

# Le simulateur électrique standard SPICE

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Le fichier de simulation</b>	<b>3</b>
2.1	La structure . . . . .	3
2.2	La syntaxe et la grammaire . . . . .	3
2.3	Les composants élémentaires . . . . .	5
2.4	Les lignes de transmission . . . . .	5
2.5	Les composants actifs . . . . .	5
2.6	Les sources de tension et de courant indépendantes . . . . .	5
2.7	Les sources dépendantes linéaires . . . . .	5
2.8	Les sources dépendantes non-linéaires . . . . .	5
2.9	Les commandes d'analyse . . . . .	5
2.10	Les commandes de sortie . . . . .	6
2.11	Les directives de simulation . . . . .	6
2.12	Les structures de donnée . . . . .	6
<b>3</b>	<b>Les analyses basiques et le conditionnement des algorithmes correspondant</b>	<b>6</b>
3.1	L'analyse statique <b>.DC</b> . . . . .	6
3.1.1	Fonction . . . . .	6
3.1.2	Principe de calcul . . . . .	7
3.2	Le point de fonctionnement <b>.OP</b> . . . . .	8
3.2.1	Fonction . . . . .	8
3.2.2	Principes de calcul . . . . .	8
3.3	L'analyse transitoire <b>.TRAN</b> . . . . .	8
3.3.1	Fonction . . . . .	8
3.3.2	Principe de calcul . . . . .	8
3.4	L'analyse alternative petit signal <b>.AC</b> . . . . .	9
3.4.1	Fonction . . . . .	9
3.4.2	Principe de calcul . . . . .	9
3.5	La directive de simulation <b>.OPTION</b> . . . . .	10
3.5.1	Les paramètres de contrôle d'impression . . . . .	10
3.5.2	Les paramètres de contrôle des modèles . . . . .	10
3.5.3	Les paramètres de contrôle de la méthode de Newton-Raphson . . . . .	10
3.5.4	Les paramètres de contrôle spécifiques à <b>.DC</b> . . . . .	10
3.5.5	Les paramètres de contrôle spécifiques à <b>.TRAN</b> . . . . .	10
3.5.6	Les paramètres de contrôle de la méthode de d'élimination de Gauss . . . . .	11
<b>4</b>	<b>L'exécution du fichier de simulation</b>	<b>11</b>
4.1	Exécution en mode non-interactif (batch) : . . . . .	11
4.2	Exécution en mode interactif . . . . .	11

<b>5</b>	<b>Quelques exemples de fichiers de simulation SPICE</b>	<b>12</b>
5.1	Caractérisation d'un transistor NMOS . . . . .	12
5.1.1	Une première méthode . . . . .	12
5.1.2	Une deuxième méthode . . . . .	13
5.2	Analyses alternatives d'un filtre passe-bas d'ordre deux . . . . .	15
<b>6</b>	<b>Ce que fait pas ou fait mal SPICE3F5</b>	<b>16</b>

# 1 Introduction

SPICE (**S**imulation **P**rogram with **I**ntegrated **C**ircuit **E**mphasis) est un [simulateur électrique standard](#) qui permet l'analyse statique et transitoire des circuits non-linéaires et l'analyse alternative petit signal des circuits linéarisés. Les circuits peuvent contenir des résistances, des condensateurs, des inductances, des inductances mutuelles, des lignes de transmission, des lignes RC distribuées, des sources de tension et de courant commandées et indépendantes, des interrupteurs, ainsi que des modèles de dispositifs actifs : diodes, transistors bipolaires, transistors à effet de champ à jonction (JFET), transistors à arsénure de gallium (ASGA) à effet de champ métal-semiconducteur (MESFET) et des transistors à effet de champ métal-oxyde-semiconducteur (MOS). Le simulateur a été développé à l'université de Berkeley (<http://www.eecs.berkeley.edu>) à partir des années 1967 et jusqu'en 1990 : BIAS3 en 1967 pour l'analyse statique des transistors bipolaires puis CANCER ( **C**omputer **A**nalysis of **N**onlinear **C**ircuits, **E**xcluding **R**adiation en 1970, SPICE1 en 1973, SPICE2 en 1975 et SPICE3 en 1980. Il est actuellement figé, et sa dernière version SPICE3F5 est distribuée librement sur le réseau internet. C'est un simulateur électrique de deuxième génération fonctionnellement construit sur quatre méthodes numériques de base :

1. une méthode numérique de formulation des équations : la MNA
2. une méthode numérique de résolution d'un système linéaire : la décomposition LU
3. une méthode numérique d'intégration : trapézoïdale ou Gear2
4. une méthode numérique d'analyse non linéaire : Newton-Raphson

Il est représentable par le diagramme fonctionnel de la *figure 1*. Les résultats de simulation peuvent être imprimés dans un fichier ou traiter graphiquement à partir d'un interpréteur interactif de commandes.

