femto-st sciences & technologies

ensmm
École Nationale Supérieure
de Mécanique et des Microtechniques

UFC
Université de Franche-Comté

utbm
Université de Technologie
Belfort-Montbéliard

CNRS
CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE

Initiation à Spice3

Rémi Brendel

Spice3

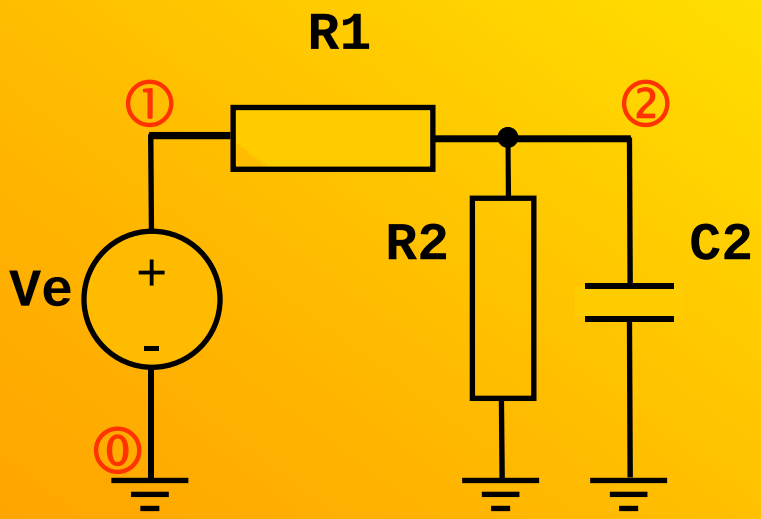
Caractéristiques principales

- version « interactive » de Spice2
- circuit décrit par un fichier d'entrée chargé dans une fenêtre de commande
- exécution des instructions d'analyse grâce à l'interface interactive
- examen et traitement des résultats grâce au post processeur intégré

} *nutmeg*

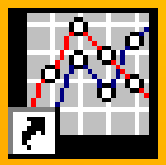
- plusieurs versions dérivées : WinSpice, Spice Opus, Spice+…

Utilisation de Spice3 : exemple 1

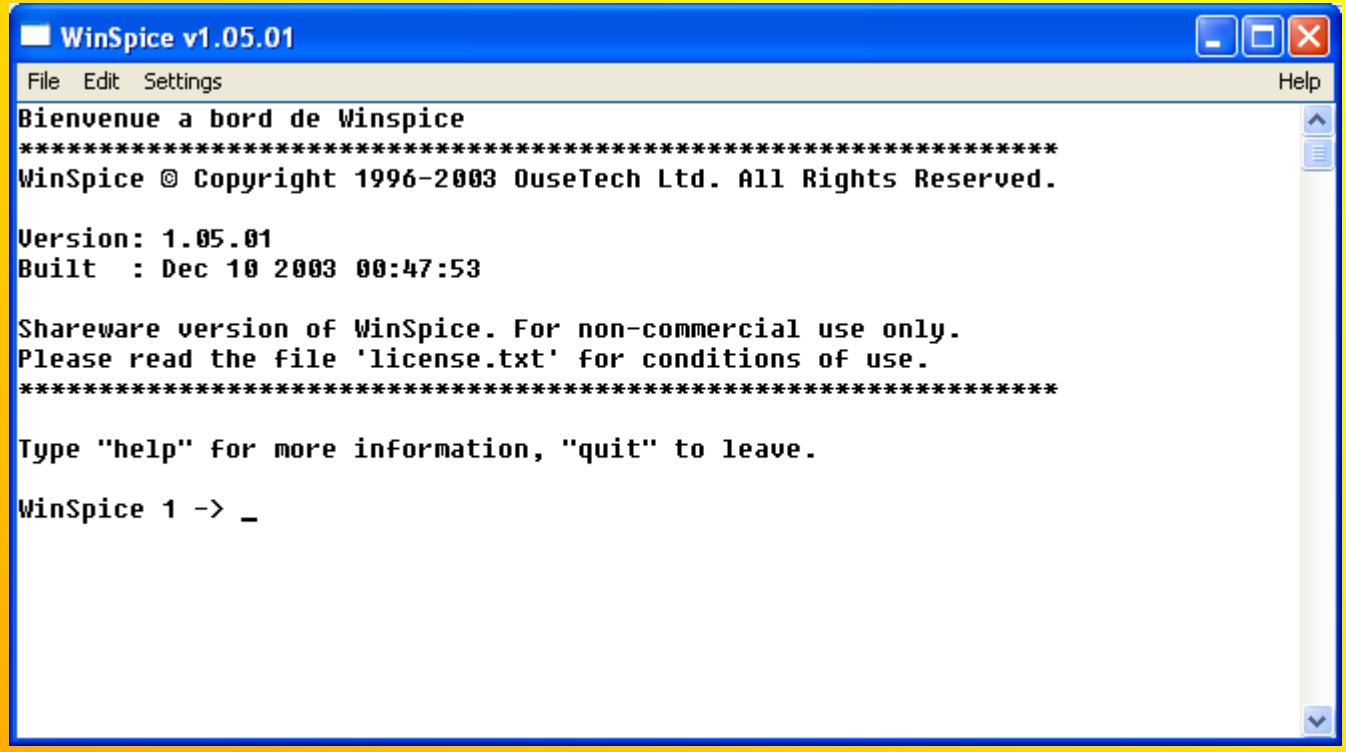



```
* Simple Pont RC *
* RC.cir
.control
.endc
R1  1  2  1K
R2  2  0  5K
C2  2  0  100n
* Source de tension
Ve  1  0  DC  1
+         AC  1
+         SIN(0 100m 10k)
.END
```

