

MÉTODOS DE SIMULACIÓN Y MODELADO

Solución al Trabajo Práctico - Enero de 2026

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.

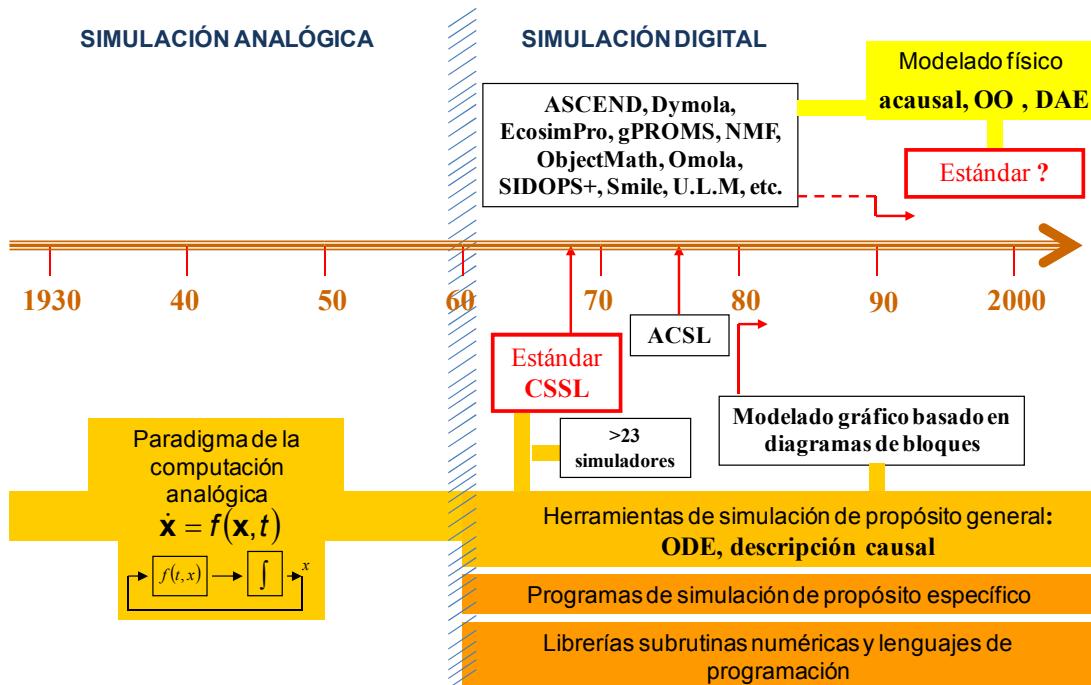


Figura 1: Evolución del modelado y simulación de tiempo continuo.

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 diagrama de un sistema compuesto por un depósito de líquido dotado de un sensor de nivel, una válvula, dos bombas de líquido y un controlador. El depósito tiene dos salidas de líquido en su base. A una de estas salidas está conectada una tubería que desagua un caudal F_3 al entorno a través de una válvula cuya apertura se mantiene fija a un valor constante. A la otra salida está conectada una bomba que recibe una señal de consigna $F_{2,SP}$ que es una función conocida del tiempo. Una segunda bomba introduce líquido a través de la parte superior del depósito. El valor de consigna para el caudal de esta bomba es la señal ϕ , generada por el controlador. La entrada h_{SP} al controlador es una función conocida del tiempo. El área de la base del depósito es el parámetro S .

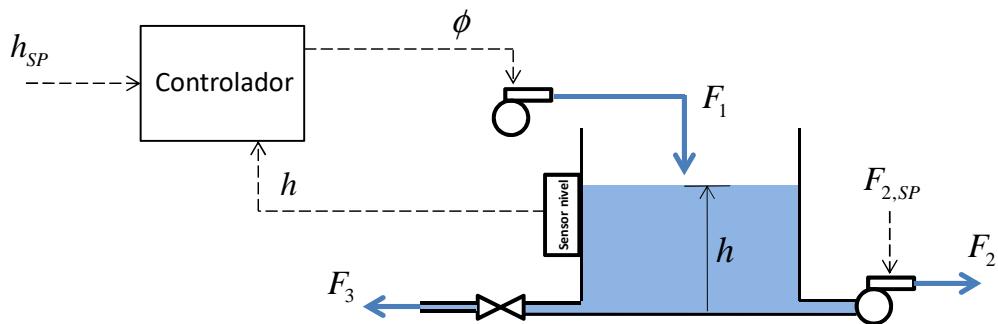


Figura 2: Diagrama del sistema.

Suponemos que la densidad del líquido es constante. También, que el caudal a través de la válvula (F_3) es proporcional al nivel de líquido:

$$F_3 = \begin{cases} \alpha \cdot h & \text{si } h > 0 \\ 0 & \text{en caso contrario} \end{cases} \quad (1.1)$$

donde α es un parámetro del modelo. El caudal de la bomba que introduce líquido por la parte superior del depósito es igual a su valor de consigna, que es la señal ϕ generada por el controlador:

$$F_1 = \phi \quad (1.2)$$

Mientras haya líquido en el depósito, el caudal F_2 de salida de la bomba es igual a su valor de consigna $F_{2,SP}$. Sin embargo, si el depósito está vacío, el caudal de salida de la bomba será nulo.

$$F_2 = \begin{cases} F_{2,SP} & \text{si } h > 0 \\ 0 & \text{en caso contrario} \end{cases} \quad (1.3)$$

