Facultad de Ingeniería | Universidad de Buenos Aires

95.10 | Modelación Numérica

91.13 | Análisis Numérico I

95.13 | Métodos Matemáticos y Numéricos

Trabajo Práctico N°3 – Cuatrimestre 1 de 2020

Vaciado de un tanque de agua esférico

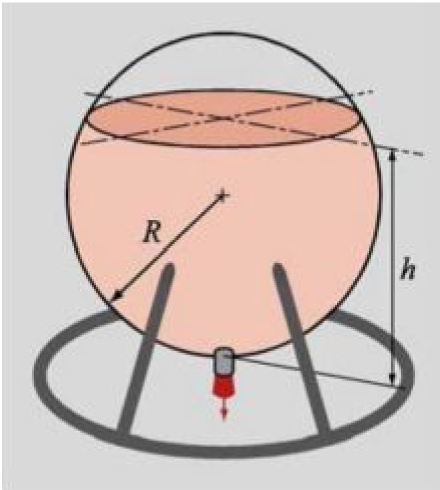
Grupo Nº 23		
	Mauricio Fernandez	**94363**
	Ricardo Brandan	**96970**
	Iñaki Ustariz	**102252**

Fecha	**Correcciones / Observaciones**	**Docente**

Calificación Final	**Docente**	**Fecha**

1. Introducción

En el presente trabajo se desea plantear 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. Para realizar este planteo, se toma al tanque con un radio R y una altura h como se indica en la figura. Además se supone que el agua fluye a través del orificio ubicado en la base cuyo radio es r.



Este comportamiento se describe con un modelo matemático que será corregido mediante un análisis de sensibilidad. Por último, se calculan dos integrales por dos métodos distintos (Simpson y del trapecio).

2. Desarrollo

En el presente informe se busca poder obtener soluciones numéricas para el problema planteado por distintos métodos. El problema consiste en calcular el vaciado de un tanque esférico y se puede modelar de la siguiente manera:

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

- Gravedad (g) = $9,81\ m/s^2$
- Radio esfera (R) = $4\ m$
- Radio orificio de salida r = $0,12\ m$
- Altura inicial del agua ($h_0\ en\ t_0 = 0$) = $6,5\ m$

La dificultad que se presenta es que cuenta con la derivada primera del modelo y se utilizará un método numérico para resolver una Ecuación Diferencial Ordinaria (EDO).

Las ecuaciones diferenciales ordinarias consisten en una ecuación diferencial que relaciona una función desconocida de una variable independiente con sus derivadas. Es decir, una sola variable independiente (a diferencia de las ecuaciones diferenciales parciales que involucran derivadas parciales de varias variables), y una o más de sus derivadas respecto de tal variable.

Métodos utilizados

Para resolver dicho problema, se utilizan tres métodos distintos: Euler, Runge-Kutta 2 y Runge-Kutta 4. Se sintetiza cada método a continuación:

<u>Método de Euler:</u> Se avanza de un nodo de discretización al otro a través de la pendiente de la recta tangente.

$u_{i+1} = u_i + h * f(x_i, u_i)$ donde u_i es el valor donde se parte, h es el paso y $f(x_i, u_i)$ la función evaluada en el punto i.

<u>Método de Runge-Kutta 2</u>: Se avanza a través del promedio aritmético de dos nodos de discretización a través de la pendiente de la recta tangente, el primero es un valor dado y el segundo es un valor estimado. Los nodos son en este caso, uno al principio del paso y otra al final del paso.

$u_{i+1} = u_i + \frac{1}{2}(k_1 + k_2)$ donde k_1 y k_2 corresponden a una pendiente inicial en el paso y una pendiente final en el paso.

$k_1 = h * f(x_i, u_i)$ igual que en el Método de Euler.
$k_2 = h * f(x_{i+1}, u_{i+1}{}^*)$ la aplicación del Método de Euler a la estimación de u_{i+1}

Método de Runge-Kutta 4: Se avanza a través de un solo paso, donde la llegada a la nueva solución es la ponderación de cuatro pendientes, la inicial (que es dato) y tres estimadas (dos a mitad de paso y una al final del paso), generando un mayor beneficio en la precisión.

$u_{i+1} = u_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$ donde tienen mayor peso las pendientes del centro del paso.

$k_1 = h * f(x_i, u_i)$
$k_2 = h * f(x_i + \frac{1}{2}h, u_i + \frac{1}{2}k_1)$
$k_3 = h * f(x_i + \frac{1}{2}h, u_i + \frac{1}{2}k_2)$
$k_4 = h * f(x_i + h, u_i + k_3)$

A nivel de resumen, se puede realizar una comparación de métodos en la siguiente tabla:

Método	Orden de precisión	Estimación de la pendiente en el paso	Evaluaciones de f(x,u) por paso
Euler	1	Valor inicial	1
Runge-Kutta 2	2	Promedio aritmético de la pendiente inicial y final estimada	2
Runge-Kutta 4	4	Promedio ponderado de cuatro valores	4

Tabla 1: comparación de métodos

Explicación de la programación

Para la programación de dichos métodos se crea una función que sigue el modelo matemático del fenómeno. Dicha función es variable de un valor inicial, donde partirá la evaluación, que en este caso es tiempo igual a cero y posición 6,5. Además se tiene en cuenta un paso, que mide cómo aumenta la variable en cada iteración y por último un valor que se llama T en el código de programación que presenta hasta donde se realiza la iteración.

Para cada método se crea una función donde sólo se ingresan los valores y se repite el proceso tantas veces sea necesario al producirse un cambio de paso por ejemplo.

En este caso, el problema que se plantea cuenta con una pequeña modificación porque una variable sólo es afectada que es la posición (representada por x) y el tiempo no aparece en esta derivada. Las variaciones por lo tanto se reflejan solamente en el valor de x y no del tiempo.

Resultados

Método de Euler

Luego, se programó en el programa Octave una función que se la llamó Euler, donde se cargó el modelo matemático y da como resultado al tiempo de 10 minutos (600 segundos) lo siguiente:

Paso	Euler
10 segundos	1.255997882
5 segundos	1.271114929
1 segundo	1.282868154

Tabla 2: Resultados método de Euler

El programa Octave con su configuración predeterminada brinda menos cifras significativas. Para este análisis, se aumenta la cantidad de cifras. En el gráfico 1, se muestra los resultados obtenidos.

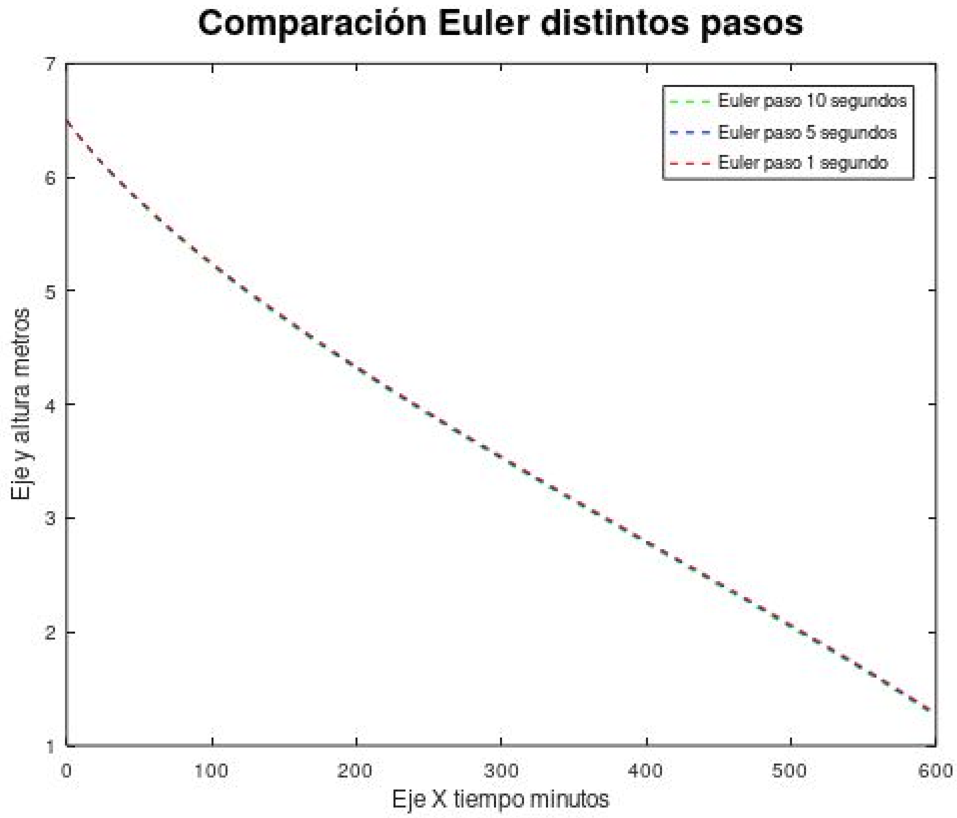


Gráfico 1: comparación de resultados Euler

Si se realiza un detalle del gráfico, se puede apreciar lo siguiente:

