

Modélisation du Stockage du Dioxyde de Carbone

Introduction

Ce projet a pour objectif de modéliser les échanges de carbone entre les différents compartiments de l'écosystème à l'aide d'un système d'équations différentielles ordinaires (EDO). Le modèle étudié ne prend pas en compte l'ensemble des paramètres physiques, cela mènerait à un système plus complexe. Pour aller plus loin dans la modélisation et obtenir une description plus réaliste des échanges, on peut utiliser des équations aux dérivées partielles. Le modèle présenté ici, basé sur des EDO qui ne tiennent pas compte de la dimension spatiale, les EDP permettent de modéliser la répartition du carbone dans l'espace par exemple, sa diffusion dans le sol ou son transport dans l'atmosphère. Ces équations sont essentielles pour représenter la variabilité des sols, du climat ou de la végétation à grande échelle, ainsi que pour étudier des phénomènes tels que la décomposition de la matière organique en profondeur. Des modèles plus complexes sont utilisés dans les simulateurs climatiques et les modèles globaux de végétation, et nécessitent des méthodes numériques avancées pour être résolus.

Formulation Mathématique

Dans ce projet, nous étudions l'évolution de trois variables en fonction du temps :

- $C_A(t)$: Quantité de carbone dans l'atmosphère
- $C_T(t)$: Quantité de carbone dans les arbres
- $C_S(t)$: Quantité de carbone dans les sols

Le système d'équations gouvernant les échanges de carbone dans ce projet peut être modélisé par:

$$(1) \quad \frac{dC_A}{dt} = -S(C_T) + \beta C_T + \delta C_S \quad (1)$$

$$(2) \quad \frac{dC_T}{dt} = S(C_T) - \beta C_T - \delta C_T - \gamma C_T \quad (2)$$

$$(3) \quad \frac{dC_S}{dt} = \gamma C_T + \delta C_T - \delta C_S \quad (3)$$

avec:

- $S(C_T) = \alpha C_T \left(1 - \frac{C_T}{K}\right)$: taux de séquestration du carbone dans les arbres,
- βC_T : respiration des arbres vers l'atmosphère,
- δC_T : transfert de carbone des arbres vers les sols,
- δC_S : respiration des sols vers l'atmosphère,
- γC_T : litière végétale (feuilles mortes, débris organiques transférée des arbres vers les sols).

Méthodes et Implémentation

On pose:

$$C(t) = \begin{pmatrix} C_A(t) \\ C_T(t) \\ C_S(t) \end{pmatrix}$$

et

$$f(t, C(t)) = \begin{pmatrix} -\alpha C_T(t) \left(1 - \frac{C_T(t)}{K}\right) + \beta C_T(t) + \delta C_S(t) \\ \alpha C_T(t) \left(1 - \frac{C_T(t)}{K}\right) - \beta C_T(t) - \delta C_T(t) - \gamma C_T(t) \\ \gamma C_T(t) - \delta C_S(t) + \delta C_T(t) \end{pmatrix}$$

Ainsi le système revient à:

$$\frac{dC(t)}{dt} = f(t, C(t))$$

Si on discrétise uniformément l'intervalle de temps en intervalles $[t_n, t_{n+1}]$, avec $n = \{0, 1, \dots, N-1\}$, tel que $t_{n+1} = t_n + h$ et $h = \frac{1}{N}$. On peut appliquer une méthode d'Euler:

$$C_n = C(t_n) \tag{4}$$

$$C_{A,n} = C_A(t_n) \tag{5}$$

$$C_{T,n} = C_T(t_n) \tag{6}$$

$$C_{S,n} = C_S(t_n) \tag{7}$$

$$\int_{t_n}^{t_{n+1}} \frac{dC(t)}{dt} dt = \int_{t_n}^{t_{n+1}} f(t, C(t)) dt$$

Euler Implicite

$$C_{n+1} - C_n \approx (t_{n+1} - t_n)f(t_{n+1}, C_{n+1})$$

$$C_{n+1} = C_n + h \cdot f(t_{n+1}, C_{n+1})$$

Cette équation étant non-linéaire, il est mieux de se ramener à un problème de point fixe :

$$F(C_{n+1}) = C_n + h \cdot f(t_n, C_{n+1})$$

On considère alors $\forall n \geq 1$ la suite $(C_{n,k})_{k \in \mathbb{N}}$ telle que:

$$\begin{cases} C_{n,k+1} &= F(C_{n,k}) \\ C_{n,0} &= C_{n-1} \end{cases} \quad (8)$$

Pour h assez petit, on peut montrer que F est contractante, de plus le premier terme de la suite étant assez proche de la limite, la méthode converge assez bien pour des valeurs de h raisonnables.

