Санкт-Петербургский Политехнический Университет Петра Великого

Институт прикладной математики и механики

**Кафедра прикладной математики**

**Курсовой проект**

**по дисциплине «Введение в информационные технологии»**

Вариант задания №20

Выполнил студент гр. 13631/1 Д. В. Хрипунков

Преподаватель: А. Б. Смирнов

Санкт-Петербург

2018

Оглавление

[*Задание №1*: «График трансцендентной функции одной переменной и полинома». 3](#_Toc532291380)

* [1) Построение графика функции 3](#_Toc532291383)
* [2) Построение графика полинома 3](#_Toc532291384)

[Задание №2: «Графики функции с разрывом и кусочно-заданной функции 11](#_Toc532291385)

* [1) Построение графика функции с разрывом. 11](#_Toc532291387)
* [2) Построение графика «кусочно-заданной» функции.. 11](#_Toc532291388)

[Задание №3: «Построение двух поверхностей второго порядка» 18](#_Toc532291389)

* [1) Построение графиков элиптического параболоида и двуполостного гиперболоида 18](#_Toc532291391)
* [2) Построение графика элиптического параболоида. 18](#_Toc532291392)
* [3) Построение графика двуплостного гиперболоида. 18](#_Toc532291393)

[Задание №4 «Построение взаимного расположения двух поверхностей второго порядка» 24](#_Toc532291394)

* [1) В декартовой системе координат 24](#_Toc532291396)
* [2) Используя параметрическое задание поверхностей 24](#_Toc532291397)

[Задание №5«Исследование свойств полиномов высокого порядка» 29](#_Toc532291398)

* [1) Построение графика полинома 29](#_Toc532291400)
* [2) Построение графика полинома на более узких интервалах. 29](#_Toc532291401)
* [3) Поиск первых 3 производных и их вещественных корней. 29](#_Toc532291402)
* [4) Построение графиков полинома и его производных 29](#_Toc532291403)
* [5) Сравнить координаты точек экстремумов, вычисленных двумя способами 29](#_Toc532291404)
* [6) Внесение случайного изменения в коэффициент полинома 29](#_Toc532291405)

[Задание №6 «Решение задач алгебры». 39](#_Toc532291406)

* [1) По правилу Крамера 39](#_Toc532291408)
* [2) С использованием обратной матрицы 39](#_Toc532291409)
* [3) При помощи операции "\". Сравнение решений. 39](#_Toc532291410)
* [4) Случайное изменение правой части системы. 40](#_Toc532291411)

[Задание №7 «Решение задач средствами символьных вычислений». 44](#_Toc532291415)

* [1) Вычисление производный и пределов. 44](#_Toc532291417)
* [2) Решение кубического уравнения. 44](#_Toc532291418)
* [3) Разложение функции в ряд Тейлора 44](#_Toc532291419)
* [4) Решение случайной системы уравнений 44](#_Toc532291420)

# Задание №1

# Название работы: «График трансцендентной функции одной переменной и полинома».

**Необходимо выполнить:**

# а) Построение графика функции в своих осях с нанесением характерных точек (корней, локальных экстремумов). Построение графика полинома, который имеет коэффициенты [0.9 -4.23 2.07 3.83] с нанесением характерных точек (корней, локальных экстремумов, точек перегибов).

б) Построение графиков каждой функции в четырёх подграфиках без характерных точек, используя разные стили. Требуется, чтобы на графике были маркеры (от 5 до 10) в выбранных точках.

с) Построение графиков всех функций в одних осях (на 4 подграфиках) без характерных точек и с характерными точками без подписей.

**Примечание к работе:**

1) Все графики должны содержать заголовки и подписи осей, подписи характерных точек с выносными линиями.

2) Интервалы построения графиков выбирать с учётом ОДЗ.

**Описание хода работы:**

## 1) Построение графика функции:

* Строится график на промежутке
* С помощью встроенной функции *fzero* находятся корни функции (для данного промежутка корней 5) и выделяются на графике красными кругами
* Находятся минимумы с помощью встроенной функции *fminbnd* (для данного промежутка 3 локальных минимума) и максимумы с помощью встроенной функции *fminbnd* вызванной от заданной функции с отрицательным знаком (для данного промежутка 2 локальных максимума) и выделяются на графике зелеными звездами

## 2) Построение графика полинома:

* Строится график на промежутке
* С помощью встроенной функции *roots* находятся корни полинома (для данного интервала их 3) и выделяются на графике красными кругами
* Экстремумы полинома находятся с помощью встроенной функции нахождения производной *polyder* и функции нахождения корней производной полинома *roots* (для данного промежутка 2 экстремума: 1 минимум и 1 максимум) и выделяются на графике зелеными звездами
* Находится точка перегиба полинома с помощью функции нахождения второй производной (функция polyder применяется к первой производной) и функции нахождения корней roots (для данного интервала один перегиб) и выделяется на графике розовым квадратом

3) Оформление графика в соответствии с требованиями:

* Делаются выносные линии с подписями с помощью встроенных функций *line* и *text*
* Делаются заголовки для графиков с помощью встроенной функции *title*
* Делаются обозначения осей с помощью встроенных функций *xlabel* и *ylabel*

**Код работы:**

1. Построение графика функции
2. В отдельных окнах создаются две собственные функции y = f11 и y = f12:

function y = f11(x)

y = atan(1.7+0.7.\*x).\*cos(2.6+0.3.\*x.\*x);

function y = f12(x)

y = -atan(1.7+0.7.\*x).\*cos(2.6+0.3.\*x.\*x);

1. В новом окне строится график на заданном промежутке, подписываются оси и заглавие:

x = -5 : 0.05 : 5;

y = f11(x);

plot(x, y, 'b')

title('график функции f(x)')

xlabel('x')

ylabel('y')

grid on

hold on

1. В этом же окне ищутся корни функции с помощью встроенной функции fzero:

rx = fzero(@f11, [-5, -3]);

x0 = rx(imag(rx) == 0);

plot(x0, 0, 'ro')

text(rx-0.5, -0.1, 'root');

rx = fzero(@f11, -3);

x0 = rx(imag(rx) == 0);

plot(x0, 0, 'ro')

text(rx-0.2, -0.1, 'roots');

rx = fzero(@f11, -17/7);

x0 = rx(imag(rx) == 0);

plot(x0, 0, 'ro')

rx = fzero(@f11, [-2, 3]);

x0 = rx(imag(rx) == 0);

plot(x0, 0, 'ro')

text(rx-0.6, -0.1, 'root');

rx = fzero(@f11, [3, 5]);

x0 = rx(imag(rx) == 0);

plot(x0, 0, 'ro')

text(rx-0.5, -0.1, 'root');

Корни рисуются на графике красными кругами и подписываются с помощью встроенной функции text

1. Поиск максимума:

[x, y] = fminbnd(@f12, -3, 0);

plot(x, y, 'g\*')

text(x-0.2, y+0.2, 'max');

[x, y] = fminbnd(@f12, 0, 5);

plot(x, -y, 'g\*')

text(x-0.3, -y-0.2, 'max');

Максимумы функции ищутся с помощью встроенной функции fminbnd от заданной функции, взятой со знаком минус, на промежутках [-3, 0] и [0, 5]. Выделяются зелеными звездами, подписываются точки с помощью встроенной функции text

1. Ищем минимумы:

[x, y] = fminbnd(@f11, -5, -3);

plot(x, y, 'g\*')

text(x-0.2, y-0.2, 'min');

[x, y] = fminbnd(@f11, -3, 2);

plot(x, y, 'g\*')

text(x-0.2, y+0.2, 'min');

[x, y] = fminbnd(@f11, 2, 5);

plot(x, y, 'g\*')

text(x-0.6, y, 'min');

Минимумы функции ищутся с помощью встроенной функции fminbnd от заданной функции на промежутках [-5, -3], [-3, -2] и [2, 5]. Выделяются зелеными звездами, подписываются точки с помощью встроенной функции text

1. Построение графика полинома
2. В новом окне задается вектор-строка коэффициентов полинома, область определения построения и строится график:

p = [0.9 -4.23 2.07 3.83];

x = -2 : 0.05 : 5;

y = polyval(p, x);

plot(x, y, 'b')

title('график полинома')

xlabel('x')

ylabel('y')

grid on

hold on

Функциями xlabel и ylabel подписываются оси на графике и функцией title заглавие.

1. Обозначение корней:

r = roots(p);

x0 = r(imag(r) == 0);

plot(x0, 0, 'ro')

line([x0 x0], [0 -4], 'Color', 'b')

text(x0(1)-0.1, -4, 'root')

text(x0(2)-0.1, -4, 'root')

text(x0(3)-0.1, -4, 'root')

Корни полинома вычисляются с помощью встроенной функции roots, отделяются действительные корни и отмечаются на графике красными звездами. Делаются выносные линии и подписи.

1. Нахождение экстремумов:

p1 = polyder(p);

r1 = roots(p1);

x01 = r1(imag(r1) == 0);

y01 = polyval(p, x01);

plot(x01, y01, 'g\*')

text(x01(1)-0.1, -5, 'min')

text(x01(2)-0.1, 5.5, 'max')

С помощью функции polyder находится производная от полинома, функция roots находит ее корни, из которых выделяются только действительные, они и являются точками экстремумов. Они наносятся на график зелеными звездами и подписываются функцией text

1. Перегибы:

p2 = polyder(p1);

r2 = roots(p2);

x02 = r2(imag(r2) == 0);

y02 = polyval(p, x02);

plot(x02, y02, 'ms')

line([x02 x02], [y02 y02+5], 'Color', 'm')

text(x02-0.3, y02+5, 'knuckle point')

С помощью функции polyder находится вторая производная от полинома, функция roots находит ее корни, из которых выделяется только один действительный, он и является точкой перегиба. Наносится на график розовым квадратом и подписывается функцией text

1. Построение графика полинома в виде четырёх подграфиков:
2. График полинома:

p = [0.9 -4.23 2.07 3.83];

x = -2 : 0.05 : 5;

y = polyval(p, x);

xm = -2 : 1 : 5;

ym = polyval(p, xm);

Задаются интервалы для построения графика и интервал для нанесения на него маркеров.

1. Первый подграфик:

subplot(2, 2, 1)

plot(x , y, 'r-')

hold on

plot(xm , ym, 'k+')

xlabel('x')

ylabel('y')

title('График полиноминальной функции P(x)')

grid on

1. Второй подграфик:

subplot(2, 2, 2)

plot(x, y, 'g:')

hold on

plot(xm, ym, 'bo')

title('График полиноминальной функции P(x)')

xlabel('x')

ylabel('y')

grid on

1. Третий подграфик:

subplot(2, 2, 3)

plot(x, y, 'b')

hold on

plot(xm, ym, 'rs')

title('График полиноминальной функции P(x)')

xlabel('x')

ylabel('y')

grid on

1. Четвёртый подграфик:

subplot(2, 2, 4)

plot(x, y, 'b')

hold on

plot(xm, ym, 'm\*')

title('График полиноминальной функции P(x)')

xlabel('x')

ylabel('y')

grid on

1. Построение графиков всех функций в одних осях:
2. Задаются интервалы для построения полинома и графика:

x = -3 : 0.05 : 4;

y = f11(x);

p = [0.9 -4.23 2.07 3.83];

xp = -1 : 0.05 : 4;

yp = polyval(p, xp);

grid on

hold on

Интервал для полинома берется меньше, чем для функции, чтобы соотнести масштабы графиков

1. Подграфик без характерных точек:

subplot(2, 2, 1)

hold on

grid on

plot(xp, yp, 'k')

plot(x, y, 'b')

xlabel('x')

ylabel('y')

title('График полинома и функции')

legend('Полином', 'Функция')

1. Второй подграфик с корнями функции и полинома:

subplot(2, 2, 2)

hold on

grid on

plot(xp, yp, 'k')

plot(x, y, 'b')

rx = fzero(@f11, [-3, 4]);

x0 = rx(imag(rx) == 0);

plot(x0, 0, 'ro')

r = roots(p);

x0 = r(imag(r) == 0);

plot(x0, 0, 'go')

rx = fzero(@f11, -3);

x0 = rx(imag(rx) == 0);

plot(x0, 0, 'ro')

xlabel('x')

ylabel('y')

title('График полинома и функции с корнями')

legend('Полином', 'Функция', 'Корни функции', 'Корни полинома')

