16 AVRIL 2021

APPROXIMATION NUMERIQUE DU MODELE DE CROISSANCE BACTERIENNE DE MONOD.

UTILISATION D'UN MODELE DE MONOD POUR L'ETUDE DE LA PROLIFERATION DE ACHROMOBACTER SP PROVENANT DES EAUX USEES D'UNE STATION D'EPURATION DANS UN MILIEU GLUCOSE.

TEOMAN SAPMAZ & HUSAMETTIN YESIL
ÉTUDIANT EN L2 MATHEMATIQUES
CY Cergy Paris Université

SOMMAIRE

RESOLUTION DE L'EQUATION DIFFERENTIELLE

SUBDIVISION DU PROBLEME DE CAUCHY

Afin d'étudier la prolifération de la bactérie Achromobacter sp provenant des eaux usées d'une station d'épuration dans un milieu glucosé, nous devons faire une approximation numérique du modèle de croissance bactérienne de Monod suivant :

$$X'(t) = r_{max}\frac{S(t)}{K + S(t)}X(t)$$

Supposons que l'on se trouve sur un intervalle de temps avec t ϵ [t_0 , t_f] tel que,

t_0 le temps initial

t_f le temps final

Alors on se contente à résoudre un problème de Cauchy tel que :

$$X'(t) = r_{max}\frac{S(t)}{K + S(t)}X(t)$$
$$X(0) = X_0$$

On a une suite de points $t_0 < t_1 < \ldots < t_N = t_f$

Calculons des valeurs approchées de la solution en chacun de ces points.

On appelle « pas » d'une subdivision le réel positif $\sigma = |a_{i+1}-a_i|$

On pose $h = t_{n+1} - t_n$, $\forall n \in \mathbb{N}$, h est une subdivision[1]

METHODE D'EULER EXPLICITE

Soit le problème de Cauchy tel que :

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(t_0) = y_0 \end{cases}$$

On cherche a résoudre l'équation sur une subdivision $h = t_{n+1} - t_n$

D'après l'équation du problème de Cauchy, on a :

$$\frac{dy}{dt} = f(t, y(t))$$

$$y = \int f(t, y(t))$$

La méthode d'Euler explicite[2] consiste à intégrer l'équation tel que :

$$y(t_{n+1}) - y(t_n) = \int_{t_n}^{t_{n+1}} f\big(s, y(s)\big)\, ds$$

METHODE DES RECTANGLES A GAUCHE

Pour résoudre l'équation obtenue par la méthode d'Euler explicite on doit appliquer la méthode des rectangles à gauche[3]

D'après le théorème des fonctions continues par morceaux on a :

Si f est une fonction continue par morceaux, pour toute suite (a_n) de subdivision [a, b] tel que (a_n)->0 lorsque n->+inf on a :

$$\lim_{n\to\infty} S_{a_n}(f) = \int_a^b f(s, y(s))\, ds$$

On choisit alors pour subdivision σ = $|a_{i+1}-a_i|$ tel que a est un point de cette subdivision tel que $a = i + \frac{k(b-a)}{n}$

Si on coupe l'intervalle [a, b] en n segments, on a : $h = \frac{b-a}{n}$

Ainsi on a la somme de Riemann noté S tel que :

$$S = \frac{b-a}{n} \sum_{n}^{n-1} f(a, y(a))$$

$$S = h * f(t, y(t))$$

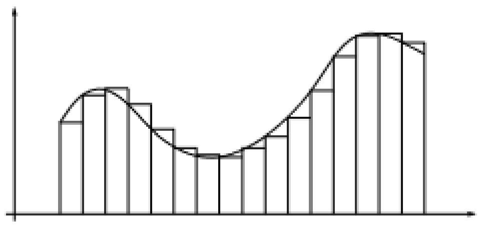


FIGURE 2.2 - Rectangles à gauche

Donc on obtient :

$$y(t_{n+1}) - y(t_n) = (t_{n+1} - t_n) * f(t, y(t_n))$$

$$y(t_{n+1}) = y(t_n) + (t_{n+1} - t_n) * f(t, y(t_n))$$

D'où sur la subdivision h, on a donc :

$$y(t_{n+1}) = y(t_n) + h * f(t, y(t_n))$$

METHODE DE RESOLUTION D'EQUATION GENERALE

Soit une équation de type : $a'(x) = C * b'(x)$ sur un intervalle I = [0, x]

