Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingeniería Aeronáutica y del Espacio

Máster Universitario en Sistemas Espaciales

# Milestone 5: Problema de los N cuerpos

**Ampliación de Matemáticas I:**
**Cálculo numérico**

**07 de diciembre de 2022**

**Autor:**

- Daniel Cerón Granda

# ÍNDICE

**PÁGINA**

1. **Introducción……………………………………………………………………………………3**
2. **Problema de los N cuerpos……………………………………………………………………3**
3. **Función del problema de los N cuerpos………………………………………………………4**
4. **Ejemplo del problema de los N cuerpos………………………………………………………6**
5. **Relaciones funcionales del programa utilizado ……………………………………………… 8**
6. **Crítica del código …………………………………………………………………………… 8**

# 1.Introducción

Hay dos objetivos principales en la realización de este trabajo. El primero es la realización de una función del problema de los N cuerpos, que simule qué ocurriría en una situación en la que hubiera un número determinado de cuerpos situados en unas coordenadas espaciales y sometidos únicamente a las leyes de gravitación universal entre ellos. Posteriormente, se realizará un ejemplo de una simulación del problema con un número de cuerpos elegido y se discutirán los resultados.

# 2.Problema de los N cuerpos

Antes de empezar con el código, hay que entender la física del problema analizado. El problema de los N cuerpos tiene como origen el estudio de los objetos astronómicos del sistema solar. Al ser un problema complejo, sólo hay solución analítica descubierta para unos pocos casos, como el problema de los dos cuerpos o el problema de los tres cuerpos restringido (se desprecia la masa de uno de los tres cuerpos por ser muy pequeña).

El problema de los N cuerpos tiene como base la ley de gravitación universal de Newton, que determina la fuerza que ejerce un cuerpo i sobre otro j (y también j sobre i):

$$\vec{F}_{ij} = \frac{G * m_i * m_j * \vec{r}}{|\vec{r}|^3}$$

*Ecuación 1: Ley de gravitación universal*

Siendo G la constante de gravitación universal, m las masas de los dos cuerpos involucrados y r el vector de posición entre ellos. A partir de la segunda ley de Newton se pueden obtener las ecuaciones del movimiento de los cuerpos involucrados en el problema como un sumatorio de todas las fuerzas que actúan sobre un cuerpo i:

$$\vec{F}_i = m_i * \frac{d\vec{v}_i}{dt} = \sum_{j=1, j\neq i}^{N} m_j * \frac{(\vec{r}_j - \vec{r}_i)}{|\vec{r}_j - \vec{r}_i|^3}$$

*Ecuación 2: Ecuación de la fuerza total que actúa sobre un cuerpo i*

Una de las simplificaciones que se van a realizar para una mayor facilidad a la hora de realizar este código es la suposición de que todas las masas estudiadas son similares, por lo que la ecuación a estudiar queda como:

$$\frac{d\vec{v}_i}{dt} \cong \sum_{j=1, j\neq i}^{N} \frac{(\vec{r}_j - \vec{r}_i)}{|\vec{r}_j - \vec{r}_i|^3}$$

*Ecuación 3: Ecuación simplificada del problema de los N cuerpos*

Por lo que a la hora de programar esta función se puede expresar como:

$$\frac{dU}{dt} = F(U,t) \rightarrow U = \begin{pmatrix} \vec{r}_1 \\ \vec{v}_1 \\ \vec{r}_2 \\ \vec{v}_2 \\ \cdots \\ \vec{r}_N \\ \vec{v}_N \end{pmatrix}, \qquad F = \begin{pmatrix} \vec{v}_1 \\ \vec{a}_1 \\ \vec{v}_2 \\ \vec{a}_2 \\ \cdots \\ \vec{v}_N \\ \vec{a}_N \end{pmatrix}$$

