

MÉTODOS DE SIMULACIÓN Y MODELADO

Solución al Trabajo Práctico - Enero de 2025

Ejercicio 1

Lea el artículo citado a continuación, que puede descargar de la página web de la asignatura, y conteste **detalladamente** 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. Explique detalladamente la Figura 1 basándose en el artículo.

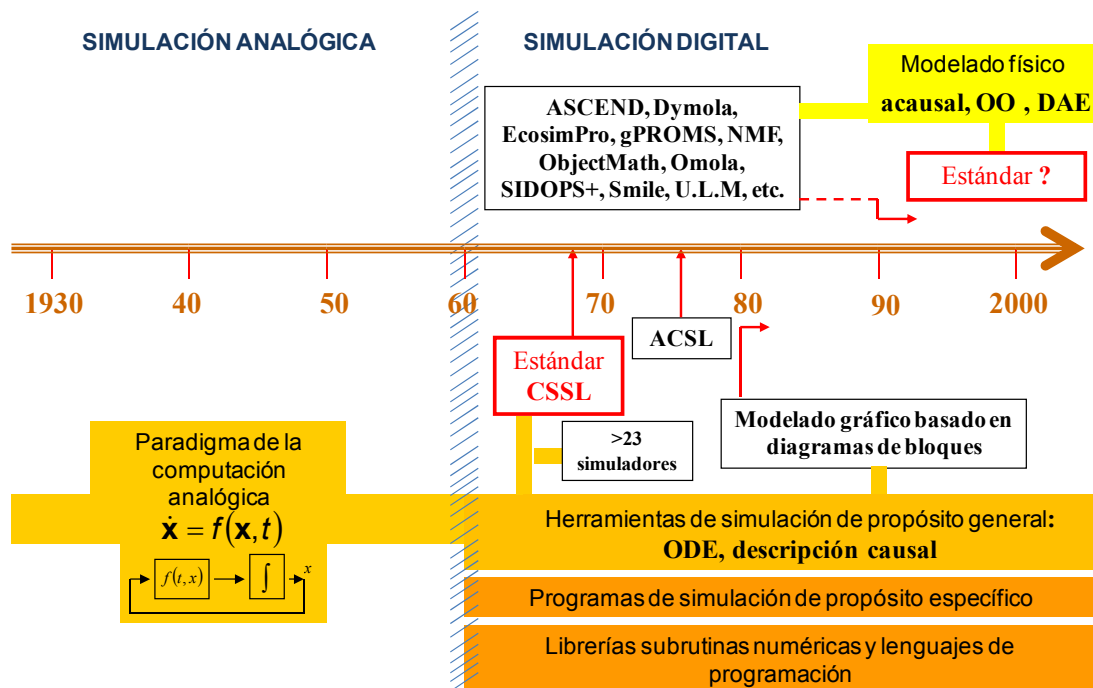


Figura 1: Evolución del modelado y simulación de tiempo continuo.

Solución al Ejercicio 1

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

Ejercicio 2

En la Figura 2 se muestra el modelo circuital de una batería de ion litio tipo LFP (litio-ferrofosfato), conectado a un cargador de baterías. El modelo de la batería describe el comportamiento de la misma durante el proceso de carga que lleva a cabo este cargador en concreto.

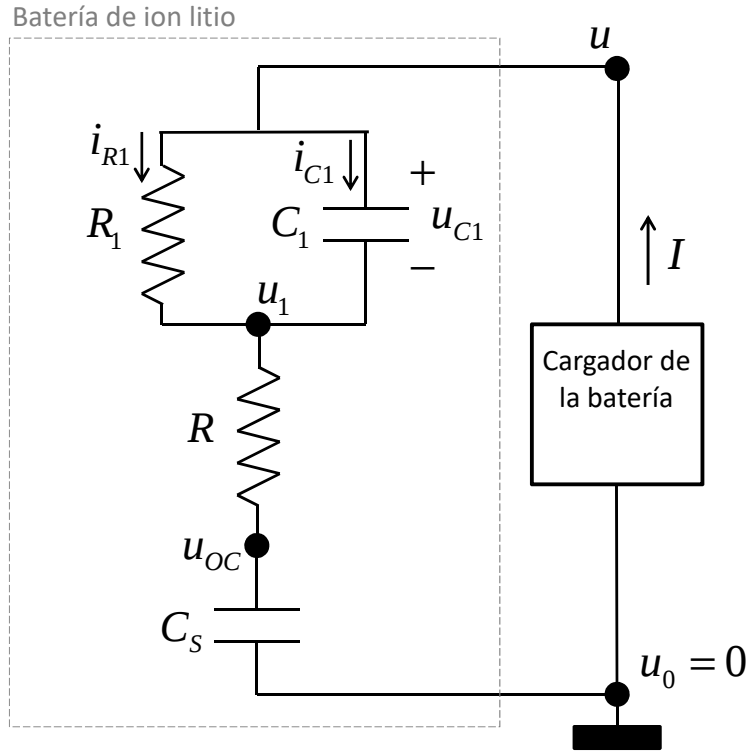


Figura 2: Modelo de una batería de ion litio y su cargador.

En el circuito se ha asignado nombre a los voltajes en los nodos y a las corrientes que circulan a través de los componentes. Se ha escogido el nodo u_0 como referencia para el voltaje: $u_0 = 0$.

El modelo de circuito equivalente de la batería está compuesto por dos resistencias de valor constante $R = 0.1212 \, \Omega$ y $R_1 = 1.1017 \, \Omega$, y dos condensadores de capacidad constante $C_S = 122.1975 \, \text{F}$ y $C_1 = 13437 \, \text{F}$. En la inicialización de la simulación, el voltaje entre los bornes del condensador C_S es $u_{OC} = 10.5 \, \text{V}$ y el voltaje entre los bornes del condensador C_1 es $u_{C1} = 0 \, \text{V}$.

El cargador puede encontrarse en cada instante de la simulación en uno de los tres modos de operación siguientes: {"Carga a CC", "Carga a VC", "Carga Completa"}. En la inicialización de la simulación, el cargador está en el modo "Carga a CC".