Avec C, a(0) et b(0) des constantes tel que C différent de 0

Alors on peut exprimer a(x) en fonction de b(x) et réciproquement, on a :

$$\frac{da}{dx} = C * \frac{db}{dx}$$

$$\int_0^x \frac{da}{dx} = C * \int_0^x \frac{db}{dx}$$

$$a(x) - a(0) = C * (b(x) - b(0))$$

Donc[4] :

$$a(x) = a(0) + C * (b(x) - b(0))$$

Et :

$$b(x) = \frac{a(x) - a(0)}{C} + b(0)$$

APPLICATION

Données

Soit l'équation du modèle de croissance de bactérienne de Monod suivant :

$$X'(t) = r_{max} \frac{S(t)}{K + S(t)} X(t)$$

Et lorsque l'on suppose que la biomasse produite est proportionnelle au substrat consommé, on a l'équation :

$$X'(t) = -R * S'(t)$$

On prend pour intervalle le temps, donc I = [0, t]

Avec :

i. X'(t) : la dérivée en t de la masse de bactéries en $g.l^{-1}$ de solution à l'instant t
ii. X(t) : la masse de bactéries en $g.l^{-1}$ de solution à l'instant t
iii. S'(t) : la dérivée en t du substrat présent dans le milieu en $g.l^{-1}$ à l'instant t
iv. S(t) : la concentration du substrat présent dans le milieu en $g.l^{-1}$ à l'instant t
v. $r_{max} = 0{,}0437\ h^{-1}$: le taux maximal de croissance bactérienne
vi. $K = 0{,}0850\ mg.l^{-1}$: la concentration en substrat avec un taux de croissance de $\frac{1}{2} r_{max}$
vii. $R = 0{,}68$: le rendement de la conversion du substrat en biomasse

Résolution de l'équation X(t) en fonction de S(t)

On pose :

$$X'(t) = f(t, X(t))$$

Alors :

$$f(t, X(t)) = r_{max} \frac{S(t)}{K + S(t)} X(t)$$

On a :

$$X = \int f(t, X(t))$$

D'après la méthode d'Euler explicite on obtient :

$$X(t_{n+1}) - X(t_n) = \int_{t_n}^{t_{n+1}} f\big(s, X(s)\big)\, ds$$

D'après la méthode des rectangles à gauche on a :

$$X(t_{n+1}) - X(t_n) \approx (t_{n+1} - t_n) * f(t, X(t_n))$$

Sur la subdivision h = t_{n+1} -t_n on a alors :

$$X(t_{n+1}) \approx X(t_n) + h * f(t, y(t_n))$$

Donc

$$X(t_{n+1}) \approx X(t_n) + h * r_{max} \frac{S(t_n)}{K + S(t_n)} X(t_n)$$

Résolution de l'équation S(t)

On a :

$$X'(t) = -R * S'(t)$$

$$\frac{dX}{dt} = -R * \frac{dS}{dt}$$

$$\int_0^t \frac{dX}{dt} = -R * \int_0^t \frac{dS}{dt}$$

$$X(t) - X(0) = -R * (S(t) - S(0))$$

$$\frac{X(t) - X(0)}{-R} = S(t) - S(0)$$

Donc

$$S(t) = \frac{X(t) - X(0)}{-R} + S(0)$$

Avec S(0) = 1 $mg.l^{-1}$ et X(0) = 0,1 $mg.l^{-1}$ lorsque l'on suppose que la biomasse produite est proportionnelle au substrat consommé

Solution générale de X(t)

D'après tous les résultats obtenus précédemment, on peut donc en déduire de la solution de l'équation différentielle X'(t)

Donc

$$\mathrm{X}\big(t_{(n+1)}\big) \simeq \mathrm{X}(t_n) + h * (r_{max} * \frac{\frac{X(t_n) - X(0)}{-R} + S(0)}{K + \frac{X(t_n) - X(0)}{-R} + S(0)} * \mathrm{X}(t_n))$$

ETUDE DU PREMIER CAS

IMPLEMENTATION DES DONNEES EN PYTHON

import numpy as np import matplotlib.pyplot as plt from numpy import *	Nous importons les librairies python nécessaires au fonctionnement du programme (numpy[5] et matplotlib[6]).
def monod(h,phi,xtn,r,k): return (xtn+(((h*phi*(((xtn-x0)/-r)+s0))/((k+(xtn-x0)/-r)+s0))*xtn))	Nous définissons notre fonction Monod trouvé précédemment.
l=[] x0=0.1 s0=1 xtn=0.1 phi=0.0437 h=1 r=0.68 k=0.085 fxtn=0	Nous créons ici une liste l. Nous renseignons ensuite nos différentes données de notre problème.

