

MÉTODOS DE SIMULACIÓN Y MODELADO

Solución al Trabajo Práctico - Enero 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.

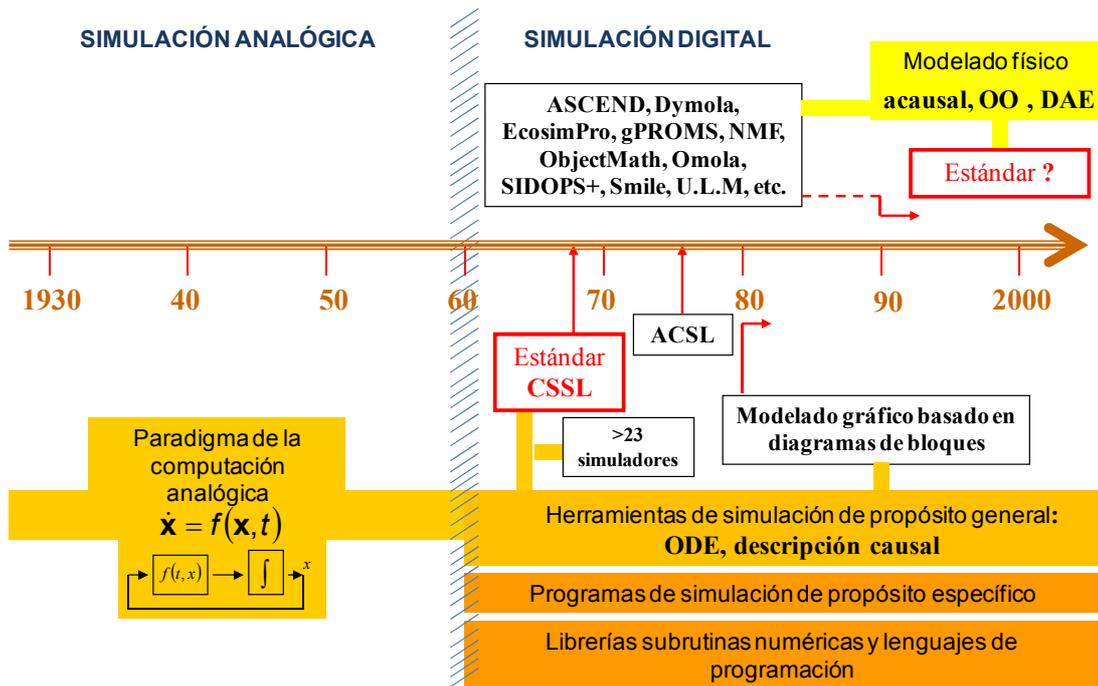


Figura 1.1: Evolución del modelado y simulación de tiempo continuo.

Solución al Ejercicio 1

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

EJERCICIO 2

Consideremos el sistema mostrado en la Figura 1.2, que consiste en dos objetos de masas constantes m_1 y m_2 , dos muelles ideales con coeficientes k_1 y k_2 , un amortiguador ideal con coeficiente b_1 , y una polea sin masa ni rozamiento. Las superficies señaladas como pared, techo y suelo se encuentran en reposo. El objeto de masa m_1 desliza sin rozamiento sobre el suelo y el de masa m_2 desliza con rozamiento sobre la pared.

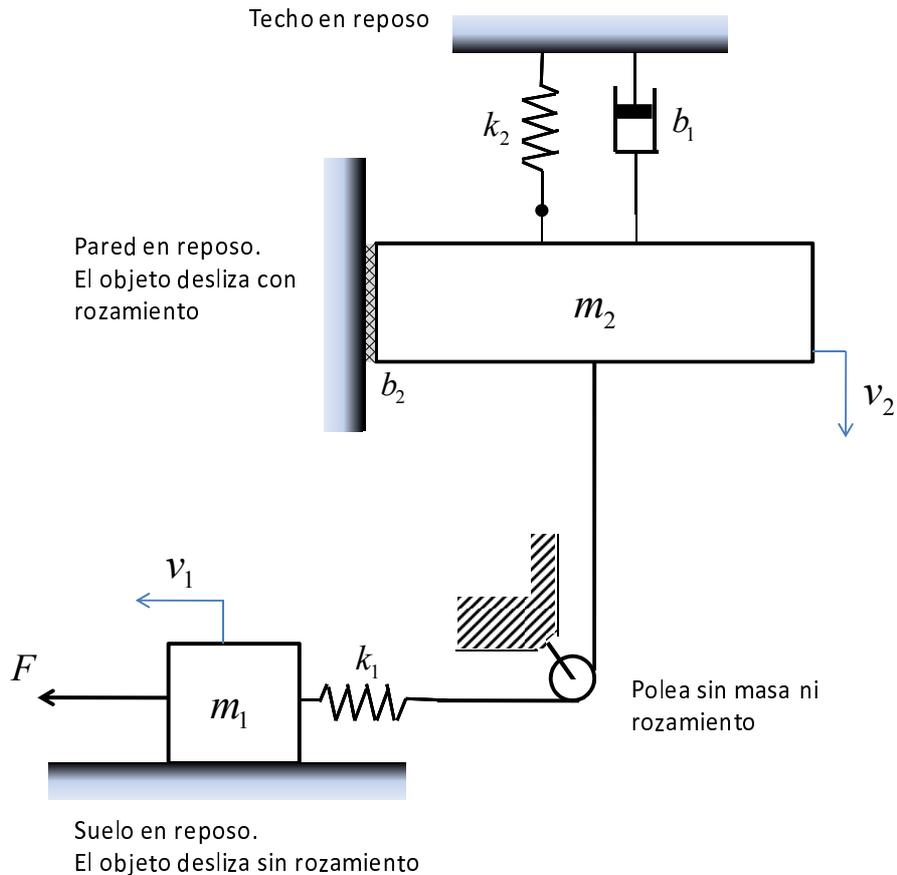


Figura 1.2: Diagrama del sistema mecánico.

