Qucs-S: Getting started analog simulation

Vadim Kuznetsov

October 11, 2023

1 Introduction

Qucs-S was forked from the Qucs cross-platform circuit simulator in 2017. "S" letter indicates SPICE. The purpose of the Qucs-S subproject is to use free SPICE circuit simulation kernels with the Qucs GUI. It merges the power of SPICE and the simplicity of the Qucs GUI. Qucs intentionally uses its own SPICE incompatible simulation kernel Qucsator. It has advanced RF and AC domain simulation features, but most of the existing industrial SPICE models are incompatible with it. Qucs-S is not a simulator by itself, but requires to use a simulation backend with it. The schematic document format of Qucs and Qucs-S are fully compatible. Qucs-S allows to use the following simulation kernels with it:

- Ngspice is recommended to use. Ngspice is powerful mixed-level/mixed-signal circuit simulator. The most of industrial SPICE models are compatible with Ngspice. It has an excellent performance for time-domain simulation of switching circuits and powerful postprocessor.

- XYCE is a new SPICE-compatible circuit simulator written by Sandia from the scratch.

- SpiceOpus is developed by the Faculty of Electrical Engineering of the Ljubljana University. It based on the SPICE-3f5 code.

- Qucsator as backward compatible and for RF simulation with microwave devices and microstrip lines. Not recommended for general purpose circuits.

Qucs-S is a cross-platform software and supports a number of Linux distributions alongside with Windows™. The Linux packages are generated automatically with the Open Build Service (OBS) system. Check the official website to get the list of supported distributions. Please keep in mind that the installation packages doesn't provide the simulation kernel. It need to be installed separately. The Ngspice is recommended. For Debian and Ubuntu it is installed automatically as the dependency. Refer to Ngspice website for installation instructions for other platforms.

2 Setup on the first start

Once the Qucs-S is installed, it asks to select the simulation kernel. Since the version 2.0.0 the application tries to find Ngspice in some usual system locations on the first start. If the Ngspice executable was found automatically the information message will be shown. Otherwise the user will be prompted to configure the simulators manually. The following window (Fig. 1) is opened after the first start of the application. Select the default simulator using the
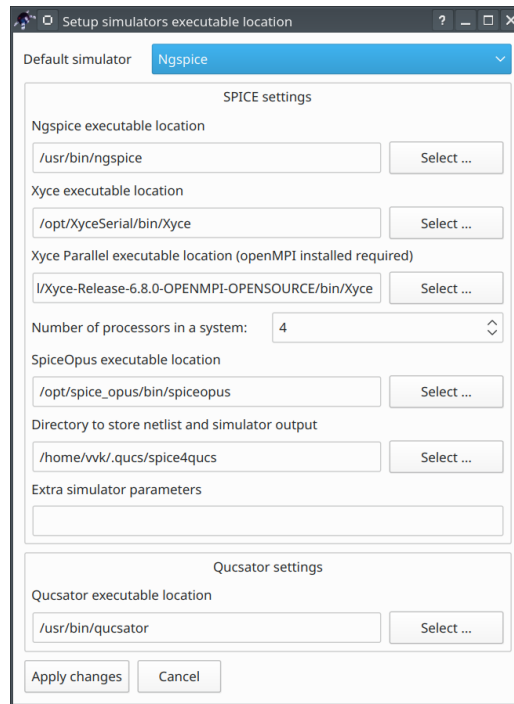
Figure 1: Simulator setup dialog

drop-down list and correct the simulator executable paths if necessary. These settings could be corrected later using the *Simulation->Select default simulator* menu.

Quick switching of the simulation kernel without application restart is supported since version 2.0.0. See the section 9 for more details.

3 Main window structure

The Qucs-S main window is shown in the Fig. 2. The main window consists of two areas: schematic editor (4) on the right side and main dock (1) on the left side. Several schematics could be opened simultaneously. It's possible to switch between the opened schematics using the tabular bar above the schematic editor area.

The dock on the left side has four tabs (3): *Projects*, *Content*, *Components*, and *Libraries*. The *Projects* tab is opened after the application start. It is empty, because there is no projects after the fresh installation. The *Components* tab (see Fig. 2) contains a list of primitive devices available in Qucs. The components are divided into several categories (lumped devices, nonlinear devices, simulation, etc.). The categories could be selected from the drop-down list (2). The status bar is located on the bottom side of the main window. The active simulation kernel is displayed on the status bar (for example Ngspice on Fig.2).

The toolbar area is located on the top side of the main window below the main menu. Some important buttons are located on the toolbar. They are: *Wire* (6), *Node name* (7), *Ground* (8), and *Marker* (9).

Everything in Qucs-S is considered to be a component. The simulations, equations, and SPICE directives are also special components that could be found in dedicated categories in the component pallet. The diagrams are also special components and could be found in the *diagrams* category in the drop-down list (2).

To place the component on schematic choose desired category from the drop-down list (2) (for example "lumped components") and click on the symbol (for example "Resistor"). Moving
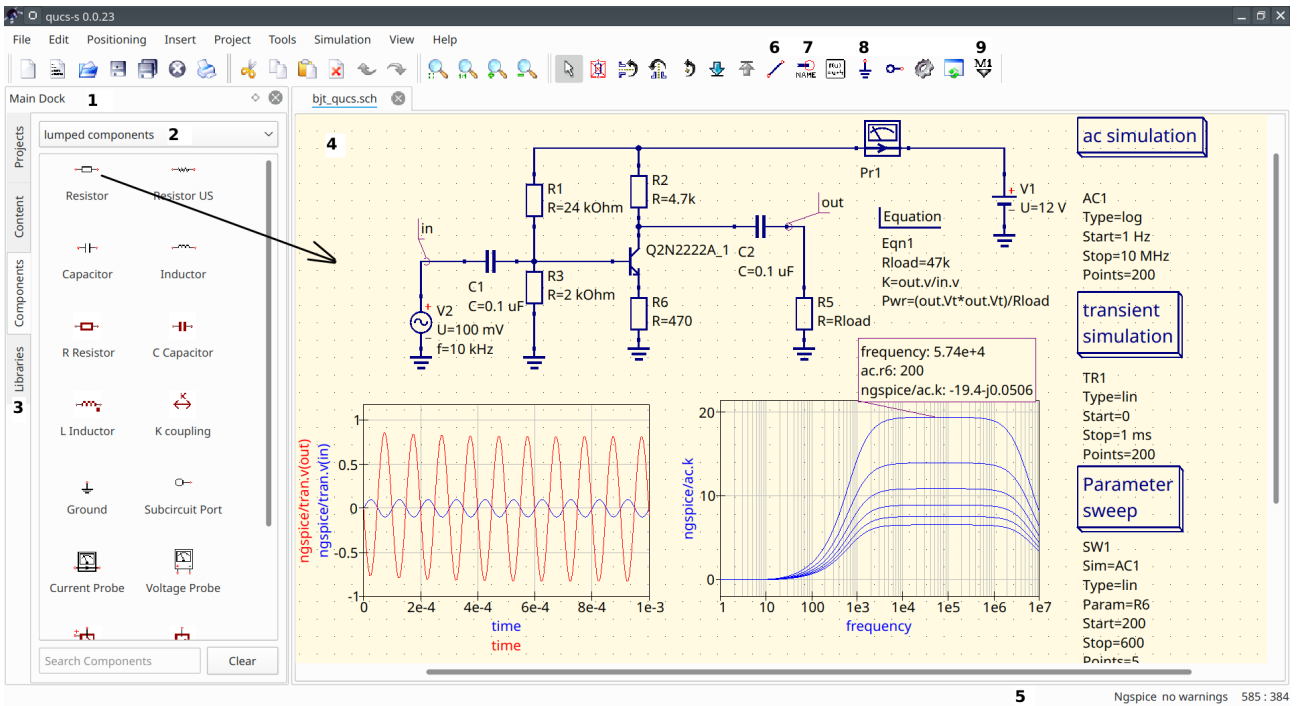
Figure 2: Qucs-S main window. See text for explanations

the mouse cursor into the working area (4) you are carrying a drawing of a resistor symbol. Pressing the right mouse button rotates the symbol, pressing the left mouse button places the component onto the schematic.

