La Sation Spatiale Internationale: mythe ou réalité?

1 La trajectoire de l'ISS

A partir du site de l'ESA (https://www.esa.int/Science_Exploration/Human_and_Robotic_Exploration/International_Space_Station/Where_is_the_International_Space_Station), oberver la représentation de la trajectoire de l'ISS autour de la Terre

1) Quel désaccord semble-t-il y avoir entre la trajectoire de l'ISS et la première loi de Kepler ? Émettre une hypothèse explicative.

L'ISS semble ne pas avoir une trajectoire fermée. Cette observation est probablement due au fait qu'on observe ici le mouvement de l'ISS par rapport au sol et non dans le référentiel géocentrique.

2) Dans le but de vérifier votre hypothèse, proposer un protocole basé sur le relevé des coordonnées de l'ISS.

On relève la position de l'ISS au cours du temps et on tient compte de la rotation de la Terre sur elle-même pour obtenir la trajectoire de la station spatiale dans le référentiel géocentrique.

2 Exploitation des données expérimentales

Relever les données de positionnement de l'ISS autour de la Terre toutes les minutes environ durant 2 heures et reporter ces données dans la feuille de calcul partagée (https://docs.google.com/spreadsheets/d/1_oQINFfnq2PHekRRWiy6PR1OYR2C9pEC9BNfY3aONm4/edit?usp=sharing).

A l'image d'une communauté scientifique, c'est à vous de nommer un directeur d'étude qui répartira les tâches et le travail de chacun.

3) A partir de votre feuille de calcul partagée, `Fichier - Télécharger - Microsoft Excel XLSX` . Déplacer ensuite le fichier `dataset_iss.xlsx` dans le répertoire Activite_ISS.

2.1 Lecture du fichier Excel

Dans un premier temps, nous allons simplement vérifier que le fichier Excel est lu correctement.

4) Exécuter pas à pas les morceaux de codes proposés dans les céllules ci-dessous.

In [1]:
```
import pandas as pd # La bibliothèquee pandas permet de lire et traiter un fichier Excel
dataframe=pd.read_excel('dataset_iss.xlsx')# récupère les données dans le tableau dataframe
head()#Affiche les premières lignes du tableau de données
```

Out[1]:

	Jour	Heure	Latitudes	Longitudes	Altitudes
0	2020-10-20	12:56:06	42.6S	178.3E	433
1	2020-10-20	12:57:11	40.2S	177.2W	432
2	2020-10-20	12:58:11	37.8S	173.5W	431
3	2020-10-20	12:59:12	35.4S	170.0W	430
4	2020-10-20	13:00:12	32.8S	166.7W	429

Le fichier Python `dataset_iss.py` va être importé. Il va se charger de transformer ces données de type texte, en nombres afin de pouvoir les exploiter dans nos calculs et graphiques.

Dans le code Python, les listes seront maintenant appelées :

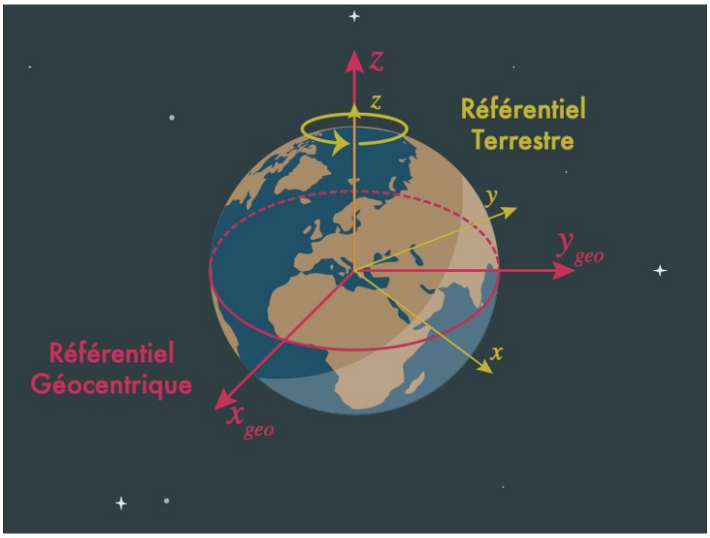
- Temps nommée `t`
- Latitude nommée `lat`
- Longitude nommée `long`
- Altitude nommée `alt`

In [2]:
```
from dataset_iss import * # permet d'accéder au ficher écrit en langage Python contenant les différentes procédures nécessaires au traitement des données
dataframe.head()# On affiche de nouveau le tableau de données une fois celui-ci traité par le fichier Python
```

Out[2]:

	Temps	Latitudes	Longitudes	Altitudes
0	0.0	-42.6	178.3	433
1	65.0	-40.2	-177.2	432
2	125.0	-37.8	-173.5	431
3	186.0	-35.4	-170.0	430
4	246.0	-32.8	-166.7	429

À partir des listes Latitudes, Longitudes et Altitudes le code calcule alors les coordonnées $\begin{cases} x(t) \\ y(t) \\ z(t) \end{cases}$ du vecteur position $\vec{OM}$, **dans le référentiel terrestre** $\mathcal{R}_T$.



