

3

L'interface graphique

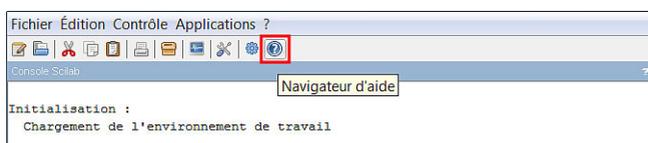
Outre la console, l'interface graphique de Scilab est composée de différentes fenêtres qui peuvent être insérées/séparées de la fenêtre principale (voir la vidéo de la [Figure 2.2](#)). Dans ce chapitre, vous allez découvrir les fonctionnalités de base des principales fenêtres associées à Scilab.

3.1. L'aide en ligne

La première fenêtre à connaître est assurément celle du navigateur d'aide, que vous pouvez appeler de plusieurs manières :

- depuis le la barre de menu en cliquant sur ? puis AIDE DE SCILAB ;
- en pressant la touche F1 ;
- depuis le raccourci dans la barre d'outil (voir [Figure 3.1](#)) ;

Figure 3.1 : Raccourci pour accéder à l'aide en ligne dans la barre d'outils

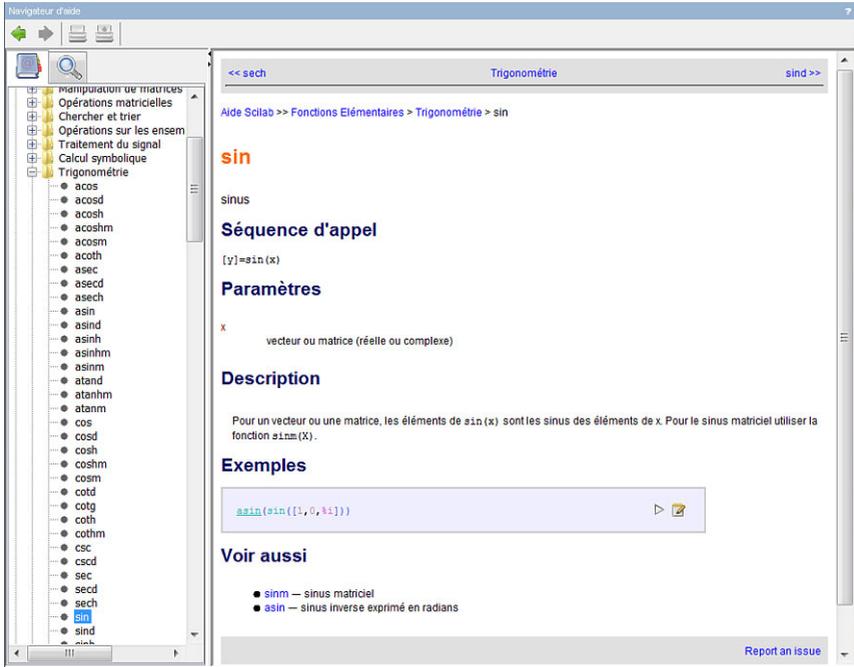


- depuis la console avec l'une des commandes :
 - `help` qui permet de savoir ce que fait une commande et comment l'utiliser ;
 - `apropos` qui permet de connaître les commandes en rapport avec un thème ou un mot clé.

```
-->apropos sinus
-->help sin
```

Dans tous les cas, la fenêtre de la [Figure 3.2](#) s'ouvrira.

Figure 3.2 : Fenêtre d'aide



Astuce > si vous voulez obtenir de l'aide sur la commande `sin`, saisissez `help sin` mais si vous voulez obtenir de l'aide sur la notion mathématique de `sinus`, préférez `apropos sinus`. Dans les deux cas, vous devriez obtenir un lien vers, ou directement, la page d'aide de la fonction `sin`.

Pour chaque commande, l'aide fournie est composée de plusieurs rubriques :

Séquence d'appel

Décrit comment on doit appeler la commande pour effectuer son évaluation dans la console de Scilab. En général la séquence d'appel s'écrit sous la forme :

```
[sortie1, sortie2, ...]=commande(entrée1, entrée2, [option1, [option2]])
```

Cette écriture signifie que `commande` prend en entrée deux valeurs, désignées par `entrée1` et `entrée2`, plus une ou deux valeurs optionnelles, `option1` et `option2`, qui sont encadrées par des crochets `[` et `]`. En sortie, on peut récupérer les valeurs `sortie1` et `sortie2` en utilisant les crochets. Il y a donc plusieurs manières d'appeler la commande, par exemple :

- `sortie1=commande(entrée1,entrée2)`
- `[sortie1,sortie2]=commande(entrée1,entrée2)`
- `[sortie1,sortie2]=commande(entrée1,entrée2,option1)`
- `sortie1=commande(entrée1,entrée2,option1,option2)`
- ...

