

**Einführung in das  
Simulationsprogramm**

# **SPICE**

**SPICE 3f4, PSPICE, LT-SPICE**

Prof. Dr.-Ing. Karl-Hermann Cordes

Version 2.7; Dez. 2013

# Inhalt

<b>1</b>	<b>Vorbemerkungen</b>	<b>4</b>
<b>2</b>	<b>Installation</b>	<b>5</b>
2.1	Einrichten eines Ordners „CIRCUIT“	5
2.2	Einrichten eines Ordners „SCHEMATIC“	5
2.3	Einrichten eines Ordners „SPICELIB“	5
2.4	SPICE 3f4	5
2.5	PSPICE	6
2.6	LT-SPICE	6
<b>3</b>	<b>Erzeugen des Simulations-Files (Circuit-Datei)</b>	<b>7</b>
3.1	Schaltbild (Schematic)	7
3.2	Simulations-File (Circuit-Datei)	7
<b>4</b>	<b>Simulatoren</b>	<b>9</b>
4.1	Simulator SPICE 3f4	9
4.2	Simulator PSPICE	10
4.3	Simulator LT-SPICE	11
<b>5</b>	<b>Konventionen</b>	<b>12</b>
5.1	Zahlenformate	12
5.2	Bauteile und deren Kennbuchstaben	12
5.3	Bauteiltypen und deren Kennung in der Modell-Anweisung	13
<b>6</b>	<b>Bauteile</b>	<b>14</b>
6.1	Passive Bauteile	15
6.1.1	Widerstand (RES)	15
6.1.2	Kondensator (CAP)	16
6.1.3	Spule (IND)	16
6.1.4	Gekoppelte Spulen (CORE)	17
6.2	Quellen	18
6.2.1	Unabhängige Spannungsquelle (V)	18
6.2.2	Unabhängige Stromquelle (I)	20
6.2.3	Spannungsgesteuerte Spannungsquelle (E)	20
6.2.4	Spannungsgesteuerte Stromquelle (G)	21
6.2.5	Nichtlineare Quellen (Spannungs- oder Stromquelle) in <b>SPICE 3f4</b> (B)	21
6.2.6	Stromgesteuerte Spannungsquelle (H)	21
6.2.7	Stromgesteuerte Stromquelle (F)	21
6.3	Aktive Bauteile	22
6.3.1	Diode (D)	22
6.3.2	Bipolar-Transistoren (NPN bzw. PNP)	22
6.3.3	Sperrschicht-FET (NJF bzw. PJF)	23
6.3.4	MOS-FET (NMOS bzw. PMOS)	24
6.4	Teilschaltungen (Subcircuits)	25

<b>7</b>	<b>Analysearten</b>	<b>26</b>
7.1	DC-Analyse (.DC)	27
7.1.1	DC-Sweep einer unabhängigen Quelle	27
7.1.2	DC-Sweep der Temperatur in <b>PSPICE</b> und <b>LT-SPICE</b>	28
7.2	AC-Analyse (.AC)	28
7.3	Rausch-Analyse in <b>PSPICE</b> und <b>LT-SPICE</b> (.NOISE)	31
7.4	Analyse im Zeitbereich (.TRAN)	33
7.5	Fourier-Analyse in <b>PSPICE</b> und <b>LT-SPICE</b> (.FOUR)	33
7.6	Fast-Fourier-Transformation in <b>PSPICE</b> und <b>LT-SPICE</b> (FFT-Plot)	36
<b>8</b>	<b>Mathematische Operatoren, arithmetische Ausdrücke</b>	<b>37</b>
8.1	Eingebaute Operationen und Ausdrücke	37
8.2	Anwenderdefinierte Funktionen	38
<b>9</b>	<b>Optionen und Kontrollparameter</b>	<b>39</b>
9.1	Die Anweisung .OPTIONS	39
9.2	Temperatur	40
9.3	Rechengenauigkeit	40
<b>10</b>	<b>Parameter-Variation in <b>PSPICE</b> und <b>LT-SPICE</b></b>	<b>41</b>
<b>11</b>	<b>Literatur</b>	<b>43</b>
11.1	Literatur zu den Simulationsprogrammen SPICE 3f4, <b>PSPICE</b> und <b>LT-SPICE</b>	43
11.2	Literatur zur Modellparameter-Bestimmung	43

# 1 Vorbemerkungen

**SPICE** (Simulation Program with Integrated Circuit Emphasis) wurde Ende der sechziger Jahre an der Universität Berkeley, California, entwickelt und gilt immer noch als der Weltstandard für die Simulation elektrischer Schaltungen. Verschiedene Versionen, wie z.B. PSPICE, Electronics Workbench, H-SPICE, T-SPICE, Win-SPICE, LT-SPICE, ..., sind hiervon abgeleitet.

SPICE löst die Kirchhoff'schen Gleichungen (Knoten- und Maschengleichungen) einer Schaltung; und dieses nicht nur für Gleichspannungen und -ströme (DC-Analyse; .DC), sondern auch für sinusförmige Wechselgrößen (AC-Kleinsignal-Analyse; .AC) und allgemeine Zeitfunktionen (Transienten-Analyse; .TRAN). Neben diesen drei Grund-Analysearten stehen noch weitere, sehr nützliche Erweiterungen zur Verfügung: DC-Empfindlichkeits-Analyse (.SENS), Kleinsignal-Transferfunktion (.TF), Rausch-Analyse (.NOISE), Verzerrungs-Analyse (.DISTO), Fourier-Analyse (.FOUR).

Die Lösung der Kirchhoff'schen Gleichungen erfolgt *numerisch*.

Es sind alle wichtigen elektrischen Bauelemente (passive und aktive Bauelemente, lineare und nichtlineare Bauelemente, unabhängige und gesteuerte Quellen) in SPICE implementiert. Auch mathematische Ausdrücke können eingebunden werden.

Die zu simulierende Schaltung wird in Form einer Circuit-Datei (Endung „CIR“ oder „NET“) dem Simulator übergeben. Eine solche Circuit-Datei enthält – neben einer Titelzeile und einem .END-Statement – die Netzliste, die die Verbindungen der Bauteile untereinander beschreibt, die Spannungs- und Stromquellen, sowie die Simulationsanweisung und eventuell einige Optionen (.OPTIONS).

Das Simulationsergebnis wird vorzugsweise grafisch dargestellt. Eine Ausgabe in Tabellenform oder als File ist ebenfalls möglich.

Auf den folgenden Seiten soll in kurzer Form der Umgang mit SPICE gezeigt werden. Wir wollen uns hier auf drei Programmversionen, nämlich SPICE 3f4 (Berkeley-SPICE, von Wolfgang Mues für PC-Windows lauffähig gemacht und kostenfrei erhältlich [2]), PSPICE (Firma MicroSim) und LT-SPICE [6] beschränken. Während die Version SPICE 3f4 nur durch den verfügbaren Speicher im Rechner begrenzt ist, ist die PSPICE-*Demo*-Version (nur die *Demo*-Version ist kostenfrei erhältlich) „naturgemäß“ nur für kleine Schaltungen mit wenigen Bauteilen und Knoten geeignet. Dagegen bietet PSPICE mehr Komfort. LT-SPICE ist eine relativ neue, besonders gute und zudem noch kostenfreie SPICE-Vollversion der Firma Linear Technology.

Obwohl es bei größeren Schaltungen üblich ist, die Netzliste mittels eines speziellen grafischen Werkzeuges (Schematic Entry) zu generieren, soll hier bewusst darauf verzichtet werden. Für den erfolgreichen Umgang mit SPICE ist das Verstehen der Circuit-Datei wichtig, denn auch andere Werkzeuge können ein SPICE-File liefern und man muss dieses dann für eine Simulation vorbereiten können. Außerdem ist die Circuit-Datei stets das Eingangs-File für den Simulator und hat für alle SPICE-Derivate prinzipiell den gleichen Aufbau.

Zuerst muss das Programm auf dem Rechner installiert werden. Sodann werden wir den folgenden Fragen nachzugehen haben:

- Wie wird ein Simulations-File erstellt und wo wird es gespeichert?
- Wie wird die Simulation gestartet?
- Wo ist das Ergebnis zu finden und wie kann es dargestellt werden?

Alle Bauteile und Analyse-Anweisungen werden durch Abkürzungen ausgedrückt. Eine Zusammenstellung der wichtigsten Kürzel ist deshalb ganz hilfreich. – Das folgende Skript ist so gegliedert, dass es einerseits der Reihe nach durchgearbeitet werden kann, andererseits aber auch zum Nachschlagen gute Dienste leistet.

Leider unterscheiden sich einige Analyse-Anweisungen in den drei Programmen etwas voneinander. Da hier nur eine kurze Einführung gegeben werden soll und nicht auf alle Unterschiede eingegangen werden kann, wird in Zweifelsfällen auf die recht ausführlichen „eingebauten“ Hilfen verwiesen, die in allen drei genannten Programmen enthalten sind.

## 2 Installation

Im Folgenden soll die Installation der drei oben genannten *kostenfrei* zur Verfügung stehenden Simulatoren beschrieben werden. Bevor jedoch mit der Installation des eigentlichen Simulators begonnen wird, sollte man sich, unabhängig davon, mit welchem Simulator man arbeiten möchte, auf der Arbeits-Partition der Festplatte (z.B. D:) drei neue Ordner anlegen: „CIRCUIT“, „SCHEMATIC“ und „SPICELIB“.

### 2.1 Einrichten eines Ordners „CIRCUIT“

Zum Abspeichern der Simulationsdateien wird ein Ordner CIRCUIT eingerichtet. Alle hier abgelegten Circuit-Dateien erhalten die Endung „CIR“. Diese Endung ist zwar nicht unbedingt notwendig, sie erhöht aber die Übersicht. **PSPICE** legt z.B. auch die Ergebnisdateien im CIRCUIT-Ordner ab, allerdings mit anderen Endungen.

### 2.2 Einrichten eines Ordners „SCHEMATIC“

Wenn man mit **PSPICE** oder **LT-SPICE** arbeiten möchte, besteht die Möglichkeit der grafischen Schaltungsplaneingabe. Beide Simulatoren verfügen nämlich über einen eingebauten Schaltplan-Editor. Im Ordner SCHEMATIC können dann die eigenen Schaltungen abgespeichert werden; bei **PSPICE** mit der Endung „sch“ und bei **LT-SPICE** mit der Endung „asc“.

### 2.3 Einrichten eines Ordners „SPICELIB“

Damit Modelle von Bauelementen (Transistoren usw.) leicht zu finden sind, sollte ein Bibliotheks-Ordner, z.B. SPICELIB eingerichtet werden. Hier können Modelle (auch selbsterstellte Modelle) und Subcircuits abgelegt werden. Unter SPICELIB können auch noch weitere Unterordner eingerichtet werden.

### 2.4 SPICE 3f4

- 1 **SPICE** (Programm siehe [2]) direkt auf der Partition C: installieren. Dazu die gepackte Datei „SPICE3F4.ZIP“ extrahieren. Ziel: direkt ins Laufwerk C: Dort wird ein Ordner „SPICE“ angelegt, der alle Programm-Dateien enthält. Der Pfad sollte nicht verändert werden.

- 2 In der **SPICE**-Startdatei (C:\SPICE\LIB\SCRIPTS\SPINIT) das SPICE-LIB-Directory und einen Text-Editor (z.B. Notepad) angeben:

```
SET SPICE_LIB_DIR=C:\SPICE\LIB
SET EDITOR=C:\WINDOWS\NOTEPAD.EXE
```

- 3 Windows-Erweiterung WIN32S installieren (nur bei Windows 3.1x), auf keinen Fall bei den neueren Windows-Versionen!
- 4 Die Datei CTL3D32.DLL ins Windows-System-Verzeichnis kopieren; aber nur, wenn dort noch keine (neuere) vorhanden ist. Beispielsweise kann die Datei „CTL3D32.KHC“ verwendet werden:

```
COPY C:\SPICE\CTL3D32.KHC C:\WINDOWS\SYSTEM32\CTL3D32.DLL ↵
```

- 5 Anlegen des Programms **SPICE 3f4** auf dem Desktop:

a) Windows 3.1x:

Datei → Neu → Programm → OK ; Eingabe:

Beschreibung: SPICE3f4 (oder einfach nur SPICE)

Befehl: C:\SPICE\BIN\SPICE.EXE

Arbeitsverzeichnis:

Bestätigen der Eingaben mit OK

b) Windows 95, 98, 2000, NT, XP:

Auf dem Desktop mittels des Windows-Explorers eine Verknüpfung mit

`C:\SPICE\BIN\SPICE.EXE`

herstellen. Das Symbol  erscheint auf dem Desktop

**6 SPICE** starten durch Doppelklick auf das obige Symbol.

**Anmerkung:** Bei dieser Art der Installation erscheint **SPICE 3f4** nicht in der Registry.

Nun kann man loslegen. Die Eingabe der Befehle erfolgt in das Kommandofenster. Von hier kann auch der Text-Editor mit dem Befehl `edit D:\CIRCUIT\...` gestartet werden. Der Umgang mit dem Kommandofenster wird später an einem konkreten Beispiel gezeigt.

Ein User-Manual ist in den Dateien `C:\SPICE\SPICE.TXT` und `\SPICE.pdf` enthalten.

## 2.5 PSPICE

**1** Das Programm **PSPICE** wird von der CD-ROM aus entsprechend den Anweisungen installiert. Ab Version 8 läuft **PSPICE** nur auf Windows 95 (und höher) und NT; nicht mehr auf 3.1x!

**2** Rechner und Windows neu starten.

**3** Den Designmanager des Programms **PSPICE** über

**Start** – Programme – Design Lab Release x<sup>\*)</sup> – Design Manager öffnen.

<sup>\*)</sup> Der Programmname kann, je nach Ausgabe der CD-ROM, unterschiedlich sein!

Wir benötigen für unsere Arbeiten hier nur

- das Simulationsprogramm **PSpiceA\_D** oder **PSPICE**
- und einen Text-Editor (z.B. MicroSim Text Edit)



Diese beiden Programmteile öffnen wir vom Design Manager aus.

Pspice 8

Text Edit

Wir können mittels des Windows Explorers auch eine Verknüpfung auf dem Desktop herstellen.

**4** Mit dem Text-Editor wird eine neue Circuit-Datei erstellt bzw. eine zu bearbeitende Datei geöffnet (siehe später).

## 2.6 LT-SPICE (SwCADIII; neue Version: LTspiceIV) [6], [14]

**1** Die Webseite von Linear Technology ( [www.linear.com/software](http://www.linear.com/software) ) aufsuchen und die „gepackte“ Datei `LTspiceIV.exe` herunterladen und z.B. im Ordner „Temp“ *temporär* speichern.

**2** Von dort aus die Installation durch Doppelklick auf die Datei `LTspiceIV.exe` starten. Um Probleme mit der Benutzerkontenverwaltung bei neueren Windows-Versionen (ab Windows 7) zu umgehen, wird das Programm am besten direkt unter C: installiert, also `C:\LTC\LTspiceIV\`. (d.h. nicht den empfohlenen Pfad `C:\Programme\LTC\LTspiceIV\` verwenden!).

**3** Auf dem Desktop wird automatisch eine Verknüpfung mit `LTspiceIV.exe` hergestellt. Das Symbol



LTspice IV

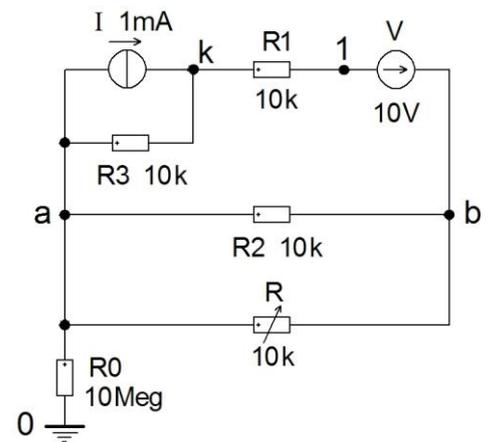
### 3 Erzeugen des Simulations-Files (Circuit-Datei)

Bevor wir den Umgang mit den drei Simulatoren besprechen, wollen wir uns den Aufbau des SPICE-Eingangs-Files, die Circuit-Datei, ansehen. Dies gelingt am besten mit einem einfachen Beispiel. Gegeben sei die in **Bild 1** dargestellte Schaltung.

#### 3.1 Schaltbild (Schematic)

Beim Zeichnen des Schaltbildes sollte man sich bemühen, gleich die in SPICE üblichen Regeln zu beachten. Für die Bauelemente sind „Kennbuchstaben“ definiert, jedem Bauelement muss ein Wert zugewiesen werden, der Bezug zur „Masse“ (GND) darf nicht fehlen und jeder Knoten sollte mit einem Namen versehen werden:

- Als Knoten-Namen können Buchstaben oder Ziffern verwendet werden.
- Der Knoten „0“ (= GND) darf nicht fehlen.
- Sämtliche Knoten einer Schaltung müssen einen DC-Pfad zum Knoten „0“ haben. Es ist zu bedenken, dass ideale Stromquellen und Kondensatoren einen *unendlich* großen DC-Widerstand haben! Wenn die Schaltung praktisch „erdfrei“ bleiben soll, kann ein zusätzlich eingefügter hochohmiger Widerstand (hier R0, der Wert ist unkritisch) das Problem lösen.
- Für die Schaltungselemente werden definierte „Kennbuchstaben“ verwendet: Unabhängige Stromquellen werden mit „I“ und unabhängige Spannungsquellen mit „V“ (nicht mit „U“) bezeichnet. Ohmsche Widerstände erhalten den Kennbuchstaben „R“, siehe Abschnitt 5.2.
- In SPICE wird nicht zwischen Groß- und Kleinschreibung unterschieden. Trotzdem sind in Schaltbildern große Anfangsbuchstaben der Bauelemente üblich.



**Bild 1**

Beispiel B1 Zweipol mit einer Strom- und einer Spannungsquelle

#### 3.2 Simulations-File (Circuit-Datei)

Das Simulations-File ist eine ganz normale Textdatei und kann mit jedem beliebigen Text-Editor erstellt werden. Zwischen Groß- und Kleinschreibung wird nicht unterschieden. Sonderzeichen und besondere Formatierungen sind nicht zulässig. Wichtig zu bemerken ist ferner, dass statt eines Dezimal-Kommas ein Punkt zu verwenden ist. Jede Zeile muss *ganz links beginnen!* Eine Circuit-Datei ist prinzipiell wie folgt aufgebaut:

- Titel-Anweisung

Die **erste Zeile** einer Circuit-Datei ist die „Titel-Anweisung“. Sie dient als Überschrift oder Information zur Schaltung. Will man keine Überschrift, bleibt die erste Zeile leer. Die erste Zeile wird von SPICE immer als Kommentar verstanden und somit ignoriert. **SPICE 3f4** erfordert mindestens *ein* Zeichen.

- Element-Anweisungen zur Schaltungsbeschreibung (Netzliste).

Jedes Element wird entsprechend seiner „Lage“ in der Schaltung durch eine Element-Anweisung der Form  
 Element Knoten1 Knoten2 ... Kennwert

in der „Netzliste“ aufgeführt. Dabei ist für jeden Elementtyp (Referenz) – wie im Schaltbild – ein charakteristischer **Anfangsbuchstabe** vorgeschrieben (Kennbuchstabe, siehe Abschnitt 5.2). Der Widerstand R2 (das Element R2), der zwischen den Knoten „a“ und „b“ liegt und den Kennwert 10 kΩ besitzt, wird also durch die folgende Zeile in die Netzliste eingefügt:

R2 a b 10kOhm (zwischen Maßzahl und Einheit darf bei SPICE kein Leerzeichen stehen!)