Корни полинома ищутся с помощью функции roots, после чего из них выделяются только действительные, которые обозначаются на графике зелеными кругами. Корни функции находятся с помощью функции fzero, выделяются действительные и отмечаются на графике красными кругами

1. Третий подграфик с экстремумами функции и полинома:

subplot(2, 2, 2)

hold on

grid on

plot(xp, yp, 'k')

plot(x, y, 'b')

p1 = polyder(p);

r1 = roots(p1);

x01 = r1(imag(r1) == 0);

y01 = polyval(p, x01);

plot(x01, y01, 'g\*')

[xm, ym] = fminbnd(@f11, -3, 1.5);

plot(xm, ym, 'r\*')

[xm, ym] = fminbnd(@f12, 1.5, 4);

plot(xm, -ym, 'r\*')

xlabel('x')

ylabel('y')

title('График полинома и функции с экстремумами')

legend('Полином', 'Функция', 'Экстремумы полинома', 'Экстремумы функции')

Находится первая производная полинома при помощи функции polyder, отделяются действительные корни и вычисляются значения в этих точках, отмечаются на графике зелеными звездами. Экстремумы функции находятся через функцию fminbnd и отмечаются на графике красными звездами

1. Четвертый подграфик с перегибом полинома:

subplot(2, 2, 4)

hold on

grid on

plot(xp, yp, 'k')

plot(x, y, 'b')

p2 = polyder(p1);

r2 = roots(p2);

x02 = r2(imag(r2) == 0);

y02 = polyval(p, x02);

plot(x02, y02, 'ro')

xlabel('x')

ylabel('y')

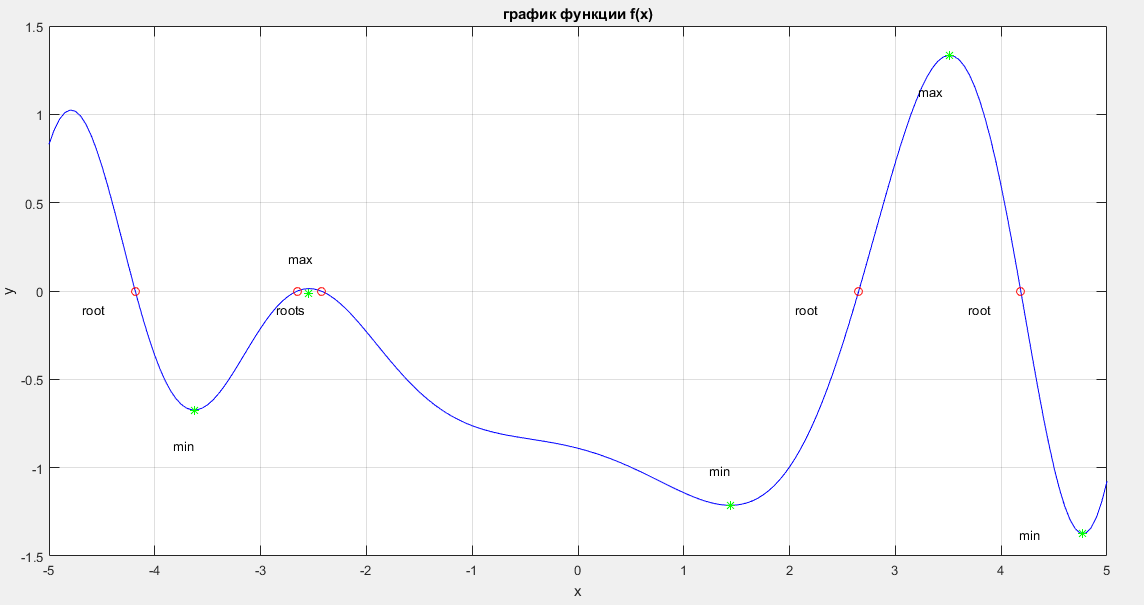
title('График полинома и функции с перегибами')

legend('Полином', 'Функция', 'Перегибы полинома')

Берётся вторая производная полинома, находится ее действительный корень и отмечается на графике красным кругом

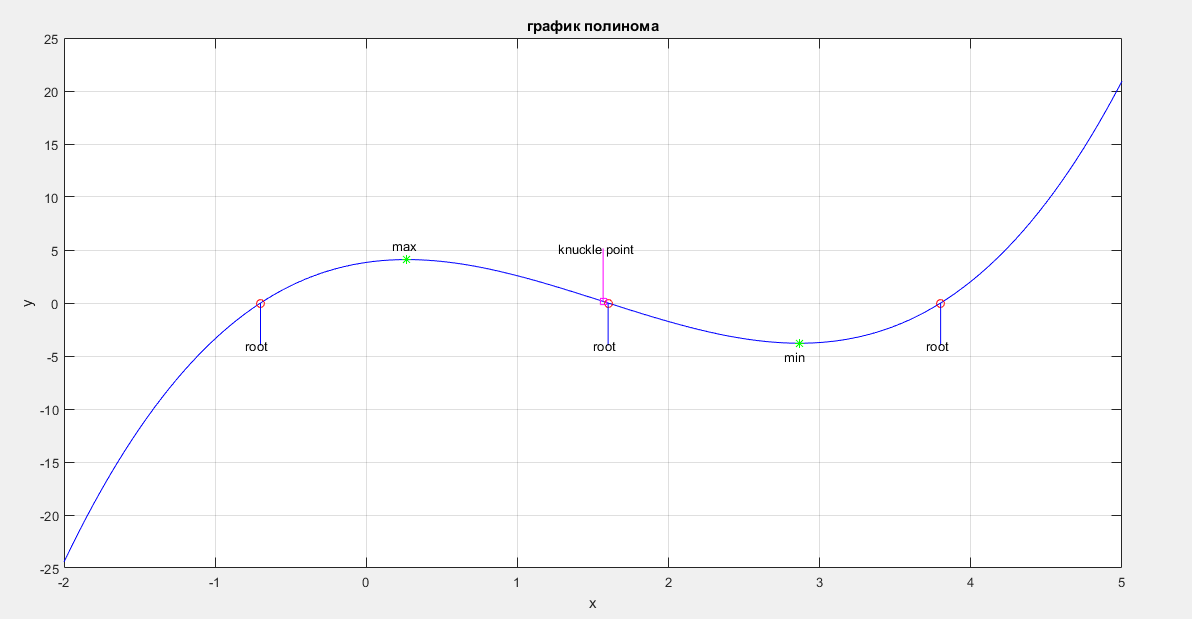
**Полученные в ходе работы результаты:**

1. Построение графика функции с характерными точками



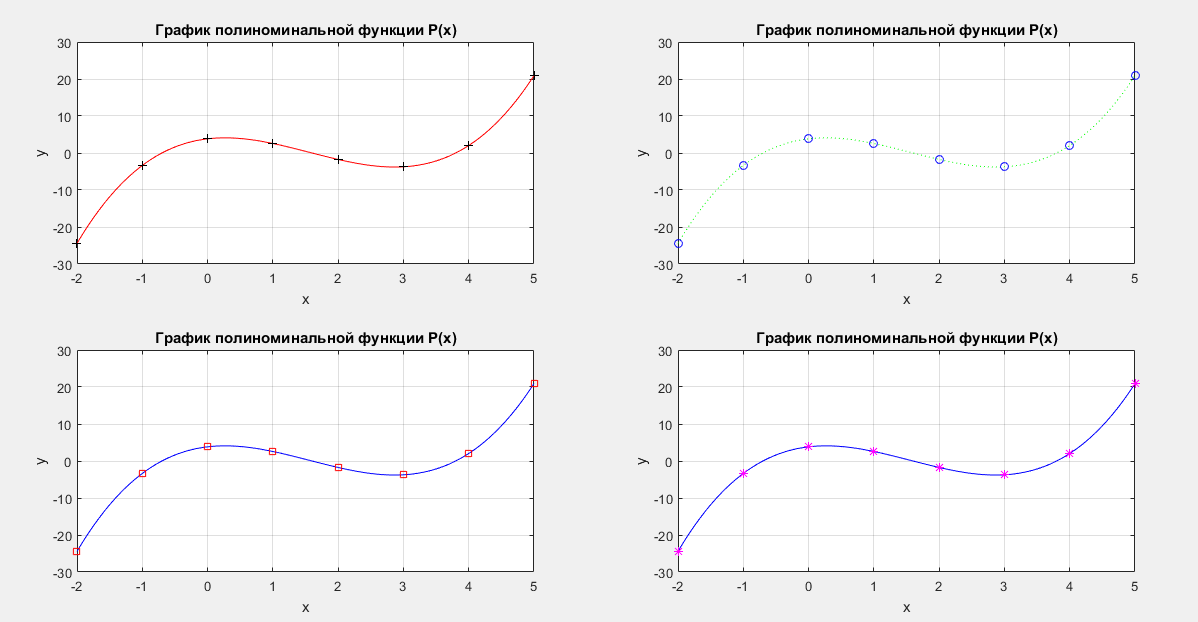
Рисунок

1. Построение графика полинома c характерными точками



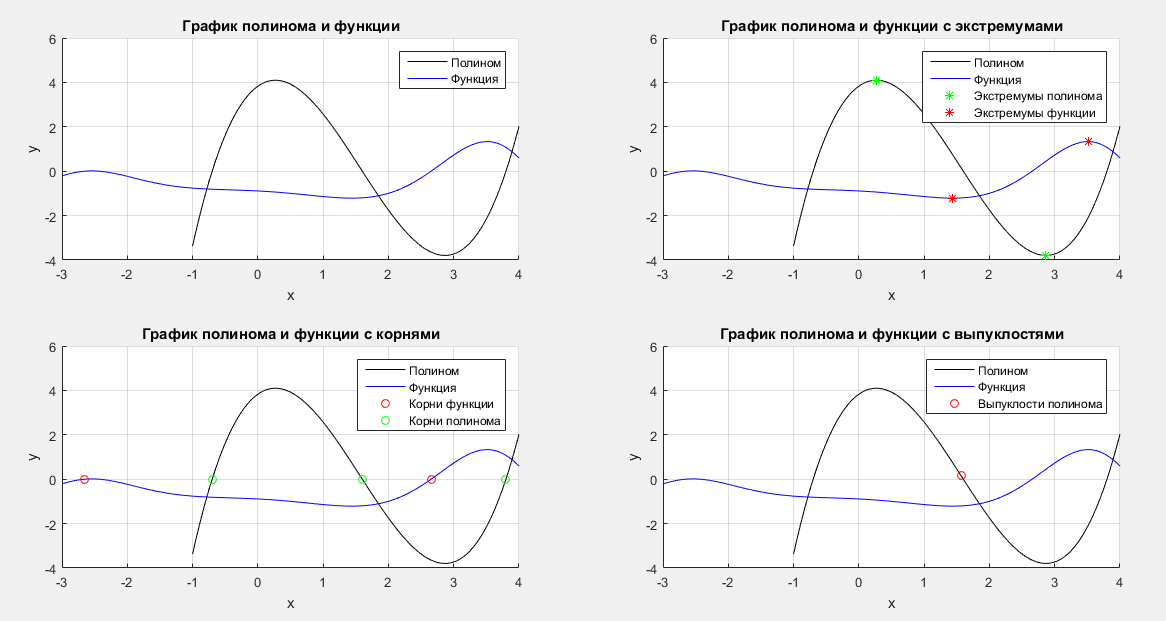
Рисунок

3. Построение графика полинома в виде четырёх подграфиков



Рисунок

4. Построение графиков всех функций в одних осях



Рисунок

# Задание №2

# Название работы: *«Графики функции одной переменной с разрывом и кусочно-заданной функции».*

**Необходимо выполнить:**

## а) Построение графика на интервале [-0.2, 7.8], содержащем корень полинома и нанесение асимптоты в точке разрыва.

б) Сделать «усечение» функции (изобразить только значения -3 ≤ g(x) ≤ 3) по оси ординат.

## с) Построение графика «кусочно-заданной» функции, определённой на трёх интервалах: [-10; 0], [0; 10], [10; 20] как у(х), Р(х), и g(x) соответственно. Подписать каждую часть графика, используя легенду.

**Примечание к работе:**

1. Интервалы [x1; x2], [x2; x3], [x3; x4] подобрать так, чтобы значения функций не сильно отличались, но содержали некоторые характерные точки. Интервалы построения графиков выбирать с учётом ОДЗ.

2) Все графики должны содержать заголовки и подписи осей, подписи характерных точек с выносными линиями.

3) Для «усечения» не использовать функции axis, ylim и аналогичные.

