

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»  
(БГТУ им. В. Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и  
автоматизированных систем

### **Лабораторная работа № 1**

по дисциплине: Системное моделирование

тема: «Поведение механических системы в статике»

Выполнил: ст. группы ПВ-223

Игнатъев Артур Олегович

Проверил:

доц. Полунин Александр Иванович

Белгород 2024г.

## Лабораторная работа №1

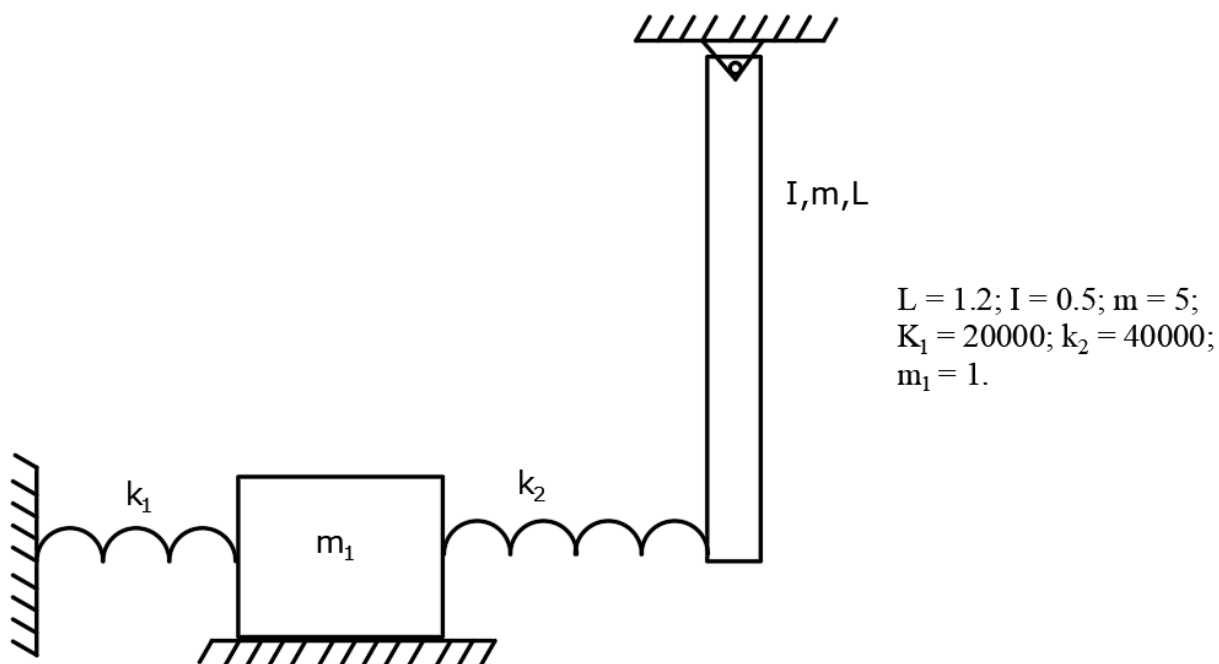
### «Поведение механических системы в статике»

