

Libro: Quarteroni, "Scientific Computing with MATLAB and Octave"

### PROBLEMA 8.1 – TERMODINÁMICA (pag. 271)

Considere un cuerpo que tiene una temperatura interna  $T$  colocado en un ambiente cuya temperatura  $T_e$  es constante. Entonces la 1ra ley de termodinámica y la ley de Stefan-Boltzmann establecen que

$$\frac{dT}{dt} = -k(T^4 - T_e^4), \quad (1)$$

donde:  $k = \frac{\epsilon \gamma S}{m C}$ ,  $\epsilon = 5.6 \times 10^{-8} \text{ J}/(\text{m}^2 \text{K}^4 \text{s})$  (constante de Stefan-Boltzmann),  $\gamma$  es la constante de emisividad del cuerpo,  $S$  es el área de su superficie,  $m$  es su masa y  $C$  es su calor específico.

Para este problema, se plantean los siguientes objetivos:

1. Si es posible, encontrar analíticamente la temperatura  $T$  del cuerpo para un tiempo  $t$ .
2. Utilizar el método del Trapecio y el método de Euler mejorado para encontrar valores aproximados a la temperatura exacta  $T$  o a una temperatura de referencia (en caso no se consiga la temperatura exacta  $T$ ).
3. Verificar la convergencia de ambos métodos numéricos.

#### 1. SOLUCIÓN ANALÍTICA

La ecuación diferencial Ec. (1) no es difícil de ser resuelta analíticamente; sin embargo, la temperatura  $T$  del cuerpo para un tiempo  $t$  quedará definida implícitamente, como se ve a continuación. Si la temperatura del cuerpo en el instante  $t = 0$  es  $T_0$ , entonces, al resolver la Ec. (1), se tiene una expresión en la que el tiempo  $t$  queda en función de la temperatura  $T$

$$t = \frac{1}{4kT_e^3} \left[ \ln \left( \frac{T + T_e}{T_0 + T_e} \right) - \ln \left( \frac{T - T_e}{T_0 - T_e} \right) + 2 \arctan \left( \frac{T}{T_e} \right) - 2 \arctan \left( \frac{T_0}{T_e} \right) \right]. \quad (2)$$

En lo que sigue de este problema, se considera un cuerpo específico: un cubo cuyas aristas miden 1m, la masa es de 1 kg, el calor específico  $C$  del material del cubo es de  $100 \text{ J}/(\text{kg/K})$  y la emisividad  $\gamma$  del cubo es de 0.5. El cubo reposa en un ambiente cuya temperatura constante  $T_e$  es de 200 K y la temperatura inicial del cubo  $T_0$  es de 180 K.

Evidentemente, este sistema estará en equilibrio cuando la temperatura  $T$  del cuerpo alcance la temperatura ambiente (200 K); pero, la Ec. (2) indica que eso ocurrirá en un tiempo  $t = \infty$ . La Tabla 1 da los valores del tiempo  $t$  que corresponden a ciertos valores de temperatura  $T$  del cuerpo. Se observa que la temperatura  $T$  aumenta de 180 K a 199 K en poco más de 58 segundos. Además, como se esperaba, esta temperatura tiende al valor de la temperatura ambiente (200 K); de hecho, se puede decir que a los 58 segundos el cuerpo ya está a temperatura ambiente porque a partir de ese instante, la temperatura  $T$  sólo crece fracciones de Kelvin aproximándose lentamente a 200 K. Vale indicar que la columna "t(s)" de los tiempos (Tabla 1) se calcula a partir de los valores de temperatura de la columna "T(K)" y la Ec. (2).

En la siguiente Sección, se necesitan los valores exactos de la temperatura  $T$  del cuerpo para determinados instantes de tiempo  $t$ . Usar la Ec. (2) para encontrar la temperatura  $T$  a partir de un tiempo  $t$  dado requiere resolver esa ecuación. Con el fin de evitar este camino, se calcula, para un tiempo  $t$  dado, una temperatura suficientemente próxima al valor exacto  $T$  de modo que sea útil en la siguiente Sección. En el siguiente párrafo, se define el método que servirá para calcular estas temperaturas de referencia.

