

- `title`. Заголовок графика, пишется сверху. Для переноса на новую строку можно использовать командный знак переноса `\n`, принятый в формате вывода. Допустимы русские буквы. Однако при сохранении рисунка в виде графического файла eps (Encapsulated PostScript) русские буквы не воспринимаются.
- `titlefont`. Шрифт заголовка, см. `axesfont`, с. 259.
- `view=[xmin..xmax, ymin..ymax]`. Границы графика. Здесь `xmin` и `xmax` — соответственно нижняя и верхняя границы оси x . Аналогично `ymin` и `ymax` границы оси y . Таким образом, можно обрезать график до нужных размеров.

5.5.3. Пакет ImageTools

Для работы с изображениями в Maple имеется специальный пакет `ImageTools`. Возможности этого пакета не столь велики, как, например, у аналогичных пакетов MATLAB, однако для альтернативного способа построения графиков, импорта файлов изображений и их обработки в нем есть достаточно операторов. Есть также и команды работы со звуком (`Clip`). Приведем программу построения графика функции $y = 61 + 60 \sin(\pi x/60)$ (рис. 329).

```
> with(ImageTools):
> f:=(r,c) -> if r=round(-sin(c*3.14/60)*60+61)
>               then 1 else 0 fi:
> img := Create(121,120,1,f):
> View(1-img);
```

Здесь оператор `Create(121,120,1,f)` создает изображение, фактически массив из 121 строк и 120 столбцов вещественных чисел от 0 до 1. Белый цвет соответствует 1, черный — 0. Функция создается процедурой `f` с аргументами `r` и `c` (`row` — ряд, `column` — столбец). Аргументы этой функции пробегают свои области определения `1..121` и `1..120`. Если номер ряда совпадает с округленным значением функции $y(x)$, то в это место выставляется белая точка, иначе — черная.

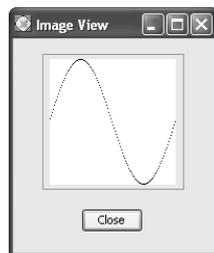


Рис. 329

Потом график строится в негативе `View(1-img)`, хотя можно было бы оставить его и в такой форме, как мелом на черной доске. Результат

выводится в отдельное окно, которое потом требуется закрыть, для того, чтобы программа работала дальше.

Другой способ построения изображения — непосредственное присвоение значений массиву `img`. Но сначала надо его создать (пока пустой) оператором `img := Create(121,120)`. Затем в цикле легко получить изображение:

```
> for i to 120 do
>   img[round(-sin(i*2*Pi/120)*60+61),i]:=1:
> od:
```

Таким образом пакет `ImageTools` позволяет получить доступ к любому пикселю изображения. Это можно использовать, например, при обработке готовых графических файлов. В следующей короткой программе для выделения контуров на фотографии применяется фильтр Собеля¹. Этот алгоритм используется во многих графических редакторах, например Adobe Photoshop².

Для работы фильтра создается матрица $G_1 = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$,

которая собирает восемь пикселей вокруг пикселя с номером ячейки i, j в сумму $X = \sum_{k=1}^3 \sum_{m=1}^3 G_{1,km} A_{km}$, а транспонированная матрица

$G_2 = G_1^T$ собирает другую сумму $Y = \sum_{k=1}^3 \sum_{m=1}^3 G_{2,km} A_{km}$, где A —

подматрица размером 3×3 , вырезанная из матрицы изображения вокруг элемента i, j , т.е. $A_{2,2} = img_{i,j}$. Затем на место i, j изображения ставится новое значение $\sqrt{X^2 + Y^2}$. Так из исходного изображения на рис. 330 выделяется его контурный вариант, рис. 331.



Рис. 330



Рис. 331

¹*Sobel Irwin* — американский ученый, сотрудник Hewlett-Packard Laboratories.