Ces listes seront désormais nommées : `x, y, z` dans le code. Nous allons les afficher ci-dessous.

In [3]:
```
dataframe2 = pd.DataFrame(list(zip(t, x, y, z)), columns = ['$t$ (en s)','$x$ (en m)','$y$ (en m)','$z$ (en m)'])
dataframe2 head()  # On affiche le tableau des coordonées dans le référentiel terrestre
```

Out[3]:

	t (en s)	x (en m)	y (en m)	z (en m)
0	0.0	-5.006200e+06	1.485806e+05	-4.605464e+06
1	65.0	-5.189901e+06	-2.538285e+05	-4.391049e+06
2	125.0	-5.340085e+06	-6.084259e+05	-4.168994e+06
3	186.0	-5.459463e+06	-9.626506e+05	-3.939691e+06
4	246.0	-5.562547e+06	-1.314930e+06	-3.683616e+06

Le code calcule également les coordonnées $\begin{cases} x_{geo}(t) \\ y_{geo}(t) \\ z(t) \end{cases}$ **dans le référentiel géocentrique** $\mathcal{R}_G$ à partir de $\begin{cases} x(t) \\ y(t) \\ z(t) \end{cases}$.

Ces listes sont nommées `xgeo`, `ygeo`, `z` dans le code.

In [4]:
```
dataframe3 = pd.DataFrame(list(zip(t, xgeo, ygeo, z)), columns = ['$t$ (en s)','$x_{geo}$ (en m)',
'$y_{geo}$ (en m)', '$z$ (en m)'])
 dataframe3.head() # On affiche le tableau des coordonées dans le référentiel terrestre
```

Out[4]:

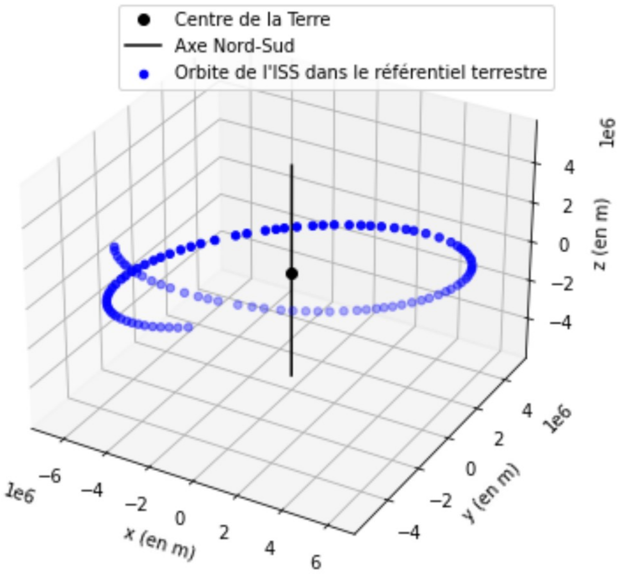
	t (en s)	x_{geo} (en m)	y_{geo} (en m)	z (en m)
0	0.0	-5.006200e+06	1.485806e+05	-4.605464e+06
1	65.0	-5.188640e+06	-2.784251e+05	-4.391049e+06
2	125.0	-5.334318e+06	-6.570756e+05	-4.168994e+06
3	186.0	-5.445905e+06	-1.036608e+06	-3.939691e+06
4	246.0	-5.538066e+06	-1.414498e+06	-3.683616e+06

2.2 Tracé de l'orbite dans les deux référentiels

Le graphe ci-dessous représente la trajectoire de l'ISS **dans le référentiel terrestre** $\mathcal{R}_T$

In [5]:
```
%matplotlib inline
#%matplotlib notebook
fig = plt.figure(1,figsize=(6,6))
graphe = fig.gca(projection='3d')
graphe.plot([0],[0],[0],'ok',label='Centre de la Terre')
graphe.plot([0,0],[0,0],[min(z),max(z)],'-k',label='Axe Nord-Sud')
graphe.scatter(x,y,z,color='blue', label="Orbite de l'ISS dans le référentiel terrestre")
graphe.legend(loc='upper right')
graphe.set_xlabel('x (en m)')
graphe.set_ylabel('y (en m)')
graphe.set_zlabel('z (en m)')
plt show()
```

