

13.3 Библиотека *plots*

Кроме основных функций построения графиков `plot` и `plot3d` Maple V release 4 содержит две библиотеки, значительно расширяющие графические возможности пакета.

Библиотека plots – позволяет строить разнообразные двухмерные и трехмерные графики для различных математических объектов.

Библиотека plottools – содержит функции для построения различных графических объектов (примитивов).

Библиотека `plots` содержит 48 функций для построения различных типов графиков (см. таблицу).

При таком количестве функций подробное рассмотрение формата каждой функции займет достаточно много места, поэтому ограничимся кратким описанием каждой команды и приведем небольшие примеры, иллюстрирующие работу команд.

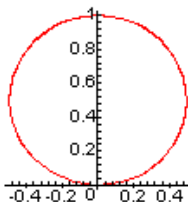
Заметим, что для использования функций, содержащихся в библиотеке `plots`, применимы обычные приемы подключения библиотеки целиком или отдельных функций при помощи оператора `with`.

animate3d – анимация трехмерных поверхностей.

```
> plots[animate3d](cos(t*x)*sin(t*y), x=-Pi..Pi,
y=-Pi..Pi, t=1..2);
```

changecoords – изменение систем координат (для 2D и 3D).

```
> with(plots):
> p := plot([sin(x), x, x=0..2*Pi]);
> changecoords(p, polar);
```



Последний график можно получить, выполнив следующую команду:

```
> plot([sin(x)], x=0..2*Pi, coords=polar);
```

complexplot – построение комплекснозначной функции.

В качестве примера построим АФЧХ для системы управления, которая описывается передаточной функцией $w1(s)$.

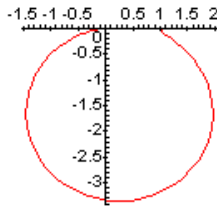
```
> with(plots) :
```

```
> w1:=1/(s^2+0.3*s+1) :
```

ФУНКЦИЯ	ОПИСАНИЕ
animate	анимация двумерных графиков
animate3d	анимация трехмерных поверхностей
changecoords	изменение систем координат (для 2D и 3D)
complexplot	построение комплекснозначной функции
complexplot3d	сложное построение графика
conformal	конформное отображение комплекснозначной функции
contourplot, contourplot3d	контурное построение поверхности
coordplot, coordplot3d	построение координатной сетки для различных систем координат
cylinderplot	построение поверхности в цилиндрических координатах
densityplot	построение плотностного графика
display, display3d	перерисовка графиков при изменении параметров построения
fieldplot, fieldplot3d	построение векторного поля на плоскости и в пространстве
gradplot, gradplot3d	построение векторного поля градиента
implicitplot, implicitplot3d	построение графика неявной функции
inequal	построение областей плоскости, отвечающих заданным неравенствам
listcontplot, listcontplot3d	построение контуров по заданной сетке значений
listdensityplot	построение плотностного графика по списку значений
listplot, listplot3d	построение ломаной, заданной списком значений ординат
loglogplot	построение графика в логарифмическом масштабе для оси X и Y

ФУНКЦИЯ	ОПИСАНИЕ
logplot	построение графика в логарифмическом масштабе по оси Y
matrixplot	изображение значений матрицы в виде трехмерного графика или гистограммы
odeplot	построение графика численного решения дифференциального уравнения
pareto	построение диаграммы Парето
pointplot, pointplot3d	построение точек на координатной плоскости или в пространстве
polarplot	построение в полярных координатах
polygonplot, polygonplot3d	построение многоугольников на координатной плоскости или в пространстве
polyhedraplotreplot	построение точек в пространстве в форме многогранников
rootlocus	строит комплексные корни уравнения
semilogplot	построение в логарифмическом масштабе (логарифмический масштаб откладывается только по оси X)
setoptions, setoptions3d	задание параметров "по умолчанию" для построения двумерной и трехмерной графики
spacecurve	параметрическое построение кривой в пространстве
sparsematrixplot	построение точек на координатной плоскости при ненулевых соответствующих элементах в матрице
sphereplot	построение поверхности в сферических координатах
surfdata	построение поверхности по данным
textplot, textplot3d	изображение на чертеже текстового объекта
tubeplot	параметрическое построение поверхности, получаемой из параметрической кривой путем придания ей объема через задание радиуса

```
> complexplot(subs(s=I*w,w1), w=0..infinity);
```