El coeficiente del amortiguador (b_1), los coeficientes de los muelles (k_1 , k_2), el coeficiente de fricción (b_2) y las masas de los objetos (m_1 , m_2) son parámetros de valor conocido.

$$\begin{aligned} m_1 &= 200 \text{ kg} \\ k_1 &= 1.2e3 \text{ N/m} \\ b_1 &= 50 \text{ N}\cdot\text{s/m} \end{aligned}$$

$$\begin{aligned} m_2 &= 300 \text{ kg} \\ k_2 &= 3.5e3 \text{ N/m} \\ b_2 &= 125 \text{ N}\cdot\text{s/m} \end{aligned}$$

Suponga que la fuerza de rozamiento entre la pared y el objeto de masa m_2 es proporcional a la velocidad relativa entre ambas superficies. El coeficiente de proporcionalidad es b_2 .

La fuerza externa F que actúa sobre el objeto de masa m_1 vale cero durante los primeros 60 s. En el instante 60 s pasa instantáneamente a valer 250 N, manteniendo dicho valor hasta el instante 120 s, en el cual pasa instantáneamente a valer cero, manteniendo indefinidamente dicho valor.

1. Escriba las ecuaciones del modelo del sistema mecánico. El modelo debe describir la evolución de las elongaciones de los dos muelles y las velocidades de los dos objetos. El criterio de signos para las velocidades (positivas en el sentido indicado por las flechas) está señalado en la figura.
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 180 s.
4. Programe el algoritmo anterior en lenguaje R y ejecute la simulación. Represente gráficamente las elongaciones y velocidades frente al tiempo. Explique qué criterio ha seguido para escoger el tamaño del paso de integración.

Solución al Ejercicio 2

El sistema mecánico puede modelarse de la forma descrita a continuación. No se muestran las asignaciones de valor a las constantes y parámetros.

$$F = \begin{cases} 250 & \text{si } 60 < t < 120 \\ 0 & \text{en caso contrario} \end{cases} \quad (1.1)$$

$$m_1 \cdot \frac{dv_1}{dt} = F - k_1 \cdot e_1 \quad (1.2)$$

$$m_2 \cdot \frac{dv_2}{dt} = k_1 \cdot e_1 - k_2 \cdot e_2 - b_1 \cdot v_2 - b_2 \cdot v_2 + m_2 \cdot g \quad (1.3)$$

$$\frac{de_1}{dt} = v_1 - v_2 \quad (1.4)$$

$$\frac{de_2}{dt} = v_2 \quad (1.5)$$

La variable t representa el tiempo. Las variables v_1 y v_2 representan las velocidades de los objetos, y las variables e_1 y e_2 las elongaciones de los muelles, definidas como la diferencia entre la longitud del muelle y su longitud natural.

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

- Parámetros y constantes: $g, m_1, m_2, k_1, k_2, b_1, b_2$
- Variables de estado: v_1, v_2, e_1, e_2
- Variables algebraicas: F

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

$$\begin{array}{ll} \frac{dv_1}{dt} \rightarrow deriv_1 & \frac{dv_2}{dt} \rightarrow deriv_2 \\ \frac{de_1}{dt} \rightarrow dere_1 & \frac{de_2}{dt} \rightarrow dere_2 \end{array}$$

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

$$F = \begin{cases} 250 & \text{si } 60 < t < 120 \\ 0 & \text{en caso contrario} \end{cases} \quad (1.6)$$

$$m_1 \cdot deriv_1 = F - k_1 \cdot e_1 \quad (1.7)$$

$$m_2 \cdot deriv_2 = k_1 \cdot e_1 - k_2 \cdot e_2 - b_1 \cdot v_2 - b_2 \cdot v_2 + m_2 \cdot g \quad (1.8)$$

$$dere_1 = v_1 - v_2 \quad (1.9)$$

$$dere_2 = v_2 \quad (1.10)$$

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
 $g, m_1, m_2, k_1, k_2, b_1, b_2$
 v_1, v_2, e_1, e_2
- Desconocidas: F
 $deriv_1, deriv_2, dere_1, dere_2$

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 5 ecuaciones y 5 incógnitas.
2. Cada incógnita puede emparejarse con una ecuación en que aparezca y con la cual no se haya emparejado ya otra incógnita.

$$F \rightarrow F = \begin{cases} 250 & \text{si } 60 < t < 120 \\ 0 & \text{en caso contrario} \end{cases} \quad \text{Ec. (1.6)}$$

$$deriv_1 \rightarrow m_1 \cdot deriv_1 = F - k_1 \cdot e_1 \quad \text{Ec. (1.7)}$$

$$deriv_2 \rightarrow m_2 \cdot deriv_2 = k_1 \cdot e_1 - k_2 \cdot e_2 - b_1 \cdot v_2 - b_2 \cdot v_2 + m_2 \cdot g \quad \text{Ec. (1.8)}$$

$$dere_1 \rightarrow dere_1 = v_1 - v_2 \quad \text{Ec. (1.9)}$$

$$dere_2 \rightarrow dere_2 = v_2 \quad \text{Ec. (1.10)}$$

Finalmente, las ecuaciones del modelo, con la asignación de la causalidad computacional señalada, son las siguientes:

$$[F] = \begin{cases} 250 & \text{si } 60 < t < 120 \\ 0 & \text{en caso contrario} \end{cases} \quad (1.11)$$

$$m_1 \cdot [deriv_1] = F - k_1 \cdot e_1 \quad (1.12)$$

$$m_2 \cdot [deriv_2] = k_1 \cdot e_1 - k_2 \cdot e_2 - b_1 \cdot v_2 - b_2 \cdot v_2 + m_2 \cdot g \quad (1.13)$$

$$[dere_1] = v_1 - v_2 \quad (1.14)$$

$$[dere_2] = v_2 \quad (1.15)$$

El modelo ordenado y resuelto es el siguiente (existen otras posibles ordenaciones de las ecuaciones que son igualmente válidas):

$$[F] = \begin{cases} 250 & \text{si } 60 < t < 120 \\ 0 & \text{en caso contrario} \end{cases} \quad (1.16)$$

$$[deriv_1] = \frac{F - k_1 \cdot e_1}{m_1} \quad (1.17)$$

$$[deriv_2] = \frac{k_1 \cdot e_1 - k_2 \cdot e_2 - b_1 \cdot v_2 - b_2 \cdot v_2}{m_2} + g \quad (1.18)$$

$$[dere_1] = v_1 - v_2 \quad (1.19)$$

$$[dere_2] = v_2 \quad (1.20)$$

El modelo posee cuatro variables de estado, por lo que tiene cuatro grados de libertad. 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 180 s. Se han asignado valores arbitrarios a los valores iniciales de las variables de estado.

Se ha seleccionado un tamaño del paso igual a 10^{-4} s. Para ello, se ha repetido la simulación empleando diferentes tamaños del paso, encontrándose que el error cometido escogiendo 10^{-4} s es admisible para el propósito de este estudio.

Definimos un intervalo de comunicación de 0.2 s para el almacenamiento de los resultados.

A continuación, se muestra el código R y en la Figura 1.4 están las gráficas obtenidas de su ejecución.

