

# SOLUTION DE POSITIONNEMENT STANDARD GPS BASÉE SUR PYTHON

Elaboré par :

L'HAMDOUCHI Driss N°35

Encadré par :

Professeur Moha EL-AYACHI

## Introduction :

Créer un programme python pour obtenir une solution de positionnement pour un récepteur GPS à l'aide de ses fichiers RINEX d'observation et de navigation. Cela permettra l'application des concepts appris dans les cours du fondement GNSS et l'infrastructure géodésique à travers un projet pratique qui allie théorie et pratique réelle. Cela apportera une plus grande clarté sur le fonctionnement du GPS et GNSS généralement.

## Objectives :

- Développer des fonctions Python pour traiter les fichiers d'observation et de navigation RINEX et stocker les données dans une structure de données.
- Développer une fonction Python pour déterminer la position des satellites à l'aide des fichiers de navigation.
- Développer une fonction Python permettant le traçage de l'orbite des satellites pendant une certaine période de temps.
- Développer une fonction Python permettant la génération de fichiers CSV contenant les résultats des calculs, à savoir la position des satellites.
- Développer une fonction Python permettant d'effectuer des calculs de la position de l'utilisateur par époque par moindre carré simple en exploitant les pseudorange C1
- Autrement dit, Exécutez un ajustement des moindres carrés époque par époque pour déterminer l'emplacement et le décalage d'horloge du récepteur.

## Afin d'atteindre ces objectives, il est nécessaire de :

- Comprendre la structure d'organisation des données et des informations dans les fichiers de navigation et d'observation rinex.
- Comprendre les algorithmes de calcul de la position des satellites à partir des éléments du message de navigation.
- Comprendre les différentes approches du calcul de la position du récepteur GPS, à savoir les Méthodes de compensation

## **Rinex :**

RINEX signifie Receiver Independent Exchange Format, il est stocké dans un format ascii et contient des observations GNSS.

Afin d'exploiter les données contenues dans les fichiers de navigation et d'observation, Il a été décidé que ces informations seraient stockées dans une structure de données de **dictionnaire python**. Un dictionnaire contient un ensemble de paires « clé-valeur » uniques qui permettent d'accéder rapidement aux informations, sans avoir à gérer des index de tableau désordonnés. L'implémentation n'est pas importante, il suffit de garder à l'esprit que les dictionnaires permettent de stocker les informations présentes

dans les deux fichiers RINEX et d'y accéder rapidement ultérieurement avec des mots-clés significatifs.

#### **Le programme se compose de quatre parties principales :**

**Parse\_rinex\_navigation\_file** : englobe l'ensemble des fonctions nécessaire pour la lecture du fichier rinex de navigation et l'extraction de ses données.

**Parse\_rinex\_observation\_file** : englobe l'ensemble des fonctions nécessaire pour la lecture du fichier rinex d'observation et l'extraction de ses données.

**SatPos** : les fonctions mathématiques du calcul de la position du satellite qui prennent en argument les objets générés par Parse\_rinex\_navigation\_file et Parse\_rinex\_observation\_file

**Least\_squares** : fonction de calcul et d'ajustement de la position de l'utilisateur par la méthode de compensation par moindre carré ,ainsi quelques fonctions de transformations géodésique .

**Main** : fonctions de lecture et traçage graphique des résultats, ainsi que programmation de l'interface console de l'application

#### **Calcul de la position des satellites :**

Pour calculer les coordonnées du récepteur, les coordonnées du satellite doivent d'abord être déterminées. Il s'agit d'un processus assez simple pour déterminer les coordonnées du satellite, les éléments képlériens du satellite sont diffusés dans le fichier de navigation. Chaque satellite de la constellation GPS effectue cette diffusion de tous les satellites GPS de la constellation et diffuse sa prédiction de position en orbite toutes les deux heures. Une fois que tous les 29 paramètres du fichier de navigation RINEX sont lus et stockés, le processus de détermination de la position du satellite se produit.

Le processus de détermination des positions satellites a été décrit en détail dans le document ICD. L'algorithme est défini dans la section 20.3.3.4.3 Algorithme utilisateur pour la détermination de l'éphéméride Pages 97-100. Les équations sont définies dans le tableau 20-IV du document ICD

Le tableau 1 fournit les paramètres des éphémérides de diffusion GPS pour calculer leurs coordonnées satellites à n'importe quelle époque d'observation. Ces paramètres sont renouvelés périodiquement (typiquement toutes les 2 heures pour le GPS) et ne doivent pas être utilisés hors du temps prescrit (environ 4 heures), car l'erreur d'extrapolation croît de façon exponentielle au-delà de sa période de validité.

