

Calculs de distances entre deux points du globe par Python

Compétences **travaillées** [REA] Programmer sous Python

Comment utiliser un logiciel de programmation pour calculer des distances à la surface de la Terre à partir des coordonnées en latitude et en longitude ?

Document 1 : La géodésique

La géodésique est la trajectoire correspondant à la **distance minimale entre deux points** sur une surface. Dans le cas de la sphère, c'est un **arc de grand cercle**.

Si l'on considère deux points A et B sur la sphère, de latitudes ϕ_A et ϕ_B et de longitudes λ_A et λ_B , alors $S_{(A-B)}$ est la distance angulaire en radians entre A et B.

Les latitudes ϕ_A et ϕ_B et les longitudes λ_A et λ_B sont exprimés en degrés décimaux, $(\lambda_A - \lambda_B)$ étant la différence entre les deux longitudes.

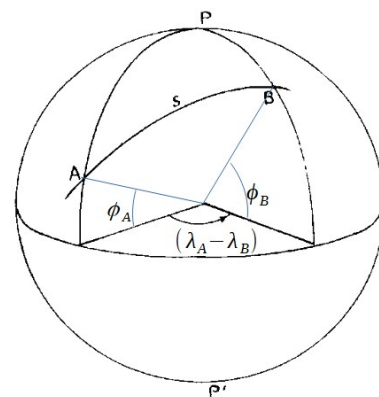


Figure 1: D'après : <https://geodesie.ign.fr> > contenu > fichiers > Distance_longitude_latitude

Document 2 : Principe de calcul par la méthode de Pythagore

Pour de petites distances à la surface de la Terre, la méthode utilisant le théorème de Pythagore donne des résultats corrects.

On calcule : $x = (\lambda_B - \lambda_A) \times \cos\left(\frac{\phi_A + \phi_B}{2}\right)$ et $y = \phi_B - \phi_A$ pour en déduire $z = \sqrt{x^2 + y^2}$.

La distance $S_{(A-B)}$ est alors définie par : $S_{(A-B)} = 1852 \times 60 \times z$, exprimé en mètres.

Document 3 : Principe de calcul par la méthode des sinus

Pour des distances plus grandes, la méthode des sinus donne des résultats plus justes.

La distance $S_{(A-B)}$ entre A et B est donnée par la relation :

$$S_{(A-B)} = 6378137 \times \arccos(\sin \phi_A \sin \phi_B + \cos \phi_A \cos \phi_B \cos(\lambda_B - \lambda_A))$$

Le calcul utilise la valeur d'un rayon conventionnel de la Terre : 6 378 137 mètres.

D'après : <https://geodesie.ign.fr> > contenu > fichiers > Distance_longitude_latitude

Document 4 : Lexique mathématiques Python :

La valeur de π en python se note " *math.pi* ".

fonctions & opérations	+	-	x	/	cosinus	sinus	racine	arccos	arcsin
programme Python	+	-	*	/	math.cos	math.sin	math.sqrt	math.acos	math.asin

Affichage d'une valeur :

Arrondir une valeur X à une valeur entière : round (X)

Arrondir à un chiffre après la virgule : round (X,1)

Arrondir à trois chiffres après la virgule : round (X,3)

Travail à réaliser

Paramétrages des données :

1. D'après les documents, quelles sont les grandeurs utilisées pour le calcul de la distance entre deux lieux situés sur la surface du globe terrestre? Quelles sont leurs unités?
2. Dans la première partie (lignes 3 à 7), on demande à l'utilisateur de rentrer les valeurs en longitude et latitude des deux lieux étudiés. Compléter les lignes 6 et 7 pour la demande de ces valeurs.

Calcul par la méthode utilisant Pythagore :

3. A l'aide du lexique des formules mathématiques utilisées sous Python, compléter les lignes de calculs 18, 19, 20 et 22. Paramétrer pour un affichage de la valeur de S en valeur entière et exprimée en km.

On donne les latitudes et longitudes de Nantes et Montaigu :

Nantes : longitude : $\lambda_A = -1,553621^\circ$ latitude : $\phi_A = 47,218371^\circ$
 Montaigu : longitude : $\lambda_B = -1,316667^\circ$ latitude : $\phi_B = 46,983333^\circ$

4. La distance donnée par un calculateur en ligne est S=32,5 km. Vérifier la validité de votre programme. Le corriger si besoin.



appel 1

Appeler le professeur pour la vérification de votre programme, ou, en cas de difficulté.