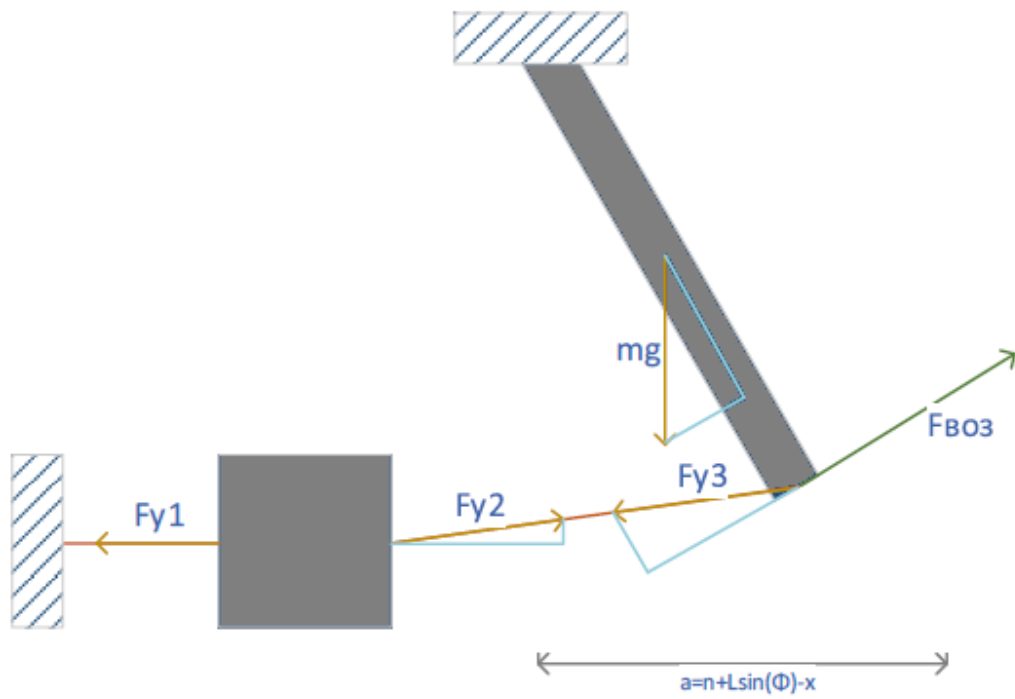
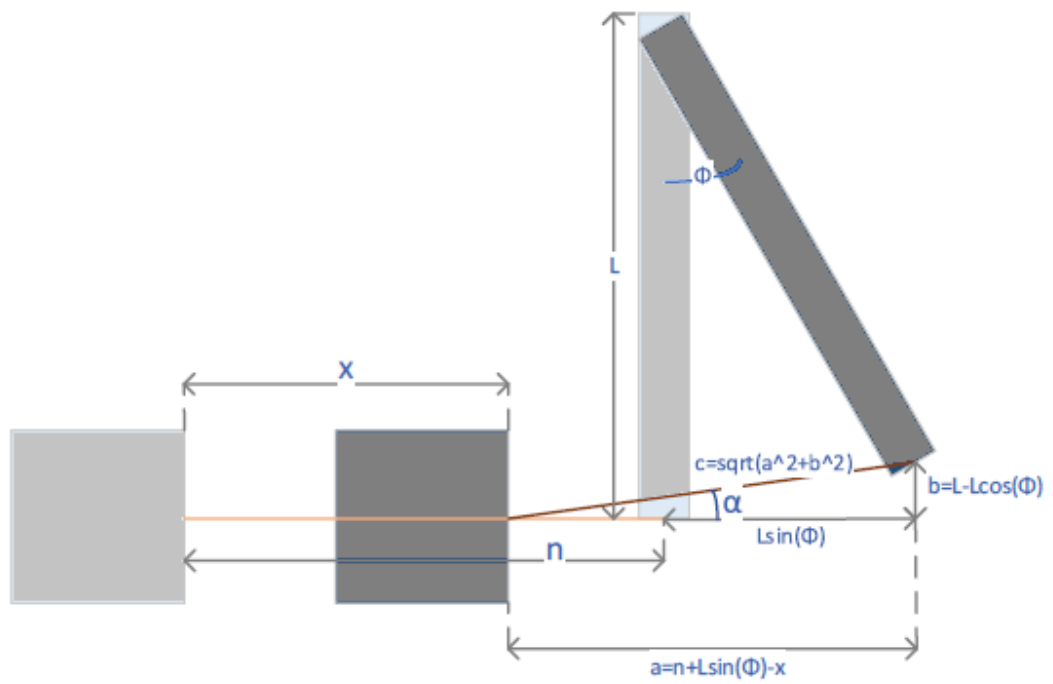
**Цель работы:** Изучить особенности организации вычислительных процессов, связанные с погрешностями, приближенным характером вычислений на компьютерах современного типа, вычислительной устойчивостью.

