

MODÉLISATION MATHÉMATIQUE ET INFORMATIQUE

Durée : 3 heures

L'usage d'une calculatrice est autorisé pour cette épreuve.

Chaque candidate ou candidat est responsable de la vérification de son sujet d'épreuve : pagination et impression de chaque page. Ce contrôle doit être fait en début d'épreuve. En cas de doute, il convient d'alerter au plus tôt l'équipe de surveillance qui vérifiera et, éventuellement, remplacera le sujet.

Ce sujet comporte 10 pages numérotées de 1 à 10.

Si, au cours de l'épreuve, une candidate ou un candidat repère ce qui lui semble être une erreur d'énoncé, elle ou il le signale sur sa copie et poursuit sa composition en expliquant les raisons des initiatives prises.

Le sujet se compose d'un problème en trois parties portant sur la modélisation de l'expression génétique bactérienne. La première partie présente une première approche déterministe. La seconde partie propose une seconde façon de modéliser cette situation, en utilisant une approche probabiliste et un algorithme appelé l'algorithme de Gillespie. La troisième partie fait le lien entre ces deux approches.

Les parties 1 et 2 sont parfaitement indépendantes. La partie 3 dépend des deux premières. Les résultats des deux premières parties pourront être admis pour traiter la partie 3.

Les questions 15, 16, 20, 21, 23 et 24 sont des questions algorithmiques utilisant le langage Python. Un rappel de certaines fonctions Python est donné en annexe.

MODÉLISATION DE L'EXPRESSION GÉNÉTIQUE BACTÉRIENNE

Notations

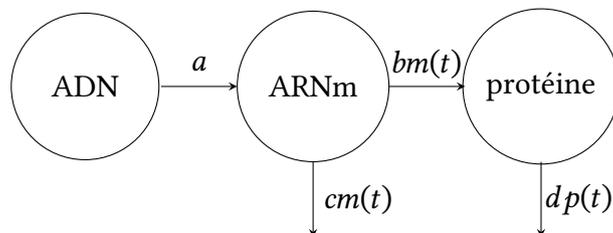
On notera $\mathcal{E}(\lambda)$ la loi exponentielle de paramètre $\lambda > 0$ c'est-à-dire la loi de la variable aléatoire de densité

$$f_\lambda : [0, +\infty[\rightarrow \mathbb{R} \\ x \mapsto \lambda \exp(-\lambda x)$$

Phénomène biologique

Nous proposons ici deux manières de modéliser l'expression génétique bactérienne chez *Escherichia coli*, une famille de bactéries naturellement présente dans les intestins mais responsable de nombreuses infections humaines. Pour rappel à la première étape de l'expression génétique, l'ADN est **transcrit** en ARN messager sans dégradation de l'ADN. L'ARN messager devient ensuite accessible aux ribosomes qui s'y lient afin de former des protéines. Ce processus est appelé **la traduction**, il s'effectue sans dégradation de l'ARN messager. L'ARN messager et les protéines peuvent enfin être **dégradés**.

On modélise le phénomène d'expression génétique par le schéma suivant où sont notés les flux :



Les quantités seront exprimées en picogrammes et les constantes a , b , c et d seront strictement positives.

- $m(t)$ correspond à la quantité d'ARN messenger
- $p(t)$ correspond à la quantité de protéine
- a correspond au flux constant de la réaction de **transcription**
- b correspond à la constante de vitesse de la réaction de **traduction** de l'ARN messenger en protéine
- c correspond à la constante de vitesse de la réaction de **dégradation de l'ARN messenger**
- d correspond à la constante de vitesse de la réaction de **dégradation de la protéine**

On s'intéressera ici au cas particulier d'une protéine répressive, ou **protéine Rep**, une protéine-clé dans la régulation de la duplication cellulaire (mitose) et donc dans la survie de la population bactérienne. On supposera que la mitose n'est possible que si la quantité de protéine Rep dépasse une **valeur seuil** p^* et pourra être compromise par l'ajout d'un **antibiotique bactériostatique** dans le milieu (Sous-Partie 3.1).

Dans tout le sujet le temps est continu, exprimé en seconde et sera noté t .

