

# SPICE

<b>1</b>	<b>Was ist es und Geschichte</b>	<b>2</b>
1.1	SPICE Analyse mittels Netlist	2
1.2	Arbeitspunkt, .op Analyse	2
1.3	DC-Sweep	3
1.4	AC Analyse	5
1.5	.TRANS – Transient Analysis	6
<b>2</b>	<b>Aus der Dokumentation zu LT-Spice</b>	<b>7</b>
2.1	General Structure and Conventions	7
2.2	Overview of the lexicon of LTspice:	7
<b>3</b>	<b>Schaltungselemente</b>	<b>8</b>
3.1	Quellen	8
3.1.1	Voltage Source (Spannungsquelle)	8
3.1.2	Current source (Stromquelle)	8
<b>4</b>	<b>Analyse und Simulationsbefehle</b>	<b>9</b>
4.1	.OP - Find the DC Operating Point	9
4.2	.NET – Compute Network Parameters in a .AC Analysis.	9
4.3	.TRAN -- Perform a Nonlinear Transient Analysis	9
<b>5</b>	<b>Parameter Sweep</b>	<b>10</b>
<b>6</b>	<b>Plot</b>	<b>10</b>
<b>7</b>	<b>Beispiele</b>	<b>11</b>
7.1	Sternschaltung ohne Mittelpunktsteiter	11
7.2	Kollektorschaltung	12

# 1 Erste Schritte

SPICE (*Simulation Program with Integrated Circuit Emphasis*) ist eine [Software](#) zur [Simulation](#) analoger, digitaler und gemischter [elektrischer Schaltungen](#) ([Schaltungssimulation](#)).

SPICE wurde ursprünglich an der University of California in Berkeley entwickelt und steht noch heute als Quellcode zur allgemeinen Verfügung. Auf dieser Entwicklung beruhen auch etliche kommerzielle und freie Ableger, die das Original insbesondere um eine graphische Benutzeroberfläche, Schaltplaneditoren und eine Darstellung der Ergebnisse in Diagrammen oder in der Schaltung erweitern. Zusätzlich enthalten diese Ableger oft firmenspezifische Bauteilbibliotheken.

Mehr dazu: [http://de.wikipedia.org/wiki/SPICE\\_\(Software\)](http://de.wikipedia.org/wiki/SPICE_(Software))

Die folgende knappe Einführung in die Simulation mit SPICE versucht, das Thema möglichst unabhängig von diversen Implementierungen darzustellen. Dazu ist der Weg über die Netlist (.cir-Files) ein geeigneter Weg. Zur Analyse habe ich LTSpice verwendet. Obwohl laut der Hilfeseiten von LTSpice die Syntax der Bauteilanweisungen und der Analysebefehle manchmal etwas von der hier verwendeten Syntax abweicht, kann LTSpice die .cir-Files ohne Fehler lesen und abarbeiten.

## 1.1 SPICE Analyse mittels Netlist

Die vielleicht einfachste und effizienteste Art, in die Welt der SPICE-Simulation einzusteigen. Man stellt die Daten für die Simulation mittels einer Textdatei zur Verfügung. In der SPICE Welt nennt man das eine Netlist. Das ist auch das, was die Schaltplaneditoren der verschiedenen SPICE Versionen erzeugen, und dem eigentlichen Simulationsprogramm als Eingabedaten übergeben.

### Aufbau einer Netlist:

Kommentarzeilen beginnen mit einem \*.

Groß- Kleinschreibung spielt keine Rolle.

Für einfache Bauteile ist die Syntax für eine Zeile:

Typ und Name des Bauteils, der erste Buchstaben der Zeile kennzeichnet den Typ des Bauteils.

R      Widerstand

V      Spannungsquelle

I      Stromquelle

Anschlussknoten 1

Anschlussknoten 2

Wert und/oder weitere Parameter

Zusätzlich zur Bauteilliste benötigt SPICE eine Anweisungen zur Analyse. SPICE kennt an die 10 verschiedene Analyseformen.

