Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

УТВЕРЖДАЮ
Зав. кафедрой системного анализа
и автоматического управления,
к.ф.-м.н., доцент
уч. ст., уч. зв.

________________ И.Е. Тананко
подпись, дата инициалы, фамилия

ОТЧЕТ О ПРАКТИКЕ

Студента 2 курса факультета КНиИТ направления 27.03.03 – Системный анализ и управление

Черневского Алексея Дмитриевича
фамилия, имя, отчество

Учебная (ознакомительная) практикыа
вид практики

кафедра системного анализа и автоматического управления
кафедра

курс 2

семестр 4

продолжительность 2 недели, с 30.06.2017 г. по 13.07.2017 г.
кол. недель, сроки практики

Руководитель практики,
доцент кафедры
системного анализа
и автоматического
управления,
к.ф.-м.н., доцент ________________ Е.С. Рогачко
должность, уч. ст., уч. зв. подпись, дата инициалы, фамилия

Саратов, 2017

Тема практики:

«Программные средства математического моделирования.

Вариант №9»

Оглавление

ВВЕДЕНИЕ

GNU Octave – свободная система для математических вычислений, использующая совместимый с M*ATLAB* язык высокого уровня. С помощью GNU Octave можно решить такие задачи, как: выполнение простейших вычислений, табулирование функций, решение СЛАУ, вычисление интегралов, решение дифференциальных уравнений и поиск оптимального решения [1 – 5].

Цель ознакомительной практики – закрепить теоретические знания по системному анализу и математическому моделированию и получить навыки их практического применения путем решения задач в среде разработки Octave.

На ознакомительной практике решались следующие задачи:

1. Табулирование функций;
2. Решение систем линейных алгебраических уравнений;
3. Аппроксимация функций;
4. Решение обыкновенных дифференциальных уравнений;
5. Вычисление определённых интегралов;
6. Решение нелинейных уравнений;
7. Поиск минимума функции одной переменной;
8. Поиск минимума функций нескольких переменных;
9. Моделирование и исследование примера детерминированной системы;
10. Моделирование и исследование примера стохастической системы.

1 Математические вычисления

1.1 Табулирование функций

Задача 1.1. Составить программу вычисления значений функции y_i для значений аргумента x_i. Данные взять из таблицы 1.

Таблица 1 – Исходные данные для задачи

№ п/п	Функция $y_i = f(x_i)$	Задача А					Задача В				
		a	b	x_H	x_k	Δx	x_1	x_2	x_3	x_4	x_5
9	$y = \frac{lg(x^2-1)}{\log_5(ax^2-b)}$	1.1	0.09	1.2	2.2	0.2	1.21	1.76	2.53	3.48	4.52

Код программы А:

```
a = 1.1; b = 0.09;
x = 1.2 : 0.2 : 2.2;

y = log(x.^2 - 1) ./ (log(a * x.^2 .- b) ./ log(5)); [x; y]
```

Результаты работы программы А:

```
y = -3.291304
y = -0.090545
y = 0.713671
y = 1.042292
y = 1.210283
y = 1.308293
```

Код программы B:

```
a = 1.1; b = 0.09;
x(1) = 1.21; x(2) = 1.76;
x(3) = 2.53; x(4) = 3.48;
x(5) = 4.52;

for i = 1 : 5
 y(i) = log(x(i)^2 - 1) / (log(a * x(i)^2 - b) / log(5));

end;
```

Результаты работы программы B:

```
y = -2.94835
y = 0.99424
y = 1.39999
y = 1.50056
y = 1.53618
```

1.2 Решение систем линейных алгебраических уравнений

Задача 1.2. Решить систему линейных алгебраических уравнений. Данные приведены в таблице 2.

Таблица 2 – Исходные данные для задач

9	$a_{ij}=\begin{bmatrix} 1 & 2 & 15 & 11 \\ -2 & 4 & 3 & 61 \\ -3 & -8 & 11 & 12 \\ 15 & 7 & 8 & -4 \end{bmatrix}$; $b_i=\begin{bmatrix} 2 \\ 4 \\ 5 \\ 9 \end{bmatrix}$.

Код программы:

```
a = [1 2 15 11;
   -2 4 3 61;
   3 -8 11 12;
```

```
  15 7 8 -4];
b = [2; 4; 5; 9];
X = a \ b

X2 = b' / a'

X3 = b' * a'^(-1)

X4 = b' * inv(a')
```

Результаты работы программы:

```
X = 0.684009  -0.169090  0.039021  0.097169

X2 = 0.684009  -0.169090   0.039021   0.097169

X3 = 0.684009  -0.169090   0.039021   0.097169

X4 = 0.684009  -0.169090   0.039021   0.097169
```

1.3 Аппроксимация функций

Задача 1.3. Используя линейную и полиномиальную аппроксимации, получить эмпирические формулы и оценить их погрешность для функции $y=f(x)$, заданной в табличном виде:

Таблица 3 – Исходные данные для задач

x_i	10.0	12.0	13.0	15.0	18.0	20.0	21.0
y_i	0.66	0.89	1.24	1.36	1.56	1.76	1.92

Код программы:

```
x = [10.0 12.0 13.0 15.0 18.0 20.0 21.0]
y = [0.66 0.89 1.24 1.36 1.56 1.76 1.92]

plot(x, y, 'o')
hold on

[p1, s1] = polyfit(x, y, 1)
y1 = polyval(p1, x);
plot(x, y1, 'g')
hold on

[p1, s1] = polyfit(x, y, 2)
y1 = polyval(p1, x);
plot(x, y1, 'k')
hold on

[p1, s1] = polyfit(x, y, 3)
y1 = polyval(p1, x);
plot(x, y1, 'm')
hold on

[p1, s1] = polyfit(x, y, 4)
y1 = polyval(p1, x);
plot(x, y1, 'b')
hold on

[p1, s1] = polyfit(x, y, 5)
y1 = polyval(p1, x);
plot(x, y1, 'r')
hold on

[p1, s1] = polyfit(x, y, 6)
y1 = polyval(p1, x);
plot(x, y1, 'c')
hold on

h = legend("graph", "polyfit (k = 1)", "polyfit (k = 2)", "polyfit (k = 3)",
"polyfit (k = 4)", "polyfit (k = 5)", "polyfit (k = 6)");
```

Результаты работы программы:

Эмпирические формулы и погрешности представлены на рисунке 1.

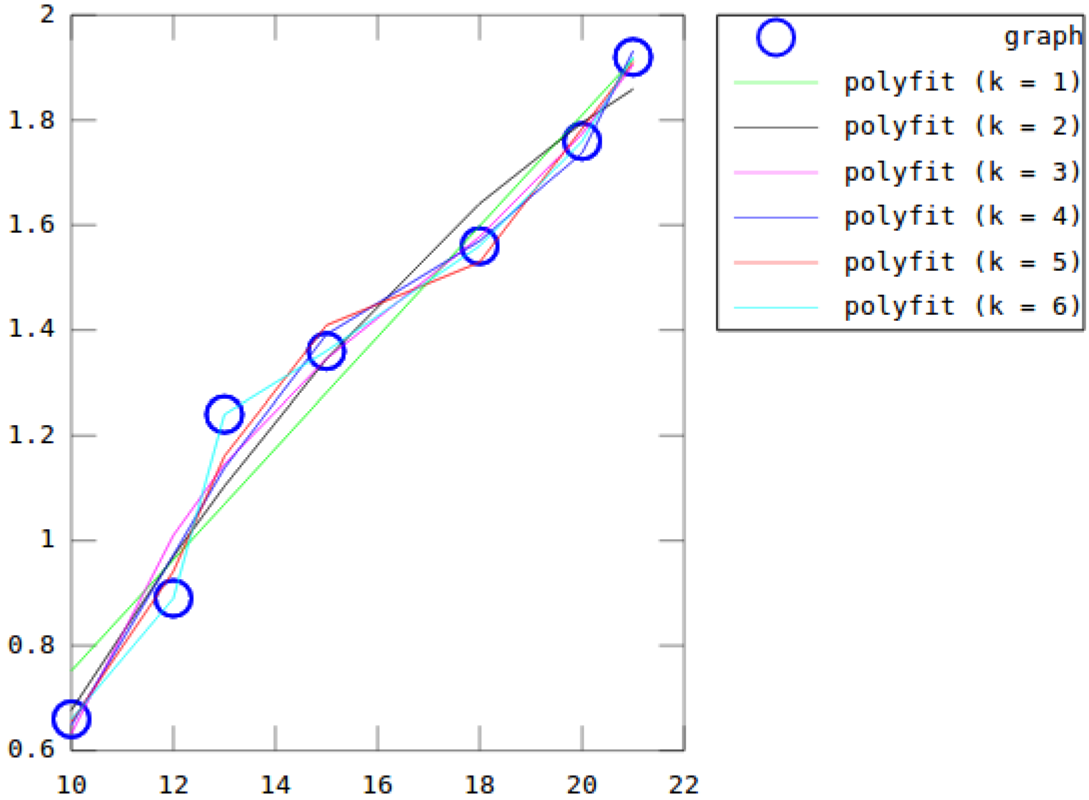

Рисунок 1 – Эмпирические формулы и погрешности.

1.4 Решение обыкновенных дифференциальных уравнений.

Задача 1.4. Построить график и вывести в виде таблицы решение задачи Коши на интервале $[0,1]$ методом Рунге-Кутта. Данные взяты из таблицы 4.

Таблица 4 – Исходные данные для задачи

№ п/п	$f(x,y)$	y_0
9	$x^2 \cos y + 0.1$	0.3

Код программы:

```
function y = g(t, x)
  y = t^2 * cos(x) + 0.1;
end

par = odeset ("RelTol", 1e-5, "AbsTol", 1e-5, "InitialStep", 0.1, "MaxStep", 0.15);
[X23, Y23] = ode45(@g, [0 1], 0.3, par)
```

```
plot(X23, Y23);
```

Результаты работы программы:

График решения задачи Коши представлен на рисунке 2.

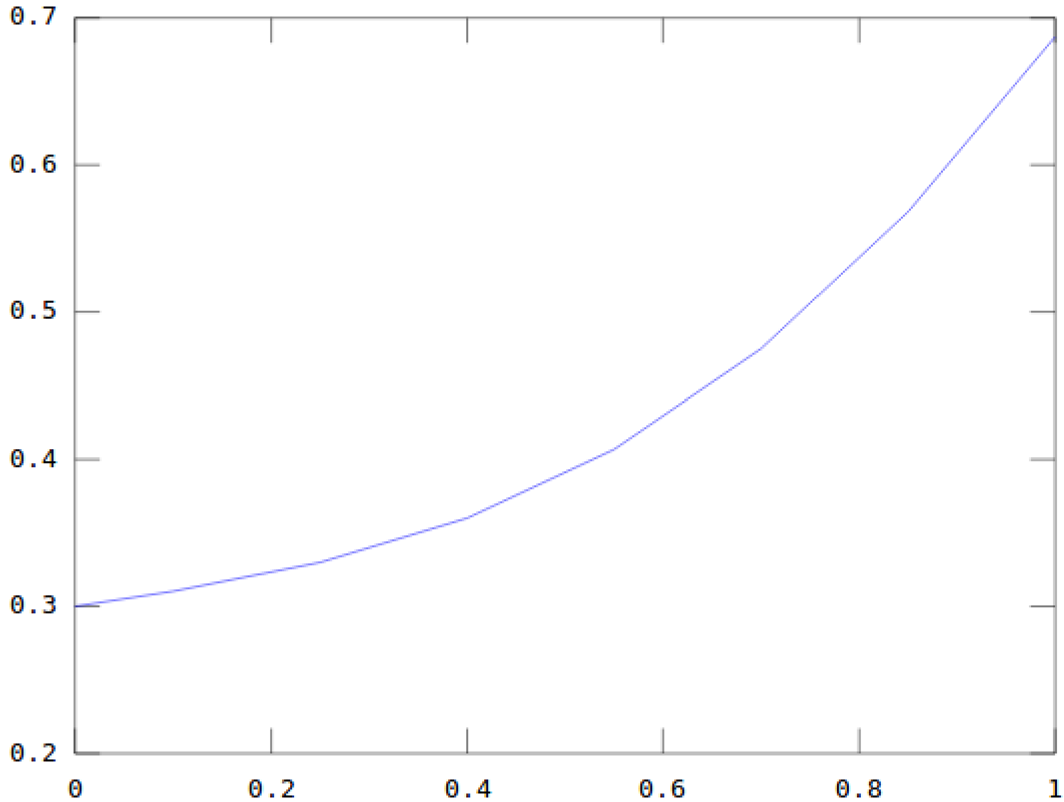

Рисунок 2 – График решения задачи Коши.

