Programy třídy SPICE a programy s určitou „SPICE-kompatibilitou“

1972 – SPICE (Simulation Program with Integrated Circuit Emphasis), Berkeley, USA

Dnes:

Programy třídy SPICE -

..HSpice AccuSim Spectre CADENCE-SPICE IG-SPICE I-SPICE Precise SmartSpice SPICE-Plus..

..PSpice..

..DSPICE B2SPICE AllSpice ICAP/4 Z-SPICE SPICE-IT! AimSpice TopSpice WinSpice..

pro PC

pro UNIXové pracovní stanice

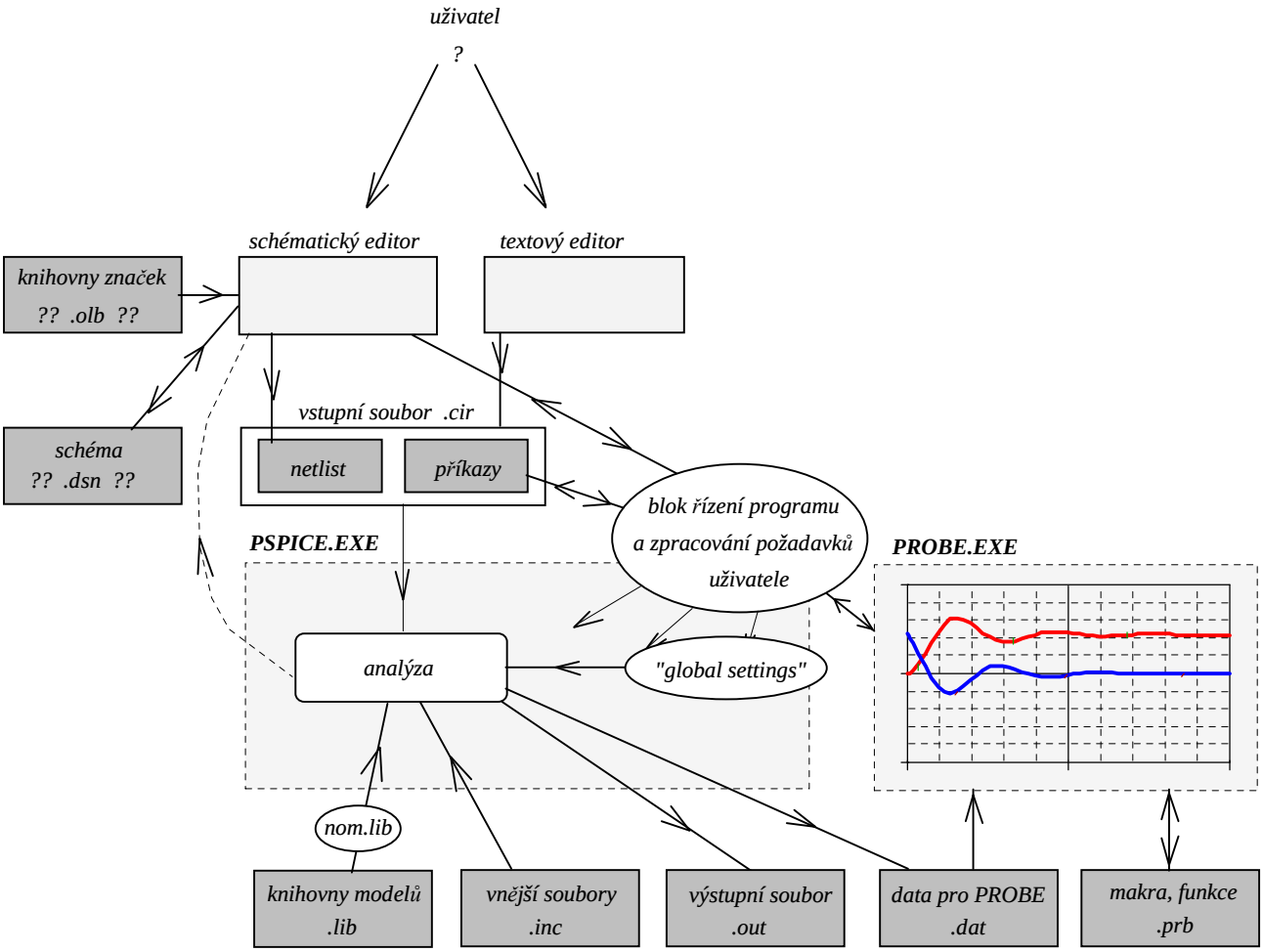
Programy, umožňující práci s modely SPICE nebo přímo využívat jazyka SPICE –

..MicroCap Electronics Workbench (MultiSIM) TINA..

pro PC

Standardy jazyka SPICE - ..SPICE2 (verze G6) SPICE3 .. PSPICE

Typická struktura simulačního programu založeného na PSPICE



Schéma, vstupní soubor, netlist, simulace [1]

VSTUPNÍ SOUBOR (*.cir)

Hlavička
Popis obvodu = netlist
Příkazy pro řízení simulace
.end

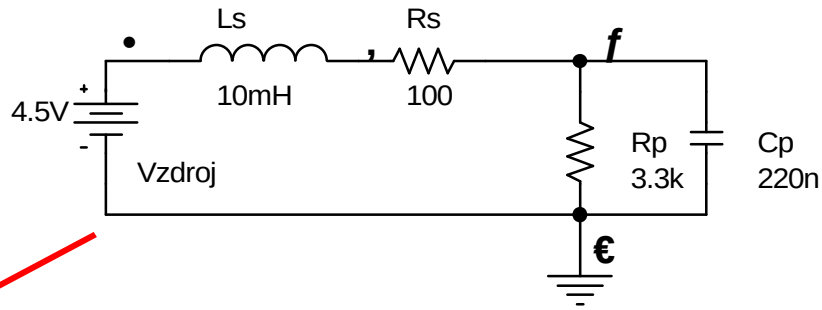
data o simulovaném obvodu

požadavky, jaké vlastnosti obvodu se mají simulovat

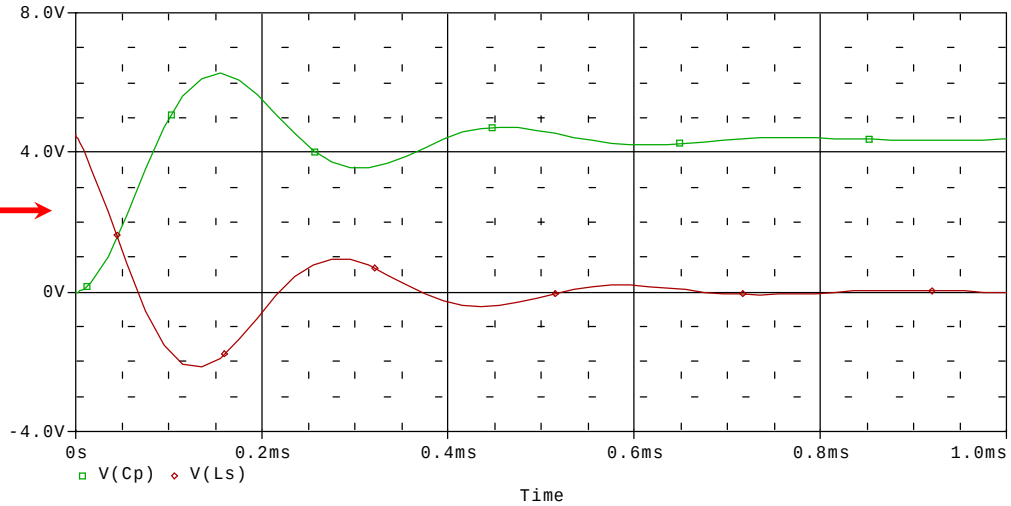
SIMULÁTOR

Simulace chování obvodu podle dat ve vstupním souboru

Pořadí zápisu není nutno dodržovat, je však doporučeno v zájmu přehlednosti




```
RLC obvod
*
Vzdroj 1 0 4.5V
Ls 1 2 10mH
Rs 2 3 100
Rp 3 0 3.3k
Cp 3 0 220nF
*
.TRAN 1u 1m SKIPBP
.PROBE V(Cp) V(Ls)
*
.end
```



Typy analýz, analyzační módy a režimy

Příklad členění pro MicroCap [2]. Škrty znázorňují omezení SPICE.

analyzační módy

- *Klasický*
- ~~*Probe (sonda)*~~

základní typy analýz

- *Transient (časová)...*
  +ss. prac. bod + Fourier. analýza a DSP
- *AC (kmitočtová, střídavá)*
  + šumová analýza a DSP
- *DC (stejnosměrná)*

analyzační režimy

- *Klasický*
- *Stepping (krokování jednoduché a vícenásobné)*
- *Teplotní analýza*
- *Performance Analysis (vyhodnocovací analýza)*
- *Monte Carlo a Worst Case (statistická analýza)*
- *Optimization (optimalizace)*

rozšiřující typy analýz

- ~~*Dynamic DC (dynamická stejnosměrná)*~~
- ~~*Dynamic AC (dynamická střídavá)*~~
- *Transfer Function (přenosová funkce)*
- *Sensitivity (citlivostní analýza)*

.....

Zvláštnost SPICE: Lze spustit analýzu, i když nejsou zadány požadavky na to, co se má analyzovat. Pak SPICE vypočítá to, co standardně počítá vždy – stejnosměrný pracovní bod.

Některá omezení SPICE pro každou analýzu:
Odpor, kapacita a indukčnost nemohou nabývat nulových hodnot.
Ke každému uzlu musí být připojeny aspoň 2 prvky.
V obvodu musí být uzel „0“ – „uzemnění“.
Každý uzel musí mít stejnosměrnou cestu na zem.

Základní pravidla pro tvorbu textových souborů (vstupní soubor, netlist, knihovny..)

Malá a velká písmena se nerozlišují.
Názvy všech obvodových prvků musí být jedinečné. Nelze např. nazvat dvě baterie stejně, i když mají stejná napětí.
První řádek ve vstupním souboru je „titulek“, SPICE jej ignoruje.
Prázdné řádky se ignorují.
Vícenásobné mezery „navíc“ v textu se ignorují (místo jedné mezery jich může být více po sobě).
Každý řádek (kromě prázdného řádku a prvního řádku) musí začínat jedním z těchto symbolů:

Klíčovým znakem určeným pro identifikaci obvodového prvku, např. R pro rezistor
* Obsah řádku je pak chápán jako poznámka.
\+ pokračování textu z předchozího řádku (pro lepší čitelnost v případě dlouhých řádek).
. Tečkou začínají dohodnuté příkazy.

