4. Построение двумерных графиков

4.1. Функция plot

Для построения графиков вида `y=f(x)` в `Scilab` существует функция `plot`.. Синтаксис этой функции сильно изменился в четвертой версии пакета, поэтому сначала рассмотрим синтаксис предыдущих версий `Scilab`, а затем посмотрим что изменилось в четвертой версии.

В предыдущих версиях `Scilab` обращение к функции `plot` предназначена для построения графика функции `y=f(x)` (`Scilab 3`). Обращение к функции имеет вид:

```
plot(x,y,[xcap,ycap,caption])
```

Здесь `x` – массив абсцисс, `y` – массив ординат, `xcap, ycap, caption` – подписи осей `X`, `Y` и графика соответственно. Рассмотрим ее использование на примере построения функции `y=sin(cos(x))`.

```
x=-2*%pi:0.1:2*%pi;
y=sin(cos(x));
plot(x,y,'X','Y','plot function y=sin(cos(x))');
```

Листинг 4.1.

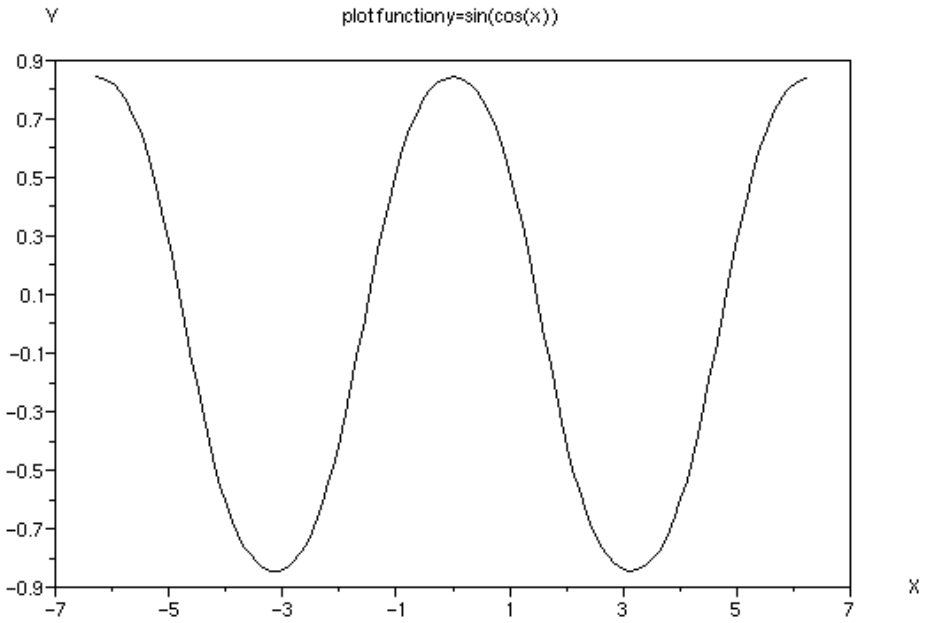

Рис. 4.1. График функции y=sin(cos(x))

В простейшем случае обращение к функции имеет вид `plot(y)` , в качестве массива `x` выступает массив номеров точек массива `y`. В этом случае с помощью функции можно построить графики нескольких функций (см. листинг 4.2 и рис. 4.2).

```
x=-2*%pi:0.1:2*%pi;
plot([sin(cos(x));cos(sin(x));exp(sin(x));exp(cos(x))]);
```

Листинг 4.2.

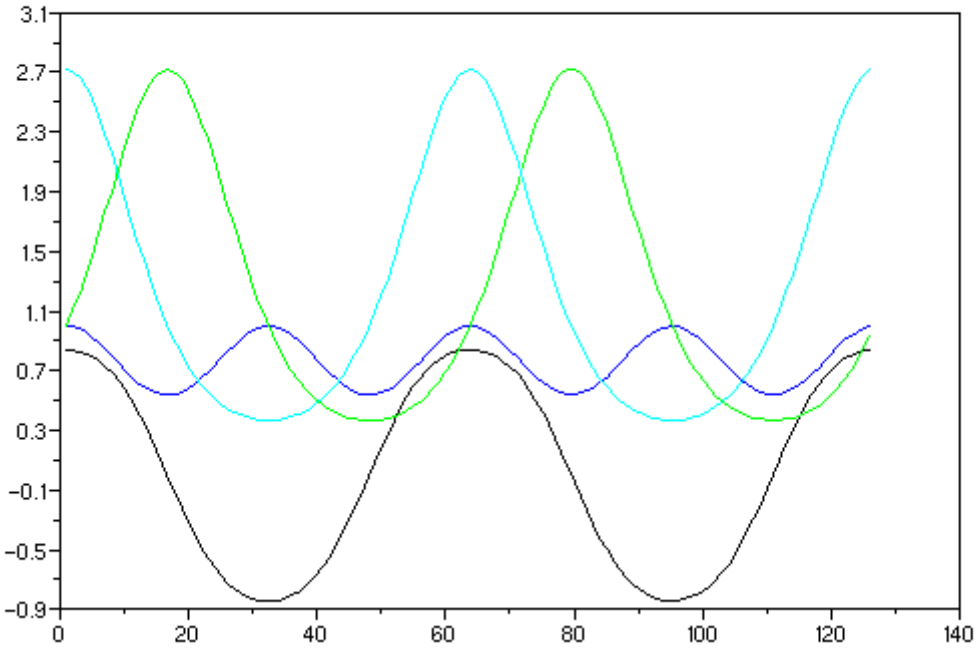

Рис. 4.2.Графики функций $y=sin(cos(x))$, $z=cos(sin(x))$, $u=e^{sin(x)}$, $v=e^{cos(x)}$

Как видно, из рис. 4.2, в Scilab 3 функция plot не позволяет построить полноценные графики нескольких функций.

Поэтому в Scilab 4.0 функция plot значительно модифицирована и ее возможности теперь значительно расширены и сопоставимы с возможностями функции plot из Matlab 7. При простейшем обращении к функции plot(x,y) будет создано окно с именем Scilab Graphic (0), в котором будет построен график функции y(x) на интервале (см. на рис. 6.3 график функции y=sin(cos(x))).

Если повторно обратиться к функции plot, то будет создано новое графическое окно и в нем будет построен новый график. Для построения нескольких графиков в одной системе координат можно поступить одним из следующих способов:

1. Обратиться к функции plot следующим образом plot(x1,y1,x2,y2,… xn,yn), где x1, y1 – массивы абсцисс и ординат первого графика; x2, y2 – массивы абсцисс и ординат второго графика; …, xn, yn – массивы абсцисс и ординат n-ого графика. Например

```
x=-6.28:0.02:6.28;
y=sin(x/2);
z=cos(x);
v=exp(cos(x));
plot(x,y,x,z,x,v);
```

Листинг 4.4.

Полученный график представлен на рис. 4.4.

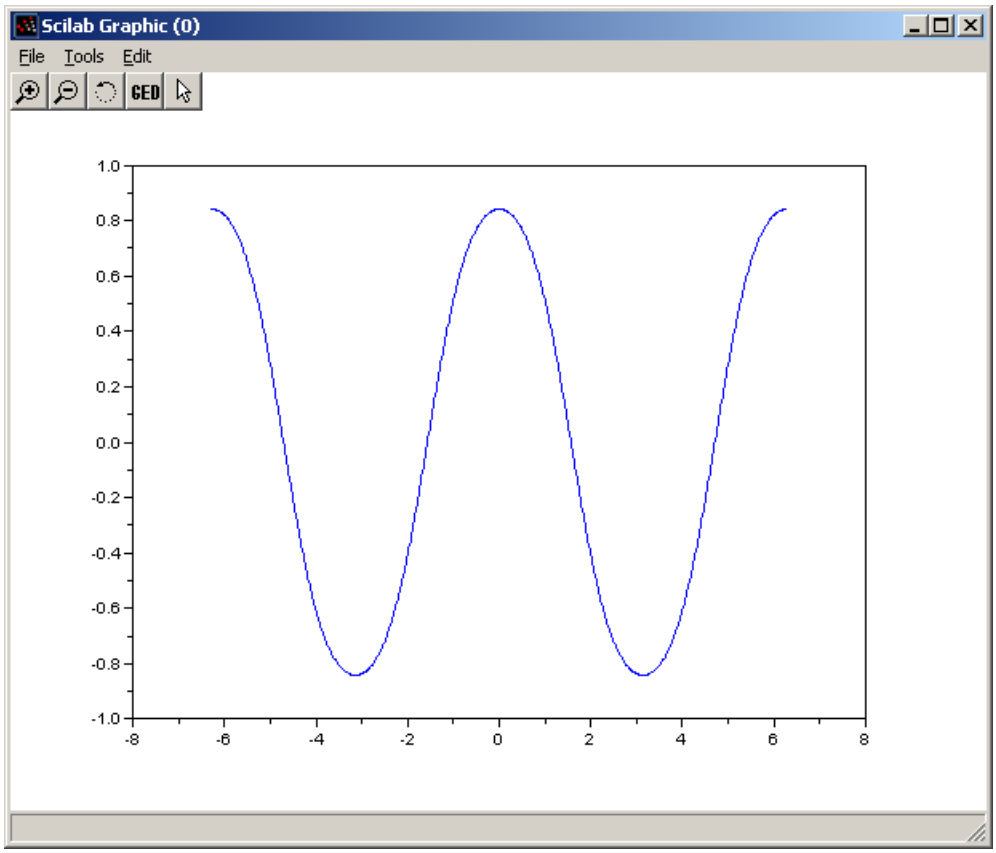

Рис. 4.3: График функции y=sin(cos(x))

2. Каждый график изображать с помощью функции `plot(x,y)`, но перед обращением к функциям `plot(x2,y2)`, `plot(x3,y3)`, …, `plot(xn,yn)` вызвать команду `mtlb_hold(' on')` , которая блокирует режим создания нового окна.

```
x=-6.28:0.02:6.28;
y=sin(x/2);
plot(x,y);
mtlb_hold(' on');
z=cos(x);
plot(x,z);
mtlb_hold(' on');
v=exp(cos(x));
plot(x,v);
```

Листинг 4.5

Обратите внимание, что при построении графиков первым способом Scilab автоматически изменяет цвета изображаемых в одной системе координат графиков. Однако управлять цветом и видом каждого из изображаемых графиков может и пользователь, для чего необходимо воспользоваться полной формой функции `plot`:

```
plot(x1, y1, s1, x2, y2, s2, …, xn, yn, sn)
```

где `x1, x2, …, xn` – массивы абсцисс графиков; `y1, y2, …, yn` – массивы ординат графиков; `s1, s2, …, sn` – строка, состоящая из трех символов, которые определяют цвет линии, тип маркера и тип линии графиков (см. табл. 4.1-4.3), в строке могут использоваться один или два символа.