1. Un premier modèle : un modèle déterministe

1.1. Étude de la solution

Dans cette partie on supposera que les quantités d'ARN messenger m et de protéine p sont des fonctions \mathcal{C}^1 du temps $t \in \mathbb{R}_+$. On notera $m_0 = m(0)$ et $p_0 = p(0)$ les conditions initiales. La situation peut alors être modélisée par le système suivant :

$$\begin{cases} \frac{dm}{dt} = a - cm \\ \frac{dp}{dt} = bm - dp \end{cases} \quad (1)$$

On supposera enfin que $c \neq d$.

Q 1. Quelles sont les unités des constantes a , c , b et d ?

Q 2. Interpréter qualitativement ce qu'il se passe si on augmente la constante c .

On admettra que la forme générale des solutions est :

$$m(t) = \left(m_0 - \frac{a}{c}\right)e^{-ct} + \frac{a}{c} \quad (2)$$

et

$$p(t) = \left(p_0 - \frac{b(m_0 - \frac{a}{c})}{d - c} - \frac{ab}{cd}\right)e^{-dt} + \frac{b(m_0 - \frac{a}{c})}{d - c}e^{-ct} + \frac{ab}{cd} \quad (3)$$

Q 3.1. Calculer la limite m_∞ de $m(t)$ quand le temps t tend vers l'infini.

Q 3.2. Calculer l'équation de la tangente à la courbe représentative de m en zéro.

Q 3.3. Tracer l'allure de la fonction m sur $[0, +\infty[$ dans le cas où $m_0 > \frac{a}{c}$ et dans le cas où $m_0 < \frac{a}{c}$.

Q 4.1. Calculer la limite p_∞ de $p(t)$ quand le temps t tend vers l'infini et interpréter cette quantité.

Q 4.2. Supposons le rapport a/d constant. Décrire le comportement de p_∞ en fonction de la valeur du rapport b/d et interpréter.

Q 4.3. Supposons le rapport b/d constant. Décrire le comportement de p_∞ en fonction de la valeur du rapport a/c et interpréter.

1. 2. Résolution en découplant par changement de base

Dans cette sous-partie nous prendrons les valeurs numériques suivantes :

$$b = 1 ; c = 1 ; d = 2$$

et nous proposons une méthode de résolution du système (1), c'est-à-dire de retrouver les fonctions données en (2) et (3), en utilisant un changement de base.

Q 5. Soit $\vec{x}(t) = \begin{pmatrix} m(t) \\ p(t) \end{pmatrix}$. Justifier que le système (1) s'écrit sous la forme :

$$\frac{d\vec{x}}{dt} = A\vec{x} + B \quad (4)$$

avec $A = \begin{pmatrix} -1 & 0 \\ 1 & -2 \end{pmatrix}$ et $B = \begin{pmatrix} a \\ 0 \end{pmatrix}$.

Q 6.1. Justifier que la matrice A est diagonalisable et donner ses valeurs propres λ_1 et λ_2 ainsi que deux vecteurs propres associés dont la coordonnée sur la deuxième composante vaut 1.

Q 6.2. En déduire une matrice P inversible telle que :

$$A = P \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} P^{-1} \quad (5)$$

Q 6.3. Calculer P^{-1} .

Q 7. Soit $\vec{z} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = P^{-1}\vec{x}$. Montrer que $\vec{z} = \begin{pmatrix} m(t) \\ -m(t) + p(t) \end{pmatrix}$. En déduire que \vec{z} vérifie le système différentiel suivant :

$$\begin{cases} \frac{dz_1}{dt} = -z_1 + a \\ \frac{dz_2}{dt} = -2z_2 - a \end{cases} \quad (6)$$

Q 8. Résoudre chacune des équations du système (6).

Q 9. Montrer que $\vec{x} = \begin{pmatrix} z_1 \\ z_1 + z_2 \end{pmatrix}$. En déduire que :

$$\begin{cases} m(t) = (m_0 - a)e^{-t} + a \\ p(t) = (m_0 - a)e^{-t} + \left(p_0 - m_0 + \frac{a}{2}\right)e^{-2t} + \frac{a}{2} \end{cases} \quad (7)$$

Q 10. Par quelle autre méthode aurait-on pu résoudre le système (1) ?

