

MÉTODOS DE SIMULACIÓN Y MODELADO

Solución al Trabajo Práctico - Septiembre de 2019

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 12th 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.

Solución al Ejercicio 1

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

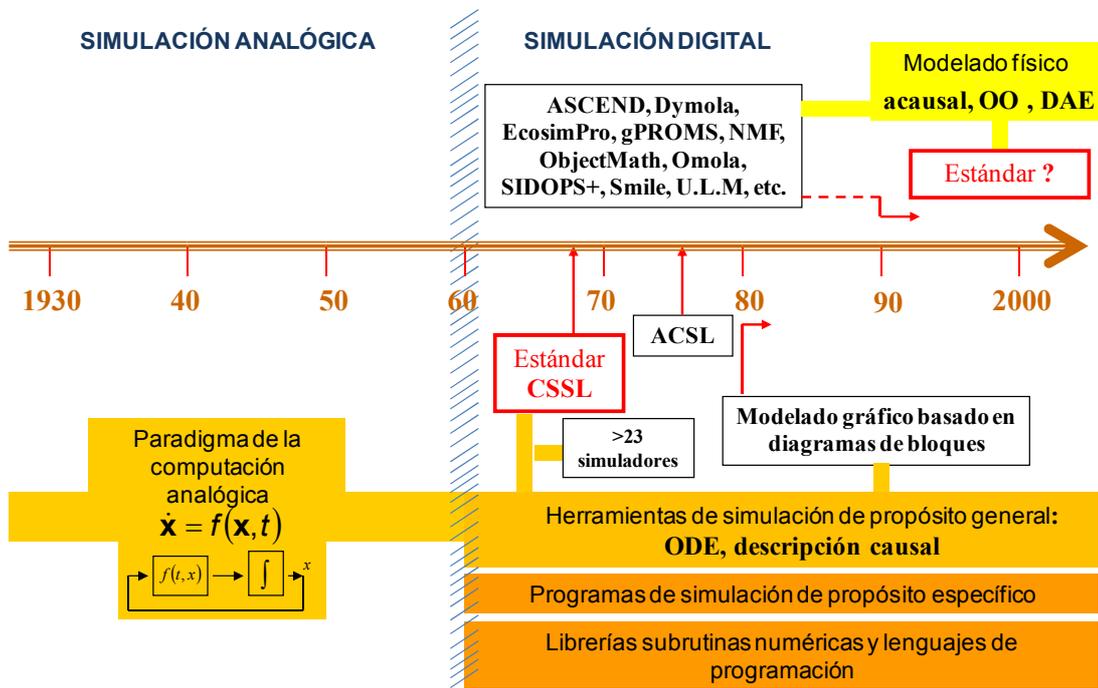


Figura 1.1: Evolución del modelado y simulación de tiempo continuo.

EJERCICIO 2

En la Figura 1.2 se muestra un circuito eléctrico compuesto por un generador sinusoidal de tensión, un diodo, dos condensadores, dos resistencias y un interruptor. Se ha señalado el nodo escogido como referencia para el voltaje. Se ha asignado nombre a la tensión en los nodos y a la corriente que circula por los componentes.

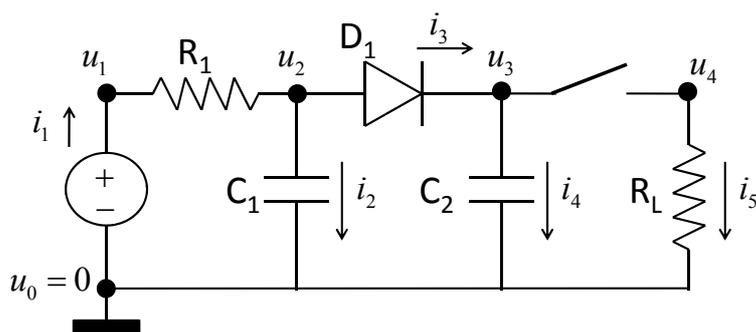


Figura 1.2: Diagrama de un circuito eléctrico.

Los valores de las resistencias y capacidades son conocidos y constantes: $R_1 = 20 \text{ ohm}$, $R_L = 1000 \text{ ohm}$, $C_1 = 10^{-5} \text{ F}$, $C_2 = 0.01 \text{ F}$. El generador sinusoidal de tensión tiene una amplitud de 12 V y una frecuencia de 60 Hz . El diodo es descrito empleando el modelo de Shockley y asignando a sus parámetros los valores siguientes: corriente de saturación, $I_s = 10^{-9} \text{ A}$; tensión térmica, $V_t = 0.025 \text{ V}$.

El interruptor eléctrico es ideal. Puede encontrarse en dos fases: abierto y cerrado. Mientras está abierto no permite el paso de corriente. Mientras está cerrado la caída de tensión entre sus bornes es cero. El interruptor cambia de fase cada T segundos, siendo $T = 0.05 \text{ s}$ un parámetro del modelo.

1. Escriba las ecuaciones del modelo del circuito.
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 es que el tiempo simulado alcance el valor 0.5 s .
4. Programe el algoritmo anterior en lenguaje R y ejecute la simulación. Represente gráficamente u_1 , u_3 y u_4 frente al tiempo. Explique qué criterio ha seguido para escoger el tamaño del paso de integración.

Solución al Ejercicio 2

El circuito puede modelarse de la forma descrita a continuación, donde t representa el tiempo. No se muestran las ecuaciones en las cuales se asigna valor a las constantes y parámetros.

$$\begin{aligned}
 u_1 &= U_1 \cdot \sin(2 \cdot \pi \cdot f_1 \cdot t) \\
 u_1 - u_2 &= i_1 \cdot R_1 \\
 C_1 \cdot \frac{du_2}{dt} &= i_2 \\
 i_3 &= I_s \cdot \left(\exp\left(\frac{u_2 - u_3}{V_t}\right) - 1 \right) \\
 C_2 \cdot \frac{du_3}{dt} &= i_4 \\
 0 &= \alpha \cdot (u_3 - u_4) + (1 - \alpha) \cdot i_5 \\
 \alpha &= \text{STEP}(t, T) \\
 u_4 &= i_5 \cdot R_L \\
 i_1 &= i_2 + i_3 \\
 i_3 &= i_4 + i_5
 \end{aligned}$$

La función del tiempo STEP describe una forma de onda rectangular, que con periodo T va alternando entre los valores cero y uno. Mientras $\alpha = 0$ el interruptor está abierto ($i_5 = 0$) y mientras $\alpha = 1$ está cerrado ($u_3 = u_4$).

