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)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУ «Информатика и системы управления»

КАФЕДРА ИУ-1 «Системы автоматического управления»

ОТЧЕТ

по лабораторной работе №3
«Численные методы решения ОДУ»

по дисциплине
«Методы вычислений»

Выполнил: **Шевченко А.Д.**

Группа: **ИУ1-32Б**

Проверил: **Бобков А.В.**

Работа выполнена: **23.11.2022**

Отчет сдан:

Оценка:

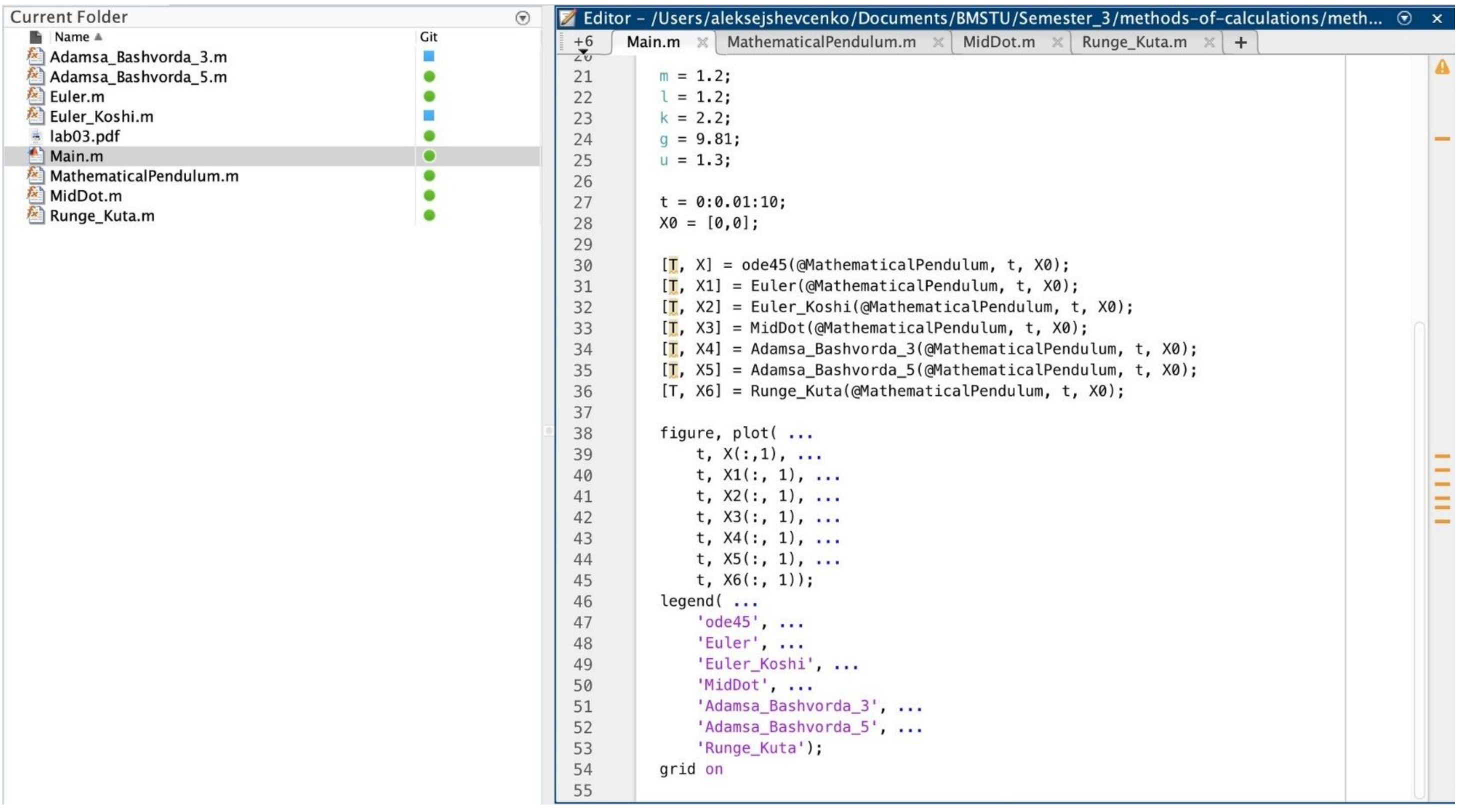
Москва 2022

Оглавление

Цель работы

Реализация численных методов решения обыкновенных дифференциальных уравнений (ОДУ). Использование встроенного MatLAB функции ode45, реализация метода Рунге-Кутты 4-го порядка и метода Адамса-Башфорта 5-го порядка. Оценка точности численного решения ОДУ.