2. Un second modèle : un modèle probabiliste

2. 1. Introduction du modèle

Dans cette partie on ne modélisera plus les quantités d'ARNm et de protéines par des fonctions continues du temps mais comme des variables aléatoires M_t et P_t dépendant du temps et prenant leurs valeurs dans l'ensemble des entiers \mathbb{N} . Le flux d'ADN est toujours constant et on le supposera dorénavant entier et strictement positif : $a \in \mathbb{N}^*$. On supposera également qu'il existe une suite de variables aléatoire T_1, T_2, \dots modélisant des instants telle que :

- La suite $(T_k)_{k \in \mathbb{N}}$ est strictement croissante : $T_1 < T_2 < \dots$
- Aucune réaction n'a lieu aux temps $t \in]T_{k-1}, T_k[$ pour tout $k \in \mathbb{N}^*$.
- Une (unique) réaction a lieu à chaque instant $T_k, k \in \mathbb{N}$ parmi les réactions suivantes :
 - A : « Transcription d'un brin d'ADN »
 - B : « Traduction d'un brin d'ARNm »
 - C : « Dégradation d'un brin d'ARNm »
 - D : « Dégradation d'une protéine »

Q 11. *Exemple* : supposons que sur une expérience où $m_0 = 5$ et $p_0 = 10$ on ait obtenu les réactions suivantes :

Temps (seconde)	42	55	77	79	100	126	138	151	188
Réaction	A	B	A	C	C	A	B	D	A

Q 11.1. Quelle quantité d'ARNm et de protéine a-t-on après 3 minutes ?

Q 11.2. Représenter entre les instants $t = 0$ et $t = 200s$ l'évolution des fonctions $t \mapsto M_t$ et $t \mapsto P_t$ pour cette expérience-ci. Ces courbes sont appelées des **trajectoires**.

Q 12. Exprimer M_{T_k} en fonction de $M_{T_{k-1}}$ et P_{T_k} en fonction de $P_{T_{k-1}}$ si la réaction ayant lieu à l'instant T_k est A, B, C puis D .

2. 2. Tirage aléatoire de la réaction

Avertissement : Dans toute cette partie les fonctions Python demandées seront supposées précédées des lignes suivantes qui définissent les valeurs des paramètres du modèle et des conditions initiales pour les applications numériques, qu'il ne sera pas nécessaire de rappeler par la suite.

```
1 from random import random
2
3 m0 = 80
4 p0 = 20
5 a = 2
6 b = 1
7 c = 1
8 d = 3
```

On définit le **taux de réaction** à l'instant t de la façon suivante :

$$R_t = a + (b + c)M_t + dP_t \quad (8)$$

On suppose alors qu'à chaque instant T_k , $k \in \mathbb{N}$ la réaction ayant lieu est choisie parmi A, B, C ou D avec les probabilités suivantes :

$$\mathbb{P}(A) = \frac{a}{R_t} \quad (9)$$

$$\mathbb{P}(B) = \frac{bM_t}{R_t} \quad (10)$$

$$\mathbb{P}(C) = \frac{cM_t}{R_t} \quad (11)$$

$$\mathbb{P}(D) = \frac{dP_t}{R_t} \quad (12)$$

Q 13. Proposer une interprétation pour le taux de réaction R_t .

Q 14. Comment évolue la probabilité de l'évènement B lorsque M_t augmente à P_t constant ? Expliquer qualitativement ce qu'il se passe.

Q 15. L'objectif de cette question est d'écrire une fonction Python qui permet de simuler une variable aléatoire X suivant une loi discrète à support fini $\{a_0, \dots, a_{I-1}\}$ avec $I \in \mathbb{N}^*$. Pour tout $i \in \llbracket 0, I-1 \rrbracket$ on notera :

$$\mathbb{P}(X = a_i) = p_i \text{ avec } \sum_{i=0}^{I-1} p_i = 1.$$

Q 15.1. La fonction `mystere` simule une réalisation d'une variable aléatoire, laquelle ?

```

1 def mystere(p) :
2     X = random()
3     if X >= 1-p :
4         return 1
5     else:
6         return 0

```

Q 15.2. Soit U une variable aléatoire suivant une loi uniforme sur $[0,1]$. Montrer que pour tout $i \in \llbracket 0, I-1 \rrbracket$ on a :

$$\mathbb{P}(X = a_i) = \mathbb{P}\left(U \in \left[\sum_{j=0}^i p_j, \sum_{j=0}^{i+1} p_j\right]\right) \quad (13)$$

Comment généraliser la stratégie de simulation de la question précédente au cas d'une loi discrète à support fini de taille I quelconque ? On pourra accompagner la réponse d'un schéma explicatif.

