Détermination de la masse de la planète Uranus

T.P. écrit par A. Klotz, F. Colas, D. Darson

Ce Travail Pratique (TP) propose de calculer la masse de la planète Uranus à partir de l'observation de ses satellites.

- Durée des observations : 1 heure pendant 4 nuits consécutives
- Instrument : Caméra infrarouge de préférence, sinon caméra visible.
- Contraintes de Lune : Non
- Théorie : application de la troisième loi de Kepler.
- Analyse d'images : Savoir afficher une image FITS, mesurer la position d'un astre avec un ajustement gaussien
- Analyse des données : Python, numpy array, ajustement d'une ellipse, ajustement d'une droite. Matplotlib pour afficher des graphiques.

Ce fascicule apporte une méthode et quelques aides mathématiques et de programmation de façon non exhaustive.

1. Préparation des observations

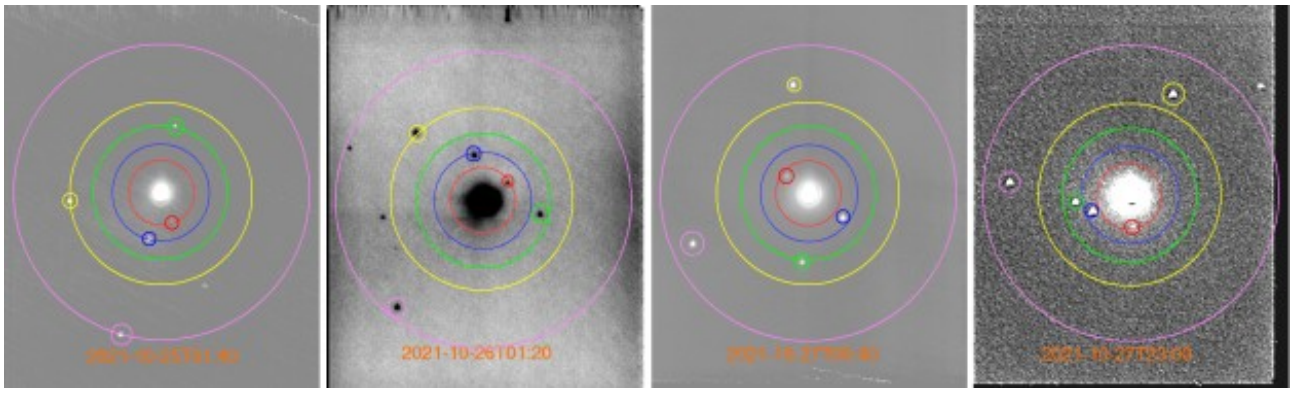
Calculer les coordonnées Ra,Dec de la planète Uranus (cf. annexe 1). Déduire si la planète est observable pendant la période de TP (chercher les moments de la nuit où l'élévation est supérieure à 15°). Idéalement on observera pendant que la planète passe au méridien (plus haute élévation).

2. Observations et enregistrement des images

Après avoir centré la planète sur le capteur, enregistrer des séries de courtes poses. Le but est de ne pas (trop) saturer la planète en faisant monter le flux des satellites.

3. Identifier les satellites sur les images

On peut identifier 5 satellites. Idéalement il faut des images étalées sur au moins quatre nuits. Il est important de ne pas se tromper d'assignation. Pour y arriver, dans l'exemple ci-dessous, on trace des cercles de différentes couleurs passant par les points candidats satellites. On reporte les mêmes cercles (même rayon) sur les images suivantes prises entre le 25 et le 28 octobre 2021 au T1M :



Au voisinage des cercles, on repère la position des satellites d'image en images. Sur l'exemple ci-dessus on voit bien le satellite jaune (Titania) tourner dans le sens des aiguilles d'une montre de nuit en nuit.

A ce niveau, notre approximation d'orbite par des cercles est très approximative et ne sert juste qu'à identifier la position de chaque satellite sur les images.

Rouge = Miranda
Bleu = Ariel
Vert = Umbriel
Jaune = Titania
Magenta = Oberon

4. Mesurer les positions (x,y) des satellites et d'Uranus sur les images

Pour chaque image, on mesure la position (x,y) des satellites et de la planète. On pourra, par exemple, choisir un ajustement gaussien 2D.