Umístíme-li kamkoliv na řádek středník ; , pak vše vpravo od něj je chápáno jako poznámka.
Vstupní soubor musí končit řetězcem .END

Uzly můžeme označovat přirozenými čísly (původní standard SPICE), nebo textovými řetězci, které nesmí obsahovat mezery (umožňuje většina současných verzí SPICE). Jeden z uzlů musí být označen číslem 0.
Čísla vyjadřujeme v přirozeném tvaru (0.052) s desetinnou tečkou, nikoliv čárkou, nebo v exponenciálním tvaru (5.2E-2), nebo pomocí inženýrské notace (52m). V rámci řetězce se nesmí vyskytovat mezery. V inženýrské notaci jsou tyto zvláštnosti: M nebo m je mili, MEG je mega, u je mikro (velikost písmen je nevýznamná). **Textový řetězec za číslem se ignoruje (1u = 1uF)**.

Klíčové znaky (první znak v jméně) analogových součástek

R rezistor **C** kapacitor **L** induktor
K .. přiřazení magnetické vazby mezi induktory nebo mag. jádra induktoru (induktorům)

V .. nezávislý zdroj napětí ..**I** .. nezávislý zdroj proudu

E .. zdroj napětí řízený napětím
F.. zdroj proudu řízený proudem
G.. zdroj proudu řízený napětím
H.. zdroj napětí řízený proudem

D .. dioda

B .. tranzistor GaAsFET (pozor – od SPICE verze 3F5: **B** = univerzální nelineární zdroj!!!)
J .. tranzistor JFET
M .. tranzistor MOSFET
Q .. bipolární tranzistor
Z .. IGBT tranzistor

součástky vyžadující příkaz .MODEL

S .. spínač řízený napětím
W .. spínač řízený proudem

T .. vedení

X .. podobvod (subobvod, funkční blok, makro)

Definice součástky v netlistu, která nevyužívá příkaz .MODEL
(např. pasivní součástky RLC a zdroje)

<jméno součástky> <seznam uzlů, na které je připojena> <její vlastnosti> [upřesnění]

Ostré závorky < > značí, že obsah, který ohraničují, je povinný.
Hranaté závorky [ ] značí, že obsah, který ohraničují, je nepovinný.
Jméno součástky musí začínat jejím klíčovým znakem.

<u>Nejjednodušší možné syntaxe pasivních prvků R, L, C</u>:

	Příklad
Rxx..xx <N+> <N-> <odpor>	Rochranny 3 5 150; odpor 150 ohmu mezi uzly 3 a 5
Lxx..xx <N+> N-> <indukčnost>	Ltlumivka 1 4 9H; inductor 9H mezi uzly 1 a 4
Cxx..xx <N+> N-> <kapacita>	Capacitor 2 0 15u; kapacitor 15uF mezi uzly 2 a 0

Napětí i proud se uvažují v orientaci od uzlu N+ do N-.

<u>Nejjednodušší možné syntaxe nezávislých stejnosměrných zdrojů napětí a proudů</u>:

	Příklad
Vxx..xx <N+> <N-> <napětí>	Vbaterie 1 0 4.5V; baterie o napeti 4.5V mezi uzly 1 a 0
Ixx..xx <N+> N-> <proud>	Ix 3 12 2m; zdroj o proudu2mA mezi uzly 3 a 12

Napětí i proud se uvažují v orientaci od uzlu N+ do N- (pozor, není dodržována konvence o zdrojové orientaci čítacích šipek napětí a proudu!!!).
Definice součástky v netlistu, která využívá příkaz .MODEL

(zejména diody a tranzistory)

<jméno součástky> <seznam uzlů, na které je připojena> <jméno modelu> [upřesnění]

Seznam uzlů musí nyní být zapsán v dohodnutém pořadí, neboť např. záměna anody a katody by měla pro funkci obvodu fatální důsledky.
Příklady seznamu uzlů pro diodu a bipolární tranzistor:
<seznam uzlů> = <anoda katoda> … dioda
<seznam uzlů> = <kolektor báze emitor> … bipolární tranzistor

Jméno modelu si volíme libovolně. Model pak přesně specifikujeme příkazem .MODEL, který může být umístěn buď přímo v netlistu nebo v jiném souboru (většinou knihovně), na který „vidí“ SPICE.

Obecná syntaxe modelu:

.MODEL <jméno modelu> <kód modelu> [( ] parametry modelu oddělené mezerami [ )]

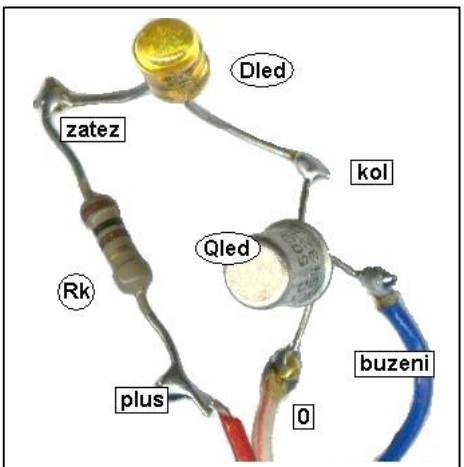
Kulaté závorky jsou nepovinné.
Vybrané kódy modelu:

D ..dioda, **NPN** ..bipolární tranzistor NPN, ..**PNP** ..bipolární tranzistor PNP.

Nepovinně lze model přiřadit i pasivním prvkům, např.:

CAP .. kapacitor, **RES** ..rezistor, **IND** ..induktor

Příklad, převzatý z [1]:



LED indikace

```
LED indikace
*
*zapojeni
*
Rk plus zatez 100
Dled zatez kol ledka
Qled kol buzeni 0 KC509
*
*modely soucastek
*
.model ledka D (N=1.7 RS=.7 CJO=23.9P
+ IS=85.3p BV=6 IBV=10U VJ=0.75 TT=4.32U)
.model KC509 NPN (BF=1.50K BR=2 IS=12.0P
+ CJC=2.80P CJE=1.40P RC=1.01U VAF=100
+ TF=718.4P TR=10N MJC=375.7M VJC=690.0M
+ MJE=375.77M VJE=690.00M NF=1.259
+ ISE=33.7P ISC=0.001566F IKF=9.44M
+ IKR=4.62K NE=2.18 RE=32.4M VTF=113.5
+ ITF=533.9K XTF=19.25M)
.end
```

Parametry modelů lze nalézt v manuálech. Každý typ modelu má číselnou sadu svých tzv. implicitních hodnot parametrů. Pokud daný parametr nedefinujeme příkazem .MODEL, platí implicitní hodnota. Pokud v příkazu neuvedeme žádný parametr, modelujeme součástku s implicitními parametry.

Některé základní příkazy

.TEMP .. definice teploty (temperature)

.TEMP <výčet teplot oddělených mezerami>

Příklady:

.TEMP 58 ; analýza proběhne jednou, pro teplotu 58 stupňů Celsia

.TEMP -28 11 54.5 ; analýza proběhne opakovaně pro teploty -28, 11 a 54,5 stupňů Celsia

.PARAM .. definice globálních parametrů (proměnných)

.PARAM <název parametru>=<hodnota parametru>

Při použití parametru ve výrazu se parametr musí umístit do složených závorek { }.

Příklady:

.PARAM odpor1=1k odpor2=5.2k kapacita=5nF

Rparalelni vstup zem {odpor1*odpor2/(odpor1+odpor2)}

.STEP .. krokování, vícenásobná analýza pro různé hodnoty krokované proměnné

.STEP <název proměnné> LIST <výčet hodnot oddělených mezerami>
.STEP [LIN] < název proměnné > <počáteční hodnota> <konečná hodnota> <krok>
.STEP OCT < název proměnné > <počáteční hodnota> <konečná hodnota> <počet bodů>
.STEP DEC < název proměnné > <počáteční hodnota> <konečná hodnota> <počet bodů>

LIN (nepovinné): lineární krokování, t.j. další hodnota=současná hodnota+krok
OCT, DEC: logaritmické krokování, t.j. další hodnota=současná hodnota*číslo větší než 1.
Počet bodů u logaritmického krokování: číslo větší než 1 se zvolí tak, aby se každá oktáva (OCT) nebo dekáda (DEC) rozdělila na daný počet bodů.
Dekáda je interval ohraničený hodnotami, jejichž poměr je 10, např. 8 a 80.
Oktáva je interval ohraničený hodnotami, jejichž poměr je 2, např. 8 a 16.

Jaké jsou možné názvy krokované proměnné:

- Název nezávislého zdroje napětí nebo proudu. Krokuje se napětí nebo proud tohoto zdroje.
- Typ modelu a jméno modelu, následované krokovaným parametrem v závorkách ( ).
- Teplota TEMP; Toto krokování nelze současně používat s příkazem .TEMP.
- Globální parametr, formát **PARAM=<název parametru>**. Rovnítko není povinné.

Příklady:

.STEP Vin 0V 3V 1V; analýza proběhne pro napětí zdroje Vin = 0V, 1V, 2V a 3V

.STEP PARAM=odpor LIST 1k 0.1K 2meg; analýza proběhne opakovaně pro 1kΩ, 100Ω,2MΩ

.STEP DEC NPN Q1(BF) 10 1000 1; krokuje se parametr BF tranzistoru Q1, BF=10, 100, 1000

.STEP TEMP 10 100 20; krokuje se teplota v hodnotách 10, 30, 50, 70 a 90 stupňů Celsia

Současně nelze krokovat více proměnných – podstatné omezení SPICE.

.LIB .. napojení knihovny modelů na vstupní soubor

.LIB [název souboru]

Příklady:

.LIB pokus.lib	; napojení na soubor s názvem pokus.lib
.LIB “c:\data\ana.lib“	; napojení na soubor v adresáři, na který normálně SPICE „nevidí“
.LIB NOM.LIB	;napojení na všechny knihovny, které jsou vypsány v souboru NOM.LIB
.LIB	;stejný význam jako .LIB NOM.LIB (NOM.LIB je tzv. Master Library)

Co všechno může být obsaženo v souboru *.lib:
Poznámky
Definice modelů (.MODEL)
Definice podobvodů (.SUBCKT)
Definice globálních parametrů (.PARAM)
Definice vlastních funkcí (.FUNC)
Příkazy .LIB

.INC .. včlenění obsahu vnějšího souboru

.INC <název souboru>

Příklady:

.INC zesilovac.cir	začlenění celého netlistu do vstupního souboru
.INC “c:\data\pok.txt“	

Význam .INC:
Efekt je stejný, jako kdyby se v místě příkazu .INC objevil text z včleněného souboru. Například ve vstupním souboru místo celého netlistu umístíme pouze odkaz na něj.

