

Document de Synthèse

Table des matières

1	Objectif du projet	3
	Code global	3
2	Analyse de la version initiale du programme	3
3	Exécution du code	3
4	Fonctions ajoutées	4
4.1	Fonction <code>calc_power_temp</code> (modélisation de l'atmosphère)	4
4.2	Fonction <code>change_temp</code> (variation température au cours du temps)	5
4.3	Fonction <code>project_to_geographic()</code>	5
4.4	Fonction <code>temp_dans_csv()</code>	5
4.5	Fonction <code>get_cp()</code>	6
4.6	Fonction <code>puissance_cond()</code>	6
	Diffusion	6
5	Objectifs	6
6	Recherches - Données	7
7	Présentation des différentes formes de diffusion	7
7.1	Résolution de l'équation de diffusion	7
7.2	Comparaison entre diffusion radiale et diffusion orthoradiale	8
7.3	Approximation diffusion radiale, orthoradiale	9
8	Modélisation diffusion orthoradiale	10
8.1	Modèle 1	10
8.2	Modèle 2	11
8.3	Modèle 3	11
9	Modélisation diffusion radiale	12
9.1	Modèle 4	12
9.2	Modèle 5	13
	Capacité thermique	14
10	Objectif	14

11 Recherches - Données	14
12 Programme réalisé	14
12.1 Premier Code	14
12.2 Deuxième Code	15
12.3 Troisième Code	15
Convection	15
13 Objectif	15
14 Flux thermique convectif	16
15 Détermination du coefficient de convection h	16
16 Calcul de la température de l'atmosphère T_{air}	16
16.1 Calcul de $\frac{dT_{\text{ij}}}{dt}$	17
17 Limites et améliorations	18

1 Objectif du projet

Comprendre le code récupéré du projet du "groupe B" de l'année dernière et l'adapter pour pouvoir ajouter les nouvelles fonctions développées et ainsi modéliser la température terrestre en fonction du temps.

Code global

2 Analyse de la version initiale du programme

Initialement, la température terrestre était déjà modélisée en utilisant et combinant plusieurs fichiers de code.

- Le fichier code “modélisation” crée la Terre, ses repères et la divise en 30*60 surfaces élémentaires d’une coquille sphérique en fonction des longitudes et latitudes.
- Le fichier code “construction” crée des fichiers csv qui à chaque longitude-latitude associe une valeur d’albédo.
- Le fichier code “remplissage” attribue à chaque carré de la surface de la Terre la nuance de couleur correspondant à sa température.
- Le fichier code “échelle” crée une échelle de couleur en fonction de la température.
- Le fichier code “fonctions” contenant les fonctions suivantes :
 - `update_sun_vector`
 - `project_to_sphere`
 - `get_shape`
 - `get_albedo`
 - `slider_update`
 - `set_mois`
 - `calc_power_temp`
 - `change_temp`
 - `update_plot`

3 Exécution du code

- Il faut au préalable importer les modules shapefile, numpy et pandas si ce n’est pas déjà fait.
- Il faut télécharger le dossier "Final" qui regroupe tous les fichiers nécessaires pour exécuter les codes.
- Une fois l’interface chargée, pour pouvoir changer de mois, il faut sélectionner le mois voulu, cliquer sur ok dans le pop up chargement, puis faire bouger un peu la Terre pour que les couleurs s’actualisent.

4 Fonctions ajoutées

4.1 Fonction `calc_power_temp` (modélisation de l'atmosphère)

Fonction : `calc_power_temp`

Code Modèle 2_2025/Final/fonctions_modifie2.py

L'effet de serre était abordé dans la documentation commune et un nom de fonction lui avait été associé dans le programme (`calc_power_temp`) mais uniquement sous forme de commentaire. Cette fonction n'avait pas été codée à proprement parlé. Nous avons modéliser l'effet de serre puis codé dans la fonction `calc_power_temp`

Simplifications :

- La Terre et l'atmosphère sont considérées comme des corps noirs.
- L'atmosphère est transparente aux rayons du visible ([400 nm, 700 nm]) et opaque au rayonnement infrarouge (> 700 nm)
- La température de l'atmosphère est supposée fixe : 288 K

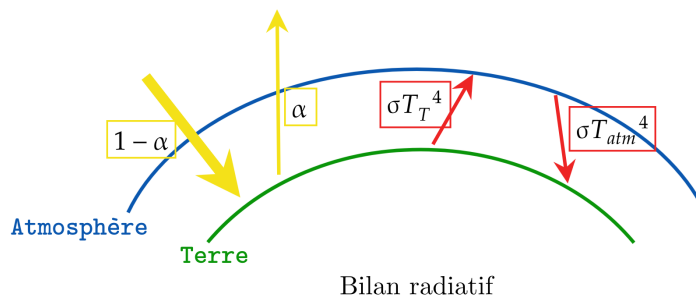
Modélisation :

Les rayons lumineux émis par le Soleil sont absorbés par la Terre.

La Terre réfléchit une proportion α de ces rayons, qui sont renvoyés vers l'espace (car l'atmosphère est transparente aux rayons lumineux) et en absorbe $1-\alpha$ (avec α qui dépend de l'albédo de la surface considérée).

