

# NL5 User's Manual

Ver.3.12



## VERSION

NL5 User's Manual version 3.12.38, 04/24/2024

The latest versions of NL5 documents can be found at [sidelinesoft.com/nl5](https://sidelinesoft.com/nl5).

## LIMITED LIABILITY

NL5, together with all accompanying materials, is provided on a “as is” basis, without warranty of any kind. The author makes no warranty, either expressed, implied, or stationery, including but not limited to any implied warranties of merchantability or fitness for any purpose. In no event will the author be liable to anyone for direct, incidental or consequential damages or losses arising from use or inability to use NL5.

## COPYRIGHTS

© 2024, A.Smirnov. The program and User's Manual are copyrighted. No portion of this Manual can be translated or reproduced for commercial purposes without the express written permission from the copyright holder. On publication of results obtained from use of NL5 citation is appreciated.

---

“**Smith**” is a registered trademark of Analog Instruments Company, New Providence, NJ. **Microsoft**, **Windows**, and **Microsoft Visual C++** are registered trademarks of Microsoft Corporation. **MATLAB** is a registered trademark of The MathWorks, Inc. **PYTHON** is a registered trademark of the Python Software Foundation. **National Instruments** is a registered trademark of National Instruments Corporation. Built with **Indy** ([www.indyproject.com](http://www.indyproject.com)). **Verilog** is a registered trademark of Cadence Design Systems. **Xilinx** and **Vivado** are registered trademarks of Xilinx.

# Table of Contents

<b>I. Introduction .....</b>	<b>6</b>
Install and run NL5 .....	7
NL5 License .....	7
NL5 file types.....	8
NL5 Help.....	8
NL5 preferences .....	9
NL5 DLL.....	11
Create and Simulate Your First Schematic.....	12
<b>II. User Interface.....</b>	<b>15</b>
Application.....	16
Navigation bar.....	17
Settings window.....	18
Context menus.....	19
Shortcuts .....	20
Mouse .....	21
Image Copy, Save, and Print .....	22
Data format.....	23
Numbers .....	23
Names .....	25
Operators.....	28
Functions.....	28
Expressions .....	29
C language .....	30
Script .....	31
Script syntax.....	31
Script commands.....	31
<b>III. Schematic .....</b>	<b>33</b>
Edit schematic .....	34
Cursor.....	34
Wire.....	34
Ground .....	34
Connection .....	35
Component.....	35
Label .....	36
Drawing.....	37
Text .....	37

Variables .....	37
Zoom and scrolling .....	37
Selection.....	38
General commands and shortcuts .....	38
<b>Open and save schematic.....</b>	<b>39</b>
Open .....	39
Save.....	39
<b>Components.....</b>	<b>40</b>
<b>Working with special component types and models.....</b>	<b>43</b>
Subcircuit model .....	43
PWL model .....	46
List model (voltage and current source) .....	50
List model (switch and logic generator) .....	51
Table model .....	52
2D Table model.....	53
C-code component .....	54
DLL component .....	58
Roots model .....	63
Attachments .....	64
NL5 components (*.nlc) .....	64
Customized component.....	65
Encrypted component .....	65
<b>Schematic variables .....</b>	<b>66</b>
<b>Schematic sheets.....</b>	<b>67</b>
<b>Schematic groups .....</b>	<b>68</b>
<b>Check schematic.....</b>	<b>69</b>
<b>Schematic Settings and tools.....</b>	<b>70</b>
<b>IV. Transient .....</b>	<b>72</b>
<b>Algorithm.....</b>	<b>73</b>
<b>Running transient .....</b>	<b>75</b>
<b>Transient window.....</b>	<b>76</b>
Status bar.....	80
Cursors .....	80
Labels.....	80
Legend.....	81
Annotations .....	82
Text .....	83
Table .....	83
Storage .....	84
<b>Transient settings.....</b>	<b>85</b>
Simulation.....	85
Add traces .....	88

Format traces.....	94
Screen.....	98
Cursors .....	100
Table .....	101
Storage .....	103
<b>More commands (toolbar).....</b>	<b>104</b>
<b>Transient Tools.....</b>	<b>108</b>
XY diagram.....	108
Eye diagram .....	109
Amplitude histogram .....	110
FFT.....	113
Transient histogram .....	117
Power .....	119
DC sweep.....	121
<b>V. AC analysis .....</b>	<b>123</b>
<b>Algorithm.....</b>	<b>124</b>
<b>Running AC analysis .....</b>	<b>125</b>
<b>AC window .....</b>	<b>126</b>
Status bar.....	129
Cursors .....	129
Legend.....	129
Annotations.....	130
Text .....	131
Table .....	131
Storage .....	132
<b>AC settings.....</b>	<b>133</b>
Simulation.....	133
Add traces .....	136
Format traces.....	140
Screen.....	142
Cursors .....	143
Table .....	144
Storage .....	146
<b>More commands (toolbar).....</b>	<b>147</b>
<b>AC Tools .....</b>	<b>151</b>
Smith chart.....	151
Nyquist plot.....	153
AC histogram.....	154
<b>VI. Tools .....</b>	<b>156</b>
<b>Script .....</b>	<b>157</b>
<b>Console.....</b>	<b>158</b>
<b>Sweep.....</b>	<b>159</b>

---

<b>Optimization</b> .....	<b>161</b>
<b>HTTP link</b> .....	<b>162</b>
NL5-MATLAB link example .....	166
<b>Scope</b> .....	<b>168</b>
Working with oscilloscope .....	170
Working with waveform generator .....	174
Working with segmented memory .....	178

# I. Introduction

## Install and run NL5

The only file required for NL5 operation is executable `n15.exe`. It can be placed into any directory. There could be several copies of `n15.exe` in different directories on one computer.

It is recommended to run `n15.exe` with administrator privileges, to have full access to the **Registry**. To set up administrator mode, right-click on the NL5 icon, select **Properties**, select **Compatibility** tab, check **Run this program as an administrator**.

The following switches and parameters can be used.

### Switches:


```
-http or /http      : start HTTP link
-scope or /scope    : open Scope tool window
```

### Parameters:

```
>n15.exe rc.n15      : open schematic rc.n15
>n15.exe rc.n1t      : load transient data from rc.n1t
>n15.exe rc.n1f      : load AC data from rc.n1f
>n15.exe pref.nlp    : load preferences from pref.nlp
>n15.exe script.txt  : open and run script from script.txt
```

## NL5 License

Without a license, NL5 operates as a **Demo version**. Demo version has all full function features available, however the total number of components in the schematic is limited to **20**. As an exception, NL5 Demo version can simulate demo schematic with an unlimited number of components, if that schematic was created with a special License option. Ask for that option if you want to distribute big schematic files to someone who does not have a license.

Different types of NL5 License are available at NL5 website. The License file `n15.n11`  should be placed in the same directory as an executable file `n15.exe`.

NL5 License also supports **NL5 DLL**.

Use **NL5 lite** for simulation with unlimited components without a license.




## NL5 file types

The following file extensions are used by NL5:

Extension	Description
n15	Schematic
n15~	Schematic backup
nlp	Preferences
nlc	NL5 component
nlt	Transient data (binary)
nlf	AC data (binary)
nll	License

Information about NL5 file types and icons is stored in the registry. Please run NL5 as administrator to set up this information properly.

## NL5 Help


NL5 help file `n15.chm`  is not required, but if used, it should be placed in the same directory as `n15.exe`. The file contains only reference information, such as descriptions of operators, functions, commands, components, and models. For detailed information refer to the **NL5 User's Manual** and **NL5 User's Reference**.

For context Help, press **F1**, or click **Help** button , which is available in some windows.

If you cannot see the content of Help file, most likely the file is blocked. To unblock:

- Locate `n15.chm` file in the NL5 directory.
- Right-click the file, then click **Properties**.
- Select **General** tab.
- Click **Unblock**.
- If **Unblock** button is not visible, delete `n15.chm` file from the directory, copy it to NL5 directory again from `n15.zip` download package, and repeat this procedure.

## NL5 preferences

**Preferences**  are stored in the file `n15.n1p`, typically located in the same directory as `n15.exe`. NL5 reads preferences from the file at start-up and saves into the file every time **Apply** or **OK** button in the **Preferences** window, and on exit.



Please note that if `n15.exe` is located anywhere inside `C:\Program Files` directory, it may not be able to save preferences file `n15.n1p` due to Windows writing restrictions. In this case, the preferences file will be saved into the `C:\Users\user_name\Documents\n15` directory (or similar directory, depending on specific OS).

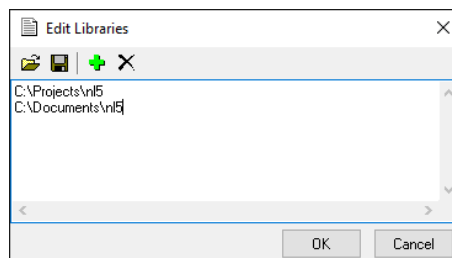
Preferences can also be saved in a custom preferences file with extension `*.n1p`, and then opened back from the file, so you can have different “profiles” and switch between them quickly.


To open **Preferences** window click  in the Main menu, or in some pop-up context menus.


Some useful **Preferences** settings:

### Application/Library.

**Library path (subcircuits, DLLs, and other files):** specify default directory paths list for files used by components. Click **Edit** , then enter library paths (one path per line) or click  to select from folders list.

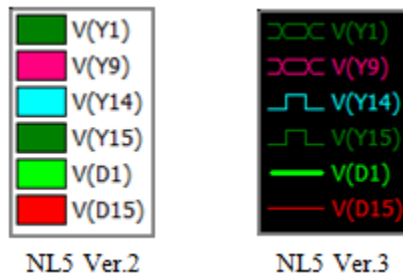


Library path could be full path, or path relative to the directory where schematic file is located. When a new schematic is created, library paths specified in Preferences will be copied into the schematic library list. To modify library paths list of a specific schematic, go to **Schematic Settings and tools** , **Library path** tab.

**Default directory for saved components:** specify default directory path for components saved into `*.n1c` file. Enter manually or click **Select directory**  to select from folders list.

**Document/Graphs/Legend.**

**Legend style.** Select **Ver.2** or **Ver.3** to set old legend view as in NL5 Ver.2, or a new NL5 Ver.3 view:



After a general legend view is selected, its specific features can be manually changed by selecting desired options.

## NL5 DLL

NL5 DLL is a dynamic-linked library available for **Windows** (`n15_dll.dll`), **Linux** (`n15_dll.so`), and **macOS x64 and arm64** (`n15_dll.dylib`). It performs transient and AC simulation of circuits created by NL5 Circuit Simulator, provides raw simulation data, allows modification of circuit parameters, adding data traces, and some other operations through DLL API functions. NL5 DLL can be used as an analog simulator which is started and controlled from other applications and tools (MATLAB, Python, custom C/C++ code), and as an analog co-simulation tool working with digital simulation tools, for example SystemVerilog simulators (through DPI interface).


Typically, the released **Version** and **Revision** of NL5 DLL is always the same as Version and Revision of NL5 Circuit Simulator (for example 3.12). This guarantees full compatibility in terms of algorithm, components, models, features, and performance. However, there is nothing wrong in using different Versions and Revisions of DLL and NL5.

Current **Build** of DLL (for example 3.12.59.23) can be different from NL5, due to possible fixes and modifications specific to NL5 or DLL.

Without a license, NL5 DLL operates as a **Demo version**. NL5 DLL Demo version has all full function features available, however the total number of components in the schematic is limited to **20**.

For unlimited number of components, NL5 DLL should use **NL5 License**.

## Create and Simulate Your First Schematic

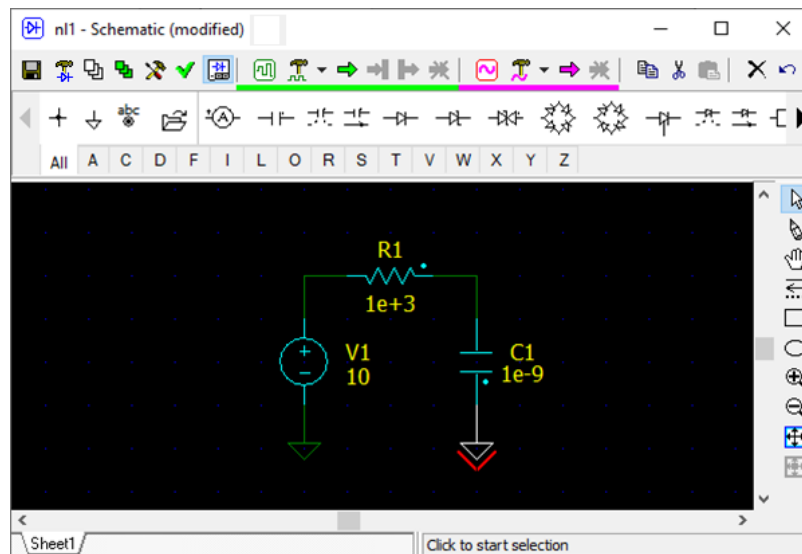
By default, a new schematic window should be opened automatically at start up. If not, click **New** toolbar button  in the main NL5 window.

### Enter schematic.


Entering and editing can be done using keyboard keys, mouse, or both. Here are step-by-step instructions how to enter a simple schematic using keyboard.

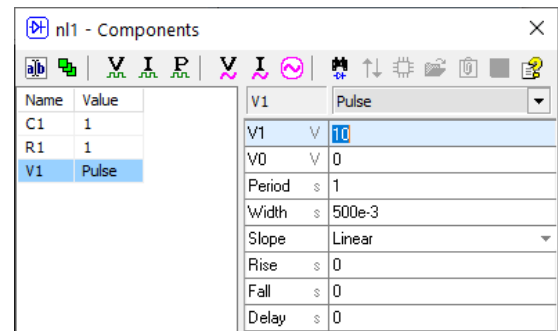
When started, a red cursor is located in the middle of the screen and is pointing to the right.

- Press **Space** to switch to drawing mode.
- Press **Arrow Down** several times to draw short wire downward.
- Press **V** key and then press **Enter** to place a voltage source.
- Press **G** key to place a ground. Now cursor is switched back to selection mode.
- Press **Arrow Up** several times to move cursor back to the starting point.
- Press **Arrow Right** to change direction; then press **Space** to switch to drawing mode.
- Press **Arrow Right** several times to draw a short horizontal wire.
- Press **R** key and then press **Enter** to place a resistor.
- Press **Arrow Right** several times again; then press **Arrow Down** several times.
- Press **C** key and then press **Enter** to place a capacitor.
- Press **G** key to place a ground. Schematic is ready.





### Edit component parameters.


- Double click on the voltage source **V1** on the schematic. A **Components** window will show up.
- Click on **V1** in the components list. Click  right to the model name (top-right), select **Pulse**.
- Click on the resistor **R1**, enter 1 in the parameter field **R**.
- Click on the capacitor **C1**, enter 1 in the parameter field **C**.

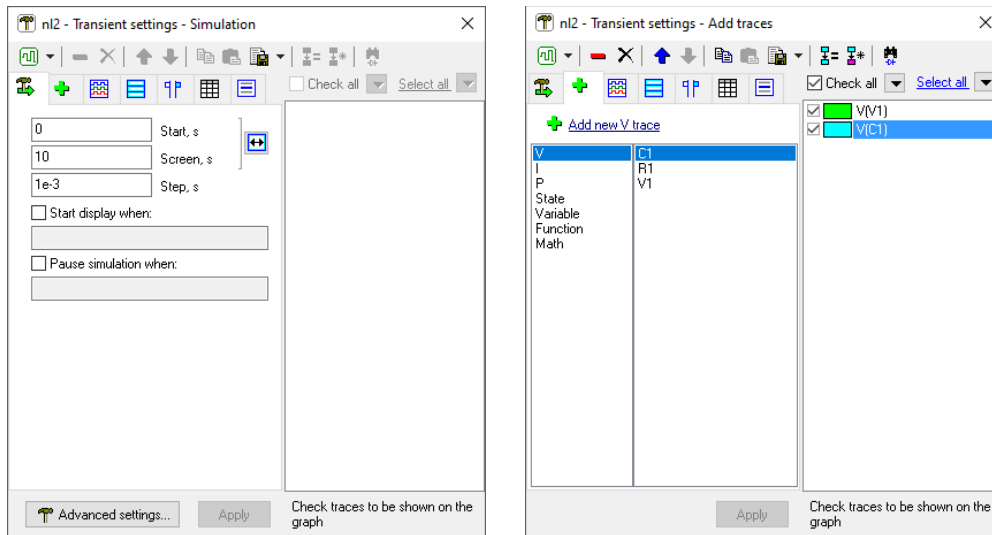


## View transient settings.

Click **Transient settings**  in the schematic window toolbar, select **Simulation**  to go to **Simulation** tab. You do not need to change anything here, but you can, if you wish.

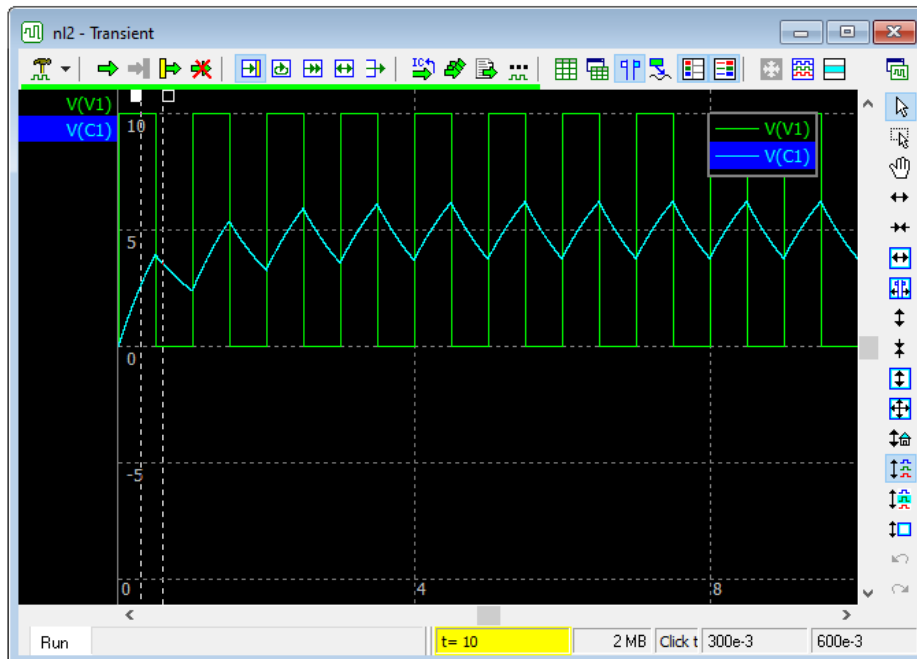
## Add transient traces.

Select **Add traces** tab , select **V** in the left window, then double-click on **V1** and **C1** in the components list. Voltage traces will be added to the trace list.





## Run transient.


Click **Start transient**  to run simulation:

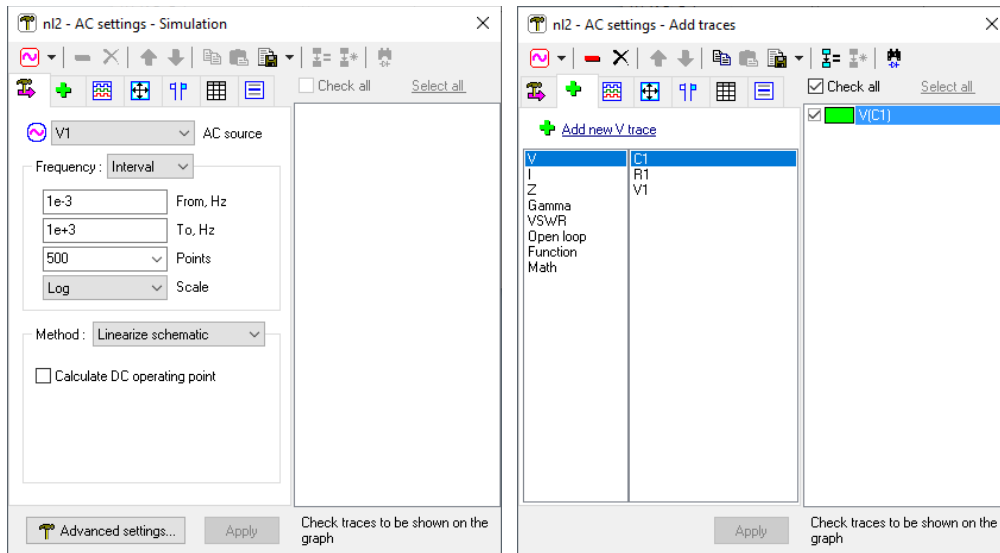


### View and edit AC settings.

Click **AC settings**  in the schematic window toolbar, select **Simulation**  to go to **Simulation** tab. Click on the **AC source** drop-down list, select **V1**.

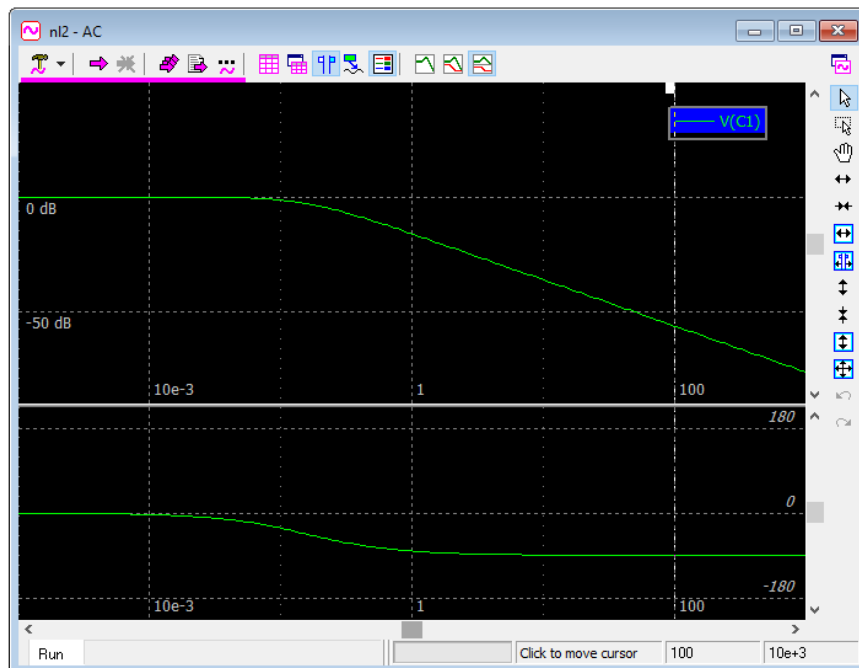
### Add AC traces.

Select **Add traces** tab , select **V** in the left window, then double-click on **C1** in the components list. AC voltage trace will be added to the traces list.



### Run AC.

Click **Start AC**  to run AC simulation:



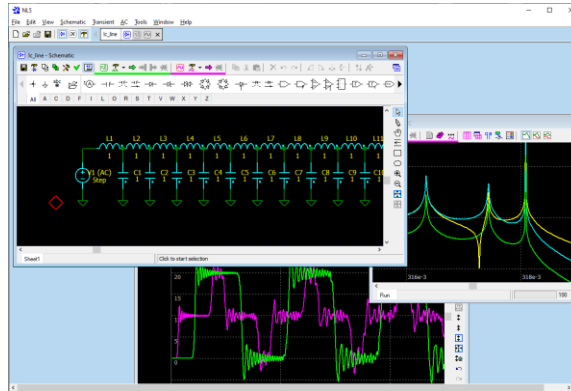
## II. User Interface



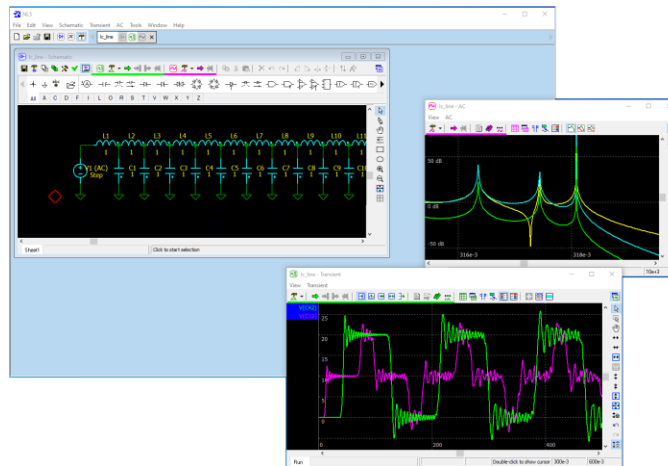
## Application

NL5 uses standard Windows multiple-document interface (**MDI**) with an important and useful addition: **floating windows**.

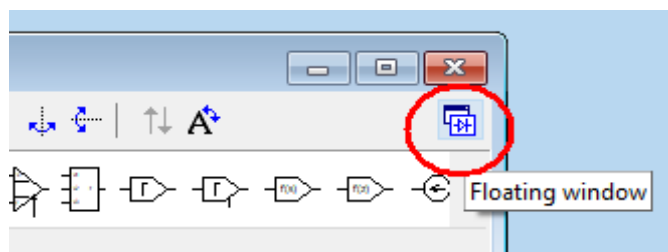
All schematic-related windows (Schematic, Transient, AC, Transient and AC tools) are first created in “MDI child” mode, and are located inside the “main” window:



After that, any window can be switched to a “floating” mode and moved outside the “main” window (for example, to a different monitor), maximized to full screen, etc.:

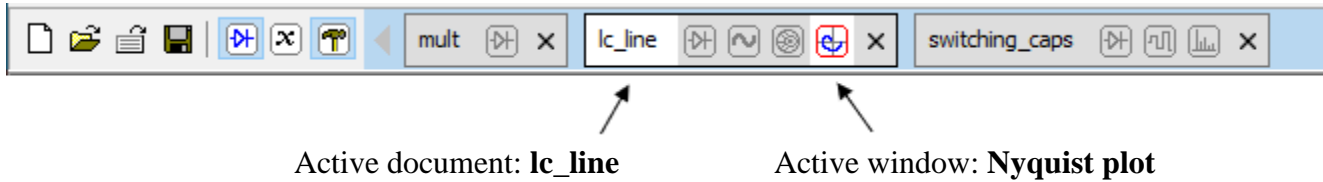



To turn “floating” mode on/off, use **View/Floating window** main menu command, **Floating window** context pop-up menu command (**Right-click** on the window), or **Floating window** button on the right end of window top toolbar:



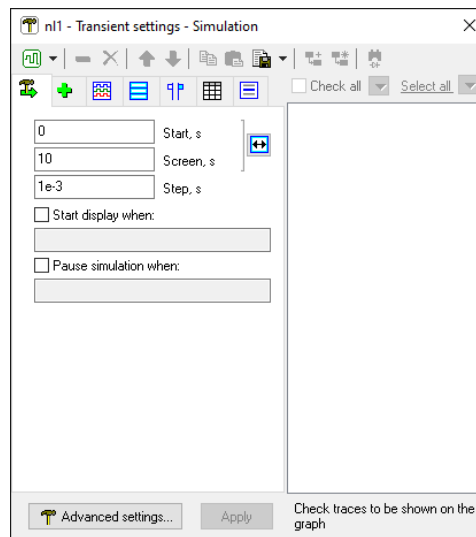
## Navigation bar

**Navigation bar** shows currently opened documents (schematics), with highlighted active document and window:



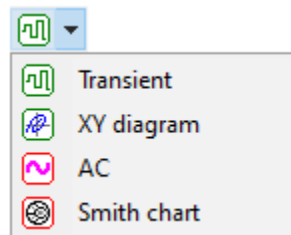
- Click on the document name to activate the document and bring all its opened windows to the top.
- Click on the window icon to activate the document and selected window.
- Right-click to see pop-up menu window with related commands.
- Click  to close the document.

## Settings window



**Settings window** shows settings for **Transient**, **AC**, or **Tool** window. **Settings window** always shows information for active document (schematic). The name of the active document is shown in the header of **Settings window**.

When **Transient**, **AC**, or **Tool** window is activated, **Settings window** will automatically show settings for that window. Also, you can click **Select settings** button (top left) and select any window from the list of currently opened windows for active document:



If numeric parameters in **Settings window** are modified, their background color will change to light-yellow:



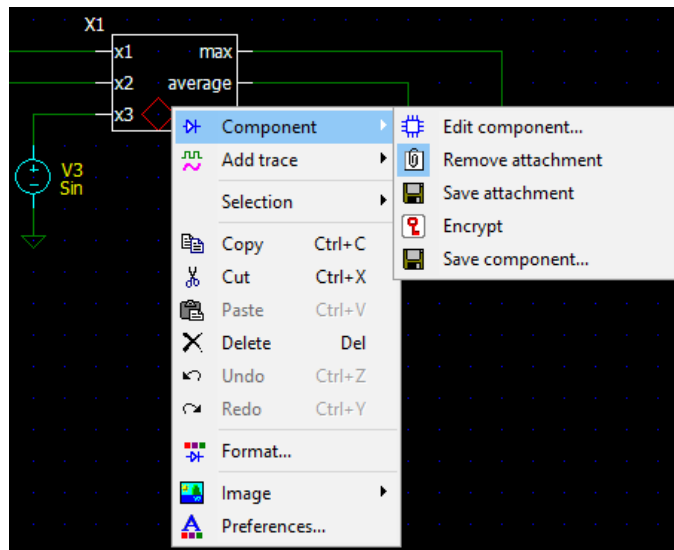
Press **Enter** or click **Apply** to accept changes. Changes of checkboxes and selection of drop-down boxes are accepted automatically.

**Settings window** can always be opened.

## Context menus

Almost everywhere, mouse right-click will open context pop-up menu window. The content of the context menu depends on many factors, such as current window, selection, position of mouse pointer, etc. If you are not sure how to perform specific operation, try right-click at the point/area of interest: desired command might be shown in the context menu.

For example, right click on the DLL component shows all commands that can be applied to that component:



Please note that there could be minor changes in context menu content and arrangement in different NL5 revisions and builds.

## Shortcuts

**F1** – Help (context help on a specific window, selected component, C function, etc.).

**F2** – show Schematic window.

**F3** – show Components window.

**F4** – show Variables window.

**F5** – show Transient window.

**F6** – start transient.

**F7** – continue transient.

**F8** – show AC window.

**F9** – start AC analysis.

**F10** – show Settings window.

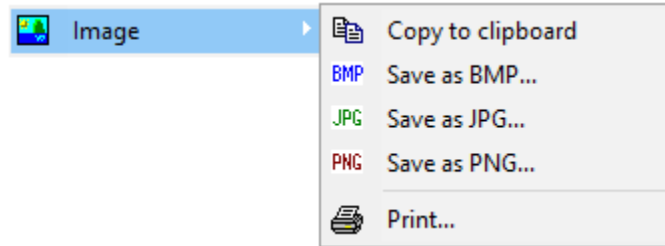
## Mouse

Most mouse and mouse wheel operations are similar in **Schematic**, **Transient**, **AC**, and **Tool** windows.

Some options of mouse wheel operation can be changed in **Preferences/Document/Schematic/Mouse**, and **Preferences/Document/Graphs/Mouse**.

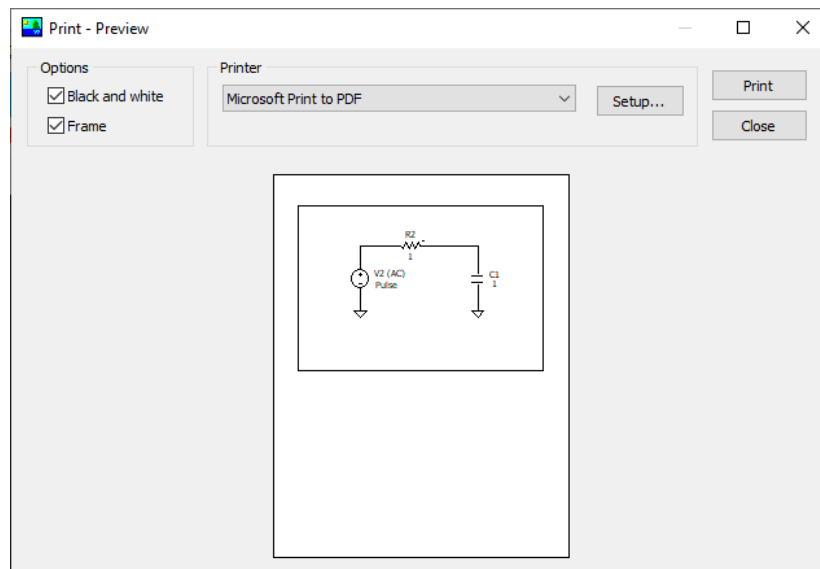
## Image Copy, Save, and Print

**Right-click** on Schematic, Transient, AC, or Tool window, select **Image** in the context menu:



**Copy to clipboard** or **Save** image into the file in any of 3 available formats (**PNG** format is suggested).

**Print** image (or save as PDF, if available). **Preview** window will show up:



Select **Options**:

- **Black and white**,
- **Frame** (Schematic window only).

For printing, select **Printer** from the drop-down list, or click **Setup...** for printer setup.

Click **Copy/Save/Print** button to proceed.

Image Copy/Save operations are also available for other NL5 windows, such as **PWL** window, **Roots** window, and more.

## Data format

**Data format** used in NL5 mostly complies with common engineering and scientific practice.

### Numbers

**Boolean** number can be entered as `false` or `true` (case-insensitive):

```
bool i = true;
bool retvalue = FALSE;
```

When converted to other types, `true` is considered as 1, `false` as 0. When other types are converted to `bool`, non-zero value is considered as `true`, zero value as `false`.

**Integer** (`int`, `int64`) number can be entered in decimal, binary, octal, or hexadecimal formats.

**Binary:** use `'0b'` or `'0B'` prefix, then use digits 0 and 1:

```
0b1111111111, 0B10101010, 0b10
```

**Octal:** start number with prefix 0 (zero), then use digits 0...7:

```
0377, 0123456
```

**Hexadecimal:** use `'0x'` or `'0X'` prefix. Then use 0...9, and capital or low-case A, B, C, D, E, F:

```
0xFF, 0X10aa, 0x10000
```

If a value of a number exceeds 32-bit range, it will be automatically converted to `int64` type. Use `i64` suffix to explicitly define 64-bit integer:

```
0i64, 0xfffffffffi64
```

**Floating point** (`float`, `double`) number can use exponential multipliers **E** or **e**, or **case-sensitive** letter multipliers:

Letter	Multiplier
T	$10^{12}$
G	$10^9$
M	$10^6$
k, K	$10^3$
m	$10^{-3}$
u	$10^{-6}$
n	$10^{-9}$
p	$10^{-12}$
f	$10^{-15}$

For example:

```
1.3e+3, 47E-9, 100k, 0.33u
```



Letter multiplier can be used instead of a decimal point. Zero before decimal point or letter multiplier can be omitted:

```
1k3, .47, n47
```

Infinite value is denoted by:

```
inf
```

**Complex** number consists of real and imaginary parts of floating point type. Imaginary part of a complex number has **lower case** letter 'j' at the end of a number. Letter 'j' cannot be used alone, only as a suffix:

```
50+45j
1+1e-3j = 1+.001j
30j
1+j : wrong! Correct format: 1+1j
```

The following **predefined constants** (case-insensitive) can be used:

```
PI = Pi = pi = 3.14159265359
RAD = Rad = rad = 180/pi = 57.2957795131
LOW = Low = low : low logical level
HIGH = High = high : high logical level
```

Constant RAD can be used to convert degrees to radians and radians to degrees:

```
degrees = radians * RAD
radians = degrees / RAD
```

where `degrees` is value in degrees, and `radians` is value in radians.

All numerical component parameters, and most of other parameters in NL5 are floating point (`double`). Those parameters can be entered in any format, however after that they are automatically converted and stored in the floating point format.

When floating point number is displayed, an engineering notation, with exponential multiplier and power of ten to be multiple of three, is used:

Entered	Displayed
1k3	1.3e+3
47e-8	470e-9
5600000	5.6e+6

## Names

**Component.** When a new component is created, it is assigned a default name: ‘letter’ plus number:

```
R1
V2
```

Then the component can be renamed. The name is case-insensitive and may contain any (almost) characters and symbols. However, to be used in formulas, functions, C-code, etc. it should satisfy C-language standard for the variable name: start with letter or ‘\_’, and then contain only letters, digits, and ‘\_’.

If the component has been renamed, its name will be automatically modified in all appearances of the component name in trace names, formulas, and functions.

To access component parameter in the formula, function, or script expression, use component name followed by dot ‘.’ and parameter name:

```
R1.R
V2.slope
C123.IC
```

If parameter name is not specified, a first parameter of the component will be used:

```
R1 = R1.R
C2 = C2.C
```

If current component model does not have parameters, a component model name will be used:

```
A1 = Amperemeter
```

To access component, which is part of the subcircuit, use subcircuit component name followed by dot ‘.’ and component name in the subcircuit. A nesting level is unlimited: components inside subcircuit, which in turn is part of subcircuit, can be accesses by similar notation:

```
X1.X2.V3.period
```

where x1 and x2 are subcircuits.

To access global variable of the C model of Code component x1 (in a script or command-line) use the following notation:

```
X1.variable_name
```

To access component model name (in a script or command-line) use component name followed by dot ‘.’ and “model”:

```
V1.model = pulse
```

If current component model does not have parameters, “.model” can be omitted, since this is the only component attribute:

```
S1 = Off
```

**Schematic variable.** Schematic variable (a variable defined in the **Variables window**) name has the same format as a component, except it does not have parameters. For example:

```
Freq
X1.var
```

**Trace.** The basic name of transient or AC trace consists of the letter specifying type of the trace (V, I, P, S), followed by component's name in parentheses:

```
V(R1)
I(C2)
P(L3)
```

The trace can be renamed to an arbitrary text:

```
Copy of V(R1)
new trace
V pulse
```

To be used in formulas and expressions, trace name should satisfy C-language standard for the variable name: start with letter or '\_', and then contain only letters, digits, and '\_'.

**Trace and cursors data.** Trace and cursors data shown in the transient or AC table (value at cursor, min, max, mean, etc.) and trace value at specified time can be used in the script expressions (for example, `logdata` command). Please note that to access trace data from Console, Script, and HTTP link, the trace must exist, although it does not need to be shown on the graph or in the table. `left` and `right` are values specified as table "Interval" in the **Settings/Table** window: **Cursors, Screen, All data, or Selection**.

To access data, use the following notation:

- **Cursors (screen)**

```
left           – left interval value.
right          – right interval value.
delta          = right-left
```

- **Transient**

```
V(R1).(1.2)    – value at t=1.2
V(R1).left     – value at "left".
V(R1).right or V(R1) – value at "right".
V(R1).delta    – V(R1).right - V(R1).left.
V(R1).min      – min.
V(R1).max      – max.
V(R1).pp       – peak-to-peak (max – min).
V(R1).mean     – mean (average).
V(R1).rms      – rms.
V(R1).sd       – standard deviation (rms of the signal with subtracted mean value).
V(R1).freq     – frequency.
V(R1).period   – period.
```

`V(R1).integral` – integral.

- **AC magnitude**

`V(R1).(1.2)` – value at  $f=1.2$   
`V(R1).left` – value at “left”.  
`V(R1).right` or `V(R1)` – value at “right”.  
`V(R1).delta` – `V(R1).right - V(R1).left`.  
`V(R1).min` – min.  
`V(R1).max` – max.  
`V(R1).pp` – peak-to-peak (max – min).  
`V(R1).slope` – slope of the gain, dB/dec.

- **AC phase**

`V(R1).phase.(1.2)` – phase value at  $f=1.2$   
`V(R1).phase.left` – phase left.  
`V(R1).phase.right` – phase right.  
`V(R1).phase.delta` – phase delta.  
`V(R1).phase.min` – phase min.  
`V(R1).phase.max` – phase max.  
`V(R1).phase.pp` – phase peak-to-peak (max – min).

If storage data exists, trace data are taken from selected current Run or storage.

To access trace data, trace name should satisfy C-language standard for the variable name: start with letter or ‘\_’, and then contain only letters, digits, and ‘\_’. If the original trace name does not satisfy this requirement, please specify a custom trace name at **Transient/Settings/Format traces** or **AC/Settings/Format traces** page, and then use it in expressions.

## Operators

NL5 supports the following arithmetic and logical operators:

++	--	+	-	*	/	%
!	~	<<	>>	&	^	
<	<=	>	>=	==	!=	
&&		?:				
+=	-=	*=	/=	%=		
&=	^=	=	<<=	>>=		

and type-casting operators:

(bool) (int) (int64) (float) (double) (complex)

See **NL5 User's Reference** for details.

## Functions

NL5 offers many standard and NL5-specific functions. The functions can be used in the C code (script, Code component), traces of **Function** type, and in **Function** model of some components.

For your convenience, there may be several names used for the same function (for example `log10` and `lg`), so you can use the name he/she is more comfortable with. The following functions are available:

sin	sqrt	mag, abs	par	sum
cos	sqr	phase	random, rand	mean
tan, tg	sq	re	gauss	max
asin	pow	im	limit, lim	min
acos	pwr	sign	islow	bool
atan	exp	round	ishigh	int
atan2	ln, log	floor	db	int64
	lg, log10	ceil		float
	lb, log2			double
				complex

Please note that the argument of functions **sin**, **cos**, **tan**, **tg** is in radians, not degrees. Similarly, functions **asin**, **acos**, **atan**, **phase** return radians, not degrees. This is different from NL5 Ver.2.

See **NL5 User's Reference** for details.

## Expressions

Expression may consist of:

- Numbers.
- Predefined constants.
- Names of components, parameters, and variables.
- Local script or C-code variables.
- Operators.
- Functions.
- Parentheses with unlimited nesting level.

For example:

```
2*2
2<<3
sin(2*PI*f) // "f" is schematic variable.
max(R1,R2,R3)
1/((R1+R2)*C1)
```

Expression can be used instead of number in most entry fields in the NL5 windows, and for some component parameters. When **Enter** key is pressed, or **OK** or **Apply** button (if exists) is clicked, the expression is immediately evaluated and replaced with the numerical value.

Please note that division of two integer values will produce rounded integer result according to C-language rules **only** in expressions used in C-code components and script. In all other places, integer values will be considered as double values, and a result will be of double type. This is similar to how calculations are performed in calculators, Excel spreadsheets, etc. For example:

```
1/100 = 0.01 (will be 0 in C-code)
5/2 = 2.5 (will be 2 in C-code)
```

## C language

Simplified C language interpreter is implemented in the NL5. It is used in the **script** and in the **C-code** component.

The following data types are supported:

`bool` – boolean (true/false).  
`int` – 32-bit signed integer.  
`int64` – 64-bit signed integer.  
`float` – same as `double`.  
`double` – 8-byte floating point.  
`complex` – consists of `double` real and imaginary parts.

Only one-dimensional arrays are supported.

The following C keywords, statements, and operators are available:

<code>bool</code>	<code>if...else</code>	<code>continue</code>
<code>int</code>	<code>for</code>	<code>break</code>
<code>int64</code>	<code>while</code>	<code>return</code>
<code>float</code>	<code>do..while</code>	
<code>double</code>	<code>switch</code>	
<code>complex</code>	<code>case</code>	
	<code>default</code>	

The following C language features are currently **not** implemented in NL5:

- Preprocessor (`#define`, `#include`)
- Structures and unions.
- Pointers and references.
- `goto` statement.

Please refer to publicly available resources for general C language syntax description and reference.

## Script

Script is a program which allows automating simulation process. By using script, you can open/close schematic files, modify component parameters, run simulation, analyze the data, save/export simulation results, etc.

Script can be executed from NL5 command line, or at **Script** tab of **Tools** window. Script commands can also be executed from command line in the **Console** tab of **Tools** window, and through **HTTP link** from external applications. This allows using NL5 as an “add-on” simulation engine with popular engineering tools such as MATLAB®, Python, and others.

### Script syntax

Script is written on C language and is executed by C language interpreter built in NL5. All C operators and functions can be used in the script. In addition, NL5-specific script commands can be used to run simulation, analyze data, and save simulation results. See **NL5 User's Reference** for details.

**Component parameters** and schematic variables can be used in expressions and can be modified similar to C variables:

```
R1 = tau / C1;
freq = 1. / V1.period;    // "freq" is schematic variable
```

**Trace and cursors data** can be used in expressions. The data corresponds to the values configured in the Transient or AC Data table. To use trace data, the trace should be added into the Transient or AC data but does not need to be displayed on the graph or in the data table. For example:

```
x = V(out).max - V(out).min;
if(V(C1).(3.45) > threshold) break;
charge = I(C1).mean * delta;
```

### Script commands

Script commands are used to open schematic, control simulation process (run and continue simulation), and save simulation data in different formats. The following general script commands are available:

ac	exit	return	sleep
clear	export	rununtil	stop
close	import	save	store
cmd	logdata	savedata	storetext
cont	open	saveic	traces
cursors	pause	show	tracename
display	ready	silent	tran



Script commands related to Scope tool:

<code>scope.cmd</code>	<code>scope.off</code>	<code>scope.select</code>
<code>scope.get</code>	<code>scope.on</code>	<code>scope.single</code>
<code>scope.getn</code>	<code>scope.read</code>	<code>scope.status</code>
<code>scope.image</code>	<code>scope.refresh</code>	<code>scope.stop</code>
<code>scope.log</code>	<code>scope.run</code>	<code>scope.update</code>

Please note that unlike standard C function calls, script commands parameters do not need to be enclosed into parentheses. For example:

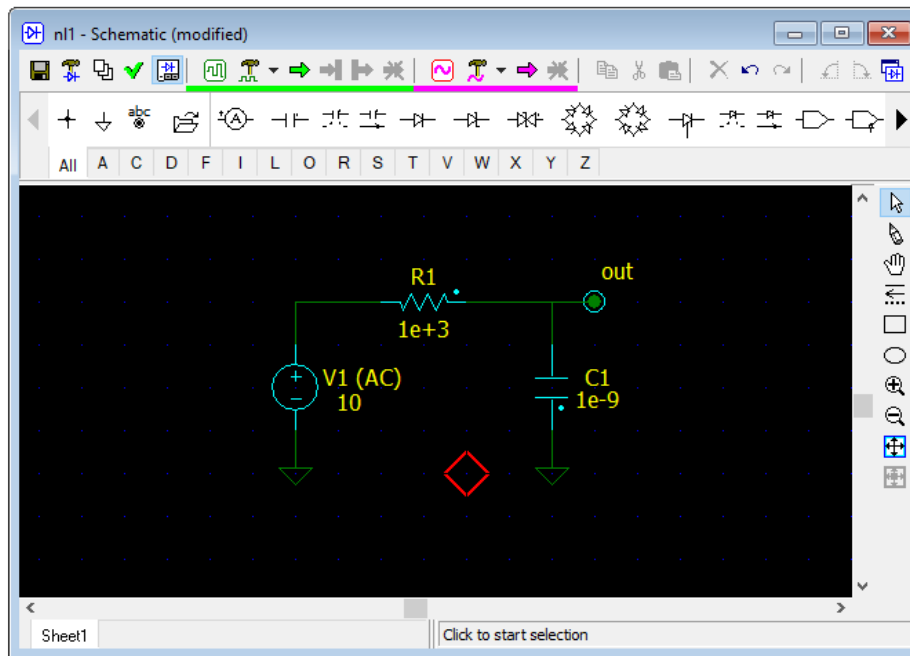
```
open(rc.nl5);
open rc.nl5;
tran(0, 1, 1m);
close();
clear;
```

However, it is highly recommended using standard C notation as much as possible since script syntax might change in the future to satisfy C standard.

