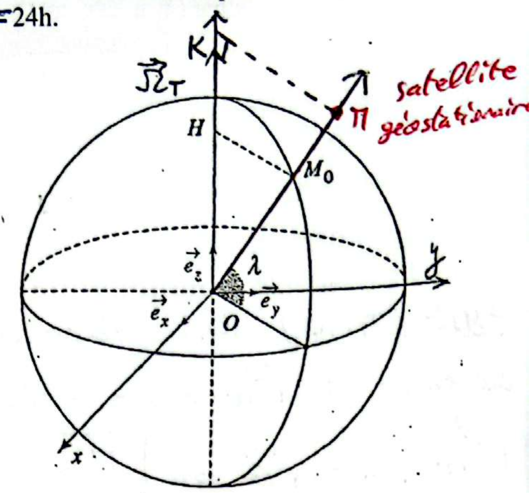
b) Satellites géostationnaires : Satellites de télécommunication ou de télédiffusion, météorologique ou d'alerte.

Un satellite géostationnaire paraît fixe pour un observateur lié à la terre. Il est situé dans le plan de l'équateur, à une altitude z ≈ 36 000 km et décrit une trajectoire circulaire de période Tsol ≃ 24h.

Période sidérale Tsid = 86 164s.



Satellite d'alerte : conçus pour détecter des tirs de missiles balistiques avec détecteur IR

Π décrit un cercle de centre K, proj orth de Π sur l'axe de rotation de la terre de période $T_{sid} \simeq T_{sol}$

TMC appliqué à Π en O fixe dans R_0 galiléen (réf géocentrique de centre O, les axes pointent vers 3 étoiles fixes) Π(m) soumis uniquement à $\vec{F}_{m_T \to m} = -G\frac{m_T m}{r^2}\vec{e_r}$

$$\frac{d\vec{L_O}(\Pi/R)}{dt} = \vec{M_O}(\vec{F}_{m_T\to m}) = \vec{O\Pi} \wedge \vec{F}_{m_T \to m} = \vec{0}$$

$\vec{L_O}(\Pi/R) = \vec{O\Pi} \wedge m\vec{v}(\Pi/R) = \vec{cste}$

$\forall t \begin{vmatrix} \vec{O\Pi} \perp \vec{L_O}(\Pi/R) \\ \vec{v}(\Pi/R) \perp \vec{L_O}(\Pi/R) \end{vmatrix}$

⇒ Le mvt se fait ds 1 plan passant par O

Or Π décrit un cercle de centre centré sur l'axe de rotation

⇒ Le mvt de Π se fait dans le plan équatorial (= passant par O et ⊥ à l'axe de rota de la terre)

LFD : $m\vec{a} = \vec{F}_{m_T \to m}$ ds le plan équatorial

$\vec{O\Pi} = r\vec{e_r}$ (coordonnées polaires)

$\vec{v} = r\dot{\theta}\vec{e_\theta}$

$\vec{a} = r\ddot{\theta}\vec{e_\theta} - r\dot{\theta}^2\vec{e_r}$

$$m(r\ddot{\theta}\vec{e_\theta} - r\dot{\theta}^2\vec{e_r}) = -G\frac{m_T m}{r^2}\vec{e_r}$$

Sur $(\vec{e_r})$: $r\dot{\theta}^2 = \frac{Gm_T}{r^2}$

$$\Rightarrow \boxed{\frac{v^2}{r} = \frac{Gm_T}{r^2}} \quad (1)$$

or $v = \frac{2\pi r}{T} \Rightarrow v^2 = \frac{4\pi^2 r^2}{T^2}$

d'où r = cste

$$(1) \Rightarrow v^2 = \frac{4\pi^2 r^2}{T^2} = \frac{Gm_T}{r}$$

$$\Rightarrow r^3 = \frac{Gm_T T^2}{4\pi^2}$$

$$\Rightarrow r = \left(\frac{Gm_T T^2}{4\pi^2}\right)^{1/3}$$

Or $Gm_T = g_0 R_T^2$

$$\Rightarrow \boxed{r = \left(\frac{g_0 R_T^2 T^2}{4\pi^2}\right)^{1/3}}$$

$r = R_T + z$

$$\Rightarrow \boxed{z = \left(\frac{g_0 R_T^2 T^2}{4\pi^2}\right)^{1/3} - R_T}$$