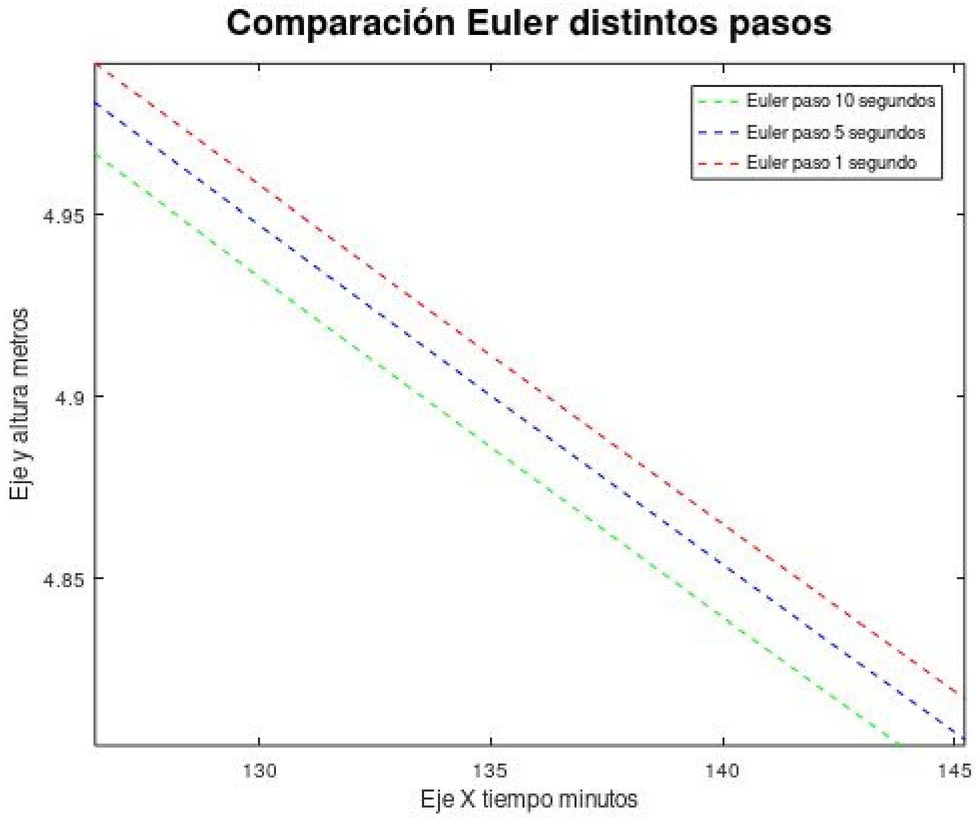


Gráfico 2: zoom comparación de resultados Euler

El método de Euler con un paso de 1 segundo va arrojando valores superiores y a medida que aumenta el paso, se puede ver como el resultado va disminuyendo.

Como conclusión de este método, se aprecia en la tabla 2 que entre cada cambio de paso, se produjeron cambios en los resultados a partir de la tercera cifra significativa.

Método de Runge-Kutta 2

De la misma manera se procede a calcular el valor del nivel del agua en el tanque a los 10 minutos. En este caso, se procede a mostrar los resultados que se han obtenido con el método de Runge-Kutta 2 en la tabla 3.

Paso	Runge-Kutta 2
10 segundos	1.285390452
5 segundos	1.285672422
1 segundo	1.285757819

Tabla 2: Resultados método de Runge-Kutta 2

De la misma manera se visualiza en el gráfico 3, los resultados obtenidos por el método de Runge-Kutta 2.

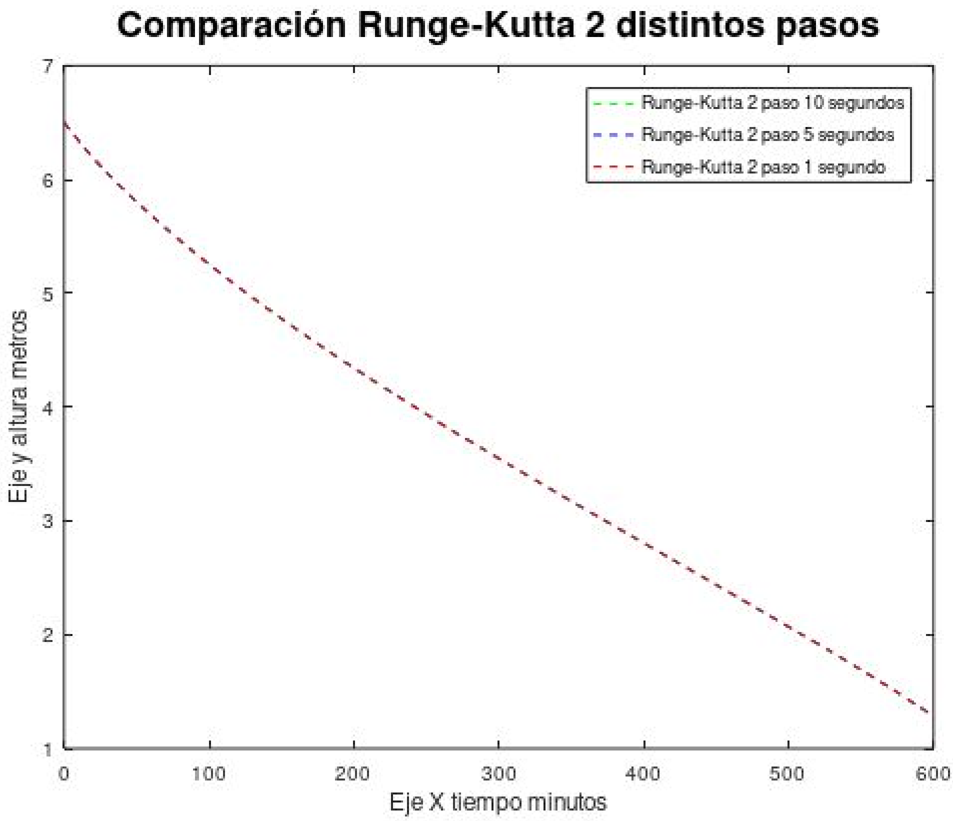


Gráfico 3: comparación de resultados Runge-Kutta 2.

Siguiendo los pasos utilizados con el método de Euler, se procede a realizar un zoom localmente para que se visualice los resultados al tener distintos pasos.

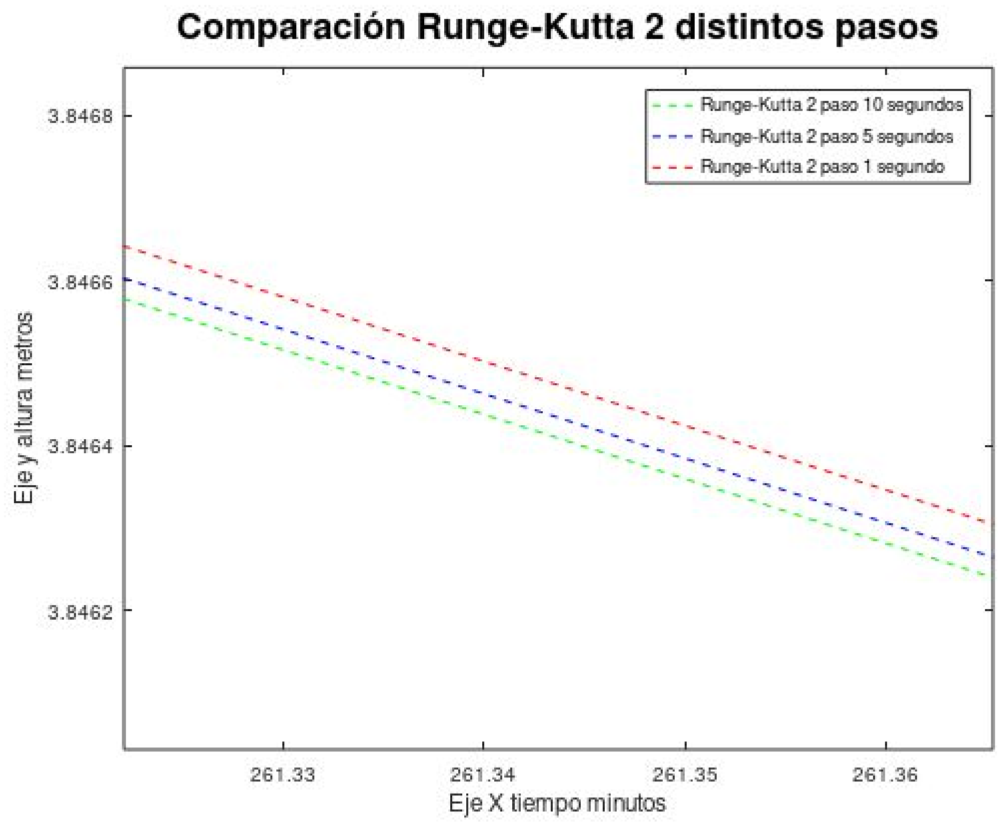


Gráfico 3: zoom de resultados Runge-Kutta 2.