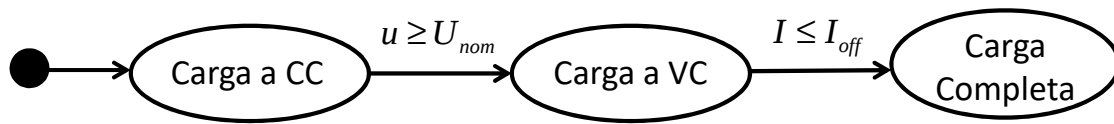


Figura 3: Transición entre los modos del cargador.

A continuación se describen las condiciones de transición entre los modos (mostradas en la Figura 3) y el comportamiento del cargador en cada modo de operación.

- Modo “Carga a CC” (Corriente Constante). Mientras el cargador está en este modo, el cargador impone que la corriente (I) valga un valor constante igual a $I_{1C} = 2.3$ A. Cuando se satisface la condición de que el voltaje de la batería es mayor o igual que $U_{nom} = 13.2$ V, se produce la transición al modo “Carga a VC”.
- Modo “Carga a VC” (Voltaje Constante). Mientras el cargador está en este modo, el cargador impone que el voltaje entre los bornes de la batería valga $u = U_{nom}$. Cuando se satisface la condición de que la corriente entrante a la batería (I) es menor o igual que $I_{off} = 0.02$ A, se produce la transición al modo “Carga Completa”.
- Modo “Carga Completa”. Mientras el cargador está en este modo, el cargador impone que la corriente valga $I = 0$. El cargador permanece indefinidamente en este modo de operación.

El modelo debe describir la evolución de los voltajes y las corrientes, así como de tres variables Booleanas llamadas *CargaCC*, *CargaVC* y *CargaCompleta* que describen el modo de operación del cargador: su valor es *true* mientras el cargador se encuentra respectivamente en el modo de operación “Carga a CC”, “Carga a VC” y “Carga Completa”, y *false* en caso contrario.

Realice las tareas siguientes:

1. Escriba las ecuaciones del modelo.
2. Asigne la causalidad computacional. Explique detalladamente el procedimiento seguido para ello. Si el modelo contiene un lazo algebraico lineal, resuélvalo “a mano”.
3. Dibuje 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 alcance el valor 300 s.

4. Programe el algoritmo anterior en lenguaje R y ejecute la simulación. Explique detalladamente cómo ha escogido el tamaño del paso de integración. Represente gráficamente frente al tiempo las variables u , I , $CargaCC$, $CargaVC$ y $CargaCompleta$.

Solución al Ejercicio 2

Las ecuaciones del modelo de la batería se muestran a continuación. La variable t representa el tiempo medido en segundos.

$$\begin{aligned}
 u &= u_{OC} + I \cdot R + u_{C1} \\
 u_{C1} &= i_{R1} \cdot R_1 \\
 C_1 \cdot \frac{du_{C1}}{dt} &= i_{C1} \\
 u_1 - u_{OC} &= I \cdot R \\
 C_S \cdot \frac{du_{OC}}{dt} &= I \\
 I &= i_{R1} + i_{C1}
 \end{aligned}$$

Las ecuaciones que describen el cargador de la batería son las mostradas a continuación. Se emplea la función $\text{pre}()$ en el sentido que tiene en el lenguaje Modelica.

$$\begin{aligned}
 0 &= \begin{cases} I - I_{1C} & \text{si } CargaCC \\ u - U_{nom} & \text{si } CargaVC \\ I & \text{si } CargaCompleta \end{cases} \\
 CargaCC &= \text{pre}(CargaCC) \text{ and not } u \geq U_{nom} \\
 CargaVC &= \text{pre}(CargaCC) \text{ and } u \geq U_{nom} \text{ or} \\
 &\quad \text{pre}(CargaVC) \text{ and not } I \leq I_{off} \\
 CargaCompleta &= \text{pre}(CargaVC) \text{ and } I \leq I_{off} \text{ or} \\
 &\quad \text{pre}(CargaCompleta)
 \end{aligned}$$

Los parámetros del modelo tienen los valores siguientes: $R = 0.1212 \, \Omega$, $R_1 = 1.1017 \, \Omega$, $C_S = 122.1975 \, \text{F}$, $C_1 = 13437 \, \text{F}$, $I_{1C} = 2.3 \, \text{A}$, $U_{nom} = 13.2 \, \text{V}$, $I_{off} = 0.02 \, \text{A}$.

Para realizar la partición, asumimos que las variables de tiempo discreto $CargaCC$, $CargaVC$ y $CargaCompleta$ son conocidas durante la resolución del problema de tiempo continuo, ya que su valor solo cambia en la ejecución de los eventos.

Asumiendo que las variables que aparecen derivadas (u_{C1} , u_{OC}) pueden ser seleccionadas como variables de estado de tiempo continuo, las variables del modelo pueden clasificarse de la forma siguiente:

- Parámetros y constantes: $R, R_1, C_S, C_1, I_{1C}, U_{nom}, I_{off}$
- Variables de estado: u_{C1}, u_{OC}
- Variables algebraicas: $u, u_1, I, i_{R1}, i_{C1}$

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

$$\frac{du_{C1}}{dt} \rightarrow deru_{C1}, \quad \frac{du_{OC}}{dt} \rightarrow deru_{OC}$$

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

$$u_{C1}, u_{OC} \quad \text{Variables de estado} \quad (1.1)$$

$$u = u_{OC} + I \cdot R + u_{C1} \quad (1.1)$$

$$u_{C1} = i_{R1} \cdot R_1 \quad (1.2)$$

$$C_1 \cdot deru_{C1} = i_{C1} \quad (1.3)$$

$$u_1 - u_{OC} = I \cdot R \quad (1.4)$$

$$C_S \cdot deru_{OC} = I \quad (1.5)$$

$$I = i_{R1} + i_{C1} \quad (1.6)$$

$$0 = \begin{cases} I - I_{1C} & \text{si } CargaCC \\ u - U_{nom} & \text{si } CargaVC \\ I & \text{si } CargaCompleta \end{cases} \quad (1.7)$$

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). A continuación se muestra dicha clasificación.