Le tableau suivant reporte les positions mesurées :

========= image 25.069 ==========		
Body	x (pixel)	y (pixel)
Uranus	255.81	324.30
Miranda	273.00	274.70
Ariel	236.04	247.62
Umbriel	277.88	431.30
Titania	107.40	310.37
Oberon	189.87	93.56
========= image 26.056 ==========		
Body	x (pixel)	y (pixel)
Uranus	253.03	307.07
Miranda	294.41	338.13
Ariel	240.18	385.05
Umbriel	346.90	288.74
Titania	145.71	421.14
Oberon	115.19	138.37

| ========= image 27.028 ========== |
Body	x (pixel)	y (pixel)
Uranus	256.47	320.85
Miranda	220.39	348.33
Ariel	310.89	282.27
Umbriel	244.83	211.05
Titania	230.20	497.69
Oberon	67.56	239.93

| ========= image 27.958 ========== |
Body	x (pixel)	y (pixel)
Uranus	252.44	315.34
Miranda	259.87	268.23
Ariel	194.53	292.21
Umbriel	166.07	309.73
Titania	324.38	482.14
Oberon	60.07	340.82

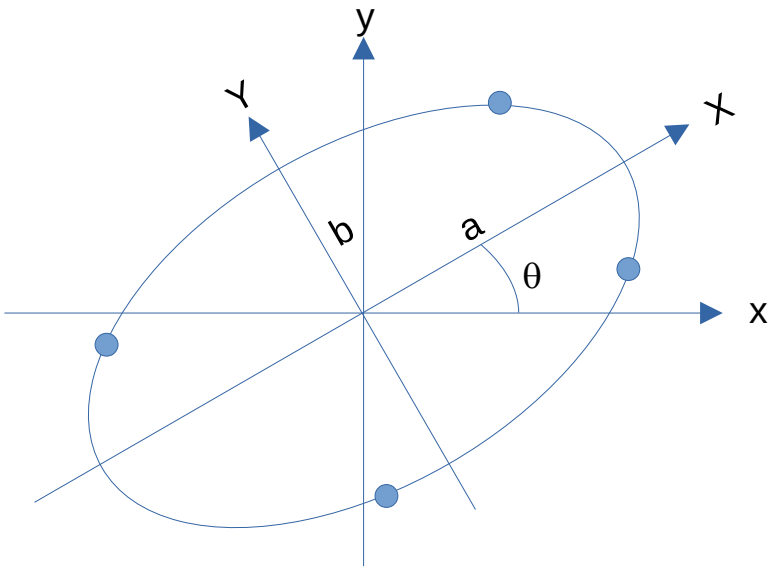
5. Equations de l'ellipse à ajuster

Les mesures de photocentre (x,y) sont à corriger de la position du centre de la planète. On se trouve alors dans un système de coordonnées où les axes (x,y) correspondent aux axes des pixels du détecteur.

En considérant que les orbites des satellites d'Uranus sont circulaires et d'inclinaison nulle par rapport à l'équateur de la planète, on n'aura à corriger qu'un effet de perspective du au fait que le pôle d'Uranus ne nous fait face. Cet angle transforme les cercles des orbites en orbites centrées sur la planète. On désignera (a,b) les demi grand et demi petits axes. Ces axes seront portés par le système de coordonnées (X,Y) tel que :

$$1=\left(\frac{X}{a}\right)^2+\left(\frac{Y}{b}\right)^2$$

Enfin, les ellipses apparentes seront aussi inclinées par rapport au système (x,y) d'un angle θ comme le montre le schéma ci-dessous (les points bleus sont les positions mesurées d'un satellite sur les quatre dates) :



