Санкт-Петербургский политехнический университет Петра Великого

Институт прикладной математики и механики

**Кафедра «Высшая школа теоретической механики»**

**ОТЧЕТНАЯ РАБОТА**

по дисциплине «Математическое моделирование»

Выполнил

студент гр.3630103/90001 <*подпись*> М.А. Бенюх

Руководитель

преподаватель <*подпись*> Р.Л. Лапин

«\_11\_» \_\_декабря\_\_\_ 2019\_ г.

Санкт-Петербург

2020

**СОДЕРЖАНИЕ**

Введение………………………………………………………………………..….4

**1. Задание №1: Работа с функцией одной переменной**………….....5

1. Цель работы, исходные данные…………………………..…………………...5

2. Построение функций Q(X), R(X) и P(X) в одном окне в разных системах координат……………………………………………………………………….....6

3. Построение кусочно-заданной функции…………………………………...…8

4. Построение производных каждой функции вместе с самой функцией в одном окне в трех различных системах координат………………………………………………………………..………...9

5. Построение Q(x), R(x) и P(x) в одной системе координат………………………………………………………………………...11

6. Вывод………………………………………………………………………......12

**2. Задание №2: Работа с поверхностями 1-го и 2-го порядка**….13

1. Цель работы, исходные данные…………………………..…………………12

2. Построение поверхности второго порядка (цилиндр) с выбранной точкой обзора (каркасным построением или пленочным) ……………………………………………………………………...12

3. Построение поверхности F(x,y,z) в одном окне в нескольких системах координат с разным типом отображения ……………………………………...14

4. Построение поверхности F(x,y,z) разными цветовыми стилями….………………………………………………………………..………16

5. Построение в одном окне сферы и её сечения плоскостью (с разными точками обзора)………………………………………………………………….18

6. Вывод…………………………………………………………………………..20

**3. Задание №3: Матрицы и полиномы**………………………………..21

1. Цель работы, исходные данные…………………………..………………….21

2. Создание матрицы 6×6, заполненной случайными числами в определенном диапазоне...……………………………………………………………………….21

3. Математические операции с матрицей………………………………………22

4. Построение характеристического полинома для матрицы а4, нахождение его корней, первой и второй производных, построение графика этого полинома с отображением особых точек………………………………………22

5. Вывод графика полинома с его особыми точками в отдельном окне в одной системе координат……………………………………………………………….22

6. Построение графиков в одном окне с несколькими системами координат.23

7. Вывод…………………………………………………………………………..25

**4. Задание №4: Символьные вычисления**……………………………27

1. Цель работы, исходные данные……………………………………………...27

2. Объявление всех символьных переменных…………………………………27

3. Символьное задание функции……………..…………………………………27

4. Построение графика функции с помощью «ezplot»………………………...28

5. Построение графика функции с помощью «plot»…………………………..28

6. Вывод…………………………………………………………………………..29

**Введение:**

MATLAB (от matrix laboratory) — пакет прикладных программ для решения задач технических вычислений. Пакет используют более миллиона инженерных и научных работников, он работает на большинстве современных операционных систем Особенностями пакета являются ориентация на работу с матрицами (отсюда неофициальный лозунг языка — Think vectorized, т.е. Думай векторами), готовые реализации ряда полезных алгоритмов, богатые возможности визуализации данных и взаимодействия с другими языками.

Язык MATLAB был создан в конце 1970-ых годов; он предназначался для работы с библиотеками численных методов, написанными на FORTRAN, не зная самого языка. Язык быстро обрел популярность среди людей, занимающихся прикладной математикой.

MATLAB позволяет нам строить графики с любой степенью точности. В лабораторной работе №1 проводится построение графиков функций одной переменной с их характерными точками, а также графиков их производны, используя пакет MATLAB.

Также в данном пакете доступна работа с поверхностями 1-го и 2-го порядков. Благодаря чему в лабораторной работе №2 производится их построение и работа над их внешними характеристиками.

Также в MATLAB можно производить различные операции над матрицами и полиномами. В лабораторной работе №3 демонстрируется работа с ними: создание матрицы и полинома, построение графиков и нахождение особых точек.

А в лабораторной работе №4 применяются символьные вычисления, которые также входят в инструментную базу данного пакета.

**1. Задание №1: Работа с функцией одной переменной**

Цель работы:

Построить данные функции в одной системе координат, построить кусочно-заданную функцию, найти корни каждой функции и локальные минимумы и максимумы, построить производные каждой функции вместе с самой функцией в одном окне в трех различных системах координат.

Исходные данные:

Функции:

Ход работы:

1. Построение функций Q(X), R(X) и P(X) в одном окне в разных системах координат, выделение особых точек.
2. Создаем функции Q(X), R(X) и P(X).

function y = fQ(x)

y=cos(x.^2).\*x;

end

function y = fR(x)

y=abs(sin(x).\*x+cos(x).\*x.^2+atan(x).\*x.^3);

end

function y = fF(x)

y=(cos(x.^2).\*x)./(abs(sin(x).\*x+cos(x).\*x.^2+atan(x).\*x.^3));

end

1. Задаем интервал для x. Задаём функции Q(X), R(X) и P(X) на нём.

x = 10:0.001:10;

y\_q=fQ(x);

y\_r=fR(x);

y\_f=fF(x);

1. Создадим функции, обратные Q(X), R(X) и P(X), для поиска максимумов:

function y = fQ2(x)

y=-(cos(x.^2).\*x);

end

function y = fR2(x)

y=-(abs(sin(x).\*x+cos(x).\*x.^2+atan(x).\*x.^3));

end

function y = fR2(x)

y=-(abs(sin(x).\*x+cos(x).\*x.^2+atan(x).\*x.^3));

end

1. Найдем корни функций в окрестности определённых точек:

* Корень P(x):

x\_f\_z1=fzero('fF',1.5);

x\_f\_z2=fzero('fF',1);

x\_f\_z3=fzero('fF',-6);

y\_f\_z1=fQ(x\_f\_z1)./(fR(x\_f\_z1)~=0);

y\_f\_z2=fQ(x\_f\_z2)./(fR(x\_f\_z2)~=0);

y\_f\_z3=fQ(x\_f\_z3)./(fR(x\_f\_z3)~=0);

* Корень Q(x):

x\_q\_z1=fzero('fQ',6);

x\_q\_z2=fzero('fQ',0);

x\_q\_z3=fzero('fQ',-5);

y\_q\_z1=fQ(x\_q\_z1);

y\_q\_z2=fQ(x\_q\_z2);

y\_q\_z3=fQ(x\_q\_z3);

* Корень R(x):

x\_r\_z1=fzero('fR',5);

x\_r\_z2=fzero('fR',0);

x\_r\_z3=fzero('fR',-6);

y\_r\_z1=fR(x\_r\_z1);

y\_r\_z2=fR(x\_r\_z2);

y\_r\_z3=fR(x\_r\_z3);

1. Найдем экстремумы Q(X), R(X) и P(X) c помощью функции «fminbnd»:

* Для P(x):

Min:

x\_f\_min1=fminbnd ('fF',0,4);

x\_f\_min2=fminbnd ('fF',-2,2);

x\_f\_min3=fminbnd ('fF',4,6);

y\_f\_min1=fQ(x\_f\_min1)./(fR(x\_f\_min1)~=0);

y\_f\_min2=fQ(x\_f\_min2)./(fR(x\_f\_min2)~=0);

y\_f\_min3=fQ(x\_f\_min3)./(fR(x\_f\_min3)~=0);

Max:

x\_f\_max1=fminbnd ('fF2',0,4);

x\_f\_max2=fminbnd ('fF2',-4,0);

x\_f\_max3=fminbnd ('fF2',4,6);

y\_f\_max1=fQ(x\_f\_max1)./(fR(x\_f\_max1)~=0);

y\_f\_max2=fQ(x\_f\_max2)./(fR(x\_f\_max2)~=0);

y\_f\_max3=fQ(x\_f\_max3)./(fR(x\_f\_max3)~=0);

* Для Q(x):

Min:

x\_q\_min1=fminbnd ('fQ',0,4);

x\_q\_min2=fminbnd ('fQ',-4,0);

x\_q\_min3=fminbnd ('fQ',4,6);

y\_q\_min1=fQ(x\_q\_min1);

y\_q\_min2=fQ(x\_q\_min2);

y\_q\_min3=fQ(x\_q\_min3);

Max:

x\_q\_max1=fminbnd ('fQ2',0,4);

x\_q\_max2=fminbnd ('fQ2',-4,0);

x\_q\_max3=fminbnd ('fQ2',4,6);

y\_q\_max1=fQ2(x\_q\_max1);

y\_q\_max2=fQ2(x\_q\_max2);

y\_q\_max3=fQ2(x\_q\_max3);

* Для R(x):

Min:

x\_r\_min1=fminbnd ('fR',0,4);

x\_r\_min2=fminbnd ('fR',-4,0);

x\_r\_min3=fminbnd ('fR',4,6);

y\_r\_min1=fR(x\_r\_min1);

y\_r\_min2=fR(x\_r\_min2);

y\_r\_min3=fR(x\_r\_min3);

Max:

x\_r\_max1=fminbnd ('fR2',0,4);

x\_r\_max2=fminbnd ('fR2',-4,0);

x\_r\_max3=fminbnd ('fR2',4,6);

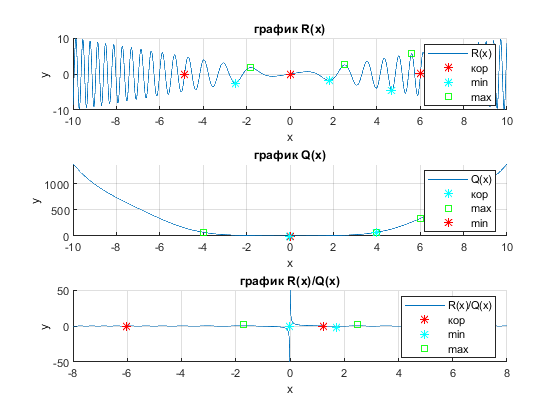
y\_r\_max1=fR2(x\_r\_max1);

y\_r\_max2=fR2(x\_r\_max2);

y\_r\_max3=fR2(x\_r\_max3);

1. Создаем окно для построения графиков функций, используя «figure» . С помощью «subplot» создадим систему координат для каждой функции. Отметим на этих графиках особые очки данных функций. (Рис. 1)
2. figure
3. subplot(3,1,1)
4. hold on
5. grid on
6. title('график R(x)')
7. plot(x,y\_q)
8. plot(x\_q\_z1,y\_q\_z1,'r\*')
9. plot(x\_q\_min1,y\_q\_min1,'c\*')
10. plot(x\_q\_max1,-y\_q\_max1,'gs')