Algorithme:

```
Fonction point_fixe(X0, F)
  Initialiser Xk ← X0
  Initialiser Xk_1 ← X0

  Pour i allant de 0 à max_iter - 1 faire
    Xk_1 ← F(Xk, X0)
    Si la norme de (Xk - Xk_1) est inférieure à eps alors
      Sortir de la boucle
    Fin Si
    Xk ← Xk_1
  Fin Pour

  Retourner Xk_1
Fin Fonction

Fonction eulerImplicite(C0)
  Initialiser t ← t0
  Initialiser C #tableau contenant les grandeurs
  Créer une liste T contenant t0 #tableau des temps
  Mettre la première colonne de C égale à C0
  Initialiser k ← 1
```

```

Tant que t < Tf faire
    t ← t + h
    Cn_plus_1 ← point_fixe(C[:, k-1], F)
    Ajouter Cn_plus_1 comme nouvelle colonne à C
    Ajouter t à la liste T
    k ← k + 1
Fin Tant que

Retourner C et T
Fin Fonction

```

Euler Explicite

$$C_{n+1} - C_n \approx (t_{n+1} - t_n)f(t_n, C_n)$$

$$C_{n+1} = C_n + h.f(t_n, C_n)$$

Algorithme:

```

Fonction eulerExplicite(C0)
    Initialiser t ← t0
    Créer un tableau C #tableau contenant les grandeurs
    Créer une liste T contenant t0
    Mettre la première colonne de C égale à C0
    Initialiser k ← 1

    Tant que t < Tf faire
        t ← t + h
        Cn_plus_1 ← C[:, k-1] + h × f(C[:, k-1])
        Ajouter Cn_plus_1 comme nouvelle colonne à C
        Ajouter t à la liste T
        k ← k + 1
    Fin Tant que

    Retourner C et T
Fin Fonction

```

Méthode de Trapèze La méthode des trapèzes permet de mieux approximer l'intégrale, et donc donner un modèle plus précis en plus de permettre des valeurs de h plus grandes, donc moins de calculs.

$$C_{n+1} = C_n + \frac{h}{2}[f(t_{n+1}, C_{n+1}) + f(t_n, C_n)]$$

Tout comme la méthode d'Euler Implicite, nous avons une équation non-linéaire, cette fois si nous allons utiliser une méthode de Newton qui va nous garantir une convergence quadratique et qui est donc plus performant que la méthode de point fixe, on pose alors:

$$F(C_{n+1}) = C_n + \frac{h}{2}[f(t_{n+1}, C_{n+1}) + f(t_n, C_n)]$$

et

$$f_{newton}(C_{n+1}) = F(C_{n+1}) - C_{n+1}$$

Cela revient à résoudre $f_{newton}(C_{n+1}) = 0$

$$\frac{df_{newton}(C_{n+1})}{d(C_{n+1})} = \frac{d}{dC_{n+1}}F(C_{n+1}) - \frac{d}{dC_{n+1}}C_{n+1} \quad (9)$$

$$= \frac{d}{dC_{n+1}}(C_n + \frac{h}{2}(f(t_{n+1}, C_{n+1}) + f(t_n, C_n)) - I_3 \quad (10)$$

$$= \frac{h}{2} \frac{d}{dC_{n+1}}f(t_{n+1}, C_{n+1}) - I_3 \quad (11)$$

$$(12)$$

ainsi:

$$\frac{df_{newton}(C_{n+1})}{d(C_{n+1})} = \frac{h}{2} \begin{pmatrix} 0 & -\alpha + \frac{2C_{T,n+1}}{K} + \beta & \delta \\ 0 & \alpha - \frac{2C_{T,n+1}}{K} - \beta - \delta - \gamma & 0 \\ 0 & \gamma + \delta & -\delta \end{pmatrix} - I_3$$

On à alors que chaque C_n est solution du problème $f_{newton}(X) = 0$ telle que:

$$\begin{cases} X_{n,0} &= C_{n-1} \\ X_{n,k+1} &= X_{n,k} - (\frac{df_{newton}(X_{n,k-1})}{X_{n,k-1}})^{-1} \cdot f_{newton}(X_{n,k-1}) \end{cases} \quad (13)$$