Решение задачи Коши:

X =	*Y* =
0.00000	0.30000
0.10000	0.31032
0.25000	0.32994
0.40000	0.36011
0.55000	0.40674

0.70000	0.47496
0.85000	0.56825
1.00000	0.68721

1.5 Вычисление определённых интегралов.

Задача 1.5. Вычислить и вывести на экран значения определенного интеграла методом Симпсона и методом трапеций. Данные взять из таблицы 5.

Таблица 5 – Исходные данные для задачи

№ п/п	Подынтегральная функция $f(x)$	Интервал интегрирования $[a,b]$	Точность вычислений интеграла
9	$\sin(1/x)x^4$	$[1.0; 2.5]$	0.0005

Код программы:

```
x = 1.0 : 0.0005 : 2.5;
y = sin(1./x).*x.^4;

sim = trapz(x, y)

clear x y z;

tr = quad(inline("(sin(1./x).*x.^4)"), 1.0, 2.5, 0.0005)
```

Результат работы программы:

```
sim =  9.0857
tr =  9.0857
```

Как видно, решение определенного интеграла методом Симпсона и трапеций дает одинаковые ответы, с точностью до сотых.

1.6 Решение нелинейных уравнений.

Задача 1.6. Построить график и найти корень нелинейного уравнения. Данные взяты из таблицы 6.

№ п/п	Уравнение $f(x)=0$	Отрезок $[a, b]$
9	$x^4+2x^3-x-1=0$	$[0.8; 1.0]$

Таблица 6 – Исходные данные для задачи.

Код программы:

```
x = 0.8 : 0.001 : 1.0;

y = x.^4 + 2*x.^3 - x - 1;

plot(x, y);

x1 = fzero(inline("x.^4 + 2*x.^3 - x - 1"), [0.8 1.0])

x2 = fsolve(inline("x.^4 + 2*x.^3 - x - 1"), 0.8 : 1.0)
```

Результаты работы программы:

```
x1 =  0.86676
x2 =  0.86676
```

На рисунке 3 показан график нелинейного уравнения, точечное решение которого было получено с помощью функций *fsolve* и *fzero*.

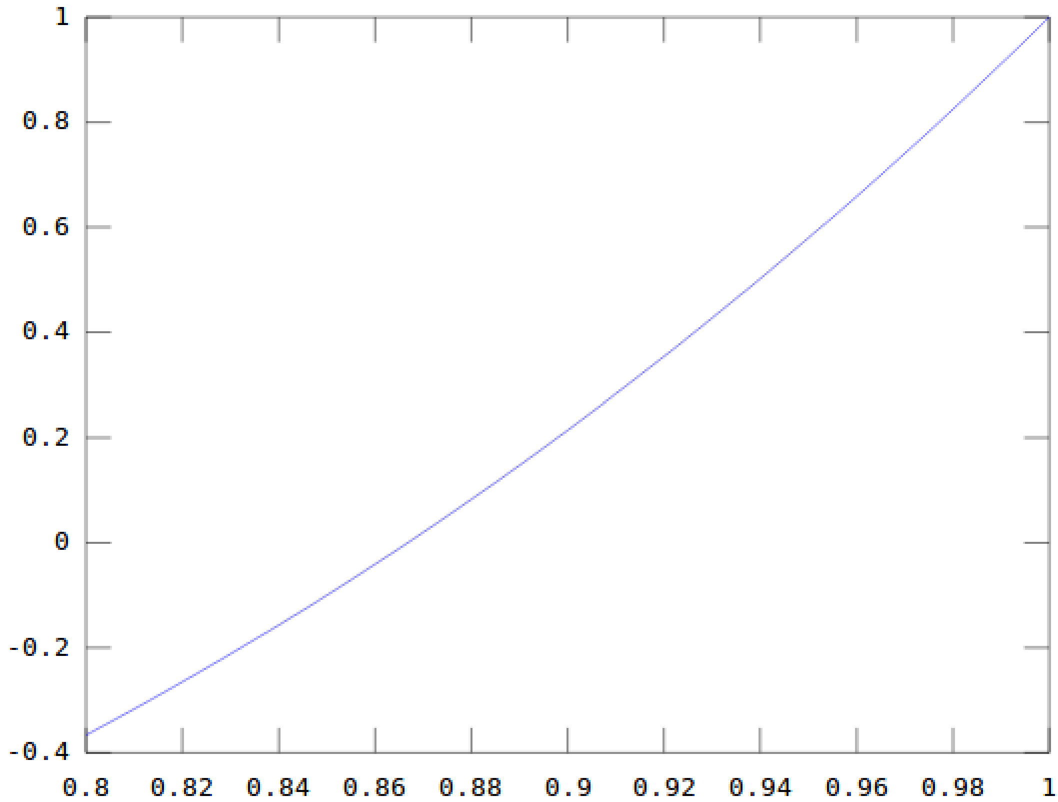

Рисунок 3 – График нелинейного уравнения.

1.7 Поиск минимума функции одной переменной.

Задача 1.7. Найти и вывести на экран координату и минимальное значение функции $f(x)$ на $[a,b]$. Данные взяты из таблицы 7.

Таблица 7 – Исходные данные для задачи.

№ п/п	Функция $f(x)$	Отрезок $[a,b]$
9	$f(x)=(x-2)^5(2x+1)^4$	$[-0.5, 1.5]$

Код программы:

```
x = -0.5 : 0.001 : 1.5;

y = (x - 2).^5 .* (2*x + 1).^4;
```

```
plot(x, y);

[x1, y1] = fminbnd(inline("(x - 2).^5 .* (2*x + 1).^4"), -0.5, 20.0)
```

Результаты работы программы:

```
x1 = 0.61111
y1 = -126.03
```

На рисунке 4 показан график, из которого видно соответствие найденных программой точки минимума и минимального значения функции $f(x)$.

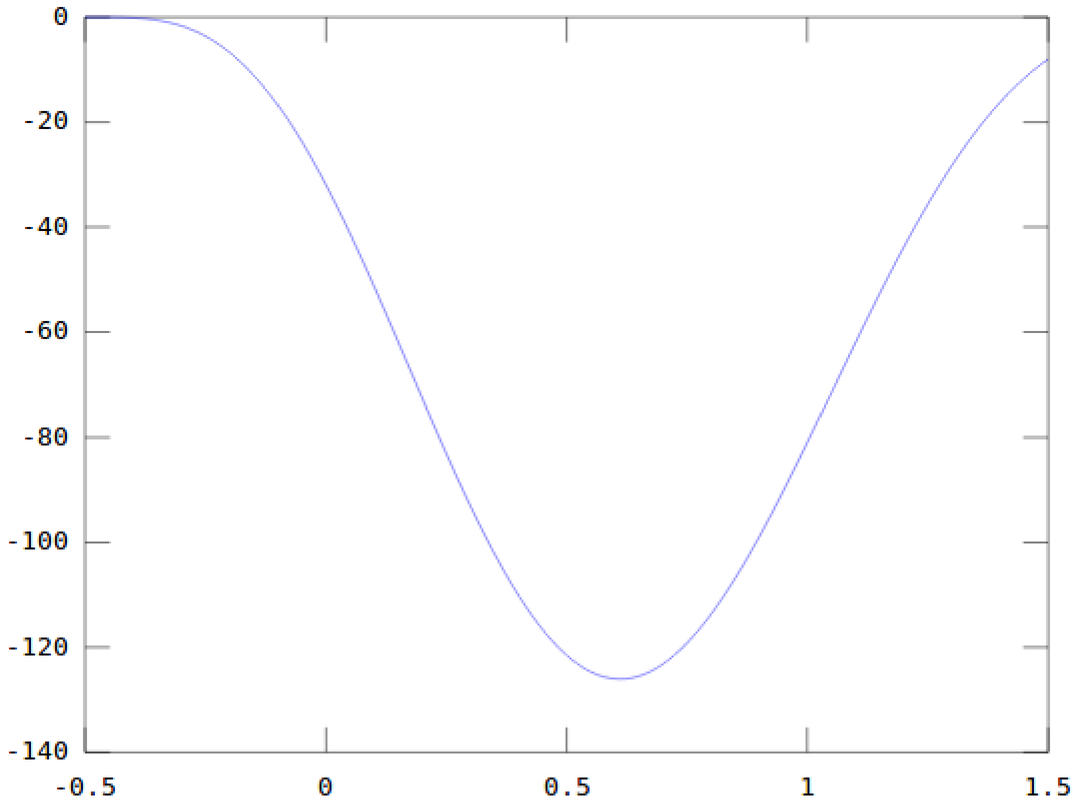

Рисунок 4 – График функции $f(x)$.

1.8 Поиск минимума функций нескольких переменных.

Задача 1.8. Найти и вывести на экран координаты и минимальное значение двух переменных. Поиск начать с точки $M(x, y)$. Данные взяты из таблицы 8.

Таблица 8 – Исходные данные для задачи.

№ п/п	Функция $f(x, y)$	Координаты начальной точки $M_0(x_0, y_0)$
9	$\ln(1+x^2+y^2)^2+(x-y-1)^2$	$(2,2)$

Код программы:

```
[x, y] = meshgrid (-0.5 : 0.01 : 0.5, 0 : 0.01 : 0.5);

z = log(1 + x.^2 + y.^2).^2 + (x - y -1).^2;

plot3 (x, y, z)

function f = Fxy(x)
    f = log(1 + x(1)^2 + x(2)^2)^2 + (x(1) - x(2) - 1)^2;
end;

[xmin, minf] = fminsearch (@ Fxy, [2; 2])
```

Результаты работы программы:

```
xmin =

   0.41074

  -0.41071

minf =  0.11640
```

На рисунке 5 построен трехмерный график функции $f(x, y)$, с помощью которого можно убедиться в наличии минимума функции.

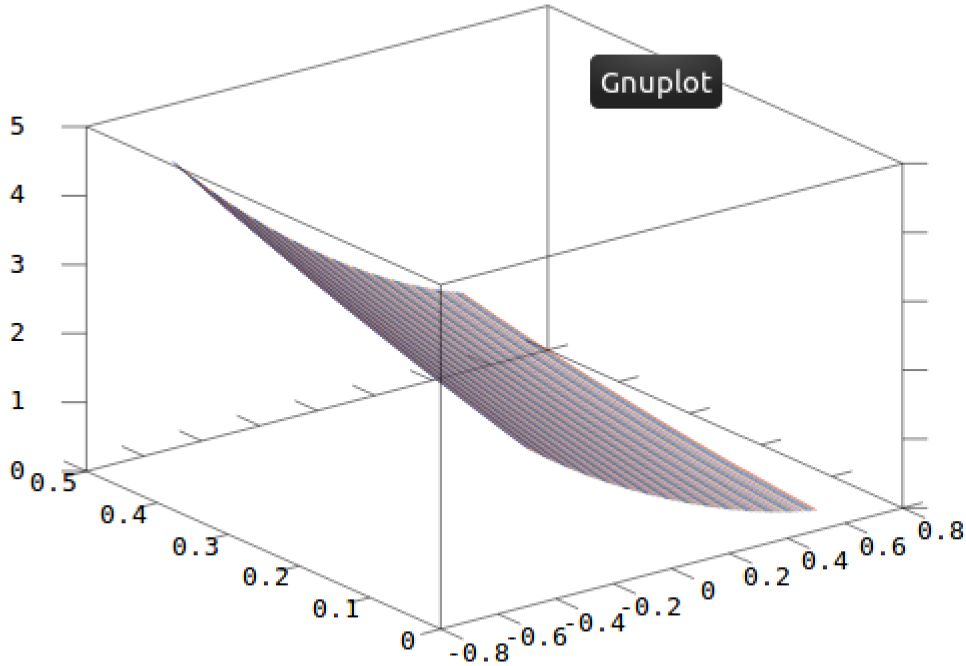

Рисунок 5 – График функции $f(x, y)$.

2 Математические модели систем

2.1 Детерминированные системы. Спящий полицейский

Формулировка задачи

Автомобиль движется по ровной дороге и наезжает на искусственное препятствие – « спящий полицейский ». Исследовать кинематику и динамику движения автомобиля .

Ограничимся рассмотрением наезда на препятствие только одного колеса . Горизонтальная составляющая скорости автомобиля не меняется . Воздействие препятствия сводится только к возбуждению вертикального перемещения автомобиля . Колесо при движении полностью повторяет профиль препятствия . Подвеска состоит из упругой пружины и демпфера.