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

- Parámetros y constantes: $U_1, \pi, f_1, R_1, C_1, I_s, V_t, C_2, T, R_L$
- Variables de estado: u_2, u_3
- Variables algebraicas: $u_1, u_4, i_1, i_2, i_3, i_4, i_5, \alpha$

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

$$\frac{du_2}{dt} \rightarrow \text{der}u_2 \qquad \frac{du_3}{dt} \rightarrow \text{der}u_3$$

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

$$u_1 = U_1 \cdot \sin(2 \cdot \pi \cdot f_1 \cdot t) \quad (1.1)$$

$$u_1 - u_2 = i_1 \cdot R_1 \quad (1.2)$$

$$C_1 \cdot \text{der}u_2 = i_2 \quad (1.3)$$

$$i_3 = I_s \cdot \left(\exp\left(\frac{u_2 - u_3}{V_t}\right) - 1 \right) \quad (1.4)$$

$$C_2 \cdot \text{der}u_3 = i_4 \quad (1.5)$$

$$0 = \alpha \cdot (u_3 - u_4) + (1 - \alpha) \cdot i_5 \quad (1.6)$$

$$\alpha = \text{STEP}(t, T) \quad (1.7)$$

$$u_4 = i_5 \cdot R_L \quad (1.8)$$

$$i_1 = i_2 + i_3 \quad (1.9)$$

$$i_3 = i_4 + i_5 \quad (1.10)$$

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
 $U_1, \pi, f_1, R_1, C_1, I_s, V_t, C_2, T, R_L$
 u_2, u_3
- Desconocidas: $u_1, u_4, i_1, i_2, i_3, i_4, i_5, \alpha$
 $\text{der}u_2, \text{der}u_3$

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 10 ecuaciones y 10 incógnitas.
2. Cada incógnita puede emparejarse con una ecuación en que aparezca y con la cual no se haya emparejado ya otra incógnita.

$$u_1 \rightarrow u_1 = U_1 \cdot \sin(2 \cdot \pi \cdot f_1 \cdot t) \quad \text{Ec. (1.1)}$$

$$i_1 \rightarrow u_1 - u_2 = i_1 \cdot R_1 \quad \text{Ec. (1.2)}$$

$$deru_2 \rightarrow C_1 \cdot deru_2 = i_2 \quad \text{Ec. (1.3)}$$

$$i_3 \rightarrow i_3 = I_s \cdot \left(\exp\left(\frac{u_2 - u_3}{V_t}\right) - 1 \right) \quad \text{Ec. (1.4)}$$

$$deru_3 \rightarrow C_2 \cdot deru_3 = i_4 \quad \text{Ec. (1.5)}$$

$$u_4 \rightarrow 0 = \alpha \cdot (u_3 - u_4) + (1 - \alpha) \cdot i_5 \quad \text{Ec. (1.6)}$$

$$\alpha \rightarrow \alpha = \text{STEP}(t, T) \quad \text{Ec. (1.7)}$$

$$i_5 \rightarrow u_4 = i_5 \cdot R_L \quad \text{Ec. (1.8)}$$

$$i_2 \rightarrow i_1 = i_2 + i_3 \quad \text{Ec. (1.9)}$$

$$i_4 \rightarrow i_3 = i_4 + i_5 \quad \text{Ec. (1.10)}$$

Aplicando el algoritmo de la partición, se obtiene la matriz de incidencia BLT. Se muestra a continuación, con los bloques diagonales señalados. Obsérvese que las Ecs. (1.6) y (1.8) forman un lazo algebraico lineal del cual se evalúan u_4 e i_5 .

$$\begin{array}{l}
 u_1 = U_1 \cdot \sin(2 \cdot \pi \cdot f_1 \cdot t) \\
 \alpha = \text{STEP}(t, T) \\
 u_1 - u_2 = i_1 \cdot R_1 \\
 i_3 = I_s \cdot \left(\exp\left(\frac{u_2 - u_3}{V_t}\right) - 1 \right) \\
 i_1 = i_2 + i_3 \\
 0 = \alpha \cdot (u_3 - u_4) + (1 - \alpha) \cdot i_5 \\
 u_4 = i_5 \cdot R_L \\
 i_3 = i_4 + i_5 \\
 C_1 \cdot deru_2 = i_2 \\
 C_2 \cdot deru_3 = i_4
 \end{array}
 \begin{pmatrix}
 \boxed{X} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & \boxed{X} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 X & 0 & \boxed{X} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & \boxed{X} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & X & X & \boxed{X} & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & X & 0 & 0 & 0 & \boxed{X} & \boxed{X} & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & \boxed{X} & \boxed{X} & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & X & 0 & 0 & X & \boxed{X} & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & X & 0 & 0 & 0 & \boxed{X} & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & X & 0 & \boxed{X} & 0
 \end{pmatrix}$$

Resolviendo el lazo algebraico lineal y despejando de cada una de las demás ecuaciones la incógnita a evaluar, se obtiene el siguiente modelo ordenado y resuelto (existen otras posibles ordenaciones de las ecuaciones que son igualmente válidas):

$$\begin{aligned}
[u_1] &= U_1 \cdot \sin(2 \cdot \pi \cdot f_1 \cdot t) \\
[\alpha] &= \text{STEP}(t, T) \\
[i_1] &= \frac{u_1 - u_2}{R_1} \\
[i_3] &= I_s \cdot \left(\exp\left(\frac{u_2 - u_3}{V_t}\right) - 1 \right) \\
[i_2] &= \frac{i_1 - i_3}{\alpha \cdot u_3} \\
[i_5] &= \frac{i_2}{\alpha - 1 + \alpha \cdot R_L} \\
[u_4] &= i_5 \cdot R_L \\
[i_4] &= i_3 - i_5 \\
[deru_2] &= \frac{i_2}{C_1} \\
[deru_3] &= \frac{i_4}{C_2}
\end{aligned}$$

En la Figura 1.3 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 0.5 s. Se han asignado valores arbitrarios a los valores iniciales de las variables de estado. Se ha seleccionado un tamaño del paso para el cual el error del método de integración está por debajo de un valor admisible para este estudio.