13. plot(x\_q\_z2,y\_q\_z2,'r\*')
14. plot(x\_q\_z3,y\_q\_z3,'r\*')
15. xlabel('x')
16. ylabel('y')
17. plot(x\_q\_min2,y\_q\_min2,'c\*')
18. plot(x\_q\_min3,y\_q\_min3,'c\*')
20. plot(x\_q\_max2,-y\_q\_max2,'gs')
21. plot(x\_q\_max3,-y\_q\_max3,'gs')
23. legend('R(x)','кор','min','max')
25. subplot(3,1,2)
26. hold on
27. grid on
28. plot(x,y\_r)
29. plot(x\_r\_min1,y\_r\_min1,'c\*')
30. plot(x\_r\_max1,-y\_r\_max1,'gs')
32. plot(x\_r\_z1,y\_r\_z1,'r\*')
33. plot(x\_r\_z2,y\_r\_z2,'r\*')
34. plot(x\_r\_z3,y\_r\_z3,'r\*')
36. plot(x\_r\_min2,y\_r\_min2,'c\*')
37. plot(x\_r\_min3,y\_r\_min3,'c\*')
38. xlabel('x')
39. ylabel('y')
40. plot(x\_r\_max2,-y\_r\_max2,'gs')
41. plot(x\_r\_max3,-y\_r\_max3,'gs')
42. title('график Q(x)')
43. legend('Q(x)','кор','max','min')
45. subplot(3,1,3)
46. hold on
47. grid on
48. plot(x,y\_f)
49. plot(x\_f\_z1,y\_f\_z1,'r\*')
50. plot(x\_f\_min1,y\_f\_min1,'c\*')
51. plot(x\_f\_max1,y\_f\_max1,'gs')
53. axis([-8 8 -50 50])
54. plot(x\_f\_z2,y\_f\_z2,'r\*')
55. plot(x\_f\_z3,y\_f\_z3,'r\*')
56. xlabel('x')
57. ylabel('y')
58. plot(x\_f\_min2,y\_f\_min2,'c\*')
59. plot(x\_f\_min3,y\_f\_min3,'c\*')
60. title('график R(x)/Q(x)')
61. plot(x\_f\_max2,y\_f\_max2,'gs')
62. plot(x\_f\_max3,y\_f\_max3,'gs')
63. legend('R(x)/Q(x)','кор','min','max')



*Рис. 1 Графики различных функций*

*Рис. 1. Функции Q(x), P(x), R(x) в разных системах координат в одном окне*

1. Построение кусочно-заданной функции P(x)= F(x), x С (-inf;a), Q(x), x C (a;b), R(x), x C (b;inf). (рис.2)
2. Создаём массивы значений для ‘кусков’ функций F(x), Q(x), R(x)(область определения):

x\_r=-10:0.001:0;

x\_q=0:0.001:5;

x\_f=5:0.001:10;

Строим кусочно-заданной функцию из фукций F(x), Q(x), R(x) ограниченных областью определения. Приближаем график для более корректного отображения графиков функций axis([-4 8 -10 20]) ,добавляем легенду legend('Q(x)','R(x)','Q(x)/R(x)').

figure

hold on

grid on

plot(x\_q,y\_q2)

plot(x\_r,y\_r2)

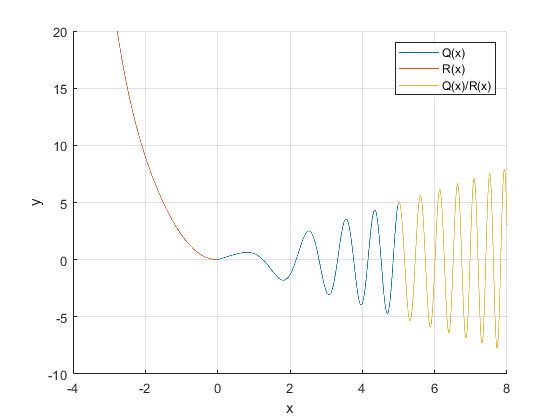
plot(x\_f,y\_f2)

xlabel('x')

ylabel('y')

legend('Q(x)','R(x)','Q(x)/R(x)')

axis([-4 8 -10 20])



*Рис. 2 график кусочно-заданной функции*

1. Построение производных функций совместносте с самой функцией в одном окне в трех различных системах координат:
2. Полкчаем значения дифференцируя фуекции

dq=diff(fQ(x));

dr=diff(fR(x));

df=diff(fF(x));

1. Строим в трех системах координат функции F(x), Q(x), R(x) и поверх них строим производные этих функций (рис. 3):

figure

subplot(3,1,1)

hold on

grid on

xlabel('x')

ylabel('y')

plot(x,y\_q)

plot(x(1:20000),dq)

axis([-8 -2 -2 2])

title('гр.Производной Q(x) и график Q(x)')

subplot(3,1,2)

hold on

grid on

plot(x,y\_r)

plot(x(1:20000),dr)

axis([-6 6 -20 50])

xlabel('x')

title('гр.Производной R(x) и график R(x)')

ylabel('y')

subplot(3,1,3)

hold on

grid on

plot(x,y\_f)

plot(x(1:20000),df)

axis([-0.2 0.2 -40 20])

title('гр.Производной Q(x)/R(x) и график Q(x)/R(x)')

xlabel('x')

ylabel('y')

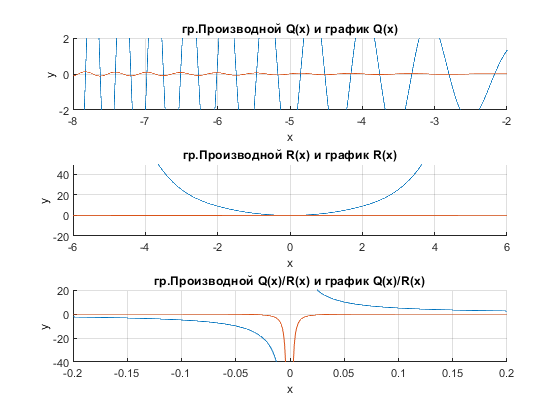


Рис.3 Графики производных различных функций

1. Построение Q(x), R(x) и P(x) в одной системе координат(Рис.4):

figure

hold on

grid on

plot(x,y\_q)

plot(x,y\_r)

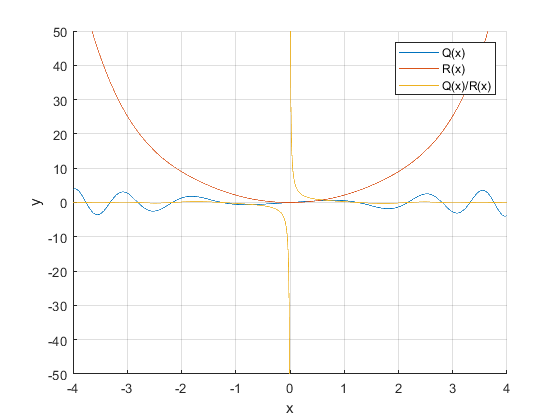
plot(x,y\_f)

xlabel('x')

ylabel('y')

axis([-4 4 -50 50])

legend('Q(x)','R(x)','Q(x)/R(x)')



*Рис. 4 Отображение 3х функций в одной системе координат*

Вывод:

Были построены функции в одной и разных системах координат, оформлены графики, найдены локальные экстремумы и корни функций, продифференцированы функции и построены кусочно-заданные функции с помощью среды MatLab.

