PROBLEMA DE LOS N-CUERPOS

$$\frac{d\vec{r}_i}{dt} = \vec{v}_i \qquad i = 1, \dots, N_b$$

$j \to$ cuerpo j $\quad$ $i \to$ cuerpo i

$$\frac{d\vec{v}_i}{dt} = \sum_{\substack{j=1 \\ i \neq j}}^{N_b} \frac{\vec{x}_j - \vec{x}_i}{\|\vec{r}_j - \vec{r}_i\|^3} \qquad i = 1, \dots, N_b \to \text{Fuerzas gravitacionales}$$

El objetivo es vincular lo anterior al prob. de Cauchy $\left(\frac{dU}{dt} = F(U, t)\right)$ (vector columna) mediante punteros. Habrá que garantizar que los gdl de los cuerpos por N_b sea consistente.

$$U = \begin{pmatrix} \vec{r}_1 \\ \vec{v}_1 \\ \vec{r}_2 \\ \vec{v}_2 \\ \vdots \\ \vec{r}_{Nb} \\ \vec{v}_{Nb} \end{pmatrix}$$

a su vez, vector columna con sus gdl.

$$\underbrace{\frac{dU}{dt} = \begin{pmatrix} \vec{v}_1 \\ \sum \frac{\vec{r}_i - \vec{r}_j}{\cdots} \\ \\ \end{pmatrix}}$$

Esta sería la F que necesito para el esquema temporal

$$U^{n+1} = U^n + \Delta t \cdot F(U^n)$$

¿Cómo se hace? A través de punteros

Ej Python

```
L = [1, 2, 3] (lista)
L1 = L          → La lista L1 es la lista 1
L[0] = 4        → Que la posición 0 valga 4
print (L) → # [4, 2, 3]
```

Esto hace un "alias"
En L1 = L estamos apodando a L con L1

Los punteros son como vistas, en forma de matriz, del vector columna.

$\boxed{\text{El puntero es un alias}}$ No se crea nada nuevo en L1

Cloning

```
L = [1, 2, 3]
L1 = [:]
L1 [0] = 4
print (L)      # L = [1, 2, 3]
                 L1 = [4, 2, 3]
```

Aquí lo que se hace crear una nueva variable.
Para copiar en python hay que importar una librería.

En nuestro caso, queremos aleas en vectores

```
from numpy import array, reshape
```

v = array([1,2,3,4]) (vector columna) | v[0] = 7

pv = reshape(v,(2,2)) | print(pv) ⇒ $\begin{pmatrix} 7 & 2 \\ 3 & 4 \end{pmatrix}$

print(pv) # $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$

pv[0,0] = 8

print(v, pv) ⟶ v = [8, 2,3,4]

pv = $\begin{pmatrix} 8 & 2 \\ 3 & 4 \end{pmatrix}$

Problema

Nb = 100

Nc = 3

v = zeros((2·Nb·Nc) → vector columna

Comenzamos con los punteros

vs = reshape(v, (Nb, Nc, 2)) (↱ body, ↱ cuerpos, ↱ velocidad, posición)

x = reshape(vs[:, :, 0], (Nb, Nc))

lo que cogemos

para todos los Nb de todos los Nc del término posición (la 0)

v = reshape(vs[:, :, 1], (Nb, Nc))

$\boxed{F}$

F = zeros((2·Nb·Nc) → vector columna

Fs = reshape(F, (Nb, Nc, 2)) (↱ body, ↱ cuerpos, ↱ velocidad, posición)

dr/dt = reshape(Fs[:, :, 0], (Nb, Nc))

dv/dt = reshape(Fv[:, :, 1], (Nb, Nc))

% % % % % % %

drdt = v

$$dvdt = \varepsilon \frac{\bar{r}_j - \bar{r}_i}{\| r_j - \bar{r}_i \|^3}$$