*Ecuación 4: Ecuación a programar en el código*

Este programa, por la complejidad que presenta, se debe integrar con un esquema numérico muy preciso, por lo que se ha escogido el método RK4, que de los esquemas numéricos analizados hasta el momento, ha demostrado ser el más preciso con diferencia. Para una mayor simplicidad, se ha decidido estudiar un problema dentro de un mismo plano, que puede ser una simplificación aceptable para el estudio de problemas como el del sistema solar, por lo que el número de coordenadas estudiadas es 2. Se ha decidido estudiar un problema con 5 cuerpos, con una complejidad suficiente como para poder ver el efecto de distintas interacciones entre cuerpos; pero sin ser demasiado alta como para tener demasiados cuerpos que analizar o que el programa pueda fallar por la alta complejidad.

# 3.Función del problema de los N cuerpos

La función del problema de los N cuerpos que se ha desarrollado es la siguiente:

```
#Función del problema de los N cuerpos
def F_NBody(U,t):
    Nb = 5
    Nc = 2

    Us = reshape( U, (Nb, Nc, 2) )
    F =  zeros(len(U))
    Fs = reshape(F, (Nb, Nc, 2))

    r = reshape(Us[:, :, 0], (Nb, Nc))
    v = reshape(Us[:, :, 1], (Nb, Nc))

    drdt = reshape(Fs[:, :, 0], (Nb, Nc))
    dvdt = reshape(Fs[:, :, 1], (Nb, Nc))
    dvdt[:,:] = 0

```

*Figura 1: Función del problema de los N cuerpos I*

```python
    for i in range(Nb):
        drdt[i,:] = v[i,:]
        for j in range(Nb):
            if j != i:
                d = r[j,:] - r[i,:]
                dvdt[i,:] = dvdt[i,:] + d[:]/norm(d)**3

    return F

```

*Figura 2: Función del problema de los N cuerpos II*

En primer lugar, se definen una serie de matrices que son reshapes de los vectores U y F, para poder separar los vectores velocidad y posición, y sus respectivas derivadas, de forma adecuada. Son estos reshapes los que se modifican en el problema de los N cuerpos, pero como un reshape mantiene la relación con el vector original, este vector también se ve modificado a su vez. De hecho, lo que devuelve la función es el vector F que se definió como un vector inicial de ceros, pero al hacerse las modificaciones a su reshape, sus valores varían. Esto es lo que se conoce en programación como un alias, hacer una equivalencia entre dos objetos diferentes de tal forma que las operaciones que se realicen sobre uno se reflejan sobre el otro. Estas operaciones se realizan de tal forma que reflejan en cada iteración del problema de los N cuerpos la ecuación 3 vista en el apartado anterior.

Esta función realiza una iteración del problema de los N cuerpos, pero se debe hacer una función en la que se realice la simulación del problema para los tiempos elegidos y un número de iteraciones determinado. Esta función es la siguiente:

```python
#Función de integración del problema de los N-cuerpos
def Int_N_Body(Nb, Nc, N, t0, tf, U0, Scheme):
    deltat = (tf-t0)/(N+1)
    F0 = array(zeros(Nc*2*Nb))
    U = Cauchy_problem(Scheme, U0, N, deltat, F_NBody, F0, 2*Nb*Nc)
    Us  = reshape(U, (N, Nb, Nc, 2))
    r   = reshape(Us[:, :, :, 0], (N, Nb, Nc))
    #Meter gráficos, títulos...
    for i in range(Nb):
        plt.plot(r[:, i, 0], r[:, i, 1])
        plt.plot(r[N-1, i, 0], r[N-1, i, 1], marker = "o", color = "black")

    plt.axis('equal')
    plt.xlabel("x")
    plt.ylabel("y")
    plt.title("Ejemplo del problema de los N cuerpos")
    plt.grid()
    plt.show()
    return U
```

*Figura 3: Función de integración del problema de los N cuerpos*

En la que se dan las condiciones iniciales del problema de los N cuerpos y en la que se realiza la integración a través de la función del problema de Cauchy ya vista. Además, se ha incluido la representación gráfica de las trayectorias de los cuerpos. Por como está definido el problema, es escalable en el espacio, por lo que las unidades de x e y pueden ser desde kilómetros hasta nanómetros si se quisiera, por eso no se definen unidades en los ejes.

