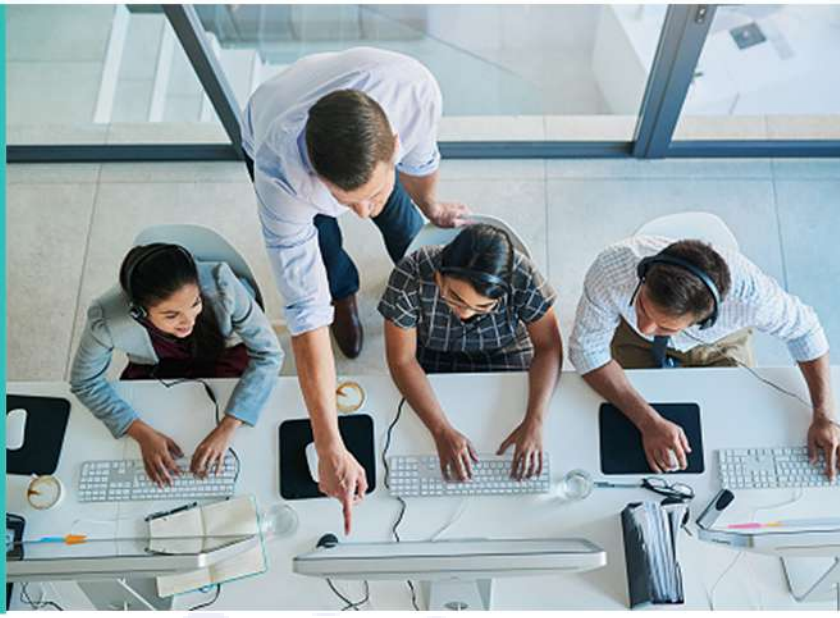
Andhra Pradesh State Skill Development Corporation

Skill AP
APSSDC

eSkill AP
Learn Anytime Anywhere
Andhra Pradesh State Skill Development Corporation



SciLab

Getting started with Scilab

Getting started with Scilab

INTRODUCTION

In this brief chapter, we discuss some very basic features of the Scilab environment which are helpful in working with Scilab effectively and efficiently. These are mainly house-keeping features of Scilab.

The Console

The first way is to use Scilab interactively, by typing commands in the console, analyzing Scilab result, continuing this process until the final result is computed. This document is designed so that the Scilab examples which are printed here can be copied into the console. The goal is that the reader can experiment by himself Scilab behavior. This is indeed a good way of understanding the behavior of the program and, most of the time, it allows a quick and smooth way of performing the desired computation.

The Scilab MENU BAR

The Scilab window always displays a menu bar with the following menus: File, Edit, Preference, Control, Applications,? and Toolbox. Let us look at some of these.

File Menu

The File Menu has the following menu options which are self-explanatory. These are identical to File menus in most other generic application software.

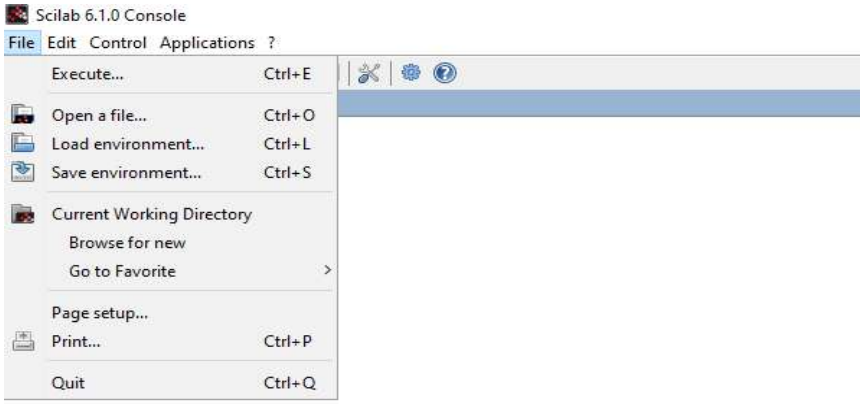


Fig.The File Menu

Edit Menu

The common edit options like Cut, Copy, Paste, Empty/Clipboard & Select all are available in this menu.