L'algorithme fourni provient du [IS-GPS-200, tableau 20-IV]

NAVSTAR GPS Space Segment/Navigation User Interfaces

<https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf>

Parameter	Explanation
$t_{oe}$	Ephemerides reference epoch in seconds within the week
$\sqrt{a}$	Square root of semi-major axis
$e$	Eccentricity
$M_o$	Mean anomaly at reference epoch
$\omega$	Argument of perigee
$i_o$	Inclination at reference epoch
$\Omega_0$	Longitude of ascending node at the beginning of the week
$\Delta n$	Mean motion difference
$\dot{i}$	Rate of inclination angle
$\dot{\Omega}$	Rate of node's right ascension
$c_{uc}, c_{us}$	Latitude argument correction
$c_{rc}, c_{rs}$	Orbital radius correction
$c_{ic}, c_{is}$	Inclination correction
$a_0$	SV clock offset
$a_1$	SV clock drift
$a_2$	SV clock drift rate

Afin de calculer les coordonnées du satellite à partir du message de navigation, l'algorithme fourni comme suit doit être utilisé.

• Calculer le temps  $t_k$  à partir de l'époque de référence des éphémérides  $t_{oe}$  ( $t$  et  $t_{oe}$  sont exprimés en secondes dans la semaine GPS) :  $t_k = t - t_{oe}$

Si  $t_k > 302400$  sec, soustrayez 604800 sec de  $t_k$ .

Si  $t_k < -302400$  sec, ajouter 604800 sec.

- Calculer l'anomalie moyenne pour  $t_k$ ,

$$M_k = M_o + \left( \frac{\sqrt{\mu}}{\sqrt{a^3}} + \Delta n \right) t_k$$

- Résolvez (itérativement) l'équation de Kepler pour l'anomalie d'excentricité  $E_k$  :

$$M_k = E_k - e \sin E_k$$

- Calculez la véritable anomalie  $v_k$  :

$$v_k = \arctan \left( \frac{\sqrt{1-e^2} \sin E_k}{\cos E_k - e} \right)$$

- Calculer l'argument de latitude  $u_k$  à partir de l'argument du périégée  $\omega$ , de l'anomalie vraie  $v_k$  et des corrections  $c_{uc}$  et  $c_{us}$  :

$$u_k = \omega + v_k + c_{uc} \cos 2(\omega + v_k) + c_{us} \sin 2(\omega + v_k)$$

- Calculer la distance radiale  $r_k$  en tenant compte des corrections  $c_{rc}$  et  $c_{rs}$  :

$$r_k = a (1 - e \cos E_k) + c_{rc} \cos 2 (\omega + v_k) + c_{rs} \sin 2 (\omega + v_k)$$

- Calculer l'inclinaison  $i_k$  du plan orbital à partir de l'inclinaison  $i_o$  au temps de référence  $t_{oe}$ , et des corrections  $c_{ic}$  et  $c_{is}$  :

$$i_k = i_o + \dot{i} t_k + c_{ic} \cos 2 (\omega + v_k) + c_{is} \sin 2 (\omega + v_k)$$

- Calculer la longitude du nœud ascendant  $\lambda_k$  (par rapport à Greenwich). Ce calcul utilise l'ascension droite du début de la semaine en cours ( $\Omega_o$ ), la correction de la variation apparente du temps sidéral à Greenwich entre le début de la semaine et l'heure de référence  $t_k = t - t_{oe}$ , et le changement de longitude de l'ascendant nœud à partir de la pointe de temps de référence :

$$\lambda_k = \Omega_o + \left( \dot{\Omega} - \omega_E \right) t_k - \omega_E t_{oe}$$

- Calculer les coordonnées dans le référentiel TRS en appliquant trois rotations (autour de  $u_k$ ,  $i_k$  et  $\lambda_k$ ) :

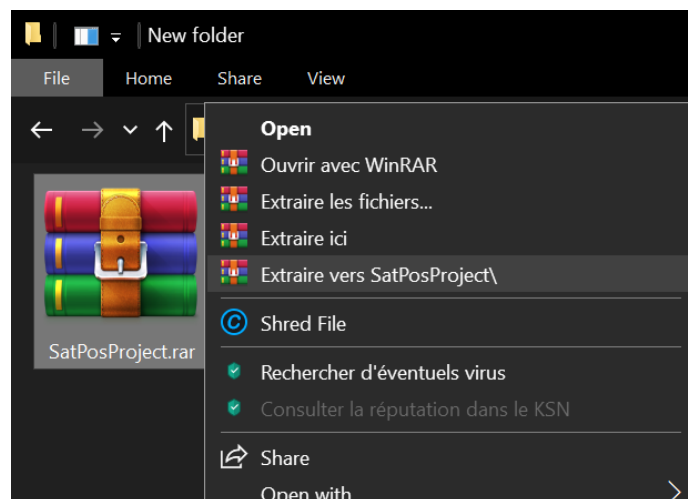
$$\begin{bmatrix} X_k \\ Y_k \\ Z_k \end{bmatrix} = \mathbf{R}_3 (-\lambda_k) \mathbf{R}_1 (-i_k) \mathbf{R}_3 (-u_k) \begin{bmatrix} r_k \\ 0 \\ 0 \end{bmatrix}$$

