

Notice d'utilisation simplifiée de LTspice



Christophe Viallon, 2010 - 2014
Université Paul Sabatier, Toulouse III

Table des matières

1.Introduction.....	2
2.Présentation.....	2
3.Créer un nouveau projet.....	3
4.Ajouter des composants.....	3
5.Je ne trouve pas le composant que je souhaite utiliser.....	3
5.1. <i>Chercher dans la liste des composants connus du logiciel.....</i>	3
5.2. <i>Mais il ne figure pas dans cette liste. Que faire ?.....</i>	4
5.3. <i>Et si maintenant je veux rajouter un circuit intégré inconnu de LTspice.....</i>	5
5.3.1.Créer sa propre librairie.....	6
5.3.2.Rajouter manuellement le modèle à la simulation.....	7
6.Effectuer une simulation.....	7
6.1. <i>Définir la source.....</i>	8
6.2. <i>Définir le type de simulation.....</i>	8
6.3. <i>Pour les problèmes de convergence.....</i>	9
6.4. <i>Effectuer une simulation en balayant une variable entre deux extrêmes.....</i>	9
6.5. <i>Analyses Monte Carlo et pire cas.....</i>	9
6.5.1. <i>Analyse Monte-Carlo en utilisant la fonction mc(x,y) intégrée au logiciel.....</i>	9
6.5.2. <i>Analyse pire-cas à partir d'une fonction créée par l'utilisateur.....</i>	9
6.5.3. <i>Définir une analyse Monte-Carlo à partir d'une distribution gaussienne de valeurs.....</i>	10
7.Affichage des résultats.....	10
7.1. <i>Ajouter enlever une courbe.....</i>	11
7.2. <i>Gestion des curseurs.....</i>	11
7.3. <i>Effectuer des opérations mathématiques sur les résultats.....</i>	11
7.4. <i>Quelques autres options disponibles.....</i>	12
7.5. <i>Affichage automatique des mêmes grandeurs à chaque simulation.....</i>	13
7.6. <i>Sauvegarder une configuration d'affichage.....</i>	13

1. Introduction

LTspice¹ est un logiciel de simulation électrique de la société Linear Technology, qui a été développé essentiellement pour faire la promotion des circuits intégrés conçus par ce fabricant, en permettant de simuler le fonctionnement d'une grande partie des circuits analogiques présents dans leur catalogue (régulateurs à découpage, amplificateurs opérationnels, filtres à capacités commutées, etc.). Le logiciel n'est pas commercialisé mais plutôt mis à la disposition des visiteurs de leur site web. Toutefois, et contrairement à l'immense majorité des logiciels de simulations gratuits en libre accès, celui-ci ne possède aucune limitation volontaire quant au nombre maximum de nœuds du circuit à simuler.

LTspice est basé sur le moteur de simulation SPICE, dont les débuts remontent aux années 1970, et qui était alors développé au *Electronics Research Laboratory* de l'université de Californie, Berkeley. Certaines améliorations du code ont été tout spécialement intégrées à LTspice pour lui permettre d'accélérer notablement les temps de simulations pour le cas très spécifique des circuits analogiques à découpages (ex. convertisseurs DC-DC, filtres à capacités commutées), une spécialité du fabricant.

La gratuité de ce logiciel a permis l'émergence d'une importante communauté d'utilisateurs. Parmi les sources d'informations disponibles sur le net, citons le groupe *Yahoo!* dédié à LTspice, qui regroupe de nombreuses astuces d'utilisation du logiciel avec une documentation impressionnante : <http://groups.yahoo.com/group/LTspice/> . Son accès nécessite une inscription préalable.

Enfin, bien qu'initialement développé pour les systèmes d'exploitations *Microsoft Windows*, celui-ci fonctionne parfaitement sous *Linux*, par l'intermédiaire de *Wine* et, par extension, sous *MacOS* avec *CrossOver*, qui n'est autre que l'équivalent payant de *Wine*.

L'ambition de ce document est de permettre à tout nouvel utilisateur de prendre le logiciel en main très rapidement, sans avoir à lire des "tonnes" de documentation.

2. Présentation

Au lancement de l'application, le logiciel se présente comme un éditeur graphique de schéma électrique. Le logiciel dispose également d'un outils de visualisation des résultats de simulation, et de plusieurs bibliothèques de composants pour les éléments passifs traditionnels (Résistances, condensateurs, inductances, diodes, etc.), pour quelques composants actifs (essentiellement des produits *LT* mais aussi des transistors de différents types) ainsi que pour des fonctions mathématiques basiques que nous décrivons plus tard (sources idéales contrôlées, ligne de transmission, etc.).

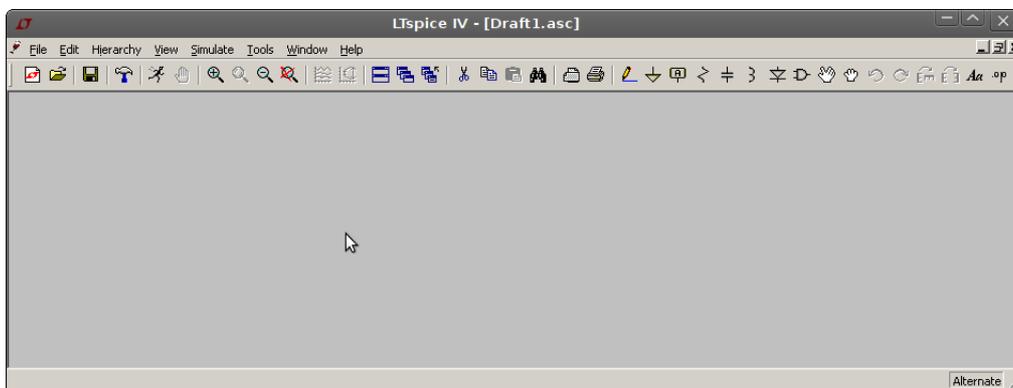


Illustration 1: LTspice une fois lancé.