**Описание хода работы:**

1. Построение графика на интервале:

* График функции строится в интервале
* Строится асимптота красной пунктирной линией для графика, проходящая вертикально через координаты **x = 3.8**

1. Построение графика на интервале с усечением:

* График функции строится в интервале
* Строятся три асимптоты красной штриховой линией для графика, проходящие вертикально через координаты **x = -0.7, x = 1.6** и **x = 3.8**
* Получившийся график усекается по оси ординат (изображаются только значения -3 ≤ g(x) ≤ 3)

1. Построение графика «кусочно-заданной» функции:

* График функции строится в интервале сплошной зеленой линией
* График полинома строится в интервале синей штрихпунктирной линией. Корни полинома обозначены красной звездой
* График функции строится в интервале черной пунктирной линией. Корень обозначен синим кругом

**Код работы:**

1. Построение графика на интервале

function y = g11(x)

y = atan(1.7+0.7.\*x).\*cos(2.6+0.3.\*x.\*x)./(0.9\*x.^3-4.23\*x.^2+2.07\*x+3.83);

В отдельном окне создается функция g11

p = [0.9 -4.23 2.07 3.83];

r = roots(p);

sort(r);

x1 = r(1)-4: 0.1 : r(1)-0.000000000000001;

x2 = r(1)-0.000000000000001 : 0.1 : r(1)+4;

y1 = g11(x1);

y2 = g11(x2);

subplot(1,2,1)

grid on

hold on

plot(x1,y1,'b');

plot(x2,y2,'b');

line([r(1), r(1)],[-8\*10^13,2\*10^13], 'Linestyle', '--', 'Color', 'r')

xlabel('X');

ylabel('Y');

title('График с одним корнем полинома')

Задается полином, находятся его корни. Относительно положения корня строится график функции, рисуется асимптота, подписываются оси и заглавие

1. Построение графика на интервале с усечением

x = r(1)-10:0.1:r(3)+10;

y = g11(x)

b=abs(y)>3;

y(b)=NaN;

subplot(1,2,2)

grid on

hold on

plot(x,y,'b')

line([r(1), r(1)],[-3,3], 'Linestyle', '--', 'Color', 'r')

line([r(2), r(2)],[-3,3], 'Linestyle', '--', 'Color', 'r')

line([r(3), r(3)],[-3,3], 'Linestyle', '--', 'Color', 'r')

xlabel('X')

ylabel('Y')

title('График с тремя корнями полинома')

По функции g11 строится график и делается усечение по оси ординат от -3 до 3, рисуются 3 асимптоты и подписываются оси и заглавие

1. Построение графика «кусочно-заданной» функции

x1 = -10:0.01:0;

x2 = 0:0.01:10;

x3 = 10:0.01:20;

p = [0.9 -4.23 2.07 3.83];

y = f11(x1);

yp = polyval(p, x2);

yg = g11(x3);

hold on

plot(x1, y, 'k--')

plot(x2, yp, 'b-.')

plot(x3, yg, 'g')

Построение графиков y(x), P(x) и g(x) на разных интервалах в разных стилях

x01 = fzero(@f11, -1)

y01 = f11(x01)

plot(x01, y01, 'bo')

r=roots(p)

r = r(r > 0 & r < 10);

yr=polyval(p,r)

plot(r, yr, 'r\*')

x03=fzero(@g11,10)

y03=g11(x03)

plot(x03, y03, 'ks')

Отметка корней всех трех интервалов в разных стилях

title('График кусочно заданной функции')

grid on

xlabel('x')

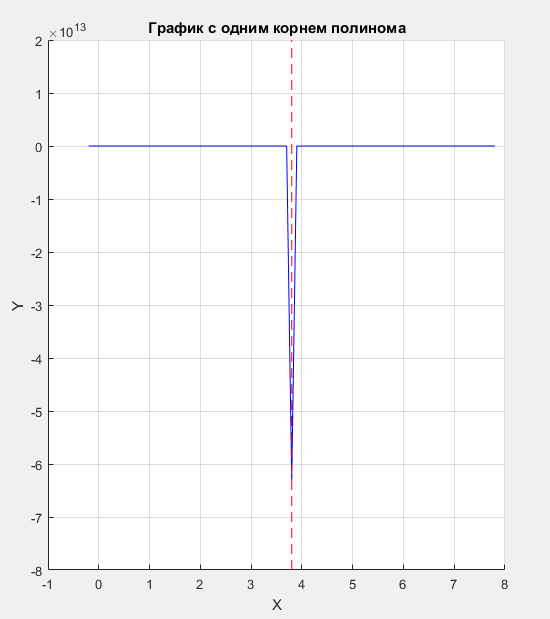
ylabel('y')

legend('График функции f(x)','График полинома P(x)','График функции g(x)', 'Корни f(x)', 'Корни P(x)', 'Корни g(x)', 'Location','northwest')

Пишется заглавие, оси и легенда графика

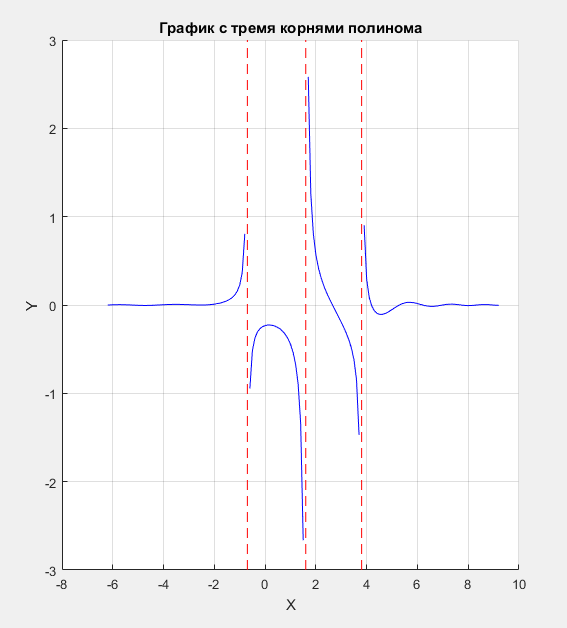
**Полученные в ходе работы результаты:**

1. Построение графика на интервале



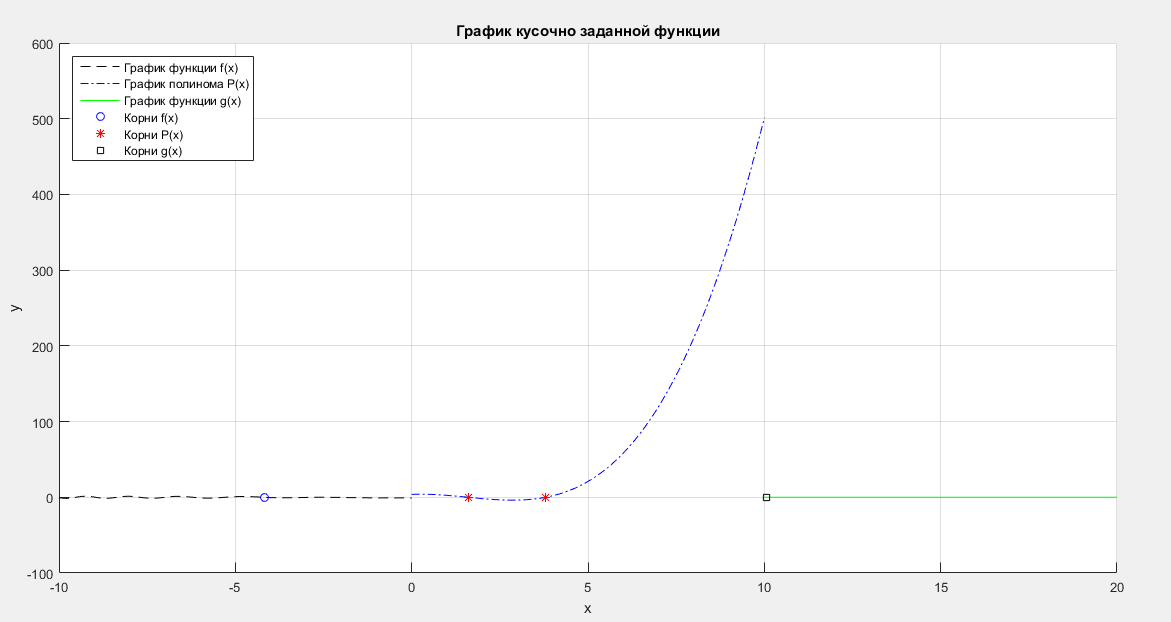
Рисунок

1. Построение графика на интервале с усечением



Рисунок

1. Построение графика «кусочно-заданной» функции



Рисунок

# 

# Задание №3

# Название работы: *«Построение двух поверхностей второго порядка»*

**Необходимо выполнить:**

Задание двух поверхностей:

Эллиптического параболоида: , где

Двуполостной гиперболоид: , где

а) Построение графиков каждой поверхности на отдельных осях (одна – каркасная, другая – плёночная, используя разные цветовые палитры)

## б) Построение графиков одной из поверхностей на 4 подграфиках (каркасная, плёночная – два варианта с разной закраской) задавая разные точки обзора.

## с) Построение графиков другой поверхности, освещённой источником света на 4-х подграфиках, задав два разных положения источника и две разные точки обзора.

**Описание хода работы:**

1. Построение эллиптического параболоида:

* Строится каркасная поверхность в декартовой системе координат, где

Задается сетка на плоскости x-y в виде двумерных массивов X, Y, которые определяются одномерными массивами x и y. Строки массива X являются копиями вектора x, а столбцы - копиями вектора y.

* Из изначального уравнения выражается z через x и y. Строится поверхность с помощью функции mesh в каркасном виде и функции surf в пленочном.
* Строится эта же поверхность в виде плёночной на четырёх подграфиках в цветовой палитре autumn с разными точками обзора и разным точками освещения.

1. Построение двуполостного гиперболоида:

* Строится каркасная поверхность в декартовой системе координат, где

Задается сетку на плоскости x-y в виде двумерных массивов X,Y, которые определяются одномерными массивами x и y. Строки массива X являются копиями вектора x, а столбцы - копиями вектора y.

* Из изначального уравнения выражается зависимость z от x и y. Z выражается как плюс корень и минус корень, за счет чего получаются две симметричные полости
* Строится эта же поверхность на четырёх подграфиках в каркасном и пленочном виде, задавая две разные точки обзора.

**Код работы:**

1. Построение эллиптического параболоида

1) Построение графика:

a = 9.47;

b = 8.11;

x0 = 0.84;

y0 = 1.88;

z0 = 1.99;

x = -50 : 1 : 50;

y = -50 : 1 : 50;

[X, Y] = meshgrid(x, y);

Z = (X-x0).^2/a^2+(Y-y0).^2/b^2+z0;

figure

subplot(1, 2, 1)

mesh(X, Y, Z)

subplot(1, 2, 2)

surf(X, Y, Z)

shading interp

Задаются промежутки для x и y. Строится сетку с помощью встроенной функции meshgrid.

С помощью функций mesh и surf строятся соответственно каркасная и пленочная поверхности.