Or la Terre est un corps noir de température non nulle donc elle réémet $\sigma T_{\text{Terre}}^4$ W·m⁻² dans toutes les directions sous forme de rayonnement infrarouge.

Or l'atmosphère est opaque aux rayonnements infrarouges donc l'atmosphère absorbe $\sigma T_{\text{Terre}}^4$ W·m⁻². L'atmosphère étant un corps noir de température non nulle, elle émet donc σT_{Atm}^4 dans toutes les directions.



4.2 Fonction change_temp (variation température au cours du temps)

Fonction : change_temp

Code Modèle 2_2025/Final/fonctions_modifie2.py

Nous avons créé une nouvelle fonction afin de modéliser et calculer la température heure par heure en fonction de la surface élémentaire de la Terre considérée. Pour ce faire, nous avons utilisé le premier principe de la thermodynamique sous la forme d'une suite récurrente avec une initialisation à T_0 , qui est la température du carré au premier jour du mois, première heure, calculée par la méthode statique :

Modèle temporel : $T_{n+1} = T_n + \frac{1}{C_p} (-\sigma T_n^4 + P_{r1}) \left(\frac{S_{Terre}}{1800} \right) dt$

Où :

- T_0 température à l'instant initial
- T_n température à n heures après l'instant initial
- T_{n+1} la température à n+1 heures après l'instant initial
- σ la constante de Boltzmann
- P_{r1} la puissance reçu par la Terre 1h après l'instant initial
- S_{Terre} la surface de la Terre
- 1800 le nombre de carrés du maillage de la surface de la Terre
- C_p la capacité thermique avec $C_p \frac{dT}{dt} = P_r S_r - P_t S_t$
 - P_r la puissance reçue par la Terre
 - P_t la puissance émise par la Terre
 - S_r la surface de la Terre éclairée par le Soleil
 - S_t la surface totale de la Terre

Afin de calculer plus facilement les températures pour chaque carré heure par heure, le code répertorie toutes les valeurs de température dans des fichiers csv (un fichier csv pour chaque mois). Chaque colonne donne la température du carré en fonction de chaque heure sur une journée ; il y a donc autant de lignes que de carrés. On utilise alors une boucle pour créer les csv d'une journée.

4.3 Fonction project_to_geographic()

Code Modèle 2_2025/Final/fonctions_modifie2.py

Cette fonction transforme des coordonnées selon x,y,z en latitude et longitude en degrés.

4.4 Fonction temp_dans_csv()

Code Modèle 2_2025/Final/fonctions_modifie2.py

Cette fonction lit des températures déjà calculées (fichiers csv) et affiche leur répartition sur le modèle 3D de la Terre. Elle colorie la surface selon la valeur de température à l'aide d'une échelle de couleurs et trace les continents. À chaque appel, elle efface l'ancienne surface (ax.clear()) puis trace la nouvelle distribution thermique pour le mois et l'heure choisis.

4.5 Fonction `get_cp()`

Code Modèle 2_2025/Final/fonctions_modifie2.py

Cette fonction renvoie C_p (la capacité thermique) du point géographique le plus proche des coordonnées (x,y,z) entrées en paramètres. Pour ce faire la fonction cherche dans un DataFrame (`list_cp`) contenant des coordonnées et leur capacité thermique associée et récupère les capacités des k points les plus proches du point (x,y,z) considéré. Ainsi la capacité du point (x,y,z) prend comme valeur, la moyenne des capacités des k points les plus proches.

4.6 Fonction `puissance_cond()`

Code Modèle 2_2025/Final/fonctions_modifie2.py

Calcule la puissance surfacique moyenne reçue à la surface par conduction pendant un temps choisie (normalement 1h), pour une température de surface `T_surf`. L'état thermique interne est mémorisé d'un appel à l'autre pour garantir la continuité. Pour cela, on utilise l'équation de diffusion avec la formule de Taylor.

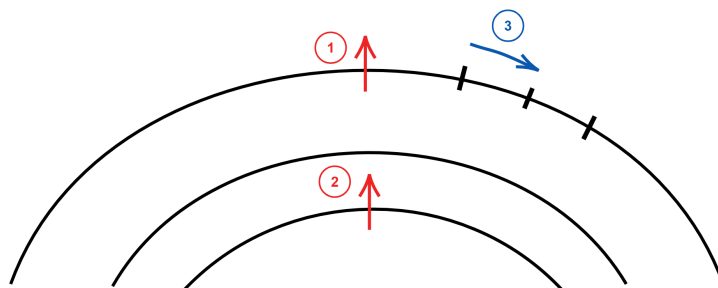
Diffusion

5 Objectifs

L'objectif est de modéliser la diffusion au niveau du sol afin de mesurer son impact sur la température terrestre au cours du temps.

On distingue 3 formes de diffusion :

- (2) celle entre les différentes couches terrestre (radiale)
- (3) celle à la surface de la terre (orthoradiale)
- (1) celle entre le sol et l'air (radiale)



Diffusion radiale / orthoradiale

6 Recherches - Données

Recherches intéressantes qui nous ont aidés et inspirés sur le sujet :

- PDF sur la chaleur et la température (ENS).
- PDF sur les transferts de chaleur dans la Terre (HAL).