En este caso, se aprecia que para obtener la misma diferencia entre valores se tuvo que recurrir a una escala mucho mayor que en el caso del método de Euler. Lo que significa en otras palabras, que el método nos brinda soluciones con menor error.

En la tabla 2, se observa que los resultados varían a partir de la quinta cifra significativa.

Método de Runge-Kutta 2

Por último, se recurre al método de Runge-Kutta 4 y se observan los siguientes resultados.

Paso	Runge-Kutta 4
10 segundos	1.285760891
5 segundos	1.285761231
1 segundo	1.285761253

Tabla 3: Resultados método de Runge-Kutta 4

De la misma manera se procede a mostrar los resultados obtenidos por este método en un gráfico.

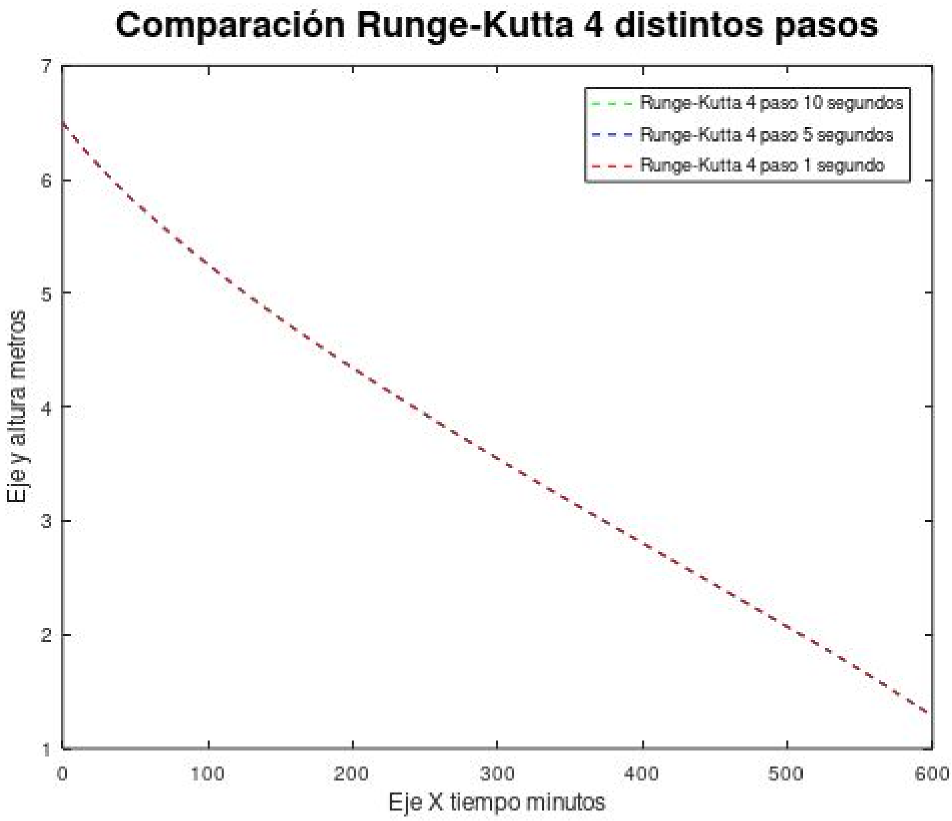


Gráfico 4: resultados Runge-Kutta 4.

Debido a su difícil comprensión a gran escala, nuevamente se procede a realizar un zoom de los resultados obtenidos.

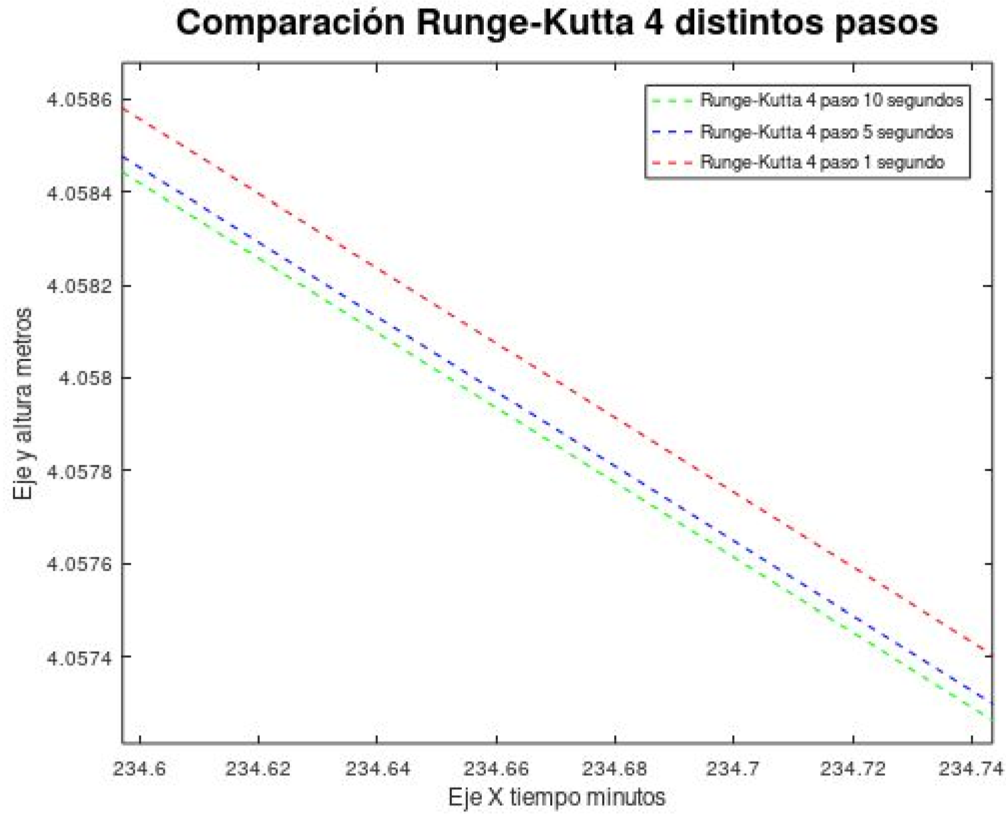


Gráfico 5: zoom resultados Runge-Kutta 4.

Nuevamente, se necesita acercarse mucho al gráfico para poder observar las diferencias de resultados debido al paso.

En este caso, en la tabla 3 se aprecia que en la séptima cifra significativa empiezan a aparecer variaciones.

Resumen y conclusiones de los métodos

Como resumen de los métodos utilizados se presenta la siguiente tabla:

Paso	Euler	Runge-Kutta 2	Runge-Kutta 4
10 segundos	1.255997882	1.285390452	1.285760891
5 segundos	1.271114929	1.285672422	1.285761231
1 segundo	1.282868154	1.285757819	1.285761253

Tabla 4: Comparación de métodos

Se observa que el método de Runge-Kutta 4 con el mayor paso de 10 segundos, logra resultados con menor error que en los otros dos métodos. Si se grafica todos los métodos con sus pasos en un gráfico y se los compara quedaría lo siguiente:

Si se realiza un zoom sobre los puntos resultados finales obtenidos se puede ver lo siguiente:

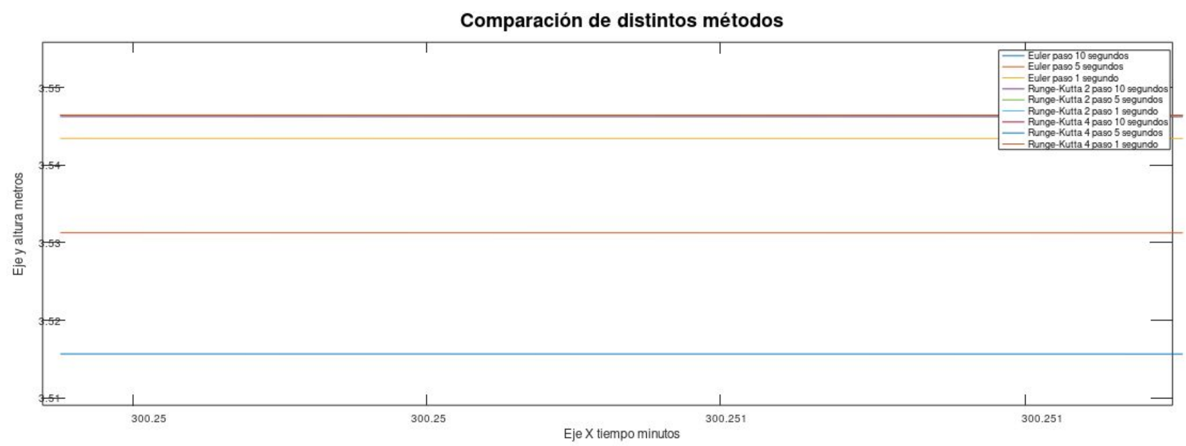


Gráfico 6: Comparación de métodos

Cuando se visualiza en una gráfica, se puede apreciar de abajo para arriba en línea celeste, el método de Euler con un paso de 10 segundos, luego Euler con un paso de 5 segundos en color rojo, y en amarillo, el método de Euler con un paso de 1 segundo. Lo que verifica que el método de euler tiene menor precisión, y los métodos de Runge-Kutta cuentan con menor error.

Para la siguiente tabla, donde se visualizan los errores de truncamiento, se utilizará como valor exacto de la solución el obtenido por el método de Runge Kutta-4 a los 10 minutos cuyo valor es 1.285761253.

Paso	Euler	Runge-Kutta 2	Runge-Kutta 4
10 segundos	0.03	$4 \cdot 10^{-4}$	$4 \cdot 10^{-7}$
5 segundos	0.01	$9 \cdot 10^{-5}$	$2 \cdot 10^{-8}$
1 segundo	0.003	$3 \cdot 10^{-6}$	0

Tabla 5: Errores de los métodos

La tabla número 5 resume las conclusiones obtenidas en los comentarios anteriores.

Por último, se calcula la razón de errores según pasos de discretización sucesivos. En este caso, no hay relación en el caso de Runge-Kutta 4 porque se supone que es el valor exacto.

Paso	Euler	Runge-Kutta 2	Runge-Kutta 4
10 segundos	-	-	-
5 segundos	2.032139327	4.174229717	16.454545421
1 segundo	5.062503565	25.868083868	-

Tabla 6: Razón de errores de los métodos

En este caso, se demuestra que al reducir los pasos de a mitades, se cumplen las proporciones entre los errores esperados en los órdenes de magnitud. En el método de Euler se obtiene del orden de 2^1 , en el método de Runge-Kutta 2 el orden de 2^2 y por último en el método de Runge-Kutta 4 del orden de 2^4 .

Análisis de sensibilidad

Se procede a realizar un análisis de sensibilidad utilizando Runge-Kutta 4. Al modelo presentado anteriormente, se suma un Coeficiente de contracción “Cc” que toma los valores de 0.65 y 0.55. Por lo tanto, el modelo que se utilizó fue el siguiente:

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

La única variación con el modelo anterior es multiplicar el factor agregado y se obtiene como resultado final los valores que se muestran en la tabla 7. Se considera como Coeficiente de contracción = 1, al del modelo utilizado anteriormente.

Paso	Coeficiente contracción = 1	Coeficiente contracción = 0.65	Coeficiente contracción = 0.55
10 segundos	1.285760891	2.878176646	3.322109039
5 segundos	1.285761231	2.878176697	3.322109065
1 segundo	1.285761253	2.878176700	3.322109067

Tabla 7: Análisis de sensibilidad

Algo que se observa a simple vista en la tabla, es la variación de la altura final, aumentando aproximadamente un 135% al disminuir el coeficiente de contracción en un valor de 0.35. De todos modos, al reducir en valor de 0.10 se puede notar una mayor variación entre los valores. Esto se puede observar en el siguiente gráfico de manera más clara.

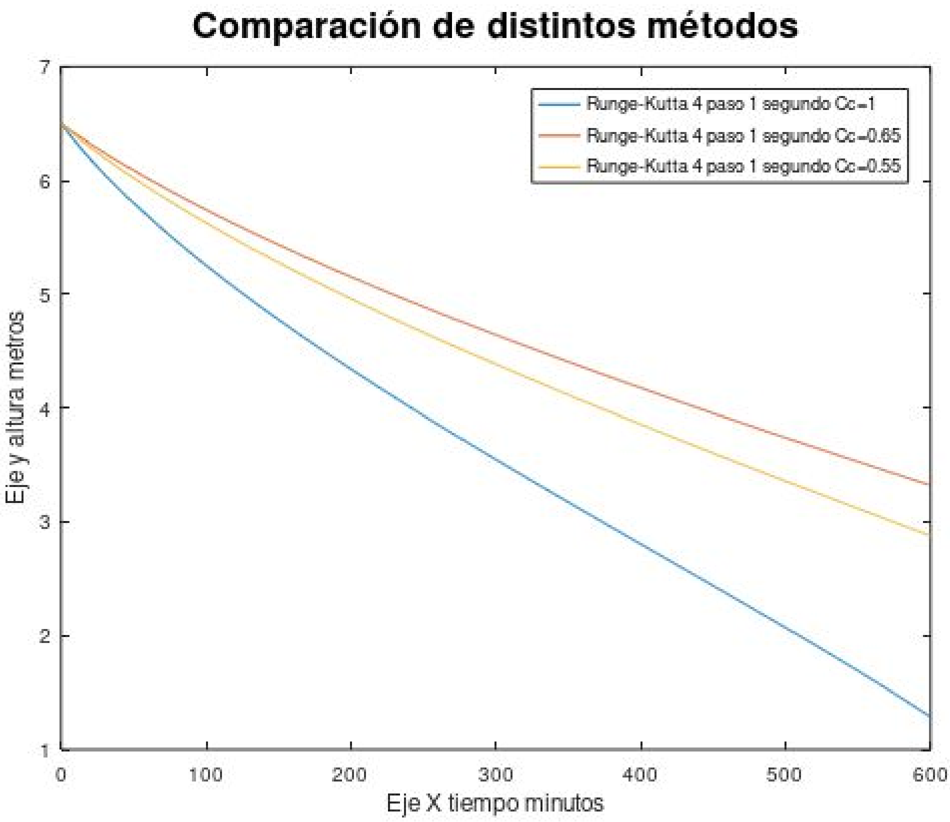


Gráfico 7: Análisis de sensibilidad.

Se aprecia en el gráfico que se observa una variación mucho mayor al disminuir 0.10 el Coeficiente de Contracción entre 0.65 y 0.55 que entre 1 y 0.65.

Calculo de integral

Luego del análisis realizado, se utilizan los datos recolectados para el cálculo de las integrales durante el primer minuto (desde el segundo 0 a 60) y del último minuto (segundo 540 a 600).

Se utilizan dos métodos, el método de Simpson y el método de los trapecios que se pueden modelar de la siguiente manera:

Método del trapecio:

$$T(h) = \frac{h}{2} * (f(a) + f(b) + 2 * \sum_{k=1}^{i-1} f(x_i))$$

Donde $\frac{h}{2}$ paso de cálculo divido 2, $f(a)\ y\ f(b)$ los valores de la función en los extremos y $\sum_{k=1}^{i-1} f(x_i)$ la sumatoria de los valores de la función en el interior del intervalo (no tiene en cuenta los extremos).

Método de Simpson:

Es muy similar al método del trapecio pero varía en la ponderación de los valores internos de los datos obtenidos. En este caso, el paso de cálculo se divide 3 y los nodos pares se multiplican por 2 y los nodos impares se multiplican por 4.

$$T(h) = \frac{h}{3} * (f(a) + f(b) + 2 * \sum_{k=1}^{(n-2)/2} f(x_{2k}) + 4 * \sum_{k=0}^{(n-2)/2} f(x_{2k+1}))$$

Resultados de integración

En este caso, se pueden observar los siguientes resultados:

Método de Trapecio	Primer Minuto	Último minuto
Paso 10 segundos	364,123066	166,913089
Paso 5 segundos	364,091130	91,731022
Paso 1 segundo	364,080890	91,732379

Tabla 8: Resultados de integración trapecio

Con el método del trapecio, sin importar el paso utilizado se logran resultados bastantes similares en el primer minuto. De todos modos, no sucede lo mismo en el último minuto, debido a que el paso de 10 segundos presenta un valor mucho mayor.

