Facultad de Ingeniería | Universidad de Buenos Aires

2do. Cuatrimestre | 2024

95.10/CB051 | Modelación numérica
75.12 | 95.04 | Análisis numérico I A
95.13 | Métodos matemáticos y numéricos

Trabajo Práctico #2

Vaciado de un tanque esférico de agua

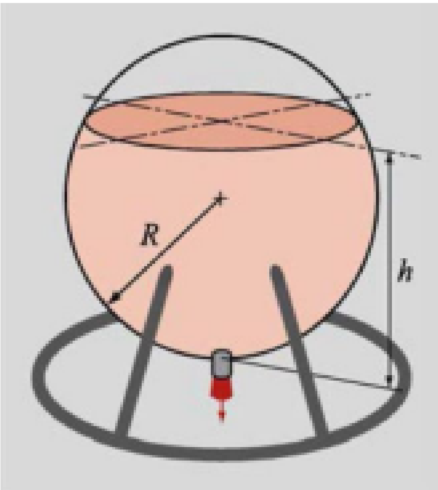
Grupo N° 11	Padrón
Balbuena Victoria Sofía	111031
Cardoso Landaburu Juan Gabriel	110845
Gallo Genis Juan Bautista	111547

Fecha	Correcciones	Docente

Calificación Final	Docente	Fecha

Problema

Se plantea una opción de modelo matemático de vaciado de un tanque de agua esférico a través de un orificio situado en su base, donde la forma geométrica del recipiente determina el comportamiento del agua.




Considerando el tanque esférico de radio R con agua hasta una altura *h*, y suponiendo que el agua fluye a través de un orificio de radio *r* el cual está ubicado en la base del tanque, la ecuación diferencial que describe este comportamiento es la siguiente:

$$\frac{dh}{dt} = \frac{r^2\sqrt{2gh}}{2hR-h^2}$$

Este modelo se denomina M1, siendo la constante $g = 9.81\ m/s^2$ representa la aceleración de la gravedad, el radio R del tanque es de $4\ m$ y el radio r del orificio de salida es de $0.02\ m$. La altura inicial del agua es de $h_0 = 6.5\ m$ (en $t_0=0$).

Una opción de mejora del modelo matemático presentado consiste en sumar el efecto de contracción del flujo en la salida del orificio, por lo que la ecuación presentada sería:

$$\frac{dh}{dt} = - \frac{r^2(c_c\sqrt{2gh}}{2hR - h^2}$$

donde c_c es el coeficiente de contracción que se impone como 0.6, en el modelo que se denomina M2.

Tareas

a) Resolver el problema numérico utilizando el modelo M1 con el método de Euler y obtener el nivel de agua a los 10 minutos. Calcular con tres pasos de discretización diferentes: $\Delta t=10$, $\Delta t=5$ y $\Delta t=1$ (en segundos). Estimar errores de truncamiento y evaluar orden de precisión. Graficar los resultados.

b) Idem *a)* pero utilizando el método de Runge-Kutta 4.

c) Verificar experimentalmente el Orden de Precisión del método de Euler y del método de Runge-Kutta 4

d) Análisis de sensibilidad 1. Utilizar el modelo M2 resuelto con el método de Runge-Kutta 4 y resolver igual que en el ítem b). Comparar con resultados de M1.

e) Análisis de sensibilidad 2. Utilizar los modelos M1 y M2 resueltos con el método de Runge-Kutta 4, adoptar $\Delta t=1$ y repetir cálculos con $c_c=0.55$ y $c_c=0.65$ y comparar resultados evaluando sensibilidad a esa variable.

