Лабораторная работа №2

Фукнции

test_function_1

```
function y = test_function_1(t, u)
    y = u + t^2 + 1;
end
```

test_function_2

```
function f = test_function_2(t, u)
    om = [sin(t) cos(t) sin(t+pi/4)];
    Omega = [0 -om(3) om(2); om(3) 0 -om(1); -om(2) om(1) 0];
    f = Omega*u;
end
```

runge_kutta

```
function  [t, u] = runge_kutta(f, t_start, t_end, u0, N)
    % создаём матрицу промежуточных результатов
    w = zeros(length(u0), 1);

    % Инициализируем u
    u = zeros(length(u0), N);
    u(:, 1) = u0;

    % Вычисляем шаг по оси t и создаём эту ось
    h = (t_end - t_start) / N;
    t = t_start:h:t_end;

    for j = 1:N
            w1 = h * feval(f, t(j),  u(:,j));
            w2 = h * feval(f, t(j) + h / 2, u(:,j) + w1 / 2);
            w3 = h * feval(f, t(j) + h / 2, u(:,j) + w2 / 2);
            w4 = h * feval(f, t(j) + h, u(:,j) + w3);
            u(:, j+1) = u(:, j) + (w1 + 2 * w2 + 2 * w3 + w4) / 6;
    end

    t = t';
    u = u';
end
```

