

DIEL
Calculation of circuits by a tensor method and diakoptic

Y.N.Sohor

dia-connect@hotmail.com
<http://diakoptic.ru>

On May, 5th 2014

Here the general description of the program DIEL, of format of the input and output data for results of calculations is presented. The present version it is test and is intended for development and improvement of program implementation of algorithms, and also for the educational purposes. The program is distributed under license GPL.

The general description.....	4
The basic works in Windows	5
Installation	5
Start on calculation	7
Formation of a database of subcircuits.....	7
The basic works in Linux	9
Installation	9
Start on calculation	10
Formation of a database of subcircuits.....	10
1. Input data.....	12
1.1. The circuit description.....	12
1.1.1. The general agreements	12
1.1.2. The netlist	12
1.1.2.1. Subcircuits (multipoles)	12
1.1.2.2. Branches communications (two-poles).....	13
1.1.2.2.1. The diode (DIOD)	13
1.1.2.2.2. The thyristor (TIR)	14
1.1.2.2.3. The keys controlled the functor (SWP, SWI).....	15
1.1.3. The job for calculation in the frequency area	16
1.1.3.1. Command of the job of calculation of the frequency characteristics.....	16
1.1.3.2. The job for output in a format txt.....	16
1.1.3.3. Fast displaying of results of calculations	17
1.1.3.4. Displaying of graphics during calculation	17
1.1.3.5. The job for output in a format csv.....	18
1.1.4. The job of calculation of transient process	19
1.1.4.1. Command of the job of calculation of transient process.....	19
1.1.4.2. Command of the job of a method of calculation.....	19
1.1.4.3. Job for output	19
1.1.4.4. Fast displaying of results of calculations	20
1.1.4.5. Displaying of graphics during calculation	20
1.1.4.6. The job for output in a format csv.....	21
1.1.5. Modification of subcircuits.....	22
1.1.5.1. Change of nominal's of subcircuits by a command .mod	22
1.1.5.2. Change of nominal's of subcircuits by a command .param	23
1.1.5.3. Change of sources of subcircuits by a command .mod.....	23
1.1.5.4. The description of functions.....	24
1.2. The description of subcircuits	26
1.2.1. Composition of files.	26
1.2.2. The netlist for subcircuits	26
1.2.3. The description of subcircuits.....	28
1.2.4. Parameterization of subcircuits	28
1.2.5. Input of functions in subcircuits	29
1.2.5.1. Input of hierarchical functions	29
2. Output data.....	31
2.1. Resultants files in a format txt (calculation in frequency area).....	31
2.2. Resultant files in a format csv (calculation in frequency area)	32
2.3. Resultant files in a format txt (calculation in temporal area)	32
3. Example	34
3.1. Filter	34
3.1.1. Preparation of a subcircuit of the amplifier	34
3.1.1.1. Preparation of the netlist	34
3.1.1.2. Documentation preparation.....	34
3.1.1.3. Preparation of a file with rated determination of a subcircuit in a binary format	34
3.1.2. Preparation of a subcircuit of a source of voltage	35
3.1.2.1. Preparation of the netlist	35

3.1.2.2. Documentation preparation.....	35
3.1.2.3. Preparation of a file with rated determination of a subcircuit in a binary format.....	35
3.1.3. Preparation of a subcircuit with capacity (for calculations in temporal area)	35
3.1.1 Preparation of the netlist	35
3.1.2 Documentation preparation.....	35
3.1.3 Preparation of rated files in a binary and text format	35
3.1.4 Database formation.....	36
3.1.5 Preparation of a file with the rated description of the circuit	36
3.1.5.1. Preparation of the netlist of the rated circuit of the filter for calculation in the frequency area.....	36
3.1.5.2. Preparation of the netlist of the rated circuit of the filter for calculation in temporal area.	36
3.1.6. Calculation performance.....	37
3.1.7. Displaying of results	37
4. Preparation of graphical representation of subcircuits.....	39
4.1. Graphical representation of internal composition of subcircuits.....	39
4.1.1. Preparation of graphic library Two-poles.....	39
4.1.2. The assembly of a subcircuit from library elements.....	40
4.2. Graphical representation of a subcircuit in the form of a multipole	41
4.3. Input of the coordinates and scales.....	41

The general description

The software is developed for calculations of hierarchy circuits by methods of the tensor analysis of networks and diakoptic. It is planned also to application for joint calculation of circuits and the fields presented by electrical equivalent circuits.

Calculation in the present version is fulfilled in a command line: the executed file which reads files with the input data is launched. Depending on the job for calculation, the program forms files with results of calculation.

The software has been tested with usage of following compilers:

- In operating systems OpenSuse and RedHat:
 - Intel® Fortran Composer XE 2011 and above
 - g95
 - gfortran
- In operating systems Windows XP, 7:
 - g95
 - gfortran

Now versions for calculation of transient and the frequency analysis are offered:

For Linux32:

- The single-processor versions compiled with **gfortran**
- The multiprocessor version (technology, coarray) compiled with **ifort**

For Win32:

- The version the single-processor version compiled with **gfortran**

The basic works in Windows

Installation

Before installation **DIEL** check up the necessary software:

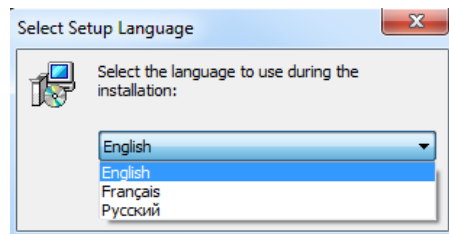
- The text editor
- The program of reading of files **pdf** (Adobe)
- For operation with graphic displays of results of calculation can be demanded in addition setting of packets gnuplot, scilab, paraview, etc.

For installation **DIEL** is not required the rights of the administration. The software is installed in folder **DIEL**. At an installation stage it is a name and a path it is possible to change. After installation and installation paths it is not recommended to change a name, change of paths to executed files and a database otherwise is required. At installation does not register any data in the register, or other places, except an installation directory.

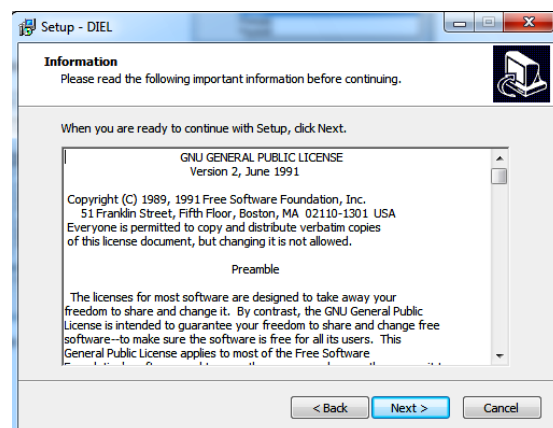
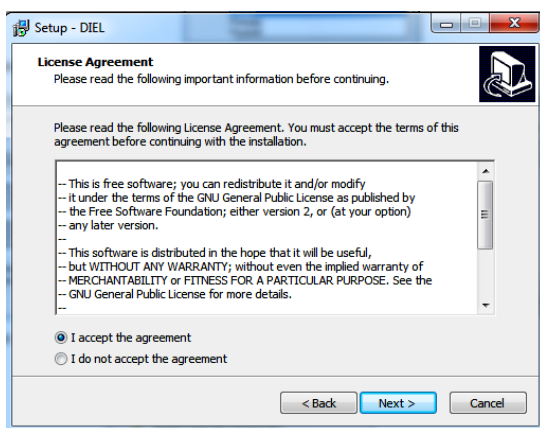
After setting to fulfill calculations it is possible from the circuit editor or from a command line - see further.

The main stages of setting:

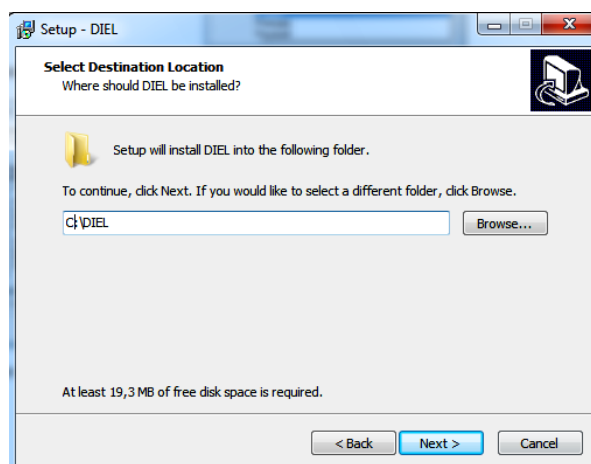
- A choice of language of setting



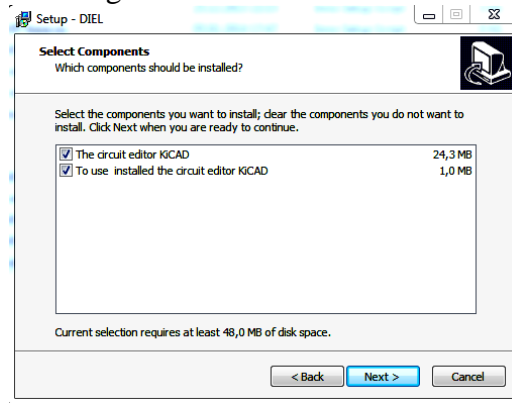
- Acceptance of software license agreements



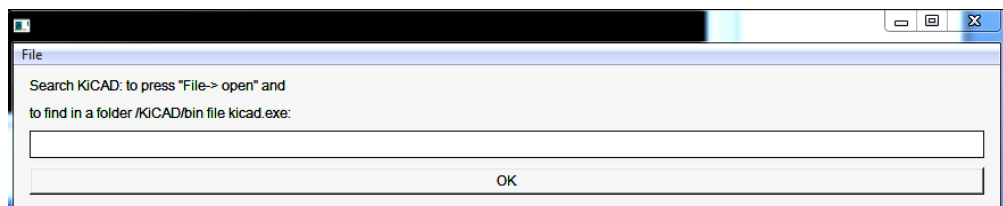
- A choice of a directory of installation



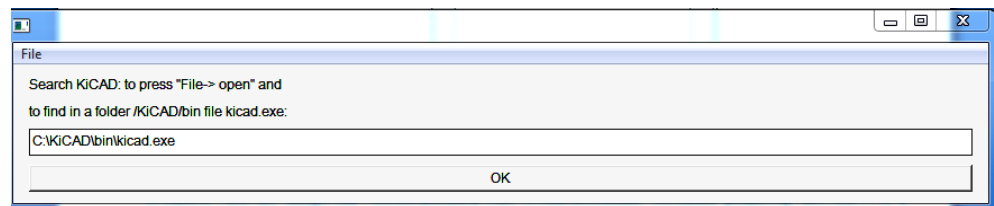
- The invitation for a choice of additional setting of circuit editor KiCAD



Possible variants of installation: - if to remove all ticks then installation without the circuit editor will be executed. If to leave a tick opposite «The circuit editor» then circuit editor KiCAD and command files for start of DIEL from the circuit editor will be installed. If to leave a tick opposite «To use installed the circuit editor KiCAD» then in the end of installation of DIEL there will be an invitation for file search **kicad.exe**:



After pressing *File-> open* it is necessary to find a file **kicad.exe** - it is in a folder **bin** systems **KiCAD**:



After pressing **OK** command files **diel.bat** and **c2c.bat** will be copied in a folder **bin** systems **KiCAD**.

After that, all is ready to setting. After its end, there will be an invitation to reading **README.txt**. As a result in a root folder following basic elements will be installed:

\base - a database of subcircuits

\base\DOC\en - The description of operation with a database

\bin – contains executed files **DIEL**

\example - contains examples of calculation of circuits

\RLC - the elementary circuit for initiating learning.

\FiltrT - filter calculation

\Maillon - an example of modifications of subcircuits.

\ad-3faz - Calculation of start-up of the asynchronous engine

...

