**Université de Laghouat**

**Département d’électrotechnique**

**3eme année Lic/ELT Cours : Logiciels de simulation**

**Programme**

**Chapitre 1 : Généralités sur les logiciels de simulation**

* **Introduction**
* **Les étapes de simulation**
* **Les logiciels de simulation utilisés en génie électrique**
* **Notions sur la programmation et la simulation sous Matlab**

**Chapitre 2 : Modélisation et Simulation des circuits électrique**

* **Simulation des équations différentielles**
* **Simulation de circuits RC, RL, RLC**

**Chapitre 3 : Modélisation et Simulation des circuits d’électronique de puissance**

* **Modélisation et simulation des circuits non commandés**
* **Modélisation et simulation des circuits commandés**

**Chapitre 4 : Simulation des machines électriques**

* **Modélisation et simulation des machines à courant continu**
* **Modélisation et simulation des machines à courant alternatif**

**Chapitre 5 : simulation des circuits de régulation**

* **Modélisation et simulation de régulateur PI**
* **Modélisation et simulation de régulateur PID**
* **Application 1 : Régulation de vitesse de machine a courant continu**

**Chapitre 6 : Simulation des réseaux électriques**

**Chapitre I : Généralités sur les logiciels de simulation**

**I.1 Introduction**

La **simulation** est l’un des outils d’aide à la décision les plus efficaces à la disposition des concepteurs et des gestionnaires des systèmes complexes. Elle consiste à construire un modèle d’un système réel , conduire des expériences sur ce modèle et interpréter les observations fournies par le déroulement du modèle afin de comprendre le comportement de ce système, formuler des décisions et d’en améliorer les performances du système.

.

Le but c’est de comprendre le comportement dynamique du système, de comparer des configurations, d’évaluer différentes stratégies de pilotage, d’évaluer et d’optimiser des performances.

**Le système**

**Bloc 1**

**Bloc 2**

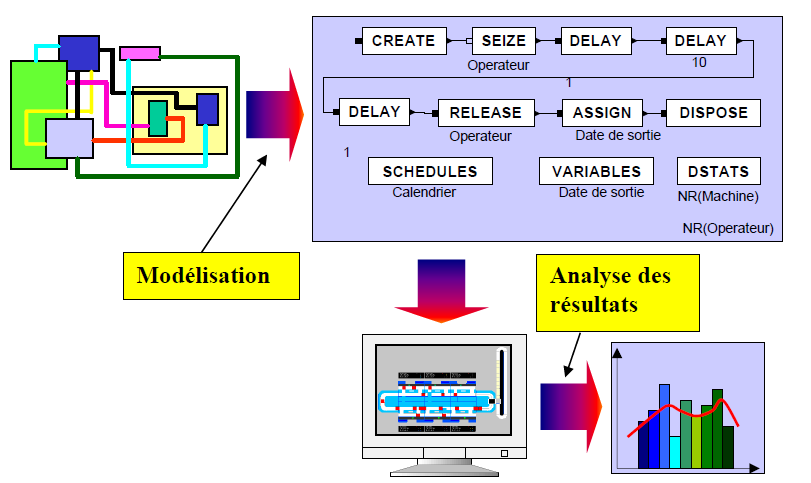
**Bloc 3**

**Bloc 4**

**Implémentation de modèle**

**simulation**

**Simulation**

****

Les **logiciels de simulation** permettent de simuler le comportement dynamique d'un système qui est représenté par un modèle mathématique. À chaque étape de la simulation du modèle, l'état de chaque partie du système est calculé à l'aide de solveurs basés sur le temps ou sur événement. En règle générale, les logiciels de simulation intègrent également des outils de visualisation, tels que des systèmes d'affichage des données, qui permettent de surveiller la simulation pendant son exécution.

Les ingénieurs et les scientifiques utilisent les logiciels de simulation pour de nombreuses raisons :

* Il est souvent moins coûteux et plus simple de créer et simuler un modèle que de créer et tester un prototype matériel.
* Si le prototype matériel n'est pas disponible suffisamment tôt au cours du processus de développement, les logiciels de simulation permettent d’explorer l'espace de conception et tester différents scénarios.

**I.2 Les étapes de la simulation**

**I.2.1 Analyse du problème**

Elle permet de préciser le contexte de l'étude. Elle se décompose en:

* Identification du problème; spécification des objectifs.
* Réalisation d'une première ébauche du modèle qui a pour but d'en délimiter les frontières et de spécifier les données dont on a besoin.
* Validation auprès de l'utilisateur (celui qui est à l'origine de l'étude).

Le but à atteindre dans cette étape est de construire un modèle valide qui soit le plus simple possible, tout en restant cohérent avec les objectifs de l'étude. Il faut donc tout d’abord formuler explicitement ces objectifs, et les divers scénarios à étudier. Le compromis difficile à trouver; en effet, le concepteur du modèle cherche toujours la simplification, alors que l'utilisateur souhaite que soient finement représentés les constituants du système, la condition indispensable pour que l'utilisateur accepte les résultats de la simulation est qu'il soit convaincu que le modèle construit est fidèle à la réalité.

Par exemple, s'il s'agit de modéliser une machine, l’utilisateur imposerait à modéliser tous les processus physiques générateurs de délais (changements d'outils, bridage, réglage, usinage, ...). Cette dernière approche, si elle a l'inconvénient d'alourdir le modèle, présente cependant un avantage important:

**I.2.2 Construction du modèle**

Elle comprend la modélisation logico-mathématique, ou il est important de construire un programme facilement modifiable. Cette étape se termine par une validation qui consiste à comparer le comportement du modèle avec celui du système physique qu'il est censé représenter.

**I.2.3 Exploitation du modèle**

Quand le modèle est validé, il peut servir à l'évaluation du comportement du système. Cette phase nécessite une définition précise de la campagne d'exploitation (quelles hypothèses veut-on vérifier, dans quel contexte), la production de mesures par la simulation, la mise en forme et la comparaison des résultats obtenus aux objectifs poursuivis. S'ils n'ont pas été atteints, de nouveaux scénarios sont proposés et testés jusqu'à satisfaction.

