

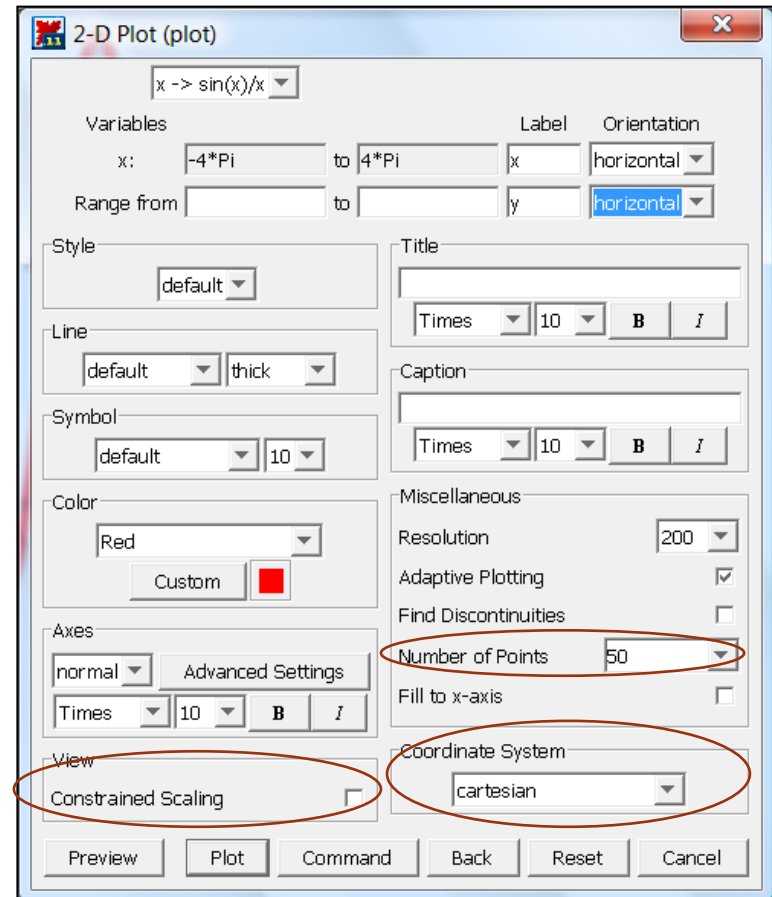
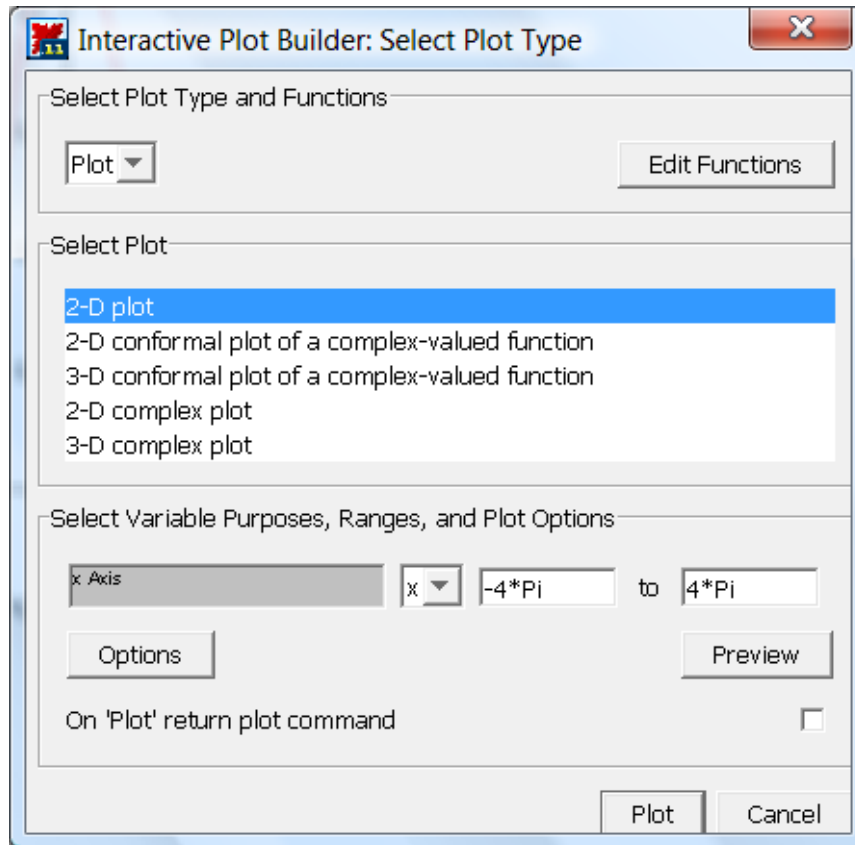
Двумерные графики

- Использование Plot Builder
- Команда `plot` и ее параметры
- График функции в полярных координатах
- График параметрической кривой
- Два графика на одном рисунке (в одних осях)
- График кривой по точкам
- Команды пакета `plots`: `implicitplot`, `textplot`, `pointplot`, `display`, `inequal`
- Двумерные графические структуры

Построение графиков с помощью Plot Builder

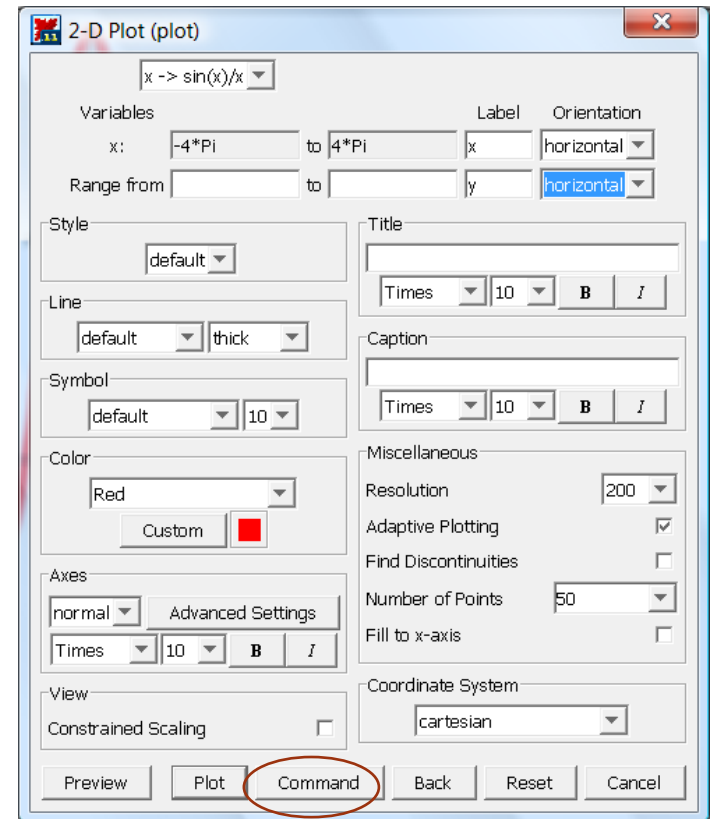
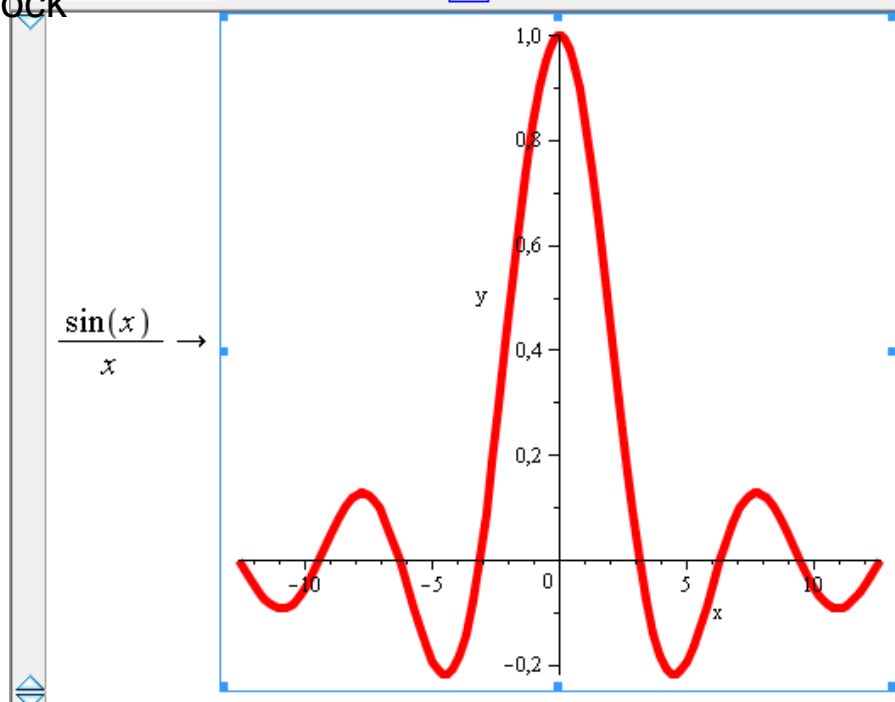
Дана функция $y = \frac{\sin x}{x}$ Введем правую часть: $\frac{\sin(x)}{x}$

Для установки интервала, ширины линии и подписи осей будем использовать Plot Builder. (Контекстное меню Plots->Plot Builder)



Использование Plot Builder в документном блоке и рабочей группе

Из Worksheet mode: Format->Create Document Block



Итоговая команда из Plot Builder

```
> sin(x)/x
> 'plot(sin(x)/x, x = -4 * pi .. 4 * pi, thickness = 5, labels = ["x", "y"])'
plot(sin(x)/x, x = -4 * pi .. 4 * pi, thickness = 5, labels = ["x", "y"])
```

(1)

Команда plot и ее параметры

plot(f(x), x=a..b, y=c..d, parameters)

Команда для построения двумерного графика явной функции.

Обязателен только первый аргумент. Если не указать интервалы по осям, то по умолчанию интервал по оси абсцисс $x=-10..10$, интервал по оси ординат подбирается автоматически.

В качестве **parameters** можно указать следующие (полный список см. в Help: Plot Options):

1) **title="text"** или **caption="text"**, где **text** – заголовок (сверху) или подпись (снизу) рисунка. Текст можно оставлять без кавычек, если он содержит только латинские буквы без пробелов. Для ввода математических выражений или изменяющихся значений можно использовать функцию typeset: `title = typeset("A plot of ", $x^2/(x+5)$, ".")`

2) **coords** – установка системы координат: **coords=polar** – график в полярных координат (по умолчанию установлены декартовы).

3) **axes** – установка типа координатных осей: **axes=NORMAL** – обычные оси; **axes=BOXED** – график в рамке со шкалой; **axes=FRAME** – оси с центром в левом нижнем углу рисунка; **axes=NONE** – без осей.

Команда plot и ее параметры

4) **scaling** – установка масштаба рисунка:

scaling=CONSTRAINED – одинаковый масштаб по осям;

scaling=UNCONSTRAINED – график масштабируется по размерам окна (установлен по умолчанию).

5) **style=LINE(POINT)** – вывод линиями (или точками).

6) **numpoints=n** – число вычисляемых точек графика (значение по умолчанию равно 50), можно использовать для повышения гладкости изображения, либо: **grid=[m,n]**, где m и n – размеры прямоугольной сетки (число узлов) по оси абсцисс и ординат, соответственно

7) **color** – установка цвета линии: английское название цвета

- с маленькой буквы: **black, blue, red, green, yellow, magenta** и др. (стандартные цвета)

- с большой буквы в кавычках: "Azure", "BlanchedAlmond", "DarkRed " и др. (новая расширенная палитра цветов)

- в виде RGB: [159, 159, 95] – цвет хаки

Команда plot и ее параметры

8) **xtickmarks=nx** и **ytickmarks=ny** – число меток по оси Ox и оси Oy , соответственно.

9) **thickness=n**, где **n=0,1,2,3...** – толщина линии (по умолчанию **n=0**)

10) **linestyle=t** – тип линии: непрерывная, пунктирная и т. д.

t может быть одним из зарезервированных имен (**solid, dot, dash, dashdot, longdash, spacedash, spacedot**), либо может быть числом от 1 до 7 (**n=1** – непрерывная, установлено по умолчанию).

11) **symbol=s** – тип символа, которым помечают точки: **BOX, CROSS, CIRCLE, POINT, DIAMOND**.

12) **font=[f,style,size]** – установка типа шрифта для вывода текста: **f** задает название шрифтов: **TIMES, COURIER, HELVETICA, SYMBOL**; **style** задает стиль шрифта: **BOLD, ITALIC, UNDERLINE**; **size** – размер шрифта в pt.

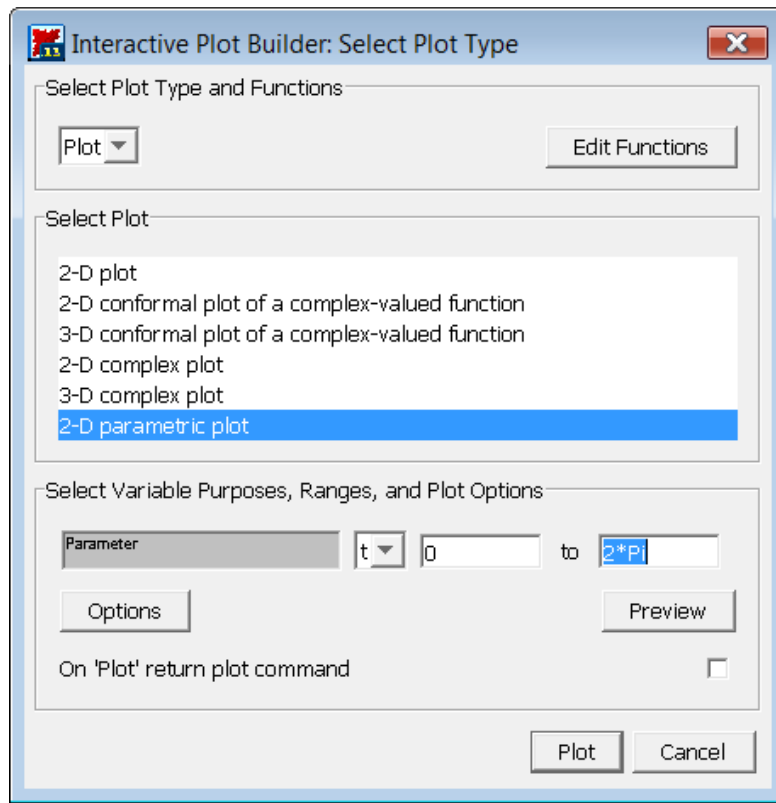
13) **labels=[tx,ty]** – подписи осей координат: **tx** – подпись оси Ox , **ty** – подпись оси Oy . Подписи появятся автоматически, если указать интервалы изменения осей.

