MÉTODOS DE SIMULACIÓN Y MODELADO

Solución al Trabajo Práctico - Enero de 2020

EJERCICIO 1

Lea el artículo citado a continuación, que puede descargar de la página web de la asignatura, y conteste a las preguntas.

> Åström, K.J., Elmqvist, H., Mattsson, S.E. *Evolution of continuous-time modeling and simulation*. The 12^{th} European Simulation Multiconference, ESM'98, June 16–19, 1998, Manchester, UK.

1. ¿Qué analogías pueden establecerse entre el modelado basado en diagramas de bloques y el paradigma de la simulación analógica?
2. ¿Qué es el paradigma de modelado físico? ¿Qué tipo de modelos matemáticos se obtienen de aplicar el paradigma del modelado físico?
3. ¿Qué diferencias hay entre el paradigma de la simulación analógica y el paradigma del modelado físico?
4. Observe la Figura 1.1 y comente su contenido basándose en el artículo.

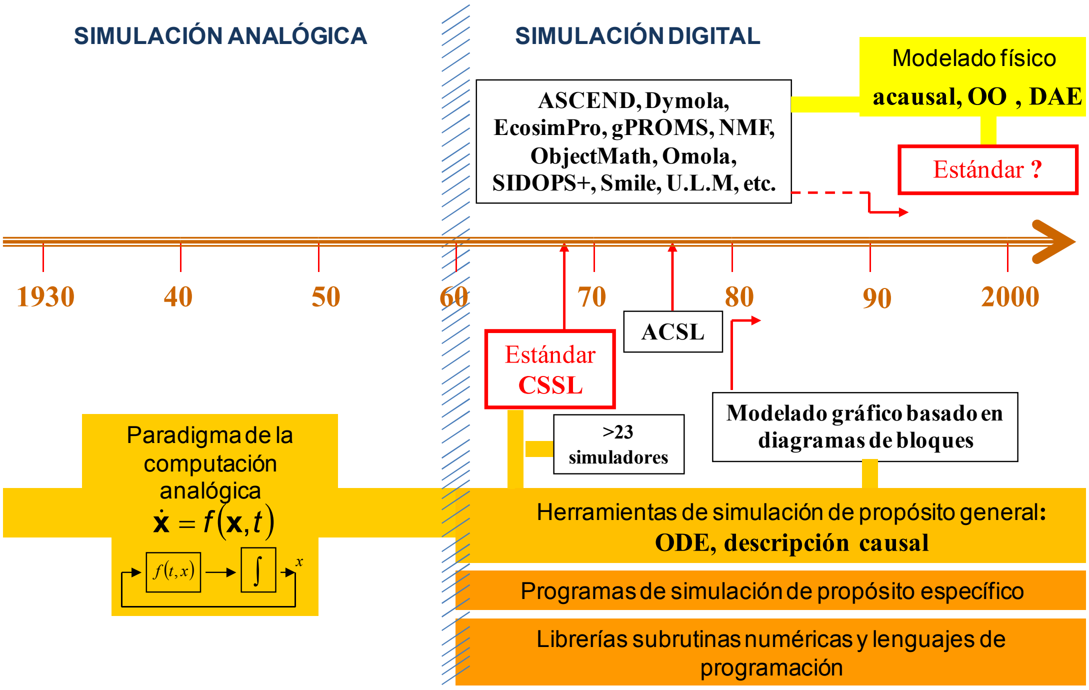

Figura 1.1: Evolución del modelado y simulación de tiempo continuo.

Solución al Ejercicio 1

En este ejercicio el alumno debe contestar, con sus propias palabras, a las cuestiones planteadas.

EJERCICIO 2

Se ha modelado matemáticamente el comportamiento térmico de una vivienda que dispone de un sistema de climatización regulado por un controlador PI. En la figura siguiente se muestra un diagrama del sistema.

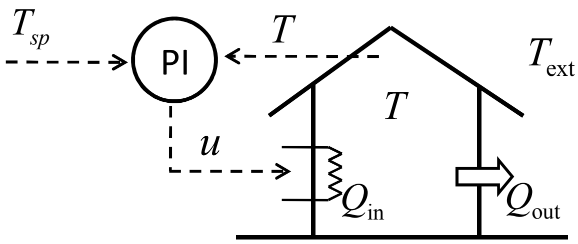

Figura 1.2: Vivienda con sistema de climatización.

La temperatura a la que se encuentra el interior de la vivienda es T y la temperatura en su exterior T_{ext}. Ambas están expresadas en Kelvin. El sistema de climatización suministra a la vivienda una potencia calorífica Q_{in}. La potencia calorífica intercambiada entre la vivienda y su entorno, debido a la transferencia de calor, es Q_{out}.

El controlador PI, representado en la figura mediante una circunferencia con la etiqueta PI, tiene como entradas el valor actual de la temperatura en el interior de la vivienda (T) y el valor deseado de la misma (T_{sp}, denominado valor de consigna de la temperatura). El controlador PI tiene como salida el voltaje u, que es la señal de entrada al sistema de climatización. Las ecuaciones mostradas a continuación describen el comportamiento térmico de la casa y el comportamiento del controlador. La variable t representa el tiempo en segundos.

$$
\begin{aligned}
C \cdot \frac{dT}{dt} &= Q_{\text{in}} - Q_{\text{out}} \\
Q_{\text{out}} &= k \cdot (T - T_{\text{ext}}) \\
T_{\text{ext}} &= 275 + 10 \cdot \sin\left(\frac{2 \cdot \pi \cdot t}{3600 \cdot 24}\right) \\
e &= T_{sp} - T \\
\frac{dI}{dt} &= e \\
u &= k_p \cdot e + \frac{1}{k_I} \cdot I \\
Q_{\text{in}} &= K_c \cdot u
\end{aligned}
$$

La capacidad calorífica de la casa vale $C = 1E7$ J/K. La constante de proporcionalidad $k = 2E3$ W/K representa la conductividad térmica de las paredes y ventanas. Los parámetros del controlador PI valen $k_p = 7$ V/K, $k_I = 2E2$ K·s/V. La constante de proporcionalidad del sistema climatizador vale $K_c = 4E3$ W/V.

