

LM206 INITIATION À SCILAB - PROJETS

Remarques :

- chaque projet donnera lieu à une courte présentation (5-10 minutes) en dernière séance, et un court rapport (2 pages ou plus) à me transmettre par courriel le 15 décembre au plus tard ;
- le rapport devra inclure la description du projet, les outils et techniques utilisés, les difficultés rencontrées, et les parties intéressantes du code SCILAB¹ ; n'importe quel format est acceptable, avec une préférence pour les fichiers PDF rédigés avec L^AT_EX, mais j'accepte aussi, par exemple, les documents Word ;
- la documentation, l'initiative personnelles, la qualité du résultat sont plus pertinents (ici) que l'exactitude scientifique ;
- en ce qui concerne le code SCILAB, il est particulièrement important d'utiliser au maximum les fonctions pré-définies ; les boucles dont à éviter au maximum ;
- avant de commencer la partie pratique du travail, consultez les références proposées, et définissez clairement ce que vous voulez réaliser avec SCILAB ;
- pour des raisons pratiques (fiabilité, disponibilité de nombreux outils), il est conseillé de travailler dans l'environnement Linux ;
- références générales mathématiques : <http://mathworld.wolfram.com/> et <http://fr.wikipedia.org/wiki/Portail:Mathematiques>

1 Projets

1.1 Finance statistique

- Source possible de données (format texte) : <http://www.oanda.com/convert/fxhistory> ; des exemples de fichier de données (change dollar-euro en 2004 et 2005) sont disponibles sur <http://nicolas.limare.net/tmp/>
- Une référence possible : <http://www.lapasserelle.com/econometrie/index.html>
- Lire dans SCILAB l'historique du cours d'un actif financier (change, cours d'une action ou d'un indice boursier, ..), le stocker dans un tableau ;
- En extraire des informations statistiques simples (moyenne, écart-type, tendances, périodicité) ;
- Essayer d'y ajuster les paramètres d'une fonction (linéaire, polynomiale) ;
- Calculer la corrélation entre plusieurs actifs financiers ;
- ...

¹Le code intégral sera transmis en fichier joint

1.2 Marche aléatoire et mouvement brownien

- Références : http://fr.wikipedia.org/wiki/Mouvement_brownien, http://fr.wikipedia.org/wiki/Marche_au_hasard et http://www.physique2005mip.asso.fr/marche_au_hasard/
- Construire une fonction de marche aléatoire en une dimension (approximation du mouvement brownien), puis en deux ou trois dimensions ;
- Observer les propriétés géométriques de cette marche aléatoire ;
- Calculer les propriétés statistiques de cette marche aléatoire ;
- Comparer l'intégrale du mouvement brownien, et l'intégrale de la modélisation effectuée ;
- ...

1.3 Générateurs aléatoires

- Référence générale : <http://tinyurl.com/ykdkh9>
- Vérifier que le générateur aléatoire de SCILAB produit effectivement des données pseudo-aléatoires ;
- Construire plusieurs fonctions générant des nombres aléatoires, selon plusieurs techniques existantes ;
- Tester les caractéristiques de ces générateurs ;
- Comparer ces générateurs, leur qualité et leur rapidité ;
- ...

1.4 Cryptographie

- Refs. : <http://fr.wikipedia.org/wiki/Cryptographie> et www.irem.univ-mrs.fr/recentes/concret1.pdf
- Explorer les diverses fonctions SCILAB d'arithmétique (pgcd, ppcm, restes, nombres premiers)
- Implémenter le chiffre de Vigenère
- Implémenter le principe d'échange de clés de Diffie-Hellman
- Implémenter le principe de signature chiffrée RSA
- Implémenter le principe de hachage MD5
- ...

1.5 Linguistique

- Sources possibles : les discours présidentiels, disponibles sur <http://elysee.fr> ; des exemples de fichier de données (voeux présidentiels 2001-2004, "nettoyés") sont disponibles sur <http://nicolas.limare.net/tmp/>

- On ne travaillera que sur des fichiers texte brut (pas de fichiers mis en page .doc), sans accents ni ponctuation, modifiés pour avoir un mot par ligne.
- Lire un texte dans SCILAB, et en calculer la fréquence d'utilisation des mots ;
- Raffiner la méthode pour conserver uniquement les mots suffisamment significatifs ;
- Comparer avec cela deux textes ; deux ensembles de textes (discours politiques par exemple) ;
- Evaluer l'évolution des textes d'une même personne dans le temps ;
- Représenter graphiquement la position de plusieurs textes en fonction de cette analyse ;
- Effectuer le même genre de calculs sur en tenant compte de la proximité des mots ;
- ...

1.6 Imagerie

- Références : <http://www.iro.umontreal.ca/~mignotte/ift6150>, <http://siptoolbox.sourceforge.net/> ; l'image standard de test en imagerie (lena_std.png) est disponible sur <http://nicolas.limare.net/tmp/>
- Installer SIPToolbox si nécessaire et afficher une image avec SCILAB ; on ne travaillera qu'avec des images en niveaux de gris ;
- Écrire des fonctions de mesure des caractéristiques de l'image, dont l'histogramme de luminosité, et tracer cet histogramme ;
- Écrire des fonctions de contraste, de flou, de régularisation, ... et les appliquer à une image ;
- Implémenter l'algorithme de Sobel pour détecter les contours ;
- Ecrire une fonction calculant le décalage (mouvement) entre deux images ; l'utiliser pour stabiliser un film ;
- ...