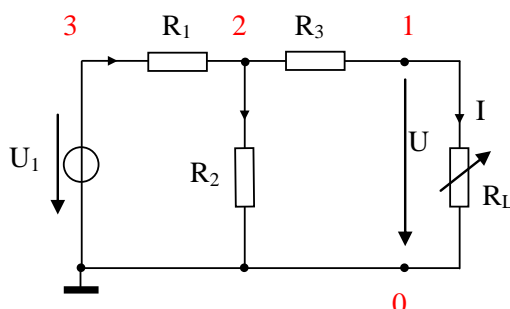
## 1.2 Arbeitspunkt, .op Analyse

Diese Analyse, man nennt sie "Bias Point Analyse" oder "Operating Point" Analyse, entspricht ungefähr unseren bisherigen Berechnungen. Als Resultat liefert sie die Potentiale an den Knotenpunkten.

Es gibt auch einen .DC Analysebefehl, das ist aber nicht das, was wir uns zunächst vorstellen.

Um eine Netlist zu erstellen, müssen wir in der Schaltung alle Anschlusspunkte der Bauteile durchnummerieren. Dabei ergeben sich auch "unechte" Knoten, so nenne ich einen Knoten auf einer einzelnen Leitung, also mit nur einem Strom.

Beispiel:



$$U1 = 12 \text{ V}$$

$$R1 = 4 \Omega$$

$$R2 = 8 \Omega$$

$$R3 = 4 \Omega$$

$$R_L = 10/3 \Omega$$

Mit LTSpice kann man keine neue Netlist erstellen, man kann jedoch eine Textdatei mit der Endung .cir als Netlist öffnen und editieren.

```
* M:\ze\GET\Simulationen\LTSpice\Exampel01.cir
R1 3 2 4
R2 2 0 8
R3 2 1 4
RL 1 0 3.333333
V1 3 0 DC 12
* operating point
.op
.end
```

Als Resultat dieser Analyse erzeugt LTSpice folgende Ausgabe:

```
--- Operating Point ---
V(3): 12          voltage
V(2): 5.86667    voltage
V(1): 2.66667    voltage
I(Rlast): 0.8    device_current
I(R3): 0.8       device_current
I(R2): 0.733333 device_current
I(R1): 1.53333  device_current
I(V1): -1.53333 device_current
```

Das die hier etwas verbesserte Ausgabe zuerst scheußlich formatiert ist, keine Einheiten und keine Vorsilben wie m, k ... verwendet, muss man in Kauf nehmen.

Jedenfalls bestätigen die Resultat unsere Berechnung mit der Methode der Ersatzquelle.

### 1.3 DC-Sweep

Diese Analyse ändert den Wert von einer oder mehreren DC-Quellen und berechnet die Ströme und Spannungen als Funktion der variablen Spannung(en).

Beispiel: Dioden-Kennlinie

```
* Diodenkennlinie.cir
V1 1 0 DC Rser=100
D1 1 0 MyDiode
.model MyDiode D(Ron=0.2 Roff=1Meg Vfwd=0.65 Epsilon=0.2)
.dc V1 0 10 0.1
.end
```

#### **Beschreibung:**

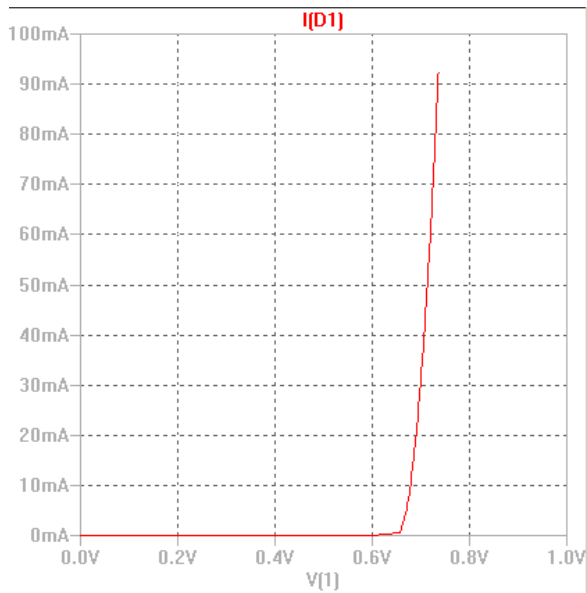
```
.dc V1 0 10 0.1
```

Im .DC Analysebefehl wird angegeben, für welche Quelle(n) die Spannungen verändert wird (V1). Die nächsten Parameter sind Anfangswert (0 V), Endwert (10 V) und Schrittweite (0.1 V).

Für die Diode zwischen Punkt 1 und 0 ist ein Modell angegeben, das mit .model definiert wird.

