

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана»
(национальный исследовательский университет)

Кафедра «Прикладная математика»



Домашняя работа

по дисциплине «Методы численного решения обыкновенных
дифференциальных уравнений»

Вариант 5

Выполнила студентка группы ФН2-41М

Матюхина Д.И.

Преподаватель

Родин А.С.

1. Постановка задачи

Решить систему ОДУ:

$$\begin{aligned} y_1' &= 2t y_1 \ln(\max(y_2, 10^{-3})), \\ y_2' &= -2t y_2 \ln(\max(y_1, 10^{-3})), \end{aligned}$$

с известным аналитическим решением

$$y_1(t) = \exp(\sin(t^2)), \quad y_2(t) = \exp(\cos(t^2)).$$

Интервал интегрирования $[t_1, t_2]$, где $t_1 = N_{var} * 0.1$, $N_{var} = 5$, $t_2 = t_1 + 4$. В качестве начальных условий берутся значения аналитических решений $y_1(t_1), y_2(t_1)$.

Необходимо решить поставленную задачу методом Рунге-Кутты 4 порядка с 4 стадиями и методом Дормана-Принса 5(4). Нужно использовать процедуру автоматического управления длиной шага (для РК 4 порядка применить экстраполяцию Ричардсона, для Дормана-Принса 5(4) – оценку погрешности, полученную для двух значений численного решения).

2. Основные теоретические положения

Пусть s – целое положительное число (число стадий) и $a_{21}, a_{31}, a_{32}, \dots, a_{s1}, a_{s2}, a_{s,s-1}, \dots, b_1, \dots, b_s, c_2, \dots, c_s$ – вещественные коэффициенты. Тогда метод

$$\begin{aligned} k_1 &= f(x_0, y_0), \\ k_2 &= f(x_0 + c_2 h, y_0 + h a_{21} k_1), \\ k_3 &= f(x_0 + c_3 h, y_0 + h(a_{31} k_1 + a_{32} k_2)), \\ &\dots \\ k_s &= f(x_0 + c_s h, y_0 + h(a_{s1} k_1 + \dots + a_{s,s-1} k_{s-1})), \\ y_1 &= y_0 + h(b_1 k_1 + \dots + b_s k_s) \end{aligned}$$

называется s -стадийным явным методом Рунге-Кутты для задачи

$$y' = f(x, y), \quad y(x_0) = y_0.$$

Для поиска численного решения исходной системы уравнений методом Рунге-Кутты 4 порядка ($p = 4$) с 4 стадиями ($s = 4$) выбираем правило 3/8, со следующими значениями коэффициентов из таблицы Бутчера:

$$\begin{aligned} a_{21} &= \frac{1}{3}, \quad a_{31} = -\frac{1}{3}, \quad a_{32} = 1, \quad a_{41} = 1, \quad a_{42} = -1, \quad a_{43} = 1, \\ c_2 &= \frac{1}{3}, \quad c_3 = \frac{2}{3}, \quad c_4 = 1, \\ b_1 &= \frac{1}{8}, \quad b_2 = \frac{3}{8}, \quad b_3 = \frac{3}{8}, \quad b_4 = \frac{1}{8}. \end{aligned}$$

Автоматическое управление длиной шага с применением экстраполяции Ричардсона осуществляется следующим образом. Находим численные решения y_1, y_2 методом Рунге-Кутты для двух шагов τ , а затем одно численное решение ω для шага величиной 2τ . Локальная погрешность вычисляется по формуле

$$err = \frac{1}{2^p - 1} \max_{i=1, \dots, n} \frac{|y_{2,i} - \omega_i|}{d_i},$$

где p – порядок метода Рунге-Кутты, d_i – масштабирующий множитель. Затем погрешность err сравнивается с заданной допустимой величиной tol . Если $err \leq tol$, вычисления принимаются (решение для текущего шага принимается равным y_2), дальнейший поиск решения из y_2 или $\hat{y}_2 = y_2 + (y_2 - \omega)/(2^p - 1)$ проводится с шагом

$$\tau_{new} = \tau \cdot \min(facmax, \max(facmin, fac \cdot (tol/err)^{1/(p+1)})),$$

где fac – гарантийный фактор, $facmax, facmin$ – максимальный и минимальный коэффициенты увеличения шага. Если же $err > tol$, найденные численные решения отбрасываются, вычисления повторяются с новой длиной шага τ_{new} .

Для численного решения системы обыкновенных дифференциальных уравнений методом Дормана-Принса используем следующую таблицу Бутчера.

0							
$\frac{1}{5}$	$\frac{1}{5}$						
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$					
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$				
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$			
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$		
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	
y_1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0
\hat{y}_1	$\frac{5179}{57600}$	0	$\frac{7571}{16695}$	$\frac{393}{640}$	$-\frac{92097}{339200}$	$\frac{187}{2100}$	$\frac{1}{40}$

Рис. 1. Метод Дормана-Принса 5(4)

Порядок решения y_1 : $p_1 = 5$, порядок оценщика погрешности \hat{y}_1 : $p_2 = 4$. Для процедуры автоматического управления длиной шага вычисляем погрешность

$$err = \max_{i=1, \dots, n} \frac{|y_{1,i} - \hat{y}_{1,i}|}{d_i}. \quad (1)$$

3. Задание 1

Исследуем влияние значения начального шага τ_0 , гарантийного фактора fac , коэффициентов $fastax$, $fastin$ при заданной допустимой погрешности $tol = 10^{-4}$ на следующие параметры: общее количество шагов, количество отбракованных шагов, время счета, итоговая глобальная погрешность.

Таблица 1. Результаты расчетов для различных τ_0 при $fac = 0.8$, $fastax = 1.5$, $fastin = 0.4$

τ_0	Шаги	Отбракованные шаги	Время счета, с	Итоговая погрешность
0.1	52	7	0.0069	0.0017
0.01	57	7	0.0073	0.0017
0.001	61	6	0.0080	0.0027
0.0001	67	6	0.0097	0.0017

Таблица 2. Результаты расчетов при различных fac при $\tau_0 = 0.01$, $fastax = 1.5$, $fastin = 0.4$

fac	Шаги	Отбракованные шаги	Время счета, с	Итоговая погрешность
0.8	57	7	0.0026	0.0017
0.9	56	11	0.0076	0.0030
$(0.25)^{(1/(p+1))}$	56	6	0.0078	0.0016
$(0.38)^{(1/(p+1))}$	55	8	0.0077	0.0029

Таблица 3. Результаты расчетов при различных $fastax$ при $\tau_0 = 0.01$, $fac = 0.8$, $fastin = 0.4$

