

Recommandations générales

Édition : Utiliser l'éditeur intégré à scilab. Placer les fonctions dans des fichiers suffixés `.sci` et les scripts d'exécution dans des fichiers suffixés `.sce`. Ne pas utiliser de caractères spéciaux : accents, espaces, etc. pour nommer les fichiers, variables et fonctions. Utiliser des noms de variables et de fonctions qui facilitent la programmation (sans trop l'alourdir).

Commenter efficacement ses programmes à l'aide du symbole `//` (pour décrire l'utilité d'une fonction ou les hypothèses non-évidentes auxquelles sont soumis ses arguments d'entrée).

Exécution : Ne pas oublier de se placer préalablement dans le répertoire de travail qui contient les fichiers à utiliser, au niveau de la ligne de commande de Scilab.

Les fonctions utilisables par le logiciel sont celles ayant été mises préalablement en mémoire (`exec fichier.sci`). Toute modification d'une fonction ne sera effective qu'après nouvelle exécution du fichier la contenant (on peut aussi utiliser la touche `play` de l'éditeur).

Debugage : Pour la recherche d'une **erreur de programmation ou de code**, on veillera à réfléchir plutôt qu'à modifier le programme au hasard...

La commande `pause` glissée dans une fonction ou un programme permet d'interrompre l'exécution et de récupérer la main dans la ligne de commande afin d'examiner les variables ; l'exécution est reprise par `resume` ou interrompue définitivement par `abort`.

Un message d'erreur de scilab indique la ligne à partir de laquelle une erreur se manifeste mais il faut souvent examiner les lignes précédentes qui peuvent en être la source réelle.

Dans la recherche d'une erreur affectant le résultat d'un algorithme, une **stratégie de debugage** efficace consiste à décliner une série de tests élémentaires et progressifs qui permettent de valider les différentes étapes de l'algorithme indépendamment les unes des autres. Cette validation peut/doit être effectuée au fil de la programmation qui est alors elle-même progressive.

Définition de vecteurs et matrices

| | |
|------------------------------|--|
| <code>[]</code> | Définition d'un vecteur ou d'une matrice (séparateurs <code>,</code> et <code>;</code>) |
| <code>a:h:b</code> | Vecteur incrémental de pas <code>h</code> entre <code>a</code> et <code>b</code> |
| <code>linspace(a,b,N)</code> | Vecteur de <code>N</code> valeurs équiréparties entre <code>a</code> et <code>b</code> |
| <code>zeros(m,n)</code> | Matrice nulle de taille <code>m×n</code> |
| <code>ones(m,n)</code> | Matrice de 1 de taille <code>m×n</code> |
| <code>rand(m,n)</code> | Matrice aléatoire (coeff. unif. distribués dans <code>[0, 1]</code>) de taille <code>m×n</code> |
| <code>eye(m,n)</code> | Matrice identité de taille <code>m×n</code> |
| <code>diag(X)</code> | Construction de la matrice carrée dont la diagonale est le vecteur <code>X</code> |
| <code>diag(X,i)</code> | Construction de la matrice carrée dont la <code>i</code> -ième diagonale est le vecteur <code>X</code> |

Manipulation de vecteurs et matrices

| | |
|--------------------------|---|
| <code>length(X)</code> | Longueur d'un vecteur (nombre de composantes d'une matrice <code>X</code>) |
| <code>size(A)</code> | Taille <code>m×n</code> de la matrice <code>A</code> |
| <code>size(A,1)</code> | Nombre de lignes de la matrice <code>A</code> |
| <code>size(A,2)</code> | Nombre de colonnes de la matrice <code>A</code> |
| <code>A(I,J)</code> | Extraction de la sous-matrice des coeff. de <code>A</code> situés aux lignes <code>I</code> et colonnes <code>J</code> |
| <code>A(:,1)</code> | Extraction de la première colonne de <code>A</code> |
| <code>A(2:\$-1,1)</code> | Extraction de la première colonne de <code>A</code> excepté les premier et dernier coefficients |
| <code>diag(A)</code> | Extraction du vecteur formant la diagonale de <code>A</code> |
| <code>diag(A,i)</code> | Extraction du vecteur formant la <code>i</code> -ième diagonale de <code>A</code> |
| <code>A.#B</code> | où <code>#</code> ∈ <code>{*,./,^}</code> , Opération composante par composante entre les matrices <code>A</code> et <code>B</code> |
| <code>A'</code> | Adjoint de la matrice <code>A</code> |
| <code>A.'</code> | Transposée de la matrice <code>A</code> |