La variable de entrada al controlador h_{SP} , que representa el valor de consigna para el nivel de líquido, y la variable $F_{2,SP}$, que representa el valor de consigna de la bomba, son funciones conocidas del tiempo, donde A, B, C son parámetros:

$$h_{SP} = \begin{cases} 4.5 \text{ m} & \text{si } t \in [1200, 1800] \cup [5000, 6000] \text{ s} \\ 3 \text{ m} & \text{si } t \in [2400, 3200] \text{ s} \\ 0.5 \text{ m} & \text{en cualquier otro caso} \end{cases} \quad (1.4)$$

$$F_{2,SP} = A \cdot (\sin(B \cdot t) \cdot \cos(C \cdot t))^2 \quad (1.5)$$

El controlador calcula el valor de la variable ϕ , que es el caudal de consigna de la bomba que introduce líquido en el depósito, siguiendo la estrategia de control PI muestreado descrita a continuación. Comenzando en el instante T_0 , cada T segundos el controlador reevalúa la señal de error (e), la integral de la señal de error (I) y la variable manipulada (ϕ) de la forma descrita a continuación, donde k_P y k_I son parámetros:

$$e = h_{SP} - h \quad (1.6)$$

$$I = \max(0, \text{pre}(I) + T \cdot e) \quad (1.7)$$

$$\phi = \max\left(0, k_P \cdot e + \frac{I}{k_I}\right) \quad (1.8)$$

A continuación se indica el valor de los parámetros del modelo.

$$S = 7 \text{ m}^2$$

$$T = 6 \text{ s}$$

$$T_0 = 3 \text{ s}$$

$$k_P = 0.2 \text{ m}^2 \cdot \text{s}^{-1}$$

$$k_I = 150 \text{ s}^2 \cdot \text{m}^{-2}$$

$$\alpha = 0.15 \text{ m}^2 \cdot \text{s}^{-1}$$

$$A = 0.3 \text{ m}^3 \cdot \text{s}^{-1}$$

$$B = 0.001 \text{ s}^{-1}$$

$$C = 0.002 \text{ s}^{-1}$$

En el instante inicial de la simulación ($t = 0$), el nivel de líquido en el depósito es $h(0) = 2.5 \text{ m}$.

Realice las tareas siguientes:

1. Escriba las ecuaciones del modelo.
2. Asigne la causalidad computacional. Explique detalladamente el procedimiento seguido para ello.
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 7200 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 frente al tiempo en una gráfica las variables F_1 , F_2 y F_3 , y en una segunda gráfica las variables h y h_{SP} .

Solución al Ejercicio 2

Las ecuaciones del modelo son:

$$S \cdot \frac{dh}{dt} = F_1 - F_2 - F_3 \quad (1.9)$$

$$F_1 = \phi \quad (1.10)$$

$$F_{2,SP} = A \cdot (\sin(B \cdot t) \cdot \cos(C \cdot t))^2 \quad (1.11)$$

$$F_2 = \text{if } h > 0 \text{ then } F_{2,SP} \text{ else } 0 \quad (1.12)$$

$$F_3 = \text{if } h > 0 \text{ then } \alpha \cdot h \text{ else } 0 \quad (1.13)$$

$$\begin{aligned} h_{SP} = & \text{if } t \geq 1200 \text{ and } t \leq 1800 \text{ or } t \geq 5000 \text{ and } t \leq 6000 \\ & \text{then } 4.5 \\ & \text{else if } t \geq 2400 \text{ and } t \leq 3200 \text{ then } 3 \text{ else } 0.5 \end{aligned} \quad (1.14)$$

$$\text{when sample}(T_0, T) \text{ then} \quad (1.15)$$

$$e = h_{SP} - h$$

$$I = \max(0, \text{pre}(I) + T \cdot e)$$

$$\phi = \max(0, k_P \cdot e + \frac{I}{k_I})$$

end when

Para asignar la causalidad computacional a la parte de tiempo continuo del modelo, asumimos que las variables de tiempo discreto (e , I , ϕ , F_1) son conocidas durante la resolución del problema de tiempo continuo, ya que su valor solo cambia en la ejecución de los eventos.

Asumiendo que la variable que aparece derivada (h) pueden ser seleccionada como variable de estado de tiempo continuo, las variables de la parte de tiempo continuo del modelo pueden clasificarse de la forma siguiente:

- Parámetros y constantes: S, A, B, C, α
- Variable de estado: h
- Variables algebraicas: $F_2, F_3, F_{2,SP}, h_{SP}$

Para asignar la causalidad computacional al modelo, se sustituye la derivada de la variable de estado de tiempo continuo por una variable muda:

$$\frac{dh}{dt} \rightarrow derh$$

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

h Variable de estado

$$S \cdot derh = F_1 - F_2 - F_3 \quad (1.16)$$

$$F_{2,SP} = A \cdot (\sin(B \cdot t) \cdot \cos(C \cdot t))^2 \quad (1.17)$$

$$F_2 = \text{if } h > 0 \text{ then } F_{2,SP} \text{ else } 0 \quad (1.18)$$

$$F_3 = \text{if } h > 0 \text{ then } \alpha \cdot h \text{ else } 0 \quad (1.19)$$

$$\begin{aligned} h_{SP} = & \text{if } t \geq 1200 \text{ and } t \leq 1800 \text{ or } t \geq 5000 \text{ and } t \leq 6000 \\ & \text{then } 4.5 \\ & \text{else if } t \geq 2400 \text{ and } t \leq 3200 \text{ then } 3 \text{ else } 0.5 \end{aligned} \quad (1.20)$$

Para asignar la causalidad computacional al problema de tiempo continuo, las variables pueden clasificarse en conocidas (el tiempo, los parámetros, las constantes, las variables de tiempo discreto 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
 S, A, B, C, α
 F_1
 h
- Desconocidas: $F_2, F_3, F_{2,SP}, h_{SP}$
 $derh$

