

Задание: вычислить значение функции $V(x) = \int_0^x \frac{\cos(t) - 1}{t}$ на отрезке [0,5] с шагом h, используя квадратурные формулы правых прямоугольников, трапеций, Симпсона и Гаусса.

Суть метода заключается в разбиении площади под графиком функции на геометрические фигуры, площадь которых можно без проблем найти. В методе Симпсона в качестве такой фигуры используется парабола с вершиной в середине отрезка $[x_i, x_{i+1}]$. Формула Гаусса использует трапеции, но сетка сдвигается. Ниже приведены формулы для каждого из методов.

Формула средних прямоугольников:

$$\int_a^b f(x) = f(b) * (b - a)$$

Формула трапеций:

$$\int_a^b f(x) = \frac{(f(a) + f(b))}{2} * (b - a)$$

Формула Симпсона:

$$\int_a^b f(x) = \left(f(a) + f(b) + 4 \left(f\left(\frac{a+b}{2}\right) \right) \right) * \frac{(b-a)}{6}$$

Формула Гаусса:

$$a' = a + \frac{b-a}{2} \left(1 - \frac{1}{\sqrt{3}} \right)$$

$$b' = a + \frac{b-a}{2} \left(1 + \frac{1}{\sqrt{3}} \right)$$

$$\int_a^b f(x) = \frac{(f(a') + f(b'))}{2} (b - a)$$

Вычисление функции происходит следующим образом:

Отрезок разбивается на n промежутков, и по одной из формул считается площадь каждого из них.

n находится путём вычисления разницы между S(n) и S(2n), где S(t) - функция разбиения площади на t частей. n увеличивается до тех пор, пока не достигнет максимального значения или пока разность не станет меньше заданного значения.

В качестве точных значений функции взяты значения, найденные по полиному Тейлора.

Вывод таблиц погрешностей для каждой формулы:

```
clear all;
close all;
clc;
x=0:0.5:5;
```

```
yt=intcos(x,1e-20);
%Средние прямоугольники
[y,n]=Func(x,@Pravpr,@F1,1e-6,500);
t=table(x',y',(y-yt)',n','VariableNames',{'x','y','Погрешность','n'});
display(t);
```

t = 11x4 table

	x	y	Погрешность	n
1	0	0	0	5
2	0.5000	-0.0619	-4.7856e-05	640
3	1	-0.2400	-1.7970e-04	640
4	1.5000	-0.5127	-3.6325e-04	640
5	2	-0.8479	-5.5353e-04	640
6	2.5000	-1.2083	-7.0395e-04	640
7	3	-1.5570	-7.7764e-04	640
8	3.5000	-1.8629	-7.5655e-04	640
9	4	-2.1051	-6.4581e-04	640
10	4.5000	-2.2753	-4.7254e-04	640
11	5	-2.3770	-2.7912e-04	640

%Трапеции

```
[y,n]=Func(x,@Trap,@F1,1e-6,500);
t=table(x',y',(y-yt)',n','VariableNames',{'x','y','Погрешность','n'});
display(t);
```

t = 11x4 table

	x	y	Погрешность	n
1	0	0	0	5
2	0.5000	-0.0619	1.0288e-07	40
3	1	-0.2398	9.6817e-08	160
4	1.5000	-0.5123	1.1389e-07	320
5	2	-0.8474	3.2604e-07	320
6	2.5000	-1.2076	1.7473e-07	640
7	3	-1.5562	3.0905e-07	640
8	3.5000	-1.8621	4.7321e-07	640
9	4	-2.1045	6.4599e-07	640
10	4.5000	-2.2748	8.0156e-07	640
11	5	-2.3767	9.1752e-07	640

% %Симпсона

```
[y,n]=Func(x,@Simp,@F1,1e-6,500);
```

```
t=table(x',y',(y-yt)',n','VariableNames',{'x','y','Погрешность','n'});
display(t);
```

t = 11x4 table

	x	y	Погрешность	n
1	0	0	0	5
2	0.5000	-0.0619	-6.7844e-11	5
3	1	-0.2398	-4.1429e-09	5
4	1.5000	-0.5123	-2.1944e-09	10
5	2	-0.8474	-1.1017e-08	10
6	2.5000	-1.2076	-3.6265e-08	10
7	3	-1.5562	-5.0691e-09	20
8	3.5000	-1.8621	-1.0206e-08	20
9	4	-2.1045	-1.7373e-08	20
10	4.5000	-2.2748	-2.5633e-08	20
11	5	-2.3767	-3.3296e-08	20