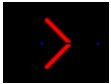
See **NL5 User's Reference** for details.

# III. Schematic

## Edit schematic




### Cursor



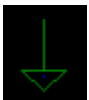
- **Double-click** - set cursor at that point and center it on the screen.
- **Left, Up, Right, Down** – change cursor direction, move cursor.
- **Click** on the cursor close to the cursor corner to change cursor direction.
- **Home** - center cursor on the screen.


### Wire



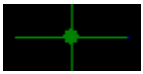
- Press **Space** or click **Wire**  to switch to **Wire** mode.
- **Click** and drag to draw wire.
- **Left, Up, Right, Down** – change cursor direction, draw wire.
- Press and hold **Ctrl** to draw diagonal wire.
- Press **Space** or **ESC** to switch back to **Selection** mode.

### Ground



- Press **G** or click  on the **Components bar**.

## Connection








- Press '.' (dot) or click  on the **Components bar**.

Three wires coming to one point are connected automatically: a connection point will be automatically added during schematic check. Two crossing wires are not connected by default and should be connected manually. All unnecessary connection points will be automatically removed during schematic check. **Warning:** diagonal wire cannot be connected to the wires it is crossing.

## Component













- Select tab with component letter on the **Components bar**, click on the component.  
**OR**
- Press component letter key (for example 'R') one or more time to select required component.
- Move component by keyboard/mouse, rotate, mirror, flip, select view if needed.
- **Right-click** on the component to see context menu, or use toolbar buttons or shortcuts to do the following:
  -  or **Ctrl-L** – rotate component left.
  -  or **Ctrl-R** – rotate component right.
  -  or **Ctrl-M** – mirror component.
  -  or **Ctrl-F** – flip component.
  -  or +/- key – show next component view (if applicable).
- Press **Enter** to place component or press **Del** or **ESC** to cancel.

When component is placed, click on the component to select it, then use the following ways to access component editing commands:

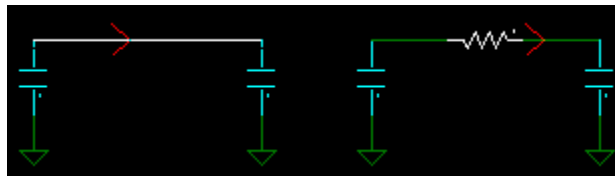
- toolbar buttons,
- keyboard shortcuts,
- **Main menu/Edit, Main menu/Edit/Selection,**
- **Right-click** on the component to see pop-up context menu.

The following component editing commands are available:

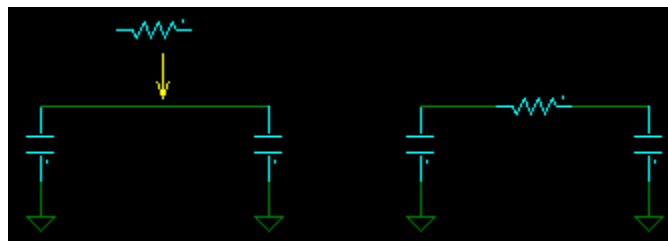
-  – set component at 45 degrees (available for some R, L, C, S, and D components).
-  or **Ctrl-T** – rotate attribute of selected component.
-  or +/- key – show next component view (if applicable).
-  or **Ctrl-L** – rotate component left.

-  or **Ctrl-R** – rotate component right.
-  or **Ctrl-M** – mirror component.
-  or **Ctrl-F** – flip component.
-  or +/- key – show next component view (if applicable).
- **Component**  : more component related commands.
- **Add trace**  : add trace for the component.
- **Double-click** on the component to edit component parameters in the **Components window**. Then, if you finish editing parameters by pressing **Enter** or **Esc**, you will switch back to the schematic.

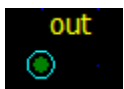
When component is placed above existing wire, a piece of the wire underneath the component is automatically removed, so that no editing of the wire is required:






Similarly, when component is moved/copied above existing wire, a piece of the wire underneath the component will be automatically removed:

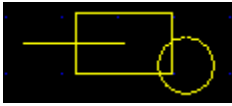






## Label



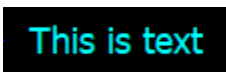
- Press **Enter** or click  on the **Components bar**.
- For new label, enter label **Name** and **Description** (optional), click **OK**.
- For existing label, enter label **Name** or select from the list of existing labels, click **OK**.
- **Double-click** to edit label parameters in the **Components window**.
- **Right-click** on the label to see relevant commands. Some useful commands:
  - **Component**  : more label related commands.
  - **Add trace**  : add trace for the label.


## Drawing



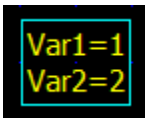
- Click **Line** , **Rectangle** , or **Oval**  to switch to drawing mode.
- **Click** and drag to draw.
- Press and hold **Ctrl** to draw square, circle, or line in the direction multiple of 45 degrees.
- **Double-click** or **Right-click/Format**  to format drawing.


## Text








- **Right-click/Insert text**  to add text to the schematic.
- **Double-click** to edit and format.

## Variables




- **Right-click/Insert variables**  to show schematic variables and their values on the schematic. The list is automatically updated when variables or their values change.









## Zoom and scrolling

- Click  to switch to **Scrolling** mode, then click and drag to scroll the screen.
- OR**
- Press and hold **Shift**, then click and drag to scroll the screen.
- **Shift-Left, Shift-Up, Shift-Right, Shift-Down** - scroll the screen.
- **Ctrl-mouse wheel** - scroll horizontally.
- **Shift- mouse wheel** - scroll vertically.
-  or **PgUp** - zoom-in.
-  or **PgDo** - zoom out.
-  or **Ctrl-Home** - fit all schematic to the screen.
-  or **Shift-Home** - fit selection to the screen.
- **Mouse-wheel** – zoom in/out.







## Selection

- **Click** on the schematic element to select.
- Point on empty space, **click** and drag to select a block (rectangle).
- Press and hold **Ctrl** while selecting – add to existing selection.
- **Ctrl-A** – select all schematic.
- **Right-click/Select net** - selects all wires connected to the selection either directly, or through labels (including other sheets).
- **Click** on empty space or press **Esc** to unselect (press **Esc** twice to unselect a block).
- **Left, Up, Right, Down** – move selected block.
- **Click** and drag to move element or selection.
- **Click** and drag to move attribute of selected component.
- **Shift + Click** and drag to move element or selection with **rubber bands**.
- **Ctrl + Click** and drag to copy element or selection.
- **Ctrl + Shift + Click** and drag to copy element or selection with **rubber bands**.
- **Right-click/Format**  to format selection.
- **Right-click/Disable** or **Ctrl-D** to disable selection.
- **Right-click/Enable** or **Ctrl-E** to enable selection.

Disabled schematic elements are shown in “disabled” color and are not used for simulation. Disabling elements allows temporarily exclude elements from simulation without deleting.


-  or **Ctrl-C** – copy selection.
-  or **Ctrl-X** – cut selection.
-  or **Ctrl-V** – paste selection.
-  or **Del** - delete selection.
-  or **Ctrl-L** – rotate selection left.
-  or **Ctrl-R** – rotate selection right.
-  or **Ctrl-M** – mirror selection.
-  or **Ctrl-F** – flip selection.

## General commands and shortcuts

- Click  to show/hide **Components bar**, resize the bar by mouse if needed.
- Press **Tab** to change attributes display for all components.
- **Right-click** to open context menu. Some useful commands:
  - **Component**  - access component/label related commands.
  - **Add trace**  - add trace for the component/label.
  - **Image**  - copy schematic image to clipboard or save into the file.
-  or **Ctrl-Z** – Undo.
-  or **Ctrl-Y** – Redo.

## Open and save schematic


### Open

When **Open**  schematic of a special type, some actions may be required.


**Encrypted schematic.** When opening encrypted schematic, a password should be entered. Please be aware that for security purposes it is impossible to open (decrypt) such a schematic without a password.

**Ver.2 schematic.** When schematic with old Ver.2 format s opened, it is automatically converted to a new Ver.3 format, and then the window with conversion notes shows up. When saved, this converted schematic will be saved in a new Ver.3 format and cannot be opened by NL5 Ver.2 anymore.

### Save

**Save**  and **Save As** are standard saving commands.

**Save As Encrypted.** Enter password and save encrypted schematic. Please be aware that for security purposes it is impossible to open (decrypt) such a schematic without a password.

To save schematic with transient/AC data or script text, select saving options in the **Schematic Settings** and tools , **Save options** tab.



## Components

All available components are shown in the **Components bar**:



Click  to show/hide **Component bar**. Resize the bar height by mouse if needed.

Each **component** has a letter associated with it. When a new component is placed, its default **name** consists of a letter and number. The name can be changed to arbitrary text, however, to be used in formulas, it should satisfy C-language standard for the variable name (start with letter or '\_', then consist of letters, digits, or '\_').





**Label** is a special type of component: there can be many labels with the same name in the schematic. All labels with the same name are electrically connected. Labels in the subcircuit are local to the subcircuit and are not connected with labels of the same name in the main schematic or other subcircuits.

Labels can be used:

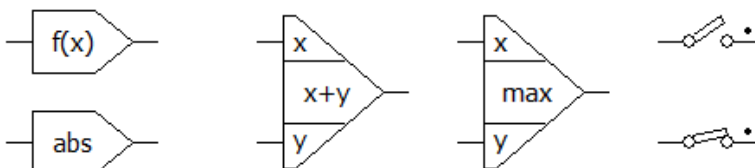
- To connect different points of the schematic without wire.
- To connect parts of the schematic located on different sheets.
- As a simulation probe (**V trace**).
- As a voltage source.

Please be aware that when copy/paste label into schematic, the label will be automatically (without notification) connected to labels with the same name, if such labels already exist in the schematic.

**Symbol.** Most components have just one symbol associated with it, which cannot be changed. Some components may have several **views**, when the symbol is modified depending on different pin locations (for example, location of positive/negative input pins), and different functionality (for example, inversion of logical signals). For logical gates, changing **view** may also be used to change gate logical function (AND, OR, XOR).

Select component and click  to change a **view**, or press '+/-' keys to go through all applicable combinations of **view** , **mirror** , and **flip** .

Also, some components may have different symbols for different component **model** or component parameters. For example, symbol of the **Function** component indicates functionality of a selected model, and symbol of the **Switch** component indicates switch position in non-active state:



**Model.** Component **model** defines specific component functionality. For example, voltage source models include **Pulse**, **Sin**, **Step**, **File**, and more. Select component model in the drop-down list above parameters.

**Parameters.** Component parameters may be of different types: floating point, text, etc. Floating point parameter can be entered as an expression, for example:


```
R1.R*2.5
```

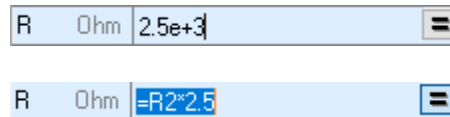
Such an expression will be immediately calculated and replaced by a resulting value.

Some parameters of floating point type may be **blank** (empty), which means the value is not defined. For example, blank **IC** (initial condition) parameter of capacitor means that its initial voltage at  $t=0$  is not defined and will be determined during DC operating point calculation.

**Formula.** Floating point parameter can be entered as a formula if started with '=' symbol:

```
=R1.R*2.5
```

Enter formula and press **Enter**, the formula will be calculated immediately. Then it will be recalculated automatically when required. The parameter entered as formula will have  button on the right. Click the button to see and edit formula expression:



If formula expression cannot be evaluated, #VAL text will be displayed instead of a number. Erase the formula to leave its last value as a constant.

**Function.** Some component parameters represent a **function**. Function is an expression which is recalculated at every transient or AC calculation step. Function can use the following variables:

**t** – current transient time, s.

**f** – current AC frequency, Hz.

**w** – angular AC frequency,  $w = 2\pi f$ .

**s** or **p** – Laplace parameter,  $s = p = j*2\pi f$ .

**x, y** – input signals for **Function** model.


**V(name)** – voltage on the component **name**. V trace should be available for the component.

**I(name)** – current on the component **name**. I trace should be available for the component.


**P(name)** – power on the component **name**. P trace should be available for the component.

For example:

```
sin(t*1000)*(1+cos(t*10))
(t%2>1)?1:-1
sq(V(r1))/r1
1/(1+s*R1*C1)
```

**Logical levels and threshold** for all components are defined in the **Schematic Settings and tools** , **Logical levels** tab.

**Attributes.** There are 2 **attributes** of the component that can be displayed on the schematic: **Name** and **Value**.

**Name** is automatically assigned to the component when it is placed and can be changed later. Also, an arbitrary text, **Description**, can also be added and displayed along with the **Name**. To edit component **Name** and **Description**, click **Rename**  on the **Components window** toolbar.

**Value** content depends on the component type. By default, it is the first parameter of the model, model name, or empty. It can be customized to display any of component parameters in different format, formula expression, etc.

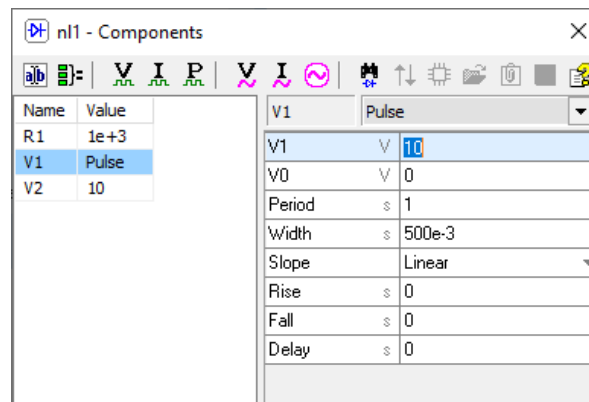
Press **Tab** to toggle attributes display mode for all components in the schematic:

- **Name** only
- **Name and Value**
- **Value** only
- No attributes

To move attributes, select component first, then click and drag the attribute.


To format attributes of specific component, **right-click** on the component, select **Format** , then select **Attributes** tab.

Component model and parameters can be selected and edited in the **Components windows**:



**Double-click** on the component or press **F3** to open or switch to the **Components window**. If **double-click** on the component in the schematic window was used, edit parameter and press **Enter** to switch back to schematic window.

Operations available for selected component are available on the toolbar.

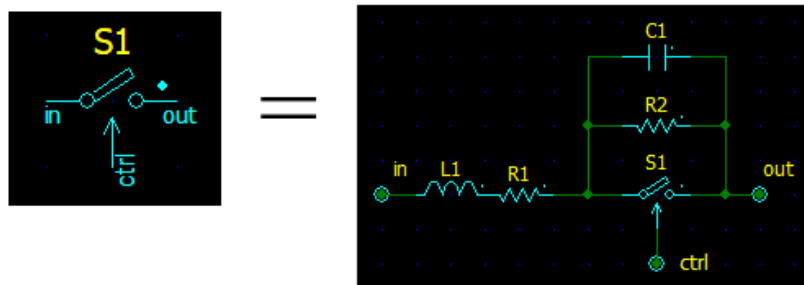
Click **Find component**  to find select component on the schematic. The component symbol will be selected and centered on the screen. For **label**, click again to see the next label with this name on all schematic sheets.

## Working with special component types and models

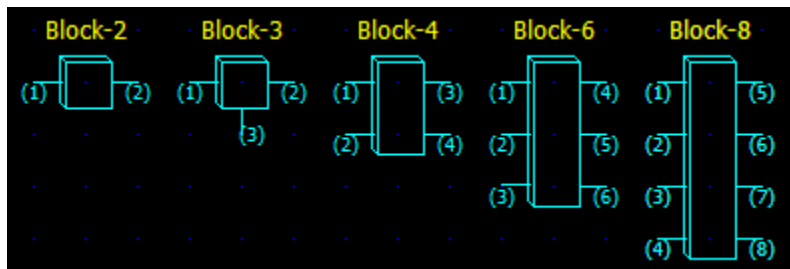
### Subcircuit model



Subcircuit model (**SubCir**) allows creating simple and readable schematic by substituting some part of the schematic with one component (symbol). When simulation starts, the component with **SubCir** model is replaced by actual schematic loaded from subcircuit file.

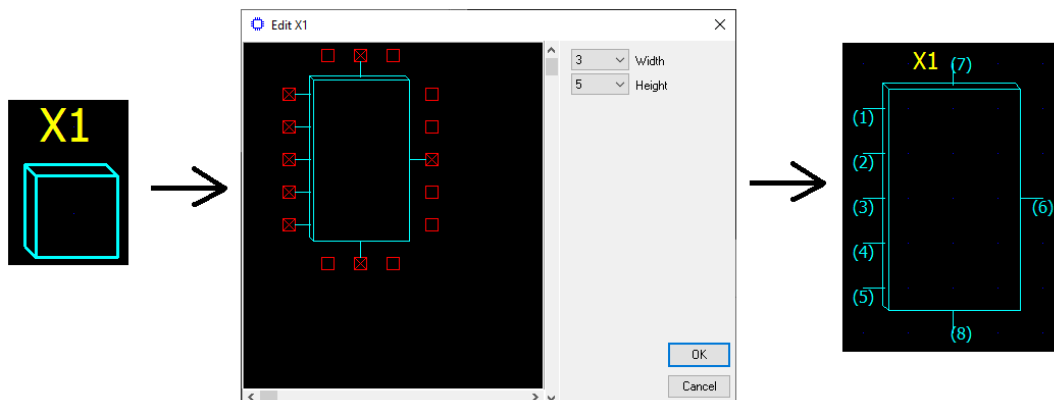
**SubCir** model is available almost for all components, so that some simple symbol can in fact represent quite complex circuit. For example, **switch** component with **SubCir** model may represent a complex circuit with additional resistors and parasitics:



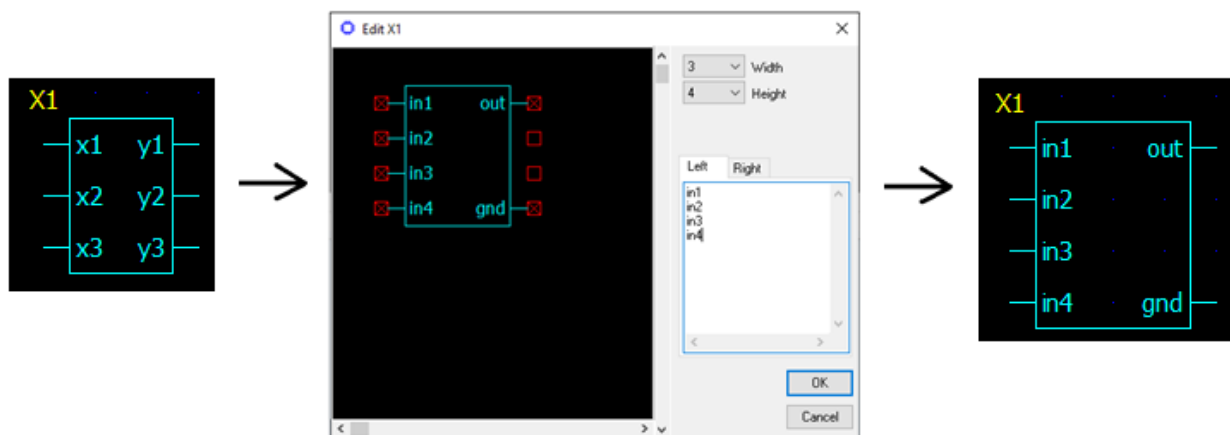
Several **X** components **Block-2**, **Block-3**, **Block-4**, **Block-6**, and **Block-8** are offered to be used for subcircuit with fixed predefined symbol and number of pins:



Component **X**, **Custom block** is a customized component. When placed on the schematic, it does not have any pins. **Right-click** on the component, select **Component** , then **Edit component**  to open **Edit component** window:




**NL5 circuit** is a special component with **SubCir** model. It is a customized component as well, but in addition, names of labels connected to component pins are also defined in **Edit component** window. Pin names will be shown on the component symbol:




**SubCir** model has the same parameters for all component types (except **NL5 circuit**, which does not have **Pin1...PinN** parameters):

Model	Parameter	Units	Description
SubCir	<b>File</b>		File name of subcircuit schematic.
	<b>Pin1</b>		Name of subcircuit label connected to pin 1
	...		...
	<b>PinN</b>		Name of subcircuit label connected to pin N
	<b>Cmd</b>		Subcircuit start-up command string
	<b>IC</b>		Subcircuit Initial conditions string

**File** is a file name of subcircuit schematic file. Enter file name manually or click  to select the file. File extension `.nl5` can be omitted. If **File** does not have a full path, NL5 will search for the file in the following order:

- In the directory where schematic file is located.
- In the directories specified in the **Library** list (see **Schematic Settings and tools** chapter).
- In the directories specified in the **Library** list relative to the directory of schematic file.

Parameters **Pin1...PinN** define subcircuit labels connected to the pins. If label name is defined, it is displayed on the component image, otherwise a pin number in parentheses is displayed. Any **PinN** parameters can be empty, which means that the pin is not connected to the circuit. To disable labels/pin numbers display, **right-click** on the component, select **Format** , go to **Attributes** tab, unselect **Show subcircuit labels**.

**Cmd** is a string with comma-separated expressions in “name=value” format, which is executed when subcircuit is loaded. This allows using the same subcircuit file with modified values of different components. For example:



```
R1=1k,R2=12k,C1=5n
```



where R1, R2, and C1 are subcircuit components .

**IC** is a string similar to **Cmd**. It defines initial conditions of subcircuit components, for example:


```
C1.IC=10,O2.IC=0
```

where C1 and O2 are subcircuit components. Unlike **Cmd**, **IC** parameter can be automatically modified by the following commands:

- **Save initial conditions**  : fills in **IC** string with current IC's of all components in the subcircuit.
- **Schematic Settings and tools**  , **Initial Conditions** tab, check box **Clear subcircuits IC**: clears **IC** string.

Subcircuit is always being loaded from the file when simulation starts, even if subcircuit is currently opened in the same NL5 instance. If subcircuit was modified, it should be saved into the file before next simulation start in order to changes take effect. If transient was paused, subcircuit modified, and then simulation continued, subcircuit will not be automatically loaded. Instead, it should be “reloaded” manually: **right-click** on the component, select **Component** , then **Reload subcircuit**  .

To access and modify subcircuit component parameters from command line, script, HTTP-link, and NL5 DLL functions, use X1.R2 notation, where X1 is component with **SubCir** model, and R2 is component inside the subcircuit. Please note that subcircuit component may be accessed only when simulation is started and then paused, so that subcircuit file is loaded into NL5.

Subcircuit can be **attached**  to the component. Attached subcircuit becomes a part of the component, it will be saved into schematic file with the component, so that subcircuit file is not needed for simulation anymore. Attached subcircuit can be edited and saved back to the attachment, so that it will affect only component it is attached to. As a result, several components can use the same subcircuit, with modifications specific to that component. Also, subcircuit components may be accessed any time, even when simulation is not running.

## PWL model

**PWL** (PieceWise Linear) model describes non-linear characteristic of the component with piecewise linear approximation.

Please note that **pwl** parameter of the **PWL** model may actually specify PWC (PieceWise Constant) function, which represents sensitivity (derivative) of PWL function. For instance, **pwl** parameter of a resistor specifies  $R(V)$ , which is PWC function, while resistor  $I(V)$  characteristic is true PWL function. The parameter is still called **pwl** (not **pwc**) for historical reasons.

The following table shows PWC (**pwl** parameter) and corresponding PWL functions for NL5 components:

Component	PWC	PWL
Resistor, diode, zener	$R(V)$	$I(V)$
Resistor	$R(I)$	$V(I)$
Capacitor	$C(V)$	$Q(V)$
Inductor	$L(I)$	$H(I)$
Voltage controlled voltage source, OpAmp	$K(V)$	$V(V)$
Current controlled voltage source	$K(I)$	$V(I)$
Voltage controlled current source	$K(V)$	$I(V)$
Current controlled current source	$K(I)$	$I(I)$

For voltage/current controlled linear components (R, C, L, and amplifier) **pwl** parameter is also PWC function, describing change of component main parameter (R, C, L, K) with control signal:

Component	PWC
Voltage controlled resistor	$R(V_{in})$
Current controlled resistor	$R(I_{in})$
Voltage controlled capacitor	$C(V_{in})$
Current controlled capacitor	$C(I_{in})$
Voltage controlled inductor	$L(V_{in})$
Current controlled inductor	$L(I_{in})$
Voltage controlled amplifier	$K(V_{in})$
Current controlled amplifier	$K(I_{in})$

**PWL** model of a resistor is described here as an example.

**pwl** parameter is a comma-separated string, describing PWC function:

$R_0, V_1, R_1, V_2, R_2, \dots, V_N, R_N$

where:

$R_0$  is resistance while voltage across the resistor is less than  $V_1$ .

$R_1$  is resistance while voltage across the resistor is between  $V_1$  and  $V_2$ .

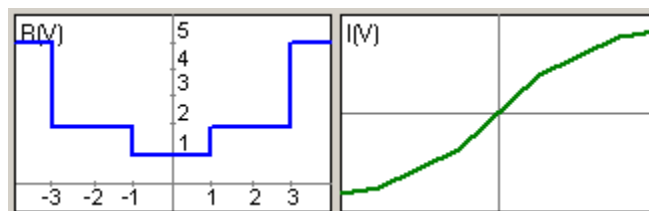
...

$R_N$  is resistance while voltage across the resistor is greater than  $V_N$ .

Values  $V_1 \dots V_N$  should be given in ascending order. The resulting PWL characteristic is calculated automatically, and always goes through the origin (0,0). Please note that only numbers can be used in **pwl** parameter: formulas are not allowed.

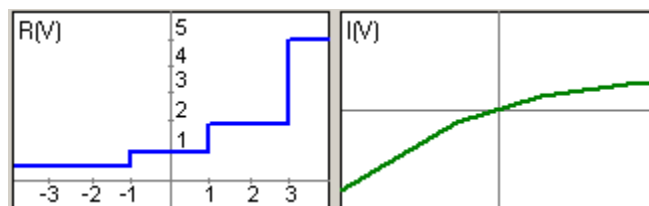
PWC function can be symmetrical or non-symmetrical. **Symmetrical** function is defined only in the interval from zero to plus infinity; the negative part of PWC function is symmetrical to positive one:

**pwl** = 1, 1, 2, 3, 5



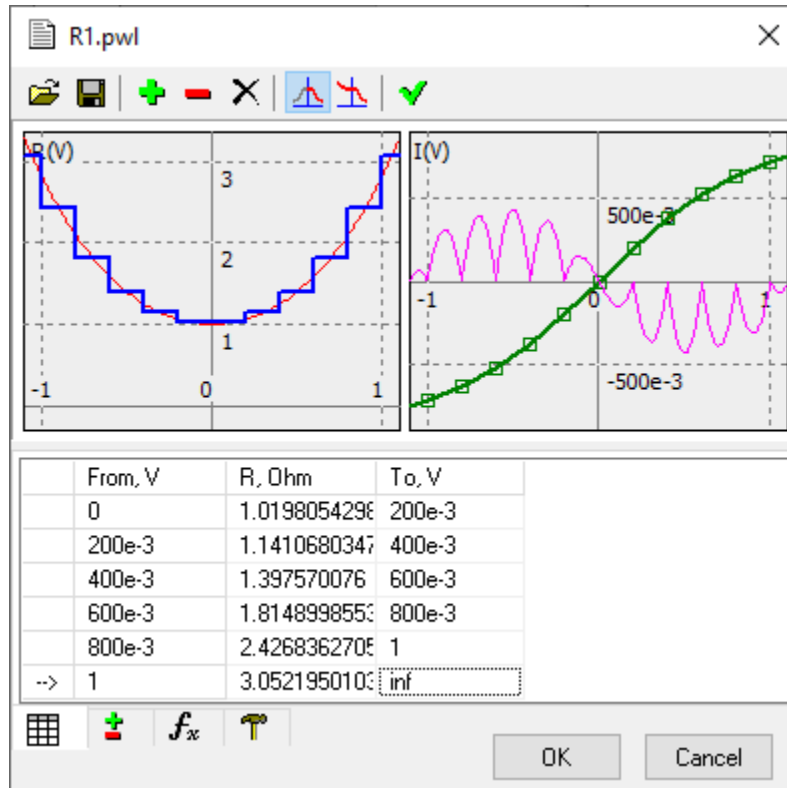
**Non-symmetrical** function is defined from minus to plus infinity. One of the argument points must be **zero**: it serves as an indicator of non-symmetrical characteristic:

**pwl** = .5, -1, 1, 0, 1, 1, 2, 3, 5






To edit **pwl** parameter, select the parameter and click  to open **PWL** window:




**Right-click** on the graphs area to see context menu with relevant commands.  
The following tabs are available:


**Table** . Edit PWL data in the table.


 in the first column indicates selected row.


Select cell and edit the number: **From** value is updated automatically.


 - symmetrical PWC. The first **From** value will be **zero**.


 - non-symmetrical PWC. The first **From** value will be **-inf**.

 - split selected row.

 - remove selected row.

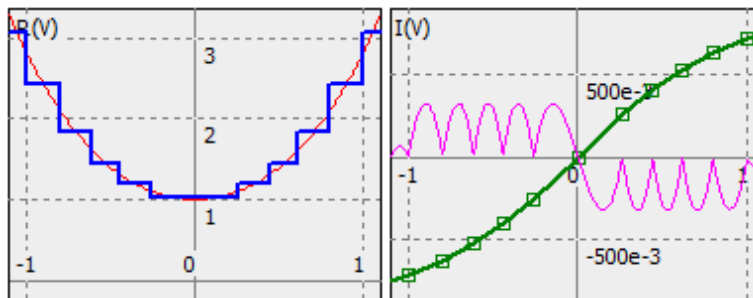
 - clear all data.

 or **Enter** - refresh graphs.

**Add/remove** . Add or remove points (rows) in the specified range.

**Approximate  $f_x$** . Approximate arbitrary function.

- Select PWL parameter or characteristic from the drop-down list. For instance, select R(V) or I(V) for resistor.
- Enter  $f(x)$  as a function of parameter  $x$ .
- Click or press **Enter** to refresh graphs. Both R(V) and I(V) functions will be calculated and shown on the graph.
- To approximate function  $f(x)$  using existing points (rows) in the table, fill in **To** column of the table, click **Approximate using existing points**. Select **Automatic update** to perform approximation automatically on any changes in the data table.
- To approximate function with automatic points selection to provide minimal approximation error, specify **From**, **To**, and **Interval** of approximation, click **Approximate in the range**:




## List model (voltage and current source)

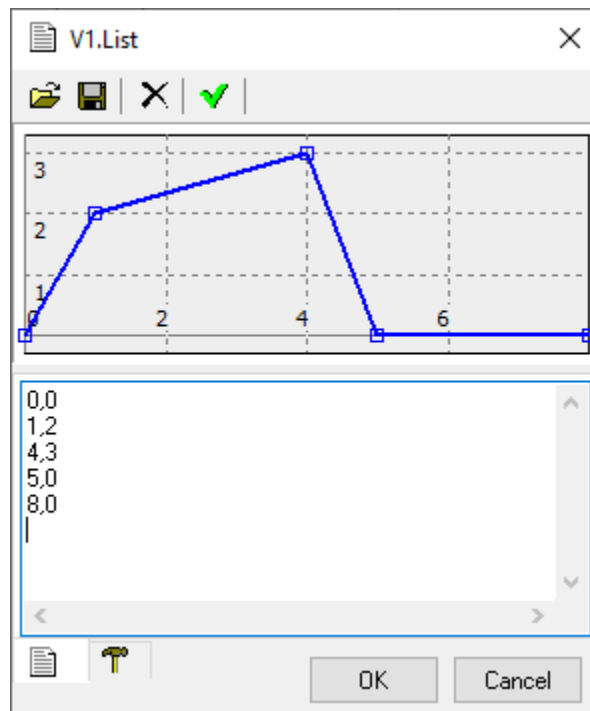
**List** model describes piecewise linear voltage or current source.

**List** parameter is a comma-separated string with time/value pairs:

$$T_1, V_1, T_2, V_2, \dots, T_N, V_N$$

Signal value between specified points is linearly interpolated. Signal value before  $T_1$  is  $V_1$ , signal value after  $T_N$  is  $V_N$ . Values  $T_1 \dots T_N$  should be given in ascending order. Although the signal is defined on the interval  $T_1 \dots T_N$ , it can be repeated continuously, or delayed by setting component parameters **Delay** and **Cycle**. Please note that only numbers can be used in **List** parameter: formulas are not allowed.

Edit **List** parameter manually, or click  to open **List** window:



Enter data in the table as comma-separated time/value pairs. The data will be automatically sorted in ascending order when refreshed.

## List model (switch and logic generator)

**List** model describes switching sequence of **Switch** component, and logical signal of **Logic Generator** component.


**List** parameter is a comma-separated string with time/value pairs:

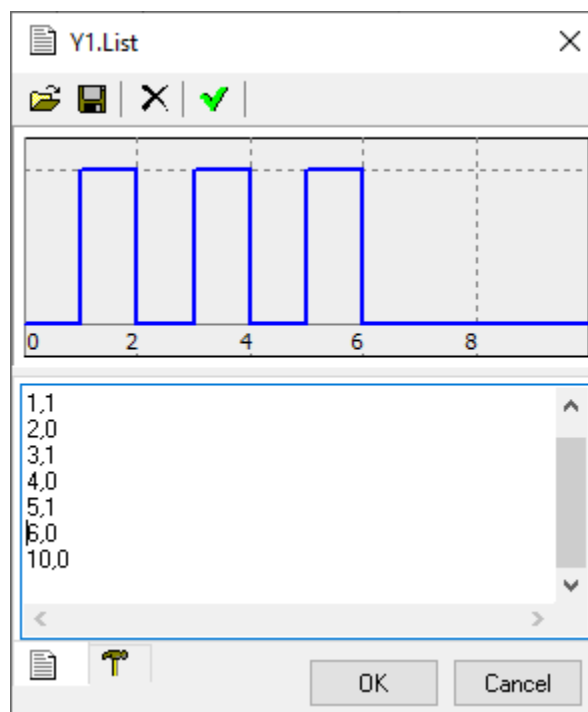
$$T_1, S_1, T_2, S_2, \dots, T_N, S_N$$

where  $T_i, S_i$  pair defines state of the signal at specified time:

- Positive value corresponds to **On** state of the switch, or **High** state of the logic generator.
- Zero or negative value corresponds to **Off** state of the switch, or **Low** state of the logic generator.

Signal state before  $T_1$  is  $S_1$ , signal state after  $T_N$  is  $S_N$ . Values  $T_1 \dots T_N$  should be given in ascending order. Although the signal is defined on the interval  $T_1 \dots T_N$ , it can be repeated continuously, or delayed by setting component parameters **Delay** and **Cycle**. Please note that only numbers can be used in **List** parameter: formulas are not allowed.

Edit **List** parameter manually, or click  to open **List** window:



Enter data in the table as comma-separated time/value pairs. The data will be automatically sorted in ascending order when refreshed.

## Table model

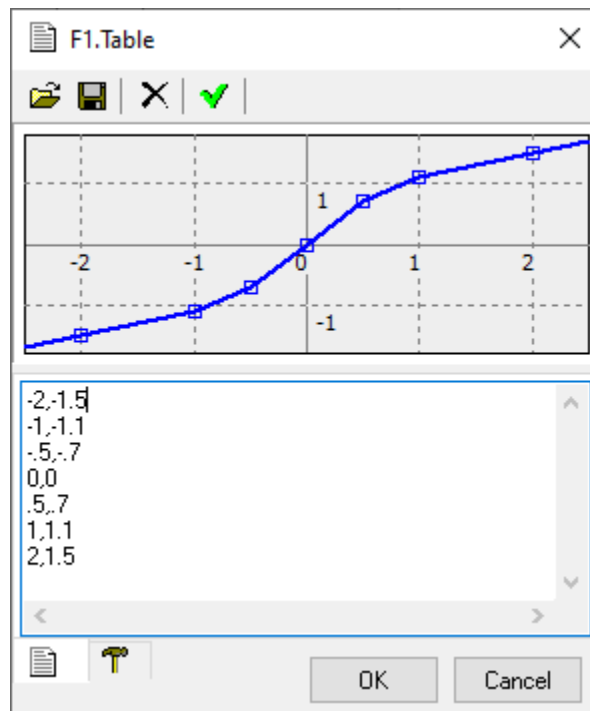
**Table** model describes look-up table of the **Function** component.

**Table** parameter is a comma-separated string with input/output pairs:

$$X_1, Y_1, X_2, Y_2, \dots, X_N, Y_N$$

where  $x_i, y_i$  pair defines input value  $x$  and output value  $y$ . Output value between specified points is linearly interpolated. Output value below  $x_1$  is linearly extrapolated using  $x_1 \dots x_2$  interval data, output value above  $x_N$  is linearly extrapolated using  $x_{(N-1)} \dots x_N$  interval data. Values  $x_1 \dots x_N$  should be given in ascending order. Please note that only numbers can be used in **Table** parameter: formulas are not allowed.

To edit **Table** parameter, select the parameter and click  to open **Table** window:



Enter data in the table as comma-separated input/output pairs. The data will be automatically sorted in ascending order when refreshed.

## 2D Table model

**Table** model describes 2D (two-dimensional) look-up table of the **Function-2** component.


**Table** parameter of the model defines output value **z** as a function of **x** and **y** inputs of the component in the following format:

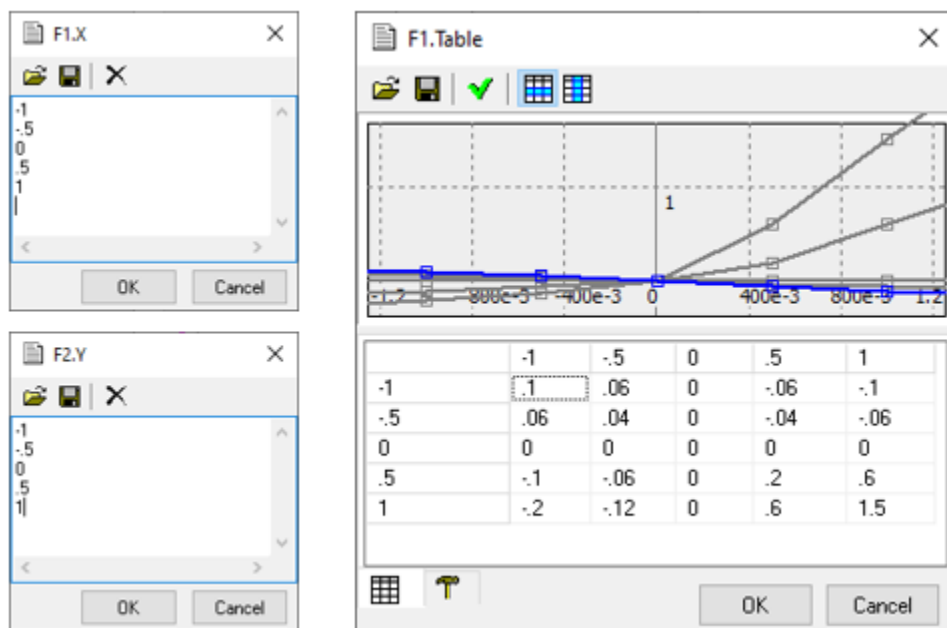
$$z_{11}, z_{12}, \dots, z_{1N}, z_{21}, z_{22}, \dots, z_{2N}, \dots, z_{M1}, z_{M2}, \dots, z_{MN}$$

where:

- $z_{ij}$  defines output of the function for input values  $x_i$  and  $y_j$ .
- **n** is total number of **x** points, defined by **X** parameter of the component.
- **m** is total number of **y** points, defined by **Y** parameter of the component.

Output value between specified **x** and **y** points is linearly interpolated on both coordinates. Output value below  $x_1$  is linearly extrapolated using  $x_1 \dots x_2$  interval data, output value above  $x_N$  is linearly extrapolated using  $x_{(N-1)} \dots x_N$  interval data. The same rule is applied to **y** coordinate. Please note that only numbers can be used in **Table**, **X**, and **Y** parameters: formulas are not allowed.


First, enter **X** and **Y** parameters, then enter **Table** values (select the parameter and click ). Columns correspond to **X** parameter, rows – to **Y** parameter:




The image shows three windows from the software interface. On the left, there are two windows for entering parameters: 'F1.X' and 'F2.Y'. Both windows have a list of input values: -1, -0.5, 0, 0.5, 1. On the right is a larger window titled 'F1.Table'. It features a graph at the top with a grid and a blue line representing a function. Below the graph is a table with the following data:

	-1	-0.5	0	0.5	1
-1	.1	.06	0	-.06	-.1
-0.5	.06	.04	0	-.04	-.06
0	0	0	0	0	0
0.5	-.1	-.06	0	.2	.6
1	-.2	-.12	0	.6	1.5



At the bottom of the 'F1.Table' window, there are two icons: a grid icon and a T-shaped cursor icon.

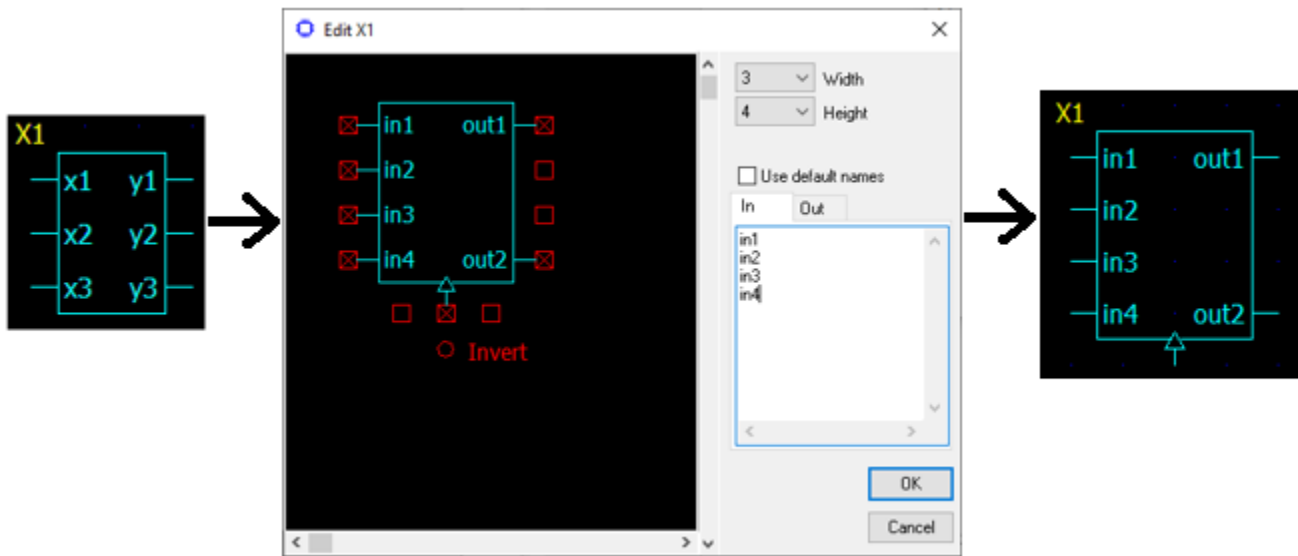
 - show  $z(x)$  traces: each trace represents one row of the table. Total number of traces is **m**, highlighted trace corresponds to the selected cell of the table.

 - show  $z(y)$  traces: each trace represents one column of the table. Total number of traces is **n**, highlighted trace corresponds to the selected cell of the table.

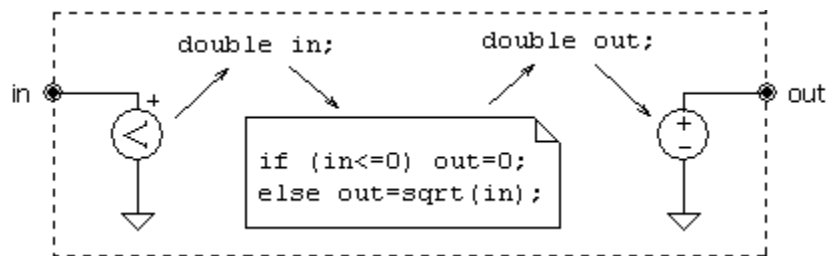
## C-code component

In **C-code** component, the component function code can be written in C language. The code is interpreted by NL5 during transient simulation. Although C code interpretation is relatively slow, using **C-code** component allows fast iterations of the code and easy access to C-code variables for debugging. When the code is finalized, it can be compiled and built as DLL for faster simulation (see **Working with DLL** chapter).

**C-code component** is a customized component. **Right-click** on the component, select **Component** , then **Edit component**  to open **Edit component** window and specify symbol size, inputs, outputs, and clock (optional):

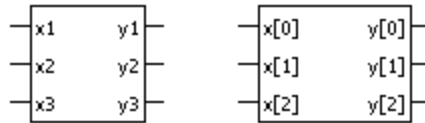


**Execution.** A principle of C code execution is shown on the following diagram:



Each input/output pin with a name has a variable of the same name in the C code associated with the pin. A voltage on the input pin of the component (for example, **in**) is measured by a “virtual” voltmeter, and is assigned to the variable `double in`. While C code is executed, a new value of output variable `double out` is evaluated. A voltage equal to `out` value is set to a “virtual” grounded voltage source, connected to the output pin of the same name **out**.

Inputs and outputs can also be assigned to array elements. For example, inputs `x[0]`, `x[1]`, and `x[2]` correspond to the array `double x[3]`.



Input/output variables corresponding to inputs and outputs (arrays) are **global** variables and should be declared as variables of `double` type in the beginning of the code, outside of `init` or `main` functions. (See **global variables** chapter below for details). If not explicitly declared, all required input/output variables will be automatically created during initialization phase of the analysis at  $t=0$ .