¹ On peut le télécharger gratuitement ici : <http://www.linear.com/designtools/software/ltspice.jsp>

3. Créer un nouveau projet

Ici, la notion de projet n'est pas aussi contraignante que sur certains autres simulateurs de type SPICE. LTspice crée juste un nouveau fichier *.asc qui contiendra le dessin du circuit. Pour créer un nouveau circuit il suffit de faire **File** → **New schematic** ou bien de cliquer sur l'icône correspondante parmi toutes celles mises à disposition de l'utilisateur.

Une nouvelle page s'ouvre et il est alors maintenant possible de rajouter tous les composants que l'on désire.

4. Ajouter des composants

Tout est accessible depuis les différents boutons mis à disposition. Pour connaître les raccourcis clavier correspondants, il suffit de faire dérouler le menu **Edit** pour les avoir.

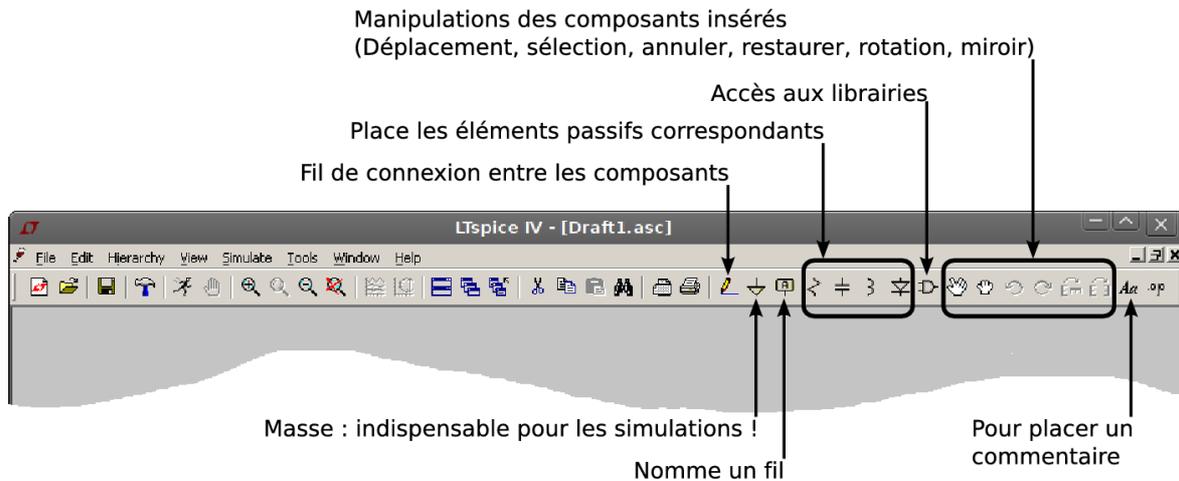


Illustration 2: boutons relatifs à la manipulation de composants.

Pour modifier la valeur d'un composant inséré, pointer la souris au-dessus du composant en question. Un doigt apparaît, et il suffit alors de faire un clic droit de la souris pour avoir accès aux propriétés du composant.

Petit rappel, les unités sous SPICE sont définies de la manière suivante :

$K = k = \text{kilo} = 10^3$	$M = m = \text{milli} = 10^{-3}$
$MEG = \text{meg} = 10^6$	$U = u = \text{micro} = 10^{-6}$
$G = g = \text{giga} = 10^9$	$N = n = \text{nano} = 10^{-9}$
$T = t = \text{terra} = 10^{12}$	$P = p = \text{pico} = 10^{-12}$

5. Je ne trouve pas le composant que je souhaite utiliser

Plusieurs cas peuvent se présenter, que nous allons énumérer ici.

5.1. Chercher dans la liste des composants connus du logiciel

Si le composant n'est pas une résistance, ni un condensateur, ni une inductance, ni une diode (tous ces composants peuvent être placés sur le plan de travail directement à partir des boutons prévus à cet effet), il faut aller chercher dans les bibliothèques de LTspice à l'aide du bouton correspondant, ce qui ouvre la fenêtre ci-contre :

Là, on sélectionne le type de composant souhaité, un transistor bipolaire de type NPN dans le cas présent. On positionne le composant dans le schéma puis on édite ses propriétés (bouton droit de la souris au-dessus du composant).

Pour définir le transistor, il suffit alors de cliquer sur le bouton **Pick new transistor** pour choisir le composant souhaité. Voir la capture d'écran suivante à titre d'exemple.

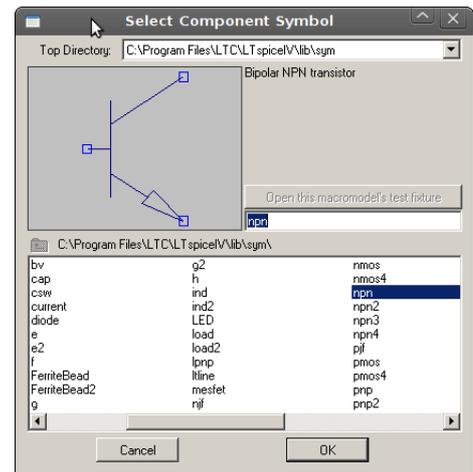


Illustration 3: Recherche d'un composant dans la librairie LTspice.

