Actividad 4 - Órbita planetaria

Exactas Programa

Verano 2020

Queremos simular el movimiento de dos cuerpos celestes en el espacio: uno fijo en el centro (el Sol), y otro orbitando alrededor de él (la Tierra), usando el algoritmo de Verlet.

En la Figura 1 pueden ver un esquema del movimiento, donde se marcaron las posiciones x[i] e y[i] que se usan para predecir las posiciones futuras x[i+1] e y[i+1]. Notar que además la aceleración A[i] depende de la fuerza gravitatoria, que depende de la distancia entre ambos cuerpos. Con lo cual necesitaremos las posiciones para calcular también la aceleración y sus componentes Ax[i] y Ay[i].

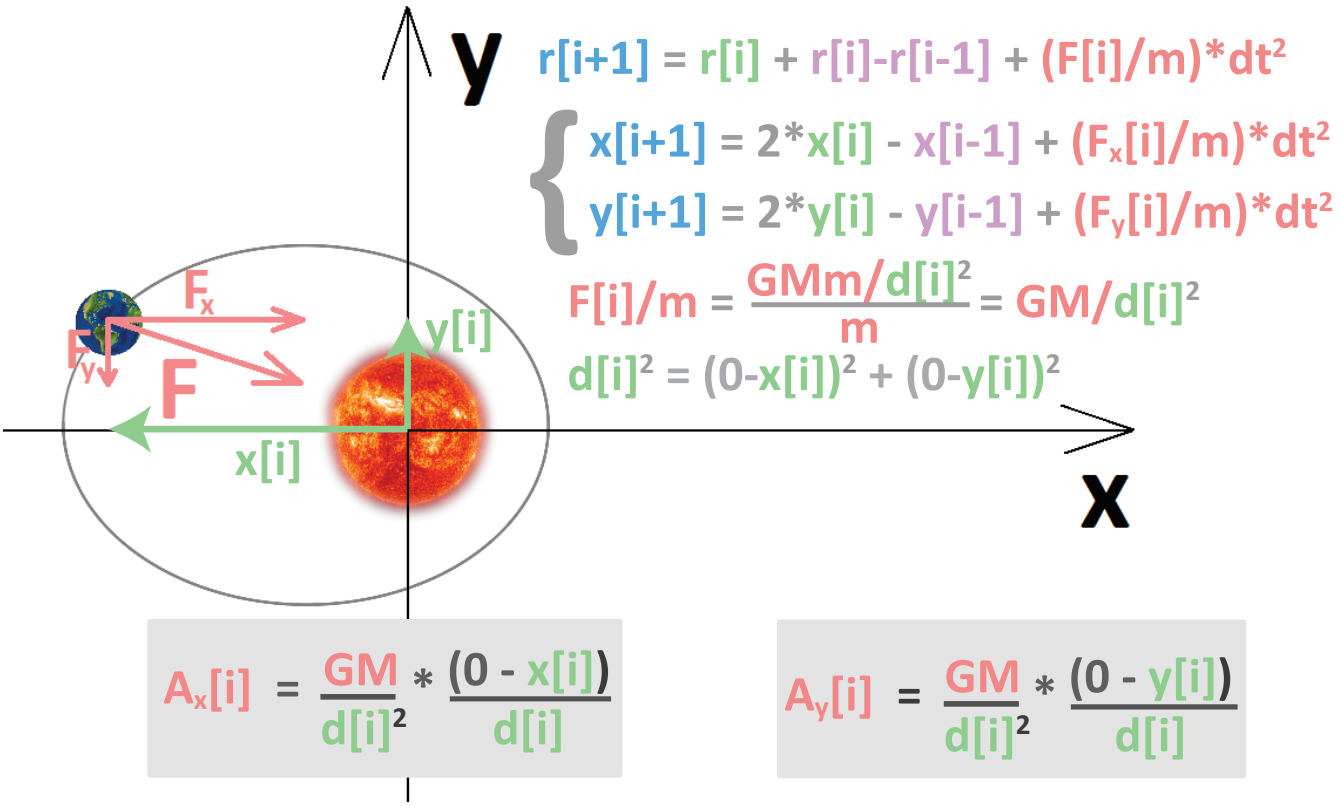

Figura 1: Resumen de expresiones utilizadas para resolver la actividad

1. Ejercicios

Recuerden que para utilizar el algoritmo de Verlet, necesitamos dos posiciones iniciales, es decir un par `(x[i-1], x[i])` y otro `(y[i-1],y[i])`. Tomamos éstas de un momento en particular de la órbita terrestre, sacadas de la página de la NASA[1]. Consideramos que la Tierra comienza moviéndose en la dirección del eje y. Definamos las posiciones en x e y como listas donde iremos guardando las posiciones futuras.