T (K)	t(s)
180	0
190	14,3790545599281
199	58,4947180080381
199,9	101,451388167438
199,99	144,294770751300
199,999	187,126850262745
199,9999	229,957799722959
199,99999	272,788636175897
199,999999	315,619461386764
199,9999999	358,450286472097

Tabla 1. Tiempo  $t$  que corresponde a la temperatura  $T$ , según la Ec. (2), del cuerpo considerado en este problema.

T (K)	t(s)	T_ode (K)	Error relativo
180	0	180,000000000000	0
190	14,3790545599281	189,999999800854	1,0E-09
199	58,4947180080381	198,999999537784	2,3E-09
199,9	101,451388167438	199,899999756263	1,2E-09
199,99	144,294770751300	199,990001680166	8,4E-09
199,999	187,126850262745	199,999000506888	2,5E-09
199,9999	229,957799722959	199,999899405927	3,0E-09
199,99999	272,788636175897	199,999990702097	3,5E-09
199,999999	315,619461386764	199,999998857485	7,1E-10
199,9999999	358,450286472097	199,99999923535	1,2E-10

Tabla 2. Temperatura  $T_{ode}$  que corresponde al tiempo  $t$ , calculada con la función `ode113` del MATLAB. El error relativo entre  $T_{ode}$  y  $T$  (temperatura exacta) indica que  $T_{ode}$  tiene entre 8 y 9 cifras exactas.

La Tabla 2 muestra las temperaturas  $T_{ode}$  calculadas con la función `ode113` del MATLAB aplicada a los tiempos  $t$  de la columna “t(s)” (es la misma columna de la Tabla 1). Se observa que, para un tiempo  $t$  dado, la temperatura  $T_{ode}$  del cuerpo es aproximadamente igual a la temperatura exacta  $T$  del cuerpo. El error relativo entre  $T_{ode}$  y  $T$  (columna “Error relativo”) indica que  $T_{ode}$  tiene entre 8 y 9 cifras exactas; esta precisión se mantiene en cada instante de tiempo  $t$  de la Tabla 2. El desempeño de la función `ode113` del MATLAB (verificado en esta Sección) y sus características (descritas en el help del MATLAB) justifican su uso, en la siguiente Sección, para calcular las temperaturas de referencia que se piden.

Vale indicar que la función `ode113` del MATLAB resuelve numéricamente ecuaciones diferenciales ordinarias y está basado en un esquema multipaso Adams-Bashforth-Moulton; además, como otras funciones `ode` del MATLAB, permite definir una tolerancia al error relativo y, por lo tanto, el tamaño de paso es variable. A continuación, se muestra un extracto del help del MATLAB:

```
>> help ode113
ode113 Solve non-stiff differential equations, variable order method.
[TOUT,YOUT] = ode113(ODEFUN,TSPAN,Y0) with TSPAN = [T0 TFINAL] integrates
the system of differential equations  $y' = f(t,y)$  from time T0 to TFINAL
with initial conditions Y0. ODEFUN is a function handle. For a scalar T
and a vector Y, ODEFUN(T,Y) must return a column vector corresponding
to  $f(t,y)$ . Each row in the solution array YOUT corresponds to a time
returned in the column vector TOUT. To obtain solutions at specific
```

times T0,T1,...,TFINAL (all increasing or all decreasing), use TSPAN = [T0 T1 ... TFINAL].

[TOUT,YOUT] = ode113(ODEFUN,TSPAN,Y0,OPTIONS) solves as above with default integration properties replaced by values in OPTIONS, an argument created with the ODESET function. See ODESET for details. Commonly used options are scalar relative error tolerance 'RelTol' (1e-3 by default) and vector of absolute error tolerances 'AbsTol' (all components 1e-6 by default). If certain components of the solution must be non-negative, use ODESET to set the 'NonNegative' property to the indices of these components.