Attention > Quelle que soit la syntaxe utilisée, on ne peut jamais récupérer la valeur `sortie2` sans avoir récupéré celle de `sortie1`. De même, on ne peut pas spécifier `option2` sans avoir spécifié `option1`.

Paramètres

Décrit le type de paramètres attendu des différentes valeurs (`entrée1,entrée2,option1,... sortie2`) qui apparaissent dans la séquence d'appel (voir Chapitre [Variables, constantes et types](#)).

Description

Donne une explication plus détaillée de ce que fait `commande`.

Exemples

Montre comment utiliser `commande` dans des cas plus ou moins simples. Le code source affiché dans un cadre grisé peut être directement interprété dans la console ou ouvert dans l'éditeur de texte grâce aux deux boutons en haut à droite du cadre (Voir [Figure 3.3](#)).

Figure 3.3 : Code source dans l'aide en ligne



Voir aussi

Renvoie vers d'autres pages de l'aide qui peuvent compléter celle de `commande`.

3.2. L'éditeur de texte

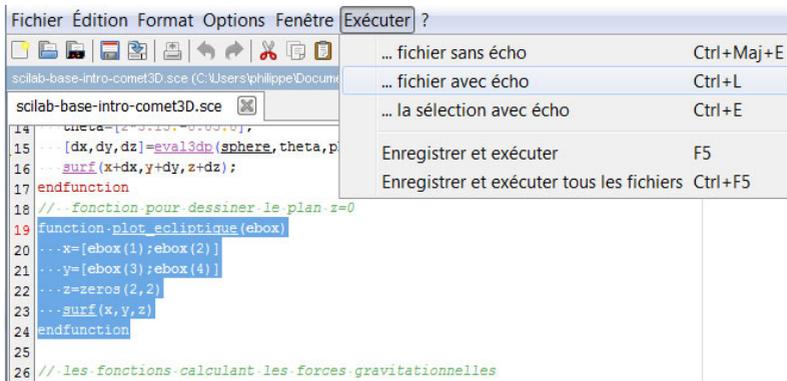
Scilab intègre un éditeur de texte appelé SciNotes, qui peut être appelé de plusieurs manières :

- depuis la barre de menu APPLICATIONS/SCINOTES ;
- depuis le bouton  dans la barre d'outil ;
- depuis la console avec une des commande `editor` ou `scinotes`.

Cet éditeur possède toutes les fonctionnalités d'un éditeur de texte destiné à la programmation :

- formatage semi-automatisé du texte comme commenter/décommenter des parties de code, indentation des commandes imbriquées ;
- coloration syntaxique et complétion des commandes, fermeture des caractères de parenthésage ;
- fonction de recherche/remplacement, avec utilisation d'expressions régulières ;
- plusieurs modes d'exécution du code de l'éditeur dans Scilab (avec/sans écho) (voir Figure 3.4) ;
- gestion des formats de fichiers Windows/Linux (fins de lignes CR et/ou LF) et des encodages en entrée et en sortie (UTF-8, Latin1, ASCII, etc.).

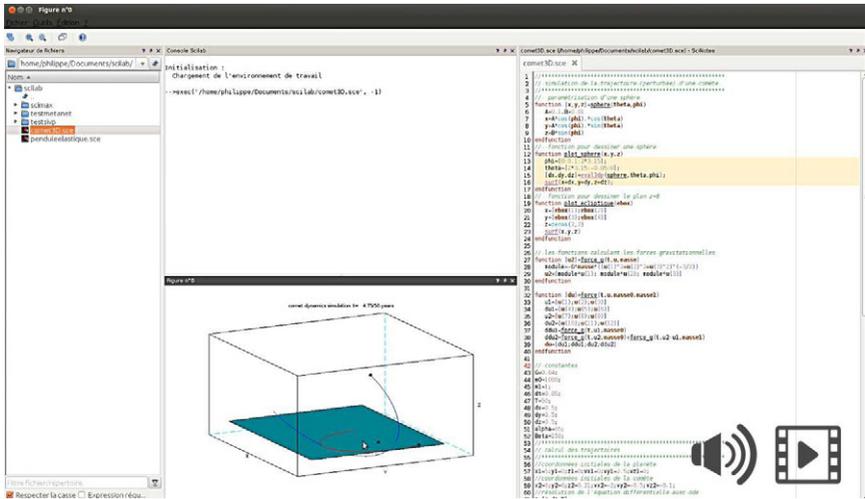
Figure 3.4 : L'éditeur de texte SciNotes



Dans la [Partie Programmer](#), nous utiliserons cet éditeur de texte pour écrire des programmes en langage Scilab. Vous trouverez, dans l'interface graphique de l'éditeur et

de la console, des fonctionnalités spécifiques pour contrôler le lancement et l'exécution des programmes Scilab. Elles sont accessibles depuis le menu CONTRÔLE. Vous pouvez en voir un exemple d'utilisation sur la vidéo [suivante](#).