Avec :

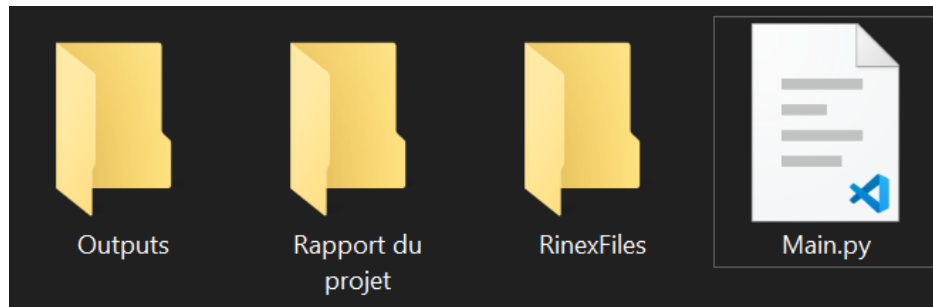
$$\mathbf{R}_1[\theta] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \mathbf{R}_2[\theta] = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \mathbf{R}_3[\theta] = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

## Guide d'utilisation de l'application :

### 1. Télécharger et extraire le fichier compressé :



## 2. Le dossier contient un fichier python et trois sous-dossiers :



**Main.py** : fichier python contenant le code de l'application

**Outputs** : les résultats après exécution du code ( fichiers csv et txt)

**Rapport du projet** : rapports pdf et word

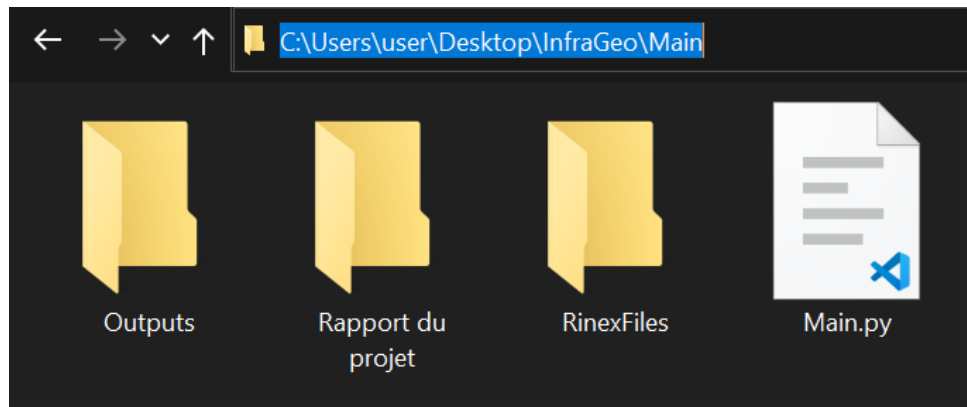
**RinexFiles** : fichiers rinex de navigation et d'observation de la IGS zimm correspondant  
À la date 21/12/14 (zimm3480.21n zimm3480.21o)

Lancer le code python soit par votre éditeur python soit directement dans le **cmd**  
comme il est montré ci-dessous

```
Command Prompt
Microsoft Windows [Version 10.0.19043.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>
```

Assurez vous de changer le répertoire courant par la commande **cd** :



Copiez le chemin et utilisez la commande `cd`

```
Command Prompt
Microsoft Windows [Version 10.0.19043.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>cd C:\Users\user\Desktop\InfraGeo\Main

C:\Users\user\Desktop\InfraGeo\Main>
```

