- Métodos computacionales:
  Alejandro Segura
  Taller 1
  Fecha de entrega: Viernes 26/02/2021, 11:00 pm

Diccionario

- Demostrar: Encontrar un resultado general de forma analítica.
- Mostrar: Dibujar un resultado usando `Python`.
- Estimar: Encontrar una solución numérica aproximada.
- Implementar: Escribir una rutina de código funcional que resuelva un problema.

1 Caída libre

1. **(20 Puntos)** La ecuación de movimiento para una caída libre está dada por:

$$m\frac{d^2y}{dt^2} = -mg, \tag{1}$$

donde y representa la posición de la partícula, m su masa y g es la aceleración debida a la gravedad ($g = 9.8\ m/s^2$). Esta ecuación está sujeta a las condiciones iniciales ($y_0 = 0,\ v_0 = +50\ m/s$), que representan la posición y velocidad iniciales respectivamente. La solución analítica de esta ecuación está dada por:

$$y(t) = y_0 + v_0 t - \frac{1}{2}gt^2. \tag{2}$$

Dibuje la posición y velocidad como función del tiempo, iterando hasta que el sistema llegue hasta la posición inicial.

Ahora asuma que el sistema está sometido a una fuerza resistiva proporcional al cuadrado de la velocidad instantánea, tal que:

$$m\frac{d^2y}{dt^2} = -mg + \frac{1}{2}C\rho A v^2, \tag{3}$$

donde $C = 0.8$ es un factor experimental asociado a la resistencia del aire, $\rho = 1.225\ kg/m^3$ es la densidad del aire y $A = \pi R^2$ es el área eficaz de la esfera (tomar inicialmente $R = 0.05\ m$).

a) Demuestre que la solución analítica está dada por:

$$\begin{aligned} y(t) &= y_0 + \frac{1}{\gamma^2} ln\left[\frac{cosh(tanh^{-1}(\frac{\gamma v_0}{\sqrt{g}}))}{cosh(-\gamma\sqrt{g}t + tanh^{-1}(\frac{\gamma v_0}{\sqrt{g}}))}\right], \\ v(t) &= \frac{\sqrt{g}}{\gamma} tanh\bigg(-\gamma\sqrt{g}t + tanh^{-1}\Big(\frac{\gamma v_0}{\sqrt{g}}\Big)\bigg) \end{aligned} \tag{4}$$

donde $\gamma = \sqrt{\frac{C\rho A}{2m}}$ es el parámetro de amortiguamiento, con $m = 1\ kg$. Note que existe una restricción cinemática en este sistema $-1. < \frac{\gamma v_0}{\sqrt{g}} < +1.$

b) Dibuje la posición y velocidad como función del tiempo en la misma gráfica del caso ideal.

c) ¿Cuál es el tiempo de vuelo para ambos casos?

d) Estudie el comportamiento del tiempo de vuelo como función de C.

2 Números Primos y Fibonnaci

1. **(10 Puntos)** Escriba en código que calcule los primeros 1000 números primos.
2. Dibuje los números en función de su posición, es decir, 2 es el primero, 3 es el segundo, 5 es ... ¿Encuentra alguna regularidad en la gráfica?
3. Genere los primeros 30 números de la serie de Fibonnaci. Use la sucesión para calcular el número áureo $\varphi = \frac{1+\sqrt{5}}{2} \approx 1.61803398$ (Opcionalmente, ¿Cuál es el origen de este número?).
4. Estime un error relativo $\epsilon = \frac{|Estimated - Real|}{Real}$
5. Haga una gráfica entre la precisión (ϵ) vs la cantidad de números generados en la sucesión. Adicionalmente, haga un ajuste polinomial para encontrar la dependencia. Esto debería ser algo como: $\epsilon \propto \frac{1}{\sqrt{N}}$.

3 Fitting

1. **(10 Puntos)** Bajar los datos disponibles en `https://github.com/asegura4488/MetodosCompu2021/blob/main/Week2/Data/Hw_data.dat`
2. Gráficar los datos para visualizar la función.
3. Hacer un ajuste (usando `scipy`) a los datos para encontrar los parámetros A y B:

$$f(t) = \frac{A}{1 + e^{-Bt}}. \tag{5}$$

4 Método de Newton-Raphson

1. **(10 Puntos)** Modifique el código visto en clase para estimar automáticamente todas las raíces reales del polinomio (importante para el último punto):

$$f(x) = 3x^5 + 5x^4 - x^3 \tag{6}$$

5 Derivadas

Para el siguiente punto, hacer el código en `C++` y luego pintar en `Python`.

1. **(10 Puntos)** Usando $h = 0.05$, estimar la derivada central de la función:

$$f(x) = x - sin(x), \tag{7}$$

   a) En el punto $x = 0$
   b) En el intervalo $-1 \leq x \leq 1$.
   c) Estimar el error en cada nodo y el error global de la aproximación en el intervalo anterior.

2. **(10 Puntos)** Demuestre que la segunda derivada discreta puede también escribirse como:

$$\frac{df(x_j)}{dx^2} = \frac{f(x_{j+2}) - 2f(x_j) + f(x_{j-2})}{4h^2} \tag{8}$$

   Usando la función del punto anterior, compare el error punto a punto de ambas formulas. ¿Las dos definiciones son consistentes?

6 Integración

1. **(10 Puntos)** Implementar la regla de Simpson 3/8 y $h = 0.01$ para estimar:

$$\int_0^1 e^{-x^2} dx. \tag{9}$$

Compare los errores entre la regla de Simpson 1/3 y 3/8.

2. **(20 Puntos)** Dada la aproximación de cuadratura gausiana:

$$\int_{-1}^{1} f(x) = \sum_{k=0}^{n} w_k f(x_k), \tag{10}$$

donde $w, w_1, ...w_n$ son los coeficientes ponderados.

(a) Halle numéricamente los ceros de los cuatro primeros polinomios de Legendre.

(b) Halle analíticamente el polinomio $p_2(x) = 1 + 2x + x^2$ en la base de Legendre.

(c) Calcule analíticamente las funciones de ponderación (w_k) en la base de Legendre, para 3 y 4 puntos, es decir $n = 2, 3$ usando:

$$w_k = \frac{1}{\frac{d(P_{n+1}(x_k))}{dx}} \int_{-1}^{1} \frac{P_{n+1}(x)}{x - x_k} dx. \tag{11}$$

o bien usando:

$$w_k = \frac{2}{(1 - x_k^2)[P_n'(x_k)]^2} \tag{12}$$

(d) Usando la librería `sympy` de `Python` y la Ecuación (12), calcular los pesos de ponderación para los primeros 10 polinomios de Legendre. Compare que los resultados del paquete `legendre.leggauss` de `Python`.

(e) Estime la siguiente integral usando el método de cuadratura Gausiana:

$$\int_0^{\infty} \frac{1}{x^4 + 1} \approx 1.110721 \tag{13}$$