

**CENTRALE  
LYON**

ÉCOLE CENTRALE LYON

BERNARD L'HERMITE

JUIN 2025

RAPPORT

---

## Synthèse du Projet CREPES & Pistes pour la suite

---

***Contributeurs :***

Baptiste BONELLI Zahara NASR Léo  
YOUSSEFI

Blandine DALLÈS Mélusine RIONDET  
Nolane LAVENU

Line LEMMET-VIDAL Soufiane  
KABCHI Maxence LUCAS MOREL  
Lilas BESSON Celyan DERPET

***Enseignants :***

CHEVEREAU - BERNARD

# Table des matières

- 1 Comment ouvrir un fichier.md ?**
- 2 Code principal**
- 3 Convection**
- 4 GUI**
- 5 Sensibilité du modèle à différents paramètres.**
- 6 Pistes d'amélioration**
  - 6.1 Variation de la température avec l'altitude . . . . .
  - 6.2 Température initiale en fonction du jour de l'année et de la zone géographique
  - 6.3 Découpage plus fin des capacités thermiques . . . . .
  - 6.4 Prise en compte de tous les termes sources . . . . .
  - 6.5 Pistes pour le Modèle de la Convection . . . . .
- 7 Annexe - Tuto StackEdit**

## 1 Comment ouvrir un fichier.md ?

Un fichier avec une extension ".md" est appelé un fichier "markdown". Quand il n'est pas compilé et est ouvert dans un éditeur de texte, son contenu peut paraître assez opaque, mais une fois compilé, il permet une mise en forme bien plus intéressante que celle permise pas une simple fichier texte.

Pour ouvrir un fichier avec une extension ".md", vous avez plusieurs solutions :

- Copier-coller son contenu sur GitHub ou l'importer sur Github. Une fois le fichier upload, il vous suffira de double-cliquer dessus et Github se chargera de le compiler.
- Faire appel à un éditeur en ligne, comme par exemple [StackEdit](#).

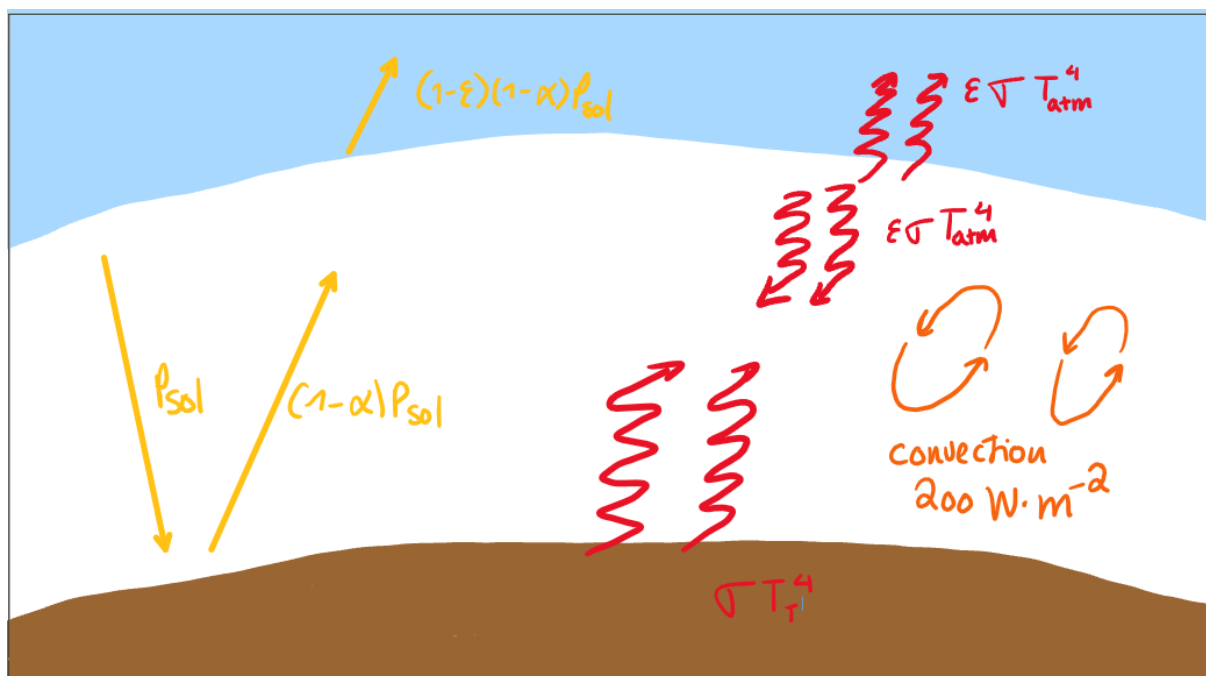
[Petit tuto pour StackEdit](#)

