13. Графика в Maple V

Maple V release 4 – незаменимый помощник при построении любых типов графиков.

В отличие от предыдущей версии, можно размещать графику сразу же в рабочем документе. Разумеется, при необходимости ее можно построить и в отдельном окне.

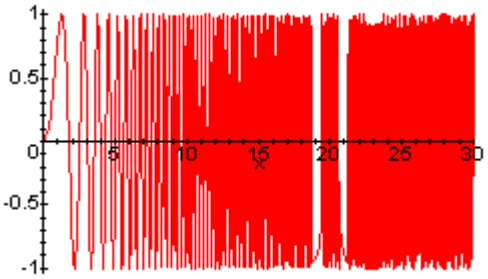
После построения графика параметры изображения можно изменить с помощью специального меню пиктограмм или всплывающего меню, которое вызывается нажатием правой кнопки мыши. Подробно меню рассмотрено при описании интерфейса системы.

С помощью "мыши" можно узнать точные координаты интересующей точки, просто указав на нее и нажав левую кнопку. А при построении трехмерных графиков можно визуально "крутить" оси координат, просматривая сложные участки поверхности с разных точек зрения.

Ограничения

При построении быстроизменяющейся функции может возникнуть эффект, связанный с дискретным шагом построения графика. Хотя Maple использует адаптивный алгоритм, в некоторых случаях шаг построения все-таки оказывается слишком большим, чтобы проанализировать все точки графика. Проиллюстрируем этот факт примером построения сильноколебательной функции:

```
> plot (sin(x^2),x=0..30);
```



Устройства вывода

Установить устройство вывода можно с помощью команды: interface(plotdevice = x), где x - параметр.

Параметр может принимать следующие значения:

- default: восстанавливает установки по умолчанию устройства графического вывода;
- inline: используется для вывода в текущий рабочий документ;
- window: используется для вывода в отдельное окно;
- mac, win: синоним “window”, используется для обеспечения совместимости с предыдущими версиями Maple V;
- ps, postscript: вывод графики в формате encapsulated PostScript в файл, заданный интерфейсной переменной plotoutput;
- gif: GIF-драйвер в данной версии Maple V не поддерживается;
- jpeg: вывод в 24-битовый графический файл в формате JPEG. Можно задать размер изображения, например, plotoptions= ‘height=200, width=320‘. По умолчанию установлена высота в 360 и ширина в 480 пикселей. В 32-битных платформах, таких как Sun SPARC, Windows, Macintosh, установки по умолчанию соответствуют приблизительно максимально возможному размеру изображения.
- pcx: вывод в файл формата PCX (256 цветов). Аналогично драйверу JPEG задаются высота и ширина изображения. По умолчанию установлена высота в 400 и ширина в 640 пикселей;
- hpgl: вывод в формате HP GL. Вывод подходит для распечатки на перьевых HP-плоттерах. Если plotoptions содержит ключевое слово “laserjet”, тогда полученный файл можно будет распечатать на HP Laserjet принтерах.

Кроме вышеупомянутых устройств Maple V release 4 поддерживает следующие устройства: tek, x11, char, vt100, pic, unix, regis, i300, cps, hplj, ln03.

В команде plotsetup устанавливаются опции вывода графики и указывается имя файла для тех драйверов, в которых это необходимо.

Формат plotsetup следующий:

plotsetup(DeviceType, TerminalType, options...)
plotsetup(DeviceType, options...)
plotsetup(options...)

DeviceType – тип графического устройства (например, ps, tek, x11, char)

TerminalType – необязательный параметр, определяющий терминал или устройство, поддерживающее графику. Например, тип tektronix поддерживается под xterm, kermit, ln03 и другими терминалами и принтерами.

options	– каждая опция задается в форме имя=значение, где имя - одно из следующих ключевых слов plotdevice, plotoutput, preplot, postplot, plotoptions.
plotdevice	– определяет используемый графический драйвер.
plotoutput	– определяет имя файла для вывода изображения.
preplot	– список целых чисел (ESC-последовательность), определяющий начальные установки устройства.
postplot	– список целых чисел, определяющий завершение работы устройства.
plotoptions	– строка со значениями, известными для драйвера, разделенными запятыми.

```
> interface(plotdevice=jpeg);
> interface(plotoutput=`frame.jpg`);
> plotsetup( plotoptions=`height=480,
width=640`);
> plot3d(sin(x)*sin(y),x=-Pi..Pi, y=-Pi.. Pi,
style=patch);
```

Теперь можно перейти в просмотрщик файлов формата JPEG и просмотреть красочный график трехмрной поверхности, записанный в файле frame.jpg.