Задание 2

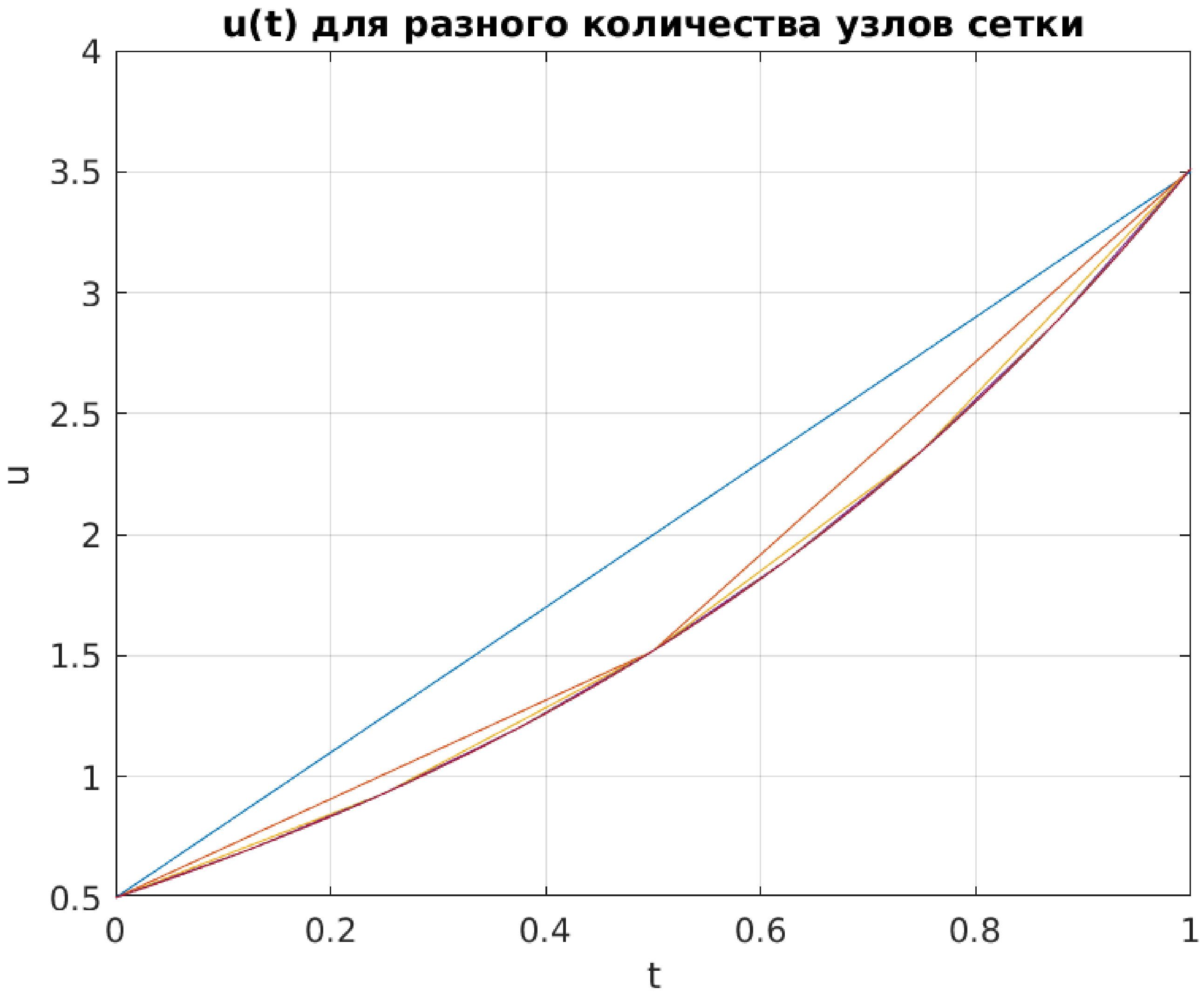
task_2.m

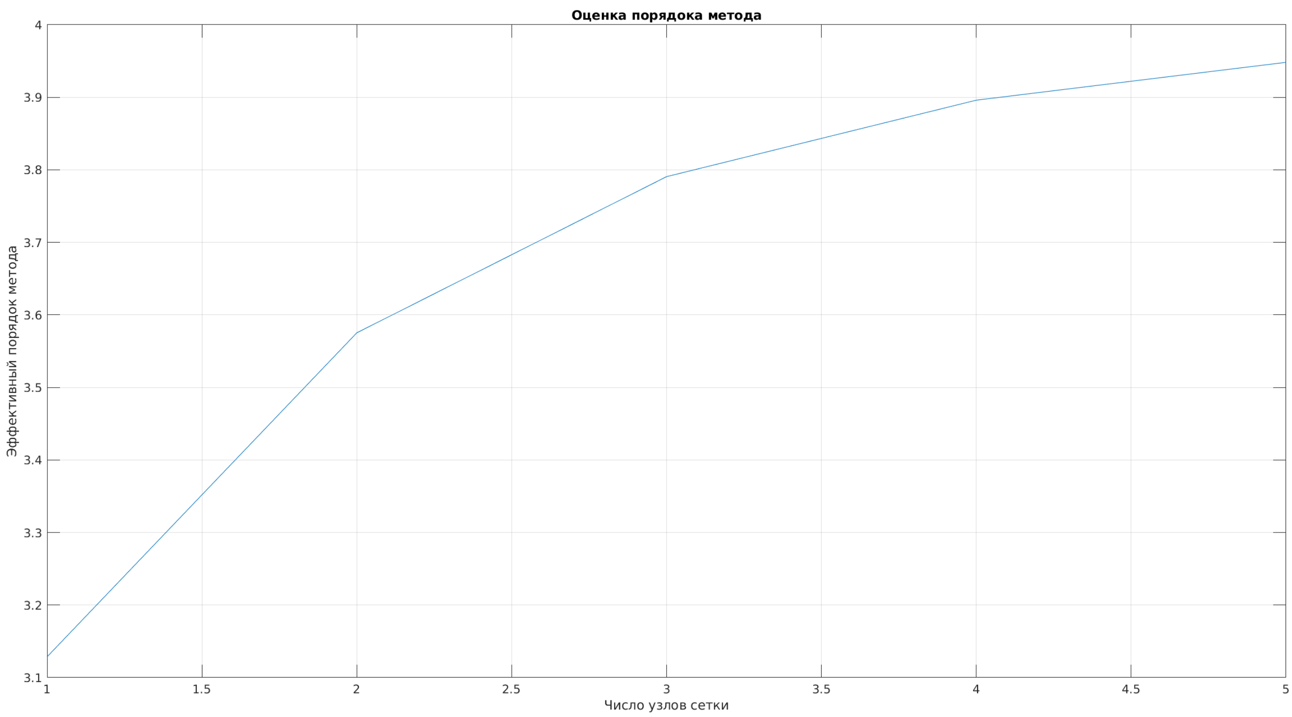
```
clear;
u0 = 0.5;
t_start = 0;
t_end = 1;

last_node_u = zeros(7,1);
figure(1)
for i = 1:7
    [t, u] = runge_kutta('test_function_1', t_start, t_end, u0, 2^(i - 1));
    last_node_u(i) = u(length(u));
    plot(t, u)
    title('u(t) для разного количества узлов сетки')
    xlabel('t')
    ylabel('u')
    grid on
    hold on
end

for i=1:5
    order_estimate(i) = log((last_node_u(i+2) - last_node_u(i+1)) / (last_node_u(i+1) -
last_node_u(i))) / log(0.5);
end

figure(2)
plot([1 2 3 4 5], order_estimate)
grid on
title('Оценка порядока метода')
ylabel('Эффективный порядок метода')
xlabel('Число узлов сетки')
```