Co nesmí být obsaženo ve včleňovaném souboru:
Úvodní hlavička, pokud není označena jako poznámka (* nebo ;)

Je povoleno „vnořování“ těchto souborů do 4 úrovní.

.OP .. způsobí detailní výpis stejnosměrného pracovního bodu do výstupního souboru

Příklad:
.OP

Pozor, nejde o příkaz spouštějící analýzu, pouze jde o výpis výsledků. Pracovní bod se počítá vždy, i při chybějícím příkazu .OP.
Co se vypisuje bez příkazu .OP:
Uzlová napětí, proudy přes zdroje napětí, celkový ztrátový výkon.
Co se navíc vypisuje s příkazem .OP:
Linearizované malosignálové parametry polovodičových prvků a nelineárních řízených zdrojů.

Práce s podobvody

Podobvod (subcircuit) je relativně samostatný obvod, který komunikuje s okolím prostřednictvím konkrétních vývodů. Nejprve jej musíme definovat (*deklarovat*), tj. vytvořit jeho model, uložit jej do dané knihovny nebo přímo do vstupního souboru, a pak jej můžeme využívat (*volat*) jako každou jinou součástku.
Pokud podobvod deklarujeme přímo ve vstupním souboru analyzovaného obvodu: Model podobvodu bude k dispozici jen pro analýzu tohoto obvodu.
Pokud podobvod umístíme do externí knihovny knihovna.lib: Model podobvodu bude k dispozici každému, kdo může využívat tuto knihovnu. Na knihovnu se napojíme tak, že do vstupního souboru umístíme příkaz

.lib knihovna.lib

Podobvod může být sestavený ze součástek, jejichž parametry můžeme měnit při volání podobvodu. Pak se jedná o volání s parametrem (parametry). Této možnosti se příliš nevyužívá.

Deklarace podobvodu - syntaxe:

.SUBCKT <jméno podobvodu> <seznam vývodů> [params: <parametr=číselná hodnota>]

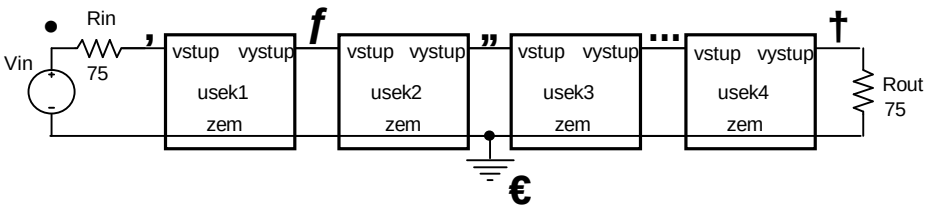
<popis vnitřního zapojení>

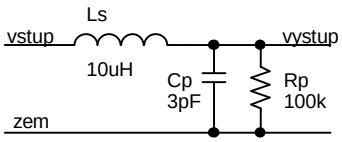
.ENDS

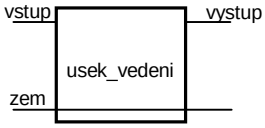
Použití (volání) podobvodu - syntaxe:

Xxx..xx <seznam uzlů podobvodu> <jméno podobvodu> [params: <parametr>=<číselná hodnota>]

Příklad – podobvod bez parametrů:







model vedeni
Vin 1 0 1V
Rin 1 2 75ohmu
Xusek1 2 3 0 usek_vedeni
Xusek2 3 4 0 usek_vedeni
Xusek3 4 5 0 usek_vedeni
Xusek4 5 6 0 usek_vedeni
Rout 6 0 75ohmu
*definice podobvodu
.subckt usek_vedeni vstup vystup zem
Ls vstup vystup 10uH
Cp vystup zem 3pF
Rp vystup zem 100kohmu
.ends usek_vedeni
.end

Nepovinné, ale je dobrým zvykem zařazovat za **.ends** název „končícího“ podobvodu. Přispívá to k větší přehlednosti.

Příklad – podobvod s parametry:

…….
Xusek1 2 3 0 usek_vedeni params: Ls=10u Cp=3p Rp=100k
……..
.subckt usek_vedeni vstup vystup zem params: indukcnost=10u kapacita=3p odpor=100k
Ls vstup vystup {indukcnost}
Cp vystup zem (kapacita}
Rp vystup zem {odpor}
.ends usek_vedeni
.end

Zabudované funkce PSpice pro modelování prvků - výběr

funkce	význam	poznámka
ABS(x)	$\lvert x \rvert$	
COS(x)	$\cos(x)$	x je v radiánech
DDT(x)	dx/dt	jen u analýzy TRANSIENT
EXP(x)	e^x	
IF(a,x,y)	x když a=TRUE, jinak y	a je logický výraz, například IF(I(L1)>0,V(1),-V(1))
IMG(x)	imaginární část x	
LIMIT(x,MIN, MAX)	MIN když x<MIN, MAX když x>MAX, jinak x	oboustranný okrajovač
LOG(x)	$\ln(x)$	základ e
LOG10(x)	$\log(x)$	základ 10
M(x)	$\lvert x \rvert$	modul komplexního čísla x
MAX(x,y)	větší z čísel x a y	
MIN(x,y)	menší z čísel x a y	
P(x)	argument komplexního čísla	
PWR(x,y)={x**y}	$\lvert x \rvert^y$	mocninná funkce
PWRS(x,y)	$\lvert x \rvert^y$ když x>0, $-\lvert x \rvert^y$ když x<0	mocninná funkce se znaménkem
R(x)	reálná část x	
SDT(x)	$\int_0^t x(t)dt$	jen u analýzy TRANSIENT
SGN(x)	1 pro x>0, -1 pro x<0, 0 pro x=0	znaménková funkce
SIN(x)	$\sin(x)$	x je v radiánech
STP(x)	1 když x>0, jinak 0	jednotkový skok
SQRT(x)	$\sqrt{x}$	
TABLE(x,x1,y1,x2,y2,..,xn,yn)	Graf funkce vznikne pospojováním bodů o souřadnicích [xk,yk] lomenou čarou. Pro x<x1 je výsledek y1, pro x>xn je výsledek yn.	

Vytváření vlastních funkcí – příkaz .FUNC

.FUNC <jméno> ([argumenty]) {<vzorec>}

Před vzorcem je přípustné i rovnítko, složené závorky nejsou nutné.

<u>Příklady</u>:

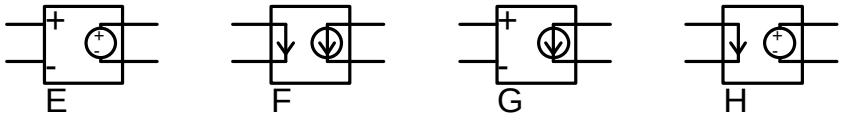
.FUNC LIMITER(x,gain) {LIMIT(gain*x,-15,15)} ; omezovač se zesílením GAIN, signál x větší + než +15V a menší než -15V ořeže

.FUNC MIN3(A,B,C) {MIN(A,MIN(B,C))} ; minimální z čísel A, B, C

Funkce lze psát do zvláštního souboru. Ze vstupního souboru na ně bude „vidět“ po vložení příkazu **.INC soubor**

Řízené (závislé) zdroje pro behaviorální modelování (ABM –Analog Behavioral Modeling)

E..zdroj napětí řízený napětím
F..zdroj proudu řízený proudem
G..zdroj proudu řízený napětím
H..zdroj napětí řízený proudem



Napětí na svorkách i proudy uvnitř zdrojů jsou orientovány od svorky + do svorky -.

<u>Zdroje E a G</u> mají stejnou syntaxi. Níže vysvětleno zejména pomocí zdrojů E.

1) Zdroj s řídicími svorkami, lineární zařízení:

E<název> <výstup+> <výstup-> <vstup+> <vstup-> <přenos>

Vzorec: výstup = přenos*vstup

<u>Příklady</u>:
Ezesil 2 0 1 0 100 ; lineární zesilovač, vstup 1-0, výstup 2-0, zesílení 100
GOTA out 0 plus minus {gm} ; OTA zesilovač s výstupem out, vstupy plus-minus
+a transkonduktancí gm, která je definována jako globální parametr

Přenos může být i vzorec sestavený z konstant a globálních parametrů.

2) Zdroj s řídicími svorkami, nelineární zařízení:

E<název> <výstup+> <výstup-> POLY(<dimenze>) {<vstup+> <vstup->} {<koeficient>}

seznam dvojic řídicích svorek seznam koeficientů polynomu

Vzorec: Nelineární polynomiální závislosti výstupu na řídicích vstupech, viz tabulka

exp. u	p0	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	…
x1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	…
x1 x2		1 0	0 1	2 0	1 1	0 2	3 0	2 1	1 2	0 3	4 0	3 1	2 2	1 3	0 4	…
x1 x2 x3		1 0 0	0 1 0	0 0 1	2 0 0	1 1 0	1 0 1	0 2 0	0 1 1	0 0 2	3 0 0	2 1 0	2 0 1	1 2 0	1 1 1	…
…	…															

Dimenze je rovna počtu řídicích vstupů.
a) Dimenze = 1, je jeden vstup x1, výstup y:

$$y = p_0 + p_1 x_1 + p_2 x_1^2 + \ldots$$

<u>Příklady</u>:
Ekvadrat 2 0 POLY(1) 1 0 0 0 0.1 ; vstupní napětí mezi 1 a 0, výstupní napětí mezi 2 a 0. Jde o
+kvadrátor o vzorci V(2)=V(1)*V(1)/10.
Ex out 0 POLY(1) in1 in2 0.5 -0.5 ; vstupní napětí mezi in1 a in2, výstupní napětí mezi out a 0.
+Funkční měnič V(out)=(1-V(in1,in2))/2.

b) Dimenze = 2, jsou dva vstupy x1 a x2, výstup y:

$$y = p_0 + p_1 x_1 + p_2 x_2 + p_3 x_1^2 + p_4 x_1 x_2 + p_5 x_2^2 + p_6 x_1^3 + p_7 x_1^2 x_2 + p_8 x_1 x_2^2 + p_9 x_2^3 + \ldots$$

<u>Příklady</u>:
Emulti 3 0 POLY(2) 1 0 2 0 0 0 0 0 .1 ; dvojice vstupních napětí mezi 1-0 a 2-0, výstupní napětí
+mezi 3-0. Jde o násobičku V(3)=V(1)*V(2)/10.
Esum out 0 POLY(2) in1 0 in2 0 0 1 1 ; dvojice vstupních V(in1) a V(in2), výstupní napětí V(out).
+Sumátor V(out)=V(in1)+V(in2).