Pour les calculs, on utilise le coefficient de diffusion thermique D qui dépend de λ , μ et C_p . On a donc effectué des recherches sur ces différents termes pour se donner une idée des ordres de grandeur. Source : Wikipedia.

Matériau	Masse volumique (kg.m^{-3})	Capacité thermique ($\text{J.K}^{-1}.\text{kg}^{-1}$)	Conductivité thermique ($\text{W.m}^{-1}.\text{K}^{-1}$)
Eau	1000	4185	0.6
Glace	917	2040	2.25
Béton	2400	880	1
Sable	1650	800	0.27
Granit	2750	890	2.9

7 Présentation des différentes formes de diffusion

7.1 Résolution de l'équation de diffusion

Pour modéliser notre démarche, nous sommes partis de l'équation de diffusion :

$$\frac{\partial T}{\partial t} = D \Delta T$$

- T la température (en K)
- ΔT le Laplacien de la température
- D le coefficient de diffusion (en $\text{m}^2.\text{s}^{-1}$)

et on a

$$D = \frac{\lambda}{\mu c_p}$$

Puis avec la discrétisation de la formule par Taylor, on retrouve cette formule :

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = D \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{(\Delta x)^2}$$

avec :

- i la position
- n l'itération dans la boucle

Puis, on peut résoudre cette équation informatiquement avec la méthode d'Euler (donc avec une boucle) grâce à un programme python. On va donc modéliser plusieurs situations (en partant d'une situation très simple pour la complexifier au fur et à mesure) et réaliser un code python pour chacune d'entre elles.

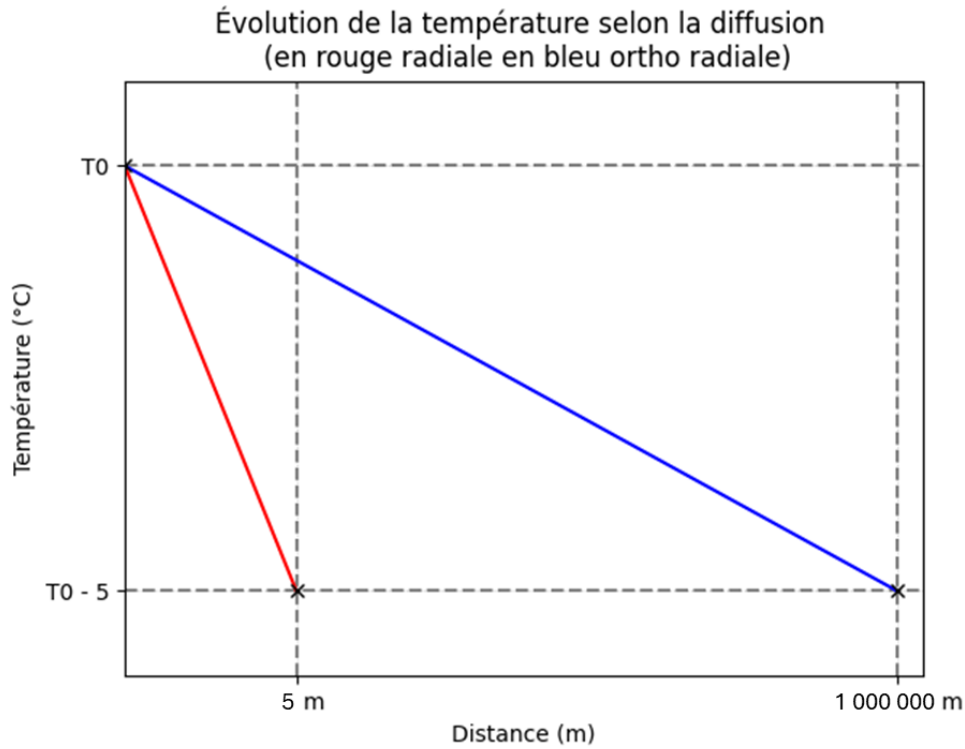
7.2 Comparaison entre diffusion radiale et diffusion orthoradiale

On s'intéresse au gradient de la température en coordonnées sphériques. Il possède une composante radiale (selon \vec{e}_r) et deux composantes orthoradiales (selon \vec{e}_θ et \vec{e}_φ). Cependant, les composantes orthoradiales sont négligeables devant la composante radiale.

Considérons une différence de température de 5°C entre 2 points.

A la surface de la Terre (composante orthoradiale), pour obtenir une différence de 5°C, il faut parcourir une distance de plusieurs centaines voir même mille kilomètres. Par exemple en regardant la météo, une différence de 5°C est présente entre ville voir entre pays (centaines de kilomètres)

En profondeur de la Terre (composante radiale), pour obtenir une différence de 5°C il suffit de quelques mètres. Par exemple, entre un sol en plein soleil et une cave à 5m sous terre, on retrouve déjà une différence de 5°C.



On remarque que la pente rouge est bien plus élevée que la pente bleue. La composante de ∇T sur \vec{e}_r est donc bien plus élevée que sur \vec{e}_θ et \vec{e}_φ (le gradient de température étant plus élevé là où la température grandit ou diminue plus rapidement donc là où la pente est plus élevée).

