Devoir : Statistiques, séries et dénombrement

Exercice : une formule combinatoire

① *Démonstration combinatoire.* On considère une urne U contenant $n+1$ boules dont b sont noires et $a+1$ sont blanches. On extrait les boules une à une jusqu'à vider l'urne.

a) *Combien de tirages différents sont possibles ?* Les $n+1$ boules, indiscernables entre elles, étant extraites, il y a autant de tirages possibles que de façon de placer les $a+1$ boules blanches parmi $n+1$ places possibles. Les boules noires sont alors placées.

Conclusion : $\boxed{\text{Card}(\Omega) = \binom{n+1}{a+1}}$

b) Soit M_k l'événement : « La dernière boule blanche occupe la $(k+1)$-ième place ». *Déterminons Card(M_k) pour des valeurs de k qu'on précisera :*
Les valeurs de k possibles sont prises dans l'ensemble $[\![a, n]\!]$.
La place de la dernière boule blanche étant fixée, il y a autant de cas favorables que de façon de placer les a boules restantes parmi les k places restantes.

Conclusion : $\boxed{\text{Card}(M_k) = \binom{k}{a}}$

c) Pour conclure, il suffit de dire que $\{M_k, k \in [\![a, n]\!]\}$ forme une partition de Ω.
Dès lors :

$$\text{Card}(\Omega) = \binom{n+1}{a+1} = \sum_{k=a}^{n} \binom{k}{a}$$

② Fournir la démonstration algébrique de cette égalité par télescopages.
On rappelle que d'après la formule de Pascal :

$$\binom{k}{a} + \binom{k}{a+1} = \binom{k+1}{a+1}$$

Donc :

$$\begin{aligned} S_n = \sum_{k=a}^{n} \binom{k}{a} &= \binom{a}{a} + \sum_{k=a+1}^{n} \left(\binom{k+1}{a+1} - \binom{k}{a+1} \right) \\ &= \binom{a}{a} + \binom{n+1}{a+1} - \binom{a+1}{a+1} = \binom{n+1}{a+1} \text{ par télescopages} \end{aligned}$$

Conclusion : $\boxed{\sum_{k=a}^{n} \binom{k}{a} = \binom{n+1}{a+1} = S}$

③ *Modélisation*

a) *Écrivons une fonction Python* `simulTirage(N,A)` *de paramètres d'entrée le nombres* `N` *($N = n+1$) de boules dans l'urne et le nombre* `A` *de blanches ($A = a+1$), et qui modélise le résultat d'un tel tirage en retournant une liste formée de* 0 *et de* 1*, selon que la boule est noire ou blanche :*

On choisit de numéroter les boules de 1 à $N = n+1$ et notamment les boules blanches de 1 à $A = a+1$.
On représente alors l'urne par la liste U des entiers compris entre 1 et N en écrivant : `U = range(1,N+1)`
Dès lors, la fonction `sample()` de la bibliothèque `random` permet de modéliser un tirage successif sans remise de $N = n+1$ boules, autrement dit de vider l'urne. Il suffit alors d'affecter 1 à chaque valeur de la liste dont le numéro est inférieur ou égale à A...

```
def simulTirage(N,A):
    U = range(1,N+1)
    Lb = rdm.sample(U,N)
    L = [0]*(N) # initialisation du tirage
    for k in range(len(Lb)):
       if Lb[k] <= A:
          L[k] = 1
    return L
```

b) *Écrivons une fonction* ***rangDerniereBlanche(L)*** *qui retourne k si la dernière boule blanche occupe la $(k+1)$-ième place (les places étant numérotées de 1 à $n+1$) :*

On parcourt la liste `L` en partant du dernier indice, à savoir `i = len(L)-1`.
Tant que `L[i]` n'est pas égale à 1, on recule d'un rang dans la liste et dès que `L[i]` vaut 1, c'est qu'on a obtenu une boule blanche et on retourne son rang.
✐ On notera que les indices k demandés par l'énoncé coïncident avec l'indexation Python des termes de la liste...
Par exemple, si `L = [1,0,1,1]`, alors la dernière boule blanche occupe la 4-ième place et on retourne $k = 3$ qui est son indice Python.
si `L = [1,1,0,0]`, alors la dernière boule blanche occupe la 2-ième place et on retourne $k = 1$ qui est bien son indice Python...

Une rédaction possible consiste à écrire :

```
def rangDerniereBlanche(L):
    i = len(L)-1
    while L[i] == 0:
       i -= 1
    return i
```

c) oit A_k l'événement : « la dernière boule blanche occupe la $(k+1)$-ième place ».
Que vaut $\mathbb{P}(A_k)$? La réponse est donnée par la question 1. car, les tirages étant équiprobables, il suffit de faire le rapport entre le nombre de cas favorables et le nombre de cas possibles. A savoir :

$$\mathbb{P}(A_k) = \frac{\binom{k}{a}}{\binom{n+1}{a+1}}$$

Écrivons une fonction ***estimProba(N,A,k,m)*** *d'arguments le nombre $N = n+1$ de boules dans l'urne, le nombre $A = a+1$ de blanches, l'indice* **k** *de l'événement et un entier* **m** *supposé grand (de l'ordre du millier) qui retourne une estimation de $\mathbb{P}(A_k)$, en faisant appel un grand nombre de fois à la fonction* ***simulTirage()*** *:*

```
def estimProba(N,A,k,m):
    # N = n+1 et A = a+1
    cpt = 0
    for i in range(m):
       L = simulTirage(N,A)
       r = rangDerniereBlanche(L)
       if r == k:
          cpt += 1
    return cpt/m
```

A titre d'exemple, pour $n = 10$, $a = 5$, $k = 10$ et $m = 10000$, `estimProba(n+1,a+1,k,m)` retourne `0.5491` alors que la probabilité de A_k vaut : `binom(k,a)/binom(n+1,a+1) = 0.5454`.