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.

$$\begin{aligned}
 derh &\rightarrow \text{Ec. (1.16)} \\
 F_{2,SP} &\rightarrow \text{Ec. (1.17)} \\
 F_2 &\rightarrow \text{Ec. (1.18)} \\
 F_3 &\rightarrow \text{Ec. (1.19)} \\
 h_{SP} &\rightarrow \text{Ec. (1.20)}
 \end{aligned}$$

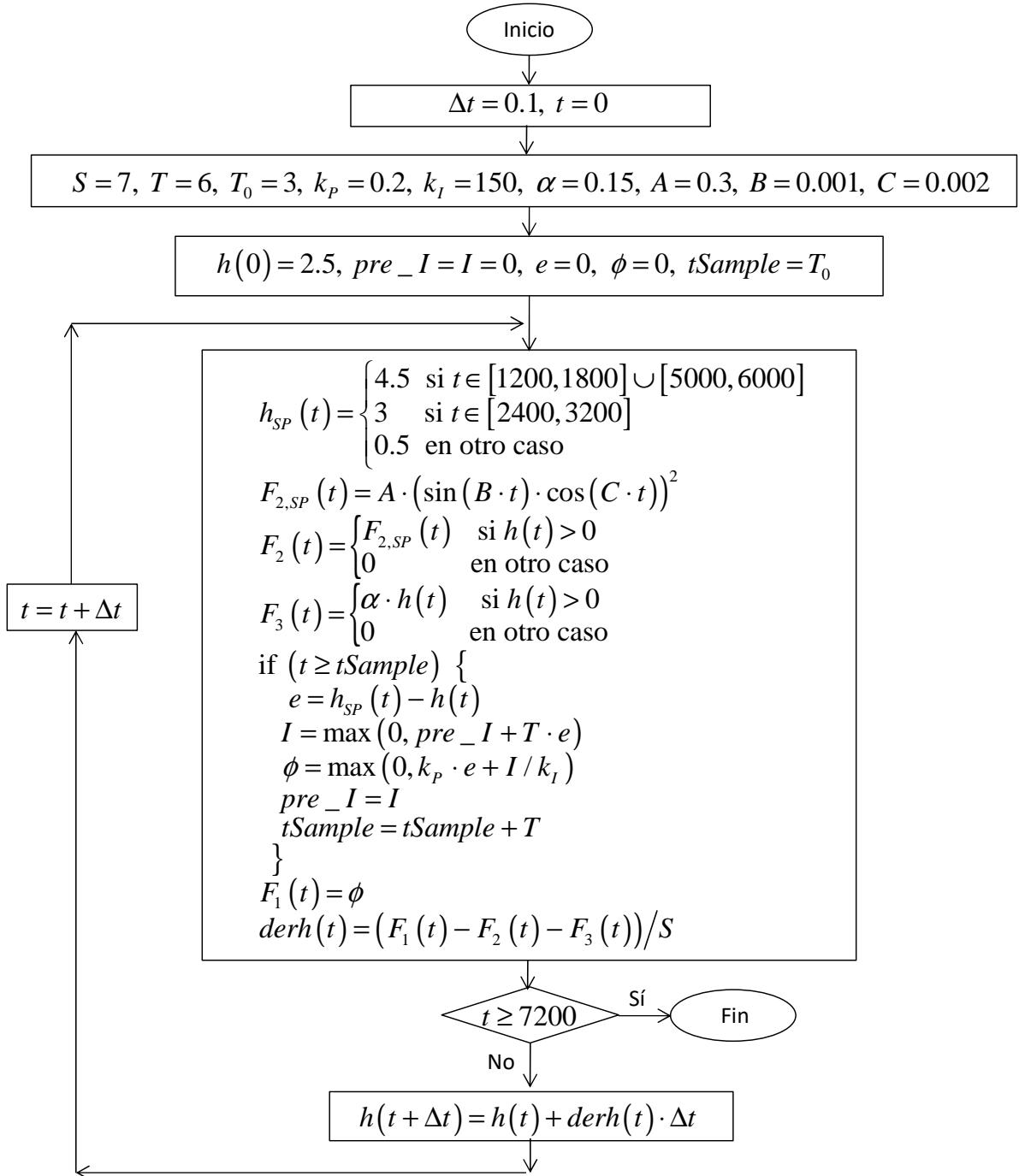
Aplicando el algoritmo de la partición, se obtiene el siguiente modelo ordenado y resuelto (existen otras posibles formas de ordenar el modelo que son igualmente válidas), con la causalidad computacional señalada.

$$\begin{aligned}
 h & \quad \text{Variable de estado} \\
 [h_{SP}] &= \begin{aligned} & \text{if } t \geq 1200 \text{ and } t \leq 1800 \text{ or } t \geq 5000 \text{ and } t \leq 6000 \\ & \text{then } 4.5 \\ & \text{else if } t \geq 2400 \text{ and } t \leq 3200 \text{ then } 3 \text{ else } 0.5 \end{aligned} \\
 [F_{2,SP}] &= A \cdot (\sin(B \cdot t) \cdot \cos(C \cdot t))^2 \\
 [F_2] &= \text{if } h > 0 \text{ then } F_{2,SP} \text{ else } 0 \\
 [F_3] &= \text{if } h > 0 \text{ then } \alpha \cdot h \text{ else } 0 \\
 [derh] &= \frac{F_1 - F_2 - F_3}{S}
 \end{aligned}$$

En la Figura 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 7200 s.

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.

Se muestra también el código R que implementa el algoritmo mostrado en la Figura 3, con un intervalo de comunicación igual al tamaño del paso de integración. En la Figura 4 se muestran las gráficas obtenida al ejecutar el código R.