**2. Задание №2: Работа с поверхностями 1-го и 2-го порядка.**

Цель работы: построить цилиндр с выбранной точкой обзора каркасным построением, построить цилиндр в одном окне в нескольких системах координат с разным типом отображения (каркасная, пленочная с каркасом, пленочная), построить цилиндр с разными цветовыми стилями, построить в одном окне цилиндр и её сечение плоскостью с разными точками обзора.

Исходные данные:

Параметрическое уравнение цилиндра:

Уравнение плоскости:

(𝐿): 𝐴∙𝑥+𝐵∙𝑦+𝐶∙𝑧+𝐷=0

Константы:

x0 = -2;

y0 = -11;

z0 = -10;

A = 17;

B = -12;

C = 17;

D = 1;

r = 20;

Ход работы:

1. Построение поверхности второго порядка (цилиндра) с выбранной точкой обзора (каркасным построением или пленочным) :
2. Задаём интервалы для x, y, f, где f - образующая угловая переменная:

x = -36:5:10 ;

y = -11:5:29 ;

f = (0:10:360)\*pi/180 ;

1. С помощью функции [X, Y] = meshgrid(x, y) формируем массивы X и Y, которые определяют координаты узлов прямоугольника, задаваемого векторами x и y. Этот прямоугольник задает область определения функции от двух переменных, которую можно построить в виде 3D-поверхности.

[X1,Y1] = meshgrid(x,y);

[Fi, Z] = meshgrid(f,z);

1. Задаем поверхность F(x,y,z) и строим ее c помощью ‘surf(X,Y,Z)’ в точке обзора (40,25) с помощью view(40,25) (рис.1)

X = r.\*cos(Fi);

Y = r.\*cos(Fi);

figure

view(40,25)

grid on

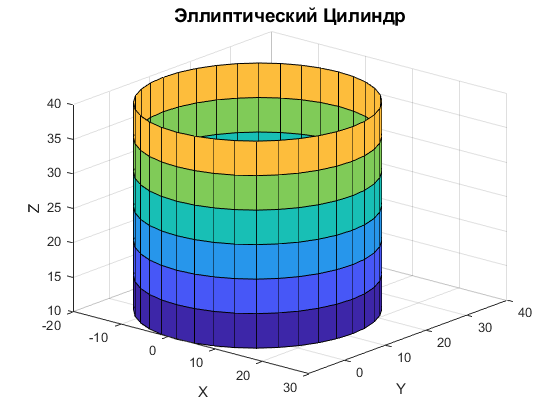
hold on

rotate3d on

surf(X-x0,Y-y0,Z-z0);

title('Эллиптический Цилиндр','FontSize', 14)

xlabel('X'); ylabel('Y'); zlabel('Z');



*Рис. 1 Поверхность F(x,y,z)*

1. Построение поверхности F(x,y,z) в одном окне в нескольких системах координат с разным типом отображения. (рис.2)
2. Создаём окно для графиков, определяем параметры координатных осей:

figure

grid on

hold on

rotate3d on

1. С помощью ‘surf(X,Y,Z)’ строим поверхность F(x,y,z) с пленочным типом отображения с каркасом.

figure

view(40,25)

grid on

hold on

rotate3d on

subplot(2,2,1)

view(40,25)

grid on

hold on

rotate3d on

alpha(0.95)

surf(X-x0,Y-y0,Z-z0);

title('залитая цветом каркасная поверхность(surf)')

xlabel('X'); ylabel('Y'); zlabel('Z');

1. С помощью ‘surf1(X,Y,Z)’ строим поверхность F(x,y,z) с пленочным сглаженным типом отображения (освещённая поверхность):

subplot(2,2,2)

view(40,25)

grid on

hold on

rotate3d on

surfl(X-x0,Y-y0,Z-z0);

title('освещенная поверхность(surfl)')

xlabel('X'); ylabel('Y'); zlabel('Z');

1. С помощью ‘mesh(X,Y,Z)’ строим поверхность F(x,y,z) с каркасным типом отображения.

subplot(2,2,3)

view(40,25)

grid on

hold on

rotate3d on

mesh(X-x0,Y-y0,Z-z0);

title('каркасная поверхность(mesh)')

xlabel('X'); ylabel('Y'); zlabel('Z');

1. С помощью ‘plot3’ строим поверхность F(x,y,z), используя линии и точки в трёхмерном пространстве:

subplot(2,2,4)

view(40,25)

grid on

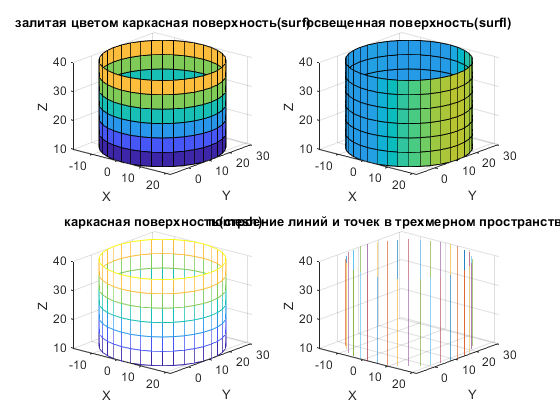
hold on

rotate3d on

plot3(X-x0,Y-y0,Z-z0);

title('построение линий и точек в трехмерном пространстве(plot3)')

xlabel('X'); ylabel('Y'); zlabel('Z');