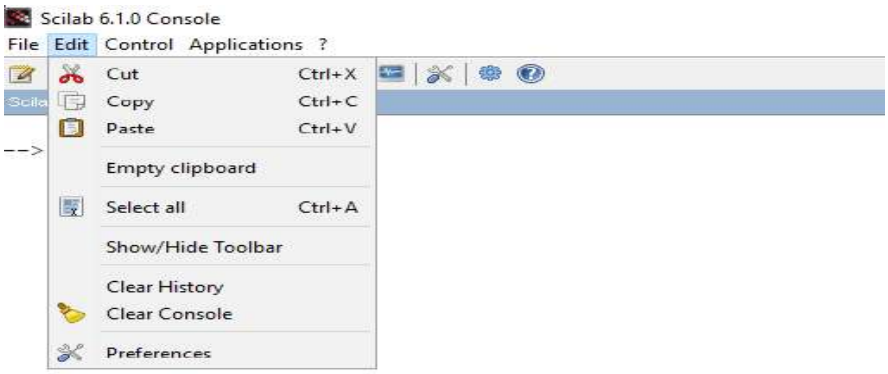


Fig. The Edit menu

Preferences Menu

Menu items in this menu are self-explanatory, they let us set the color font toolbar views history, etc.

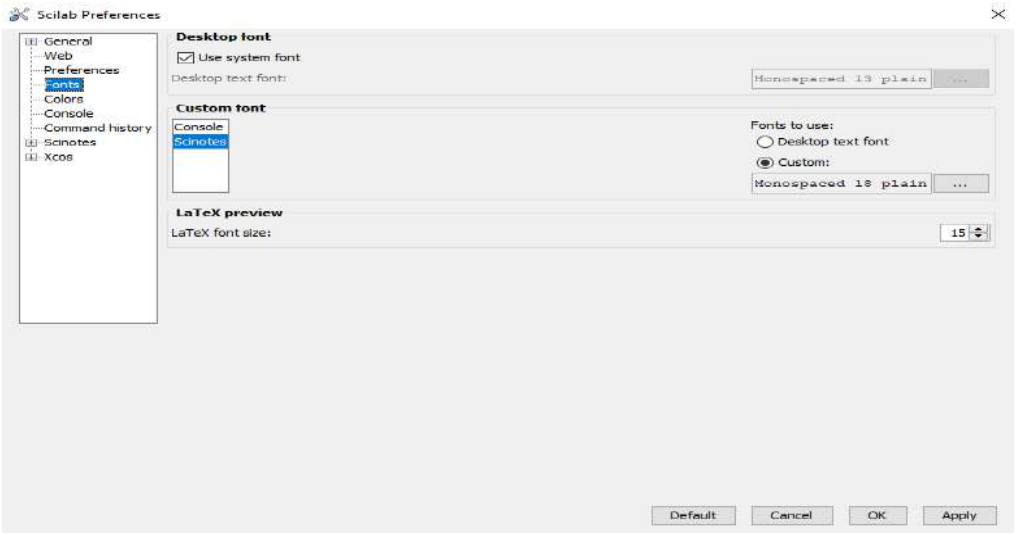


Fig. The Preferences menu

Control Menu

The Scilab window has the following Control buttons.

- Interrupt Interrupts execution of Scilab and enters in pause mode
- Resume continues execution after a pause entered as a command in a function or generated by the Stop button or Control C.
- Abort aborts execution after one (or several) pauses, and return to top-level prompt.

Fig.The Control menu

Applications Menu

We can use Scinotes, Xcos, Matlab to Scilab translator, Module manager ATOMS,variable Browser, command history, File browser in the Applications menu.