## 2 Le fichier de simulation

On se limitera aux caractéristiques générales du fichier de simulation SPICE, les différentes syntaxes et paramètres d'instanciation seront trouvés dans la notice d'utilisation SPICE3F5 (NGSPICE) [1].

### 2.1 La structure

Le fichier de simulation SPICE est composée

- d'un titre
- d'une "netlist" donnant une description topologique du circuit à analyser à partir du type et de la valeur des différents composants actifs et passifs et de leur interconnexion
- d'un jeu de commandes pour fixer les analyses et le type de sortie à réaliser
- d'un jeu de directives de simulation
- d'un jeu de paramètres de contrôle
- d'un jeu de structures de données.
- d'une indication de fin de fichier

### 2.2 La syntaxe et la grammaire

La syntaxe et la grammaire SPICE sont très rudimentaires :

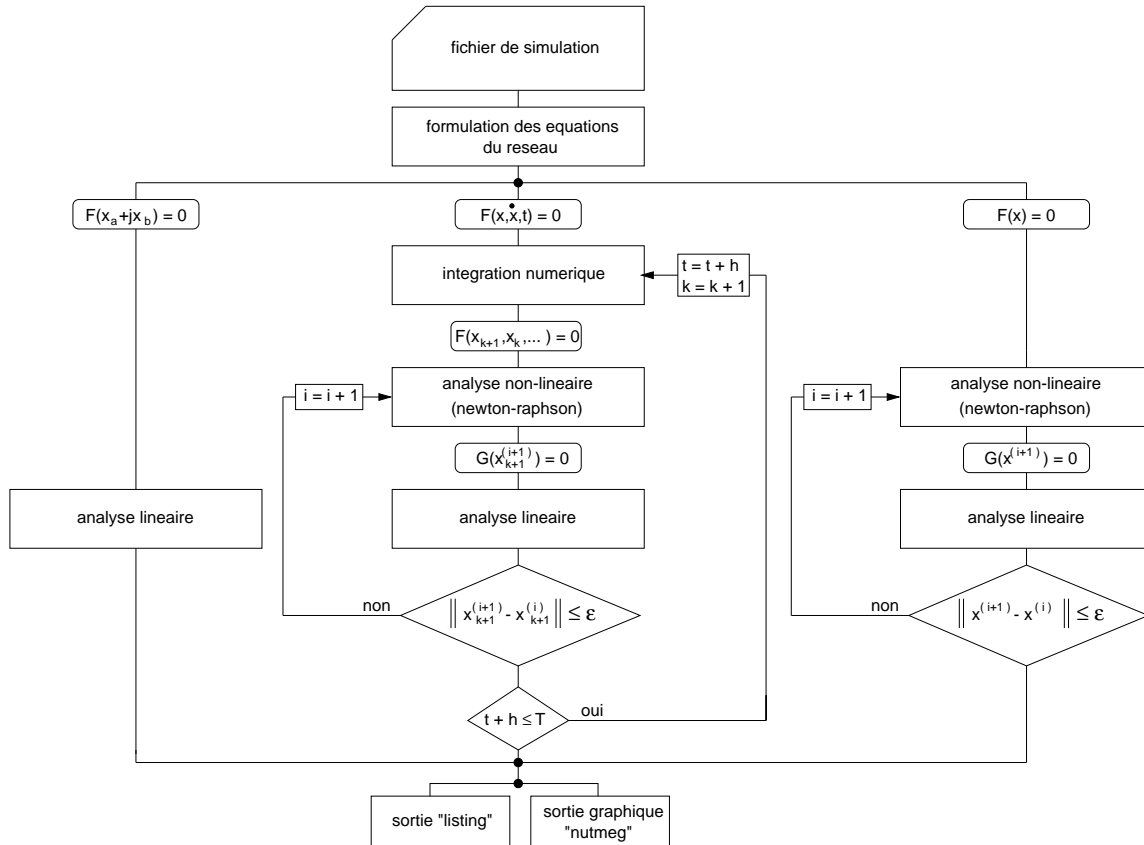


FIGURE 1 –

*Représentation fonctionnelle de SPICE*

- la première ligne du fichier de description est toujours interprétée comme le titre
- le fichier de simulation doit être terminée par la directive de simulation **.END**
- l'ordre dans lequel le fichier est généré est arbitraire à l'exception du titre et de la fin de fichier
- une ligne débutant par un astérisque (\*) n'est pas interprétée, c'est un commentaire
- une ligne débutant par un plus (+) est interprétée comme la continuation de la ligne précédente
- une ligne débutant par un point (.) est interprétée comme une commande ou une directive de simulation
- les parenthèses et les espaces sont traités comme un seul espace
- les identificateurs sont insensibles au casse (Npn=NPN=npn= ...)
- une instance de composant de la netlist est spécifiée par un nom de dispositif suivi par les labels des nœuds de connexion et par des paramètres de valeur et de contrôle
- le premier caractère d'un nom de dispositif est associé à un composant (**R**=résistance, **C**=condensateur, **M**=transistor MOS,...)
- un nœud peut être labélisé par n'importe quel identificateur alphanumérique, sauf le label 0 qui est implicitement interprété comme la masse du circuit.