14) **discont=true** – указание для построения бесконечных разрывов.

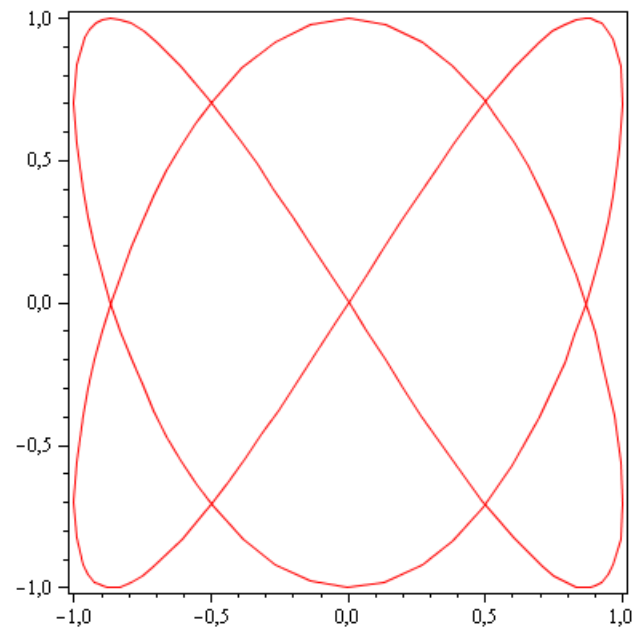
График параметрической кривой: PlotBuilder

Построить график параметрической кривой $x(t) = \sin 2t$, $y(t) = \cos 3t$ с параметром t $0 \leq t \leq 2\pi$ в рамке.

$\sin(2 \cdot t)$, $\cos(3 \cdot t)$



$\sin(2 \cdot t)$, $\cos(3 \cdot t) \rightarrow$



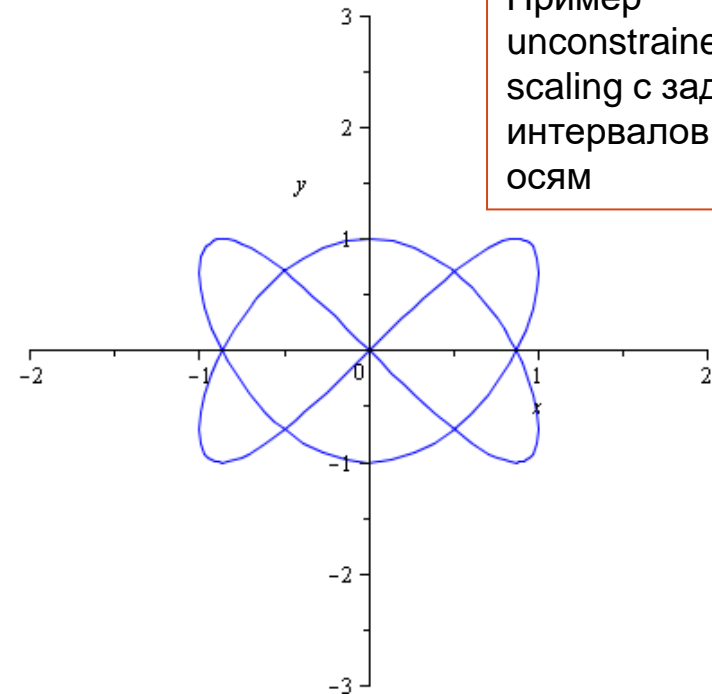
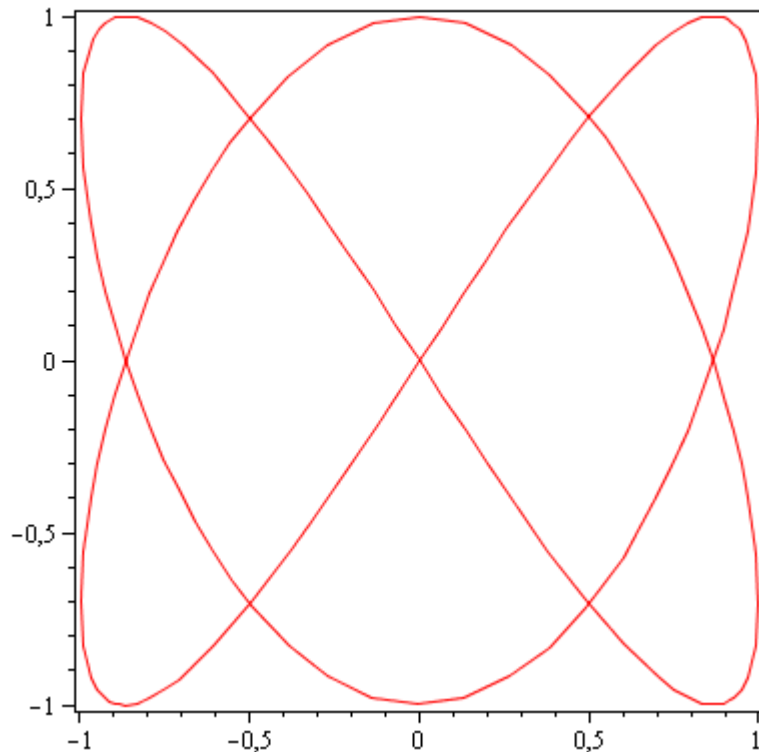
В Options задайте тип осей: Axes->boxed

График параметрической кривой: команда plot с интервалом изменения параметра

➤ Для построения графика параметрической функции описание интервала изменения параметра должно находиться внутри списка функций правых частей (как первый аргумент команды plot)

plot([x(t), y(t), t=t1..t2], x=x1..x2, y=y1..y2, options)

> `plot([sin(2·t), cos(3·t), t=0..2·π], axes=boxed)` > `plot([sin(2·t), cos(3·t), t=0..2·π], x=-2..2, y=-3..3, color=blue)`



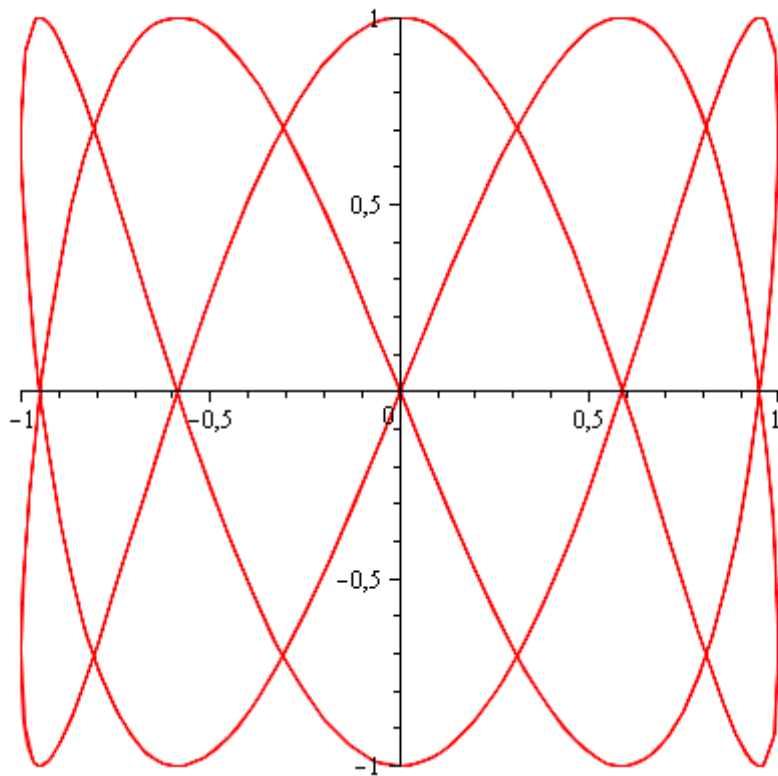
Пример
unconstrained
scaling с заданием
интервалов по
осям

График параметрической кривой: порядок аргументов в команде plot

Для графика параметрической кривой важен порядок аргументов:

$$x=\sin(2t), y=\cos(5t)$$

```
> plot([sin(2·t), cos(5·t), t=-2·π..2·π])
```



$$x=\cos(5t), y=\sin(2t)$$

```
> plot([cos(5·t), sin(2·t), t=-2·π..2·π])
```

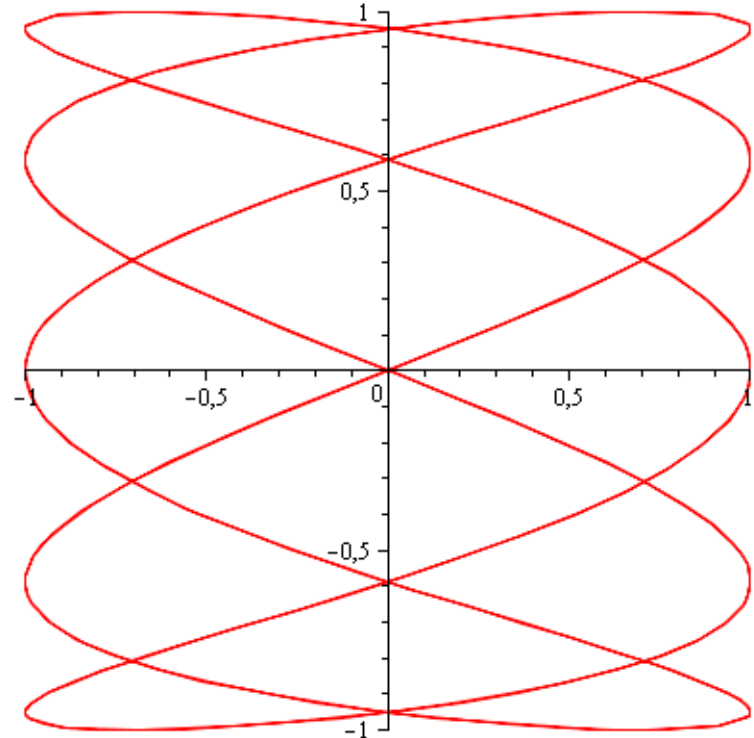


График функции в полярных координатах: PlotBuilder

Построить в полярных координатах график кардиоиды $\rho = 1 + \cos \varphi$, $\varphi \in [0, 2\pi]$, с подписью «Кардиоида». Цвет линии – coral. Шрифт подписи – Helvetica, жирный, 14 пт.

$$\left[\begin{array}{l} > 1 + \cos(\text{phi}) \\ & 1 + \cos(\phi) \end{array} \right]$$

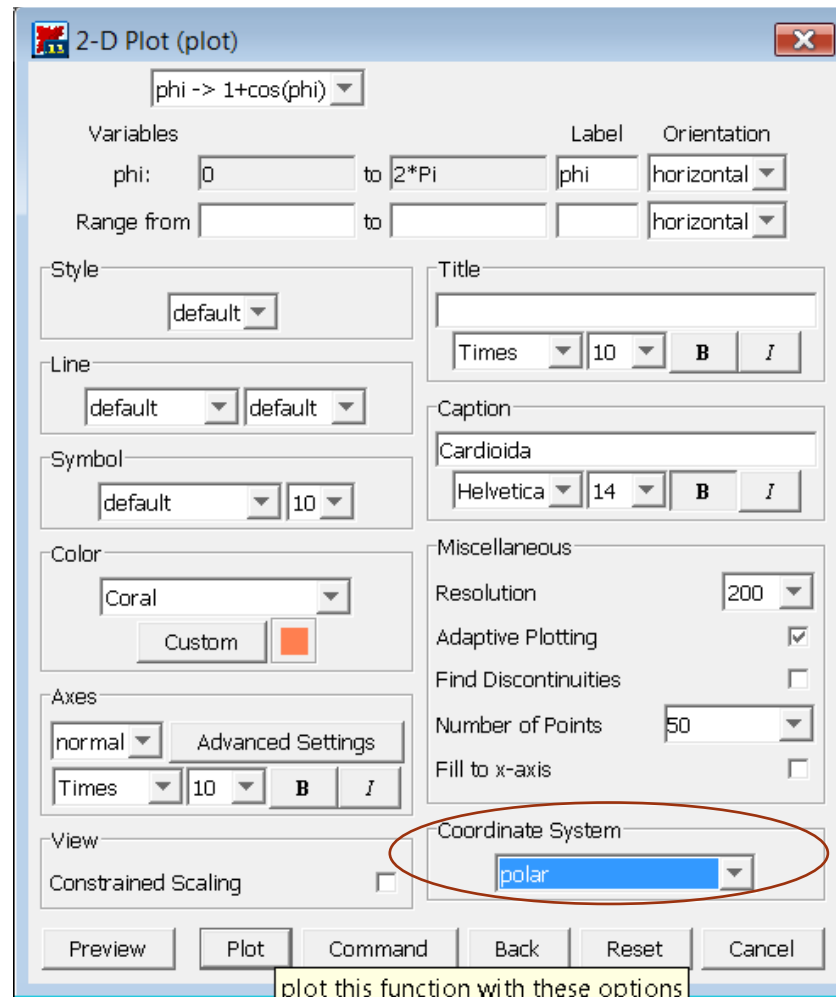
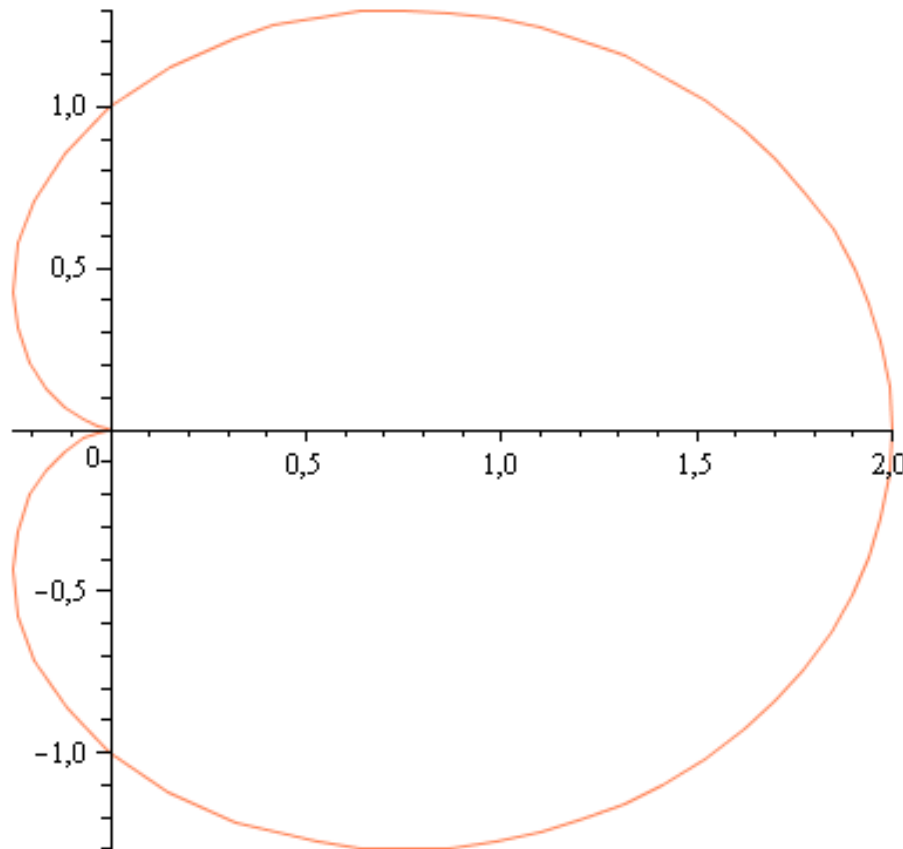



