FACULTAD
DE INGENIERIA
Universidad de Buenos Aires

TP2-Oumuamua

[75.12] Análisis Numérico
Segundo cuatrimestre 2022

Padron	Alumno
105226	Franco Gentile
100589	Verónica Beatriz Leguizamón

Índice

1. Introducción

El objetivo de este informe es hallar un objeto interestelar que esta perdido en el espacio. Para hacerlo, lanzaremos una sonda desde la tierra y esta será la encargada de encontrarlo.
Este trabajo se realizara a través de la modelación de los distintos fenómenos físicos, como por ejemplo, la trayectoria de la tierra alrededor del sol, la trayectoria del objeto interestelar y la trayectoria de la sonda.
Esto lo haremos a través de la utilización de métodos numéricos que permiten la resolución de ecuaciones diferenciales, utilizando determinadas condiciones iniciales.

2. Desarrollo

2.1. Proceso de Discretización

A continuación expondremos el proceso de discretización para el método de Euler Explicito y para el método de Runge Kutta 2
Paso 1) Discretización del tiempo:

$$t \to t^n \quad n = 0, 1, 2, ...$$
$$t^n = nh$$

Paso 2) Discretizar la EDO:
Notación

$$v_x(t^n) = v_x^n \approx u^n$$
$$v_y(t^n) = v_y^n \approx v^n$$
$$x(t^n) = x^n \approx w^n$$
$$y(t^n) = y^n \approx z^n$$

$$1)\frac{dv_x}{dt} = G\frac{m_s}{d_s^2}cos\alpha_{fs} = f(v_x, t)$$
$$2)\frac{dv_y}{dt} = G\frac{m_s}{d_s^2}sen\alpha_{fs} = f(v_y, t)$$
$$3)\frac{dx}{dt} = v_x = f(x, t)$$
$$4)\frac{dy}{dt} = v_y = f(y, t)$$

Discretización del termino fuente:

$$f(v_x, t) \to f(u^n, t)$$
$$f(v_y, t) \to f(v^n, t)$$
$$f(x, t) \to f(w^n, t)$$
$$f(y, t) \to f(z^n, t)$$

En el presente trabajo práctico uno de los métodos utilizados fue Euler explícito cuya expresión resulta $u_{n+1} = u_n + hf(u^n, t^n)$. Por lo tanto las ecuaciones utilizadas fueron las siguientes:

1. $u_{n+1} = u_n + h\left[G\frac{m_s}{(x_s-w^n)^2+(y_s-z^n)^2}cos(atg(\frac{y_s-z^n}{x_s-w^n})\right]$
2. $v_{n+1} = v_n + h\left[G\frac{m_s}{(x_s-w^n)^2+(y_s-z^n)^2}sen(atg(\frac{y_s-z^n}{x_s-w^n})\right]$
3. $w_{n+1} = w_n + hu_n$

4. $z_{n+1} = z_n + hv_n$

Por otro lado también implementamos Euler modificado, Runge Kutta de orden 2, cuyas expresiones son:

- Predictor: $u^{*(n+1)} = u^n + hf(u^n, t^n)$
- Corrector: $u^{n+1} = u^n + \frac{h}{2}\left[f(u^n, t^n) + f(u^{*(n+1)}, t^{*(n+1)})\right]$

Entonces discretizando las ecuaciones diferenciales brindadas obtuvimos las siguientes expresiones:

Predictores:

1. $u^{*(n+1)} = u^n + h[G\frac{m_s}{(x_s-w^n)^2+(y_s-z^n)^2}cos(atg(\frac{y_s-z^n}{x_s-w^n})]$
2. $v^{*(n+1)} = v^n + h\left[G\frac{m_s}{(x_s-w^n)^2+(y_s-z^n)^2}sen(atg(\frac{y_s-z^n}{x_s-w^n})\right]$
3. $w^{*(n+1)} = w^n + h\left[u^n\right]$
4. $z^{*(n+1)} = z^n + h\left[v^n\right]$

Correctores:

1. $u^{n+1} = u^n + \frac{h}{2}\left[G\frac{m_s}{(x_s-w^n)^2+(y_s-z^n)^2}cos(atg(\frac{y_s-z^n}{x_s-w^n}) + G\frac{m_s}{(x_s-w^{*(n+1)})^2+(y_s-z^{*(n+1)})^2}cos(atg(\frac{y_s-z^{*(n+1)}}{x_s-w^{*(n+1)}})\right]$
2. $v^{n+1} = v^n + \frac{h}{2}\left[G\frac{m_s}{(x_s-w^n)^2+(y_s-z^n)^2}sen(atg(\frac{y_s-z^n}{x_s-w^n}) + G\frac{m_s}{(x_s-w^{*(n+1)})^2+(y_s-z^{*(n+1)})^2}cos(atg(\frac{y_s-z^{*(n+1)}}{x_s-w^{*(n+1)}})\right]$
3. $w^{(n+1)} = w^n + \frac{h}{2}\left[u^n + u^{*(n+1)}\right]$
4. $z^{(n+1)} = z^n + \frac{h}{2}\left[v^n + v^{*(n+1)}\right]$

Paso 3) Discretización de las condiciones iniciales:

Para el ejercicio 1 tenemos:

$$x_T(0) = d_{ST}\ ;\ Y_T = 0; Vx_T(0) = 0\ ;\ Vy_T(0) = \sqrt{\frac{Gm_s}{d_{st}}}$$

Por lo que;

$$w^0 = d_{ST}\ ;\ z^0 = 0\ ;\ u^0 = 0\ ;\ v^0 = \sqrt{\frac{Gm_s}{d_{st}}}$$

Para el ejercicio 2 tenemos:

$$x_U(0) = 0\ ;\ y_U(0) = -3{,}81E10\ ;\ v_{xU}(0) = 8{,}771E4\ ;\ v_{yU}(0) = 0$$

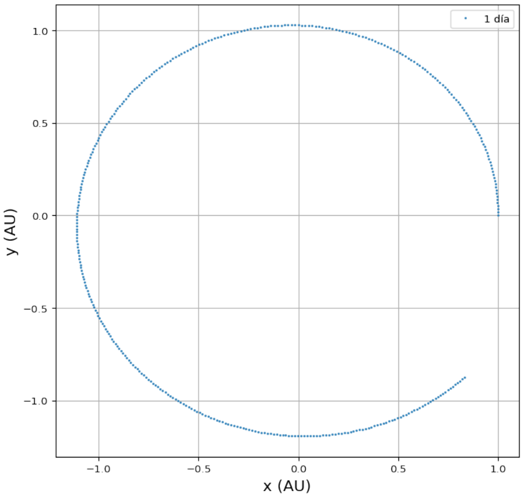
Por lo que:

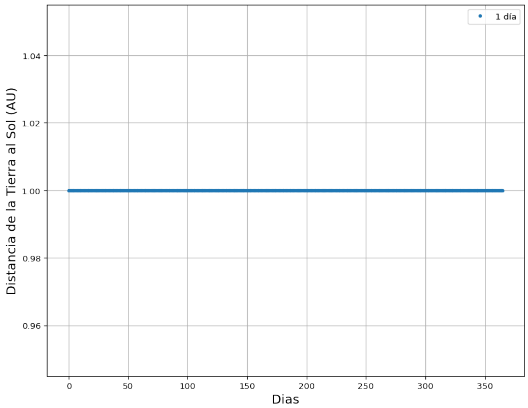
$$w^0 = 0\ ;\ z^0 = -3{,}81E10\ ;\ u^0 = 8{,}771E4\ ;\ v^0 = 0$$

2.2. Análisis de convergencia

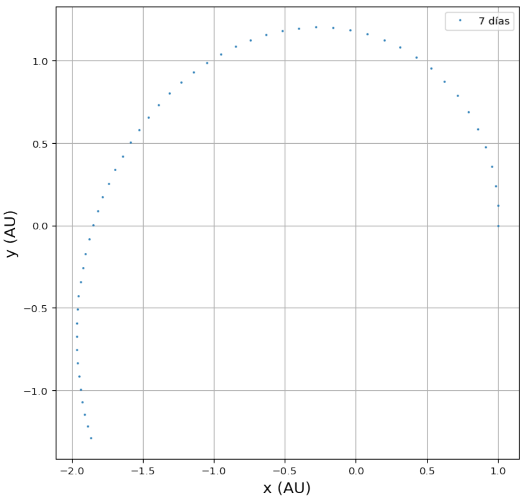
Al momento de analizar la convergencia en ambos métodos sabíamos que por el Teorema de Lax, necesitábamos un modelo que sea consistente y estable, y que a esto lo haríamos buscando la conservación de la distancia entre la tierra y el sol, de la Energía, y observando la trayectoria de la tierra alrededor del sol.
Por lo que, nuestro análisis se baso en el estudio de los gráficos para distintos pasos h.
A continuación, podrá observarse como al ir con h más chicos a 7 días, la trayectoria de la tierra y por ende, su distancia al sol, iban pareciéndose más a lo que nosotros estábamos buscando.