Figure 3.5 : Contrôle de l'exécution des programmes Scilab (vidéo)



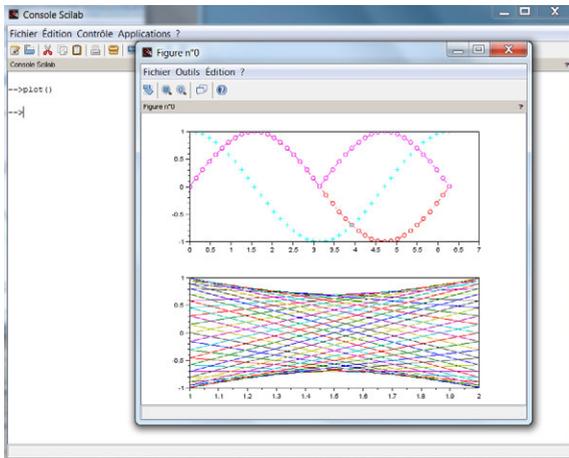
3.3. Les autres fenêtres

Scilab possède d'autres fenêtres dédiées à des fonctionnalités spécifiques, dont voici un petit aperçu.

Fenêtre graphique

Scilab dispose de nombreuses fonctionnalités graphiques, dont l'utilisation déclenche l'ouverture d'une fenêtre graphique. Par exemple, lancer la commande `plot` depuis la console ouvrira la fenêtre graphique de la [Figure 3.6](#).

Figure 3.6 : Fenêtre graphique de Scilab

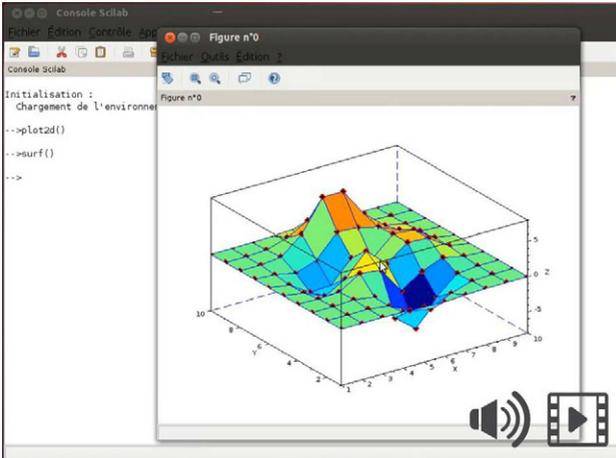


Nous étudierons en détails les propriétés graphiques de Scilab dans la [Partie Créer des graphiques](#) et vous verrez comment réaliser diverses figures à deux mais aussi à trois dimensions ! La fenêtre graphique permet d'interagir facilement avec ces figures, en particulier à l'aide de la souris vous pouvez :

- effectuer des zooms, et revenir à l'échelle de départ ;
- modifier l'angle de vue pour les figures en trois dimensions.

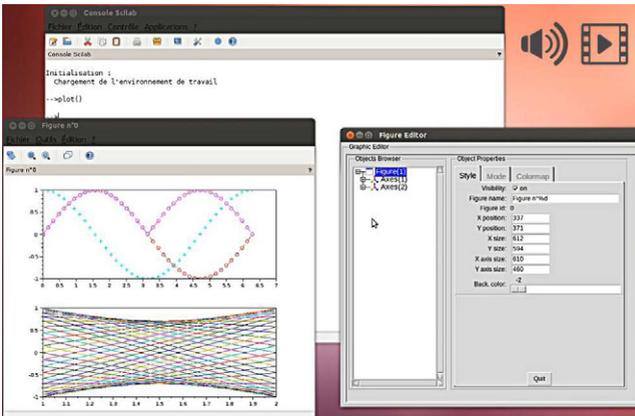
La vidéo [suivante](#) vous montre comment utiliser ces fonctionnalités.

Figure 3.7 : Zooms et modification de l'angle de vue (vidéo)



Les capacités graphiques de Scilab reposent sur une représentation hiérarchique très détaillée des objets graphiques, qui permet non seulement de créer des figures complexes mais aussi de nouvelles interfaces graphiques (voir [Section 23.4, Créer ses propres interfaces graphiques](#)). Un éditeur d'entités graphiques, accessible depuis la barre de menu de la fenêtre graphique (menu ÉDITION puis PROPRIÉTÉS DE LA FIGURE) permet de naviguer parmi les différents paramètres qui composent un graphique et aussi de les modifier. La fenêtre graphique prend en compte ces modifications en temps réel (voir la vidéo [ci-après](#)).

