((h.^5)/90).\*d4y(i);

tempS = (h/3)\*(f(i-h) + 4\*f(i)+ f(i + h));

sympAns = sympAns + tempS;

n = n + 1;

end

if abs(oldAns - sympAns)< e

break;

else

h = h/2;

oldAns = sympAns;

sympAns = 0;

n = 0;

end

end

fprintf('\nОтвет методом Симпсона с точностью %.4f: %f\n\tn = %d; h=%.4f',...

e, sympAns, n, h);

% Вычисление интеграла методом трапеций c помощью встроенных функций

trpz = trapz(x, y);

fprintf('\nОтвет методом Трапеций встроенной функцией ML: %f', trpz);

% Вычисление интеграла методом Симпсона c помощью встроенных функций

e = 1e-4;

smp = quad('cos(x).\*exp(x.^2)',a, b, e);

fprintf('\nОтвет методом Симпсона встроенной функцией ML: %f', smp);

% Сравнение полученных результатов

fprintf('\nПогрешности в сравнении с аналитическим ответом:\n');

fprintf('\tМетод трапеций: %f\n', anAns-trpzAns);

fprintf('\tМетод Симпсона: %f\n', anAns-sympAns);

fprintf('\tВстроенный метод трапеций: %f\n', anAns-trpz);

fprintf('\tВстроенный метод Симпсона: %f\n', anAns-smp);

function y = f(m)

y = cos(m).\*exp(m.^2);

end

function y = funcddy(m)

y = 2\*m\*exp(m.^2)\*cos(m) - exp(m.^2)\*sin(m);

end

function y = d4y(m)

y = (-12\*(1 + 2\*m^2)\*cos(m) + 4\*(3 + 4\*m^4 + 12\*m^2)\*cos(m) + 8\*m\*sin(m)...

- 16\*m\*(3 + 2\*m^2)\*sin(m) + cos(m))\*exp(m^2);

end

**z2\_3.m**

clc

%несобственный интеграл

syms a p x

int(a^x\*exp(-x),x)

%неопределенный интеграл

int((1+x)/((x+a)^(p+1)), x, 0, inf)

**Результат выполнения программы**

**Lab10.m**

*Ответ аналитически: 1.753032*

*Ответ методом Трапеций с точностью 0.01: 1.749737*

*n = 14; h=0.100*

*Массив погрешностей зависимости от величины шага:*

*0.292479*

*0.013279*

*0.003295*

*0.000813*

*0.000201*

*0.000050*

*0.000012*

*0.000003*

*0.000001*

*0.000000*

*Ответ методом Симпсона с точностью 0.0001: 1.753026*

*n = 14; h=0.0500*

*Ответ методом Трапеций встроенной функцией ML: 1.753031*

*Ответ методом Симпсона встроенной функцией ML: 1.753011*

*Погрешности в сравнении с аналитическим ответом:*

*Метод трапеций: 0.003295*

*Метод Симпсона: 0.000006*

*Встроенный метод трапеций: 0.000000*

*Встроенный метод Симпсона: 0.000021*

**z2\_3.m**

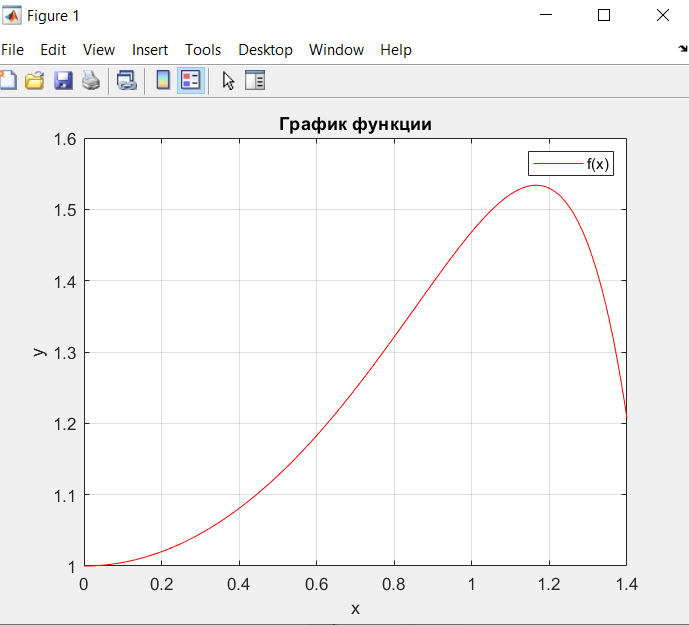
*ans =*

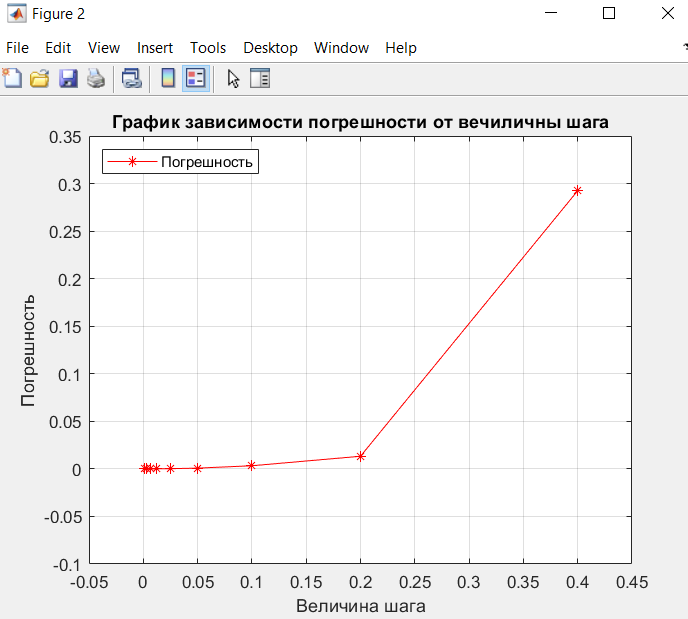
*(a^x\*exp(-x))/(log(a) - 1)*

*ans =*

*piecewise(in(p, 'integer') & ~a <= 0 & 2 <= p, (a + p - 1)/(a^p\*p\*(p - 1)), (~a <= 0 | p <= -1) & in(p, 'integer') & p <= 0, Inf + (a + p - 1)/(a^p\*p\*(p - 1)), a\*p + 1 < a + p & in(p, 'integer') & a <= 0 & 2 <= p, Inf - limit((a + p + x\*p - 1)/(a + x)^p, x, -a, 'Left')/(p\*(p - 1)), ~in(p, 'integer') | (1 <= p | a <= 0 & 0 <= p) & (a <= 0 | p <= 1) & (a + p <= a\*p + 1 | ~a <= 0 | p <= 1), int((x + 1)/(a + x)^(p + 1), x, 0, Inf))*

**Графики**





**Выводы**

Каждый из представленных методов прекрасно справился со своей задачей. Можно отметить, что метод Симпсона сделал столько же итераций, как метод трапеций, хотя и точность в первом случае была намного выше. Так же стоит отметь, что в методе Симпсона большее количество расчётов, так что, лучше смотреть для каждой задачи, как будет сделать быстрее.

По второму графику видим, что тем меньше шаг, тем меньше будет погрешность, точнее будет конечный ответ.