

MÉTODOS DE SIMULACIÓN Y MODELADO

Solución al Trabajo Práctico - Enero de 2018

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.

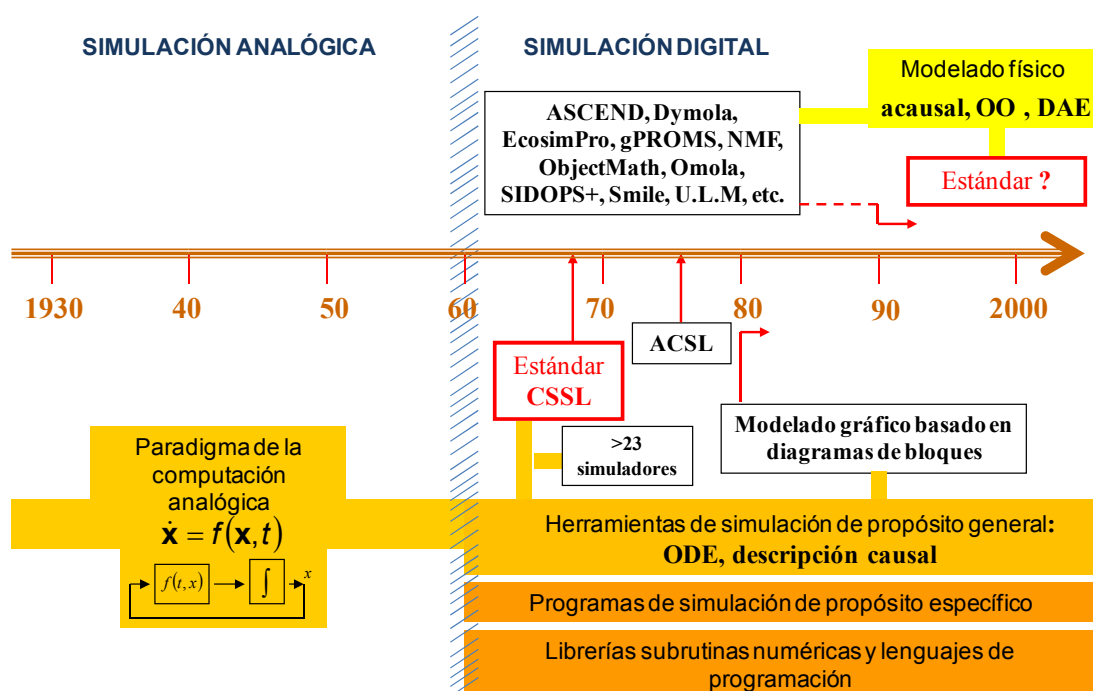


Figura 1.1: Evolución del modelado y simulación de tiempo continuo.

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 un objeto de masa constante m_1 , un muelle y un amortiguador. La pared lateral y el suelo se encuentran en reposo. El objeto desliza sin rozamiento sobre el suelo, y se encuentra unido a la pared lateral mediante el muelle y el amortiguador.

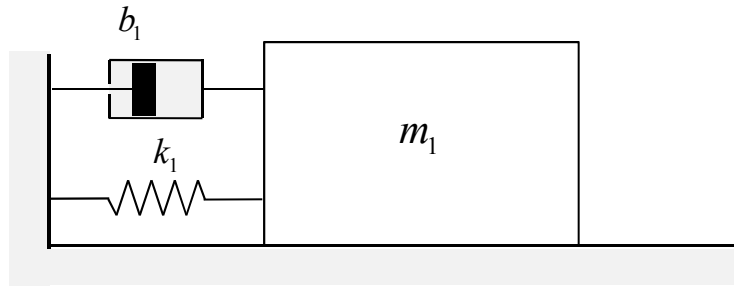


Figura 1.2: Diagrama del sistema mecánico.