## 2 Code principal

[Lien vers le code](#)

Pour modéliser la Terre en dehors de son équilibre thermique, on applique à une couche de Terre et à une couche de l'atmosphère le premier principe de la thermodynamique.

On considère uniquement comme termes sources les rayons incidents du Soleil (dans le domaine du visible), les rayons émis par la Terre (dans le domaine de l'infrarouge) et ceux émis par l'atmosphère (dans le domaine de l'infrarouge). Les différentes contributions sont résumées dans le schéma suivant :



On aboutit au système suivant :

$$\begin{cases} dT_{Terre} = [(1 - \alpha)P_{sol} + \sigma(\epsilon T_{atmo}^4 - T_{Terre}^4)] \frac{dt}{C_{Terre} \rho_{Terre} Profondeur} \\ dT_{atmo} = [\sigma(\epsilon T_{Terre}^4 - 2\epsilon T_{atmo}^4) + P_{conv}] \frac{dt}{C_{atmo} \rho_{atmo} e_{atm}} \end{cases}$$

C'est ce système qui sert de base à notre code et que nous implémentons dans ce dernier par le biais de la méthode d'Euler.

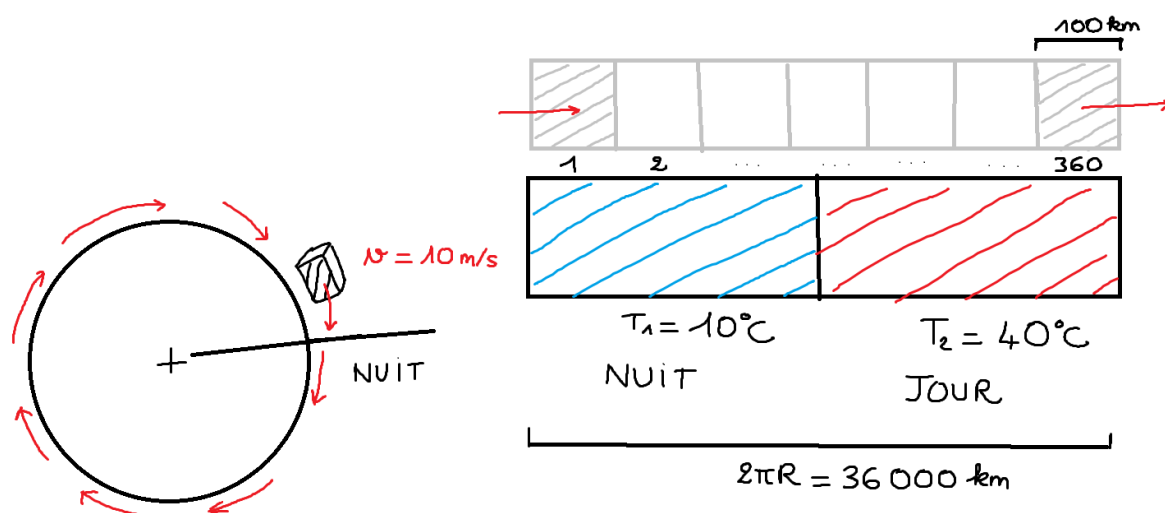
$\alpha$  correspond à l'albédo de la surface Terrestre considérée et est obtenu grâce à l'API de la NASA,  $\epsilon$  correspond à l'émissivité de l'atmosphère et est (si on oublie la dépendance avec les gaz à effet de serre) tabulée. "Profondeur" est la profondeur de Terre dans laquelle la température varie significativement au cours d'une journée. Nous l'avons calculée grâce à l'exercice n°1 du TD 11 de seconde année. Les autres notations correspondent aux grandeurs qui leur sont habituellement associées (Puissances, capacités thermiques, masse volumique, etc...).