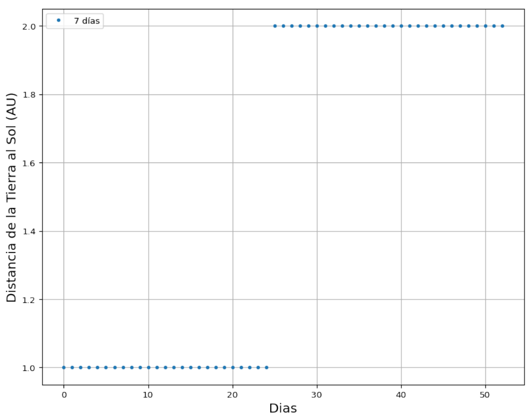
En el caso de Euler Explicito:



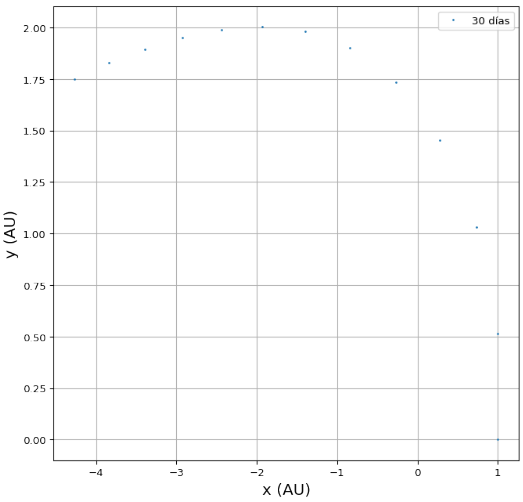


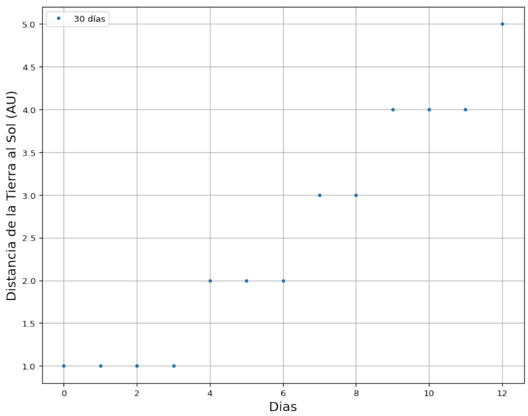
Tomando h = 1 día podemos notar que los resultados son acorde a lo esperado y que la distancia sigue conservándose. Todavía resulta un paso estable.





Esta vez aumentamos h = 7 días. A pesar de que la trayectoria comienza a deformarse en relación a lo que esperábamos, tiene una ligera trayectoria acorde a lo que podría ser una órbita, pero comienza a alejarse. Por otro lado la distancia comienza a variar.

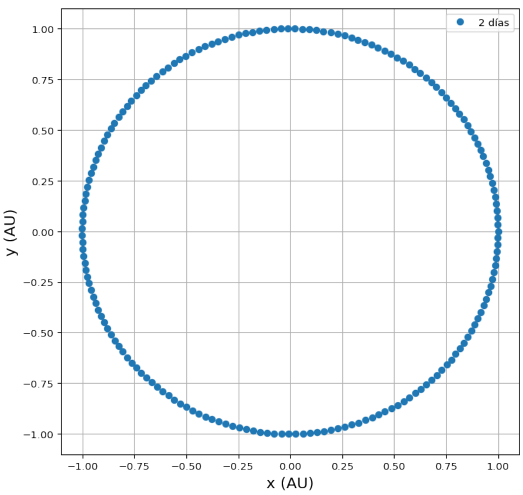


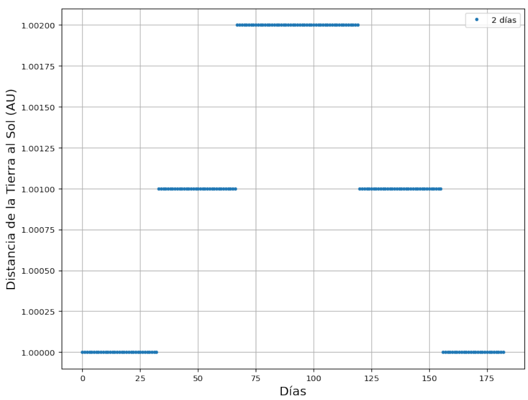


Por último probamos con h = 30 días y pudimos notar que la trayectoria difiere totalmente a lo esperado y la distancia comienza a aumentar considerablemente.

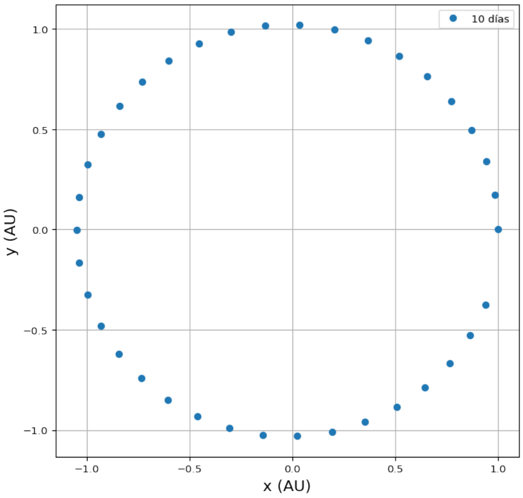
Consideramos que le modelo es estable para $h < 7$ días $= 604800s$.

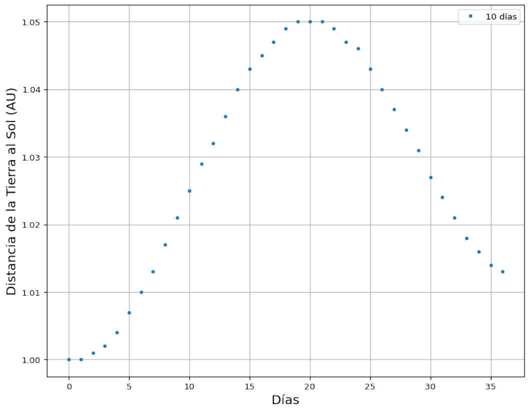
En el caso de Euler Modificado (Runge Kutta 2):



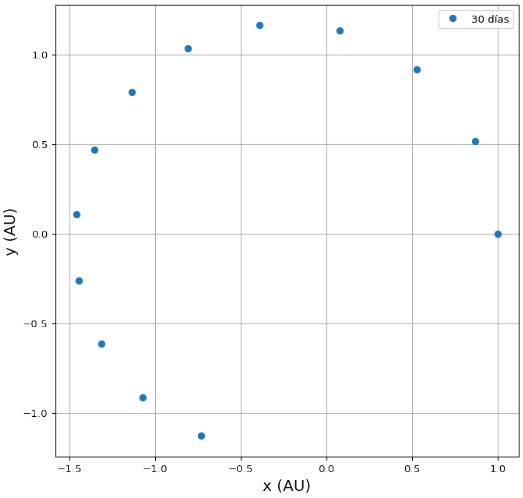


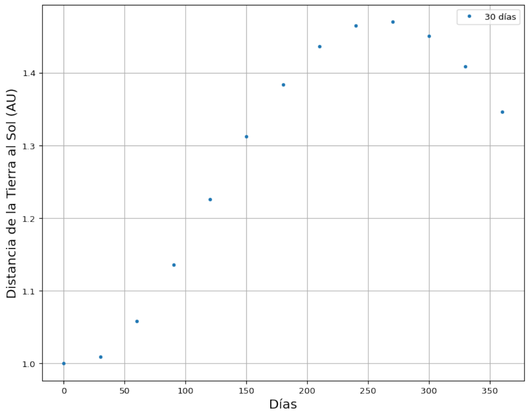
Comenzamos tomando h = 2 días. La trayectoria tiene relación con lo esperado pero la distancia comienza a variar en su segundo decimal. Sigue siendo estable.



