Université de Paris 7 **TP03** **agreg 2012-2013 (F.Han [1])**

Syntaxes du jour à retenir : time, dessins, avoir une idée la meilleure façon de calculer un déterminant selon la situation, evaluer un symbole en une valeur voulue (ie transformer un symbole en une fonction. cf II-1.c). Méthode pour faire de nombreux essais.

Exercice I: Déterminants. (Illustrations et Calculs)

Etudiez dans la documentation html de xcas les différentes options de la fonction déterminant, et les différents algorithmes possibles.

1) Quelle option correspond au pivot de Gauss usuel ? Estimez le nombre d'opérations sur le corps de base pour le pivot de Gauss avec la notation $O()$.

a) Etudiez l'instruction `time` et comment n'obtenir qu'un seul nombre en retour.

b) Donnez 2 exemples de situations classiques où le coût d'une multiplication (resp division...) peut être modélisé par une constante.

c) Technique pour faire de nombreux essais : affectez un variable. Ex `n:=10;` puis créez des matrices de taille $n \times n$ aléatoires à coefficients dans $\mathbb{Z}/101\mathbb{Z}$. Calculez ce déterminant par la méthode du pivot usuel déjà implémentée dans xcas. puis affichez le temps. Augmentez n pour tenter d'illuster l'asymptotique (ne pas perdre de temps, c'est souvent tres difficile à illustrer). Comparez avec la méthode des mineurs et montrez ses limites. Qui l'emporte rapidement dans ce modèle ? Observez dans l'aide html-algorithmes la partie sur les déterminants. Montrez que vos réponses sont des expressions [2] identiques pour justifier que vous comparez la même chose. Calculez avec xcas la somme formelle : $\sum_{k=0}^{n} \binom{n}{k}$.

2) Créez une assez grande matrice à coefficients entiers de valeurs absolue au plus 10. Comparez le pivot de Gauss usuel à celui de Bareiss. Comparez ensuite avec `det(M)` sans options. Quel est cette fois l'algorithme utilisé (Cf nombres premiers applications) ?

3) a) Essayer maintenant `n:=4` et `purge(a)`; (pour être sur d'avoir une variable formelle) `A:=matrix(n,n,(u,v)->a[u,v])`. (Retenir cette syntaxe). Que fait `d1:=det(A,rational_det)` par rapport à `d2:=det(A,bareiss)` ? Comparez aussi les différentes facons de développer : `simplify`, `normal`, `expand`.

Pour les simplifications rationnelles lourdes, il on préférera toujours normal aux autres instructions.

b) Avec $n = 5$ ou 6 comparez le pivot usuel, celui par bareiss et la méthode des mineurs. Conclusion [3] ?

Exercice II: Polynôme minimal. (Cf Problèmes de rang et applications)

1) a) Comment créer un polynôme à partir de la liste de ses coefficients. (NB : Pour xcas, un polynôme n'est autre que la liste des coefficients. Ex : $x^2 + 1$ est un symbole pour xcas.

b) Comment renverser une liste ?

c) Comment évaluer un polynôme lorsqu'il est sous forme d'un symbole. Comment le faire à partir de la liste de ses coefficients. Soit v un vecteur et u un endomorphisme de E (E ev de dim n). On note $P_{u,v}$ un générateur de l'idéal : $\{P \in k[x], P(u)(v) = 0\}$.

2) Expliquer sur un exemple de taille n aléatoire (on commencera par $n = 8$)comment on peut trouver $P_{u,v}$ à partir d'une recherche de noyau.

3) Recommencer sur un exemple où $P_{u,v}$ est de degré inférieur à la dimension de E.

a) Optimiser le calcul des $A^k(v)$, et regardez si le polynôme obtenu annule A. Estimer le nombre d'opérations sur le corps de base.

b) Justifier l'interet de cette méthode pour trouver le polynôme minimal de u.

4) Etudiez dans la documentation de xcas les différentes options pour calculer le polynôme caractéristique. Expliquez l'intéret de l'option pmin. Illustrez sur des exemples.