График функции в полярных координатах: команда `plot` с опцией `coords=polar`

```
> plot(1 + cos( $\phi$ ),  $\phi = 0 .. 2 * \pi$ , color = "Coral", caption = "Cardioida",  
captionfont = [HELVETICA, BOLD, 14], coords = polar)
```

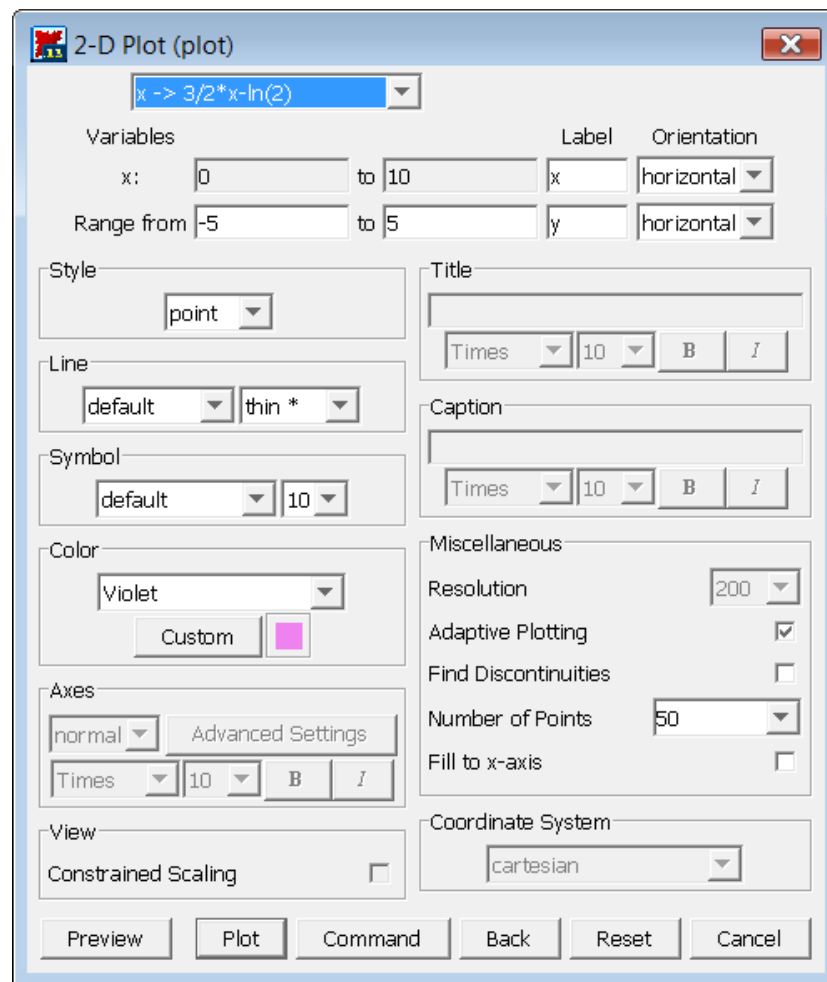
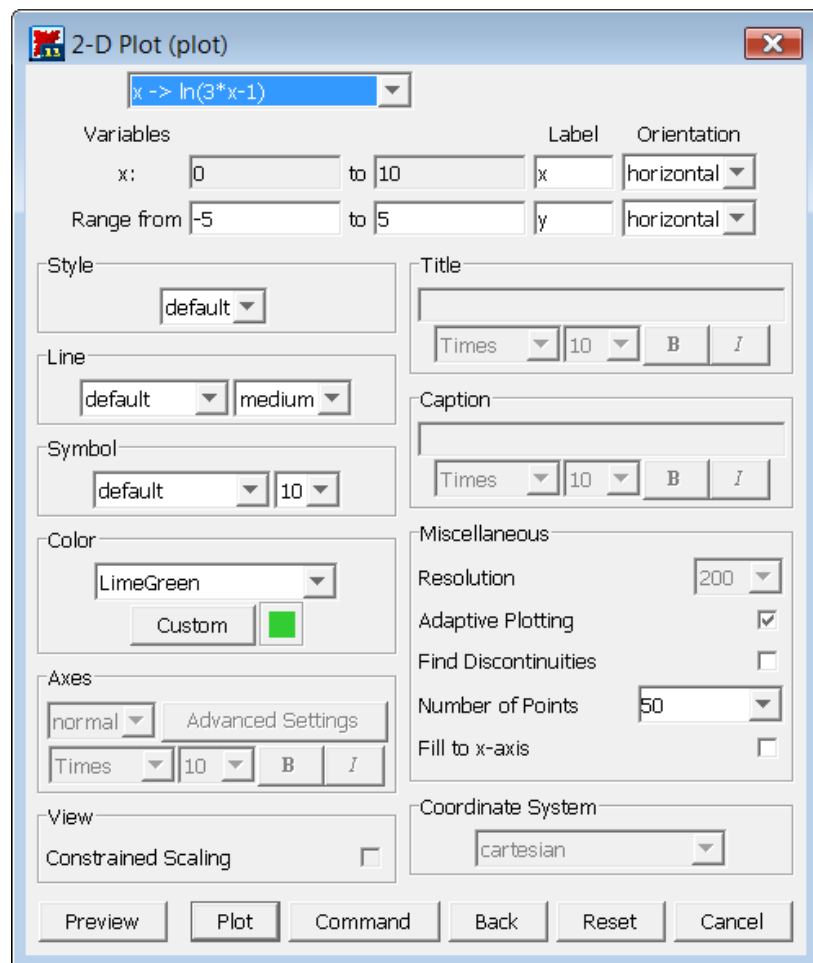


Cardioida

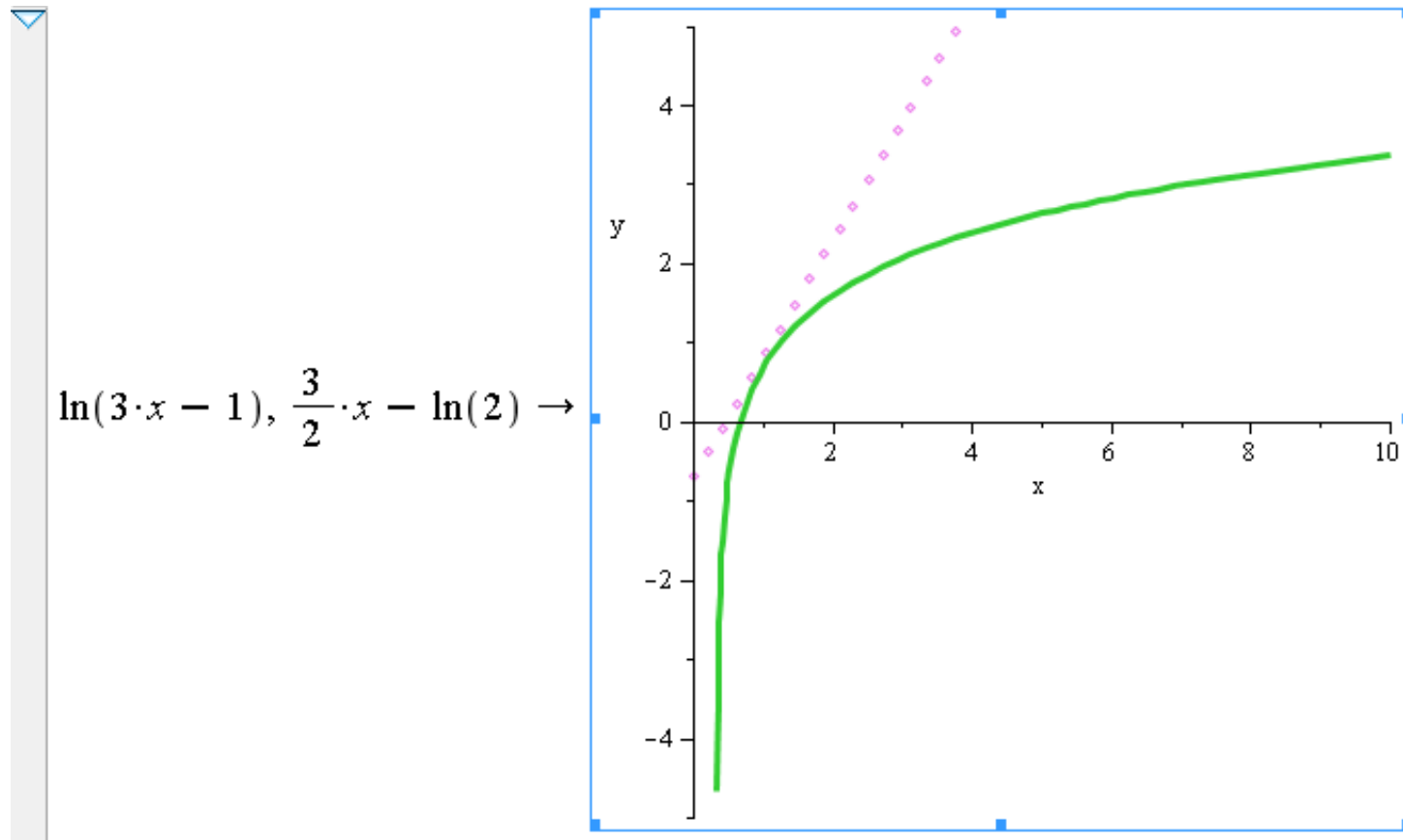
Два графика на одном рисунке (в одних осях): PlotBuilder


$$\ln(3 \cdot x - 1), \frac{3}{2} \cdot x - \ln(2)$$

Построить два графика на одном рисунке: график функции $y = \ln(3x - 1)$ и касательную к нему $y = \frac{3}{2}x - \ln 2$. Выбрать разные цвета и толщины для линий.



Два графика на одном рисунке (в одних осях)



Итоговая команда из Plot Builder: две команды plot!

```
> plots[display](plot(3/2*x - ln(2), x = 0 .. 10, style = point, thickness = 1, color = "Violet"),  
plot(ln(3*x - 1), x = 0 .. 10, thickness = 3, color = "LimeGreen"),  
view = [DEFAULT, -5 .. 5], labels = ["x", "y"])
```

Два графика на одном рисунке (в одних осях): одна команда plot

➤ Для построения графиков двух или более функций в одних осях функции надо поместить в список как первый аргумент команды plot, а интервал изменения общей переменной должен быть снаружи списка (второй аргумент команды plot).

Для предыдущего графика можно записать:

```
> plot([ln(3·x - 1),  $\frac{3}{2} \cdot x - \ln(2)$ ],  $x = 0 \dots 10$ ,  $y = -5 \dots 5$ ,  $color = ["LimeGreen", "Violet"]$ ,  
style = [line, point],  $thickness = [3, 1]$ ,  $labels = [x, y]$ )
```

Другой пример:

```
> plot([x, x2, x3],  $x = -5 \dots 5$ ,  $y = -5 \dots 5$ ,  $thickness = 3$ ,  $color = [red, blue, green]$ )
```

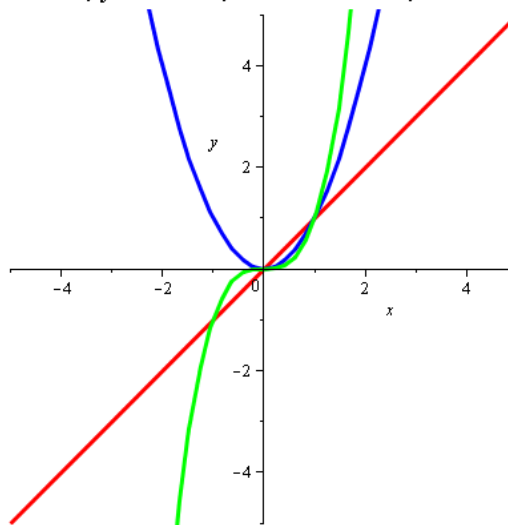
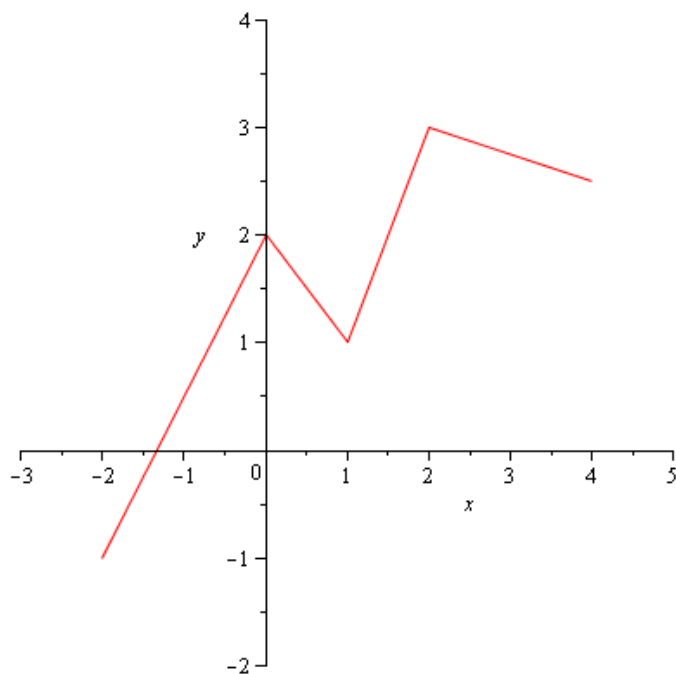