La variable α , que describe el modo en el cual se encuentra el interruptor, es de tiempo discreto. La variable t_{evLast} , también de tiempo discreto, almacena el instante en el cual se produjo el último cambio en el valor de α . Obsérvese que la condición de evento ($t - t_{\text{evLast}} \geq T$) determina si debe avanzarse en la resolución del problema de tiempo continuo o si, por el contrario, deben ejecutarse las acciones asociadas al evento.

A continuación, se muestra el código R y en la Figura 1.4 están las gráficas obtenidas de su ejecución. Obsérvese que se ha definido un paso de integración de 10^{-6} s y un intervalo de comunicación de 10^{-3} s. Únicamente se va almacenando durante la ejecución de la simulación, en los instantes de comunicación, el valor del tiempo (t) y de los voltajes u_1 , u_3 y u_4 .

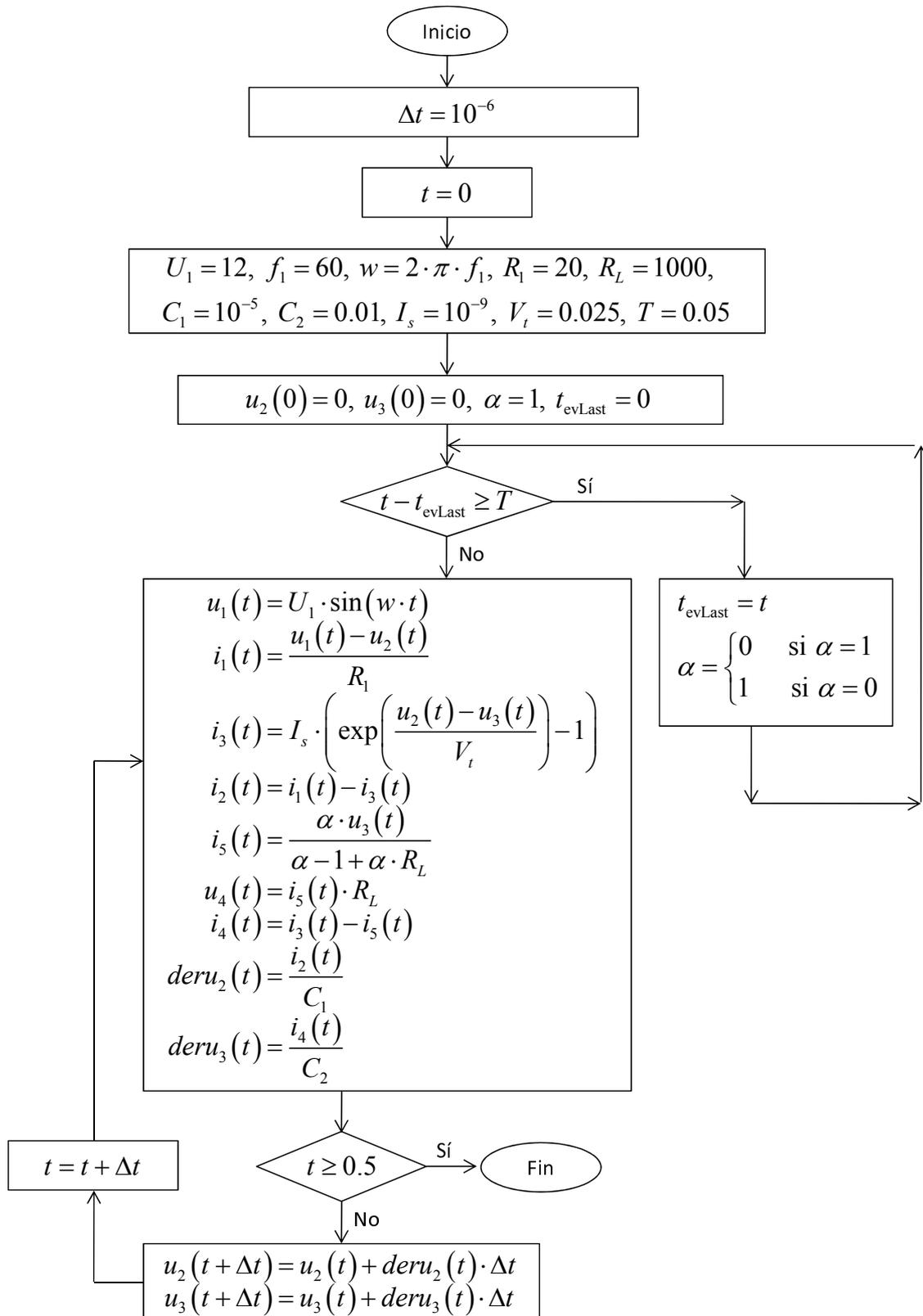


Figura 1.3: Diagrama de flujo de la simulación del modelo del circuito.

```

options(digits=12)
t_fin <- 0.5      # Valor final del tiempo
h      <- 1e-6   # Tamaño del paso de integración
t_com  <- 0.001  # Intervalo de comunicación
# Parámetros
U1 <- 12;  f1 <- 60;    w  <- 2*pi*f1
R1 <- 20;  RL <- 1000
C1 <- 1e-5; C2 <- 0.01
Is <- 1e-9; Vt <- 0.025
T  <- 0.05
# Inicialización del tiempo
t <- 0
# Valor inicial del estado
u2 <- 0
u3 <- 0
alfa <- 1
tEvLast <- 0
# Inicialización resultados salida
t_plot <- numeric(0); u1_plot <- numeric(0)
u3_plot <- numeric(0); u4_plot <- numeric(0)
t_ultimaCom <- -Inf
# Bucle de la simulación
termina <- FALSE
while ( !termina ) {
  if ( t - tEvLast >= T ) { # Ejecución evento
    tEvLast <- t
    alfa <- ifelse(alfa==1, 0, 1)
  } else {
    # Cálculo variables algebraicas en t
    u1 <- U1*sin(w*t)
    i1 <- (u1-u2)/R1
    i3 <- Is*( exp((u2-u3)/Vt) - 1)
    i2 <- i1 - i3
    i5 <- alfa*u3/(alfa-1+alfa*RL)
    u4 <- i5*RL
    i4 <- i3 - i5
    deru2 <- i2/C1
    deru3 <- i4/C2
    # Almacenar resultados de salida
    if ( t - t_ultimaCom >= t_com | t >= t_fin ) {
      t_plot <- c(t_plot, t)
      u1_plot <- c(u1_plot, u1)
      u3_plot <- c(u3_plot, u3)
      u4_plot <- c(u4_plot, u4)
      t_ultimaCom <- t
    }
    # Condición de finalización
    if ( t >= t_fin ) {
      termina <- TRUE
    } else {
      # Valor en t+h de los estados
      u2 <- u2 + h*deru2