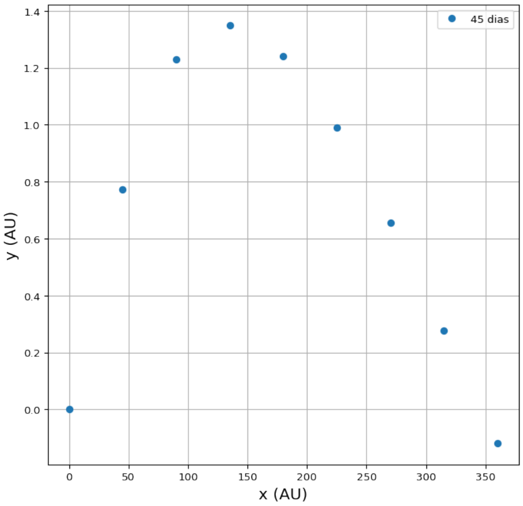


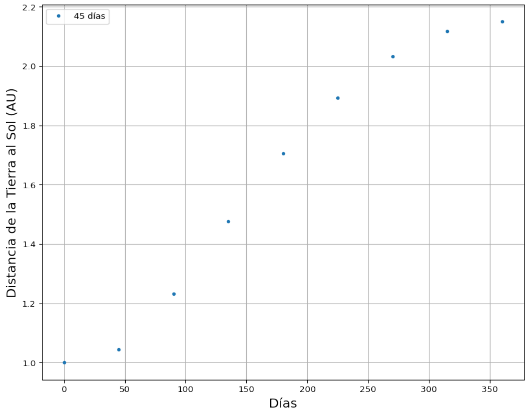
Esta vez aumentamos h = 10 días. La trayectoria sigue dando buenos resultados y la distancia comienza a variar un poco más.





Con h = 30 días notamos que a pesar de que la distancia también varia bastante, la trayectoria de la órbita terrestre sigue dando como resultado algo coherente dentro del contexto del trabajo práctico.





Por último tomamos h = 45 días. La trayectoria de la órbita terrestre ya no es la esperada y la distancia comienza a aumentar bastante.

Llegamos a la conclusión de que el método es estable si tomamos h $<$45 días = 3888000s.

Al trabajar con métodos de la derivada con dos puntos que resulta ser una resolución adecuada para la derivada primera, el método también es consistente. Por lo tanto, por el teorema de LAX, son convergentes.

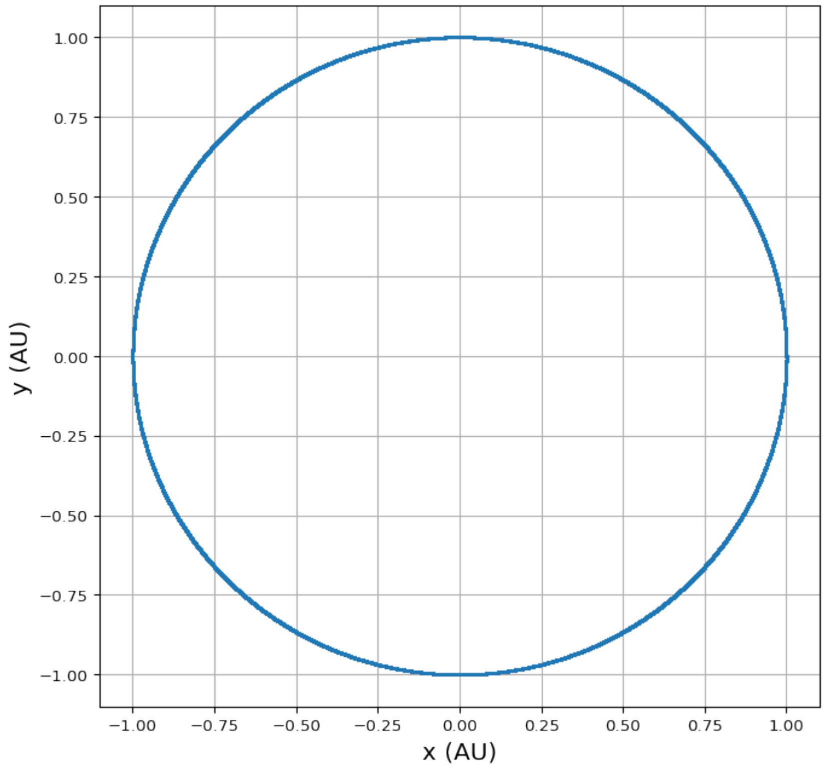
3. Ejercicio 1: Órbita terrestre.

En este apartado nos encargaremos de modelar la trayectoria de la tierra alrededor del sol, hasta que la tierra de una vuelta completa. Resolvimos el ejercicio con ambos métodos considerando el período como $T = 2\pi \frac{d_{st}}{v_{yT}} = 3{,}16e^7 s$ =1 año.

3.1. Resolución con Euler Explicito

Considerando un paso $h = 60$s modelamos la órbita terreste a lo largo de un año obteniendo una nube de puntos como presentamos en el gráfico siguiente.

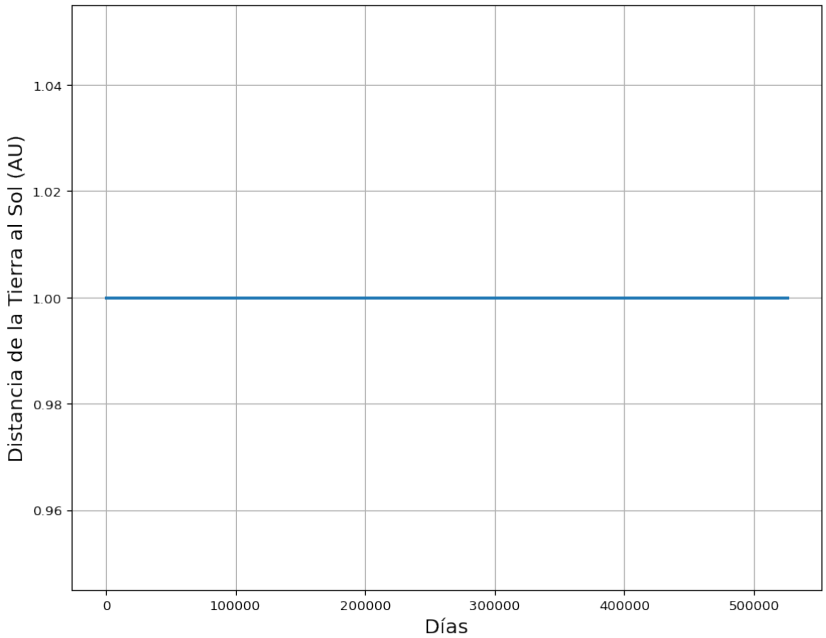
Para resolver el sistema, hicieron falta 525941 iteraciones.



Cada uno de los puntos representa la posición de la tierra en cada minuto del año. Se recuerda que el Sol se encuentra en la posición (0,0). Como se observa se llegó a una órbita circular tal como el presente trabajo práctico propone cuya distancia al Sol, a simple vista, parece ser constante.

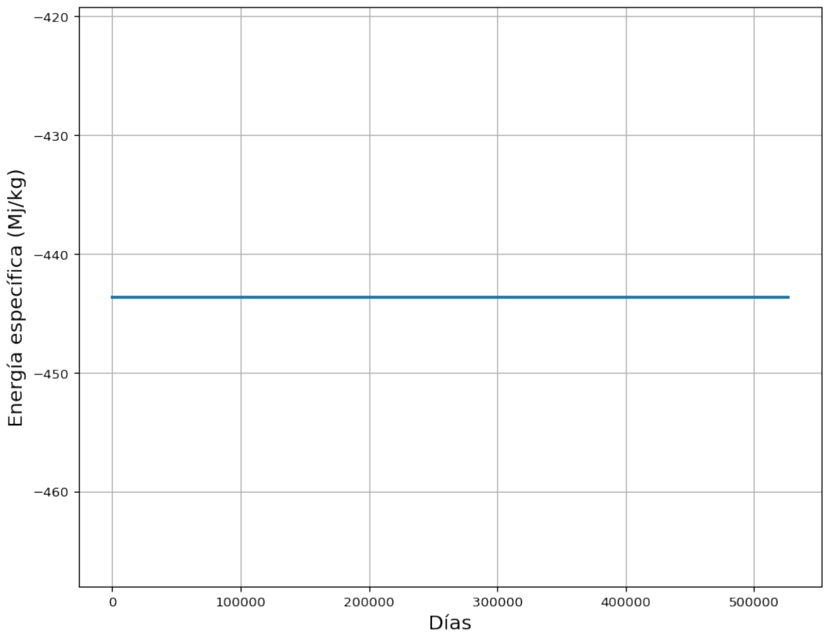
Procedimos a verificar esta distancia.

Para $h = 60s$ calculamos la distancia de la Tierra al Sol.