Ahora se muestran los resultados obtenidos por el método de Simpson.

Método de Simpson	Primer Minuto	Último minuto
Paso 10 segundos	364.080686	91.732407

Paso 5 segundos	364.080477	109.895863
Paso 1 segundo	364.080463	91.732436

Tabla 9: Resultados de integración Simpson

En este caso, se pueden observar los valores que se obtienen mediante el método de Simpson. La variación entre los resultados es mayor que en el método del trapecio. Resulta llamativo los valores hallados en el último minuto.

3. Conclusión

Se concluye en el informe luego del análisis de distintas metodologías para calcular soluciones de ecuaciones diferenciales de primer orden que el método de Runge-Kutta presenta menor error que el método de Euler, y a pesar de la variación que hay al cambiar los pasos, no hay diferencias sustanciales entre ellos en las primeras cifras significativas.

Además, resulta llamativo con el análisis de sensibilidad como varía el modelo cambiando el coeficiente de contracción en 0.10 entre los valores de 0.65 y 0.55. Este punto es de importancia para el modelo y genera grandes cambios en la modelación.

Por último, se analiza dos métodos de integración numérica y se visualiza que el método del Trapecio tuvo resultados más similares entre pasos, situación que no sucede con el cambio de paso con el método de Simpson.

4. Anexo: Código de programación

```
output_precision(10)

g = 9.81;
R = 4;
r = 0.12;

f = @(t,x) -(((r^2)*sqrt(2*g*x))/((2*x*R)-((x)^2)));
t0 = 0;
y0 = 6.5;

h = 10;
T = 600;
N = ceil((T - t0)/h);

% Metodo de Euler con un paso de 10 segundos
function [tt, yy] = euler(f,t0,y0,h,N)
 k = N+1;
 tt = zeros(k,1);
 yy = zeros(k,1);
 tt(1) = t0;
 yy(1) = y0;
 for i = 2:k
 tt(i) = tt(i-1) + h;
 m = f(tt(i-1),yy(i-1));
 yy(i) = yy(i-1) + h*m;
 end
end

[tt ye] = euler(f,t0,y0,h,N);
'Euler con paso de 10 segundos'
[tt ye];

%Cambio el paso a 5 segundos

g = 9.81;
R = 4;
r = 0.12;

f = @(t,x) -(((r^2)*sqrt(2*g*x))/((2*x*R)-((x)^2)));
t0 = 0;
y0 = 6.5;

h1 = 5;
T = 600;
```

```
N = ceil((T - t0)/h1);


[tt1 ye1] = euler(f,t0,y0,h1,N);
'Euler con paso de 5 segundos'
[tt1 ye1];

%Cambio el paso a 1 segundo

g = 9.81;
R = 4;
r = 0.12;

f = @(t,x) -(((r^2)*sqrt(2*g*x))/((2*x*R)-((x)^2)));
t0 = 0;
y0 = 6.5;

h2 = 1;
T = 600;
N = ceil((T - t0)/h2);

[tt2 ye2] = euler(f,t0,y0,h2,N);
'Euler con paso de 1 segundos'
[tt2 ye2];


%Ahora voy a usar el método de Runga Kutta 2 con un paso de tiempo 10 segundos

g = 9.81;
R = 4;
r = 0.12;

f = @(t,x) -(((r^2)*sqrt(2*g*x))/((2*x*R)-((x)^2)));
t0 = 0;
y0 = 6.5;

h3 = 10;
T = 600;
N = ceil((T - t0)/h3);

function [tt3, yy3] = rk2(f,t0,y0,h3,N)
 k = N+1;
 tt3 = zeros(k,1);
 yy3 = zeros(k,1);
```

```
 tt3(1) = t0;
 yy3(1) = y0;
 for i = 2:k
 tt3(i) = tt3(i-1) + h3;
 m1 = f(tt3(i-1),yy3(i-1));
 m2 = f(tt3(i),yy3(i-1) + h3*m1);
 yy3(i) = yy3(i-1) + h3*(m1 + m2)/2;
 end
end

[tt3 yy3] = rk2(f,t0,y0,h3,N);
'Runge Kuta 2 con paso de 10 segundos'
[tt3 yy3];

%Ahora voy a usar el método de Runga Kutta 2 con un paso de tiempo 5 segundos

g = 9.81;
R = 4;
r = 0.12;

f = @(t,x) -(((r^2)*sqrt(2*g*x))/((2*x*R)-((x)^2)));
t0 = 0;
y0 = 6.5;

h4 = 5;
T = 600;
N = ceil((T - t0)/h4);

[tt4 yy4] = rk2(f,t0,y0,h4,N);
'Runge Kuta 2 con paso de 5 segundos'
[tt4 yy4];

%Ahora voy a usar el método de Runga Kutta 2 con un paso de tiempo 1 segundos

g = 9.81;
R = 4;
r = 0.12;

f = @(t,x) -(((r^2)*sqrt(2*g*x))/((2*x*R)-((x)^2)));
t0 = 0;
y0 = 6.5;

h5 = 1;
T = 600;
```

```
N = ceil((T - t0)/h5);

[tt5 yy5] = rk2(f,t0,y0,h5,N);
'Runge Kuta 2 con paso de 1 segundos'
[tt5 yy5];

%Usando el metodo de Runge-Kutta4
%Paso de 10 segundos

g = 9.81;
R = 4;
r = 0.12;

f = @(t,x) -(((r^2)*sqrt(2*g*x))/((2*x*R)-((x)^2)));
t0 = 0;
y0 = 6.5;

h6 = 10;
T = 600;
N = ceil((T - t0)/h6);


function [tt6, yy6] = rk4(f,t0,y0,h6,N)
 k = N+1;
 tt6 = zeros(k,1);
 yy6 = zeros(k,1);
 tt6(1) = t0;
 yy6(1) = y0;
 for i = 2:k
 tt6(i) = tt6(i-1) + h6;
 m1 = f(tt6(i-1),yy6(i-1));
 m2 = f(tt6(i-1) + (h6/2),yy6(i-1) + (h6/2)*m1);
 m3 = f(tt6(i-1) + (h6/2),yy6(i-1) + (h6/2)*m2);
 m4 = f(tt6(i),yy6(i-1) + h6*m3);
 yy6(i) = yy6(i-1) + h6*(m1 + 2*m2 + 2*m3 + m4)/6;
 end
end

[tt6 yy6] = rk4(f,t0,y0,h6,N);
'Runge Kuta 4 con paso de 10 segundos'
[tt6 yy6];

%Paso de 5 segundos
```

```
g = 9.81;
R = 4;
r = 0.12;

f = @(t,x) -(((r^2)*sqrt(2*g*x))/((2*x*R)-((x)^2)));
t0 = 0;
y0 = 6.5;

h7 = 5;
T = 600;
N = ceil((T - t0)/h7);

[tt7 yy7] = rk4(f,t0,y0,h7,N);
'Runge Kuta 4 con paso de 5 segundos'
[tt7 yy7];

%Paso de 1 segundos

g = 9.81;
R = 4;
r = 0.12;

f = @(t,x) -(((r^2)*sqrt(2*g*x))/((2*x*R)-((x)^2)));
t0 = 0;
y0 = 6.5;

h8 = 1;
T = 600;
N = ceil((T - t0)/h8);


[tt8 yy8] = rk4(f,t0,y0,h8,N);
'Runge Kuta 4 con paso de 1 segundos'
[tt8 yy8];

%Voy a realizar un analisis de sensibilidad. Para este caso voy a usar el modelo M2 que
agrega un coeficiente CC

%Paso de 10 segundos Cc=0.55

g = 9.81;
R = 4;
r = 0.12;
c = 0.55;
```

```
f1 = @(t,x) -(((r^2)*(c*sqrt(2*g*x)))/((2*x*R)-((x)^2)));
t0 = 0;
y0 = 6.5;

h9 = 10;
T = 600;
N = ceil((T - t0)/h9);


function [tt9, yy9] = rk4cc1(f1,t0,y0,h9,N)
 k = N+1;
 tt9 = zeros(k,1);
 yy9 = zeros(k,1);
 tt9(1) = t0;
 yy9(1) = y0;
 for i = 2:k
 tt9(i) = tt9(i-1) + h9;
 m1 = f1(tt9(i-1),yy9(i-1));
 m2 = f1(tt9(i-1) + (h9/2),yy9(i-1) + (h9/2)*m1);
 m3 = f1(tt9(i-1) + (h9/2),yy9(i-1) + (h9/2)*m2);
 m4 = f1(tt9(i),yy9(i-1) + h9*m3);
 yy9(i) = yy9(i-1) + h9*(m1 + 2*m2 + 2*m3 + m4)/6;
 end
end

[tt9 yy9] = rk4cc1(f1,t0,y0,h9,N);
'Runge Kuta 4 con paso de 10 segundos cc 0.55'
[tt9 yy9];

%Paso de 5 segundos Cc=0.55

g = 9.81;
R = 4;
r = 0.12;
c = 0.55;

f1 = @(t,x) -(((r^2)*(c*sqrt(2*g*x)))/((2*x*R)-((x)^2)));
t0 = 0;
y0 = 6.5;

h10 = 5;
T = 600;
N = ceil((T - t0)/h10);
```

```
[tt10, yy10] = rk4cc1(f1,t0,y0,h10,N);
[tt10, yy10];

%Paso de 1 segundos Cc=0.55

g = 9.81;
R = 4;
r = 0.12;
c = 0.55;

f1 = @(t,x) -(((r^2)*(c*sqrt(2*g*x)))/((2*x*R)-((x)^2)));
t0 = 0;
y0 = 6.5;

h11 = 1;
T = 600;
N = ceil((T - t0)/h11);


[tt11, yy11] = rk4cc1(f1,t0,y0,h11,N);
[tt11, yy11];

%Paso de 10 segundos Cc=0.65

g = 9.81;
R = 4;
r = 0.12;
c = 0.65;

f1 = @(t,x) -(((r^2)*(c*sqrt(2*g*x)))/((2*x*R)-((x)^2)));
t0 = 0;
y0 = 6.5;

h12 = 10;
T = 600;
N = ceil((T - t0)/h12);


[tt12, yy12] = rk4cc1(f1,t0,y0,h12,N);
[tt12, yy12];

%Paso de 5 segundos Cc=0.65
```

```
g = 9.81;
R = 4;
r = 0.12;
c = 0.65;

f1 = @(t,x) -(((r^2)*(c*sqrt(2*g*x)))/((2*x*R)-((x)^2)));
t0 = 0;
y0 = 6.5;

h13 = 5;
T = 600;
N = ceil((T - t0)/h13);

[tt13, yy13] = rk4cc1(f1,t0,y0,h13,N);
[tt13, yy13];

%Paso de 1 segundos Cc=0.65

g = 9.81;
R = 4;
r = 0.12;
c = 0.65;

f1 = @(t,x) -(((r^2)*(c*sqrt(2*g*x)))/((2*x*R)-((x)^2)));
t0 = 0;
y0 = 6.5;

h14 = 1;
T = 600;
N = ceil((T - t0)/h14);

[tt14, yy14] = rk4cc1(f1,t0,y0,h14,N);
[tt14, yy14];

%Metodo del trapecio para RK4 con cc=1 primer minuto
%Vector solucion de con paso de 10 segundos [tt6 yy6]
'Runge Kuta 4 con paso de 10 segundos cc 1 primer minuto'

puntoa = tt6(1)
puntob = tt6(7)
fa = yy6(1)
fb= yy6(7)
```

```
puntomedio = 10/2
suma =0;

for i =2:6
  suma = yy6(i) + suma;
  suma = suma;
endfor
suma;

integralRK10min1 = puntomedio*(fa + fb + 2*suma);
integralRK10min1