3) Zdroj bez řídicích svorek, výstupní napětí je modelováno vzorcem:

E<název> <výstup+> <výstup-> VALUE [=] {<vzorec>}

Rovnítko je nepovinné.

Příklady:

Ekvadrat out 0 VALUE={V(in)^2/10}	; kvadrátor o vzorci V(out)=V(1)*V(1)/10.
Ex 2 0 VALUE {10*(-0.5+STP(V(1)))}	; komparátor: V(2) je buď +5V nebo -5V podle toho, +zda V(1) je kladné nebo záporné.

4) Zdroj bez řídicích svorek, výstupní napětí je modelováno po úsecích lineární závislostí:

E<název> <výstup+> <výstup-> TABLE {<vzorec>} [=] (seznam dvojic <xk yk>)

Příklady:

Ekvadrat out 0 VALUE={V(in)^2/10}	; kvadrátor o vzorci V(out)=V(1)*V(1)/10.
Ex 2 0 VALUE {10*(-0.5+STP(V(1)))}	; komparátor: V(2) je buď +5V nebo -5V podle toho, +zda V(1) je kladné nebo záporné.
Ex 2 0 TABLE {V(1)} (-4 -5 4 5)	; pro V(1)<-4V je V(2)=-5V, pro V(1)>4V V(2)=5V +pro -4V<V(1)<4V je graf dán spojnicí obou bodů
Ex 2 0 TABLE {10*V(1)} (-4 -5 4 5)	; pro 10*V(1)<-4V je V(2)=-5V, + pro 10*V(1)>4V V(2)=5V + pro -4V<10*V(1)<4V je graf dán spojnicí bodů + [-0.4, -5], [0.4, 5]

Poznámka:

Tyto zápisy jsou ekvivalentní: (-4 -5 4 5) a (-4 -5) (4 5) a (-4, -5, 4, 5) a (-4, -5) (4, 5) atd.

5) Zdroj bez řídicích svorek, výstupní signál je modelováno klíčovými slovy LAPLACE nebo FREQ:

Bude popsáno později.

Zdroje F a H mají stejnou syntaxi. Níže vysvětleno zejména pomocí zdrojů F.

Jsou to zdroje napětí (H) a proudu (F), řízené **proudem**. Proud je snímán vždy na nezávislém zdroji napětí, který musí být specifikován v definici řízeného zdroje namísto řídicích svorek. Chceme-li, aby se tento zdroj choval jako ideální „ampérmetr“, musíme jeho napětí položit rovné nule.

1) Lineární řízený zdroj:

F<název> <výstup+> <výstup-> V<název řídicího zdroje> <přenos>

Vzorec: výstup = přenos*proud řídicím zdrojem

Příklad:

Fzesil 2 0 Vin 100	; lineární proudový zesilovač, vstup 1-0, výstup 2-0, zesílení 100
Vin 1 0 0V	; pomocný zdroj napětí mezi svorkami 1-0 o nulovém napětí

2) Nelineární řízený zdroj:

F<název> <výstup+> <výstup-> POLY(<dimenze>) {V<název říd. zdroje>} {<koeficient>}

seznam řídicích zdrojů seznam koeficientů polynomu

Vzorec: Nelineární polynomiální závislosti výstupu na řídicích vstupech, viz zdroje E a G

Příklad:

Fdif out 0 POLY(2) V1 V2 0 1 -1 ; výstupní proud je rozdíl proudů tekoucích přes zdroje V1 a V2

Řízené (závislé) spínače pro behaviorální modelování

S .. spínač řízený napětím
W .. spínač řízený proudem

Spínač „S“:

S<název> <spínač+> <spínač-> <řízení+> <řízení-> <jméno modelu>

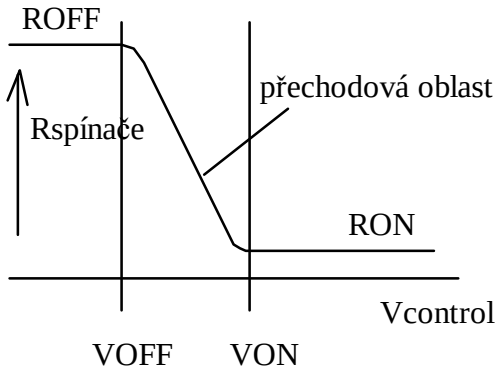
Definice modelu spínače:

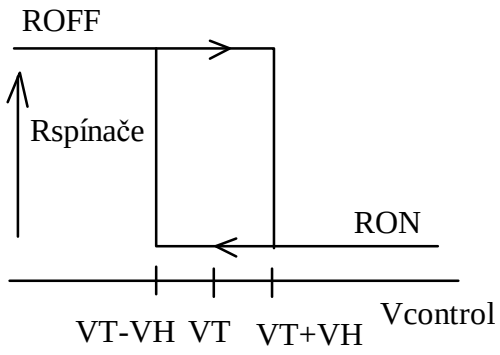
.MODEL <jméno modelu> VSWITCH [parametry modelu]

V PSpice existují dva různé modely spínače: „Variable-Resistance Switch“ (VRS) a „Short-Transition Switch“ (STS). PSpice rozezná typ modelu podle jeho zadaných parametrů. Implicitním modelem je model VRS.

VRS			
parametr	popis	jednotka	default
ROFF	odpor v rozpojeném stavu	Ohm	1MEG
RON	odpor v sepnutém stavu	Ohm	1
VOFF	vypínací napětí	V	0
VON	spínací napětí	V	1

STS			
parametr	popis	jednotka	default
ROFF	odpor v rozpojeném stavu	Ohm	1T
RON	odpor v sepnutém stavu	Ohm	1
VT	prahové napětí	V	0
VH	hysterezí napětí	V	0
TD	časové zpoždění	s	0





Parametr TD znamená časové zpoždění od přivedení překlápěcí úrovně napětí k faktickému přepnutí spínače.

Spínač „W“:

S<název> <spínač+> <spínač-> V<název řídicího zdroje> <jméno modelu>

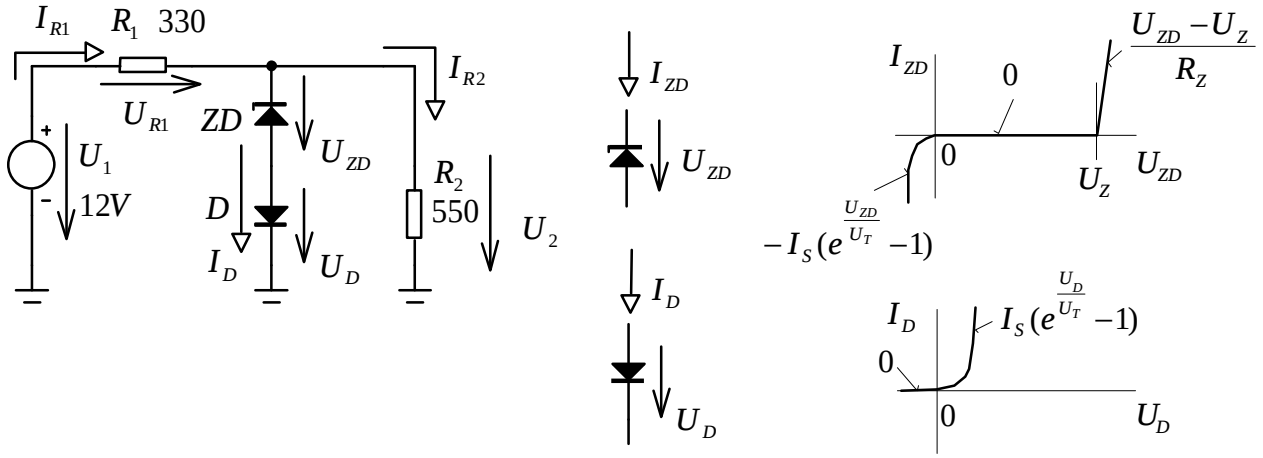
Definice modelu spínače:

.MODEL <jméno modelu> ISWITCH [parametry modelu]

Spínač se nyní řídí proudem, který teče řídicím zdrojem napětí. Parametry modelu jsou obdobné jako u modelu VSWITCH, pouze napětí VOFF, VON, VT, VH jsou nahrazena proudy IOFF, ION, IT, IH.

Výpočet ss pracovního bodu – základní analýza

Analýza stabilizátoru napětí. ZD je Zenerova dioda 1N3826 o napětí U_Z = 5,1 V, D je „klasická“ křemíková dioda 1N459.



$U_Z = 5{,}1V$ … Zenerovo napětí.

$R_z = 7\Omega$ … diferenciální odpor Zenerovy diody v oblasti Zenerova průrazu.

$I_S = 3{,}5.10^{-13} A$ … saturační proud; u obou diod je uvažována stejná hodnota.

$U_T = kT/q \approx 25{,}8mV$ … teplotní napětí při absolutní teplotě T = 300 K (27°C), k je Boltzmannova konstanta $1{,}38.10^{-23}$ J/K, q je elementární elektrický náboj $1{,}602.10^{-19}$ C.