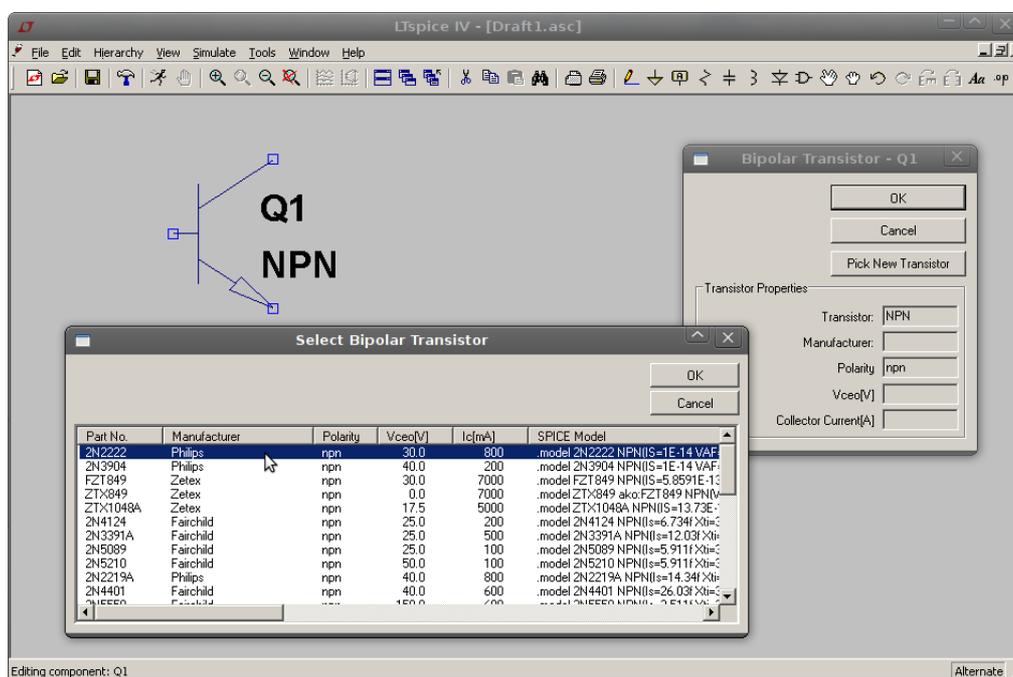


Illustration 4: Choisir un composant dans la liste déjà connue de LTspice.

Cette technique fonctionne pour tous les types de transistors classiques, diodes, optocoupleurs, etc. Pour les amplificateurs opérationnels, cela se gâte car, mis à part ceux vendus par LT, aucun modèle n'est disponible par défaut... mais pas de panique, des solutions existent, nous les verrons plus tard !

5.2. Mais il ne figure pas dans cette liste. Que faire ?

Ce cas de figure se présente lorsque l'on souhaite utiliser un composant électronique dont le symbole est déjà dans la bibliothèque (ex. transistor bipolaire, MOSFET, etc.) mais pas son modèle SPICE.

En pointant vers le répertoire d'installation de LTspice, on constate que la librairie contenant tous les composants directement accessibles à l'utilisateur se trouvent dans le sous-dossier **lib**. À l'intérieur, trois répertoires sont visibles :

- le dossier **cmp**, comportant les composants de base accompagnés de leur modèles. Tous sont des fichiers textes :

standard.bjt	modèles de transistors bipolaires
standard.cap	modèles de condensateurs
standard.dio	modèles de diodes
standard.ind	modèles d'inductances
standard.jfet	modèles de transistors JFET
standard.mos	modèles de transistors MOSFET
standard.res	modèles de résistances

- le dossier **sub**, qui contient les sous-circuits (fichiers ***.sub**) dont sont constitués certains composants disponibles, ainsi que des modèles SPICE (fichiers ***.mod** et / ou ***.lib**)
- et le dossier **sym** qui renferme l'ensemble des symboles (fichiers ***.asy**) présents dans la librairie accessible à l'utilisateur par l'intermédiaire du bouton d'accès correspondant (voir dans la partie 4).

Pour utiliser un composant dont le modèle ne figure pas dans la liste existante MAIS dont le symbole existe déjà, il faut procéder en deux temps. D'abord, il faut récupérer le modèle SPICE de ce composant, ce qui peut être fait en effectuant une recherche sur internet. Attention à bien confronter le modèle récupéré aux données techniques d'un fabricant avant de l'utiliser afin de le valider !

Le modèle SPICE se présente normalement sous la forme d'un simple fichier texte contenant des valeurs numériques à appliquer aux différents paramètres constituant le modèle générique du composant. Pour le transistor bipolaire PNP BC640 par exemple, ce modèle est le suivant :

```
.MODEL BC640_1 PNP(
+   AF= 1.00E+00   BF= 1.85E+02   BR= 4.28E+00   CJC= 8.00E-11
+   CJE= 1.27E-10   CJS= 0.00E+00   EG= 1.11E+00   FC= 5.00E-01
+   IKF= 9.69E-01   IKR= 1.00E+00   IRB= 1.62E-01   IS= 4.60E-12
+   ISC= 9.45E-17   ISE= 1.99E-14   ITF= 4.53E-01   KF= 0.00E+00
+   MJC= 5.07E-01   MJE= 4.94E-01   MJS= 3.30E-01   NC= 2.00E+00
+   NE= 1.23E+00   NF= 1.19E+00   NR= 1.23E+00   PTF= 0.00E+00
+   RB= 4.84E+00   RBM= 1.00E+01   RC= 2.62E-01   RE= 1.00E-02
+   TF= 1.07E-09   TR= 0.00E+00   VAF= 1.44E+02   VAR= 1.97E+01
+   VJC= 3.62E-01   VJE= 5.43E-01   VJS= 7.50E-01   VTF= 9.99E+05
+   XCJC= 1.00E+00   XTB= 0.00E+00   XTF= 6.31E-01   XTI= 3.00E+00)
```

Pour rajouter ce composant à la bibliothèque de LTSpice, il suffit de rajouter le texte du modèle SPICE récupéré au fichier texte **standard.*** qui s'y rapporte. Par exemple, si je souhaite rajouter le transistor bipolaire BC640, je rajoute le bout de texte ci-dessus à la fin du fichier **standard.bjt**.