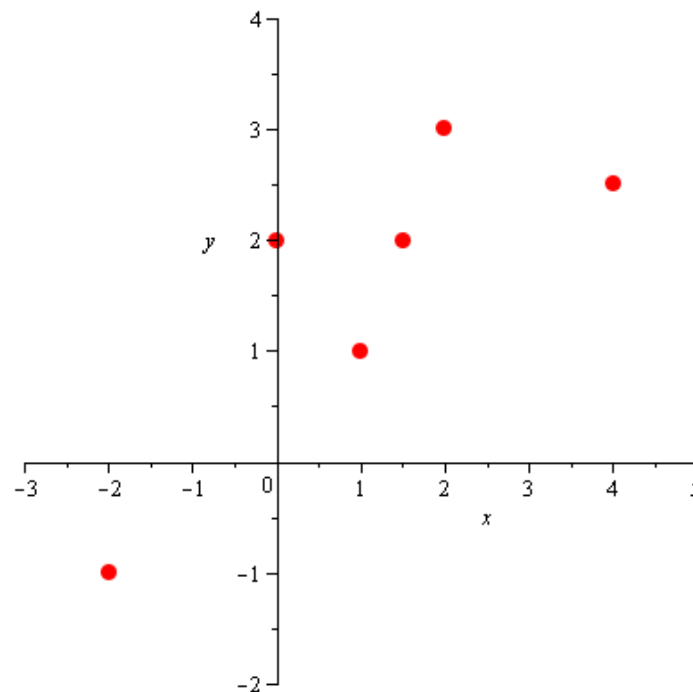
График кривой по точкам

Для построения графика функции, заданной таблично (по точкам), можно использовать команду **plot**, задав набор точек в виде списка: **plot([[x1,y1],[x2,y2],...,[xn,yn]],options)**. По умолчанию точки будут соединены линиями. Точки без линий: **style=point**

> `plot([[-2,-1], [0, 2], [1, 1], [1.5, 2], [2, 3], [4, 2.5]],`
`x=-3 ..5, y=-2 ..4);`



> `plot([[-2,-1], [0, 2], [1, 1], [1.5, 2], [2, 3], [4, 2.5]],`
`x=-3 ..5, y=-2 ..4, style=point, symbol`
`=solidcircle, symbolsize=20);`



Команды пакета **plots**:

implicitplot, textplot, pointplot

Построение графика функции, заданной неявно (команда **implicitplot**)

Функция задана неявно, если она задана уравнением $F(x,y)=0$.

Для построения двумерного графика неявной функции используется команда **implicitplot** из графического пакета **plots** в виде

implicitplot(F(x,y)=0, x=x1..x2, y=y1..y2,options)

Вывод текстовых комментариев на рисунок (команда **textplot**)

В пакете **plots** имеется команда **textplot** для вывода текстовых комментариев на рисунок: **textplot([x0,y0,"text"], options)**, где x_0, y_0 – координаты точки, где начинается вывод текста **text**.

Построение двумерных точек (команда **pointplot**)

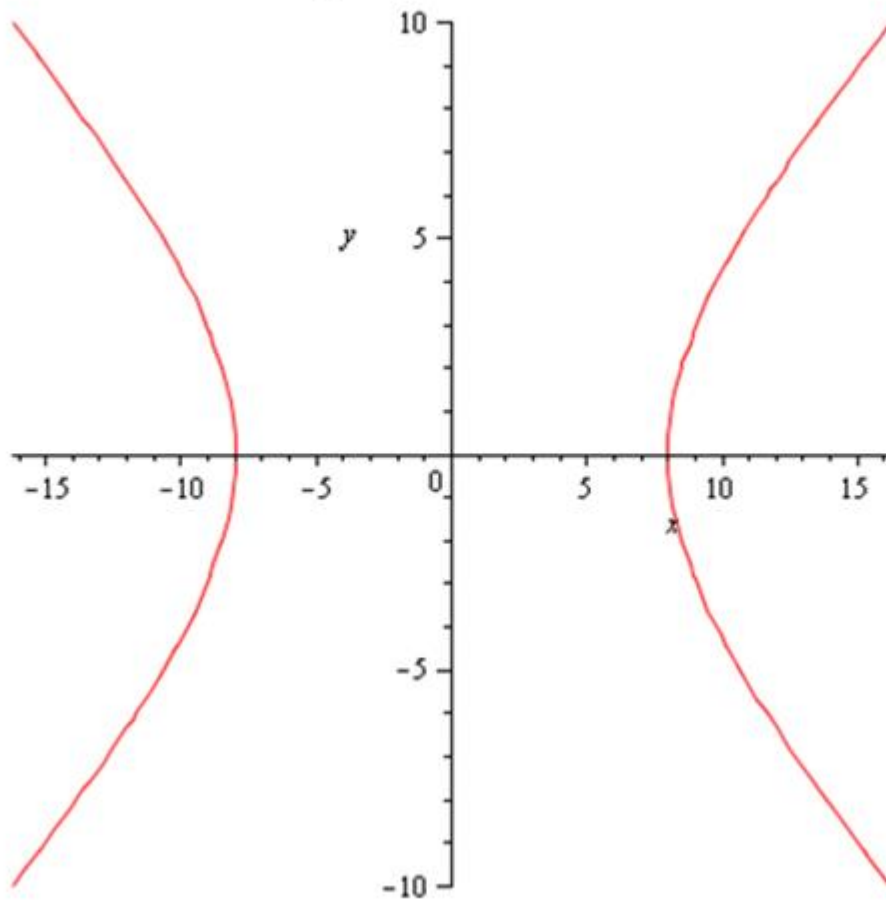
Используется для построения набора точек на плоскости. Набор точек может быть задан в виде списка или множества

pointplot([[x1,y1],[x2,y2],...,[xn,yn]],options)

Точки помощью команды **plot**: **plot([[x1,y1],[x2,y2],...,[xn,yn]],style=point)**

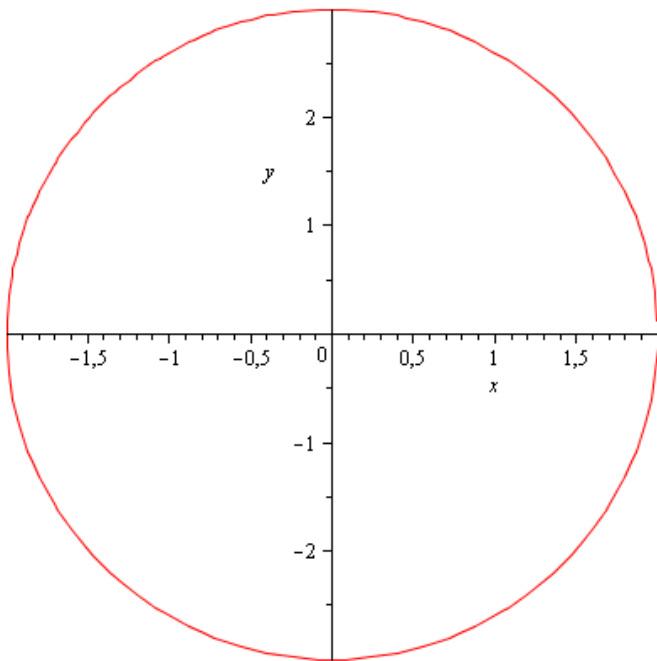
Команды пакета plots: пример implicitplot (график неявной функции)

> with(plots) : implicitplot $\left(\frac{x^2}{4} - \frac{y^2}{2} = 16, x = -20 \dots 20, y = -10 \dots 10 \right)$

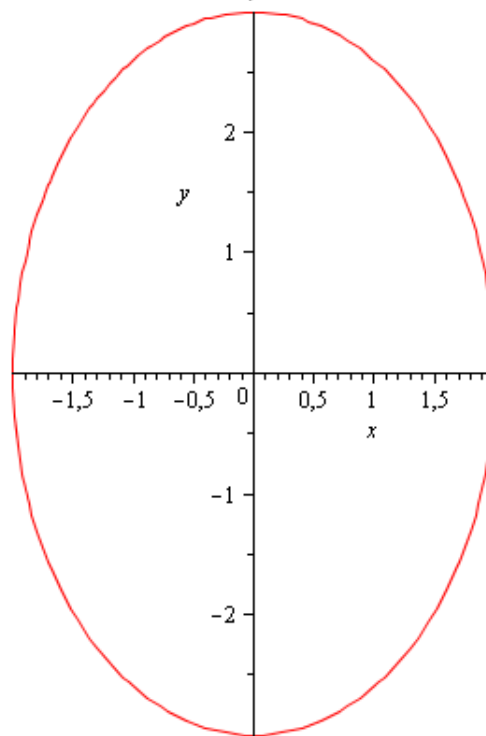


Команды пакета plots: пример `implicitplot` без и с заданием масштаба по осям (график эллипса)

> `plots[implicitplot]` $\left(\frac{x^2}{4} + \frac{y^2}{9} = 1, x = -2..2, y = -3..3\right)$

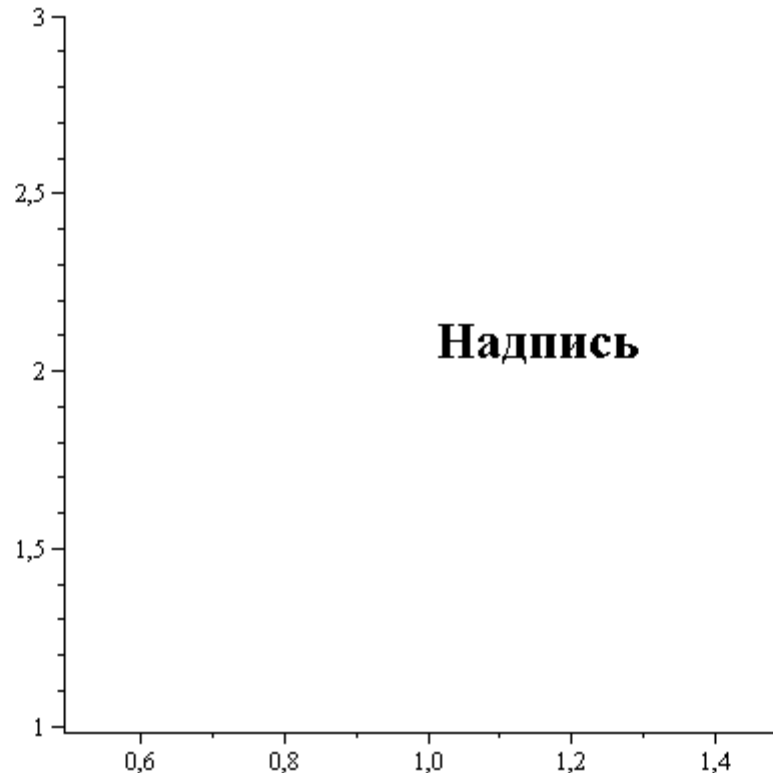


> `plots[implicitplot]` $\left(\frac{x^2}{4} + \frac{y^2}{9} = 1, x = -2..2, y = -3..3, \right.$
 $\left. \text{scaling} = \text{constrained} \right)$



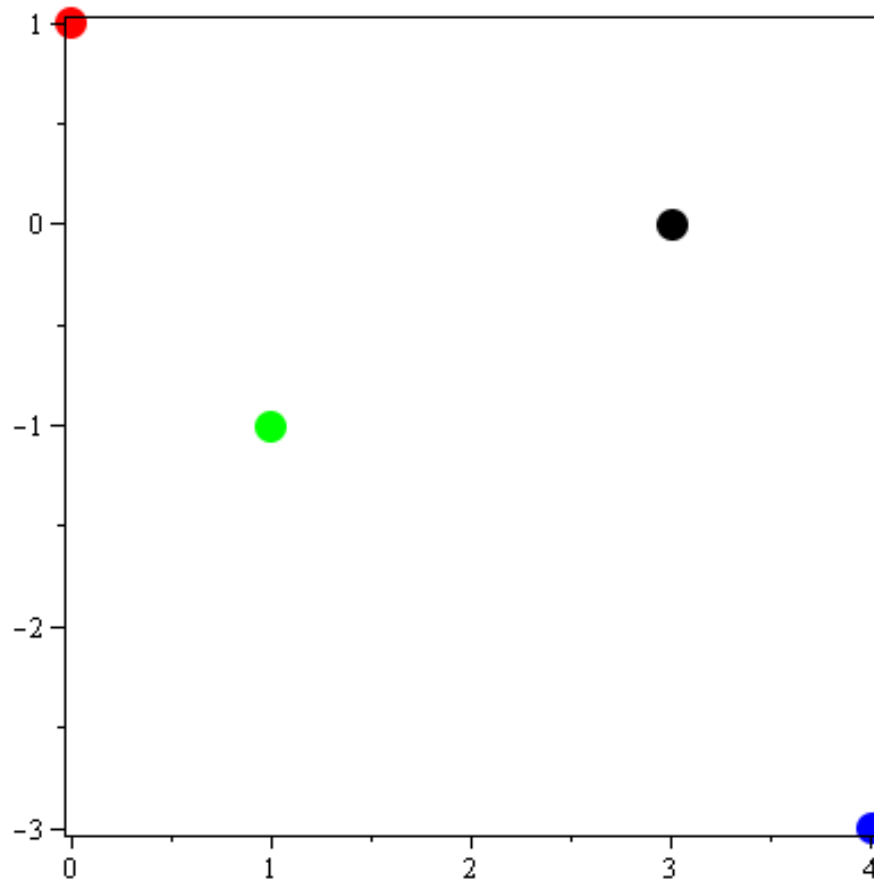
Команды пакета plots: пример textplot (вывод текстовой надписи)

```
> with(plots) :  
> textplot( [ 1, 2, "Надпись" ], align = { above, right }, font = [ times, bold, 20 ] )
```



