Лабораторная работа 1

Вычисление определённых интегралов
методом Монте-Карло

Выполнил:
студент гр. P4106
Игнашов Иван Максимович
Вариант 8

1. Цель работы

Изучение метода Монте-Карло, определение точности вычисления определенных интегралов методом Монте-Карло.

Порядок работы:

1. Записать математически анализируемую функцию

$$f_{res} = \begin{cases} 5 * sin(2\pi t) + 1 & t < 1 \\ 5 * sin(2\pi(t-1)) + 1 & 1 \leq t \leq 2 \\ 2,5 * \frac{2}{(t-2)+1} & t > 2 \end{cases} \quad (1)$$

2. Вычислить аналитически определенный интеграл $F = \int_0^3 f_{res}(t)dt$

3. Разработать программу, вычисляющую величину F методом Монте-Карло при заданном числе экспериментов

4. При помощи разработанной программы вычислить определенный интеграл $\hat{F}$ при $N = 2^i$ экспериментах, где i = 0...14

2. График функции $f_{res}(t)$

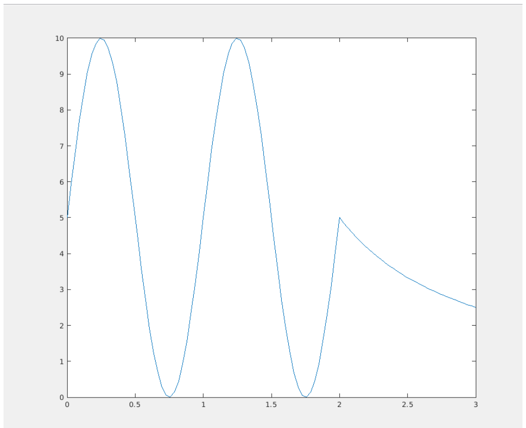

Рис. 1: График функции $f_{res}(t)$

Можно заметить, что на интересующей области $t = [0, 3]$ функция принимает значения от 0 до 10

3. Аналитический расчет величины F

Проинтегрируем кусочно-заданную функцию отдельно для каждого участка

	$f_1(t)$	$f_2(t)$	$f_3(t)$
Функция	$5 * sin(2\pi t) + 1$	$5 * sin(2\pi(t-1)) + 1$	$2,5 * \frac{2}{(t-2)+1}$
Неопр. интеграл	$5t - \frac{5cos(2\pi t)}{2\pi}$	$5t - \frac{5cos(2\pi t)}{2\pi}$	$5\log(t-1)$
Область	$0 \leq t \leq 1$	$1 \leq t \leq 2$	$2 \leq t \leq 3$
Значение	5.0	5.0	3.46

Просуммировав получим $F = 13.46574$

4. Описание разработанной программы

```matlab
% Лабораторная работа 1
% Вычисление определённых интегралов методом Монте-Карло

function lab1
    %x = linspace(0, 3);
    %plot(x, arrayfun(@(x) piecewise_func_lab1(x), x));

    MK_params = 1:14;
    [results_S, results_P] = arrayfun(@(i) MKsolve(2^i), MK_params)
    real_val = 5 + 5 + 3.46574;

    plot(MK_params,results_S,'r', ...
        MK_params, results_P, 'g', ...
        MK_params, real_val*ones(size(MK_params)), 'm');
    title('Сравнение методов')
    legend('Простой МК', 'Точный МК', 'Реальная величина');
    xlabel('log(Количества точек)')
    ylabel('Вычисленное значение')

end
```

Рис. 2: Код сценария перебора экспериментов

Скрипт-сценарий ***lab1.m*** - точка входа программы, предназначен для запуска всех остальных функции и скриптов, а так же для вывода графика результатов работы.

