Université Bordeaux 1 — **Agrégation**
Mathématiques — Année 2019–2020

**FEUILLE D'EXERCICES n° 8**

# Algorithme d'Euclide et applications

**Exercice 1** – [Algorithme d'Euclide]
Programmer l'algorithme d'Euclide `AE` et l'algorithme d'Euclide étendu `AEE`. À l'aide de votre fonction AEE, programmer la fonction `Inverse(k,n)` qui pour deux entiers $k$ et $n$ premiers entre eux calcule l'inverse de $k$ modulo $n$.

Bien sûr, ce n'est qu'un exercice : sur sage, on peut par exemple utiliser la commande `mod(k,n)**(-1)`, ou bien `mod(k,n)**(-1).lift` si l'on veut le résultat dans $\mathbb{Z}$.

**Exercice 2** – [Autour de Bézout]

**1)** Le logiciel sage permet de calculer le pgcd de deux entiers à l'aide de la commande `gcd`. La commande `xgcd` donne en plus les coefficients de Bézout. En utilisant cette commande, écrire un algorithme qui prend en entrée une famille d'entiers $(a_1, \dots, a_n)$ et donne en sortie $(u_1, \dots, u_n) \in \mathbb{Z}^n$ tel que

$$\sum_{i=1}^{n} a_i u_i = \mathrm{pgcd}(a_1, \dots, a_n).$$

**2)** À partir du résultat donné par l'algorithme d'Euclide étendu, trouver une matrice $U$ de déterminant $\pm 1$ telle que $(a, b)U = (\mathrm{pgcd}(a, b), 0)$.

**3)** En s'inspirant de la question précédente, montrer qu'il existe une matrice $U$ dans $\mathrm{GL}(n, \mathbb{Z})$ (entière, de déterminant $\pm 1$) telle que

$$(a_1, \dots, a_n)U = (\mathrm{pgcd}(a_1, \dots, a_n), 0, \dots, 0),$$

et programmer le calcul de cette matrice. [*ne traiter que 2 coordonnées à la fois*]

**Note.** Cette question est une version "concrète" (d'un cas particulier) du théorème des diviseurs élémentaires pour les modules de type fini sur les anneaux principaux [*appliqué au dual* $(\mathbb{Z}^n)^*$ *et au sous* $\mathbb{Z}$*-module engendré par la forme linéaire* $(a_1, \dots, a_n)$]. Les diviseurs élémentaires sont donnés par le membre de droite.

Programmer le calcul de $U$ [*se rappeler l'astuce de la matrice identité auxiliaire dans l'algorithme du pivot de Gauss*]. Que dire de sa première colonne ? Si $b$ et $(a_1, \dots, a_n)$ sont des entiers fixés, expliquer comment résoudre l'équation $\sum a_i x_i = b$ en nombre entiers $(x_i)$ [*intercaler* $UU^{-1} = \mathrm{Id}$]. Résoudre les équations $1009x + 345y + 56z = 1$ et $143x + 195y + 165z = 3$.

**4)** Comment résoudre un système de plusieurs équations sur $\mathbb{Z}^n$ ? [*la question précédente permet de se ramener à un système triangulaire*]

**Exercice 3** – [Restes Chinois]

**1)** Résoudre les systèmes

$$\begin{cases} x \equiv -1 \mod 21 \\ x \equiv 11 \mod 15 \\ x \equiv 1 \mod 10 \end{cases} \text{ et } \begin{cases} x \equiv -1 \mod 21 \\ x \equiv 10 \mod 15 \\ x \equiv 1 \mod 10 \end{cases}$$

(on pourra utiliser la commande `crt`).

**Note.** Cette commande `crt` s'applique à tout anneau où l'algorithme des restes chinois s'applique, par exemple à $k[x]$, où $k$ est un corps.

On connaît bien l'algorithme correspondant dans le cas où les modules sont deux à deux premiers entre eux. Il ne s'applique pas tel quel aux exemples précédents. Les questions suivantes portent sur le cas général.

**2)** Soient $a$ et $b$ deux entiers $> 0$. On considère le système d'inconnue $N$

$$\begin{cases} N \equiv \alpha \pmod a \\ N \equiv \beta \pmod b \end{cases}$$

et on pose $\delta = \text{pgcd}(a, b)$, puis $u$ et $v$ deux entiers tels que $au + bv = \delta$.

a) Montrer que le système n'a pas de solution si $\alpha \not\equiv \beta \pmod \delta$.

b) Sinon, montrer que

$$N := \alpha + u\frac{a}{\delta}(\beta - \alpha) = \beta + v\frac{b}{\delta}(\alpha - \beta) = u\frac{a}{\delta}\beta + v\frac{b}{\delta}\alpha$$

convient. Montrer que cette solution est unique modulo $ab/\delta$.