Da SPICE ein *numerisches* Lösungsverfahren anwendet, werden alle Werte nur durch Zahlen angegeben. Für Zehnerpotenzen werden die üblichen Abkürzungen ( $k = 10^3 = E3$ , siehe aber Abschnitt 5.1) verwendet und *kein* Leerzeichen zwischen 10 und k gesetzt. Die Angabe Ohm hinter dem Skalierungszeichen k wird nur als Kommentar verstanden und ignoriert und kann deshalb auch weggelassen werden. Sonderzeichen wie z.B.  $\Omega$  dürfen nicht verwendet werden.

- Modell-Anweisungen

Bauelemente, die nicht nur durch einen schlichten Wert angegeben werden können (z.B. Halbleiterbauelemente), erhalten als *Wertangabe* Modell-Namen und werden durch Modell-Anweisungen näher spezifiziert. Eine Modell-Anweisung beginnt stets mit einem Punkt, gefolgt von dem Kennwort „MODEL“, also

```
.MODEL Model-Name ...
```

(siehe Abschnitt 5.3)

- Simulationsanweisungen

Mit den Simulationsanweisungen werden Art und Umfang der Analyse spezifiziert. Auch eine Simulationsanweisung beginnt stets mit einem Punkt, gefolgt von einem charakteristischen Kennbuchstaben (siehe Kapitel 7).

Wir wollen in unserem Beispiel den DC-Strom der Quelle I variieren und zwar im Bereich von  $200 \mu A$  bis  $2 mA$  in Schritten von  $100 \mu A$ . Dies bringen wir durch die Simulationsanweisung

```
.DC I 200uA 2mA 100uA;
```

kein Leerzeichen zwischen Maßzahl u. Einheit!  
zum Ausdruck ( $\mu$  ist ein unzulässiges Sonderzeichen und ist durch u zu ersetzen!).

Erläuternder Text kann, beginnend mit dem Stern (\*), eingefügt werden. Der Text nach diesem Zeichen wird dann ignoriert. In **PSPICE** und **LT-SPICE** kann auch nach einem *Semikolon* ein Kommentar stehen (dies ist in **SPICE 3f4** nicht zulässig!). Leere Zeilen werden ignoriert. Sie werden gern zur Verbesserung der Übersicht eingefügt (in allen Programmen).

Eine Circuit-Datei muss mit der „End-Anweisung“ `.END` abgeschlossen werden. Fehlt die End-Anweisung, bricht der Rechner eventuell das Programm ab. Sicherheitshalber sollte man den Cursor nach dem `.END`-Statement durch einmaliges Drücken von ENTER an den Anfang der nächsten Zeile bewegen.

Zu der obigen Schaltung kann dann das Simulations-File die folgende Form haben:

(Achtung: In **SPICE 3f4** ist die Kommentarangabe in Verbindung mit dem Semikolon nicht zulässig!)

```
* B1.CIR (Zweipol mit einer Strom- und einer Spannungsquelle)

* Element-Anweisung (Netzliste):
I   a   k   1mA
V   1   b   10V
R1  k   1   10k
R3  a   k   10k
R2  a   b   10k
R   a   b   10k
R0  a   0   10Meg ; als DC-Pfad nach GND

* Simulationsanweisung:
.DC I 200uA 2mA 100uA
.END
|
```

Nachdem die Circuit-Datei geschrieben ist, wird sie mit der Endung CIR im Verzeichnis CIRCUIT gespeichert. Für dieses Beispiel wollen wir den Namen `B1.CIR` vergeben.

## 4 Simulatoren

Nach diesen Vorbereitungen soll der Umgang mit den drei genannten Simulatoren kurz erläutert werden.

### 4.1 Simulator SPICE 3f4

**SPICE 3f4** [2] hat ein Textfenster, das aus einem einzeiligen Kommandofenster und einem getrennten Ausgabebereich besteht. **SPICE 3f4** wird kommandogesteuert: Ins Eingabefenster werden sämtliche Anweisungen, die nicht schon in der Circuit-Datei stehen, eingetippt. Durch Verwenden der Pfeiltaten  $\uparrow$  bzw.  $\downarrow$  kann man zurück bzw. vorwärts scrollen.

Im Statusfenster – ganz unten – ist erkennbar, welches File gerade aktuell ist. Auch der Fortschritt der Simulation wird angezeigt.

Wir wollen nun annehmen, dass die Circuit-Datei `B1.CIR` noch nicht existiert und wir diese mit dem in **SPICE** eingebundenen Text-Editor (z.B. Notepad, siehe **SPICE**-Startdatei) erstellen wollen.

- 1 **SPICE** durch Doppelklick auf das Symbol  starten.
- 2 In das Kommandofenster eintippen: `edit D:\CIRCUIT\B1.CIR ↵`
- 3 Erstellen der Circuit-Datei und anschließend speichern (nicht „Speichern unter ...“, weil der Name schon mit dem Edit-Kommando vorgegeben ist). Bevor das **SPICE**-Eingabefenster *Kommandos* erkennt, muss erst der Text-Editor geschlossen werden (leider). Dieser Nachteil wird durch die Möglichkeit zu scrollen aber weitgehend ausgeglichen.

- 4 Starten der Simulation im Kommandofenster mit dem Befehl: `run ↵`

Wenn die zu simulierende Circuit-Datei bereits existiert, kann auf diese verwiesen werden durch:

```
source <filename> ↵ (z.B.: source D:\CIRCUIT\B1.CIR ↵)
```

Anschließend wird die Simulation mit dem Befehl `run ↵` gestartet.

- 5 Darstellung der Simulationsergebnisse:

Grafische Darstellung der Ergebnisse mit dem Befehl „plot ...“

z.B. Potential des Knotens „b“: `plot v(b) ↵`

Möchte man den Spannungsabfall am Widerstand R1 wissen, also die Potentialdifferenz zwischen den Knoten „k“ und „1“, heißt der Befehl: `plot v(k,1) ↵`

Es können mehrere Kurvenverläufe gleichzeitig dargestellt werden. Beispiel:

```
plot v(b) v(a,b) v(k,1) ↵
```

**Ströme** durch *unabhängige Spannungsquellen* (Beispiel: Quelle V) können durch das Kommando

```
plot i(Quelle) ↵ (Beispiel: plot i(V) ↵)
```

dargestellt werden. **Hier muss i kleingeschrieben werden!** Die Richtung des Stromes zählt dabei vom positiven zum negativen Knoten der angegebenen Quelle. In Zweigen, in denen keine unabhängige Quelle enthalten ist, kann durch Einfügen eines „Strommessers“ (Spannungsquelle mit dem Wert *0 Volt*) eine „Stromdarstellung“ vorgenommen werden.

Interessiert man sich nur für einen Ausschnitt, kann dies durch die Angabe `xlimit Anfangswert Endwert` und/oder `ylimit Anfangswert Endwert` hinter dem `plot` - Befehl erfolgen. Weitere Parameter zur Darstellung sind im Manual auf Seite 72 zu finden.

Beispiel: `plot v(b,a) xlimit 400uA 600uA ylimit -1.2V -0.8V ↵`

Man kann auch verschiedene Kurvenverläufe miteinander mathematisch verknüpfen oder Berechnungen durchführen. So ist z.B. die im Widerstand  $R_1$  in Wärme umgesetzte Leistung:  $v(k,1)^2/R_1$ . Durch den Befehl `plot v(k,1)^2/10k ↵` kann dies dargestellt werden. Die wichtigsten mathematischen Operatoren und arithmetischen Ausdrücke sind im Kapitel 8 zu finden.

Wird der Befehl „plot“ durch „print“ ersetzt, erscheint das Ergebnis in Tabellenform.

Zum Speichern der Tabelle als Datei wird das *Umleitungszeichen* „>“ verwendet und dahinter der Pfad angegeben. Wir wollen für unser Beispiel den Spannungsabfall an  $R_1$  im Temp-Ordner unter dem Namen „VR1.txt“ abspeichern (D:\TEMP ist ein Ordner, der von Zeit zu Zeit aufgeräumt werden sollte). Dazu der Befehl: `plot v(k,1) > D:\TEMP\VR1.TXT ↵`.

## 4.2 Simulator PSPICE

Auch bei **PSPICE** ist die Circuit-Datei das Eingangs-File. **PSPICE** enthält zwar einen eingebauten Schaltungs-Editor, doch wollen wir darauf hier nicht weiter eingehen. Erzeugung einer Circuit-Datei:

- 1 **Text Edit** und **PSPICE** öffnen (durch Doppelklick auf die entsprechenden Symbole, siehe hierzu Abschnitt 2.5).
- 2 Öffnen der Circuit-Datei, die man simulieren will, mit dem **Text-Editor**. In unserem Beispiel ist dies die Datei `D:\CIRCUIT\B1.CIR`. Das Statement „.PROBE“ hinzufügen (siehe unten)!

Sind Änderungen vorgenommen worden, darf das Sichern nicht vergessen werden. Die Datei kann jedoch – anders als bei **SPICE 3f4** – geöffnet bleiben.

- 3 Starten der Simulation durch Aufrufen der zu simulierenden Circuit-Datei im Simulator-Fenster **PSPICE**. In unserem Beispiel ist dies die Datei `D:\CIRCUIT\B1.CIR`.

Im Statusfenster kann der Fortschritt der Simulation beobachtet werden. Am Ende der Analyse erscheint – wenn alles fehlerfrei ist – die Meldung:

Simulation completed successfully

Das Simulationsergebnis legt **PSPICE** als *Text-Datei* im Verzeichnis `CIRCUIT` mit der Endung `OUT` ab (in unserem Beispiel: `D:\CITCUIT\B1.OUT`). Den Inhalt kann man sich gleich am Ende eines Simulationslaufes durch Öffnen des Output-Files ansehen. In diesem Output-File werden auch eventuelle Probleme oder Fehler der Circuit-Datei sichtbar.

Eine weitere Datei mit *Binärdaten* und der Endung `DAT` wird ebenfalls im Verzeichnis `CIRCUIT` abgelegt. Diese wird für die grafische Darstellung unter `PROBE` benötigt. Damit diese Datei erzeugt wird, muss dies im Simulations-File angewiesen werden. Dies geschieht einfach durch Einfügen der zusätzlichen Zeile

.PROBE

in die Circuit-Datei. Obwohl diese Anweisung prinzipiell irgendwo stehen kann, empfiehlt es sich, sie immer als letzte Zeile *vor* die End-Anweisung `.End` zu setzen.

- 4 Starten des Programms `PROBE` (`PROBE` ist ein Zusatzprogramm zur grafischen Darstellung der Simulationsergebnisse und wird heute zusammen mit **PSPICE** ausgeliefert).

Trace → Add → Trace Expression: `v(b) ↵` für das Potential des Knotens „b“; oder `v(k,1)` für die Spannung am Widerstand  $R_1$ ; oder `i(R1)` für den **Strom** durch  $R_1$ . In **PSPICE** können Ströme durch beliebige Bauteile aufgerufen werden; im Gegensatz zu **SPICE 3f4** muss `i` nicht kleingeschrieben werden. Die in  $R_1$  in Wärme umgesetzte Leistung wird in `PROBE` mit

`pwr(v(k,1),2)/10k` dargestellt, vergleiche. Kapitel 8.

Wir sehen, die Befehlseingabe ist zu der von **SPICE 3f4** sehr ähnlich; einige Befehle werden nur etwas anders formuliert. Man beachte z.B. die andere Definition der Potenzfunktion, siehe Kapitel 8. Probe bietet zahlreiche Gestaltungswerkzeuge für die grafische Aufbereitung der Simulationsergebnisse.

### 4.3 Simulator LT-SPICE

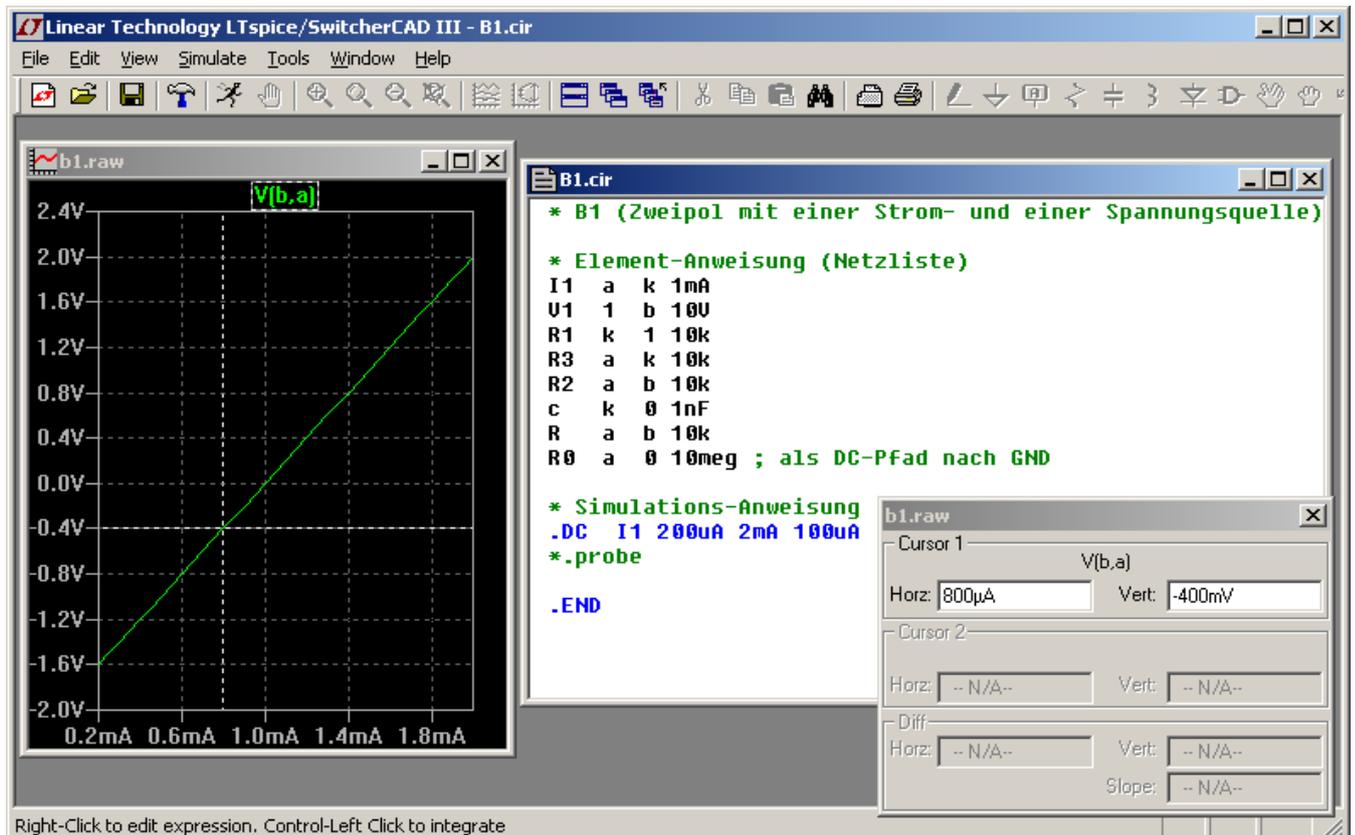
Auch **LT-SPICE** (LTspiceIV, früher scad3) hat neben dem Text-Editor zur Eingabe der Circuit-Datei einen Schaltungs-Editor. Wir wollen hier aber nur den Umgang mit der Text-Datei besprechen. Die Handhabung des Schaltungs-Editors ist sehr einfach und gut in der „eingebauten“ Hilfe erklärt (siehe aber auch Skript „Kurze Anleitung für das Simulationsprogramm LT-SPICE [14]“). Die Circuit-Datei wird, wie bei den anderen Simulatoren, mit einem beliebigen Text-Editor (z.B. Notepad) geschrieben (ohne Formatierung) und mit der Endung **CIR** im Ordner **CIRCUIT** abgespeichert. Man kann natürlich auch sofort das Programm **LT-SPICE** starten und den sehr guten eingebauten Editor verwenden. Dies wollen wir hier tun:

- 1 **LT-SPICE** (LTspiceIV, früher scad3) starten durch Doppelklick auf das Symbol  LTspice IV
- 2 Die zu simulierende Datei öffnen: **File – Open – Dateiname:**

D:\CIRCUIT\B1.CIR ↵. Wenn es die Datei noch nicht gibt, muss die Frage „Do you want to create a new netlist file?“ mit „yes“ beantwortet werden. Dann kann die Datei erstellt werden. Wenn die Datei schon existiert, kann sie einfach editiert werden.

Ein Speichern ist nicht erforderlich; dies geschieht automatisch, wenn die Simulation gestartet wird.

- 3 Simulation starten: **Simulate – Run** oder Klick auf das Symbol 
- 4 Grafische Darstellung des Ergebnisses: Nach der Simulation öffnet sich ein Plot-Fenster. Dann: **Plot Settings – Visible Traces**. Es erscheint die Aufforderung: „Select Visible Waveforms“. Man kann z.B. **v(b)** anklicken und erhält das Potential des Knotens „b“ gegen den Knoten „0“ (= GND). Man kann dann durch Klicken mit der rechten Maustaste ein Menü öffnen und „Add Trace“ anklicken. In der Kommando-Zeile kann nun eingetragen werden, was dargestellt werden soll. Wir wollen z.B. die Spannung zwischen den Knoten „b“ und „a“ sehen:  $v(b) - v(a)$ , oder kürzer:  $v(b,a)$ .



Die grafische Ergebnisdarstellung in **LT-SPICE** ist ähnlich vielseitig wie die im Programm PROBE von **PSPICE**. Auch die Schreibweise mathematischer Verknüpfungen ist mit der von **PSPICE** vergleichbar (siehe Kapitel 8). **LT-SPICE** ist außerordentlich leistungsfähig und wird ständig „gepflegt“ [6]. Nähere Hinweise sind im Skript „Kurze Anleitung für das Simulationsprogramm LT-SPICE“ zu finden [14].

Wir wollen nun ein wenig auf die SPICE-Syntax eingehen. Dies ist für ein erfolgreiches Arbeiten mit SPICE unerlässlich. Wertvolle Literaturstellen zu diesem Thema: [7] und [8].

## 5 Konventionen

**Grundsätzlich gilt: Es wird nicht zwischen Groß- und Kleinschreibung unterschieden.**

### 5.1 Zahlenformate

Zehnerpotenzen werden in SPICE wie folgt abgekürzt:

Abkürzung		Potenz	Abkürzung		Potenz
T	Tera	$10^{12}$	M	Milli	$10^{-3}$
G	Giga	$10^9$	U	Mikro	$10^{-6}$
MEG	Mega	$10^6$	N	Nano	$10^{-9}$
K	Kilo	$10^3$	P	Piko	$10^{-12}$
			F	Femto	$10^{-15}$

**Achtung:** Für Mega muss MEG (nicht M) verwendet werden. M steht für Milli !!

Eine beliebige Zehnerpotenz kann auch durch die Potenzdarstellung  $E\langle N \rangle$  bzw.  $E\langle -N \rangle$  ( $N =$  ganze Zahl) erfolgen. Beispiele:

$$3E6 = 3 \cdot 10^6 = 3MEG \quad (\text{oder } 3Meg \text{ oder } 3meg; \text{ kein Leerzeichen zwischen } 3 \text{ und Meg})$$

$$3E-6 = 3 \cdot 10^{-6} = 3U \quad (\text{das } u \text{ ist der Ersatz für das verbotene Sonderzeichen } \mu)$$

- Zwischen Zahl und Zehnerpotenz darf *kein Leerzeichen* verwendet werden! Die physikalische Einheit kann fehlen; Einheiten, die nicht zu einer Verwechslung mit Zehnerpotenzen führen, sind aber zulässig.
- Achtung bei der Einheit der Kapazität:  $F = Femto!$   $C = 2F = 2 \cdot 10^{-15} F$ ;  $C = 2MF = 2 \cdot 10^{-3} F$ .
- Statt eines Dezimal-Kommas wird ein Punkt verwendet.

