

Documentation du projet

Formule de Boltzmann :

La loi de Stefan-Boltzmann est une formule utilisée pour déterminer la quantité totale d'énergie radiative émise par un objet en fonction de sa température.

$$P = \sigma A T^4$$

Termes Décomposés

1. Puissance (P) : C'est la quantité totale d'énergie rayonnée par l'objet en une seconde. Pensez à cela comme à la quantité totale de chaleur ou de lumière émise. en watts (W).
2. Constante de Stefan-Boltzmann (σ) : C'est un nombre fixe que nous utilisons dans la formule. Sa valeur est $\sigma = 5.67 \times 10^{-8} \text{ W m}^{-2} \text{ K}^{-4}$.
3. Surface (A) : C'est la surface totale de l'objet qui émet la chaleur ou la lumière. Imaginez une lampe chauffante : plus la surface de la lampe est grande, plus elle peut émettre de chaleur.
4. Température (T) : C'est la température de l'objet, mesurée en kelvins (K). Plus un objet est chaud, plus il émet d'énergie.

Exemple Simple

Imaginez deux objets :

1. Objet A : Température = 300 K (environ 27°C)
2. Objet B : Température = 600 K (environ 327°C)

Si l'objet B à une température double de l'objet A, il émettra $2^4 = 16$ fois plus d'énergie que l'objet A. Cela montre à quel point la température a un grand effet sur la quantité totale d'énergie rayonnée.

Explication du code python:

update_sun_vector(mois):

But : Met à jour le vecteur du soleil en fonction du mois de l'année.

Comment fonctionne la def :

Calcule l'angle d'inclinaison saisonnier basé sur le mois donné.

Créer une matrice de rotation pour ce nouvel angle d'inclinaison.

Applique cette matrice de rotation au vecteur du soleil d'origine pour obtenir le nouveau vecteur du soleil.

Retourne le vecteur du soleil mis à jour.

calc_power_temp(time, mois):

But : Calcule la puissance solaire reçue et la température en fonction de l'heure et du mois.
Comment fonctionne la def :

- Calcule l'angle de rotation en fonction de l'heure donnée.
- Créer une matrice de rotation pour cet angle.
- Applique cette matrice de rotation au vecteur du soleil mis à jour pour le mois.
- Calcule le cosinus de l'angle d'incidence des rayons solaires.
- Calcule la puissance solaire reçue par unité de surface.
- Calcule la température en utilisant la loi de Stefan-Boltzmann.
- Retourne la puissance reçue et la température.

effet_de_serre(puissance_recue, C_CO2=C_CO2_moy, C_H2O=C_H2O_moy)

But : Calcule l'effet de serre basé sur la puissance reçue et les concentrations de CO2 et H2O.

Comment fonctionne la def :

- Calcule la puissance de référence émise par la Terre à 15°C.
- Calcule un coefficient moyen basé sur la différence entre cette puissance de référence et la puissance reçue.
- Ajuste ce coefficient moyen en fonction des concentrations de CO2 et de H2O.
- Créer un masque pour éviter les divisions par zéro.
- Calcule la puissance émise en utilisant les coefficients ajustés.
- Calcule la température résultante en combinant la puissance reçue et émise.
- Retourne la puissance totale (reçue + émise).

update_plot(time, mois=3)

La fonction *update_plot* met à jour un graphique 3D pour visualiser la puissance radiative reçue par une sphère représentant un astre. Elle calcule la puissance reçue en fonction de l'heure et du mois, puis trace la surface de l'astre et son atmosphère. Elle ajoute également des contours côtiers projetés sur la sphère et configure les axes et le titre du graphique. Enfin, elle rafraîchit la figure pour afficher les nouvelles données.

slider_update(val)

But : Met à jour le graphique lorsque la valeur du slider change.
Comment fonctionne la def :

- Appelle *update_plot* avec la valeur actuelle du slider et le mois courant.

set_mois(mois)

But : Met à jour le mois courant et réinitialise le graphique.
Comment ça marche :

- Change le mois courant.
- Met à jour le vecteur du soleil pour le nouveau mois.
- Réinitialise le slider à zéro.
- Met à jour le graphique pour le nouveau mois

project_to_sphere(lon, lat, radius=1)

But : Convertir des coordonnées de longitude et latitude en coordonnées cartésiennes sur une sphère de rayon donné. On l'utilise pour projeter les coordonnées des contours des continents sur le globe.