Problème 1 : Séries numériques

1. Étude de la somme d'une série

Pour tout entier naturel $n \geq 1$, on pose :

$$S_n = \sum_{k=1}^{n} \frac{1}{k^2}$$

① *Quel est le résultat de cours qui permet d'affirmer que la suite (S_n) converge ? Le montrer :*

Le cours nous assure que la série de terme général $\sum \frac{1}{n^2}$ converge. Autrement dit la suite (S_n) converge.
Démontrons-le :
$\forall k \geq 2,\ 0 < \frac{1}{k^2} \leq \frac{1}{k(k-1)}$ car $k^2 \geq k(k-1)$ et la fonction inverse est décroissante sur $\mathbb{R}_+^*$.

Or, si on pose $\sum \frac{1}{k(k-1)} = (T_n)$ où

$$T_n = \sum_{k=2}^{n} \frac{1}{k(k-1)} = \sum_{k=2}^{n} \left(\frac{1}{k-1} - \frac{1}{k} \right) = 1 - \frac{1}{n} \text{ par télescopage}$$

alors il est immédiat que la suite (T_n) converge vers 1, autrement dit,

$$\text{La série } \sum \frac{1}{k(k-1)} \text{ converge de somme égale à 1.}$$

Par théorème de convergence par comparaison des séries à termes positifs, on a donc :

Conclusion : $\boxed{\text{La séries } \sum \frac{1}{n^2} \text{ converge ou encore } (S_n) \text{ converge}}$.

Dans la suite du problème, on notera $l = \lim\limits_{n \to \infty} S_n$.

② *Écrivons une fonction Python* `sommeP(n)` *qui prend en argument n et qui renvoie la valeur de S_n :*

```
def sommeP(n):
    return sum([1/k**2 for k in range(1,n+1)])
```

③ a) Soit $f : t \longmapsto \frac{1}{t^2}$.
Pour répondre à la question, il suffit de dire que la fonction f est décroissante sur $\mathbb{R}_+$ et qu'en conséquence :

$$\forall t \in [k, k+1],\ f(t) \leq f(k)$$

En intégrant de chaque côté de l'inégalité, les bornes d'intégration étant dans le « bon »sens, on obtient :

$$\int_k^{k+1} f(t)dt \leq \int_k^{k+1} f(k)dt = f(k) \int_k^{k+1} dt = f(k)$$

Conclusion : $\boxed{\forall x \in [k, k+1],\ \int_k^{k+1} \frac{1}{t^2}dt \leq \frac{1}{k^2}}$

Montrons plus généralement que pour tout entier $k \geq 2$, on a : $\int_k^{k+1} \frac{1}{t^2}dt \leq \frac{1}{k^2} \leq \int_{k-1}^{k} \frac{1}{t^2}dt$
On a déjà obtenu l'inégalité de gauche. Il suffit donc d'obtenir celle de droite, et pour ça de noter grâce à la décroissance de f que :

$$\forall t \in [k-1, k],\ f(t) \geq f(k)$$

et par passage à l'intégrale, les bornes étant dans le bon sens :

$$\int_{k-1}^{k} \frac{1}{t^2}dt \geq \int_{k-1}^{k} f(k)dt = f(k) = \frac{1}{k^2}$$

Conclusion : $\boxed{\int_k^{k+1} \frac{1}{t^2}dt \leq \frac{1}{k^2} \leq \int_{k-1}^{k} \frac{1}{t^2}dt}$

b) Soit p et n deux entiers naturels non nuls.
démontrons que :

$$\int_{n+1}^{n+p+1} \frac{1}{t^2}dt \leq S_{n+p} - S_n \leq \int_n^{n+p} \frac{1}{t^2}dt$$

On commence par préciser que $S_{n+p} - S_n = \sum_{k=n+1}^{n+p} \frac{1}{k^2}$.
Pour conclure, il suffit donc de sommer l'inégalité obtenue à la question précédente qui donne :

$$\sum_{k=n+1}^{n+p} \int_k^{k+1} \frac{1}{t^2}dt \leq \sum_{k=n+1}^{n+p} \frac{1}{k^2} \leq \sum_{k=n+1}^{n+p} \int_{k-1}^{k} \frac{1}{t^2}dt$$

Par application de la relation de Chasles :

$$\sum_{k=n+1}^{n+p} \int_k^{k+1} \frac{1}{t^2}dt = \int_{n+1}^{n+p+1} \frac{1}{t^2}dt \text{ et } \sum_{k=n+1}^{n+p} \int_{k-1}^{k} \frac{1}{t^2}dt = \int_n^{n+p} \frac{1}{t^2}dt$$

Conclusion : $\boxed{\int_{n+1}^{n+p+1} \frac{1}{t^2}dt \leq S_{n+p} - S_n \leq \int_n^{n+p} \frac{1}{t^2}dt}$

c) *Démontrons que :*

$$\frac{1}{n+1} \leq l - S_n \leq \frac{1}{n}$$

Un calcul rapide d'intégrale donne :

$$\int_n^{n+p} \frac{1}{t^2}dt = \left[-\frac{1}{t}\right]_n^{n+p} = \frac{1}{n} - \frac{1}{n+p} \text{ et } \int_{n+1}^{n+p+1} \frac{1}{t^2}dt = \frac{1}{n+1} - \frac{1}{n+p+1}$$

On a donc d'après 3.b) :

$$\frac{1}{n+1} - \frac{1}{n+p+1} \leq S_{n+p} - S_n \leq \frac{1}{n} - \frac{1}{n+p}$$

Dès lors, pour tout entier naturel, on fait tendre p vers l'infini dans l'inégalité précédente et on obtient :

