



Nom :		Prénom :		Classe :			
Informatique et Objets Numériques - Géolocalisation							
TD	TP	Devoir	Cours	Evaluation	Document ressource	Document constructeur	Dossier technique

UTILISATION DU LOGICIEL SCILAB

Le logiciel **Scilab** est un logiciel de **calcul numérique**, qui permet de manipuler la plupart des objets mathématiques courants (nombres réels, booléens, matrices, fonctions, polynômes, graphes, ...).

Techniquement, **Scilab** est un interpréteur de commandes. Il se présente donc comme une super calculatrice : les commandes, tapées ou lancées à partir d'un programme, sont traduites dans un autre langage, compilées, puis exécutées par le noyau du système d'exploitation, le résultat (ou l'erreur) est récupérée par **Scilab** et affichée dans la console.

Lorsque vous démarrez **Scilab** la fenêtre principale s'ouvre sur la console. Un curseur apparaît juste après l'invite des commandes (-->). C'est à cet endroit que vous pourrez lancer les lignes de commandes qui seront exécutées séquentiellement, de la même manière qu'avec une calculatrice.

```

Console
Fichier Edit Préférences Contrôle Applications ?
Console
-----
scilab-5.1.1

Consortium Scilab (DIGITEO)
Copyright (c) 1989-2009 (INRIA)
Copyright (c) 1989-2007 (ENPC)
-----

Initialisation:
Chargement de l'environnement de travail

-->|

```

Quelques règles pour utiliser la console :

- Chaque commande **Scilab** doit se terminer par un retour à la ligne (↵).
- Une commande peut aussi se terminer par une virgule (,) afin de mettre plusieurs commandes par ligne.
- Elle peut également se terminer par un point virgule (;) qui permet aussi d'exécuter plusieurs commandes par ligne tout en bloquant l'affichage du résultat précédant le point virgule. Il est très utile pour masquer le résultat d'un calcul intermédiaire.
- Il est possible de **copier/coller** des instructions avec la souris (mais pas modifier une ligne de commande déjà exécutée).
- On peut **rappeler les commandes** précédemment exécutée en utilisant les touches ↑ et ↓ du clavier.
- Dans une ligne tout ce qui suit un double slash (//) est ignoré (ce qui permet d'insérer du commentaire).
- help commande** permet connaître le rôle d'une commande dont vous connaissez déjà le nom.



2. CALCULS NUMERIQUES SIMPLES

Dans **Scilab** tous les calculs sont numériques. Les nombres ont une précision maximale de **16 chiffres décimaux** et leur valeur absolue est comprise entre environ 10^{-308} et 10^{+308} .

1. Lancez le logiciel **Scilab**.  scilab-5.1.1

2. Tapez les commandes suivantes et commentez les résultats.

```
-->2+3
-->%pi
```

Par défaut les résultats sont affichés avec **10 caractères**, comprenant le point décimal et le signe. Si l'on veut plus de chiffres on utilise la fonction **format**.

3. Tapez les commandes suivantes et commentez les résultats.

```
-->format(20)
-->%pi
```

4. Tapez les commandes suivantes.

```
-->format(10)
-->sin(%pi/4)
-->sin(%pi)
```

5. Indiquez pourquoi n'obtient-on pas $\sin(\pi) = 0$.

6. Calculez $N = 1/3$. Notez la valeur qui s'affiche pour N . Recopiez cette valeur, multipliez-la par 3. Donnez le résultat. Calculez maintenant $N \times 3$. Concluez.

7. Tapez les commandes suivantes et commentez les résultats.

```
-->(1+sqrt(5))/2
-->(1+sqrt(5))/2;
```

8. Tapez les commandes suivantes et commentez les résultats.

```
-->-2^3+9
-->3/4*5-5^2*2-3
-->ans
-->//ans
```

3. LES MATRICES

Une matrice est un tableau le plus souvent à **2 dimensions**. Il s'agit de la structure de données la plus importante dans **Scilab**. Il est possible également de définir un vecteur (ligne ou colonne) qui se présente comme un tableau à une dimension. Dans **Scilab** toute donnée est une matrice (un tableau) : par exemple un **réel** est un **tableau de dimension 1×1** .