El valor de consigna para la temperatura (T_{sp}) va alternando cada 6 horas entre los valores 293 K y 300 K. En la figura se muestra la evolución de T_{sp} durante el primer día. El comportamiento es el mismo (alternante entre 293 K y 300 K) los siguientes días.

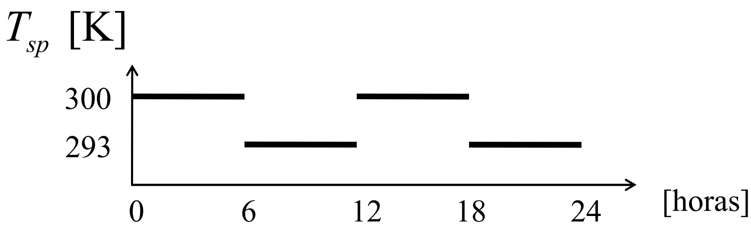

Figura 1.3: Evolución del valor de consigna para la temperatura.

1. Escriba las ecuaciones del modelo completo del sistema.
2. Asigne la causalidad computacional. Indique cuántos grados de libertad tiene el modelo.
3. Escriba el diagrama de flujo del algoritmo para la simulación de este modelo. Emplee el método de integración de Euler explícito. La condición de finalización de la simulación es que el tiempo simulado alcance el valor 86400 s.
4. Programe el algoritmo anterior en lenguaje R y ejecute la simulación. Represente gráficamente la evolución frente al tiempo de las tres siguientes variables: la temperatura a la que se encuentra el interior de la vivienda (T), el valor de consigna para dicha temperatura (T_{sp}) y la temperatura exterior (T_{ext}). Explique qué criterio ha seguido para escoger el tamaño del paso de integración.

Solución al Ejercicio 2

Las ecuaciones del modelo se muestran a continuación. No se muestran las asignaciones de valor a las constantes y parámetros.

$$
\begin{aligned}
C \cdot \frac{dT}{dt} &= Q_{\text{in}} - Q_{\text{out}} && (1.1)\\
Q_{\text{out}} &= k \cdot (T - T_{\text{ext}}) && (1.2)\\
T_{\text{ext}} &= 275 + 10 \cdot \sin\left(\frac{2 \cdot \pi \cdot t}{3600 \cdot 24}\right) && (1.3)\\
e &= T_{sp} - T && (1.4)\\
\frac{dI}{dt} &= e && (1.5)\\
u &= k_p \cdot e + \frac{1}{k_I} \cdot I && (1.6)\\
Q_{\text{in}} &= K_c \cdot u && (1.7)\\
T_{sp} &= f(t) && (1.8)
\end{aligned}
$$

Obsérvese que la variación en el tiempo de la variable T_{sp} mostrada en la Figura 1.3 se ha representado mediante una función f del tiempo. Una posible forma de definir dicha función es:

$$
T_{sp} = \begin{cases} 300 & \text{si } t \,\%\, (12 \cdot 3600) < 6 \cdot 3600 \\ 293 & \text{en caso contrario} \end{cases} \qquad (1.9)
$$

donde $\%$ es el operador resto de la división entera.

Asumiendo que las dos variables que aparecen derivadas pueden ser seleccionadas como variables de estado, las variables del modelo pueden clasificarse de la forma siguiente:

– Parámetros y constantes: C, k, k_p, k_I, K_c
– Variables de estado: T, I
– Variables algebraicas: Q_{in}, Q_{out}, T_{ext}, e, T_{sp}, u

Para asignar la causalidad computacional al modelo, se sustituyen las derivadas de las variables de estado por variables mudas:

$$
\tfrac{dT}{dt} \rightarrow derT \qquad\qquad \tfrac{dI}{dt} \rightarrow derI
$$

Omitiendo las ecuaciones en las que se asigna valor a los parámetros y las constantes, se obtiene el modelo siguiente:

$$
\begin{aligned}
C \cdot derT &= Q_{\text{in}} - Q_{\text{out}} && (1.10)\\
Q_{\text{out}} &= k \cdot (T - T_{\text{ext}}) && (1.11)\\
T_{\text{ext}} &= 275 + 10 \cdot \sin\left(\frac{2 \cdot \pi \cdot t}{3600 \cdot 24}\right) && (1.12)\\
e &= T_{sp} - T && (1.13)\\
derI &= e && (1.14)\\
u &= k_p \cdot e + \frac{1}{k_I} \cdot I && (1.15)\\
Q_{\text{in}} &= K_c \cdot u && (1.16)\\
T_{sp} &= f(t) && (1.17)
\end{aligned}
$$

Las variables del modelo pueden clasificarse en conocidas (el tiempo, los parámetros, las constantes y las variables de estado) y desconocidas (las variables algebraicas y las derivadas de las variables de estado):

– Conocidas: t
C, k, k_p, k_I, K_c
T, I

– Desconocidas: Q_{in}, Q_{out}, T_{ext}, e, T_{sp}, u
$derT$, $derI$

Con el fin de analizar si el modelo es *estructuralmente singular*, se comprueba que:

1. El número de ecuaciones y de variables desconocidas (incógnitas) es el mismo. Este modelo tiene 8 ecuaciones y 8 incógnitas.