**Code structure.** A typical code structure is the following:

```
// Global variables
...

// Initialization function
init()
{
    ...
}

// Main function
main()
{
    ...
}
```

If complex initialization of global variables is not required, initialization function can be omitted. Then the code will look like this:

```
// Global variables
...

// Main function
main()
{
    ...
}
```

If global variables (other than inputs and outputs) are not used, both initialization and main functions can be omitted, and the C-code will look like that:

```
// The code
...
```

**Global variables.** Global variables are **public** variables of the code. They can be accessed outside the C code: from the script (for example, for data logging), from the command line, displayed as a trace on the transient graph, used in the expressions for V/I sources, etc. Also, global variables are **static**: they persist and keep their values during entire transient simulation, between calls of C-code functions.

Global variables can be used for storing code data that are calculated once during initialization, and then used in the `main` function on each simulation step. Global variables can also be used for storing values that are calculated on one simulation step, and then are used on the next simulation step.



Global variables should be declared in the beginning of the code outside of `init` and `main` functions. Variables can also be initialized during declaration. However, if some code is required for initialization, it should be done in the `init` function.

Although input/output variables are always global, they can be explicitly declared as global variables as well. If not explicitly declared, all required input/output variables will be created automatically during initialization phase of the analysis at  $t=0$ .

To access global variables outside the C-code use the following notation:

```
component_name.variable_name
```

For array elements, use the following notation:

```
component_name.array_name[index]
```

For example, to display global variable as a trace on the transient graph, add a trace of **function** type, and enter variable name, or any expression containing the variable:

```
X1.out1
Xcode.array[3]
X1.X2.temp      (X1 – subcircuit, X2 – C-code component)
```

**Initialization function** is executed once at the beginning of simulation at  $t=0$ . The name of the initialization function should be `init`. Initialization code is used to initialize global variables if some complex code is required for initialization. Initialization function is optional and can be omitted. Variables declared in the `init` function are **local** and persist only during execution of that function.

**Main function** calculates output variables using current values of input variables during transient simulation. If **clock** pin does not exist, the function is executed on every calculation step. If **clock** pin exists, the function is executed only on rising or falling edge of logical clock signal. The name of the main function should be `main`.

Variables declared in the `main` function are **local** and persist only during execution of that function. Use global variables for storing values that are calculated on one simulation step, and then are used on the next simulation step.

Along with global and local variables, the following **read-only** variables and functions can be used in the main function:

```
Component parameters (such as R1, C2, V.period, etc.),
t - current time
V(name) - voltage on the component name
I(name) - current through the component name
P(name) – power on the component name
S(name) – state of the component name
```

where *name* is the name of any component in the schematic, and V, I, P, or S value is available for that component.



Code of `main` function can be modified during transient simulation: pause the transient, make changes, and continue transient. The changes will take effect immediately.

**Initial conditions (IC parameter).** Initial conditions is a string with comma-separated expressions in “name=value” format assigning initial values to output variables and global variables of the code. For example:


```
y1=1.2,y2=0,y3=2.345,integral=-4.19,counter=100
```

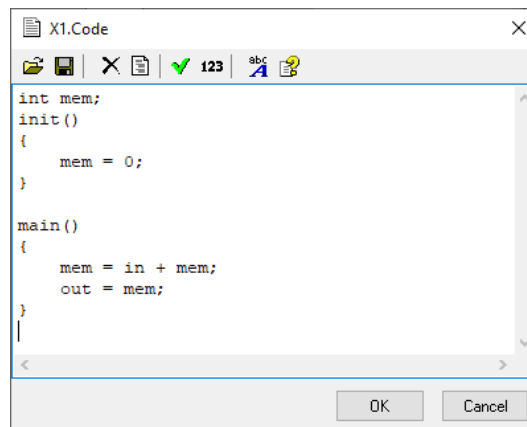
Here `y1`, `y2`, and `y3` are output variables; `integral` and `counter` are global variables.

IC string will be automatically modified by the following commands:

- **Save initial conditions** : fills in IC string with current variable values.
- **Schematic Settings and tools** , **Initial Conditions** tab, check box **Clear C-code and DLL IC**: clears IC string.




If not empty, IC will be executed right after initialization (`init` function), and before the first transient simulation step at `t=0`.

**Editing C-code.** To edit the **Code**, select the parameter and click  to open **Code** window:





```
X1.Code
int mem;
init()
{
    mem = 0;
}

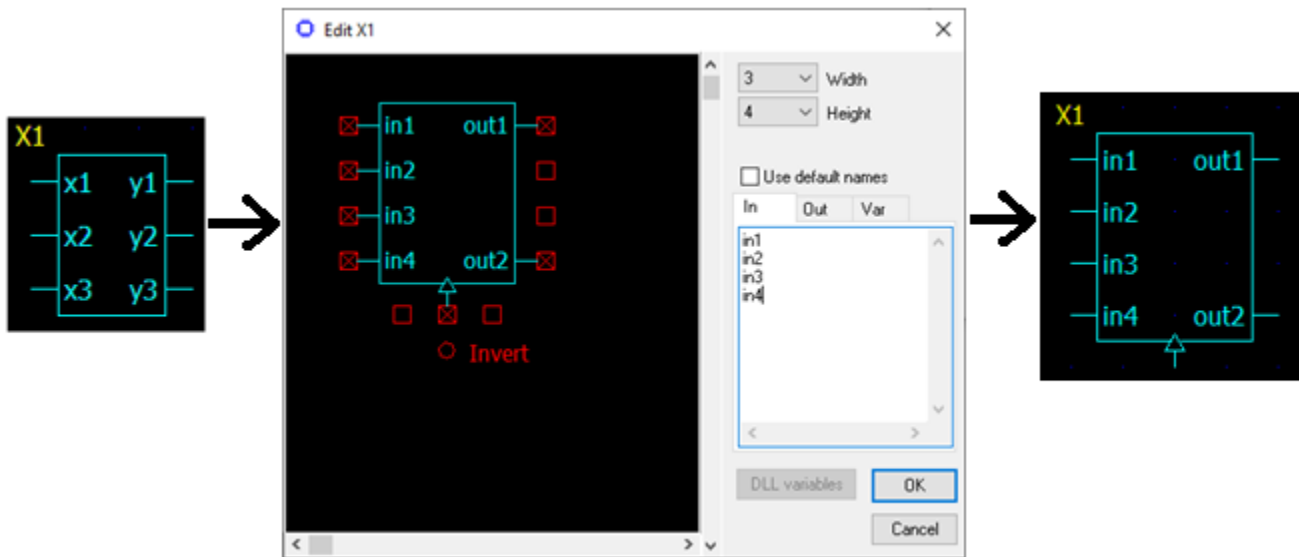
main()
{
    mem = in + mem;
    out = mem;
}
|
OK Cancel
```

-  - clear code and set code template with empty `init()` and `main()` functions.
-  - check code. A message box with errors or warnings found will be shown.
- 123** - show line numbers (no editing allowed in this mode).
-  - Help (**F1**). Opens Help topic on statement, operator, or function, where cursor is located.

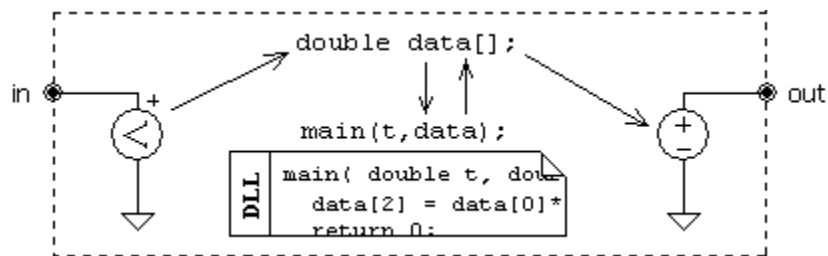
## DLL component

In **DLL** component, the component function code can be written in C/C++ language, compiled by C compiler, and built as Windows DLL (Dynamic-Linked Library). DLL functions will be called by NL5 during transient simulation. DLL code will be executed much faster than C code in **C-code** component. However, changing the code requires recompiling the code and rebuilding the DLL.

**DLL component** is a customized component. **Right-click** on the component, select **Component** , then **Edit component**  to open **Edit component** window and specify symbol size, inputs, outputs, component variables, and clock (optional):



**Execution.** A principle of DLL code execution is shown on the following diagram:



A voltage on the input pin of the component **in** is measured by a “virtual” voltmeter and is assigned to the corresponding element of the array `data` of `double` type. The pointer to the array is passed as a parameter to the DLL function `main`. The function is executed; a new value of output **out** is evaluated and assigned to the corresponding element of the array `data`. A voltage equal to that value is set to a “virtual” grounded voltage source, connected to the corresponding output pin **out**.

Array `double data[]` is used to pass input/output values to/from DLL function. Array size is `<number of inputs> + <number of outputs>`. When calling DLL function, first `<number of inputs>` elements of the array are set to input values in the same order as input pins are shown on the component symbol (left

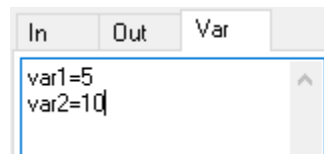
side, top-to-bottom). Output values calculated in the DLL are placed in the next *<number of outputs>* elements of the array in the same order as output pins are shown on the component symbol (right side, top-to-bottom). For example, if the component has two inputs and two outputs, the following code will assign minimum input value to the first output, and maximum input value to the second output:

```
data[2] = min(data[0], data[1]);
data[3] = max(data[0], data[1]);
```

**Component variables.** Any variables created in the DLL code are not accessible by NL5. DLL **component variables** are used for storing values that are accessible both from DLL and NL5.

Component variables are instantiated in NL5 and can be accessed and modified both by NL5 and DLL. Component variables can be used, for example, for debugging purposes: they can be displayed on the graph as a trace without passing them through output pins of the component. Component variables can be changed by NL5 script or command line when simulation is paused. Also, values of component variables will be saved as initial conditions in the **IC** parameter of the component.

Input, output, and component variable names are defined in the In, Out, and Var tabs. You can also assign initial values (**IC**) to outputs and component variables in the form `name=value`:



If component variables are used, the size of the `data[]` array should be *<number of inputs> + <number of outputs> + <number of component variables>*. When calling DLL functions, corresponding elements of the array (following output values) are filled in with current values of component variables. DLL can modify component variables by assigning new values to corresponding elements of the array.

For example, if DLL component with 2 inputs and 2 outputs has a component variable `counter`, it can be modified by NL5 using following notation:

```
component_name.counter = 0;
```

DLL, in turn, can modify value of `counter` as:

```
data[4] = counter;
```

Variable `counter` can be displayed on the graph as a trace of **Function** type using the following notation:

```
component_name.counter
```

When simulation is already started and paused, open **Edit component** window for DLL component and click **DLL variables** button. DLL component variables and their current values will be shown (in floating point format). Those values can be edited and will be sent to the DLL when pressed **OK**.

**Initialization function (Init parameter).** The function should be declared as follows:

```
extern "C" __declspec (dllexport) int NAME(double t, double* data);
```

where:

*NAME* – function name, for instance `init`  
*t* – current simulation time (always = 0)  
*data* – pointer to the array with input/output data

The function returns zero if no errors occur, or user-defined non-zero integer error code. If an error occurs, the error code will be displayed in the NL5 error message window.

Initialization function is executed once at the beginning of simulation at  $t=0$ . The function can be used to assign initial (default) values to outputs and component variables (if defined) by setting corresponding elements of the array `data[]`, declare and initialize static DLL variables and arrays, allocate additional memory which will be used by DLL, open data logging files, etc.

This function is optional and can be omitted. Leave **Init** parameter blank if function is not used.

**Main function (Main parameter).** The function should be declared as follows:

```
extern "C" __declspec (dllexport) int NAME(double t, double* data);
```

where:

*NAME* – function name, for instance `main`  
*t* – current simulation time  
*data* – pointer to the array with input/output data

The function returns zero if no errors occur, or user-defined non-zero integer error code. If an error occurs, the error code will be displayed in the NL5 error message window.

The function calculates output variables using current values of input variables during transient simulation. If **clock** pin does not exist, the function is executed on every calculation step. If **clock** pin exists, the function is executed only on rising/falling edge of logical clock signal.

For example, function `main` below calculates sum of 8 inputs:

```
extern "C" __declspec (dllexport) int main(double t, double* x)
{
    double y=0;
    for(int i=0; i<8; ++i) y += x[i];
    x[8] = y;
    return 0;
}
```

This function is optional and can be omitted. Leave **Main** parameter blank if function is not used.

**Pause function (Pause parameter).** The function should be declared as follows:

```
extern "C" __declspec (dllexport) void NAME();
```

where:

*NAME* – function name, for instance `on_pause`

The function is called when transient simulation is paused. Use this function to perform operations such as saving data into the file, reallocate unused memory, etc.

This function is optional and can be omitted. Leave **Pause** parameter blank if function is not used.

**Exit function (Exit parameter).** The function should be declared as follows:

```
extern "C" __declspec (dllexport) void NAME();
```


where:

*NAME* – function name, for instance `on_exit`

The function is called when DLL is being closed and DLL component destroyed. Use this function to free the memory allocated by other DLL functions, close files used for data logging, etc.

This function is optional and can be omitted. Leave **Exit** parameter blank if function is not used.

**Using one DLL by several components.** Several DLL components may use the same DLL file. Be aware that in this case only one copy of DLL is loaded into the memory, so that the same static variables will be shared by all components. In this case, **Init**, **Pause**, and **Exit** functions may be called only by one component. Also, different components can use different functions from the same DLL.



If you need several components to be using the same DLL with separate static variables, use several copies of the DLL file with different names. Another solution would be **attaching**  DLL to the component. If DLL is attached, NL5 will create a unique temporary DLL file for each component during simulation, so that each component will be using its own DLL copy.

**Initial conditions (IC parameter).** Initial conditions is a string with comma-separated expressions in “name=value” format assigning initial values to outputs and component variables of the DLL. For example:

```
min=-1.23,max=4.56,counter=100
```

where `min` and `max` are outputs, and `counter` is component variable.

**IC** string will be automatically modified by the following commands:

- **Save initial conditions**  : fills in the string with current variable values.
- **Schematic Settings and tools** , **Initial Conditions** tab, check box **Clear C-code and DLL IC**: clears this string.

If not empty, **IC** will be executed right before calling DLL initialization function, so that DLL initialization function could use parameters defined in the **IC** string.

## Roots model

**Roots** model describes zeroes and poles of transfer function of **s-function** component:

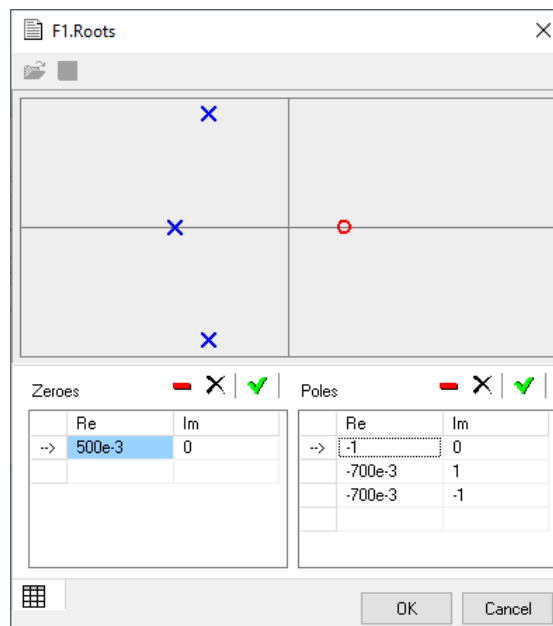
$$f(s) = \mathbf{K} * (s-z_1)*(s-z_2)... / (s-p_1)/(s-p_2)...$$

and impedance of **Impedance** component.

$$z(s) = \mathbf{Z} * (s-z_1)*(s-z_2)... / (s-p_1)/(s-p_2)...$$

**where**  $z_n$  are zeroes,  $p_n$  are poles.

Zeros and poles are entered in the **Roots** window:







To enter new zero/pole, select the last (empty) table cell, **Re** column, enter the value, and press **Enter**, or click to update the table. Then enter **Im** value if needed. If non-zero **Im** value is entered, a conjugate zero/pole is added automatically.

To edit existing zero/pole, select the cell, enter new value, and press **Enter**, or click to update the table. Click to remove selected zero/pole, marked by --> in the first column, or click to clear the table.



## Attachments


Any files/data used by NL5 components (**File**, **SubCir**, **DLL**, etc.) can be **attached** to the NL5 schematic file. Then, the schematic file will contain all the data needed for simulation, which makes it easy to maintain, encrypt, and distribute.

To attach the data to the component, click **Attach**  on the **Component window** toolbar, or **right-click** on the component, select **Component** , then **Attach** . To remove attachment click **Remove attachment** .



Attached data/schematic can be reviewed and edited by clicking **Open attachment/subcircuit** .

If the attached subcircuit of **SubCir** model is opened and edited, it can be saved back to the attachment only by **Save / Save attachment** command, so that original subcircuit file will not be changed. If the same subcircuit is attached to several components, it can be modified for each component separately.

Please note that attaching **DLL** may result in large size of the schematic file, since binary data of the dll file are converted into text format.

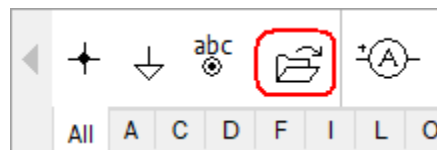
Current attachments of all components of the schematic can be reviewed and managed in the **Schematic Settings and tools**  window, **Attachments** tab.

## NL5 components (\*.nlc)

NL5 component can be saved into a **component file**: schematic file with `nlc` extension. Then the component can be loaded from the file and placed onto schematic as a standard component. For components with external data (**SubCir** and **File** models, **DLL**), attach the data to the component and save it in the component file: then the **component file** can be easily distributed without additional files. The component can also be encrypted: **right-click** on the component, select **Component** , then **Encrypt** ).




To save a component into **component file**, **right-click** on the component, select **Component** , then **Save component** .

To place component onto schematic, click **Load component** in the **Components bar**:



## Customized component

Some components are **customized**, so that their symbol and/or number of pins can be edited by the user. To edit component symbol, open **Edit component** window:

- **Right-click** on the component, select **Component** , then **Edit component**  (if available).  
**OR**
- Click **Edit component**  on the **Components window** toolbar.



The following component properties can be edited for all customized components:


- Symbol width.
- Symbol height.
- Number of pins (inputs and/or outputs).
- Pins location.

The following properties can be edited for specific component types:

- Pin names.
- Subcircuit label names.
- Clock signal, inversion of clock signal.
- Inversion of the output.
- Logical function.
- Number of windings.
- Beginning and end of windings.
- DLL component variables.

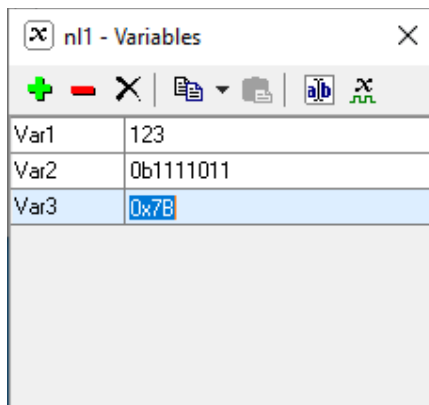
## Encrypted component


Any component in the schematic can be **encrypted**: **right-click** on the component, select **Component** , then **Encrypt** . The model and parameters of encrypted component are non-visible and non-editable, so that it can be distributed to other users with guarantee that its content will remain unknown.

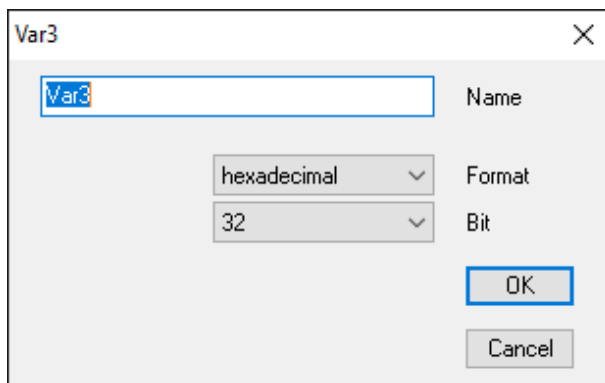
Encrypted component can be decrypted back until schematic file is closed: **right-click** on the component, select **Component** , then **Decrypt** . Once schematic with encrypted component is saved and closed, the component cannot be decrypted: this is done to maximize security of encryption.

## Schematic variables



Schematic variable is a floating point variable which belongs to the schematic. The variable and its value can be defined in the **Variables window**:

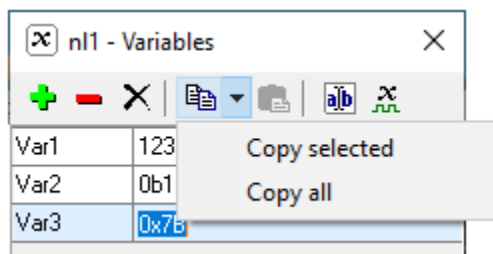


Click **Rename**  button to rename the variable or specify its format:



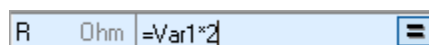
**Format** (double, integer, binary, octal, hexadecimal) specifies how the variable is displayed. Binary, octal and hexadecimal also have **Bit** settings. The variable value can be entered in any format, and then it will be displayed in specified format.

Click **Copy**  to copy variable to the clipboard, then click **Paste**  to paste it to the same or another schematic. To copy all variables, click drop-down arrow next to the **Copy** button, then select **Copy all**:




Please note that variables should be copied/pasted separately from schematic components.

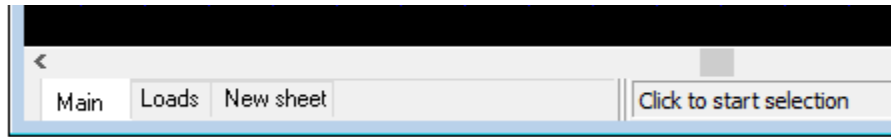
Variables can be used in formulas, functions, script, C-code, etc. For example:





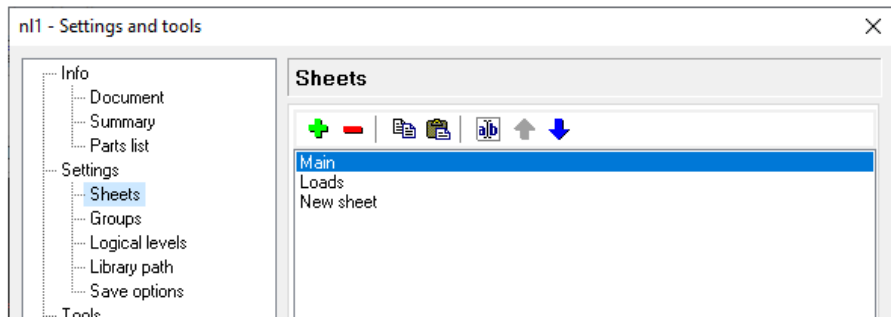
## Schematic sheets

Schematic may contain several **Sheets** . Electrical connection between schematic sheets can be done through labels and functions.


Existing sheets are shown in the **Sheet tab** of **Schematic window**:



**Right-click** on the tab to access sheets-related commands in the pop-up context menu, or click **Sheets**  to manage sheets in the **Schematic Settings and tools** , **Sheets** tab:

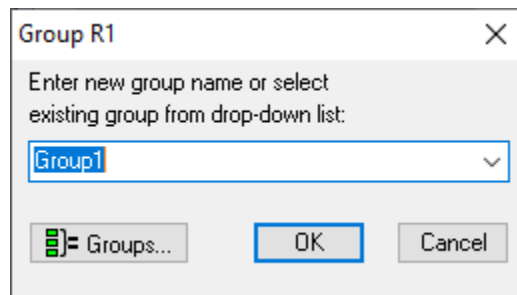




## Schematic groups

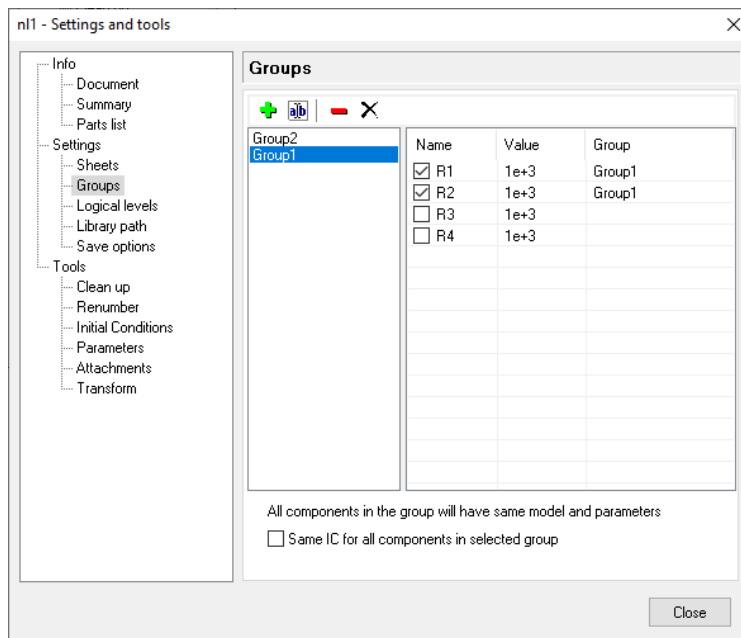
**Group**  is a set of components that always have the same model and parameter values. When the model or parameter of any component in the group changes, all other components are changed automatically. An exception is IC-type (Initial Condition) parameter (or several IC parameters), which can be different for components in the group, if not specified otherwise.

Only components of the same type can be assigned to the same group. If component is assigned to the group, the group name is shown in the **Group** column of **Components window**.

To assign selected component to a new or existing group, click **Group**  in the **Components window**:




Click  Groups... to manage existing groups, add new group, and assign components to the group in the **Schematic Settings and tools** , **Groups** tab:

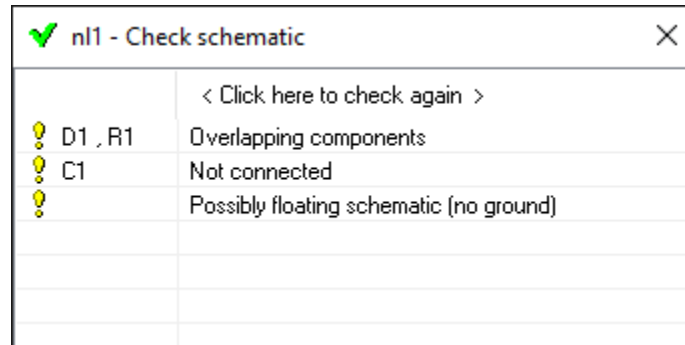


If **Same IC for all components in selected group** is checked, Initial Condition parameter(s) for all components in that group will always be the same, otherwise they can be different.

Deleting the group does not affect components assigned to that group.


## Check schematic

**Check schematic**  command performs extensive check of the schematic for potential problems, and reports results in the window:



Also, the check function removes unnecessary connection points and adds connection points where needed (three wires).

Click on the row to select (highlight) a schematic element(s) for which the problem was detected.

Go to **Preferences**  , **Warnings** tab to select/unselect reported issues.



**Right-click** to access check-related commands in the pop-up context menu.

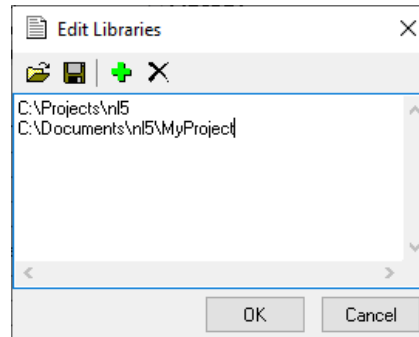
Schematic check is automatically performed at **Transient** and **AC** simulation start. The problems reported by the function do not prevent from running simulation, and typically do not affect simulation results. However, they could indicate not accurate schematic design, and should be fixed if possible.


## Schematic Settings and tools

Here are details on some **Schematic Settings and tools**  tabs.

**Logical levels.** Specify logical levels and threshold for all components in the schematic.

**Library path.** Specify library paths list for component files (**DLL**, **SubCir**, **File** model, etc.). Click **Edit** , then enter library paths manually, one path per line, or click  to select from folders list:



When new schematic is created, a default list is copied from **Preferences** , **Library** tab. Library path could be full path, or path relative to the directory where schematic file is located.

If specified file name of the component does not have a full path, NL5 will search for the file in the following order:

- In the directory where schematic file is located.
- In the directories specified in the **Library path** list.
- In the directories specified in the **Library path** list relative to the directory of schematic file.

**Ver.2.** If schematic was converted from NL5 Ver.2 format, this tab will show conversion notes: for example, what component types/models were renamed, replaced, removed, etc. For schematic in NL5 Ver.3 format, this tab is not visible.

**Clean Up.** Clean up schematic, select what operations to perform.

**Renumber.** Renumber components on the schematic. Select **Order** and **Names** option, see example of renumbering in the **Example** window. **Numerical** name is the name that starts with a letter, and the rest of the name is a number, otherwise the name is considered as **Text**.

**Initial Conditions.** Set Initial Conditions (**IC**) of selected component types to specified value.

**Parameters.** Set selected parameters of all components to specified value. Select parameters, enter new parameter value, or select from drop-down list.

**Set period.** Set **Period** parameter of all **Sin** and **Pulse** voltage and current sources, and **Pulse** switches. **Width**, **Rise**, **Fall**, and **Delay** values will be adjusted proportionally to period.

**Attachments.** The table shows components with **File**, **SubCir**, and **DLL** models, the file name, and whether the file is currently attached (checked) or not attached (unchecked). Change checkbox states as needed to attach files or remove attachments. If **When removing attachment, save to file** option is selected, you will be asked for a file name.

### Transform.

**Frequency.** Transform frequency response of the schematic by changing R, C, and L values. Enter f1 and f2 values, or place cursors at f1 and f2 frequencies on the AC graph (active cursor at f2). Select one of the following options:

- **R = const.** C and L will change as follows:  $C=C*f1/f2$ ,  $L=L*f1/f2$
- **C = const.** R and L will change as follows:  $R=R*f1/f2$ ,  $L=L*(f1/f2)^2$
- **L = const.** R and C will change as follows:  $R=R*f2/f1$ ,  $C=C*(f1/f2)^2$

**Impedance.** Transform schematic impedance from r1 to r2 by changing R, C, and L values. Enter r1 and r2 values, then click **Execute** to proceed. R, C, and L values will change as follows:

$$R=R*r2/r1$$

$$C=C*r1/r2$$

$$L=L*r2/r1$$



# IV. Transient

## Algorithm

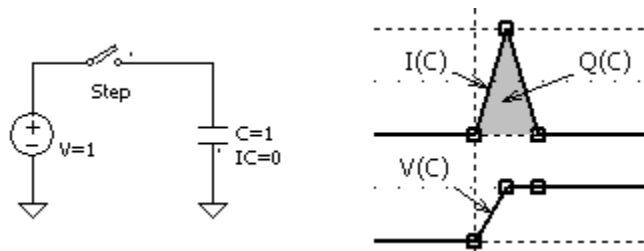
NL5 performs simulation with ideal and piecewise linear (PWL) components. For instance, an ideal diode is either short circuit (zero resistance), or open circuit (infinite resistance), so that its PWL representation consists of just two linear segments. As long as all of the components are staying within their current segment, the circuit is described by the system of linear differential equations. The system is modified only at the moments when at least one component changes its linear segment.

**Start.** When simulation starts at  $t=0$ , a **Direct Current (DC) operating point** is calculated first. Some of the components may have specified **Initial Condition (IC)**: these ICs will be used during DC operating point calculation. If circuit has more than one steady state, it can be set to a desired state by defining proper ICs. After DC operating point is found, simulation continues as a sequence of **linear segment** simulations, with **switching** between them.

**Linear segment.** In the linear segment, the circuit is described by the system of linear differential equations, which is solved by Trapezoidal integration method. The method provides sufficient accuracy with good robustness and calculation speed. During linear segment simulation, the algorithm is performing “switching point detection”: checking conditions on all components that may change their state or linear segment.

**Switching.** When switching point is detected, the current linear range ends, **switching algorithm** performs instantaneous switching, and a new linear segment starts. Switching algorithm can be optimized for specific circuit by selecting from several options at **Transient settings/Advanced settings**.

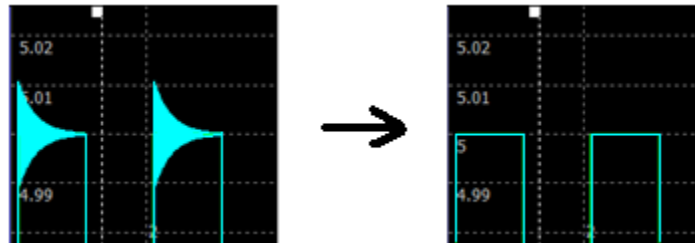
Instantaneous switching of ideal components may produce infinite voltage or current pulses. For example, when capacitor is connected to voltage source through ideal switch, an infinitely short current pulse with infinite amplitude may occur. The area (integral over time) of such pulse is limited and is equal to the total charge coming to or out of the capacitor during switching. Similar situations may occur when current through the inductor is discontinued, which results in an infinite voltage pulse across the inductor. The integral of the voltage over time corresponds to a magnetic flux in the inductor. Such a pulse is known as Dirac pulse, or delta-function. In NL5, the current or voltage delta-function is shown as a triangle pulse with the duration of each slope equal to minimal calculation step used at that moment, and the area satisfies charge or magnetic flux conservation law.



**Simulation step.** Unlike many analog simulators, NL5 does not perform automatic step control. Selecting simulation step is the user's responsibility. This gives the user full control of simulation, although requires certain experience and understanding of the process. The rule of thumb is keeping simulation step below smallest time constant in the circuit, otherwise the integration method may get

unstable, and produce “numerical oscillations” when signal is “jumping” up and down at each simulation step.

Very often, such oscillation caused by non-optimal step would not affect overall functionality of the circuit and can be ignored. To get rid of that type of oscillation, reduce simulation step to reasonably small value, or use **Suppress oscillations** option (**Transient settings/Advanced settings/Transient/Simulation step**). Here is an example of transient improvement with this option:



Although simulation step is specified by user, NL5 still can **automatically reduce the step** to satisfy the following conditions:

- Period of sine source contains at least 16 steps.
- Pulse or switch On/Off state contains at least 4 steps.
- Non-zero rising or falling edge contains at least 4 steps.
- Interval between two points of the source component, interpolating the signal between two points, contains at least 4 steps.
- Delay time of transmission line and “delay” component contains at least 2 steps.

Also, simulation step can be reduced to detect switching point and perform accurate switching.

Automatic step reduction can be disabled by selecting **Constant step** option (**Transient settings/Advanced settings/Transient/Simulation step**).

**Traces.** During simulation, NL5 only stores data for specified **traces**. Pause or stop simulation to add/remove traces, then continue or start simulation. A special type of trace is **Math** (post-processing). That trace is calculated from other traces only when simulation is paused or stopped.

**Memory.** Simulation data is stored in the operating memory. The memory is allocated as needed by relatively small blocks. If available operating memory is not enough for all the data, some of the blocks currently storing the very beginning of the trace will be reassigned to a new data, and thus some traces may be “truncated” at the beginning. The first time this happens, the warning message will be displayed in the status bar of **Transient window**.

By default, NL5 saves all calculated data for all traces. To reduce memory consumption, a simple data compression method can be used: if several data points in a row have the same amplitude, only first and last point of such sequence are being stored. To use this method, select **Data compression** option (**Transient settings/Advanced settings/Transient**).

## Running transient



- Go to **Transient settings window** (use drop-down menu to select desired Settings tab) to set up simulation parameters, traces, etc.



- **Start** transient (available on the **Schematic** and **Transient window** toolbars). When simulation is running, schematic cannot be edited, and component parameters/model cannot be changed. **Stop** or **Pause** simulation to make any changes.



- **Pause** transient (or press **Space**).




- **Continue** transient (or press **Space**).



- **Stop** transient. Clear extra memory allocated for simulation. Simulation cannot be continued.

### Transient modes:



**Single**: simulation paused at the right edge of the screen. Please note that simulation step is not adjusted to stop exactly at the screen edge, so the last calculated point could have time greater than screen edge. Press **Space** or click **Continue**  to continue simulation.



**Scope**: when the right edge of the screen is reached, the screen shifts so that its right edge becomes the left, and simulation continues.



**Scroll**: when the right edge of the screen is reached, the screen is scrolled to the left by some small amount (about 1%), and simulation continues.



**Zoom**: when the right edge of the screen is reached, screen size increases by some small amount (about 1%) keeping left edge the same, and simulation continues.



**Continuous**: when the right edge of the screen is reached, simulation continues further without changing the screen. Use this mode to view simulation results at specific time, while simulation is running.



- **Save IC**. Stop simulation, move component states to component initial condition (**IC**). Also, a **Delay** parameter of all sources will change accordingly to provide specific starting phase of the signal. All saved states will be deleted.

Please note that not all components/models provide initial conditions for all possible states. For instance, some models of controlled sources and switches (**One-shot**, **Pulse**, **Sin**, **Clock**, etc.) will always start being in inactive state.

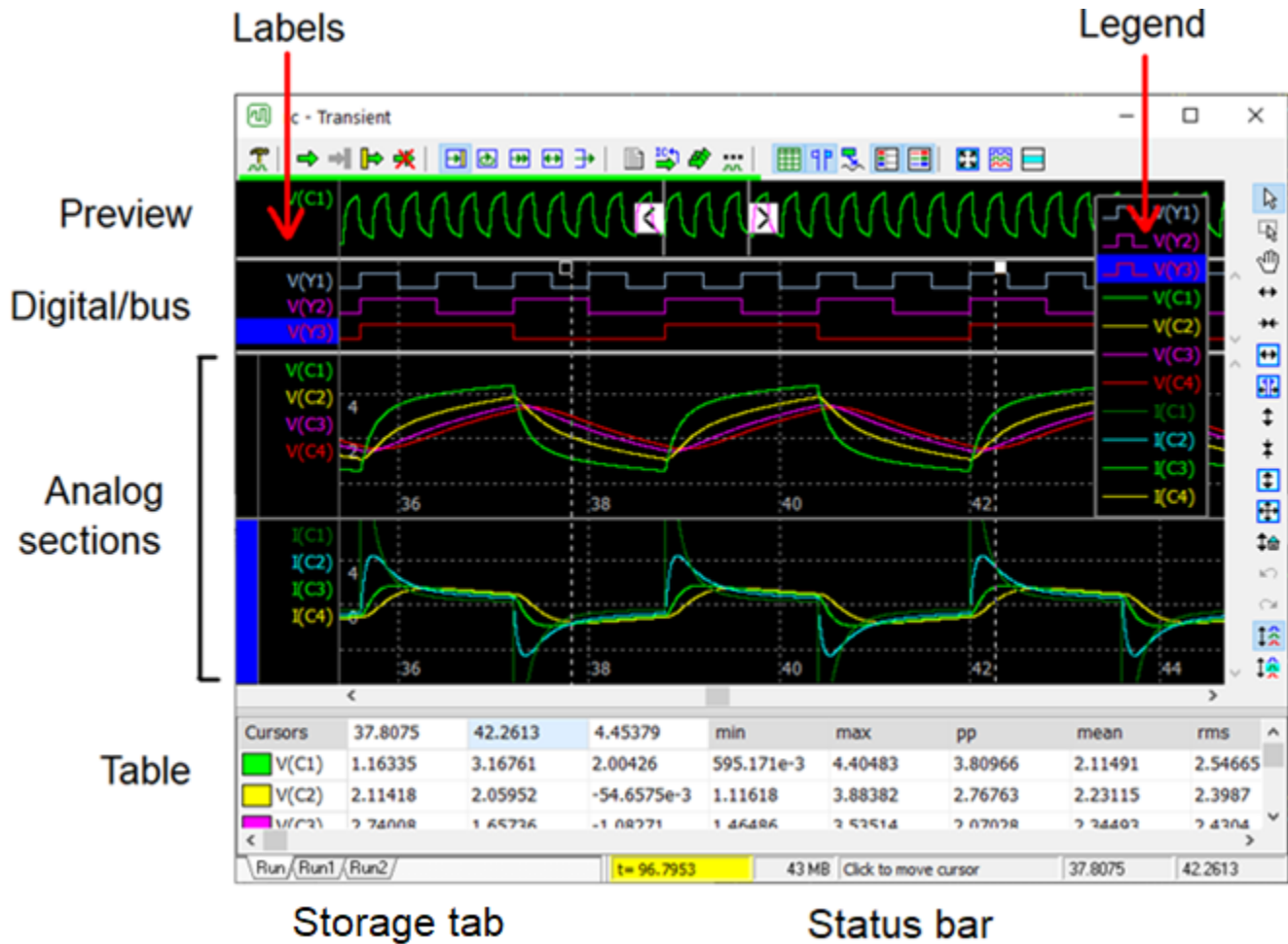


- **Sweep parameter**. Open **Tools** window, **Sweep** tab. See **Tools** chapter for details.








- **Show transient log**.



## Transient window



Add transient traces using any of the following methods:

- Go to **Settings window** , **Add traces**  tab, see detailed instruction below.
- **Right-click** on the component in the schematic window, select **Add trace** , then select trace type.
- Select component in the **Components window**, click **Add trace**  button.
- Select variable in the **Variables window**, click **Add trace**  button.

Go to **Settings window** , **Format traces**  tab to select trace type, color, scales, etc.


Go to **Settings window** , **Screen**  tab to configure **Transient window** sections, scales, and other properties.

There are 3 types of **sections** displaying transient data: **Preview**, **Digital**, and **Analog**.

**Preview** section shows **all data** of one selected trace and can be used to select desired horizontal range for analog and digital sections. The trace is always auto-scaled horizontally and vertically.



- Move **preview cursors** by mouse (just one cursor or both simultaneously) or click on the section to select screen horizontal range.
- Use **mouse-wheel** to change screen horizontal range.
- **Right-click** on the section to select another preview trace or hide the section.
- Use **split bar** below the section to change its height.

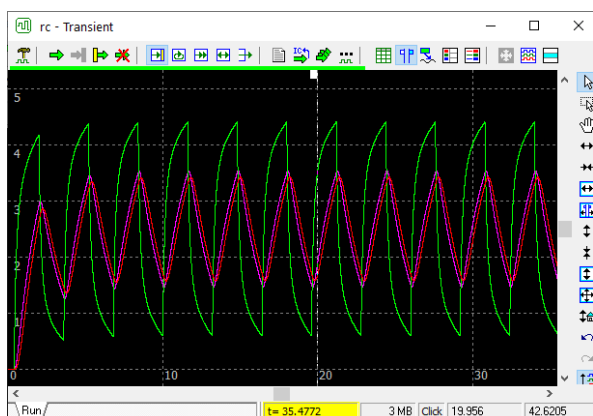
**Digital** section contains all traces of **Digital** and **Bus** type.

- Use **split bar** below the section to adjust section height manually.
- **Double-click** the split bar to set the height so that all traces in the section are visible.
- Go to **Preferences** , **Graphs, Digital/Bus** tab to configure size and format of digital/bus traces.

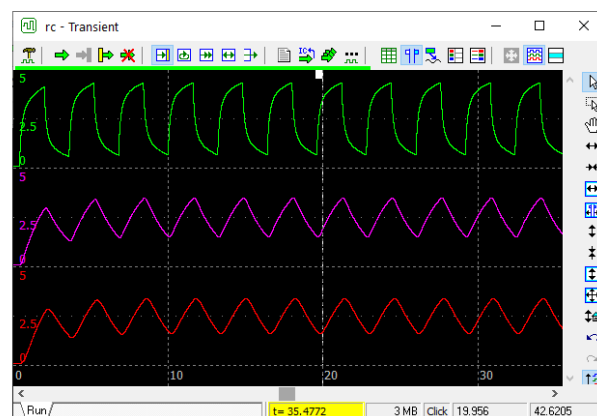
**Analog sections** contain traces of **Analog** type. Every analog trace can be assigned either to **Digital** or one of **Analog** section. Having just one analog section could be sufficient for most applications. Multiple analog sections (up to 4) may be used to combine several traces with similar properties (function, scale, etc.): for example, one section for voltage traces, and another section for current traces.

Some useful operations on **Analog** sections:


- **Maximize analog section**  - temporarily maximized selected analog section to the full window size.
- **Separate traces**  mode (press **Tab**) - show each trace in its own smaller area separated by horizontal lines:





Separate traces Off



Separate traces On

- **Show selected background**  - use highlighted background of the selected traces. This feature is applicable for digital/bus traces, and analog traces in **Separate traces** mode.
- Use **split bars** between analog sections to adjust section height.
- Vertical scaling operations and vertical scrollbar apply to currently active **Analog** section.

**Vertical scale.** Each trace in the analog section is shown with its individual **Scale** and **Mid** values, specified at **Settings window** , **Format traces**  tab. Vertical scale and gridlines are shown for selected trace (or several selected traces if their **Scale** and **Mid** values are the same). If several traces with different **Scale** and **Mid** values are selected, vertical scale is not shown.

Depending on selected **vertical mode**, all vertical zoom/scroll/reset operations (when buttons, keys, mouse, mouse wheel) will apply to:



- **All traces** of the active analog section.



- **Selected traces** of the active analog section.




- **Screen.** In this mode, zoom and offset apply to the whole screen image, while scales and offsets of traces remain unchanged. This mode is similar to NL5 Ver.2 functionality. Please note that when switching back to either of the first two modes, screen zoom and offset are reset to default, and the image of the graph may change.

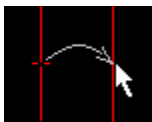
Click **Reset vertical scale**  to set **Scale** and **Mid** of all or selected traces in active analog section to default values.

**Mouse.** There are 3 modes of **mouse operation** in analog sections:

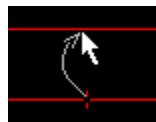
**Cursors**  mode. Some mouse operations apply to transient cursors :

- **Double-click** the screen to show cursors and/or reset cursors lock mode.
- **Click** on the screen to move the closest cursor to that point.
- **Place mouse pointer** over the cursor, then click and drag to move the cursor.
- Press **Left/Right** arrow key with **Ctrl**, **Shift**, or **Ctrl-Shift** keys to move active cursor with different step resolution.

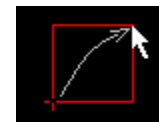
**Zoom**  mode (press and hold **Ctrl** key to switch to **zoom** mode temporarily). Click and drag to select horizontal, vertical, or rectangle area to be zoomed-in:




Move mouse pointer horizontally to select horizontal area.



