



جامعة عباس لغرور خنشلة
UNIVERSITE ABBES LAGHROUR KHENCHELA
ABBES LAGHROUR UNIVERSITY OF KHENCHELA

Outils de programmation pour les mathématiques 2

RAHAB Hichem

rahab_hichem @yahoo.fr

rahab.e-monsite.com

2017 /2018

Chapitre 1

Introduction au langage MATLAB

1.1 Rappel sur les langages de calculs scientifique

Un langage de calcul scientifique est un langage de programmation destiné à être utilisé par la communauté scientifique dans des calculs scientifiques complexes. Et à cette finalité, il est riche en terme fonctions et de bibliothèques facilitant la tâche d'un programmeur dans un domaine de recherche, qui n'a pas nécessairement des compétences de programmation avancées. On peut classer les langages de calculs scientifiques en langages compilé et langages interprété. Dans un langage de programmation interprété un programme supplémentaire (l'interpréteur) est nécessaire, celui-ci va générer l'exécutable des instructions et les exécuter au fur et à mesure de l'exécution du programme, donc on n'a pas dans ce cas un code exécutable complet, et à chaque fois on a besoin du code source initiale pour réexécuter le programme. Par contre un langage compilé va traduire (compilé) le programme en son intégralité vers un code exécutable qui peut être utilisé ultérieurement sans avoir besoin du code source initiale.

1.1.1 Langages compilés

On peut citer à titre d'explication quelques langages comiplés :

1. Fortran (FORmula TRANslator) est un langage compilé developpé pr IBM vers 1954, est le plus ancien langage de programmation de haut niveau, c'est un langage de programmation destiné principalement pour le calcul scientifique.
2. Langage C
3. Langage C++
4. etc ...

1.1.2 Langages interprétés

Il y a aussi une varité de langages interprétés dont :

1. Matlab C'est le langage qu'on va étudier le long de ce cours.
2. Le langage R est un langage interprété très adaptée au calcul scientifique et à la fouille de données. Il possède une large collection d'outils statistiques et graphiques, relayée par une communauté très active. R est à la fois un logiciel de statistique et un langage de programmation. R est un logiciel de traitement statistique des données.
3. Scilab (Scientific Laboratory) est un logiciel libre de calcul numérique multi-plateforme fournissant un environnement de calcul pour des applications scientifiques. Il possède un langage de programmation orienté calcul numérique de haut niveau. Il peut être utilisé pour le traitement du signal, l'analyse statistique, le traitement d'images, la modélisation et la simulation.¹ Scilab est disponible pour Windows, Mac OS X, GNU/Linux. La syntaxe

1. <https://fr.wikipedia.org/wiki/Scilab>

et les possibilités offertes par Scilab sont similaires à celles de Matlab.

4. etc ...

1.2 Le langage Matlab

MATLAB est un environnement de calcul numérique matriciel, il est basé sur le principe de matrice. Tous les types dans Matlab sont à la base des matrices, un scalaire est une matrice de dimension 1×1 , un vecteur est une matrice de $1 \times n$ ou $n \times 1$. Ce principe est primordial à comprendre pour pouvoir travailler avec Matlab. Matlab crée une variable lors de son affectation, de ce fait on n'a pas besoin de déclarer les variables avant leur utilisation.

MATLAB est un langage interprété qui s'exécute dans une fenêtre dite d'exécution. L'intérêt de Matlab tient, d'une part, à sa simplicité d'utilisation : pas de compilation, pas besoin de déclaration des variables utilisées, et d'autre part, à sa richesse fonctionnelle : arithmétique matricielle et nombreuses fonctions de haut niveau dans divers domaines (analyse numérique, statistique, représentation graphique, ...).

On peut utiliser Matlab en deux modes :

Mode ligne de commande

Le mode ligne de commande permet d'obtenir des résultats rapides qui ne sont pas sauvegardés, c'est-à-dire saisir des commandes dans la fenêtre et les exécutés au fur et à mesure.

Mode script

En écrivant dans des fichiers séparés (*.m) l'enchaînement des commandes, ces fichiers s'appellent des scripts et on les construit à l'aide de n'importe quel éditeur de texte. Le mode script ou le mode programmation, quant à lui, permet de développer des applications plus complexes, ainsi que les programmes sont sauvegarder pour faciliter une utilisation ultérieure.