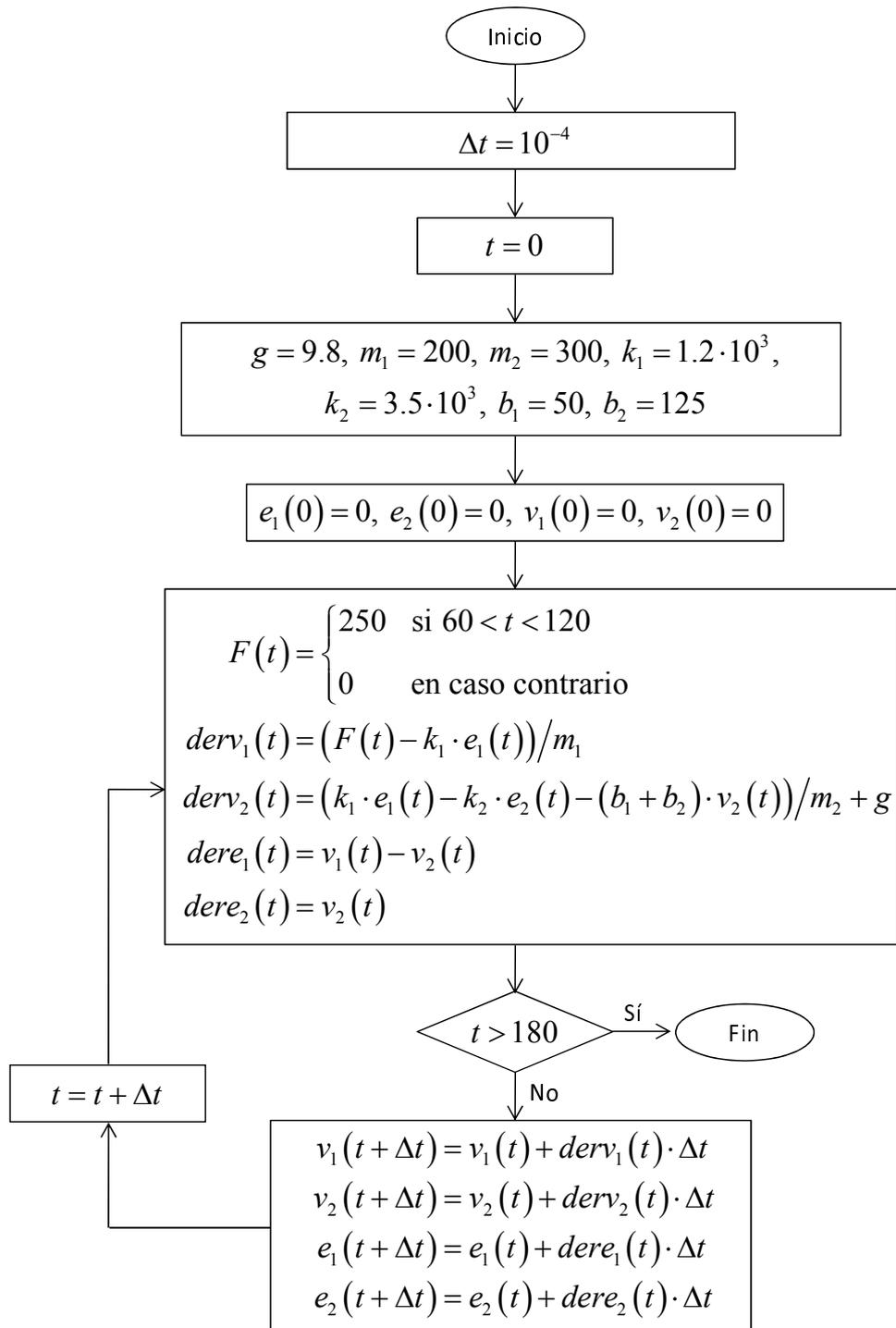


Figura 1.3: Diagrama de flujo de la simulación del modelo.

```

t_fin <- 180      # Valor final del tiempo
h      <- 0.0001 # Tamaño del paso de integración
t_com <- 0.2     # Intervalo de comunicación
# Parámetros
g      <- 9.8
m1     <- 200
m2     <- 300
k1     <- 1.2e3
k2     <- 3.5e3
b1     <- 50
b2     <- 125
# Inicialización del tiempo
t      <- 0
# Valor inicial del estado
e1     <- 0
e2     <- 0
v1     <- 0
v2     <- 0
# Inicialización resultados salida
t_plot      <- numeric(0)
F_plot      <- numeric(0)
e1_plot     <- numeric(0)
e2_plot     <- numeric(0)
v1_plot     <- numeric(0)
v2_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 < 60 | t > 120 ) {
    F <- 0
  } else {
    F <- 250
  }
  derv1 <- ( F - k1*e1 ) / m1
  derv2 <- ( k1*e1 - k2*e2 - ( b1 + b2 ) * v2 ) / m2 + g
  dere1 <- v1 - v2
  dere2 <- v2
  # Almacenar resultados de salida
  if ( ( t - t_ultimaCom ) >= t_com | termina ) {
    t_plot <- c( t_plot, t )
    F_plot <- c( F_plot, F )
    e1_plot <- c( e1_plot, e1 )
    e2_plot <- c( e2_plot, e2 )
    v1_plot <- c( v1_plot, v1 )
    v2_plot <- c( v2_plot, v2 )
    t_ultimaCom <- t
  }
  # Valor en t+h de los estados
  v1 <- v1 + h*derv1
  v2 <- v2 + h*derv2
  e1 <- e1 + h*dere1
  e2 <- e2 + h*dere2
  # 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

par( mfrow = c(2,1) )
# F vs t
par( mai = c(1,1,0.8,0.2), cex = 1.2)
plot(t_plot, F_plot, lty = 1, type = "l",
      xlim = c(0,t_fin), ylim = c(-20,260),
      yaxt = "n", xlab = "t [s]", ylab = "F [N]", lwd = 2 )
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(-20,260,10),
        lwd = 0.5, lty = 3)
abline(h = 0, lwd = 1, lty = 1)
title("Fuerza externa", cex = 1)

par( mfrow = c(2,1) )
# e1 vs t
par( mai = c(0.4,1,0.8,0.2), cex = 1.2 )
plot(t_plot, e1_plot, lty = 1, type = "l",
      xlim = c(0,t_fin), ylim = c(-0.8,1.4),
      yaxt = "n", xlab = "", ylab = "e1 [m]", lwd = 2 )
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(-0.8,1.4,0.05),
        lwd = 0.5, lty = 3)
abline(h = 0, lwd = 1, lty = 1)
title("Elongaciones", cex = 1)

# e2 vs t
par( mai = c(1,1,0.2,0.2), cex = 1.2 )
plot(t_plot, e2_plot, lty = 1, type = "l",
      xlim = c(0,t_fin), ylim = c(-0.8,1.4),
      yaxt = "n", xlab = "t [s]", ylab = "e2 [m]", lwd = 2 )
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(-0.8,1.4,0.05),
        lwd = 0.5, lty = 3)
abline(h = 0, lwd = 1, lty = 1)

par( mfrow = c(2,1) )
# v1 vs t
par( mai = c(0.4,1,0.8,0.2), cex = 1.2 )
plot(t_plot, v1_plot, lty = 1, type = "l",
      xlim = c(0,t_fin), ylim = c(-2.2,2.8),
      yaxt = "n", xlab = "", ylab = "v1 [m/s]", lwd = 2)
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(-2.2,2.8,0.2),
        lwd = 0.5, lty=3)
abline(h = 0, lwd = 1, lty=1)
title("Velocidades", cex = 1)

```

```
# v2 vs t
par( mai = c(1,1,0.2,0.2), cex = 1.2 )
plot(t_plot, v2_plot, lty = 1, type = "l",
      xlim = c(0,t_fin), ylim = c(-2.2,2.8),
      yaxt = "n", xlab = "t [s]", ylab = "v2 [m/s]", lwd = 2)
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(-2.2,2.8,0.2),
       lwd = 0.5, lty=3)
abline(h = 0, lwd = 1, lty=1)
```

