РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО

ЛАБОРАТОРНОЙ РАБОТЕ №7

дисциплина: Научное программирование

Студентка: Голос Елизавета Сергеевна
Группа: НПМмд-02-20
Ст. билет № 1032202186

Цель работы
Научиться строить различные виды графиков: параметрические, неявных функций, в полярных координатах. Обучиться работе с комплексными числами, изображать их на координатной плоскости.

Ход работы

Параметрические графики

В самом начале работы включим журналирование. Построим график трёх периодов циклоиды радиуса 2. Для этого определим параметр как вектор в некотором диапазоне, затем вычислим x и y. Выполение команд показано на рисунке 1.

```
>> diary on
>> t = linspace (0,6*pi,50);
>> r = 2;
>> x = r*(t-sin(t));
>> y = r*(1-cos(t));
>> plot (x,y)
>> axis('equal');
>> axis([0 12*pi 0 4])
>> savefig cycloid.pdf
```

Рис.1 Команды для построения графика

Полученный график изображен на рисунке 2.

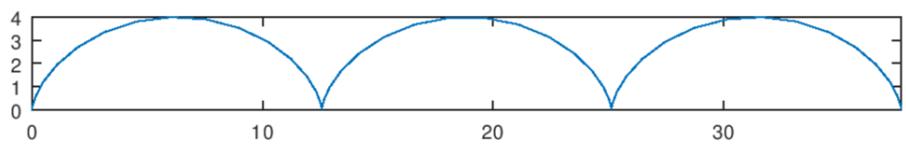

Рис.2 График циклоиды

Полярные координаты

Графики в полярных координатах строятся аналогичным образом. Построим улитку Паскаля. Ход работы показан на рисунке 3

```
>> theta = linspace (0,2*pi,100);
>> r = 1-2*sin(theta);
>> x = r.*cos(theta);
>> y = r.*sin(theta);
>> plot (x,y)
>> print -dpdf limacon.pdf
>> print -dpng limacon.png
```

Рис.3 Построение графика в полярных координатах

Полученный график можно увидеть на рисунке 4.

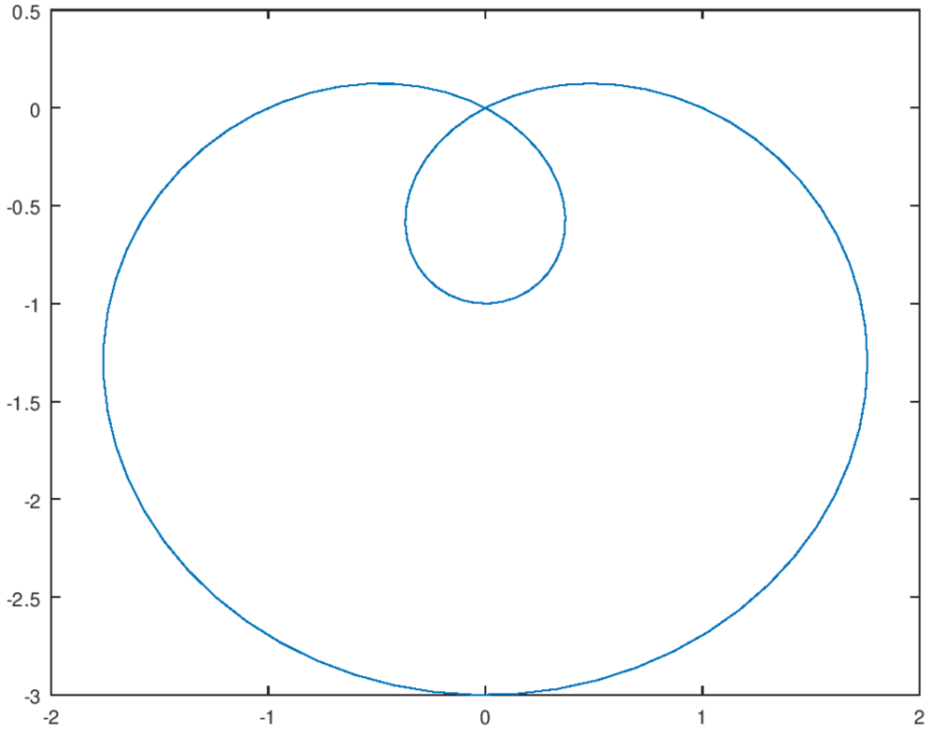

Рис.4 Улитка Паскаля.