Algorithme:

Fonction newton(X0, f, df)

Initialiser Xk \leftarrow X0

Pour i allant de 0 à max_iter - 1 faire

Si la norme de f(Xk, X0) est inférieure à eps alors

Sortir de la boucle

```

        Fin Si

         $X_k \leftarrow X_k - \text{inverse}(\text{df}(X_k)) \times f(X_k, X_0)$ 
    Fin Pour

    Retourner  $X_k$ 
Fin Fonction

Fonction trapeze_newton(C0)
    Initialiser  $t \leftarrow t_0$ 
    Créer un tableau C #tableau contenant les grandeurs
    Créer une liste T contenant  $t_0$ 
    Mettre la première colonne de C égale à C0
    Initialiser  $k \leftarrow 1$ 

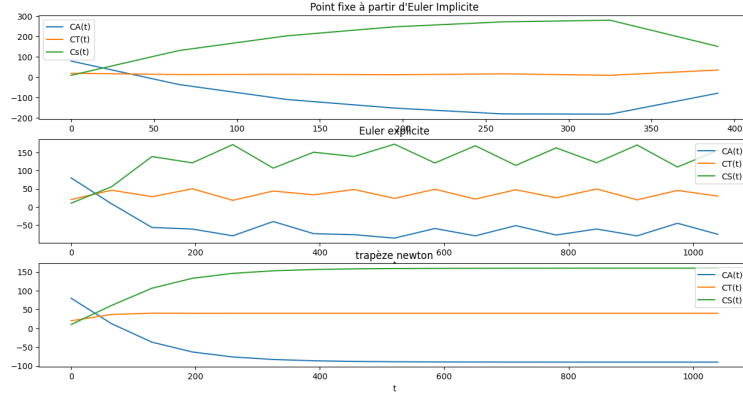
    Tant que  $t < T_f$  faire
         $t \leftarrow t + h$ 
         $C_{n\_plus\_1} \leftarrow \text{newton}(C[:, k-1], f\_newton, df\_newton)$ 
        Ajouter  $C_{n\_plus\_1}$  comme nouvelle colonne à C
        Ajouter  $t$  à la liste T
         $k \leftarrow k + 1$ 
    Fin Tant que

    Retourner C et T
Fin Fonction

```

Résultats

On teste nos méthode avec comme paramètre: $\alpha = 0.1, \beta = 0.02, \gamma = 0.03, \delta = 0.01, K = 100$ et comme pas $h = 65$.



On voit la stabilité par rapport au pas de la méthode Trapèze avec Newton, comparée aux deux autres.

Impact des Paramètres sur la Séquestration du Carbone

Le paramètre α Le paramètre α influence le taux de séquestration du carbone par les arbres. Il apparaît dans la fonction non linéaire :

$$S(C_T) = \alpha C_T \left(1 - \frac{C_T}{K} \right)$$

Cette fonction décrit la capture du CO_2 par les arbres. Lorsque α augmente (pour un C_T et un K fixés), la valeur de $S(C_T)$ augmente également. Cela signifie que α influence directement la rapidité d'absorption du carbone, ce qui conduit à un stockage plus rapide dans la biomasse végétale.

Le paramètre K Le paramètre K représente la quantité maximale de carbone que les arbres peuvent stocker. En reprenant l'équation :

$$S(C_T) = \alpha C_T \left(1 - \frac{C_T}{K} \right)$$

on observe que faire varier K modifie le taux de séquestration. Une valeur élevée de K permet aux arbres d'absorber du carbone pendant une période plus longue avant d'atteindre leur capacité maximale.

Le paramètre β Le paramètre β , qui intervient dans le terme βC_T , représente la fraction de carbone que les arbres restituent à l'atmosphère par respiration. Plus β est élevé, plus le retour de CO_2 dans l'air est important, ce qui réduit l'efficacité globale de la séquestration du carbone.

Le paramètre γ Le paramètre γ contrôle le flux de matière organique morte (litière) des arbres vers les sols, modélisé par le terme γC_T . Il ne prend pas en compte la respiration des arbres vers le sol, qui est décrite par le terme δC_T . Une valeur élevée de γ favorise l'enrichissement des sols en matière organique, ce qui améliore leur fertilité et leur capacité à stocker du carbone sur le long terme.