- Conocidas: t
 $R, R_1, C_S, C_1, I_{1C}, U_{nom}, I_{off}$
 u_{C1}, u_{OC}
- Desconocidas: $u, u_1, I, i_{R1}, i_{C1}$
 $deru_{C1}, deru_{OC}$

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 7 ecuaciones y 7 incógnitas.

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

$$\begin{aligned}
 u &\rightarrow \text{Ec. (1.1)} \\
 i_{R1} &\rightarrow \text{Ec. (1.2)} \\
 deru_{C1} &\rightarrow \text{Ec. (1.3)} \\
 u_1 &\rightarrow \text{Ec. (1.4)} \\
 deru_{OC} &\rightarrow \text{Ec. (1.5)} \\
 i_{C1} &\rightarrow \text{Ec. (1.6)} \\
 I &\rightarrow \text{Ec. (1.7)}
 \end{aligned}$$

Aplicando el algoritmo de la partición, se obtiene el siguiente modelo ordenado (existen otras posibles formas de ordenar el modelo que son igualmente válidas), con la causalidad computacional señalada. Obsérvese que las variables u e I deben calcularse resolviendo un lazo algebraico lineal.

$$\begin{array}{rcl}
 & & u_{C1}, u_{OC} \quad \text{Variables de estado} \\
 u & = & u_{OC} + I \cdot R + u_{C1} \\
 0 & = & \left\{ \begin{array}{ll} I - I_{1C} & \text{si } CargaCC \\ u - U_{nom} & \text{si } CargaVC \\ I & \text{si } CargaCompleta \end{array} \right. \quad [u], [I] \\
 & & u_{C1} = [i_{R1}] \cdot R_1 \\
 & & I = i_{R1} + [i_{C1}] \\
 & & [u_1] - u_{OC} = I \cdot R \\
 & & C_1 \cdot [deru_{C1}] = i_{C1} \\
 & & C_S \cdot [deru_{OC}] = I
 \end{array}$$

Manipulando simbólicamente las ecuaciones, se obtiene el modelo ordenado y resuelto (existen otras posibles ordenaciones de las ecuaciones que son igualmente válidas).

$$\begin{aligned}
u_{C1}, u_{OC} & \quad \text{Variables de estado} \\
[I] &= \begin{cases} I_{1C} & \text{si } CargaCC \\ \frac{U_{nom} - u_{OC} - u_{C1}}{R} & \text{si } CargaVC \\ 0 & \text{si } CargaCompleta \end{cases} \\
[u] &= u_{OC} + I \cdot R + u_{C1} \\
[i_{R1}] &= \frac{u_{C1}}{R_1} \\
[i_{C1}] &= I - i_{R1} \\
[u_1] &= u_{OC} + I \cdot R \\
[deru_{C1}] &= \frac{i_{C1}}{C_1} \\
[deru_{OC}] &= \frac{I}{C_S}
\end{aligned}$$

En la Figura 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 300 s.

Se ha seleccionado un tamaño del paso igual a 0.01 s. Para ello, se ha repetido la simulación empleando diferentes tamaños del paso, encontrándose que el error cometido escogiendo 0.01 s es admisible para el propósito de este estudio.

Se muestra también el código R que implementa el algoritmo mostrado en la Figura 4, con un intervalo de comunicación igual al tamaño del paso de integración, esto es, 0.01 s. En las Figuras 5 y 6 se muestran las gráficas obtenida al ejecutar el código R.