„Ruční“ řešení:

$$U_{ZD} = R_Z I_D + U_Z\,,\; U_D = U_T \ln(1+\frac{I_D}{I_S})\,.$$

$$U_2 = R_Z I_D + U_Z + U_T \ln(1+\frac{I_D}{I_S})\,,\; I_D \geq 0 \qquad (1)$$

$$U_2 = R_2 I_{R2} = R_2(I_{R1} - I_D) = R_2\left(\frac{U_1 - U_2}{R_1} - I_D\right).$$

$$U_2 = \frac{R_2}{R_1+R_2}U_1 - \frac{R_1 R_2}{R_1+R_2} I_D\,.$$

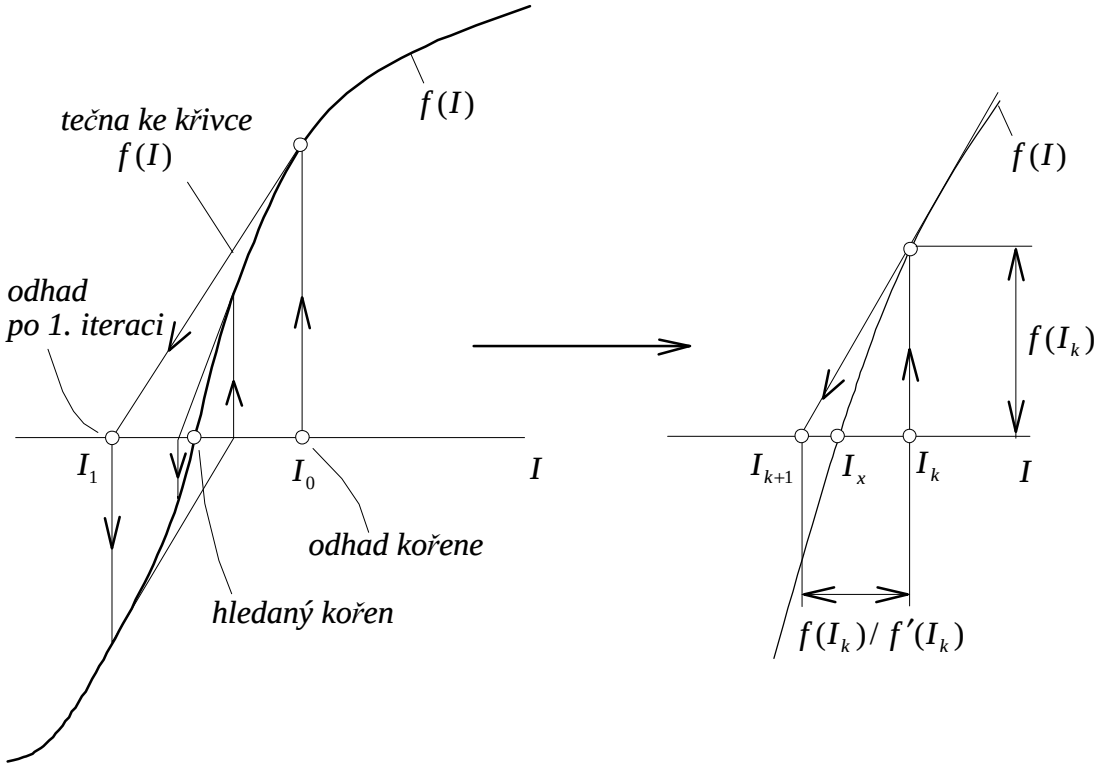
Dosadíme do levé strany rovnice (1). Tím dostáváme rovnici o jediné neznámé I_D, která získá po úpravě tvar

$$\left(R_Z + \frac{R_1 R_2}{R_1+R_2}\right) I_D + U_T \ln\left(1+\frac{I_D}{I_S}\right) + U_Z - \frac{R_2}{R_1+R_2}U_1 = 0\,. \qquad (2)$$

Na levé straně rovnice je nelineární funkce proudu I_D, kterou označíme $f(I_D)$. Hledáme takový proud, pro nějž je funkce nulová. Po dosazení číselných hodnot lze (2) zapsat takto:

$$f(I_D) \approx I_D + 1{,}2118.10^{-4} \ln(1+2{,}8571.10^{12} I_D) - 1{,}1125.10^{-2} = 0\,. \qquad (3)$$

Rovnici vyřešíme Newtonovou iterační metodou, jejíž princip je ilustrován na obr. Vyjdeme z počátečního odhadu řešení I_0 a na základě naznačené konstrukce dospějeme k „zpřesněnému“ odhadu I_1. Opakujeme tak dlouho, až se posloupnost I_0, I_1, I_2, … ustálí v rámci akceptovatelné chyby na hodnotě hledaného řešení I_x.



Z geometrického významu derivace funkce f v bodě I_k vyplývá známý vztah (Newtonova formule) mezi řešeními v sousedních iteracích:

$$I_{k+1} = I_k - \frac{f(I_k)}{f'(I_k)}. \tag{4}$$

Aplikujeme-li tento vzorec na naši konkrétní funkci (3), dostaneme po úpravě výpočetní algoritmus

$$I_{k+1} = I_k - \frac{I_k + 1{,}2118.10^{-4}\ln(1+2{,}8571.10^{12} I_k) - 1{,}1125.10^{-2}}{1+\dfrac{3{,}4624.10^{8}}{1+2{,}8571.10^{12} I_k}}. \tag{5}$$

Zvolíme-li počáteční odhad I_0 = 0A, postupným dosazováním do (4), dostaneme posloupnost

(0, 3.2504e-8, 2.9345e-6, 2.4897e-4, 1.9975e-2, 1.3034, 7.9644, 8.3584, 8.3585, 8.3585, …) *m*A.

Hledaný proud diodami je tedy (v rámci přesnosti použitých modelů) s přesností na 4 platné cifry

I_D = 8,3585 *m*A.

Můžete se přesvědčit o tom, že při počátečním odhadu řešení 10 *m*A se ustálí iterační algoritmus na tomto řešení hned v 2. kroku.

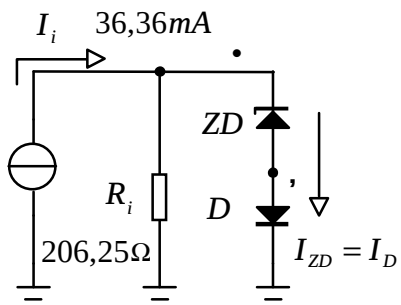
Pro danou hodnotu proudu I_D pak z modelů diod vycházejí napětí

$$U_{ZD} = 5{,}159V\ ,\ U_D = 0{,}617V\ .$$

Algoritmický postup při řešení simulačním programem

Simulační program nejprve algoritmicky sestaví obvodové rovnice metodou uzlových napětí a pak počítá všechny neznámé, tj. uzlová napětí. Program sestaví tolik nelineárních rovnic, kolik je uzlových napětí, a hledá iterací stejný počet neznámých napětí. Iterační metoda tedy musí být zobecněná pro více proměnných. Nazývá se *Newtonova-Raphsonova iterační metoda*. V různých modifikacích je zabudována do všech stávajících simulačních programů do procedur pro hledání stejnosměrných pracovních bodů.

Pro jednoduchost převedeme zdroj napětí v stabilizátoru na ekvivalentní zdroj proudu:



Pro obvod platí dvě rovnice 1. Kirchhoffova zákona s neznámými uzlovými napětími U_1 a U_2:

$$\begin{array}{c} \bullet \\ , \end{array} \begin{array}{|c|} \hline f_1(U_1,U_2) \\ \hline f_2(U_1,U_2) \\ \hline \end{array} = \begin{array}{|c|} \hline G_i U_1 + I_{ZD} - I_i \\ \hline I_D - I_{ZD} \\ \hline \end{array} = \begin{array}{|c|} \hline 0 \\ \hline 0 \\ \hline \end{array} \qquad (6)$$

vektor $\boldsymbol{f}$

kde kladné proudy diodami můžeme vyjádřit rovnicemi

$$I_{ZD} = G_Z(U_1 - U_2 - U_Z),\ I_D = I_S\left(e^{\frac{U_2}{U_T}} - 1\right). \qquad (7)$$

Vektor $\boldsymbol{f}$ tedy po dosazení (7) do (6) vypadá takto:

$$\boldsymbol{f} = \begin{array}{|c|} \hline G_i U_1 + G_Z(U_1 - U_2 - U_Z) - I_i \\ \hline I_S\left(e^{\frac{U_2}{U_T}} - 1\right) - G_Z(U_1 - U_2 - U_Z) \\ \hline \end{array} = \begin{array}{|c|} \hline 0 \\ \hline 0 \\ \hline \end{array} \qquad (8)$$

Úkolem iterační metody je nalézt takové hodnoty U_1 a U_2, abychom vynulovali vektor $\boldsymbol{f}$, tedy abychom současně vynulovali obě dílčí funkce f_1 a f_2.

Newtonova-Raphsonova metoda je zobecněním „jednorozměrné“ Newtonovy metody a je tedy rovněž založena na vzorci (4), který je však nyní zobecněný pro více proměnných. Pro náš případ dvou proměnných vychází:

$$\begin{array}{|c|} \hline U_{1,k+1} \\ \hline U_{2,k+1} \\ \hline \end{array} = \begin{array}{|c|} \hline U_{1,k} \\ \hline U_{2,k} \\ \hline \end{array} - \left(\begin{array}{|c|c|} \hline \frac{\partial f_{1,k}}{\partial U_1} & \frac{\partial f_{1,k}}{\partial U_2} \\ \hline \frac{\partial f_{2,k}}{\partial U_1} & \frac{\partial f_{2,k}}{\partial U_2} \\ \hline \end{array} \right)^{-1} \cdot \begin{array}{|c|} \hline f_{1,k} \\ \hline f_{2,k} \\ \hline \end{array} \qquad (9)$$

Obecně

$$\boldsymbol{U}_{k+1} = \boldsymbol{U}_k - \boldsymbol{J}^{-1} \cdot \boldsymbol{f}_k$$

Čtvercová matice $\boldsymbol{J}$ se nazývá *Jakobiho matice*, její determinant *jakobián*. Derivujeme-li funkci (8) podle U_1 a U_2, dostaneme pro analyzovaný obvod *Jakobiho matici* ve tvaru

$$\boldsymbol{J} = \begin{array}{|c|c|} \hline G_i + G_Z & -G_Z \\ \hline -G_Z & \frac{I_S}{U_T} e^{\frac{U_2}{U_T}} + G_Z \\ \hline \end{array} = \begin{array}{|c|c|} \hline 0{,}1477 & -0{,}1429 \\ \hline -0{,}1429 & 1{,}3566.10^{-11} e^{38{,}76 U_2} + 0{,}1429 \\ \hline \end{array} \qquad (10)$$

Při algoritmickém řešení stejnosměrných poměrů v nelineárním obvodu tedy simulační program postupuje zhruba takto:

Algoritmus stejnosměrné analýzy nelineárních obvodů:

1) Nastaví se počáteční hodnota vektoru neznámých veličin, tj. uzlových napětí.
2) Na základě Kirchhoffových rovnic pro nezávislé uzly, modelů součástek a aktuálních hodnot uzlových napětí se vyplní prvky vektoru $\boldsymbol{f}$ (8) a prvky Jakobiho matice (10).
3) Pomocí rekurentní Newtonovy-Raphsonovy formule (9) se vypočte nový odhad uzlových napětí.
4) Body 2 a 3 se opakují v cyklu tak dlouho, dokud není splněno zadané chybové kritérium.

