Некоторые относительно новые возможности команды **plot**

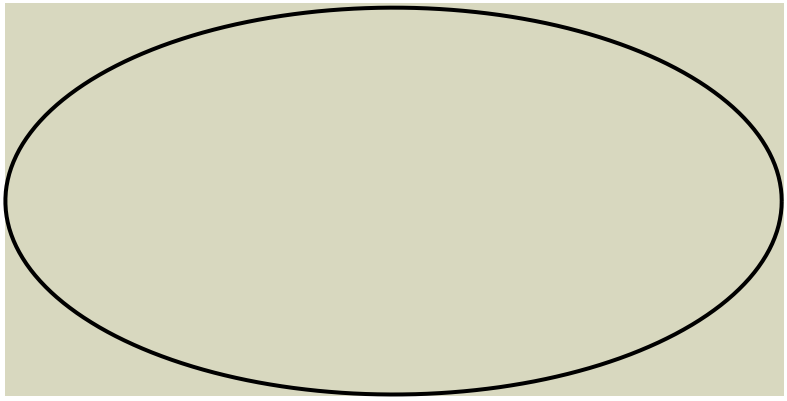
В этом документе приводятся некоторые из опций команды **plot**, которые появились в системе компьютерной математики уже после выхода в свет Maple 17, то есть за последние 5 лет.

```
> with(ColorTools):
```

Maple 18

Именно в этой версии появилась возможность задавать фон, на котором выводится график интересующей нас зависимости. Появилась новая опция *background*. В простейшем случае значением этой опции может быть имя одного из известных системе цветов.

```
> plot([2*cos(t),sin(t),t=0..2*Pi],scaling=constrained,
      axes=none,color=black,background=wheat);
```



Возможности работы с цветом были значительно расширены ещё в Maple 16 путём добавления в систему нового пакета **ColorTools**. Средства этого пакета позволяют работать с цветами из нескольких встроенных палитр, а также создавать собственные цветовые палитры.

Список цветов в строковом формате, которые доступны без обращения к палитрам, можно узнать, обратившись к процедуре *GetColorNames*. У этой процедуры есть опция *new*. Установив значение этой опции в *false,* получаем список цветов, которые известны Maple на протяжении более чем двух десятилетий. Как правило, данные цвета доступны по имени, как в предыдущем примере, и могут использоваться без кавычек.

```
> GetColorNames('new'=false);
```

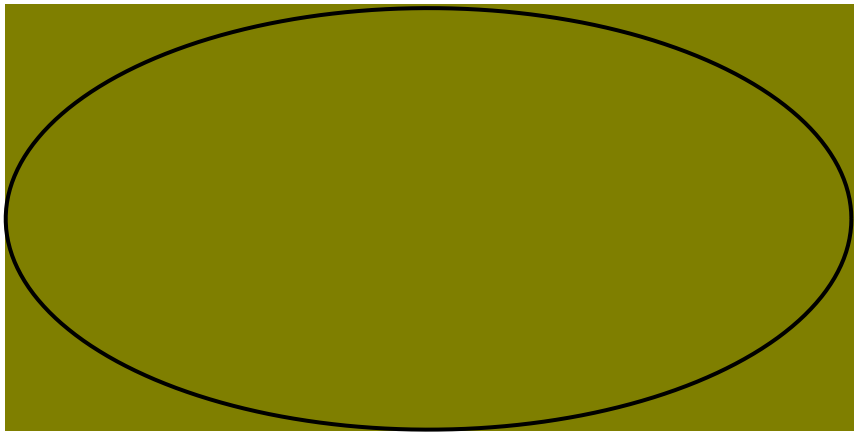
```
["aquamarine", "black", "blue", "brown", "coral", "cyan", "gold", "gray", "green", "khaki", "magenta",
    "maroon", "navy", "orange", "pink", "plum", "red", "sienna", "tan", "turquoise", "violet", "wheat",
    "white", "yellow", "grey"]
```

В новом наборе содержится гораздо больше цветов. Для проверки данного утверждения следующую команду следует завершить не двоеточием а точкой с запятой.

```
> GetColorNames('new' = true):
```

