Departamento de Matemática Aplicada a la Ingeniería Industrial

Guion para la Práctica 1 de Ecuaciones Diferenciales

Una introducción a la resolución numérica de EDO de primer orden

---

Índice

---

1. Conocimientos previos en MatLab

En este apartado se explican herramientas que más tarde utilizaremos en MatLab. También se recomienda consultar la documentación adjunta en la carpeta *Documentos de ayuda* en Moodle, que incluye una guía de iniciación a MatLab que explica en detalle otras herramientas elementales que pueden resultar de utilidad (trabajo con vectores, uso de funciones elementales o implementación de gráficos bidimensionales).

1.1. Scripts

En MatLab, podemos escribir las instrucciones que queremos ejecutar bien individualmente en la ventana de comandos (en inglés, *Command Window*) o bien utilizando un *script*, un archivo con extensión .m que contiene varias líneas secuenciales de comandos y llamadas a funciones de MatLab. Su utilización es recomendable cuando se quiere modificar un algoritmo secuencialmente.

- Para crear un *script* pulse el botón **Home→New** o **Home→New Script**. Se abrirá en el editor de MatLab un fichero llamado *Untitled.m*, cuyo nombre puede modificarse pulsando el botón **Home→Save as**.
- Una vez creado el *script*, puede introducir en él las líneas de código de su algoritmo de MatLab.
- Para ejecutar su *script*, tiene dos opciones: o bien pulsar el botón **Editor→Run** o bien teclear el nombre de su *script* en la ventana de comandos.

1.2. Funciones anónimas

Algunas funciones "sencillas", que devuelvan el resultado de una expresión, se pueden definir mediante una sóla instrucción, en mitad de un programa (*script* o función) o en la línea de comandos. Se llaman funciones anónimas (en inglés, *function handle*).

Ejemplo

```
fun1 = @(x) sin(2*x);
fun1(pi/4) % evaluación de la función sobre el escalar pi/4
>> ans= 1
fun1([pi/8 pi/4 pi/2]) % evaluación de la función sobre el vector [pi/8 pi/4 pi/2]
>> ans= 0.7071   1.0000   0.0000
```

Pueden depender de varias variables y hacer uso de parámetros previamente definidos.

Ejemplo

```
a=2;
fun2 = @(x,t) cos(a*x*t);
fun2(pi/2,1)
>> ans= -1
```

Si, con posterioridad a la definición de la función `fun2`, se cambia el valor del parámetro `a`, la función no se modifica: en el caso del ejemplo, seguirá siendo `fun2(x,t)=cos(2*x*t)`.

En el caso en que la función anónima solo dependa de una variable, como es el caso de `fun1 = @(x) sin(2*x)`, podemos representarla gráficamente utilizando cualquiera de las técnicas que se explican a continuación:

1. Evaluando la función sobre un vector y seguidamente empleando el comando `plot()` de MatLab.

   ```
   x1=[-5:0.1:5]; % creamos el vector sobre el que evaluaremos la función.
   plot(x1, fun1(x1));
   ```

2. Empleando el comando `fplot()` de MatLab.

   ```
   fplot(fun1);
   ```

   Por defecto, esta herramienta dibuja la curva de la función en el intervalo $x \in [-5, 5]$. Para conocer más opciones puede teclear `help fplot`.

Observación. Si la función anónima solo depende de una variable pero está expresada de manera implícita, como es el caso de $y + e^y = x$, podemos representarla gráficamente utilizando el comando `fimplicit()` de MatLab.

```
fun3=@(x,y) y + exp(y)- x;
fimplicit(fun3);
```

Por defecto, esta herramienta dibuja la curva de la función en el intervalo $x \in [-5, 5]$. Para conocer más opciones puede teclear `help fimplicit`.