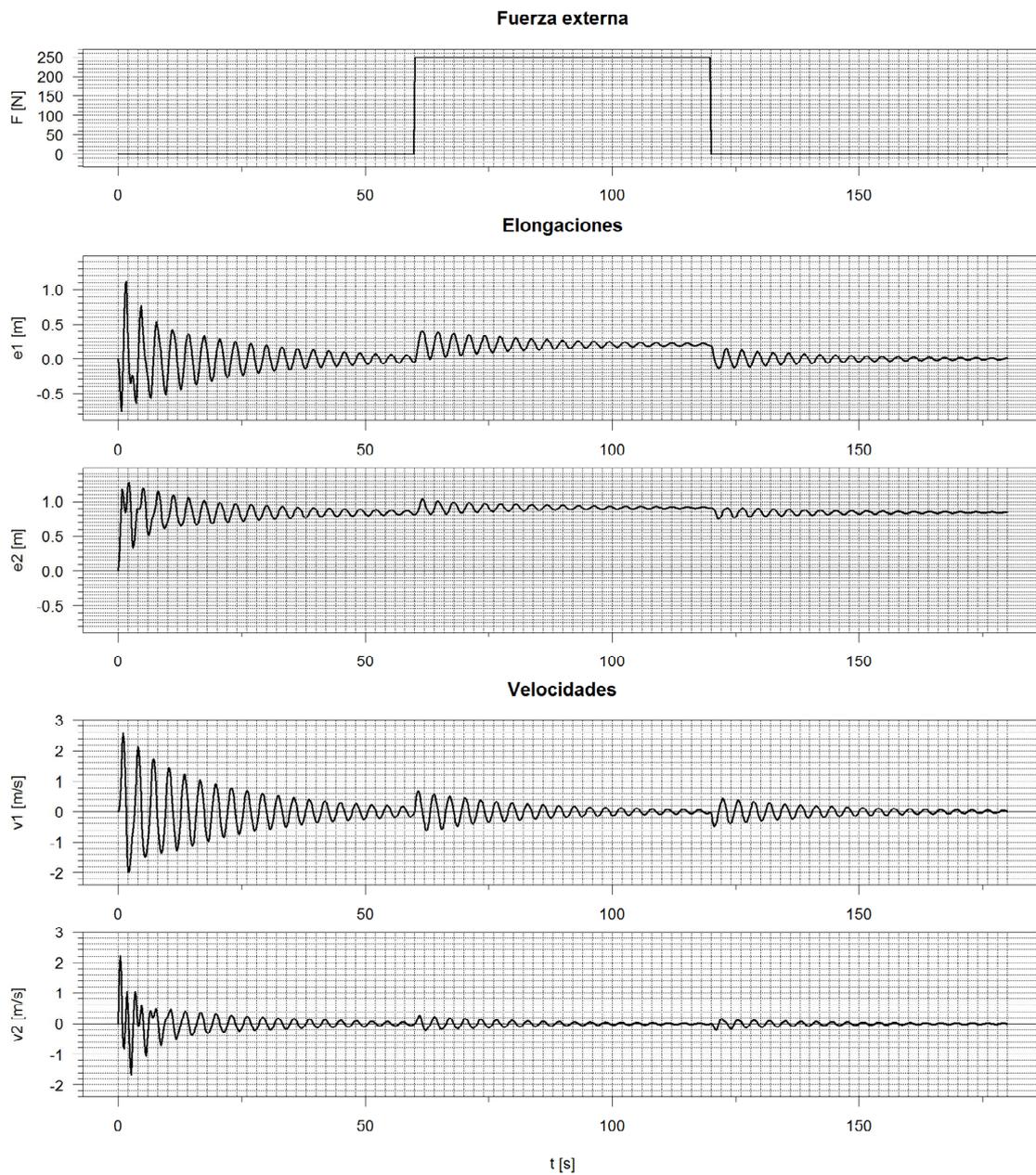


Figura 1.4: Resultado de la simulación del código R.

EJERCICIO 3

Describa en lenguaje Modelica el sistema mecánico 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 contestar a la pregunta anterior.
2. Programe una librería mecánica que contenga los componentes necesarios para componer el sistema mecánico de la Figura 1.2. A continuación, defina dicho sistema como un modelo compuesto, instanciando y conectando componentes de la librería que ha creado.

Asigne a los parámetros los valores indicados en el ejercicio anterior y simule los dos modelos anteriores durante 180 s. Represente gráficamente las elongaciones de los muelles frente al tiempo y las velocidades de los objetos frente al tiempo, mostrando las gráficas en la memoria.

Solución al Ejercicio 3

El modelo atómico en lenguaje Modelica se muestra en Código 1.1. El resultado de la simulación está representado en la Figura 1.5.

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 mecánico mediante dichos componentes. Obsérvese que los valores iniciales de las elongaciones de los muelles y las velocidades de los objetos se especifican al definir el experimento.

```

model sistMecanico
  import SI = Modelica.SIunits;

  constant SI.Acceleration g = 9.8;
  // Masas
  parameter SI.Mass m1 = 200;
  parameter SI.Mass m2 = 300;
  // Muelles, amortiguador, friccion
  parameter SI.TranslationalSpringConstant k1=1.2e3;
  parameter SI.TranslationalSpringConstant k2=3.5e3;
  parameter SI.TranslationalDampingConstant b1=50;
  parameter SI.TranslationalDampingConstant b2=125;
  // Longitud - Longitud natural de los muelles
  SI.Length e1(start=0, fixed=true);
  SI.Length e2(start=0, fixed=true);
  // Velocidades de los objetos
  SI.Velocity v1(start=0, fixed=true);
  SI.Velocity v2(start=0, fixed=true);
  // Fuerza externa
  SI.Force F;
equation
  // Fuerza externa
  F = if time > 60 and time < 120 then 250 else 0;
  // Objeto masa 1
  m1*der(v1) = F - k1*e1;
  // Objeto masa 2
  m2*der(v2) = k1*e1 - k2*e2 - b1*v2 - b2*v2 + m2*g;
  // Muelle 1
  der(e1) = v1 - v2;
  // Muelle 2
  der(e2) = v2;
  annotation (uses(Modelica(version="3.2.2")));
end sistMecanico;

```

Código 1.1: Ejercicio 3.1: modelo atómico del sistema mecánico.