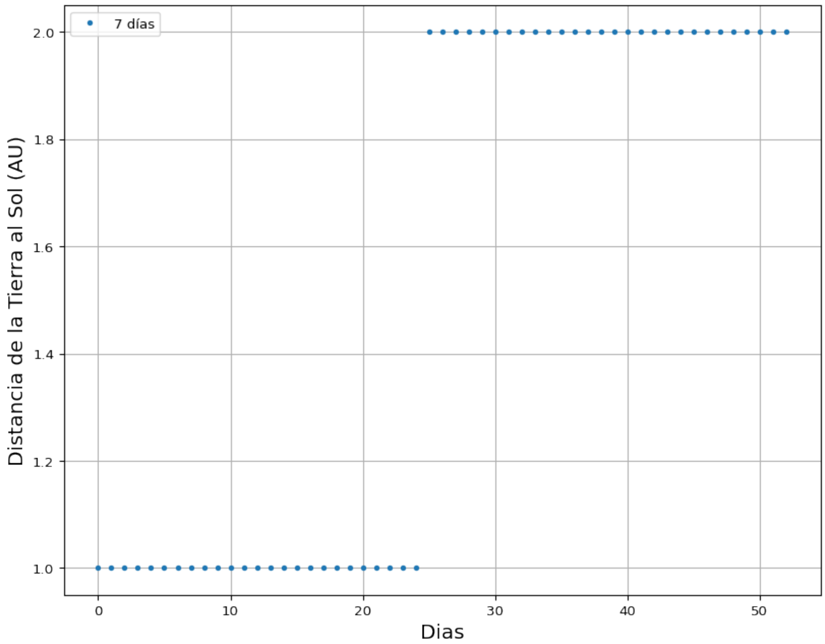
Esta efectivamente se mantiene constante tal y como esperábamos.

Por otra parte, calculamos la energía mecánica específica como: $E_m = \frac{v^2}{2} - G\frac{m_{sol}}{d_{sol}}$ con el paso utilizado para modelar la órbita terreste, obteniendo el gráfico expuesto a continuación.

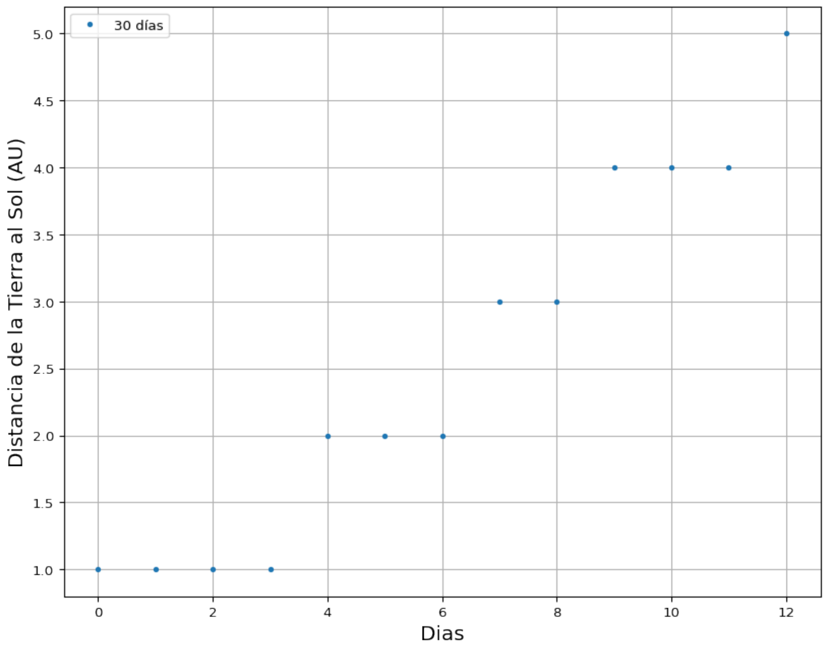


Esta efectivamente también se mantiene constante.

El ejercicio propone analizar qué sucede con la distancia del Sol y la Tierra a medida que se modifica el paso. Primero tomamos $h = 7$ días $= 604800s$ obteniéndose el gráfico siguiente. Podemos notar que la distancia ya no es constante y varía un dígito.



Volvimos a aumentar el paso esta vez tomando $h = 30$ días $= 2592000s$. Se puede apreciar que la distancia comienza a variar bastante más pero recordando, también, que para ese valor de paso el modelo deja de ser estable.

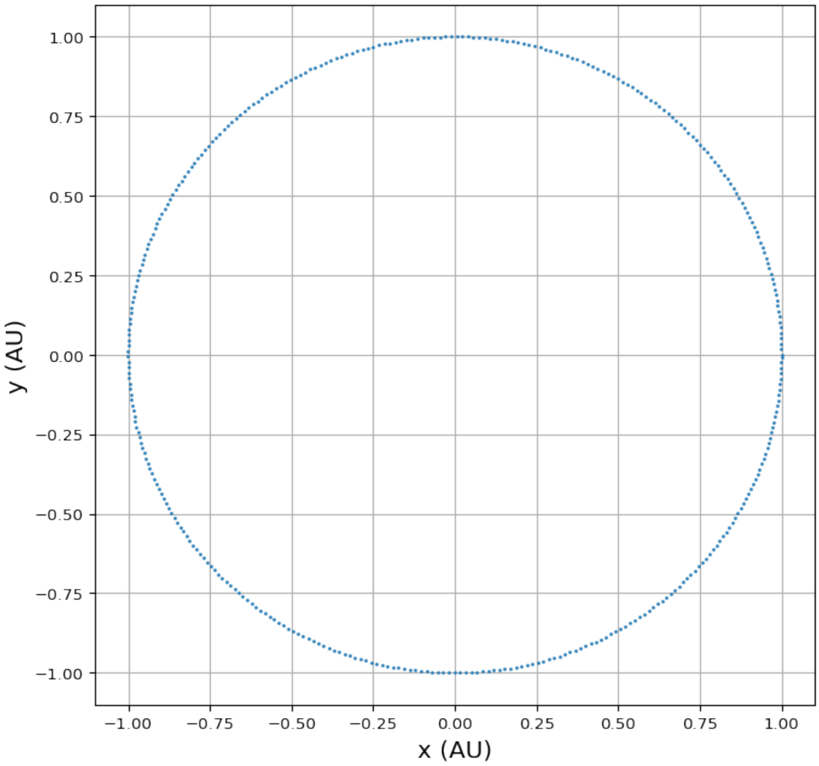


3.2. Resolución con Runge Kutta 2

Al igual de lo realizado con Euler Explícito, modelamos la órbita terrestre pero esta vez con Euler modificado (Runge Kutta de orden 2).

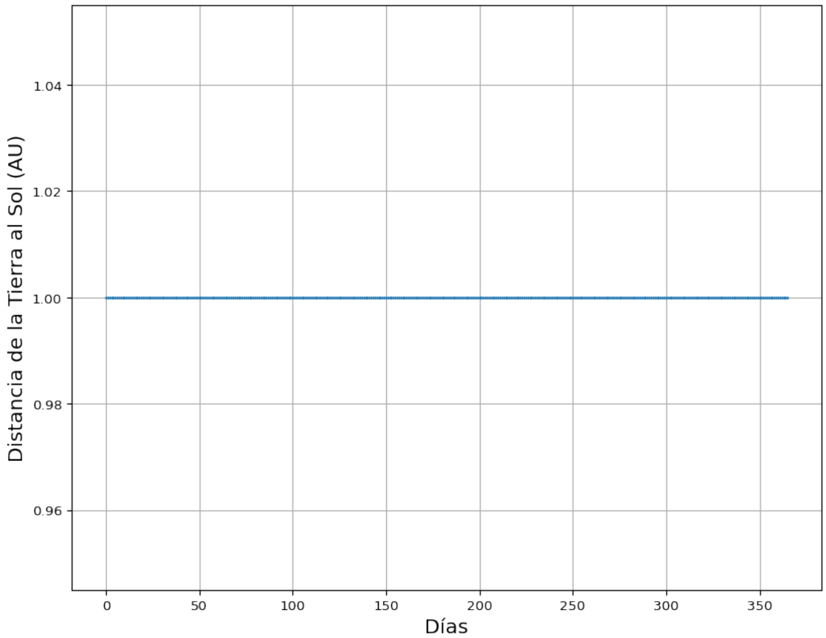
En este caso, para resolver el sistema hicieron falta 366 iteraciones.

Utilizamos $h = 86400$s, es decir 1 día. Completando un mismo período de un año, recordando que cada punto representa la posición de la tierra en cada día del año, obtuvimos el siguiente resultado.