The components properties could be edited using the properties dialog (Fig. 3) that is called from the context menu of the component or after the left mouse button double click.
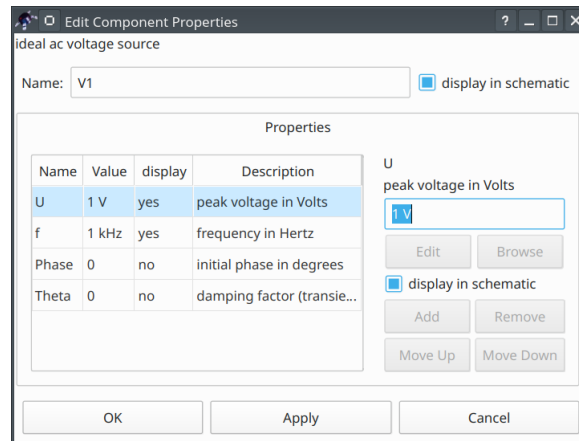

Figure 3: Component properties dialog

4   Step by step guide how to setup an analog simulation

4.1   AC and transient simulation

Let's simulated an RC-circuit using Qucs-S with Ngspice backend. The schematic is shown in the Fig. 4. Let's perform the AC and transient simulations and plot the magnitude response

of the circuit and the waveforms on the input and output nodes. Perform the following steps to draw the schematic and setup the simulation.
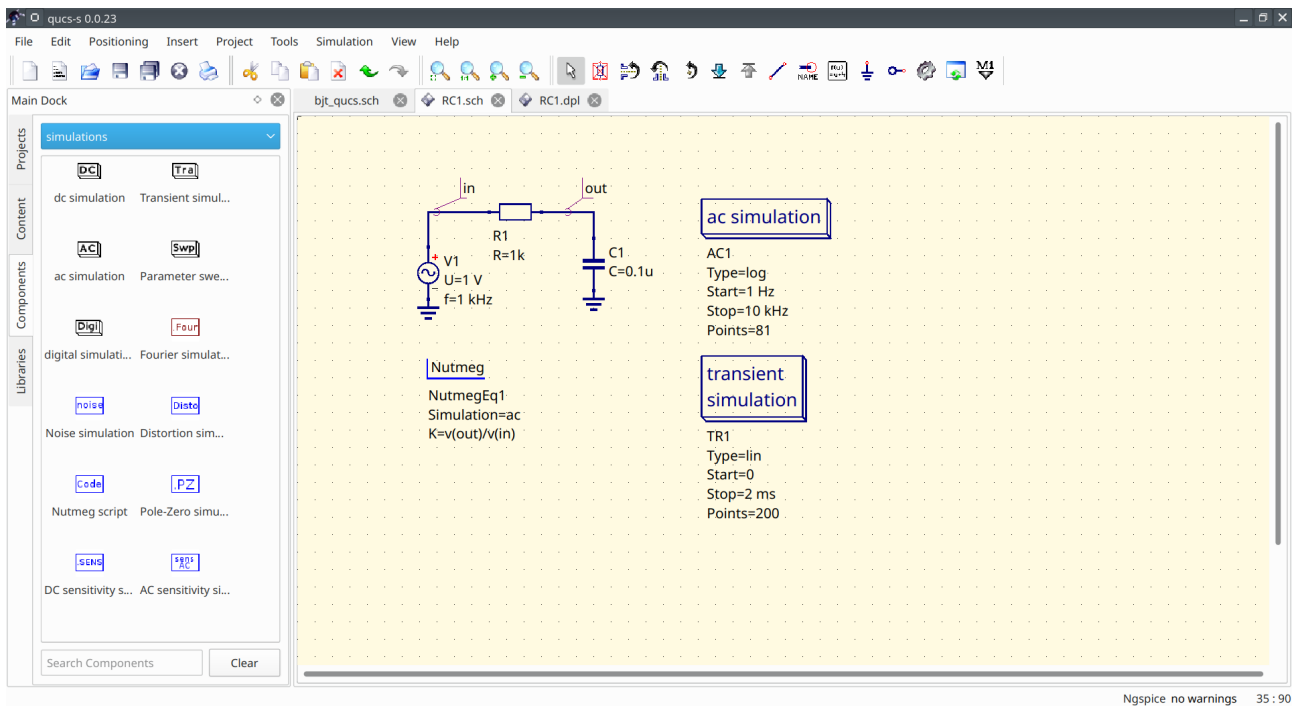

Figure 4: RC-circuit in Qucs-S

- **Step 1:** place components on the schematic and set their properties as show in the Fig. 4. We need resitor, capacitor (both could be found in the *lumped devices* category) and AC voltage source (from *sources* category). The ground and wires could be taken from the application tool bar (see Fig. 2) using the buttons (6) and (8).

- **Step 2:** place simulations on the schematic. We need AC simulation and Transient simulation. Both could be taken from the *simulations* category (see Fig. 4). The procedure is same as for usual components. Setup the properties of the simulations as shown in the Fig. 4. A special dialog window opens upon the double click on the simulation device (Fig. 5). For AC simulation enter the start (1 Hz) and stop (10 kHz) frequencies and select the logarithmic sweep type. For transient simulation enter the stop time (2 ms) and points number (200 points).
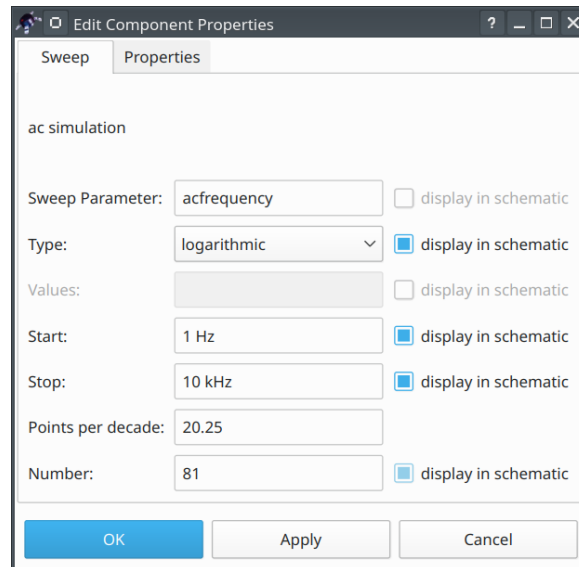
4

Figure 5: Simulation properties dialog

- **Step 3:** place the node labels `in` and `out` using the button (6) on the toolbar (see Fig. 2). The node name could be set in a special dialog window (Fig. 6).
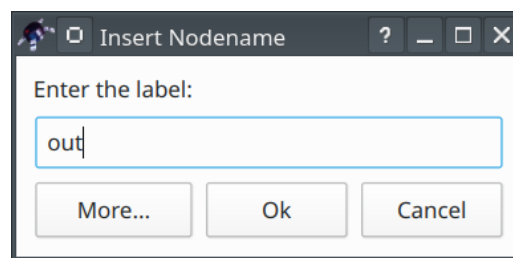

Figure 6: Node label setup dialog

- **Step 4:** Place equation for the magnitude response $K = V_{out}/V_{in}$ on the schematic. Equation is the special component called *Nutmeg equation* that could be found in the *spice specific sections* category. The full SPICE syntax for mathematical functions and node names is supported. Refer to Ngspice manual for more details. The simulation should be specified for each Nutmeg equation. We will calculate the magnitude response for the AC simulation only and therefore we need to enter `ac` as the first property for this device as shown on the Fig. 7.

- **Step 5:** Save the schematic document as the file with the `*.sch` extension using the *File -> Save as* menu or press the `Ctrl-S` shortcut. The schematic is ready for the simulation now.

- **Step 6:** simulate the schematic. Press the *Simulation -> Simulate* or press the F2 hotkey on the keyboard. The dialog window containing the simulator log messages will appear shortly (Fig. 8). This dialog window contains the simulation console where the messages from the Ngspice are shown.

