

Notice d'utilisation simplifiée de LTspice



Christophe Viallon, 2010 - 2020
Université Paul Sabatier, Toulouse III

Table des matières

1. Introduction.....	3
2. Présentation.....	3
3. Créer un nouveau projet.....	4
4. Placer un composant électronique sur la feuille de travail.....	4
4.1. Composants passifs.....	4
4.2. Accès aux bibliothèques contenant d'autres composants et circuits.....	5
4.3. Je ne trouve pas le composant que je souhaite utiliser.....	6
4.4. Placer le modèle directement sur la feuille de travail.....	6
4.5. ... ou bien l'intégrer dans le dossier d'installation de LTspice.....	7
4.6. Créer un symbole à associer à un modèle.....	7
4.7. Le modèle et son symbole sont placés dans le dossier d'installation de LTspice.....	9
4.8. Le modèle et son symbole sont placés dans le dossier de travail.....	9
5. Effectuer une simulation.....	9
5.1. Définir la source.....	10
5.2. Définir le type de simulation.....	11
5.3. Pour les problèmes de convergence.....	11
5.4. Effectuer des simulations paramétriques.....	11
5.5. Analyses Monte Carlo et pire cas.....	12
5.5.1. Analyse Monte-Carlo en utilisant la fonction mc(x,y) intégrée au logiciel.....	12
5.5.2. Analyse pire-cas à partir d'une fonction personnalisée avec « .func ».....	12
5.5.3. Définir une analyse Monte-Carlo à partir d'une distribution gaussienne de valeurs... ..	12
6. Affichage des résultats.....	13
6.1. Ajouter et enlever une courbe.....	13
6.2. Tracer une grandeur en fonction d'une variable définie dans un « .step ».....	14
6.3. Observer le résultat d'une expression mathématique à partir d'un « .meas ».....	14
6.3.1. Evaluer une équation.....	14
6.4. Gestion des curseurs.....	14
6.5. Quelques autres options disponibles.....	14
6.6. Sauvegarder une configuration d'affichage.....	15
7. Fonctions, opérands et constantes reconnues dans LTspice.....	16

1. Introduction

LTspice¹ est un logiciel de simulation électrique de la société Linear Technology aujourd'hui absorbée par Analog Devices, 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 io dédié à LTspice, qui regroupe de nombreuses astuces d'utilisation du logiciel avec une documentation impressionnante : <https://groups.io/g/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.

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.).

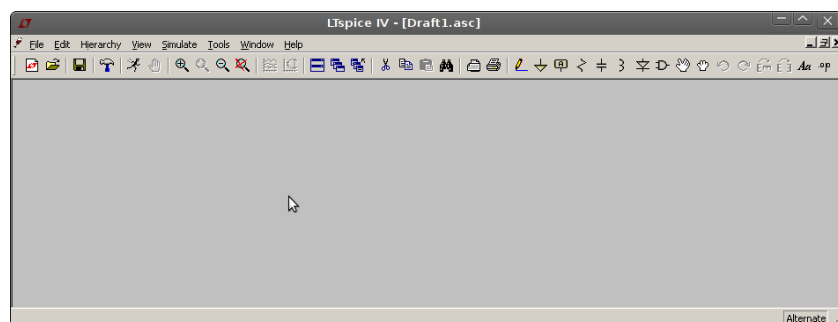


Figure 1: LTspice une fois lancé.

1 Téléchargeable gratuitement ici : <https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html>

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. Placer un composant électronique sur la feuille de travail

Dans LTspice, et dans pratiquement tous les simulateurs circuits, un composant électronique est constitué de deux éléments : le symbole (un dessin avec des pattes), et son modèle mathématique sous-jacent. Évidemment, le modèle mathématique s'attache aux pattes définies au niveau du symbole (attention aux bêtises avec les modèles perso !).

4.1. Composants passifs

Les composants passifs ont un accès direct depuis le plan de travail à partir des boutons correspondants, comme illustré ci-dessous. Pour gagner du temps et connaître les raccourcis clavier correspondants, il suffit de faire dérouler le menu **Edit**.