**Figura 3:** Diagrama de flujo de la simulación del modelo.

Código R del algoritmo de la simulación

```

1 rm( list = ls() )
2 tiempoFin    <- 7200      # Valor final del tiempo
3 incTiempo    <- 0.1       # Paso de integracion
4
5 # Parametros
6 S            <- 7
7 T            <- 6
8 T_0          <- 3
9 k_P          <- 0.2
10 k_I         <- 150
11 alfa        <- 0.15
12 A            <- 0.3
13 B            <- 0.001
14 C            <- 0.002
15
16 # Vector que contiene los instantes de tiempo simulado
17 tiempo <- seq(0, tiempoFin, by = incTiempo)
18
19 # Inicializacion vectores que almacenan las variables
20 nPoints   <- length(tiempo)
21 h          <- numeric(nPoints)
22 h_SP       <- numeric(nPoints)
23 F_1        <- numeric(nPoints)
24 F_2SP      <- numeric(nPoints)
25 F_2        <- numeric(nPoints)
26 F_3        <- numeric(nPoints)
27 derh       <- numeric(nPoints)
28
29 # Condicion clausula when
30 tSample    <- T_0
31
32 # Valor inicial variable de estado de tiempo continuo
33 h[1]       <- 2.5
34
35 # Valor inicial variables de tiempo discreto
36 e          <- 0
37 I          <- 0
38 phi        <- 0
39 pre_I      <- 0
40
41 # Bucle de la simulacion
42 for ( nStep in c(1:nPoints) ) {
43
44     if ( ( tiempo[nStep] >= 1200 && tiempo[nStep] <= 1800 ) ||
45         ( tiempo[nStep] >= 5000 && tiempo[nStep] <= 6000 ) ) {
46         h_SP[nStep] <- 4.5
47     } else if ( tiempo[nStep] >= 2400 && tiempo[nStep] <= 3200 ) {
48         h_SP[nStep] <- 3
49     } else {
50         h_SP[nStep] <- 0.5
51     }
52

```

```

53 F_2SP[nStep] <- A*(sin(B*tiempo[nStep])*cos(C*tiempo[nStep]))^2
54
55 if ( h[nStep] > 0 ) {
56   F_2[nStep] <- F_2SP[nStep]
57 } else {
58   F_2[nStep] <- 0;
59 }
60
61 if ( h[nStep] > 0 ) {
62   F_3[nStep] <- alfa * h[nStep]
63 } else {
64   F_3[nStep] <- 0;
65 }
66
67 if ( tiempo[nStep] >= tSample ) {
68   e       <- h_SP[nStep] - h[nStep]
69   I       <- max( c(0, pre_I + T*e) )
70   phi     <- max( c(0, k_P*e + I/k_I) )
71   pre_I   <- I
72   tSample <- tSample + T
73 }
74
75 F_1[nStep] <- phi
76 derh[nStep] <- ( F_1[nStep] - F_2[nStep] - F_3[nStep] ) / s
77
78 # Calculo en el siguiente instante de la variable de estado
79 if ( nStep < nPoints ) {
80   h[nStep+1] <- h[nStep] + incTiempo * derh[nStep]
81 }
82
83 } # fin del bucle de la simulacion
84
85 # Representacion grafica
86 dev.new()
87 plot(tiempo, h_SP, xlab = "t [s]", ylab = "[m]",
88       type="l", col="blue", lwd=1.5, ylim=c(0,5),
89       panel.first = abline(v = seq(0,tiempoFin,by=1000),
90                             h = seq(0,5,by=0.1),
91                             lwd = 0.5, lty = 3, col="grey") )
92 points(tiempo, h, type="l", col="red", lwd=1.5)
93 legend("topright", legend=c("h_SP", "h"),
94        col=c("blue", "red"), lty=c(1,1), cex=0.8)
95
96 dev.new()
97 plot(tiempo, F_1, xlab = "t [s]", ylab = "[m3/s]",
98       type="l", col="blue", lwd=1.5, ylim=c(0,1.4),
99       panel.first = abline(v = seq(0,tiempoFin,by=1000),
100                            h = seq(0,1.4,by=0.05),
101                            lwd = 0.5, lty = 3, col="grey") )
102 points(tiempo, F_2, type="l", col="red", lwd=1.5)
103 points(tiempo, F_3, type="l", col="darkgreen", lwd=1.5)
104 legend("topright", legend=c("F_1", "F_2", "F_3"),
105        col=c("blue", "red", "darkgreen"), lty=c(1,1,1), cex=0.8)