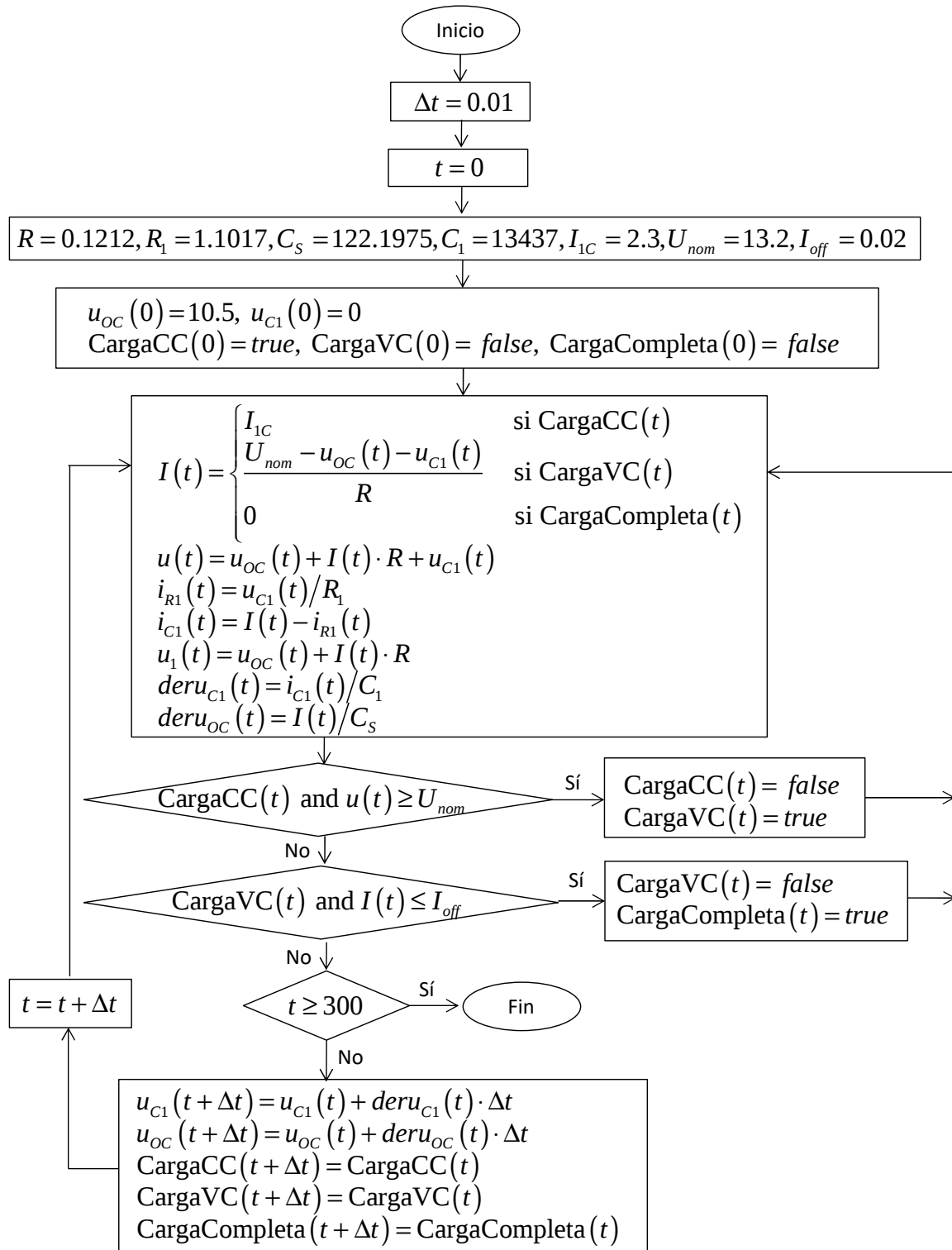


Figura 4: Diagrama de flujo de la simulación del modelo.

Código R del algoritmo de la simulación

```

1 rm( list = ls() )
2 tiempoFin    <- 300      # Valor final del tiempo
3 incTiempo    <- 0.01     # Paso de integracion
4
5 # Parametros
6 R            <- 0.1212
7 R_1          <- 1.1017
8 C_S          <- 122.1975
9 C_1          <- 13437
10 I_1C         <- 2.3
11 U_nom        <- 13.2
12 I_off        <- 0.02
13
14 # Vector que contiene los instantes de tiempo simulado
15 tiempo <- seq(0, tiempoFin, by = incTiempo)
16
17 # Inicializacion vectores que almacenan las variables del modelo
18 nPoints      <- length(tiempo)
19 u_OC          <- numeric(nPoints)
20 u_C1          <- numeric(nPoints)
21 I             <- numeric(nPoints)
22 I             <- numeric(nPoints)
23 u             <- numeric(nPoints)
24 i_R1          <- numeric(nPoints)
25 i_C1          <- numeric(nPoints)
26 u_1           <- numeric(nPoints)
27 deru_C1       <- numeric(nPoints)
28 deru_OC       <- numeric(nPoints)
29 CargaCC       <- logical(nPoints)
30 CargaVC       <- logical(nPoints)
31 CargaCompleta <- logical(nPoints)
32
33 # Valor inicial del estado de tiempo continuo
34 u_OC[1] <- 10.5
35 u_C1[1] <- 0
36
37 # Valor inicial del modo del cargador
38 CargaCC[1] <- TRUE
39 CargaVC[1] <- FALSE
40 CargaCompleta[1] <- FALSE
41
42 # Bucle de la simulacion
43 for ( nStep in c(1:nPoints) ) {
44
45     repeat {
46         # Calculo variables algebraicas en t
47         if (CargaCC[nStep]) {
48             I[nStep] <- I_1C
49         } else if (CargaVC[nStep]) {
50             I[nStep] <- (U_nom - u_OC[nStep] - u_C1[nStep]) / R
51         } else {

```

```

52     I[nStep] <- 0
53   }
54   u[nStep] <- u_OC[nStep] + I[nStep] * R + u_C1[nStep]
55   i_R1[nStep] <- u_C1[nStep] / R_1
56   i_C1[nStep] <- u_C1[nStep] / R_1
57   u_1[nStep] <- u_OC[nStep] + I[nStep] * R
58   deru_C1[nStep] <- i_C1[nStep] / C_1
59   deru_OC[nStep] <- I[nStep] / C_S
60
61   # Eventos
62   EVENTO <- FALSE
63   if ( CargaCC[nStep] && u[nStep] >= U_nom ) {
64     CargaCC[nStep] <- FALSE
65     CargaVC[nStep] <- TRUE
66     EVENTO <- TRUE
67   } else if ( CargaVC[nStep] && I[nStep] <= I_off ) {
68     CargaVC[nStep] <- FALSE
69     CargaCompleta[nStep] <- TRUE
70     EVENTO <- TRUE
71   }
72   if ( !EVENTO ) {break}
73 } # repeat
74
75 # Calculo en el siguiente instante de la variable de estado
76 if ( nStep < nPoints ) {
77   u_C1[nStep+1] <- u_C1[nStep] + incTiempo * deru_C1[nStep]
78   u_OC[nStep+1] <- u_OC[nStep] + incTiempo * deru_OC[nStep]
79 }
80 if ( nStep < nPoints ) {
81   CargaCC[nStep+1] <- CargaCC[nStep]
82   CargaVC[nStep+1] <- CargaVC[nStep]
83   CargaCompleta[nStep+1] <- CargaCompleta[nStep]
84 }
85
86 } # fin del bucle de la simulacion
87
88 # Representacion grafica
89 dev.new()
90 plot(tiempo, u, xlab = "t [s]", ylab = "u [V]",
91      type="l", col="blue", lwd=1.5, ylim=c(10.5,13.5),
92      panel.first = abline(v = seq(0,tiempoFin,by=10),
93                           h = seq(10,14,by=0.1),
94                           lwd = 0.5, lty = 3, col="grey") )
95 dev.new()
96 plot(tiempo, I, xlab = "t [s]", ylab = "I [A]",
97      type="l", col="blue", lwd=1.5,
98      panel.first = abline(v = seq(0,tiempoFin,by=10),
99                           h = seq(0,2.6,by=0.1),
100                          lwd = 0.5, lty = 3, col="grey") )
101 dev.new()
102 plot(tiempo, CargaCC, xlab = "t [s]", ylab = "CargaCC",
103      type="l", col="blue", lwd=1.5,
104      panel.first = abline(v = seq(0,tiempoFin,by=10),
105                          h = seq(0,1),

```

```

106         lwd = 0.5, lty = 3, col="grey") )
107 dev.new()
108 plot(tiempo, CargaVC, xlab = "t [s]", ylab = "CargaCV",
109      type="l", col="blue", lwd=1.5,
110      panel.first = abline(v = seq(0,tiempoFin,by=10),
111                          h = seq(0,1),
112                          lwd = 0.5, lty = 3, col="grey") )
113 dev.new()
114 plot(tiempo, CargaCompleta, xlab = "t [s]", ylab = "CargaCompleta",
115      type="l", col="blue", lwd=1.5,
116      panel.first = abline(v = seq(0,tiempoFin,by=10),
117                          h = seq(0,1),
118                          lwd = 0.5, lty = 3, col="grey") )

```

Código 1: Programación en R del algoritmo de la simulación.