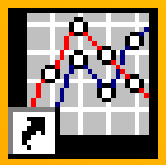
Exécution de Spice3



Winspice.lnk



Exécution de Spice3



Winspice.lnk

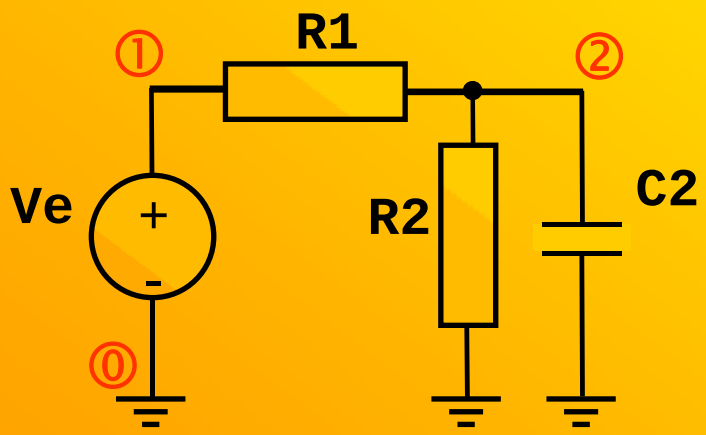
```
WinSpice 1 -> cd
Current directory:
C:\_RB_Disque_D\Spectre\Expose
WinSpice 2 -> source "RC.cir"
Reading .\RC.cir

Circuit: * Simple Pont RC *

WinSpice 3 -> listing
        * Simple Pont RC *

From file 'RC.cir':-
     3 : .control
     4 : .endc
     5 : r1  1  2  1k
     6 : r2  2  0  5k
     7 : c2  2  0  100n
     9 : ve  1  0  dc  1
    10 : +         ac  1
    11 : +         sin(0 100m 10k)
    13 : .end
```

Principales commandes d'analyse




```
WinSpice 4 -> dc ve -10 10 10m

TEMP=27 deg C
DC analysis ... 100%


WinSpice 5 -> ac dec 100 1 10meg

TEMP=27 deg C
AC analysis ... 100%


WinSpice 6 -> tran 1u 300u 0 1u

TEMP=27 deg C
Transient analysis ... 100%
```

Résultats d'analyse

« Feuilles de calculs (*plot*)»

```
WinSpice 7 -> setplot
        Type the name of the desired plot:

        new     New plot
Current tran1   * Simple Pont RC * (Transient Analysis)
        ac1     * Simple Pont RC * (AC Analysis)
        dc1     * Simple Pont RC * (DC transfer characteristic)
        const   Constant values (constants)
? const
```

Constantes prédéfinies

```
WinSpice 8 -> print all
boltz = 1.380620e-23
c = 2.997925e+08
e = 2.718282e+00
echarge = 1.602190e-19
false = 0.000000e+00
i = 0.000000e+00,1.000000e+00
kelvin = -2.73150e+02
no = 0.000000e+00
pi = 3.141593e+00
planck = 6.626200e-34
true = 1.000000e+00
yes = 1.000000e+00
```

Visualisation des résultats

Graphes

```
WinSpice 10 -> setplot
        Type the name of the desired plot:

        new     New plot
        tran1   * Simple Pont RC * (Transient Analysis)
        ac1     * Simple Pont RC * (AC Analysis)
        dc1     * Simple Pont RC * (DC transfer characteristic)
Current const   Constant values (constants)
? dc1
WinSpice 11 -> plot v(2) i(ve) v(2)*i(ve)
WinSpice 12 -> plot v(2) xlimit 0 10 ylimit 0 5

WinSpice 14 -> display all
    V(1)                : voltage, real, 2001 long
    V(2)                : voltage, real, 2001 long
    sweep               : voltage, real, 2001 long [default scale]
    ve#branch           : current, real, 2001 long
```

Vecteurs

Visualisation des résultats

Tableaux de variables réelles

```
WinSpice 15 -> print all
* Simple Pont RC *
                          DC transfer characteristic  Fri Jun 06 12:28:01 2008
--------------------------------------------------------------------------------
Index   sweep           V(1)            V(2)            sweep
--------------------------------------------------------------------------------
0       -1.000000e+01   -1.000000e+01   -8.333333e+00   -1.000000e+01
1       -9.990000e+00   -9.990000e+00   -8.325000e+00   -9.990000e+00
2       -9.980000e+00   -9.980000e+00   -8.316667e+00   -9.980000e+00
3       -9.970000e+00   -9.970000e+00   -8.308333e+00   -9.970000e+00
4       -9.960000e+00   -9.960000e+00   -8.300000e+00   -9.960000e+00
5       -9.950000e+00   -9.950000e+00   -8.291667e+00   -9.950000e+00

-- hit return for more, ? for help -- q
```