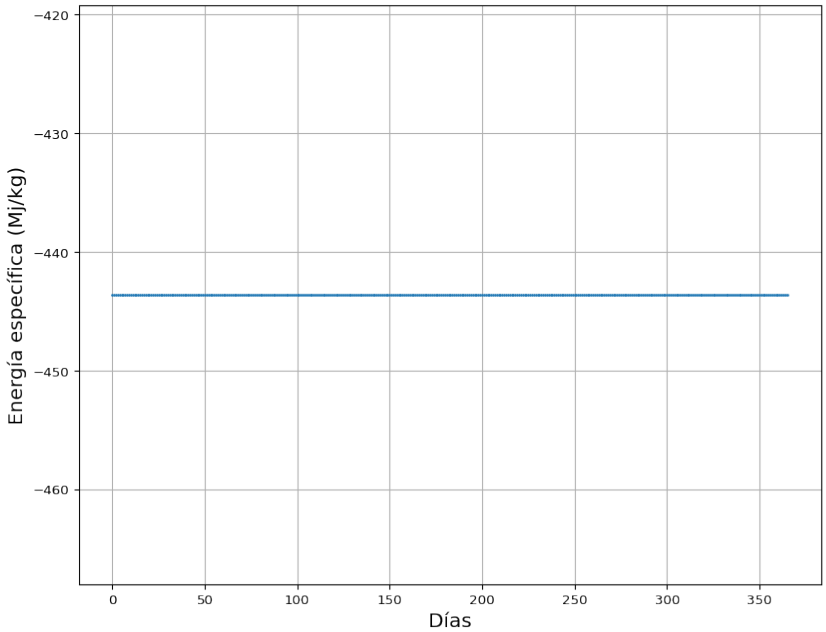


Como se puede observar también la órbita obtenida es un círculo centrado en el origen de coordenadas donde se encuentra el Sol.

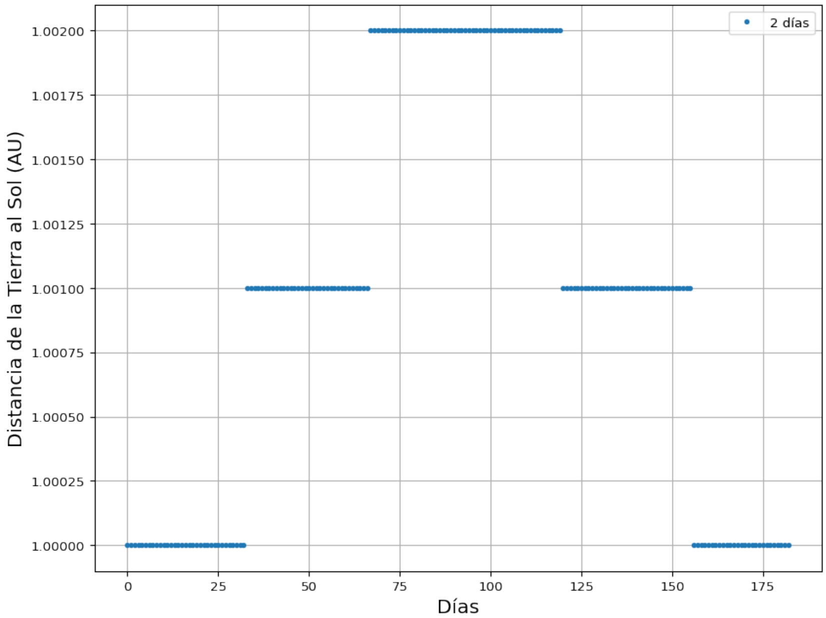
Volvimos a graficar la distancia de la Tierra al Sol, y como puede verse, esta se mantuvo constante.



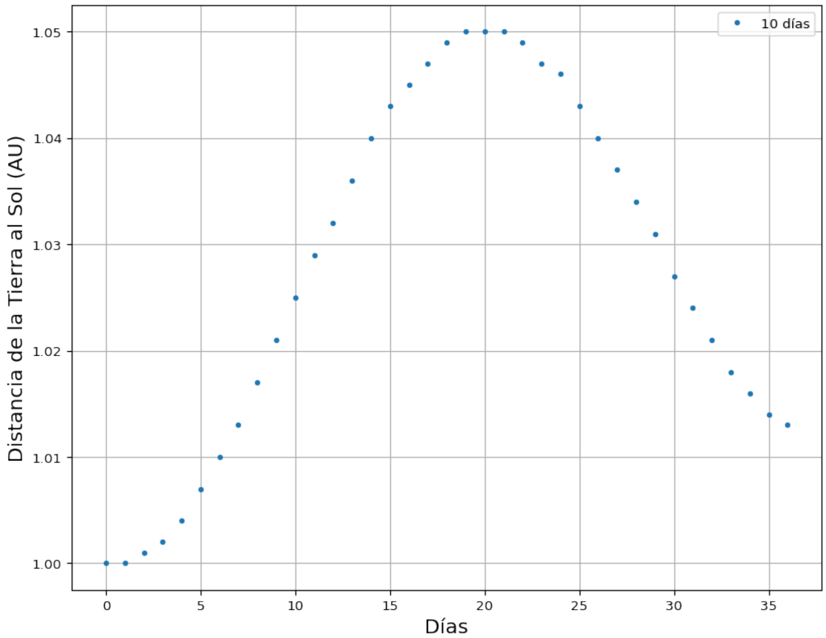
y la Energía también se mantuvo constante.



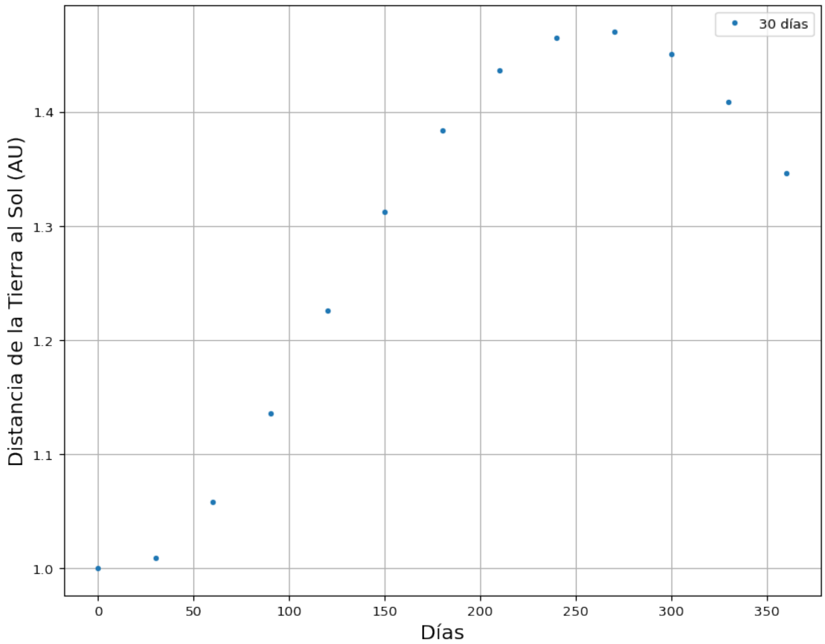
Al igual que con Euler Explícito, analizamos qué sucede con la distancia a medida que agrandamos el paso. Comenzamos calculando $h = 2$ días $= 172800s$. La distancia comienza a variar pero recién en su tercer decimal.



Seguimos agrandando el paso esta vez considerando $h = 10$días $= 864000s$. A pesar de que la distancia ahora tiene solamente un decimal de precisión, sigue obteniéndose un resultado bastante lógico.



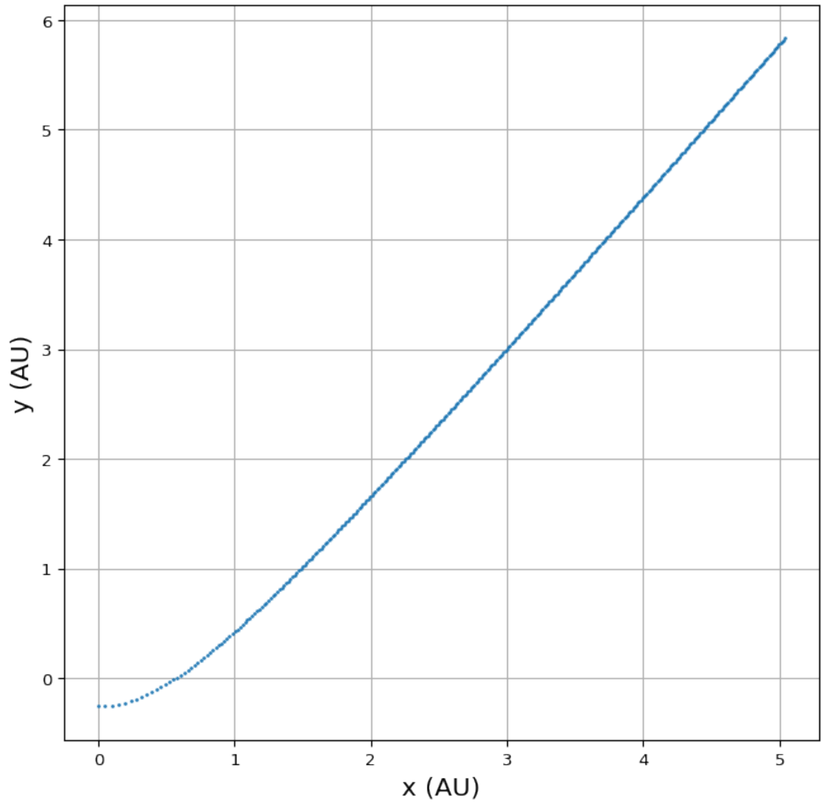
Por último agrandamos $h = 30\text{días} = 2592000s$. La distancia comienza a variar en su primer decimal.



