

## Редко используемые опции команды `plot`

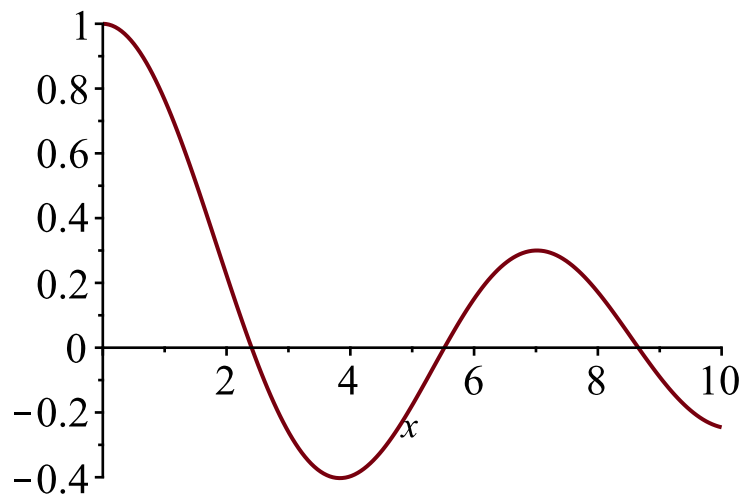
> `with(ColorTools) :`

В этом документе собраны некоторые опции команды `plot`, которые на практике используются достаточно редко. Одной из возможных причин сложившейся ситуации может быть то, что некоторые из этих опций появились в Maple совсем недавно и ещё не получили подробного освещения в учебной литературе. Вторая предполагаемая причина кроется в том, что часть опций команды `plot` предназначена для решения весьма специфических задач и в большинстве практических приложений без них можно обойтись. Тем не менее, в отдельных случаях именно благодаря использованию рассматриваемых здесь опций удаётся построить качественное изображение.

В справочной системе Maple возможности цветовой схемы `"valuesplit"` иллюстрируются примерами раскраски поверхностей и дискретных наборов точек. В то же время эта схема с успехом применима и для выделения определённых областей на обычных двумерных графиках.

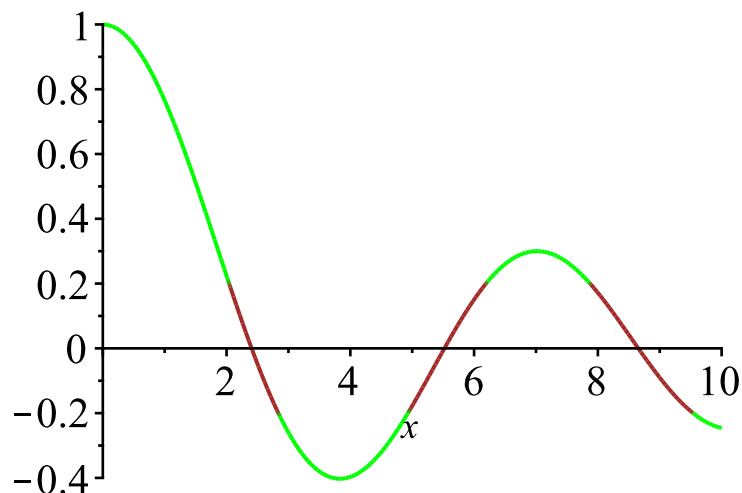
Построим, например, график функции Бесселя нулевого порядка.

> `plot(BesselJ(0,x), x=0..10, size=[300,200]) ;`



Предположим, что нам понадобилось закрасить те части графика, где функция по модулю не превышает 0.2, более тёмным цветом, а оставшуюся часть перекрасить в зелёный цвет. Тогда мы можем применить опцию `colorscheme`, указав в ней цветовую схему `"valuesplit"` с дополнительным параметром `transformdata`.