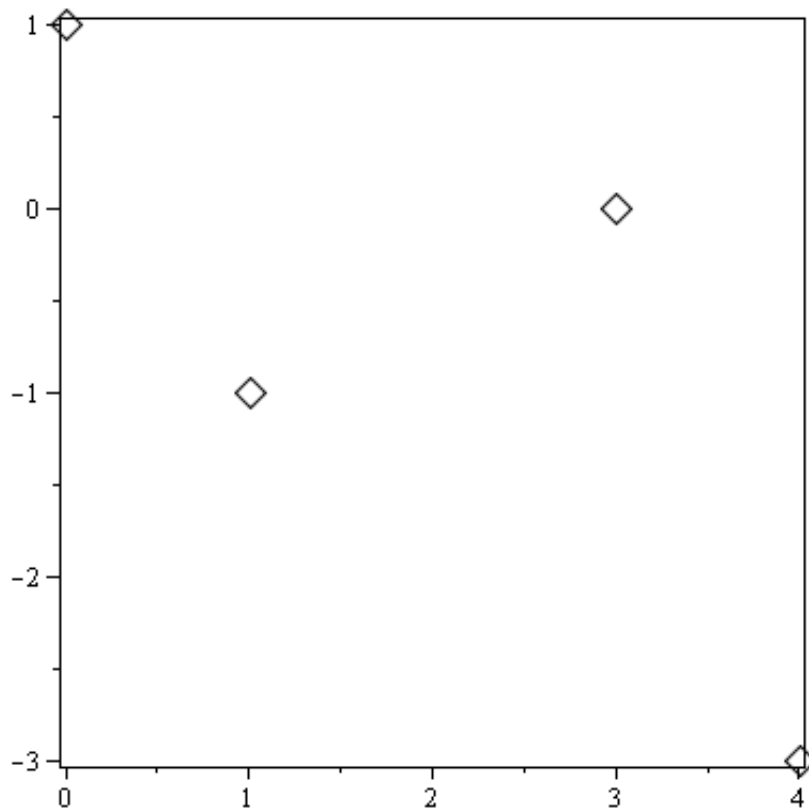
Команды пакета plots: пример pointplot (изображение точек на плоскости)

```
> with(plots) :  
> pointplot([[0, 1], [1, -1], [3, 0], [4, -3]], axes = boxed, symbol  
    = solidcircle, symbolsize = 30, color = [red, green, black, blue])
```



Точки с помощью команды plot

> `plot([[0, 1], [1, -1], [3, 0], [4, -3]], style = point, axes = boxed, symbol = diamond, symbolsize = 30, color = black);`



Команды пакета **plots**: **display**

Вывод нескольких графических объектов на один рисунок (команда **display**)

Эта команда необходима для совмещения на одном рисунке нескольких графических объектов, полученных при помощи различных графических команд, например, результата команды **plot** (построение графика) и результата команды **textplot** (текстовые подписи к графику). Для этого результат действия графической команды присваивается некоторой переменной:

```
> p:=plot(...): t:=textplot(...):
```

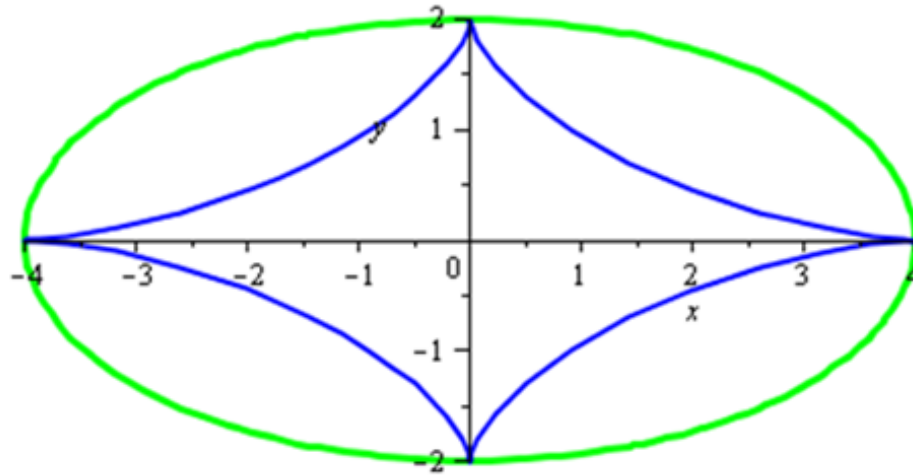
При этом на экран вывод не производится.

Для вывода графических изображений необходимо выполнить команду **display** из пакета **plots**, предварительно подключив этот пакет:

```
> with(plots): display([p,t], options)
```

Команды пакета plots: пример display (разные графические объекты на одном рисунке)

- > `with(plots) :`
- > `eq1 := $\frac{x^2}{16} + \frac{y^2}{4} = 1$:`
- > `eq2 := $4 \cdot \cos(t)^3, 2 \cdot \sin(t)^3$:`
- > `ellips := implicitplot(eq1, x = -4 .. 4, y = -2 .. 2, color = green, thickness = 3, scaling = CONSTRAINED) :`
- > `astroida := plot([eq2, t = 0 .. 2·π], color = blue, thickness = 2, scaling = CONSTRAINED) :`
- > `display([ellips, astroida])`



Команды пакета plots: `inequal`

Построение двумерной области, заданной линейными неравенствами (команда `inequal`)

Если необходимо построить двумерную область, заданную системой линейных неравенств от двух неизвестных, то для этого можно использовать команду **`inequal`** из пакета **`plots`**.

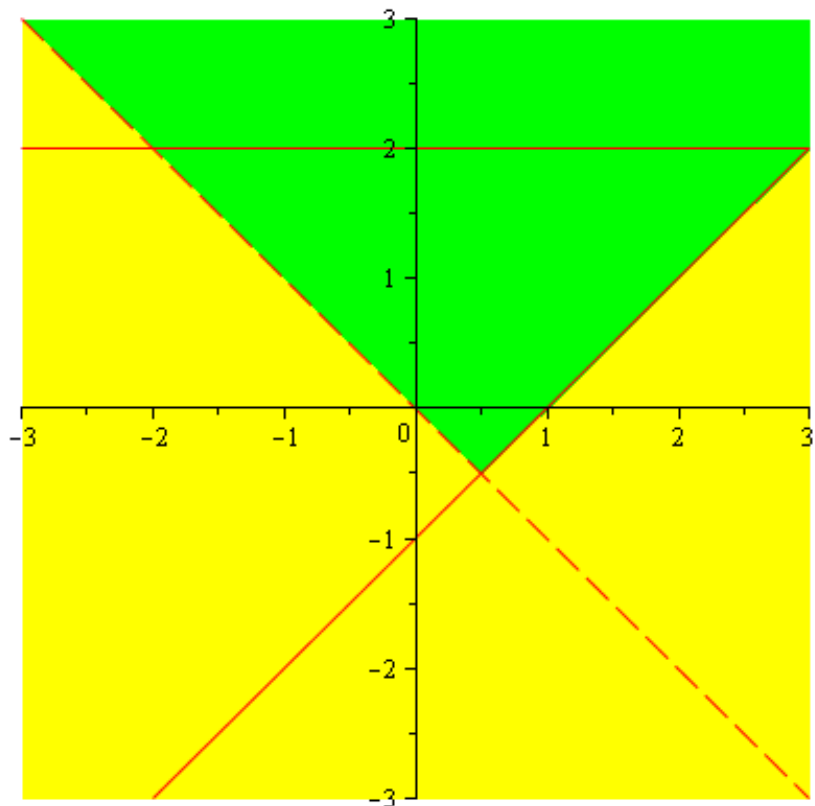
В команде **`inequal({f1(x,y)>c1,...,fn(x,y)>cn}, x=x1...x2, y=y1..y2, options)`** в фигурных скобках указывается система неравенств, определяющих область, затем размеры координатных осей и параметры.

Параметры регулируют цвета открытых и закрытых границ, цвета внешней и внутренней областей, а также толщину линий границ:

- **`optionsfeasible=(color=red)`** – установка цвета внутренней области;
- **`optionsexcluded=(color=yellow)`** – установка цвета внешней области;
- **`optionsopen=(color=blue, thickness=2)`** – установка цвета и толщины линии открытой границы;
- **`optionsclosed=(color=green,thickness=3)`** – установка цвета и толщины линии закрытой границы.

Команды пакета plots: пример `inequal`

> `inequal({ $x - y \leq 1$, $x + y > 0$, $y = 2$ }, $x = -3 \dots 3$, $y = -3 \dots 3$,
optionsopen = (color = red, linestyle = dash),
optionsclosed = (color = red), optionsfeasible = (color = green),
optionsexcluded = (color = yellow));`



Двумерные графические структуры

Незамкнутые кривые, m наборов по n точек

CURVES($[[x_{11}, y_{11}], \dots [x_{1n}, y_{1n}]], [[x_{21}, y_{21}], \dots [x_{2n}, y_{2n}]], \dots$
 $[[x_{m1}, y_{m1}], \dots [x_{mn}, y_{mn}]]$) – точки каждого набора соединяются отрезками

Замкнутые кривые, m наборов по n точек

POLYGONS($[[x_{11}, y_{11}], \dots [x_{1n}, y_{1n}]], [[x_{21}, y_{21}], \dots [x_{2n}, y_{2n}]], \dots$
 $[[x_{m1}, y_{m1}], \dots [x_{mn}, y_{mn}]]$) – точки каждого набора соединяются отрезками, первая точка соединяется с последней; в результате получается многоугольник с заливкой

Двумерные точки

POINTS($[x_1, y_1], [x_2, y_2], \dots [x_n, y_n]$)

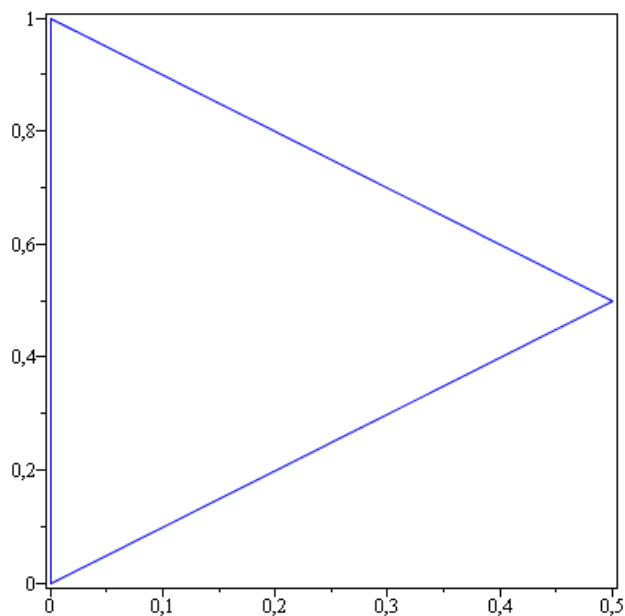
Вывод текста в заданной позиции

TEXT($[x, y], \text{string}, \text{horizontal}, \text{vertical}$)

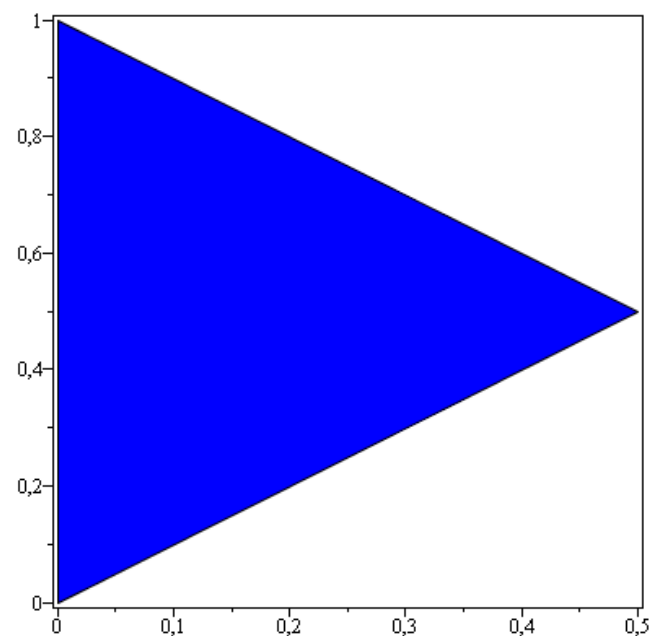
Данные структуры выводятся с помощью команды **PLOT**,
опции этой команды записываются большими буквами и имеют свой
специальный синтаксис

Двумерные графические структуры: CURVES и POLYGONS

> *PLOT(CURVES([[0, 0], [0, 1], [0.5, 0.5], [0, 0]]),
COLOR(RGB, 0, 0, 1), AXESSTYLE(BOX))*

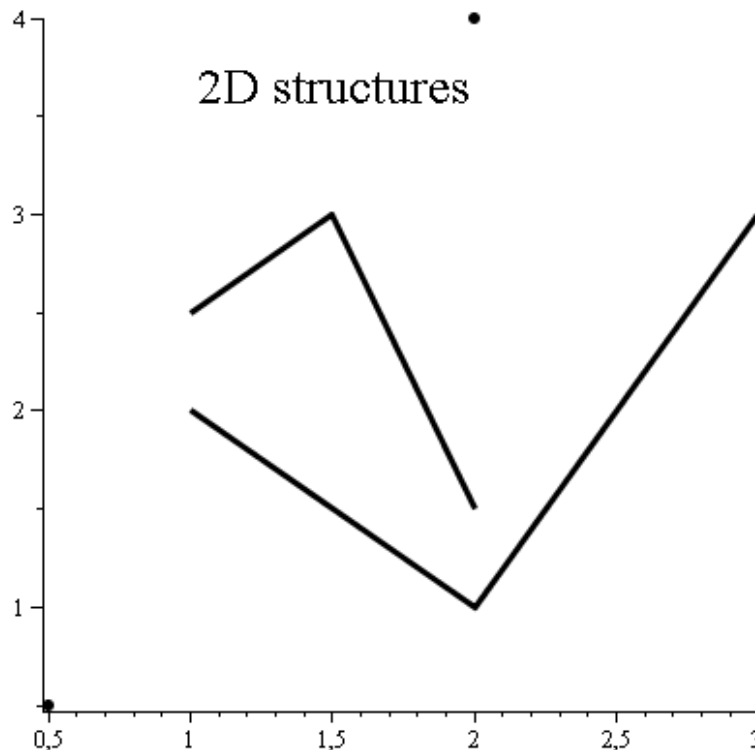


> *PLOT(POLYGONS([[0, 0], [0, 1], [0.5, 0.5]]),
COLOR(RGB, 0, 0, 1), AXESSTYLE(BOX))*



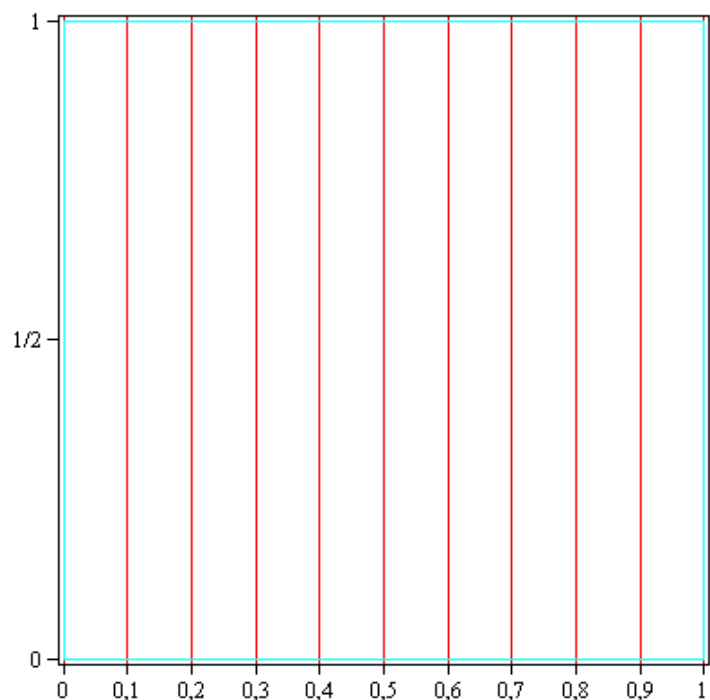
Двумерные графические структуры: CURVES, POINTS и TEXT

```
> c := CURVES([ [1, 2], [2, 1], [3, 3] ], [ [1, 2.5], [1.5, 3], [2, 1.5] ],  
    THICKNESS(3)) :  
> p := POINTS([0.5, 0.5], [2, 4], SYMBOL(_SOLIDCIRCLE, 15)) :  
> t := TEXT([1, 3.5], "2D structures", FONT(TIMES, ROMAN, 20),  
    ALIGNRIGHT, ALIGNABOVE) :  
> PLOT(c, p, t)
```



