1. Задание:  
   Создать файл, имитирующий эксперимент по изменению времени при падении шарика с разной высоты. Добавить шумы и отклонения, имитирующие реальный эксперимент. Получить минимум 10 значений для одной высоты.

Полученные результаты выгрузить в таблицу Excel.

Код выполнения задания:

function falling\_time(H)

t = sqrt(2 \* H / 9.81)';

infelicity = 0.8 + (1.1 - 0.8) \* rand(1, 10);

result = [H' round(infelicity.\*t, 3)];

writematrix(result, 'falling\_ball.xlsx');

end

Результат выполнения:

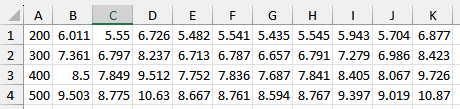


Рис. 6 Exel file со значениями высоты и времени падения мяча

1. Создать отдельный файл, загружающий из таблицы массив с результатами измерений. Выполнить:

2.1. Отсортировать данные по времени (sort).

2.2.Найти минимальное, максимальное, среднее время для одной высоты.

2.3. Построить график "Высота vs Время".

2.4. Попытаться подобрать коэффициент k для формулы времени падения t  методом наименьших квадратов в лоб (перебором k в разумном диапазоне, расчет суммы квадратов отклонений для каждого k, поиск минимума). Построить подобранную кривую на том же графике со значениями эксперимента.

Код выполнения задания:

function [max\_value, min\_value, mean\_value, k\_best] = falling\_ball\_grath()

mx = readmatrix('falling\_ball.xlsx');

H = mx(:, 1);

times = mx(:, 2:end);

times = sortrows(times')';

mx2 = [H times];

writematrix(mx2, 'falling\_ball\_s.xlsx');

max\_value = times(:, end);

min\_value = times(:, 1);

mean\_value = mean(times, 2);

mx\_stats = [min\_value mean(times, 2) max\_value];

mx2 = readmatrix('falling\_ball\_s.xlsx');

H = mx2(:,1);

times = mx2(:,2:end);

mean\_t = mean(times, 2);

k\_vals = 0.5:0.001:2;

SSE = zeros(size(k\_vals));

g = 9.81;

for i = 1:length(k\_vals)

k = k\_vals(i);

t\_pred = k .\* ( sqrt(2 .\* H ./ g)); % ваша модель

SSE(i) = sum( (times - t\_pred).^2, "all" );

end

[~, idx\_min] = min(SSE);

k\_best = k\_vals(idx\_min);

% Строим график

figure; hold on; grid on;

axis([H(1, 1) \*0.9 H(end, end)\*1.1 times(1, 1)\*0.9 times(end, end)\*1.1])

scatter(H, times, 'filled', "green", 'o');

scatter(H, mx\_stats, 'filled', "black", 'd');

H\_smooth = linspace(min(H), max(H), 200);

t\_fit = k\_best \* sqrt(2\*H\_smooth ./ g);

plot(H\_smooth, t\_fit, 'b-', 'LineWidth',1.5, 'DisplayName',sprintf('Модель t, k=%.3f', k\_best));

xlabel('Высота H, м');

ylabel('Время падения t, с');

title('Высота vs Время');

end

Результат выполнения:

max\_value =  
  
 6.8770  
 8.4230  
 9.7260  
 10.8740  
min\_value =  
  
 5.4350  
 6.6570  
 7.6870  
 8.5940  
  
mean\_value =  
  
 5.8814  
 7.2031  
 8.3175  
 9.2991  
  
k\_best =  
  
 0.9210

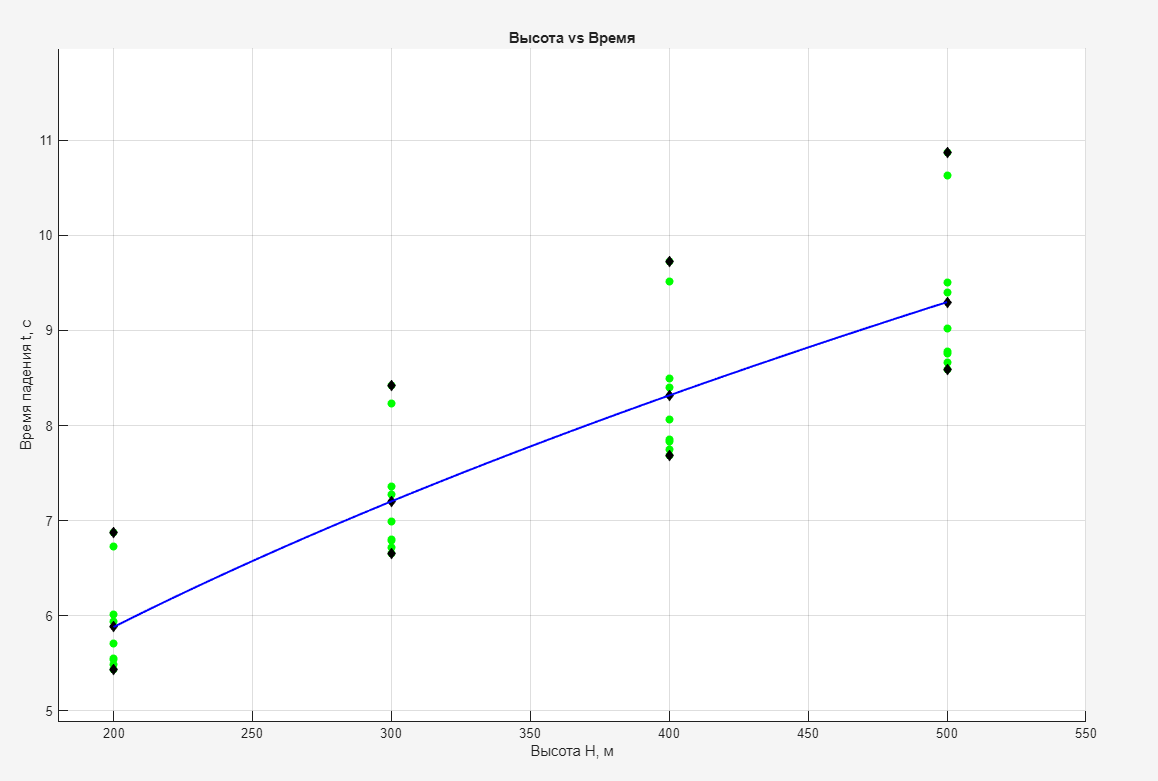


Рис 7. График "Высота vs Время".