```

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

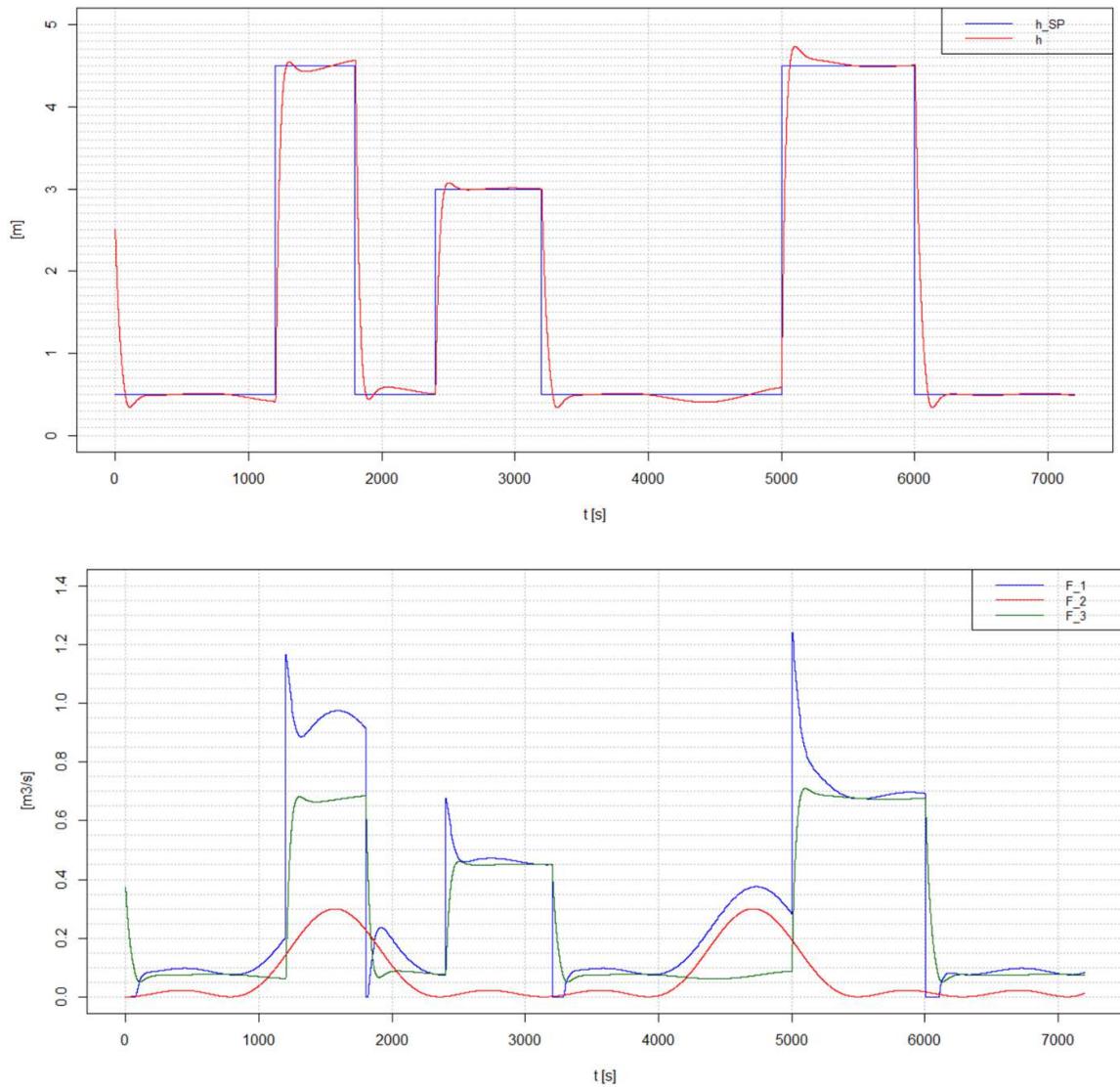


Figura 4: Resultado de la ejecución del código R.

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. 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 el modelo atómico y el modelo compuesto en Modelica durante 7200 s y compruebe que obtiene los mismos resultados.

Represente frente al tiempo las siguientes variables del modelo atómico en Modelica: en una gráfica las variables F_1 , F_2 y F_3 , y en una segunda gráfica las variables h y h_{SP} . Compare estos resultados con los que obtuvo al contestar al Ejercicio 2.4 (ejecución del algoritmo programado en lenguaje R).

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 5.