2. Introducción a los métodos numéricos

En esta práctica abordamos la resolución de un problema de valor inicial:

$$\left\{ \begin{array}{lcl} y'(t) & = & f(t, y(t)) \\ y(t_0) & = & y^0. \end{array} \right. \tag{1}$$

Supondremos que las funciones f y $\dfrac{\partial f}{\partial y}$ son continuas en un algún rectángulo $R := \{(t, y) : \ a < t < b, \ \ c < y < d\}$ que contiene al punto (t_0, y^0). En ese caso el teorema de existencia y unicidad asegura que existe una solución

única $y(t)$ del problema de valor inicial (1) en algún intervalo alrededor de t_0.

El objetivo consiste en estimar numéricamente el valor de dicha solución en algunos valores de t de un determinado intervalo de la forma $[t_0, T]$. Para ello, se realiza una partición del intervalo

$$t_0 < t_1 < t_2 < \cdots < t_n < \cdots < T$$

y se calculan valores aproximados $y^0, y^1, y^2, \ldots$ de la solución $y(t)$ en los puntos de la partición, esto es,

$$y^n \approx y(t_n), \qquad n = 1, 2, \ldots.$$

Nótese que el primer valor "aproximante" y^0 es exacto, puesto que el problema (1) impone la condición $y(t_0) = y^0$. A continuación se introducen dos métodos distintos para calcular los valores $y^1, y^2, \ldots$.

2.1. Método de Euler

El método de Euler permite calcular el valor $y^{n+1} \approx y(t_{n+1})$ a partir del valor anterior $y^n \approx y(t_n)$. La ley de recurrencia se deduce integrando ambos miembros de la ecuación diferencial del problema (1) en el intervalo $[t_n, t_{n+1}]$, para obtener

$$y(t_{n+1}) - y(t_n) = \int_{t_n}^{t_{n+1}} y'(s)ds = \int_{t_n}^{t_{n+1}} f(s, y(s))ds,$$

o bien,

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(s, y(s))ds. \tag{2}$$

Sin conocer $y(t)$ no se puede integrar $f(t, y(t))$, por lo que debemos aproximar la integral que aparece en la ecuación (2). Suponiendo que ya se ha encontrado y^n, el planteamiento más sencillo consiste en aproximarla por el área del rectángulo con base $(t_{n+1} - t_n)$ y altura $f(t_n, y(t_n))$. Esto da lugar a

$$y(t_{n+1}) \approx y(t_n) + (t_{n+1} - t_n)f(t_n, y(t_n)). \tag{3}$$

Si en la partición del intervalo $[t_0, T]$ se escogen puntos equiespaciados a distancia Δt, entonces el método de Euler viene dado por la expresión:

$$\boxed{y^{n+1} = y^n + \Delta t f(t_n, y^n) \qquad n = 0, 1, 2, \ldots} \tag{4}$$

Observación.- Si llamamos y^1 al valor aproximado de $y(t_1)$ lo que se tiene es

$$y^1 = y^0 + f(t_0, y^0)(t_1 - t_0).$$

La recta $y = y^0 + y'(t_0)(t - t_0)$ es la recta tangente a la solución en el punto (t_0, y^0), así que con este método lo que hacemos es tomar como y^1 el valor que está sobre la recta tangente en lugar de sobre la curva solución. Este procedimiento se emplea de manera recursiva para obtener $y^2, y^3, \ldots$ tal y como se muestra en la Figura 1. Si la función solución es suficientemente suave y los puntos de partición próximos, podemos pensar que estamos comentiendo un error pequeño. El estudio del error se escapa de esta pequeña introducción, lo abordaremos a través de los distintos ejemplos de esta práctica.

2.1.1. Ejemplo en MatLab

Vamos a plantear un problema de valor inicial sencillo del cual pueda obtenerse su solución analítica, por ejemplo

$$\begin{cases} y'(t) &= \cos(t)y(t) \\ y(0) &= 1 \end{cases} \tag{5}$$

cuya única solución es $y(t) = e^{sen(t)}$.