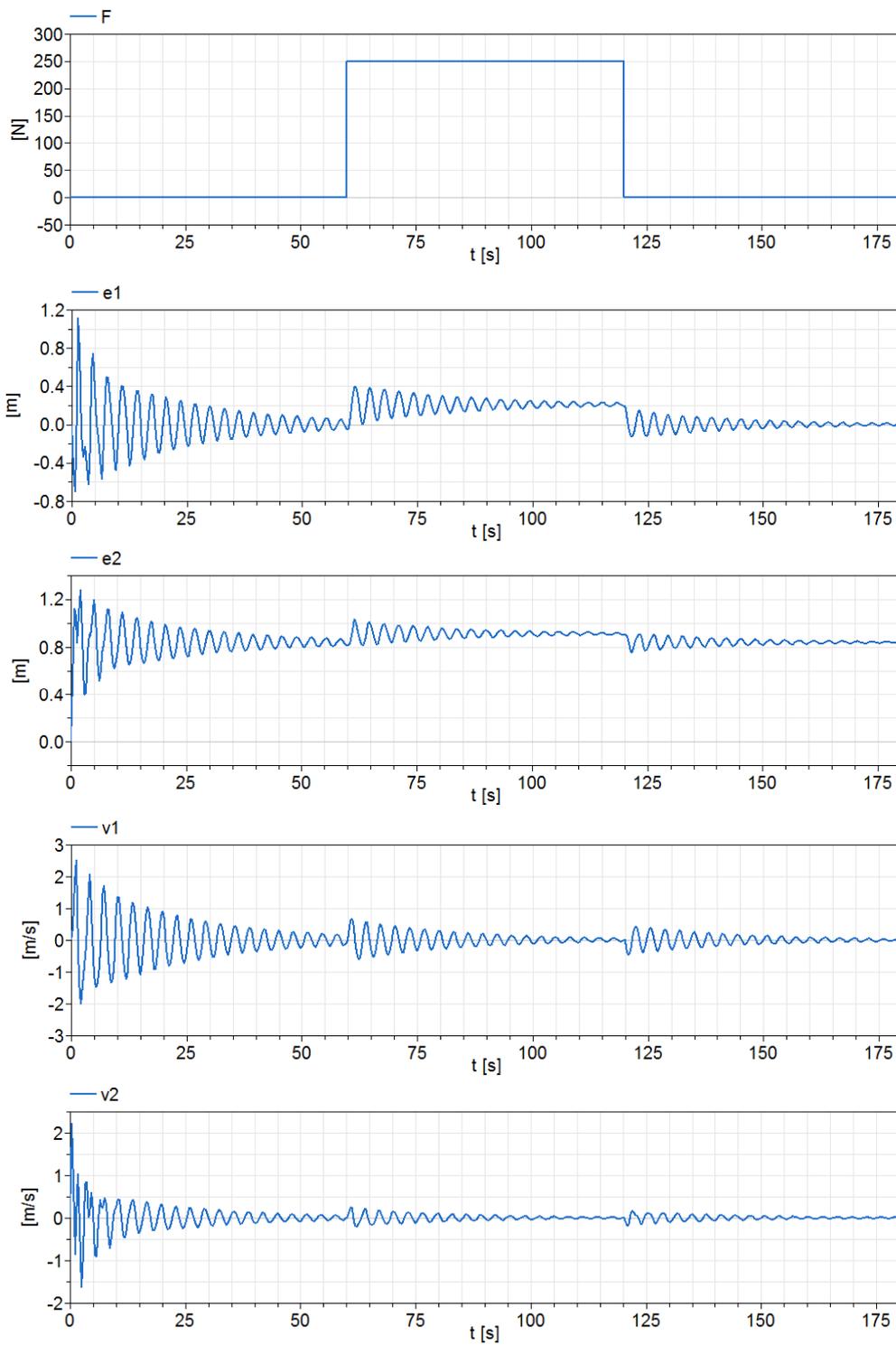


Figura 1.5: Resultado de la simulación usando Dymola.

```

package libMec1D "Libreria mecanica 1D"

import SI = Modelica.SIunits;

package Interfaz

    connector v_f "Velocidad / Fuerza"
        SI.Velocity v;
        flow SI.Force f;
    end v_f;

end Interfaz;

package Componentes

    model Muelle
        // (coord espacial p1) < (coord espacial p2)
        Interfaz.v_f p1;
        Interfaz.v_f p2;
        parameter SI.TranslationalSpringConstant k = 1;
        SI.Length e "Elongacion";
    equation
        der(e) = p2.v - p1.v;
        p1.f = k*e;
        p2.f = -p1.f;
    end Muelle;

    model Masa
        Interfaz.v_f p;
        parameter SI.Mass m=1 "Masa";
    equation
        -p.f = m*der(p.v);
    end Masa;

    model Reposo
        Interfaz.v_f p;
    equation
        p.v = 0;
    end Reposo;

    model Amortiguador
        // (coord espacial p1) < (coord espacial p2)
        Interfaz.v_f p1;
        Interfaz.v_f p2;
        parameter SI.TranslationalDampingConstant b=1;
    equation
        p1.f = b*(p2.v - p1.v);
        p2.f = -p1.f;
    end Amortiguador;