%Metodo del trapecio para RK4 con cc=1 minuto 9 a 10
%Vector solucion de con paso de 10 segundos [tt6 yy6]
'Runge Kuta 4 con paso de 10 segundos cc 1 minuto 9 a 10'

puntoa = tt6(55)
puntob = tt6(61)
fa = yy6(55)
fb= yy6(61)

puntomedio = 10/2
suma =0;

for i =52:60
  suma = yy6(i) + suma;
  suma = suma;
endfor
suma;

integralRK10min9 = puntomedio*(fa + fb + 2*suma);
integralRK10min9

%Metodo del trapecio para RK4 con cc=0.65 primer minuto
%Vector solucion de con paso de 10 segundos [tt9 yy9]
'Runge Kuta 4 con paso de 10 segundos cc 0.55 primer minuto'

puntoa = tt9(1)
puntob = tt9(7)
fa = yy9(1)
fb= yy9(7)

puntomedio = 10/2
suma =0;
```

```
for i =2:6
  suma = yy9(i) + suma;
  suma = suma;
endfor
suma;

integralRK10cc055min1 = puntomedio*(fa + fb + 2*suma);
integralRK10cc055min1

%Metodo del trapecio para RK4 con cc=0.65 primer minuto
%Vector solucion de con paso de 10 segundos [tt9 yy9]
'Runge Kuta 4 con paso de 10 segundos cc 0.55 minuto 9 a 10'

puntoa = tt9(55)
puntob = tt9(61)
fa = yy9(55)
fb= yy9(61)

puntomedio = 10/2
suma =0;

for i =56:60
  suma = yy9(i) + suma;
  suma = suma;
endfor
suma;

integralRK10cc055min9 = puntomedio*(fa + fb + 2*suma);
integralRK10cc055min9

%Metodo del trapecio para RK4 con cc=0.65 primer minuto
%Vector solucion de con paso de 10 segundos [tt12 yy12]
'Runge Kuta 4 con paso de 10 segundos cc 0.65 primer minuto'

puntoa = tt12(1)
puntob = tt12(7)
fa = yy12(1)
fb= yy12(7)

puntomedio = 10/2
suma =0;

for i =2:6
```

```
  suma = yy12(i) + suma;
  suma = suma;
endfor
suma;

integralRK10cc065min1 = puntomedio*(fa + fb + 2*suma);
integralRK10cc065min1

%Metodo del trapecio para RK4 con cc=0.65 minuto 9 a 10
%Vector solucion de con paso de 10 segundos [tt12 yy12]
'Runge Kuta 4 con paso de 10 segundos cc 0.65 minuto 9 a 10'

puntoa = tt12(55)
puntob = tt12(61)
fa = yy12(55)
fb= yy12(61)

puntomedio = 10/2
suma =0;

for i =56:60
  suma = yy12(i) + suma;
  suma = suma;
endfor
suma;

integralRK10cc065min9 = puntomedio*(fa + fb + 2*suma);
integralRK10cc065min9

%Metodo del trapecio para RK4 con cc=1 primer minuto
%Vector solucion de con paso de 5 segundos [tt7 yy7]
'Runge Kuta 4 con paso de 5 segundos cc1 primer minuto'

puntoa = tt7(1)
puntob = tt7(13)
fa = yy7(1)
fb= yy7(13)

puntomedio = 5/2
suma =0;

for i =2:12
  suma = yy7(i) + suma;
  suma = suma;
```

```
endfor
suma;

integralRK5cc1min1 = puntomedio*(fa + fb + 2*suma);
integralRK5cc1min1

%Metodo del trapecio para RK4 con cc=1 minuto 9 a 10
%Vector solucion de con paso de 5 segundos [tt7 yy7]
'Runge Kuta 4 con paso de 5 segundos cc1 minuto 9 a 10'

puntoa = tt7(109)
puntob = tt7(121)
fa = yy7(109)
fb= yy7(121)

puntomedio = 5/2
suma =0;

for i =110:120
  suma = yy7(i) + suma;
  suma = suma;
endfor
suma;

integralRK5cc1min9 = puntomedio*(fa + fb + 2*suma);
integralRK5cc1min9

%Metodo del trapecio para RK4 con cc=0.55 primer minuto
%Vector solucion de con paso de 5 segundos [tt10 yy10]
'Runge Kuta 4 con paso de 5 segundos cc 0.55 primer minuto'

puntoa = tt10(1)
puntob = tt10(13)
fa = yy10(1)
fb= yy10(13)

puntomedio = 5/2
suma =0;

for i =2:12
  suma = yy10(i) + suma;
  suma = suma;
endfor
suma;
```

```
integralRK5cc55 = puntomedio*(fa + fb + 2*suma);
integralRK5cc55

%Metodo del trapecio para RK4 con cc=0.55 minuto 9 a 10
%Vector solucion de con paso de 5 segundos [tt10 yy10]
'Runge Kuta 4 con paso de 5 segundos cc 0.55 minuto 9 a 10'

puntoa = tt10(109)
puntob = tt10(121)
fa = yy10(109)
fb= yy10(121)

puntomedio = 5/2
suma =0;

for i =110:120
  suma = yy10(i) + suma;
  suma = suma;
endfor
suma;

integralRK5cc55min9 = puntomedio*(fa + fb + 2*suma);
integralRK5cc55min9

%Metodo del trapecio para RK4 con cc=0.65 primer minuto
%Vector solucion de con paso de 5 segundos [tt13 yy13]
'Runge Kuta 4 con paso de 5 segundos cc 0.65 primer minuto'

puntoa = tt13(1)
puntob = tt13(13)
fa = yy13(1)
fb= yy13(13)

puntomedio = (5)/2
suma =0;

for i =2:12
  suma = yy13(i) + suma;
  suma = suma;
endfor
suma;

integralRK5cc65min1 = puntomedio*(fa + fb + 2*suma);
```

```
integralRK5cc65min1

%Metodo del trapecio para RK4 con cc=0.65 minuto 9 a 10
%Vector solucion de con paso de 5 segundos [tt13 yy13]
'Runge Kuta 4 con paso de 5 segundos cc 0.65 minuto 9 a 10'

puntoa = tt13(109)
puntob = tt13(121)
fa = yy13(109)
fb= yy13(121)

puntomedio = 5/2
suma =0;

for i =110:120
  suma = yy13(i) + suma;
  suma = suma;
endfor
suma;

integralRK5cc65min9 = puntomedio*(fa + fb + 2*suma);
integralRK5cc65min9

%Metodo del trapecio para RK4 con cc=1 primer minuto
%Vector solucion de con paso de 1 segundos [tt8 yy8]
'Runge Kuta 4 con paso de 1 segundos cc1 primer minuto'

puntoa = tt8(1)
puntob = tt8(61)
fa = yy8(1)
fb= yy8(61)

puntomedio = 1/2
suma =0;

for i =2:60
  suma = yy8(i) + suma;
  suma = suma;
endfor
suma;

integralRK1cc1min1 = puntomedio*(fa + fb + 2*suma);
integralRK1cc1min1
```

```
%Metodo del trapecio para RK4 con cc=1 minuto 9 a 10
%Vector solucion de con paso de 1 segundos [tt8 yy8]
'Runge Kuta 4 con paso de 1 segundos cc1 minuto 9 a 10'

puntoa = tt8(541)
puntob = tt8(601)
fa = yy8(541)
fb= yy8(601)

puntomedio = 1/2
suma =0;

for i =542:600
  suma = yy8(i) + suma;
  suma = suma;
endfor
suma;

integralRK1cc1min9 = puntomedio*(fa + fb + 2*suma);
integralRK1cc1min9

%Metodo del trapecio para RK4 con cc=0.55 primer minuto
%Vector solucion de con paso de 1 segundos [tt11 yy11]
'Runge Kuta 4 con paso de 1 segundos cc 0.55 primer minuto'

puntoa = tt11(1)
puntob = tt11(61)
fa = yy11(1)
fb= yy11(61)

puntomedio = 1/2
suma =0;

for i =2:60
  suma = yy11(i) + suma;
  suma = suma;
endfor
suma;

integralRK1cc55min9 = puntomedio*(fa + fb + 2*suma);
integralRK1cc55min9

%Metodo del trapecio para RK4 con cc=0.55 minuto 9 a 10
%Vector solucion de con paso de 1 segundos [tt11 yy11]
```

```
'Runge Kuta 4 con paso de 1 segundos cc 0.55 minuto 9 a 10'

puntoa = tt11(541)
puntob = tt11(601)
fa = yy11(541)
fb= yy11(601)

puntomedio = 1/2
suma =0;

for i =542:600
  suma = yy11(i) + suma;
  suma = suma;
endfor
suma;

integralRK1cc55min9 = puntomedio*(fa + fb + 2*suma);
integralRK1cc55min9

%Metodo del trapecio para RK4 con cc=0.65 primer minuto
%Vector solucion de con paso de 5 segundos [tt14 yy14]
'Runge Kuta 4 con paso de 1 segundos cc 0.65'

puntoa = tt14(1)
puntob = tt14(61)
fa = yy14(1)
fb= yy14(61)

puntomedio = 1/2
suma =0;

for i =2:60
  suma = yy14(i) + suma;
  suma = suma;
endfor
suma;

integralRK1cc65min1 = puntomedio*(fa + fb + 2*suma);
integralRK1cc65min1

%Metodo del trapecio para RK4 con cc=0.65 minuto 9 a 10
%Vector solucion de con paso de 5 segundos [tt14 yy14]
'Runge Kuta 4 con paso de 1 segundos cc 0.65 minuto 9 a 10'
```

```
puntoa = tt14(541)
puntob = tt14(601)
fa = yy14(541)
fb= yy14(601)

puntomedio = 1/2
suma =0;

for i =542:600
  suma = yy14(i) + suma;
  suma = suma;
endfor
suma;

integralRK1cc65min9 = puntomedio*(fa + fb + 2*suma);
integralRK1cc65min9

   %%integral de simpson
%Metodo de simpson para RK4 con cc=1 primer minuto
%Vector solucion de con paso de 10 segundos [tt6 yy6]
'Simpson con paso de 10 segundos cc 1 primer minuto'

% En la siguiente matriz llamada tablaResultados