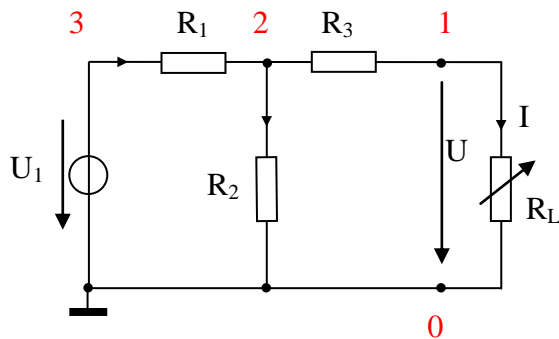
Ron ist der differentielle Widerstand in Durchlassrichtung, Roff der Widerstand in Sperrrichtung, Vfwd ist die Spannung, bei der die Diode leitend wird und mit Epsilon kann man den Übergang um den Knickpunkt "glätten".

Das Resultat in einem Diagramm dargestellt:



**Beispiel:**

Nun bauen wir diese Diode als Last ( $R_L$ ) in eine Schaltung ein.



```
* ...\Exampel02.cir
R1 3 2 4
R2 2 0 8
R3 2 1 4
D1 1 0 DIdeal
.model DIdeal D(Ron=0.2 Roff=1Meg Vfwd=0.65 Epsilon=0.2)
V1 3 0 DC 12
* DC Analyse , operation point
.op
.end
```

**Resultat der Analyse:**

```
--- Operating Point ---
V(3): 12          voltage
V(2): 5.18447    voltage
V(1): 0.96116    voltage
I(D1): 1.05583    device_current
I(R3): 1.05583    device_current
I(R2): 0.64805    device_current
I(R1): 1.70388    device_current
I(V1): -1.70388  device_current
```

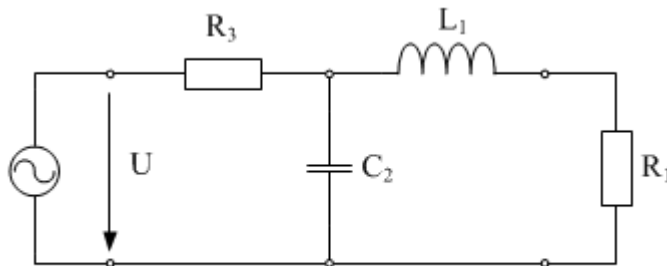
## 1.4 AC Analyse

Für die Berechnung der Spannungen und Ströme in einem AC Netzwerk definiert man eine Spannungsquelle mit der Option AC amplitude, phase

Syntax für AC Spannungsquelle:

VName n+ n- [AC=<amplitude>, phase] [Rser=<value>] [Cpar=<value>]

Als Beispiel die Aufgabe 4 des Übungsblattes "Berechnung und Analyse von AC-Schaltungen - Teil 1":



AC14a

```
* AC14a.cir
```

```
VIn 1 0 AC 10, 0 ; Sinus-AC Quelle für die AC Analyse
```

```
R1 1 2 50
```

```
C3 2 0 1.27u
```

```
L2 2 3 20m
```

```
R2 3 0 200
```

```
* AC Analyse mit Amplitude and Phase der Quelle VIn und f = 1 kHz
```

```
.ac List 1k
```

```
.net VIn ; erweitert die AC Analyse um die Berechnung von Zin und Yin
```

```
.op
```

```
.end
```

**Das (gekürzte) Resultat der Analyse:**

frequency:	1000 Hz			
V(1):	mag: 10	phase: 0°	voltage	
V(2):	mag: 8.24055	phase: -13.6494°	voltage	
V(3):	mag: 6.97752	phase: -45.7912°	voltage	
Zin(vin):	mag: 179.602	phase: -44.3076°	impedance	
Yin(vin):	mag: 0.00556785	phase: 44.3076°	conductance	
I(C3):	mag: 0.0657567	phase: 76.3506°	device_current	
I(L2):	mag: 0.0348876	phase: -45.7912°	device_current	
I(R2):	mag: 0.0348876	phase: -45.7912°	device_current	
I(R1):	mag: 0.0556785	phase: 44.3076°	device_current	

## 1.5 .TRANS – Transient Analysis

Diese Analyse berechnet den zeitlichen Verlauf der Potentiale und Ströme, also eigentlich den Einschalt/Einschwingvorgang.

