Université Paris-Dauphine
Département MIDO
DE MI2E deuxième année

Méthodes numériques
année 2013/2014

feuille de travaux pratiques

Séance 4 : quelques premières applications du calcul scientifique

Le symbole $\diamond$ indique un exercice optionnel. Le travail demandé peut être effectué indifféremment avec MATLAB ou le logiciel libre GNU OCTAVE (`http://www.gnu.org/software/octave/`).

Exercice 1 (calcul d'une valeur approchée de π par la méthode de Monte-Carlo[1]). Pour obtenir une valeur approchée du nombre π par la méthode de Monte-Carlo, on tire « au hasard »[2], dans un carré de côté de longueur égale à 2, des points de coordonnées (x, y) et l'on vérifie s'ils appartiennent ou non au disque de rayon 1 et de centre le centre du carré. Ces points pouvant être tirés avec la même probabilité dans l'ensemble du carré, le rapport entre le nombre de points tirés dans le disque et le nombre de points tirés au total tend, lorsque le nombre de tirages tend vers l'infini, vers le rapport des surfaces du cercle et du carré, soit $\frac{\pi}{4}$, en vertu de la loi des grands nombres.

1. Au moyen de la commande `rand`, qui génère une suite de nombres réels jouant le rôle d'une réalisation d'une suite de variables aléatoires continues, indépendantes et identiquement distribuées selon la loi uniforme sur l'intervalle $[0, 1]$, écrire une fonction prenant comme argument le nombre de tirages à réaliser et renvoyant la valeur approchée de π obtenue par la méthode de Monte-Carlo décrite ci-dessus (pour simplifier, on pourra se restreindre au quart de carré contenu dans l'orthant positif de $\mathbb{R}^2$).
2. Donner un ordre du nombre de tirages nécessaires pour obtenir plus de deux décimales exactes de π. Que dire de l'efficacité de cette méthode ?

Exercice 2 $\diamond$ (applications linéaires entre espaces de polynômes). Soit n un entier naturel non nul. On note $\mathbb{R}_n[X]$ l'espace vectoriel des polynômes à coefficients réels de degré inférieur ou égal à n.

1. Pour P et Q deux éléments de $\mathbb{R}_n[X]$, on note $R(P, Q)$ le reste de la division euclidienne de P par Q. On considère l'application linéaire :

$$\begin{array}{cccc} \mathcal{R} & : & \mathbb{R}_n[X] & \to & \mathbb{R}_n[X] \\ & & P & \mapsto & R(P, X^2). \end{array}$$

   a. Écrire une fonction prenant comme paramètre un entier n et renvoie la matrice $M_{\mathcal{R}}$ de l'application $\mathcal{R}$ dans la base canonique $\{1, X, X^2, \dots, X^n\}$ de $\mathbb{R}_n[X]$.

   b. À l'aide de cette fonction, calculer le reste de la division par x^2 pour les polynômes suivants :

      (i) $7\,x^8 + 411\,x^7 - 231\,x^5 + 31\,x^4 + 451\,x^3 - 231\,x - 42$,

      (ii) $x^7 + \dfrac{5}{21}\,x^5 + 0,432\,x^4 - 22\,x^3 + 51\,x^2 - \dfrac{1}{39}\,x + 4,431$.

   c. En utilisant la commande `null`, déterminer le noyau de $\mathcal{R}$ pour $n = 6$, 7 et 8. Que constate-t-on ?

1. On appelle méthode de Monte-Carlo toute méthode visant à calculer une approximation numérique par utilisation d'un procédé aléatoire. Le nom de ce type de méthode, qui fait allusion aux jeux de hasard pratiqués dans le célèbre casino d'un des quartiers de la cité-État de la principauté de Monaco, a été inventé en 1947 par Nicholas Metropolis et publié pour la première fois en 1949 dans un article co-écrit avec Stanislas Ulam.
2. C'est-à-dire selon une loi de probabilité uniforme.