5.3. Et si je désire maintenant rajouter un circuit intégré inconnu de LTSpice

Le cas des circuits intégrés est plus complexe à traiter que les simples composants élémentaires dont le symbole existe déjà. On trouve bien une bonne partie des circuits commercialisés par *LT*, mais pour le reste, encore une fois, il faut se débrouiller.

Nous allons ici traiter l'exemple d'un amplificateur opérationnel différentiel un peu particulier, le *THS4505*, commercialisé par *Texas Instrument*. La première partie de la procédure consiste, ici encore, à récupérer le modèle SPICE du composant, auprès du fabricant si possible. Une fois le modèle SPICE en possession (un fichier texte encore), deux solutions possibles :

- on crée une librairie propre que l'on vient rajouter à l'intérieur du dossier **sym**,
- on place le tout dans le répertoire de travail, là où sera sauvegardé le circuit électrique à simuler.

5.3.1. Créer sa propre librairie

Pour créer sa propre librairie, deux nouveaux dossiers sont à créer, à l'intérieur des répertoires **sym** et **sub**. Dans notre cas, cela donne : `../LtspiceIV/lib/sym/TI` et `../LtspiceIV/lib/sub/TI`. Le premier recevra le symbole, qu'il va falloir créer, tandis que le second accueillera le fichier contenant le modèle SPICE du composant. Ces deux fichiers doivent avoir le même nom, par exemple **THS4505.asy** (symbole) et **THS4505.lib** (modèle SPICE).

Pour créer le symbole, nous avons le choix entre partir d'un symbole existant que l'on recopie pour le modifier, ou bien démarrer d'une feuille vierge avec la commande **File** → **New symbol**. Pour ce faire, nous avons à notre disposition une palette d'outils accessibles depuis le menu **Draw** (cf. capture d'écran ci-dessous). Attention à nommer les pattes de connexions du composant conformément au fichier du modèle SPICE associé ! C'est le champ **Netlist Order** qui importe ici. Le champ **Label** n'est pas relié au fichier modèle SPICE.

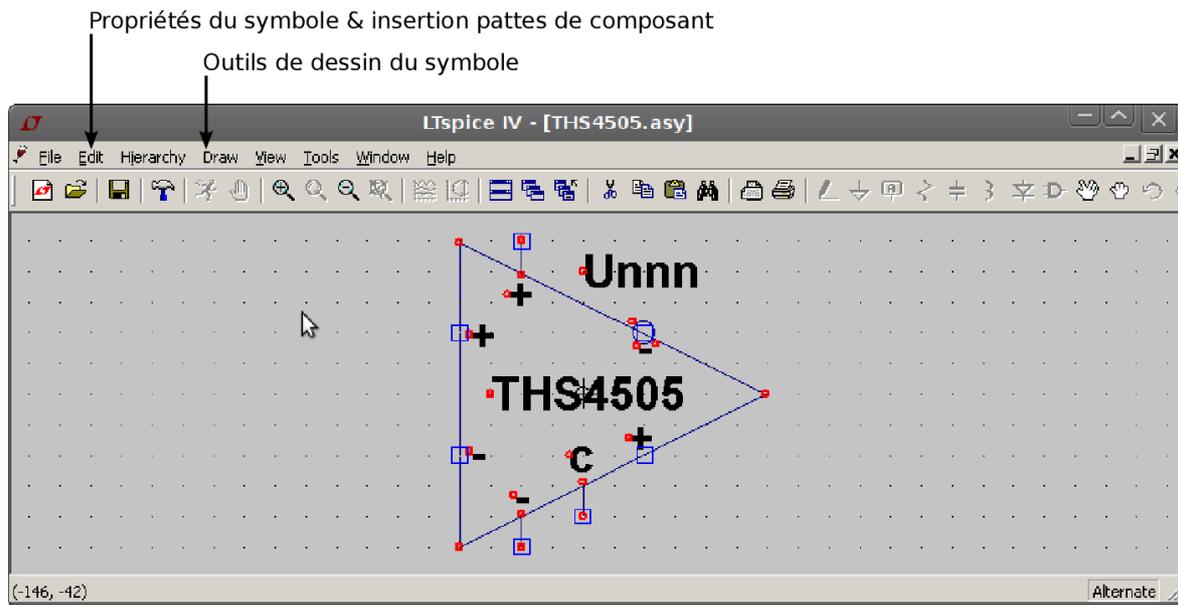


Illustration 5: outils de manipulation d'un symbole.

Une fois le symbole créé, il faut faire comprendre à LTspice qu'il est lié au fichier modèle du même nom. Pour ce faire, aller dans **Edit** → **Attributes** → **Edit Attributes**.

Nous devons maintenant remplir les différents attributs avec attention car ceux-ci conditionnent la détection correcte du modèle par LTspice.

Si le symbole représente une page d'un schéma hiérarchisé, le champ **Symbol Type** devra être positionné sur **Block** avec tous les autres attributs laissés vides. Dans le cas qui nous concerne, ce champ doit être positionné sur **Cell**.

Le champ **prefix** correspond à la dénomination qu'aura le symbole une fois placé dans la feuille contenant le circuit à simuler, à quelques exceptions près, comme ici où le champ doit être renseigné par un **X**, pour signifier à LTspice que ce symbole est associé à un sous-circuit décrit par une librairie.

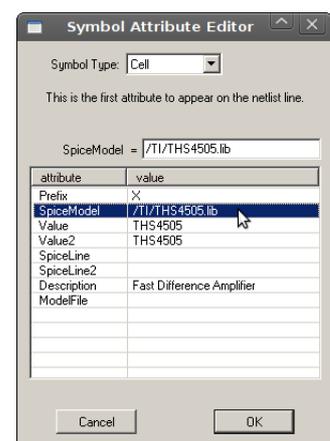


Illustration 6: Propriétés d'un symbole.