Finalmente, se presentan los comandos MATLAB usados para calcular las temperaturas  $T_{ode}$  de la Tabla 2. Obsérvese que uno de los datos de entrada de la función ode113 es el vector  $t$  que contiene los tiempos de la columna “t(s)” de las Tablas 1 y 2. Además, como lo indica el help, la función ode113 proporciona el vector solución Tode cuyas componentes corresponden a cada tiempo del vector  $t$ . Es decir, este dato de salida constituye la columna “T\_ode (K)” de la Tabla 2.

```
>> t % VECTOR DE TIEMPOS t DE LA TABLA 1
t =
    0
 1.437905455992810e+01
 5.849471800803812e+01
 1.014513881674383e+02
 1.442947707513002e+02
 1.871268502627448e+02
 2.299577997229586e+02
 2.727886361758972e+02
 3.156194613867644e+02
 3.584502864720967e+02
>> T0=180;
>> options=odeset('RelTol',1.e-10); % SE DEFINE LA TOLERANCIA AL
% ERROR RELATIVO COMO 10^(-10)
>> [tode,Tode]=ode113(@ftermo,t,T0,options);
```

donde ftermo es la función asociada al lado derecho de la Ec. (1):

```
function [f] = ftermo (t,T)
    ee=5.6*10^(-8);
    gamma=0.5;
    S=6;
    m=1;
    C=100;
    k=ee*gamma*S/(m*C);
    Te=200;
    f=-k*(T^4-Te^4);
end
```

NOTA: el lector puede verificar que las otras funciones ode del MATLAB (ode23, ode45, ode23s,...) también pueden tener un desempeño similar al del ode113.

## 2. SOLUCIÓN NUMÉRICA

Así como en la Sección anterior, aquí se considera el problema del cubo que reposa en un ambiente cuya temperatura constante  $T_e$  es de 200 K; la temperatura inicial  $T_0$  del cubo es de 180 K y, para un tiempo  $t$  arbitrario, la temperatura  $T$  del cubo obedece la ecuación diferencial Ec. (1). Sin embargo, a diferencia de la Sección anterior, aquí se buscan soluciones numéricas a este problema; en ese sentido, son considerados dos métodos: el método del Trapecio y el método de Euler mejorado. Además, las soluciones numéricas son comparadas con la solución de referencia (calculada mediante la función ode113 del MATLAB) con el fin de verificar la convergencia de los métodos numéricos.

### a. Método del Trapecio:

Es un método implícito, de un paso, y tiene orden 2 de convergencia. La Figura 1 muestra una solución  $T_{tra}$  calculada con este método para un tamaño de paso  $h = 30$  segundos, y otra solución  $T_{tra}$  para un  $h = 15$  segundos. La línea continua es la solución de referencia  $T_{ode}$  calculada con la función ode113 del MATLAB cuya tolerancia al error relativo es definida como  $10^{-10}$  (ver Sección anterior). Se observa que en  $t = 30$  segundos, la solución  $T_{tra}$  para  $h = 15$  es más precisa que la solución  $T_{tra}$  para  $h = 30$ . Dado que el método del Trapecio tiene orden 2 de convergencia, se espera que el error disminuya a la cuarta parte cuando  $h$  disminuye a la mitad. De hecho, esto parece estar ocurriendo en  $t = 30$  y en  $t = 60$  segundos.