2) Первый подграфик:

figure

subplot(2, 2, 1)

surfl(X, Y, Z, [100 100])

view(50, 50)

shading interp

С помощью функции surfl строится пленочная поверхность с координатами источника света [100; 100] и точкой обзора [50; 50]

1. Второй подграфик:

subplot(2, 2, 2)

surfl(X, Y, Z, [100 100])

view(100, 100)

shading interp

С помощью функции surfl строится пленочная поверхность с координатами источника света [100; 100] и точкой обзора [100; 100]

4) Третий подграфик:

subplot(2, 2, 3)

surfl(X, Y, Z, [150 50])

shading interp

view(50, 50)

С помощью функции surfl строится пленочная поверхность с координатами источника света [150; 150] и точкой обзора [50; 50]

5) Четвертый подграфик:

subplot(2, 2, 4)

surfl(X, Y, Z, [150 50])

shading interp

view(100, 100)

colormap autumn

С помощью функции surfl строится пленочная поверхность с координатами источника света [150; 150] и точкой обзора [100; 100], изменяется цветовая палитра

1. Построение двуполостного гиперболоида:

1) Построение графика:

a= 1.50;

b= 9.58;

c= -5.30;

x0= -1.70;

y0= -0.75;

z0= 1.58;

x = -5 : 1 : 5;

y = -25 : 1 : 25;

[X, Y] = meshgrid(x, y);

Z1 = sqrt((1 + (X-x0).^2/a^2 + (Y-y0).^2/b^2)\*c^2)+z0;

Z2 = -sqrt((1 + (X-x0).^2/a^2 + (Y-y0).^2/b^2)\*c^2)+z0;

figure

subplot(1, 2, 1)

mesh(X, Y, Z1)

hold on

mesh(X, Y, Z2)

subplot(1, 2, 2)

surf(X, Y, Z1)

hold on

surf(X, Y, Z2)

shading interp

colormap summer

Задаются промежутки для x и y. Строится сетка с помощью встроенной функции meshgrid. С помощью функций mesh и surf строятся соответственно каркасная и пленочная поверхности

3) Первый подграфик:

subplot(2, 2, 1)

mesh(X, Y, Z1)

hold on

mesh(X, Y, Z2)

view(150, 50)

С помощью встроенной функции mesh строится каркасная поверхность, функция view задает точку обзора [150; 50]

4) Второй подграфик:

subplot(2, 2, 2)

mesh(X, Y, Z1)

hold on

mesh(X, Y, Z2)

view(50, 50)

С помощью встроенной функции mesh строится каркасная поверхность, функция view задает точку обзора [50; 50]

5) Третий подграфик:

subplot(2, 2, 3)

surf(X, Y, Z1)

hold on

surf(X, Y, Z2)

shading interp

view(50, 150)

С помощью встроенной функции surf строится пленочная поверхность, функция view задает точку обзора [50; 150]

6) Четвёртый подграфик:

subplot(2, 2, 4)

surf(X, Y, Z1)

hold on

surf(X, Y, Z2)

shading interp

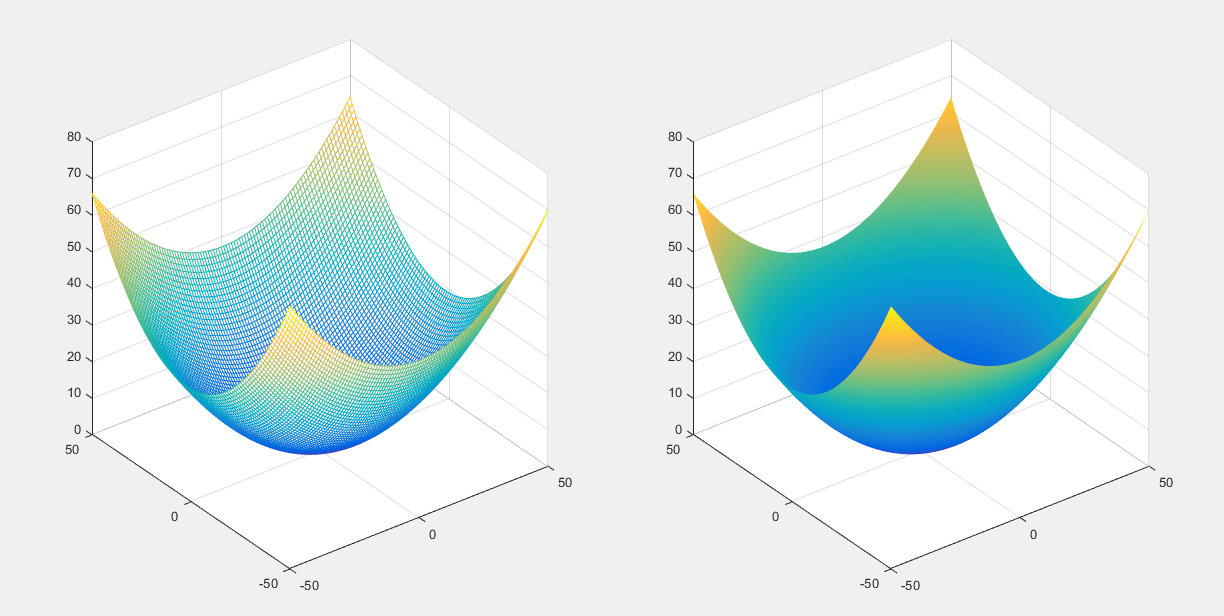
view(150, 150)

colormap winter

С помощью встроенной функции surf строится пленочная поверхность, функция view задает точку обзора [150; 150]

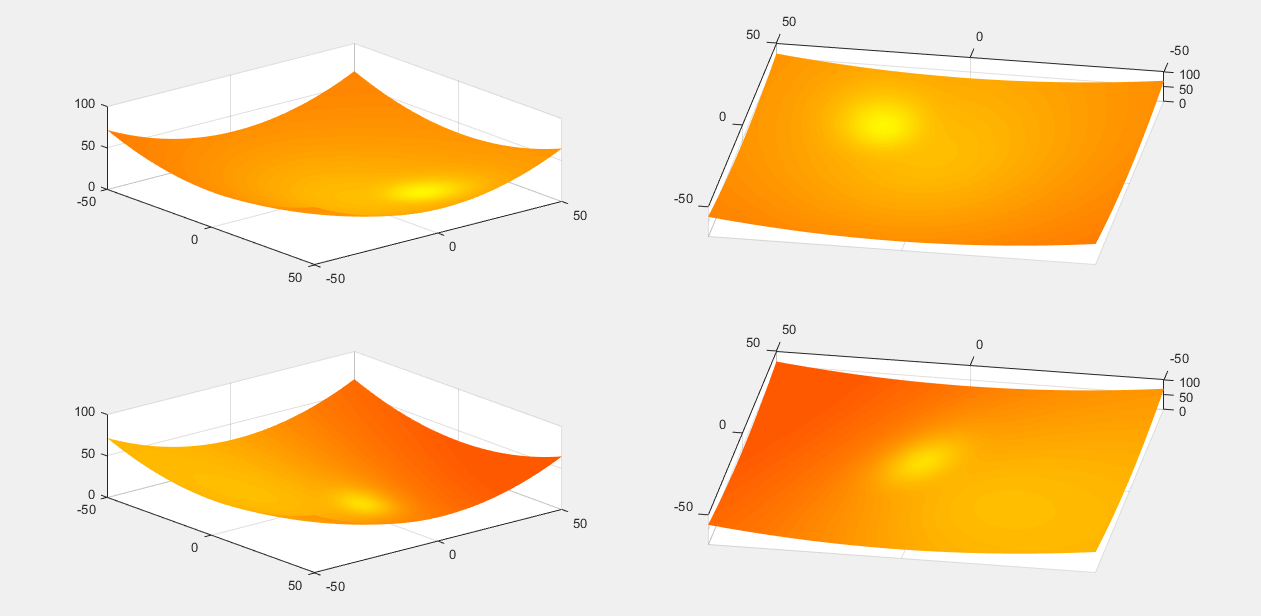
**Полученные в ходе работы результаты:**

1. Построение эллиптического параболоида



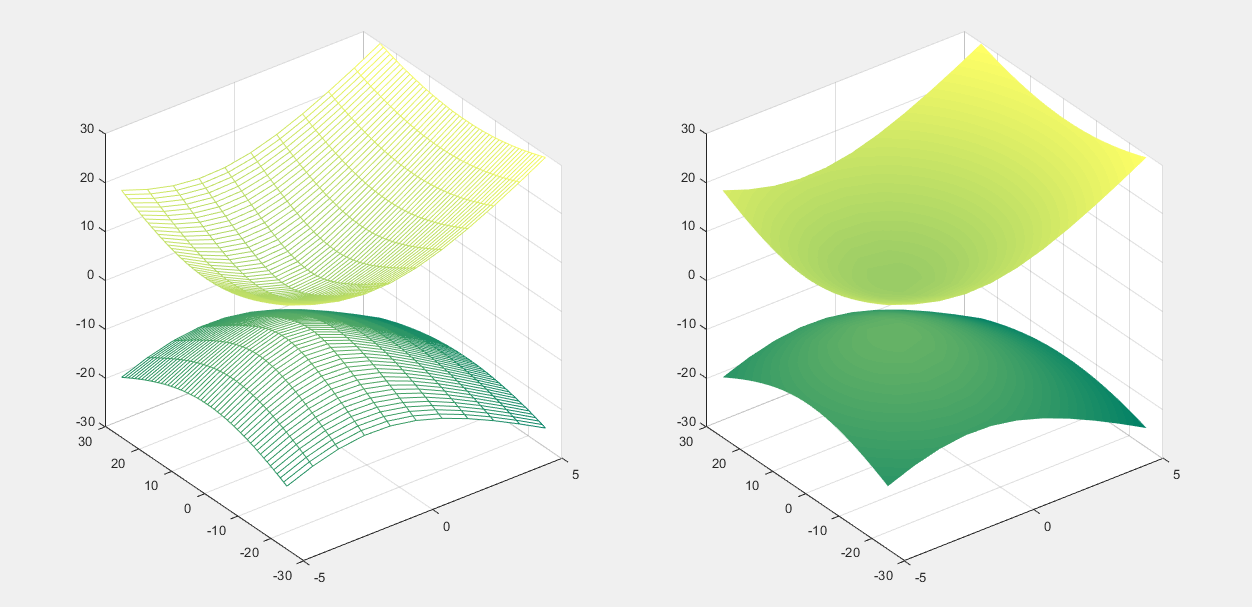
Рисунок

2. Эллиптический параболоид в четырёх подграфиках



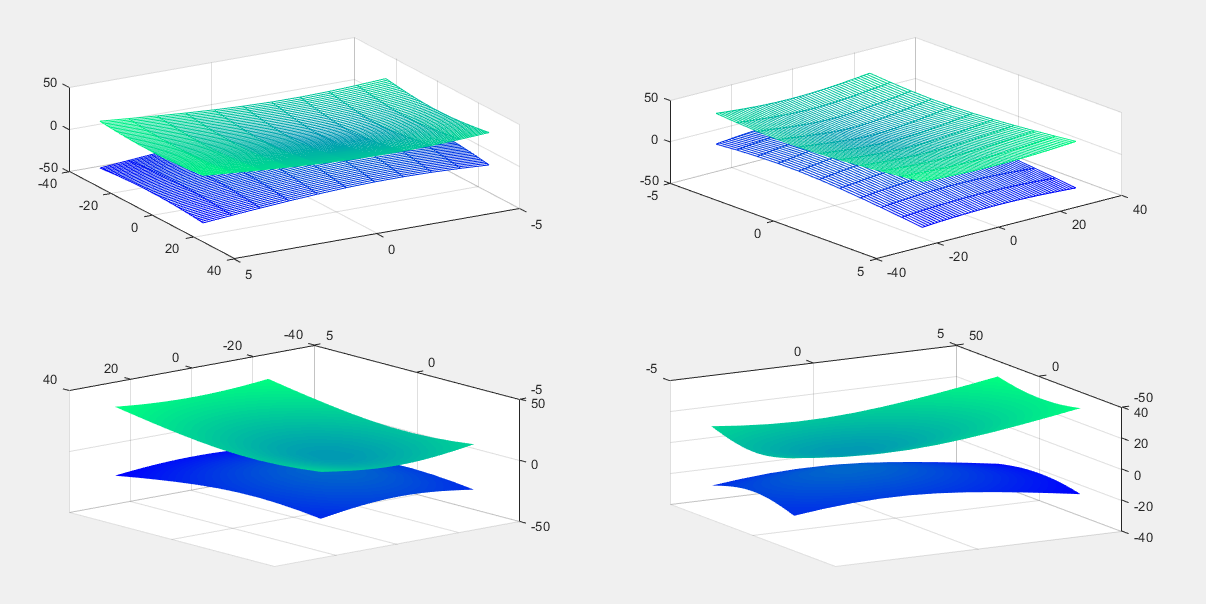
Рисунок

3. Построение двуполостного гиперболоида



Рисунок

4. Построение двуполостного гиперболоида в четырёх подграфиках



Рисунок

# Задание №4

# Название работы: *«Построение взаимного расположения двух поверхностей второго порядка»*

**Необходимо выполнить:**

Графическое представление взаимного расположения и пересечения поверхностей, заданных в предыдущем задании, выбрав три разных точки обзора. При необходимости, выбрать область построения каждой поверхности независимо так, чтобы обе поверхности были хорошо видны. Одна поверхность изображается как каркасная, другая как пленочная.

## а) в декартовой системе координат

## б) используя параметрическое задание поверхностей (цилиндрические или сферические координаты)

Описание хода работы:

1) Задание поверхностей в декартовой системе

* Строится эллиптический параболоид на интервалах так же, как в предыдущем задании.
* Строится двуполостной гиперболоид, заданный на интервалах
* Строятся разные подграфики, каждый раз меняя точки обзора.

