Simulation sur Scilab

Travail réalisé par :

ZAKI Khaoula

ELMAHSSANI Widad

ELBERRY Fedoua

MAHER Maha

Présenté à :

ELMEROUANI Mohammed

Master GIE

Année universitaire : 2011-2012

Présentation de Scilab

Scilab est un progiciel scientifique pour les calculs numériques dans un environnement convivial.

Il comporte :

Des structures élaborées de données (polynômes, chaîne de caractères, listes, systèmes linéaires

Multivariables, ...).

Un interprète pour un langage de programmation avec un syntaxe proche de Matlab.

Des centaines de fonctions mathématiques intégrées (de nouvelles primitives peuvent facilement être ajoutées).

Des multiples fonctions graphiques (2d, 3d, animation).

Un logiciel ouvert (interface facile avec Fortran et C par lien dynamique en ligne).

De nombreuses bibliothèques intégrées :

Algèbre linéaire (y compris matrices creuses, forme de Kronecker, forme de Schur ordonnée).

Automatique, commande (classique, LQG, ...).

Module pour l'optimisation LMI (inégalités linéaires des matrices).

Traitement du signal

Simulation (diverses variantes autour de solveurs d'équations différentielles, DASSL, ...).

Optimisation (différentiable et non différentiable, solveur LQ).

Scicos, un environnement interactif pour modéliser et simuler des systèmes dynamiques.

Metanet (édition, analyse et optimisation de graphes).

Une interface avec le logiciel symbolique Maple.

..etc.

Un "Progiciel" est un terme né de la contraction de produit et logiciel.

C'est en faite un logiciel applicatif, libre ou propriétaire, prêt-à-porter,

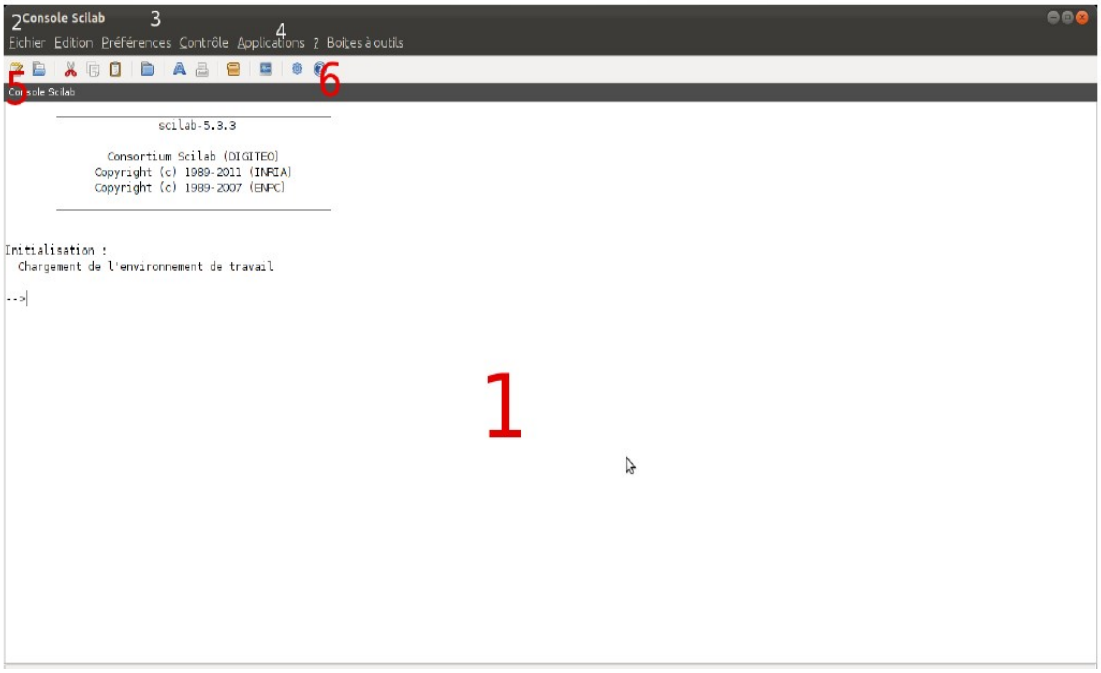
Standardisé et générique, prévu pour répondre à des besoins ordinaires.

Développé depuis 1998 par des chercheurs de l'INRIA (Institut National de Recherche en Informatique) et de l'ENPC (École National des Ponts et Chaussées), il est développé par le consortium

Scilab depuis mai 2003, consortium développé et maintenu par l'INRIA jusqu'en juillet 2008 puis depuis par la fondation de la coopération scientifique Digiteo.