Visualisation des résultats

Changement de feuille

```
WinSpice 16 -> setplot ac1
WinSpice 17 -> plot vdb(2)
WinSpice 18 -> plot vp(2)
WinSpice 19 -> plot vp(2)*180/pi
WinSpice 20 -> plot vi(2) vs vr(2)
WinSpice 21 -> display all
    V(1)              : voltage, complex, 701 long
    V(2)              : voltage, complex, 701 long
    frequency         : frequency, complex, 701 long, grid = xlog [default scale]
    ve#branch         : current, complex, 701 long
```

Visualisation des résultats

Tableaux de variables complexes

```
WinSpice 7 -> print all
...

        -- hit return for more, ? for help --

 * Simple Pont RC *
                                AC Analysis  Mon Jun 09 08:47:43  2008
--------------------------------------------------------------------------
Index   frequency                          V(2)
--------------------------------------------------------------------------

0         1.000000e+00,   0.000000e+00     8.333331e-01,   -4.363322e-04
1         1.258925e+00,   0.000000e+00     8.333330e-01,   -5.493096e-04
2         1.584893e+00,   0.000000e+00     8.333328e-01,   -6.915396e-04
3         1.995262e+00,   0.000000e+00     8.333324e-01,   -8.705965e-04
4         2.511886e+00,   0.000000e+00     8.333319e-01,   -1.096015e-03
5         3.162278e+00,   0.000000e+00     8.333310e-01,   -1.379800e-03
6         3.981072e+00,   0.000000e+00     8.333297e-01,   -1.737063e-03
7         5.011872e+00,   0.000000e+00     8.333276e-01,   -2.186827e-03
8         6.309573e+00,   0.000000e+00     8.333242e-01,   -2.753041e-03
9         7.943282e+00,   0.000000e+00     8.333189e-01,   -3.465851e-03
10        1.000000e+01,   0.000000e+00     8.333105e-01,   -4.363204e-03
11        1.258925e+01,   0.000000e+00     8.332971e-01,   -5.492860e-03
```

Complexes

Visualisation des résultats

État de polarisation des composants

```
WinSpice 20 -> op

TEMP=27 deg C
DC Operating Point ... 100%

WinSpice 21 ->
```

Visualisation des résultats

État de polarisation des composants

```
WinSpice 21 -> show all
 Capacitor: Fixed capacitor
 device            c2
 model              C
capacitanc      1e-07
         i   2.04e-307
         p   2.04e-307

 Resistor: Simple linear resistor
 device            r2          r1
 model              R           R
resistance      5e+03       1e+03
         i   0.000167   0.000167
         p   0.000139   2.78e-05
       tc1           0           0
       tc2           0           0

 Vsource: Independent voltage source
 device            ve
        dc           1
     acmag           1
         i  -0.000167
         p   0.000167
```

Visualisation des résultats

État de polarisation des composants

```
WinSpice 21 -> show all : all
...
...
```

Visualisation des résultats

Création de feuille

```
WinSpice 10 -> setplot tran1
WinSpice 11 -> display all
    V(1)                : voltage, real, 309 long
    V(2)                : voltage, real, 309 long
    time                : time, real, 309 long [default scale]
    ve#branch           : current, real, 309 long
WinSpice 12 -> plot v(2)


WinSpice 17 -> setplot new
WinSpice 18 -> plot v(2)
Error: no such vector v(2)
WinSpice 19 -> plot dc1.v(2)
```

Visualisation des résultats

Opérations entre feuilles

```
WinSpice 23 -> alter c2=500n
WinSpice 24 -> ac dec 100 1 10meg

TEMP=27 deg C
AC analysis ... 100%
WinSpice 26 -> plot ac1.v(2) ac2.v(2)


WinSpice 27 -> plot ac1.vdb(2) ac2.vdb(2)
Error: no such function as 'ac1.vdb'.
WinSpice 29 -> setplot ac1
WinSpice 30 -> let v100 = vdb(2)
WinSpice 31 -> setplot ac2
WinSpice 32 -> let v500 = vdb(2)
WinSpice 33 -> plot v100 v500
WinSpice 34 -> plot v100-v500
```

Visualisation des résultats

Destruction de feuilles

```
WinSpice 21 -> setplot
        Type the name of the desired plot:
        new     New plot
Current tran3   * Test structures controle * (Transient Analysis)
        tran2   * Test structures controle * (Transient Analysis)
        tran1   * Test structures controle * (Transient Analysis)
        const   Constant values (constants)
?

WinSpice 22 -> destroy tran2
WinSpice 23 -> setplot
        Type the name of the desired plot:
        new     New plot
Current tran3   * Test structures controle * (Transient Analysis)
        tran1   * Test structures controle * (Transient Analysis)
        const   Constant values (constants)
?

WinSpice 24 -> destroy all
WinSpice 25 -> setplot
        Type the name of the desired plot:
        new     New plot
Current const   Constant values (constants)
?
```

Sauvegarde des résultats

```
WinSpice 34 -> print v100 v500 > RC.out
WinSpice 37 -> write RC2.out v100 v500
WinSpice 38 -> shell dir
```