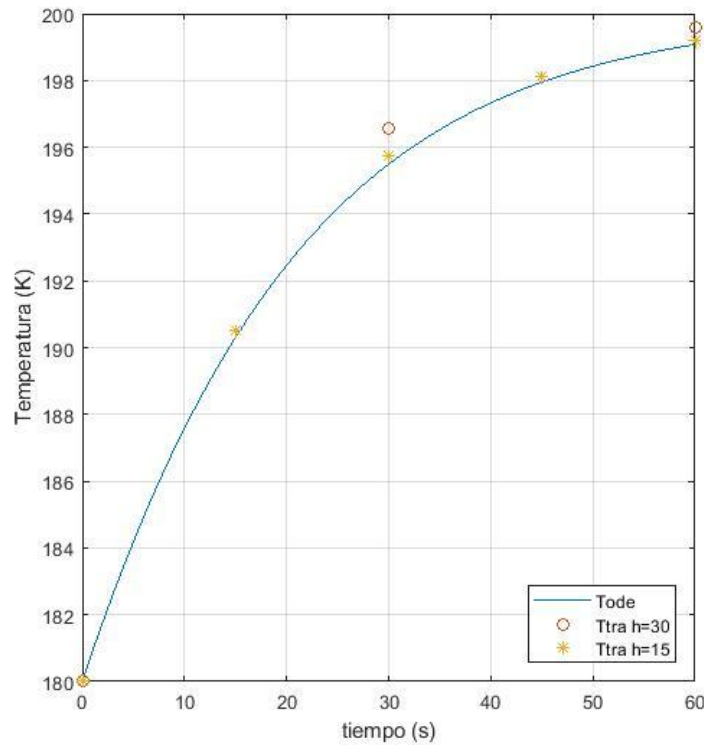


Figura 1. Solución  $T_{tra}$  calculada con el método del Trapecio para un tamaño de paso  $h = 30$  y otra solución  $T_{tra}$  para  $h = 15$ .

Para continuar verificando la convergencia del método, se realizan diez simulaciones de este problema. En cada simulación se consideran  $N + 1$  puntos del intervalo cerrado  $[0, 120]$ , es decir  $0 = t_0, t_1, t_2, \dots, t_N = 120$ , igualmente espaciados por un tamaño de paso  $h = 120/N$ . La Tabla 3 muestra los resultados encontrados para cada simulación. El error máximo se calcula de la siguiente manera

$$\text{Error máximo} = \max_{n=0, \dots, N} |(T_{tra})_n - (T_{ode})_n|$$

donde  $(T_{tra})_n$  y  $(T_{ode})_n$  son, respectivamente, la solución con el método del Trapecio y la solución de referencia en el punto  $t_n$ . Por ejemplo, en la simulación con  $N = 4$  (o  $h = 30$ ) se tiene un error máximo de 1,07 (grados Kelvin); observe que este error aparece en la Figura 1. El método del Trapecio sí resulta ser convergente pues el error máximo converge a cero conforme  $N$  aumenta (o  $h$  disminuye). La columna “Orden  $p$ ” de la Tabla 3 registra el orden de convergencia efectivo  $p$  de este método; éste se calcula a partir de la siguiente observación

$$\text{Error máximo} \propto h^p$$

(ver Süli, “An Introduction to Numerical Analysis”, Teorema de Equivalencia de Dahlquist para un método consistente con orden  $p$  de precisión). Los resultados confirman que el método del Trapecio es de orden 2 de convergencia. En la siguiente Sección, se comenta nuevamente este tema.

N	h	Error máximo	Orden p
4	30	1,07E+00	-
8	15	2,38E-01	2,17
16	7,5	5,77E-02	2,04
32	3,75	1,44E-02	2,00
64	1,875	3,60E-03	2,00
128	0,9375	9,02E-04	2,00
256	0,46875	2,26E-04	2,00
512	0,234375	5,69E-05	1,99
1024	0,1171875	1,46E-05	1,96
2048	0,05859375	4,13E-06	1,83

Tabla 3. Error máximo encontrado en cada simulación, y orden de convergencia efectivo del método del Trapecio.

Para concluir el análisis del método del Trapecio, se presentan los comandos MATLAB usados para calcular los resultados mostrados en la Tabla 3. Observe que la función `cranknic` es la implementación del método del Trapecio (ver Quarteroni, programa 8.3).

```

N=4;
tspan=[0,120];
T0=180;
options=odeset('RelTol',1.e-10);
e=zeros(10,1); % VECTOR PARA
                % ERRORES MÁXIMOS

for k=1:10
    [ttra,Ttra]=cranknic(@ftermo,tspan,T0,N); % SOLUCIÓN NUMÉRICA
    [tode,Tode]=ode113(@ftermo,ttra,T0,options); % SOLUCIÓN DE REFERENCIA
    e(k)= max(abs(Ttra-Tode)); % ERROR MÁXIMO
                                % DE LA SIMULACIÓN k

    N=2*N;
end
p=log(abs(e(1:end-1)./e(2:end)))/log(2); % ORDEN p DE CONVERGENCIA