Or on a la formule du vecteur de diffusion de flux thermique qui est :

$$\vec{j} = -\lambda \nabla T$$

avec λ la conductivité thermique du matériau en $\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$. Donc comme les composantes du gradient de température en coordonnées orthoradiales sont négligeables, on a :

$$\vec{j} \approx -\lambda \frac{\partial T}{\partial r} \vec{e}_r$$

7.3 Approximation diffusion radiale, orthoradiale

Pour se donner une idée en ordre de grandeur du temps nécessaire à ce que la diffusion se fasse entièrement, on peut simplifier l'équation de diffusion :

$$\frac{\partial T}{\partial t} = D \frac{\partial^2 T}{\partial x^2}$$

En raisonnant en ordre de grandeur,

$$\tau \sim \frac{L^2}{D}$$

Avec L la distance caractéristique de diffusion et τ le temps caractéristique de diffusion.

Pour la diffusion orthoradiale, on prend comme distance typique de diffusion $L = 1$ km entre deux points. Avec le coefficient de la Terre (soit environ $10^{-6} \text{m}^2/\text{s}$) on obtient $\tau = 10^{12}$ s soit plus de 10^4 ans. Avec celui de l'eau (soit environ $10^{-7} \text{m}^2/\text{s}$), on obtient $\tau = 10^{13}$ s soit plus de 10^5 ans.

Si on s'intéresse aux ordres de grandeur de la diffusion radiale, on prend maintenant $L = 1$ m. Avec le coefficient de la Terre, on obtient $\tau = 10^6$ s, soit plus de 11 jours. Avec celui de l'eau, on obtient $\tau = 10^7$ s soit environ 3,8 mois. Ces temps caractéristiques de diffusion sont bien plus rapides.

8 Modélisation diffusion orthoradiale

8.1 Modèle 1

Explication

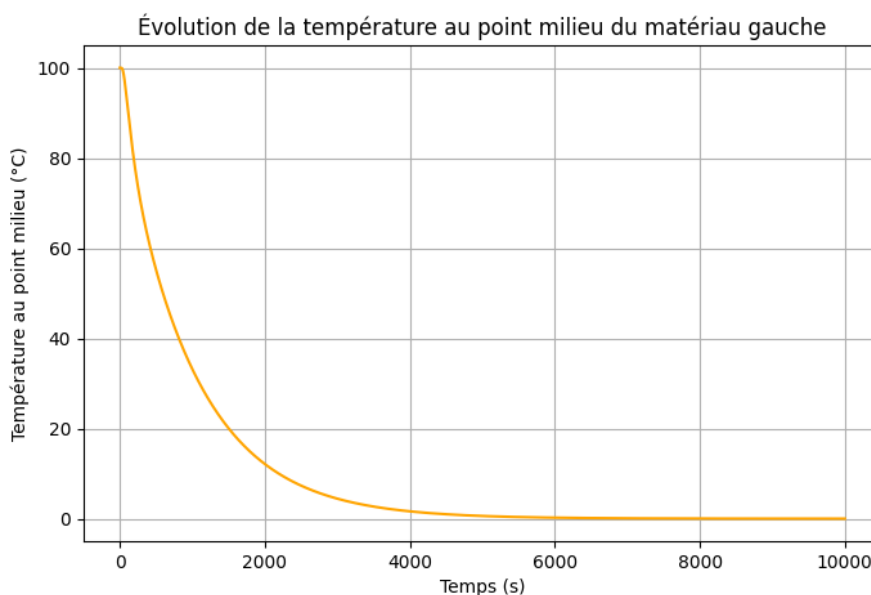
On considère deux plaques rectangulaires identiques côte à côte mais une à température T_1 , l'autre à température T_2 . Celle à T_2 reste fixe, T_1 varie jusqu'à atteindre T_2 . On prend comme données :

- $D = 10^{-4} \text{m}^2/\text{s}$ (coefficient de diffusion thermique)
- $T_1 = 100^\circ\text{C}$
- $T_2 = 0^\circ\text{C}$