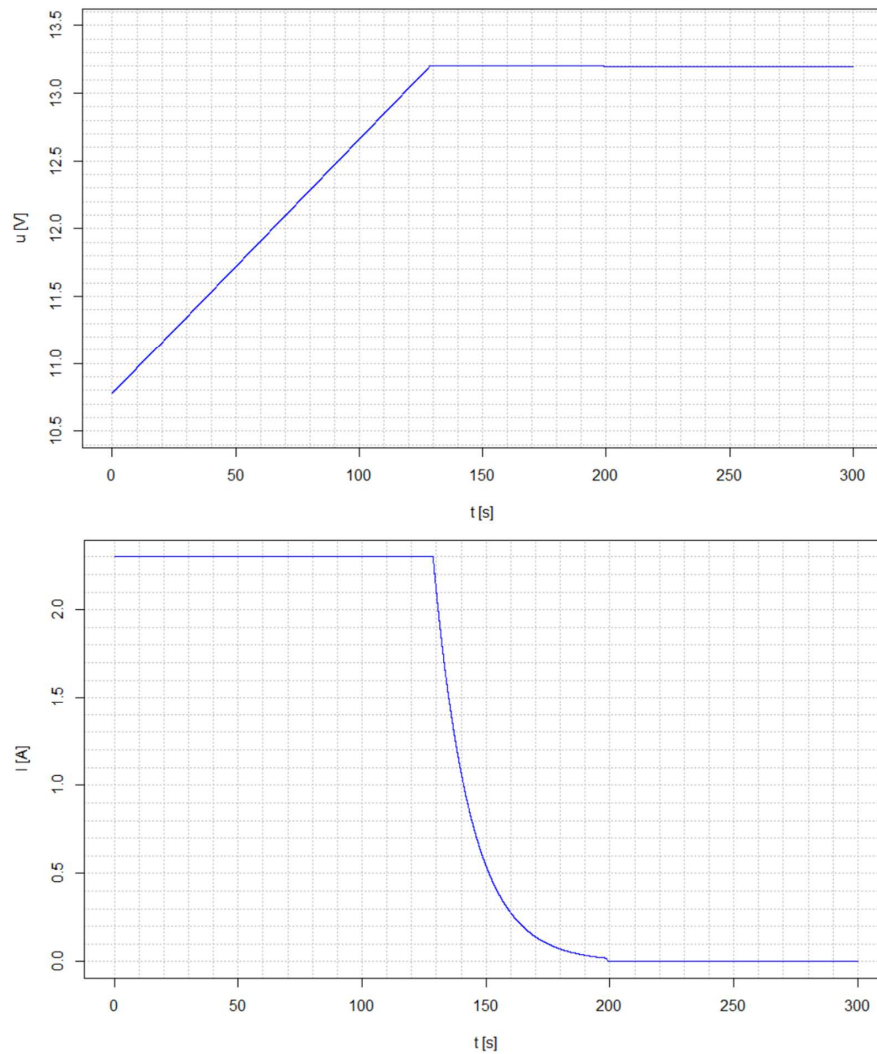


Figura 5: Resultado de la ejecución del código R.

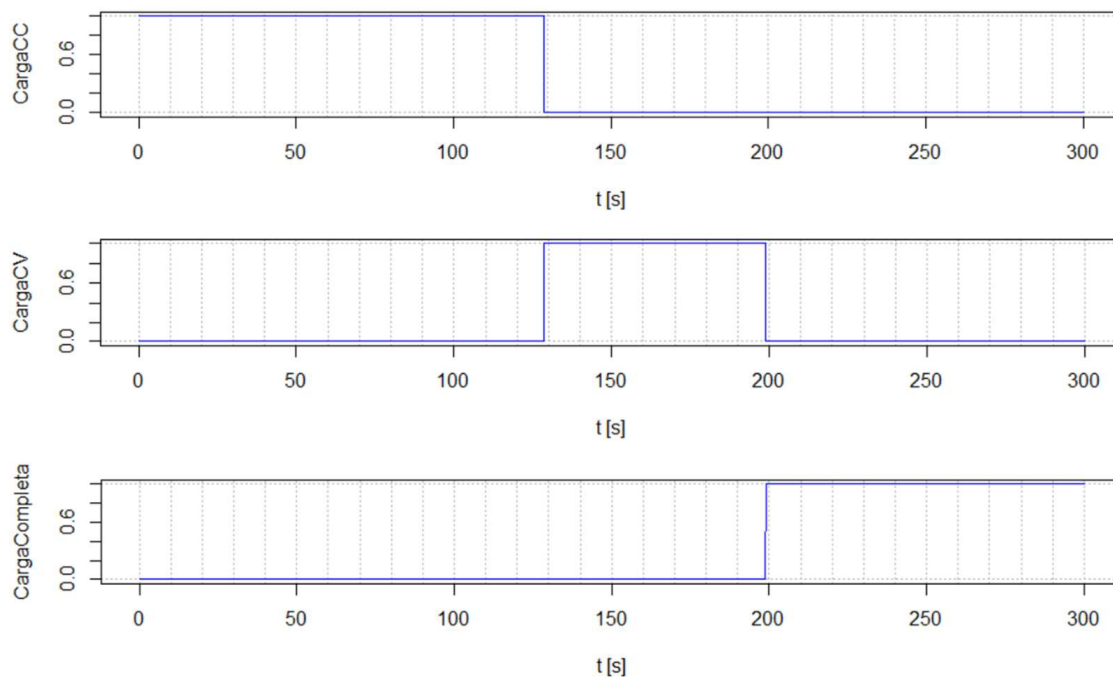


Figura 6: Resultado de la ejecución del código R.