Resolución

a) Se pide resolver el problema con el modelo M1 mediante el método de Euler para aproximar la solución de la ecuación diferencial que describe el vaciado del tanque esférico. Esta ecuación, que relaciona la variación de la altura del agua con el tiempo (dh/dt), fue evaluada iterativamente con los parámetros dados: radio del tanque (R=4m), radio del orificio (r=0.02m) y la gravedad (g=9.81 m/s2). La ecuación empleada para este método es la siguiente +

$$u_{n+1} = u_n + h.f(t_n, u_n)$$

siendo u la altura en cada iteración, h los intervalos de tiempo impuestos, y la función

$$f(t_n, u_n) = \frac{dh}{dt} = \frac{-r^2.\sqrt{2.g.h}}{2.h.R-h^2}$$

Se realizaron cálculos con tres tamaños de paso h ($\Delta t = 10s,\ 5s,\ 1s$), iterando desde una altura inicial $u = 6,5\ m$, hasta $t = 600s$. Los resultados muestran que a medida que el tamaño del paso disminuye, la solución es más precisa. Las alturas finales aproximadas fueron

Para $\Delta t = 10s \rightarrow 6,2411\ m$.

Para $\Delta t = 5s \rightarrow 6,2413\ m$.

Para $\Delta t = 1s \rightarrow 6,2414\ m$.

Una vez obtenidos los resultados, procedemos a estimar el error de truncamiento causado por el método; se calculan por separado los errores de cada solución (para los distintos Δt), restando la solución obtenida para $\Delta t = 1s$ a las otras dos soluciones, por separado. Es decir que para calcular, por ejemplo, el error de la solución obtenida para $\Delta t = 5s$, se deberá restar el resultado de la solución obtenida para $\Delta t = 1s$. El mismo procedimiento se realiza para calcular el error de la solución obtenida con $\Delta t = 10s$. Luego se divide el primer error por el segundo y se obtiene la estimación del error requerida.

b) Se solicita resolver el mismo problema, también con el modelo M1, pero utilizando el método de Runge-Kutta 4.

Las ecuaciones empleadas para este método son las siguientes

$$u_{*n+1/2} = u_n + \frac{\Delta t}{2}.f(t_n, u_n)$$

$$u_{**n+1/2} = u_n + \frac{\Delta t}{2}.f(t_{n+1}, u_{*n+1/2})$$

$$u_{*n+1/2} = u_n + \frac{\Delta t}{2}.f(t_{n+1/2}, u_{**n+1/2})$$

$$u_{n+1} = u_n + \frac{\Delta t}{6}.[f(t_n, u_n) + 2f(t_{n+1/2}, u_{*n+1/2}) + 2f(t_{n+1/2}, u_{**n+1/2}) + 2f(t_{n+1}, u_{n+1})]$$

Siendo los resultados de la altura final los siguientes:

Para $\Delta t = 10s \rightarrow 6,2414\, m.$

Para $\Delta t = 5s \rightarrow 6,2414\, m.$

Para $\Delta t = 1s \rightarrow 6,2414\, m.$

De la misma manera que en el item a) estimaremos el error de los resultados

Para $\Delta t = 10s \rightarrow Er = 8,6153 * 10^{-13}.$

Para $\Delta t = 5s \rightarrow Er = 8,5265 * 10^{-14}.$

Para $\Delta t = 1s \rightarrow Er = 2,6024 * 10^{-14}.$

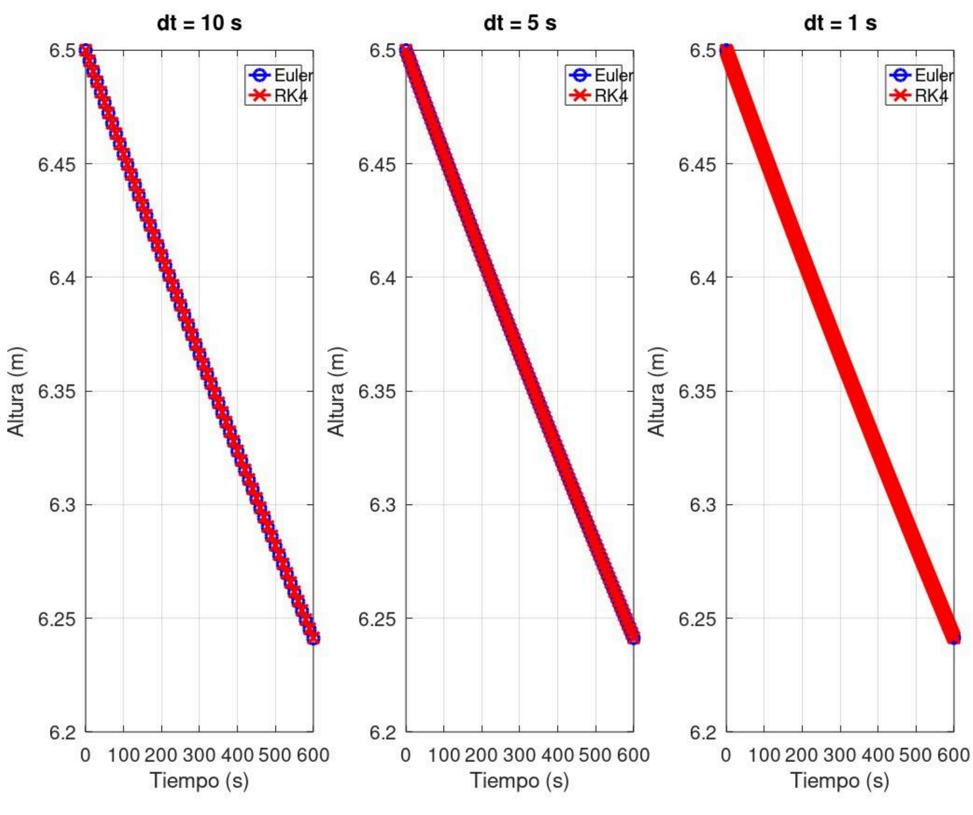


Gráfico para ítems a) y b)

c) Para verificar el orden de precisión de un método numérico tenemos en comprobar que el error global del método disminuye con un orden predecible al reducir el tamaño del paso (Δt). Este análisis se realiza generalmente mediante el cálculo del error global para diferentes tamaños de paso y comparando cómo varía dicho error con respecto a Δt. Para ello empezamos definiendo una solución de referencia utilizando un tamaño de paso bien pequeño para aproximar la solución exacta (Δt = 0,01s). Luego utilizando los valores obtenidos con Euler y RK4 para los Δt = 10s, 5s, 1s, calculamos el error global utilizando la siguiente formula

$$Error = \left| h_{\Delta t}(t_{fin}) - h_{ref}(t_{fin}) \right|$$

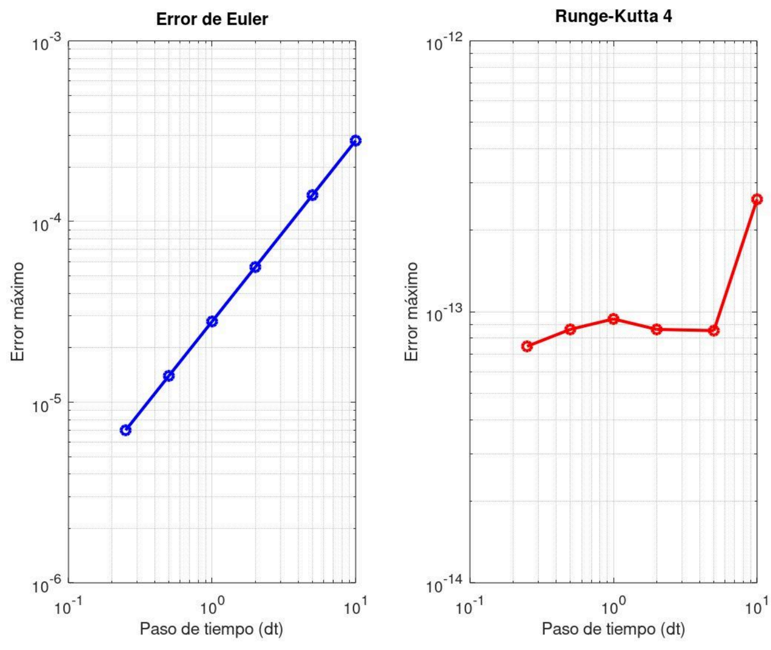
Siendo $h_{\Delta t}(t_{fin})$ la altura final del intervalo elegido y $h_{ref}(t_{fin})$ la solución de referencia. Y por último solo queda estimar el orden de precisión, esto es:

$$p = log_2\left(\frac{Error_{\ \Delta t1}}{Error_{\ \Delta t2}}\right)$$

Siendo los ordenes de precision obtenidos los siguientes:

Orden de presicion Euler: 2,0600

Orden de presicion RK4: 0,69562



d) En este ejercicio emplearemos el método de Runge-Kutta pero para el modelo M2 que tiene la particularidad de integrar consigo el coeficiente de contracción $Cc = 0,6$, modificando así la ecuación diferencial utilizando la siguiente expresión para $f(t_n, u_n)$

$$\frac{dh}{dt} = -\frac{r^2 . Cc . \sqrt{2gh}}{2hR - h^2}$$

respecto al paso, para este caso usaremos los mismos intervalos de tiempo ($\Delta t = 10s,\ 5s,\ 1s$), que para los casos anteriores así se podrá comparar los resultados con M1. Las alturas finales aproximadas para este modelo fueron:

Para $\Delta t = 10s \rightarrow 6,3406\, m$.

Para $\Delta t = 5s \rightarrow 6,3406\, m$.

Para $\Delta t = 1s \rightarrow 6,3406\, m$.

Luego, procedemos de igual manera que en los ítems a) y b) para estimar errores

Para $\Delta t = 10s \rightarrow Er = 7,4607 * 10^{-14}$.

Para $\Delta t = 5s \rightarrow Er = 8,6153 * 10^{-14}$.

Para $\Delta t = 1s \rightarrow Er = 9,1447 * 10^{-14}$.

Una vez finalizado el desarrollo, podemos percibir las diferencias entre los dos modelos. El modelo M2 al incluir un coeficiente contracción para el chorro de salida del 0,6 dando como resultado alturas mayores que los obtenidos por el modelo M1.

e) Para este último ítem debemos resolver el problema con los modelos M1 y M2 utilizando el método Runge-Kutta 4 con un paso únicamente de $\Delta t = 1s$. El método sigue siendo el mismo, utilizando las ecuaciones escritas en inciso b), la diferencia está en el modelo M2 ya que se nos impone utilizar diferentes coeficientes de contracción. Además de $Cc_1 = 0,6$; usaremos $Cc_2 = 0,65$ y $Cc_3 = 0,55$ y luego compararemos resultados. Consiguiendo así las siguientes alturas finales:

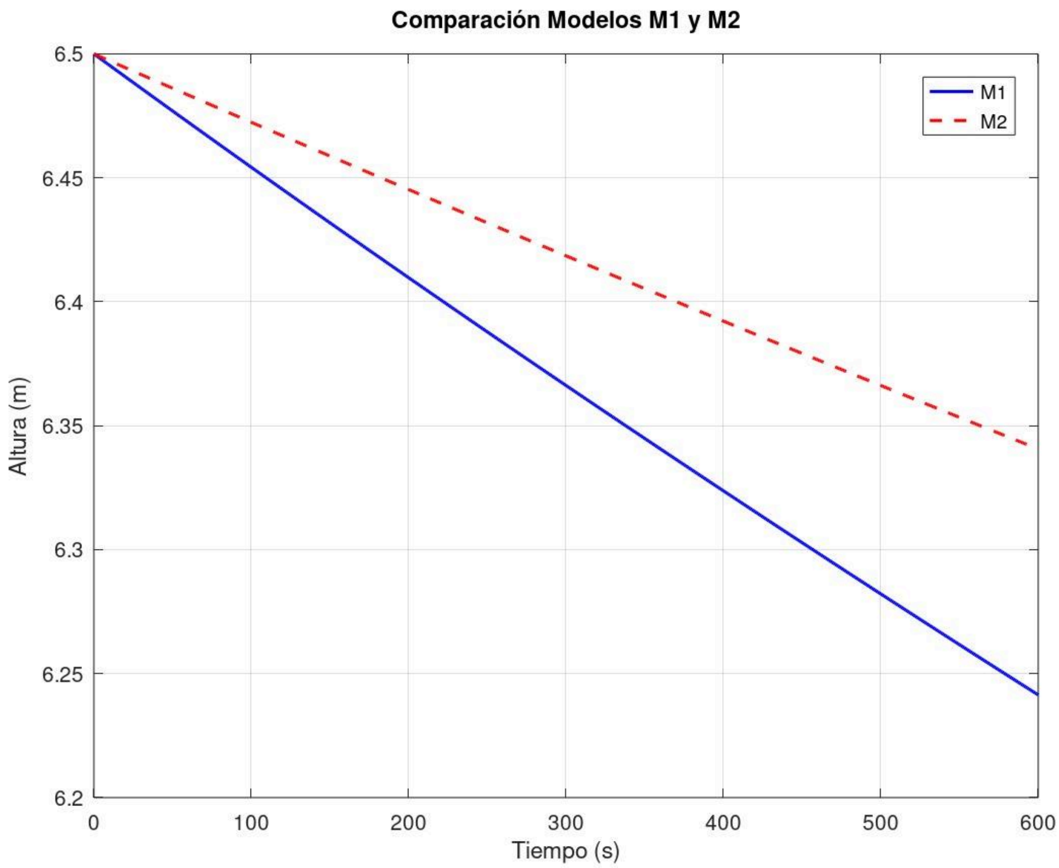
Para M1 con $\Delta t = 1s \rightarrow 6,2414\,m$

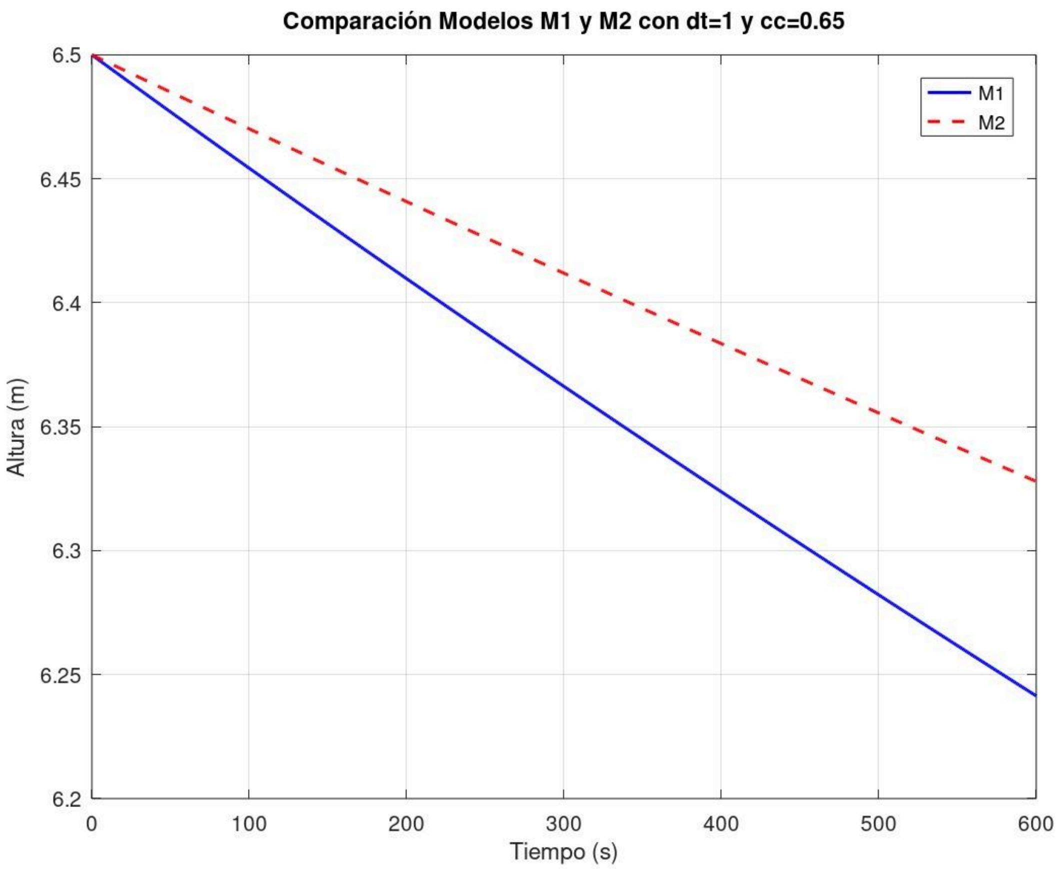
Para M2 ($Cc_1 = 0,6$) con $\Delta t = 1s \rightarrow 6,3406\,m$