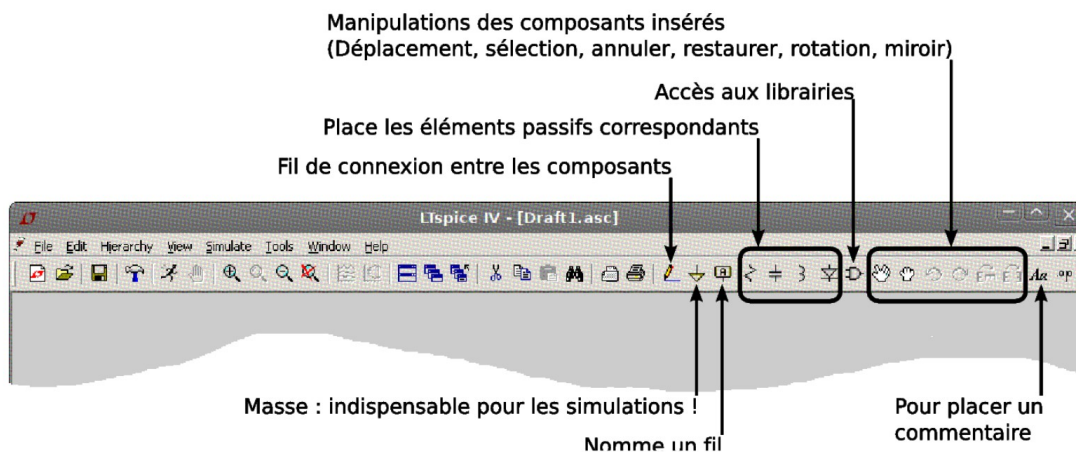


Figure 2: boutons relatifs à la manipulation de composants.

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. Ceci n'est valable que pour les composants passifs.

Pour donner une valeur à ces composants, rappelons comment sont définies les unités sous SPICE :

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

4.2. Accès aux bibliothèques contenant d'autres composants et circuits

Pour tous les autres composants et circuits, il faut aller le chercher dans les bibliothèques de LTspice à l'aide du bouton correspondant, ce qui ouvre la fenêtre ci-contre :

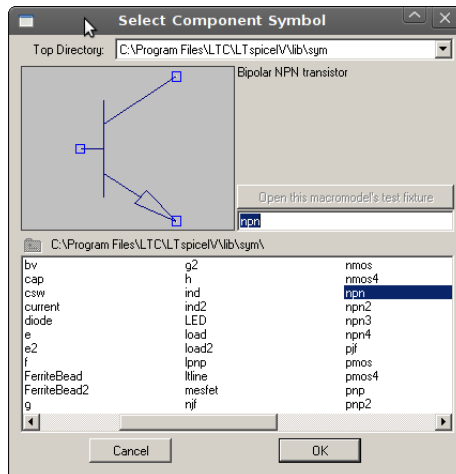


Figure 3: Recherche d'un composant dans la librairie LTspice.

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.

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 !