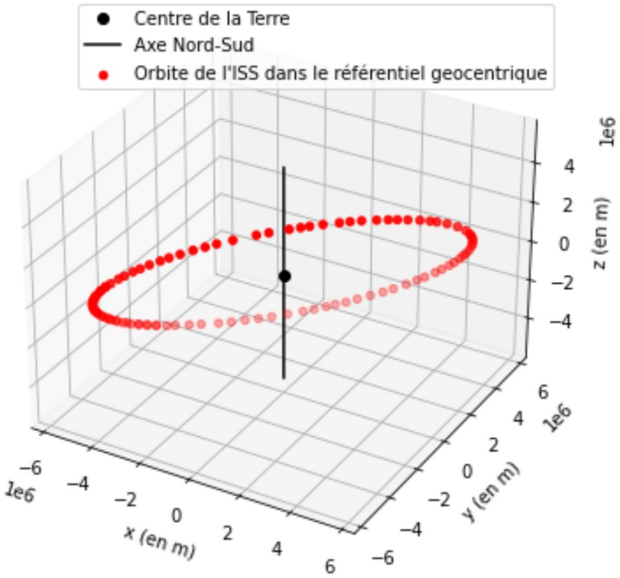
Out[5]:



5) En vous appuyant sur le code précédent, tracer la trajectoire de l'ISS **dans le référentiel geocentrique** $\mathcal{R}_G$.

In [6]:
```
%matplotlib inline
#%matplotlib notebook
fig = plt.figure(2,figsize=(6,6))
graphe = fig.gca(projection='3d')
graphe.plot([0],[0],[0],'ok',label='Centre de la Terre')
graphe.plot([0,0],[0,0],[min(z),max(z)],'-k',label='Axe Nord-Sud')
graphe.scatter(xgeo,ygeo,z,color='red', label="Orbite de l'ISS dans le référentiel geoce
graphe.legend(loc='upper right')
graphe.set_xlabel('x (en m)')
graphe.set_ylabel('y (en m)')
graphe.set_zlabel('z (en m)') p
lt show()
```

Out[6]:



6) Conclure quant à l'hypthèse émise en question 1

Dans le référentiel géocentrique, la trajectoire de l'ISS est bien fermée. Il semble raisonnable de considérer que l'orbite est circulaire.

3 Confrontation de la vitesse de l'ISS au modèle de Kepler

3.1 Quelques rappels

La deuxième loi de Newton permet d'établir dans le cas particulier d'une orbite circulaire les relations suivantes sur la période T et la vitesse v de l'ISS:

$$\frac{T^2}{a^3} = \frac{4\pi^2}{GM_T} \qquad v = \sqrt{\frac{GM_T}{r}}$$

Avec:

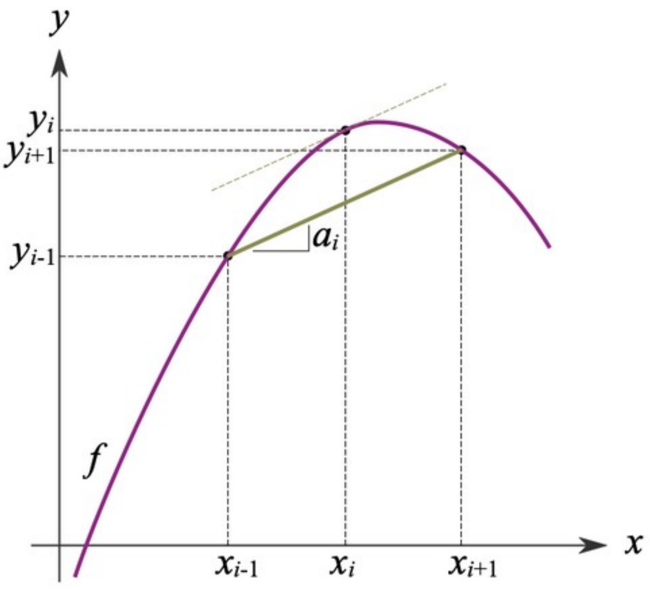
- $G = 6,67.10^{-11}$ SI
- $M_T = 5,97.10^{24}$ kg
- $r = R_T + h$ où $R_T = 6371.10^3$ m (Rayon moyen de la Terre) et h l'altitude de l'orbite.

