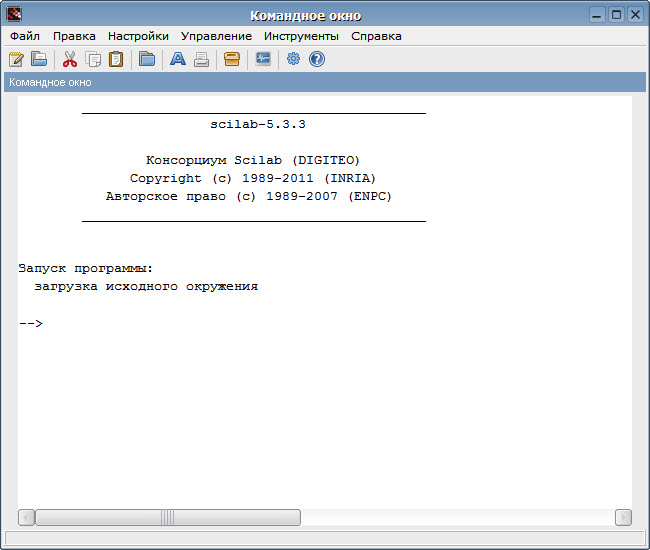
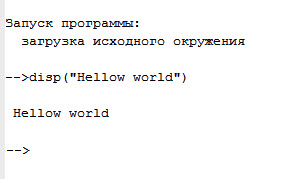
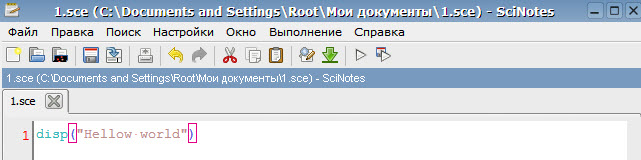
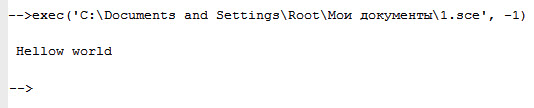
**Основы работы в SciLab. Пользовательские и системные переменные. Математические выражения. Комментарии**  
  
При запуске открывается командное окно.  
  
Есть 2 варианта работы: 1 — это работа в том же командном окне, 2 — открыть SciNotes(что-то вроде блокнота с подсветкой) где можно написать код, который позднее запустить, результат выполнения появится в командном окне.  
  
Для примера я рассмотрю вывод Hellow world.  
  
Командная строка. Используем функцию вывода на дисплей disp()  
  
  
При работе в SciNotes вы получите что то похожее  
  
для выполнения кода, надо или нажать на стрелочку в право :) (как во многих средах разработки)  
или Выполнение->… без отображение команд  
в принципе можно использовать и другие методы выполнения, и не использовать вывод на экран  
Результатом выполнения будет:  
  
т.к. можно сказать что интерфейс изучен, далее я буду приводить просто код и результат выполнения  
  
SciLab чувствителен к реестру, т.е. А и а — разные переменные переменные.

*a=1,A=3  
//Каждая операция начинается с новой строки или через запятую  
//коментарии можно оставлять после двух символоф слэш  
b=3  
c=a+b  
disp©*

**Основные операции:**  
+ сложение  
— вычитание  
\* умножение  
/ деление справа, т.е. x/y = xy^(-1)  
\ деление слева, т.е. x\y = x^(-1)y  
^ возведение в степень, т.е. x^y  
\*\* возведение в степень (эквивалентно ^)  
’ эрмитово сопряжение (комплексное сопряжение и транспонирование)  
  
**3. Основы работы в SciLab. Функции и их типы. Способы объявления пользовательских функций. Использование функций. Файлы-сценарии и их применение для хранения функций.**  
  
Элементарные математические функции.

acos acosd acosh acoshm acosm acot acotd acoth  
acsc acscd acsch asec asecd asech asin asind  
asinh asinhm asinm atan atand atanh atanhm atanm  
cos cosd cosh coshm cosm cotd cotg coth  
cothm csc cscd csch sec secd sech sin  
sinc sind sinh sinhm sinm tan tand tanh  
tanhm tanm  
exp expm log log10 log1p log2 logm max  
maxi min mini modulo pmodulo sign signm sqrt  
sqrtm

