


Universidad Politécnica de Madrid

Ampliación de Matemáticas I

Informe Hito 5

Autor:
Guillermo García del Río

Índice

1. Introducción **1**

2. Enunciado del hito **1**

3. Problema a resolver **1**
3.1. Problema de los N cuerpos 1

4. Exposición del código **2**
4.1. Código del problema de los N cuerpos 3
4.2. Código principal del Hito 5 3

5. Resultados **5**
5.1. Simulación de dos cuerpos 5
5.2. Simulación de tres cuerpos 6
5.3. Simulación de cuatro cuerpos 7

Índice de figuras

4.1. Dependencias entre archivos 2
5.1. Resultados obtenidos para 2 cuerpos 6
5.2. Resultados obtenidos para 3 cuerpos 7
5.3. Resultados obtenidos para 4 cuerpos 7

1. Introducción

En este informe se presentan y comentan los resultados obtenidos en el hito 5 propuesto en la asignatura de Ampliación de Matemáticas I. En primer lugar se mostrará el enunciado que marca las pautas del proyecto, seguido de un análisis del problema a resolver, la exposición del código y los resultados obtenidos con él. Para terminar se comentarán y compararán dichos resultados.

2. Enunciado del hito

Los objetivos de este hito quedan establecidos por los siguientes puntos:

1 Escribir una función para integrar el problema de los N cuerpos.

2 Simular un ejemplo y comentar los resultados.

3. Problema a resolver

3.1. Problema de los N cuerpos

El problema de los N cuerpos trata considera N puntos de masa m_i $(i = 1, 2, ..., N)$ en un sistema de referencia inercial en las tres dimensiones del espacio moviéndose bajo la influencia de la atracción gravitacional mútua. Cada masa tendrá un vector de posición asociado $\boldsymbol{r_i}$. Por lo tanto, la fuerza gravitatoria que experimenta una masa m_i, por el efecto de otra masa m_j viene dada por:

$$m_i \frac{d^2 \boldsymbol{r_i}}{dt^2} = \sum_{i=1, i \neq j}^{N} m_i m_j \frac{(\boldsymbol{r_j} - \boldsymbol{r_i})}{\|\boldsymbol{r_j} - \boldsymbol{r_i}\|^3} \tag{3.1.1}$$

Es a partir de esta ecuación diferencial de la cual se obtiene el vector de estado $\boldsymbol{U}$, el cual está formado por las coordenadas de posición y velocidad de cada partícula. En este código los vectores $\boldsymbol{U}$ y $\boldsymbol{F(U, t)}$ tendrán la siguiente forma:

$$U(t) = \begin{pmatrix} \begin{pmatrix} x_1 \\ \dot{x_1} \\ y_1 \\ \dot{y_1} \\ z_1 \\ \dot{z_1} \end{pmatrix} \\ \begin{pmatrix} . \\ . \\ . \end{pmatrix} \\ \begin{pmatrix} x_N \\ \dot{x_N} \\ y_N \\ \dot{y_N} \\ z_N \\ \dot{z_N} \end{pmatrix} \end{pmatrix} \qquad F(U,t) = \begin{pmatrix} \begin{pmatrix} \dot{x_1} \\ \ddot{x_1} \\ \dot{y_1} \\ \ddot{y_1} \\ \dot{z_1} \\ \ddot{z_1} \end{pmatrix} \\ \begin{pmatrix} . \\ . \\ . \end{pmatrix} \\ \begin{pmatrix} \dot{x_N} \\ \ddot{x_N} \\ \dot{y_N} \\ \ddot{y_N} \\ \dot{z_N} \\ \ddot{z_N} \end{pmatrix} \end{pmatrix} \tag{3.1.2}$$

4. Exposición del código

De cara a poder entender lo que se está haciendo se va a proceder a exponer el código desarrollado para solucionar el problema propuesto. En este informe tan solo se incluirán los códigos desarrollados para el la integración problema de los N cuerpos y el archivo principal del Hito 5, ya que son las únicas novedades con respecto a hitos anteriores. Una visión general de la dependencia entre módulos y archivos se muestra en la figura 4.1:

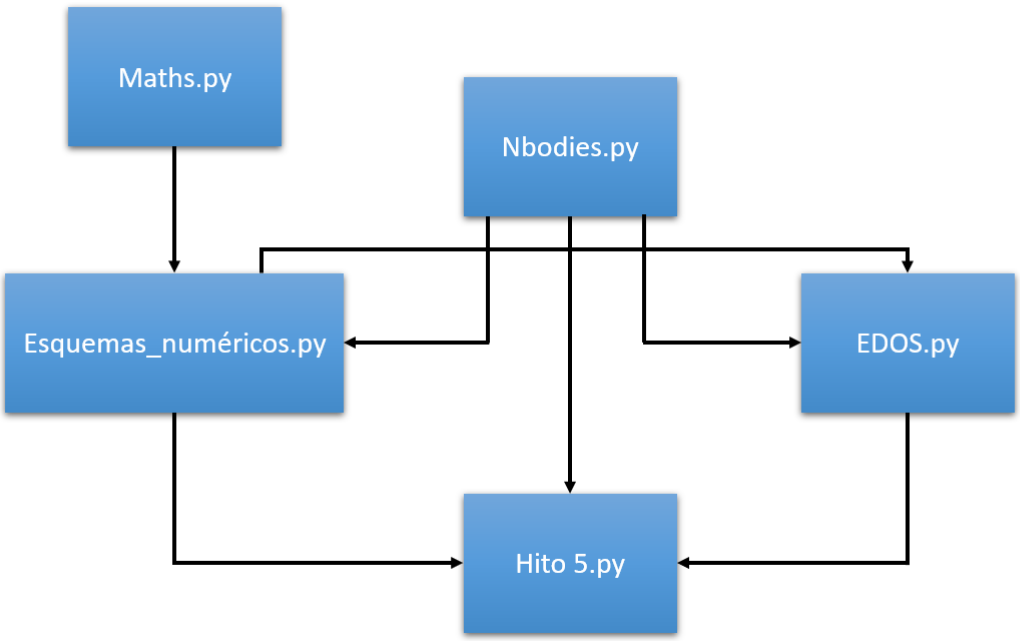


Figura 4.1: Dependencias entre archivos

4.1. Código del problema de los N cuerpos

```
1
2  def N_Bodies(U,t):
3
4      (Nb,Nc) = (4,3)                      # Number of bodies and number of
       coordinates
5
6      # Pointers
7      Us = reshape(U,(Nb,Nc,2))
8      F  = zeros(len(U))
9      Fs = reshape(F,(Nb,Nc,2))            # Respect Us order
10
11     r  = reshape(Us[:,:,0],(Nb,Nc))      # Position, first index is the body and
       second its position
12     v  = reshape(Us[:,:,1],(Nb,Nc))      # Velocity, first index is the body and
       second its velocity
13
14     drdt = reshape(Fs[:,:,0], (Nb,Nc))
15     dvdt = reshape(Fs[:,:,1], (Nb,Nc))
16
17     dvdt[:,:] = 0
18
19     for i in range(Nb):
20
21         drdt[i,:] = v[i,:]
22
23         for j in range(Nb):
24
25             if j != i:                    # Only aplicable for different bodies
26
27                 d = r[j,:] - r[i,:]
28                 dvdt[i,:] = dvdt[i,:] + d[:]/(linalg.norm(d)**3)
29
30     return F
```

Algoritmo 4.1: Problema de los N cuerpos

De este código se puede destacar el uso de punteros con el objetivo de optimizar el uso de memoria y escribir un código compacto para la función. Si bien es cierto que Python no tiene variables puntero, en este lenguaje todas las variables se tratan como punteros internamente. El uso del *reshape* de *Numpy* permite crear los punteros $\boldsymbol{U}$ y $\boldsymbol{U_s}$ y así sucesivamente con el resto de variables. Como se puede comprobar, el código resultante es fácilmente legible y se sabe qué variables utilizar para aplicar las fórmulas del problema.

