MAT

Métodos numéricos para ecuaciones diferenciales ordinarias

Métodos numéricos

Marilina Carena
Anibal Chicco Ruiz
Lucas Genzelis
Elisabet Haye

ediciones UNL

Métodos numéricos para ecuaciones diferenciales ordinarias

Métodos numéricos para ecuaciones diferenciales ordinarias

Marilina Carena
Anibal Chicco Ruiz
Lucas Genzelis
Elisabet Haye

ediciones UNL

CÁTEDRA

UNIVERSIDAD NACIONAL DEL LITORAL

ediciones UNL

Consejo Asesor
Colección Cátedra
Alicia Camilloni
Daniel Comba
Bárbara Mántaras
Isabel Molinas
Héctor Odetti
Andrea Pacífico
Ivana Tosti

Dirección editorial
Ivana Tosti
Coordinación editorial
María Alejandra Sedrán
Coordinación comercial
José Díaz

Corrección
Ediciones UNL
Diagramación interior
Marilina Carena y Ediciones UNL
Diseño de tapa
Verónica Rainaudo

—

Sugerencias y comentarios
editorial@unl.edu.ar
www.unl.edu.ar/editorial

Métodos numéricos para ecuaciones diferenciales ordinarias /
Marilina Carena... [et al.]. – 1ra. ed. –
Santa Fe : Ediciones UNL, 2026.
Libro digital, PDF/A – (Cátedra)

Archivo Digital: descarga y online
ISBN 978–987–749–544-7

1. Matemática. 2. Números. 3. Ecuaciones.
I. Carena, Marilina
CDD 540

———————————

Índice general

Presentación

Aun cuando se pueda demostrar la existencia de soluciones para una ecuación diferencial, no siempre es posible expresarlas de forma explícita, e incluso, a veces, ni siquiera de manera implícita. Un ejemplo clásico es la ecuación diferencial de Airy:

$$y'' - xy = 0,$$

que tiene un aspecto muy sencillo, pero cuyas soluciones no pueden escribirse en términos de funciones elementales. Esta ecuación aparece en diversos modelos físicos, como veremos más adelante.

Como ocurre con la ecuación de Airy, en la mayoría de los casos prácticos debemos recurrir a aproximaciones que representen la solución en una lista de valores. Afortunadamente, es posible obtener resultados con tanta precisión como se desee mediante métodos numéricos. Aunque no se trate de la solución exacta, la estimación obtenida suele ser suficientemente buena para fines prácticos.

Los métodos numéricos que permiten obtener soluciones aproximadas de ecuaciones diferenciales existen desde el siglo XVIII, pero fueron adquiriendo mayor relevancia con el avance de la tecnología. En este texto se abordan algunos de los métodos más destacados, seleccionados por su importancia histórica y conceptual, su utilidad en diversas aplicaciones y contextos de enseñanza, y su adecuación al tiempo con que se cuenta habitualmente en una asignatura sobre ecuaciones diferenciales ordinarias orientada a ingenierías. Muchos resultados se enunciarán sin demostración, pero se proporcionarán referencias bibliográficas para quienes deseen profundizar en estos temas. Asimismo, no abordaremos aproximaciones en intervalos donde la solución no exista o no sea única.

>_ Este símbolo indica que el ejercicio o trabajo deberá realizarse con ayuda de la computadora. Hemos elegido Octave como herramienta para implementar los métodos desarrollados, aunque puede utilizarse cualquier otro programa o lenguaje, si se prefiere*. No se requieren conocimientos avanzados de Octave para seguir este texto, ya que se presentará progresivamente todo lo necesario. Para comenzar, basta con visitar https://octave.org/ para descargarlo e instalarlo. Además, algunos aspectos básicos se incluyen en el Apéndice.

El contenido de este libro tiene aplicaciones prácticas de gran relevancia en el ejercicio profesional. Por ello, es fundamental que los estudiantes no se limiten a la aplicación directa de fórmulas o a la utilización sistemática de los métodos presentados, sino que desarrollen su capacidad de razonamiento. Esto les permitirá adaptar lo aprendido a una amplia variedad de situaciones que puedan surgir en su futuro profesional.

*Los códigos presentados en este texto pueden adaptarse fácilmente al lenguaje Python.

Con este propósito, los ejercicios incluidos no solo buscan aplicar y consolidar lo presentado en el texto, sino también fomentar la integración de conceptos y el descubrimiento de nuevas herramientas. Esto promueve una autonomía progresiva en el proceso de aprendizaje.

Dado que muchos ejercicios complementan el desarrollo teórico–práctico del texto, hemos incluido un solucionario completo con explicaciones detalladas. No obstante, recomendamos utilizarlo únicamente después de intentar resolver los ejercicios, para comparar resultados y aprovechar los comentarios adicionales.

Capítulo 1

Problemas de primer orden con valores iniciales

En este capítulo trataremos de aproximar de forma numérica la solución a un problema de valores iniciales (PVI) de la forma

$$\begin{cases} y'(x) &= f(x,y), \\ y(x_0) &= y_0, \end{cases} \tag{1.1}$$

para $x \in [x_0, x_F]$. Esto será de gran utilidad en todos aquellos casos donde el PVI no pueda resolverse explícitamente mediante los métodos trabajados, o donde resulte complicado hacerlo.

Recordemos que contamos con un teorema que nos da condiciones suficientes para garantizar la existencia y unicidad de solución para un PVI como el anterior. Trabajaremos exclusivamente con ecuaciones diferenciales que satisfagan las hipótesis de dicho teorema.

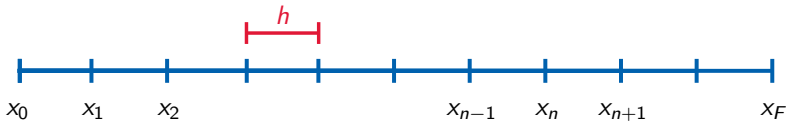
La idea detrás de los métodos numéricos es considerar una lista de puntos x_n en $[x_0, x_F]$, y hallar valores y_n que sean aproximaciones suficientemente buenas para los resultados reales $y(x_n)$, siendo y la solución de (1.1). Esto es, $y_n \approx y(x_n)$.

1.1. Método de Euler

Lo primero que se hace es particionar el intervalo $[x_0, x_F]$ en N subintervalos de igual longitud $h = (x_F - x_0)/N$. Este valor h se conoce como **tamaño de paso** para el método. Se genera entonces la lista de valores

$$x_0, \qquad x_1 = x_0 + h, \qquad x_2 = x_0 + 2h, \qquad x_3 = x_0 + 3h, \qquad \ldots \qquad x_F.$$

De forma general, se tiene que $x_n = x_0 + nh$, para $n = 0, 1, \ldots, N$.



Debido a la condición inicial del PVI, conocemos exactamente el valor de $y(x_0)$, dado por y_0. El objetivo es, entonces, construir aproximaciones

$$y_1, \quad y_2, \quad y_3, \quad \ldots \quad y_F$$

a los verdaderos valores

$$y(x_1), \quad y(x_2), \quad y(x_3), \quad \ldots \quad y(x_F)$$

de la solución de (1.1). Para ello usaremos la función pendiente $f(x, y)$. Más precisamente, cuando $x = x_0$, la velocidad de cambio de y con respecto a x es $y'(x_0) = f(x_0, y_0)$. Podemos obtener y_1, la aproximación para $y(x_1)$, suponiendo que y continúa cambiando con esa velocidad hasta $x_1 = x_0 + h$. Esto es,

$$y_1 = y_0 + h \cdot f(x_0, y_0).$$

Esto puede observarse en la Figura 1.1, donde L representa la recta tangente a la gráfica de y por el punto (x_0, y_0). Puesto que su pendiente es $m = y'(x_0) = f(x_0, y_0)$, tenemos que su ecuación es

$$L(x) = f(x_0, y_0) \cdot (x - x_0) + y_0.$$

Así, $y(x_1) \approx L(x_1) = f(x_0, y_0) \cdot (x_1 - x_0) + y_0 = f(x_0, y_0) \cdot h + y_0$, que es la expresión dada para y_1.

El proceso se repite siguiendo la misma idea: tomamos

$$y_2 = y_1 + h \cdot f(x_1, y_1)$$

como aproximación a $y(x_2)$ y, una vez calculado el valor aproximado y_n para $y(x_n)$, calculamos

$$y_{n+1} = y_n + h \cdot f(x_n, y_n).$$

Notar que a partir del primer paso ya no se trata de una tangente real, ya que (x_1, y_1) está sobre la primera tangente y no sobre la curva solución.

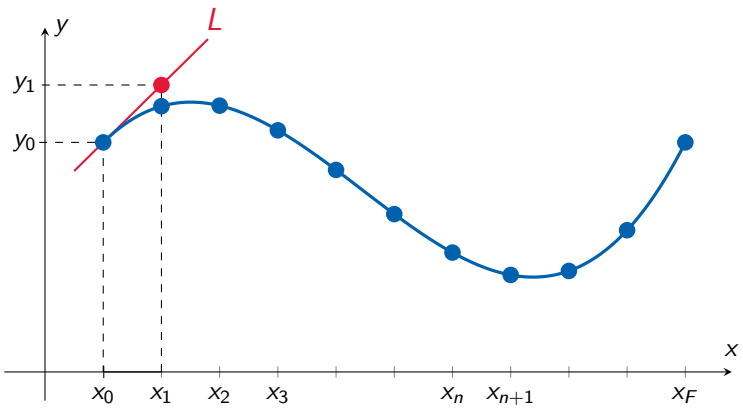


Figura 1.1: Idea geométrica del método de Euler.

Método de Euler: el algoritmo

Dado el problema con condiciones iniciales

$$y'(x) = f(x, y), \qquad y(x_0) = y_0,$$

el **método de Euler** con tamaño de paso h consiste en aplicar la fórmula iterativa

$$y_{n+1} = y_n + h \cdot f(x_n, y_n) \qquad (n \geq 0) \tag{1.2}$$

para calcular las aproximaciones sucesivas y_i de los valores exactos $y(x_i)$.

Aunque las aplicaciones importantes del método de Euler son en ecuaciones donde no sabemos hallar la solución exacta, ilustraremos el método con una que sí sabemos resolver, para poder comparar el resultado del mismo con el valor exacto.

Ejemplo 1. Aplique el método de Euler con tamaño de paso $h = 0.1$ para aproximar la solución en $[0, 1]$ del PVI dado por

$$y'(x) = x + y, \qquad y(0) = 1.$$

Solución. Al dividir el intervalo $[0, 1]$ en subintervalos de longitud $h = 0.1$, la lista de puntos x_i generada es

$$x_0 = 0, \quad x_1 = 0.1, \quad x_2 = 0.2, \quad \ldots \quad x_n = 0.1 \cdot n, \quad \ldots \quad x_{10} = 1.$$

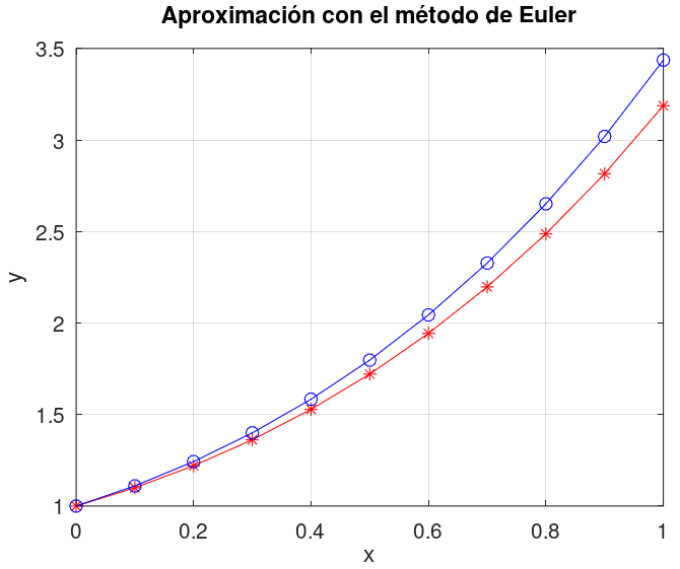
Por la condición inicial tenemos $y_0 = 1$. Además, para $n \geq 0$ la fórmula (1.2) nos arroja, luego de redondear a 4 cifras decimales, los siguientes resultados:

$$y_1 = 1 + 0.1 \cdot (0 + 1) = 1.1,$$
$$y_2 = 1.1 + 0.1 \cdot (0.1 + 1.1) = 1.22,$$
$$y_3 = 1.22 + 0.1 \cdot (0.2 + 1.22) = 1.362.$$

Siguiendo de esta manera se obtiene la siguiente tabla de valores, que fue elaborada implementando el Código 1.1 en Octave, incluido a continuación, con explicaciones para iniciar el recorrido.

x_i	0	0.1	0.2	0.3	0.4	0.5
y_i	1	1.1	1.22	1.362	1.5282	1.721
x_i	0.6	0.7	0.8	0.9	1	
y_i	1.9431	2.1974	2.4872	2.8159	3.1875	

El código también arroja el gráfico de la función obtenida al unir de forma lineal los puntos (x_i, y_i) (en rojo con $*$). En la imagen siguiente hemos incluido, además, el correspondiente gráfico para los puntos $(x_i, y(x_i))$ (en color azul con $\circ$), siendo $y(x) = 2e^x - x - 1$ la solución exacta del PVI, para comparar los resultados.



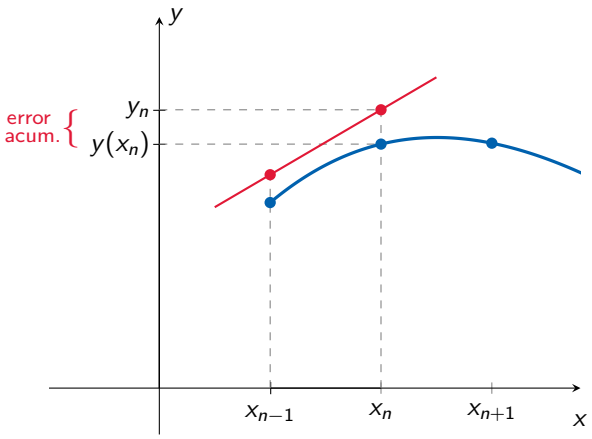
```
% Definir la función f(x,y)=dy/dx
f=@(x,y) x+y;

% Condiciones iniciales
x0=0;
y0=1;
h=0.1;  % Tamaño del paso
N=10;   % Número de iteraciones

% Inicializar los vectores para almacenar los resultados
x=zeros(1,N+1);
y=zeros(1,N+1);

% Asignar los valores iniciales
x(1)=x0;
y(1)=y0;

% Método de Euler
for n=1:N
    y(n+1)=y(n)+h*f(x(n),y(n));
    x(n+1)=x(n)+h;
end

% Mostrar resultados
disp("Valores de x:");
disp(x);
disp("Valores de y:");
disp(y);

% Graficar la aproximación
plot(x,y,'-r*');
xlabel('x');
ylabel('y');
title('Aproximación con el método de Euler');
grid on;
```

Código 1.1: Método de Euler en Octave.

El error cometido

Es claro que la precisión de la aproximación $L(x_1) \approx y(x_1)$ depende fuertemente del tamaño del incremento h, y que resultará mejor mientras más pequeño sea h. Esto ocurre en cada paso: suponiendo que y_{n-1} es la solución exacta, y_n tendrá un **error de truncamiento local** producido por la linealización del método. Pero como mencionamos antes, luego del primer paso el método ya no involucra una tangente a la curva solución, sino a la que pasa por el punto obtenido en el paso previo, cuya ordenada es una aproximación al valor real. Más precisamente, para calcular y_n, el punto de partida (x_{n-1}, y_{n-1}) no está, en general, sobre la curva solución, puesto que y_{n-1} es una aproximación para $y(x_{n-1})$. Entonces, aun cuando h sea pequeño, en cada paso se comete un error que se **acumula** con los anteriores. Así, y_n sufre

los efectos acumulados de todos los errores introducidos en los pasos previos. El **error acumulado** en el paso n del método de Euler es la diferencia entre el valor real y el arrojado por el método: $|y(x_n) - y_n|$.



Se llama **error global o absoluto** al máximo de los errores acumulados:

$$\text{error global} = \max_{1 \le n \le N} |y(x_n) - y_n| \ .$$

Teorema 1 (Error en el método de Euler con paso $h > 0$)**.** Supongamos que el problema con condiciones iniciales

$$y'(x) = f(x, y), \qquad y(x_0) = y_0,$$

tiene una solución única en el intervalo cerrado $[x_0, x_F]$, y que esta función tiene derivada segunda continua allí. Entonces existe una constante C tal que

$$|y(x_n) - y_n| \le Ch, \qquad \text{para todo } n = 1, 2, \dots, N.$$

Para analizar los errores que surgen del uso de métodos numéricos se denota con $e(h)$ al error en un cálculo numérico que depende de h. Se dice que $e(h)$ es de **orden** h^n, y se denota $\mathrm{e}(h) = O(h^n)$, si $|e(h)| \le Ch^n$ para h suficientemente pequeña[1]. El teorema anterior, cuya demostración omitiremos[2], establece que *el error global con el método de Euler con paso h es de orden h, y se escribe* $\mathrm{e}(h) = O(h)$. Por supuesto, estamos excluyendo los errores de redondeo.

[1] Como se considera h pequeño, un método de orden h^2 proporciona **mayor precisión** que uno de orden h.

[2] Usando el teorema de Taylor, puede demostrarse que $|L(x_1) - y(x_1)| \le Ch^2$. Como mencionamos, esta cota solo se cumple en el primer paso, y el error acumulado en los pasos sucesivos puede reducir la precisión global del método.

Esto nos dice que si, por ejemplo, dividimos el paso en la mitad, el error se reducirá también a la mitad: $e(\frac{h}{2}) \simeq \frac{1}{2} e(h)$. Para ilustrar, si el error con un paso $h = 0.1$ es de aproximadamente 0.02, al reducir el paso a $h = 0.05$ el error disminuirá a 0.01. Así, podemos obtener (en principio) cualquier grado de exactitud que deseemos eligiendo h suficientemente chico. Sin embargo, disminuir el valor de h equivale a aumentar la cantidad de puntos x_n de la lista, lo que requiere más tiempo de cálculo. Además, la computadora aporta errores de redondeo en cada etapa, por lo que a veces valores de h muy pequeños pueden ser un problema. Por otro lado, si h es demasiado grande, las aproximaciones del método de Euler no son muy buenas.

Así, la elección adecuada de h es difícil de determinar tanto en la práctica como en la teoría, ya que depende de la función f, las condiciones iniciales, el código y la computadora. Se sugiere:

Elección del tamaño del paso

- Aplicar el método con un valor razonable para h.
- Repetir con $h/2$, $h/4$, y así, dividiendo en cada etapa el paso a la mitad.
- Continuar hasta que los resultados obtenidos en una etapa concuerden con los de la etapa anterior, con la precisión deseada.

Ejercicios Sección 1.1

Respuestas en página 71

1. Utilice el método de Euler "a mano" con $h = 0.1$ para obtener una aproximación para $y(x_F)$. Escriba su respuesta con 4 decimales.

   a) $y' = x + y^2$, $y(0) = 0$; $x_F = 0.2$.
   b) $y' = (x - y)^2$, $y(0) = 0.5$; $x_F = 0.5$.
   c) $y' = xy + \sqrt{y}$, $y(0) = 1$; $x_F = 0.5$.

2. Utilice Octave para aplicar el método de Euler con $h = 0.05$ en cada uno de los PVI del ejercicio anterior. *Precaución:* si desea fijar primero h y x_F, para luego definir $N = (x_F - x_0)/h$, tenga siempre cuidado de que esta cantidad resulte un número entero. Es conveniente definir h en términos del resto de los valores.

3. Calcule $y(1)$ redondeado a 4 cifras decimales, siendo y la solución exacta para el PVI del Ejemplo 1. Halle el error cometido al aproximar por el método de Euler de paso $h = 0.1$ en dicho ejemplo. Luego, utilice Octave para aplicar el método de Euler con $h = 0.05$ y calcule el error cometido. Compare e(0.1) con e(0.05) para ver si, salvo por errores de redondeo, coincide con lo esperado. El comando `y(end)` puede ser útil.

4. >_ El Código 1.1 fue armado de manera básica, siguiendo las ideas presentadas previamente. A continuación proponemos leves modificaciones a modo de ejercitación en Octave.

   a) Investigue la acción del comando `linspace`. Como se indica en el Apéndice, la instrucción `help nombre_comando` puede ayudar.

   b) ¿Cómo se puede modificar el Código 1.1 usando este comando?

   c) Transforme el *script* en una función

   ```
   [x,y]=Feuler(f,inter,y0,N)
   ```

   de modo que reciba como entradas a f, el intervalo $[x_0, x_F]$, el valor y_0 y la cantidad de subintervalos N, y devuelva la lista de valores x_i, y_i.

5. >_ **Aproximando números famosos.** La resolución numérica de los siguientes PVI nos permitirá obtener aproximaciones de números irracionales "famosos". En cada caso, use el algoritmo de Euler en Octave para calcular $y(x_F)$ con $N = 50$, $N = 100$, $N = 200$ y $N = 400$. Redondee a 4 cifras decimales. ¿Qué número se aproxima en cada PVI? Justifique.

   a) $y' = 4/(1+x^2)$, $y(0) = 0$; $x_F = 1$.

   b) $y' = y$, $y(0) = 1$; $x_F = 1$.

   c) $y' = 1/x$, $y(1) = 0$; $x_F = 2$.

6. >_ **Instrucciones útiles en Octave.** Aprovecharemos este problema, extraído de [3], para aprender algunas funciones de Octave que aplicaremos luego. Considere el PVI

   $$y' = -\cos(2\pi x^2) - xy, \qquad y(0) = 2, \qquad x \in [0, 4].$$

   a) Resuelva con el método de Euler. Utilice diferentes valores de N (por ejemplo, considere $N = 10, 20, 40, 80, 160, \ldots$) y grafique las aproximaciones de la solución que se van obteniendo en un mismo gráfico (el comando `hold on` puede ser útil para ello).

   b) Elija N tal que el tamaño de paso h sea 0.01. Para la solución numérica obtenida, calcule el máximo y el mínimo, y obtenga los valores de x donde estos se alcanzan. Para esto cuenta con la instrucción `[M,p]=max(y)` y su análogo para el mínimo.

   c) Encuentre el valor aproximado de x para el cual $y(x) = 1$. Para ello resulta útil el comando `find`. Pruebe con `find(y==1)`, `find(y<1)` y `find(y>1)` y obtenga sus conclusiones.

7. (De [3]) Considere el PVI formado por la ecuación autónoma

   $$y' = y(1-y)(1+y)$$

   y la condición inicial $y(0) = y_0$.

a) >_ Utilice el método de Euler con $N = 10000$ para resolver el PVI en $[0, 4]$, con $y_0 = -1.4, -1.3, -1.2, \ldots, 1.3, 1.4$. Superponga todas las soluciones en un mismo gráfico y explique el comportamiento cualitativo de las soluciones.

b) Justifique mediante conceptos teóricos lo observado en el gráfico obtenido.

8. >_ Veamos cómo lograr que Octave muestre todos los resultados con más cifras decimales de manera global. Comience escribiendo `pi` en la ventana de comandos y observe el resultado. Luego, ingrese la instrucción `format long` y vuelva a escribir `pi`. Compare los resultados. Ahora escriba `e^-4`, ingrese luego la instrucción `format long g` y vuelva a escribir `e^-4`. ¿Qué cambió? Para volver al formato por defecto, escriba `format short`.

9. >_ **Inestabilidad numérica.** Una ecuación diferencial ordinaria (o un sistema de ecuaciones) se considera **rígida** (*stiff*) cuando su resolución numérica requiere una disminución considerable del paso h para generar buenas aproximaciones de la solución exacta[3]. Para comprender eso, considere los siguientes PVI:

a) $y' = -2y + 2$, $y(0) = 2$; $x_F = 1$;

b) $y' = -20y + 20$, $y(0) = 2$; $x_F = 1$.

(i) Para cada uno de los PVI, halle la solución explícita y su valor en x_F redondeado a 4 cifras decimales.

(ii) Aplique el método de Euler con $N = 10$ y observe los resultados para cada caso, no solo para x_F, sino para todos los valores de la lista.

(iii) Para el PVI dado en *b*), repita con $N = 20$ y $N = 40$ y vuelva a observar lo obtenido para `y(N+1)` en cada caso (también puede usar la instrucción `y(end)`).

(iv) Muestre los valores `x(5)` e `y(5)` que arroja Octave al aplicar el método de Euler al PVI del inciso *b*) con $N = 40$. Compare con el valor obtenido mediante la solución explícita evaluada en el valor de x correspondiente.

(v) Siguiendo con el PVI del inciso *b*), incremente el valor de N hasta que $y(0.1)$ se aproxime con una precisión de 2 cifras decimales exactas. ¿Qué valor tiene h?

[3] Para la resolución numérica de ecuaciones diferenciales rígidas también se emplean métodos implícitos. Si bien estos métodos exceden el alcance de este material, se presentará su idea general en la página 17.

Sobre ecuaciones rígidas o sistemas rígidos. Las ecuaciones diferenciales rígidas suelen aparecer cuando intentamos describir situaciones en las que las cosas cambian a distintas velocidades. Por ejemplo, imaginemos un sistema en el que algo cambia muy rápido al principio, pero mucho más despacio después. Esto es frecuente en fenómenos físicos o químicos como los que mencionamos en los ejemplos siguientes:

Reacciones químicas. Algunos procesos de reacción involucran etapas con velocidades muy diferentes. Por ejemplo consideremos una del tipo $A \to B \to C$, con constantes de reacción k_1 y k_2, con k_1 mucho mayor que k_2. Esto es, una reacción rápida es seguida de otra más lenta. Esto genera rigidez en el sistema.

Circuitos eléctricos. Algunos circuitos tienen componentes (como resistencias, capacitadores e inductores) que reaccionan a diferentes velocidades. Por ejemplo, en un circuito RC (Resistencia–Capacitancia) con valores muy pequeños para R y muy grandes para C, la ecuación para q puede ser rígida.

Problemas de enfriamiento. Un modelo de enfriamiento o calentamiento también puede ser rígido si ocurre una transferencia de calor rápida en una etapa inicial, seguida de una etapa más lenta. Por ejemplo, cuando un objeto muy caliente es puesto en un ambiente frío, con constante de enfriamiento k muy grande.

1.2. Método de Euler mejorado

El método de Euler mejorado tendrá un error global del orden de h^2 pero se pagará un costo computacional por ello, ya que su fórmula involucra en cada paso el promedio de f en dos valores distintos, como explicaremos a continuación.

Supongamos que, como antes, queremos hallar una aproximación y_{n+1} habiendo calculado ya y_n. Comenzamos empleando el método de Euler para obtener una primera estimación

$$y^*_{n+1} = y_n + hf(x_n, y_n).$$

En la Figura 1.2, en donde para ilustrar hemos supuesto que (x_n, y_n) está sobre la curva solución, la recta L_n considerada en el método de Euler clásico tiene pendiente $k_1 = f(x_n, y_n)$, y lo análogo vale para L_{n+1}, cuya pendiente será entonces $k_2 = f(x_{n+1}, y^*_{n+1})$. El método de Euler mejorado consiste en tomar un promedio entre estas dos pendientes k_1 y k_2 para mejorar la estimación de la pendiente en todo el subintervalo $[x_n, x_{n+1}]$, obteniendo

$$y_{n+1} = y_n + h \cdot \frac{f(x_n, y_n) + f(x_{n+1}, y^*_{n+1})}{2}.$$

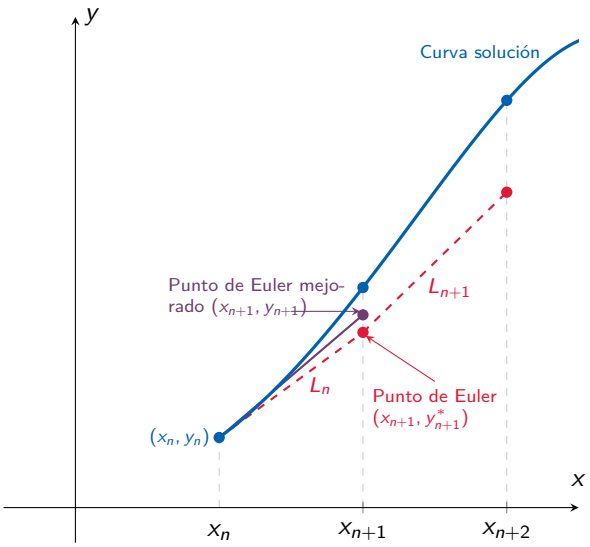


Figura 1.2: Idea geométrica del método de Euler mejorado.

El valor y^*_{n+1} se conoce como **predictor**, ya que permite predecir la pendiente k_2 que sigue en el método de Euler clásico. Al usar el promedio entre k_2 y la pendiente k_1 anterior, se realiza una corrección para ajustar el valor final del siguiente punto, por lo que y_{n+1} es llamado **corrector**. Por este motivo el método de Euler mejorado forma parte del grupo de técnicas numéricas conocidas como métodos predictor-corrector.

Método de Euler mejorado: el algoritmo y su error global

Dado el problema con condiciones iniciales

$$y'(x) = f(x, y), \qquad y(x_0) = y_0,$$

el **método de Euler mejorado** con tamaño de paso h consiste en aplicar la fórmula iterativa

$$y_{n+1} = y_n + h \cdot \frac{f(x_n, y_n) + f(x_{n+1}, y^*_{n+1})}{2}, \tag{1.3}$$

donde

$$y^*_{n+1} = y_n + hf(x_n, y_n), \tag{1.4}$$

para calcular las aproximaciones sucesivas y_i de los valores exactos $y(x_i)$. Puede probarse que si la solución exacta tiene derivada tercera continua en $[x_0, x_F]$, entonces $e(h) = O\left(h^2\right)$.

Ejercicios Sección 1.2

Respuestas en página 76

1. Sea y la solución al PVI

$$y'(x) = e^{-xy^2}, \qquad y(0) = 1.$$

Considerando una partición del intervalo $[0, 1]$ en 2 subintervalos, utilice a mano el método de Euler mejorado para aproximar $y(1)$. Escriba sus respuestas con 4 decimales.

2. Compare $e(\frac{h}{2})$ con $e(h)$ para el método de Euler mejorado. Ejemplifique suponiendo que $e(h) = 0.01$.

3. >_ Modifique el Código 1.1 para implementar en Octave el algoritmo de Euler mejorado mediante las fórmulas (1.3) y (1.4). Luego, ejecute con $h = 0.1$ para comprobar que los resultados arrojados, redondeados a 4 cifras decimales, sean:

x_i	0	0.1	0.2	0.3	0.4	0.5
y_i	1	1.11	1.2421	1.3985	1.5818	1.7949

x_i	0.6	0.7	0.8	0.9	1	
y_i	2.0409	2.3231	2.6456	3.0124	3.4282	

4. *a*) Usando la solución exacta del PVI del Ejemplo 1 y lo obtenido en el ejercicio anterior, halle el error cometido al aproximar $y(1)$ con el método de Euler mejorado con $h = 0.1$.

   b) >_ Ejecute el código del Ejercicio 3 ahora con $h = 0.05$. Calcule el error cometido respecto al valor exacto de $y(1)$, y compare con $e(0.1)$ y con lo esperado según el Ejercicio 2 (tenga presente errores de redondeo).

5. >_ **Aproximando números famosos.** Repita el Ejercicio 5 de la Sección 1.1 con el algoritmo de Euler mejorado, tomando $N = 10, 20, 40$ y 80. Compare los resultados con los anteriores.

1.3. Métodos de Runge–Kutta

Existen métodos de Runge–Kutta de diferentes órdenes. El más común y usado en la práctica es el de Runge–Kutta de cuarto orden (RK4), que logra un equilibrio entre precisión y eficiencia. A diferencia del método de Euler que usa solo la pendiente inicial en cada paso, o el Euler mejorado que utiliza dos, el método RK4 calcula cuatro pendientes en distintos puntos

intermedios, y luego hace un promedio ponderado entre ellas. Esto permite obtener una aproximación mucho más precisa sin reducir el tamaño de los pasos. En efecto, cuando la solución es suficientemente buena, el error global es del orden de h^4.

De forma general, estos métodos plantean el cálculo de la aproximación y_{n+1} habiendo calculado ya y_n, como

$$y_{n+1} = y_n + h\underbrace{(w_1k_1 + w_2k_2 + \cdots + w_mk_m)}_{\text{promedio ponderado}}. \tag{1.5}$$

Los pesos w_i del promedio ponderado son constantes que, en general, suman 1. Los valores k_i se calculan evaluando f en puntos seleccionados, y se definen recursivamente. El número m se llama **orden** del método. Bajo hipótesis adecuadas, el error global para el método de RK de orden m es $O(h^m)$. Notar que el método de Euler clásico es un caso particular con $m = 1$ (RK de primer orden, con $k_1 = f(x_n, y_n)$ y $w_1 = 1$), mientras que el de Euler mejorado es un método de RK de segundo orden ($m = 2$, $w_1 = w_2 = 1/2$, $k_1 = f(x_n, y_n)$ y $k_2 = f(x_n + h, y_n + hk_1)$), por lo que suele denotarse como RK2.

El promedio en (1.5) no es al azar, sino que los parámetros se eligen para que haya una relación con el polinomio de Taylor de grado m asociado a la solución.

El método clásico de Runge–Kutta es el de orden 4, denotado RK4, cuyo algoritmo emplea las siguientes fórmulas:

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \\ k_1 &= f(x_n, y_n), \\ k_2 &= f\left(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}hk_1\right), \\ k_3 &= f\left(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}hk_2\right), \\ k_4 &= f(x_n + h, y_n + hk_3). \end{aligned} \tag{1.6}$$

Recordemos que la parte más costosa desde el punto de vista computacional es la evaluación de f. Así, podemos decir que los métodos de Euler, RK2 y RK4 cuestan 1, 2 y 4 "operaciones" por paso de tiempo, respectivamente. Por lo tanto, para aplicar cada método con N pasos, la cantidad de evaluaciones de f requerida es:

Euler: N evaluaciones, RK2: $2N$ evaluaciones, RK4: $4N$ evaluaciones.

En este punto surge una pregunta natural: ¿cuál es la ganancia de usar un método de orden superior si es computacionalmente más costoso en cada paso

de tiempo? Para analizarlo consideremos el caso en que los errores de Euler, RK2 y RK4 son a lo sumo h, h^2 y h^4, respectivamente (constante $C = 1$ para la cota del error en todos los casos). En la siguiente tabla mostramos el costo y el error máximo de cada método para diferentes cantidades de pasos N en el intervalo $[0, 1]$.

N	Euler		RK2		RK4	
	costo	error	costo	error	costo	error
10	10	0.1	20	0.01	40	0.0001
20	20	0.05	40	0.0025	80	0.00000625
40	40	0.025	80	0.000625	160	0.000000390625
80	80	0.0125	160	0.00015625	320	$2.44140625 \cdot 10^{-8}$
100	100	0.01	200	0.0001	400	0.00000001

Como podemos observar hasta la penúltima fila, cada vez que duplicamos N el error por el método de Euler se reduce a la mitad, pero el error por RK2 se divide por 4 y el error por RK4 se divide por 16.

Así, por ejemplo, para asegurar un error máximo de 10^{-4}, pagaremos un costo de 40 operaciones con RK4, 200 con RK2, ¡y más de 10 000 operaciones con el método de Euler! Para determinar este valor, buscamos un número natural n tal que

$$\frac{0.1}{2^n} \leq 0.0001,$$

lo que implica $n \geq 10$. Así, debemos duplicar 10 veces el valor inicial $N = 10$, lo que arroja $N = 10\,240$. Por lo tanto, los métodos de orden superior resultan, en general, más eficientes. Esto podrá comprobarse en el Ejercicio 3.

Ejercicios Sección 1.3

Respuestas en página 76

1. Modifique el Código 1.1 para implementar en Octave el algoritmo de RK4 mediante las fórmulas dadas en (1.6). Ejecute con $h = 0.1$ para comprobar que los resultados arrojados, redondeados a 4 cifras decimales, sean los contenidos en el siguiente cuadro. Compare lo obtenido con el valor generado por la solución exacta.

x_i	0	0.1	0.2	0.3	0.4	0.5
y_i	1	1.1103	1.2428	1.3997	1.5836	1.7974

x_i	0.6	0.7	0.8	0.9	1	
y_i	2.0442	2.3275	2.6511	3.0192	3.4366	

2. **Aproximando números famosos.** Repita el Ejercicio 5 de la Sección 1.1 con el algoritmo de RK4, para $N = 10$, 20, 40 y 80, redondeando a 4 cifras decimales. Compare los resultados con los obtenidos con Euler y Euler mejorado.