$fastax$	Шаги	Отбракованные шаги	Время счета, с	Итоговая погрешность
1.5	57	7	0.0026	0.0017
2.5	54	7	0.0080	0.0017
3.5	52	6	0.0070	0.0017
4.5	53	7	0.0069	0.0017
5	49	4	0.0065	0.0019

Таблица 4. Результаты расчетов при различных $fastin$ при $\tau_0 = 0.01$, $fac = 0.8$, $fastax = 5$

$fastin$	Шаги	Отбракованные шаги	Время счета, с	Итоговая погрешность
0.2	49	4	0.0044	0.0019
0.4	49	4	0.0071	0.0019
0.6	49	4	0.0066	0.0019
0.66	49	4	0.0081	0.0019

Из таблицы 1 видно, что уменьшение начального шага τ_0 приводит к увеличению общего количества шагов. Из таблицы 2 видно, что оптимальным значением гарантийного фактора является $fac = (0.25)^{(1/(p+1))}$ – уменьшается количество отбракованных шагов и глобальная погрешность. Из таблицы 3 следует, что с увеличением значения коэффициента $fastax$ уменьшается общее количество шагов. Таблица 4

показывает, что выбор значения *fastin* оказывает незначительное влияние только на время счета.

4. Задание 2

Проведем расчеты двумя методами для фиксированного набора параметров.

Таблица 5. Результаты расчетов методом Рунге-Кутты при $\tau_0 = 0.01$, $fac = (0.25)^{(1/(p+1))}$, $fac_{max} = 5$, $fac_{min} = 0.4$

tol	Шаги	Отбракованные шаги	Время счета, с	Итоговая погрешность
10^{-3}	31	4	$9.462 \cdot 10^{-3}$	$3.2 \cdot 10^{-3}$
10^{-4}	49	7	$1.073 \cdot 10^{-2}$	$4.199 \cdot 10^{-4}$
10^{-6}	106	4	$1.402 \cdot 10^{-2}$	$7.183 \cdot 10^{-6}$

Таблица 6. Результаты расчетов методом Дормана-Принса при $\tau_0 = 0.01$, $fac = (0.25)^{(1/(p+1))}$, $fac_{max} = 5$, $fac_{min} = 0.4$

tol	Шаги	Отбракованные шаги	Время счета, с	Итоговая погрешность
10^{-3}	33	3	$6.857 \cdot 10^{-3}$	$2.93 \cdot 10^{-2}$
10^{-4}	53	7	$7.351 \cdot 10^{-3}$	$1.7 \cdot 10^{-3}$
10^{-6}	108	0	$9.438 \cdot 10^{-3}$	$9.750 \cdot 10^{-6}$

Из таблиц 5 и 6 видно, что итоговые погрешности для решений, полученных двумя методами при $tol = 10^{-6}$, имеют одинаковый порядок. Однако, время работы метода Дормана-Принса на порядок меньше времени работы метода Рунге-Кутты. Далее приведены графики зависимости погрешностей и длины шага от времени, а также численного решения системы для различных значений допустимой погрешности tol .