> `plot(BesselJ(0,x), x=0..10, size=[300,200], colorscheme=["valuesplit", [0..0.2=brown, 0.2..1=green], transformdata=abs)) ;`



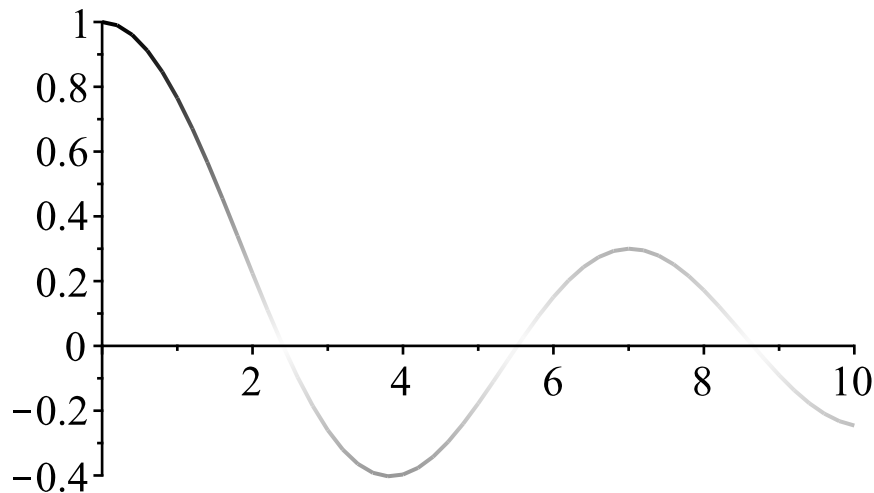
Опция `colorscheme` с цветовой схемой `"valuesplit"` имеет следующий синтаксис: `colorscheme=["valuesplit", V, mapping]`. Необязательный параметр `mapping` может представлять собой не только список, как в

приведенном выше примере, но и функцию, которая имеет один аргумент. Данная функция должна возвращать результат типа *Color*.

```
> TrGray:=x->Color("RGB",[1-abs(x),1-abs(x),1-abs(x)]);  
TrGray := x ↦ ColorTools:-Color("RGB",[1-|x|,1-|x|,1-|x|])
```

С помощью функции *TrGray* можно отобразить график функции Бесселя в оттенках серого. Те части графика, где функция близка к нулю будут выглядеть совсем светлыми. Чем сильнее значения функции отличаются от 0, тем темнее будет цвет графика.

```
> Data:=seq([k/5.,BesselJ(0,k/5.)],k=0..50):  
> plot(Data,size=[350,200],colorscheme=["valuesplit",  
map(z->z[2],Data),TrGray]);
```



В Maple 2017 у команды **plot** появилась новая опция *annotation*. С помощью этой опции можно добавить к рисунку поясняющий текст, который будет отображаться в тот момент, когда курсор мыши пересекает график плоской кривой. В простейшем случае значением этой опции является просто текстовая строка.

```
> plot(BesselJ(0,x),x=0..10,size=[300,200],  
annotation="Функция Бесселя");
```