Функция отображения графиков



Функция вектор дифференциального уравнения

```
function [Xp] = MathematicalPendulum (t, X)
%
% MathematicalPendulum решает дифференциальное уравнение математического
% маятника:
%
%     m * l^2 * Q'' + k * l^2 * Q' + m * g * l * sinQ = u
%
% с входными параметрами:
%
%     m = 1,2 kg
%     l = 1,2 m
%     k = 2,2 H/m
%     g = 9,81 H/m^2
%     u = 1,3 H*m
%

global m l k g u

Xp = zeros(2, 1);
Xp(1) = X(2);
Xp(2) = (u - k * l^2 * X(2) - m * g * l * sin(X(1))) / (m * l^2);

end
```

Различные методы решения ОДУ:

1. Метод Эйлер

```
function [T, X] = Euler(func, t, X0)
%
% Euler решает дифференциальное уравнение математического маятника
% методом Эйлера:
%
%     m * l^2 * Q'' + k * l^2 * Q' + m * g * l * sinQ = u
%
% с входными параметрами:
%
%     m = 1,2 kg
%     l = 1,2 m
%     k = 2,2 H/m
%     g = 9,81 H/m^2
%     u = 1,3 H*m
%

n = length(t);
m = length(X0);
T = t;
X = zeros(n, m);
X(1,:) = X0;

for i = 1:n - 1
    dt = T(i + 1) - T(i);
    K = dt * func (T(i), X(i, :));
    X(i + 1, :) = X(i, :) + K';
end
end

```

2. Метод Эйлера - Коши

```
function [T, X] = Euler_Koshi(func, t, X0)
%
%  Euler_Koshi решает дифференциальное уравнение математического
% маятника методом Эйлера-Коши:
%
%     m * l^2 * Q'' + k * l^2 * Q' + m * g * l * sinQ = u
%
% с входными параметрами:
%
%     m = 1,2 kg
%     l = 1,2 m
%     k = 2,2 H/m
%     g = 9,81 H/m^2
%     u = 1,3 H*m
%
n = length(t);
m = length(X0);
T = t;
X = zeros(n, m);
X(1, :) = X0;

for i = 1: n - 1
    dt = T(i + 1) - T(i);
    K1 = dt * func (T(i), X(i,  :));
    K2 = dt * func (T(i) + dt, X(i, :) + K1');
    X(i + 1, :) = X(i, :) + (K1' + K2') / 2;
end
end
```

3. Метод Средней точки

```
function [T, X] = MidDot(func , t, X0)
%
% MathematicalPendulum решает дифференциальное уравнение математического
% маятника методом средней точки:
%
%     m * l^2 * Q'' + k * l^2 * Q' + m * g * l * sinQ = u
%
% с входными параметрами:
%
%     m = 1,2 kg
%     l = 1,2 m
%     k = 2,2 H/m
%     g = 9,81 H/m^2
%     u = 1,3 H*m
%
n = length(t);
m = length(X0);
T = t;
X = zeros(n, m);
X(1, :) = X0;

for i = 1: n - 1
    dt = T(i + 1) - T(i);
    K1 = dt * func(T(i), X(i, :));
    K2 = dt * func(T(i) + dt / 2, X(i, :) + K1' / 2);
    X(i + 1, :) = X(i, :) + K2';
end
end
```

4. Метод Рунге – Кутта 4-го порядка

```
function [T, X] = Runge_Kuta(func, t, X0)
%
% Runge_Kuta решает дифференциальное уравнение математического
% маятника методом Рунге Кута:
%
%     m * l^2 * Q'' + k * l^2 * Q' + m * g * l * sinQ = u
%
% с входными параметрами:
%
%     m = 1,2 kg
%     l = 1,2 m
%     k = 2,2 H/m
%     g = 9,81 H/m^2
%     u = 1,3 H*m
%

n = length(t);
m = length(X0);
T = t;
X = zeros(n, m);
X(1, :) = X0;

for i = 1 : n-1
    dt = T(i+1)-T(i);
    K1 = dt * func(T(i),X(i,:));
    K2 = dt * func(T(i)+dt/2, X(i,:)+K1'/2);
    K3 = dt * func(T(i)+dt/2, X(i,:)+K2'/2);
    K4 = dt * func(T(i)+dt, X(i,:)+K1');
    X(i+1, :) = X(i,:)+1/6*(K1'+2*K2'+2*K3'+K4');
end

end
```