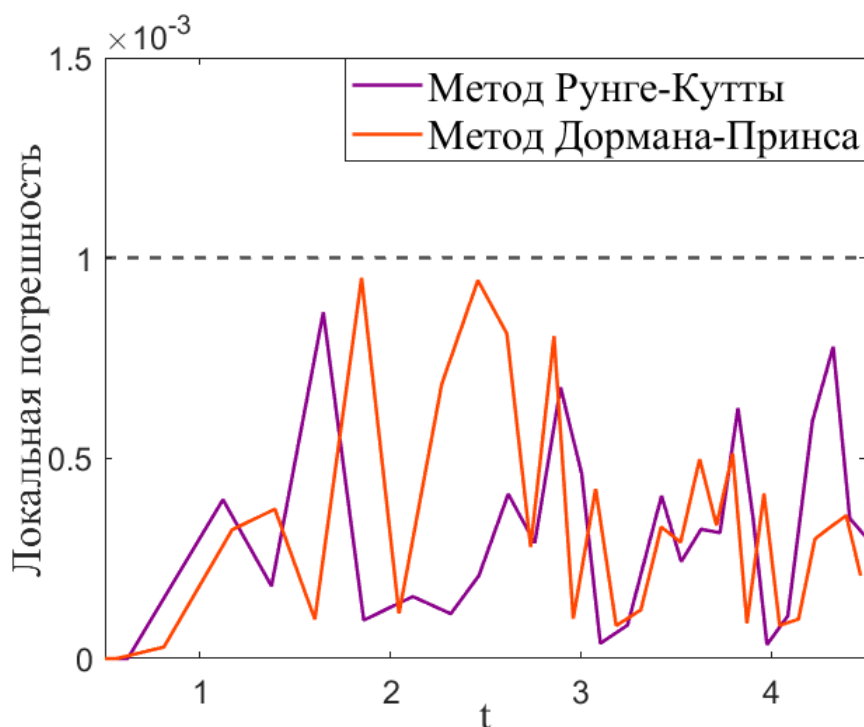


Рис. 2. Зависимость локальной погрешности от времени при $tol = 10^{-3}$

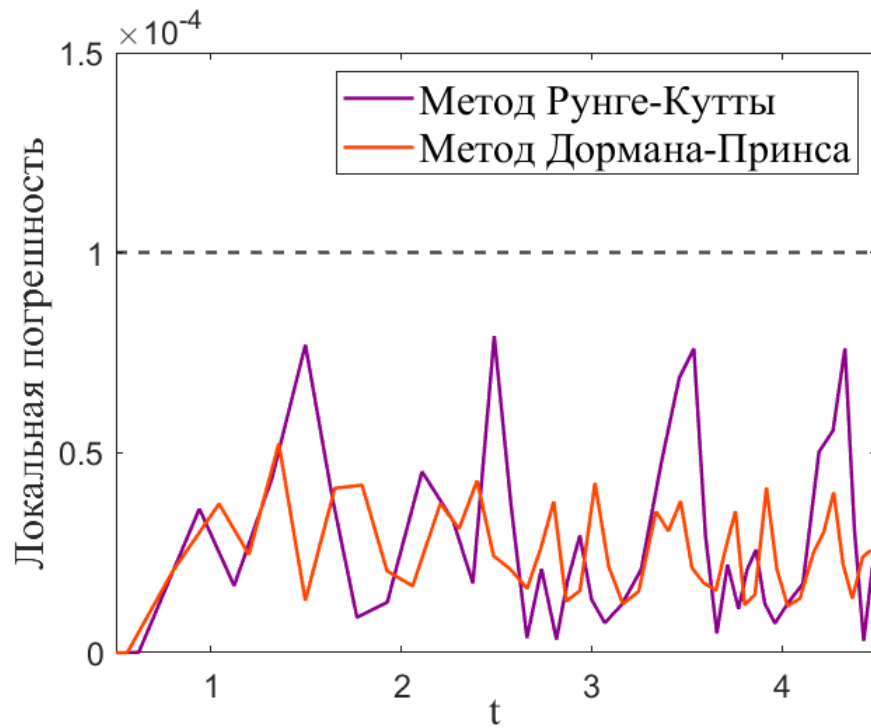


Рис. 3. Зависимость локальной погрешности от времени при $tol = 10^{-4}$

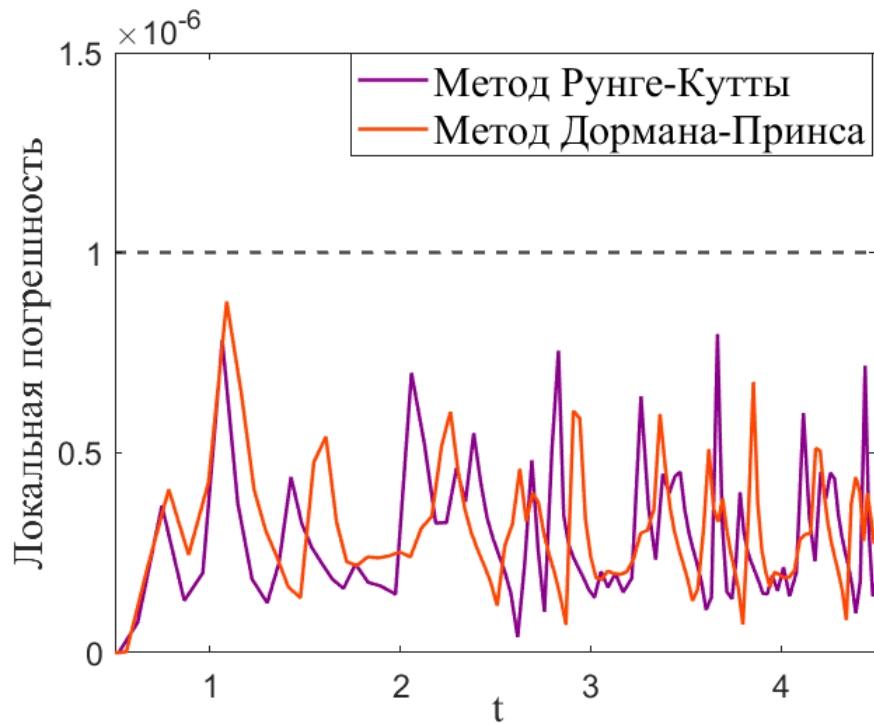


Рис. 4. Зависимость локальной погрешности от времени при $tol = 10^{-6}$

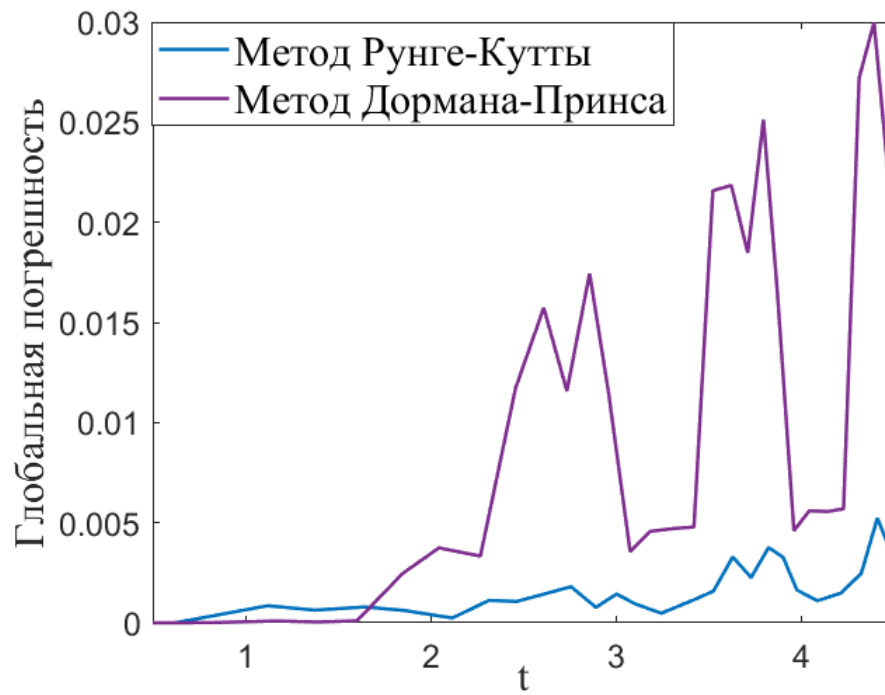


Рис. 5. Зависимость глобальной погрешности от времени при $tol = 10^{-3}$