Те же установки можно задать в виде списка в одной команде plotsetup:

```
> plotsetup (plotdevice=jpeg, plotoutput=
`frame.jpg`, plotoptions= `height=480,
width=640`);
```

Для возврата в режим вывода в текущий рабочий документ достаточно ввести следующую команду:

```
> plotsetup(plotdevice=inline);
```

Следует отметить, что сохранить график можно как переменную Maple. Это можно проиллюстрировать следующим примером.

```
> gr1:=plot(sin,3..10):
> gr2:=plot3d(sin(x*y)*cos(x*y),x=-10..10, y=-
10..10):
> save gr1,gr2,`temp`:
```

Считать записанную в файл информацию можно при помощи команды read. И далее воспроизвести при помощи команды eval, причем в качестве аргумента должно быть имя переменной, содержащей информацию о графике.

13.1 Графика 2D

Задание областей

Область - это окно декартовой системы координат, в котором строится график.

Синтаксис определения области:

x=*нижняя граница..верхняя граница*.

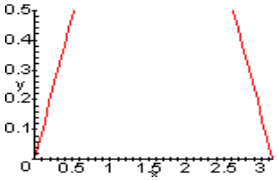
Числа, определяющие границы, должны быть действительными.

plot(f,x=low..hi,y=low..hi) - пример задания.

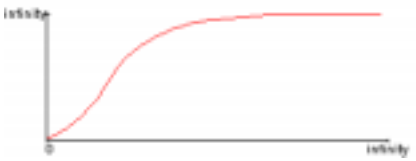
Области можно задавать с использованием констант.

Например: infinity, Pi, exp(8) и т. д. По умолчанию выбирается диапазон -10..10 для оси абсцисс. Если указан один диапазон, то считается, что он для оси абсцисс, а для оси ординат область изменения выбирается автоматически.

```
> plot(sin(x),x=0..Pi,y=0..0.5);
```




```
> plot(exp,0..infinity);
```



Стили

При построении можно выбрать стиль (тип) интерполирования. Задается стиль с помощью ключевого слова *style*:

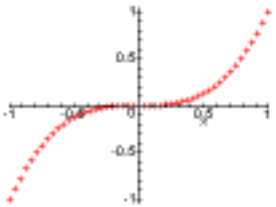
plot(f,h,v,style=x).

Существуют три стиля:

POINT	- построение по точкам;
LINE	- линейная интерполяция;
PATCH	- стиль для многоугольников.

Стиль point - график будет строиться по точкам. Точки могут быть заданы парами в виде списка: [[x1, y1], [x2, y2], ..., [xn, yn]]

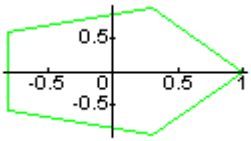
```
> plot(x^3,x= - 1..1,style=point);
```



Стиль line - точки будут соединяться прямыми. Данный стиль выбирается по умолчанию.

Стиль patch - применяется для построения раскрашенных многоугольников.

```
> plot([seq([ cos(2*Pi*i/5), sin(2*Pi*i/5) ], i
= 1..5), [cos(2*Pi/5), sin(2*Pi/5)]],
style=patch, color=green);
```



Параметры

Параметры перечисляются в команде plot после указания областей в форме

< имя параметра > = <значение>.

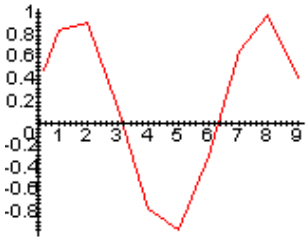
Список параметров приведен в таблице.