**Note.** c'est une généralisation du "théorème chinois" au cas ou les modules ne sont pas nécessairement premiers entre eux. Si $\delta = \text{pgcd}(a, b) = 1$, on trouve bien un isomorphisme (explicite) entre $\mathbb{Z}/a\mathbb{Z} \times \mathbb{Z}/b\mathbb{Z}$ et $\mathbb{Z}/ab\mathbb{Z}$ (qui à $(\alpha, \beta)$ associe $N$).

**3)** En déduire un algorithme pour résoudre un nombre quelconque de congruences simultanées, et le programmer. Il s'agit là d'un exercice. Pour ce calcul, il faudra ensuite utiliser la commande `crt`.

**4)** Après une série de rapines, une troupe de 14 pirates partage (équitablement) le butin et laisse le reliquat, 3 écus, au cuisinier chinois, le $15^{\text{ème}}$ homme d'équipage. Le lendemain, un flibustier tombe à la mer et n'est pas repéché à temps ; après avoir envisagé le versement de sa part à des œuvres, les pirates refont le partage en incluant sa part ; le cuistot reçoit 2 écus. La semaine se passe sans encombres, mais trois pirates ivres se disputent sur leurs parts respectives et deux d'entre eux sont tués. Notre cuisinier récupère 5 écus. La fin du mois est mauvaise et 3 pirates périssent dans une embuscade ; mais le cuistot est content : il garde ses 5 écus.

Quel magot peut-il espérer empocher quand il décide d'empoisonner le reste de la bande ?

**Exercice 4** – [Interpolations de Lagrange et Hermite]

**1)** En utilisant la commande `lagrange_polynomial`, déterminer le polynôme $P$ de $\mathbb{Q}[x]$ de degré inférieur où égal à 3 tel que $P(0) = 1$, $P(1) = -1$, $P(2) = -3$, $P(3) = 2$.

**2)** Même exercice pour le polynôme $P$ de $\mathbb{F}_7[x]$ de degré inférieur où égal à 3 tel que $P(0) = 2$, $P(1) = -3$, $P(2) = -1$, $P(3) = 1$.

**3)** Considérant que $P(a) = b$ signifie que $P \equiv b \mod x - a$, retrouver le résultat précédent en utilisant `crt`.

**4)** On peut étendre ceci à l'interpolation de Hermite : trouver le polynôme de $\mathbb{Q}[x]$ de degré minimal tel que $P(0) = 1$, $P(1) = 2$, $P'(1) = -2$, $P''(1) = 0$, $P'''(1) = 1$, $P(2) = -1$, $P'(2) = 1$.
*Indication.* On rappelle que si $P$ est un polynôme de $\mathbb{K}[x]$ de degré $n$ et si $a$ appartient au corps $K$,

$$P(x) = \sum_{k=0}^{n} \frac{P^{(k)}(a)}{k!}(x-a)^k$$

C'est la formule de Taylor pour les polynômes (qui s'obtient en appliquant la formule du binôme à $((x-a)+a)^i$ pour tout $i$).

**Exercice 5** – [Déterminant modulaire]
Soit $A \in M_n(\mathbb{Z})$. On se propose de calculer $\det A$ de façon modulaire, c'est-à-dire de calculer $\det A \pmod p$ à partir des $a_{i,j} \pmod p$, pour des $p$ premiers et petits, et d'en déduire la valeur de $\det A$. Pour cela il faudra prendre des $p$ dont le produit est plus grand que $2|\det A|$ et se servir du lemme chinois. On rappelle l'inégalité de Hadamard :

$$|\det A|^2 \leqslant \prod_{j=1}^{n} \Big( \sum_{i=1}^{n} a_{i,j}^2 \Big),$$

qui aidera à déterminer une famille de $p$ adaptée.

**1)** Écrire une procédure admettant en entrée $A$ et qui détermine successivement une famille appropriée de premiers, les déterminants modulaires puis le déterminant de $A$. Faire attention aux fonctions $\texttt{mod}(*, d)$ ou $*\%d$ qui renvoient un entier de l'intervalle $[0, d\,[$ et non de $]-\frac{d}{2}, \frac{d}{2}]$.