for i in range (1,100):
   fxtn = monod(h, phi, xtn, r, k)
   l.append(fxtn)
   xtn = fxtn
	Nous créons une boucle qui va nous permettre d'appliquer notre fonction sur des valeurs allant de 1 à 100.
	Dans cette boucle la variable « fxtn » reçoit le résultat de X_{tn+1} en fonction de $X_{tn.}$ La liste l enregistre ensuite cette valeur, puis nous procédons à un changement de variable afin de calculer les valeurs suivantes.

fig=plt.figure() fig.patch.set_facecolor('xkcd:grey') plt.plot(l,'r') plt.plot(10,l[10],"-bo") print("A t=10 nous avons ",l[10],"mg/L") plt.axis([0,100,0.1,0.8]) plt.xlabel("Temps(en h)") plt.ylabel("Concentration de bactérie en mg/l") plt.title("Concentration de bactérie")	Une fois les valeurs calculées, nous allons les afficher sous forme de graphique pour ensuite les interpréter : -fig=plt.figure() sert à créer une figure. -plt.plot() sert à créer notre graphique si on y insère une liste ou une fonction ou bien on peut aussi marquer un point si on y insert des coordonnées. -plt.axis permet de configurer la fenêtre d'affichage. -fig.patch.set_facecolor('xkcd:grey') permet d'avoir un fond gris -print() nous sert à faire affichages textuels

APPROXIMATION DE X(10)

On suppose une concentration d'Achromobacter sp de 0, 1 mg.l^{-1} , dans un milieu glucosé à 1 mg.l^{-1}

Étudions l'évolution de cette concentration dans le temps et estimons la quantité de bactéries au bout de 10 heures

En implémentant un point a t=10 dans notre code PYTHON, on obtient le graphique suivant :

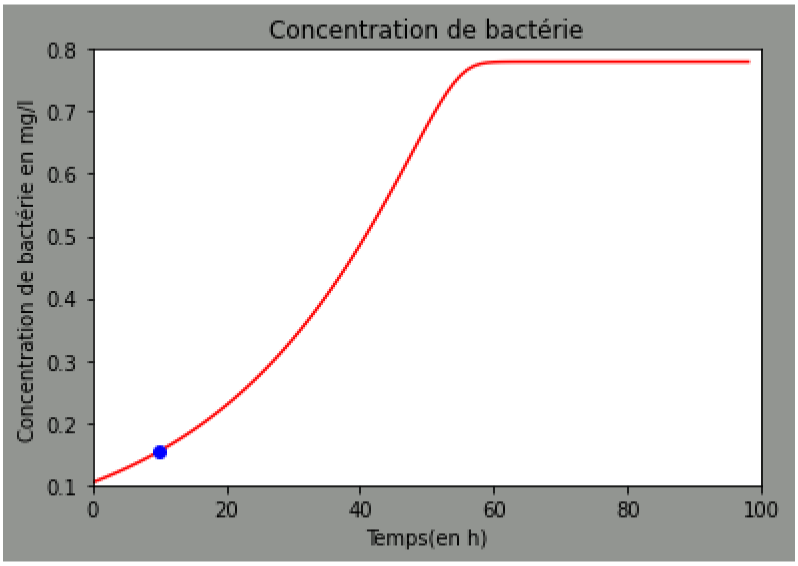


Figure 1 : La concentration de bactérie en fonction du temps

On peut alors observer notre explorateur de variables :

Ind	Type	Size	Value
6	float	1	0.1317837011143438
7	float	1	0.13707117648077094
8	float	1	0.14256709761021513
9	float	1	0.14827931733768068
10	float	1	0.15421594502511474
11	float	1	0.16038535008941437
12	float	1	0.16679616482283152
13	float	1	0.1734572863569106
14	float	1	0.18037787759547944

Figure 2 : L'explorateur de variables de notre code PYTHON

De plus dans notre code PYTHON en récupérant notre 10ème valeur de notre liste représentant notre fonction X(t), on en conclut donc que :

$$X(10) = 0{,}154 \ mg.l^{-1}$$

ETUDE DU SECOND CAS

RESOLUTION DE L'EQUATION X(T) EN FONCTION DU PROBLEME

On suppose que la concentration en substrat ne varie pas au cours du temps alors :

Soit $\forall\, t \ \in N\ , S(t) = 1\, mg.\, l^{-1}$

Alors S(0) = S(1) = … = S(t)

Soit X(0) = 0,1 mg.l^{-1} .