2. Cada incógnita puede emparejarse con una ecuación en que aparezca y con la cual no se haya emparejado ya otra incógnita.

$$
\begin{array}{lll l}
derT & \rightarrow & C \cdot derT = Q_{\text{in}} - Q_{\text{out}} & \text{Ec. (1.10)}\\
Q_{\text{out}} & \rightarrow & Q_{\text{out}} = k \cdot (T - T_{\text{ext}}) & \text{Ec. (1.11)}\\
T_{\text{ext}} & \rightarrow & T_{\text{ext}} = 275 + 10 \cdot \sin\left(\frac{2\cdot\pi\cdot t}{3600\cdot 24}\right) & \text{Ec. (1.12)}\\
e & \rightarrow & e = T_{sp} - T & \text{Ec. (1.13)}\\
derI & \rightarrow & derI = e & \text{Ec. (1.14)}\\
u & \rightarrow & u = k_p \cdot e + \frac{1}{k_I} \cdot I & \text{Ec. (1.15)}\\
Q_{\text{in}} & \rightarrow & Q_{\text{in}} = K_c \cdot u & \text{Ec. (1.16)}\\
T_{sp} & \rightarrow & T_{sp} = f(t) & \text{Ec. (1.17)}
\end{array}
$$

Aplicando el algoritmo de la partición, se obtiene el siguiente modelo ordenado y resuelto (existen otras posibles ordenaciones de las ecuaciones que son igualmente válidas):

$$
\begin{aligned}
[T_{sp}] &= f(t) && (1.18)\\
[T_{\text{ext}}] &= 275 + 10 \cdot \sin\left(\frac{2 \cdot \pi \cdot t}{3600 \cdot 24}\right) && (1.19)\\
[e] &= T_{sp} - T && (1.20)\\
[u] &= k_p \cdot e + \frac{1}{k_I} \cdot I && (1.21)\\
[Q_{\text{in}}] &= K_c \cdot u && (1.22)\\
[Q_{\text{out}}] &= k \cdot (T - T_{\text{ext}}) && (1.23)\\
[derT] &= \frac{Q_{\text{in}} - Q_{\text{out}}}{C} && (1.24)\\
[derI] &= e && (1.25)
\end{aligned}
$$

El modelo posee dos variables de estado, por lo que tiene dos grados de libertad. En la Figura 1.4 se muestra el diagrama de flujo para la simulación del modelo, empleando el método de integración de Euler explícito y considerando como condición de finalización que el tiempo simulado alcance el valor 86400 s. Se han asignado valores arbitrarios a los valores iniciales de las variables de estado. Se han definido tres parámetros adicionales, que se emplean en el cálculo del valor de consigna de la temperatura: $\tau = 6 \cdot 3600$ s, $T_{sp,1} = 300$ K y $T_{sp,2} = 293$ K.

Se ha seleccionado un tamaño del paso igual a 0.1 s. Para ello, se ha repetido la simulación empleando diferentes tamaños del paso, encontrándose que el error cometido escogiendo 0.1 s es admisible para el propósito de este estudio.

Obsérvese que, en el algoritmo mostrado en la Figura 1.4, la variable T_{sp} es tratada como una variable de tiempo continuo, calculándose su valor en cada paso de integración. Otra opción hubiera sido considerar a la variable T_{sp} como una variable de tiempo discreto, cuyo valor cambia en determinados eventos, manteniéndose constante durante la resolución del problema de tiempo continuo.

En la Figura 1.5 se muestra el correspondiente algoritmo para la simulación. La variable de estado discreta α determina si la consigna para la temperatura se encuentra en su valor alto ($\alpha = true$) o bajo ($\alpha = false$). La condición de disparo del evento es que el tiempo transcurrido desde el anterior evento sea igual o mayor que $\tau = 6 \cdot 3600$ s.