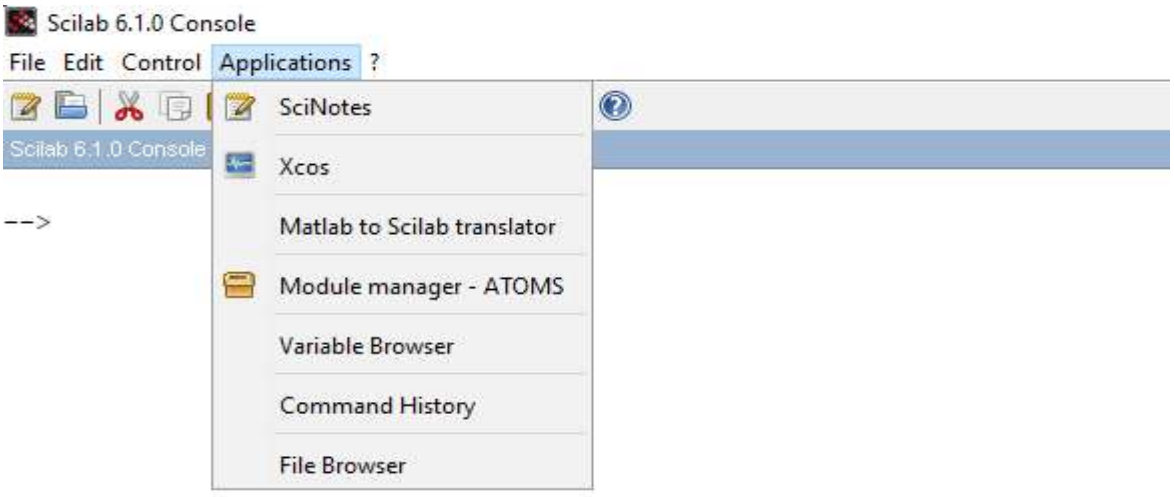


Fig.The Applications menu

Module manager(ATOMS)

ATOMS(Automatic modules management for Scilab) is the repository for packaged extension modules("Tool Boxes"). Toolboxes a collection of specialized commands/functions catering to specific areas. A large number of Toolboxes that work with Scilab are available and more are produced from time to time. Some examples are Xcos, IPCV(Image Processing & Computer Vision toolbox), Statistics. To any toolbox, it has to be downloaded and installed. See the following screenshot of Scilab with IPCV installed. The toolbox can be launched by clicking on the Applications menu option in the Toolbox menu.