1. **Définissez** le vecteur colonne V .

```
-->V=[1;2;3]
```

2. **Définissez** le vecteur ligne W .

```
-->W=[4 5 6]
```

3. **Définissez** la matrice M .

```
-->M=[1 2 3;4 5 6;7 8 9]
```

4. **Tapez** la commande suivante. **Commentez** vos résultats.

```
-->M(2,3)
```

5. **Tapez** la commande suivante. **Commentez** vos résultats.

```
-->M(2,3)=23
```

6. **Tapez** les commandes suivantes. **Donnez** le rôle de la fonction `size`.

```
-->size(M)
```

```
-->size(V)
```

```
-->size(W)
```

7. **Définissez** les matrices A et B . Réalisez les opérations suivantes : $A + B$ et $A - B$. **Commentez** vos résultats.

```
-->A=[1 2;3 4]
```

```
-->B=[5 6;7 8]
```

8. **Tapez** les commandes suivantes. **Donnez** le rôle des fonctions `sum` et `max`.

```
-->sum(A)
```

```
-->max(M)
```

4. LES TRACES EN 2 DIMENSIONS

Scilab possède un grand nombre de commandes qui permettent de tracer des courbes, des nuages de points, des histogrammes, etc. et de modifier les caractéristiques des tracés : type du tracé, couleur, épaisseur, graduation des axes, etc.

1. **Définissez** un vecteur de **10 valeurs** régulièrement espacées entre -10 et $+10$.

```
-->x=linspace(-10,10,10);
```



2. **Tracez** la représentation graphique de l'équation $y = 2 \times x$.
-->plot(x,2*x)
3. **Tracez** la représentation graphique de l'équation $y = 10 + 2 \times x$. **Commentez** les graphes obtenus.
4. **Tapez** les commandes suivantes. **Donnez** le rôle de la commande *clf*.
-->clf;plot(x,3*x)
5. **Tapez** les commandes suivantes. **Commentez**.
-->clf;plot(x,2*x,x,10+2*x)
6. **Tapez** les commandes suivantes.
-->clf;plot(x,2*x,"+r",x,10+2*x,"*-g")
7. **Tracez** la représentation graphique de l'équation $y = 2 \times x$ en bleu avec une ligne continue et des points en forme de « * », et la représentation graphique de l'équation $y = 10 + 2 \times x$ en jaune avec uniquement des points en forme de « + ».

Soit l'équation affine par morceaux :

$$\begin{cases} x \in [0,2[\Rightarrow y = 4 \times x \\ x \in [2,4[\Rightarrow y = 4 + 2 \times x \\ x \in [4,6[\Rightarrow y = 8 + x \\ x \in [6,8[\Rightarrow y = 11 + 0.5 \times x \\ x \in [8,10[\Rightarrow y = 13 + 0.25 \times x \end{cases}$$

8. **Tracez** la représentation graphique de cette équation en trait plein (sans points de forme) avec les couleurs suivants : **vert** pour $0 \leq x < 2$, **rouge** pour $2 \leq x < 4$, **bleu** pour $4 \leq x < 6$, **magenta** pour $6 \leq x < 8$ et **cyan** pour $8 \leq x < 10$.

5. UTILISATION DU SCRIPT SCILAB

Afin d'exécuter une suite d'instructions plus longues ou plus complexes, il est préférable d'écrire celles-ci dans un fichier plutôt que de les taper dans la fenêtre de commande comme précédemment. On parle dans ce cas de **l'écriture d'un script**. Il est ensuite possible d'exécuter ce script en utilisant la commande **Exécuter** accessible dans le menu **Fichier** de la console.

1. **Lancez** l'éditeur de texte **SciPad** . **Editez** le texte ci-dessous puis **enregistrez** votre fichier sous le nom *essai.sce*.

```
x=linspace(0,10,10);
y=3+2*x
plot(x,y)
```



2. **Exécutez** le script précédent en allant dans le menu **fichier** et en utilisant la fonction **Exécuter**.

6. FONCTIONS SCILAB

Il est possible de construire avec **Scilab** de nouvelles fonctions mathématiques, en complément de celles existantes. Ces fonctions peuvent prendre divers types d'arguments (scalaire, matrice, etc...) et peuvent réaliser des opérations mathématiques plus ou moins complexes.

1. **Lancez** l'éditeur de texte **SciPad** . **Editez** le texte ci-dessous puis **enregistrez** votre fichier sous le nom fonction1.sci.

```
function [y]=f(x)
y=1+x^2
endfunction
```

2. **Exécutez** la fonction précédente en allant dans le menu **fichier** et en utilisant la fonction **Exécuter**.

3. **Tapez** la commande suivante. **Commentez** vos résultats.

```
-->f(2)
```

7. APPLICATIONS

1. **Editez** la fonction *dec.sci* qui permet de convertir en décimal un nombre binaire de **4 bits** (matrice 1×4). **Exécutez** votre fonction et **vérifiez** que les résultats sont corrects.
2. **Editez** la fonction *hex.sci* qui permet de convertir en hexadécimal un nombre binaire de **4 bits** (matrice 1×4). **Exécutez** votre fonction et **vérifiez** que les résultats sont corrects.