Code

Modèle 2_2025/diffusion/conduction2plaquesfixe.py

Graphe



8.2 Modèle 2

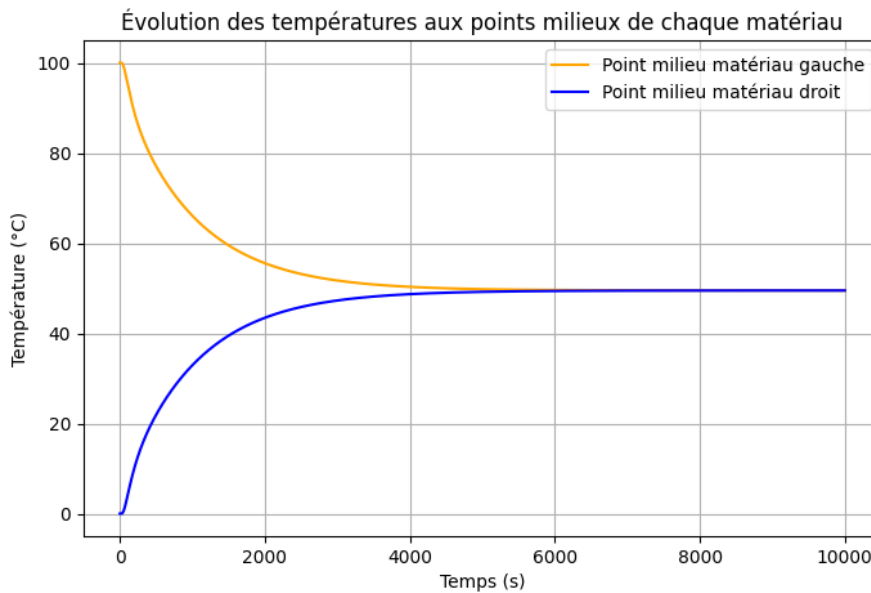
Explication

On reprend le modèle précédent mais cette fois T_2 n'est pas fixe, elle varie jusqu'à trouver un équilibre avec T_1 . Comme les plaques 1 et 2 sont identiques (mis à part leur température) la température d'équilibre entre les deux plaques doit être la moyenne des deux températures.

Code

Modèle 2_2025/diffusion/conduction2plaques.py

Graphe



8.3 Modèle 3

Explication

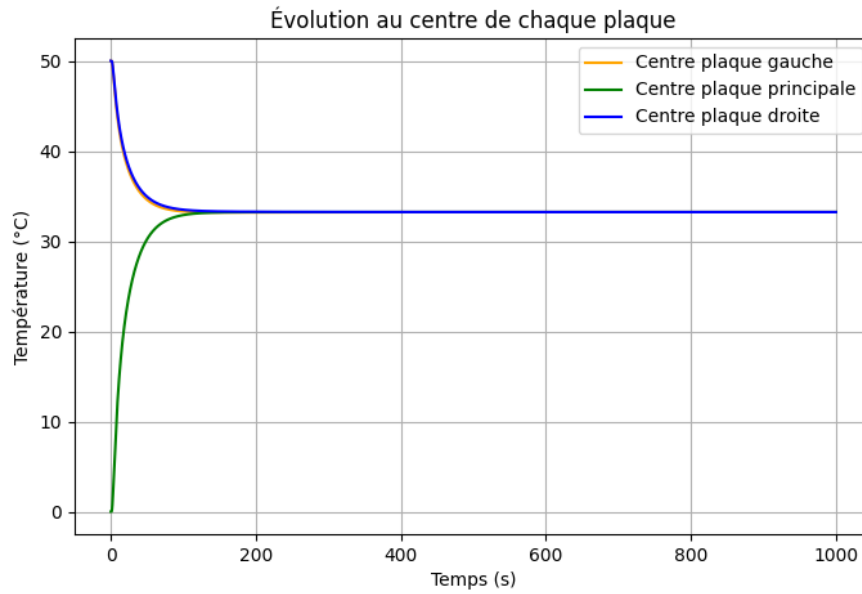
On considère trois plaques, toujours en 1D (donc toutes alignées). La plaque de gauche est à température T_1 , celle du milieu à T_2 et celle de droite à T_3 .

- $T_1 = 50^\circ\text{C}$
- $T_2 = 0^\circ\text{C}$
- $T_3 = 50^\circ\text{C}$

Code

Modèle 2_2025/diffusion/conduction3plaques.py

Graphe



9 Modélisation diffusion radiale

9.1 Modèle 4

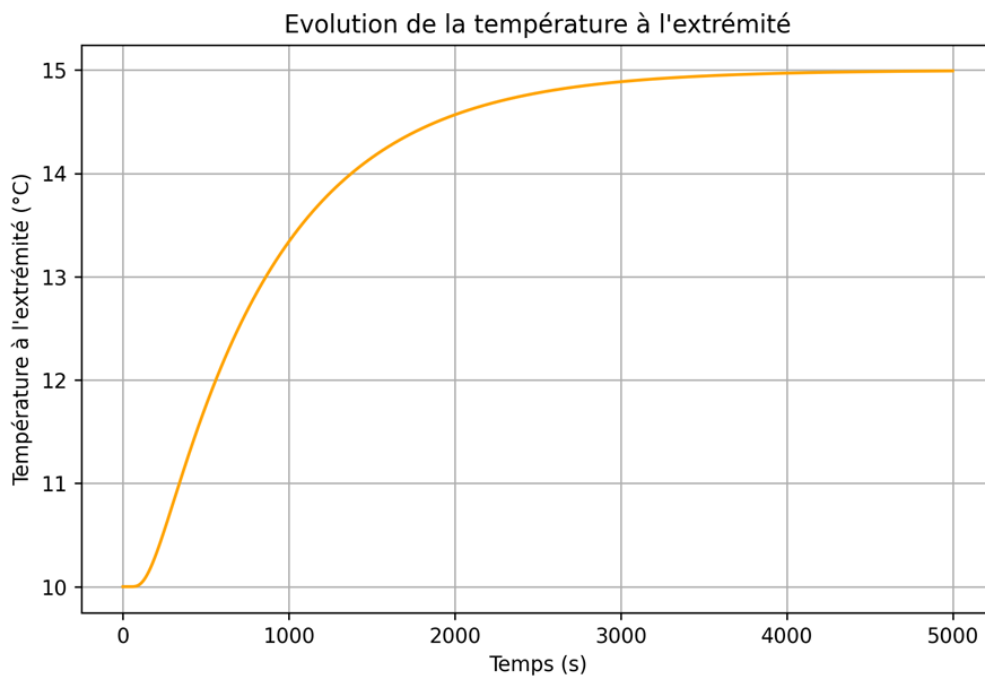
Explication

On est reparti du Modèle 1 de la partie diffusion orthoradiale mais en l'adaptant pour une diffusion radiale. on voit la variation de la température en surface avec une température en profondeur fixée (ce qui est semblable à une cave).

