
Графики

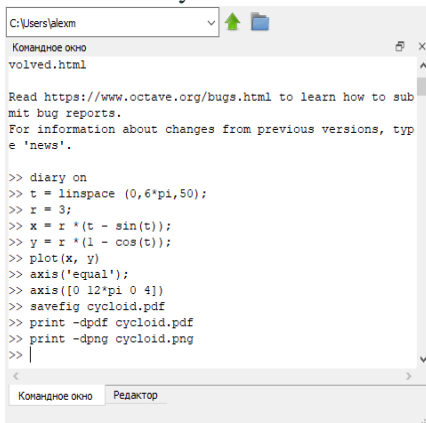
Милёхин Александр НПМмд-02-21

Цель работы

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

Параметрические графики

Включим журналирование. Построим график трёх периодов циклоиды радиуса 2. Для этого определим параметр как вектор в некотором диапазоне, затем вычислим x и y .



```
C:\Users\alexm
Командное окно
volved.html

Read https://www.octave.org/bugs.html to learn how to submit bug reports.
For information about changes from previous versions, type 'news'.

>> diary on
>> t = linspace(0,6*pi,50);
>> r = 3;
>> x = r*(t - sin(t));
>> y = r*(1 - cos(t));
>> plot(x, y)
>> axis('equal');
>> axis([0 12*pi 0 4])
>> savefig cycloid.pdf
>> print -dpdf cycloid.pdf
>> print -dpng cycloid.png
>> |
```

Figure 1: Команды для построения графика

Параметрические графики

Получим график циклоиды.

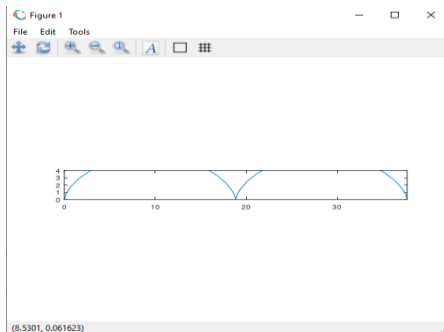
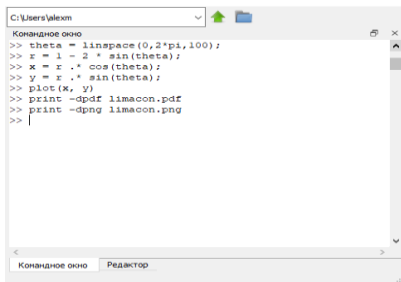


Figure 2: График циклоиды

Полярные координаты

Графики в полярных координатах строятся аналогичным образом.

Построим улитку Паскаля.



```
C:\Users\alexm
Конандное окно
>> 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
>>
```

Figure 3: Построение графика в полярных координатах

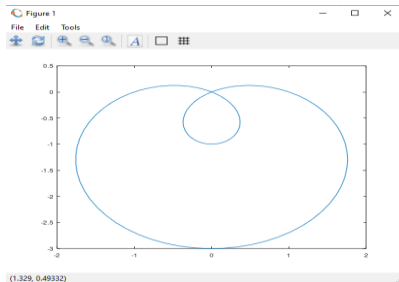
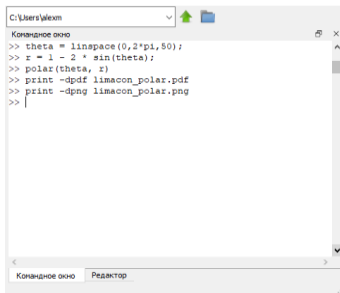


Figure 4: Улитка Паскаля

Полярные координаты

Построим данный график в полярных осях.



```
C:\Users\alexm
Командное окно
>> theta = linspace(0,2*pi,50);
>> r = 1 - 2 * sin(theta);
>> polar(theta, r)
>> print -dpdf limacon_polar.pdf
>> print -dpng limacon_polar.png
>>
```

Figure 5: Реализация улитки Паскаля в полярных осях

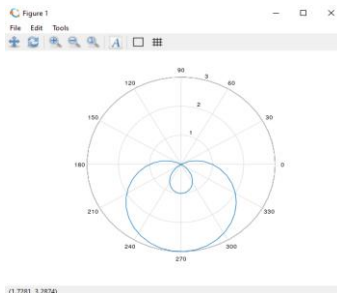
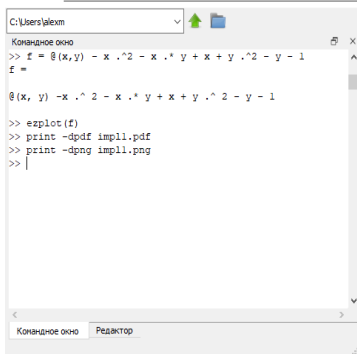


Figure 6: График улитки Паскаля в полярных осях

Графики неявных функций

Построим неявно определённую функцию с помощью `ezplot`.