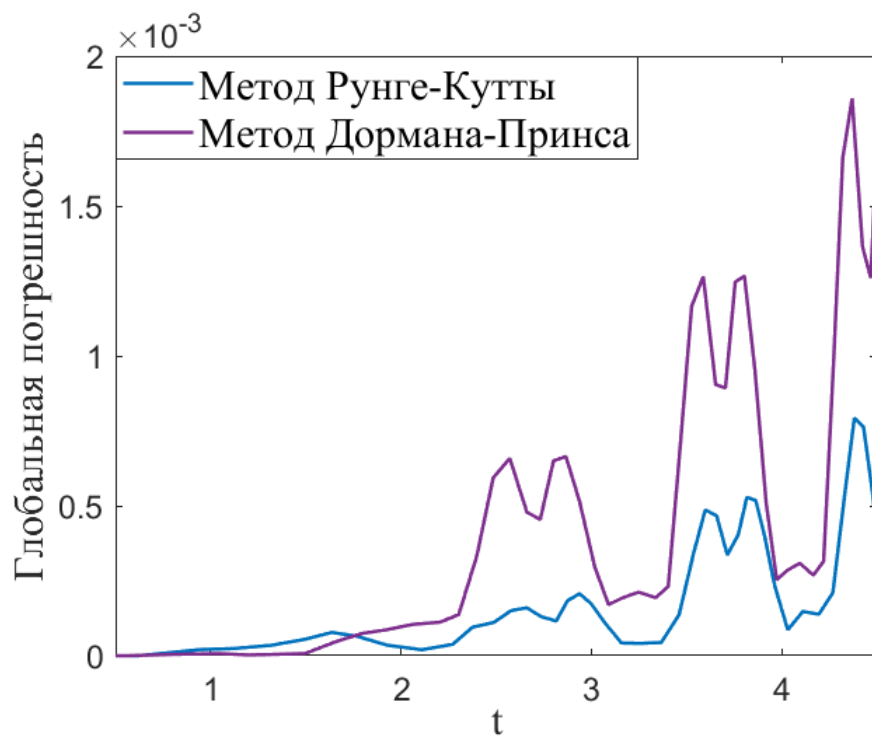


Рис. 6. Зависимость глобальной погрешности от времени при $tol = 10^{-4}$

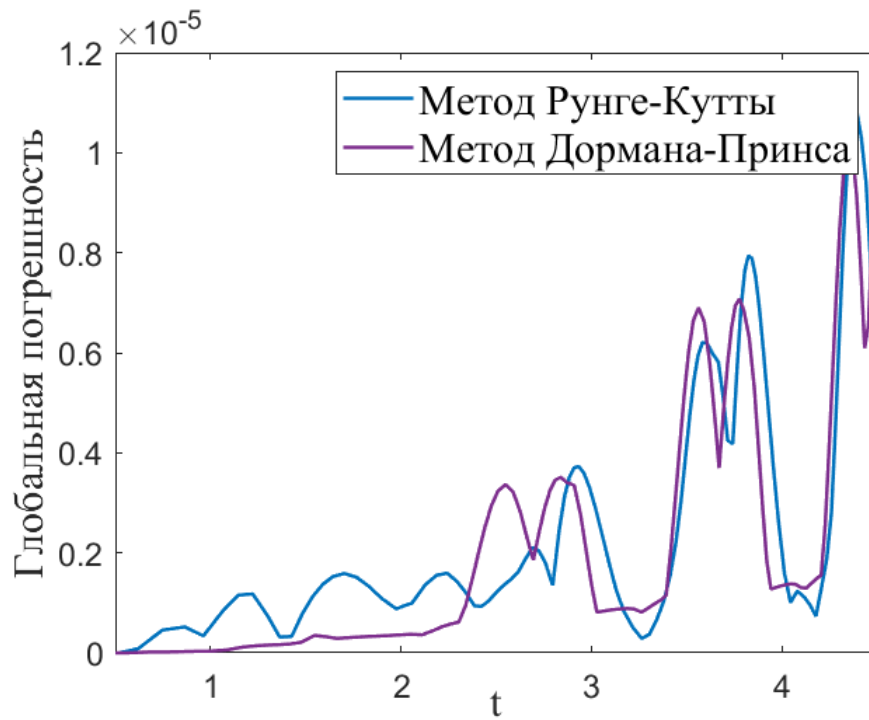


Рис. 7. Зависимость глобальной погрешности от времени при $tol = 10^{-6}$

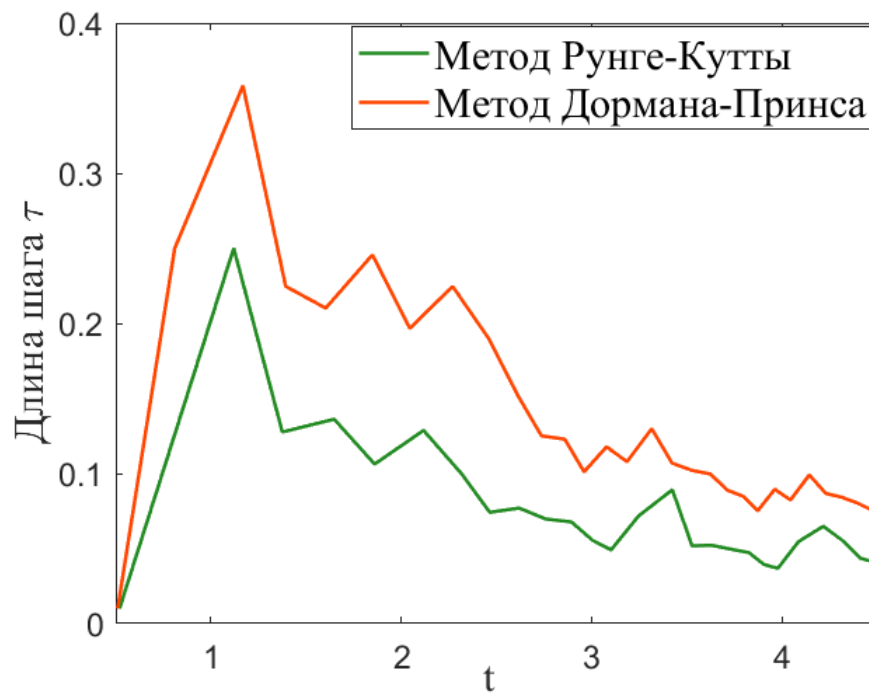


Рис. 8. Зависимость длины шага от времени при $tol = 10^{-3}$

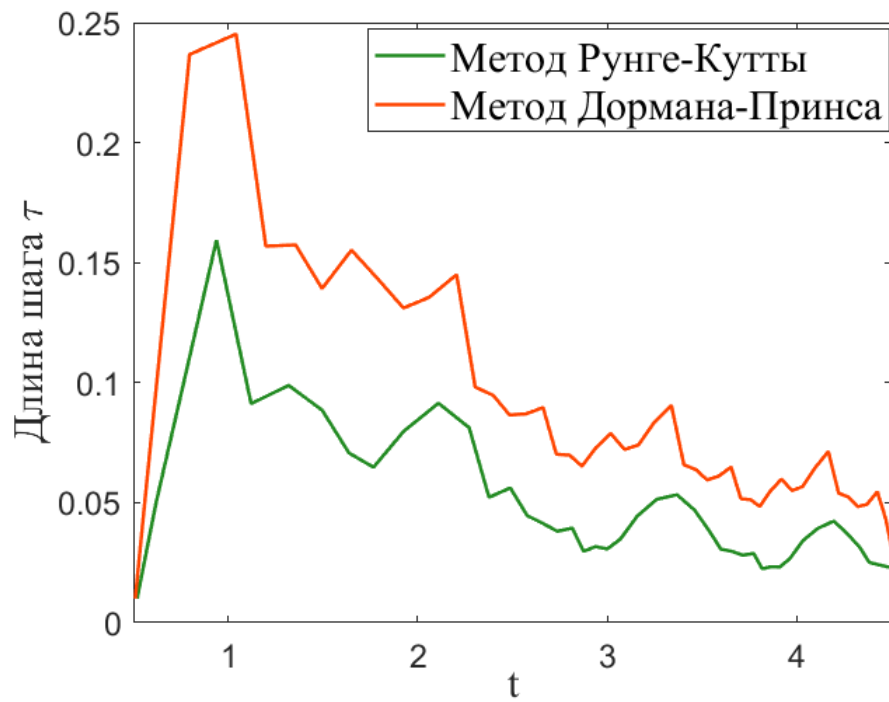


Рис. 9. Зависимость длины шага от времени при $tol = 10^{-4}$