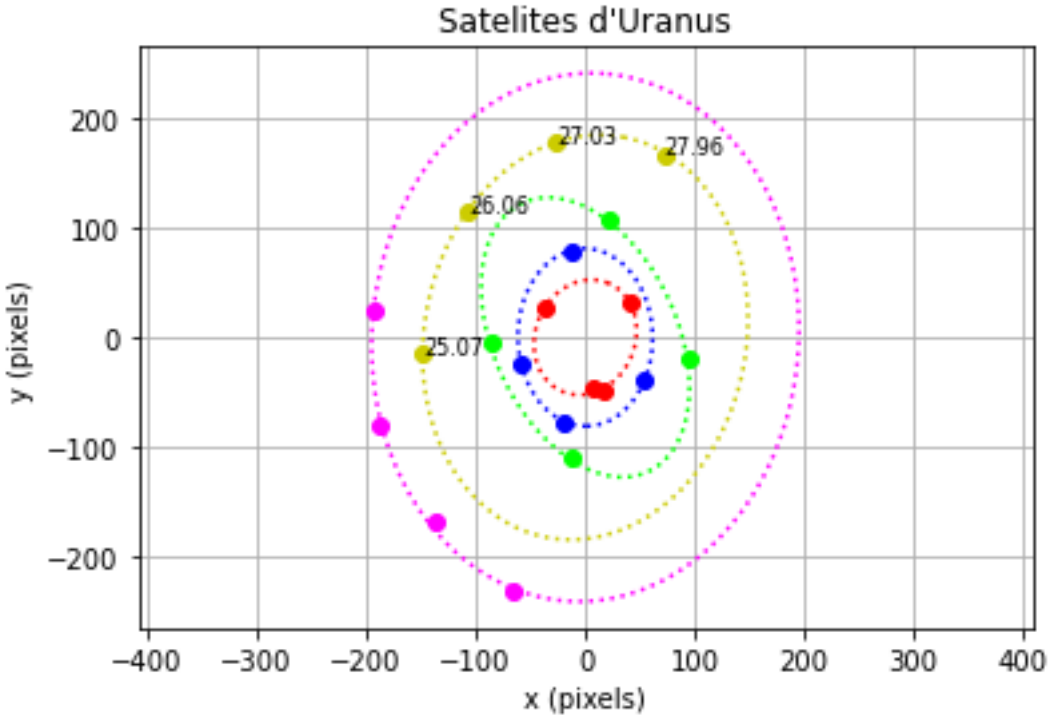
Les équations matricielles de passage (x,y) vers (X,Y) et vice-cersa sont :

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix} \qquad \begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

6. Méthode d'ajustement des ellipses

Pour une ellipse (c'est à dire pour un satellite) nous avons quatre couples de valeur (x,y) et il faut trouver le triplet (a, b, θ) qui s'ajuste au mieux sur les quatre points. Vue non complexité du problème, on peut se permettre d'effectuer une recherche en imbriquant trois boucles sur les variables (a, b, θ). On se référera à l'annexe 2 pour la résolution de cette méthodes en Python.

A priori les rapports b/a et l'angle θ doivent être les mêmes pour tous les satellites. Évidemment ça ne sera probablement pas le cas car nos mesures sont entachées de diverses incertitudes. On a même un cas particulier avec le satellite Umbriel qui tourne en presque exactement 4 jours. Si l'on a 4 observations espacées de 24h, on aura alors quatre points de mesure qui forment un quadrilatère menant à plusieurs solutions d'ellipses possibles.