El coeficiente del amortiguador (b_1), el coeficiente del muelle (k_1) y la masa del objeto (m_1) son parámetros de valor conocido.

$$m_1 = 200 \text{ kg} \qquad k_1 = 1.2 \text{ N/m} \qquad b_1 = 1.8 \text{ N}\cdot\text{s/m}$$

1. Escriba las ecuaciones del modelo del sistema mecánico. El modelo debe describir la evolución de la posición y la velocidad del objeto.
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 400 s.
4. Programe el algoritmo anterior en lenguaje R y ejecute la simulación. Represente gráficamente la posición del objeto frente al tiempo y la velocidad del objeto 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.

$$m_1 \cdot \frac{dv}{dt} = -k_1 \cdot (x - L) - b_1 \cdot v \quad (1.1)$$

$$\frac{dx}{dt} = v \quad (1.2)$$

La variable t representa el tiempo. El parámetro L representa la longitud natural del muelle y x la longitud actual del muelle, que coincide con la posición en la que se encuentra la masa. Se toma como origen de coordenadas el punto de anclaje del muelle a la pared. La velocidad de la masa es v . La velocidad es positiva cuando la masa se desplaza hacia la derecha, es decir, cuando aumenta la elongación del muelle.

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: m_1, k_1, b_1, L
- Variables de estado: x, v
- Variables algebraicas:

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

$$\frac{dx}{dt} \rightarrow derx \qquad \frac{dv}{dt} \rightarrow deriv$$

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

$$m_1 \cdot deriv = -k_1 \cdot (x - L) - b_1 \cdot v \quad (1.3)$$

$$derx = v \quad (1.4)$$

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
 m_1, k_1, b_1, L
 x, v
- Desconocidas: $derx, deriv$

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

$$deriv \rightarrow m_1 \cdot deriv = -k_1 \cdot (x - L) - b_1 \cdot v \quad \text{Ec. (1.3)}$$

$$derx \rightarrow derx = v \quad \text{Ec. (1.4)}$$

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

$$m_1 \cdot [deriv] = -k_1 \cdot (x - L) - b_1 \cdot v \quad (1.5)$$

$$[derx] = v \quad (1.6)$$

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

$$[deriv] = \frac{-k_1 \cdot (x - L) - b_1 \cdot v}{m_1} \quad (1.7)$$

$$[derx] = v \quad (1.8)$$

El modelo posee dos variables de estado, por lo que tiene dos 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 400 s. Se han asignado valores arbitrarios a los valores iniciales de las variables de estado.

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.

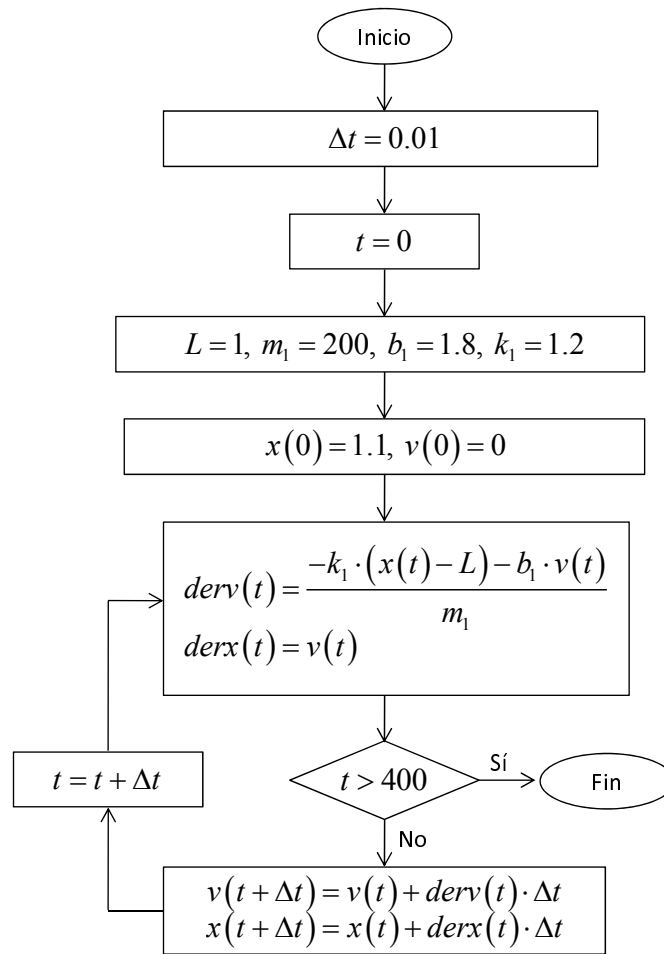


Figura 1.3: Diagrama de flujo de la simulación del modelo del circuito.