Основные переменные ***lab1.m***:

- *results_S* - результаты оценки интеграла простым методом Монте-Карло для разных количеств экспериментов
- *results_P* - результаты оценки интеграла методом Монте-Карло с повышенной точностью для разных количеств экспериментов
- *real_val* - реальное значение интеграла

```
% Функция для Варианта 8
% Кусочно-заданная функция:
%   веса   5; 5; 2,5
%   номера 1; 1; 4

function y = piecewise_func_lab1(t)
    function y = f1(t)
        y = sin(2 * pi * t) + 1;
    end

    function y = f2(t)
        y = 2 * t - 1;
    end

    function y = f3(t)
        y = 4 * t^2 - 1;
    end

    function y = f4(t)
        y = 2 / (t + 1);
    end


    if (t < 1)
        y = 5 * f1(t);
    elseif (1 <= t && t <= 2)
        y = 5 * f1(t - 1);
    else
        y = 2.5 * f4(t - 2);
    end
end
```

Рис. 3: Код кусочно-заданной функции для Варианта 8

Скрипт-функция ***piecewise_func_lab1.m*** - записанная в MatLab функция $f_{res}(t)$, интеграл которой и необходимо проанализировать.

Основные переменные ***piecewise_func_lab1.m***:

- *f1(t), f2(t), f3(t), f4(t)* - функции расчёта из перечня для вариантов лабораторной работы

```
% Методы Монте-Карло для вычисления интеграла кусочно-заданной функции
% Принимает: N - количество экспериментов
% Возвращает: simple, precise - вычисленные значения для различных способов

function [simple, precise] = MKsolve(N)
    x_min = 0;
    x_max = 3;

    % Простой способ
    y_min = 0;
    y_max = 10;
    xs = rand(1, N) * (x_max - x_min) + x_min;
    ys = rand(1, N) * (y_max - y_min) + y_min;

    [n_pos, n_neg] = deal(0, 0);
    for i = 1:N
        func_yi = piecewise_func_lab1(xs(i));
        if func_yi < ys(i) && ys(i) < 0
            n_neg = n_neg + 1;
        elseif 0 <= ys(i) && ys(i) <= func_yi
            n_pos = n_pos + 1;
        end
    end
    points_portion = (n_pos - n_neg)/N;

    simple = points_portion * (x_max - x_min) * (y_max - y_min);

    % Повышенная точность
    xs = rand(1, N) * (x_max - x_min) + x_min;
    func_ys = arrayfun(@(x) piecewise_func_lab1(x), xs);

    precise = sum(func_ys)*(x_max - x_min) / N;
end
```

Рис. 4: Код функции вычисления интеграла

Скрипт-функция ***MKsolve.m*** - функция реализующая методы Монте-Карло для нахождения интеграла функции ***piecewise_func_lab1***

Основные переменные ***MKsolve.m***:

- *[x_min, x_max]* - область определения функции f_{res}, задаваемая задачей
- *[y_min, y_max]* - область значений функции f_{res}, необходимая для простого метода Монте-Карло
- *xs* - случайные точки, сгенерированные равномерным распределением на области определения (используются в обоих методах)

- *ys* - случайные значения для точек *xs* в простом методе
- *func_yi* - значение функции f_{res} в точке из *xs*
- *simple, precise* - найденные значения интеграла методами "простым"и "более точным"соответственно

5. Табличное представление результатов моделирования $F(N)$

```
>> lab1

results_S =

        0    7.5000    7.5000   16.8750   13.1250   15.4688   11.9531   13.4766   15.1172   14.0039   13.8721   13.6230   13.6047   13.3466


results_P =

   20.5820   10.5516   12.8245   14.0721   15.0206   12.4323   13.3442   13.1423   13.1189   13.3275   13.2052   13.2764   13.4516   13.5542

>>
```

Рис. 5: Вывод программы

Получим талицу значений для двух подходов:

$log_2(N)$	"Простой"	"Точный"
1	0.00	20.58
2	7.50	10.55
3	7.50	12.82
4	16.87	14.07
5	13.12	15.02
6	15.47	12.43
7	11.95	13.34
8	13.48	13.14
9	15.12	13.12
10	14.00	13.33
11	13.88	13.20
12	13.62	13.27
13	13.60	13.45
14	13.35	13.55