Le paramètre δ Le paramètre δ intervient à deux niveaux : dans la respiration des arbres vers les sols (δC_T), et dans la respiration des sols vers l'atmosphère (δC_S). Il régule donc les pertes de carbone par respiration. Une valeur élevée de δ accélère le cycle du carbone, augmentant la quantité de CO_2 retournant à l'atmosphère. À l'inverse, un δ faible réduit ces pertes, ce qui favorise le stockage du carbone, notamment dans les sols. Cependant, une respiration trop faible peut ralentir le recyclage des nutriments essentiels à la croissance des plantes.

Analyse de l'impact des paramètres sur la modélisation

Impact de K La capacité de stockage maximale des arbres est représentée par le paramètre K . Si la quantité de carbone initiale dans les arbres C_{T0} est supérieure à K , alors la fonction de séquestration du carbone, $S(C_T)$ est négative. Donc du carbone présent dans les arbres est rejeté dans le sol et dans l'atmosphère pour que C_T passe sous K .

On remarque qu'en effet au début le carbone dans l'air augmente et celui dans le sol a aussi une augmentation plus forte.

Il y a une autre remarque : c'est le fait que la modélisation pour $h = 10$ et Euler Implicite ne marche pas. On peut imaginer que si $C_{T0} > K$ alors C_T évolue rapidement pour passer sous K , tandis que C_S et C_A sont plus lents. Il faut donc un pas faible pour pouvoir capter correctement ces variations.

Absorption de carbone par les arbres uniquement Dans ce cas, $\gamma = \beta = \delta = 0$ et $\alpha = 0.3$, les arbres absorbent du carbone sans jamais le relâcher, ni dans l'atmosphère, ni dans le sol. Dans ce scénario, le carbone absorbé par les arbres augmente jusqu'à atteindre la capacité maximale K , tandis que le carbone dans l'atmosphère diminue proportionnellement, jusqu'à ce que le niveau de C_A atteigne $C_{A0} - (K - C_{T0})$.

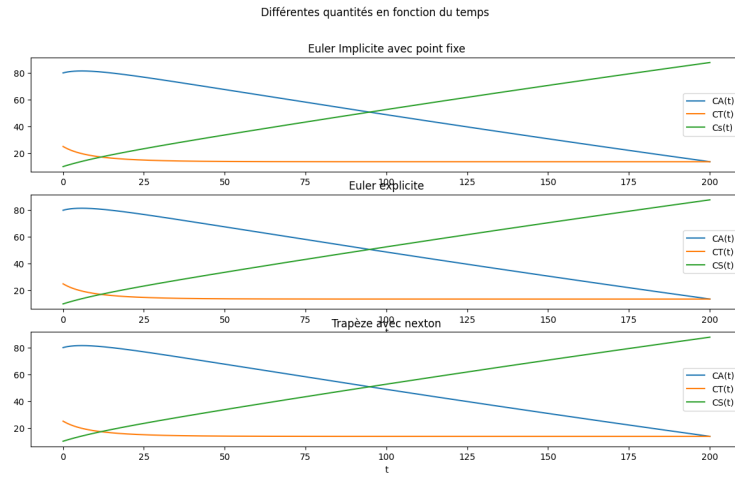


Figure 1: Courbe avec ($C_{T0} > K$) et ($h = 10$)