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

Dado que el modelo se simula durante 400 s y el método de integración avanza a pasos de 0.01 s, se producen 40,000 pasos de integración. Para el propósito de este estudio no es necesario almacenar el valor de las variables x y v en todos esos instantes de tiempo. Así pues, se define un intervalo de comunicación de 0.5 s para el almacenamiento de los resultados. Durante la ejecución de la simulación únicamente se va almacenando el valor del tiempo, la posición (x) y la velocidad (v) en los instantes de comunicación. De esta manera, el número de valores de cada variable que se almacena para su representación gráfica es aproximadamente 800, en lugar de 40,000.

```

t_fin <- 400      # Valor final del tiempo
h      <- 0.01    # Tamaño del paso de integración
t_com  <- 0.5     # Intervalo de comunicación
# Parámetros
L      <- 1
m1     <- 200
b1     <- 1.8
k1     <- 1.2
# Inicialización del tiempo
t      <- 0
# Valor inicial del estado
x      <- 1.1
v      <- 0
# Inicialización resultados salida
t_plot <- numeric(0)
x_plot <- numeric(0)
v_plot <- numeric(0)
t_ultimaCom <- -Inf
# Bucle de la simulación
termina <- F
while ( !termina ) {
  termina <- (t > t_fin)
  # Cálculo variables algebraicas en t
  deriv <- (-k1*(x-L)-b1*v)/m1
  derx <- v
  # Almacenar resultados de salida
  if ( (t - t_ultimaCom) >= t_com | termina ) {
    t_plot <- c(t_plot, t)
    x_plot <- c(x_plot, x)
    v_plot <- c(v_plot, v)
    t_ultimaCom <- t
  }
  # Valor en t+h de los estados
  v <- v + h*deriv
  x <- x + h*derx
  # 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) )
# x vs t
par( mai = c(0.3,1.2,0.4,0.1), cex = 1.3 )
plot(t_plot, x_plot, lty = 1, type = "l",
      xlim = c(0,t_fin), ylim = c(0.9,1.12), xaxt = "n",
      yaxt = "n", xlab = "", ylab = "x [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(0.9,1.12,0.01),
       lwd = 0.5, lty = 3)
abline(h = 0, lwd = 1, lty = 1)
title("Posición y velocidad", cex = 1)

```

```
# v vs t
par( mai=c(1.1,1.2,0,0.1), cex = 1.3 )
plot(t_plot, v_plot, lty = 1, type = "l",
      xlim = c(0,t_fin), ylim = c(-0.008,0.008),
      yaxt = "n", xlab = "t [s]", ylab = "v [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(-0.008,0.008,0.001),
       lwd = 0.5, lty=3)
abline(h = 0, lwd = 1, lty=1)
```