Fichier RC.out

```
* Simple Pont RC *
                                AC Analysis  Fri Jun 06 12:43:00  2008
--------------------------------------------------------------------------------
Index    frequency                              v100              v500
--------------------------------------------------------------------------------
0         1.000000e+00,    0.000000e+00    -1.583626e+00    -1.583655e+00
1         1.023293e+00,    0.000000e+00    -1.583626e+00    -1.583656e+00
2         1.047129e+00,    0.000000e+00    -1.583626e+00    -1.583658e+00
3         1.071519e+00,    0.000000e+00    -1.583626e+00    -1.583659e+00
4         1.096478e+00,    0.000000e+00    -1.583626e+00    -1.583661e+00
5         1.122018e+00,    0.000000e+00    -1.583626e+00    -1.583662e+00
6         1.148154e+00,    0.000000e+00    -1.583626e+00    -1.583664e+00
7         1.174898e+00,    0.000000e+00    -1.583627e+00    -1.583666e+00
8         1.202264e+00,    0.000000e+00    -1.583627e+00    -1.583668e+00
9         1.230269e+00,    0.000000e+00    -1.583627e+00    -1.583670e+00
10        1.258925e+00,    0.000000e+00    -1.583627e+00    -1.583672e+00
...
```

Sauvegarde des résultats

Fichier RC2.out

```
Title: * Simple Pont RC *
Date: Fri Jun 06 12:43:00  2008
Plotname: AC Analysis
Flags: complex
No. Variables: 2
No. Points: 701
Variables:
          0         frequency frequency grid=3
          1         v100      decibel
Values:
 0        1.000000000000000e+00,0.000000000000000e+00
          -1.583626111595316e+00,0.0
 1        1.023292992280754e+00,0.000000000000000e+00
          -1.583626167708574e+00,0.0
...
700       9.999999999999788e+06,0.000000000000000e+00
          -7.596359752557375e+01,0.0
Title: * Simple Pont RC *
Date: Fri Jun 06 14:32:50  2008
Plotname: AC Analysis
Flags: complex
No. Variables: 2
No. Points: 701
Variables:
          0         frequency frequency grid=3
          1         v500      decibel
Values:
 0        1.000000000000000e+00,0.000000000000000e+00
          -1.583654686925004e+00,0.0
 1        1.023292992280754e+00,0.000000000000000e+00
          -1.583656089747036e+00,0.0
...
```

Les objets de Spice3

Constantes

• Constantes = constantes numériques prédéfinies : c, e, pi… utilisables dans des expressions

```
WinSpice 48 -> print 2*pi*1k
2*pi*1k = 6.283185e+03

WinSpice 52 -> print const.all
boltz = 1.380620e-23
c = 2.997925e+08
e = 2.718282e+00
echarge = 1.602190e-19
false = 0.000000e+00
i = 0.000000e+00,1.000000e+00
kelvin = -2.73150e+02
no = 0.000000e+00
pi = 3.141593e+00
planck = 6.626200e-34
true = 1.000000e+00
yes = 1.000000e+00
```

Les objets de Spice3

Variables

• **Variables** = mot-clés auxquels sont affectées des valeurs numériques, logiques ou chaînes de caractères

Certaines variables prédéfinies permettent de modifier le fonctionnement du simulateur :

```
WinSpice 49 -> plot v(2)
WinSpice 50 -> set color2=blue
WinSpice 51 -> plot v(2)
WinSpice 59 -> set plotstyle=pointplot
WinSpice 60 -> plot tran1.v(2) tran2.v(2) tran3.v(2)
WinSpice 61 -> set plotstyle=combplot
WinSpice 62 -> plot v(2)
WinSpice 63 -> set plotstyle=linplot
WinSpice 64 -> plot v(2)
WinSpice 78 -> set width=132
WinSpice 79 -> print all
```

Les objets de Spice3

Variables

• **Variables** = mot-clés auxquels sont affectées des valeurs numériques, logiques ou chaînes de caractères

Les variables peuvent être crées et utilisées dans des commandes ou des expressions :

```
WinSpice 80 -> set step=0.5u
WinSpice 81 -> set fin=50u
WinSpice 82 -> set deb=0
WinSpice 83 -> set seuil=0.5u
WinSpice 84 -> tran $step $fin $deb $seuil
WinSpice 93 -> echo step
step
WinSpice 94 -> echo $step
0.5u
WinSpice 95 -> print step
Error: step: no such vector.
WinSpice 97 -> print $step*10meg
0.5u*10meg = 5.000000e+00
```

Les objets de Spice3

Variables

• **Variables** = mot-clés auxquels sont affectées des valeurs numériques, logiques ou chaînes de caractères

Les variables peuvent être des listes

```
.control
  set fruit=( abricot banane cerise )
  foreach f $fruit
    echo $f
  end
.endc
```

```
WinSpice 21 -> source "VAR.cir"
Reading .\VAR.cir
abricot
banane
cerise
WinSpice 19 -> echo $?fruit
1
WinSpice 20 -> echo $#fruit
3
WinSpice 21 ->
```

Les objets de Spice3

Variables

• **Variables** = mot-clés auxquels sont affectées des valeurs numériques, logiques ou chaînes de caractères

Les variables peuvent être détruites :

```
WinSpice 25 -> set toto
WinSpice 26 -> echo $toto
true
WinSpice 27 -> echo $?toto
1
WinSpice 28 -> unset toto
WinSpice 29 -> echo $toto
Error: toto: no such variable.
WinSpice 30 -> echo $?toto
0
```