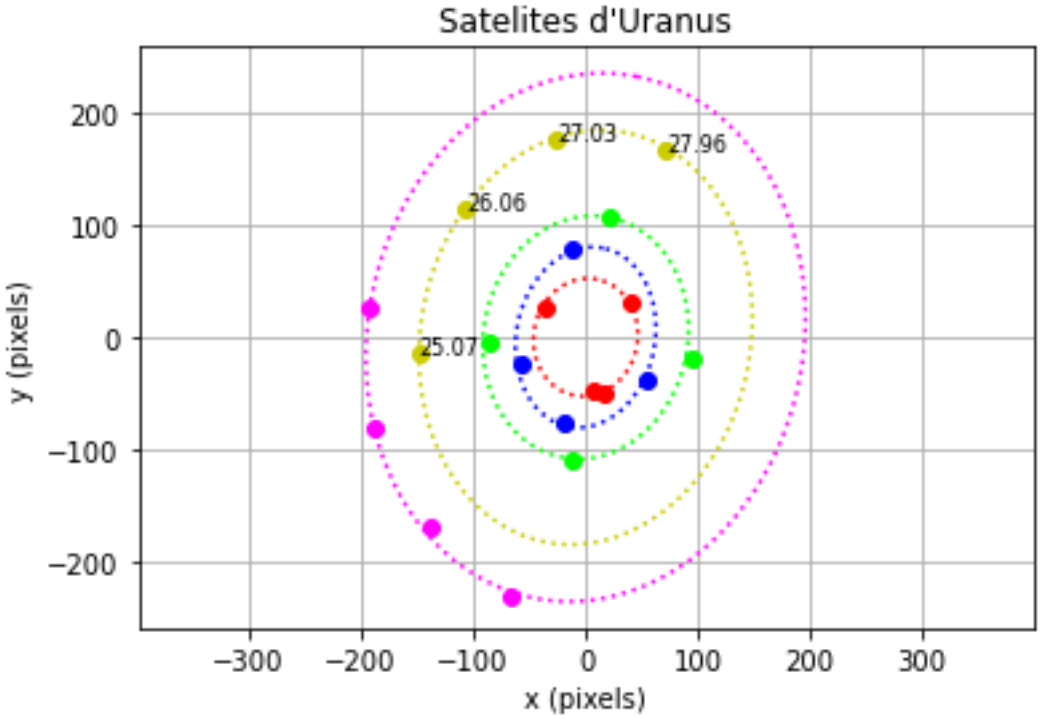
Le satellite Umbriel (en vert) n'est pas adapté à l'ajustement d'ellipse pour 4 observations séparées de 24h. Sur le satellite Titania (en jaune) on a indiqué les dates de mesures.

On va remédier au problème de Umbriel en calculant les rapports b/a et l'angle θ médian des satellites. Par exemple, dans notre cas, on trouve :

Ratio b/a median = 0.7911
Theta median = 1.1671 radian

Ainsi on peut rejouer l'ajustement mais on n'ajuste que a et on impose b = a / 0.7911 et θ = 1.1671 radian. Ces valeurs dépendront de la date d'observation de votre TP.

Ainsi, une fois imposés le rapport b/a et q à la valeur médiane, on a une représentation plus satisfaisante après un nouvel ajustement de a uniquement :



Après contrainte sur la rapport b/a et l'angle θ, les ajustements sont corrects.

7. Détermination des demi-grands axes des orbites

Le demi grand axe des ellipses (variable a) est actuellement exprimé en pixels. Il convient de déterminer l'échelle permettant de convertir des pixels en des longueurs à la distance de la planète Uranus.

La distance d'Uranus se calcule avec les éphémérides (cf. annexe 1).
L'échantillonnage spatial est l'échelle exprimée généralement en arcsec/pixel. On la détermine grâce à deux étoiles présentes sur l'image du 26 octobre. A partir de la position (x,y) des deux étoiles, on en déduit leur distance en pixels (380.98 pix) et, avec l'outil Aladin, on peut mesurer leur séparation angulaire (1.121 arcmin). On en déduit alors l'échantillonnage spatial de 0.177 arcsec/pixel.

Grâce à la valeur d'échantillonnage spatial, on peut convertir a (pixel) en a (radians) puis, grâce à la distance d'Uranus, on en déduit a (m) :