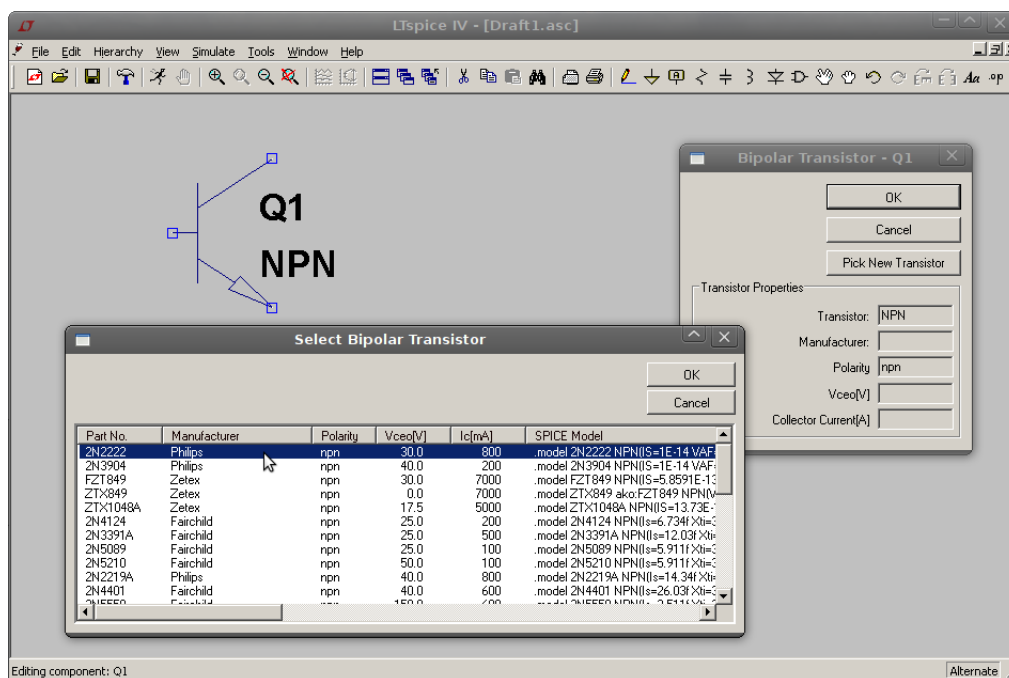


Figure 4: Choisir un composant dans la liste déjà connue de LTspice.

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

LTspice contient des modèles pour un grand nombre de circuits intégrés tels que régulateurs de tension linéaires ou à découpages, des AOp, etc.... ainsi que des diodes, transistors bipolaires, JFET et MOSFETs. Pour les circuits intégrés, seuls ceux commercialisés par LT sont intégrés au logiciel. Pour placer un circuit inconnu, il faut importer son modèle. Plusieurs solutions existent :

Si le composant correspond à un symbole standard déjà présent dans la bibliothèque LTspice :

- placer son modèle directement sur la feuille de travail, aller au §4.4. À utiliser pour déclarer des composants simples comme une diode ou un transistor dont le symbole est déjà disponible.
- placer son modèle dans un des sous-dossiers d'installation de LTspice (cf. §4.5).

Si le composant à intégrer ne dispose pas de symbole standard, ce qui est généralement le cas des circuits intégrés,

- il faut le créer (cf. §4.6) et l'associer au fichier modèle récupéré, ou bien que l'on a créé soi-même,
- placer l'ensemble dans les dossiers d'installation de LTspice (cf. §4.7).
- ou bien directement dans le dossier de travail (cf. §4.8),

4.4. Placer le modèle directement sur la feuille de travail

Avec cette méthode, le modèle SPICE du composant à déclarer se fait à l'aide du bouton **.op**. A titre d'exemple, si je souhaite utiliser une diode shottky de type HSMS285X (Avago), je place le modèle ci-dessous dans la boîte ouverte avec le **.op** :

```
.model HSMS_285X D(IS=3E-6, CJO=0.18E-12, VJ=.35, BV=3.8, IBV=3E-4 + EG=0.69, N=1.06, RS=25, XTI=2, M=0.5)
```

Le modèle `HSMS_285X` est ensuite lié au symbole en récupérant le composant **shottky** depuis le bouton donnant accès aux différents modèles (voir figure 3). Un clic droit sur la valeur de la diode ouvre la fenêtre ci-dessous, et dans laquelle on place `HSMS_285X`.