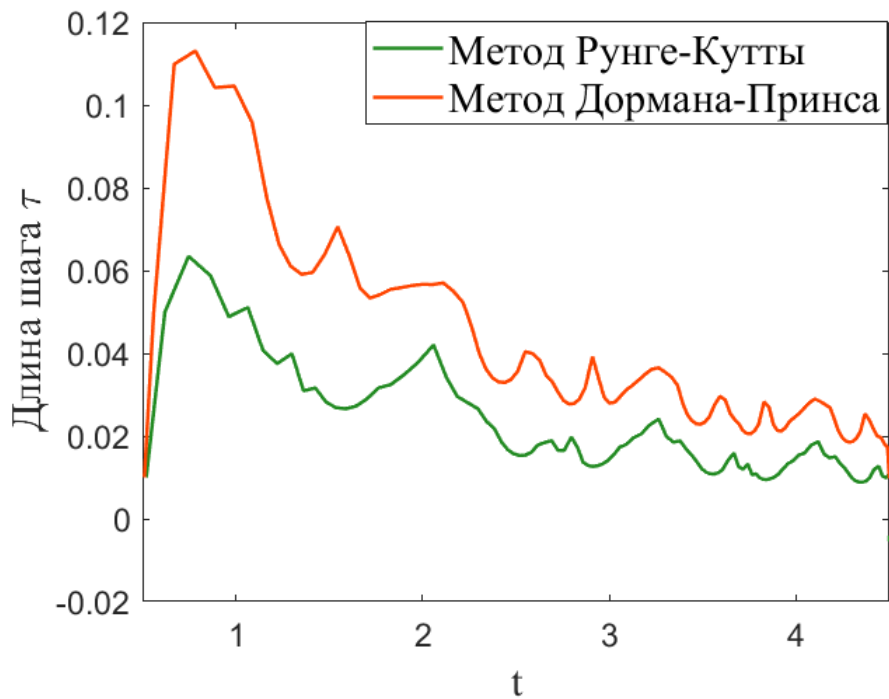


Рис. 10. Зависимость длины шага от времени при $tol = 10^{-6}$

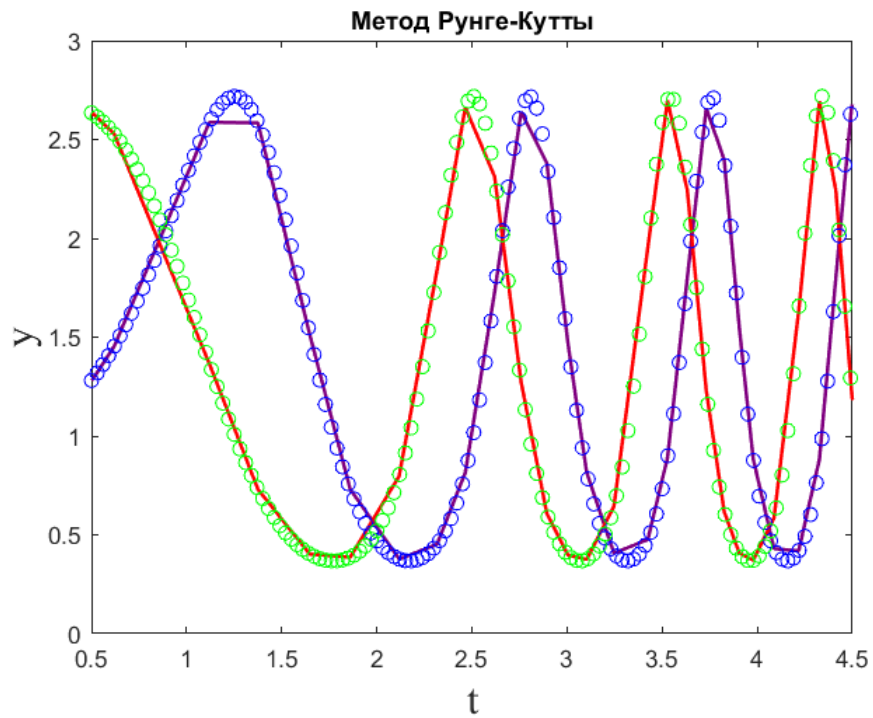


Рис. 11. Численное решение (сплошные линии), полученное методом Рунге-Кутты при $tol = 10^{-3}$. Круги – аналитическое решение системы

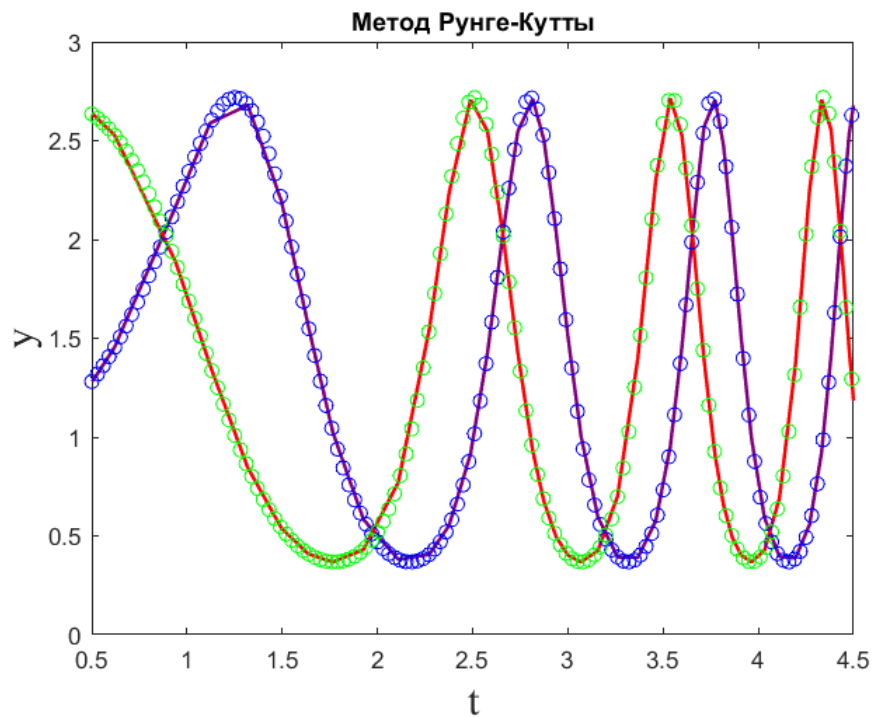


Рис. 12. Численное решение (сплошные линии), полученное методом Рунге-Кутты при $tol = 10^{-4}$. Круги – аналитическое решение системы