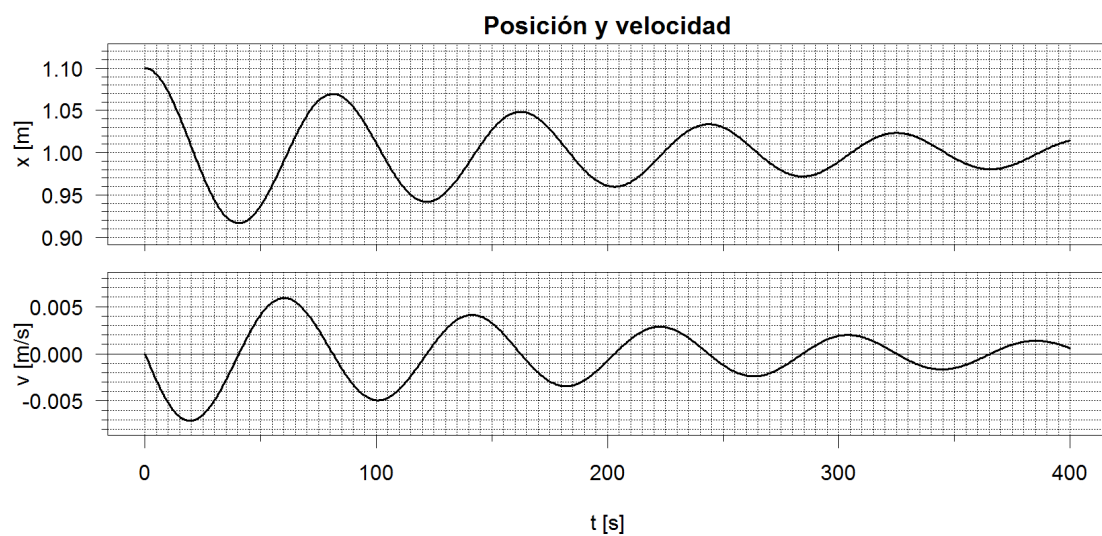


Figura 1.4: Resultado de la simulación del código R.

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 400 s. Represente gráficamente la posición y velocidad del objeto 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.

```

model sistMecanico
  import SI = Modelica.SIunits;
  parameter SI.Mass m1=200 "Masa del objeto";
  parameter SI.TranslationalDampingConstant b1=1.8 "Amortiguador";
  parameter SI.TranslationalSpringConstant k1=1.2 "Muelle";
  parameter SI.Length L = 1 "Longitud natural muelle";
  SI.Position x(start=1.1,fixed=true) "Posicion del objeto";
  SI.Velocity v(start=0,fixed=true) "Velocidad del objeto";
equation
  der(x) = v;
  m1*der(v) = - b1*v - k1*(x-L);
end sistMecanico;

```

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

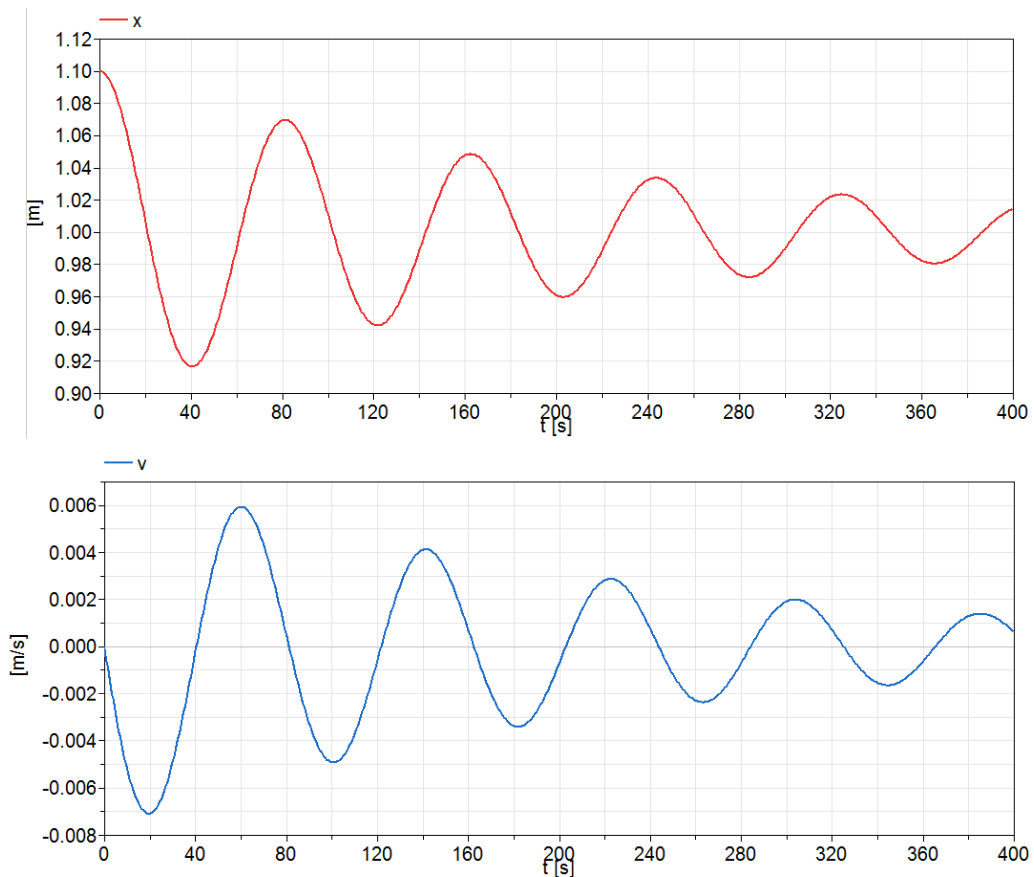


Figura 1.5: Resultado de la simulación usando Dymola.