  It should report that the simulation finished without errors. If you see error messages check your schematic and application settings. After the Ngspice reports about the fin-
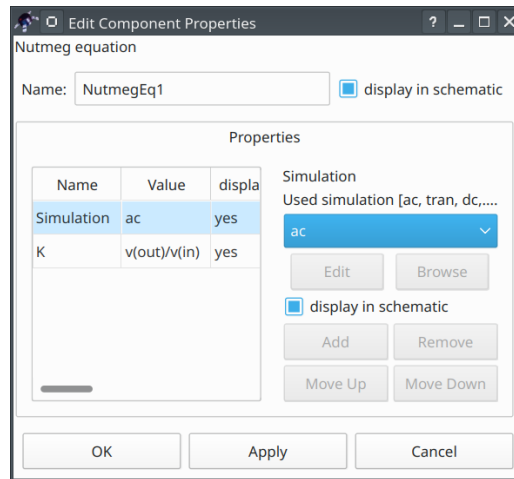
5

Figure 7: Equation setup dialog

ished simulation, close this window clicking the *Exit* button. Now the diagrams could be added on schematic field. It is also possible to save the netlist file from this dialog .
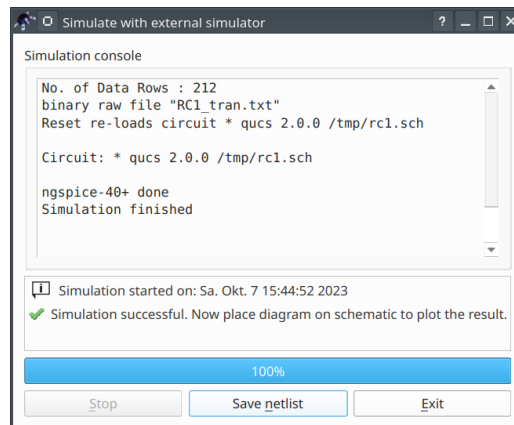

Figure 8: Simulation dialog

- **Step 7:** place diagrams on the schematic (Fig. 9). We need two cartesian diagrams for magnitude response and transient waveforms. The diagrams are also special components that could be found in the *diagrams* category. The traces could be selected in the special diagram properties dialog (Fig. 10). Select the `ac.k` traces for one diagram and `tran.v(in)` and `tran.v(out)` traces for another diagram. It is also possible to setup the axis limits and type and the color scheme for the diagram. Select logarithmic X-axis for the magnitude response graph using the settings in the *Properties* tab. The markers could be placed on graphs using the button (9) from the toolbar (Fig. 2). The position of the marker could be adjiusted using the arrow keys on the keyboard.

4.2 DC analysis

Qucs-S unlike Qucs has no special DC simulation mode. If only *DC simulation* component is placed on the schematic no simulation will be launched and error message will be shown. This component is kept for backward compatibility with old Qucs schematics.

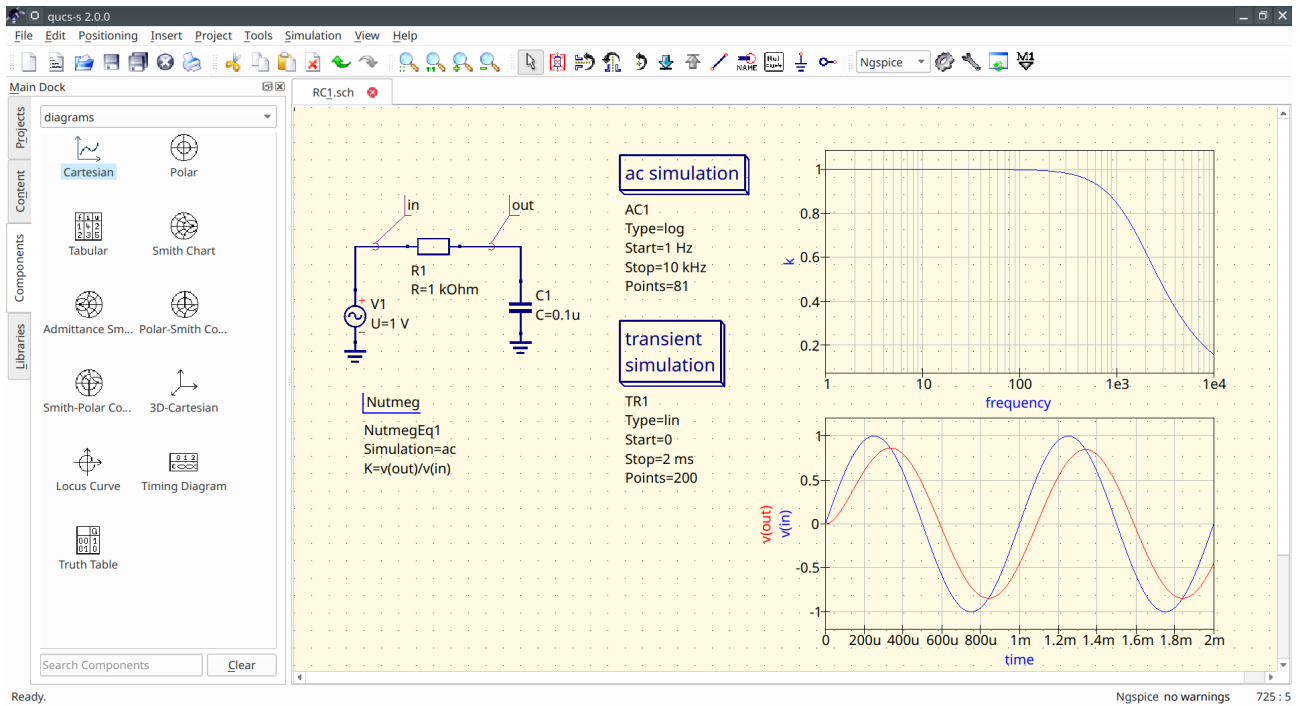But there exists two modes of the DC simulation:
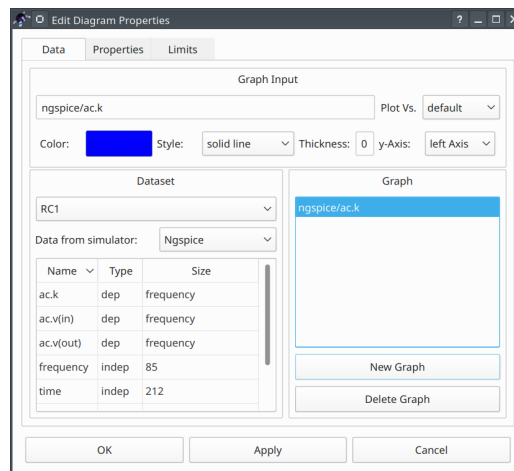
Figure 9: Schematic with two diagrams added

Figure 10: Diagram properties dialog

- Show operating point directly on the circuit. This mode is activted when pressing F8 keyboard shortcut or using *Simulation-¿Show bias* menu entry. See the Fig. 11 for example of such simulation.