Использование опций команды PLOT

```
> PLOT(CURVES([[0, 0], [0, 1], [1, 1], [1, 0], [0, 0]]), AXESSTYLE(BOX),  
AXESTICKS(DEFAULT, [0 = '0', 0.5 = '1/2', 1 = '1']),  
_AXIS1(_GRIDLINES(10, COLOR(RGB, 1.0, 0.0, 0.0))), COLOR(HUE, 0.5))
```



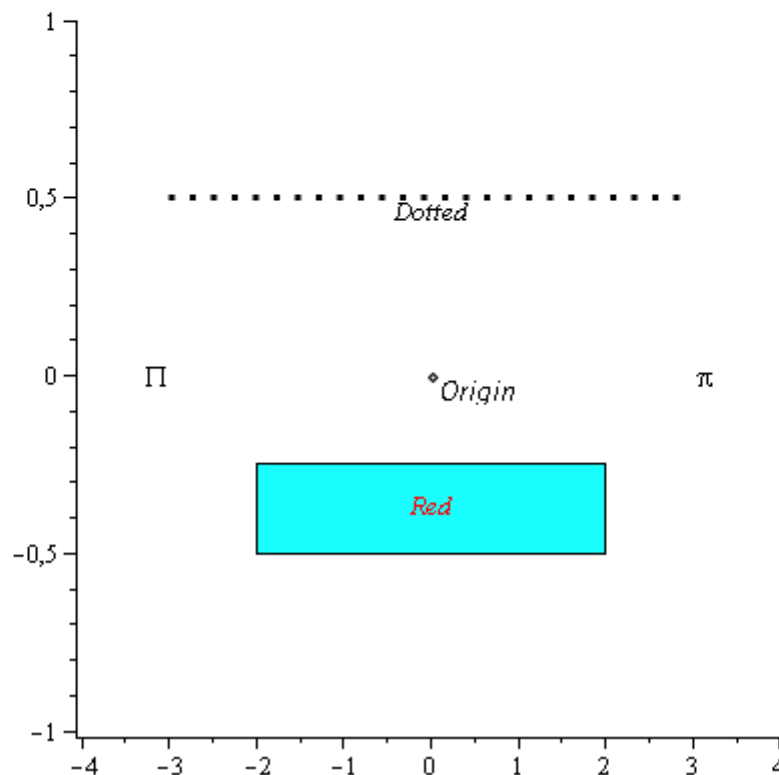
AXESTICKS - здесь заданы специальные маркеры шкалы ординат

_AXIS₁ - информация о первой оси (абсцисс) *_AXIS[1]*, здесь задана сетка красного цвета
информация о второй оси (ординат) здесь не задается

HUE - формат цвета с указанием одного значения: *COLOR(HUE, 0.0)* - красный,
COLOR(HUE, 0.5) - голубой как на рисунке (этим цветом нарисованы *CURVES*)

Пример из справочной системы Maple: ВСЕ СТРУКТУРЫ

```
> PLOT(POINTS([0, 0], SYMBOL(DIAMOND)), TEXT([0, 0], 'Origin',  
    ALIGNBELOW, ALIGNRIGHT, FONT(HELVETICA, OBLIQUE, 10)),  
    CURVES([[ -3, 0.5], [3, 0.5]], THICKNESS(3), LIFESTYLE(DOT)),  
    TEXT([0, 0.5], 'Dotted', ALIGNBELOW), TEXT([3.1415, 0], 'p',  
    FONT(SYMBOL, 12)), TEXT([-3.1415, 0], 'P', FONT(SYMBOL, 12)),  
    POLYGONS([[ -2, -0.25], [-2, -0.5], [2, -0.5], [2, -0.25]],  
    COLOR(HUE, 0.5)), TEXT([0, -0.37], 'Red', COLOR(RED, 1, 0, 0)),  
    AXESSTYLE(FRAME), VIEW(-4 .. 4, -1 .. 1))
```



Трёхмерные графики

- Команда `plot3d`
- Команды пакета `plots`: `implicitplot3d`, `pointplot3d`, `textplot3d`, `spacecurve`
- Трёхмерные графические структуры

Команда plot3d

График явной функции $z=f(x,y)$ в пространстве можно нарисовать, используя команду **plot3d(f(x,y), x=x1...x2, y=y1...y2, options)**.

Многие параметры этой команды совпадают с параметрами команды **plot**. Некоторые специальные опции команды **plot3d**:

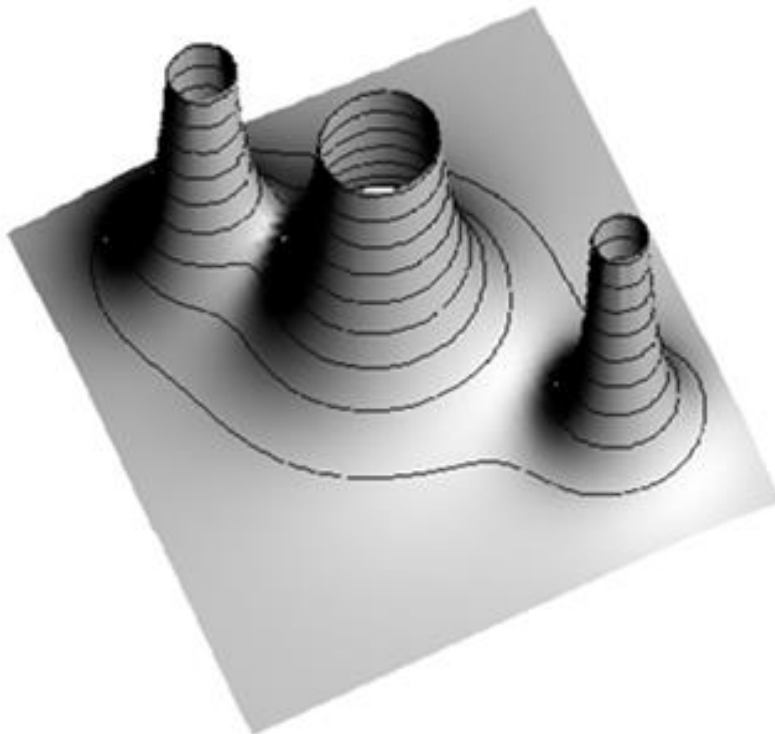
- **grid=[m,n]**, где m и n – размеры прямоугольной сетки (количество вычисляемых узлов)
- **light=[angl1, angl2, c1, c2, c3]** – задание подсветки поверхности, создаваемой источником света из точки со сферическими координатами (**angl1, angl2**). Цвет определяется долями красного (**c1**), зеленого (**c2**) и синего (**c3**) цветов, которые находятся в интервале [0,1].
- **style=opt** задает стиль рисунка: **POINT** – точки, **LINE** – линии, **HIDDEN** – сетка с удалением невидимых линий, **PATCH** – заполнитель (установлен по умолчанию), **WIREFRAME** – сетка с выводом невидимых линий, **CONTOUR** – линии уровня, **PATCHCONTOUR** – заполнитель и линии уровня.
- **shading=opt** задает функцию интенсивности заполнителя, его значение равно **xyz** – по умолчанию, **NONE** – без раскраски.

Команда plot3d: пример

```
> plot3d( $\frac{1}{x^2 + y^2} + \frac{0.2}{(x + 1.2)^2 + (y - 1.5)^2} + \frac{0.3}{(x - 0.9)^2 + (y + 1.1)^2}$ ,
```

```
x = -2 .. 2, y = -2 .. 2.5, view = 0 .. 6, grid = [60, 60],
```

```
shading = NONE, light = [100, 30, 1, 1, 1], orientation = [65, 20], style = PATCHCONTOUR)
```



Команда `plot3d`: параметрически заданная поверхность в тороидальной системе координат

```
> plot3d([1, phi, psi], phi = 0 .. 2 * pi, psi = 0 .. 2 * pi, coords = toroidal,  
scaling = constrained)
```

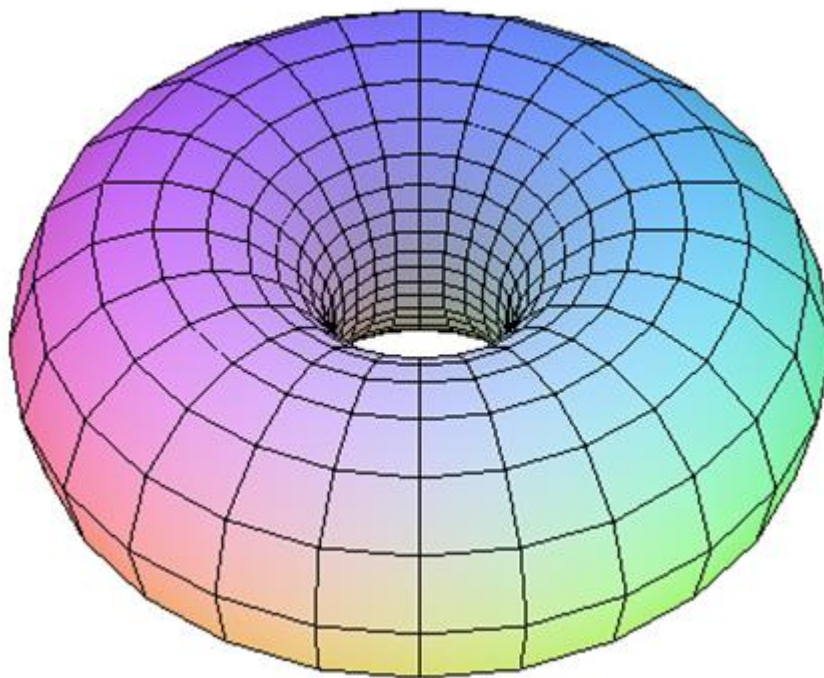
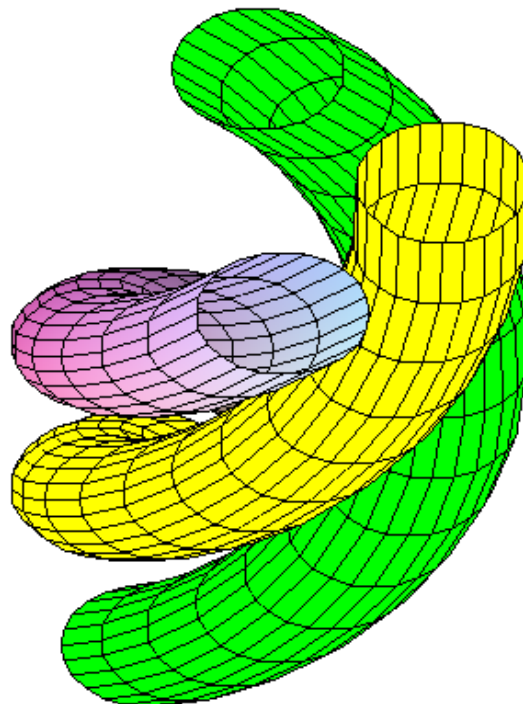


График нескольких поверхностей на одном рисунке (в одних осях)

```
> c1 := [cos(x) - 2*cos(0.4*y), sin(x) - 2*sin(0.4*y), y]:  
   c2 := [cos(x) + 2*cos(0.4*y), sin(x) + 2*sin(0.4*y), y]:  
   c3 := [cos(x) + 2*sin(0.4*y), sin(x) - 2*cos(0.4*y), y]:  
> plot3d([c1, c2, c3], x=0..2*π, y=0..10, grid=[25, 15], style=patch,  
         color=[default, green, yellow])
```



Трёхмерные команды пакета **plots**: **implicitplot3d, pointplot3d, textplot3d, spacecurve**

График поверхности, заданной неявно

Трёхмерный график поверхности, заданной неявно уравнением $F(x,y,z)=c$, строится с помощью команды пакета **plots**:
implicitplot3d(F(x,y,z)=c, x=x1..x2, y=y1..y2, z=z1..z2), где указывается уравнение поверхности $F(x,y,z)=c$ и интервалы по координатным осям.