En MatLab abriremos un nuevo fichero *script* y haremos lo siguiente:

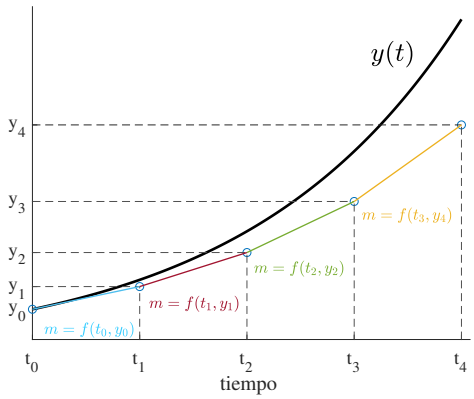

Figura 1: Ilustración del método de Euler. La curva $y(t)$ es la función solución, mientras que la cadena poligonal es la aproximación numérica. En cada segmento de la poligonal se indica la pendiente m de la recta asociada.

- **Paso 1. Introducimos los valores de entrada**: $f(t, y)$, y^0, t_0, T y Δt (en MatLab lo denotamos `dt`). Como T y dt no están definidos en el problema propuesto, les damos el valor que deseemos.

```
f=@(t,y) cos(t)*y;
y0=1;
t0=0;
T=20;
dt=0.1;
```

- **Paso 2. Definimos un vector t_n con la partición del intervalo** $[t_0, T]$:

```
tn=t0:dt:T;
N=(T-t0)/dt; % número de subintervalos de la partición
```

- **Paso 3. Definimos un vector y^n para almacenar la aproximación de la solución** $y(t)$:

```
yn=zeros(1,N+1); % vector de ceros con 1 fila y N+1 columnas
yn(1)=y0; % el valor inicial y0 se almacena en la primera componente
```

- **Paso 4. Calculamos los valores del vector y^n con el método de Euler**:

```
for k=1:N
   yn(k+1)=yn(k)+dt*f(tn(k), yn(k));
end
```

Una vez se ha calculado la solución aproximada, podemos utilizar diversas herramientas de postprocesado en Matlab para obtener información acerca de la misma. Por ejemplo:

– Realizar una representación gráfica: Con el comando `plot` podemos dibujar la solución aproximada obtenida. Para ello, basta con introducir la instrucción:

```
plot(tn,yn,'r') % El carácter 'r' denota que se dibuja la curva en color rojo
```

Además, en este caso conocemos la solución exacta al problema de valor inicial (5), $y(t) = e^{sen(t)}$. Dibujar ambas soluciones en una misma ventana gráfica nos puede dar una idea del error cometido en la aproximación. Para ello, utilizamos la primera forma de dibujar funciones anónimas presentada en la Sección 1.2 y añadimos las siguientes líneas de código:

```
hold on
plot(tn,exp(sin(tn)),'b') % El carácter 'b' denota que se dibuja la curva en color azul
hold off
```

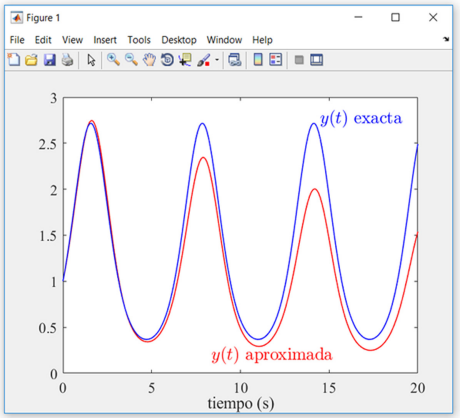

Figura 2: Representación gráfica de la solución del problema de valor inicial (5). La curva azul representa la solución exacta, mientras que la curva roja representa la solución aproximada obtenida con el método de Euler y un paso de tiempo $\Delta t = 0{,}1$.