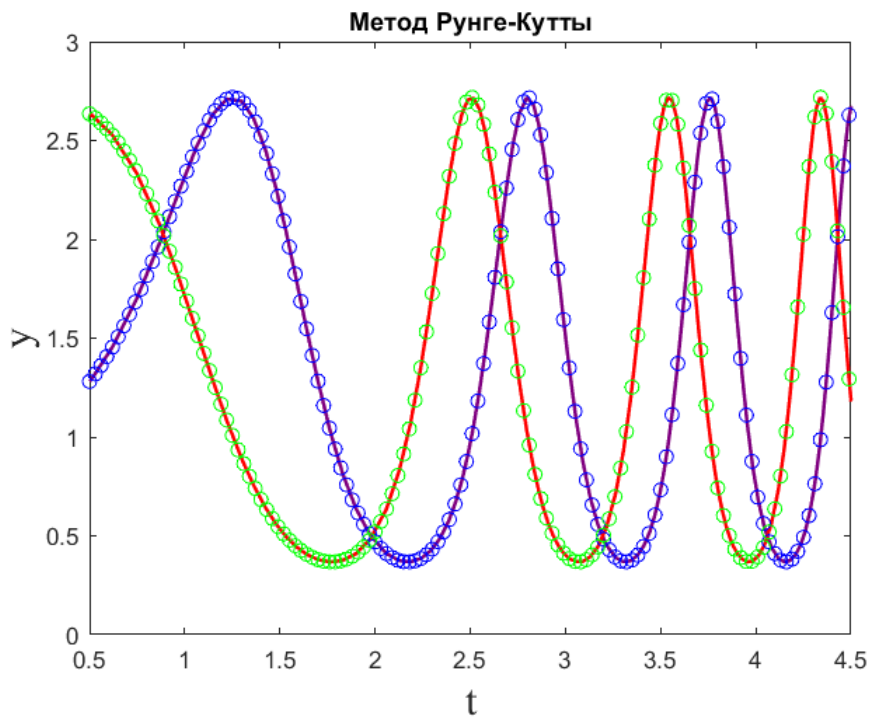


Рис. 13. Численное решение (сплошные линии), полученное методом Рунге-Кутты при $tol = 10^{-6}$. Круги – аналитическое решение системы

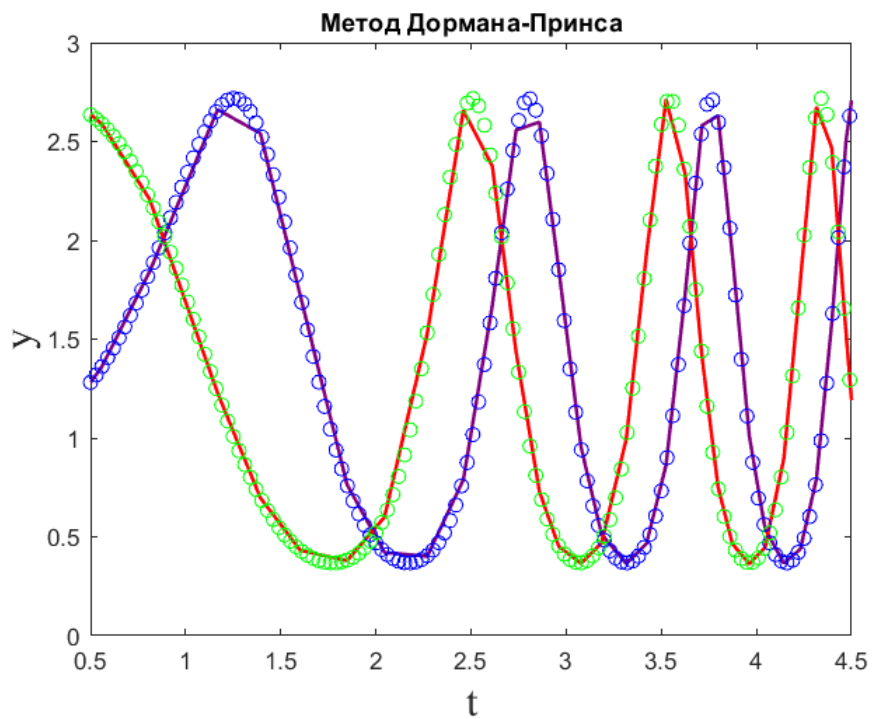


Рис. 14. Численное решение (сплошные линии), полученное методом Дормана-Принса при $tol = 10^{-3}$. Круги – аналитическое решение системы

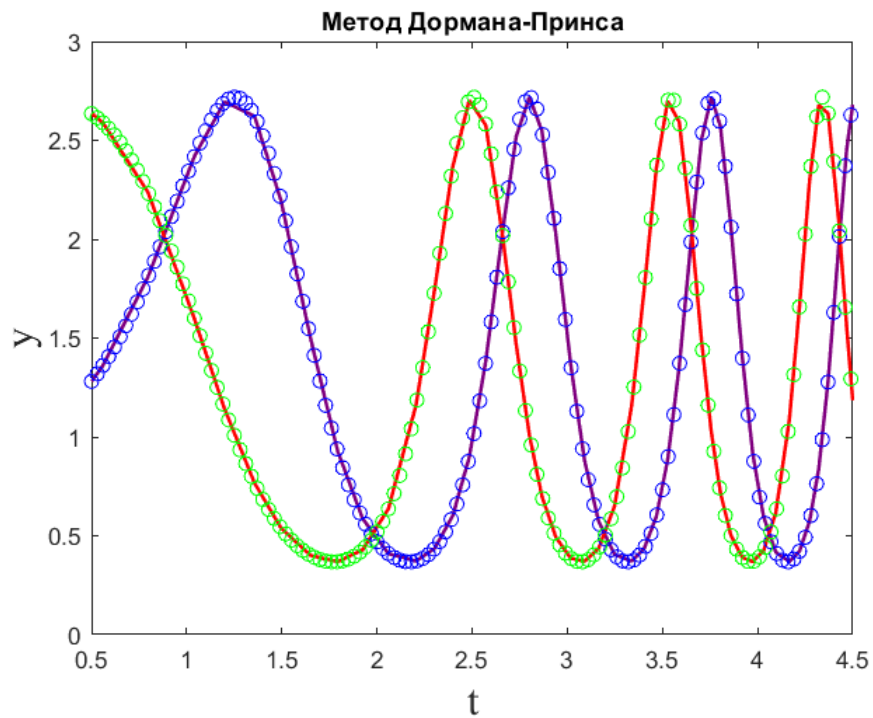


Рис. 15. Численное решение (сплошные линии), полученное методом Дормана-Принса при $tol = 10^{-4}$. Круги – аналитическое решение системы