Comment ça marche :

Convertit les angles de longitude et latitude de degrés en radians.

Calcule les coordonnées x, y, z correspondantes sur la sphère.

Retourne les coordonnées x, y, z.

Limites du code/ éléments non pris en compte:

-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».

- courant marin : gulf stream, courant équatorial, courant du Labrador

- modélisation de l'effet de serre plus précise, avec les différents gaz dans l'atmosphère leur rôle dans l'effet de serre c a d coeff d'absorption.

-Aérosols et particules Atmosphériques : contre l'effet de serre permet le forçage radiatif négatif et de diminuer la température de surface.

-Humidité et évaporation : l'évaporation a un effet sur la répartition de la chaleur.

-Transfert thermique par convection et conduction entre la surface de la terre et l'atmosphère.

-Effet de Coriolis dû à la rotation de la terre influençant la direction des vents et des courants océaniques.

-Topographie terrestre, relief etc ...

-approximation sur la l'épaisseur de l'atmosphère.

L'approximation sur l'albédo

-Circulation atmosphérique : Les vents transportent des masses d'air chaud ou froid.

-Couverture nuageuse : affecte la quantité de rayonnement solaire reçue et la perte de chaleur par rayonnement infrarouge.

- Une ré-implémentation de manière plus intelligente d'un temps caractéristique, lié à la diffusion dans le sol/ dans l'atmosphère.

- Augmentation de la précision du maillage. Ici on a que 1800 cases différentes, mais on pourrait augmenter ce nombre pour avoir des données plus précises.
- Acquisition des données albedo faites avec des appels API. Problème : les appels ont été faits pour une seule coordonnée (celle du centre de chaque case) donc il y a une probabilité que l'albedo ne corresponde pas la valeur de cette case. On a compté ici sur la dilution statistique du fait du grand nombre de cases.
- Implémentation d'une fonction, qui permettrait, dans la console, d'afficher la température locale d'un point en fonction de ses coordonnées (fait dans la version 1).
- Interface utilisateur pas fluide. Beaucoup de calculs sont refaits à chaque update. Optimisation du code est nécessaire pour avoir un rendu plus agréable.
- Curseur pour faire varier le temps : pas une bonne idée car pas très fluide, remplacer par simplement une commande dans la console pour un eventlistener sur une touche du clavier pour avancer dans le temps avec un pas défini (h dans notre cas)
- Echelle de température ne prend pas les valeurs minimum et maximum de la température en argument. Lorsque la modélisation sera améliorée, que le minimum ne sera pas de 0K, l'échelle ne correspondra plus.
- Effet de serre est calculé de manière statistique. On part du résultat (température moyenne sur la terre) pour calculer les effets de serre locaux = c'est pas une modélisation physique fiable. On pourra faire le même raisonnement global, avec un bilan énergétique qui a été fait, prenant l'atmosphère comme un corps noir également, mais à une échelle locale pour améliorer notre modèle.
- Des équations différentielles peuvent facilement modéliser la condition dans le sol de la température. Nous savons que d'autres groupes ont avancé sur ce plan, or nous ne l'avons absolument pas traité.
- Changement des mois avec les boutons sur le plot, ce n'est pas l'idée. On pourra penser à une solution plus pratique et esthétique, type menu déroulant.
- La chaleur dégagée par le noyau de la terre en fusion n'a pas dutout été pris en compte : on peut imaginer que c'est une partie non négligeable de l'nergie terrestre à la surface.

Explication des modèles:

Modèle 1:

On a considéré la Terre comme un point et comme un corps noir (100% de la puissance reçue est absorbée : formule de Stephan-Boltzmann)

$P = \sigma \cdot T^4$ où T est la température de surface en kelvin (K) $\rightarrow T = (P/\sigma)^{1/4}$

La puissance solaire moyenne reçue au sol est d'environ 342 W par m^2 de surface terrestre notée P :

AN : $T = 278.75K = 5.5^\circ C$

Modèle 2: Prise en compte de l'albédo ($a=0.3$).

Avec cette formule : $T = ((1-a) \cdot P/\sigma)^{1/4}$ issu d'un rapport de stage de thèse d'un étudiant de l'ENS de Lyon. Avec cette modélisation, on a obtenu une température de surface uniforme de $-18^\circ C$.