- DC sweep mode. You need to insert *Parameter sweep* simulation component and attach it to the DC simulation. This simulation mode could be used for plotting IV-curves or getting table output. See the Fig. 12 for example of this simulation.
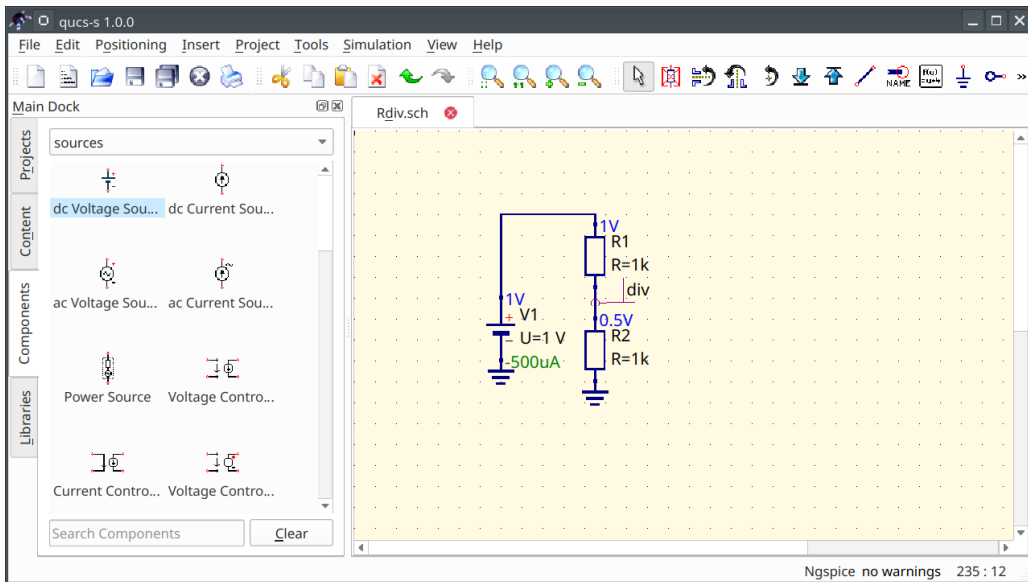
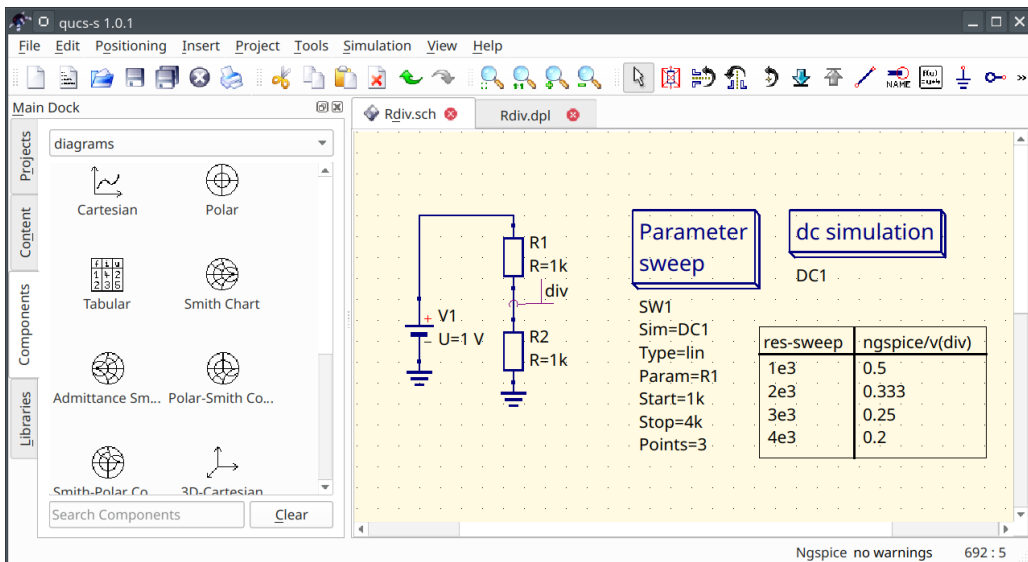Figure 11: Operating point simulation of the voltage divider

Figure 12: Using DC sweep to get table output of the resistive divider voltage

5 Parameter sweep

5.1 Device sweep

Parameter sweep is defined as the special simulation type that is attached to any other simulation. Let's consider an example and attach parameter sweep to the circuit shown in the Figure 4. We will see the influence of the resistance of the RC-circuit on the cutoff frequency.

Important note: unlike Qucs before version 0.0.20, Qucs-S doesn't support the sweep of variables directly. Legacy Qucs schematics using variable sweep will not work. Parameter sweep supports only device sweep. Resistor, capacitor or source are accepted.

Perform the following steps to add parameter sweep:

- **Step 1:** Place parameter sweep component on the schematic sheet (Figure 13). The *Parameter sweep* device could be found in the *simulations* group.
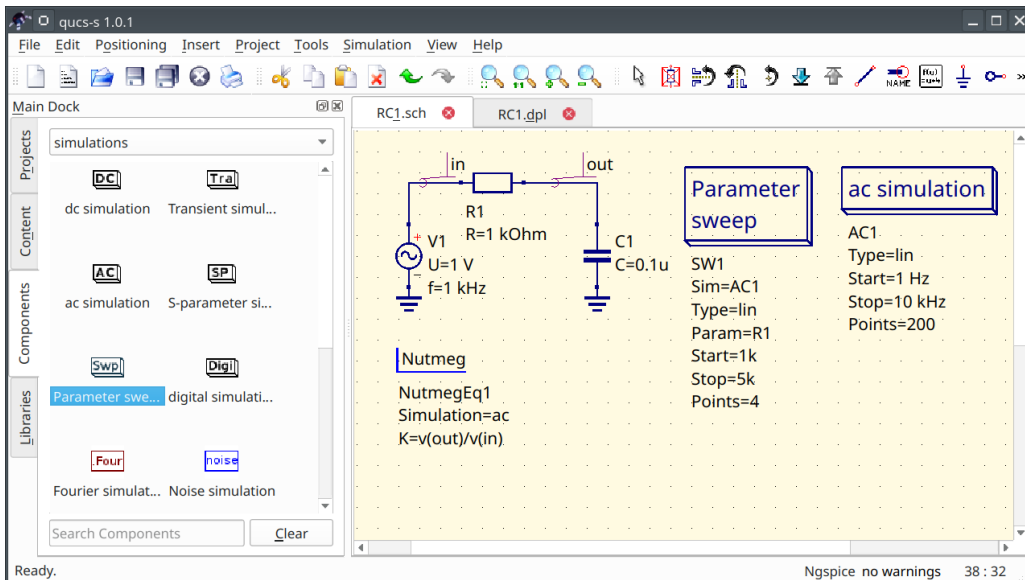
Figure 13: Adding parameter sweep on the schematic

- **Step 2:** Set the properties of the parameter sweep simulation. After the double click the dialog window will be opened (Figure 14). The *Simulation* drop-down list is the the name of simulation component to which the parameter sweep is attached. For our example we need to select the `AC1` simulation from the drop-down list, because we attaching the parameter sweep to the AC simulation. The *Sweep parameter* input field is the **device** name. Variables or other objects name are not acceptable here! Enter the `R1` device name here. The *Start*, *Stop*, *Step*, and *Number* input fields serve for the definition of the sweep range. We will sweep the `R1` value from 1 kOhm to 5 kOhm and take 4 sweep points.
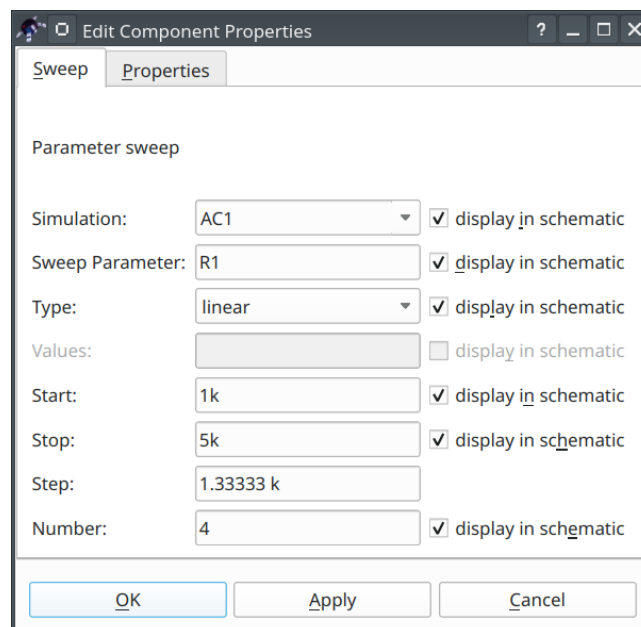

Figure 14: Setting parameter sweep properties

- **Step 3:** Run the simulation and place the diagrams as for usual simulation (Figure 15). You should see the family of the curves. You may place a marker on the diagram and

9

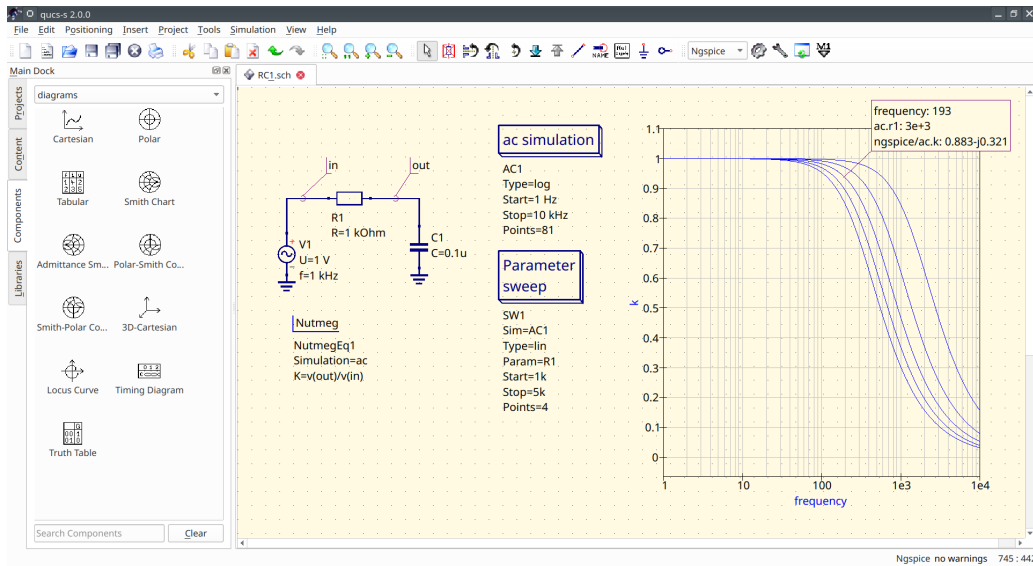move it between the sweep curves with *Arrow Up* and *Arrow Down* keys.

