

Input/Output of data and file manipulation in *SMath Studio*

Prepared by Gilberto E. Urroz, September 2009

SMath Studio provides functions *wfile*, *rfile*, and *dfile* for output into a file, input from a file, and deleting an existing file. To illustrate the use of these functions we could start by writing some data to a file and exploring the contents of the file. Before we do that, however, I want to address an issue related to the location of the *SMath Studio* installation folder.

The location of the *SMath Studio* installation folder

When I first attempted this after installing *SMath Studio* in my *Windows Vista Ultimate 64-bit* machine I kept getting a message indicating that access to the file location was not permitted. This was so because *SMath Studio*, by default, sends output to and reads input from the folder */*SMath Studio Installation Folder*/user/*. In my computer, the **SMath Studio Installation Folder** corresponds to the directory *C:\Program Files(x86)\SMath\SMath Studio* and it requires *Administrator* access to modify its contents. To be able to write data from and read data into *SMath Studio*, therefore, I copied the folder *C:\Program Files(x86)\SMath* to a different location. In my case the location chosen was:

C:\Users\Gilberto E. Urroz\Documents\NUMERICAL_APPLICATIONS\SMath

In your computer it could be any location where you don't need *Administrator* rights to modify content. I then proceeded to create the folder *user*, where the data files would reside. Thus, the full address of this new folder, in my case, is:

C:\Users\Gilberto E. Urroz\Documents\NUMERICAL_APPLICATIONS\SMath\SMath Studio\user

To facilitate access to the program, at this point, I created a new shortcut to *SMath Studio* in my desktop. Now we proceed to illustrate the use of the *wfile* command.

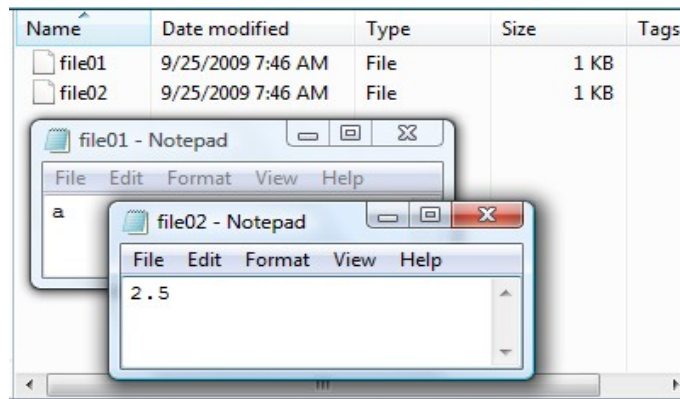
Writing data to a file

You can use function *wfile* to write a single item of data into a file. Consider the following examples¹:

<code>a:=2.5</code>	
<code>wfile(a, file01)=1</code>	Stores the letter 'a' in file "file01"
<code>wfile(2.5, file02)=1</code>	Stores the value "2.5" in file "file02"

You can open the files using, for example, *Notepad* in a *Windows* system to check the contents of the file:

¹ I recommend you use only lowercase letters in the filenames used to store data. I had difficulties getting *SMath Studio* to read data out of files with uppercase letters in their names.

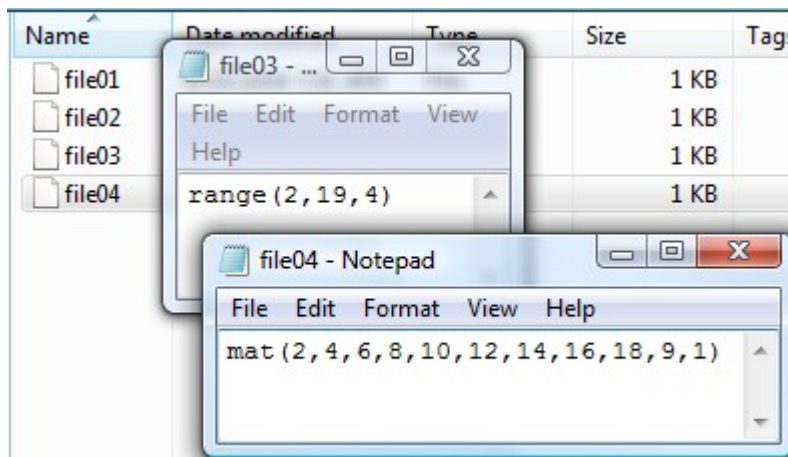


Thus, you'll need to use the numerical result that you want to write out to a file as the first argument of *wfile*. Using the reference to a variable will only write the variable name. The second argument to *wfile* is the name of the file, entered without quotes. The file name can only use letters and numbers. No other characters are allowed in the name of a file.

You can store a range or the resulting vector to a file, e.g.,

```
Two ways to write a range to a file:
(1) As a range                                (2) As a vector
s:=2, 4 .. 19
wfile(2, 4 .. 19, file03)=1
s = [ 2
      4
      6
      8
     10
     12
     14
     16
     18 ]
wfile(s, file04)=1
```

The contents of the corresponding files are shown below:



Regarding the notation $mat(2,4,6,10,12,14,16,18,9,1)$ written in file *file04*: this specification contains the 9 elements of the vector shown as *s*, above, plus the number of rows (9) and columns (1) defining the vector. Consider, for example, the writing of matrices to files. In the following two examples, the two matrices have 12 elements, each, but arranged differently. Matrix A has 3 rows and 4 columns, while matrix B has 3 rows and four columns. They get stored into files *file05* and *file06*, respectively, as shown below:

```

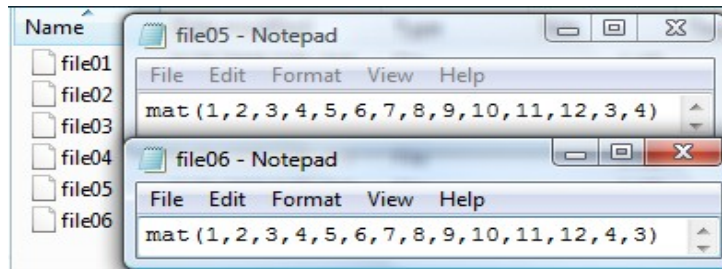
Writing a matrix to a file:

A:=  $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix}$       wfile  $\left( \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix}, file05 \right) = 1$ 

B:=  $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}$       wfile  $\left( \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}, file06 \right) = 1$ 

```

Next, we show the contents of the two files. Notice the specification of rows and columns as the last two values in the list of data:



Here is another example of using function *wfile*:

```

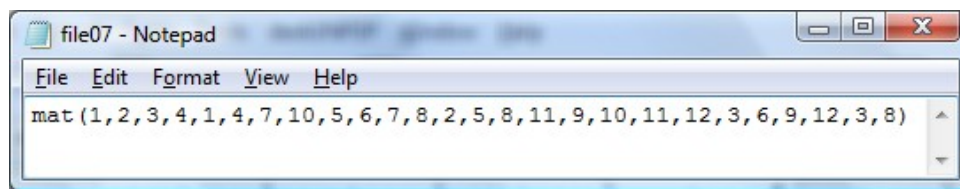
Another example of writing a matrix:

C:=augment(A, B^T)      C=  $\begin{pmatrix} 1 & 2 & 3 & 4 & 1 & 4 & 7 & 10 \\ 5 & 6 & 7 & 8 & 2 & 5 & 8 & 11 \\ 9 & 10 & 11 & 12 & 3 & 6 & 9 & 12 \end{pmatrix}$ 

wfile  $\left( \begin{pmatrix} 1 & 2 & 3 & 4 & 1 & 4 & 7 & 10 \\ 5 & 6 & 7 & 8 & 2 & 5 & 8 & 11 \\ 9 & 10 & 11 & 12 & 3 & 6 & 9 & 12 \end{pmatrix}, file07 \right) = 1$ 

```

The contents of the file are:



Reading data from a file

To read data from a file use function *rfile*. The following examples use the files we created above (i.e., *file01*, *file02*, ..., *file07*) to illustrate the use of function *rfile*. The result from reading data can be simply shown by using an equal sign(=), or assigned to a variable by using the assignment operator (:=). Try the following examples:

```
Reading files: (1) Showing result only:
rfile(file02)=2.5      Use = after command
rfile(file02)→ $\frac{5}{2}$       Use Ctrl+. (symbolic)
Reading files: (2) Assigning to a variable:
a:=rfile(file02)      a=2.5      a→ $\frac{5}{2}$ 
```

Additional examples of reading from a file are shown below:

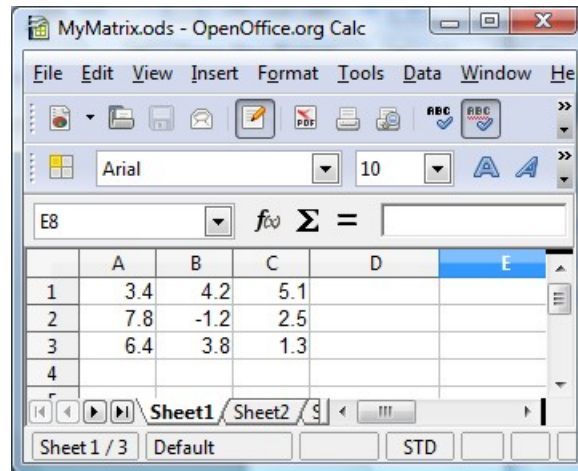
```
A:=rfile(file06)      A= $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}$       B:=rfile(file05)      B= $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix}$ 
rfile(file07)= $\begin{pmatrix} 1 & 2 & 3 & 4 & 1 & 4 & 7 & 10 \\ 5 & 6 & 7 & 8 & 2 & 5 & 8 & 11 \\ 9 & 10 & 11 & 12 & 3 & 6 & 9 & 12 \end{pmatrix}$       rfile(file03)= $\begin{pmatrix} 2 \\ 4 \\ 6 \\ 8 \\ 10 \\ 12 \\ 14 \\ 16 \\ 18 \end{pmatrix}$ 
```

If you attempt to read from a non-existing file you'll get an error message:

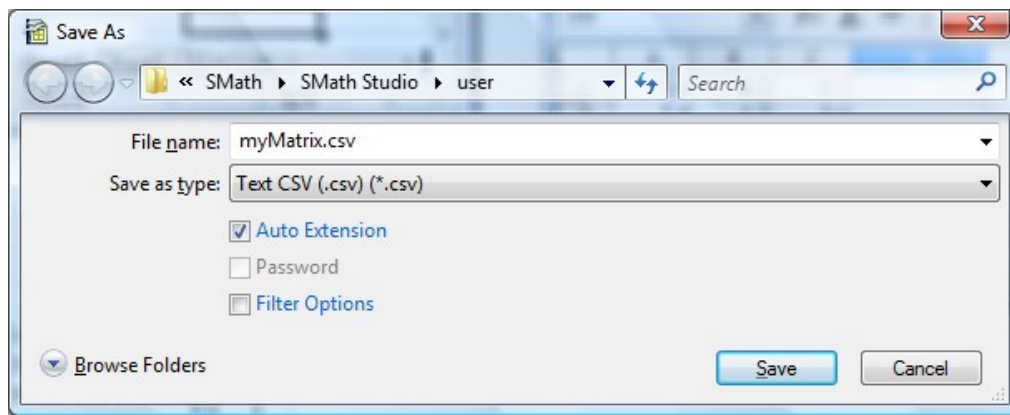
```
rfile(file08)=  
Could not find file 'C:\Users\Gilberto  
E. Urroz\Documents\NUMERICAL_APPLICATIONS\SMath\SMath  
Studio\user\file08'.
```

Copying a matrix from a spreadsheet

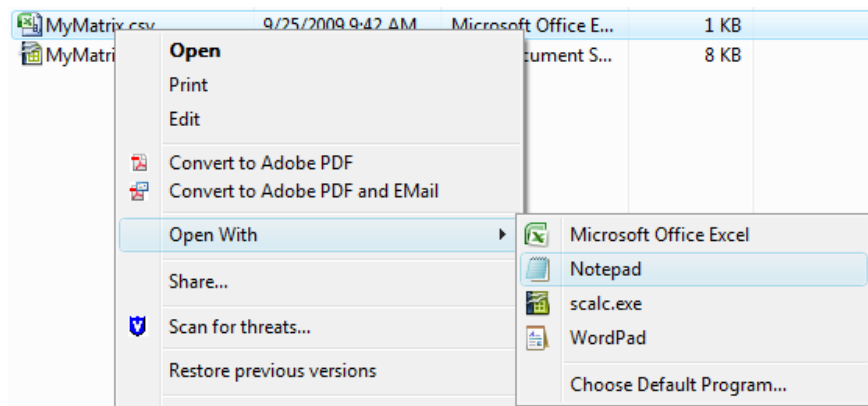
Suppose you have the following 3x3 matrix (only the numbers, no labels, etc.) in an OpenOffice.org Calc spreadsheet named *myMatrix.ods*:



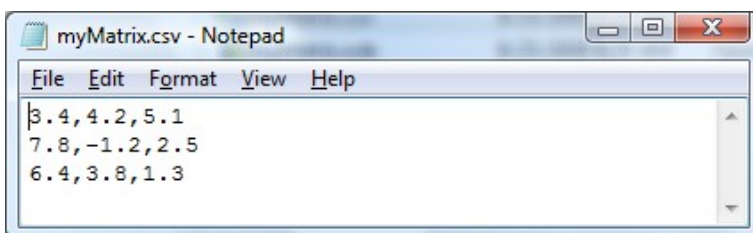
Save the file in *csv* (comma-separated values) format into your *user* folder in your *SMath Studio* installation:



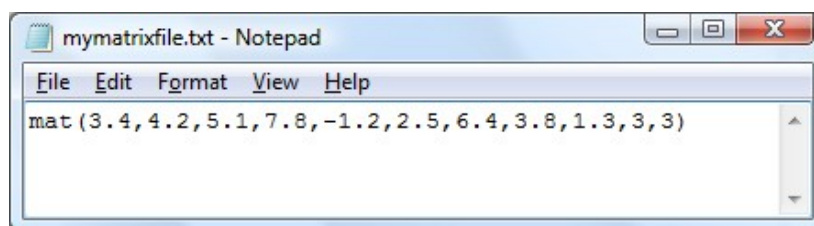
Then, close the newly saved file *myMatrix.csv*, and open it using *Notepad* (right-click on the icon, and select *Open with ...*):



to see the contents of the file:



Copy the file contents into another text file, let's call it *mymatrixfile*², and edit it to read:



Thus, the editing consisted in:

1. adding “mat(“ at the beginning of the line
2. placing all the data into a single line
3. replacing the missing commas (between 5.1 and 7.8, and between 2.5 and 6.4)
4. adding “3,3)” at the end of the line.

Save this file and then try the command:

```
rfile(mymatrixfile) =  $\begin{pmatrix} 3.4 & 4.2 & 5.1 \\ 7.8 & -1.2 & 2.5 \\ 6.4 & 3.8 & 1.3 \end{pmatrix}$ 
```

Of course, to get this simple 3x3 matrix into *SMath Studio* it will be easier to type it in into the worksheet. The procedure illustrated above would be more practical for very large spreadsheet data sets.

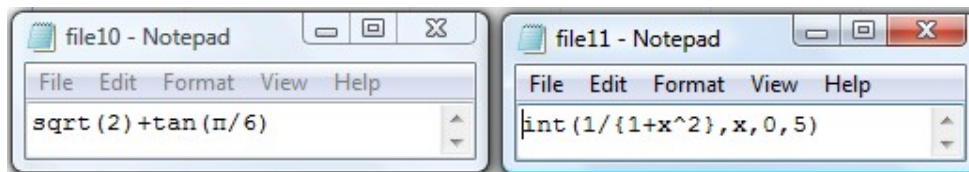
2 A few notes: (1) Use only lowercase letters in the name of the file. I couldn't get *SMath Studio* to recognize filenames that include uppercase letters. (2) Make sure the filename to be read has no extension, therefore, the file should be named, in this case, *mymatrixfile*, and not *mymatrixfile.txt* or *mymatrixfile.dat*, etc. (3) Do not try to simply use the same *.csv* file without an extension as your input file to *SMath Studio*. It won't work. You need to create a new file.

Writing and reading symbolic expressions

The following examples illustrate the writing of expressions to a file, and the reading of the same expressions from files.

Writing and reading symbolic expressions:	
Example 1:	Example 2:
Writing to a file:	Writing to a file:
$wfile\left(\sqrt{2} + \tan\left(\frac{\pi}{6}\right), \text{file10}\right) = 1$	$wfile\left(\int_0^5 \frac{1}{1+x^2} dx, \text{file11}\right) = 1$
Two ways of reading from the file:	Two ways of reading from the file:
$rfile(\text{file10}) = 1.9916$	$rfile(\text{file11}) = 1.3734$
$rfile(\text{file10}) \rightarrow \frac{\sqrt{2} \cdot \sqrt{3} + 1}{\sqrt{3}}$	$rfile(\text{file11}) \rightarrow \frac{68670038346567}{5000000000000}$

The contents of the files are shown below:



The following example shows writing and reading of a matrix of symbolic expressions:

Writing a matrix of symbolic expressions to a file:

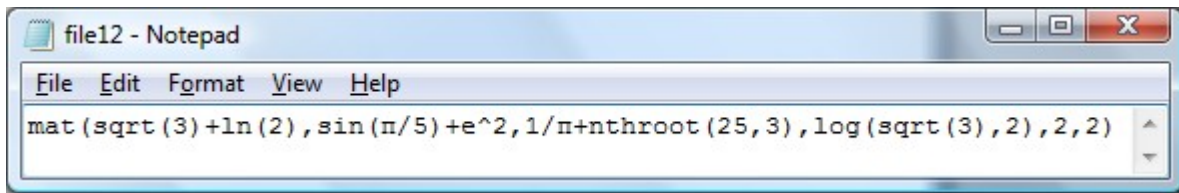
$$wfile\left(\begin{pmatrix} \sqrt{3} + \ln(2) & \sin\left(\frac{\pi}{5}\right) + e^2 \\ \frac{1}{\pi} + \sqrt[3]{25} & \log_2(\sqrt{3}) \end{pmatrix}, \text{file12}\right) = 1$$

Reading the file with a matrix of symbolic expressions:

$$rfile(\text{file12}) = \begin{pmatrix} 2.4252 & 7.9768 \\ 3.2423 & 0.7925 \end{pmatrix}$$

$$rfile(\text{file12}) \rightarrow \begin{pmatrix} \sqrt{3} + \ln(2) & e^2 + \sin\left(\frac{\pi}{5}\right) \\ \frac{1 + \sqrt[3]{25} \cdot \pi}{\pi} & \frac{\ln(3)}{2 \cdot \ln(2)} \end{pmatrix} \quad +$$

The content of the file is shown below:



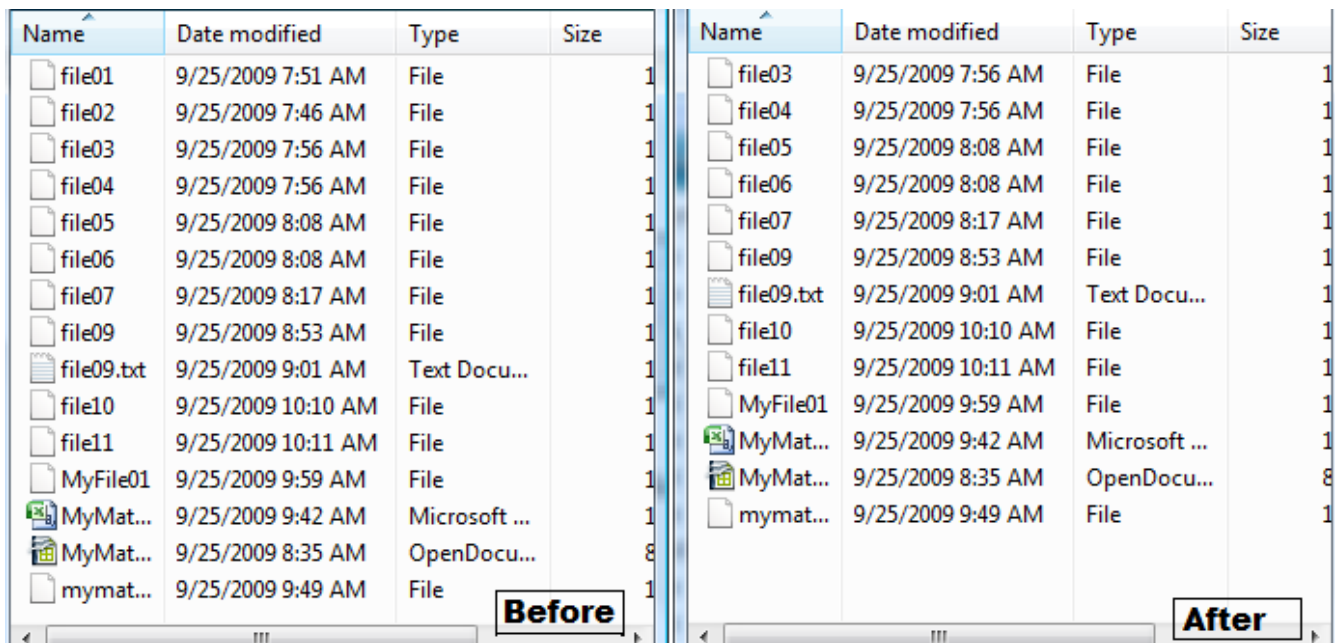
Deleting files

Use function *dfile(filename)* to delete files in the folder */*SMath Studio Installation Folder*/user*.

Examples:

```
Deleting file:
dfile(file01)=1    dfile(file02)=0
```

The directory */*SMath Studio Installation Folder*/user* is shown below before and after the execution of the *dfile* commands shown above:



Application of functions *wfile*, *rfile*, and *dfile* could be useful in programming if there is a need to store data temporarily in a file.