### 3 Convection

[Lien vers le code](#)

Ce script simule le déplacement de blocs d'air fragmentés (représentant l'atmosphère) au-dessus d'un sol divisé en deux zones thermiques : une zone de nuit et une zone de jour, ayant des températures différentes.

Le principe est résumé dans le schéma suivant :



L'objectif est de visualiser les échanges thermiques selon la loi de Newton du refroidissement :

$$T_{\text{air}}(t + \Delta t) = T_{\text{air}}(t) + k \cdot (T_{\text{sol}} - T_{\text{air}}(t)) \cdot \Delta t \quad (1)$$

Avec :

- $T_{\text{sol}}$  : température du sol
- $T_{\text{air}}$  : température du bloc d'air
- $\Delta t$  : durée du pas de temps
- $k = \frac{h \cdot A}{m \cdot c}$
- $h$  : coefficient de convection thermique (10 W/m<sup>2</sup>/K)
- $A$  : surface d'échange du bloc
- $m$  : masse du bloc d'air

—  $c$  : capacité thermique massique de l'air ( $\approx 1000 \text{ J/kg/K}$ )

On voit ainsi l'évolution de la température des masses d'air au dessus de deux thermostats au cours du temps.

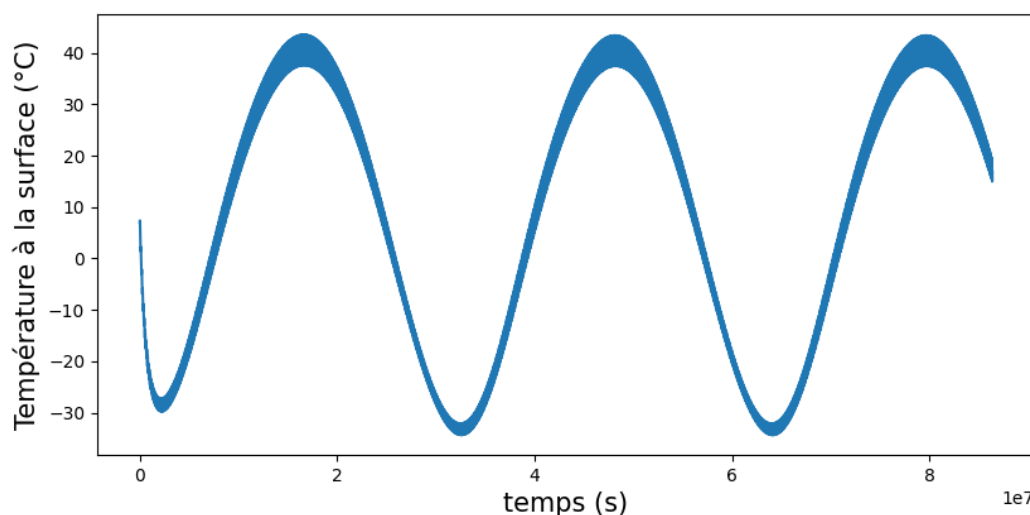
## 4 GUI

[Lien vers le code](#)

Ce programme Python utilise Tkinter pour afficher une carte du monde à gauche et une courbe de température à droite. La carte est colorée selon des estimations simplifiées des capacités calorifiques et des valeurs d'albédo moyennes basées sur des données NASA. En cliquant sur un point de la carte, le programme récupère sa latitude et sa longitude, les utilise comme entrée de la fonction Temp (dans Code\_avec\_appel\_biblio.py), et trace l'évolution annuelle de la température à cet emplacement. La courbe est interactive (zoom, déplacement). La durée de simulation est d'un an par défaut (modifiable dans le code). Bibliothèques à import : pip install (pandas, requests, cartopy).