Les objets de Spice3

Vecteurs et scalaires

• Vecteurs = tableaux de valeurs réelles ou complexes

Les vecteurs sont créés par le simulateur (tensions, courants, etc.) ou par l'utilisateur :

```
WinSpice 98 -> let x=[1 5 9]
WinSpice 99 -> print x
WinSpice 102 -> set noprintscale
WinSpice 103 -> print x
--------------------------------------------------------------------------------
Index   x
--------------------------------------------------------------------------------
0       1.000000e+00
1       5.000000e+00
2       9.000000e+00
```

Un scalaire est un vecteur à une seule composante

Les vecteurs créés par le simulateur sont accessibles dans la feuille en cours ou par *plotname.vecname*

Les vecteurs créés par l'utilisateur sont accessibles partout

Les objets de Spice3

Vecteurs et scalaires

• **Vecteurs** = tableaux de valeurs réelles ou complexes

Les vecteurs sont créés par le simulateur (tensions, courants, etc.) ou par l'utilisateur :

```
WinSpice 106 -> let y=[-2 0 10]
WinSpice 107 -> print x y
--------------------------------------------------------------------------------
Index   x               y
--------------------------------------------------------------------------------
0        1.000000e+00   -2.000000e+00
1        5.000000e+00    0.000000e+00
2        9.000000e+00    1.000000e+01
```

Les objets de Spice3

Vecteurs et scalaires

- **Vecteurs** = tableaux de valeurs réelles ou complexes

Les vecteurs sont créés par le simulateur (tensions, courants, etc.) ou par l'utilisateur :

```
WinSpice 108 -> let z = x+i*y
WinSpice 109 -> print x y z
------------------------------------------------------------------------------
Index   x               y               z
------------------------------------------------------------------------------
0        1.000000e+00   -2.000000e+00    1.000000e+00,  -2.000000e+00
1        5.000000e+00    0.000000e+00    5.000000e+00,   0.000000e+00
2        9.000000e+00    1.000000e+01    9.000000e+00,   1.000000e+01


WinSpice 110 -> print x[0]+y[2]
x[0]+y[2] = 1.100000e+01
```

Les objets de Spice3

Vecteurs et scalaires

• **Vecteurs** = tableaux de valeurs réelles ou complexes

Les vecteurs sont créés par le simulateur (tensions, courants, etc.) ou par l'utilisateur :

```
WinSpice 1 -> let x=vector(100)
WinSpice 2 -> print x
WinSpice 3 -> let y=sin(pi*x/25)
WinSpice 4 -> plot y vs x
```

Les objets de Spice3

Vecteurs et scalaires

- Vecteurs = tableaux de valeurs réelles ou complexes

Les vecteurs peuvent être détruits :

```
WinSpice 27 -> let x=[1 5 9]
WinSpice 28 -> print x
                                Constant values
                                constants  Sat Aug 16 10:55:15 PDT 1986
--------------------------------------------------------------------------
Index   x
--------------------------------------------------------------------------
0        1.000000e+00
1        5.000000e+00
2        9.000000e+00
WinSpice 29 -> unlet x
WinSpice 30 -> print x
Error: x: no such vector.
```

Les objets de Spice3

Expressions, opérateurs et fonctions

- Expression = combinaison de constantes, variables, vecteurs, opérateurs et fonctions

- Opérateurs algébriques : `+  -  *  /  ^  %`

  relationnels : `> (gt)  < (lt)  >= (ge)  <= (le)  <> (ne)  = (eq)`

  logiques : `& (and)  | (or)  ! (not)`

- Fonctions : `mag() ph() j() real() imag() db() log() ln() exp() abs() sqrt() ...`

```
WinSpice 157 -> plot abs(v(2))
```

- Fonctions définies par l'utilisateur :

```
WinSpice 11 -> define th(x) (exp(x)-exp(-x))/(exp(x)+exp(-x))
WinSpice 12 -> plot th(100k*time)
```