2) Параметрическое задание плоскостей

* Задаются параметрические плоскости
* Эллиптический параболоид задается следующей зависимостью
* В матрице Z делаются одинаковые столбцы, чтобы их число совпадало с длиной вектора , для чего используется следующее присваивание:
* Двуполостной гиперболоид задается следующей зависимостью
* Строятся разные подграфики, каждый раз меняя точки обзора.

**Код работы:**

1. Задание поверхностей в декартовой системе

1) Эллиптический параболоид

a = 9.47;

b = 8.11;

x0 = 0.84;

y0 = 1.88;

z0 = 1.99;

x = -5 : 1 : 5;

y = -25 : 1 : 25;

[X, Y] = meshgrid(x, y);

Z = (X-x0).^2/a^2+(Y-y0).^2/b^2+z0;

Задаются промежутки для х и y. Строится сетка с помощью встроенной функции meshgrid.

2) Двуполостной гиперболоид

a= 1.50;

b= 9.58;

c= -5.30;

x0= -1.70;

y0= -0.75;

z0= 1.58;

x = -5 : 1 : 5;

y = -25 : 1 : 25;

[X, Y] = meshgrid(x, y);

Z1 = sqrt((1 + (X-x0).^2/a^2 + (Y-y0).^2/b^2)\*c^2)+z0;

Z2 = -sqrt((1 + (X-x0).^2/a^2 + (Y-y0).^2/b^2)\*c^2)+z0;

Задаются промежутки для x и y. Строится сетка с помощью встроенной функции meshgrid. Z задается двумя уравнениями для отрисовки двух симметричных полостей

3) Первый подграфик

subplot(1, 3, 1)

mesh(X, Y, Z)

hold on

shading interp

surf(X, Y, Z1)

surf(X, Y, Z2)

shading interp

view(0, 10)

Совмещаются две поверхности вместе, угол обзора ставится (0; 10).

4) Второй подграфик

subplot(1, 3, 2)

mesh(X, Y, Z)

hold on

shading interp

surf(X, Y, Z1)

hold on

surf(X, Y, Z2)

shading interp

view(20, 30)

Совмещаются две поверхности вместе, угол обзора ставится (20; 30).

5) Третий подграфик

subplot(1, 3, 3)

mesh(X, Y, Z)

hold on

shading interp

surf(X, Y, Z1)

hold on

surf(X, Y, Z2)

shading interp

colormap summer

view(70, 10)

Совмещаются две поверхности вместе, угол обзора ставится (70; 10). Меняется цветовая палитра.

1. Параметрическое задание плоскостей

1) Эллиптический параболоид

a = 9.47;

b = 8.11;

x0 = 0.84;

y0 = 1.88;

z0 = 1.99;

u = (-40:0.5:40)';

v = [0:0.05\*pi:2\*pi];

X = x0 + a\*u\*cos(v);

Y = y0 + b\*u\*sin(v);

Z = z0 + u.^2\*ones(size(v));

Параметрически задается эллиптический параболоид, исходя из выше написанных зависимостей

2) Двуполостной гиперболоид

a= 1.50;

b= 9.58;

c= -5.30;

x0= -1.70;

y0= -0.75;

z0= 1.58;

u1 = (-10:0.1:10)';

v1 = 0:pi/100:2\*pi;

[U, V1] = meshgrid(u1,v1);

X1 = x0 + a\*sinh(V1).\*cos(U);

Y1 = y0 + b\*sinh(V1).\*sin(U);

Z1 = z0 + c\*cosh(V1);

Z2 = z0 - c\*cosh(V1);

Параметрически задается двуполостной гиперболоид, исходя из выше написанных зависимостей

1. Первый подграфик

subplot(1, 3, 1)

surf(X, Y, Z)

hold on

shading interp

mesh(X1, Y1, Z1)

mesh(X1, Y1, Z2)

shading interp

view(0, 10)

Совмещаются две поверхности вместе, угол обзора ставится (0; 10).

4) Второй подграфик

subplot(1, 3, 2)

surf(X, Y, Z)

hold on

shading interp

mesh(X1, Y1, Z1)

hold on

mesh(X1, Y1, Z2)

shading interp

view(20, 30)

Совмещаются две поверхности вместе, угол обзора ставится (20; 30).

5) Третий подграфик

subplot(1, 3, 3)

surf(X, Y, Z)

hold on

shading interp

mesh(X1, Y1, Z1)

hold on

mesh(X1, Y1, Z2)

shading interp

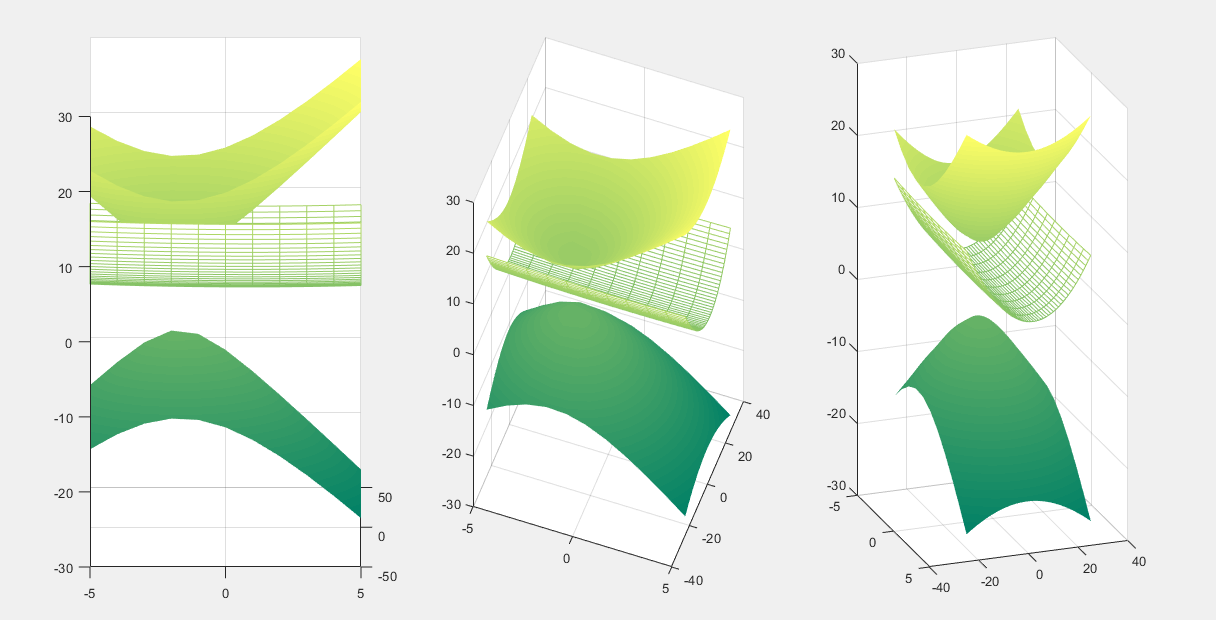
colormap summer

view(70, 10)

Совмещаются две поверхности вместе, угол обзора ставится (70; 10). Меняется цветовая палитра.

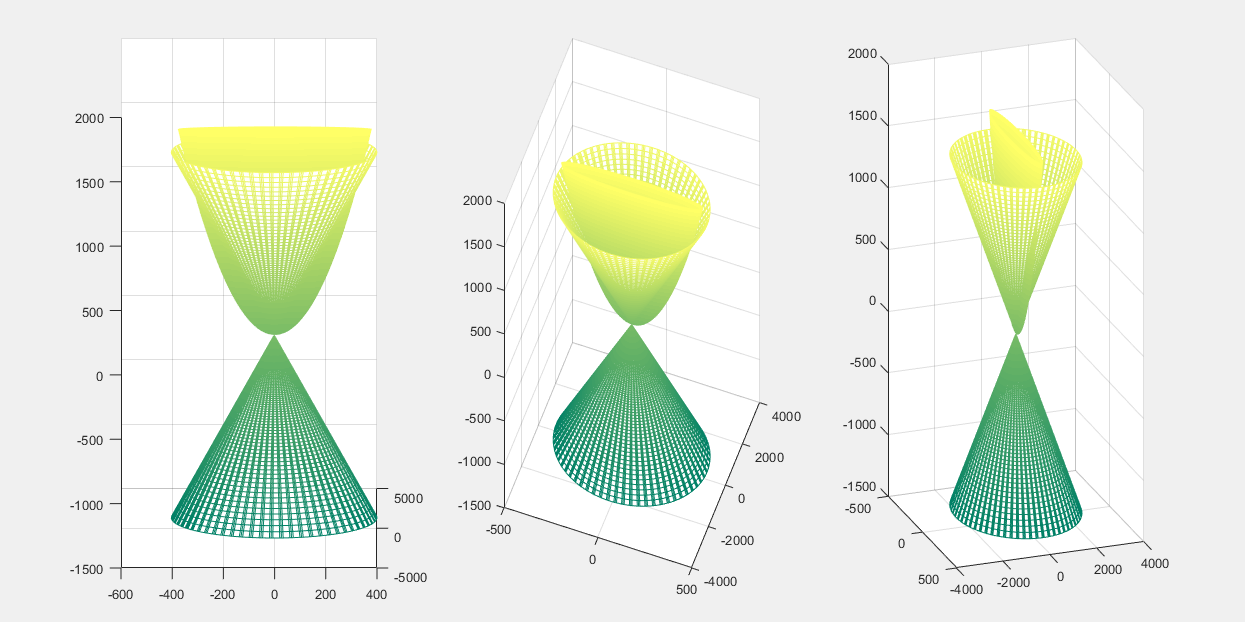
**Полученные в ходе работы результаты:**

1. Совмещение поверхностей в декартовой системе координат



Рисунок

2. Совмещение поверхностей, заданных параметрически



Рисунок

**Вывод:**

Исходя из проделанной работы, можно наблюдать, что изображения в декартовой системе координат и полярной немного отличаются друг от друга. Причиной тому являются разные интервалы, на которых задаются плоскости. В декартовой с.к. по оси Ох и Оу взяты интервалы . В то время в полярной с.к. интервалы

# Задание №5

# Название работы: *«Исследование свойств полиномов высокого порядка»*

**Необходимо выполнить:**

Для полинома

## a) Построение графика на интервале, содержащем все корни.

## б) Построение графиков на 2-х более узких интервалах, где следует уточнить поведение функции.

## с) Поиск первых 3 производных и их вещественных корней.

## d) Построение графиков полинома и его производных на четырех подграфиках друг под другом, обозначая на них точки, соответствующие вещественным корням.

## е) Сравнить координаты точек экстремумов, вычисленных двумя способами:

* используя функцию поиска минимума,
* как корень производной.

## f) Внесение случайного изменения, используя функцию rand, во второй коэффициент полинома (не более 5 %) и сравнение (вычисление относительной погрешности) каждого корня полинома.

**Описание хода работы:**

1) Построение полинома

* Построение графика полинома на интервале синей сплошной линией.
* Находятся корни полинома, принадлежащие этому интервалу, отделяются все действительные и обозначаются на графике с помощью встроенной функции plot.

2) Два фрагмента для лучшего рассмотрения поведения полинома

* Строится фрагмент полинома на интервале [-**1; 1**] синей сплошной линией. С помощью встроенных функций xlabel и ylabel подписываются оси.
* На этом промежутке находятся корни полинома, обозначенные на графике красным кругом.
* То же самое проделывается на интервале [-**6; -3**]

3) Построение полинома и его производных

* Находятся первые три производные полинома, каждый раз с применением функции polyder. Отделяются действительные корни.
* Рисуются 4 подграфика с изображением полинома и 3 его производных с отметкой корней на каждом графике

4) Сравнение результатов по поиску экстремумов

* Ищутся экстремумы полинома с помощью встроенной функции fminbnd на интервалах , , и . Сравниваются ранее полученные результаты и вычисляется разность в вычислениях.

5) Внесение случайных изменений

* Случайное изменение вносится во второй коэффициент в пределах 5%, находятся новые корни получившегося полинома.
* Новые корни полинома сравниваются с изначальными, вычисляется их разность по модулю.

