Physique

Capacité numérique 5 – Variations de température et de pression dans l'atmosphère

Pour le mercredi 19 juin 10:00

Objectifs

- À l'aide d'un langage de programmation, étudier les variations de température et de pression dans l'atmosphère.
- Utiliser la fonction `odeint` de la bibliothèque `scipy.integrate`.

Le but de ce sujet est d'utiliser l'outil numérique afin de déterminer température et pression en fonction de l'altitude dans des modélisations plus fines que celle de l'atmosphère isotherme vue en cours.

Dans le cas isotherme, $P(z) = P_0 \exp\left(-\frac{Mg}{RT_0} z\right)$. Sur Terre, la température de surface moyenne est de $T_0 = 15\,°\mathrm{C}$[1].

1. Tracer, dans une image « `fig1.png` », un graphique double :
— à gauche, la température en °C ;
— à droite, la pression en bar.
L'altitude sera prise entre 0 et 29 km, par pas de 10 m.
Afin de rendre ces tracés plus intuitifs, l'axe des ordonnées est *communs aux deux sous-graphiques* et correspond à l'altitude en km. Le code permettant de réaliser les deux graphiques côte à côte avec ordonnée commune est :

```python
# création de la figure
figure1 = plt.figure()

# utilisation de sous-graphiques
graphT = figure1.add_subplot(1, 2, 1)
graphP = figure1.add_subplot(
    1, 2, 2,
    sharey = graphT, # partage des ordonnées
)
```

Pour tracer sur un des graphiques, utiliser par exemple `graphT.plot` au lieu de `plt.plot`.

Une autre modélisation possible est celle dite de l'atmosphère adiabatique.

2. Rappeler la loi de Laplace pour les variables P et V. En déduire l'équivalent pour les variables P et T en supposant que l'atmosphère peut être modélisée par un gaz parfait diatomique. Obtenir alors T en fonction de P, P_0 et T_0.

3. Déterminer aussi l'équivalent pour les variables P et ρ la masse volumique de l'atmosphère. Obtenir alors $\frac{\mathrm{d}\rho}{\mathrm{d}P}$ en fonction de ρ, P et γ.

4. Retrouver l'expression de ρ_0 la masse volumique de l'air à $z = 0$ et faire l'application numérique.

5. À partir de l'équation fondamentale de la statique des fluides, obtenir une équation différentielle sur P faisant intervenir $\rho(z)$.

6. En déduire $\frac{\mathrm{d}\rho}{\mathrm{d}z}$.

1. Du moins, tant que le dérèglement climatique ne change pas la donne...

La bibliothèque `scipy.integrate` comporte une fonction `odeint` dont la documentation complète est disponible en ligne[2]. Pour un système tel que $\frac{\mathrm{d}X}{\mathrm{d}t} = F(X)$ avec X pouvant être un vecteur, son utilisation peut être simplifiée, avec les lignes nécessaires en amont, en

```python
from scipy.integrate import odeint # import de la fonction
X0 = (theta_init, omega_init) # le vecteur X à l'instant initial, comme avant
temps = ... # la liste des instants t auxquels obtenir X(t)
X_liste = odeint(F, X0, temps)
```

avec `F` une fonction qui prend deux paramètres, `X` et `temps`.

7. L'idée est ici de remplacer la variable temporelle par l'altitude. Alors, `odeint` permet de résoudre les équations différentielles sur ρ et P précédemment obtenues.
Tracer ainsi, dans une image « `fig2.png` », l'équivalent de `fig1.png` mais dans le cas adiabatique en utilisant `odeint`.

8. Pourquoi ne pas aller jusqu'à 30 km d'altitude avec ce modèle ?

Les gradients de température sont en réalité connus pour l'atmosphère et consignés dans le tableau 1.

Couche	Altitude de début (km)	Gradient thermique ($\mathrm{K{\cdot}km^{-1}}$)
Troposphère	0	−6,5
Tropopause	11	0
Stratosphère	20	+1,0
Stratosphère	32	+2,8
Stratopause	47	0
Mésosphère	51	−2,8
Mésosphère	71	−2,0
Mésopause	85	0
Thermosphère	90	+3,3

Tableau 1 – *Données atmosphériques.*

9. Écrire le code définissant la fonction `T_atm`, prenant comme paramètres la valeur `z` de l'altitude en mètres et qui rend la température à cette altitude en kelvins en se basant sur les données du tableau 1.

10. En montant significativement en altitude, le champ de pesanteur g ne peut plus être considéré comme constant. Montrer qu'en réalité

$$g(z) = g_0 \frac{R_\mathrm{T}^2}{(R_\mathrm{T} + z)^2}$$

avec $g_0 = 9{,}81\,\mathrm{m{\cdot}s^{-2}}$, z l'altitude et $R_\mathrm{T} = 6371\,\mathrm{km}$ le rayon de la Terre.

11. Cet effet devra être pris en compte dans la suite de la simulation. Écrire le code définissant la fonction `g_Terre`, prenant comme paramètres la valeur `z` de l'altitude en mètres et qui rend la valeur de g à l'altitude z.

12. Exprimer ρ en fonction de P, T, M et R.

13. En déduire le moyen d'obtenir le champ de pression dans l'atmosphère en utilisant `odeint`. Tracer comme précédemment T et P de 0 à 29 km par pas de 10 m dans « `fig3.png` ».

14. Dans « `fig4.png` », de 0 à 120 km par pas de 10 m, comparer les températures et pression obtenues dans les modèles isotherme (et g constant) et celui avec le gradient de température (et g variable). Afin de faciliter la lecture des valeurs de pression, les afficher en échelle logarithmique. Commenter.

2. https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.odeint.html