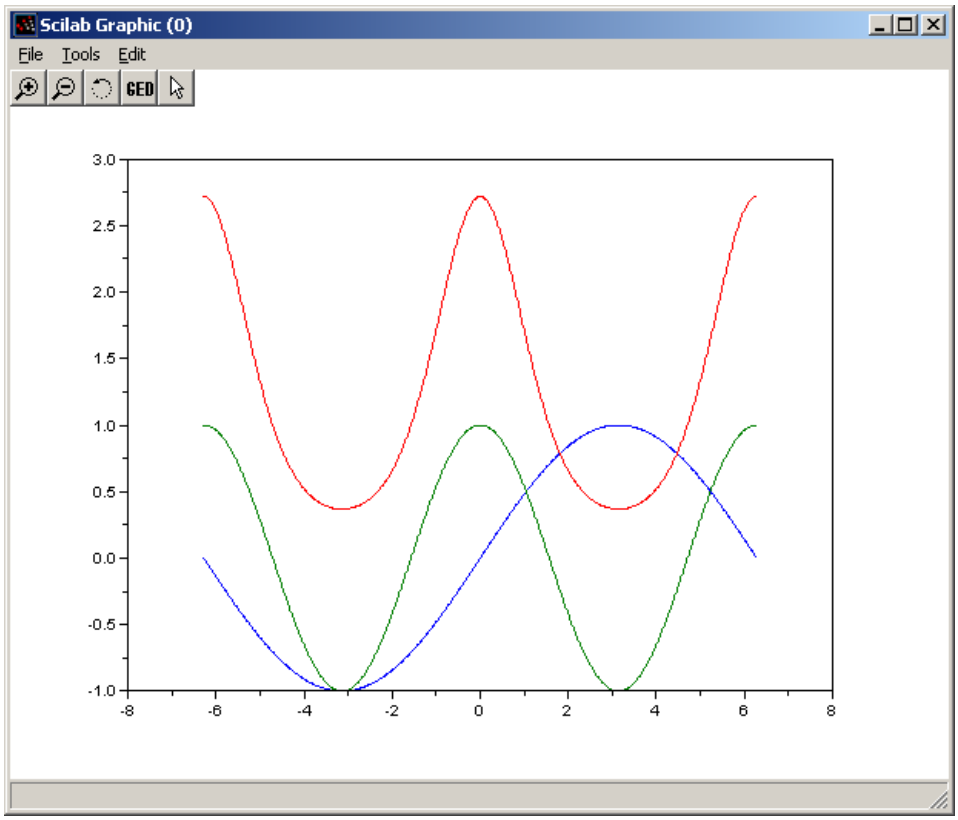

Рис. 4.4: График функции y=sin(x/2), z=cos(x);v=exp(cos(x)) в Scilab 4

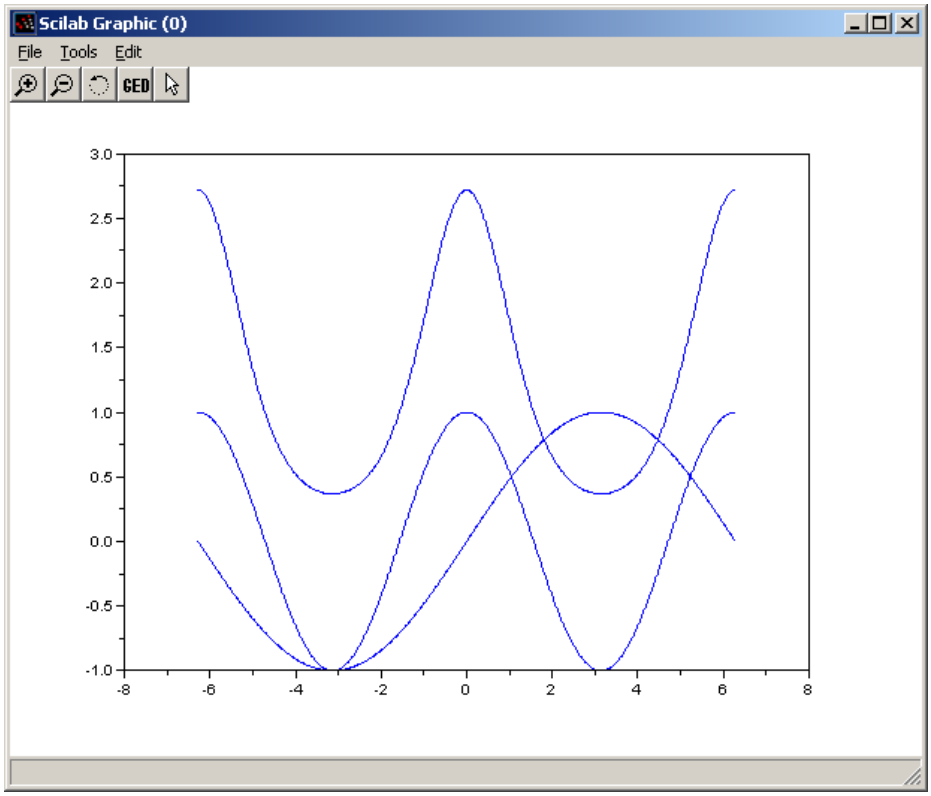

Рис. 4.5: Построение графиков функций y=sin(x/2), z=cos(x);v=exp(cos(x)) в Scilab 4 вторым способом

Таблица 4.1. Символы цветов линий

Символ	Описание
y	желтый
m	розовый
c	голубой
r	красный
g	зеленый
b	синий
w	белый
k	черный

Таблица 4.2. Символы типов маркера

Символ	Описание
.	точка
o	кружок
x	крестик
+	знак "плюс"
*	звездочка
s	квадрат
d	ромб
v	треугольник вершиной вниз
^	треугольник вершиной вверх
<	треугольник вершиной влево
>	треугольник вершиной вправо
p	пятиконечная звезда
h	шестиконечная звезда

Таблица 4.3. Символы типов линий

Символ	Описание
-	сплошная
:	пунктирная
-.	штрихпунктирная
--	штриховая

В качестве примера рассмотри построение графиков функций `y=sin(x/2)`, `z=cos(x)`, `v=exp(cos(2x))` на интервале с использованием управлением свойствами линий, в результате чего формируется окно, представленного на рис. 4.6.

```
x=-6.28:0.02:6.28;
y=sin(x/2);
plot(x,y,'r-');
mtlb_hold('on');
```

```
z=cos(x);
plot(x,z,'k:');
mtlb_hold('on');
v=exp(cos(x));
plot(x,v,'b-');
```

Листинг 4.6

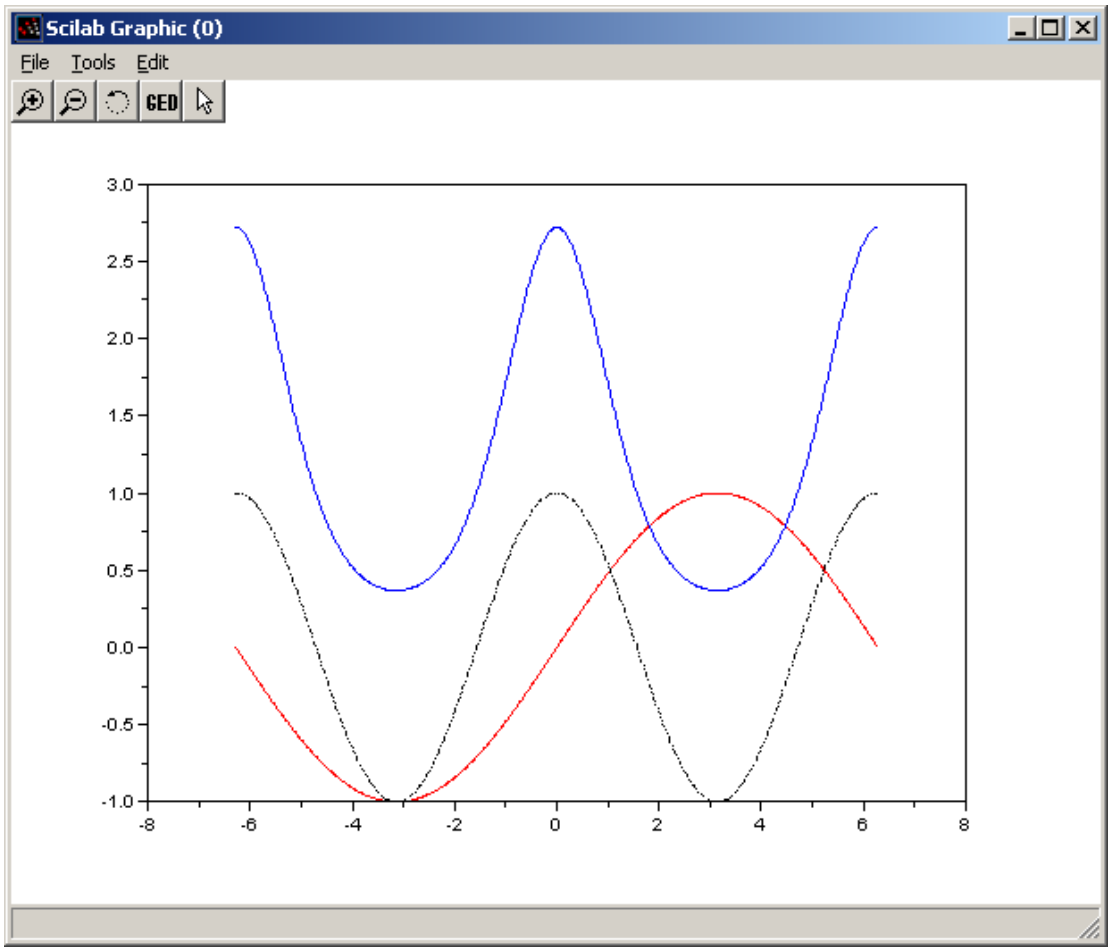

Рис. 4.6: Построение графиков в Scilab с указанием свойств линий

Следующей функцией, которая может быть использована для построения графиков, является функция `plot2d`.

4.2. Функция plot2d

В общем виде обращение к функции имеет вид:

```
plot2d([loglog],x,y,[key1=value1,key2=value2, ..., keyn=valuen]
```

- `logflag` – строка из двух символов, каждый из которых определяет тип осей (`n` - нормальная ось, `l` – логарифмическая ось), по умолчанию `"nn"`;
- `x` – массив абсцисс;
- `y` – массив ординат(или матрица, каждый столбец которого содержит массив ординат очередного графика) (количество элементов в массиве `x` и `y` должно быть одинаковым), если `x` и `y` – являются матрицами одного размера, то в этом случае, каждый столбец матрицы `y` отображается относительно соответствующего столбца

матрицы **x**;

- `keyi=valuei` – последовательность значений параметров графиков, возможны следующие значения параметров: `style` – определяет массив (`mas`) числовых значений цветов графика (`id` цвета), количество элементов массива совпадает с количеством изображаемых графиков, по умолчанию, по умолчанию представляет собой массив mas_i=i, цвет i-й линии совпадает с номером i, для формирования `id` соответствующего цвета (`кода цвета`) можно воспользоваться функцией `color`, которая по названию (`color("цвет")`) или коду `grb` (`color(r,g,b)`) цвета формирует нужный `id` (код) цвета. Если значение стиля отрицательное то это будет точечный график без соединения точек между собой линиями. Пример построения нескольких графиков различного цвета приведен ниже (см. листинг 4.7 и рис.4.7).

```
x=[-2*%pi:0.1:2*%pi];
y=[sin(x); cos(x)];
plot2d(x,y',style=[color("red"), color("blue")]);
```

Листинг 4.7.