- $T_{\text{surf}} = 10^{\circ}\text{C}$
- $T_{\text{fixe}} = 15^{\circ}\text{C}$

Code Modèle 2_2025/diffusion/conductionradiale.py

Graphe



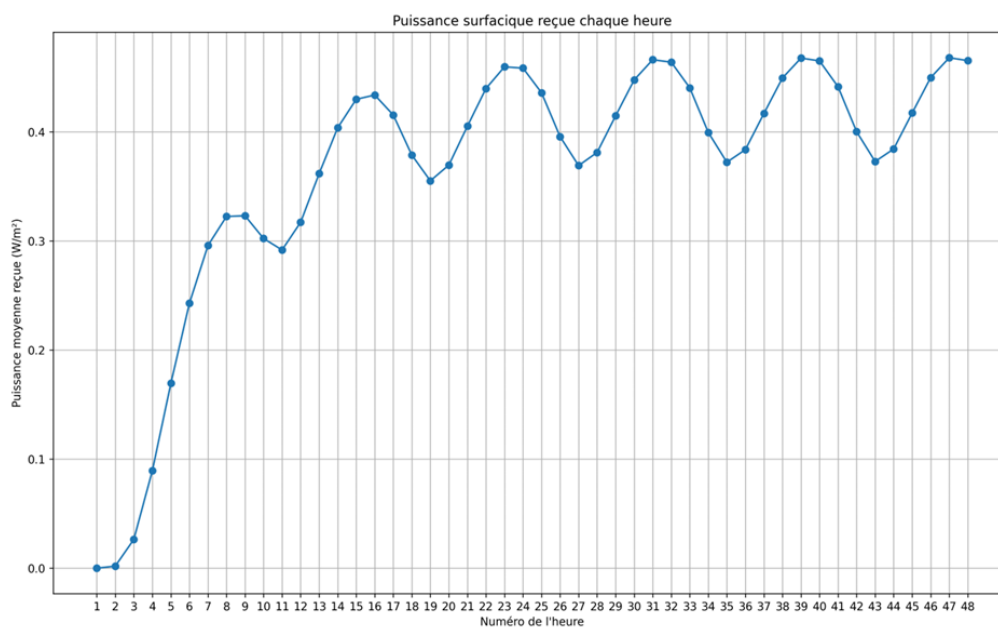
9.2 Modèle 5

Explication

On repart du modèle précédent mais en faisant un bilan de puissance pour pouvoir l'intégrer au code principal. Le code tient compte du fait que, chaque heure, la puissance repart de la valeur précédente. - Exemple de variation de la puissance avec une température de surface oscillante.

Code Modèle 2_2025/diffusion/diffusionpuissancefinal.py

Graphe



Capacité thermique

10 Objectif

Calculer la capacité thermique d'une parcelle et ainsi connaître la capacité thermique de chaque carré constituant le maillage de la Terre par notre programme.

11 Recherches - Données

Définition : la capacité thermique est la quantité de chaleur à fournir pour augmenter la température d'un corps de 1 degrés Kelvin (ou Celsius). La capacité thermique dépend donc du matériau considéré.

Matériau	Capacité thermique C_p (en $J \cdot kg^{-1} \cdot K^{-1}$)
Eau	4185
Glace	2040
Sol	1000
Béton	800
Sable	890

Simplification : Pour simplifier, on considère que la surface de la Terre est constituée uniquement de parcelles d'eau, de glace et de sol.

12 Programme réalisé

12.1 Premier Code

Code : Modèle 2_2025/capacité thermique/Utile /proportion terre glace eau.py

Code qui divise la Terre en 1800 parcelles (maillage de la Terre) de surface différentes car délimitées par des coordonnées GPS.

Puis à partir d'un raster geotiff implémenté au code grâce au package python rasterio, le code détermine la proportion d'eau, de glace et de sol sur les différentes parcelles.

Enfin le code renvoie un fichier csv contenant des différents pourcentages de matériau en fonction des coordonnées GPS.