**Код работы:**

1. Построение полинома

1) Создание функции

function y = f11(x)

y = -1.29.\*x^11-8.21.\*x^10-4.85.\*x^9-4.6.\*x^8-6.27.\*x^7+5.81.\*x^6+6.92.\*x^5+6.83.\*x^4+3.53.\*x^3+5.81.\*x^2+0.11.\*x-6.66;

В отдельном окне создается функция f11

2) Полином

p = [-1.29 -8.21 -4.85 -4.60 -6.27 5.81 6.92 6.83 3.53 5.81 0.11 -6.66];

x = -5.5 : 0.05 : 3;

y = polyval(p , x);

plot(x, y);

xlabel('x')

ylabel('y')

title('График полинома')

grid on

hold on

Задается область определения полинома, рисуется график сплошной синей линией и подписываются оси и заглавие

3) Корни

r = roots(p);

x0 = r(imag(r) == 0);

x0 = x0(x0 > -5.5 & x0 < 3);

plot(x0, 0, 'ro');

С помощью функции roots находятся корни полинома, отделяются действительные корни на заданном промежутке и отмечаются на графике.

1. Построение полинома на двух интервалах

1) Построение на промежутке от -1 до 1

p = [-1.29 -8.21 -4.85 -4.60 -6.27 5.81 6.92 6.83 3.53 5.81 0.11 -6.66];

x = -1 : 0.01 : 1;

y = polyval(p , x);

subplot(2, 1, 1)

plot(x, y)

xlabel('x')

ylabel('y')

grid on

hold on

title('График полинома от -1 до 1')

Задается промежуток, рисуется график и подписываются оси и заглавие

2) Корни

r = roots(p);

x0 = r(imag(r) == 0);

x0 = x0(x0 > -1 & x0 < 1);

plot(x0, 0, 'ro')

legend('График полинома', 'корни полинома', 'Location', 'southeast')

Ищутся корни полинома, отделяются действительные на заданном промежутке и выводятся на график

5) Второй интервал от -6 до -3

x = -6 : 0.01 : -3;

y = polyval(p , x);

subplot(2, 1, 2)

plot(x, y)

xlabel('x')

ylabel('y')

title('График полинома от -6 до -3')

grid on

hold on

r = roots(p);

x0 = r(imag(r) == 0);

x0 = x0(x0 > -6 & x0 < -3);

plot(x0, 0, 'ro')

legend('График полинома', 'корни полинома', 'Location', 'southeast')

Так же строится график для второго интервала и находятся корни

1. Производные

1) Полином

p = [-1.29 -8.21 -4.85 -4.60 -6.27 5.81 6.92 6.83 3.53 5.81 0.11 -6.66];

x = -6 : 0.5 : 3;

y = polyval(p , x);

subplot(4, 1, 1)

plot(x, y);

xlabel('x')

ylabel('y')

grid on

hold on

title('График полинома')

r = roots(y);

x0 = r(imag(r) == 0)

plot(x0, 0, 'ro');

1) Первая производная

p1 = polyder(p);

y1 = polyval(p1, x);

subplot(4, 1, 2)

plot(x, y1);

xlabel('x')

ylabel('y')

grid on

hold on

title('График первой производной полинома')

r1 = roots(p1);

x0 = r1(imag(r1) == 0);

plot(x0, 0, 'ro');

2) Вторая производная

p2 = polyder(p1);

y2 = polyval(p2, x);

subplot(4, 1, 3)

plot(x, y2);

xlabel('x')

ylabel('y')

grid on

hold on

title('График второй производной полинома')

r2 = roots(p2);

x0 = r2(imag(r2) == 0);

plot(x0, 0, 'ro');

3) Третья производная

p3 = polyder(p2);

y3 = polyval(p3, x);

subplot(4, 1, 4)

plot(x, y3);

xlabel('x')

ylabel('y')

grid on

hold on

title('График третьей производной полинома')

r3 = roots(p3);

x0 = r3(imag(r3) == 0);

plot(x0, 0, 'ro');

1. Экстремумы встроенной функцией

p = [-1.29 -8.21 -4.85 -4.60 -6.27 5.81 6.92 6.83 3.53 5.81 0.11 -6.66];

x = -5.5 : 0.05 : 3;

y = polyval(p , x);

p1 = polyder(p);

r1 = roots(p1);

x01 = r1(imag(r1) == 0);

x01 = sort(x01)

y01 = polyval(p, x01);

r21 = fminbnd(@f11, -5, -2);

r22 = fminbnd(@f12, -1.5, -0.5);

r23 = fminbnd(@f11, -0.5, 0);

r24 = fminbnd(@f12, 0, 2);

С помощью встроенной функции fminbnd ищутся два максимума и два минимума.

r2 = [r21 r22 r23 r24];

r2 = r2';

r2 = sort(r2)

delta = abs(x01 - r2)./abs(x01)

%Вывод: Нахождение экстремумов через производную дает более точный

%результат

Ранее найденные экстремумы заполняются в массив и сортируются. Высчитывается относительная погрешность нахождения экстремумов

1. Внесение погрешности во второй коэффициент

pr = [-1.29 (rand\*(-8.21)\*0.05-8.21) -4.85 -4.60 -6.27 5.81 6.92 6.83 3.53 5.81 0.11 -6.66];

С помощью функции rand вносится случайное изменение значения коэффициента в пределах 5%.

p = [-1.29 -8.21 -4.85 -4.60 -6.27 5.81 6.92 6.83 3.53 5.81 0.11 -6.66];

x = -6 : 0.05 : 3;

y = polyval(p , x);

r = roots(p)

xr = -6 : 0.05 : 3;

yr = polyval(pr , xr);

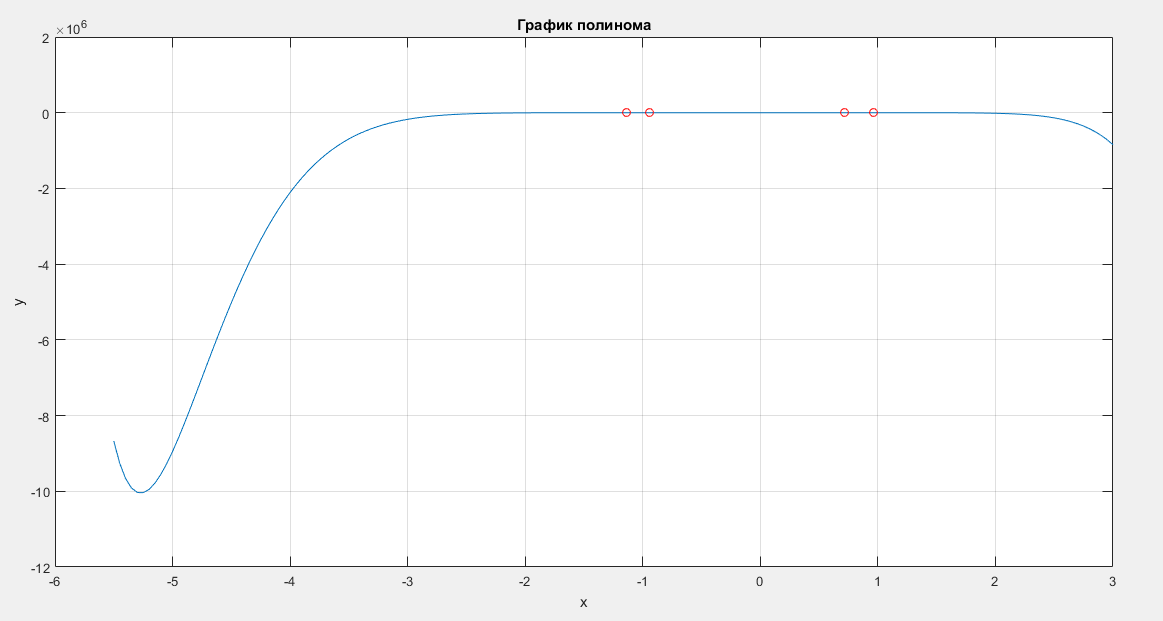
r2 = roots(pr)

delta = abs(r - r2)./abs(x0) %относительная погрешность всех корней

Задается изначальный полином, считаются корни исходного и нового полиномов, считается относительная погрешность каждого из корней

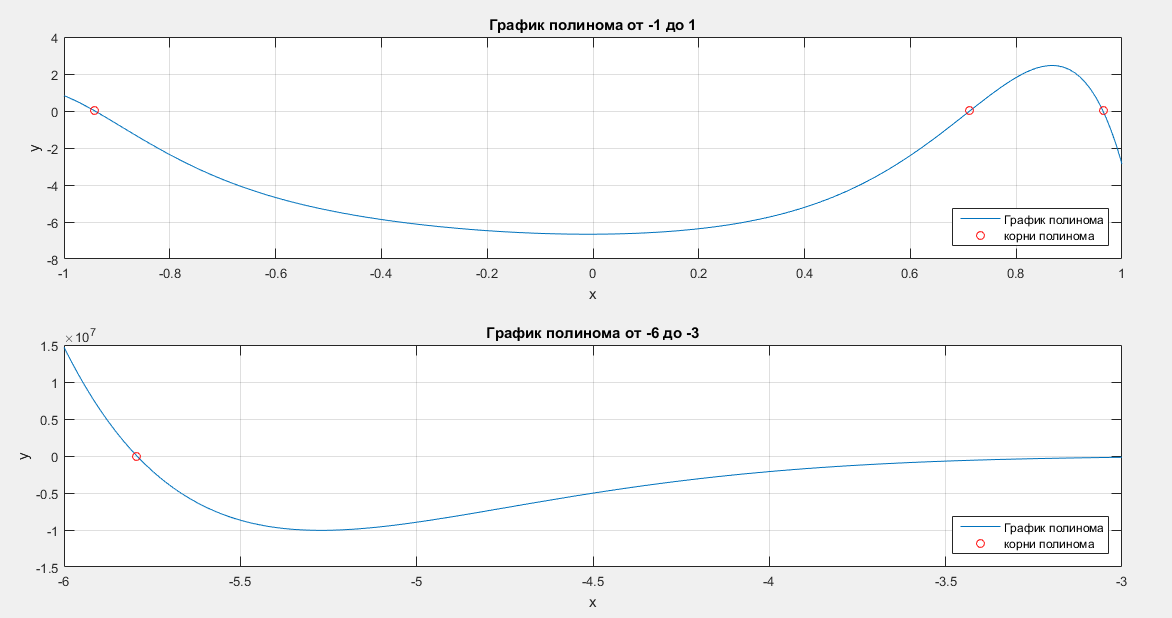
Полученные в ходе работы результаты:

1. Построение полинома



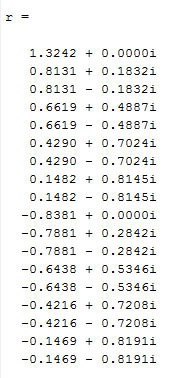
Рисунок

2. Построение графиков на 2-х более узких интервалах



Рисунок

3. Корни полинома

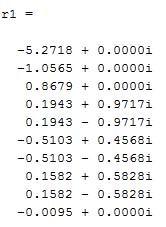


Рисунок



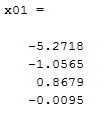
Рисунок

3. Корни производных полинома



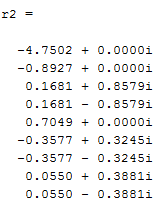
Рисунок

Корни первой производной



Рисунок

Действительные корни первой производной



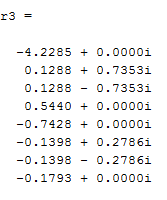
Рисунок

Корни второй производной



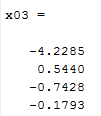
Рисунок

Действительные корни второй производной



Рисунок

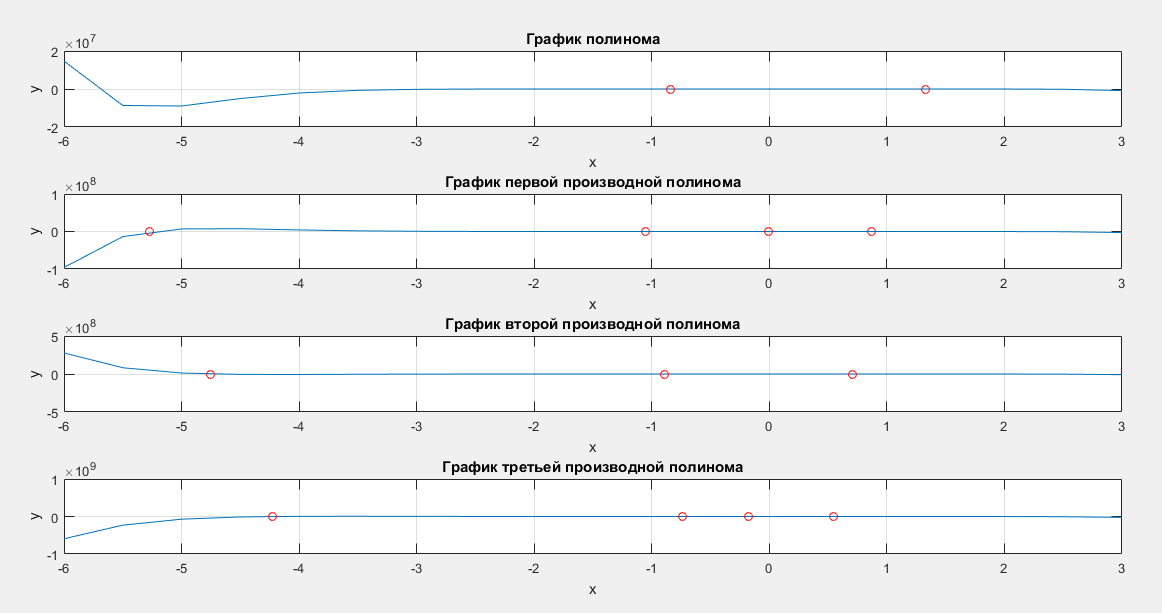
Корни третьей производной



Рисунок

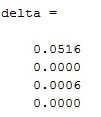
Действительные корни третьей производной

