Cahier des charges — Projet UE Informatique L3 Physique

Relativité Générale - Étude du mouvement d'une étoile autour d'un trou noir de Schwarzschild

Klara Piotrowska, Laly Boyer

October 9, 2024

Contents

1 Introduction

Le but de notre projet est de décrire la trajectoire d'une étoile autour d'un trou noir pour différentes géométrie (2D et 3D), et étudier les conséquences physiques du trou noir sur l'étoile. Pour cela, nous utiliserons les outils mathématiques de la relativité générale (que nous décrirons par la suite) pour résoudre numériquement les équations du mouvement. De même, nous souhaitons également étudier les effets du trou noir sur la lumière, en particulier le décalage gravitationnel vers le rouge et l'inflexion des rayons lumineux.

2 Objectifs principaux

- Étude de la trajectoire de l'étoile autour du trou noir en fonction des conditions initiales imposées sur le systèmes (géodésique sur surface 2D courbe)
- Représentation en 3D de la trajectoire de l'étoile autour du trou noir
- Représentation du mouvement elliptique (Newton) et du mouvement du périhélie,
- Inflexion des rayons lumineux
- Décalage gravitationnel de la lumière vers le rouge
- Effet du trou noir sur l'étoile (temps propre beaucoup plus lent, déformation si c'est trop proche ??)

3 Les outils mathématiques pour la relativité générale

Afin de procéder à une résolution numérique de ce problème dans le cadre de la relativité générale, on commence par définir les équations qui nous permettent de caractériser l'espace en question.

On veut effectuer une simulation de l'espace définie par la métrique de Schwarzschild (en coordonnées sphériques).

Soit la métrique de Schwarzschild définie telle que:

$$ds^2 = -\left(1 - \frac{2GM}{c^2 r}\right) dt^2 + \left(1 - \frac{2GM}{c^2 r}\right)^{-1} dr^2 + r^2\left(d\theta^2 + \sin(\theta) d\varphi^2\right)$$

Puisqu'on se place dans un plan défini par θ constant et égale à $\frac{\pi}{2}$, on peut simplifier l'expression de l'intervalle d'espace-temps:

$$ds^2 = -\left(1 - \frac{2GM}{c^2 r}\right) dt^2 + \left(1 - \frac{2GM}{c^2 r}\right)^{-1} dr^2 + r^2 d\varphi^2$$

Pour simuler numériquement un tel espace, nous avons d'abord besoin d'établir les équations géodésiques, qui vont définir le chemin le plus court entre deux points de l'espace. Pour cela, on résout les équations d'Euler Lagrange associées au système. Le lagrangien est défini comme :

$$\mathbf{L} = \frac{1}{2}\left(\left(1 - \frac{2GM}{c^2 r}\right) c^2 \dot{t}^2 - \left(1 - \frac{2GM}{c^2 r}\right)^{-1} \dot{r}^2 - r^2 \dot{\varphi}^2\right)$$

Résolution des équations EL :

- t et φ sont des variables cycliques qui donnent les deux constantes de mouvement E et L telles que :

$$\left(1 - \frac{2GM}{c^2 r}\right)\dot{t} = E = \text{cte} = \text{énergie totale de la particule conservée} \tag{1}$$

$$r^2\dot{\varphi} = L = \text{cte} = \text{moment cinétique conservé} \tag{2}$$

- sur r :

$$\frac{d\dot{r}}{d\tau} = -\frac{GM}{c^2 r^2}\frac{E^2}{1 - \frac{GM}{c^2 r}} + \frac{L^2}{r^3}\left(1 - \frac{GM}{c^2 r}\right) + \frac{GM}{c^2 r^2}\left(1 - \frac{GM}{c^2 r}\right)^{-1}\dot{r}^2$$

qu'on peut réécrire dans le système d'unités géométrique ($c = G = 1$) comme:

$$\frac{d\dot{r}}{d\tau} = -\frac{2M}{r^2}\mathbf{L} + \frac{L^2}{r^3}\left(1 - \frac{3M}{r}\right)$$

Ensuite, on choisit de fixer $\mathbf{L}$ à $\frac{1}{2}$ dans le cas du mouvement des particules massives, et à 0 dans le cas de la lumière. On a donc :

- Pour la lumière ($\mathbf{L} = 0$) :

$$\frac{d\dot{r}}{d\tau} = \frac{L^2}{r^3}\left(1 - \frac{3M}{r}\right) \tag{3}$$

- Pour une particule massive ($\mathbf{L} = \frac{1}{2}$):

$$\frac{d\dot{r}}{d\tau} = -\frac{M}{r^2} + \frac{L^2}{r^3}\left(1 - \frac{3M}{r}\right) \tag{4}$$

4 Application — Recherche de trajectoire

Nous avons ainsi établi les équations différentielles qui déterminent le mouvement dans l'espace; il nous est donc possible de **simuler la trajectoire d'objets célestes pour différentes conditions initiales**, à l'aide des modules `numpy`, `scipy` et `matplotlib`.

Pour cela, nous utiliserons les **équations géodésiques** (3, 4) , obtenus grâce au lagrangien, ainsi que les **équations de conservation** (1, 2) afin d'incrémenter les variables $(r, \varphi, t, \dot{r})$.

Pour résoudre les équations différentielles, nous utiliserons la fonction `odeint`. Après avoir résolu le système, il sera possible de convertir les coordonnées sphériques en coordonnées cartésiennes et tracer la trajectoire dans le plan 2D, et éventuellement dans l'espace 3D si on se limite pas au cas θ constant.