типы в SciLab

y = int8(x) 8-битовое число со знаком [-2^7; (2^7)-1] = [-128; 127]  
y = uint8(x) 8-битовое число без знака [0; (2^8)-1] = [0; 255]  
y = int16(x) 16-битовое число со знаком [-2^15; (2^15)-1] = [-32768; 32767]  
y = uint16(x) 16-битовое число без знака [0; (2^16)-1] = [0; 65535]  
y = int32(x) 32-битовое число со знаком [-2^31; (2^31)-1] = [-2147483648; 2147483647]  
y = uint32(x) 32-битовое число без знака [0; (2^32)-1] = [0; 4294967295]

iconvert преобразование к целочисленному представлению  
inttype определение типа целого числа  
  
простейший способ вызова пользовательской функции:

*outvar = myfunction ( invar )*

пример пользовательской фунции:

*function y = myfunction ( x )  
y = 2 \* x  
endfunction*

Сохраняем её. Далее приведён пример вызова данной функции

*-->exec('D:\PRIVATE\Учёба\КСВЭ\myfunction.sci', -1)  
-->y = myfunction ( 3 )  
y =  
  
6.*

**4. Определение одномерный и многомерных массивов. Основные действия над массивами.**  
  
Пример, как задаётся одномерный массив:

*-->A = [1, 2, 3, 4, 5, 6]  
A =   
1. 2. 3. 4. 5. 6.*

Задание двухмерного массива:

*-->A = [1, 2, 3; 4, 5, 6]  
A =   
1. 2. 3.   
4. 5. 6.*

квадратные скобки ”[” и ”]” обозначают начало и конец перечисления  
элементов матрицы,  
запятой ”,” отделяются элементы матрицы, находящиеся в одной строке,  
точка с запятой ”;” разделяет строки матрицы.  
  
size определить размер матрицы  
matrix изменить размер матрицы  
resize\_matrix создать новую матрицу заданного размера и скопировать  
в нее элементы из исходной матрицы  
  
Операции над матрицами:  
**Обращение к элементам матрицы**  
i = 1; 2, а j = 3; 4  
для этого возьмём уже готовую матрицу

*-->A = testmatrix (" hilb ", 5)  
A =  
25. — 300. 1050. — 1400. 630.  
— 300. 4800. — 18900. 26880. — 12600.  
1050. — 18900. 79380. — 117600. 56700.  
— 1400. 26880. — 117600. 179200. — 88200.  
630. — 12600. 56700. — 88200. 44100.  
-->A(1: 2, 3: 4)  
ans =  
1050. — 1400.  
— 18900. 26880*

A матрица целиком  
A(:,:) матрица целиком  
A(i:j,k) элементы матрицы в k-ом столбце с i-ой по j-ую строку  
A(i,j:k) элементы матрицы в i-ой строке с j-ого по k-ый столбец  
A(i,:) i-ая строка матрицы  
A(:,j) j-ый столбец матрицы  
  
**Генерация единичной матрицы**

*-->A = ones (3, 3)  
A =   
1. 1. 1.   
1. 1. 1.   
1. 1. 1.*

**Операции над матрицами**  
+ сложение .+ поэлементное сложение  
— вычитание .- поэлементное вычитание  
\* умножение .\* поэлементное умножение  
/ деление справа ./ поэлементное деление справа  
\ деление слева .\ поэлементное деление слева  
^ или \* возведение в степень :^ поэлементное возведение в степень  
’ эрмитово сопряжение (комплексное сопряжение и транспонирование)  
.’ транспонирование без сопряжения  
  
пример умножения числа на еденичную матрицу 2 на 2

*-->B = 2 \* ones (2, 2)  
B =   
2. 2.   
2. 2.*

**5. Определение одномерный и многомерных массивов. Специальные матричные функции**  
  
**функции работы с матрицами**  
chol разложение Холесского  
companion сопровождающая матрица  
cond число обусловленности  
det определитель матрицы  
inv обратная матрица  
linsolve решение систем линейных уравнений  
lsq метод наименьших квадратов  
lu LU-разложение с выбором опорного элемента  
qr QR-разложение  
rcond обратное число обусловленности  
spec собственные значения и векторы  
svd разложение по сингулярным числам матрицы  
testmatrix генерация специальных матриц (Гильберта, Франка и др.)  
trace след матрицы  
  