Move mouse pointer vertically to select vertical area.



Move mouse pointer diagonally to select rectangle area.

**Scrolling**  mode (press and hold **Shift** key to switch to **scrolling** mode temporarily). Click and drag to scroll the screen.

**Mouse wheel:**


**Mouse wheel** – horizontal zoom (relative to mouse pointer position).


**Shift-mouse wheel** – horizontal scroll.


**Ctrl-mouse wheel** – vertical zoom (relative to mouse pointer position).

**Ctrl-Shift-mouse wheel** – vertical scroll.


**Keys and shortcuts:**

**F5** – show transient window .

**F6** - start transient .


**F7** - stop transient .

**Space** – pause/continue transient.

**Tab** - separate traces in active analog section .

**Up, Down** – vertical scroll.

**PgUp, PgDn** – vertical zoom , .



**Home** – vertical fit the screen .


**Left, Right** – horizontal scroll.

**End** – center beginning of traces (set to the middle of the screen).


**Ctrl-End** – center end of traces.

**Shift-End** – center middle of the traces.

**Ctrl-PgUp, Ctrl-PgDn** – horizontal zoom , .

**Ctrl-Home** – horizontal fit the screen .

**Shift-PgUp, Shift-PgDn** – vertical and horizontal zoom.

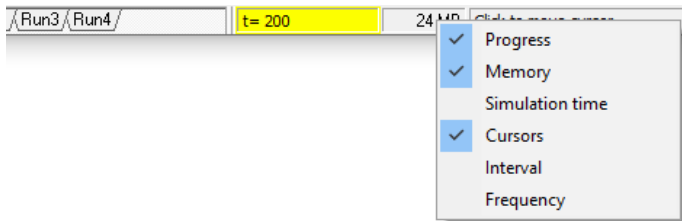
**Shift-Home** – fit the screen .

**Ctrl-Right, Ctrl-Left, Shift-Right, Shift-Left, Ctrl-Shift-Right, Ctrl-Shift-Left** - move selected cursor with different resolution.



## Status bar





Right-click status bar to configure:




- **Progress** – current transient time, simulation status (running, paused).
- **Memory** – amount of memory allocated for calculation and data.
- **Simulation time** –simulation (processor) time.
- **Cursors** – show cursors position (left and right).
- **Interval** – interval between cursors.
- **Frequency** = 1 / Interval.


## Cursors

**Cursors**  specify interval that can be used for **Transient table** and **Transient tools**.


- Go to **Cursors**  tab of **Settings window**  to configure cursors options and perform some operations on cursors.
- Click **Show cursors**  to show/hide cursors. Please note that table and transient tools may use specified **Cursors** interval type even when cursors are hidden.
- Click **Cursors to screen**  to set horizontal scale to cursors position.



## Labels

**Labels**  are trace names shown in the area on the left side of the graph. Highlighted background of the label indicates selected traces.

- Click **Show labels**  to show/hide labels.
- **Click** on the label to select one trace.
- **Ctrl-click** to add trace to selection or unselect the trace.
- **Shift-click** to select a range of traces.
- **Right-click** on the label to see labels-related context menu.

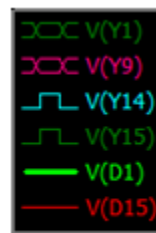
## Legend

**Legend**  shows list of traces currently displayed on the graph. Legend is also available in all **Transient tools**.

- Click **Show legend**  to show/hide legend.
  - **Click** on the border of the legend and drag to move the legend window.
  - **Click** on the trace in the legend to select one trace.
  - **Ctrl-click** to add trace to selection or unselect the trace.
  - **Shift-click** to select a range of traces.
  - **Right-click** to see context menu with more selection options.
- Go to **Preferences** , **Legend** tab to select legend style (colors, font). The style can be defined manually or selected from two predefined options (**NL5 Ver.2** and **NL5 Ver.3**), and then adjusted manually if needed:




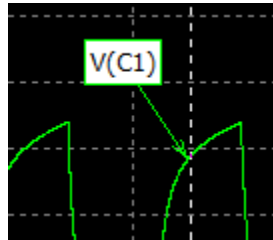
NL5 Ver.2








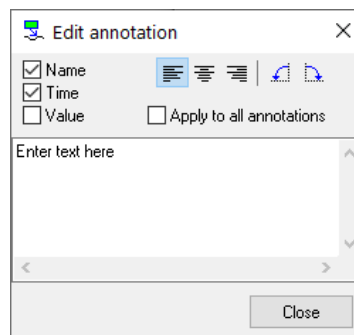
NL5 Ver.3

## Annotations

**Annotation**  is a text with a pointer, which always points to the same data point of a trace, even when graph is zoomed or scrolled:




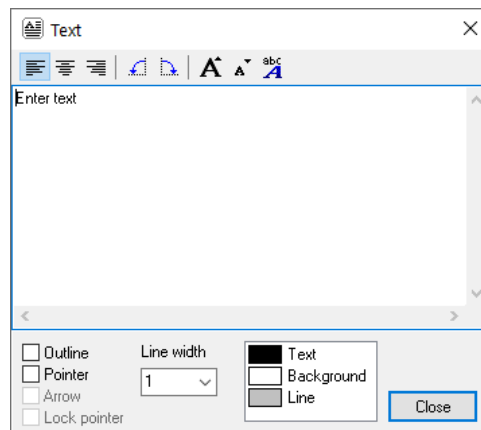
- To add annotation, set cursor to the time point where annotation is needed, **right-click** the graph, select **Annotate** , then select **Selected traces**  or **All traces** .
- Go to **Preferences** , **Annotation** tab to configure annotation colors, font, and some other common properties.
- **Double-click** on the annotation, or **right-click** the annotation and select **Edit annotation**  to edit (customize) specific annotation:






- **Click** and drag annotation text to move the text while keeping pointer at the same data point.
- **Click** and drag the end of the pointer to move annotation pointer to a different data point. The pointer will follow the mouse pointer, staying at the nearest point of the trace (trace data between data points are linearly interpolated).
- Annotations defined on the transient graph can also be displayed on the **XY diagram**, **Eye diagram**, and **Amplitude histogram** Transient tools.

## Text


**Right-click** on the graph, select **Insert text** :









If graph is zoomed or scrolled, the text stays at the same place, anchored to the left-top corner of the graph window.


- **Double-click** on the text, or **right-click** on the text and select **Edit text**  to edit or configure the text.
- **Click** and drag the text window to move it. If pointer is locked, only text will move.
- **Click** and drag the end of the pointer to move the pointer only.
- **Right-click** on the text and select **Copy text**  to copy text to clipboard. Then **Right-click** on the graph and select **Paste text**  to paste text on the graph.

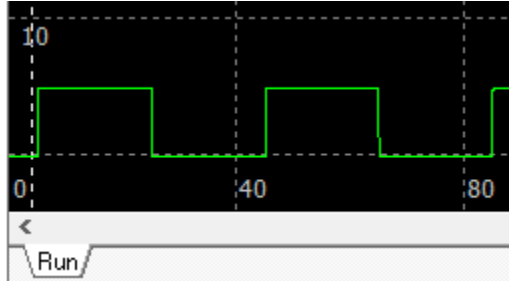
## Table

**Table**  shows cursors position, trace values, and some characteristics of the traces calculated in specified interval (**Cursors**, **Screen**, **All data**, **Selection**).

- Go to of **Settings window** , **Table**  tab to configure the table.
- Click **Show table**  to show/hide the table.
- Click **Floating table**  to show the table in a floating window. Click  and  any time to switch between window modes.
- **Right-click** the table to see context menu with table-related commands.
- **Right-click** / **Copy** to copy content of the table cell to the clipboard.

## Storage

**Storage**  can be used to keep simulation data for further analysis and comparison. When simulation is performed, transient data are shown on the graph, and a tab named **Run** corresponding to current transient data is shown in the **Storage tab**:




When a new simulation is started, current simulation data is deleted. To keep it for further analysis and comparison, it can be saved into **storage**.

**Right-click** the graph or **Storage tab**, and select **Move Run to storage** , then enter storage name:

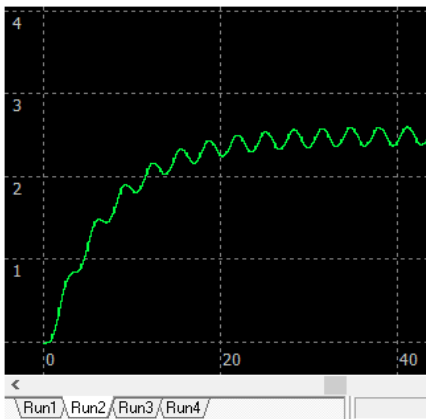
Storage

Move run to storage

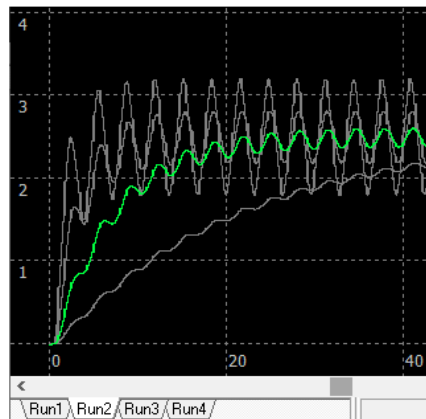
Enter storage name:

Simulation data will be moved to the storage, and a tab corresponding to the stored data will appear in the **Storage tab**. Also, storage is used for **Sweep parameter**  operation.

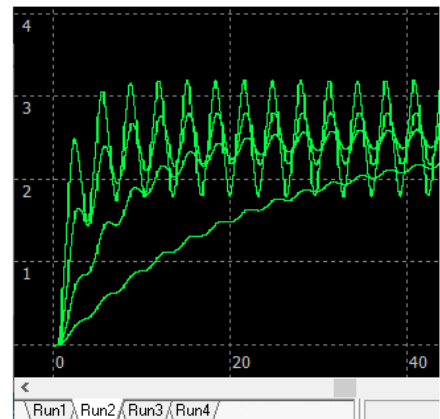
Stored data can be displayed in different ways, for example:



Selected only



Selected and dimmed

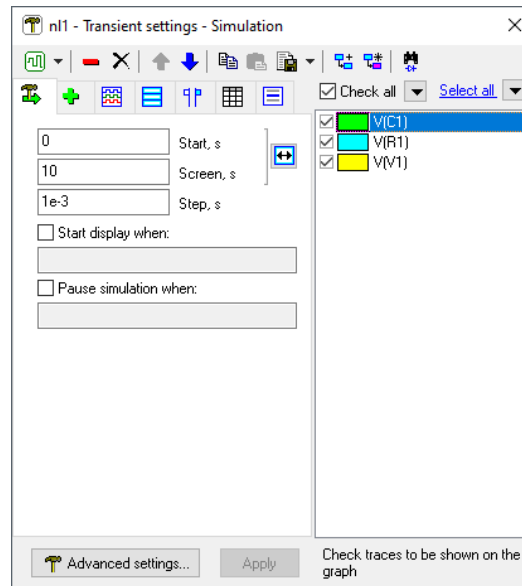



All

- Go to **Storage**  tab of **Settings window**  to configure storage.
- **Right-click** the graph or **Storage tab** for storage-related commands.

## Transient settings

### Simulation



**Start, Screen.** Screen settings at simulation start (simulation itself always starts at  $t=0$ ). Click  to apply current screen settings to **Start** and **Screen** parameters.

**Start display when.** Select this option and enter condition to start display when condition is met (at time greater than **Start**). For example:

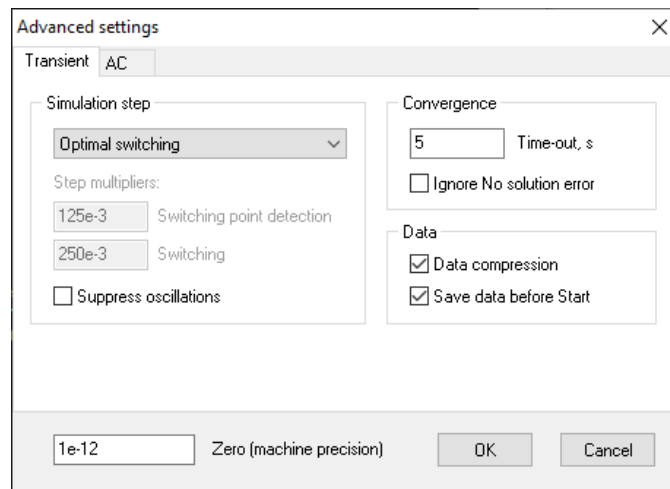
$$V(C1) > 10.0$$

**Pause simulation when.** Select this option and enter condition to pause simulation when condition is met. For example:

$$P(R1) > 1.0$$

If simulation paused at specified condition and then continued, it will be performed until condition is not met, and then paused when condition is met again.

**Advanced settings**  - more algorithm, transient, and data settings.



**Simulation step.** Select simulation step control mode:

- **Constant step** – step will never be reduced by NL5 and will always be as defined by the user. This option provides the fastest simulation, however switching may not be reliable.
- **Fast switching** - fast simulation with acceptable switching convergence for most circuits.
- **Optimal switching** - good convergence and performance for most circuits.
- **Accurate switching** - best convergence, however simulation time might be longer.
- **Custom** – use your own **Step multipliers**:

**Switching point detection** - current step can be reduced by this coefficient to detect switching point more accurately.

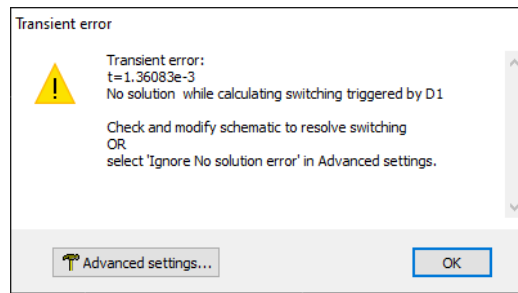
**Switching** – after switching occurred, the current step will be reduced by this coefficient for at least two simulation steps to provide more accurate results. Also, this will define the displayed amplitude of possible “infinite” voltage or current pulses for simulation with ideal components. This value is recommended to be at least two times greater than **Switching point detection** value.

**Suppress oscillations.** Reduce oscillations caused by high simulation step. Using this option may increase simulation time.

**Convergence.**

**Time-out** specifies maximum time allowed for calculating one simulation step. If calculation takes longer, simulation is stopped with error message. If time-out occurred due to unresolved switching iterations, the error message will indicate a component that started switching process. Time-out may also occur due to infinite while/do/for loops of C-code. Set **Time-out = 0** to disable time-out detection.

**Ignore No solution error.** Sometimes switching process cannot be resolved, and error message is being displayed, for example:



Such a problem is most likely to happen during switching process in a complex circuit with ideal components, where infinite loop gain, infinite voltage/current, or other extreme (non-realistic) conditions may occur. To resolve the problem:

- First, check the schematic to fix possible general schematic issues.
- Identify critical ideal components which may cause the problem and replace them with more “realistic” models (use finite gain and bandwidth, non-zero and non-infinite resistance, etc.).
- Add small resistors in series, large resistors in parallel, and/or small capacitors in parallel to critical points.

If none of those actions help, select **Ignore ‘No solution’ error** option. With this option enabled, if switching cannot be resolved, NL5 will accept some, probably invalid, solution, and continue transient simulation. Although one or more data points may not be entirely correct, there is a high chance that the solution will converge to the correct one very soon.

## Data.

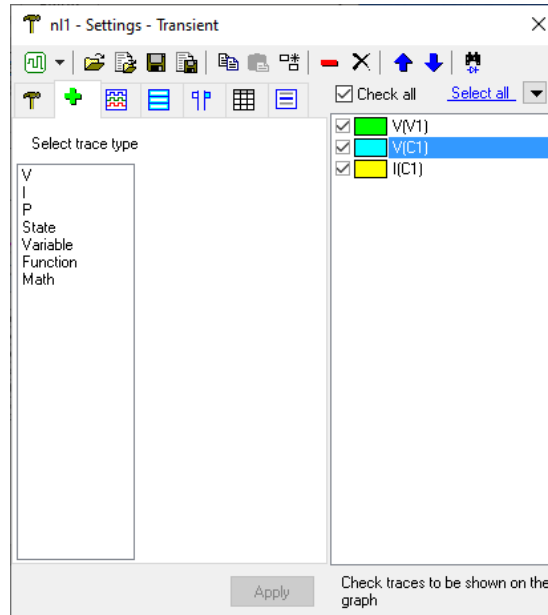
**Data compression.** If several data points in a row have the same amplitude, only first and last point of such sequence will be stored.

**Save data before Start.** Although simulation always starts at  $t=0$ , the data before **Start** may be not stores, to reduce memory consumption.

**Zero (machine precision)** – relative difference between two floating point numbers which can be reliably recognized. Increasing **Zero** may help to improve convergence at switching points, however, for most of circuits changing **Zero** is not recommended.



## Add traces



### V, I, P, State, Variable:

Select one or more components in the list, then click **Add new trace**, or **double-click** on the component in the list.

To add trace in the **Schematic window**, **right-click** on the component, select **Add trace** , then select trace type. **Digital** type is adding **V** trace with trace type set to **Digital**.

To add traces in the **Components window**, select component and click , , or .

To add **Variable** trace in the **Variables window**, select variable and click .

**State** trace may be used to clearly indicate a state of the component instead of looking at current and/or voltage on the component. For example, for diodes and switches, it is 0 if device is in **Off** state (non-conducting), and 1 if device is in **On** state (conducting). For **PWL** models, it indicates **PWL** characteristic piece number which is currently being used.

### Function:

Click **Add new Function trace**, then enter function. The function may consist of arithmetic operators and functions, component parameters, global variables of **C-code** components, names of existing traces, current transient time  $t$ , and V, I, and P on the component (even if V, I, P trace for the component is not defined):

```
sin(t*1000)*(1+cos(t*10))
V(in)/I(A1)
X2.param (value of param variable of C-code component X2)
X2.arr[5] (value of arr[5] of C-code component X2)
```

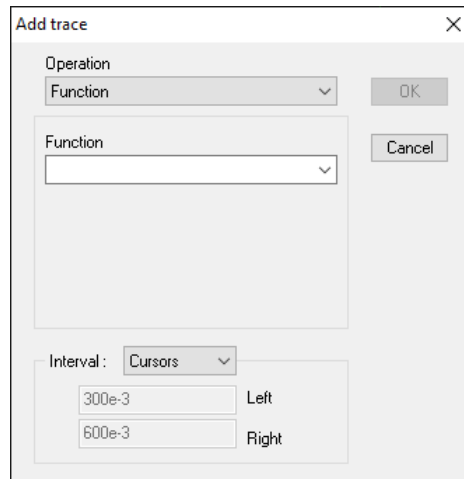
Use function to display V/I/P on the components inside subcircuits, for example:

V(X1.R1) - voltage at R1 in the subcircuit X1.

The function is calculated only during simulation: if function is changed, it will not be immediately recalculated.

## Math:

Click  **Add new Math trace**, then specify **Math** trace :



Select **Operation**, then enter parameters required for that operation (if any).

Specify **Interval** where **Math** trace will be calculated.

Some trace types (**Min, Max, Mean, Regression**) have a parameter **Show over entire range of the Trace**. If enabled, the trace will be calculated in the selected **Interval**, but displayed in the entire range of the trace. This may be used, for example, to see how approximation of the trace data performed between cursors correlates with the rest of the trace.

The **Math** trace is **post-processing** type of trace: it is recalculated automatically when simulation paused or stopped, and if trace type, parameters, or interval changed (unless **Recalc Math traces manually** option is not selected at **Format traces** tab).

The **Math** trace also applies to source **storage** data and is displayed as a **Math** trace storage. While normal trace storage is never changing, **Math** trace storage is automatically recalculated in case of operation or parameters changed, or trace range changed.

Please note that for some operations, the time interval of resulting **Math** trace could be smaller than that of a source trace. For example, one moving average data point is using a source data in the time interval of an averaging window size. As a result, moving average trace will start at half of an averaging window size later and stop at half of an averaging window size before a source trace.

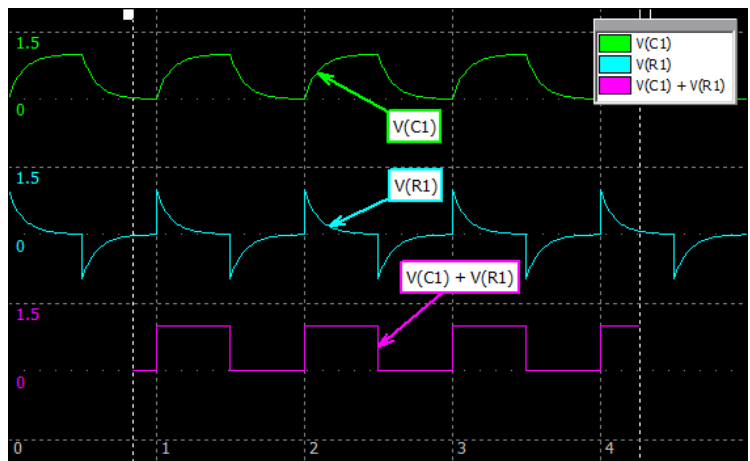
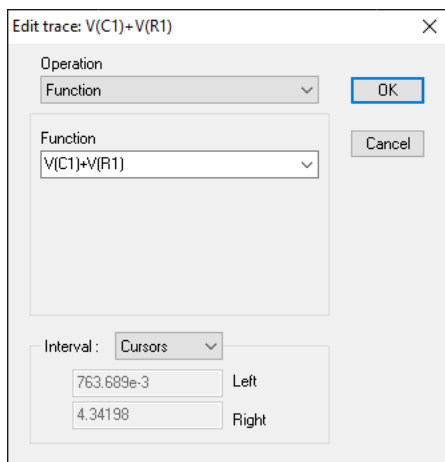
Details and examples of some **Math** types:

**Function.** Enter a function expression in the **Function** window. The function may consist of arithmetic operators and functions, component parameters, global variables of **C-code** components, any other traces, current transient time  $t$ . For example:

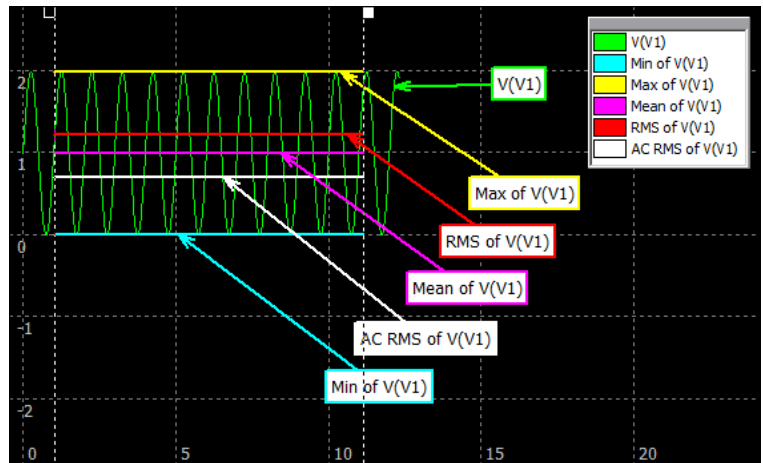
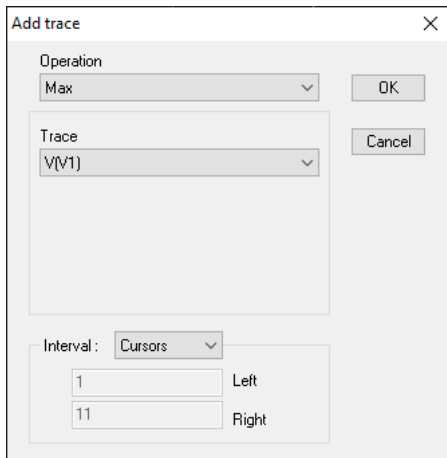
```
sin(t*1000.)*(1+cos(t*10.))
V(R1)/R1.R
old_trace + new_trace
```

To access a specific storage data, use a `trace_name.storage_name` notation:

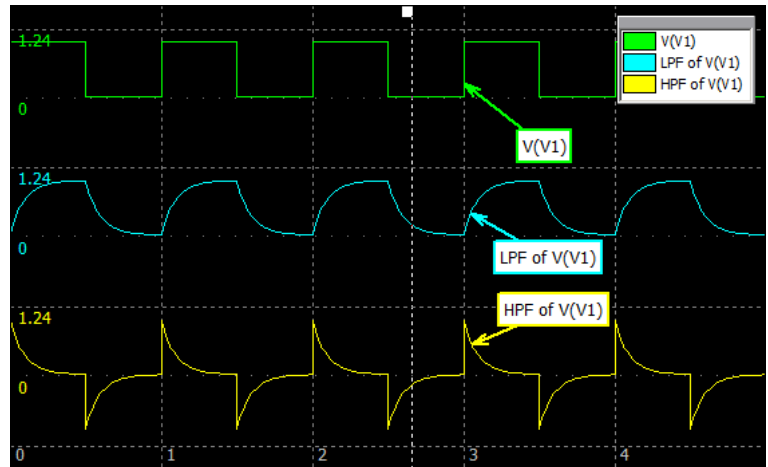
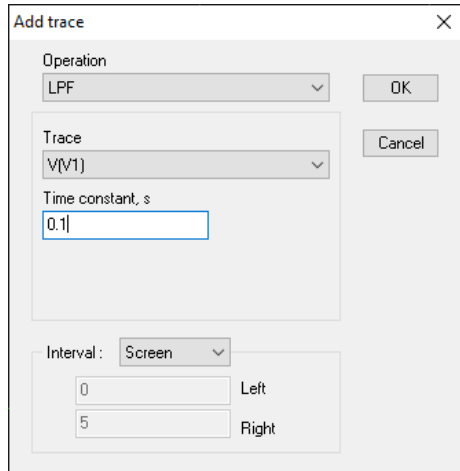
```
trace.Run1 - trace.Run2
```



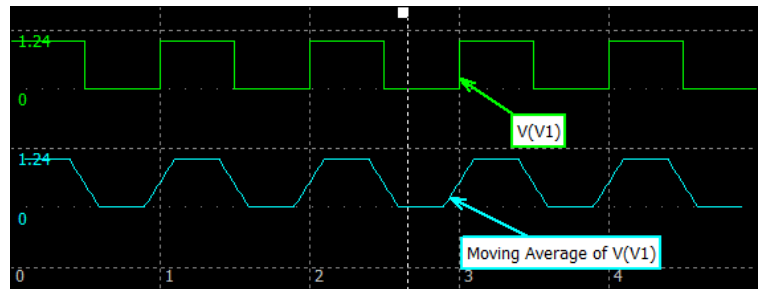
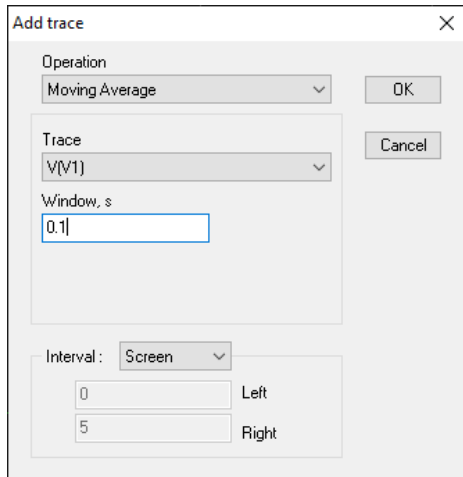
**Min, Max, Mean, RMS, SD.** Calculate specified value of selected trace.



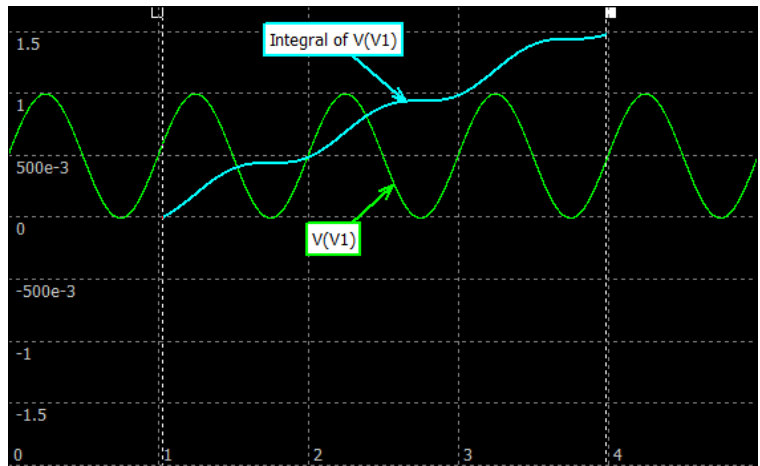
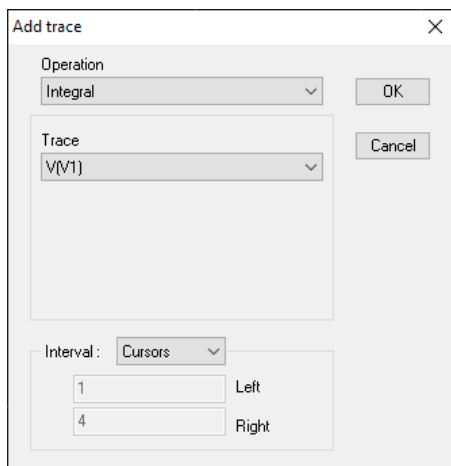
**LPF, HPF.** Apply low-pass or high-pass filter to selected trace.



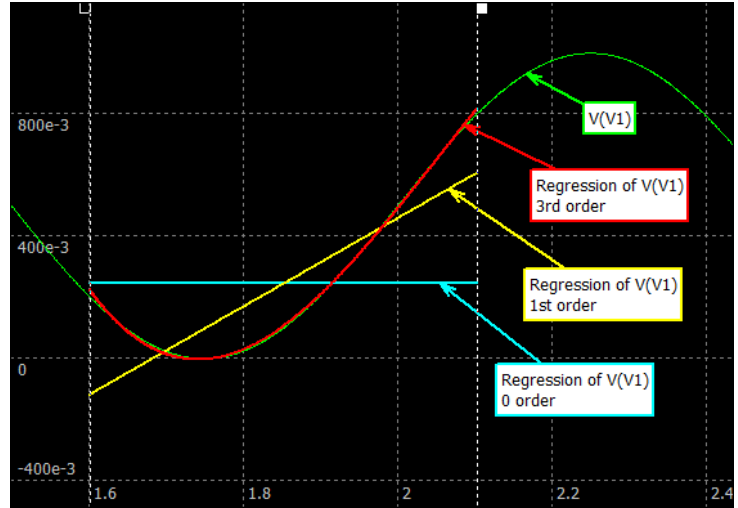
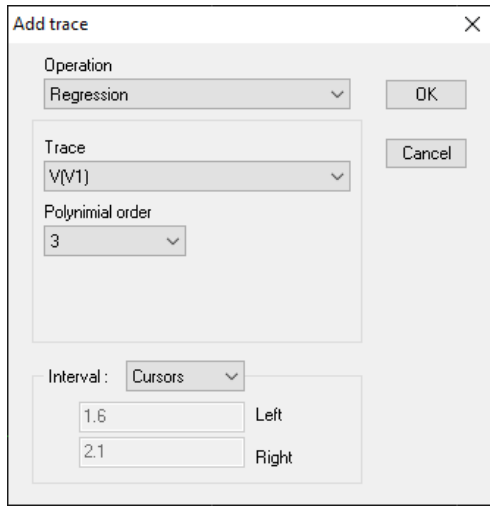
**Moving Average, Moving RMS, Moving SD.** Calculate specified moving average type.



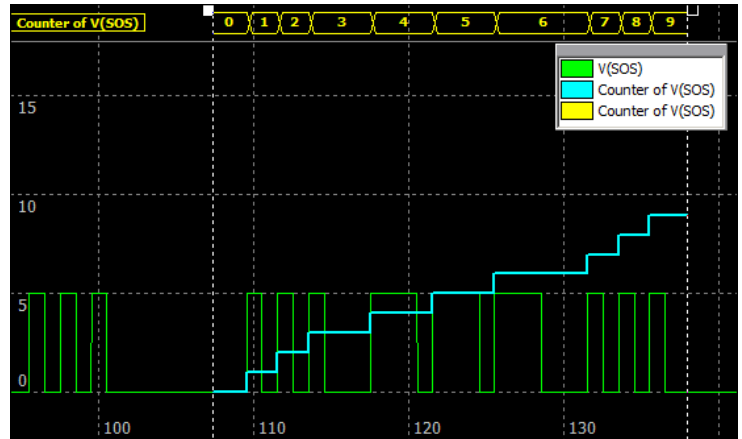
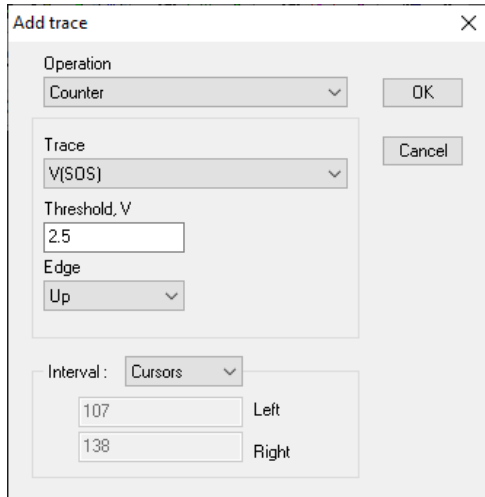
**Integral.** Calculate integral over time. Integral value starts from zero.



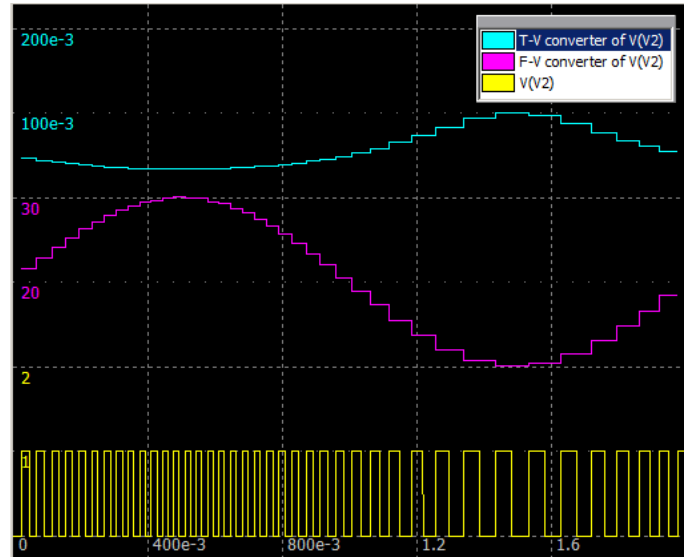
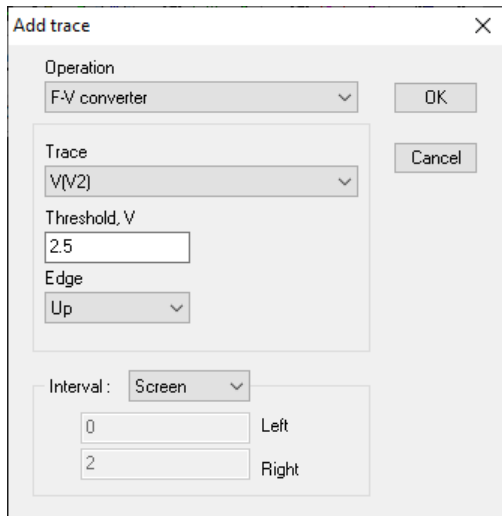
**Regression.** Calculate polynomial regression of selected order (0...5).



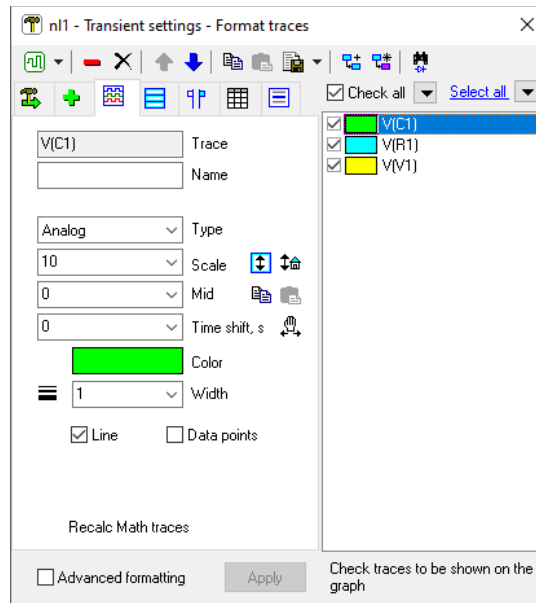
**Counter.** Count transitions of specified type (Up/Down).



**F-V converter, T-V converter.** Frequency-to-Voltage and Time-to-Voltage conversion. Trace voltage is equal to a frequency or a period of the trace signal, calculated using time between transition with specified threshold and direction.





## Format traces




**Check** traces to be shown on the transient graph. Click **Check All** to check/uncheck all traces or click  for more checking options.

**Select** one or more traces (using **Ctrl** and **Shift** keys) to perform operations on selected traces.

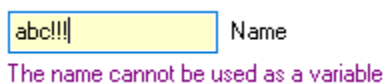
Click **Select All** to select all traces or click  for more selection options. Traces can also be selected in the **Legend** and **Labels** area of the Transient window. The top toolbar provides various operations on selected traces and trace data.

If more than one trace of a different type is selected, only parameters applicable to **all** selected traces will be shown. If parameter has different values for different traces, its field will be blank. If color is different, it will be displayed as:  Color. If a new value is entered in an empty field, that value will be assigned to all selected traces.

Trace configuration fields may be different for different trace types.

**Trace/Function/Math.** This field indicates trace type and is available for editing for **Function** trace only. For **Function** and **Math** trace, click  to edit the trace in the **Function** or **Edit trace** window.

**Name.** Enter trace name which will be shown instead of **Trace**. To be used in formulas, it should satisfy C-language standard for the variable name: start with letter or ‘\_’, and then contain only letters, digits, and ‘\_’. Otherwise, you will be given a warning:



**Type** defines how the trace will be displayed:

- **Analog** – trace will be displayed in analog format in **Analog section**.
- **Digital** and **Bus** – trace will be displayed in digital or bus format in **Digital section**.

**Analog.**

**Scale, Mid** – scale values for analog trace. Click  to select from previously used values.



- auto-scale selected traces.



- reset vertical scale of selected traces.



- copy currently displayed Scale and Mid values.

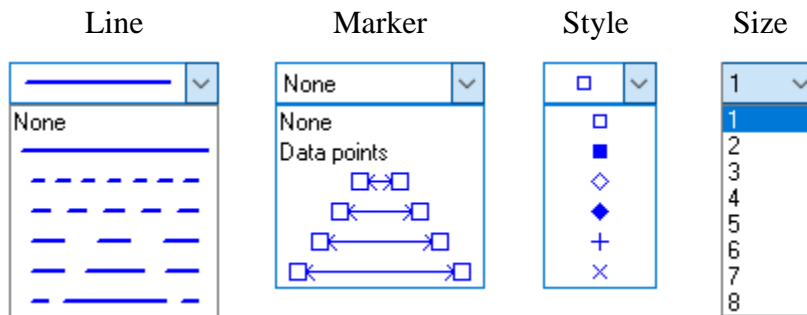


- paste Scale and Mid to selected traces

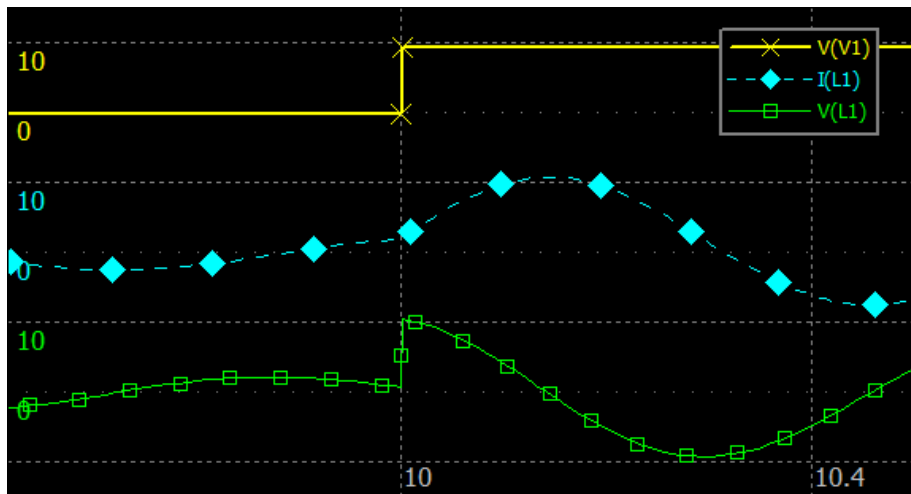
**Line** – show lines.

**Data points** – show simulation data points. If **Data compression** option is selected, some simulation data points of a constant signal will not be stored and displayed.

Select **Advanced formatting** to allow advanced formatting capabilities:





**Markers** may be located at calculated data points, similar to standard formatting **Data points** option, or at the same distance along the trace line, regardless of actual data points and trace shape:

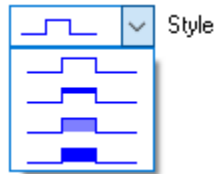




**Digital.**

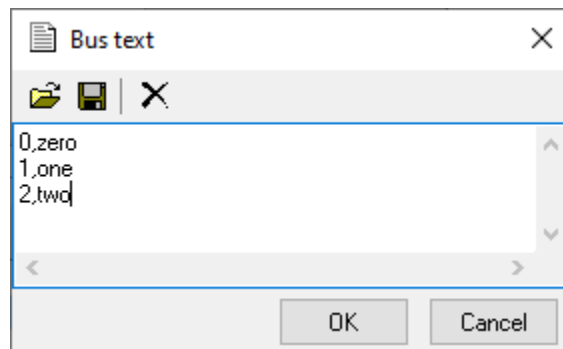
**Threshold** – is used to convert analog signal to logical low/high level. It can be different from transient **Logical Threshold**. Click  to see and select current Logical Threshold or click  to select current Logical Threshold.

**Style** – select style of displaying high level state:




**Bus.**

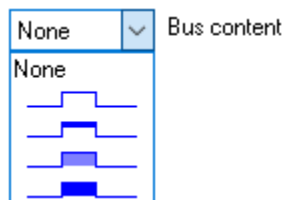
**Format** –format of the number displayed. For **Text** format, click  to associate displayed text with the value in the value, text format:



If the value is not found in the list, no text will be displayed.

**Fixed point** - number of fractional bits of a bus number, when displayed in a fixed point format. A number prefix format can be selected in the **Preferences** , **Digital/Bus** tab.


**Bus content** – select if individual bits of the bus are displayed as separate traces, and how high logic state is shown:



**Time shift, s** – trace will be displayed shifted horizontally. Click (check)  button to change time shift of selected traces by mouse in transient window. Click this button again (uncheck) to turn this mode off.

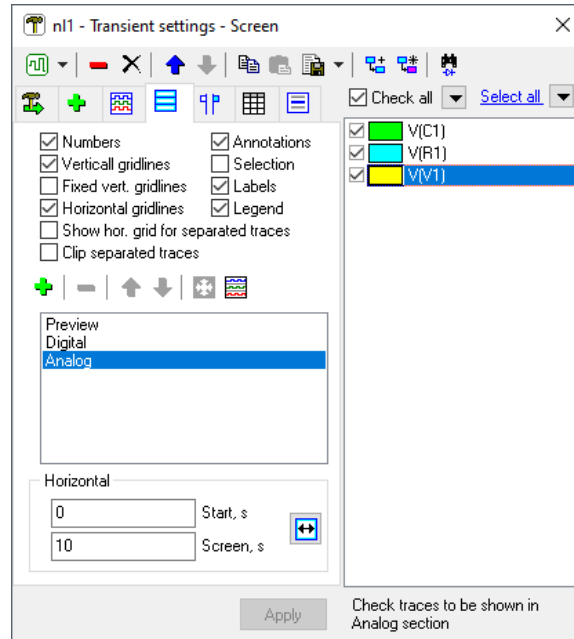
**Color** - click colored rectangle to select color of selected traces.

**Width** - enter or select from drop-down list width of selected traces (in pixels).

**Recalc Math traces**  – recalculate **Math** traces (visible only if **Math** traces exist).




**Auto recalc** - recalculate **Math** traces automatically (visible only if **Math** traces exist). Please note that automatic recalculation may take a long time. Uncheck this option and use manual recalculation if immediate updating of **Math** traces is not required.

## Screen



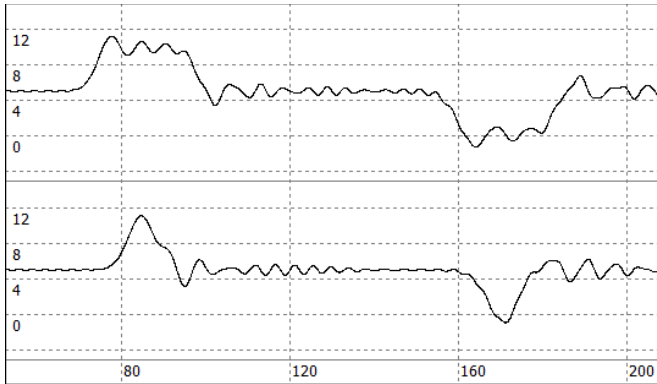
**Select** what is shown on the transient screen (numbers, gridlines, annotations, labels, etc.).

Configure transient graph **sections**:

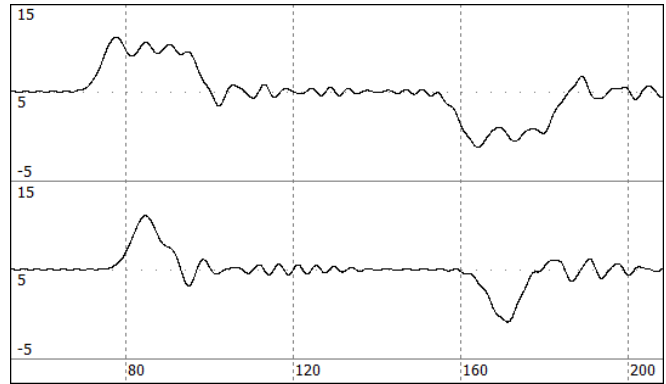
- Select **Preview** in the list, then check one trace to be shown in the **Preview** section. Uncheck the trace to hide **Preview** section.
- Select **Digital** in the list, then check traces to be shown in the **Digital** section in **Digital** or **Bus** format. When checked, the trace will be automatically removed from the section where it is currently displayed. When unchecked, the trace will not be shown on the graph.
- **Add analog section**  - add new **Analog** section (up to 4).
- Select **Analog N** in the list, then check traces to be shown in that section in **Analog** format. When checked, the trace will be automatically removed from the section where it is currently displayed. When unchecked, the trace will not be shown on the graph.
- **Maximize analog section**  - maximize selected analog section.
- **Separate traces**  - separate traces in the selected analog section.