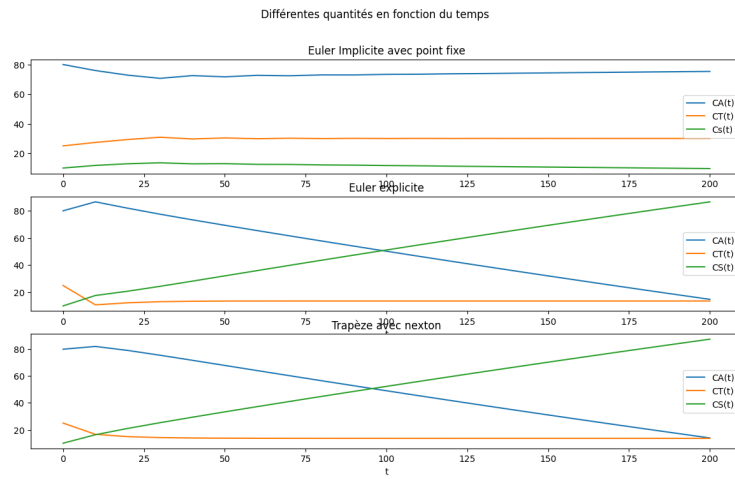
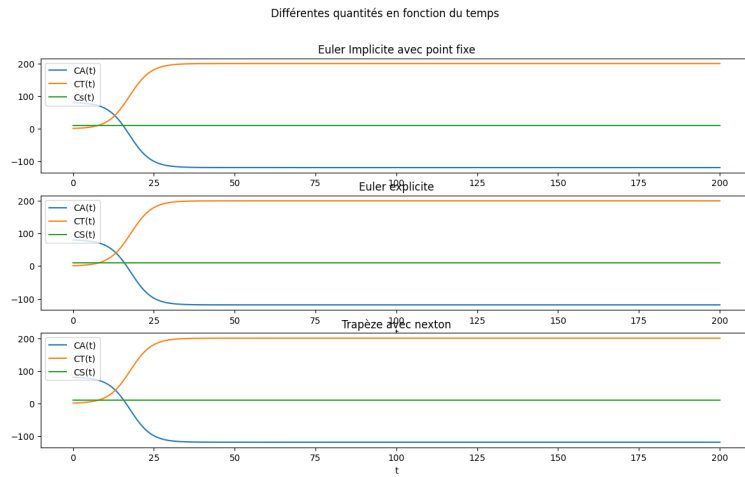


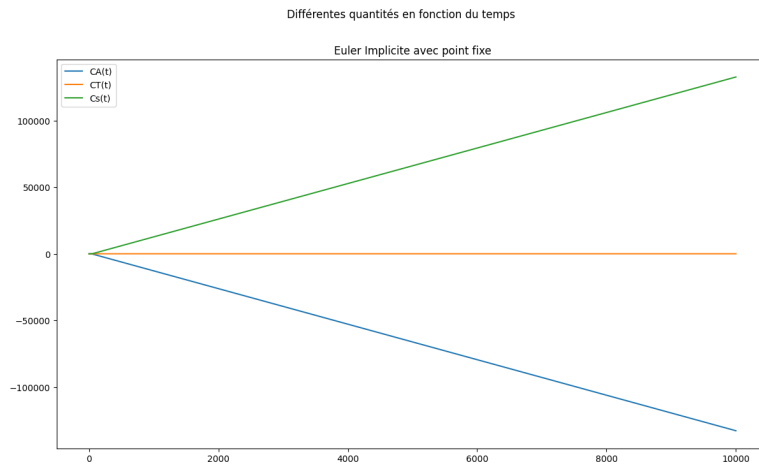
Figure 2: Courbe avec ($C_{T0} > K$) et ($h = 0.1$)



Paramètre : $K = 200$, $C_{A0} = 80$, $C_{T0} = 1$, $C_{S0} = 10$

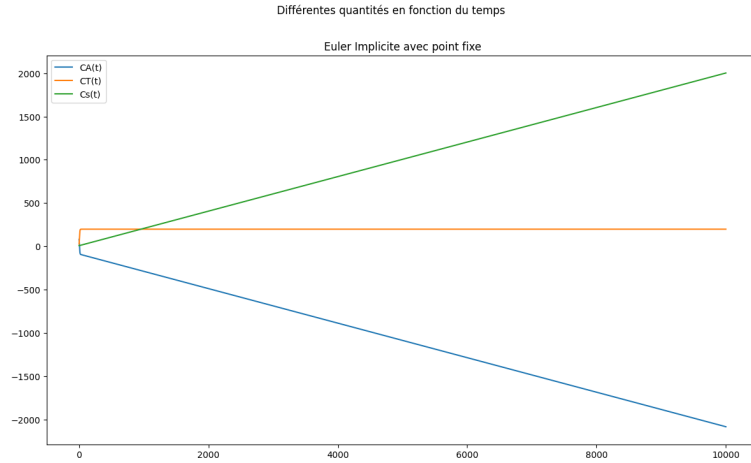
Transfert de carbone vers le sol Si on permet le transfert de carbone vers le sol grâce à la litière (feuilles mortes) en posant $\gamma = 0.2$

Le carbone absorbé par les arbres est transféré dans le sol, ce qui permet une séquestration plus efficace.