% Cada columna tiene los pasos, o sea que esta matriz tiene 3 columnas: paso 10, 5 y 1
tablaResultados = [] % Aca se guardan todos los resultados de simpson

tablaConDatos1 = [tt6(1:7)';yy6(1:7)'];
integrals10cc1min1 = simpson(10,tablaConDatos1);
tablaResultados(1,1) = integrals10cc1min1;

%Metodo de simpson para RK4 con cc=1 minuto 9 a 10
%Vector solucion de con paso de 10 segundos [tt6 yy6]
'Simpson con paso de 10 segundos cc 1 minuto 9 a 10'

tablaConDatos2 = [tt6(55:61)';yy6(55:61)'];
integrals10cc1min9 = simpson(10,tablaConDatos2);
tablaResultados(2,1) = integrals10cc1min9;

            %%integral de simpson
%Metodo de simpson para RK4 con cc=1 primer minuto
%Vector solucion de con paso de 10 segundos [tt6 yy6]
'Simpson con paso de 10 segundos cc 0.55 primer minuto'

tablaConDatos3 = [tt9(1:7)';yy9(1:7)'];
```

```
integrals10cc55min1 = simpson(10,tablaConDatos3);
tablaResultados(3,1) = integrals10cc55min1;

%Metodo de simpson para RK4 con cc=1 minuto 9 a 10
%Vector solucion de con paso de 10 segundos [tt6 yy6]
'Simpson con paso de 10 segundos cc 0.55 minuto 9 a 10'

tablaConDatos4 = [tt9(55:61)';yy9(55:61)'];
integrals10cc55min9 = simpson(10,tablaConDatos4);
tablaResultados(4,1) = integrals10cc55min9;

        %%integral de simpson
%Metodo de simpson para RK4 con cc=1 primer minuto
%Vector solucion de con paso de 10 segundos [tt6 yy6]
'Simpson con paso de 10 segundos cc 0.65 primer minuto'

tablaConDatos5 = [tt12(1:7)';yy12(1:7)'];
integrals10cc65min1 = simpson(10,tablaConDatos5);
tablaResultados(5,1) = integrals10cc65min1;

%Metodo de simpson para RK4 con cc=1 minuto 9 a 10
%Vector solucion de con paso de 10 segundos [tt6 yy6]
'Simpson con paso de 10 segundos cc 0.55 minuto 9 a 10'

tablaConDatos6 = [tt12(55:61)';yy12(55:61)'];
integrals10cc65min9 = simpson(10,tablaConDatos6);
tablaResultados(6,1) = integrals10cc65min9;

  %%integral de simpson
%Metodo de simpson para RK4 con cc=1 primer minuto
%Vector solucion de con paso de 5 segundos [tt7 yy7]
'Simpson con paso de 5 segundos cc 1 primer minuto'

tablaConDatos7 = [tt7(1:13)';yy7(1:13)'];
integrals5cc1min1 = simpson(5,tablaConDatos7);
tablaResultados(1,2) = integrals5cc1min1;

%Metodo de simpson para RK4 con cc=1 minuto 9 a 10
%Vector solucion de con paso de 5 segundos [tt7 yy7]
'Simpson con paso de 5 segundos cc 1 minuto 9 a 10'

tablaConDatos8 = [tt7(55:61)';yy7(55:61)'];
integrals5cc1min9 = simpson(5,tablaConDatos8);
tablaResultados(2,2) = integrals5cc1min9;
```

```
        %%integral de simpson
%Metodo de simpson para RK4 con cc=1 primer minuto
%Vector solucion de con paso de 5 segundos [tt7 yy7]
'Simpson con paso de 5 segundos cc 0.55 primer minuto'

tablaConDatos9 = [tt10(1:13)';yy10(1:13)'];
integrals5cc55min1 = simpson(5,tablaConDatos9);
tablaResultados(3,2) = integrals5cc55min1;

%Metodo de simpson para RK4 con cc=1 minuto 9 a 10
%Vector solucion de con paso de 5 segundos [tt7 yy7]
'Simpson con paso de 5 segundos cc 0.55 minuto 9 a 10'

tablaConDatos10 = [tt10(55:61)';yy10(55:61)'];
integrals5cc55min9 = simpson(5,tablaConDatos10);
tablaResultados(4,2) = integrals5cc55min9;

        %%integral de simpson
%Metodo de simpson para RK4 con cc=1 primer minuto
%Vector solucion de con paso de 5 segundos [tt7 yy7]
'Simpson con paso de 5 segundos cc 0.65 primer minuto'

tablaConDatos11 = [tt13(1:13)';yy13(1:13)'];
integrals5cc65min1 = simpson(5,tablaConDatos11);
tablaResultados(5,2) = integrals5cc65min1;

%Metodo de simpson para RK4 con cc=1 minuto 9 a 10
%Vector solucion de con paso de 5 segundos [tt7 yy7]
'Simpson con paso de 5 segundos cc 0.55 minuto 9 a 10'

tablaConDatos12 = [tt13(55:61)';yy13(55:61)'];
integrals5cc65min9 = simpson(5,tablaConDatos12);
tablaResultados(6,2) = integrals5cc65min9;

  %%integral de simpson
%Metodo de simpson para RK4 con cc=1 primer minuto
%Vector solucion de con paso de 1 segundo [tt8 yy8]
'Simpson con paso de 1 segundo cc 1 primer minuto'

tablaConDatos13 = [tt8(1:61)';yy8(1:61)'];
integrals1cc1min1 = simpson(1,tablaConDatos13);
tablaResultados(1,3) = integrals1cc1min1;
```

```
%Metodo de simpson para RK4 con cc=1 minuto 9 a 10
%Vector solucion de con paso de 1 segundo [tt8 yy8]
'Simpson con paso de 1 segundo cc 1 minuto 9 a 10'

tablaConDatos14 = [tt8(541:601)';yy8(541:601)'];
integrals1cc1min9 = simpson(1,tablaConDatos14);
tablaResultados(2,3) = integrals1cc1min9;

        %%integral de simpson
%Metodo de simpson para RK4 con cc=1 primer minuto
%Vector solucion de con paso de 1 segundo [tt8 yy8]
'Simpson con paso de 1 segundo cc 0.541 primer minuto'

tablaConDatos15 = [tt11(1:61)';yy11(1:61)'];
integrals1cc55min1 = simpson(1,tablaConDatos15);
tablaResultados(3,3) = integrals1cc55min1;

%Metodo de simpson para RK4 con cc=1 minuto 9 a 10
%Vector solucion de con paso de 1 segundo [tt8 yy8]
'Simpson con paso de 1 segundo cc 0.55 minuto 9 a 10'

tablaConDatos16 = [tt11(61:541)';yy11(61:541)'];
integrals1cc55min9 = simpson(1,tablaConDatos16);
tablaResultados(4,3) = integrals1cc55min9;

        %%integral de simpson
%Metodo de simpson para RK4 con cc=1 primer minuto
%Vector solucion de con paso de 1 segundo [tt8 yy8]
'Simpson con paso de 1 segundo cc 0.65 primer minuto'

tablaConDatos17 = [tt14(1:61)';yy14(1:61)'];
integrals1cc65min1 = simpson(1,tablaConDatos17);
tablaResultados(5,3) = integrals1cc65min1;


%Metodo de simpson para RK4 con cc=1 minuto 9 a 10
%Vector solucion de con paso de 1 segundo [tt8 yy8]
'Simpson con paso de 1 segundo cc 0.55 minuto 9 a 10'

tablaConDatos18 = [tt14(61:541)';yy14(61:541)'];
integrals1cc65min9 = simpson(1,tablaConDatos18);
tablaResultados(6,3) = integrals1cc65min9;

% Muestra la tabla con todos los resultados
```

```
%
%En la tabla que se muestra a continuacion la primera fila corresponde
%al primer minuto y la segunda al minuto de 9 a 10, ambas con cc = 1.
%La tercera fila corresponde al primer minuto y la cuarta al minuto 9 a 10 con cc=0.55
%La quinta fila corresponde al primer minuto y la cuarta al minuto 9 a 10 con cc=0.65
%
%Los datos de la primera columna se calcularon con paso 10, la segunda con paso 5
%y la ultima con paso 1

disp('')
disp('TABLA CON RESULTADOS DE SIMPSON')
disp('    PASO 10         PASO 5          PASO 1')
disp('--------------------------------------------------')
disp(tablaResultados);

%plot(tt,ye,'g--',tt3,yy3,'b--',tt6,yy6,'r--')
%xlabel('Eje X tiempo','fontsize',12)
%ylabel('Eje y altura','fontsize',12)
%title('Comparacion metodos con paso 10 segundos','fontsize',18)
%legend('Euler','Runge-Kutta 2','Runge-Kutta 4')
%xlim ([560, 600])
%ylim([1, 1.8])

%plot(tt,ye,'g--',tt1,ye1,'b--',tt2,ye2,'r--')
%xlabel('Eje X tiempo minutos','fontsize',12)
%ylabel('Eje y altura metros','fontsize',12)
%title('Comparacion Euler distintos pasos','fontsize',18)
%legend('Euler paso 10 segundos','Euler paso 5 segundos','Euler paso 1 segundo')
%xlim ([15, 30])
%ylim ([6, 7])
%print(plot)

plot(tt8,yy8,tt11,yy11,tt14,yy14)
xlabel('Tiempo(minutos)','fontsize',12)
ylabel('Altura(mts)','fontsize',12)
title('Comparacion de distintos metodos','fontsize',18)
legend('Runge-Kutta 4 paso 1 segundo Cc=1','Runge-Kutta 4 paso 1 segundo
Cc=0.65','Runge-Kutta 4 paso 1 segundo Cc=0.55')
```