Le champ **SpiceModel** pointe vers la librairie. Les champs **Value** et **Value2** doivent être identiques à la directive SPICE présente derrière le **.model** ou bien **.subckt** du fichier modèle.

Enfin, pour plus de renseignement concernant le remplissage des différents champs de cette boîte

de dialogue, la meilleure source d'information sera indiscutablement le manuel d'utilisation, téléchargeable avec Ltspice.

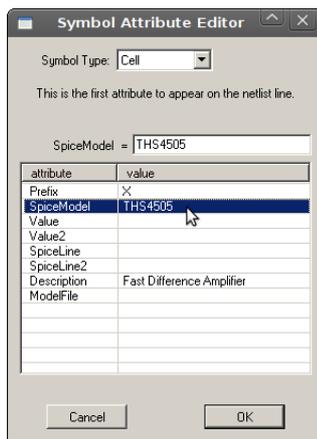
Le préfixe et la référence du composant peuvent être affichés à côté du symbole en récupérant les zones de texte correspondantes depuis **Edit** → **Attributes** → **Attribute window**.

Une fois toutes ces opérations effectuées, le THS4505 est reconnu dans la liste des composants accessibles par LTspice.

5.3.2. Rajouter manuellement le modèle à la simulation

Cette seconde méthode est plus simple et surtout ne nécessite pas de modifications dans le répertoire d'installation du logiciel.

Pour cela, il suffit de créer un symbole et de le placer, avec son fichier modèle, dans le répertoire de travail. Les deux fichiers doivent toujours posséder le même nom, mais les attributs symbole sont plus simples à renseigner (cf. capture d'écran ci-contre).



Seul impératif, rajouter la directive SPICE suivante sur la feuille de dessin :

```
.include THS4505.lib
```

Cette directive renvoie vers le fichier contenant le modèle THS4505 renseigné au niveau du **Symbol Attribute Editor** (cf. capture d'écran ci-contre). Le fichier en question est à placer à l'intérieur du dossier contenant votre projet.

Illustration 7: propriétés d'un symbole.

6. Effectuer une simulation

La première chose à faire est évidemment de savoir quoi simuler en plaçant la bonne source dans le schéma électrique. Ne pas oublier à mettre une masse quelque part dans le circuit. Celle-ci doit s'appeler 0 (= "zero"), qui est le nom par défaut du symbole lorsqu'il est placé sur le schéma et si celui-ci n'est pas modifié par la suite avec un label.

Si ce n'est pas fait, le simulateur renverra un message d'erreur en guise de résultat de simulation. Les boutons disponibles et relatifs à la simulation sont définis ci-dessous.

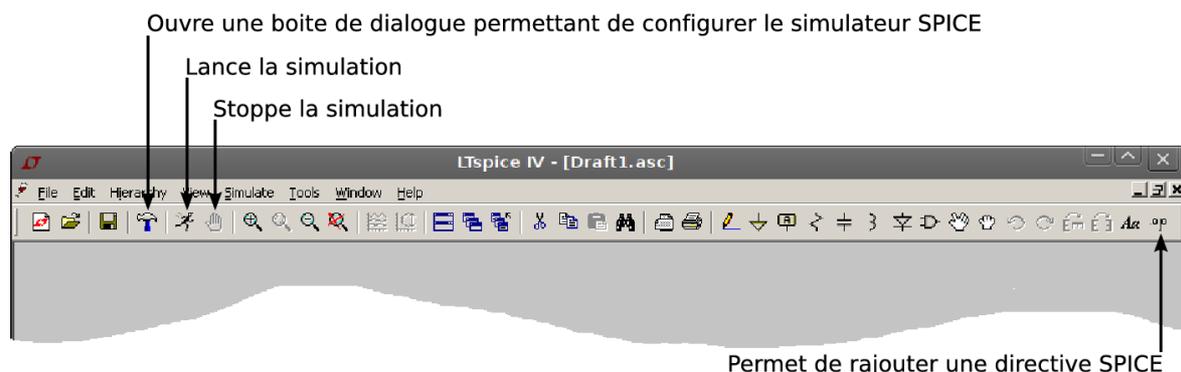


Illustration 8: boutons relatifs à la simulation d'un circuit.

6.1. Définir la source

Les sources sont définies dans la boîte de dialogue donnant accès aux différentes bibliothèques : **Edit** → **Component**. LTspice est doté de plusieurs types de sources :

- certaines permettant de construire des macro-modèles linéaires à partir de sources contrôlées (sources comportementales permettant de définir une fonction de transfert : bi, bi2, bv)
- des sources (e, e2, f, g, g2, h) et des commutateurs contrôlés (csw, sw),
- des sources basiques nommées **current** et **voltage**.

Ce sont ces deux dernières sources qui permettent de générer le signal à utiliser dans les simulations. Elles peuvent être définies comme continue, sinusoïdale, exponentielles, rectangulaires, etc. (cf. capture d'écran ci-dessous).

