

SPICE (Simulation Program with Integrated Circuit Emphasis)

カリフォルニア大学バークレー校で1973年に開発。電子回路のシミュレーション・ソフト。

テキストファイルで回路を記述。

ex1.cir

OHM'S LAW

* The above line is the title.

* Current source

i 0 a 2

* Resistors

r1 a 0 10

.control

op

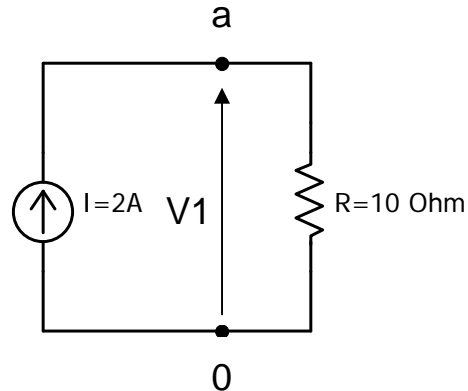
* Print messages using echo.

echo V1 is:

print v(a)

.endc

.end



実行結果

```
SpiceOpus Command Window
File Edit Control Window Help

Welcome to Program: SpiceOpus (c), version: 2.25 Light $Revision: 32 $
Date built: Dec 5 2006

Based on:
  SPICE 3f4 (patched to 3f5) by
    Electronics Research Laboratory
    College of Engineering
    University of California, Berkeley
  XSPICE by
    Georgia Tech Research Institute

University of Ljubljana, Slovenia
Faculty of Electrical Engineering
Group For Computer Aided Circuit Design
http://fides.fe.uni-lj.si/spice/

SpiceOpus (c) 1 -> cd c:/takenaka/program
current directory: C:/takenaka/program
SpiceOpus (c) 2 -> cd SPICE/ex1
current directory: C:/takenaka/program/SPICE/ex1
SpiceOpus (c) 3 -> source ex1.cir
V1 is:
v(a) = 2.000000e+001 ← V1=20Vと求まっている
SpiceOpus (c) 4 -> |
```

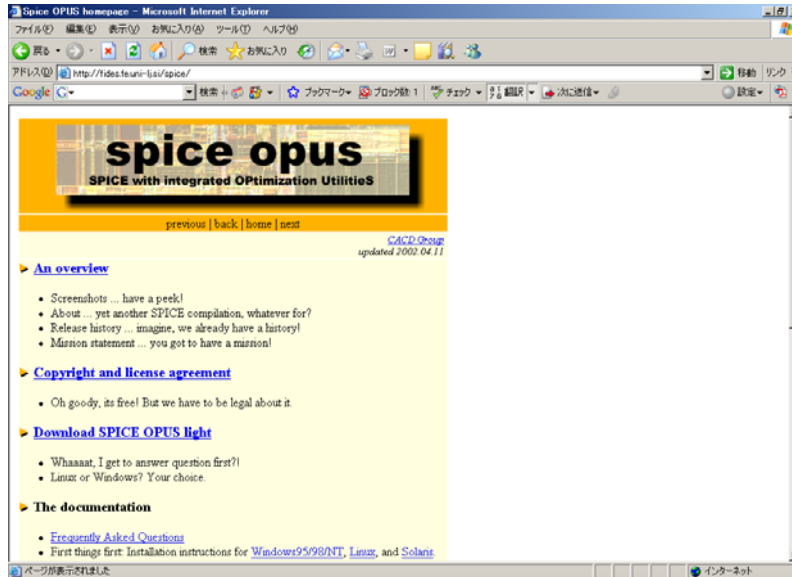
- 交流信号、過渡応答、トランジスタ回路なども解析可能
- 商用版: ex. PSPICE, HSPICE, etc.
- フリー版: ex. SPICE OPUS (<http://fides.fe.uni-lj.si/spice/>)

Fig. 1

SPICE OPUS インストール

1. HPからダウンロード

<http://fides.fe.uni-lj.si/spice/>



2. Windows版のインストール

デフォルトではc:\¥SpiceOpusにインストールされる。

この演習では、このフォルダにインストールされたことを前提に話を進める。

* Tutorialなどのドキュメントもある。

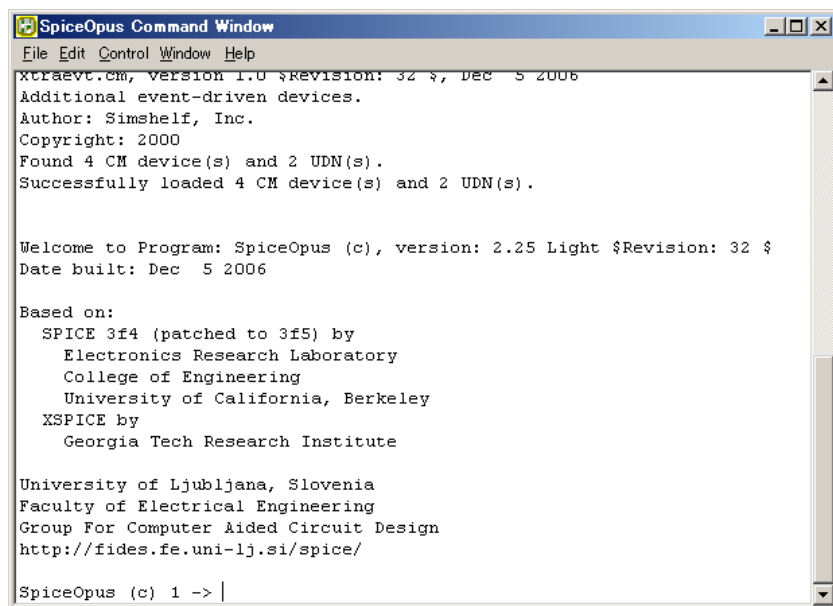
Fig. 2

SPICE OPUS操作概略

1. SPICE OPUS起動

Windowsメニューに追加されているはず。または
c:¥SpiceOpus¥bin¥spiceopus.exeを実行。

下图のターミナル画面がでる。



```
SpiceOpus Command Window
File Edit Control Window Help
xttraevt.cm, version 1.0 $Revision: 32 $, Dec 5 2006
Additional event-driven devices.
Author: Simshelf, Inc.
Copyright: 2000
Found 4 CM device(s) and 2 UDN(s).
Successfully loaded 4 CM device(s) and 2 UDN(s).

Welcome to Program: SpiceOpus (c), version: 2.25 Light $Revision: 32 $
Date built: Dec 5 2006

Based on:
  SPICE 3f4 (patched to 3f5) by
    Electronics Research Laboratory
    College of Engineering
    University of California, Berkeley
  XSPICE by
    Georgia Tech Research Institute

University of Ljubljana, Slovenia
Faculty of Electrical Engineering
Group For Computer Aided Circuit Design
http://fides.fe.uni-lj.si/spice/

SpiceOpus (c) 1 -> |
```

