# Rapport de TP du module Arithmétique Modulaire

Ziane Ceryne

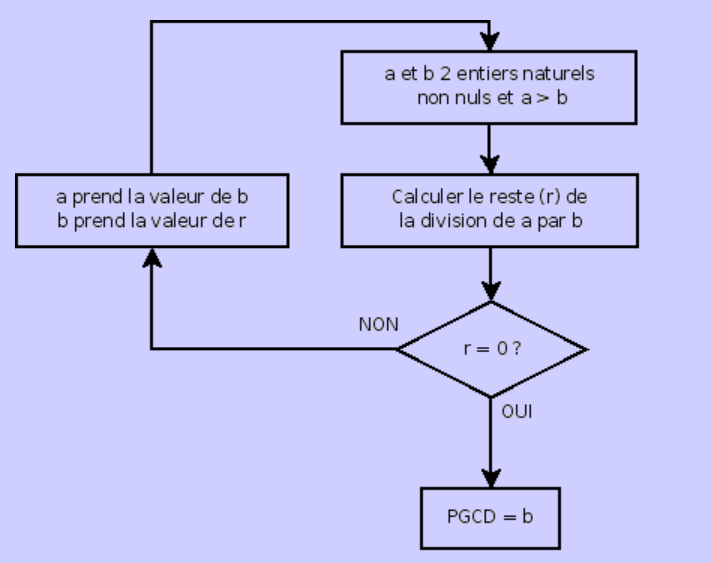
M1 SSI 07.01.2024

### TP1: Algorithme d’Euclide

L'algorithme d'Euclide est une méthode efficace pour déterminer le Plus Grand Commun Diviseur (PGCD) de deux entiers naturels, qui est un diviseur commun à ces deux nombres.

**Les étapes de l'algorithme:**

1. Effectuer une division euclidienne du plus grand des deux nombres, noté a, par le second, noté b.
2. Trouver le quotient q et le reste r.
3. Conserver les valeurs de b et r.
4. Répéter les étapes 1 à 3 en utilisant les valeurs actuelles de b comme le nouveau a et r comme le nouveau b.
5. Continuez ce processus de division euclidienne jusqu'à ce que le reste r soit égal à zéro.
6. Le PGCD de a et b de départ est égal au dernier reste non nul obtenu lors de l'étape 4.



### TP2: Identité de Bézout

L'algorithme d'Euclide étendu, une extension de l'algorithme d'Euclide traditionnel, est une suite de divisions euclidiennes qui permet de déterminer les constantes du théorème de Bachet-Bézout. Ce théorème énonce que pour deux entiers relatifs non tout les deux nuls, a et b, il existe deux autres entiers relatifs, u et v, tels que **au + bv = PGCD (a, b)**. Les valeurs de u et v sont connues sous le nom de coefficients de Bézout.

L'algorithme d'Euclide étendu fonctionne en itérant sur une série de divisions euclidiennes successives jusqu'à ce que le reste devienne nul. À chaque étape, les coefficients de Bézout sont calculés pour exprimer le PGCD en termes des entiers d'origine a et b.

**Implémentation en Python de l'algorithme d'Euclide étendu:**

# Algorithme d'Euclide étendu - Version Itérative

def bezout\_identity(a, b):

u = 0; old\_u = 1

v = 1; old\_v = 0

r = b; old\_r = a

while r != 0:

q = old\_r // r

old\_r, r = r, old\_r - q \* r

old\_u, u = u, old\_u - q \* u

old\_v, v = v, old\_v - q \* v

return [old\_r, old\_u, old\_v]

### TP3: Ensemble des diviseurs positifs d’un entier naturel et Indicatrice d’Euler

1. **Ensemble des Diviseurs Positifs :** Un diviseur d'un entier n est un nombre entier d tel que la division euclidienne de n par d donne un reste r égal à zéro. Mathématiquement, cela se formule comme **n = d × q**, où q est un entier et r=0. L'ensemble des diviseurs positifs d'un entier naturel n est alors constitué de tous les entiers positifs qui divisent n.
2. **Indicatrice d'Euler (φ) :** L'indicatrice d'Euler, notée **φ(n)**, est définie comme le nombre d'entiers positifs inférieurs à n qui sont premiers avec n. Cela se traduit par le nombre d'entiers k tels que 1≤k≤n et **PGCD(n,k) = 1**.

Exemples: φ(1) = 1, φ(2) = 1, φ(3) = 2, et pour tout nombre premier p, φ(p) = p−1. De manière générale, pour tout nombre premier p et tout entier m≥1, on a:

**φ(pm) = pm − pm−1 = pm−1(p − 1) = pm(1 − 1/p)**

Pour déterminer la valeur de l'indicatrice d'Euler pour un entier n > 0, on utilise la décomposition en facteurs premiers de n, qui consiste à écrire n sous forme d'un produit de nombres premiers.

### TP4: Génération des nombres aléatoires

Un nombre aléatoire est un nombre sélectionné de manière imprévisible à partir d'un ensemble défini de nombres, comme l'indique son nom. Dans une distribution spécifiée, chaque nombre a une probabilité égale d'être choisi au hasard.

Pour qu'un nombre soit considéré comme aléatoire, deux conditions doivent être remplies :

- **uniformité** : les valeurs doivent être uniformément réparties dans un intervalle ou un ensemble déterminé.