[1] `https://ssd.jpl.nasa.gov/horizons.cgi`

1.
   - `x_lista = [-147095000000.0, -147095000000.0]` (atención que es necesario poner el .0 decimal para que se pueda usar `np.sqrt()` para la raíz cuadrada)
   - `y_lista = [0.0, 2617920000.0]`
   - `dt = 60 * 60 * 24` (para que el paso del tiempo sea un día en) segundos.
   - `tiempo_total = 400` (para simular un poco más de un año.)
   - `x_sol = 0, y_sol = 0` (recuerden que consideramos quieto al Sol)

   Para realizar la simulación, primero definamos funciones que devuelvan la aceleración tomando las posiciones actuales del Sol y la Tierra, usando las ecuaciones de la figura:

   - La función `calcula_delta(x_sol, x_tierra)` que recibe dos posiciones en una dimensión (x o y) y retorna la diferencia entre ambas
   - La función `calcula_distancia(pos_sol,pos_tierra)`, que recibe dos listas, una con la posición `[x_sol,y_sol]` y otra con la posicion `[x_tierra,y_tierra]` y retorna la distancia entre ambas.
   - La función `calcula_aceleracion(pos_sol,pos_tierra)` que recibe dos listas, una con la posición `[x_sol,y_sol]` y otra con la posicion `[x_tierra,y_tierra]` y usando las dos funciones anteriores, calcula la aceleración gravitatoria, retornando una lista con dos valores `[aceleracion_x,aceleracion_y]`.

2. Definan las variables y funciones a utilizar y, usando las funciones, calculen las dos primeras aceleraciones correspondientes a las dos primeras posiciones dadas.

```
def calcula_delta(x_sol, x_tierra):
    ___COMPLETAR___

def calcula_distancia(pos_sol,pos_tierra):
    ___COMPLETAR___

def calcula_aceleracion(pos_sol,pos_tierra):
    G = 6.693e-11 # Constante de gravitacion en notacion cientifica
    M = 1.98e30 # Masa del Sol en notacion cientifica
    ___COMPLETAR___

dias = [0,1]
pos_sol = [0,0] # (x,y) del Sol

lista_x = ___COMPLETAR___
lista_y = ___COMPLETAR___
dt = ___COMPLETAR___

lista_aceleracion_x=[]
lista_aceleracion_y=[]

pos_tierra=___COMPLETAR___
aceleraciones=calcula_aceleracion(pos_sol,pos_tierra)
lista_aceleracion_x.append(___COMPLETAR___)
lista_aceleracion_y.append(___COMPLETAR___)

pos_tierra=___COMPLETAR___
aceleraciones=calcula_aceleracion(pos_sol,pos_tierra)
lista_aceleracion_x.append(___COMPLETAR___)
lista_aceleracion_y.append(___COMPLETAR___)
```

3. Defina la función `realiza_verlet(pos_anterior,pos_actual,aceleracion_actual,dt)`, que recibe dos posiciones en forma de listas, `pos_anterior= [x_anterior,y_anterior]`, `pos_actual= [x_actual,y_actual]`, y la `aceleracion_actual=[aceleracion_x,aceleracion_y]`, y usando las ecuaciones de la figura devuelva `pos_posterior = [x_posterior,y_posterior]`.

4. Realice un ciclo que utilizando la función `realiza_verlet` vaya calculando y guardando la trayectoria terrestre, los días correspondientes y las aceleraciones en las listas definidas.

```
tiempo_total=400
for i in range(1, tiempo_total - 1):
    # Genero listas con las posiciones
    pos_actual = ___COMPLETAR___
    pos_anterior = ___COMPLETAR___
    # Calculo la aceleracion
    aceleracion = ___COMPLETAR___
    # Calculo la posicion futura
    pos_posterior = ___COMPLETAR___
    # Guardo las ultimas posiciones
    lista_x.append(___COMPLETAR___)
    lista_y.append(___COMPLETAR___)
    # Guardo las ultimas aceleraciones
    lista_aceleracion_x.append(___COMPLETAR___)
    lista_aceleracion_y.append(___COMPLETAR___)
    # Guardo el dia
    dias.append(i)
```

Para graficar la trayectoria terrestre usaremos la función plot de la biblioteca `matplotlib`.