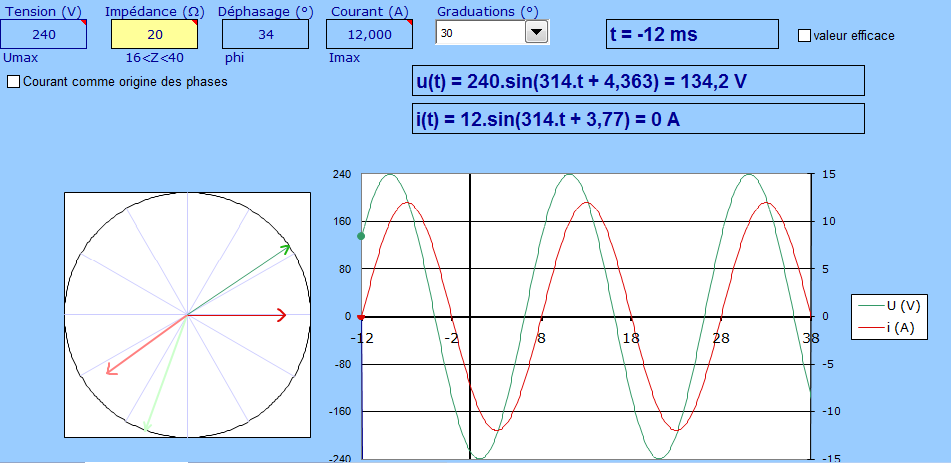
Cette étape nécessite que soient déterminés avec rigueur la durée de la simulation et le nombre de réplications (exécutions du modèle de simulation).

**I.3 Les logiciels de simulation utilisés en Génie électrique**

Plusieurs logiciels sont disponibles dans le domaine génie électrique, nous allons citer quelques logiciels les plus utilisés.

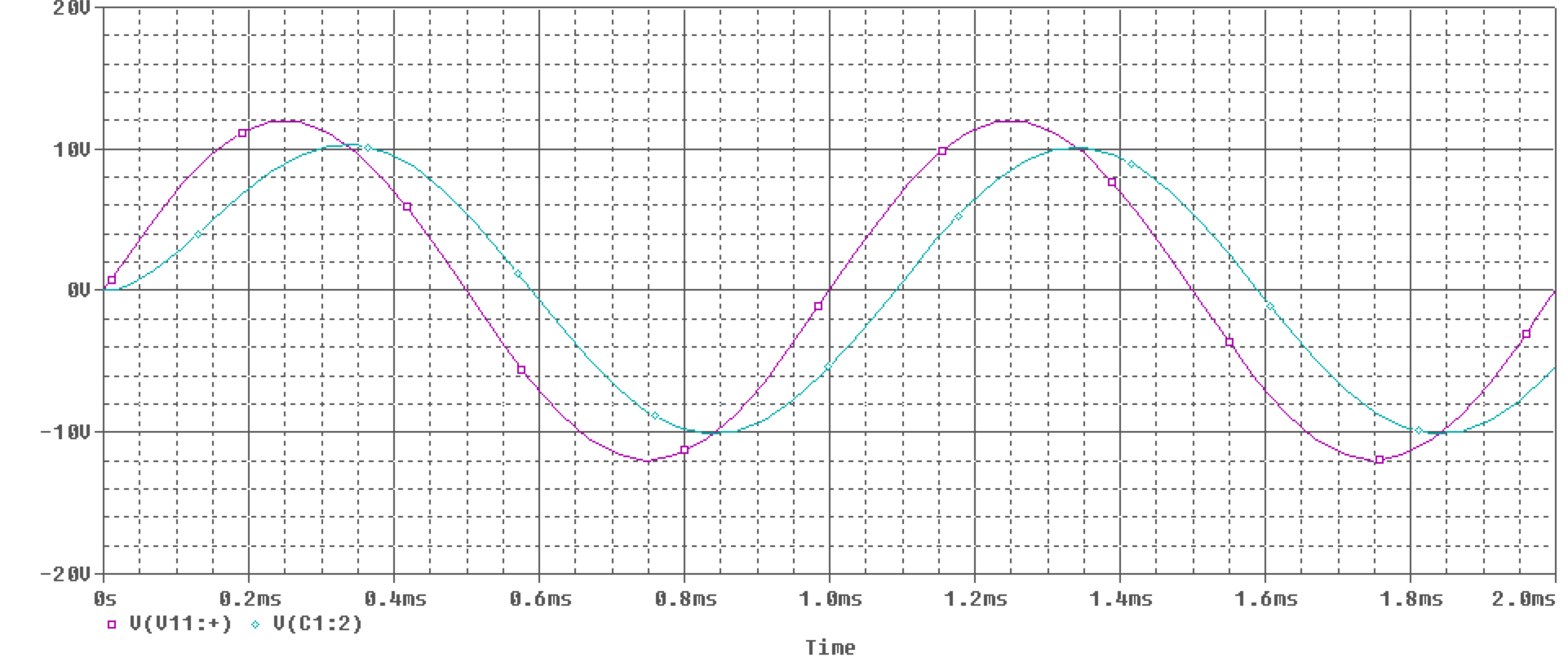
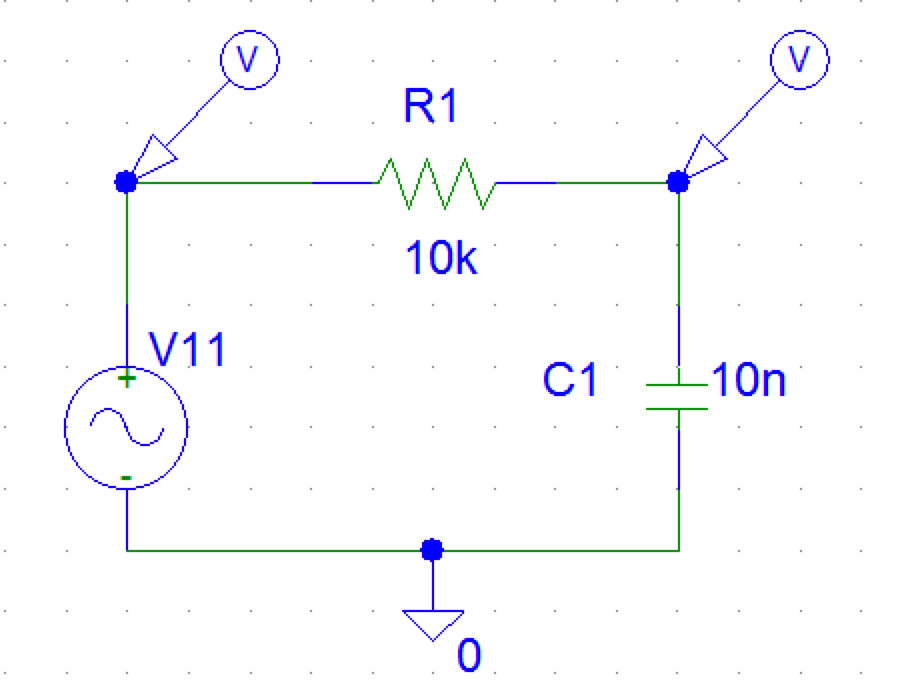
**I.3.1 Simulation des circuits avec Excel :**

