

# Défi - Thème: Cartographie et Localisation

## Objectif

Approfondir la mécanique céleste. Créer un tableau de valeur avec python représentant les vitesses et les périodes de rotation d'un satellite en fonction de son altitude.

### [1 Présentation de la mécanique céleste](#)

#### [1.1 Le jeu](#)

#### [1.2 Je comprends...](#)

##### [1.2.1 Question sur la vitesse et le rayon](#)

##### [1.2.2 Question sur le rayon et la période](#)

### [2 Approche Python](#)

#### [2.1 Les constantes](#)

#### [2.2 Les définitions de fonctions](#)

##### [2.2.1 Cas de la vitesse](#)

##### [2.2.2 Programme en python](#)

##### [2.2.3 Tester ce programme et répondez aux questions](#)

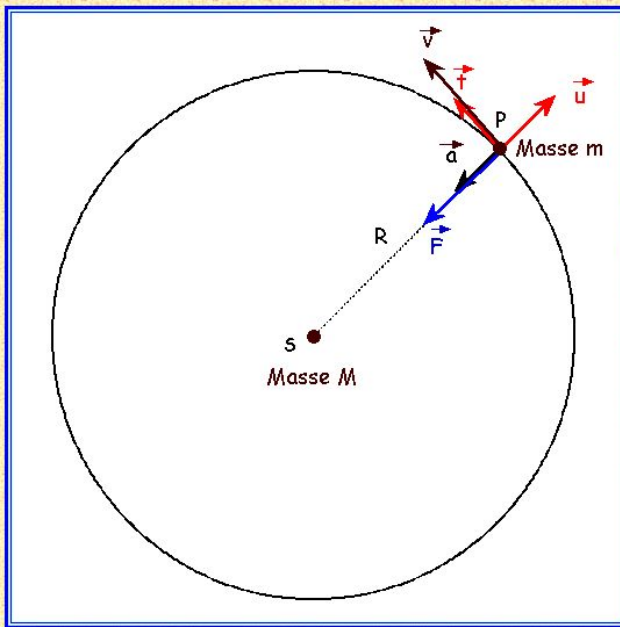
##### [2.2.4 Les fonctions sous python](#)

###### [2.2.4.1 Intérêt](#)

###### [2.2.4.2 Application directe](#)

###### [2.2.4.3 Définition de la fonction période](#)

### [3 Mise en application - recherche](#)



- $SP = R$ , rayon de l'orbite circulaire est, dans ce cas, le demi grand axe. (On l'appelle parfois le rayon vecteur).
- $\vec{u}$  est un vecteur unitaire centripète.
- $\vec{t}$  est un vecteur unitaire tangentiel, de sens celui du mouvement.
- Le repère d'origine P et de vecteurs de base  $\vec{u}$  et  $\vec{t}$  est appelé repère de Frenet.

Ce n'est pas un repère galiléen, mais une fois qu'on a appliqué la deuxième loi de Newton dans un repère galiléen (héliocentrique, non représenté sur le dessin) on peut utiliser le repère de Frenet pour projeter cette 2<sup>ème</sup> loi de Newton.

- Nous avons vu les *relations suivantes*:

- (1)  $\vec{v} = v \vec{t}$     (2)  $\vec{a} = -\frac{v^2}{R} \vec{u}$     (3)  $\vec{F} = -\frac{GMm}{R^2} \vec{u}$     (4)  $\vec{F} = m \vec{a}$
- (3) et (4)  $\Rightarrow \vec{a} = -\frac{GM}{R^2} \vec{u}$  et avec (2),  $-\frac{v^2}{R} \vec{u} = -\frac{GM}{R^2} \vec{u} \Rightarrow v^2 = \frac{GM}{R}$  (5)
- De plus  $v = \frac{\text{circonférence}}{\text{période}} \Rightarrow v = \frac{2\pi R}{T} \Rightarrow v^2 = \frac{4\pi^2 R^2}{T^2}$  (6)
- (5) et (6)  $\Rightarrow \frac{GM}{R} = \frac{4\pi^2 R^2}{T^2} \Rightarrow (7) \frac{R^3}{T^2} = \frac{GM}{4\pi^2} = \text{constante}$