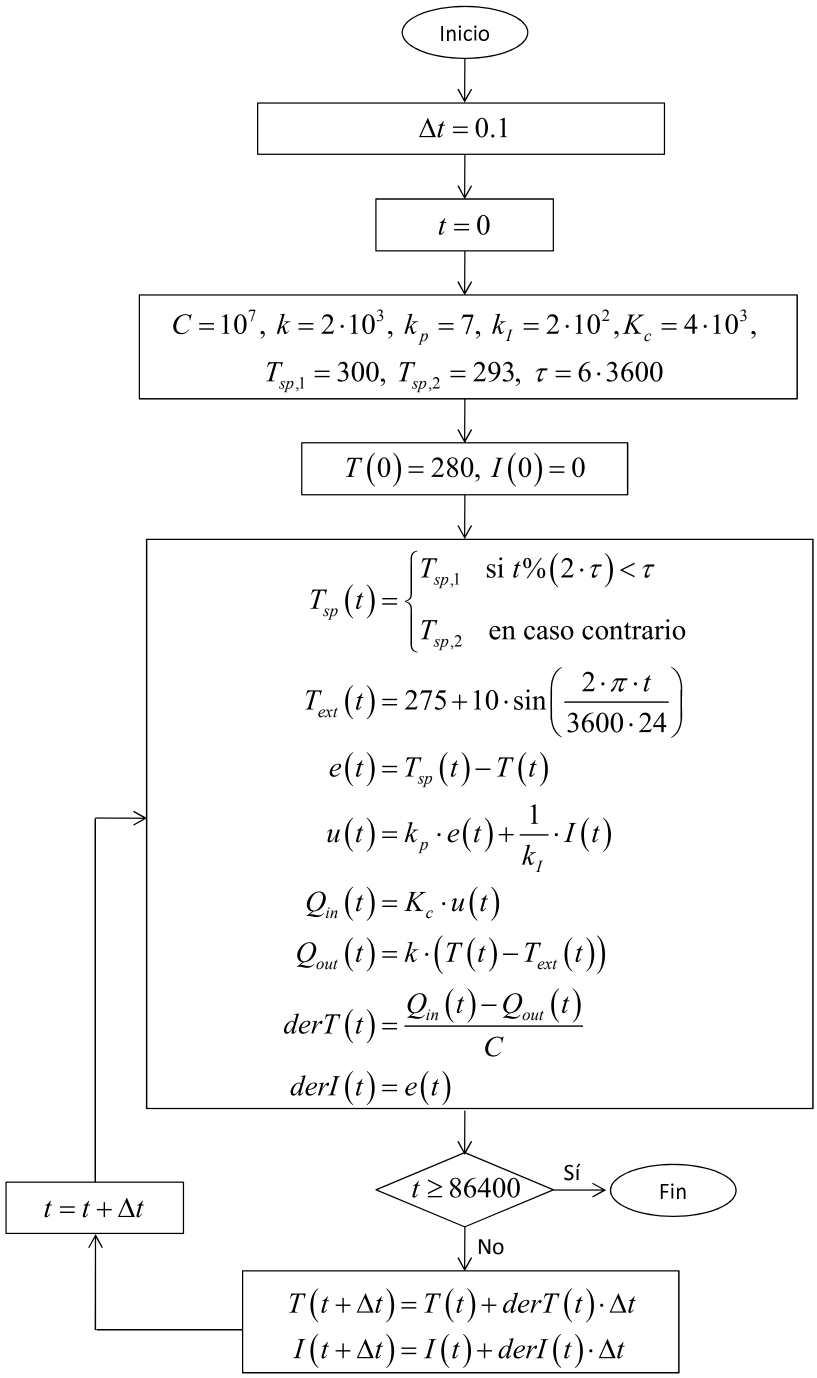

Figura 1.4: Diagrama de flujo de la simulación del modelo, evaluando T_{sp} como variable de tiempo continuo.

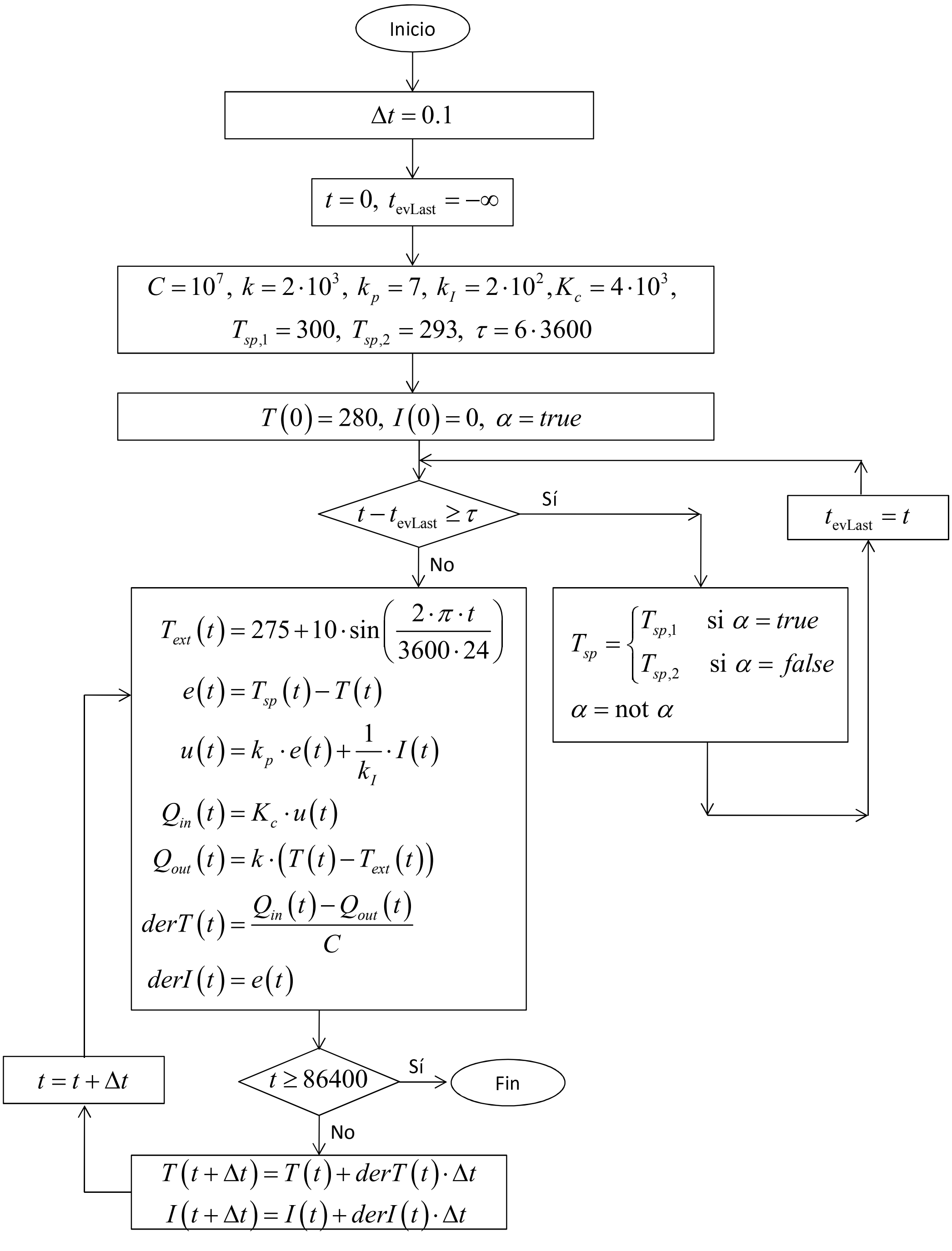

Figura 1.5: Diagrama de flujo de la simulación del modelo, evaluando T_{sp} como variable de tiempo discreto.