Excel est un outil de gestion de données et peut aussi faire la simulation, en effet, le classeur Excel comprend un grand nombre de formule dépendante des unes et des autres plus a des outils d'analyse de simulation qui permettent de connaître les situations possibles grâce au modèle déjà en place et donc en émettant des hypothèses et connaître les résultats de ces hypothèses. Il existe plusieurs outils sous Excel, mais les deux plus importants sont la valeur cible (qui permet de déterminer le résultat voulu et Excel va trouver une combinaison possible et réelle) et le solveur (permettant de faire les calcul ).

****

**I.3.2 Le logiciel**  **PSPICE**

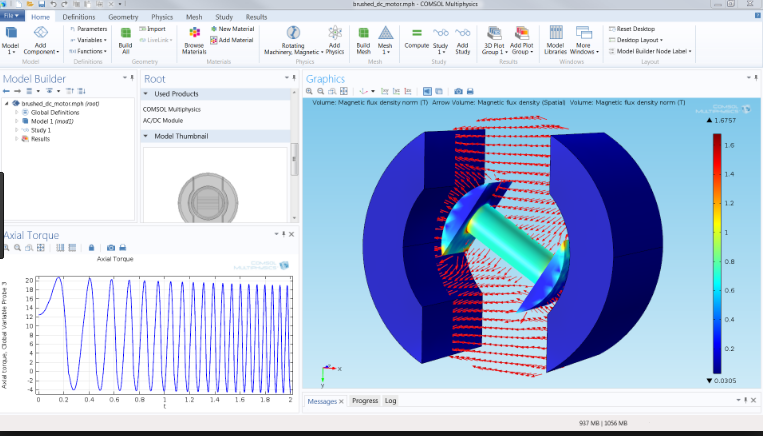
PSPICE est un programme de simulation et des logiciels utiles qui permet de créer, analyser et simuler les performances des circuits électriques ou électroniques dans un laboratoire virtuel puissant.

****

**I.3.3 COMSOL Multiphysics**

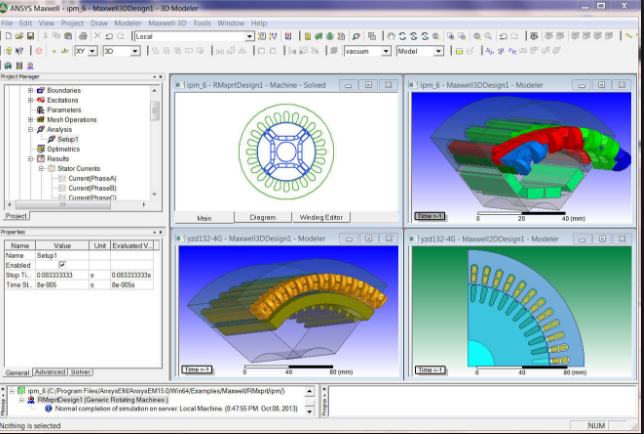
Le COMSOL Multiphysics (ancien nom : Femlab), est un logiciel de [simulation numérique](https://fr.wikipedia.org/wiki/Simulation_num%C3%A9rique) qui prend comme méthode de calcul les éléments finis. Ce logiciel permet de simuler de nombreuses [physiques](https://fr.wikipedia.org/wiki/Physique) et applications en [ingénierie](https://fr.wikipedia.org/wiki/Ing%C3%A9nierie), il contient plusieurs modules parmi les quelles nous citons ;

* **Le Semiconductor Module** qui permet une analyse détaillée du fonctionnement de composants semi-conducteurs.
* **L'AC/DC Module**  qui permet de simuler des champs électriques, magnétiques et électromagnétiques dans des applications statiques et à basses fréquences. Les applications type concernées sont les condensateurs, les inducteurs, les isolateurs, les bobines, les moteurs, les actionneurs et les capteurs
* **Le Wave Optics Module** qui propose des outils dédiés à la simulation de la propagation des ondes électromagnétiques dans des milieux optiques linéaires et non linéaires, afin de permettre une simulation précise des composants optiques et une optimisation de leurs designs. A l’aide de ce module, vous pourrez simuler des ondes électromagnétiques à haute fréquence dans les domaines temporel ou fréquentiel au sein de structures optiques.
* **Heat Transfer Module**, il est possible d'étudier les effets du chauffage et du refroidissement dans les dispositifs, les composants ou les procédés



**I.3.4 les logiciels ANSYS**

ANSYS développe des logiciel de simulation en 2 et 3D, parmi les quels on trouve ANSYS Maxwell qui est le logiciel de simulation de champs électromagnétiques pour la conception et l'analyse des moteurs électriques, actionneurs, capteurs, transformateurs et autres dispositifs électromagnétiques et électromécaniques. il comprend des interfaces de conception spécialisées pour les machines électriques et les convertisseurs de puissance.