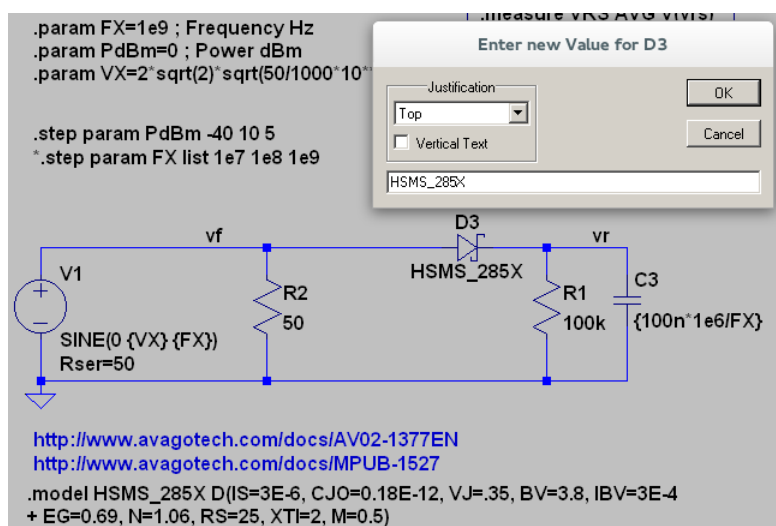


Figure 1: association du modèle HSMS_285X à une diode de la librairie LTspice

4.5. ... ou bien l'intégrer dans le dossier d'installation de LTspice

Toujours dans la même situation, c'est-à-dire avec un modèle dont le symbole est déjà présent à l'installation de LTspice, il est possible d'ajouter le modèle à la liste déjà existante de composants. Cette méthode possède un avantage mais aussi un inconvénient. L'avantage, c'est que cette opération va rendre le composant disponible pour tous les futurs projets, au même titre que les composants standards déjà fournis par LT. Cependant, une mise à jour du logiciel est susceptible d'effacer le composant ce qui fait que je ne conseille pas trop cette méthode.

En pointant vers le répertoire d'installation de LTspice, toutes les bibliothèques sont contenues dans le dossier *./LtpiceIV/lib*. A l'intérieur de ce dossier, le sous-dossier *./LtpiceIV/lib/cmp* comporte les composants de base accompagnés de leur modèles. Tous sont des fichiers textes :