3. Considere el siguiente PVI:

$$y' = -y + \operatorname{sen} x + \cos x, \qquad y(0) = 0.$$

Se desea estimar el valor de la variable de estado y cuando $x = 2$.

a) Halle la solución exacta y calcule $y(2)$ en Octave, usando `format long`.

b) Complete la siguiente tabla con la aproximación de $y(2)$ obtenida utilizando los métodos de Euler, RK2 (o Euler mejorado) y RK4, con los tamaños de pasos indicados.

h	N	Euler	RK2	RK4
1/10				
1/20				
1/40				
1/80				
1/160				
1/320				

c) Determine la cantidad de pasos N y el número de evaluaciones de f que se necesitan en cada método para obtener $y(2)$ con:

(i) 3 decimales correctos;
(ii) 6 decimales correctos;
(iii) 10 decimales correctos.

En caso de no haber llegado a dicha precisión con los pasos considerados en la tabla, estime el valor de N a partir de lo obtenido.

1.4. Métodos adaptativos: ODE45

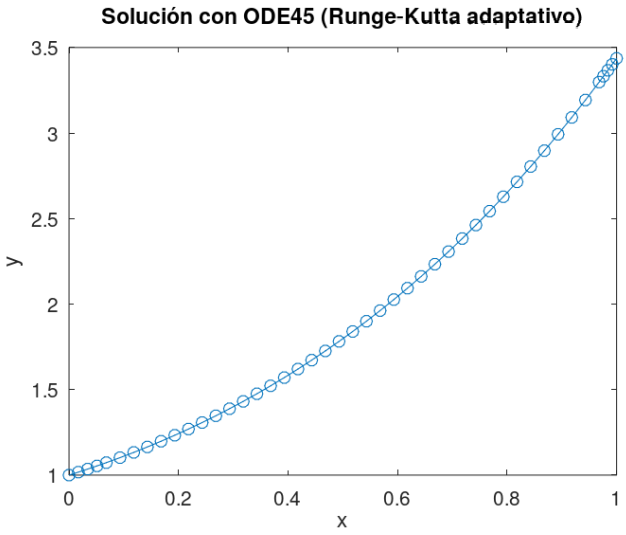
Vimos que la precisión de un método numérico para aproximar soluciones de ecuaciones diferenciales mejora cuando se reduce el tamaño de paso h. También mencionamos que esta precisión tiene un costo: incremento en el tiempo de cálculos y mayor posibilidad de errores de redondeo.

El intervalo de interés podría contener subintervalos en los que un tamaño de h no tan pequeño sea suficiente para mantener el error dentro de los límites deseados, y otros en los que se requiera un tamaño de h menor.

Existen algoritmos numéricos que usan un tamaño de paso variable según sea necesario en los distintos subintervalos, y se conocen como **métodos adaptativos.** Aunque no estudiaremos aquí estos métodos, es importante mencionar que Octave dispone de comandos para implementarlos. Por ejemplo, la función ode45 combina dos métodos de Runge–Kutta de distintos órdenes (cuarto y quinto) para estimar el error y adaptar el tamaño de paso. Asimismo, Octave cuenta con el comando ode23, basado en un método de menor orden.

```
f=@(x,y) x+y;      % Definir la función derivada
x0=0;              % Valor inicial de x
y0=1;              % Valor inicial de y
xF=1;              % Valor final de x
[x,y]=ode45(f,[x0 xF],y0)   % Resolución usando Runge-Kutta
                              adaptativo

% Graficar la solución
plot(x,y,'-o');
xlabel('x');
ylabel('y');
title('Solución con ODE45 (Runge-Kutta adaptativo)');
grid on;
```

Código 1.2: Código básico para aplicar ODE45 para el PVI del Ejemplo 1.



Si bien ode45 ofrece muchas ventajas para resolver ecuaciones diferenciales en Octave, como su facilidad de uso y la adaptación automática del paso, también presenta ciertas desventajas frente a la implementación manual de un método de Runge—Kutta de cuarto orden (RK4).

En primer lugar, con frecuencia resulta útil poder fijar un paso constante h, ya que conocer ese valor facilita otras estimaciones relacionadas con el problema. Con `ode45`, los valores se calculan en puntos no equiespaciados, lo que puede complicar la comparación con otras funciones o con soluciones analíticas. Además, una implementación propia permite ajustar de forma sencilla las tolerancias, incorporar condiciones personalizadas y, sobre todo, comprender a fondo el algoritmo al programarlo uno mismo.

1.5. Estabilidad de los métodos numéricos

La estabilidad de un método numérico se refiere a cómo evolucionan los errores de redondeo o pequeñas perturbaciones a lo largo del proceso de cálculo. En pocas palabras, un método se considera **estable** si, aplicado a un problema dado y con un tamaño de paso adecuado, no amplifica significativamente dichos errores a medida que avanzan las iteraciones. Por el contrario, si estos errores crecen de manera descontrolada, se dice que el método es **inestable** en ese contexto.

La estabilidad es crucial en muchos problemas prácticos, donde las soluciones de las ecuaciones diferenciales pueden ser sensibles a perturbaciones. Si se utiliza un método inadecuado o un tamaño de paso demasiado grande, los resultados pueden volverse inservibles, incluso cuando el modelo matemático y las condiciones iniciales sean precisos. En cambio, cuando el método y el paso son apropiados, los errores introducidos ya sea por redondeo, aproximación o medición, no se amplifican a lo largo del cálculo, manteniendo las soluciones estimadas cercanas a sus valores exactos.

La conexión entre estabilidad y ecuaciones **rígidas** (*stiff*) es especialmente importante, ya que estos problemas exigen fuertes restricciones sobre el tamaño de paso para muchos métodos. Por lo tanto, en problemas rígidos es esencial utilizar métodos con buenas propiedades de estabilidad. Aunque en principio podría lograrse estabilidad reduciendo mucho el tamaño del paso, esto suele requerir valores tan pequeños que hacen que el método sea ineficiente o inadecuado en la práctica.

En el estudio numérico es habitual distinguir entre métodos explícitos e implícitos. Si bien esta clasificación aparece en diversos contextos del análisis numérico, en este libro la introducimos en el marco de la resolución de ecuaciones diferenciales. Un método es **explícito** cuando permite calcular la aproximación en el paso siguiente a partir de información ya conocida del paso actual, sin necesidad de resolver ecuaciones adicionales. En cambio, un método es **implícito** cuando la aproximación en el nuevo paso aparece definida de manera indirecta, a partir de una relación que involucra tanto el estado presente como el futuro, lo que obliga a resolver una ecuación (en general no lineal) en cada iteración. Esto hace que los métodos implícitos sean, en

términos generales, más costosos y complejos de implementar. Sin embargo, tienden a presentar mejores propiedades de estabilidad.

Los métodos explícitos, como el de Euler, suelen tener restricciones severas en el tamaño del paso h para mantener la estabilidad cuando se resuelven ecuaciones rígidas. Si el paso es demasiado grande, estos métodos generan soluciones numéricas inestables que no reflejan el comportamiento real.

Para ilustrarlo, considere el PVI

$$y' = -100y, \qquad y(0) = 1.$$

Es fácil comprobar que la solución exacta es $y(x) = e^{-100x}$. Así, por ejemplo, $y(0.5)$ es prácticamente cero. Veamos qué aproximaciones se obtienen para este valor mediante el método de Euler con diferentes tamaños de paso h:

- $h = 0.1 \longrightarrow y(0.5) \approx -59049$;
- $h = 0.05$: $\longrightarrow y(0.5) \approx 1.0486 \cdot 10^{6}$;
- $h = 0.025$:$\longrightarrow y(0.5) \approx 3.3253 \cdot 10^{3}$;
- $h = 0.02$: $\longrightarrow y(0.5) \approx -1$;
- $h = 0.0125$: $\longrightarrow y(0.5) \approx 8.2718 \cdot 10^{-25}$;
- $h = 0.01$: $\longrightarrow y(0.5) \approx 0$.

Como puede verse, al disminuir el valor de h, las soluciones numéricas obtenidas se aproximan mejor al valor exacto. En cambio, para valores más grandes de h, las soluciones difieren significativamente de la solución analítica.

Métodos implícitos como Euler implícito o Crank–Nicolson son más adecuados en estos casos, ya que permiten resolver ecuaciones rígidas de manera más eficiente y precisa, combinando estabilidad numérica con una buena precisión. No consideraremos estos métodos en este texto, ya que nos centraremos exclusivamente en métodos explícitos, debido a su simplicidad de implementación y su valor formativo inicial. En cuanto a Octave, dispone de funciones como `ode23s` y `ode15s` diseñadas para la resolución de problemas rígidos mediante métodos implícitos.

Para un estudio más detallado sobre la estabilidad, consulte un libro de análisis numérico. En general, los métodos presentados en este texto tienen buenas propiedades de estabilidad para los problemas tratados.

Capítulo 2

Sistemas de ecuaciones y ecuaciones de orden superior

Hasta ahora vimos varios métodos para resolver un PVI de primer orden de la forma

$$y'(x) = f(x, y), \qquad y(x_0) = y_0.$$

Pero toda ecuación diferencial de orden superior, que pueda expresarse en su forma **normal**[1] puede llevarse a un sistema de primer orden del tipo

$$\boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y}),$$

donde x es la variable independiente, mientras que $\boldsymbol{y} = (y_1, y_2, \cdots, y_m)$ y $\boldsymbol{f} = (f_1, f_2, \cdots, f_m)$ son funciones vectoriales. Lo mismo ocurre con todo sistema de orden superior que pueda expresarse en su forma normal. Por eso resulta importante poder aplicar los métodos trabajados hasta ahora para obtener aproximaciones numéricas para la solución de sistemas de primer orden con valores iniciales. Abordaremos esto en la siguiente sección, y luego lo aplicaremos a PVI y sistemas de orden superior.

[1] Una ecuación diferencial ordinaria de orden m, dada por $F\left(x, y, y', y'', \ldots, y^{(m)}\right) = 0$, está en forma **normal** si puede expresarse como $y^{(m)} = g\left(x, y, y', \ldots, y^{(m-1)}\right)$, es decir, si es posible despejar $y^{(m)}$ como una función explícita de la variable independiente x, de la función y, y de sus derivadas hasta el orden $m-1$.

2.1. Solución numérica de un sistema de primer orden

Consideremos el PVI dado por

$$\boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y}), \qquad \boldsymbol{y}(x_0) = \boldsymbol{y}_0, \tag{2.1}$$

donde x es la variable independiente, y las funciones $\boldsymbol{y} = (y_1, y_2, \cdots, y_m)$ y $\boldsymbol{f} = (f_1, f_2, \cdots, f_m)$ son vectoriales. Si las funciones componentes de $\boldsymbol{f}$ y sus derivadas parciales de primer orden son todas continuas en una vecindad de $(x_0, \boldsymbol{y}_0)$, puede probarse que existe una única solución $\boldsymbol{y} = \boldsymbol{y}(x)$ para (2.1) en algún intervalo que contiene a x_0. Buscaremos soluciones numéricas para problemas que satisfagan esto.

Métodos de Euler para sistemas

Haciendo unas pequeñas modificaciones al código para el algoritmo de Euler podremos aplicarlo para obtener la solución de sistemas como (2.1). Para ilustrarlo consideremos primero el caso $m = 2$, es decir, un sistema como

$$\begin{array}{ll} y_1'(x) = f_1(x, y_1, y_2), & y_1(x_0) = y_0^1; \\ y_2'(x) = f_2(x, y_1, y_2), & y_2(x_0) = y_0^2. \end{array} \tag{2.2}$$

Podemos pensar que cada una de las funciones vectoriales f_1 y f_2 recibe dos tipos de argumentos: un escalar x y un vector columna $\boldsymbol{y}$ de dos componentes y_1 e y_2. Además, cada una de ellas devuelve un valor real. En consecuencia, definimos

$$\boldsymbol{f}(x, \boldsymbol{y}) = \begin{pmatrix} f_1(x, \boldsymbol{y}) \\ f_2(x, \boldsymbol{y}) \end{pmatrix}, \qquad \text{donde } \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}.$$

El sistema (2.2) puede verse entonces como el PVI

$$\boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y}), \qquad \boldsymbol{y}(x_0) = \boldsymbol{y}_0,$$

siendo $\boldsymbol{y}_0 = \begin{pmatrix} y_0^1 \\ y_0^2 \end{pmatrix}$. Así, solo hace falta modificar el algoritmo de Euler para que admita entradas vectoriales, como mostramos a continuación. Antes de ver el código, analicemos los elementos involucrados:

- $\boldsymbol{f}(x, \boldsymbol{y})$ recibirá como entradas un escalar x y un vector **columna** $\boldsymbol{y}$ de m componentes, y retornará otro vector columna de igual dimensión con las derivadas. Por ejemplo

```
f=@(x,y) [-x*y(2); x*(y(1)-y(2))]
```

es la función para el sistema

$$\begin{aligned} y_1'(x) &= -xy_2, \\ y_2'(x) &= x(y_1 - y_2). \end{aligned} \tag{2.3}$$

- $[x_0, x_F]$ es un vector fila de dos componentes que indican el intervalo donde se desea conocer la solución. La coma puede estar o no, es lo mismo para Octave.
- $\boldsymbol{y}_0$ es un vector columna que indica el dato inicial.
- N indica la cantidad de sub-intervalos en que se divide $[x_0, x_F]$ para resolver por el método de Euler, y determina a $h = (x_F - x_0)/N$.
- Como no sabemos el valor de m y queremos que el código funcione no solo para $m = 2$, lo determinamos como `m=length(y0)`.

Crearemos la función `eulerSist` que tendrá como entrada los elementos anteriores. La salida de la instrucción `[x,y] = eulerSist(f,[x0 xF], y0, N)` será

- x: un vector fila de $N + 1$ componentes;
- y: una matriz con m filas y $N+1$ columnas. La n-ésima fila debe contener la aproximación para los valores de la función desconocida y_n en los puntos x_i. Así, el valor aproximado de cada función buscada en x_F se observará en la **última columna.** Algunos prefieren dar como salida la transpuesta de esta matriz, para que los resultados buscados se observen en la última fila, como en el método ode45 disponible en Octave. Cada usuario optará por lo que prefiera, pero deberá tenerlo presente sobre todo para graficar.

```
function [x,y]=eulerSist(f,inter,y0,N)
    % function [x,y]=eulerSist(f,[x0,xF],y0,N)
    % Método de Euler para resolver
    %    y' = f(x,y) en [x0,xF]
    %    y(x0) = y0
    % Usando N pasos
    % y0 puede ser vectorial (columna) o escalar

x=linspace(inter(1),inter(2),N+1);    % División del intervalo
h=(inter(2)-inter(1))/N;              % Tamaño de paso

% Reservamos lugar en memoria para y
y=zeros(length(y0),N+1);  % Dimensión de y0 por N+1 columnas
y(:,1)=y0;                % Condición inicial

% Método de Euler
for n=1:N
    y(:,n+1)=y(:,n)+h*f(x(n),y(:,n));
end

end
```

Código 2.1: Función para el método de Euler para sistemas.

Si se desea graficar las funciones, para el caso $m = 2$ puede agregarse lo siguiente al código anterior, antes del último `end`:

```
% Graficar las soluciones y1(x) e y2(x):
plot(x,y(1,:),'-r',x,y(2,:),'-b');
xlabel('x');
ylabel('y');
title('Método de Euler para un sistema de ecuaciones
    diferenciales');
legend('y1','y2');
grid on;
```

Ejemplo 2. Resuelva el siguiente sistema autónomo en $[0, 1]$ con $N = 10$, en el que las funciones desconocidas son y_1 e y_2:

$$\begin{aligned} y_1'(x) &= 3y_1 - 2y_2, & y_1(0) &= 3; \\ y_2'(x) &= 5y_1 - 4y_2, & y_2(0) &= 6. \end{aligned}$$

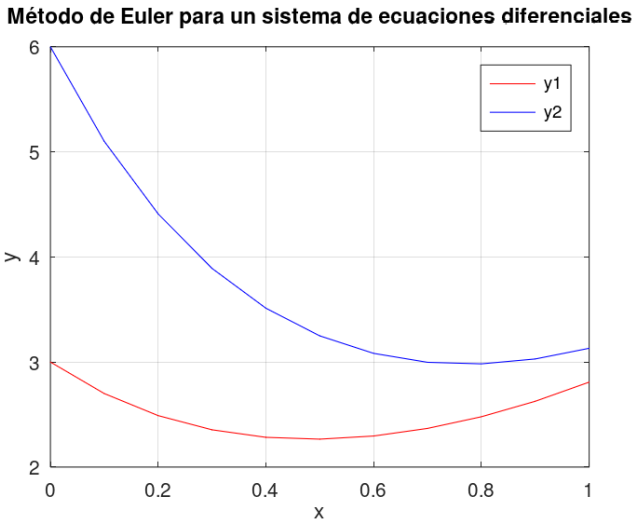
Solución. Sabemos que la solución exacta del sistema es

$$y_1(x) = 2e^{-2x} + e^x, \qquad y_2(x) = 5e^{-2x} + e^x,$$

pero aplicaremos el método de Euler para ver los resultados. Vamos a ejecutar el Código 2.1 escribiendo:

```
f=@(x,y) [3*y(1)-2*y(2); 5*y(1)-4*y(2)];

y0=[3;6];

[x,y]=eulerSist(f,[0 1],y0,10)

x =

0  0.1000  0.2000  0.3000  0.4000  0.5000  0.6000  0.7000  0.8000  0.9000
     1.0000

y =

3.0000  2.7000  2.4900  2.3550  2.2833  2.2659  2.2958  2.3681  2.4791
    2.6264  2.8085

6.0000  5.1000  4.4100  3.8910  3.5121  3.2489  3.0823  2.9973  2.9824
    3.0290  3.1306
```

Esto nos da, por ejemplo, las aproximaciones $y_1(1) = 2.8085$ (fila 1, columna 11 de la matriz `y`) e $y_2(1) = 3.1306$ (fila 2, columna 11 de la matriz `y`). Para ver solo estos valores, escribimos `y(:,end)`. A partir de las soluciones exactas sabemos que los valores reales, redondeados a 4 cifras decimales, son $y_1(1) = 2.9890$ e $y_2(1) = 3.3950$. Por supuesto podemos mejorar la precisión tomando h más chico, o con los métodos de Euler mejorado (ver Ejercicio 3) o de RK. Si agregamos el código para los gráficos, el resultado es el siguiente:



La notación usada en (2.2) corresponde a la analogía con lo trabajado previamente. Sin embargo, muchos sistemas provienen del modelado de situaciones reales en las que la variable independiente es el tiempo. Por ello, la forma más usual de encontrar escrito un sistema como (2.2) es

$$\begin{aligned} x'(t) &= f(t,x,y), & x(t_0) &= x_0; \\ y'(t) &= g(t,x,y), & y(t_0) &= y_0. \end{aligned} \tag{2.4}$$

Por ejemplo, el sistema (2.3) puede escribirse en estos términos como

$$\begin{aligned} x'(t) &= -ty, \\ y'(t) &= t(x-y), \end{aligned}$$

mientras que el sistema autónomo del Ejemplo 2 se escribe

$$\begin{aligned} x'(t) &= 3x-2y, & x(0) &= 3; \\ y'(t) &= 5x-4y, & y(0) &= 6. \end{aligned} \tag{2.5}$$

Resultará útil, sobre todo para las aplicaciones "a mano", tener disponibles las fórmulas del método de Euler para sistemas como (2.4). Como antes, partiendo del intervalo $[t_0, t_F]$ y un paso $h > 0$ fijo dado por $(t_F - t_0)/N$, siendo N la cantidad de subintervalos considerados, definimos la lista de puntos

$$t_0, \qquad t_1 = t_0 + h, \qquad t_2 = t_0 + 2h, \qquad \dots \qquad t_F,$$

donde $t_{n+1} = t_0 + (n+1)h = t_n + h$, para $n = 0, 1, 2, \dots, N$. El método de Euler para el sistema (2.4) es, entonces,

$$\begin{aligned} x_{n+1} &= x_n + h \cdot f(t_n, x_n, y_n), \\ y_{n+1} &= y_n + h \cdot g(t_n, x_n, y_n). \end{aligned} \tag{2.6}$$

El **método de Euler mejorado** se extiende a sistemas de manera análoga: primero se calcula el predictor

$$\boldsymbol{y}_{n+1}^{*} = \boldsymbol{y}_n + h\boldsymbol{f}(x_n, \boldsymbol{y}_n), \tag{2.7}$$

y después el corrector

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + \frac{h}{2}\left[\boldsymbol{f}(x_n, \boldsymbol{y}_n) + \boldsymbol{f}(x_{n+1}, \boldsymbol{y}_{n+1}^{*})\right]. \tag{2.8}$$

Para el caso particular del sistema (2.4), las ecuaciones para el método de Euler mejorado son:

$$\begin{aligned} x_{n+1}^{*} &= x_n + h \cdot f(t_n, x_n, y_n), \\ y_{n+1}^{*} &= y_n + h \cdot g(t_n, x_n, y_n). \end{aligned} \tag{2.9}$$

y

$$\begin{aligned} x_{n+1} &= x_n + \tfrac{h}{2}\left[f(t_n, x_n, y_n) + f(t_{n+1}, x_{n+1}^{*}, y_{n+1}^{*})\right], \\ y_{n+1} &= y_n + \tfrac{h}{2}\left[g(t_n, x_n, y_n) + g(t_{n+1}, x_{n+1}^{*}, y_{n+1}^{*})\right]. \end{aligned} \tag{2.10}$$

Ejemplo 3 (Euler mejorado)**.** Aplique las fórmulas (2.9) y (2.10) con $h = 0.2$ para aproximar la solución del PVI dado en (2.5) para $t = 0.2$.

Solución. Debemos hacer un solo paso. Partiendo de $x_0 = 3$ e $y_0 = 6$, tenemos los predictores

$$\begin{aligned} x_1^{*} &= 3 + (0.2)\cdot[3\cdot 3 - 2\cdot 6] = 2.4, \\ y_1^{*} &= 6 + (0.2)\cdot[5\cdot 3 - 4\cdot 6] = 4.2. \end{aligned}$$

Entonces los correctores son

$$\begin{aligned} x_1 &= 3 + (0.1)\cdot([3\cdot 3 - 2\cdot 6] + [3\cdot(2.4) - 2\cdot(4.2)]) = 2.58, \\ y_1 &= 6 + (0.1)\cdot([5\cdot 3 - 4\cdot 6] + [5\cdot(2.4) - 4\cdot(4.2)]) = 4.62. \end{aligned}$$

Observemos que un solo paso del método de Euler mejorado produce mejor precisión que dos pasos de Euler ordinario. Para verlo, sin tener que hacer a mano estos dos pasos, observemos la columna 3 de la matriz y que dio como salida Octave en el Ejemplo 2. Las aproximaciones resultantes para $t = 0.2$ son 2.49 y 4.41, respectivamente. A partir de las soluciones exactas sabemos que los valores reales, redondeados a 4 cifras decimales, son $x(0.2) = 2.5620$ e $y(0.2) = 4.5730$.

Métodos de Runge–Kutta para sistemas

La versión vectorial de la fórmula iterativa del método de Runge–Kutta 4 es

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + \frac{h}{6}\left(\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3 + \boldsymbol{k}_4\right), \tag{2.11}$$

donde los vectores $\boldsymbol{k}_i$ se definen, en analogía con (1.6), como

$$\begin{aligned}
\boldsymbol{k}_1 &= \boldsymbol{f}(x_n, \boldsymbol{y}_n),\\
\boldsymbol{k}_2 &= \boldsymbol{f}(x_n + \tfrac{1}{2}h, \boldsymbol{y}_n + \tfrac{1}{2}h\boldsymbol{k}_1),\\
\boldsymbol{k}_3 &= \boldsymbol{f}(x_n + \tfrac{1}{2}h, \boldsymbol{y}_n + \tfrac{1}{2}h\boldsymbol{k}_2),\\
\boldsymbol{k}_4 &= \boldsymbol{f}(x_n + h, \boldsymbol{y}_n + h\boldsymbol{k}_3).
\end{aligned}$$

Para el caso particular del sistema (2.4), las ecuaciones anteriores se convierten en:

$$\begin{aligned}
x_{n+1} &= x_n + \tfrac{h}{6}(m_1 + 2m_2 + 2m_3 + m_4),\\
y_{n+1} &= y_n + \tfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4),
\end{aligned} \tag{2.12}$$

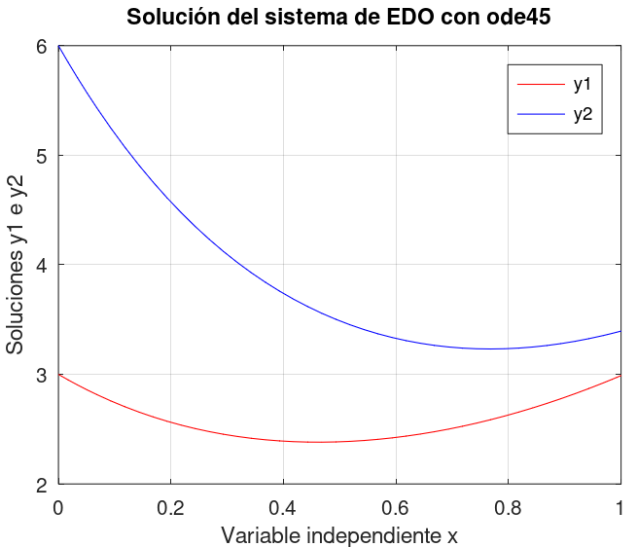
donde

$$\begin{aligned}
m_1 &= f(t_n, x_n, y_n),\\
k_1 &= g(t_n, x_n, y_n),\\
m_2 &= f(t_n + \tfrac{1}{2}h, x_n + \tfrac{1}{2}hm_1, y_n + \tfrac{1}{2}hk_1),\\
k_2 &= g(t_n + \tfrac{1}{2}h, x_n + \tfrac{1}{2}hm_1, y_n + \tfrac{1}{2}hk_1),\\
m_3 &= f(t_n + \tfrac{1}{2}h, x_n + \tfrac{1}{2}hm_2, y_n + \tfrac{1}{2}hk_2),\\
k_3 &= g(t_n + \tfrac{1}{2}h, x_n + \tfrac{1}{2}hm_2, y_n + \tfrac{1}{2}hk_2),\\
m_4 &= f(t_n + h, x_n + hm_3, y_n + hk_3),\\
k_4 &= g(t_n + h, x_n + hm_3, y_n + hk_3).
\end{aligned} \tag{2.13}$$

Como es de suponer, la función ode45 admite entradas vectoriales, por lo que es una herramienta excelente para resolver numéricamente sistemas de ecuaciones diferenciales. Por ejemplo, la siguiente instrucción llama al método para el PVI del Ejemplo 2.

```
f=@(x,y) [3*y(1)-2*y(2); 5*y(1)-4*y(2)];
y0=[3;6];
[x,y] = ode45(f,[0,1],y0);
```

La última **fila** de la matriz y de salida corresponde a las aproximaciones obtenidas por el método para y_1 e y_2 en el extremo final del intervalo. En este caso, escribiendo `y(end,:)` se obtiene que $y_1(1) = 2.9890$ e $y_2(1) = 3.3950$, resultados que coinciden exactamente cuando se calculan mediante las soluciones explícitas redondeadas a 4 cifras decimales.



Ejercicios Sección 2.1

Respuestas en página 78

En los Ejercicios 1 y 2, para aproximar $x(0.2)$ e $y(0.2)$, use:

a) dos pasos del método de Euler con $h = 0.1$;

b) un paso del método de Euler mejorado con $h = 0.2$;

c) un paso del método RK4 con $h = 0.2$;

d) >_ la función `ode45` de Octave. ¿Cuántos pasos realizó?

.

1. $\begin{cases} x' &= x + 2y; & x(0) = 1, \\ y' &= 4x + 3y; & y(0) = 1. \end{cases}$ 2. $\begin{cases} x' &= t - y; & x(0) = -3, \\ y' &= x - t; & y(0) = 5. \end{cases}$

3. >_ Use las fórmulas (2.7) y (2.8) para crear un código que permita aplicar el método de Euler mejorado a sistemas como (2.1). Úselo para resolver el PVI del Ejemplo 2 con $N = 10$ y compare los resultados para $x = 1$ con lo arrojado por Euler en dicho ejemplo.

4. (De [3]) La trayectoria de una partícula que inicialmente se encuentra en el punto $(-1, 1)$ y se mueve en el plano, está dada por la curva $(x(t), y(t))$, donde las funciones x e y son la solución del siguiente sistema de ecuaciones diferenciales:

$$\begin{cases} x' = -ty, \\ y' = tx - ty. \end{cases}$$

>_ Modifique el Código 2.1 para implementar el método de Runge–Kutta 4 en sistemas de ecuaciones diferenciales de primer orden. Úselo para resolver el problema para $0 \leq t \leq 20$ con paso $h = 0.1$.

a) A partir de los datos arrojados por el código anterior, determine la posición de la partícula y su velocidad a los 3 segundos.

b) ¿Cuántos dígitos correctos tiene la aproximación para la posición de la partícula hallada en el inciso anterior? Explique cómo lo determinó.

c) Grafique la trayectoria de la partícula durante los primeros 20 segundos. Esto es, grafique los valores $(x(t), y(t))$ resultantes.

d) Recuerde que la longitud de la trayectoria de la partícula durante los T primeros segundos está dada por $\int_0^T \sqrt{(x')^2 + (y')^2}\, dt$. Calcule la distancia recorrida por la partícula durante los primeros 3 segundos. Explique cómo lo hizo. ¿Cuál será la distancia recorrida por los siguientes 17 segundos? *Sugerencia:* vea el Apéndice para recordar cómo elevar al cuadrado cada elemento de un vector.

e) Determine el primer instante en el cual la partícula se halla a una distancia menor que 0.01 del origen de coordenadas.

5. **Modelo predador-presa de Lotka–Volterra.** Consideremos el siguiente modelo

$$\begin{cases} x'(t) = x(3 - 0.002y), \\ y'(t) = -y(0.5 - 0.0006x). \end{cases}$$

a) ¿Qué variable representa al predador y cuál representa a la presa? Para determinar esto, tenga en cuenta las siguientes preguntas:

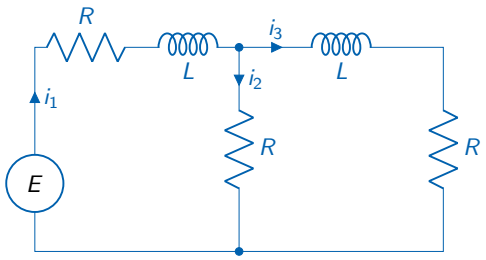
- ¿Cuál de las dos especies puede sobrevivir sin la existencia de la otra?
- ¿Qué especie se extingue si la otra no existe?

b) >_ Grafique la evolución de ambas poblaciones en el tiempo, asumiendo que la población inicial de presas es 1600 y la de predadores es 800, y que el tiempo t se mide en meses. Grafique el diagrama de fases. Describa el fenómeno representado. Utilice RK4 y un intervalo de tiempo que sea representativo de lo que sucede.

c) >_ Determine la duración del ciclo o período en este modelo. Explique cómo lo hizo.

d) >_ Determine el primer momento en que ambas poblaciones coinciden. Identifique la tasa de crecimiento instantánea, explicando si ambas poblaciones crecen, ambas decrecen, o si una crece y la otra decrece.

6. (Ejercicio 6, Sección 9.4 de [4]) Cuando $E = 100$ V, $R = 10\,\Omega$ y $L = 1$ H, el sistema de ecuaciones diferenciales para las corrientes $i_1(t)$ e $i_3(t)$ en la red eléctrica del gráfico es:



$$i_1' = -20i_1 + 10i_3 + 100,$$
$$i_3' = 10i_1 - 20i_3,$$
$$i_1(0) = 0,$$
$$i_3(0) = 0.$$

Usando cualquiera de los métodos de resolución numérica, obtenga la gráfica de la solución para el intervalo $0 \leq t \leq 5$. Use las gráficas para predecir el comportamiento de ambas corrientes cuando $t \to \infty$.

2.2. Problemas de orden superior con valores iniciales

Consideremos primero un PVI de la forma

$$y'' = g(x, y, y'), \qquad y(x_0) = y_0, \qquad y'(x_0) = u_0, \tag{2.14}$$

donde y es la función desconocida con variable independiente x. Como siempre, supondremos que tenemos garantizada existencia y unicidad de la solución en un intervalo que contiene a x_0.

Si definimos $y_1 = y$ e $y_2 = y'$, entonces el PVI (2.14) se puede reescribir como

$$\begin{aligned} y_1' &= y_2, & y_1(x_0) &= y_0; \\ y_2' &= g(x, y_1, y_2), & y_2(x_0) &= u_0, \end{aligned}$$

que es de la forma de (2.2) con $f_1(x, y_1, y_2) = y_2$. Similarmente, si el PVI está dado como

$$x'' = g(t, x, x'), \qquad x(t_0) = x_0, \qquad x'(t_0) = y_0, \tag{2.15}$$

entonces llamando $y = x'$ el mismo se reescribe como el sistema equivalente

$$\begin{aligned} x' &= y, & x(t_0) &= x_0; \\ y' &= g(t, x, y), & y(t_0) &= y_0, \end{aligned} \tag{2.16}$$

que es de la forma de (2.4) con $f(t, x, y) = y$. Así, para este caso, las fórmulas dadas en (2.6) se convierten en

$$\text{Euler para (2.16)} \longrightarrow \quad \begin{aligned} x_{n+1} &= x_n + h \cdot y_n, \\ y_{n+1} &= y_n + h \cdot g(t_n, x_n, y_n). \end{aligned} \tag{2.17}$$

Ejemplo 4. (Método de Euler para un PVI de segundo orden) Aplique dos pasos del método de Euler para obtener el valor aproximado de $x(0.2)$, siendo x la solución al PVI

$$x'' + tx' + x = 0, \qquad x(0) = 1, \qquad x'(0) = 2.$$

Solución. Si reescribimos la ecuación diferencial como en (2.15), nos queda

$$x'' = -tx' - x = g(t, x, x').$$

Además $t_0 = 0$, $x_0 = 1$ e $y_0 = 2$. Usando un tamaño de paso $h = 0.1$ tenemos que

$$\begin{aligned}
x_1 &= x_0 + h \cdot y_0 = 1 + 0.1 \cdot 2 = 1.2,\\
y_1 &= y_0 + h \cdot [-t_0 y_0 - x_0] = 2 + 0.1 \cdot [-0 \cdot 2 - 1] = 1.9,\\
x_2 &= x_1 + h \cdot y_1 = 1.2 + 0.1 \cdot 1.9 = 1.39,\\
y_2 &= y_1 + h \cdot [-t_1 y_1 - x_1] = 1.9 + 0.1 \cdot [-(0.1) \cdot (1.9) - 1.2] = 1.761.
\end{aligned}$$

Así, concluimos que $x(0.2) \approx 1.39$. El segundo valor corresponde a una aproximación para la derivada en ese instante, es decir, $x'(0.2) \approx 1.761$.