**6. Определение одномерный и многомерных массивов. Решение СЛАУ. Символьные массивы и операции над ними**  
  
**Текст файла–сценария с решением задачи по формулам Крамера**

*-->A=[2 1 -5 1;1 -3 0 -6;0 2 -1 2;1 4 -7 6];//Матрица коэффициентов  
-->b=[8;9;-5;0]; //Вектор свободных коэффициентов  
-->//Первая вспомогательная матрица  
-->A1=A;A1(:,1)=b;  
-->//Вторая вспомогательная матрица  
-->A2=A;A2(:,2)=b;  
-->//Третья вспомогательная матрица  
-->A3=A;A3(:,3)=b;  
-->//Четвертая вспомогательная матрица  
-->A4=A;A4(:,4)=b;  
-->//Главный определитель отличен от нуля  
-->D=det(A);  
-->//Определители вспомогательных матриц  
-->d(1)=det(A1);  
-->d(2)=det(A2);  
-->d(3)=det(A3);  
-->d(4)=det(A4);  
-->//Вектор неизвестных  
-->x=d/D  
x =   
3.   
— 4.   
— 1.   
1.   
-->//Проверка  
-->P=A\*x-b  
P =   
0.   
0.   
— 8.882D-16   
2.665D-15*

**Решение системы методом Гаусса**

*-->A=[2 -1 1;3 2 -5;1 3 -2];  
-->b=[0;1;4];  
-->//Приведение расширенной матрицы к треугольному виду  
-->C=rref([A b]);  
-->//Выделение последнего столбца из матрицы,  
-->//x — решение системы  
-->x=C(1:3,4:4)  
x =   
0.4642857   
1.6785714   
0.75   
-->A\*x //Проверка  
ans =   
— 5.551D-16   
1.   
4.*

**7. Численное интегрирование. Подходы к интегрированию. Интегрирование функций заданных пользователем**  
  
Численное интегрирование (историческое название: (численная) квадратура) — вычисление значения определённого интеграла (как правило, приближённое). Под численным интегрированием понимают набор численных методов отыскания значения определённого интеграла.  
**Интегрирование по методу трапеций**  
проинтегрируем функцию, корень из 2\*x-1 на отрезке от 1 до 10 с разбиением в 1 шаг

*-->x=1:10  
x =   
1. 2. 3. 4. 5. 6. 7. 8. 9. 10.   
-->y=sqrt(2\*x-1)   
y =   
column 1 to 6   
1. 1.7320508 2.236068 2.6457513 3. 3.3166248   
column 7 to 10   
3.6055513 3.8729833 4.1231056 4.3588989   
-->inttrap(x,y)  
ans =   
27.211585*

**Квадратурные формулы Ньютона Котеса**

*-->integrate('(2\*x-1)^0.5','x',5,13)  
ans =   
32.666667*

**8.Численное дифференцирование. Подходы к дифференцированию.**  
  
Численное дифференцирование — совокупность методов вычисления значения производной дискретно заданной функции.  
В основе численного дифференцирования лежит аппроксимация функции, от которой берется производная, интерполяционным многочленом. Все основные формулы численного дифференцирования могут быть получены при помощи первого интерполяционного многочлена Ньютона (формулы Ньютона для начала таблицы).  
в точке

*-->function f=myf(x), f=(x+2)^3+5\*x, endfunction;   
-->numdiff(myf,1)  
ans =   
32.   
-->x=1;3\*(x+2)^2+5  
ans =   
32.*

