

```

1  from numba import njit → Se compilará para ejecución más rápida, sin las verificaciones típicas Python.
2  from numpy import zeros, float64, float32
3  #import miscellaneous.decorators
4  from miscellaneous import decorators
5
6
7  #@decorators.exceptions
8  @decorators.profiling
9  #@njit
10 def Cauchy_problem( F, t, U0, Temporal_scheme, q=None, Tolerance=None ):
11     """
12         Eu este apartado definimos el problema de Cauchy
13
14     Inputs:
15         F(U,t) : Function  $dU/dt = F(U,t)$  Función que describe el sist de ecuaciones
16                  t : time partition t (vector of length N+1) diferenciales (por ejemplo, el problema de Kepler)
17         U0 : initial condition at t=0
18         Temporal_schem
19         q : order o scheme (optional)
20         Tolerance: Error tolerance (optional)
21
22     Return:
23         U: matrix[N+1, Nv], Nv state values at N+1 time steps
24     """
25
26     N = len(t)-1 Número de pasos en el tiempo.
27     Nv = len(U0) Número de variables en el estado (por ejemplo 4 si tenemos posiciones y velocidades en 2D)
28     #U = zeros( (N+1, Nv), dtype=type(U0) )
29     U = zeros( (N+1, Nv), dtype=float64 ) Matriz de ceros que almacena los valores del
30                                             estado del sistema en cada instante de tiempo
31
32     U[0,:] = U0
33
34     for n in range(N):
35
36         if q != None:
37             U[n+1,:] = Temporal_scheme( U[n, :], t[n+1] - t[n], t[n], F, q, Tolerance )
38
39         else:
40             U[n+1,:] = Temporal_scheme( U[n, :], t[n+1] - t[n], t[n], F )
41
42     Este bucle:
43     Recorre cada paso de tiempo desde to hasta tn
44     En cada iteración utiliza el esquema de integración temporal para calcular el estado del sistema en
45     el siguiente instante de tiempo

```

Problema de Cauchy

Es un problema de valor inicial que consiste en encontrar una solución para unas EDOs. Se conoce:

- Estado inicial (U_0): en el contexto de un problema físico pueden ser las posiciones, velocidades...
- Una ecuación diferencial: representada por $F(U,t)$ que describe cómo cambian las variables de estado con el tiempo

Como se resuelve un problema de Cauchy?

Aproximamos la solución numéricamente. Para ello dividimos el tiempo en pequeños pasos Δt y utilizamos métodos de integración numérica para avanzar en el tiempo y actualizar el estado

Código

El código del **problema de Cauchy** implementa un método para resolver ecuaciones diferenciales ordinarias (EDO) usando integración numérica. Toma un **estado inicial** y una función que describe cómo evoluciona el sistema, y avanza en el tiempo utilizando un **esquema temporal** (como Euler, Runge-Kutta, etc.). El objetivo es calcular el estado del sistema en cada paso de tiempo, almacenando los resultados en una matriz. En el código del problema de Cauchy, no se especifican directamente ni el esquema temporal ni la función que describe la evolución del sistema. Estos se pasan como argumentos a la función `Cauchy_problem` cuando se llama (el código espera que cuando llames a la función `Cauchy_problem` le digas que `esquematemporal` y que función del sistema quieres usar)