Regla del Trapecio

```
% Calcula aproximacion de una integral utilizando el metodo del trapecio
%
% datos es la tabla que tiene 2 filas y n columnas con los datos. Donde la primera
% fila son los x y la segunda son los f(x)
% lim_inf es el limite inferior de la integral
% lim_sup es el limite superior de la integral
% h es el paso de discretizacion
%
% Devuelve la aproximacion calculada
function res = trapecio(f,lim_inf,lim_sup,h)
  datos = crear_tabla(@f,lim_inf,lim_sup,h);
  dimension = columns(datos);

  # Evaluacion en los extremos
  f_a = datos(2,1);
  f_b = datos(2,dimension);

  # Calcula la sumatoria
  suma = 0;
  for i=2:dimension-1
        suma += datos(2,i);
  endfor

  res = (h./2).*(f_a + f_b + (2.*suma));
endfunction
```

Tabla utilizada por el algoritmo del trapcio

```
% Crea tabla de datos que se usa con el metodo del trapecio. En el caso de simpson
% En este trabajo le se le pasa la tabla ya construida de forma manual, no se usa esta
% funcion. Esta tabla tiene
% En la primera fila los valores t y en la segunda los valores f(t).
% f es la funcion que se quiere integrar
% lim_inf es el limite de integracion inferior
% lim_sup es el limite de integracion superior
function tabla_resultado = crear_tabla(f,lim_inf,lim_sup,h)
  y = f(lim_inf);
  x_sig = lim_inf + h;
```

```
  tabla = [lim_inf; y];
  columna = 2;
  while (x_sig <= lim_sup)
          tabla(1,columna) = x_sig;
          tabla(2,columna) = f(x_sig);
          x_sig += h;
          columna += 1;
  endwhile
  tabla_resultado = tabla;
endfunction
```

Regla de Simpson

```
function res = simpson(h,datos)
  % h es el paso
  % datos es una matriz con 2 filas y varias columnas, en la primera fila
  % se tiene los valores t y en la segunda fila los f(t).

  dimension = columns(datos);
  suma_pares = 0;
  suma_impares = 0;
  f_a = datos(2,1);
  f_b = datos(2,dimension);

  % Sumatoria de pares
  for i=2:round((dimension-2)./2)
          suma_pares += datos(2,2*i-1); # En realidad 2*i-1 es siempre impar pero esto se
debe a que el indice de octave comienza en 1 y el de los ejercicios de la guia comienzan en
0.
  endfor

  % Sumatoria de impares
  for i=1:round((dimension-2)./2)
          suma_impares += datos(2,2*i); # Como en el caso anterior 2*i es par siempre pero
esto se debe a que los indices de octave y nuestros ejercicios empiezan diferente.
  endfor

  res = ((h/3).*(f_a + f_b + (2.*suma_pares) + (4.*suma_impares)));
endfunction
```