Special options for **Separate traces** mode:

- **Show hor. Grid for separated traces.** If selected, a standard horizontal grid lines are shown for each trace, otherwise only **Top**, **Mid**, and **Bottom** values (if vertical size allows):

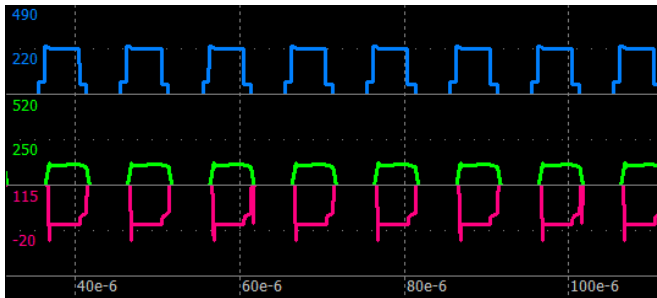


**Selected.**

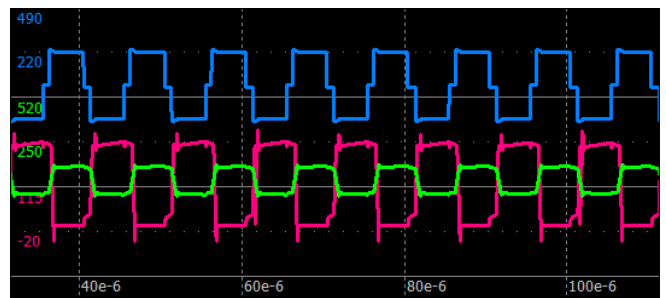


**Not selected.**

- **Clip separated traces.** If selected, a trace will be “clipped” within its area and won’t overlap with other traces, otherwise traces from different areas may overlap:

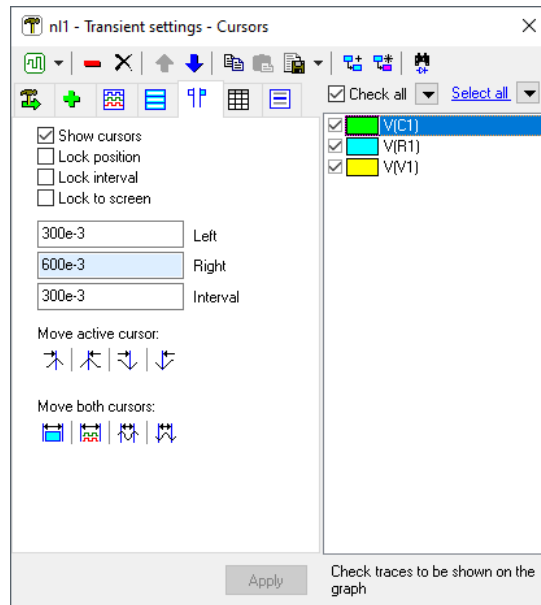


**Selected.**



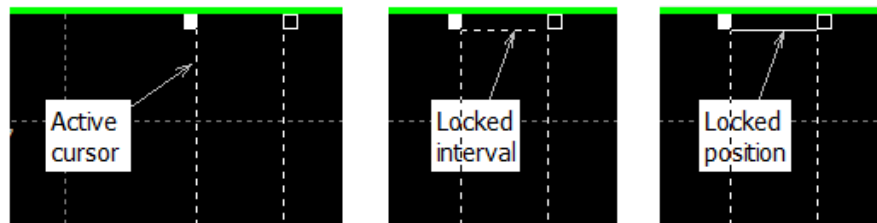
**Not selected.**

# Cursors



**Lock position** – cursors are not moving.

**Lock interval** – both cursors are moving together keeping same interval between them.



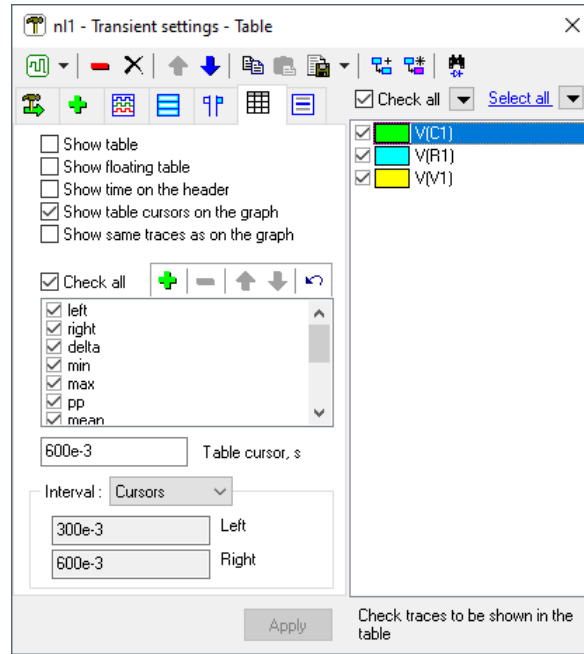
**Lock to screen** – keep cursors at screen edges.

**Left, Right, Interval** – enter position of the cursors or interval between cursors.

**Move active cursor** – click to move active cursor right/left to min/max data point of selected trace.

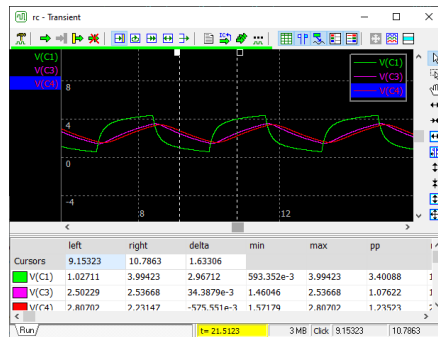
**Move both cursors** – click to move both cursors to left and right edges of the screen, start and end of the data, or min/max data points of selected trace.

# Table

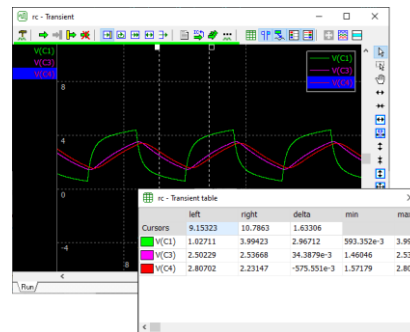


**Show table** – show/hide the table.

**Show floating table** – show/hide table in floating window.



**Table**



**Floating table**


**Show time on the header:**

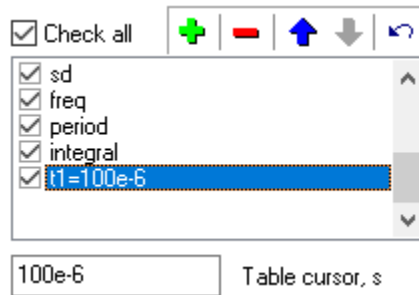
	left	right	delta	min	max
Cursors	9.15323	10.7863	1.63306		
V(C1)	1.02711	3.99423	2.96712	593.352e-3	3.99423
V(C3)	2.50229	2.53668	34.3879e-3	1.46046	2.53668
V(C4)	2.80702	2.23147	-575.551e-3	1.57179	2.80702

**Off**

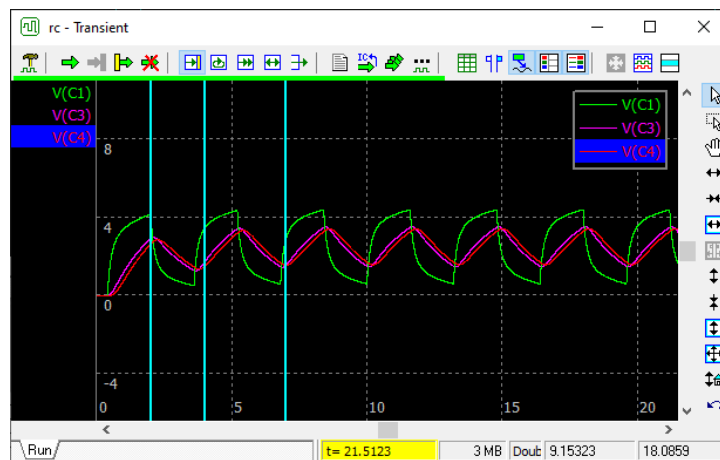
	left	right	delta	min	max
Cursors	9.15323	10.7863	1.63306		
V(C1)	1.02711	3.99423	2.96712	593.352e-3	3.99423
V(C3)	2.50229	2.53668	34.3879e-3	1.46046	2.53668
V(C4)	2.80702	2.23147	-575.551e-3	1.57179	2.80702

**On**

**Show table cursors on the graph.** Custom **Table cursors** can be added to the table. To add a table cursor, enter time in the **Table cursor** field, or set active cursor in the **Transient window** to a desired time, and click **Add table cursor**  button:



A new item is added to the list. Now, the **t1** column in the table will show trace values at specified time. Select **Show table cursors on the graph** to show table cursors on the graph:



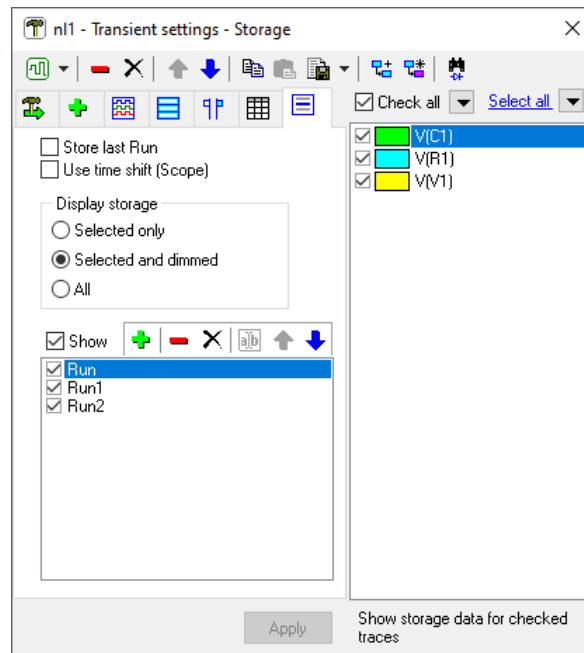
**Show same traces as on the graph** – if checked, the table will always show the same traces as shown on the graph.

**Check** trace parameters to be shown in the table.

Select **Interval** where table data are calculated.

**Check** traces to be shown in the transient table. If **Show same traces as on the graph** is checked, checked traces will also be shown on the graph.

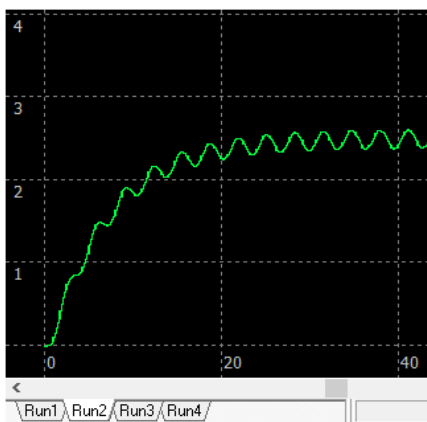
## Storage



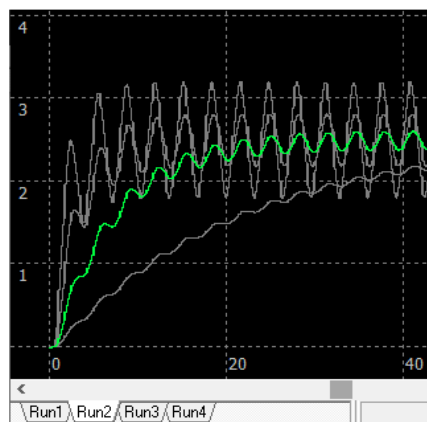
**Store last Run** – when transient starts, move current transient run to the storage under the name **Last**. You can always see the difference between a new and previous run.

**Use time shift (Scope)** - select how segmented memory data are being displayed. See **Working with segmented memory** chapter of the **Scope** tool description.

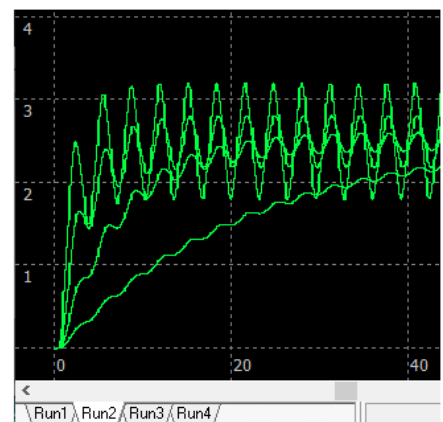
**Display storage** – specify how to show storage data on the graph:



**Selected only**



**Selected and dimmed**



**All**


**Check** storage data in the list to show it on the graph and table.

**Check** traces to show storage data for those traces.





## More commands (toolbar)


**Copy selected traces**  - copy selected traces with data to the clipboard.

**Paste traces**  - paste traces with data from the clipboard.

**Data...**  drop-down menu:

**Open**  - open (load) traces and data from a file of **nlt** type. You can select one or multiple files to open. Prefix “Copy of...” will be added to a trace name.

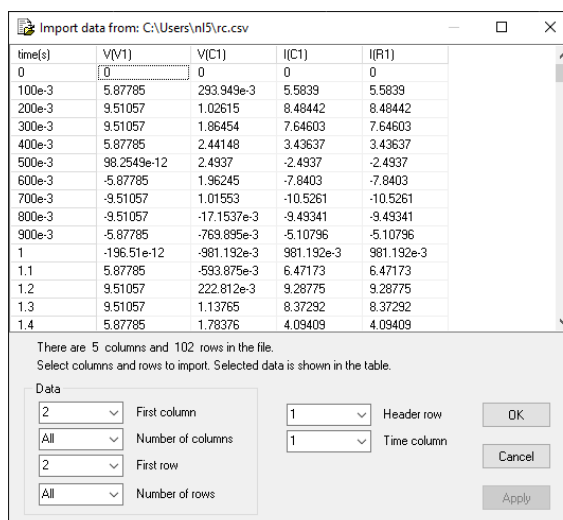
**Save**  - save selected traces with data in a file of **nlt** type, in special NL5 transient data binary format. Traces without data will not be saved.

**Import from file**  - import traces and data from text or binary files created by NL5 or other software tools or instruments. You can select one or multiple files to open. The following file formats (defined by file name extension) are supported:

- **txt, csv** – text format (comma-separated).
- **wfm** – Tektronix waveform format.
- **isf** – Tektronix internal format.
- **bin** – Keysight Technologies (Agilent) binary format.
- **trc** – LeCroy binary format.
- **raw** – LTSpice data format.
- **qraw** – QSPICE data format.

### Text format (txt, csv).

When file is selected, its content is shown in the window for verification and editing:



**First column** - select first column of the data to be imported.

**Number of columns** - select number of data columns to be imported. To import all available data (after the first column), select **All** from drop-down list (or set this parameter to -1).

**First row** - select first row of the data to be imported.

**Number of rows** – select number of data rows to be imported. To import all available data (after the first row), select **All** from drop-down list (or set this parameter to -1).

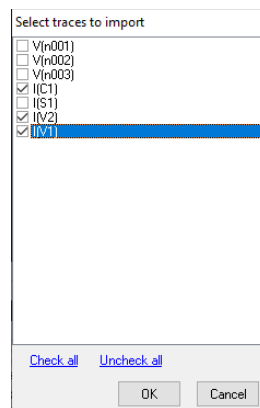
**Header row** - select the row with trace names. If such a row does not exist in the data file, select **Add header** from drop-down list (or set this parameter to 0), and trace names will be added automatically.

**Time column** - select the column with time. If such a column does not exist in the data file, select **Add time** from drop-down list (or set this parameter to 0), and time will be added automatically with time step specified in **Time step** parameter.

Both **comma** and **Tab** delimiters of the text data are accepted.

### LTSpice format (raw), QSPICE format (qrow).


List of traces (variables) available in the file will be shown in the **Select traces to import** window. Check traces and click **OK**:



Check traces and Click **OK** to confirm import.

### Scope data formats (wfm, isf, bin, trc).

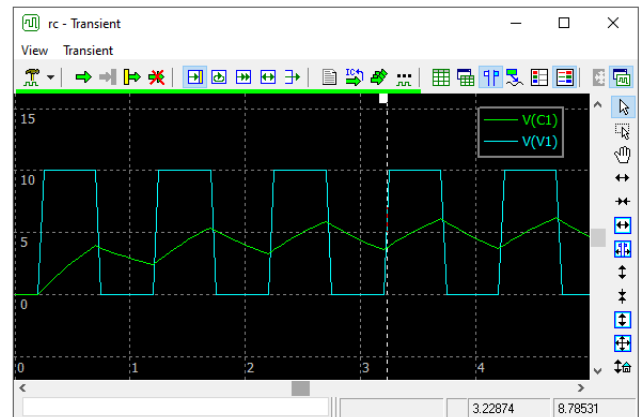
Please be aware that not all format versions and format-specific options are implemented and tested. If you receive error or warning message, please contact us to provide details about the failure: it would help us fixing the problem and adding new features and options. If you want more formats to be implemented in the NL5, feel free to contact us as well.


**Import from clipboard**  - paste traces in text format from the clipboard. The same selection window as for **Import from file** command will be shown with data obtained from clipboard.

Both **comma** and **Tab** delimiters of the text are accepted.

For example, you can select a range with trace data in MS Excel, **Copy** data to clipboard, then **Import from clipboard** to NL5 transient window:

	A	B	C	D	E	F	G
1		time(s)	V(C1)	V(V1)			
2		0	0	0			
3		5.00E-02	0	0			
4		1.00E-01	0	0			
5		1.50E-01	0	0			
6		2.00E-01	0.00E+00	0.00E+00			
7		2.50E-01	4.88E-01	10			
8		3.00E-01	9.52E-01	10			
9		3.50E-01	1.39292	10			
10		4.00E-01	1.81269	10			
11		4.50E-01	2.21199	10			
12		5.00E-01	2.59182	10			
13		5.50E-01	2.95312	10			
14		6.00E-01	3.2968	10			



**View/Export**  - view trace data in the table and save in a file or copy to clipboard in text format. Specify export parameters:

View/Export data			
0	0	0	
50e-3	0	0	
100e-3	0	0	
150e-3	0	0	
200e-3	0	0	
250e-3	488e-3	10	
300e-3	952e-3	10	
350e-3	1.39292	10	
400e-3	1.81269	10	
450e-3	2.21199	10	
500e-3	2.59182	10	
550e-3	2.95312	10	
600e-3	3.2968	10	
650e-3	3.62372	10	
700e-3	3.93469	10	
750e-3	3.7428	0	
800e-3	3.56026	0	
850e-3	3.38662	0	
900e-3	3.22145	0	
950e-3	3.06434	0	
1	2.91489	0	
1.05	2.77273	0	
1.1	2.6375	0	
1.15	2.50887	0	
1.2	2.38651	44.4089e-15	
1.25	2.27583	10	

Check all  
 V(C1)  
 V(V1)

Interval: All data  
 0 Left  
 5 Right  
 50e-3 Step

Significant digits  
 6 Time  
 6 Data

Tab Delimiter

**Check** traces to be exported.

**Interval** – data interval to be exported.

**Step** – the data are exported with fixed specified time step. Transient data between calculated data points are linearly interpolated.

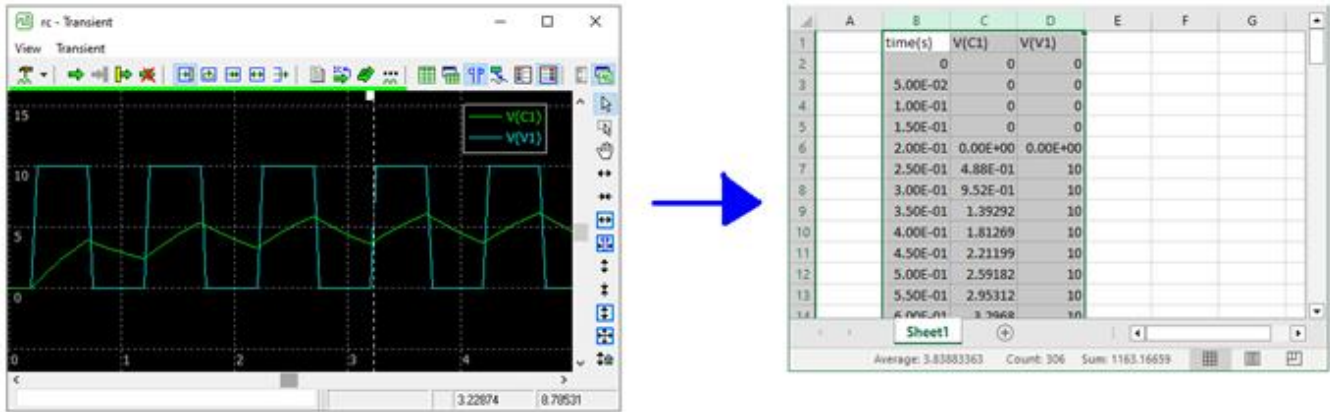
**Significant digits** - select significant digits for time and data.


**Delimiter** – select **comma** or **Tab**.

Click **Copy** to copy data to clipboard.

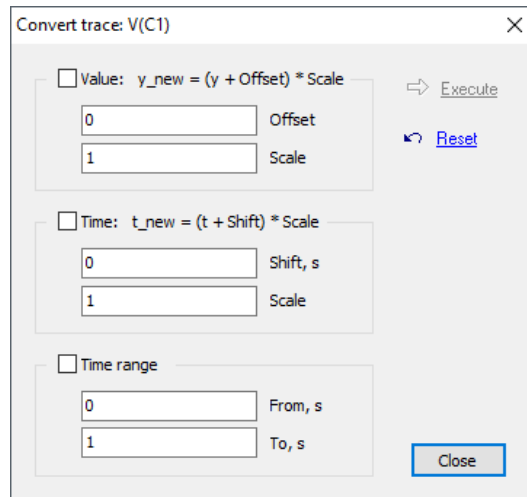
Click **Export** to save into the file. Select file name and extension (**csv, txt**).



To transfer data from NL5 to MS Excel, set **Tab** delimiter, **Copy** data to clipboard, then **Paste** data to Excel spreadsheet:



**Duplicate selected traces**  - duplicate (same as **Copy/Paste**) selected traces with data.

**Convert selected traces**  - duplicate selected traces with specified data conversion:



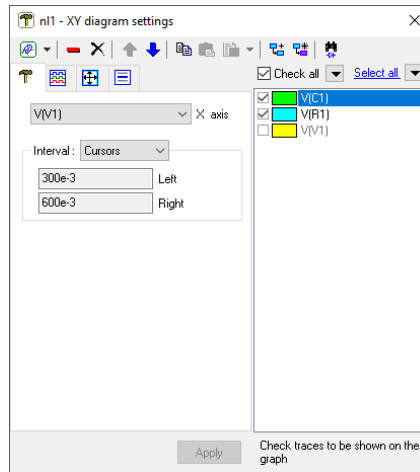
Specify desired conversion of trace value and time (offset and scale multiplier), and clipping trace time range. Check desired actions and click  **Execute** to proceed. Click  **Reset** to reset to initial settings.

**Find component**  - show (select) the component used for V/I/P trace on the schematic.

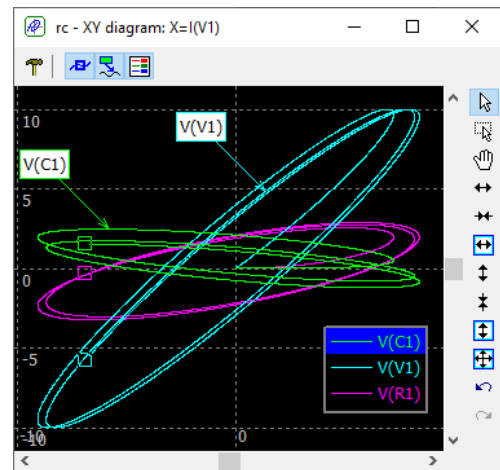
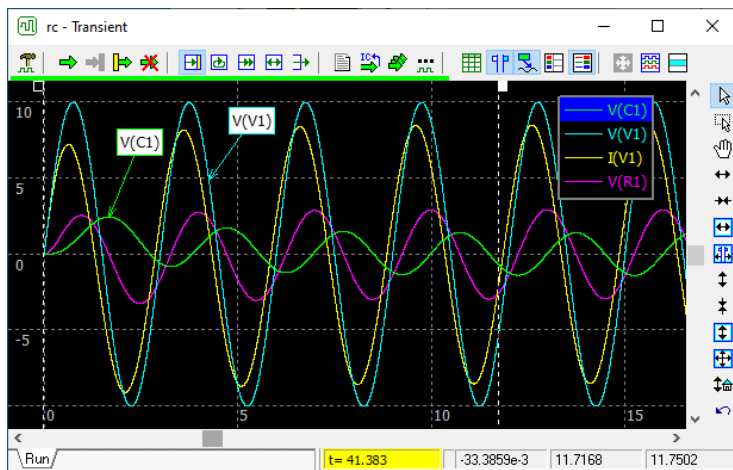
# Transient Tools


## XY diagram

Select **Transient/Transient tools**  or click **Transient tools**  on the **Transient window** toolbar, select **XY diagram** . Click **Tool settings**  to configure XY diagram.



Select trace to be used as **X axis**. Trace name is shown in the windows header.  
 Select **Interval** of transient data to be shown in the XY diagram.  
**Check** traces in the list to be shown on the graph.



XY diagram shows traces with **Scale** and **Mid** values specified in Transient settings. In addition, XY diagram graph can be zoomed/scroll by toolbar buttons, mouse (similar to Transient window), or by setting parameters **Multiplier** and **Offset** in the settings **Screen**  tab of Settings window.

**Double-click** to center screen.


**Right-click** on the graph to show context menu.

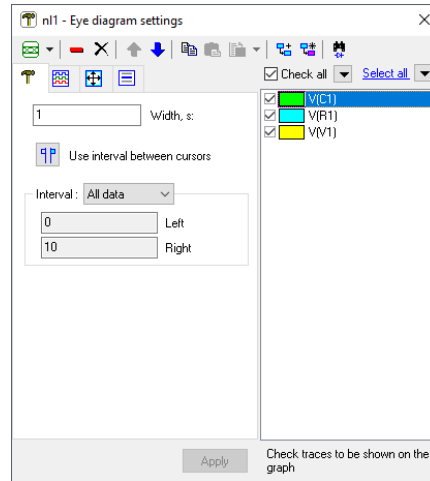
**Show markers**  - show markers corresponding to active cursor in Transient window.

**Show annotations**  - show annotations defined in Transient window.


## Eye diagram

Select **Transient/Transient tools**  or click **Transient tools**  on the **Transient window** toolbar, select **Eye diagram** .

Click **Tool settings**  to configure Eye diagram.

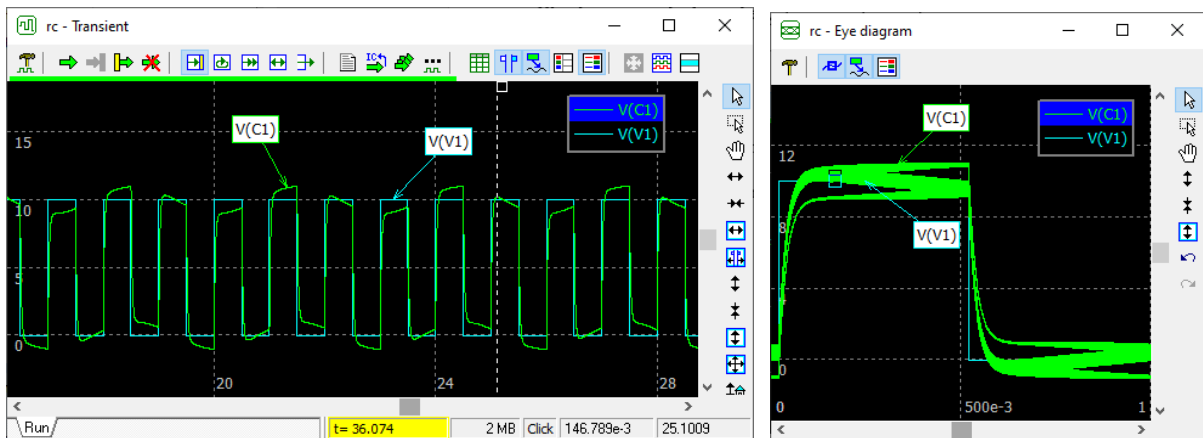



Select **Width** of Eye diagram.

Click  to use interval between Transient cursors.

Select **Interval** of transient data to be shown in the Eye diagram.

**Check** traces in the list to be shown on the graph.



Eye diagram shows traces with **Scale** and **Mid** values specified in Transient settings. In addition, Eye diagram graph is zoomed/scrolls by toolbar buttons, mouse (similar to Transient window), or by setting parameters **Multiplier** and **Offset** in the settings **Screen**  tab.

**Double-click** to center screen.


**Right-click** on the graph to show context menu.

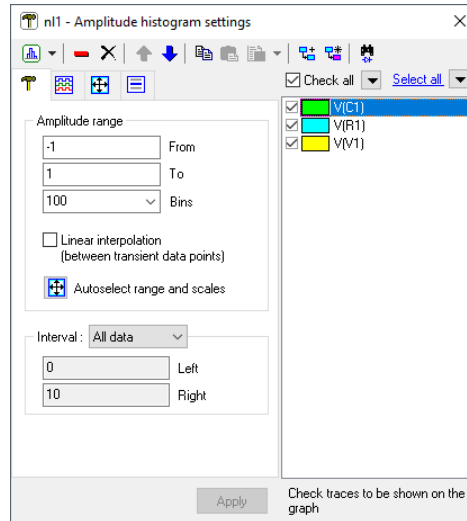
**Show markers**  - show markers corresponding to active cursor in Transient window.

**Show annotations**  - show annotations defined in Transient window.

## Amplitude histogram

Select **Transient/Transient tools**  or click **Transient tools**  on the **Transient window** toolbar, select **Amplitude histogram** .


Click **Tool settings**  to configure Amplitude histogram.



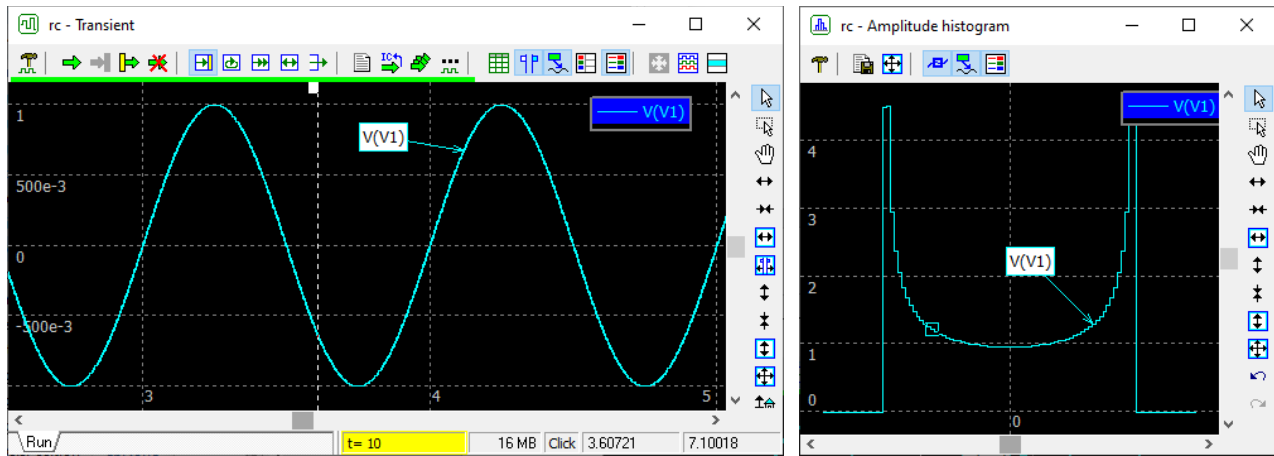
Select **From** and **To** of Amplitude range.  
Select number of **Bins** within Amplitude range.

**Linear interpolation** (between transient data points). NL5 calculates transient response at discrete time points and then displays traces connecting those points by straight lines. If **Linear interpolation** option is selected, such linearly interpolated signal will be used as a source for amplitude histogram. For slow changing signal linear interpolation provides quite accurate approximation of a real signal, so that amplitude histogram will give quite accurate and reliable results.

However, if the signal is supposed to have only discrete amplitude levels with fast transition between those levels comparing to simulation step, such an interpolation may produce wrong results: the histogram may show signal amplitudes between discrete states which do not exist in real signal. Use **Linear interpolation** option to obtain accurate histogram for specific signals.

**Autoselect range and Scales**  - set Amplitude range to minimum and maximum of the signal and perform **Fit the screen** of the histogram.

Select **Interval** of transient data to be shown in the Eye diagram.  
**Check** traces in the list to be shown on the graph.




Zoom, scroll, mouse, and mouse-wheel operations are the same as in Transient window.

**Double-click** to center screen.

**Right-click** on the graph to show context menu.

**Show markers**  - show markers corresponding to active cursor in Transient window.

**Show annotations**  - show annotations defined in Transient window.

**View/Export**  - view histogram data in the table and export into text file or copy to clipboard:

bin_start	V(V1)
-10	0
-9.33333	0
-8.66667	1.16832e-3
-8	1.16832e-3
-7.33333	11.6832e-3
-6.66667	28.0396e-3
-6	75.9405e-3
-5.33333	172.911e-3
-4.66667	389.049e-3
-4	778.098e-3
-3.33333	1.27346
-2.66667	2.13568
-2	2.79227
-1.33333	3.56219
-.666667e-3	3.93985
0	3.95299
.666667e-3	3.39396
1.33333	2.73882
2	2.04222
2.66667	1.29333
3.33333	761.742e-3
4	357.504e-3

Check all

V(V1)

Significant digits:  Data

Delimiter:

Copy    Export...    Close

**Check** traces to be exported.

**Significant digits** - select significant digits for data.

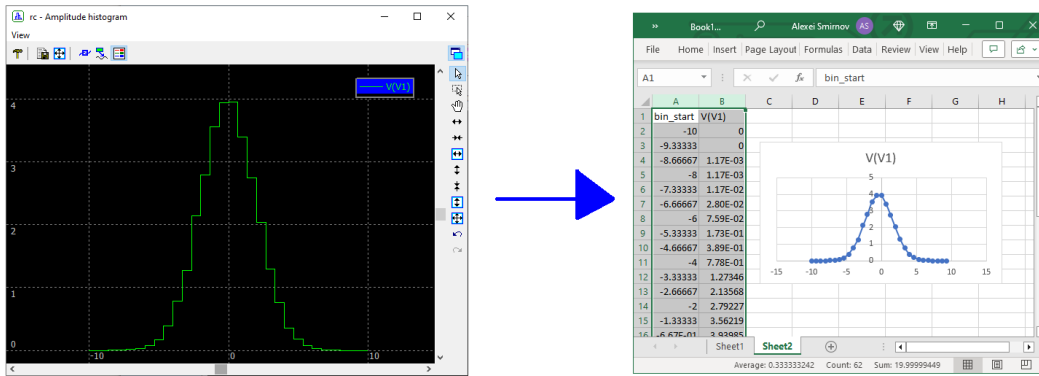
**Delimiter** – select **comma** or **Tab**.

Click **Copy** to copy data to clipboard.

Click **Export** to save into the file. Select file name and extension (**csv**, **txt**).



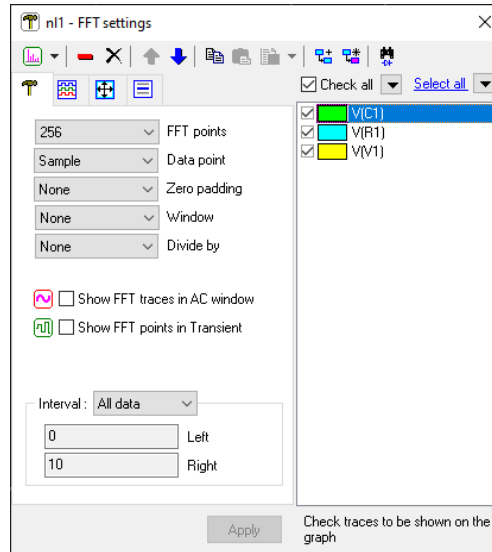
To transfer data from NL5 to MS Excel, set **Tab** delimiter, **Copy** data to clipboard, then **Paste** data to Excel spreadsheet:



## FFT

Select **Transient/Transient tools**  or click **Transient tools**  on the **Transient window** toolbar, select **FFT** .

Click **Tool settings**  to configure FFT.



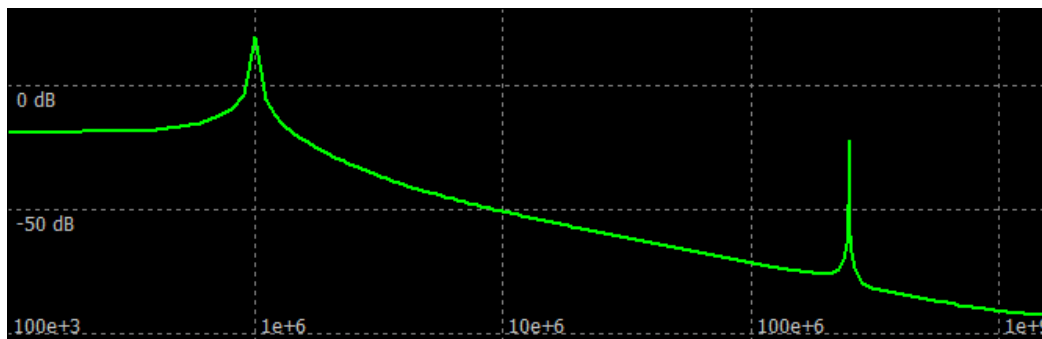
Select **Interval** of transient data to be used for FFT calculation.

**Points** – number of FFT points.

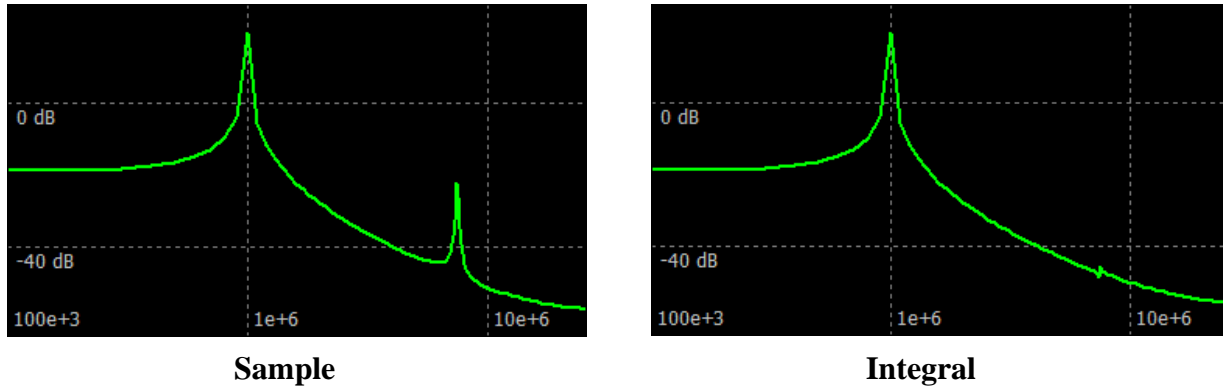
**Data point** – method of taking transient data point:

**Sample:** trace value at sampling time point,


**Integral:** average (integral) of the trace between current and next sampling time point. This method may help to reduce aliasing effects of wide-spectrum input signals. For example, this is FFT of a wide-spectrum signal taken with 32768 FFT points:

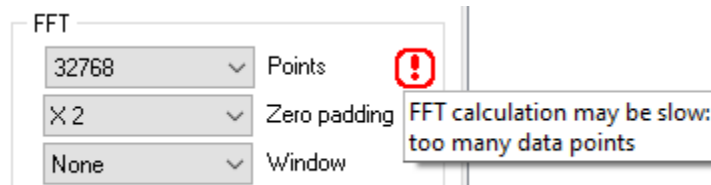


512 points FFT with **Sample** method gives a high peak at 7.5 MHz, which is an alias of high frequency signal component. With **Integral** method, this peak is significantly reduced:



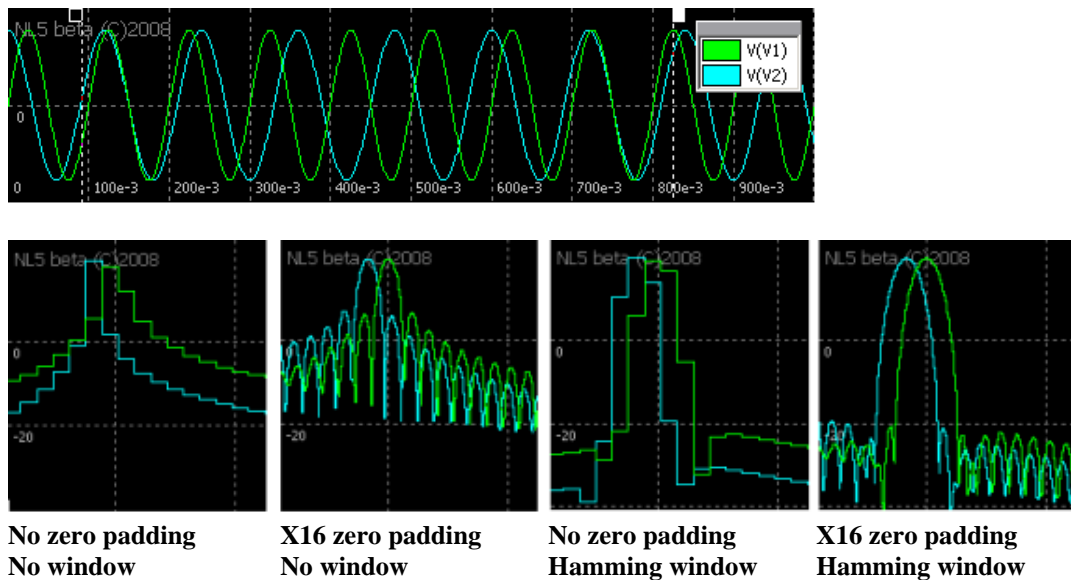
**Zero padding** - add zeroes to transient samples. Zero padding is a standard technique to improve spectral resolution.

If total number of FFT points, including zero padding, is too high ( $\geq 65536$ ), a warning sign  will show up. When move mouse over the sign, a hint message will be shown:



**Window** – select window type. Windowing is a standard technique to reduce leakage effects and improve spectral resolution.

Here are examples of **Zero padding** and **Window** effect on FFT result:




No zero padding  
No window


X16 zero padding  
No window

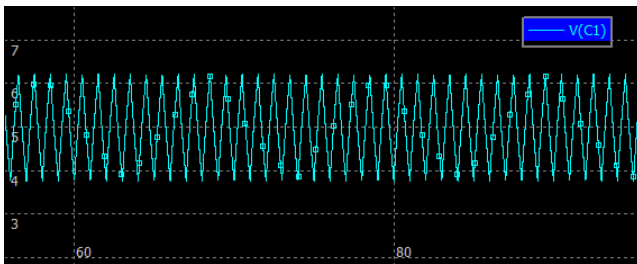
No zero padding  
Hamming window

X16 zero padding  
Hamming window

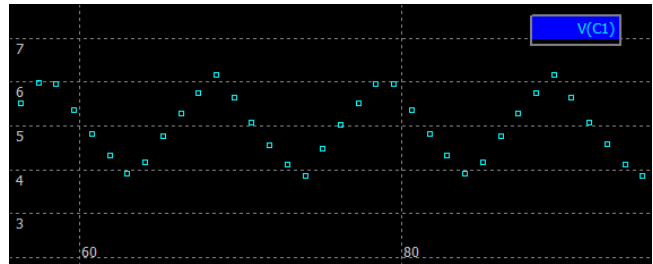
**Divide by** – FFT for all traces will be divided by FFT of specified trace (except that specified trace).

**Show FFT traces in AC window**  - show FFT traces in AC window as well as in FFT window. The name of the traces will be “FFT of *trace name*”. Working with AC traces in AC window allows using cursors and data table.

**Show FFT points in Transient**  - show points of transient trace which are used to calculate FFT for that trace. This feature may help to detect possible aliasing or other problems. Here is an example of aliasing when signal frequency is higher than Nyquist frequency:




Signal with FFT points



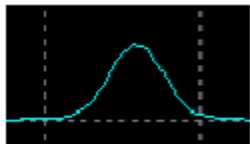
FFT points only

**Check** traces in the list to be shown on the graph.

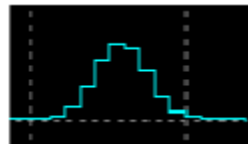
Select **Format traces**  tab to specify FFT traces format:

Show traces as:

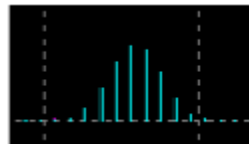
<input checked="" type="checkbox"/> Line	<input type="checkbox"/> Bars
<input type="checkbox"/> Histogram	<input type="checkbox"/> Points



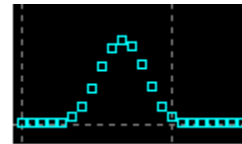
Line



Histogram



Bars



Points




Several trace options can be selected at a time. Some options are available only for specific lin/log selection of horizontal and/or vertical axes of the graph.

Zoom, scroll, mouse, and mouse wheel operations are the same as in **AC Window**.

**Double-click** to center screen.


**Right-click** on the graph to show context menu.

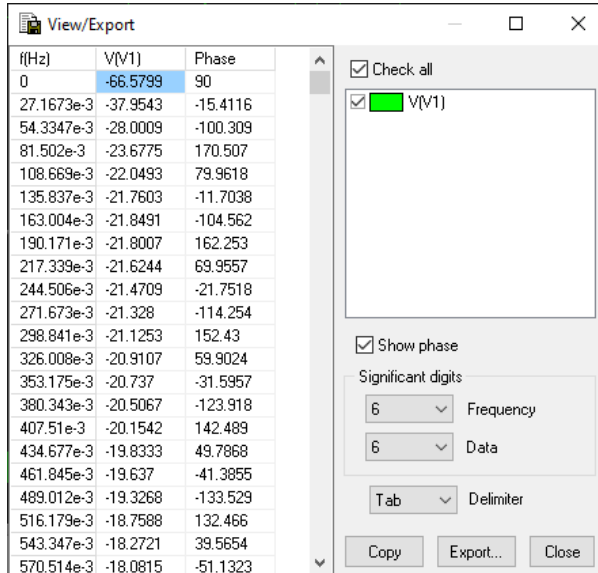
**Phase display:**

- **Phase Off**  : phase not shown.
- **Phase On**  : phase is shown in the same window as magnitude.
- **Phase separate**  : phase is shown in separate window (bottom section).