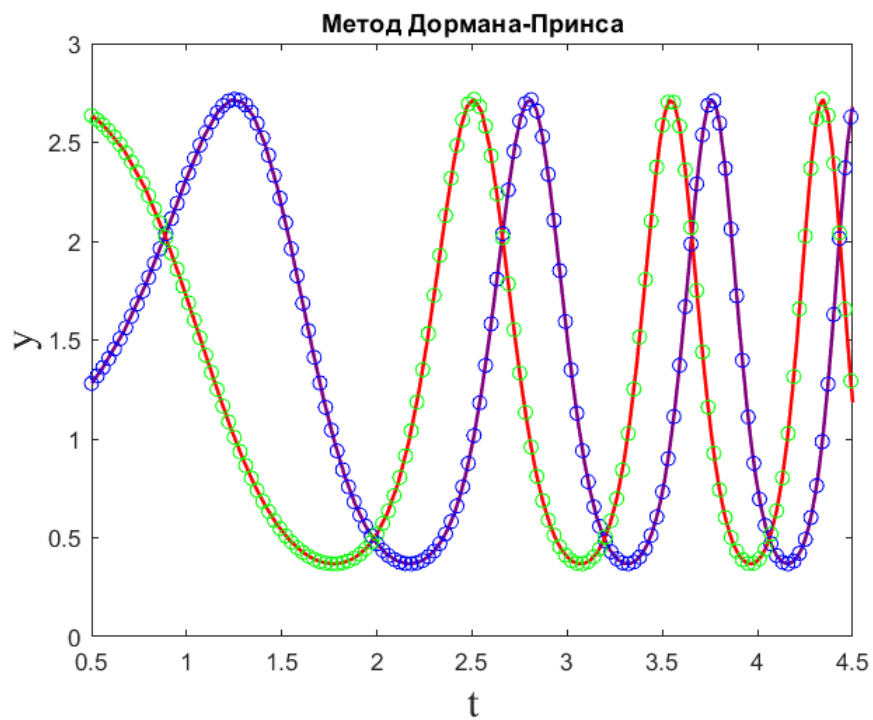


Рис. 16. Численное решение (сплошные линии), полученное методом Дормана-Принса при $tol = 10^{-6}$. Круги – аналитическое решение системы

5. Листинг кода

```

clear;
clc;
close all
N_var = 5;
t1 = N_var*0.1;
tau = 0.01;
t = t1:tau:t1+4;
tol = 1e-6;

%Метод Рунге-Кутты, s = 4, p = 4
a1 = [0 0 0 0; 1/3 0 0 0; -1/3 1 0 0; 1 -1 1 0];
b1 = [1/8 3/8 3/8 1/8];
c1 = [0 1/3 2/3 1];

%Метод Дорманда-Принса 5(4) (Порядок основного метода p = 5)
a2 = [0 0 0 0 0 0 0;
      1/5 0 0 0 0 0 0;
      3/40 9/40 0 0 0 0 0;
      44/45 -56/15 32/9 0 0 0 0;
      19372/6561 -25360/2187 64448/6561 -212/729 0 0 0;
      9017/3168 -355/33 46732/5247 49/176 -5103/18656 0 0;
      35/384 0 500/1113 125/192 -2187/6784 11/84 0];
b21 = [35/384 0 500/1113 125/192 -2187/6784 11/84 0];
b22 = [5179/57600 0 7571/16695 393/640 -92097/339200 187/2100 1/40];
c2 = [0 1/5 3/10 4/5 8/9 1 1];

sol_y1 = exp(sin(t.^2));
sol_y2 = exp(cos(t.^2));
sol = @(t) [exp(sin(t^2)) exp(cos(t^2))];
% y = zeros(length(t), 2);
y_analyt = sol(t(end));

%Решение методом Рунге-Кутты
% %Без автоматического выбора шага
% y0 = sol(t(1));
% for i = 2:length(t)
%     y_numeric = runge_kutta(t(i-1),y0, a1, b1, c1, tau);
%     y0 = y_numeric;
% end

%С автоматическим выбором шага
general_step_rk = 0;
rejected_step_rk = 0;
y0 = sol(t(1));
t_auto = t(1);

```

```

%Момент времени t, глобальная погрешность, локальная погрешность err,
%длина шага tau, численное решение (в текущее t)
control_rk = [t_auto 0 0 0 y0];
tic
while t_auto + tau < t(end)
    y1 = runge_kutta(t_auto,y0,a1,b1,c1, tau);
    y2 = runge_kutta(t_auto + tau,y1,a1,b1,c1, tau);
    omega = runge_kutta(t_auto, y0,a1, b1, c1, 2*tau);
    [flag, tau_new, err] = auto_step_rk(y2,omega,tau,tol);
    general_step_rk = general_step_rk + 1;
    if flag == 1
        y0 = y2;
        t_auto = t_auto + 2*tau;
        global_err_rk = max(abs(y2 - sol(t_auto)));
        control_rk(end+1,:) = [t_auto global_err_rk err tau y2];
    elseif flag == 0
        rejected_step_rk = rejected_step_rk + 1;
    end
    tau = tau_new;
end

y_rk = runge_kutta(t_auto,y2,a1,b1,c1, t(end) - t_auto);
toc
global_err_rk = max(abs(y_rk - y_analyt));
control_rk(end+1,:) = [t(end) global_err_rk global_err_rk t(end) - t_auto y_rk];

%Решение методом Дорманда-Принса
tau = 0.01;
t_auto = t(1);
y0 = sol(t(1));
general_step_dp = 0;
rejected_step_dp = 0;

%Момент времени t, глобальная погрешность, локальная погрешность err,
% длина шага tau, численное решение (для текущего t)
control_dp = [t_auto 0 0 0 y0];

tic
while t_auto + tau < t(end)
    [y_dp, err] = dormand_prince(t_auto,y0,a2,b21,b22,c2,tau);
    [flag, tau_new] = auto_step_dp(err,tau,tol);
    general_step_dp = general_step_dp + 1;
    if flag == 1
        y0 = y_dp;
        t_auto = t_auto + tau;
        global_err_dp = max(abs(y_dp - sol(t_auto)));
        control_dp(end+1,:) = [t_auto global_err_dp err tau y_dp];
    end
end