Lancer le code python par la commande : `py Main.py`

```
Command Prompt
Microsoft Windows [Version 10.0.19043.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>cd C:\Users\user\Desktop\InfraGeo\Main

C:\Users\user\Desktop\InfraGeo\Main>py Main.py_
```

```
Command Prompt - py Main.py
Microsoft Windows [Version 10.0.19043.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>cd C:\Users\user\Desktop\InfraGeo\Main

C:\Users\user\Desktop\InfraGeo\Main>py Main.py

----- Application de calcul des positions des satellites GPS et de la position de l'utilisateur -----
----- @author: Driss L'hamdouchi , lhamdouchidrisss2@gmail.com -----

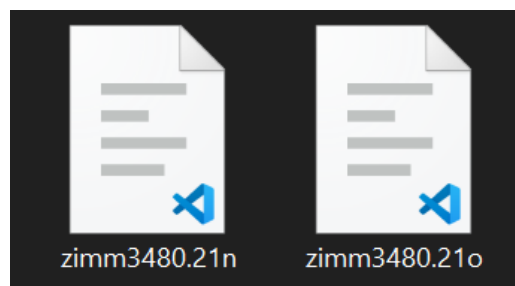
1 -- Choisir/changer le fichier rinex de navigation :
2 -- Choisir/changer le fichier rinex d'observation :
3 -- Régler Tstart ,Tfinish et l'intervalle
4 -- Afficher et tracer l'orbite des satellites
5 -- Calculer la position du récepteur pour chaque époque du fichier rinex d'observation
6 -- tracer ces positions
7 -- Exit

Fichier de navigation : Pas encore choisi
Fichier d'observation : Pas encore choisi
Tstart : Pas encore choisi
Tfinish : Pas encore choisi
intervale : Pas encore choisi

Entrez votre choix : _
```

Après suivissiez les étapes suivantes :

Tapez 1 et indiquez le nom de votre fichier rinex navigation assurer vous que ce dernier existe déjà dans le dossier RinexFiles



```
----- Application de calcul des positions des satellites GPS et de la position de l'utilisateur -----
----- @author: Driss L'hamdouchi , lhamdouchidrisss2@gmail.com -----

1 -- Choisir/changer le fichier rinex de navigation :
2 -- Choisir/changer le fichier rinex d'observation :
3 -- Régler Tstart ,Tfinish et l'intervalle
4 -- Afficher et tracer l'orbite des satellites
5 -- Calculer la position du récepteur pour chaque époque du fichier rinex d'observation
6 -- tracer ces positions
7 -- Exit

Fichier de navigation : Pas encore choisi
Fichier d'observation : Pas encore choisi
Tstart : Pas encore choisi
Tfinish : Pas encore choisi
intervale : Pas encore choisi

Entrez votre choix : 1

Veuillez écrire le nom du fichier rinex de navigation
Fichier de navigation : zimm3480.21n
```

Cliquez sur entrez et faites du même avec le fichier d'observation



**Remarque : j'ai effectué des modifications sur le fichier d'observation zimm3480.21o, notamment j'ai extrait seulement la partie d'observation jusqu'à 6 heure du matin afin de ne pas alourdir l'application**

```
Command Prompt - py Main.py
Entrez votre choix : 1

Veuillez écrire le nom du fichier rinex de navigation
Fichier de navigation : zimm3480.21n

1 -- Choisir/changer le fichier rinex de navigation :
2 -- Choisir/changer le fichier rinex d'observation :
3 -- Régler Tstart ,Tfinish et l'intervalle
4 -- Afficher et tracer l'orbite des satellites
5 -- Calculer la position du récepteur pour chaque époque du fichier rinex d'observation
6 -- tracer ces positions
7 -- Exit

Fichier de navigation : zimm3480.21n
Fichier d'observation : Pas encore choisi
Tstart : Pas encore choisi
Tfinish : Pas encore choisi
Intervale : Pas encore choisi

Entrez votre choix : 2

Veuillez écrire le nom du fichier rinex d'observation
Fichier d'observation : zimm3480.21o
```

**Choisissez l'option 3 :**

```
Command Prompt - py Main.py
Entrez votre choix : 2

Veuillez écrire le nom du fichier rinex d'observation
Fichier d'observation : zimm3480.21o

1 -- Choisir/changer le fichier rinex de navigation :
2 -- Choisir/changer le fichier rinex d'observation :
3 -- Régler Tstart ,Tfinish et l'intervalle
4 -- Afficher et tracer l'orbite des satellites
5 -- Calculer la position du récepteur pour chaque époque du fichier rinex d'observation
6 -- tracer ces positions
7 -- Exit

Fichier de navigation : zimm3480.21n
Fichier d'observation : zimm3480.21o
Tstart : Pas encore choisi
Tfinish : Pas encore choisi
Intervale : Pas encore choisi

Entrez votre choix : 3
Veuillez écrire les époques sous la format suivante 21:12:14:00:00:00
time start : 21:12:14:00:00:00
time finish : 21:12:15:00:00:00
intervalle en secondes : 1000_
```

**Indiquez la première époque Time Tstart que vous voulez et la dernière Time Tfinish en respectant la forme comme il est indiqué ci-dessus. Intervalle est en seconde c'est le pas du temps entre une position et une autre**