Distribué gratuitement avec son code source via l'internet depuis 1994, il est disponible précompilé pour un grand nombre d'architectures. Néanmoins, il ne s'agissait ni d'un logiciel opensource (on donnera une signification à ce terme dans la suite) selon l'Open Source Initiative, ni d'un logiciel libre. En en et, l'ancienne licence Scilab n'autorise pas la distribution commerciale
d'une version modifiée. Selon la classification de la FSF (Free Software Foundation), il s'agissait donc plutôt d'un logiciel semi-libre. Scilab est donc devenu un logiciel libre lors du changement de licence : il est distribué sous la licence CeCILL (abréviation de CEA CNRS INRIA logiciel libre) depuis la version 5.0.

Figure n 1 : Fenêtre principale de Scilab

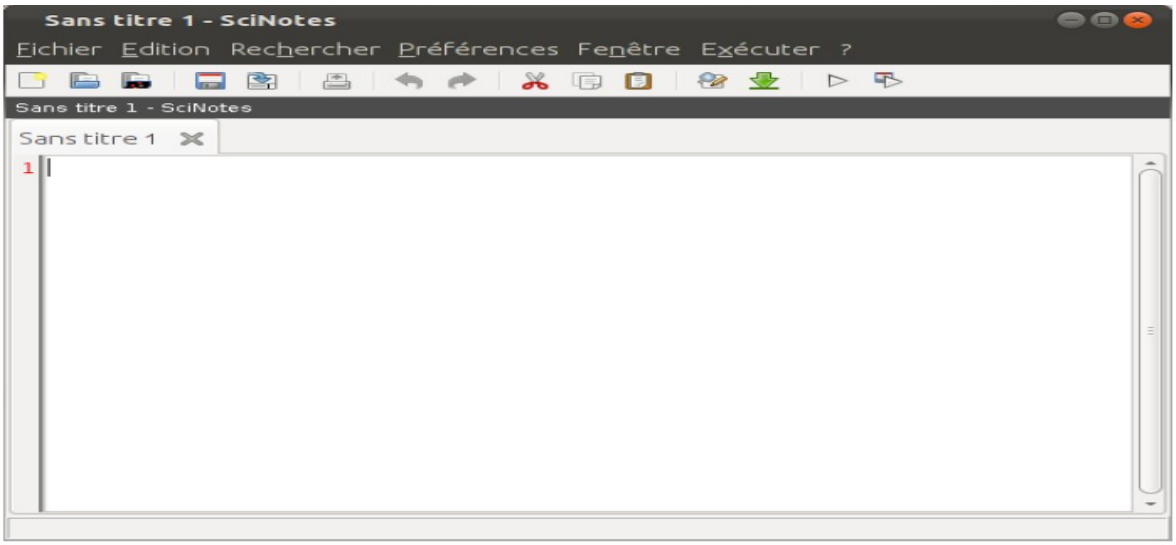


1. C'est la console Scilab, là ou on introduit nos commandes juste après le signe prompt __>_.
2. Le menu Fichier_ contient essentiellement les options d'exécution et de chargement de fichiers de commande, on peut aussi enregistrer notre travail ainsi que changer le répertoire de travail si on veut.
3. Le menu – Préférences- , sert surtout à régler et personnaliser le look général de Scilab (couleurs, polices, ...etc.)
4. Le menu – Application- celui qui nous intéresse le plus, contient notamment le module _ Xcos _
(On en a déjà parlé mais on ne l'utilisera pas), le convertisseur de fichier Matlab vers Scilab, et en n Scinotes l'application qu'on va utiliser le plus et qui contiendra nos commandes.
5. On peut directement démarrer _ SciNotes _ à partir de ce bouton.

6. Un _ navigateur help_ ou vous trouvez toute l'aide nécessaire sur le fonctionnement de Scilab.

Voilà, on a vu les principales caractéristiques de la fenêtre graphique Scilab. La fenêtre _SciNote_ se présente comme suit :

Figure n 2 éditeur SciNotes



En n, pour terminer notre présentation de Scilab, il convient de donner les raisons qui nous ont poussés vers le choix de ce logiciel, et pourquoi on l'a préféré à Matlab. Ces fameuses raisons on peut les résumer dans les 3 points suivants :

_ Scilab est «’open source ». Cette désignation s'applique aux logiciels dont la licence respecte des critères précisément établis par l'Open Source Initiative, c'est-à-dire la possibilité de libre redistribution, d'accès au code source et aux travaux dérivés, et tout ceci techniquement et légalement (donc un logiciel gratuit).

_ Matlab et un logiciel propriétaire et payant. C'est tout le contraire de l'open source (Malgré qu'on le trouve en téléchargement gratuit sur le net, mais cette pratique est normalement

_ illégale_ et n'offre pas un - Matlab d'origine_ comme celui qu'on achète !).