Modèle 3 : On a eu l'idée de séparer la Terre en un grand nombre de parcelles afin de leur attribuer une valeur d'albédo locale. On a trouvé ces données dans l'API de la NASA

(https://power.larc.nasa.gov/api/temporal/daily/point?parameters=ALLSKY_SFC_SW_DWN,ALLSKY_SFC_SW_UP&community=RE&longitude={lon}&latitude={lat}&start={start_date}&end={end_date}&format=JSON), cela donne une bonne idée de l'albédo sur une petite surface mais reste une approximation.

Modèle 4 : Ajout de l'angle incident qui diffère

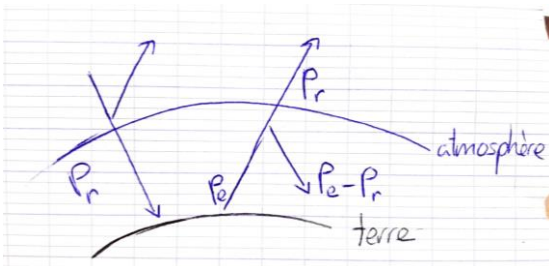
On a également voulu prendre en compte l'angle incident des rayons du Soleil sur la Terre en fonction de la localisation du Soleil et du moment dans l'année. Prise en compte de l'angle d'incidence des rayons du soleil.

On a calculé en fonction de l'inclinaison saisonnière de l'axe terrestre et de la rotation de la Terre sur son axe en fonction du mois. Cela a permis de modéliser les saisons et l'alternance jour/nuit.

Modèle 5 :

Ajout de la fonction effet_de_serre:

C'est une fonction qui prend en entrée la puissance reçue en un point (et les concentrations atmosphériques de CO_2 et d' H_2O), et qui renvoie la puissance à la surface de la terre en prenant en compte l'effet de serre.



Nous nous sommes basés sur un équilibre thermique, la puissance reçue P_r est égale à la puissance sortant de l'atmosphère (P_r est la puissance reçue si l'on prend en compte que les rayons renvoyés par l'albédo sont renvoyés avant même d'entrer dans l'atmosphère). Nous savons que la terre émet un rayonnement de puissance

P_e . Ce rayonnement est en partie renvoyé par les gaz à effet de serre, et en partie sort de l'atmosphère. (cf. schéma).

Nous cherchons donc ici à trouver le coefficient $C = \frac{P_e - P_r}{P_r}$.

Nous avons commencé par un calcul très simple d'un coef moyen, coef_moy, qui correspond à une température au sol de 15°C.

Nous calculons ensuite le coefficient en modifiant ce coef_moy. Nous avons écrit l'équation : $\text{coef} = 0.25 \cdot \text{coef_moy} + 0.25 \cdot \text{coef_moy} \cdot (\text{C_CO2} / \text{C_CO2_moy}) + 0.5 \cdot \text{coef_moy} \cdot (\text{C_H2O} / \text{C_H2O_moy})$. Ici, le 0.25 et 0.5 correspondent à un coefficient qui pondère l'effet des différents GES sur l'effet de serre. Nous avons multiplié par $\text{C_CO2} / \text{C_CO2_moy}$ et $\text{C_H2O} / \text{C_H2O_moy}$ pour qu'aux concentration moyenne, on ait le coefficient moyen. On a bien une température qui augmente lorsque la concentration en GES augmente.

Cependant, la formule ne se basant sur aucune preuve scientifique, nous avons modifié petit à petit le coefficient à la puissance de $\text{C_CO2} / \text{C_CO2_moy}$ pour nous rapprocher au mieux du tableau de données ci-dessous. Nous avons donc validé cette formule non pas avec des preuves scientifiques, mais parce qu'elle fonctionne.

Les valeurs utilisées sont :

Augmentation de la température (°C)	Concentration de CO2 (ppm)
0	316
0.1	325
0.3	337
0.6	353
0.8	369
1.0	388
1.1	412

Sources du tableau :

Valeurs de température :

https://www.ncei.noaa.gov/access/monitoring/climate-at-a-glance/global/time-series/globe/land_ocean/1/1/1960-2024?filter=true&filterType=running-mean

(avec : Region = global, surface = land and ocean, smoothed time series = 5-Years mean)

Valeurs de concentration de CO2 :

<https://www.climate.gov/news-features/understanding-climate/climate-change-atmospheric-carbon-dioxide> , et ipcc : chapter-2/figure-2-5.