Tras introducir estos comandos se obtiene la Figura 2. Observamos que la solución numérica aproxima bien la solución exacta para tiempos pequeños (por ejemplo, para $t < 5$) pero empeora para tiempos grandes.

– Calcular el valor de la solución en cierto instante de tiempo $t = t^*$: Obtenemos la posición k del vector t_n tal que $t_n(k) = t^*$, bien buscando directamente sobre el vector o bien utilizando el comando `find()` de MatLab. Una vez hallado el natural k, el valor deseado es $y^n(k)$.
En el ejemplo, si queremos obtener el valor de la solución aproximada en $t = 7{,}5$ (s), tecleamos:

```
yn(76) % valor aproximado. Buscamos la componente 76 porque tn(76)=7.5
```

Además, en este caso conocemos la solución exacta al problema de valor inicial (5), $y(t) = e^{sen(t)}$, y podemos comparar los valores aproximado y exacto para $t = 7{,}5$ (s).

```
exp(sin(7.5)) % valor exacto. Evaluamos la solución analítica en t=7.5
```

Para calcular la diferencia entre ambos valores (es decir, el error cometido por la aproximación en $t = 7{,}5$(s)), podemos utilizar el comando

```
dif= abs(exp(sin(7.5))-yn(76)) % diferencia en valor absoluto
```

con el cual se obtiene `dif = 0.3871`.

– Obtener el tiempo de cálculo: Con los comandos `tic; toc;` podemos saber el tiempo (en segundos) que tarda MatLab en calcular la solución aproximada. Para ello, basta con introducir el comando `tic;` antes del Paso 1 y el comando `toc;` después del Paso 4. MatLab muestra por pantalla un mensaje del tipo:

```
>> Elapsed time is 0.003771 seconds.
```

El error cometido al calcular la aproximación numérica a la solución puede deberse a una mala elección del paso de tiempo Δt. Sería conveniente que dedique ahora unos pocos minutos a analizar el impacto de este valor en la solución aproximada obtenida (en términos de tiempo de cálculo y de error cometido). Para ello, vuelva a ejecutar el código pero tomando ahora otros pasos de tiempo, por ejemplo $\Delta t = 0{,}01$ o $\Delta t = 0{,}05$.

Observación.- La elección de Δt se considera un arte, en el que se debe elegir un valor suficientemente pequeño para obtener una buena aproximación a la solución pero que, además, proporcione tiempos de computación razonables. Si no se dispone de la solución exacta - lo cual sucede habitualmente - se puede reducir recursivamente el valor de Δt hasta que dejen de observarse cambios en la gráfica de la solución aproximada.

2.2. Método de Heun

El método de Euler puede mejorarse si en la ecuación (2) afinamos un poco la aproximación de la integral

$$\int_{t_n}^{t_{n+1}} f(t, y(t))dt.$$

Una opción consiste en tomar, en lugar del rectángulo anterior con altura $f(t_n, y^n)$, el valor

$$\frac{1}{2}\left(f(t_n, y^n) + f(t_{n+1}, y^{n+1})\right),$$

que equivale a aproximar la integral por el área de un trapecio. Esto da lugar a

$$y(t_{n+1}) \approx y(t_n) + \frac{(t_{n+1} - t_n)}{2}\left(f(t_n, y^n) + f(t_{n+1}, y^{n+1})\right). \tag{6}$$

La incógnita y^{n+1} aparece como uno de los argumentos de f en el segundo término de (6) y a menudo resulta complicado despejarla de esta ecuación para obtener una fórmula explícita. Para superar esta dificultad, se sustituye y^{n+1} en el segundo miembro por el valor obtenido al aplicar la fórmula de Euler (3). Por tanto:

$$y(t_{n+1}) \approx y(t_n) + \frac{(t_{n+1} - t_n)}{2}\left(f(t_n, y^n) + f(t_{n+1}, y(t_n) + (t_{n+1} - t_n)f(t_n, y(t_n)))\right). \tag{7}$$