Conclusion : $\boxed{\frac{1}{n+1} \leq l - S_n \leq \frac{1}{n}}$

Multiplions de chaque côté de l'expression précédente par n : $\frac{n}{n+1} \leq n(l - S_n) \leq \frac{n}{n} = 1$.
On obtient alors par théorème d'encadrement des limites que :

$$\lim_{n\to\infty} n(l - S_n) = 1$$

Conclusion : $\boxed{l - S_n \underset{n\to\infty}{\sim} \frac{1}{n}}$

On nous demande enfin ce qu'on peut dire de $S_n + \frac{1}{n}$ pour n suffisamment grand. Attention au piège car il ne s'agit surtout pas d'utiliser l'équivalent précédent puisque toute addition est interdite.
Dès lors, on reprend la double inégalité du début de question pour encadrer S_n.
Soit :

$$l - \frac{1}{n} \leq S_n \leq l - \frac{1}{n+1}$$

et donc

$$l \leq S_n + \frac{1}{n} \leq l + \frac{1}{n} - \frac{1}{n+1} = l + \frac{1}{n(n+1)}$$

Par passage à la limite et application du théorème d'encadrement des limites, on obtient que $\lim_{n\to\infty} \left(S_n + \frac{1}{n}\right) = l$
Et comme $l \neq 0$, on peut écrire :

Conclusion : $\boxed{S_n + \frac{1}{n} \underset{n\to\infty}{\sim} l}$

④ On considère la fonction suivante :

```
def estimation(n):
   s = sommeP(n)
   s = sqrt(6*(s+1/n))
   return (s)
```

L'instruction `estimation(1000)` renvoie la valeur : 3.141593130895433.

Que peut-on conjecturer quant à la probable valeur de l ? La première ligne de la fonction `estimation()` affecte à `s` la valeur de S_n.
Or, d'après ce qui précède $s + \dfrac{1}{n} \underset{n\to\infty}{\sim} l$ donc par multiplication par 6, $6(s+1/n) \underset{n\to\infty}{\sim} 6l$.
La fonction $\sqrt{}$ étant continue sur $\mathbb{R}_+^*$, on peut dire que $\lim\limits_{n\to\infty} \sqrt{6(s+1/n)} = \sqrt{6l}$.
La fonction `estimation()` étant appelée avec une valeur de n suffisamment grande ($n = 1000$), on peut estimer que

$$\sqrt{6l} \approx \pi$$

Conclusion : On peut conjecturer que la probable valeur de l est $\pi^2/6$

2. Étude de la nature d'une série

Pour tout entier naturel $n \geq 1$, on pose :

$$u_n = \prod_{k=1}^{n} \left(1 + \frac{1}{k^2}\right) = (1+1)\left(1+\frac{1}{4}\right)\cdots\left(1+\frac{1}{n^2}\right)$$

① *Donnons, sous forme d'entiers ou de fractions simplifiées, les valeurs de u_1, u_2 et u_3 :*
Un calcul rapide donne $u_1 = 2$, $u_2 = u_1\dfrac{5}{4} = \dfrac{5}{2}$ et $u_3 = u_2\left(1+\dfrac{1}{9}\right) = \dfrac{5}{2}\dfrac{10}{9} = \dfrac{25}{9}$.

② a) *Montrons par récurrence que, pour tout entier naturel $n \geq 1$, on a : $u_n \geq 2$.*

- *initialisation :* $u_1 = 2 \geq 2$.
- On suppose que $u_n \geq 2$, pour n fixé ($n \geq 1$).
- Alors, sachant que $u_{n+1} = \prod\limits_{k=1}^{n+1}\left(1+\dfrac{1}{k^2}\right) = u_n\left(1+\dfrac{1}{(n+1)^2}\right)$,
  on a immédiatement $u_{n+1} \geq 2$ puisqu'il est bien évident que $1 + \dfrac{1}{(n+1)^2} \geq 1$...
- ***Conclusion*** : $u_n \geq 2, \forall n \geq 1$

b) *Étudions la monotonie de la suite (u_n) :* On exploite à nouveau que :

$$\frac{u_{n+1}}{u_n} = 1 + \frac{1}{(n+1)^2} > 1$$

Tous les termes de la suite (u_n) étant positifs puisque $\forall k \in \mathbb{N}^*$, $1 + \dfrac{1}{k^2} > 1$ et ln est bijective de $[1, +\infty[$ sur $\mathbb{R}_+$, on peut donc conclure : (u_n) est croissante

③ a) *Montrons que, pour tout réel x strictement supérieur à -1, on a : $\ln(1+x) \leq x$:*
Ce résultat peut se démontrer par application du théorème des accroissements finis. En effet :
La fonction $x \longmapsto \ln(1+x)$ est de classe $\mathcal{C}^1$ sur tout intervalle $[0, x] \subset]-1, +\infty[$. Dès lors :

$$\exists! c \in]0, x[/ \ln(1+x) = \frac{x}{1+c}$$

- Si $x > 0$, alors $0 < c < x \Rightarrow \dfrac{x}{1+c} < x$ et donc $\ln(1+x) < x$.

– Si $x < 0$, alors $x < c < 0 \Rightarrow \dfrac{1}{1+c} > 1 \Rightarrow \dfrac{x}{1+c} < x$ et donc $\ln(1+x) < x$.

Conclusion : $\boxed{\forall x > -1,\ \ln(1+x) \leq x}$ [en considérant le cas $x = 0$]

b) *Démontrons que, pour tout $n \in \mathbb{N}^*$, on a :* $\ln(u_n) \leq l$
Par définition de u_n et application de la fonction logarithme népérien, on obtient :

$$\ln(u_n) = \sum_{k=1}^{n} \ln\left(1 + \frac{1}{k^2}\right)$$

Par application de l'inégalité précédente, on en déduit :

$$\ln(u_n) \leq \sum_{k=1}^{n} \frac{1}{k^2} = S_n$$

avec $S_n \leq l$ puisque (S_n) est croissante (en effet $\sum \dfrac{1}{n^2}$ est une série à termes positifs...) et convergente et donc majorée par sa limite l en l'infini.

Conclusion : $\boxed{\forall n \in \mathbb{N}^*,\ \ln(u_n) \leq l}$

④ *Montrons que la suite (u_n) converge vers un réel l', élément de $[2, e^l]$:*
La suite (u_n) est croissante et de premier terme $u_1 = 2$ donc $\forall n \in \mathbb{N}^*,\ u_n \geq 2$.
Par ailleurs, $\ln(u_n) \leq l$ donc, par passage à l'exponentielle qui est une fonction croissante : $u_n \leq e^l$.