Пример одного из новых цветов

```
> plot([2*cos(t),sin(t),t=0..2*Pi],scaling=constrained,
      axes=none,color=black,background="Olive");
```



В качестве фона теперь можно использовать файлы с различными изображениями.
Несколько изображений идёт в комплекте с самой системой компьютерной алгебры.
По умолчанию рисунок будет принимать размеры фонового изображения.

```
> plot([2*cos(t),sin(t),t=0..2*Pi],scaling=constrained,axes=none,
  color=black,background=cat(kernelopts(datadir),"/images/tree.jpg"));
```

Фоновые изображения можно создавать средствами пакета **ImageTools**.

```
> with(ImageTools):
```

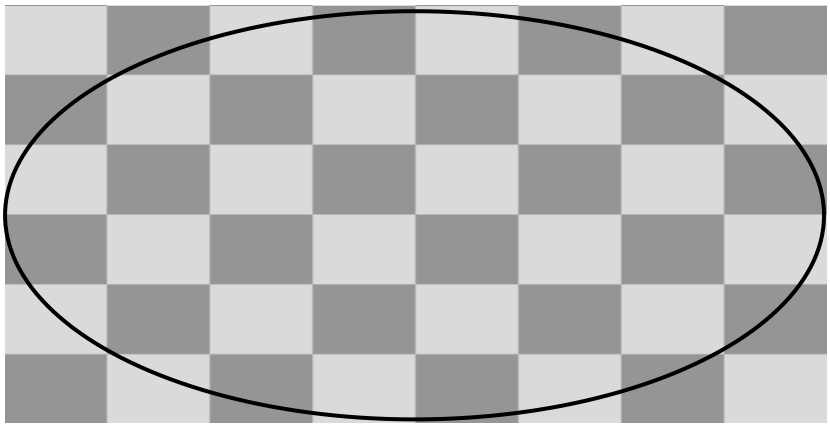
В этом пакете есть команда рисования шахматной доски с заданным количеством клеток по каждому направлению. Первый параметр *Checkerboard* является размером одной клетки доски в пикселях.
Второй и третий параметры обозначают количество клеток по каждому направлению. Размер сгенерированного массива в точности соответствует заданным параметрам команды.

```
> BackImg:=Checkerboard(40,6,8,foreground=.7,background=.3);
```

$$BackImg := \begin{bmatrix} 1..240 \times 1..320\ Array \\ Data\ Type: float_8 \\ Storage: rectangular \\ Order: C\_order \end{bmatrix}$$

Синтезированное изображение может быть использовано в качестве фона.

```
> plot([2*cos(t),sin(t),t=0..2*Pi],scaling=constrained,axes=none,
  color=black,background=BackImg);
```



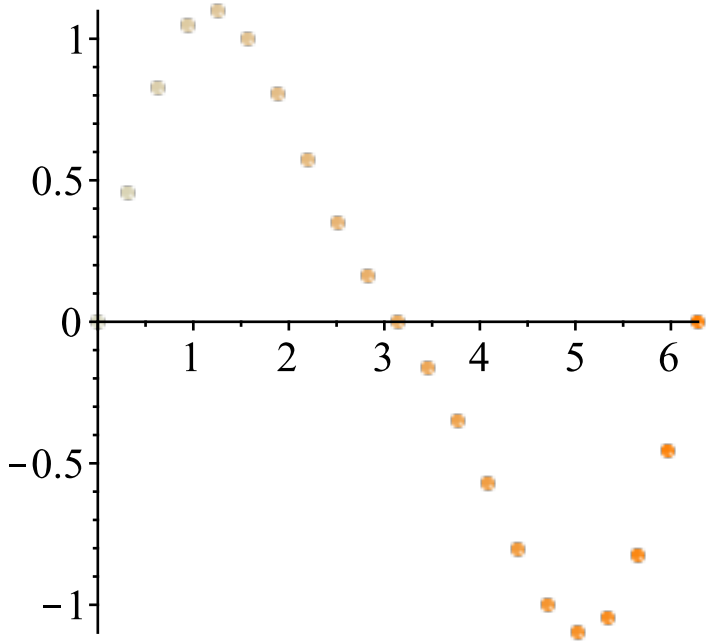
С цветами тесно связана ещё одна опция, введённая в Maple 18. Опция *colorscheme* позволяет применить цветовую схему к кривым и наборам точек, отображаемым с помощью команды **plot**.

