14
Scripts

When you use a numerical computational software, you often are faced with the need to repeat the same calculation several times while modifying only some of the parameters. Therefore, it is convenient to be able to save steps of a complicated calculation to later come back and modify specific commands. For this purpose, Scilab provides the ability to write scripts which can be easily executed and modified from the text editor, as seen in chapter Preview of Scilab with the pendulum simulation or planetary motion. Here, we define a script as a list of instructions stored within a text file. These instructions can then be executed sequentially using Scilab.

14.1. Writing and executing scripts

In order to understand what a Scilab script is, we are going to look at a short example:

```
1. A=[1 2;3 4];y=[3;5];
2. x1=linsolve(A,-y);    // not displayed, even "with echo"
3. x2=A^(-1)*y           // displayed if using "with echo"
4. disp(x1,'x=')         // displayed, even if using "with no echo"
```

This script is four lines long and contains the following instructions:

1. Two instructions which define the matrices `A` and `y`.

2. Computes the solution to the system of equations $A*x=y$ by using the `linsolve` command, and stores it inside the `x1` variable. `x1` is not displayed since the line ending prevents it.

3. Computes the solution to the system of equations $A*x=y$ by using matrix multiplication with the inverse of `A`, and stores it in the variable `x2`.

4. Displays the content of variable `x1` with the command `disp`.

Several comments are also present at the end of lines 2, 3 and 4 (after the `//`). These instructions need to be saved inside a text file, for example by using the SciNotes text editor. When saving, name the file and end it with the extension `.sce`, for example `testscript.sce`.

Tip › Long instructions can also span several lines with the use of the command `...` *This command can be inserted at the location that denotes the breaking point of the logical line which divides it into two distinct lines. This can be useful to improve script legibility and avoid writing very long command lines, for instance when calling a function with several input parameters. Similarly, to define a matrix, we can enter the coefficients by returning to a new line after each* `;` *or at any other point since in this case the use of* `..` *is optional. In all cases, the result will be the same regardless of the number of lines spanned by the command.*

```
// matrix defined across several lines
M= [ 1 2 3;
    4,5,6]
// instructions spanning several lines
sum(M,..
'c')
```

which returns:

```
-->//matrix defined across several lines

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

-->//instructions spanning several lines

-->sum(M,..
-->'c')
 ans  =
    6.
    15.
```
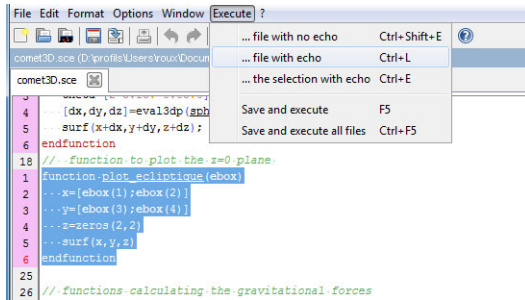
Executing a script

Once a file is saved, there are several methods you can use to execute the commands it contains:

- from the SciNotes menu bar (or the equivalent keyboard shortcut):

    – EXECUTE > ...FILE WITH NO ECHO (Ctrl+Shift+E)

    – EXECUTE > ...FILE WITH ECHO (Ctrl+L)

    – EXECUTE > ...UNTIL THE CARET, WITH ECHO or, if you've selected part of the script EXECUTER > THE SELECTION WITH ECHO (Ctrl+E for both)

***Figure 14.1 :*** *Launching a script from the SciNotes menu bar*

- from the following SciNotes toolbar icons:

  – EXECUTE

  – SAVE AND EXECUTE

  – SAVE AND EXECUTE ALL FILES

Figure 14.2 : *Launching a script from the SciNotes toolbar*
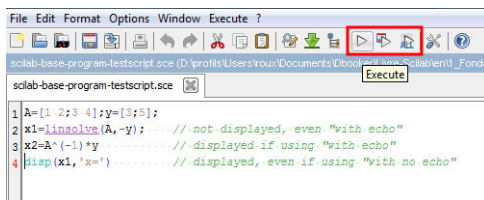
- from the Scilab console:

  – with a copy/paste of the script text into the console

  – with the menu FILE > EXECUTE (display without echo)

  – by using the `exec` command (display with or without echo according to the chosen option)

Figure 14.3 : *Launching a script from the console menu bar*
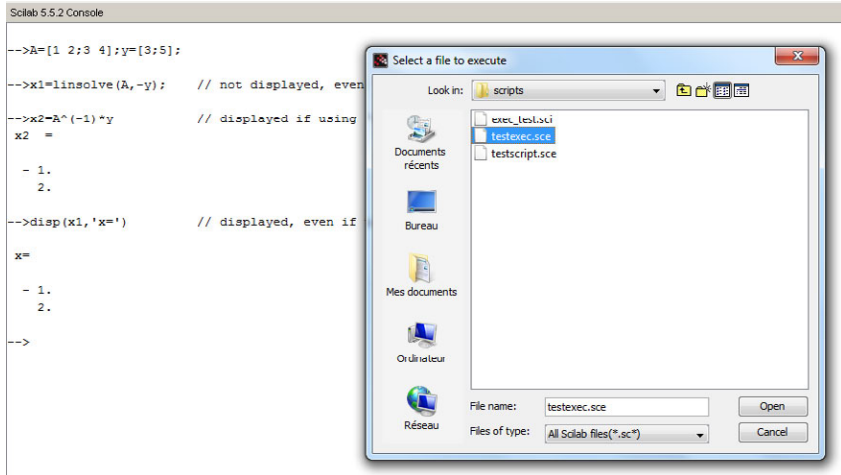
Setting the results display

Regardless of the method chosen to execute a script (out of the available options), Scilab executes the lines in the file one by one. However, during execution, what is displayed depends on the way in which the script is executed. If the chosen method is:

With echo

The commands along with their results are displayed in the console.

Without echo

The script is executed without console display.

You can also more precisely customize the way the results are displayed in the Scilab console by using the command exec which, according to the chosen option, leaves more or less space in the console between the commands and their results:

```
    0.
-->exec('testscript.sce',-1) // no echo, except for "disp"

  x=
```