Figure 15: Magnitude response of the RC-circuit with the parameter sweep

5.2 DC sweep

DC sweep simulation takes place when parameter sweep is attached to the DC simulation. Such simulation mode is explained in the Section 4.2. DC sweep only accepts device name (resistor or source) as the sweep parameter name. The DC sweep with variable name will not work. This simulation mode usually serves for obtaining the IV-curves.

5.3 Variable sweep

The recent version of the Ngspice support sweeping of the varibles. The variable should be defined using the `.PARAM` statement (available at the *SPICE specific sections* group). Qucs-S doesn't create the variable automatically. Refer to the schematic shown in the Figure 16 for the example of such simulation mode.

6 Digital simulation

6.1 Digital devices with analog model

Digital devices including logic gates, flip-flops, and more complex devices could be found in the group *Digital components*. Some of these devices have both analog and digital model. Ngspice simulation operates using XSPICE models. Here is the list of these devices:

- Logic gates, inverter and buffer

- D-, JK-, T-flip-flops

- Digital source

The usage of these devices in analog simulation mode has no special requirements. The example shown in the Figure 17 illustrates the transient simulation of the RC oscillator circuit containing two inverter logic gates. The power supply voltage for digital devices should be set using the `VCC` parameter with `.PARAM` section.
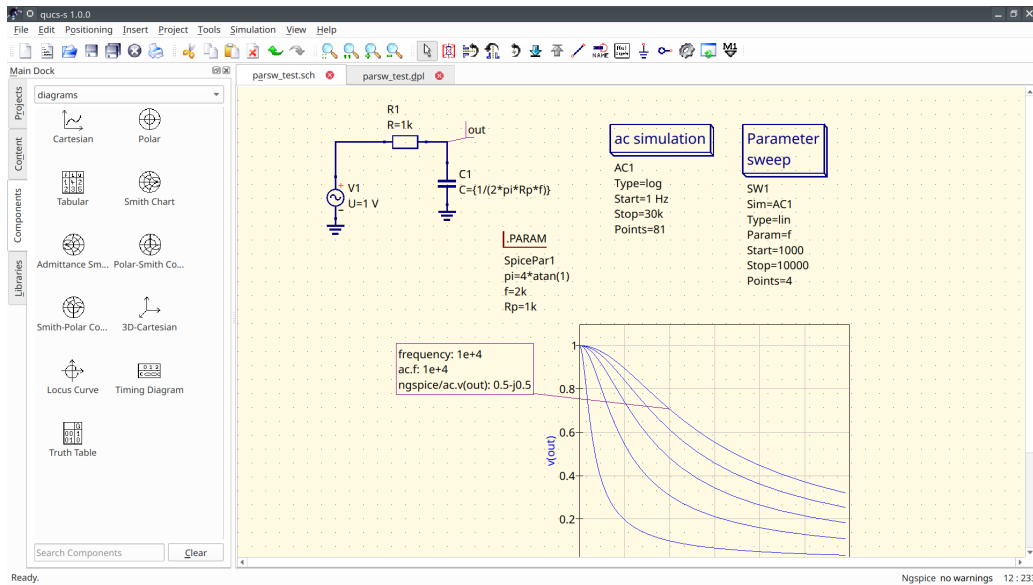
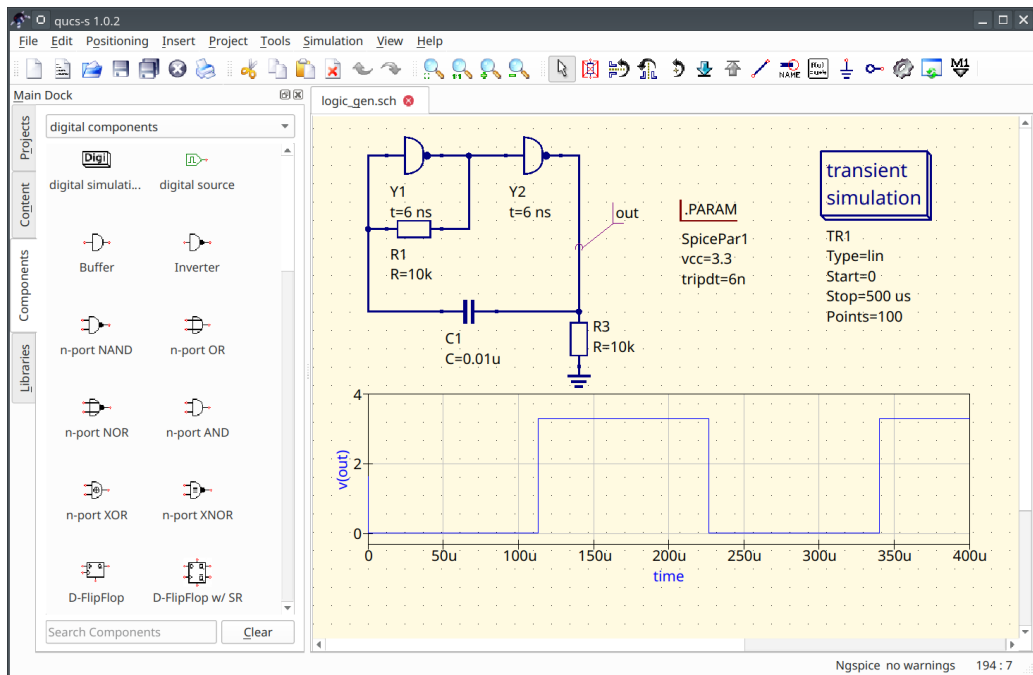Figure 16: Magnitude response of the RC-circuit with the parameter sweep

Figure 17: Oscillator example using logic gates

6.2 Digital simulation with Verilog/VHDL

The most of digital components except of the listed in the previous section doesn't contain analog SPICE model. These devices are supported by digital simulation mode only. **Mixing digital and analog devices (like RCL) in one circuit is not supported!** You need to install one the following free backends to perform digital simulation

- Icarus Verilog `iverilog https://steveicarus.github.io/iverilog/`

- FreeHDL `http://freehdl.seul.org/`

Place on the schematic field a special device named *Digital simulation* to switch Qucs-S

11

to digital mode. Verilog or VHDL backend will be started automatically instead of SPICE if digital simulation is found in the schematic. User doesn't need to switch Qucs-S to the digital mode in the application settings.

There are two modes of the digital simulation: truth table and timing diagram. An example of the digital simulation of the 2-bit decoder/demultiplexer is shown in the Figure 18. This simulation uses Verilog and timing diagram mode.
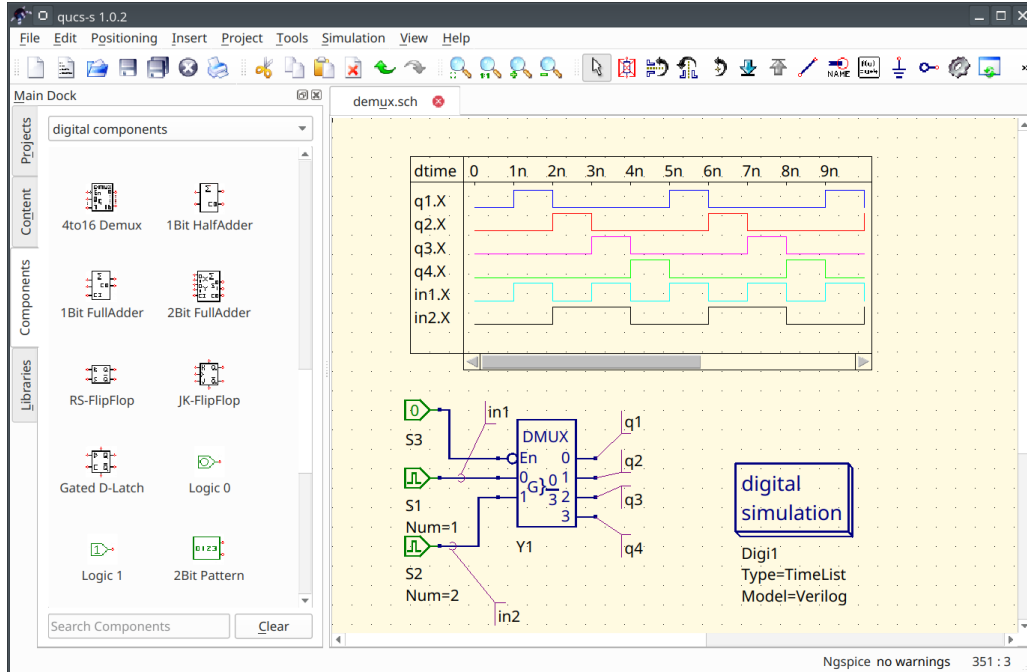