Ejercicio 3

Describa en lenguaje Modelica el sistema del ejercicio anterior, de las dos maneras siguientes:

1. Como un modelo atómico, que viene descrito por las ecuaciones que usted ha planteado al contestar a la pregunta anterior.
2. Programe una librería en Modelica que contenga los componentes necesarios para componer el sistema descrito en el Ejercicio 2. La librería debe incluir los modelos de los componentes electrónicos (resistencia, condensador y tierra) y del cargador. A continuación, defina el sistema descrito en el Ejercicio 2 como un modelo compuesto, instanciando y conectando componentes de la librería que ha programado.

Asigne a los parámetros y a las condiciones iniciales los valores indicados en el ejercicio anterior. Simule los dos modelos anteriores durante 300 s, y represente gráficamente frente al tiempo las variables u , I , $CargaCC$, $CargaVC$ y $CargaCompleta$.

Solución al Ejercicio 3

El modelo atómico en lenguaje Modelica se muestra en Código 2. El resultado de la simulación está representado en las Figura 7.

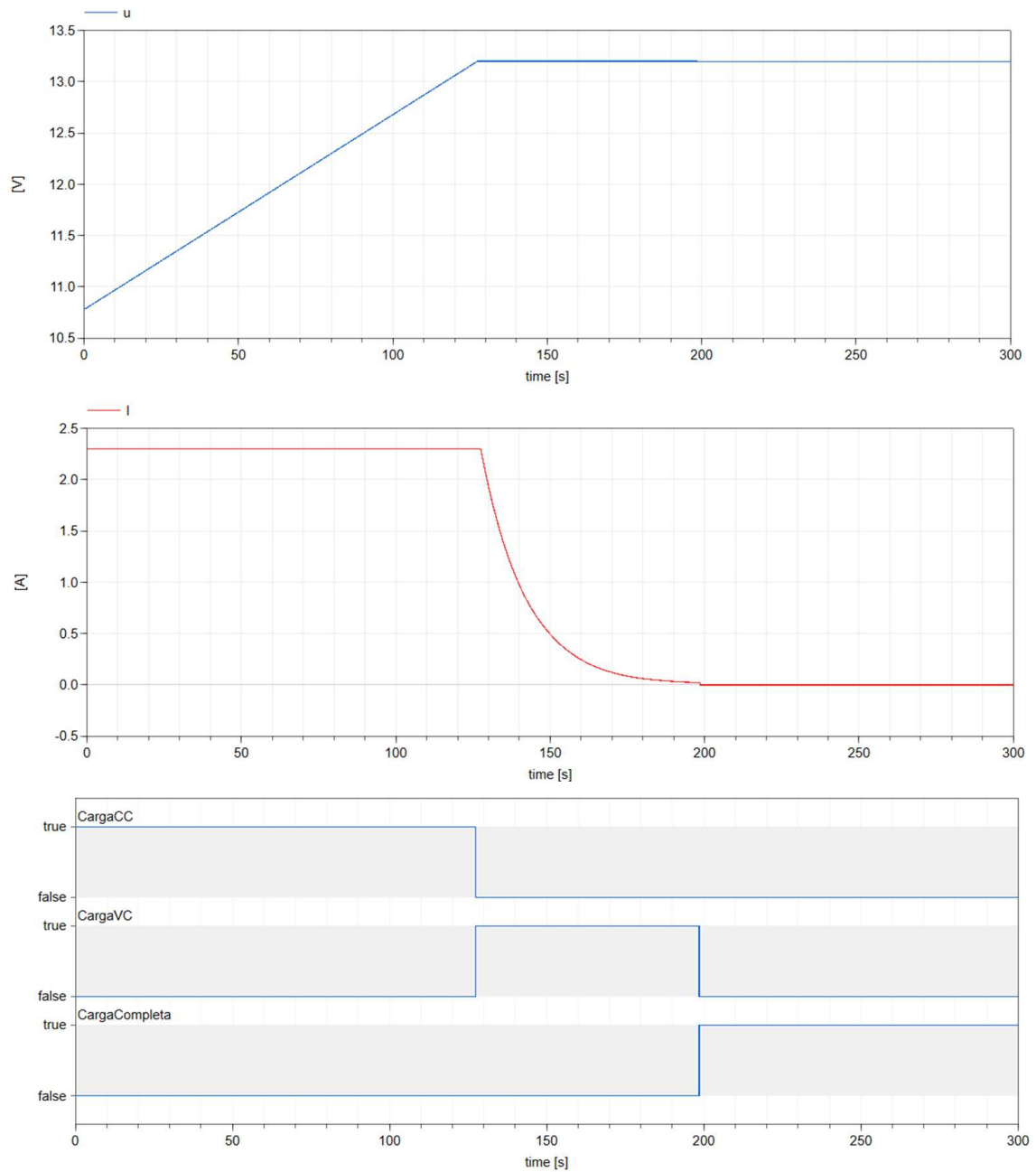
En el Código 3 se muestra una posible forma de modelar los componentes, organizándolos en una librería, y de componer el sistema mediante dichos componentes.

```

1 model CargaBateria
2   import      Modelica.Units.SI;
3
4   constant SI.Voltage unitVoltage = 1;
5   constant SI.Current unitCurrent = 1;
6
7   parameter SI.Resistance R = 0.1212, R1 = 1.1017;
8   parameter SI.Capacitance Cs = 122.1975, C1 = 13437;
9   parameter SI.Voltage U_nom = 13.2;
10  parameter SI.Current I_1C = 2.3, I_off = 0.02;
11
12  SI.Voltage u, u1;
13  SI.Current iR1, iC1, I;
14  SI.Voltage uC1 (start=0, fixed=true);
15  SI.Voltage uOC (start=10.5, fixed=true);
16
17  Boolean CargaCC (start=true, fixed=true);
18  Boolean CargaVC (start=false, fixed=true);
19  Boolean CargaCompleta(start=false, fixed=true);
20
21  equation
22    CargaCC = pre(CargaCC) and not u>=U_nom;
23    CargaVC = pre(CargaCC) and u>=U_nom or
24              pre(CargaVC) and not I<=I_off;
25    CargaCompleta = pre(CargaVC) and I<=I_off or
26                    pre(CargaCompleta);
27    u = uOC + I*R + uC1;
28    uC1 = iR1 * R1;
29    C1 * der(uC1) = iC1;
30    u1 - uOC = I * R;
31    Cs * der(uOC) = I;
32    0 = if CargaCompleta
33        then I/unitCurrent
34        else if CargaCC then (I - I_1C)/unitCurrent
35            else (u - U_nom)/unitVoltage;
36    I = iR1 + iC1;
37  annotation (experiment(StopTime=300));
38 end CargaBateria;