### 5.2 Bauteile und deren Kennbuchstaben

Zum Aufruf eines Bauteils in der Netzliste (Element-Anweisung) wird für den ersten Buchstaben ein charakteristischer Kennbuchstabe vorgeschrieben (vergleiche auch Abschnitte 3.1 und 3.2). Danach können beliebige Ziffern oder Buchstaben folgen. Die wichtigsten Kennbuchstaben:

#### Passive Bauteile

R	Widerstand	(Resistor)
C	Kondensator	(Capacitor)
L	Spule	(Inductor)
K	gekoppelte Spulen	(Transformer)

## Quellen

- V unabhängige Spannungsquelle
- I unabhängige Stromquelle
- E spannungsgesteuerte Spannungsquelle; nicht in **SPICE3f4!**
- G spannungsgesteuerte Stromquelle; nicht in **SPICE3f4!**
- B nichtlineare Quelle (Spannungs-oder Stromquelle); nicht in **PSPICE!**
- H stromgesteuerte Spannungsquelle; nicht in **SPICE3f4!**
- F stromgesteuerte Stromquelle; nicht in **SPICE3f4!**

## Aktive Bauteile

- D Diode (Diode)
- Q Bipolar-Transistor (BJT)
- J Sperrschicht-FET (JFET)
- M MOS-FET (MOSFET)

## Teilschaltungen (Subcircuits)

- X Teilschaltung (Subcircuit)

## 5.3 Bauteiltypen und deren Kennung in der Modell-Anweisung

Ein Bauelement, das nicht nur einfach mit einem festen Wert in die Netzliste „eingebaut“ werden soll, kann durch ein „Modell“ genauer beschrieben werden. Dies gilt beispielsweise für temperaturabhängige Bauteile und besonders für Halbleiter-Bauelemente. Für die verschiedenen Bauteil-Typen sind in SPICE zur Kennung bestimmte Abkürzungen (Typ-Namen) vorgeschrieben:

### Passive Bauteile

Bauteil	engl. Bezeichnung	Kennbuchstabe (siehe Abschnitt 5.2)	Typ-Name (in der Modell-Anweisung)
Widerstand	Resistor	R	RES
Kondensator	Capacitor	C	CAP
Spule	Inductor	L	IND
Gekoppelte Spulen	Transformer Core	K	CORE

### Aktive Bauteile

Bauteil	engl. Bezeichnung	Kennbuchstabe (siehe Abschnitt 5.2)	Typ-Name (in der Modell-Anweisung)
Diode	Diode	D	D
NPN-Transistor	BJT, NPN	Q	NPN
PNP-Transistor	BJT, PNP	Q	PNP
N-Kanal J-FET	JFET, N	J	NJF

Bauteil	engl. Bezeichnung	Kennbuchstabe (siehe Abschnitt 5.2)	Typ-Name (in der Modell-Anweisung)
P-Kanal J-FET	JFET, P	J	PJF
N-Kanal MOS-FET	MOSFET, N	M	NMOS
P-Kanal MOS-FET	MOSFET, P	M	PMOS

## 6 Bauteile

Einem Bauteil kann im einfachsten Fall ein fester Wert zugewiesen werden (z.B. dem Widerstand R2 der Wert  $10\text{ k}\Omega$ ). Reicht dies zur Beschreibung nicht aus (z.B. bei Berücksichtigung der Temperaturabhängigkeit oder bei Halbleiter-Bauelementen), wird ein Bauteil durch ein „Modell“ genauer beschrieben. Es erhält dann statt eines festen Kennwertes einen Modell-Namen (dies kann ein beliebiger Name sein) und eine Modell-Beschreibung, durch die das Bauelement detailliert definiert wird. Die Syntax für eine solche Modell-Anweisung hat die Form:

```
.MODEL Model-Name [AKO: Reference-Model-Name] Typ-Name
+ (Parameter1=Wert1 Parameter2=Wert2 ... )
```

Während der Modell-Name beliebig gewählt werden darf, wird für den Typ-Namen eine festgelegte Abkürzung verwendet (siehe Abschnitt 5.3). In den Klammern ( ) werden die Modell-Parameter angegeben. Für nicht spezifizierte Parameter werden Default-Werte verwendet (siehe Abschnitte 6.1 und 6.3).

- Solch eine Parameter-Liste kann – besonders für Halbleiter-Bauelemente – länger als eine Zeile ausfallen. Jede neue Zeile muss dann, direkt am Zeilenanfang, mit dem Fortsetzungszeichen „+“ begonnen werden. Danach folgt mindestens ein Leerzeichen.
- Durch den Zusatz **AKO:** (**A Kind Of**) besteht die Möglichkeit, nur einige wenige Parameter eines bestehenden Modells zu modifizieren. Alle übrigen Parameter erhalten die Werte des Referenzmodells.

In der Simulationsdatei wird einem Bauelement, das durch ein Modell definiert wird, die Modell-Anweisung hinzugefügt.

Beispiel:

Der Widerstand R2 in **Bild 1** sei ein temperaturabhängiger Widerstand mit dem Nennwert  $10\text{ k}\Omega$ . Der Name sei Rmod. Angabe in der Simulationsdatei (stets *ganz links* beginnen):

```
R2 a b Rmod 10kOhm
.MODEL Rmod RES (R=1 TC1=1.5E-3 TC2=5E-6)
```

Der „Typ-Name“ für den Widerstand (Resistor) heißt „RES“ (siehe Abschnitt 5.3)

Die Modell-Definitionen können direkt in die Circuit-Datei geschrieben werden. Sollen solche Bauteile aber auch in anderen Simulationsaufgaben Verwendung finden, werden deren Modelle zweckmäßigerweise in einer „Modell-Bibliothek“ (z.B. `D:\SPICELIB\...`) „gesammelt“, vergl. **Abschnitt 2.3**. Im Simulations-File braucht dann nur auf die entsprechende „Bibliothek“ verwiesen zu werden, etwa durch das „Include-Statement“:

```
.INCLUDE D:\SPICELIB\...
```

**PSPICE** und **LT-SPICE** bieten hierzu auch die Möglichkeit, dies mit der speziellen „Bibliotheks-Anweisung“ vorzunehmen:

```
.LIB D:\SPICELIB\...
```

Der Unterschied ist, dass ein Bauteil erst dann in der angegebenen Bibliothek „gesucht“ wird, wenn seine Beschreibung nicht im Simulations-File steht. Durch die „Include-Anweisung“ erhalten dagegen die in der Bibli-

othek angegebenen Definitionen die gleiche Priorität wie solche, die direkt in der Circuit-Datei stehen. Dies kann zu Problemen führen: Wenn nämlich derselbe Modell-Name sowohl in der Circuit-Datei als auch in der angegebenen Bibliothek auftaucht, führt dies zu einer Doppelanweisung und damit zu einer Fehlermeldung. Doppelanweisungen sind unzulässig! Dieses Problem tritt beim Verwenden der LIB-Anweisung nicht auf, weil die durch sie angegebene Bibliothek die „zweite Priorität“ hat.

Die folgenden Angaben gelten vorzugsweise für **PSPICE** und **LT-SPICE**. In **SPICE 3f4** werden einige Bauelemente etwas anders beschrieben (siehe Manual). Halbleiter-Bauelemente werden in allen drei Simulationsprogrammen überwiegend durch dieselben Modell-Parameter definiert.

## 6.1 Passive Bauteile

### 6.1.1 Widerstand (RES)

Im einfachsten Fall genügt für einen Widerstand die Angabe eines festen Wertes. Beispiel:

```
R2 a b 10k
```

Der Wert eines *temperaturabhängigen* Widerstandes (z.B. R2) wird in **PSPICE** und **LT-SPICE** durch die Modell-Gleichung beschrieben (Achtung: Temperaturangaben in °C):

$$R_2 = \text{Nennwert} \cdot R \cdot (1 + TC1 \cdot (T - Tnom) + TC2 \cdot (T - Tnom)^2).$$

Modell-Form:

```
.MODEL Modell-Name RES (Parameter1 = Wert1 ... )
```

Parameter	Bedeutung	Einheit	Default
$R$	Widerstands-Multiplikator	1	1
$TC1$	linearer Temperaturkoeffizient	$K^{-1}$	0
$TC2$	quadratischer Temperaturkoeffizient	$K^{-2}$	0

Für nicht spezifizierte Parameter werden automatisch die Default-Werte verwendet.

Die Nenntemperatur  $Tnom$  kann durch eine Options-Anweisung (siehe auch Kapitel 9) angegeben werden, z.B.:

```
.OPTIONS TNOM = 20
```

- Wichtig ist, dass dann die Parameter aller temperaturabhängigen Bauelemente einer Schaltung für *dieselbe Nenntemperatur* gelten. Bei Parametersätzen mit unterschiedlichen Nenntemperaturen müssen alle Parameter auf eine einheitliche Nenntemperatur umgerechnet werden (siehe auch Abschnitt 9.2)! Einfacher:
- Die Nenntemperatur kann aber auch als individueller Bauteil-Parameter mit ins Modell aufgenommen werden, z.B. durch die Angabe „TNOM = 20“. Dann muss die Options-Anweisung entfallen!
- In aller Regel werden die Modell-Parameter bei einer Temperatur von  $\vartheta = 27^\circ C (\triangleq 300 K)$  ermittelt. Dann ist  $TNOM = 27$  und braucht dann nicht angegeben zu werden; dies ist der Default-Wert.

Beispiel:

In der Schaltung nach **Bild 1** soll R2 ein temperaturabhängiger Widerstand sein. Bei der Nenntemperatur von beispielsweise  $\vartheta = 20^\circ C (TNOM = 20^\circ C)$  sei der Wert  $10 k\Omega$ . Als Modell-Name werde Rmod verwendet (ein beliebiger Name ist möglich). Die Zeile in der Netzliste lautet dann:

```
R2 a b Rmod 10kOhm
```

Das zugehörige Modell sei

```
.MODEL Rmod RES (R=1 TC1=1.5E-3 TC2=5E-6 TNOM=20)
```

### 6.1.2 Kondensator (CAP)

Im einfachsten Fall wird einem Kondensator ein fester Wert zugewiesen. Beispiel:

```
C5 2 3 10nF
```

In **PSPICE** und **LT-SPICE** kann aber ein Kondensator (z.B. C5) auch durch die folgende Modell-Gleichung beschrieben werden:

$$C_5 = \text{Nennwert} \cdot C \cdot (1 + TC1 \cdot (T - Tnom) + TC2 \cdot (T - Tnom)^2) \cdot (1 + VC1 \cdot V_{CAP} + VC2 \cdot V_{CAP}^2)$$

Der zweite Klammerausdruck beschreibt die Spannungsabhängigkeit der Kapazität des Kondensators.  $V_{CAP}$  ist die Spannung am Kondensator C5, gerichtet vom ersten Knoten zum zweiten Knoten.

Modell-Form:

```
.MODEL Modell-Name CAP (Parameter1 = Wert1 ... )
```

Parameter	Bedeutung	Einheit	Default
$C$	Kapazitäts-Multiplikator	1	1
$TC1$	linearer Temperaturkoeffizient	$K^{-1}$	0
$TC2$	quadratischer Temperaturkoeffizient	$K^{-2}$	0
$VC1$	lineare Spannungsabhängigkeit	$Volt^{-1}$	0
$VC2$	quadratische Spannungsabhängigkeit	$Volt^{-2}$	0

Für nicht spezifizierte Parameter werden automatisch die Default-Werte verwendet.

Die Angabe der Nenntemperatur  $TNOM$  erfolgt – wie beim Widerstand – durch die Options-Anweisung oder sie kann direkt im Modell als zusätzlicher Parameter stehen (vergleiche Abschnitt 6.1.1).

Beispiel:

```
C5 2 3 Cmod 10pF
.MODEL Cmod CAP (C=1 TC1=3E-5 TNOM=22)
```

### 6.1.3 Spule (IND)

Im einfachsten Fall wird einer Spule ein fester Wert zugewiesen. Beispiel:

```
L2 2 5 1.2mH
```

In **PSPICE** und **LTSPICE** kann eine Spule (z.B. L2) durch die folgende Modell-Gleichung beschrieben werden:

$$L_2 = \text{Nennwert} \cdot L \cdot (1 + TC1 \cdot (T - Tnom) + TC2 \cdot (T - Tnom)^2) \cdot (1 + IL1 \cdot I_{IND} + IL2 \cdot I_{IND}^2)$$

Der zweite Klammerausdruck beschreibt die Stromabhängigkeit der Induktivität.  $I_{IND}$  ist der Strom durch die Spule L2, gerichtet vom ersten Knoten zum zweiten Knoten.

Modell-Form:

```
.MODEL Modell-Name IND (Parameter1 = Wert1 ...)
```

Parameter	Bedeutung	Einheit	Default
$L$	Induktivitäts-Multiplikator	1	1
$TC1$	linearer Temperaturkoeffizient	$K^{-1}$	0
$TC2$	quadratischer Temperaturkoeffizient	$K^{-2}$	0

Parameter	Bedeutung	Einheit	Default
<i>IL1</i>	lineare Stromabhängigkeit	<i>Amp</i> <sup>-1</sup>	0
<i>IL2</i>	quadratische Stromabhängigkeit	<i>Amp</i> <sup>-2</sup>	0

Für nicht spezifizierte Parameter werden automatisch die Default-Werte verwendet.

Die Angabe der Nenntemperatur *TNOM* erfolgt – wie beim Widerstand – durch die Options-Anweisung oder sie kann direkt im Modell als zusätzlicher Parameter stehen (vergleiche Abschnitt 6.1.1).

Beispiel:

```
L5 2 3 Lmod 10mH
.MODEL Lmod IND (L=1 TC1=3E-5 TNOM=22)
```

#### 6.1.4 Gekoppelte Spulen (CORE)

Bei einem Spulensystem, bestehend aus den gekoppelten Spulen  $L_1, L_2, \dots, L_i, \dots$ , mit den Induktivitätswerten  $L_1, L_2, \dots, L_i, \dots$ , wird die Spannung der Spule  $L_i$  durch die Gleichung beschrieben:

$$V_i = L_i \frac{dI_i}{dt} + M_{ij} \frac{dI_j}{dt} + M_{ik} \frac{dI_k}{dt} \dots$$

In der Netzliste wird deshalb zunächst für jede Spule eine Netzlisten-Zeile vorgesehen. Außerdem müssen sämtliche Kopplungen zwischen den einzelnen Spulen angegeben werden. Diese Beschreibung gilt auch für **SPICE 3f4**. Ein kleines Beispiel möge dies verdeutlichen. Der erläuternde Text in Verbindung mit dem Semikolon ist in **SPICE 3f4** unzulässig!

```
L1 1 0 10mH ; Primärspule
L2 2 0 10mH ; Sekundärspule 1 (Windungszahl wie L1)
L3 3 4 40mH ; Sekundärspule 2 (Windungszahl doppelt wie L1)
R2 2 0 600ohm ; Belastungswiderstand der Spule L2
R3 3 4 1000ohm ; Belastungswiderstand der Spule L3
R0 3 0 100Meg ; DC-Pfad zum Knoten 0 (nicht vergessen!!!)
K12 L1 L2 0.99 ; Kopplung zwischen L1 und L2
K13 L1 L3 0.99 ; Kopplung zwischen L1 und L3
K23 L2 L3 0.99 ; Kopplung zwischen L2 und L3
```

Durch die Zeile `K12 ...` wird beispielsweise die Kopplung zwischen den Spulen  $L_1$  und  $L_2$  zum Ausdruck gebracht. Die Kopplung (der Kopplungsfaktor) beträgt hier  $0.99$ .

**Achtung:** Die Sekundärspule  $L_3$  muss einen DC-Pfad nach GND haben (z.B. 100Meg zum Knoten „0“)!!

Der Kopplungsfaktor ist durch die Beziehung

$$\text{Kopplung} = \frac{M_{ij}}{\sqrt{L_i L_j}} ; 0 \leq \text{Kopplung} \leq 1$$

definiert.  $M_{ij}$  ist die Gegeninduktivität der beiden gekoppelten Spulen  $L_i$  und  $L_j$ .

In **PSpICE** und **LT-SPICE** können die Zeilen, die die Kopplung beschreiben, auch durch eine einzige Zeile angegeben werden, wenn alle Kopplungen den gleichen Wert haben:

```
K123 L1 L2 L3 0.99
```

In **PSpICE** kann man einen Eisenkern auch durch ein Modell beschreiben. Die Modell-Parameter können aus der Magnetisierungskurve gewonnen werden. Die Modell-Anweisung lautet:

```
.MODEL Modell-Name CORE (Parameter1 = Wert1 ... )
```

Parameter	Bedeutung	Einheit	Default
<i>AREA</i>	mittlere magnetische Querschnittsfläche	<i>cm</i> <sup>2</sup>	0.1
<i>PATH</i>	mittlere magnetische Eisenlänge	<i>cm</i>	1
<i>GAP</i>	effektive Luftspaldlänge	<i>cm</i>	0
<i>PACK</i>	Füll- oder Schichtungsfaktor	1	1
<i>MS</i>	Sättigungsmagnetisierung	<i>A/m</i>	1E+6
<i>A</i>	Thermischer Energie-Parameter	<i>A/m</i>	1E+3
<i>C</i>	Domänen-Beweglichkeit	1	0.2
<i>K</i>	Domänen-Anisotropie	<i>A/m</i>	500

Für nicht spezifizierte Parameter werden automatisch die Default-Werte verwendet.

### Achtung:

Wenn ein Trafo durch ein solches Modell beschrieben wird, ändert sich folgendes:

- Die gekoppelten Spulen werden *nichtlinear*,
- Der „Wert“ der einzelnen Spulen wird nicht mehr in Henry angegeben. Stattdessen gibt man die Windungszahl an. Hierauf ist besonders zu achten!
- Die Kopplungen der Spulen werden nur durch eine Zeile angegeben (siehe folgendes Beispiel).
- Eine Modell-Anweisung ist erforderlich.

Beispiel, z.Z. nur für **PSPICE** (in **LT-SPICE** wird es voraussichtlich bald ein ähnliches Modell geben):

```

I1  1  0  AC  1mA
L1  1  0  44 ; Windungen
L2  2  0  44 ; Windungen (L2 = L1)
L3  3  0  88 ; Windungen (L3 = 4·L1; L ~ N2)
R2  2  0  600ohm
R3  3  0  1000ohm
K123 L1 L2 L3 0.99 KMOD
.MODEL KMOD CORE (AREA=1.2  PATH=4.3  GAP=0.01)

```

## 6.2 Quellen

### 6.2.1 Unabhängige Spannungsquelle (V)

Die wichtigsten unabhängigen Spannungsquellen:

- **Gleichspannung** (DC-Wert)

Beispiel:

Vbias 2 0 2.3mV ; Gleichspannung zwischen den Knoten 2 und 0  
oder:

```
Vbias 2 0 DC 2.3mV
```

- **Wechselspannung** (sinusförmig, für die AC-Analyse; AC-Effektivwert)

Beispiel:

`Vin 3 4 AC 1mV ; Sinusform, Effektivwert 1 mV`

Die Frequenz wird durch die AC-Analyse-Anweisung `.AC` (siehe Abschnitt 7.2) vorgegeben.