Идентичный результат можно получить с помощью функции plot:

```
> plot([Re(subs(s=I*w,w1)), Im(subs(s=I*w,w1)), w=0..infinity]);
```

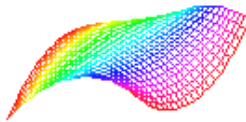
complexplot3d – сложное построение графика.

Аргументом этой функции может являться либо список из двух выражений, либо комплекснозначная функция. В зависимости от того, какой вариант задания аргумента используется выполняется различный алгоритм построения.

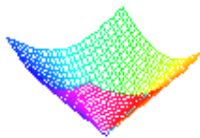
1. Список из двух выражений или отображение $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ – строится первая компонента списка, поверхность раскрашивается в зависимости от второго параметра.

2. Комплекснозначная функция или отображение $\mathbb{C} \rightarrow \mathbb{C}$ – строится абсолютная величина комплекснозначной функции и раскрашивается в зависимости от аргумента.

```
> complexplot3d([x^2*y - y^2, 2*y], x = -2..2, y = -2..2);
```

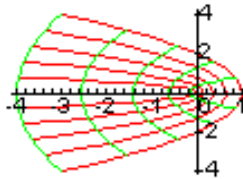


```
> complexplot3d(z), z = -2 - 2*I .. 2 + 2*I);
```



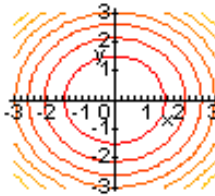
conformal – конформное отображение комплекснозначной функции.

> **conformal**($z^2, z=-1-2*I..1+2*I$) ;



contourplot, *contourplot3d* – контурное построение поверхности (подобно уровням высот на географических картах).

> **contourplot**($(x^2+y^2), x=-3..3, y=-3..3$) ;

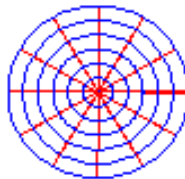


> **contourplot3d**($(x^2+y^2), x=-3..3, y=-3..3$) ;

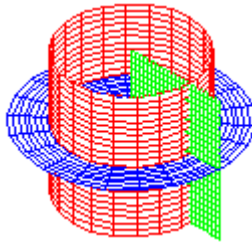


coordplot, *coordplot3d* – построение координатной сетки для различных систем координат (подобно миллиметровой бумаге).

> **coordplot**(*polar*) ;



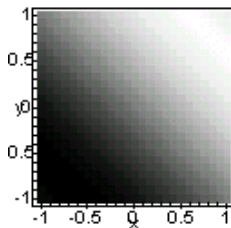
```
> coordplot3d(cylindrical);
```



cylinderplot – построение поверхности в цилиндрических координатах.

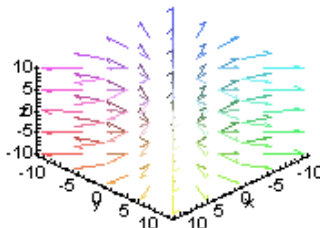
densityplot – построение плотностного графика.

```
> densityplot(sin(x+y), x=-1..1, y=-1..1,
axes=boxed, style=patchnogrid);
```



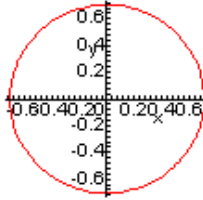
fieldplot, *fieldplot3d* – построение векторного поля на плоскости и в пространстве.

```
> fieldplot3d([2*x, 2*y, 1], x=-10..10, y=-10..10, z=-10..10,
axes=frame, grid=[5, 5, 5]);
```