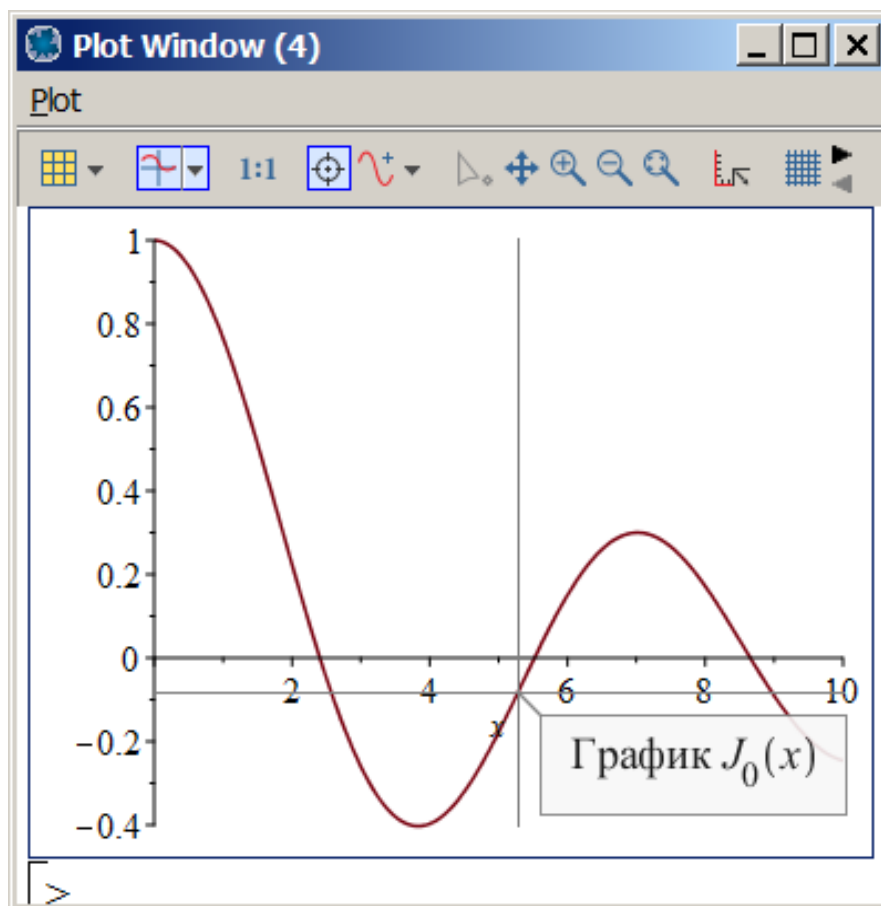
При экспорте документа в другие форматы отображаемая на экране аннотация не сохраняется, поэтому её пришлось вставить как обычный рисунок. Если выполнить команду **plotsetup(maplet)**, то в окне Maplet приложения аннотация не появляется.



Если же перенаправить вывод графиков в другое окно Maple, выполнив команду **plotsetup(window)**, то с аннотацией всё будет в порядке.

В приводимом ниже примере показано, что команда **typeset** может использоваться при создании аннотаций.

```
> plot(BesselJ(0,x),x=0..10,size=[300,200],  
annotation=typeset("График ",J[0](x)));
```



В системе Maple уже достаточно давно существует возможность тонкой настройки вида координатных осей. Для этого предназначена опция *axis*. Её значением является список вспомогательных опций.

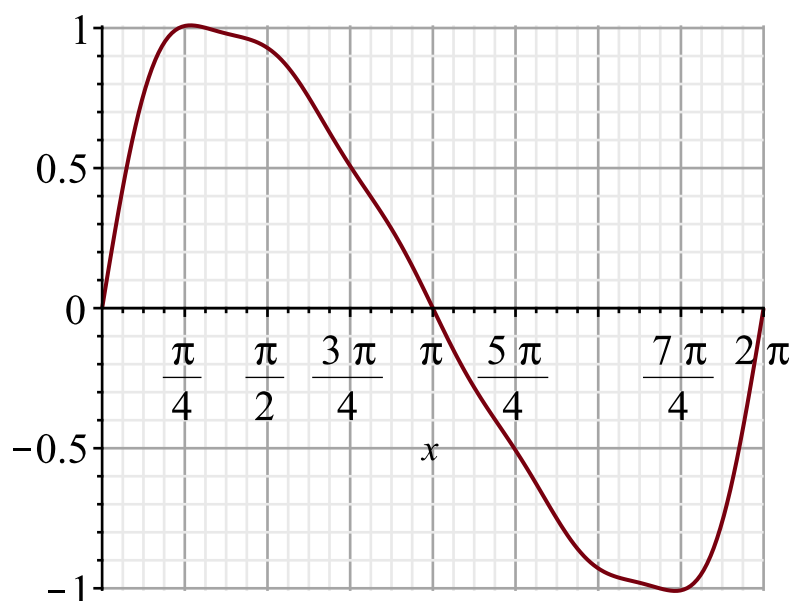
На примере построения графика простой функции рассмотрим некоторые из имеющихся возможностей.

```
> C1:=x->add(sin(k*x)/k^2,k=1..5);
```