Kdybychom si daný postup vyzkoušeli naprogramovat například v MATLABu, zjistili bychom, že při počátečním nastavení $[U_1\ U_2] = [0\ 0]$ bude algoritmu trvat 92 iterací, než se v rámci 4 platných cifer ustálí na řešení

$$[U_1\ U_2] = [5.7751\text{V}\ 0.6165\text{V}].$$

Problémy s konvergencí - úvod

Nelineární obvody mohou mít několik stejnosměrných pracovních bodů. Např. bistabilní klopný obvod s 2 tranzistory T1 a T2 má 3 pracovní body – T1 otevřen a T2 zavřen, T1 zavřen a T2 otevřen, T1 i T2 částečně otevřeny. První dva mohou v praxi běžně nastat, třetí je nestabilní a v reálném obvodu neudržitelný, tedy nepozorovatelný. Bohužel program „s oblibou“ najde většinou tento pracovní bod.
Obecně mohou nastat tyto případy:

1. Program najde pracovní bod, který nás zajímá.
2. Program najde pracovní bod, o který nestojíme.
3. Program nenajde v daném počtu iterací žádné řešení.

Výsledek můžeme ovlivnit volbou prvního odhadu řešení, tj. volbou souřadnic počáteční, nulté iterace. Pokud bude odhad blízký pracovnímu bodu, který nás zajímá, je naděje, že k němu program dospěje.

.NODESET … stanovení počátečního odhadu řešení některých uzlových napětí a proudů induktory

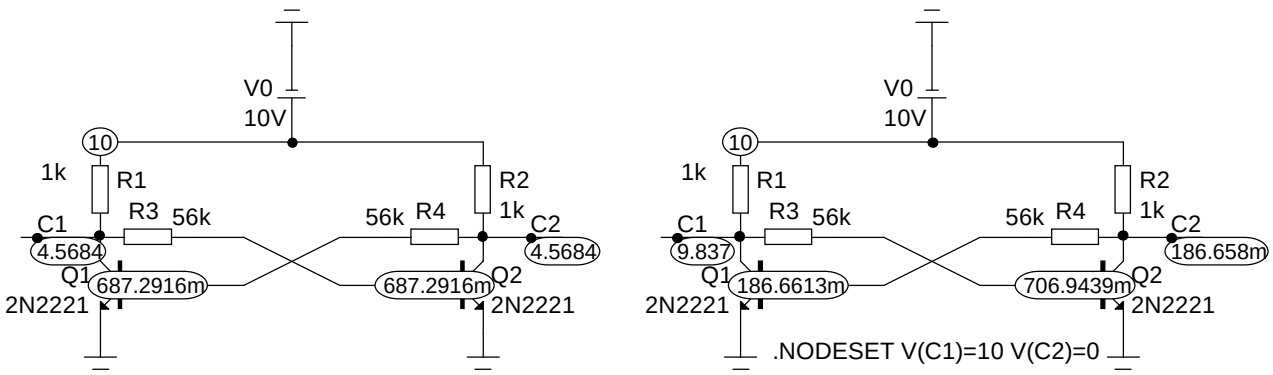
Formát:

.NODESET <V(<node> [,<node>])=<value> >*
.NODESET <I(<inductor>)=<value>>

<u>Příklad</u>:
.NODESET V(12)=5V V(in,out)=-5 I(Lx)=12mA

<u>Příklad analýzy bistabilního klopného obvodu</u> [2]:

a) Bez příkazu .NODESET, výpočet probíhá z nulových počátečních hodnot napětí. Program nalezne nestabilní pracovní bod.
b) S příkazem .NODESET, který představuje otevřený tranzistor Q1 a uzavřený Q2. Program nalezne stabilní pracovní bod odpovídající tomuto stavu.

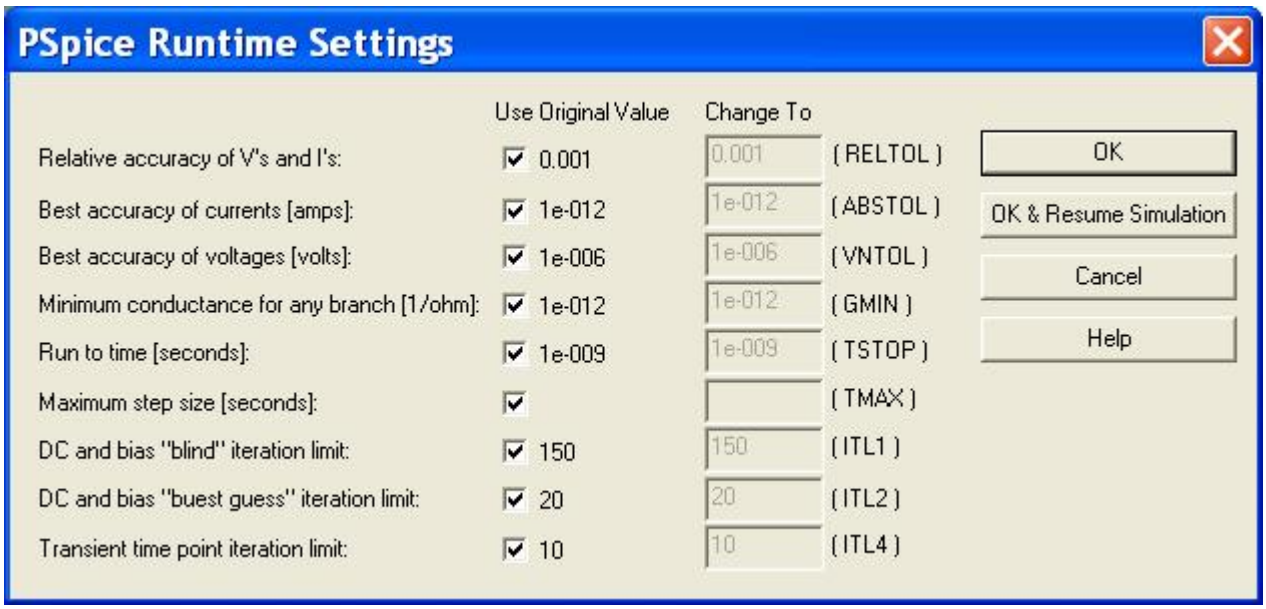


a) *b)*

<u>Další pravidla pro používání příkazu .NODESET:</u>

Je-li současně přítomen příkaz .IC, pak je .NODESET ignorován.
.NODESET se uplatňuje při výpočtu klidového pracovního bodu:
-pokud není uveden žádný příkaz pro analýzu, tedy pokud se počítá pouze ss pracovní bod,
- při analýze .TRANSIENT (pak se uplatní při výpočtu počátečního bodu časového průběhu),
- při analýze .AC
- při analýze .DC (pak se uplatní při výpočtu prvního bodu křivky).

Globální podmínky simulace - úvod



<u>Přesnost výpočtů</u> je řízena třemi kritérii:

- *ABSTOL* – Absolutní chyba ve výpočtu proudů.
- *VNTOL* – Absolutní chyba ve výpočtu napětí.
- *RELTOL* – Relativní chyba ve výpočtu napětí a proudů.

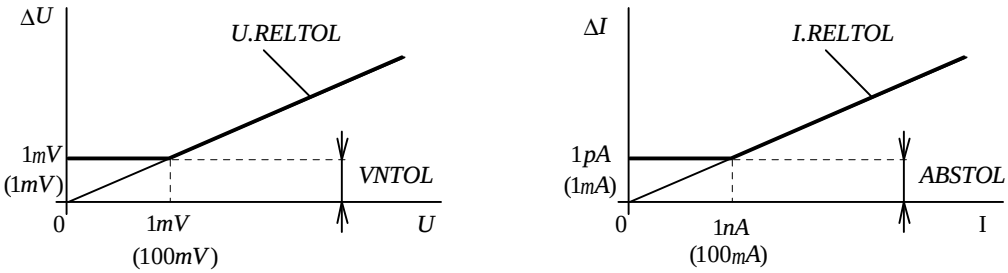
Při iteračním řešení rovnic se uvedená čísla využívají k rozhodování, zda řešení ukončit, následujícím způsobem:

1. V dané iteraci se modifikovanou metodou uzlových napětí vypočtou všechna uzlová napětí plus případné další pomocné veličiny, například proudy singulárními prvky. Z těchto hodnot se aplikací Ohmova zákona dopočítají proudy všemi větvemi.

2. Pro každé vypočtené napětí U a pro každý vypočtený proud I se určí čísla ΔU a ΔI podle těchto vzorců:

$$\Delta U = MAX(VNTOL, U.RELTOL)\,,\ \Delta I = MAX(ABSTOL, I.RELTOL)\,.$$

Grafické znázornění závislostí veličin ΔU a ΔI na napětí U a proudu I je na obr. Číselné hodnoty jsou uvedeny pro nastavení „*Standard Default*“, hodnoty v závorkách pro „*Power Default*“.



3. Porovnává se změna <u>každého</u> napětí U a <u>každého</u> proudu I oproti jejich velikostem v předchozím iteračním kroku. Výpočet se zastaví (tj. sada hodnot U a I je prohlášena za řešení), pokud jsou změny <u>všech</u> napětí menší než ΔU a současně změny <u>všech</u> proudů menší než ΔI.

Jestliže se programu nepodaří nalézt řešení, objeví se příslušné chybové hlášení, například „*Convergence problem in bias point calculation*“.

Pokud je při standardním nastavení testované napětí větší než 1 *m*V a testovaný proud větší než 1 *n*A, pak se iterace ukončí, když rozdíl po sobě jdoucích hodnot nepřekročí jednu tisícinu („*RELTOL*“) aktuálního napětí, resp. proudu. Při relativně nízkých hodnotách napětí nebo proudů by však tato chyba vycházela velmi malá (při nulových hodnotách dokonce nulová). Pokud by byla porovnatelná s numerickými chybami výpočtů, pak bychom se konvergence nemuseli dočkat vůbec. Z těchto důvodů jsou zavedeny absolutní chyby „*VNTOL*“ a „*ABSTOL*“, které při malých hodnotách U a I reprezentují povolené hranice odchylek ΔU a ΔI namísto parametru „*RELTOL*“.