      u3 <- u3 + h*deru3
      # Incremento 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
par( mfrow = c(2,1) )
# u1 vs t
par( mai = c(0.3,1.2,0.4,0.1) )
plot(t_plot, u1_plot, lty=1, type="l", col="blue",
      xlim = c(0,t_fin), ylim = c(-U1,U1), xaxt = "n",
      yaxt = "n", xlab = "", ylab = "u1 [V]", cex.lab=0.8 )
axis(2, las = 2)
minor.tick(nx = 2, ny = 5, tick.ratio = 0.5)
abline(v = seq(0,t_fin,10*t_com), h = seq(-U1,U1,1),
        lwd = 0.5, lty = 3)
abline(h = 0, lwd = 1, lty = 1)
# u3, u4 vs t
par( mai=c(1.1,1.2,0,0.1) )
plot(t_plot, u3_plot, type = "n",
      xlim = c(0,t_fin), ylim = c(-1,6),
      yaxt = "n", xlab = "t [s]", ylab = "u3 (azul), u4 (rojo) [V]",
      cex.lab=0.8)
axis(2, las = 2)
minor.tick(nx = 2, ny = 5, tick.ratio = 0.5)
abline(v = seq(0,t_fin,10*t_com), h = seq(-1,6,1),
        lwd = 0.5, lty=3)
abline(h = 0, lwd = 1, lty=1)
points(t_plot, u3_plot, col="blue", pch=20, cex=0.2)
points(t_plot, u4_plot, col="red", pch=20, cex=0.2)

```

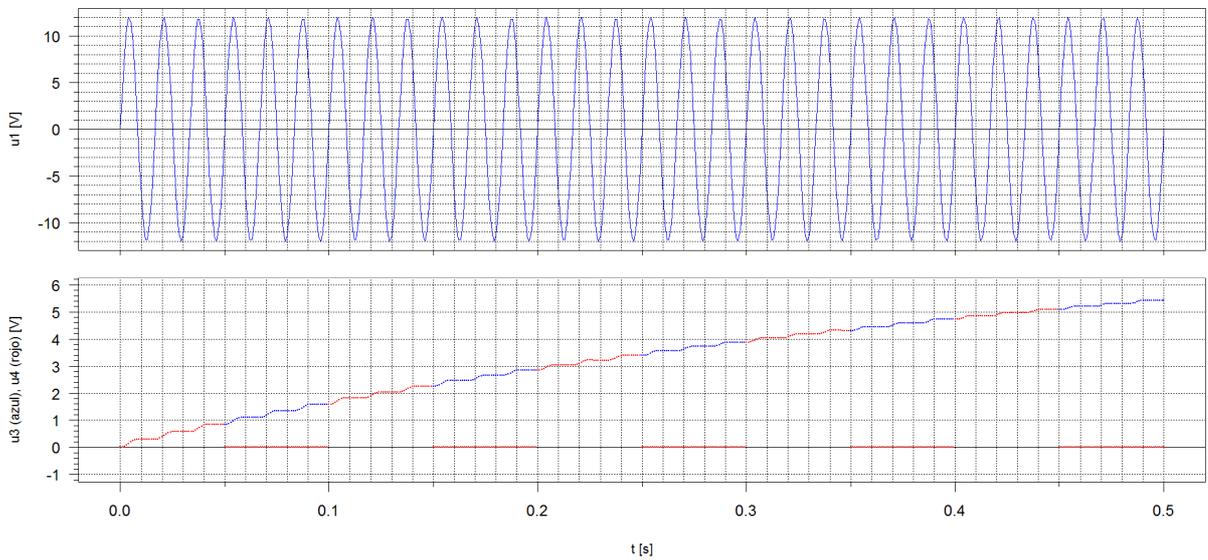


Figura 1.4: Resultado de la simulación del código R.

EJERCICIO 3

Describa en lenguaje Modelica el circuito mostrado en la Figura 1.2 de las tres maneras siguientes:

1. Como un modelo compuesto, empleando componentes de la Librería Estándar de Modelica. Emplee la versión de dicha librería que proporcione el entorno de modelado que usted esté empleando.
2. Como un modelo atómico, que viene descrito por las ecuaciones que usted ha planteado al contestar a la pregunta anterior.
3. Cree su propia librería eléctrica, de manera que contenga únicamente los componentes necesarios para componer el circuito de la Figura 1.2. Para desarrollar su librería, puede basarse en la librería eléctrica que aparece descrita en el texto base o en la Librería Estándar de Modelica. A continuación, defina el circuito como un modelo compuesto, instanciando y conectando componentes de la librería que ha creado.

Simule los tres modelos anteriores del circuito durante 0.5 s. Represente gráficamente u_1 , u_3 y u_4 frente al tiempo en cada caso, mostrando las gráficas en la memoria.

Solución al Ejercicio 3

En la Figura 1.5 se muestra el diagrama del circuito compuesto empleando Dymola versión 2019 FD01 y la MSL 3.2.2. En el Código 1.1 y 1.2, se muestra el correspondiente código Modelica.