*Рис. 2 Построение цилиндров разными типами сетки*

1. Построение поверхности F(x,y,z) разными цветовыми стилями. (рис.3)
2. Обозначим первое построение за ‘ax1’ и с помощью ‘colormap’ построим поверхность F(x,y,z) в цветовом стиле ‘hot(8)’; с помощью функции ‘shading interp’ сделаем, чтобы заливка была плавной:

figure

subplot(2,2,1)

grid on

surf(X-x0,Y-y0,Z-z0);

view(40,25)

colormap(subplot(2,2,1),'hot(8)')

shading interp %плавная заливка

xlabel('X'); ylabel('Y'); zlabel('Z');

title('палитра:hot,без линий')

1. С помощью «'FaceColor','k','EdgeColor','b'» построим

поверхность F(x,y,z) с черными полями и синим каркасом:

subplot(2,2,2)

grid on

rotate3d on

surf(X-x0,Y-y0,Z-z0,'FaceColor','k','EdgeColor','b' )

view(40,25)

xlabel('X'); ylabel('Y'); zlabel('Z');

title('черныйые поля, синие линии' )

1. С помощью «'FaceColor','y','EdgeColor','g'» построим поверхность F(x,y,z) с желтыми полями и зелёным каркасом:

subplot(2,2,3)

grid on

surf(X-x0,Y-y0,Z-z0,'FaceColor','y','EdgeColor','g' )

view(40,25)

xlabel('X'); ylabel('Y'); zlabel('Z');

title('желтые поля, зеленые линии' )

1. Обозначим первое построение за ‘ax4’ и с помощью ‘colormap’ построим поверхность F(x,y,z) в цветовом стиле ‘parula’:

subplot(2,2,4)

grid on

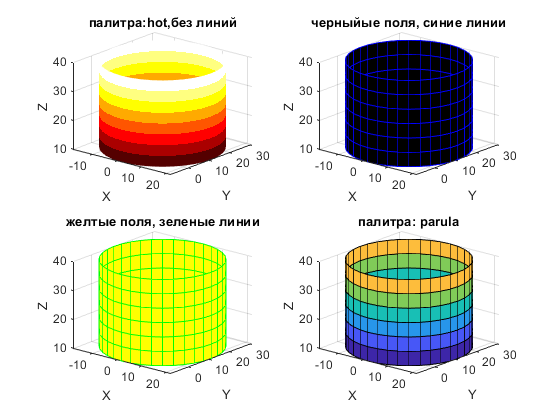
surf(X-x0,Y-y0,Z-z0);

view(40,25)

colormap(subplot(2,2,4), 'parula')

xlabel('X'); ylabel('Y'); zlabel('Z');

title('палитра: parula' )



*Рис. 3 Построение цилиндров в различных цветах*

1. Построение в одном окне цилиндры и его сечения плоскостью (с разными точками обзора):
2. Задаем плоскость

Z1 = (A.\*(X1-x0) + B.\*(Y1-y0) + D). /(-C);

1. Строим в четырех системах координат поверхность F(x,y,z) и плоскость c разными точками обзора (рис. 4):

Figure

subplot(2,2,1)

grid on

hold on

rotate3d on

surf(X-35,Y+15,Z+10, 'FaceColor','w','EdgeColor','m')

surf(X1-16,Y1+9.5,Z1-10, 'FaceColor','k','EdgeColor','m')

view([-141, 20])

xlabel('X'); ylabel('Y'); zlabel('Z');

title('обзор 1' )

subplot(2,2,2)

grid on

hold on

rotate3d on

surf(X-35,Y+15,Z+10, 'FaceColor','w','EdgeColor','m')

surf(X1-16,Y1+9.5,Z1-10, 'FaceColor','k','EdgeColor','m')

view([138, 35])

xlabel('X'); ylabel('Y'); zlabel('Z');

title('обзор 2' )

subplot(2,2,3)

grid on

hold on

rotate3d on

surf(X-35,Y+15,Z+10, 'FaceColor','w','EdgeColor','m')

surf(X1-16,Y1+9.5,Z1-10, 'FaceColor','k','EdgeColor','m')

view([-130, 2])

xlabel('X'); ylabel('Y'); zlabel('Z');

title('обзор 3' )

subplot(2,2,4)

grid on

hold on

rotate3d on

surf(X-35,Y+15,Z+10, 'FaceColor','w','EdgeColor','m')

surf(X1-16,Y1+9.5,Z1-10, 'FaceColor','k','EdgeColor','m')

view([9, 9])

xlabel('X'); ylabel('Y'); zlabel('Z');

title('обзор 4' )

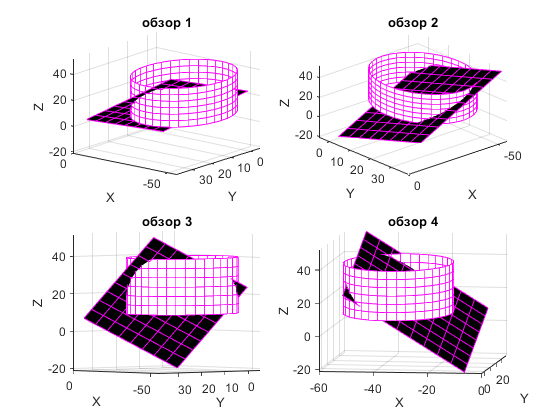


Рис.4 Поверхность F(x,y,z) и её сечение плоскостью в разных точках обзора.