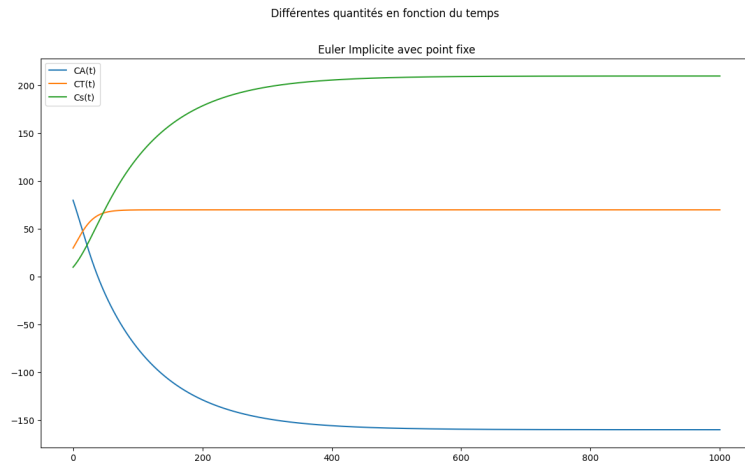
Cette représentation nous montre qu'ici le transfert de carbone entre les arbres et le sol est assez important pour avoir une séquestration très forte et très rapide.

Si on pose $\gamma = 0.001$,



On remarque que la séquestration est plus faible. **Mais quelque soit la valeur des paramètres α et γ si $\beta = \delta = 0$ on n'atteindra pas d'équilibre** puisque aucun carbone n'est rejeté dans l'atmosphère.

Transfert arbres, sol et atmosphère Lorsque le carbone est autorisé à être rejeté dans le sol par les arbres et dans l'atmosphère par le sol grâce à la respiration, avec des valeurs spécifiques pour $\delta = 0.01$, $\alpha = 0.1$, et $\gamma = 0.02$ et toujours $\beta = 0$, un équilibre est atteint. Dans cet équilibre, la quantité de carbone absorbée par les arbres est égale à celle rejetée dans l'atmosphère et le sol.



Paramètres : $C_{A0} = 80$, $C_{T0} = 30$, $C_{S0} = 10$

Transfert entre arbres et atmosphère On pose $\beta = 0.01$, on autorise la respiration directe entre les arbres et l'atmosphère.

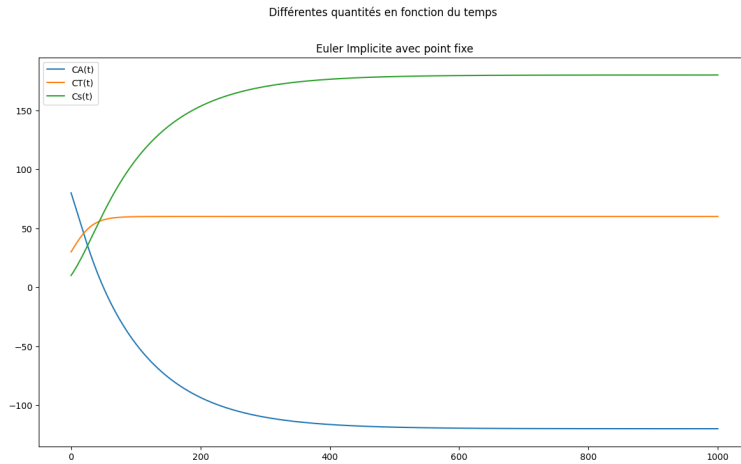


Figure 3: Courbe avec $\beta = 0.01$

On atteint l'équilibre à une quantité de séquestration plus faible car le système perd plus de carbone dans l'atmosphère en raison de l'augmentation de β , et la capacité d'absorption des arbres est limitée. On pourrait compenser cette perte en augmentant α et on pourra retrouver le même graphique que précédemment.

Si on augmente encore β avec $\beta = 0.08$ on observe une augmentation du carbone dans l'atmosphère. La quantité de carbone rejetée est plus élevée que celle absorbée.