La descripción del circuito como modelo atómico es el Código 1.3. El resultado de la ejecución de este modelo se muestra en la Figura 1.6.

Finalmente, la librería eléctrica y el circuito compuesto usándola están en Código 1.4 – 1.6.

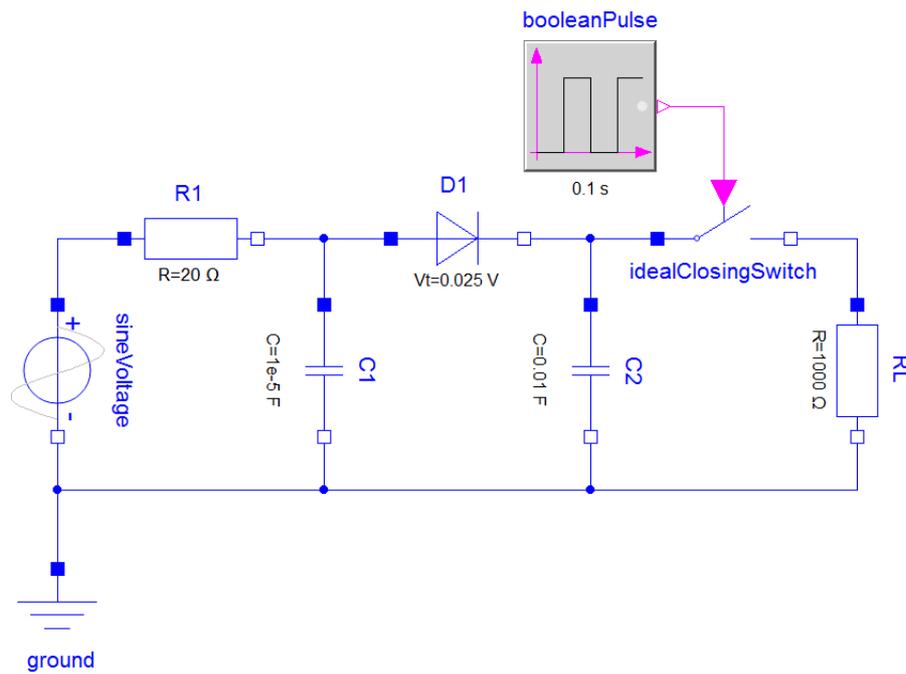


Figura 1.5: Diagrama del circuito compuesto usando la MSL 3.2.2.