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

Une fois récupéré, le modèle SPICE se présente normalement sous la forme d'un simple fichier texte contenant des valeurs numériques pour chacun des paramètres du modèle mathématique associé. Pour le transistor bipolaire BC640 par exemple, ce modèle mathématique est celui du PNP, et les valeurs numériques correspondant au composant sont les suivantes :

```
.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 ci-dessus au fichier texte **standard.*** correspondant au composant. 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**.

4.6. Créer un symbole à associer à un modèle

Les symboles des composants standards sont disponibles dans le sous-dossier *./LtpiceIV/lib/sym/*. 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 construire ce symbole, 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.

Nous allons ici traiter l'exemple d'un amplificateur opérationnel différentiel un peu particulier, le *THS4505*, commercialisé par *Texas Instrument*. TI fournit un fichier modèle que l'on peut télécharger, mais il faut créer le symbole sois-même.

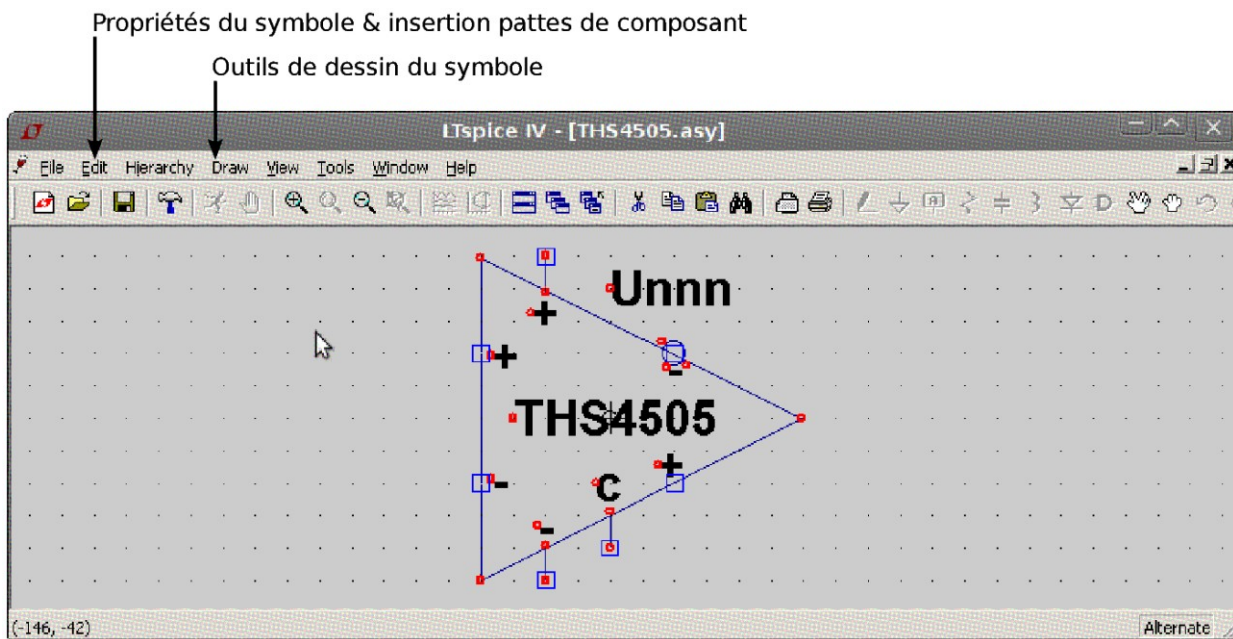


Figure 2: outils de manipulation d'un symbole

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.

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.

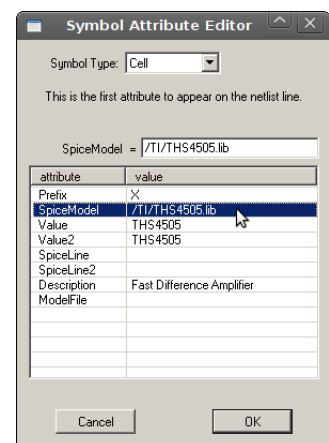


Figure 3: propriétés du symbole

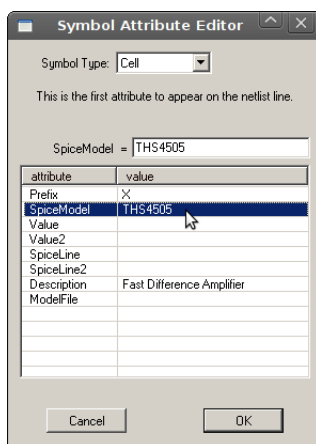
4.7. Le modèle et son symbole sont placés dans le dossier d'installation de LTspice

Dans ce cas, deux nouveaux dossiers sont à créer à l'intérieur des répertoires `./LtspiceIV/lib/sym/` et `./LtspiceIV/lib/sub/`. Nous avons choisi de créer un dossier TI, ce qui donne : `./LtspiceIV/lib/sym/TI` et `./LtspiceIV/lib/sub/TI`. Le premier recevra le symbole que nous venons de 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).