Q 15.3. Compléter la fonction suivante qui recevra en paramètre d'entrée une liste `liste_p` contenant I probabilités p_i et qui doit renvoyer le numéro $i \in \llbracket 0, I-1 \rrbracket$ de l'évènement tiré au hasard.

```

1 def loi_discrete_finie(liste_p):
2     I = len(liste_p)
3     X = random()
4     pcumul = 0
5     for i in range(I) :
6         pcumul = ???
7         if ??? :
8             return ???

```

Q 16. En déduire une fonction Python `reaction`, prenant en entrée deux arguments M_t et P_t , tirant aléatoirement un évènement parmi A, B, C, D et qui renvoie le caractère correspondant ('A', 'B', 'C' ou 'D').

2. 3. Modélisation de la durée entre deux réactions

Soit $k \in \mathbb{N}^*$. Supposons que $k-1$ réactions aient eu lieu et notons t_{k-1} l'instant où a eu lieu la $(k-1)$ -ième réaction et $r_{t_{k-1}}$ le taux de réaction associé.

On pose $\Delta_k = T_k - t_{k-1}$ la variable aléatoire de la durée avant la réaction suivante. On supposera que Δ_k suit une loi exponentielle de paramètre $r_{t_{k-1}}$:

$$\Delta_k = T_k - t_{k-1} \sim \mathcal{E}(r_{t_{k-1}}) \quad (14)$$

et que les Δ_k sont indépendantes.

Q 17.1. Montrer que pour tout $t, s > 0$ l'équation suivante est vérifiée :

$$\mathbb{P}_{\Delta_k > t}(\Delta_k \geq t + s) = \mathbb{P}(\Delta_k \geq s) \quad (15)$$

Q 17.2. Justifier le choix de modéliser cette situation par une loi exponentielle.

Q 18. Donner (sans calcul) l'espérance de Δ_k et justifier le choix dans cette modélisation de choisir $r_{t_{k-1}}$ comme paramètre.

Q 19. Soit $\mu > 0$. Soit U une variable aléatoire de loi uniforme sur $[0,1]$ et soit $\Delta = -\frac{\ln(U)}{\mu}$. Montrer que Δ suit la loi exponentielle de paramètre μ : $\Delta \sim \mathcal{E}(\mu)$. On pourra calculer la fonction de répartition de Δ .

Q 20. Écrire une fonction Python `duree` qui permet de simuler le temps entre deux réactions qui suit une loi exponentielle de paramètre R_t . Cette fonction prendra en paramètres d'entrée les valeurs `Mt` et `Pt` de M_t et P_t . On pourra utiliser la fonction `random()` de l'annexe.

2. 4. Simulation de trajectoire - Algorithme de Gillespie

Cette partie réutilise les fonctions `reaction` et `duree` des **Sous-Parties 2.2** et **2.3** respectivement ainsi que le résultat de la question 12. On pourra admettre ces résultats.

Nous proposons de simuler une trajectoire de M_t et de P_t via l'algorithme suivant, dit de **Gillespie** :

- **Entrée** : `t_max`
- Initialiser une liste `liste_T` pour stocker les instants des réactions, initialisée à `[0]`
- Créer deux listes `liste_M` et `liste_P` initialisées à `[m_0]` et `[p_0]` respectivement
- Tant que le temps ne dépasse pas `t_max` :
 1. Simuler le temps avant la prochaine réaction via la fonction `duree`
 2. Stocker le dernier instant de réaction dans la liste `liste_T`
 3. Tirer aléatoirement une réaction parmi `A`, `B`, `C` et `D` via la fonction `reaction`
 4. Réactualiser `liste_M` et `liste_P` en fonction de la réaction tirée précédemment
- Modifier les différentes listes de sorte que le dernier instant coïncide avec `t_max`.
- **Sortie** : `[liste_T, liste_M et liste_P]`