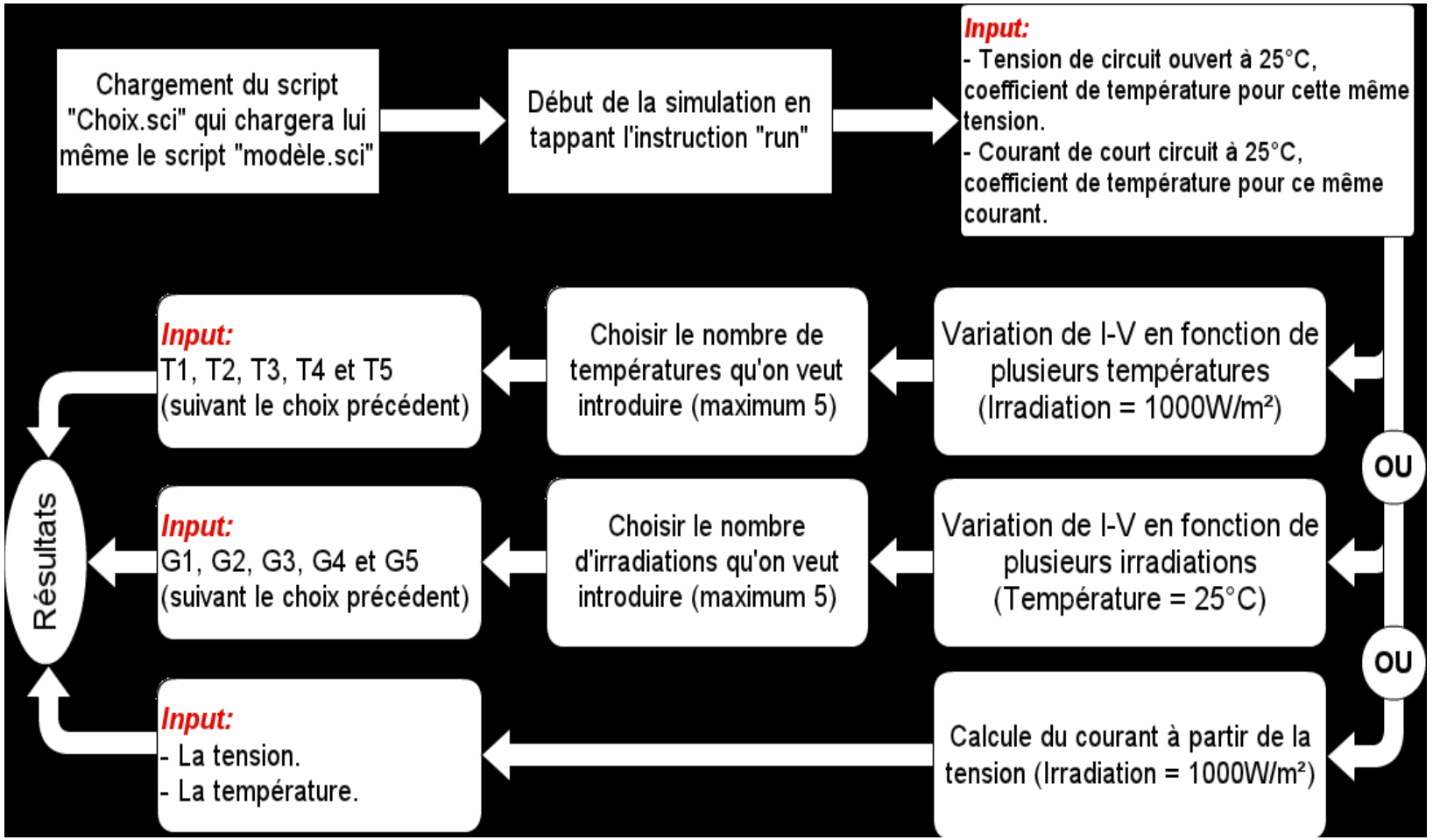
_ En fin, Scilab, au contraire de Matlab, est disponible pour Windows, Mac OS X, Linux et BSD.

Il est très léger (il ne prend pas beaucoup de place) et très performant (au même niveau que Matlab). Donc, c'est le candidat idéal pour les simulations scientifiques à moindre coût.

Présentation du script de simulation

Notre simulation se base essentiellement sur deux scripts contenant chacun une fonction bien précise. Le premier script, qu'on a appelé _ Choix _, donnera à l'utilisateur, comme son nom

l'indique, la possibilité de choisir entre différentes options fournies par le programme de simulation. Ce premier script est aussi très important car c'est lui qui donnera à l'utilisateur la possibilité de
rentrer les valeurs essentielles au bon fonctionnement de la simulation (Vco , Icc , nombre de cellules, ...etc).
Le deuxième script, nommé _ modele _ (voir annexe 2), contiendra la fonction essentielle qui calcule le courant délivré par le module à partir de trois données (fournies par l'utilisateur) :
_ Tension délivrée.
_ Irradiation (1 correspond à 1000W:m□2).
_ Température.
Avant de passer aux étapes de fonctionnement du programme, il faut dire que le but _nal sera de trouver le courant de sortie du module. Cela se fait en cherchant la racine de l'équation 1, et dans notre programme on a utilisé une méthode numérique, qui n'est d'autre que la méthode de
Newton-Raphson (voir annexe 1), pour y arriver.
Ainsi, d'une manière globale, les étapes de fonctionnement de notre programme de simulation sont comme suit :
Étapes de fonctionnement du programme de simulation.