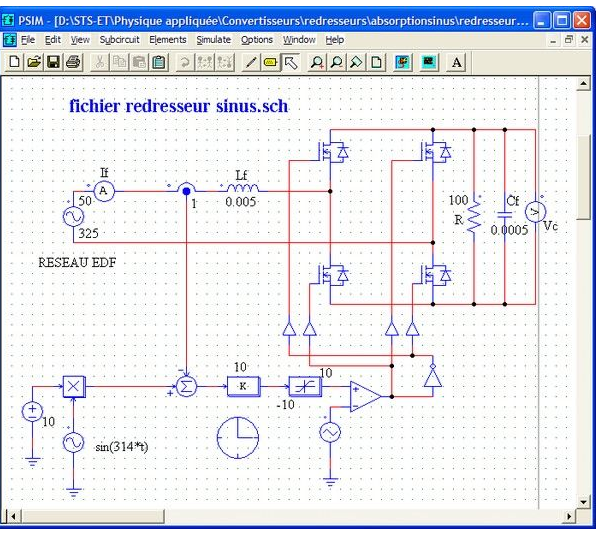
**I.3.5 Logiciel Psim**

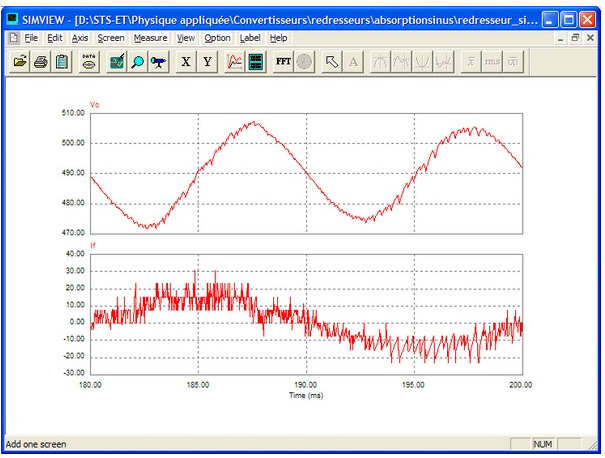
Le logiciel Psim est un logiciel de simulation pour l’électricité, l’électronique et l’électrotechnique.

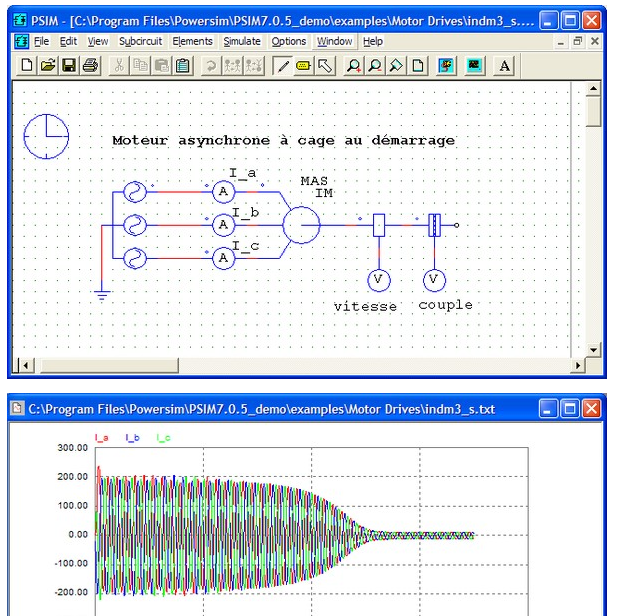
Le module Psim permet de :

* de réaliser le schéma du montage, à partir des éléments de la bibliothèque (composants, machines, transformateurs, interrupteurs électroniques et leurs éléments de commande, appareils de mesures…).
* de configurer et de lancer la simulation des grandeurs (électriques, mécaniques).

Le module Simview.exe permet de visualiser les courbes de variation des grandeurs définies par les appareils de mesure.

****

****

****

**I.3.6 Simulink sous Matlab**

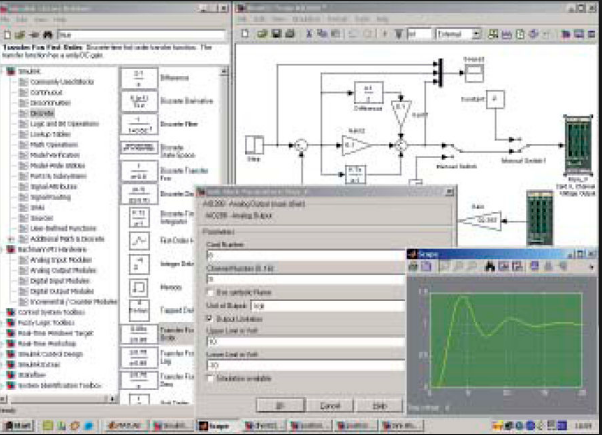
**Simulink** est un [logiciel](https://fr.wikipedia.org/wiki/Logiciel) de [modélisation](https://fr.wikipedia.org/wiki/Mod%C3%A8le_math%C3%A9matique) système multi-physique, il peut modéliser un système, simuler son comportement, décomposer le design avant son implémentation. Avec Simulink, il est possible de créer des diagrammes hiérarchiques de blocs pour la modélisation haut niveau d’un système,

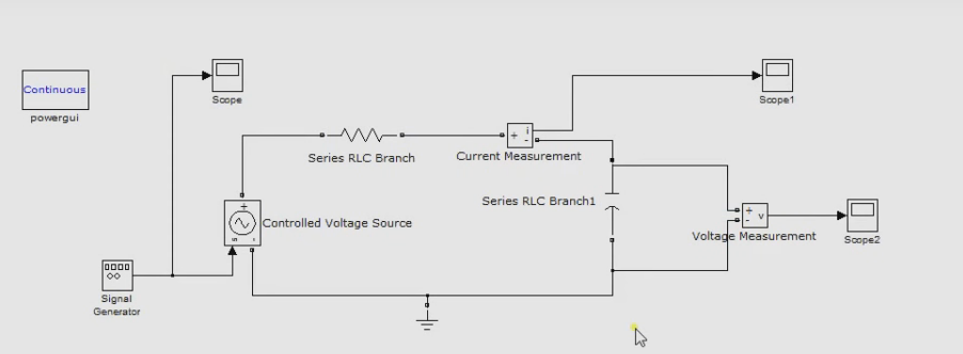
La simulation permet de s'assurer que le système correspond aux spécifications. La simulation est paramétrée de manière à optimiser les performances.