## 2.3 Les composants élémentaires

- la résistance discrète (**R**)
- la résistance intégrée (**R**)
- le condensateur discret (**C**)
- le condensateur intégré (**C**)
- l'inductance (**L**)
- l'inductance mutuelle (**K**)
- les commutateurs (**S**) et (**W**)

## 2.4 Les lignes de transmission

- la ligne sans perte (**T**)
- la ligne avec pertes (**O**)
- la ligne RC uniformément distribuées (**U**)

## 2.5 Les composants actifs

- la diode (**D**)
- les transistors bipolaires (**Q**) (Ebers-Moll et Gummel-Poon)
- le transistor à effet de champ (**J**)
- les transistors MOS (**M**) (niveau 1 à 6 dans la version distribuée mais extensible à tous les modèles BSIM de BERKELEY)
- le transistors MESFET (**Z**)

## 2.6 Les sources de tension et de courant indépendantes

- les sources statiques (**DC**)
- les sources alternatives petit signal (**AC**)
- les sources impulsionnelles (**PULSE**)
- les sources sinusoïdales (**SIN**)
- les sources exponentielles (**EXP**)
- les sources continues par morceau et répétitives (**PWL**)
- les sources modulées en fréquence (**SFFM**)

## 2.7 Les sources dépendantes linéaires

- le transducteur tension-tension (**E**)
- le transducteur courant-courant (**F**)
- le transducteur tension-courant (**G**)
- le transducteur courant-tension (**H**)

## 2.8 Les sources dépendantes non-linéaires

Le simulateur SPICE peut interpréter des sources dépendantes non-linéaires en tension et en courant (**B**).

## 2.9 Les commandes d'analyse

- l'analyse alternative petit signal (**.AC**)

- l’analyse en bruit (**.NOISE**)
- l’analyse pôles-zéros (**.PZ**)
- l’analyse en sensibilité (**.SENS**)
- l’analyse de la fonction de transfert (**.TF**)
- l’analyse des distorsions (**.DISTO**)
- l’analyse statique multipoint (**.DC**)
- l’analyse statique monopoint (**.OP**)
- l’analyse transitoire (**.TRAN**)
- l’analyse de Fourier (**.FOUR**)

## 2.10 Les commandes de sortie

- sortie listing d’un tableau des valeurs calculées (**.PRINT**)
- sortie listing d’un graphique des valeurs calculées (**.PLOT**)

## 2.11 Les directives de simulation

- les options (**.OPTION**)
- les conditions initiales pour le statique (**.NODESET**)
- les conditions initiales pour le transitoire (**.IC**)
- l’inclusion de fichiers (**.INCLUDE**)
- la fin de fichier (**.END**)

## 2.12 Les structures de donnée

- les modèles (**.MODEL**) : certains composants (principalement les dispositifs intégrés) nécessitent pour leur instanciation un grand nombre de paramètres, et très souvent un circuit utilise beaucoup de ces composants avec le même jeu de paramètres (même technologie). Pour cette raison, SPICE permet de définir la notion de modèle qui est une structure de données définie une seule fois avec un nom unique et ensuite instanciée autant de fois que l’on veut en faisant référence à ce nom unique
- les sous-circuits (**.SUBCKT**) : le sous-circuit est une structure de données répondant au mêmes principes que les modèles, à la différence près qu’il permet le regroupement de plusieurs instances dans un corps de fichier commençant par la directive de simulation **.SUBCKT** et se terminant par la directive de simulation **.ENDS**, et que l’instanciation se fait avec le nom du sous-circuit précédé de la lettre **X**.

# 3 Les analyses basiques et le conditionnement des algorithmes correspondant

## 3.1 L’analyse statique **.DC**

### 3.1.1 Fonction

La commande **.DC** est utilisée pour déterminer une caractéristique de transfert d’un circuit. Une source de tension ou de courant (le stimuli) est spécifiée pour varier

dans une plage définie et pour chaque valeur du stimuli, une valeur de la variable spécifiée en sortie est calculée.

### 3.1.2 Principe de calcul

Toutes les sources indépendantes sont considérées comme statiques et tous les composants réactifs sont neutralisés : les inductances sont court-circuitées (valeur nulle) et les condensateurs sont ouverts (valeur infinie). SPICE doit ainsi résoudre un système d'équations différentielles implicite de la forme

$$F(\mathbf{x}) = 0$$

L'opérateur  $F$  étant en général un opérateur non linéaire, les tensions et les courants du circuit à analyser sont déterminés par l'algorithme de Newton-Raphson.