A continuación se muestra una posible forma de modelar los componentes, organizándolos en una librería, y de componer el sistema mecánico mediante dichos componentes. Obsérvese que el valor inicial de la posición y velocidad de la masa se especifica al definir el experimento.

```

package libMec1D "Libreria mecanica 1D"

import SI = Modelica.SIunits;

package Interfaz

    connector x_f "Posicion / Fuerza"
        SI.Position x;
        flow SI.Force f;
    end x_f;

end Interfaz;

package Componentes

    model Muelle
        Interfaz.x_f p1 "Conector a la izquierda";
        Interfaz.x_f p2 "Conector a la derecha";
        parameter SI.TranslationalSpringConstant k = 1;
        parameter SI.Length L=1 "Longitud natural";
    equation
        p1.f = k*(p2.x - p1.x - L);
        p2.f = -p1.f;
    end Muelle;

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

    model Pared
        Interfaz.x_f p;
        parameter SI.Position x0 = 0;
    equation
        p.x = x0;
    end Pared;

```

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

```

model Amortiguador
  Interfaz.x_f p1 "Conector a la izquierda";
  Interfaz.x_f p2 "Conector a la derecha";
  parameter SI.TranslationalDampingConstant b=1;
  equation
    p1.f = b*(der(p2.x) - der(p1.x));
    p2.f = -p1.f;
  end Amortiguador;

end Componentes;

model SistemaMecanico
  Componentes.Pared pared;
  Componentes.Masa masa(m=200);
  Componentes.Amortiguador amortiguador(b=1.8);
  Componentes.Muelle muelle(k=1.2);
  equation
    connect(pared.p, amortiguador.p1);
    connect(pared.p, muelle.p1);
    connect(amortiguador.p2, masa.p);
    connect(muelle.p2, masa.p);
  end SistemaMecanico;

end libMec1D;

```

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

EJERCICIO 4

Consideremos el problema en dos dimensiones de un tubo cilíndrico fabricado con un material aislante que contiene en su interior dos tubos cilíndricos de igual radio que transportan agua a diferente temperatura. En la Figura 1.6 se muestra el sistema descrito en dos dimensiones, que es totalmente simétrico.

El radio del tubo exterior es 0.5 m y el radio de los tubos internos es 0.1 m. La distancia d mostrada en la Figura 1.6 vale 0.15 m. El tubo exterior está a 273 K, y los tubos interiores izquierdo y derecho están, respectivamente, a 323 K y a 353 K. La conductividad térmica del medio es $\kappa = 0.03 \frac{W}{K \cdot m}$.

Obtenga un gráfico con las curvas de nivel de la temperatura empleando FlexPDE. Escriba el código del *script* de FlexPDE correspondiente y muestre el gráfico generado por FlexPDE.

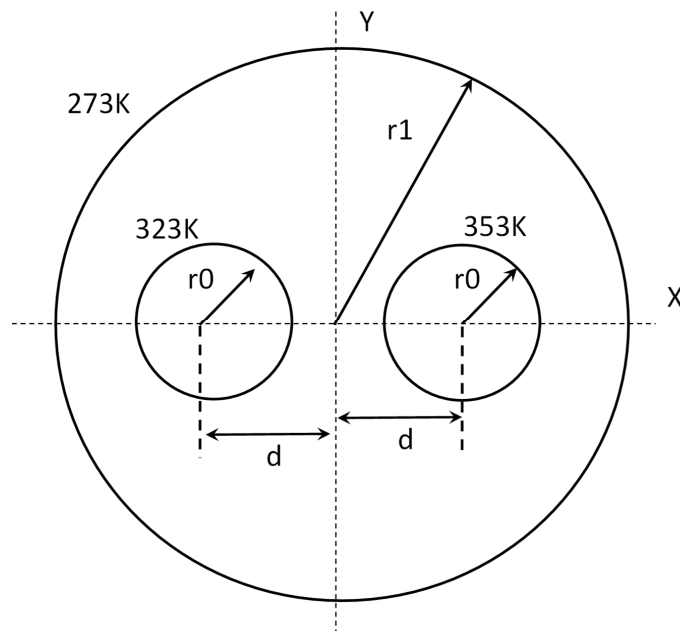


Figura 1.6: Sistema térmico.