Une fois l'opération terminée, le nouveau composant est accessible au même titre que les modèles fournis par LT.

4.8. Le modèle et son symbole sont placés dans le dossier de travail

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 comme précédemment 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.

Figure 4: propriété du symbole

Avec cette méthode, le composant peut être récupéré toujours en cliquant sur la boîte **component**, mais cette fois-ci, il faut changer le dossier vers lequel LTspice pointe pour chercher les modèles (voir **top directory**).

5. 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 de 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.

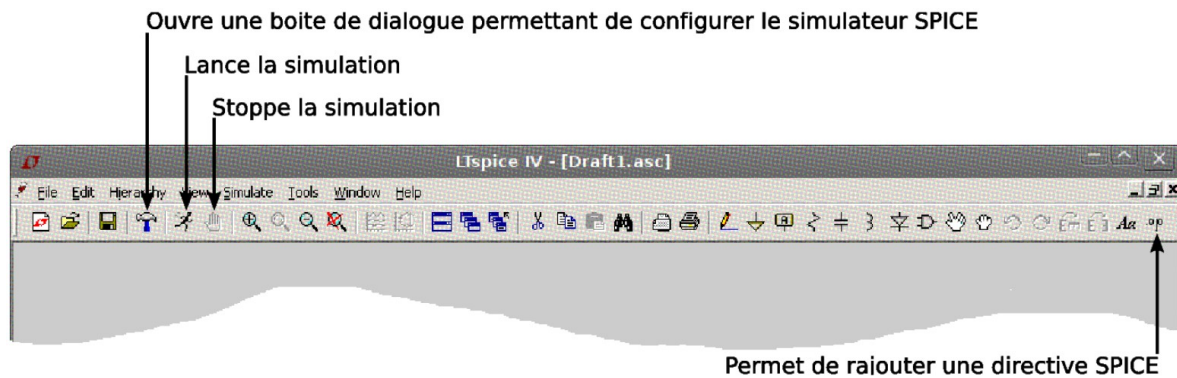


Figure 5: boutons relatifs à la simulation d'un circuit

5.1. 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).

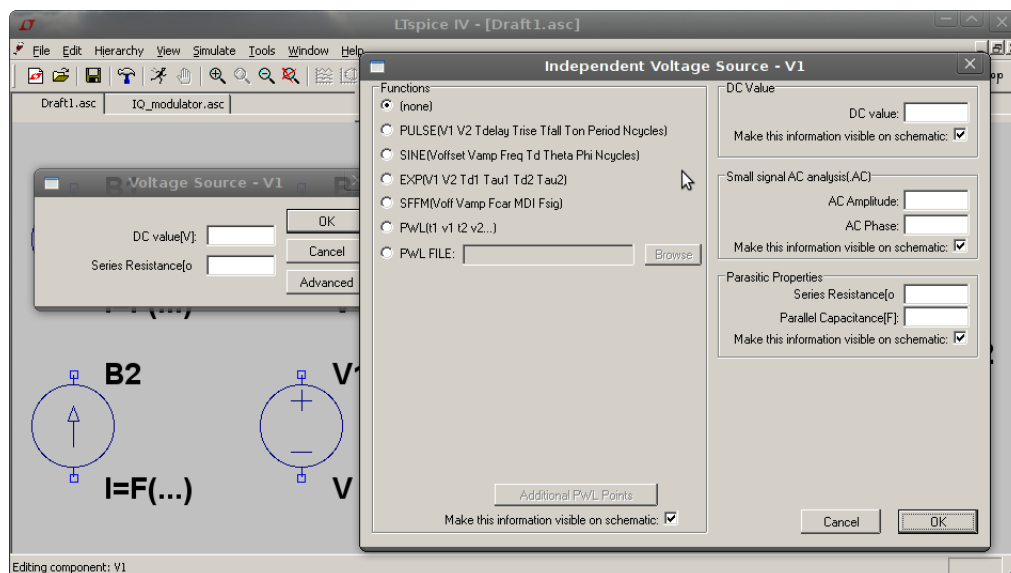


Figure 6: définition des caractéristique d'une source

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.

5.2. 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.

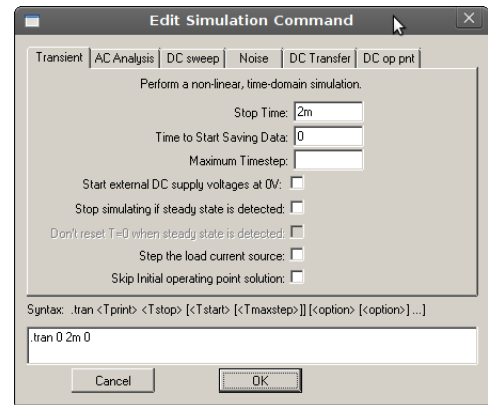


Figure 7: Paramétrage d'une simulation

5.3. 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).

5.4. Effectuer des simulations paramétriques

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 `.param res=1k`, LTspice va se comporter exactement comme si la valeur 1k avait été saisie dans la résistance.

En plaçant maintenant la directive suivante : `.step param res list 500 1k 2k 3k 5k`, LTspice va effectuer 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 i1 10u 100u 10u    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.
```