Более того, можно построить данный график в полярных осях. Команды показаны на рисунке 5.

```
>> theta = linspace (0,2*pi,50);
>> r = 1-2*sin(theta);
```

```
>> polar(theta,r)
>> print -dpdf limacon-polar.pdf
>> print -dpng limacon-polar.png
```

Рис.5 Реализация улитки Паскаля в полярных осях.

А сам график показан на рисунке 6.

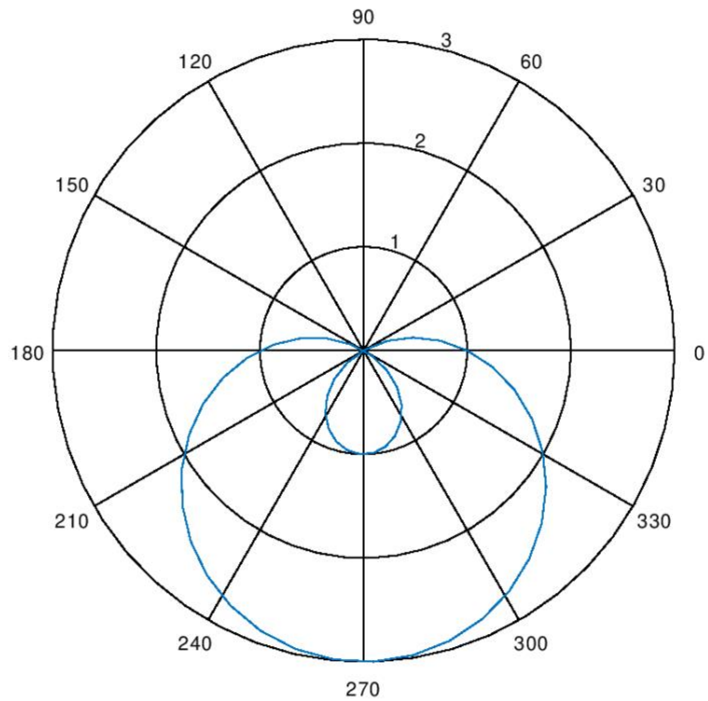

Рис.6 График улитки Паскаля в полярных осях.

Графики неявных функций

Следует построить неявно определённую функцию с помощью ezplot. Зададим график функции, используя лямбда-функцию, как показано на рисунке 7.

```
>> f = @(x,y) -x.^2-x.*y+x+y.^2-y-1
f =

@(x, y) -x .^ 2 - x .* y + x + y .^ 2 - y - 1

>> ezplot(f)
>> print -dpdf impl1.pdf
```

Рис.7 Реализация неявно определенной функции

После чего построим ее график. См. рисунок 8.

