

Top Ten Scilab Errors for BioCalc

FSU Math

January 4, 2010

Contents

1	The exponential function is exp	2
2	Omitting * for multiplication	2
3	Spelling counts	2
4	Function name has been assigned a value.	2
5	Out of order commands	2
6	Rushing the assignment	2
7	Those annoying parentheses	3
8	Which quotes to use.	3
9	Confusing : with ;	3
10	Not enough points	3
11	Getting frustrated	3
12	Not naming all output	3
13	Getting to the end.	4
14	Function parameters are input and output	4

15 Exec lab.sce	4
16 Make it clear to me	4

1 The exponential function is exp

To express

e^x

you must use the exp function `exp(x)`. It is not

e^x

because `e` is just another variable to scilab and it is not

`exp^x`

2 Omitting * for multiplication

Scilab thinks `ab` is the name of a variable and `a b` is just an error. To do multiplication use `a*b`. Sometimes `a .* b` if `a` and `b` are vectors.

3 Spelling counts

Scilab thinks `sin` and `Sin` are different and of course misspellings will also cause errors.

4 Function name has been assigned a value.

For example `sin(1:10)` works but after `sin = 7`, `sin(1:10)` will fail. You need to clear `sin` to get rid of the assignment.

5 Out of order commands

```
plot(x,y);  
y=sin(x);  
x = - %pi: %pi/20 : %pi;
```

are the wrong order but if you execute it often enough it will work.

6 Rushing the assignment

Read the problem assignment before you attempt to do it.

7 Those annoying parentheses

Getting each (to match with a). The editor will help or always type () and then go back and fill in the missing elements.

8 Which quotes to use.

Cut and paste will often give you the wrong kind of quotes. Scilab uses ' and " which are both on the key to the right of the colon and semicolon which is to the right of the L key.

9 Confusing : with ;

```
z = 0:1;2;  
y = 0;1:2;  
x = 0:1:2;  
w = 0;1;2;
```

All do different things

10 Not enough points

There need to be at least 40 points on x-axis to plot a smooth function. So `x=0:10; y=sin(x); plot(x,y);` looks wrong while `x=0:0.25:10; y=sin(x); plot(x,y);` is much smoother.

11 Getting frustrated

Put your commands in a dot sce file and execute them. Look at the console for error messages. If you don't remember how to use a command try the help command.

12 Not naming all output

Retyping numbers introduces round off error, variables save more digits.

```
-->x=sqrt(2)
x =
    1.4142136
-->1.4142136^2
ans =
    2.0000001
-->x^2
ans =
    2.
```

13 Getting to the end.

A for loop starts with the word `for` and continues til the word `end`. A function definition starts with the word `function` and continues til the word `(no space) endfunction`.

14 Function parameters are input and output

The line `function z = f(x, y)` says the function `f` has two inputs, `x` and `y`, and one output, `z`. Somewhere in your function should be a line that starts `z =`

that gives the output of the function.

More often than not, the computation will just use the inputs x and y .

15 Exec lab.sce

One must download the lab.sce file to the desktop and point scilab to the desktop and exec lab.sce in order to define the functions in the lab file. Otherwise scilab thinks the functions is undefined.

If scilab thinks there is no such file, either you have not downloaded the file or scilab is not looking for the file in its download location.

16 Make it clear to me

If you do a clear command, all the functions and variables you have in your current session are removed. You have to exec lab.sce again.

If you do a clf command clears the current plot. While a clf command is often good at the start of xx.sce file that is doing plots, you don't want it in the middle or it will blank the graph you just carefully drew.

If you do a clc command, it clears the console window but does change plots, variables or functions. Not very useful.