```

```

elseif flag == 0
    rejected_step_dp = rejected_step_dp + 1;
end
tau = tau_new;
end

[y_dp,err] = dormand_prince(t_auto,y_dp,a2,b21,b22,c2, t(end) - t_auto);
toc
global_err_dp = max(abs(y_dp - y_analyt));
control_dp(end+1,:) = [t(end) global_err_dp err t(end) - t_auto y_dp];

%Графики зависимости локальной погрешности от времени
figure;
plot(control_rk(1:end-1,1), control_rk(1:end-1,3),'LineWidth',1.5,...
'Color','#8B008B');
hold on
plot(control_dp(1:end-1,1), control_dp(1:end-1,3),'LineWidth',1.5,...
'Color','#FF4500');
yline(tol, 'k--', 'LineWidth',1.5);
set(gca,'FontSize', 14);
xlim([t(1) t(end)])
ylim([0 tol+0.5*tol])
xlabel('t','FontSize', 18,'FontName', 'Times New Roman')
ylabel('Локальная погрешность','FontSize', 18,'FontName', 'Times New Roman')
legend('Метод Рунге-Кутты','Метод Дормана-Принса', 'FontSize',18,...
'FontName', 'Times New Roman')

%Графики зависимости глобальной погрешности от времени
figure;
plot(control_rk(:,1), control_rk(:,2),'LineWidth',1.5,...
'Color',[0 0.4470 0.7410]));
hold on
plot(control_dp(:,1), control_dp(:,2),'LineWidth',1.5,...
'Color',[0.4940 0.1840 0.5560]));
set(gca,'FontSize', 14);
xlim([t(1) t(end)])
xlabel('t','FontSize', 18,'FontName', 'Times New Roman')
ylabel('Глобальная погрешность','FontSize', 18,'FontName', 'Times New Roman')
legend('Метод Рунге-Кутты','Метод Дормана-Принса', 'FontSize',18,...
'FontName', 'Times New Roman')

%Графики зависимости длины шага от времени
figure;
plot(control_rk(2:end,1), control_rk(2:end,4),'LineWidth',1.5,...
'Color','#228B22');
hold on
plot(control_dp(2:end,1), control_dp(2:end,4),'LineWidth',1.5,...

```



```

'Color','#FF4500');
set(gca,'FontSize', 14);
xlim([t(1) t(end)])
xlabel('t','FontSize', 18,'FontName', 'Times New Roman')
ylabel('Длина шага \tau','FontSize', 18,'FontName', 'Times New Roman')
legend('Метод Рунге-Кутты','Метод Дормана-Принса', 'FontSize',18,...
'FontName', 'Times New Roman')

```

```

%Графики зависимости численного решения (РК) от времени
figure;
plot(control_rk(:,1), control_rk(:,5),'LineWidth',1.5,...
'Color','#800080');
hold on
plot(control_rk(:,1), control_rk(:,6),'LineWidth',1.5,...
'Color','#FF0000');
xlim([t(1) t(end)])
plot(t, sol_y1,'bo','LineWidth',0.2,'MarkerIndices',1:3:length(t));
plot(t, sol_y2,'o','Color','00FF00','LineWidth',0.2,...
'MarkerIndices',1:3:length(t));
xlabel('t','FontSize', 18,'FontName', 'Times New Roman')
ylabel('y','FontSize', 18,'FontName', 'Times New Roman')
title('Метод Рунге-Кутты')

```

```

%Графики зависимости численного решения (ДП) от времени
figure;
plot(control_dp(:,1), control_dp(:,5),'LineWidth',...
1.5,'Color','#800080');
hold on
plot(control_dp(:,1), control_dp(:,6),'LineWidth',...
1.5,'Color','#FF0000');
xlim([t(1) t(end)])
plot(t, sol_y1,'bo','LineWidth',0.2,'MarkerIndices',1:3:length(t));
plot(t, sol_y2,'o','Color','00FF00','LineWidth',0.2,...
'MarkerIndices',1:3:length(t));
xlabel('t','FontSize', 18,'FontName', 'Times New Roman')
ylabel('y','FontSize', 18,'FontName', 'Times New Roman')
title('Метод Дормана-Принса')

```

```

function dydt = eqsystem(t,y)
dydt(1) = 2*t*y(1)*log(max(y(2),1e-3));
dydt(2) = -2*t*y(2)*log(max(y(1),1e-3));
end

```

```

function y = runge_kutta(t, y, a, b, c, tau)
s = length(c);
k = zeros(s,2);
k(1,:) = eqsystem(t, y);

```

```

for i = 2:s
    k(i,:) = eqsystem(t + c(i)*tau, y + tau*(a(i, 1:i-1)*k(1:i-1,:)));
end
y = y + tau*b*k;
end

function [flag,tau_new,err] = auto_step_rk(y2,omega,tau,tol)
p = 4;
% y2_updated = y2 + (y2-omega)/(2^p-1);
d = 1;
% d = abs(y2_updated);
factmax = 5;
factmin = 0.4;
fact = (0.25)^(1/(p+1));
err = max(abs(y2 - omega)./d)/((2^p - 1));
tau_opt = (tol/err)^(1/(p+1));
if err <= tol
    flag = 1;
    tau_new = tau*min(factmax, max(factmin, fact*tau_opt));
else
    flag = 0;
    tau_new = tau*min(1, max(factmin, fact*tau_opt));
end
end

function [y1,err] = dormand_prince(t,y,a,b1,b2,c,tau)
s = length(c);
k = zeros(s,2);
k(1,:) = eqsystem(t, y);
for i = 2:s
    k(i,:) = eqsystem(t + c(i)*tau, y + tau*(a(i, 1:i-1)*k(1:i-1,:)));
end
y1 = y + tau*b1*k;
y2 = y + tau*b2*k; %Оценщик погрешности
err = max(abs(y2-y1));
end

function [flag,tau_new] = auto_step_dp(err,tau,tol)
p = 5;
factmax = 5;
%factmax = 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5
factmin = 0.4;
%factmin = 0.2, 0.3, 0.4, 0.5, 0.6, 0.66
% fact = 0.8, 0.9, (0.25)^(1/(p+1)), (0.38)^(1/(p+1));
fact = (0.25)^(1/(p+1));
tau_opt = (tol/err)^(1/(p+1));
if err <= tol

```

```
    flag = 1;
    tau_new = tau*min(factmax, max(factmin, fact*tau_opt));
else
    flag = 0;
    tau_new = tau*min(1, max(factmin, fact*tau_opt));
end
end
```