Étudions l'instant t la biomasse quand elle atteint 1 g.l^{-1}

Soit (P_n) une hypothèse, on pose :

$$(P_n)\ :\ X(t_{n+1}) \approx X(t_n) + h * (r_{max} \frac{S(0)}{K + S(0)} * X(t_n))$$

IMPLEMENTATION DES DONNEES EN PYTHON

import numpy as np import matplotlib.pyplot as plt from numpy import *	Nous importons les librairies python nécessaires au fonctionnement du programme.
def monod(h,phi,xtn,r,k): return (xtn+((h*phi*s0))/((k+s0))*xtn))	Nous définissons notre fonction mais cette fois-ci en remplaçant S(t) par S_0 car on suppose que notre concentration en substrat est constante.

def date(n):
   j=n//24
   h=n%24
   print("soit",j,"jours et",h,"heures.")
	Nous définissons encore une fonction mais cette fois-ci juste pour un meilleur affichage au lancement du programme.

l=[] x0=0.1 s0=1 xtn=0.1 phi=0.0437 h=1 r=0.68 k=0.085 fxtn=0 a=0 b=100000 c=0	Nous créons ici une liste l. Nous renseignons ensuite nos différentes données de notre problème ainsi que d'autres variables essentielles au fonctionnement de notre boucle ci-dessous.

for i in range (1,300):
   fxtn = monod(h, phi, xtn, r, k)
   l.append(fxtn)
   xtn = fxtn
	Même principe que notre premier programme pour le calcul de nos différentes valeurs mais avec notre fonction où S(t) est constant.

a=abs(l[i-1]-1000) if a<b: b=a c=i-1	Afin de trouver automatiquement le temps pour une concentration en bactérie de 1000mg/l on cherche la valeur absolue de la différence entre 1000mg/l et la concentration au temps t. On sauvegarde ensuite la valeur de t pour laquelle cette valeur est la plus petite.
print("On obtient une concentration de 1 g/L au bout de",c,"heures,") d=date(c) fig=plt.figure() fig.patch.set_facecolor('xkcd:grey') plt.plot(l,"r") plt.axis([0,300,0,5000]) plt.plot(c,l[c],"-bo") plt.xlabel("Temps(en h)") plt.ylabel("Concentration de Bactérie en mg/l") plt.title("Biomasse de bactérie avec substrat constant")	Enfin, nous procédons à l'affichage de notre graphique ainsi qu'un point rouge représentant la valeur de 1000g/l atteinte. De plus un affichage détaillé du temps nécessaire est aussi présent.

RECHERCHE DE LA SOLUTION X(T) = 1 G.L^{-1}

En implémentant dans notre code PYTHON on a le traçage de la courbe X(t) avec S(t) qui ne varie pas en fonction du temps on obtient :

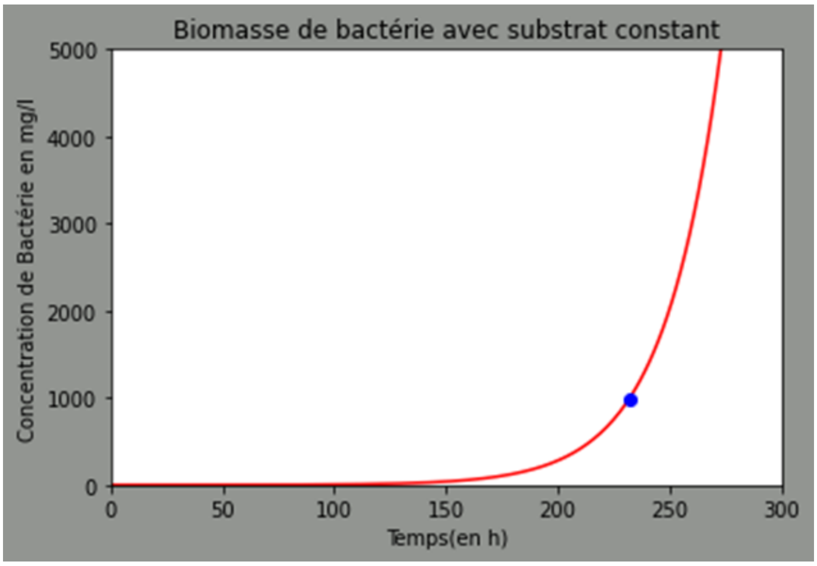


Figure 3 : La concentration de bactéries en fonction du temps lorsque la concentration en substrat ne varie pas