Se muestra a continuación el código R que implementa el algoritmo mostrado en la Figura 1.4, con un intervalo de comunicación de 1 s.

```
t_fin <- 86400   # Valor final del tiempo
h     <- 0.1     # Tamaño del paso de integración
t_com <- 1       # Intervalo de comunicación
# Parámetros
C    <- 1E7
k    <- 2E3
kp   <- 7
kI   <- 2E2
Kc   <- 4e3
Tsp1 <- 300
Tsp2 <- 293
tau  <- 6*3600
# Inicialización del tiempo
t <- 0
# Valor inicial del estado
T <- 280
I <- 0
# Inicialización resultados salida
t_plot      <- numeric(0)
T_plot      <- numeric(0)
Tsp_plot    <- numeric(0)
Text_plot   <- numeric(0)
t_ultimaCom <- -Inf
# Bucle de la simulación
termina <- F
while ( !termina ) {
   termina <- (t > t_fin)
   # Cálculo variables algebraicas en t
   if ( t %% (2*tau) < tau ) {
      Tsp <- Tsp1
   } else {
      Tsp <- Tsp2
   }
   Text <- 275+10*sin(2*pi*t/(3600*24))
   e    <- Tsp - T
   u    <- kp*e + I/kI
   Qin  <- Kc*u
   Qout <- k*(T-Text)
   derT <- (Qin - Qout)/C
   derI <- e
   # Almacenar resultados de salida
   if ( (t - t_ultimaCom) >= t_com | termina ) {
      t_plot    <- c(t_plot, t)
      T_plot    <- c(T_plot, T)
      Tsp_plot  <- c(Tsp_plot, Tsp)
      Text_plot <- c(Text_plot, Text)
      t_ultimaCom <- t
   }
   # Valor en t+h de los estados
   T <- T + h*derT
   I <- I + h*derI
   # Actualización del valor del tiempo
   t <- t + h
}
```

```
# Representación gráfica
# Para poder usar minor.tick es necesario instalar el paquete Hmisc
# Configuración de la representación gráfica

plot(c(0,t_fin), c(260,310), type = "n", xlab = "t [s]",
     ylab = "T (rojo), Tsp (azul), Text (naranja) [K]" )
minor.tick(nx = 10, ny = 10, tick.ratio = 0.5)
abline(v = seq(0,t_fin,3600), h = seq(260,310,10),
       lwd = 0.5, lty = 3)
points(t_plot, T_plot, pch = ".", col="red")
points(t_plot, Tsp_plot, pch = ".", col="blue")
points(t_plot, Text_plot, pch = ".", col="orange")
title("Temperaturas", cex = 1)
```

En la Figura 1.6 se muestra la gráfica obtenida al ejecutar el código R anterior.

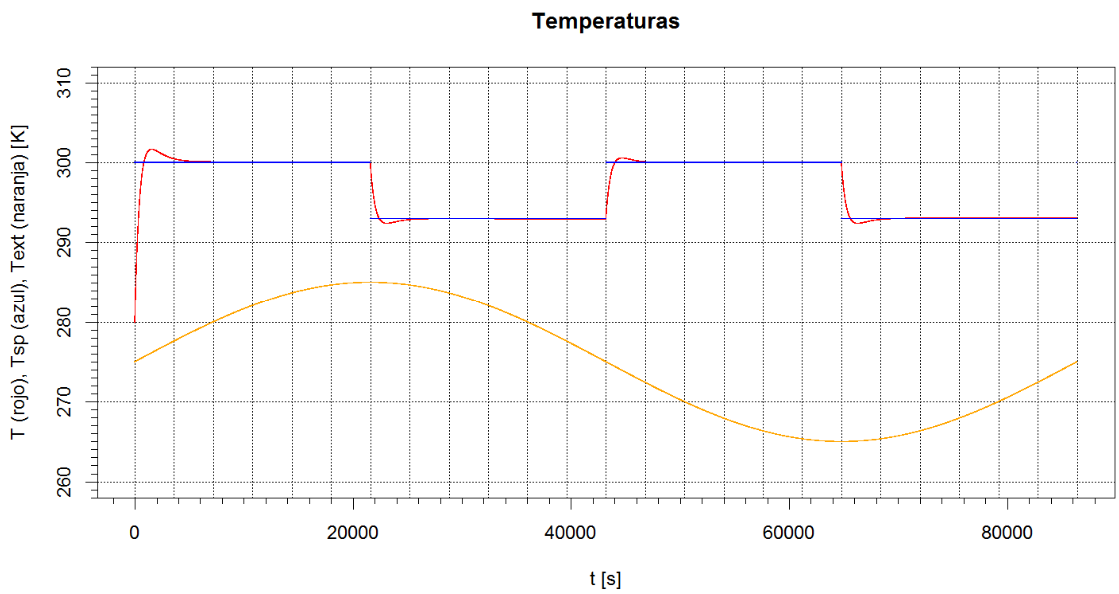

Figura 1.6: Resultado de la simulación del código R.

EJERCICIO 3

Describa en lenguaje Modelica el sistema mostrado en la Figura 1.2 de las dos maneras siguientes:

1. Como un modelo atómico, que viene descrito por las ecuaciones que usted ha planteado al resolver el ejercicio anterior.

2. Descomponga el sistema en partes y programe una librería en lenguaje Modelica cuyos componentes describan estas partes. Emplee dicha librería para modelar en Modelica el sistema como un modelo compuesto.

Asigne a los parámetros los valores indicados en el ejercicio anterior y simule los dos modelos anteriores durante 86400 s. Represente gráficamente la evolución frente al tiempo de T, T_{sp} y T_{ext}, mostrando las gráficas en la memoria.