Простейший способ применения этой опции состоит в том, чтобы присвоить ей в качестве значения список из двух цветов. Цвета можно задавать в любом виде, который распознаёт опция *color*.

```
> Data:=[seq([Pi*j/10,sin(Pi*j/10)+sin(Pi*j/5)/4],j=0..20)]:
```

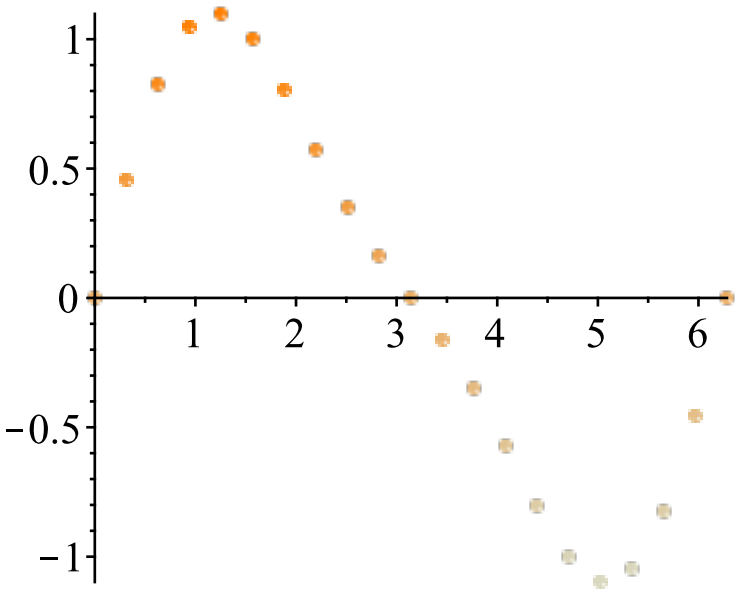
Простой пример со старым набором цветов. В этом примере цвет точек изменяется от *wheat* к *coral* по мере увеличения аргумента.

```
> plot(Data,style=point,symbol=solidcircle,symbolsize=14,
        colorscheme=[wheat,coral]);
```



Более сложная форма вызова позволяет явно указать одну из встроенных цветовых схем. Имя схемы должно передаваться в виде строки, то есть заключаться в двойные кавычки.

```
> plot(Data,style=point,symbol=solidcircle,symbolsize=14,
       colorscheme=["ygradient",[wheat,coral]]);
```

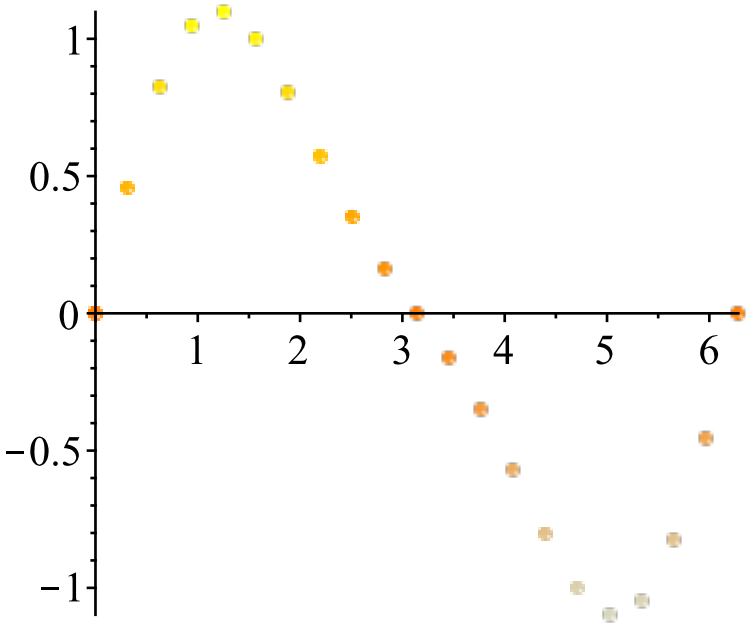


Если этого не сделать, то возникает ошибка.

```
> plot(Data,style=point,symbol=solidcircle,symbolsize=14,
       colorscheme=[ygradient,[wheat,coral]]);
Error, (in plot) incorrect specification of colorscheme
```