D'après notre code PYTHON, on obtient :

$$X(t) = 1\ g.l^{-1}$$

$$\Leftrightarrow\ t = 232\ heures$$

ETUDE DU DERNIER CAS

RESOLUTION DES EQUATIONS GENERALE

On suppose que l'expérience se déroule dans un réservoir de 100 m^3 brassé en permanence

Et qu'en permanence de l'eau glucosée arrive en remplacement de l'eau polluée du milieu à raison de 1 $l.s^{-1}$

On sait que :

$$\frac{\delta X}{\delta t} = -R\frac{\delta S}{\delta t}$$

D'après la méthode de la résolution générale, on a :

$$X = -R * \left(S(t) - S(0)\right) + X(0)$$

De plus, on a :

$$S(t) = \frac{X(t) - X(0)}{-R} + S(0)$$

CALCUL DE LA SORTIE D'EAU POLLUEE ET ENTREE D'EAU GLUCOSEE

Supposition

On sait que dans le réservoir de 100 m^3 il y a déjà une solution de glucose et de bactéries tel que :

$$X(0) = 0{,}1\, mg.l^{-1}$$

$$S(0) = 1\, mg.l^{-1}$$

On suppose qu'à chaque seconde, 1l d'eau glucosée avec une concentration de 1 $mg.l^{-1}$ se rajoute et 1l d'eau polluée sort

De plus on sait que 100 m^3 = 100 000 l et 1h = 3600s

Étudions la biomasse en sortie de réservoir lorsqu'elle tend vers un équilibre

Hypothèses

Entrée d'eau glucosée

Soit par la méthode de dilution, on sait que $C_1 * V_1 = C_2 * V_2$

Dans notre cas :

i. C_1 = 1 mg. l^{-1}
ii. V_1 = 1 l
iii. V_2 = 100 000 l

Déterminons C_2 la concentration d'eau entrante diluée dans le réservoir

Donc $C_2 = \frac{1}{100\ 000}\ mg.l^{-1}$

Or nous voulons cette dilution chaque heure, on multiplie donc le résultat par seconde trouvée par 3600

D'où $C_2 = \frac{3600}{100\ 000} = 0{,}036\ mg.l^{-1}$

Donc la concentration d'eau glucosée qui rajoute chaque heure est de 0,036 mg.l^{-1}

Sortie d'eau polluée

Etudions la concentration en bactérie et en glucose lorsque l'eau polluée sort

On pose (A_n) notre hypothèse pour la concentration en bactérie et (B_n) notre hypothèse pour la concentration en glucose

On a :

$$(A_n) : X_f(t) = X(t) - \frac{3600 * X(t)}{100\ 000}$$

Concentration en bactérie présent dans le réservoir

Concentration de bactérie pour 1l qui sort par heure

Concentration de glucose qui entre chaque heure

$$(B_n) : S_f(t) = S(t) - \frac{3600 * S(t)}{100\ 000} + 0{,}036$$

Concentration en glucose présent dans le réservoir

Concentration de glucose pour 1l qui sort par heure

IMPLEMENTATION DES DONNEES EN PYTHON

import numpy as np import matplotlib.pyplot as plt from numpy import * from pylab import *	Nous importons les librairies python nécessaires au fonctionnement du programme.
def bacterie(stn,s0,x0,r): return (-r)*(stn-s0)+x0	Nous définissons la fonction de la concentration en bactérie. Voir paragraphe précédent.
def glucose(xtn,s0,x0,r): return ((xtn-x0)/(-r))+s0	Nous définissons la fonction de la concentration en substrat (glucose). Voir paragraphe précédent.
def monod(h,phi,xtn,r,k): return (xtn+(((h*phi*(((xtn-x0)/-r)+s0))/((k+(xtn-x0)/-r)+s0))*xtn))	Nous définissons ici la même fonction représentant notre équation de Monod que le premier cas.
l1=[] l2=[]	Nous créons 2 listes qui vont nous permettre de prendre les différentes valeurs de X(t) et de S(t).
x0=0.1 s0=1	Puis nous renseignons nos valeurs déjà connues de notre situation.