### Syntax:

```
.TRAN <Tstep> <Tstop> [Tstart [dTmax]] [modifiers]
```

Tstep ist die Schrittweite für die Resultate der Berechnung und auch der Anfangswert für die Schrittweite der iterativen Berechnung.

Tstop ist die Zeit für das Ende der Berechnung

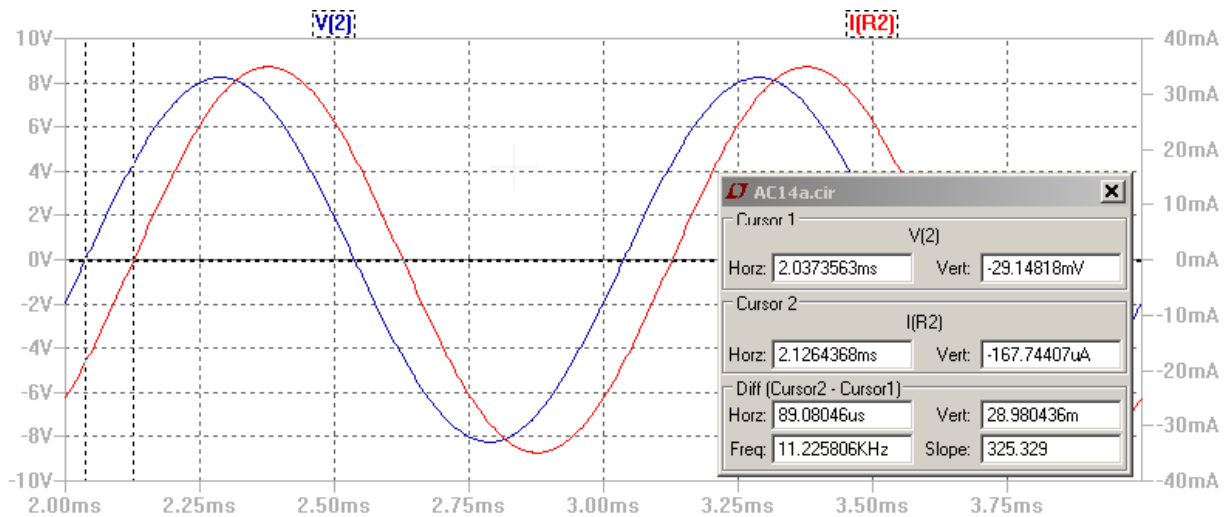
Tstart: bis zu dieser Zeit werden die Ausgaben unterdrückt

Voraussetzung für die Transienten-Analyse ist eine Quelle vom Typ SINE oder eine andere zeitabhängige Quelle.

### Beispiel:

```
* AC14a.cir
* Sine wave voltage source for TRANS analyse
VIn 1 0 SINE(0 10 1k 0 0 0)
R1 1 2 50
C3 2 0 1.27u
L2 2 3 20m
R2 3 0 200
.TRANS 2u 4m
.end
```

### Das Resultat mit Cursordaten:



Von der AC-Analyse hatten wir für das Potential am Punkt 2, das ist die Spannung am Zweig L+R2 einen Wert von 8.24 V / -13.6°. Für den Strom einen Wert von 35 mA / -45,8°.

Cursor 1 beim Nulldurchgang der Spannung steht bei 2,037 ms, das sind bei 1 KHz - 13,3 °, der zweite Cursor beim Nulldurchgang des Stromes steht bei 2,126 ms, das sind ca. - 45,4 °.

Auch die Amplituden, ca. 8,2 V und 34,9 mA stimmen mit der vorigen Analyse und der "händischen" Lösung der Aufgabe überein.

## 2 Aus der Dokumentation zu LT-Spice

### 2.1 General Structure and Conventions

The circuit to be analysed is described by a text file called a netlist. The first line in the netlist is ignored, that is, it is assumed to be a comment. The last line of the netlist is usually simply the line ".END", but this can be omitted. Any lines after the line ".END" are ignored.