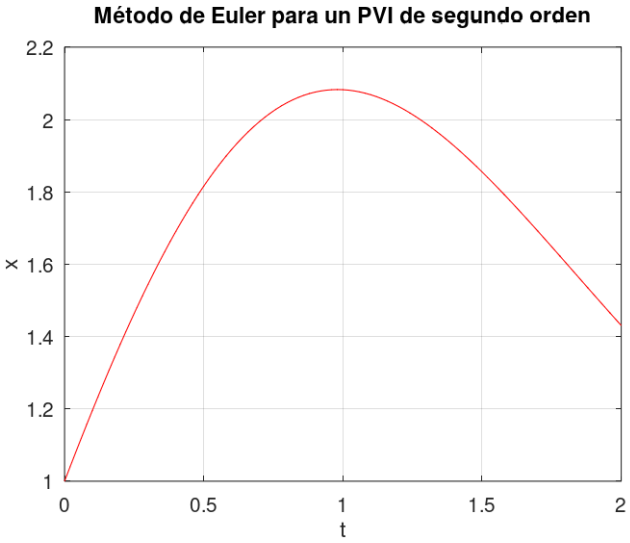
Para resolver el sistema (2.16) en Octave con el método de Euler dado en el Código 2.1 no necesitamos las fórmulas (2.17). Es suficiente con ejecutarlo con $\boldsymbol{f}(t, x, y) = (y, g(t, x, y))$. Para el caso del ejemplo anterior, escribimos:

```
F=@(t,y) [y(2); -t*y(2)-y(1)];
y0=[1;2];
[t,y]=eulerSist(F,[0 0.2],y0,2)
t =

0   0.1000   0.2000

y =

1.0000   1.2000   1.3900
2.0000   1.9000   1.7610
```

Notar que los valores obtenidos para x estarán en la **primera fila** de la matriz y de salida. Estos corresponden a la solución buscada para (2.15). Así, por ejemplo, el valor de la solución x del PVI (2.15) en t_F se encontrará en la columna $N+1$ de la fila 1 de la matriz de salida, esto es, `y[1,N+1]`. Recordar que ode45 arroja los resultados de forma transpuesta, por lo que correspondería al valor `y[N+1,1]`. Así, para graficar lo obtenido con el Código 2.1, escribimos

```
% Grafico de x(t):
plot(t,y(1,:),'-r');
xlabel('t');
ylabel('x');
title('Método de Euler para un PVI de segundo orden');
grid on;
```

En el ejemplo previo tomamos $N = 2$, pero podemos aumentar su valor para disminuir el tamaño del paso h y así obtener una mejor precisión. El siguiente gráfico se obtuvo con las instrucciones anteriores, como solución al PVI del Ejemplo 4 en $[0, 2]$, tomando $N = 100$.



En el Ejercicio 3 de esta sección se piden las fórmulas que se obtienen a partir de (2.9) y (2.10) para aplicar el método de Euler mejorado al sistema (2.16). A continuación escribimos las que se obtienen de (2.12) y (2.13) para el método RK4 aplicado a este sistema particular:

$$\begin{aligned} x_{n+1} &= x_n + \tfrac{h}{6}(m_1 + 2m_2 + 2m_3 + m_4), \\ y_{n+1} &= y_n + \tfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \end{aligned} \tag{2.18}$$

donde

$$\begin{aligned} m_1 &= y_n, & k_1 &= g(t_n, x_n, y_n), \\ m_2 &= y_n + \tfrac{1}{2}hk_1, & k_2 &= g(t_n + \tfrac{1}{2}h, x_n + \tfrac{1}{2}hm_1, y_n + \tfrac{1}{2}hk_1), \\ m_3 &= y_n + \tfrac{1}{2}hk_2, & k_3 &= g(t_n + \tfrac{1}{2}h, x_n + \tfrac{1}{2}hm_2, y_n + \tfrac{1}{2}hk_2), \\ m_4 &= y_n + hk_3, & k_4 &= g(t_n + h, x_n + hm_3, y_n + hk_3). \end{aligned} \tag{2.19}$$

Como antes, estas fórmulas resultan útiles cuando se quiere aplicar los métodos "a mano" a sistemas como (2.16), que son bastantes frecuentes. Pero para utilizar Octave solamente debemos aplicar el código correspondiente a sistemas, con $\boldsymbol{f}(t, x, y) = (y, g(t, x, y))$, siendo g como en (2.15).

Como ilustración del procedimiento, retomaremos el PVI del Ejemplo 4 para resolverlo a mano con RK4.

Ejemplo 5. (RK4 para un PVI de segundo orden) Use dos pasos del método de Runge–Kutta 4 para obtener el valor aproximado de $x(0.2)$, siendo x la solución al PVI

$$x'' + tx' + x = 0, \qquad x(0) = 1, \qquad x'(0) = 2.$$

Solución. Como antes, reescribimos la ecuación como

$$x'' = -tx' - x = g(t, x, x'),$$

y usaremos nuevamente un tamaño de paso $h = 0.1$.

- **Primer paso:** de $t = 0$ a $t = 0.1$. Tenemos $t_0 = 0$, $x_0 = 1$, $y_0 = 2$, y reemplazando en (2.19) y redondeando a 4 cifras decimales, obtenemos

  $$\begin{aligned} m_1 &= 2, & m_2 &= 1.95, & m_3 &= 1.9401, & m_4 &= 1.8805, \\ k_1 &= -1, & k_2 &= -1.1975, & k_3 &= -1.1945, & k_4 &= -1.3821. \end{aligned}$$

  Reemplazando ahora en (2.18), tenemos

  $$x_1 = 1.1943, \qquad y_1 = 1.8806.$$

- **Segundo paso:** de $t = 0.1$ a $t = 0.2$. Ahora $t_1 = 0.1$, $x_1 = 1.1943$, $y_1 = 1.8806$ y

  $$\begin{aligned} m_1 &= 1.8806, & m_2 &= 1.8114, & m_3 &= 1.8026, & m_4 &= 1.7250, \\ k_1 &= -1.3824, & k_2 &= -1.5601, & k_3 &= -1.5553, & k_4 &= -1.7196. \end{aligned}$$

  Por lo tanto

  $$x_2 = 1.3749, \qquad y_2 = 1.7250.$$

Así, concluimos que $x(0.2) \approx 1.3749$.

Como se ve en ejemplo anterior, aun cuando no hemos especificado los pasos intermedios, aplicar RK4 "a mano" es bastante tedioso (mucho más que Euler o Euler mejorado). En el Ejercicio 4 de la Sección 2.1 se pidió modificar el Código 2.1 para implementar el método de Runge–Kutta 4 en sistemas de ecuaciones diferenciales de primer orden. Lo utilizaremos en el siguiente ejemplo.

Ejemplo 6 (Euler versus RK4)**.** La solución exacta del problema con condiciones iniciales

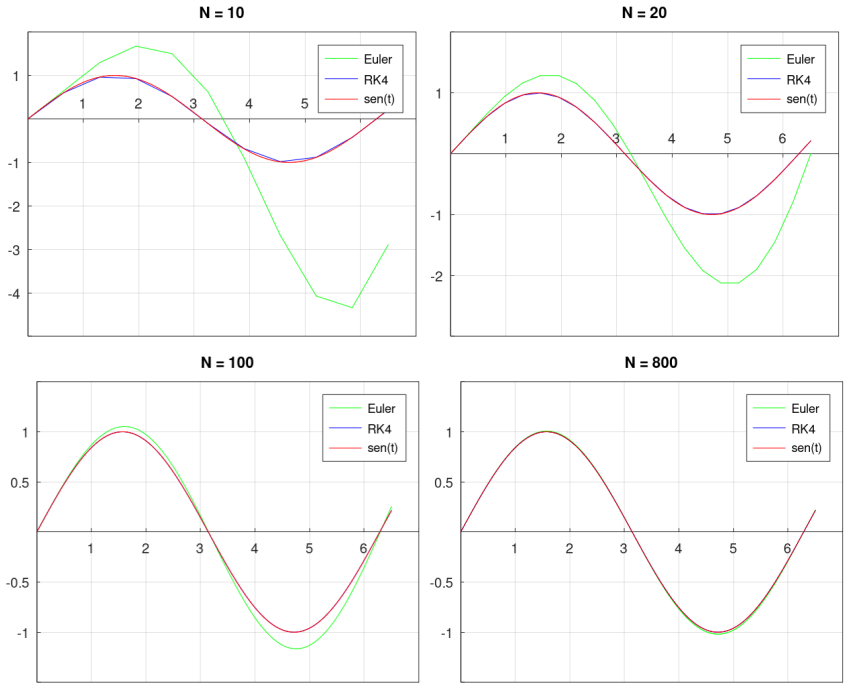
$$x'' = -x, \qquad x(0) = 0, \qquad x'(0) = 1$$

es $x(t) = \text{sen}(t)$. Aplique los métodos de Euler y RK4 con diferentes pasos para resolver el PVI para $0 \leq t \leq 6.5$. Compare gráficamente los resultados con el de la solución exacta.

Solución. Usaremos las funciones que definimos nosotros previamente, `eulerSist` y `rk4`, ambas para resolver sistemas de ecuaciones diferenciales. Escribimos:

```
%Datos
f=@(t,y) [y(2); -y(1)];   % definimos la función
y0=[0;1];                 % condiciones iniciales
inter=[0 6.5];            % intervalo de solución
N=10;                     % cantidad de pasos

%Aplicamos ambos métodos
[t,E] = eulerSist(f, inter, y0, N);     % Método Euler
[t,R] = rk4(f, inter, y0, N);           % Método RK4
```

Al repetir para diferentes valores de N obtenemos los siguientes gráficos:



Como puede verse, el método de Euler aproxima muy mal cuando N es chico, mientras que RK4 se mantiene siempre cerca de la solución real. Notar que desde $N = 10$, el valor de la aproximación por RK4 en los valores t_i están prácticamente sobre la gráfica de la solución real. En las gráficas para $N = 100$ y $N = 800$, las curvas azules generadas con RK4 no se aprecian porque coinciden con la curva solución.

Ejemplo 7 (La ecuación de Airy)**.** La ecuación diferencial

$$y'' - xy = 0$$

es llamada **ecuación de Airy.**[2] Parece muy sencilla, pero no tiene soluciones que sean funciones elementales. Es posible hallar dos soluciones linealmente independientes definidas por series infinitas, las que denotaremos como A y B. La función A puede obtenerse imponiendo las condiciones iniciales $y(0) = 1$ e $y'(0) = 0$, mientras que para B se plantea $y(0) = 0$ y $y'(0) = 1$.

Esbozaremos los gráficos de estas funciones resolviendo numéricamente ambos PVI y dibujando las correspondientes soluciones.

Notar que para cualquiera de los métodos numéricos trabajados, para aproximar la solución en $[x_0, x_F]$ **no es necesario que x_F sea mayor que x_0.** Si fuera menor, estamos resolviendo "hacia atrás" de x_0. Así, para graficar soluciones en un intervalo alrededor de x_0, podemos dividir el problema en dos partes:

- resolvemos para $[x_0, x_2]$ con $x_2 > x_0$ y ciertas condiciones iniciales en x_0;
- resolvemos para $[x_0, x_1]$ con $x_1 < x_0$ usando las mismas condiciones iniciales en x_0. Esto asegura que la solución sea continua y suave a ambos lados del origen;
- combinamos las soluciones obtenidas en una sola gráfica.

Para este ejemplo elegimos la función ode45 de Octave, recordando para graficar que esto devuelve la información buscada en la primera *columna* de la matriz de salida. Las funciones obtenidas se incluyen en la Figura 2.1. Compare con los gráficos de y_1 e y_2 en la Figura 6.2.2 de la página 251 del libro [4].

Sobre la ecuación de Airy. Aparece en diversos modelos físicos: se encuentra en el estudio de la difracción de la luz, la difracción de ondas de radio alrededor de la Tierra, la aerodinámica, la deflexión de una columna vertical, en sistemas con constantes de resorte variables, entre otros lados. Las funciones de Ai y Bi, conocidas como **funciones de Airy,** son dos soluciones linealmente independientes que satisfacen condiciones en $x = 0$ similares a las del ejemplo anterior:

$$\mathrm{Ai}(0) = \frac{1}{3^{2/3}\Gamma\left(\frac{2}{3}\right)} \approx 0.35502805, \qquad \mathrm{Bi}(0) = \frac{1}{3^{1/6}\Gamma\left(\frac{2}{3}\right)} \approx 0.61492663,$$

$$\mathrm{Ai}'(0) = -\frac{1}{3^{1/3}\Gamma\left(\frac{1}{3}\right)} \approx -0.25881940, \qquad \mathrm{Bi}'(0) = \frac{3^{1/6}}{\Gamma\left(\frac{1}{3}\right)} \approx 0.44828835;$$

donde Γ denota la función gamma. Lo anterior implica que el wronskiano de $\mathrm{Ai}(x)$ y $\mathrm{Bi}(x)$ es $1/\pi$ para todo x.

[2] La forma general de una ecuación de Airy es $y'' \pm \alpha^2 xy = 0$.

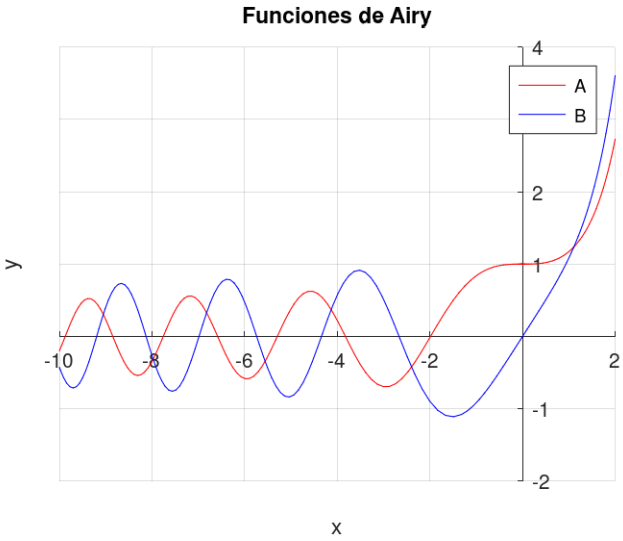


Figura 2.1: Funciones de Airy para el Ejemplo 7.

Puede probarse que, para $x > 0$, Ai es positiva, convexa y decrece exponencialmente a cero, mientras que Bi es positiva, convexa y crece exponencialmente. Cuando x es negativa, Ai y Bi oscilan alrededor de cero con frecuencia creciente y amplitud decreciente. Vamos a verificar esto resolviendo numéricamente con ode45 ambos PVI y graficando las correspondientes soluciones (ver Figura 2.2) Fuente y más detalles pueden encontrarse en https://es.wikipedia.org/wiki/Funcion_de_Airy.

PVI de orden superior

Por supuesto, lo hecho en esta sección puede aplicarse a problemas con valores iniciales de orden mayor que 2. En general, una ecuación diferencial de m-ésimo orden de la forma

$$y^{(m)} = g\left(x, y, y', \dots, y^{(m-1)}\right)$$

se puede transformar en un sistema de m ecuaciones de primer orden como (2.1) usando las sustituciones $y_1 = y$, $y_2 = y'$, $y_3 = y''$, $\dots$, $y_m = y^{(m-1)}$.

Ejemplo 8. Utilice el método RK4 para aproximar $y(2\pi)$, siendo y la solución del siguiente PVI de tercer orden:

$$y''' + 3y'' + 2y' - 6y = \operatorname{sen}(2x), \qquad y(0) = 1, \qquad y'(0) = 0, \qquad y''(0) = -1.$$

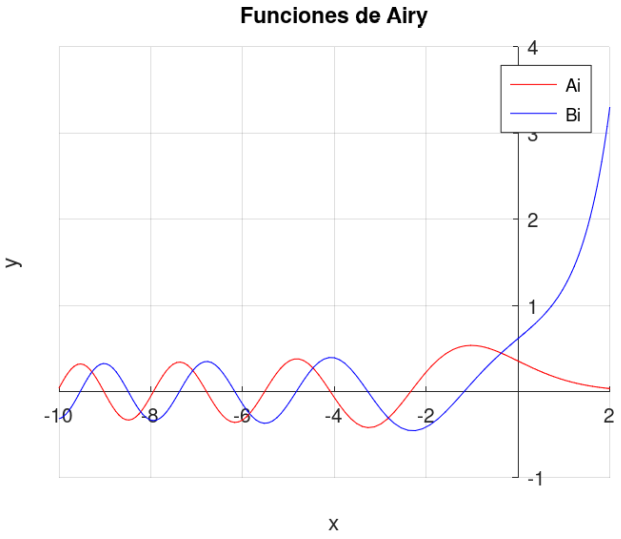


Figura 2.2: Aproximaciones de las funciones Ai y Bi de Airy con ode45.

Solución. Comenzamos escribiendo

$$y''' = \operatorname{sen}(2x) - 3y'' - 2y' + 6y = g(x, y, y', y'').$$

Ahora llamamos $y_1 = y$, $y_2 = y'$, $y_3 = y''$. Así, tenemos que

$$\begin{aligned} y_1' &= y_2, \\ y_2' &= y_3, \\ y_3' &= g(x, y_1, y_2, y_3) = \operatorname{sen}(2x) - 3y_3 - 2y_2 + 6y_1. \end{aligned}$$

Además, las condiciones iniciales se traducen como $y_1(0) = 1$, $y_2(0) = 0$, $y_3(0) = -1$. Este PVI es de la forma (2.1) con $m = 3$, $x_0 = 0$,

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}, \qquad \boldsymbol{f} = \begin{pmatrix} y_2 \\ y_3 \\ \operatorname{sen}(2x) - 3y_3 - 2y_2 + 6y_1 \end{pmatrix}, \qquad \boldsymbol{y}_0 = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}.$$

Para resolver, y graficar, podemos escribir lo siguiente:

```
% definimos la función
f=@(x,y) [y(2);y(3);sin(2*x)-3*y(3)-2*y(2)+6*y(1)];

y0=[1;0;-1];                          % condiciones iniciales
inter=[0 2*pi];                       % intervalo de solución
N=100;
```

```
[x,y] = rk4(f, inter, y0, N);

% Gráfico
plot(x,y(1,:));
xlabel('x');
ylabel('y');
title('Gráfico de la solución aproximada por RK4');
grid on
```

La instrucción `y(1,N+1)` nos dice que $y(2\pi) \approx 262.89$, lo que puede apreciarse en el gráfico que arroja el código previo.

Ejercicios Sección 2.2

Respuestas en página 86

1. Use el método de Euler con paso $h = 0.1$ para aproximar $y(0.2)$, siendo y la solución al PVI

$$y'' - 4y' + 4y = 0, \qquad y(0) = -2, \qquad y'(0) = 1.$$

   Halle la solución analítica del problema y compare $y(0.2)$ con y_2.

2. Use el método de Euler con paso $h = 0.1$ para aproximar $x(1.2)$, siendo x la solución al PVI

$$t^2x'' - 2tx' + 2x = 0, \qquad x(1) = 4, \qquad x'(1) = 9.$$

   Halle la solución analítica del problema en $(0, \infty)$ y compare $x(1.2)$ con x_2.

3. Escriba las fórmulas que se obtienen a partir de (2.9) y (2.10), para aplicar el método de Euler mejorado al sistema (2.16). Úselas para repetir lo realizado en los Ejercicios 1 y 2 con este método, para el mismo tamaño de paso.

4. Repita lo realizado en el Ejercicio 1 usando RK4.

5. Repita lo realizado en el Ejercicio 2 usando RK4.

6. **Sistema con constante de resorte variable.** El modelo de un sistema de resorte/masa que se somete a un ambiente en el cual la temperatura disminuye con rapidez está dado por la ecuación de Airy $4x'' + tx = 0$ (ver página 201 de [4]).

   a) ¿Qué comportamiento se espera luego de un período largo?

   b) >_ Compruebe su conjetura graficando la solución de la ecuación considerando diferentes variantes para las condiciones iniciales $x(0) = x_0$ y $x'(0) = y_0$. Utilice `ode45` y un `for` para que $x_0, y_0 \in \{-1, 0, 1\}$. Recuerde que `ode45` devuelve la información buscada para x en la primera *columna* de la matriz de salida. Investigue la instrucción `set(gca,'YDir','reverse')`, que puede resultar muy útil para problemas de resortes.

7. Considere siguiente PVI de orden 3:

$$y''' + 4y'' + 5y' + 2y = -4\,\mathrm{sen}(x) - 2\cos(x),$$
$$y(0) = 1, \qquad y'(0) = 0, \qquad y''(0) = -1.$$

   a) Reescriba el problema como un sistema de primer orden, con sus respectivos valores iniciales.

   b) >_ Grafique la solución y obtenga el valor de la variable de estado y en $x = 2.5$, con 6 cifras decimales.

   c) >_ Indique cuántas veces se anula la función $y'(x)$ en el intervalo $[0, 15]$. *Sugerencia:* vea el Ejemplo 13.

8. >_ **Resortes no lineales sin amortiguamiento.** Las ecuaciones diferenciales

$$x'' + x + x^3 = 0 \qquad \text{y} \qquad x'' + x - x^3 = 0$$

   modelan el movimiento de resortes no lineales sin amortiguamiento, uno duro y otro suave, respectivamente. Su origen puede encontrarse en las páginas 222 y 223 del libro [4].

   a) Resuelva numéricamente y grafique la solución de cada uno de los PVI no lineales de segundo orden originados por estas ecuaciones más las condiciones iniciales $x(0) = 2$, $x'(0) = -3$.

b) Repita con las condiciones iniciales $x(0) = 2$, $x'(0) = 0$.

c) En un mismo gráfico incluya las curvas correspondientes al resorte duro, en color rojo para el primer inciso, y en azul para el segundo. Compare con la Figura 5.3.2 a) del libro [4].

d) En otro gráfico incluya las curvas correspondientes al resorte suave, en color rojo para el primer inciso, y en azul para el segundo. Compare con la Figura 5.3.2 b) del libro [4].

9. **Resorte no lineal amortiguado.** (Ejercicios 9 y 10, Sección 5.3 de [4]) Consideraremos las ecuaciones para el movimiento de los resortes del ejercicio anterior, suponiendo un amortiguamiento proporcional a la velocidad instantánea:

$$x'' + x' + x + x^3 = 0 \qquad \text{y} \qquad x'' + x' + x - x^3 = 0.$$

>_ Use un método numérico para graficar las curvas solución que satisfacen las condiciones iniciales en cada caso, para $0 \leq t \leq 5$. Estime, a partir de lo obtenido, el comportamiento de cada sistema cuando $t \to \infty$.

a) Para el resorte duro: $x(0) = -3$ y $x'(0) = 4$; $x(0) = 0$ y $x'(0) = -8$.

b) Para el resorte suave: $x(0) = 0$ y $x'(0) = \frac{3}{2}$; $x(0) = -1$ y $x'(0) = 1$.

10. **Ecuación diferencial de Duffing.** (Ejercicios 11 y 12, Sección 5.3 de [4]) Considere el PVI

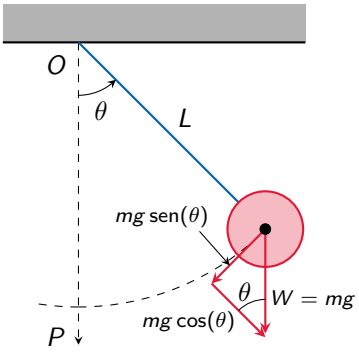
$$x'' + x + k_1 x^3 = 5\cos(t), \qquad x(0) = 1, \qquad x'(0) = 0,$$

que describe el movimiento de un sistema resorte/masa no lineal, no amortiguado y forzado en forma periódica.

a) >_ Grafique las curvas solución para $k_1 = 0.01, 1, 10, 100$. Exprese sus conclusiones.

b) >_ Mediante exploración gráfica, encuentre los valores de $k_1 < 0$ para los cuales la respuesta del sistema es oscilatoria.

11. **Péndulo no lineal.** Cualquier objeto que oscila de un lado a otro se llama *péndulo físico*. El *péndulo simple* es un caso particular de péndulo físico y consiste en una varilla de longitud L a la que se fija una masa m en el extremo. Para describir su movimiento en un plano vertical supondremos que la masa de la varilla es despreciable y que ninguna fuerza externa actúa sobre el sistema (por lo tanto, el rozamiento del aire se considera despreciable).



Siguiendo lo dicho en la página 223 de [4], el ángulo de desplazamiento θ del péndulo, medido desde la vertical OP, se considera positivo cuando se mide a la derecha de OP, y crece en sentido antihorario. Como allí se explica, el ángulo θ satisface la siguiente ecuación diferencial de orden dos:

$$\frac{d^2\theta}{dt^2} + \frac{g}{L}\operatorname{sen}(\theta) = 0.$$

Por simplicidad, para este ejercicio tomaremos $L = 9.81$ m, por lo que $\omega^2 = g/L = 1$.

>_ Resuelva esta ecuación numéricamente para graficar el movimiento en el intervalo de tiempo $[0, 12]$ (segundos) y explique, en cada uno de los siguientes casos, la situación física descripta (condiciones iniciales y evolución). Compare lo obtenido en los incisos a) y b) con las curvas de la Figura 5.3.4 de la página 224 del libro [4].

a) $\theta(0) = \frac{1}{2}$, $\theta'(0) = \frac{1}{2}$
b) $\theta(0) = \frac{1}{2}$, $\theta'(0) = 2$
c) $\theta(0) = 0.1$, $\theta'(0) = 0$
d) $\theta(0) = 0.7$, $\theta'(0) = 0$
e) $\theta(0) = 3$, $\theta'(0) = 0$
f) $\theta(0) = 3.5$, $\theta'(0) = 0$
g) $\theta(0) = 0$, $\theta'(0) = 1$
h) $\theta(0) = 0$, $\theta'(0) = 1.99$
i) $\theta(0) = 0$, $\theta'(0) = 2$
j) $\theta(0) = 0$, $\theta'(0) = 2.1$

12. >_ (Ejercicio 26, Sección 4.10 de [4]) Investigue en forma gráfica las soluciones de la ecuación

$$\frac{d^2\theta}{dt^2} + \frac{d\theta}{dt} + \operatorname{sen}(\theta) = 0,$$

sujeta a $\theta(0) = 0$ y $\theta'(0) = \theta_1$, para diferentes elecciones de $\theta_1 \geq 0$ y para $t \geq 0$. Proponga una interpretación física posible para el término $d\theta/dt$.

13. **Péndulo no lineal amortiguado libre.** (Ejercicio 13, Sección 5.3 de [4]) Considere el modelo del péndulo no lineal amortiguado libre dado por

$$\frac{d^2\theta}{dt^2} + 2\lambda\frac{d\theta}{dt} + \omega^2\operatorname{sen}(\theta) = 0.$$

>_ Utilice aproximaciones numéricas para graficar el movimiento en los casos sobreamortiguado ($\lambda^2 - \omega > 0$) y subamortiguado ($\lambda^2 - \omega < 0$).

a) Sobreamortiguado: use $\lambda = 2$, $\omega = 1$, $\theta(0) = 1$ y $\theta'(0) = 2$.

b) Subamortiguado: use $\lambda = 1/3$, $\omega = 1$, $\theta(0) = -2$ y $\theta'(0) = 4$.

14. **Péndulo en la Luna.** (Ejercicio 23, Sección 5.3 de [4]) ¿Un péndulo oscila más rápido en la Tierra o en la Luna? Considere el PVI

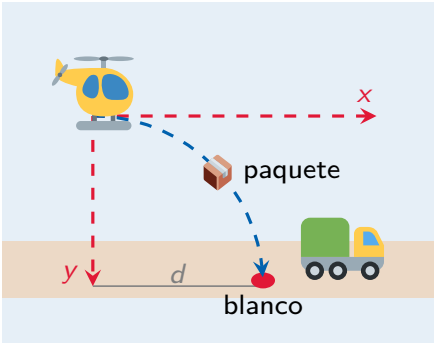
$$\frac{d^2\theta}{dt^2} + \frac{g}{L}\operatorname{sen}(\theta) = 0, \qquad \theta(0) = 1, \qquad \theta'(0) = 0.5.$$

a) Genere la curva solución con $L = 3$ y $g = 9.8$.

b) Repita con $L = 3$ y $g = 1.63$ (la gravedad en la Luna es aproximadamente $1/6$ de la gravedad de la Tierra).

c) ¿Qué péndulo oscila más rápido? ¿Cuál tiene mayor amplitud de movimiento?

15. **Suministro de ayuda.** (Ejercicio 20, Sección 5.3 de [4]) Un helicóptero que vuela a una velocidad constante v_0 suelta un paquete de suministros de ayuda a tierra. Suponga que el origen es el punto donde se libera el paquete, que el eje x positivo apunta hacia adelante y que el eje y positivo apunta hacia abajo. Si la posición del paquete está dada por $\vec{r}(t) = (x(t), y(t))$, entonces su velocidad es $\vec{v}(t) = (dx/dt, dy/dt)$. Asumiremos que las componentes horizontal y vertical de la resistencia al aire son proporcionales a $(dx/dt)^2$ y $(dy/dt)^2$, respectivamente.



Por la segunda ley de Newton en cada componente, si m es la masa del paquete, se tiene que

$$m\frac{d^2x}{dt^2} = -k\left(\frac{dx}{dt}\right)^2,$$

$$m\frac{d^2y}{dt^2} = mg - k\left(\frac{dy}{dt}\right)^2.$$

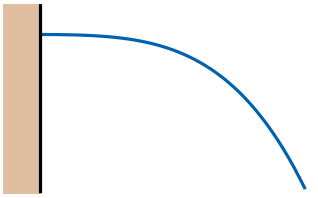
Además, se tienen las condiciones iniciales $x(0) = 0$, $x'(0) = v_0$, $y(0) = 0$, $y'(0) = 0$.

Suponga que el helicóptero vuela a una altitud de 300 metros y que su rapidez constante es 500 km/h. Además, asuma que la constante de proporcionalidad de la resistencia al aire es $k = 0.0053$ y que el paquete pesa 100 N. Determine, aproximadamente, qué distancia horizontal viaja el paquete, medida desde su punto de liberación hasta el punto donde pega en el suelo. *Sugerencia:* pueden resultar útiles los comandos `find` y `end`.

16. **Deflexión de una viga.** Consideremos una viga recta de longitud L, homogénea y con secciones transversales uniformes a lo largo de su longitud. Ubiquemos la viga en un sistema de coordenadas de forma que su eje de simetría corresponda a $y = 0$. Siguiendo lo explicado en la página 214, Sección 5.2 de [4], si se aplica una carga a la viga en un plano vertical que contiene al eje de simetría, la viga experimenta una distorsión. Para cada $0 \leq x \leq L$, la **curva de deflexión** o **curva elástica** $y(x)$ mide el desplazamiento vertical (positivo hacia abajo) del eje de simetría producido por la aplicación de dicha carga. El momento de flexión $M(x)$ satisface la ecuación diferencial

$$M(x) = EI\kappa,$$

donde $\kappa = \dfrac{y''}{(1+(y')^2)^{\frac{3}{2}}}$ es la curvatura, E es la constante de elasticidad del material, e I es el momento de inercia de cada sección transversal (también constante).



Supongamos que la viga está **en voladizo**, esto es, el extremo izquierdo está empotrado a una pared, mientras que el derecho está libre. Las condiciones que deben cumplirse en este caso son

$$y(0) = y'(0) = 0.$$

Si se aplica una carga P en el extremo libre, el momento flector es $M(x) = P(L - x)$.

Considere $L = 120$ cm, $E = 2.1 \times 10^6$ kg/cm^3, $I = 4250$ cm^4 y $P = 3000$ kg.

a) ¿En qué posición se produce el máximo desplazamiento con respecto al eje de simetría? Determine dicho desplazamiento.

b) ¿En qué posición la pendiente de la curva comienza a ser mayor que 0.0019?

2.3. Sistemas reducidos a primer orden

En la sección anterior vimos que toda ecuación diferencial de m-ésimo orden de la forma

$$y^{(m)} = g\left(x, y, y', \ldots, y^{(m-1)}\right)$$

se puede transformar en un sistema equivalente de m ecuaciones de primer orden como (2.1) usando las sustituciones $y_1 = y$, $y_2 = y'$, $y_3 = y''$, $\ldots$, $y_m = y^{(m-1)}$.

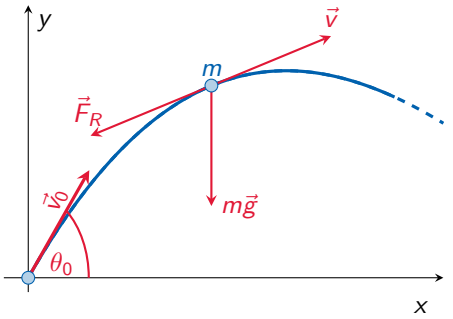
Entonces, todo sistema formado por n ecuaciones diferenciales de orden superior que puedan escribirse de dicha forma, se podrá reescribir de forma equivalente como un sistema de primer orden como (2.1), en donde la cantidad de ecuaciones dependerá del orden de cada una de las n ecuaciones del sistema de orden superior original. Por ejemplo, el sistema

$$x''' = f(t, x, x', x'', y, y')$$
$$y'' = g(t, x, x', x'', y, y')$$

será equivalente a otro formado por 5 ecuaciones de primer orden. Ilustraremos esto en el ejemplo siguiente.

Ejemplo 9. Suponga que una pelota de béisbol es bateada con una rapidez inicial v_0 y con ángulo de inclinación inicial θ_0. Suponga que, además de la aceleración hacia abajo de la gravedad, la pelota experimenta una aceleración debida a la resistencia del aire. Más precisamente, actúa una fuerza $\vec{F}_R$ con dirección opuesta al vector velocidad $\vec{v}$ y que tiene una magnitud kv^2, siendo $v = |\vec{v}|$ la rapidez de la pelota. Halle a qué distancia horizontal llegará al suelo la pelota y qué altura máxima alcanza, así como los instantes aproximados en que ocurre, suponiendo que $v_0 = 49$ m/s, $\theta_0 = 30°$, $m = 100$ g y $k = 0.00025$ kg/m.

Solución. Comencemos hallando las ecuaciones que describen el movimiento de la pelota.



Supongamos que la pelota parte desde el origen con velocidad inicial

$$\vec{v}_0 = (v_0 \cos(\theta_0), v_0 \operatorname{sen}(\theta_0)),$$

y sea $\vec{g} = (0, g)$, donde la constante $g = 9.8$ m/s^2 es la aceleración gravitacional.

Si $\vec{r}(t) = (x(t), y(t))$ es la posición de la pelota en el instante t, entonces su velocidad es $\vec{v} = (x', y')$ y su aceleración es $\vec{a} = (x'', y'')$. Por la segunda ley de Newton $\vec{F} = m\vec{a}$ se tiene que

$$m\vec{a} = -m\vec{g} - \vec{F}_r.$$

Así, las ecuaciones diferenciales del movimiento son:

$$x'' = -\frac{1}{m}F_{rx}, \qquad y'' = -g - \frac{1}{m}F_{ry}, \tag{2.20}$$

donde F_{rx} y F_{ry} son las componentes de $\vec{F}_r$ en las direcciones x e y, respectivamente. Para este caso, sabemos que $\vec{F}_r = kv(x', y')$, siendo $v = |\vec{v}| = \sqrt{(x')^2 + (y')^2}$. Así, las ecuaciones del movimiento son

$$x'' = -\frac{k}{m}vx', \qquad y'' = -g - \frac{k}{m}vy',$$

sujeto a $x(0) = y(0) = 0$, $x'(0) = v_0\cos(\theta_0)$, $y'(0) = v_0\,\mathrm{sen}(\theta_0)$.

Para escribir un sistema de primer orden equivalente, llamamos $y_1 = x$, $y_2 = y$, $y_3 = x'$, $y_4 = y'$. Entonces el sistema equivalente es

$$\begin{cases} y_1' = y_3, \\ y_2' = y_4, \\ y_3' = -\frac{k}{m}y_3\sqrt{y_3^2 + y_4^2}, \\ y_4' = -g - \frac{k}{m}y_4\sqrt{y_3^2 + y_4^2}. \end{cases}$$

Las condiciones iniciales se traducen en $y_1(0) = 0$, $y_2(0) = 0$, $y_3(0) = v_0\cos(\theta_0)$, $y_4(0) = v_0\,\mathrm{sen}(\theta_0)$.

El siguiente código permite obtener respuestas a los interrogantes planteados.