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 SistDeposito
2     import      Modelica.Units.SI;
3     parameter SI.Area   S           = 7;
4     parameter SI.Time   T           = 6;
5     parameter SI.Time   T0          = 3;
6     parameter Real     kP(unit="m2/s") = 0.2;
7     parameter Real     kI(unit="s2/m2") = 150;
8     parameter Real     alfa(unit="m2/s") = 0.15;
9     parameter Real     A(unit="m3/s")   = 0.3;
10    parameter Real    B(unit="s-1")    = 0.001;
11    parameter Real    C(unit="s-1")    = 0.002;
12    SI.VolumeFlowRate F1, F2, F3, F2sp;
13    SI.Height h(start=2.5, fixed=true);
14    SI.Height hSP;
15    SI.Length e(start=0, fixed=true);
16    Real I(unit="m.s", start=0, fixed=true);
17    Real phi(unit="1");
18 equation
19     // Balance en el deposito
20     S * der(h) = F1 - F2 - F3;
21     // Bomba superior
22     F1 = phi;
23     // Entradas al controlador
24     hSP = if time >= 1200 and time <= 1800 or
25         time >= 5000 and time <= 6000
26         then 4.5
27         else if time >= 2400 and time <= 3200
28             then 3
29             else 0.5;
30     // Bomba inferior
31     F2sp = A * ( sin(B*time) * cos(C*time) )^2;
32     F2 = if h > 0 then F2sp else 0;
33     // Caudal a traves valvula
34     F3 = if h > 0 then alfa * h else 0;
35     // Controlador
36     when sample(T0, T) then
37         e = hSP - h;
38         I = max(0, pre(I) + T*e);
39         phi = max(0, kP*e + I/kI);
40     end when;
41 end SistDeposito;
```

Código 2: Modelo atómico en Modelica.

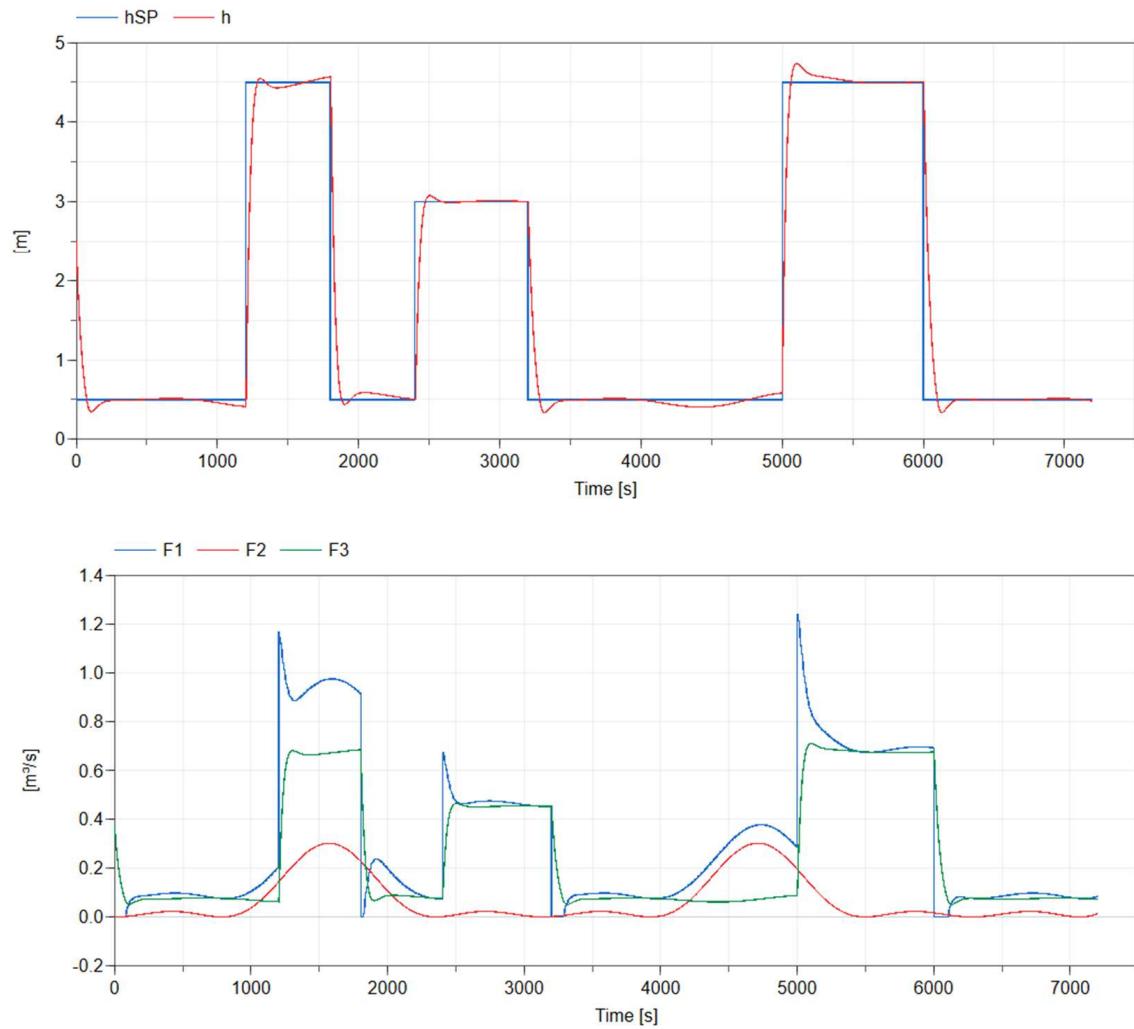


Figura 5: Resultado de la ejecución del código Modelica.

```

1 package SistDepositoLib
2
3     import      Modelica.Units.SI;
4
5 package Interfaz
6
7     connector PortLiq
8         SI.Height h;
9         flow SI.VolumeFlowRate F;
10    end PortLiq;
11
12    connector PortSenal
13        Real s;
14    end PortSenal;
15
16    partial model Bomba
17        PortLiq portLiq;
18        PortSenal consigna;
19    end Bomba;
20
21 end Interfaz;
22
23 package Componentes
24
25     model Deposito
26         Interfaz.PortLiq portLiq;
27         Interfaz.PortSenal sensorNivel;
28         parameter SI.Area S;
29         parameter SI.Height h_inicial;
30         SI.Height h(start=h_inicial, fixed=true);
31     equation
32         // Balance en el deposito
33         S * der(h) = portLiq.F;
34         portLiq.h = h;
35         sensorNivel.s = h;
36     end Deposito;
37
38     model Valvula
39         Interfaz.PortLiq portLiq;
40         parameter Real alfa(unit="m2/s");
41     equation
42         portLiq.F = if portLiq.h > 0 then alfa*portLiq.h else 0;
43     end Valvula;
44
45     model BombaInf
46         extends Interfaz.Bomba;
47     equation
48         portLiq.F = if portLiq.h > 0 then consigna.s else 0;
49     end BombaInf;
50
51     model BombaSup
52         extends Interfaz.Bomba;
53     equation

```

```

54     portLiq.F = -consigna.s;
55 end BombaSup;

56
57 model Controlador_PI
58     Interfaz.PortSenal senal_salida;
59     Interfaz.PortSenal senal_nivel;
60     parameter SI.Time    T;
61     parameter SI.Time    T0;
62     parameter Real       kP(unit="m2/s");
63     parameter Real       kI(unit="s2/m2");
64     input SI.Height hSP;
65 protected
66     SI.Height h;
67     SI.Length e(start=0, fixed=true);
68     Real I(unit="m.s", start=0, fixed=true);
69     Real phi(unit="1");
70 equation
71     when sample(T0, T) then
72         e = hSP - h;
73         I = max(0, pre(I) + T*e);
74         phi = max(0, kP*e + I/kI);
75     end when;
76     senal_salida.s = phi;
77     senal_nivel.s = h;
78 end Controlador_PI;

79
80 end Componentes;