**Phase** is shown in -180...+180 degrees range. However, for FFT trace divided by another trace, the phase is adjusted to show continuous phase in wide range.

FFT data point at **zero frequency** is not shown.

**View/Export**  - view FFT data in the table and export into text file (comma-separated):



f[Hz]	V[V1]	Phase
0	-66.5799	90
27.1673e-3	-37.9543	-15.4116
54.3347e-3	-28.0009	-100.309
81.502e-3	-23.6775	170.507
108.669e-3	-22.0493	79.9618
135.837e-3	-21.7603	-11.7038
163.004e-3	-21.8491	-104.562
190.171e-3	-21.8007	162.253
217.339e-3	-21.6244	69.9557
244.506e-3	-21.4709	-21.7518
271.673e-3	-21.328	-114.254
298.841e-3	-21.1253	152.43
326.008e-3	-20.9107	59.9024
353.175e-3	-20.737	-31.5957
380.343e-3	-20.5067	-123.918
407.51e-3	-20.1542	142.489
434.677e-3	-19.8333	49.7868
461.845e-3	-19.637	-41.3855
489.012e-3	-19.3268	-133.529
516.179e-3	-18.7588	132.466
543.347e-3	-18.2721	39.5654
570.514e-3	-18.0815	-51.1323

View/Export dialog options:

- Check all
- V[V1]
- Show phase
- Significant digits:
  - Frequency: 6
  - Data: 6
- Delimiter: Tab
- Buttons: Copy, Export..., Close

**Check** traces to be exported.

**Significant digits** - select significant digits for frequency and data.


**Delimiter** – select **comma** or **Tab** (use Tab to export data to MS Excel).

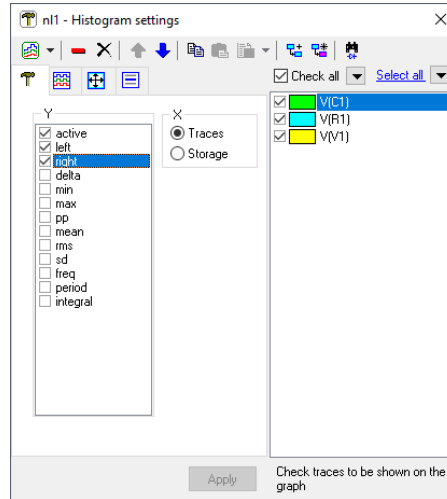
Click **Copy** to copy data to clipboard.

Click **Export** to save into the file. Select file name and extension (**csv**, **txt**).

## Transient histogram

Select **Transient/Transient tools**  or click **Transient tools**  on the **Transient window** toolbar, select **Transient histogram** .

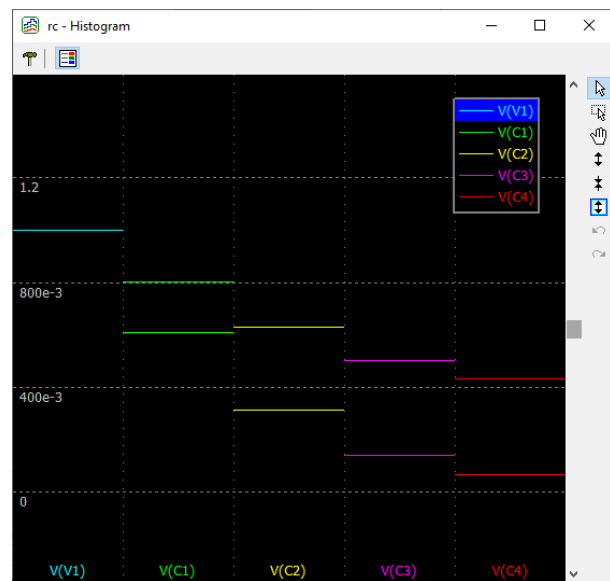
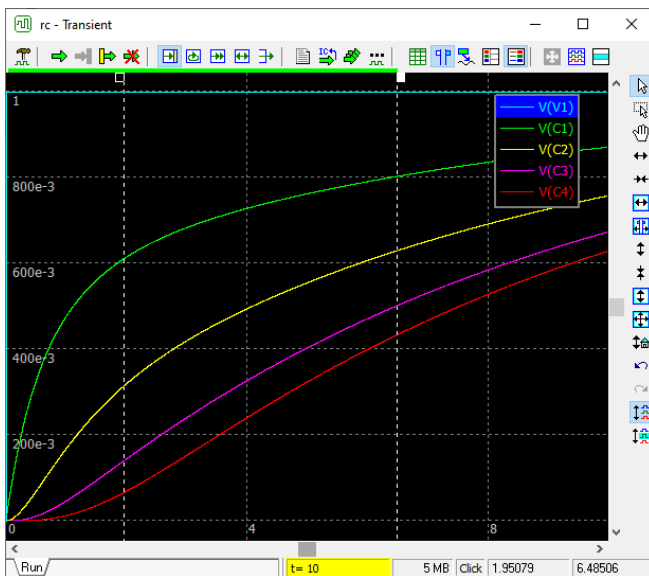
Click **Tool settings**  to configure Transient histogram.



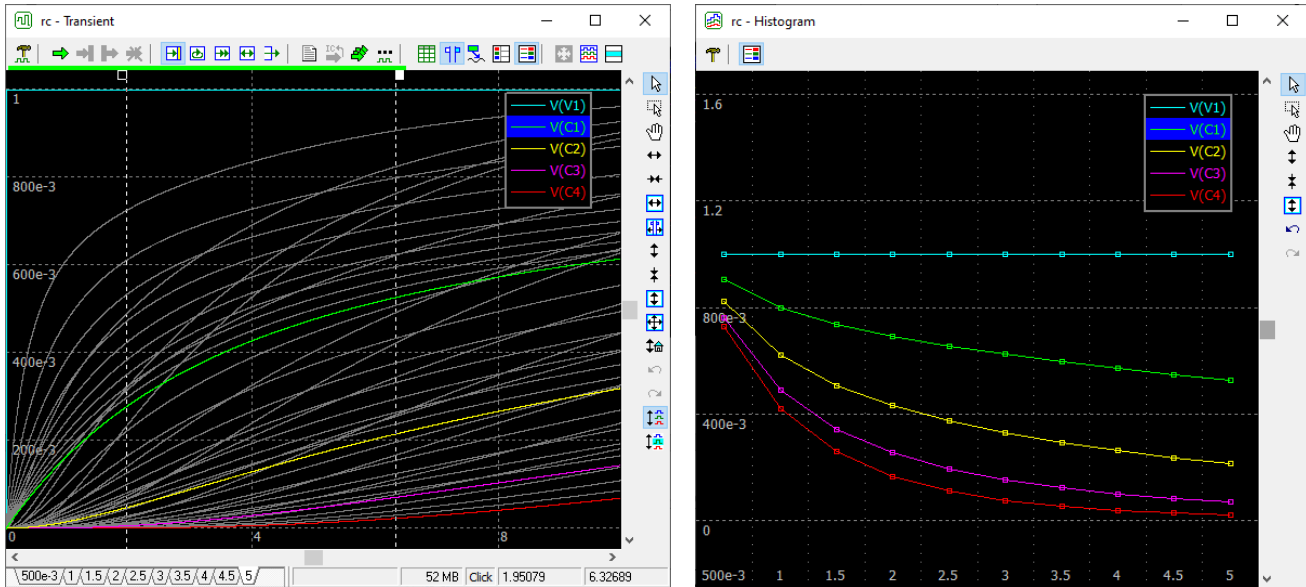
**Y** - histogram will display data calculated in the interval specified for Transient table (Transient table does not need to be shown).

**X**:

- **Traces**: show data as function of traces:



- **Storage:** show data as function of storage:



In **Storage** mode, select **Format traces**  tab to specify traces format:


Show traces as:

Line       Histogram

**Check** traces in the list to be shown on the graph.

Zoom, scroll, mouse, and mouse-wheel operations are the same as in Transient window.

**Right-click** on the graph to show context menu.

On the **Screen** tab  of Settings window, select **Trace type: Line or Histogram.**

## Power

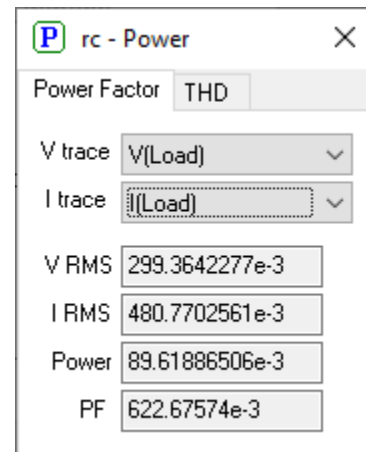
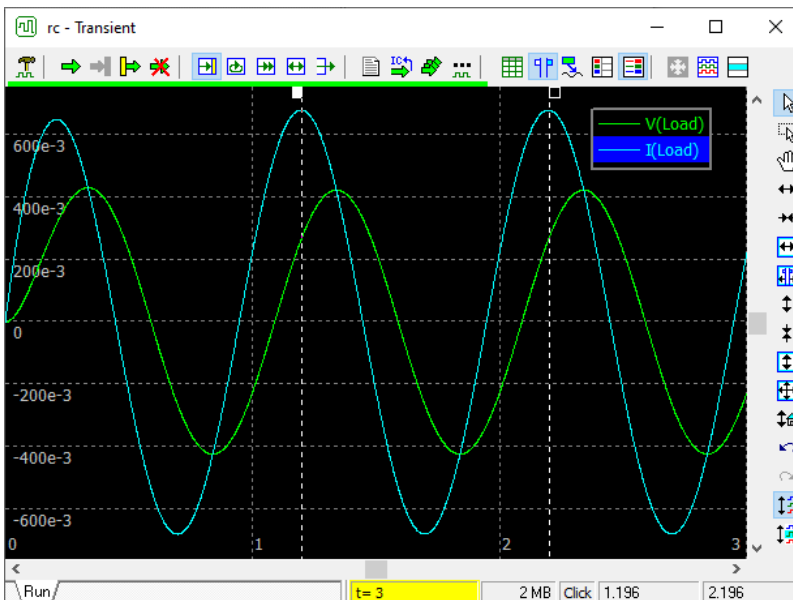
Select **Transient/Transient tools**  or click **Transient tools**  on the **Transient window** toolbar, select **Power** .

Select **Power Factor** tab to calculate voltage and current RMS, power, and Power Factor.

- **V trace:** select voltage trace from the drop-down list.
- **I trace:** select current trace from the drop-down list.

Power factor **PF** is calculated as: 
$$PF = \frac{Power}{V_{RMS} \times I_{RMS}}$$

All parameters are calculated between cursors. For better accuracy, provide integer number of signal cycles between cursors.



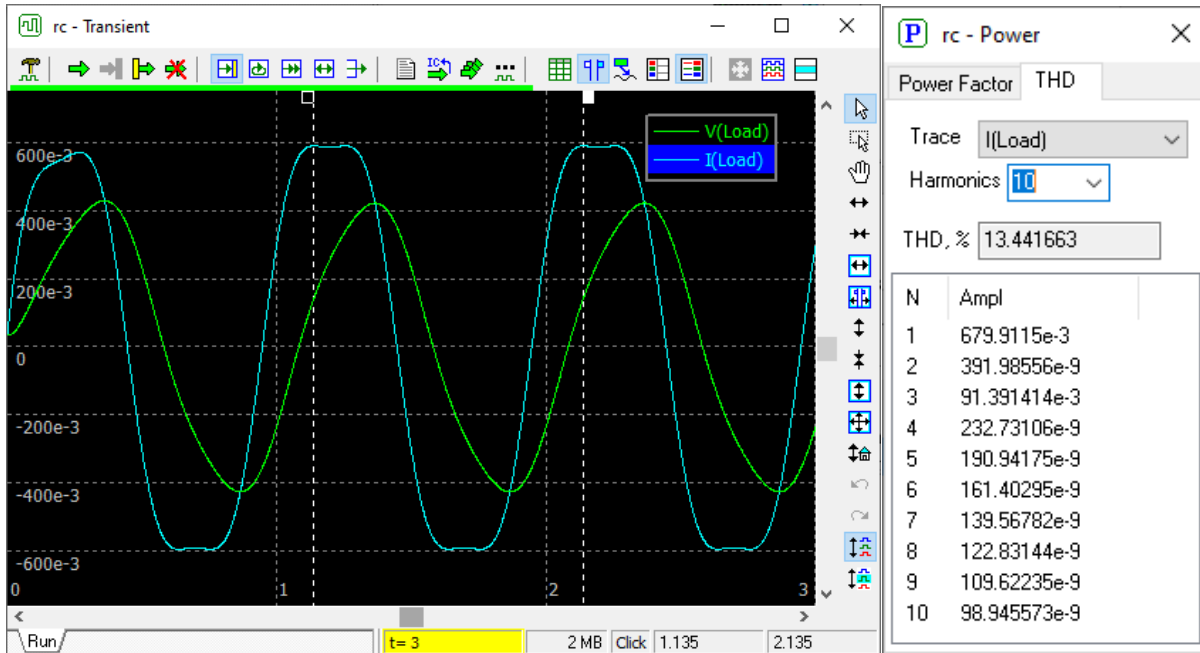


Select **THD** tab to calculate Total Harmonic Distortion.

- **Trace:** select trace from the drop-down list.
- **Harmonics:** enter or select from drop-down list number of harmonics to calculate.

**THD** is calculated as:  $THD = \frac{\sqrt{V_2^2 + V_3^2 + V_4^2 + \dots}}{V_1}$ , where  $V_n$  is amplitude of  $n^{\text{th}}$  harmonic.

THD is calculated between cursors. Please make sure that selected interval consists of exactly **1 period** of the signal fundamental (1<sup>st</sup> harmonic).



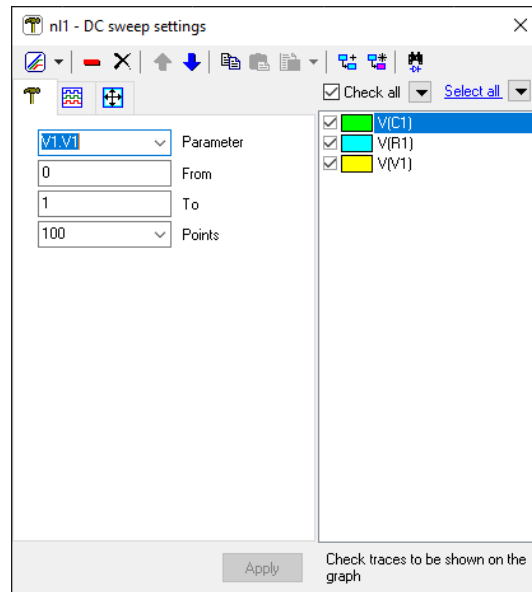
If calculation step is too big, higher harmonics may not satisfy Nyquist criteria: harmonic frequency exceeds half of discretization frequency. Such harmonics will be marked with an asterisk (\*) symbol in the harmonics list.

## DC sweep

Select **Transient/Transient tools**  or click **Transient tools**  on the **Transient window** toolbar, select **DC sweep** .

DC sweep tool performs calculation of DC operating point while changing specified component parameter or variable in specified range and plots transient traces as a function of that parameter. The name of the parameter is shown in the header of the DC sweep window.

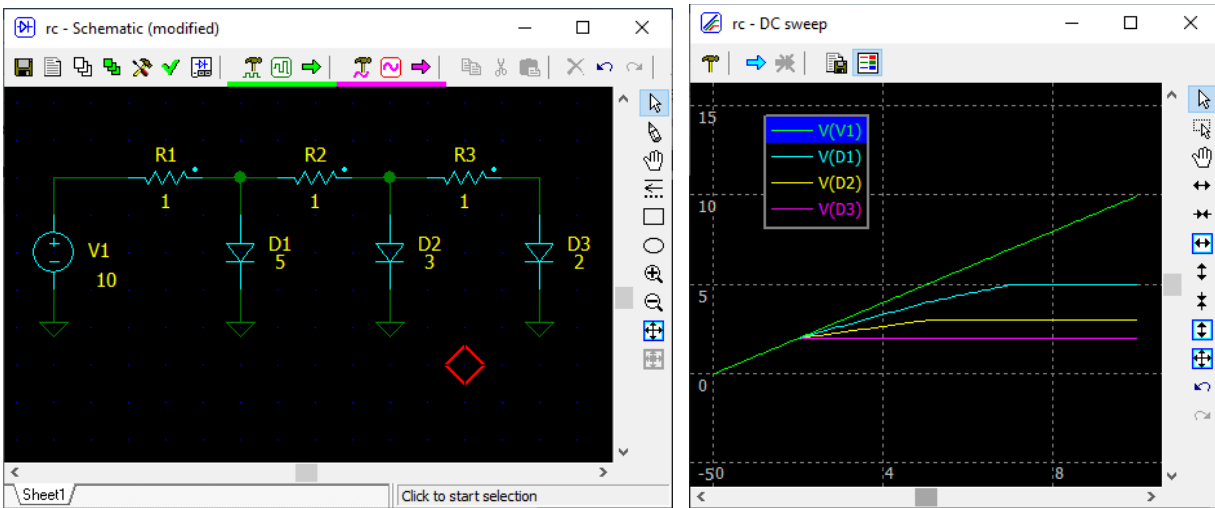
Click **Tool settings**  to configure DC sweep.



- **Parameter:** name of the parameter to be changed. Enter parameter name manually or select from drop-down list. The list shows parameters of the component which is currently selected in the schematic.
- **From, To:** range of parameter change.
- **Points:** number of points.

**Check** traces in the list to be shown on the graph.


Click **Start DC sweep**  to perform DC sweep.

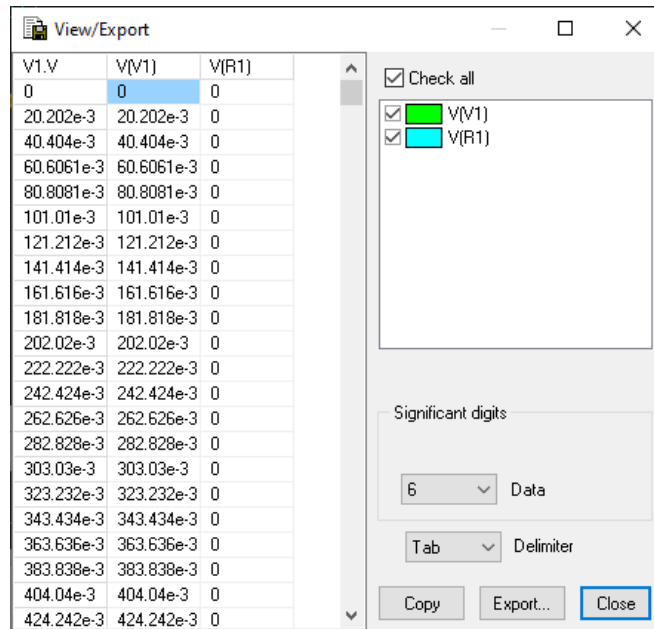


**Double-click** to center screen.

**Right-click** on the graph to show context menu.

Zoom, scroll, mouse, and mouse-wheel operations are the same as in Transient window.

**View/Export**  - view DC sweep data in the table and export into text file or copy to clipboard:



**Check** traces to be exported.

**Significant digits** - select significant digits for data.

**Delimiter** – select **comma** or **Tab** (use Tab to export data to MS Excel).

Click **Copy** to copy data to clipboard.

Click **Export** to save into the file. Select file name and extension (**csv**, **txt**).

# V. AC analysis


## Algorithm

There are two methods of AC analysis in NL5: **Linearize schematic** and **Sweep AC source**.


**Linearize schematic** is a standard low-signal AC analysis. First, all non-linear components are replaced with linear equivalents at their operating point. Second, a signal of specified frequency with unit amplitude and zero phase is applied to the input node, and signals at other nodes are found by solving a system of linear equations. The process is repeated for specified number of frequencies.

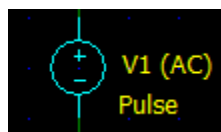
To linearize schematic, state of all components should be known. It can be done manually by setting Initial Conditions (IC) for non-linear components, diodes, and controlled switches, or by automatic calculation of DC operating point (**Calculate DC operating point** check box in the **AC Settings**). DC operating point is calculated exactly as in transient analysis.

**Sweep AC source** method allows calculating AC response for any type of circuit, including switching circuits. A real sine-wave signal of specified amplitude and frequency is applied to the input node; the transient analysis is automatically performed; finally, the harmonic of specified frequency is extracted from signals of interest by Discrete Fourier Transform (DFT). AC response is calculated by comparison of the harmonic amplitude and phase of the specified signal to the input signal. This process is repeated for specified frequency range.

The algorithm does not calculate “periodic operating point” which may be required for accurate results. It is highly recommended to run transient manually until periodic steady state is reached, and then save transient Initial Conditions .

Transient simulation is automatically started and controlled by the method. Currently selected transient simulation step will be used. Please note that all existing transient data will be deleted when AC simulation starts.

**AC source** . Any voltage source, current source, or label can be used as AC source. Component selected as AC source will be marked with “(AC)” text on the schematic and in the **Components window**:



R2	1
R4	1
V1 (AC)	Pulse

The AC source component may have any model (except **SubCir**, **Label**, and **IC**): the model will be ignored for **Linearized schematic** method and temporarily set to **Sin** for **Sweep AC source** method. DC voltage/current of the component will be set to its DC value at t=0, and AC voltage/current required for AC analysis will be added to that DC level.

**Traces.** NL5 only stores data for specified **traces**.

## Running AC analysis



- Go to **Settings window** to select simulation method, simulation parameters, traces, etc.



- **Start AC** (available on the **Schematic** and **AC window** toolbars). When simulation is running, schematic cannot be edited, and component parameters/model cannot be changed.



- **Stop AC**.



- **Sweep parameter**. Open **Tools** window, **Sweep** tab. See **Tools** chapter for details.








- **Show AC log**.



## AC window



Add AC traces using any of the following methods:




- Go to **Settings window** , **Add traces**  tab, see detailed instruction below.
- **Right-click** on the component in the schematic window, select **Add trace** , then select trace type.
- Select component in the **Components window**, click **Add trace**   button.

Go to **Settings window** , **Format traces**  tab to select trace color, scales, etc.



Go to **Settings window** , **Screen**  tab to configure **AC window** scales and other properties.

**Scales.** All traces are shown with the same scales, linear or logarithmic.


**Phase display method selection:**

- **Phase off**  - do not show phase, magnitude only.
- **Phase On**  - show magnitude and phase in the same window.
- **Phase separate**  - show magnitude and phase in separate sections of the window.

**Mouse.** There are 3 modes of **mouse operation**:

**Cursors**  mode. Some mouse operations apply to AC cursors .

- **Double-click** the screen to show cursors and/or reset cursors lock modes.
- **Click** on the screen to move closest cursor to that point.
- **Place mouse pointer** over the cursor, then click and drag to move the cursor.
- Press **Left/Right** arrow key with **Ctrl**, **Shift**, or **Ctrl-Shift** keys to move active cursor with different step resolution.

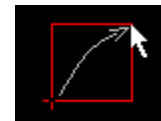
**Zoom**  mode (press and hold **Ctrl** key to switch to **zoom** mode temporarily). Click and drag to select horizontal, vertical, or rectangle area to be zoomed-in:




Move mouse pointer horizontally to select horizontal area.



Move mouse pointer vertically to select vertical area.



Move mouse pointer diagonally to select rectangle area.

**Scrolling**  mode (press and hold **Shift** key to switch to **scrolling** mode temporarily). Click and drag to scroll the screen.

**Mouse wheel:**

**Mouse wheel** – horizontal zoom (relative to mouse pointer position).

**Shift-mouse wheel** – horizontal scroll.

**Ctrl-mouse wheel** – vertical zoom (relative to mouse pointer position).

**Ctrl-Shift-mouse wheel** – vertical scroll.



**Keys and shortcuts:**


**F8** – show AC window  .

**F9** - start AC  .

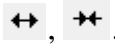
**Tab** – toggle **Phase** display mode  .

**Up, Down** – vertical scroll.

**PgUp, PgDn** – vertical zoom  .


**Home** – vertical fit the screen  .

**Left, Right** – horizontal scroll.

**Ctrl-PgUp, Ctrl-PgDn** – horizontal zoom  .

**Ctrl-Home** – horizontal fit the screen  .

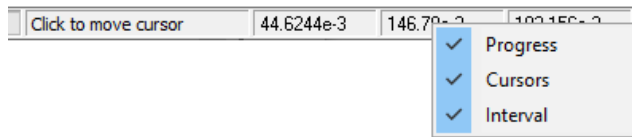
**Shift-PgUp, Shift-PgDn** – vertical and horizontal zoom.

**Shift-Home** – fit the screen  .

**Ctrl-Right, Ctrl-Left, Shift-Right, Shift-Left, Ctrl-Shift-Right, Ctrl-Shift-Left** - move selected cursor with different resolution.


## Status bar





Right-click status bar to configure:




- **Progress** – current frequency of AC simulation.
- **Cursors** – show cursors position (left and right).
- **Interval** – interval between cursors.



## Cursors

**Cursors**  specify interval that can be used for **AC table** and **AC tools**.


- Go to **Cursors**  tab of **Settings window**  to configure cursors options and perform some operations on cursors.
- Click **Show cursors**  to show/hide cursors. Please note that table and AC tools may use specified **Cursors** interval type even when cursors are hidden.
- Click **Cursors to screen**  to set horizontal scale to cursors position.

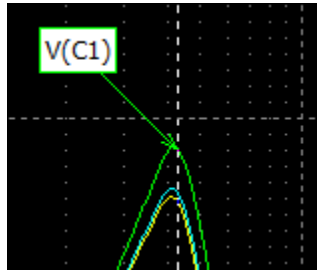
## Legend






**Legend**  shows list of traces currently displayed on the graph. Legend is also available in all **AC tools**.

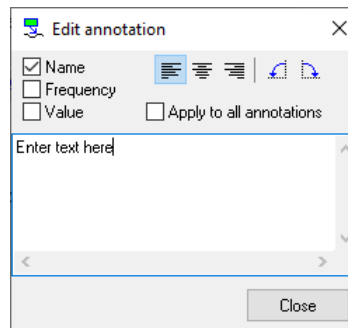
- Click **Show legend**  to show/hide legend.
- **Click** on the border of the legend and drag to move the legend window.
- **Click** on the trace in the legend to select one trace.
- **Ctrl-click** to select more than one trace or unselect the trace.
- **Right-click** to see context menu with more selection options.
- Go to **Preferences** , **Legend** tab to select legend style (colors, font). The style can be defined manually or selected from two predefined options (NL5 Ver.2 and NL5 Ver.3), and then adjusted manually if needed.

## Annotations

**Annotation**  is a text with a pointer, which always points to the same data point of a trace, even when graph is zoomed or scrolled:




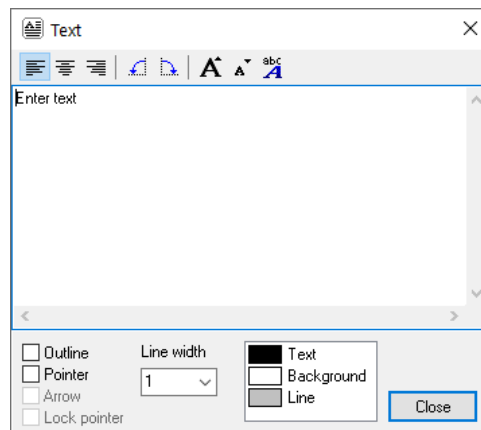
- To add annotation, set cursor to the frequency point where annotation is needed, **right-click** the graph, select **Annotate** , then select **Selected traces**  or **All traces** .
- Go to **Preferences** , **Annotation** tab to configure annotation colors, font, and some other common properties.
- **Double-click** on the annotation, or **right-click** the annotation and select **Edit annotation**  to edit (customize) specific annotation:






- **Click** and drag annotation text to move the text while keeping pointer at the same data point.
- **Click** and drag the end of the pointer to move annotation pointer to a different data point. The pointer will follow the mouse pointer, staying at the nearest point of the trace (trace data between data points are linearly interpolated).
- Annotations defined on the AC graph can also be displayed on the **Smith chart** and **Nyquist plot** AC tools.

## Text


**Right-click** on the graph, select **Insert text** :









If graph is zoomed or scrolled, the text stays at the same place, anchored to the left-top corner of the graph window.


- **Double-click** on the text, or **right-click** on the text and select **Edit text**  to edit or configure the text.
- **Click** and drag the text window to move it. If pointer is locked, only text will move.
- **Click** and drag the end of the pointer to move the pointer only.
- **Right-click** on the text and select **Copy text**  to copy text to clipboard. Then **Right-click** on the graph and select **Paste text**  to paste text on the graph.

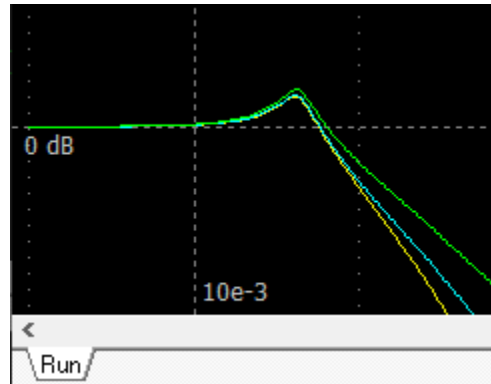
## Table

**Table**  shows cursors position, trace values, and some characteristics of the traces calculated in specified interval (**Cursors**, **Screen**, **All data**, **Selection**).

- Go to **Settings window** , **Table**  tab to configure the table.
- Click **Show table**  to show/hide the table.
- Click **Floating table**  to show the table in a floating window. Click  and  any time to switch between window modes.
- **Right-click** the table to see context menu with table-related commands.
- **Right-click** / **Copy** to copy content of the table cell to the clipboard.

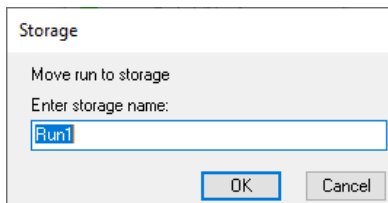
## Storage


**Storage**  can be used to keep the data for further analysis and comparison. When simulation is performed, AC data are shown on the graph, and a tab named **Run** corresponding to current transient data is shown in the **Storage tab**:



When a new simulation is started, current simulation data is deleted. To keep it for further analysis and comparison, it can be saved into **storage**.

**Right-click** the graph or **Storage tab**, and select **Move Run to storage** , then enter storage name:

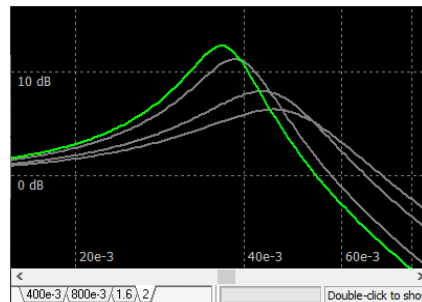


Simulation data will be moved to the storage, and a tab corresponding to the stored data will appear in the **Storage tab**. Also, storage is used for **Sweep parameter**  operation.

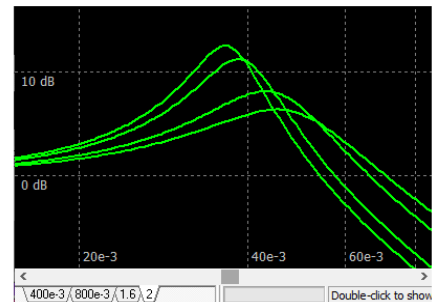
Stored data can be displayed in different ways, for example:





Selected only



Selected and dimmed

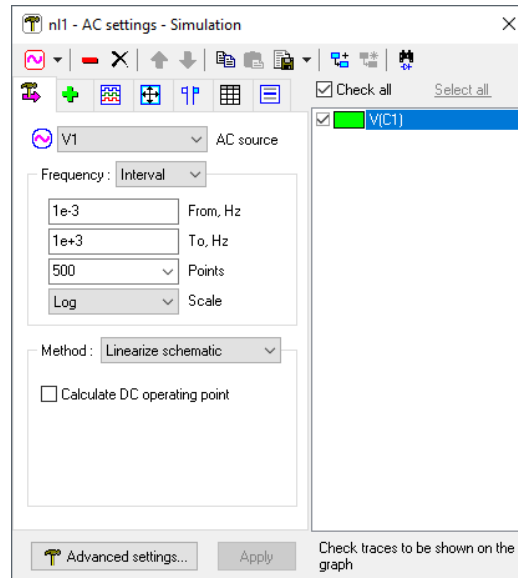


All

- Go to **Storage**  tab of **Settings window**  to configure storage.
- **Right-click** the graph or **Storage tab** for storage-related commands.



## AC settings

### Simulation



**AC source** – enter the name of AC source component or select the name from drop-down list.



AC source can also be selected using any of the following methods:

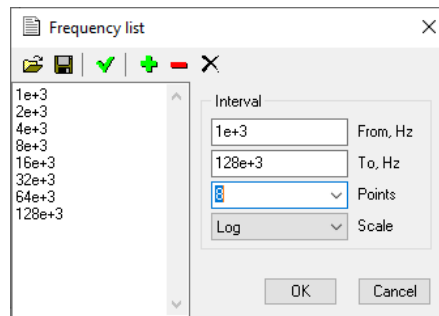
- **Right-click** on the component in the schematic window, select **Add trace** , then select **Set AC source** .

- Select component in the **Components window**, click **Set AC source**  button.

The component selected as AC source will be marked with **(AC)** text on the schematic and in the **Components window**.

**Frequency** - select how frequency simulation points are specified.

- **Interval**. Specify frequency interval, number of simulation points, and scale (**Lin/Log**).
- **List**. Specify custom frequencies. Enter frequencies one per line in any order. Click **Sort and Refresh**  to verify and sort entered numbers. Click **Edit frequency list**  to create and edit frequency list in the **Frequency list** window:

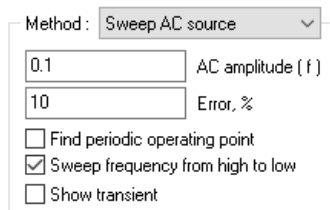


**Method** – select AC method.

### Linearized schematic:

- **Calculate DC operating point** – use this option for non-linear schematic if needed.

### Sweep AC source:



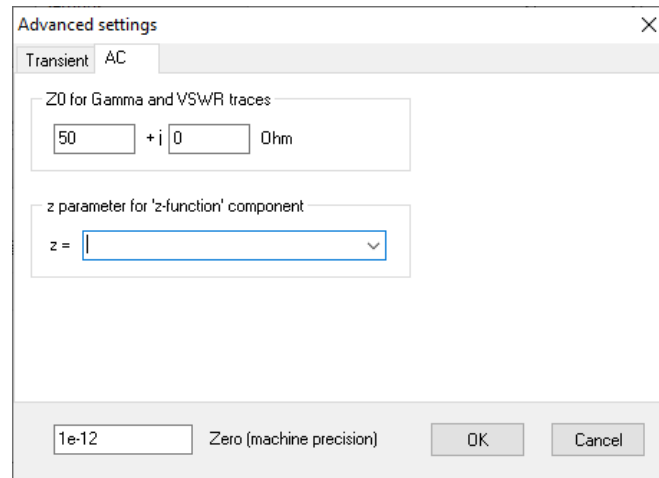
- **AC amplitude ( f )**. Amplitude of AC source, **V** for voltage source and label, **A** for current source. Amplitude can be a constant or function of frequency **f**. For example:

```
0.1
1m*f
1000/f
```

Frequency-dependent amplitude may be useful to improve signal-to-noise ratio for switching circuits with frequency-dependent gain.

- **Error, %**. Expected error of AC analysis.
  - **Error = 0** provides the most accurate simulation results. Transient simulation at each frequency will be performed for 128 periods of that frequency, which may result in a long simulation time.
  - **0 < Error < 100** provides reasonable accuracy for most schematics. Transient simulation at each frequency will be performed for 2...128 periods of that frequency; specific number of periods is automatically selected based on requested accuracy.
  - **Error = 100** provides fastest simulation. Transient simulation at each frequency will be performed for 2 periods of that frequency, which may result in poor accuracy.
- **Sweep frequency from high to low**. Perform AC simulation starting from higher frequencies. This option allows to start seeing simulation results faster.
- **Show transient**. Show Transient window during simulation and keep transient data obtained during AC simulation. Transient data might be useful for finding optimal AC analysis settings and for troubleshooting.

**Advanced settings**  - more **AC** settings.



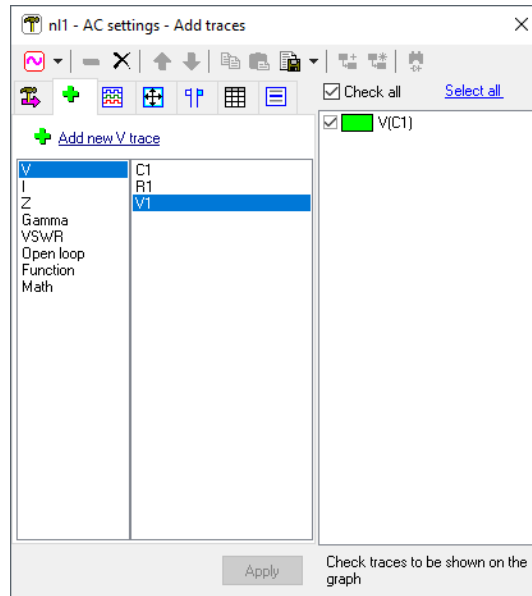
- **Z0 for Gamma and VSWR** - characteristic impedance for Gamma and VSWR traces.
- **Z parameter** - define custom formula for Laplace-space approximation of z-transform parameter z or select formula from drop-down list. For example:

```
exp (s*1e-6)
exp (s*T)
(2+s*T) / (2-s*T)
```

This parameter is used only by **z-function** components.



## Add traces



### V, I:

Select one or more components in the list, then click **Add new trace** or **double-click** on the component in the list.

To add trace in the **Schematic window**, **right-click** on the component, select **Add trace** , then select trace type.

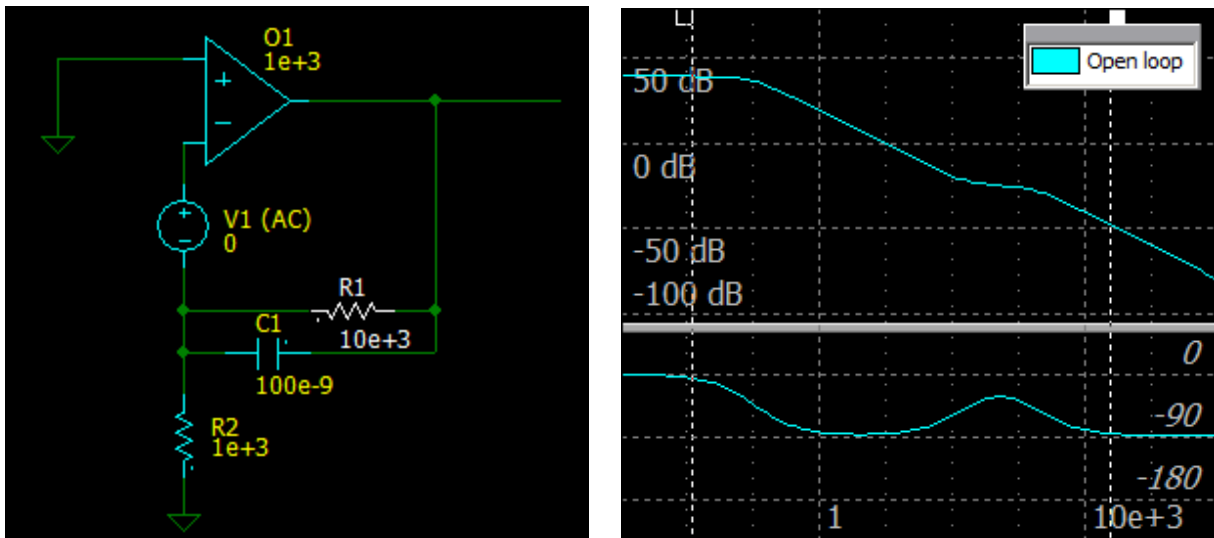
To add traces in the **Components window**, select component and click  or .

### Z, Gamma, VSWR:


Click **Add new trace** or **double-click** on the trace type name in the list. The trace shows impedance (**Z**), reflection coefficient (**Gamma**), or voltage standing-wave ratio (**VSWR**) relative to AC source.

### Open loop:

Click **Add new Open loop trace** or **double-click** on the trace type name in the list. **Open loop** trace shows open loop transfer function of the circuit operating in close loop mode. To use **Open loop** trace, insert AC source at proper place inside the closed loop, for example:



### Function:

Click  **Add new Function trace**, then enter function. The function may consist of arithmetic operators and functions, component parameters, global variables of **C-code** components, names of existing traces, V and I on the component, and the following variables:

- $f$  – current AC frequency.
- $\omega$  – angular AC frequency,  $\omega = 2\pi f$ .
- $s$  or  $p$  – Laplace parameter,  $s = p = j*2\pi f$ .

For example:

```
V(C1)/I(C1)
1/(1+s)
V(R1)*f
```

Please note that in **Linearized schematic** method, V/I on the component can be used even if V/I trace for that component is not defined. In **Sweep AC source** method V/I trace should be defined.

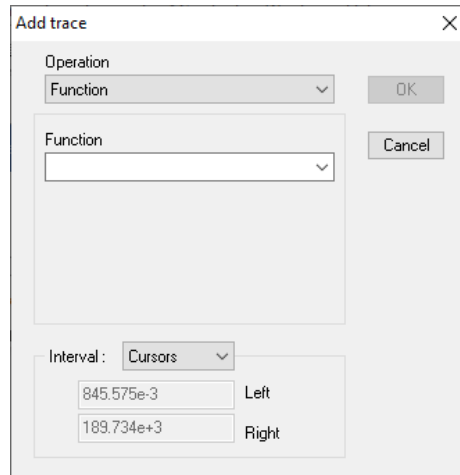
Use function to display V/I/P on the components inside subcircuits, for example:

```
V(X1.R1) - voltage at R1 in the subcircuit X1 .
```

The function is calculated only during simulation: if function is changed, it will not be immediately recalculated.

**Math:**

Click  **Add new Math trace**, then specify **Math** trace :



Select **Operation**, then enter parameters required for that operation (if any).

Specify **Interval** where **Math** trace will be calculated.

Some trace types (**Min, Max**) have a parameter **Show over entire range of the Trace**. If enabled, the trace will be calculated in the selected **Interval**, but displayed in the entire range of the trace.

The **Math** trace is **post-processing** type of trace: it is recalculated automatically when simulation completed, and if trace type, parameters, or interval changed.

The **Math** trace also applies to source **storage** data and is displayed as a **Math** trace storage. While normal trace storage is never changing, **Math** trace storage is automatically recalculated in case of operation or parameters changed, or trace range changed.

Details and examples of some **Math** types:

**Function.** Enter a function expression in the **Function** window. The function may consist of arithmetic operators and functions, component parameters, global variables of **C-code** components, any other traces, and the following variables:

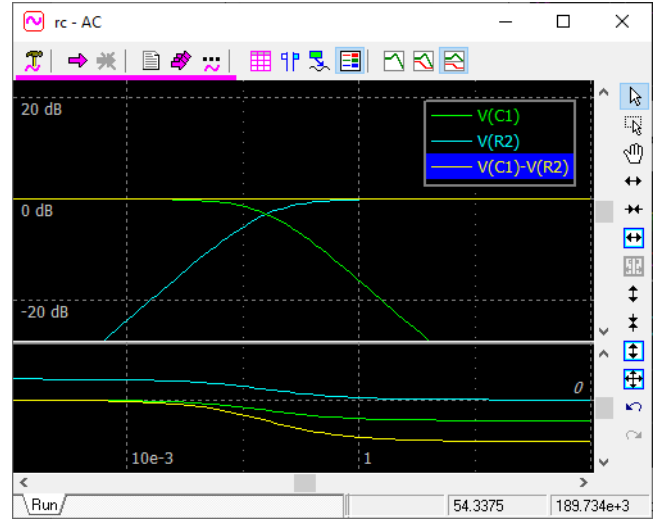
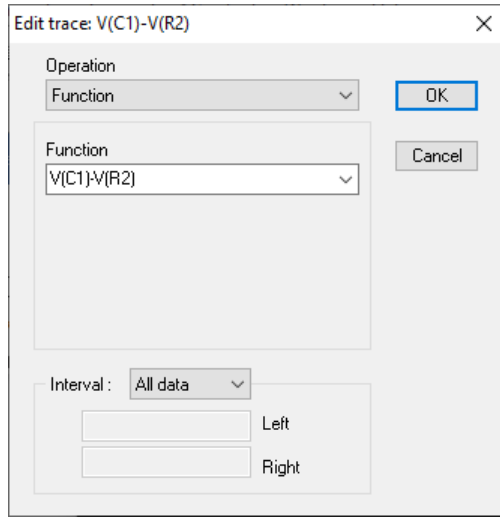
- $f$  – current AC frequency.
- $w$  – angular AC frequency,  $w = 2\pi f$ .
- $s$  or  $p$  – Laplace parameter,  $s = p = j*2\pi f$ .