Построим график эффективного порядка метода для функции:

```
y = u+t^2+1
```

С начальным условием: `u0 = 0.5`





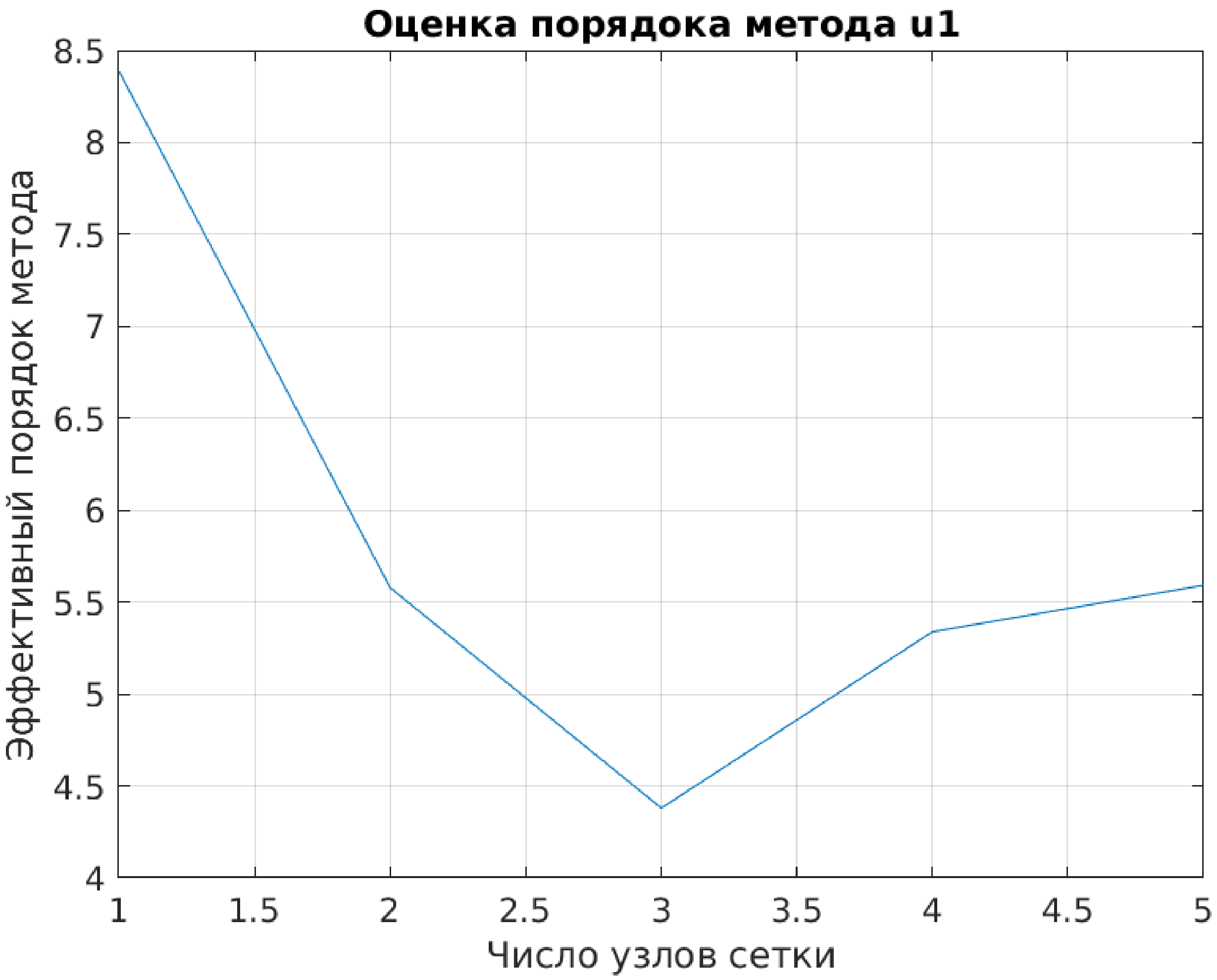
Из графика эффективного порядок точности метода, что мы добились аппроксимации четвертого порядка, что соответствует теоретическим данным.