**Choisissez l'option 4, le résultat est comme montré :**

```
Command Prompt - py Main.py
Veuillez écrire les époques sous la format suivante 21:12:14:00:00:00
time start : 21:12:14:00:00:00
time finish : 21:12:15:00:00:00
intervalle en secondes : 1000

1 -- Choisir/changer le fichier rinex de navigation :
2 -- Choisir/changer le fichier rinex d'observation :
3 -- Régler Tstart ,Tfinish et l'intervalle
4 -- Afficher et tracer l'orbite des satellites
5 -- Calculer la position du récepteur pour chaque époque du fichier rinex d'observation
6 -- tracer ces positions
7 -- Exit

Fichier de navigation : zimm3480.21n
Fichier d'observation : zimm3480.21o
Tstart : 21:12:14:00:00:00
Tfinish : 21:12:15:00:00:00
intervale : 1000

Entrez votre choix : 4
G16//G21//G15//G08//G26//G18//G23//G07//G10//G27//G32//G01//G22//G14//G03//G17//G04//G19//G31//G09//G06//G11//G02//G20//
G30//G05//G29//G13//G24//G12//G25//

Veuillez choisir un satellite ( exemple : G18 ):
Entrez votre choix :
```

Après choisissez le satellite que vous voulez visualiser l'orbite :

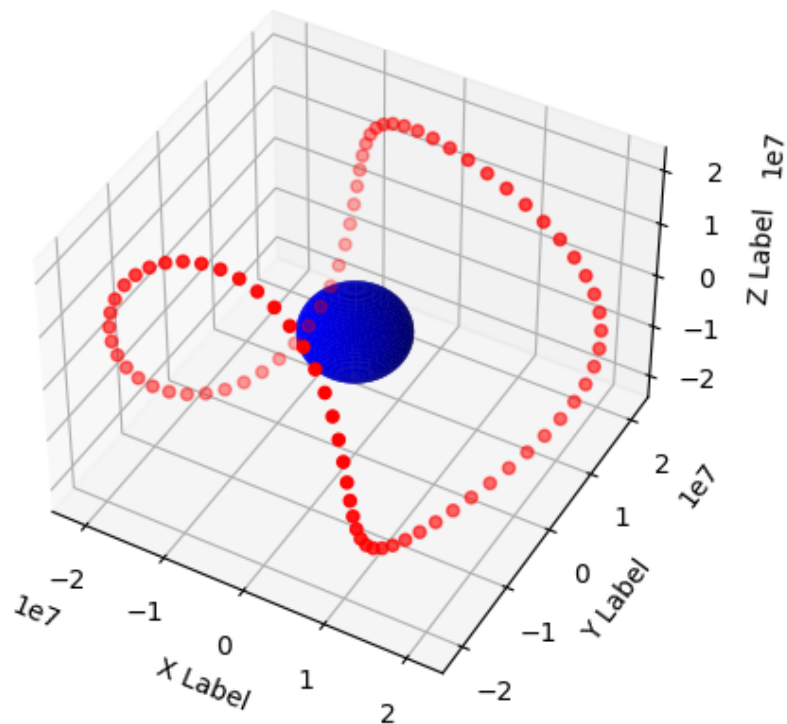
Veuillez choisir un satellite ( exemple : G18 ):  
Entrez votre choix : G14

```
Command Prompt - py Main.py
satellite : G14 At epoch 2021-12-14 01:56:40 x : -5630402.196279291 y : -14354394.283132834 z : 21626063.249971833
satellite : G14 At epoch 2021-12-14 02:13:20 x : -2991105.2354139006 y : -15134922.03839633 z : 21614055.91095553
satellite : G14 At epoch 2021-12-14 02:30:00 x : -428856.6935434388 y : -16053631.481889622 z : 21143006.634659767
satellite : G14 At epoch 2021-12-14 02:46:40 x : 1997492.7362024654 y : -17080252.8993679 z : 20222912.992654115
satellite : G14 At epoch 2021-12-14 03:03:20 x : 4236400.340398986 y : -18174727.549939442 z : 18873305.31026879
satellite : G14 At epoch 2021-12-14 03:20:00 x : 6245747.11075708 y : -19289069.526394233 z : 17122833.81843131
satellite : G14 At epoch 2021-12-14 03:36:40 x : 7994502.30630769 y : -20369630.74575507 z : 15008662.502656348
satellite : G14 At epoch 2021-12-14 03:53:20 x : 9463859.269817933 y : -21359661.41364123 z : 12575682.086069552
satellite : G14 At epoch 2021-12-14 04:10:00 x : 10647789.975469813 y : -22202043.984054383 z : 9875558.433291592
satellite : G14 At epoch 2021-12-14 04:26:40 x : 11552993.392822837 y : -22842070.089877453 z : 6965636.247211736
satellite : G14 At epoch 2021-12-14 04:43:20 x : 12198241.614281451 y : -23230127.592468783 z : 3907721.158268823