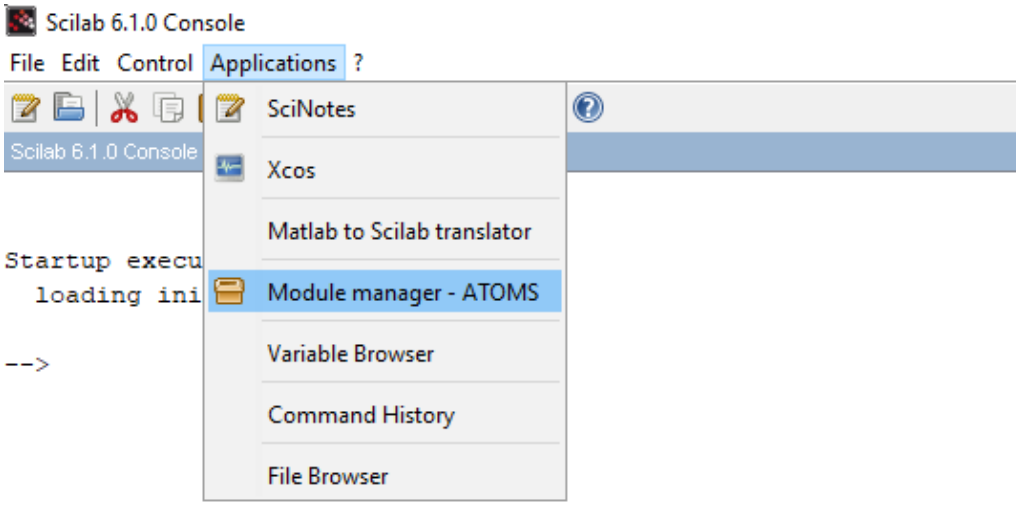


Fig. The toolboxes menu

Creating real variables

In this section, we create real variables and perform simple operations with them. Scilab is an interpreted language, which implies that there is no need to declare a variable before using it. Variables are created at the moment where they are first set. In the following example, we create and set the real variable x to 1 and perform a multiplication on this variable. In Scilab, the "=" operator means that we want to set the variable on the left hand side to the value associated to the right hand side (it is not the comparison operator, which syntax is associated to the "==" operator).

-->x =1

x =

1.

-->x = x * 2

x =

2.

The value of the variable is displayed each time a statement is executed. That behavior can be suppressed if the line ends with the semicolon ";" character, as in the following example. -->y =1; -->y=y *2; All the common algebraic operators presented in figure are available in Scilab. Notice that the power operator is represented by the hat "^" character so that computing x 2 in Scilab is performed by the "x^2" expression or equivalently by the "x**2" expression.

+ addition
- substraction
* multiplication
/ right division i.e. x/y = xy^{-1}
\ left division i.e. x\y = $x^{-1}y$
^ power i.e. x^y
** power (same as ^)
' transpose conjugate

Fig. Scilab mathematical elementary operators.

Variable name

Variable names may be as long as the user wants, but only the first 24 characters are taken into account in Scilab. For consistency, we should consider only variable names which are not made of more than 24 characters. All ASCII letters from ”a” to ”z”, from ”A” to ”Z” and from ”0” to ”9” are allowed, with the additional letters ”%”, ” ”, ”#”, ”!”, ”$”, ”?”. Notice though that variable names which first letter is ”%” have a special meaning in Scilab,which presents the mathematical pre-defined variables. Scilab is case sensitive, which means that upper and lower case letters are considered to be different by Scilab. In the following script, we define the two variables A and a and check that these two variables are considered to be different by Scilab.

```
-->A = 2

A  =

2.

 -->a = 1

 a  =

1.

-->A

A  =

2.

-->a

a  =

1.
```

MATHEMATICAL FUNCTIONS ON SCALARS

There are a small set of functions applicable to scalars alone. We introduce them below.

modulo ( )

This function is used to calculate the remainder of the division of two numbers.

- modulo(5,2)

ans = 1.

rat ( )

This function is used to find the numerator and denominator of theoretical approximation of a floating point number.

- [n,d]=rat(0.353535)

d = 28529.

n = 10086.

The result can be verified by calculating n/d

- n/d

ans = 0.353535

Yet another example is 0.3333333333333

- [n,d]=rat(0.333333333333)

d = 3.

n = 1.

Here we have to provide a relatively large number of decimals to force the result. A small number of decimal points will not produce the required result. The function rat can be called using a second argument which represents error or tolerance allowed in the approximation.

- [n,d]=rat(0.333333333,le -4)

d = 3.

n = 1.

Here the tolerance of 1/10000 produces the approximation ⅓

sqrt( )

This function returns the square root of a real or complex number. Let us see an example of a real number.

sqrt(2345.67)

ans= 48.432117

The remaining functions in this chapter are exponential functions.

exp( )

The exponential function exp returns the value of e for a real number, where e constitutes the basis of the natural logarithm. This value is given as a constant %e in Scilab.

- %e

  %e= 2.7182818

Let us see an example of finding the exponent of a real number.

- exp(2.5)

  ans= 12.182494
- exp(-3.6)

  ans=0.0273237

log( )

The logarithm is referred to as log( ). Let us see the natural logarithm of a real number.