Математическая модель

При наезде колеса на препятствие колесо перемещается в вертикальном направлении. Это перемещение описывается переменной x. На кузов автомобиля воздействие со стороны дороги передается посредством подвески (пружины с жесткостью K и демпфера с коэффициентом демпфирования B). Силовое воздействие посредством пружины определяется относительным смещением кузова , описываемым переменными x и y, силовое воздействие со стороны демпфера – относительной скоростью этих перемещений dx/dt и dy/dt . В уравнении движения постоянное воздействие на пружину кузова , компенсируемое равной и противоположно направленной силой упругости пружины ($Mg = K\,\Delta x$), не будем учитывать.

С учетом сделанных предположений в соответствии со вторым законом Ньютона уравнение движения рассматриваемой системы имеет вид :

$$M\frac{d^2y}{dt^2} = B\left(\frac{dx}{dt} - \frac{dy}{dt}\right) + K(x - y),$$

$$\frac{d^2y}{dt^2} + \frac{B}{M}\frac{dy}{dt} + \frac{K}{M}y = \frac{B}{M}\frac{dx}{dt} + \frac{K}{M}x.$$

Вертикальное ускорение кузова автомобиля d^2y/dt^2 является функцией скорости автомобиля в горизонтальном направлении, так как горизонтальное перемещение вследствие неровности сопровождается вертикальным перемещением. При этом профиль дороги (кривизна, ширина, высота) играет существенную роль.

Выберем в качестве математической модели неровности дороги функцию:

$$x(s) = \frac{H}{2}\left(1 - \cos\left(\frac{2\pi s}{L}\right)\right),\ 0 \le s \le L,$$

где H – высота , а L – ширина неровности .

При постоянной скорости V_0 автомобиля в горизонтальном направлении $s = V_0 t$, функция профиля дороги и ее производная принимают вид:

$$x(t)=\frac{H}{2}\left(1-\cos\left(\frac{2\pi V_0 t}{L}\right)\right), 0 \le t \le L/V_0,$$

$$\frac{dx}{dt}=\frac{H}{L}\pi V_0 \sin\left(\frac{2\pi V_0 t}{L}\right).$$

Подставив dx/dt в уравнение движения и выполнив соответствующую замену переменных, получим систему двух дифференциальных уравнений первого порядка.

Задания

1. Решите систему дифференциальных уравнений и постройте графики зависимости перемещения, скорости и ускорения вдоль вертикальной оси от времени.

Код программы:

```
clc
clear all
global H = 0.05;
global L = 0.8;
global V0 = 20;
global M = 450;
global K = 3500;
global B =7300;
function q = sist(t, an)
global H;
global L;
global V0;
global M;
global K;
global B;
#Описание математической модели неровности дороги
x = (H / 2) * (1 - cos((2 * pi * V0 * t) / L));
#Производная функции неровности дороги
dxdt = (H / L) * pi * V0 * sin((2 * pi * V0 * t)/L);
q = [an(2); ((B / M) * dxdt + (K / M) * x - (B / M) * an(2) - (K / M)*an(1))];
endfunction;

t0 = 0;
tf =  L / V0;
x0 = [0, 0];

par = odeset ("RelTol", 1e-5, "AbsTol", 1e-5, "InitialStep", 0.05,
```

```
"MaxStep", 0.001);
[t, an] = ode45(@sist, [t0 tf], x0, par);
plot (t, an(:,1));
title('nazvanie');
xlabel('t');
ylabel('y');
figure();
plot (t, an(:,2));
xlabel('t');
ylabel('dy/dt');

x = (H / 2) * (1 - cos((2 * pi * V0 * t) / L));
dxdt = (H / L) * pi * V0 * sin((2 * pi * V0 * t)/L);
an3=((B / M) * dxdt + (K / M) * x - (B / M) * an(:,2) - (K / M)*an(:,1));

figure();
plot (t, an3);
xlabel('t');
ylabel('d^2y/dt^2');
```

Результатом работы программы являются три графика зависимости перемещения, скорости и ускорения вдоль вертикальной оси от времени.

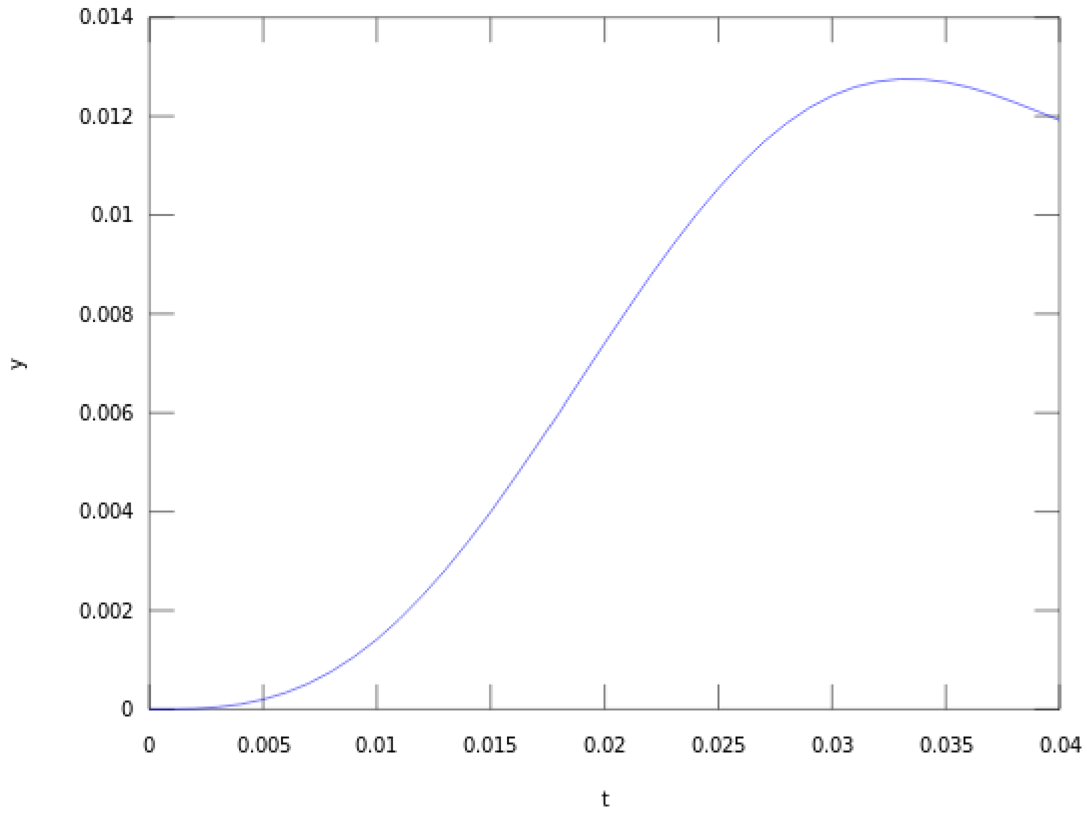

Рисунок 6 – График зависимости перемещения от времени.

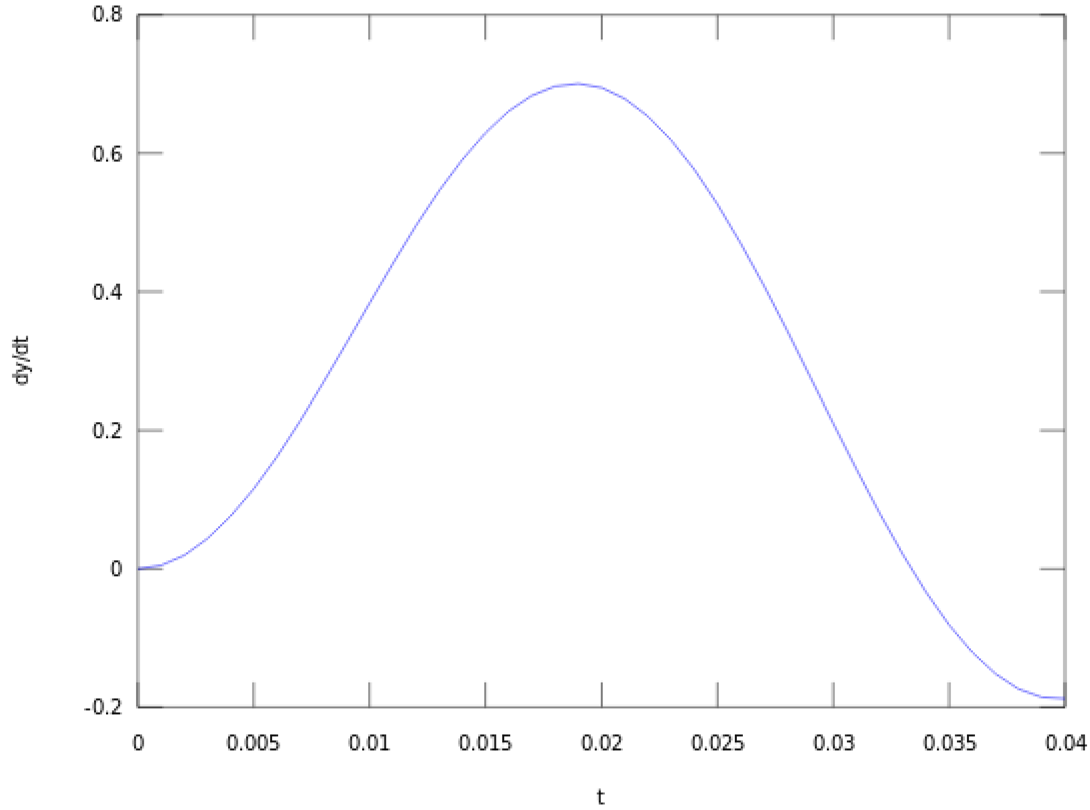

Рисунок 7 – График зависимости скорости от времени.

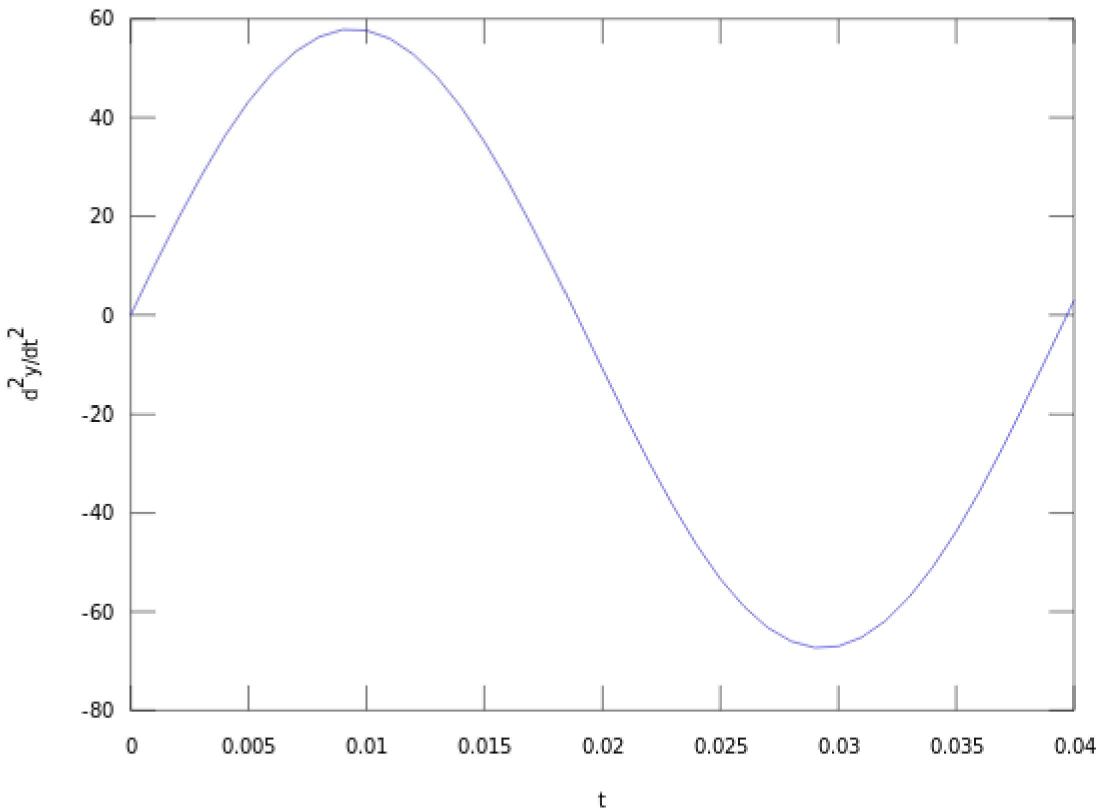

Рисунок 8 – График зависимости ускорения от времени.