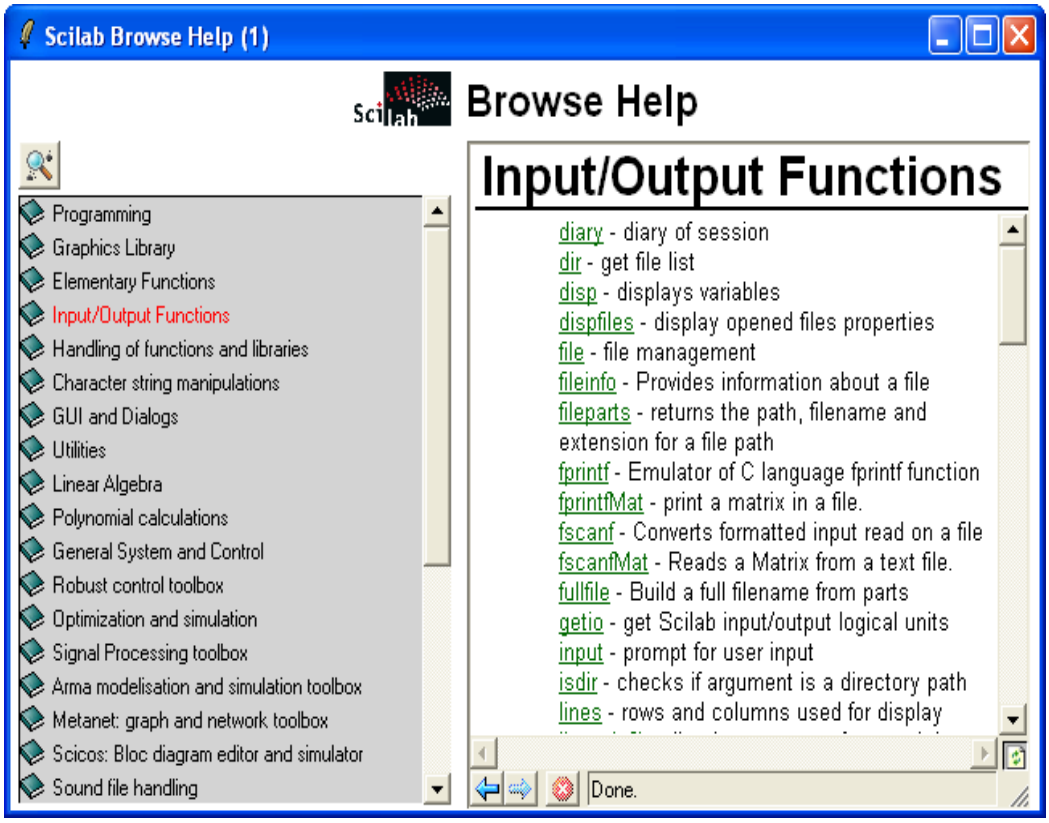

Chargement du script "Choix.sci" qui chargera lui même le script "modèle.sci"
Début de la simulation en tappant l'instruction "run"
Input:
- Tension de circuit ouvert à 25°C, coefficient de température pour cette même tension.
- Courant de court circuit à 25°C, coefficient de température pour ce même courant.
Variation de I-V en fonction de plusieurs températures (Irradiation = 1000W/m²)
Choisir le nombre de températures qu'on veut introduire (maximum 5)
Input:
T1, T2, T3, T4 et T5 (suivant le choix précédent)
OU
Variation de I-V en fonction de plusieurs irradiations (Température = 25°C)
Choisir le nombre d'irradiations qu'on veut introduire (maximum 5)
Input:
G1, G2, G3, G4 et G5 (suivant le choix précédent)
OU
Calcule du courant à partir de la tension (Irradiation = 1000W/m²)
Input:
- La tension.
- La température.
Résultats

Quelques instructions utiles

- size : taille d'une matrice
- for … end : boucle for
- if .. then … else : condition
- find : identifier des éléments dans une matrice
- rand : générer des nombres aléatoires
- Grand : générer des nombres pesudo aléatoires
- plot2d : Génération d'un graphique
- cdfbin: fonction de répartition de la distribution binomiale
- Cdfnor: fonction de répartition de la distribution normale
- Cdfpoi:fonction de répartition de la distribution de Poisson
- jouter des commentaires → //
- Fin de vecteur → $
- Matrice → [ ]

Pour plus d'information taper help sur Scilab

Figure



Etude de cas 1:

- **a société x fabrique des plaques de plâtre**

sur rapport cartonné destinées à la construction préfabriqué .