satellite : G14 At epoch 2021-12-14 05:00:00 x : 12613156.436362393 y : -23324168.941205718 z : 766766.0781429985
satellite : G14 At epoch 2021-12-14 05:16:40 x : 12836476.321715724 y : -23091842.28474397 z : -2390510.0881523346
satellite : G14 At epoch 2021-12-14 05:33:20 x : 12913898.068783423 y : -22512182.762566753 z : -5497041.689144703
satellite : G14 At epoch 2021-12-14 05:50:00 x : 12895597.877717596 y : -21576782.36198387 z : -8486842.957739746
satellite : G14 At epoch 2021-12-14 06:06:40 x : 12833551.819440026 y : -20290381.62956271 z : -11296411.29041283
satellite : G14 At epoch 2021-12-14 06:23:20 x : 12778785.237236584 y : -18670854.152464084 z : -13866077.014468638
satellite : G14 At epoch 2021-12-14 06:40:00 x : 12778683.879151996 y : -16748583.715421041 z : -16141270.766927084
satellite : G14 At epoch 2021-12-14 06:56:40 x : 12874496.433704345 y : -14565262.975273108 z : -18073681.545516163
satellite : G14 At epoch 2021-12-14 07:13:20 x : 13099148.799437016 y : -12172169.977004634 z : -19622281.00537982
satellite : G14 At epoch 2021-12-14 07:30:00 x : 13475475.34180132 y : -9628003.552997872 z : -20754192.574677378
satellite : G14 At epoch 2021-12-14 07:46:40 x : 14014952.330325082 y : -6996379.443186572 z : -21445387.354473036
satellite : G14 At epoch 2021-12-14 08:03:20 x : 14716994.682338564 y : -4343104.893254481 z : -21681192.463083733
satellite : G14 At epoch 2021-12-14 08:20:00 x : 15568850.217904454 y : -1733359.8156042835 z : -21456601.401225727
satellite : G14 At epoch 2021-12-14 08:36:40 x : 16546097.123450797 y : 771083.1189605221 z : -20776380.08178616
satellite : G14 At epoch 2021-12-14 08:53:20 x : 17613721.56006758 y : 3114469.0297000166 z : -19654966.326703113
satellite : G14 At epoch 2021-12-14 09:10:00 x : 18727724.67360328 y : 5249379.646781278 z : -18116164.82926287
satellite : G14 At epoch 2021-12-14 09:26:40 x : 19837182.956272766 y : 7138651.850211296 z : -16192643.759867622
satellite : G14 At epoch 2021-12-14 09:43:20 x : 20886664.160815947 y : 8756806.3105056 z : -13925243.29888331
satellite : G14 At epoch 2021-12-14 10:00:00 x : 21818883.813236278 y : 10090919.814333335 z : -11362110.344120272
satellite : G14 At epoch 2021-12-14 10:16:40 x : 22577475.642925575 y : 11140901.252808971 z : -8557677.383935206
satellite : G14 At epoch 2021-12-14 10:33:20 x : 23109743.539407868 y : 11919159.6098598 z : -5571506.960318839
satellite : G14 At epoch 2021-12-14 10:50:00 x : 23369263.262958795 y : 12449681.272823963 z : -2467026.17330774
```

