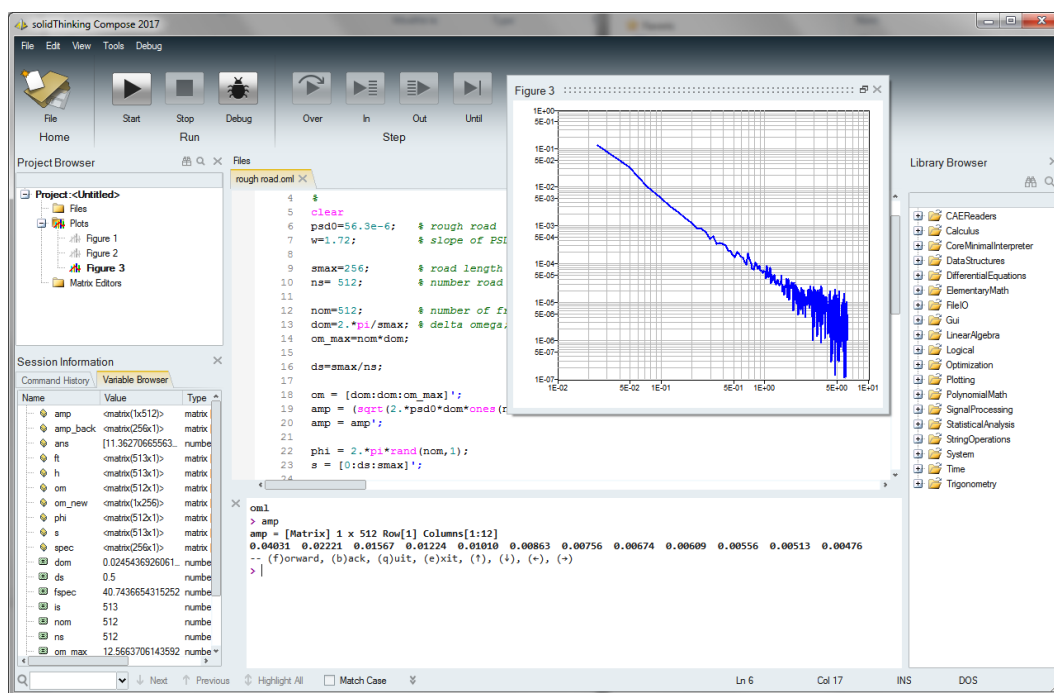


Compose 2017.1 Release Notes

INTRODUCTION

solidThinking Compose is a high level, matrix-based numerical computing language as well as an interactive and unified programming environment for all types of math. Whether you're looking to solve matrix analysis, differential equations, perform signal analysis, or robustly study control design, **Compose** offers you a modern, comprehensive set of tools to enable rapid development and also a powerful engine and an interactive debugging environment for streamlined troubleshooting.



solidThinking Compose 2017.1 offers these exciting features:

- OML, a High-level matrix-based interpreted language for numerical computing
- Integrated development environment for authoring and debugging all types of math, including multi-language support
- Extensive math libraries
- Built-in connectivity to pre/post-process engineering and Computer Aided Engineering (CAE) data
- Interactive command line interface
- Rich plotting, with floating plots
- Multi-dimensional matrices support
- Linux support
- GUI creation commands
- Python language support, and a bidirectional bridge between OML and Python (new in 2017.1)

Learn more at solidThinking.com/Compose | 1 |

PLATFORM SUPPORT

Platform		
OS	Version	Architecture
Windows	10/8.1/7	x86_64
Linux	RHEL and CentOS 6.6 and 7.2 SLES 12 SP1	x86_64

MATH & SCRIPTING SUPPORT

Open Matrix Language (OML)	<ul style="list-style-type: none"> • OML files can now be encrypted for IP protection • Underscore is now supported in variable names
Functions	<ul style="list-style-type: none"> • <code>fminsearch</code> function • Rainflow counting is implemented (<code>rainflow</code>) • Added flip commands (<code>flip</code>, <code>fliplr</code>, <code>flipud</code>) • <code>invfreqs</code>, <code>infreqz</code> • <code>upsample</code>, <code>dowsample</code> • <code>str2double</code>
Python Support	<ul style="list-style-type: none"> • Compose 2017.1 offers a two-way bridge between OML and Python. OML scripts can be called from Python and vice-versa. Similarly, variables can be exchanged between OML and Python. • Support of Python language in Compose IDE is improved (History)
Plotting	<ul style="list-style-type: none"> • Plots now have an independent autofit for the X and Y axis
GUI Creation	<ul style="list-style-type: none"> • New and improved commands to create user interfaces and interactive dialogs (<code>uitable</code>, <code>uiputfile</code>, <code>uigetfile</code>)

GENERAL / USER EXPERIENCE

ReadVectorBuilder	<ul style="list-style-type: none"> • The ReadVectorBuilder utility covers the extended capabilities of the <code>readmultivectors</code> command
Miscellaneous	<ul style="list-style-type: none"> • Various documentation updates
HyperWorks Integration	<ul style="list-style-type: none"> • Starting with Compose 2017.1, a strong connection with HyperWorks Desktop products has been initiated. OML functions can be registered from Compose and can be reused in HyperGraph or in the HyperMesh Matrix Browser. This is an experimental feature in release 2017.1. In following releases, this connection will be extended and also made available to other products, such as HyperStudy.

The following issues, and more, have been resolved for **solidThinking Compose 2017.1**:

Add new file in Project Browser will lose the file extension
UIControl button isn't hooked with callback correctly when plotting is set to floating in preferences
path() will return a null search path
assert(false == 1) fails
cell extraction can lead to a crash in some cases
copyfile() issues
Function inside struct with multiple outputs
Highlight issue in Compose Editor
readmultvectors indexing issue
Subplot with only one argument is not supported
regexp arguments are swapped
Cut in Command history will close Compose on Linux
Cannot close a figure with a callback
mat2str() cannot specify precision in Linux
OML editor incorrectly indenting some constructs
Adding new files in the Project Browser on Linux freezes Compose
Memory issue with optimization
ReadVectorBuilder() does not load a DATX file properly
Variables will be treated as 0 when exporting a small value from OML to Python
Using complex as input in fminsearch() can result in a crash
Issue with Dongle license support
Wrong results with the function "interp1"
sscanf issue on mixed types
Editor does not properly display single quotes pair inside a cell
Performance issue when trying to change a symbol in a 3D plotting
Using complex as input data inuitable crashes Compose
Inconsistent handling of filtered lists with the CAE Reader commands getXXXlist
Crash when accessing pointer to a redefined function
Call exit in ipython window does not close Compose
while loop on a single line is not supported
Cannot add item in the Watch Window when debugging
Cannot print a multi-dimensional matrix using printf()
Font preferences do not apply to the Python Editor
printf() command produces extra lines when run in batch mode using Compose_Batch.bat
subsasgn() with cell index does not return the correct result
setfield() does not work in some cases
Pagination does not work in the Compose Console on Linux
copyfile always returns success code 1
assert on an empty matrix should return an error
Using an endless loop to cross-evaluate python and OML scripts crashes Compose
Bode plot formatting issues

Incorrect behavior with <code>setfield</code>
<code>omlfilename</code> does not work if the file is being edited/not saved/read-only
Color issue when using <code>'</code> to transpose a matrix
Extend <code>uigetfile</code> to support cell array inputs which can specify a list of file filters
Miss the possibility to have a pre-selected extension when using <code>uiputfile</code>
Function handle doesn't work in a struct
The Variable Browser is not updated after running a callback function
Print <code>nan</code> and <code>inf</code> using <code>printf</code> return wrong result
<code>exist(name)</code> wrong result when the name is a folder on Linux
Error message for <code>fscanf</code> is incomplete
<code>Compose_Console -e</code> doesn't work properly on Linux
When using <code>uiputfile</code> , the file browser does not display the existing files of the directory
Format should change back when there are no complex values in a matrix
<code>subsasgn()</code> does not return the correct result
Comma in the <code>switch</code> statement leads to a syntax error
<code>toascii(char(M))</code> should return the same <code>M</code> input
<code>readmultivectors</code> is not reporting out of bound message for time index
<code>system('pwd')</code> returns different results in Console mode
<code>besself</code> should generate an error when the frequency band (ω_p) is complex
Extraction of a non-existing field from a struct should return an error
<code>rainflow()</code> should raise an error when <code>minrange</code> is larger than <code>maxrange</code>
<code>rainflow()</code> should raise an error when <code>minrange/maxrange</code> is NaN
<code>unique</code> command does not work for multi-dimensional matrices
Incorrect result with <code>exist</code> applied to a function name
<code>~exist</code> should throw an error if the input is not a string