4.2. Código principal del Hito 5

```
1
2  ## Temporal variables ##
3
4  T = 80         # Integration duration [s]
5  dt = array([0.01])   # Integration step [s]
6
7  Nb = 4         # Number of bodies
8
9  ## Initial condition ##
10
11 U0 = zeros(Nb*2*3)
12
13 if Nb == 2: # T = 80s
14     r01 = array([1, 1, 0])
15     v01 = array([-0.5, 0, 0])
16
```

```
    r02 = array([-1, -1, 0])
    v02 = array([0.5, 0, 0])

    r = concatenate((r01,r02),axis=0)
    v = concatenate((v01,v02),axis=0)

elif Nb == 3:

    r01 = array([2, 2, 0])
    v01 = array([0.5, 0, -0.1])

    r02 = array([-2, -2, 0])
    v02 = array([-0.5, 0, -0.1])

    r03 = array([0, 0, 0])
    v03 = array([0 , 0, 0])

    r = concatenate((r01,r02,r03),axis=0)
    v = concatenate((v01,v02,v03),axis=0)

elif Nb == 4:

    r01 = array([2, 2, 0])
    v01 = array([-0.4, 0, 0])

    r02 = array([-2,2,0])
    v02 = array([0,-0.4,0])

    r03 = array([-2, -2, 0])
    v03 = array([0.4, 0, 0])

    r04 = array([2, -2, 0])
    v04 = array([0, 0.4, 0])

    r = concatenate((r01,r02,r03,r04),axis=0)
    v = concatenate((v01,v02,v03,v04),axis=0)


for i in range(len(r)):

    U0[2*i] = r[i]
    U0[2*i + 1] = v[i]

#methods = [Euler,RK4, Crank_Nicolson, Euler_inverso, leapfrog]
methods = [RK4]

for j in range (size(methods)):

    for i in range(size(dt)):

        n = int(T/dt[i])                    # Number of steps
        t  = linspace(0,T,n)                # Time array

        U = Cauchy_Problem(N_Bodies,t,U0,methods[j])


        plt.plot(U[:,0], U[:,2], "b")
        plt.plot(U[:,6], U[:,8], "r")
        if Nb == 3:
            plt.plot(U[:,12],U[:,14],"k.")
        elif Nb == 4:
            plt.plot(U[:,12],U[:,14],"k")
            plt.plot(U[:,18],U[:,20],"purple")

        plt.xlabel("X")
        plt.ylabel("Y",rotation = 0)
        plt.title(f'Proyección en el plano XY de {Nb} cuerpos')
```

```
84
85          plt.grid()
86          plt.show()
87          plt.savefig('Plots/Hito 5/ ' + str(Nb) + methods[j].__name__ + str(dt[i])+'
        2D.png')
88
89          fig = plt.figure()
90          ax1 = fig.add_subplot(111,projection='3d')
91          ax1.plot_wireframe(U[:,0].reshape((-1, 1)), U[:,2].reshape((-1, 1)), U
        [:,4].reshape((-1, 1)), color= "red", label = 'Primer cuerpo')
92          ax1.plot_wireframe(U[:,6].reshape((-1, 1)), U[:,8].reshape((-1, 1)), U
        [:,10].reshape((-1, 1)), color= "blue", label = 'Segundo cuerpo')
93          if Nb == 3:
94              ax1.plot_wireframe(U[:,12].reshape((-1, 1)), U[:,14].reshape((-1, 1)),
        U[:,16].reshape((-1, 1)), color= "black", label = 'Tercer cuerpo')
95          elif Nb == 4:
96              ax1.plot_wireframe(U[:,12].reshape((-1, 1)), U[:,14].reshape((-1, 1)),
        U[:,16].reshape((-1, 1)), color= "black", label = 'Tercer cuerpo')
97              ax1.plot_wireframe(U[:,18].reshape((-1, 1)), U[:,20].reshape((-1, 1)),
        U[:,22].reshape((-1, 1)), color= "purple", label = 'Cuarto cuerpo')
98
99          plt.title(f'{Nb} cuerpos con {methods[j].__name__} y dt = {dt[i]}')
100         plt.xlabel("X")
101         plt.ylabel("Y",rotation = 0)
102         plt.grid()
103
104         plt.legend(loc = 'best')
105         plt.show()
106         plt.savefig('Plots/Hito 5/ ' + str(Nb) + methods[j].__name__ + str(dt[i])+'
        3D.png')
```