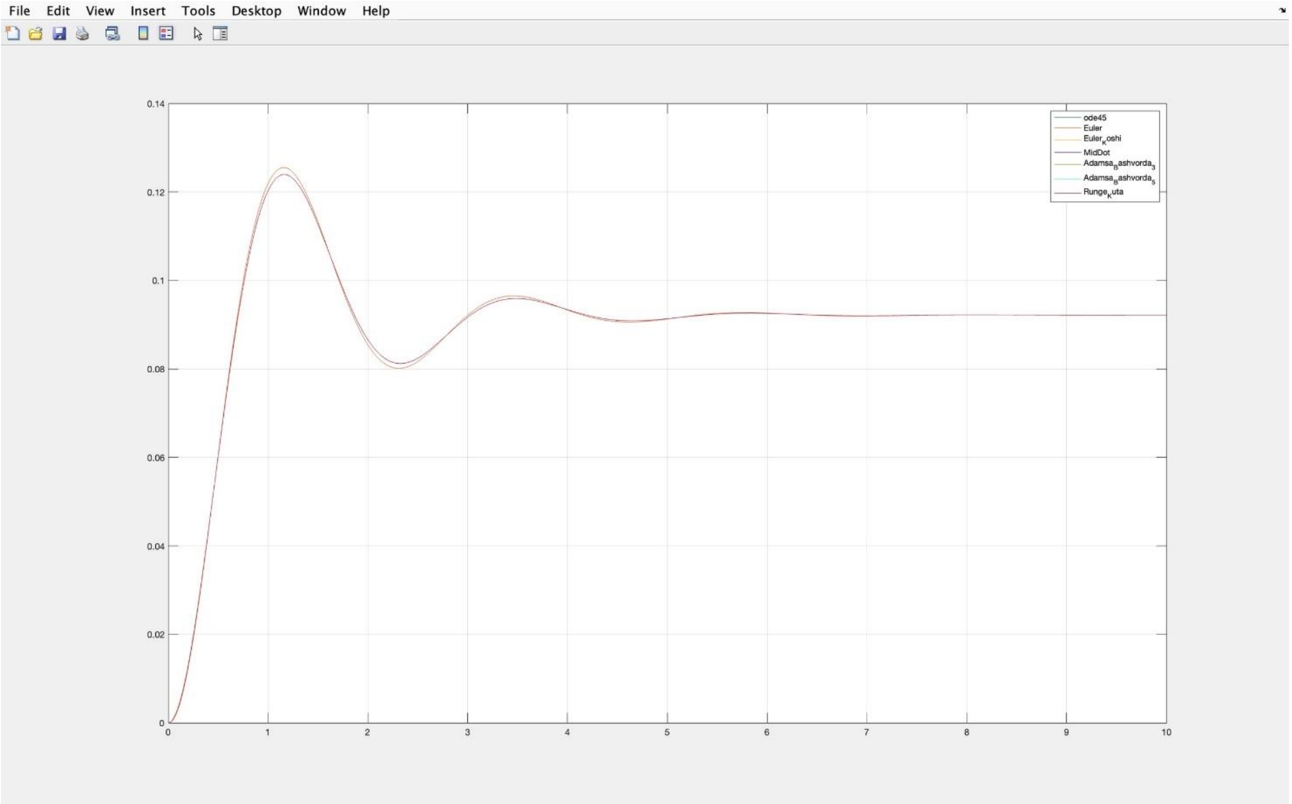
5. Метод Адамса – Башфорда 3-го порядка

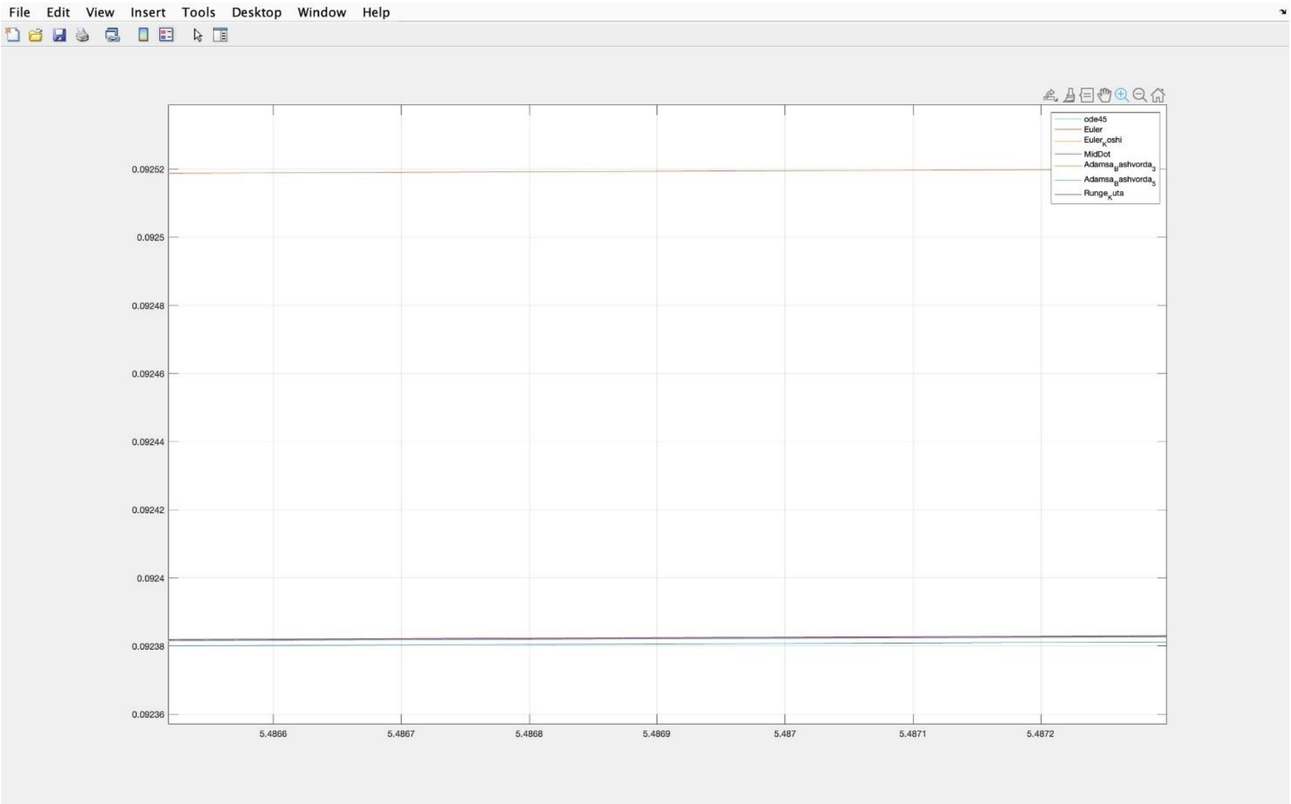
```
function [T, X] = Adamsa_Bashvorda_3(func, t, X0)
%
% Adams_Bashvorda_3 решает дифференциальное уравнение математического маятника
% методом Адамса Башворда 3-го порядка:
%
%     m * l^2 * Q'' + k * l^2 * Q' + m * g * l * sinQ = u
%
% с входными параметрами:
%
%     m = 1,2 kg
%     l = 1,2 m
%     k = 2,2 H/m
%     g = 9,81 H/m^2
%     u = 1,3 H*m
%

n=length(t);
m=length(X0);
T=t;
X=zeros(n,m);
X(1,:)=X0;

[T(1:3), X(1:3,:)] = Runge_Kuta(func, t(1:3), X0);
for i=3:n-1
    dt=T(i+1)-T(i);
    X(i+1,:)=X(i,:)+(23/12)*dt*func(t(i),X(i,:))'...
        -(4/3)*dt*func(t(i-1), X(i-1,:))'...
        +(5/12)*dt*func(t(i-2), X(i-2,:))';
end

end
```