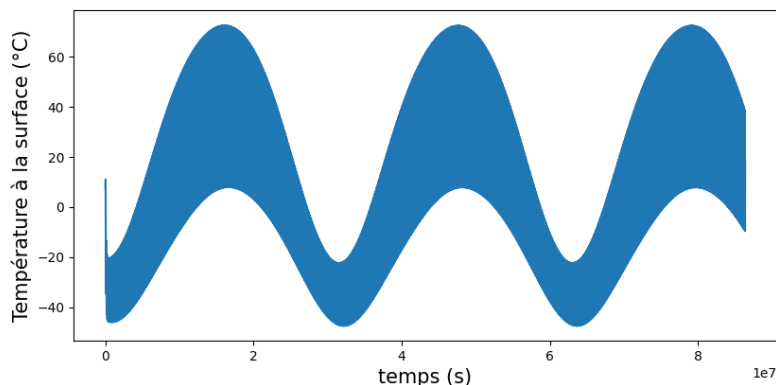
## 5 Sensibilité du modèle à différents paramètres.

Nous allons ici essayer de modifier certaines valeurs dans notre code et montrer comment elles impactent les résultats de notre modèle. Notre point de test sera Latitude 45 et Longitude 4 (proche du Puit en Velay, en FRANCE). Avec nos paramètres actuels, notre modèle renvoie la courbe suivante :

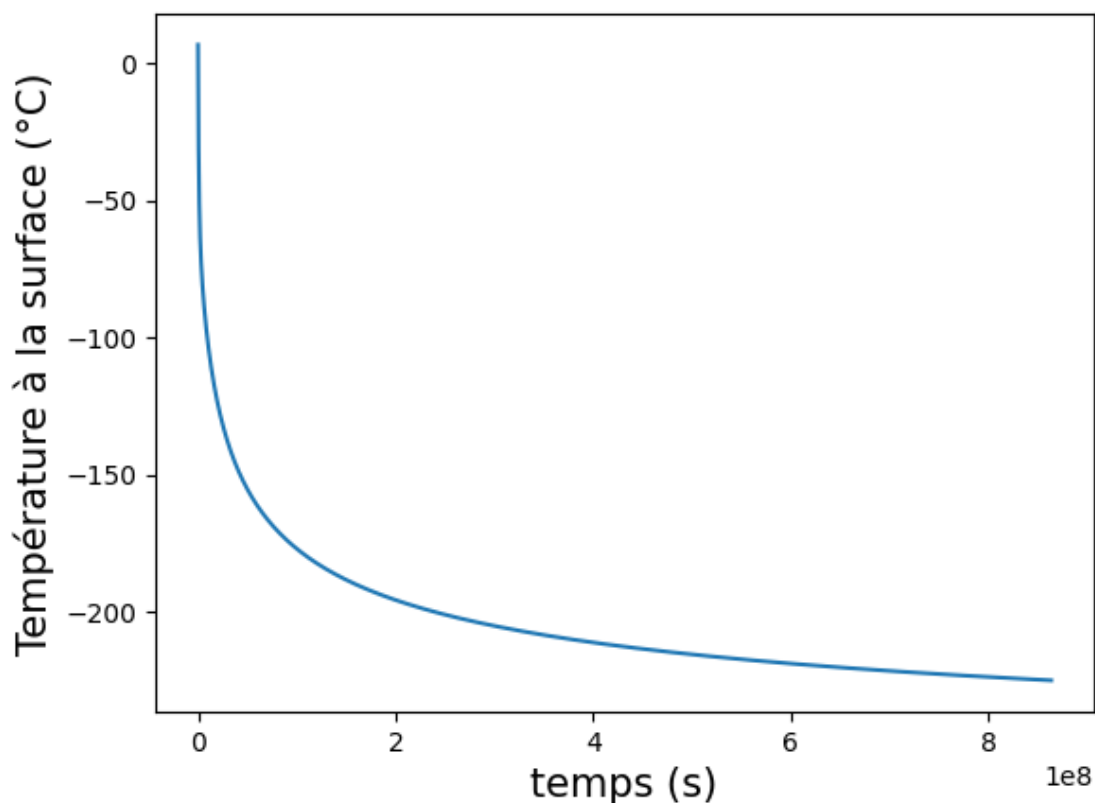


En faisant passer la profondeur caractéristiques de variation de 0.6m à 0.05m, on obtient la courbe suivant :

On observe que les variations quotidiennes de température sont beaucoup plus importantes que précédemment. Effectivement, les 5 premiers centimètres de profondeur de la Terre sont plus sensibles que les suivants.