на отрезке

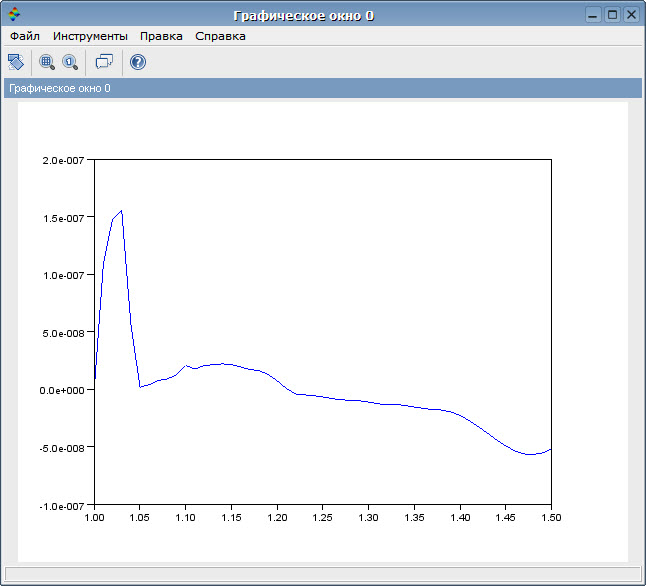
*v=0:3;   
-->numdiff(my,v)  
ans =   
17. 0. 0. 0.   
0. 32. 0. 0.   
0. 0. 52.999999 0.   
0. 0. 0. 80.000002*

и проверка

*-->function f1=my1(x), f1=3\*(x+2)^2+5, endfunction;   
-->my1(v)  
ans =   
17. 32. 53. 80.*

**9. Решение ОДУ средствами SciLab. Функции применяемые для решение ОДУ. Решение краевых задач.**  
  
Существует 4 способа для решения ОДУ:  
1. С помощью команды ode, которая является солвером для решения обыкновенного  
дифференциального уравнения.   
2. С помощью команды odedc, которая вычисляет решение смешанной дискретно-  
непрерывной системы.  
3. Команда dassl, которая дает решение неявно выраженного дифференциального уравнения.  
4. С помощью команды impl, которая дает решение неявно выраженного линейного  
дифференциального уравнения.  
  
Пример:

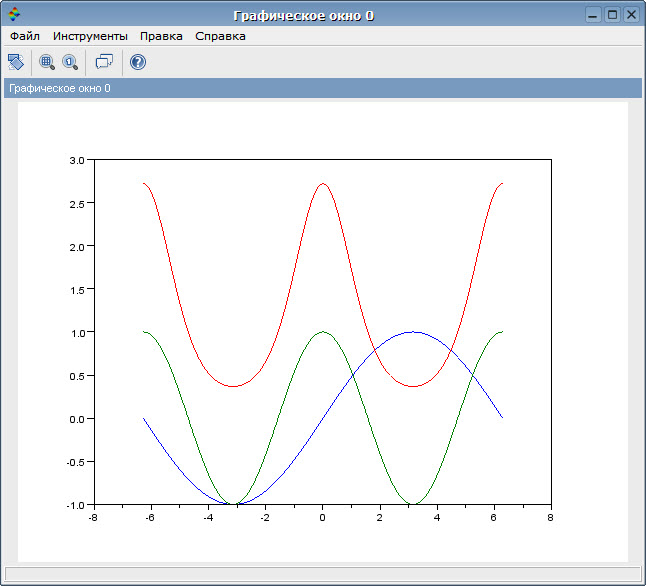
*-->y0=1;   
-->t0=1;   
-->t=1:0.01:1.5;   
-->deff("[ydot]=f(t,y)",«ydot=y^(1/3)\*t»)   
-->y=ode(y0,t0,t,f);   
-->y\_exact=((t^2+2)/3)^(1.5);// это функция точного решения для сравнения   
-->my\_er=y-y\_exact;   
-->plot(t,y-y\_exact) // это график ошибки вычисления от аргумента t*

результатом является такой график  
  
за одно можно увидеть Графическое окно. Построение графиков будет подробно рассмотрено далее.  
  