2. Измените массу автомобиля – увеличьте до $980\,кг$. Как изменилась амплитуда вертикального перемещения? Скорость? Ускорение?

Строку 6, кода программы, приведем к виду следующему виду

6	global M = 980;

В итоге получим:

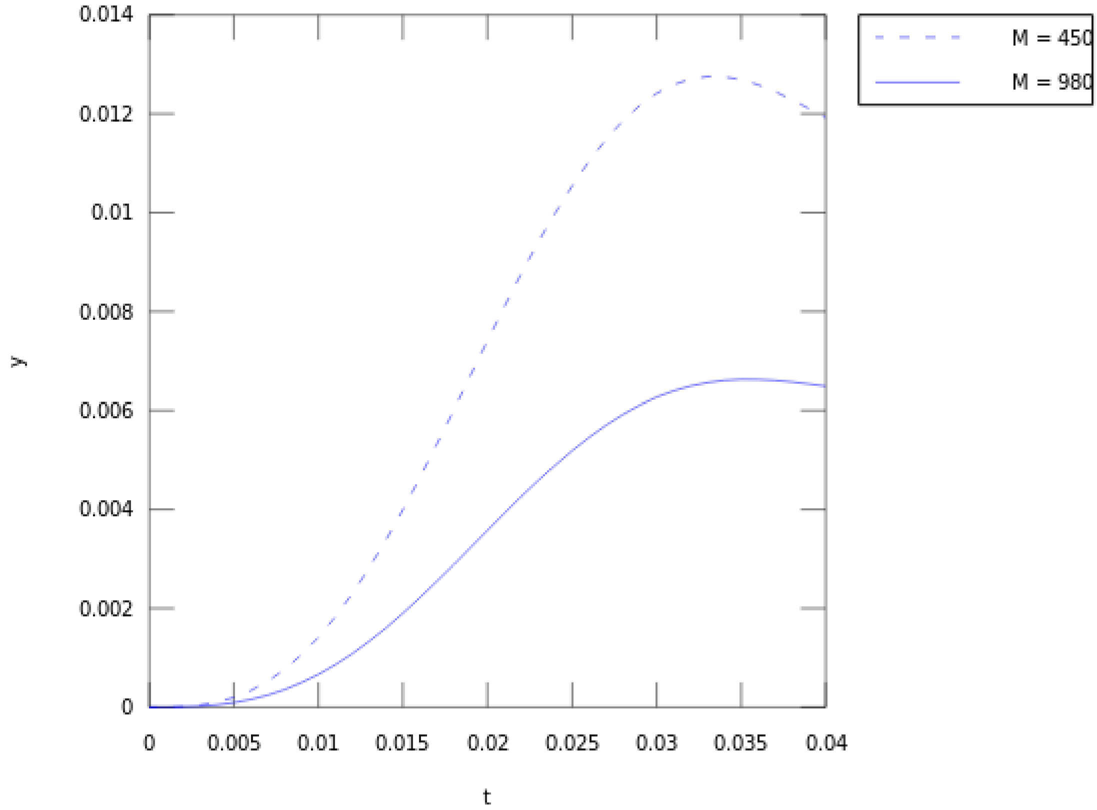

Рисунок 9 – График зависимости перемещения от времени при $M = 980$, в сравнении с аналогичным графиком при $M = 450$.

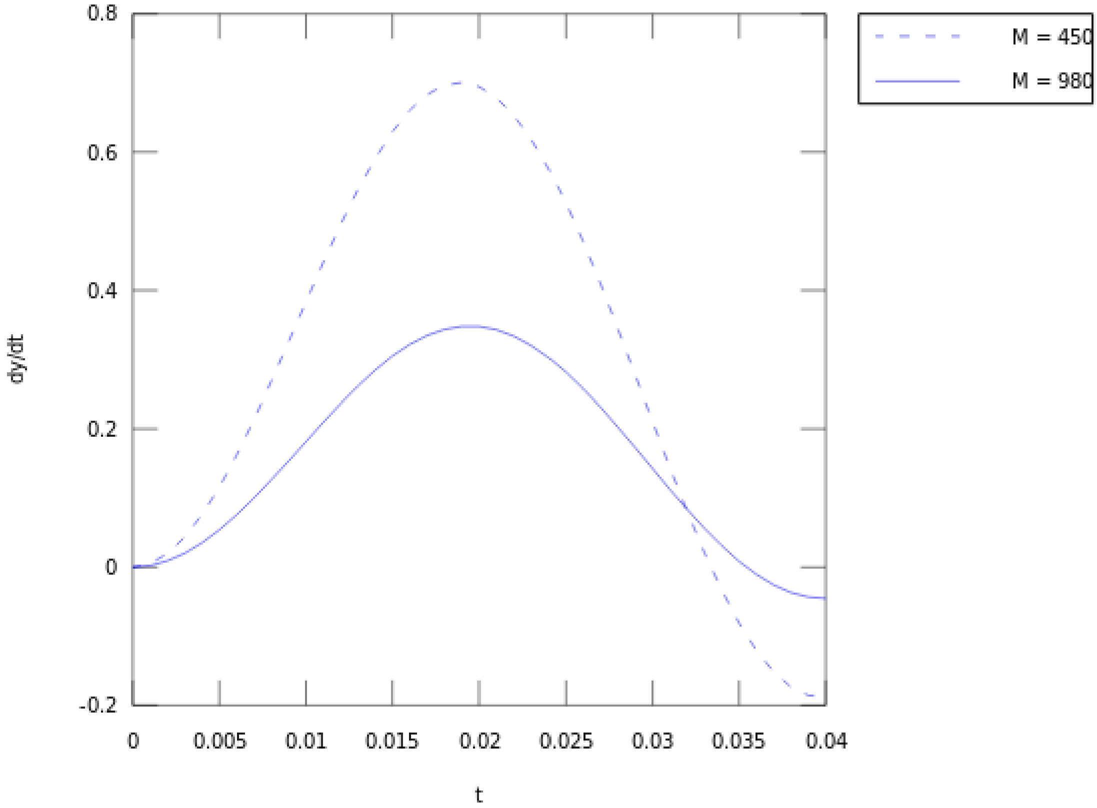

Рисунок 10 – График зависимости скорости от времени при $M = 980$, в сравнении с аналогичным графиком при $M = 450$.

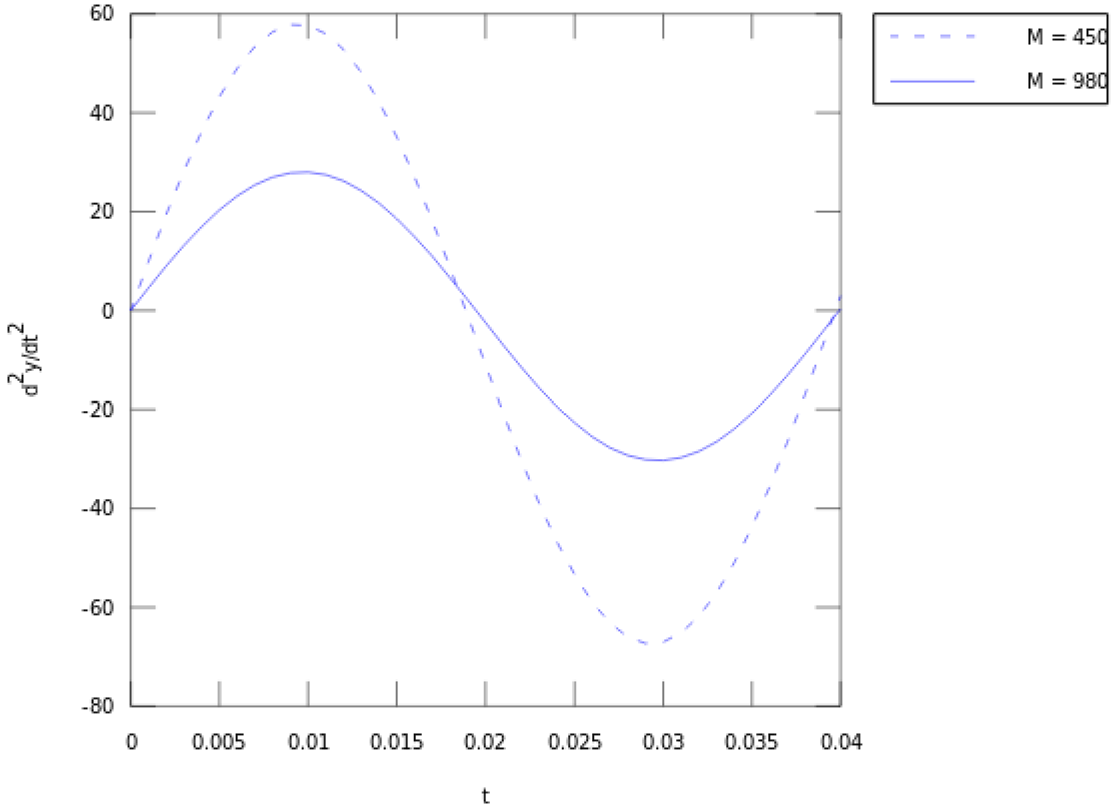

Рисунок 11 – График зависимости ускорения от времени при $M = 980$, в сравнении с аналогичным графиком при $M = 450$.

Из графиков видно, что автомобиль с большей массой обладает большей инертностью, и поэтому гораздо медленнее набирает вертикальное перемещение, скорость и ускорение [6].

3. Уменьшите жесткость (упругость) подвески до $2000\, H/м$. Сопоставьте графики перемещения, скорости, ускорения.

   Приведем строку 6 кода программы к первоначальному виду

6	global M = 450;

и изменим строку 7. Она должна иметь вид:

7	global K = 2000;

После запуска измененной программы, получим следующие графики:

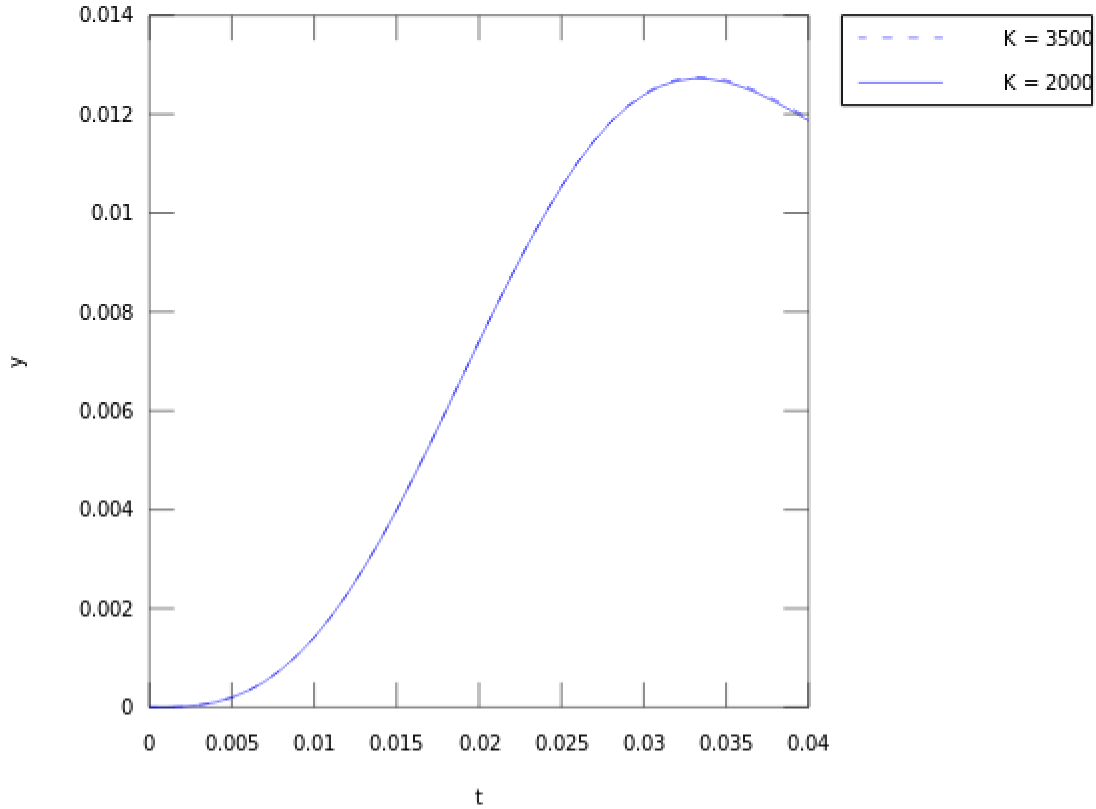

Рисунок 12 – График зависимости перемещения от времени при $K = 2000$, в сравнении с аналогичным графиком при $K = 3500$.