Le résultat est la position du satellite dans le référentiel ECEF entre Tstart et Tfinish avec le pas intervalle

L'application génère dans le dossier Output un fichier csv contenant les résultats

la position du satellite :G14



Ainsi que l'orbite du satellite dans le référentiel ECEF qui est liée à la terre

La visualisation est un moyen de vérifier que le calcul est bien fait, c'est exactement la forme de l'orbite des satellites dans ce référentiel qu'est liée à la terre.

Pour calculer la position de l'utilisateur choisissez l'option 5 :

```
Select Command Prompt - py Main.py
satellite : G14 At epoch 2021-12-14 23:03:20 x : -23333146.750111107 y : -12750100.609593306 z : 447949.83875283774
satellite : G14 At epoch 2021-12-14 23:20:00 x : -22970786.521489836 y : -12907029.077760682 z : 3593625.8696988043
satellite : G14 At epoch 2021-12-14 23:36:40 x : -22255998.568430573 y : -12936844.09538031 z : 6662931.504401984

1 -- Choisir/changer le fichier rinex de navigation :
2 -- Choisir/changer le fichier rinex d'observation :
3 -- Régler Tstart ,Tfinish et l'intervalle
4 -- Afficher et tracer l'orbite des satellites
5 -- Calculer la position du récepteur pour chaque époque du fichier rinex d'observation
6 -- tracer ces positions
7 -- Exit

Fichier de navigation : zimm3480.21n
Fichier d'observation : zimm3480.21o
Tstart : 21:12:14:00:00:00
Tfinish : 21:12:15:00:00:00
intervale : 1000

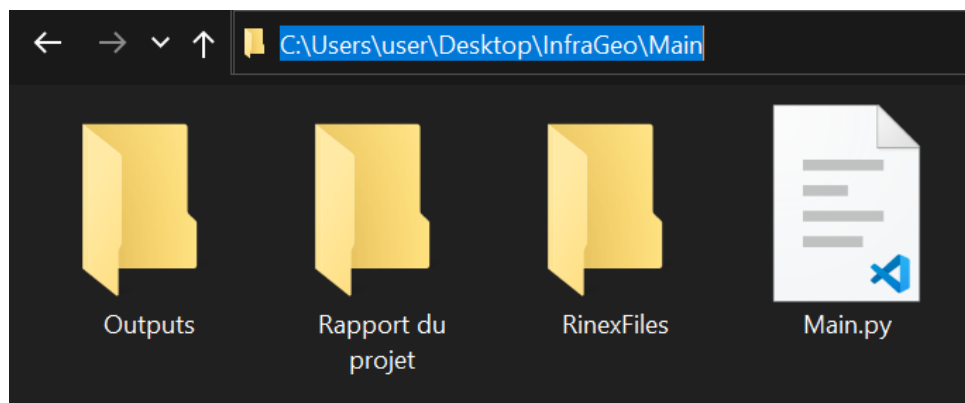
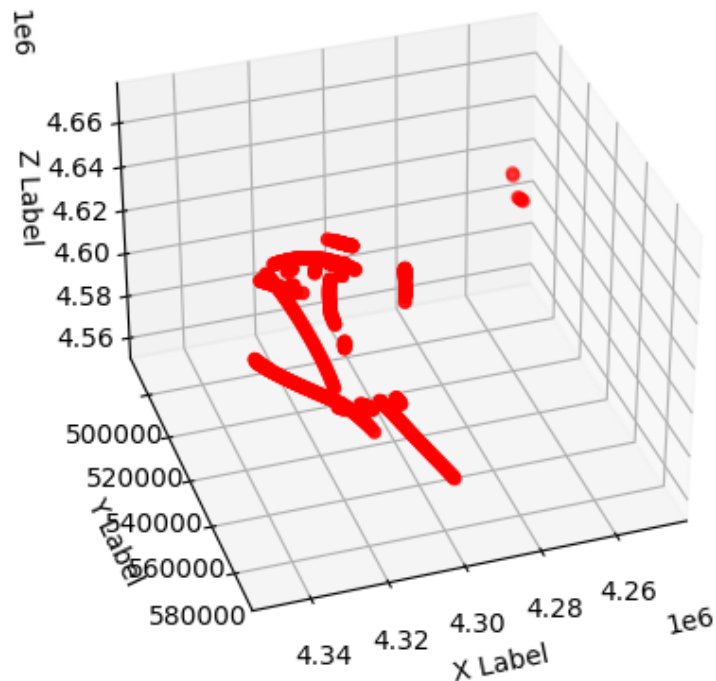
Entrez votre choix : 5
```

```
Command Prompt - py Main.py
21:12:14:0:52:0.000000 [4249394.63515041 494204.63264249 4604970.95504662]
21:12:14:0:52:30.000000 [4249603.1196241 494225.26484227 4605259.0339428 ]
21:12:14:0:53:0.000000 [4249820.33450064 494253.8251542 4605546.62242419]
21:12:14:0:53:30.000000 [4250047.21097713 494288.69064174 4605838.97256196]
21:12:14:0:54:0.000000 [4318393.31315857 554549.71572869 4647398.97242949]
21:12:14:0:54:30.000000 [4318406.49952267 554610.99371773 4647394.4855346 ]
21:12:14:0:55:0.000000 [4318418.19075517 554671.43976308 4647389.81279805]
21:12:14:0:55:30.000000 [4330650.96390699 561919.77516711 4649978.98348994]
21:12:14:0:56:0.000000 [4318442.56071666 554792.68561606 4647378.71297703]
21:12:14:0:56:30.000000 [4318454.57410047 554852.77991566 4647372.88301617]
21:12:14:0:57:0.000000 [4318464.0084595 554912.5454933 4647364.7797625]
21:12:14:0:57:30.000000 [4318476.90767557 554973.10592736 4647356.7375013 ]
21:12:14:0:58:0.000000 [4318487.76059046 555032.8901612 4647347.75572103]
21:12:14:0:58:30.000000 [4318498.38720017 555093.06905564 4647336.78742873]
21:12:14:0:59:0.000000 [4318506.99432861 555152.88086276 4647327.35955479]
21:12:14:0:59:30.000000 [4315224.93897754 553467.75209178 4657621.37007265]
21:12:14:1:0:0.000000 [4315286.34630212 553540.34808126 4657777.62028329]
21:12:14:1:0:30.000000 [4315349.0038228 553614.17464505 4657934.03113755]
21:12:14:1:1:0.000000 [4315413.51011276 553689.0121145 4658089.85156005]
21:12:14:1:1:30.000000 [4315477.86940845 553764.55031377 4658245.87366593]
21:12:14:1:2:0.000000 [4315542.8389091 553839.84433567 4658401.13467089]
21:12:14:1:2:30.000000 [4315609.72654169 553916.58589728 4658555.57295659]
21:12:14:1:3:0.000000 [4316448.98180779 553885.06649353 4658957.05458477]
21:12:14:1:3:30.000000 [4316620.46327601 553948.97843954 4659147.93985527]
21:12:14:1:4:0.000000 [4316794.53991299 554014.38790849 4659339.95068178]
21:12:14:1:4:30.000000 [4316967.38290531 554079.86788602 4659531.04196937]
21:12:14:1:5:0.000000 [4317143.22516478 554146.19707603 4659724.39689637]
21:12:14:1:5:30.000000 [4317318.97879032 554213.22877805 4659917.63557688]
21:12:14:1:6:0.000000 [4317494.09838876 554280.73823332 4660109.87315586]
21:12:14:1:6:30.000000 [4317670.37620715 554349.8581602 4660302.80201741]
21:12:14:1:7:0.000000 [4317848.42099098 554418.48058844 4660497.06872433]
21:12:14:1:7:30.000000 [4318025.65843897 554488.20438354 4660690.49051749]
```