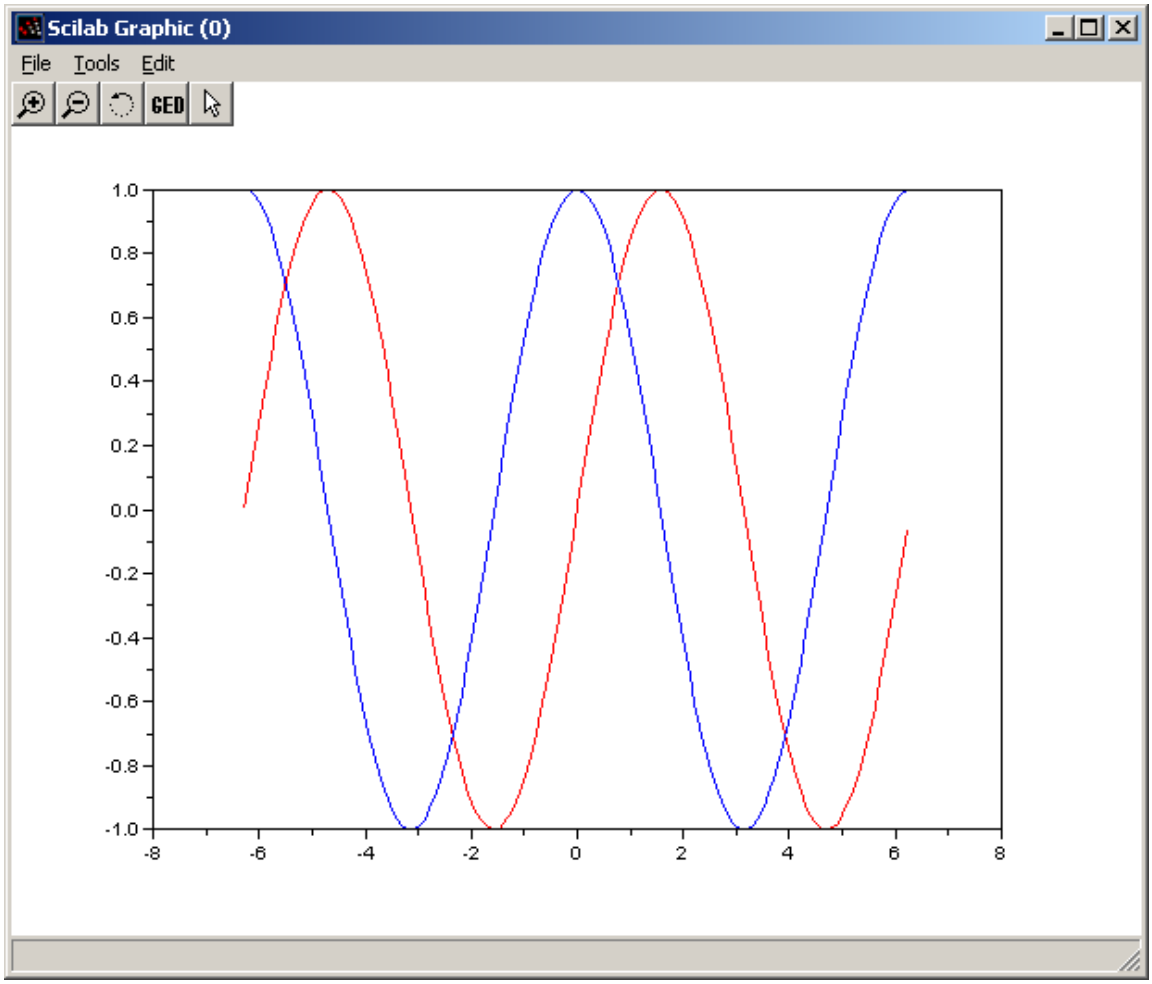

Рис. 4.7. Использование параметра style в функции plot2d

- `rect` – этот вектор `[xmin, ymin, xmax, ymax]` определяет размер окна вокруг графика, пример использования этого параметра приведен на листинге 4.8 и

рис. 4.8;

```
x=[-2*%pi:0.1:2*%pi];
y=[sin(x); cos(x)];
plot2d(x,y',style=[color("red"),color("blue")],rect=[-8,-2,8,2]);
```

Листинг 4.8.

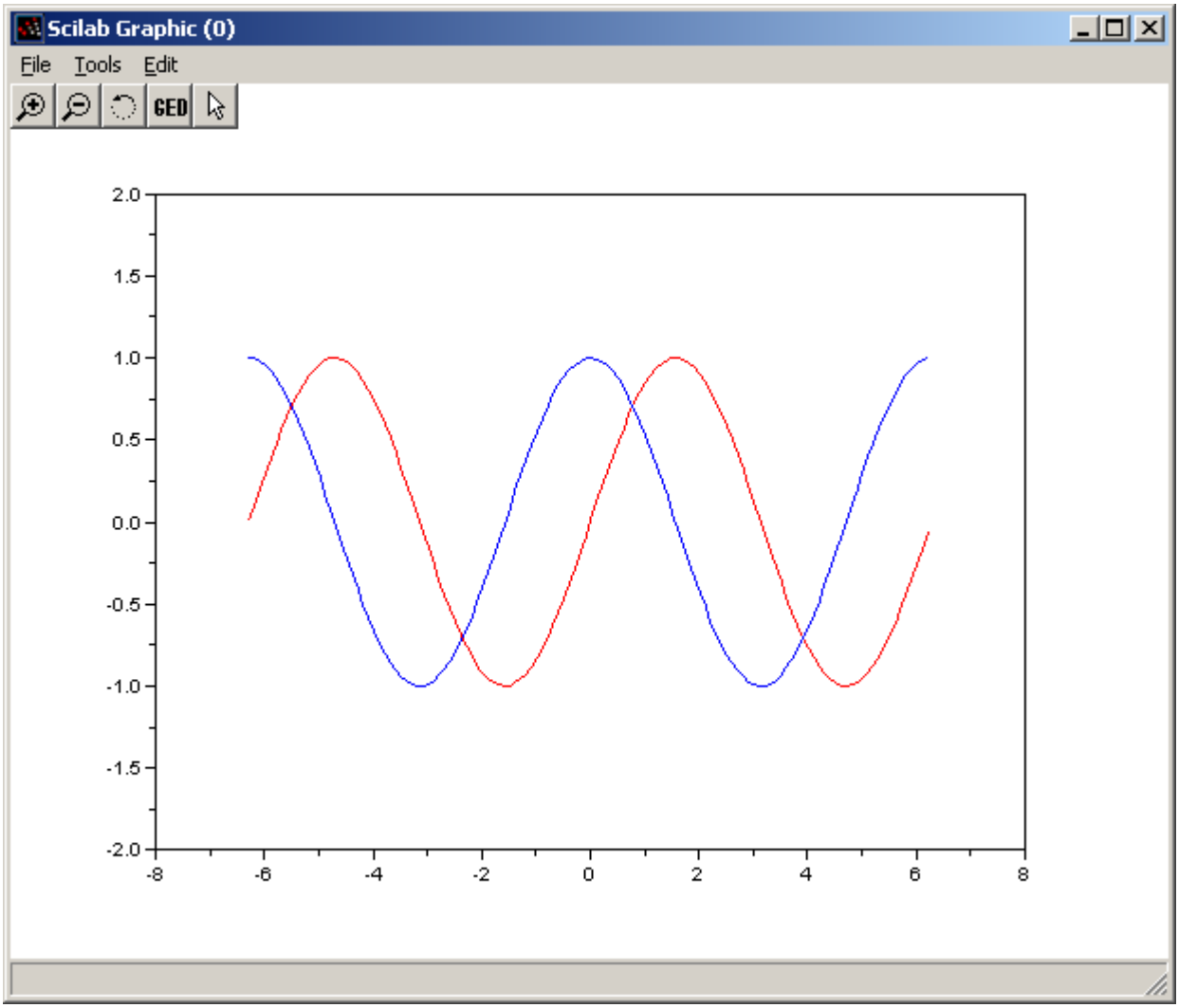

Рис. 4.8. Использование параметра rect в функции plot2d

- `frameflag` – параметр определяет окно в котором, будет изображаться график, он может принимать следующие значения: 0 – не вычислять размеры окна, использовать значения по умолчанию или значения из предыдущего графика, 1 – размер окна определяется параметром `rect`, 2 – размер окна определяется из соотношения между минимальным или максимальным значениями `x` и `y`, 3 – размер окна определяется параметром `rect` в в изометрическом масштабе, 4– размер окна определяется из соотношения между минимальным или максимальным значениями **x** и **y** в изометрическом масштабе,
- `axesflag` - параметр, который определяет рамку вокруг графика, следует выделить следующие значения этого параметра: 0 – нет рамки вокруг графика (см. листинг 4.9 и рис. 4.9); 1 – изображение рамки, ось **y** слева (см. рис. 4.10); 3 –

изображение рамки, ось **y** справа (см. рис. 4.11); 5 – изображение осей проходящих через точку (0,0) (см. рис. 4.12);

```
x=[-2*%pi:0.1:2*%pi];
y=[sin(x); cos(x)];
plot2d(x,y',style=[color("red"), color("blue")], axesflag=0);
```

Листинг 4.9.

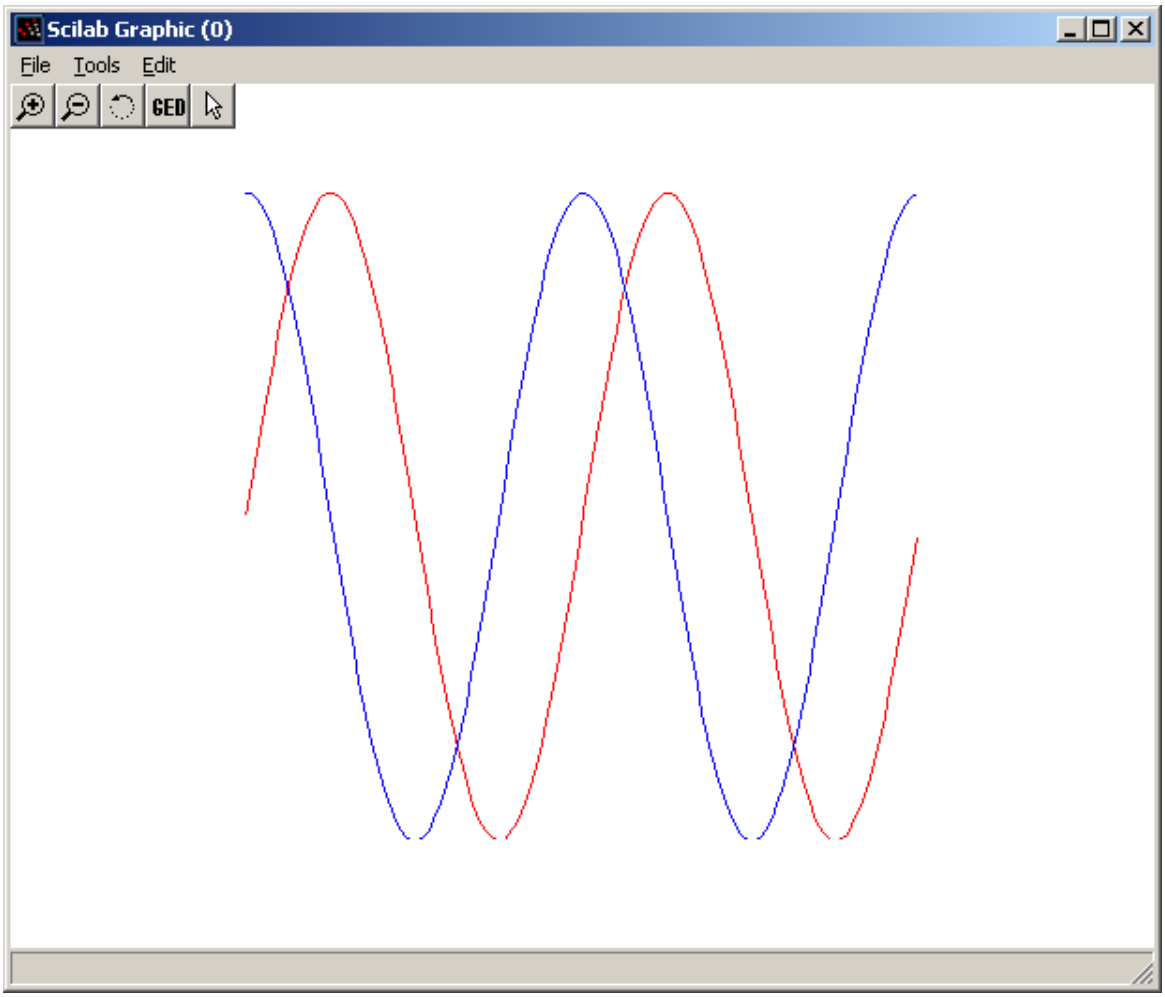

Рис. 4.9. Параметр axesflag=0 в функции plot2d

- `nax` – этот параметр используют, если параметр `axesflag` равен 1, nax представляет массив из четырех значений `[nx, Nx, ny, Ny]` – где `Nx` (`Ny`) – число основных делений с подписями под осью `X` (`Y`), `nx` (`ny`) – число промежуточных делений;
- `leg` – строка, определяющая легенды для каждого графика, структура строки такая: `"leg1@leg2@leg3@...@legn"`, где `leg1` – легенда первого графика, ..., `legn` – легенда первого графика.