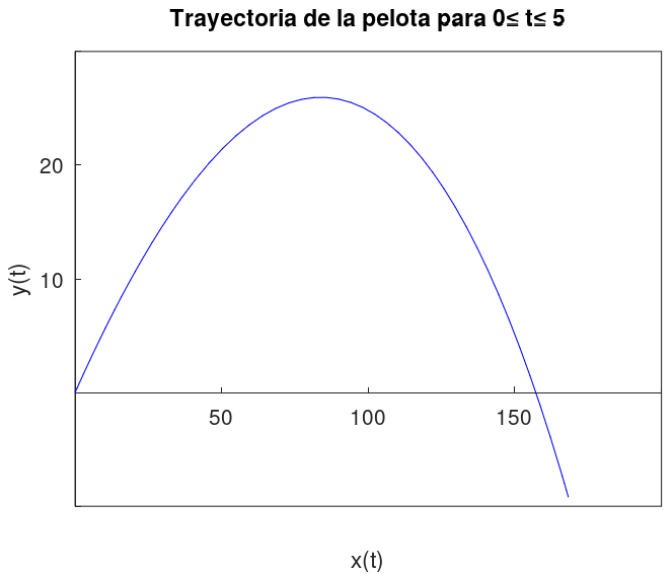
```
% Constantes del problema
g=9.8; k=0.00025;
m=0.1;        %en kg
v0=49;
theta0=30;    %en grados
inter=[0 5]; %intervalo de tiempo
N=500;        %cantidad de pasos

% Condiciones iniciales
y0=[0;0;v0*cosd(theta0);v0*sind(theta0)]; %arg. en grados

% Función del sistema
f=@(t,y) [y(3); y(4); -(k/m)*y(3)*((y(3)^2+y(4)^2)^(0.5));
-g-(k/m)*y(4)*((y(3)^2+y(4)^2)^(0.5))];

[t,y]=rk4(f,inter,y0,N);  %Aplico RK4 con h=0.01

% Gráfico y preferencias
plot(y(1,:),y(2,:),'-b'); % x vs. y
xlabel('x(t)');
ylabel('y(t)');
title('Trayectoria de la pelota para 0\leq t\leq 5');
%ejes centrados:
set(gca,'XAxisLocation','origin','YAxisLocation','origin');

% Preguntas del problema
T=find(y(2,:)>0);    %posiciones de los tiempos donde
                     %no llegó al piso
```

```
t0=T(end);              %me interesa el último
d=y(1,t0);              %distancia horizontal recorrida
[M,p]=max(y(2,:));      %Máxima altura y su posición

% Mostrar la leyenda con los resultados
disp(['Distancia horizontal recorrida: ',num2str(d),' metros'
    ]);
disp(['Tiempo transcurrido: ', num2str(t(t0)), ' segundos']);
disp(['Altura máxima alcanzada: ', num2str(M), ' metros']);
disp(['Tiempo en alcanzarla: ', num2str(t(p)), ' segundos']);
```

El código arroja las siguientes respuestas:

```
Distancia horizontal recorrida: 157.0228 metros
Tiempo transcurrido: 4.59 segundos
Altura máxima alcanzada: 25.9678 metros
Tiempo en alcanzarla: 2.22 segundos
```

Además, genera el gráfico siguiente, que representa la trayectoria de la pelota.



En problemas como los anteriores, una forma de ver si lo obtenido tiene sentido es considerar el problema sin resistencia al aire. En tal caso, tenemos

$$x(t) = v_0 \cos(\theta_0)t, \qquad y(t) = v_0 \operatorname{sen}(\theta_0)t - \frac{g}{2}t^2.$$

Con los datos del problema anterior, $y(t) = 0$ en $t = 0$ y en $t = 5$. Es decir, la pelota demora 5 segundos en llegar al suelo. Así, el alcance horizontal es $x(5) \approx 212$ metros. La altura máxima se alcanza cuando $t = 2.5$ segundos, y es de $y(2.5) \approx 30.6$ metros. Si consideramos resistencia al aire, la distancia horizontal recorrida, así como la altura máxima alcanzada, serán

menores. También deberán disminuir levemente los tiempos correspondientes. Esto coincide con las aproximaciones numéricas que obtuvimos.

Ejercicios Sección 2.3

Respuestas en página 100

1. Sean $(x(t), y(t))$ las coordenadas del movimiento de una partícula de masa m bajo la influencia de una fuerza dirigida hacia el origen con magnitud de $k/(x^2 + y^2)$. Esto es, actúa un campo de fuerzas inversamente proporcional al cuadrado de la distancia al origen. En estos casos, se cumple que

$$mx'' = \frac{-kx}{(x^2 + y^2)^{3/2}}, \qquad my'' = \frac{-ky}{(x^2 + y^2)^{3/2}}.$$

   a) Transforme el sistema en otro equivalente formado por ecuaciones diferenciales de primer orden.

   b) Para $k/m = 1$ y las condiciones iniciales $x(0) = 1$, $y(0) = 0$, $x'(0) = 0$, $y'(0) = 1$, grafique la trayectoria de la partícula para $0 \leq t \leq 200$, utilizando el método RK4 con paso $h = 0.01$. Observe que es una circunferencia de radio 1 alrededor del origen.

   c) Para las condiciones del inciso anterior, grafique utilizando `ode45` para $0 \leq t \leq 20$ y luego para $0 \leq t \leq 200$. ¿Qué observa? Investigue qué aspecto debería tener la trayectoria y qué puede estar ocurriendo, teniendo en cuenta que, en general, `ode45` es más preciso que RK4.

2. **Leyes de Kepler para el movimiento de planetas y satélites.** Considere un satélite en órbita elíptica alrededor de un planeta de masa M, y suponga que las unidades físicas se escogieron de modo que $G \cdot M = 1$, siendo G la constante gravitacional. Si el planeta está localizado en el plano xy, las ecuaciones del movimiento del satélite son

$$x'' = \frac{-x}{(x^2 + y^2)^{3/2}}, \qquad y'' = \frac{-y}{(x^2 + y^2)^{3/2}}.$$

   Si T denota el período de revolución del satélite, la tercera ley de Kepler dice que el cuadrado de T es proporcional al cubo del semieje mayor A de su órbita elíptica. En particular, cuando $G \cdot M = 1$, se tiene que

$$T^2 = 4\pi^2 A^3.$$

   Las condiciones iniciales $x(0) = 1$, $y(0) = 0$, $x'(0) = 0$, $y'(0) = 1$ corresponden a una órbita circular de radio $A = 1$, de modo que la ecuación anterior da $T = 2\pi$. Así, $x(t) = \cos(t)$ e $y(t) = \operatorname{sen}(t)$. Compruebe si lo obtenido en el inciso *b*) del ejercicio previo, resolviendo en $[0, \pi]$ con paso $h = \pi/1000$, concuerda de forma aproximada con los valores siguientes:

t	$x(t)$	$y(t)$
$\frac{\pi}{4}$	$\frac{1}{2}\sqrt{2}$	$\frac{1}{2}\sqrt{2}$
$\frac{\pi}{2}$	0	1
$\frac{3\pi}{4}$	$-\frac{1}{2}\sqrt{2}$	$\frac{1}{2}\sqrt{2}$
π	-1	0

3. (De [3]) Luego del accidente sufrido en un circo chileno en julio de 2018, donde el hombre bala salió despedido del cañón con demasiada potencia y cayó muchos metros después de la red de seguridad, la Comisión Internacional del Circo decidió diseñar un nuevo traje reglamentario, con un sistema de frenado inteligente. El mismo consiste en una fuerza de resistencia proporcional a la velocidad del individuo que se activa cuando este supera los 10 metros de altura. Más precisamente, llamando $\vec{r}(t) = (x(t), y(t))$ a la posición del hombre bala en un plano vertical, la fuerza resultante que actúa sobre el sujeto es

$$\vec{F} = \begin{cases} -m\vec{g}, & \text{si } y < 10, \\ -m\vec{g} - 20\vec{v}, & \text{si } y \geq 10, \end{cases}$$

donde $\vec{v}(t) = \frac{d\vec{r}}{dt}$ es el vector velocidad y $\vec{g} = (0, g)$, siendo $g = 9.8$ m/s^2 la constante de aceleración gravitacional.

a) Utilizando la segunda ley de Newton $\vec{F} = m\vec{a}$, escriba el problema a valores iniciales correspondiente para las variables de estado $x(t)$ e $y(t)$, considerando un sujeto de masa m que sale desde el origen con velocidad inicial $\vec{v}_0 = (v_0 \cos(\theta_0), v_0 \operatorname{sen}(\theta_0))$. Escriba el PVI como un sistema de primer orden. *Sugerencia:* vea lo realizado en el Ejemplo 9.

b) >_ Encuentre la altura máxima que alcanza el hombre bala y el tiempo aproximado en el cual la alcanza, si consideramos que los datos son $m = 75$ kg, $v_0 = 22$ m/s y $\theta_0 = \pi/3$. *Sugerencia:* vea el código del Ejemplo 9, y use que la función de Heaviside $\mathscr{U}(x - 10)$ se ingresa en Octave como `f=@(x) (x >= 10)`.

c) >_ Determine la distancia horizontal recorrida por el hombre bala desde que sale despedido del cañón hasta que toca la red de seguridad, que está colocada a 2 m de altura.

d) >_ ¿Cuánto tiempo permanece en el aire antes de hacer contacto con la red?

Capítulo 3

Problemas de segundo orden con valores en la frontera

En este capítulo veremos un método para aproximar la solución de problemas con valores en la frontera (PVF) de segundo orden, esto es, uno de la forma

$$y'' = f(x, y, y'), \qquad y(a) = \alpha, \qquad y(b) = \beta. \tag{3.1}$$

Además de estas condiciones de borde, conocidas como de tipo Dirichlet, consideraremos las llamadas de tipo Neumann y Robin.

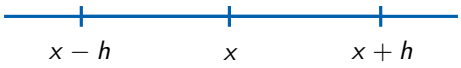
A diferencia de lo hecho para PVI de segundo orden, para resolver un PVF no escribiremos la ecuación de segundo orden como un sistema de primer orden. Además, no tenemos garantizada ahora la existencia de solución.

3.1. Aproximación por diferencias finitas

La idea del método que trabajaremos en la próxima sección consiste en reemplazar las derivadas por cocientes de diferencias. Para ello, recordemos que

$$y'(x) = \lim_{h \to 0} \frac{y(x+h) - y(x)}{h}.$$

La idea es tomar $h > 0$ pequeño y aproximar la derivada de distintas formas, que utilizaremos según conveniencia o necesidad.



Diferencia adelantada: $y'(x) \approx \dfrac{y(x+h)-y(x)}{h}$,

con error $= O(h)$ si $y \in \mathcal{C}^2$.

Diferencia atrasada: $y'(x) \approx \dfrac{y(x)-y(x-h)}{h}$,

con error $= O(h)$ cuando $y \in \mathcal{C}^2$.

Diferencia centrada; $y'(x) \approx \dfrac{y(x+h)-y(x-h)}{2h}$,

con error $= O(h^2)$ si $y \in \mathcal{C}^3$.

El orden de los errores al hacer estas aproximaciones se obtiene del desarrollo en series de Taylor y usando el teorema del residuo de Taylor. Veamos la primera para ilustrarlo. Si expandimos $y(x+h)$ obtenemos

$$y(x+h) = y(x) + y'(x)h + y''(x)\frac{h^2}{2} + y'''(x)\frac{h^3}{6} + \ldots \tag{3.2}$$

Luego, sustituyendo por esta expresión, nos queda que

$$\begin{aligned}\frac{y(x+h)-y(x)}{h} &= \frac{\left(y(x) + y'(x)h + y''(x)\frac{h^2}{2} + y'''(x)\frac{h^3}{6} + \ldots\right) - y(x)}{h} \\ &= y'(x) + \frac{h}{2}y''(x) + \frac{h^2}{6}y'''(x) + \ldots\end{aligned}$$

De esto se deduce que $y'(x)$ es igual a lo que llamamos "diferencia adelantada", más un término de error principal de orden $O(h)$ dado por $\frac{h}{2}y''(x)$. De forma similar se procede para la diferencia atrasada al escribir la expansión

$$y(x-h) = y(x) - y'(x)h + y''(x)\frac{h^2}{2} - y'''(x)\frac{h^3}{6} + \ldots \tag{3.3}$$

Para el error en la diferencia centrada se resta (3.3) de (3.2).

Al sumar (3.2) y (3.3) obtenemos la siguiente aproximación para la derivada segunda:

Diferencia centrada: $y''(x) \approx \dfrac{y(x+h)-2y(x)+y(x-h)}{h^2}$,

con error $= O(h^2)$ si $y \in \mathcal{C}^4$.

Ejemplo 10 (Aproximando derivadas con diferencias finitas)**.** Considere la función $f(x) = \operatorname{sen}(x)$. Aproxime su derivada en $x = \pi/4$ mediante diferencias finitas adelantadas, atrasadas y centradas para $h = 0.1$, $h = 0.01$ y $h = 0.001$. Use el valor exacto de la derivada para hallar el error absoluto en cada caso.

Solución. El valor exacto es $f'(\pi/4) = \cos(\pi/4) \approx 0.70711$. Organizamos los resultados obtenidos con las aproximaciones en la siguiente tabla, redondeadas a 5 cifras decimales.

h	adelantada	atrasada	centrada
0.1	0.6706	0.74125	0.70593
0.01	0.70356	0.71063	0.70709
0.001	0.70675	0.70746	0.70711

h	|error ad.|	|error at.|	|error cent.|
0.1	0.0365	0.03415	0.00118
0.01	0.00355	0.00352	0.00001
0.001	0.00035	0.00035	0

3.2. El método de diferencias finitas

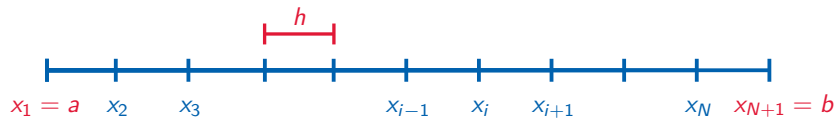
Consideremos el problema lineal de segundo orden con valores en la frontera dado por

$$y'' + P(x)y' + Q(x)y = f(x), \qquad y(a) = \alpha, \quad y(b) = \beta, \tag{3.4}$$

para $x \in [a, b]$. Como antes, dividimos $[a, b]$ en N subintervalos de igual longitud $h = (b - a)/N$, agregando los puntos

$$a = x_1 < x_2 < x_3 < \ldots < x_N < x_{N+1} = b,$$

donde $x_i = a + (i - 1) \cdot h$, para $i = 1, 2, \ldots, N + 1$.



Los puntos $x_2, x_3, \ldots, x_N$ se llaman **puntos interiores de la malla.** Nuestro objetivo es determinar los valores para y_i, que serán las aproximaciones para la solución buscada en los puntos de la malla. Esto es, $y_i \approx y(x_i)$. Nos interesa solo en los interiores, ya que las condiciones de borde nos dicen que $y_1 = \alpha$ e $y_{N+1} = \beta$. Para dichos puntos interiores vamos a reemplazar y' e y'' por las correspondientes diferencias finitas centradas:

$$y'(x_i) \approx \frac{y_{i+1} - y_{i-1}}{2h}, \qquad y''(x_i) \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}.$$

Para soluciones suficientemente suaves, el error aquí será del orden de h^2 y, a diferencia de lo ocurrido en los PVI, no habrá error acumulado. Esto sucede porque todas las aproximaciones se calcularán de forma simultánea al resolver un sistema, y no de forma iterativa, donde cada valor depende del anterior. Así, con el método de diferencias finitas que describiremos a continuación, tendremos que $|y_i - y(x_i)| \leq Ch^2$ para todo i.

Denotemos

$$P_i = P(x_i), \qquad Q_i = Q(x_i) \qquad \text{y} \qquad f_i = f(x_i).$$

Si en la ecuación diferencial incluida en (3.4) reemplazamos por esto y por las aproximaciones mediante diferencias finitas para y' e y'', obtenemos:

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + P_i \frac{y_{i+1} - y_{i-1}}{2h} + Q_i y_i = f_i,$$

para $i = 2, 3, \ldots, N$. Multiplicando por h^2 y agrupando, nos queda

$$y_{i-1}\left(1 - \tfrac{h}{2}P_i\right) + y_i\left(-2 + Q_i h^2\right) + y_{i+1}\left(1 + \tfrac{h}{2}P_i\right) = f_i h^2. \tag{3.5}$$

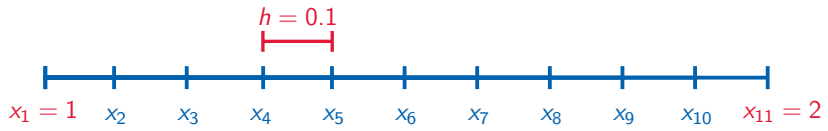
Estas $N-1$ ecuaciones, junto con las dos condiciones de borde, generan un sistema lineal de $N+1$ ecuaciones con $N+1$ incógnitas y_i. Al resolver este sistema habremos obtenido la solución aproximada al PVF dado en (3.4).

En el siguiente ejemplo, al igual que hicimos con los PVI, aplicaremos el método "a mano" para comprender su funcionamiento. Esto ayudará a su posterior implementación en Octave, para lo que necesitaremos el manejo de varias instrucciones y comandos.

Ejemplo 11. Use (3.5) con $N = 10$ para aproximar la solución de

$$y'' + 3y' + 2y = 4x^2, \qquad y(1) = 1, \quad y(2) = 6.$$

Solución. Aquí tenemos $P(x) = 3$, $Q(x) = 2$ y $f(x) = 4x^2$. Además, $h = (2-1)/10 = 0.1$. La malla queda



Por lo tanto, los puntos interiores son $x_2 = 1.1$, $x_3 = 1.2$, $x_4 = 1.3$, $x_5 = 1.4$, $x_6 = 1.5$, $x_7 = 1.6$, $x_8 = 1.7$, $x_9 = 1.8$ y $x_{10} = 1.9$. De (3.5) y las condiciones de frontera, tenemos:

$$
\begin{aligned}
y_1 &= 1 \\
0.85y_1 - 1.98y_2 + 1.15y_3 &= 0.0484 \\
0.85y_2 - 1.98y_3 + 1.15y_4 &= 0.0576 \\
0.85y_3 - 1.98y_4 + 1.15y_5 &= 0.0676 \\
0.85y_4 - 1.98y_5 + 1.15y_6 &= 0.0784 \\
0.85y_5 - 1.98y_6 + 1.15y_7 &= 0.0900 \\
0.85y_6 - 1.98y_7 + 1.15y_8 &= 0.1024 \\
0.85y_7 - 1.98y_8 + 1.15y_9 &= 0.1156 \\
0.85y_8 - 1.98y_9 + 1.15y_{10} &= 0.1296 \\
0.85y_9 - 1.98y_{10} + 1.15y_{11} &= 0.1444 \\
y_{11} &= 6.
\end{aligned}
$$

Notar que, incluyendo las ecuaciones para y_1 e y_{11}, lo anterior puede verse como un sistema de 11 ecuaciones con 11 incógnitas y_i para $i = 1, 2, \ldots, 11$, cuya matriz de coeficientes es:

$$
A = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.85 & -1.98 & 1.15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.85 & -1.98 & 1.15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.85 & -1.98 & 1.15 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.85 & -1.98 & 1.15 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.85 & -1.98 & 1.15 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.85 & -1.98 & 1.15 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.85 & -1.98 & 1.15 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.85 & -1.98 & 1.15 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.85 & -1.98 & 1.15 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}.
$$

Con Octave es muy sencillo resolver un sistema de la forma $A\mathbf{y} = \mathbf{b}$. Simplemente se debe escribir `y=A\b`. El resultado al que se llega para los nodos interiores es $y_2 = 2.4047$, $y_3 = 3.4432$, $y_4 = 4.2010$, $y_5 = 4.7469$, $y_6 = 5.1359$, $y_7 = 5.4124$, $y_8 = 5.6117$, $y_9 = 5.7620$, $y_{10} = 5.8855$.

El PVF del Ejemplo 11 puede resolverse analíticamente: la solución general de la ecuación diferencial es $y(x) = c_1 e^{-x} + c_2 e^{-2x} + 2x^2 - 6x + 7$. En la Figura 3.1 se incluye el gráfico de la solución que cumple con las condiciones de frontera, junto con el de las aproximaciones obtenidas.

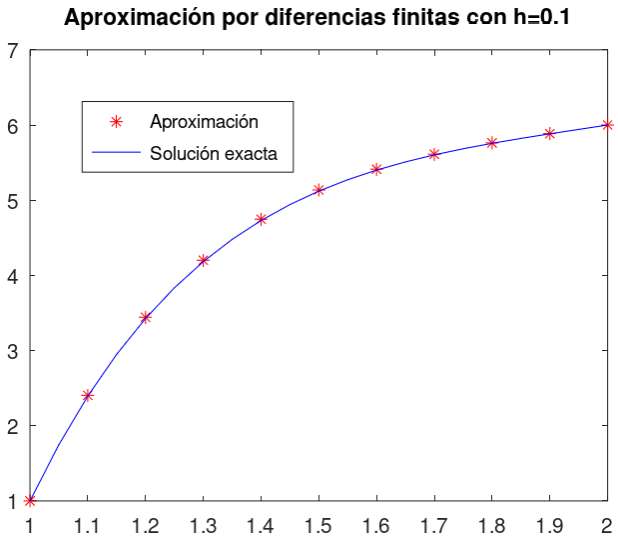


Figura 3.1: Gráfico de los resultados del Ejemplo 11.

La matriz A del ejemplo anterior es una **matriz tridiagonal**, ya que los elementos distintos de cero están únicamente en la diagonal principal y en las diagonales adyacentes, por encima y por debajo de esta. Este tipo de matrices dispersas son comunes en álgebra lineal numérica y en la resolución de problemas de física computacional, especialmente cuando se utilizan diferencias finitas para aproximar ecuaciones diferenciales (como la ecuación de Poisson o la ecuación del calor que veremos después), particularmente en problemas unidimensionales. Debido a su estructura, existen algoritmos y reglas específicas que permiten operar con ellas de manera mucho más eficiente que con matrices genéricas. Dedicaremos el resto de la sección a analizar cómo construir este tipo de matrices empleando Octave.

En el ejemplo previo, dado que P y Q son funciones constantes, cada fila interior de A (esto es, si no tenemos en cuenta la primera ni la última fila) contiene siempre los mismos tres elementos no nulos. Sin embargo, esto no siempre ocurre cuando P y Q varían. Para poder visualizar la forma general de la matriz A, escribamos la correspondiente al PVF (3.4) para $N = 5$. Los puntos de la malla son

$$x_1 = a, \qquad x_2, \qquad x_3, \qquad x_4, \qquad x_5, \qquad x_6 = b.$$

Las condiciones de frontera son $y_1 = \alpha$ e $y_6 = \beta$. Para $i = 2, 3, 4$ y 5 usamos (3.5). Estas 6 igualdades pueden escribirse juntas en forma matricial como

$$A \cdot \mathbf{y} = \mathbf{b},$$

donde

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1-\frac{h}{2}P_2 & -2+Q_2h^2 & 1+\frac{h}{2}P_2 & 0 & 0 & 0 \\ 0 & 1-\frac{h}{2}P_3 & -2+Q_3h^2 & 1+\frac{h}{2}P_3 & 0 & 0 \\ 0 & 0 & 1-\frac{h}{2}P_4 & -2+Q_4h^2 & 1+\frac{h}{2}P_4 & 0 \\ 0 & 0 & 0 & 1-\frac{h}{2}P_5 & -2+Q_5h^2 & 1+\frac{h}{2}P_5 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{pmatrix} \quad \text{y} \quad \mathbf{b} = \begin{pmatrix} \alpha \\ f_2h^2 \\ f_3h^2 \\ f_4h^2 \\ f_5h^2 \\ \beta \end{pmatrix}$$

Esto nos permite visualizar la forma general de la matriz A del método de diferencias finitas para (3.4), así como la del vector de términos independientes.

Construcción de la matriz en Octave

El objetivo ahora es poder construir la matriz A, así como el vector de términos independientes, de forma automática conociendo N y las funciones P, Q y f. Para ello iremos explorando distintos comandos. Algunos no los usaremos de forma directa, pero ayudarán a comprender a los otros. Para la construcción de la matriz avanzaremos desde casos simples hasta el más general.

El comando `ones(m,n)`. Construye una matriz de m filas y n columnas, con un 1 en cada lugar.

```
M=ones(3,2)
M =

   1   1
   1   1
   1   1
```

El comando `diag(v)`. Dado un vector **v** de n componentes (fila o columna), este comando construye una matriz cuadrada de dimensión n, que tiene a los elementos de **v** en su diagonal.

```
v=[1 2 3];
diag(v)
ans =

Diagonal Matrix
```

```
1    0    0
0    2    0
0    0    3
```

El comando `diag(v,k)`. Si k es un número entero, este comando construirá una matriz cuadrada que tiene a los elementos de **v** en la k-ésima diagonal, hacia arriba si k es positivo, y hacia abajo si es negativo. Así, `diag(v,0)` equivale a `diag(v)`. Se entenderá mejor con los siguientes ejemplos, en donde **v** es el vector ingresado previamente.

```
diag(v,1)
ans =

   0    1    0    0
   0    0    2    0
   0    0    0    3
   0    0    0    0

diag(v,2)
ans =

   0    0    1    0    0
   0    0    0    2    0
   0    0    0    0    3
   0    0    0    0    0
   0    0    0    0    0

diag(v,-1)
ans =

   0    0    0    0
   1    0    0    0
   0    2    0    0
   0    0    3    0
```

El comando `spdiags(B, d, m, n)`. Se usa para crear matrices dispersas (matrices donde la mayoría de los elementos son cero) a partir de los elementos que se encuentran en las diagonales. Aquí:

- `B` es la matriz cuyas columnas irán en las diagonales especificadas;
- `d` es un vector que indica las diagonales donde las columnas de `B` serán colocadas. Por ejemplo, `d=0` es la diagonal principal, `d=1` es la diagonal superior, etc;
- `m` es el número de filas de la matriz resultante;
- `n` es el número de columnas de la matriz resultante.

Además, solo se almacenan los valores no nulos para ahorrar memoria. Si queremos trabajar con la matriz completa, podemos usar `full(A)` para convertirla en una matriz densa (lo usaremos para observar el resultado y entender mejor).

Nos vamos a detener en este comando ya que será fundamental para construir las matrices necesarias para el método de diferencias finitas, por lo que necesitamos comprender su uso y tener en cuenta algunos detalles.

- ✔ Aunque m y n pueden ser distintos para este comando, las matrices que necesitaremos construir serán cuadradas de orden $N+1$.
- ✔ Como queremos construir una matriz tridiagonal, siempre usaremos `d=[-1,0,1]`.
- ✔ Para obtener la matriz A del método de diferencias finitas haremos primero una construcción general y, luego, corregiremos la primera y la última fila de acuerdo con las condiciones de borde.
- ✔ Si `B` es más grande que lo necesario para las diagonales de la matriz dispersa resultante, Octave elimina automáticamente los valores que no caben. Será importante comprender qué elementos descarta, ya que esto varía según si se trata de diagonales superior o inferiores. Veamos un ejemplo concreto.

```
B=[1 2 3;4 5 6;7 8 9]
B =

   1   2   3
   4   5   6
   7   8   9

d=[-1,0,1];
A=spdiags(B,[-1,0,1],3,3);
full(A)
ans =

   2   6   0
   1   5   9
   0   4   8
```

Notar que el segundo elemento de `d` es cero, por lo que la segunda columna de `B` va completa a la diagonal principal. Con esa no hay problema, ya que no sobran elementos. La primera columna de `B` debe ir a la primera diagonal **inferior** según indica `d`, pero sobra un elemento. En este caso, Octave **ignora el último.** También sobra un elemento en la tercera columna de `B`, la que debe ir en la diagonal **superior** según indica el vector `d`. Ahora, octave **elimina el primero.** Esto es un poco más complejo cuando sobran más elementos, pero solo necesitamos entender este caso aquí.

Lamentablemente, necesitaremos que Octave haga justo al revés. Vamos a ilustrarlo para comprenderlo y ver cómo se soluciona. Supongamos que tenemos los vectores columna

$$\mathbf{v}_1 = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}, \qquad \mathbf{v}_2 = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}, \qquad \mathbf{v}_3 = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}.$$

Si en Octave escribimos `A=spdiags([v1 v2 v3],[-1,0,1],3,3)`, el resultado será la matriz

$$A = \begin{pmatrix} b_1 & c_2 & 0 \\ a_1 & b_2 & c_3 \\ 0 & a_2 & b_3 \end{pmatrix},$$

eliminando los valores a_3 y c_1 (compare con el ejemplo previo). ¿Cómo podría hacer, dados esos mismos vectores $\mathbf{v}_i$ para generar con el comando `spdiags` la matriz

$$M = \begin{pmatrix} b_1 & c_1 & 0 \\ a_2 & b_2 & c_2 \\ 0 & a_3 & b_3 \end{pmatrix},$$

en la que se perdieron a_1 y c_3? **Observe la concordancia de los subíndices con el número de fila**, tal como en la matriz del método de diferencias finitas. Ilustremos la pregunta con un ejemplo.

```
v1=[1;2;3];
v2=[4;5;6];
v3=[7;8;9];
A=spdiags([v1,v2,v3],[-1,0,1],3,3);
full(A)
ans =

   4   8   0
   1   5   9
   0   2   6
```

Nos gustaría, con el mismo comando, obtener $M = \begin{pmatrix} 4 & 7 & 0 \\ 2 & 5 & 8 \\ 0 & 3 & 6 \end{pmatrix}$. La respuesta es bastante simple: solo debemos desplazar hacia arriba los elementos de $\mathbf{v}_1$ (perdemos el elemento a_1 pero no lo necesitamos), y hacia abajo los de $\mathbf{v}_3$ (perdemos c_3 pero no importa), y completar con ceros (en cada vector, el cero está en el lugar que el comando ignora), escribiendo:

```
w1=[v1(2:end);0];      w3=[0;v3(1:end-1)];
```

A continuación podemos observar el efecto de estos desplazamientos:

```
v=[1;2;3;4;5;6];
u=[0;v(1:end-1)];
w=[v(2:end);0];
u', w'
ans =

   0   1   2   3   4   5

ans =

   2   3   4   5   6   0
```

Entonces, para obtener la matriz M es suficiente con escribir la instrucción `M=spdiags([w1 v2 w3],[-1,0,1],3,3)`:

```
v1=[1;2;3];
v2=[4;5;6];
v3=[7;8;9];
A=spdiags([v1,v2,v3],[-1,0,1],3,3);
full(A)
ans =

   4   8   0
   1   5   9
   0   2   6

w1=[v1(2:end);0];
w3=[0;v3(1:end-1)];
M=spdiags([w1 v2 w3],[-1,0,1],3,3);
full(M)
ans =

   4   7   0
   2   5   8
   0   3   6
```

Una forma alternativa de construir la matriz M del código anterior es invertir entre sí los roles de v_1 y v_3, y transponer luego el resultado obtenido:

```
v1=[1;2;3];
v2=[4;5;6];
v3=[7;8;9];
B=spdiags([v1,v2,v3],[1,0,-1],3,3); %Se intercambian 1 con -1
M=B';
full(M)
ans =

   4   7   0
   2   5   8
   0   3   6
```

Estamos en condiciones de construir de forma automática la matriz A en Octave. Comencemos con el caso más simple que es suponer $P = Q = 0$. Así, para $N = 5$ la matriz es

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Queremos darle instrucciones a Octave para que, dado N arbitrario, construya esta matriz. Como mencionamos, primero se crea una matriz que tiene las filas interiores correctas y, luego, se corrigen la primera fila y la última:

```
% Construcción de la matriz cuando P=Q=0
N=5;

v1=ones(N+1,1); %vector columna formado por unos
v2=-2*v1;       %dejamos listo para el caso general
v3=v1;          %dejamos listo para el caso general

d=[-1,0,1]; B=[v1,v2,v3];
A=spdiags(B,d,N+1,N+1);

%Corregimos fila 1
A(1,1:2)=[1 0];

%Corregimos fila N+1
A(N+1,N:N+1)=[0 1];

full(A) %para ver y verificar
```

El resultado, tal como queríamos, es

```
ans =

  1   0   0   0   0   0
  1  -2   1   0   0   0
  0   1  -2   1   0   0
  0   0   1  -2   1   0
  0   0   0   1  -2   1
  0   0   0   0   0   1
```

Para el caso general solo debemos modificar adecuadamente los vectores v1, v2 y v3, sin olvidar las eliminaciones que hace el comando spdiags. Para ello hemos elegido el primer método presentado, pero el lector puede utilizar el que prefiera. La corrección debe hacerse en cualquiera de los casos. Ahora debemos contar previamente con los valores de h, P_i y Q_i. Lo ilustramos a continuación con el código y la salida correspondientes al Ejemplo 11.

```
% Construcción de la matriz para el caso general
N=10;
a=1; b=2;
P= @(x) 3* ones(size(x)); %función constante en este caso
Q= @(x) 2* ones(size(x)); %función constante en este caso

h=(b-a)/N;
x=linspace(a,b,N+1);
p=P(x)'; %vector columna Pi
q=Q(x)'; %vector columna Qi

v1=ones(N+1,1)-h*0.5*p;     %vector columna para diag inferior
w1=[v1(2:end);0];           %desplazo arriba porque spdiags
                            %ignora el último
v2=-2*ones(N+1,1)+h^2*q;    %vector columna para diag pincipal
v3=ones(N+1,1)+h*0.5*p;     %vector columna para diag superior
w3=[0;v3(1:end-1)];         %desplazo abajo porque spdiags
                            %ignora el primero

d=[-1,0,1]; B=[w1,v2,w3];
```

```
A=spdiags(B,d,N+1,N+1);

%Corregimos fila 1
A(1,1:2)=[1 0];

%Corregimos fila N+1
A(N+1,N:N+1)=[0 1];
```

Escribiendo `full(A)` podemos ver que la salida coincide con la matriz calculada en dicho ejemplo.

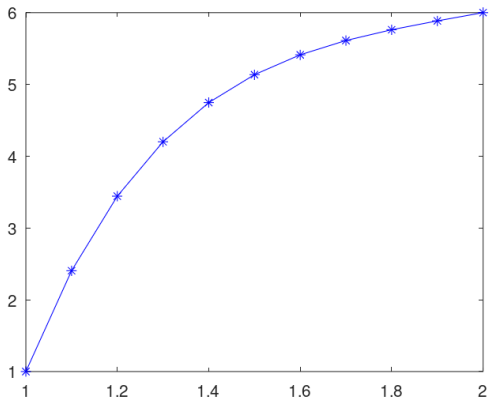
Aquí no se hubiera notado si no hacíamos los desplazamientos, ya que las funciones P y Q son constantes, pero no será correcto cuando no lo sean.

Ya podemos resolver el sistema que permite aproximar los valores y_i. Para eso solamente falta ingresar el vector de los términos independientes y resolver. Con las siguientes líneas hacemos esto y, además, el gráfico de la curva solución.

```
Alfa=1; Beta=6; % CF
f= @(x) 4*x.^2; % función del lado derecho

F=h^2.*f(x); F(1)=Alfa; F(N+1)=Beta; %término independiente
                                     %como fila

% Resolución y gráfico
y=A\F';
plot(x,y,'-*b')
```

Los valores que se obtienen para y_i coinciden con los dados Ejemplo 11:

```
y
y =

   1.0000
   2.4047
   3.4432
   4.2010
   4.7469
   5.1359
   5.4124
   5.6117
   5.7620
   5.8855
   6.0000
```



Hemos construido así, uniendo lo dos últimos códigos, un algoritmo que permite aproximar mediante el **método de diferencias finitas** la solución para el PVF (3.4), que es un problema con condiciones de frontera de tipo Dirichlet. Sin embargo, existen otros tipos de problemas.

A continuación presentamos las condiciones de frontera más frecuentes. Volveremos a ellas en la sección siguiente, interpretadas en más detalle en términos de la difusión del calor.

Condición de tipo Dirichlet. Se especifica el valor de la solución en la frontera del dominio: $y(a) = \alpha$. En el problema del calor, esto podría representar el hecho de mantener una barra a una temperatura fija en uno de los extremos.

Condición de tipo Neumann. Se especifica el valor de la derivada de la solución en la frontera: $y'(a) = m$. En el problema del calor puede representar aislamiento térmico tomando $m = 0$ (flujo nulo).

Condición de tipo Robin. Es una combinación de las condiciones de Dirichlet y Neumann. Se especifica una relación lineal entre la solución y su derivada en la frontera: $Cy(a) + Dy'(a) = G$. En problemas de transferencia de calor, esto puede representar una pérdida de calor proporcional a la diferencia de temperatura con el ambiente.

De forma análoga, se definen las condiciones de frontera para el extremo derecho del dominio $x = b$, y pueden combinarse con las condiciones en el extremo izquierdo según el caso, pudiendo variar el tipo en cada uno de ellos.

El método de diferencias finitas que trabajamos previamente es para problemas con condiciones de tipo Dirichlet en ambos valores de la frontera. A continuación veremos cómo varía el método cuando aparece la derivada en una de las condiciones.

Cabe señalar que las condiciones de frontera solamente modificarán la primera o la última ecuación (es decir, la primera o la última fila de la matriz A), según si la derivada aparece en la frontera izquierda o derecha, respectivamente. Las ecuaciones (3.5) para los nodos centrales quedarán iguales.