\doc

\Contains description DIEL

\KiCAD – the circuit editor (it is not installed at a canceling of setting **KiCAD**)

\bin - contains executed files **KiCAD** and command files **DIEL**

\doc – the description of operation with **KiCAD**

\share – language superstructures **KiCAD**

\utilite - contains the additional software

Label **DIEL** - for start of the circuit editor (at a canceling of setting **KiCAD** at an installation stage, this label the same is not installed)

DIEL.bat - the command file for start **DIEL** in a command line mode

c2c.bat - the command file for converter start (**xcir2sun-cep.exe**) from format **Spice** in format **DIEL**

uninst000.dat, **uninst000.exe** – files for an uninstallation (it is possible to delete also means Windows)

Start on calculation

Start methods:

1st method: (if the circuit editor is installed at installation **DIEL**)

- In a root folder, where the program is installed to launch label **DIEL**;
- The circuit editor will as a result be launched;
- In the circuit editor to create the project or to open the ready project in one of folders **DIEL\example**;
- In the circuit editor to create the circuit (or to load the ready circuit), to prepare it for calculation (The library for operation in **DIEL** is in a directory **\base** and has a name: **diel.lib**);
- In the circuit editor to set a command **Tools - generate netlist**;
- To select bookmark **Spice**;
- To mark "Use net names";
- In line "Simulator command" to print **DIEL.bat**;
- To launch calculation.

2nd method: start in command line

- To create a folder, where to create the text file with **.cir** extension with the netlist and the job for calculation, for example **mysch.cir** (if it is supposed to deduce results in 3d a format – that should be present still the file with the same name with the circuit with **sch** extension);
- To copy from a root folder the command file **DIEL.bat**;
- To print in command line: **DIEL.bat mysch.cir**;
- To launch calculation.

Formation of a database of subcircuits.

For calculations the description of subcircuits in binary files with **SUN** extension is used. For obtaining of such file it is possible to use 2 methods:

1st method: (if the circuit editor is installed at installation **DIEL**)

- To create a folder with the project, for example, **soussch** or it is possible to use a ready folder with database **DIEL\base**;
- To launch label **DIEL** in root folder **DIEL**;
- To create the subcircuit project, for example, **soussch.pro**;
- In the circuit editor to create a subcircuit, for example, **soussch.sch**
the library for operation in **DIEL** is in a directory **\base** and has a name: **elem.lib**;
- In the circuit editor to set a command **Tools - generate netlist**;
- To select bookmark **Spice**;
- To mark "Use net names";
- In line "Simulator command" to print **c2c.bat**;
- To launch calculation;
- Files should be as a result generated: a binary file **soussch.sun** and in addition the text file **soussch.cep**.
The received descriptions of subcircuits can be used in the same directory, as the circuit, or it is possible to copy in a folder with database **DIEL\base**.

2nd method: start in command line

- To create a folder with the project, for example, **soussch** or it is possible to use a ready folder
With database **DIEL\base**;
- In a folder to create the text file with the list of connections of a subcircuit, for example, **soussch.cir**
- Further two variants are possible:
 - if in a working folder there is only one file with the list of connections,
that in command line to print **c2c1.bat** and to launch calculation.

- if in a working folder there are some files with lists of connections,
that in command line to print **c2c.bat mysch.cir** where **mysch** – name of the circuit and to launch calculation.
 - Files should be as a result generated: a binary file **soussch.SUN** and in addition the text file **soussch.CEP**. The received descriptions of subcircuits can be used in the same directory, as the circuit, or it is possible to copy in a folder with a database **\base**
- Graphical representations of subcircuits in the form of multiport form in system **KiCAD** and are saved in a file with lib extension (sections 3, 4 and a manual on circuit editor **KiCAD** see in the present description).

The basic works in Linux

Installation

Before installation **DIEL** check up the necessary software:

- The text editor
- The reader program of files **pdf** (Adobe)
- For operation with graphic displays of results of calculation setting of packets gnuplot, scilab, paraview, etc. can be demanded in addition

main stages of installation:

- To unpack archive in a house folder
- To launch the utility **create-diel**
- To copy generated label **Diel-kicad.desktop** on a desktop

It is possible to work from the circuit editor through label start on a desktop or from command line (the operating procedure see further).

Now in more details.

archive is unpacked in one of the folders allocated in a directory **/home /** As a result in a root folder following basic elements will be installed:

/base - a database of subcircuits

/base/DOC – a folder with descriptions of operation with a database

/bin – contains executed files **DIEL**

/example - contains examples of calculation of circuits:

/RLC - the elementary circuit for initiating learning.

/FiltrT - filter calculation

/Maillon - an example of modifications of subcircuits.

/ad-3faz - calculation of start-up of the asynchronous engine

...

/doc

/ contains description **DIEL**

/kicad – the circuit editor

/bin - contains executed files **kicad** and command files **DIEL**

/doc – the description of operation with **kicad**

/lib – a folder for library of circuit elements **DIEL**

/share – language superstructures **kicad**

/dislin - contains the additional software

remove-lnk-diel, create-lnk-diel,– utilities for formation of command files and labels **DIEL**

README.txt – a short manual for start **DIEL**

After archive unpacking it is necessary to launch the utility **create-diel** for formation of command files and labels:

./create-diel

Will be as a result created:

- Command files in a folder **.../kicad/bin**

diel, diel1 – for start of the main program from the circuit editor

c2c, c2c1 – for start of the program of formation of the description of subcircuits from the circuit editor

- Links to command files from a folder **/home / .../bin**, allowing working in a command line mode:

@diel, @diel1, @c2c – for start of the main program and the program of formation of elements of a database from command line

- Label for layout on a desktop, allowing launching calculations from the circuit editor:

Diel-kicad.desktop

If needed, to specify a folder path \$HOME/bin. In a file \$HOME/.bashrc to print out of a line:
PATH=\$PATH:\$HOME/bin
export \$PATH

At usage of the parallel version it is necessary to modify, if needed, a line in files **diel** and **diel1**:
source /opt/intel/composer_xe_2013.1.117/bin/compilervars.sh ia32
where the path to the Intel compiler is underlined

After setting and setting paths it is not recommended to change a name, change of paths to executed files and a database otherwise is required.

Start on calculation

Start methods:

1st method: (usage of the circuit editor)

- To launch label **Diel-kicad** on a desktop;
- The circuit editor will as a result be launched;
- In the circuit editor to create the project or to open the ready project in one of folders **/example**;
- In the circuit editor to create the circuit (or to load the ready circuit), to prepare it for calculation (The library for operation in **DIEL** is in a directory **/base** and has a name: **diel.lib**);
- In the circuit editor to set a command **Tools - generate netlist**;
- To select bookmark **Spice**;
- To mark "Use net names";
- Further two variants are possible:
 - if in a working folder there is only one file with the circuit in a line "Simulator command" to print **diel1** and to launch calculation.
 - if in a working folder there are some files with circuits in a line "Simulator command" to print **diel mysch.cir** where **mysch** – a name of the circuit and to launch calculation.

2nd method: start in command line

- To create a folder, where to create the text file with .cir extension with the list of connections and the job for calculation, for example **mysch.cir** (if it is supposed to deduce results in 3d a format – that should be present the file with the same name with the circuit with **sch** extension);
- Further two variants are possible:
 - if in a working folder there is only one file with the list of connections in command line to print **diel1** and to launch calculation.
 - if in a working folder there are some files with lists of connections in command line to print **diel mysch.cir** where **mysch** – a name of the circuit and to launch calculation.

Formation of a database of subcircuits.

For calculations the description of subcircuits in binary files with **SUN** extension is used. For obtaining of such file it is possible to use 2 methods:

1st method: (usage of the circuit editor)

- To create a folder with the project, for example, **soussch** or it is possible to use a ready folder with database **/base**;
- To launch label **Diel-kicad** on a desktop;
- To create the subcircuit project, for example, **soussch.pro**;
- In the circuit editor to create a subcircuit, for example, **soussch.sch**
The library for operation in **DIEL** is in a directory **/base** and has a name: **elem.lib**;
- In the circuit editor to set a command **Tools - generate netlist**;
- To select bookmark **Spice**;
- To mark "Use net names";
- Further two variants are possible:

- if in a working folder there is only one file with the circuit in a line "Simulator command" to print **c2c1** and to launch calculation.
 - if in a working folder there are some files with circuits in a line "Simulator command" to print **c2c mysch.cir** where **mysch** – a name of the circuit and to launch calculation.
- Files should be as a result generated: a binary file **soussch.sun** and in addition the text file **soussch.cep**.
The received descriptions of subcircuits can be used in the same directory, as the circuit, or it is possible to copy in a folder with a database **/base**.

2nd method: start in command line

- To create a folder with the project, for example, **soussch** or it is possible to use a ready folder with database **/base**;
- In a folder to create the text file with the list of connections of a subcircuit, for example, **soussch.cir**;
- Further two variants are possible:
 - if in a working folder there is only one file with the list of connections in command line to print **c2c1** and to launch calculation.
 - if in a working folder there are some files with lists of connections in command line to print **c2c mysch.cir** where **mysch** – a name of the circuit and to launch calculation.

Files should be as a result generated: a binary file **soussch.SUN** and in addition the text file **soussch.CEP**.

The received descriptions of subcircuits can be used in the same directory, as the circuit, or it is possible to copy in a folder with a database **/base**.

Graphical representations of subcircuits in the form of multiport form in system **kicad** and are saved in a file with lib extension (sections 3, 4 and a manual on the circuit editor see in the present description).

1. Input data

The input data are: a file with the description of the circuit and files with the description of subcircuits.

1.1. The circuit description

The circuit description is allocated in the text file with cir extension and consists of three parts:

- 1) the netlist
- 2) the job for calculation
- 3) modification of subcircuits (if needed)

Lines of the netlist include the information on subcircuits, branches of communications and names of circuit nodes.

Lines with the job for calculation should begin with a keyword in which beginning the point is allocated.

Lines with modification of subcircuits also should begin with a keyword in which beginning the point is allocated.

1.1.1. The general agreements

The register of characters has no value (all characters during handling are translated in an uppercase). The format of the data does not suppose transposition of a line for a new line.

Format of the job of numbers. If in line numbers in certain cases these numbers can be supplied with scale factors are resulted. The scale factor can be installed in the end of number, in the middle of number instead of a decimal point, but not in the number beginning. Gaps in number are inadmissible, it is impossible to put also gaps between a scale factor and digits. Following scale factors are admissible:

PET – 10^{15} , **T** – 10^{12} , **G** – 10^9 , **MEG** – 10^6 , **K** – 10^3 , **M** – 10^{-3} , **U** – 10^{-6} , **N** – 10^{-9} , **P** – 10^{-12} , **F** – 10^{-15} .

Instead of scale factors it is possible to apply an engineering record format of number with the job of a decimal level. A decimal level of number separates **E** character. Instead of a decimal point it is possible to use a decimal comma.

Example: $1200 = 1.2k = 1.2K = 1k2 = 1K2 = 1.2e3 = 1,2e3 = 1,2k = 1.2E3 = 1,2E3 = 1,2K$

The description of the circuit and the job for calculation should come to an end with line which begins with **.END** characters

1.1.2. The netlist

Initial data about the circuit prepares in the form of the netlist (netlist) two-poles and multipoles. The circuit should be connected. Number of subcircuits in the circuit not less than 2. The netlist consists of separate text lines. To comments are applied: * character in the beginning of a line and the character a semicolon ";" in any place of a line. After a semicolon it is possible to place the comment.

Length of words the netlist – no more than 8 characters.

1.1.2.1. Subcircuits (multipoles)

In line with the subcircuit description the words divided by gaps are allocated. A word order following:

<Name> N1 N2 ... Nk <Base> [Zmod] [Smod]

Name - a subcircuit name. The name should begin with ò character.

N1 N2 ... Nk - names of circuit nodes. Names of circuit nodes are allocated as numbers of outputs of a subcircuit. If the name of a circuit node is presented by the character "?" (Question mark) it is considered that a subcircuit output is not connected to the circuit.