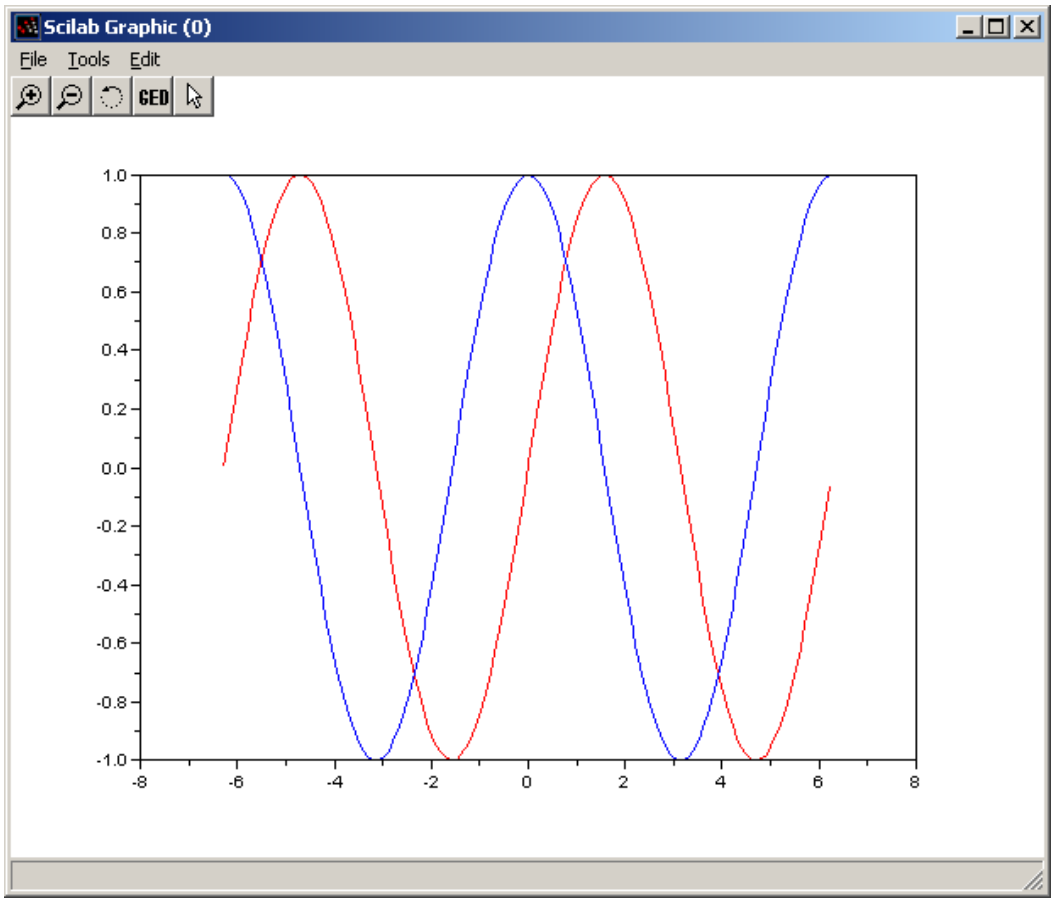

Рис. 4.10. Параметр axesflag=1 в функции plot2d

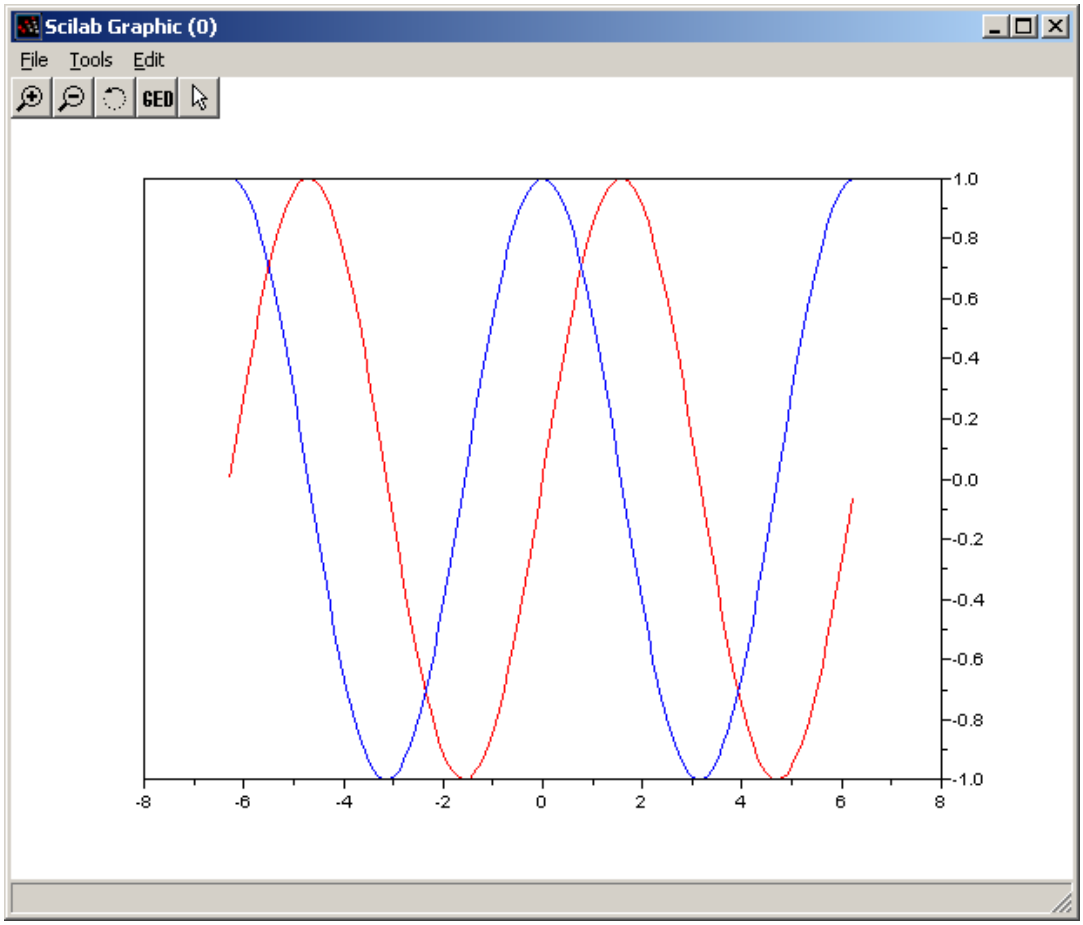

Рис. 4.11. Параметр axesflag=3 в функции plot2d

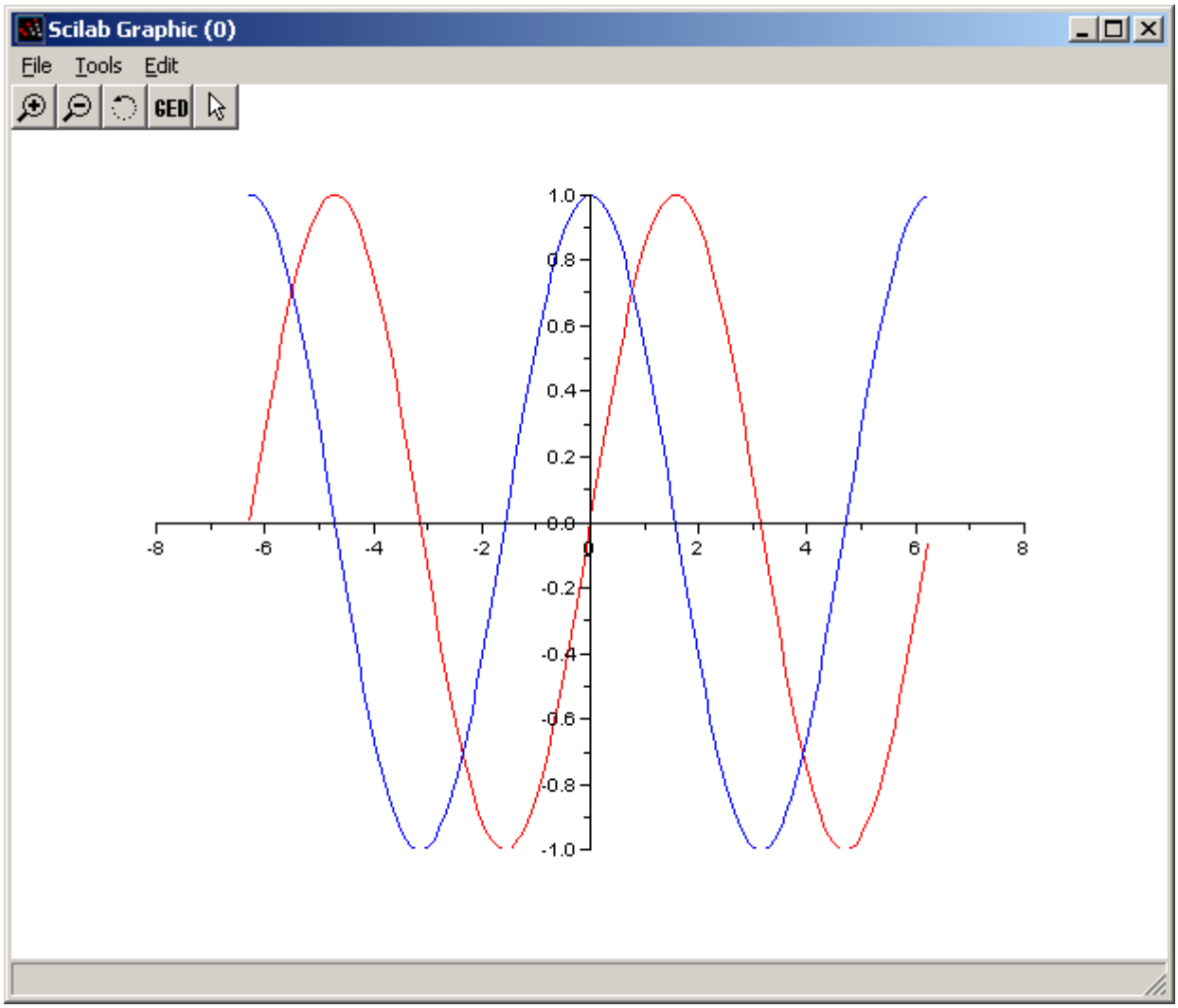

Рис. 4.12. Параметр axesflag=5 в функции plot2d

На листинге 4.8 и рис. 4.13 приведен пример построения графиков функций с использованием параметра `nax` при построении функции `plot2d`.

```
  x=[-8:0.1:8];
  y=[sin(x); cos(x)];
  plot2d(x,y',style=[color("red"),color("blue")],axesflag=1,
nax=[4,9,3,6]);
```

Листинг 4.9

На листинге 4.9 приведен пример построения графиков функции с использование легенд.

```
  x=[-2*%pi:0.1:2*%pi];
  y=[sin(x); cos(x)];
  plot2d(x,y',style=[color("red"),    color("blue")],    axesflag=5,
leg="sin(x)@cos(x)");
```

Листинг 4.10.