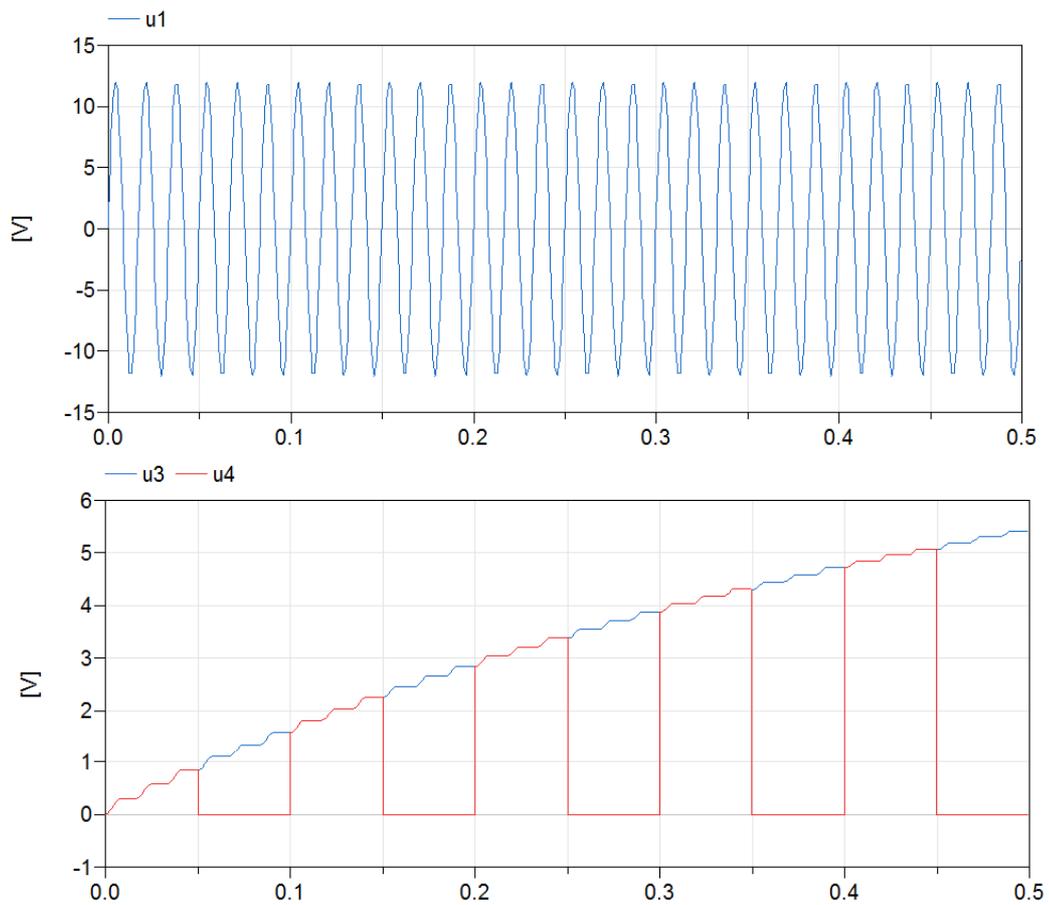


Figura 1.6: Resultado de la simulación del modelo atómico.

```

model circuitoMSL
  Modelica.Electrical.Analog.Sources.SineVoltage
    sineVoltage(V=12,freqHz=60)
      annotation (Placement(transformation(
        extent={{-10,-10},{10,10}},
        rotation=-90,origin={-50,10})));
  Modelica.Electrical.Analog.Basic.Ground ground
    annotation (Placement(transformation(
      extent={{-60,-40},{-40,-20})));
  Modelica.Electrical.Analog.Basic.Resistor R1(R=20)
    annotation (Placement(transformation(
      extent={{-40,20},{-20,40})));
  Modelica.Electrical.Analog.Basic.Capacitor C1(v(start=0),C=1e-5)
    annotation (Placement(transformation(
      extent={{-10,-10},{10,10}},
      rotation=-90,origin={-10,10})));
  Modelica.Electrical.Analog.Basic.Capacitor C2(v(start=0),C=0.01)
    annotation (Placement(transformation(
      extent={{-10,-10},{10,10}},
      rotation=-90,origin={30,10})));
  Modelica.Electrical.Analog.Basic.Resistor RL(R=1000)
    annotation (Placement(transformation(
      extent={{-10,-10},{10,10}},
      rotation=-90,origin={70,10})));
  Modelica.Blocks.Sources.BooleanPulse booleanPulse(period=0.1)
    annotation (Placement(transformation(
      extent={{20,40},{40,60})));
  Modelica.Electrical.Analog.Ideal.IdealClosingSwitch
    idealClosingSwitch(Ron=0,Goff=0)
      annotation (Placement(transformation(
        extent={{40,20},{60,40})));
  Modelica.Electrical.Analog.Semiconductors.Diode
    D1(Ids=1e-9,Vt=0.025,Maxexp=1e3,R=1e20)
      annotation (Placement(
        transformation(extent={{0,20},{20,40})));

```

Código 1.1: Modelo compuesto empleando componentes de la MSL (1/2).

```

equation
  connect(sineVoltage.p,R1.p)
    annotation (Line(points={{-50,20},{-50,30},{-40,30}},
      color={0,0,255}));
  connect(R1.n,C1.p)
    annotation (Line(points={{-20,30},{-10,30},{-10,20}},
      color={0,0,255}));
  connect(sineVoltage.n,ground.p)
    annotation (Line(points={{-50,0},{-50,-20},{-50,-20}},
      color={0,0,255}));
  connect(C1.n,ground.p)
    annotation (Line(points={{-10,0},{-10,-8},
      {-50,-8},{-50,-20}},
      color={0,0,255}));
  connect(C2.n,ground.p)
    annotation (Line(points={{30,0},{30,-8},
      {-50,-8},{-50,-20}},
      color={0,0,255}));
  connect(RL.n,ground.p)
    annotation (Line(points={{70,0},{70,-8},
      {-50,-8},{-50,-20}},
      color={0,0,255}));
  connect(idealClosingSwitch.n,RL.p)
    annotation (Line(points={{60,30},{70,30},{70,20}},
      color={0,0,255}));
  connect(booleanPulse.y,idealClosingSwitch.control)
    annotation (Line(points={{41,50},{50,50},{50,37}},
      color={255,0,255}));
  connect(R1.n,D1.p)
    annotation (Line(points={{-20,30},{0,30}},
      color={0,0,255}));
  connect(D1.n,idealClosingSwitch.p)
    annotation (Line(points={{20,30},{40,30}},
      color={0,0,255}));
  connect(D1.n,C2.p)
    annotation (Line(points={{20,30},{30,30},{30,20}},
      color={0,0,255}));
  annotation (uses(Modelica(version="3.2.2")));
end circuitoMSL;

```

Código 1.2: Modelo compuesto empleando componentes de la MSL (2/2).

```

model circuitol
  import SI = Modelica.SIunits;
  SI.Current i1, i2, i3, i4, i5;
  SI.Voltage u1;
  SI.Voltage u2(start=0, fixed=true);
  SI.Voltage u3(start=0, fixed=true);
  SI.Voltage u4;
  Boolean open(start=true, fixed=true);
  // Parametros del generador
  parameter SI.Voltage U1 = 12;
  parameter SI.Frequency f1 = 60;
  parameter SI.AngularFrequency w = 2*Modelica.Constants.pi*f1;
  // Resistencias
  parameter SI.Resistance R1=20;
  parameter SI.Resistance RL=1000;
  // Condensadores
  parameter SI.Capacitance C1=1e-5;
  parameter SI.Capacitance C2=0.01;
  // Diodo
  parameter SI.Current Is=1e-9;
  parameter SI.Voltage Vt=0.025;
  // Interruptor
  parameter SI.Time T = 0.05;
equation
  // Ecuaciones en los nodos
  i1 = i2 + i3;
  i3 = i4 + i5;
  // Relaciones constitutivas
  u1 = U1*sin(w*time);
  u1 - u2 = i1*R1;
  C1*der(u2) = i2;
  i3 = Is*(exp((u2-u3)/Vt) - 1);
  C2*der(u3) = i4;
  0 = if open then i5 else u3 - u4;
  u4 = i5*RL;
  when sample(0,T) then
    open = not pre(open);
  end when;
  annotation (experiment(StopTime=0.5));
end circuitol;

```

Código 1.3: Ejercicio 3.2: modelo atómico.

```

encapsulated package LibElectrica

import SI = Modelica.SIunits;
import Modelica.Constants;

package Interfaces

    connector Pin
        SI.Voltage u;
        flow SI.Current i;
    end Pin;

    connector SignalCntrl
        Boolean s;
    end SignalCntrl;

    partial model DosPines
        Pin p, n;
    protected
        SI.Voltage u "Voltaje entre pines (= p.u - n.u)";
    equation
        u = p.u - n.u;
    end DosPines;

    partial model UnPuerto
        extends DosPines;
        parameter Boolean activo=false;
    protected
        SI.Current i "Corriente a traves del componente";
    equation
        if (activo) then
            i = n.i;
        else
            i = p.i;
        end if;
        p.i = -n.i;
    end UnPuerto;

end Interfaces;