2. シミュレーション実行

コマンドラインで、Circuit fileを読み込む。

source filename.cir

*Circuit file

回路構成や計算内容を記述したテキストファイル。

拡張子は.cir。

デフォルトでは、c:¥SpiceOpusにあるファイルが読み込まれる。

異なるフォルダ(ex. c:¥takenaka)にあるファイルは、

cd c:/takenaka

とカレントディレクトリを移動してから、実行

cd c:¥takenakaではないことに注意

UNIXとの整合性の問題

Fig. 3

Circuit file書式概説

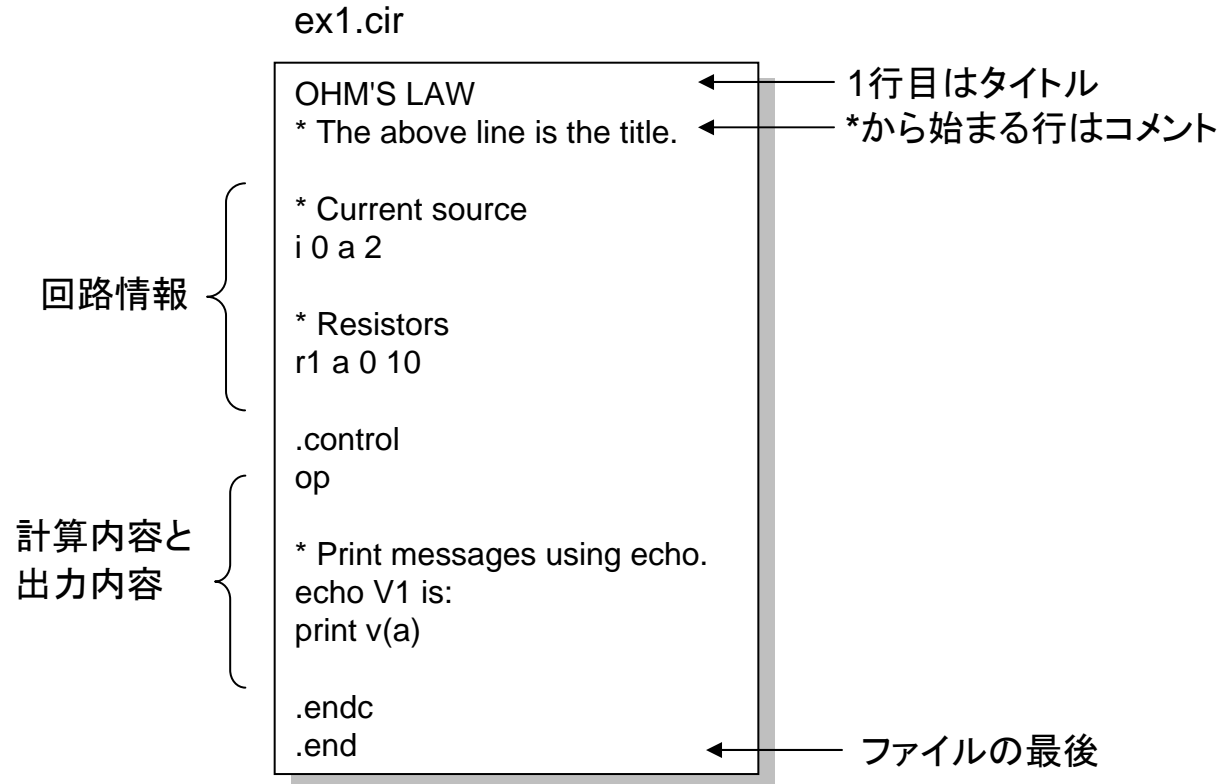


Fig. 4

回路情報の記述の仕方

ex1.cir

```
OHM'S LAW
* The above line is the title.

* Current source
i 0 a 2

* Resistors
r1 a 0 10

.control
op

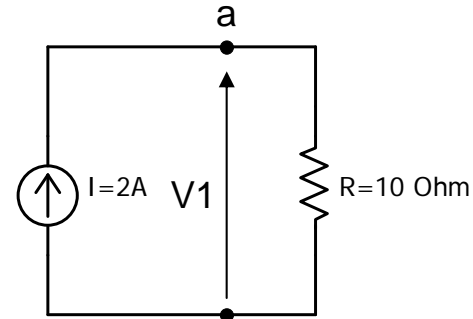
* Print messages using echo.
echo V1 is:
print v(a)

.endc
.end
```

回路情報

1. 回路のノードに名前をつける。

* 0はグラウンドと決まっている。



デフォルトでグラウンド

2. ノードの接続情報を入力

電流源:

i 0 a 2

2Aの意味

↑

ノード0-a間に接続

電流源の意味

抵抗:

r1 a 0 10

10Ωの意味

rは抵抗の意味

ノードa-0間に接続

1は抵抗の名称

Fig. 5

制御領域の記述の仕方

ex1.cir

```
OHM'S LAW
* The above line is the title.

* Current source
i 0 a 2

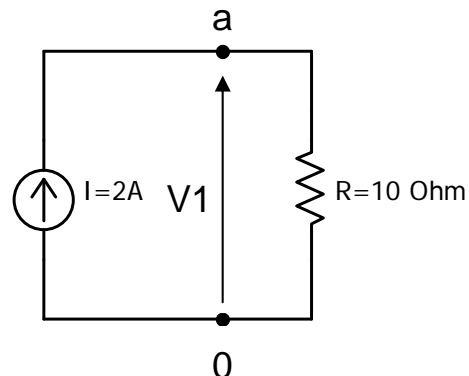
* Resistors
r1 a 0 10

.control
op

* Print messages using echo.
echo V1 is:
print v(a)

.endc
.end
```

制御領域



← 計算実行(直流の計算)

← echo以下の文字が画面に出力される
← ノードaの電圧を画面に出力

実行結果

```
SpiceOpus (c) 3 -> source ex1.cir
V1 is:
v(a) = 2.000000e+001
```

V1=20Vであることが分かる

Fig. 6

The screenshot shows the SpiceOpus Command Window with the following text:

```
SpiceOpus Command Window
File Edit Control Window Help

Welcome to Program: SpiceOpus (c), version: 2.25 Light $Revision: 32 $
Date built: Dec 5 2006

Based on:
SPICE 3f4 (patched to 3f5) by
  Electronics Research Laboratory
  College of Engineering
  University of California, Berkeley
XSPICE by
  Georgia Tech Research Institute

University of Ljubljana, Slovenia
Faculty of Electrical Engineering
Group For Computer Aided Circuit Design
http://fides.fe.uni-lj.si/spice/

SpiceOpus (c) 1 -> cd c:/takenaka/program
current directory: C:/takenaka/program
SpiceOpus (c) 2 -> cd SPICE/ex1
current directory: C:/takenaka/program/SPICE/ex1
SpiceOpus (c) 3 -> source ex1.cir
V1 is:
(a) = 2.000000e+001
SpiceOpus (c) 4 ->
```

電流値の求め方

ex2.cir

Current Measurement

* The above line is the title.

* Voltage source

v1 a 0 10 ← 10Vの電圧源

* Dummy voltage source

vab a b 0 ← ダミー電圧源
電圧=0とする

* Resistor

r1 b 0 2

.control

op

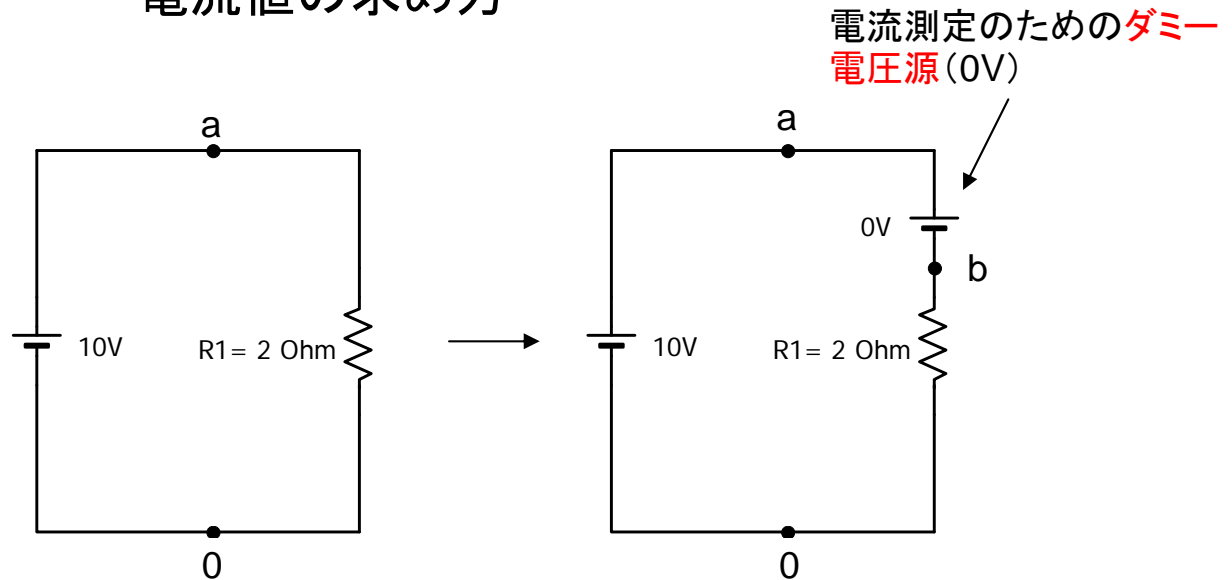
* Print messages using echo.

echo I is:

print i(vab) ← ダミー電圧源の
電流値を表示

.endc

.end



実行結果

```
SpiceOpus (c) 2 -> source ex2.cir
I is:
i(vab) = -5.00000e+000
```

電流値が5Aであることが分かる。

Fig. 7

補助単位

値	SPICEでの記述
10^{12}	T
10^9	G
10^6	Meg
10^3	K
10^{-3}	m
10^{-6}	u (or M)
10^{-9}	n
10^{-12}	p
10^{-15}	f

Fig. 8

ex.3:キルヒホッフの電流則、電圧則

ex.3: KCL, KVL

* The above line is the title.

* Voltage source

v1 d 0 2

* Current source

i 0 c 3

* Dummy voltage source

vbe b e 0

vbf b f 0

vg0 g 0 0

vh0 h 0 0

* Resistor

r1 a d 1

r2 a e 1

r3 f c 1

r4 b g 1

r5 c h 1

.control

op

* Print messages using echo.

echo Va, Vb, Vc:

print v(a) v(b) v(c)

echo KCL: i1 + i2 + i3:

print i(vbe) + i(vbf) + i(vg0)

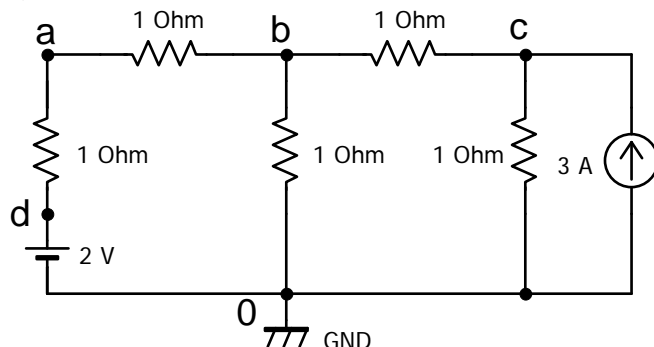
echo KVL: loop 1

* $v(a,d) = v(a) - v(d)$

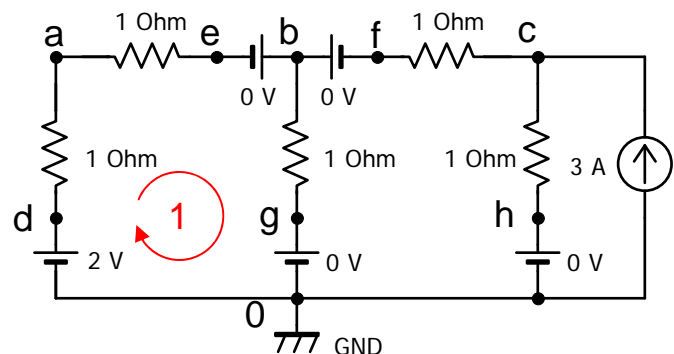
print v(d) + v(a,d) + v(e,a) + v(b,e) +
v(g,b) - v(g)