Количество цветов в списке, который является значением опции *colorscheme*, может быть больше двух.

```
> plot(Data,style=point,symbol=solidcircle,symbolsize=14,
       colorscheme=["ygradient",[wheat,coral,yellow]]);
```

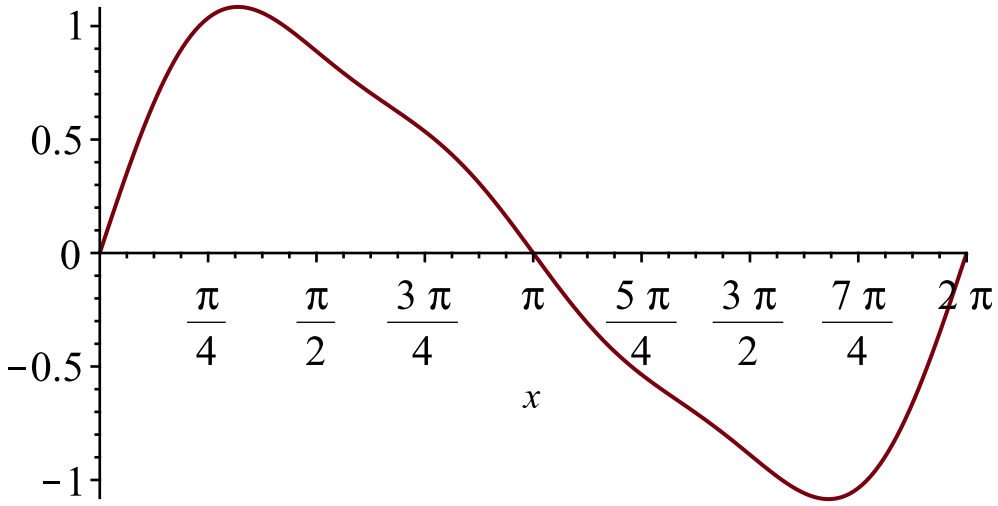


Когда важны точные размеры рисунка, то чрезвычайно полезной оказывается опция *size*. Её значением является список из двух величин. В первом элементе списка содержится информация о ширине рисунка, а во втором – о высоте.

Ширина рисунка должна представлять собой либо положительное число, либо строку *"default"*. Аналогичное соглашение касается и высоты рисунка. Однако в этом случае кроме *"default"* допускаются ещё строки *"golden"* или же *"square"*.

В документации к системе предупреждается, что если по крайней мере один элемент списка установлен в *"default"*, то размер рисунка по умолчанию может оказаться зависящим от используемого графического интерфейса. В результате экспорта приводимого ниже рисунка в PNG файл в 64-битовой версии Maple под Windows на диске появился файл с размерами 400 × 200 пикселей.

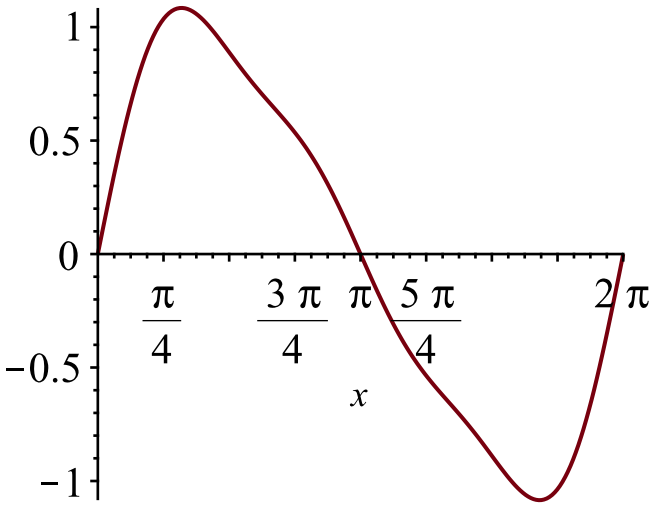
```
> plot(sin(x)+sin(2*x)/4+sin(3*x)/9,x=0..2*Pi,size=[default,200]);
```