```

Código 1.4: Librería eléctrica y modelo (1/3).

```

package Componentes

  model Resistencia "Resistencia ideal"
    extends Interfaces.UnPuerto;
    parameter SI.Resistance R "Resistencia";
  equation
    u = R*i;
  end Resistencia;

  model Diodo "Diodo ideal"
    extends Interfaces.UnPuerto;
    parameter SI.Current Is=1e-9 "Corriente saturacion";
    parameter SI.Voltage Vt=0.025 "Tension termica";
  equation
    i = Is*(exp(u/Vt) - 1);
  end Diodo;

  model Interruptor "Interruptor ideal"
    extends Interfaces.UnPuerto;
    Interfaces.SignalCntrl signalSw;
  equation
    0 = if signalSw.s then i else u;
  end Interruptor;

  model Condensador "Condensador ideal"
    extends Interfaces.UnPuerto;
    parameter SI.Capacitance C "Capacidad";
  equation
    C*der(u) = i;
  end Condensador;

  model VfuelleAC "Fuente tension AC"
    extends Interfaces.UnPuerto(activo=true);
    parameter SI.Voltage U0;
    parameter SI.Frequency frec;
    parameter SI.Angle phi;
  protected
    parameter SI.AngularFrequency w=2*Constants.pi*frec;
  equation
    u = U0*sin(w*time + phi);
  end VfuelleAC;

  model Tierra "Tierra"
    Interfaces.Pin p;
  equation
    p.u = 0;
  end Tierra;

end Componentes;

```

Código 1.5: Librería eléctrica y modelo (2/3).

```

package GeneradoresSenal

  model OndaCuadrada
    Interfaces.SignalCntrl sgn;
    parameter SI.Time T = 1;
    parameter SI.Time tInicial = 0;
    parameter Boolean sInicial = true;
  protected
    Boolean open(start=sInicial, fixed=true);
  equation
    when sample(tInicial,T) then
      open = not pre(open);
    end when;
    sgn.s = open;
  end OndaCuadrada;

end GeneradoresSenal;

package Ejemplos

  model circuito
    Componentes.VfuenteAC V(U0=12, frec=60, phi=0);
    Componentes.Condensador C1(C=1e-5);
    Componentes.Condensador C2(C=0.01);
    Componentes.Diodo D1(Is=1e-9, Vt=0.025);
    Componentes.Interruptor Sw;
    Componentes.Tierra ground;
    Componentes.Resistencia R1(R=20);
    Componentes.Resistencia RL(R=1000);
    GeneradoresSenal.OndaCuadrada CntrlSw(T=0.05);
  equation
    connect(V.p, R1.p);
    connect(R1.n, C1.p);
    connect(R1.n, D1.p);
    connect(D1.n, C2.p);
    connect(D1.n, Sw.p);
    connect(Sw.n, RL.p);
    connect(ground.p, V.n);
    connect(ground.p, C1.n);
    connect(ground.p, C2.n);
    connect(ground.p, RL.n);
    connect(CntrlSw.sgn, Sw.signalSw);
    annotation (experiment(StopTime=0.5));
  end circuito;

end Ejemplos;

end LibElectrica;

```

Código 1.6: Librería eléctrica y modelo (3/3).

EJERCICIO 4

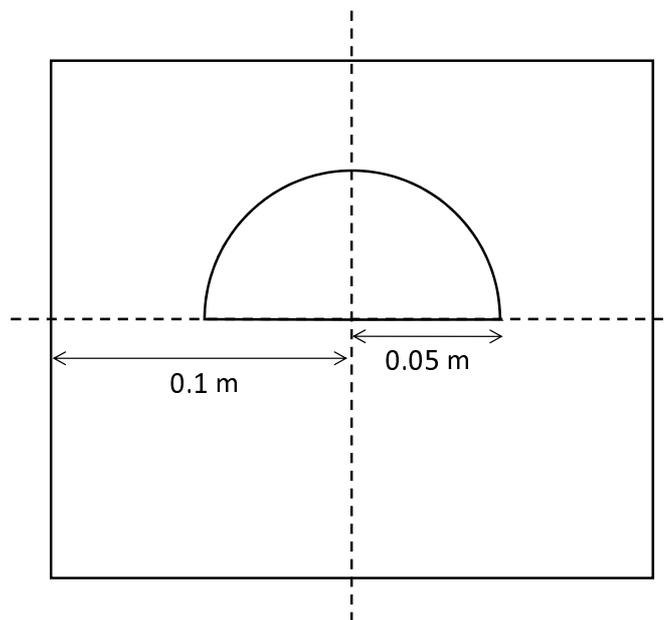
Escriba un *script* en FlexPDE para obtener la temperatura y los vectores correspondientes al flujo de calor del sistema descrito a continuación.

El sistema está formado por una varilla de sección semicircular de acero inoxidable rodeada por un bloque de granito. El bloque de granito tiene una conductividad térmica (κ_1) de 3.5 en unidades del sistema internacional. El acero inoxidable tiene una conductividad térmica (κ_2) de 45 en unidades del sistema internacional.

La superficie exterior del bloque de granito se mantiene a una temperatura constante de 273 K.

A través de la varilla de acero circula una corriente eléctrica uniforme que genera una potencia calorífica por unidad de volumen (*calor*) igual a 10^6 en unidades del sistema internacional.

El sistema se trata como un problema bidimensional. En la figura se muestra una sección transversal, en la cual el granito tiene una forma cuadrada de lado 0.2 m y la varilla de acero tiene un radio (r_0) de 0.05 m. Las dos líneas de trazo discontinuo se cortan en el centro de la sección cuadrada de granito.



Este fenómeno se describe por la ecuación diferencial siguiente, donde *temp* es la temperatura.

$$\nabla \cdot (-\kappa \cdot \nabla temp) = calor$$

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* se muestra a continuación. El mapa del contorno de la temperatura y los vectores correspondientes al flujo de calor se muestran respectivamente en las Figuras 1.7 y 1.8.

```

TITLE 'Calor'
COORDINATES cartesian2
VARIABLES
    temp
SELECT
    errlim=3e-4 ngrid=1 spectral_colors
DEFINITIONS
    Lx=0.1 Ly=0.1 r0=0.05
    k heat
    fluxd_x=-k*dx(temp) fluxd_y=-k*dy(temp)
    fluxd=vector(fluxd_x,fluxd_y)
EQUATIONS
    div(fluxd)=heat
BOUNDARIES
    REGION 'dominio' k=3.5 heat=0
        START 'exterior' (-Lx,-Ly) value(temp)=273
        LINE TO (Lx,-Ly) to (Lx,Ly) TO (-Lx,Ly) to FINISH
    REGION 'acero' k=45 heat=1e6
        START 'interior' (-r0,0) LINE TO (r0,0) ARC TO (0,r0) TO FINISH
PLOTS
    CONTOUR(temp)
    VECTOR(fluxd)
END