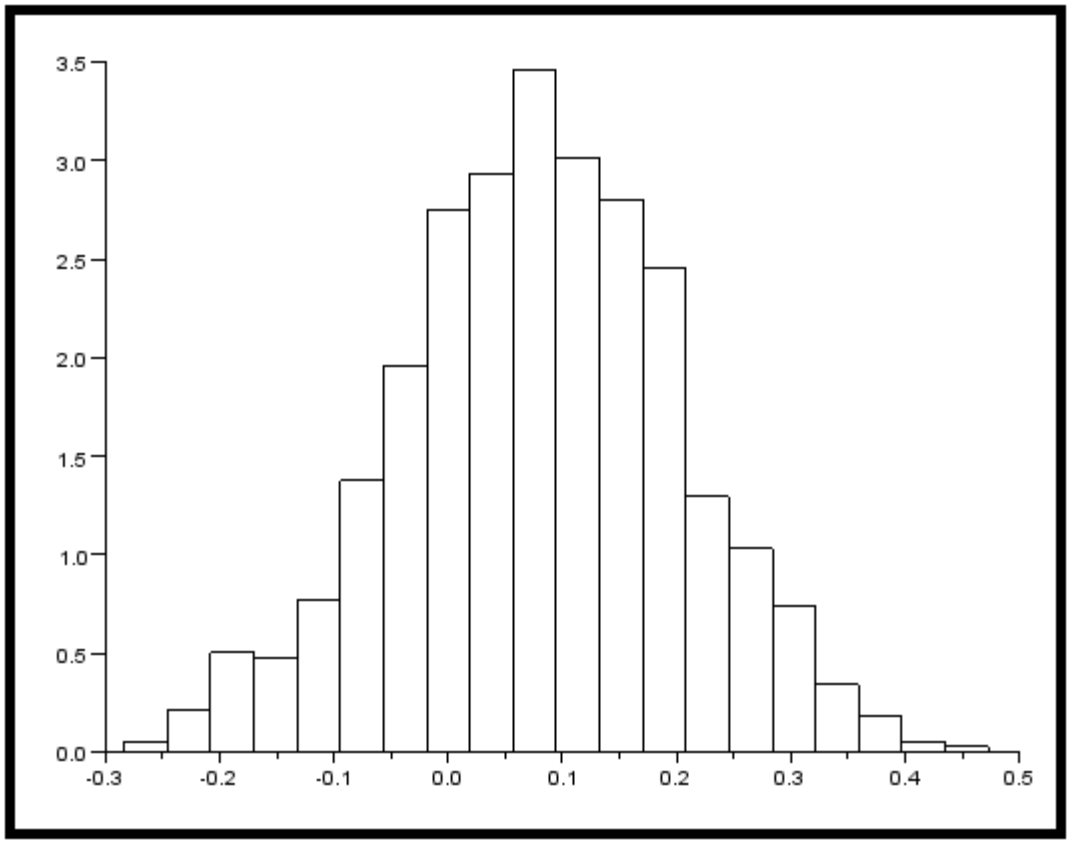
La production fortement mécanisée est organisée de façon a ce que les périodes de congé, même d'été ne la perturbent pas on peut considérer que cette production s'étend au cours de l'année que 50 semaines un état de la production en quantité est établi chaque semaine. Depuis trois ans la production annuelle est restée sensiblement constate, et une étude statistique a montré que le chiffre d'affaires hebdomadaires de la production peuvent être considérés comme régis par une loi normale (gaussienne) de moyenne arithmétique 0.08 et d'écart type de 0.12

1. **Générer des variables pseudo_aléatoires par la loi normale dans scilab sur un échantillon de 1000**

```
Y = grand(n,1,"nor",0.08,0.12)
```

