

**NOTE D'INFORMATION**  
**SUR L'ÉPREUVE FACULTATIVE D'INFORMATIQUE**  
**DES CONCOURS AGRO-VÉTO**  
**(Concours A des filières BCPST et TB)**  
(Cette note annule et remplace la note d'application du 28/08/01)

**1 - Présentation**

Le programme d'informatique commun aux classes préparatoires BCPST et TB est détaillé dans le BOEN hors série N° 3 du 26 juin 2003.

Pour les classes TB, son enseignement a débuté en septembre 2004 et la première épreuve facultative est proposée aux candidats de cette filière au concours 2006.

La présente note d'information a pour objet de préciser dans quelles conditions les candidats aux concours de ces deux filières sont évalués tant dans les épreuves écrites de mathématiques susceptibles de contenir des parties algorithmiques (pour les candidats BCPST) que dans l'épreuve orale facultative d'informatique.

**2 - Modalités**

Les langages servant de support à l'évaluation des candidats sont exclusivement :

- le langage du logiciel MATLAB
- le langage du logiciel SCILAB.

**Epreuves écrites** : Les questions présentant un aspect algorithmique susceptibles d'être posées pourront être traitées par les candidats quel que soit le langage enseigné, parmi les deux autorisés.

**Epreuve orale facultative** : Le programme présenté par le candidat pourra être écrit :

- soit en langage propre au logiciel MATLAB
- soit au langage propre au logiciel SCILAB.

**3 - Caractéristiques et fonctionnalités à connaître par les candidats.**

**Avant propos**

Il est difficile de fixer des limites rigides à l'utilisation de type Matlab ( du nom du logiciel le plus répandu, mais ne se limitant pas à Matlab mais incluant notamment Scilab) et d'obliger les étudiants à n'utiliser qu'un sous-ensemble figé des fonctionnalités offertes par ces logiciels Matlab ou Scilab. Une telle rigidité est contraire à l'esprit même de l'informatique, qui a été de développer toujours de nouveaux outils plus souples et plus abstraits afin de pouvoir résoudre plus rapidement une certaine catégorie de problèmes. (pour ne citer que quelques exemples : Perl pour la manipulation et transformation de textes, Python ou CAML pour des outils de prototypage impliquant la manipulation de structures de données

plus complexes, Java pour la programmation des machines à café, même si ce dernier a connu une expansion en dehors de son domaine initial).

A l'opposé de Pascal, qui a été conçu dès l'origine par Niklaus Wirth comme un langage d'enseignement, les logiciels de la famille Matlab ont été introduits afin de simplifier les problèmes de calcul numérique et à terme de se substituer, au moins pour les applications de prototypage ou de calcul ne nécessitant pas une puissance de calcul énorme, à Fortran ou au C.

Si les logiciels du type de Matlab ont été retenus, ce n'est pas pour « émuler un sous-ensemble d'un langage héritant d'Algol (comme Pascal, C ou Java) », mais comme un environnement moderne et performant permettant de résoudre une grande classe de problèmes relevant de domaines aussi divers que le calcul numérique pur (Monte-Carlo), l'automatique, l'intelligence artificielle (réseaux de neurones), l'algorithmique graphique (filtrage, détection de contours), etc. Réduire l'utilisation de la famille de Matlab à un sous-ensemble bien précis signifierait restreindre la classe de problèmes que l'étudiant pourra traiter avec Matlab.

En conséquence, il nous a paru plus judicieux de définir un niveau plancher – ce que l'étudiant doit absolument savoir, et un niveau plafond—défini par ce que l'étudiant ne doit en aucun cas utiliser. Pour le reste, l'étudiant ou son préparateur devra savoir pourquoi il fait appel à telle ou telle fonction ou fonctionnalité de Matlab et être en mesure d'expliquer comment implémenter cette fonctionnalité en utilisant un algorithme naïf en Matlab ou en utilisant le langage d'un logiciel équivalent.

Le programme et les fonctions prédéfinies ci-dessous qui ont été détaillés pour **Matlab** peuvent être aisément transposables pour les langages de la même famille, notamment **Scilab**.

## Programme

Soit un problème donné  $Y$ , nous appelons algorithme une suite d'instructions  $P$  travaillant sur un ensemble d'objets  $D$ . Outre l'usage de fonctions définies par l'algorithme, celui-ci peut faire appel à un ensemble de fonctions prédéfinies que nous dénommons  $PF$ . La première partie décrit l'ensemble des objets  $D$  que le processus  $P$  peut manipuler, la seconde partie décrit l'ensemble des inscriptions composant un processus  $P$ , et enfin la dernière partie décrit l'ensemble des fonctions prédéfinies  $PF$  que le candidat peut utiliser dans son projet.

### Les objets manipulés

Nous considérons les objets pouvant être utilisés dans les algorithmes. Ces objets se scindent en deux catégories, d'une part les objets élémentaires ou qui seront considérés comme élémentaires, d'autre part les objets structurés

### Les objets élémentaires

Les objets élémentaires correspondent aux types de valeur simple : les valeurs numériques, les nombres complexes ainsi que les chaînes de caractères.

Il est toutefois nécessaire d'attirer l'attention des étudiants sur la notion de valeurs logiques, même si, à proprement parler, il n'est pas défini de valeurs booléennes.