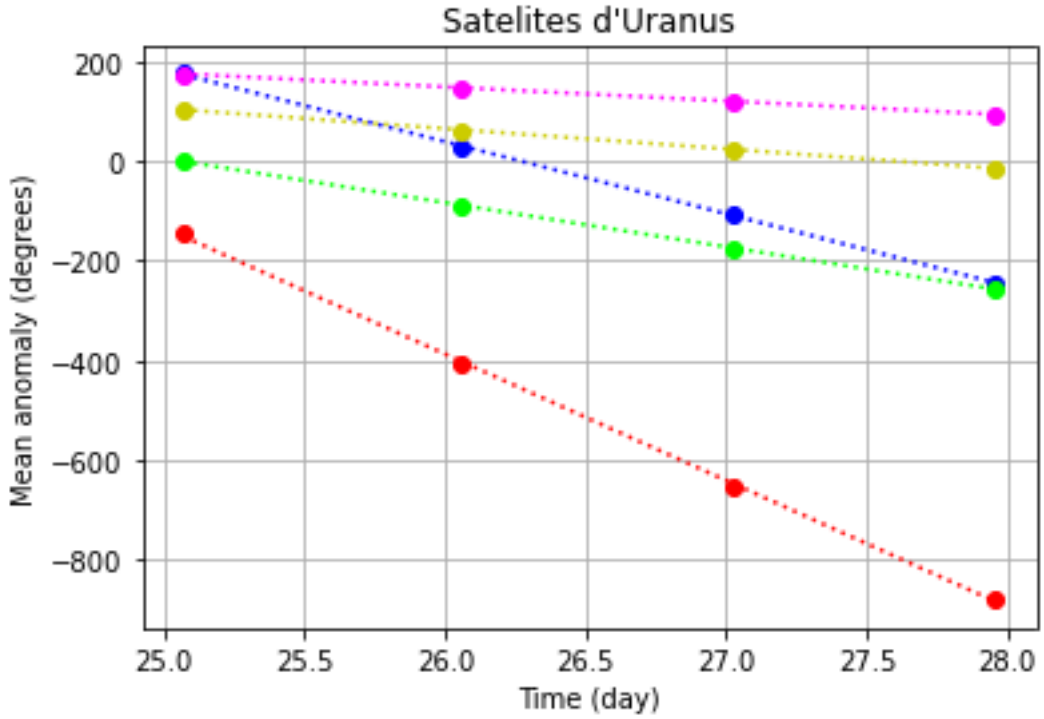
```
Planet distance = 18.753 a.u. (from ephemeris)
Spatial sampling = 0.177 arcsec/pix (from separation of two stars)
Sensor pixel size = 15.0 um (from manufacturer)
Focal length of optics = 17.525 m (computed)
-------------------------------------
| Satellite | a (pixel) |   a (m)   |
| Miranda   |     52.7  | 1.266e+08 |
| Ariel     |     81.0  | 1.944e+08 |
| Umbriel   |    108.8  | 2.613e+08 |
| Titania   |    185.7  | 4.460e+08 |
| Oberon    |    236.5  | 5.680e+08 |
-------------------------------------
```

8. Détermination des périodes des orbites

La détermination de la période est plus difficile que pour le demi-grand axe. Pour une orbite donnée, on a effectuer une transformation des coordonnées mesurées (x,y) en (X,Y) puisque l'on connaît (a,b,θ).

Maintenant nous allons nous placer face au pôle de l'orbite en étirant l'axe Y de l'effet de perspective. Pour cela on divise simplement Y par le rapport (b/a). X ne change pas.

A partir des couples (X,Y) "vus de face", on en déduit l'angle, désigné habituellement par anomalie moyenne M tel que M = atan2(Y,X). On va ainsi obtenir 4 valeurs de l'angle M pour les 4 dates. M doit varier linéairement en fonction du temps. Ainsi, l'ajustement d'une droite sur la représentation de M en fonction du temps, permet d'en déduire précisément la période avec la pente :



Il est important de comprendre dans quel sens évolue M en fonction du temps. Dans notre exemple, M décroît avec le temps qui passe (les pentes sont négatives). A noter que la fonction atan2 effectue un module 360° qu'il conviendra de corriger au besoin afin de garder les points alignés sur une droite comme sur le graphique ci-dessus.