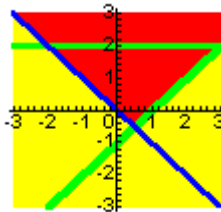
implicitplot, *implicitplot3d* – построение графика неявной функции вида $f(x,y)=c$ или $f(x,y,z)=c$.

```
> implicitplot(x^2 + y^2 = 0.5, x=-1..1, y=-1..1);
```



inequal – построение областей плоскости, отвечающих заданным неравенствам.

```
> inequal( { x+y>0, x-y<=1, y=2 }, x=-3..3,
>          y=-3..3,
>          optionsfeasible=(color=red),
>          optionsopen=(color=blue, thickness=2),
>          optionsclosed=(color=green,
>          thickness=3),
>          optionsexcluded=(color=yellow) );
```

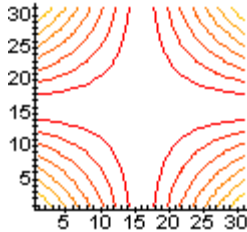


Чтобы раскрасить области разными цветами применяются следующие опции:

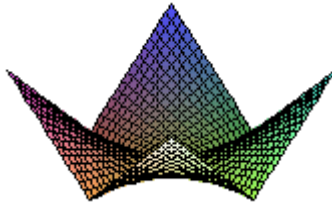
- `optionsfeasible` – для областей, которые удовлетворяют всем неравенствам;
- `optionsexcluded` – для областей, которые нарушают хотя бы одно неравенство;
- `optionsopen` – для линий, которые являются границами областей, описываемых строгими неравенствами;
- `optionsclosed` – для линий, которые являются границами областей, описываемых нестрогими неравенствами или для линий, которые описываются уравнениями.

listcontplot, *listcontplot3d* – построение контуров по заданной сетке значений (сетка – список списков).

```
> listcontplot([seq([seq( abs(x*y), x=-15..15)],
y=-15..15)]);
```

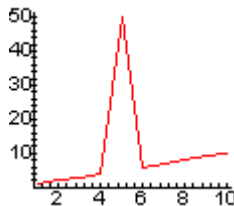


```
> listcontplot3d([seq([seq( abs(x*y), x=-
15..15)], y=-15..15)], style=patch);
```



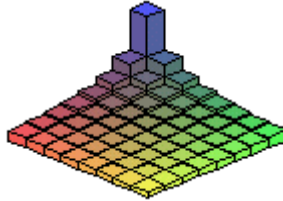
listplot, *listplot3d* – построение ломаной, заданной списком значений ординат (значение абсциссы – порядковый номер значения ординаты в списке).

```
> listplot([1,2,3,4,50,6,7,8,9,10], color=red);
```



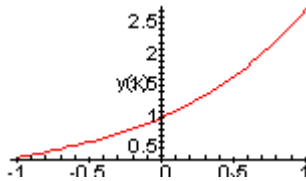
matrixplot – изображение значений матрицы в виде трехмерного графика или гистограммы.

```
> with(linalg):A:= hilbert(8):
Warning, new definition for norm
Warning, new definition for trace
> matrixplot(A,style=patch, heights=histogram);
```



odeplot – построение графика численного решения дифференциального уравнения.

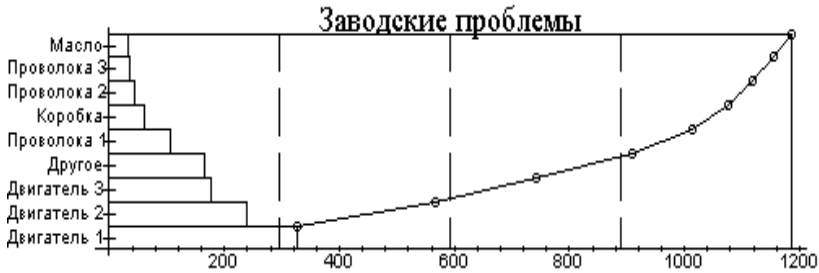
```
> p:= dsolve({D(y)(x) = y(x), y(0)=1},
y(x),type=numeric):
> odeplot(p,[x,y(x)],-1..1, labels=[`x`,
`y(x)`]);
```