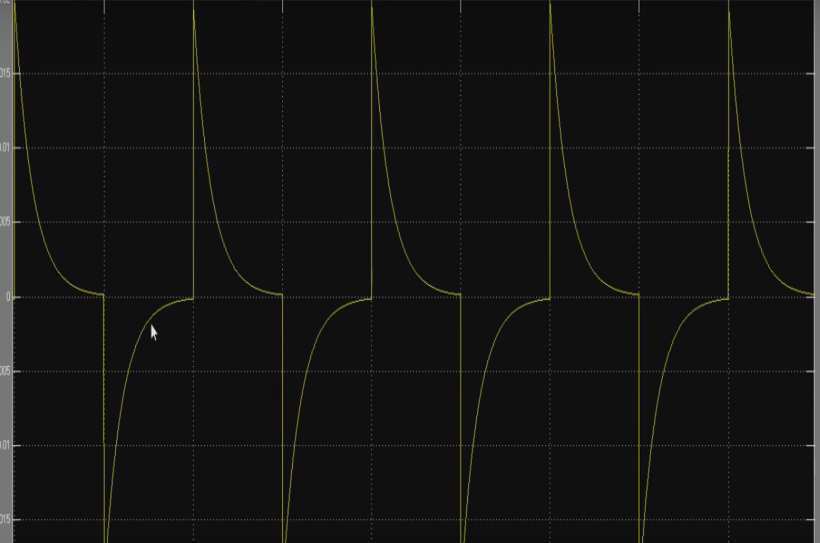
Simulink peut modéliser des données simples ou multicanaux, des composants linéaires ou non. Simulink peut simuler des composants numériques, analogiques ou mixtes. Il peut modéliser des sources de signaux et les visualiser.

.

****

****

****

****

**I.4 Les applications de la simulation**

Les applications de la simulation sont illimitées. Parmi les domaines dans lesquels elle est le plus utilisée, on peut citer :

* L’informatique : recherche de configurations, réseaux, architecture de bases de données, ...
* La production : gestion des ressources de fabrication, machines, stocks, moyens de manutention, ...
* La gestion : marketing, tarification, prévisions, gestion du personnel, ...l’administration : gestion du trafic, du système hospitalier, de la démographie, ...
* L’environnement : pollution et assainissement, météorologie, catastrophes naturelles, ...
* etc ...

**I.5. Notions sur la programmation à l’aide de Matlab**

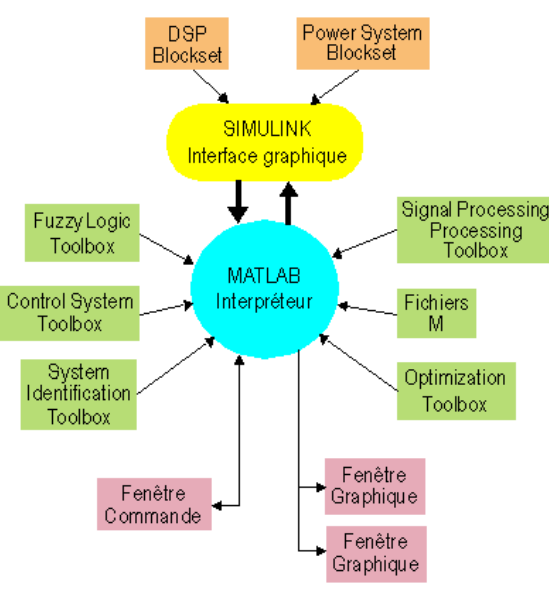
**I.5.1) Définition**

MATLAB (matrix laboratory) est un langage de programmation émulé par un environnement de développement du même nom ; il est utilisé dans des domaines très différents comme l’ingénierie, les sciences et l’économie dans un contexte aussi bien industriel que pour la recherche à des fins de calcul numérique. Développé par la société The MathWorks, MATLAB permet de manipuler des matrices, d'afficher des courbes et des données, de mettre en œuvre des algorithmes, de créer des interfaces utilisateurs, et peut s’interfacer avec d’autres langages comme le C, C++, Java, et Fortran. Matlab peut s’utiliser seul ou bien avec des toolbox (boîte à outils).

Il existe deux modes de fonctionnement:

1. mode interactif: MATLAB exécute les instructions au fur et à mesure qu'elles sont données par l'usager.

2. mode exécutif: MATLAB exécute ligne par ligne un "fichier M" (programme en langage MATLAB).



**Environnement MATLAB**

* **Fenêtre Commande :** Dans cette fenêtre, l'usager donne les instructions et MATLAB retourne les résultats.
* **Fenêtres Graphique :** MATLAB trace les graphiques dans ces fenêtres.
* **Fichiers M :** Ce sont des programmes en langage MATLAB (écrits par l'usager).
* **Toolboxes:** Ce sont des collections de fichiers M développés pour des domaines d'application spécifiques (Signal Processing Toolbox, System Identification Toolbox, Control System Tool-box, u-Synthesis and Analysis Toolbox, Robust Control Toolbox, Optimization Toolbox, Neural Network Toolbox, Spline Toolbox, Chemometrics Toolbox, Fuzzy Logic Toolbox, etc.)
* Simulink: C'est l'extension graphique de MATLAB permettant de travailler avec des diagrammes en blocs.
* Blocksets: Ce sont des collections de blocs Simulink développés pour des domaines d'application spécifiques (DSP Blockset, Power System Blockset, etc.).

**I.5.2) L’espace de travail (workspace)**

Les variables sont définies au fur et à mesure que l'on donne leurs noms et leurs valeurs numériques ou leurs expressions mathématiques.

Les variables ainsi définies sont stockées dans l'espace de travail et peuvent être utilisées dan les calculs subséquents.

Pour obtenir la liste des variables actives de l’espace de travail on dispose des commandes **who** et **whos**.

* La commande **who** affiche le nom des variables actives.
* La commande **whos** donne plus d’informations : le nom, la taille du tableau (nombre de lignes et de colonnes) associé, l’espace mémoire utilisé (en Bytes)
* La commande **clear** permet de nettoyer l’espace de travail

**Exemple :**

>> x=2\*pi/3; y=sin(x); z=cos(x);

>> A = [ 1 3; 4 2 ]; B = A\*A;

>> t = ’bonjour’;

>> who

>> whos

>> clear x y t

**I.5.3) L’enregistrement sous Matlab**

L’instruction **save nom-fi**c enregistre toutes les variables de l’espace de travail dans le fichier nom-fic.mat. Si aucun nom de fichier n’est précisé, le fichier par défaut est matlab.mat.