Base - standard, or a typical name of a subcircuit with which it is present at a database. The typical name of a subcircuit is underlined in a line of the description of a subcircuit right after listings of numbers of nodes.

Zmod - a modifier name on parameters, **Smod** - a modifier name on sources. Names of modifiers are used for change of basic parameters of subcircuits. Commands **.mod** which description is considered further are for this purpose applied.

Example:

X1 n1 n2 DER0 Z1 S1

- The subcircuit of type **DER0** has circuit name **X1**, is connected to circuit nodes **n1 n2**. The subcircuit will be modified on parameters and on influences. A modification name on parameters - **Z1**, a modification name on influences - **S1**.

1.1.2.2. Branches communications (two-poles)

For connection of subcircuits it is possible to apply **RLC** branches – to calculations in the frequency area and **R** branches for calculations in temporal area. Sources in branches of communications in the given version are not installed. The job of a branch of communication: a name of two circuit node and a nominal.

<Name> N1 N2 <Value> >|<Type>

Name – a branch name.

N1 N2 – names of the first and second node of the circuit. The positive direction of a current of a branch: from the first node to the second.

Value – a branch nominal. Number in which the scale factor can be used.

Type – branch type. For transient can accept values:

DIOD - the diode,

TIR - the thyristor,

SWP - a key controlled the positive values of the functor,

SWI - a key controlled the negative values of the functor

For the frequency analysis the branch type can accept values:

SWP - the key controlled the positive values of a real part of the functor,

SWI - the key controlled the negative values of a real part of the functor

On the first character of a name the branch type is defined

*For the frequency analysis following characters are admissible:

R – resistor, **L** – inductance, **C** – capacity, **S** – controlled key (for **SWP**, **SWI**)

* For transient following characters are admissible:

R – the resistor

D – the diode

S – controlled key (for **TIR**, **SWP**, **SWI**)

Example:

R1 n1 n2 10k

C1 n-1 n-2 1u

Capacity C1 is connected to nodes **n-1** and **n-2** and has a nominal 1 mkF

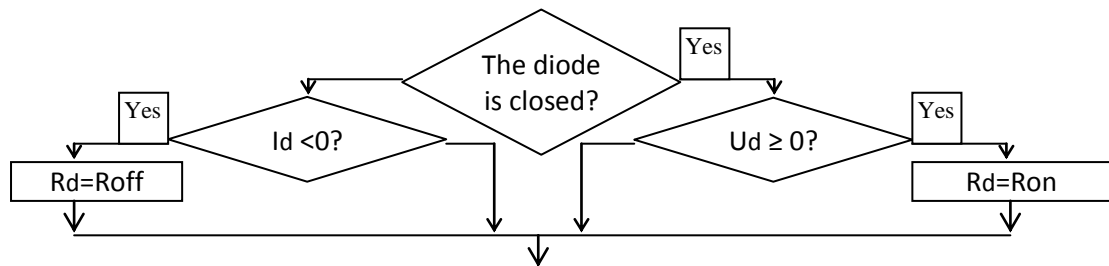
1.1.2.2.1. The diode (DIOD)

The rated circuit of the diode is presented a kusochno-broken line by a line. By default resistance of the open diode equally 0.1mOhm. Resistance of the closed diode by default is equal 1GOhm.

These values can be redefined. For this purpose in line after type of branch **DIOD**, the keyword and through a gap, numerical value is underlined. Following keywords are admissible:

Roff - for the closed diode and **Ron** for the open diode

Diode state is defined by following algorithm:



U_d – voltage on the diode, I_d – a diode current, R_d - resistance of the diode, R_{off} - resistance of the closed diode, R_{on} - resistance of the open diode.

Final state of diodes is defined after iterations.

Specification of transition point of a current (voltage) through a zero in the given version is not provided. In starting state all diodes are closed.

Example:

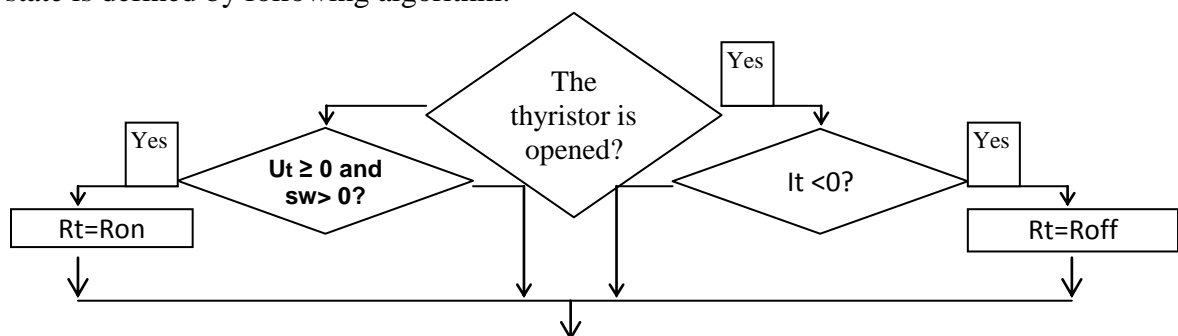
D1 n1 n2 DIOD Ron 1u Roff 1Meg

1.1.2.2. The thyristor (TIR)

The rated circuit of the thyristor is presented a kusochno-broken line by a line. By default resistance of the open thyristor equally 0.1MOM. Resistance of the closed thyristor by default is equal 1GOM. For redefinition of these values in line after type of branch **TIR**, the keyword and through a gap, numerical value is underlined. Following keywords are admissible:

Roff - for the closed thyristor and **Ron** for the open thyristor

thyristor state is defined by following algorithm:



U_t – voltage on the thyristor, I_t – a thyristor current, sw – controlling pulse of the thyristor, R_t - resistance of the thyristor, R_{off} - resistance of the closed thyristor, R_{on} - resistance of the open thyristor.

At the assembly of circuits names of thyristor branches should begin with S character and have typical name **TIR**.

Controlling pulses of thyristors are defined in a text line after a keyword **.sw**. Steams of words consisting of a name of the thyristor and a name of controlling function Further should follow. For example,

.sw S1 imp1 S2 imp2.

Here names of controlling functions are defined: for thyristor **S1** function name **imp1**, for thyristor **S2** function name **imp2**. Functions and their arguments should be presented in separate lines. For example:

.fun imp1=cos (w*t-fi)

.arg w = ' 2*pi*50 ' t=t fi ='pi'

.fun imp2=cos (w*t)

.arg w = ' 2*pi*50 ' t=t

At those moments of time when controlling function is more than zero, it is considered that on the thyristor controlling pulse is submitted.

The final state of thyristors is defined after iterations. Specification of transition point of a current (voltage) through a zero in the given version is not provided. In starting state all thyristors are closed.

1.1.2.2.3. The keys controlled the functor (SWP, SWI)

This type of a key is completely controlled key. By default resistance of public key equally 0.1mOhm. Resistance of private key by default equally 1GOhm.

For redefinition of these values in line after branch type, the keyword and through a gap, numerical value is underlined. Following words are admissible:

Roff - for private key and **Ron** for public key

Two varieties of a controlled key are admissible: **SWP** and **SWI**.

SWP – a key unlocked by the positive values of the functor

SWI - a key unlocked by the negative values of the functor

Allocation of functors on keys is produced in line which begins with a command **.sw** Further streams of words consisting of a name of a key and a name of controlling function should follow.

Example

Let in the circuit there is two keys **S1** having type **SWP** and key **S2** having type **SWI**

If keys have to work in an antiphase they can be controlled by one function **F** having passages through a zero:

.sw S1 F S2 F

Function and its arguments:

.fun F=cos (w*t)

.arg w = ' 2*pi*50 ' t=T

Now, during those moments of time when controlling function **F** is more than zero, on key **S1** the unlocking signal is given, simultaneously on **S2** the disabling signal is given.

The key can work in trigger Shmitt mode if to add two values:

At <val> Ah <val>

In this case unlocking of key **SWP** will happen, if value of controlling function is more or equally **At**, key lock-out will be at values of controlling function less **Ah**

By default **At=Ah=0**

For example:

S1 n1 n2 SWP At 5 Ah-3

1.1.3. The job for calculation in the frequency area

Commands on control of calculation are allocated in the netlist in separate lines.

Each line begins with a keyword. In the beginning of each keyword there should be a point (for spice programs these commands name dot-commands).

The applied program of calculation **diel** is not the spice-program, but uses spice-like language for the netlist and commands for control of calculation.

Let's result the main commands.

1.1.3.1. Command of the job of calculation of the frequency characteristics

Examples:

.AC dec 5 10MEG 2G – calculation will be fulfilled for 5 points for decade, change of frequency from 10MHz to 2GHz.

.AC list 10 50 5k 20MEG 0.1G calculation will be fulfilled for frequencies 10Hz, 50Hz, 5kHz, 20 MHz 0.1GHz

The general description:

The job can be in two formats:

.AC <oct | dec | lin> <Npoints> <StartF> <EndF>

.AC <list> <f1> [<f2> [<f3> [...]]]

In the first format after a keyword the type of rise of frequency at first is underlined: in octaves or decades or the linear rise of frequency. Further it is underlined **Npoints** - number of points. For the linear rise of frequency is a total number of values of frequency. For octaves and decades – number of points on an octave or decade. Then it is underlined initial **StartF** and finite **EndF** frequencies. By default a unit of measure of initial and finite frequency - in Hertz (it is possible to install scale factors).

In the second format frequency is set by the list. After keywords **.AC** and **list** values of frequencies through a gap are resulted.

1.1.3.2. The job for output in a format txt

Example:

.print Ad U (Rvny (X1)) Ph (U (Rvny (X1)) Re I (C1 (-)) - print the amplitude of voltage in decibels on resistance **Rvny** of subcircuit **X1**, a phase of the same voltage, a real part of a current in a communication branch - capacities **C1** (does not enter into subcircuits) will be printed.

The general description:

.print <Vid₁> <Tip₁> <Br₁> <X₁> [<Vid₂> <Tip₂> <Br₂> <X₂> [<Vid₃> <Tip₃> <Br₃> <X₃> [...]]]

After a keyword **.print** quadruples of words of the following content are specified:

Vid - form type. Following words are admissible:

Am – amplitude

Ad – amplitude in decibels (20lg)

Ph – a phase (in degrees)

Re – a real part of number

Im – an imaginary part of number

Cm – complex number: real and imaginary parts in a format (Re, Im)

Tip - value type. Following words are admissible:

U – voltage on nodes of the generalized branch ($U=V-e$, where **V** voltage on an internal branch, **e** - a source of voltage of the generalized branch)

I – a branch current

P – capacity of a branch ($P=U*I$)

Br – a branch name. To learn names of the branches entering into a subcircuit it is possible only in its description. Names of branches of the communications connecting subcircuits, are resulted in the netlist of the circuit and installed at circuit creation.

X - a subcircuit name. All names of subcircuits should begin with the character **X**. These names are resulted in the netlist of the circuit and installed at circuit creation. For branches of communications instead of a subcircuit name the character "-" (minus) should be specified

As a separator between words the gap is used. For convenience of perception as a separator brackets can be used.

Command action **.print** leads to print results of calculation in files:

W.txt – Values of frequencies, **Result<i>.txt** – values of the calculated values where **i** changes from 1 to the number corresponding to an amount of schedules, defined in the job.

1.1.3.3. Fast displaying of results of calculations

If to set a command

.QPLOT [img]

After of terminations of calculation, results will be displayed in the form of the graphs sequentially displayed in windows. The amount of windows corresponds to number of the graphs defined in a command **.print**

img – an option, allowing to output images in graphic files. The option can accept following values: **gif**, **png**, **jpg**

Commands:

.leg <text1> <text2> ... <text i> or

.LEG <text1> <text2> ... <text i>

allow to enter inscriptions at the top of graphics. Text fields should correspond strictly to the variables set in a command **.print**. In a text field there should not be blanks. If the blank nevertheless is necessary in an inscription in a text field the underlining sign.