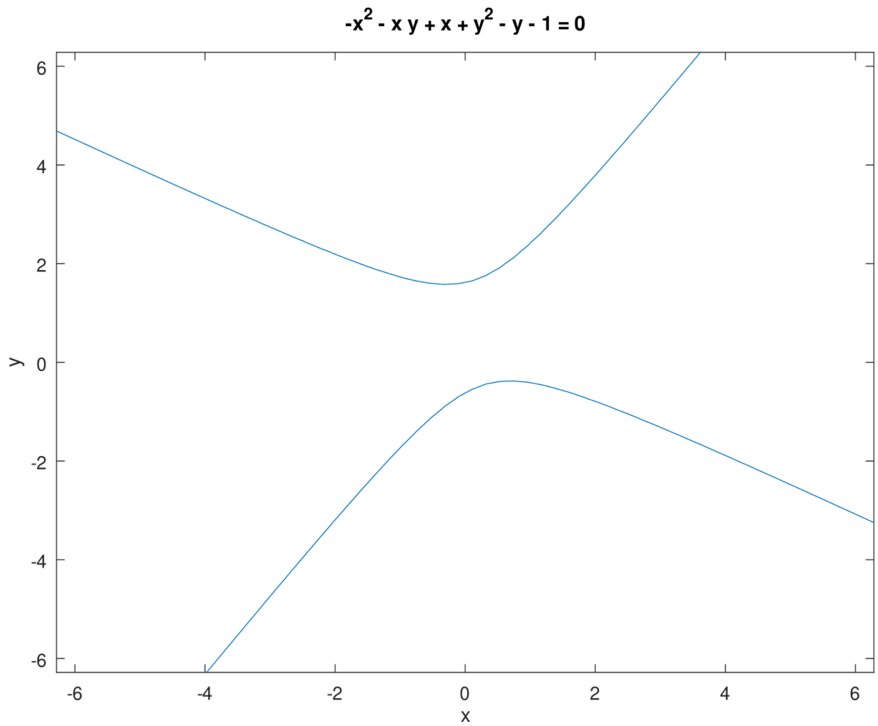

Рис.8 Реализация неявно определенной функции

Найдём уравнение касательной к некоторой окружности. Сначала построим круг, используя лямбда-функцию. Далее по правилу дифференцирования найдём уравнение касательной и изобразим ее на графике. См. рис. 9

```
>> f = @(x,y) (x-2).^2+y.^2-25;
>> ezplot(f, [-6 10 -8 8])
>> x = [-6:10];
>> y = 3/4*x+19/4;
>> hold on
>> plot (x,y,'r--')
>> print -dpdf impl2.pdf
```

Рис.9 Построение касательной к окружности

Полученный график можно увидеть на рисунке 10.

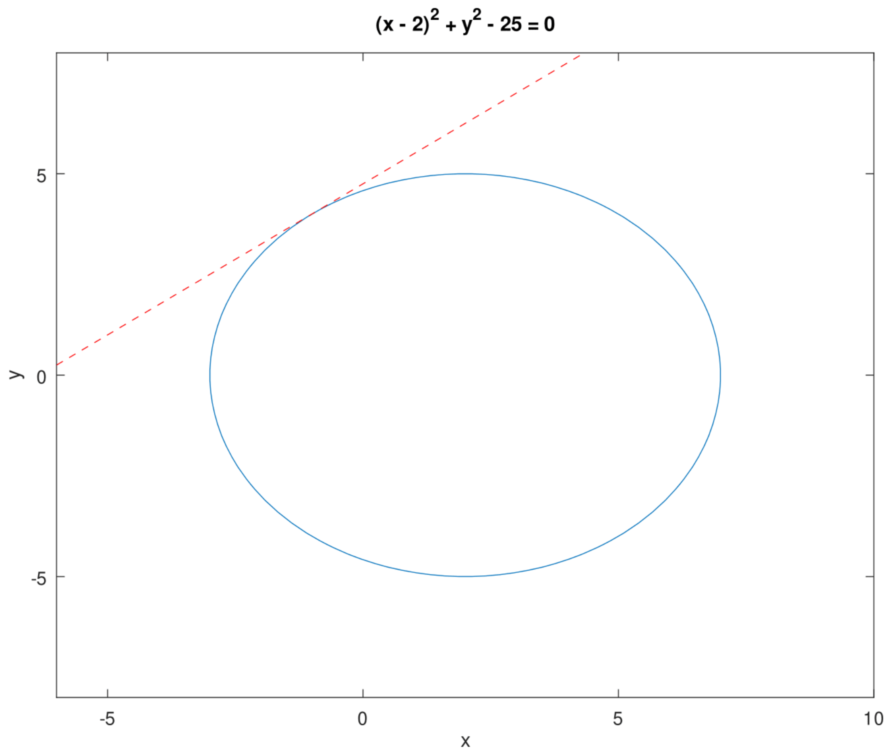

Рис.10 График касательной к окружности

Комплексные числа

Зададим два комплексных числа и запишем основные арифметические операции с ними: сложение,вычитание, умножение, деление. См. рисунок 11.

```
>> z1 = 1+2*i;
>> z2 = 2-3*i;
>> z1+z2
ans =  3 - 1i
>> z1-z2
ans = -1 + 5i
>> z1*z2
ans =  8 + 1i
>> z1/z2
ans = -0.30769 + 0.53846i
>> clf
```

Рис.11 Действия с комплексными числами

Построим графики в комплексной плоскости, используя команду compass, используя команды, показанные на рисунке 12.

```
>> z1 = 1+2*i;
>> z2 = 2-3*i;
>> compass (z1,'b')
>> compass (z1,'b')
>> hold on
>> compass (z2,'r')
>> compass (z1+z2,'k--')
>> legend('z_1','z_2','z_1+z_2')
>> print -dpdf complex.pdf
```

Рис.12 Построение графиков в комплексной плоскости

Изображение графиков показано на рисунке 13.