Algoritmo 4.2: Código principal del Hito 5

El diseño de este código tenía como objetivo facilitar la elección del número de cuerpos sin tener que hacer grandes cambios en el mismo cada vez que se quisiera estudiar un caso u otro. Si bien es cierto que tan solo contempla las posibilidades de 2, 3 y 4 cuerpos, la implementación de más cuerpos sería trivial. Además, también se pensó una manera para facilitar la introducción de las condiciones iniciales de posición y velocidad de cada cuerpo

5. Resultados

A diferencia de los hitos anteriores, en este hito se ha decidido integrar el problema únicamente con el esquema numérico Runge Kutta de cuarto orden. De la misma manera, las integraciones se han llevado a cabo con un paso de integración de valor $\Delta t = 0{,}01s$.

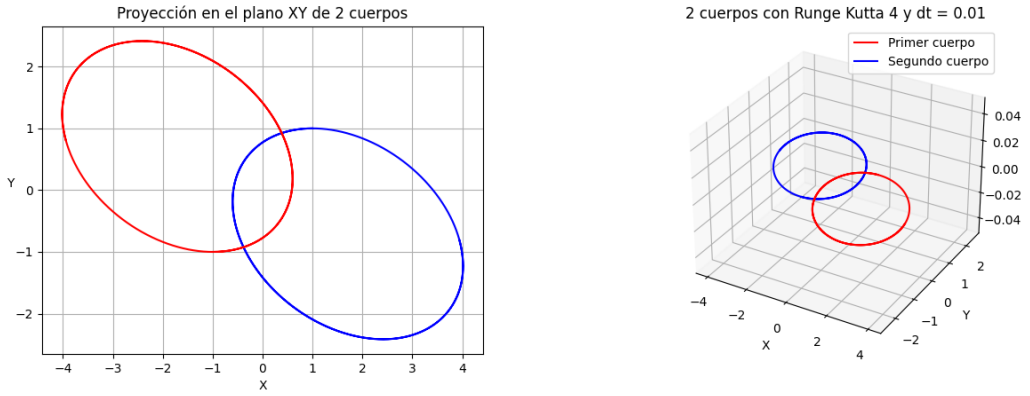
Por otro lado, se ha establecido un tiempo de 80 segundos para todas las simulaciones llevadas a cabo, las cuales han sido para 2, 3 y 4 cuerpos. En cada uno de los casos se han tenido que elegir unas condiciones iniciales específicas y con cierta simetría, ya que este problema es enormemente sensible a las condiciones iniciales con las que empieza la integración.

5.1. Simulación de dos cuerpos

Las condiciones iniciales para este caso son las siguientes:

$$\boldsymbol{r_{01}} = \begin{Bmatrix} 1 \\ 1 \\ 0 \end{Bmatrix} \quad \boldsymbol{v_{01}} = \begin{Bmatrix} -0{,}5 \\ 0 \\ 0 \end{Bmatrix} \quad \boldsymbol{r_{02}} = \begin{Bmatrix} -1 \\ -1 \\ 0 \end{Bmatrix} \quad \boldsymbol{v_{02}} = \begin{Bmatrix} 0{,}5 \\ 0 \\ 0 \end{Bmatrix} \tag{5.1.1}$$

Los resultados obtenidos con estas condiciones iniciales se muestran en la figura 5.1.