Q 21. Compléter la fonction `trajectoire` suivante qui met en œuvre l'algorithme de Gillespie et préciser à quoi servent les lignes 24, 25 et 26.

```

1 def trajectoire(tmax):
2     liste_T = [0]
3     liste_M = [m0]
4     liste_P = [p0]
5     i = 0
6
7     while ...
8         Delta = ...
9         liste_T.append( ... )
10        typeR = ...
11        if typeR=='A' :
12            ...
13            ...
14        elif typeR=='B' :
15            ...
16            ...
17        elif typeR=='C' :
18            ...
19            ...
20        else :
21            ...
22            ...
23        i = i+1
24        liste_T[i] = tmax
25        liste_M[i] = liste_M[i-1]
26        liste_P[i] = liste_P[i-1]
27
28    return [liste_T,liste_M,liste_P]

```

3. Limite grande population

Dans cette partie nous continuerons de noter m et p les solutions du système déterministe (1) pour les conditions initiales fixées ainsi que m_∞ et p_∞ leur valeur limite quand t tend vers l'infini. Nous supposerons également qu'il existe n **trajectoires indépendantes** associées au modèle de la Partie 2 avec $n \in \mathbb{N}^*$, chacune de ces trajectoires correspondant à l'évolution de la quantité d'ARN et de protéine d'une cellule bactérienne. Pour tout $i \in \{1, \dots, n\}$ on notera $M_t^{(i)}$ et $P_t^{(i)}$ les valeurs de la i -ème trajectoire au temps $t \in \mathbb{R}_+$.

3.1. Propriétés théoriques

L'objectif des questions suivantes est de calculer les moyennes empiriques des trajectoires définies par

$$\overline{M}_t = \frac{1}{n} \sum_{i=1}^n M_t^{(i)} \quad (16)$$

$$\overline{P}_t = \frac{1}{n} \sum_{i=1}^n P_t^{(i)} \quad (17)$$

pour tout $t \in \mathbb{R}_+$ puis d'étudier leurs propriétés.

Q 22. En utilisant un schéma, expliquer quelle est la difficulté de faire les moyennes entre deux trajectoires tirées aléatoirement par la fonction trajectoire.

Q 23. Voici une fonction aligne. On suppose que dans les listes d'instant croissants contenus dans `liste_T1` et `liste_T2`, le premier instant est le même ainsi que le dernier instant.

```

1 def aligne(liste_T1,liste_M1,liste_P1,liste_T2,liste_M2,liste_P2):
2     i1 = 0
3     i2 = 0
4     liste_Ta = [ ]
5     liste_M1a = [ ]
6     liste_P1a = [ ]
7     liste_M2a = [ ]
8     liste_P2a = [ ]
9     while (i1<len(liste_T1)) and (i2<len(liste_T2)) :
10        if liste_T1[i1]<liste_T2[i2] :
11            liste_Ta.append(liste_T1[i1])
12            liste_M1a.append(liste_M1[i1])
13            liste_P1a.append(liste_P1[i1])
14            liste_M2a.append(liste_M2[i2-1])
15            liste_P2a.append(liste_P2[i2-1])
16            i1 = i1+1
17        elif liste_T1[i1]>liste_T2[i2] :
18            liste_Ta.append(liste_T2[i2])
19            liste_M1a.append(liste_M1[i1-1])
20            liste_P1a.append(liste_P1[i1-1])
21            liste_M2a.append(liste_M2[i2])
22            liste_P2a.append(liste_P2[i2])
23            i2 = i2+1
24        else :
25            liste_Ta.append(liste_T1[i1])
26            liste_M1a.append(liste_M1[i1])
27            liste_P1a.append(liste_P1[i1])
28            liste_M2a.append(liste_M2[i2])
29            liste_P2a.append(liste_P2[i2])
30            i1 = i1+1
31            i2 = i2+1
32    return [liste_Ta,liste_M1a,liste_P1a,liste_M2a,liste_P2a]

```

Q 23.1. Que fait la partie comprise entre les lignes 10 et 16 de la fonction aligne.?

Q 23.2. Que fait la partie comprise entre les lignes 24 et 29 fonction aligne? A quel moment est-on certain qu'elle est exécutée?