Raster geotiff : fichier image géoréférencé qui associe des données spatiales (géographiques) à un format de type TIFF (Tagged Image File Format)

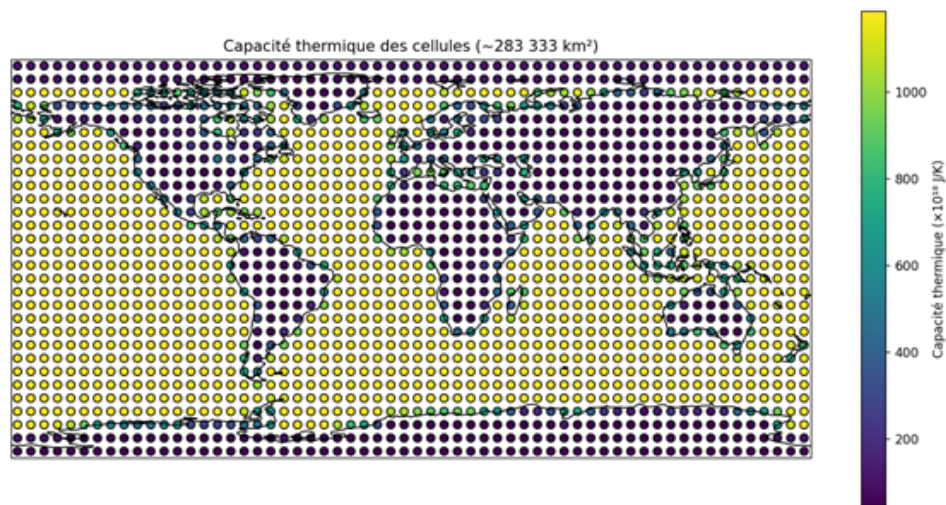
12.2 Deuxième Code

Code : Modèle 2_2025/capacité thermique/Utile/Calcul_1800_capacité_fraction_terrestre.py

Le deuxième code utilise les fichiers csv créés par le premier code. Il utilise les proportions de matériau de chaque pacerelle afin de leur associer une capacité thermique moyenne.

Ce code renvoie :

- un fichier csv des différentes capacités thermiques moyennes en fonction des coordonnées GPS
- une carte représentant sous la forme d'une échelle de couleur la capacité thermique moyenne d'une pacerelle



12.3 Troisième Code

Code :

Modèle 2_2025/capacité thermique/Programmes test capa thermique/sum_capacity.py

Le troisième code somme les capacités thermiques moyenne des différentes pacerelles pour donner la capacité thermique de la Terre sur une épaisseur de 5 cm. On obtient ainsi grace au programme une valeur de capacité thermique pour la Terre de $8.12 \times 10^{19} \text{ J} \cdot \text{K}^{-1}$

Convection

13 Objectif

L'objectif est de modéliser la convection terrestre afin de mesurer son impact sur la température terrestre au cours du temps.

14 Flux thermique convectif

Pour modéliser la convection, on cherche le flux thermique convectif entre le sol et l'air :

$$\vec{j} = -h(T_{\text{sol}} - T_{\text{air}})\vec{e}_x$$

Pour obtenir j il faut donc trouver avoir h , T_{air} et T_{sol} . La température du sol T_{sol} est déterminée par le groupe de diffusion. Il reste déterminer la valeur h et la température de l'air T_{air} .

15 Détermination du coefficient de convection h

Pour trouver h , on utilise la définition de différents nombres :

- le nombre de Nusselt : $Nu = \frac{hL_c}{\lambda}$
- le nombre de Grashof : $G_r = \frac{g\beta(T_{\text{sol}} - T_{\text{air}})L_c^3}{\nu^2}$
- le nombre de Prandtl : $Pr = \frac{\nu}{\alpha}$
- le nombre de Rayleigh : $Ra = G_r Pr$

Avec :

- L_c la longueur caractéristique (5×10^{-2} m)
- λ la conductivité thermique de l'air $26 \times 10^{-3} \text{ W.m}^{-1}.\text{K}^{-1}$
- g la constante de gravité (9.81 m.s^{-2})
- $\beta = 1/T_{\text{air}}(K - 1)$
- ν viscosité cinématique de l'air ($1.5 \times 10^{-5} \text{ m}^2.\text{S}^{-1}$)
- α diffusivité thermique de l'air ($2 \times 10^{-5} \text{ m}^2.\text{S}^{-1}$)

En mettant toutes ces formules en relation, on obtient donc une approximation de h pour chacune des 1800 plaques de la Terre en fonction de la température du sol et de celle de l'air au niveau de ces plaques.

Code pour calculer h : <https://github.com/KekeLeChef/Ornithorynquietant/blob/main/Mod>

16 Calcul de la température de l'atmosphère T_{air}

La température de l'atmosphère va dépendre de chaque plaque. On nomme T_{ij} la température de l'atmosphère à la position ij . En faisant une discrétisation, on obtient :

$$T_{ij}(t + dt) = T_{ij}(t) + \frac{dT_{ij}}{dt}dt$$

Avec :

- $dt = 3600\text{s}$ (choix)
- $T_{ij}(t = 0\text{s}) = 15^\circ\text{C}$
- $\frac{dT_{ij}}{dt}$ à déterminer

16.1 Calcul de $\frac{dT_{ij}}{dt}$

On a : $\frac{dT_{ij}}{dt} = \frac{P_{th}}{C_{th}} + v_{atm} \vec{\nabla} T$

Avec :

- P_{th} la puissance que reçoit l'atmosphère au total
- C_{th} la capacité thermique de l'atmosphère
- v_{atm} la vitesse de l'atmosphère
- $\vec{\nabla} T$ le gradient de température de l'atmosphère

Calcul de P_{th} :

La puissance solaire reçue est calculée à partir de l'angle d'inclinaison du Soleil, qui varie selon les mois. La formule utilisée est :

$$P_{solaire} = S_0 \cdot \cos(\theta)$$

où $S_0 = 1361 \text{ W/m}^2$ est la constante solaire (puissance du rayonnement solaire au sommet de l'atmosphère), et θ est l'angle d'incidence calculé en fonction de l'inclinaison de la Terre ($23,44^\circ$). Il n'y a pas de rayonnement si le Soleil est sous l'horizon.