4. Графики производных



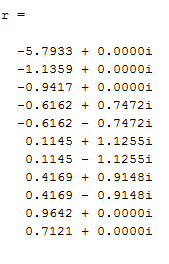
Рисунок

1. Сравнение координат точек экстремумов

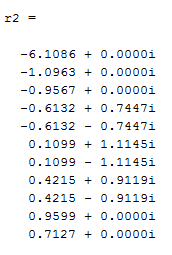


Рисунок

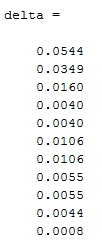
5. Внесение изменения во второй коэффициент (не более 5%)



Рисунок



Рисунок



Рисунок

**Выводы:**

1) При сравнении экстремумов, найденных разными способами, наибольшее изменение в первом значении начинается со второго знака и **.** Дальнейшие изменения незначительны и не превышают 0.0006

2) При сравнении корней полинома до и после внесения изменения во второй коэффициент значения начинают изменяться со 2 знака после запятой и уменьшается по мере увеличения значений. Наибольшая дельта в первом корне

# Задание №6

# Название работы: *«Решение задач алгебры».*

Дана матрица:

4.0036 0.0349 0.1033 0.1814 -1.3681 -0.9444 -1.7343 -0.7499 -1.6131 -1.3416

-0.4427 0.5106 0.3078 1.2297 0.0864 0.9803 -0.4900 -0.2279 -0.2210 -0.8128

-1.0580 -0.6734 3.5960 -0.2885 -0.4730 -0.8787 -0.8734 0.4132 0.2820 0.0363

-0.1594 0.7474 -0.2877 0.6398 -0.0479 2.2220 -0.7357 -0.2582 0.0648 0.0444

-1.0251 -0.1159 -0.6956 -0.5091 3.6188 0.0770 -1.1748 -0.2900 -0.8564 -0.4043

1.2350 2.9842 0.1709 2.7426 -0.0517 5.6438 -0.2620 -1.4299 -0.5839 0.3542

-0.3026 0.5208 -0.3356 0.2652 -1.0246 0.5587 3.5237 -0.4150 -0.3854 0.1960

-0.0379 0.1112 -0.0528 0.0676 -0.1665 0.1393 -0.2615 -0.0340 0.9803 0.6166

-0.0094 0.4097 -0.1027 0.3028 -0.3578 0.6116 -0.6353 -0.5988 3.2024 0.4989

0.3794 0.7839 0.0837 0.7431 0.0942 1.4693 -0.2148 -0.5111 -0.2363 0.0355

И ее свободные члены:

0.6664 -4.3578 -4.2221 -4.2681 6.2706 -15.6073 -1.0886 -1.0674 -3.7180 -3.7203

**Необходимо выполнить:**

## а) Решение СЛАУ по правилу Крамера

## б) Решение СЛАУ с использованием обратной матрицы

## с) Решение СЛАУ при помощи операции "\". Сравнить решения по значению максимальной компоненты вектора невязок

## d) Внести случайное изменение (в пределах 2%, 3%, 4%, 5 %) в правую часть системы, используя функцию rand, и построить график относительной погрешности решения, полученного с помощью операции "\", от относительной погрешности исходных данных.

**Описание хода работы:**

* Из текстовых файлов считывается матрица и ее свободные члены, правая часть транспонируется.
* Находится определитель матрицы не равный нулю.
* В цикле с каждой итерацией на i-тое место подставляется вектор свободных коэффициентов и вычисляется определитель. Для нахождения корня он делится на определитель исходной матрицы.

* Ищутся корни СЛАУ с помощью метода обратной матрицы используя функцию inv.
* Ищутся корни СЛАУ с помощью метода обратной черты
* Вектор корней транспонируется с помощью функции «’». Вычисляется максимальная разница, максимальные компоненты вектора невязки.
* Вводится погрешность в пределах 2%, 3%, 4% и 5%. Строится график погрешности.

**Код работы:**

1. Решение СЛАУ по правилу Крамера

1) Считывание данных, с которыми предстоит работать.

m = load('matrix.txt')

right = load('rightpart.txt')

y = det(m)

Функция load загружает матрицу из файла matrix.txt в переменную m и правую часть из файла rightpart.txt в переменную right. Функция det считает определитель матрицы m и записывает его в переменную y.

2) Вычисление корней

l = length(right);

for i = 1:l

maxr = m;

maxr(:,i) = right;

x(i) = det(maxr)/y

end

В цикле подставляется вектор коэффициентов каждый раз на нужное место, затем делится полученный новый определитель на уже имеющийся.

1. Решение СЛАУ с использованием обратной матрицы

right = right';

x1=inv(m)\*right

Правая часть транспонируется. Для нахождения корней обратная матрица умножается на вектор свободных коэффициентов

1. Нахождение корней СЛАУ с помощью метода обратной черты

x2=m\right

Для нахождения корней матрица делится на правую часть

1. Расчет максимального вектора невязки

1) Для метода Крамера

xT = x'

k0 = m\*xT - right %вектор невязки

t0 = max(abs(k0)) %находим максимальную разницу

2) Для метода обратной матрицы

k1 = m\*x1 - right %вектор невязки

t1 = max(abs(k1)) %находим максимальную разницу

3) Для метода обратной черты

k2 = m\*x2 - right; %вектор невязки

t2 = max(abs(k2)) %находим максимальную разницу

1. Работа с погрешностью

h = [0.02, 0.03, 0.04, 0.05];

for i=1:4

ri = right.\*(rand(size(right))\*h(i)) + right

x3 = m\ri

deltar(i)= max(abs((x2-x3)./x1)) %находим относительную погрешность правой части

end

plot(h, deltar);

title('Относительная погрешность правой части');

xlabel('% изменения');

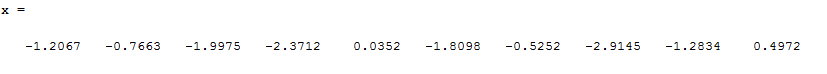
ylabel('погрешность');

grid on

Создается массив, в котором хранится погрешность. В цикле на каждой итерации считается, что получится исходя из заданной погрешности. График строится по рассчитанным значениям.

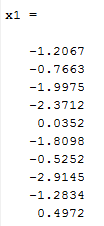
**Полученные в ходе работы результаты:**

1. Результаты, полученные с помощью метода Крамера



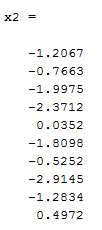
Рисунок

1. Результаты, полученные с помощью метода обратной матрицы



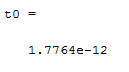
Рисунок

3. Результаты, полученные с помощью метода косой черты



Рисунок

4. Вектор невязки



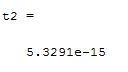
Рисунок

Вектор невязки для метода Крамера



Рисунок

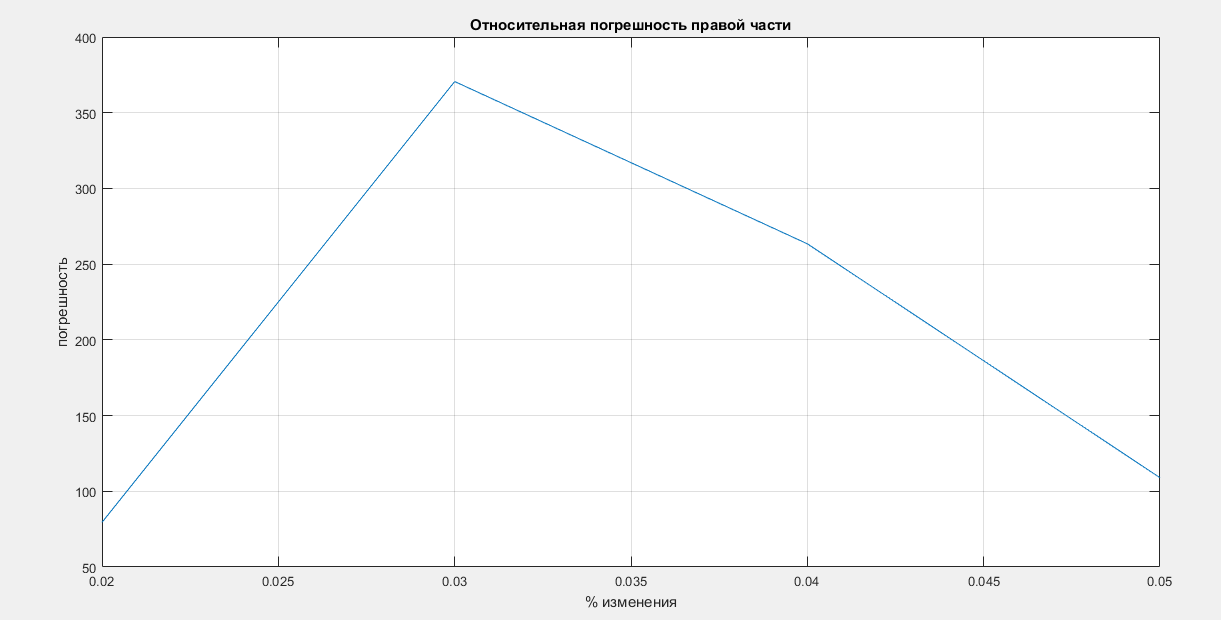
Вектор невязки для метода обратной матрицы



Рисунок

Вектор невязки для метода косой черты

5. Относительная погрешность



Рисунок

**Выводы:**

# 1. Лучшим способом нахождения корней СЛАУ является способ косой черты.

# 2. Худшим способом нахождения корней является метод по формулам Крамера.

# 3. С увеличением погрешности исходных данных увеличивается относительная погрешность корней СЛАУ.

# Задание №7

# Название работы: *«Решение задач средствами символьных вычислений».*

**Необходимо выполнить:**

## а) Используя средства символьных преобразований пакета вычислить, упростить и представить в удобной форме, выражения для пределов, производных.

## б) Решить кубическое уравнение из задания 1 аналитически и вычислить корни, используя функцию VPA. Сравнить их с посчитанными численно (функциями roots и fzero).

## в) Разложить заданную функцию в ряд Тейлора, удерживая разное число членов (2, 3, 4, 5). Вычислить коэффициенты полиномов и построить графики функции отрезка ряда Тейлора на симметричном (относительно точки разложения) интервале, используя функцию plot.

## г) Используя функцию rand, построить систему линейных уравнений третьего порядка, имеющую единственное решение, с целыми коэффициентами из интервала [- 9, 9]. Решить систему линейных уравнений аналитически, применяя функцию solve и обращая матрицу функцией inv. Удерживая 10 значащих цифр при помощи функции VPA, сравнить результаты с решением СЛАУ при помощи функции обратный слэш.

**Описание хода работы:**

* Вычислить пределы:
* Посчитать производные:

**y=**

Для каждого примера вывести формулу «красиво» с помощью pretty;

## Для кубического уравнения вычислить корни тремя способами:

* С помощью функции *solve* и привести к числу с помощью функции *vpa*;
* Как корни полинома *roots*;
* Как нули функции *fzero*;

Сравнить значения, посчитанные с помощью функции *solve* со значениями, полученными функциями *roots* и *fzero*.