²В меню Adobe Photoshop: Фильтры->Стилизация->Свечение краев или Filter->Stylize->Trace Contour

Оператор `Read` считывает файл изображения (фотография главного корпуса МЭИ, Красноказарменная, 17), а оператор `RGBtoGray` преобразует его в черно-белый вариант¹. Размер изображения определяется операторами `Width` и `Height`. Для того, чтобы воспользоваться оператором выделения подматрицы нужных размеров из матрицы изображений, необходимо преобразовать изображение, которое по форме только похоже на матрицу, в настоящую матрицу `IMG:=convert(im,Matrix)`. Переменная `IMG` теперь имеет тип матрицы и оператор `SubMatrix` может с ней работать. Фильтр Собеля в этой программе немного обобщен. Введен параметр z , который принимает значение 2 для классического фильтра Собеля. С увеличением z растет контрастность изображения. Для аналогичного фильтра Щарра² имеем `s:=[3,10,3]`.

Результат записывается в файл оператором `Write` и сообщается размер записанного файла, в данном примере — 10929. Отметим, что упомянутые здесь операторы чтения и записи принадлежат пакету `ImageTools`. В других пакетах (`AudioTools`, `FileTools`, `Sockets`) есть одноименные операторы, но они работают иначе.

Программа 1

```
> restart;with(ImageTools):# Фильтр Собеля
> with(LinearAlgebra):
> img0:= RGBtoGray(Read("C:\\МЭИ.jpg")):
> Фильтр:=proc(z,im)
> local IMG,s,G1,G2,i,j,A,X,Y,w,h:
> global img;
> w:=Width(im): h:=Height(im):
> img:=Create(h,w):
> IMG:=convert(im,Matrix):
> s:=[z/2,z,z/2]:
> G1:=Matrix([s,[0,0,0],[-s]]);
> G2:=Transpose(G1):
> for i from 2 to h-1 do
>   for j from 2 to w-1 do
>     A:=SubMatrix(IMG,[i-1,i,i+1],[j-1,j,j+1]):
>     X:=add(add(A[k,m]*G1[k,m],k=1..3),m=1..3);
```

¹ Другой способ преобразования в черно-белое изображение: отделение одного из слоев `GetLayer(Read("C:\\МЭИ.jpg"),1)`.

² В. Jahne, H. Scharf, and S. Korkel. *Principles of filter design*. In *Handbook of Computer Vision and Applications*. Academic Press, 1999.

```

> Y:=add(add(A[k,m]*G2[k,m],k=1..3),m=1..3);
> img[i,j]:=sqrt(X^2+Y^2):
> od:
> od:
> img:
> end proc:
> Фильтр(1,img0):img1:=1-img: View(img1):
> Write("C:\\MPEI2.jpg",img1);

```

10929

5.5.4. Пакет plots

Как всегда, пакет вызывается командой `with(plots)`. Если после команды ставить точку с запятой, то на экране появятся все операторы этого пакета. В Maple 13 их 57. Перечислим некоторые.

- `animate`. Анимация графиков. Например, можно показать процесс рисования эллипса, имитируя движение карандаша по бумаге.

```

> animate( plot, [[4*cos(t), sin(t), t=0..T]],
>          T=0..-2*Pi, scaling=constrained, frames=10);

```

После того, как сработает этот оператор, на экране появятся пустые оси координат. Далее надо в верхней панели инструментов и команд Maple найти магнитофонную панель (рис. 400, с. 347) и запустить анимацию клавишей «Пуск» (Play the animation).

- `animatecurve`. Анимация процесса построения кривой. По сути совпадает с `animate`, но имеет несколько другую форму. Так, например, предыдущий пример можно реализовать следующим образом:

```

> animatecurve([4*cos(t), -sin(t), t=0..2*Pi],
>              scaling=constrained, frames=10);

```

Для того, чтобы кривая вычерчивалась по часовой стрелке, знак параметра t в уравнении эллипса был заменен на обратный. Верхний предел для t должен быть положительным.

- `animate3d`. Анимация трехмерных графиков. В отличие от `animate` здесь поверхности не рисуются, а меняются. Последний параметр t отвечает за изменение формы, два первых x и y — аргументы функции:

```

> animate3d(t^2*cos(2*x)+sin(y*t),
>           x=-2..2, y=0..2, t=1..3, frames=50);

```