```

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

```

model PulsoFuerza
  Interfaz.v_f p;
  parameter SI.Time t1,t2;
  parameter SI.Force F1,F2;
equation
  p.f = if time < t1 or time > t2 then F1 else F2;
end PulsoFuerza;

model FuerzaConstante
  Interfaz.v_f p;
  parameter SI.Force F0;
equation
  p.f = F0;
end FuerzaConstante;
end Componentes;

model SistemaMecanico
  import libMec1D.Componentes.*;
  constant SI.Acceleration g = 9.8;
  parameter SI.Mass m1 = 200;
  parameter SI.Mass m2 = 300;
  parameter SI.TranslationalSpringConstant k1 = 1.2e3;
  parameter SI.TranslationalSpringConstant k2 = 3.5e3;
  parameter SI.TranslationalDampingConstant b1 = 50;
  parameter SI.TranslationalDampingConstant b2 = 125;
  parameter SI.Time t1 = 60;
  parameter SI.Time t2 = 120;
  parameter SI.Force Fext = 250;
  Reposo techo;
  Reposo pared;
  Masa masa1(m=m1);
  Masa masa2(m=m2);
  Amortiguador amortiguador(b=b1);
  Amortiguador friccionPared(b=b2);
  Muelle muelle1(k=k1);
  Muelle muelle2(k=k2);
  PulsoFuerza fuerzaExterna(t1=t1,t2=t2,F1=0,F2=Fext);
  FuerzaConstante pesoM2(F0=m2*g);
equation
  connect(techo.p,amortiguador.p1);
  connect(techo.p,muelle2.p1);
  connect(amortiguador.p2,masa2.p);
  connect(muelle2.p2,masa2.p);
  connect(masa2.p,pesoM2.p);
  connect(pared.p,friccionPared.p1);
  connect(friccionPared.p2,masa2.p);
  connect(muelle1.p1,masa2.p);
  connect(muelle1.p2,masa1.p);
  connect(masa1.p,fuerzaExterna.p);
end SistemaMecanico;

end libMec1D;

```

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

EJERCICIO 4

Consideremos el problema bidimensional de la distribución de calor en un bloque rectangular de un determinado material, que tiene un hueco rectangular en su centro. La geometría se muestra en la Figura 1.6. Se han etiquetado los lados del bloque de $L1$ a $L8$. El bloque rectangular tiene dos lados de longitud 0.8 y 1.6 m. El hueco interior tiene forma rectangular y sus lados miden 0.1 y 0.8 m.

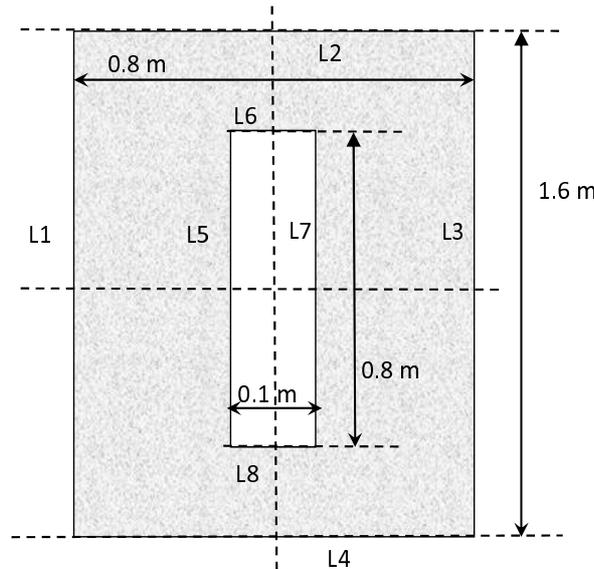


Figura 1.6: Sistema térmico.

El fenómeno térmico que tiene lugar en el bloque se describe por la siguiente ecuación diferencial:

$$\nabla \cdot (-k \cdot \nabla temp) + r_{cp} \cdot \frac{dtemp}{dt} = heat$$

donde $r_{cp} = 1$ en unidades del SI. La conductividad térmica del material (κ) depende de la temperatura ($temp$): $\kappa = 1 + \frac{10}{temp}$.

El lado $L1$ se mantiene a una temperatura constante de 400 K. Existe una pérdida de calor en el lado $L3$ a una tasa constante de 10 W/m^2 . El resto de lados están bien aislados. La temperatura inicial del bloque, en $t = 0$, es de 273 K.

Obtenga, para $t = 5 \text{ s}$, los dos siguientes gráficos empleando FLEXPDE: un gráfico con las curvas de nivel de la temperatura y un gráfico vectorial del flujo de calor. Escriba el código del *script* de FlexPDE correspondiente y muestre los dos gráficos generados por FlexPDE.