$$C1 := x \mapsto \text{add}\left(\frac{\sin(kx)}{k^2}, k=1..5\right)$$

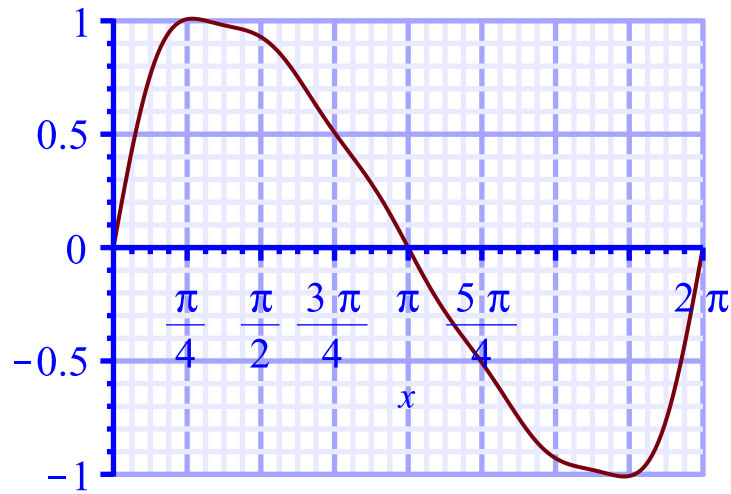
Сначала построим график когда опции выбраны по умолчанию.

```
> plot(C1(x), x=0..2*Pi, gridlines=true, size=[320,240]);
```



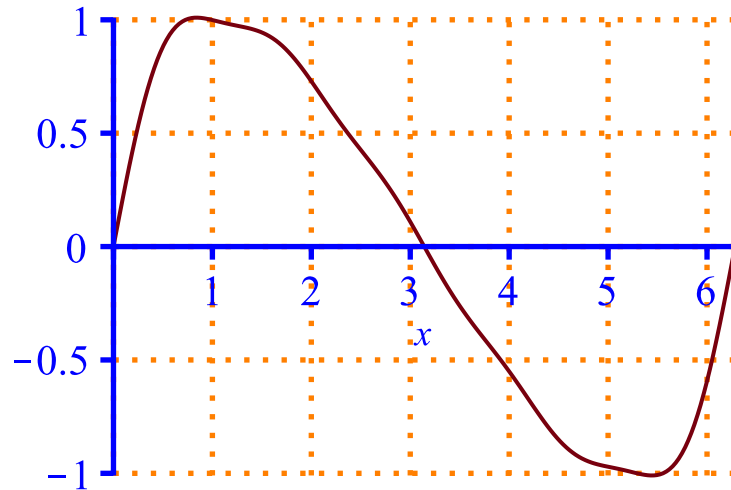
Попробуем теперь сделать координатные оси потолще, а заодно изменим их цвет на синий.

```
> plot(C1(x), x=0..2*Pi, gridlines=true, size=[300,200],
axis=[color=blue, thickness=2]);
```



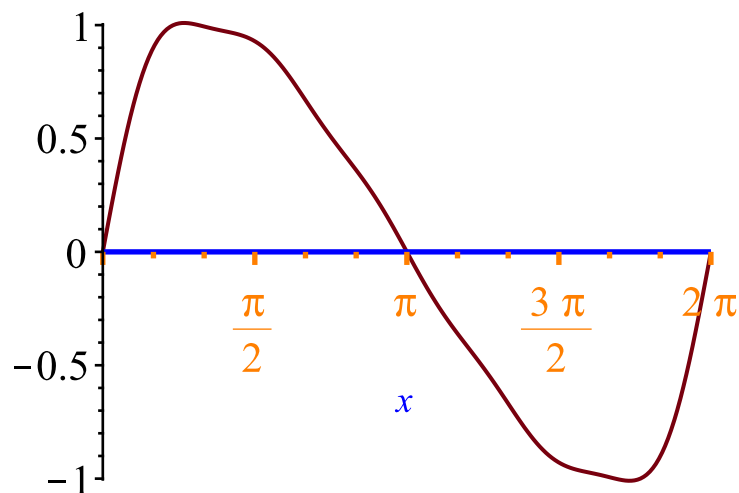
Как мы видим, изменение толщины координатных осей автоматически изменяет и толщину линий сетки. Однако если опцию *gridlines* переместить внутрь *axis*, то можно добиться независимой настройки параметров сетки.

```
> plot(Cl(x), x=0..2*Pi, size=[300,200], axis=[color=blue,
    thickness=2, gridlines=[6, color=coral, linestyle=dot]]);
```