Conclusion : $\boxed{\forall n \in \mathbb{N}^*,\ u_n \in [2, e^l]}$

Il découle de cet résultat que (u_n) est croissante et majorée par e^l. $\boxed{\text{Elle converge vers } l' \in [2, e^l]}$.

⑤ On se propose dans cette question de déterminer la nature de la série de terme général $l' - u_n$.

a) On justifie que la suite $(\ln(u_n))_{n\in\mathbb{N}^*}$ converge en rappelant que $\ln(u_n) \leq l$ et que cette suite est croissante (en effet, $u_n \leq u_{n+1} \Rightarrow \ln(u_n) \leq \ln(u_{n+1})$ puisque la fonction ln est croissante sur $[0, +\infty[$). Elle est donc croissante et majorée.
enfin $\lim\limits_{n\to\infty} \ln(u_n) = \ln(l')$ car la fonction ln est continue sur $[2, e^l] \subset]0, +\infty[$. Soit :

$$\boxed{\ln(l') = \lim_{n\to\infty} \sum_{k=1}^{n} \left(1 + \frac{1}{k^2}\right) = \sum_{k=1}^{+\infty} \ln\left(1 + \frac{1}{k^2}\right)}$$

b) *Montrons que, pour tout $n \in \mathbb{N}^*$, on a :* $\ln\left(\dfrac{l'}{u_n}\right) = \sum\limits_{k=n+1}^{+\infty} \ln\left(1 + \dfrac{1}{k^2}\right)$

C'est immédiat :

$$\ln\left(\frac{l'}{u_n}\right) = \ln(l') - \ln(u_n) = \sum_{k=1}^{\infty} \ln\left(1 + \frac{1}{k^2}\right) - \sum_{k=1}^{n} \ln\left(1 + \frac{1}{k^2}\right) = \sum_{k=n+1}^{\infty} \ln\left(1 + \frac{1}{k^2}\right)$$

c) *Écrivons en Python une fonction* `reste` *qui prend en argument n et qui renvoie une bonne valeur approchée de :*

$$\sum_{k=n+1}^{\infty} \ln\left(1 + \frac{1}{k^2}\right)$$

Il s'agit ici de calculer le reste de la série convergente $\sum\limits_{k\geq 1} \ln\left(1 + \dfrac{1}{k^2}\right)$.

On peut se contenter de faire :

```
def reste(n):
    return np.sum([np.log(1+1/k**2) for k in range(n+1,10**6)])
```

d) Voici les résultats de quelques appels de cette fonction (supposée correctement écrite) :

```
> > > reste(9)
0.10497285546814518

> > > reste(49)
0.020198959583845037

> > > reste(99)
0.010048997486379833

> > > reste(199)
0.0050114998428795786

> > > reste(499)
0.0020009999955394315

> > > reste(999)
0.0009994999929165772
```

Que peut-on alors conjecturer comme équivalent en l'infini très simple de $\sum_{k=n+1}^{\infty} \ln\left(1+\frac{1}{k^2}\right)$ *?*

Et bien tout simplement que $\boxed{R_n = \sum_{k=n+1}^{\infty} \ln\left(1+\frac{1}{k^2}\right) \underset{n\to\infty}{\sim} \frac{1}{n}}$

e) En admettant le résultat obtenu en 5.d), on peut en déduire que $\lim\limits_{n\to\infty} nR_n = 1$.
Dès lors, d'après la définition de la limite,

$$\exists n_0 \in \mathbb{N} / \forall n \geq n_0,\ nR_n \geq 0,9$$

D'où, pour tout $n \geq n_0$, on a bien :

$$\boxed{R_n = \sum_{k=n+1}^{\infty} \ln\left(1+\frac{1}{k^2}\right) \geq \frac{0,9}{n}}$$

f) On rappelle que, grâce à la question 5.b), on a : $R_n = \ln\left(\frac{l'}{u_n}\right)$.

On peut donc dire que $\ln\left(\frac{l'}{u_n}\right) \geq \frac{0,9}{n}$, $\forall n \geq n_0$.

Par application de la fonction exponentielle qui est croissante, on a :

$$\frac{l'}{u_n} \geq e^{0,9/n} \Leftrightarrow l' \geq u_n e^{0,9/n} \text{ car } u_n > 0,\ \forall n \in \mathbb{N}^*$$

Conclusion : $\boxed{\forall n \geq n_0,\ l' - u_n \geq u_n(e^{0,9/n} - 1)}$

g) On admet que pour tout réel x, on a : $e^x - 1 \geq x$ (ce qui se démontrer également grâce au théorème des accroissements finis...).

On applique l'indication ci-dessus pour dire que : $e^{0.9/n} - 1 \geq \frac{0,9}{n}$.
Dès lors, d'après f)

$$l' - u_n \geq u_n \cdot \frac{0,9}{n} \geq \frac{1,8}{n} \text{ puisqu'on rappelle que } u_n \geq 2,\ \forall n \in \mathbb{N}^*$$

Conclusion : $\boxed{\forall n \geq n_0,\ l' - u_n \geq \frac{1,8}{n}}$

Concluons quant à la nature de la série de terme général $l' - u_n$ *:*

Nous savons que la série harmonique $\sum \frac{1}{n}$ diverge donc la série $\sum \frac{1,8}{n}$ diverge.
Par application du théorème de convergence par comparaison des séries à termes positifs, on a :

Conclusion : $\boxed{\text{La série } \sum(l' - u_n) \text{ diverge}}$

Problème 2 : Agro-Véto B 2018

D'après les travaux de Michaelis et Menten, lors d'une réaction chimique, un substrat, noté S, est catalysé par une enzyme, notée E, pour obtenir un produit, noté P. Plus précisément, le substrat se fixe sur l'enzyme pour former un complexe transitoire appelé « enzyme-substrat », noté C, qui se décompose ensuite pour donner le produit et l'enzyme. Dans la suite, et pour simplifier les écritures, on notera les différentes concentration (exprimées en $mol.L^{-1}$) par des lettres minuscules :

$$s = [S],\ e = [E] \text{ et } p = [P],$$

en remarquant que les concentrations sont des fonctions dépendant du temps.
On complète cette introduction en précisant les conditions initiales, à savoir :

$$s(0) = s_0,\ e(0) = e_0,\ c(0) = 0 \text{ et } p(0) = 0, \text{ avec } s_0 > 0 \text{ et } e_0 > 0$$