`xtn=0.1`
`stn=1`
`r=0.68`
`phi=0.0437`
`h=1`
`k=0.085`
`c=100`	

`x=arange(1,10000,100 )`	On assigne à la variable x un intervalle allant de 1 à 100 avec à l'intérieur 10000 valeurs de x.
`for i in x:`	Nous pouvons maintenant débuter nos calculs grâce à une boucle.
`stn = glucose(xtn, s0, x0, r)`	On commence par calculer la première valeur de S(t) car nous possédons déjà la valeur de X(t) pour le même t (en effet, S(t) s'exprime en fonction de X(t) et inversement).
`xtn = bacterie(stn, s0, x0, r)`	On calcule ensuite la valeur de X(t) correspondant.
`l1.append(xtn-(xtn/100000)*3600)`	Notre liste l1 prend sauvegarde ensuite la valeur de X(t) à laquelle nous avons enlevé la perte de bactéries dans les eaux usées.
`l2.append(stn-((stn/100000)*3600)+0.036)`	Notre liste l2 fait de même mais pour S(t).

`a=stn-xtn`
`if abs(a)<c:`
`b=i/100`
`c=a`
	Afin de trouver le point d'intersection entre nos 2 courbes S(t) et X(t) nous cherchons la plus petite valeur absolue de la différence entre les valeurs de X(t) et S(t) pour un même t. Nous enregistrons ensuite le temps ainsi que la valeur pour laquelle les courbes se croisent.

`xtn=monod(h, phi, xtn, r, k)`	On applique enfin notre fonction de notre équation de Monod afin de trouver la prochaine concentration en bactérie X(t) pour relancer notre boucle avec la fonction de S(t). On recommence cet enchainement jusqu'à n=100 pour 10 000 valeurs.

`fig=plt.figure()`
`fig.patch.set_facecolor('xkcd:grey')`
`print ("le point d'équilibre se trouve pour t=",b)`
`plt.plot(l1,"b")`
`plt.plot(l2,"r")`
`plt.axis([0,100,0,1.1])`
`plt.xlabel("Temps(en h)")`
`plt.ylabel("Concentration en mg/l")`
`plt.title("Le point d'équilibre")`
	Nous pouvons afficher maintenant notre graphique avec nos 2 courbes ainsi que leur point d'intersection. Cet affichage est accompagné de phrases nous indiquant pour quelle valeur et pour quel temps le point d'intersection est obtenu.

LA BIOMASSE EN SORTIE DE RESERVOIR

Étudions la sortie de la biomasse lorsqu'elle tend vers un équilibre

On suppose que la biomasse tend vers un équilibre donc les limites de X_f et S_f existe

Cherchons alors $\lim_{t \to +\infty} X_f(t)$ et $\lim_{t \to +\infty} S_f(t)$

D'après le graphique obtenu suite à notre code PYTHON on a :

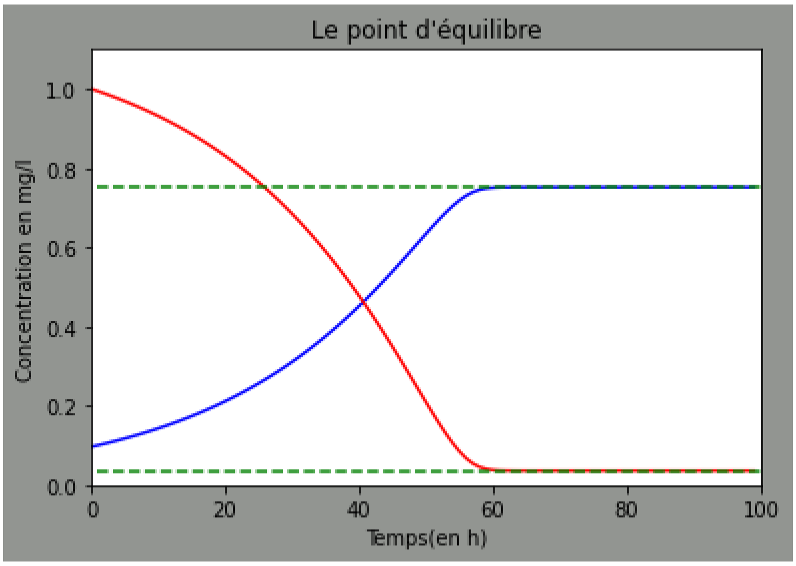


Figure 4 : Le point d'équilibre (S_f en rouge et X_f en bleu)

On remarque qu'il existe une intersection entre les 2 courbes que l'on va poser comme point I