Если элемент списка представляет собой число, превышающее 10, то это означает, что соответствующий параметр рисунка задан в пикселях. Если же ширина рисунка задана в виде положительного числа меньше 10, то размер в пикселях получается умножением данной величины на ширину окна рабочего листа, в котором отображается данный рисунок.

Так выглядит рисунок, ширина которого составляет половину ширины окна рабочего листа Maple.

```
> plot(sin(x)+sin(2*x)/4+sin(3*x)/9,x=0..2*Pi,size=[0.5,200]);
```

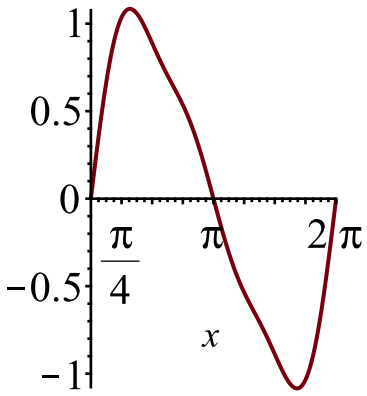


Когда второй элемент списка оказывается числом, меньшим 10, то высота рисунка определяется путём умножения этого числа на ширину. Рисунок с одинаковыми значениями ширины и высоты получается при условии, что второй элемент списка равен 1. Именно для этой ситуации и предназначено значение *"square"*. В свою очередь, значение *"golden"* приводит к отношению высоты рисунка к ширине равной $\frac{(\mathrm{sqrt}(5)-1)}{2}$.

$$\frac{\sqrt{5}}{2}-\frac{1}{2}$$

Приводимый ниже пример иллюстрирует как выглядит рисунок при выборе значения *"square"*.

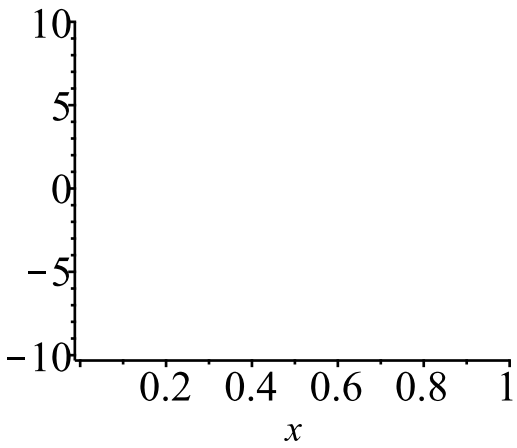
```
> plot(sin(x)+sin(2*x)/4+sin(3*x)/9,x=0..2*Pi,size=[0.3,1]);
```