The order of the lines between the comment and end is irrelevant. Lines can be comments, circuit element declarations or simulation directives. Let's start with an example:

```
* This first line is ignored
* The circuit below represents an RC circuit driven
* with a 1MHz square wave signal
R1 n1 n2 1K ; a 1KOhm resistor between nodes n1 and n2
C1 n2 0 100p ; a 100pF capacitor between nodes n2 and ground
V1 n1 0 PULSE(0 1 0 0 0 .5μ 1μ) ; a 1Mhz square wave
.tran 3μ ; do a 3μs long transient analysis
.end
```

The first two lines are comments. Any line starting with a "\*" is a comment and is ignored. The line starting with "R1" declares that there is a 1K resistor connected between nodes n1 and n2. Note that the semicolon, ";", can be used to start a comment in the middle of a line. The line starting with "C1" declares that there is a 100pF capacitor between nodes n2 and ground. The node "0" is the global circuit common ground.

### 2.2 Overview of the lexicon of LTspice:

Letter case, leading spaces, blanks, and tabs are ignored.

The first non-blank character of a line defines the type of circuit element.

Leading Character	Type of line
*	Comment
A	Special function device
B	Arbitrary behavioral source
C	Capacitor
D	Diode
E	Voltage dependent voltage source
F	Current dependent current source
G	Voltage dependent current source
H	Current dependent voltage source
I	Independent current source
J	JFET transistor
K	Mutual inductance
L	Inductor
M	MOSFET transistor
O	Lossy transmission line
Q	Bipolar transistor
R	Resistor
S	Voltage controlled switch
T	Lossless transmission line

U	Uniform RC-line
V	Independent voltage source
W	Current controlled switch
X	Subcircuit Invocation
Z	MESFET transistor
.	A simulation directive, For example: .options reitoll=1e-4
+	A continuation of the previous line. The "+" is removed and the remainder of the line is considered part of the prior line.

Numbers can be expressed not only in scientific notation; e.g., 1e12; but also using engineering multipliers. That is, 1000.0 or 1e3 can also be written as 1K. Below is a table of understood multipliers:

Suffix Multiplier: T, G, Meg, K, m, u(or  $\mu$ ), n, p, f

The suffixes are not case sensitive. Unrecognized letters immediately following a number or engineering multiplier are ignored. Hence, 10, 10V, 10Volts, and 10Hz all represent the same number, and M, MA, MSec, and MMhos all represent the same scale factor(.001). A common error is to draft a resistor with value of 1M, thinking of a one Megaohm resistor, however, 1M is interpreted as a one milliohm resistor. This is necessary for compatibility with standard SPICE practice.

LTspice will accept numbers written in the form 6K34 to mean 6.34K. This works for any of the multipliers above. It can be turned off by going to Tools=>Control Panel=>SPICE and unchecking "Accept 3K4 as 3.4K".

Nodes names may be arbitrary character strings. Global circuit common node(ground) is "0", though "GND" is special synonym. Note that since nodes are character strings, "0" and "00" are distinct nodes.

Throughout the following sections of the manual, angle brackets are placed around data fields that need to be filled with specific information; for example, "<srcname>" would be the name of some specific source. Square brackets indicate that the enclosed data field is optional.

## 3 Schaltungselemente

### 3.1 Quellen

#### 3.1.1 Voltage Source (Spannungsquelle)

Das ist die Gleichspannungsquelle, hat aber einen AC Eintrag, dessen Wert als Amplitude bei der AC Analyse verwendet wird. Die Frequenz wird mit dem .AC Befehl festgelegt.

Syntax:

```
Vxxx n+ n- <voltage> [AC=<amplitude>] [Rser=<value>] [Cpar=<value>]
```

This element sources a constant voltage between nodes n+ and n-. For AC analysis, the value of AC is used as the amplitude of the source at the analysis frequency. A series resistance and parallel capacitance can be defined.

Voltage sources have historically been used as the current meters in SPICE and are used as current sensors for current-controlled elements. If Rser is specified, the voltage source cannot be used as a sense element for F, H, or W elements. However, the current of any circuit element, including the voltage source, can be plotted.

#### 3.1.2 Current source (Stromquelle)

Syntax:

```
Ixxx n+ n- <current> [AC=<amplitude>] [load]
```



This circuit element sources a constant current between nodes n+ and n-. If the source is flagged as a load, the source is forced to be dissipative, that is, the current goes to zero if the voltage between nodes n+ and n- goes to zero or a negative value. The purpose of this option is to model a current load on a power supply that doesn't draw current if the output voltage is zero.