2. On considère à présent l'application linéaire dérivée :

$$\begin{array}{rccc} \mathcal{D} & : & \mathbb{R}_n[X] & \to & \mathbb{R}_n[X] \\ & & P & \mapsto & P', \end{array}$$

associant à tout polynôme P de $\mathbb{R}_n[X]$ ayant pour coefficients a_k, $k = 0, \ldots, n$, le polynôme P' ayant pour fonction polynomiale $P'(x) = \sum_{k=1}^{n} k\, a_k\, x^{k-1}$. Reprendre la question précédente avec $\mathcal{D}$ en place de $\mathcal{R}$.

3. On considère enfin les deux applications composées $\mathcal{D} \circ \mathcal{R}$ et $\mathcal{R} \circ \mathcal{D}$. Que font-elles ? Sont-elles identiques ? Quel est leur lien avec les produits de matrices $M_{\mathcal{D}} M_{\mathcal{R}}$ et $M_{\mathcal{R}} M_{\mathcal{D}}$?

Exercice 3 (procédé d'orthonormalisation de Gram–Schmidt). On rappelle que, partant d'une famille $\mathcal{B} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$ de vecteurs linéairement indépendants de $\mathbb{R}^n$, avec m et n des entiers tels que $2 \leq m \leq n$, le procédé d'orthonormalisation de Gram–Schmidt permet de construire une famille $\mathcal{B}' = \{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_m\}$ de vecteurs orthonormaux donnés par

$$\boldsymbol{q}_1 = \frac{\boldsymbol{x}_1}{\|\boldsymbol{x}_1\|},$$

$$\widetilde{\boldsymbol{q}}_{k+1} = \boldsymbol{x}_{k+1} - \sum_{i=1}^{k} (\boldsymbol{x}_{k+1}, \boldsymbol{q}_i)\, \boldsymbol{q}_i,\ \boldsymbol{q}_{k+1} = \frac{\widetilde{\boldsymbol{q}}_{k+1}}{\|\widetilde{\boldsymbol{q}}_{k+1}\|},\ k = 1, \ldots, m-1.$$

1. Écrire une fonction nommée `gramschmidt`, prenant comme paramètre d'entrée une matrice ayant pour colonnes les m vecteurs de la famille $\mathcal{B}$ et retournant en sortie une matrice ayant pour colonnes les m vecteurs de la famille $\mathcal{B}'$, obtenue en appliquant à $\mathcal{B}$ le procédé d'orthonormalisation de Gram–Schmidt. Penser à inclure une procédure vérifiant que la famille $\mathcal{B}$ fournie lors de l'appel de la fonction est bien libre.

2. On pose $\varepsilon = 10^{-8}$. Tester la fonction `gramschmidt` avec la famille

$$\mathcal{B} = \left\{ \begin{pmatrix} 1 \\ \varepsilon \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ \varepsilon \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \\ \varepsilon \end{pmatrix} \right\},$$

puis vérifier que les vecteurs obtenus sont bien orthogonaux deux à deux. Que constate-t-on ?

3. Pour pallier les défauts d'orthogonalité observés des vecteurs de la famille $\mathcal{B}'$ (qui sont dûs à l'accumulation des erreurs d'arrondi, les calculs étant faits en arithmétique à virgule flottante), il faut utiliser une version plus stable du procédé. Celle-ci consiste à opérer de la manière suivante :

$$q_1 = \frac{\boldsymbol{x}_1}{\|\boldsymbol{x}_1\|},$$

$$\boldsymbol{q}_{k+1}^{(0)} = \boldsymbol{x}_{k+1},\ \boldsymbol{q}_{k+1}^{(i)} = \boldsymbol{q}_{k+1}^{(i-1)} - \left(\boldsymbol{q}_{k+1}^{(i-1)}, \boldsymbol{q}_i\right) \boldsymbol{q}_i,\ i = 1, \ldots, k,\ \boldsymbol{q}_{k+1} = \frac{\boldsymbol{q}_{k+1}^{(k)}}{\|\boldsymbol{q}_{k+1}^{(k)}\|},\ k = 1, \ldots, m-1.$$

Mettre en œuvre, en modifiant la fonction déjà existante, cette variante pour obtenir une nouvelle fonction qu'on nommera `modgramschmidt` (pour procédé de Gram–Schmidt modifié). Effectuer ensuite l'orthonormalisation de la famille $\mathcal{B}$ donnée dans la question précédente.