## 4.Ejemplo del problema de los N cuerpos

Como parte del programa, se ha incluido un ejemplo de simulación del problema de los N cuerpos:

```
#Simulación de ejemplo del problema de los N cuerpos
Nb = 5 #Define el número de cuerpos
Nc = 2 #Define el número de coordenadas
t0 = 0
tf = 5 #Tiempo final de simulación

#Condiciones iniciales (se pueden variar)
U0 = array(zeros([Nc*2*Nb]))
U0 = array([2,0,2,0, 1,-1,3,1, 1,-1,0,-2, 0,-1,0,1, -2,1,-3,1]) #Para cada cuerpo el orden es (x0, vx0, y0, vy0)

matrizU = Int_N_Body(Nb, Nc, 100*tf, t0, tf, U0, RK4)
```

*Figura 4: Ejemplo del problema de los N cuerpos: condiciones iniciales*

Como ya se ha mencionado, se va a estudiar el problema de los N cuerpos para 5 cuerpos y sólo estudiándolos en un mismo plano (2 coordenadas únicamente). El tiempo de simulación escogido es 5 (como en el espacio, también es un problema escalable en el tiempo, mientras las condiciones iniciales se expresen correctamente en las unidades adecuadas) y las condiciones iniciales son las siguientes, expresadas en ejes (x,y):

- Cuerpo 1: Posición inicial: (2,2), velocidad inicial (0,0)
- Cuerpo 2: Posición inicial: (1,3), velocidad inicial (-1,1)
- Cuerpo 3: Posición inicial: (1,0), velocidad inicial (-1,-2)
- Cuerpo 4: Posición inicial: (0,0), velocidad inicial (-2,-1)
- Cuerpo 5: Posición inicial: (-2,-3), velocidad inicial (1,-1)

Para asegurar un $\Delta t$ suficientemente pequeño como para ser preciso, se ha decidido multiplicar por 100 el tiempo final a la hora de definir el número de particiones de la simulación, lo que origina para cualquier tiempo final un $\Delta t = 0{,}01$. El esquema elegido para la simulación, como ya se ha comentado, es un Range-Kutta de orden 4.

Los resultados de la simulación son los siguientes:

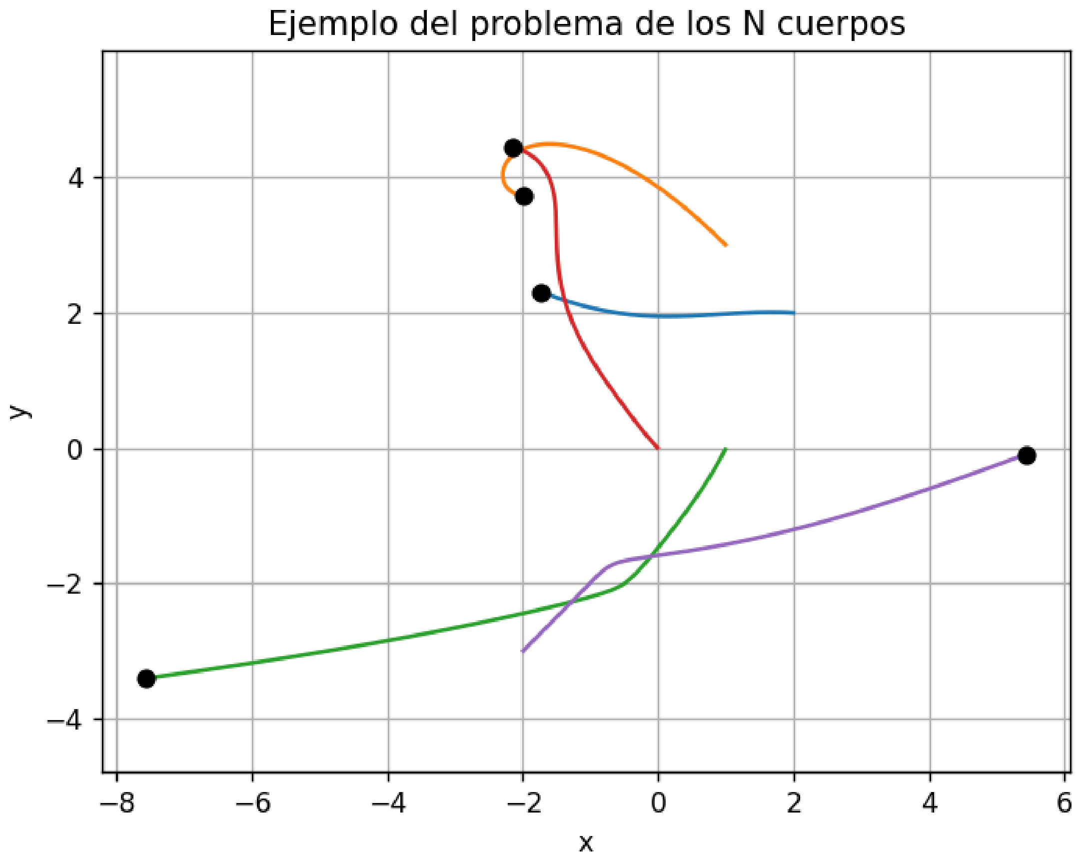


*Figura 5: Ejemplo del problema de los N cuerpos: trayectorias*

El cuerpo 1 es la línea azul, el cuerpo 2 es la línea naranja, el cuerpo 3 es la línea verde, el cuerpo 4 es la línea roja y el cuerpo 5 es la línea morada, destacándose los puntos finales de cada cuerpo con el punto negro. El cuerpo 1 es el que menos cerca llega a pasar del resto de cuerpos, por lo que su trayectoria es la menos curvada de todas. Al tener velocidad inicial nula, no se movería si el resto de los cuerpos no ejercieran una fuerza sobre él. El efecto principal que tienen el resto de los cuerpos sobre él es el desplazamiento a la izquierda, por lo que hacia allí se acaba desplazando. Como en los instantes finales los cuerpos 3 y 5 están muy alejados del mismo, los cuerpos 2 y 4 son los que más influencia tienen al final, y esto se puede comprobar por el curvado que sufre su trayectoria en los instantes finales, atraído por los cuerpos 2 y 4 que están por encima de él mientras él se desplaza hacia la izquierda. El cuerpo 4 sufre una desviación de su trayectoria inicial a causa de pasar cerca del cuerpo 1, que curva su trayectoria para que se vuelva más vertical. Posteriormente, se encuentra muy cerca del cuerpo 2, por lo que ambos cuerpos sufren una gran desviación de su trayectoria, sobre todo el cuerpo 2 que se ve el pronunciado giro que sufre.

En cuanto a los cuerpos 3 y 5, son más independientes de los otros tres cuerpos, su trayectoria sufre desviaciones muy poco pronunciadas por el efecto de los cuerpos 1, 2 y 4; demostrando que, en efecto, la distancia es un parámetro fundamental a la hora de analizar la influencia de un cuerpo sobre otro. Sin embargo, mientras estos cuerpos están con una trayectoria cuya influencia principal está fuertemente marcada por su velocidad inicial, pasan a encontrarse muy cerca entre ellos, por lo que la fuerza gravitacional entre estos dos cuerpos pasa a ser dominante en su trayectoria. Sus trayectorias se curvan en gran medida en el tiempo que pasan cerca el uno del otro, provocando como efecto principal un cambio en la dirección de la trayectoria. Las trayectorias que siguen recuerdan a las de las maniobras de asistencia gravitatoria que se realizan utilizando la fuerza de atracción gravitatoria de un planeta para cambiar su dirección. La principal diferencia es que en este caso la masa y tamaño de uno de los dos cuerpos no son despreciables respecto a los del otro, por lo que ambos cuerpos sufren un cambio en su trayectoria similar. Una vez alejados lo suficiente uno del otro, la velocidad inicial (entendiéndose la velocidad inicial ahora como la velocidad que tienen los cuerpos tras el encontronazo y cambio de dirección) vuelve a ser la acción dominante en los desplazamientos y vuelve a ser la trayectoria muy similar a una recta.