## 4 Analyse und Simulationsbefehle

### 4.1 .OP - Find the DC Operating Point

Perform a DC operating point solution with capacitances open circuited and inductances short circuited. Usually a DC solution is performed as part of another analysis in order to find the operating point of the circuit. Use .op if you wish only this operating point to be found. The results will appear in a dialog box. After a .OP simulation, when you point at a node or current the .OP solution will appear on the status bar.

Syntax:

```
.OP
```

### 4.2 .NET – Compute Network Parameters in a .AC Analysis.

This statement is used with a small signal(.AC) analysis to compute the input and output admittance, impedance, Y-parameters, Z-parameters, H-parameters, and S-parameters of a 2-port network. It can also be used to compute the input admittance and impedance of a 1-port network. This must be used with a .AC statement, which determines the frequency sweep of the network analysis.

Syntax:

```
.net [V(out[,ref])|I(Rout)] <Vin|Iin> [Rin=<val>] [Rout=<val>]
```

The network input is specified by either an independent voltage source, <Vin>, or an independent current source, <Iin>.

The optional output port is specified either with a node, V(out), or a resistor, I(Rout). The ports will be terminated with resistances Rin and Rout.

If unspecified, the termination impedances default to 1 Ohm except in the case of the Voltage source with an Rser specified or an output port specified with a resistor. In those two cases the termination resistances defaults to the device impedance. Termination values specified on the .NET statement will override device impedances for the .NET calculation, but not for the normal .AC node voltages and currents. That is, the .NET statement will not impose terminating impedances on the network for the normal voltages and currents computed as part of the .AC analysis.

### 4.3 .TRAN -- Perform a Nonlinear Transient Analysis

Perform a transient analysis. This is the most direct simulation of a circuit. It basically computes what happens when the circuit is powered up. Test signals are often applied as independent sources.

Syntax: .

```
.TRAN <Tstep> <Tstop> [Tstart [dTmax]] [modifiers]  
.TRAN <Tstop> [modifiers]
```

The first form is the traditional .tran SPICE command.

Tstep is the plotting increment for the waveforms but is also used as an initial step-size guess. LTspice uses waveform compression, so this parameter is of little value and can be omitted or set to zero.

Tstop is the duration of the simulation. Transient analyses always start at time equal to zero. However, if Tstart is specified, the waveform data between zero and Tstart is not saved. This is a means of managing the size of waveform files by allowing startup transients to be ignored.

dTmax, is the maximum time step to take while integrating the circuit equations. If Tstart or dTmax is specified, Tstep must be specified.

Modifiers:

UIC: Skip the D.C. operating solution and use user-specified initial conditions.

steady: Stop the simulation when steady state has been reached.

nodiscard: Don't delete the part of the transient simulation before steady state is reached.

startup: Solve the initial operating point with independent voltage and current sources turned off.

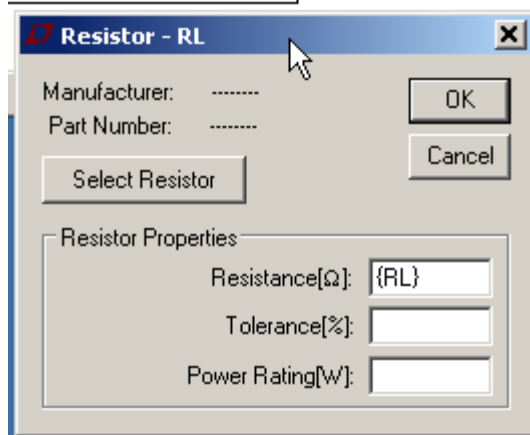
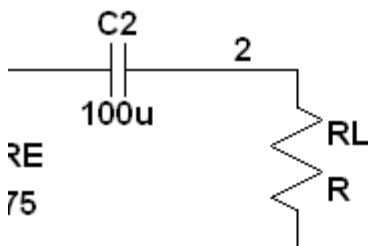
Then start the transient analysis and turn these sources on in the first 20 us of the simulation.

step: Compute the step response of the circuit.

## 5 Parameter Sweep

Um den Wert eines Bauteils zu verändern, ist folgende Vorgangsweise erforderlich:

Beispiel: Das Verhalten einer Schaltung bei unterschiedlichem Lastwiderstand RL soll untersucht werden.



Widerstand auswählen und als Wert {name}, hier eben {RL} eingeben.

Dann als Spice Directive folgendes Kommando eingeben:

```
.step param RL 0.1 5 0.1
```

Dazu macht man z.B. eine AC Analyse mit einer Frequenz.