- **indépendance**: il doit être impossible de prédire les valeurs futures en se basant sur les valeurs passées ou présentes.

Les nombres aléatoires jouent un rôle crucial dans des domaines tels que l'analyse statistique, la théorie des probabilités, les simulations informatiques modernes, la cryptographie numérique et la gestion des portefeuilles de crypto-monnaie.

1. **Loi uniforme**
2. Générateurs congruentiels linéaire (GCL)

Les (GCL) sont des algorithmes déterministes, considérés comme l’un des méthodes de base les plus efficaces pour produire une séquence de nombres pseudo-aléatoires, suivant une distribution uniforme sur l'intervalle [0, 1]. La séquence générée, notée Un = Xn / m, représente une suite où la séquence d'entiers Xi prend ses valeurs entre 0 et m - 1.

Le GCL fonctionne selon une relation de récurrence définie comme suit :

**Xn+1 = (a × Xn + c) mod m**, où a, c, m sont des entiers, m étant un nombre premier très grand : m > a, c, X0.

**Xn :** le nombre pseudo-aléatoire à l'étape n

**X0 :** la valeur initiale ou graine

**a :** le coefficient multiplicatif

**c :** le terme increment constant, premier avec m

**m:** le module, déterminant taille de la sequence.

Un GCL est dit mixte lorsque c > 0 et multiplicatif lorsque c = 0

La fonction de densité de probabilité de la distribution uniforme dans l’intervalle [a, b) est définie comme : **P(x) = 1 / (b - a)**.

Source : <https://www.apprendre-en-ligne.net/random/ChapVI.pdf>

1. **Échantillon exponentielle**

La loi exponentielle est souvent utilisée pour modéliser le temps entre des événements continus et indépendants, tels que les intervalles de temps entre les arrivées de clients dans une file d'attente, durée de vie de composants électroniques, tremblement de terre...

Pour générer des nombres aléatoires selon une distribution exponentielle, la méthode de l’inverse est employée. Ayant pour densité de probabilité la fonction f définie sur [0;+∞[ par : **f (x) =λe−λx avec x > 0**. Soit λ un réel strictement positif et x une variable aléatoire, la fonction de répartition est décrite comme suit : **F(x) =1 - e−λx**

L'échantillonnage inverse consiste à inverser cette fonction pour obtenir la variable aléatoire x à partir d'une variable aléatoire uniforme U dans l'intervalle [0, 1]. L'inverse de la fonction F, **F−1(U) = -ln(1 - U) / λ**

Paramètre de taux (λ) :

Dans le contexte de la distribution exponentielle, λ mesure la fréquence des événements qui est inversement proportionnel au temps d'attente moyen entre les événements. Si λ est grand, les événements se produisent plus fréquemment, entraînant un temps d'attente moyen plus court. Si λ est petit, les événements se produisent moins fréquemment, ce qui se traduit par un temps d'attente moyen plus long.

Nombre d'échantillons :

Ce paramètre détermine le nombre d'échantillons aléatoires qu’on souhaite générer à partir de la distribution exponentielle.

**Implémentation en python:**

import numpy as np

def exponential(n, lam):

    # Générer la séquence U avec la loi uniforme

    U = np.random.uniform(0, 1, n)

    print(f"Données générer avec la loi uniforme: {U}")

    # F^−1(U) = -ln(1 - U) / λ Échantillonnage inverse pour trouver x

    X = -np.log(1 - U) / lam

    print(f"Données échantillonnées: {X}")

    return X

# Paramètres de la fonction de distribution exponentielle

lam = 0.5  # le taux lambda

nbr\_echant = 10 # taille de la sequence de nbr aléatoires à générer

# Appel à la fonction de distribution exponentielle

X = exponential(nbr\_echant, lam)

# f (x) =λe(−λx) Calcule de la densité

fd = lam \* np.exp(-lam \* X)

print(f'Densité de Probabilité: {fd}')

**Output:**

Données générer avec la loi uniforme: [0.02196741 0.69410499 0.51251445 0.85583979 0.18899512 0.88411203 0.8146171 0.44652292 0.38684367 0.93528709]

Données échantillonnées: [0.04442457 2.36902665 1.43698926 3.87366002 0.41896241 4.31026268 3.37066377 1.18306988 0.9782707 5.47558901]

Densité de Probabilité: [0.4890163 0.15294751 0.24374278 0.07208011 0.40550244 0.05794398 0.09269145 0.27673854 0.30657816 0.03235646]

Source: <https://numpy.org/doc/stable/reference/random/generated/numpy.random.exponential.html>

1. **Loi de Poisson**

La loi de Poisson est une loi de probabilité discrète, qui estime le nombre de fois un événement peut se produire dans un laps de temps spécifié. Cette distribution est souvent utilisée pour générer des nombres aléatoires dans des situations où les événements se produisent de manière indépendante et à un taux constant.

La fonction de probabilité de la loi de poisson est définit ainsi:

IMG_256

où :

**X :** la variable aléatoire représentant le nombre d'événements.

**k :** le nombre d'événements, k >= 0.

**λ** : le nombre moyen d'événements dans un intervalle de temps.

Source: <https://ressources.univ-lemans.fr/AccesLibre/UM/Pedago/physique/02/divers/loipoisson.html>