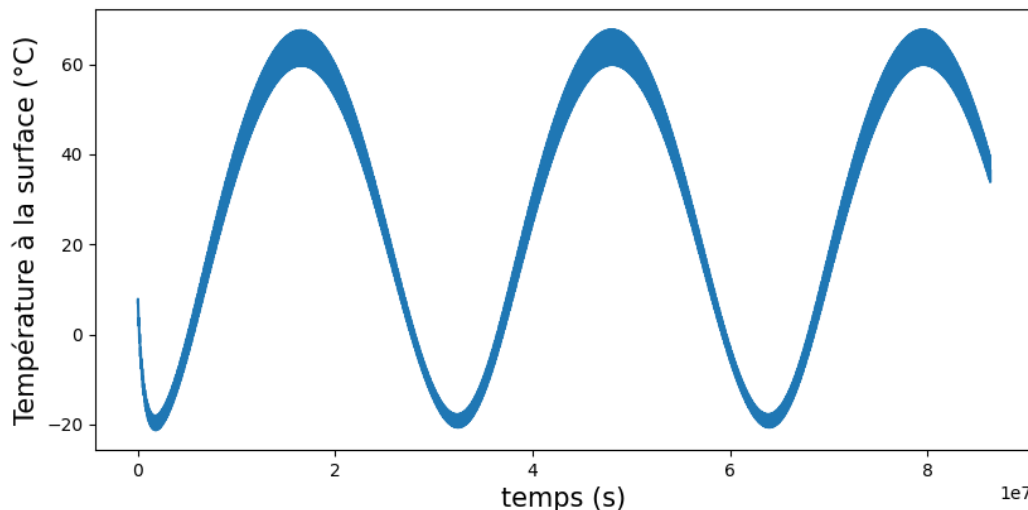


En revenant à une profondeur caractéristique de variation de 0.6 mètres, mais en prenant un albédo de 1 systématiquement (tout le rayonnement venant du Soleil est réfléchi par la surface Terrestre), on obtient la courbe suivante :



On remarque deux choses : Il n'y a plus de variations saisonnières ni quotidiennes, et la température ne fait que diminuer. Cela paraît cohérent : De jour comme de nuit et en été comme en hiver, aucun rayonnement n'est absorbé par la surface : Ainsi, il est logique, qu'il n'y ait aucune variation à cause de ces paramètres. De plus, la Terre n'absorbe littéralement aucune énergie, mais rayonne du fait de sa température non nulle (corps noir), il est donc logique que sa température ne fasse que diminuer : Elle tendra vers 0K.

En prenant, au contraire, un albédo constant égal à 0, on obtient la courbe suivante :



Cette courbe ressemble beaucoup à la courbe test, car l'albédo pris sur le point est très proche de 0.

## 6 Pistes d'amélioration

### 6.1 Variation de la température avec l'altitude

Nous avons jusqu'ici considéré que la Terre est en 2D, sans prendre en compte l'altitude. Il faudrait corriger cela en associant, à chaque couple de latitude et de longitude, une altitude correspondante. Pour connaître l'altitude d'un point à la surface de longitude et de latitude données, on peut s'appuyer sur des API de MNT (Modèle numérique de terrain), comme celle de Google Earth ([ici](#)), ou celle de SRTM ([ici](#)). Une fois qu'on a l'altitude, on peut modéliser la température de l'air, par exemple avec un modèle d'atmosphère isotherme et remonter (ou plutôt, redescendre), à la température de la surface de la Terre. De plus, le fait de monter en altitude diminue l'effet de serre car il y a moins de gaz à effet de serre au dessus du sol à hautes altitudes.