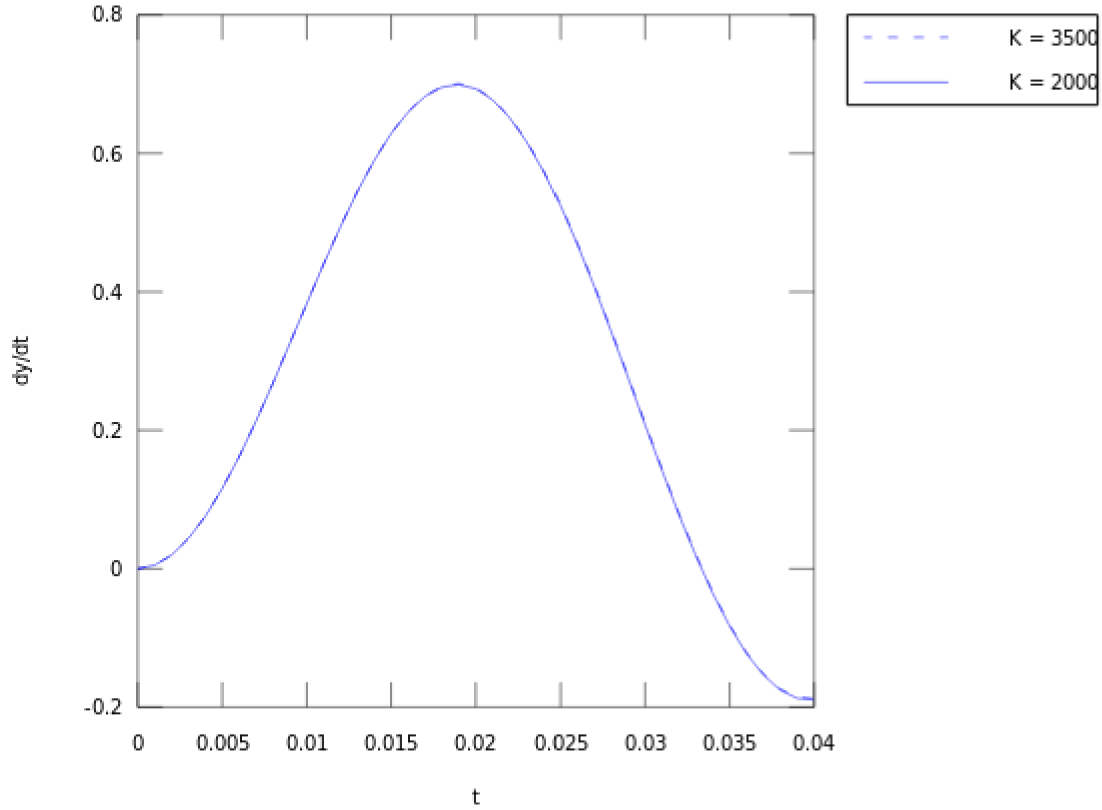

Рисунок 13 – График зависимости скорости от времени при $K = 2000$, в сравнении с аналогичным графиком при $K = 3500$.

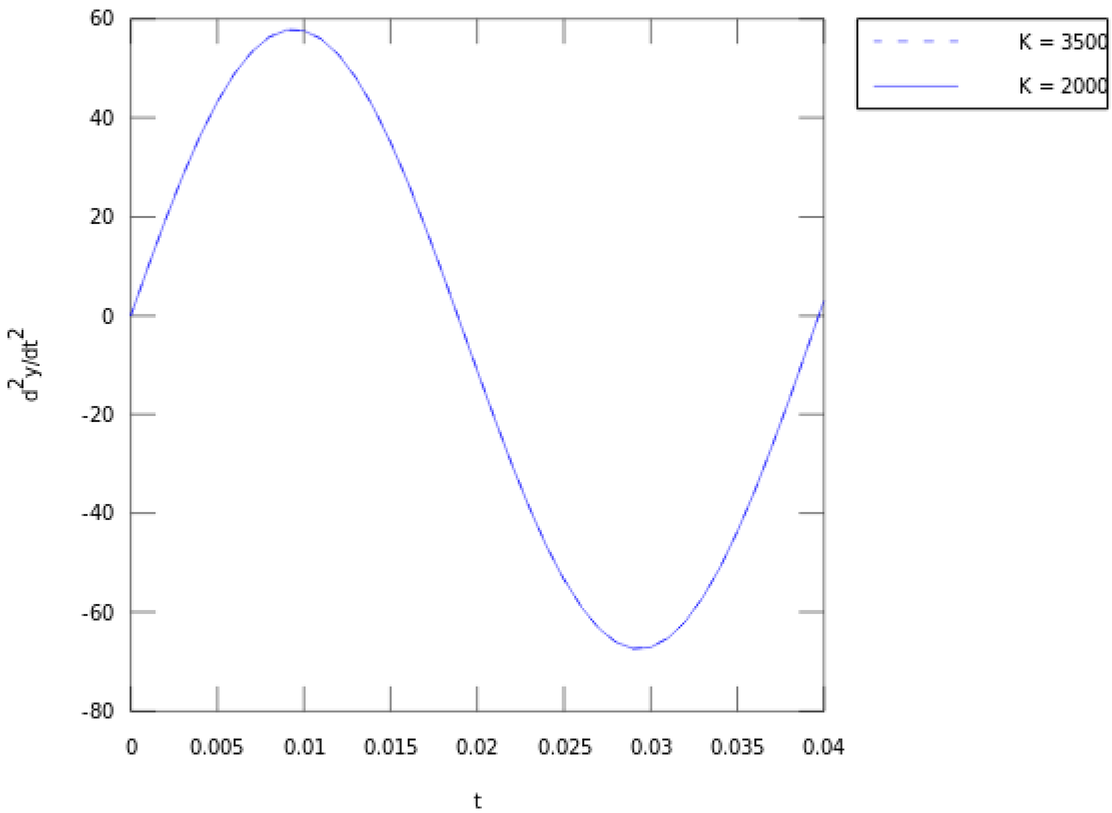

Рисунок 14 – График зависимости ускорения от времени при $K = 2000$, в сравнении с аналогичным графиком при $K = 3500$.

Анализируя графики, приходим к выводу, что уменьшение жесткости пружины приводит к уменьшению высоты, скорости и ускорения за счет того, что пружина с меньшей жесткостью оказывает меньшее давление на демпфер.

4. Уменьшите постоянную демпфирования в 3 раза, до нуля. Как изменился процесс при движении по препятствию? После?

Приведем программу к первоначальному виду. После, изменим строчку под номером 8 , так, что выйдет:

8	global B = 7300 / 3;

или

8	global B = 0;

После применения любого из приведенных выше изменений получим три графика. Сравнение графиков при разных значениях B приведено на следующих трех рисунках:

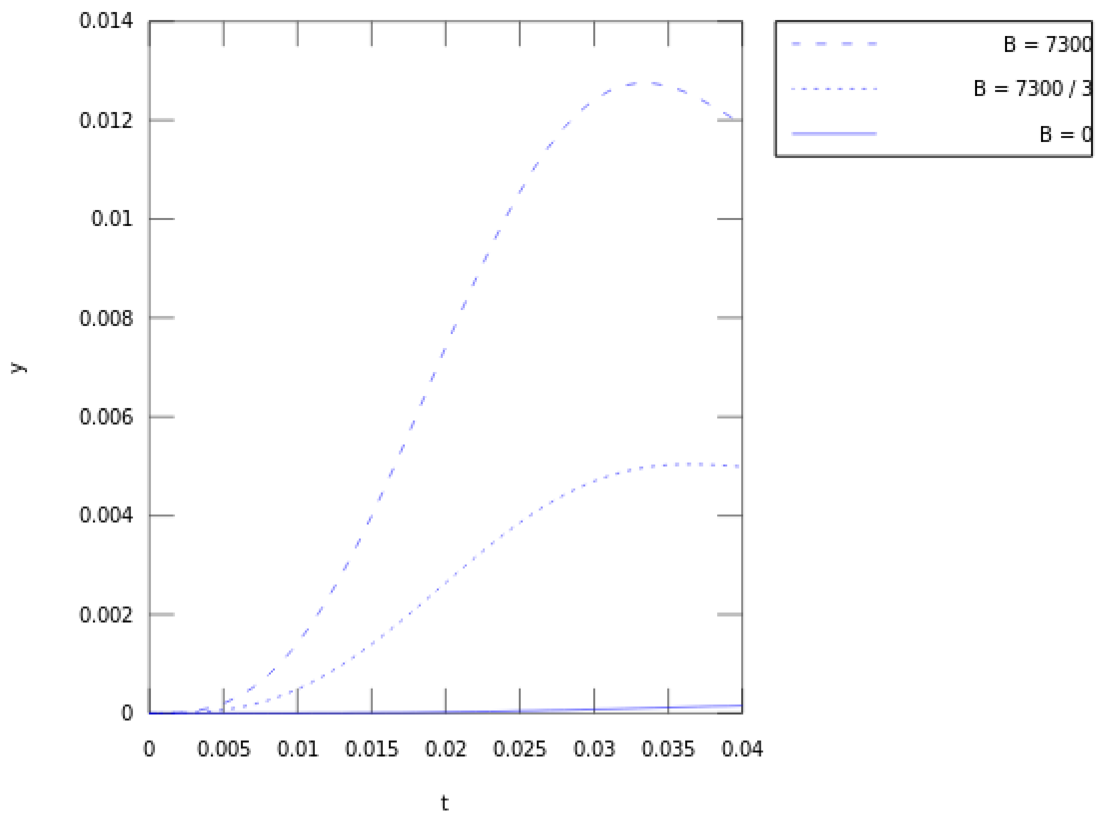

Рисунок 15 – Графики зависимости перемещения от времени при $B=7300$, $B=7300/3$, $B=0$.

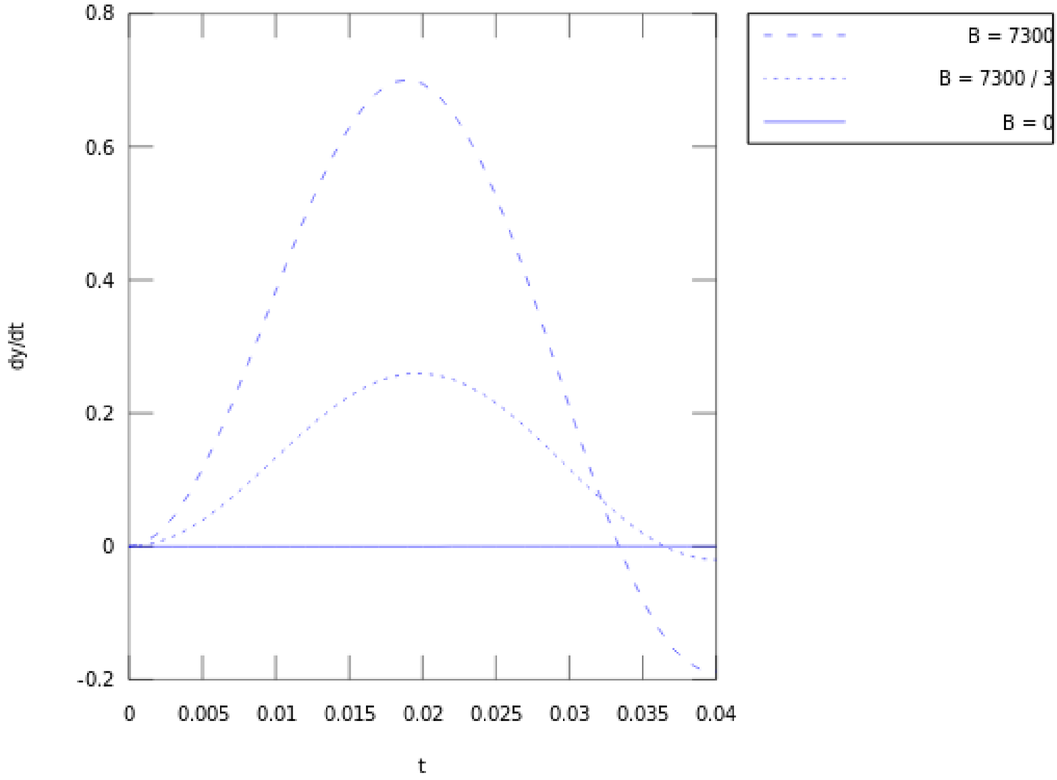

Рисунок 16 – Графики зависимости скорости от времени при $B=7300$, $B=7300/3$, $B=0$.