Solución al Ejercicio 4

El *script* se muestra en el Código 1.4, el gráfico con las curvas de nivel de la temperatura en la Figura 1.7 y el gráfico vectorial del flujo de calor en la Figura 1.8.

```

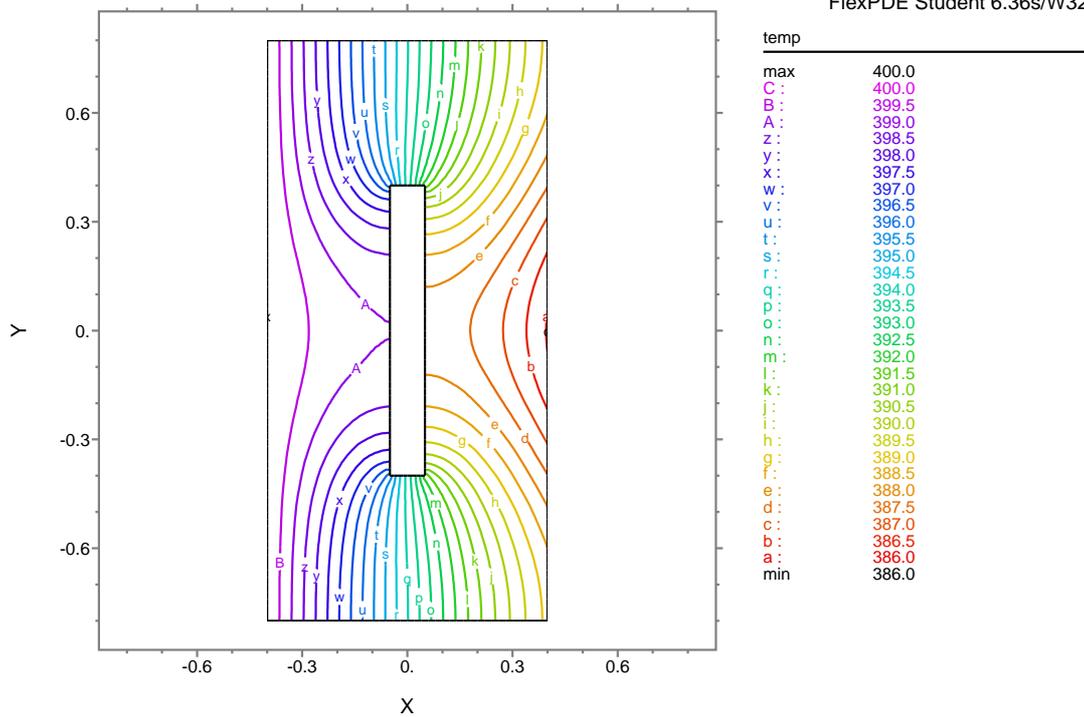
TITLE 'Bloque con cavidad' { Flujo de calor en un bloque con cavidad interior }
SELECT
  errlim=1e-3 spectral_colors
VARIABLES temp
DEFINITIONS
  Lx=0.4 Ly = 0.8 Cx = 0.05 Cy = 0.4
  rcp = 1
  heat = 0 k = 1+10/temp
  fluxd_x = -k*dx(temp) fluxd_y=-k*dy(temp)
  fluxd=vector(fluxd_x,fluxd_y)
INITIAL VALUES
  temp = 273
EQUATIONS
  div( fluxd)+ rcp*dt( temp)=heat
BOUNDARIES
region 'dominio' {Define el dominio del problema}
  start (-Lx,Ly) value(temp) = 400
    line to (-Lx,-Ly) natural(temp) = 0
    line to (Lx, -Ly) natural(temp) = 10
    line to (Lx, Ly) natural(temp)=0 line to finish
  start(-Cx,Cy) line to (-Cx,-Cy) {Region excluida}
    line to (Cx,-Cy)
    line to (Cx, Cy)
    line to finish
TIME FROM 0 TO 5 by 0.05
PLOTS
FOR T = 0 2.5 5
CONTOUR(temp)
VECTOR(fluxd)
end
END

```

Código 1.4: Distribución de calor en un bloque rectangular con cavidad.

Bloque con cavidad

11:05:42 1/13/19
FlexPDE Student 6.36s/W32



bloque: Cycle=153 Time= 5.0000 dt= 0.4060 P2 Nodes=552 Cells=248 RMS Err= 7.4e-4
Integral= 472.4923

Figura 1.7: Curvas de temperatura.

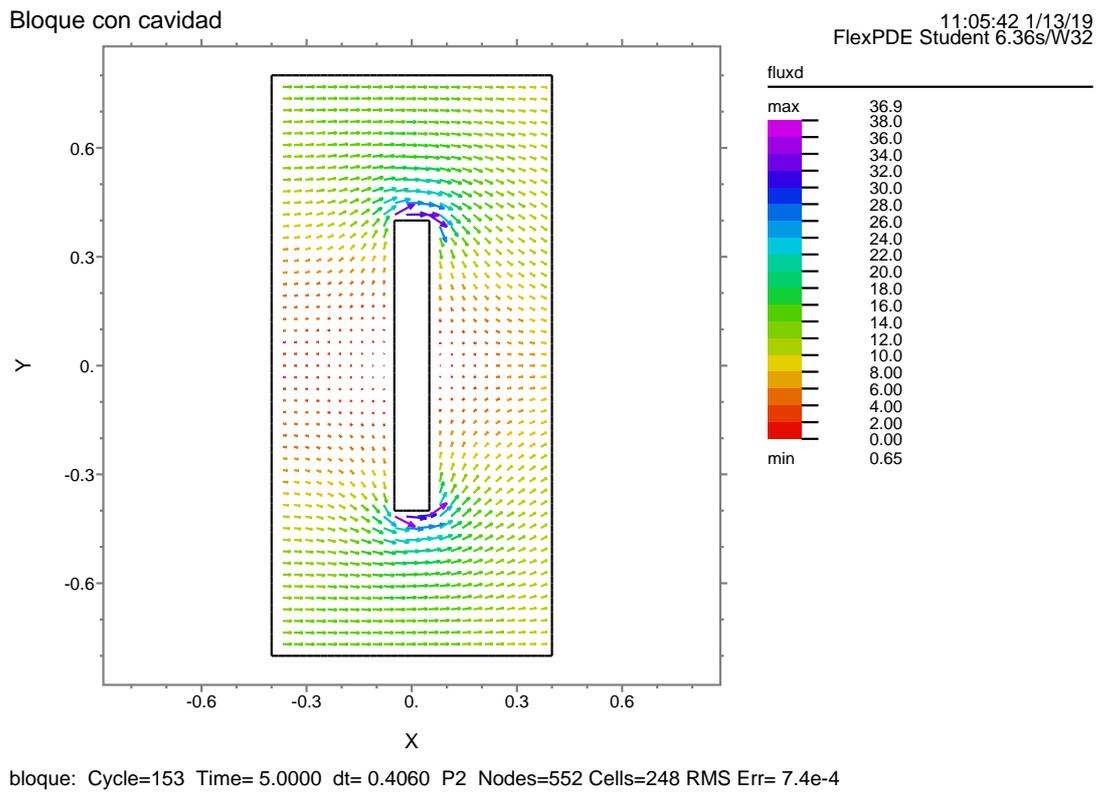


Figura 1.8: Flujo de calor