- Le critère de convergence de l'algorithme de Newton-Raphson

L'algorithme de [Newton-Raphson](#) considère que la solution a convergée sur l'itération  $j$ , lorsque le critère de la différence

$$|v_n^{(j)} - v_n^{(j-1)}| < \mathbf{vntol} + \mathbf{reltol} \cdot \max\{|v_n^{(j)}|, |v_n^{(j-1)}|\}$$

et le critère du résidu

$$|f(v_1^{(j)}, \dots, v_n^{(j)}) - f(v_1^{(j-1)}, \dots, v_n^{(j-1)})| < \mathbf{abstol} + \mathbf{reltol} \cdot \max\{|f(v_1^{(j)}, \dots, v_n^{(j)})|, |f(v_1^{(j-1)}, \dots, v_n^{(j-1)})|\}$$

sont simultanément respectés. On notera que le critère du résidu ne garantit pas que les lois de Kirchhoff sur le courant sont respectées, il assure simplement qu'elles sont approximativement les mêmes sur deux échantillons successifs. La convergence de l'algorithme de Newton-Raphson est contrôlable à partir de la directive de simulation **.OPTION**.

- Fixation d'un jeu de valeurs initiales par la directive de simulation **.NODESET**

La directive de simulation **.NODESET** facilite grandement la convergence de Newton-Raphson, notamment pour les circuits possédant par conception plusieurs points de fonctionnement (astables, bistables, ...), en permettant à l'utilisateur de fournir un jeu de valeurs initiales connues pour être proches de la solution finale.

- Le conditionnement de l'algorithme de Newton-Raphson

En cas de divergence de Newton-Raphson, le [principe du continuum](#)

est utilisé avec la technique du gmin graduel puis en cas d'insuccès avec la technique de la source graduelle. Par défaut, SPICE fixe le paramètre de contrôle de modèle **GMIN** de la directive de simulation **.OPTION** à  $10^{-12}$  et place implicitement cette conductance en parallèle sur toutes les jonctions PN des composants actifs du circuit. D'autre part, la technique du "bypass" qui permet une accélération de la vitesse de convergence est automatiquement utilisée (sans possibilité de la neutraliser).

## 3.2 Le point de fonctionnement `.OP`

### 3.2.1 Fonction

Une détermination des tensions et des courants de repos (points de fonctionnement) d'un circuit est implicitement effectuée lors d'une analyse transitoire ou une analyse alternative petit signal. Elle peut également être demandée explicitement par la directive `.OP`. Cette commande est très souvent utilisée pour vérifier l'état électrique des dispositifs actifs (typiquement la zone de fonctionnement et le régime des transistors MOS, des transistors bipolaires, ...) pour lesquels un listage des paramètres électriques des modèles est automatiquement réalisé.

### 3.2.2 Principes de calcul

C'est un cas particulier de l'analyse statique sans stimuli explicitement défini.

## 3.3 L'analyse transitoire `.TRAN`

### 3.3.1 Fonction

L'analyse transitoire est utilisée pour déterminer la réponse temporelle d'un circuit pour une durée d'observation  $[0, T]$ .

### 3.3.2 Principe de calcul

Pour l'analyse transitoire, le fonctionnement du circuit est formalisé par un système d'équations différentielles implicite de la forme

$$F(\mathbf{x}, \dot{\mathbf{x}}, t) = 0$$

L'opérateur  $F$  est en général un opérateur non linéaire fonction du temps par l'intermédiaire du vecteur  $\mathbf{u}(t)$  des sources indépendantes, du vecteur des variables inconnues  $\mathbf{x}$  et du vecteur dérivé  $\dot{\mathbf{x}}$ . L'intervalle d'observation  $[0, T]$  est discrétisé pour déterminer les instants de calcul  $[0, t_1, t_2, \dots, T]$ . Une détermination des points de fonctionnement est préalablement réalisée par une analyse statique afin d'établir un jeu de valeurs initiales. Puis, pour chaque instant de calcul une intégration numérique est effectuée par une méthode numérique d'intégration, afin de transformer le système d'équations différentielles en système d'équations algébriques. Enfin, le système est linéarisé par l'algorithme de Newton-Raphson pour délivrer la solution par la résolution du système d'équations linéaires obtenu.

- Les méthodes d'intégration numérique

Le simulateur SPICE permet de choisir la [méthode d'intégration](#)

Gear2 ou trapézoïdale par la directive de simulation `.OPTION`, par défaut SPICE sélectionne la méthode trapézoïdale.

- Le conditionnement des méthodes d'intégration numérique

SPICE permet un réglage dynamique du pas d'intégration

- par un contrôle de l'erreur de troncature locale (LTE) prédite par extrapolation polynomiale sur les échantillons précédents à partir de la directive de simulation `.OPTION`



- par une commutation automatique d’algorithme sur les transitions des sources indépendantes pour Gear2 (commutation sur Euler implicite)
- par comptage du nombre d’itérations dans l’algorithme de Newton-Raphson à partir de la directive de simulation **.OPTION**
- par fixation du pas maximum d’intégration à partir du paramètre de contrôle **TMAX** de la commande **.TRAN**, par défaut SPICE impose

$$TMAX = \frac{TSTOP - TSTART}{50}$$

- La méthode de Newton-Raphson

Les conditions de fonctionnement de l’algorithme de Newton-Raphson sont celles de l’analyse statique. Dès le deuxième pas d’intégration, les valeurs initiales sont celles du pas précédent et le nombre d’itérations est contrôlé par la méthode d’intégration pour assurer une meilleur convergence.