Повторим процедуру для функции:

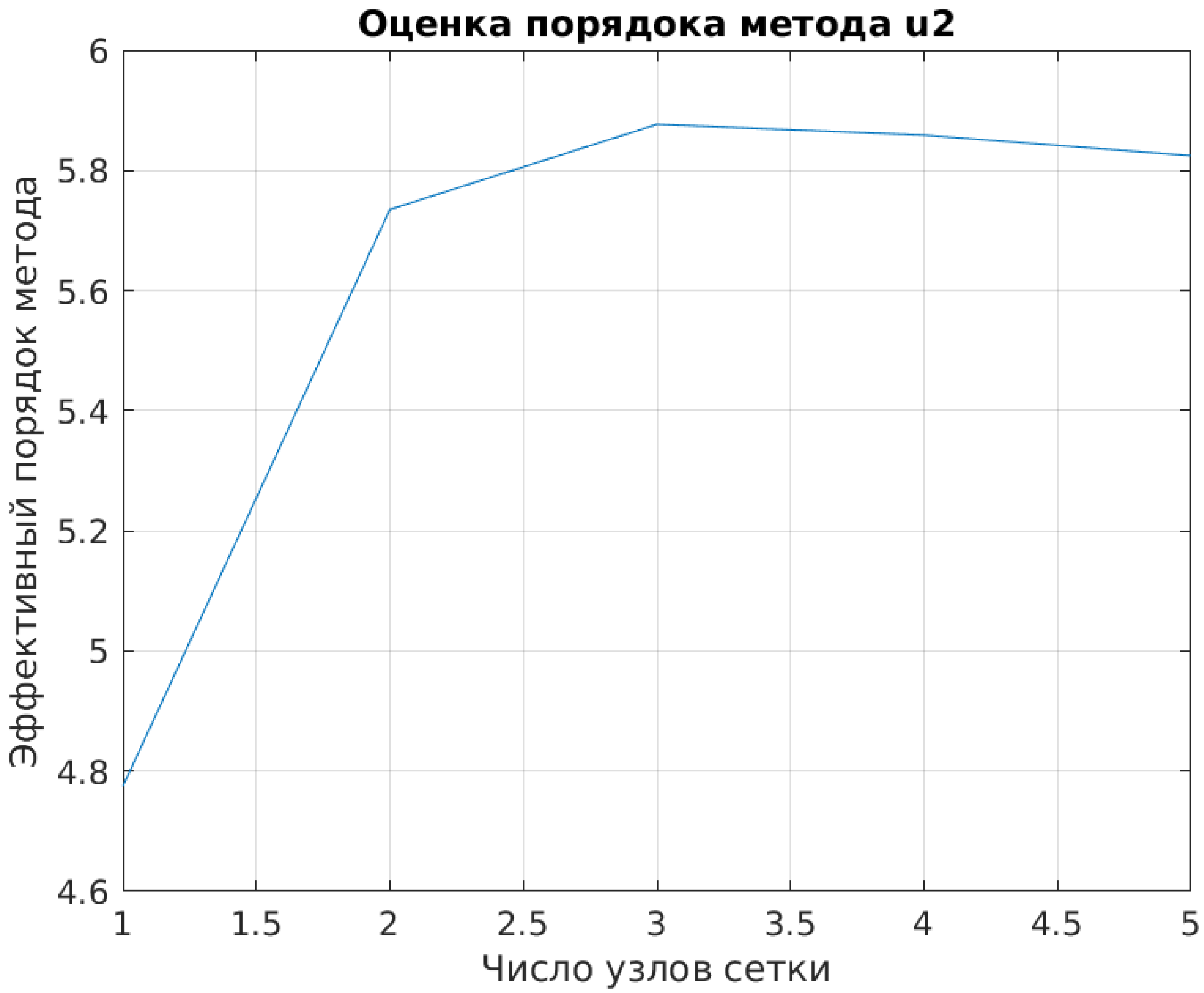
```
function f = ff(t, u)
om = [sin(t) cos(t) sin(t+pi/4)];
Omega = [0 -om(3) om(2); om(3) 0 -om(1); -om(2) om(1) 0];
f = Omega*u;
```