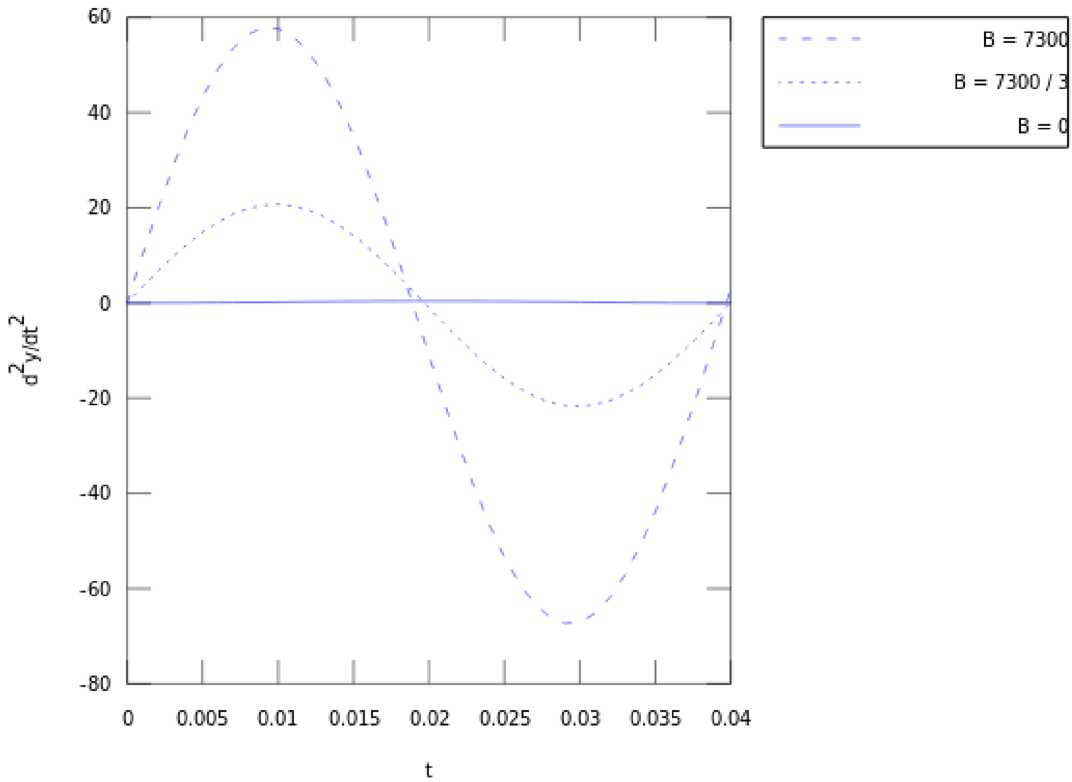

Рисунок 17 – Графики зависимости скорости от времени при $B=7300$, $B=7300/3$, $B=0$.

Из рисунков 14, 15, 16 можно сделать вывод, что перемещение, скорость и ускорение прямо пропорциональны постоянной демпфирования.

5. Особое внимание уделите графику и абсолютному значению ускорения при проезде препятствия. Превышение $5g$ ($g=9,8\ м/с^2$) становится чрезвычайно опасным для жизни.

Действительно, некоторые из выше описанных изменений параметров математической модели приводят к превышению $5g$.

6. Измените параметры препятствия. Рассмотрите варианты узкого и высокого препятствия, широкого и низкого (высота и ширина: $0{,}1\,м$ и $0{,}4\,м$, $0{,}04\,м$ и $1\,м$). Сделайте выводы об условиях безопасного проезда «спящего полицейского».

Изменим строчки 3 и 4

3	global H = 0.05;
4	global L = 0.8;

Вместо старых значений H и L подставим по очереди, соответственно $0{,}1\,м$ и $0{,}4\,м$, $0{,}04\,м$ и $1\,м$.

Полученные графики приведены ниже:

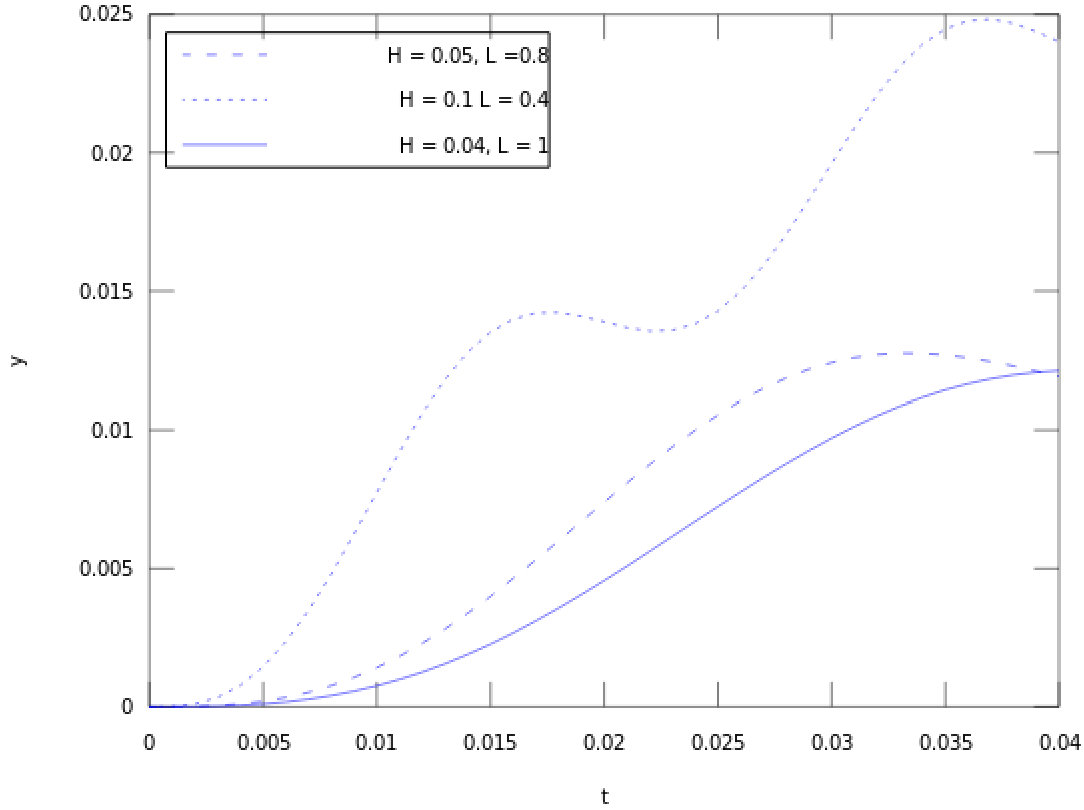

Рисунок 18 – Графики зависимости перемещения от времени при разных видах препятствия.

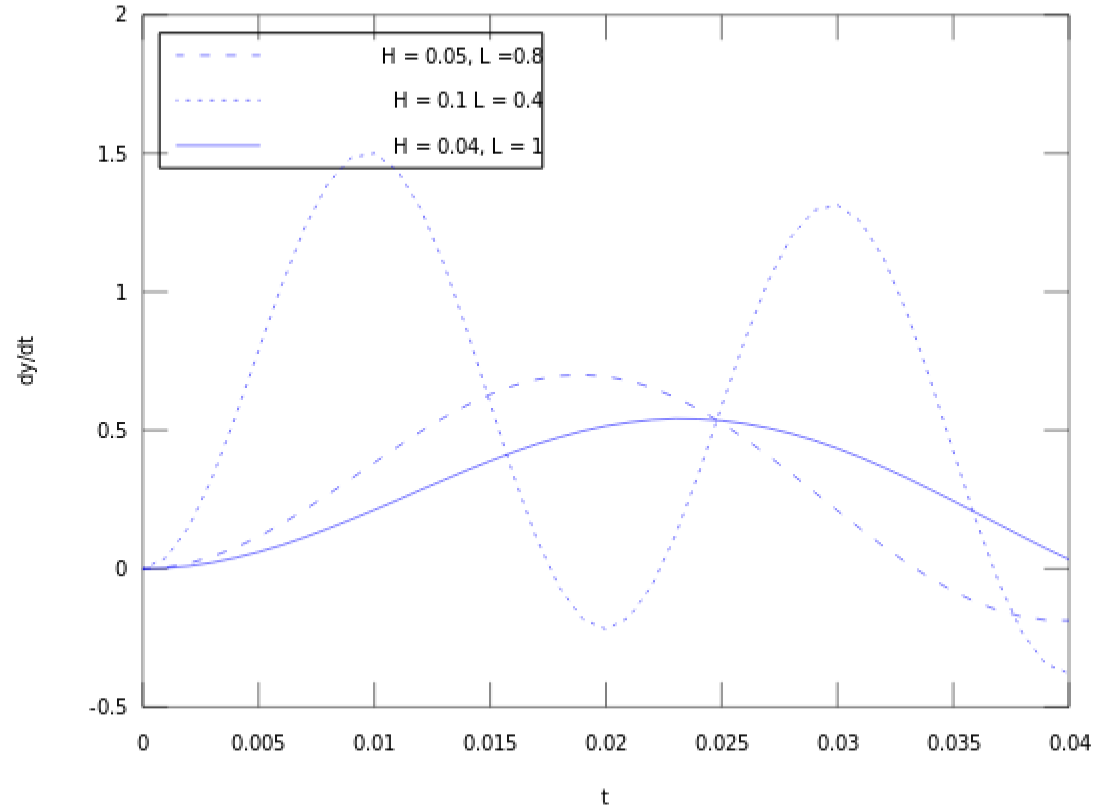

Рисунок 19 – Графики зависимости скорости от времени при разных видах препятствия.

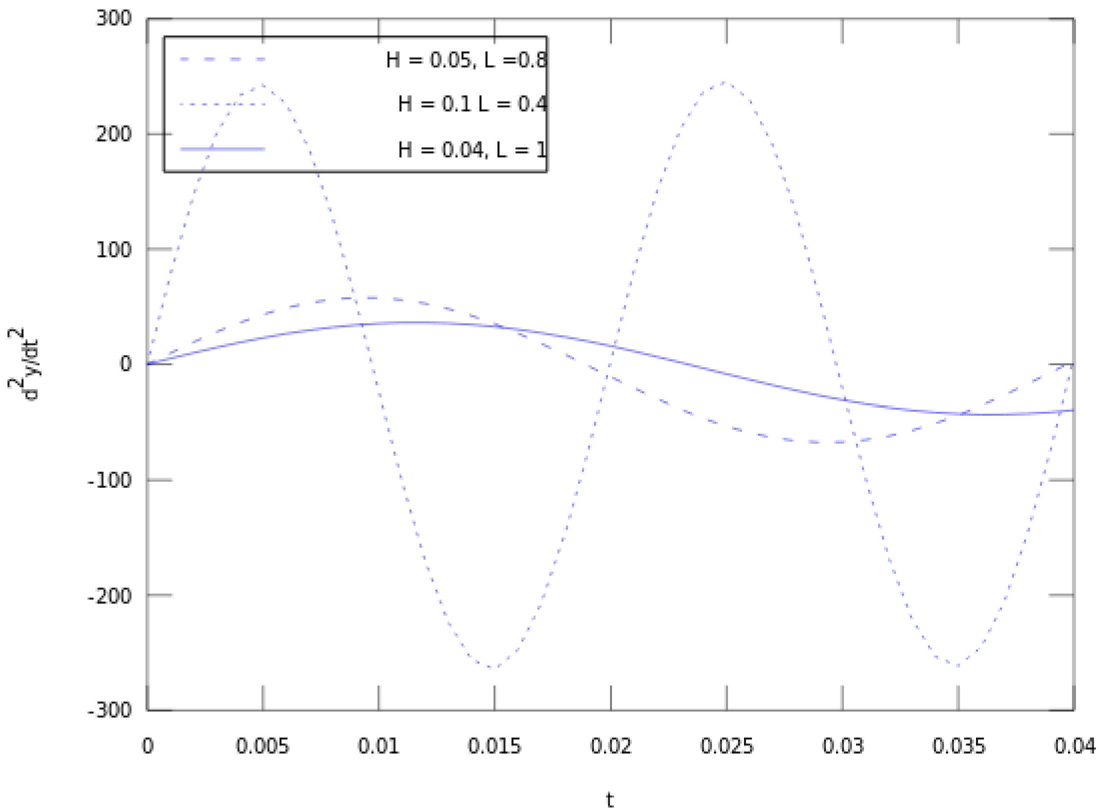

Рисунок 20 – Графики зависимости ускорения от времени при разных видах препятствия.