### Вариант 3

#### Ход выполнения лабораторной работы:

1. Разработать математическую модель, описывающую поведение элементов механической системы в статике.





$$\begin{cases} M_1 + M_2 + M_3 = 0 \\ \overrightarrow{F_{y1}} + \overrightarrow{F_{y2}} = 0 \end{cases}$$

$$M_1 = F_{BO3} * L$$

$$M_2 = -mg * \sin(\phi) * \frac{L}{2}$$

$$M_3 = -F_{y3} * \cos(\phi - \alpha) * L$$

$$\cos(\phi - \alpha) = \cos\alpha * \cos\phi + \sin\alpha * \sin\phi$$

$$\sin\alpha = \frac{a}{c}$$

$$\cos\alpha = \frac{b}{c}$$

$$c = \sqrt{a^2 + b^2}$$

$$a = n + L \sin \phi - x$$

$$b = L(1 - \cos\phi)$$

$$|F_{y3}| = |F_{y2}| = k_2 * \Delta l_2$$

$$\Delta l_2 = c - n = \sqrt{(n + L \sin \phi - x)^2 + L^2(1 - \cos\phi)^2} - n$$

$$M_3 = -k_2 \left( \sqrt{(n + L \sin \phi - x)^2 + L^2(1 - \cos\phi)^2} - n \right) *$$

$$\left( \frac{n + L \sin \phi - x}{\sqrt{(n + L \sin \phi - x)^2 + L^2(1 - \cos\phi)^2}} \cos\phi + \frac{L(1 - \cos\phi)}{\sqrt{(n + L \sin \phi - x)^2 + L^2(1 - \cos\phi)^2}} \sin\phi \right) * L$$

$$F_{y1} = k_1 * \Delta l_1$$

$$\Delta l_1 = x$$

$$F_{y1} = k_1 * x$$

$$F_{y2} = k_2 * \Delta l_2 * \cos\alpha$$

$$= k_2 \left( \sqrt{(n + L \sin \phi - x)^2 + L^2(1 - \cos\phi)^2} - n \right) *$$

$$* \frac{n + L \sin \phi - x}{\sqrt{(n + L \sin \phi - x)^2 + L^2(1 - \cos\phi)^2}}$$

$$\left\{ \begin{array}{l} F_{\text{BO3}} * L - mg * \sin(\phi) * \frac{L}{2} - k_2 * \\ \left( \sqrt{(n + L \sin \phi - x)^2 + L^2(1 - \cos\phi)^2} - n \right) * \\ \left( \frac{n + L \sin \phi - x}{\sqrt{(n + L \sin \phi - x)^2 + L^2(1 - \cos\phi)^2}} \cos\phi + \right. \\ \left. \frac{L(1 - \cos\phi)}{\sqrt{(n + L \sin \phi - x)^2 + L^2(1 - \cos\phi)^2}} \sin\phi \right) * L = 0 \\ -k_1 * x + k_2 \left( \sqrt{(n + L \sin \phi - x)^2 + L^2(1 - \cos\phi)^2} - n \right) * \\ \frac{n + L \sin \phi - x}{\sqrt{(n + L \sin \phi - x)^2 + L^2(1 - \cos\phi)^2}} = 0 \end{array} \right.$$