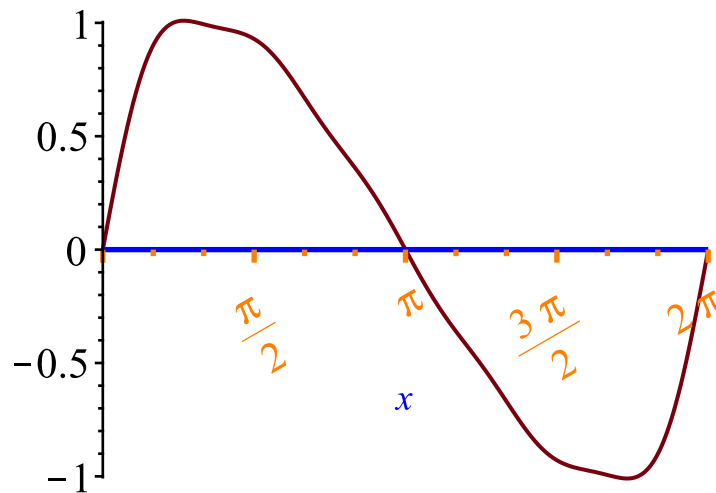
Есть возможность настраивать параметры только какой-то одной оси, указав её индекс: 1 – для горизонтальной оси и 2 – для вертикальной.

```
> plot(Cl(x), x=0..2*Pi, size=[300,200], axis[1]=[color=blue,
    thickness=2, tickmarks=[spacing(Pi/2), subticks=2, color=coral]]);
```



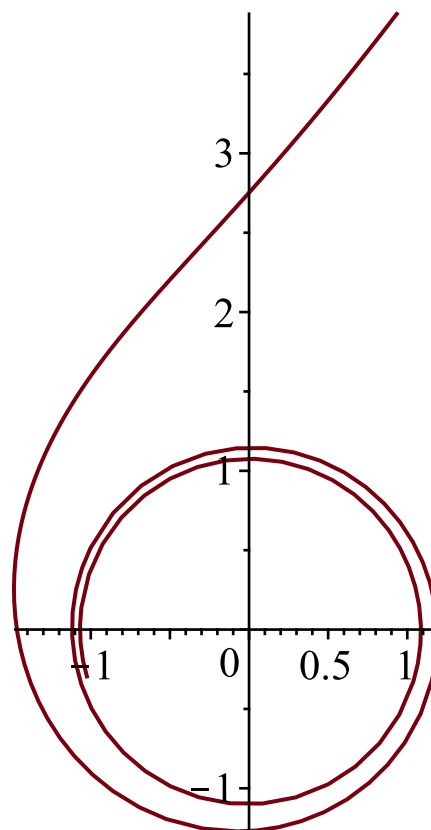
В Maple 2018 наконец-то была встроена поддержка наклонного текста. Теперь команда **plot** может писать масштабные метки под углом.

```
> plot(C1(x), x=0..2*Pi, size=[300,200], axis[1]=[thickness=2,
    tickmarks=[spacing(Pi/2), subticks=2, color=coral, rotation=Pi/6]]);
```



Качество изображений, которые строит система компьютерной алгебры, существенно зависит от минимального количества точек, используемых для построения графика. В последних версиях Maple эта величина равна 200, что позволяет без дополнительных настроек изображать довольно-таки замысловатые кривые.

```
> plot([rho, rho/(rho-1), rho=16/15..4], coords=polar,
    scaling=constrained, size=[320,320]);
```