On rappelle également quelques fonctionnalités statistiques permettant l'évaluation d'une incertitude-type:

- La moyenne $\bar{x}$: `xmoy=x.mean()`
- L'écart-type expérimental s_x: `sx=x.std(ddof=1)`
- Le nombre n d'éléments dans une liste: `n=len(liste)`
- La racine carrée $\sqrt{x}$: `np.sqrt(x)`

Enfin, pour dériver une fonction numérique (connue sous forme de valeurs discrètes et non sous forme algébrique), on peut utiliser le théorème des accroissements finis. La valeur de la dérivée de f au point d'abscisse x_i, noté $f(x_i)$, peut être approché par la pente de la corde aux points entourans x_i. On a alors la formule approchée:

$$f'(x_i) \approx \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}}$$



3.2 Calcul de la vitesse mesurée

7) Compléter le code ci-dessous afin de calculer les coordonnées $\begin{cases} v_x(t) \\ v_y(t) \\ v_z(t) \end{cases}$, du vecteur vitesse $\vec{v}$, dans **dans le référentiel géocentrique** $\mathcal{R}_G$.

In [7]:
```
vx,vy,vz=[],[],[]
for i in range(1,len(t)-1):
    vx.append((xgeo[i+1]-xgeo[i-1])/(t[i+1]-t[i-1]))
    vy.append((ygeo[i+1]-ygeo[i-1])/(t[i+1]-t[i-1]))
    vz.append((z[i+1]-z[i-1])/(t[i+1]-t[i-1]))
```

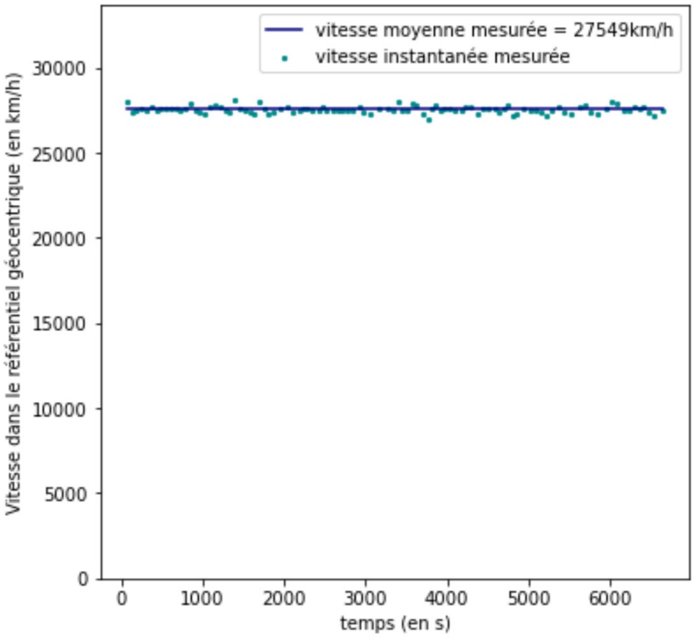
8) À partir de la liste `v` , calculée en $km \cdot h^{-1}$, évaluer la valeur mesurée de la vitesse, nommée `vmes` accompagnée de son incertitude-type `uvmes` .

In [8]:
```
v=3.6*np.sqrt(np.array(vx)**2+np.array(vy)**2+np.array(vz)**2)

vmes=v.mean()
uvmes=v.std(ddof=1)/np.sqrt(len(v))
print("v = ",vmes,"±",uvmes,"km/h")

%matplotlib notebook
%matplotlib inline
fig = plt.figure(2,figsize=(6,6))
plt.scatter(t[1:-1],v,color='darkcyan',s=5,label='vitesse instantanée mesurée')
plt.plot(t[1:-1],[vmes]*len(t[1:-1]),color='navy', label='vitesse moyenne mesurée = '+str(int(vmes))
+'km/h')
plt.ylim(0,1.2*max(v))
plt.xlabel("temps (en s)")
plt.ylabel('Vitesse dans le référentiel géocentrique (en km/h)')
plt.legend()
plt.show()
```

Out[8]: v = 27549.072077180062 ± 19.380372339652755 km/h



3.3 Comparaison à la valeur théorique du modèle de Kepler

9) Compléter le code suivant afin de calculer en $km \cdot h^{-1}$ la valeur théorique de la vitesse `vth` de l'ISS d'après le modèle de Kepler, puis comparer la valeur mesurée à cette valeur de référence $\frac{|v_{mes} - v_{th}|}{u_v}$.