AN : T = ~~Tsol~~ Tsid $\begin{vmatrix} g_0 = 9{,}81\ m.s^{-2} \\ R_T = 6380\ km \end{vmatrix}$

⇒ z = 35 840 . 10^3 m

≃ 36 000 km

$$T_{sid} = \frac{2\pi \times T_{sol}}{2\pi + \alpha} = \frac{T_{sol}}{1 + \frac{\alpha}{2\pi}}$$

Or $\frac{\alpha}{2\pi} = \frac{1}{365{,}25}$

$$\Rightarrow T_{sid} = \frac{T_{sol}}{1 + \frac{1}{365{,}25}} = 86\,164\,s$$

*

Rmq : si Tsid n'est pas donnée, on utilise Tsol = 24h

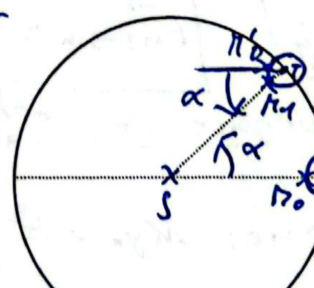
Rmq : Diff jour idéal, jour observé

entre Π_0 et Π_0 1 jour idéal

entre Π_0 et Π_1 1 jour observé

Au bout d'1 an, Π_0 est revenue à la mm position

$\alpha \times 365{,}25 = 2\pi$



Tsol	Tsid
$2\pi + \alpha$	2π

(voir encadré *)

Remarques : Autres satellites terrestres

- Satellites en orbite basse, de forte inclinaison (survolant quasiment les pôles) : 300km < z < 2000km T= 1 à 2h. Station spatiale internationale, satellites de communication, observation terrestre.
- Satellites en orbite moyenne, d'orbites inclinées d'environ 50° : z ≈ 20 000 km T= 12h. Satellites de navigation (GPS)

IV Script python : Satellite soumis à la force d'interaction gravitationnelle, tracé des trajectoires.

1.) Principe

On écrit la loi fondamentale de la dynamique appliquée au satellite dans le référentiel géocentrique galiléen

LFD : $m\vec{a} = \vec{F}_{m_0 \to m} = -\dfrac{G m_0 m}{r^2}\vec{e_r}$

$m_0 = m_T$

$\vec{a} = -\dfrac{G m_T}{r^2}\vec{e_r} = -\dfrac{G m_T}{r^3} \times r\,\vec{e_r}$

$\vec{O\pi} = \vec{r} = r\vec{e_r} \Rightarrow \boxed{\vec{a} = -\dfrac{G m_T}{r^3}\vec{r}}$

Sur (Ox) $a_x = \ddot{x} = -\dfrac{G m_T}{r^3} x$

$a_y = \ddot{y} = -\dfrac{G m_T}{r^3} y$

À résoudre par la méthode d'Euler

$\dot{x} = \dfrac{x_{t+1} - x_t}{h} \Rightarrow \boxed{x_{t+1} = x_t + \dot{x} h}$

où $h = t_{+1} - t$

$\ddot{x} = \dfrac{\dot{x}_{t+1} - \dot{x}_t}{h} \Rightarrow \dot{x}_{t+1} = \dot{x}_t + \ddot{x} h$

$\boxed{\dot{x}_{t+1} = \dot{x}_t - \dfrac{G m_T}{r^3} x h}$

De même : $\boxed{y_{t+1} = y_t + \dot{y} h}$ et $\boxed{\dot{y}_{t+1} = \dot{y}_t - \dfrac{G m_T}{r^3} y h}$

$r^2 = y^2 + x^2$

2.) Mise en oeuvre

```
##Résolution d'une équation différentielle avec Euler
#on prendra G.M=1 et ro=1 soit xo=1 et yo=0, vox=0 et voy=vo
#voc=1 pour la trajectoire circulaire, vop=1,414 pour la trajectoire parabolique

import matplotlib.pyplot as plt
import numpy as np
from scipy.integrate import odeint
```