6. График по рассчитанной таблице

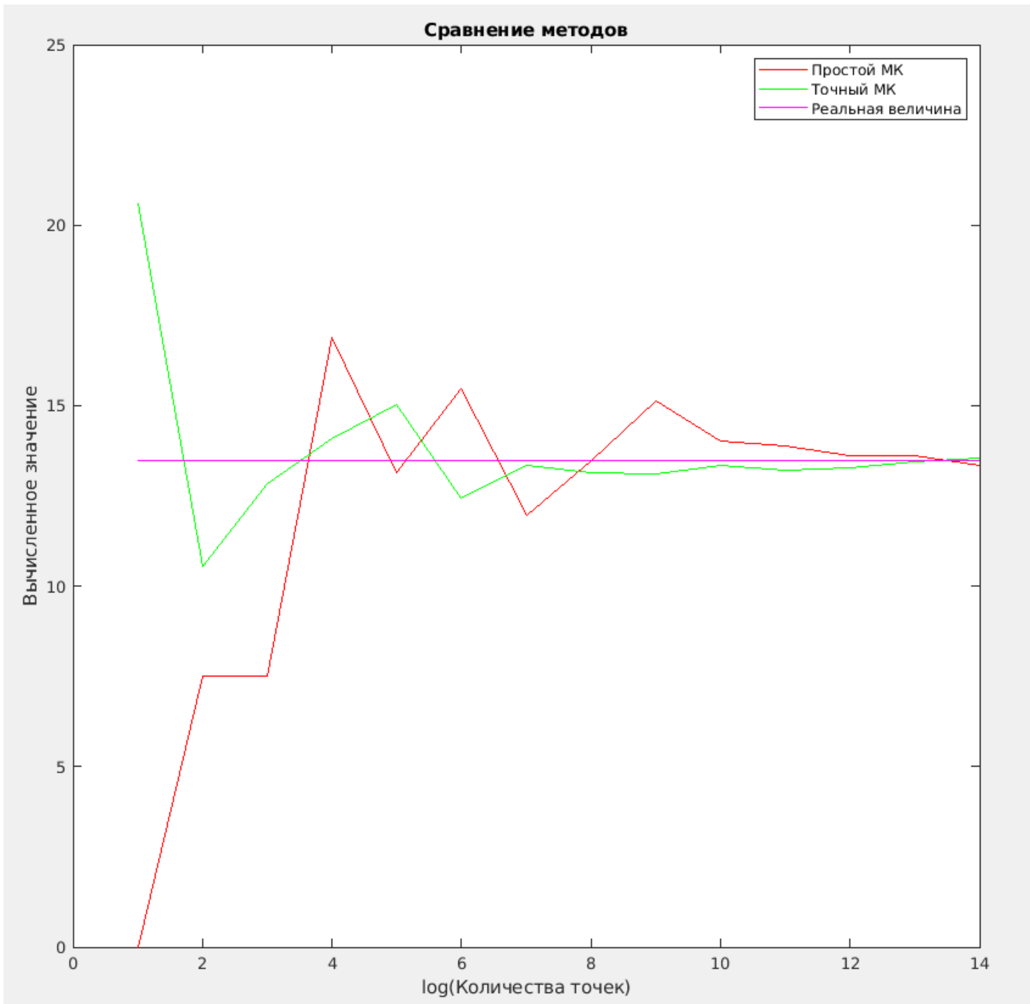

Рис. 6: Сравнение графиков методов

На графике можно увидеть более быструю сходимость значений точного метода Монте-Карло к реальной величине интеграла

7. Выводы

Целью данной лабораторной работы было изучение метода Монте-Карло и его применение. В процессе выполнения были реализованы 2 метода оценки интеграла функции:

- простой - основанный на площадях фигур
- с повышенной точностью - вычисление функции на случайных величинах $a_1 \dots a_N$

На основе вывода программы были получены оценки инеграла функции двумя методами для различного количества случайных точек; построен график сравнения оценок с исходным, вычисленым аналитически, значением интеграла функции.