1.3 Installation de Matlab

Bien qu'il y a plusieurs versions de Matlab, on a trouvé que la version 7.0 et la plus convenable pour notre TP, premièrement la taille du programme, moins de 700 Mo, est très acceptable pour la copie et l'installation, aussi son interface est basique et ça ce qu'on veut pour un cours d'initiation au langage. J'ai trouvé utile de présenter en premier l'installation du langage que je recommande à mes étudiants de faire sur leurs ordinateurs personnels.

Commençant avec la modification du thème vers "Windows classique" avec lequel travail cette version.

1. clique droit sur le bureau et choisir Personnaliser, voir Figure 1.1.
2. Choisir le thème Windows Classique, voir Figure 1.2.

Ensuite aller dans le répertoire Matlab7 . Et cliquer sur setup.exe (Figure 1.3)

Dans la Figure 1.4 L'installation commence :

Dans la Figure 1.5 on doit saisir les coordonnées de l'utilisateur et le numéro de série du produit :

Alors comme illustrer dans la Figure 1.6 on va dans le dossier Crack, et on lance le keygen comme dans Figure 1.7 :

Copier le numéro de série voir Figure 1.8

Coller le numéro de série dans l'emplacement spécifier, Figure 1.9

Accepter le termes de la licence pour continuer 1.10

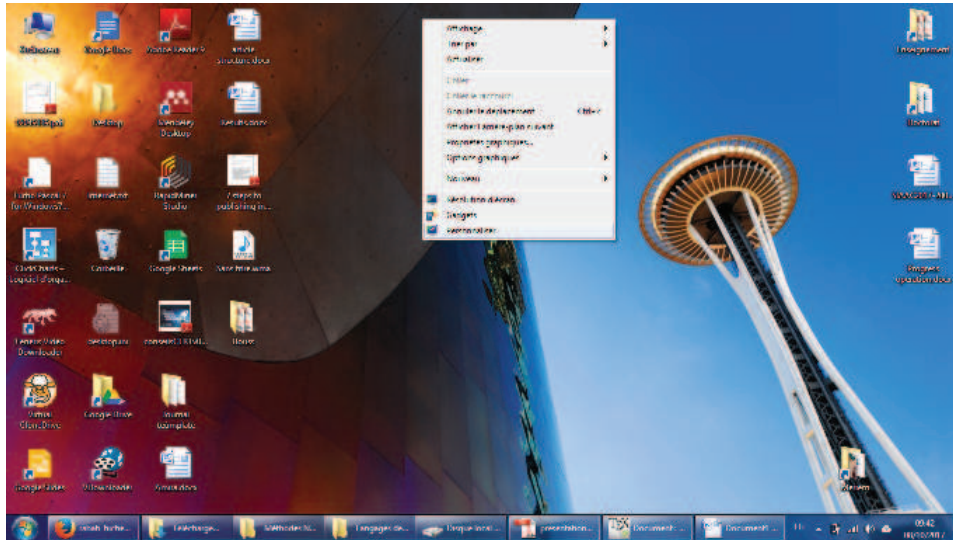


FIGURE 1.1 – Modification de thème windows

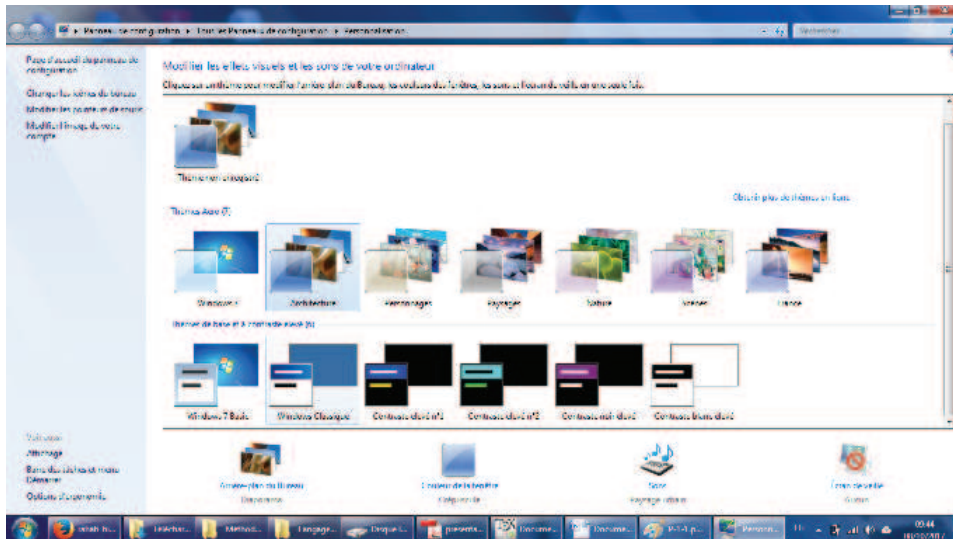


FIGURE 1.2 – Choisir le thème Windows Classique

Choisir installation typique, Figure 1.11

Choisir l'emplacement de l'installation, Figure 1.12

Dans la Figure 1.13 confirmer les options d'installation choisies. Et cliquer Instal.

Dans la Figure 1.15 le CD-2 est demandé alors qu'il n'est pas disponible, donc on clique sur Skip CD-2.

Et on poursuit dans Figure 1.16.

L'installation se termine en Figure ??.

et le programme Matlab est lancer comme illustrer dans Figure 1.17.

1.4 Les commandes Matlab

Matlab est très riche en nombre et qualité des commandes qu'il offre, on peut donner à titre d'illustration les commandes suivantes :