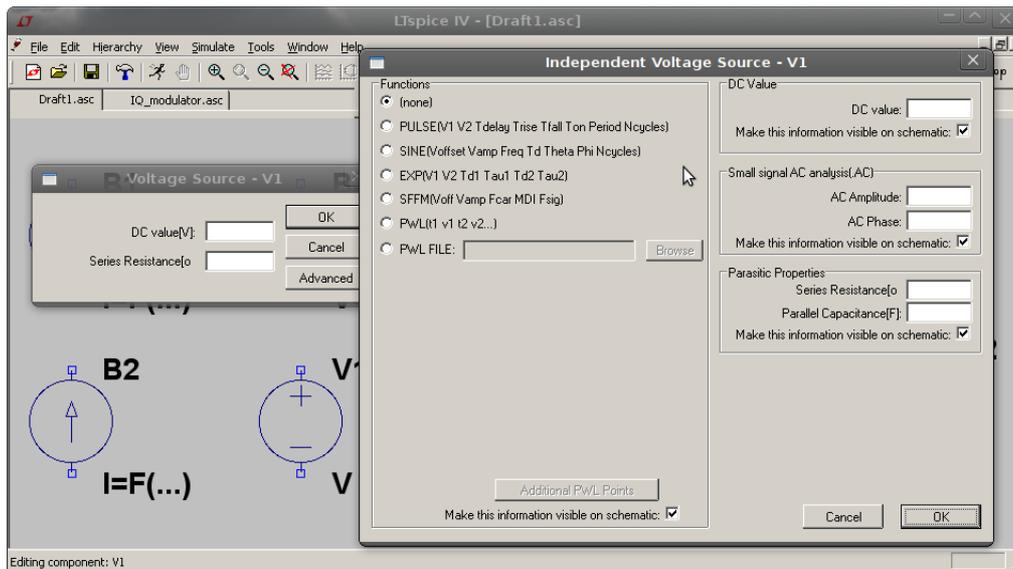


Illustration 9: définition des caractéristique d'une source.

En fonction du type de simulation à effectuer, différents champs sont à spécifier. Pour une simulation de type AC. Par exemple, les champs de la zone **Small signal AC analysis** doivent être complétés. Pour une simulation transitoire, c'est la zone **Functions** qui doit être renseignée.

6.2. Définir le type de simulation

Si aucune simulation n'est définie, une pression sur le bouton de lancement de la simulation (le bonhomme qui galope) ouvre une boîte de dialogue qui propose alors de choisir parmi les différents types de simulation accessibles (cf. ci-contre).

En renseignant les différents champs, la directive SPICE associée et visible en bas de la boîte de dialogue se remplit. Une fois la simulation configurée, cette directive est à coller sur la feuille de travail. La simulation peut être lancée.

À tout moment, cette directive, présente sur le schéma peut être éditée par un clic droit de la souris.

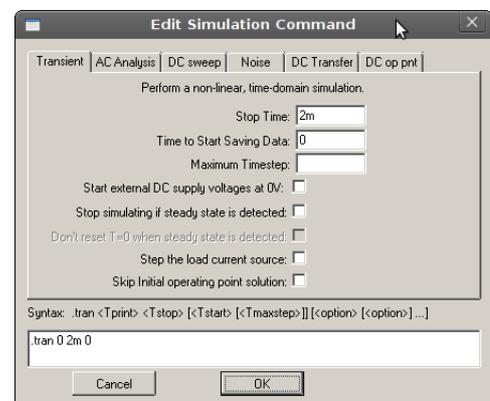


Illustration 10: Paramétrage d'une simulation.

6.3. Pour les problèmes de convergence

... Il existe un moyen de configurer le simulateur, avec les tolérances, les méthodes d'intégration, le solver, etc. Ce panneau de configuration est accessible depuis **Simulate** → **Control Panel** ou bien directement par le bouton correspondant en forme de marteau (cf. capture visible plus haut).

6.4. Effectuer une simulation en balayant une variable entre deux extrêmes

Cette manipulation nécessite d'introduire quelques directives SPICE sur la feuille contenant le circuit à étudier. Admettons à titre d'exemple que nous souhaitons faire varier la valeur d'une résistance entre deux valeurs extrêmes à définir plus tard.

La première chose à faire consiste à remplacer la valeur numérique de la résistance par un nom de variable que l'on met entre accolades, ex : {res}

En rajoutant la directive suivante : `.param res=1k`, le système se comporte exactement de la même façon que si nous avons directement placé 1k comme valeur pour cette résistance.

En plaçant maintenant la directive suivante : `.step param res list 500 1k 2k 3k 5k`, le système effectue la simulation demandée pour chaque valeur de résistance définie dans la directive.

Beaucoup de possibilités existent dès lors que l'on connaît la syntaxe SPICE. On renverra l'utilisateur désireux d'approfondir ses connaissances vers le manuel d'utilisateur disponible en téléchargement sur le site web de **LT**. À titre d'exemple, voici quelques types d'analyses paramétriques qu'il est possible de faire :

```
.step oct v1 1 20 5      le générateur v1 varie de 1 à 20 volts avec 5 points / octave,  
.step il 10μ 100μ 10μ   la valeur du générateur i1 de 10μA à 100μA par pas de 10μA,  
.step NPN 2N222(VAF) 50 100 25 le paramètre VAF du modèle de transistor NPN est changé de  
                        50 à 100 par pas de 25,  
.step temp -55 125 10  la température est modifiée dans la gamme -55°C à +125°C par pas  
                        de 10°C.
```

6.5. Analyses Monte Carlo et pire cas

Il existe plusieurs façons pour effectuer une analyse de type Monte Carlo sur un circuit.

6.5.1. Analyse Monte-Carlo en utilisant la fonction *mc(x,y)* intégrée au logiciel

Ltspice possède une fonction *mc(x,y)* qui génère un nombre aléatoire à l'intérieur de la plage de valeurs $x(1 \pm y)$ avec une distribution uniforme.

- Pour effectuer une analyse Monte-Carlo, la première étape consiste à remplacer la valeur nominale « *nom* » dans le champ correspondant du ou des composants concernés par l'analyse, en spécifiant la tolérance « *tol* » :

```
{mc(nom,tol)}  
sans oublier de définir la tolérance : .param tol=0.05
```

- Pour lancer plusieurs simulations, on procède de la façon suivante : `.step param run 1 10 1`

Cette commande définit un paramètre nommé « *run* » qui n'est utilisé que pour compter le nombre de simulations effectuées, c'est-à-dire 10 simulations puisque *run* démarre avec la valeur 1 jusqu'à 10 par pas de 1. Côté affichage, les dix simulations apparaissent simultanément.