5. On donne les latitudes et longitudes de Paris et Sydney :

Paris : longitude : $\lambda_A = 2,3522219^\circ$ latitude : $\phi_A = 48,856614^\circ$
 Sydney : longitude : $\lambda_B = 151,206990^\circ$ latitude : $\phi_B = -33,867487^\circ$

La distance donnée par un calculateur en ligne est S=16 965,3 km. Calculer la distance entre Paris et Sydney avec votre programme. Comparer avec la valeur donnée ci-dessus. Conclure quant à la validité du calcul avec la méthode par Pythagore.

Calcul par la méthode des sinus :

6. Supprimer les " #" des lignes 26 et 27 pour pouvoir utiliser les lignes de calculs par la méthode des sinus.
7. Relancer le programme pour déterminer la distance Paris – Sydney avec la méthode de Pythagore et la méthode des sinus. Conclure quant à la fiabilité des calculs effectués par ces différentes méthodes.
8. Maintenant, amusez-vous à voyager !!! Rechercher des coordonnées de villes dans le monde et déterminer leur distance à partir de Montaigu.

Lexique mathématiques Python :

- La valeur de π en python se note " *math.pi* " .

fonctions & opérations	+	-	x	/	cosinus	sinus	racine	arccos	arcsin
programme Python	+	-	*	/	math.cos	math.sin	math.sqrt	math.acos	math.asin

Affichage d'une valeur :

Arrondir une valeur X à une valeur entière : round (X)

Arrondir à un chiffre après la virgule : round (X,1)

Arrondir à trois chiffres après la virgule : round (X,3)

Le programme incomplet

Le programme à compléter est donné sur le serveur du lycée.

Avec Python, les fonctions trigonométriques sont à utiliser avec des grandeurs en radians, en dehors des fonctions trigonométriques, les grandeurs à utiliser sont exprimées en degrés.

Voici les lignes de code :

```
1. import math
2.
3. def PGDistance1():
4.     lambda1 = float(input("Longitude en décimale du premier lieu ?"))
5.     phi1 = float(input("Latitude en décimale du premier lieu ?"))
6.     ..... = float(input("Longitude en décimale du deuxieme lieu ?"))
7.     ..... = float(input("Latitude en décimale du deuxieme lieu ?"))
8.
9.     #conversion des angles en radians
10.    phi1r = float (math.pi/180* phi1)
11.    phi2r = float (math.pi/180* phi2 )
12.    lambda1r = .....
13.    lambda2r = .....
14.
15.    #Calcul de la distance par la méthode de Pythagore
16.    #On calcule x et y des distances exprimées en degrés (degrés décimaux).
17.    #Puis, z la distance cherchée exprimée en "degrés". z = racine (x2+y2)
18.    x = .....
19.    y = .....
20.    z = .....
21.    # z est l'arc de cercle en degrés. La distance S en m se calcule par S = 60*1852*z.
22.    S= .....
23.    print("La distance selon Pythagore est",distP,"km")
24.
25.    #Calcul de la distance selon la loi des sinus :
26.    #S=6378.137 * math.acos(math.sin(phi1r) * math.sin(phi2r) + math.cos(phi1r) * math.cos(phi2r) *
    math.cos(lambda2r-lambda1r))
27.    # print ("La distance selon la loi des sinus est",round (S),"km")
28.
29. PGDistance1()
```

Correction :

```
1. import math
2.
3. def PGDistance1():
4.     lambda1 = float(input("Longitude en décimale du premier lieu ?"))
5.     phi1 = float(input("Latitude en décimale du premier lieu ?"))
6.     lambda2 = float(input("Longitude en décimale du deuxieme lieu ?"))
7.     phi2 = float(input("Latitude en décimale du deuxieme lieu ?"))
8.
9.     #conversion des angles en radians
10.    phi1r = float (math.pi/180* phi1)
11.    phi2r = float (math.pi/180* phi2 )
12.    lambda1r = float (math.pi/180*lambda1 )
13.    lambda2r = float (math.pi/180*lambda2)
14.
15.    #Calcul de la distance par la méthode de Pythagore
16.    #On calcule x et y des distances exprimées en degrés (degrés décimaux).
17.    #Puis, z la distance cherchée exprimée en "degrés".
18.    x = (lambda2-lambda1)*math.cos((phi1r + phi2r)/2)
19.    y = phi2-phi1
20.    z = math.sqrt(x*x + y*y)
21.    # z est l'arc de cercle en degrés. La distance S en m se calcule par S = 60*1852*z.
```

```
22. S=60*1.852*z
23. print("La distance selon Pythagore est",round (S),"km")
24.
25. #Calcul de la distance selon la loi des sinus :
26. S = 6378.137 * math.acos(math.sin(phi1r) * math.sin(phi2r) + math.cos(phi1r) * math.cos(phi2r) *
    math.cos(lambda2r-lambda1r))
27. print ("La distance selon la loi des sinus est",round (S),"km")
28.
29. PGDistance1()
```