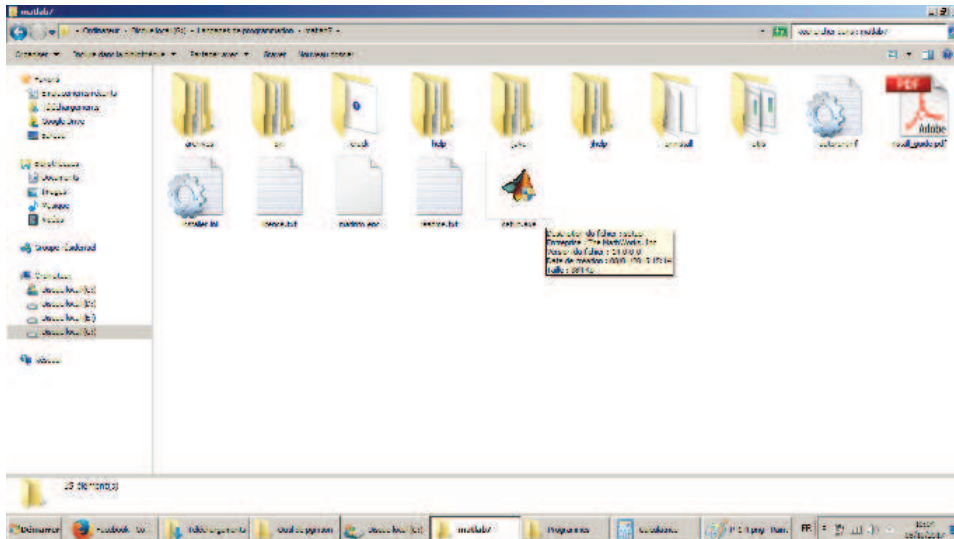


FIGURE 1.3 – Cliquer sur setup.exe

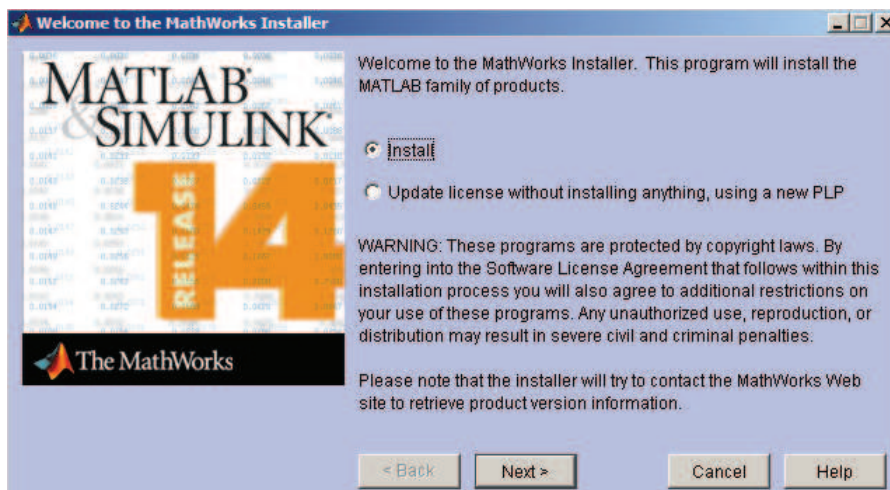


FIGURE 1.4 – L'installation commence

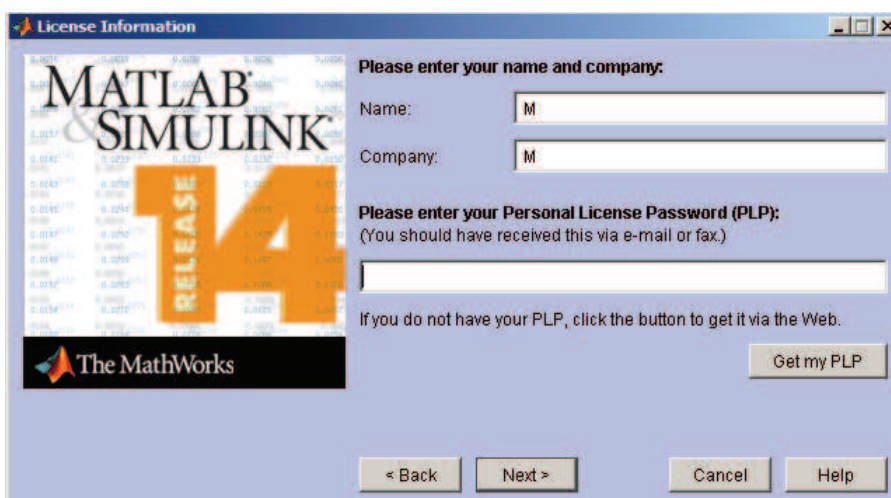


FIGURE 1.5 – Demande de numérod de série

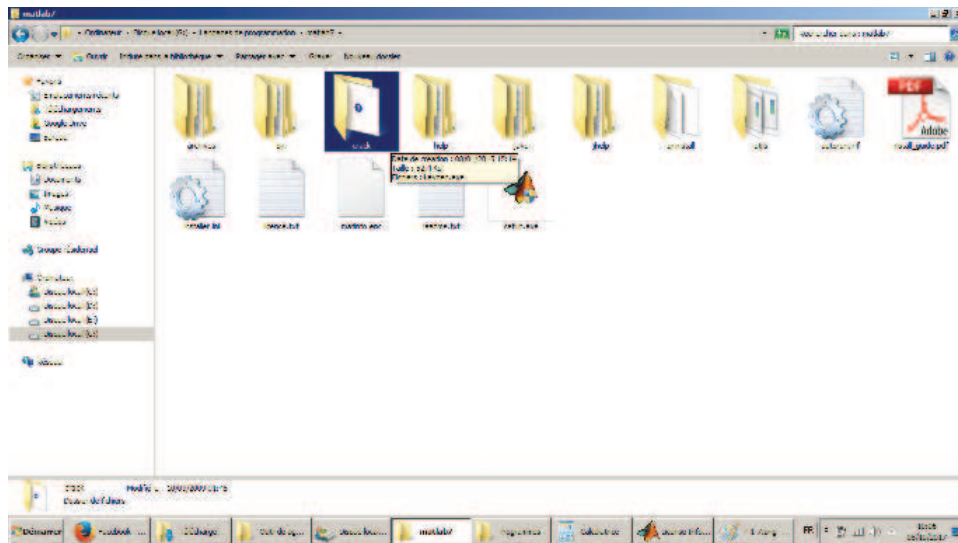


FIGURE 1.6 – Accéder au fichier Crack



FIGURE 1.7 – Exécuter le keygen



FIGURE 1.8 – Copier le numéro de série

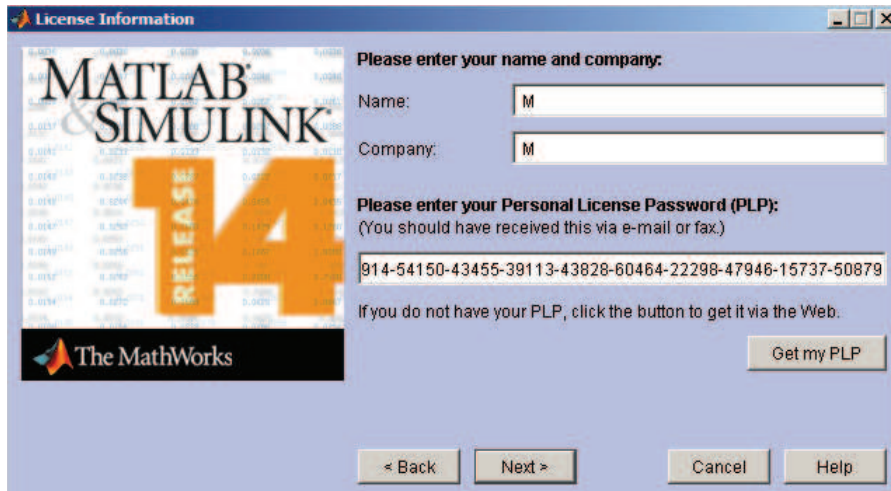


FIGURE 1.9 – Coller le numéro de série

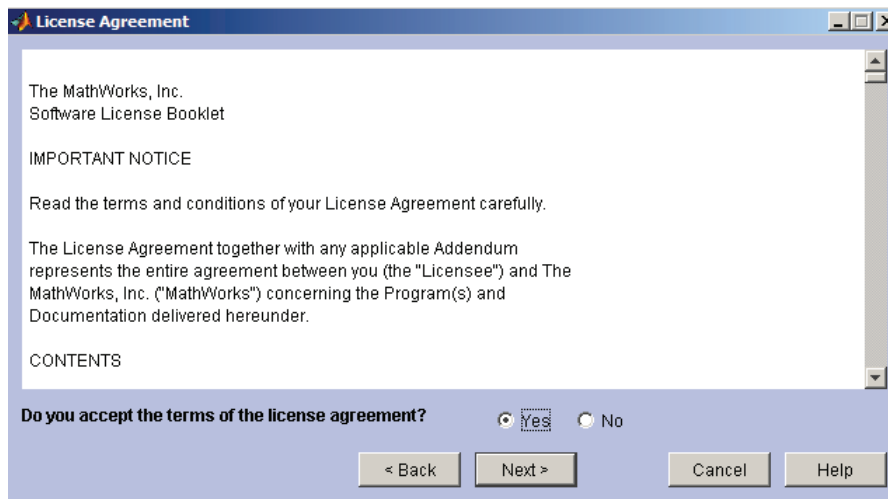


FIGURE 1.10 – Accepter le termes de la licence

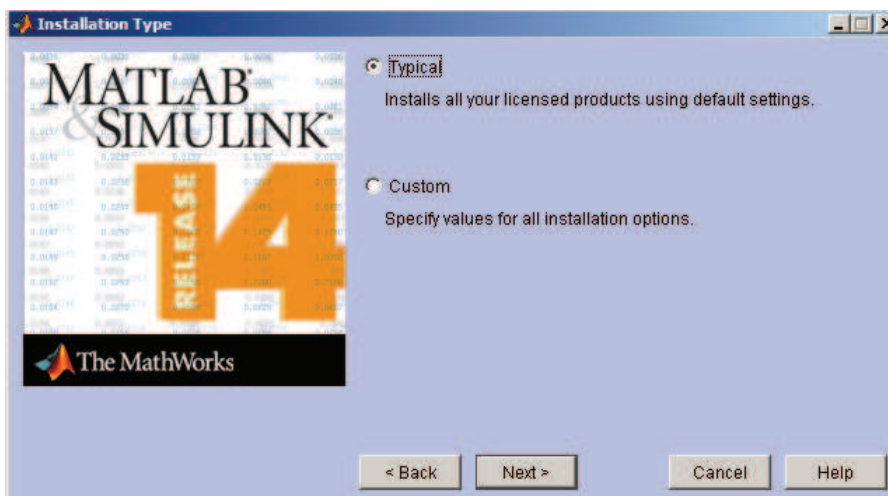


FIGURE 1.11 – Choisir installation typique