L'objectif de cette partie est d'étudier la vitesse $\dfrac{dp}{dt}$ à laquelle le produit se forme en admettant que :

$$\frac{dp}{dt}(t) = \frac{v_{max}s(t)}{K_M + s(t)} \text{ où } K_M = \frac{k_2 + k_{-1}}{k_1} > 0 \text{ et } v_{max} = k_2 e_0 > 0.$$

1.1. Étude du modèle

On considère l'équation suivante, appelée équation de Michaelis-Menten :

$$\frac{dp}{dt} = \frac{v_{max}s}{K_M + s}$$

Pour étudier cette équation, on définit sur $\mathbb{R}_+$ la fonction de Michaelis-Menten, notée f, par :

$$f(x) = \frac{v_{max}s}{K_M + s}$$

① *Montrons que la fonction f est croissante et déterminer ses limites aux bords du domaine de définition :*
f est dérivable sur $\mathbb{R}_+$ comme quotient de deux fonctions dérivables sur $\mathbb{R}_+$ dont le dénominateur ne s'annule pas (en effet K_M est supposé strictement positif). On obtient :

$$f'(s) = \frac{v_{max}(K_M + s) - v_{max}s}{(K_M + s)^2} = \frac{K_M v_{max}}{(K_M + s)^2} > 0,\ \forall s \in \mathbb{R}_+$$

Conclusion : $\boxed{f \text{ est croissante sur } \mathbb{R}_+}$

Quant aux limites aux bornes :

$$\boxed{f(0) = 0} \text{ et } f(s) \underset{s\to\infty}{\sim} \frac{v_{max}s}{s} = v_{max} \text{ donc } \boxed{\lim_{s\to+\infty} f(s) = v_{max}}$$

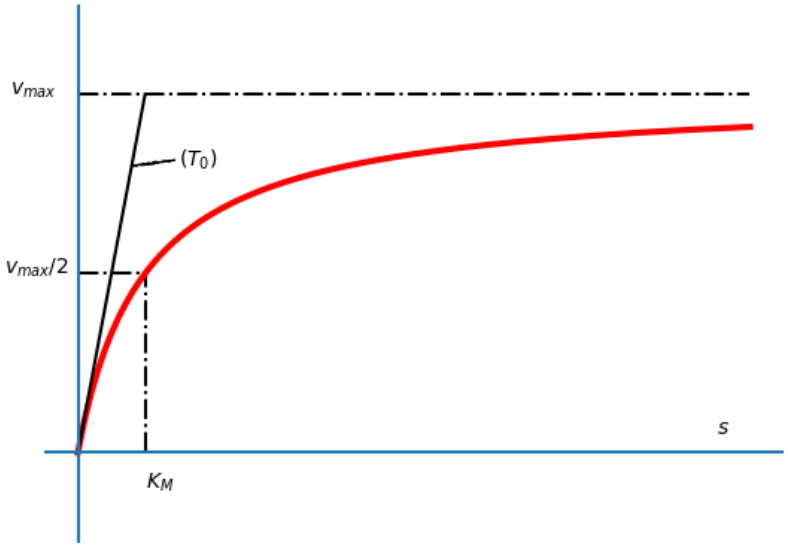
② *Pour quelle valeur de s a-t-on $f(x) = \dfrac{v_{max}}{2}$?* Il suffit de résoudre l'équation :

$$\forall s \in \mathbb{R}_+,\ f(s) = \frac{v_{max}}{2} \Leftrightarrow \frac{v_{max}s}{K_M + s} = \frac{v_{max}}{2} \Leftrightarrow \frac{s}{K_M + s} = \frac{1}{2} \Leftrightarrow s = \frac{K_M}{2} + \frac{s}{2} \Leftrightarrow s = K_M$$

Conclusion : $\boxed{f(K_M) = \dfrac{v_{max}}{2}}$

③ *Traçons la courbe représentative de f en y faisant figurer des informations pertinentes :*
On choisit de représenter les informations suivantes :

- L'asymptote horizontale d'équation $(y = v_{max})$.
- Le point $\left(K_M, \frac{v_{max}}{2}\right)$.
- La tangente en 0 d'équation $y = f'(0)s + f(0) = \frac{v_{max}}{K_M}s$ qui coupe l'asymptote au point d'abscisse K_M.



1.2. Identification expérimentale des paramètres

Dans la suite, la quantité $\frac{dp}{dt}(t)$ est notée $v(t)$. On note par ailleurs v_i la vitesse initiale et on obtient :

$$v_i = \frac{v_{max} s_0}{K_M + s_0}$$

① *Établissons une relation de la forme $v_i^{-1} = \alpha s_0^{-1} + \beta$ où les constantes α et β sont à déterminer :*
En inversant la relation, on obtient :

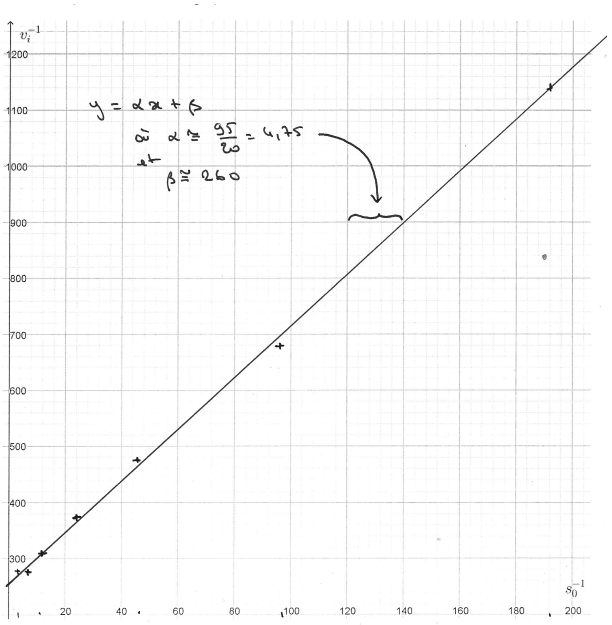
$$\frac{1}{v_i} = \frac{K_M + s_0}{v_{max} s_0} = \frac{K_M}{v_{max}} \cdot \frac{1}{s_0} + \frac{1}{v_{max}}$$