6.5.2. Analyse pire-cas à partir d'une fonction créée par l'utilisateur

Une analyse pire-cas propose d'utiliser trois valeurs possibles pour un composant dont on souhaite évaluer les effets : la valeur nominale « *nom* » et les deux extrêmes $nom(1 \pm tol)$

Pour effectuer cette analyse, nous allons définir nous-mêmes une fonction $wc(nom,tol)$, wc pour Worst-Case. La syntaxe est la suivante :

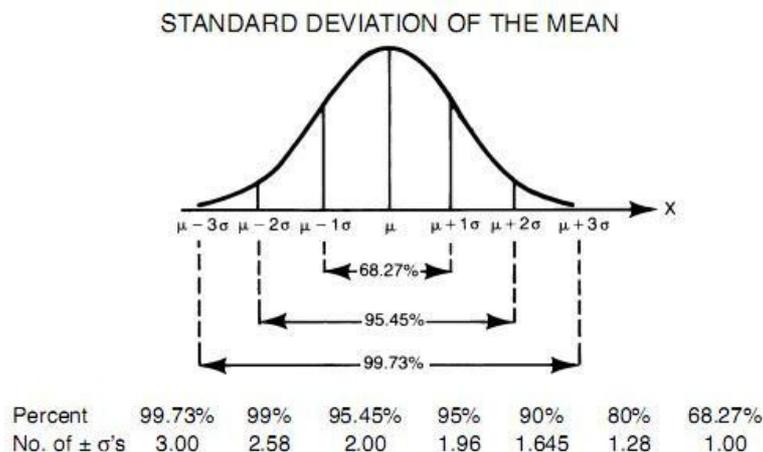
```
.function wc(tol,nom) if(run==1, nom, if(flat(1)>0,nom*(1+tol),nom*(1-tol)))
```

Avec cette fonction, la première simulation effectuée utilisera les valeurs nominales du composant. Les simulations suivantes utilisent soit la valeur maximale soit la valeur minimale, et ce de manière aléatoire. Une fois cette fonction définie, son utilisation est identique au cas présenté dans le paragraphe précédent.

6.5.3. Définir une analyse Monte-Carlo à partir d'une distribution gaussienne de valeurs

La fonction mc ne propose qu'une distribution uniforme de valeurs à l'intérieur d'une plage définie. Il est possible de créer soit-même la fonction à utiliser et ainsi de proposer une distribution gaussienne (loi normale).

Il est habituel en micro-électronique de proposer des tirages aléatoires contenus dans l'intervalle $\pm 3\sigma$ de la loi de distribution, qui prends en compte 99,73 % des événements possibles. Un petit rappel avec le graphe ci-dessous :



Il est possible de remplacer la fonction $mc(nom,tol)$ pas la fonction suivante, appelée « $test1$ » qui propose de distribuer les échantillons dans la plage $\pm 3\sigma$:

```
.function test1(tol,nom) nom*(1+gauss(tol/3))
```

Et pour faire la même chose sur la plage $\pm 4\sigma$:

```
.function test2(tol,nom) nom*(1+gauss(tol/4))
```

Si l'on souhaite maintenant effectuer une simulation pire-cas à partir d'une distribution gaussienne, la fonction « $test3$ » ci-dessous peut être utilisée, pour une plage $\pm 2\sigma$:

```
.function test3(tol,nom) if(run==1, nom, nom*(1+gauss(tol/2)))
```

Pour plus de détails, voici un lien particulièrement intéressant concernant les analyses pire-cas sous Ltspice : <http://k6jca.blogspot.fr/2012/07/monte-carlo-and-worst-case-circuit.html>

7. Affichage des résultats

Lorsqu'une simulation est lancée, la fenêtre de visualisation est automatiquement démarrée. Celle-ci reste toutefois vierge jusqu'à ce que l'utilisateur intervienne. Pour ajouter des graphes, il suffit de placer la souris au-dessus du schéma électrique. Son symbole change selon que la souris survole un fil ou un composant.

7.1. Ajouter enlever une courbe



Lorsque l'on survole un fil, un clic gauche permet d'afficher la tension correspondante sur la fenêtre de visualisation. En maintenant enfoncée le bouton gauche de la souris depuis le premier fil jusqu'à un second plus loin, il devient possible d'afficher la différence de potentiel correspondante.



Lorsque l'on survole un dipôle, la souris se transforme en sonde de courant, qu'il est également possible d'afficher par un clic gauche de la souris.

Pour retirer des graphes, plusieurs méthodes existent. La première consiste à cliquer au bouton droit de la souris sur l'étiquette de la courbe à effacer. Une fenêtre s'ouvre. Elle permet de modifier les attributs du graphe (couleur, type de trait, etc.) et de l'effacer.

Une seconde méthode consiste, au niveau du schéma, à effectuer un double-clic gauche de la souris sur le signal à afficher. On constate alors que tous les autres signaux affichés sont retiré au profil de ce dernier.

7.2. Gestion des curseurs

La gestion des curseurs s'effectue en cliquant sur les étiquettes des courbes avec le bouton gauche de la souris. Le curseur peut être déplacé à la souris, en attrapant le point fixé et en le déplaçant sans relâcher le bouton gauche de la souris.