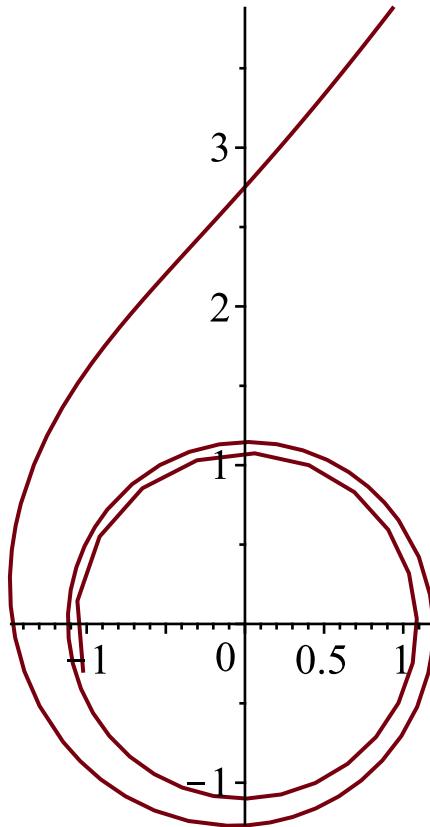


Пользователь имеет возможность указать минимально необходимое для построения графика количество точек с помощью опции *numpoints*. Если на промежутке между двумя соседними точками график достаточно сильно отличается от прямой, то промежуток делится на части и производятся дополнительные вычисления, при условии, что включена опция *adaptive*.

В старые добрые времена, когда компьютеры были не такие быстрые, по умолчанию использовалось значение *numpoints*=50. Для показанного ниже графика этого значения недостаточно. Об этом свидетельствуют неестественные изломы, которых попросту не должно быть.

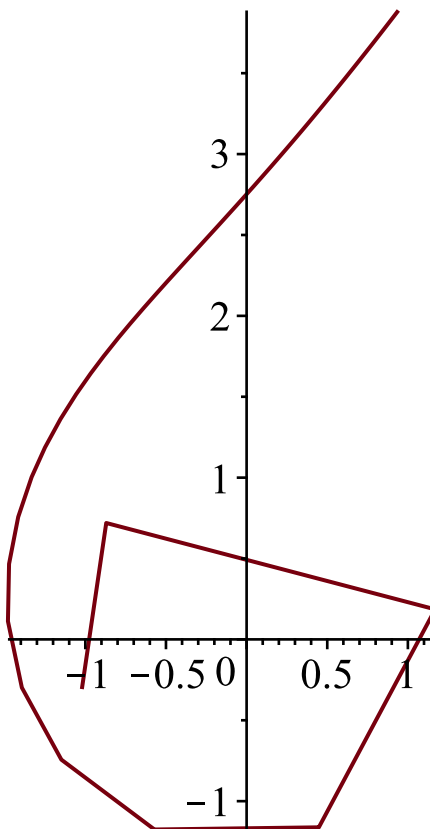
```
> plot([rho, rho/(rho-1), rho=16/15..4], coords=polar,
```

```
scaling=constrained,numpoints=50,size=[320,320]);
```



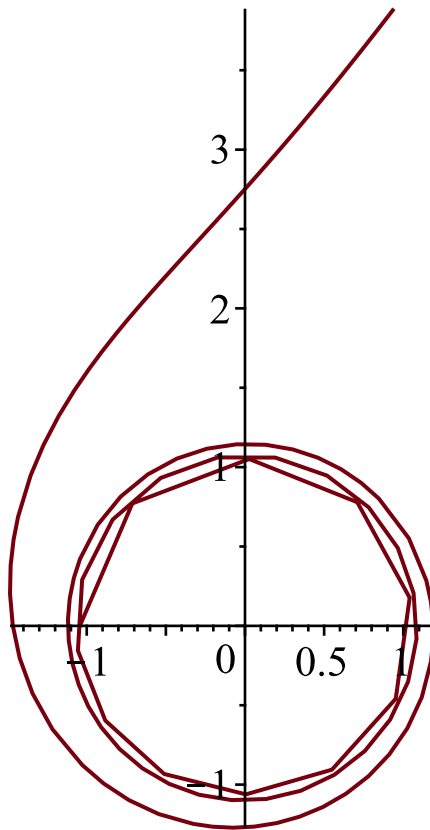
Ситуация становится совершенно нетерпимой, если ещё отключить адаптивную схему рисования.

```
> plot([rho,rho/(rho-1),rho=16/15..4],coords=polar,adaptive=false,  
scaling=constrained,numpoints=50,size=[320,320]);
```