Si en la partición del intervalo $[t_0, T]$ se escogen puntos equiespaciados a distancia Δt, entonces el método de Heun viene dado por:

$$y^{n+1} = y^n + \frac{\Delta t}{2}(K_1 + K_2) \qquad n = 0, 1, 2, \ldots \tag{8}$$

donde

$$K_1 = f(t_n, y^n)$$
$$K_2 = f(t_n + \Delta t, y^n + \Delta t * K1)$$

Este método también se suele denominar método de Euler mejorado, ya que permite obtener una buena aproximación a la solución con un paso de tiempo Δt más grande que el empleado con el método de Euler, lo que resulta en un menor tiempo de computación.

2.2.1. Ejemplo en MatLab

En este apartado resolvemos el mismo problema de valor inicial que en la Sección 2.1.1, es decir el problema (5), pero esta vez utilizando el método de Heun. Para ello, en MatLab abrimos un nuevo fichero *script* y reescribimos el código MatLab de la Sección 2.1.1 modificando únicamente el Paso 4.

- **Paso 4. Calculamos los valores del vector y^n con el método de Heun**:

```
for k=1:N
   K1=f(tn(k), yn(k));
   K2=f(tn(k)+dt, yn(k)+dt*K1);
   yn(k+1)=yn(k)+0.5*dt*(K1+K2);
end
```

En la Figura 3 se ha dibujado la solución exacta al problema (5) (curva azul) junto con la aproximación numérica obtenida con el método de Heun utilizando un paso de tiempo $\Delta t = 0{,}1$ (curva roja). Resulta difícil diferenciar las curvas en la imagen; esto es debido a que la aproximación numérica se ajusta a la solución exacta con tanta precisión que sus representaciones gráficas se superponen. Tal y como hicimos en la Sección 2.1.1, hallamos la diferencia (en valor absoluto) entre la solución exacta y la solución aproximada en $t = 7{,}5$ (s), en este caso obteniendo `dif = 0.0050`. Este es otro indicador de que el método de Heun mejora el método de Euler.

Una vez estamos seguros de que la solución numérica aproxima adecuadamente a la solución exacta, podemos analizar la gráfica para obtener más conclusiones. Por ejemplo:

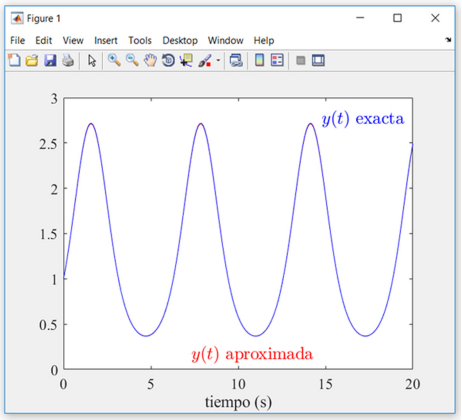

Figura 3: Representación gráfica de la solución del problema de valor inicial (5). La curva azul representa la solución exacta, mientras que la curva roja representa la solución aproximada obtenida con el método de Heun y un paso de tiempo $\Delta t = 0{,}1$.

– Estimar un instante de tiempo en el que la solución se aproxima a cierto valor $y = y^*$. Identificamos una posición k del vector y^n tal que $y^n(k) = y^*$ buscando directamente sobre el vector. Una vez hallado el número natural k, el instante de tiempo buscado es $t_n(k)$.

En el problema (5), podríamos estar interesados en estimar el instante de tiempo más pequeño para el que la solución aproximada vuelve a tomar el valor $y = 1$ (nótese que la condición inicial era $y(0) = 1$).