Et on remarque aussi qu'après les 60 h, les deux concentrations tendent vers un équilibre et stagne

Donc concluons des valeurs du point I et des $\lim_{t \to +\infty} X_f(t)$ et $\lim_{t \to +\infty} S_f(t)$ à l'aide de l'explorateur de variables de PYTHON

D'après notre code PYTHON le point I est à t = 40,1

39	float	1	0.4353391331787092
40	float	1	0.45151623410073966
41	float	1	0.46816032955748926

Figure 5 : La valeur du point d'équilibre

Donc

$$I = (40{,}1\,;40{,}515)$$

D'après notre code PYTHON les limites des fonctions peuvent être connues à partir de t = 60.

58	float	1	0.745275253375532
59	float	1	0.748745361218437
60	float	1	0.7505091234391582
61	float	1	0.7513190798094004
62	float	1	0.7516693950874219
63	float	1	0.7518164718602257
64	float	1	0.7518774031858627
65	float	1	0.7519025029031111
66	float	1	0.7519128178675458

Figure 6 : Les images de X_f

58	float	1	0.04577168621245287
59	float	1	0.04066858644347493
60	float	1	0.03807481847182628
61	float	1	0.036883706162646496
62	float	1	0.036368536636144234
63	float	1	0.036152247264374056
64	float	1	0.03606264237373136
65	float	1	0.0360257310248367
66	float	1	0.036010561959491476

Figure 7 : Les images de S_f

Donc on en déduit que

$$\lim_{t \to +\infty} X_f(t) = 0{,}752$$

$$\lim_{t \to +\infty} S_f(t) = 0{,}036$$

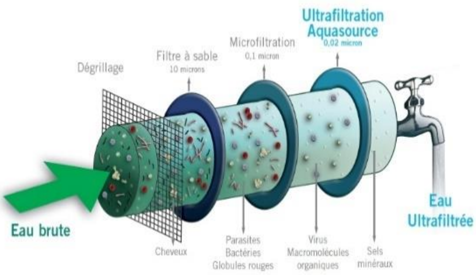
Donc la biomasse au bout d'un certain temps est composée de bactérie à 0,752 $mg.l^{-1}$ et de glucose à 0,036 $mg.l^{-1}$

CONCLUSION

SOLUTION

A travers l'étude de l'Achromobacter sous 3 différents cas nous pouvons conclure que la bactérie et le glucose sont finement liés. En effet, d'après le premier cas on remarque que la concentration en bactéries évolue avec une première phase exponentielle puis reste constant au bout d'un certain temps. On peut donc supposer qu'il existe un élément limitant du développement de la bactérie. Dans le deuxième cas nous avons supposé que la concentration en glucose soit constante au cours du temps. Nous avons donc remarqué que la bactérie se développait à une vitesse exponentielle sans s'arrêter, on peut donc supposer ici que la bactérie se nourrit de glucose pour se développer et donc que du glucose est consommé. Pour affirmer nos suppositions, nous avons procédé à un 3ème cas. Pour ce faire, on se situe dans un bassin de 100m3. On a donc réussi à créer un environnement se rapprochant plus de la réalité que les cas précédents avec entrée et sortie de différentes solutions. Après le tracé de notre graphique nous avons bien remarqué que plus la concentration en bactéries augmentait et plus la concentration en glucose diminuait jusqu'à atteindre un équilibre. Nos suppositions sont donc confirmées par ce 3ème cas, on cherche alors maintenant une solution afin de lutter contre le glucose, car le glucose est à l'origine de la prolifération bactérienne.

Afin de lutter contre la présence de ce glucose, nous proposons une méthode de filtration membranaire des eaux usées. En effet, la molécule de glucose $C_6H_{12}O_6$ est plus encombrante que la molécule d'eau H_2O. Nous pouvons donc filtrer l'eau usée à travers de membranes de plus en plus fines afin de laisser passer que les molécules d'eau H_2O.