For example:

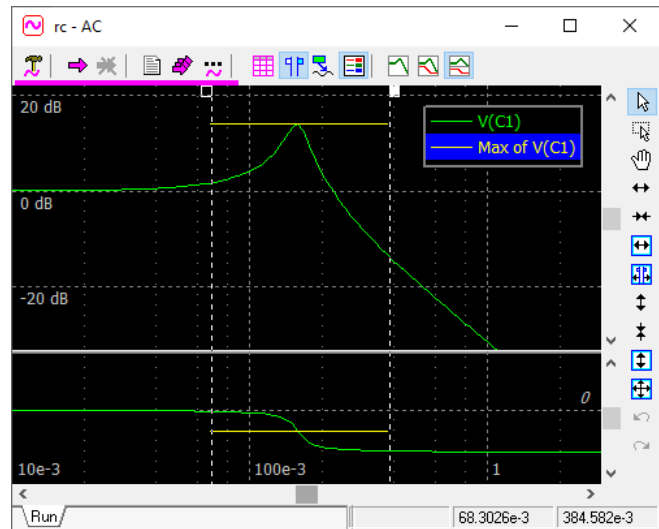
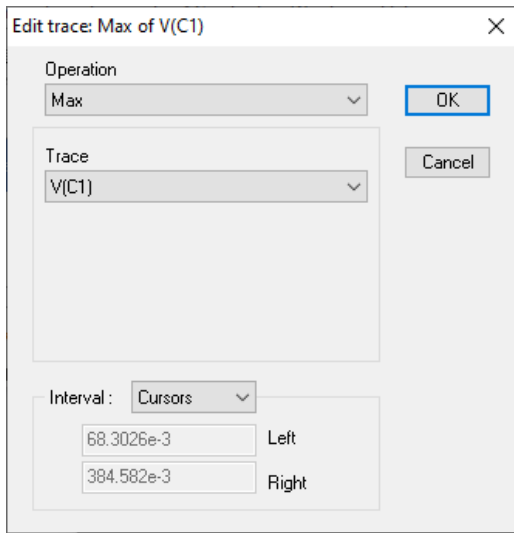
```
V(C1) / R1.R
1 / (1+s)
V(R1) * f
```

To access a specific storage data, use a `trace_name.storage_name` notation:

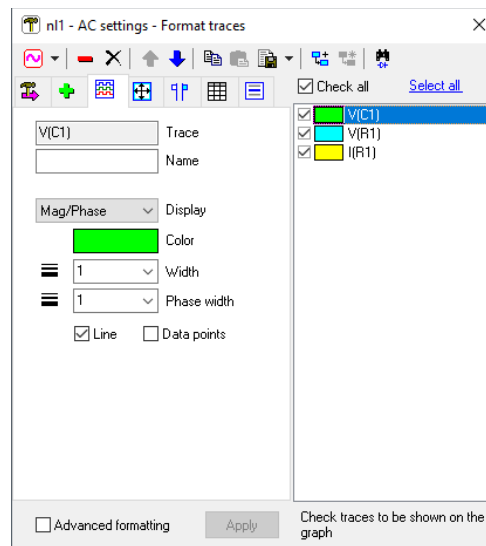
trace.Run1-trace.Run2



**Min, Max.** Calculate specified value of selected trace.



## Format traces




**Check** traces to be shown on the AC graph.

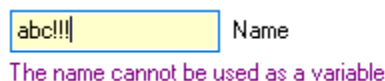
**Select** one or more traces (using **Ctrl** and **Shift** keys) to perform operations on selected traces. Click **Select All** to select all traces. Traces can also be selected in the **Legend** area of the AC window. The top toolbar provides various operations on selected traces and trace data.

If more than one trace of a different type is selected, only parameters applicable to **all** selected traces will be shown. If parameter has different values for different traces, its field will be blank.

If color is different, it will be displayed like this:  Color. If a new value is entered in an empty field, that value will be assigned to all selected traces.

**Trace/Function/Math.** This field indicates trace type and is available for editing for **Function** trace only. For **Function** and **Math** trace, click  to edit the trace.

**Name.** Enter trace name which will be shown instead of **Trace**. To be used in formulas, it should satisfy C-language standard for the variable name: start with letter or ‘\_’, and then contain only letters, digits, and ‘\_’. Otherwise, you will be given a warning:



**Display** defines how the trace will be displayed:

- **Mag/Phase** - magnitude and phase.
- **Re** – real part.
- **Im** – imaginary part.

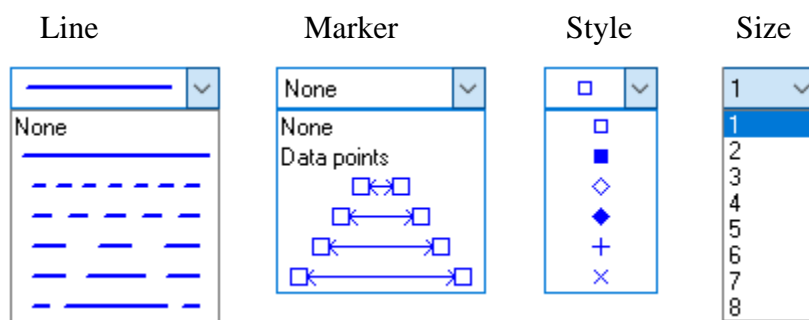
**Color** - click colored rectangle to select color of selected traces.

**Width** - enter or select from drop-down list width of selected traces (in pixels).

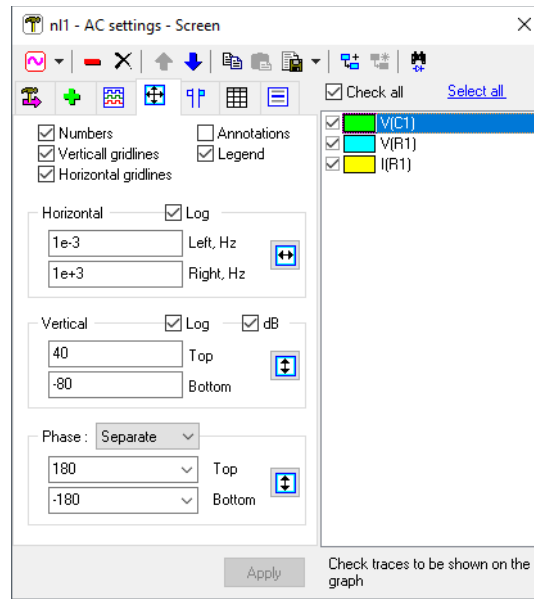
**Line** – show lines.

**Data points** – show simulation data points.

Select **Advanced formatting** to allow advanced formatting capabilities:



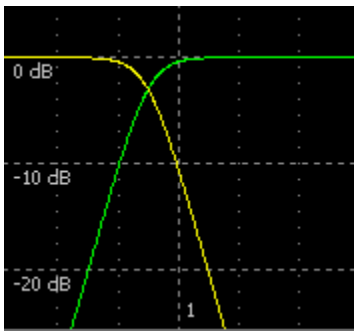
## Screen



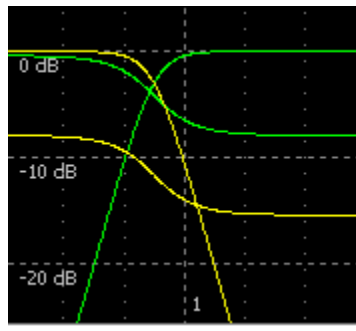
Select what is shown on the AC screen, set scales of the graph.

**Phase** - select phase display mode:

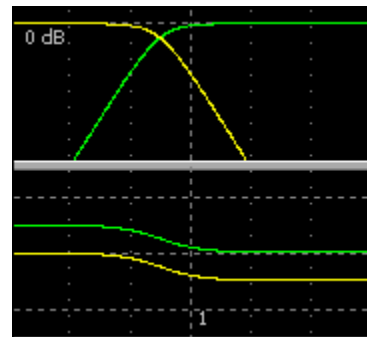
- **Off** - do not show phase.
- **On** - show magnitude and phase in the same area of the graph.
- **Separate** - show magnitude and phase in separate areas of the graph.



**Phase Off**



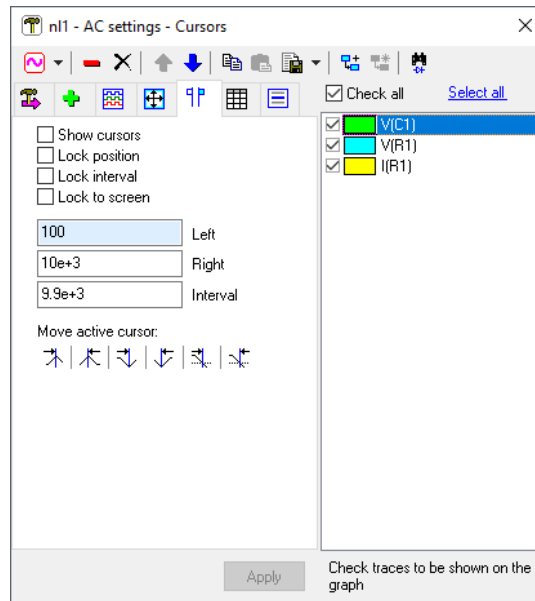
**Phase On**



**Phase separate**

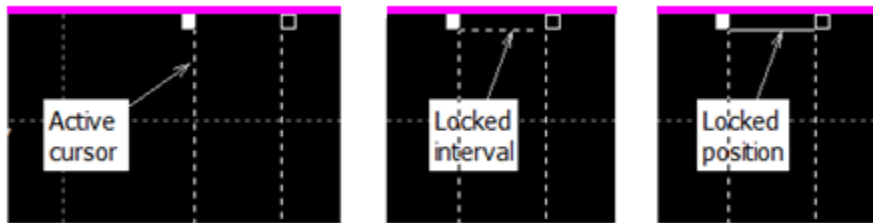
Press **Tab** in the AC Window to toggle **Phase** display mode.

# Cursors



**Lock position** – fix cursors position, do not move.

**Lock interval** – always move both cursors keeping same interval between them.



**Lock to screen** – keep cursors at screen edges.

**Left, Right, Interval** – enter position of the cursors or interval between cursors.

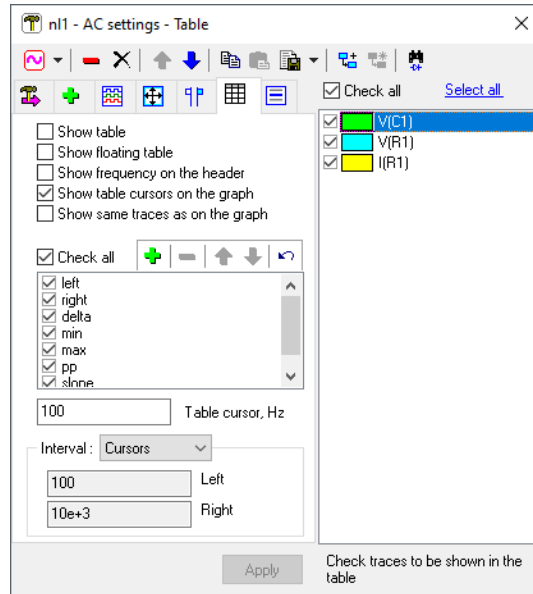
**Move active cursor:**

– move active cursor right/left to min/max data point of selected trace.

– move active cursor right/left to unity gain point of selected trace.

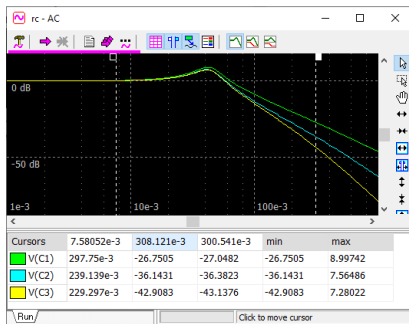


# Table

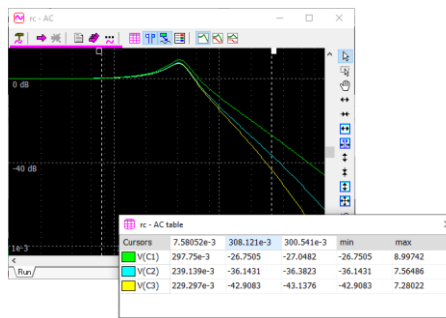


**Show table** – show/hide the table.

**Show floating table** – show/hide table in floating window.



**Table**



**Floating table**


**Show frequency on the header:**

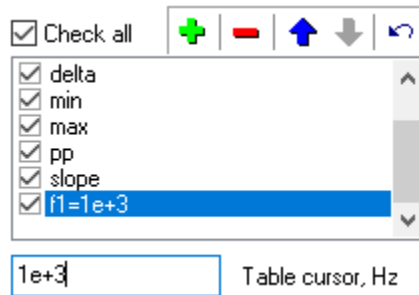
	left	right	delta	min	max
Cursors	7.58052e-3	308.121e-3	300.541e-3		
V(C1)	297.75e-3	-26.7505	-27.0482	-26.7505	8.99742
V(C2)	239.139e-3	-36.1431	-36.3823	-36.1431	7.56486
V(C3)	229.297e-3	-42.9083	-43.1376	-42.9083	7.28022

**Off**

Cursors	7.58052e-3	308.121e-3	300.541e-3	min	max
V(C1)	297.75e-3	-26.7505	-27.0482	-26.7505	8.99742
V(C2)	239.139e-3	-36.1431	-36.3823	-36.1431	7.56486
V(C3)	229.297e-3	-42.9083	-43.1376	-42.9083	7.28022

**On**

**Show table cursors on the graph.** Custom **Table cursors** can be added to the table. To add a table cursor, enter frequency in the **Table cursor** field, or set active cursor in the **AC window** to a desired frequency, and click **Add table cursor**  button:



A new item is added to the list. Now, the **f1** column in the table will show trace values at specified frequency. Select **Show table cursors on the graph** to show selected table cursors on the graph;



**Show same traces as on the graph** – if checked, the table will always show the same traces as shown on the graph.

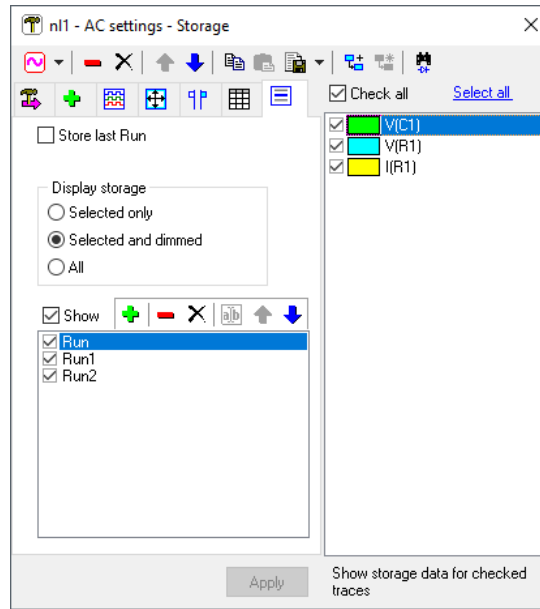
**Check** trace parameters to be shown in the table.

**slope** – magnitude slope between cursors, in dB/decade. If cursors are located at the same frequency, slope is calculated as magnitude derivative at this frequency, otherwise slope is calculated as  $(\mathbf{Mag}_{\text{right}} - \mathbf{Mag}_{\text{left}}) / (\mathbf{f}_{\text{right}} - \mathbf{f}_{\text{left}})$ .

Select **Interval** where table data are calculated.

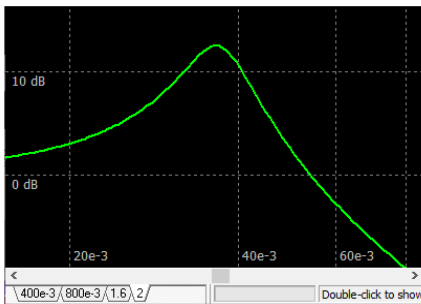
**Check** traces to be shown in the AC table. If **Show same traces as on the graph** is checked, checked traces will also be shown on the graph.

# Storage

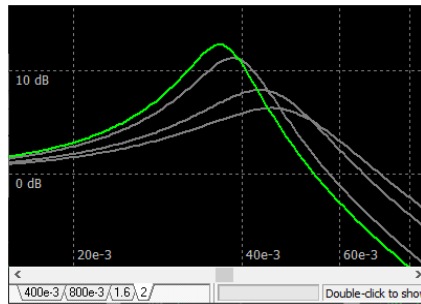


**Store last Run** – when AC starts, move current AC run to the storage under the name **Last**. You can always see the difference between a new and previous run.

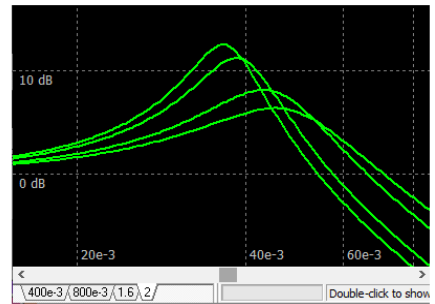
**Display storage** – specify how to show storage data on the graph:



**Selected only**



**Selected and dimmed**




**All**

**Check** storage data in list to show it on the graph and table.


**Check** traces to show storage data for those traces.


## More commands (toolbar).


**Copy selected traces**  - copy selected traces with data to the clipboard.

**Paste traces**  - paste traces with data from the clipboard.

**Data...**  **drop-down menu:**

**Open**  - open traces and data from a file of **nlf** type. You can select one or multiple files to open. Prefix “Copy of...” will be added to a trace name.

**Save**  - select traces and save traces with data in a file of **nlf** type, in special NL5 AC data binary format. Traces without data will not be saved.

**Import from file**  - import traces and data from text or binary files created by NL5 or other software tools/applications. You can select one or multiple files to open. The following file formats (file name extension) are supported:

- **txt, csv** – text format (comma-separated).
- **raw** – LTSpice data format.
- **qraw** – QSPICE data format.

### Text format (txt, csv).

Empty lines and comment lines started with ‘;’ symbol are skipped.

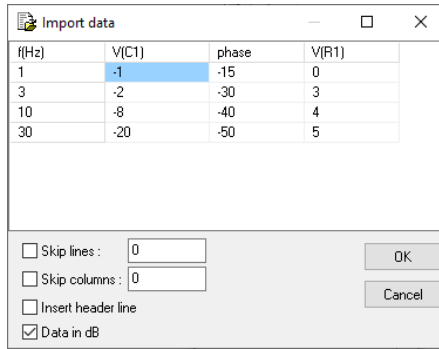
The first line (after skipped) can be a header line, having any text in the first column, followed by comma, space, or Tab separated names of the traces. If name of the trace consists of symbols other than numbers and letters, it should be enclosed in quotes.

All other lines are considered as trace data. The first column of the data consists of frequency (in Hz), other columns are trace data.

Trace data may have one column per trace, or two columns per trace. The first column is magnitude (absolute value, or dB), and it has trace name in the header line. The second column, if exists, is phase (in degrees), and it has “phase” text in the header line. If second trace column does not exist, the phase is set to zero. The data can be comma, space, or Tab separated. For example:

```
f (Hz) , V (C1) , phase , V (R1)
1, -1, -15, 0
3, -2, -30, 3
10, -8, -40, 4
30, -20, -50, 5
```

When file is selected, its content is shown in the window for verification:



**Skip lines** - specified number of lines in the beginning to skip.

**Skip columns** - specified number of columns starting from the left to skip.

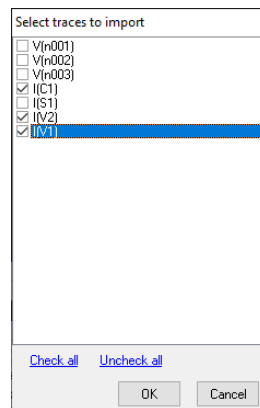
**Insert header line** – add header line “f (Hz) , trace1, trace2 ...”.

**Data in dB** - a magnitude is dB.


Check traces and Click **OK** to confirm import.

**LTSpice format (raw), QSPICE format (qrow).**

List of traces (variables) available in the file will be shown in the **Select traces to import** window. Check traces and click **OK**:

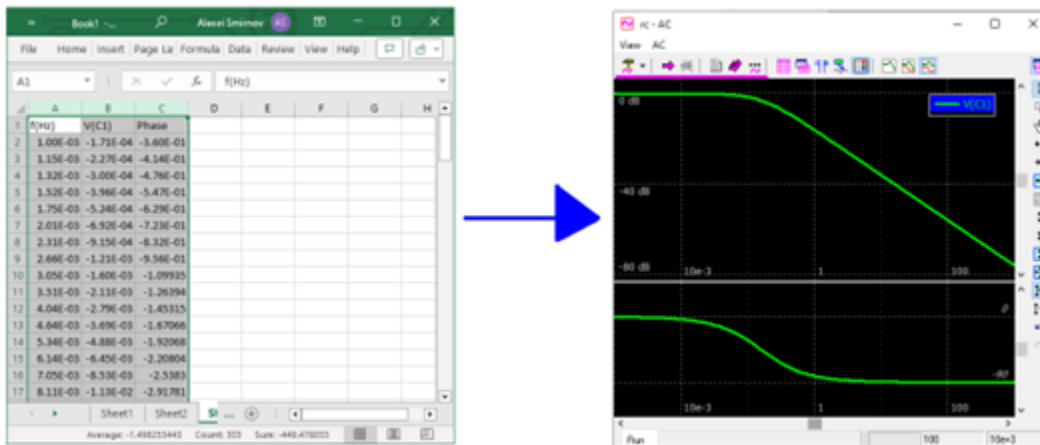



Check traces and Click **OK** to confirm import.

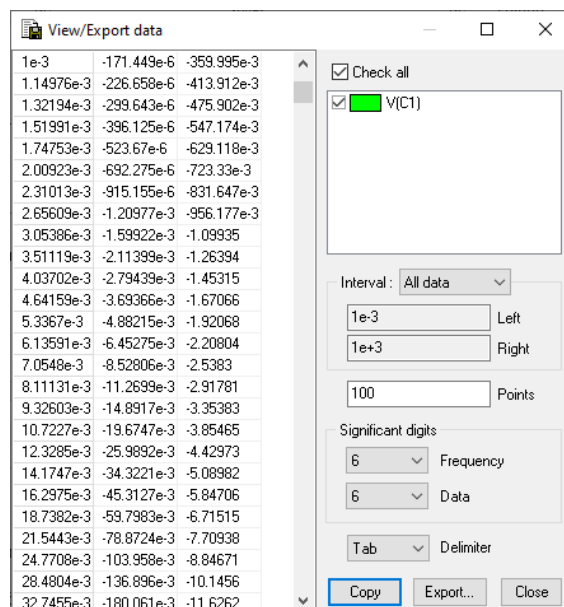
**Import from clipboard**  - paste traces in text format from the clipboard. The same selection window as for **Import from file** command will be shown with data obtained from clipboard.

Both **comma** and **Tab** delimiters of the text are accepted.

For example, you can select a range with trace data in MS Excel, **Copy** data to clipboard, then **Import from clipboard** to NL5 AC window:



**View/Export data**  - view trace data in the table and save in a file or copy to clipboard in text format. Specify export parameters:



**Check** traces to be exported in the list.

**Interval** – data interval to be exported.

**Points** – number of frequency points to view/export. AC data between calculated data points are linearly interpolated.

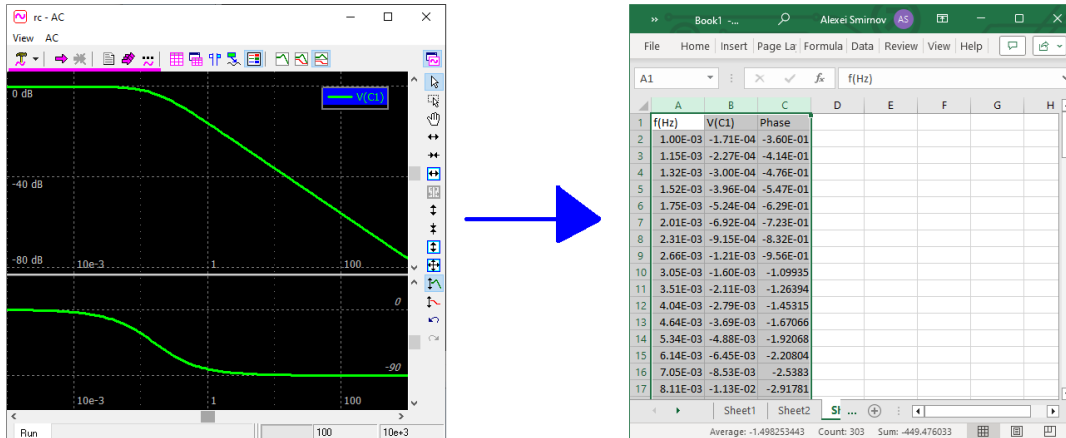
**Significant digits** - select significant digits for time and data.

**Delimiter** – select **comma** or **Tab**.


Click **Copy** to copy data to clipboard.

Click **Export** to save into the file. Select file name and extension (**csv**, **txt**).

To transfer data from NL5 to MS Excel, set **Tab** delimiter, **Copy** data to clipboard, then **Paste** data to Excel spreadsheet:







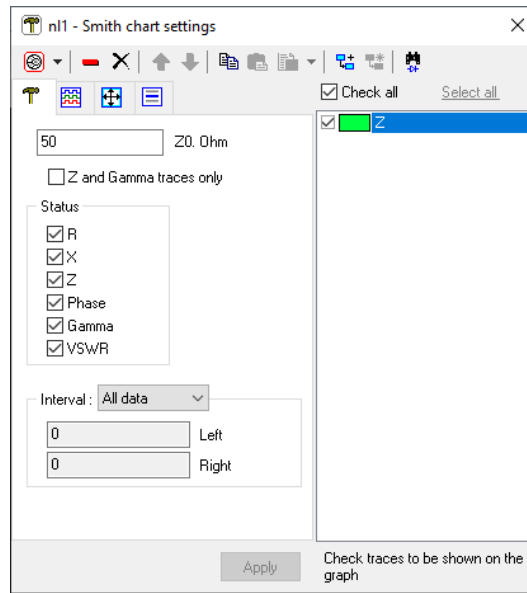
**Duplicate selected traces**  - duplicate (same as **Copy/Paste**) selected traces with data.

**Find component**  - show (select) the component used for V/I/Z/Gamma/VSWR/Open loop trace on the schematic.

# AC Tools

## Smith chart

Select **AC/AC tools**  or click **AC tools**  on the **AC window toolbar**, select **Smith chart** . Click **Tool settings**  to configure Smith chart.



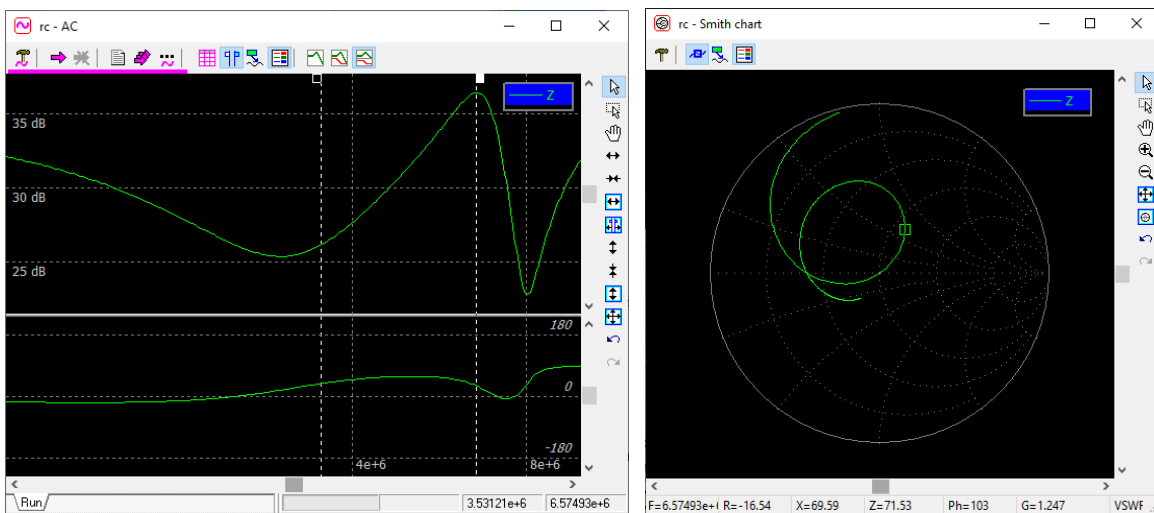
**Z0, Ohm** - characteristic impedance of the Smith chart.

**Z and Gamma traces only** - if selected, show only Z (impedance) and Gamma (reflection coefficient) traces, otherwise show all traces interpreted as complex impedance.

**Status** - what to show on the window status bar. Values shown correspond to mouse pointer position.

Select **Interval** of AC data to be shown in the Smith chart.

**Check** traces in the list to be shown on the graph.






Zoom, scroll, mouse, and mouse-wheel operations are the same as in AC window.

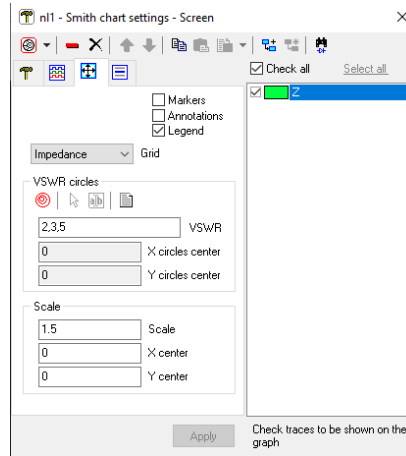
**Double-click** to center screen.

**Right-click** on the graph to show context menu.

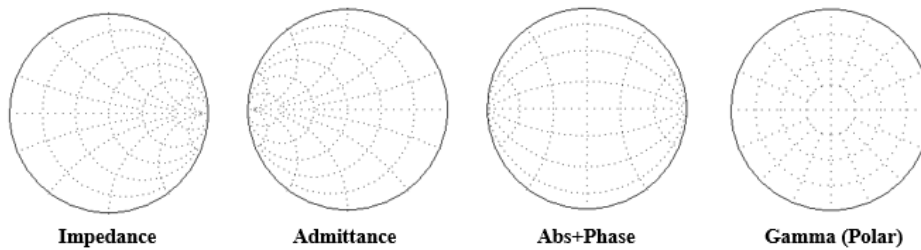
**Show markers**  - show markers corresponding to active cursor in AC window.

**Show annotations**  - show annotations defined in AC window.



Click **Screen**  to see **Screen** tab of Settings window:




**Grid** – select grid type (or press **Tab** in Tool window to toggle grid type):

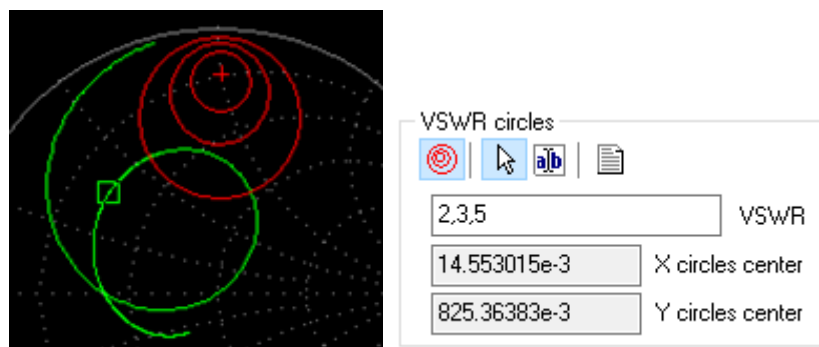


**VSWR circles.**





 - **Show VSWR circles** with specified VSWR values around selected point. Enter VSWR values in the **VSWR** field (comma-separated) or click **Edit VSWR circles**  to enter values in the list.

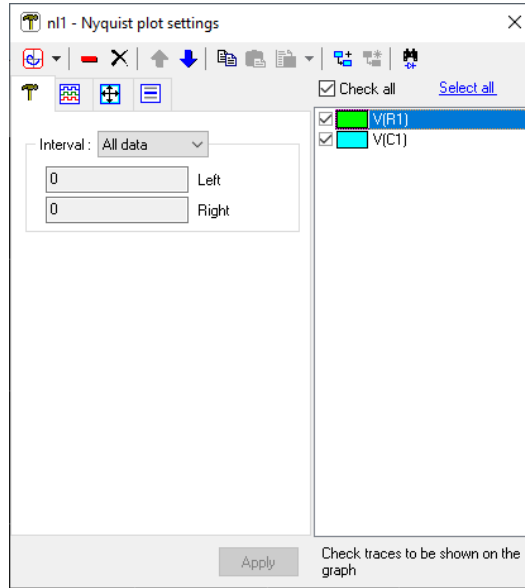
 - **Click on the chart to center VSWR circles.**

 - **Edit VSWR circles center.** Enter the center in the **X** and **Y circles center** fields.

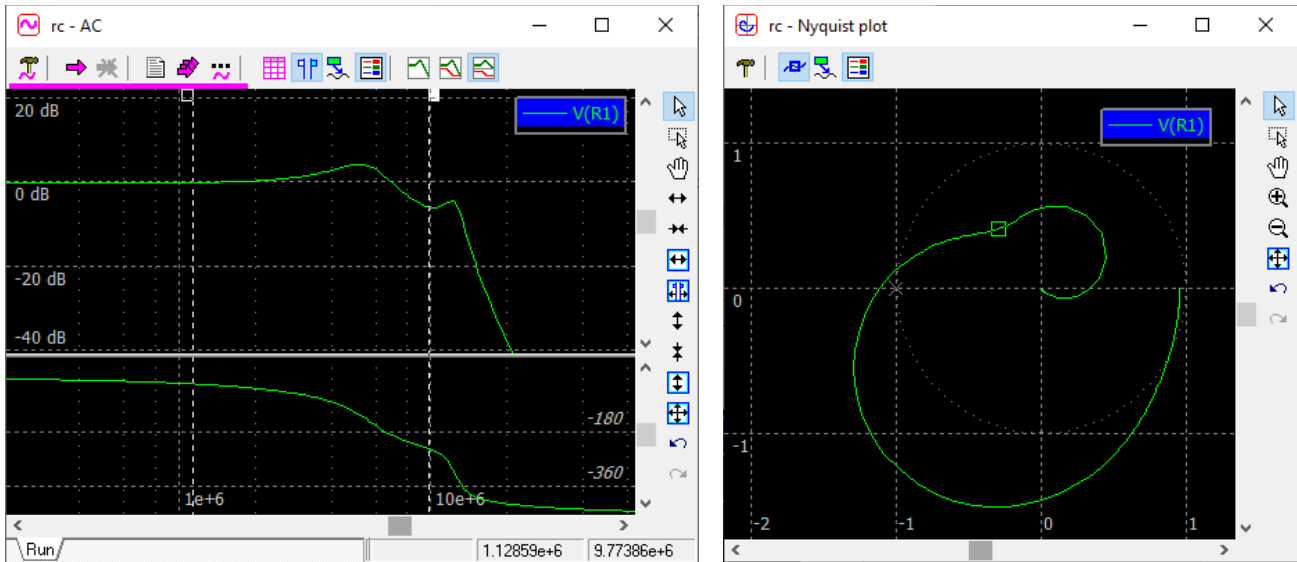


## Nyquist plot

Select **AC/AC tools**  or click **AC tools**  on the **AC window** toolbar, select **Nyquist plot** . Click **Tool settings**  to configure Nyquist plot.



Select **Interval** of AC data to be shown in the Nyquist plot.  
**Check** traces in the list to be shown on the graph.







Zoom, scroll, mouse, and mouse-wheel operations are the same as in AC window.

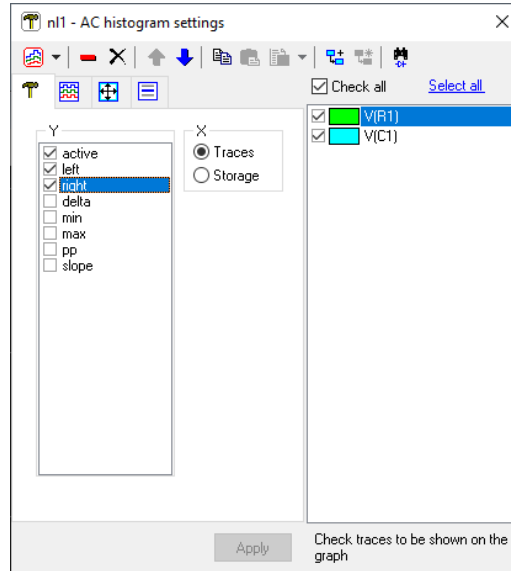
**Double-click** to center screen.

**Right-click** on the graph to show context menu.

Press **Tab** to show/hide unity circle.

## AC histogram

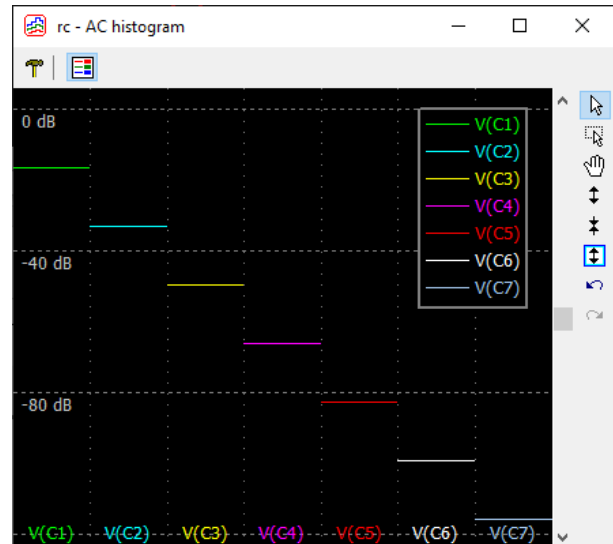
Select **AC/AC tools**  or click **AC tools**  on the **AC window toolbar**, select **AC histogram** . Click **Tool settings**  to configure AC histogram.



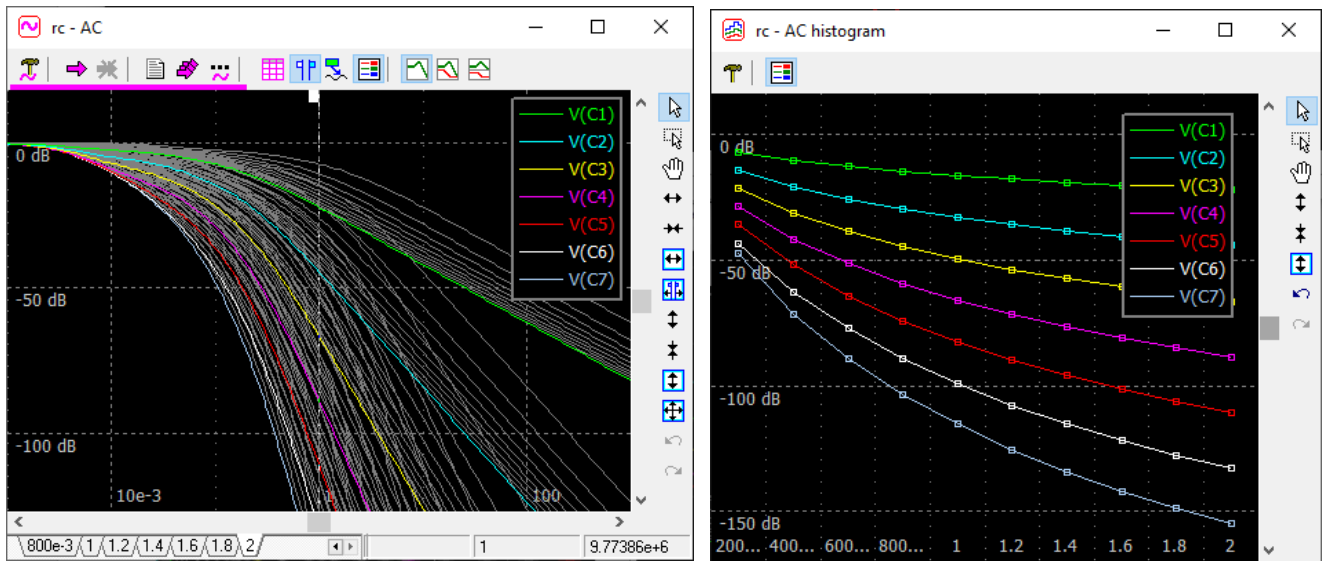
**Y** - histogram will display data calculated in the interval specified for AC table (AC table does not need to be shown).

**X:**

- **Traces:** show data as function of traces:




- **Storage:** show data as function of storage:



**Check** traces in the list to be shown on the graph.


Zoom, scroll, mouse, and mouse-wheel operations are the same as in AC window.

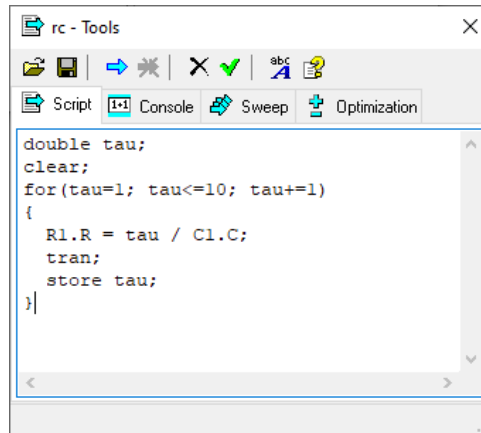
**Right-click** on the graph to show context menu.


On the **Screen** tab  of Settings window, select **Trace** type: **Line** or **Histogram**.

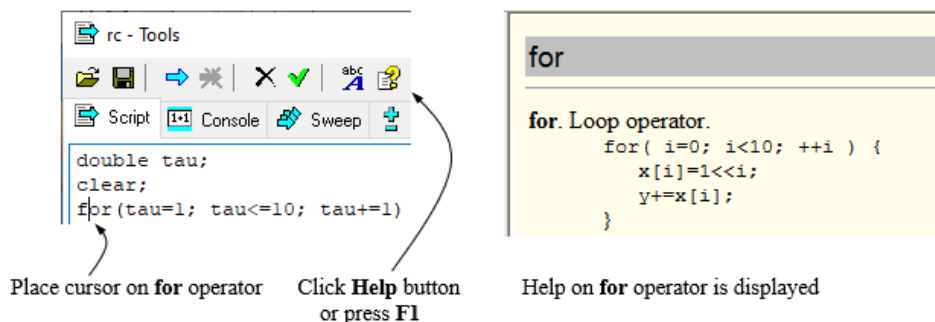
# VI. Tools

## Script

**Script**  is a program written in C language. In addition to standard operators and mathematical functions, it may use NL5-specific **script commands**, to change schematic parameters, run simulation, analyze data, and save simulation results.




To get help on specific operator, function, or command place cursor on the word in the script text and press **F1** or click **Help**  button:




Click **Run script**  to run the script. Click **Stop script**  to stop the script. Script is applied to the active document.

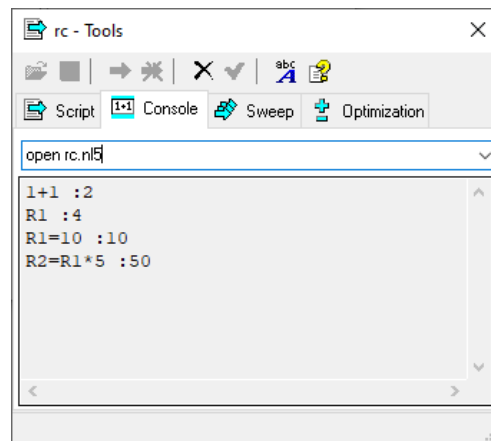
When transient simulation commands `tran` or `cont` are executed by the script, transient mode is automatically set to **Single**, so that transient will always pause at the end of the screen. The next script command will not be executed until transient simulation is paused.


When AC simulation command `ac` is executed by the script, the next script command will not be executed until AC simulation is complete.

Script log and error messages are displayed on the **Console**  tab.

## Console

**Console**  has a command line, which can be used as a calculator or execute some script commands.



Similar to **Script** tab, to get help on specific operator, function, or command place cursor on the word in the command line and press **F1** or click **Help**  button.

**Command line** can be used to evaluate expressions. Expression may contain:

- Numbers (including complex numbers).
- Component parameters and schematic variables of active document.
- Arithmetic operators and functions.
- Some script commands.
- Trace data.

Enter expression in the command line and press **Enter** to evaluate:

```
1+1 :2
```

To change component parameters or variables of active document, enter parameter name followed by equal sign and expression:


```
R1=10 :10
R2=R1*5 :50
```

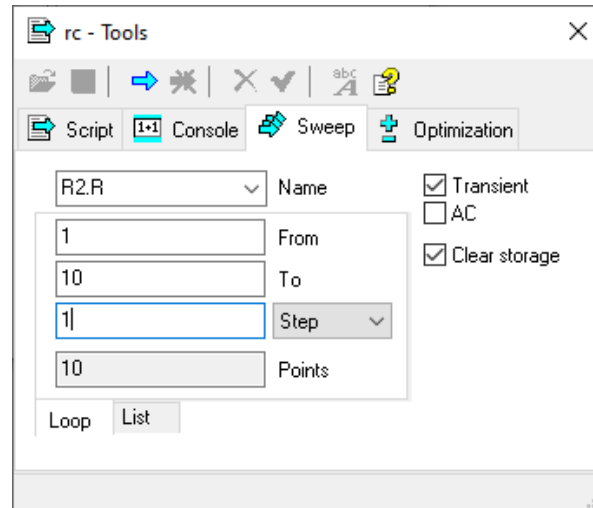
You can also execute script commands from command line (all commands except `logdata` and `exit`). In addition, `pause` command will pause transient simulation. Commands are applied to active document. For example:

```
open rc.nl5
tran 0,1m,.1u
store
cont
```

Previous expressions and commands can be retrieved from drop-down list or by pressing **Up** and **Down** arrow keys. Press **Esc** to clear command line.

## Sweep

**Sweep**  tab can be used to change component parameter in specified range, perform transient and/or AC analysis, and stores traces into storage.



Click **Run script**  to run the sweep. Click **Stop script**  to stop the sweep. Sweep is applied to the active document.

**Name.** Name of the parameter to be changed. Enter parameter name manually or select from drop-down list. The list shows all numerical parameters of the selected component.

**Transient.** Select to run transient.

**AC.** Select to run AC analysis.

**Clear storage.** Select to clear storage when sweep starts. If not cleared, sweep runs will be added to existing storage.

There are two types of sweep: **Loop** and **List**.

### Loop.

**From.** Start parameter's value.

**To.** End parameter's value.

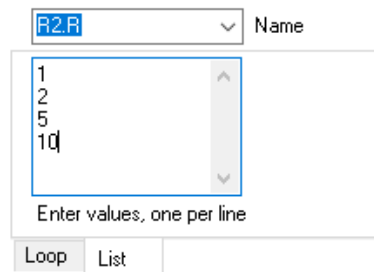
#### Step type:

- **Step.** Parameter is incremented (or decremented) by **Step** value.
- **Step, %.** Parameter is changed by specified **percentage**.
- **Step, X.** Parameter is **multiplied** by specified value.

**Points** field shows total number of steps to be performed.

**List.** Enter parameter values in the text field, one value per line:







R2.R Name


1  
2  
5  
10

Enter values, one per line

Loop List


Sweep results will be saved into storage. Only traces with enabled storage will be saved.