Les objets de Spice3

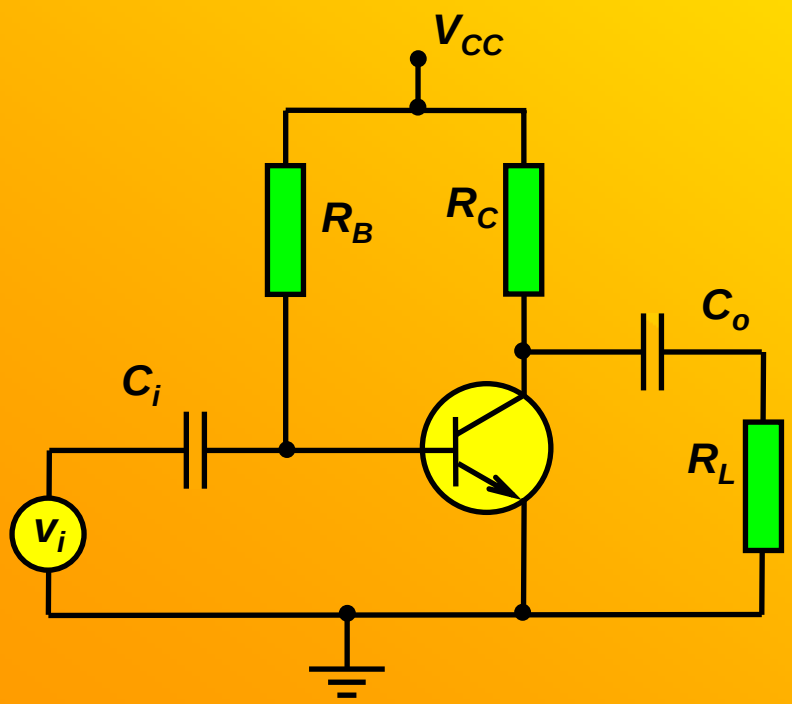
Expressions, opérateurs et fonctions

Les fonctions définies par l'utilisateur peuvent être détruites :

```
WinSpice 39 -> define ex2(x) exp(x)*exp(x)
WinSpice 40 -> plot ex2(1000*time)
WinSpice 41 -> undefine ex2
WinSpice 42 -> plot ex2(1000*time)
Error: no function as 'ex2' with that arity.
```

Exemple 2

Circuit amplificateur




```
* Ampli BJT *
* BJT.cir
.control
.endc
Vcc        10         0          10
Rb         10         4          600k
Rc         10         3          2.5k
Q          3          4          0          Q2n2857
Ci         1          4          100u
Co         3          2          100u
RL         2          0          10meg
vi         0          1          dc 0
+          ac 1
+          sin(0  1m   100k)
.MODEL Q2N2857  NPN(IS=4.0e-16 XTI=3 EG=1.11
+          VAF=120 BF=120 NE=1.5
+          ISE=3.6e-15 IKF=2.8e-2 XTB=1.5 BR=2.23
+          NC=1.7 ISC=1.2e-15
+          IKR=2.8e-2 RC=12 CJC=1.1e-12
+          MJC=.33 VJC=.75 FC=.5 CJE=1.51e-12
+          MJE=.33 VJE=.75 TR=4N TF=5.56e-11
+          ITF=1.1e-2 VTF=5  XTF=4 RB=26)
.end
```

Exemple 2

Réinitialisation et chargement

```
WinSpice 12 -> destroy all
WinSpice 13 -> reset
WinSpice 14 -> cd
Current directory: C:\_RB_Disque_D\Spectre\Expose
WinSpice 15 -> source "BJT.cir"
Reading .\BJT.cir

Circuit: * Ampli BJT *
```

Exemple 2

Paramètres de modèles

```
WinSpice 16 -> ac dec 100 1 10g

TEMP=27 deg C
AC analysis ... 100%
WinSpice 17 -> showmod
 BJT models (Bipolar Junction Transistor)
 model       q2n2857

      type       npn
        is     4e-16
        bf       120
        nf         1
       vaf       120
       cjs         0
...
       ccs         0
       vjs      0.75
       mjs         0
       xtb       1.5
        eg      1.11
       xti         3
        fc       0.5
      tnom        27
        kf         0
        af         0
```

Exemple 2

État de polarisation et schéma équivalent dynamique

```
WinSpice 20 -> show q
 BJT: Bipolar Junction Transistor
 device            q
 model       q2n2857
       ic   0.00171
       ib  1.54e-05
       ie  -0.00172
      vbe     0.753
      vbc     -4.96
       gm    0.0625
      gpi  0.000584
      gmu  4.49e-13
       gx    0.0385
       go  1.37e-05
      cpi  6.37e-12
      cmu  5.63e-13
      cbx         0
      ccs         0
       ft  1.43e+09
```

Exemple 2

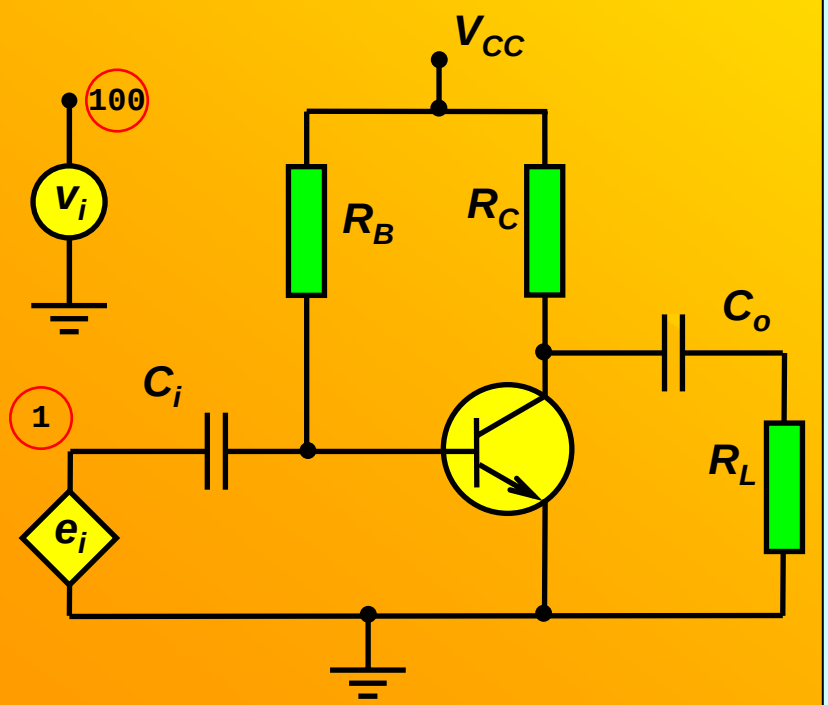
Accès aux grandeurs internes

```
WinSpice 25 -> save all @q[cpi] @q[ib] @q[ic]
WinSpice 26 -> tran 50n 30u 0 50n

TEMP=27 deg C
Transient analysis ... 100%
WinSpice 28 -> let ib=@q[ib]
WinSpice 31 -> let ic=@q[ic]
WinSpice 32 -> settype current ib ic
WinSpice 33 -> plot ic vs ib
```