pareto – построение диаграммы Парето.

```
> with(plots):
> Pdata:=[ `Двигатель 1`=327,
> `Двигатель 2`= 240, `Двигатель 3`=176,
> `Проволока 1`=105, `Проволока 2`=43,
> `Проволока 3`=36, `Масло`=33,
> `Коробка`=61, `Другое`=166]:
> Fdata:=map(rhs,Pdata):
> Lab:=map(lhs,Pdata):
> pareto(Fdata, tags=Lab, title=`Заводские
```

проблемы`);

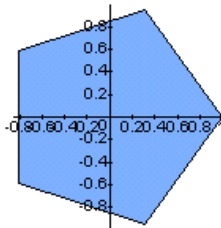


polygonplot, *polygonplot3d* – построение многоугольников на координатной плоскости или в пространстве.

```
> ngon := n -> [seq([ cos(2*Pi*k/n), sin(2*Pi*k/n) ], k = 1..n)]:
```

```
> polygonplot(ngon(5), color=COLOR(RGB, 0.5, 0.7, 1), xtickmarks=5, ytickmarks=5);
```

Warning, `k` in call to `seq` is not local



polyhedraplot – построение точек в пространстве в форме многогранников.

```
> with(plots):
```

```
> polyhedraplot([0,0,0], polytype=dodecahedron, style=PATCH, scaling=CONSTRAINED,
```

```
orientation=[71,66];
```



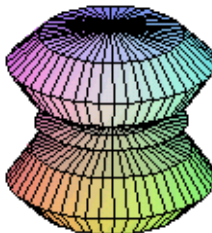
```
> points:=map(evalf,[seq([14*sin(Pi*T/7),
14*cos(Pi*T/7), 25*T/50], T=10..50)]):
> polyhedraplot(points, style=patch,
polytype=icosahedron, polyscale=3);
```



rootlocus – строит комплексные корни уравнения вида $1 + k f(s) = 0$ при вызове *rootlocus*(*f*(*s*),*s*,*k*=*kmin*..*kmax*) или корни уравнения $d(s) + k n(s)$ в случае $f(s)=n(s)/d(s)$.

sphereplot – построение поверхности в сферических координатах.

```
> sphereplot((5*cos(y)^2 -1)/2, x=0..Pi, y=-
Pi..Pi, style=PATCH);
```

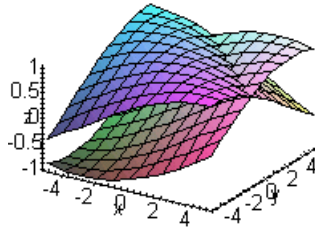


surfdata – построение поверхности по данным.

```

> data1 := [seq([ seq([i,j,evalf(cos((i+j)/5))],
i=-5..5)], j=-5..5)];
> data2 := [seq([ seq([i,j,evalf(sin((i+j)/5))],
i=-5..5)], j=-5..5)];
> surfdata( {data1,data2}, axes=frame,
labels=[x,y,z], style=patch, orientation=[-
58,54] );

```



textplot, *textplot3d* – изображение на чертеже текстового объекта.

```

> p := plot(sin(x),x=-Pi..Pi): delta := 0.05:
> t1 := textplot([Pi/2,1+delta,`Локальный
максимум (Pi/2, 1)`], align=ABOVE):
> t2 := textplot([-Pi/2,-1,`Локальный минимум (-
Pi/2, -1)`], align=BELOW):
> display({p,t1,t2});

```



tubeplot – параметрическое построение поверхности, получаемой из параметрической кривой путем придания ей объема через задание радиуса.

```
> tubeplot([3*sin(t), t, 3*cos(t)], t=-3*Pi..4*Pi,  
radius=1.2+cos(t), numpoints=80, style=patch);
```