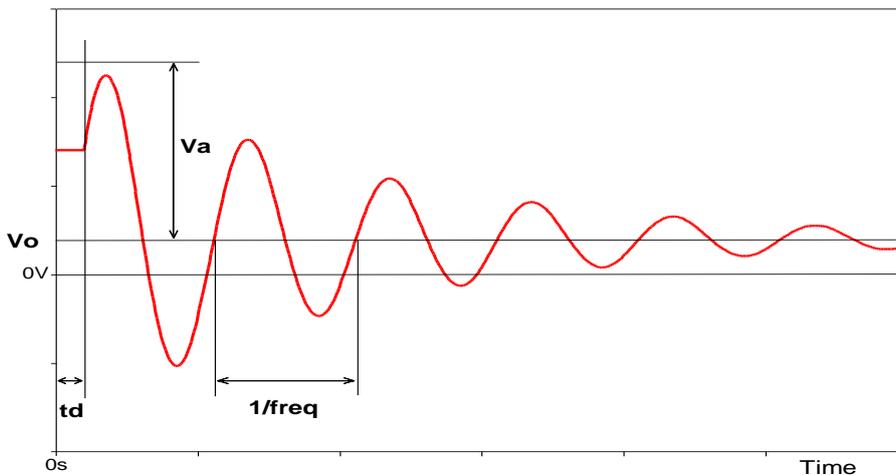
- **Sinusform** (für die Analyse im Zeitbereich; die Sinusform wird detailliert beschrieben):

`SIN (Vo Va freq td df phase)`

SIN (Parameter)	Bedeutung	Einheit	Default
$V_o$	Offsetspannung	<i>Volt</i>	Wert <sup>1)</sup>
$V_a$	Amplitude	<i>Volt</i>	Wert <sup>1)</sup>
$freq$	Frequenz	<i>Hz</i>	$1/T_{stop}$ <sup>2)</sup>
$t_d$	Delay-Zeit	<i>sec</i>	0
$t_f$	Dämpfungsfaktor	<i>sec<sup>-1</sup></i>	0
$phase$	Phasenlage	Grad (PSPICE, LT-SPICE) Bogenmaß (SPICE 3f4)	0 0

<sup>1)</sup> Für  $V_o$  und  $V_a$  müssen Werte angegeben werden!

<sup>2)</sup>  $T_{stop}$  ist die Zeit, die bei der Transienten-Analyse als Stoppzeit angegeben wird (Abschnitt 7.4)



**Bild 2**

Zur Erklärung der Argumente einer Quelle mit sinusförmiger Zeitfunktion

Beispiel:

`Vin 3 4 SIN(0.2V 1.8V 3kHz)`

- **Pulsform** (für die Analyse im Zeitbereich)

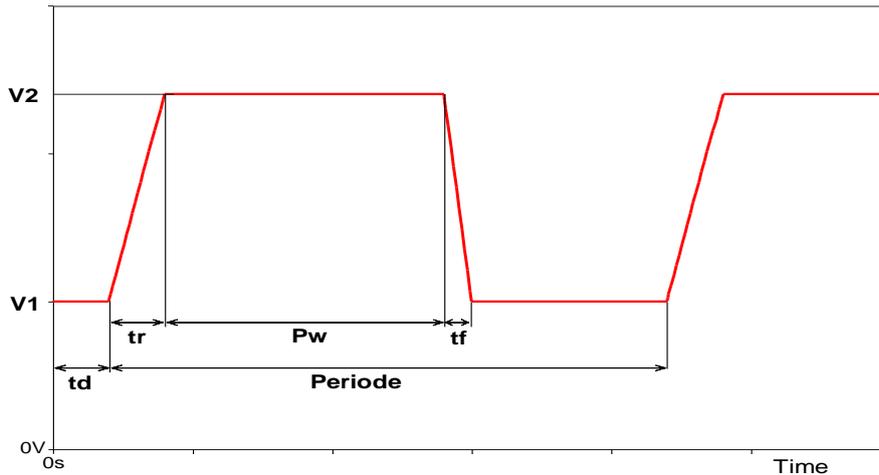
`PULSE (V1 V2 td tr tf Pw Periode)`

PULSE (Parameter)	Bedeutung	Einheit	Default
$V_1$	Anfangswert	<i>Volt</i>	Wert <sup>1)</sup>
$V_2$	Spitzenwert	<i>Volt</i>	Wert <sup>1)</sup>
$t_d$	Delay-Zeit	<i>sec</i>	0

$t_r$	Anstiegszeit (rise time)	sec	$dT_{max}^{2)}$
$t_f$	Abfallzeit (fall time)	sec	$dT_{max}^{2)}$
$P_w$	Pulsweite	sec	$T_{stop}^{2)}$
<i>Periode</i>	Periode	sec	$T_{stop}^{2)}$

<sup>1)</sup> Für V1 und V2 müssen Werte angegeben werden!

<sup>2)</sup>  $T_{stop}$  ist die Zeit, die bei der *Transienten*-Analyse als Stoppzeit (Analyse-Zeitraum) angegeben wird und  $dT_{max}$  ist die Schrittweite. Näheres wird in den Abschnitten 7.4 bis 7.6 besprochen. Normalerweise werden alle Werte angegeben; bei LT-SPICE reicht meist  $T_{stop}$  allein.



**Bild 3**

Zur Erklärung der Argumente einer Quelle mit pulsförmiger Zeitfunktion

Beispiel:

```
Vin 3 4 PULSE(0.2V 1.8V 1us 1us 500ns 5us 10us)
```

### 6.2.2 Unabhängige Stromquelle (I)

Unabhängige Stromquellen werden genauso beschrieben wie unabhängige Spannungsquellen; nur der Kennbuchstabe ist I (statt V).

### 6.2.3 Spannungsgesteuerte Spannungsquelle (E)

Die spannungsgesteuerte Spannungsquelle (Kennbuchstabe E) ist ein sehr wichtiges Element, das auch für die Formulierung eigener Modelle (in Form von Subcircuits) verwendet werden kann. Allgemeine Form:

```
Ename +Knoten -Knoten +Input,-Input Spannungsverstärkung
```

Beispiel:

```
E1 2 0 10,11 100
```

In diesem Beispiel ist E1 eine Spannungsquelle zwischen den Knoten „2“ und „0“. Der Wert der Spannung ist gleich der Spannung zwischen den beiden Knoten „10“ und „11“, multipliziert mit der Verstärkung 100.

In **PSPICE** und **LT-SPICE** ist auch die folgende Formulierung einer spannungsgesteuerten Spannungsquelle möglich:

```
Ename +Knoten -Knoten VALUE = {mathematischer Ausdruck}
```

Dies ist ein sehr wertvolles Instrument für die Bildung von neuen Bauteil-Modellen.

Beispiel:

```
E3 2 0 VALUE = { (PWR (V (10, 11) , 2) ) / V (12) }
```

In diesem Beispiel ist E3 eine Spannungsquelle zwischen den Knoten „2“ und „0“. Der Wert wird durch Quadrieren der Spannung zwischen den Knoten „10“ und „11“ und Dividieren durch das Potential des Knotens „12“ gebildet (mathematische Operationen siehe Kapitel 8).

**Achtung:** Nicht die geschweiften Klammern { } vergessen!

Es sind noch weitere Formulierungen einer spannungsgesteuerten Spannungsquelle möglich. Hier sei jedoch auf die Handbücher oder „eingebauten“ Hilfen verwiesen ([5] oder [6]).

## 6.2.4 Spannungsgesteuerte Stromquelle (G)

Spannungsgesteuerte Stromquellen sind in ihrer Beschreibung vergleichbar mit spannungsgesteuerten Spannungsquellen; nur der Kennbuchstabe ist G (statt E) und es ist eine Stromquelle.

## 6.2.5 Nichtlineare Quelle (Spannungs- oder Stromquelle) in **SPICE 3f4** und **LT-SPICE (B)**

In **SPICE 3f4** und **LT-SPICE** gibt es eine besondere nichtlineare gesteuerte „B-Quelle“:

```
Bname +Knoten -Knoten V = mathematischer Ausdruck
```

bzw.:

```
Bname +Knoten -Knoten I = mathematischer Ausdruck
```

Durch  $V =$  bzw.  $I =$  wird B als Spannungs- bzw. Stromquelle definiert.

Beispiel:

```
B3 0 2 I = cos (V (1) ) + sin (V (5) )
```

Hier ist B3 eine *Stromquelle*, deren Wert von den Spannungen  $V(1)$  und  $V(5)$  abhängt.

Mögliche mathematische Operationen (für reelle Variablen) siehe Kapitel 8.

## 6.2.6 Stromgesteuerte Spannungsquelle (H)

Allgemeine Form einer stromgesteuerten Spannungsquelle (Kennbuchstabe ist H):

```
Hname +Knoten -Knoten Vname Widerstand
```

Beispiel:

```
H1 2 0 Vz 0.5kOhm
```

In diesem Beispiel ist H1 eine Spannungsquelle zwischen den Knoten „2“ und „0“. Vz ist der Name einer Spannungsquelle, durch die der Kontrollstrom fließt. Der Wert der Spannung ist gleich dem Strom durch die Spannungsquelle Vz, multipliziert mit dem Widerstand  $0.5\text{ k}\Omega$ . Die Spannungsquelle Vz (allgemein: Vname) muss eine *unabhängige* Quelle sein, deren DC-Wert auch null sein darf (Strommesser).

## 6.2.7 Stromgesteuerte Stromquelle (F)

Allgemeine Form einer stromgesteuerten Stromquelle (Kennbuchstabe ist F):

```
Fname +Knoten -Knoten Vname Stromverstärkung
```

Beispiel:

```
F1 2 0 Vz 100
```

In diesem Beispiel ist F1 eine Stromquelle zwischen den Knoten „2“ und „0“. Vz ist der Name einer Spannungsquelle, durch die der Kontrollstrom fließt. Der Wert des Stromes ist gleich dem Strom durch die Spannungsquelle Vz, multipliziert mit der Stromverstärkung 100. Die Spannungsquelle Vz (allgemein: Vname) muss eine *unabhängige* Quelle sein, deren DC-Wert auch null sein darf (Strommesser).

## 6.3 Aktive Bauteile

### 6.3.1 Diode (D)

Modell-Form:

```
.MODEL Modell-Name D (Parameter1 = Wert1 ... )
```

Parameter	Bedeutung	Einheit	Default
<i>IS</i>	Sättigungsstrom	A	1E-14
<i>N</i>	Emissions-Koeffizient	1	1
<i>BV</i>	Sperrspannung	<i>Volt</i>	$\infty$
<i>TNOM</i>	Temperatur, bei der die Parameter bestimmt wurden	$^{\circ}\text{C}$	27

Die Tabelle ist nicht vollständig; weitere Modell-Parameter sind im Manual zu finden. Die Werte stellt gewöhnlich der Dioden-Hersteller zur Verfügung. Für nicht angegebene Modell-Parameter werden automatisch die Default-Werte gesetzt.

Einbau in die Netzliste:

```
Dname +Knoten -Knoten Modell-Name [AREA]
```

+Knoten ist der Anodenanschluss der Diode (P-Gebiet). Durch *AREA* können die flächenabhängigen Parameter wie z.B. *IS* skaliert werden. *AREA* ist nicht die Fläche, sondern eine einheitenlose Zahl. Wird *AREA* nicht angegeben, wird mit dem Default-Wert 1 gerechnet.

Beispiel:

```
D3 4 6 D1N4148 2
.MODEL D1N4148 D (IS=0.1pA N=1.6 BV=100V)
```

Die Diode (Modell-Name oder Typ D1N4148) liegt zwischen den Knoten „4“ und „6“. Hier ist *AREA* gleich 2 gesetzt. Dies bedeutet, dass *zwei* Dioden des Typs D1N4148 parallelgeschaltet sind.

### 6.3.2 Bipolar-Transistor (NPN bzw. PNP)

Modell-Form:

```
.MODEL Modell-Name NPN (Parameter1 = Wert1 ... )
```

bzw.:

```
.MODEL Modell-Name PNP (Parameter1 = Wert1 ... )
```

Parameter	Bedeutung	Einheit	Default
$I_S$	Sättigungsstrom	A	1E-16
$N_F$	Emissions-Koeffizient	1	1
$B_F$	Stromverstärkung	1	100
$V_{AF}$	Early-Spannung	V	$\infty$
$T_{NOM}$	Temperatur, bei der die Parameter bestimmt wurden	°C	27

Die Tabelle ist nicht vollständig; weitere Modell-Parameter sind im Manual zu finden. Die Werte stellt gewöhnlich der Transistor-Hersteller zur Verfügung. Für nicht angegebene Modell-Parameter werden automatisch die Default-Werte gesetzt.

Einbau in die Netzliste:

```
Qname Kollektor Basis Emitter [Substrat] Modell-Name [AREA]
```

Ein Bipolar-Transistor ist oft ein drei-poliges Element mit den Anschlüssen „Kollektor“, „Basis“ und „Emitter“. Bei integrierten Transistoren kommt noch ein vierter Anschluss, der Substrat-Anschluss, hinzu. In genau dieser Reihenfolge müssen die Transistoranschlüsse in der Netzliste eingetragen werden. Wird der Substrat-Anschluss nicht angegeben, wird er defaultmäßig auf „0“ gelegt. Dies kann zu einem Problem führen, wenn das Substrat in einer Anwendung an ein anderes Potential, z.B. die negative Versorgungsspannung, angeschlossen wird. In solchen Fällen ist unbedingt der Substrat-Knoten richtig anzugeben! Der Parameter *AREA* dient, wie bei der Diode, zur Skalierung der flächenabhängigen Parameter.

Beispiel:

```
Q3 4 6 7 C3ZN02 8
.MODEL C3ZN02 NPN (IS=1.1E-16 NF=1.01 BF=190 VAF=105V)
```

In diesem Beispiel besteht der NPN-Transistor Q3 aus acht parallelgeschalteten Transistoren des Typs C3ZN02. Der Kollektor ist mit dem Knoten „4“, die Basis mit dem Knoten „6“, der Emitter mit dem Knoten „7“ und der Substrat-Anschluss mit dem Knoten „0“ verbunden (defaultmäßig, da nicht angegeben).

### 6.3.3 Sperrschicht-FET (NJF bzw. PJF)

Modell-Form:

```
.MODEL Modell-Name NJF (Parameter1 = Wert1 ... )
```

bzw.:

```
.MODEL Modell-Name PJF (Parameter1 = Wert1 ... )
```

Parameter	Bedeutung	Einheit	Default
$V_{TO}$ (nicht $V_{T0}$ )	Threshold-Spannung	V	-2 *)
$BETA$	Transconductance-Parameter = $(K_P/2) \cdot (W/L)$	$A/V^2$	1E-4
$LAMBDA$	Kanallängen-Modulation ( $\triangleq 1/\text{Early-Spannung}$ )	$V^{-1}$	0
$T_{NOM}$	Temperatur, bei der die Parameter bestimmt wurden	°C	27

\*) Achtung:  $V_{TO} < 0$ . (für N-Kanal-J-FET *und* P-Kanal-J-FET) Dies ist in SPICE so vereinbart, obwohl  $V_{TO}$  physikalisch bei einem N-Kanal-J-FET negativ und bei einem P-Kanal-J-FET positiv ist.

Die Tabelle ist nicht vollständig; weitere Modell-Parameter sind im Manual zu finden. Die Werte stellt gewöhnlich der Transistor-Hersteller zur Verfügung. Für nicht angegebene Modell-Parameter werden automatisch die Default-Werte gesetzt.

Einbau in die Netzliste:

```
Jname Drain Gate Source Modell-Name [AREA]
```

Ein Sperrschicht-FET ist ein dreipoliges Element mit den Anschlüssen „Drain“, „Gate“ und „Source“. In genau dieser Reihenfolge müssen die Transistoranschlüsse in der Netzliste eingetragen werden. *AREA* dient, wie bei der Diode, zur Skalierung der flächenabhängigen Parameter.

Beispiel:

```
J3 4 6 7 BF245 2
.MODEL BF245 NJF (VTO=-1.2V BETA=1.8E-3 LAMBDA=0.008)
```

In diesem Beispiel besteht der N-Kanal-J-FET J3 aus zwei parallelgeschalteten Transistoren des Typs BF245. Drain ist mit dem Knoten „4“, Gate mit dem Knoten „6“ und Source mit dem Knoten „7“ verbunden.

### 6.3.4 MOS-FET (NMOS bzw. PMOS)

Modell-Form:

```
.MODEL Modell-Name NMOS (Parameter1 = Wert1 ... )
```

bzw.:

```
.MODEL Modell-Name PMOS (Parameter1 = Wert1 ... )
```

Parameter	Bedeutung	Einheit	Default
$W, L$ <sup>1)</sup>	Weite des Kanals, Länge des Kanals	<i>m</i>	100E-6
$M$ <sup>1)</sup>	Zahl der parallelgeschalteten Elemente	1	1
$AD, AS$ <sup>1)</sup>	Drain-Fläche, Source-Fläche	<i>m</i> <sup>2</sup>	0
$PD, PS$ <sup>1)</sup>	Drain-Umfang, Source-Umfang (Perimeter)	<i>m</i>	0
$VTO$ (nicht $VT0$ )	Threshold-Spannung für $V_{SB} = 0$	<i>V</i>	0 <sup>2)</sup>
$KP$	Transconductance-Parameter	<i>A/V</i> <sup>2</sup>	2E-5
$LAMBDA$	Kanallängen-Modulation	<i>V</i> <sup>1</sup>	0
$TNOM$	Temperatur, bei der die Parameter bestimmt wurden	°C	27

<sup>1)</sup> Designmaße werden stets in die Element-Anweisung geschrieben (siehe unten) und nicht ins Modell.

<sup>2)</sup> Anreicherungstypen: N-Kanal:  $VTO > 0$ , P-Kanal:  $VTO < 0$  (bei  $V_{GS} = 0$  ist der Kanal gesperrt)

<sup>3)</sup> Verarmungstypen: N-Kanal:  $VTO < 0$ , P-Kanal:  $VTO > 0$  (bei  $V_{GS} = 0$  ist der Kanal leitend)

Weitere Modell-Parameter sind im Manual zu finden. Die Werte stellt gewöhnlich der Transistor-Hersteller zur Verfügung. Für nicht angegebene Modell-Parameter werden automatisch die Default-Werte gesetzt.

Einbau in die Netzliste:

```
Mname Drain Gate Source Bulk Modell-Name [W=Wert] [L=Wert] [M=Wert]
+ [AD=Wert] [AS=Wert] [PD=Wert] [PS=Wert]
```

Ein MOS-FET ist stets ein vier-poliges Element mit den Anschlüssen „Drain“, „Gate“, „Source“ und „Bulk“ (Substrat). In genau dieser Reihenfolge müssen die Transistoranschlüsse in der Netzliste eingetragen werden.