.endc

.end



電流測定のための
ダミー電圧源 (0V)



実行結果

```
SpiceOpus (c) 17 -> source ex3.cir
Va Vb Vc:
v(a) = 1.625000e+000
v(b) = 1.250000e+000
v(c) = 2.125000e+000
KCL: i1 + i2 + i3:
i(vbe) + i(vbf) + i(vg0) = 0.000000e+000
KVL: loop 1
v(d) + v(a,d) + v(e,a) + v(b,e) + v(g,b) - v(g) = 0.000000e+000
```

ノードbにおけるKCL

ループ1のKVL

Fig. 9

ex. 4 最大電力の取り出し

ex4.cir

ex.4: Maxium Power Transfer Theorem

* Voltage source

vin a 0 dc 50

* Dummy voltage source

vx c 0 dc 0

* Resistor

rin a b 100

rv b c 1

.control

dc rv 1 400 1

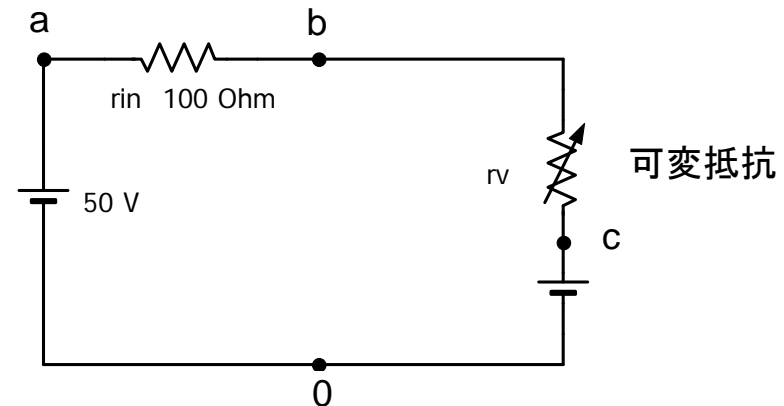
plot i(vx)*v(b,c)

.endc

.end

rvの抵抗を1→400Ω(1Ωステップ)
で変化させて、直流解析する

← i*vをプロット



実行結果

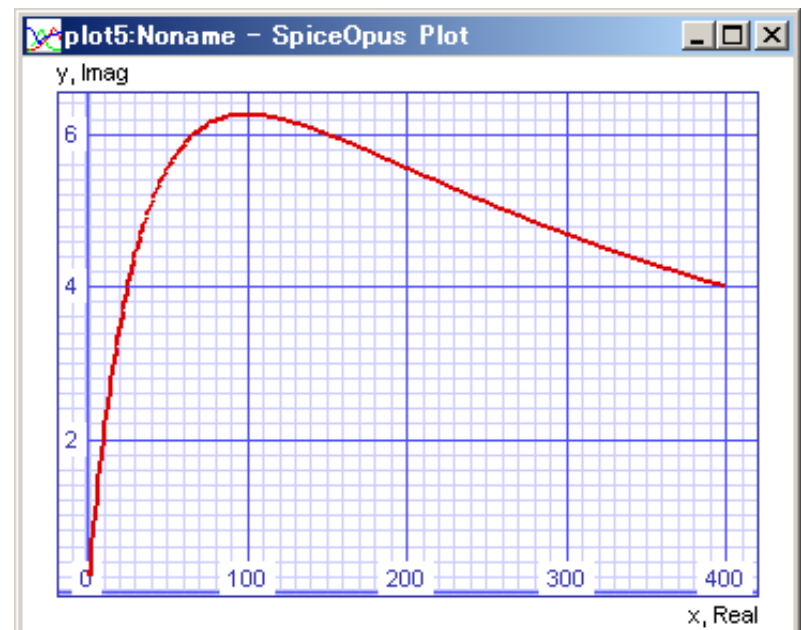


Fig. 10

交流回路の解析

ex5.cir

ex.5: AC circuit

* AC Voltage source

v1 a 0 dc 0 ac 100

* Resistor

r1 a 0 100

交流(実効値)
直流成分=0V

.control

set units=degree

destroy all

ac dec 1 50 50

echo Va:

print v(a)

echo Amplitude and phase of Va:

print mag(v(a)) ph(v(a))

echo I1, I1*r1:

print i(v1) i(v1) * @r1

r1の値の意味

.endc

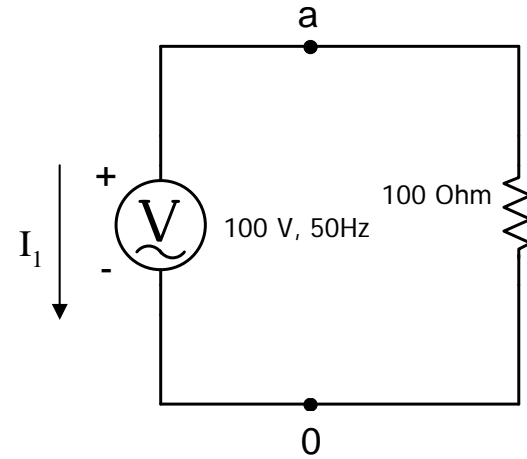
.end

位相表示をdegreeに変更

(デフォルトはradian)

以前の計算結果を消去

周波数50Hzの交流計算を実行



実行結果

```
SpiceOpus (c) 23 -> source ex4.cir
Va:
v(a) = 1.000000e+002,0.000000e+000
Amplitude and phase of Va:
mag(v(a)) = 1.000000e+002
ph(v(a)) = 0.000000e+000
I1 I1*r1:
i(v1) = -1.000000e+000,0.000000e+000
i(v1) * @r1 = -1.000000e+002,0.000000e+000
```

Fig. 11

ex6.cir

ex.6: LCR resonator

* AC Voltage source
v1 a 0 dc 0 ac 100

* Inductance
l1 a b 10m ← コイル

* Resistor
r1 b c 100

* Capacitor
c1 c 0 0.1u ← キャパシタ

.control

set units=degree

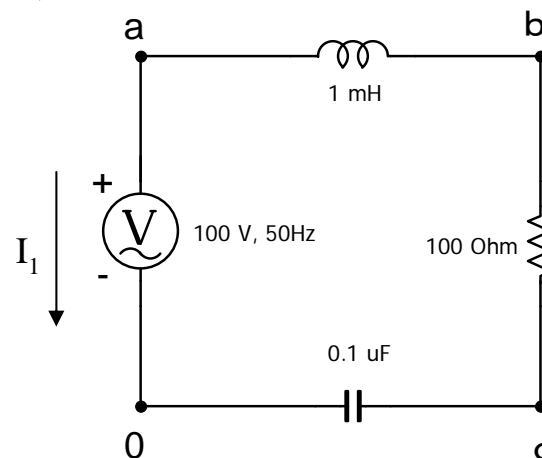
destroy all

ac dec 100 25 1Meg ←

* plot: i1 vs freq.
plot mag(i(v1))
+ xlabel 'Freq. [Hz]' ylabel 'I1 [Red]'
+ title 'I1 vs freq.'