Используя лямбда-функцию зададим график.



```
C:\Users\alexm
Командное окно
>> 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
>> print -dpng impl1.png
>> |
```

Figure 7: Реализация неявно определенной функции

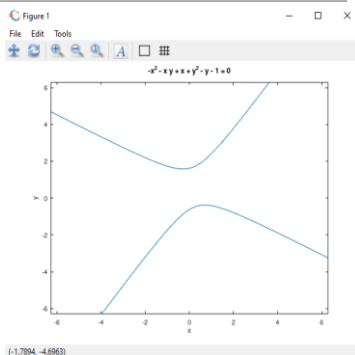
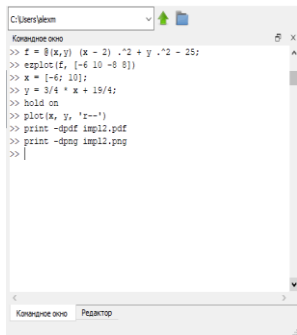


Figure 8: График неявно определенной функции

Графики неявных функций

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



```
C:\Users\alexm
Кандидное окно
>> 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
>> print -dpng impl2.png
>>
```

Figure 9: Построение касательной к окружности

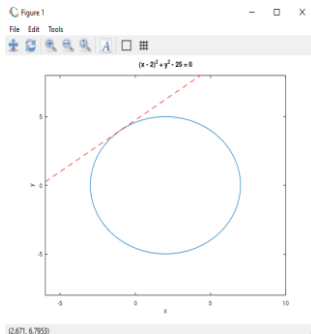
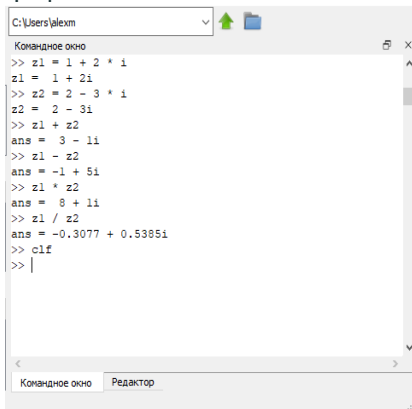


Figure 10: График касательной к окружности

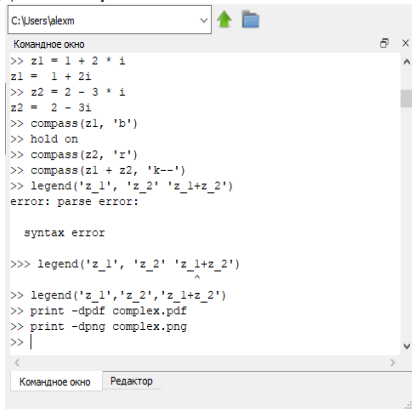
Комплексные числа

Зададим два комплексных числа и запишем основные арифметические операции с ними: сложение, вычитание, умножение, деление. Построим графики в комплексной плоскости командой compass.



```
C:\Users\alexm
Командное окно
>> z1 = 1 + 2 * i
z1 = 1 + 2i
>> z2 = 2 - 3 * i
z2 = 2 - 3i
>> z1 + z2
ans = 3 - 1i
>> z1 - z2
ans = -1 + 5i
>> z1 * z2
ans = 8 + 1i
>> z1 / z2
ans = -0.3077 + 0.5385i
>> clf
>> |
```

Figure 11: Действия с комплексными числами



```
C:\Users\alexm
Командное окно
>> z1 = 1 + 2 * i
z1 = 1 + 2i
>> z2 = 2 - 3 * i
z2 = 2 - 3i
>> compass(z1, 'b')
>> hold on
>> compass(z2, 'r')
>> compass(z1 + z2, 'k--')
>> legend('z_1', 'z_2' 'z_1+z_2')
error: parse error:

syntax error

>>> legend('z_1', 'z_2' 'z_1+z_2')
>> legend('z_1','z_2','z_1+z_2')
>> print -dpdf complex.pdf
>> print -dpng complex.png
>> |
```