In [9]:
```
#Compléter le code
h=alt.mean()
G=6.67e-11
MT=5.97e24
RT=6371e3

vth=3.6*np.sqrt(G*MT/(RT+h*1e3))
zscore=abs(vth-vmes)/uvmes

print("Vitesse théorique =",vth,"km/h")
print("Vitesse mesurée =",vmes,"±",uvmes,"km/h")
print("Écart ",zscore," fois la valeur de l'incertitude")
```

Out[9]:
```
Vitesse théorique = 27557.168033547 km/h
Vitesse mesuree = 27549.072077180062 ± 19.380372339652755 km/h
Écart  0.4177399806903575  fois la valeur de l'incertitude
```

La vitesse mesurée et la vitesse prévue par le modèle de Kepler sont donc compatibles.

4 Confrontation de la période de l'ISS au modèle de Kepler

10) Le bloc code ci-dessous, permet de tracer les coordonnées $\begin{cases} x_{geo}(t) \\ y_{geo}(t) \\ z(t) \end{cases}$,du vecteur position $\vec{OM}$ **dans le référentiel géocentrique** $\mathcal{R}_G$.

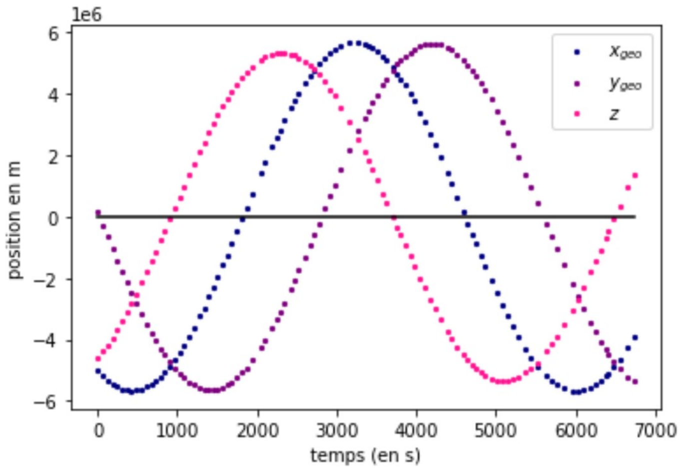
À partir de ce graphique proposer une démarche pour déterminer la valeur mesurée de la période de l'ISS nommée `Tmes` accompagnée de l'évaluation de type B de son incertitude-type `uTmes` . Enfin, comparer la valeur mesurée à la valeur de référence $\frac{|T_{mes} - T_{th}|}{u_T}$.

In [10]:

```
%matplotlib inline
#%matplotlib notebook

plt.figure(3)
plt.plot([0,max(t)],[0,0],'-k')
plt.scatter(t,xgeo,color='navy',s=5,label='$x_{geo}$')
plt.scatter(t,ygeo,color='purple',s=5,label='$y_{geo}$')
plt.scatter(t,z,color='deeppink',s=5,label='$z$')
plt.legend()
plt.xlabel('temps (en s)')
plt.ylabel('position en m')
plt show()
```

Out[10]:



In [11]:

```
tzZero=[]
for i in range(0,len(z)-1):
    if z[i+1]*z[i]<0:
        print("Passage à zéro entre ",t[i]," s  et ",t[i+1],"s")
        tzZero.append(t[i])
        tzZero.append(t[i+1])

Tmax=tzZero[5]-tzZero[0]
Tmin=tzZero[4]-tzZero[1]
Tmes=(Tmax+Tmin)/2
uTmes=(Tmax-Tmin)/2
uTmes=(Tmax-Tmin) #Rq: Cette différence peut être considérée comme une première approximation de
l'incertitude-type.
print("Periode mesurée : Tmes =",Tmes,"±",uTmes,"s")

Tth=np.sqrt(4*np.pi**2*(RT+h*1e3)**3/(G*MT))
print("Periode théorique : Tth =",Tth,"s")

zscore=abs(Tmes-Tth)/uTmes
print("Écart ",zscore," fois la valeur de l'incertitude")
```

Out[11]:

```
Passage à zéro entre  905.0  s  et  966.0 s
Passage à zéro entre  3699.0  s  et  3764.0 s
Passage à zéro entre  6479.0  s  et  6550.0 s
Periode mesurée : Tmes = 5579.0 ± 132.0 s
Periode théorique : Tth = 5578.06772026996 s
Écart  0.00706272522757679  fois la valeur de l'incertitude
```

4.1 Conclusion sur la confrontation entre l'orbite de l'ISS et le modéle de Kepler

Nous avons pu vérifier que l'orbite de l'ISS autour de la Terre peut être modélisée par une orbite circulaire. La vitesse de l'ISS dans le référentiel géocentrique ainsi que la période de révolution de la station spatiale internationale autour de la Terre sont en accord avec la troisième loi de Kepler. Le mouvement de l'ISS est donc en accord avec le modèle de Kepler.