```

#### b. Método de Euler mejorado (Improved Euler):

Es un método Runge-Kutta explícito, de un paso y tiene orden 2 de convergencia. La Figura 2 muestra una solución  $T_{ieu}$  calculada con este método para un tamaño de paso  $h = 30$  segundos, y otra solución  $T_{ieu}$  para un  $h = 15$  segundos. Evidentemente, el error disminuye cuando  $h$  disminuye, sin embargo el error es mayor a aquel encontrado con el método del Trapecio: compare las soluciones  $T_{ieu}$  y  $T_{tra}$  para  $h = 30$ , por ejemplo.

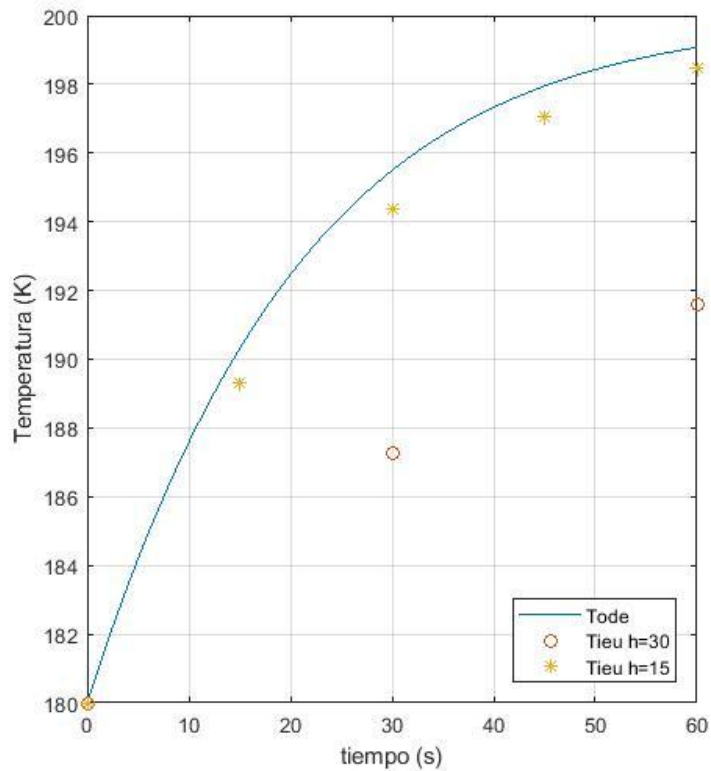


Figura 2. Solución  $T_{ieu}$  calculada con el método Improved Euler para un tamaño de paso  $h = 30$  y otra solución  $T_{ieu}$  para  $h = 15$ .

De manera análoga al método de Trapecio, se realizan diez simulaciones con el método Improved Euler. La Tabla 4 muestra los resultados encontrados para cada simulación. El error máximo se calcula de la siguiente manera

$$\text{Error máximo} = \max_{n=0,\dots,N} |(T_{ieu})_n - (T_{ode})_n|.$$

Por ejemplo, en la simulación con  $N = 4$  (o  $h = 30$ ) se tiene un error máximo de 8,23 (grados Kelvin); note que este error aparece en la Figura 2. El método Improved Euler resulta ser convergente pues el error máximo converge a cero conforme  $N$  aumenta (o  $h$  disminuye); y la convergencia se da con orden 2 (ver columna "Orden p" de la Tabla 4).

N	h	Error máximo	Orden p
4	30	8,23E+00	-
8	15	1,14E+00	2,86
16	7,5	2,11E-01	2,43
32	3,75	4,54E-02	2,21
64	1,875	1,06E-02	2,10
128	0,9375	2,55E-03	2,05
256	0,46875	6,26E-04	2,03
512	0,234375	1,55E-04	2,01
1024	0,1171875	3,83E-05	2,02
2048	0,05859375	9,35E-06	2,03

Tabla 4. Error máximo encontrado en cada simulación, y orden de convergencia efectivo del método Improved Euler.