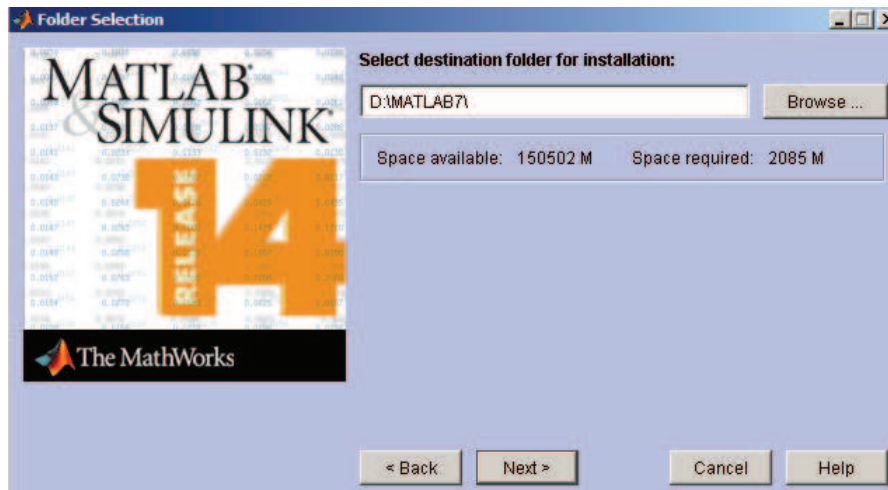


FIGURE 1.12 – Choisir l'emplacement de l'installation

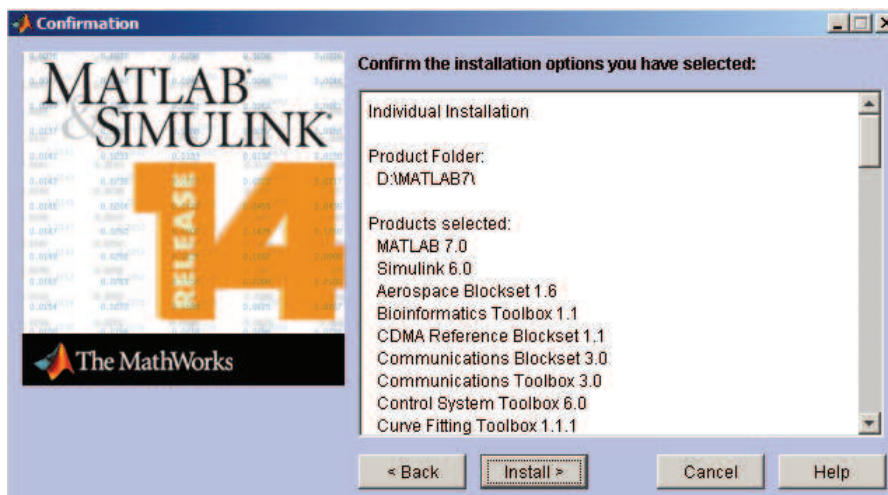


FIGURE 1.13 – Confirmer les options d'installation choisies

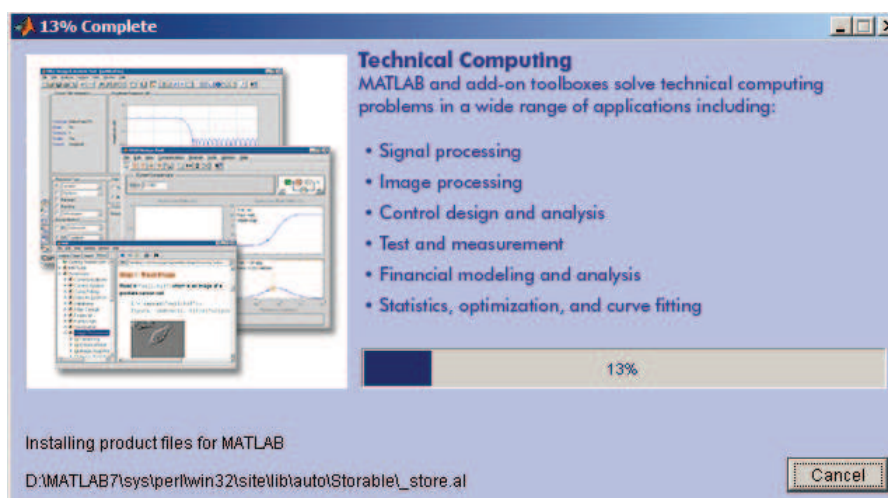


FIGURE 1.14 – Progression de l'installation

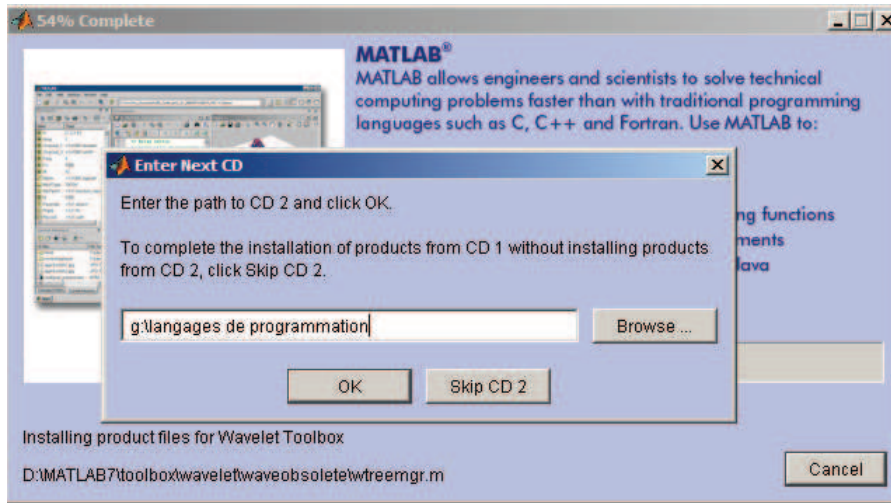


FIGURE 1.15 – Le CD-2 est demandé

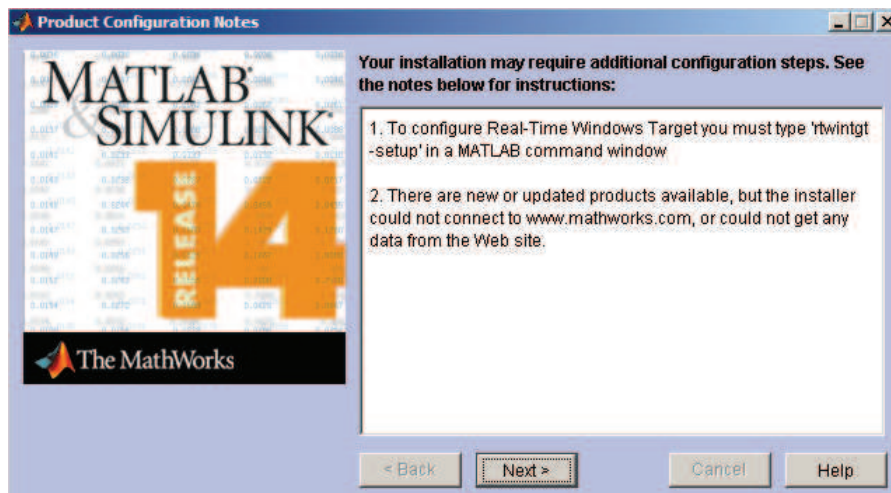


FIGURE 1.16 – Poursuivre l'installation

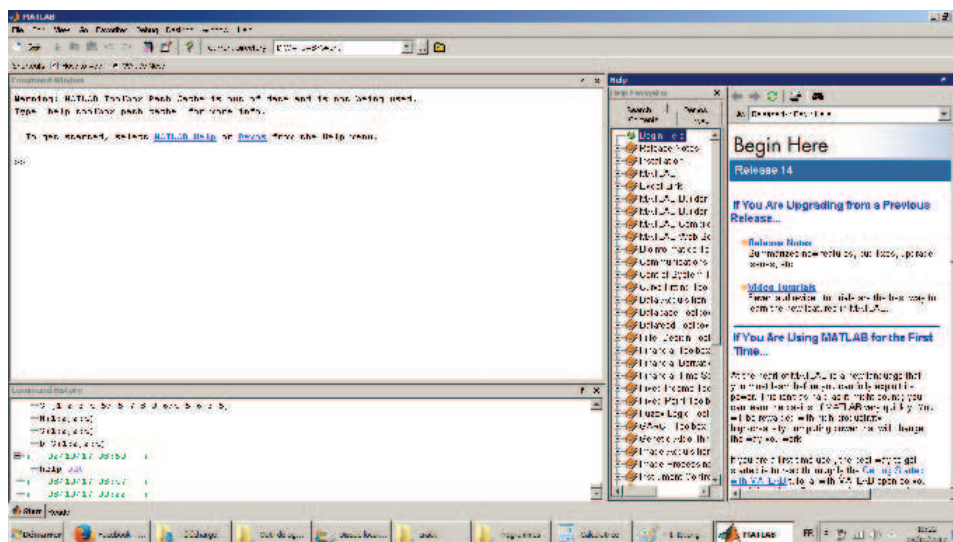


FIGURE 1.17 – Lancement du Matlab 7

- `help` : on utilise cet commande pour obtenir l'aide sur une méthode donnée.
- `clock` : affiche l'année, le mois, le jour, l'heure, les minutes et les secondes.