Consideremos nuevamente la ecuación

$$y'' + P(x)y' + Q(x)y = f(x) \tag{3.6}$$

para $x \in [a, b]$, pero supongamos que ahora tenemos la condición de frontera $y'(a) = m$ en lugar de $y(a) = \alpha$. Una primera solución es reemplazar $y'(x_1)$ por la diferencia finita adelantada, esto es,

$$y'(x_1) \approx \frac{y_2 - y_1}{h}.$$

Igualando esta expresión a la condición de borde, se obtiene la ecuación

$$-y_1 + y_2 = m \cdot h,$$

que modificaría la primera fila de la matriz A del sistema. Si la derivada apareciera en el extremo derecho, procedemos de igual forma aproximando $y'(x_{N+1})$ por la diferencia finita atrasada.

El procedimiento anterior es válido, pero ahora el método tiene error global de orden h, ya que una aproximación de menor orden en los nodos de borde puede propagarse al resto de la solución y reducir el orden global del método. Podemos evitar esto y mantener el orden h^2 agregando un **nodo ficticio** x_0 a la izquierda de x_1, y aproximando $y'(x_1)$ mediante la diferencia finita centrada:

$$y'(x_1) \approx \frac{y_2 - y_0}{2h},$$

donde y_0 representa el valor de la solución en x_0. Despejando, se obtiene que

$$y_0 \approx y_2 - 2hy'(x_1) = y_2 - 2hm, \tag{3.7}$$

debido a la condición de frontera establecida. También, tenemos que

$$y''(x_1) \approx \frac{y_2 - 2y_1 + y_0}{h^2}.$$

Reemplazando $y'(x_1)$ e $y''(x_1)$ en (3.6) por sus aproximaciones por diferencias finitas, tenemos que (3.5) también debe cumplirse para $i = 1$. Esto es

$$y_0\left(1 - \tfrac{h}{2}P_1\right) + y_1\left(-2 + Q_1h^2\right) + y_2\left(1 + \tfrac{h}{2}P_1\right) = f_1h^2. \tag{3.8}$$

Finalmente, usando (3.7) eliminamos y_0 de esta igualdad, obteniendo

$$(y_2 - 2hm)\left(1 - \tfrac{h}{2}P_1\right) + y_1\left(-2 + Q_1h^2\right) + y_2\left(1 + \tfrac{h}{2}P_1\right) = f_1h^2.$$

Al reagrupar obtenemos la ecuación para modificar la primera fila de la matriz A:

$$\left(-2 + Q_1h^2\right)y_1 + 2y_2 = f_1h^2 + 2hm\left(1 - \tfrac{h}{2}P_1\right). \tag{3.9}$$

Observemos que el nodo ficticio ya no forma parte del método, solo fue un apoyo temporal para poder construirlo. Así, el número de filas de la matriz, al igual que el número de incógnitas, continúa siendo $N + 1$.

Ejemplo 12 (Caso general para condiciones de Robin o Neumann)**.** Considere el PVF con ambas condiciones de tipo Robin

$$\begin{aligned} &y'' + P(x)y' + Q(x)y = f(x), \\ &C_1y(a) + D_1y'(a) = G_1, \quad C_2y(b) + D_2y'(b) = G_2, \end{aligned} \tag{3.10}$$

para $x \in [a, b]$. Aquí tanto D_1 como D_2 son coeficientes no nulos[1]. Obtenga las ecuaciones que permiten resolver mediante el método de diferencias finitas manteniendo el orden h^2, y construya el código correspondiente en Octave.

Solución. Seguiremos las mismas líneas que en lo previo al ejemplo, dejando los detalles menores como ejercicio para el lector. De la aproximación por diferencias finitas centradas para $y'(x_1)$ y la condición de frontera en $x = a$, tenemos que

$$y_0 = y_2 - \frac{2h}{D_1}\left(G_1 - C_1y_1\right).$$

Reemplazando esta expresión en (3.8) y agrupando, obtenemos

$$\left[\frac{2hC_1}{D_1}\left(1 - \tfrac{h}{2}P_1\right) - 2 + Q_1h^2\right]y_1 + 2y_2 = f_1h^2 + 2h\frac{G_1}{D_1}\left(1 - \tfrac{h}{2}P_1\right). \tag{3.11}$$

Similarmente, para la condición en el extremo derecho agregamos un nodo ficticio x_{N+2} a la derecha de x_{N+1}. Haciendo

$$y'(x_{N+1}) \approx \frac{y_{N+2} - y_N}{2h},$$

[1] Si alguno fuera nulo, usamos para esa frontera lo trabajado al inicio de la sección. Por ejemplo, si $D_1 = 0$, entonces debe ser $C_1 \neq 0$, y tenemos la condición de borde de tipo Dirichlet $y_1 = G_1/C_1$ (ver Ejercicio 2).

e imponiendo la condición de frontera, tenemos que

$$y_{N+2} = y_N + \frac{2h}{D_2}\left(G_2 - C_2 y_{N+1}\right).$$

Reemplazando esto en la expresión obtenida tomando $i = N+1$ en (3.5), obtenemos

$$2y_N + \left[\frac{-2hC_2}{D_2}\left(1 + \tfrac{h}{2}P_{N+1}\right) - 2 + Q_{N+1}h^2\right] y_{N+1} = f_{N+1}h^2 - 2h\frac{G_2}{D_2}\left(1 + \tfrac{h}{2}P_{N+1}\right).$$

✔ Observe que las condiciones de frontera de Robin generalizan a las de Neumann, pues basta con tomar $C_j = 0$ para $j = 1, 2$. Por ejemplo, tomando $C_1 = 0$, $D_1 = 1$ y $G_1 = m$ en (3.11) se obtiene (3.9). Por lo tanto, la matriz construida en el ejemplo previo funciona para cualquier tipo combinación de condiciones de frontera que involucren a la derivada (Robin o Neumann).

Las fórmulas obtenidas en el Ejemplo 12 permiten reemplazar las filas 1 y $N+1$, respectivamente, de la matriz A construida previamente para el caso de Dirichlet, así como los correspondientes elementos del vector de términos independientes. Recordemos que estas dos filas se corrigen luego de la creación, así que la modificación del código resulta muy sencilla.

```
% Método de diferencias finitas para condiciones Robin/Neumann
% y''+P(x)y'+Q(x)y=f(x)
% C1y(a)+ D1y'(a)=G1  condición de borde izquierda
% C2y(b)+ D2y'(b)=G2  condición de borde derecha
% C1 y/o C2 pueden ser cero (Neumann)
%¡¡¡ni D1 ni D2 pueden ser nulas!!!!

%% Cambiar este bloque por los datos del problema %%%
N=10;            % Cantidad de pasos
a=1; b=2;        % intervalo [a,b]
C1=1; D1=1; G1=1;  % CB 1
C2=1; D2=1; G2=1;  % CB 2
P= @(x) 3* ones(size(x)); %función constante P(x) = 3
Q= @(x) 2* ones(size(x)); %función constante Q(x) = 2
f= @(x) 4*x.^2; %función  f(x)=4x^2
%%%%%%%%%%%%%%%%%%%%%%%%%%

h=(b-a)/N;
x=linspace(a,b,N+1);
p=P(x)'; %vector columna Pi
q=Q(x)'; %vector columna Qi

% Construcción de la matriz
v1=ones(N+1,1)-h*0.5*p;     %vector columna para diag inferior
w1=[v1(2:end);0];           %desplazo arriba porque spdiags
                            %ignora el último
v2=-2*ones(N+1,1)+h^2*q;    %vector columna para diag ppal
v3=ones(N+1,1)+h*0.5*p;     %vector columna para diag superior
w3=[0;v3(1:end-1)];         %desplazo abajo porque spdiags
                            %ignora el primero

d=[-1,0,1]; B=[w1,v2,w3];
A=spdiags(B,d,N+1,N+1);
```

```
% Corregimos fila 1
A(1,1:2)=[2*h*C1*(1-h*0.5*p(1))/D1-2+q(1)*h^2 2];

% Corregimos fila N+1
A(N+1,N:N+1)=[2 -2*h*C2*(1+h*0.5*p(N+1))/D2-2+q(N+1)*h^2];

% Vector de términos independientes
Fa=2*h*G1*(1-h*0.5*p(1))/D1;
Fb=2*h*G2*(1+h*0.5*p(N+1))/D2;
F=h^2.*f(x);  %está como fila
F(1)=F(1)+Fa; F(N+1)=F(N+1)-Fb; %corrección primero y último

% Resolución y gráfico
y=A\F';
plot(x,y,'-*b')
```

Código 3.1: Método de diferencias finitas para condiciones de tipo Neumann/Robin.

Método de tanteo o disparo

Otro método para aproximar soluciones de un PVF como (3.1) consiste en reemplazarlo por un PVI

$$y'' = f(x, y, y'), \qquad y(a) = \alpha, \qquad y'(a) = m_1.$$

El número m_1 es solo una suposición del valor de la pendiente de la curva desconocida en el punto $(a, y(a))$. Se aplica entonces cualquiera de los métodos presentados para construir la solución de este PVI, obteniendo así una aproximación β_1 para $y(b)$. Si β_1 se acerca a β con una tolerancia previamente especificada, se detiene el cálculo. De lo contrario, se repite el procedimiento con una suposición distinta $y'(a) = m_2$, para obtener una segunda aproximación β_2 para $y(b)$, y así sucesivamente. Si la ecuación diferencial y las condiciones de frontera son lineales, puede demostrarse que basta con realizar dos intentos para obtener la aproximación buscada, la cual se expresa como una combinación lineal de las soluciones aproximadas obtenidas en cada intento.

En pocas palabras, este método consiste en "adivinar" valores iniciales, resolver como un PVI y ajustar las conjeturas hasta que se satisfagan, aproximadamente, las condiciones de frontera. De aquí proviene su nombre, que en inglés se conoce como *shooting method*.

Su principal ventaja es que, en general, es simple de implementar, aun si el PVF involucra una ecuación diferencial no lineal. El método de diferencias finitas, por el contrario, requeriría en ese caso resolver sistemas de ecuaciones no lineales, lo que puede resultar muy complejo en la práctica. Sin embargo, el método de disparo requiere iteraciones y ajustes, que suele ser computacionalmente costoso. Otra desventaja es que resulta ineficiente o inestable en problemas con comportamiento altamente sensible a las condiciones iniciales. Pese a esto, será una herramienta útil para aproximar soluciones de PVF en los que los métodos de diferencias finitas presentados hasta aquí no resulten aplicables (ver Ejercicio 4).

Ejercicios Sección 3.2

Respuestas en página 104

Los dos primeros ejercicios, cuya solución exacta puede determinarse de forma explícita con métodos clásicos, tienen como objetivo verificar que los códigos estén correctamente implementados en Octave.

1. Considere el PVF

$$x^2y'' - xy' + y = \ln(x), \qquad y(1) = 0, \quad y(2) = -2.$$

   a) Halle analíticamente la solución exacta.

   b) Use el método de diferencias finitas con $N = 8$ para resolver numéricamente el PVF en $[1, 2]$. En un mismo gráfico incluya estos valores y la curva de la solución exacta.

2. Repita el ejercicio anterior en $[1, e]$, pero con condición de borde de tipo Neumann en el extremo derecho $y'(e) = 1/e$, manteniendo la condición $y(1) = 0$ en la frontera izquierda.

3. Use el método de diferencias finitas para aproximar y esbozar la gráfica de la solución para la ecuación de Airy $y'' + xy = 0$ en $[0, 1]$, sujeta a las condiciones de frontera $y'(0) = 0$, $y(1) = -1$.

4. (Ejercicio 14, Sección 9.5 de [4]) Considere el PVF dado por

$$y'' = y' - \operatorname{sen}(xy), \qquad y(0) = 1, \quad y(1) = 1.5.$$

   a) Aplique el método de tanteo para aproximar la curva solución para el PVF. Use RK4 con paso $h = 0.01$ y determine un valor de m tal que la condición inicial $y'(0) = m$ asegure que $|y_{N+1} - 1.5| < 10^{-2}$. *Sugerencia:* comience incluyendo en un mismo gráfico las curvas para `m=-2:0.5:2`. Observando estas curvas, repita el procedimiento con valores de m más adecuados.

   b) ¿El PVF tiene el aspecto de los presentados previamente en los métodos de diferencias finitas? Justifique.

5. Considere el PVF

$$y'' - 5y = -100x(1 - x), \qquad y(0) = 0, \quad y(1) = 0.$$

   a) Utilice el método de tanteo para aproximar la solución. Use RK4 con paso $h = 0.01$ y determine un valor de m tal que la condición inicial $y'(0) = m$ asegure $|y_{N+1}| < 10^{-2}$.

   b) Resuelva el PVF con el método de diferencias finitas, con $N = 40$.

   c) Con lo obtenido en el inciso anterior, aproxime $y'(0)$ por su diferencia adelantada. Compare con el/los valores de m hallados en el primer inciso.

6. Considere una pequeña variante del PVF del problema anterior:

$$y'' - 5y = -100x(1 - x), \qquad y(0) = 0, \quad y'(1) = 0.$$

   Esto es, solamente se cambió la condición de frontera en el extremo derecho.

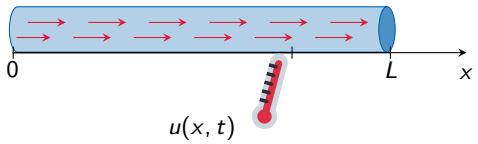
a) Supongamos que se conocen estimaciones para la derivada en $x = 0$ de la solución al PVF. Más precisamente, a partir de mediciones se sabe que $6.742 \leq y'(0) \leq 6.744$. Utilice el método de tanteo para aproximar la solución con RK4 con paso $h = 0.01$, y determine un valor de m tal que la condición inicial $y'(0) = m$ asegure `abs(y(2,N+1))<0.001`, siendo `y` la matriz de salida del método. Interprete qué significa esto en relación con el problema.

b) Resuelva el PVF con el método de diferencias finitas, con $N = 400$.

c) Con lo obtenido en el inciso anterior, aproxime $y'(0)$ por su diferencia adelantada, así como $y'(1)$ por su diferencia atrasada. Compare con valores hallados en el primer inciso.

3.3. La ecuación del calor estacionaria

Aplicaremos lo aprendido en la sección previa a un problema clásico: la ecuación del calor en una dimensión. Imaginemos una barra uniforme de longitud fija L, con sección transversal constante y material homogéneo. Supondremos que las propiedades del material (conductividad térmica, densidad, capacidad calorífica, etc.) son uniformes a lo largo de la barra y en los planos perpendiculares al eje x. En consecuencia, la temperatura es constante dentro de cada sección transversal de la varilla, por lo que las variaciones y el flujo de calor ocurren únicamente en la dirección x. Finalmente, asumimos que la superficie lateral de la barra está perfectamente aislada, de modo que no hay intercambio de calor con el ambiente a través de ella: el flujo transversal es nulo.



La función $u(x, t)$ representa la temperatura en un punto x a lo largo de la barra en un tiempo t. Nuestro objetivo es determinar cómo se distribuye la temperatura $u(x)$ a lo largo de la barra cuando se alcanza el **estado estacionario**, es decir, cuando la temperatura en cada punto de la barra ya no varía con el tiempo. En este caso, la función $u(x)$ satisface la ecuación de Poisson en una dimensión:

$$-ku''(x) = E(x), \qquad 0 < x < L.$$

Aquí, E representa una fuente interna de energía térmica (que puede ser generación o absorción de calor), mientras que la constante positiva k mide la capacidad del material para conducir el calor, y es conocida como la **conductividad térmica**[2]. Aunque k puede depender de x en casos generales (ver Ejercicio 4), aquí consideramos que es constante porque supusimos que la barra es homogénea, por lo que k

[2]En la ecuación del calor no estacionaria $\frac{\partial u}{\partial t} = K\frac{\partial^2 u}{\partial x^2} + \frac{E}{c\rho}$, la constante $K = \frac{k}{c\rho}$ es la **difusividad térmica**, siendo c el calor específico, ρ la densidad y k la conductividad térmica de la varilla.

depende solo del material. Por otro lado, en el Ejercicio 2 se aborda un caso en el que la fuente interna E depende directamente de la temperatura.

Esta ecuación puede estar acompañada de diferentes condiciones de frontera, que determinan cómo interactúa la barra con su entorno. A continuación, presentamos las más comunes, enfocándonos en el extremo izquierdo de la barra:

Temperatura prescripta. En este caso, el extremo de la barra se mantiene a una temperatura constante. Es un condición de tipo Dirichlet, que matemáticamente se expresa como $u(0) = u_0$.

Frontera aislada. Si no hay flujo de calor en el extremo de la barra, esto implica que el gradiente de temperatura en la frontera es cero. Corresponde a una condición de tipo Neumann, que se escribe como:

$$u'(0) = 0.$$

Ley de enfriamiento de Newton. Cuando un extremo de la barra unidimensional está en contacto con un fluido en movimiento (como aire, agua o aceite), las condiciones vistas anteriormente pueden no ser suficientes para describir el comportamiento del sistema. Por ejemplo, si la barra está caliente y en contacto con aire en movimiento, el calor fluirá desde la barra hacia el aire, calentándolo. Además, el movimiento del aire ayudará a dispersar este calor, enfriando la barra. La ley de enfriamiento de Newton establece que el flujo de calor hacia afuera de la barra es proporcional a la diferencia de temperaturas entre la barra y el medio circundante. Esto es

$$u'(0) = -H(u(0) - U_m),$$

donde $H > 0$ es una constante de proporcionalidad conocida como el coeficiente de transferencia de calor o coeficiente de convección, y U_m denota la temperatura del medio ambiente. Esta es una condición de tipo Robin.

De forma análoga, se pueden definir condiciones de frontera en $x = L$, esto es, para el extremo derecho de la barra. En cada extremo puede imponerse un tipo diferente de condición. Todos son casos particulares de lo trabajado en la sección previa.

Ejercicios Sección 3.3

Respuestas en página 109

1. Considere una barra de 1 metro de longitud con conductividad térmica constante igual a 0.5, sometida a una fuente dada por $E(x) = 20e^{-10(x-0.7)^2}$. Se prescribe una temperatura de 0 grados en cada borde.

   a) Plantee el PVF correspondiente a la temperatura u en su estado estacionario.

   b) Utilice el código de Octave para encontrar el valor de u en la mitad de la barra con al menos 3 decimales exactos. Indique el tamaño del paso h utilizado.

c) Observe el gráfico obtenido. ¿En qué punto espera que se alcance el valor máximo de u, teniendo en cuenta la forma de la fuente y las condiciones de borde? Compare con el resultado obtenido y analice si es consistente con lo esperado.

2. El coeficiente de reacción $r(x)$ en la ecuación del calor (o ecuaciones similares) representa la intensidad con la que el sistema reacciona térmicamente a la temperatura en cada punto. Este término aparece cuando hay procesos de disipación, absorción o generación de energía que son proporcionales a la temperatura $u(x)$. Por ejemplo, suponga que la ecuación es:

$$-ku''(x) + r(x)u(x) = E(x).$$

Repita lo hecho en el ejercicio anterior, considerando un coeficiente de reacción constante $r(x) = c$, para $c = 1$ y $c = 5$. Compare los tres resultados obtenidos e interprete el efecto de la reacción.

3. Considere el problema

$$\begin{cases} -ku'' = E(x) \\ u(0) = 0, \\ u'(L) = -(u(L) - 1), \end{cases}$$

con E, k y L como en los ejercicios anteriores.

a) ¿Qué tipo de condición hay en cada frontera? Resuelva en $[0, L]$ para obtener la gráfica de u y aproxime el valor de la temperatura en la mitad de la barra.

b) Utilice diferencias atrasadas para corroborar que los valores obtenidos en Octave satisfacen, aproximadamente, la condición para $x = L$.

c) En cada punto x de la barra, el flujo de calor de izquierda a derecha está dado por $q(x) = -ku'(x)$. Aproxime el valor de este flujo en todos los puntos de la barra y determine en qué posiciones alcanza sus valores máximo y mínimo.

4. Considere una barra unidimensional de longitud $L = 10$ cm, sin flujo de calor en la dirección perpendicular al eje de la barra y sin fuentes adicionales de energía térmica. La barra tiene una conductividad térmica que varía con la posición según $k(x) = 1 + 0.1x$. Los extremos de la barra están sometidos a las siguientes condiciones de temperatura:

$$u(0) = 0\,^{\circ}\mathrm{C}, \qquad u(L) = 100\,^{\circ}\mathrm{C}.$$

La ecuación del calor en estado estacionario para este sistema es:

$$\frac{d}{dx}\left(k(x)\frac{du}{dx}\right) = 0.$$

Halle una aproximación gráfica de la temperatura en cada punto de la barra.

Referencias bibliográficas

[1] Edwards, Henry and Penney, David (2007). *Differential Equations and Boundary Value Problems: Computing and Modeling* (4th Edition). Pearson Educación.

[2] Edwards, Henry and Penney, David (2009). *Ecuaciones diferenciales y problemas con valores de la frontera*. Pearson Educación.

[3] Facultad de Ingeniería Química de la UNL (2023). *Apunte de la cátedra Matemática D*.

[4] Zill Dennis (2018). *Ecuaciones diferenciales con problemas de valores en la frontera* (novena edición). Cengage.

Solucionario

Sección 1.1

Ver enunciados en página 7

1. *a*) $y(0.2) = 0.01$ *b*) $y(0.5) = 0.5639$ *c*) $y(0.5) = 1.6902$

2. *a*) $y(0.2) = 0.015$ *b*) $y(0.5) = 0.5565$ *c*) $y(0.5) = 1.7219$

3. La solución al PVI es $y = -x + 2e^x - 1$. Así, $y(1) \approx 3.4366$. Con $h = 0.1$ se obtiene $y_{10} = 3.1875$ por lo que $e(0.1) = 0.2491$. Con $h = 0.05$ tenemos $y_{20} = 3.3066$ y $e(0.05) = 0.13$. Notar que $\frac{1}{2}e(0.1) = 0.12455$ que es cercano $e(0.05)$. Así, salvo por errores de redondeo, era lo esperado.

4. *a*) El comando `linspace` en Octave se utiliza para crear un vector con valores distribuidos de manera uniforme entre dos extremos especificados. La sintaxis general es `linspace(valor_inicial, valor_final, num_puntos)`, donde

   - `valor_inicial`: denota el primer valor en el vector.
   - `valor_final`: último valor en el vector.
   - `num_puntos`: número total de puntos deseados en el vector, incluyendo los valores inicial y final. Este argumento es opcional; si no se especifica, por defecto generará 100 puntos.

   b) La línea 11 del Código 1.1 puede reemplazarse por

   ```
   x=linspace(0,1,N+1);
   ```

   y con esto la línea 21 ya no es necesaria.

   c) Un posible código para la función es el siguiente, y puede aplicarse escribiendo:

   ```
   [x,y]=Feuler(@(x,y) x+y,[0,1],1,10)
   ```

```
1 function [x,y]=Feuler(f,inter,y0,N)
2     % function [x,y]=Feuler(f,[x0,xF],y0,N)
3     % Metodo de Euler para resolver
4     %    y'=f(x,y) en [x0,xF]
5     %    y(x0)=y0
6     % Usando N pasos o subintervalos
```

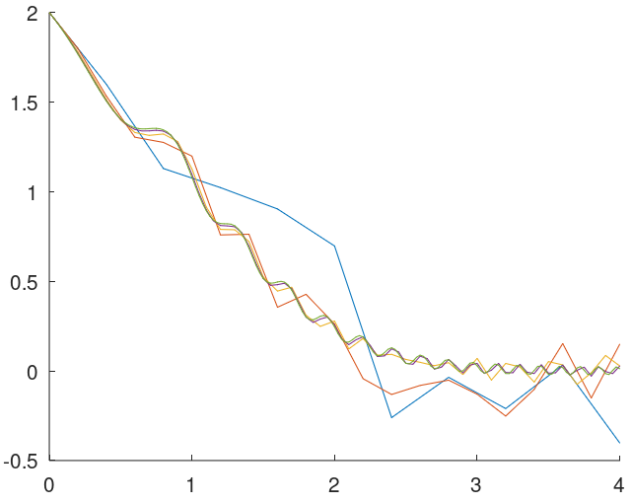
```
7
8  x=linspace(inter(1),inter(2),N+1); %División del
       intervalo
9  h=(inter(2)-inter(1))/N;           %Tamaño de paso
10
11 %Reserva de memoria para y
12 y=zeros(1,N+1);    %Dimensión de y0 por N+1 columnas
13 y(1)= y0;          %Asignación de condición inicial
14
15 for n=1:N
16   y(n+1)=y(n)+ h*f(x(n),y(n));
17 end
18
19 end
```

Código 3.2: Método de Euler como función.

5. Los números que se aproximan en cada inciso son π, e y $\ln(2)$, respectivamente. Para verlo, basta con resolver analíticamente las ecuaciones diferenciales. El número que se aproxima en cada caso es $y(x_F)$. Ordenamos en el siguiente cuadro los valores obtenidos para $y(x_F)$ mediante el método de Euler:

PVI	$N = 50$	$N = 100$	$N = 200$	$N = 400$
a)	3.1615	3.1516	3.1466	3.1441
b)	2.6916	2.7048	2.7115	2.7149
c)	0.6982	0.6957	0.6944	0.6938

6. *a*) El siguiente gráfico contiene las curvas solución para $N = 10$, 20, 40, 80 y $N = 160$.



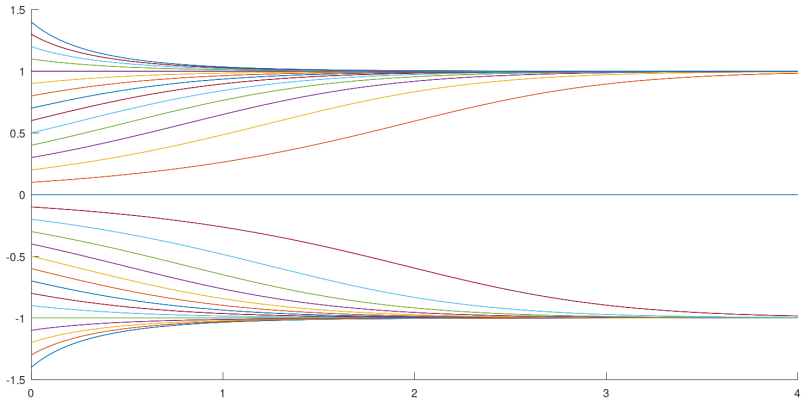
b) Con $N = 400$, el máximo de la solución es $M = 2$ y se alcanza cuando $x = 0$. Para deducir esto, escribimos `[M,p]=max(y)`, lo que arroja como respuesta `M=2` y `p=1`. Esto indica que el máximo $M = 2$ se alcanza en `x(1)=0`. El mínimo es $m = -0.020274$ y se alcanza cuando $x = 3.78$. Ambos comandos devuelven solo la *primera posición* donde se alcanzan los extremos. Si quisiéramos obtener *todas las posiciones* donde se alcanzan, podemos combinar con el comando `find`. Por ejemplo, para el caso del mínimo, podemos escribir primero `m=min(y)` y luego `find(y==m)`. Vemos que la única respuesta es `ans=379`, por lo que el mínimo ocurre solamente en `x(379)=3.7800`.

c) En relación con el comando `find`, es importante notar que la instrucción `find(y==1)` difícilmente arroje un resultado favorable cuando se calculan soluciones numéricas, debido a los redondeos y aproximaciones. Por ello, si se quiere resolver una igualdad de este tipo, resulta conveniente analizar desigualdades y obtener conclusiones a partir de ellas. Por ejemplo, vemos que $y(i) > 1$ en toda posición $i \leq 104$, y que $y(i) < 1$ para todo $i \geq 105$. Además, `x(104)=1.03` y `x(105)=1.04`, por lo que $y(x) = 1$ en $x = 1.035$, aproximadamente.

7. *a*) Suponiendo que se cuenta con la función `Feuler` (ver Código 3.2) definida en el espacio de trabajo, podemos escribir:

```
f=@(x,y) y*(1-y)*(1+y);

hold on

for y0=-1.4:0.1:1.4
[x,y]=Feuler(f,[0 4],y0,10000);
plot(x,y)
end

hold off
```

De esto se obtiene el siguiente gráfico:



Podemos observar que cuando $0 < y_0$, las curvas tienden a la solución de equilibrio $y = 1$, mientras que convergen a $y = -1$ siempre que $y_0 < 0$.

b) Notar que la ecuación diferencial dada es autónoma de primer orden, con puntos críticos $y = -1$, $y = 0$ e $y = 1$, que son sus soluciones constantes. Realizando un diagrama fase puede observarse que $y = 0$ es **repulsor**, mientras que los otros dos puntos críticos son **atractores**, lo que coincide con lo observado en el gráfico.

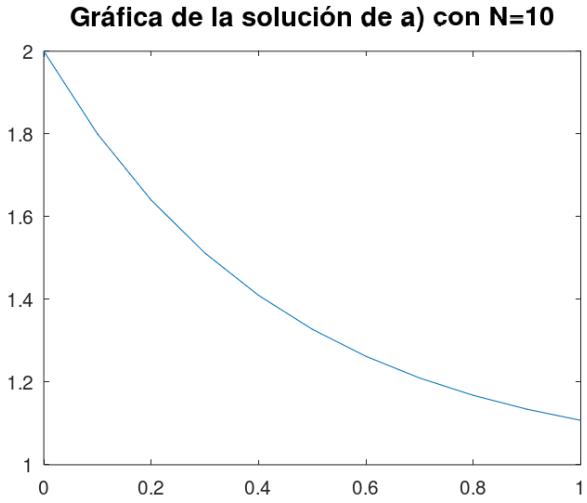
9. (i) La solución del PVI del inciso *a*) es $y_a = e^{-2x} + 1$, por lo que $y_a(1) \approx 1.1353$. Por otro lado, para *b*) tenemos la solución $y_b = e^{-20x} + 1$, por lo que $y_b(1) \approx 1$.

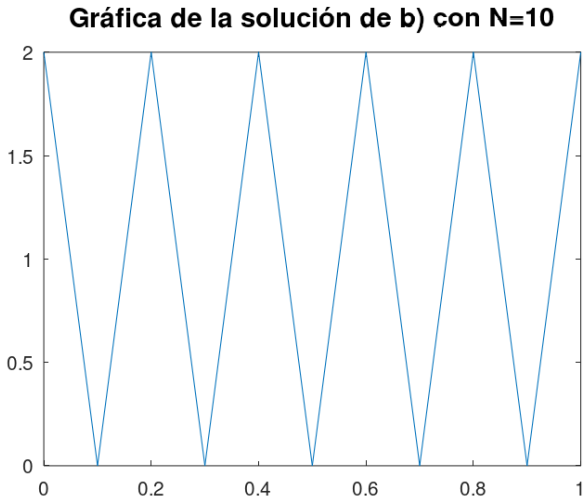
(ii) Aplicando el método de Euler con $N = 10$, obtenemos que $y_a(1) \approx 1.1074$, que es una aproximación exacta hasta la primera cifra decimal. Sin embargo, para el otro PVI la aproximación no es buena, obteniendo $y_b(1) \approx 2$.

Observemos los resultados, además, de forma gráfica. Por ejemplo, suponiendo que se cuenta con la función `Feuler` (ver Código 3.2) definida en el espacio de trabajo, para el segundo PVI escribimos:

```
[x,y]=Feuler(@(x,y) -20*y+20,[0,1],2,10)
x =

    0    0.1    0.2    0.3    0.4    0.5    0.6    0.7    0.8    0.9
      1.0

y =

    2    0    2    0    2    0    2    0    2    0    2

plot(x,y)
title('Gráfica de la solución de b) con N=10',
        'Fontsize',14)
```

De forma análoga lo hacemos para el primer PVI. Los gráficos son los siguientes.





(iii) Para b), repetimos con $N = 20$ y $N = 40$, y en ambos casos obtenemos una buena aproximación para $y(1)$. Sin embargo, veremos a continuación que la aproximación no es buena en el resto de los valores.

(iv) Luego de resolver para $N = 40$, comparamos:

```
x(5)
ans = 0.1000
y(5)
ans = 1.0625
e^(-20*0.1)+1
ans = 1.1353
```

Esto nos permite ver que la aproximación para $x = 0.1$ obtenida está aún algo lejos del valor real de la solución.

(v) Vamos duplicando el valor de N, llegando a la precisión deseada con $h = 1/640 = 0.0015625$.

```
[x,y]=Feuler(@(x,y) -20*y+20,[0,1],2,80);
x(9)
ans = 0.1000
y(9)
ans = 1.1001
[x,y]=Feuler(@(x,y) -20*y+20,[0,1],2,160);
x(17)
ans = 0.1000
y(17)
ans = 1.1181
[x,y]=Feuler(@(x,y) -20*y+20,[0,1],2,320);
x(33)
ans = 0.1000
y(33)
ans = 1.1268
[x,y]=Feuler(@(x,y) -20*y+20,[0,1],2,640);
x(65)
```

```
ans = 0.1000
y(65)
ans = 1.1311
h=1/640
h = 1.5625e-03
```

Sección 1.2

Ver enunciados en página 12

1. $y(1) = 1.4578$.
2. $e(\frac{h}{2}) \simeq \frac{1}{4} e(h)$. Así, si $e(h) = 0.01$, entonces $e(\frac{h}{2})$ será aproximadamente 0.0025.
4. *a*) $e(0.1) = 0.0084$.

   b) $y_{20} = 3.4344$ y $e(0.05) = 0.0022$. Notar que $\frac{1}{4} e(0.1) = 0.0021$ que es cercano a $e(0.05)$. Así, salvo por errores de redondeo, coincide con lo esperado.
5. Como vimos en el Ejercicio 5 de la Sección 1.1, los números que se aproximan en cada inciso son los irracionales π, e y $\ln(2)$, respectivamente. Ordenamos en el siguiente cuadro los valores obtenidos para $y(x_F)$ mediante el método de Euler mejorado:

PVI	$N = 10$	$N = 20$	$N = 40$	$N = 80$
a)	3.1399	3.1412	3.1415	3.1416
b)	2.7141	2.7172	2.7180	2.7182
c)	0.6938	0.6933	0.6932	0.6932

Observar que con $N = 80$ los irracionales se aproximan con 3 cifras decimales exactas (4 cifras para el caso de e) mediante Euler mejorado, mientras que con el método de Euler clásico alcanzamos una precisión de solo 2 cifras decimales exactas con $N = 400$.

Sección 1.3

Ver enunciados en página 14

2. En el siguiente cuadro reunimos la información sobre los valores obtenidos para $y(x_F)$ mediante el método RK4, redondeando los resultados a 4 cifras decimales:

PVI	$N = 10$	$N = 20$	$N = 40$	$N = 80$
a)	3.1416	3.1416	3.1416	3.1416
b)	2.7183	2.7183	2.7183	2.7183
c)	0.6931	0.6931	0.6931	0.6931

Como podemos observar, la precisión de 3 cifras decimales se obtiene ya con $N = 10$ en los tres casos.

3. *a*) La solución exacta al PVI es $y = \operatorname{sen}(x)$, por lo que $y(2) \approx 0.909297426825682$.

   b) Completamos la tabla especificando `format long`, e indicando con color rojo las cifras correctas:

h	N	Euler	RK2
$1/10$	20	0.946489878324600	0.907227422917302
$1/20$	40	0.927722383988581	0.908792022422675
$1/40$	80	0.918468234483594	0.909172544700043
$1/80$	160	0.913872551462343	0.909266387252688
$1/160$	320	0.911582436380203	0.909289689386820
$1/320$	640	0.910439295523970	0.909295495262454

h	N	RK4
$1/10$	20	0.909296701522917
$1/20$	40	0.909297382606387
$1/40$	80	0.909297424096423
$1/80$	160	0.909297426656175
$1/160$	320	0.909297426815121
$1/320$	640	0.909297426825023

   c) Tomando $N = 5$ en RK4 se obtiene $y(2) \approx 0.909083643448336$, lo que tiene una precisión de 3 cifras decimales. Además, aplicando RK2 con $N = 1280$ y redondeando a 6 cifras decimales, se tiene $y(2) \approx 0.909297$. Incluimos esto en la tabla que aparece a continuación. En ella, los valores coloreados fueron estimados observando los errores iniciales y los sucesivos. A partir del valor exacto podemos ver que, tal como esperamos, el error en el método de Euler se reduce a la mitad al duplicar N. Teniendo en cuenta que $e(1/10) \approx 0.0372$, nos preguntamos para qué n se cumple que $0.0372/2^n < 10^{-3}$ (menor estricto para que 3 cifras decimales sean exactas). De aquí se deduce $n = 6$, por lo que $N = 20 \cdot 2^6$. Efectivamente, aplicando el método de Euler con $N = 1280$ obtenemos $y(2) \approx 0.909868202418011$, que tiene 3 cifras exactas. De igual forma se estiman los otros dos valores para Euler.