Et appliquez les relations (5) et (7) de cette image

On donne pour la suite les données suivantes compatible python

# constantes

G=6.67\*10\*\*-11 # constante universelle

M= 5.97\*10\*\*24 # Masse de la terre en kg

Rt=6378\*10\*\*3 # Rayon terrestre en m

## 1 Présentation de la mécanique céleste

### 1.1 Le jeu

Le plus simple c'est de se rendre là:

[https://www.sciences.univ-nantes.fr/sites/genevieve\\_tulloue/Meca/Planetes/transfert.html](https://www.sciences.univ-nantes.fr/sites/genevieve_tulloue/Meca/Planetes/transfert.html)

Pour vous aidez à jouer, voir la vidéo du dessous

<https://youtu.be/jyHEub7wc4Q>

Spationaute, avez-vous réussi votre changement d'orbite?



### 1.2 Je comprends...

La relation 5

$$v^2 = \frac{GM}{R} \quad (5)$$

G étant constant et M représentant la masse de la terre, alors v varie en fonction de R qui correspond au rayon terrestre et à mon altitude!!!!

La relation 7

$$(7) \frac{R^3}{T^2} = \frac{GM}{4\pi^2} = \text{constante}$$

#### 1.2.1 Question sur la vitesse et le rayon

Si  $R_1 < R_2$  alors

> comparer  $V_1$  ???  $V_2$

#### 1.2.2 Question sur le rayon et la période

Si  $R_1 < R_2$  alors

> comparer  $T_1$  ???  $T_2$

## 2 Approche Python

### 2.1 Les constantes

En dessous se trouve 4 lignes que vous pouvez utiliser sous python.

```
# constantes
G=6.67*10**-11 # constante universelle
Mt= 5.97*10**24 # Masse de la terre en kg
Rt=6378*10**3 # Rayon terrestre en m
```

Exprimez les valeurs de Mt et Rt du dessus comme dans votre livre de physique. Pour vous aider je dirais que

G = constante de gravitation universelle

G =  $6,67 \cdot 10^{-11} \text{ N.kg}^{-2}.\text{m}^2$



## Thème n°3 - Défi - Cartographie et Localisation

---

Que représente les \*\* sous python dans l'expression 10\*\*-11 ?



### 2.2 Les définitions de fonctions

#### 2.2.1 Cas de la vitesse

La relation (5) est donc

$$v = \sqrt{\frac{G * Mt}{Rt + h}}$$

avec

G=6.67\*10\*\*-11 # constante universelle

Mt= 5.97\*10\*\*24 # Masse de la terre en kg

Rt=6378\*10\*\*3 # Rayon terrestre en m

#### 2.2.2 Programme en python

```
# debut
```

```
# import de la fonction racine carree  
from math import sqrt
```

```
# constantes
```

```
G=6.67*10**-11 # constante universelle
```

```
Mt= 5.97*10**24 # Masse de la terre en kg
```

```
Rt=6378*10**3 # Rayon terrestre en m
```

```
# declaration de la fonction vitesse de paramètre h
```

```
def vitesse(h):
```

```
    v=sqrt(G*Mt/(Rt+h))
```

```
    return v
```

```
# test
```

```
print (vitesse(0),"--",vitesse(10000),"--",vitesse(100000))
```

```
# fin
```

#### 2.2.3 Tester ce programme et répondez aux questions

A supposer que la terre soit parfaitement sphérique, qu'il n'y a aucune montagne et que cette planète ne dispose pas d'atmosphère, alors on pourrait satelliser un objet au raz du sol. Quelle vitesse aurait-il en m/s à 0m et à 100km?

# Thème n°3 - Défi - Cartographie et Localisation

> vitesse à une altitude de 0m

> vitesse à 100 km

Transformer cette valeur en km/h!



## 2.2.4 Les fonctions sous python

### 2.2.4.1 Intérêt

Source <https://courspython.com/fonctions.html#>

“Lorsqu’une tâche doit être réalisée plusieurs fois par un programme avec seulement des paramètres différents, on peut l’isoler au sein d’une fonction. Cette approche est également intéressante si la personne qui définit la fonction est différente de celle qui l’utilise.”