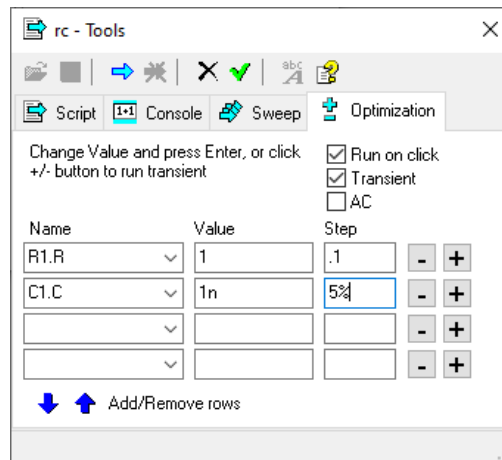
**Script**  tab contains text of the script that was executed, and **Console**  tab contains log of the script execution.



Transient **Sweep parameter**  command opens sweep tool with **Transient** checkbox selected.

AC **Sweep parameter**  command opens sweep tool with **AC** checkbox selected.

## Optimization

**Optimization**  performs transient and/or AC analysis while manually iterating selected parameters.






Select up to 16 schematic parameters to iterate. Use  and  buttons to add and remove rows with parameter values. If all rows do not fit the window, warning message will show up in the status bar: “Resize window to see more rows and controls”.




Specify the following parameters:

**Name** - name of the parameters to be changed. Enter parameter’s name and press **Enter** or select parameter from drop-down list. The list shows all numerical parameters of the selected component.


**Value** - current values of specified parameters.

**Step** - how the value will change when **Plus**  or **Minus**  button is clicked. If **Step** is a number, the number will be added to current value. If **Step** is a number with ‘%’ character at the end, the value will be changed by specified percentage.

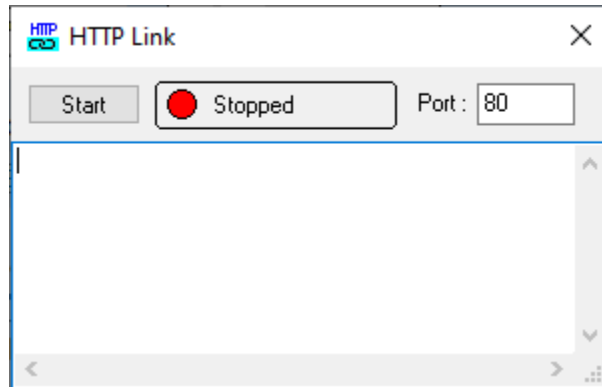
Click  to clean up and update optimization table. It removes rows with empty name field, moves rows up to fill up gaps, and fills in **Value** fields with current component values.

If **Run on click** is selected, specified analysis will be started immediately when **Plus**  or **Minus**  button is clicked, or when **Enter** key is pressed at the **Value** field. Otherwise, click **Run script**  to run analysis.

## HTTP link

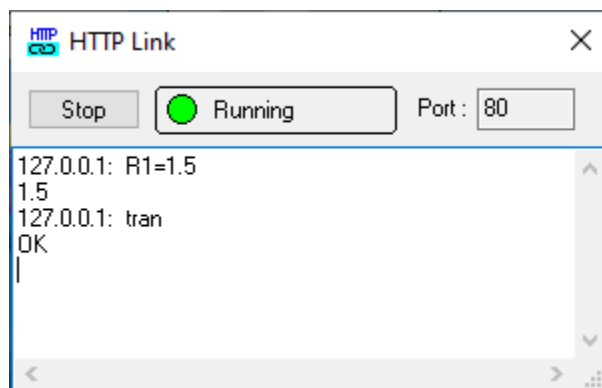
**HTTP link**  provides remote access to NL5 from external application. NL5 serves as a “server”, running built-in HTTP server. External application is a “client”. NL5 and a client application may run on the same computer or on different computers, communicating through local network or Internet.

Select **Tools/HTTP Link** or start NL5 in command line with `-http` or `/http` switch to open **HTTP Link** window:



Default **Port** number is 80, which is standard port for HTTP protocol. If this port is already used, select any other available port number. Make sure the client application uses the same port number as NL5 in URL request. Please note that port number can be changed only before HTTP link is started the first time. To change the port number after that, close NL5 application and start it again. If NL5 and client application are running on different computers, make sure that any existing firewalls and routers are properly configured to allow TCP communication through specified port.

Click **Start** to start HTTP server. While server is running, the window will display IP address of the client, command received, and result sent to the client:



To disable the HTTP log, uncheck **Show log** option on **HTTP server** tab of **Preferences** window, **Right-click** on the text area and select **Clear log** to clear the HTTP log.

The window can be closed at any time without affecting server operation. Click **Stop** to stop HTTP server.

**Sending URL request.** Client applications can issue commands and obtain data from NL5 by sending URL request with parameters and receiving back response in the text format. The “URL read” function name, syntax, and parameters may vary for different applications. For example, in the MATLAB `urlread` function can be used:

```
s = urlread('url')
```

**Please be aware that according to recent MATLAB guidance, function `urlread` is not recommended. Use `webread` instead.**

URL string has the following general format:

```
http://host_name:port/?cmd=command
```

where:

- **host\_name** is the name or IP address of the computer where NL5 is running. If client application is running on the same computer with NL5, **host\_name** can be `127.0.0.1`, or `localhost`.
- **port** is the port number. It should be the same as port number specified in NL5. If default HTTP port number (80) is used, it can be omitted, so that URL string will look as follows:

```
http://host_name/?cmd=command
```

- **command** is a script command or an expression.

NL5 executes command or evaluates expression and responds with a text, which can be:

- “OK”.
- result of the expression in text format.
- comma-separated string for trace data request.
- error message (text).

For example:

**Request:**

```
http://localhost/?cmd=open rc.n15
http://127.0.0.1/?cmd=R1
http://192.168.0.1/?cmd=C1=2n2
http://public025:2119/?cmd=V(C1)%200,1,.2
http://localhost/?cmd=open test.n15
```

**Response:**

```
“OK”
“100”
“2.2e-9”
“0,9.99999424754,9.9999944731,…”
“Error opening file test.n15”
```

If client application allows, the following modifications can be applied to URL string:

- Text “http://” can be omitted.
- Text “cmd=” can be omitted.

For example:

**Request:**

```
127.0.0.1/?open rc
192.168.0.101/?R1
http://public025:222/?store R1
```

**Response:**

```
“OK”
“100e+3”
“OK”
```

Please be aware that not all characters can be transmitted over Internet in the URL request. Some characters are “reserved” and have a special meaning. Being used in the NL5 command, those characters may be misinterpreted and result in wrong operation. For instance, character ‘+’ is typically interpreted as a “space”. This makes it impossible to use exponential form of a number: “1.23E+6” will be interpreted as “1.23E 6” and will produce HTTP server error.

To be used in the NL5 command, reserved characters should be converted into “URL safe” format, where the character is represented by ‘%’ symbol followed by two-digit hexadecimal number: ASCII code of the character (“percent-encoding”).

In many cases, the only reserved characters to be used in the NL5 commands would be *space* and ‘+’. To ensure correct operation replace *space* with its code %20, and ‘+’ character with %2B. For instance, “1.23E+6” should be sent as “1.23E%2B6”.

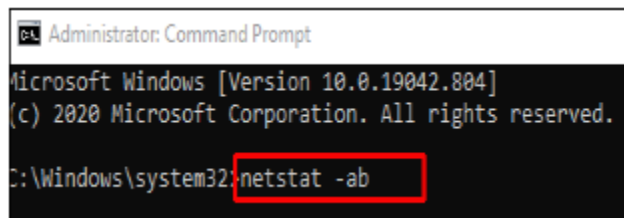
All reserved characters and their codes can be found in public Internet resources.

**Running simulation.** When transient simulation command `tran` (or `cont`), or AC simulation command `ac` is issued through HTTP link, NL5 sends “OK” response immediately without waiting for simulation to complete. This is done to avoid time-out condition in case of long simulation time. The client should wait for simulation end by periodically sending command `ready` and checking NL5 response. Response “0” means that simulation is still running, response “1” means that simulation is completed, and a new command can be issued. If the error occurs during simulation, NL5 responds with error message.

**Reading simulation data.** Please note that length of the returned string may be limited, and the limit may be different for different applications. If the size of requested data is too big, you may get a time-out for your request. To fix this issue, reduce the number of data points requested by one command.

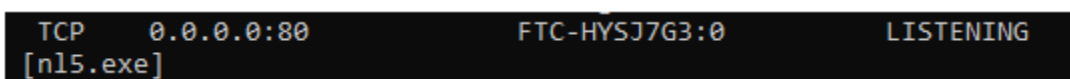
**Troubleshooting.** If there are any problems with obtaining response from NL5 with HTTP link, please try the following:

1. Start NL5 "As administrator" - it helps avoiding many problems related to Win 10 security.
2. Start HTTP server (**Tools/HTTP Link, Start**).
3. In Windows: open “Start menu,” type “Command Prompt,” then select “Run as administrator.”
4. Type “netstat -ab” without quotes, then press “Enter.”:



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Windows\system32>netstat -ab
```

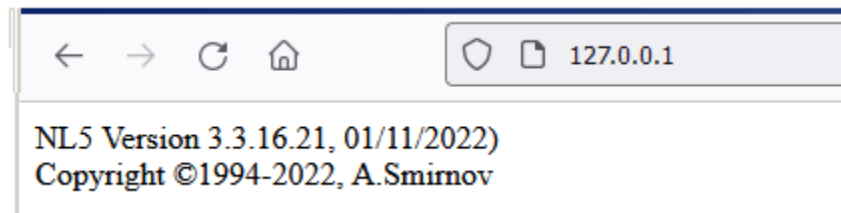
5. Wait for the results to load. Port names will be listed next to each local IP address. You should see some text which says that nl5.exe is listening port 80 (TCP X.X.X.X:80)



```
TCP 0.0.0.0:80 [nl5.exe] FTC-HYSJ7G3:0 LISTENING
```

If there is no such line, then NL5 server probably is not running. Check the list to see if there is a line with port 80 used by another application. If so, close that application, or use another port in NL5 (**Tools/HTTP Link, Port**).

6. Open some browser and enter address 127.0.0.1, you should see response from NL5 server:

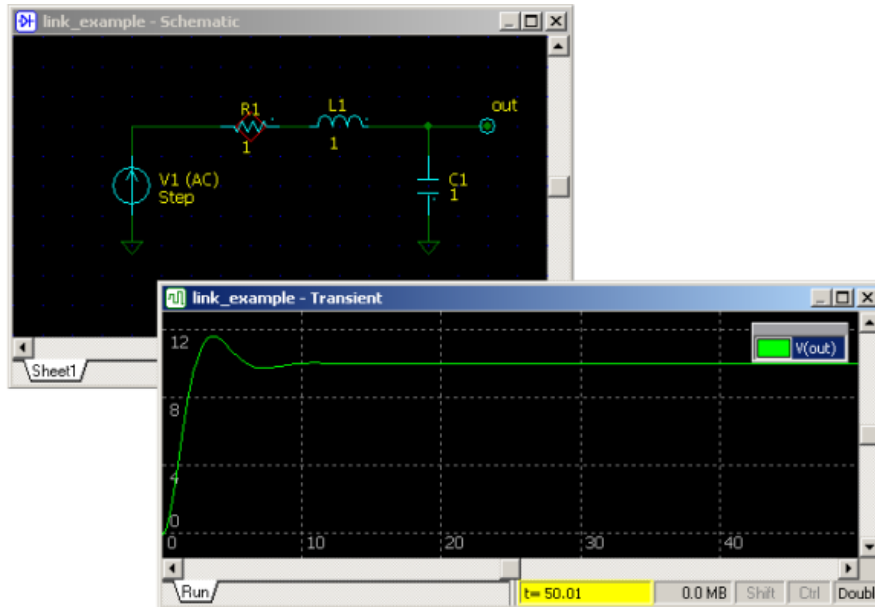


If you don't see that, please contact us and provide information about what exactly is shown, any error/warning messages, etc.

7. Make sure there are no active firewalls, proxy servers, etc., which may restrict communication with HTTP server.

## NL5-MATLAB link example

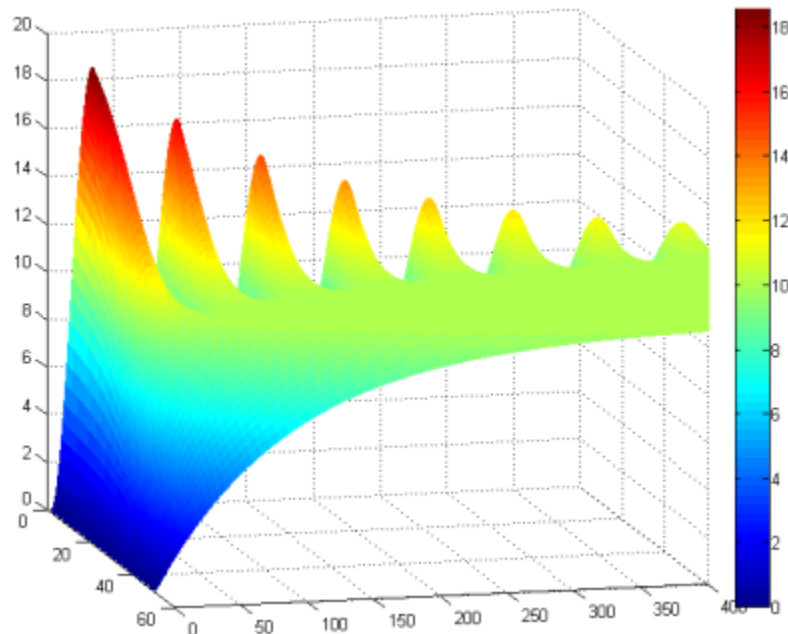
The following example shows using NL5 HTTP link with MATLAB. NL5 has HTTP server running, and schematic “link\_example.nl5” loaded. Transient window shows schematic response on V1 step function for R1=1:



The code listed below has been loaded into MATLAB from the file “link\_example.m” and executed:

```
clear
clc
close all
R=logspace(-1,1);
Header='http://127.0.0.1/?';
Cmd2=[Header,'tran'];
Cmd3=[Header,'ready'];
Cmd4=[Header,'V(out)%200,50,.1'];
for k=1:length(R)
    Cmd1=[Header,'R1=',num2str(R(k))];
    urlread(Cmd1);
    urlread(Cmd2);
    Response='0';
    while strcmp(Response,'0')
        Response=urlread(Cmd3);
    end
    Graph(k,:)=str2num(urlread(Cmd4));
end
Graph=Graph';
surf(Graph)
shading flat
colormap jet
colorbar
ylim([0 400])
```


The code changes R1 in the range 0.1...10 with logarithmic step, calculates transient, reads V(out) trace data, and displays V(out) at 3-D graph as a function of time and R1 value:



Example schematic and MATLAB files are located in the `Examples/MATLAB` folder of the NL5 download package.



## Scope

**Scope**  tool provides access to popular models of oscilloscopes (Keysight Technologies, Tektronix, LeCroy). You can configure channels and modes, obtain measured traces, and configure Waveform Generator (Arbitrary Function Generator).

To communicate with the oscilloscope, NL5 is using National Instruments **NI-VISA** drivers, which can be downloaded at National Instruments website ([www.ni.com](http://www.ni.com)). NL5 loads `visa64.dll` library when the Scope tool window opens first time. If library is not available, an error message will show up.

Select **Tools/Scope** or start NL5 in command line with `-scope` or `/scope` switch to open **Scope** window. Please note that the screenshots below are for illustration purposes only: a real screen view and controls may vary depending on a specific oscilloscope model.

**Toolbar** buttons perform the following operations:



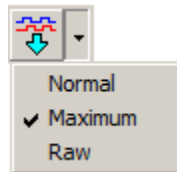
**Refresh resources.** Show list of instruments, select instrument.



**Update scope settings.** Read settings from the oscilloscope. Should be used synchronize oscilloscope and NL5 settings, in case if anything was changed through the scope interface.



**Read traces** from the oscilloscope. NL5 will stop data acquisition, and read traces currently displayed on the oscilloscope. For Keysight Technology oscilloscope, click arrow-down button on the right to select the data record to be transferred from the oscilloscope:



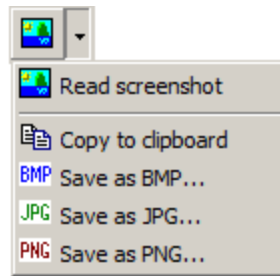
**Normal** - the measurement record is retrieved.

**Raw** - the raw acquisition record is used.

**Maximum** - whichever record contains the maximum number of points is used.

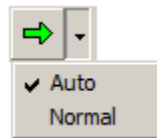


**Screenshot.** Download and display a picture from the oscilloscope screen. To copy the picture into the clipboard, or to save into the file in different formats, click arrow-down button on the right to open drop-down menu, and select an operation:



➡ **Run** the oscilloscope in **single** mode.

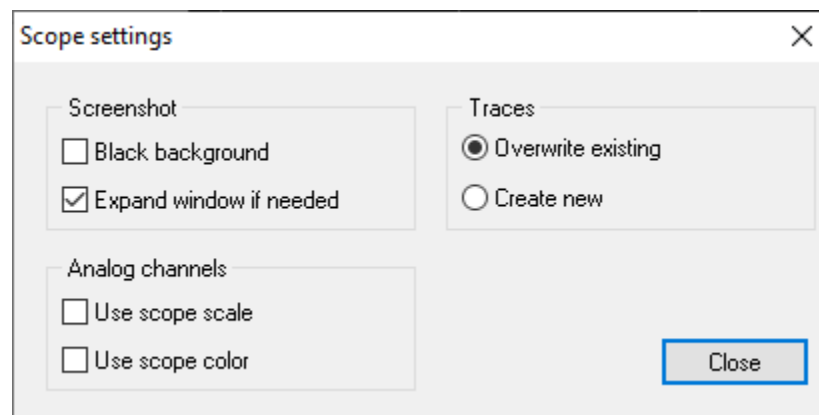
➡ **Run** the oscilloscope in **continuous** mode. For LeCroy oscilloscope, click arrow-down button on the right to select triggering mode:



✖ **Stop** the oscilloscope.

⬆ **Send** configdata data to the waveform generator.

⚙ **Settings.** Open **Scope settings** window:



### Screenshot.

**Black background.** If selected, read screen picture with black background.

**Expand window if needed.** When screen picture is bigger than current size of the Scope window, expand the window so that the picture fits in properly.

### Analog channels.

**Use scope scale.** When reading traces from the oscilloscope, set scale and offset of the trace in NL5 to a current scale of the trace on the oscilloscope (not used for LeCroy oscilloscopes).

**Use scope color.** When reading traces from the oscilloscope, set a color of the trace to a trace color on the oscilloscope.

### Traces.

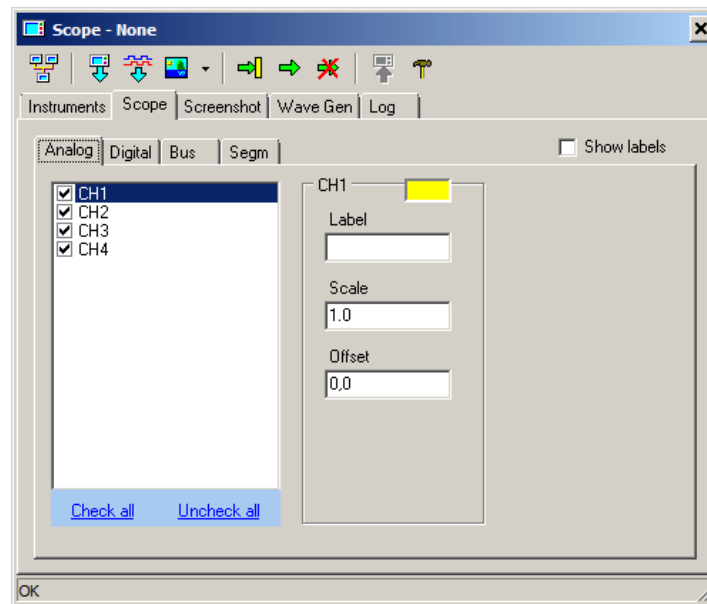
**Override existing.** When reading traces from the oscilloscope, if a trace with the same name already exists, delete existing data download new data into existing trace.


**Create new.** When reading traces from the oscilloscope, if a trace with the same name already exists, create a new trace.

**Keep scope window on top.** If selected, keep Scope window on top of other NL5 windows.

## Working with oscilloscope

**Configuring oscilloscope.** When instrument is selected, NL5 reads some configuration information from the oscilloscope, and shows it on the **Scope** tab:

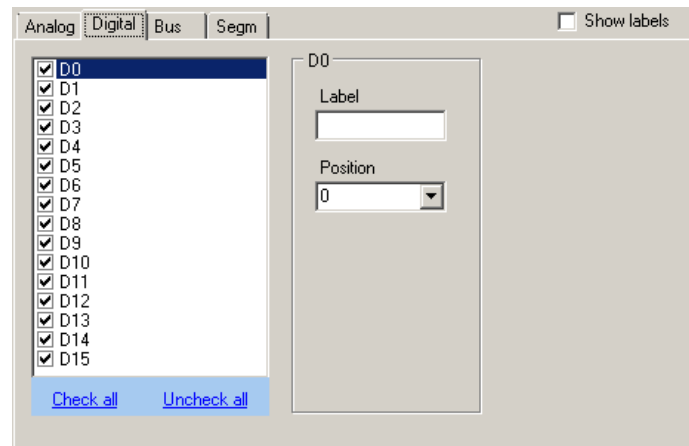


Depending on the oscilloscope model, there could be up to four tabs, describing analog, digital, bus, and segmented memory settings. If any of displayed settings were changed directly through the oscilloscope controls, or any other oscilloscope interface, click **Update scope settings**  to retrieve new settings, and update information in the Scope tool.

Check/uncheck **Show labels** checkbox to show/hide channel labels on the oscilloscope screen (Keysight Technologies only).

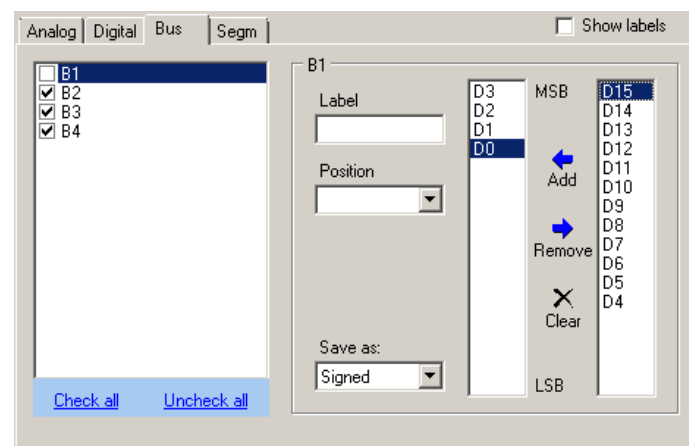
**Analog** tab displays analog channel settings. List of channels shows all available analog channels of the oscilloscope. State of the checkbox indicates whether the channel is visible on the oscilloscope. Color box indicates trace color on the oscilloscope screen. Enter new **Label**, **Scale**, or **Offset** and press **Enter** to submit (not available for LeCroy oscilloscopes).

**Digital** tab displays digital channel settings:






List of channels shows all available digital channels of the oscilloscope. State of the checkbox indicates whether the channel is visible on the oscilloscope. Enter a new **Label** or **Position** value and press **Enter** to submit.

**Bus** tab displays bus channel settings:

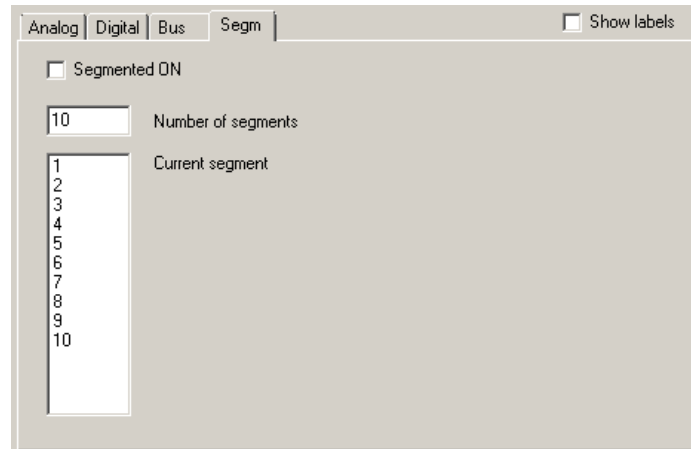


**Bus** configuration tab may vary depending on a specific oscilloscope model.

List of channels shows all available bus channels of the oscilloscope. State of the checkbox indicates whether the channel is visible on the oscilloscope. Enter a new **Label** or **Position** value and press **Enter** to submit. Select **Signed** or **Unsigned** mode from **Save As** drop-down list to specify how the bus value is converted into NL5 number representation. This will affect how the bus numbers will be displayed on the NL5 graph.


**To configure bus bits**, click **Add** , **Remove** , or **Clear**  to fill in the left box with required bus bits. Right box consists of bits currently available for the bus. **MSB** and **LSB** labels indicate most- and least-significant bit of the bus.

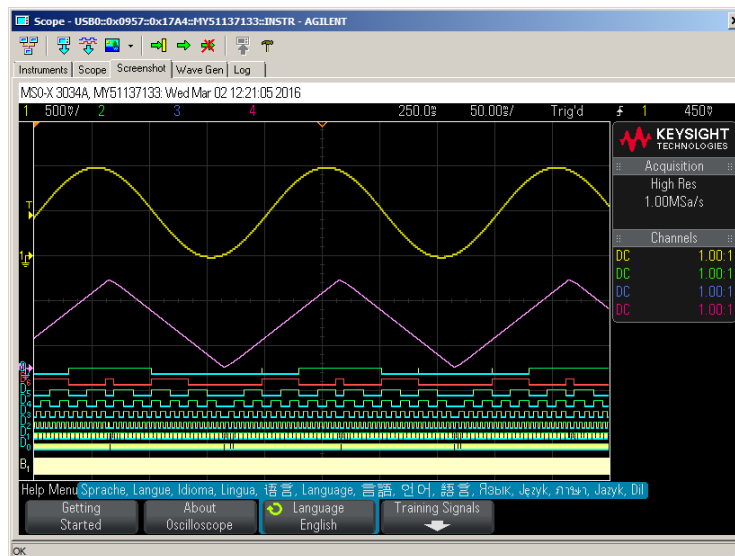
**Segm (or Fast frame)** tab displays segmented memory operation settings:



Segmented memory configuration tab may vary depending on a specific oscilloscope model.

Select **Segmented ON** check box to enable segmented memory operation. Enter **Number of segments**, press **Enter** to submit. Select **Current segment** in the list, to display it on the oscilloscope.

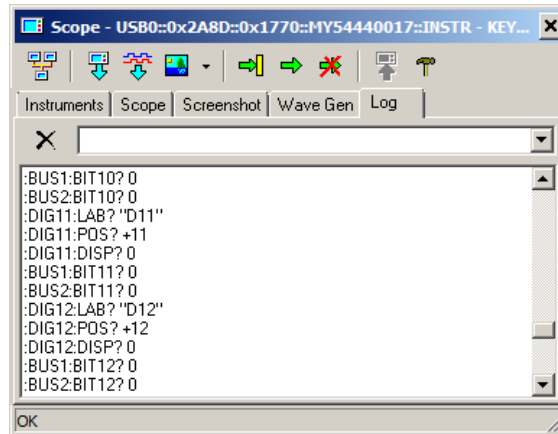
**Screenshot** tab displays an oscilloscope screenshot. Click **Read screenshot**  to download a screenshot and display it on the tab:



Depending on **Scope settings** the screenshot may be obtained with black or white background. Scope window size may also be expanded automatically to fit the image.


Please note that Tektronix 3000/4000 series requires a flash drive to be present at the E: drive USB port of the oscilloscope (usually located at front panel). It is used for storing an intermediate image file.

**Log** tab shows commands executed in the last operation:



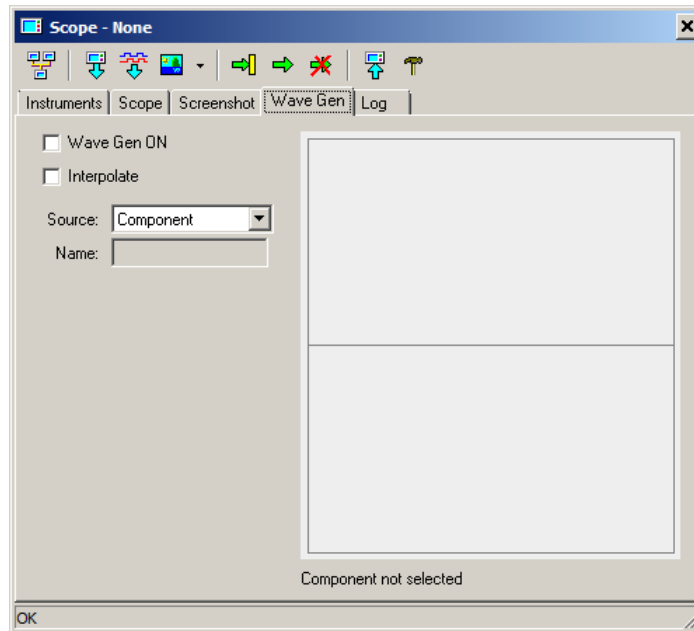
It also has a command line option. Enter a command or select previously issued commands from the drop-down list, then press **Enter** to execute:



Click **Clear log**  to clear the log window.

## Working with waveform generator

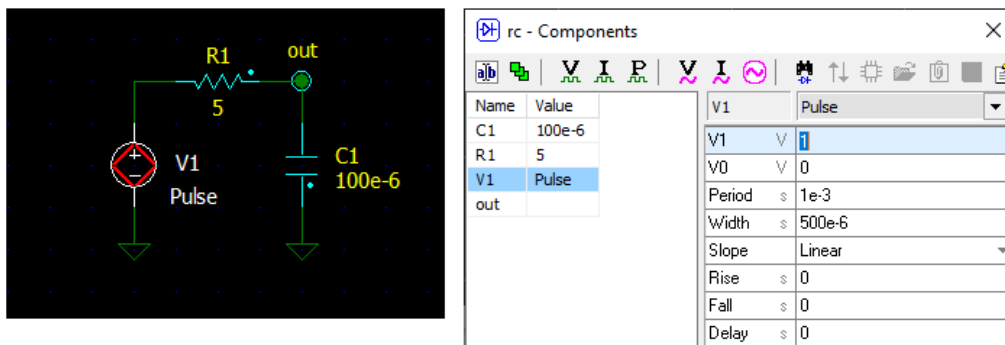
**Wave Gen** tab is used to create a signal for Waveform Generator (Arbitrary Function Generator), and send it to the oscilloscope:



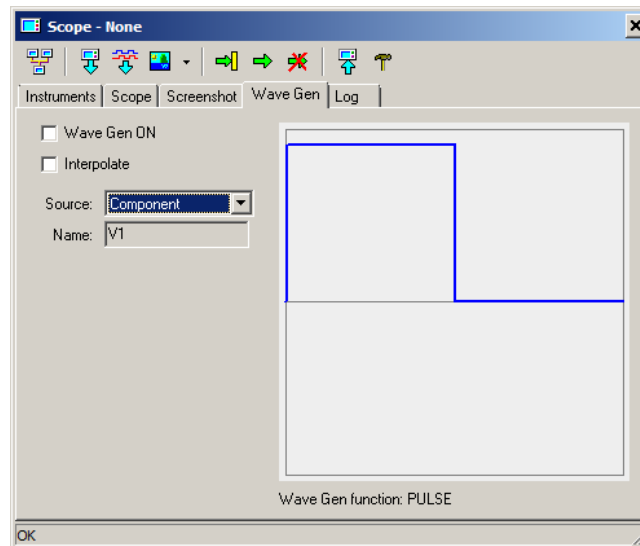
Select **Wave Gen ON** check box to enable waveform generator. Select **Interpolate** checkbox to enable signal interpolation (Keysight Technologies only).

Select **Source** to use signal source. There are four source options available for Wave Generator.

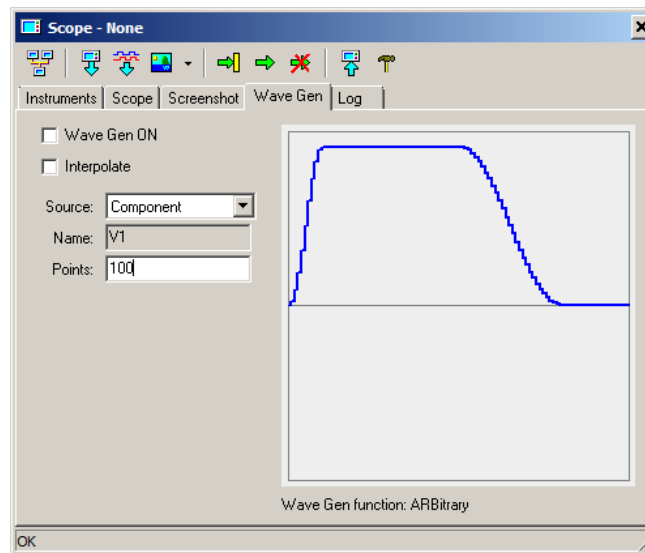
**Component.** Use **Voltage source** or **Label NL5** component settings to specify a waveform. The following models of the component are supported: **V, Pulse, Sin, PWL, Function, File, Trace**. Select the component in the schematic:



A preview of the waveform will be shown in the Scope window:



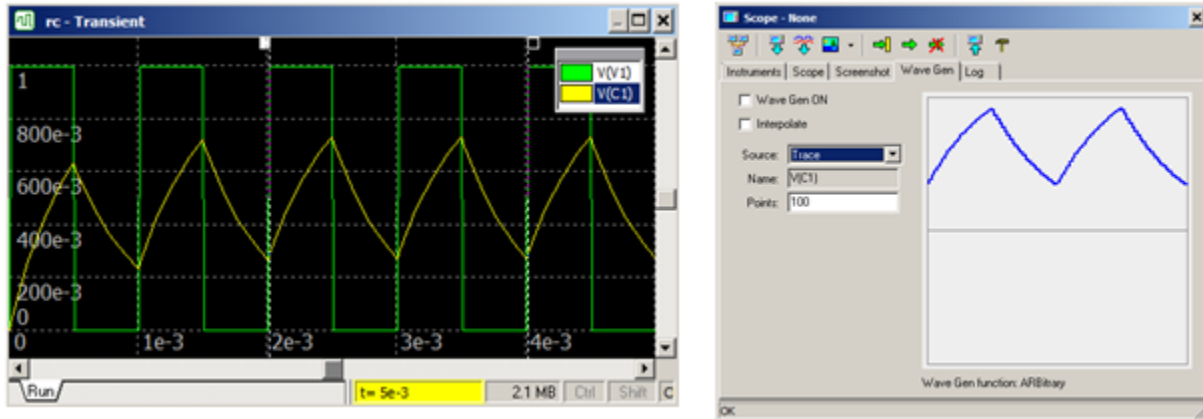
If component model and parameters allow, a built-in wave form generator function will be used (**DC**, **SIN**, **PULSE**). Otherwise, and arbitrary generator waveform **ARB** will be used. For example, the following signal will require an arbitrary function:



For arbitrary function, number of **Points** should be specified: enter the number and press **Enter**.

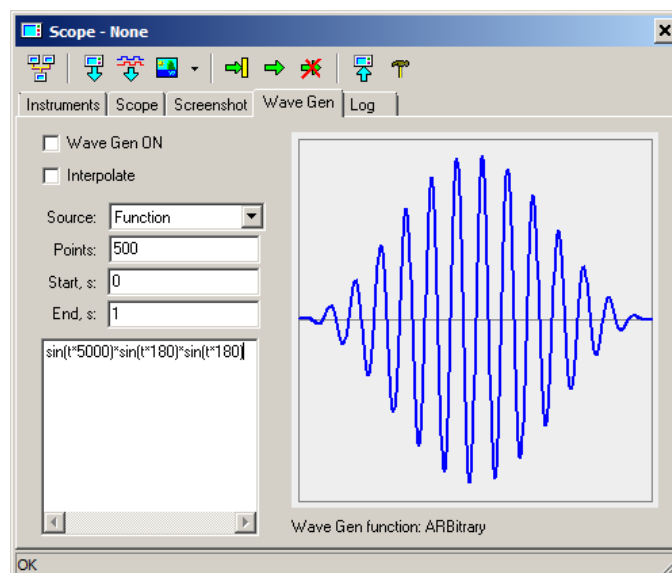


**Trace.** Select a trace in the NL5 transient window, and use cursors to specify a trace interval which will be used for the waveform. A preview of the waveform will be shown in the Scope window:

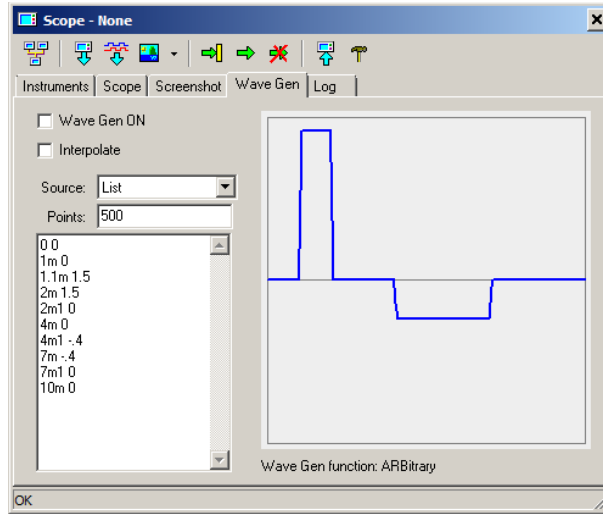


Specify number of **Points** and press **Enter**.

**Function.** Enter an arbitrary function for a waveform, use **t** parameter as a time. Also, enter number of **Points**, **Start**, and **End** time for the function:



**List.** Enter waveform points as a list of numbers in “time,amplitude” format:

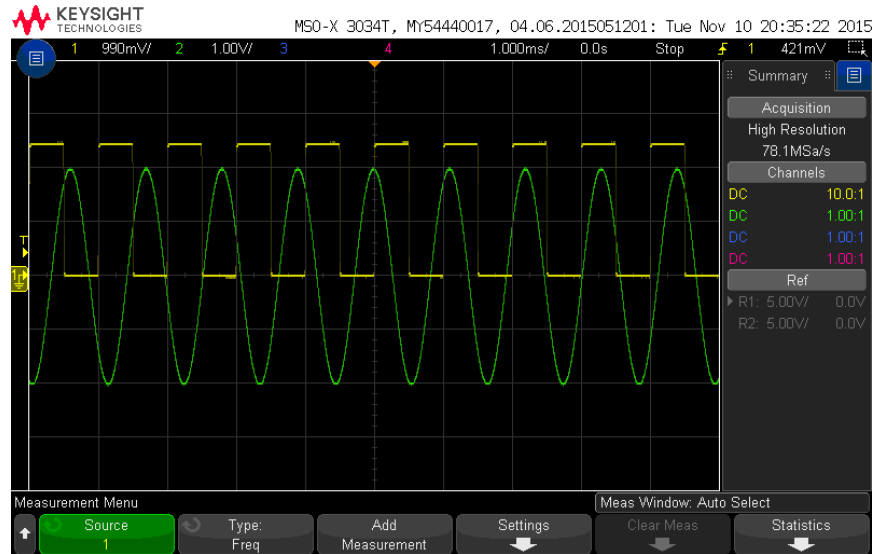


Specify number of **Points** and press **Enter**. Please note that number of points is not number of points in the list: it is number of samples to generate waveform signal.

When waveform signal is ready, click **Send to scope**  to send waveform points to the Waveform Generator.

## Working with segmented memory

Here are two signals (pulse and sine) with small frequency difference:



Turn on segmented memory, enter number of segments, and press **Enter**:

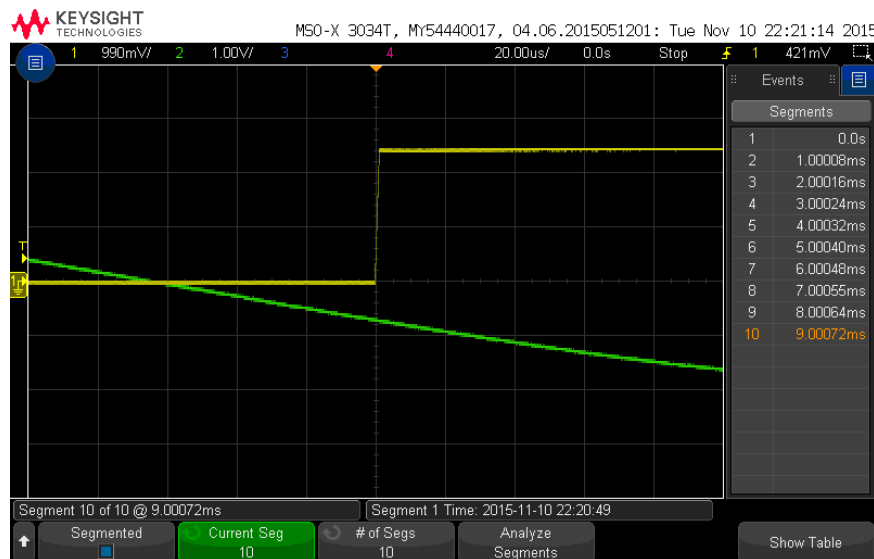
Analog | Digital | Bus | **Segm**


Segmented ON

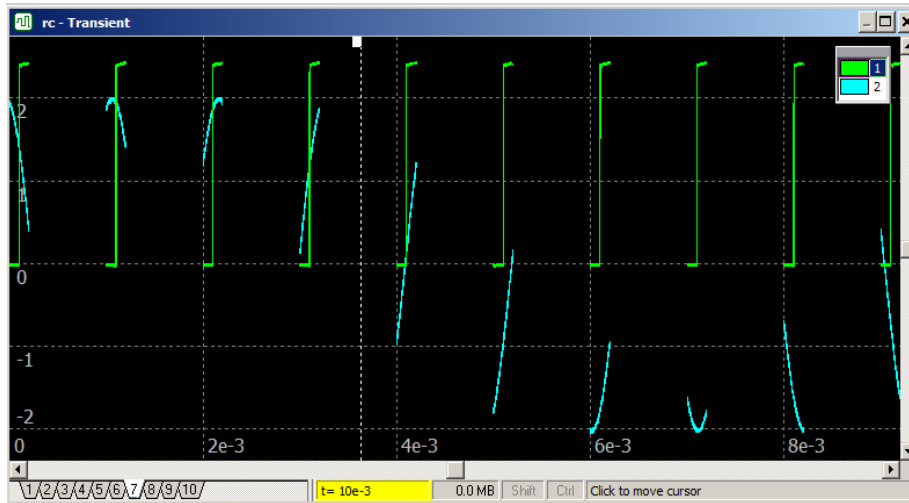
Number of segments

Current segment


**Run** the oscilloscope (in single or continuous mode):

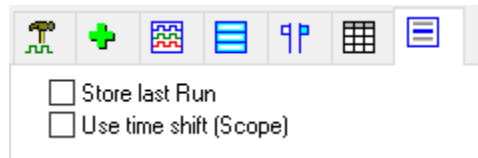



Read traces  from the oscilloscope:

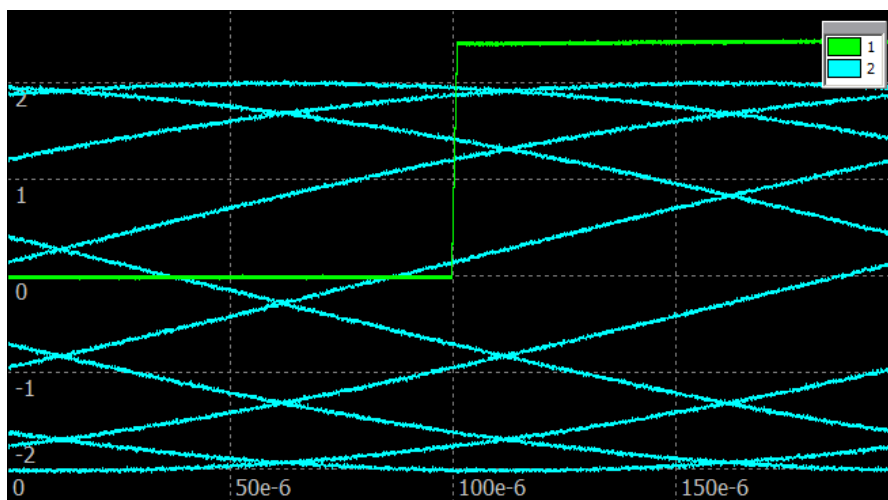


On this graph, segmented memory pieces are displayed according to their absolute time.

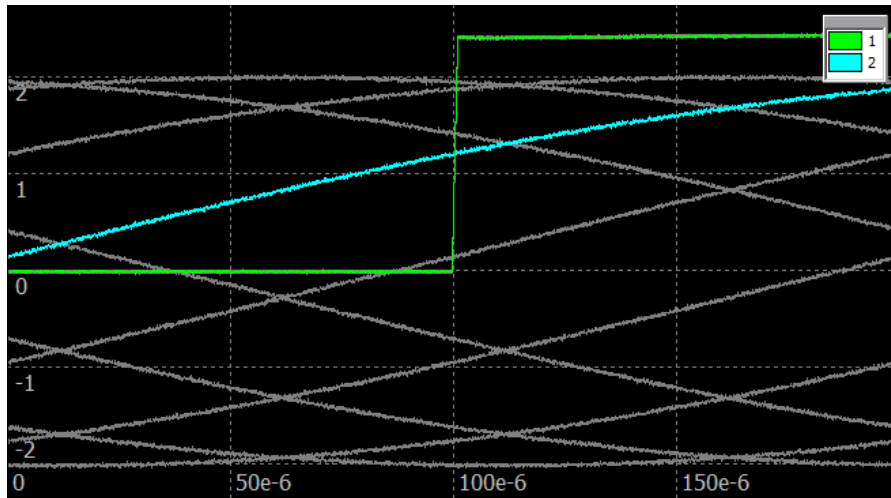
To display segments in “overlaid” mode (relative to the time when segment was triggered), go to **Transient settings**  window, **Storage** tab, unselect “Use time shift (Scope)” option:



Here is an “overlaid” view of the segmented memory data (you may need to auto-scale the graph horizontally by clicking **Fit the screen** ):



To see individual segments, select **Selected only** or **Selected and dimmed** storage display mode.



**Look** through the segments on the graph by **left-clicking** on the storage list, and then pressing **Up/Down** keyboard keys:

