Calcul Formel & Courbes et Surfaces

Durée : 2 h le lundi 14 mai
Documents et calculatrices de l'ULCO autorisés
Les deux parties du devoir seront traitées sur des copies différentes.

Calcul Formel

Exercice 1 (10 points). *Supposons que nous ayons une classe (ou structure)* `Matrice` *permettant de représenter les matrices carrées à coefficients dans* $\mathbb{Z}$ *(les entiers sont représentés par le type* `int`*). Les fonctions utiles sont:*

- `Matrice M(n)` qui crée une matrice de taille $\texttt{n} \times \texttt{n}$;
- `M[i][j]` qui accède au coefficient à la position $(\texttt{i}, \texttt{j})$ de la matrice `M`, les indices commençant à 1;
- `M.taille` qui retourne la taille de `M`.

Une matrice carrée $A = (a_{i,j})$ *de taille* $n \times n$ *est dite symétrique si* $a_{i,j}$ *est égale à* $a_{j,i}$ *pour tous les couples* (i, j) *de* $\{1, ..., n\}^2$.

– **1)** Parmi les matrices suivantes, lesquelles sont symétriques :

$$A_1 = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix},\ A_2 = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix},\ A_3 = \begin{bmatrix} 1 & 1 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix},\ A_4 = \begin{bmatrix} 2 & -1 & 1 \\ -1 & 1 & 2 \\ 1 & 2 & -1 \end{bmatrix}$$

– **2)** Ecrire en pseudo-langage ou en `C++` une fonction `est_symétrique(Matrice A)` qui retourne `true` si `A` est symétrique et `false` sinon.

Soit $A = (a_{i,j})$ *une matrice carrée, la matrice transposée de* A*, notée* ${}^tA$ *est la matrice* ${}^tA = (a_{j,i})$. *Exemple :*

$$A_5 = \begin{bmatrix} 2 & 1 & 3 \\ 3 & 2 & 1 \\ 1 & 3 & 2 \end{bmatrix} \quad et \quad {}^tA_5 = \begin{bmatrix} 2 & 3 & 1 \\ 1 & 2 & 3 \\ 3 & 1 & 2 \end{bmatrix}$$

– **3)** Calculer la transposée des matrices A_1, A_2, A_3 et A_4.

– **4)** Ecrire en pseudo-langage ou en `C++` une fonction `transposee(Matrice A)` qui retourne la transposée de la matrice `A`.

On dit qu'une matrice carrée A *est un carré magique si les sommes des coefficients sur chaque ligne et chaque colonne sont égales.*

– **5)** Trouver les trois carrés magiques parmi A_1, A_2, A_3, A_4 et A_5.

– **6)** Ecrire en pseudo-langage ou en `C++` les fonctions `int somme_ligne(Matrice A,int i)` et `int somme_colonne(Matrice A,int j)` qui retourne respectivement la somme des coefficients de la ligne `i` de `A` et la somme des coefficients de la colonne `j` de `A`.

– **7)** Ecrire en pseudo-langage ou en `C++` une fonction `est_carré_magique(Matrice A)` qui retourne `true` si `A` est un carré magique et `false` sinon.

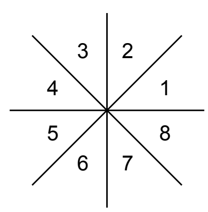
Exercice 2 (10 points)**.** Les éléments de $\mathbb{F}_2^n$ pourront être notés sous forme de n-uplets ou de vecteurs colonnes. Soit φ le code correcteur défini par

$$\begin{array}{rccc} \varphi : & \mathbb{F}_2^2 & \to & \mathbb{F}_2^5 \\ & (b_1, b_2) & \mapsto & (b_1, b_2, b_1 + b_2, b_2, b_1) \end{array}$$

- **1)** Quel est le paramètre du code φ?
- **2)** Quelle est l'image du code φ.
- **3)** Quelle est la distance minimale de φ?
- **4)** Quelles sont les capacités de détection et de correction pour φ ?
- **5)** Le code φ est-il linéaire ? systématique ?
- **6)** Le code φ est-il MDS ?
- **7)** Donner la matrice génératrice de φ.
- **8)** Donner une matrice de contrôle pour φ.
- **9)** Calculer la table de décodage de φ.
- **10)** Décoder le mot 01010.

