



- Métodos computacionales 2:  
Alejandro Segura
- **Naive integration**
  - a) Incluir el código Notebook (.ipynb).
  - b) Subir el código y procedimientos con el nombre **Semana1\_Nombre1\_Nombre2**, a la plataforma bloque-neon y sin comprimir.
  - c) **Hacer una sola entrega por grupo.**

## Contents

<b>1 Ordinary Differential Equations</b>	<b>3</b>
1.1 Caída con restitución . . . . .	4

## List of Figures

# 1 Ordinary Differential Equations

## 1.1 Caída con restitución

1. Considere la simulación de n-cuerpos vista en clase. Ajuste el número de partículas a  $N = 1$  y radio  $R = 1$  con las siguientes condiciones iniciales:

a)  $\vec{r}_0 = (-15, 5)$

b)  $\vec{v}_0 = (1, 0)$

c)  $\vec{a}(t) = (0, -9.8)$

Suponga que en cada colisión contra el suelo se pierde una cierta cantidad de energía. Esta pérdida es controlada por el coeficiente de restitución:

$$e = -\frac{v_2 - v_1}{u_2 - u_1} = 0.9 \quad (1)$$

$v$  se refiere a velocidades finales y  $u$  a velocidades iniciales, en este caso,  $u_1 = v_1 = 0$  dado que el piso no se mueve. En el método `CheckWallLimits()` ajuste esta condición de la simulación. Finalmente, ajuste el área de la caja a:  $A = 20 \times 20$ . Para que se vea fluido, debe reducir el tamaño de la simulación en un factor de 10.

Itere los primeros 30 segundos con un paso temporal de  $\Delta t = 0.01$ .

- (a) ¿Qué tiempo tarda la pelota en dejar de rebotar? Haga su mejor estimación.
- (b) Dibuje la energía mecánica de la pelota como una función del tiempo.

Debería obtener algo como: <https://youtu.be/xRV206d51xY>

2. Construir la animación para el caso 3D. Simule  $N = 100$  partículas libres contenidas en un volumen de  $V = 10 \times 10 \times 10$ . Para que se vea fluido, debe reducir el tamaño de la simulación en un factor de 10.

Debería obtener algo como: <https://youtu.be/3SoUsMuNzxw>