**10. Построение двухмерных графиков в системе SciLab. Основные функции и типы графиков.**  
  
**Функция plot**  
рассмотрим пример:

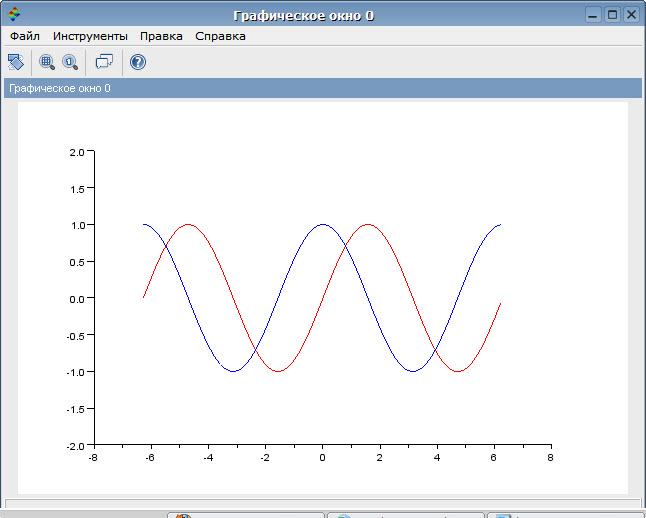
*x=-2\*%pi:0.1:2\*%pi;  
y=sin(cos(x));  
plot(x,y);*

как можно заметить первый параметр функции — это отрезок, а второй функции  
так же существует возможность нарисовать сразу несколько функций, если их перечислить:

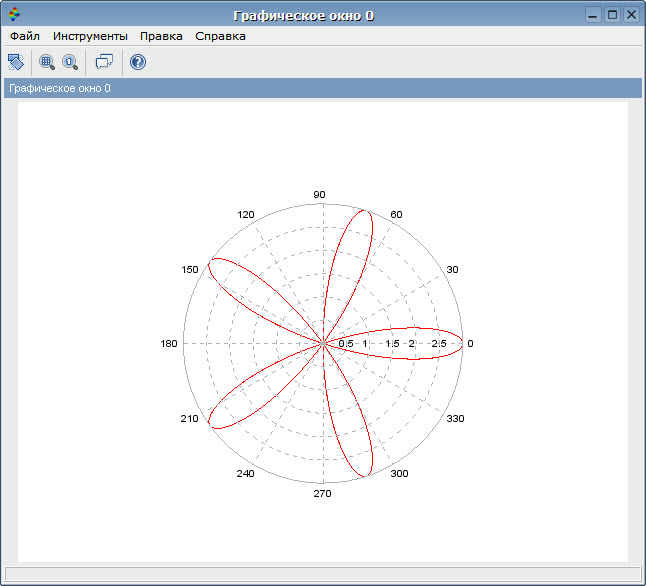
*x=-6.28:0.02:6.28;  
y=sin(x/2);  
z=cos(x);  
v=exp(cos(x));  
plot(x,y,x,z,x,v);*

  
  
**Функция plot2d**  
Рассмотрим функцию опять на примере:

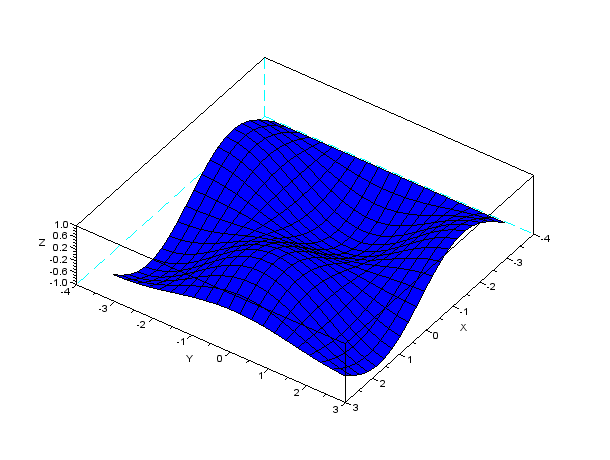
*x=[-2\*%pi:0.1:2\*%pi];  
y=[sin(x); cos(x)];  
plot2d(x,y',style=[color(«red»),color(«blue»)],rect=[-8,-2,8,2])*

  
как мы можем увидеть, у этой функции намного больше функционала.  
фунции передаются сразу массивом, так же можно указать цвет линий и отрезок.  
  