Analyse paramétrique

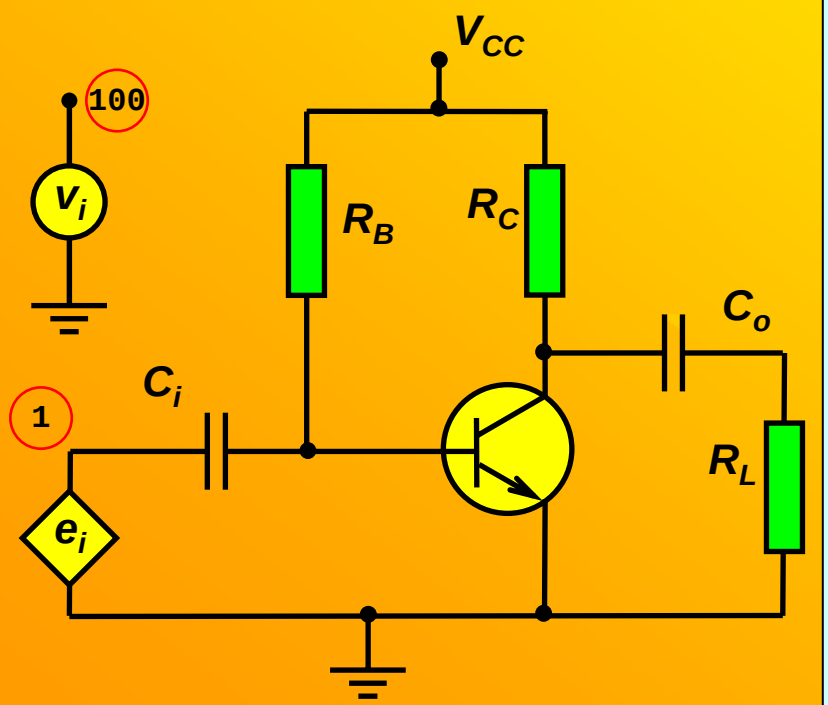
Première solution : utilisation d'une source commandée linéaire




```
* Ampli BJT *
* BJT.cir
Vcc       10        0         10
Rb        10        4         600k
Rc        10        3         2.5k
Q         3         4         0         Q2n2857
...
ei        0         1         0         100       1
vi        0         100       dc 0
+         ac 1
+         sin(0  10m   100k)
.control
  let k = 1
  while k < 8
    alter @ei[gain]=k
    tran 50n 30u  0  50n
    k = k*2
  end
.endc
.end
```

Analyse paramétrique

Autre solution : modification des caractéristiques temporelles de la source




```
* Ampli BJT *
* BJT.cir
Vcc       10        0         10
Rb        10        4         600k
Rc        10        3         2.5k
Q         3         4         0         Q2n2857
...
vi        0         1         dc 0
+         ac 1
+         sin(0  10m   100k)
.control
  let carsin = [0  10m  100k]
  let k = 1
  while k < 8
    let carsin[1] = k*10m
    alter @vi[sin] = carsin
    tran 50n 30u  0  50n
    k = k*2
  end
.end
```

Analyse de Fourier

```
* Ampli BJT *
* BJT.cir
Vcc      10       0        10
Rb       10       4        600k
Rc       10       3        2.5k
Q        3        4        0        Q2n2857
...
ei       0        1        0        100      1
vi       0        100      dc 0
+        ac 1
+        sin(0  10m   100k)
.control
  let k = 1
  while k < 8
    alter @ei[gain]=k
    tran 10n 100u  0  10n
    k = k*2
  end
.endc
.end
```

Analyse de Fourier

```
WinSpice 3 -> linearize
WinSpice 4 -> spec 1 700k 10k v(2)
Fourier transform ... 100%
WinSpice 5 -> plot vm(2)
WinSpice 7 -> print v(2)
                    Spectrum  Mon Jun 09 13:17:52  2008
--------------------------------------------------------------------
Index   v(2)
--------------------------------------------------------------------
0       -9.513264e-01,   0.000000e+00
1       -7.630302e-07,  -1.243599e-03
2        3.701758e-07,   2.762805e-04
3       -7.772298e-08,   6.898139e-05
4       -1.848356e-07,   2.748459e-05
...
WinSpice 8 -> print vm(2) vp(2)
                     Spectrum  Mon Jun 09 13:17:52  2008
--------------------------------------------------------------------
Index   mag(v(2))       ph(v(2))
--------------------------------------------------------------------
0        9.513264e-01    3.141593e+00
1        1.243600e-03   -1.571410e+00
2        2.762807e-04    1.569456e+00
3        6.898143e-05    1.571923e+00
4        2.748521e-05    1.577521e+00
...
```