Para concluir el análisis del método Improved Euler, se presentan los comandos MATLAB usados para calcular los resultados mostrados en la Tabla 4. Observe que la función rk2 implementa el método Improved Euler (ver Quarteroni, programa 10.2).

```

N=4;
tspan=[0,120];
T0=180;
options=odeset('RelTol',1.e-10);
e=zeros(10,1);

for k=1:10
    [tieu,Tieu]=rk2(@ftermo,tspan,T0,N);
    [tode,Tode]=ode113(@ftermo,tieu,T0,options);
    e(k)=max(abs(Tieu-Tode));

    N=2*N;
end
p=log(abs(e(1:end-1)./e(2:end)))/log(2);

```

% VECTOR PARA  
 % ERRORES MÁXIMOS  
  
 % SOLUCIÓN NUMÉRICA  
 % SOLUCIÓN DE REFERENCIA  
 % ERROR MÁXIMO  
 % DE LA SIMULACIÓN k  
  
  
 % ORDEN p DE CONVERGENCIA

### 3. OBSERVACIONES FINALES

Este problema considera un cuerpo que es colocado en un ambiente cuya temperatura es constante (200 K). El cuerpo tiene una temperatura inicial de 180 K y para un tiempo  $t$  arbitrario, su temperatura  $T$  es modelada por la ecuación diferencial Ec. (1). La temperatura del cuerpo aumenta a 199 K en poco más de 58 segundos y se aproxima lentamente a la temperatura ambiente (200 K).

El comportamiento de la temperatura a lo largo del tiempo es simulado por dos métodos numéricos: el método del Trapecio y el método Improved Euler (ver Figuras 1 y 2). Ambos métodos dan resultados satisfactorios: las soluciones numéricas son próximas a la solución exacta  $T$  y convergen a ésta conforme el tamaño de paso  $h$  disminuye.

Los errores encontrados por los dos métodos numéricos al resolver la Ec. (1) en el intervalo  $[0, 120]$  se muestran en la Figura 3. Esta figura confirma que existe convergencia a la solución exacta  $T$  pues el error converge a cero cuando  $N$  aumenta ( $N$  es el número de puntos en el intervalo  $[0, 120]$ ).

Sin embargo, a pesar de que ambos métodos presentan el mismo orden de convergencia (orden 2), el método del Trapecio es más eficiente en el siguiente sentido: la solución con este método resulta ser más precisa que la solución proporcionada por el método Improved Euler, cuando ambos métodos usan la misma malla (el mismo número  $N$ ). Por ejemplo, la Figura 3 indica que, para  $N = 128$ , el método del Trapecio comete un error aproximado de  $10^{-3}$  (grados Kelvin) mientras que el método Improved Euler da un error mayor a  $2 \times 10^{-3}$  (grados Kelvin).