Вывод:

Был построен цилиндр с разными способами отображения, в разных цветовых стилях и были построены плоскость и поверхность на одном графике c разными точками обзора, где плоскость является сечением поверхности 2-го порядка.

**3. Задание №3: Матрицы и полиномы**

Цель работы: Построить матрицу 6×6, заполненную случайными числами в определенном диапазоне; для данной матрицы проделать 4 математических операции; для получившейся матрицы построить характеристический полином, определить его корни и первые две производные; построить график полинома с отображением на нем корней, экстремумов и точек перегиба; построить графики в одном окне с несколькими системами координат: полином с отображением корней, экстремумов и точек перегиба; первую производную с корнями; вторую производную с корнями.

Исходные данные:

* A – матрица 6×6;

a = 2.4;

b = 5,1 ;

(где a – правая граница диапазона заполнения матрицы, а b – левая граница)

* Математические операции:

1. sqrt ;
2. exp ;
3. ;

Ход работы:

1. Создание матрицы 6×6, заполненной случайными числами в определенном диапазоне.
2. Зададим значение границ диапазона:

a = 2.4;

b = 5,1;

1. Создадим матрицу 6×6, заполненную случайными числами в диапазоне от ‘a’ до ‘b’:

A = a + (b-a).\*rand(6,6) ;

(уравнение взято из документации Matlab)

(данные из «Workspace»)

4.0014 4.2559 4.0537 1.9284 4.2046 3.4700

2.6416 4.3627 4.2169 3.2941 2.4183 1.9990

4.9208 2.1727 2.4937 4.9551 3.3214 2.0375

1.6240 3.2632 3.9469 2.7254 4.0167 2.4270

3.0795 3.1041 3.8584 3.6070 4.7073 4.5266

2.8736 3.8267 2.0854 2.3057 4.9534 2.4154

1. Математические операции с матрицей:

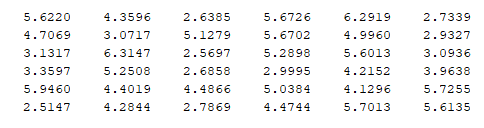
a1 = sqrt(A)

a2 = exp(a1)

a3 = a2.^2

a4 = a3./13

Получаем матрицу ‘а4’ 6×6 вида:

(данные из «Workspace)  


*Рис. 1 Полученная рандомная матица*

* *Все дальнейшие операции будут производиться с матрицей a4.*

1. Построение графика полинома с отображением на нем корней, экстремумов, точек перегиба. (рис.2)
2. Для получившейся рандомной матрицы построить характеристический полином, нахождение экстремумов:

p = poly(E);

X = -50:0.00002:50;

y = polyval(p,X);  
  
r1 = roots(p1);

r2 = roots(p2);

y1 = polyval(p, r1);

y2 = polyval(p, r2);

1. Поиск корней данного полинома.

r = roots(p);

plot(r,zeros(6,1), 'k\*')

1. Поиск первых дух производных данного полинома:

p1 = polyder(p);

p2 = polyder(p1);

y\_1 = polyval(p1,X);

y\_2 = polyval(p2,X);

1. Построение графика полинома:

figure

hold on

grid on

title('График полинома,экстремумы и перегибы')

plot(X,y)

plot(r,zeros(6,1), 'k\*')

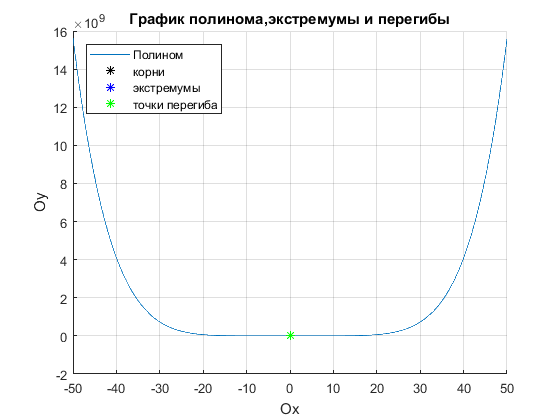
plot(r1,y1, 'b\*') %экстремумы

plot(r2,y2, 'g\*')

xlabel('Ox')

ylabel('Oy')

legend('Полином','корни', 'экстремумы', 'точки перегиба' ,'Location','southwest','Location','northwest')



*Рис. 2.1 График полинома*

1. Приближение графика полинома:

figure

hold on

grid on

title('График полинома,экстремумы и перегибы')

axis([0.0003 0.0018 -0.00000000000000004 0.0000000000000001])

plot(X,y)

plot(r,zeros(6,1), 'k\*')