### Les objets structurés

Il est demandé de connaître d'une part les objets correspondant à une notion mathématique, c'est à dire les vecteurs et les matrices. D'autre part, les étudiants devront être à même d'utiliser la notion de tableaux au sens représentation de données structurées. Ainsi les tableaux à plus de deux dimensions sont autorisés. Cependant, pour les tableaux à une ou deux dimensions, les étudiants devront être à même d'expliquer s'ils utilisent ce « tableau » en tant qu'objet de l'algèbre linéaire ( i.e matrice ou vecteur) ou en tant que structure de données au sens informatique du terme (i.e tableaux de plusieurs éléments).

Même s'il n'est pas possible de déclarer le type d'un objet, l'étudiant doit impérativement :

- être en mesure de définir le type de chaque objet utilisé dans l'algorithme,
- être en mesure d'expliquer pourquoi telle structure a été employée et non telle autre (eg. pourquoi un tableau de vecteur et non un vecteur de matrices),
- être capable de construire un élément d'un type donné.

### Les instructions composant l'algorithme

1. Construction et initialisation des objets.
  - Affecter à une variable une valeur numérique, un nombre complexe ou une chaîne de caractères
  - Créer un vecteur ou une matrice et l'initialiser.
2. Accéder et manipuler les objets
  - Accéder à un élément d'un tableau, d'une matrice ou d'un vecteur,
  - Accéder à une sous-matrice ou à un sous-vecteur.
  - Affectation d'un tableau, d'une matrice ou d'un vecteur, (le candidat devra pouvoir écrire l'algorithme d'affectation à l'aide de boucles *for* ou *while*)
3. Les opérateurs traditionnels sur les objets simples
  - les quatre opérations,
  - les opérateurs relationnels (comparaison),
  - les opérateurs logiques.
4. Les opérateurs sur les objets structurés
  - l'addition et la multiplication de matrices,
  - les opérateurs relationnels
  - Pour ces deux classes d'opérateurs, les candidats doivent être en mesure d'écrire les actions effectuées à l'aide de boucles *for* ou *while*.
  - les quatre opérations sur les tableaux, c'est à dire les opérations fonctionnant sur des tableaux à  $n$  entrées et non limités aux vecteurs et matrices. Il s'agit de +, -, .\* et ./.
  - les autres opérateurs comme la transposition, la division à gauche ou à droite ne sont pas sensés être connus. Ils peuvent toutefois être utilisés mais le candidat doit être en mesure d'expliquer pourquoi il les a employés et ce qu'ils sont censés faire et ce de manière précise.
5. Les instructions de contrôle
  - la boucle *for*,
  - la boucle *while*
  - les branchements *if*
6. L'appel de fonctions
- Toutefois la récursivité est hors programme
7. La définition de fonctions
  - la définition d'une fonction,
  - les règles de portée quant à la définition d'une fonction.

### Les fonctions prédéfinies de Matlab

Avertissement : les fonctions de Matlab étant nombreuses et évoluant, on ne peut pas en faire une liste exhaustive. On ne peut que se contenter de conseiller aux préparateurs de ne faire utiliser aux candidats que des fonctions faisant appel à des algorithmes élémentaires et quand elles répondent à un problème tout à fait annexe à celui traité. Un programme présenté à l'épreuve orale d'informatique ne saurait être l'appel successif de quatre fonctions Matlab !

On divise les instructions et fonctions de Matlab en cinq catégories :

**1. Instructions et fonctions faisant partie du programme :**

- la syntaxe de Matlab
- fonctions de dimensionnement de matrices : ones, zeros, ...
- fonctions donnant la taille des objets manipulés : length, size, ndims,...
- fonctions d'entrée input et de sortie disp.
- Fonctions mathématiques de base : sin, acos, exp, abs, floor, ceil

**2. Instructions et fonctions pouvant être utilisées par le candidat sans qu'elles donnent lieu à des questions :**

- fonctions graphiques.

**3. Instructions et fonctions hors programme, susceptibles de donner lieu à des questions si le candidat les utilise. (Questions portant notamment sur l'algorithme correspondant )**

- fonctions élémentaires de définition de vecteurs ou de matrices : linspace, meshgrid, eye, diag, ...
- fonctions élémentaires de manipulation de matrices :
- *max, min, sum, ...*
- transposée, *fliplr, flipud, ...*
- *sort* (l'élève devra alors connaître un algorithme de tri).
- calcul matriciel *.\* ./ et .^*
- résolution de systèmes linéaires  $\backslash, /$ , (ou *inv*) sous condition que le candidat sache programmer un pivot de Gauss.
- fonctions de manipulation de chaînes de caractères.

**4. Instructions et fonctions dont l'utilisation est prohibée.**

Toutes les fonctions Matlab concernant :

- l'optimisation et la recherche de zéros (*fminbnd, fminsearch, fzero...*)
- le filtrage et les produits de convolution (*filter, conv, fft, ifft, ...*)
- l'intégration numérique (*quad, ode23, ode45, ...*)
- l'algèbre linéaire (*det, cond, eig, chol, ...*)
- les polynômes (*roots, poly, ...*)

**5. Les instructions et fonctions non citées ci-dessus relèvent automatiquement de la troisième catégorie : le candidat doit être en mesure de :**

- expliquer et réécrire ou donner un algorithme utilisé par la fonction.
- justifier de l'obligation d'utiliser cette fonction.

Le chef du service des concours  
Philippe de La Mettrie