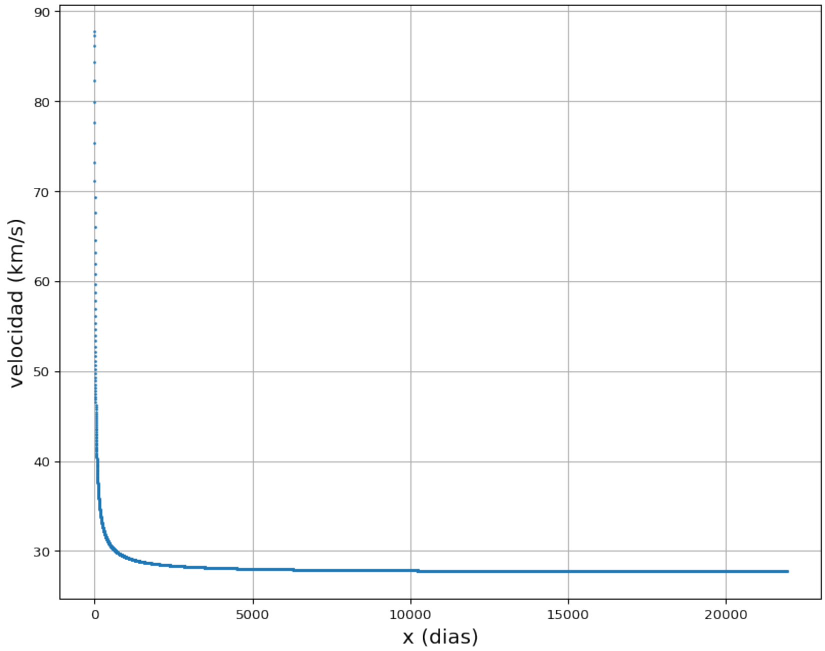
Se debe recordar que no tenemos una precisión infinita al realizar los cálculos y también arrastrar errores producidos al usar la función atan2 de python como también sqrt al calcular la raíz.

4. Ejercicio 2: Trayectoria de Oumuamua.

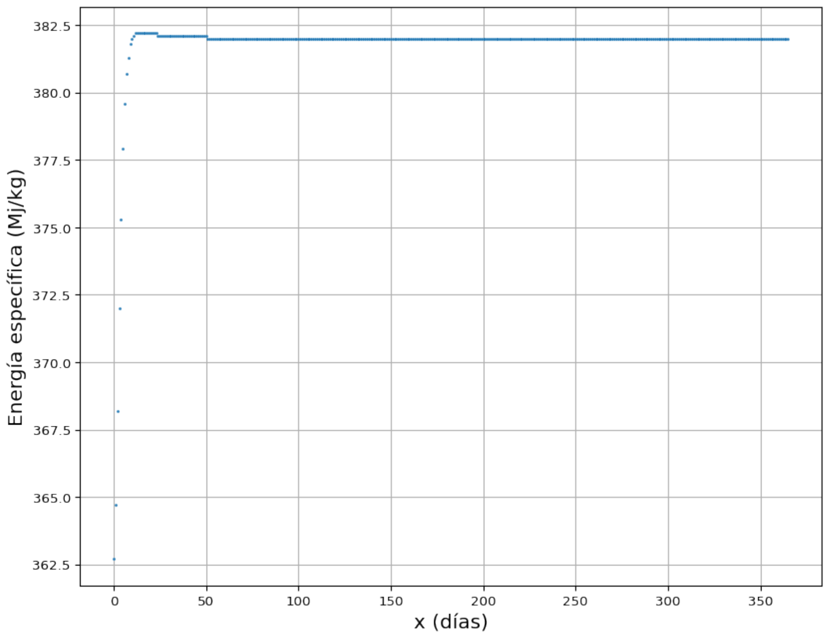
Utilizando Euler Modificado calculamos la trayectoria del objeto no identificado Oumuamua con $h = 86400s$. Elegimos este método al poder modelar las trayectorias con menor costo computacional al tener que realizar pocas iteraciones. Al principio quisimos visualizar la trayectoria en ese primer año que cruza muy cerca del Sol por lo que limitamos el período a nuevamente un año recordando que la estrella de nuestro sistema solar se encuentra en el origen de coordenadas. Obtuvimos el gráfico siguiente.



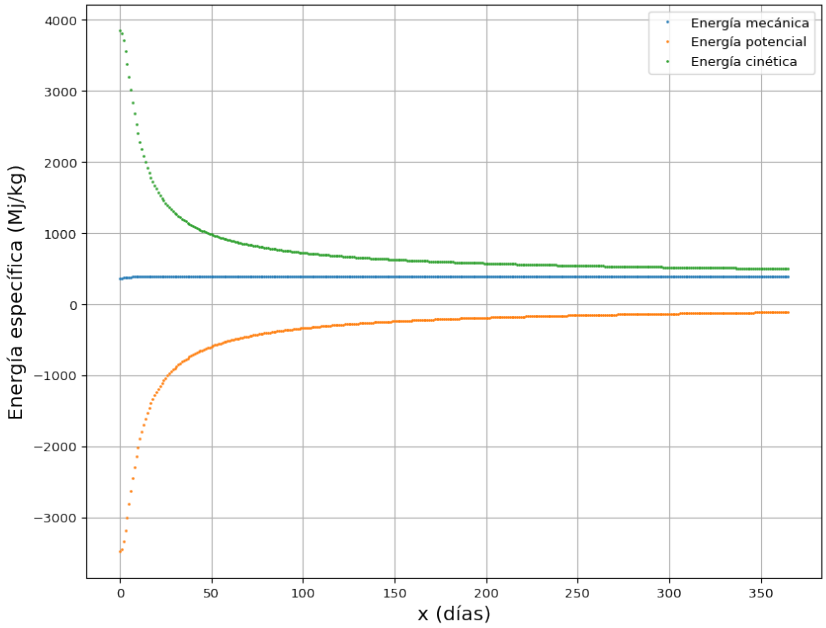
El ejercicio propone extender la simulación hasta que la velocidad de Oumuamua no varíe, en cuyo caso se lo considerará $v_{\infty U}$. Para ello decidimos calcular la velocidad del objeto a lo largo de diferentes períodos de tiempo hasta que decidimos tomar 60 años como nuestro período a analizar. Obtuvimos el gráfico siguiente en donde se puede notar que la velocidad comienza a no variar de forma considerable.



Evaluamos la energía mecánica a lo largo de los sesenta años y pudimos notar que, los primeros años esta no se conserva, pero cuando comienza a alejarse del Sol sí.



Nos centramos en graficar la evolución de las energías intervinientes en el primer año de estudio para poder visualizar mejor lo que ocurre.



Se puede observar que no se conserva la energía mecánica cuando el objeto está cercano al Sol, pero a medida que pasa el tiempo y se aleja del mismo, esta comienza a ser constante.