Example

.print Ad U(CJ1(X2)) Ad U(CJ1(X4))

.leg w_rad/s_for_2faz w_rad/s_for_3faz

If the command **.LEG** is not specified or on a corresponding place in this command instead of a text field the sign a minus as an inscription the text set in a command **.print** is used.

1.1.3.4. Displaying of graphics during calculation

The command

.FLOW [(t_i t_j ... t_k)] [(t_i t_m ... t_n)] [(x_i x_j)] [(x_m x_n)] [B[UFSIZE]=N] [D[ELAY]=M] [C[OLORS]=B[LACK]]

The command **flow** is used together with a command **.print** and leads to that graphics are displayed during calculation. In parentheses it is possible to group graphics with instructions of their numbers. Number should correspond to number of a position of the graphic in a command **.print**. At grouping before number it is necessary to deliver either character **T** or character **X**. If character **T** specified, then graphics will be deduced in function from frequency. If character **X** specified, then graphics should be grouped on two, frequency is used as parameter, and graphics will be deduced as function of dependence of second number from the first number.

Total number of graphics - is no more 10.

Additional keys:

B[UFSIZE] =N - number **N** specifies in an amount of the points displayed in a window (all points are by default displayed),

D[ELAY] =M - number **M** specifies in milliseconds in a time lag at display of graphics (by default - without a time delay)

[C[OLORS]] =B[LACK]] - the job of a black background for graphics (by default a background white)

The command without keys leads to displaying of all graphics in one window

Example:

.print Am I(R1(X1)) Am I(R2(X1)) Am I(R3(X1)) Am I(R4(X1)) Am I(R5(X1)) Am I(R6(X1))

.flow (t2 t5 t1) (t4 t1) (x1 x3) b=1000 d=2 c=b

Here the command **print** sets an output of graphics of a current in resistors with 1st on 6th.

The command **flow** sets displaying of three groups of graphics. As function of frequency two groups: 2nd 5th and 1st; 4th and 1st. One more graphic will be displayed in the form of dependence of a current in resistor R3 from a current in resistor R1. The size of the buffer - 1000 points, a time delay - 2msec, a background - is black.

Note: for displaying of graphics on command **flow** is used the utility **q2d** (<http://sourceforge.net/projects/q2d/>) .

1.1.3.5. The job for output in a format csv

It is one more type of the job for an output of the rated information. It is applied to an output of great volumes of the data in 3D a format. Each point is allocated on a surface. Coordinates X, Y correspond to branch coordinates on the schematic diagram, coordinate Z corresponds to value of the calculated value. The command format is similar to a command format **print**:

.csv <Vid₁> <Tip₁> <Br₁> <X₁> [<Vid₂> <Tip₂> <Br₂> <X₂> [<Vid₃> <Tip₃> <Br₃> <X₃> [...]]]

Besides, in names of branches and subcircuits the template character * can be used. Examples:

.csv Am U (* (X*)) - the amplitude of voltage on all branches of all subcircuits will be printed

.csv Am U (RL* (X1)) Am U (* (X2)) - the amplitude of voltage of the branches which names begin with **RL** in subcircuit X1 and voltage of all branches of subcircuit X2 will be printed.

The remark:

- For nodal potentials as value for **Tip** * character can be used only. That is all nodal potentials can be printed only.
- The template is applied to names of subcircuits only in the form of separate * character.

Command action **.csv** leads to print results of calculation in files:

W.txt – Values of frequencies

W<i>.csv – values of coordinates of branches **X, Y** and values of the calculated values in branches where **i** changes from 1 to the number corresponding to an amount of values of frequencies, defined in the job (and being in file **W.txt**).

1.1.4. The job of calculation of transient process

1.1.4.1. Command of the job of calculation of transient process

Examples:

.tran 1u 0.01u – an estimated time 1mks, a rated step 0.01mks

.TRAN 1m 1u 0.01m – calculation will be fulfilled for process by duration 1ms, with rated step 1μs, the information will be printed everyone 0.01ms.

.tran 1m 1u udc – calculation will be fulfilled for process by duration 1ms, with rated step 1mks, on the schedule the information too through everyone 1mks will be printed, during the initial moment of time calculation on a direct current (**udc** - use direct curant) will be fulfilled.

The general determination:

.TRAN <time> <dt> [dtt [udc]]

time – duration of process

dt – calculation step

dtt – derivation step of the rated information

udc – (use direct curant) during the initial moment of time calculation on a direct current will be fulfilled. If the information derivation step **dtt** the information will be printed on each rated step **dt** is not specified.

1.1.4.2. Command of the job of a method of calculation

The general determination:

.METHOD <TRAP> | <EULER>

Where

TRAP – a method of trapezes.

EULER – Euler's method of the first order.

The method of trapezes is by default accepted.

1.1.4.3. Job for output

Example:

.print Am U (Rvny (X1)) - the amplitude of voltage on resistance **Rvny** of subcircuit **X1** will be printed.

The general description:

.print <Vid> <Tip₁> <Br₁> <X₁> [<Vid> <Tip₂> <Br₂> <X₂> [<Vid> <Tip₃> <Br₃> <X₃> [...]]]

After a keyword quadruples of words of the following content are specified:

<Vid> - value form. In the present version type **Am** – amplitude is admissible only.

<Tip> - value type. Following words are admissible:

U – voltage on nodes of the generalized branch ($U=V-e$, where **V** voltage on an internal branch, **e** - a source of voltage of the generalized branch)

I – a branch current

P – power of a branch ($P=U*I$)

**
** – a branch name. To learn names of the branches entering into a subcircuit it is possible only in its description. Names of branches of the communications connecting subcircuits, are resulted in the netlist of the circuit and installed at circuit creation.

<X> - a subcircuit name. All names of subcircuits should begin with the character **X** and contain no more than 8 characters. These names are resulted in the netlist of the circuit and installed at circuit creation. For branches of communications instead of a subcircuit name the character "-" (minus) should be specified

As a separator between words the gap is used. For convenience of perception as a separator brackets can be used.

Command action **.print** leads to print results of calculation in files:

T.txt – Values of time

Result<i>.txt – values of the calculated values where **i** changes from 1 to the number corresponding to an amount of schedules, defined in the job.

1.1.4.4. Fast displaying of results of calculations

If to set a command

.QPLOT [img]

After of terminations of calculation, results will be displayed in the form of the graphs sequentially displayed in windows. The amount of windows corresponds to number of the graphs defined in a command **.print**

img – an option, allowing to output images in graphic files. The option can accept following values: **gif**, **png**, **jpg**

Commands:

.leg <text1> <text2> ... <text i> or

.LEG <text1> <text2> ... <text i>

allow to enter inscriptions at the top of graphics. Text fields should correspond strictly to the variables set in a command **.print**. In a text field there should not be blanks. If the blank nevertheless is necessary in an inscription in a text field the underlining sign.

Example

.print Am U(CJ1(X2)) Am U(CJ1(X4))

.leg w_rad/s_for_2faz w_rad/s_for_3faz

If the command **.LEG** is not specified or on a corresponding place in this command instead of a text field the sign a minus as an inscription the text set in a command **.print** is used

1.1.4.5. Displaying of graphics during calculation

The command

.FLOW [(t_i t_j ... t_k)] [(t_i t_m ... t_n)] [(x_i x_j)] [(x_m x_n)] [**B[UFSIZE]=N**] [**D[ELAY]=M**] [**C[OLORS]=B[LACK]**]

The command **flow** is used together with a command **.print** and leads to that graphics are displayed during calculation. In parentheses it is possible to group graphics with instructions of their numbers. Number should correspond to number of a position of the graphic in a command **.print**. At grouping before number it is necessary to deliver either character **T** or character **X**. If character **T** specified, then graphics will be deduced in function from time. If character **X** specified, then graphics should be grouped on two, time is used as parameter, and graphics will be deduced as function of dependence of second number from the first number.

Total number of graphics - is no more 10.

Additional keys:

B[UFSIZE] =N - number **N** specifies in an amount of the points displayed in a window (all points are by default displayed),

D[ELAY] =M - number **M** specifies in milliseconds in a time lag at display of graphics (by default - without a time delay)

[C[OLORS]] =B[LACK] - the job of a black background for graphics (by default a background white)

The command without keys leads to displaying of all graphics in one window

Example:

.print Am I(R1(X1)) Am I(R2(X1)) Am I(R3(X1)) Am I(R4(X1)) Am I(R5(X1)) Am I(R6(X1))

.flow (t2 t5 t1) (t4 t1) (x1 x3) b=1000 d=2 c=b

Here the command **print** sets an output of graphics of a current in resistors with 1st on 6th.

The command **flow** sets displaying of three groups of graphics. As function of time two groups: 2nd 5th and 1st; 4th and 1st. One more graphic will be displayed in the form of dependence of a current in resistor R3 from a current in resistor R1. The size of the buffer - 1000 points, a time delay - 2msec, a background - is black.

Note: for displaying of graphics on command **flow** is used the utility **q2d** (<http://sourceforge.net/projects/q2d/>) .

1.1.4.6. The job for output in a format csv

The job for an output in a format **csv** is installed also as well as for the frequency analysis. Except for that, we admit only one type **Am** – amplitude

As a result of command action **.csv** files are formed:

T.txt – Values of time

T<i>.csv – values of coordinates **X, Y** and values of the values calculated for given branches, où **i** changes from 1 to the number corresponding to an amount of graphs (to each moment of time one corresponds **i**-й a file).

1.1.5. Modification of subcircuits

Parameters of branches of subcircuits are set at creation of the description of a subcircuit in a database. In the netlist parameters of subcircuits operate by default. It is supposed that in each branch of a subcircuit the source of voltage connected sequentially with a branch and a source of a current, connected parallelly with a branch operates. To alter subcircuit parameters, this subcircuit should be identified in the netlist. Two modifiers are for this purpose used: for modification of nominal's the modifier name should begin with **Z**, for modification of sources the modifier name should begin with **S**. After that, the subcircuits marked thus, it is possible to change a command .mod. We consider in more details record of modifiers.

Modifiers are allocated in the netlist in a line end with the subcircuit description.

Example:

Let the subcircuit with name **X1** is connected to nodes of circuit **N-01**, **N-02** and **N-03**. Subcircuit type – **OY1**.

Then in the netlist there can be such record:

X1 N-01 N-02 N-03 ? ? ? OY1

Here question signs mark not connected nodes of a subcircuit. The typical name defines structure of a subcircuit and its parameters in a database. Now, if it is necessary to alter these parameters on sources and nominal's it is necessary to enter modifiers in the list connections, for example:

X1 N-01 N-02 N-03 ? ? ? OY1 S1 Z1

After that it is possible to enter a command .mod which has two formats: for changes of sources and changes of nominal's.

1.1.5.1. Change of nominal's of subcircuits by a command .mod

.mod <Zmod> <Base> <Br₁> = <Val₁> [<Br₂> = <Val₂> [<Br₃> = <Val₃> [...]]]

Zmod - a modifier name

Base - standard, or a typical name of a subcircuit with which it is present at a database. The typical name of a subcircuit is underlined in a sink of the description of a subcircuit right after listings of numbers of nodes.

Br - branch name of a subcircuit. The name can be set in the form of a template if to specify * character. In this case all branches names which begin with the characters facing * will be modified. For example, **R*** means that nominals of all branches of the subcircuit which names begin with **R** will be changed. If a branch – a controlled source for change of transmission ratio of a source it is necessary to add **K** character to a name of a branch. For example, if the controlled source of voltage is implemented on a branch with name **E1** for transmission ratio change, we admit on number 10000, it is necessary to enter **E1k=10000**

Val - new value of a nominal. Value of a nominal can be resulted in three types:

1) in the form of number. In this case it is possible to specify a scale factor. The number can be only real.