   Para RK2 se procede igual, observando que el error se divide por 4 al duplicar el valor de N. Así, para lograr 10 cifras exactas, partimos de que $e(1/10) \approx 0.00207$, y buscamos n de modo que $0.00207/4^n < 10^{-10}$. Esto arroja $n = 13$ y $N = 20 \cdot 2^{13} = 163\,840$. Para las 6 cifras exactas, este procedimiento arroja $n = 6$, lo que indica $N = 20 \cdot 2^6 = 1280$.

Método	Cifras decimales exactas	N	Evaluaciones de f
Euler	3	1280	1280
	6	1 310 720	1 310 720
	10	$20 \cdot 2^{29}$	$20 \cdot 2^{29}$
RK2	3	80	160
	6	1280	2560
	10	163 840	327 680
RK4	3	5	20
	6	40	160
	10	320	1280

Sección 2.1

Ver enunciados en página 26

1. De la solución exacta del sistema

$$x(t) = \tfrac{2}{3}e^{5t} + \tfrac{1}{3}e^{-t}; \qquad y(t) = \tfrac{4}{3}e^{5t} - \tfrac{1}{3}e^{-t},$$

se tiene que $x(0.2) \approx 2.0851$, $y(0.2) \approx 3.3515$. Veamos las aproximaciones obtenidas.

a) $x(0.2) \approx 1.77$; $y(0.2) \approx 2.73$;
b) $x(0.2) \approx 1.94$; $y(0.2) \approx 3.06$;
c) $x(0.2) \approx 2.0785$; $y(0.2) \approx 3.3382$;
d) $x(0.2) \approx 2.0851$; $y(0.2) \approx 3.3515$.

Observemos que un solo paso del método de Euler mejorado produce mejor precisión que dos pasos de Euler ordinario. RK4 con un paso disminuye aún más el error en la aproximación. El método ode45 tiene una precisión excelente. Sin embargo, observe lo siguiente:

```
[x,y]=ode45(f,[0 0.2],[1;1]);
y(end,:)
ans =

  2.0851    3.3515

size(y)
ans =

  41     2
```

Esto significa que realizó 40 pasos para este sistema.

2. Aquí la solución exacta del sistema es

$$x(t) = -6\,\mathrm{sen}(t) - 4\cos(t) + t + 1; \qquad y(t) = -4\,\mathrm{sen}(t) + 6\cos(t) + t - 1,$$

por lo que $x(0.2) \approx -3.9123$, $y(0.2) \approx 4.2857$. Veamos las aproximaciones obtenidas.

a) $x(0.2) \approx -3.96$; $y(0.2) \approx 4.34$;

b) $x(0.2) \approx -3.92$; $y(0.2) \approx 4.28$;

c) $x(0.2) \approx -3.9123$; $y(0.2) \approx 4.2857$;

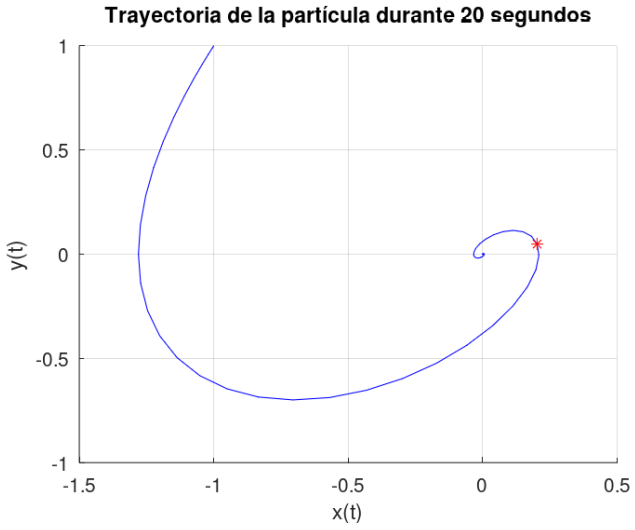
d) $x(0.2) \approx -3.9123$; $y(0.2) \approx 4.2857$.

Notar que un paso de RK4 arroja las mismas aproximaciones que ode45, que realizó 40 pasos nuevamente.

3. El método de Euler mejorado con $N = 10$ arroja $y_1(1) \approx 2.9890$, $y_2(1) \approx 3.4013$. Con el método de Euler, para el mismo paso, se obtuvo $y_1(1) = 2.8085$ e $y_2(1) = 3.1306$ en el Ejemplo 2, y se enunciaron los valores obtenidos de la solución exacta, redondeados a 4 cifras decimales: $y_1(1) = 2.9890$ e $y_2(1) = 3.3950$. Como se puede observar, la aproximación mejora con RK2.

4. El siguiente código aproxima la solución y grafica la trayectoria, tal como se pide en *c*), y también indica con * la posición de la partícula a los 3 segundos.

```
f=@(t,y) [-t*y(2); t*(y(1)-y(2))];
y0=[-1;1];
[t,y]=rk4(f,[0,20],y0,200); %N=200 implica h=0.1
hold on
plot(y(1,:), y(2,:), '-b'); % x vs. y
xlabel('x(t)');
ylabel('y(t)');
title('Trayectoria de la partícula durante 20 segundos');
grid on;

%Posición cuanto t=3
plot(y(1,31), y(2,31), '*r')
hold off
```



a) Una forma sería volver a correr el código en el intervalo $[0, 3]$ y tomar los valores arrojados por `y(:,end)`, como haremos en el inciso siguiente. Sin embargo, se pide usar los resultados obtenidos con el código en $[0, 20]$. El tiempo $t = 3$ corresponde a `t(31)`, por lo que la posición a los 3 segundos es:

```
y(:,31)
ans =

    0.201915
    0.048473
```

Para calcular la velocidad, debido al significado de la función f en el problema, simplemente escribimos

```
f(3,y(:,31))
ans =

  -0.1454
   0.4603
```

Podemos concluir que la posición y la velocidad a los 3 segundos son, respectivamente,

$$\vec{r}(3) \approx (0.201915, 0.048473), \qquad \vec{v}(3) \approx (-0.1454, 0.4603).$$

b) Para analizar los decimales correctos, siguiendo lo sugerido en el recuadro "Elección del tamaño del paso" dado en la página 7, escribimos:

```
f=@(t,y) [-t*y(2); t*(y(1)-y(2))];
y0=[-1;1];

N=30; % comienza con h=0.1 porque resuelvo en [0,3]

for i=1:1:6
[t,y]=rk4(f,[0,3],y0,N);
y(:,end)'
N=2*N;
end
```

La respuesta es la siguiente e indica que $\vec{r}(3)$ tiene al menos 3 decimales correctos, considerando ambas componentes.

```
ans =

  0.201915   0.048473

ans =

  0.201896   0.048453

ans =

  0.201895   0.048452
```

```
ans =

  0.201895   0.048452

ans =

  0.201895   0.048452

ans =

  0.201895   0.048452
```

d) Vamos a aproximar la integral por la suma de Riemann evaluada en los extremos izquierdos y derechos, respectivamente. Más precisamente, definimos $h = (b-a)/N$ y dividimos el intervalo $[a, b]$ en N intervalos de igual longitud.

Sabemos que

$$\int_a^b F(t)\,dt \approx L_N = \sum_{i=0}^{N-1} F(t_i)h \qquad \text{y} \qquad \int_a^b F(t)\,dt \approx R_N = \sum_{i=1}^{N} F(t_i)h,$$

y, si F es integrable, ambos valores tienden a un mismo número cuando hacemos $N \to \infty$. Aquí $F(t) = \sqrt{(x'(t))^2 + (y'(t))^2}$. Utilizaremos las función f, junto con los valores obtenidos para x e y en los tiempos t_i, para calcular las derivadas x' e y' en dichos instantes. Con el siguiente código se obtiene $L = 3.3518$ $R = 3.3567$, de lo que deducimos la longitud de la trayectoria durante los primeros 3 segundos es, aproximadamente, $\ell(3) = 3.35$ unidades.

```
f=@(t,y) [-t*y(2); t*(y(1)-y(2))];
y0=[-1;1];
N=300;
inter=[0 3];
h=(inter(2)-inter(1))/N;

[t,y]=rk4(f,inter,y0,N);  %RK4 para la solución

%inicializamos  derivada
D=zeros(2,N+1);

% Aproximar las derivadas usando lo obtenido
for i=1:N+1
  D(:,i)=f(t(i),y(:,i));
end

%derivadas de x e y
dx=D(1,:);
dy=D(2,:);

%Discretización de F
F=sqrt(dx.^2+dy.^2);

%Suma de Riemann por izquierda
L=sum(F(1:N))*h

%Suma de Riemann por derecha
R=sum(F(2:N+1))*h
```

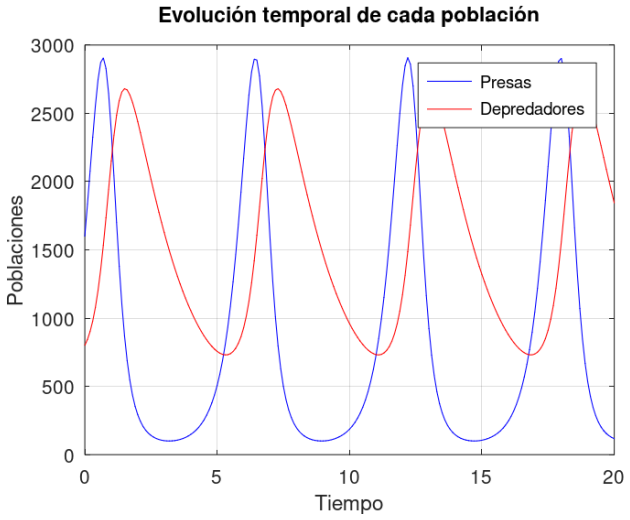
Con `N=2000` e `inter=[0 20]` tenemos $\ell(20) \approx 3.74$. Para los próximos 17 segundos podemos hacer $\ell(20) - \ell(3)$. Así, en los últimos 17 segundos la partícula recorre 0.4 unidades de longitud.

e) Ejecutando el siguiente código se obtiene $t_0 = 4.54$.

```
f=@(t,y) [-t*y(2); t*(y(1)-y(2))];
y0=[-1;1];
N=4000;
inter=[0 20];
h=(inter(2)-inter(1))/N;

[t,y]=rk4(f,inter,y0,N);

%Distancias al origen
d=sqrt(y(1,:).^2+y(2,:).^2);

cumple = find(d < 0.01);  % Índices donde la condición
                          % se cumple
p0=cumple(1); %primer índice
t0=t(p0)      %tiempo correspondiente
```

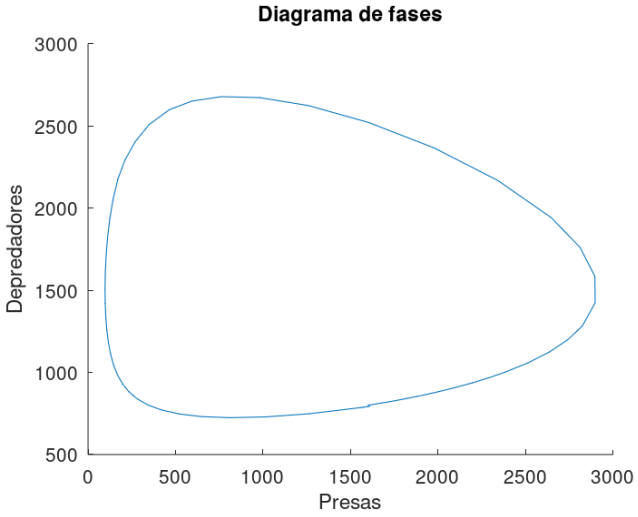
5. *a*) La especie que no puede sobrevivir sin la otra es la de depredadores, que necesitan a las presas para alimentarse. Supongamos que $x = 0$. Entonces la segunda ecuación queda $y' = -0.5y$, esto es, la población de y decrece hasta extinguirse. En cambio, si $y = 0$, la primera ecuación queda $x' = 3x$, por lo que la población x crecería indefinidamente. Esto nos permite concluir que x representa las presas, mientras que y denota a la cantidad de depredadores.

   b) Representemos primero cómo varía cada especie en el tiempo. Podemos observar que las poblaciones de presas y depredadores oscilan periódicamente entre un máximo y un mínimo, debido a la dinámica de interacción, y que este ciclo se repite periódicamente.



Otra opción es graficar el espacio de fases, donde se representan x (presas) y y (depredadores), y se observa la típica trayectoria cerrada, lo que indica

que las poblaciones de presas y depredadores oscilan periódicamente. La curva graficada describe la evolución del sistema en el tiempo. Cada punto de la curva representa un estado del sistema (presas y depredadores). El paso del tiempo se refleja en la dirección en la que se recorre la trayectoria en el espacio de fases (en este ejemplo, en sentido antihorario, lo que puede detectarse modificando el intervalo de tiempo en Octave), lo que indica cómo evolucionan las poblaciones a lo largo del tiempo.



Algo interesante de señalar es que el modelo tiene un **punto de equilibrio** donde las poblaciones de presas y depredadores permanecen constantes. Matemáticamente, el punto es donde ambas derivadas se anulan, lo que denominamos **punto crítico** del sistema. Este punto aparece en el espacio de fases como una especie de centro del ciclo cerrado.

En este ejemplo el punto de equilibrio es:

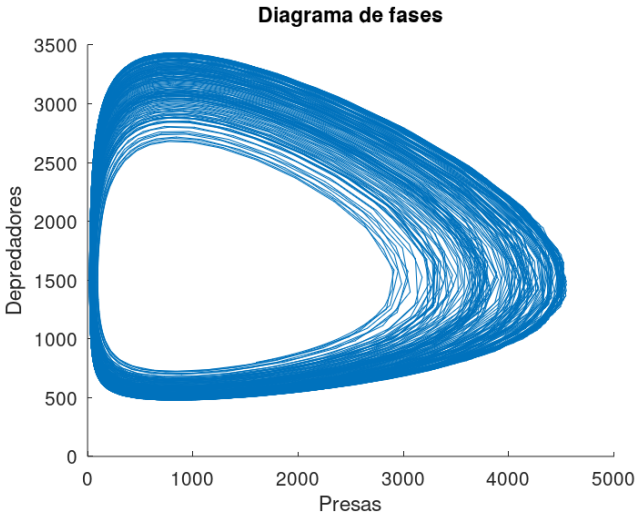
$$x^* = \frac{0.5}{0.0006} \approx 833, \qquad y^* = \frac{3}{0.002} = 1500.$$

En cualquiera de los gráficos puede observarse la dinámica de un modelo de este tipo, en el que el ciclo cerrado tiene las siguientes etapas.

- **Inicio:** si las presas son altas y los depredadores son bajos, estos últimos comienzan a crecer porque hay mucha comida disponible.
- **Aumento de depredadores:** a medida que los depredadores crecen, cazan más presas, lo que produce una reducción en esta población. Esto conduce a que las presas alcancen un mínimo de individuos.
- **Disminución de depredadores:** con menos presas, los depredadores comienzan a disminuir por falta de alimento. Esto permite que las presas se recuperen porque la depredación es baja.

Recuperación de las presas: una vez que las presas comienzan a recuperarse, el ciclo se repite. El sistema vuelve a sus condiciones iniciales, aunque puede estar desplazado por pequeñas variaciones debidas a la aproximación del método y a errores de redondeo.

Este desplazamiento debido a variaciones puede observarse en el siguiente gráfico, generado con Octave tomando valores grandes para el tiempo (en el anterior, limitamos el tiempo al intervalo $0 \leq t \leq 5.77$, que es el tiempo del ciclo que se estima observando la primera gráfica).



c) Como mencionamos, el gráfico que muestra la evolución temporal de cada una de las poblaciones permite establecer que el ciclo dura un poco menos de 6 meses. Esto se determina mirando el período de dichas gráficas, aunque también puede determinarse analíticamente, escribiendo:

```
for i=1:N
inicio(i) = (y(1,i) <= 1600 && y(1,i+1) >= 1600);
end

pos=find(inicio==1);
ciclo=(t(pos(2))+t(pos(2)+1))/2
```

Lo anterior primero construye un vector lógico que compara elementos consecutivos del vector de presas, para ver cuándo pasa de ser menor o igual que $x_0 = 1600$, a mayor o igual. En el vector `pos` se guardan las posiciones que satisfacen esto, y de todas estas nos quedamos con la segunda (la primera es cuando $t = 0$). El ciclo se obtiene promediando el tiempo en que esto pasa con el siguiente. El resultado es `ciclo = 5.7500`, como esperábamos.

d) Debido al aspecto de las gráficas en este problema en particular, podemos hallar el menor tiempo tal que la población de presas sea menor que la de depredadores. Luego, buscamos el punto medio entre este tiempo con el anterior, para acercarnos al momento en que son iguales. En lo siguiente

hemos considerado el intervalo de tiempo $[0, 20]$ y RK4 con $N = 200$ (notar que $h = 0.1$ significa un paso de 3 días, suponiendo meses de 30 días).

```
p=find(y(1,:)<y(2,:));      % posiciones donde
                            % presas<predadores
p(1)                        % primera posición que pasa
ans = 12
t(12)                       % tiempo correspondiente
ans = 1.1000
t(11)
ans = 1
(t(12)+t(11))/2             % promedio
ans = 1.0500
```

Así, ambas poblaciones coinciden por primera vez, aproximadamente, luego de un mes del inicio del ciclo. Para aproximar la cantidad de individuos de cada especie podemos mostrar ambas poblaciones en ambos tiempos, y obtener una estimación:

```
y(:,11:12)      % ambas filas, columnas 11 y 12
ans =

  2368.8   2040.4
  2151.2   2336.1
```

Lo anterior establece que al tiempo $t(11) = 1$ hay aproximadamente 2369 presas y 2151 depredadores, y luego de 3 días hay 2040 presas y 2336 depredadores. Podemos estimar la cantidad de individuos en ambas poblaciones como 2200 (que coincide con lo que se observa en el gráfico). También puede observarse que las presas están decreciendo, mientras los depredadores crecen. La tasas puedes aproximarse reemplazando en el sistema:

$$x' = 2200(3 - 0.002 \cdot 2200) = -3080,$$
$$y' = -2200(0.5 - 0.0006 \cdot 2200) = 1804.$$

Las presas están disminuyendo a una velocidad mucho mayor que la velocidad de aumento de los depredadores. Esto indica que los depredadores están en una etapa de crecimiento, porque todavía hay presas disponibles.

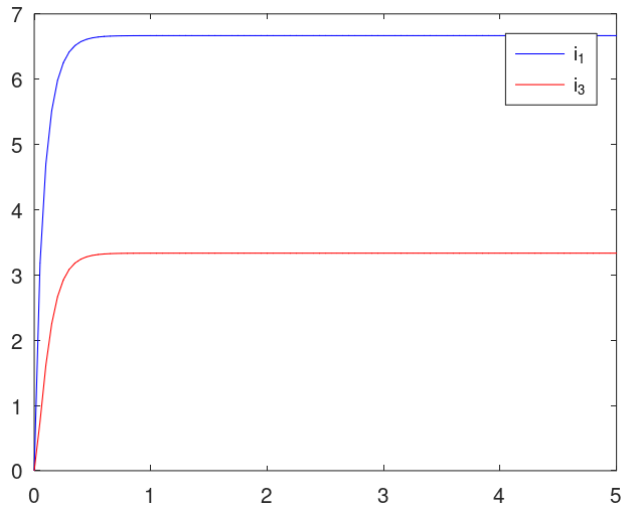
6. Si se resuelve el sistema analíticamente, se obtiene que la solución particular es

$$\mathbf{i}_p = \begin{pmatrix} 20/3 \\ 10/3 \end{pmatrix}.$$

Además, vería que los términos de la solución complementaria decaen exponencialmente, por lo que la solución tenderá a $\mathbf{i}_p$. Así, esperamos que ocurra que $i_1 \to 6.67$ y que $i_3 \to 3.33$, aproximadamente, cuando $t \to \infty$. Efectivamente, escribiendo:

```
f=@(t,y) [-20*y(1)+10*y(2)+100;10*y(1)-20*y(2)];
y0=[0;0];
N=100;
[t,y]=rk4(f,[0 5],y0,N);
plot(t,y(1,:),'-b',t,y(2,:),'-r');
legend('i_1', 'i_3', 'Interpreter', 'tex');
```

se obtiene el gráfico siguiente.



¿Qué ocurre si tomamos $N = 50$? Copie el código, ejecute y observe.

Técnicamente no es un sistema rígido, ya que ambos autovalores son grandes en módulo ($\lambda_1 = -10$, $\lambda_2 = -30$). Sin embargo, el método RK4 tiene una región de estabilidad limitada, determinada por la condición $|\lambda \Delta t| \leq 2.8$, donde λ es el autovalor de mayor módulo del sistema lineal y Δt es el tamaño del paso temporal. El valor 2.8 es propio del método. En este caso, $\Delta t = 5/N$, por lo que $N = 50$ produce $30 \cdot 5/50 = 3 > 2.8$.

Sección 2.2

Ver enunciados en página 36

1. Aplicando el método de Euler con paso $h = 0.1$ tenemos que $y(0.2) \approx -1.68$. La solución exacta es $y(x) = e^{2x}(5x - 2)$, por lo que $y(0.2) \approx -1.4918$.
2. Aplicando el método de Euler con paso $h = 0.1$ tenemos que $x(1.2) \approx 5.9$. La solución exacta es $x(t) = 5t^2 - t$, por lo que $x(1.2) = 6$.
3. Las ecuaciones del método de Euler mejorado aplicadas al sistema (2.16) son:

$$x^*_{n+1} = x_n + h \cdot y_n,$$
$$y^*_{n+1} = y_n + h \cdot g(t_n, x_n, y_n).$$

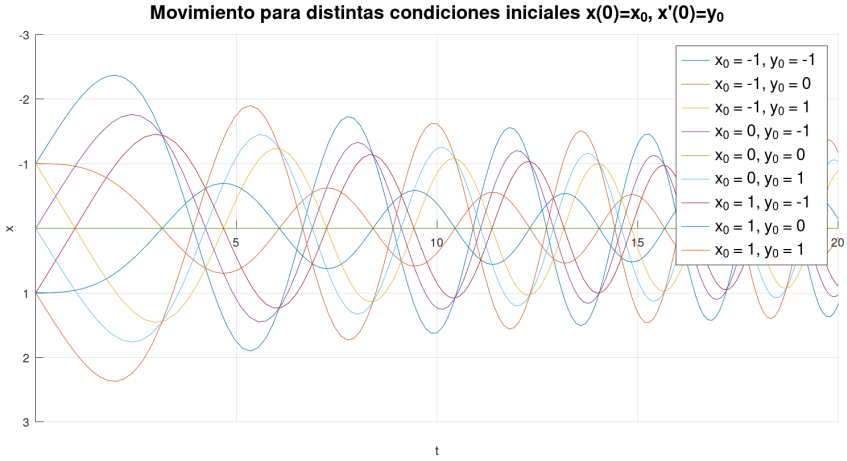
y

$$x_{n+1} = x_n + \tfrac{h}{2}\left[y_n + y^*_{n+1}\right],$$
$$y_{n+1} = y_n + \tfrac{h}{2}\left[g(t_n, x_n, y_n) + g(t_{n+1}, x^*_{n+1}, y^*_{n+1})\right].$$

Aplicando el método de Euler mejorado con paso $h = 0.1$ para el PVI del primer ejercicio se obtiene $y(0.2) \approx -1.5128$. Para el del segundo ejercicio, arroja $x(1.2) \approx 6.0005$.

4. RK4 con $h = 0.1$ arroja $y(0.2) \approx -1.4919$ como aproximación para el PVI del primer ejercicio.
5. Aplicando RK4 con $h = 0.1$ se obtiene $x(1.2) = 6$ como aproximación para el PVI del Ejercicio 2.
6. *a*) La ecuación puede reescribirse como $x'' + \frac{t}{4}x = 0$, que es del tipo de Airy. Físicamente, el término $\frac{t}{4}x$ puede interpretarse como una resistencia que se intensifica con el tiempo, lo que inhibe el movimiento. A medida que t aumenta, las oscilaciones de la masa en el resorte se amortiguan más rápidamente, hasta que el sistema prácticamente se detiene.

   b) Como puede verse en el siguiente gráfico, independientemente de las condiciones iniciales, el efecto estabilizador lleva a $x(t)$ a oscilar con amplitud decreciente y el sistema tenderá a estabilizarse en la posición de equilibrio.



El gráfico previo fue realizado con el siguiente código.

```
f=@(t,y) [y(2); -0.25*t*y(1)]; %función de este problema

hold on
etiquetas = {}; % Inicializa una celda para almacenar las
    etiquetas

for x0=-1:1:1
  for y0=-1:1:1
      [t,y]=ode45(f,[0 20],[x0;y0]);
      % Grafico de x(t):
      plot(t,y(:,1));
      % Genera la etiqueta dinámica para las condiciones
    iniciales
      etiquetas{end + 1} = sprintf('x_0 = %d, y_0 = %d', x0
    , y0);
  end
end

% Añade las leyendas al gráfico
legend(etiquetas, 'Interpreter', 'tex','Fontsize',14);
```

```
xlabel('t');
ylabel('x');
title('Movimiento para distintas condiciones iniciales x(0)
    =x_0, x''(0)=y_0','Fontsize',16);
grid on;
set(gca, 'YDir', 'reverse'); %sentido positivo hacia abajo
set(gca, 'XAxisLocation', 'origin', 'YAxisLocation', '
    origin'); %ejes centrado
hold off
```

7. *a*) Haciendo las sustituciones $y_1 = y$, $y_2 = y'$ e $y_3 = y''$, el PVI se transforma en el siguiente sistema

$$\begin{cases} y_1'(x) = y_2, \\ y_2'(x) = y_3, \\ y_3'(x) = -4\,\mathrm{sen}(x) - 2\cos(x) - 4y_3 - 5y_2 - 2y_1, \end{cases}$$

sujeto a $y_1(0) = 1$, $y_2(0) = 0$, $y_3(0) = -1$. Se puede ver que la solución al PVI dado es $y(x) = \cos(x)$, por lo que $y(2.5) = -0.801143615546934$ (redondeado por Octave con `format long`).

b) Con las instrucciones dadas a continuación, que siguen lo sugerido en la página 7, se obtienen las siguientes aproximaciones y el gráfico contenido en la Figura S.1.

```
f=@(x,y) [y(2); y(3);-4*y(3)-5*y(2)-2*y(1)-4*sin(x)-2*
    cos(x)];
y0=[1;0;-1];

N=10;
format long

for i=1:1:6
  [x,y]=rk4(f,[0 2.5],y0,N);
  y(1,N+1)
  N=2*N;
end

format short
```

```
ans =  -0.801138668416732
ans =  -0.801144286668995
ans =  -0.801143682782340
ans =  -0.801143620464108
ans =  -0.801143615875471
ans =  -0.801143615568113
```

Así, $y(2.5) \approx -0.801143$.

c) Los resultados para y' están en la segunda fila de la matriz de salida. Queremos ver cuántas veces se anulan los elementos de dicha fila. Al saber la solución exacta, sabemos que $y'(x) = -\,\mathrm{sen}(x)$, por lo que esperamos que

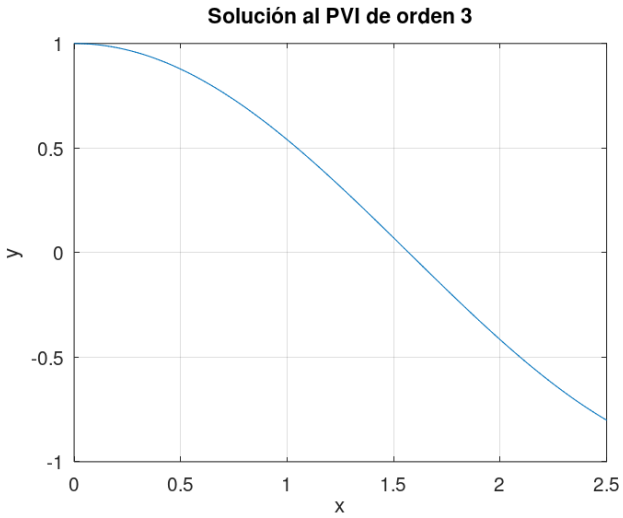
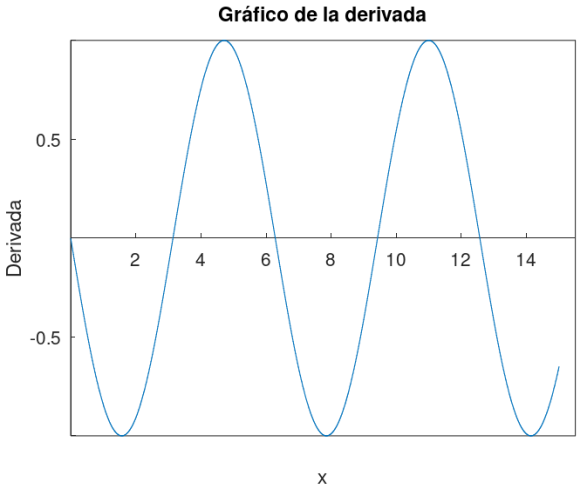


Figura S.1: Gráfico para el Ejercicio 7*b*.

se anule 5 veces en $[0, 15]$. Como son aproximaciones, es probable que ninguno de los valores en la fila sea exactamente cero. Es posible definir una tolerancia que nos indique cuándo consideramos cero un valor, como 10^{-8} por ejemplo. Sin embargo, es difícil ajustar la tolerancia y el paso para que no cuente mal, ya que en los métodos numéricos muchos valores consecutivos se parecen entre sí. Una opción factible es graficar la derivada en $[0, 15]$ y observar cuántas raíces tiene. Otra es hacer que Octave cuente las veces que hay un cambio de signo en el vector de las derivadas. El siguiente código hace ambas cosas, dando como resultado el gráfico y la salida que aparece debajo del mismo, coincidiendo con lo esperado.

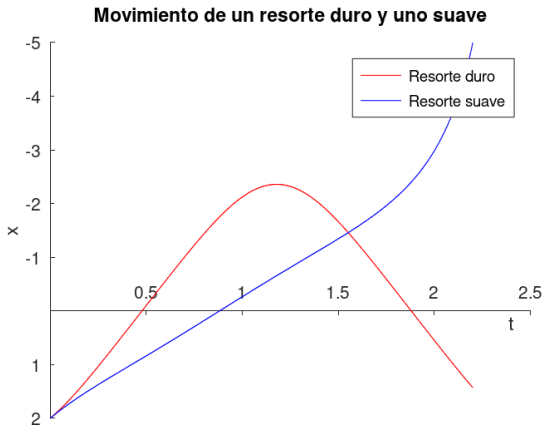
```
f=@(x,y) [y(2); y(3);-4*y(3)-5*y(2)-2*y(1)-4*sin(x)-2*
    cos(x)];
y0=[1;0;-1]; % CI

N=2000;       %cantidad de pasos

[x,y]=rk4(f,[0 15],y0,N);  %Método de RK

% Identificar valores cercanos a cero
for i=1:N
     %Comprobar si hay cambio de signo en y(2,:) entre
     %dos puntos consecutivos creando un vector lógico
     quasizeros(i) = (y(2,i) < 0 && y(2,i+1) > 0)  ||
    (y(2,i) > 0 && y(2,i+1) < 0);
end

% Contar cuántas veces se "anula"
num_ceros=sum(quasizeros)+sum(y(2)==0);
```

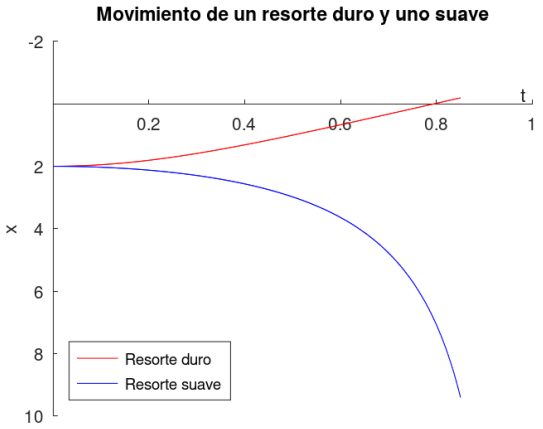
```
% Mostrar resultado
fprintf('La derivada se anula %d veces en el intervalo.\
    n', num_ceros);

% Graficar para corroborar
plot(x,y(2,:));
xlabel('x');
ylabel('Derivada');
title('Gráfico de la derivada');
% Ejes centrados
set(gca, 'XAxisLocation', 'origin', 'YAxisLocation', '
    origin');
xlim([0 15.5]); %limito el dominio
```



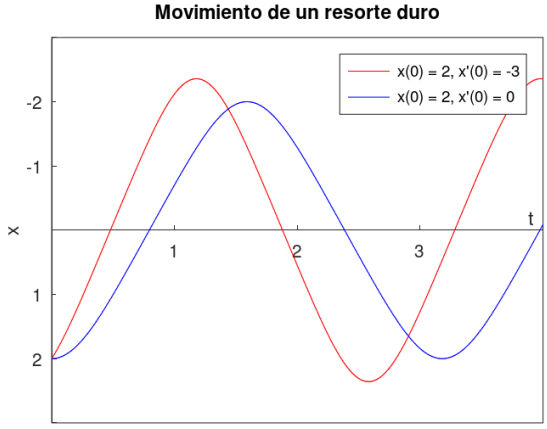

```
La derivada se anula 5 veces en el intervalo.
```

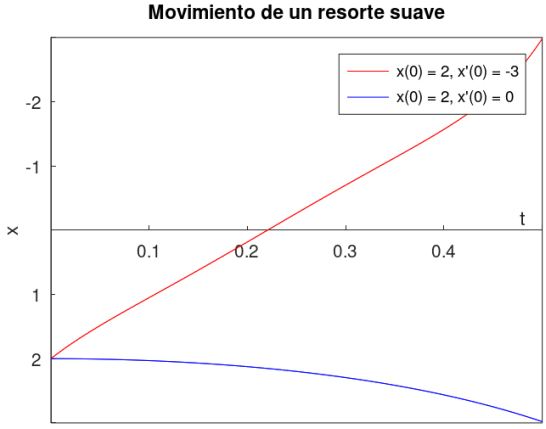
8. Los gráficos para los incisos *a*) y *b*), respectivamente, son:



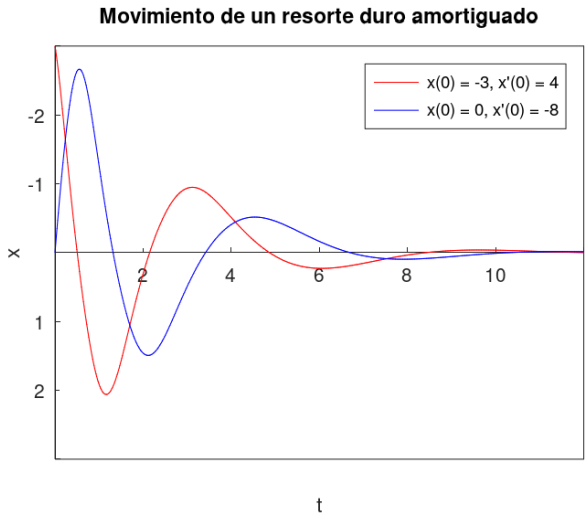


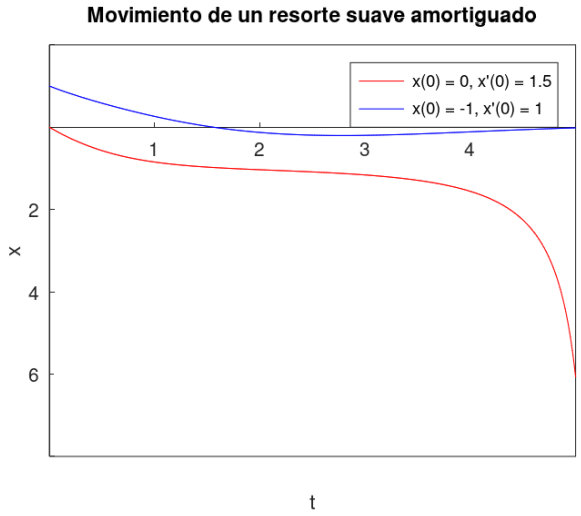
Para los incisos c) y d), obtenemos lo mismo que en la Figura 5.3.2 de [4], solo que allí se graficó el sentido positivo hacia arriba. Además, el gráfico de d) se realizó en [4] sobre un intervalo mayor para t.