En estos cuerpos se puede comprobar que, una vez alejados lo suficiente, todas las trayectorias de los cuerpos van a ser rectas y se van a ir a infinito, alejándose tanto entre ellos que la gravitación no es una fuerza que vaya a cambiar su trayectoria, a menos que las condiciones iniciales sean tales que acaben encontrando la forma de orbitarse entre ellos. Una forma de solucionar esto sería darle a los cuerpos masas diferentes, y ver si podría darse el caso de que haya cuerpos que acaben orbitando a otro con las condiciones iniciales adecuadas.

# 5.Relaciones funcionales del programa utilizado

En el siguiente esquema se pueden ver el esquema de relaciones funcionales utilizado en este programa:

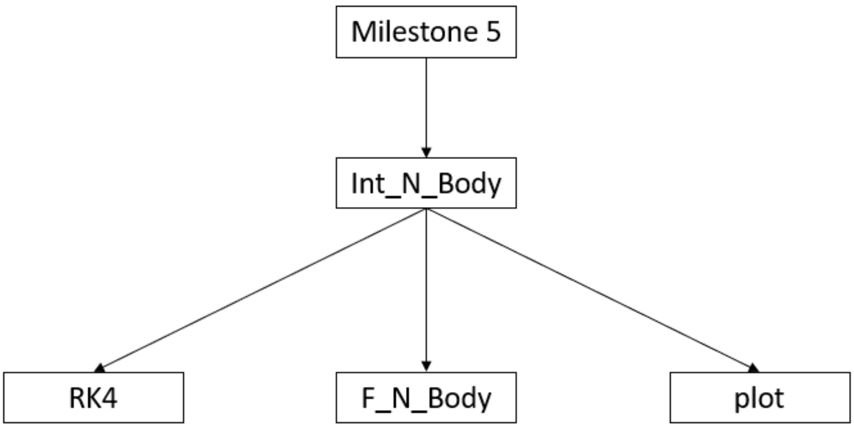


*Figura 6: Esquema de relaciones funcionales del programa*

# 6.Crítica del código

La principal crítica hacia el código es el hecho de que las funciones estén incluidas dentro del propio código, en vez de estar fuera de él. He realizado varios intentos para poder sacarlas, pero todos ellos me hubieran requerido editar o rehacer algunas otras funciones, y en muchos casos varios de esos cambios hubieran hecho que la función no sirviera en otros casos. Aun así, no es un código largo, y el uso de funciones sigue recortando mucho la cantidad de líneas de código en el caso de que después se requiriera la computación de varias condiciones iniciales, por lo que he optado por dejarlo de esta manera.

Otro punto de mejoría es que la representación se ha realizado exclusivamente para el caso de dos coordenadas, por lo que habría que editar esa parte del código si se quisiera pasar a la representación en 3D, pero con unos sencillos retoques podría llegar a lograrse.

En el resto de los aspectos, el código continúa con la mejoría en cuanto a la estructuración y organización respecto a los códigos vistos en los primeros Milestones. A diferencia del Milestone 4, no se han utilizado funciones wrapped por no considerarse necesarias en ninguno de los aspectos del código, pero sí que en la función del problema de los N cuerpos se han utilizado varios alias.