.endc
.end

ex.6: 直列共振回路



実行結果

25Hz→1MHzの特性を計算
(100点/1桁)

グラフを軸名、
タイトル

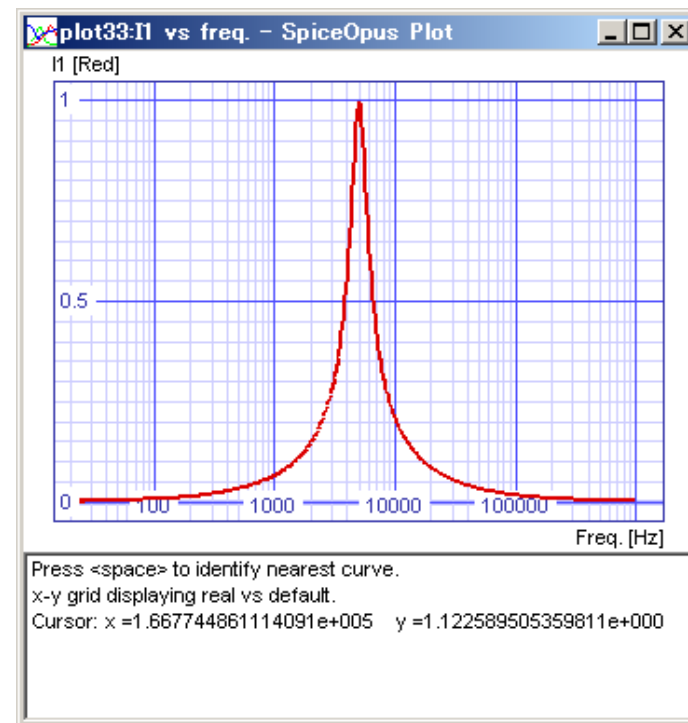


Fig. 12

ex7.cir

ex.7: RC low pass filter

* AC Voltage source

v1 a 0 dc 0 ac 1

* Resistor

r1 a b 10

* Capacitor

c1 b 0 1u

.control

set units=degree

destroy all

alter r1=10

ac dec 100 25 10Meg

計算1

R=10Ω

alter r1=100

ac dec 100 25 10Meg

計算2

R=100Ω

alter r1=1000

ac dec 100 25 10Meg

計算3

R=1000Ω

* plot: i1 vs freq.

plot mag(ac1.v(b)) mag(ac2.v(b))

+ mag(ac3.v(b))

+ xlabel 'Frequency[Hz]'

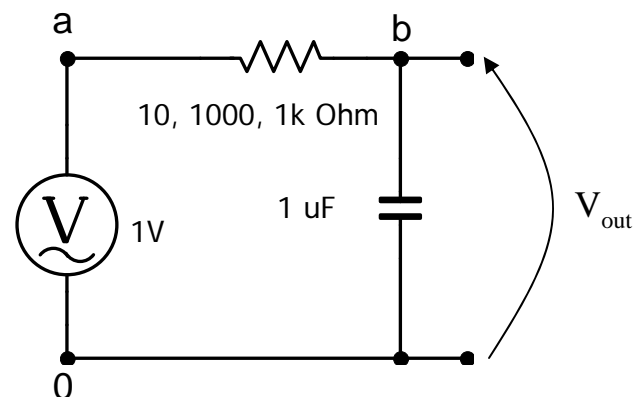
+ ylabel 'Red: 10 Ohm, Green: 100 Ohm,
Blue: 1k Ohm'

+ title 'Low pass filter'

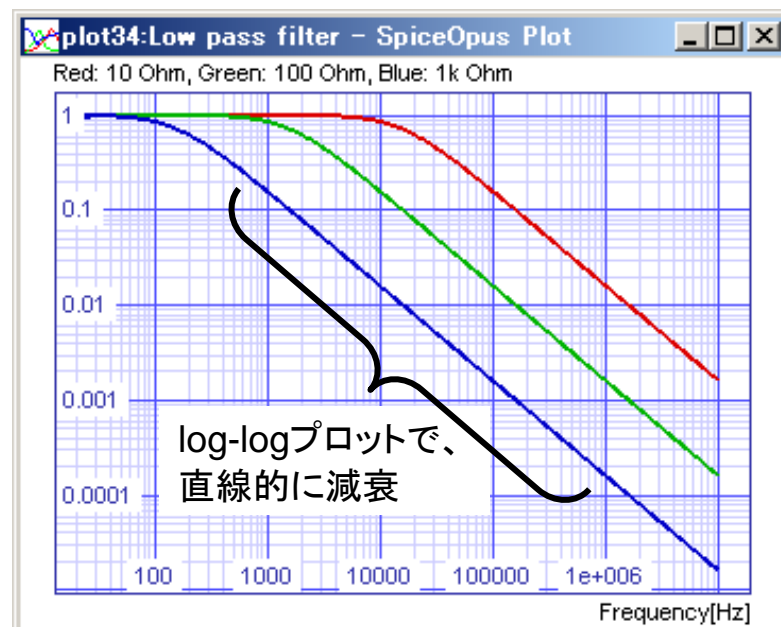
.endc

.end

ex. 7 ローパスフィルタ



実行結果



計算1,2,3の
プロット

Fig. 13

時間解析

ex8.cir

ex.8: Time domain analysis

* AC Voltage source

vin a 0 dc 0 ac 1mv 0 sin(0 1mv 10kHz)

* Resistor

r1 a 0 100

.control

set units=degree

destroy all

tran 1us 500u ← 1 μ sステップで、500 μ sまで時間解析

plot v(a) vs (time*1000)

.endc

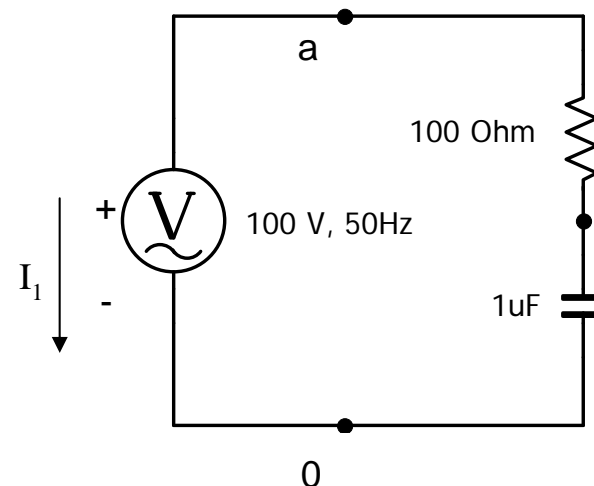
.end

ac解析用

直流成分=0V

時間解析用

sin波:dc=0, 振幅1mV
fr=10kHz



実行結果

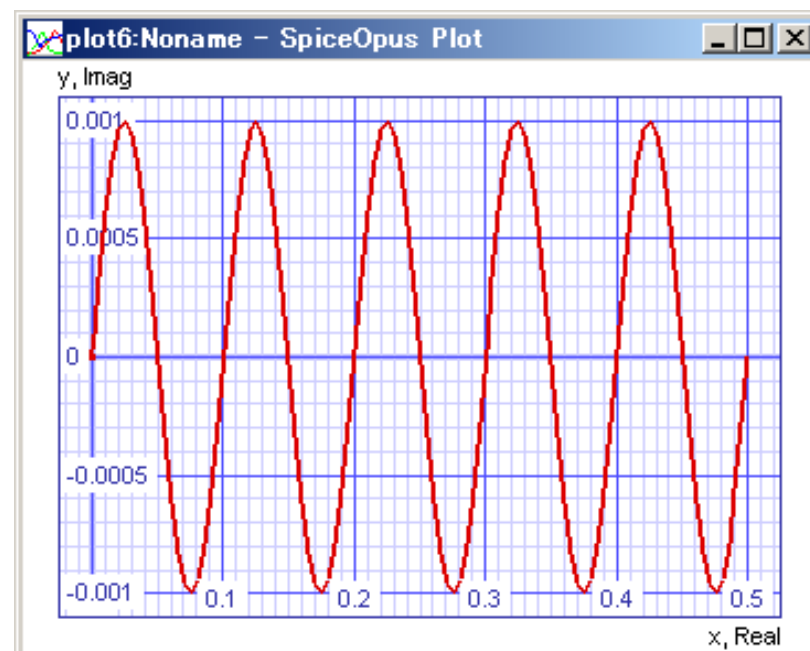


Fig. 14

対称三相交流

ex9.cir

ex.9: Three phase AC

* AC Voltage source

v1 u 0 dc 0 ac 1mv 0 sin(0 200 50Hz 0 0 0)

v2 v 0 dc 0 ac 1mv 0 sin(0 200 50Hz 0 0 120)

v3 w 0 dc 0 ac 1mv 0 sin(0 200 50Hz 0 0 240)

位相(度)

* Resistor

r1 u 0 100

r2 v 0 100

r3 w 0 100

.control

set units=degree

destroy all

tran 0.1ms 60ms

plot v(u) v(v) v(w)

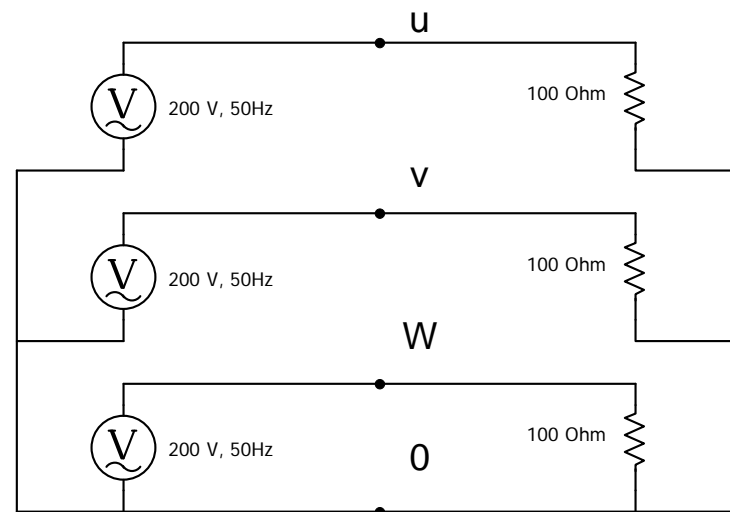
plot -v(u)*i(v1)-v(v)*i(v2)-v(w)*i(v3)