Таблица параметров функции plot

ПАРАМЕТРЫ	ЗНАЧЕНИЯ	ОПИСАНИЕ
adaptive	true, false	использование адаптивного алгоритма построения
axes	FRAME, BOXED, NORMAL, NONE	тип координатных осей
axesfont	[family, style, size]	шрифт для осей
color	зарезервированное слово или процедура	цвет графика
coords	имя системы координат	тип системы координат
discont	true, false	для построения выражений с разрывами
font	[family, style, size]	шрифт для текста
labelfont	[family, style, size]	шрифт для меток осей
labels	[str1, str2]	названия осей
linestyle	целое число	тип линии
numpoints	целое число	точки по оси абсцисс
resolution	целое число	горизонтальное разрешение устройства вывода
sample	[x1,x2,...xn]	список точек, в которых будет построена функция (adaptive=false)
scaling	CONSTRAINED, UNCONSTRAINED	масштабирование
style	POINT, LINE, PATCH	тип интерполяции
symbol	BOX, CROSS, CIRCLE, POINT, DIAMOND	символ точек чертежа
thickness	0, 1, 2, 3	толщина линий
title	строка	заголовок чертежа
titlefont	[family, style, size]	шрифт для заголовка
view	[x1..x2, y1..y2]	окно координатной плоскости
xtickmarks, ytickmarks	целое число	количество отметок на осях X и Y

```
> plot(sin, sample=[0.5,1,2,3,4,5,6,7,8,9],
```

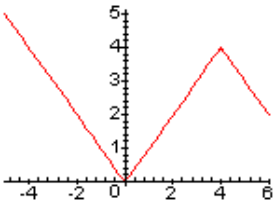
```
adaptive=false);
```



Кусочные функции

Для построения кусочной функции надо просто определить описывающую ее процедуру, а затем как обычно воспользоваться командой plot.

```
> w:=proc(x) if x<0 then  - x elif (x>0) and
(x<4) then x else  - x+8 fi end:
> plot(w, - 5..6,color=red);
```



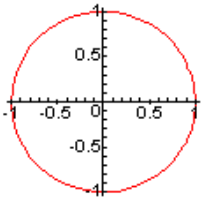
Параметрическая графика

При построении параметрических функций используется следующий синтаксис команды plot:

plot([x(t),y(t),t=(диапазон изменения t)],h,v,options)

```
> plot( [(t^2 - 1)/(t^2+1), 2*t/(t^2+1), t= -
```

```
infinity..infinity] );
```

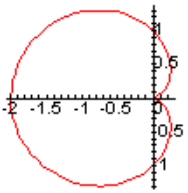


Построение графиков в различных системах координат

При необходимости можно выбрать систему координат, отличную от декартовой. Всего Maple может использовать 15 типов систем координат (для двухмерного построения), которые задаются параметром coords=”имя координат”.

Приведем пример построения в полярных координатах:

```
> plot ([1 - cos(t), t, t=0..2*Pi],
coords=polar);
```



Анимация 2D графиков

В Maple возможна анимация двухмерных графиков. Причем, если в предыдущей версии это можно было сделать только в отдельном окне, то теперь анимация работает в самом документе и при этом пользователь может продолжать свою работу!

```
> with(plots):
> animate( {x - x^3/u,sin(u*x)}, x=0..Pi/
2,u=1..16, color= red);
```

Синтаксис команды animate:

animate(F, x, t,...).

Здесь F=F(x,t) - функция двух переменных; x, t - диапазоны изменения x и t.

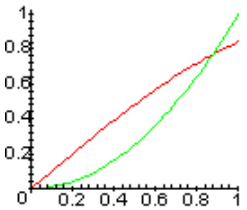
При анимации происходит следующее: изменяются значения t и при фиксированных значениях t строится график F(x,t). Количество выводимых кадров можно устанавливать параметром frames (по умолчанию frames =16).

Совмещение графиков

Можно построить несколько графиков на одной координатной плоскости. Для этого достаточно указать в команде plot множество или список функций, т.е. записать через запятую функции и заключить их в фигурные или квадратные скобки. В этом случае Maple автоматически выбирает разные цвета для графиков.

Допустимо совмещать обычную и параметрическую графику.

```
> plot({x,[x^2,x,x=0..1]},x=0..1);
```



При необходимости для каждой функции можно указать конкретный цвет и стиль построения.

```
> plot([cos(x),cos(x+0.1)], x=0..2,
color=[red,blue],   style=[point,line],
symbol=diamond);
```