2.2 Стохастические системы. Планирование в атомной энергетике

Формулировка задачи

Решается вопрос , как много атомных станций строить в стране каждые пять лет. Стоимость строительства атомной станции составляет $20 \cdot 10^6$ денежных ед., а обеспечение условий работы и содержание станции в течение пяти лет обходятся в $5 \cdot 10^6$ ед . Чтобы построить атомную станцию, требуется пять лет. Если станция работает в течение пятилетнего периода , то к началу следующего периода она будет работать только с вероятностью $0{,}6$. Лоббисты по охране окружающей среды провели законопроект о том , что в пятилетний период в стране может быть построено и/или работать не более трех атомных станций. Производительности одной атомной станции достаточно, чтобы удовлетворить потребности страны в энергии, иначе обеспечение в течение пяти лет альтернативных ресурсов энергии будет стоить $60 \cdot 10^6$ ед. Пусть состояния: 1 – нет работающих атомных станций; 2 – одна работающая атомная станция; 3 – две работающие атомные станции; 4 – три работающие атомные станции; управления 0 , 1 , 2 , 3 – число строящихся атомных станций. Определите оптимальную стратегию строительства атомных станций в стране для последующего периода времени.

Математическая модель

Данная система описывается марковским процессом принятия решений (управляемым марковским процессом) с множеством состояний $I = \{1, 2, 3, 4\}$ и множествами решений (управлений) $K(1) = \{0, 1, 2, 3\}$, $K(2) = \{0, 1, 2\}$, $K(3) = \{0, 1\}$, $K(4) = \{0\}$. Обозначим : p_{ij}^k, $i, j \in I$, $k \in K(i)$, – вероятность перехода процесса из состояния i в состояние j при управлении k ; q_i^k – ожидаемые затраты в состоянии i при управлении k .

Полные ожидаемые затраты $\nu_i(n)$ за n последующих шагов, если в начальный момент марковский процесс с доходами находится в состоянии i, определяется рекуррентным соотношением:

$$\nu(n)=q+P\,\nu(n-1),\ n=1,2,3,\ldots,$$

где $\nu_j(0)=0,\ \ j\in I$.

Для управляемого марковского процесса обозначим $\nu_i^*(n),\ j\in I,\ n=1,2,3,\ldots,$ – оптимальный полный ожидаемый доход за n шагов при начальном состоянии i. Согласно рекуррентному методу

$$\nu_i^*(n)=\max_{k\in K(i)}[q_i^k+\sum_{j\in I}p_{ij}^k\nu_j^*(n-1)],$$

где $\nu_j^*(0)=0$, $j\in I$.

Обозначим $\delta_i^*(n)$, $j\in I$, $n=1,2,3,\ldots,$ – оптимальное управление в состоянии i при n оставшихся шагах,

$$\delta_i^*(n)=arg\max_{k\in K(i)}[q_i^k+\sum_{j\in I}p_{ij}^k v_j^*(n-1)]$$

Метод полного перебора

Шаг 1. Построить множество Δ – множество всех возможных векторов управления $\delta=(\delta_i)$, $i\in I$, $\delta_i\in K(i)$.

Шаг 2. Для всех $\delta\in\Delta$ вычислить

$$g^\delta=\pi^\delta q^\delta$$

где π^δ и q^δ – вектор-строка предельных вероятностей и вектор-столбец ожидаемых доходов, соответствующие вектору управления δ.

Шаг 3. Определить

$$g^*=\max_{\delta\in\Delta}g^\delta,$$

$$\delta^*=arg\max_{\delta\in\Delta}g^\delta.$$

Задания.

1) Оформите исходные данные для задачи в виде таблицы вероятностей и доходов

i	k	p_{i1}^k	p_{i2}^k	p_{i3}^k	p_{i4}^k	q_i^k
1	0	1	0	0	0	-60
	1	0	1	0	0	-80
	2	0	0	1	0	-100
	3	0	0	0	1	-120
2	0	0.4	0.6	0	0	-5
	1	0	0.4	0.6	0	-25
	2	0	0	0.4	0.6	-45
3	0	0.16	0.48	0.36	0	-10
	1	0	0.16	0.48	0.36	-30
4	0	0.064	0.288	0.432	0.216	-15

Таблица 9 – исходные данные для задачи.

2) Предположим, что в пятилетний период в стране может быть построена только одна атомная станция. Чему равны в этом случае полные ожидаемые затраты за 10 лет, 20 лет, 30 лет, если в начальный момент времени в стране не было работающих атомных станций? была одна работающая атомная станция?

Код программы:

```
P = [0 1 0 0;
   0 0.4 0.6 0;
   0 0.16 0.48 0.36;
   0.064 0.288 0.432 0.216];
q = [-80; -25; -30; -15];

N = 1;
while(rem(N, 5) != 0)
    N = input ("Input the number of years, multiple of 5: ")
endwhile
N = N / 5;

vAns = zeros(N, 4);
v = [q(1); q(2); q(3); q(4)];
T = zeros(N, 1);

for i = 1 : N

    T(i) = 5 * i;
```

```
    vAns(i, 1) = v(1);
    vAns(i, 2) = v(2);
    vAns(i, 3) = v(3);
    vAns(i, 4) = v(4);
    disp(sprintf('( %g: %g; %g; %g; %g;) ' , 5 * i , v(1), v(2), v(3), v(4)))
    v = q + P * v;
end

plot(T, vAns(:, 1), '.');
hold on;
plot(T, vAns(:, 1), 'g');

plot(T, vAns(:, 2), '.');
hold on;
plot(T, vAns(:, 2), 'm');

plot(T, vAns(:, 3), '.');
hold on;
plot(T, vAns(:, 3), 'b');

plot(T, vAns(:, 4), '.');
hold on;
plot(T, vAns(:, 4), 'c');

xlabel('t, years');
ylabel('v(n)');
```

Результат работы программы:

```
N =  30
( 5: -80; -25; -30; -15;)
( 10: -105; -53; -53.8; -43.52;)
( 15: -133; -78.48; -79.9712; -69.6259;)
( 20: -158.48; -104.375; -106.008; -95.701;)
( 25: -184.375; -130.355; -132.036; -121.67;)
( 30: -210.355; - 156.364; -158.035; -147.663;)
```

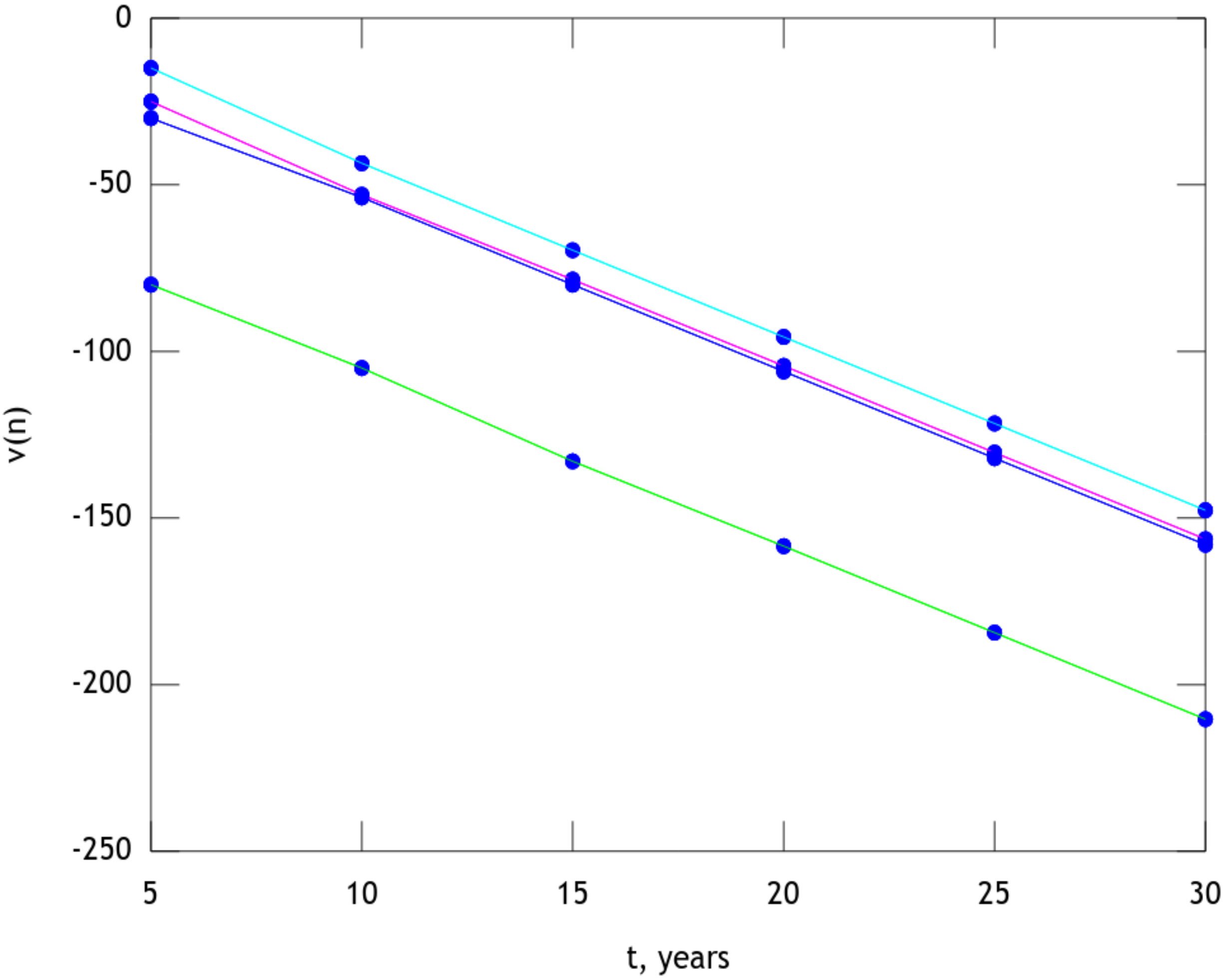

Рисунок 21 – Графики зависимости полных ожидаемых затрат от времени.

На этом графике зеленым, фиолетовым, синим и голубым нарисованы, графики зависимости полных ожидаемых затрат от времени, при начальных состояниях, 1, 2, 3, 4, соответственно.

Как видно из рисунка 21 и результатов, затраты на содержание и постройку атомных станций постоянно растут. Если в начальный момент в стране не было работающих атомных станций, то к 30 году общие ожидаемые затраты составят 210.355. В случае, если в начальный момент в стране была одна работающая атомная станция, к 30 году общие ожидаемые затраты составят 156.364 [8, 10].

3) В случае выполнения предположения задания 2 каковы предельные вероятности того , что в стране работают 2 атомные станции , 3 атомные станции ?

Код программы:

```
P = [0 1 0 0;
   0 0.4 0.6 0;
   0 0.16 0.48 0.36;
   0.064 0.288 0.432 0.216];
P = P - eye(4, 4);
P.';
P(4, :) = ones(1, 4);

B = zeros (4, 1);
B(4) = 1;

pi = P \ B
```

Результат работы программы:

```
pi =


   0.014515
   0.264767
   0.493918
   0.226799
```

4) Предположим, что в стране в пятилетний период могут быть построены не более чем 3 атомные станции. Определите оптимальные полные ожидаемые затраты за 10 лет, 20 лет, 30 лет при условии, что в начальный момент времени в стране не было работающих атомных станций. Какова при этом оптимальная стратегия строительства атомных станций в стране в указанные периоды времени? Постройте графики зависимости затрат от числа пятилеток для всех начальных состояний атомной энергетики страны.