Observación.- Dado que, en este caso, conocemos la solución exacta al problema de valor inicial (5), $y(t) = e^{sen(t)}$, podemos calcular analíticamente el instante de tiempo buscado. Para ello, calculamos valores de t para los que $y(t) = 1$:

$$e^{sen(t)} = 1 \Leftrightarrow t = \arcsin\left(\ln(1)\right) = \text{arcsen}(0) = 0 + m\pi \quad m = 0, 1, 2, \ldots.$$

Se concluye fácilmente que el primer instante de tiempo (sin contar $t = 0$) para el que se cumple la condición $y(t) = 1$ es $t = \pi$ (s).

A continuación, presentamos una alternativa para estimar el instante de tiempo deseado en aquellos casos en los que no se conozca la solución exacta del problema.

1. Dibujamos la solución aproximada junto con la función constante $z(t) = 1$ (ver Figura 4).

   ```
   plot(tn,yn);
   hold on
   plot(tn,ones(1,N+1)); % vector de unos con 1 fila y N+1 columnas
   hold off
   ```

   Observamos que en este caso hay 7 puntos de corte, es decir, 7 instantes en los que la solución aproximada toma el valor 1. De entre ellos nos interesa el segundo, que parece encontrarse en el intervalo $t \in [2, 4]$.

2. Buscamos el punto de corte en el vector `yn`.

   ```
   yn=
   Columns 31 through 40
   1.1511   1.0423   0.9434   0.8544   0.7750   0.7047   0.6431   0.5894   0.5431   0.5035
   ```

   A parte de `yn(1)=1`, el vector `yn` no contiene otras componentes que valgan exactamente 1, pero podemos identificar dónde se encuentra el punto de corte. Para ello, buscamos dos componentes consecutivas donde el valor de `yn` pase de ser mayor a menor que 1 (fíjese en las componentes encuadras en `yn`). Una posible elección de k consiste en tomar la posición del valor de `yn` más cercano a 1 (de entre los dos valores encuadrados), en este caso `1.0423`, por lo que tomamos $k = 32$.

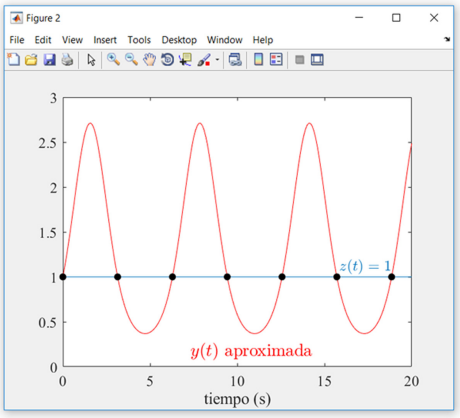

Figura 4: Representación gráfica de la solución del problema (5) con el método de Heun y $\Delta t = 0{,}1$ (curva roja) y de la función constante $z(t) = 1$ (recta azul). Los puntos negros representan los cortes entre ambas gráficas.

3. Obtenemos el instante de tiempo asociado al punto de corte.
   El instante de tiempo buscado es `tn(32)=3.1` (s).

3. Problema práctico de resolución de problemas de valor inicial

Se propone a continuación un problema cuya resolución se puede abordar utilizando una combinación de técnicas analíticas para la resolución de ecuaciones diferenciales ordinarias (vistas en las sesiones de teoría) y los métodos numéricos de Euler y Heun. En Moodle se ha habilitado un cuestionario denominado *Cuestionario Guión Práctica 1* que incluye las preguntas planteadas a lo largo del problema. Debe tener en cuenta que estas preguntas son similares a las que se propondrán durante la práctica, la cual contará con un cuestionario similar y se trabajará en el aula de ordenadores. Conviene, por tanto, familiarizarse con el cuestionario y con el uso de los algoritmos propuestos en MatLab para resolverlo.

Las preguntas que se identifican con la letra **Q** se contestarán en el cuestionario de Moodle y las marcadas con **T** son preguntas teóricas que, durante la práctica, deberán contestarse a mano. En este guión, se han proporcionado datos para chequear si las soluciones obtenidas analíticamente son correctas.