Pour ramener dans l’espace de travail les variables sauvegardées dans le fichier nom-fic.mat, taper **load nom-fic**.

**Exemple :**

>> x=2\*pi/3, y=sin(x), z=cos(x)

x = 2.0944 y = 0.8660 z = -0.5000

>> save nom y z

>> load nom

**I.5.4) Obtenir de l’aide dans une session matlab**,

* **FONCTION "HELP"**

Pour obtenir de l'aide sur un sujet, une instruction ou une fonction, on tape help suivi par le

sujet, l'instruction ou la fonction désirée.

**I.5.5) Opérations mathématiques :**

**a) opérations arithmétiques**

+ Addition ; - Soustraction ; \* Multiplication ; / Division à droite ; \ Division à gauche

^ Puissance

**b) Vecteurs et matrices**

**- VECTEURS**

On peut définir un vecteur x en donnant la liste de ses éléments:

>> x=[0.5 1.2 -3.75 5.82 -0.735]

x =

0.5000 1.2000 -3.7500 5.8200 -0.7350

ou en donnant la suite qui forme le vecteur:

>> x=2:0.6:5

x =

2.0000 2.6000 3.2000 3.8000 4.4000 5.0000

ou en utilisant une fonction qui génère un vecteur:

>> x=linspace(1,10,6)

x =

1.0000 2.8000 4.6000 6.4000 8.2000 10.0000

ou:

>> y=logspace(1,3,7)

y =

1.0e+003 \*

0.0100 0.0215 0.0464 0.1000 0.2154 0.4642 1.0000

**- MATRICES**

On définit une matrice A en donnant ses éléments:

>> A=[0.5 2.7 3.9;4.5 0.85 -1.23;-5.12 2.47 9.03]

A =

0.5000 2.7000 3.9000

4.5000 0.8500 -1.2300

-5.1200 2.4700 9.0300

Matrice unitaire:

>> B=eye(4)

B =

1 0 0 0

0 1 0 0

0 0 1 0

0 0 0 1

* **EMPLOI DES INDICES**

Les éléments d'un vecteur ou d'une matrice peuvent être adressés en utilisant les indices sous

la forme suivante:

t(10) élément no. 10 du vecteur t

A(2,9) élément se trouvant à ligne 2, colonne 9 de la matrice A

B(:,7) la colonne 7 de la matrice B

C(3,:) la ligne 3 de la matrice B

**I.5.6) Opérations matricielles :**

Les opérations matricielles exécutées par MATLAB sont illustrées dans le tableau suivant:

B = A' La matrice B est égale à la matrice A transposée

E = inv(A) La matrice E est égale à la matrice A inversée

C = A + B Addition

D = A - B Soustraction

Z = X\*Y Multiplication

X = A\B Équivalent à inv(A)\*B

X = B/A Équivalent à B\*inv(A)

**Les opération «élément par élément»**

Les opérations «élément par élément» des vecteurs et des matrices sont effectuées en ajoutant

un point (.) avant les opérations \* / \ ^ '

**exemple 2:**

>> A=[1 2 3 4 5];

>> B=[6 7 8 9 10];

>> C=A.\*B

C =

6 14 24 36 50

>> D=A./B

D =

0.1667 0.2857 0.3750 0.4444 0.5000

**I.5.7) Variables et fonctions**

**- VARIABLES**

On définit une variable en donnant son nom et sa valeur numérique ou son expression mathé-matique :

a =1.25;

x = 0:0.5:10;

y = a\*x;

z = y.^2;

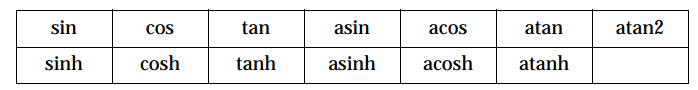
* **EXPRESSIONS MATHÉMATIQUES**

On écrit les expressions mathématiques de la façon habituelle:

z = 5\*exp(-0.4\*x).\*sin(7.5\*y);

* **FONCTIONS MATHÉMATIQUES**

****

****

**Exemple:**

>> x=-2+5i

x =

-2.0000 + 5.0000i

>> a=real(x)

a =

-2

>> b=imag(x)

b =

5

>> X=abs(x)

X =

5.3852

>> alfa=angle(x)

alfa =

1.9513

**I.5.8) Programmation et utilisation des fonctions**

Il est possible d’enregistrer une séquence d’instructions dans un fichier (appelé un « M-file ») et de les faire exécuter par matlab. Un tel fichier doit obligatoirement avoir une extension de la forme .m (d’où le nom M-file) pour être considéré par matlab comme un fichier d’instructions. On distingue 2 types de M-file, les fichiers de scripts et les fichiers de fonctions.

Pour que Matlab retrouve vos scripts, il met à votre disposition plusieurs commandes d’environnement:

– **path** : permet de savoir quels sont les dossiers auxquels Matlab a accès et de spécifier de nouveaux dossiers où se trouvent vos ressources personnelles. Pour référencer un nouveau dossier, taper : **addpath /mesfichiersmatlab** ce qui indique à Matlab qu’il peut trouver des scripts dans le dossier /mesfichiersmatlab durant la session en cours.

**– cd :** positionne Matlab dans un dossier, par exemple : **cd /mesfichiersmatlab**. Matlab utilise en priorité les scripts se trouvant dans le dossier courant.

– **dir** ou **ls** permet d’avoir la liste des fichiers du répertoire courant.

**- Scripts:**

Un script est un simple fichier texte avec l’extension .m qui contient une suite de commandes Matlab. Ce fichier peut être créé avec n’importe quel éditeur de texte et doit être placé dans le répertoire courant ou dans un répertoire du path. Matlab contient un éditeur de script intégré que vous pouvez utiliser

Par exemple, créez un fichier premierscript.m dans le répertoire courant qui contient les lignes suivantes :

A=2\*eye(4);

B=4\*ones(4);

A\*B

Vous pouvez ensuite exécuter ce script en entrant son nom sur la ligne de commande

>> premierscript

ans = 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8

Notez qu’une ligne de commentaire commençant par % est ignorée dans les scripts.