5.5. Analyses Monte Carlo et pire cas

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

5.5.1. 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 itérations, on procède de la façon suivante : `.step param run 1 10 1`

Cette directive 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. Coté affichage, les dix simulations apparaissent simultanément.

5.5.2. Analyse pire-cas à partir d'une fonction personnalisée avec « *.func* »

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 (on peut utiliser indistinctement `.function` ou `.func`) :

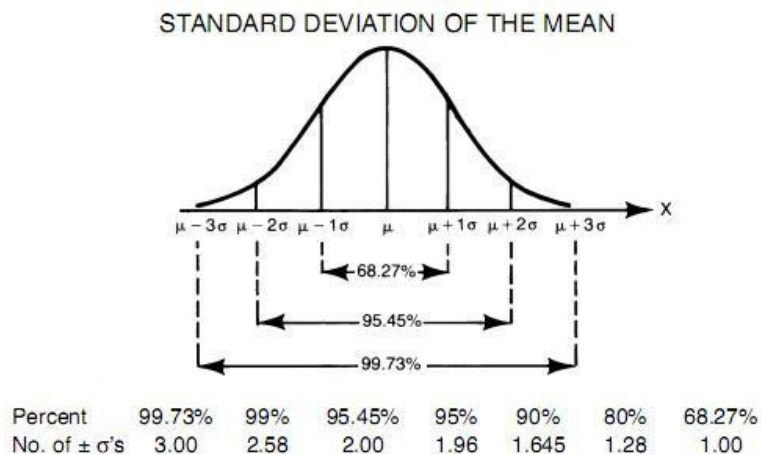
```
.func wc(tol,nom) if(run==1, nom, if(flat(1)>0,nom*(1+tol),nom*(1-tol)))
```

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.

5.5.3. 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)` par 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)))
```

Toutes ces fonctions peuvent être introduites dans le champs « valeur » des composants sur lesquels on souhaite faire les analyses évoquées dans cette partie (cf. §5.5.1).

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>

6. Affichage des résultats

Lorsqu'une simulation est achevée, 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 (tension) ou un composant (courant).

6.1. Ajouter et enlever une courbe



Lorsque l'on survole un fil avec la souris, 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.

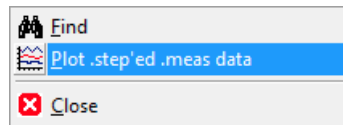
Dans tous les cas, les grandeurs du circuit sont tracées en fonction du paramètre que le simulateur fait varier, c'est-à-dire :

- une tension ou un courant statique dans une simulation de type `.dc`
- le temps pour une simulation de type `.tran`
- la fréquence avec une simulation de type `.ac`

Nous avons déjà vu en §5.4 qu'il est possible d'ajouter d'autres grandeurs à paramétrer à l'aide d'un `.step`. Mais dans ce cas, il serait intéressant de pouvoir afficher des résultats en fonction de cette variable paramétrée. La solution se trouve dans le paragraphe suivant.

6.2. Tracer une grandeur en fonction d'une variable définie dans un « .step »

Cette fonctionnalité est possible dès que la fonction `.step` est invoquée (cf. §5.4). Après avoir lancé et terminé la simulation, on ouvre le fichier log (CTRL+L) depuis la feuille schématique. Un clic droit de la souris ouvre la boîte suivante :



Et là, il faut choisir « `Plot .step'ed .meas data` ». La variable placée dans le `.step` sera utilisée pour l'axe des x, tandis que l'axe des y sera défini par la tension, le courant souhaité, ou bien par une fonction calculée à l'aide de la directive `.meas`. L'utilisation de cette directive est détaillée en suivant.

6.3. Observer le résultat d'une expression mathématique à partir d'un « .meas »

La directive `.meas` permet à l'utilisateur de définir lui-même ce qu'il souhaite observer. C'est une directive extrêmement puissante, mais également quelque peu complexe à utiliser puisque le format d'utilisation dépend du type de simulation effectuée (`.dc`, `.ac` ou `.tran`). Avec `.meas`, Il est possible de mesurer :

- un temps de montée, de descente, un délais,
- trouver une valeur sur une condition,
- évaluer une équation,
- calculer une valeur moyenne, RMS, min, max, et peak-to-peak,
- faire une intégration ou une dérivation,
- tracer une fonction d'erreur après une optimisation.

6.3.1. Evaluer une équation

Dans ce cas, la syntaxe est la suivante :