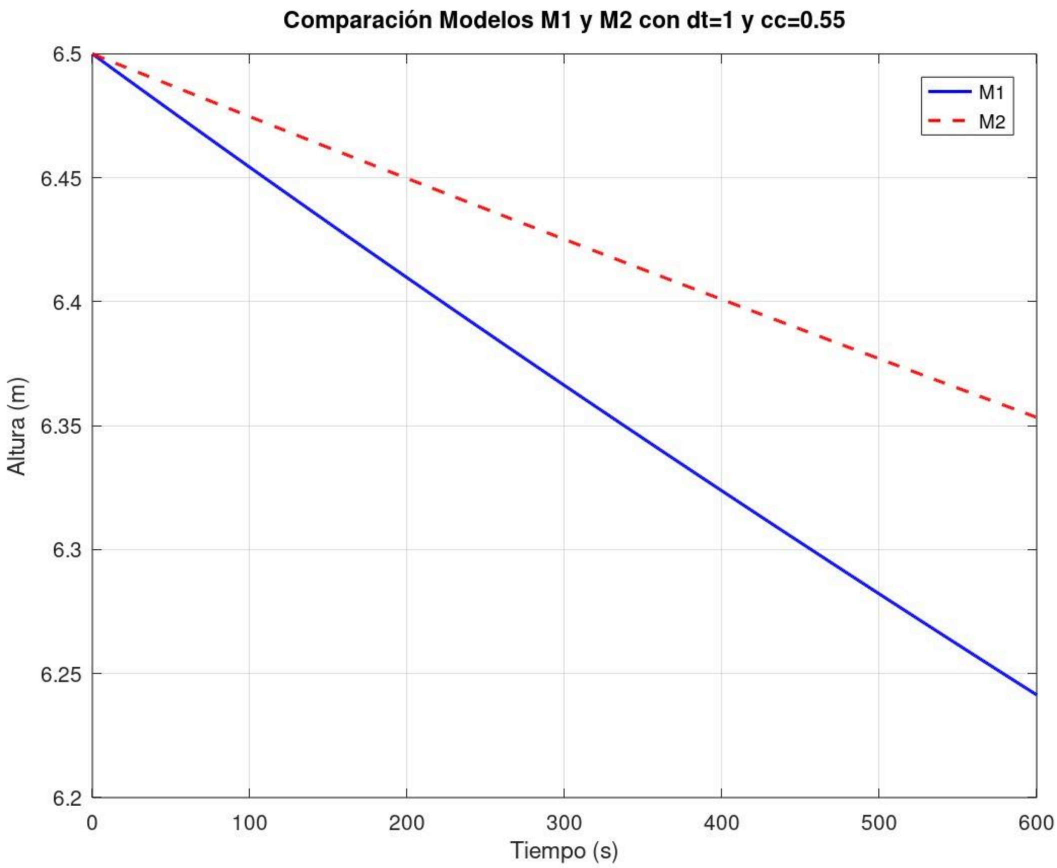
Para M2 ($Cc_2 = 0,65$)con $\Delta t = 1s \rightarrow 6,3534\,m$

Para M2 ($Cc_3 = 0,55$) con $\Delta t = 1s \rightarrow 6,3279\,m$

Luego de esto podemos darnos cuenta que el cambio mínimo del coeficiente de contracción provoca cambios significativos en los resultados finales, siendo así a mayor Cc menor vaciado y viceversa.