(a) Proyección en el plano X-Y para 2 cuerpos

(b) Visualización 3D de la simulación

Figura 5.1: Resultados obtenidos para 2 cuerpos

5.2. Simulación de tres cuerpos

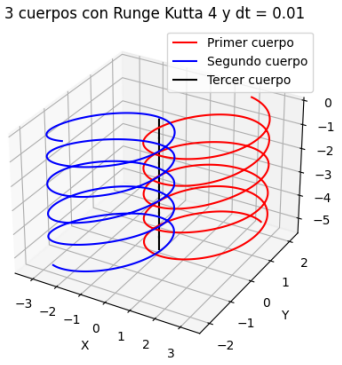
Las condiciones iniciales para este caso son las siguientes:

$$\boldsymbol{r_{01}} = \begin{Bmatrix} 2 \\ 2 \\ 0 \end{Bmatrix} \quad \boldsymbol{v_{01}} = \begin{Bmatrix} 0{,}5 \\ 0 \\ -0{,}1 \end{Bmatrix} \quad \boldsymbol{r_{02}} = \begin{Bmatrix} -2 \\ -2 \\ 0 \end{Bmatrix} \quad \boldsymbol{v_{02}} = \begin{Bmatrix} -0{,}5 \\ 0 \\ -0{,}1 \end{Bmatrix} \tag{5.2.1}$$

$$\boldsymbol{r_{03}} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \quad \boldsymbol{v_{03}} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \tag{5.2.2}$$

Los resultados obtenidos con estas condiciones iniciales se muestran en la figura 5.2.

(a) Proyección en el plano X-Y para 2 cuerpos



(b) Visualización 3D de la simulación

Figura 5.2: Resultados obtenidos para 3 cuerpos

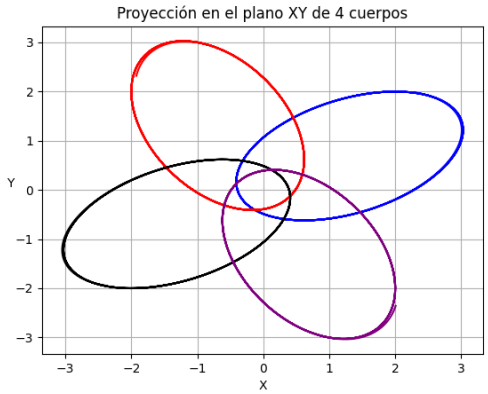
5.3. Simulación de cuatro cuerpos

Las condiciones iniciales para este caso son las siguientes:

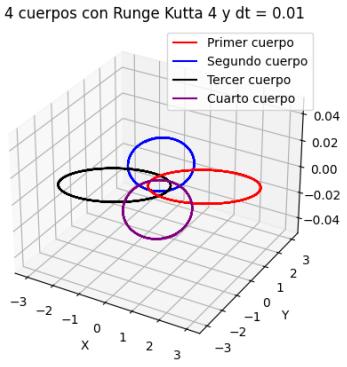
$$\boldsymbol{r_{01}} = \begin{Bmatrix} 2 \\ 2 \\ 0 \end{Bmatrix} \quad \boldsymbol{v_{01}} = \begin{Bmatrix} -0{,}4 \\ 0 \\ 0 \end{Bmatrix} \quad \boldsymbol{r_{02}} = \begin{Bmatrix} -2 \\ 2 \\ 0 \end{Bmatrix} \quad \boldsymbol{v_{02}} = \begin{Bmatrix} 0 \\ -0{,}4 \\ 0 \end{Bmatrix} \tag{5.3.1}$$

$$\boldsymbol{r_{03}} = \begin{Bmatrix} -2 \\ -2 \\ 0 \end{Bmatrix} \quad \boldsymbol{v_{03}} = \begin{Bmatrix} 0{,}4 \\ 0 \\ 0 \end{Bmatrix} \quad \boldsymbol{r_{04}} = \begin{Bmatrix} 2 \\ -2 \\ 0 \end{Bmatrix} \quad \boldsymbol{v_{02}} = \begin{Bmatrix} 0 \\ 0{,}4 \\ 0 \end{Bmatrix} \tag{5.3.2}$$

Los resultados obtenidos con estas condiciones iniciales se muestran en la figura 5.3.



(a) Proyección en el plano X-Y para 2 cuerpos



(b) Visualización 3D de la simulación

Figura 5.3: Resultados obtenidos para 4 cuerpos