Solución al Ejercicio 3

El modelo atómico en lenguaje Modelica se muestra en Código 1.1. En este caso se ha optado por definir una variable de estado Booleana (`alpha`) que determina si el valor de consigna de la temperatura está a su nivel alto o bajo. El valor de la variable `alpha` cambia en los eventos definidos por una cláusula when. El resultado de la simulación está representado en la Figura 1.7.

```
model TermicoVivienda
  parameter Real C(unit="J/K") = 1e7;
  parameter Real k(unit="W/K") = 2e3;
  parameter Real kp(unit="V/K") = 7;
  parameter Real kI(unit="K.s/V") = 2e2;
  parameter Real Kc(unit="W/V") = 4e3;
  Real T(unit="K",start=280,fixed=true);
  Real Tsp(unit="K");
  Real Qin(unit="W"),Qout(unit="W");
  Real Text(unit="K");
  Real e(unit="K");
  Real I(unit="K.s",start=0,fixed=true);
  Real u(unit="V");
  parameter Real tau(unit="s")  = 6*3600;
  parameter Real Tsp1(unit="K") = 300;
  parameter Real Tsp2(unit="K") = 293;
  Boolean alpha(start=true, fixed=true);
equation
  C*der(T) = Qin -Qout;
  Qout = k*(T-Text);
  Text = 275+10*sin(2*Modelica.Constants.pi*time/3600/24);
  e = Tsp - T;
  der(I) = e;
  u = kp*e + 1/kI*I;
  Qin = Kc*u;
  // Tsp = if mod(time,2*tau)<tau then Tsp1 else Tsp2;
  Tsp = if alpha then Tsp1 else Tsp2;
  when sample(tau, tau) then
    alpha = not pre(alpha);
  end when;
  annotation (experiment(StopTime=86400));
end TermicoVivienda;
```

Código 1.1: Ejercicio 3.1: modelo atómico del sistema.

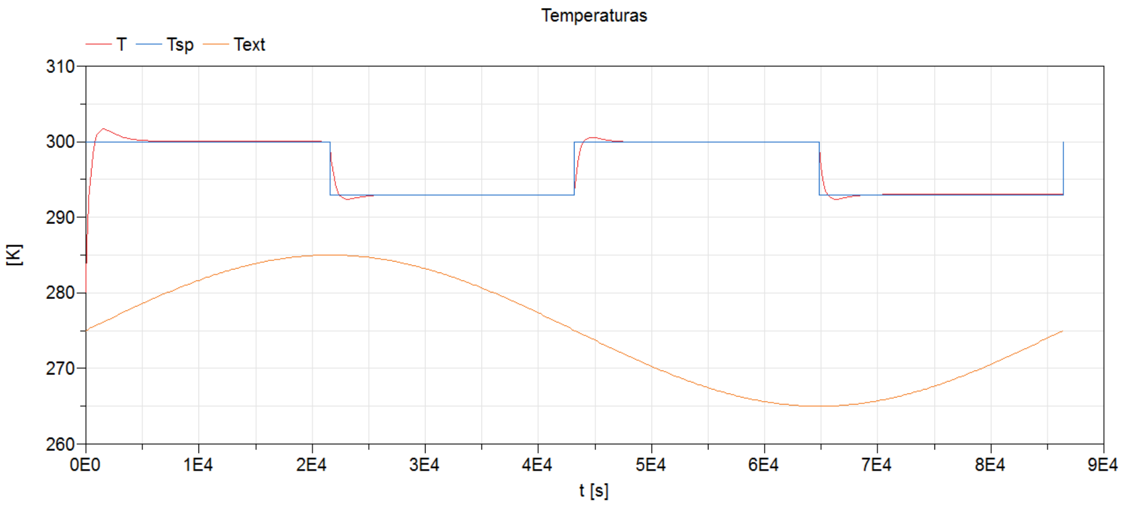

Figura 1.7: Resultado de la simulación usando Dymola.

En el Código 1.2 y 1.3 se muestra una posible forma de modelar los componentes, organizándolos en una librería, y de componer el sistema mediante dichos componentes.

```
package ViviendaTermico

import SI = Modelica.SIunits;

package Interfaces

  connector ConectCalor
     SI.Temperature T;
     flow SI.Power Q;
  end ConectCalor;

  connector ConectSignal
     Real s;
  end ConectSignal;

end Interfaces;

package Componentes

model Vivienda
  Interfaces.ConectCalor conectCalor;
  Interfaces.ConectSignal sensorTemp;
  parameter SI.HeatCapacity C = 1e7;
equation
  C*der(conectCalor.T) = conectCalor.Q;
  sensorTemp.s = conectCalor.T;
equation

end Vivienda;

model ControladorPI
  Interfaces.ConectSignal conect_I,conect_O,conect_SP;
  parameter Real kp(unit="V/K") = 7;
  parameter Real kI(unit="K.s/V") = 2e2;
  Real e(unit="K");
  Real I(unit="K.s",start=0,fixed=true);
equation
  e = conect_SP.s - conect_I.s;
  der(I) = e;
  conect_O.s = kp*e + 1/kI*I;
end ControladorPI;


model Climatizacion
  Interfaces.ConectCalor  connectQ;
  Interfaces.ConectSignal connectU;
  parameter Real Kc(unit="W/V") = 4e3;
equation
  -connectQ.Q = Kc*connectU.s;
end Climatizacion;
```