**Designmaße**, wie die Kanalweite  $W$  und die Kanallänge  $L$  und die Angaben  $AD$  und  $AS$  (Fläche des Drain- bzw. des Source-Gebietes), sowie  $PD$  und  $PS$  (Umfang oder Perimeter des Drain- bzw. des Source-Gebietes) werden direkt in die Element-Anweisung geschrieben.  $M$  gibt die Zahl der parallelgeschalteten Transistoren an.  $M$  ist vergleichbar mit der  $AREA$ -Angabe beim Bipolar-Transistor, nur heißt es:  $M=Wert$ .

Beispiel:

```
M3 4 6 7 0 MN7 W=60u L=10u AD=120p PD=124u M=4
.MODEL MN7 NMOS (VTO=0.84V KP=80E-6 LAMBDA=0.02)
```

In diesem Beispiel wird der Transistor M3 durch das Modell MN7 näher beschrieben. Es handelt sich um einen N-Kanal-MOS-FET. Drain ist mit dem Knoten „4“, Gate mit dem Knoten „6“, Source mit dem Knoten „7“ und Bulk mit dem Knoten „0“ verbunden.

Der Transistor M3 hat eine Kanal-Weite von  $W = 60 \mu\text{m}$  und eine Kanal-Länge von  $L = 10 \mu\text{m}$ . Die Drain-Fläche beträgt  $AD = 120 \cdot 10^{-12} \text{ m}^2$  und die Randlänge des Drain-Gebietes ist  $PD = 124 \mu\text{m}$ . Es sind  $M = 4$  solche Elemente parallelgeschaltet.

## 6.4 Teilschaltungen (Subcircuits)

Wird ein Schaltungsteil, bestehend aus mehreren Bauteilen, in einer größeren Schaltung mehrfach benutzt, ist es sicher sinnvoll, eine solche Teilschaltung als Block zusammenzufassen. Dies ist in SPICE möglich. Eine Teilschaltung beginnt mit dem Statement `.SUBCKT`, erhält einen beliebigen Namen, gefolgt von den Knoten der Teilschaltung. Die Bauteile der Schaltung werden in Form einer Netzliste aufgeführt. Am Ende muss das Statement `.ENDS` stehen (S steht für Subcircuit und darf nicht vergessen werden).

Definition einer Teilschaltung:

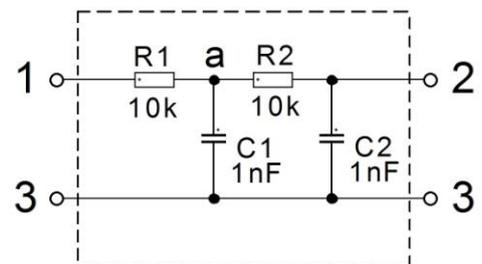
```
.SUBCKT Name Knoten1 Knoten2 Knoten3 ...
```

Netzliste der Teilschaltung

```
.ENDS
```

Beispiel TP-A, vergleiche Bild 4:

```
.SUBCKT TP-A 1 2 3
R1 1 a 10k
R2 a 2 10k
C1 a 3 1nF
C2 2 3 1nF
.ENDS
```



**Bild 4**

Innenschaltung der Teilschaltung TP-A

Diese Teilschaltung mit dem Namen TP-A hat nach *außen* die drei Anschlüsse „1“, „2“ und „3“. Im Inneren ist sie entsprechend der obigen Netzliste (oder Schaltung) aufgebaut.

Diese Teilschaltung kann nun wie ein neues Bauelement in eine größere Schaltung eingebunden werden. Es ist nur darauf zu achten, dass *jede* Teilschaltung mit dem Kennbuchstaben X beginnt. Außerdem muss jeder Anschluss der Teilschaltung entsprechend der obigen Definition verwendet werden und zwar in der angegebenen Reihenfolge.

Ein Beispiel soll dies deutlich machen:

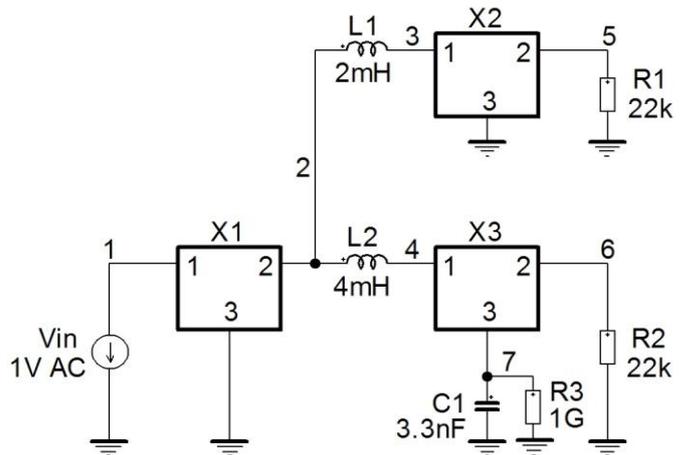
Die obige Teilschaltung mit dem Namen TP-A soll z.B. an drei Stellen in einer größeren Schaltung mit dem Namen Filter-A eingesetzt werden. Das komplette Simulations-File (Circuit-Datei) könnte dann etwa folgendermaßen aussehen:

```
Filter-A.cir
```

```
Vin 1 0 AC 1V
L1 2 3 2mH
L2 2 4 4mH
R1 5 0 22k
R2 6 0 22k
R3 7 0 1G; DC-Pfad nach GND
C1 7 0 3.3nF
X1 1 2 0 TP-A
X2 3 5 0 TP-A
X3 4 6 7 TP-A

.SUBCKT TP-A 1 2 3
R1 1 a 10k
R2 a 2 10k
C1 a 3 1nF
C2 2 3 1nF
.ENDS

.AC Dec 20 10Hz 100kHz
.PROBE
.END
```



**Bild 5**

Beispiel einer Schaltung mit Teilschaltungen

- Teilschaltungen (Subcircuits) können, ähnlich wie Modelle anderer Bauteile, in einem Bauteil-Verzeichnis (z.B. D:\SPICELIB\Beispiel.lib) „gesammelt“ werden. Dies ist besonders dann sinnvoll, wenn die Teilschaltung in verschiedenen Schaltungen Verwendung finden sollen. In dem Simulations-File kann dann mit Hilfe des Include-Statements die entsprechende „Bibliothek“ mit eingebunden werden. Die Anweisung lautet dann:

```
.INCLUDE D:\SPICELIB\Beispiel.lib
```

**PSPICE** und **LT-SPICE** bieten hierzu auch die Möglichkeit, dies mit der Bibliotheks-Anweisung vorzunehmen:

```
.LIB D:\SPICELIB\Beispiel.lib
```

Der Unterschied ist, dass ein Bauteil erst dann in der angegebenen Bibliothek „gesucht“ wird, wenn es nicht im Simulations-File steht. Durch die Include-Anweisung erhalten dagegen die in der Bibliothek angegebenen Definitionen die gleiche Priorität wie solche, die direkt in der Circuit-Datei stehen. Dies kann zu Problemen führen: Wenn nämlich derselbe Bauteil-Name sowohl in der Circuit-Datei als auch in der angegebenen Bibliothek auftaucht, führt dies zu einer Doppelanweisung und damit zu einer Fehlermeldung. Doppelanweisungen sind unzulässig! Dieses Problem tritt bei der LIB-Anweisung nicht auf (vergleiche Kapitel 6, Seite 14).

## 7 Analysearten

Durch eine Analyse-Anweisung wird dem Simulator SPICE mitgeteilt, welche Art der Analyse durchgeführt werden soll.

- Eine solche Anweisung (Statement) muss mit einem Punkt „.“ begonnen werden (z.B.: .DC).

Es gibt drei Hauptgruppen von Analyse-Anweisungen, nämlich die DC-Analyse (.DC), die AC-Analyse (.AC) und die Transienten-Analyse (.TRAN). Zu jeder dieser Gruppen sind sehr nützliche Ergänzungen vorhanden. Die wichtigsten sollen hier kurz erläutert werden.

## 7.1 DC-Analyse (.DC)

### 7.1.1 DC-Sweep *einer* unabhängigen Quelle

Die DC-Analyse dient dazu, eine Schaltung *gleichspannungs-* oder *-strommäßig* zu untersuchen. Insbesondere erlaubt sie das Durchlaufen eines bestimmten Wertebereiches einer unabhängigen Spannungs- oder Stromquelle. Die Analyse (Lösung der Kirchhoff'schen Gleichungen unter Benutzung der vollständigen Bauelementgleichungen) beginnt dann mit einem *Start-Wert*. Danach wird der Wert der betreffenden Quelle um ein vorgegebenes Inkrement verändert, die Kirchhoff'schen Gleichungen erneut gelöst, der Wert wieder um ein Inkrement verändert usw., bis zu einem *End-Wert*. Dies nennt man DC-Sweep. Syntax:

```
.DC Quelle Start-Wert End-Wert Schrittweite
```

Die in der Netzliste angegebene Wert-Zuweisung der Quelle wird dann durch die DC-Anweisung „überschrieben“.

Beispiel:

In einer Circuit-Datei möge beispielsweise die Quelle I2 wie folgt angegeben sein:

```
I2 3 6 SIN(1V 2V 1kHz) ,
```

also sinusförmig. Der Strom der unabhängigen Quelle I2 soll nun aber DC-mäßig verändert werden und zwar im Bereich von  $1\text{ mA}$  bis  $2\text{ mA}$  in Schritten von  $0,1\text{ mA}$ . Die DC-Anweisung

```
.DC I2 1mA 2mA 0.1mA
```

überschreibt dann die vorherige Angabe und erwirkt einen „DC-Sweep“.

Ein ähnliches Beispiel wurde bereits im Kapitel 3 (Seite 8, Beispiel B1) vorgestellt. Der dort angegebene Wert von  $I = 1\text{ mA}$  der Stromquelle I wird durch die DC-Anweisung überschrieben.

**PSPICE** und **LT-SPICE** bieten neben dieser *linearen* Variation (mit konstanten Schrittweiten) auch die Möglichkeit einer *dekadischen* Variation der Werte an.

Beispiel:

```
.DC DEC I2 1nA 100mA 20
```

In diesem Beispiel wird der Strom der Quelle I2 im Bereich von  $1\text{ nA}$  bis  $100\text{ mA}$  verändert und zwar mit 20 Schritten pro Dekade. Diese dekadische Variation ist besonders dann sehr wertvoll, wenn ein großer Wertebereich durchlaufen werden soll.

- Ein DC-Sweep ist während eines Simulationslaufes nur für *eine* unabhängige Quelle erlaubt. Mitunter möchte man aber *gleichzeitig* mehrere Quellen variieren (z.B. zwei Versorgungsspannungen). In solchen Fällen können gesteuerte Quellen gute Dienste leisten, die alle von ein und derselben unabhängigen Quelle „gesteuert“ werden.
- Die DC-Analyse erlaubt es, eine weitere Quelle als „Parameter“ anzugeben. Dies ist beispielsweise bei der Darstellung von Kennlinien-*Feldern* sehr hilfreich. Man setzt dann einfach die „Parameter-Quelle“ mit ihrer Variations-Definition hinter die Sweep-Variable.

Beispiel:

```
* Q1.cir (Kennlinienfeld IC = f(VCE); VBE = Parameter)
Q 2 3 0 C3ZN02
VCE 1 0 1V ; Kollektor-Emitter-Spannung
VIC 1 2 0V ; "Strommesser" für IC
VBE 3 0 1V ; Basis-Emitter-Spannung
.MODEL C3ZN02 NPN (IS=1.1E-16 NF=1.01 BF=190 VAF=105V)
.DC VCE 0V 30V 10mV VBE 660mV 760mV 20mV
.PROBE; nur bei PSPICE
.END
```

Durch die vier Zeilen der Netzliste (nach der Überschrift) wird die Schaltung zur Darstellung des Ausgangskennlinienfeldes eines Transistors (Typ C3ZN02) beschrieben. Die Spannungsquelle VIC hat den Wert  $0\text{ Volt}$ . Sie dient nur als „Strommesser“ für  $I_C$ . Ein solcher Strommesser ist jedoch bei **PSPICE** und **LT-SPICE** nicht erforderlich, in manchen Fällen aber trotzdem sinnvoll.

Die DC-Anweisung soll einen DC-Sweep der Kollektor-Emitter-Spannung VCE des Transistors erwirken. Die dahinter angegebene zweite Quelle VBE mit ihrer Definition ist der Parameter: Zuerst wird ein DC-Sweep erzeugt mit  $VBE = 660\text{ mV} = \text{konst.}$  Anschließend wird VBE um  $20\text{ mV}$  erhöht und ein neuer DC-Sweep durchgeführt usw. Man erhält schließlich eine Kurvenschar.

- **Anmerkung:** Das Semikolon mit dem dahinter stehenden Kommentar ist in **PSPICE** und **LT-SPICE** zulässig, nicht aber in **SPICE 3f4**!
- **Step-Anweisung:** In **PSPICE** und **LT-SPICE** kann die parametrische Darstellung von Kurvenverläufen auch mit Hilfe der so genannten „Step-Anweisung“ vorgenommen werden. Syntax:

```
.STEP Step-Variable Start-Wert End-Wert Schrittweite
```

Das STEP-Statement ist der DC-Anweisung sehr ähnlich. In der obigen Circuit-Datei könnten die Anweisungen lauten:

```
.DC VCE 0V 30V 10mV
.STEP VBE 660mV 760mV 20mV
```

Die STEP-Anweisung kann auch in Form einer *Liste* mit der Angabe der gewünschten Werte erfolgen:

```
.STEP VBE List 600mV 650mV 675mV 690mV 700mV
```

Die Ergebnisse können, wie im Kapitel 4 erläutert, grafisch dargestellt werden. Möchte man etwa in obigem Beispiel das Ausgangskennlinienfeld sehen, würde man den Kollektorstrom aufrufen:

- SPICE 3f4:** `plot i(VIC);` der Buchstabe *i* muss kleingeschrieben werden!
- PSPICE** und **LT-SPICE:** `i(VIC)` oder `IC(Q)`; beliebige Schreibweise für den Buchstaben *i*!

### 7.1.2 DC-Sweep der Temperatur in **PSPICE** und **LT-SPICE**

Die Temperatur ist ein sehr wichtiger Kontrollparameter; siehe hierzu auch **Abschnitt 9.2** (Seite 40). **PSPICE** und **LT-SPICE** bieten deshalb die Möglichkeit, die *Temperatur* durch eine DC-Anweisung zu variieren.

Syntax:

```
.DC TEMP Start-Wert End-Wert Schrittweite
```

Beispiel:

```
.DC TEMP -50 150 5
```

In diesem Beispiel wird die Temperatur von  $-50\text{ }^\circ\text{C}$  bis  $150\text{ }^\circ\text{C}$  in Schritten von  $5\text{ K}$  verändert; die Bauteilwerte werden für jeden einzelnen Temperaturwert neu errechnet. Dies ist für die Untersuchung von Temperaturabhängigkeiten ein sehr wichtiges Hilfsmittel, das in **SPICE 3f4** leider nicht zur Verfügung steht.

## 7.2 AC - Analyse (.AC)

Die AC-Analyse wird verwendet, um das *Kleinsignalverhalten* einer Schaltung über den *Frequenzbereich* zu untersuchen. Dazu wird zunächst der DC-Arbeitspunkt für die Schaltung unter Benutzung der vollständigen Bauelementgleichungen bestimmt. Für diesen Arbeitspunkt werden die Bauelementgleichungen anschließend *linearisiert* und das Frequenzverhalten analysiert. Übersteuerungen werden bei dieser Analyse nicht berücksichtigt. Dies bedeutet beispielsweise für einen Verstärker, der eine Spannungsverstärkung von 1000 hat und mit einer Eingangsspannung von  $1\text{ V}$  angesteuert wird, eine Ausgangsspannung von  $1000\text{ V}$ ! Dies führt *nicht* zu einem Problem.

Jeder unabhängigen Spannungs- und Stromquelle kann direkt in der Netzliste sowohl ein DC- als auch ein AC-Wert zugewiesen werden. Der DC-Wert geht direkt in die Arbeitspunktbestimmung ein; der AC-Wert wird dagegen nur für die Berechnung des Frequenzverhaltens benötigt.

Für den AC-Wert wird Sinusform vorausgesetzt und der angegebene Betrag als Effektivwert interpretiert.

Beispiel:

```
V2 4 5 DC 5V AC 1V
```

Die Spannungsquelle V2 liegt zwischen den Knoten „4“ und „5“ und hat einen DC-Wert von 5 V und einen sinusförmigen AC-Wert von 1 V (effektiv).

Die Frequenzvariation wird durch die AC-Anweisung beschrieben. Sie kann linear, dekadisch oder auch in Oktav-Schritten erfolgen. Syntax:

```
.AC {LIN|DEC|OCT} Zahl der Punkte fstart fstop
```

Beispiele:

```
.AC LIN 51 100Hz 200Hz
```

```
.AC DEC 20 10Hz 1Meg
```

```
.AC OCT 10 1kHz 16kHz
```

Im ersten Beispiel wird die Frequenz mit konstanter Schrittweite (linear) von 100 Hz bis 200 Hz variiert und zwar mit insgesamt 51 Frequenzwerten.

Im zweiten Beispiel wird dagegen die Frequenz mit 20 Werten pro Dekade von 10 Hz bis 1 MHz erhöht.

Im dritten Beispiel wird die Frequenz mit 10 Werten pro Oktave von 1 kHz bis 16 kHz gesteigert.

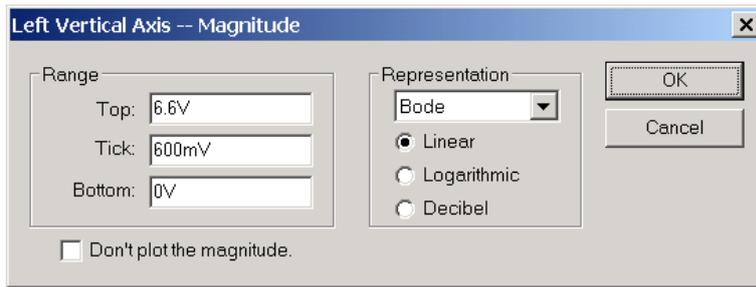
- Die AC-Anweisung (.AC . . .) bezieht sich immer auf alle AC-Quellen, d.h. alle unabhängigen Spannungs- und Stromquellen, für die ein AC-Wert angegeben ist, tragen zur AC-Analyse bei.
- Die AC-Analyse wird gern zur Untersuchung von Verstärkern eingesetzt. Da die Schaltung für den DC-Arbeitspunkt linearisiert wird, Übersteuerungen also unberücksichtigt bleiben, wählt man für die AC-Eingangsspannung zweckmäßigerweise den Wert AC 1V (d.h. Effektivwert = 1 V); dann entspricht die Ausgangsspannung direkt der Verstärkung. Analoges gilt für den Strom: AC 1A.
- Die grafische Darstellung erfolgt bei **PSPICE** in PROBE (siehe Abschnitt 4.2, Seite 10). PROBE bietet eine Reihe sehr hilfreicher Operationen für die grafische Aufbereitung der Simulationsergebnisse. Besonders nützlich ist die Darstellung des BODE-Diagramms. Die Frequenzabhängigkeit der Verstärkung eines Systems kann z.B. durch den Aufruf VDB (Kout1, Kout2) direkt in dB dargestellt werden, wenn die Eingangsquelle den Wert 1 V hat und „Kout1“, „Kout2“ die Ausgangsknoten sind. Wenn der Knoten „Kout2“ auf null liegt, reicht natürlich VDB (Kout1).