- Fixation d’un jeu de valeurs initiales par la directive de simulation **.IC**

La directive de simulation **.IC** peut être utilisée pour faciliter la convergence de l’analyse transitoire, elle a deux interprétations différentes selon le paramètre de contrôle **UIC** figurant à la fin de la commande **.TRAN**

1. si **UIC** est spécifié, les conditions initiales sur les condensateurs, les inductances et les composants actifs sont fixées à partir des valeurs fournies par **.IC**, l’analyse transitoire n’étant pas initialisée par une analyse statique, il faut veiller à ce que toutes les tensions nécessaires figurent dans **.IC**, le paramètre de contrôle **IC** des différents composants à la priorité sur **.IC**
2. si **UIC** n’est pas spécifié, une analyse statique est réalisée avant l’analyse transitoire pour déterminer les points de fonctionnement non imposés par **.IC** et au cours du transitoire, les contraintes sur tous les nœuds du circuit sont relâchées

## 3.4 L’analyse alternative petit signal **.AC**

### 3.4.1 Fonction

L’analyse alternative petit signal calcule les variables de sortie spécifiées en fonction de la fréquence. Cette commande est utilisée pour caractériser un circuit en fréquences (fonctions de transfert, bruit, analyse pôles-zéros, sensibilités) . On peut noter que l’amplitude des signaux alternatifs n’ayant pas d’incidence sur la linéarisation du circuit, il est souvent judicieux de spécifier une amplitude unité en entrée, puisque dans ce cas les variables de sortie correspondent directement aux fonctions de transfert.

### 3.4.2 Principe de calcul

Une détermination des points de fonctionnement par une analyse **.DC** est préalablement effectuée pour déterminer les paramètres des modèles linéarisés de tous les composants non linéaires présents dans le circuit. Ensuite le système d’équations linéaires est [résolu](#)

par l’algorithme de décomposition LU travaillant en arithmétique complexe.

## 3.5 La directive de simulation .OPTION

### 3.5.1 Les paramètres de contrôle d'impression

- **ACCT** : impression des statistiques de simulation
- **LIST** : impression du fichier de description d'entrée
- **NODE** : impression de la table des nœuds du circuit
- **NOMOD** : suppression de l'impression des paramètres des modèles
- **NOPAGE** : suppression des sauts de page

### 3.5.2 Les paramètres de contrôle des modèles

Toutes ces options peuvent être annulées pour un composant particulier si la spécification considérée est explicitement formulée dans les paramètres d'appel du composant.

- **GMIN** : conductance minimum implicite , par défaut elle est fixée à  $10^{-12}$ , cette conductance est placée en parallèle sur toutes les jonctions PN des composants actifs du circuit
- **TEMP** : indique la température implicite de simulation du circuit, par défaut elle est fixée à  $27^{\circ}C$
- **TNOM** : indique la température implicite de caractérisation des modèles , par défaut elle est fixée à  $27^{\circ}C$
- **BADMOS3** : utilisation de la première version pour un transistor de niveau 3
- **DEFAD** : indique la valeur implicite de l'aire de drain pour le calcul des capacités de jonction des transistor MOS, par défaut elle est fixée à 0
- **DEFAS** : indique la valeur implicite de l'aire de source pour le calcul des capacités de jonction des transistor MOS, par défaut elle est fixée à 0
- **DEFL** : indique la valeur implicite de la longueur des transistor MOS, par défaut elle est fixée à  $100\mu m$
- **DEFL** : indique la valeur implicite de la largeur des transistor MOS, par défaut elle est fixée à  $100\mu m$

### 3.5.3 Les paramètres de contrôle de la méthode de Newton-Raphson

- **VNTOL** : tolérance absolue pour le critère de la convergence de la différence , par défaut elle est fixée à  $1\mu V$
- **ABSTOL** : tolérance absolue pour le critère de la convergence résiduel, par défaut elle est fixée à  $1pA$
- **RELTOL** : tolérance relative, par défaut elle est fixée à 0.001

### 3.5.4 Les paramètres de contrôle spécifiques à .DC

- **ITL1** : nombre d'itérations maximum pour une analyse statique sur un point, par défaut il est fixé à 100
- **ITL2** : nombre d'itérations maximum pour une analyse statique sur un multi-point (caractéristiques de transfert), par défaut il est fixé à 50

### 3.5.5 Les paramètres de contrôle spécifiques à .TRAN

- **METHOD** : méthode d'intégration "gear" ou "trapézoïdal", par défaut elle est fixée à "trapézoïdal"

- **CHGTOL** : tolérance absolue pour la variable considérée (charge pour un condensateur ou flux pour une inductance) lors du contrôle du pas d'intégration par l'erreur de troncature locale (LTE), par défaut elle est fixée à  $10^{-14}$
- **TRTOL** : facteur de pondération de l'erreur de troncature locale (LTE), par défaut il est fixé à 7
- **ITL4** : nombre d'itérations maximum sur un pas d'intégration, par défaut il est fixé à 10

### 3.5.6 Les paramètres de contrôle de la méthode de d'élimination de Gauss

- **PIVTOL** : valeur minimum pour qu'un coefficient puisse être considéré comme pivot, par défaut il est fixé à  $10^{-13}$
- **PIVREL** : rapport entre une valeur acceptable de pivot et le plus grand coefficient de la colonne correspondante, par défaut il est fixé à  $10^{-3}$ .