+ ylimit 0 700 ← 縦軸のスケール設定

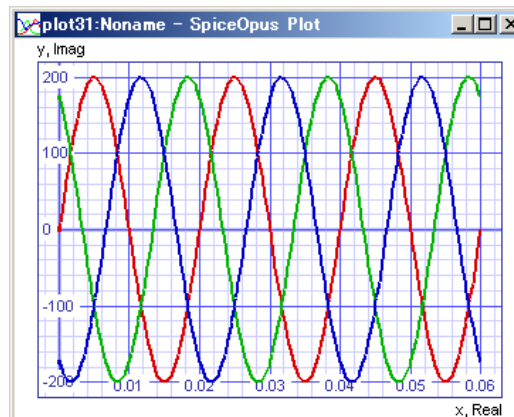
.endc

.end

三相電源



電源



合計電力

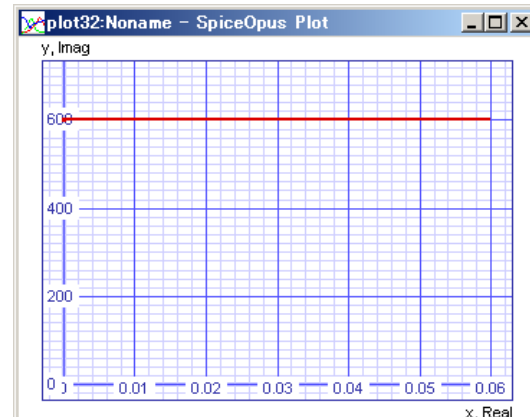


Fig. 15

方形波入力

ex10.cir

ex.10: Rectangular input

* AC Voltage source

v1 a 0 pulse(-1V 1V 0ms 0ms 0ms 5ms 10ms)

パルス幅5ms, 周期10ms

* Resistor

r1 a b 1k

* Capacitor

c1 b 0 1u

.control

set units=degree

destroy all

tran 0.1ms 30ms

plot v(a) v(b)

.endc

.end

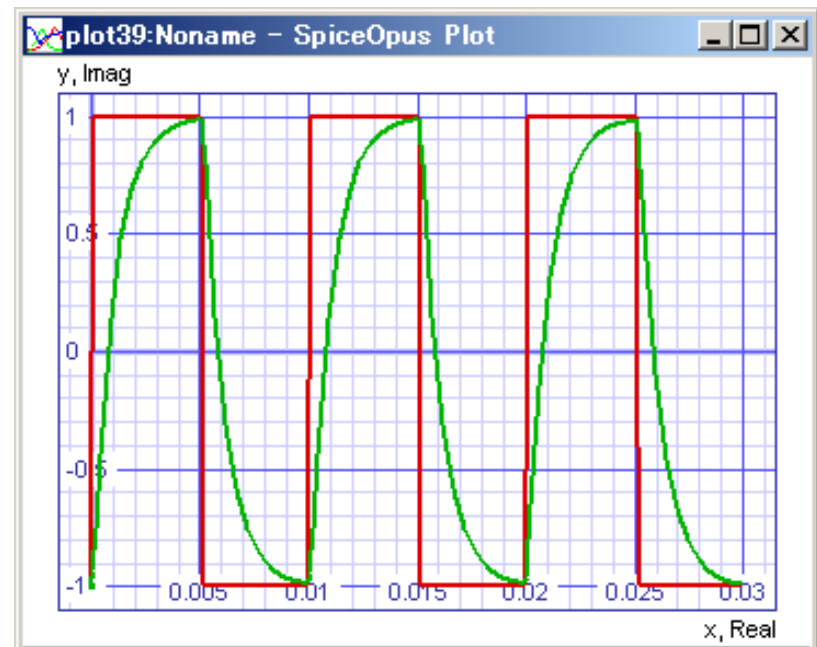
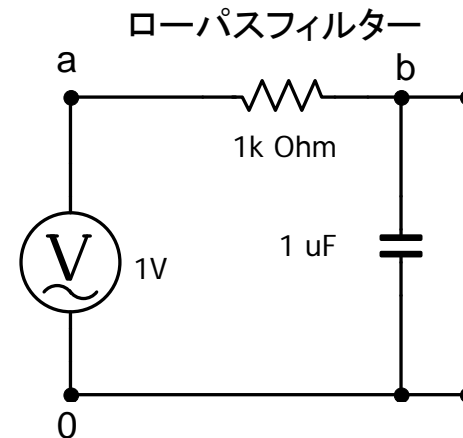


Fig. 16

三角波

ex11.cir

ex.11: Triangular input

* AC Voltage source

v1 a 0 pulse(-1V 1V 0ms 5ms 5ms 0.001ms 10ms)

* Resistor

r1 a b 2k

* Capacitor

c1 b 0 1u

.control

set units=degree

destroy all

tran 0.1ms 100ms

plot v(a) v(b) xlimit 50m 100m

.endc

.end

パルスの立ち上がり、立下り

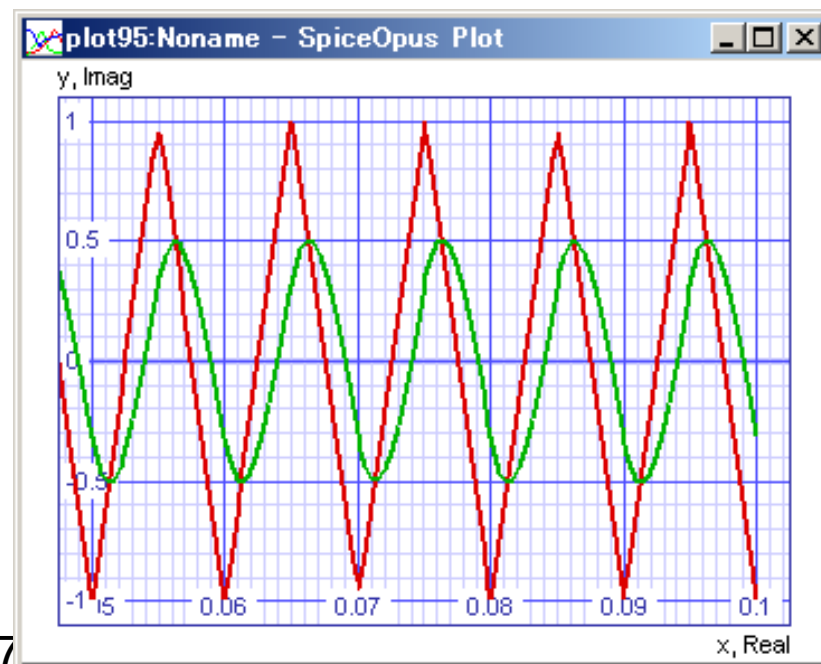
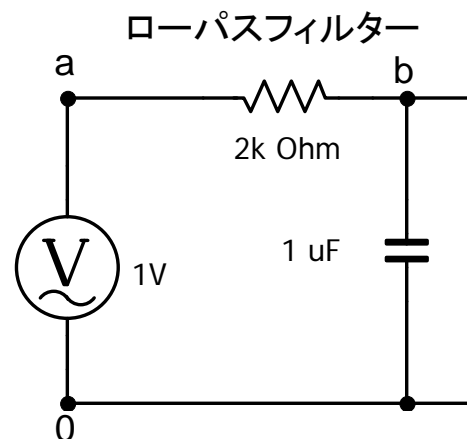


Fig. 17

ex12.cir

```
ex.12: Fourier analysis

* AC Voltage source
v1 a 0 pulse(-100V 100V 0ms 0ms 0ms 10ms 20ms)

* Resistor
r1 a 0 100

.control
set units=degree
destroy all

tran 0.001ms 100ms
plot v(a)

50Hzの高調波を解析
0-1000Hz(ステップ10Hz)のスペクトルを解析
解析時間100msの逆数より
大きい必要あり

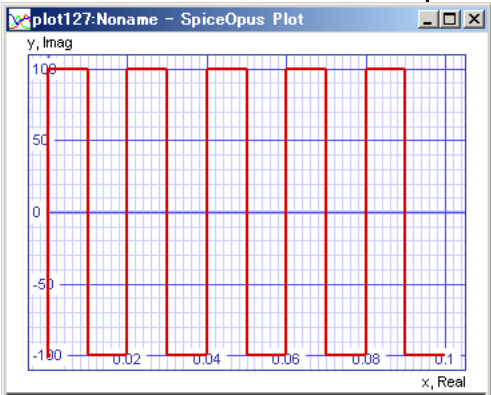
fourier 50Hz v(a)
spec 0 1000 10 v(a)
plot db(norm(mag(v(a))))

v(a)のスペクトルをプロット
norm()は最大値で割って規格化する
db()は20log10を計算する

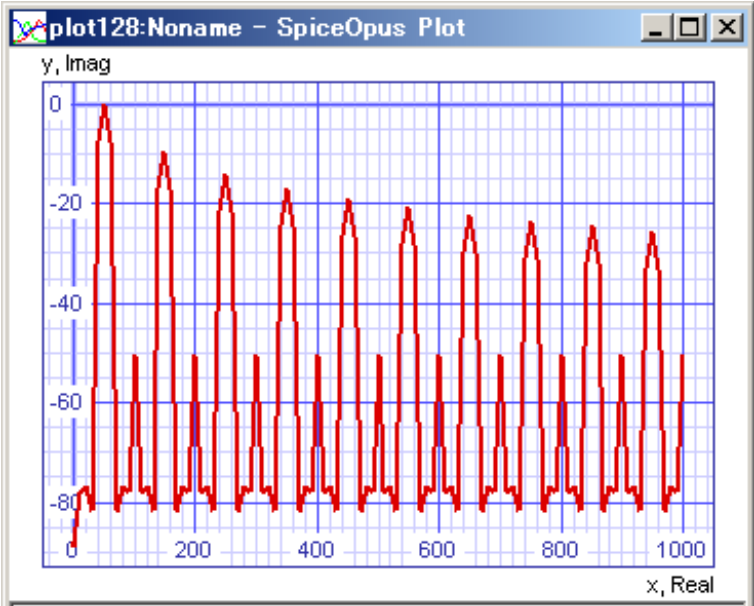
.endc
.end
```

フーリエ解析

50 Hz 方形波



フーリエ解析結果



SpiceOpus (c) 103 -> source ex12.cir
Fourier analysis for v(a):
No. Harmonics: 10, THD: 42.9161 %, Gridsize: 200, Interpolation Degree: 1

Harmonic	Frequency	Magnitude	Phase	Norm. Mag	Norm. Phase
0	0	0	0	0	0
1	50	127.329	-90.9	1	0
2	100	8.67605e-015	-18.194	6.81387e-017	72.7062
3	150	42.457	-92.7	0.333443	-1.8
4	200	3.02945e-014	141.141	2.37923e-016	232.041
5	250	25.491	-94.5	0.200198	-3.6
6	300	1.32856e-014	-64.668	1.04341e-016	26.2316
7	350	18.2258	-96.3	0.14314	-5.4
8	400	2.00124e-014	-70.504	1.5717e-016	20.3965
9	450	14.1943	-98.1	0.111478	-7.2

Fig. 18

過渡応答

ex13.cir

ex.13: RCL circuit

* AC Voltage source

v1 1 0 pulse(0V 1V 0ms 0ms 0ms 1000ms 2000ms)

* Resistor

r1 1 2 10

* Capacitor

c1 2 3 100u

* Inductor

l1 3 0 1

.control

set units=degree

destroy all

tran 1ms 4000ms 1000ms

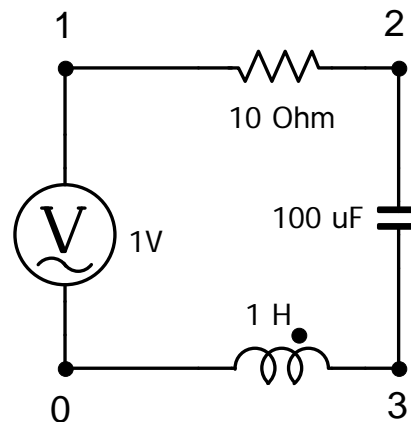
plot v(1)

plot -i(v1)

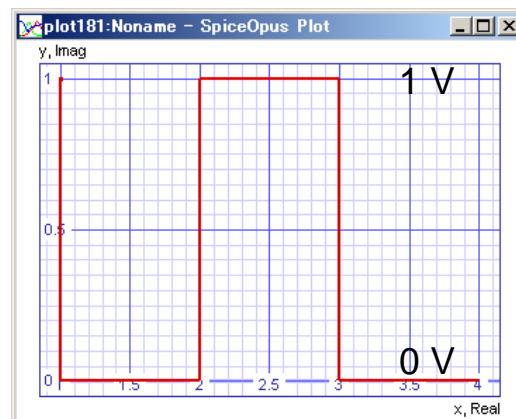
.endc

.end

1000msまでは結果を保存しない。



入力電圧



電流

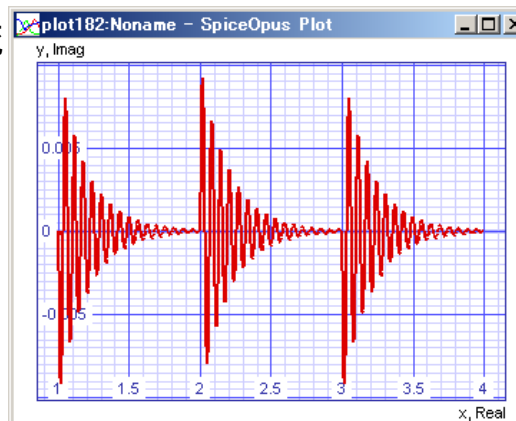


Fig. 19

デバイスモデルの利用

R, C, Lだけでなく、半導体デバイス(ダイオード、トランジスタ)なども計算可能

ex14.cir

ex.14: Diode

* AC Voltage source

v1 1 0 sin(0V 100V 50Hz)

* Resistor

r1 1 2 10

* Diode

d1 2 0 1N4007

モデル名

```
.MODEL 1N4007 D(IS = 3.872E-09 RS = 1.66E-02  
+ N = 1.776 XTI = 3.0 EG = 1.110 CJO = 1.519E-11  
+ M = 0.3554 VJ = 0.5928 FC = 0.5 BV = 1000.0 IBV = 1.0E-03)
```

ダイオードの各種パラメータ

(各社が製品情報として提供)

.control

set units=degree

destroy all

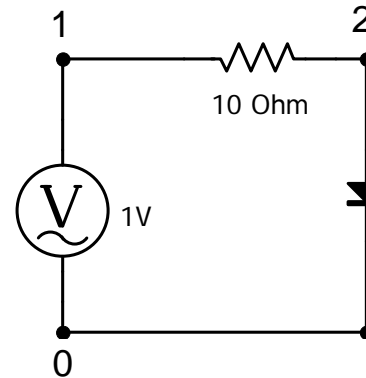
tran 0.1ms 100ms

plot v(1)

plot -i(v1)

.endc

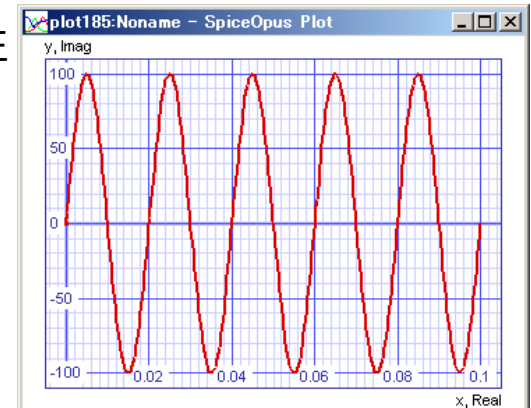
.end



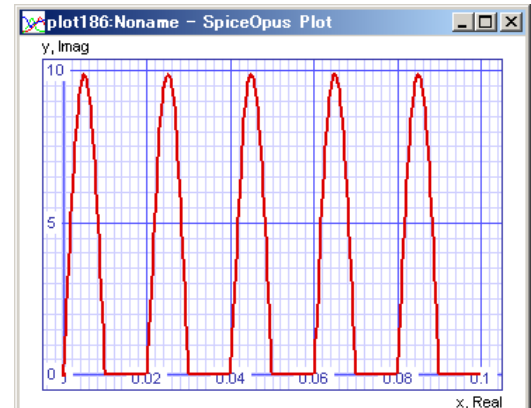
ダイオード

(電流が一方向にしか流れない)

入力電圧

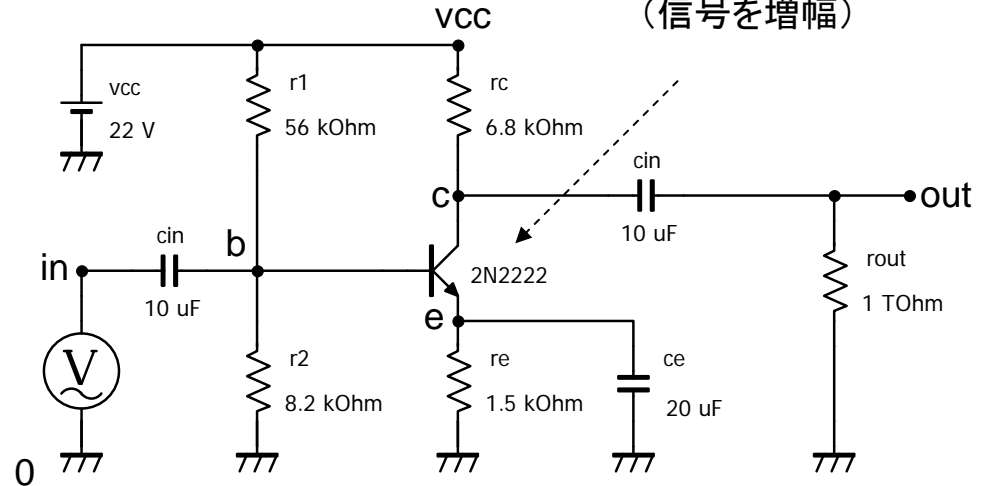


電流



バイポーラトランジスタ (1)

バイポーラトランジスタ
(信号を増幅)



ex.15: BT Amp.
 * Biasing Supply
 vcc vcc 0 dc 22
 * Input Signal Source
 vin in 0 dc 0 ac 1mv 0 sin(0 1mv 10kHz)
 * Biasing Resistors
 r1 vcc b 56k
 r2 b 0 8.2k
 re e 0 1.5k
 rc vcc c 6.8k
 * Coupling Capacitors
 cin in b 10uf
 cout c out 10uf
 * Emitter Capacitor
 ce e 0 20uf
 * Load Resistance
 rout out 0 1e12
 * Transistor
 q c b e q2n2222

モデル名

.include q2n2222.lib

.control
 set units=degree
 destroy all

時間領域解析

tran 1us 500us
 plot v(out) (v(in)*100) vs (time*1000)

ac dec 10 10Hz 100MegHz
 let gain=v(out)/v(in)
 plot db(gain)
 plot ph(gain)

変数gainの定義

.endc
 .end
 周波数領域解析

トランジスタの定義ファイル
 の読み込み

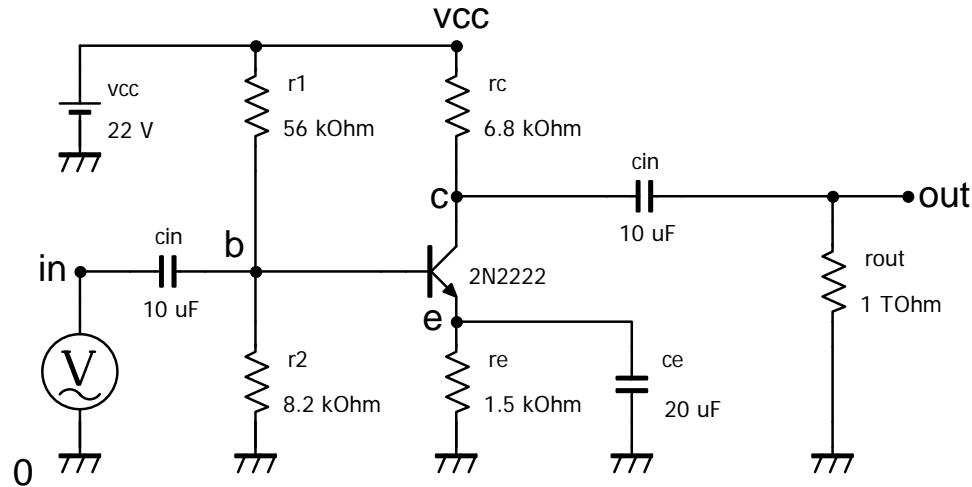
c:\SpiceOpus\lib\scripts\q2n2222.lib

```
.model q2n2222 NPN(Is=14.34f Xti=3 Eg=1.11 Vaf=74.03 Bf=255.9 Ne=1.307
+ Ise=14.34f Ikf=.2847 Xtb=1.5 Br=6.092 Nc=2 Isc=0 Ikr=0 Rc=1
+ Cjc=7.306p Mjc=.3416 Vjc=.75 Fc=.5 Cje=22.01p Mje=.377 Vje=.75
+ Tr=46.91n Tf=411.1p Itf=.6 Vtf=1.7 Xtf=3 Rb=10)
```

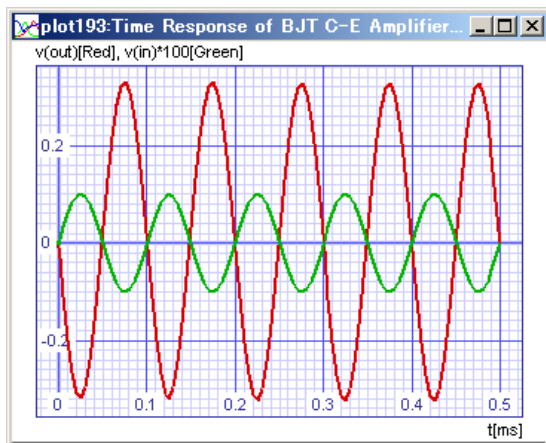
デバイスの定義を別のファイルから読み込むことができる。
 SpiceOpusは、デフォルトではc:\SpiceOpus\lib\scriptsのフォルダに
 あるファイルを読み込む。

Fig. 21

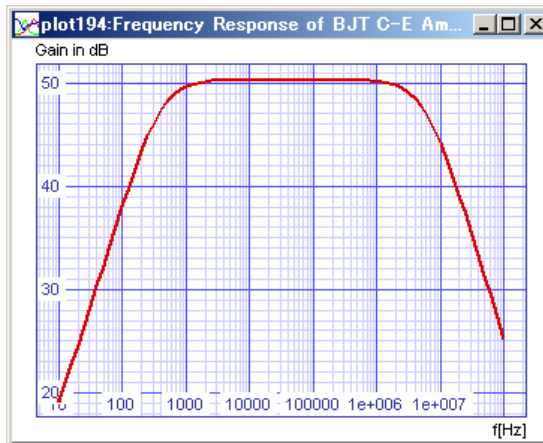
バイポーラトランジスタ (2)



入力電圧&出力電圧



増幅特性(振幅)vs 周波数



増幅特性(位相)vs 周波数

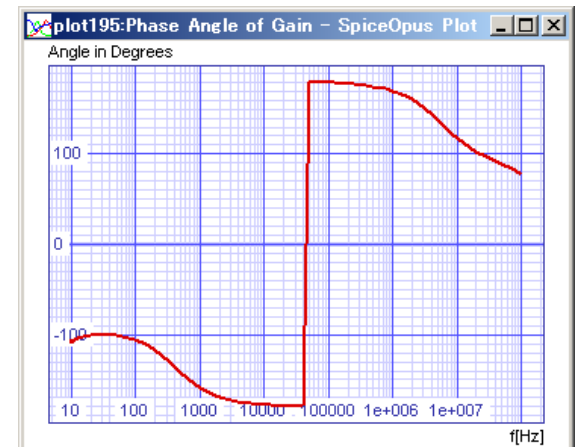


Fig. 22

MOSFET

ex16.cir

ex.16 MOSFET

vds drain 0 dc 0
vgs gate 0 dc 0

MOSFETの定義

* MOSFET

m1 drain gate 0 0 MOD1 L=4U W=6U AD=10P AS=10P
.MODEL MOD1 NMOS VTO=-2 NSUB=1.0E15 UO=550

.control
set units=degree
destroy all

vds: 0-10V (0.5Vステップ) で計算

vgs: 0-5V (1Vステップ) で計算

dc vds 0 10V 0.5V vgs 0 5V 1V

let ivgs0 = -i(vds)[0, 20] ← vgs=0Vの場合のidsをivgs0にコピー
let ivgs1 = -i(vds)[21, 41]
let ivgs2 = -i(vds)[42, 62]
let ivgs3 = -i(vds)[63, 83]
let ivgs4 = -i(vds)[84, 104]
let ivgs5 = -i(vds)[105, 125]

[0,20]の意味: 配列i(vds)の0番から20番のデータをコピー (0, 0.5V, 1.0V... 19.5V, 20.0V = 21個のデータ)

plot ivgs0 ivgs1 ivgs2 ivgs3 ivgs4 ivgs5
+xlabel 'Vds [V]'
+ylabel 'Ids [A]'

.endc
.end

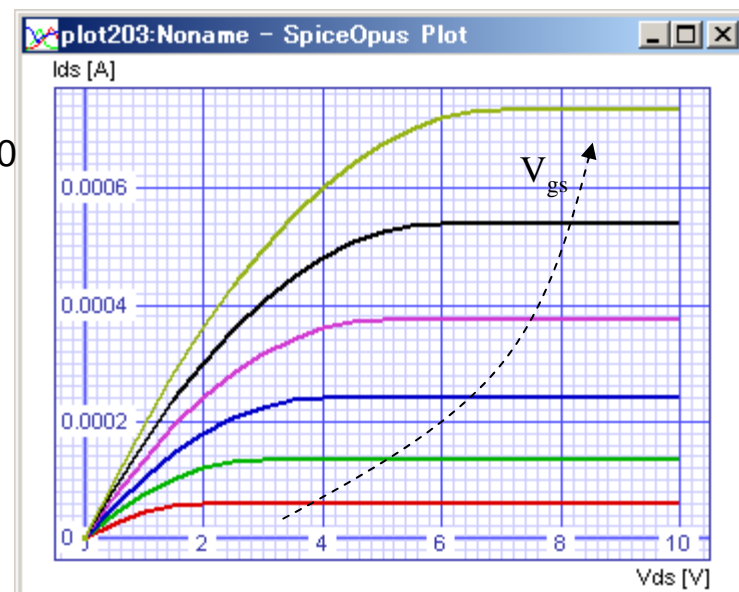
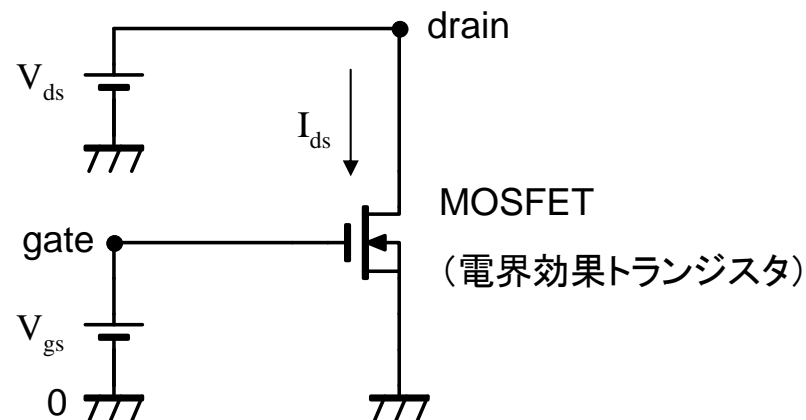