Функцию `plot2d` можно использовать для построения точечных графиков. В этом случае обращение к функции имеет вид

```
plot2d(x,y,d),
```

здесь d – отрицательное число, определяющее тип маркера (см. листинг 4.10 и рис. 4.14).

```
x=[-2*%pi:0.25:2*%pi];
y=sin(x);
plot2d(x,y,-3);
```

Листинг 4.10

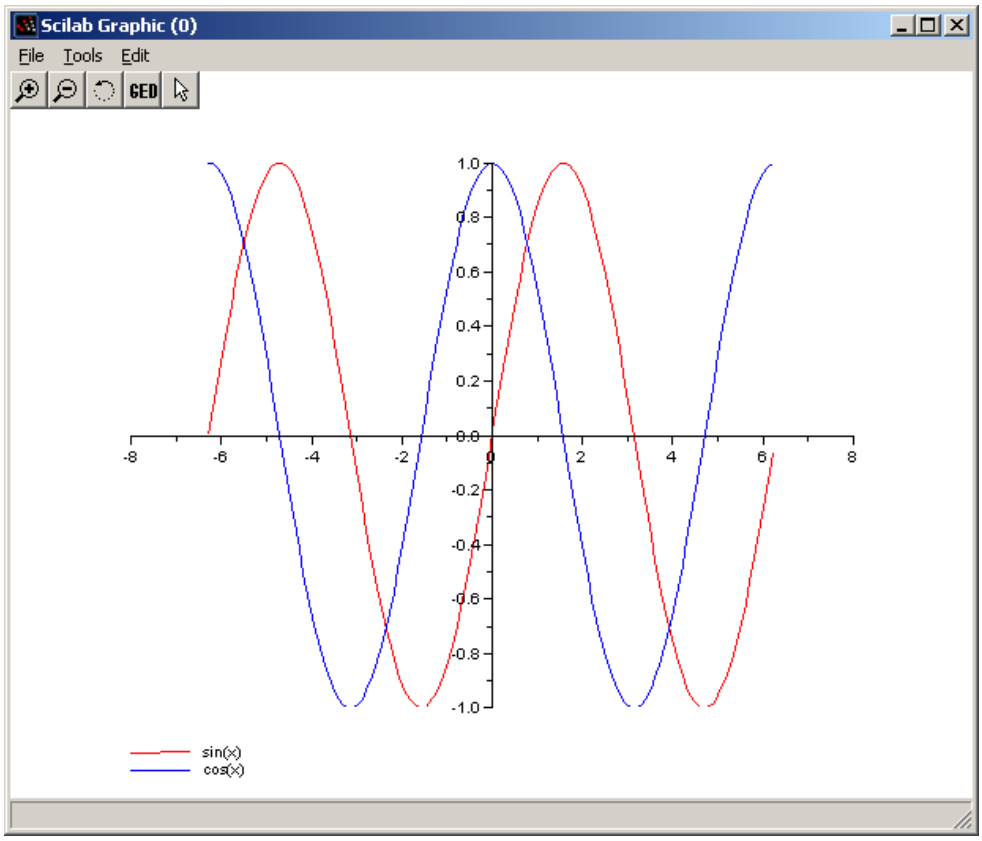

Рис. 4.13. Использование параметра leg в функции plot2d

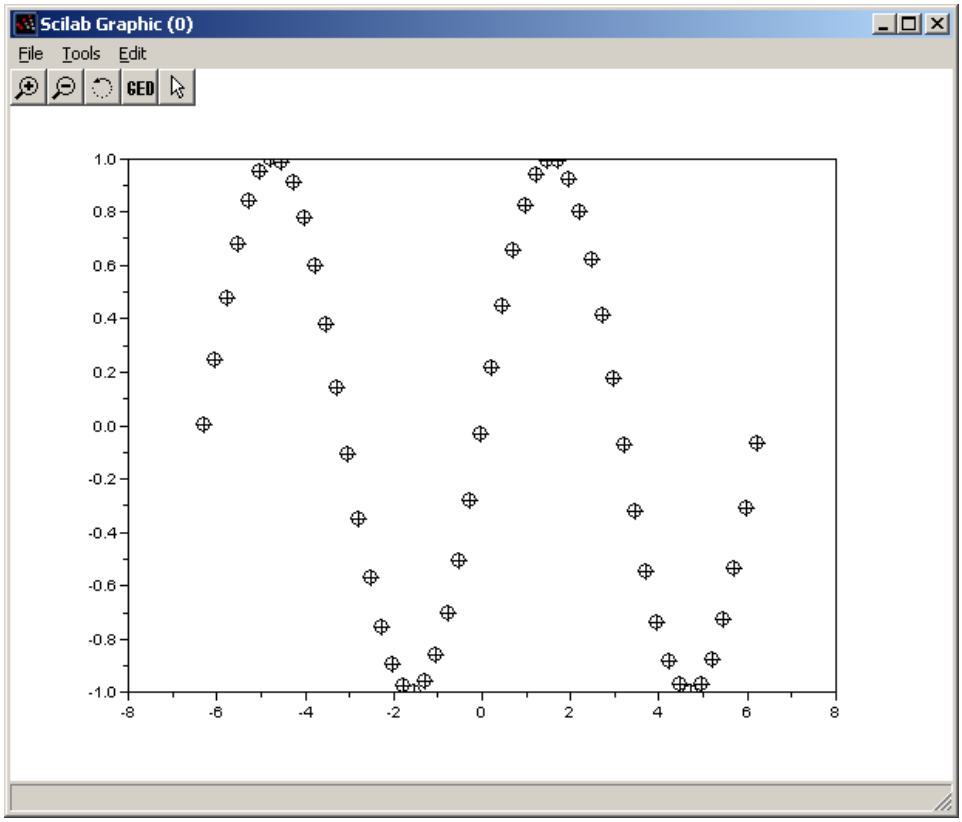

Рис. 4.14.Построение точечного графика

Для изображения графика в виде ступенчатой линий в Scilab есть функция `plot2d2(x,y)`.

Пример ее использования приведен на листинге 4.11 и на рис. 4.15.

```
x=[1911,1941,1961,1981,1991,1996];
y=[20,300,350,1100,1030,1020];
plot2d2(x,y);
```

Листинг 4.11.

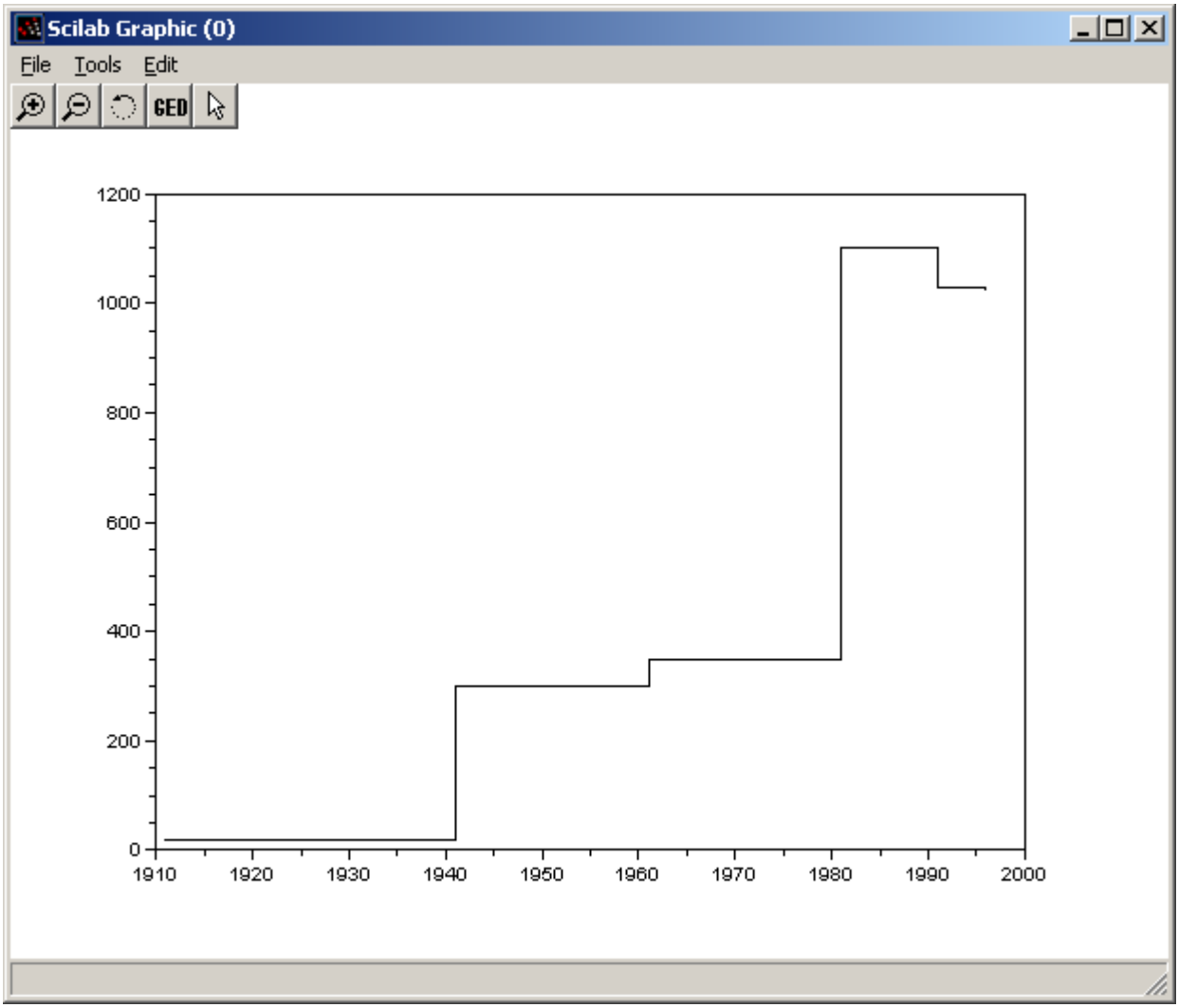

Рис. 4.15. Пример изображения графика в виде ступенчатой линии

4.3. Построение полярных графиков

Для построения графиков в полярной системе координат в `Scilab` служит функция `polarplot`

```
polarplot(fi,ro,[key1=value1,key2=value2, ..., keyn=valuen])
```

Здесь `fi` - полярный угол, `ro` – полярный радиус.

Рассмотрим пример построения полярных графиков $\rho = 3\cos(5\varphi), \rho_1 = 3\cos(3\varphi)$ (см. листинг 4.12, рис. 4.15).

```
fi=0:0.01:2*%pi;
```

```
ro=3*cos(5*fi);
ro1=3*cos(3*fi);
polarplot(fi,ro,style=color("red"));
polarplot(fi,ro1,style=color("blue"));
```

Листинг 4.12.