2. Разработать программу на основании математической модели и произвести расчёты.

```
clc; close all; clear;
fig = uifigure;
F = uislider(fig, "Position", [100 400 400 3], "ValueChangedFcn", @(F,event) Update(F.Value));
F.Limits = [0,10000];
F.Value = 0;
Update(F.Value);
GetBifurcationPoint();
T = uitable(fig, "Data", GetT);
function SetBP(newBP)
global BP
BP = newBP;
end
function x = GetBP
global BP
x = BP;
end
function SetT(newT)
global T
T = newT;
end
function x = GetT
global T
x = T;
end
function x = GetK1
x = 20000;
end
function x = GetK2
x = 40000;
end
function x = GetMass
x = 5;
end
function x = GetL
x = 1.2;
end
function x = GetN
x = 10;
end
function GetBifurcationPoint()
F = 0;
e = 0.00001;
step = 1000;
x0 = 0;
phi0 = 0;
[xr, phir, ex] = Newton(F, [x0 phi0]);
F_i = [F];
X_i = [xr];
Phi_i = [phir];
i = 2;
while (step > e)
F = F + step;
[xr, phir, ex] = Newton(F, [x0 phi0]);
if (ex)
F_i = [F_i;F];
X_i = [X_i;xr];
```

```

Phi_i = [Phi_i;phir];
end
while (ex)
F = F + step;
[xr, phir, ex] = Newton(F, [x0 phi0]);
i = i + 1;
if (ex)
F_i = [F_i;F];
X_i = [X_i;xr];
Phi_i = [Phi_i;phir];
end
end
F = F - step;
step = step / 2;
end
SetT(table(F_i, X_i, Phi_i));
SetBP([F xr phir]);
end
function Update(F)
syms x phi;
DrawPlots(F, [x phi]);
[xr, phir, ex] = Newton(F, [0 0]);
if (ex)
DrawPoint([xr phir], "g");
title("Static")
else
BP = GetBP;
str = "Bifurcation at F=" + string(BP(1)) + " x=" + string(BP(2)) + " phi=" +
string(BP(3));
title(str, "Color","r");
end
hold off;
OutputInf(F, [xr, phir]);
end
function OutputInf(F, PointR)
Fr = GetFunctions(F, PointR(1), PointR(2));
str = "F=" + string(F) + " R=" + string(Fr(1)) + " M=" + string(Fr(2));
xlabel(str);
end
function DrawPlots(F, Args)
Fun = GetFunctions(F, Args(1), Args(2));
Draw(Fun(1), "r", [0 pi/2]);
Draw(Fun(2), "b", [0 pi/2]);
end
function Fun = GetFunctions(F, x, phi)
Fun = [GetR(F, x, phi); GetM(F, x, phi)];
end
function R = GetR(F, x, phi)
L=GetL;
m=GetMass;
k2=GetK2;
k1=GetK1;
n=GetN;
sin_phi = sin(phi);
cos_phi = cos(phi);
a = n + L * sin_phi - x;
b = L * (1 - cos_phi);
c = sqrt(a^2+b^2);
cos_alpha = a / c;
d11 = x;
d12 = c - n;
Fy1 = k1 * d11;
Fy2 = k2 * d12;

```

```

R = -Fy1 + Fy2 * cos_alpha;
end
function M = GetM(F, x, phi)
L=GetL;
m=GetMass;
k2=GetK2;
k1=GetK1;
n=GetN;
sin_phi = sin(phi);
cos_phi = cos(phi);
a = n + L * sin_phi - x;
b = L * (1 - cos_phi);
c = sqrt(a^2+b^2);
cos_alpha = a / c;
sin_alpha = b / c;
cos_phi_minus_alpha = cos_alpha * cos_phi + sin_alpha * sin_phi;
dl2 = c - n;
mg = m * 9.81;
Fy2 = k2 * dl2;
M1 = -mg * sin_phi * L / 2;
M2 = F * L;
M3 = -Fy2 * cos_phi_minus_alpha * L;
M = M1 + M2 + M3;
end
function Draw(Fun, Color, Range)
graph(1) = ezplot(Fun, Range); hold on; grid on;
set(graph(1), "Color", Color);
end
function DrawPoint(Point, Color)
str = "(" + string(Point(1)) + "," + string(Point(2)) + ")";
text(Point(2), Point(1), str);
plot(Point(2), Point(1), "*", "Color", Color);
end
function Der = GetDerivativeRx(F, Point)
e = 0.000000001;
Der = (GetR(F, Point(1) + e, Point(2)) - GetR(F, Point(1), Point(2))) / e;
end
function Der = GetDerivativeRphi(F, Point)
e = 0.000000001;
Der = (GetR(F, Point(1), Point(2) + e) - GetR(F, Point(1), Point(2))) / e;
end
function Der = GetDerivativeMx(F, Point)
e = 0.000000001;
Der = (GetM(F, Point(1) + e, Point(2)) - GetM(F, Point(1), Point(2))) / e;
end
function Der = GetDerivativeMphi(F, Point)
e = 0.000000001;
Der = (GetM(F, Point(1), Point(2) + e) - GetM(F, Point(1), Point(2))) / e;
end
function w = GetW(F, Point0)
dRx = GetDerivativeRx(F, Point0);
dRphi = GetDerivativeRphi(F, Point0);
dMx = GetDerivativeMx(F, Point0);
dMphi = GetDerivativeMphi(F, Point0);
w = [dRx dRphi; dMx dMphi];
end
function LessEps = IsLessEps(A, Eps)
LessEps = abs(A) <= Eps;
end
function GreaterEps = IsGreaterEps(A, Eps)
GreaterEps = abs(A) > Eps;
end
function Delta = GetDelta(F, Point0)