### 6.2 Température initiale en fonction du jour de l'année et de la zone géographique

Actuellement, nous initialisons la température de manière constante, cela rend la stabilisation de notre code plutôt lente (la température ne se stabilise vraiment qu'après plusieurs mois de simulation). Pour palier à cela, il faudrait initialiser la température différemment selon les zones géographiques et le jour de l'année.

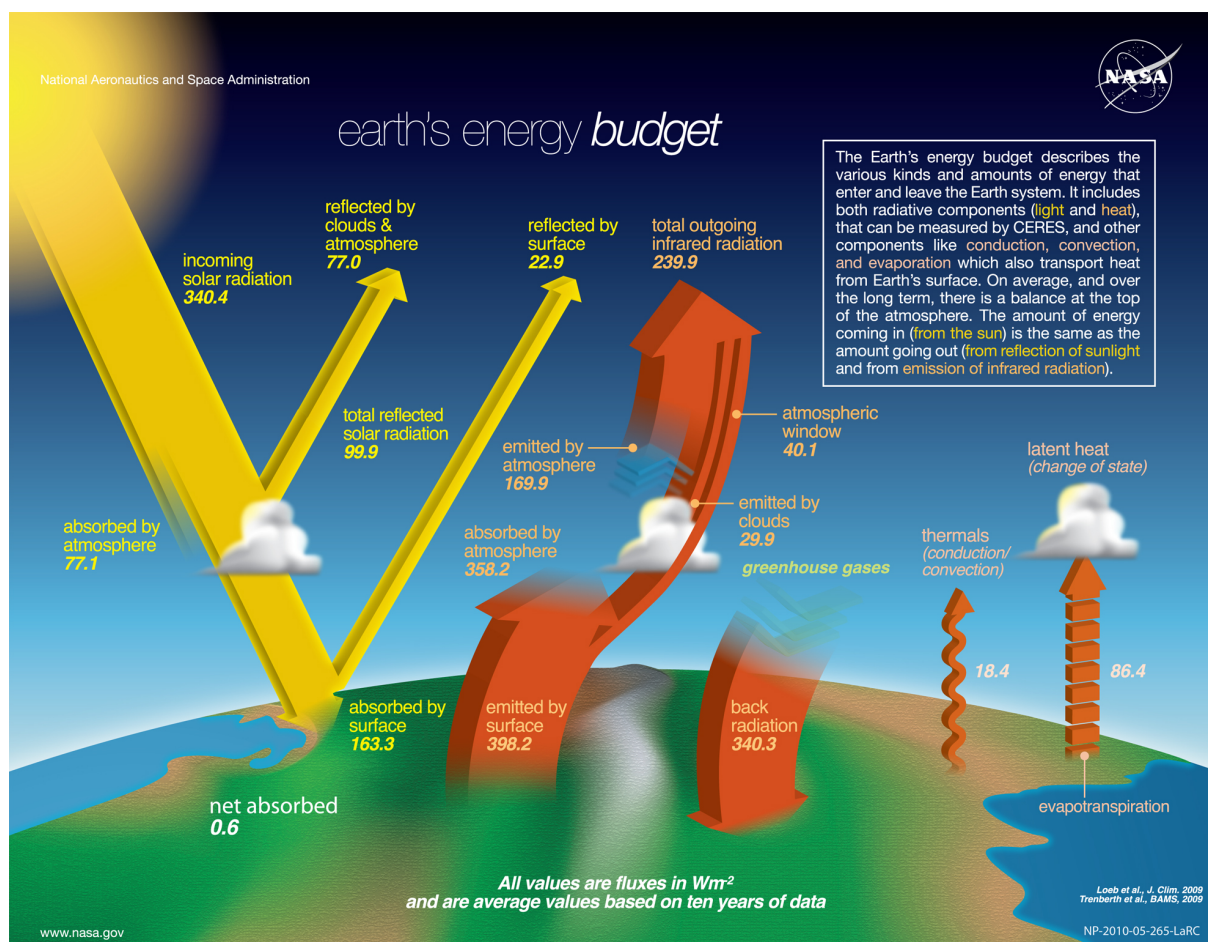
### 6.3 Découpage plus fin des capacités thermiques

Nous avons, dans le code actuel, un découpage très précis pour les albédos grâce à l'API de la NASA, mais un découpage assez sommaire pour les capacités thermiques. On

pourrait imaginer de coupler l'API de la NASA avec notre code qui calcule les capacités thermiques pour avoir un découpage plus fin pour les capacités thermiques.