## 4 L'exécution du fichier de simulation

La version publique de SPICE3F5 est couplée à un interpréteur interactif graphique (nutmeg) permettant la visualisation et le traitement des résultats simulés, l'utilisateur dispose ainsi de plusieurs modes d'exécution du fichier de simulation.

### 4.1 Exécution en mode non-interactif (batch) :

L'exécution standard du fichier de simulation s'effectue par la commande unix :  
**"spice3 -o nom-du-fichier-de-sortie -b nom-du-fichier-d'entrée"**  
 dans ce cas le fichier généré contient les informations de simulations, précisées par les paramètres de contrôle d'impression de la directive de simulation **.OPTION**, ainsi que les résultats de simulation explicitement demandés par la directive de simulation **.PRINT** ou **.PLOT**. On notera que le fichier de simulation correspondant est directement compréhensible par la plupart des simulateurs électriques commerciaux.

### 4.2 Exécution en mode interactif

On notera que les commandes à l'intérieur du mode interactif doivent impérativement être en minuscules et que les commandes de simulation ne comportent pas de point d'identification.

#### ● Exécution en mode "run" :

on entre dans l'environnement interactif par la commande unix  
**"spice3 nom-du-fichier-d'entrée"**,  
 on effectue l'exécution du fichier par la commande nutmeg  
**"run nom-du-fichier-d'entrée"**  
 on effectue l'analyse désirée par la commande nutmeg (par exemple)  
**"ac dec 20 1 100e3"**  
 et on réalise ensuite interactivement les graphes et les traitements désirés.

#### ● Exécution en mode "script" :

Les commandes SPICE sont explicitement indiquées dans le fichier de simulation, dans ce cas le jeu de commande est encapsulé dans un bloc encadré par les directives de simulation **.CONTROL** et **.ENDC**, on peut ainsi écrire des scripts "nutmeg" pour effectuer différents traitements mathématiques sur les données générées par SPICE, on notera que ces scripts doivent également être impérativement en minuscules et que les commandes de simulation ne comportent pas de point d'identification. Si l'entrée dans l'environnement interactif est effectuée par la commande unix

**"spice3"**,

l'exécution du fichier de simulation et du script sont réalisés par la commande nutmeg **"source nom-du-fichier-d'entrée"**,

et si l'entrée dans l'environnement interactif est effectuée par la commande unix

**"spice3 nom-du-fichier-d'entrée"**,

l'exécution du fichier de simulation et du script est directement réalisée.

- Exécution en mode "rawfile" :

Dans une première phase une exécution du fichier de simulation doit être réalisée par la commande unix :

**"spice3 -b nom-du-fichier-d'entrée -r nom-du-fichier-de-sortie"**,

le listing "nom-du-fichier-de-sortie" ainsi généré ne contient que des vecteurs de nombres contrôlés par la directive de simulation **.SAVE**, en son absence ou avec la ligne **.SAVE all**, toutes les tensions de nœud et tous les courants de branche sont présents dans le fichier alors qu'avec par exemple la ligne **.SAVE v(n1) v(n2)** seule la tension au nœuds **n1** et **n2** est présente. Dans une seconde phase après être entré dans l'environnement interactif par la commande unix

**"spice3"**,

on effectue le chargement du fichier "nom-du-fichier-de-sortie" par la commande nutmeg **"load nom-du-fichier-de-sortie"**

et on réalise ensuite interactivement les graphes et les traitements désirés.

On notera que l'usage consiste à mettre l'extension ".sp" ou ".cir" sur les noms des fichiers de simulation.

## 5 Quelques exemples de fichiers de simulation SPICE

### 5.1 Caractérisation d'un transistor NMOS

La caractérisation statique d'un dispositif (composant actif ou circuit) est une opération courante en électronique, analogique. ainsi les montages de la *figure 2* permet de déterminer la caractéristique  $I_{ds} = f(V_{ds})$ .

#### 5.1.1 Une première méthode

Pour obtenir la caractéristique  $I_{ds} = f(V_{ds})$ , le courant est prélevé dans une sonde de tension (*figure 2*), et la conductance  $g_{ds} = \frac{\partial I_{ds}}{\partial V_{ds}}$  peut être déterminée par la fonction **deriv** de nutmeg, on génère directement les graphes et les fichiers postscript (*figure 3* et *figure 4*) correspondant. La température de caractérisation est fixée à  $37^{\circ}C$ .