- log(2.35)

  ans=0.8544153

Besides the natural logarithm (logarithm of base e), Scilab also provides functions log10()and log2( ) that calculate logarithms of base 10 and 2 respectively.

log10( )

Here the base of the logarithm is taken as10. If x=10y then y=log10(x). Here is an example to illustrate this:

- log10(1000)

  ans=3

log2( )

Here 2 is the base of logarithm If x=2y then y=log2(x). Let us see an example.

- log2(8)

  ans= 3.

COMPLEX NUMBERS

A complex number z can be written as z=x+iy where x and y are real numbers and is-1. an imaginary number widely used in Mathematics. Scilab supports complex numbers. We can say that x is the real part of z and y is the imaginary part of z. A complex number z=2.3+5.5 i can be written in Scilab as follows:

z=2.3+5.5*%i

z= 2.3+5.5i

We can separate out the real imaginary parts with the real( )and imag( ) functions

- a=real(z)

a= 2.3

- b=imag(z)

b=5.5

- c=real(0.6-0.8*%i)

c=0.6

- d=imag(0.6-0.8*%i)

d=-0.8

The sign function when applied to complex numbers. Return z/|z|. The resulting complex number will have a magnitude of unity.

- sign(3-4*%i)

ans= 0.6-0.8i

The abs( )function returns the magnitude of the complex number

- abs(0.6-0.8*i)

ans= 1

The function clean, ceil, floor, int, round and fix can be applied to complex numbers also

z=2.3+5.5*%i

z=2.3+5.5i

- ceil(z)

ans= 3. +6.i

- floor(z)

ans= 2.+5.i

- int(z)

ans= 2. +5.i

- round(z)

ans= 2. +6.i

- fix(z)

ans= 2.+5.i

The square root of a negative real number(x<0) is calculated as x=i|x|

- sqrt(-2345.67)

ans= 48.432117i

Logarithms also can be calculated for complex numbers

- log(2+3*%i)

ans= 1.2824747+0.9827937i

TRIGONOMETRIC FUNCTIONS

The basic trigonometric functions sine(sin), cosine(cos), and tangent (tan) are supported in Scilab. The trigonometric functions are defined in terms of the angels and sides of a right triangle.

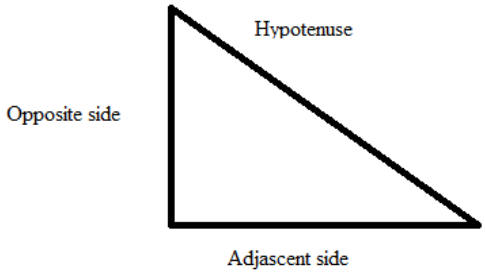


Fig. The angles and sides of a right triangle

sin(θ)=opposite side/ Hypotenuse; cos(θ)=adjacent side/ Hypotenuse; tan(θ)=opposite side/ Adjacent side

Scilab assumes the angles to be in degrees. We shall for example calculate these functions for value of θ=45

- sin(45)

ans=0.8509035

- cos(45)

ans=5253220

- tan(45)

ans=1.6197752

INVERSE TRIGONOMETRIC FUNCTIONS

If sin(y)=x then y=sin-1(x) ,If cos(y) x then y=cos-1 (x). If tan(y)=x, then y=tan-1(x) We will find these values for x=0.5 using Scilab functions a sin ( ), a cos( ) and a tan( ).

- asin(0.5)

ans=0.5235988

- acos(0.5)

ans=1.0471976

- atan(0.5)

ans= 0.4636476

HYPERBOLIC FUNCTIONS

The hyperbolic functions are the result of combining exponential functions.

sinh(x)= ex-ex2

cosh(x)=ex+e-x2

tanh(x)=sinh(x)cosh(x)

Let us see the values of hyperbolic functions for the angle x=45

- sinh(45)

ans=1.747D+19

- cosh(45)

ans= 1.747D+19

- tanh(45)

ans=1

Similar to inverse trigonometric functions, Scilab also supports inverse hyperbolic functions.