**I.5.9) Graphiques**

**A) Graphiques 2D**

On utilise l'instruction plot pour tracer un graphique 2D:

**plot(x,y)** Tracer le vecteur y en fonction du vecteur x

**plot(t,x,t,y,t,z)** Tracer x(t), y(t) et z(t) sur le même graphique

**plot(t,z,'r--')** Tracer z(t) en trait pointillé rouge

**On peut choisir le format du graphique:**

**plot(x,y)** Tracer y(x) avec échelles linéaires

**semilogx(f,A)** Tracer A(f) avec échelle log(f)

**semilogy(w,B)** Tracer B(w) avec échelle log(B)

**polar(theta,r**) Tracer r(theta) en coordonnées polaires

**bar(x,y)** Tracer y(x) sous forme des barres

**grid** Ajouter une grille

**Exemple**

|  |  |
| --- | --- |
| t=0:0.01e-3:0.06;  y=10\*exp(-60\*t).\*cos(120\*pi\*t);  z=10\*exp(-60\*t).\*sin(120\*pi\*t);  a=10\*exp(-60\*t);  plot(t,y,'r',t,z,'g', t,a,'b--', t,-a,'b--'),grid  title('Fonctions sinusoidales amorties')  xlabel('Temps , s'),ylabel('Tension , V') |  |

**B) Graphique multiple**

On peut tracer plusieurs graphiques dans la même fenêtre en utilisant l’instruction **subplot** pour diviser la fenêtre en plusieurs parties.

|  |  |
| --- | --- |
| w=logspace(0,3,1000);  s=j\*w;  H=225./(s.\*s+3\*s+225);  AdB=20\*log10(abs(H));  phase=angle(H)\*(180/pi);  subplot(2,1,1),semilogx(w,AdB),grid  xlabel('w , rad/s'),ylabel('Amplitude , dB')  subplot(2,1,2),semilogx(w,phase),grid  xlabel('w , rad/s'),ylabel('Phase , degre') |  |

* **Ajout du texte au graphique**

title('Titre du graphique') Donner un titre au graphique

xlabel('Temps') Étiquette de l'axe x

ylabel('Tension') Étiquette de l'axe y

gtext('Valeur absolue') Ajouter du texte au graphique avec la souris

* **Manipulation de graphiques**

axis([-1 5 -10 10]) Choix des échelles x = (-1,5) et y = (-10,10)

hold Garder le graphique sur l'écran (pour tracer plusieurs courbes sur le

même graphique)

* **Impression et enregistrement de graphiques**

print -dps Imprimer le graphique en PostScript

print -dpsc Imprimer le graphique en PostScript Couleur

print -dps dessin.ps Enregistrer le graphique en PostScript dans le fichier dessin.ps

**C) Graphiques 3D**

|  |  |
| --- | --- |
| t=0:0.05:25  x=exp(-0.05\*t).\*cos(t)  y=exp(-0.05\*t).\*sin(t)  z=t  plot3(x,y,z),grid |  |
| [X,Y] = meshgrid(-2:.1:2, -2:.1:2);  Z = X .\* exp(-X.^2 - Y.^2);  surf(X,Y,Z) |  |

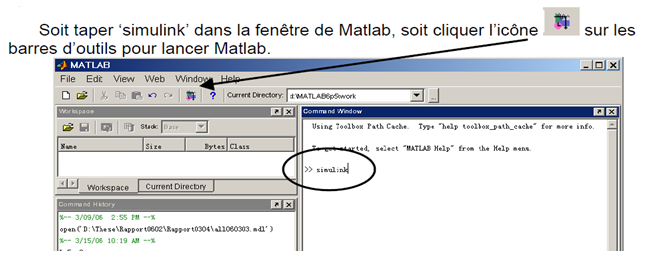
**I.6 Simulink**

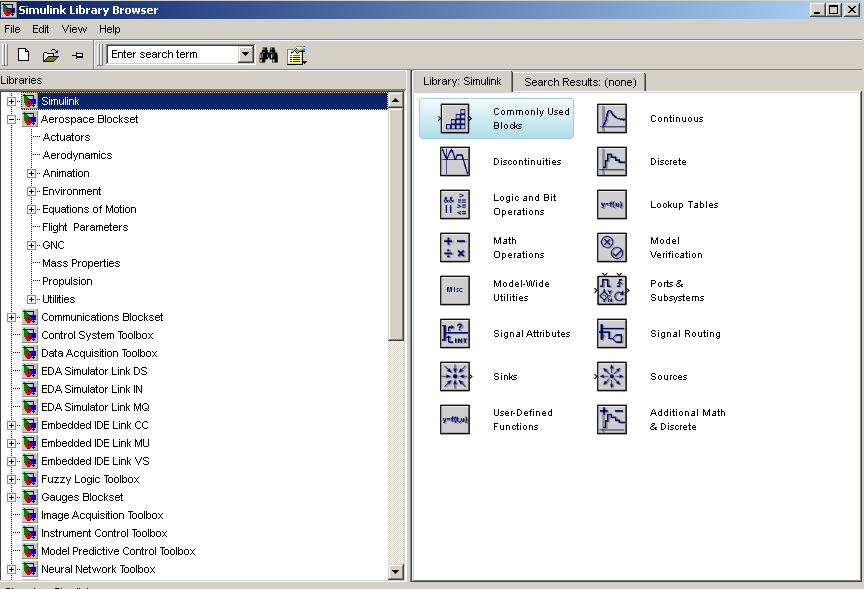
**I.6.1 Définition**

Simulink est l'extension graphique de MATLAB permettant de représenter les fonctions mathématiques et les systèmes sous forme de diagramme en blocs, et de simuler le fonctionnement de ces systèmes.