```

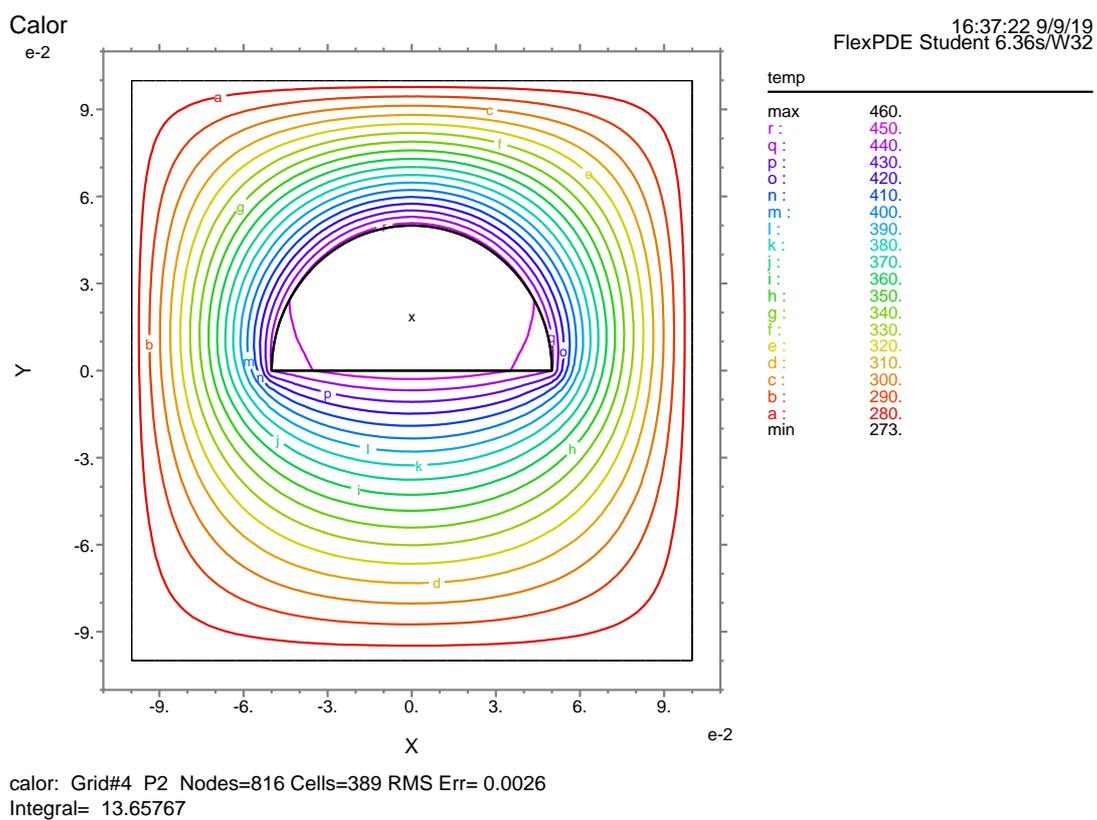


Figura 1.7: Temperatura.

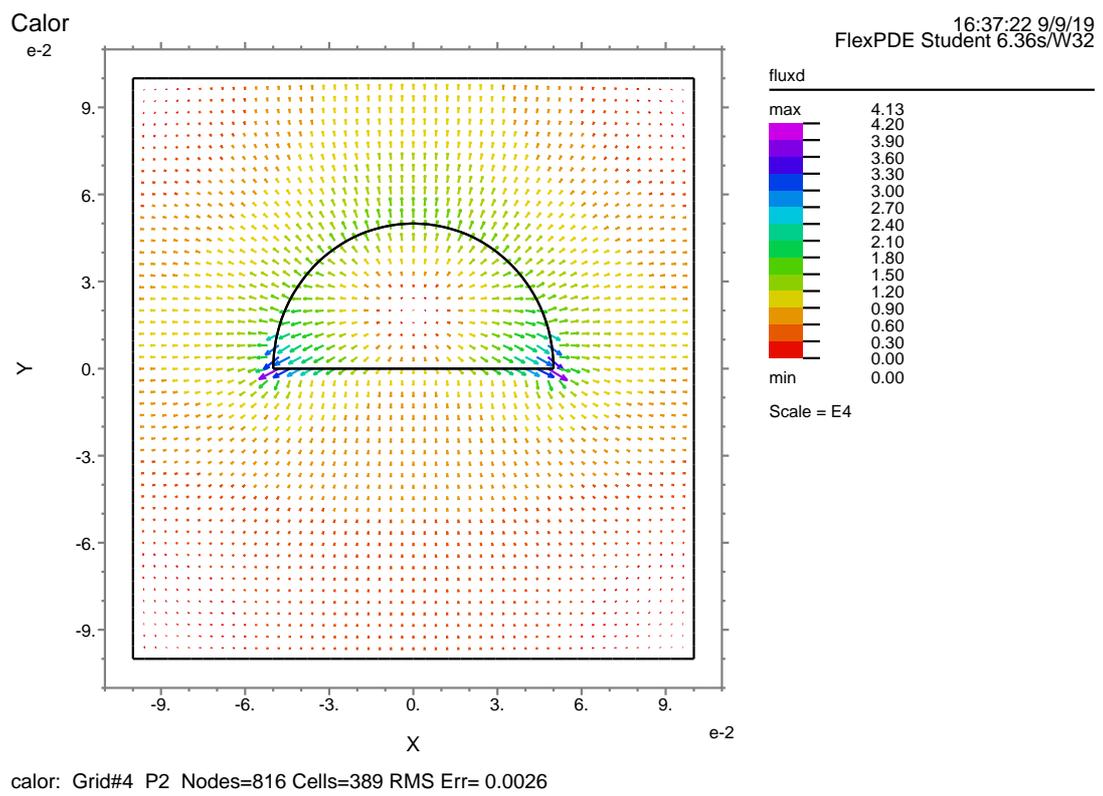


Figura 1.8: Flujo de calor.