Durch Öffnen eines zweiten Plots und Aufruf VP (Kout1, Kout2) kann man den zugehörigen Phasengang abbilden (siehe Tabelle: „Zusatzbuchstaben“). Wenn der Knoten „Kout2“ auf null liegt, reicht natürlich VP (Kout1).

- Knotenspannungen (V) und Bauelementströme (I) können in **PSPICE** als Amplitude, Phase, Real- oder Imaginärteil durch einen Zusatzbuchstaben hinter V bzw. I spezifiziert werden:

Zusatzbuchstabe	Bedeutung	Zusatzbuchstabe	Bedeutung
keiner	Effektivwert (Magnitude)	P	Phase
M	Effektivwert (Magnitude)	R	Realteil
DB	Angabe in dB ( $20 \cdot \lg(\text{Wert})$ )	I	Imaginärteil

- In **LT-SPICE** erhält man nach dem Aufruf `V(Knoten...)` sofort die Darstellung in Form des Bode-Diagramms. Man kann dies aber leicht umstellen. Durch Klick mit der *linken* Maustaste auf die Y-Achse öffnet sich ein Menü, und man kann zwischen Linear, Logarithmic und Decibel wählen. Die Darstellung



(Representation) kann ebenfalls verändert werden. Zur Auswahl stehen: Bode, Nyquist und Cartesian. Die Nyquist-Darstellung zeigt die sogenannte Ortskurve in der komplexen Ebene (Imaginärteil = Y-Achse, Realteil = X-Achse). Die Cartesian-Darstellung zeigt Real- und Imaginärteil in Abhängigkeit von der Frequenz. Außerdem kann man

bei der Darstellung zweier Plots *übereinander* durch Klick mit der *rechten* Maustaste die Synchronisation der horizontalen Achsen ausschalten (Syn. Horiz. Axis). Dies ist in **LT-SPICE** sehr gut gelöst, siehe [14].

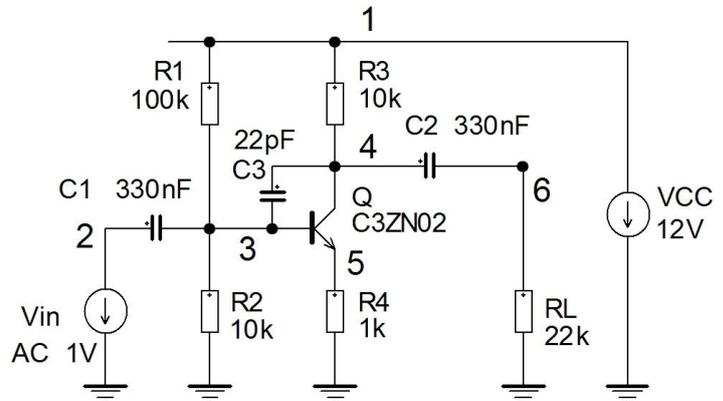
Beispiel Q2a:

\* Q2a.Cir (Transistorverstärker)

```
VCC 1 0 12V
Vin 2 0 AC 1V
C1 2 3 330nF
C2 4 6 330nF
C3 4 3 22pF
R1 1 3 100k
R2 3 0 10k
R3 1 4 10k
R4 5 0 1k
RL 6 0 22k
Q 4 3 5 C3ZN02
```

```
.MODEL C3ZN02 NPN (IS=1.1E-16
+ BF=190 VAF=105 KF=3E-15)

.AC DEC 20 1Hz 10Meg
.PROBE
.END
```



**Bild 6**  
Transistor-Verstärker

Durch die obige AC-Anweisung wird die Quelle `Vin` dekadisch im Bereich von  $1\text{ Hz}$  bis  $10\text{ MHz}$  mit 20 Werten pro Dekade variiert.

**Bild 7** zeigt das Simulationsergebnis (aus PSPICE-PROBE). Der Verstärkungsverlauf ist in  $dB$  dargestellt. Es sind gleich zwei getrennte Plot-Fenster abgebildet; eins für den Amplituden- und eins für den Phasengang. Man ruft für den Amplitudenverlauf einfach auf:

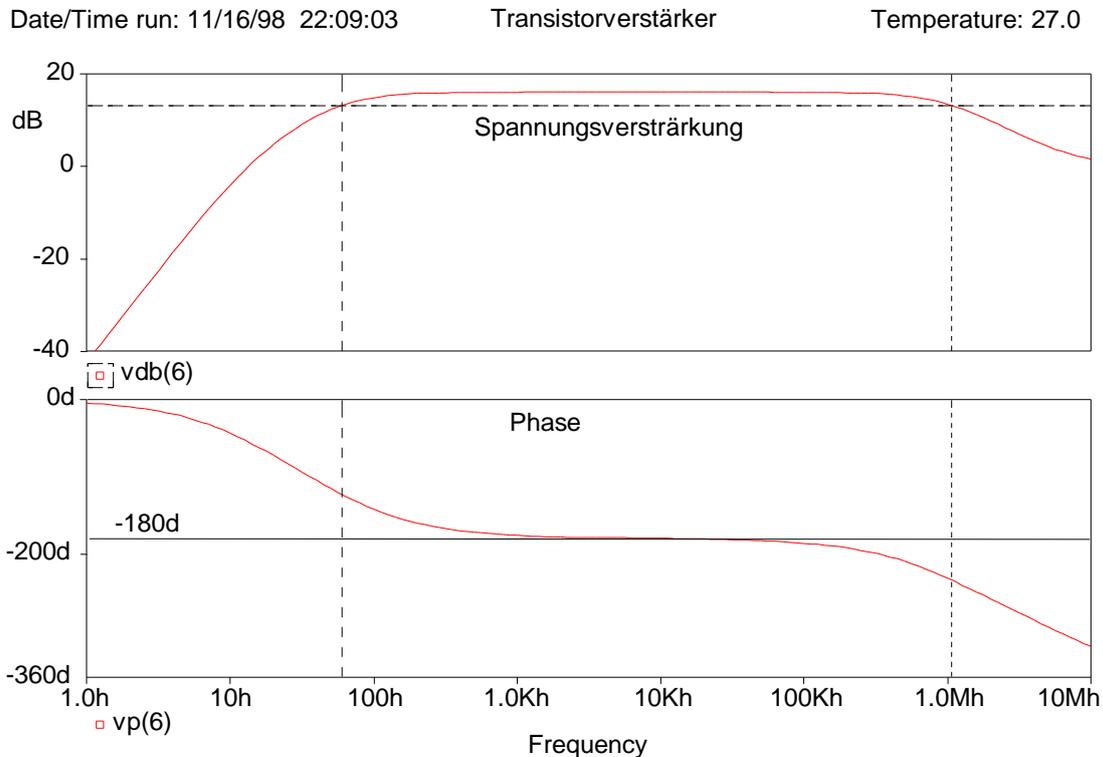
`VdB (6) .`

Es genügt die Angabe des Ausgangsknotens „6“, da die Eingangsspannung `V(2)` den Wert  $1\text{ V}$  hat; die Verstärkung ist hier also direkt mit `V(6)` identisch. Übersteuerungen der Transistorstufe bleiben wegen der Linearisierung im Arbeitspunkt unberücksichtigt. Den Phasenverlauf erhält man durch den Aufruf im zweiten Plot-Fenster:

`Vp (6) .`

- **SPICE 3f4** erlaubt eine ähnliche Darstellung mit obigen Zusatzbuchstaben, doch es gibt nur einen Plot.
- In **LT-SPICE** gibt es die Zusatzbuchstaben nicht. Trotzdem kann man die Kurven ähnlich darstellen wie dies in **Bild 7** für **PSPICE** zu sehen ist. Man öffnet einfach zwei Plot-Fenster übereinander (mit der Maus

ins Plot-Fenster gehen und mit der rechten Maus-Taste klicken, dann: **Add Plot Pane**) und stellt in beiden V(6) dar. Dann lässt sich z.B. im oberen Fenster der Phasenverlauf ausblenden und im unteren der Amplitudengang. Dazu wird die Maus auf die Skala gebracht, die nicht dargestellt werden soll. Durch Drücken der linken Maus-Taste erscheint ein Menü. Darin kann entweder „**Don't plot phase**“ (rechte Skala) oder „**Don't plot the magnitude**“ (linke Skala) ausgewählt werden.



**Bild 7** Bode-Diagramm zu Q2a: Amplituden- und Phasenverlauf der Verstärkung in „PROBE“

### 7.3 Rausch-Analyse in PSPICE und LT-SPICE (.NOISE)

Die Rausch-Analyse ist eine Ergänzung zur AC-Analyse. Von jedem Bauteil in einer Schaltung werden die Rauschbeiträge berechnet und ihre Auswirkung auf einen als Ausgang spezifizierten Knoten (oder ein Knotenpaar) ermittelt. Für diesen Ausgang werden dann die einzelnen Rauschteile effektivwertmäßig zusammengefasst. Die Berechnung wird dabei für alle in der AC-Anweisung vorgesehenen Frequenzen durchgeführt. Daher kann eine Rauschberechnung nur in Verbindung mit einer AC-Analyse erfolgen.

In der Rausch-Anweisung muss neben der Angabe des Ausganges stets eine unabhängige Eingangsquelle (Spannungs- oder Stromquelle) angegeben werden, um auch ein äquivalentes Eingangsrauschen zu erhalten. Das ist wie folgt zu verstehen: SPICE berechnet die resultierende frequenzabhängige Verstärkung (bezogen auf die unabhängige Eingangsquelle) und dividiert das Ausgangsrauschen durch die ermittelte Verstärkung. Das Ergebnis wird dann als äquivalentes Eingangsrauschen interpretiert.

Die Rausch-Anweisung hat für **PSPICE** und **SPICE 3f4** die allgemeine Form (**LT-SPICE** siehe unten):

```
.NOISE V(Knoten) Eingangsquelle
```

oder

```
.NOISE V(Knoten1, Knoten2) Eingangsquelle
```

Wenn die Eingangsquelle eine Spannungsquelle ist, wird das auf den Eingang bezogene Ausgangsrauschen in  $\text{Volt} / \sqrt{\text{Hz}}$  angegeben. Bei einer Stromquelle erfolgt die Angabe dagegen in  $\text{A} / \sqrt{\text{Hz}}$ . Das Ausgangsrauschen wird immer in  $\text{Volt} / \sqrt{\text{Hz}}$  angegeben.

Beispiel:

Es soll eine Rausch-Analyse für die Verstärker-Schaltung **Bild 6** vorgenommen werden. Als Ausgang werde der Knoten „6“ angesehen und als Bezugsquelle am Eingang die Quelle  $V_{in}$ . Die Netzliste auf Seite 30 wird dann um die Rausch-Anweisung ergänzt:

```
.NOISE V(6) Vin
```

Dies bedeutet, dass die Rauschbeiträge aller Schaltungselemente am Ausgangsknoten „6“ zum resultierenden Ausgangsrauschen zusammengefasst werden. Dies sind die Rauschbeiträge des Transistors Q (Parameter im Modell:  $KF = 3E-15$ ) und aller Widerstände. Das äquivalente Eingangsrauschen kann dann auf den Eingang (Quelle  $V_{in}$ ) bezogen werden. Der Frequenzbereich wird durch die AC-Anweisung definiert. In obigem Beispiel werden somit alle Anteile im Bereich von 1 Hz bis 10 MHz erfasst (vergleiche Netzliste auf Seite 30).

Die Darstellung des Rauschens erfolgt bei **PSPICE** in PROBE. Interessiert man sich für das resultierende Ausgangsrauschen, wird in PROBE einfach aufgerufen:

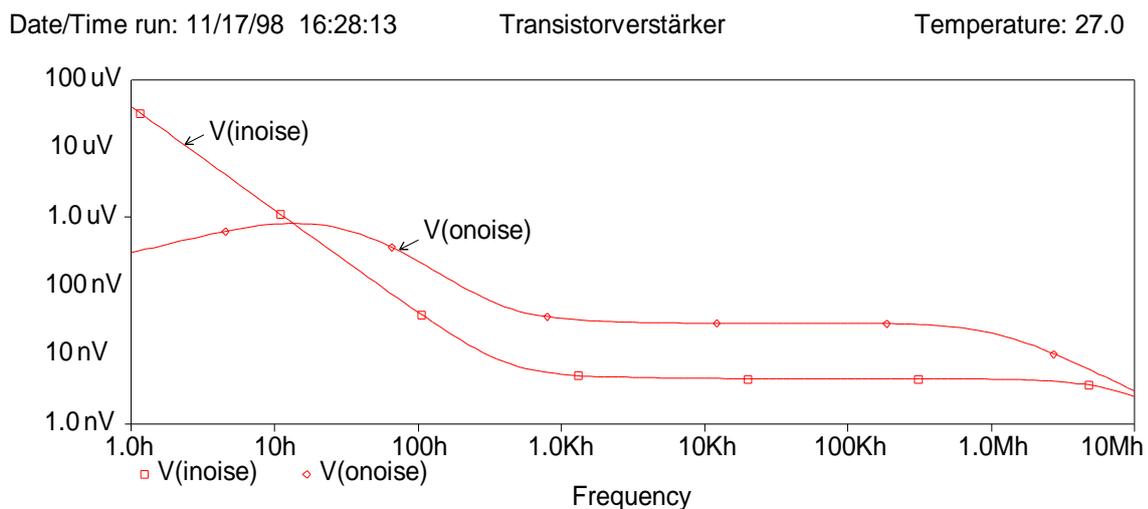
```
V(ONoise)
```

Dargestellt wird dann das resultierende Ausgangsrauschen am Knoten „6“ in Abhängigkeit von der Frequenz.

Das äquivalente Eingangsrauschen kann durch den Aufruf

```
V(INoise)
```

sichtbar gemacht werden (siehe **Bild 8**).



**Bild 8** Ergebnis der Rausch-Analyse (Q2b) zur Verstärker-Schaltung Bild 6, „PSPICE-PROBE“

In **LT-SPICE** ist die Formulierung in der Circuit-Datei geringfügig anders. Die Rauschanweisung ist die gleiche, nur werden hier die AC-Parameter (dec 20 1Hz 10Meg) gleich dahinter geschrieben:

```
.NOISE V(6) Vin dec 20 1Hz 10Meg
```

Die Zeile `.AC dec 20 1Hz 10Meg` wird dann nicht benötigt und muss durch einen Stern (\*) oder ein Semikolon (;) am Zeilenanfang ausgeschaltet werden. Arbeitet man mit dem grafischen Schaltungs-Editor von LT-SPICE, ist es nicht notwendig, das „Kommentar-Zeichen“ zu setzen. Beim Starten der Simulation erscheint die Frage, welche Simulation durchgeführt werden soll. Danach erscheint das Zeichen automatisch an der richtigen Stelle.

Die grafische Darstellung erfolgt dann wie bei den anderen Simulatoren. In LT-SPICE können auch die Rauschanteile der einzelnen Bauteile angezeigt werden. Dazu reicht ein Links-Klick auf das betreffende Bauteil im Schaltplan aus.

## 7.4 Analyse im Zeitbereich (.TRAN)

Die Transienten-Analyse berechnet das Verhalten einer Schaltung über einen angegebenen *Zeitraum*. Bei den Berechnungen werden die vollständigen Bauteilgleichungen verwendet (Großsignalverhalten). Da die Lösung der vollständigen Differentialgleichungen einen hohen Rechenaufwand erfordert, dauert eine Transienten-Analyse entsprechend lange.

Die Transienten-Analyse startet immer zum Zeitpunkt  $t = 0$  und läuft bis zur spezifizierten Endzeit  $T_{stop}$ . Eine sinnvolle Transienten-Analyse erhält man natürlich nur, wenn mindestens eine unabhängige Quelle eine Zeitfunktion enthält. Dies kann z.B. eine SIN- oder eine PULSE-Anregung sein (siehe Abschnitt 6.2).

Die typische Analyse-Anweisung hat die Form:

```
.TRAN Tstep Tstop [Tstart [dTmax]]
```

Hierin gibt  $T_{stop}$  den Analyse-Zeitraum an. Die Zeitangabe  $T_{step}$  spezifiziert das Druck- oder Plot-Intervall für Zeilendrucker. Die Angaben  $T_{start}$  und  $dT_{max}$  sind zusätzliche Optionen: Mit  $T_{start}$  können die Ergebnisse bis zu diesem Zeitpunkt unterdrückt werden. Mit der Angabe von  $dT_{max}$  kann die *maximale* Zeit-Schrittweite vorgegeben werden. Die *wahre* Schrittweite wird vom Programm automatisch an die Rechenaktivität angepasst. Standardmäßig ist  $dT_{max} = T_{stop}/50$ . Ein *kleinerer* Wert  $dT_{max}$  führt zu „glatteren“ Kurven, verlängert aber die Rechenzeit.

Die grafische Darstellung des Ergebnisses erfolgt genauso wie bei der DC-Analyse: Bei **PSPICE** in PROBE, bei **LT-SPICE** wird man zur Auswahl der darzustellenden Kurven aufgefordert und bei **SPICE 3f4** verwendet man, wie üblich, das Plot-Kommando `plot . . . .`

## 7.5 Fourier-Analyse in PSPICE und LT-SPICE (.FOUR)

Die Programme **PSPICE** und **LT-SPICE** bieten nach erfolgreicher *Transienten-Analyse mit periodischer Anregung* die Möglichkeit, eine Fourier-Analyse der berechneten Signale (Spannungen und Ströme) durchzuführen. Der Bestimmung der Fourier-Koeffizienten eines periodischen Signals wird *rechtsbündig genau die letzte Periode* (Kehrwert der Grundfrequenz  $f_1$ ) zugrundegelegt. Die Anweisung lautet:

```
.FOUR f1 [Zahl der Harmonischen] Ausgangswert1 Ausgangswert2 . . .
```

Mit `Ausgangswert1`, `Ausgangswert2`, `. . .` werden die Signale (Knotenspannungen oder Bauelementströme) ausgewählt, für die eine Fourier-Analyse durchgeführt werden soll. Die Frequenz  $f_1$  kennzeichnet die Grundfrequenz des periodischen Signals. Wahlweise kann die Zahl der zu berechnenden Harmonischen angegeben werden; der Default-Wert ist 9. Ermittelt wird auch der DC-Anteil.