Figure 12: Построение графиков в комплексной плоскости

Комплексные числа

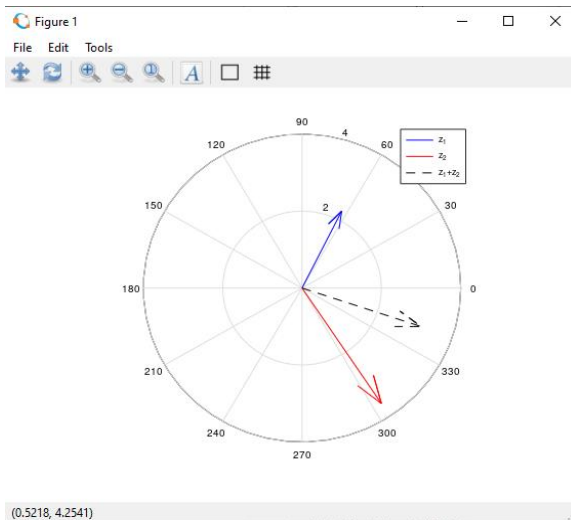
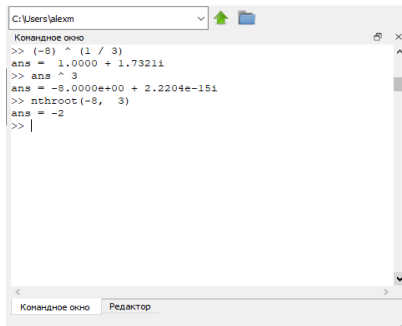


Figure 13: Графики в комплексной плоскости

Комплексные числа

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`.



```
C:\Users\alexm
Командное окно
>> (-8) ^ (1 / 3)
ans = 1.0000 + 1.7321i
>> ans ^ 3
ans = -8.0000e+00 + 2.2204e-15i
>> nthroot(-8, 3)
ans = -2
>> |
```

Figure 14: Извлечение кубического корня из отрицательного числа

Специальные функции

Построим гамма-функцию $\Gamma(x+1)$ и $n!$ на одном графике.

```
C:\Users\alexm
Командное окно
>> n = [0:1:5];
>> clf
>> n = [0:1:5];
>> x = linspace(-5,5,500);
>> plot(n,factorial(n),'*',x,gamma(x+1))
>> axis([-5 6 -10 25])
>> grid on;
>> legend('n!','gamma(x+1)')
>> print -dpdf gamma.pdf
>> print -dpng gamma.png
>> |
```

Figure 15: Построение гамма функции и факториала

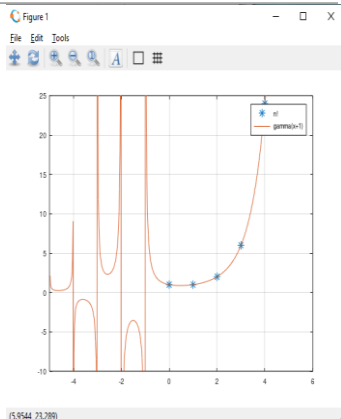


Figure 16: Изображение гамма-функции и факториала

Специальные функции

Разделив область значения на отдельные интервалы, можно ввести команды и убрать артефакты вычислений.

```
C:\Users\alexm
Командное окно
>> clf
>> 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
>> print -png gamma2.png
>> diary off
```

Figure 17: Разделение на интервалы

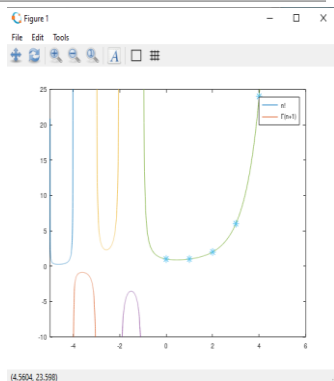


Figure 18: График гамма-функции и факториала после устранения артефактов

Результаты лабораторной работы

Я научился строить в Octave различные виды графиков: параметрические, неявных функций, в полярных координатах. Также поработал с комплексными числами, научился изображать их на координатной плоскости; построил гамма-функцию и график факториала.

Спасибо за внимание