Maple 2015

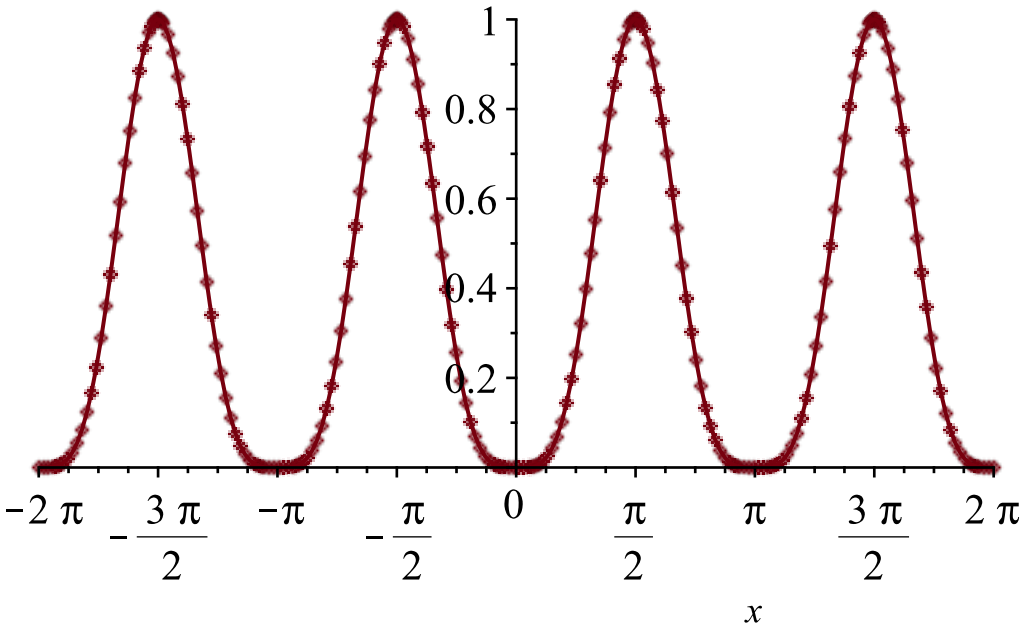
Различные новшества в области визуализации практически обошли стороной команду **plot**. Стоит упомянуть лишь возможность создания пустых рисунков. Раньше Maple ругался на отсутствие первого аргумента команды. Теперь это стало допустимо.

```
> plot(x=0..1,axes=frame);
```



Именно в этой версии у опции *style* появилось новое значение *pointline*. Теперь стало возможно изображать как саму кривую, так и принадлежащие ей точки.

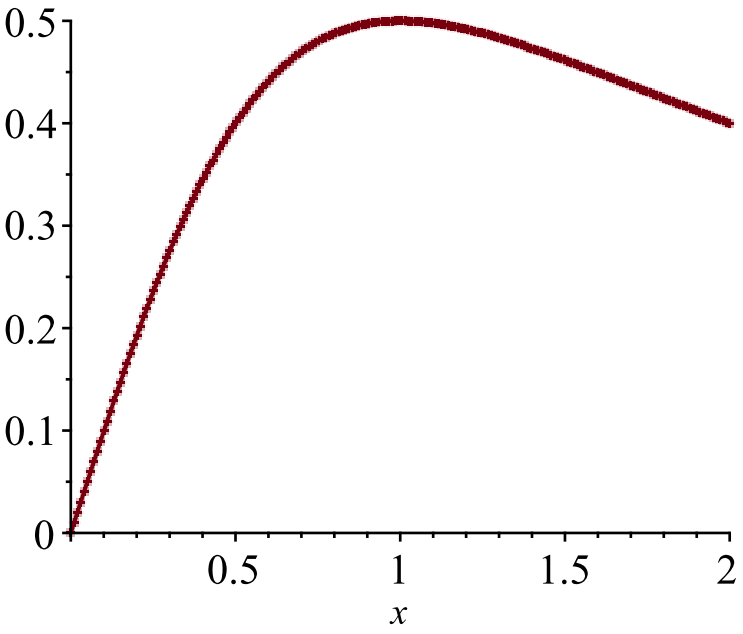
```
> plot(sin(x)^4,'style'=pointline,size=[400,250]);
```