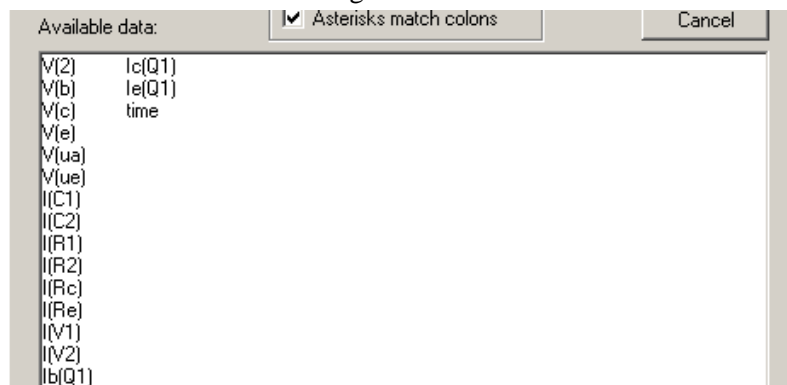
Für die Darstellung des Resultats siehe das Beispiel Kollektorschaltung.

## 6 Plot

Ein Graphik-fenster wird nur geöffnet, wenn für die Analyseform eine graphische Darstellung auch sinnvoll ist.

Beispiel: Transientenanalyse für einen Verstärker:

Wählt man über das Kontext-Menü "Add Trace" so ist klar, wie man hier Signale auswählt, die dann über der Zeit als x-Achse dargestellt werden.

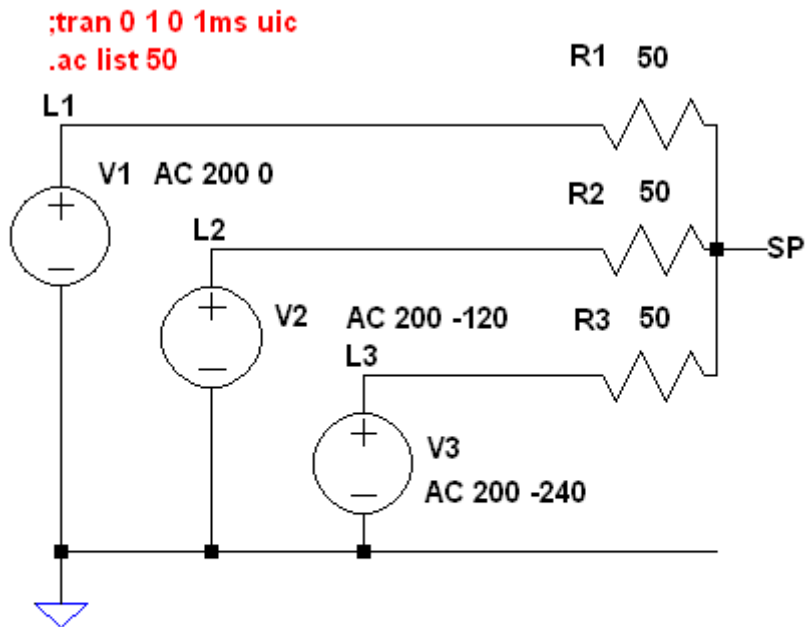


Man kann die verfügbaren Daten aber auch mit Formeln verknüpfen.

Zusätzlich stehen fast alle üblichen mathematischen Funktionen, wie sie z.B. ein C-Programmierer aus der Mathematik-Bibliothek kennt, zur Verfügung. Siehe dazu die Hilfe zu Waveform-Arithmetic.