81
82 model SistemaCompleto
83
84     Componentes.Controlador_PI PI(T=T, T0=T0, kP=kP, kI=kI, hSP=hSP);
85     Componentes.BombaSup BombaSup;
86     Componentes.BombaInf BombaInf;
87     Componentes.Valvula Valvula(alfa=alfa);
88     Componentes.Deposito Deposito(S=S, h_inicial=h_inicial);
89
90     parameter SI.Area    S          = 7;
91     parameter SI.Time    T          = 6;
92     parameter SI.Time    T0         = 3;
93     parameter Real      kP(unit="m2/s") = 0.2;
94     parameter Real      kI(unit="s2/m2") = 150;
95     parameter Real      alfa(unit="m2/s") = 0.15;
96     parameter Real      A(unit="m3/s")   = 0.3;
97     parameter Real      B(unit="s-1")    = 0.001;
98     parameter Real      C(unit="s-1")    = 0.002;
99     parameter SI.Height h_inicial   = 2.5;
100
101    SI.Height hSP;
102
103 equation
104
105     connect (PI.senal_salida, BombaSup.consigna);
106     connect (Deposito.portLiq, BombaSup.portLiq);
107     connect (Deposito.portLiq, BombaInf.portLiq);

```

```
108 connect (Deposito.portLiq, Valvula.portLiq);
109 connect (Deposito.sensorNivel, PI.senal_nivel);
110
111 hSP = if time >= 1200 and time <= 1800 or
112     time >= 5000 and time <= 6000
113     then 4.5
114     else if time >= 2400 and time <= 3200
115         then 3
116         else 0.5;
117 BombaInf.consigna.s = A * ( sin(B*time) * cos(C*time) )^2;
118
119 end SistemaCompleto;
120
121 end SistDepositoLib;
```

Código 3: Librería y modelo compuesto en Modelica.

Ejercicio 4

Escriba un script en FlexPDE para determinar el potencial y el campo gravitatorio existentes entre la Luna y la Tierra. Para realizar el estudio, se consideran sólo dos dimensiones, tal como se muestra en la Figura 6. En esta figura la Tierra está en la posición $(0, 0)$ y la Luna está en la posición $(0, d)$, siendo d la distancia entre la Luna y la Tierra. En el estudio se analiza únicamente el cuadrado de lado $2 \cdot d$ que se muestra en la figura.

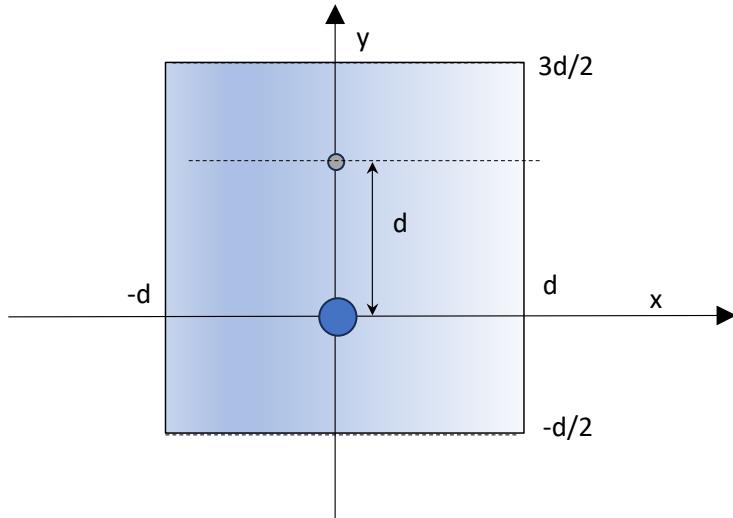


Figura 6: Sistema Tierra-Luna.

El potencial gravitatorio (U) en un punto (x, y) del sistema de coordenadas de la figura anterior, donde la Tierra está en la posición $(0, 0)$ y la Luna en la posición $(0, d)$, es el siguiente:

$$U = -G \cdot \frac{m_T}{\sqrt{x^2 + y^2}} - G \cdot \frac{m_L}{\sqrt{x^2 + (y - d)^2}} \quad (1.21)$$

Los datos del sistema Tierra-Luna en unidades del sistema internacional (SI) son los siguientes: distancia Tierra-Luna (d) es $3.84 \cdot 10^8$, la masa de la Tierra (m_T) es $5.98 \cdot 10^{24}$, la masa de la Luna (m_L) es $7.35 \cdot 10^{22}$. La constante de gravitación universal (G) en unidades del SI tiene el valor $6.67 \cdot 10^{-11}$.

Observa que el potencial sólo depende de las coordenadas cartesianas y y de parámetros, no existiendo otras variables. Por tanto, el script de FlexPDE solución de este ejercicio no tiene ni sección VARIABLES ni sección EQUATIONS y tanto los

parámetros como las ecuaciones del sistema se pueden definir dentro de la sección DEFINITIONS.

El campo gravitatorio se obtiene aplicando el gradiente. A continuación se muestran las ecuaciones para obtener las componentes x e y del campo gravitatorio (g_x y g_y), así como su módulo (g_m).

$$g_x = -\frac{\partial U}{\partial x} \quad (1.22)$$

$$g_y = -\frac{\partial U}{\partial y} \quad (1.23)$$

$$g_m = \sqrt{g_x^2 + g_y^2} \quad (1.24)$$

Escriba un *script* empleando FlexPDE para obtener un mapa de contorno de la distribución del menor potencial en escala logarítmica y un gráfico vectorial del campo gravitatorio dividido por el módulo de dicho vector en la región rectangular bajo estudio. Para mostrar estos dos gráficos, tiene que usar el siguiente código en FlexPDE, donde U es el potencial gravitatorio, gv es el vector del campo gravitatorio y gm es el módulo de este vector:

```
contour(-U) log
vector(gv/gm)
```

Solución al Ejercicio 4

El *script* se muestra en el Código 4. Los gráficos de contorno y vectorial se muestran en las Figuras 7–8.