CARATERISATION MOS N

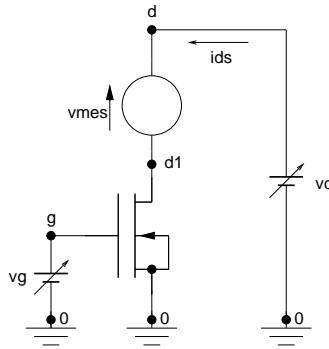


FIGURE 2 -

*Caractérisation d'un transistor MOS*

```

* on parametre le modele de niveau 1 de SPICE
.MODELE nmos1 NMOS LEVEL 1 TOX=1.264e-8
+ KP=100e-6 LAMBDA=0.03 VT0=0.7 GAMMA=0.8 PHI=0.5
+ CJ=0.35e-3 MJ=0.450 CJSW=0.31e-9 MJSW=0.49
vd d 0 0
vg g 0 0
vmes d d1 0
* on instancie le modele nmos1
mn1 d1 g 0 0 nmos1 W=47u L=1.0u
+ ad=76p as=76p pd=98u ps=98u
.control
dc vd 0 5 0.01 vg 1 5 0.5
*on nomme la variable
let ids=i(vmes)
plot ids
* on genere un fichier postscript
set hcopydevtype=postscript
hardcopy ids.ps ids
let gds=deriv(i(vmes))
plot gds
set hcopydevtype=postscript
hardcopy gds.ps gds
.endc
.option temp=37.0
* fin de fichier
.end

```

### 5.1.2 Une deuxième méthode

Le courant et la conductance sont directement prélevés sur le dispositif par le paramètre de contrôle **@dev[param]** et à partir d'une exécution en mode "rawfile" sans utilisation d'une sonde de tension.

CARATERISATION MOS N

```

* on parametre le modele de niveau 1 de SPICE

```

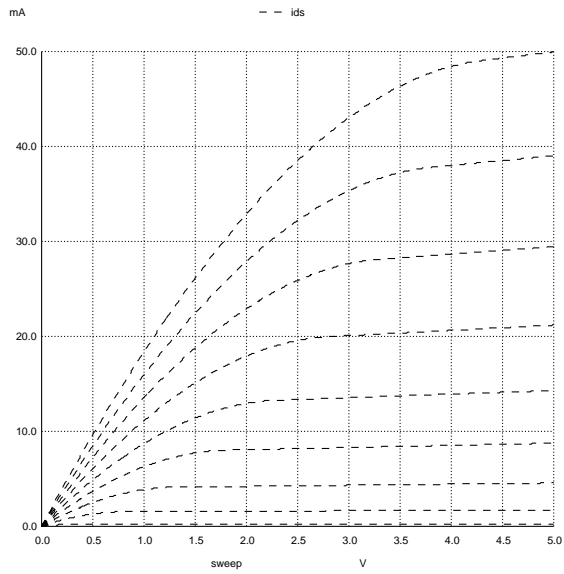


FIGURE 3 –  
*Graphes Ids vs Vds*

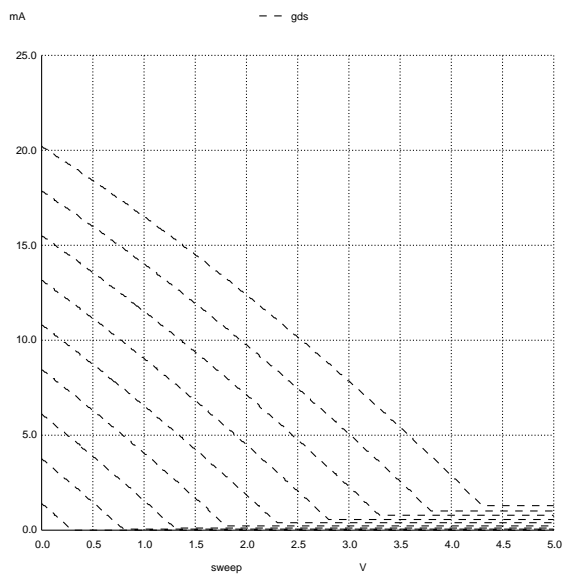


FIGURE 4 –  
*Graphes gds vs Vds*

```
.MODELE nmos1 NMOS LEVEL 1 TOX=1.264e-8
+ KP=100e-6 LAMBDA=0.03 VT0=0.7 GAMMA=0.8 PHI=0.5
+ CJ=0.35e-3 MJ=0.450 CJSW=0.31e-9 MJSW=0.49
vd d 0 5
vg g 0 5
* on instancie le modele nmos1
mn1 d g 0 0 nmos1 W=47u L=1.0u
```

```

+ ad=76p as=76p pd=98u ps=98u
.dc vd 0 2.5 0.05 vg 0.75 2.75 0.5
.save @mn1[id]
.save @mn1[gds]
* fin de fichier
.end

```

## 5.2 Analyses alternatives d'un filtre passe-bas d'ordre deux

Si on considère le fichier de simulation correspondant au biquad à variables d'état de la *figure 5*, on notera que l'amplificateur opérationnel idéal est modélisé par une source de tension commandée en tension (transducteur tension-tension) de gain très grand (infini), et que le [macro-modèle](#)

correspondant est considéré en sous-circuit. Plusieurs commandes sont enchaînées par le script.