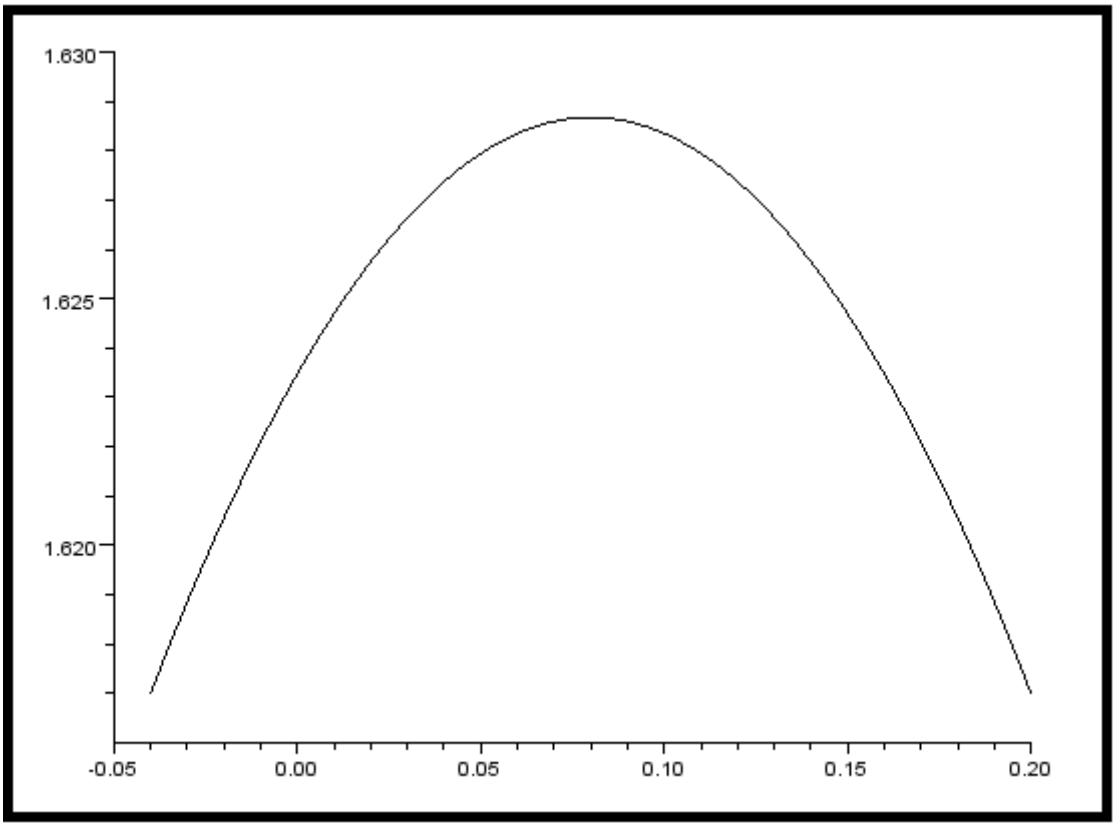
2. **Représenter graphiquement la loi normale sur un échantillon de 1000 (histogramme)**

```
Y=grand(n,1,'nor',μ,s)
```



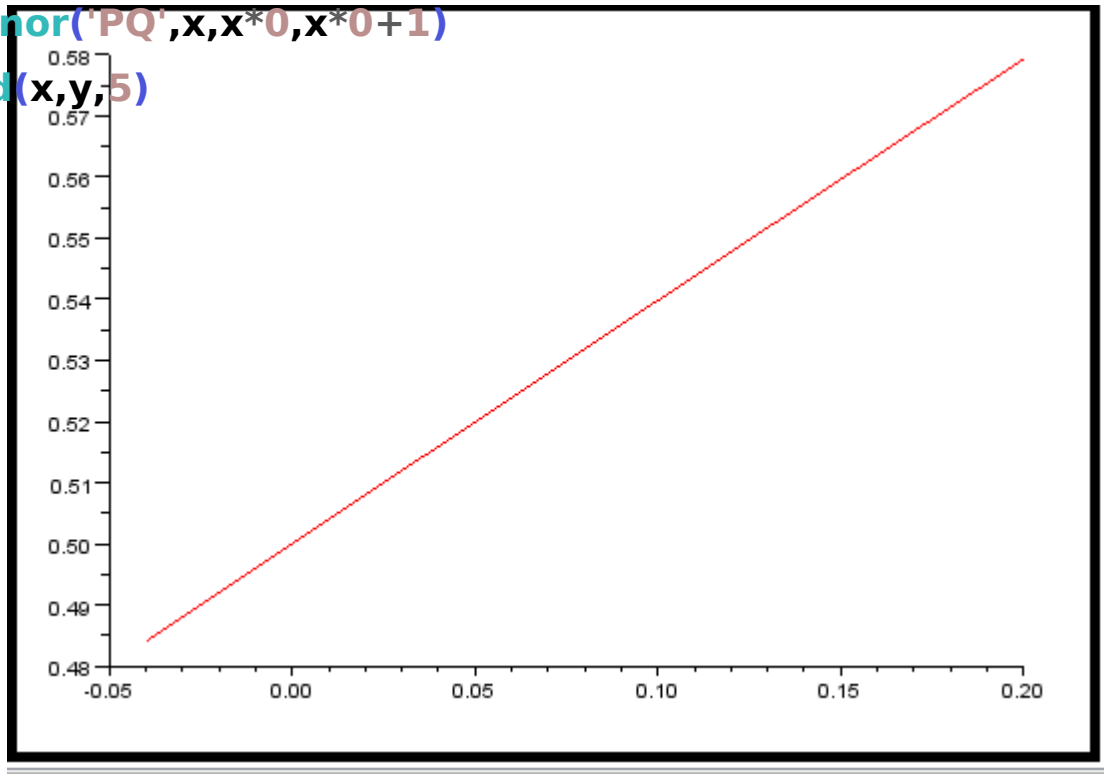
3. Construire la fonction de densité de la loi normale de paramètres μ =0.08 et σ 2 =0.12

```
x=linspace(μ-s,μ+s,1000)
y=exp(-(x-μ)^2/2)/sqrt(s*%pi)
plot2d(x,y)
```



