

# Experiencia y especificaciones (caída en un fluido)

Minjares Neriz Victor Manuel

Universidad de Sonora

vmmn16@gmail.com

Hermosillo, Sonora

14 de febrero de 2020

## 1. Experiencia

Realizando este problema en dos lenguajes diferentes de programación experimente de mayor manera las diferencias y similitudes entre ellos, como: fortran complila mucho más rápido, y python es más fácil de usar. Además entendí en gran manera la utilización de archivo en fortran, y la creación de arreglos en python, cosas que ya sabía hacer pero de manera mucho más básica, también gane callo al momento de corregir los problemas de programación y los problemas lógicos a la hora de pasar mi idea a código, tanto en python como fortran porque una similitud que tienen, que me imagino que todos los lenguajes tienen, es la forma lógica en como debes de escribir código.

## 2. Especificación

Muchas veces es imposible tener una solución analítica para una ecuación diferencial de primer grado, por ello debemos de tomar alternativas para encontrar las soluciones de nuestros problemas que tengan esta propiedad. Una de ellas son las soluciones numéricas, las cuales nos dan la solución de aproximar nuestra solución por medio de diferentes algoritmos que los científicos y matemáticos han ido desarrollando a lo largo de los años. En este caso usaremos el método de euler para resolver ecuaciones de primer grado, el cual tiene la forma :

$$y = y' + h * f(t, y)$$

Donde:  $y$  es el valor de la variable dependiente en el punto  $t$  de la variable independiente.  $y'$  es el valor anterior de la variable dependiente; en el primer punto  $y$ ,  $y^*$  es 0.  $h$  es el ancho de paso que usaras al momento de aproximar.  $f$  es la derivada que se encuentra en la ecuación diferencial y sus argumentos  $(t, y)$  son los valores en ese momento a evaluar, pero no es que se resuelva la ecuación si no, no tendría sentido el problema, más bien en un lado de la ecuación pones la derivada y en el otro todo lo demás, y en ese todo lo demás evalúas  $t, y$  en un intervalo definido donde en cada vez que evalúes en el siguiente punto le aumentarás  $h$  al tiempo  $t$ . Con esto obtendremos los valores aproximados de la ecuación diferencial  $y$ , mientras más chico sea el ancho de paso más parecida será la solución a la real.

### 2.1. Diccionario de variables

$f$ : Función a aproximar, real.

$y_0$ : Condición inicial, real.

$g$ : Constante de gravedad, real.

$m$ : Masa del objeto, real.

$a$ : Intervalo inferior, real.

$b$ : Intervalo superior, real.

$h$ : Ancho de paso, real, vector.

$n$ : Cantidad de anchos de paso a evaluar, entero.

$t$ : Tiempo a evaluar, real.

y: Punto a evaluar en t, real.  
 bf: Constante de fricción del fluido, real.  
 Vin: Velocidad en ese momento, real.  
 V0: Velocidad inicial, real.  
 nada: Variable que no vamos a usar pero la necesitamos por que la función necesita dos variables, real.  
 i: Contador para los DOs, entero.  
 nom: Nombre del archivo donde se guardaran los datos, character.  
 archi: Array para guardar los nombre de archivos para plotear, character, vector.  
 FF: Formato para escribir en el archivo a plotear, character.