Courbe et Surfaces

Exercice 3 (10 points)**.** *Dans cette exercice, on se propose d'adapter l'algorithme de Bresenham au cas d'un cercle. On note $\mathcal{C}_R$ le cercle de centre $O = (0, 0)$ et de rayon R. Comme pour la version vue en cours on divise l'espace en octant. On se contentera de tracer la partie du cercle se trouvant dans le second octant :*

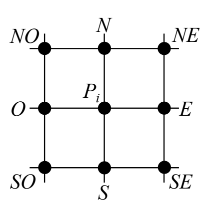


L'équation cartésienne de $\mathcal{C}_R$ est $F_R(x, y) = x^2 + y^2 - R^2$. Un point $M = (x_M, y_M)$ appartient alors à $\mathcal{C}_R$ si et seulement si $F_R(x_M, y_M) = 0$.

– **1)** Quelle est le signe de $F_R(x_M, y_M)$ pour un point $M = (x_M, y_M)$ se trouvant à l'intérieur du cercle $\mathcal{C}_R$. On pourra considérer que le signe est le même pour tous les points à l'intérieur du cercle de $\mathcal{C}_R$ et calculer la valeur de F_R en un point particulier.

Supposons que l'on trace dans le sens des aiguilles d'une montre la portion du cercle $\mathcal{C}_R$ se trouvant dans le deuxième octant. L'algorithme de Bresenham consiste alors à allumer successivement des points $P_0, P_1, P_2, \ldots.$ On note (x_i, y_i) les coordonnées du point P_i.

– **2)** Après avoir allumé le point P_i quels sont les deux pixels possibles parmi N, NE, E, SE, S, SO, O et NO pour P_{i+1}? Faire un dessin peut être très utile.



On note $M = (x_M, y_M)$ le point se trouvant au milieu du segment reliant les deux points possibles pour P_{i+1} et Q l'intersection de ce même segment avec le cercle $\mathcal{C}_R$. On choisira alors pour P_{i+1} le point du même côté que Q par rapport à M comme pour l'algorithme de tracé de segment vu en cours.

– **3)** Calculer les coordonnées (x_M, y_M) de M en fonction de (x_i, y_i).

– **4)** En fonction du signe de $F_R(x_M, y_M)$ déduire successivement, la position de M par rapport au cercle $\mathcal{C}_R$, la position de Q par rapport à M, les cordonnées (x_{i+1}, y_{i+1}) du point P_{i+1}.

– **5)** Donner une expression δ_i qui soit du même signe que $F_R(x_M, y_M)$ mais nécessitant uniquement des calculs sur les entiers. On cherchera une expression développer de δ_i de la forme $K \times F_R(x_M, y_M)$ où K est un entier ne dépendant ni de R ni de i.

– **6)** En fonction du choix pour P_{i+1} donner une formule liant δ_i à δ_{i+1}.

On suppose pour la suite que la fonction `allume(int x,int y)` *allume à l'écran le pixel de coordonnées (x, y).*

– **7)** Ecrire en pseudo-langage ou en `C++` une fonction `void dessine_cercle(int R)` qui dessine à l'écran le cercle de centre $(0, 0)$ et de rayon `R`.

Exercice 4. *Dans cet exercice, on se propose de calculer une courbe d'interpolation passant par les points $P_0 = (0, 1)$, $P_1 = (1, 0)$ et $P_2 = (2, 1)$. Le couple (x_i, y_i) désigne les coordonnées de P_i pour $i = 0, 1, 2$.*

Nous utilisons dans un premier temps un polynôme d'interpolation de Lagrange

– **1)** Calculer les polynômes $L_0(X), L_1(X)$ et $L_2(X)$ de $\mathbb{Q}[X]$ tels que $L_0(0) = 1$, $L_0(1) = 0$, $L_0(2) = 0$, $L_1(0) = 0$, $L_1(1) = 1$, $L_1(2) = 0$, $L_2(0) = 0$, $L_2(1) = 0$ et $L_2(2) = 1$.

– **2)** En déduire le polynôme $P(X)$ vérifiant $P(0) = 1$, $P(1) = 0$ et $P(2) = 1$.

Nous utilisons maintenant une spline quadratique

– **3)** Rappeler la défintion d'une spline qudaratique.

On note q_0 et q_1 les polynômes constituant la spline quadratique d'interpolation de P_0, P_1 et P_2

– **4)** Quelles sont les équations que doivent vérifier les polynômes q_i.

Pour $i = 0, 1$, on pose $q_i(X) = a_i(x - x_i)^2 - b_i(x - x_i) + c_i$

– **5)** Quelles sont les équations que doivent satisfaire a_0, a_1, b_0, b_1, c_0 et c_1. ?

– **6)** Quel est le nombre d'équations ? d'inconnues ? combien manque t-il d'équations ?

– **7)** Calculer les valeurs $a_0, a_1, b_0, b_1, c_0, c_1$ puis les polynômes q_0 et q_1 en ajoutant la condition initiale $q_0'(x_0) = 0$.