2) in the form of numerical expression. Expression should be quoted (inverted commas can be unary ' or doubled " or `). Scale factors are not admissible, but it is possible to specify an engineering format of numbers.

In expression it is possible to use following designations for constants:

PI = 3.1415926535897932384626433832795;

E = 0.27182818284590451e+001;

MU0 = 0.12566370964050292e-005;

E0 = 0.88541878176203908e-011.

At the frequency analysis the constant imaginary unit $I = \sqrt{-1}$ is in addition used

In numerical expression following mathematical operations (are admissible at calculations in temporal area):

+, -, *, /, ^, exp, lg, ln, sqrt, sh, ch, th, cth, sin, cos, tg, ctg, asin, acos, atg, abs

3) in the form of the link to functions. In this case after equal sign the function name is underlined>. Rules of the description of functions are resulted further.

Example:

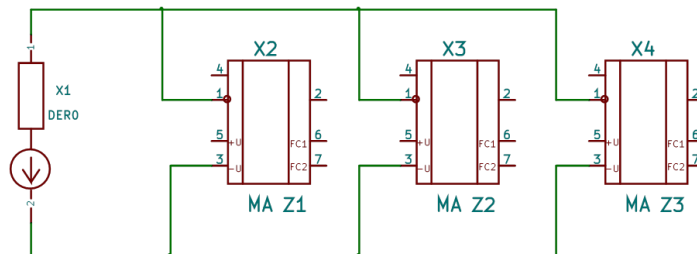
.mod Z1 sch R2=1k RL* = 2k2 R = "(2e1+i*3.0) * (exp (-i*5)) ^4" L=y1 - this record means that for subcircuits with a typical name **sch** and circuit modifier **Z1** following parameters will be changed:

Resistance **R2**, becomes equal 1kOhm

All resistance which begin on **RL**, become equal 2,2 kOhm

Resistor **R** will be presented by the complex resistance which value is calculated according to expression: **(2e1+i*3.0) * (exp (-i*5)) ^4**

Inductance **L** will be calculated under the formula set by function **y1**.



```
+gnucap .tran 0.5 0.5m 5m
.print Am U(E1(X2)) Am U(E1(X3)) Am U(E1(X4))
.qplot
.mod Z1 ma c1=0.4u R1=400k
.mod Z2 ma c1=0.2u R1=200k
.mod Z3 ma c1=0.1u R1=100k
```

1.1.5.2. Change of nominal's of subcircuits by a command .param

.param <Zmod> <Base> <Sym₁> = <Val₁> [<Sym₂> = <Val₂> [<Sym₃> = <Val₃> [...]]

The format of a line is similar to a command format **.mod** except that instead of a branch name, the symbolical reference to number is used. By preparation of subcircuits, instead of numerical value of face value, the symbolical reference is used only. Numerical face value receives values after command processing **.param**

The symbolical reference cannot look like a template, as in a command **.mod**

In more details parameterization of subcircuits is considered in the section devoted to preparation of subcircuits

1.1.5.3. Change of sources of subcircuits by a command .mod

.mod <Smod> <Base> <Tip₁> <Br₁> = <Val₁> [<Tip₂> <Br₂> = <Val₂> [<Tip₁> <Br₃> = <Val₃> [...]]

The format of change of sources is similar to a format [of change of nominals](#) except that the modifier name should begin with **S**, and to names of branches the source type is added. As a separator gaps and parentheses are used. Following designations of types of sources are admissible:

I – for a source of a current and **V** – for a voltage source.

Example:

.mod S1 sch V (R2) =5 I (RL *) = 2m2 V (R) = "(2e1+i*3.0) * (exp (-i*5)) ^4" I (L) =y1

- this record means that for subcircuits with a typical name **sch** and circuit modifier **S1** following parameters will be changed:

The source of voltage operating sequentially with **R2**, becomes equal 5 Volt;

The sources of a current operating on all resistance which begin on **RL**, become equal 2.2 mA

The source of voltage operating sequentially with **R** will be presented by the complex number which value is calculated according to expression: $(2e1+i*3.0) * (exp (-i*5)) ^4$

The current sources, operating parallel inductances **L** it will be calculated under the formula set by function **y1**.

1.1.5.4. The description of functions

Functions are set in two lines. In the first line the function name and its content is set, in the second line arguments for this function are enumerated.

The line with function is set by a command **.fun** The line with arguments set by a command **.arg**

Let's result rules of design of these commands.

The description of function by a command .fun

Example:

.fun y1 = exp (- ((V - 465.0) / 100.0) ^face)

The function name should not exceed 8 characters. After equal sign function is entered.

Into function composition enters: numbers, arguments and mathematical operations.

Numbers can be only real, scale factors are not admissible, but the engineering type of number designation is possible.

Arguments are represented by alphabetic identifiers which should not coincide with designations of mathematical operations.

Mathematical operations are admissible the following:

+, -, *, /, ^, exp, lg, ln, sqrt, sh, ch, th, cth, sin, cos, tg, ctg, asin, acos, atg, abs, re, im

At calculation in temporal area mathematical operations **re**, **im** are not used. In temporal area functions can be used: **impls** et **pos**.

IMPLS – accepts value **1** at the positive values of argument, at negative - **0**.

POS – accepts the positive values of argument, at the negative values of argument function value is equal **0**.

The job of arguments a command .arg

The argument name should not exceed 8 characters. Names of arguments should not coincide with names of the resulted mathematical operations. For example: argument **Tgs** will be interpreted as erratically gated in expression **tg (s)**.

Numbers, numerical expressions, links to parameters of branches, sources of branches or on voltage and currents of branches, links to angular frequency and links to complex angular frequency can be values of arguments.

Rules of the description of numbers and numerical expressions have been already considered in section "[Change of nominals of subcircuits by a command.mod](#)"

Format for arguments the following:

arg₁ = <Tip₁> <Br₁> <X₁> arg₂ = <Tip₂> <Br₂> <X₂>> arg₃ = <Val> ...

Here

arg –argument name.

Tip – link type. The following references are admissible:

U – voltage of a branch;

I – a branch current;

V – a source of voltage of a branch;
J – a source of a current of a branch;
Z – a branch nominal;
jW – complex angular frequency;
W – real angular frequency;

Br – a branch name. As a branch name there can be only a name of a branch of a subcircuit.

X – a subcircuit name.

Val number or numerical expression. For numbers and numerical expressions the rules resulted in section

"[Change of nominals of subcircuits by a command.mod](#)" are used

As separators of words gaps or brackets are used.

Format of links to frequencies the following:

$$\mathbf{arg}_1 = jW \quad \mathbf{arg}_2 = W$$

Here:

arg – an argument name.

jW a designation for current value of complex angular frequency $\omega = j \cdot 2\pi f$

W a designation for current real value of angular frequency $\omega = 2\pi f$

That is the argument during calculation will accept current values of angular frequency either in the real form or in the complex form depending on the gated in designation.

Examples

.arg V=U (Rvny (X2)) face = ' 4'

.arg V=Z (Rvny (X2)) face = 4

.arg a=V (Rvny (X1)) b=jw c=w d = "(2e1+i*3.0) * (exp (-i*5)) ^4"

Example of usage of function

.fun F = (1 + s*L1/R1) / (1 + s* T)

.arg s=jw L1=Z (L (X1)) R1=Z (R (X1)) T=1m

.mod S1 DER0 V (Rvny) = F

Example of usage of function in temporal area:

.fun F = A*sin (w*t)

.arg A=10 w = ' 2*pi*50 ' t=T

.mod S1 DER0 V (Rvny) = F

In the resulted examples of value of a source of voltage on branch **Rvny** in subcircuit **DER0** with modifier **S1**, it will be defined by function **F**. Arguments of function are defined through parameters of other subcircuits. Thus, not electric communication between subcircuits is carried out.

1.2. The description of subcircuits

1.2.1. Composition of files.

Descriptions of subcircuits contain in different files which can be allocated or in a working directory, or in the general directory of a database. At calculation performance at first the subcircuit description is searched in a working folder, then in the general database.

Names of files correspond to names of subcircuits. For each subcircuit files with the same name, but with following extensions prepare:

- .cir** – the text file with the netlist
- .sun** – a binary file with the description of a rated subcircuit
- .cep** – the text file with the description of a rated subcircuit
- .pdf** – the information description of composition of a subcircuit in Adobe Reader format
- .sch** – graphic representation of internal composition of a subcircuit

For graphic creation of the circuits consisting of subcircuits, libraries of exterior images of subcircuits in the form of multipoles are used. The graphic information is built in each library about a subcircuit. Thus, files are added still:

- .lib** – library graphic representation of a subcircuit in the form of a multipole
- .dcm** – the text file with the description of the library documentation about a subcircuit

During calculation the program at first tries to open a binary format if the file with such extension is not present the file with the text description is searched. If in the netlist with a typical name of a subcircuit to connect characters "-T" the text format will be read out only.

For example:

X1 n1 n2 DER0-T Z1 S1

The information on subcircuit **X1** will be read from text file **DER0.CEP**.

If to write down:

X1 n1 n2 DER0 Z1 S1

Then the information on a subcircuit will be read from binary file **DER0.SUN**. If such file is not present, attempt to read text file **DER0.CEP** will be made

The information description of composition of a subcircuit in a format **pdf** (Adobe Reader) can be used as independently, and for the built in description in graphic library of exterior images of subcircuits **lib**.

The graphic file **sch** is necessary only for display of results of calculation of the big circuits in 3D a format. In the present version this file prepares in circuit editor KiCAD.

1.2.2. The netlist for subcircuits

The netlist is saved in a file with **cir** extension and with a subcircuit name. This name then will be used for a call of the description of a subcircuit at circuit calculation.

The general agreements for the text description are similar to what have been considered for the [description of circuits](#).

The netlist subcircuits includes only two-poles. The netlist consists of separate text lines. At first the branch name is allocated. The first character of a name defines branch type. Following types (characters) are admissible:

1). The first character of a name defines branch type. Following types (characters) are admissible:

- R** - resistance;
- L** - inductance;
- C** - capacity;
- V** - an independent source of voltage;
- I** - an independent source of a current;
- E** - the source of voltage controlled voltage;
- F** - the source of a current controlled a current;
- G** - the source of a current controlled voltage;

H - the source of voltage controlled a current;
M - mutual inductance.

2). After a branch name, is placed the name of 1st node, then a name of the second node. The positive direction of a current - from the first node to the second. Length of a name of a node - no more than eight characters.

Boundary nodes should be enumerated on a lock, since 1. A format of boundary nodes: /**i**, where **i** – node number.

3). In a subcircuit the zero node should be set. A format of the job of a zero node: /**0**
Important: the zero node should not be boundary. Resistance with null value is in case of need entered.

4). The number corresponding to a nominal of a branch Further is underlined.

5). After a branch nominal it is possible to specify (optionally) one or two additional parameters.

For a voltage source it is possible to set value of internal serial resistance. This number should be written down after a sign '='. It is possible not to specify a resistance name. Zero internal resistance is by default installed.

For a current source it is possible to set value of internal parallel resistance. This number should be written down after a sign '='. It is possible not to specify a resistance name. Internal resistance equal 10 GOhm (10^{10} Ohm) is by default installed.

*For **R**, **L**, a **C** of branches* it is possible to specify values of parallely switched on source of a current and value of sequentially switched on source of voltage. For **L** branches thus it is possible to set an initial current at calculation of transient phenomena. Value of a current should be specified with the negative sign. For a branch **C** it is possible to set initial voltage at calculation of transient phenomena. Value of voltage should be specified with the negative sign. In all cases value of a source of a current is set in a format **i** = <number>, or **I** = <number>. Value of a source of voltage is set in a format **v** = <number>, or **V** = <number>.

*For controlled sources (**E**, **F**, **G**, **H**)* a description format:

<Name> **N1** **N2** <Source> <K> [**R**]

Name – a name of a controlled source

N1 **N2** – initial and finite nodes

Source – a name of a controlling branch

K - transmission coefficient

R – internal resistance of a controlled branch (by default)

6). In the end of the netlist it is possible to specify the information on mutual inductances. The name of mutual inductance should begin with **M** or **m**. Further names of two inductive branches and in the end value of mutual inductance are specified.