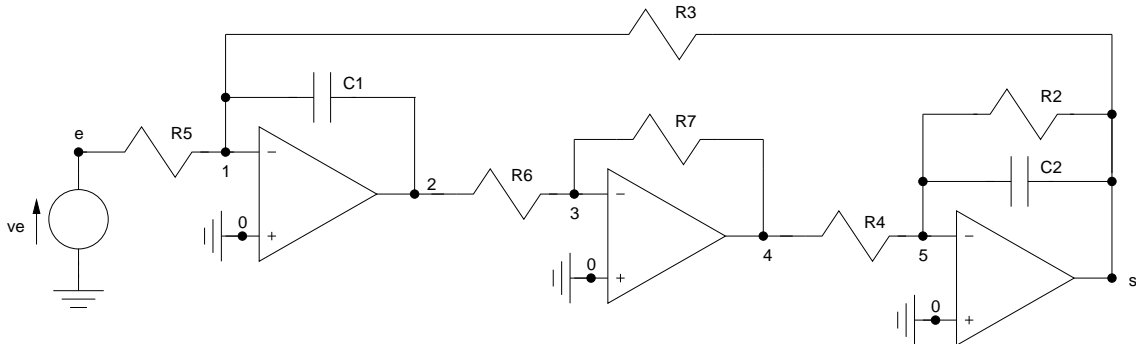


FIGURE 5 –

*Biquad à variables d'état*

```

BIQUAD RC A VARIABLES D'ETAT
* on modelise l'amplificateur operationnel
.SUBCKT AOV_IDEAL ep em sp
e1 sp 0 ep em 1.000000e+06
rfonc1 ep 0 1.0e12
rfonc2 em 0 1.0e12
rfonc3 sp 0 1.0e12
.ENDS
* on instancie l'amplificateur operationnel
xaop1 0 1 2 AOV_IDEAL
r5 e 1 5.343800e+04
c1 1 2 5.000000e-09
r3 1 s 3.783100e+04
xaop2 0 3 4 AOV_IDEAL
r6 2 3 10e3
r7 3 4 10e3
xaop3 0 5 s AOV_IDEAL
r2 5 s 4.935800e+04

```

```

r4 4 5 3.783100e+04
c2 5 s 5.000000e-09
Ve e 0 dc 0.0 ac 1.0 0.0
.control
* analyse alternative petit signal
ac dec 20 1 100e3
let vs1=db(s)
plot vs1
print vs1 > vs1.lis
*on genere le fichier postscript
set hcopydevtype=postscript
hardcopy vs1.ps vs1
* modification de C1 et C2
alter @c1[capacitance]=3e-9
alter @c2[capacitance]=3e-9
ac dec 20 1 100e3
let vs2=db(s)
plot vs2
print vs2 > vs2.lis
* analyse du bruit
noise v(s) ve dec 20 100 100e3
print inoise_total onoise_total
* on accede aux plots de densite spectrale
setplot noise2
plot onoise_spectrum
print onoise_spectrum > onoise.lis
plot inoise_spectrum
print inoise_spectrum > inoise.lis
.endc
* fin de fichier *
.end

```

## 6 Ce que fait pas ou fait mal SPICE3F5

Un certain nombre de problèmes ont été constatés sur la version SPICE3F5 que nous avons implanté. On notera que ces problèmes peuvent avoir pour origine un défaut d'implantation, une mauvaise utilisation du logiciel, ... :

- les sondes de tension permettant le prélèvement des courants (ex :vmes n1 n2 0) ne sont pas considérées en alternatif (.AC), on peut les remplacer par une source de tension commandée en tension (Gx x 0 vmes 1)
- l'accès aux capacités actives (@mn1[cgs],...) est impossible pour les niveaux de modèle supérieurs à trois, ce défaut a probablement pour origine notre implantation des modèles fournis par Berkeley.
- les valeurs de capacités actives affichées dans le fichier résultat pour les niveaux inférieur à trois ne correspondent pas du tout aux valeurs calculées par le modèle de [Meyer](#)

. Il semble que ce ne soit qu'un problème d'affichage, puisque les performances fréquentielles des dispositifs actifs mesurées avec SPICE correspondent parfaitement aux performances calculées et mesurées avec d'autres simulateurs élec-



- triques utilisant le modèle capacitif de Meyer.
- pas de repliement (M) explicite au niveau de l'instanciation des modèles des transistors MOS.

## Références

- [1] T. QUARLES, A.R. NEWTON, D.O. PEDERSON, and A. SANGIOVANNI-VINCENNELLI. "*SPICE 3 Version 3F5 user's manual*". University of California Berkeley CA 94720, 1996.