INCERTITUDE

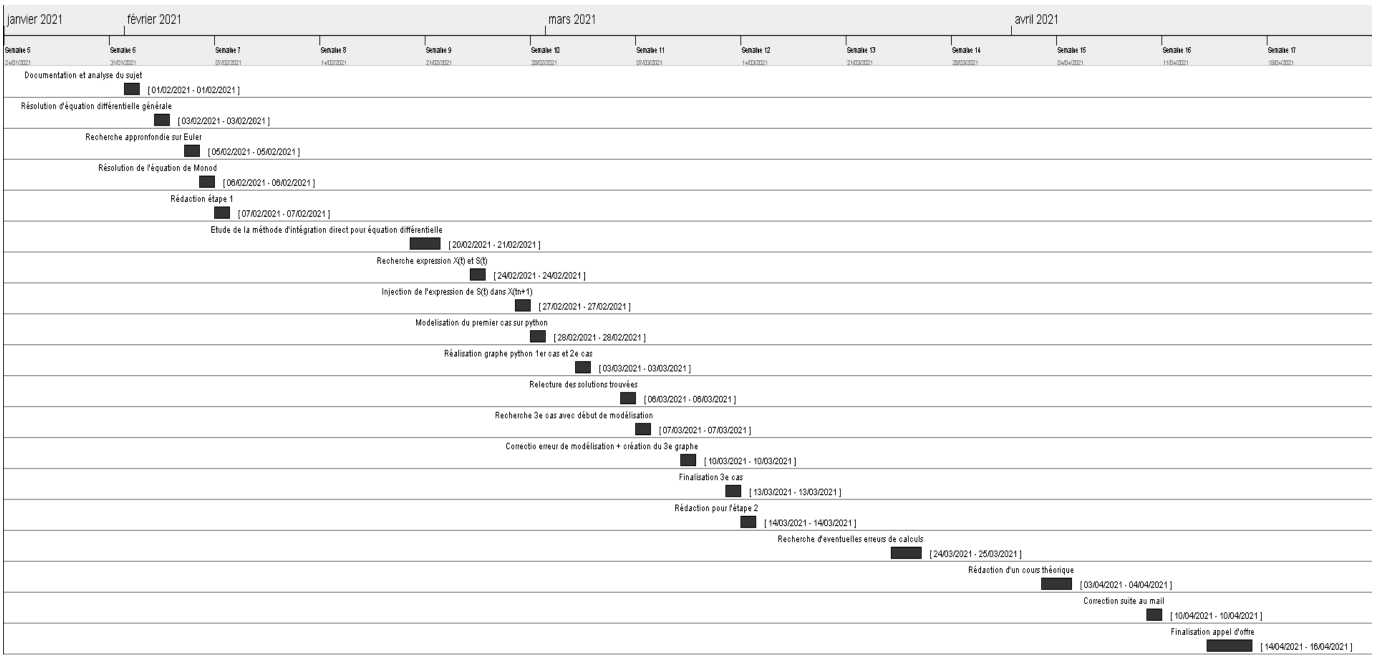
La méthode utilisée pour l'étude de la croissance de l'Achromobacter dans une eau glucosée est une approximation. Donc il y a un taux d'incertitude, les données ne tiennent donc pas compte de la réalité. Pour calculer notre taux d'erreur nous avons fait appel à la bibliothèque scikit_learn afin d'utiliser la méthode du mean_absolute_error et mean_square_error[7]. Alors pour l'étude du dernier cas nous avons tracer la courbe de l'équation de Monod initial. Donc le résultat retourné par notre code PYTHON a était : pour la moyenne d'erreur absolue, on a 0,026 et pour la moyenne d'erreur quadratique, on a 0,001. Donc notre approximation est plutôt correcte ainsi notre taux d'incertitude est plutôt faible. Donc notre méthode d'approximation est proche de la réalité donc les solutions que l'on propose peuvent être efficace.

ANNEXES

BIBLIOGRAPHIE

1. http://philippe.gravejat.perso.math.cnrs.fr/Enseignement/L2/Integr1.pdf page 9
2. https:+//bruneau.u-cergy.fr/enseignement/L3ananum/cours-analyse_numerique_2016-2017.pdf page 50 - 51
3. https://bruneau.u-cergy.fr/enseignement/L3ananum/cours-analyse_numerique_2016-2017.pdf
4. http://math.univ-lyon1.fr/capes/IMG/pdf/new.primitive.pdf page 3
5. https://numpy.org/doc/stable/user/whatisnumpy.html
6. https://matplotlib.org/stable/tutorials/introductory/pyplot.html#sphx-glr-tutorials-introductory-pyplot-py
7. https://ichi.pro/fr/quelles-sont-les-differentes-manieres-d-evaluer-un-modele-de-regression-lineaire-209018250907954

ORGANISATION



Toutes recherches, travaux et programmation ont été effectué par notre groupe.
Pour la remise de l'appel d'offre environs 40 heures ont été consacré au total.