Figure 3.8 : Utilisation de l'éditeur d'entités graphiques (vidéo)

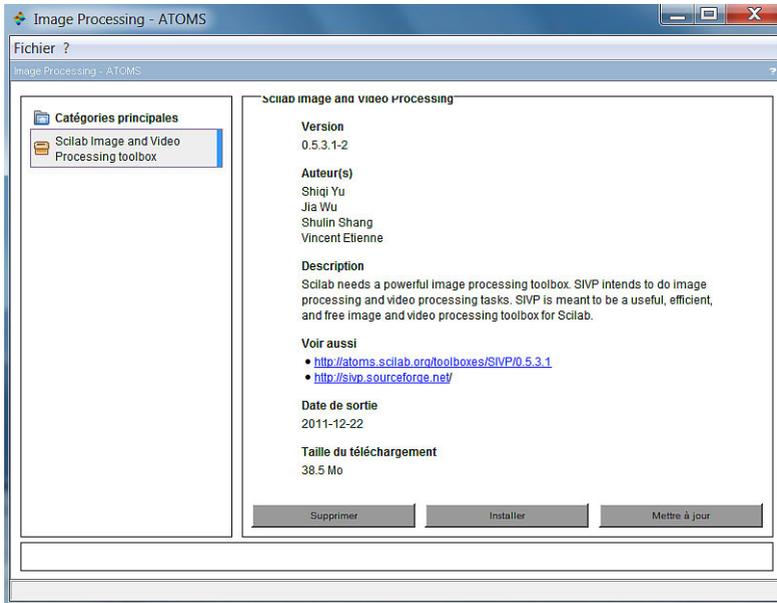


Attention > Actuellement, l'éditeur d'entités graphiques ne fonctionne pas sur les versions de Scilab pour Mac OS.

ATOMS

Le gestionnaire de modules externes ATOMS permet d'installer ou de désinstaller des modules complémentaires associés à Scilab. Ces modules ajoutent au logiciel des fonctionnalités de haut niveau, tels que SIVP pour le traitement d'images ou Metanet pour la gestion des graphes (voir aussi [Section 5.4, Les modules complémentaires sur Forge](#)).

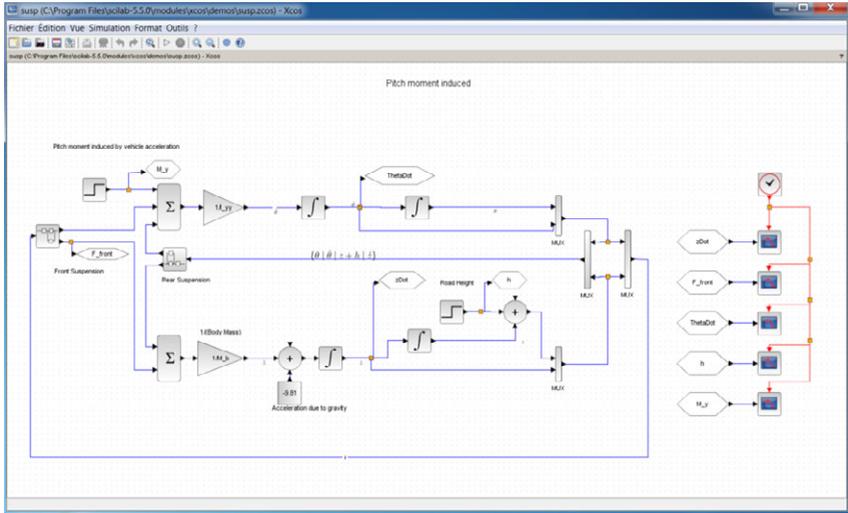
Figure 3.9 : Choix du module complémentaire SIVP dans la fenêtre ATOMS



Xcos

L'éditeur de bloc-diagrammes Xcos, équivalent Scilab de la boîte à outils Simulink de Matlab, permet de simuler des systèmes dynamiques. Pour plus d'informations sur Xcos, se reporter au [manuel](#) qui lui est consacré.

Figure 3.10 : Schéma Xcos d'une suspension d'automobile



Gestionnaires divers

Plusieurs gestionnaires ont été ajoutés à Scilab depuis la version 5 : un [gestionnaire d'historique](#), un [éditeur de variables](#) et un [navigateur de fichiers](#). Ils sont lancés par défaut lors du premier démarrage de Scilab (voir [Figure 2.1](#)). Vous pouvez aussi les lancer depuis la console avec les commandes : `filebrowser`, `browsevar`, `editvar` et `browsehistory`.