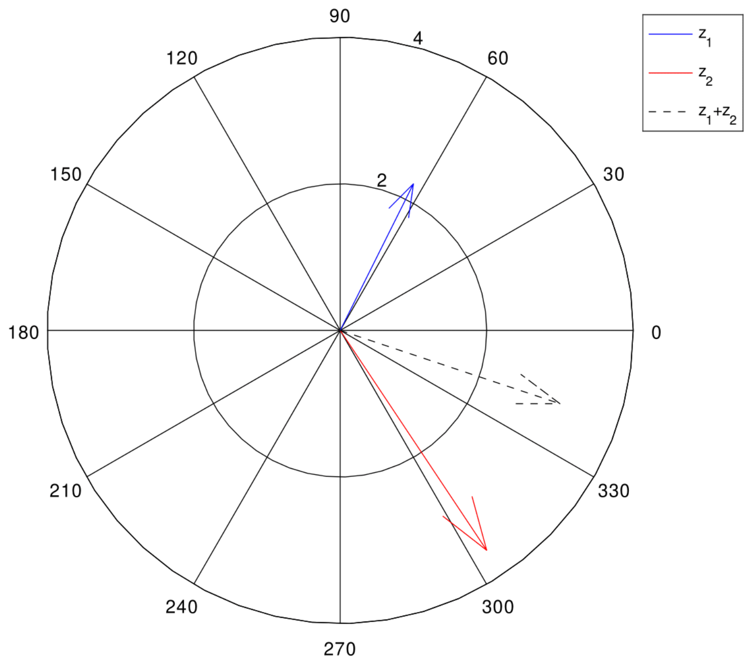

Рис.13 Графики в комплексной плоскости

Иногда мы можем получить странные результаты вывода программы. При вычислении корня третьей степени из -8, мы ожидаем ответ -2, но получаем другое число. Это объясняется тем, что Octave возвращает тот ответ, у которого меньший аргумент. Для того, чтобы получить -2, мы должны использовать команду nthroot, как показано на рисунке 14.

```
>> (-8)^(1/3)
ans =  1.0000 + 1.7321i
>> ans^3
ans = -8.0000e+00 + 9.7972e-16i
>> nthroot(-8,3)
ans = -2
```

Рис.14 Извлечение кубического корня из отрицательного числа.

Специальные функции

Построим гамма-функцию Г(x+1) и n! на одном графике, как показано на рисунке 15.

```
>> n=[0:1:5];
>> x = linspace(-5,5,500);
>> plot(n,factorial(n),'*',x,gamma(x+1))
>> clf
```

Рис.15 Построение гамма функции и факториала

Изображение показано на рисунке 16.