Solución al Ejercicio 4

El *script* se muestra en el Código 1.4 y el gráfico en la Figura 1.7.

```

TITLE      'Tuberias'
SELECT     errlim=3e-4      ngrid=1      spectral_colors
VARIABLES  temp            { unidades SI : m, K, W }
DEFINITIONS
    r0=0.1      d=0.15      r1=0.5
    k=0.03      {Conductividad termica }
EQUATIONS
    div(-k*grad(temp))=0
BOUNDARIES
region 'dominio'
    start 'exterior' (0,-r1) value(temp)=273 { Tuberia exterior }
    arc(center=0,0) angle=360
    start 'izquierda' (-d-r0,0) value(temp)=323 { Tuberia izquierda }
    arc(center=-d,0) angle=360
    start 'derecha' (d-r0,0) value(temp)=353 { Tuberia derecha }
    arc(center=d,0) angle=360
PLOTS
    contour( temp)
END

```

Código 1.4: Tuberías.

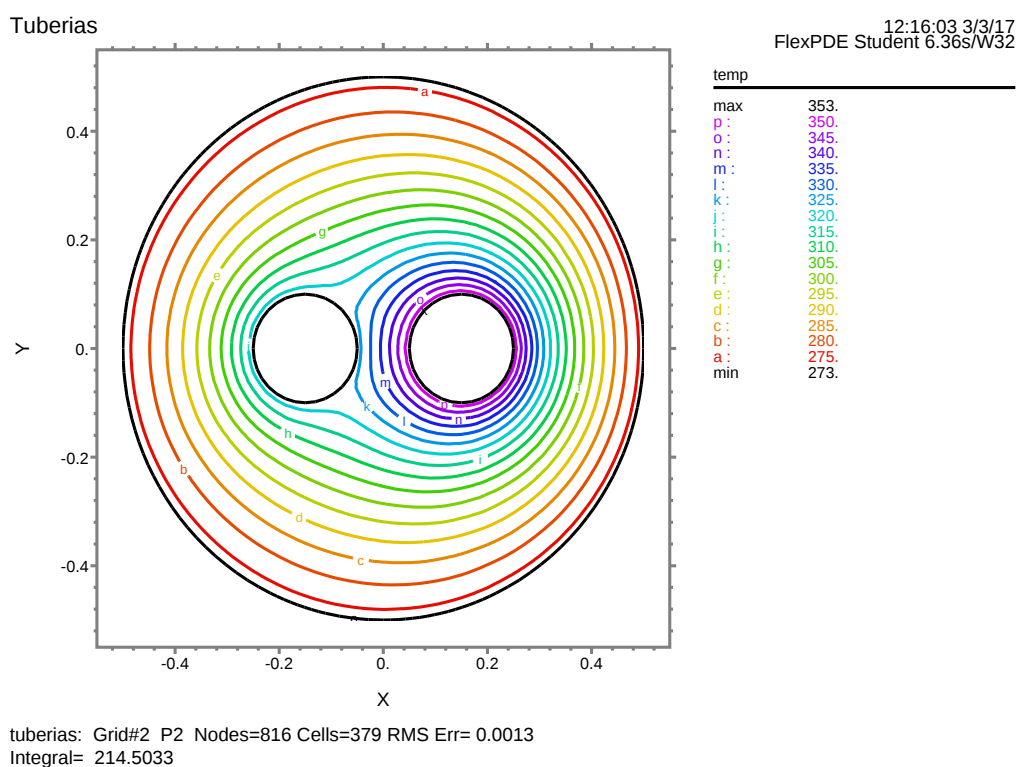


Figura 1.7: Curvas de temperatura.