4. Construire la fonction de répartition empirique de la loi normale centrée réduite avec 1000

```
x= grand(1,1000, 'nor', μ,s)
x=linspace(μ-s, μ+s,100)
y=cdfnor('PQ',x,x*0,x*0+1)
plot2d(x,y,5)
```



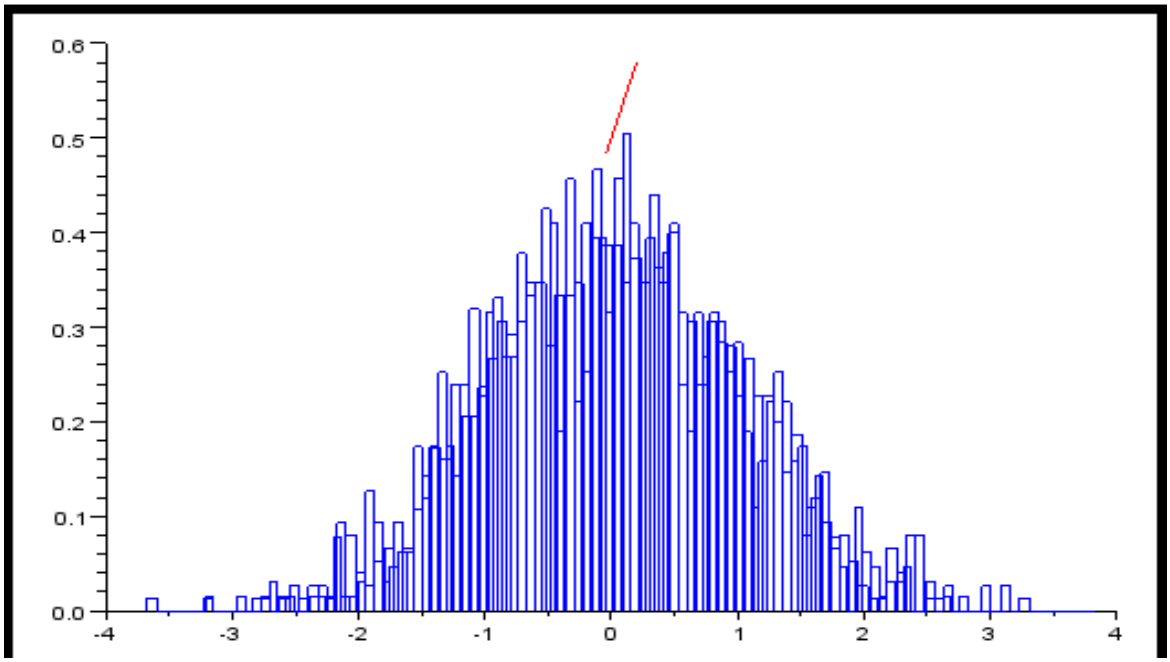
5. La fonction BOX MULLER :

```
S=1000;

x=zeros(1:S);
y=rand(S,1);
z=rand(S,1);
x=sqrt(-2*log(y)).*cos((2*%pi).*z);

w=sqrt(-2*log(y)).*sin((2*%pi).*z);

 histplot(100,x,style=2) ;
```



Etude de cas 2 :