%Гauss

```
[y,n]=Func(x,@Gaus,@F1,1e-6,500);
t=table(x',y',(y-yt)',n','VariableNames',{'x','y','Погрешность','n'});
display(t);
```

t = 11x4 table

	x	y	Погрешность	n
1	0	0	0	5
2	0.5000	-0.0619	4.5230e-11	5
3	1	-0.2398	2.7620e-09	5
4	1.5000	-0.5123	2.9073e-08	5
5	2	-0.8474	7.3450e-09	10
6	2.5000	-1.2076	2.4177e-08	10
7	3	-1.5562	3.3794e-09	20
8	3.5000	-1.8621	6.8038e-09	20
9	4	-2.1045	1.1582e-08	20
10	4.5000	-2.2748	1.7089e-08	20
11	5	-2.3767	2.2198e-08	20

Функция для ряда Тейлора:

```
function [y,ki] = intcos(x,e)
y = zeros(size(x));
```

```

kmax = 100;
if nargin == 1
    e = 1e-15;
end
if nargout == 2
    ki = y;
end
for i = 1:numel(x)
    xt = x(i);
    a = -xt.*xt./4;
    sum = a;
    k = 1;
    while(abs(a) >= e) && (k < kmax)
        q = -xt.* xt.* k / ((k + 1) * (2 * k + 1) * (2 * k + 2));
        k = k + 1;
        a = a* q;
        sum = sum + a;
    end
    y(i) = sum;
    if nargout == 2
        ki(i) = k;
    end
end
end

```

Функция для формулы правых прямоугольников:

```

function [y] =Pravpr (f,a,b,n)
y=0;
t=linspace(a,b,n);
h=t(2)-t(1);
for j=1:numel(t)-1
    y=y+h*f(t(j+1));
end
end

```

Функция для формулы трапеций:

```

function [y] =Trap (f,a,b,n)
y=0;
t=linspace(a,b,n);
h=t(2)-t(1);
for j=1:numel(t)-1
    y=y+h*(f(t(j))+f(t(j+1)))/2;
end
end

```

Функция для формулы Симпсона:

```

function [y] =Simp(f,a,b,n)

```

```

y=0;
t=linspace(a,b,n);
h=(t(2)-t(1))/6;
for j=1: numel(t)-1
    y=y+h*(f(t(j))+f(t(j+1))+4*f((t(j)+t(j+1))/2));
end
end

```

Функция для формулы Гаусса:

```

function [y] =Gaus(f,a,b,n)
y=0;
t=linspace(a,b,n);
h=t(2)-t(1);
for j=1: numel(t)-1
    t1=t(j)+(t(j+1)-t(j))/2*(1-1/sqrt(3));
    t2=t(j)+(t(j+1)-t(j))/2*(1+1/sqrt(3));
    y=y+(f(t1)+f(t2))/2*h;
end
end

```

Функция, которая находит число разбиений(n) для достижения заданной точности. В качестве параметра принимает одну из функций, описывающий метод:

```

function [y,n] = Funct(x,method,f,e,maxn)
y=zeros(size(x));
n=ones(size(x))*5;
for i=1: numel(x)
    yn=method(f,0,x(i),n(i));
    y2n=method(f,0,x(i),2*n(i));
    while((abs(yn-y2n)>e)&&(n(i)<maxn))
        n(i)=n(i)*2;
        yn=y2n;
        y2n=method(f,0,x(i),2*n(i));
    end
    y(i)=y2n;
end
end

```

Функция интегрального косинуса:

```

function [y] = F1(x)
if (x < 1e-6)
    y = 0;
else
    y=(cos(x) - 1) / x;
end
end

```

Вывод:

Как можно заметить из таблиц, формулы Гаусса и Симпсона оказались наиболее точными для вычисления интеграла. Ошибка при вычислениях связана с тем, что площадь каждого отрезка считается с некоторой погрешностью, и эти погрешности суммируются.