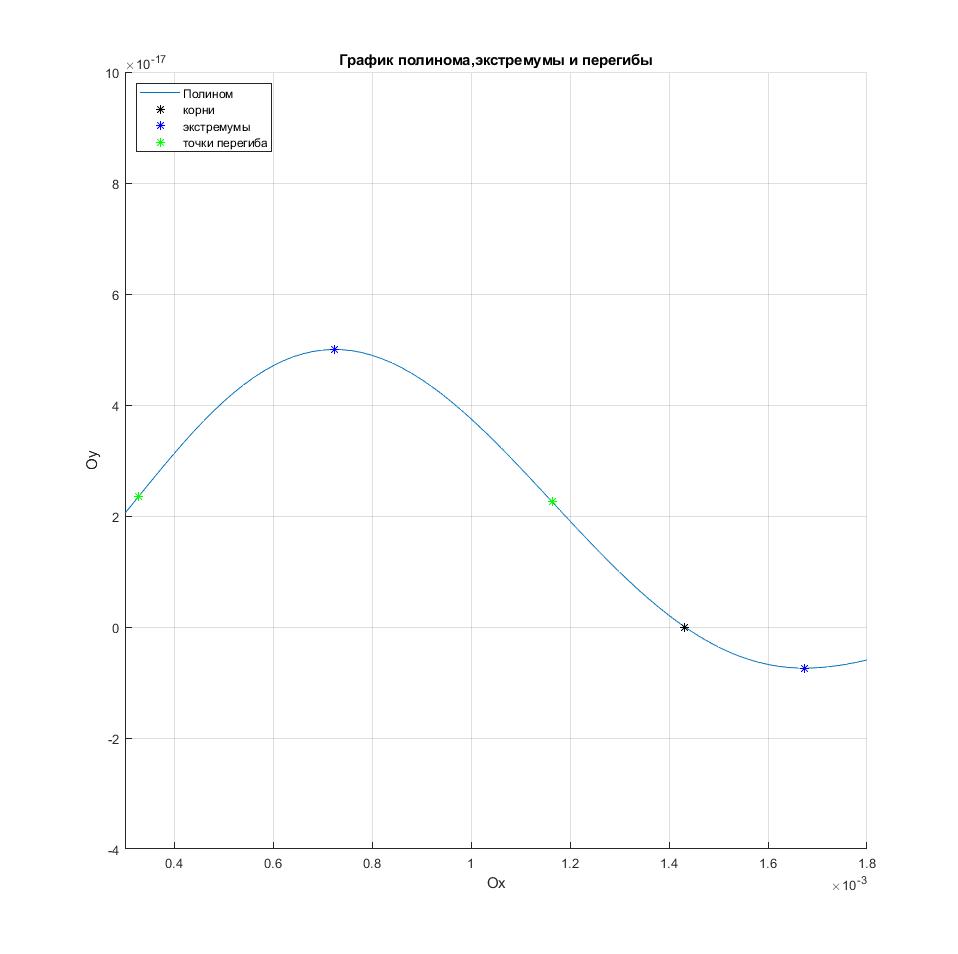
plot(r1,y1, 'b\*')

plot(r2,y2, 'g\*')

xlabel('Ox')

ylabel('Oy')

legend('Полином','корни', 'экстремумы', 'точки перегиба' ,'Location','southwest','Location','northwest')



*Рис. 2.2 График полинома*

1. Построение графиков в одном окне с несколькими системами координат: Полином с отображением корней, экстремумов и точек перегиба; первую производную с корнями; вторую производную с корнями (рис.3)
2. Построение графика полинома с нанесёнными на него очками перегиба экстремумами и корнями:

figure

subplot(3,1,1)

hold on

grid on

title('График полинома,экстремумы и перегибы')

axis([-0.02 0.015 -0.00000000000000001 0.000000000000001])

plot(X,y)

plot(r,zeros(6,1), 'k\*')

plot(r1,y1, 'b\*') %экстремумы

plot(r2,y2, 'g\*')

xlabel('Ox')

ylabel('Oy')

legend('Полином','корни', 'экстремумы', 'точки перегиба' ,'Location','southwest','Location','northwest')

axis([-0.001 0.003 -0.0000000000000001 0.0000000000000002])

1. Построение графика 1 производной с нанесёнными на него корнями:

subplot(3,1,2)

hold on

grid on

title('1я производная')

plot(X,y\_1)

plot(r1,zeros(5,1), 'b\*')

xlabel('Ox')

ylabel('Oy')

legend('1я производная','корни','Location','southwest','Location','northwest')

axis([-0.001 0.003 -0.0000000000000001 0.0000000000000002])

1. Построение графика 2 производной с нанесёнными на него корнями::

subplot(2,2,3)

grid on

surf(X-x0,Y-y0,Z-z0,'FaceColor','y','EdgeColor','g' )

view(40,25)

xlabel('X'); ylabel('Y'); zlabel('Z');

title('желтые поля, зеленые линии' )

1. Обозначим первое построение за ‘ax4’ и с помощью ‘colormap’ построим поверхность F(x,y,z) в цветовом стиле ‘parula’:

subplot(3,1,3)

hold on

grid on

title('2я производная')

% axis([-50 50 -5.\*10^6 7.\*10^6])

axis([-0.001 0.003 -0.0000000000000001 0.0000000000000002])

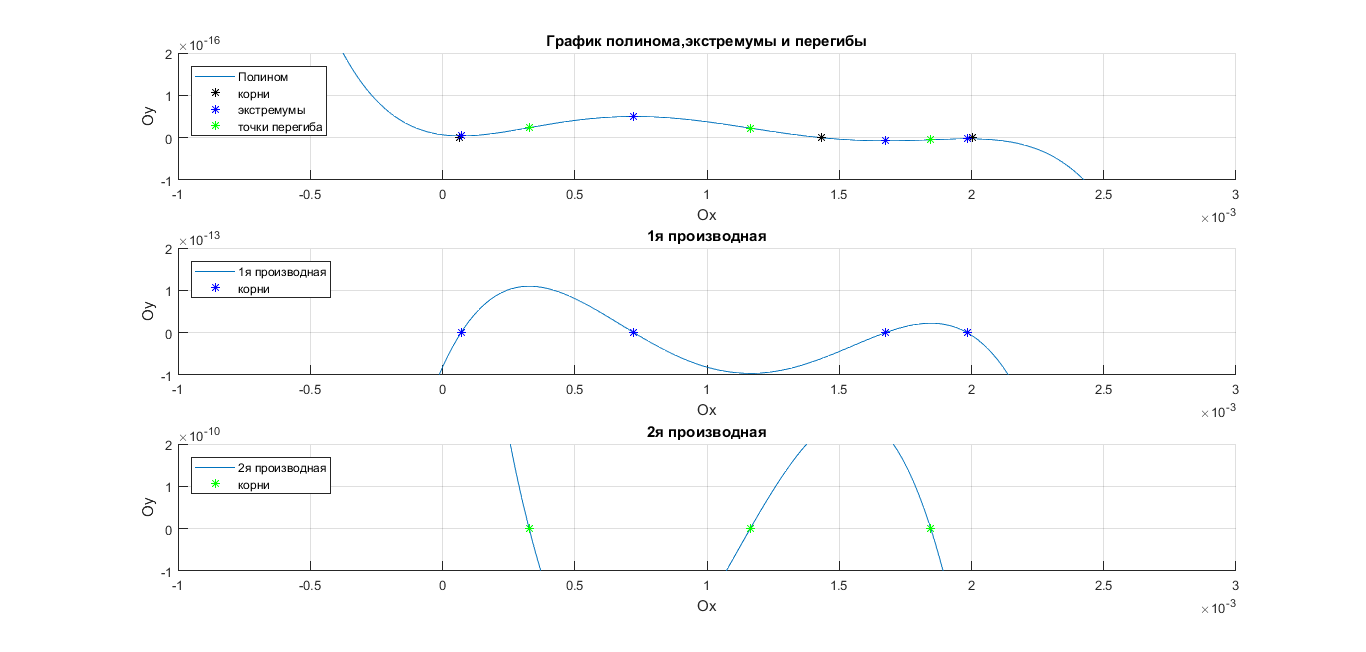
plot(X,y\_2)