**2)** Tester sur $A \in M_{10}(\mathbb{Z})$ où les $a_{i,j}$ sont aléatoires et vérifient $|a_{i,j}| < 100$. Étendre les tests.

**Exercice 6** – [Approximants de Padé]
On rappelle ici l'algorithme d'Euclide étendu appliqué à deux polynômes $F$ et $G \in K[X]$ où $K$ est un corp commutatif.

**Algorithme 1.** Algorithme d'Euclide étendu

**Entrées:** $F$, $G \in K[X]$
**Sorties:** pgcd$(F,G)$ et $A$, $B \in K[X]$ tels que $AF + BG =$pgcd$(F,G)$
1: $A_0 = 1$, $B_0 = 0$, $R_0 = F$
2: $A_1 = 0$, $B_1 = 1$, $R_1 = G$
3: $i = 1$ $\quad \{initialisations\}$
4: **tantque** $R_i \neq 0$ **faire**
5: Division de $R_{i-1}$ par $R_i \to$ quotient $Q$ et reste $R_{i+1}$
6: $A_{i+1} = A_{i-1} - QA_i$
7: $B_{i+1} = B_{i-1} - QB_i$
8: $i = i + 1$
9: Retourner le dernier $R_i$ non nul ainsi que les $A_i$ et $B_i$ correspondants

On rappelle également qu'à chaque étape de l'algorithme, si $n_k = \deg R_k$, on a

(1) $A_iF + B_iG = R_i$

(2) $\deg A_i = n_1 - n_{i-1}$ (pour $i > 1$)

(3) $\deg B_i = n_0 - n_{i-1}$ (pour $i > 0$)

**1)** Soient $\mathcal{F} = f_0 + f_1X + f_2X^2 + \cdots \in K[X] \setminus \{0\}$ et $m, n \in \mathbb{N}$.
On appelle *approximant de Padé de type* $(m,n)$ de $\mathcal{F}$ un élément $(U,V) \in K[X]^2$ vérifiant

$$\begin{cases} V \neq 0, \deg U \leqslant m \text{ et } \deg V \leqslant n \\ V\mathcal{F} - U = X^{m+n+1}R, \text{ où } R \in K[[X]]. \end{cases}$$

a) En posant $F = f_0 + \cdots + f_{m+n}X^{m+n}$ et en appliquant l'algorithme d'Euclide étendu à $F$ et $X^{m+n+1}$, montrer qu'un tel approximant existe.

b) Soient $(U_0, V_0)$ et $(U, V)$ deux approximants de Padé de type $(m,n)$ de $\mathcal{F}$ tels que $\begin{vmatrix} V_0 & U_0 \\ V & U \end{vmatrix} \neq 0$. Que peut on dire sur le degré de ce déterminant ?

c) En déduire que le quotient $U/V$ d'un tel approximant de Padé est uniquement déterminé.

**2)** Soit $\mathcal{F}$ une fraction rationnelle de $K(X)$ quotient de deux polynômes $U$ et $V$ de $K[X]$ de degrés $\leqslant n$ et premiers entre eux mais que l'on ne connaît pas. Supposons que l'on connaisse en revanche un développement en série formelle de $\mathcal{F}$ en 0 :

$$\mathcal{F} = f_0 + f_1X + f_2X^2 + \cdots$$

Comment à partir de ce développement retrouver $U$ et $V$ à une constante multiplicative près ? Ce problème est appelé le problème de la reconstruction rationnelle.

**3)** Programmer une fonction qui résout ce problème et la tester sur des exemples de quotients.

**4)** On considère le problème suivant. On connaît un certain nombre $s$ de termes consécutifs d'une suite $(u_n)$ satisfaisant une relation de récurrence d'ordre $k$ :

$$u_{n+k} = a_1 u_{n+k-1} + \cdots + a_k u_n.$$

On suppose que $s = 2k$.

a) Soient

$$A = 1 - \sum_{i=1}^{k} a_i X^i \quad \text{et} \quad \mathcal{F} = \sum_{i \geqslant 0} u_i X^i$$

Montrer que $A\mathcal{F}$ est un polynôme de degré inférieur ou égal à $k - 1$.

b) En déduire un algorithme permettant de calculer $A$ et programmer cet algorithme.

**5)** Essayer avec

— 1, 1, . . . ;
— 3, 5, 8, 13, . . . ;
— 8, 13, 21, 34, 55, 89, 114 . . .
— 1, 1, 1, 1, 10, 19, 46, 118, . . .