Начальное условие: `u0 =[1; -0.5; 0.6]`

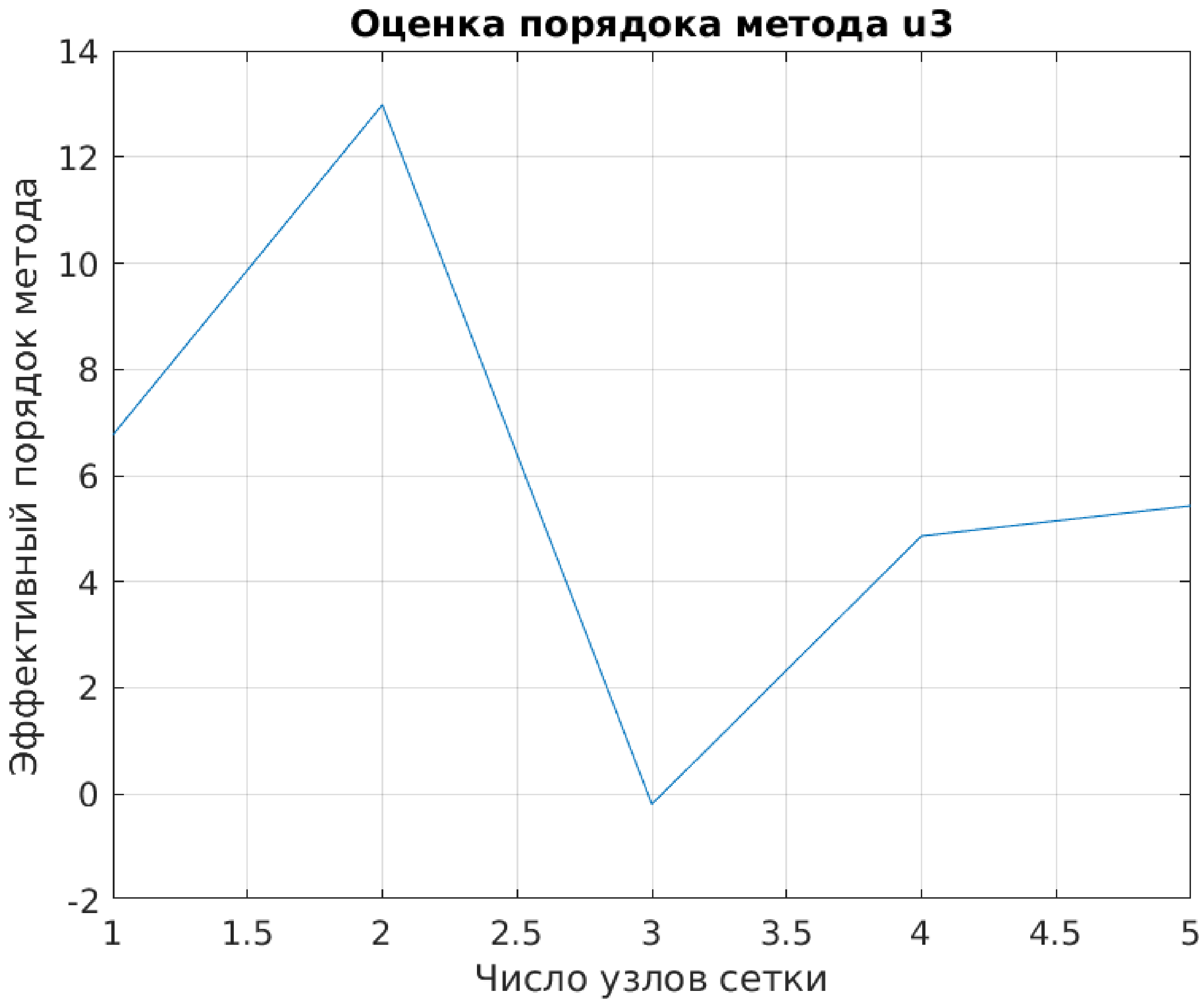
```
clear;
u0 = [1; -0.5; 0.6];
t_start = 0;
t_end = 1;
max_node_count = 7;

u_dimension = length(u0);

last_node_u = zeros(max_node_count, u_dimension);
for i = 1:max_node_count
    [t, u] = runge_kutta('test_function_2', t_start, t_end, u0, 2^(i - 1));
    last_node_u(i,:) = u(length(t), :);
end



order_estimate = zeros(max_node_count - 2, u_dimension);
for i=1:max_node_count - 2
    for j=1:u_dimension
        order_estimate(i, j) = log2((last_node_u(i+2, j) - last_node_u(i+1, j)) / (last_node_u(i+1, j) - last_node_u(i, j))) / log(0.5);
    end
end

plot(1:5, order_estimate(:, 1))
grid on
title('Оценка порядока метода u1')
ylabel('Эффективный порядок метода')
xlabel('Число узлов сетки')
```