Однако непонятен механизм расстановки точек и графический образ,

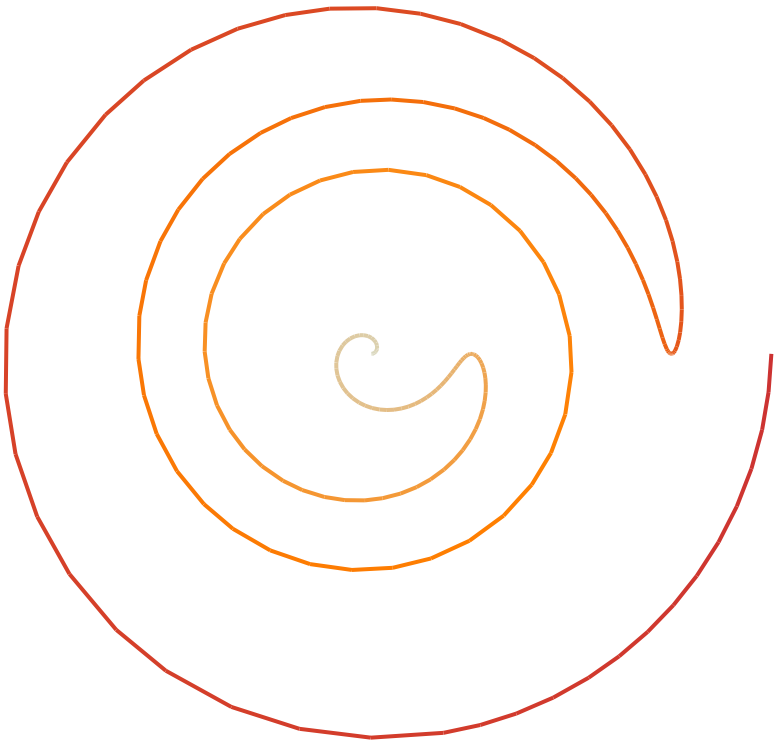
которым они выводятся. В результате во многих случаях они просто не видны.

```
> plot(x/(1+x^2),x=0..2,'style'=pointline,size=[300,250]);
```



Опция *colorscheme,* которая в предыдущей версии Maple работала только для поверхностей и наборов точечных данных, теперь применима и для плоских кривых.

```
> plot([rho,2*Pi*sin(rho),rho=0..2*Pi],coords=polar,
       scaling=constrained,axes=none,size=[300,300],
       colorscheme=[wheat,coral,orange]);
```

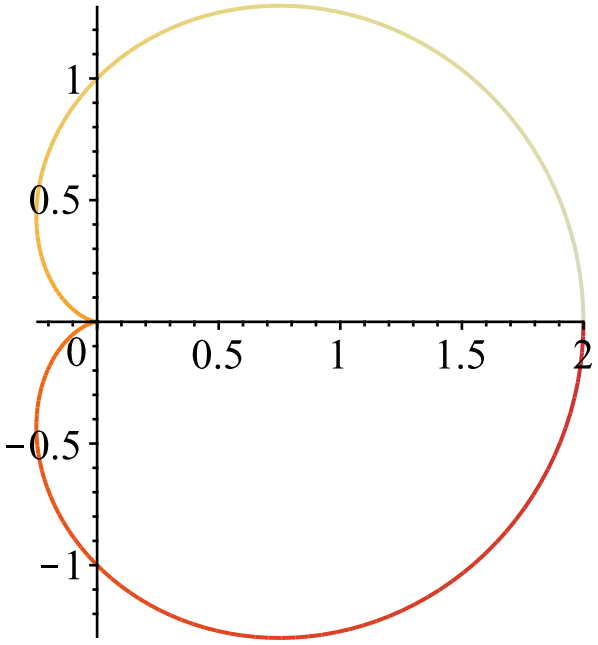


Более того, появилась возможность использовать различные цветовые пространства, поддерживаемые пакетом **ColorTools**.

Перечень цветовых пространств включает в себя:

– RGB, – CMYK, – Luv,
– HSV, – XYZ, – LCHab,
– xyY, – Lab, – LCHuv.

```
> plot(1+cos(phi),phi=0..2*Pi,coords=polar,scaling=constrained,
       size=[250,250],colorscheme=["linear",[wheat,coral,orange],
       colorspace="HSV"]);
```