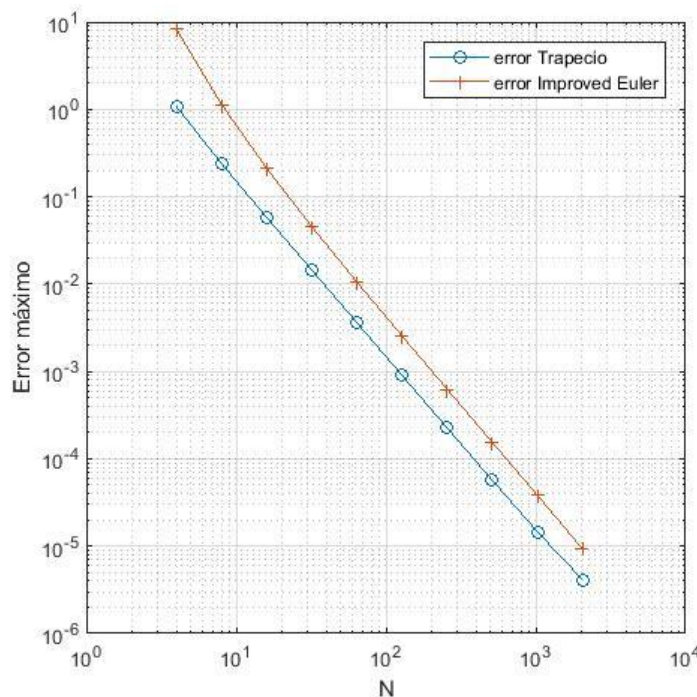


Figura 3. Errores encontrados por los dos métodos numéricos al resolver la Ec. (1) en el intervalo  $[0, 120]$ . El número  $N$  es el número de puntos en el intervalo  $[0, 120]$ .

Finalmente, otra diferencia entre ambos métodos es la región de estabilidad. Las simulaciones vistas hasta aquí fueron realizadas con un tamaño de paso  $h \leq 30$ . Ahora, se considera un tamaño de paso mayor ( $h = 40$ ). Los resultados se muestran en la Figura 4: los valores de temperatura  $T_{tra}$  calculados con el método del Trapecio respetan el comportamiento de la



solución exacta (la temperatura de referencia  $T_{ode}$  se considera como solución exacta); esto no sucede con los valores de temperatura  $T_{ieu}$  obtenidos con el método Improved Euler pues estos valores decrecen y lo hacen desde el instante inicial, cuando la temperatura del cuerpo es de 180 K. Así, con un tamaño de paso  $h$  igual a 40, el método explícito Improved Euler resulta inadecuado para resolver la Ec. (1), en cuanto que el método implícito del Trapecio sí consigue dar resultados confiables.

La ventaja de los métodos explícitos, en relación a los métodos implícitos, es la simplicidad en su algoritmo y por tanto, el bajo costo computacional en cada paso del método. La desventaja es que usualmente necesitan más pasos ( $h$  pequeño) que los utilizados por los métodos implícitos para alcanzar una precisión deseada.

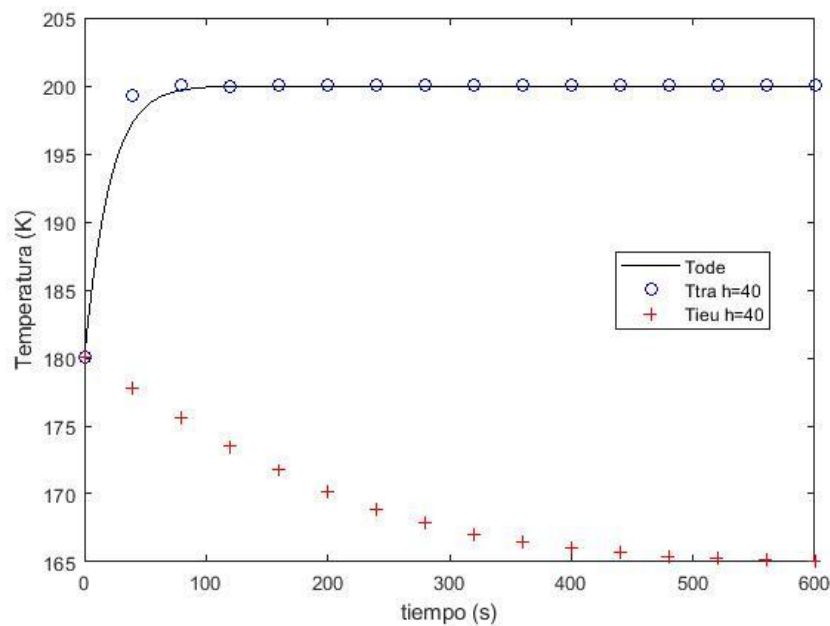


Figura 4

NOTA: el lector puede realizar simulaciones para otros valores de  $h$  (mayores o menores) y observar lo que sucede. También puede verificar la convergencia de otros métodos (Euler implícito, Adams-Moulton, ...) en la resolución de este problema.