Podmínka ukončení iteračního hledání řešení je i tak poměrně přísná. Proto se při řešení komplikovaných obvodů může stát, že v povoleném počtu iterací nedojde k jejímu naplnění. Maximální počet iterací je určen parametry *ITL*1, *ITL*2:

<u>*ITL*1 (Standard Default = 100, Power Default = 150):</u>

Určuje maximální počet iterací při počítání stejnosměrného pracovního bodu (inicializace časové analýzy, předstupeň kmitočtové analýzy).

<u>*ITL*2 (Default = 20):</u>

Určuje maximální počet iterací při počítání stejnosměrného pracovního bodu v každém kroku analýzy „*DC*“.

Z uvedeného plyne, že pokud nedojde k nalezení řešení v daném počtu iterací, je možné vyzkoušet buď <u>zvětšení parametrů</u> *RELTOL*, *ABSTOL* a *VNTOL* (zlepšení konvergence na úkor přesnosti výsledků), nebo <u>zvětšit povolený počet iterací</u> (pokusit se o konvergenci zvětšením počtu iterací, tj. na úkor rychlosti výpočtů).

Parametr *GMIN*: Jsou-li v modelech polovodičových diod použity nulové hodnoty paralelních vodivostí, pak parametr *GMIN* tyto nulové hodnoty automaticky nahradí. Zvětšování *GMIN* tedy může rovněž napomoci při problémech s konvergencí obvodů s polovodičovými strukturami.

Označování stejnosměrných napětí a proudů v PSpice

V(číslo uzlu)	napětí uzlu oproti referenčnímu uzlu
Příklad: V(18)	napětí mezi uzlem 18 a uzlem 0
V([název uzlu])	napětí uzlu, nazvaného jiným řetězcem než číslem, oproti referenčnímu uzlu
Příklad: V([vstup])	napětí mezi uzlem „vstup“ a referenčním uzlem
V(číslo uzlu1, číslo uzlu2)	Napětí mezi uzlem1 a uzlem2, stejné jako V(číslo uzlu1)-V(číslo uzlu2)
Příklad: V(3,11)	napětí mezi uzlem č. 3 a uzlem č. 11, tj. V(3)-V(11)
V(jméno dvojpólu)	napětí mezi svorkama 1 a 2 dvojpólu; svorka 1 je první svorka, uvedená v deklaraci dvojpólu
Příklad: V(Rx)	napětí na rezistoru Rx
V<jméno svorky>(jméno vícepólu)	napětí mezi svorkou vícepólu a referenčním uzlem
Příklad: Vb(Q1)	napětí mezi bází tranzistoru Q1 a referenčním uzlem
V<jméno svorky1><jméno svorky2> (jméno vícepólu)	napětí mezi svorkami 1 a 2 vícepólu
Příklad: Veb(Q1)	napětí mezi emitorem a bází tranzistoru Q1
I(jméno dvojpólu)	proud tekoucí vnitřkem dvojpólu od svorky 1 do svorky 2; svorka 1 je první svorka, uvedená v deklaraci dvojpólu
Příklad: I(D2)	proud diodou D2
I<jméno svorky>(jméno vícepólu)	proud vtékající do svorky vícepólu
Příklad: Ib(Qx)	proud tekoucí do báze tranzistoru Qx

Označování stejnosměrných napětí a proudů v PROBE

Pozn.: při volání .PROBE můžeme uvést napětí a proudy v konvencích PSpice. Automaticky bude provedena konverze do formátu PROBE. Při volání „prázdného“ příkazu .PROBE je třeba zadávat napětí a proudy podle konvencí PROBE.

Zadávání v PROBE = zadávání v PSpice s těmito výjimkami:

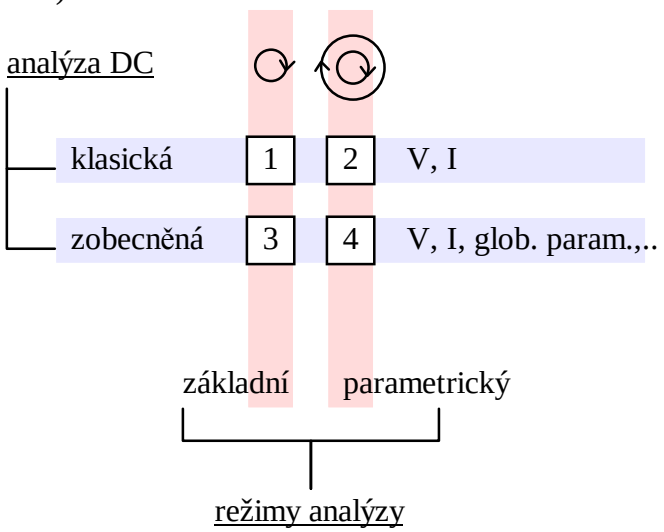
V(název uzlu)	napětí uzlu, nazvaného jiným řetězcem než číslem, oproti referenčnímu uzlu; oproti PSpice se nepoužívají závorky [ ]
Příklad: V(vstup)	napětí mezi uzlem „vstup“ a referenčním uzlem

Nejsou podporovány zápisy
V(jméno dvojpólu)
V<jméno svorky>(jméno vícepólu)
V<jméno svorky1><jméno svorky2> (jméno vícepólu)
(s výjimkou, že jsou volány s příkazem .PROBE, tj. že veličiny jsou vypočteny PSpicem a uloženy v souboru .dat)

Stejnosměrná analýza
= analýza stejnosměrných poměrů v obvodu

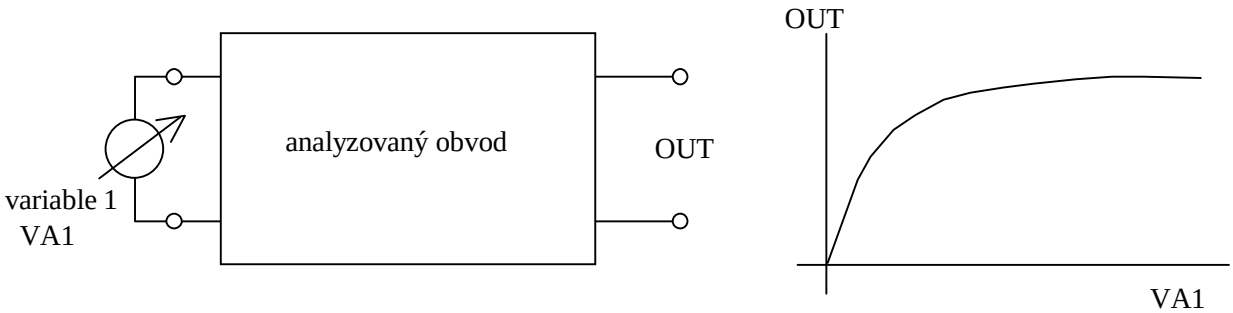
- jednorázový výpočet stejnosměrného pracovního bodu (žádný příkaz pro analýzu, příp. .OP)
- analýza DC (příkaz .DC)
- analýza „Dynamic DC“ – dynamická, interaktivní DC analýza (umožňuje MicroCap, nikoliv SPICE)

Analýza DC
= napodobování funkce přístroje pro snímání stejnosměrných charakteristik nelineárních obvodů (AV charakteristiky diod…).

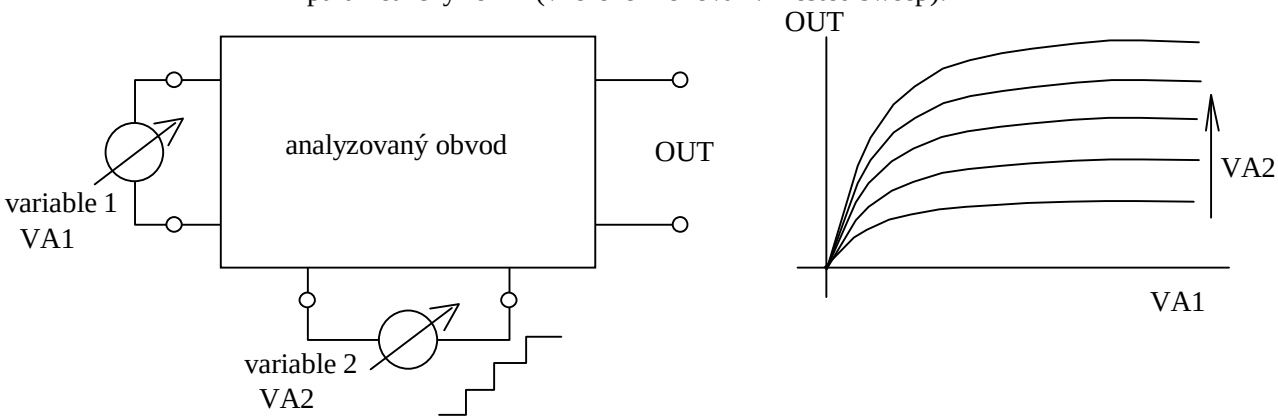


1: AV charakteristika diody, **2:** síť výstupních charakteristik tranzistoru, **3:** Ic=f(Rc), **4:** Ic=f(BF,TEMP).

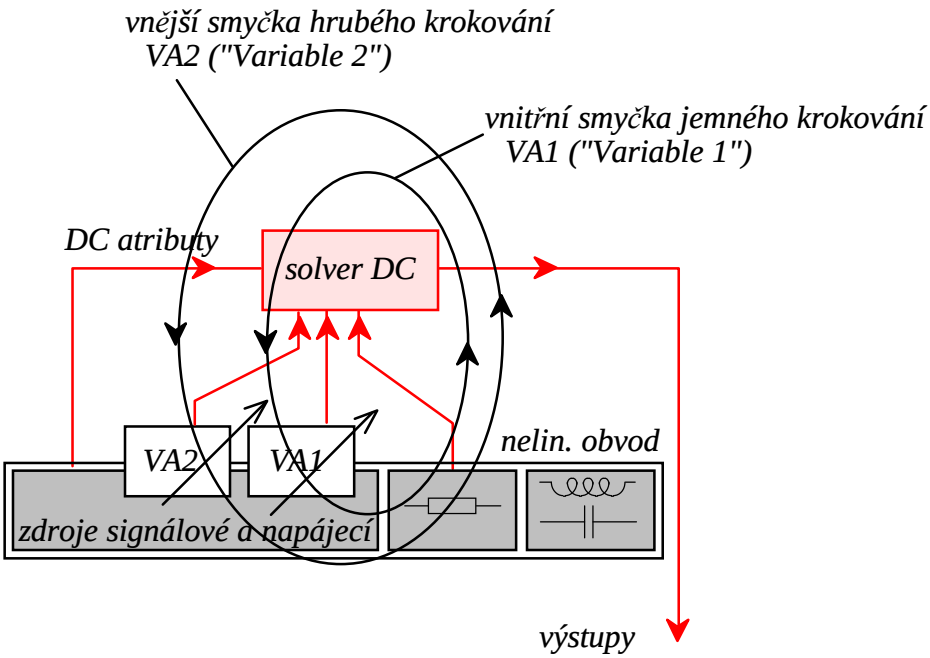
základní režim (jednoduché krokování / Single sweep):



parametrický režim (vnořené krokování/ Nested sweep):



Jak postupuje simulátor při analýze DC:



Co všechno se dá krokovat (povolené typy proměnných VA1 a VA2):

Při **klasické** stejnosměrné analýze:
-napětí a proudy nezávislých zdrojů v obvodu.