Satellite	period (s)	period (day)
Miranda	1.228e+05	1.421
Ariel	2.152e+05	2.491
Umbriel	3.508e+05	4.060
Titania	7.653e+05	8.858
Oberon	1.106e+06	12.798

9. Détermination de la masse d'Uranus

La troisième loi de Kepler permet d'obtenir la masse d'Uranus indépendamment pour chaque satellite.

$$M = \frac{a^3 4\pi^2}{G T^2}$$

Il conviendra de convertir a en mètres et T en secondes pour utiliser la constante de la gravitation G en unités du système international.

```
-----------------------------------------------------
| Satellite |    a (m)    | period (day) | mass (kg) |
| Miranda   |  1.266e+08  |     1.421    | 7.951e+25 |
| Ariel     |  1.944e+08  |     2.491    | 9.387e+25 |
| Umbriel   |  2.613e+08  |     4.060    | 8.580e+25 |
| Titania   |  4.460e+08  |     8.858    | 8.959e+25 |
| Oberon    |  5.680e+08  |    12.798    | 8.866e+25 |
-----------------------------------------------------
Uranus mean mass = 8.75e+25 kg
Uranus mass uncertainty = 9.51e+24 kg = 10.9 percent (at 2 sigmas)
Mass difference with real = 0.8 percent
```

On trouve ici un excellent résultat à mieux qu'un pourcent près.

En l'absence de comparaison avec la valeur réelle on aurait annoncé $8.8 \cdot 10^{25}$ kg avec une incertitude de 11 % à 95 % de confiance (2 sigmas).

Il est à noter que si l'on n'effectue pas la contrainte b/a et θ décrit à la section 6, on trouve un écart de 25 % à la valeur réelle et une incertitude de 64 % à 2 sigmas.

A noter que la mesure du photocentre d'Uranus elle même mériterait une approche plus précise que le simple ajustement gaussien.

Sur les images infrarouge d'Uranus on voit les anneaux. Il est intéressant de mesurer leur demi-grand axe et de discuter de la notion de limite de Roche puisque nous connaissons maintenant la masse de la planète.

Annexe 1 – Calculer une éphéméride de planète avec Python

```
import skyfield.api
from skyfield.api import N, E, wgs84

ts = skyfield.api.load.timescale()
# Entrer ici la date UTC en format year, month, day, h, m, s
t = ts.utc(2021, 10, 26, 1, 20, 0)
planets = skyfield.api.load('de421.bsp')
earth, uranus = planets['earth'], planets['uranus barycenter']

# Position du T1M sur la Terre
t1m = earth + wgs84.latlon(42.936639 * N, 0.1423 * E)
astrometric = t1m.at(t).observe(uranus)
ra, dec, distance = astrometric.radec()

print(ra)
print(dec)
print(distance)

difference = uranus - t1m
topocentric = difference.at(t)

elev, az, distance = topocentric.altaz()
print('Elevation = ', elev)
print('Az = ', az)
```

Annexe 2 – Ajustement des ellipses en Python

Le code ci-dessous montre la méthode des trois boucles imbriquées pour trouver le meilleur triplet (a, b, θ). Les points de mesure sont stockés dans le array xys pour un satellite donné. Si on a 4 dates de mesures alors array xys est composé de 4 arrays xy chacun définis par les coordonnées x et y.

```
a = 56
b = 45
aas = np.linspace(0.6*a, 1.5*a, 50)
bbs = np.linspace(0.6*b, 1.5*b, 50)
thetas = np.linspace(-np.pi/2, np.pi/2, 180)
residue2_mini = 3e10 # any very large number is OK
for a in aas:
    for b in bbs:
        if b > a:
            continue
        for theta in thetas:
            residue2 = 0
            for xy in xys:
                x, y = xy
                X =  x*np.cos(theta) + y*np.sin(theta)
                Y = -x*np.sin(theta) + y*np.cos(theta)
                residue = (X/a)*(X/a) + (Y/b)*(Y/b) - 1
                residue2 += residue*residue
            if residue2 < residue2_mini:
                # Found a beter solution
                #print(f"a={a:.2f} b={b:.2f} theta={np.degrees(theta):+7.2f} residue2={residue2}")
                residue2_mini = residue2
                solution = np.array([a, b, theta])
```

Le code Python calcule la variable résidue avec les équations de l'ellipse d'écrites dans le texte principal. La variable résidue2 est simplement le carré de résidue pour éviter les nombres négatifs. Enfin on recherche le minimum de résidue2. A chaque fois que résidue2 est inférieur au minimum déjà trouvé alors on met à jour la variable solution.