9. El resorte duro parece tender a la posición de equilibrio para tiempo suficientemente grandes. Para el resorte suave, puede depender de las condiciones iniciales: si son suficientemente pequeñas, el movimiento puede oscilar y tender a cero.





10. Cuando k_1 es pequeño, el término no lineal k_1x^3 tiene una contribución relativamente débil al comportamiento del sistema. Si este término no estuviera, entonces la ecuación sería

$$x'' + x = 5\cos(t),$$

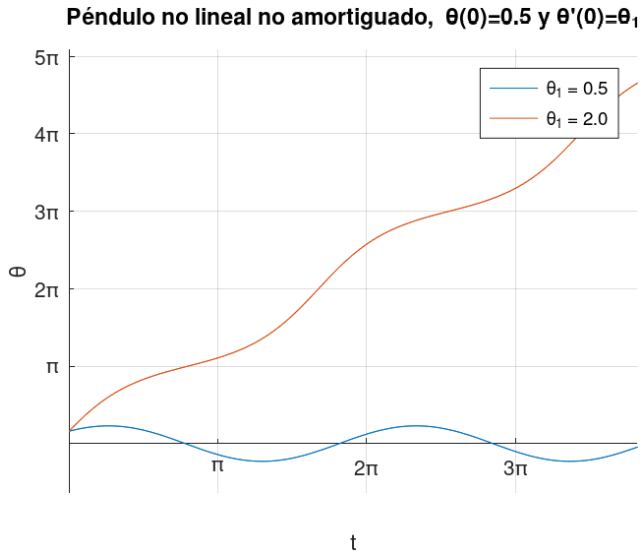
cuya solución general es $x(t) = c_1\cos(t) + c_2\,\text{sen}(t) + \frac{5}{2}t\sin(t)$. Al imponer las condiciones iniciales, tenemos que la solución al PVI es

$$x(t) = \cos(t) + \tfrac{5}{2}t\,\text{sen}(t)\,.$$

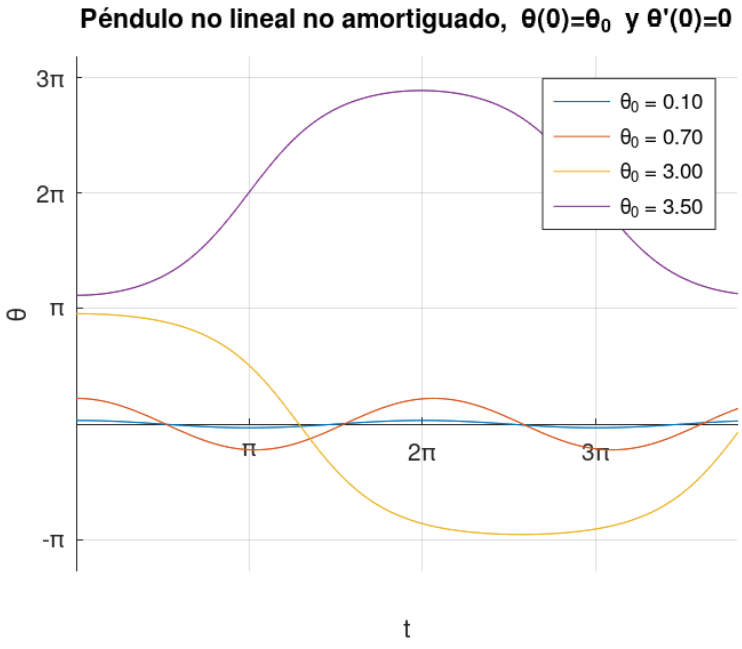
El término k_1x^3 introduce efectos no lineales en un sistema que, en ausencia de este término, exhibe resonancia cuando la frecuencia de la fuerza externa coincide con la frecuencia natural del sistema. La no linealidad puede modificar la resonancia de manera significativa, incluso cuando k_1 es pequeño (limita la amplitud resonante, estabilizando las oscilaciones).

Para $k_1 < 0$, el valor más pequeño para el que la respuesta es oscilatoria es, aproximadamente, $k_1 = -0.000487$. Con $k_1 = -0.000488$ ya no hay oscilación.

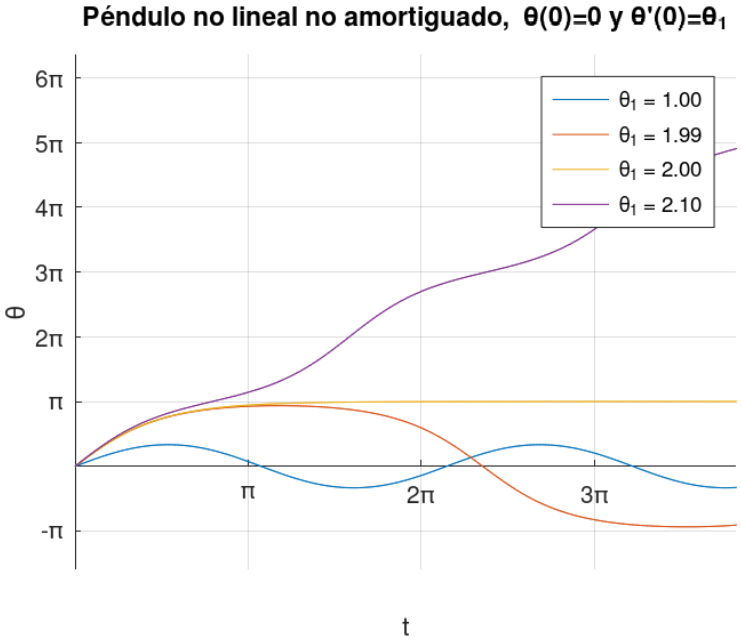
11. Para todos los incisos se utilizó el método de RK4 en el intervalo $[0, 12]$ con $N = 1000$, por lo que los resultados pueden diferir si se utilizan otros métodos o parámetros. La imagen a continuación incluye los casos *a*) y *b*) juntos, que comparten la misma posición inicial. En ambos, al igual que en los últimos cuatro, $\theta'(0) > 0$, lo que indica un movimiento inicial en sentido antihorario. Notar que coinciden con las curvas de la Figura 5.3.4 de la página 224 del libro [4]. El movimiento del péndulo simple, sea lineal o no lineal, tiene un comportamiento periódico cuando no hay rotaciones completas, y su período de oscilación tiene relación con π. Por este motivo suele etiquetarse también el eje t con múltiplos de π, como en el libro.



En el gráfico siguiente se incluyen las condiciones de los incisos *c*), *d*), *e*) y *f*), ya que todos partes desde el reposo en distintas posiciones iniciales.



Finalmente, se ilustran los últimos cuatro casos, que parten desde la posición de equilibrio con diferentes velocidades.



Algunas observaciones en relación con los gráficos anteriores. El comportamiento del péndulo cambia cualitativamente dependiendo de su energía inicial:

- Oscilaciones (sin rotación completa). En este caso el ángulo θ permanece acotado, oscilando en torno a $\theta = 0$ con un movimiento periódico.
- Rotaciones completas (el péndulo da la vuelta). Sucede cuando la energía inicial es lo suficientemente alta como para que el péndulo no regrese, sino que sigue girando en la misma dirección indefinidamente. En este caso la función θ no es periódica, ya que crece o decrece sin acotarse.

El péndulo pasa de oscilación a rotación completa cuando la energía total supera el valor necesario para alcanzar $\theta = \pi$. Iniciando desde $\theta_0 = \theta(0) = 0$, el péndulo tiene exactamente la energía para llegar a la vertical opuesta con velocidad nula cuando $\theta'(0) = 2$. Se "frena" justo en $\theta = \pi$, como se ilustra en el caso i). Con cualquier velocidad inicial menor, el movimiento es oscilatorio, y es de rotaciones completas si es mayor que 2. De forma general, este valor crítico para la energía se obtiene calculando $\omega_0 = \sqrt{2(1 + \cos(\theta_0))}$, y su relación con $\theta'(0)$ determinará el comportamiento cualitativo del péndulo.

En el caso b), a diferencia del i), la posición inicial es $\theta(0) = 0.5$. El valor crítico correspondiente es $\omega_0 \approx 1.938$, y cualquier velocidad inicial mayor a este valor producirá rotación completa.

Los gráficos anteriores ilustran cómo varía el ángulo en función del tiempo. A partir de esto podemos representar el movimiento del péndulo. En efecto, la posición de la masa colocada en el extremo de la varilla, en cada instante de tiempo, está dada por $(L\,\mathrm{sen}(\theta), -L\cos(\theta))$. El siguiente código grafica el movimiento del péndulo con las condiciones iniciales del primer inciso:

```
%Datos
f=@(t,y) [y(2); -sin(y(1))];  % defino la función
inter=[0 12];                 % intervalo de solución
N=1000;                       % cantidad de pasos
L=9.8;                        % largo de la varilla

%Condiciones iniciales
y0=[0.5;0.5];  % Caso A

[t,R] = rk4(f, inter, y0, N);     % Método RK4

for s = R(1,1:10:end)
  plot([0 L*sin(s)], [0 -L*cos(s)],'b','LineWidth', 2) %dibuja
     la varilla
  hold on
  plot(L*sin(s),-L*cos(s),'ro','MarkerSize',8,'MarkerFaceColor
    ','r') %masa
  axis([-L-1 L+1 -L-1 L+1])  %configura ejes
  axis equal                 %configura ejes
  title('Movimiento del péndulo','FontSize', 16);
  pause(0.01)
  hold off

end
```

Accediendo al siguiente enlace se puede observar el movimiento del péndulo durante los primeros 12 segundos en cada uno de los casos, junto con la gráfica del ángulo para comparar y comprender.

12. El término $d\theta/dt$ representa la velocidad angular del péndulo. Es la rapidez con la que cambia el ángulo θ en función del tiempo, e indica qué tan rápido está oscilando el péndulo en un instante dado. Su presencia en la ecuación puede incluir efectos de amortiguamiento si hay resistencia (como fricción o pérdida de energía debido al medio). La Figura S.2 contiene el gráfico de la variación de θ para las diferentes elecciones de velocidades iniciales.
13. El gráfico que ilustra el movimiento de un péndulo no lineal amortiguado puede verse en la Figura S.3.

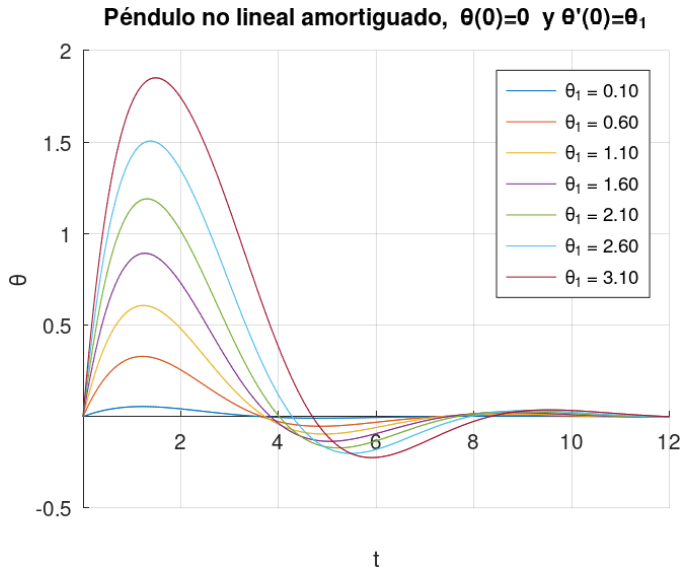


Figura S.2: Gráfico para el Ejercicio 12.

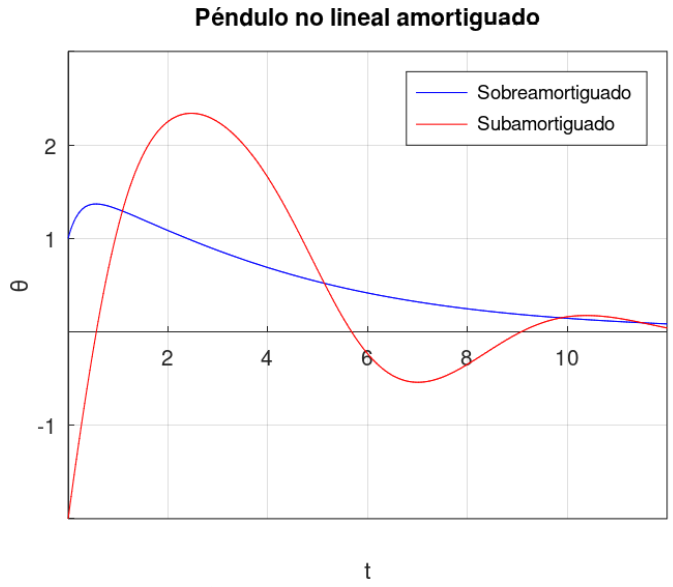
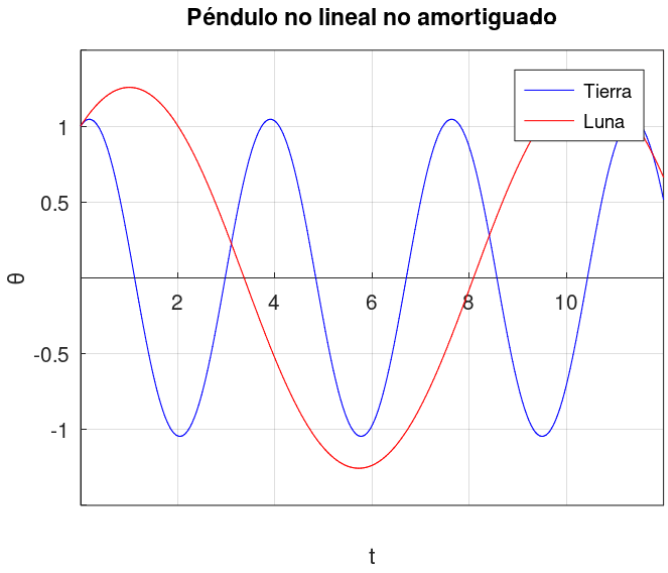


Figura S.3: Gráfico para el Ejercicio 13.

14. El péndulo oscila más rápido en la Tierra, y tiene mayor amplitud de movimiento en la Luna, como puede observase en el siguiente gráfico.



15. Primero recordar calcular la masa m y convertir los 500 km/h en m/s. Sea $K = k/m = 0.0005194$, aproximadamente. Tenemos los siguientes PVI

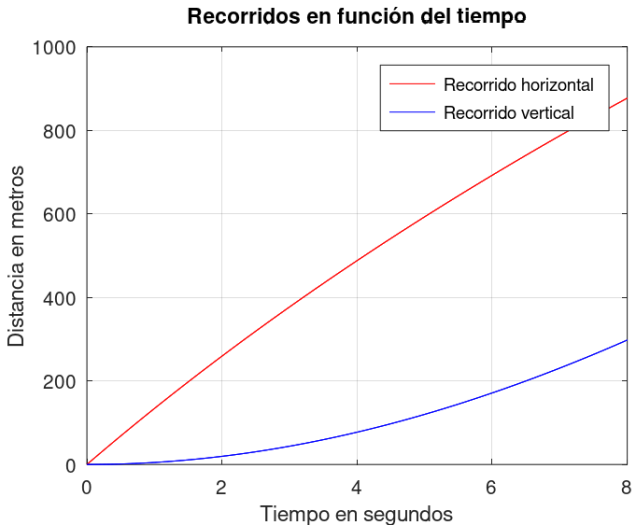
$$\text{(PVI1)} \qquad x'' = -K(x')^2, \qquad x(0) = 0, \quad x'(0) = 138.889.$$

$$\text{(PVI2)} \qquad y'' = g - K(y')^2, \qquad y(0) = 0, \quad y'(0) = 0.$$

La idea es resolver primero el PVI2 y encontrar el valor t_0 de tiempo tal que $y(t_0) \approx 300$. Luego, resolvemos el PVI1 y evaluamos $x(t_0)$. El siguiente código realiza este procedimiento, grafica y muestra la respuesta en pantalla:

```
g=9.8; K=0.0005194; %constantes del problema
x0=[0;138.889];     %ci PVI1
y0=[0;0];           %ci PVI2
N=1600;             %cantidad de pasos
inter=[0,8];        %intervalo de tiempo

fx=@(t,z) [z(2); -K*(z(2))^2];  %para x
fy=@(t,z) [z(2); g-K*(z(2))^2]; %para y

[t,x] = rk4(fx, inter, x0, N); %sol PVI1
[t,y] = rk4(fy, inter, y0, N); %sol PVI2

%gráfico de los recorrido en función del tiempo
plot(t,x(1,:),'r','DisplayName','Recorrido horizontal',t, y
    (1,:), '-b', 'DisplayName','Recorrido vertical'); %x,y
grid on
legend
xlabel('Tiempo en segundos');
ylabel('Distancia en metros');
title('Recorridos en función del tiempo');
```

```
T=find(y(1,:)<300); %posiciones de los tiempo donde no
    llego al piso
t0=T(end);          %me interesa el último
d=x(1,t0);          %distancia horizontal recorrida

% Mostrar la leyenda con distancia y tiempo
disp(['Distancia horizontal recorrida: ', num2str(d), '
    metros']);
disp(['Tiempo transcurrido: ', num2str(t(t0)), ' segundos'
    ]);
```



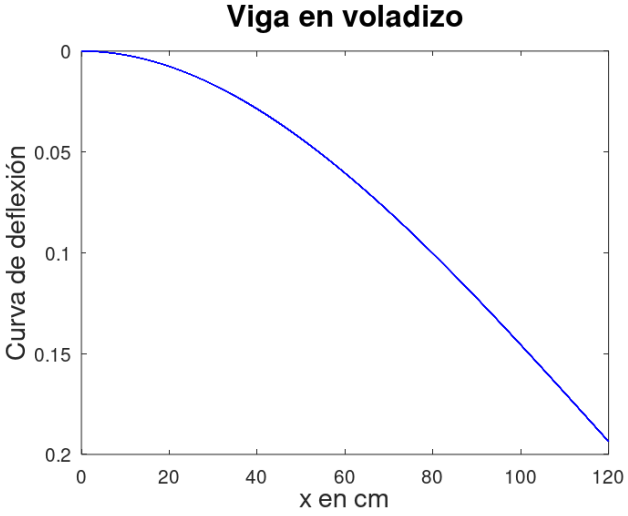
La salida del código es la imagen anterior junto con las leyendas:

```
Distancia horizontal recorrida: 877.1564 metros
Tiempo transcurrido: 8 segundos
```

16. El siguiente código resuelve el problema y muestra en pantalla las respuestas. Además, grafica la curva de deflexión de la viga.

```
P=3000; I=4250;  L=120; E=2.1e+6;  %constantes del problema
y0=[0;0];      %ci PVI

N=24000;             %cantidad de pasos
inter=[0,L];         %intervalo de tiempo

f=@(x,y) [y(2); P*(1+(y(2))^2)^(1.5)*(L-x)/(E*I)];

[x,y] = rk4(f, inter, y0, N); %sol PVI

%gráfico
plot(x,y(1,:),'b');
set(gca, 'YDir', 'reverse'); %sentido positivo hacia abajo
xlabel('x en cm','FontSize',14);
ylabel('Curva de deflexión','FontSize',14);
```

```
title('Viga en voladizo','FontSize',16);

%valores a hallar
[M,p]=max(y(1,:));      %máxima deflexión y posición
m=find(y(2,:)>0.0019); %posiciones donde la pendiente es
    mayor que 0.0019
x0=m(1);                %me interesa la primera

% Mostrar la leyenda
disp(['Máxima deflexión: ',num2str(M),' cm']);
disp(['Posición de máxima deflexión: ',num2str(x(p)),' cm'
    ]);
disp(['La pendiente es mayor que 0.0019 desde los ',num2str
    (x(x0)),' cm']);
```



La salida del código es la imagen anterior junto con las leyendas:

```
Máxima deflexión: 0.19361 cm
Posición de máxima deflexión: 120 cm
La pendiente es mayor que 0.0019 desde los 64.37 cm
```

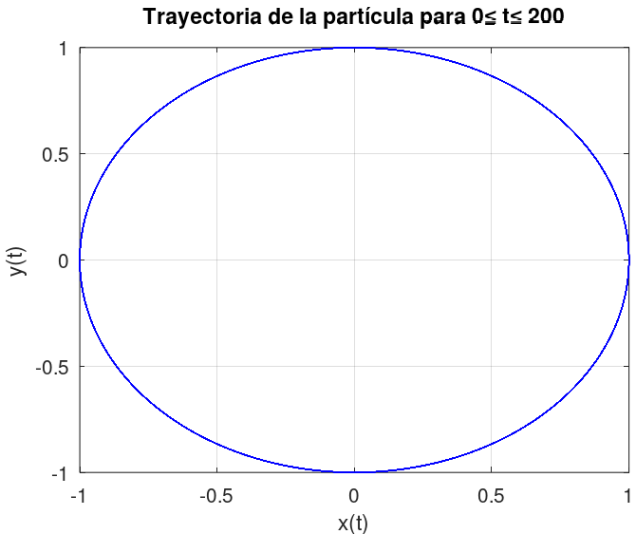
Sección 2.3

Ver enunciados en página 45

1. *a*) Llamamos $y_1 = x$, $y_2 = y$, $y_3 = x'$, $y_4 = y'$. Así, el sistema equivalente es

$$\begin{cases} y_1' = y_3 \\ y_2' = y_4 \\ y_3' = -\dfrac{k}{m}\dfrac{y_1}{(y_1^2+y_2^2)^{3/2}} \\ y_4' = -\dfrac{k}{m}\dfrac{y_2}{(y_1^2+y_2^2)^{3/2}} \end{cases}$$

b) Las condiciones iniciales equivalen a $y_1(0) = 1$, $y_2(0) = 0$, $y_3(0) = 0$, $y_4(0) = 1$. La trayectoria obtenida con el método y paso indicado es un círculo de radio 1 centrado en el origen de coordenadas.



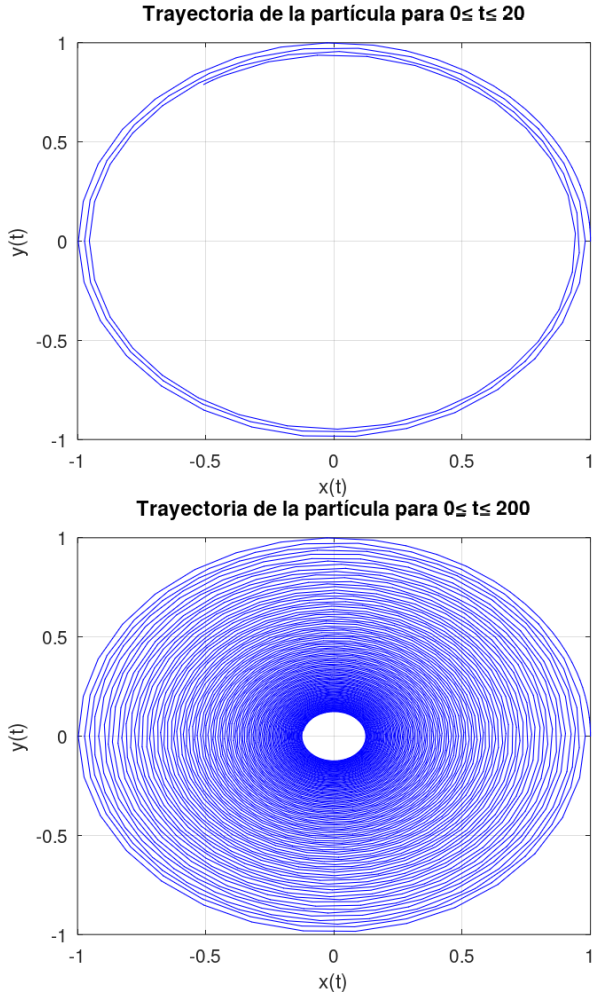
c) Las trayectorias generadas utilizando ode45 para intervalos de tiempo cada vez mayores tienen aspecto de espiral hacia el origen, como se ilustra en las imágenes posteriores.

En un sistema de fuerzas centrales, específicamente con una fuerza de tipo gravitatorio o electrostático (ley del inverso del cuadrado), las trayectorias posibles son secciones cónicas (elipse, parábola o hipérbola). Esto depende del valor de la energía total.

En particular, el movimiento es circular si la partícula se lanza tangencialmente con la velocidad adecuada para que la distancia al centro permanezca constante. Más precisamente, si la partícula inicia en una posición (x_0, y_0) a una distancia $A = \sqrt{x_0^2 + y_0^2}$, el vector de velocidad inicial $\vec{v}_0$ debe ser perpendicular al vector posición (para que no haya componente radial), y su magnitud v_0 debe ser igual a $\sqrt{\frac{k}{mA}}$. Esto produce un movimiento en el que la fuerza de atracción actúa como fuerza centrípeta, manteniendo el radio constante sin modificar la magnitud de la velocidad.

En este caso $(x_0, y_0) = (1, 0)$ y $\vec{v}_0 = (0, 1)$. Así, la trayectoria esperada es una circunferencia de radio 1 centrada en el origen, tal como se observa al aplicar RK4. Esto puede resultar contradictorio, ya que ode45 suele tener mayor precisión. Sin embargo, un método más preciso localmente no necesariamente reproduce mejor el comportamiento global del sistema. Al tratarse de una fuerza central, se conservan dos magnitudes fundamentales: el momento angular y la energía. Si la energía total disminuye artificialmente por errores numéricos acumulados, la trayectoria se vuelve una espiral hacia el origen.

Al construir aproximaciones numéricas, ninguno de estos métodos conserva exactamente estas cantidades: ambos introducen errores de distinta naturaleza, ya que no son métodos simplécticos (esto es, métodos diseñados para preservar propiedades estructurales de sistemas físicos). En este caso, el método de RK4 con un paso fijo muy pequeño puede dar una apariencia de estabilidad en intervalos de tiempo moderados. En cambio, el ajuste adaptativo de `ode45` puede producir un error acumulado en la energía, especialmente si las tolerancias no son lo suficientemente estrictas, lo que se manifiesta visualmente en forma de espiral.



Como conclusión, un método con alta precisión local en cada paso no garantiza una representación adecuada del comportamiento a largo plazo si el error acumulado afecta propiedades estructurales del sistema, en particular la conservación de la energía.

2. Escribimos

```
f=@(t,y) [y(3); y(4); -y(1)/((y(1)^2+y(2)^2)^(1.5));
          -y(2)/((y(1)^2+y(2)^2)^(1.5))];
y0=[1;0;0;1];
inter=[0 pi];
N=1000;
[t,y]=rk4(f,inter,y0,N);
```

Recordemos que $t_1 = 0$, $t_2 = h$, $t_3 = 2h, \ldots, t_i = (i-1)h$. Así, para $t = \pi/4$ corresponde la posición $i = 251$, para $t = \pi/2$ el lugar $i = 501$, para $t = 3\pi/4$ es $i = 751$, y para $t = \pi$ corresponde la última posición, que es $i = 1001$.

$$\begin{aligned} &y(1,251) = 0.7071, && y(2,251) = 0.7071;\\ &y(1,501) = -4.6641e-13, && y(2,501) = 1.0000;\\ &y(1,751) = -0.7071, && y(2,751) = 0.7071;\\ &y(1,1001) = -1.0000, && y(2,1001) = -7.0508e-12. \end{aligned}$$

Notar que todos los resultados son buenas aproximaciones de lo esperado.

3. Para el primer inciso, comencemos escribiendo $\vec{F}$ como una única expresión en términos de la función de Heaviside:

$$\begin{aligned} \vec{F}(x,y) &= -m\vec{g} - 20\vec{v}\mathscr{U}(y-10)\\ &= -20x'\mathscr{U}(y-10)\vec{i} + \big(-mg - 20y'\mathscr{U}(y-10)\big)\vec{j}. \end{aligned}$$

Esta ya es la fuerza resultante, por lo que la segunda ley de Newton implica que las ecuaciones para x e y son

$$x''(t) = -\frac{20}{m}x'\mathscr{U}(y-10), \qquad y''(t) = -g - \frac{20}{m}y'\mathscr{U}(y-10),$$

sujetas a las condiciones iniciales $x(0) = y(0) = 0$, $x'(0) = v_0\cos(\theta_0)$, $y'(0) = v_0\,\mathrm{sen}(\theta_0)$. Para escribir un sistema de primer orden equivalente, sean $y_1 = x$, $y_2 = y$, $y_3 = x'$, $y_4 = y'$. Entonces el sistema equivalente es

$$\begin{cases} y_1' = y_3\\ y_2' = y_4\\ y_3' = -\frac{20}{m}y_3\mathscr{U}(y_2-10)\\ y_4' = -g - \frac{20}{m}y_4\mathscr{U}(y_2-10), \end{cases}$$

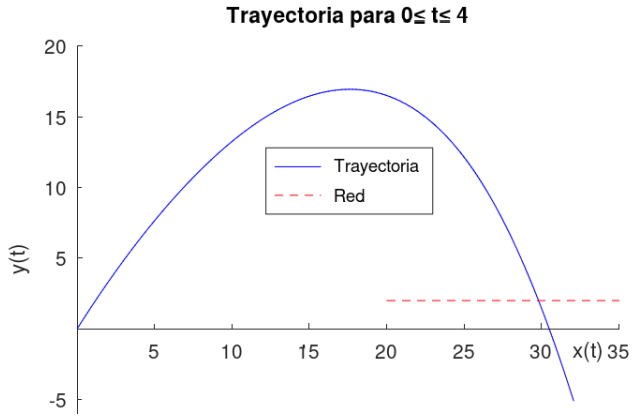
y las condiciones iniciales se traducen en

$$y_1(0) = 0, \quad y_2(0) = 0, \quad y_3(0) = v_0\cos(\theta_0), \quad y_4(0) = v_0\,\mathrm{sen}(\theta_0).$$

Para responder las preguntas de los incisos restantes se procede de igual forma que en el Ejemplo 9, cambiando solamente `T=find(y(2,:)>0)` por `T=find(y(2,:)>2)` y, por supuesto, modificando la función y las constantes del problema. Aquí, `f=@(t,y) [y(3);y(4); -(20/m)*y(3)*(y(2)>=10); -g-(20/m)*y(4)*(y(2)>=10)]`.

El código arroja las siguientes respuestas y el gráfico de la trayectoria (por supuesto, el hombre no atraviesa el suelo, de hecho queda en la red).

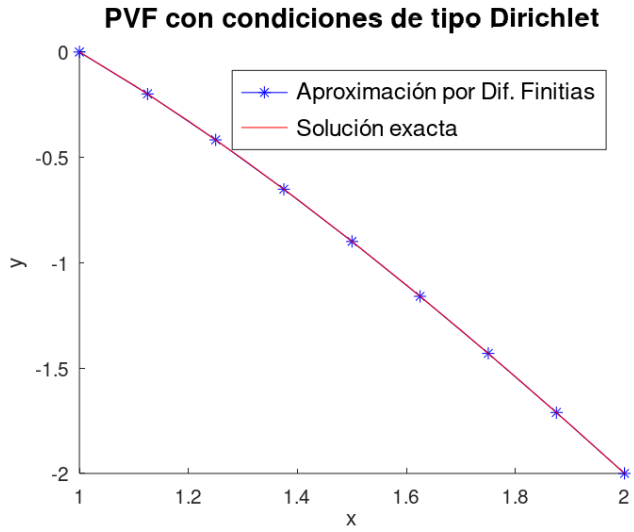
```
Altura máxima alcanzada: 16.9445 metros
Tiempo en alcanzarla: 1.76 segundos
Distancia horizontal recorrida hasta la red: 29.7999 metros
Tiempo transcurrido: 3.61 segundos
```



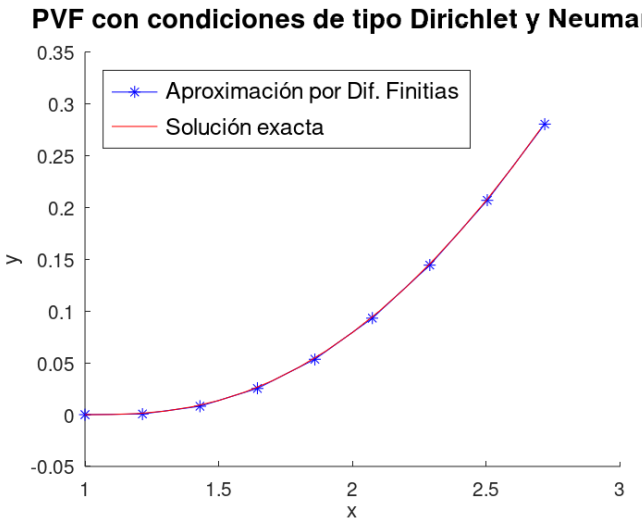
Sección 3.2

Ver enunciados en página 64

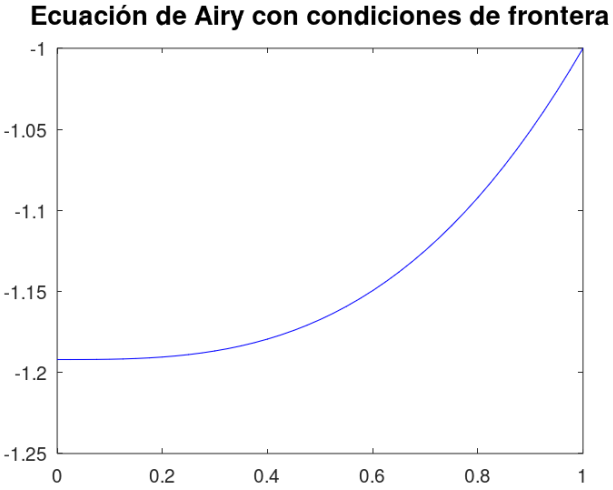
1. La solución general de la ecuación diferencial es $y(x) = c_1 x + c_2 x \ln(x) + \ln(x) + 2$. Imponiendo las condiciones de borde, se tiene que la solución al PVF es $y(x) = -2x - \frac{1}{2} x \ln(x) + \ln(x) + 2$. El siguiente gráfico contiene la curva correspondiente a esta función, así como la aproximación mediante diferencias finitas con $N = 8$.



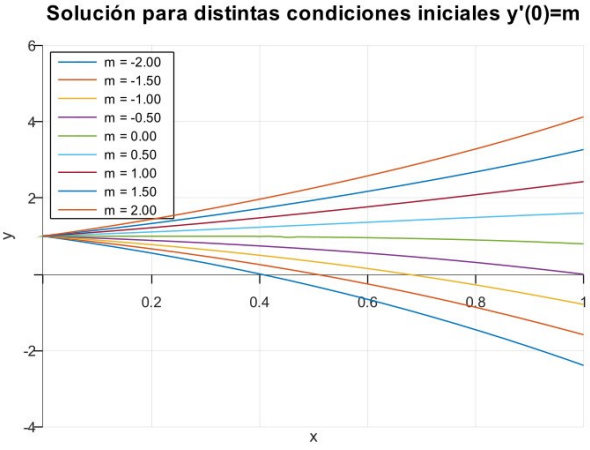
2. La solución al PVF ahora es $y(x) = -2x + (x+1)\ln(x) + 2$. Se incluye a continuación la gráfica de y junto con las aproximaciones por diferencias finitas.