---

1. http ://www.math.jussieu.fr/~han/agreg
2. En revanche, comment procerder pour montrer une egalité mathématique ? Ex : que retourne a+b==b+a
3. Avec des matrices un peu plus grandes mais contenant pas mal de 0 on observe aussi ce genre de phénomène.

Exercice III: Matrices $GL(\mathbb{Z})$-équivalentes

[4] (Cf par exemple leçon "opérations lignes/colonnes") L'algorithme permettant d'obtenir les facteurs invariants : On le fait sur $\mathbb{Z}$ ou dans le cadre Euclidien, car on n'a pas d'algorithme pour obtenir les nombres de Bezout dans le cadre principal.

Cet algorithme est déjà programmé dans xcas sur $\mathbb{Z}$. Quelle est cette commande (retenir le nom Cf livres en anglais) ? On pourrait aussi le faire dans $k[x]$, mais nous nous limiterons ici à $\mathbb{Z}$. Nous allons procéder en quelques étapes. 1a) Trouver un minimum non nul (au sens du stathme) de la matrice. 1c) par opérations lignes/colonnes on fait diminuer les autres coefficients de sa ligne et de sa colonne. 2) On recommence 1a) et 1c) tant que c'est possible. lorsque l'on a fini, le minimum n'a que des 0 sur sa ligne et sur sa colonne. On garde cette valeur, on raye la ligne et la colonne, et l'on recommence avec la matrice plus petite. On obtient alors une matrice diagonale équivalente (dans l'anneau) à la matrice de départ, ce qui est déjà bien. (Quel théorème sur les groupes abélien de type fini cela prouve t'il ?) 3) On peut passer de cette forme aux diviseurs élémentaires par lemme chinois, ou en continuant l'algorithme de maniere un peu modifiée. On aurait pu aussi modifier l'algorithme avant de rayer une ligne et une colonne de manière à n'effectuer cette étape que lorsque le min divise tous les coefficients de la matrice.

1) a) Faire une procédure `minval` qui donne le couple (i, j) tel que $A_{i,j}$ soit un coefficient de la matrice A ayant la plus petite valeur absolue non nulle. (Elle pourra éventuellement retourner $(-1, -1)$ si A est la matrice nulle, et l'on utilisera les indices du logiciel. Ex en mode xcas on commence à 0.)

b) Prendre [5] un exemple à coefficients entiers noté A. Repérer son plus petit coefficient non nul : $A_{i,j}$ en valeur absolue, et créer une matrice $U \in GL(\mathbb{Z})$ telle que $(A.U)_{i,l}$ soit le reste de la division de $A_{i,l}$ par $A_{i,j}$ pour $l \neq j$, et $(A.U)_{i,j} = A_{i,j}$.

c) Faire une procédure `trans(A,i,j)` qui pour une matrice A telle que $A_{i,j} \neq 0$ donne une matrice A', $GL(\mathbb{Z})$-équivalente à A, telle que $\forall l \in \{1 \ldots n\} - \{j\}$, $|A'_{i,l}| < |A_{i,j}|$ et $\forall l \in \{1 \ldots n\} - \{i\}$ $|A'_{l,j}| < |A_{i,j}|$.

2) a) Essayer [6] de programmer la partie qui trouve une matrice diagonale $GL(\mathbb{Z})$-équivalente à A. On appellera cette procédure `Zequiv`. (On remarquera qu'une matrice dont les coefficients non nuls sont les seuls coefficients non nuls de leur ligne et de leur colonne est équivalente via des permutations à une matrice diagonale. Dans notre procédure on ne cherchera pas à trouver cette permutation.)

b) Obtient t'on forcément les diviseurs élémentaires ?

c) Modifiez votre programme pour ne rayer une ligne et une colonne que dans le cas où le min divise tous les coefficients de la matrice.

---

4. $A \sim B \iff \exists P, Q \in GL(\mathbb{Z}),\ P^{-1}.A.Q = B$
5. On s'entraine à la main sur un exemple, pour mieux faire la procédure de la question suivante
6. Remarquer que la décroissance stricte oblige l'algorithme à se terminer, ce qui donne une preuve du résultat suivant : Tout groupe abélien de type fini se décompose en produit de groupes monogènes. Du même genre : tout groupe abélien fini est isomorphe à un produit de groupe cyclique.