```

Código 2: Modelo atómico en Modelica.

**Figura 7:** Resultado de la ejecución del código Modelica.

```

1 package LibElectrica
2
3 import SI = Modelica.Units.SI;
4
5 package Interfaces
6
7     connector Pin
8         SI.Voltage u;
9         flow SI.Current i;
10    end Pin;
11
12 end Interfaces;
13
14 package Components
15
16     model Charger
17         Interfaces.Pin p, n;
18         constant SI.Voltage unitVoltage = 1;
19         constant SI.Current unitCurrent = 1;
20         parameter SI.Voltage U_nom;
21         parameter SI.Current I_1C, I_off;
22         SI.Voltage u;
23         SI.Current I;
24         Boolean CargaCC      (start=true, fixed=true);
25         Boolean CargaVC      (start=false, fixed=true);
26         Boolean CargaCompleta(start=false, fixed=true);
27     equation
28         u = p.u - n.u;
29         p.i = -n.i;
30         I = -p.i;
31         CargaCC      = pre(CargaCC) and not u>=U_nom;
32         CargaVC      = pre(CargaCC) and u>=U_nom or
33             pre(CargaVC) and not I<=I_off;
34         CargaCompleta = pre(CargaVC) and I<=I_off or
35             pre(CargaCompleta);
36         0 = if CargaCompleta
37             then I/unitCurrent
38             else if CargaCC then (I - I_1C)/unitCurrent
39                 else (u - U_nom)/unitVoltage;
40     end Charger;
41
42     model Resistor
43         Interfaces.Pin p, n;
44         parameter SI.Resistance R;
45     equation
46         p.i = -n.i;
47         p.u - n.u = R * p.i;
48     end Resistor;
49
50     model Capacitor
51         Interfaces.Pin p, n;
52         parameter SI.Capacitance C;
53         parameter SI.Voltage u0;

```

```

54     parameter Boolean      u0_fixed;
55     SI.Voltage u(start=u0, fixed=u0_fixed);
56   equation
57     u = p.u - n.u;
58     p.i = -n.i;
59     C * der(u) = p.i;
60   end Capacitor;
61
62   model Ground
63     Interfaces.Pin p;
64   equation
65     p.u = 0;
66   end Ground;
67
68 end Components;
69
70
71 package Circuits
72
73   model circ
74     Components.Charger charger(U_nom=13.2, I_1C=2.3, I_off=0.02);
75     Components.Resistor R1(R=1.1017), R(R=0.1212);
76     Components.Capacitor C1(C=13437, u0=0, u0_fixed=true),
77                          Cs(C=122.1975, u0=10.5, u0_fixed=true);
78     Components.Ground Ground;
79   equation
80     connect(charger.p, C1.p);
81     connect(charger.p, R1.p);
82     connect(C1.n, R.p);
83     connect(R1.n, R.p);
84     connect(R.n, Cs.p);
85     connect(Cs.n, Ground.p);
86     connect(charger.n, Ground.p);
87     annotation (experiment(StopTime=300));
88   end circ;
89
90 end Circuits;
91
92 end LibElectrica;

```

Código 3: Librería y modelo compuesto en Modelica.

Ejercicio 4

Se desea determinar la distribución estacionaria de temperatura (*temp*) de una lámina de un material que tiene forma rectangular, cuya dimensión es de 0.7 m por 0.3 m. La lámina tiene un orificio en la posición mostrada en la figura, cuyo radio es 0.1 m.

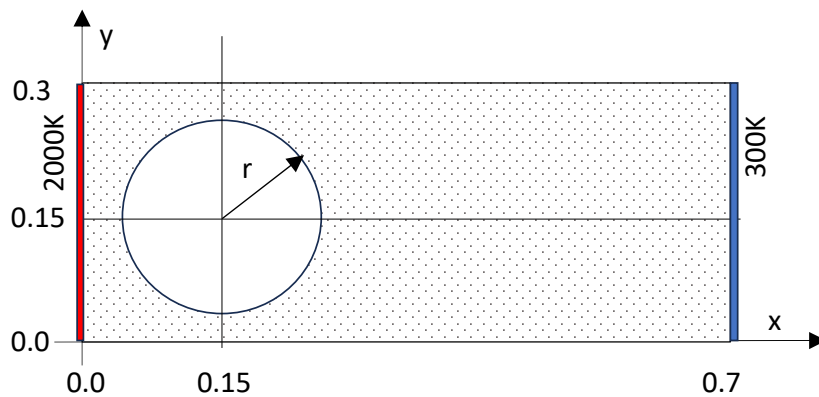


Figura 8: Sistema térmico.

La lámina está aislada y tiene una temperatura de 300 K y de 2000 K en los lados de longitud 0.3 m. La conductividad térmica de este material depende de la temperatura y en unidades del sistema internacional se calcula aplicando la fórmula $15000/temp$. Para acelerar la búsqueda de la solución, se ha de dar un valor inicial a la temperatura de 1300 K. Este problema se describe empleando la siguiente ecuación diferencial:

$$\text{div}(-k \cdot \text{grad}(temp)) = 0$$

Escriba un *script* empleando FlexPDE para obtener un mapa de contorno de la distribución de temperatura y un gráfico vectorial del flujo de calor a lo largo de la lámina.