Figure 18: Simulation of the digital circuit with iverilog

7 Tuning mode

Since the version v2.1.0 Qucs-S support the tuning simulation mode. The tuner allows to adjust the component property values using an interactive sliders and immediately see the result on diagrams. Tuning mode is supported for all simulation kernels including Ngspice and Qucsator. Tuner may act as an interactive alternative of the parameter sweep simulation.

Let's consider how to use the tuning mode for the RC-circuit Fig. 4. The schematic is required to have the simulations and diagrams placed on schematic. The tuner will not work without defined simulations and diagrams. The tuner is activated using the *Simulation-¿Tune* menu or F3 keyboard shortcut. The following dialog window (Fig. 19) is shown after the activation. Then user need to click on the device or the device property with the left mouse button. A new slider will be added to the tuner window. It is possible to manually set the upper and lower limit of the tuned device property. The default limits are within the $\pm 15\%$ range from the starting value.

After the sliders are added to the tuner window it is possible to drag the sliders and observe the changes on the diagrams. For example we can see the cutoff frequency of the RC-circuit is increasing when C is decreasing. If the tuner window is closed the user will be prompted to update or revert the components values on schematic.
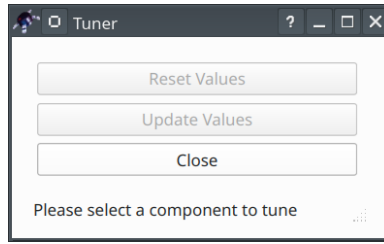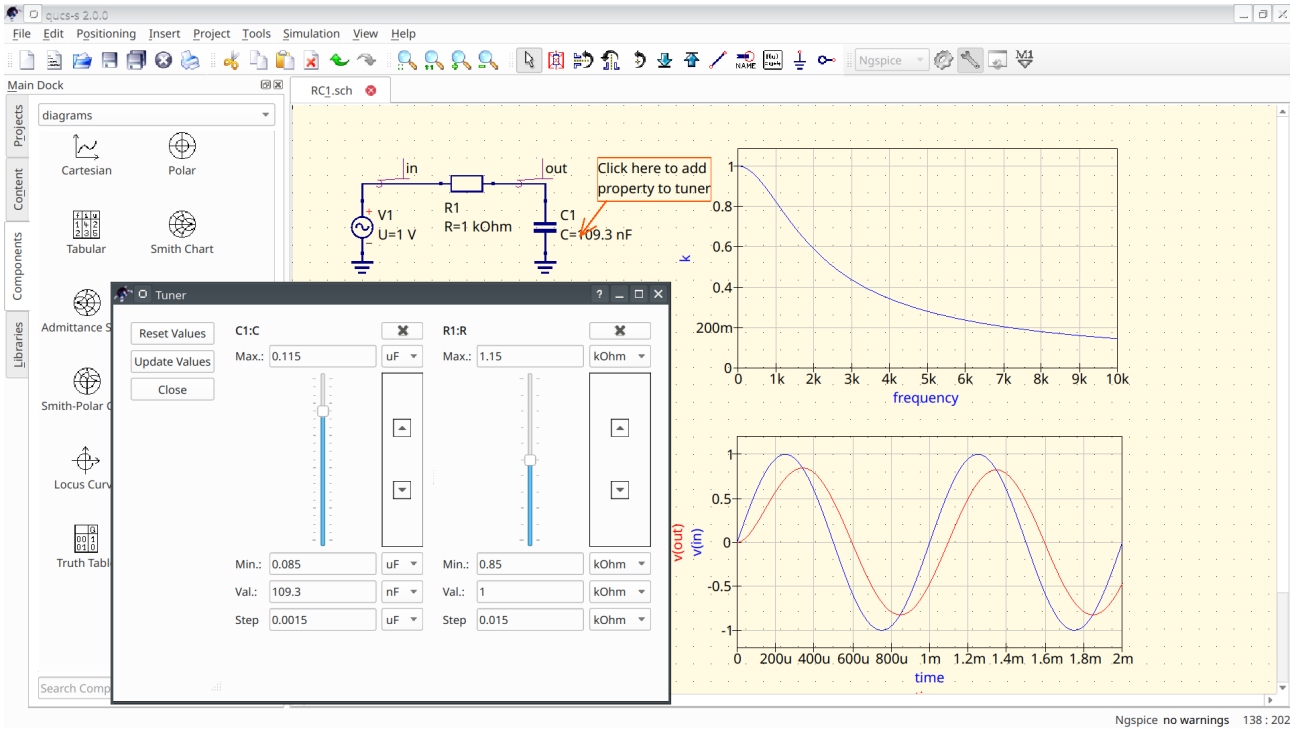
Figure 19: Tuner dialog window

Figure 20: Tuning of the RC-circuit

8 RF simulation

8.1 Common notes on RF circuits analysis

RF circuits are often described by the scattering matrix (S-parameters). Many electrical properties of networks of components (inductors, capacitors, resistors) may be expressed using S-parameters, such as gain, return loss, voltage standing wave ratio (VSWR), reflection coefficient and amplifier stability. It is possible to convert the S-parameters to the Y or Z-matrix to estimate the input and output impedances of the circuit.

The S-parameters of the two-port circuit have the following meaning:

- $S_{11} = \Gamma_1$ input reflection coefficient; if $S_{11} = 0$ the circuit is ideally matched without reflection;

- S_{21} forward power gain G_p from input (port 1) to output (port 2). For passive circuits always $G_p < 1$;

- S_{12} reverse power gain from output to input.

- $S_{22} = \Gamma_2$ output reflection coefficient;

13

The VSWR and reflection coefficients are related as the following:

$$\text{VSWR} = \frac{1 + |\Gamma|}{1 - |\Gamma|} \qquad (1)$$

8.2 RF simulation with Ngspice

Ngspice supports the S-parameter simulation since the version 37. Make sure that your Ngspice package has the version above 37. Otherwise this simulation mode will not work. There exists a special S-parameter simulation component that should be placed on schematic. The input and output ports of the circuit are represented by *power source* component (Fig. 21).
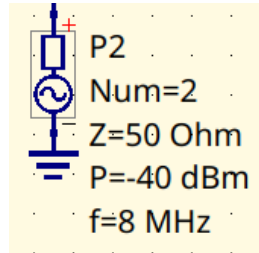

Figure 21: The power source representing 50-ohm port

The parameters of this device are port number `Num`, frequency f, and source power P expressed in dBm. Two such sources should be connected to schematic input and output nodes respectively.