Fig. 23

演算増幅器(オペアンプ)1

ex17.cir

```
ex.17: Inverting amplifier using OPAMP 741

vin input 0 dc 0 ac 0.5v 0 sin(0 0.5v 1kHz)
vp 3 0 dc 15
vn 0 4 dc 15 } 演算増幅器の電源(±15V)

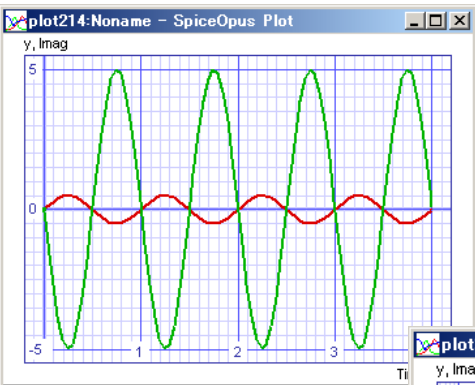
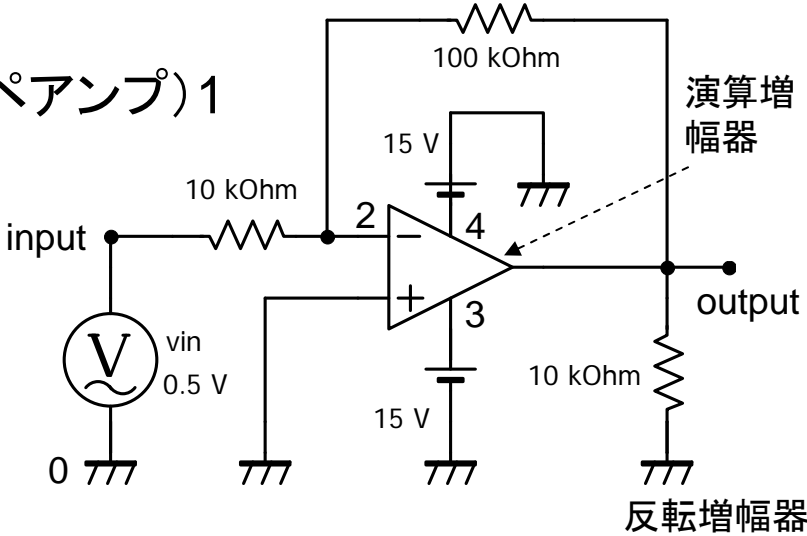
rin input 2 10k
rf 2 output 100k
rout output 0 10k

x 0 2 3 4 output ua741 ← 演算増幅器の定義
.include ua741.lib

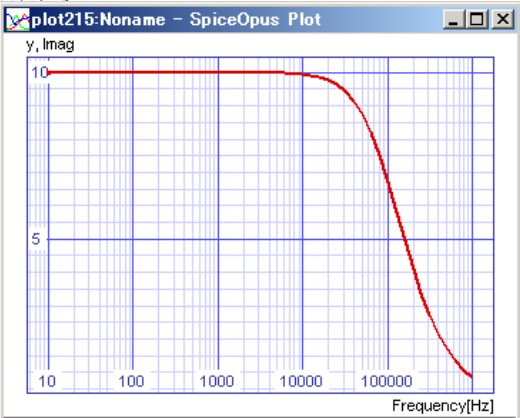
.control
destroy all

tran 0.01ms 4ms
plot v(input) vs (time*1000)
+ v(output) vs (time*1000) } 時間領域解析
+ xlabel 'Time[ms]'
destroy all
ac dec 10 10Hz 1MegHz
plot mag(v(output))/mag(v(input)) vs frequency } 周波数領域解析
+ xlabel 'Frequency[Hz]'

.endc
.end
```



時間領域解析



周波数特性

Fig. 24

演算増幅器(オペアンプ)②

```

*////////////////////////////////////
* uA741/A/C/E OPAMP MACRO-MODEL
*////////////////////////////////////
*
* connections: non-inverting input
* | inverting input
* | | positive power supply
* | | | negative power supply
* | | | | output
* | | | |
* | | | |
.SUBCKT uA741 1 2 99 50 28
*
* Features:
* Improved performance over industry standards
* Plug-in replacement for LM709,LM201,MC1439,748

* Input and output overload protection
*
*****INPUT STAGE*****
*
IOS 2 1 20N
*^Input offset current
R1 1 3 250K
R2 3 2 250K
I1 4 50 100U
R3 5 99 517
R4 6 99 517
Q1 5 2 4 QX
Q2 6 7 4 QX
*Fp2=2.55 MHz
C4 5 6 60.3614P
*****COMMON MODE EFFECT*****
*
I2 99 50 1.6MA
*^Quiescent supply current
EOS 7 1 POLY(1) 16 49 1E-3 1
*Input offset voltage.^
R8 99 49 40K
R9 49 50 40K
*

```

```

*****OUTPUT VOLTAGE LIMITING*****
V2 99 8 1.63
D1 9 8 DX
D2 10 9 DX
V3 10 50 1.63
*
*****SECOND STAGE*****
*
EH 99 98 99 49 1
G1 98 9 5 6 2.1E-3
*Fp1=5 Hz
R5 98 9 95.493MEG
C3 98 9 333.33P
*
*****POLE STAGE*****
*
*Fp=30 MHz
G3 98 15 9 49 1E-6
R12 98 15 1MEG
C5 98 15 5.3052E-15
*
*****COMMON-MODE ZERO
STAGE*****
*
*Fpcm=300 Hz
G4 98 16 3 49 3.1623E-8
L2 98 17 530.5M
R13 17 16 1K
*

```

```

*****OUTPUT STAGE*****
*
F6 50 99 POLY(1) V6 450U 1
E1 99 23 99 15 1
R16 24 23 25
D5 26 24 DX
V6 26 22 0.65V
R17 23 25 25
D6 25 27 DX
V7 22 27 0.65V
V5 22 21 0.18V
D4 21 15 DX
V4 20 22 0.18V
D3 15 20 DX
L3 22 28 100P
RL3 22 28 100K
*
*****MODELS USED*****
*
.MODEL DX D(IS=1E-15)
.MODEL QX NPN(BF=625)
*
.ENDS

```

オペアンプua741の定義ファイル
c:¥SpiceOpus¥lib¥scripts¥ua741.lib

Fig. 25

解析データの取り出し方法

- print文を利用

source ex18.cir > print_out.txt

として、コンソール出力をファイルにリダイレクト

ex.18: Data output

* AC Voltage source

v1 a 0 dc 0 ac 1

* Resistor

r1 a b 1k

* Capacitor

c1 b 0 1u

.control

set units=degree

destroy all

ac dec 100 25 10Meg

print mag(v(b))

.endc

.end

- write文を利用

ex.19: Data output

* AC Voltage source

v1 a 0 dc 0 ac 1

* Resistor

r1 a b 1k

* Capacitor

c1 b 0 1u

.control

set units=degree

destroy all

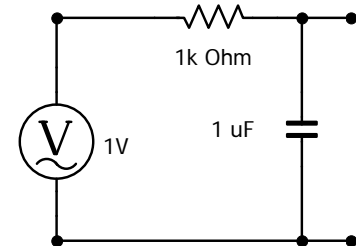
ac dec 100 25 10Meg

write write_out.txt mag(v(b))

出力するファイル名

.endc

.end



- 出力される結果は、どちらもそのままではExcelなどで読めない形式になっているので、perl, awk, sedなどのスクリプト言語で少し整形する必要がある。

Fig. 26

SPICE OPUSの使用上注意

- 一般的に使用されている他のSPICEと若干フォーマットが異なる。

通常のSPICEのドットコマンドの先頭のドットを省略して、.controlと.endcの間に書く。

ex.

SPICE OPUS: op

他のスパイス: .OP

*ただし、.model文はそのまま使える

各種コマンドは小文字で書く

ex.

op, ac, tran, print, etc.

おまけ

オフィス用ソフト

• 文章作成ソフト	• グラフ作成ソフト	• 図面作成ソフト	• プレゼンソフト
<input type="checkbox"/> MS Word: 有料	<input type="checkbox"/> MS Excel: 有料	<input type="checkbox"/> MS PowerPoint: 有料	<input type="checkbox"/> MS PowerPoint: 有料
<input type="checkbox"/> OpenOffice: フリー	<input type="checkbox"/> OpenOffice: フリー	<input type="checkbox"/> OpenOffice: フリー	<input type="checkbox"/> OpenOffice: フリー
<input type="checkbox"/> TeX: フリー	<input type="checkbox"/> gnuplot: フリー	<input type="checkbox"/> Illustrator: 有料	
<input type="checkbox"/> FrameMaker: 有料	<input type="checkbox"/> Kaleida Graph: 有料		

開発環境

• Linuxなど	• Windows
<input type="checkbox"/> c, perlなど各種ツールがフリーで利用可	<input type="checkbox"/> cygwinをインストールすれば、unixライクな環境をWindows上でもフリーで利用可能。

Fig. 28