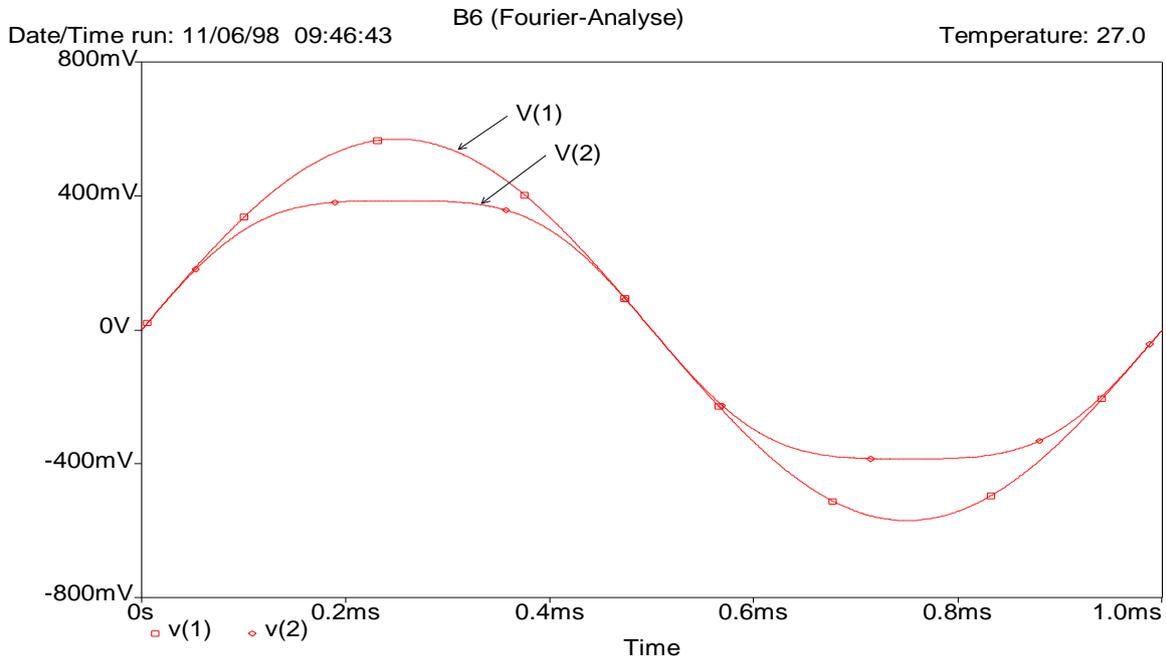
Beispiel B6:

Die Transferfunktion eines Systems soll durch die Funktion  $V_{out} = V_{in} - V_{in}^3$  beschrieben werden (mathematische Ausdrücke siehe Kapitel 8). Um dies zu realisieren, bietet sich der Einsatz einer spannungsgesteuerten Spannungsquelle E an. Die Simulationsdatei kann dann z.B. die Form haben:

```
* B6.cir (Fourier-Analyse)
Vin 1 0 SIN(0V 0.57V 1kHz) ; Eingangsspannung
Rin 1 0 1G ; Eingangswiderstand
E 2 0 VALUE = {V(1)-PWRS(V(1),3)}
Rout 2 0 1k ; Ausgangswiderstand

.TRAN 1us 1ms 0s 10ns
.FOUR 1kHz V(1) V(2)
.PROBE
.END
```

Die Transienten-Analyse wird für eine Periode ( $1\text{ ms} \hat{=} 1\text{ kHz}$ ) durchgeführt. Da Einschwingvorgänge hier nicht auftreten, reicht als Analyse-Zeitraum eine Periode aus. Das Statement `.FOUR 1kHz V(1) V(2)` veranlasst **PSpICE** oder **LT-SpICE** anschließend dazu, die Fourier-Koeffizienten für die beiden periodischen Schwingungen  $V(1)$  (= Eingangsspannung) und  $V(2)$  (= Ausgangsspannung) zu berechnen. Dabei gibt  $f1$  (hier ist  $f1 = 1\text{ kHz}$ ) die „Grundfrequenz“ an, und die Zahl der Harmonischen ist hier nicht spezifiziert.



**Bild 9** Ein- und Ausgangsschwingung zu obiger Circuit-Datei (Beispiel B6, Seite 33)

**Bild 9** zeigt die zeitlichen Spannungsverläufe. Die Fourier-Koeffizienten können dem Output-File entnommen werden. Dieses wird bei **PSpICE** im CIRCUIT-Ordner mit der Endung `OUT` gespeichert und kann direkt aus **PSpICE** heraus aufgerufen werden. Auch bei **LT-SpICE** wird die Tabelle im CIRCUIT-Ordner abgelegt, jedoch mit der Endung `LOG`. Dieses Log-File ist über **View** → **SpICE Error Log** ebenfalls einfach zu erreichen.

Für das vorliegende Beispiel erhält man für die beiden Spannungen folgendes Ergebnis (Ausschnitt aus dem File `B6.OUT` bzw. `B6.LOG`):

```
FOURIER COMPONENTS OF TRANSIENT RESPONSE V(1)
DC COMPONENT = 1.701270E-08

```

HARMONIC NO	FREQUENCY (HZ)	FOURIER COMPONENT	NORMALIZED COMPONENT	PHASE (DEG)	NORMALIZED PHASE (DEG)
1	1.000E+03	5.700E-01	1.000E+00	1.140E-06	0.000E+00
2	2.000E+03	1.109E-08	1.945E-08	-8.783E+01	-8.783E+01
3	3.000E+03	5.339E-09	9.366E-09	-8.117E+01	-8.117E+01
4	4.000E+03	3.246E-09	5.694E-09	-1.091E+02	-1.091E+02
5	5.000E+03	7.912E-10	1.388E-09	1.765E+02	1.765E+02
6	6.000E+03	1.194E-09	2.094E-09	-1.069E+01	-1.069E+01
7	7.000E+03	1.836E-09	3.222E-09	-7.464E+01	-7.464E+01
8	8.000E+03	4.614E-10	8.095E-10	-1.332E+02	-1.332E+02
9	9.000E+03	8.003E-10	1.404E-09	-1.348E+02	-1.348E+02

```
TOTAL HARMONIC DISTORTION = 2.275386E-06 PERCENT
```

## FOURIER COMPONENTS OF TRANSIENT RESPONSE V(2)

DC COMPONENT = 1.691368E-08

HARMONIC NO	FREQUENCY (HZ)	FOURIER COMPONENT	NORMALIZED COMPONENT	PHASE (DEG)	NORMALIZED PHASE (DEG)
1	1.000E+03	4.311E-01	1.000E+00	2.375E-06	0.000E+00
2	2.000E+03	6.612E-09	1.534E-08	9.006E+01	9.006E+01
3	3.000E+03	4.630E-02	1.074E-01	4.146E-06	1.771E-06
4	4.000E+03	1.254E-08	2.909E-08	-8.915E+01	-8.915E+01
5	5.000E+03	5.873E-09	1.362E-08	-9.088E+01	-9.088E+01
6	6.000E+03	3.619E-09	8.394E-09	-9.000E+01	-9.000E+01
7	7.000E+03	3.005E-09	6.971E-09	-9.289E+01	-9.289E+01
8	8.000E+03	1.377E-09	3.193E-09	-6.318E+01	-6.318E+01
9	9.000E+03	5.069E-10	1.176E-09	-1.188E+02	-1.188E+02

TOTAL HARMONIC DISTORTION = 1.073908E+01 PERCENT

Die obigen Tabellen enthalten die Amplituden und Phasen der ersten neun Harmonischen der Schwingungen  $V(1)$  und  $V(2)$ . Die Werte sind auch in einer normalisierten Form dargestellt, jeweils auf die Amplitude der Grundschwingung bezogen (hier  $f_1 = 1 \text{ kHz}$ ). Außerdem werden die DC-Komponenten und die Verzerrungsfaktoren angegeben. Folgendes fällt auf:

- Die Eingangsspannung  $V(1)$  sollte eine reine Sinusschwingung mit der Amplitude  $0.57 \text{ V}$  sein. Die Tabelle zeigt jedoch auch Werte für die höheren Harmonischen. Dies ist auf eine begrenzte Rechengenauigkeit zurückzuführen. Durch eine Options-Anweisung, z.B.: `.OPTIONS reltol=0.0001 abstol=0.1pA chgtol=0.001pC vntol=0.1uV*)`, und eine kleinere Schrittweite `dTmax` kann die Genauigkeit erhöht werden, doch wegen der damit verbundenen längeren Rechenzeit ist stets zu prüfen, ob der Aufwand gerechtfertigt ist. In vielen Fällen kann ein gewisses „numerisches Rauschen“ toleriert werden. Ähnliches gilt auch für die Phasen und den DC-Wert.

<sup>\*)</sup> Default-Werte sind: `reltol=0.001 abstol=1pA chgtol=0.01pC vntol=1uV`, vergleiche Kapitel 9 und Abschnitt 9.3 oder Manual zum Programm.

- Ein Blick auf die Tabelle für die Ausgangsschwingung  $V(2)$  zeigt (vom numerischen Rauschen abgesehen), dass neben der Grundschwingung – wie zu erwarten ist – praktisch nur die dritte Harmonische enthalten ist. Sie ist in Phase mit der Grundschwingung und hat eine Amplitude von etwa  $10.7 \%$ .
- Die Fourier-Analyse liefert auch den Verzerrungs- oder Klirrfaktor  $k$  in Prozent:

$$k = \sqrt{\frac{V_2^2 + V_3^2 + V_4^2 + \dots}{V_1^2}} \times 100\%$$

Für die Eingangsspannung  $V(1)$  erhält man wegen des numerischen Rauschens nicht exakt den Wert  $k = 0$ . Für die Ausgangsspannung  $V(2)$  ergibt sich praktisch der Wert der dritten Harmonischen:  $k \approx 10.7 \%$ .

**Achtung:** Da bei der Fourier-Analyse nur die letzte Periode verwendet wird, muss der Darstellungszeitraum ( $T_{stop} - T_{start}$ ) nicht über ganzzahlige Vielfache der Periodendauer des Signals erstreckt werden. Es muss aber mindestens eine volle Periode dargestellt werden. Sind Einschwingvorgänge zu erwarten, muss eine hinreichend lange Zeit  $T_{stop}$  vorgegeben werden, um den DC-Wert nicht zu verfälschen.

Die Rechengenauigkeit kann durch die oben erwähnte Options-Anweisung beeinflusst werden. Die Genauigkeit kann weiter erhöht werden, wenn das maximale Schrittmass `dTmax` in der Transienten-Anweisung klein im Vergleich zur Periode gewählt wird. Deshalb ist in der Beispiel-Datei ein Wert von  $10 \text{ ns}$  eingetragen. Dadurch erhöhen sich allerdings die Rechenzeit und der Umfang der Ergebnisdatei! Bei **LT-SPICE** ist außerdem zu beachten, dass für die Speicherung des Ergebnis-Files (RAW-Datei) normalerweise eine beträchtliche Datenkompression vorgenommen wird. Durch die Options-Anweisung: `.Options Plotwinsize = 0` kann die Datenkompression jedoch einfach ausgeschaltet werden.

## 7.6 Fast-Fourier-Transformation in PSPICE und LT-SPICE (FFT)

Die beiden Simulatoren **PSPICE** und **LT-SPICE** gestatten neben der Bestimmung der Fourier-Koeffizienten auch die grafische Darstellung der Harmonischen als *Frequenz-Spektrum* (FFT-Plot). Diese Darstellung ist für viele Anwendungen sehr viel aufschlussreicher als die Tabellenform. Verwendet wird alles, was in den Darstellungszeitraum ( $T_{stop} - T_{start}$ ) fällt. Deshalb sind unbedingt einige Vorüberlegungen notwendig, damit nicht Frequenzlinien *vorgetäuscht* werden, die in Wahrheit nicht auftreten:

- Der Darstellungszeitraum ( $T_{stop} - T_{start}$ ) muss über *ganzzahlige Vielfache der Periodendauer* des Signals erstreckt werden.  $T_{start}$  ist den Einschwingvorgängen anzupassen.
- Der Darstellungszeitraum ( $T_{stop} - T_{start}$ ) bestimmt sowohl die Breite der Linien als auch die Startfrequenz des Spektrums. Das gilt auch für den minimalen Frequenzschritt. Es gelten die Beziehungen:

$$\text{Linienbreite} = \frac{2}{T_{stop} - T_{start}}; \quad \text{Startfrequenz} = \text{Frequenzschrittweite} = \frac{1}{T_{stop} - T_{start}}.$$

- Der maximale Zeitschritt  $dT_{max}$  kann als Kehrwert der minimalen Samplefrequenz  $f_{s,min}$  gedeutet werden, und diese Frequenz muss entsprechend der „Shannon-Bedingung“ mindestens doppelt, besser  $20\times$  so hoch sein [9], wie die höchste im Signal vorkommende Oberwellenfrequenz  $f_{max}$ . Folglich gilt:

$$f_{s,min} = \frac{1}{dT_{max}} \geq (2 \dots 20) \cdot f_{max} \quad \rightarrow \quad dT_{max} = \frac{1}{f_{s,min}} \leq \frac{1}{(2 \dots 20) \cdot f_{max}}.$$

$dT_{max}$  ist der *maximal* vorgegebene Zeitschritt. Die tatsächliche Schrittweite wird vom Simulator automatisch den Erfordernissen angepasst und kann sehr viel kleiner ausfallen, d.h. die tatsächliche Samplefrequenz  $f_s$  ist meist höher als  $f_{s,min}$  und auch nicht konstant.

- Die Zahl der Samples für die FFT-Darstellung ergibt sich aus dem Zeitraum ( $T_{stop} - T_{start}$ ) und der Samplefrequenz  $f_s$ . Wäre  $f_s$  konstant, könnte die Samplezahl wie folgt angegeben werden:

$$\text{Samplezahl} = \text{Samplefrequenz} \times \text{Darstellungszeitraum} = f_s \cdot (T_{stop} - T_{start}).$$

Nun ist aber die Samplefrequenz  $f_s$  in der Regel größer als  $f_{s,min} = 1/dT_{max}$ . Folglich gilt:

$$\text{Samplezahl} \geq f_{s,min} \cdot (T_{stop} - T_{start}) = \frac{\text{Darstellungszeitraum}}{\text{maximaler Zeitschritt}} = \frac{T_{stop} - T_{start}}{dT_{max}}.$$

Für die Berechnung eines FFT-Plots wird die Samplezahl in Form einer Zweierpotenz  $2^N$  verwendet (512, 1024, 2048, ...).  $N$  ist ganzzahlig. Der nächst kleinere Wert reicht meist aus.

- Bei **LT-SPICE** wird man nach der Samplezahl (**Number of data samples in time**) gefragt. Eine hohe Samplezahl vergrößert zwar den Dynamikbereich der Amplitudendarstellung, doch ist es nicht sinnvoll, mehr Samples vorzugeben, als tatsächlich während der Transienten-Analyse ermittelt wurden. Unerwünschte Details im Spektrum wären die Folge. Meist reicht es aus, sich an der höchsten im Spektrum erwarteten Oberwellenfrequenz  $f_{max}$  zu orientieren:

$$\text{gewählte Samplezahl} = 2^N \approx 2 \cdot f_{max} \cdot (T_{stop} - T_{start}).$$

Bei **LT-SPICE** ist außerdem zu beachten, dass für die Speicherung des Ergebnis-Files (RAW-Datei) normalerweise eine beträchtliche Datenkompression vorgenommen wird. Die FFT-Darstellung wird dadurch deutlich verschlechtert. Durch die Options-Anweisung: `.Options Plotwinsize = 0` kann die Datenkompression jedoch einfach ausgeschaltet werden.

Für das Beispiel B6 des vorigen Abschnittes möge der Frequenzbereich bis  $10 \text{ kHz}$  interessant sein, und die Linienbreite soll  $100 \text{ Hz}$  betragen. Eine Darstellungszeit von  $T_{stop} - T_{start} = 2/(100 \text{ Hz}) = 20 \text{ ms}$  reicht somit aus. Dieser Wert ist bereits ein ganzes Vielfaches der Signal-Periode. Einschwingvorgänge sind

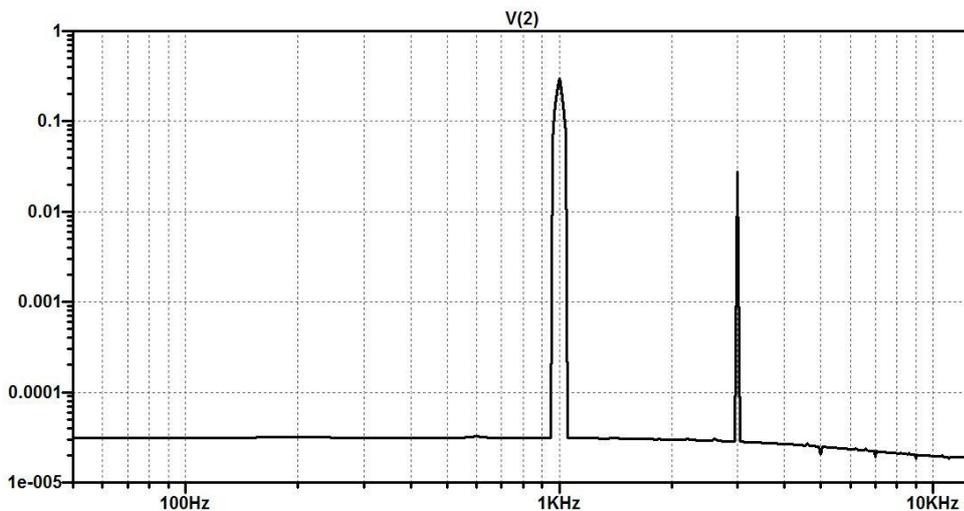
nicht zu erwarten, also kann der Startwert  $T_{start}$  auf „0“ gesetzt werden. Um das Spektrum bis zu einer Frequenz von  $f_{max} = 10 \text{ kHz}$  darstellen zu können, sollte die Samplefrequenz auf jeden Fall größer als  $20 \text{ kHz}$  sein. Als maximaler Zeitschritt  $dT_{max}$  reicht somit ein Wert von  $50 \mu\text{s}$  gerade aus. Mit  $dT_{max} = 5 \mu\text{s}$  wird jedoch eine höhere Genauigkeit erzielt.

In **TL-SPICE** kann als Samplezahl (Number of data samples in time) ein Wert von  $2 \cdot f_{max} \cdot (T_{stop} - T_{start}) = 400$  (nächste Zweierpotenz:  $512 = 2^9$ ,  $N = 9$  oder  $256$ ,  $N = 8$ ) eingesetzt werden. **Bild 10** zeigt einen solchen Plot. In der Circuit-Datei (siehe Seite 33) wird die Transienten-Anweisung wie folgt geändert:

```
.TRAN 1us 20ms 0s 5us
```

Zusätzlich wird die Datenkompression durch eine Optionsanweisung ausgeschaltet:

```
.Options Plotwinsize = 0
```



**Bild 10**

FFT-Plot zu Beispiel B6 (mit LT-SPICE ermittelt).