Amélioration du Modèle

Dans notre modèle, on autorise C_A à devenir négatif, ce qui n'est pas réaliste d'un point de vue physique. Pour améliorer le modèle, on pourrait imposer une borne inférieure à C_A . On peut faire cela en remplaçant une partie du code dans la boucle par:

```
Cn = #calcul du nouveau vecteur
Cn[0] = max(0, Cn[0]) #Cn[0] correspond au terme CA
C = np.append(C, np.transpose([Cn]))
```

Les océans sont les plus grand puit de carbone de la planète, il est donc légitime de les prendre en compte dans notre modélisation. En présence d'un océan, on

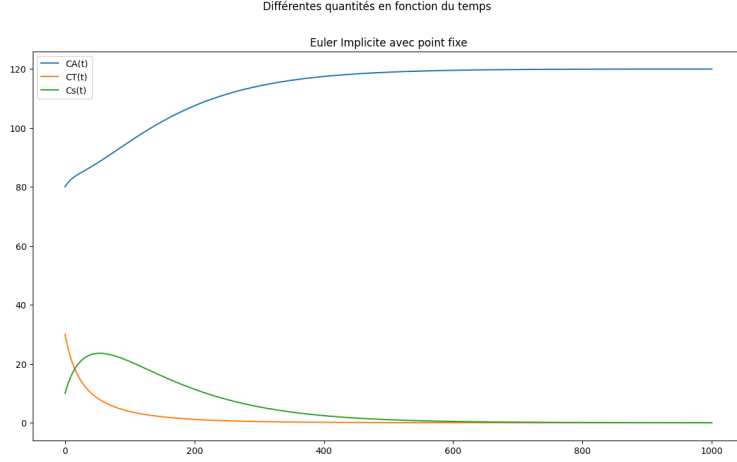


Figure 4: Courbe avec $\beta = 0.08$

peut rajouter une autre variable $C_O(t)$ la quantité de carbone dans l'eau. On a une interaction entre l'eau et l'atmosphère.

Pour rendre le modèle plus réaliste, on peut rajouter un terme source dans le terme $\frac{dC_A(t)}{dt}$ pour représenter l'émission de CO_2 par l'activité humaine. On choisie $\exp(t/1000)$ pour modéliser cela, ce choix est justifié en observant l'évolution du taux de CO_2 dans l'atmosphère causée par l'homme d'après l'étude de Global Monitoring Laboratory.

$$\begin{cases} \frac{dC_A(t)}{dt} = -S(C_T(t)) + \beta C_T(t) + \delta C_S(t) - \epsilon C_A(t) + \omega C_O(t) + e^{t/1000} \\ \frac{dC_T(t)}{dt} = S(C_T(t)) - \beta C_T(t) - \delta C_T(t) - \gamma C_T(t) \\ \frac{dC_S(t)}{dt} = \gamma C_T(t) - \delta C_S(t) + \delta C_T(t) \\ \frac{dC_O(t)}{dt} = \epsilon C_A(t) - \omega C_O(t) \end{cases}$$

avec $\epsilon C_A(t)$ qui représente le taux de séquestration du carbone dans l'océan et $\omega C_O(t)$ décrit l'effet de respiration de l'océan vers l'atmosphère.

Conclusion

Ce projet a permis de mieux intégrer les méthodes numériques vues en cours, et de les mettre en application sur un système d'équations différentielles, en plus de comprendre leur utilité dans l'application à des domaines concrets. Les résultats semblent cohérents avec nos prédictions, et la réalité (dans les limites du modèle), l'impact des différentes constantes est visible.

Annexe

L'océan, puits de carbone Source pour le CO₂ émis dans l'atmosphère

Répartition

Lucah, Amine, Gagik : prise de connaissance du sujet, lecture des sources (rapidement) : 3h

Lucah, Amine, Gagik : prise en main de github (création du projet, des branches etc) : 1h30

Gagik : Première approche théorique du problème avec la méthode implicite d'Euler + Point fixe (et code latex): 4h

Amine : implémentation des méthodes en python 2h

Lucah/Gagik : revue du code python, corrections de bugs, discussions des choix d'implémentations (plus pratique à utiliser et plus propre) 2h

Lucah : Début de la rédaction du rapport (en markdown) et contextualisation des enjeux : 2h

Gagik : amélioration du modèle en utilisant des trapèzes au lieu des rectangles : 30min

Gagik : mise en place de la méthode de newton (sur le papier puis en latex) : 2h

Gagik : implémentation des nouvelles méthodes des trapèzes ainsi que newton pour résoudre le problème : 2h30

Amine/Lucah : revue du code, correction des problèmes : 30 min

Lucah/Amine : discussions sur l'impact des différents paramètres, créations de graphes à l'aide du code : 4h

Amine/Gagik : Finalisation du rapport et (tentative de le transformer en format pdf) 2h30