5. Importe la biblioteca y grafique la trayectoria en el plano (x, y).

```
import matplotlib.pyplot as plt
```

```
#abre una nueva figura para graficar
plt.figure()
#recibe dos listas y grafica la 1ra en el eje x y la 2da en el eje y
#las dos listas deben ser del mismo largo
plt.plot(lista_x,lista_y,'grey')
#'grey' es para que la grafique en color gris
```

6. Grafique ahora la aceleración x ó y en función de los días.

Vamos a hacer un video animado del movimiento. Para eso necesitamos ir guardando una sucesión de fotos de cada día.

7. Arme una función `hacer_foto(lista_x,lista_y,pos_sol,dia)` que reciba las posiciones de la Tierra y el Sol y un día, y haga un gráfico que muestre la trayectoria en el plano (x, y) de la Tierra, el Sol como un punto amarillo y la Tierra como un punto azul en el día elegido.

```
def hacer_foto(lista_x,lista_y,pos_sol,dia):
    #borra lo que hubiera antes en la figura
    plt.clf()
    #grafico trayectoria (x,y)
    plt.plot(___COMPLETAR___,'grey')
    #grafico al Sol
    #'yo' es para hacer un punto amarillo ('y' de yellow y 'o' de punto)
    #ms elige el tamanio del punto
    plt.plot(___COMPLETAR___,'yo',ms=20)
    #grafico a la Tierra mas chiquita
    #'b' es por blue
    plt.plot(___COMPLETAR___,'bo',ms=10)
```

8. Copie la función `hacer_video(lista_x,lista_y,pos_sol,dia, nombre_video)` que reciba las posiciones de la Tierra y el Sol, y guarde un video con la animación del movimiento, con `nombre_video` como nombre del archivo. Necesitará tener instalada la biblioteca `imageio` e importarla (en los laboratorios ya está instalada).

```
import imageio
```

```
def hacer_video(lista_x,lista_y,pos_sol,nombre_video):
    lista_fotos=[] #aca voy a ir guardando las fotos
    for i in range(len(lista_x)):
        if i%2==0: #esto es para guardar 1 de cada 2 fotos y tarde menos
            hacer_foto(lista_x,lista_y,pos_sol,i)
            plt.savefig(nombre_video+'.png')
            lista_fotos.append(imageio.imread(nombre_video+'.png'))
        print(str(i)+' de '+str(len(lista_x))+' fotos guardadas')
    imageio.mimsave(nombre_video+'.mp4', lista_fotos) # funcion que crea el
    video
    print('Video Guardado')
```

En caso de no estar instalada la biblioteca `imageio`, puede instalarla desde la consola del sistema operativo (**no** desde el spyder).

- En Windows: Presionar *Tecla de Windows + R* en el teclado, escribir `cmd` y apretar la tecla enter.
- En Mac: Presionar *Command + Espacio* en el teclado para abrir el buscador y buscar `Terminal`.
- En Linux: Presionar *Ctrl+Alt+T* en el teclado.
  Una vez en la consola ejecutar el comando:
  `pip install imageio`
  Si no anda, puede probar el siguiente comando:
  `conda install -c menpo imageio`

9. ¿Cómo cambia la forma de la trayectoria si la Tierra de pronto fuera al doble de velocidad?
   (**Ayuda**: para esto deberíamos multiplicar `lista_y[1]` por 2, ¿Por qué?)

10. ¿Y si la velocidad fuera la mitad? ¿Es suficientemente chiquito el *dt*?

11. Modifique las funciones `hacer_foto` y `hacer_video` para que reciban también las listas de aceleración y la agreguen al gráfico. Para esto puede utilizar la función `arrow(x1,y1,x2,y2)` de `matplotlib`, la cual grafica una flecha que nace `(x1,y1)` y apunta en la dirección `(x2,y2)`. Modifique la siguiente línea para que la flecha nazca en la posición de la Tierra del día correspondiente, y añádala a la función `hacer_foto`.

```
plt.arrow(___COMPLETAR___, ___COMPLETAR___, aceleracion_x[dia]*10**12.5,
    aceleracion_y[dia]*10**12.5,width=10**9.5,Color='g')
```

12. (opcional) ¿Cómo calcularía la velocidad punto a punto? Grafíquela en función de los días.

13. (opcional) Explorar en la página de la NASA para obtener las posiciones y masas de distintos planetas y agregarlos a la animación. (note que están en unidades astronómicas y debe pasarlos a metros)
    `https://ssd.jpl.nasa.gov/horizons.cgi#results`

14. (opcional) Buscá en la página de la NASA la posición de la Tierra el día de tu nacimiento y fijate si podés determinar qué día ocurrió el primer perihelio o afelio de tu vida (las distancia mínima y máxima de la Tierra al Sol). Te puede ser útil el siguiente link: `https://www.timeanddate.com/date/dateadd.html`