Conclusion : $\boxed{v_i^{-1} = \alpha s_0^{-1} + \beta, \text{ avec } \alpha = \frac{K_M}{v_{max}} \text{ et } \beta = \frac{1}{v_{max}}}$

② *Expliquons comment on peut déterminer graphiquement les paramètres K_M et v_{max} à partir des données expérimentales :*
On réalise plusieurs expériences permettant de mesurer v_i et s_0.
On place les points (s_0^{-1}, v_i^{-1}) dans le plan. Si le modèle est valide, aux erreurs de mesures près, ces points sont alignés sur une droite d'équation $y = \alpha x + \beta$. La détermination graphique de α et β permet de déterminer : $v_{max} = \frac{1}{\beta}$ et $K_M = \frac{\alpha}{\beta}$.

On se propose d'appliquer l'approche précédente sur des résultats expérimentaux de Michaelis et Menten concernant l'hydrolyse de saccharose sous l'action d'une enzyme, l'invertase. Le tableau suivant donne les vitesses initiales en fonction des concentrations initiales en saccharose pour 7 expérimentations, ainsi que leurs inverses arrondis à l'unité.

③ *Reportons sur le graphique de l'annexe* 1 *les couples* (s_0^{-1}, v_i^{-1}) *:*



④ *Proposons des valeurs approchées de* K_M *et* v_{max} *avec une précision en accord avec l'approche utilisée :*
L'impression du papier millimétré n'est pas d'excellente qualité... on estime donc

$$\alpha = 4.75 \text{ et } \beta = 260$$

Dès lors, $\boxed{v_{max} = \frac{1}{260} = 3,84 \cdot 10^{-3} mol.L^{-1}.min^{1}) \text{ et } K_M = \frac{4,75}{260} = 0,018 mol.L^{-1}}$

1.3. Étude informatique de données expérimentales

① a) *Écrivons une fonction* **`inv`** *qui prend en entrée une liste de nombres (supposés non nuls)* `L` *et qui renvoie la liste composée des inverses de ces nombres :*
Une écriture possible est :

```
def inv(L):
   return [1/k for k in L]
```

b) *Écrivons une version améliorée* **`inv_ex`** *de la fonction* **`inv`** *qui prend en entrée une liste de nombres* `L`*, puis : si un de ces nombres est nul, alors elle renvoie le booléen* **`False`***, sinon elle renvoie la liste composée des inverses de ces nombres.*

```
def inv_ex(L):
    Linv = []
    for k in L:
       if k == 0:
          return False
       else:
          Linv.append(1/k)
    return Linv
```

c) *Complétons les lignes de codes suivantes afin qu'elles effectuent le tracé demandé en question 3 de la partie 1.2 (les points seront représentés par des petits cercles et ne seront pas reliés entre eux) :*
On rappelle que les données dont il s'agit de calculer les inverses sont disponibles dans les listes `La` et `Lv`.
Il suffit donc d'écrire :

```
plt.plot(inv(La),inv(Lv),"o")
plt.show()
```

② Écriture de fonctions préliminaires.

a) *Écrivons une fonction* **`moyenne`** *qui prend en entrée une liste* **`X`** *(non vide) de nombres réels et qui renvoie la moyenne des éléments de la liste :*

Si les éléments de la liste X sont $x_1, x_2, \cdots, x_n$, alors $\overline{x} = \dfrac{1}{n}\displaystyle\sum_{k=1}^{n} x_k$.

Pour calculer la moyenne, il suffit donc de calculer la somme S des x_k puis diviser par `len(X)`.

```
def moyenne(X):
   s = 0
   for k in range(len(X)):
      s += X[k]
   return s/len(X)
```

ou

```
def moyenne(X):
   s = 0
   for x in X:
      s += x
   return s/len(X)
```

b) *Écrivons une fonction* **`variance`** *qui prend en entrée une liste* **`X`** *(non vide) de nombres réels et qui renvoie la variance des éléments de la liste.*

On rappelle la formule de la variance : $s_x^2 = \dfrac{1}{n}\displaystyle\sum_{k=1}^{n} x_k^2 - \overline{x}^2$.

Il suffit, pour calculer la variance, de calculer la moyenne des termes de la liste au carré auxquels on soustraira le carré de la moyenne calculé précedemment.
Ce qui donne :

```
def variance(X):
   s = 0
   for k in range(len(X)):
      s += X[k]**2
   return s/len(X)-moyenne(X)**2
```

ou

```
def variance(X):
   s = 0
   for x in X:
      s += x**2
   return s/len(X)-moyenne(X)**2
```