Example of the netlist:

* The comment: boundary nodes n1 and n2

R1 **n1** **n2** **1.2k** **v=220** **i=1m**; here the same the comment

R2 **n2** **n1** **1m**

R0 **n2** /**0** ; "zero" resistance and a zero node

.end

1.2.3. The description of subcircuits

The description of a subcircuit intended for calculations, should be stored a file with **sun** extension. The text file with the netlist **cir** is auxiliary for formation the main files. For perusal of the description in the text editor the file **cep** – text analog of a binary file **sun** can be used.

Utilities are applied to obtaining of files **sun** and **cep** **cir2sun-cep** (or **cir2sun-cep.exe**) accordingly. For file obtaining **sun** from the information on subcircuits the appropriate utility, for example is launched:

./cir2sun-cep souschema.cir

Where **souschema.cir** - a file with the netlist. Files **souschema.sun** and **souschema.cep** will be as a result generated

The subcircuit description in a text format **cep** looks like:

```
DER0VAP NOEUD    2 BRAN    2 ELEM MX C        2 ELEM MX A        2
C+++   0 L+++   0 R+++   2 CN++   0 LN++   0 RN++   0 M+++   0
EI++   0 EU++   0 JI++   0 JU++   0 EIN+   0 JUN+   0 JIN+   0 JUN+   0 JUU+   0 JUUN   0 JII+   0 CORDS   0

      MATRIX C
|COL| LIN|PARTIE REELLE|PARTIE IMAGIN.||COL| LIN|PARTIE REELLE|PARTIE IMAGIN.|
  1|   1| 1.0000         | 0.0          |  2|   2| 1.0000         | 0.0          |
      MATRIX At
|COL| LIN|PARTIE REELLE|PARTIE IMAGIN.||COL| LIN|PARTIE REELLE|PARTIE IMAGIN.|
  1|   1| 1.0000         | 0.0          |  2|   2| 1.0000         | 0.0          |
R+++RVNY      1=0.0
R+++R0        2=0.0
LES SOURCES    1
E***          1=1
****
```

In the first line places the amount of pairs nodes, an amount of branches, an amount of elements of a contour matrix and a nodal matrixes. Further convergence on an amount of separate branches – capacities, inductances etc. are resulted.

Then values of nonzero elements of contour and nodal matrixes are located.

Here it is necessary to note that matrixes of conversions should be not mandatory topological – generally it is tensors of conversions for contravariant and covariant components of the equations. In the present version topological representation of subcircuits is used only.

Further there is a description of branches of the elementary circuit where the branch type, a branch name, number of a branch and a branch nominal is underlined. Controlled sources are described in two lines: in the first line own resistance of a controlled branch: type, a name, number and a branch nominal, in the second line – type, a name for transmission ratio, number of a controlling branch, value of transmission ratio.

Completes the description of branches the list of sources in which the source type, number of a branch and value (real number) is underlined.

1.2.4. Parameterization of subcircuits

Nominal's of branches can be parameterized:

- Identifiers
- Character expressions at which composition there are identifiers.

Values of identifiers at subcircuit creation in the circuit editor are set in a line beginning with a command **.param**. In the text file of value of identifiers are enumerated before the description of matrixes of conversions

Character expressions are made out the same as also functions. Arguments of functions in this case are identifiers.

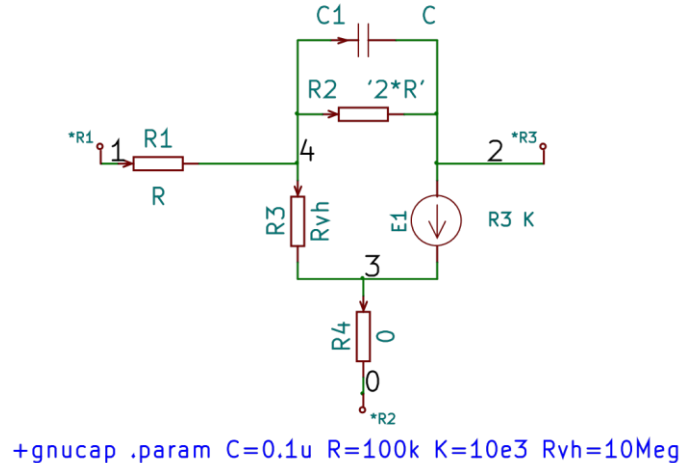
If at a subcircuit there are functions in arguments of functions the parameter identifier can be allocated in curly brackets.

Example

In the resulted subcircuit are parameterized:

- Resistance **R1** the nominal is set by identifier **R**;
- Resistance **R2** the nominal is set by expression '**2*R**';
- Capacity **C1** the nominal is set by the **C** identifier;
- Transmission **ratio** of a source of the voltage controlled voltage **E1** is set by the identifier **K**.
- Resistance **R3** the nominal is set by identifier **Rvh**.

In line **.param** to identifiers specific numerical values are appropriated



1.2.5. Input of functions in subcircuits

In whole, rules are similar to what it is used at input of functions in the circuit. Difference consists in a function name designation. Here the direct instructions on that element on which function is implemented are used.

Example:

```
.fun V(R1) = exp(-(V - 465.0) / 100.0)^face)
.arg V=U(R2) face=4
```

In this case the voltage source in branch **R1** is implemented. Except a voltage source, is admitted to implement in a branch a current source (the key character **I**) and non-linear resistance (key **Z** character). For example, **I (R1) =...**, **Z (R1) =...**

1.2.5.1. Input of hierarchical functions

Difficult formulas are entered by "parts". For example, for the previous example

```
.fun #y1= V - 465.0
.arg V=U(R2)

.fun V(R1)= exp(-(a/100.0)^face)
.arg a=#y1 face=4
```

Here it is preliminary calculated **#y1**, results are substituted in the subsequent formula. Names of auxiliary functions should begin with a symbol **#**.

It is possible to supply in addition a symbol **#** with a symbol **\$** or **%**. Then, if it is set **\$** auxiliary function will be used only for calculation of influences, if the symbol of **%** auxiliary function will be used only for calculation of resistance is set.

Example:

```
.fun #y1= V - 465.0
```

```
.arg V=U(R2)
```

```
.fun V(R1)= exp(-(a/100.0 )^face)
```

```
.arg a=#y1 face=4
```

2. Output data

2.1. Resultants files in a format txt (calculation in frequency area)

All results are allocated in text files. Names of files look like: **Result_i.txt**, where **i** - graph number. One more file - **W.txt** - contains frequencies on which calculation has been fulfilled.

Example.

Let for some circuit the following job for calculation has been made

.ac dec 5 0.1 1k

.print Ad U (E1 (X2))

It will be as a result generated two files: **W.txt** and **Result1.txt**. In **W.txt** values of frequencies contain, in values of amplitude of voltage in decibels on an output branch of a source of the voltage controlled voltage in subcircuit X2 **Result1.txt** contain. The content of files:

W.txt	Result1.txt
W	Ad U (E1
0.10000E+00	(X2))
0.15849	-0.17112E-01
0.25119	-0.42855E-01
0.39811	-0.10685
0.63096	-0.26357
1.0000	-0.63396
1.5849	-1.4451
2.5119	-2.9921
3.9811	-5.4294
6.3096	-8.6075
10.0000	-12.231
15.849	-16.072
25.119	-20.007
39.811	-23.981
63.096	-27.971
100.000	-31.966
158.49	-35.965
251.19	-39.964
398.11	-43.964
630.96	-47.964
1000.00	-51.964
	-55.964

All resultant files are allocated in one folder. For display of results it is possible to use any mathematical packet. For example, for **SciLab** it is possible to use a following template for drawing of graphs:

```
clear;
ngc = 1;//number of first file Result
ngf = 1;//number of last file Result
//further anything can to be changed, except "gln" depending on a graph type - logarithmic or not
w=read (' W.txt ', -1, 1, ' (a) '); sz=size (w); legW=w (1);
for i=2:sz (1), f (i-1) =msscanf (w (i), ' %g '); end
for k =ngc:ngf,
n=code2str (k); y=read (' Result ' +n + '.txt ', -1, 1, ' (a) '); leg (k) =y (1);
for i=2:sz (1), gr (k, i-1) =msscanf (y (i), ' %g '); end;
scf (k);
xset ("font", 2, 4); a=get ("current_axes"); a.title.font_size=3; a.x_label.font_size = 3;//set (a, "sub_tics", [4 4]);
plot2d1 ("gln", f, gr (k :));//, leg=leg (k)
xtitle (leg (k) "", ""); legends ([' w, Hz '], [-100], opt = "lr", with_box = % f);
xgrid (2);
end;
```

2.2. Resultant files in a format csv (calculation in frequency area)

All results are allocated in text files. Names of files look like: **W_i.csv**, where **i** – number corresponding to serial number of frequency in the job for calculation. One more file - **W.txt** contains frequencies on which calculation has been fulfilled.

For example, if in the job for calculation there is a command for an output of amplitude of voltage to all capacity branches of all subcircuits: **.csv Am U (C*(*))**

As a result of calculation text files about such contents will be generated:

X, Y, Z

```
493.6781616, 135.7314758, 0.055271942168
495.9978027, 135.7314758, 0.055344615132
495.9978027, 138.0511169, 0.054795630276
493.6781616, 138.0511169, 0.054709360003
498.8973389, 138.0511169, 0.054795630276
501.2169800, 138.0511169, 0.054844532162
501.2169800, 135.7314758, 0.055383961648
498.8973389, 135.7314758, 0.055344615132
504.1165161, 135.7314758, 0.055383961648
506.4361572, 135.7314758, 0.055391218513
506.4361572, 138.0511169, 0.054856795818
```

.....

Since second line, in the first and second columns – X and Y branch coordinates, in the third column – value of amplitude of voltage.

The number of files **W_i.csv** will be equal to number **n** the frequencies defined in a command **.AC**:
W1.csv, W2.csv, ... Wn.csv

2.3. Resultant files in a format txt (calculation in temporal area)

Results are allocated in text files. Names of files look like: **Result_i.txt**, where **i** – graph number. One more file - **T.txt** – contains values of time.

Example.

The job for calculation

.tran 0.05 0.001 0.0025

.print Am U (E1 (X2))

.qplot

It will be as a result generated two files: **T.txt** and **Result1.txt**. In **T.txt** values of time contain. In values of amplitudes of voltage on an output branch of a source of the voltage controlled voltage of subcircuit **X2** **Result1.txt** contain.

If **.qplot** misses, one file with two columns is formed: in one values of time T in another corresponding values of voltage.