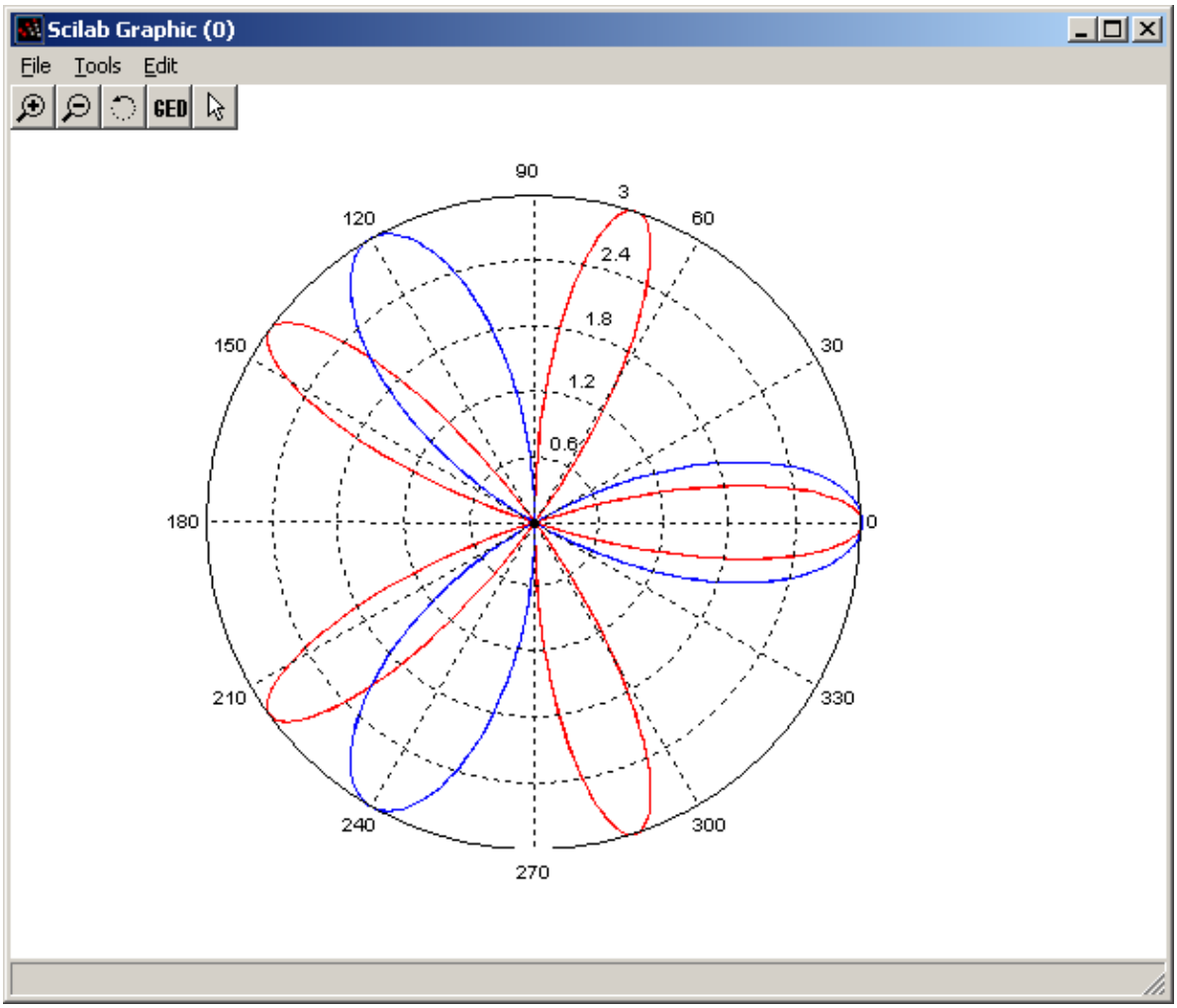

Рис. 4.16. Пример построения полярных графиков

4.4. Построение графиков в параметрической форме

Для построения графиков в параметрической форме можно воспользоваться функциями **plot2d** или **plot**. В качестве примера параметрического графика рассмотрим построение графика строфоиды $x(t)=\frac{t^2-1}{t^2+1}$, $y(t)=t\frac{t^2-1}{t^2+1}$, $t=-5,\ldots,5$ (см. листинг 4.13, рис. 4.17)

```
t=-5:0.01:5;
x=(t.^2-1)./(t.^2+1);
y=t.*(t.^2-1)./(t.^2+1);
plot(x,y);
```

Листинг 4.13

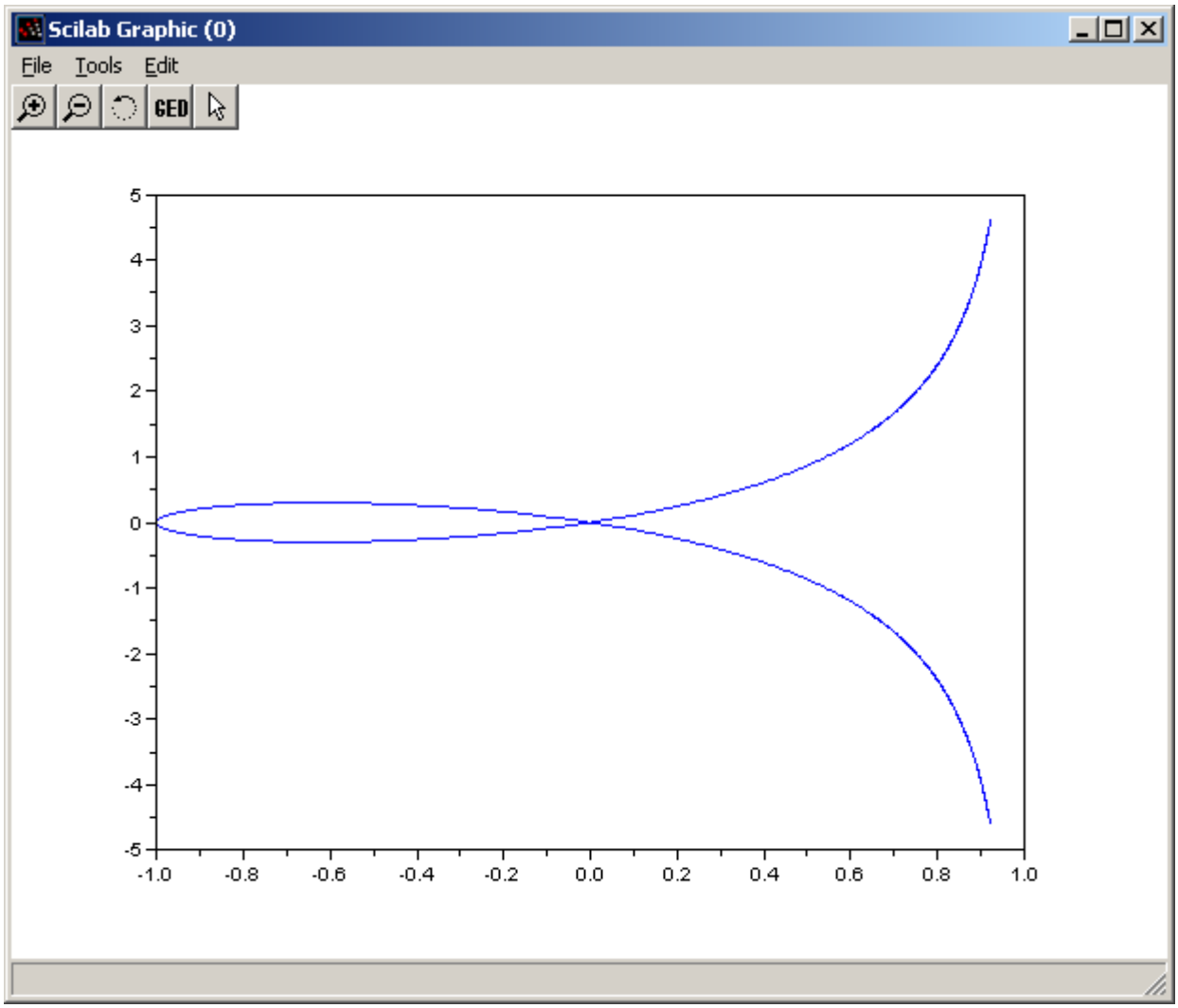

Рис. 4.17. График строфоиды

В качестве еще одного примера параметрического графика рассмотрим полукубическую параболу $\begin{matrix} x(t)=0.5t^2 \\ y(t)=0.3t^3 \end{matrix}$ $t=-3,...,3$ (см. листинг 4.13, рис. 4.17).

```
t=-3:0.01:3;
x=0.5*t.^2;
y=0.3*t.^3;
plot(x,y);
```

Листинг 4.14.

5. Оформление графиков

Рассмотрим основные возможности **Scilab** по оформлению графиков.

5.1. Изображение сетки на графике

Для изображения сетки следует воспользоваться функцией `xgrid(color)`, `color` определяет `id` цвета линии сетки.

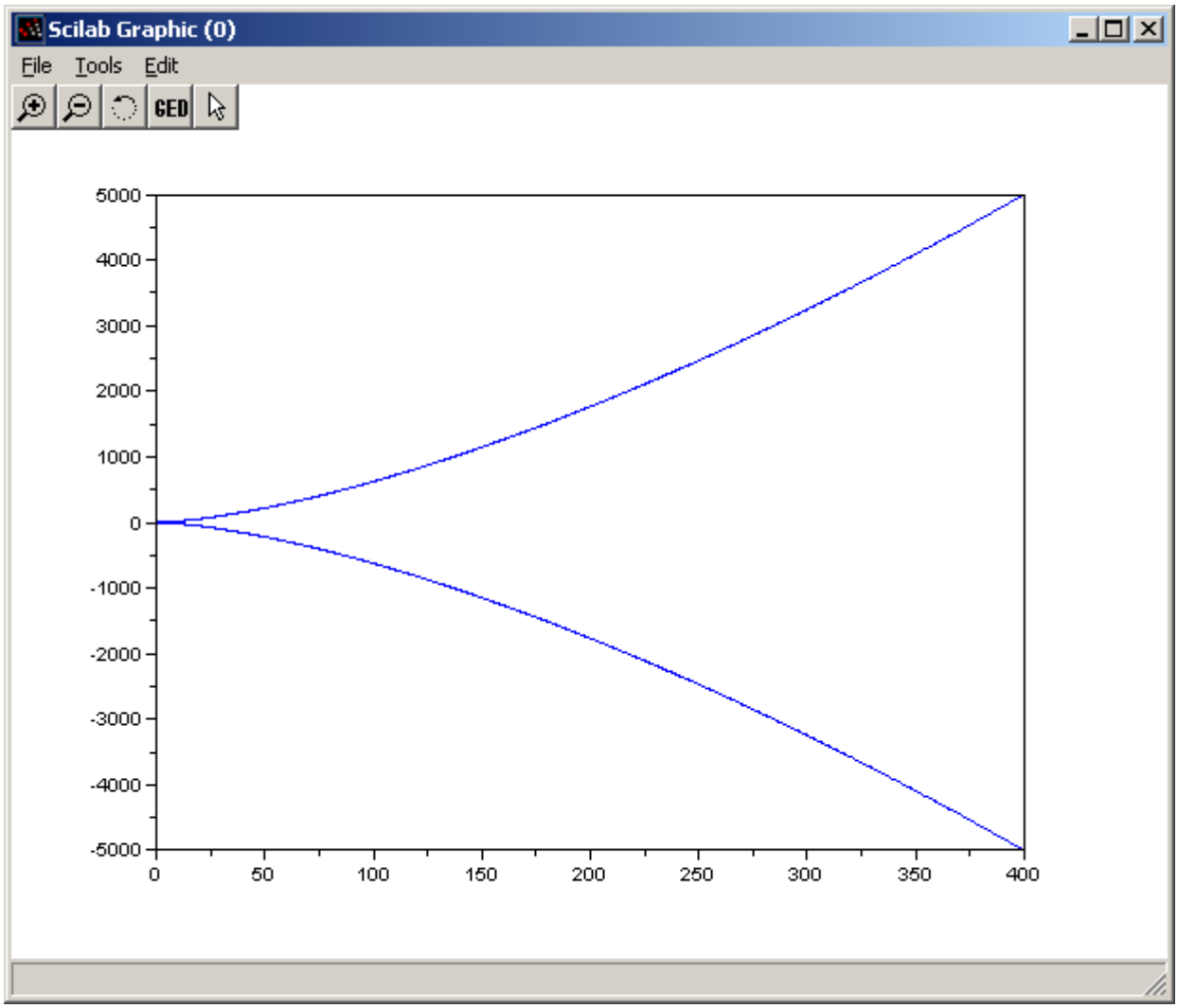

Рис. 4.18. График полукубической параболы

5.2. Заголовки на графике

Для вывода заголовков на графике служит функция

`xtitle(title, xstr, ystr),`

Здесь `title` – название графика, `xstr` – название оси `X`, `ystr` - название оси `Y`.

5.3. Нанесение легенд на график

Для нанесения легенд на график служит функция

```
legend(leg1,leg2,...,legn[,pos][,boxed])
```

`leg1` – имя первой легенды, `leg2` – имя второй легенды, ... , `legn` – имя n-й легенды; `pos` – месторасположение легенды: `1` – верхнем правом углу (по умолчанию), `2` – верхнем левом углу, `3` – нижнем левом углу, `4` – нижнем правом углу, `5` - определяется пользователем после изображения графика; `boxed` – логическая переменная (по умолчанию `%t`), которая определяет рисовать или нет рамку вокруг легенды.