Оценка порядока метода u1
Эффективный порядок метода
Число узлов сетки
8.5
8
7.5
7
6.5
6
5.5
5
4.5
4
1
1.5
2
2.5
3
3.5
4
4.5
5



Оценка порядока метода u2
Эффективный порядок метода
Число узлов сетки
6
5.8
5.6
5.4
5.2
5
4.8
4.6
1
1.5
2
2.5
3
3.5
4
4.5
5



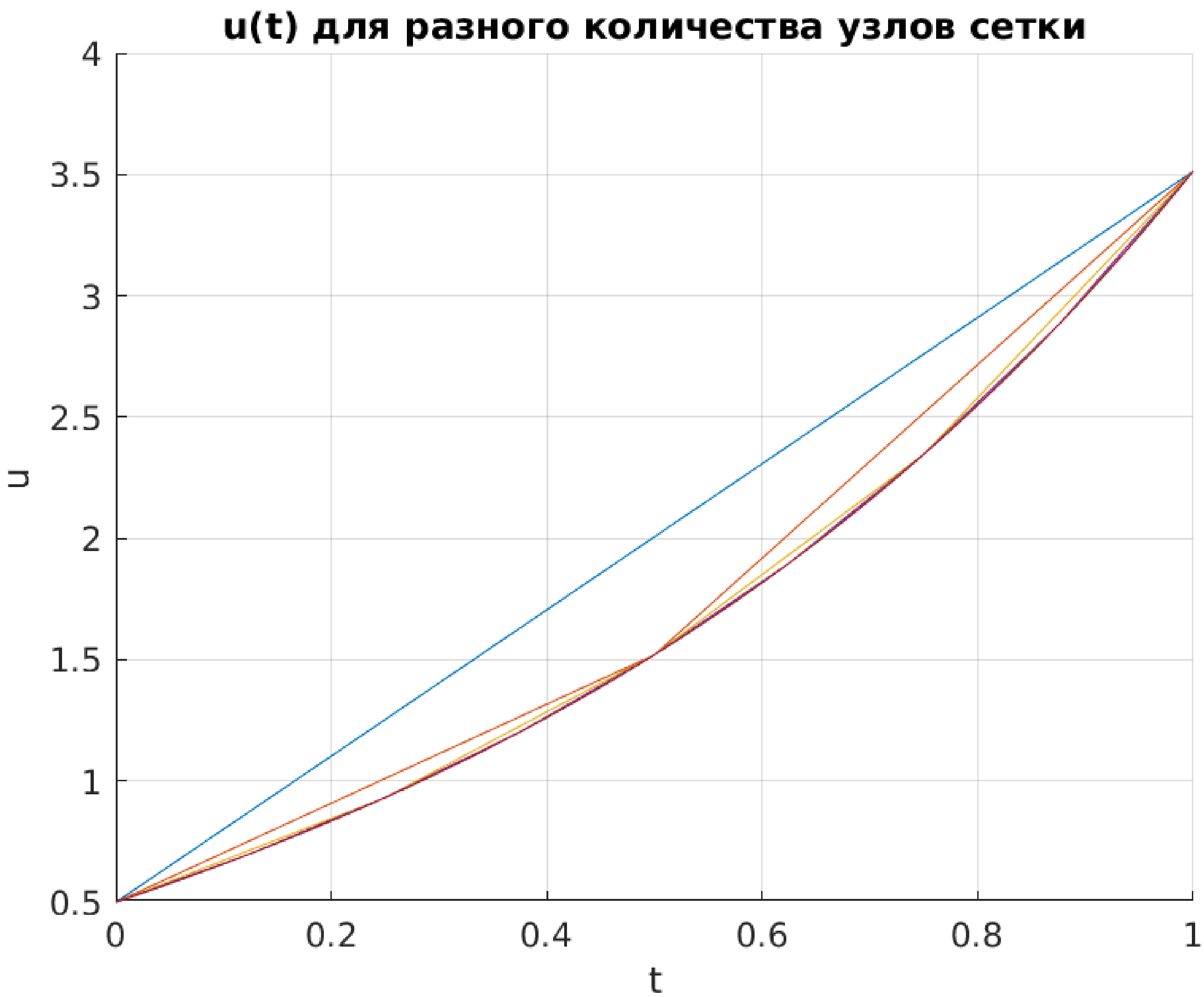
Видим, что порядок аппроксимации стремится к 5.8.