Analyse de Fourier

```
WinSpice 9 -> setplot tran3
WinSpice 10 -> fourier 100k v(2)
Fourier analysis for v(2):
  No. Harmonics: 10, THD: 18.4008 %, Gridsize: 200, Interpolation Degree: 1

Harmonic Frequency    Magnitude    Phase        Norm. Mag    Norm. Phase
-------- ---------    ---------    -----        ---------    -----------
 0       0.000000e+00 -9.48975e-01 0.000000e+00 0.000000e+00 0.000000e+00
 1       1.000000e+05 5.063985e+00 -1.90924e-01 1.000000e+00 0.000000e+00
 2       2.000000e+05 5.286170e-01 8.898129e+01 1.043876e-01 8.917222e+01
 3       3.000000e+05 6.135790e-01 -3.93474e-01 1.211653e-01 -2.02550e-01
 4       4.000000e+05 4.194293e-01 8.871216e+01 8.282595e-02 8.890309e+01
 5       5.000000e+05 8.805866e-02 1.761644e+02 1.738920e-02 1.763553e+02
 6       6.000000e+05 9.344657e-02 8.950873e+01 1.845317e-02 8.969965e+01
 7       7.000000e+05 1.239452e-01 1.775493e+02 2.447583e-02 1.777402e+02
 8       8.000000e+05 6.753246e-02 -9.43737e+01 1.333583e-02 -9.41828e+01
 9       9.000000e+05 5.970793e-03 -1.66368e+02 1.179070e-03 -1.66177e+02
```

Structures de contrôles

- while – end

```
* Test structures controle *
Ve       0        1        dc=0     sin(0  1  1k)
D1       1        2        diode
R2       2        0        1k
C2       2        0        1u
.model   diode    D()
.control
  k = 1
  while k < 4
    alter R2=5*k*1k
    tran 5u 5m 0 5u
    k = k+1
    plot v(2)
  end
.endc
.end
```

```
WinSpice 12 -> print k
k = 4.000000e+00
```

Structures de contrôles

- dowhile – end

```
* Test structures controle *
Ve       0        1        dc=0     sin(0  1  1k)
D1       1        2        diode
R2       2        0        1k
C2       2        0        1u
.model   diode    D()
.control
  k = 1
  dowhile k < 4
    alter R2=5*k*1k
    tran 5u 5m 0 5u
    k = k+1
    plot v(2)
  end
.endc
.end
```

```
WinSpice 13 -> print k
k = 4.000000e+00
```

Structures de contrôles

- repeat – end

```
* Test structures controle *
Ve       0        1        dc=0     sin(0  1  1k)
D1       1        2        diode
R2       2        0        1k
C2       2        0        1u
.model   diode    D()
.control
  k = 1
  repeat 3
    alter R2=5*k*1k
    tran 5u 5m 0 5u
    k = k+1
    plot v(2)
  end
.endc
.end
```

```
WinSpice 14 -> print k
k = 4.000000e+00
```

Structures de contrôles

- foreach – end

```
* Test structures controle *
Ve       0        1        dc=0     sin(0  1  1k)
D1       1        2        diode
R2       2        0        1k
C2       2        0        1u
.model   diode    D()
.control
  foreach res 5k 10k 15k
    alter R2=$res
    tran 5u 5m 0 5u
    plot v(2)
  end
.endc
.end
```

Structures de contrôles

- if – else – end

```
* Test if - else *
ei       0        1        0        100      1
vi       0        100      dc 0     sin(0  10m   100k)
.control
  k = 1
  repeat
    alter @ei[gain]=k
    tran 100n 100u  0  100n
    if vecmax(v(1)) > 1
       plot v(1)
       break
    else
       k = k*2
    end
  end
.endc
.end
```

Redirections

```
* Redirection *
ei       0        1        0        100      1
vi       0        100      dc 0     sin(0  10m   100k)
.control
  set nobreak
  echo "Resultat" > if.out
  echo >> if.out
  echo >> if.out
  k = 1
  repeat 4
    echo ======================== >> if.out
    alter @ei[gain]=k
    tran 1u 100u  0  1u
    print k >> if.out
    print v(1) >> if.out
    k = k*4
  end
.endc
.end
```

Redirections

```
* Redirection *
ei       0        1        0        100      1
vi       0        100      dc 0     sin(0  10m   100k)
.control
  set nobreak
  echo -n "Nom du fichier de sortie ? "
  set nfsort = $<
  echo "Resultat" > $nfsort
  echo >> $nfsort
  echo >> $nfsort
  k = 1
  repeat 4
    echo ======================== >> $nfsort
    alter @ei[gain]=k
    tran 1u 100u  0  1u
    print k >> $nfsort
    print v(1) >> $nfsort
    k = k*4
  end
.endc
.end
```

Scripts

Fichier « date »

```
* Affiche la date et l'heure dans WinSpice *
.control

begin
  setplot new
  echo $curplotdate
  destroy {$curplot}
end
```

```
WinSpice 9 -> date
Reading .\date
Reading .\date
Thu Jun 12 11:19:27  2008
WinSpice 10 ->
```