Рассмотри пример оформления графика (см. листинг 4.15, рис. 4.19).

```
x=-10:0.01:10;
y=sin(cos(x));
z=cos(sin(x));
plot(x,y,'r',x,z,'b');
xgrid();
xtitle('Grafic y=f(x)','X','Y');
legend('sin(cos(x))','cos(sin(x))',3,%f);
```

Листинг 4.15

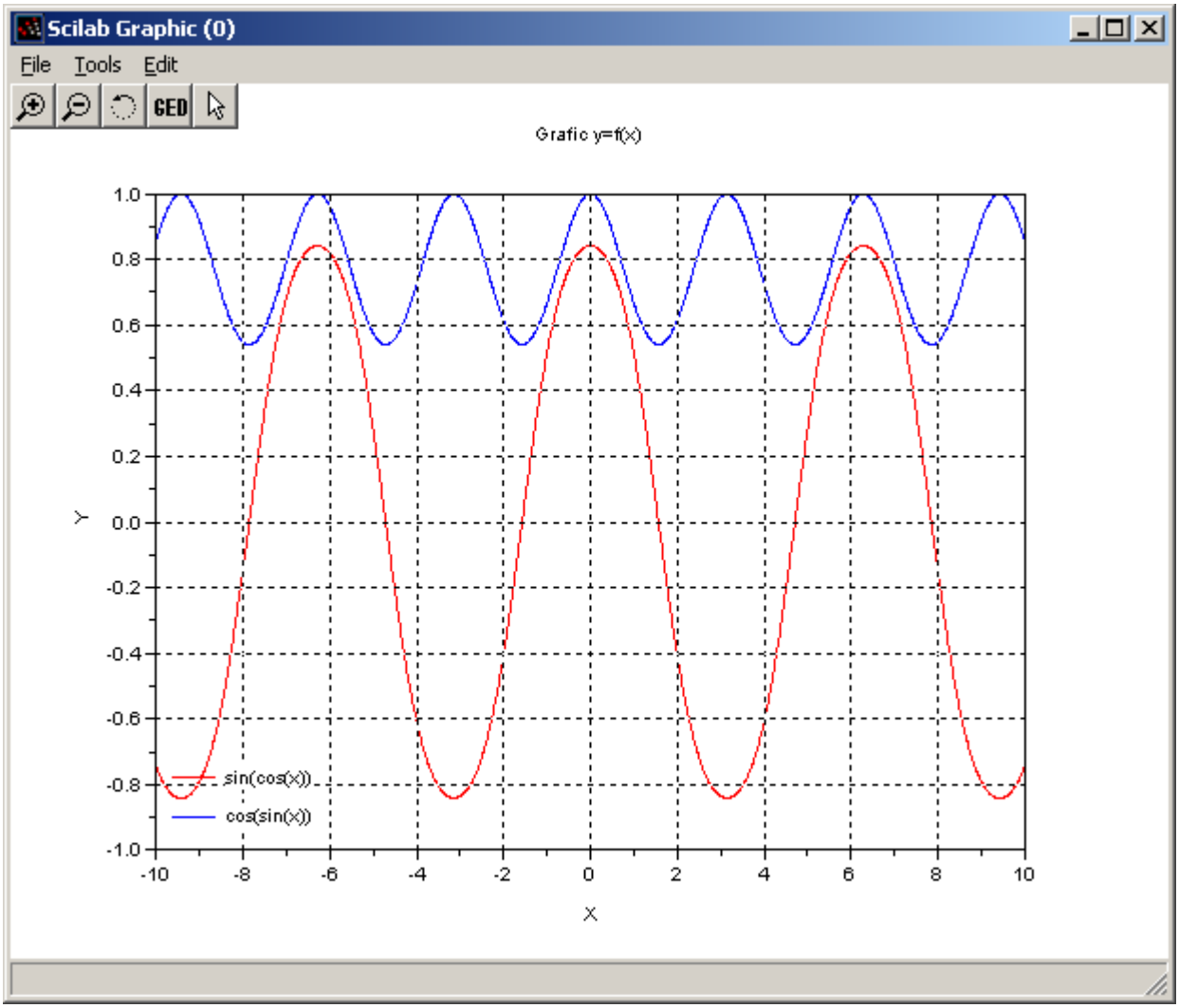

Рис. 4.19. Графики функций y=sin(cos(x)), z=cos(sin(x)).

Кроме того, Scilab позволяет после вывода графика перейти в режим форматирования с помощью команды `Edit - Figure proterties` в окне график. Окно форматирования графика представлено на рис. 4.20.

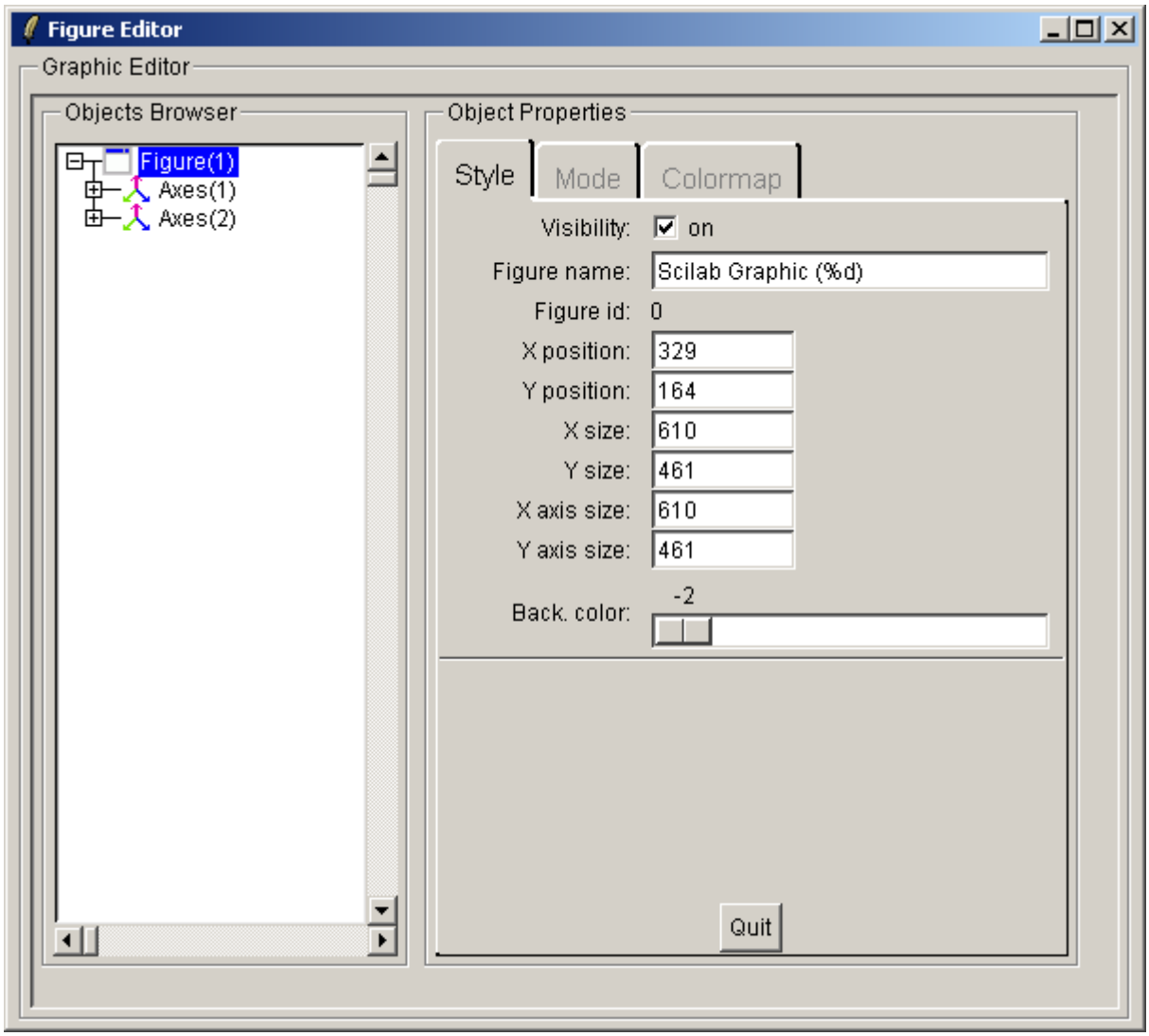

Рис. 4.20.

5.4. Построение нескольких графиков в одном графическом окне

Для построения нескольких графиков в одном графическом окне необходимо сформировать область в этом окне и в ней вывести график. Для формирования области в графическом окне служит команда

```
plotframe(rect, tics [,grid,bound, title,x-leg,y-leg,x,y,w,h])
```

`rect` – вектор `[xmin, ymin, xmax, ymax]`, который определяет границы изменения x и y координат области;

`tics` – вектор `[nx, mx, ny, my]`, который определяет количество линий сетки по оси X (`mx`) и Y (`my`), величины `nx` и `ny` должны определять число подинтервалов по осям X и Y;

`grid` – логическая переменная, которая определяет наличие или отсутствие координатной сетки;

`bound` – логическая переменная, которая при значениии true позволяет игнорировать параметры tics(2) и tics(4);

`title` – заголовок, который будет выводится над графическим окном;

x-leg**,** y-leg – подписи **x** и **y** осей графика;

x**,**y - координаты верхнего левого угла области в графическом окне, **w** – ширина, h – высота окна, значения x**,** y**,** w**,** h измеряются в относительных единицах и лежат в диапазоне [0,1].

После определения области в него можно вывести график функции с помощью plot, plot2d и т.д.

Рассмотрим пример построения четырех осей координат в графическом окне и вывода в каждую из них соответствующего графика y=sin(2x), z=cos(3x), u=cos(sin(2x)) и v=sin(cos(3x)).

```
  x=[-10:0.01:10];
  y=sin(2*x);
  z=cos(3*x);
  u=cos(sin(2*x));
  v=sin(cos(3*x));
  rect=[min(x),-1,max(x),1];
  tics=[2,11,10,5];
  plotframe(rect,tics,[%t,%t],["Function    y=sin(2x)",    "X","Y"],
[0,0,0.5,0.5])
  plot(x,y);
  plotframe(rect,tics,[%f,%f],["Function    y=cos(3x)",    "X","Y"],
[0.5,0,0.5,0.5])
  plot(x,z);
  plotframe(rect,tics,[%f,%f],["Function y=cos(sin(2x))", "X","Y"],
[0,0.5,0.5,0.5])
  plot(x,u);
  plotframe(rect,tics,[%t,%t],["Function y=sin(cos(3x))", "X","Y"],
[0.5,0.5,0.5,0.5])
  plot(x,v);
```

Листинг 4.15

Получаемый график изображен на рис. 4.21.

Еще одним способом изображения нескольких графиков в одном окне является использование функции subplot, которая разделяет графическое окно на несколько отдельных графиков. Обращение к ней имеет вид:

subplot(m,n,p) или subplot(mnp)

Графическое окно разбивается на m окон по вертикали и n окон по горизонтали, текущим окном становится окно с номером p.

В качестве примера рассмотрим построение шести графиков y=sin(x), z=cos(x), u=cos(sin(x)) и v=sin(cos(x)), w=exp(sin(x)), r=exp(cos(x)) (см. листинг 4.16 и рис. 4.22).