Código 1.2: Ejercicio 3.2: librería (1/2).

```
model ResistenciaTermica
  parameter SI.ThermalConductance k = 2e3;
  Interfaces.ConectCalor conectCalor1,conectCalor2;
equation
  conectCalor1.Q = k*(conectCalor1.T - conectCalor2.T);
  conectCalor2.Q = - conectCalor1.Q;
end ResistenciaTermica;

model Entorno
  Interfaces.ConectCalor conectCalor;
equation
  conectCalor.T = 275+10*sin(2*Modelica.Constants.pi*time/3600/24);
end Entorno;

model ConsignaT
  Interfaces.ConectSignal conectSP;
  Boolean alpha(start=true, fixed=true);
  parameter SI.Time tau  = 6*3600;
  parameter SI.Temperature Tsp1 = 300;
  parameter SI.Temperature Tsp2 = 293;
equation
  conectSP.s = if alpha then Tsp1 else Tsp2;
  when sample(tau, tau) then
    alpha = not pre(alpha);
  end when;
end ConsignaT;

end Componentes;

model Sistema
  Componentes.Vivienda vivienda(C=C);
  Componentes.ControladorPI controladorPI(kp=kp, kI=kI);
  Componentes.Climatizacion climatizacion(Kc=Kc);
  Componentes.ResistenciaTermica resistTerm(k=k);
  Componentes.Entorno entorno;
  Componentes.ConsignaT consignaT(tau=tau,Tsp1=Tsp1,Tsp2=Tsp2);
  parameter SI.HeatCapacity C = 1e7;
  parameter SI.ThermalConductance k = 2e3;
  parameter Real kp(unit="V/K") = 7;
  parameter Real kI(unit="K.s/V") = 2e2;
  parameter Real Kc(unit="W/V") = 4e3;
  parameter SI.Time tau  = 6*3600;
  parameter SI.Temperature Tsp1 = 300;
  parameter SI.Temperature Tsp2 = 293;
equation
  connect(consignaT.conectSP, controladorPI.conect_SP);
  connect(controladorPI.conect_I, vivienda.sensorTemp);
  connect(controladorPI.conect_O, climatizacion.connectU);
  connect(climatizacion.connectQ, vivienda.conectCalor);
  connect(vivienda.conectCalor, resistTerm.conectCalor1);
  connect(resistTerm.conectCalor2, entorno.conectCalor);
  annotation (experiment(StopTime=86400));
end Sistema;

end ViviendaTermico;
```

Código 1.3: Ejercicio 3.2: librería (2/2).

EJERCICIO 4

Escriba un *script* en FlexPDE para obtener el mapa de contorno de distribución de temperatura y los vectores correspondientes al flujo de calor del sistema descrito a continuación. Se trata de un problema de dos dimensiones cuya geometría se muestra en la siguiente figura.

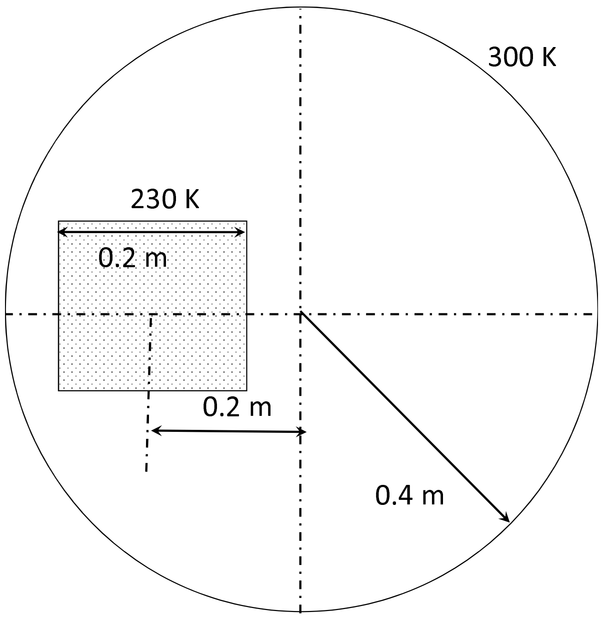

Figura 1.8: Geometría del sistema.

El sistema que se desea simular está formado por un círculo de un material de conductividad $\kappa_1 = 1$ W/(K·m), que tiene en su interior un cuadrado de un material aislante de conductividad $\kappa_2 = 0.01$ W/(K·m). El círculo tiene de radio 0.4 m y su contorno exterior se mantiene a 300 K. El cuadrado interior tiene de lado 0.2 m y su contorno exterior se mantiene a 230 K. El centro del cuadrado está separado 0.2 m del centro del círculo.

Obtenga los dos siguientes gráficos empleando FlexPDE: un gráfico con las curvas de nivel de la temperatura y un gráfico vectorial del flujo de calor. En la memoria del trabajo ha de mostrar el código del script de FlexPDE correspondiente y los dos gráficos generados por FlexPDE.

Solución al Ejercicio 4

El *script* está a continuación. El gráfico de la temperatura se muestra en la Figura 1.9 y el gráfico vectorial del flujo de calor en la Figura 1.10.

```
TITLE
   'ProblemaCalor'
SELECT
   errlim=1e-3     spectral_colors
VARIABLES
   temp
DEFINITIONS
  d=0.1     r2=0.4    k =1
  L =0.2
EQUATIONS
   div(-k* grad( temp))=0
BOUNDARIES
region 'medio1'
   start 'exterior'  (0,-r2) value(temp)=300     { Exterior}
   arc( center=0,0)   angle=360
region 'medio2'
  k =0.01
   start 'cuadrado'  (-d,-L/2)   value(temp)= 230   line to (-d,L/2) !Lado 1
   value(temp) = 230  LINE to (-L-d,L/2)       !Lado 2
   value(temp) = 230  LINE to (-L-d,-L/2)    !Lado 3
   value(temp) = 230  LINE to close    !Lado 4
PLOTS
   contour(temp)           vector( -k*grad(temp))
END
```