plot(r2,zeros(4,1), 'g\*')

xlabel('Ox')

ylabel('Oy')

legend('2я производная','корни','Location','southwest','Location','northwest')



*Рис. 3 график полинома; его 1,2 производной*

Вывод:

Была создана матрица 6\*6, с который были проделаны четыре математические операции; для полученной матрицы был построен характеристический полином, найдены его корни и первые две производные; далее был построен график данного полинома, на котором были отмечены его особые точки; затем были построены графики производных данного полинома и отмечены их корни.

**4. Задание №4: Символьные вычисления.**

Цель работы:

1. Выбрать одну из функций F(x), Q(x), R(x).
2. Данную функцию задать символьно в Matlab
3. С помощью инструментов символьных вычислений для выбранной функции проделать:
   1. Сделать подстановку x = 2p^2 - 7p
   2. Упростить выражение
   3. Разложить на множители
   4. Умножить на функцию sin(p+2)(p^3-8p^2-10)(p+3)
   5. Упростить выражение
   6. Построить график функции с помощью команд ezplot и plot.

Исходные данные:

1. исходная функция:
2. набор операций:

* подстановка: x = 2p^2 - 7p ;
* упрощение;
* разложение на множители;
* умножение на функцию;
* построение графика двумя способами.

Ход работы:

1. С помощью инструментов символьных вычислений для выбранной функции проделать: подстановку, упрощение выражения, разложение на множители, умножение на функцию, повторное ее упрочение. На основе полученных данных построить график при помощи ezplot и plot:
2. Символьно задаем f(x):

syms x y p f g

%1

f = log(x.^2.\*abs(sin(x.^5).\*sin(x.^5)))

1. Производим подстановку (2\*p^2 - 7\*p).

f1 = subs(f0, x, 2\*p^2 - 7\*p)

1. Упрощаем выражение

f2 = simplify(f1)

1. Раскладываем на множители

f3 = factor(f2)

1. Умножаем на функцию (p^3 - 8\*p^2 - 10)\*(p+3)

g = sin(p+2).\*(p^3 - 8\*p^2 - 10).\*(p+3)

f4 = f3.\*g

1. Упрощаем выражение

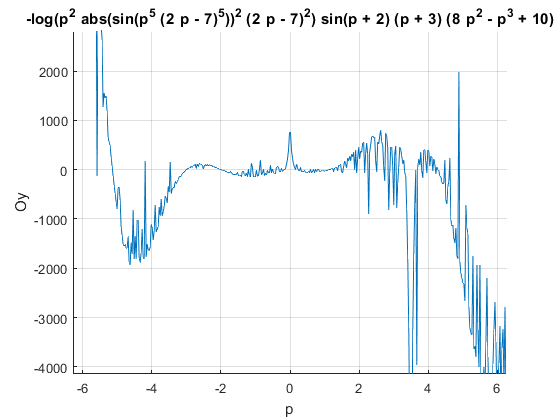
f5 = simplify(f4)

1. Построение графика функции при помощи ezplot(рис.1)

figure

hold on

grid on

ezplot(f5)

*Рис. 1 График полученный выполнением команды ezplot*

1. Построение графика функции при помощи plot(рис.2)

function Y = R(x)

Y = -log(x.^2.\*abs(sin(x.^5.\*(2.\*x - 7).^5)).^2.\*(2.\*x - 7).^2).\*sin(x + 2).\*(x + 3).\*(- x.^3 + 8.\*x.^2 + 10);

End

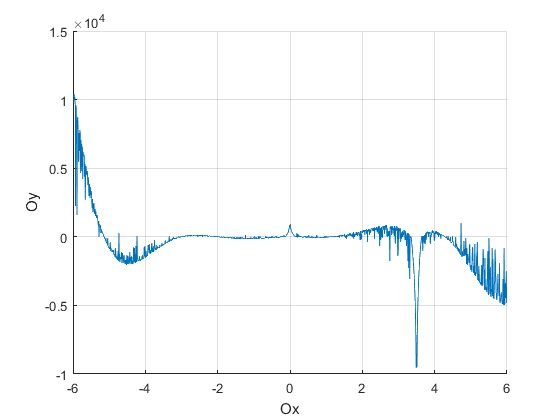
Y = R(x);

x = -6:0.01:6;

figure

hold on

grid on

plot(x,Y);

*Рис. 2 График полученный выполнением команды plot*

1. Вывод: с помощью инструментов символьных вычислений для выбранной функции, были проделаны следующие действия проделать: подстановку, упрощение выражения, разложение на множители, умножение на функцию, повторное ее упрочение. На основе полученных данных был построен график при помощи ezplot и plot.