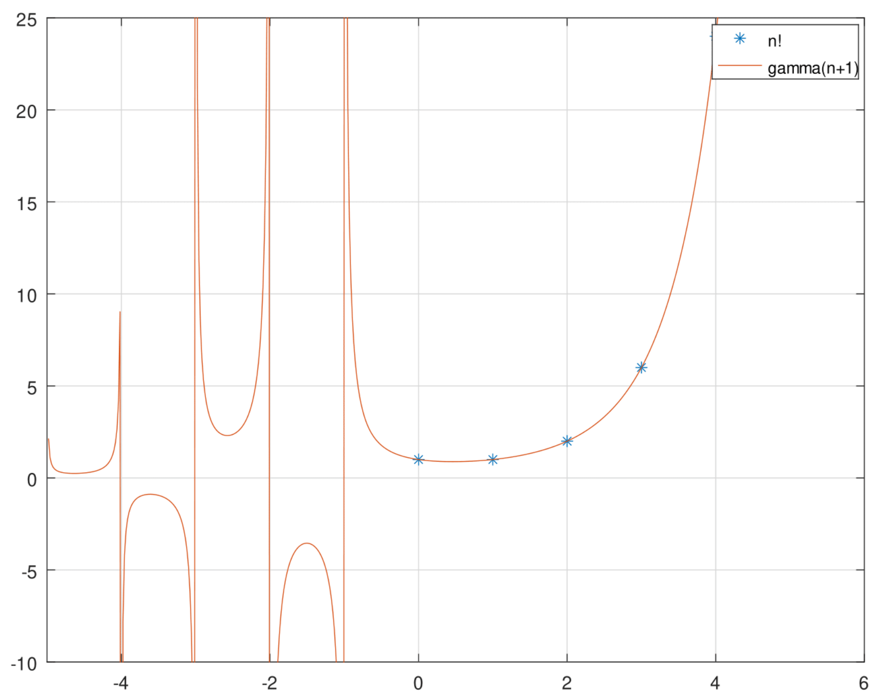

Рис.16 Изображение гамма функции и факториала

Разделив область значения на отдельные интервалы, можно убрать артефакты вычислений. Для этого следует выполнить команды, указанные на рисунке 17.

```
>> x1 = linspace(-5,-4,500);
>> x2 = linspace(-4,-3,500);
>> x3 = linspace(-3,-2,500);
>> x4 = linspace(-2,-1,500);
>> x5 = linspace(-1,5,500);
>> plot(x1,gamma(x1+1))
>> hold on
>> plot(x2,gamma(x2+1))
>> plot(x3,gamma(x3+1))
>> plot(x4,gamma(x4+1))
>> plot(x5,gamma(x5+1))
>> axis([-5 6 -10 25]);
>> plot(n,factorial(n),'*')
>> legend('n!',"\\Gamma(n+1)")
>> print -dpdf gamma2.pdf
>> diary off
```

Рис.17 Разделение на интервалы

После проведения вышеуказанных действий, построим график. См. рисунок 18

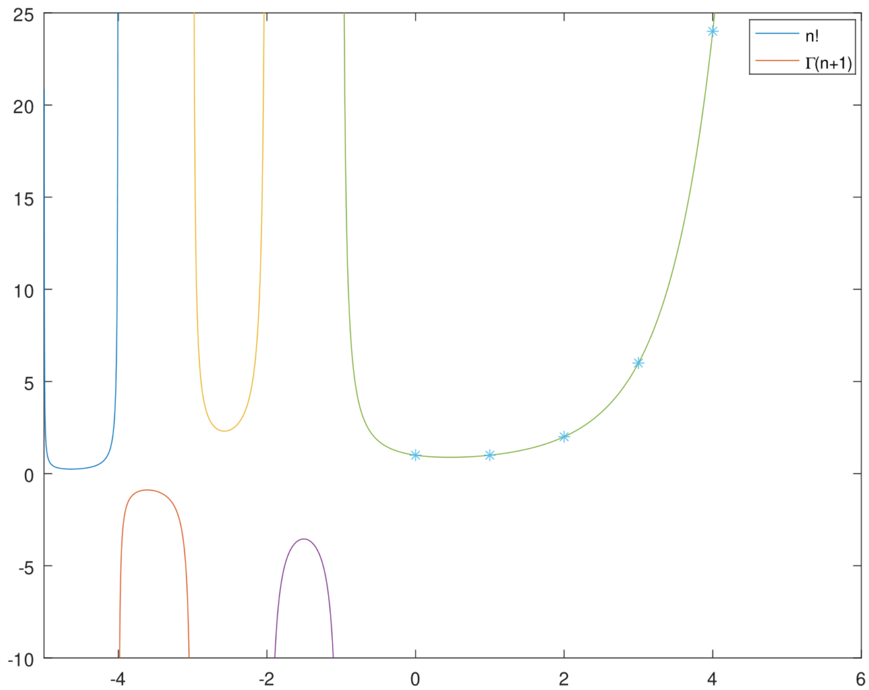

Рис.18 График гамма функции и факторила после устранения артефактов

Вывод

В ходе выполнения данной работы я научилась строить различные виды графиков: параметрические, неявных функций, в полярных координатах. Также поработала с комплексными числами, научилась изображать их на координатной плоскости. А также построила гамма-функцию и график факториала.