Le code `puissance_solaire_recu.py` génère un dossier `puissance_recu_atm`, contenant un fichier pour chaque mois et chaque heure. Ces fichiers sont ensuite exploités dans le code `modele_convectif.py`.

Calcul de C_{th} :

La capacité thermique C_{th} de chaque section est donnée par la formule :

$$C_{th} = \frac{V_{atm} \cdot \mu_{air} \cdot c_{m,th}}{1800}$$

où :

- V_{atm} est le volume total de l'atmosphère,
- μ_{air} est la masse volumique de l'air,
- $c_{m,th}$ est la capacité thermique massique de l'air,
- 1800 est le nombre de sections (ou plaques) dans le modèle.

Calcul de \vec{v}_{atm} :

L'atmosphère terrestre est constamment en mouvement. Ce mouvement est organisé à grande échelle selon des schémas de circulation appelés cellules atmosphériques. Ces cellules jouent un rôle fondamental dans la redistribution de la chaleur en équilibrant les écarts d'énergie dus à l'inégale répartition du rayonnement solaire. On a donc divisé les hémisphères en 3 couches :

Cellule de Hadley (30°N-0°,0°-30°S)

- L'air chaud s'élève près de l'équateur
- En altitude, il se déplace vers les pôles, puis redescend autour des 30° de latitude
- Ce retour d'air en surface forme les alizés, soufflant vers l'équateur.

Cellule de Ferrel (30°N-60°N,30°S-60°S)

- L'air remonte près de 60° de latitude et redescend vers 30°, à l'inverse de la Hadley.
- Responsable des vents d'ouest

Cellule polaire (60°N-90°N,60°S-90°S)

- L'air froid descend au niveau des pôles et se dirige vers les latitudes plus basses (60°) et remonte en altitude
- Génère les vents polaires.

De plus, la rotation de la Terre dévie les vents vers la droite dans l'hémisphère nord, et vers la gauche dans l'hémisphère sud (sauf pour les cellules de ferrel qui tourne dans le sens inverse car les vents se déplacent vers le nord) . Cet effet est l'effet coriolis il peut être modélisé par une force : $\vec{F}_{\text{Coriolis}} = -2m\vec{\Omega} \times \vec{v}$. Cela influence fortement la direction des vents produits par les cellules atmosphériques :

- Alizés \rightarrow (hém. Nord) -135°, (hém. Sud) 135° ; $v_{\text{atm}} = 20$ km/h
- Vents d'ouest \rightarrow (hém. Nord) 45°, (hém. Sud) -45° ; $v_{\text{atm}} = 30$ km/h
- Vents polaires \rightarrow (hém. Nord) -135°, (hém. Sud) 135° ; $v_{\text{atm}} = 15$ km/h

Étant donné que l'air chaud remonte et se refroidit en altitude, on peut étudier que le vecteur vitesse vent au contact de la surface et supposer que l'air arrive à la surface à une température constante initiale.

Calcul de $\vec{\nabla}T$:

On considère l'axe (Oy) le long de l'axe de rotation de la Terre et l'axe (Ox) le long de l'équateur. On a calculé le gradient de température sur l'axe (Oy) et (Ox) séparément. Pour ça, on résout l'équation du gradient de température avec la méthode d'Euler.

Pour la composante sur (Ox) : on le calcule pour une latitude choisie et une longitude allant de -180° à 180°, pour une heure H.

Pour la composante sur (Oy) : on le calcule pour une longitude choisie et une latitude allant de -180° à 180°, pour une heure H.

Code pour le gradient sur (Ox) : `Modèle 2_2025/convection/gradient_temperature_x.py`.

Code pour le gradient sur (Oy) : `Modèle 2_2025/convection/gradient_temperature_y.py`.

17 Limites et améliorations

- Hétérogénéités topographiques de la surface terrestre.

- Dynamique des océans non prise en compte : « absorbe 90% du surplus de la chaleur générée par l'augmentation de la concentration des gaz à effet de serre, due aux activités humaines, tandis que seulement 1% de ce surplus se retrouve dans l'atmosphère. On peut donc dire, sans hésiter, que le réchauffement global coïncide avec le réchauffement de l'océan ».
- Modélisation plus détaillée de l'effet de serre, en tenant compte des différents gaz présents dans l'atmosphère et de leur rôle respectif, notamment à travers leurs coefficients d'absorption.
- Interface utilisateur peu fluide. Il faudrait optimiser le code pour améliorer la réactivité.
- La chaleur produite par le noyau terrestre en fusion n'a pas été prise en compte. On peut supposer qu'elle représente une part non négligeable de l'énergie disponible à la surface terrestre.
- Changer le coefficient de diffusion provisoire dans `puissance_cond`.
- Trouver la température provisoire en profondeur dans `puissance_cond`.
- Séparer le maillage en surfaces égales.
- Prendre en compte que la température de la Terre varie au cours du temps et n'est pas constante (pas 15°C tout le temps).