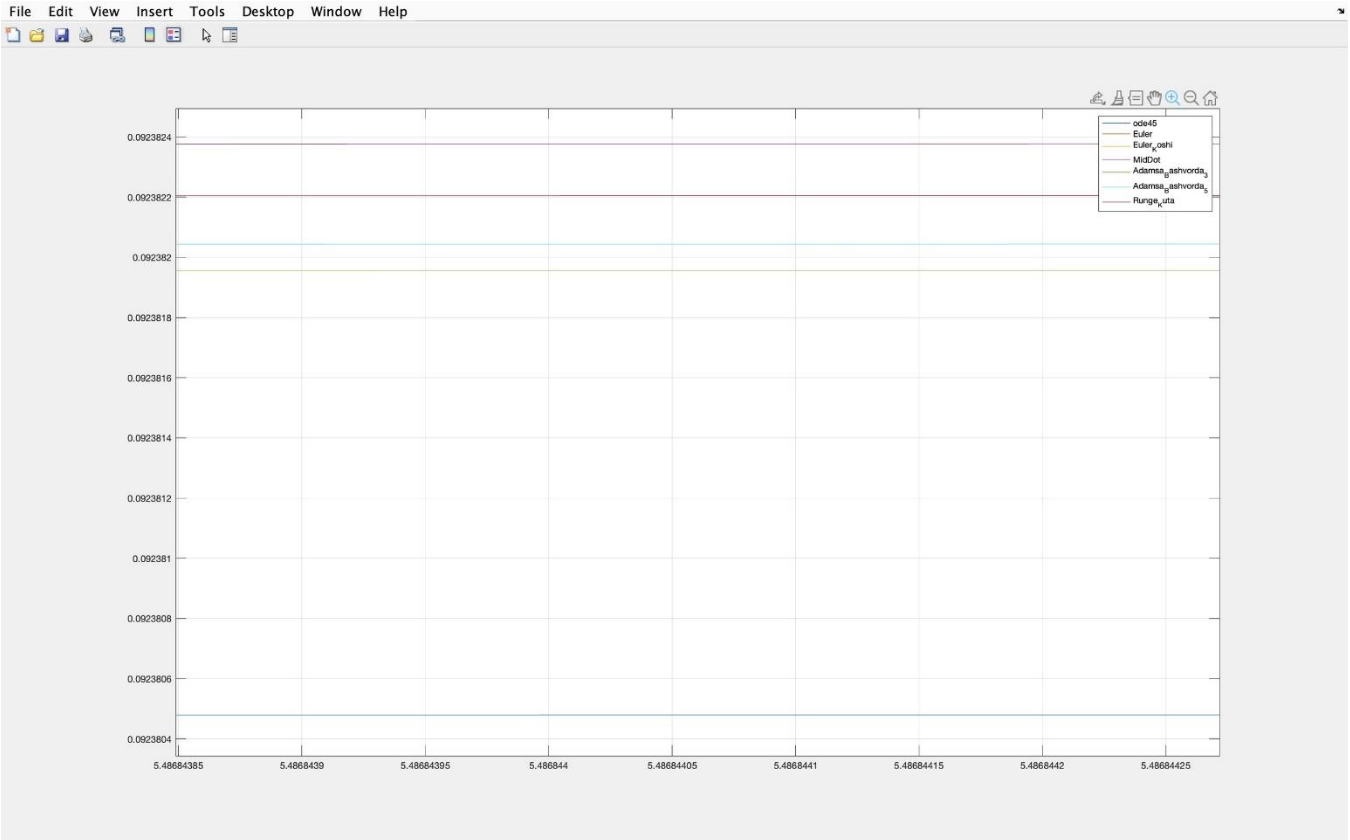
6. Метод Адамса – Башфорда 5-го порядка

```
function [T, X] = Adamsa_Bashvorda_5(func, t, X0)
%
% Adams_Bashvorda_5 решает дифференциальное уравнение математического маятника
% методом Адамса Башворда 5-го порядка:
%
%     m * l^2 * Q'' + k * l^2 * Q' + m * g * l * sinQ = u
%
% с входными параметрами:
%
%     m = 1,2 kg
%     l = 1,2 m
%     k = 2,2 H/m
%     g = 9,81 H/m^2
%     u = 1,3 H*m
%

n = length(t);
m = length(X0);
T = t;
X = zeros(n,m);
X(1,:) = X0;

[T(1:5),X(1:5,:)]= Runge_Kuta(func, t(1:5), X0);
for i = 5 : n-1
    dt = T(i+1)-T(i);
    X(i+1,:)=X(i,:)+(1901/720)*dt*func(t(i),X(i,:))'...
        -(1387/360)*dt*func(t(i-1), X(i-1,:))'...
        +(1308/360)*dt*func(t(i-2), X(i-2,:))'...
        -(637/360)*dt*func(t(i-3), X(i-3,:))'...
        +(251/720)*dt*func(t(i-4), X(i-4,:))';
end

end
```

Сравнение эффективности методов






File Edit View Insert Tools Desktop Window Help
ode45
Euler
Euler$_K$oshi
MidDot
Adamsa$_B$ashvorda$_3$
Adamsa$_B$ashvorda$_5$
Runge$_K$uta
0.0923824
0.0923822
0.092382
0.0923818
0.0923816
0.0923814
0.0923812
0.092381
0.0923808
0.0923806
0.0923804
5.48684385
5.4868439
5.48684395
5.486844
5.48684405
5.4868441
5.48684415
5.4868442
5.48684425

Вывод: