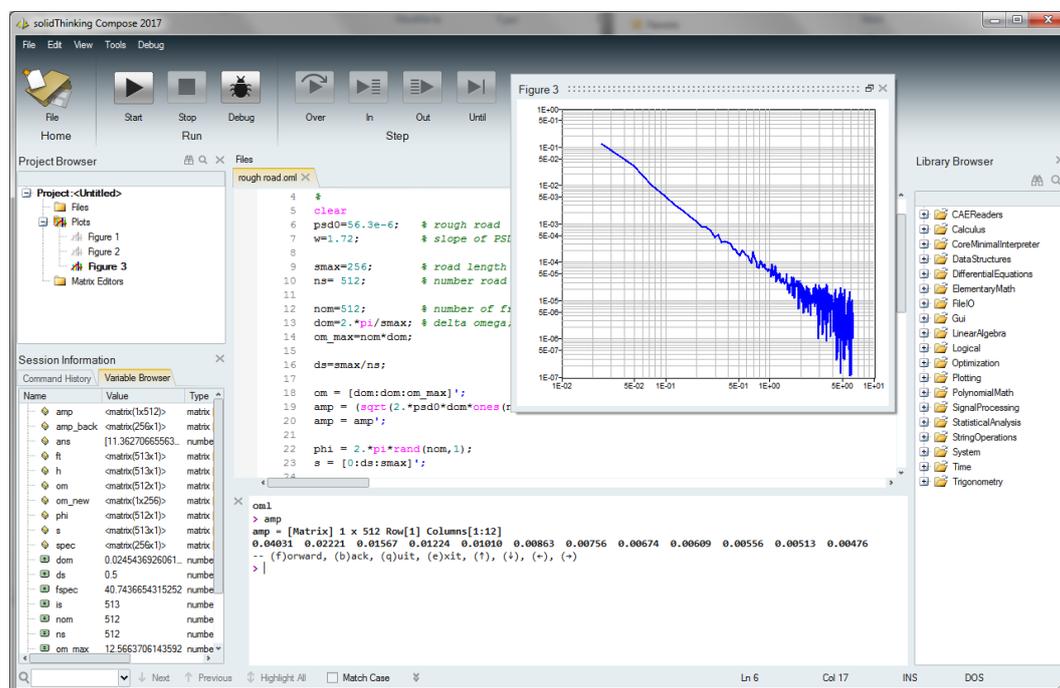


Compose 2017.2 Release Notes

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 2017.2 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"> • Improvements to the matrix math performance • Improvements to the FFT computation performance • Variable names can start with an underscore
Functions	<ul style="list-style-type: none"> • <code>importdata</code> function • <code>xlsread</code>, <code>xlswrite</code> functions to handle spreadsheets • <code>regexprep</code> • <code>circshift</code> • <code>checksyntax</code> function to test the syntax of a statement without executing it
Plotting	<ul style="list-style-type: none"> • <code>minorgrid</code> • Support for multiple axes on a figure
GUI Creation	<ul style="list-style-type: none"> • New command for user interfaces and interactive dialogs: <code>waitbar</code>
H2O	<ul style="list-style-type: none"> • A utility tool to convert scripts from HyperMath HML language to Compose OML language is provided in the installation package. Note that this is an <i>assistant</i> to help in conversion and not all of the HML language is covered.

GENERAL / USER EXPERIENCE

Error Handling	<ul style="list-style-type: none"> • Error messages in command window are hyperlinks and allow for jumping to the error line
Miscellaneous	<ul style="list-style-type: none"> • Various documentation updates • Support for Japanese localization of the user interface
HyperWorks Integration	<ul style="list-style-type: none"> • A strong connection with HyperWorks Desktop products has been implemented. OML functions can be registered from Compose and can be reused in HyperGraph, HyperStudy, MotionView or in the HyperMesh Matrix Browser. This is an experimental feature in release 2017.2

The following issues, and more, have been resolved for **solidThinking Compose 2017.2**:

Issue with Nested functions not recognized
Plot window refresh issue when doing a 3D plot
Insertion in cell array using end index doesn't work correctly
Compose_Console.bat -f option does not work
type() return unexpected error when input is filepath
Support multiple axes on a figure
mat2cell([1+i 2+i]) crash Compose
Legend values issue on plots
Grids on polar plots
Error while calling a function through a structure
Recursive nested functions can't access outer function's variables
Align Y axis for subplots
sprintf do not work properly when input is an empty string
Filebrowser path issue
Indexing issue beyond matrix limits
Colors of the curve, the text in the legend, and the curve sample are different
Under debugger quit and exit commands do not close the application
Close Compose when debugging a script will freeze compose (Linux platform only)
Compose crash after sliding middle mouse on the plot window
quit and exit commands don't work properly
mkdir do not work properly in linux
struct extraction issue
Compose crash when assign a value to an empty matrix
Slow response in plotting
The returned value of length is not correct on empty structs
uinputfile do not work on linux platform
error when using encrypted files (.omc files)
Issue when register a function with keyword 'function' in the function name
importdata() do not work properly
Auto-indent is incorrect with elseif
Minor grid display not working

Learn more at solidThinking.com/Compose | 3 |

<code>struct ([])</code> should return an empty struct
Executing <code>oml</code> in batch mode returns warning message
[Python] interactive mode in <code>matplotlib</code> does not work
Syntax highlighting should be applied to python keywords <code>True False</code>
<code>eval</code> returns error in nested function
Python iterator and generator are not recognized by debugger in watch window
Registering a commented function should not be supported
<code>exist()</code> does not return to desire result but an error in below case
<code>readvectorbuilder</code> cannot generate <code>readvector</code> command in some case
<code>type()</code> does not work properly in below case
<code>cat()</code> does not return to desire result in below case
Crash with empty cell in <code>setfield</code>
<code>strvcat</code> should ignore null strings
Built in function <code>ls</code> is improved to use system level command
If a change to the active directory occurs outside of Compose, the file list doesn't update