Neue Anweisungen in der Circuit-Datei (siehe Seite 33):

```
.TRAN 1us 20ms 0s 5us
```

```
.Options Plotwinsize=0
```

Samplezahl:  $N = 512$

## 8 Mathematische Operationen, arithmetische Ausdrücke

### 8.1 Eingebaute Operationen und Ausdrücke

Die wichtigsten mathematischen Operatoren für arithmetische Ausdrücke:

Operator	Bedeutung	Operator	Bedeutung
+	Addition	*	Multiplikation
-	Subtraktion	/	Division
=	Gleichheitszeichen		

Die folgenden mathematischen Operationen sind möglich (nur für *reelle* Variablen  $x$ ):

Funktion	Kommentar	SPIICE 3f4	PSPIICE	LT-SPIICE
$ x $	Absolutwert	ABS(x)	ABS(x)	ABS(x)
$x^{1/2}$	Quadratwurzel	SQRT(x)	SQRT(x)	SQRT(x)
$e^x$	Exponential	EXP(x)	EXP(x)	EXP(x)

Funktion	Kommentar	SPICE 3f4	PSPICE	LT-SPICE
$\ln(x)$	Log. Basis $e$	LN(x)	LOG(x)	LN(x) = LOG(x)
$\lg(x)$	Log. Basis 10	LOG(x)	LOG10(x)	LOG10(x)
$x^m$	Potenz	(x)^m		x**m
$ x ^m$	Potenz von $ x $		PWR(x,m)	PWR(x,m)
$\text{Sgn}(x) \cdot  x ^m$	+ $ x ^m$ für $x > 0$ - $ x ^m$ für $x < 0$		PWRS(x,m)	PWRS(x,m)
$\sin(x)$	$x$ im Bogenmaß	SIN(x)	SIN(x)	SIN(x)
$\cos(x)$	$x$ im Bogenmaß	COS(x)	COS(x)	COS(x)
$\tan(x)$	$x$ im Bogenmaß	TAN(x)	TAN(x)	TAN(x)
$\arcsin(x)$	Ergebnis im Bogenmaß	ASIN(x)	ASIN(x)	ASIN(x)
$\arccos(x)$	Ergebnis im Bogenmaß	ACOS(x)	ACOS(x)	ACOS(x)
$\arctan(x)$	Ergebnis im Bogenmaß	ATAN(x)	ATAN(x)	ATAN(x)
$\sinh(x)$	$\frac{e^x - e^{-x}}{2}$	SINH(x)		SINH(x)
$\cosh(x)$	$\frac{e^x + e^{-x}}{2}$	COSH(x)		COSH(x)
$\tanh(x)$	$\frac{e^x - e^{-x}}{e^x + e^{-x}}$	TANH(x)		TANH(x)
$\text{Arsinh}(x)$	Umkehrfunkt. v. $\sinh(x)$	ASINH(x)		ASINH(x)
$\text{Arcosh}(x)$	Umkehrfunkt. v. $\cosh(x)$	ACOSH(x)		ACOSH(x)
$\text{Artanh}(x)$	Umkehrfunkt. v. $\tanh(x)$	ATANH(x)		ATANH(x)

Diese Tabelle ist nicht vollständig. Es stehen z.B. auch logische Verknüpfungen zur Verfügung. Hierzu sei jedoch auf die „integrierte“ Hilfe verwiesen, die in allen drei Programmen enthalten ist.

## 8.2 Anwenderdefinierte Funktionen

In **PSPICE** und **LT-SPICE** ist es möglich, eigene Funktionen zu definieren und diese in der Simulation einzusetzen. Von dieser Möglichkeit wird vor allem dann Gebrauch gemacht, wenn längere Ausdrücke gebraucht werden. Solche Funktionen werden in **LT-SPICE** nach folgender Syntax gebildet:

```
.Func <Name> ([Argumente]) {<Ausdruck>}
```

Beispiel:

```
.Func Pythag(x, y) {sqrt(x*x+y*y)}
```

Eine besondere Bedeutung haben solche Funktionen beim Formulieren komplexer Bauteil-Modelle und Subcircuit-Definitionen. Auch Parameter können übergeben werden. Nähere Informationen sind in der LT-SPICE-Hilfe zu finden: **Hilfe** → **LTspice** → **Dot Commands** → **.FUNC**.

Als Name einer selbstdefinierten Funktion darf ein bereits bestehender nicht verwendet werden, damit es nicht zu einem Konflikt kommt. So darf z.B. „Sin“ nicht verwendet werden.

In **PSPICE** ist die Syntax ähnlich. Allerdings wird die geschweifte Klammer nicht verwendet. Nähere Informationen sind in den Handbüchern zu finden; siehe auch [8].

## 9 Optionen und Kontrollparameter

### 9.1 Die Anweisung .OPTIONS

Die Anweisung

```
.OPTIONS Optname1 Optname2 ... Optname11=Wert1 Optname12=Wert2 ...
```

erlaubt es, die Simulationsbedingungen zu beeinflussen und auch das Aussehen der Ausgabedatei zu manipulieren. Dabei gibt es Parameter, die nur als „Schalter“ wirken (*Optname1*, *Optname2*, ...) und andere, die durch Angabe eines Wertes die „Default-Werte“ überschreiben (*Optname11=Wert1*, *Optname12=Wert2*, ...). In der folgenden Tabelle sind einige dieser Kontrollparameter, die sowohl für **SPICE 3f4**, als auch für **PSPICE** und **LT-SPICE** gelten, zusammengestellt:

Parameter, die als Schalter wirken (die Tabelle ist nicht vollständig!):

Parameter	Bedeutung
<i>ACCT</i>	Ausgabe einer Zusammenfassung u. Statistikinformationen
<i>LIST</i>	Listet die verwendeten Bauelemente auf
<i>NODE</i>	Listet die Knotentabelle auf
<i>NOMOD</i>	Unterdrückt die Ausgabe der Modell-Parameter
<i>NOPAGE</i>	Unterdrückt die Seitenzahlen u. die Kennzeichnung eines Abschnittes
<i>OPTS</i>	Listet die Werte aller <i>OPTIONS</i> -Parameter auf

Parameter, die einen Wert benötigen (die Tabelle ist nicht vollständig!):

Parameter	Bedeutung	Einheit	Default
<i>ABSTOL</i>	Genauigkeitsgrenze für Ströme	<i>A</i>	1 <i>pA</i>
<i>CHGTOL</i>	Genauigkeitsgrenze für Ladungen	<i>C</i>	0.01 <i>pC</i>
<i>GMIN</i>	Leitwert parallel zu jedem pn-Übergang (um die Konvergenz zu erleichtern)	<i>A/V</i>	1 <i>pS</i>
<i>ITL1</i>	Iterationsgrenze zur Bestimmung des Arbeitspunktes (.DC)		100/40 <sup>*)</sup>
<i>ITL2</i>	Iterationsgrenze zur Bestimmung des Arbeitspunktes (.DC)		50/20 <sup>*)</sup>
<i>ITL4</i>	Iterationsgrenze während der <i>TRAN</i> -Analyse		10
<i>PIVREL</i>	Relative Größe des Pivot-Elementes zu Lösung von Matrizen		0.001
<i>PIVTOL</i>	Absolute Größe des Pivot-Elementes zu Lösung von Matrizen		0.1p
<i>RELTOL</i>	Relative Genauigkeit für Spannungen und Ströme		0.001
<i>TNOM</i>	Nenntemperatur	<i>°C</i>	27 <i>°C</i>
<i>VNTOL</i>	Genauigkeitsgrenze für Knotenpotentiale	<i>V</i>	1 <i>μV</i>

<sup>\*)</sup> Der erste Wert gilt für **SPICE 3f4** und **LT-SPICE**, der zweite für **PSPICE**.

## 9.2 Temperatur

Ein ganz wichtiger Kontrollparameter ist die Temperatur. Leider bestehen auch hier wieder einige Unterschiede zwischen den drei Programmen **SPICE 3f4**, **PSPICE** und **LT-SPICE**. Die Default-Temperatur ist in den drei Programmen  $Temp = 27\text{ }^\circ\text{C}$  ( $\triangleq T = 300\text{ K}$ ).

In **SPICE 3f4** wird die Abweichung der Temperatur vom Default-Wert ( $27\text{ }^\circ\text{C}$ ) durch die Anweisung `.OPTIONS` angegeben. Beispiel:

```
.OPTIONS TEMP=25
```

Die Programme **PSPICE** und **LT-SPICE** bieten sehr viel mehr Komfort. Hier wird der Default-Wert durch die Anweisung `.TEMP` überschrieben, wobei auch eine Liste von Temperaturwerten erlaubt ist. Beispiele:

```
.TEMP 25
```

```
.TEMP -20 0 20 40 60
```

Außerdem gibt es in **PSPICE** und **LT-SPICE** die Möglichkeit, die Temperatur durch eine DC-Anweisung zu verändern; siehe hierzu **Abschnitt 7.1.2**, Seite 28. Beispiel: `.DC TEMP -50 150 5`

### Individuelle Temperatur

Die über die Anweisungen `.TEMP ...` oder `.DC TEMP ...` zugewiesene Temperatur kann als globale Betriebstemperatur angesehen werden. Sie gilt demnach für alle Bauteile der Schaltung. In **LT-SPICE** gibt es jedoch eine einfache Möglichkeit, Bauteilen auch eine individuelle Temperatur zuzuweisen. Diese wird dem Bauteil direkt als zusätzliches Attribut in der Form `Temp=<Wert>` angeheftet [14]. Der Wert darf dabei auch als Parameter in geschweiften Klammern stehen, und kann dann mittels einer Step-Anweisung variiert werden; siehe hierzu **Kapitel 10** (Seiten 41). Beispiel für einen Leistungstransistor IRFP240, der infolge höherer Belastung die Temperatur  $T_p = 85\text{ }^\circ\text{C}$  annehmen möge:

```
M4 VCC 3 Out Out IRFP240 Temp={TP}
.Param TP=85
```

### Nenntemperatur TNOM

*TNOM* ist die Temperatur, bei der die Bauelementparameter gemessen wurden. Dies erfolgt normalerweise bei dem Default-Wert  $TNOM = 27\text{ }^\circ\text{C}$  ( $\triangleq 300\text{ K}$ ). Wenn die Parameter bei einer *anderen* Temperatur gemessen wurden, kann der Default-Wert durch die Anweisung `.OPTIONS` überschrieben werden. Beispiel:

```
.OPTIONS TNOM = 20
```

- Wichtig ist, dass die Parameter aller temperaturabhängigen Bauelemente einer Schaltung für die *selbe Nenntemperatur* gelten. Wurden die Parameter der Bauelemente bei unterschiedlichen Temperaturen gemessen, müssen bei Verwendung der obigen Options-Anweisung alle Werte auf eine *einheitliche Nenntemperatur* umgerechnet werden!
- Bei **PSPICE** und **LT-SPICE** kann *TNOM* aber auch direkt in die Modell-Anweisung geschrieben werden und gehört damit zu den Modell-Parametern. Dies hat den großen Vorteil, dass SPICE die Umrechnung dann automatisch vornimmt.

## 9.3 Rechengenauigkeit

Die gewünschte Rechengenauigkeit kann durch die Anweisung `.OPTIONS` angepasst werden. Durch Setzen der Parameter werden die Default-Werte überschrieben. Beispiel:

```
.OPTIONS RELTOL=E-4 ABSTOL=0.1pA CHGTOL=0.005pC VNTOL=0.2uV
```

Die Einheiten brauchen in SPICE nicht angegeben zu werden; sie weisen aber auf den physikalischen Hintergrund der Größen hin. – Bedeutung der Parameter mit ihren Default-Werten siehe Abschnitt 9.1 (Seite 39).

## 10 Parameter-Variation in PSPICE und LT-SPICE

In **PSPICE** und in **LT-SPICE** kann jeder Bauteilwert oder auch jeder Wert in einer Modell-Anweisung durch einen Parameter (z.B.  $b$ ) angegeben werden. Der Parameter wird in der entsprechenden Bauteilzeile in der Netzliste bzw. in der Modell-Anweisung in geschweifte Klammern gesetzt (z.B.  $\{b\}$ ) und ihm dann mittels der Parameter-Anweisung `.PARAM` der gewünschte Zahlenwert (z.B.  $b = 190$ ) zugewiesen. Durch eine Step-Anweisung `.STEP` kann schließlich auch noch eine Variation des Parameters vorgenommen werden. Hierzu möge das Beispiel Q2 (Transistorverstärker, Seite 30) dienen:

```
* Q2d.Cir (Transistorverstärker)
.
.
.
RL 6 0 {a}
Q 4 3 5 C3ZN02
.MODEL C3ZN02 NPN (IS=1.1E-16 BF={b} VAF=105 KF=3E-15)
.PARAM a=22k
.PARAM b=190
.STEP PARAM b 100 300 50
.AC DEC 20 1Hz 10Meg
.PROBE
.END
```

Die beiden Parameter  $a$  und  $b$  in dem obigen Beispiel können selbstverständlich auch in einer einzigen Zeile angegeben werden, z.B.:

```
.PARAM a=22k b=190
```

In dem obigen Beispiel wird die Stromverstärkung des Transistors von  $BF = 100$  bis  $BF = 300$  in Schritten von 50 verändert. Die Step-Anweisung darf während eines Simulationslaufes nur auf *einen* Parameter angewendet werden.

Oft ist es sinnvoll, den Parameter nicht mit einer konstanten Schrittweite zu verändern. Für solche Fälle wird die Möglichkeit geboten, die gewünschten Parameter in Form einer *Liste* anzugeben. Die Zeile für das obige Beispiel könnte dann z.B. lauten:

```
.STEP PARAM b List 100 150 280
```

### Teilschaltungen (Subcircuits)

Mit dem Schlüsselwort `PARAMS` lassen sich auch in **Teilschaltungen** (Subcircuits) Parameter verändern. Dazu müssen zunächst in der `.SUBCKT`-Definition die entsprechenden Parameter mit dem Zusatz

```
PARAMS: Parameter1=Wert1 Parameter2=Wert2 ...
```

als Default-Werte erklärt werden:

```
.SUBCKT Name K1 K2 K3 ... PARAMS: Parameter1=Wert1 Parameter2=Wert2 ...
```

und statt der Zahlenwerte hinter den Bauelementen stehen die Parameter in geschweiften Klammern  $\{\}$ .

Die einzelnen Teilschaltungen (Bauteile mit dem Kennbuchstaben  $X$  in der Netzliste) erhalten dann dieselben Zusätze wie oben, nur mit den gewünschten *geänderten* Werten. Durch diese „neuen“ Werte werden die Werte in der `.SUBCKT`-Definition überschrieben.

Wird zusätzlich die Parameter-Anweisung `.PARAM` in Verbindung mit der Step-Anweisung verwendet, kann auch noch die Variation *eines* Parameters vorgenommen werden. Zur Verdeutlichung wählen wir die Filterschaltung **Bild 5** des Abschnitts 6.4, Seite 26. Sie erhält hier den Namen `Filter-B`:

```
* Filter-B.Cir
Vin 1 0 AC 1V
.
.
.
X1 1 2 0 TP-B
X2 3 5 0 TP-B  PARAMS: R1=47k R2=47k C1=5nF C2=5nF
X3 4 6 7 TP-B  PARAMS: R1={a} R2=22k C1=2nF C2=2nF
.SUBCKT TP-B 1 2 3 PARAMS: R1=10k R2=10k C1=1nF C2=1nF
R1 1 a {R1}
R2 a 2 {R2}
C1 a 3 {C1}
C2 2 3 {C2}
.ENDS

.PARAM a=22k
.STEP PARAM a 10k 50k 10k
.AC Dec 20 10Hz 100kHz
.PROBE
.END
```

In der Zeile

```
.SUBCKT TP-B 1 2 3  PARAMS: R1=10k R2=10k C1=1nF C2=1nF
```

werden durch „`PARAMS: R1=10k ...`“ die *Default-Werte* für die Elemente in der Teilschaltung angegeben. Die Namen der Parameter können *beliebig* gewählt werden; oft ist es jedoch am einfachsten, für die Parameternamen die Bauteilnamen zu übernehmen.

Den einzelnen Bauteilen `R1`, `R2`, ... in der Subcircuit-Netzliste sollen Zahlenwerte zugewiesen werden; die zugehörigen Parameternamen sind deshalb in *geschweifte Klammern* zu setzen: `{R1}`, `{R2}`, ...!

In der Hauptschaltung werden in den Zeilen `X2 3 5 ...` bzw. `X3 4 6 ...` durch den Zusatz `PARAMS: R1=47k ...` die Bauteilwerte der Teilschaltungen `X2` und `X3` *individuell* definiert. Für die Teilschaltung `X1` gelten weiterhin die Default-Werte `R1=10kΩ R2=10kΩ C1=1nF C2=1nF`.

Durch die Anweisung

```
.PARAM a=22k
```

wird der Parameter `R1={a}` für die Teilschaltung `X3` nochmals neu definiert und kann schließlich durch die Anweisung

```
.STEP PARAM a ...
```

auch noch schrittweise variiert werden.

Viel Erfolg mit dem wertvollen Simulationsprogramm SPICE.

## 11 Literatur

### 11.1 Literatur zu den Simulationsprogrammen SPICE 3f4, PSPICE und LT-SPICE

- [1] *T. Quarles, A.R. Newton, D.O. Pederson, A. Sangiovanni-Vincentelli*: SPICE 3f- User's Manual; May 1993, Department of Electrical Engineering and Computer Sciences, Berkeley, Ca., 94720. Datei: „SPICE 3f.pdf“ oder die etwas aktuellere Datei „SPICE.TXT“ im Ordner „SPICE“
- [2] *Wolfgang Mües, 38173 Sickte*: SPICE 3f4 für Windows, Version 1.5, 28. 10. 1997.  
Hier sind auch Erklärungen des Autors zusammengestellt. Die Datei findet man unter SPICE\README.WIN
- [3] *Wolfgang Mües, 38173 Sickte*: Änderungen in den einzelnen Versionen; 28. 10. 1997  
Verzeichnis SPICE\CHANGE.WIN
- [4] *Wolfgang Mües, 38173 Sickte*: SPICE-Kommandos, die in der Dokumentation nicht enthalten sind; 28. 06. 1995 Verzeichnis SPICE\DOC\UNDOC.WRI
- [5] OrCAD PSPICE A/D, User's Guide; OrCAD Inc. 1998
- [6] *Mike Engelhardt*: Simulationsprogramm LT-SPICE (SwitcherCAD III);  
<http://www.Linear.com/software>
- [7] *Paul W. Tuinenga*: SPICE, A Guide to Circuit Simulation and Analysis Using PSPICE; Prentice Hall, 1992
- [8] *Dietmar Ehrhardt, Jürgen Schulte*: Simulieren mit PSPICE; Vieweg Verlag 1995  
Die beiden Bücher [7] und [8] sind für das Verstehen der SPICE-Syntax besonders gut geeignet.
- [9] *K. S. Kundert*: The Designer's Guide to SPICE and SPECTRE; Kluwer Academic Publishers, 1995
- [10] *G. W. Roberts, A. S. Sedra*: SPICE; Oxford University Press 1997
- [11] *Robert Heinemann*: PSPICE Elektroniksimulation; Carl Hanser Verlag 2001  
Dieses Buch beschreibt in sehr anschaulicher Form den Umgang mit den PSPICE-Versionen 8 und 9, einschließlich der Werkzeuge SCHEMATICS und OrCAD CAPTURE. Es ist zum Einarbeiten in das komplexe Programm-Paket besonders gut geeignet.
- [12] SPICE-Parameter: Hochschul-Server; G:\docs\etech\Cordes\SPICE\Spicelib\
- [13] *Karl-Hermann Cordes*: Verschiedene Beispiele und Übungen zum Simulationsprogramm SPICE; Hochschul-Server; G:\docs\etech\Cordes\SPICE\
- [14] *Karl-Hermann Cordes*: Kurze Anleitung für das Simulationsprogramm LT-SPICE; Hochschul-Server; G:\docs\etech\Cordes\SPICE\LT-SPICE\LT-SPICE.pdf

### 11.2 Literatur zur Modellparameter-Bestimmung

- [15] *P. Antognetti, G. Massobrio*: Semiconductor Device Modeling with SPICE; Mc Graw Hill, 1988
- [16] *D. Ehrhardt*: Verstärkertechnik; Vieweg Verlag 1991
- [17] *I. Getreu*: Modeling the bipolar transistor; Tektronix Inc., Beaverton, Oregon 1976
- [18] *F. Sischka*: Eine Methode zur Bestimmung der SPICE-Parameter für bipolare Transistoren; AEÜ, 1985, Band 39, Heft 4, 225-232
- [19] *B. Schwaderer*: Bestimmung der Parameter zur Modellierung bipolarer Mikrowellentransistoren mit dem Analyseprogramm SPICE2; AEÜ, 1982, Band 36, Heft 7/8, 279-384
- [20] *Ed Oxner (Siliconix Inc.)*: Parameter extraction and estimation produce accurate JFET models; EDN, August 19, 1991, 137-144