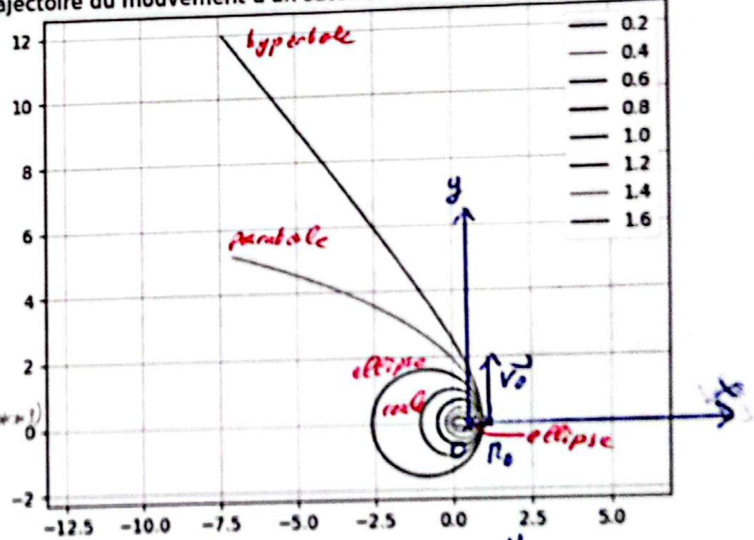
bornes — CI — nombre d'itération

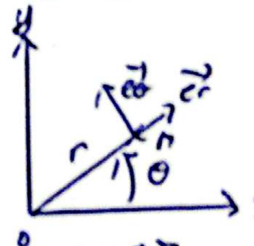
```
def eq_euler(a,b,xo,yo,vxo,vyo,n):
    x,y,vx,vy=xo,yo,vxo,vyo
    r=(x**2+y**2)**0,5
    t,h=a,(b-a)/n   # pas
    les_t,les_x,les_y=[t],[x],[y]
    les_vx,les_vy=[vx],[vy]
    for k in range(n):
        t=t+h      r=(x**2+y**2)**0.5
        x,y=x+h*vx,y+h*vy
        vx,vy=vx-h*x/(r**3),vy-h*y/(r**3)
        les_t.append(t)
        les_x.append(x)
        les_y.append(y)
        les_vx.append(vx)
        les_vy.append(vy)
```

attention r doit être recalculé dans la boucle

```
    return(les_t,les_x,les_y,les_vx,les_vy)
```

trajectoire du mouvement d'un satellite en fonction de la vitesse initiale





$r = O\pi$

$\vec{r} = r\vec{e_r}$

```
#on crée une liste de n=8 valeurs pour vo, entre 0.2 à 1.6, 0 posant pb.
n=8
les_vo=np.linspace(0.6,1.6,n)

for k in range(n):
        xo,yo,vxo,vyo=1,0,0,les_vo[k]
        les_t,les_x,les_y,les_vx,les_vy=eq_euler(0.5,xo,yo,vxo,vyo,10000)
        plt.plot(les_x,les_y,label=str(round(les_vo[k],2)))
plt.legend()
plt.grid()
plt.axis("equal")
plt.title("trajectoire du mouvement d'un satellite en fonction de la vitesse initiale")

plt.show()

##Résolution d'une équation différentielle avec odeint
#définition de la liste des temps : on prendra 10 000 points entre 0 et 15s
les_t = np.linspace(0, 15,10000)

##on définit une fonction eq_mouvement d'arguments z et les_t, qui renvoie vx, vy, ax, ay où ax et ay sont
données par les équations différentielles du mouvement, avec x,y,vx,vy=z
def eq_mouvement(z,les_t):
        x,y,vx,vy=z    # CI
        r=(x**2+y**2)**0.5
        ax=-x/(r**3)   # G=1, mT=1
        ay=-y/(r**3)
        return (vx,vy,ax,ay)

#on crée une liste de n=8 valeurs pour vo, entre 0.2 à 1.6, 0 posant pb.
n=8
les_vo=np.linspace(0.2,1.6,n)
                                    CI
#sol=odeint(eq_mouvement, (x0,y0,vx0,vy0), les_t) donne un tableau numpy.
#un element du tableau est formée d'une liste comprenant les positions et les vitesses
#on récupère les deux premières colonnes (via la commande sol[:, 0] pour la colonne 1)
#on trace les trajectoires correspondant aux différentes valeurs de v0.

for k in range(n):
(x0,y0,vx0,vy0)=(1,0,0,les_vo[k])
sol=odeint(eq_mouvement, (x0,y0,vx0,vy0), les_t)
X=sol[:,0]#on récupère toute la première colonne
Y=sol[:,1]#on récupère toute la deuxième colonne
plt.plot(X,Y,label=str(round(les_vo[k],2)))


plt.legend()
plt.grid()
plt.axis("equal")
plt.title("trajectoire du mouvement d'un satellite en fonction de la vitesse initiale")

plt.show()
```