Построение трёхмерных точек (команда **pointplot3d**)

Вывод текстовых комментариев на рисунок (команда **textplot3d**)

Данные команды аналогичны соответствующим командам для двумерных графиков

График пространственных кривых

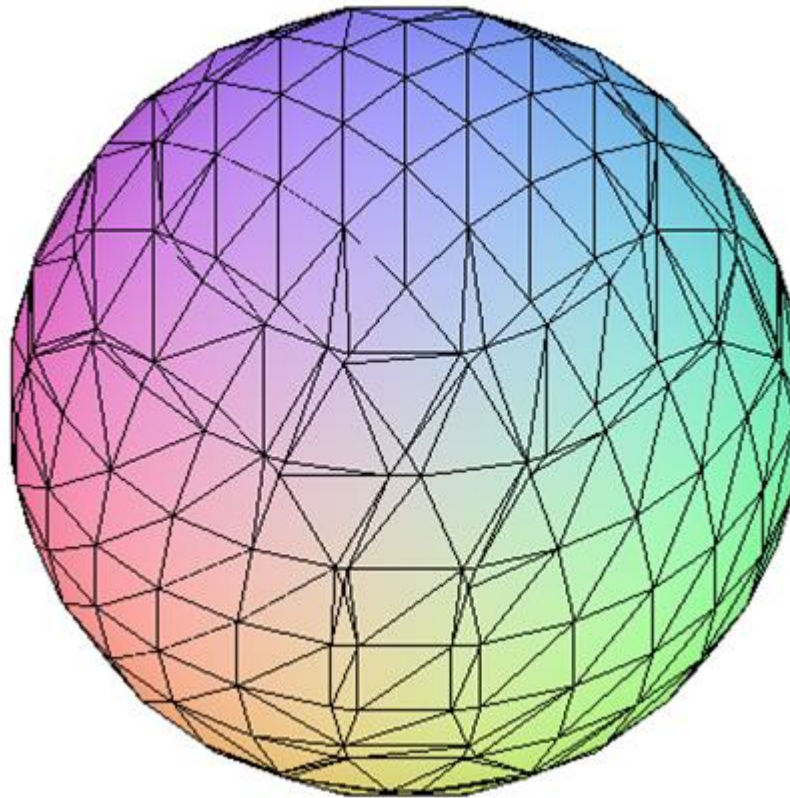
В пакете **plots** имеется команда **spacecurve** для построения пространственной кривой, заданной параметрически:

$x=x(t)$, $y=y(t)$, $z=z(t)$.

spacecurve([x(t),y(t),z(t)],t=t1..t2), где параметр **t** изменяется от **t1** до **t2**.

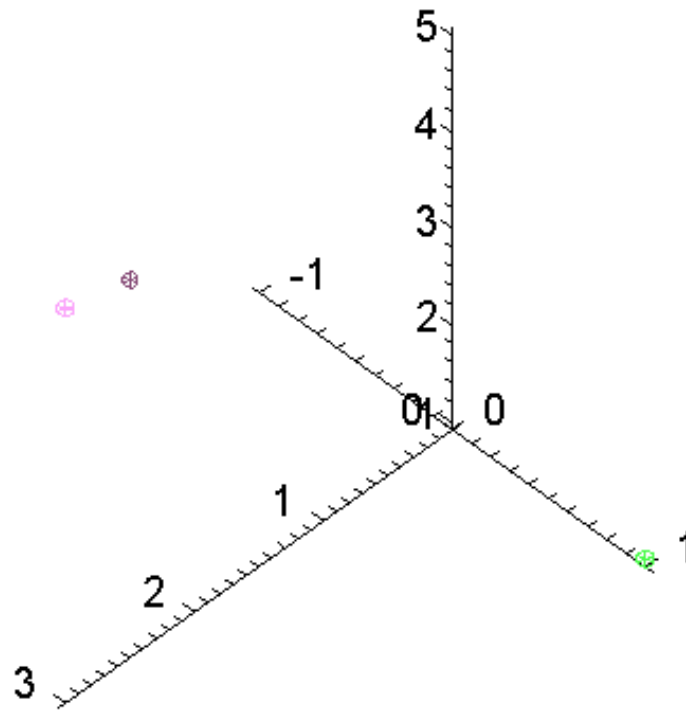
Трёхмерные команды пакета plots: `implicitplot3d` (неявно заданная поверхность)

```
> with(plots) : implicitplot3d( $x^2 + y^2 + z^2 = 4$ ,  $x = -2 .. 2$ ,  $y = -2 .. 2$ ,  $z = -2 .. 2$ ,  
scaling = constrained)
```



Трёхмерные команды пакета plots: pointplot3d (точки в пространстве)

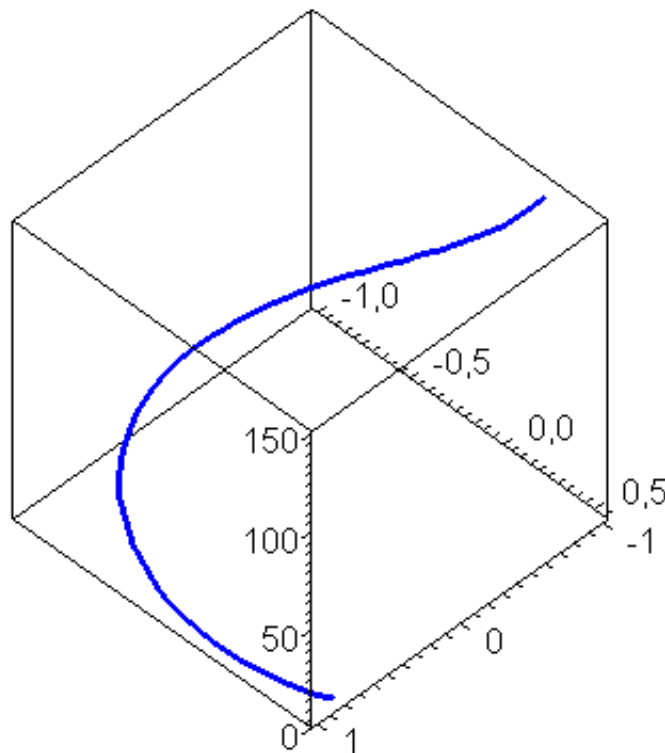
```
> pointplot3d( { [0, 1, 1], [1, -1, 2], [3, 0, 5] }, axes = normal, symbol = box);
```



Трёхмерные команды пакета plots: spacecurve

Для построения графика пространственной кривой, заданной параметрически, описание интервала изменения параметра должно находиться снаружи списка функций правых частей:

```
spacecurve([sin(t), cos(t), exp(t)], t = 1 .. 5, color = blue, thickness = 2, axes = BOXED);
```



Трехмерные графические структуры

Общие структуры для 2D и 3D

CURVES, POLYGONS, POINTS, TEXT – синтаксис аналогичен командам двумерного случая, у точек появляется третья координата

Некоторые трехмерные структуры

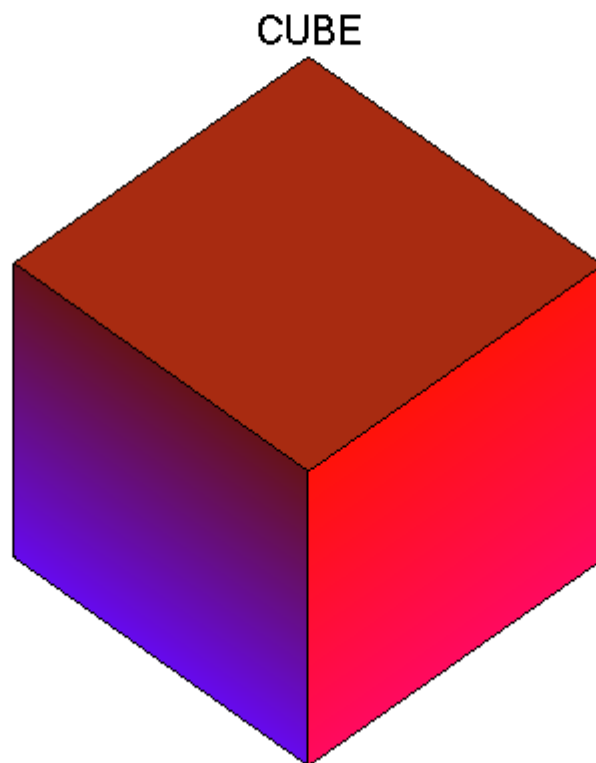
GRID(a..b, c..d, A) – задание поверхности над участком координатной плоскости $x=a\dots b$, $y=c\dots d$ при помощи двумерного массива A типа float[8].

MESH(A) – задание поверхности с помощью сетки значений, где A – трехмерный массив типа float[8]. Указание третьей координаты z позволяет задать неравномерную сетку.

Данные структуры выводятся с помощью команды **PLOT3D**, опции этой команды записываются большими буквами и имеют свой специальный синтаксис

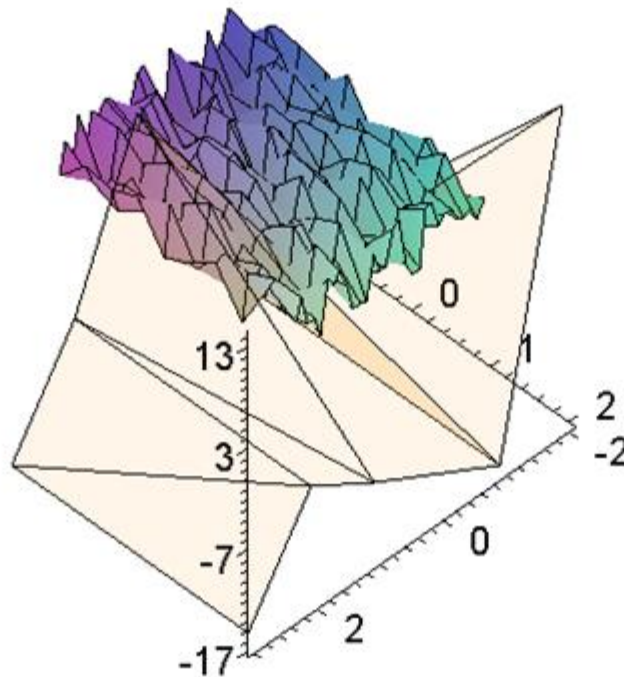
Трёхмерные графические структуры: POLYGONS

```
> PLOT3D(POLYGONS([[0, 0, 0], [1, 0, 0], [1, 1, 0], [0, 1, 0]], [[0, 0, 0], [0, 1, 0], [0, 1, 1], [0, 0, 1]], [[1, 0, 0], [1, 1, 0], [1, 1, 1], [1, 0, 1]], [[0, 0, 0], [1, 0, 0], [1, 0, 1], [0, 0, 1]], [[0, 1, 0], [1, 1, 0], [1, 1, 1], [0, 1, 1]], [[0, 0, 1], [1, 0, 1], [1, 1, 1], [0, 1, 1]]), LIGHT(0, 0, 0.0, 0.7, 0.0), LIGHT(100, 45, 0.7, 0.0, 0.0), LIGHT(100, -45, 0.0, 0.0, 0.7), AMBIENTLIGHT(0.4, 0.4, 0.4), TITLE(CUBE), STYLE(PATCH), COLOR(ZHUE))
```



Трёхмерные графические структуры: GRID и MESH

```
> A:=convert(evalm(linalg[randmatrix](30,10)/30),listlist):  
> g:=GRID(-2..1,-2..1,A):  
> m:=MESH([[-2,-1,-14],[-2,2,14]], [[-1,-1,-17],[-1,2,-17]  
], [[1,-1,10],[1,2,-10]], [[2,-1,-6],[2,2,-6]], [[3,-1,-16],  
[3,2,-16]]], GRIDSTYLE(TRIANGULAR), TRANSPARENCY(0.9)):  
> PLOT3D(g,m,AXESSTYLE(FRAME));
```



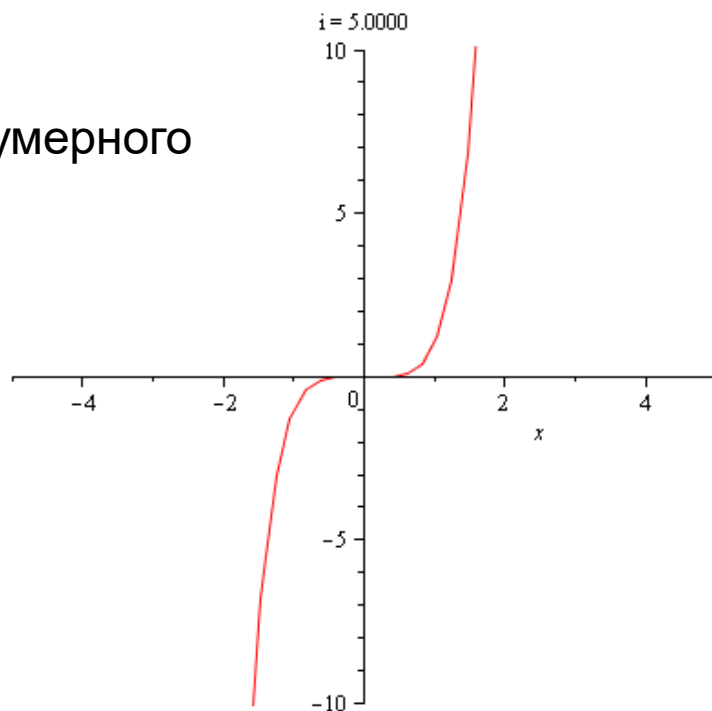
Анимация графиков

Анимация графиков

Для анимации стандартных двумерных или трехмерных графиков используется команда **animate** из пакета **plots** в виде **animate(plotcommand, plotargs, t=a..b, options)**, где **t** – параметр анимации, **plotcommand** – любая графическая команда (plot, plot3d, implicitplot, implicitplot3d, spacecurve и т.д.)

> *with(plots) : animate(plot, [xⁱ, x=-5..5, -10..10], i=1..5)*

Анимация двумерного
графика

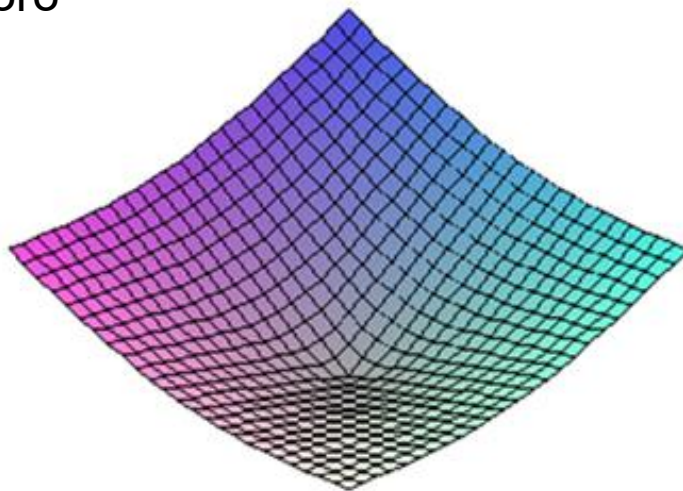


Анимация графиков: просмотр анимации

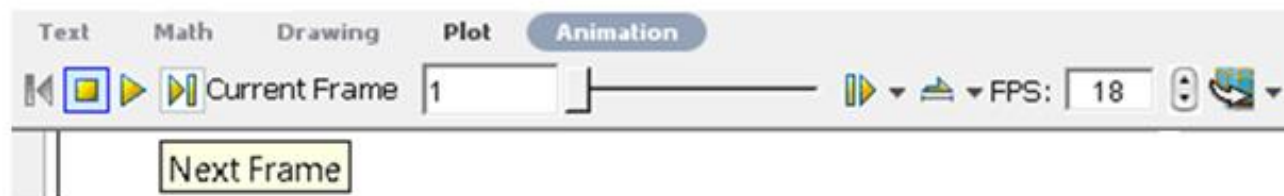
```
> with(plots) : animate( plot3d, [ sin(  $\frac{i \cdot \sqrt{x^2 + y^2}}{10}$  ), x = -6 .. 6, y = -6 .. 6 ], i = 1 .. 30 )
```

i = 1.