Код программы:

```
P1 = [1 0 0 0;
      0 1 0 0;
      0 0 1 0;
      0 0 0 1];
P2 = [0.4 0.6 0 0;
      0 0.4 0.6 0;
      0 0 0.4 0.6];
P3 = [0.16 0.48 0.36 0;
      0 0.16 0.48 0.36];
P4 = [0.064 0.288 0.432 0.216];

q1 = [-60; -80; -100; -120];
q2 = [-5; -25; -45];
q3 = [-10 -30];
q4 = [-15];

N = 1;
while(rem(N, 5) != 0)
   N = input ("Input the number of years, multiple of 5: ")
endwhile
N = N / 5;

vGraph = zeros(N, 4);
T = zeros(N, 1);
[vopt(1), I(1)] = max(q1);
[vopt(2), I(2)] = max(q2);
[vopt(3), I(3)] = max(q3);
[vopt(4), I(4)] = max(q4);

startup = -1;
while(startup != 1 && startup != 2 && startup != 3 && startup != 4)
```

```
    startup = input ("Input the startup state [1, 2, 3, 4] : ")
endwhile

for years = 1 : N

    vGraph(years, 1) = vopt(1);
    vGraph(years, 2) = vopt(2);
    vGraph(years, 3) = vopt(3);
    vGraph(years, 4) = vopt(4);
    T(years) = 5 * years;

    switch(startup)
    case 1
          disp(sprintf('Год %g Управление %g Затраты %g' , 5 * years , I(1), vopt(1)))
    case 2
          disp(sprintf('Год %g Управление %g Затраты %g' , 5 * years , I(2), vopt(2)))
    case 3
          disp(sprintf('Год %g Управление %g Затраты %g' , 5 * years , I(3), vopt(3)))
    case 4
          disp(sprintf('Год %g Управление %g Затраты %g' , 5 * years , I(4), vopt(4)))
    endswitch
    for i = 1 : 4
        switch(I(i))
            case 1
                [vopt(i), I(i)] = max( [ ( q1(1) + P1(1, 1) * vopt(1) ),
                                     ( q1(2) + P1(2, 2) * vopt(2) ),
                                     ( q1(3) + P1(3, 3) * vopt(3) ),
                                     ( q1(4) + P1(4, 4) * vopt(4) ) ] );
            case 2
                  [vopt(i), I(i)] = max( [ ( q2(1) + ( P2(1, 1) * vopt(1) + P2(1, 2) * vopt(2) ) ),
                                     ( q2(2) + ( P2(2, 2) * vopt(2) + P2(2, 3) * vopt(3) ) ),
                                     ( q2(3) + ( P2(3, 3) * vopt(3) + P2(3, 4) * vopt(4) ) ) ] );
            case 3
                  [vopt(i), I(i)] = max( [ ( q3(1) + ( P3(1, 1) * vopt(1) + P3(1,2) * vopt(2) + P3(1, 3) * vopt(3) ) ),
                                       ( q3(2) + ( P3(2, 2) * vopt(2) + P3(2, 3) * vopt(3) + P3(2, 4) * vopt(4) ) ) ] );
            case 4
                 [vopt(i), I(i)] = max(q4(1) + (P4(1) * vopt(1) + P4(2) * vopt(2) +
```

```
P4(3) * vopt(3) + P4(4) * vopt(4)));
        endswitch

    end

end

plot(T, vGraph(:, 1), '.');
hold on;
plot(T, vGraph(:, 1), 'g');

plot(T, vGraph(:, 2), '.');
hold on;
plot(T, vGraph(:, 2), 'm');

plot(T, vGraph(:, 3), '.');
hold on;
plot(T, vGraph(:, 3), 'b');

plot(T, vGraph(:, 4), '.');
hold on;
plot(T, vGraph(:, 4), 'c');

xlabel('t, years');
ylabel('v(n)');
```

Результат работы программы:

В зависимости от введенных данных, можно получить оптимальные стратегии для начальных состояний $I = 1, 2, 3, 4$. Ответ приведен для начального состояния $I = 1$ – нет работающих атомных станций [7, 9].

```
N =  30
startup =  1
Год 5 Управление 1 Затраты -60
Год 10 Управление 2 Затраты -85
Год 15 Управление 1 Затраты -90
Год 20 Управление 1 Затраты -150
Год 25 Управление 1 Затраты -210
Год 30 Управление 1 Затраты -270
```

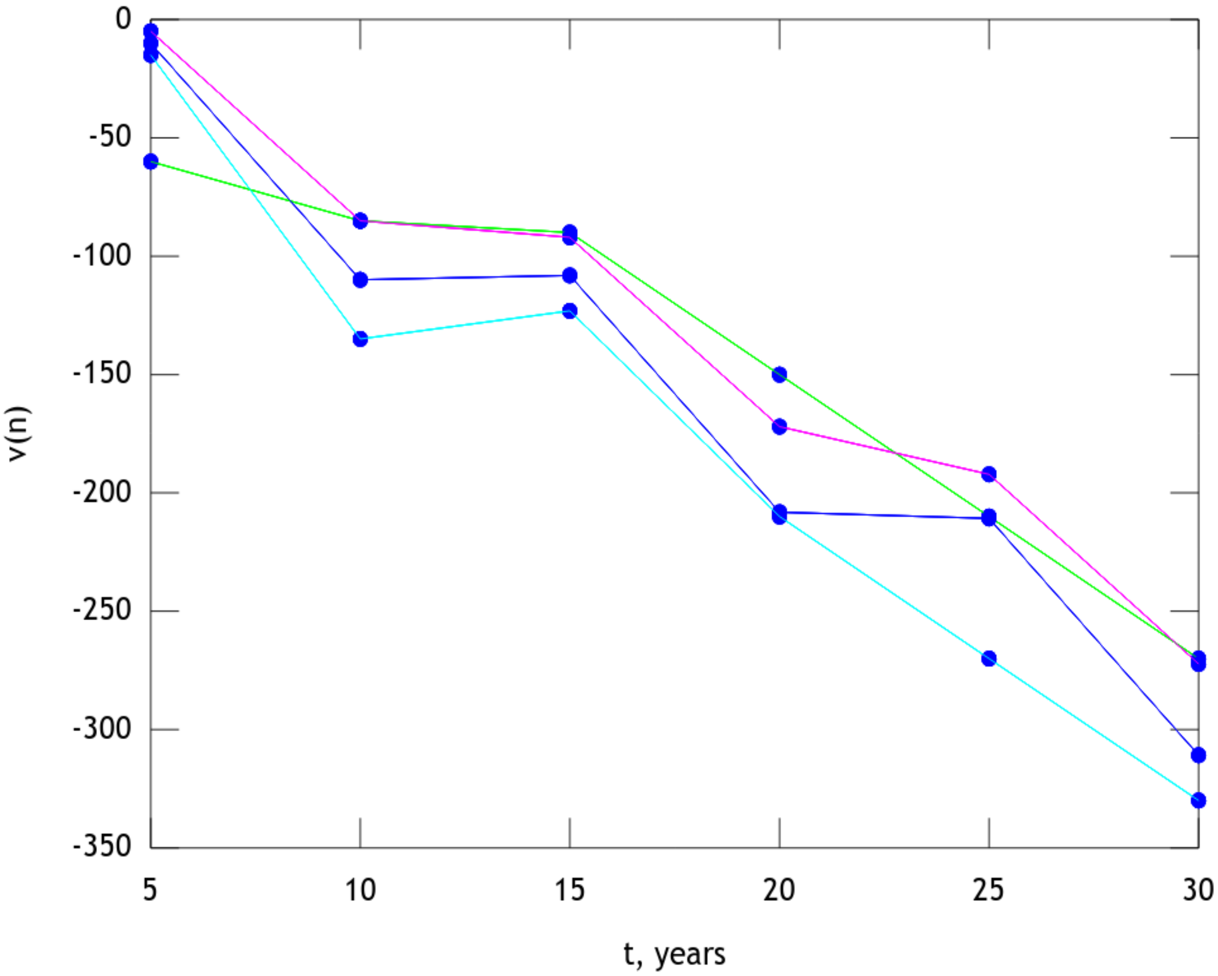

Рисунок 22 – Графики зависимости полных ожидаемых затрат от времени.

На этом графике зеленым, фиолетовым, синим и голубым нарисованы, графики зависимости оптимальных полных ожидаемых затрат от времени, при начальных состояниях, 1, 2, 3, 4, соответственно.

5) Определите оптимальную стационарную стратегию строительства атомных станций в стране при длительном плановом периоде времени и оптимальные средние затраты за пятилетний период в этом случае.

Код программы:

```
Pi1k = [1 0 0 0;
     0 1 0 0;
     0 0 1 0;
     0 0 0 1];
Pi2k = [0.4 0.6 0 0;
     0 0.4 0.6 0;
```

```
     0 0 0.4 0.6];
Pi3k = [0.16 0.48 0.36 0;
     0 0.16 0.48 0.36];
Pi4k = [0.064 0.288 0.432 0.216];

qi1k = [-60; -80; -100; -120];
qi2k = [-5; -25; -45];
qi3k = [-10; -30];
qi4k = [-15];
S = zeros(4, 4);
p=1;
ES=zeros(24,9);
for z = 1 : 1
    for j = 1 : 2
        for k = 1 : 3
            for m = 1 : 4
                S(1,:) = Pi1k(m,:);
                S(2,:) = Pi2k(k,:);
                S(3,:) = Pi3k(j,:);
                S(4,:) = Pi4k(z,:);

                Q(1,1) = qi1k(m,1);
                Q(2,1) = qi2k(k,1);
                Q(3,1) = qi3k(j,1);
                Q(4,1) = qi4k(z, 1);
                I = eye(4);
                S=S'-I;
                S(4,:)=ones(1,4);
                b=[0; 0; 0; 1];
                p0=S\b
                Es(p,5)=0;
                for s=1:4
                    ES(p,s)=p0(s);
                    ES(p,5)=ES(p,5)+p0(s)*Q(s,1);
                end
                ES(p,6) = m;
                ES(p,7) = k;
                ES(p,8) = j;
                ES(p,9) = z;
                p=p+1;
            end
        end
    end
end
ES
```

```
[g,d]=max(ES(:,5));
g
dopt=zeros(1,4);
dopt=[ES(d,6);ES(d,7);ES(d,8); ES(d,9)]
```

Результат работы программы:

```
g = -22.216

dopt =

  2

  2

  1

  1
```

Из результатов видно, что оптимальной стационарной стратегией является постройка одной атомной станции в 1 и 2 состояниях, и отказ от постройки станций в 3 и 4 состояниях. При этом убыток составит $22.216 \cdot 10^6$.

ЗАКЛЮЧЕНИЕ

В ходе учебной практики было исследовано несколько математических моделей и написаны к ним соответствующие программные приложения.

В результате анализа результатов были выявлены закономерности изменения состояний систем.

При прохождении практики, также были развиты навыки:

– программирования в среде разработки Octave;

– аппроксимация функций;

– решение СЛАУ различными методами;

– нахождение минимумов различных функций;

– решение сложных математических задач;

– исследование математических моделей детерменированных и стохастических систем;

– анализ полученных результатов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Алексеев Е.Р., Чеснокова О.В. Решение задач вычислительной математики в пакетах Mathcad 12, MATHLAB 7, Maple 9. Серия: Самоучитель. – М.: НТ Пресс, 2006 – 496 с.

2 Дащенко О.Ф., Кириллов В.Х., Л.В. Коломиец, В.Ф. Оробей. MATLAB в инженерных и научных расчетах: Монография. – Одесса: Астро-принт, 2003. – 214 с.

3 Половко А.М., Бутусов П.Н. MATLAB для студента. – Спб.: БХВ Петербург, 2005. – 320 с.

4 Ануфриев И.Е., Смирнов А. Б., Смирнова Е.Н. Matlab 7. – Спб.: БХВ Петербург, 2005. – 1104 с.

5 Худяков В.Ф., Хабузов В.А. Моделирование источников вторичного электропитания в среде MATLAB 7.x: учебное пособие. – СПб.: ГУАП, 2008. – 332 с.

6 Эндрюс Дж, Мак-Лоун Р. Математическое моделирование / Пер. с англ. – М.: Мир, 1979. – 280 с.

7 Ховард Р.А. Динамическое программирование и марковские процессы/ Пер. с англ. – М.: Сов. радио, 1964. – 189 с.

8 Митрофанов Ю.И. Системный анализ: учебное пособие. - Саратов: Научная книга, 2000. - 232с.

9 Вагнер Г. Основы исследования операций / Пер. с англ. – М.: Мир, 1973, Т. 2, 3. – 488 с., 504 с.

10 Перегудов Ф.И., Тарасенко Ф.П. Введение в системный анализ. М.: Высш. Школа, 1989. – 608 с.