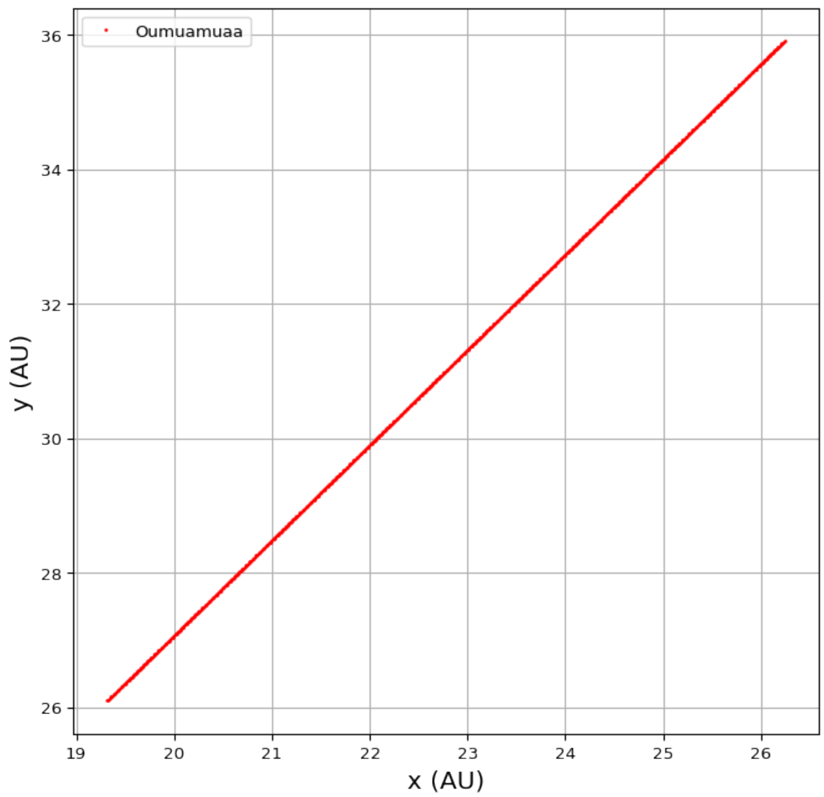
5. Ejercicio 3: Misión de búsqueda.

En este ejercicio nos centramos en la búsqueda del objeto interestelar Oumaumua.

Para cumplir con nuestro objetivo, sabíamos que lanzaríamos la sonda 5 años después de que el objeto interestelar pase por el perihelio solar.

Por lo que nuestro primer paso, fue encontrar la posición y la velocidad de Oumuauma a los 5 años de haberse visto. Posteriormente, usamos ese dato, como condición inicial para modelar el trayecto que iba a realizar posteriormente el objeto.

A continuación, el gráfico que representa la trayectoría de Oumuamua 5 años después de haber visto pasar por el perihelio solar:



Luego, para modelar el trayecto que iba a realizar la sonda, utilizamos las siguientes condiciones iniciales:

$x_u(0) = 0m$; $Y_T = -3{,}81E10m$; $Vx_u(0) = 8{,}771E4m/s$; $Vy_u(0) = 0$

Discretizando queda:

$w_n(0) = 0m$; $z_n = -3{,}81E10m$; $u_n(0) = 8{,}771E4m/s$; $v_n(0) = 0$

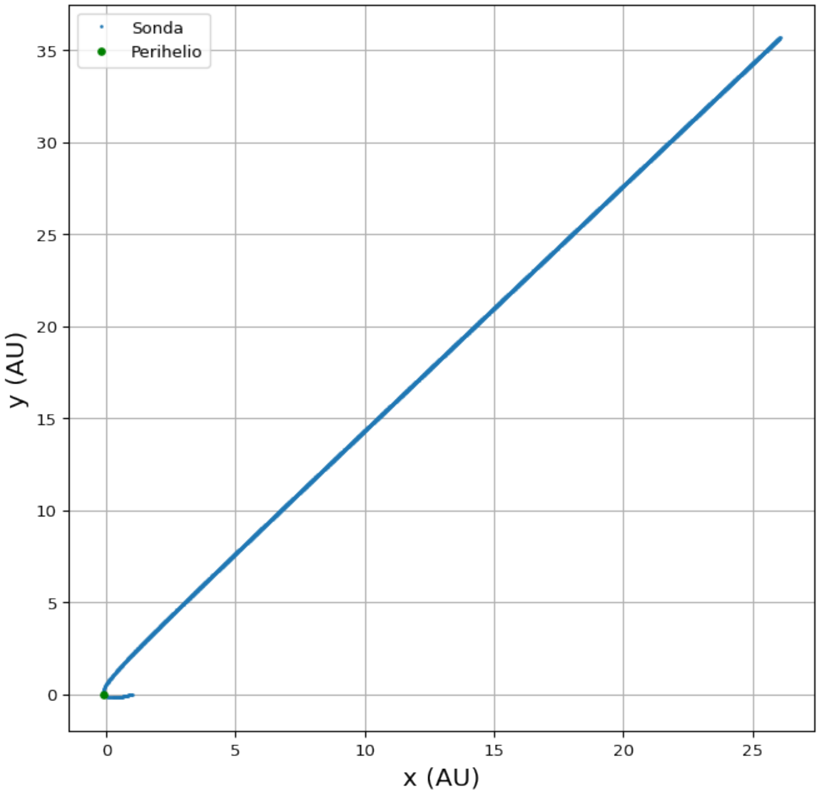
Lo que hicimos fue ir variando la velocidad de escape, el ángulo y la velocidad de impulsión. En función de esto, fuimos buscando la mejor combinación para que el trayecto de la sonda este dirigido hacía la trayectoria de el objeto interestelar.

Como mencionamos anteriormente, a la sonda le imprimimos una velocidad de impulsión, esto se hace, para que esta pueda evadír la fuerza gravitatoria que le hace el sol, y pueda partir hacía el espacio a completar la misión de búsqueda.

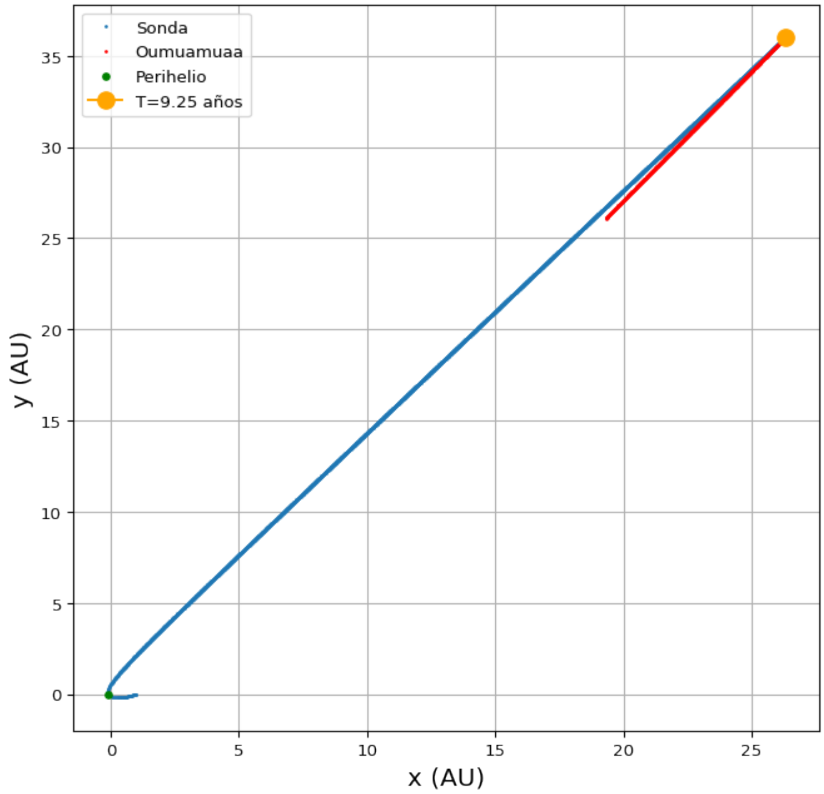
A la sonda debíamos imprimirle la velocidad de impulsión cuando esta llegue al perihelio solar, es decir, cuando este en el punto más cercano al sol, y el sentido de esta velocidad es normal al vector que forma la distancia entre la sonda y el sol.

A continuación, el gráfico que representa la trayectoria de la sonda:

$V_o = 30E3km/s$ $\alpha_{vo} = 205km/s$ $\Lambda V_{impulsin} = 1050km/s$



Una vez programado todo lo mencionado anteriormente, lo que hicimos fue ver en donde se intersecaban ambas trayectorias y cuanto demoraba esto en suceder.



Puede verse en el gráfico que pudimos encontrar el objeto 9.25 años después de que la sonda partió desde la tierra. Es decir 14.25 años después de que se vio al objeto pasar por el perihelio solar.