③ a) *Complétons le programme suivant afin qu'il renvoie la valeur de la covariance de* **`X`** *et* **`Y`** *si elle existe et le booléen* **`False`** *sinon :*
On connaît au moins deux formules pour calculer la covariance, mais au regard des deux dernières lignes du programme, sachant qu'il nous faut retourner `1/nx*S`, la seule formule possible est :

$$s_{x,y} = \frac{1}{n}\sum_{k=1}^{n}(x_k - \overline{x})(y_k - \overline{y})$$

```
def cov(X,Y):
   " " " Entree : X,Y (liste). " " "
   nx = len(X) ; ny = len(Y)
   if nx != ny or nx == 0:
      return(False)
   else:
      S = 0
      for k in range(nx):
         S = S+(X[k]-moyenne(X))*(Y[k]-moyenne(Y))
      y = 1/nx*S
      return(y)
```

b) *Parmi les quatre valeurs suivantes, lesquelles ne peuvent pas être renvoyées par la fonction* **`cov`** *? :*

i. `True`
ii. `[1,2]`
iii. `"False"`
iv. `-0.5`

Ce ne peut être 'i' car la fonction `cov` ne retourne jamais `True`.
Ce ne peut être pas 'ii' car `cov` retourne un réel et non une liste et pas davantage 'iii' car ce n'est pas une chaîne de caractère comme `"False"` qui sera retournée en cas d'impossibilité de calcul.
Conclusion : Seul la réponse 'iv' est plausible

c) On considère les fonctions `Coef` et `Trace` suivantes :

```
def Coef(X,Y):
    a = cov(X,Y)/variance(X)
    b = moyenne(Y)-cov(X,Y)/variance(X)*moyenne(X)
    return([a,b])

def Trace(X,Y):
    [a,b] = Coef(X,Y)
    xmin = min(X) ; xmax = max(X)
    plt.plot(X,Y,"*")
    plt.plot( ------- )
    plt.plot([moyenne(X)],[moyenne(Y)],"s")
    plt.grid()
    plt.show()
```

i. *Complétons la fonction* `Trace` *afin de tracer le segment d'extrémités* `(xmin,a*xmin+b)` *et* `(xmax,a*xmax+b)` *:*
Il suffit de créer la liste des abscisses `[xmin,xmax]` et la liste des ordonnées `[a*xmin+b,a*xmax+b]` et de tracer le segment qui les relie.
Le code attendu pour compléter la fonction `Trace(X,Y)` est donc :

```
plt.plot([xmin,xmax],[a*xmin+b,a*xmax+b])
```

ii. La droite qui passe par ces deux points a pour équation $y = ax + b$ avec, d'après la fonction `Coef()`, $a = \dfrac{s_{x,y}}{s_x^2}$ et $b = \overline{y} - a\overline{x}$.

iii. *Quel nom porte cette droite :* Il s'agit de la droite de régression ou de la droite des moindres carrés.

iv. On exécute la fonction `Trace` pour des listes `X` et `Y` quelconques de taille 5. *Pour chacun des trois tracés suivants, indiquons avec justification s'il peut être ou non le résultat de* `Trace` *?*
La droite de régression passe par le point de coordonnées $(\overline{x}, \overline{y})$ qui, sur les graphes proposés, est représenté par une carré. Les figures (b) et (c) sont donc exclues et seule reste plausible le tracé (a).

d) i. *Proposons un code qui calcule les valeurs de K_M et v_{max} en suivant la démarche de la partie 1.2 :*
On récupère les coefficients de la droite de régression grâce à la fonction `coef()` puis on utilise le fait que, d'après 1.2.2) :

$$v_{max} = \frac{1}{b} \text{ et } K_M = \frac{a}{b}$$

Un écriture possible est :

```
def Constantes(X,Y):
    [a,b] = Coef(inv(La),inv(Lb)
    v_max = 1/b
    K_M = v_max*a
    return K_M,v_max
```

ii. Pour ces données, le coefficient de corrélation linéaire vaut 0.9995. *Qu'en déduire ?*
Ce coefficient très proche de 1 pour un nombre de données réduit prouve la relation linéaire qui existe entre s_0^{-1} et v_i^{-1}.

On peut dès lors conclure que $v_i = \dfrac{v_{max} s_0}{K_M + s_0}$

✐ L'utilisation des fonctions Python à partir des données fournies permet d'obtenir : $K_M = 0.0183$ et $v_{max} = 3,95 \cdot 10^{-3}$, valeurs finalement assez proches de celles obtenues graphiquement en 1.2.3)

1.4. Analyse de l'équation de Michaelis-Menten par Schnelle et Mendoza

Les parties précédentes ont permis de déterminer expérimentalement les constantes K_M et v_{max}. Dans cette partie, on s'intéresse à la dépendance de s par rapport au temps, sous l'hypothèse de l'Approximation des Etats Quasi Stationnaires (AEQS) selon laquelle la variation de la concentration en complexe « enzyme-substrat » est nulle (car il est

consommé par la réaction juste après sa création), à l'exception d'une très courte phase initiale de durée $\delta > 0$. On considère que la concentration en S ne varie pas au cours de cette phase initiale.

Sous cette hypothèse, pour tout $t \in [\delta, +\infty[$, on a $\dfrac{ds}{dt}(t) = -\dfrac{dp}{dt}(t) = -\dfrac{v_{max}s(t)}{K_M + s(t)}$.
On s'intéresse désormais à l'équation différentielle :

$$\frac{ds}{dt} = -\frac{v_{max}s}{K_M + s}$$

avec pour condition initiale $s(\delta) = s_0$.

① Soit g la fonction définie sur $\mathbb{R}_+$ par $g(x) = xe^x$.

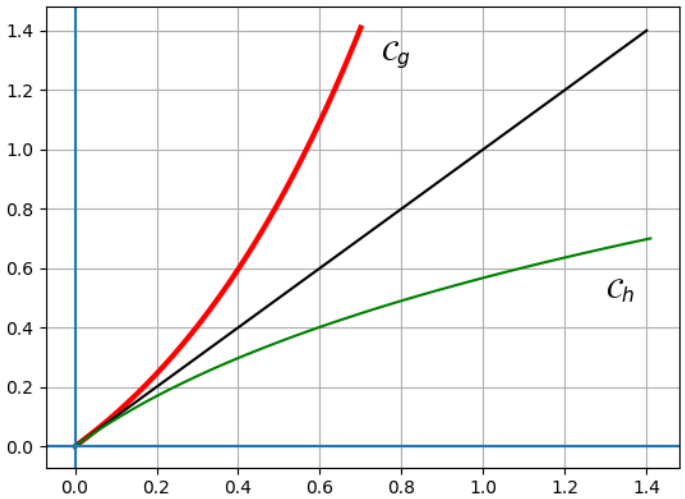
a) *Démontrons que g est strictement croissante :*
g est dérivable sur $\mathbb{R}$ comme produit de fonctions dérivables sur $\mathbb{R}$. Dès lors :

