Графики

Включите журналирование работы.

```
>> diary on
```

Параметрические графики

Параметрические уравнения для циклоиды:

$$x = r(t - \sin(t)), y = r(1 - \cos(t)).$$

Построим график трёх периодов циклоиды радиуса 2. Решение. Поскольку период 2π, нам нужно, чтобы параметр был в пределах $0 \leq t \leq 6\pi$ для трёх полных циклов. Определим параметр t как вектор в этом диапазоне, затем мы вычислим x и y.

```
>> 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
>> print -dpdf cycloid.pdf
>> print -dpng cycloid.png
```

Полярные координаты

Графики в полярных координатах строятся аналогичным образом. Для функции
$$r = f(\vartheta)$$
мы начинаем с определения независимой переменной ϑ, затем вычисляем r. Чтобы построить график, мы вычислим x и y, используем стандартное преобразование координат

$$x = r\cos(\vartheta), y = r\sin(\vartheta),$$

затем построим график в осях xy.

Построим улитку Паскаля

$$r = 1 - 2\sin(\vartheta).$$

```
>> 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
```

Также можно построить функцию

$$r = f(\vartheta)$$

в полярных осях, используя команду `polar`.

```
>> theta = linspace(0,2*pi,50);
>> r = 1-2*sin(theta);
>> polar(theta,r)
>> print -dpdf limacon-polar.pdf
>> print -dpng limacon-polar.png
```

Графики неявных функций

Пусть нужно построить функцию, неявно определённую уравнением вида

$$f(x,y)=0.$$

Самый простой способ сделать это в Octave – с помощью команды `ezplot`.

Построим кривую, определяемую уравнением

$$-x^2-xy+x+y^2-y=1.$$

Чтобы определить функцию в виде $f(x,y)=0$, вычтем 1 из обеих частей уравнения. Зададим функцию в виде λ-функции.

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

Построим график.

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

Найдём уравнение касательной к графику окружности

$$(x-2)^2+y^2=25$$

в точке $(-1,4)$. Построим график окружности и касательной.

Чтобы построить круг, сначала определим его как функцию вида $f(x,y)=0$. Зададим функцию в виде λ-функции.

```
>> f = @(x,y) (x-2).^2+y.^2-25;
```

Центр круга находится в точке $(2,0)$, а радиус равен 5. Зададим оси нашего графика так, чтобы они несколько превосходили окружность.

```
>> ezplot(f, [-6 10 -8 8])
```

Используя правило дифференцирования неявной функции, найдём

$$y' = \frac{2-x}{y}.$$

В точке $(-1, 4)$ имеем

$$y'|_{(-1,4)} = \frac{2-(-1)}{4} = \frac{3}{4}.$$

Таким образом, уравнение касательной линии будет иметь вид:

$$y - 4 = \frac{3}{4}(x-(-1)) \to y = \frac{3}{4}x + \frac{19}{4}.$$

Построим график.

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

Комплексные числа

Пусть $z_1 = 1 + 2i$, $z_2 = 2 - 3i$. Запишем основные арифметические операции с этими числами.

```
>> 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
```

Мы можем построить график в комплексной плоскости, используя команду `compass`.

Пусть $z_1 = 1+2i, z_2 = 2-3i$. Построим графики z_1, z_2 и z_1+z+2 в комплексной плоскости.

```
>> clf
>> 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
```

Иногда Octave может неожиданно выдать странные результаты для комплексных чисел. Например, вычислим $\sqrt[3]{-8}$:

```
>> (-8)^(1/3)
ans =  1.0000 + 1.7321i
```

Ожидался ответ -2, мы также можем легко проверить, что куб данного ответа действительно равен -8 (по крайней мере, до некоторой незначительной ошибки округления):

```
>> ans^3
ans = -8.0000e+00 + 2.2204e-15i
```

На самом деле существует три кубических корня из -8, и по умолчанию Octave возвращает тот, у которого наименьший аргумент (угол). Если нам просто нужен действительный корень, мы можем использовать команду `nthroot`.

```
>> nthroot(-8,3)
ans = -2
```

Специальные функции

В Octave доступно много специальных функций, таких как функции Бесселя (`bessel`), функция ошибок (`erf`), гамма-функция (`gamma`). Гамма-функция определяется как

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt.$$

Это расширение факториала, поскольку для натуральных чисел n гамма-функция удовлетворяет соотношению

$$\Gamma(n) = (n-1)!$$

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

Зададим значения аргумента $x \in [-5, 5]$ для гамма-функции и $n = 0, 1, 2, 3, 4, 5$ для факториала.

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

Обратите внимание на вертикальные асимптоты на графике в районе отрицательных целых чисел. Они не являются истинной частью графика. Это артефакты вычисления. Если мы хотим их устранить, мы должны разделить область значений на отдельные интервалы. Это даёт более точный график.

```
>> 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
```

Включите журналирование.

```
>> diary off
```

Задание

- Сделайте отчёт по лабораторной работе в формате Markdown.
- В качестве ответа просьба предоставить отчёты в 3 форматах: `pdf`, `docx` и `md` (в архиве, поскольку он должен содержать скриншоты, Makefile и т.д.)