Q 23.3. En déduire ce que renvoie la fonction aligne

Q 24. Soit la fonction suivante moyennes, incomplète, censée calculer les moyennes pour n trajectoires.

```

1 def moyennes(tmax,n):
2     for i in range(n):
3         [liste_T,liste_M,liste_P] = trajectoire(tmax)
4         if i==0:
5             nbmoy = 1
6             Tmoy = liste_T
7             Mmoy = liste_M
8             Pmoy = liste_P
9         else:
10            [Tmoy,Mmoy,Pmoy,liste_M,liste_P]=aligne(Tmoy,Mmoy,Pmoy,liste_T,liste_M,liste_P)
11            for i in range(len(Tmoy)):
12                Mmoy[i] = ...
13                Pmoy[i] = ...
14            nbmoy = nbmoy+1
15    return [Tmoy,Mmoy,Pmoy]

```

Compléter les lignes 12 et 13 de la fonction moyennes.

Q 25. On supposera admis que pour tout $i \in \mathbb{N}$:

$$\mathbb{E} \left[P_t^{(i)} \right] = p(t) \quad (18)$$

$$\text{var} \left[P_t^{(i)} \right] = \sigma^2 \quad (19)$$

avec $\sigma^2 > 0$. Interpréter ces égalités. Comment s'interprète une valeur élevée de σ dans ce contexte ?

Q 26. Calculer l'espérance et la variance de \bar{P}_t et sa limite quand n tend vers l'infini et interpréter les résultats en faisant le lien avec la Partie 1. Illustrer graphiquement la différence entre une valeur élevée de σ et une valeur de σ proche de 0.

3. 2. Application à la problématique de survie de la population de bactéries

On suppose que, pour une cellule bactérienne donnée, la capacité de la cellule à se reproduire dépend de la quantité de protéine Rep générée qui doit être supérieure à un seuil p^* . On supposera également que l'extinction de la population de taille n est assurée lorsque aucune cellule n'arrive à assurer une quantité de protéine supérieure à p^* à un certain temps.

On supposera enfin que l'ajout d'un antibiotique bactériostatique peut se modéliser par le remplacement du flux a par $ae^{-\theta}$, θ étant la quantité d'antibiotique administrée qui est strictement positive.

On se placera dans le cas où $d < c/2$ et on admettra que dans ce cas la fonction p est croissante.

Q 27. Expliquer qualitativement comment le comportement du système est modifié lorsque l'antibiotique est ajouté. Que se passe-t-il intuitivement lorsque θ tend vers $+\infty$?

Q 28. On supposera que le seuil p^* coïncide avec la valeur limite $p^* = \lim_{t \rightarrow \infty} p(t)$ (question (4)) d'une situation standard, c'est-à-dire sans l'ajout d'antibiotique. On notera p_∞ la valeur limite de la situation avec antibiotique. Justifier que :

$$p^* - p_\infty = \frac{ab(1 - e^{-\theta})}{cd} \quad (20)$$

Q 29. Justifier l'inégalité suivante pour tout $i \in \mathbb{N}^*$:

$$\mathbb{P} \left(|P_t^{(i)} - p(t)| \geq p^* - p(t) \right) \leq \frac{\sigma^2}{(p^* - p(t))^2} \quad (21)$$

puis

$$\mathbb{P} \left(P_t^{(i)} - p(t) \geq p^* - p(t) \right) \leq \frac{\sigma^2}{(p^* - p(t))^2} \quad (22)$$

Q 30. En déduire :

$$\mathbb{P} \left(P_t^{(i)} - p(t) \geq p^* - p(t) \right) \leq \left(\frac{\sigma cd}{ab(1 - e^{-\theta})} \right)^2 \quad (23)$$

Q 31. Montrer l'inégalité suivante :

$$\mathbb{P}(\text{Extinction}) \geq \left(1 - \left(\frac{\sigma cd}{ab(1 - e^{-\theta})} \right)^2 \right)^n \quad (24)$$

Q 32. Comment se comporte le terme de droite quand θ tend vers $+\infty$?

Q 33. Que dire de la probabilité d'extinction de la population de bactéries en fonction de σ ? A votre avis pourquoi ne retrouve-t-on pas l'hypothèse de la question 27. ? Donner une interprétation biologique au terme σ .

Annexe - Quelques fonctions Python

Sont utiles pour ce sujet :

```
1 from random import random
2
3 x = random()
```

La fonction `random` renvoie un tirage pseudo-aléatoire d'une loi uniforme sur $[0,1]$

```
1 import numpy as np
2
3 b = np.log(a)
4 d = np.exp(c)
```

Les fonctions `log` et `exp` renvoient respectivement le logarithme népérien et l'exponentielle.

FIN DU SUJET