## 6.4 Prise en compte de tous les termes sources

- 1 ère piste : Prendre en compte toutes (ou celles qui ne sont pas négligeables) les sources présentes sur le document ci-dessous et qui ne sont pas déjà prises en compte :



Cela regroupe donc :

- L'évapotranspiration.  
[Page Wikipedia.](#)  
[Une proposition de calculs](#)
- La chaleur latente (changements d'état)  
[Une proposition de documentation et de calculs](#)

## 6.5 Pistes pour le Modèle de la Convection

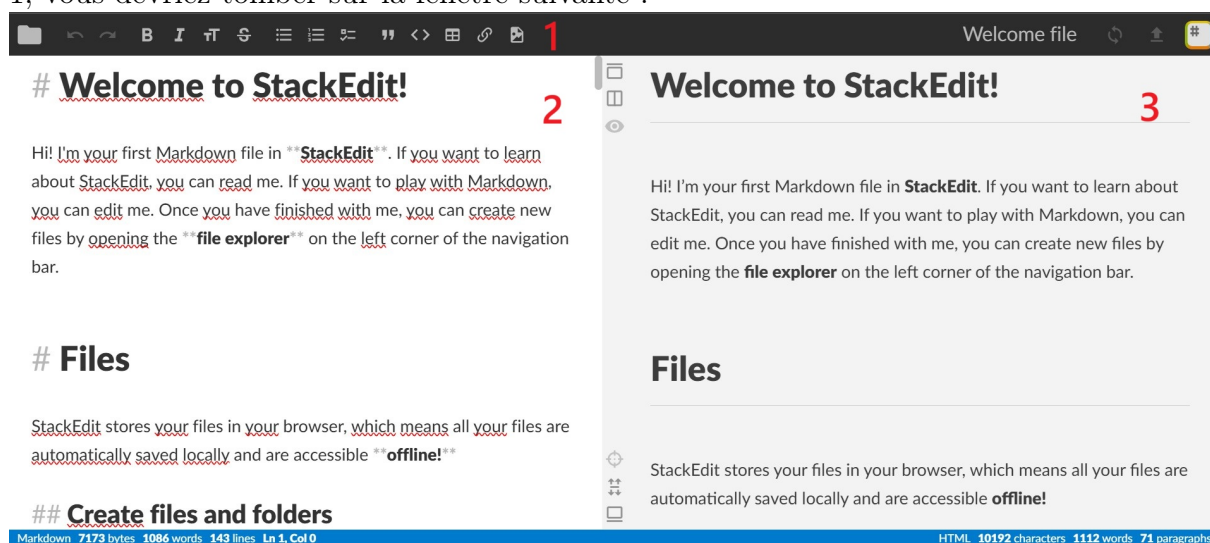
- 1D :
  - Pour le sol visuel : création d'un gradient de température entre le sol de nuit et le sol de jour



- Vitesse variable en fonction du temps
- $h$  variable en fonction de l'espace
- 2D :
  - Choix de la vitesse des vents : prise en compte de l'API des vents et ne plus prendre la seule valeur de 10m/s
  - Adaptation 1D/2D

## 7 Annexe - Tuto StackEdit

Une fois que vous avez atterri sur StackEdit par le biais du lien fourni dans la question 1, vous devriez tomber sur la fenêtre suivante :



Comme vous pouvez vous en rendre compte, la fenêtre StackEdit est divisée en trois parties distinctes :

- La première partie est le bandeau en haut de la page, numéroté "1", qui permet, à gauche, d'upload des fichiers .md depuis votre machine, pour les compiler. Il y a aussi des raccourcis pour mettre du texte en gras ou en italique.
- La seconde partie, à gauche est la partie dans laquelle vous pouvez écrire ou copier-coller avec la syntaxe des fichiers Markdown.
- La troisième partie, à droite est la partie dans laquelle on peut apercevoir le fichier compilé.