1.7 Algorithmes de réseau

- Référence : http://en.wikipedia.org/wiki/Dijkstra's_algorithm
- Décider d'un mode de représentation d'un graphe (un réseau, avec des nœuds et des arêtes ou connexions), sous forme de matrice (par exemple, $A = (a_{ij})$, avec $a_{ij} = 1$ si les nœuds i et j sont connectés, 0 sinon) ; expérimenter cette notation en l'utilisant pour construire, modifier, assembler des réseaux ;
- À partir d'un exemple fictif (ou non) de réseau, représenté de cette manière, écrire des fonctions pour tester si deux points du réseau peuvent être connectés ;
- Étudier l'algorithme de Dijkstra de calcul du plus court chemin dans un réseau ;
- L'implémenter en SCILAB (en adaptant la notation précédente des graphes, pour y ajouter la notion de distance entre deux sommets), puis l'utiliser pour calculer le plus court chemin entre deux machines, deux stations de métro, deux angles de rues ;

- Créer la carte d'un réseau de bornes wifi aléatoirement distribués, chacune ne se connectant qu'à ses voisins ; et appliquer l'algorithme de plus court chemin à ce réseau ;
- ...

1.8 Mathématiques et calcul numériques

- Références : http://www-anp.lip6.fr/cadna/Exemples_Dir/Accueil.php, <http://www.netlib.org/>
- Construire une fonction de produit matriciel sans utiliser celle pré-existante dans SCILAB (d'abord par des boucles, puis en utilisant au mieux les fonctions de SCILAB, dont `sum`, `.*`, ...);
- Construire une fonction de résolution de système linéaire sans utiliser celle pré-existante dans SCILAB ;
- Se documenter sur les bibliothèques numériques BLAS et LAPACK, et leur intégration dans Scilab ;
- Effectuer des mesures de performance des calculs élémentaires d'algèbre linéaire (produit matriciel, résolution de système, factorisation LU) ;
- Observer les cas problématiques d'erreur d'arrondi des calculs par le CPU ;
- ...

2 Annexes

2.1 Se déplacer dans les dossiers

Scilab permet de “se déplacer” dans les dossiers de l'ordinateur, afin d'accéder aisément aux données sans se soucier des différences de syntaxe pour les chemins d'accès entre divers postes (Windows perso, Windows réseau, Linux perso, Linux réseau). On dispose pour cela de plusieurs commandes, dont celles qui suivent.

- `pwd` affiche le dossier courant, celui dans lequel SCILAB va chercher les fichiers, par défaut.
- avec `cd`, SCILAB se “déplace” vers votre dossier personnel, dans lequel vous avez la possibilité d'enregistrer des fichiers.
- `ls` liste les dossiers et fichiers du répertoire courant.
- avec `cd undossier`, SCILAB se “déplace” dans le dossier `undossier`, qui doit se trouver dans le répertoire courant.

Ainsi, pour accéder au fichier `foo.txt`, qui se trouve dans le dossier `truc` de votre répertoire personnel (`/home/12345678/truc/foo.txt` ou `C:/Documents and Settings/12345678/truc/foo.txt`, selon le système), vous pouvez utiliser les commandes `cd ; cd truc ; read('foo.txt', ...)`

2.2 Lire et traiter les données d'un fichier

La commande principale est `read(...)`, avec des options dépendant du type de données lues. Voici deux exemples :

données numériques Supposons que vous disposez d'un fichier `foo.txt` de lignes de la forme `1.234567`.

– `data = read('foo.txt', -1, 1)` lit tout ce fichier et transfère ce qu'il contient dans le tableau `data`.

texte Supposons que vous disposez d'un fichier de texte `foo.txt`.

– `txt = read('foo.txt', -1, 1, '(a)')` transfère ce fichier dans le tableau `txt`, dont chaque élément contient une chaîne de caractères issue d'une ligne du fichier d'origine.

– On peut ensuite assembler toutes ces lignes, par `txt = strcat(txt)`.

– Généralement, il est souhaitable de séparer les mots. `txt = tokens(txt)` fait cela, en coupant aux espaces.

2.3 Installer SIPTtoolbox

SIPTtoolbox est une “boîte à outils” complémentaire pour SCILAB, dédiée à la manipulation d'images. Si elle n'est pas déjà installée sur votre poste, voici comment la mettre en place puis l'utiliser² :

1. sur <http://sipttoolbox.sourceforge.net/>, section “Download”, choisir le lien “Linux” puis télécharger `sip-0.4.0-bin-linux-sci3.0-i686.tar.gz` ;
2. décompresser ce fichier, par exemple par la commande `tar xvzf sip-0.4.0-bin-linux-sci3.0-i686.tar.gz` effectuée dans la console, à partir du dossier dans lequel le fichier a été enregistré, un dossier `sip-0.4.0-bin` est créé, contenant SIPTtoolbox ;
3. démarrer SCILAB et se déplacer dans le dossier en question ;
4. exécuter la commande `exec loader.sce` ; SIPTtoolbox est chargé, et utilisable ; vous pouvez commencer par les commandes `imread('uneimage.jpg')` ; `imshow(img, [0, 256])` ;

²Les instructions sont pour un environnement Linux, l'environnement Windows de l'Utès étant beaucoup plus contraignant.