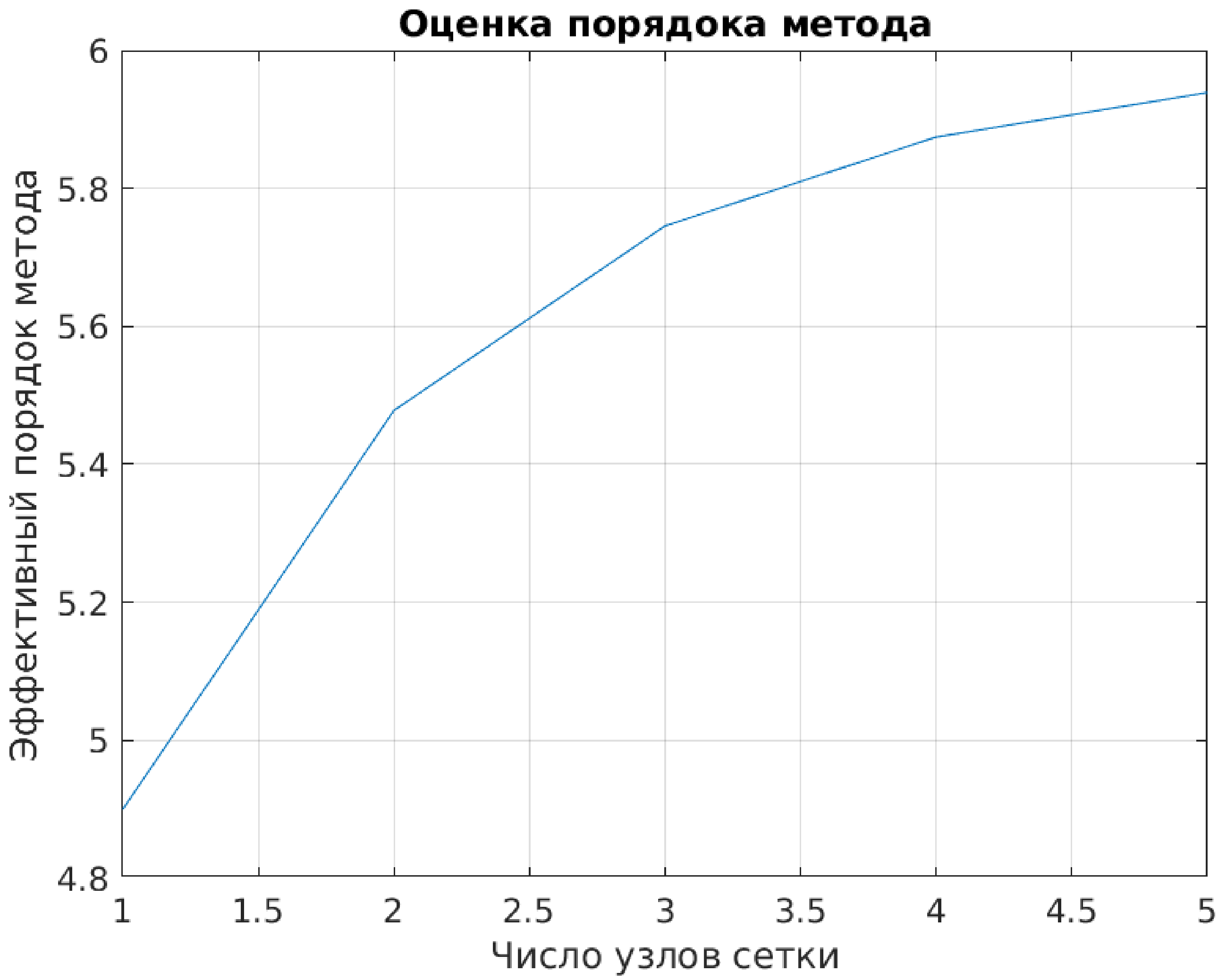
7-стадийная схема Хаммуда 6 порядка

runge_kutta_unviersal

```
function  [t, u] = runge_kutta_unviersal(f, tstart, tstop, u0, N)
butcher = [0      0       0 0 0 0 0;...
          4/7     0       0 0 0 0 0;...
          115/112 -5/16      0 0 0 0 0;...
          589/630  5/18 -16/45 0 0 0 0;...
          229/1200-29/6000*5^0.5 119/240-187/1200*5^0.5 -14/75+34/375*5^0.5 -3/100*5^0.5 0 0 0;...
          71/2400-587/12000*5^0.5 187/480-391/2400*5^0.5 -38/75+26/375*5^0.5 27/80-3/400*5^0.5 (1+5^0.5)/4 0 0;...
          -49/480+43/160*5^0.5 -425/96+51/32*5^0.5 52/15-4/5*5^0.5 -27/16+3/16*5^0.5 5/4-3/4*5^0.5 5/2-0.5*5^0.5 0];

    a = [0 4/7 5/7 6/7 (5-5^0.5)/10 (5+5^0.5)/10 1];
    b = [1/12 0 0 0 5/12 5/12 1/12];

    butcher_matrix_size = length(butcher)
    w = zeros(length(u0), butcher_matrix_size);
```

```
    u = zeros(length(u0), N);
    u(:,1) = u0;

    h = (tstop - tstart) / N;
    t = tstart:h:tstop;

    for j=1:N

        for i=1:butcher_matrix_size
            w(:,i) = h * feval(f, t(j) + h * a(i), u(:,j) + sum((w .* butcher(i,:))')');
        end

        u(:, j+1) = u(:, j) + sum((w .* b)')';
    end

    t = t';
    u = u';
    hold on
end
```





Видим, что схема действительно имеет шестой порядок аппроксимации.