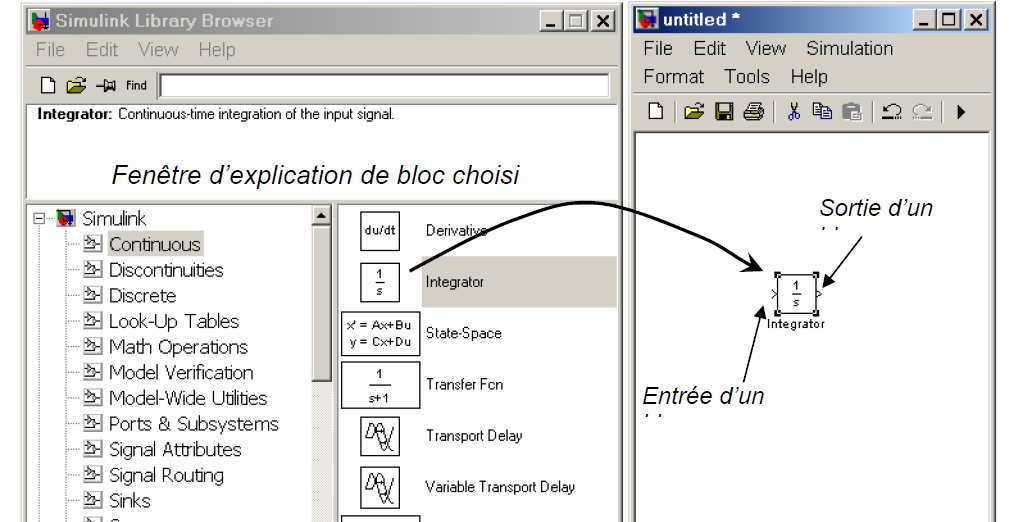
**I.6.2 Construction d'un diagramme simulink**

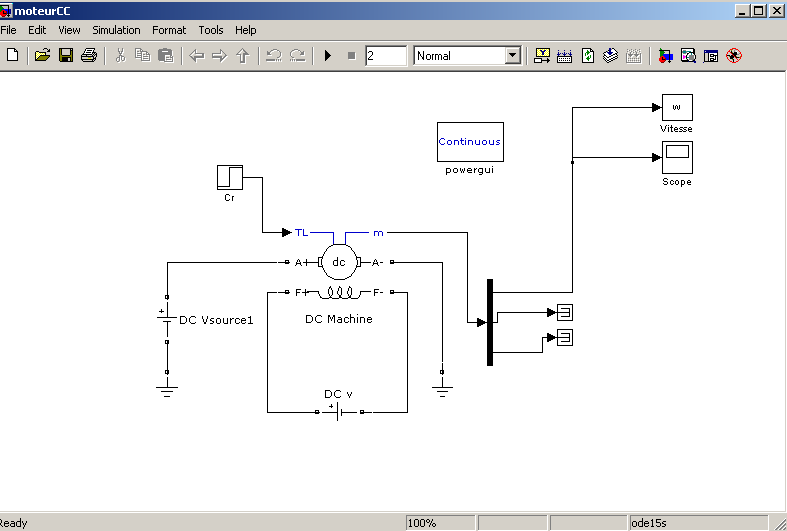
1. **Lancer Simulink**

**

**

1. **Créer un nouveau fichier**
2. **Manipulation des blocs**: ouvrir les collections de blocs en cliquant dessus (double). Faire glisser dans la fenêtre de travail les blocs dont on a besoin pour construire le diagramme. Faire des liaisons entre les blocs à l'aide de la souris.



**

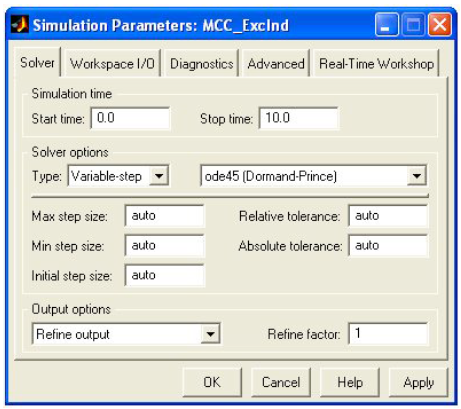
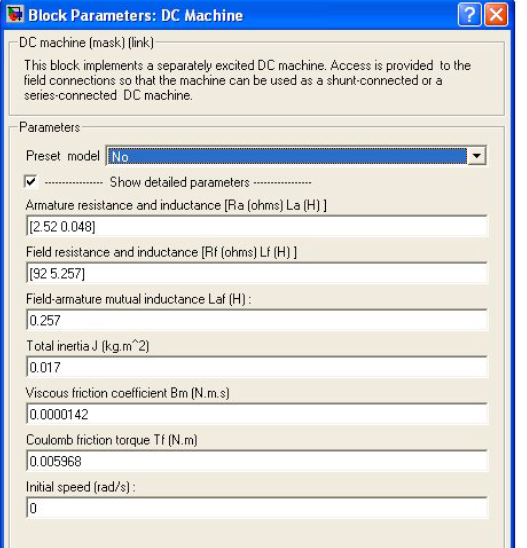
1. **Réglage des paramètres**

Avant de lancer la simulation il faut spécifier les paramètres de la simulation comme indiqué dans la figure suivante

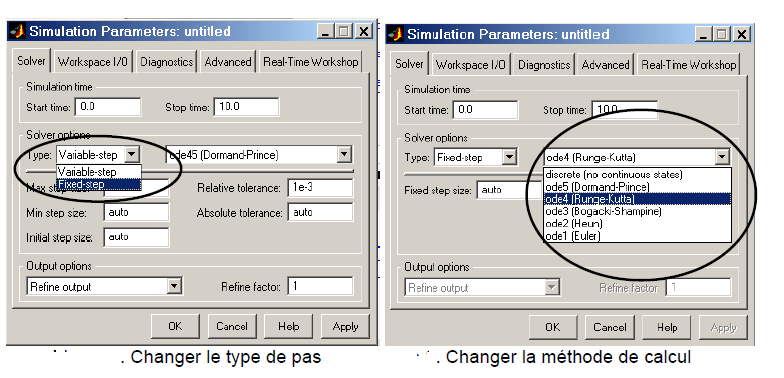


On doit aussi choisir les paramètres appropriés au modèle du système et les paramètres de simulation

Lorsqu'on clique (double) sur un bloc, une fenêtre de dialogue s'ouvrira. On peut alors changer les paramètres de ce bloc. Une fois terminé, on ferme la fenêtre de dialogue.

****

**Fenêtre de réglage des paramètres du bloc Fenêtre de réglage des paramètres de la modélisation**

****