Solución al Ejercicio 4

El *script* se muestra en el Código 4. El mapa de contorno de la distribución de temperatura se muestra en la Figura 9 y el gráfico vectorial del flujo de calor a lo largo de la lámina en la Figura 10.

```

1 TITLE 'Lamina'
2 COORDINATES cartesian2
3 VARIABLES
4   temp
5 SELECT      errlim=1e-1 ngrid=1 spectral_colors
6 DEFINITIONS
7   Lx=0.7 Ly=0.3 r=0.1 d=0.15
8   k=15000/temp
9   fluxdx=-k*dx(temp)
10  fluxdy=-k*dy(temp)
11  fluxd=vector(fluxdx,fluxdy)
12 INITIAL VALUES
13   temp=1300
14 EQUATIONS
15   div(-k*grad(temp))=0
16 BOUNDARIES      { Dominio}
17   REGION 1
18     START(0,0)   natural(temp)=0 LINE TO   (Lx,0)
19     VALUE(temp)=300 LINE TO   (Lx,Ly)
20     NATURAL(temp)=0 LINE TO   (0,Ly)
21     VALUE(temp)=2000 LINE TO FINISH
22     START (d, d-r) NATURAL(temp)=0
23     ARC (CENTER= d,d) angle=360 TO FINISH
24 PLOTS
25   CONTOUR(temp)
26   VECTOR(fluxd)
27 END

```

Código 4: Modelo descrito en FlexPDE.

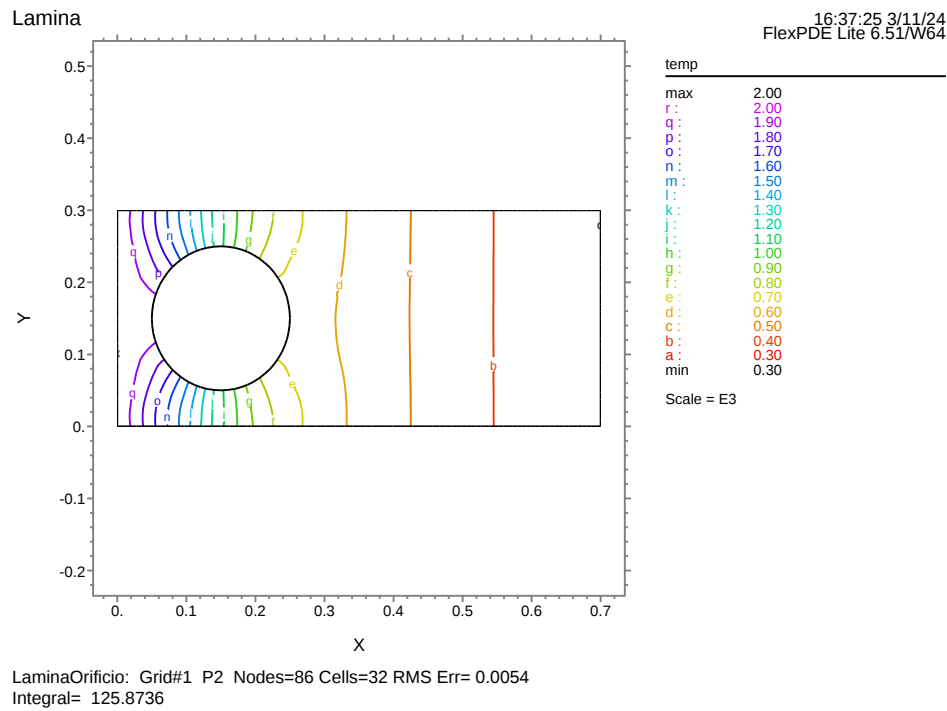


Figura 9: Mapa de contorno de la distribución de temperatura.

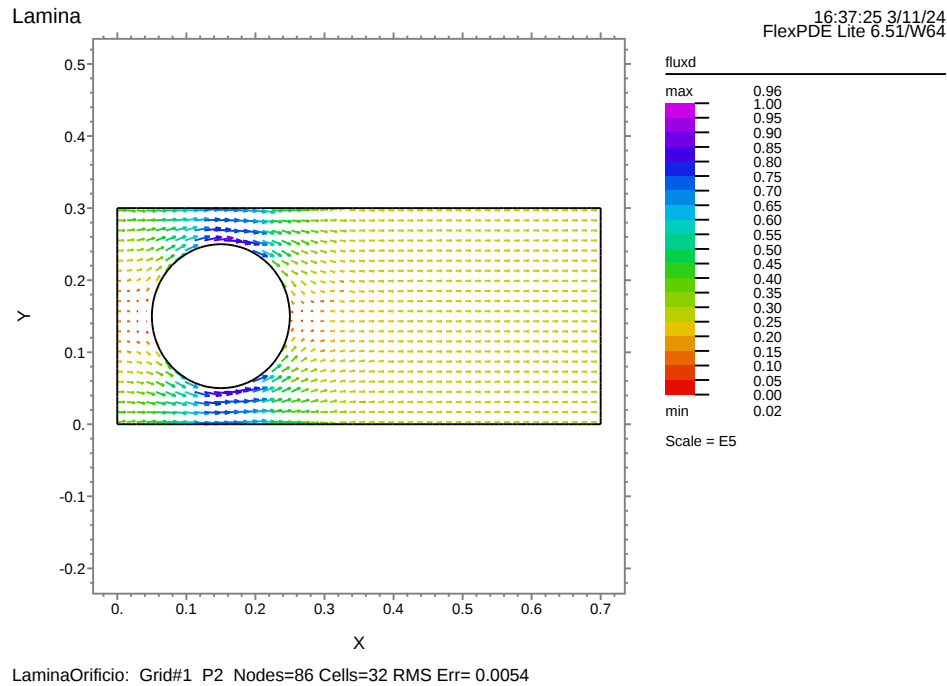


Figura 10: Gráfico vectorial del flujo de calor.