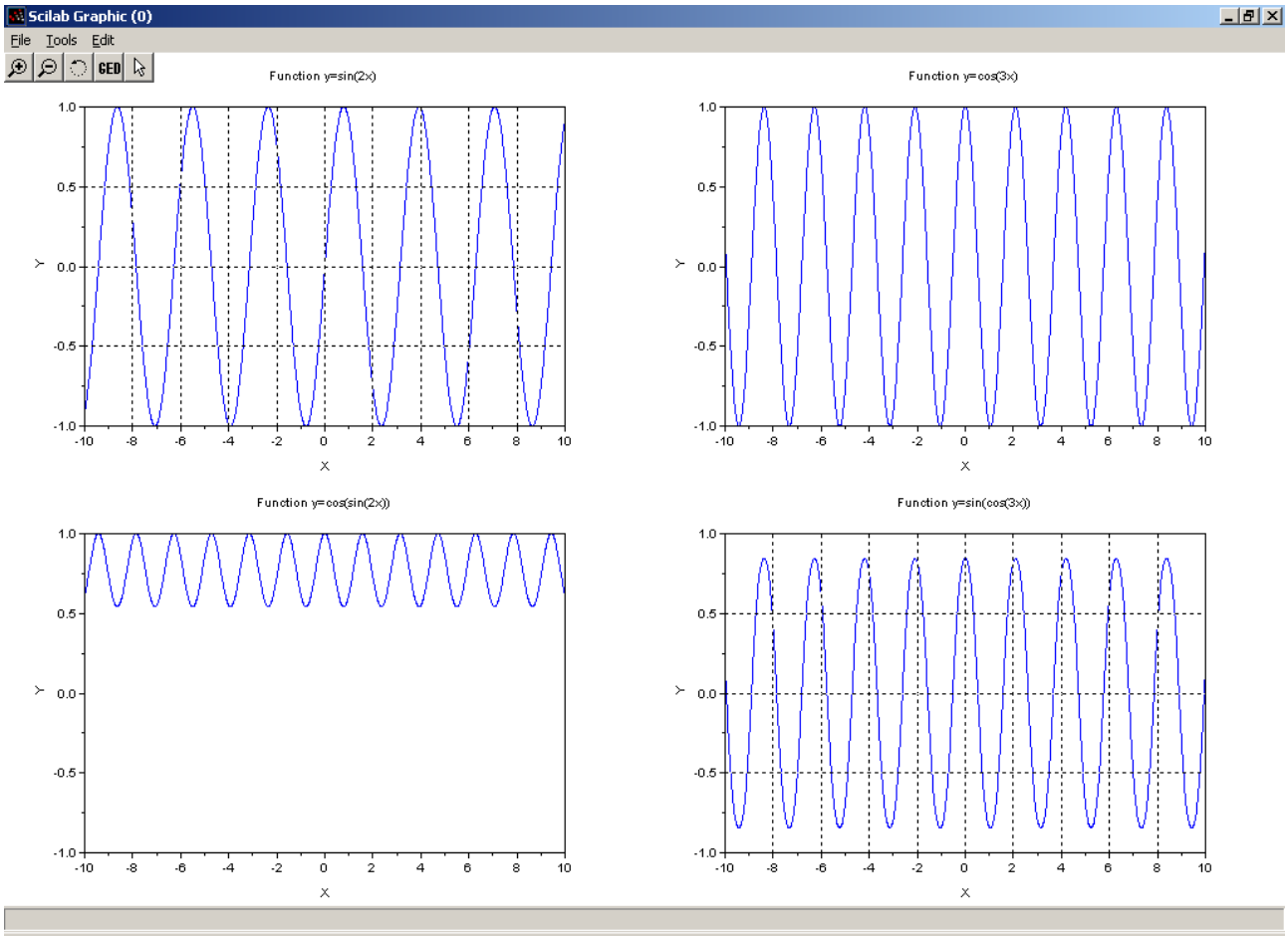

Рис. 4.21. Графики четырех функций в одном графическом окне

```
x=[-10:0.01:10];
y=sin(x);
z=cos(x);
u=cos(sin(x));
v=sin(cos(x));
w=exp(sin(x));
r=exp(cos(x));
subplot(3,2,1);
plot(x,y);
xtitle('Function y=sin(x)','X','Y');
subplot(3,2,2);
plot(x,z);
xtitle('Function z=cos(x)','X','Z');
subplot(3,2,3);
plot(x,u);
xtitle('Function u=cos(sin(x))','X','U');
subplot(3,2,4);
plot(x,v);
xtitle('Function v=sin(cos(x))','X','V');
subplot(3,2,5);
plot(x,w);
xtitle('Function w=exp(sin(x))','X','W');
subplot(3,2,6);
plot(x,r);
xtitle('Function r=sin(x)','X','R');
```

Листинг 4.16.

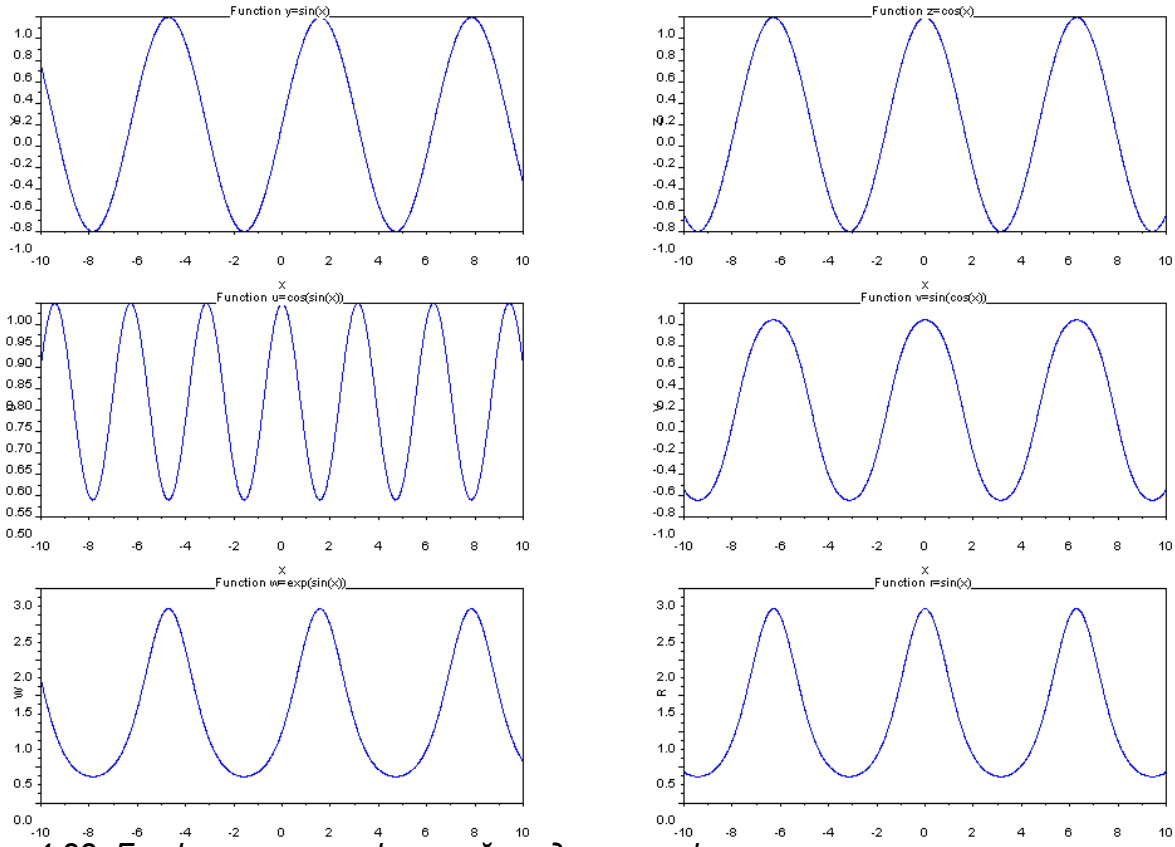

Рис. 4.22. Графики шести функций в одном графическом окне

Любой график можно экспортировать в графический файл для этого в окне графика выбрать команду `File - Export` и в появившемся окне (см. рис. 4.23) выбрать тип сохраняемого файла, цвет (`Color`(цветной), `Black and White` (черно-белый)), ориентация рисунка. После это можно выбрать папку и имя экспортируемого файла (см. рис. 4.24).

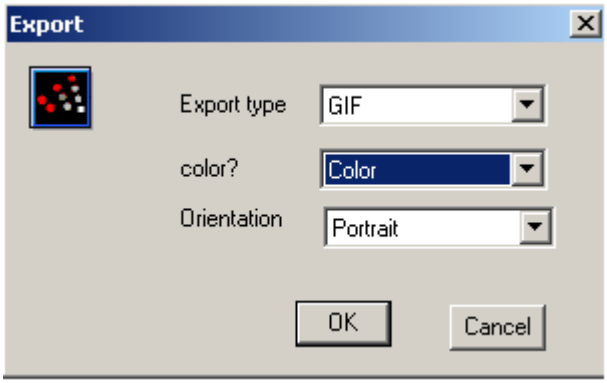

Рис. 4.23. Параметры экспорта графика в графический файл.

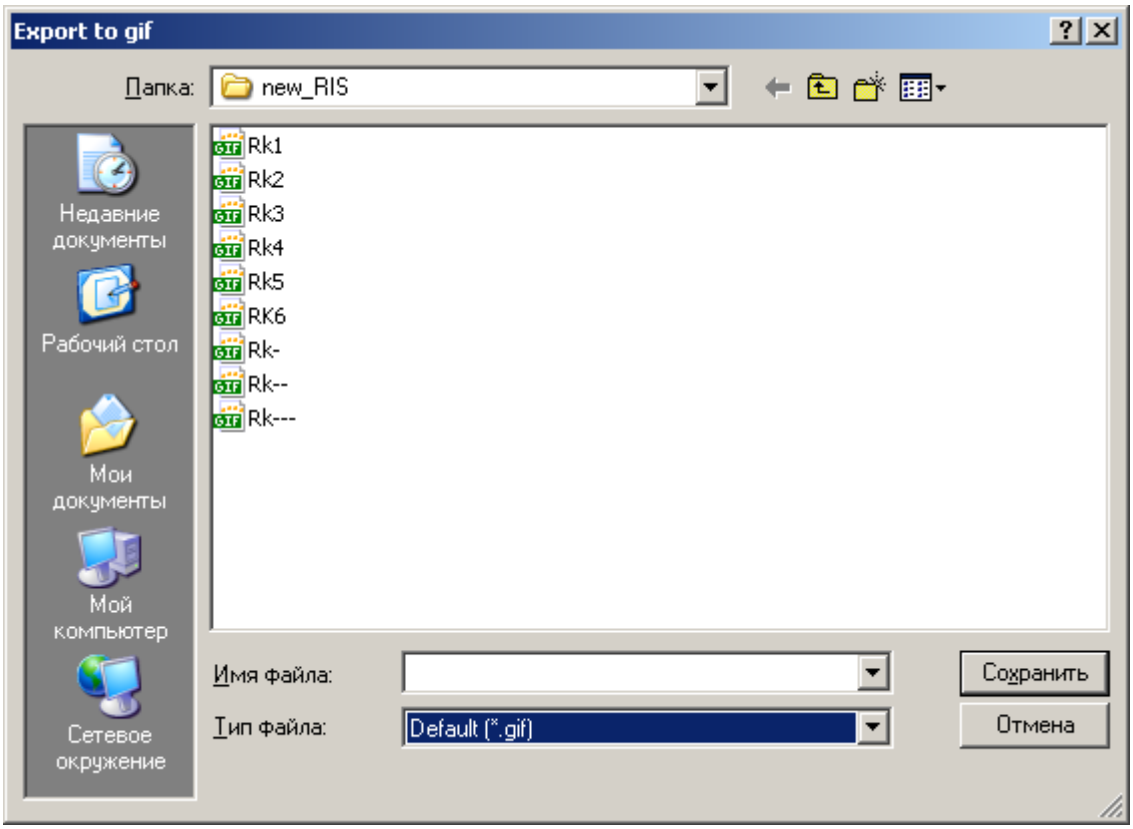

Рис. 4.24. Выбор папки и файла для экспорта графика