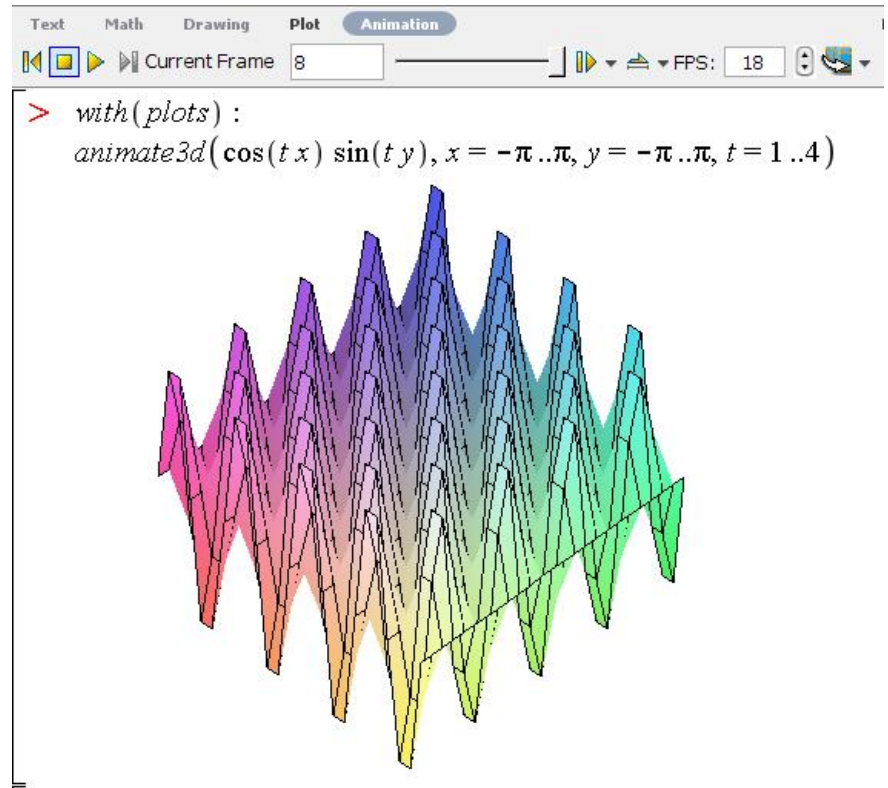
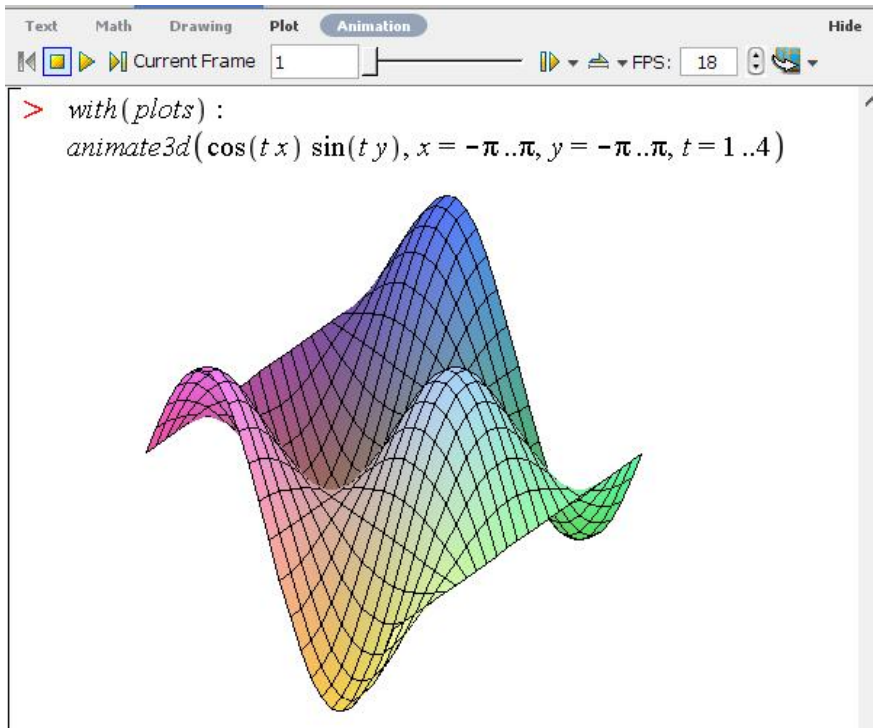
Анимация трехмерного
графика



Для просмотра анимации нажмите на график и используйте меню Animation сверху.



Устаревшая команда для анимации 3D графиков: `animate3d` из пакета `plots`

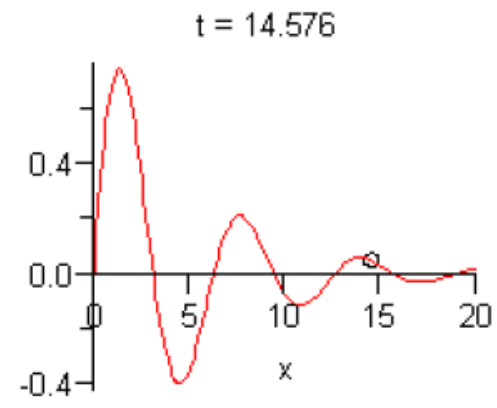
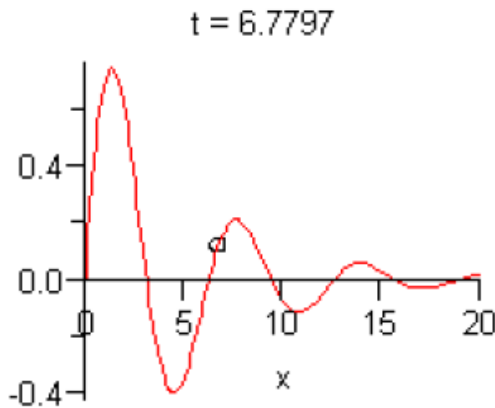
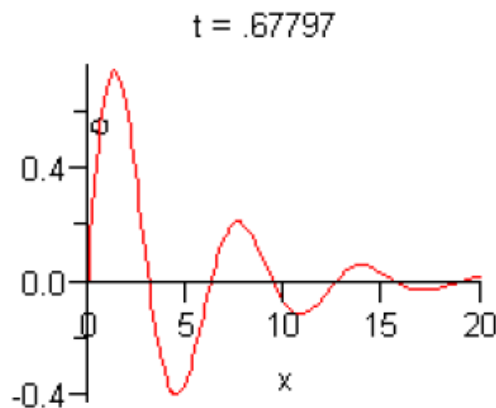


Настройка параметров анимации: количество фреймов, опция background

Пример. Анимация движения точки вдоль кривой, настройка количества фреймов

```
> sinewave := plot(sin(x) e-x/5, x = 0 .. 20) :
```

```
> animate(pointplot, [ [t, sin(t) e-t/5], symbol=circle, symbolsize=20],  
t=0 .. 20, frames=60, background=sinewave)
```



Пакеты графических команд

- Обзор команд пакета **plots**
- Обзор команд пакета **plottools**
- Иллюстративные графические команды других пакетов (student и др.)

Обзор команд пакета plots

> with(plots)

[animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot, display, fieldplot, fieldplot3d, gradplot, gradplot3d, graphplot3d, implicitplot, implicitplot3d, inequal, interactive, interactiveparams, intersectplot, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra_supported, polyhedraplot, rootlocus, semilogplot, setcolors, setoptions, setoptions3d, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d, tubeplot]

Обзор команд пакета plottools

```
> with(plottools)
```

```
[arc, arrow, circle, cone, cuboid, curve, cutin, cutout, cylinder, disk, dodecahedron, ellipse,  
ellipticArc, hemisphere, hexahedron, homothety, hyperbola, icosahedron, line, octahedron,  
parallelepiped, pieslice, point, polygon, project, rectangle, reflect, rotate, scale, semitorus,  
sphere, stellate, tetrahedron, torus, transform, translate, vrm] ]
```

Создание геометрических объектов

<u>arc</u>	<u>arrow</u>	<u>circle</u>	<u>cone</u>	<u>cuboid</u>
<u>curve</u>	<u>cutin</u>	<u>cutout</u>	<u>cylinder</u>	<u>disk</u>
<u>dodecahedron</u>	<u>ellipse</u>	<u>ellipticArc</u>	<u>hemisphere</u>	<u>hexahedron</u>
<u>hyperbola</u>	<u>icosahedron</u>	<u>line</u>	<u>octahedron</u>	<u>parallelepiped</u>
<u>pieslice</u>	<u>point</u>	<u>polygon</u>	<u>rectangle</u>	<u>semitorus</u>
<u>sphere</u>	<u>tetrahedron</u>	<u>torus</u>		

круг без заливки (указывает на circle)

круг с заливкой (указывает на disk)

многоугольник (указывает на polygon)

прямоугольник (указывает на rectangle)

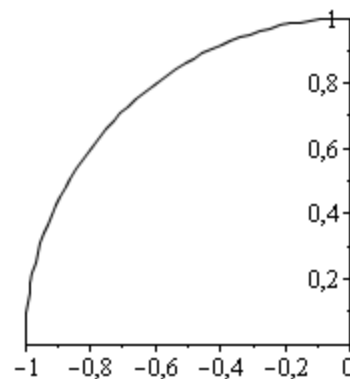
Изменение геометрических объектов

```
homothety project reflect rotate scale  
stellate transform translate
```

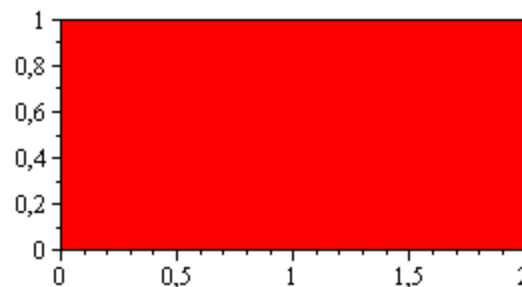
Возможности пакета `plottools`: примеры двумерных графических объектов

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

```
> display( arc( [0, 0], 1,  $\frac{\pi}{2} .. \pi$  ) )
```



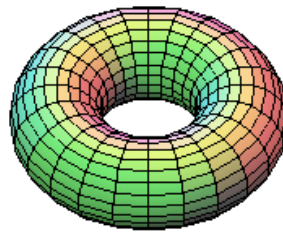
```
> display( rectangle( [0, 1], [2, 0], color = red ), scaling = constrained )
```



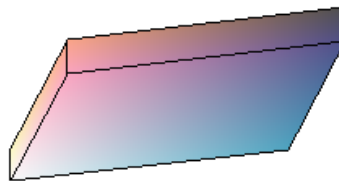
Возможности пакета `plottools`: примеры трехмерных графических объектов

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

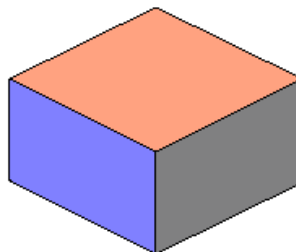
```
> display(torus([1, 1, 1], 1, 2), scaling = constrained, lightmodel = light1, shading = zgrayscale)
```



```
> display(parallelepiped([0, 0, 1], [0, 1, 1], [1, 1, 1], [0, 1, 0]), orientation = [117, -31],  
          scaling = unconstrained)
```

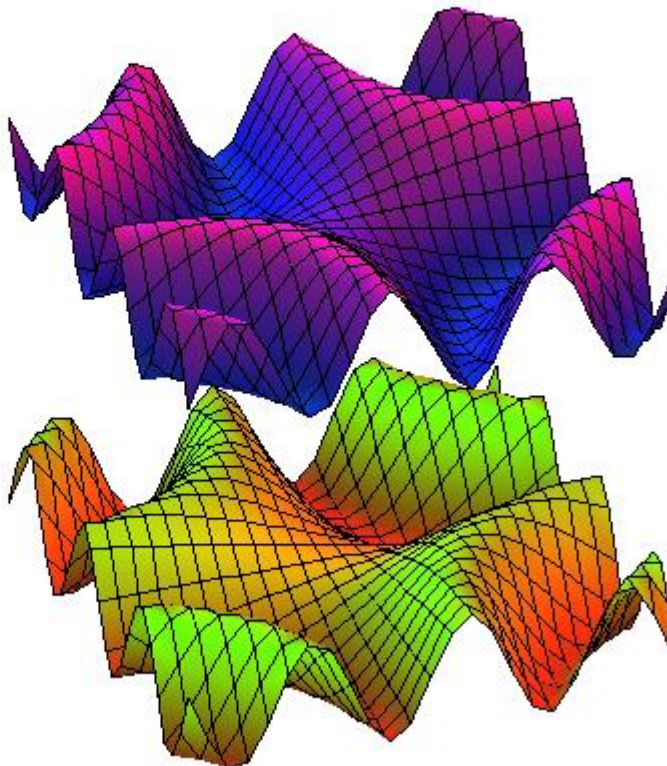


```
> display(cuboid([0, 0, 0], [1, 1, 1]), lightmodel = light2, shading = none)
```



Возможности пакета `plottools`: примеры манипуляций с графическими объектами

```
> with(plottools) : with(plots) :  
> p := plot3d(sin(x y) + 3, x = -π..π, y = -π..π) :  
  q := reflect(p, [[0, 0, 0], [1, 0, 0], [0, 1, 0]]) :  
> display([p, q], lightmodel = light1, orientation = [20, -120], shading = zhue)
```



Иллюстративные графические команды других пакетов Maple

Графические команды входят в состав многих пакетов Maple. Обычно они иллюстрируют результаты работы математических команд этих пакетов. Некоторые примеры таких графических команд:

Обучающий пакет student

showtangent – построение касательной к графику

Пакет для решения дифференциальных уравнений DEtools

phaseportrait – изображение фазового портрета системы двух уравнений первого порядка

Пакет для работы с графами networks

draw – изображение графа

Статистический пакет stats

histogram – построение гистограммы для набора данных

Использование пакета student: касательная к графику

```
[> with(student) :  
=> y := x^2 + x - 1 :  
=> showtangent(y, x = 4.5, thickness = 2)
```