Opérations d'algèbre linéaire

| | |
|----------------------------|--|
| <code>norm(X,p)</code> | $\ X\ _p$ ($p \in [1, +\infty[$ ou $p = 'inf'$) |
| <code>norm(A,p)</code> | $\ A\ _p$ (norme subordonnée lorsque $p \in \{1,2,'inf'\}$, de Frobenius si $p = 'fro'$) |
| <code>det(A)</code> | Déterminant de la matrice <code>A</code> |
| <code>rank(A)</code> | Rang de la matrice <code>A</code> |
| <code>trace(A)</code> | Trace de la matrice <code>A</code> |
| <code>inv(A)</code> | Inverse de la matrice <code>A</code> |
| <code>\</code> | Résolution d'un système linéaire $Ax = b$ par $x=A\b b$ |
| <code>kernel(A)</code> | Base orthonormée du noyau de <code>A</code> |
| <code>spec(A)</code> | Vecteur des valeurs propres de <code>A</code> avec multiplicité |
| <code>[P,D]=spec(A)</code> | $A = PDP^{-1}$, <code>P</code> contient les vecteurs propres lorsqu'ils existent |

Booléens et opérateurs logiques

| | |
|---------------------------------------|---|
| <code>==,>,<,>=,<=</code> | Tests <code>=, >, <, ≥, ≤</code> , respectivement |
| <code>&</code> | Opérateur logique ET |
| <code> </code> | Opérateur logique OU |

Structures de contrôle

| | |
|--|--|
| <code>function [y1,y2]=f(x1,x2,x3)</code> ... <code>endfunction</code> | Le programme de la fonction <code>f</code> doit impérativement définir les arguments de sortie <code>y1</code> et <code>y2</code> (elle n'est pas destinée à faire autre chose, comme du tracé graphique). |
| <code>for i=LISTE</code> ... <code>end</code> | <code>LISTE</code> est un vecteur de valeurs parcourues successivement par la variable <code>i</code> , le plus souvent un vecteur incrémental <code>1:n</code> ou décrémental <code>n:-1:1</code> |
| <code>while condition</code> ... <code>end</code> | <code>condition</code> est un booléen |
| <code>if cond_1 then</code> ... <code>elseif cond_2 then</code> ... <code>else</code> ... <code>end</code> | On peut utiliser plusieurs <code>elseif</code> successifs. Attention à la position de <code>then</code> : une erreur courante et difficile à repérer est de placer <code>then</code> sur la ligne suivant le booléen <code>cond_1</code> . |

Messages d'erreurs fréquents

| | |
|---|---|
| Index invalide. | Tentative d'accès à une composante inexistante d'une matrice : vérifier les «effets de bord» dans les boucles, la nature des indices (réels, entiers, positifs, non-nuls et inférieurs aux tailles des vecteurs). |
| Multiplication incohérente. | Produit incompatible de matrices. |
| La sous-matrice n'est pas correctement définie. | Affectation d'une sous-matrice avec dimensions incompatibles. |
| Division par zero... | Le message est limpide. |
| Attention : La matrice est presque singulière ou mal conditionnée. <code>rcond = ...</code> | Ce n'est pas une erreur mais un «warning». La résolution d'un système linéaire faisant intervenir une matrice «presque non-inversible». |