Composition of files:

T.txt	Result1.txt	Result1.txt (if .qplot misses)	
T	Am U (E1 (X2))	# T	Am U (E1 (X2))
0.30000E-02	0.12873	0.30000E-02	0.12873
0.60000E-02	0.40393	0.60000E-02	0.40393
0.90000E-02	0.59124	0.90000E-02	0.59124
0.12000E-01	0.52900	0.12000E-01	0.52900
0.15000E-01	0.26148	0.15000E-01	0.26148
0.18000E-01	0.24266E-02	0.18000E-01	0.24266E-02
0.21000E-01	-0.41225E-01	0.21000E-01	-0.41225E-01
0.24000E-01	0.16009	0.24000E-01	0.16009
0.27000E-01	0.43417	0.27000E-01	0.43417
0.30000E-01	0.54900	0.30000E-01	0.54900
0.33000E-01	0.40404	0.33000E-01	0.40404
0.36000E-01	0.11310	0.36000E-01	0.11310
0.39000E-01	-0.89489E-01	0.39000E-01	-0.89489E-01
0.42000E-01	-0.42076E-01	0.42000E-01	-0.42076E-01
0.45000E-01	0.21104	0.45000E-01	0.21104
0.48000E-01	0.45614	0.48000E-01	0.45614

Note: In the version for parallel calculations the command **.qplot** is not provided yet

All resultant files are allocated in one folder. For visualization of results any mathematical packet can be used. For example, for **SciLab** it is possible to use the following script:

```
clear;
ngc = 1;//number of first file Result
ngf = 1;//number of last file Result
w=read (' T.txt ', -1, 1, ' (a) '); sz=size (w); legW=w (1);
for i=2:sz (1), f (i-1) =msscanf (w (i), ' %g '); end
for k =ngc:ngf,
n=code2str (k); y=read (' Result ' +n + '.txt ', -1, 1, ' (a) '); leg (k) =y (1);
for i=2:sz (1), gr (k, i-1) =msscanf (y (i), ' %g '); end;
scf (k);
xset ("font", 2,4); a=get ("current_axes"); a.title.font_size=3; a.x_label.font_size = 3;//set (a, "sub_tics", [4 4]);

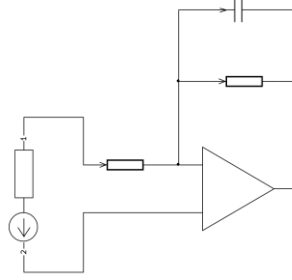
plot2d1 ("gnn", f, gr (k :));//, leg=leg (k)
xtitle (leg (k) ","); legends ([' t, sec '], [-100], opt = "lr", with_box = % f);
xgrid (2);
end;
```

The **note:** For parallel calculations there is a version «**whitout-dislin**» where the command. **qplot** is not provided.

3. Example

3.1. Filter

Let's consider as an example calculation of performances of the filter of 1st order



For calculations it is required to prepare:

- 1- Database of subcircuits and
- 2- The circuit consisting of prepared subcircuits.

Into a database enter: a subcircuit of the operational amplifier, a subcircuit with capacity and a subcircuit of an input source of voltage. Then the rated circuit of the filter will include two subcircuits: a source and the amplifier and communication branches: two resistors and capacity. For calculations in the frequency area it is required two subcircuits (the source, the amplifier) and three branches of communication – two resistance and capacity.

3.1.1. Preparation of a subcircuit of the amplifier

3.1.1.1. Preparation of the netlist

```
RIN /1 /3 10MEG
E1 /2 /3 RIN 100k 1
R0 /0 /3 0
.end
```

Branches of a source of the voltage controlled voltage enter into a subcircuit:

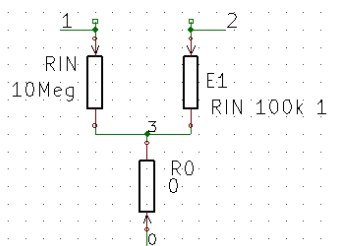
The controlling branch **RIN**, controlled branch **E1**

And as a branch of "grounding" **R0**

The netlist is saved in file **OPER.cir**

3.1.1.2. Documentation preparation

The netlist corresponds to graphical representation:



The remark: the calculation program defines branch type on the first letter of her name, instead of under its image. In this case the source of pressure controlled pressure **E1** is represented in the form of the resistor with appropriate attributes.

This graphical representation is saved in a file **OPER.pdf**

3.1.1.3. Preparation of a file with rated determination of a subcircuit in a binary format

The utility is launched

```
./cir2sun-cep OPER.cir
or for Windows:
cir2sun-cep.exe OPER.cir
```

As a result files with rated determination of subcircuit **OPER.sun** and **OPER.cep** are automatically formed

3.1.2. Preparation of a subcircuit of a source of voltage

3.1.2.1. Preparation of the netlist

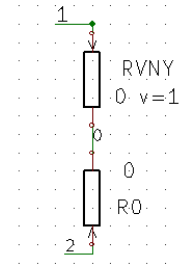
```
RVNY /1 /0 0 v=1  
R0 /2 /0 0  
.end
```

Resistor **RVNY** enters into a subcircuit with sequentially switched on source of voltage 1V
And a branch of "grounding" **R0**.

The netlist is saved in file **DER0.cir**

3.1.2.2. Documentation preparation

The netlist corresponds to graphical representation:



This graphical representation is saved in a file **DER0.pdf**

3.1.2.3. Preparation of a file with rated determination of a subcircuit in a binary format

The utility is launched

```
./cir2sun-cep DER0.cir
```

As a result files with rated determination of a subcircuit are automatically formed: **DER0.sun** and **DER0.cep**

3.1.3. Preparation of a subcircuit with capacity (for calculations in temporal area)

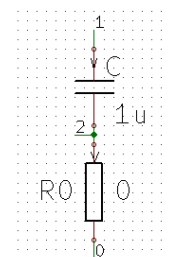
3.1.1 Preparation of the netlist

```
C /1 /2 1u  
R0 /2 /0 0  
.end
```

The netlist is saved in file **CAP.cir**

3.1.2 Documentation preparation

The netlist corresponds to following graphical representation:



This image is saved in a file **CAP.pdf**

3.1.3 Preparation of rated files in a binary and text format

The utility is launched

```
./cir2sun-cep CAP.cir
```

Files with the rated description of a subcircuit are as a result formed: **CAP.sun** et **CAP.cep**

3.1.4 Database formation

The received files with the description of subcircuits are allocated in one folder, for example **/home/user/base**

Minimum composition of files: **OPER.sun**, **DER0.sun** and **CAP.sun**

It is possible to add as:

- Files with the documentation: **OPER.pdf**, **DER0.pdf**, **CAP.pdf**
- Files with lists of connections: **OPER.cir**, **DER0.cir**, **CAP.cir**
- The utility **cir2sun-cep**

3.1.5 Preparation of a file with the rated description of the circuit

3.1.5.1. Preparation of the netlist of the rated circuit of the filter for calculation in the frequency area

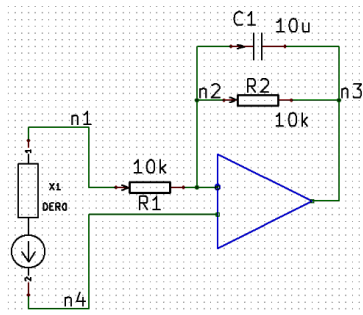
```
R2 n2 n3 10k
C1 n2 n3 10u
R1 n1 n2 10k
X1 n1 n4 DER0
X2 n2 n3 n4 OPER

.AC dec 10 0.01 1k
.print Ad U (E1 (X2))

.end
```

The file with the netlist is saved under name **FiltrW.cir**

The netlist corresponds to graphical representation of the rated circuit of the filter:



3.1.5.2. Preparation of the netlist of the rated circuit of the filter for calculation in temporal area

```
X1 n1 n4 DER0 S0
R1 n1 n2 10k
R2 n2 n3 10k
XC1 n2 n3 CAP Z1
X2 n2 n3 n4 OPER

.fun y=A*sin(w*t)
.arg A=10.0 w='2*pi*50' t=T

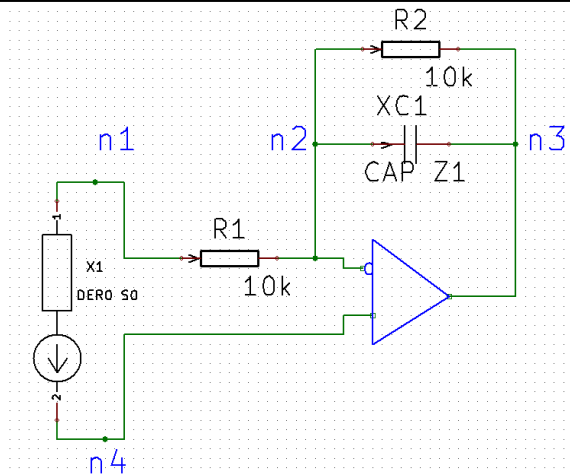
.mod S0 DER0 V (RVNY) =y
.mod Z1 CAP C=10u

.tran 0.05 0.0001 0.001

.print Am U (E1 (X2))

.end
```

The list of connections corresponds to the following image:



The file with the netlist is saved under name **FiltrT.cir**

3.1.6. Calculation performance

For calculation performance in Linux it is possible to use system **PBS** or its free analog **TORQUE** ([http:// www.adaptivecomputing.com/resources/downloads/torque/](http://www.adaptivecomputing.com/resources/downloads/torque/))

In these systems for calculations it is possible to prepare two files: ***paket** ***myjob** which should be allocated in a working folder.

The file ***paket** is necessary for start of jobs. The content of this file the following:

```
#!/bin/sh
m = `pwd`
qsub -l select=1:ncpus=1:mem=4gb -v p=$m myjob
```

The file ***myjob** contains the job for performance of programs. The content of this file:

```
#!/bin/sh
#PBS-N myjobX
#PBS-j oe
#PBS-l walltime=0:00:30
export FOR_COARRAY_NUM_IMAGES=2
source /opt/intel/composerxe-2011.0.084/bin/compilervars.sh intel64
/home/user/diel-cm -b/home/user/base/ -f${p}/FiltrW.cir
```

For calculations it is necessary to launch a file **paket**. For this purpose in a command line to print: **./paket** and to push Enter.

* As a result of calculation, it will be generated two files: one with frequency, another – with results of calculation: **W.txt**, **Result1.txt**. It is possible to look at contents of files in text editors.

For calculations in temporal area it is necessary to change a line

```
export FOR_COARRAY_NUM_IMAGES=3
```

And to enter a new name of an executed file for calculation in temporal area and a new name of a file with the netlist of the circuit:

```
/home/user/diel-rm-b/home/user/base/-f $ {p}/FiltrT.cir
```

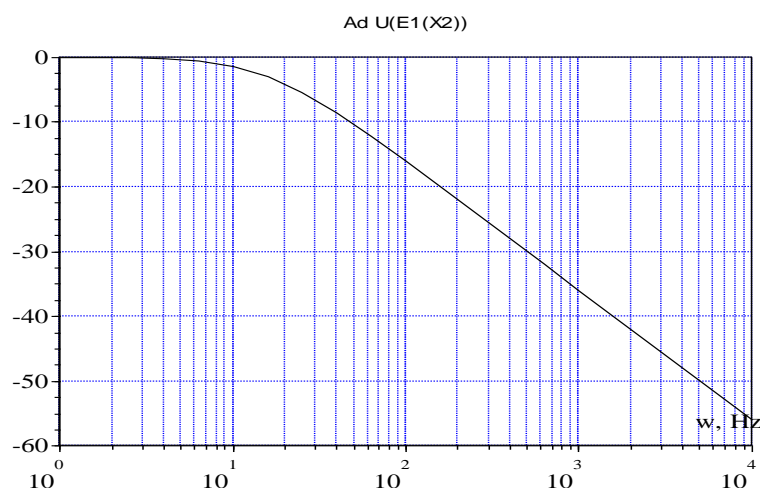
More simple variant of start on calculation:

```
#!/bin/sh
export FOR_COARRAY_NUM_IMAGES=2
source/opt/intel/composerxe-2011.0.084/bin/compilervars.sh ia32
p = `pwd`
/home/user/diel-cm -b/home/user/base/ -f${p}/FiltrW.cir
```

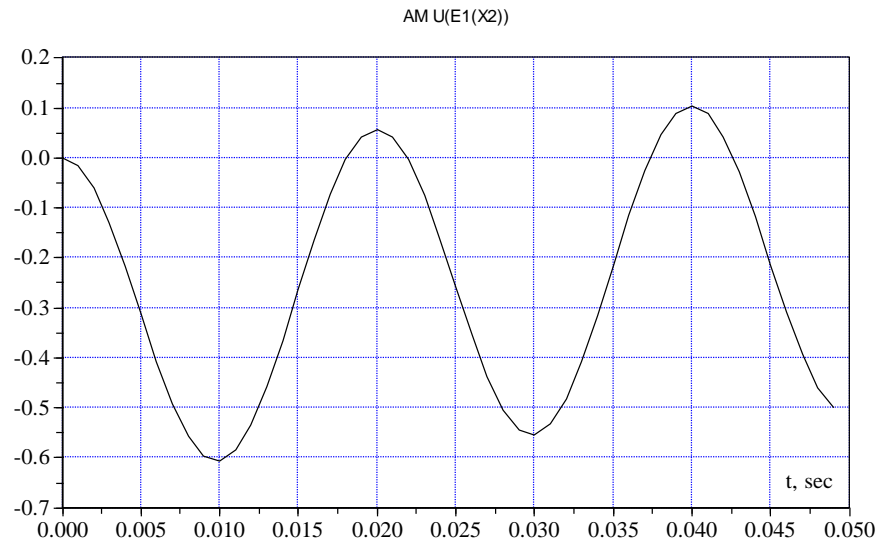
* After calculation end in temporal area two text files will be generated: one, the containing moments of time, another with results of calculation: **T.txt**, **Result1.txt**.