**Функция polarplot**  
Служит для построения графика в полярных координатах

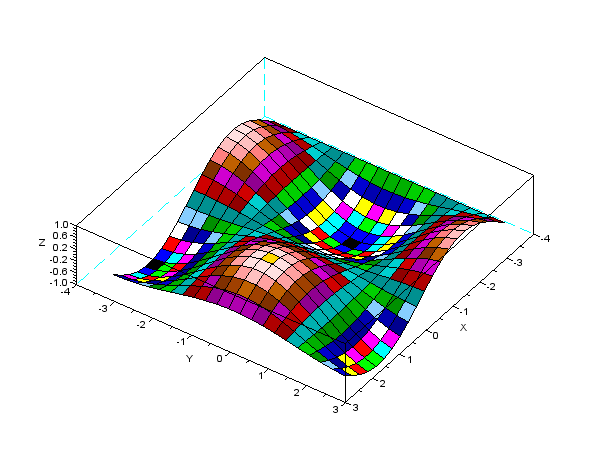
*fi=0:0.01:2\*%pi;  
ro=3\*cos(5\*fi);  
ro1=3\*cos(3\*fi);  
polarplot(fi,ro,style=color(«red»));*

получается ромашка  
  
параметры похожи как и в случае с plot2d  
  
**11. Построение трёхмерных графиков в системе SciLab. Основные функции и типы графиков.**  
  
Существует 4 способа построение графика:  
Способ 1.   
С помощью команды plot3d. Команда создает 3D график по точкам, заданным матрицами  
x, y и z.   
Способ 2.   
С помощью команды plot3d1. Команда создает 3D график по точкам, заданным  
матрицами x, y и z с помощью уровней цвета. Вещь в общем избыточная: величина  
координаты z дополнительно еще и покрашена, в зависимости от принимаемого значения  
z.   
Способ 3.   
С помощью команды fplot3d. Это аналог команды fplot3d, но изображаемая поверхность  
задана с помощью внешней функции.   
Способ 4.   
С помощью команды fplot3d1. Это аналог команды plot3d1, но изображаемая поверхность  
задана с помощью внешней функции.   
Синтаксис этих команд смотри с помощью help.   
  
Пример 1.

*t=-%pi:0.3:%pi;   
plot3d(t,t,sin(t)'\*cos(t),35,45,'X@Y@Z',[2,2,4]);*

  
забыл указать, что в графическом окне есть возможность экспорта данных, т.е. сохранить картинку  
  
Пример2.

*t=-%pi:0.3:%pi;  
plot3d1(t,t,sin(t)'\*cos(t),35,45,'X@Y@Z',[2,2,4]);*

  
  
Пример 3.

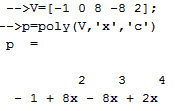
*deff('[z]=surf(x,y)','z=sin(x)\*cos(y)');   
t=-%pi:0.3:%pi;   
fplot3d(t,t,surf,35,45,«X@Y@Z»);*

Результат такой же как в примере 1  
  
Пример 4.

*deff('[z]=surf(x,y)','z=sin(x)\*cos(y)');   
t=-%pi:0.3:%pi;   
fplot3d1(t,t,surf,35,45,«X@Y@Z»);*

результат такой же, как в примере 2  
  
**12. Задача полиномов в SciLab. Символьные операции с полиномами. Решение алгебраический уравнений. Сравнение функций fsolve и roots.**  
  
Рассмотрим на примере решения уравнения 2x^4-8x^3+8x^2-1=0

*V=[-1 0 8 -8 2];   
p=poly(V,'x','c')*

  
В массиве указываются коэффициенты при х  
После построения полинома, попробуем получить решение:

*X=roots(p)  
X =   
2.306563   
1.5411961   
— 0.3065630   
0.4588039*

Для решения трансцендентных уравнений в применяют функцию Scilab fsolve(x0,f)  
задача https://habrastorage.org/storage2/e28/837/cd2/e28837cd20868be83860e8530d84c250.jpg

*-->deff('[y]=f1(x)','y1=((x-1)^2)^(1/3),y2=(x^2)^(1/3),y=y1-y2')  
-->fsolve(0,f1)  
ans =   
0.5*