Let's consider an example and simulate the frequency response of the bandpass filter for the 20m hamradio band. The schematic is shown in the Fig. 22.
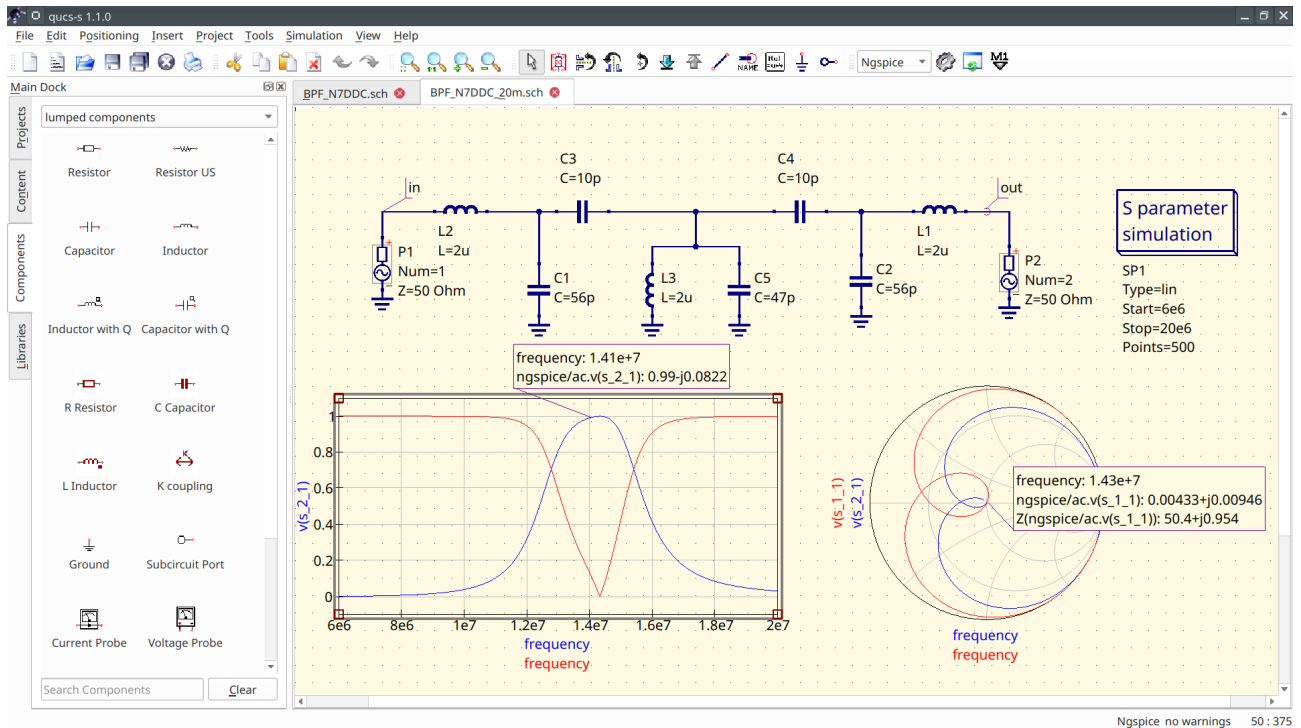

Figure 22: The simulation of the band-pass filter with Qucs-S and Ngspice

14

The filter consists of LC-devices that could be found in the *lumped devices* group. The filter schematic contains two 50-ohm port sources connected to its input and output nodes (`in` and `out`). The *S-parameter simulation* component defines the frequency sweep range from 6 MHz to 20 MHz.

The simulation of the filter circuit could be launched as usual using the F2 keyboard shortcut. After the simulation has been finished a number of diagrams could be placed on schematic. Ngspice represents the S-parameter simulation data as the virtual voltages.For example, it's need to pick the `v(s_2_1)` variable to plot the S_{21} dependency. The cartesian diagram contains the S_{11} and S_{21} dependencies. The S_{21} is the magnitude response of the filter. The same dependencies could plot on a Smith chart. Qucs-S provides a number of diagrams that could be used for RF circuits analysis. These diagrams are: Smith chart, admittance Smith chart, polar complex plane, and combined Smith chart and complex plane.

Ngspice converts the S-matrix to Y- and Z-matrix automatically. The Y and Z-parameters frequency dependencies could be plotted after the simulation. The Y and Z parameters are slow represented as the virtual voltages by Ngspice. For example the `v(z_1_1)` variable will represent the Z_{11} parameter. The simulation of the active RF circuits (like amplifiers) could be done similar to the passive circuits and will not be considered specially.

8.3 RF simulation with Qucsator

Qucsator simulation kernel provides some advanced features for RF circuits analysis. You have to install and configure Qucsator to get access to this simulator. See the section 9 for more details. It may be required to build Qucsator from source depending on your platform. Please keep in mind that Qucsator has very poor performance in the time domain and is not recommended for simulation of the general purpose circuits. Also Qucsator is not SPICE-compatible and uses another netlist syntax.

But Qucsator supports some advanced RF simulation techniques like multiport S-parameter simulation (including 1-port) and harmonic balance (HB) analysis. Qucsator provides a number of improved models for transmission lines (Fig. 23). The most frequently used transmission lines are listed below:

- 4-terminal transmission line and twisted pair;

- Coaxial cable;

- Microstrip lines, tees and corners;

- Coupled microstrips;

- Coplanar lines;

- Waveguides;

- Generic RLCG line;

Let's consider the basics of microstrip circuits simulation. Every microstrip line component requires to attach the substrate. The substrate is represented by special component too (Fig. 24). It is possible to define the substrate material properties and microstrip lines thickness. Substrate reference should be specified as the first parameter of all microstrip devices.

The simulation of the 1.5 GHz coupled microstrip bandpass filter is shown in the Fig. 25. The schematic contains two ports P1 and P2 connected to input and output, two microstrip devices MS1 and MS2, and substrate Subst1. The S-parameter simulation definition has no
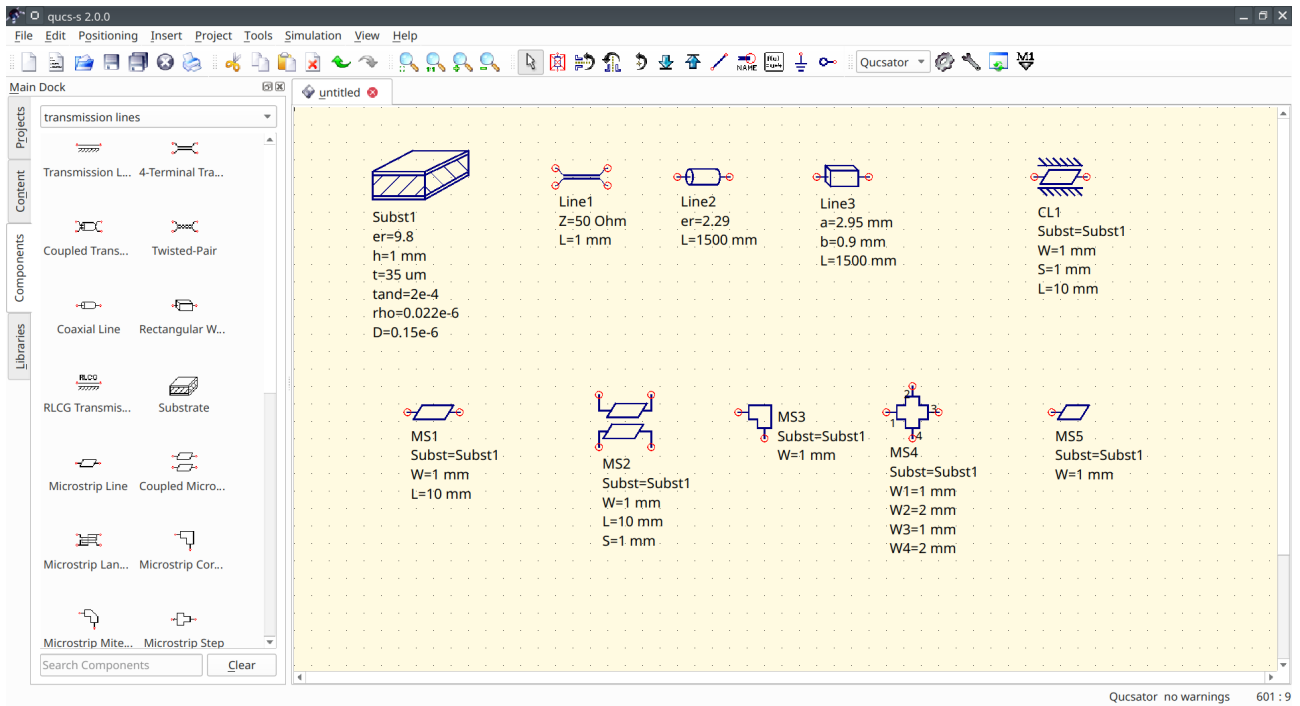
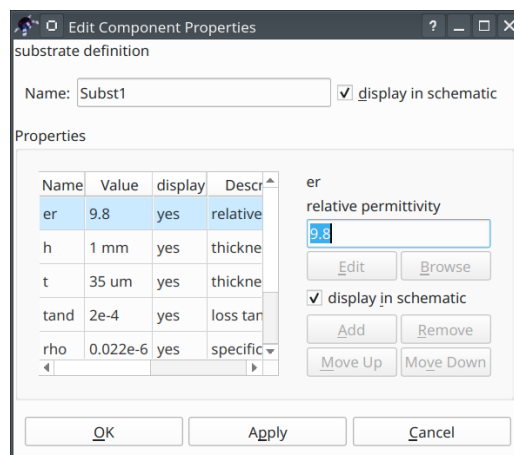Figure 23: Transmission lines models available for Qucsator backend

Figure 24: Substrate properties definition

difference to Ngspice and represented by a special component. The frequency range is defined from 1.3 to 1.7 GHz.