```

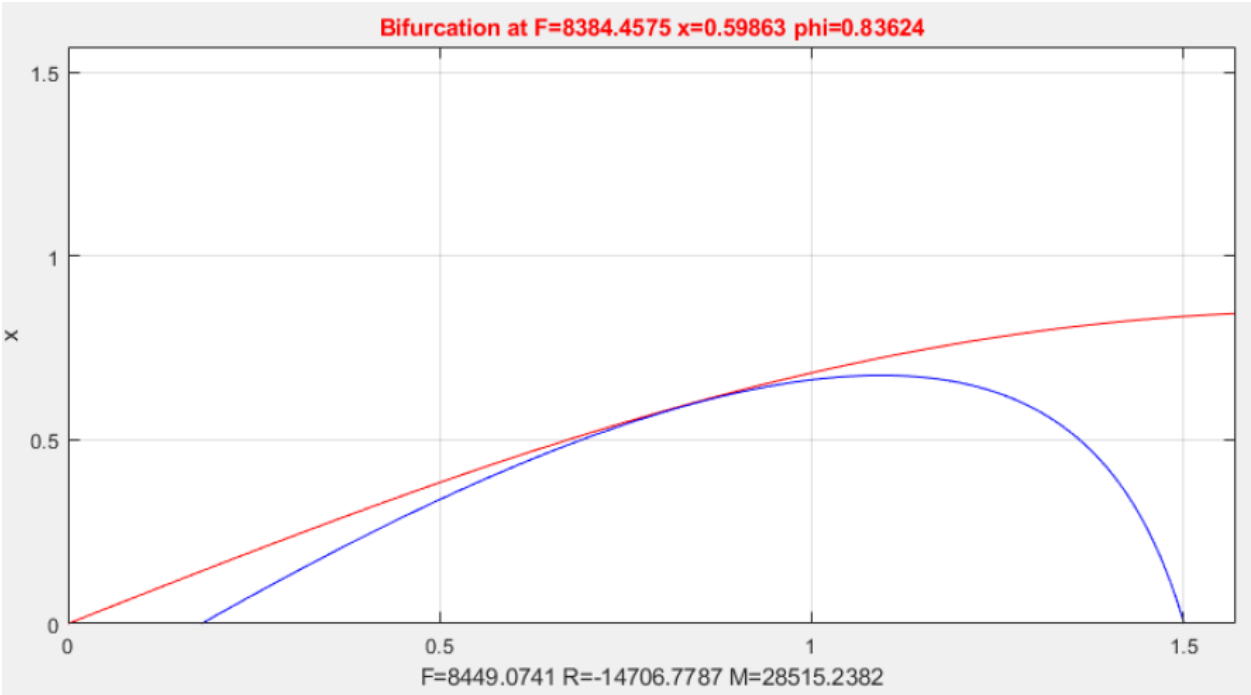
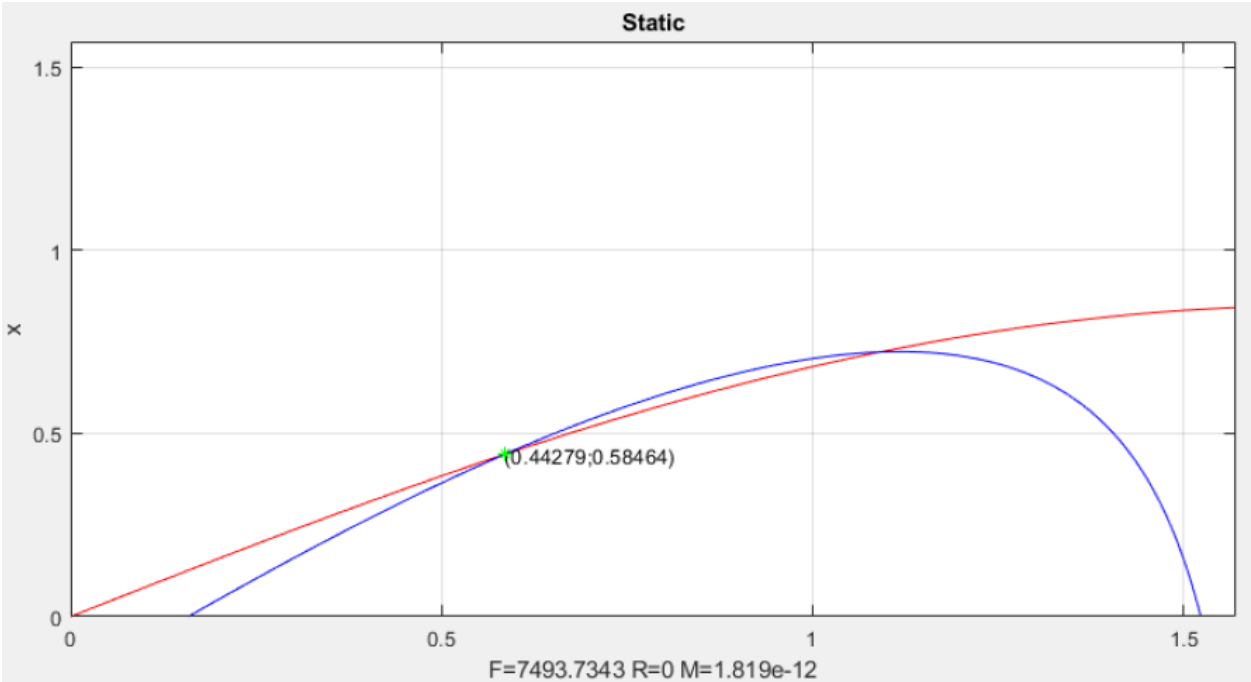
```

Delta = linsolve(GetW(F, Point0), GetFunctions(F, Point0(1), Point0(2)));
end
function [xr, phir, ex] = Newton(F, Point0)
e = 0.00000001;
Point(1) = Point0(1);
Point(2) = Point0(2);
k = 1;
Delta = GetDelta(F, Point);
while and(and(and(not(and(IsLessEps(GetM(F, Point(1), Point(2)), e), Is-
LessEps(GetR(F, Point(1), Point(2)), e))), Point(1) >= 0), Point(2) >= 0), Point(2)
<= pi/2)
Point = Point - Delta;
Delta = GetDelta(F, Point);
k = k + 1;
if (k > 1000)
break;
end
end
xr = Point(1);
phir = Point(2);
ex = and(IsLessEps(GetM(F, Point(1), Point(2)), e), IsLessEps(GetR(F, Point(1),
Point(2)), e));
end

```



Работа программы:



F <sub>i</sub>	X <sub>i</sub>	Phi <sub>i</sub>
0	0	0
1000	0.0500	0.0626
2000	0.1006	0.1261
3000	0.1525	0.1918
4000	0.2065	0.2610
5000	0.2638	0.3358
6000	0.3265	0.4199
7000	0.3994	0.5215
8000	0.4998	0.6714
8250	0.5418	0.7388
8375	0.5840	0.8104
8.3828e+03	0.5926	0.8255
8.3838e+03	0.5948	0.8294
8.3843e+03	0.5966	0.8327
8.3844e+03	0.5975	0.8342
8.3844e+03	0.5979	0.8349
8.3844e+03	0.5981	0.8353
8.3845e+03	0.5983	0.8357
8.3845e+03	0.5985	0.8360
8.3845e+03	0.5985	0.8361
8.3845e+03	0.5986	0.8362
8.3845e+03	0.5986	0.8362

**Вывод:** в ходе выполнения лабораторной работы были изучены особенности организации вычислительных процессов, связанные с погрешностями, приближенным характером вычислений на компьютерах современного типа, вычислительной устойчивостью.