$$\forall x \in \mathbb{R}_+,\ g'(x) = e^x + xe^x = (x+1)e^x > 0$$

Conclusion : la fonction g est strictement croissante sur $\mathbb{R}_+$

b) *Déduisons-en que la fonction h, réciproque de g, est bien définie et donnons son domaine de définition :*
g est dérivable donc continue sur $\mathbb{R}_+$ et strictement croissante sur $\mathbb{R}_+$.
Par ailleurs $g(0) = 0$ et $\lim\limits_{x\to+\infty} g(x) = +\infty$, donc g est une bijection de $\mathbb{R}_+$ sur $\mathbb{R}_+$.
Elle admet donc une application réciproque, h, bijection de $\mathbb{R}_+$ sur $\mathbb{R}_+$.

c) *Traçons la courbe représentative de la fonction h en expliquant la méthode graphique utilisée :*
On utilise le fait que la courbe représentant h est symétrique de celle de g par rapport à la première bissectrice d'équation $(y = x)$.
Si on note que $g'(0) = 1$, on obtient que $y = x$ est tangente à $\mathcal{C}_f$ et à $\mathcal{C}_g$ en $(0, 0)$.
Par ailleurs, $g(x) - x = xe^x - x = x(e^x - 1) > 0$, $\forall x > 0$ donc $\mathcal{C}_f$ est au-dessus de sa tangente en 0.



② On définit $y(t) = g\left(\frac{s(t)}{K_M}\right)$. *Écrivons une équation différentielle du premier ordre satisfaite par la fonction y et donnons la condition initiale correspondante :*

On applique les règles de dérivation et on obtient :

$$\begin{aligned} y'(t) &= g'\left(\frac{s(t)}{K_M}\right)\frac{s'(t)}{K_M} \\ &= e^{\frac{s(t)}{K_M}}\left(\frac{s(t)}{K_M}+1\right)\frac{1}{K_M}\frac{-v_{max}s(t)}{K_M+s(t)} \\ &= -\frac{v_{max}}{K_M}e^{\frac{s(t)}{K_M}}\frac{s(t)+K_M}{K_M}\frac{s(t)}{K_M+s(t)} \\ &= -\frac{v_{max}}{K_M}e^{\frac{s(t)}{K_M}}\frac{s(t)}{K_M} = a\frac{s(t)}{K_M}e^{\frac{s(t)}{K_M}} = ay(t) \text{ où } a = -\frac{v_{max}}{K_M} \end{aligned}$$

La condition initiale est $y(\delta) = g\left(\frac{s(\delta)}{K_M}\right) = g\left(\frac{s_0}{K_M}\right)$.

Conclusion : $\boxed{\forall t \in [\delta, +\infty[,\ y'(t) = ay(t) \text{ avec } a = -\frac{v_{max}}{K_M} \text{ et } y(\delta) = g\left(\frac{s_0}{K_M}\right)}$

③ *Déduisons des questions précédentes une expression de $y(t)$ puis de $s(t)$ en fonction de t, K_M, v_{max} et s_0 (et faisant appel à la fonction h) :*

Nous savons que les solutions des équations différentielles de la forme $y'(t) = ay(t)$ sont de la forme

$$y(t) = \lambda e^{at} \text{ où } \lambda \in \mathbb{R}.$$

Dès lors :

$$\exists \lambda \in \mathbb{R} / y(t) = \lambda e^{-\frac{v_{max}}{K_M}t}$$

Par ailleurs,

$$y(\delta) = \lambda e^{-\frac{v_{max}}{K_M}\delta} = g\left(\frac{s_0}{K_M}\right)$$

D'où

$$\lambda = g\left(\frac{s_0}{K_M}\right)e^{\frac{v_{max}}{K_M}\delta} = \frac{s_0}{K_M}e^{\frac{s_0+v_{max}}{K_M}\delta}$$

Conclusion : $\boxed{\forall t \in [\delta, +\infty[,\ y(t) = \frac{s_0}{K_M}e^{\frac{s_0}{K_M} - \frac{v_{max}}{K_M}(t-\delta)}}$

Par ailleurs, $y(t) = g\left(\frac{s(t)}{K_M}\right)$ donc $\frac{s(t)}{K_M} = h(y(t))$ puisqu'on rappelle que h est la fonction réciproque de g...

Conclusion : $\boxed{\forall t \in [\delta, +\infty[,\ s(t) = K_M \cdot h\left(\frac{s_0}{K_M}e^{\frac{s_0}{K_M}}e^{-\frac{v_{max}}{K_M}(t-\delta)}\right)}$

④ *Proposons une méthode numérique permettant d'approcher les valeurs de la fonction h :*
Le problème est, effectivement, que nous connaissons $g : x \longmapsto xe^x$ mais que nous ne savons pas exprimer sa réciproque h... Il s'agit donc de trouver une méthode numérique permettant d'approcher la solution de l'équation :

$$g(x) = xe^x = y \Leftrightarrow xe^x - y = 0$$

On rappelle que g est une bijection strictement croissante de $\mathbb{R}_+$ sur $\mathbb{R}_+$ et que c'est donc le cas également de h.
On note que $g(0) = 0 < y$ donc $h(0) = 0 < h(y)$.
De même $g(y) = ye^y > y$ et donc $y > h(y)$.

En conséquence, la fonction $x \longmapsto g(x) - y$ s'annule sur $[0, y]$ et, cette fonction étant continue sur $\mathbb{R}_+$, on peut appliquer la méthode de dichotomie ou la méthode des Newton pour déterminer où elle s'annule.
✐ Dans la pratique, c'est assez compliqué car les valeurs de y sont grandes et le calcul de `np.exp(y)` pose problème. Il faut en fait trouver le moyen de réduire l'intervalle de départ, ce qui est largement possible au regard de la croissance rapide de g...