### 2.2.4.2 Application directe

A partir du moment où vous avez lancé une fois le programme, python reconnaît une nouvelle variable de nom “vitesse” qui fonctionne avec un paramètre “h” numérique. C’est pour cela que dans le shell je peux, après avoir lancé au moins une fois le script, appeler de nouveau cette fonction et lui demander un autre calcul avec 15000!!

```
qq.py
1 # import de la fonction racine carree
2 from math import sqrt
3
4 # constantes
5 G=6.67*10**-11 # constante universelle
6 Mt= 5.97*10**24 # Masse de la terre en kg
7 Rt=6378*10**3 # Rayon terrestre en m
8
9 # declaration de la fonction vitesse de parametre h
10 def vitesse(h):
11     v=sqrt(G*Mt/(Rt+h))
12     return v
13
14 # test
15 print (vitesse(0), "--",vitesse(10000), "--",vitesse(100000))
16
17

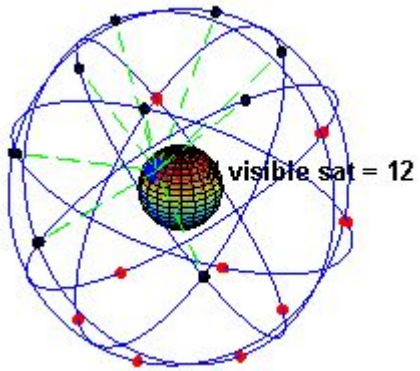
Shell
>>> %Run qq.py
7901.468718039937 -- 7895.201677142053 -- 7840.244575555992
>>> vitesse(15000)
7892.193601897748
>>> |
```

Name	Value
G	6.67e-11
Mt	5.969999999999999e+24
Rt	6378000
sqrt	<built-in function sqrt>
vitesse	<function vitesse at 0x03A25540>

## Thème n°3 - Défi - Cartographie et Localisation

---

La constellation de satellites GPS se trouvent à une altitude de 20200km. Calculez la vitesse de ces satellites en m/s puis en km/h.



> vitesse en m/s

> vitesse en km/h

# Thème n°3 - Défi - Cartographie et Localisation

## 2.2.4.3 Définition de la fonction période

$$(7) \frac{R^3}{T^2} = \frac{GM}{4\pi^2} = \text{constante}$$

Exprimez T en fonction de G, Mt, Rt, et pi (la fonction pi sera à appeler de la bibliothèque math) puis déclarer la fonction periode(h)

```
# import de la fonction racine carree et pi
from math import sqrt,pi
```

```
# constantes
```

```
G=6.67*10**-11 # constante universelle
```

```
Mt= 5.97*10**24 # Masse de la terre en kg
```

```
Rt=6378*10**3 # Rayon terrestre en m
```

```
# declaration de la fonction vitesse de
paramètre h
```

```
def vitesse(h):
```

```
    v=sqrt(G*Mt/(Rt+h))
```

```
    return v
```

```
# declaration de la fonction periode de
paramètre h
```

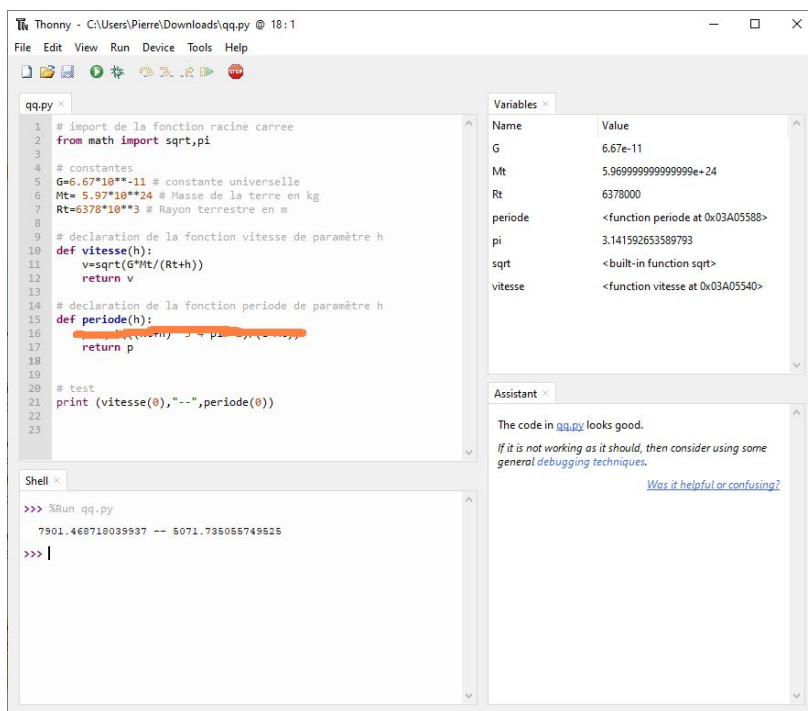
```
def periode(h):
```

```
    p=sqrt(((?+?)**?*4**2)/(?*Mt))
```

```
    return p
```