## 2.2. Algoritmo

1. Crear subrutina del metodo de euler en un archivo aparte:
  - 1.1. Las variables a pedir son: f,y0,a,b,h,archi,n.
  - 1.2. Declaramos las variables, seran las variables pedidas y cuatro internas (t,y,nom,i).
    - 1.2.1. Usamos IF antitroleo, una por si  $a > b$ .
  - 1.3. Inicializamos las variables internas,  $t = a$ ,  $y = y0$ .
  - 1.4. Creamos formato para escribir los nombres de los arvhicos y usamos el comando WRITE para que escriba el nombre el la variable nom.
    - 1.4. Usamos el comando OPEN para abrir un archivo donde guardar los datos.
      - 1.4.1 OPEN (1, FILE = TRIM(nom), ACTION = "write", STATUS = "unknown")
    - 1.5. Usamos DO para asignar el nombre al archivo donde se guardaran los datos de cada ancho de banda:
      - 1.5.3. Usamos el comando DO para obtener los puntos de la grafica:
        - 1.5.3.1. Condicion para salir del loop,  $t > b$ .
        - 1.5.3.2. WRITE(1,\*) t,y
        - 1.5.3.3.  $t = t + h$
        - 1.5.3.4.  $y = y + h*f(y,t)$
        - 1.5.3.5. Cerramos el DO.
      - 1.5.4. archi = TRIM(nom)
      - 1.5.5. Cerramos DO y OPEN.
    - 1.6. cerramos la subrutina.
  2. Creamos el programa principal:
    - 2.1. Declaramos las variables a usar y pedir: Vin,m,b,a,bf,h(:),FF,n,archi(:),t,i,g,V0.
    - 2.2. Le pedimos las variables: V0,m,b,a,bf,n y las asignamos.

2.2.1. Usamos IF antitroleos para ver que  $n > 0$ .

2.2.2. Asignamos la dimension de h y archi: `ALLOCATE(h(n)), ALLOCATE(archi(n))`.

2.2.3. Usando DO asignamos cada ancho de paso a un elemnto de h y ponemos una IF antitroleo para ver que  $h(i) > 0$ .

2.3. Usando DO y OPEN guardaremos los datos de la evaluacion en la ecuacion analitica en el primer elemento de archi:

2.3.1. `OPEN(2, FILE = "solanalitica", ACTION = "write", STATUS = "unknown")`

2.3.2.  $t = a$

2.3.3. `DO IF t > b EXIT`

2.3.4.  $V_{in} = V_0 * \exp(-(bf/m)*t) + g*(m/bf)*(1-\exp(-t*(bf/m)))$

2.3.5. `WRITE(2,*) t,Vin`

2.3.6  $t = t + (b-a)/b$

2.3.6. Cerramos OPEN y DO.

2.4. Usando DO para llamar a la subrutina del metodo de euler con cada h.

2.5. Usando el commando OPEN creamos un archivo que usaremos para plotear:

2.5.1. `OPEN(20, FILE = "plotear.plt", STATUS = "unknown")`

2.5.2. Creamos el formato FF el cual usaremos para escribir los nombres de los archivos en un archivo para plotear en gnuplot.

2.5.3. `WRITE(20,*) 'set title .^aproximaciones usando Euler de la velocidad en la caida en un fluido`

2.5.4. `WRITE(20,*) 'set xlabel "tiempo`

2.5.5. `WRITE(20,*) 'set ylabel "velocidad`

2.5.6. `WRITE(20,*) 'set terminal png'`

2.5.7. `WRITE(20,*) 'set output Caidaenunfluido.png'`

2.5.8. `WRITE(20,*) 'show label'`

2.5.9. `WRITE(20,*) 'plot "solanalitica"', ('"',TRIM(archi(i)),'"'),i = 1,n)`

2.5.10. Cerramos el DO.

2.5.11. Cerramos OPEN.

2.6. Usamos el comando CALL para que lo plotee en gnuplot: `CALL system('gnuplot -p "plotear.plt")`.

2.7. Usamos el comando CONTAINS para que las variables declaradas en el programa principal sirvan en la funcion.

2.8. Creamos una funcion para calcular la funcion  $f$  que usaremos en la subrutina:

2.8.1. Declaramos las variables que ingresan:  $V_{in}$ ,  $nada$  y declarar la funcion:  $velo$ .

2.8.2. Calculamos el cambia de velocidad aproximado:  $velo = V_{in} * b / m - g$

2.8.3. Terminamos la funcion.

2.9. Terminamos el programa.