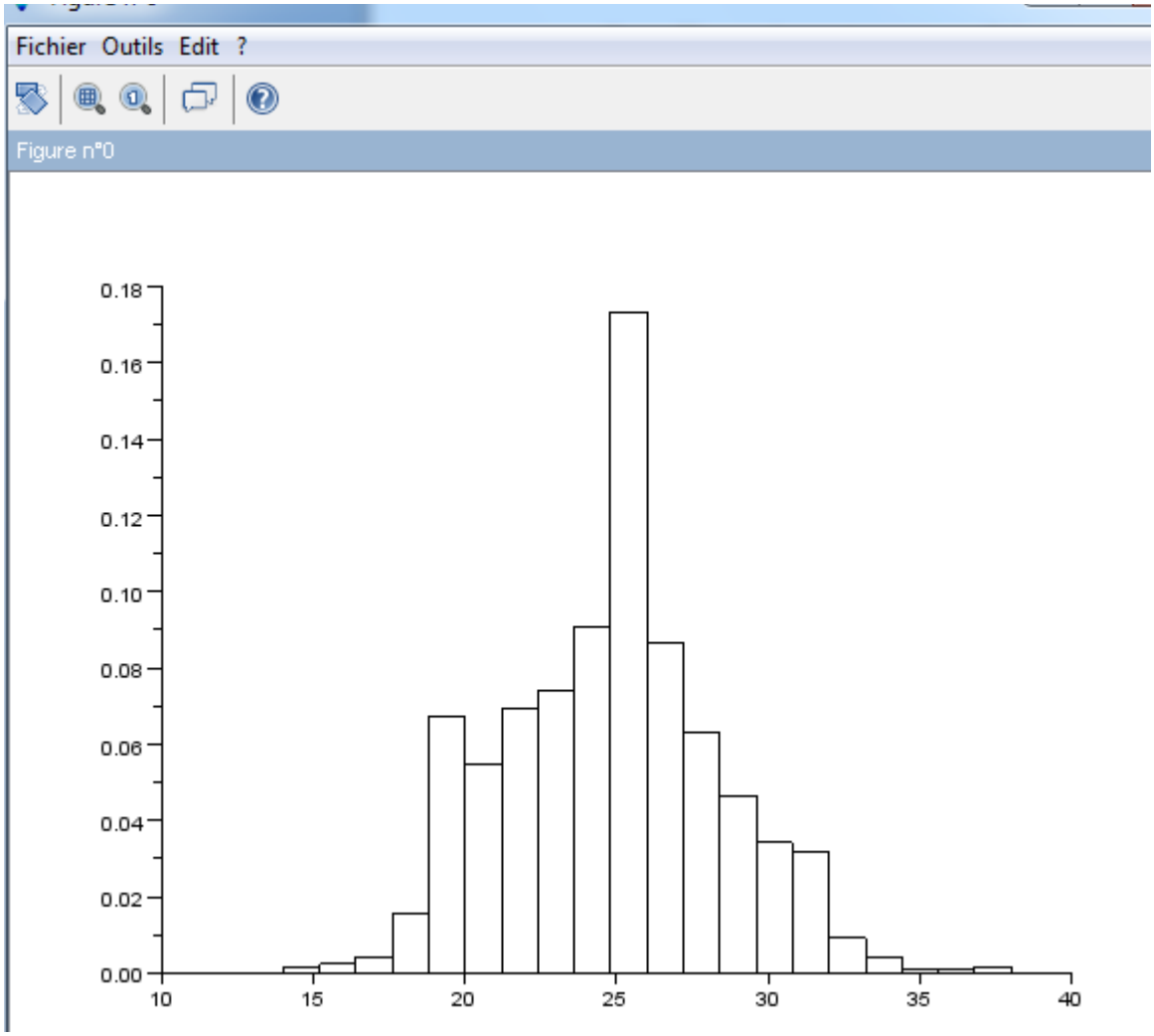
Un jeu televisé consiste à l'enoncé de 50 questions auquels le candidat repond par « oui » ou « non » , l'une des deux reponses étant juste et l'autre fausse (on va simulater pour 1000 fois) n=50 ; p =1/2 ; q=1/2

On va générer les valeurs possibles que x peut prendre à l'aide de la commande :

Grand(1000,1, "bin",50,0.5)

Représenter graphiquement la loi binomiale sur un échantillon de 1000 (histogramme)

```
Y=grand(1000,1,'bin',50,0.5)
clf();histplot(20,Y)
```



Etude de cas 3

Loi de poisson

Un industriel fabrique des pièces métalliques ; le soin apporté à la fabrication est tel que, en moyenne, le nombre de pièces à rebuter n'est que de 5 pièces sur 1000 pièces contrôlées une série de pièces a été produite ; 800 De ces pièces,prises au hasard vont être contrôlées.

- Génère des nombres aléatoires suivant la loi de Poisson de moyenne mu (réel >= 0.0).

X=grand (1000,1,'poi',4)

- Variations sur l'histogramme d'un échantillon de loi de poisson N(0,1)

clf();histplot(20,x) avec 20 nombre de classe

clf();histplot(20,x,leg='rand(1,10000,"poi")',style=5)

clf();histplot(20,x,leg='rand(1,10000,"poi")',style=16, rect=[-3,0,3,0.5])

P,Q,S,Xlam

4 vecteurs réels de même taille.

P,Q (Q=1-P)

La somme de 0 à S de la densité de Poisson. En entrée : [0,1].

S

Borne supérieure de la somme. En entrée : [0, +infini). En recherche : [0,1E300]

Xlam

Moyenne de la distribution. En entrée : [0, +infini). En recherche : [0,1E300]