>Remplacer les ? par des constantes

```
p=sqrt(((?+?)**?*4**2)/(?*Mt))
```



```
Thonny - C:\Users\Pierre\Downloads\qq.py @ 18:1
File Edit View Run Device Tools Help

qq.py x
1 # import de la fonction racine carree
2 from math import sqrt,pi
3
4 # constantes
5 G=6.67*10**-11 # constante universelle
6 Mt= 5.97*10**24 # Masse de la terre en kg
7 Rt=6378*10**3 # Rayon terrestre en m
8
9 # declaration de la fonction vitesse de paramètre h
10 def vitesse(h):
11     v=sqrt(G*Mt/(Rt+h))
12     return v
13
14 # declaration de la fonction periode de paramètre h
15 def periode(h):
16     p=sqrt(((?+?)**?*4**2)/(?*Mt))
17     return p
18
19
20 # test
21 print (vitesse(0), "--", periode(0))
22
23

Variables x
Name Value
G 6.67e-11
Mt 5.969999999999999e+24
Rt 6378000
periode <function periode at 0x03A05588>
pi 3.141592653589793
sqrt <built-in function sqrt>
vitesse <function vitesse at 0x03A05540>

Assistant x
The code in qq.py looks good.
If it is not working as it should, then consider using some
general debugging techniques.
Was it helpful or confusing?

Shell x
>>> %Run qq.py
7901.468718039937 -- 5071.738066749626
>>> |
```

## 3 Mise en application - recherche

Votre mission avant décollage!

Afin de palier à tout imprévu, vous désirez avoir à disposition un tableau contenant :

<altitude en km> <vitesse en km/h> <periode en min>

pour des altitudes partant de h = 0km et atteignant h = 36000km tous les 10 km

On rappelle:

```
for i in range(1, 10, 2):
```

Pour i allant de 1 à 10 tous les 2

## Thème n°3 - Défi - Cartographie et Localisation

---

Pour vous permettre de vérifier vos calculs voici un exemple de mon shell

```
>>> %Run qq.py
< h en km > < vitesse en km par h > < période en min >
< 0.0 > < 28445.287384943775 > < 84.5289175958254 >
< 10.0 > < 28423.014037714274 > < 84.72779351761727 >
< 20.0 > < 28400.792930386833 > < 84.9268251640764 >
< 30.0 > < 28378.62385907448 > < 85.1260124134572 >
< 40.0 > < 28356.506621002554 > < 85.32535514429922 >
...
...
...
< 35960.0 > < 11040.46623797019 > < 1445.686226296239 >
< 35970.0 > < 11039.162620551993 > < 1446.1984511020978 >
< 35980.0 > < 11037.859464803742 > < 1446.7107363895673 >
< 35990.0 > < 11036.556770453004 > < 1447.2230821515075 >
< 36000.0 > < 11035.254537227569 > < 1447.7354883807816 >
>>>
```

**Publiez votre script dans votre blog  
pour 10 points!!!**

Votre article contiendra :

- le script >>(/4)
- l'explication du script >>(/4)
- plusieurs liens vers des sites présentant le fonctionnement de la localisation par satellite >>(/2)