```

1 TITLE 'Campo gravitacional Tierra-Luna'
2 COORDINATES cartesian2
3 SELECT
4   spectral_colors
5 DEFINITIONS
6   d=3.84e8 Lx=d Ly=d
7   mT=5.98e24 mL=7.35e22
8   G=6.67e-11
9   U=-G*mT/sqrt(x^2+y^2)-G*mL/sqrt(x^2+(y-d)^2)
10  gx=-dx(U) gy=-dy(U)
11  gv=vector(gx,gy)
12  gm=sqrt(gx^2+gy^2)
13 BOUNDARIES
14   REGION 'dominio'
15     START(-Lx,-Ly/2)
16     LINE TO (Lx,-Ly/2) TO (Lx,3/2*Ly) TO (-Lx,3/2*Ly) TO CLOSE
17 PLOTS
18   CONTOUR(-u) log
19   vector(gv/gm)
20 END

```

Código 4: Modelo descrito en FlexPDE.

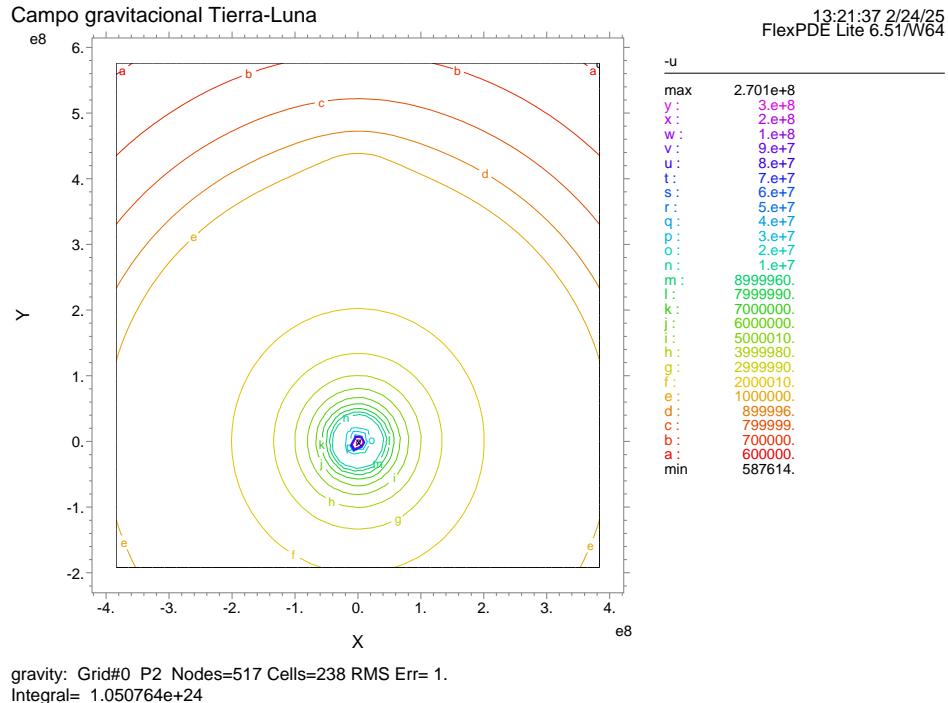


Figura 7: Curvas del potencial gravitacional.

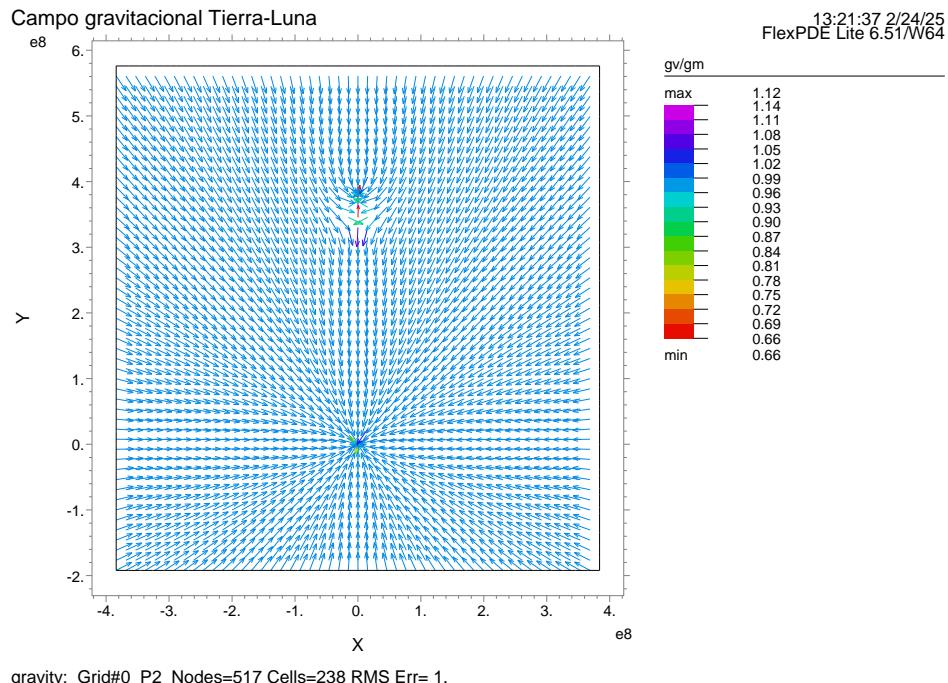


Figura 8: Vectores del campo gravitacional.