* Разложить функцию в ряд Тейлора относительно с помощью функции *taylor*, удерживая разное число членов с помощью команды *Order.* Вычислить коэффициенты полиномов с помощью функции *sym2poly,* построить график полинома и функций разложения в ряд Тейлора в одних осях.
* Решить систему линейных уравнений аналитически, пользуясь встроенными функциями.

**Код работы:**

1. Вычисление пределов

x = sym('x')

a = sym('a')

m = sym('m')

b = sym('b')

n = sym('n')

h = sym('h')

Обозначаем, что x, a, m, b, n и h являются символами для дальнейшего пользования

1) Номер 452

f452 = ((1 + a\*x)^(1/m)-(1+ x\*b)^(1/n))/x;

pretty(f452)

f452l = limit(f452, x, 0);

pretty(f452l)

2) Номер 498

f498 = (1-cot(x)^3)/(2-cot(x)-cot(x)^3);

pretty(f498)

f498l = limit(f498, x, 0);

pretty(f498l)

3) Номер 585

f585 = (atan(x+h)-acot(x))/h;

pretty(f585)

f585l = limit(f585,x,Inf);

pretty(f585l)

1. Вычисление производных

1) Номер 900

y900 = (2+3\*x)/x^4\*(1-x^3)^(1/2)+3\*log(1+sqrt(1-x^2)/x);

pretty(y900)

y900d = diff(y900);

pretty(y900d)

2) Номер 916

y916 = sqrt(1/2)\*atan(sqrt(2)/x);

pretty(y916)

y916d = diff(y916);

pretty(y916d)

3) Номер 1358

y1358 = (a^x-x^a)/(x - a);

pretty(y1358)

y1358d = diff(y1358,4);

pretty(y1358d)

1. Решение уравнения разными способами

1) Решение уравнения функцией solve:

x = sym('x');

p = [0.9 -4.23 2.07 3.83];

t=solve(0.9\*x^3-4.23\*x^2-2.07\*x+3.83);

t1=vpa(t);

t1=sort(t1)

2) Решение уравнения нахождением корней полинома:

t2=roots(p);

t2=sort(t2)

3) Решение уравнения функцией поиска нулей функции:

В новом окне создается функция f11:

function y=f11(x)

y = 0.9\*x^3-4.23\*x^2-2.07\*x+3.83

В основном окне:

t31=fzero(@f11, -1, 0);

t32=fzero(@f11, 1, 2);

t33=fzero(@f11, 4, 5);

t3 = [t31 t32 t33];

if(size(t3) == [1 3])

t3 = t3'

end

sort(t3)

4) Сравнение значений

delta1=abs(t1-t2)

delta2=abs(t1-t3)

1. Разложение функции в ряд Тейлора

x = sym('x');

t2=taylor(x^2-log(x+1), x, 'Order', 2);

pretty(t2)

t3=taylor(x^2-log(x+1), x, 'Order', 3);

pretty(t3)

t4=taylor(x^2-log(x+1), x, 'Order', 4);

pretty(t4)

t5=taylor(x^2-log(x+1), x, 'Order', 5);

pretty(t5)

Преобразование из символьного в числовое значение

k2= sym2poly(t2)

k3= sym2poly(t3)

k4= sym2poly(t4)

k5= sym2poly(t5)

1. Графическое изображение разложения в ряд Тейлора

x1= -0.5:0.01:1;

y2= f12(x1); %график функции

plot(x1, y2, 'b-')

График функции

y1= polyval(k2, x1); %график разложения 2

plot(x1, y1, 'm-')

hold on

График разложения на 2 элемента

y1= polyval(k3, x1); %график разложения 3

plot(x1, y1, 'k-')

График разложения на 3 элемента

y1= polyval(k4, x1); %график разложения 4

plot(x1, y1, 'r-')

График разложения на 4 элемента

y1= polyval(k5, x1); %график разложения 5

plot(x1, y1, 'g-')

График разложения на 5 элементов

legend('График 2 разложения Тейлора', 'График 3 разложения Тейлора', 'График 4 разложения Тейлора', 'График 5 разложения Тейлора', 'График функции')

xlabel('x')

ylabel('y')

grid on

Подписывается легенда и оси

1. Система линейных уравнений с случайными коэффициентами

syms x01 x02 x03;

matrix = random('unid', 19, 3, 3) - 10 %матрица 3 на 3 из случайных чисел от -9 до 9

right = random('unid', 19, 3, 1) - 10 %массив правых коэффициентов от -9 до 9

f1= matrix(1,1)\*x01 + matrix(1,2)\*x02 + matrix(1,3)\*x03 - right(1,1)

f2= matrix(2,1)\*x01 + matrix(2,2)\*x02 + matrix(2,3)\*x03 - right(2,1)

f3= matrix(3,1)\*x01 + matrix(3,2)\*x02 + matrix(3,3)\*x03 - right(3,1)

[rot(1), rot(2) ,rot(3)]= solve (f1, f2, f3, 'x01', 'x02', 'x03')

rot= vpa(rot, 10)

if(size(rot) == [1 3])

rot = rot'

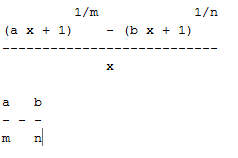
end

root1= matrix\right

delta=abs((root1 - rot)./rot)

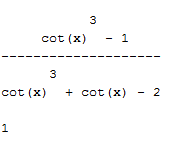
Полученные в ходе работы результаты:

1. Номер 452



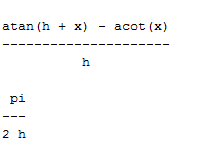
Рисунок

2. Номер 498



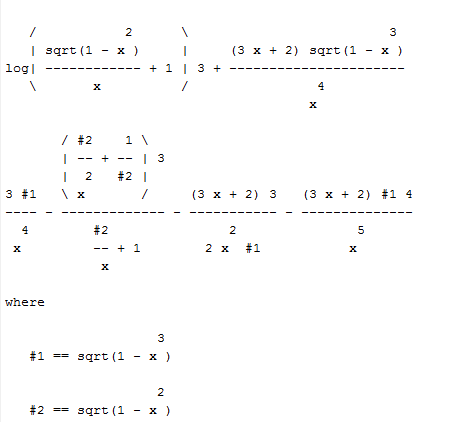
Рисунок

3. Номер 585



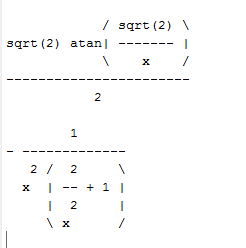
Рисунок

4. Номер 900



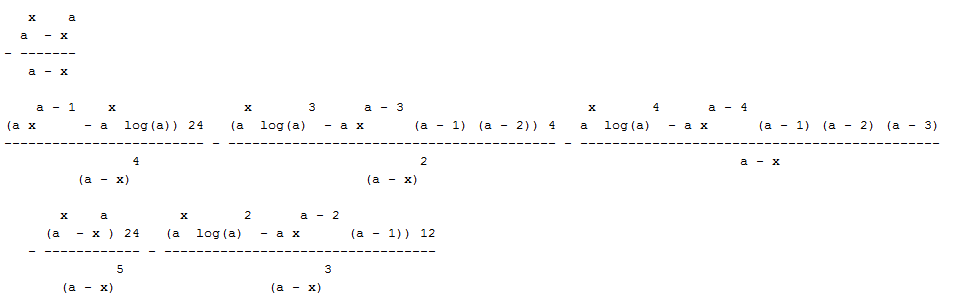
Рисунок

5. Номер 916



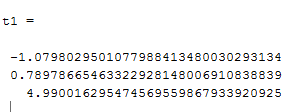
Рисунок

6. Номер 1358



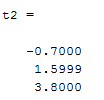
Рисунок

7. Решение уравнения функцией solve



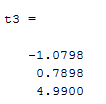
Рисунок

8. Решение уравнения нахождение корней полинома



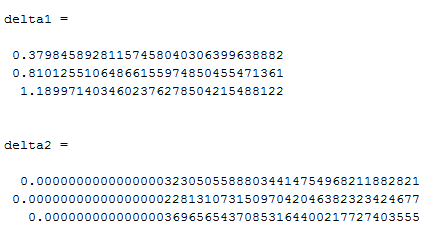
Рисунок

9. Решение уравнения функцией поиска нулей функции



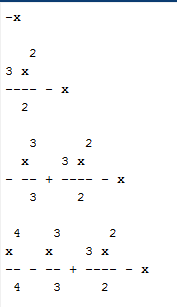
Рисунок

Сравниваем значения:



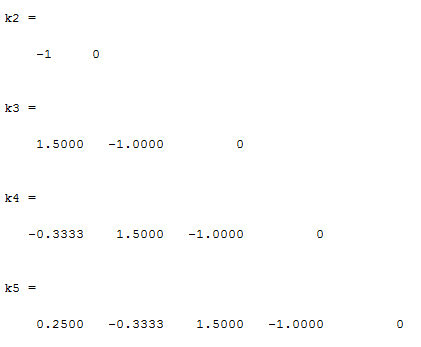
Рисунок

10. Разложение функции в ряд Тейлора



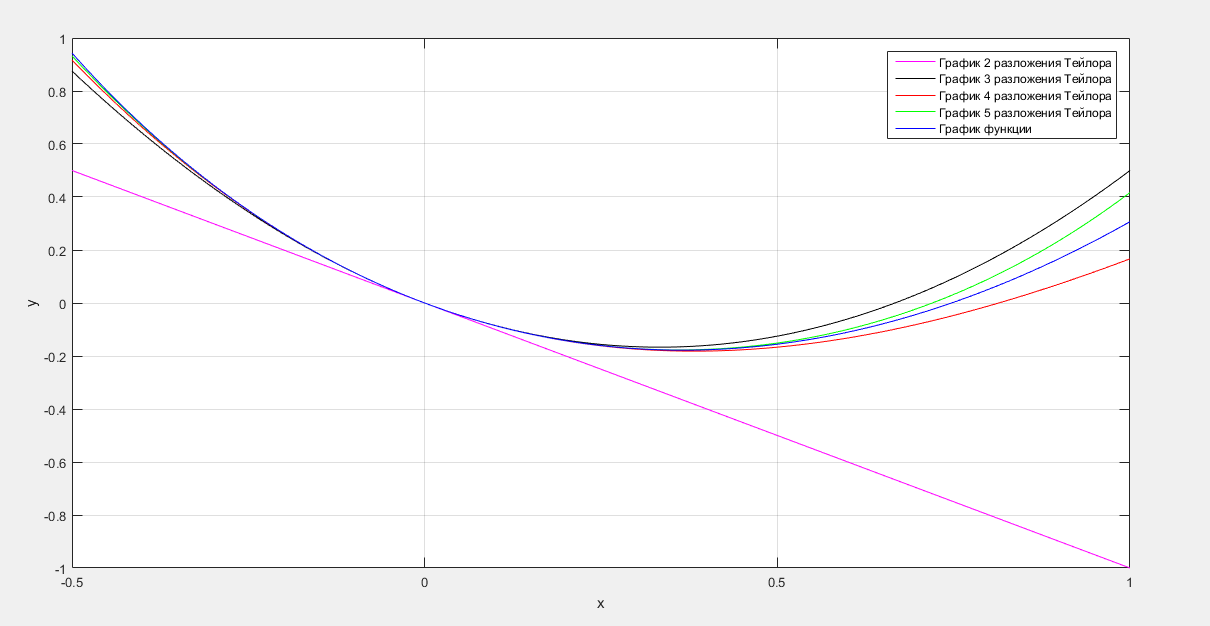
Рисунок

11. Преобразование из символьного в числовое значение



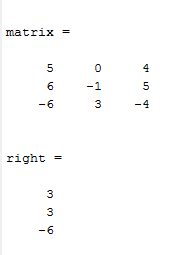
Рисунок

12. Графическое изображение разложения в ряд Тейлора



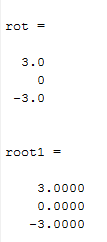
Рисунок

13. Матрица, которая задалась случайно



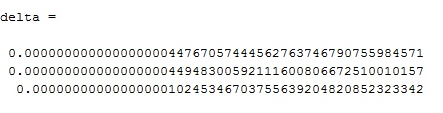
Рисунок

14. Система линейных уравнений с случайными коэффициентами



Рисунок

15. Их относительная погрешность



Рисунок