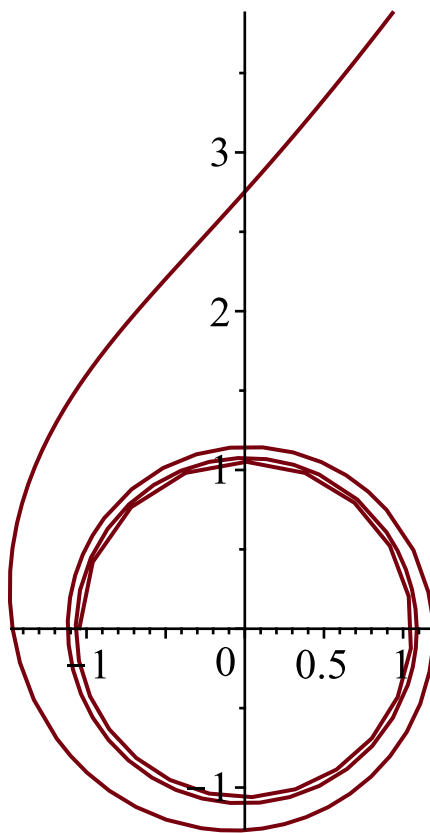
Однако в сложных ситуациях настройка вышеуказанных параметров по-прежнему необходима. Например, в следующем рисунке 50 точек явно недостаточно даже при включённой адаптивной схеме.

```
> plot([rho,rho/(rho-1),rho=22/21..4],coords=polar,numpoints=50,  
scaling=constrained,size=[320,320]);
```



Можно проследить, как меняется качество изображения при увеличении значения опции *numpoints*.

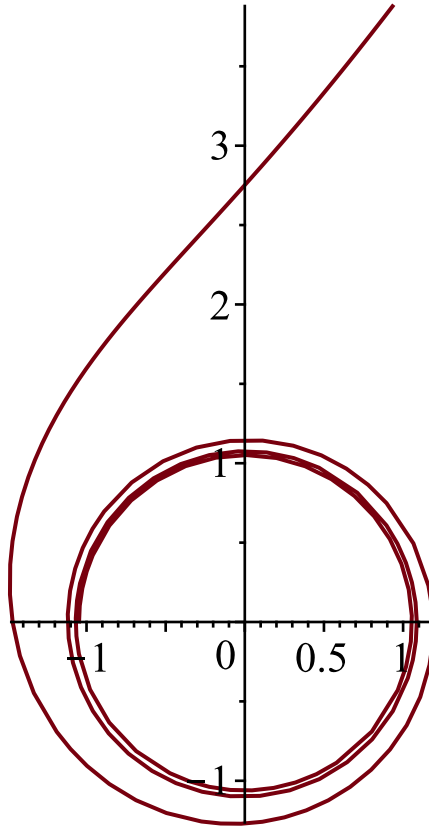
```
> plot([rho, rho/(rho-1), rho=22/21..4], coords=polar, numpoints=100,
       scaling=constrained, size=[320, 320]);
```



Вместо увеличения *numpoints* можно наращивать значение опции *adaptive*. По умолчанию оно равно 6. Однако, следует отдавать отчёт в том, что увеличение этих параметров без крайней необходимости может существенно замедлить время построения изображения.

```
> plot([rho, rho/(rho-1), rho=22/21..4], coords=polar, numpoints=100,
```

```
scaling=constrained,size=[320,320],adaptive=10);
```



Maple позволяет указать начальный набор точек, в которых произойдет вычисление функции, с помощью опции *sample*. При включенной адаптивной схеме расчетов значения в дополнительных точках будут найдены автоматически. Однако, в нетривиальных ситуациях подобрать такой исходный набор крайне сложно.

Условие завершения адаптивной схемы построения графика определяется исходя из значения опции *resolution*. Этот параметр задает разрешения монитора вашего компьютера и по умолчанию принимается равным 800. Можно указать другое значение – чуть больше, однако рассчитывать на существенное улучшение качества не стоит.