7.3. Effectuer des opérations mathématiques sur les résultats

Il est possible de créer un nouveau graphe contenant des équations dont les variables correspondent aux courants et tensions du circuit relevés par le simulateur. Voici en résumé les différentes fonctions disponibles :

Fonction	Description	Fonction	Description
$abs(x)$	Valeur absolue de x	$log10(x)$	Base 10 logarithm
$acos(x)$	Arc cosine of x	$max(x,y)$	The greater of x or y
$arccos(x)$	Synonym for $acos()$	$min(x,y)$	The smaller of x or y
$acosh(x)$	Arc hyperbolic cosine	$pow(x,y)$	$x^{**}y$
$asin(x)$	Arc sine	$pwr(x,y)$	$abs(x)^{**}y$
$arcsin(x)$	Synonym for $asin()$	$pwr(x,y)$	$sgn(x)*abs(x)^{**}y$
$asinh(x)$	Arc hyperbolic sine	$rand(x)$	Random number between 0 and 1 depending on the integer value of x .
$atan(x)$	Arc tangent of x	$random(x)$	Similar to $rand()$, but smoothly transitions between values.
$arctan(x)$	Synonym for $atan()$	$round(x)$	Nearest integer to x
$atan2(y, x)$	Four quadrant arc tangent of y/x	$sgn(x)$	Sign of x
$atanh(x)$	Arc hyperbolic tangent	$sin(x)$	Sine of x
$buf(x)$	1 if $x > .5$, else 0	$sinh(x)$	Hyperbolic sine of x
$ceil(x)$	Integer equal or greater than x	$sqrt(x)$	Square root of x
$cos(x)$	Cosine of x	$table(x,a,b,c,d,...x)$	Interpolate a value for x based on a look up table given as a set of pairs of points.
$cosh(x)$	Hyperbolic cosine of x	$tan(x)$	Tangent of x .
$d()$	Finite difference-based derivative	$tanh(x)$	Hyperbolic tangent of x
$exp(x)$	e to the x	$u(x)$	Unit step, i.e., 1 if $x > 0$, else 0.
$floor(x)$	Integer equal to or less than x	$uramp(x)$	x if $x > 0$, else 0.
$hypot(x,y)$	$sqrt(x^{**}2 + y^{**}2)$		
$if(x,y,z)$	If $x > .5$, then y else z		
$int(x)$	Convert x to integer		
$inv(x)$	0. if $x > .5$, else 1.		

<i>limit(x,y,z)</i>	<i>Intermediate value of x, y, and z</i>	<i>white(x)</i>	<i>Random number between -0.5 and 0.5 smoothly transitions between values even more smoothly than random().</i>
<i>ln(x)</i>	<i>Natural logarithm of x</i>		
<i>log(x)</i>	<i>Alternate syntax for ln()</i>		

Les fonctions suivantes ne sont pas définies si les données sont complexes : *atan2()*, *sgn()*, *u()*, *buf()*, *inv()*, *uramp()*, *int()*, *floor()*, *ceil()*, *rand()*, *min()*, *limit()*, *if()* et *table(...)*.

Au contraire, les fonctions suivantes sont définies dans le cas où des nombres complexes sont traités : *Re()*, *Im()*, *Ph()* and *Mag()* et *conj()*. Ces fonctions représentent respectivement la partie réelle, imaginaire, la phase, le module et le conjugué de la variable complexe située entre parenthèses.

Les opérandes suivantes sont accessibles :

<i>Opérande</i>	<i>Description</i>
&	<i>Convert the expressions to either side to Boolean, then AND.</i>
 	<i>Convert the expressions to either side to Boolean, then OR.</i>
^	<i>Convert the expressions to either side to Boolean, then XOR.</i>
>	<i>TRUE if expression on the left is greater than the expression on the right, otherwise FALSE.</i>
<	<i>TRUE if expression on the left is less than the expression on the right, otherwise FALSE.</i>
>=	<i>TRUE if expression on the left is less than or equal the expression on the right, otherwise FALSE.</i>
<=	<i>TRUE if expression on the left is greater than or equal the expression on the right, otherwise FALSE.</i>
+	<i>Addition</i>
-	<i>Subtraction</i>
*	<i>Multiplication</i>
/	<i>Division</i>
**	<i>Raise left hand side to power of right hand side.</i>
!	<i>Convert the following expression to Boolean and invert.</i>
@	<i>Step selection operator</i>

TRUE est numériquement égal à 1 (0 pour FALSE). Lorsqu'une conversion vers un nombre booléen est effectuée, la valeur 1 est obtenue pour tout nombre supérieur à 0,5. On obtient 0 le cas échéant.

L'opérateur **@** est utile lorsque plusieurs simulations sont disponibles tel que lorsque les directives SPICE **.step**, **.temp**, ou **.dc** sont utilisées. Par exemple, **V(1)@3** concerne les données résultant de la troisième simulation effectuée.

Les constantes suivantes sont connues de LTspice :

E = 2.7182818284590452354, **Pi** = 3.14159265358979323846 , **K** = 1.3806503e-23 et **Q** = 1.602176462e-19

7.4. Quelques autres options disponibles

Plusieurs autres options d'affichage sont accessibles depuis la fenêtre d'affichage à partir du menu déroulant qui apparaît avec un clic droit de la souris comme par exemple :

- Rajouter des courbes sur un graphe distinct
- Afficher un diagramme de l'oeil
- montrer le spectre d'un signal par calcul de FFT

Les loupes servent également à zoomer sur certaines zones de l'affichage.

CTRL + clic gauche sur une étiquette permet de visualiser la valeur moyenne et efficace du signal (*.transient*).

7.5. Sauvegarder une configuration d'affichage

Il est possible de conserver des affichages complexes mettant en oeuvre plusieurs axes, des expressions mathématiques, etc. Il suffit pour cela de se diriger vers le menu **Plot settings** → **Save plot settings**. LTspice écrit alors un fichier de type **.plt* qui sera utilisé par le simulateur aux prochaines ouvertures du projet auquel il se rapporte (en lui donnant le même nom que la feuille contenant le schéma).