## 7 Beispiele

### 7.1 Sternschaltung ohne Mittelpunktsleiter



`SINE(0 200 50 0 0 0 2)`

`SINE(0 200 50 0 0 -120 2)`

`SINE(0 200 50 0 0 -240 2)`

\* ....\YSchaltung.asc

V1 L1 0 SINE(0 200 50 0 0 0 2) AC 200 0

V2 L2 0 SINE(0 200 50 0 0 -120 2) AC 200 -120

V3 L3 0 SINE(0 200 50 0 0 -240 2) AC 200 -240

R1 L1 SP 50

R2 L2 SP 50

R3 L3 SP 50

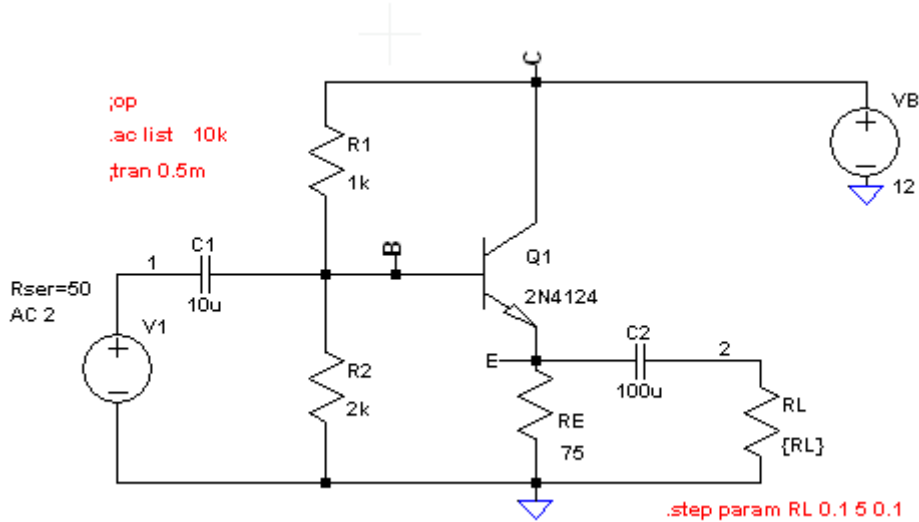
;ac list 50

.tran 0 1 0 1ms uic

.backanno

.end

## 7.2 Kollektorschaltung



Siehe die Formeln für den Betrag von  $u$  und  $i$  und die Formel für die Leistung.

Bei der AC Analyse mit einem Parameter werden als Diagramme angeboten:

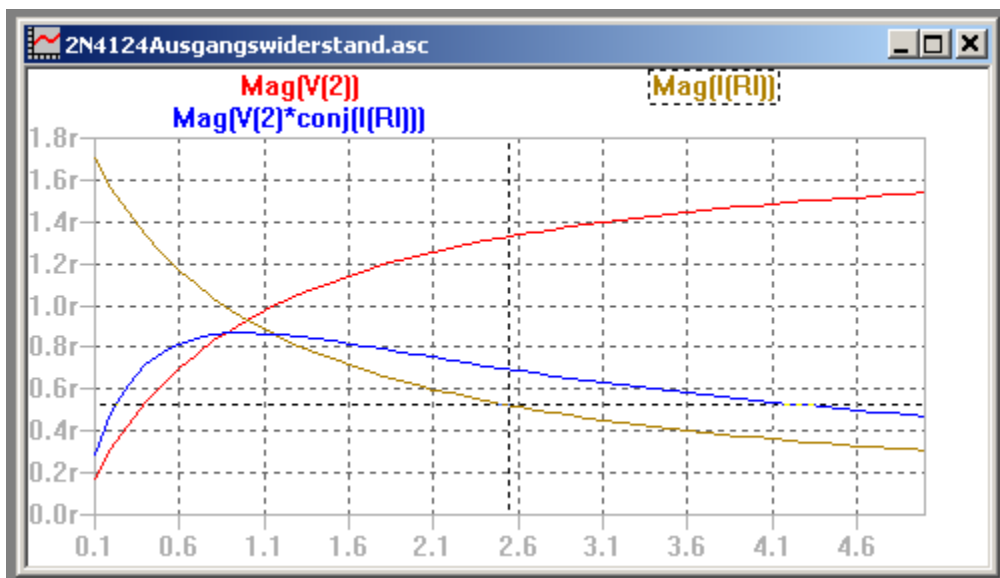
Bode

Nyquist

Cartesian

Bei Cartesian wird der Real- und Imaginärteil separat mit der Achse für den Realteil auf der linken Seite und der Achse für den Imaginärteil auf der rechten Seite, dargestellt

Klickt man auf die jeweiligen Achsen, so wird angeboten, den Real- oder eben den Imaginärteil nicht zu zeichnen. Davon wurde in dem Diagramm auch Gebrauch gemacht.



Der Kleinsignal-Ausgangswiderstand der Schaltung ist  $\sim 1 \Omega$ , die Analyse zeigt, dass bei  $RL \sim 1 \Omega$  Leistungsanpassung herrscht.