En faisant varier les différents paramètres (masse M, position ou vitesse initiale $\vec{r}_0$, $\dot{\vec{r}_0}$, ...), on pourrait dans un premier temps chercher à établir des trajectoires elliptiques, mouvement du périhélie, les orbites stables et instables. Puis, on pourrait également étudier le comportement de la lumière proche de l'horizon des événements (effet de lentille, décalage vers le rouge...).

Conditions initiales Les conditions initiales données en input $(r_0, \varphi_0, V_{r_0}, V_{\varphi_0})$ sont définies dans un référentiel d'observateur distant du système *(dans un temps ultérieur on pourrait tester également le cas où les C.I et les représentations graphiques concernent le référentiel propre, a voir si cela présente un intérêt réel).*

Pour pouvoir effectuer le calcul, on doit convertir les vitesses initiales en les dérivant par rapport au temps propre τ. On établit la relation suivante:

$$\left(\frac{dr_0}{dt}, \frac{d\varphi_0}{dt}\right) = \left(\frac{\dot{r}_0}{\dot{t}_0}, \frac{\dot{\varphi}_0}{\dot{t}_0}\right)$$

On peut alors calculer les conditions initiales dans le référentiel propre.

Pour une particule massive (cas $\mathbf{L} = \frac{1}{2}$) :

$$\mathbf{L} = \left[\left(1 - \frac{2M}{r_0}\right) - \left(1 - \frac{2M}{r_0}\right)^{-1} V_{r_0}^2 - r^2 V_{\varphi_0}^2\right] t_0^2$$

Les conditions initiales dans le référentiel propre valent donc :

$$\begin{aligned}
\frac{dt_0}{d\tau} &= \left(\left(1 - \frac{2M}{r_0}\right) - \left(1 - \frac{2M}{r_0}\right)^{-1} V_{r_0}^2 - r^2 V_{\varphi_0}^2\right)^{\frac{1}{2}} \\
\frac{dr_0}{d\tau} &= V_{r_0} \dot{t}_0 \\
\frac{d\varphi_0}{d\tau} &= V_{\varphi_0} \dot{t}_0
\end{aligned}$$

On utilise alors ces conditions pour calculer les quantités conservées E et L:

$$E = \left(1 - \frac{2M}{r_0}\dot{t}_0\right) \iff E = \left(1 - \frac{2M}{r_0}\right) \cdot \left(\left(1 - \frac{2M}{r_0}\right) - \left(1 - \frac{2M}{r_0}\right)^{-1} V_{r_0}^2 - r^2 V_{\varphi_0}^2\right)^{\frac{1}{2}}$$

$$L = r_0^2 V_{\varphi_0} \dot{t}_0 \iff L = r_0^2 V_{\varphi_0} \left(\left(1 - \frac{2M}{r_0}\right) - \left(1 - \frac{2M}{r_0}\right)^{-1} V_{r_0}^2 - r^2 V_{\varphi_0}^2\right)^{\frac{1}{2}}$$

Système différentiel Au final, on obtient à l'entrée: $(r_0, \varphi_0, \dot{r}_0, \dot{\varphi}_0)$. On peut exprimer le système différentielle par 4 relation d'incrémentation à utiliser dans `odeint` :

$$\begin{aligned}
&\frac{dt}{d\tau} = E\left(1 - \frac{2M}{r}\right)^{-1} && \rightarrow \text{increment } t \\
&\frac{dr}{d\tau} = \dot{r} && \rightarrow \text{increment } r \\
&\frac{d\varphi}{d\tau} = \frac{L}{r^2} && \rightarrow \text{increment } \varphi \\
&\frac{d\dot{r}}{d\tau} = -\frac{M}{r^2} + \frac{L^2}{r^3}\left(1 - \frac{3M}{r}\right) && \rightarrow \text{increment } \dot{r}
\end{aligned}$$

5 Structure des objets

Pour réaliser ce projet, nous allons créer plusieurs classes qui vont nous permettre de modéliser les différents objets célestes et de réaliser les calculs nécessaires.

```
class corps_celeste:
    def __init__(self, masse, position, vitesse):
        self.masse = masse
        self.position = position
        self.vitesse = vitesse

    def geodesique(self):
        #calcul des conditions initiales
        #resolution du systeme differentiel
        #conversion des coordonnees
        #representation graphique

class etoile(corps_celeste):
    def __init__(self, masse, position, vitesse):
        super().__init__(masse, position, vitesse)
        self.type = "etoile"
        self.geodesique()

class trou_noir(corps_celeste):
    def __init__(self, masse, position, vitesse):
        super().__init__(masse, position, vitesse)
        self.type = "trou noir"
        self.geodesique()

class geometry:
    def __init__(self):
        #conversion coordonnees
        #representation graphique

    def espace_2D(self):
        #representation 2D de la trajectoire

    def espace_3D(self):
        #representation 3D de la trajectoire

class calc:
    def __init__(self):
        #calculs geodesiques
        #calculs redshift
        #calculs de l'effet du trou noir sur l'etoile
        #representation graphique

    def geodesique(self):
        #systeme differentiel
        #resolution du systeme differentiel
        #conversion des coordonnees
        #representation graphique
    def redshift(self):
        #calcul du redshift
        #representation graphique
    def effet_trou_noir(self):
        #calcul de l'effet du trou noir sur l'etoile
        #representation graphique
```

6 Références

- https://media4.obspm.fr/public/M2R/appliquettes/trounoir/geodesiques_doc.html
- https://www-fourier.ujf-grenoble.fr/~faure/enseignement/projets_simulation/relativite_trou_noir/notes_trou_noir.pdf