Comparación Modelos M1 y M2 con dt=1 y cc=0.65
M1
M2
Altura (m)
Tiempo (s)
6.5
6.45
6.4
6.35
6.3
6.25
6.2
0
100
200
300
400
500
600


Comparación Modelos M1 y M2 con dt=1 y cc=0.55
M1
M2
Altura (m)
Tiempo (s)
6.5
6.45
6.4
6.35
6.3
6.25
6.2
0
100
200
300
400
500
600

Conclusiones

Para finalizar este informe intentaremos dar una breve conclusión acerca del trabajo. El problema presenta dos modelos numéricos para implementar en él y dos métodos para trabajar dichos modelos. Partamos hablando de los métodos, en estos se puede observar claramente que el método de Runge-Kutta 4 es más preciso que el método de Euler, ya que sin importar el paso que se le impuso este termino cambiando mínimamente los resultados, dando errores de truncamiento notoriamente pequeños; a diferencia del método de Euler que si bien es bueno y menos engorroso, refleja el carácter lineal, demostrando un índice de error mayor.

Pero no solo analizamos el método sino también el modelo. A nivel de precisión frente a la realidad, se puede decir que el modelo 2 es más preciso que el 1 ya que este tiene en cuenta el coeficiente de contracción, siendo este un factor que incide en el caso real ya que los fluidos no son ideales (como lo propone el primer modelo). Además sin obviar el hecho de que pocos cambios en el valor del Cc producen significativos cambios en el resultado final.

Anexo Códigos

```
function resultado = Tp2()
  g = 9.81;
  R = 4;
  r_salida_orificio = 0.02;
  h_inicial = 6.5;
  c_contraccion = 0.6;
  t_inicial = 0;
  t_final = 600;

  pasos_dt = [10, 5, 1];

  figure('Position', [100, 100, 1000, 600]);

  for i = 1:length(pasos_dt)
    dt = pasos_dt(i);

    modelo_M1 = @(h) M1(h, R, g, r_salida_orificio);

    [tiempo_euler, alturas_euler] = metodo_Euler(modelo_M1, h_inicial, t_inicial, dt, t_final);

    [tiempo_rk4, alturas_rk4] = runge_kutta_4(modelo_M1, h_inicial, t_inicial, t_final, dt);

    subplot(1, 3, i);
    plot(tiempo_euler, alturas_euler, 'b-o', 'LineWidth', 1.5, 'DisplayName', 'Euler');
    hold on;
    plot(tiempo_rk4, alturas_rk4, 'r-x', 'LineWidth', 1.5, 'DisplayName', 'RK4');

    title(['dt = ', num2str(dt), ' s']);
    xlabel('Tiempo (s)');
    ylabel('Altura (m)');
    legend('show');
    grid on;

  end

  modelo_M2 = @(h) M2(h, R, g, r_salida_orificio, c_contraccion);
  [tiempo_m2, alturas_m2] = runge_kutta_4(modelo_M2, h_inicial, t_inicial, t_final, 1);

  figure;
  plot(tiempo_rk4, alturas_rk4, 'b-', 'LineWidth', 1.5, 'DisplayName', 'M1');
  hold on;
  plot(tiempo_m2, alturas_m2, 'r--', 'LineWidth', 1.5, 'DisplayName', 'M2');
  title('Comparación Modelos M1 y M2');
  xlabel('Tiempo (s)');
```