3.1.7. Displaying of results

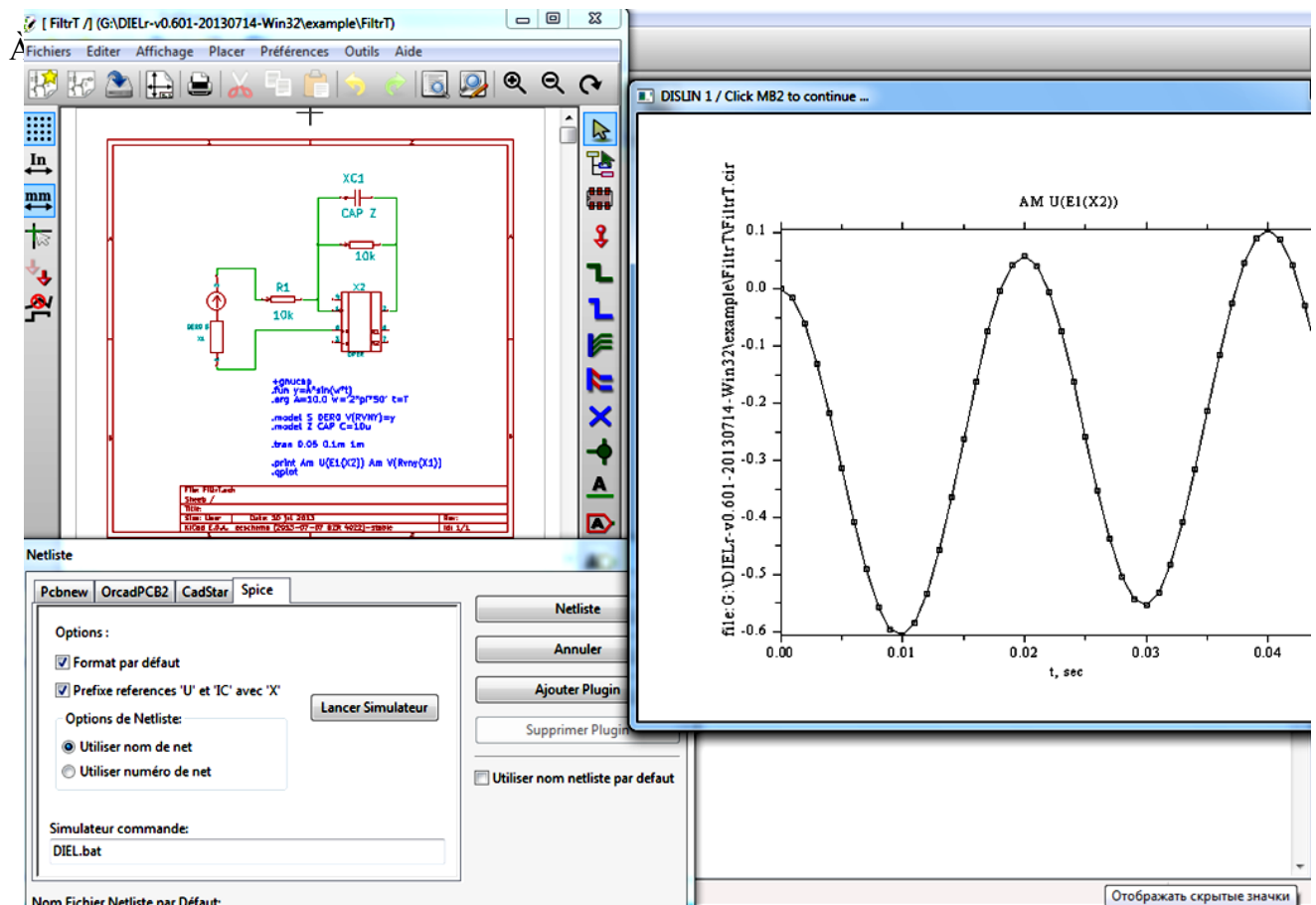
For displaying of results of calculation to graphs it is possible to use packet SciLAB and a script presented in section ["resultant files in a format txt"](#)



For displaying of results in temporal area it is possible to use packet SciLAB "[Resultant files in a format txt \(calculation in temporal area\)](#)"



If to use KiCAD:



4. Preparation of graphical representation of subcircuits

Application of graphic circuit editors, with the built in generator of the netlist, allows to automate preparation of the input and output data for calculation of the big circuits. Besides, the information on geometrical coordinates of layout of elements allows to display simultaneously results of calculation of currents and voltage of branches in 3D a format.

Here it is considered singularities of application of the editor of circuits of system **KiCAD**. Operation rules in the circuit editor are not resulted – they can be found in a manual on system.

Graphical representation of subcircuits prepares at two levels: at internal level of two-poles and exterior level of multipoles:

- For internal level there is a library of graphical representations of two-poles from which then the subcircuit gathers;
- For exterior representation of a subcircuit the library image of a subcircuit in the form of a multipole prepares.

4.1. Graphical representation of internal composition of subcircuits

4.1.1. Preparation of graphic library Two-poles

The library of two-poles is a text file with lib extension which contains the information for graphic display of elements in the circuit editor.

Here composition of the elementary lib file which contains the description of two-poles RLC

EESchema-LIBRARY Version 2.3 Date: 11/21/2012 OF 2:02:15 PM

#encoding utf-8

#

R

#

DEF R R 0 0 N Y 1 F N

F0 "R"-120 0 100 V V C CNN

F1 "R" 150 0 100 V V C CNN

DRAW

S-40 150 40-150 0 1 12 N

P 3 0 1 0 22 212 1 160-19 212 N

X ~ 1 0 250 100 D 60 60 1 1 P

X ~ 2 0-250 100 U 60 60 1 1 P

ENDDRAW

ENDDEF

#

L

#

DEF L L 0 40 N N 1 F N

F0 "L" 200 200 100 H V C CNN

F1 "L" 150-50 100 H V C CNN

DRAW

A 0-150 50-889 889 0 1 0 N 1-199 1-100

A 0-49 51-889 889 0 1 0 N 1-99 1 2

A 0 51 51-889 889 0 1 0 N 1 1 1 102

A 0 148 48-889 889 0 1 0 N 1 101 1 196

P 3 0 1 0 30 281-1 219-29 277 N

X 1 1 0 300 100 D 70 70 1 1 P

X 2 2 0-300 100 U 70 70 1 1 P

ENDDRAW

ENDDEF

#

C

#

DEF A C OF A C 0 10 N Y 1 F N

F0 "C" 50 100 100 H V L CNN

F1 "C" 100-150 100 H V L CNN

DRAW

P 2 0 1 8-100-30 100-30 N

P 2 0 1 8-100 30 100 30 N

P 3 0 1 0-19 153-1 99 14 153 N

X ~ 1 0 200 170 D 40 40 1 1 P

X ~ 2 0-200 170 U 40 40 1 1 P

ENDDRAW

ENDDEF

#

#End Library

The text file is automatically generated by operation with library in the environment of **KiCAD**.

Let's mark that the two-pole description begins with title:

```
#  
#R  
#
```

.....

Also comes to an end with line

ENDDEF

From two-poles it is possible to collect a demanded subcircuit in the circuit editor. The element type can be changed if to appropriate other name according to rules именования two-poles and adding necessary components in text attributes. It can be made immediately in the text editor, processing a file **lib**.

4.1.2. The assembly of a subcircuit from library elements

The prepared library is used for the assembly of subcircuits. The assembly order of subcircuits has some singularities connected to operation of the editor. We result the main stages and the singularities of information input connected to them.

1) Setting of branches on a working field, input of attributes of branches, connection of branches – is performed on rules of operation with the circuit editor.

2) Numbering of boundary nodes. It is fulfilled by a command именования circuits.

3) Correction of "trailing" branches. Some boundary nodes can be not connected to a remaining part of a subcircuit. At generation of the netlist, KiCAD does not include names of such nodes in the list. One of methods of a solution of a problem - input of the additional resistance, which name begins with * character. One node of such branch incorporates to a boundary node. The second node remains to free. Now the node name appears in the netlist. The additional branch will be perceived, as the comment and will be passed by the rated program.

4) Input of a zero node. The zero node is entered by a command именования circuits. To zero harm the name **0** should be appropriated. In the netlist there will be/0 characters. By subcircuit development it is necessary to consider that the zero node should not be boundary is a condition is dictated by an applied method of calculation in parts. If needed it is necessary to enter additional resistance with null value.

5) Assignment of names to branches. A singularity of the circuit editor: he demands, that the branch name came to an end with digit. In certain cases it can be excessive. To remove digit it is possible only in the text of the generated netlist.

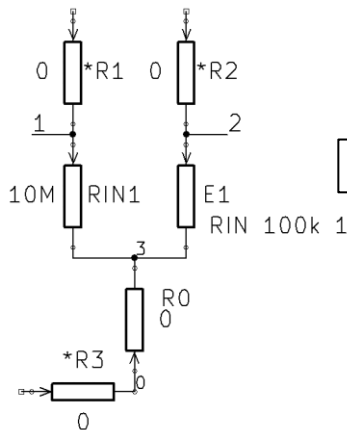
6) Determination of local coordinates of a subcircuit and its sizes.

At first units of measurements – millimeters are installed. Then the mouse is installed in the left upper corner of a subcircuit and in bottom line coordinates of position of the mouse are defined: X_0 Y_0 . Then the mouse is installed in the right lower corner of a subcircuit and coordinates are defined: X_{max} Y_{max} . The received values then will be used for scaling of coordinates of branches in the circuit where the given subcircuit enters.

To this stage it is required to return after creation of the exterior image of a subcircuit.

7) to save a file with a name, not exceeding 8 characters and **sch** extension (the extension is installed automatically). Now the appropriated name will be used in calculations. Further the command of generation of the netlist is fulfilled; the file with the same name and **cir** extension as a result is automatically generated.

Example of the prepared circuit
Here a simple example with stubs



The netlist

```
*Sheet Name:/
*R3 ? /0 0
*R2 ? /2 0
*R1 ? /1 0
RIN1 /1 /3 10M
E1 /2 /3 RIN 100k 1
R0 /0 /3 0
.end
```

Apparently, the circuit editor automatically changed a name of resistance RIN on RIN1. After correction in the text editor the following netlist is had:

```
RIN /1 /3 10M
E1 /2 /3 RIN 100k 1
R0 /0 /3 0
.end
```

4.2. Graphical representation of a subcircuit in the form of a multipole

It is performed on rules of operation with library. Numbering of outputs should correspond strictly to numbering of the boundary nodes entered at creation of a subcircuit.

4.3. Input of the coordinates and scales

At graphic input of a subcircuit for display of results of calculation in 3d a format (at the command **job.csv**), it is necessary to enter the text where coordinates of a subcircuit and the subcircuit image are installed. The line should begin with **/***. Further coordinates left upper, then the right lower corner of a subcircuit, left upper, then the right lower corner of the image of a subcircuit are specified. For example:

```
/* XLU1=66.04; YLU1=60.96; XRD1=373.4; YRD1=368.3; XLU2=0.0; YLU2=0.0; XRD2=65.9; YRD2=65.9;
```