- `Date` : Affiche la date.
- `input` : permet de lire une valeur à partir du clavier (l'instruction habituelle lire). Exemple :
(`X = input ('taper un nombre :')`).
- `disp` : permet d'afficher un tableau de valeurs numériques ou de caractères. L'autre façon d'afficher un tableau est de taper son nom. La commande `disp` se contente d'afficher le tableau sans écrire le nom de la variable, ce qui peut améliorer certaines présentations. On utilise fréquemment la commande `disp` avec un tableau qui est une chaîne de caractères pour afficher un message. Exemple : `» disp('la valeurs saisie est erronée')`.
- `clear` : permet de détruire une variable de l'espace de travail (si aucune n'est spécifiée, toutes les variables seront effacées).

- `who` : donne la liste des variables définies dans l'espace de travail actuel (essayer `whos`).
- `clc` : effacer le contenu de la fenêtre des commandes et affiche uniquement l'invite « » »

1.5 Quelques fonctions

On plus des commandes, Matlab comme tout langage de calcul scientifique et riche avec son bibliothèques de fonction mathématiques, ces fonctions sont offertes pour implémenté des calculs puissant avec un minimum de code. alors il n'est pass demandé à un tout utilisateur de Matlab de maitraiser l'implémentaion algorithméque de toutes les fonction mathématique (à savoir, sinus, cosinus, expenentielle,).

1. `exp(x)` : exponentielle de x.