The schematic contains two equation that serves to express the S-parameters in Decibels using `dB()` function. Keep in mind that equation syntax for Qucsator is different. Ngspice equations will not work with Qucsator. The S-parameter is represented in `S[i,j]` form. For example `S[2,1]` for S_{21}. Qucsator doesn't calculate Y and Z matrices automatically like Ngspice. You have to use the functions `stoy()` and `stoz()` for matrix conversion. Refer to Qucsator documentation for more details.
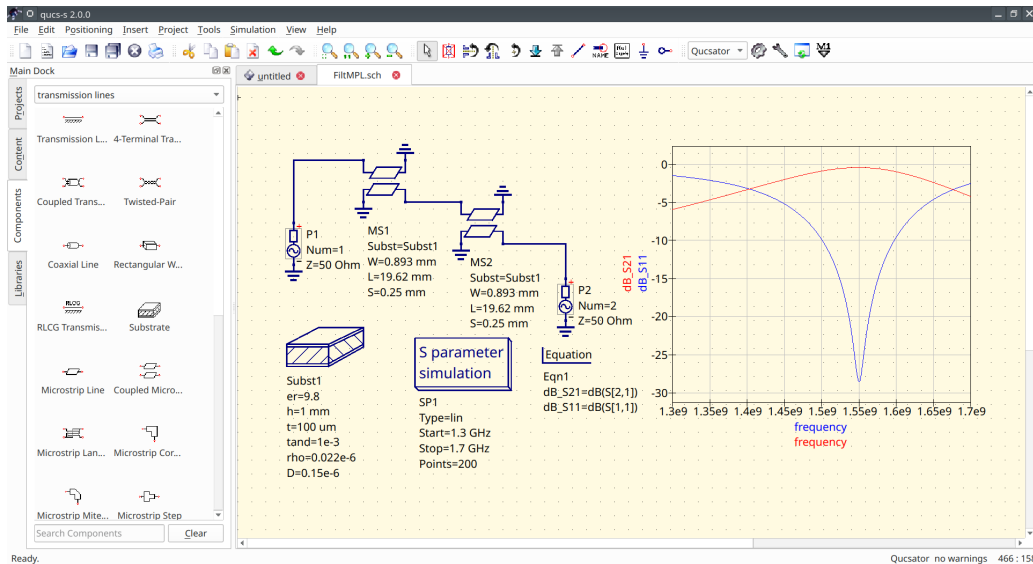
Figure 25: Coupled microstrip bandpass filter simulation

9 Quick switching of the simulation kernel

9.1 Quick simulator selection

Qucs-S supports a quick switching of the simulation kernel since the version 2.0.0. Application restart is not required anymore to apply this setting. The simulator could be selected at any time using the drop-down list on the toolbar Figure 26. The simulator location must be configure using the application settings dialog Figure 1 and the specified simulator executable binary must exist. Otherwise the simulator will be not shown in the drop-down list on the toolbar and not available for selection. The list of the supported simulation kernels could be found in the section 1.

Also keep in mind that Qucs-S distribution doesn't provide any simulation kernels and simulators must be installed separately. If you are using Linux DEB package, the Ngspice will be installed automatically as the package dependency. This selection affects only analog simulation. The digital simulation kernel will be executed automatically if the schematic contains the digital simulation (see the section 6.2).
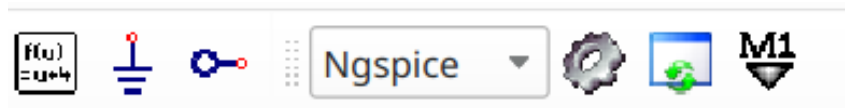

Figure 26: Simulator choose drop-down list

9.2 Incompatible components

Some of devices may be incompatible with the selected simulation kernel. For example, microstrip lines are designed for Qucsator and are not available with Ngspice. Such devices will be not available for insertion in the schematic and will be grayed out (see Figure 27) on the schematic view. An appropriate warning will be shown if the user tries to simulate the schematic containing incompatible devices.
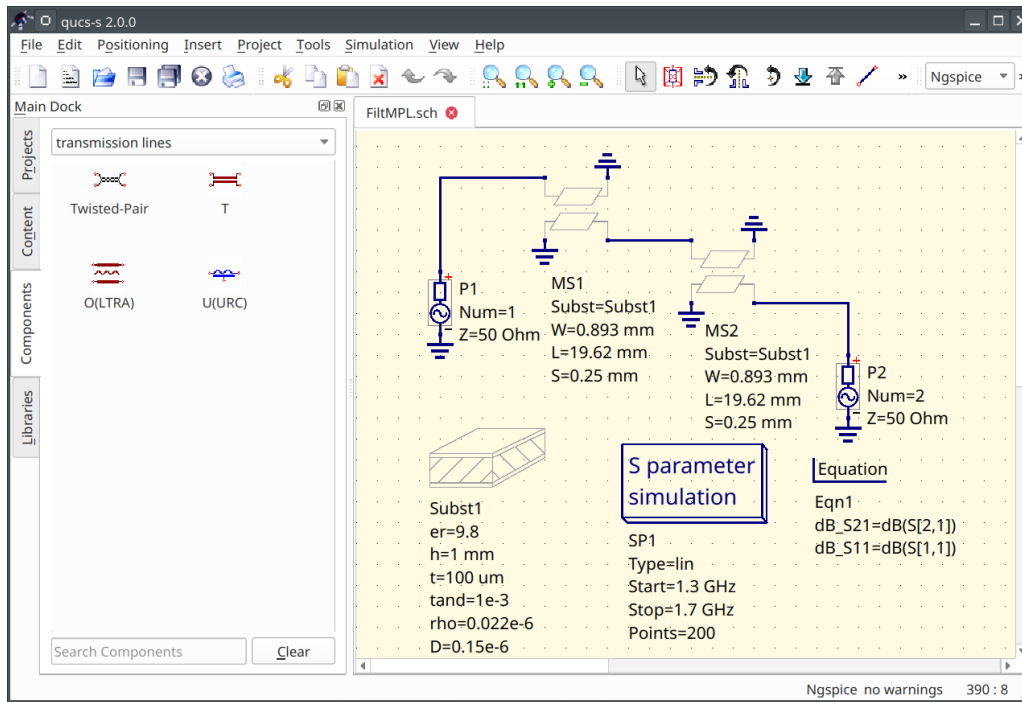
17

Figure 27: Schematic containing incompatible devices

10 List of mathematical functions supported by Nutmeg equation

The full list of the mathematical functions supported by Nutmeg equations could be found at the section 17.2 of the Ngspice manual. The next table shows the most frequently used functions. The input of all functions listed in the next table is a single vector representing voltage or current. Refer to the section 17 of the Ngspice manual for detailed description and information on additional functions.

Table 1: The list of the most frequently used Nutmeg functions

Function	Description
mag	Magnitude of vector (same as abs(vector))
ph	Phase of vector.
cph	Phase of vector. Continuous values, no discontinuity at $\pm\pi$
unwrap	Phase of vector. Continuous values, no discontinuity at $\pm\pi$ Real phase vector in degrees as input
real	The real component of vector
imag	The imaginary part of vector
conj	The complex conjugate of a vector
db	Decibels $20\log_{10}()$ of the vector
abs	The absolute value of vector (same as mag)
integ	Integrates over the given vector (versus the real component of the scale vector)
deriv	Calculates the derivative of the given vector
vecd	Compute the differential of a vector
group_delay	Calculates the group delay $-d\phi/d\omega$

11 Conclusion

In this tutorial it was considered how to perform a first start setup of the Qucs-S with the Ngspice backend and make an AC and Transient simulation of the simple RC circuit. It was shown how to add postprocessor equations and waveforms plot. Refer to official Qucs-S and Ngspice documentation to learn more advanced simulation techniques.