```
.meas <dc|tran|ac> result param='equation'+ <goal=value> <minval=value>
```

À compléter...

6.4. 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.

6.5. 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 par exemple :

- Rajouter des courbes sur un graphe distinct
- Afficher un diagramme de l'œil
- 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 (avec une simulation de type `.tran`).

6.6. Sauvegarder une configuration d'affichage

Il est possible de conserver des affichages complexes mettant en œuvre 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).

7. Fonctions, opérandes et constantes reconnues dans LTspice

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 :

<i>Fonction</i>	<i>Description</i>	<i>Fonction</i>	<i>Description</i>
<i>abs(x)</i>	Valeur absolue de x	<i>log10(x)</i>	Base 10 logarithm
<i>acos(x)</i>	Arc cosine of x	<i>mag()</i>	magnitude
<i>arccos(x)</i>	Synonym for <i>acos()</i>	<i>max(x,y)</i>	The greater of x or y
<i>acosh(x)</i>	Arc hyperbolic cosine	<i>min(x,y)</i>	The smaller of x or y
<i>asin(x)</i>	Arc sine	<i>ph()</i>	phase
<i>arcsin(x)</i>	Synonym for <i>asin()</i>	<i>pow(x,y)</i>	$x^{**}y$
<i>asinh(x)</i>	Arc hyperbolic sine	<i>pwr(x,y)</i>	$abs(x)^{**}y$
<i>atan(x)</i>	Arc tangent of x	<i>pwr(x,y)</i>	$sgn(x)*abs(x)^{**}y$
<i>arctan(x)</i>	Synonym for <i>atan()</i>	<i>rand(x)</i>	Random number between 0 and 1 depending on the integer value of x.
<i>atan2(y, x)</i>	Four quadrant arc tangent of y/x		
<i>atanh(x)</i>	Arc hyperbolic tangent	<i>random(x)</i>	Similar to <i>rand()</i> , but smoothly transitions between values.
<i>buf(x)</i>	1 if $x > .5$, else 0	<i>re()</i>	Real part
<i>ceil(x)</i>	Integer equal or greater than x	<i>round(x)</i>	Nearest integer to x
<i>cos(x)</i>	Cosine of x	<i>sgn(x)</i>	Sign of x
<i>cosh(x)</i>	Hyperbolic cosine of x	<i>sin(x)</i>	Sine of x
<i>d()</i>	Finite difference-based derivative	<i>sinh(x)</i>	Hyperbolic sine of x
<i>db()</i>	Magnitude in dB	<i>sqrt(x)</i>	Square root of x
<i>exp(x)</i>	e to the x	<i>table(x,a,b,c,d,...x)</i>	Interpolate a value for x based on a look up table given as a set of pairs of points.
<i>floor(x)</i>	Integer equal to or less than x		
<i>hypot(x,y)</i>	$sqrt(x^{**}2 + y^{**}2)$	<i>tan(x)</i>	Tangent of x.
<i>if(x,y,z)</i>	If $x > .5$, then y else z	<i>tanh(x)</i>	Hyperbolic tangent of x
<i>im()</i>	imaginary part	<i>u(x)</i>	Unit step, i.e., 1 if $x > 0.$, else 0.
<i>int(x)</i>	Convert x to integer	<i>uramp(x)</i>	x if $x > 0.$, else 0.
<i>inv(x)</i>	0. if $x > .5$, else 1.	<i>white(x)</i>	Random number between -0.5 and 0.5 smoothly transitions between values even more smoothly than <i>random()</i> .
<i>invdb(x)</i>	$10^{**(x/20)}$		
<i>limit(x,y,z)</i>	Intermediate value of x, y, and z		
<i>ln(x)</i>	Natural logarithm of x		
<i>log(x)</i>	Alternate syntax for <i>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 :

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

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 reconnues dans LTspice :

<i>Time</i>	<i>Variable</i>	<i>time in seconds (real data only)</i>
<i>freq</i>	<i>Variable</i>	<i>freq in Hertz (cmplx data only)</i>
<i>w</i>	<i>Variable</i>	<i>freq in radians (cmplx data only)</i>
<i>i</i>	<i>sqrt(-1)</i>	<i>imaginary unity (cmplx data only)</i>
<i>e</i>	2.71828	
<i>pi</i>	3.14159	
<i>c</i>	2.99792e+008	<i>speed of light in m/s</i>
<i>boltz</i>	1.38062e-023	<i>Boltzmann constant</i>
<i>planck</i>	6.62620e-034	<i>Planck's constant</i>
<i>echarge</i>	1.60219e-019	<i>charge of an electron</i>
<i>kelvin</i>	-2.73150e+002	<i>absolute zero in degrees C</i>