Exemple :

```
>> x=5
>>exp(x)
ans =
    148.4132
```

2. `log(x)` : logarithme néperien de x.

Exemple :

```
>> log(1)
ans =
     0
```

3. `log10(x)` : logarithme en base 10 de x. **Exemple :**

```
>> log10(2)
ans =
    0.3010
```

4. `x^n` : x à la puissance n.
5. `sqrt(x)` : racine carrée de x.
6. `abs(x)` : valeur absolue de x.
7. `sign(x)` : 1 si $x > 0$ et -1 si $x < 0$, et 0 si $x=0$.
8. `sin(x)` : sinus de x
9. `cos(x)` : cosinus de x
10. `tan(x)` : tangente de x
11. `round(x)` : Affiche l'arrondi d'un nombre x.

12. `floor(x)`: Arrondissement vers $-\infty$.

Exemple :

```
>> floor(5.2)
ans =
     5
>> floor(5.7)
ans =
     5
>> floor(-5.7)
ans =
    -6
```

13. `ceil(x)` : fait exactement l'inverse de `floor(x)`:

Exemple :

```
>> ceil(-5.7)
ans =
    -5
>> ceil(-5.2)
ans =
    -5
>> ceil(5.2)
ans =
     6
```

14. `rem(m,n)`: reste de la division entière de m par n.
15. `lcm(m,n)`: plus petit commun multiple de m et n.
16. `gcd(m,n)`: plus grand commun diviseur de m et n.
17. `factor(n)`: d'ecomposition en facteurs premiers de n.

1.6 Opérateurs logiques :

<code>~=</code>	L'opérateur 'NON' (différent)
<code>==</code>	L'opérateur 'égal'
<code>&</code>	L'opérateur 'et'
<code> </code>	L'opérateur 'ou'
<code>></code>	supérieur à
<code><</code>	inférieur à
<code>>=</code>	supérieur ou égal
<code><=</code>	inférieur ou égal