Při **zobecněné** stejnosměrné analýze:
-napětí a proudy nezávislých zdrojů v obvodu,
-teplota,
-libovolný parametr z modelů součástek v obvodu,
-libovolná globální proměnná.

Jak se chovají ostatní zdroje a nekrokované parametry:
- nezávislé zdroje jsou nahrazeny zdroji o parametrech DC
- řízené zdroje pracují se stejnosměrnými vzorci
- v modelech prvků se uplatňují pouze stejnosměrné parametry.

Syntaxe příkazu .DC:

Základní režim – jednoduché krokování:

<u>Lineární krokování</u>

.DC [LIN] <jméno VA1> <počáteční hodnota> <konečná hodnota> <krok>

Krok musí být větší než 0. Počáteční hodnota může být i větší než konečná hodnota.

Příklady:

.DC LIN Vbat 0V 15V 1V	; rozmítání napětí zdroje Vbat od 0V do 15V po 1V
.DC Vbat 0V 15V 1V	; stejné jako v předchozím případě
.DC Idc 100mA -100mA 1mA	; zmenšování proudu zdroje Idc od 100mA do -100mA s krokem 1mA
.DC NPN Qtranzistor(BF) 100 500 10	; rozmítání parametru BF všech tranzistorů, které mají model Qtranzistor, od 100 do 500 s krokem 10
.DC PARAM Rx 1k 10k 1k	; rozmítání globálního parametru Rx od 1k do 10k po 1k

Logaritmické krokování

.DC DEC <jméno VA1> <počáteční hodnota> <konečná hodnota> <počet kroků na dekádu>

nebo

.DC OCT <jméno VA1> <počáteční hodnota> <konečná hodnota> <počet kroků na oktávu>

Konečná hodnota musí být větší než počáteční. Počet kroků musí být celé kladné číslo.

Příklady:

.DC DEC Vx 1mV 1V 1	; rozmítání napětí zdroje Vx v hodnotách 1mV, 10mV, 100mV, 1V
.DC OCT PARAM Cx 1n 16n 1	; rozmítání globálního parametru Cx v hodnotách 1nF, 2nF, 4nF, 8nF a 16nF

Krokování „LIST“

.DC <jméno VA1> LIST <výčet hodnot>

Příklady:

.DC Vbat LIST 5V 15V	; provede se výpočet pro dvě napětí 5V a 15V
.DC PARAM beta LIST 100	; provede se výpočet pro jedinou hodnotu globálního parametru beta = 100
.DC PARAM beta 100 100 1	; stejný efekt jako v předchozím případě (jde o lineární krokování, kde konečná hodnota je rovna počáteční, na velikosti kroku nezáleží)

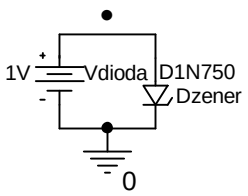
Parametrický režim – vnořené krokování:

Syntaxe navazuje na jednoduché krokování, následuje jméno druhé krokované proměnné VA2 a specifikace krokování, která je stejná jako u jednoduchého krokování.

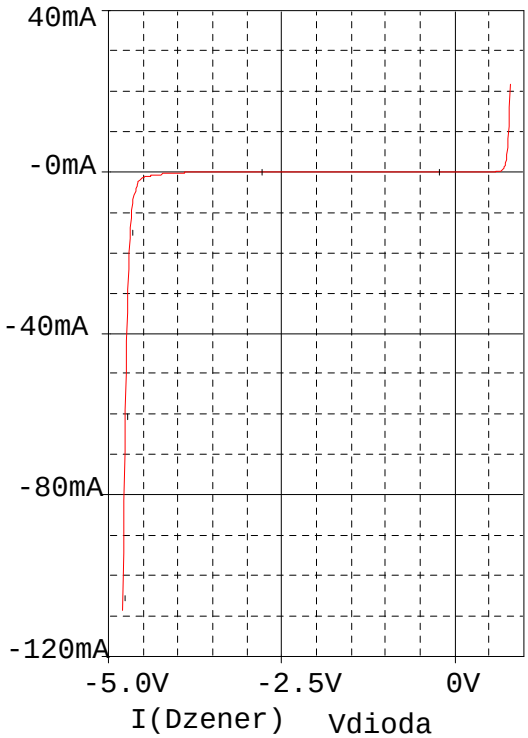
Příklad:

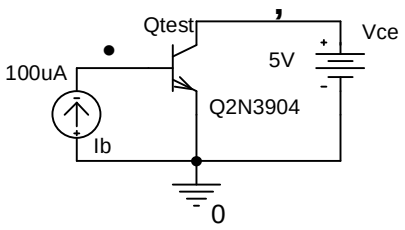
.DC Vce 0 15 0.1 Ib 0 100u 10u	; ve vnější smyčce se nastavuje proud Ib lineárně od nuly do 100uA po 10uA; pro každou hodnotu Ib proběhnou výpočty pro napětí Vbe od nuly do 15V po 0.1V.

Ukázky simulací:

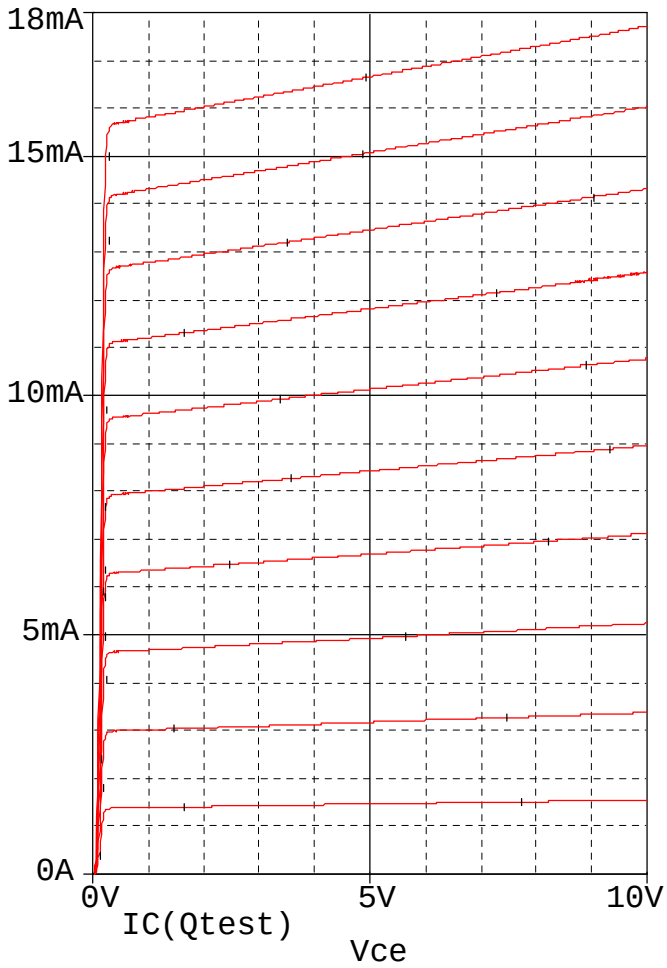


Zenerka
*
Dzener 1 0 D1N750
Vdioda 1 0 1V
.DC Vdioda -4.8 0.8 0.01
.probe
.lib
.END





AV char
*
Qtest 2 1 0 Q2N3904
Vce 2 0 5V
Ib 0 1 100uA
.DC Vce 0V 10V 0.01V Ib 0A 100uA 10uA
.probe
.lib
*
.END



Poznámka – které parametry modelů nelze krokovat:

L, W MOSFETu
Teplotní parametry, např. TC1 a TC2 rezistorů.

Příkaz .PRINT – vypisování výsledků analýzy do výstupního souboru

Tento příkaz funguje pouze ve spolupráci s příkazy pro základní analýzy, např. .DC.

.PRINT <označení analýzy> <výčet napětí a proudů>

Příklady:

.DC Vbat 15 15 1	; DC analýza pro jedinou hodnotu rozmítaného zdroje Vbat
.PRINT V(R1) I(Rx) V([uzel])	; do vstupního souboru se zapíše 1 řádek s hodnotami napětí na R1, proudu tekoucího přes Rx, a napětí mezi uzlem o názvu uzel a referenčním uzlem
.DC Vbat 5 15 1	; DC analýza pro 11 hodnot Vbat od 5V do 15V po 1V
.PRINT V(R1) I(Rx) V([uzel])	; do vstupního souboru se zapíše 11 řádků s hodnotami napětí na R1, proudu tekoucího přes Rx, a napětí mezi uzlem o názvu uzel a referenčním uzlem; každému řádku budou odpovídat hodnoty, vypočtené pro jednu z krokovaných hodnot Vbat

Počet míst vypisovaných číselných údajů je určován položkou NUMDGT v globálních podmínkách simulace. NUMDGT = počet míst mantisy za desetinnou tečkou plus 1. Standardně je NUMDGT=4. Chceme-li např. výpis na 5 desetinných míst, umístíme do vstupního souboru příkaz

.OPTIONS NUMDGT=6

Literatura

[1] BIOLEK,Z. Úvod do SPICE pomocí programu MicroCap. Učební texty SPŠE v Rožnově p.R. SENSIT HOLDING s.r.o., 2004, 34 s.

[2] BIOLEK, D. Řešíme elektronické obvody aneb kniha o jejich analýze. BEN technická literatura, 2004, 520 s.

[3] LÁNÍČEK, R. Simulační programy pro elektroniku. BEN technická literatura, 2000, 113 s.