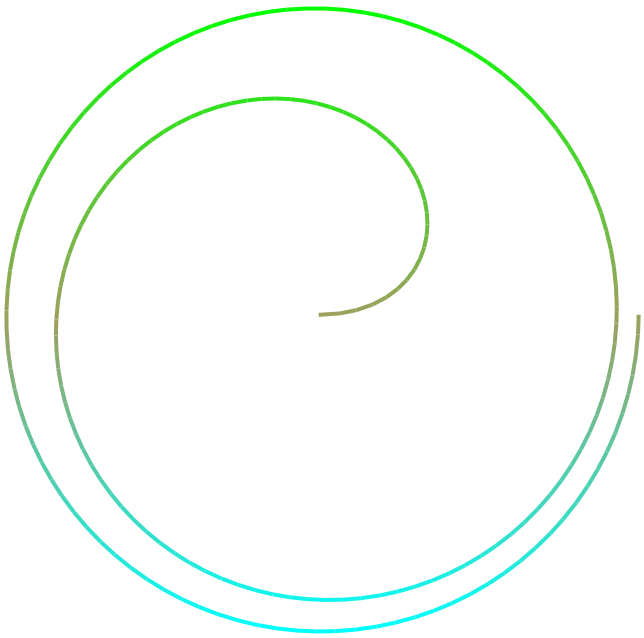


Maple 2016

В этой версии Maple были завершены работы по систематическому внедрению опции *colorscheme* во все графические команды, в том числе и в **plot**.

Помимо известных ранее, были добавлены новые градиентные схемы *"xgradient"* и *"ygradient"*. Они аналогичны существовавшей в предыдущей версии Maple схемы *"zgradient"*. Однако теперь закраска осуществляется в соответствии со значениями *x* и *y*, а не со значениями *z* как было ранее.

```
> plot(phi/(1+phi),phi=0..4*Pi,coords=polar,scaling=constrained,
      axes=none,size=[250,250],colorscheme=["ygradient",
      [cyan,khaki,green]]);
```



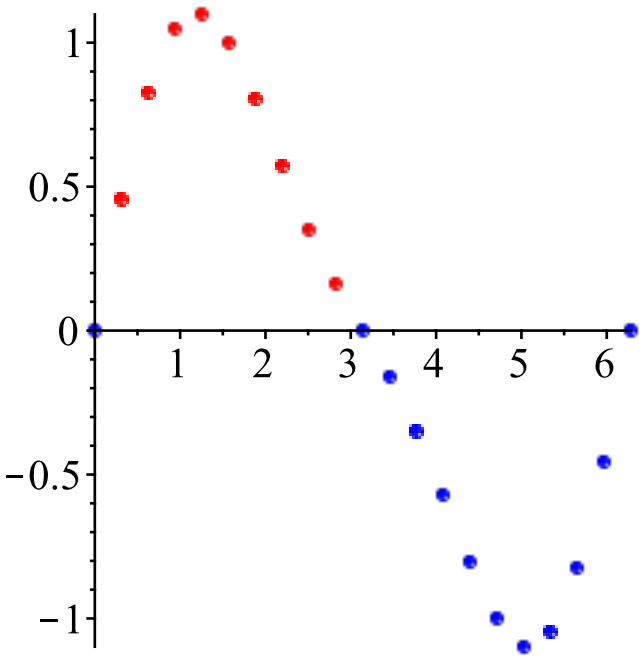
Новая цветовая схема *"valuesplit"* позволяет визуально разделить набор точечных данных на части, отображая различные точки цветом, зависящим от численных значений, которые данная точка представляет.

Простейший способ использования данной схемы состоит в том, чтобы после названия схемы указать список, каждый член которого является уравнением. В левой части уравнения задаётся числовой диапазон. В правой части − цвет, в который будут окрашиваться точки, если их ординаты попадают в заданный диапазон.

В приводимом ниже примере точки с положительными значениями *y* выводятся красным цветом, а точки, у которых ордината равна 0 или же

отрицательна – синим.

```
> plot(Data,style=point,symbol=solidcircle,symbolsize=14,
      colorscheme=["valuesplit",[-2..0=blue,0..2=red]]);
```



Полезной особенностью данной цветовой схемы является возможность указать в качестве последнего элемента списка просто цвет без соответствующего диапазона. Этим цветом и будут окрашиваться точки, ординаты которых не попали ни в один из предыдущих диапазонов.

Модифицируем предыдущий пример так, чтобы точки, у которых ординаты по модулю оказываются менее чем $\frac{1}{2}$, окрашивались в зелёный цвет.

```
> plot(Data,style=point,symbol=solidcircle,symbolsize=14,
      colorscheme=["valuesplit",[-3/2..-1/2=blue,
      1/2..3/2=red,green]]);
```