xlabel('Tiempo (s)');


```
  ylabel('Altura (m)');
  legend('show');
  grid on;

  [~, ~, errores_euler, errores_rk4] = verificar_orden_precision(R, g, r_salida_orificio, h_inicial, t_final);

  orden_euler = calcular_orden_precision(errores_euler);
  orden_rk4 = calcular_orden_precision(errores_rk4);
  disp('Orden de precisión de Euler:');
  disp(orden_euler);
  disp('Orden de precisión de Runge-Kutta 4:');
  disp(orden_rk4);
end

function orden = calcular_orden_precision(errores)
  if length(errores) < 2 || errores(end) <= 0 || errores(1) <= 0
    orden = NaN;

  end

  orden = log(errores(1) / (errores(end) + eps)) / log(length(errores));
end



function [tiempo_referencia, altura_referencia, errores_euler, errores_rk4] = verificar_orden_precision(R, g, r_salida, h_inicial, t_final)
  pasos_dt = [10, 5, 2, 1, 0.5, 0.25];

  dt_referencia = 0.01;
  [tiempo_referencia, altura_referencia] = runge_kutta_4(@(h) M1(h, R, g, r_salida), h_inicial, 0, t_final, dt_referencia);

  errores_euler = zeros(size(pasos_dt));
  errores_rk4 = zeros(size(pasos_dt));

  for i = 1:length(pasos_dt)
    dt = pasos_dt(i);

    [tiempo_euler, altura_euler] = metodo_Euler(@(h) M1(h, R, g, r_salida), h_inicial, 0, dt, t_final);
    [tiempo_rk4, altura_rk4] = runge_kutta_4(@(h) M1(h, R, g, r_salida), h_inicial, 0, t_final, dt);

    altura_ref_interpolada = interp1(tiempo_referencia, altura_referencia, tiempo_euler, 'linear', 'extrap');
    altura_ref_interpolada_rk4 = interp1(tiempo_referencia, altura_referencia, tiempo_rk4, 'linear', 'extrap');

```

```
        errores_euler(i) = max(abs(altura_euler - altura_ref_interpolada));
        errores_rk4(i) = max(abs(altura_rk4 - altura_ref_interpolada_rk4));

    end

    figure;

    subplot(1,2,1);
    loglog(pasos_dt, errores_euler, 'bo-', 'LineWidth', 2);
    title('Error de Euler');
    xlabel('Paso de tiempo (dt)');
    ylabel('Error máximo');
    grid on;

    subplot(1,2,2);
    loglog(pasos_dt, errores_rk4, 'ro-', 'LineWidth', 2);
    title('Runge-Kutta 4');
    xlabel('Paso de tiempo (dt)');
    ylabel('Error máximo');
    grid on;

    disp('Errores de Euler:');
    disp(errores_euler);
    disp('Errores de Runge-Kutta 4:');
    disp(errores_rk4);
end

function dhdt = M1(h, R, g, r_salida)
    if h <= 0
        dhdt = 0;
    else
        dhdt = - (r_salida^2 * sqrt(2 * g * h)) / (2 * h * R - h^2);
    end
end

function dhdt = M2(h, R, g, r_salida, cc)
    if h <= 0
        dhdt = 0;
    else
        dhdt = - (r_salida^2 * cc * sqrt(2 * g * h)) / (2 * h * R - h^2);
    end
end

function [tiempo, alturas] = metodo_Euler(modelo, h0, t0, dt, tf)
    tiempo = t0:dt:tf;
    alturas = zeros(size(tiempo));
    alturas(1) = h0;

    for i = 2:length(tiempo)
        alturas(i) = alturas(i-1) + dt * modelo(alturas(i-1));
        if alturas(i) < 0
            alturas(i) = 0;
        end
    end
end

function [tiempo, alturas] = runge_kutta_4(modelo, h0, t0, tf, dt)
    tiempo = t0:dt:tf;
    alturas = zeros(size(tiempo));
    alturas(1) = h0;

    for i = 2:length(tiempo)
        h = alturas(i-1);
        k1 = modelo(h);
        k2 = modelo(h + dt*k1/2);
        k3 = modelo(h + dt*k2/2);
        k4 = modelo(h + dt*k3);

        alturas(i) = h + dt * (k1 + 2*k2 + 2*k3 + k4) / 6;

        if alturas(i) < 0
            alturas(i) = 0;
        end
    end
end
```