Para calcular el costo de la misión lo que hicimos fue sumar $V_o + \Lambda V_{impulsin} + \Lambda V_\infty$ síendo V_∞ el frenado de la sonda al llegar a Oumuama. Obteniendo: $V_{cost} = 23{,}7 km/s$

6. Anexo.

Programa utilizando para calcular Euler Explícito.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

G = 6.67e-11
masa_sol = 1.99e30
dist_tierra_sol = 1.496e11
x_sol = 0
y_sol = 0
h = 60

def distancia(x,y):
  return np.sqrt( ((x_sol - x)**2) + ((y_sol - y)**2) )

def alpha(x,y):
  return np.arctan2(-y,-x)

anio = (2*np.pi*dist_tierra_sol)/(np.sqrt((G*masa_sol)/dist_tierra_sol))

def ue_n(u_n, w_n, z_n):
  return u_n + ( h * ( (G * masa_sol * np.cos(alpha(w_n,z_n)) ) / (distancia(w_n,z_n) ** 2) ))

def ve_n(v_n, w_n, z_n):
  return v_n + ( h * ( (G * masa_sol * np.sin(alpha(w_n,z_n))) / (distancia(w_n,z_n) ** 2) ))

def we_n(w_n, u_n):
  return w_n + (h*u_n)

def ze_n(z_n, v_n):
  return z_n + (h*v_n)

# Condiciones iniciales del ejericico 1.
un = [0]
vn = [np.sqrt((G*masa_sol)/dist_tierra_sol)]
wn = [dist_tierra_sol]
zn = [0]

n = 1
t = h

while(t < anio):
  un.append(ue_n(un[n-1],wn[n-1],zn[n-1]))
  vn.append(ve_n(vn[n-1],wn[n-1],zn[n-1]))
  wn.append(we_n(wn[n-1],un[n-1]))
  zn.append(ze_n(zn[n-1],vn[n-1]))
  n = n+1
  t = t+h
```

Programa utilizado para calcular Euler Modificado.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

G = 6.67e-11
masa_sol = 1.99e30
dist_tierra_sol = 1.496e11
x_sol = 0
y_sol = 0
h = 86400

def distancia(x,y):
  return np.sqrt( ((x_sol - x)**2) + ((y_sol - y)**2) )

def alpha(x,y):
  return np.arctan2(-y,-x)

anio = (2*np.pi*dist_tierra_sol)/(np.sqrt((G*masa_sol)/dist_tierra_sol))

# Predictores
def u_asterisco(u_n, w_n, z_n):
  return u_n + ( h * ( (G * masa_sol * np.cos(alpha(w_n,z_n)) ) / (distancia(w_n,z_n) ** 2) ))

def v_asterisco(v_n, w_n,z_n):
  return v_n + ( h * ( (G * masa_sol * (np.sin(alpha(w_n,z_n)))) / (distancia(w_n,z_n) ** 2) ))

def w_asterisco(w_n,u_n):
  return w_n + (h*u_n)

def z_asterisco(z_n, v_n):
  return z_n + (h*v_n)

# Correctores
def u_n(u_n, w_n, z_n, w_an, z_an):
  aux1 = (G * masa_sol * (np.cos(alpha(w_n,z_n)))) / (distancia(w_n,z_n) ** 2)
  aux2 = (G * masa_sol * (np.cos(alpha(w_an,z_an)))) / (distancia(w_an,z_an) ** 2)

  return u_n + ((h/2)*(aux1+aux2))

def v_n(v_n, w_n, z_n, w_an, z_an):
  aux1 = (G * masa_sol * (np.sin(alpha(w_n,z_n)))) / (distancia(w_n,z_n) ** 2)
  aux2 = (G * masa_sol * (np.sin(alpha(w_an,z_an)))) / (distancia(w_an,z_an) ** 2)

  return v_n + ((h/2)*(aux1+aux2))

def w_n(w_n,u_n, u_an):
  return w_n + ((h/2)*(u_n+u_an))

def z_n(z_n,v_n, v_an):
  return z_n + ((h/2)*(v_n+v_an))

# Condiciones iniciales del ejercicio 1.
```

```
un = [0]
vn = [np.sqrt((G*masa_sol)/dist_tierra_sol)]
wn = [dist_tierra_sol]
zn = [0]
n = 1
t= h

while(t < anio):
  u__asterisco = u_asterisco(un[n-1],wn[n-1],zn[n-1])
  v__asterisco = v_asterisco(vn[n-1],wn[n-1],zn[n-1])
  w__asterisco = w_asterisco(wn[n-1],un[n-1])
  z__asterisco = z_asterisco(zn[n-1],vn[n-1])

  un.append(u_n(un[n-1],wn[n-1],zn[n-1],w__asterisco,z__asterisco))
  vn.append(v_n(vn[n-1],wn[n-1],zn[n-1],w__asterisco,z__asterisco))
  wn.append(w_n(wn[n-1],un[n-1],u__asterisco))
  zn.append(z_n(zn[n-1],vn[n-1],v__asterisco))

  n = n+1
  t = t+h
```

Programa utilizado para la misión. Se utiliza Euler Modificado para resolverlo, las funciones de los correctores y predictores están colocados en la sección de arriba.

```
def se_quemo(x,y):
  return distancia(x,y) < dist_minima_sol

def lo_encontramos(x1,x2,y1,y2):
  return (((0.8*x1<x2<1.2*x1) or (0.8*x2<x1<1.2*x2)) and ((0.8*y1<y2<1.2*y1) or (0.8*y2<y1<1.2*y2

def estamos_perihelio(distancia_siguiente, distancia_anterior):
  return distancia_siguiente > distancia_anterior

dist_minima_sol = 0.1 * dist_tierra_sol
v_0 = 30e3
angulo = (205 * np.pi) / 180
v_impulso = 1050

def llego_sonda(distancia_sol_sonda, distancia_sol_ou):
  return distancia_sol_sonda >= distancia_sol_ou

#Condiciones iniciales para Oumuama a los 5 años
#(2889935949388.3105, 3903955766179.713, 16501.13496622091, 23374.49598044698)=(wn,zn,un,vn)
un_o = [16501.13496622091]
vn_o = [23374.49598044698]
wn_o = [2889935949388.3105]
zn_o = [3903955766179.713]

un = [v_0 * np.cos(angulo)]
vn = [v_0 * np.sin(angulo)]
wn = [dist_tierra_sol]
zn = [0]
```

```
n = 1
t= h
se_quemo_bu = False
todavia_no_se_impulso = True
aparecio = False
estamos_en_perihelio = False
anio = (2*np.pi*dist_tierra_sol)/(np.sqrt((G*masa_sol)/dist_tierra_sol))
anio_llego = 0

while(t < anio*9.25):
  #Sonda
  u__asterisco = u_asterisco(un[n-1],wn[n-1],zn[n-1])
  v__asterisco = v_asterisco(vn[n-1],wn[n-1],zn[n-1])
  w__asterisco = w_asterisco(wn[n-1],un[n-1])
  z__asterisco = z_asterisco(zn[n-1],vn[n-1])

  un.append(u_n(un[n-1],wn[n-1],zn[n-1],w__asterisco,z__asterisco))
  vn.append(v_n(vn[n-1],wn[n-1],zn[n-1],w__asterisco,z__asterisco))
  wn.append(w_n(wn[n-1],un[n-1],u__asterisco))
  zn.append(z_n(zn[n-1],vn[n-1],v__asterisco))

  #Oumuamua
  u__asterisco = u_asterisco(un_o[n-1],wn_o[n-1],zn_o[n-1])
  v__asterisco = v_asterisco(vn_o[n-1],wn_o[n-1],zn_o[n-1])
  w__asterisco = w_asterisco(wn_o[n-1],un_o[n-1])
  z__asterisco = z_asterisco(zn_o[n-1],vn_o[n-1])

  un_o.append(u_n(un_o[n-1],wn_o[n-1],zn_o[n-1],w__asterisco,z__asterisco))
  vn_o.append(v_n(vn_o[n-1],wn_o[n-1],zn_o[n-1],w__asterisco,z__asterisco))
  wn_o.append(w_n(wn_o[n-1],un_o[n-1],u__asterisco))
  zn_o.append(z_n(zn_o[n-1],vn_o[n-1],v__asterisco))

  if(se_quemo(wn[n],zn[n])):
    se_quemo_bu = True
    break
  if (todavia_no_se_impulso):
    if estamos_perihelio(distancia(wn[n],zn[n]),distancia(wn[n-1],zn[n-1])):
      tita=np.arctan2((wn[n]),(zn[n]))
      vn[n]=vn[n]+v_impulso*np.sin(tita-(np.pi/2))
      un[n]=un[n]+v_impulso*np.cos(tita-(np.pi/2))

      todavia_no_se_impulso = False
      estamos_en_perihelio = True
      n_perihelio=n
  if(lo_encontramos(wn[n-1],wn_o[n-1],zn[n-1],zn_o[n-1])):
    aparecio = True
    break
  if (llego_sonda(distancia(wn[n-1],zn[n-1]),distancia(wn_o[n-1],zn_o[n-1]))):
    anio_llego = t / anio
    break

  n = n+1
  t = t+h
```