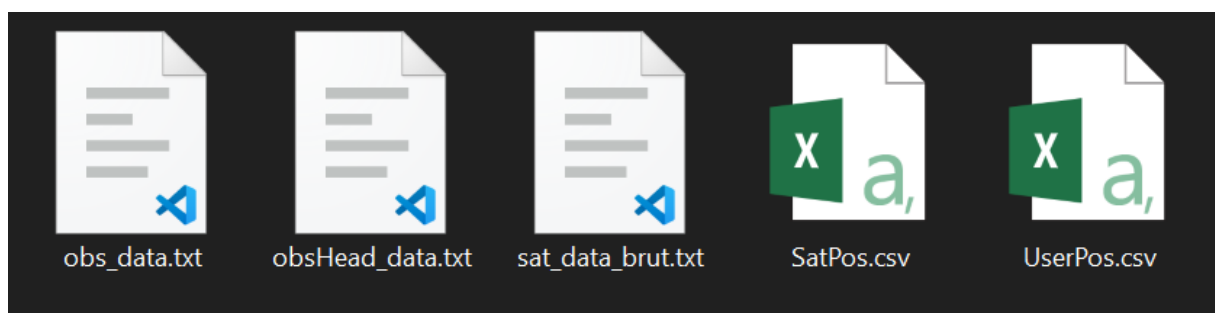
L'application commence à calculer la position de l'utilisateur à chaque époque

Et génère dans le dossier output un fichier csv contenant les résultats

Pour visualiser ces position choisissez l'option 6



Dans le dossier Outputs vous trouverez les résultats de votre manipulation de l'application



## **Analyse des résultats et un commentaire général sur le fonctionnement et la phase de recherche et de programmation de l'application :**

Ce rapport implique tous les composants logiciels nécessaires pour calculer une solution de navigation GPS Standard Positioning Service. Le logiciel a été testé pour un cas test de données d'observation, bien qu'une position ait été sortie, la qualité de la solution n'était pas acceptable. Les bogues logiciels à l'origine de l'écart n'ont pas pu être déterminés sur la durée de ce projet en raison de contraintes de temps. S'il y avait plus de temps, le retard troposphérique du signal GPS pourrait également être mis en œuvre et testé ainsi que le modèle de Klobuchar pour déterminer le retard ionosphérique pour un récepteur à fréquence unique.

Enfin je veux remercier le prof pour ce devoir puisqu'il m'a vraiment aidé à approfondir mes connaissances à propos le fonctionnement du GNSS et notamment GPS, ainsi que mes compétences en programmation sur base de la langage Python, le temps était une contrainte, on rend le travail mais ce n'est pas fini pour moi, cette application va -- inshallah- avoir des révisions et des améliorations en bien introduisant les corrections

Et en cherchant ou réside exactement le problème. Si vous avez des questions sur le n'hésiter pas de m'envoyer un mail [lhamdouchidriss2@gmail.com](mailto:lhamdouchidriss2@gmail.com)