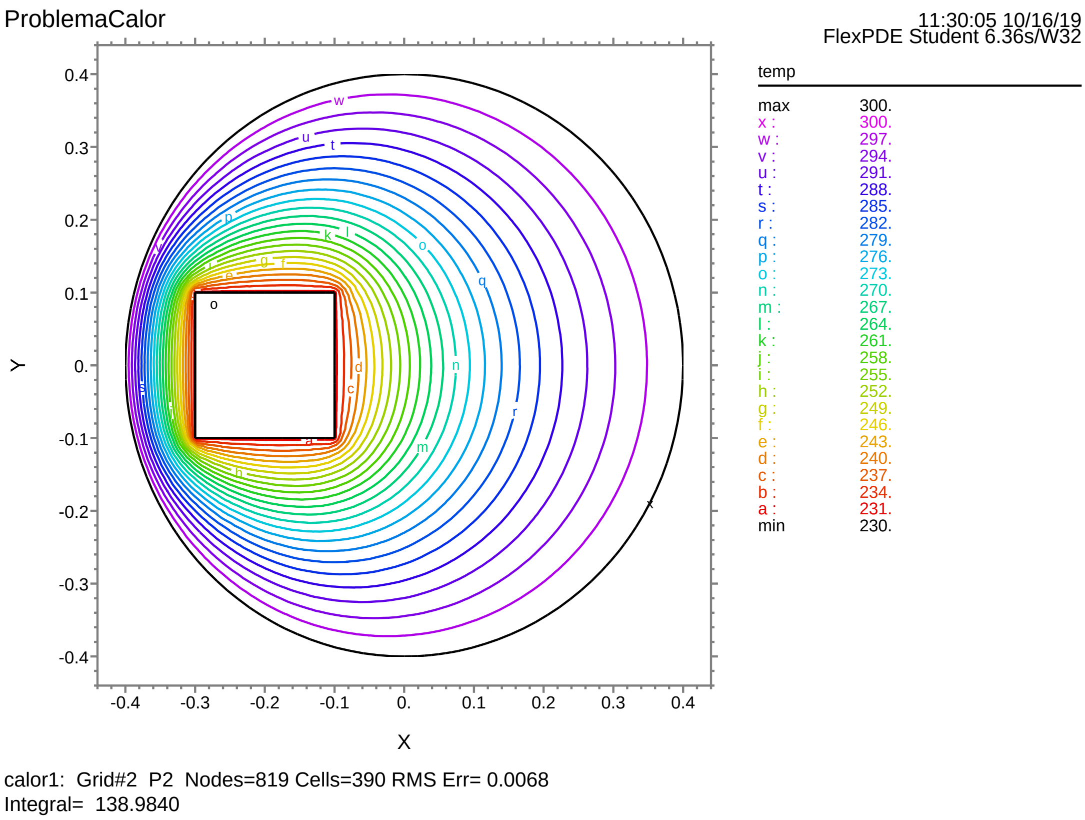

Figura 1.9: Curvas de temperatura.

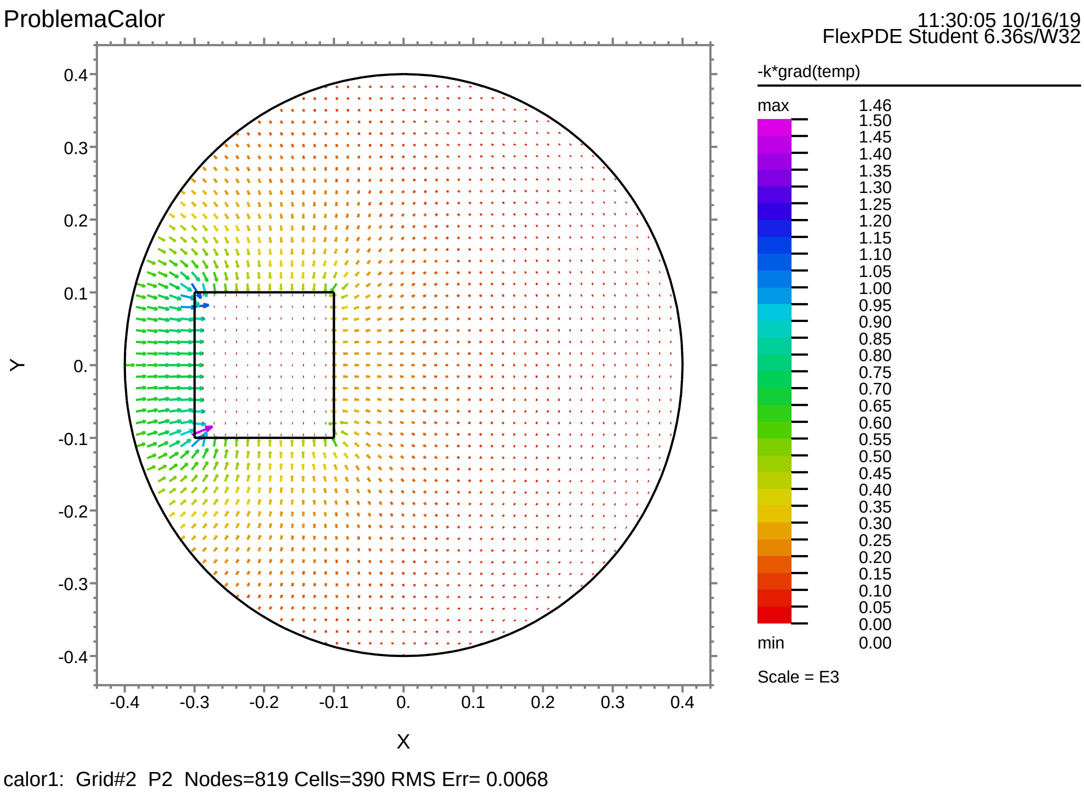

Figura 1.10: Flujo de calor