Problema práctico

Un tanque se drena a través de un tubo situado en el fondo, tal y como se muestra en la Figura 5.

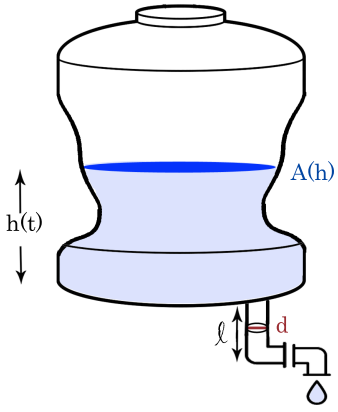

Figura 5: Representación gráfica del tanque considerado en el problema práctico.

Tras ciertas hipótesis, se obtiene el siguiente problema de valor inicial que describe cómo varía la altura del

líquido en el tanque con el tiempo:

$$\begin{cases} h'(t) = -\dfrac{\pi d^2}{4A(h(t))}\sqrt{2g(h(t)+\ell)} \\ h(0) = 6 \end{cases}$$

donde $h(t)$ es la altura del líquido en el tanque (m) en el instante de tiempo t (s), $A(h)$ el área de la superficie del líquido (m^2) a una altura h del fondo, g la constante gravitacional ($g = 9,81$ m/s^2), d el diámetro del tubo (m) y ℓ la profundidad de la salida del tubo por debajo del fondo del tanque (m). Se sabe que el diámetro del tubo es de $d = 0,25$ m.

Se consideran dos hipótesis, (H1) y (H2). Los Ejercicios 1.1 y 1.2 se resuelven en las hipótesis (H1) y el Ejercicio 2 se resuelve en la hipótesis (H2).

(H1) El tanque tiene forma de prisma y el área de la superficie es $A(h) = 2000h$ m^2. Además se supone la idealización de que $\ell = 0$.

Ejercicio 1.1. En la hipótesis (H1), se pide:

T.1.1.(a) Escribir el problema de valor inicial que describe la evolución de la altura del líquido en el tanque y resolverlo analíticamente. Para chequear la solución se puede utilizar que, a las 10 horas, la altura es aproximadamente $4,2709$ m.

Q.1.1.(b) Dibujar la solución obtenida en las primeras 20 horas.

Q.1.1.(c) Calcular la altura del tanque tras 15 horas.

Q.1.1.(d) Estimar en cuántas horas se vacía el tanque.

Ejercicio 1.2. Resolver el problema **T**.1.1.(a) numéricamente utilizando el método de Euler con un paso de tiempo de 2 horas. Se pide:

Q.1.2.(a) Dibujar la solución obtenida en las primeras 20 horas y compararla con la gráfica de **Q**.1.1.(b).

Q.1.2.(b) Hallar el valor absoluto de la diferencia entre las soluciones numérica y exacta en el instante $t = 20$ h.

Q.1.2.(c) Resolver el problema **T**.1.1.(a) con el método de Euler utilizando ahora un paso de tiempo de 2 minutos y observar la diferencia.

(H2) Tras un estudio de la topografía del terreno se ha podido realizar una estimación más precisa de la geometría del tanque y se observa que la función $A(h) = 100(h^3 - 6h^2 + 25h)$ aproxima con el grado de precisión necesario la superficie libre del depósito para los distintos valores de h. Además, se sabe que $\ell = 1$ m.

Ejercicio 2. En la hipótesis (H2), se pide:

T.2.(a) Escribir el problema de valor inicial que describe la evolución de la altura del líquido en el tanque.

Resolverlo numéricamente con el método de Heun utilizando un paso de 2 minutos. Para chequear la solución se puede utilizar que, a las 10 horas, la altura es aproximadamente $4,1670$ m.

Q.2.(b) Dibujar la solución obtenida en las primeras 17 horas.

Q.2.(c) Calcular la altura del líquido en el tanque tras 15 horas.

Q.2.(d) Estimar en cuántas horas se vacía el tanque.