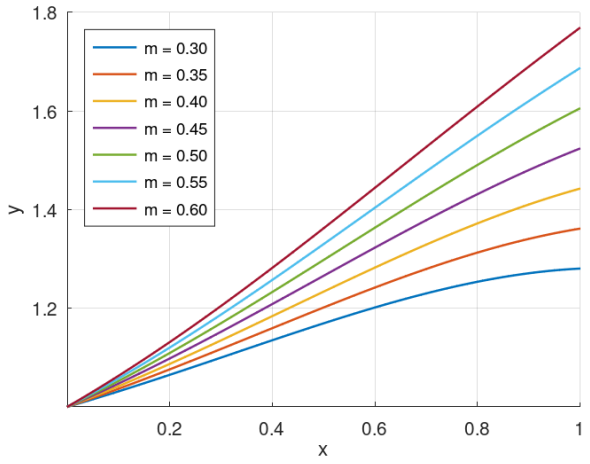


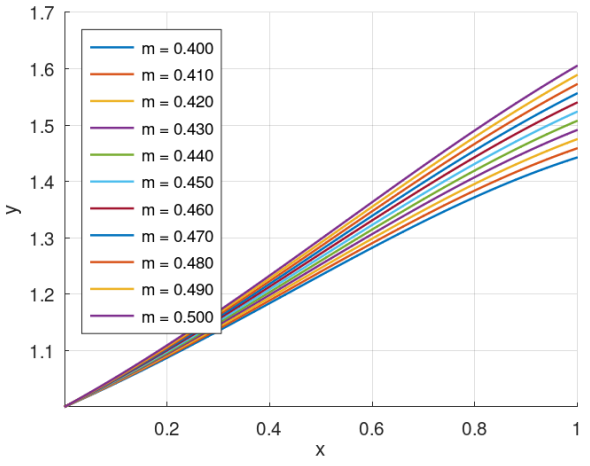
3. El gráfico de la curva solución aproximada es el siguiente.



4. Para el inciso *a*), en el primer gráfico de los tres que aparecen a continuación podemos ver que, dentro de este rango de pendientes iniciales, la que mejor se ajusta a la condición de borde $y(1) = 1.5$ es $m = 0.5$, por lo que repetimos el proceso con valores cercanos. A partir de ello construimos los dos últimos gráficos.


Solución para distintas condiciones iniciales y'(0)=m
m = -2.00
m = -1.50
m = -1.00
m = -0.50
m = 0.00
m = 0.50
m = 1.00
m = 1.50
m = 2.00
y
x


m = 0.30
m = 0.35
m = 0.40
m = 0.45
m = 0.50
m = 0.55
m = 0.60
y
x


m = 0.400
m = 0.410
m = 0.420
m = 0.430
m = 0.440
m = 0.450
m = 0.460
m = 0.470
m = 0.480
m = 0.490
m = 0.500
y
x

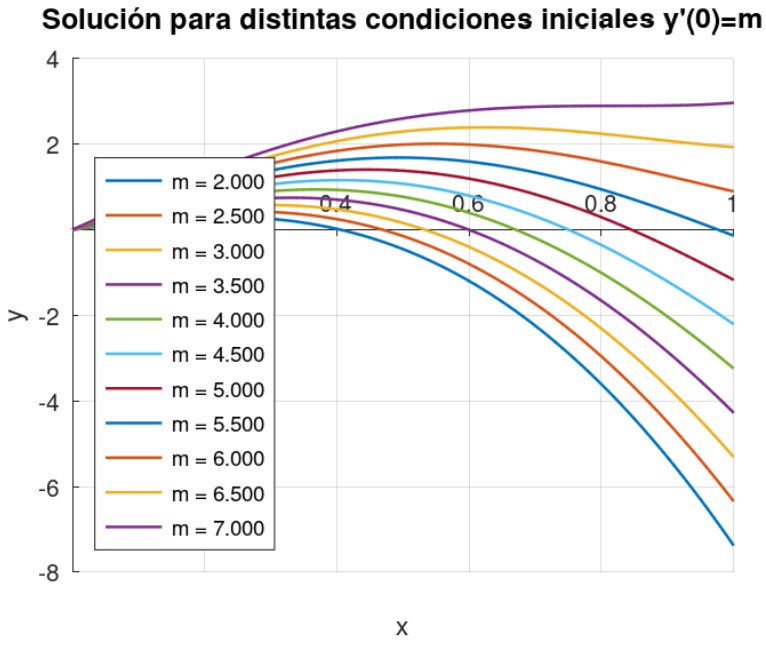
Además, dentro del `for` podemos construir una variable booleana que nos permita concluir si estamos dentro de la tolerancia, escribiendo

```
error = abs(y(1,N+1) - 1.5);      % Calcula el error actual
dentro_tolerancia = (error<0.01)   % Será 1 (Sí) o 0 (No)
```

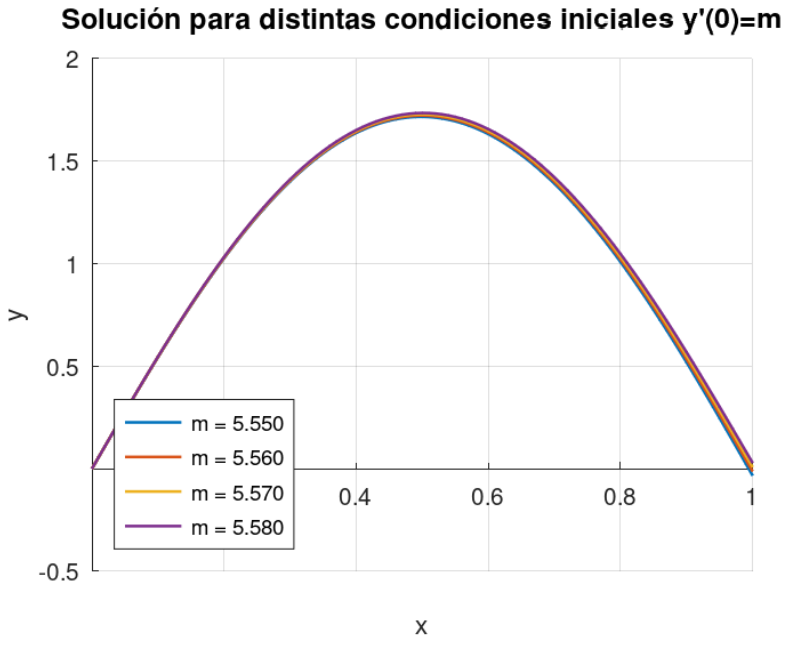
Esto nos permite concluir que, para `m=0.4:0.01:0.5` solamente obtenemos aproximaciones dentro de la tolerancia para $m = 0.43$ y $m = 0.44$.

Con respecto a la pregunta del inciso *b*), el hecho de que la ecuación dependa de y en el término sen(xy) no impide aplicar diferencias finitas. Sin embargo, introduce una no linealidad en el sistema de ecuaciones que se obtiene. Los métodos para resolver el sistemas no lineales escapan a los contenidos de este curso.

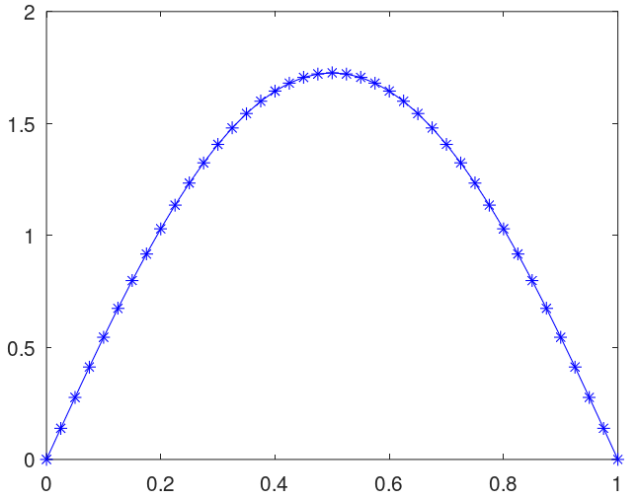
5. *a*) Comenzamos probando para diferentes valores de m, ajustando la precisión según los gráficos.



Además, mediante el uso de la variable booleana presentada en el ejercicio anterior podemos ver que $m = 5.57$ arroja una solución dentro de la tolerancia establecida: `dentro_tolerancia = 1`.



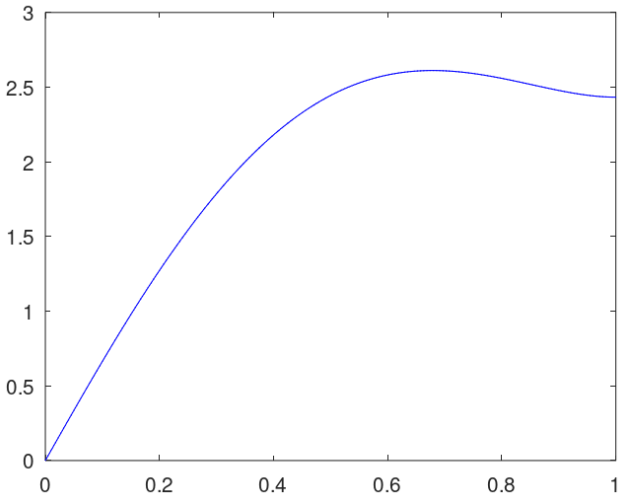
b) Utilizando el método de diferencias finitas con $N = 40$ se obtiene el siguiente gráfico como aproximación para la solución.



c) Por diferencias adelantadas, tenemos que $y'(0) \approx \frac{y_1 - y_0}{h} = \frac{y_1}{h}$, debido a la condición de borde. Para calcularlo en Octave con lo obtenido al ejecutar el código con $N = 40$, escribimos `y(2)/h`, lo que arroja como resultado `ans = 5.5613`. Notar que este valor es cercano al valor de m obtenido en el primer inciso.

6. *a*) Resolviendo por tanto para `m=6.742:0.0002:6.744`, y colocando una variable booleana `dentro_tolerancia = (abs(y(2,N+1)) < 0.001)`, se obtiene que los valores $m = 6.7420$ y $m = 6.7422$ satisfacen lo requerido. Puesto que la fila 2 de la matriz de salida almacena aproximaciones para y', la condición impuesta asegura que $y'(1) \approx 0$, tal como lo impone la condición de borde del problema.

   b) Utilizando el método de diferencias finitas con $N = 400$ se obtiene el siguiente gráfico como aproximación para la solución.



   c) Hallemos aproximaciones para $y'(0)$ y para $y'(1)$, mediante diferencias finitas adelantada y atrasada, respectivamente:

```
mi=(y(2)-y(1))/h
mi  =  6.7423
mf=(y(N+1)-y(N))/h
mf  =  -0.015206
```

   Notar que m_i, la aproximación para la pendiente inicial $y'(0)$, se acerca a los valores de m hallados en el primer inciso. Asimismo, la pendiente final m_f se aproxima a cero, como se deseaba.

Sección 3.3

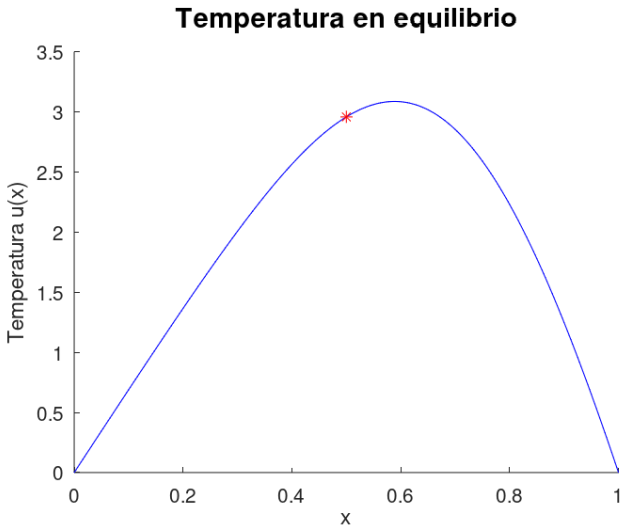
Ver enunciados en página 66

1. *a*) El PVF es $-ku'' = E(x)$ para $0 < x < 1$, $u(0) = u(1) = 0$.

   b) Para resolverlo aplicamos el método de diferencias finitas para condiciones de tipo Dirichlet, con $a = 0$, $b = 1$, $\alpha = \beta = 0$, $P(x) = Q(x) = 0$ y $f(x) = E(x)/(-k)$. Usaremos siempre N par, de modo que $x_{N/2+1} = 0.5$, correspondiendo a la mitad de la barra. Los siguientes valores se obtienen con $N = 10, 20, 40, 80, 160, 320, 640$, respectivamente:

```
u_medio = 2.9749
u_medio = 2.9632
u_medio = 2.9603
u_medio = 2.9596
u_medio = 2.9594
u_medio = 2.9593
u_medio = 2.9593
```

Podemos observar que la precisión deseada se alcanza ya para $N = 80$, que corresponde a $h = 0.0125$. El gráfico obtenido es el siguiente:



c) El gráfico obtenido muestra que la temperatura no alcanza su valor máximo exactamente en el punto donde la fuente E es máxima (en $x = 0.7$), sino en un punto cercano, desplazado hacia la izquierda (aproximadamente en $x = 0.59$).

Aunque inicialmente podría esperarse que el máximo de u ocurra donde la generación de energía es mayor, la distribución final está fuertemente condicionada por las condiciones de borde.

En efecto, como la temperatura en ambos extremos de la barra es 0 grados, el calor generado se disipa hacia ambos lados. Dado que el punto $x = 0.7$ se encuentra más próximo al extremo derecho que al izquierdo, el flujo de calor hacia la derecha es mayor que hacia la izquierda. Esta asimetría en la disipación provoca que la temperatura disminuya más rápidamente en esa dirección. Como consecuencia, el equilibrio entre generación y disipación de calor se alcanza en un punto ligeramente desplazado hacia la izquierda del máximo de la fuente, donde la temperatura resulta ser mayor.

Este fenómeno ilustra que la solución de la ecuación de transferencia de calor tiene carácter global: la distribución de temperatura no depende únicamente de la fuente, sino también de cómo el calor se propaga y se disipa en el dominio.

2. El cambio con respecto al ejercicio anterior es $Q(x) = -r(x)/k$. Así, debemos considerar los casos $Q(x) = -2$ y $Q(x) = -10$. Para $c = 1$ se obtiene que $u(0.5) \approx 2.4554$, mientras que para $c = 5$ tenemos que $u(0.5) \approx 1.4554$. Podemos observar el comportamiento de u en el gráfico que se incluye en la Figura S.4.

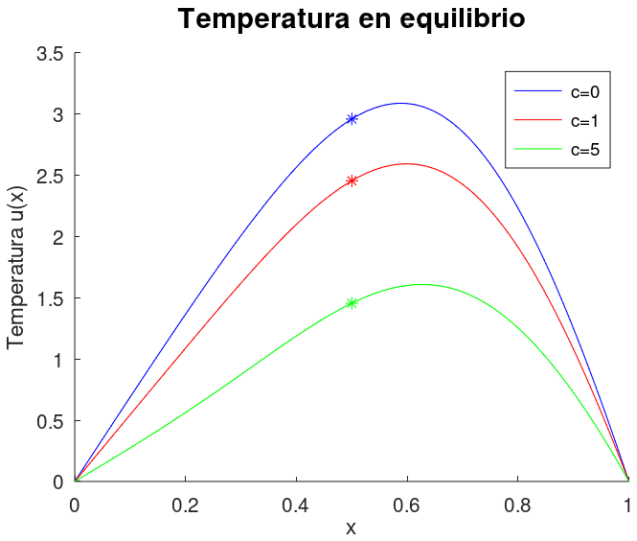


Figura S.4: Gráfico para el Ejercicio 2.

El término $cu(x)$ actúa como una pérdida proporcional de temperatura: reduce la temperatura contrarrestando el aporte de calor generado por $E(x)$. A medida que c crece, el efecto de enfriamiento (o disipación proporcional a la temperatura) es más fuerte. Esto explica por qué las gráficas obtenidas son más planas a medida que c aumenta.

En general, en la ecuación

$$-ku''(x) + r(x)u(x) = E(x).$$

el término $r(x)u(x)$ es el que describe el efecto de la reacción:

- Cuando $r(x) > 0$, el término $r(x)u(x)$ representa una pérdida de energía proporcional a la temperatura, contribuyendo a disipar calor (efecto de enfriamiento). Esto ocurre, por ejemplo, en materiales que disipan calor hacia el entorno o en sistemas donde se consuma energía térmica debido a reacciones químicas o fenómenos físicos específicos.
- Si $r(x) < 0$, el término $r(x)u(x)$ genera calor (efecto de calentamiento), y contribuye a generar energía térmica proporcional a la temperatura. Esto puede darse en sistemas donde las reacciones químicas generen calor o donde haya retroalimentación positiva en la producción de energía térmica.

3. *a*) La diferencia con el PVF del primer ejercicio es solamente la condición de frontera de tipo Robin en el extremo derecho, manteniendo la de tipo Dirichlet en el izquierdo. La condición de borde en $x = 1$ se reescribe como $u(1) + u'(1) = 1$, lo que corresponde a $C_2 = D_2 = G_2 = 1$ en (3.10). El gráfico que se obtiene se incluye en la Figura S.5.

   Con $N = 640$, la temperatura aproximada en el punto medio de la barra está dada por $u(0.5) = 6.5773$.

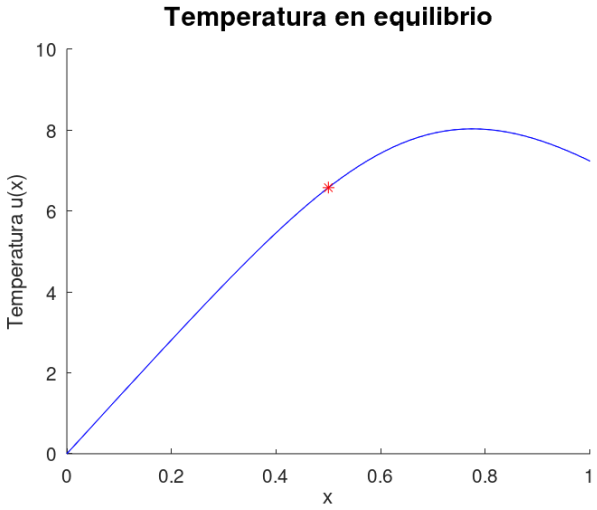


Figura S.5: Gráfico para el Ejercicio 3*a*.

b) Para corroborar que la condición en $x = 1$ se cumple, escribimos:

```
y(N+1)+(y(N+1)-y(N))/h
ans = 1.0127
```

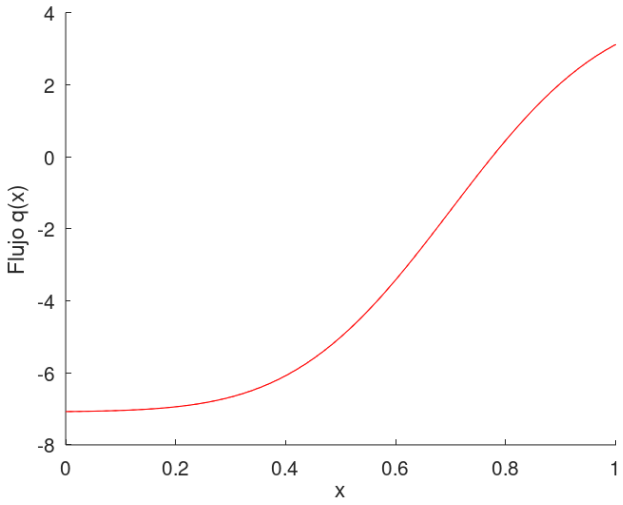
c) En el inciso anterior usamos diferencias atrasadas para aproximar $u'(1)$ mediante los valores obtenidos para u. Para $u'(0)$ usaremos diferencias adelantadas, mientras que para los valores interiores utilizaremos diferencias centradas. De esta forma tenemos una aproximación para u'. El siguiente código realiza esto y grafica el flujo q. Además calcula el flujo máximo, el mínimo, y las posiciones donde se alcanzan. Debe ejecutarse luego de resolver el problema, ya que usa lo obtenido previamente.

```
%Derivadas laterales en los extremos
du0=(y(2)-y(1))/h;
duN=(y(N+1)-y(N))/h;

%Inicializamos los vectores derivada
du=zeros(1,N+1);

%Primer y último elemento en cada uno
du(1)=du0; du(N+1)=duN;

%Aprox. las derivadas con diferencias finitas centradas
```

```
for i=2:N
    d = (y(i+1) - y(i-1)) / (2*h);
    du(i)=d;
end

q=-0.5*du;

%Valores máximos y mínimos
[M,P]=max(q);
[m,p]=min(q);

fprintf('El flujo máximo es %.4f y se alcanza en x = %.2
    f\n', M, x(P));
fprintf('El flujo mínimo es %.4f y se alcanza en x = %.2
    f\n', m, x(p));

%Gráfico
hold on
plot(x,q,'-r')
xlabel('x');
ylabel('Flujo q(x)');
hold off
```

La salida es:

```
El flujo máximo es 3.1116 y se alcanza en x = 1.00
El flujo mínimo es -7.0748 y se alcanza en x = 0.00
```

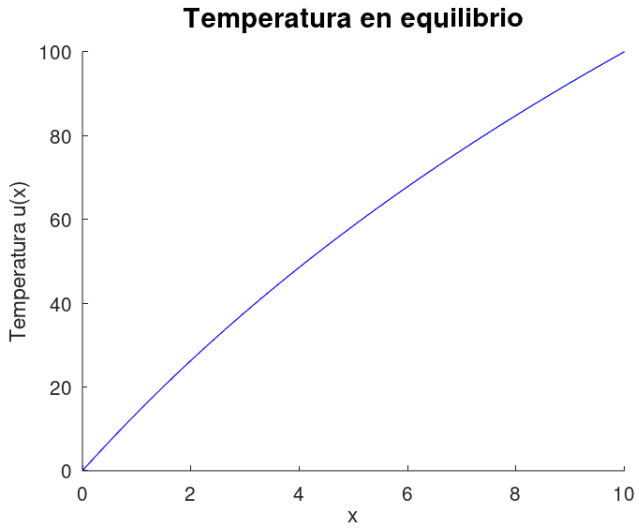


Notar que $q'(x) = (-ku'(x))' = -ku''(x) = E(x)$. En este caso, $E(x) > 0$ para todo x, lo que equivale a $q'(x) > 0$. Esto implica que el flujo de calor q crece, por lo que tiene sentido que el mínimo se alcance en el extremo izquierdo de la varilla, y el máximo en el derecho.

4. Para resolver el PVF aplicamos el método de diferencias finitas para condiciones de tipo Dirichlet, con $a = 0$, $b = 10$, $\alpha = 0$, $\beta = 100$,

$$P(x) = k'(x)/k(x) = 0.1/(1 + 0.1x),$$

$Q(x) = 0$ y $f(x) = 0$. Se obtiene la siguiente curva como aproximación al PVF:



Octave: lo básico

GNU Octave es un software libre utilizado para cálculos numéricos y científicos. Es similar a MATLAB, por lo que muchas funciones y comandos son compatibles entre ambos. En esta sección incluimos aspectos básicos sobre Octave, aunque la mayoría de ellos ya fueron descubiertos durante el desarrollo del texto.

La Ventana de comandos. Es la pantalla principal de Octave, el lugar donde se escriben las órdenes y se visualizan las respuestas. Los caracteres >> se denominan *prompt*. Indican que Octave está listo para recibir instrucciones del usuario.

El Editor. Es un espacio de trabajo que nos permite guardar secuencias de comandos o instrucciones secuenciales (llamados *script*) en archivos de extensión `.m`, así como crear funciones.

El directorio activo de trabajo (o directorio actual). Es donde Octave busca los archivos y *scripts* para ejecutar. También es el lugar donde se guardan por defecto los archivos generados. Para saber cuál es el directorio activo actual, usa el comando `pwd`.

Recomendamos crear una carpeta para guardar todo lo referido a este curso **cuyo nombre no contenga espacios**, solo para evitar problemas. Lo mismo aplica para archivos o subcarpetas. Una vez creada, se debe transformar en el directorio actual de trabajo desde el buscador que aparece arriba. Esto es importante, ya que es el punto de referencia desde el cual Octave opera al manejar archivos.

Algunas consideraciones generales sobre Octave

- Distingue entre mayúsculas y minúsculas.
- El signo ; al final de una instrucción evita ver en la pantalla la información del resultado.
- Para iniciar un comentario se utiliza el símbolo %. Todo lo que se escriba después, y hasta el final de la línea, es ignorado por Octave.
- Las teclas ↓ y ↑ sirven para buscar comandos usados previamente.
- Con las teclas ← y → nos podemos mover sobre la línea de comando.
- Para limpiar la ventana de comandos basta con escribir `clc`. Para liberar las variables utilizadas escribimos `clear all`. Las variables activas pueden visualizarse escribiendo `who`.

- Para recibir ayuda sobre cualquier comando, se escribe `help nombre_comando`. Por ejemplo, `help plot` arrojará ayuda sobre el comando `plot`, que permite graficar en Octave.
- El signo = funciona como asignación. Para comparar se utiliza ==.

Operaciones básicas con números reales

- Suma, resta, multiplicación, división y potencia entre números reales se escriben de forma usual: `a+b`, `a-b`, `a*b`, `a/b`, `a^b`.
- La prioridad de las operaciones en Octave es la usual: primero potencias, luego multiplicaciones y divisiones y, finalmente, sumas y restas. Las operaciones con igual prioridad se resuelven de izquierda a derecha.
- Se utilizan solo paréntesis para indicar distintos niveles de prioridad dentro de una expresión complicada (no se usan ni corchetes ni llaves).
- Están disponibles las operaciones/funciones clásicas como `sqrt`, `log`, `exp`, `sin`, `cos`, `abs`, `min`, `max`, entre muchas otras. Notar que `log(x)` equivale a $\ln(x)$.

Matrices y vectores

- Se definen usando corchetes, y separando filas con el signo "punto y coma". Los elementos de cada fila pueden separarse mediante comas o simplemente con espacios.
- Se usa `'` para transponer una matriz o vector: `A'` significa A^t, la transpuesta de A.
- El comando `eye(3)` crea la matriz identidad de orden 3. El comando `zeros(2,3)` crea una matriz de 2×3 formada por ceros, mientras que `ones(3,2)` crea una matriz de 3×2 formada por unos.
- Se usan paréntesis para acceder a elementos específicos:

  `M(2,3)` indica el elemento en la fila 2, columna 3 de la matriz M.

  `M(:,2)` indica toda la columna 2 de la matriz M.

  `M(end,:)` indica la última fila de la matriz M.

  `M([2 4],:)` indica las filas completas 2 y 4 de la matriz M.

  `M([2:4],:)` indica las filas completas 2 hasta 4 de la matriz M.
- La suma y resta de matrices de igual orden se hace de forma usual: `A+B`, `A-B`. El producto de una matriz A por una escalar c se escribe como `c*A`.
- El producto usual entre dos matrices compatibles A y B se escribe como `A*B`. Sin embargo, el producto elemento a elemento de dos matrices de igual orden se escribe como `A.*B`. De forma similar se divide elemento a elemento escribiendo `A./B`.
- Para elevar a la n cada elemento de una matriz A se escribe `A.^n`.
- Para resolver el sistema lineal $Ax = b$ se escribe `x=A\b`.
- Dada una matriz cuadrada M, con `det(M)` se calcula su determinante, y con `inv(M)` se halla su inversa (si existe). Con `[V,D] = eig(M)` se obtienen sus vectores y valores propios.

- El comando `size` se utiliza para obtener un vector con las dimensiones de una matriz o vector. Dice cuántas filas y columnas tiene una matriz o vector. Para la longitud de un vector se utiliza `length`.

Función versus script

- Un *script* es una lista de comandos que Octave ejecuta en secuencia, uno tras otro. Si deseamos ejecutar a menudo esta lista de instrucciones sin reescribirlas cada vez, podemos guardar el código en un archivo con extensión `.m`. Así, podremos ejecutarlo tantas veces como queramos escribiendo solo el nombre del archivo en el *prompt* de Octave.
- Las variables creadas en un *script* son globales, es decir, quedan disponibles en el entorno de trabajo después de ejecutarlo.
- Una función es un bloque de código reutilizable que toma entradas (parámetros), realiza algún cálculo o tarea, y devuelve resultados. También se guarda como un archivo `.m`, pero el nombre del archivo debe coincidir con el de la función principal dentro del archivo. Para usar una función se la llama desde el entorno de trabajo con sus parámetros.
- A diferencia de un *script*, una función tiene su propio espacio de variables. Las variables dentro de una función son locales, es decir, no afectan ni dependen del entorno de trabajo.
- Si una función está definida con varias salidas, pero no las pedimos de forma explícita, por defecto sólo devuelve la primera. Por ejemplo, consideremos la función definida como:

```
function [a,b] = mifun(x)
a = x^2;
b = x^3;
end
```

Si escribimos `mifun(2)` devolverá `ans = 4`. Escribiendo `[x,y]=mifun(2)`, la salida será `x=4` e `y=8`.

Gráficos

- Se usa `plot(x,y)` para graficar los puntos $(x(i), y(i))$, siendo x e y dos vectores de igual dimensión.
- El comando `hold on` congela el gráfico actual de modo que los siguientes se superponen en este. El comando `hold off` descongela el gráfico, de modo que el siguiente se realizará en una nueva figura y el anterior desaparecerá.
- Personalización del gráfico:
  - Agregando una cadena de caracteres como argumento adicional, se puede especificar el color, el tipo de línea y el marcador para los puntos del gráfico:

    `plot(x,y,'r--')` línea roja y discontinua sin marcadores.
    `plot(x,y,'b','LineWidth',2)` línea continua azul de grosor 2, sin marcadores.
    `plot(x,y,'mo')` marcadores circulares magenta, sin línea.
    `plot(x,y,'g*-')` línea sólida verde, con marcadores de asteriscos.

- Se puede agregar un título y etiquetas a los ejes con las indicaciones `title('Mi Título')`, `xlabel('Eje X')`, `ylabel('Eje Y')`.
- La instrucción `grid on` agrega una cuadrícula al gráfico.
- Escribiendo
  `set(gca,'XAxisLocation','origin','YAxisLocation','origin')` se centran los ejes en el origen.
- El comando `axis equal` utiliza la misma escala para graficar ambos ejes.
- El comando `legend` sirve para agregar leyendas a las gráficas, y se puede controlar dónde aparece la leyenda dentro del gráfico utilizando el argumento `'Location'`. También se puede controlar el tamaño con `'FontSize'`, su color, borde, etcétera.

Salida de texto y formateo de cadenas

- Se usa `disp` para mostrar texto o valores en pantalla. Por ejemplo, `disp('Hola mundo')` o `disp(x)`.
- Para formatear texto con control de estilo y precisión se utiliza `sprintf`. Por ejemplo, la instrucción

  ```
  disp(sprintf('El resultado es %.3f y N = %d', x, N))
  ```

  construye una cadena de texto donde:
  - `'El resultado es %.3f y N = %d'` es la plantilla o base, que es una frase con espacios reservados para los datos;
  - `%.3f` muestra un número real con 3 decimales (formato float);
  - `%d` muestra un entero;
  - `x` y `N` son los valores que se insertan en esos lugares, en ese orden;
  - `disp(...)` muestra en pantalla la cadena que devuelve `sprintf`.
- Para unir cadenas y valores convertidos en texto de forma simple también se usa `[ ... ]`. Por ejemplo, lo realizado en el inciso anterior con `sprintf` también puede hacerse como

```
disp(['El resultado es ',num2str(x, '%.3f'),' y N = ',num2str(N)])
```

  donde:
  - cada parte `'...'` es texto literal;
  - `num2str` convierte número a texto según el formato indicado. Si no se indica nada, mantiene el valor tal cual era;
  - `[...]` concatena (une) todas esas partes en una sola cadena;
  - `disp(...)` muestra el resultado en pantalla.
- La instrucción `fprintf` permite formatear texto y mostrarlo en pantalla, combinando las funcionalidades de `sprintf` y `disp`:

  ```
  fprintf('El resultado es %.3f y N = %d\n', x, N).
  ```

Aquí, `\n` indica un salto de línea (newline), ya que `fprintf` no agrega automáticamente un salto de línea al final, a diferencia de `disp`. Por eso, si queremos que el siguiente mensaje aparezca en una nueva línea, tenemos que incluir `\n`.

Conectores lógicos and y or

- El operador "and" devuelve `true (1)` si todas las condiciones son verdaderas. Si alguna condición es falsa, devuelve `false (0)`.
- El operador "or" devuelve `true (1)` si al menos una de las condiciones es verdadera. Solo devuelve `false (0)` si todas las condiciones son falsas.
- En Octave se utiliza el símbolo `&` para "and", mientras que `|` se usa para "or". En operaciones con vectores y matrices funcionan elemento a elemento.
- Versiones cortocircuitadas: si el primer argumento es suficiente para determinar el resultado, Octave no evalúa el segundo argumento, lo que puede mejorar el rendimiento. Para aplicarlas se utilizan los símbolos dobles `&&` y `||`, respectivamente. Se utiliza en condiciones escalares, ya que en matrices únicamente evaluará el primer elemento. Veamos un ejemplo.

Ejemplo 13. Dado un vector u de N componentes, determine cuántas veces cambia el signo de dos elementos consecutivos en el vector, lo que puede indicar la cantidad de raíces en caso de representar los valores de una función continua evaluada en una malla de puntos.

Solución. Escribimos

```
for i=1:N-1
% Comprobar si hay cambio de signo entre dos puntos
% consecutivos de u creando un vector lógico
  quasizeros(i)=(u(i)<0 && u(i+1)>0) || (u(i)>0 && u(i+1)<0);
end

% Contar cuántas veces se "anula"
num_ceros=sum(quasizeros)+sum(u==0); % agrega ceros exactos

% Mostrar resultado
fprintf('La función se anula %d veces en el intervalo.\n',
    num_ceros);
```

Analicemos cada instrucción. Comencemos con las condiciones del vector lógico `quasizeros`.

- **Primera condición:** `u(i) < 0 && u(i+1) > 0`.

  Esto verifica si el valor en `u(i)` es menor que 0 y si el valor en `u(i+1)` es mayor que 0. Es una condición para detectar un cruce de signo de negativo a positivo. Aquí, el uso de `&&` actúa como un operador de cortocircuito, lo que significa que si la primera condición `u(i) < 0` es falsa, Octave no evalúa la segunda parte `u(i+1) > 0`. Este "cortocircuito" ahorra tiempo de cómputo. Si la primera condición es verdadera, Octave sigue evaluando la segunda parte para determinar si la expresión total es verdadera.

- **Segunda condición:** `u(i) > 0 && u(i+1) < 0`.
  Es similar a la anterior, pero detecta un cambio de signo de positivo a negativo.
- **¿Hay optimización con el uso de | | que conecta ambas condiciones?** La operación también tiene evaluación cortocircuitada: si la primera condición es verdadera, entonces Octave no evalúa la segunda. Sin embargo, en este caso, las dos condiciones son mutuamente excluyentes salvo que sean nulos dos elementos consecutivos, cosa que raramente pase en problemas de aproximación numérica. Así, a excepción de ese caso, ambas partes del operador deben ser evaluadas en algún momento, y la optimización con cortocircuito no es tan significativa.

Ahora veamos las otras instrucciones, para entender qué hacen:

- `num_ceros` funciona como contador, ya que el vector que está sumando solo contiene ceros y unos (1 para verdadero y 0 para falso). Al sumarlo se obtiene la cantidad de veces que cambió de signo, lo que equivale a la presencia de raíces de la función involucrada. A esto se le adiciona la cantidad `sum(u==0)`, que representa el total de ceros exactos, si los hubiera.
- El ciclo `for` inicia un bucle que recorrerá los elementos del vector `u` desde el primero hasta el penúltimo. El índice `i` sirve para acceder a los elementos consecutivos del vector. Termina siempre con `end`, al igual que los ciclos `if` y `while`.
- Como vimos antes, se usa `fprintf` para imprimir un mensaje en pantalla. En este caso, el mensaje es: "La función se anula X veces en el intervalo", donde X es el número de veces que se encontró un cambio de signo en la función. Este valor es `num_ceros`.
- El `%d` es un marcador de posición en la cadena de texto que se pasa a la función `fprintf`. Su propósito es insertar un valor entero en el lugar donde aparece `%d`.
- La instrucción `\n` indica un salto de línea, y mueve el cursor a la siguiente línea después de imprimir el mensaje.

El código anterior es claro porque refleja el cambio de signo al recorrer el vector. Sin embargo, esto puede hacerse de forma más eficiente como se muestra a continuación, identificando los cambios de signo mediante la condición de producto negativo.

```
num_ceros = sum(u == 0); % ceros exactos

for i = 1:N-1
    if u(i)*u(i+1) < 0
        num_ceros = num_ceros + 1;
    end
end

% Mostrar resultado
fprintf('La función se anula %d veces en el intervalo.\n',
    num_ceros);
```

Comillas simples y comillas dobles

En Octave, las comillas simples `'...'` y las comillas dobles `''...''` no se diferencian: en ambos casos se obtiene un arreglo de caracteres (char). El tipo *string* existe, pero solo se crea explícitamente mediante la función `string(...)`. En MATLAB, en las versiones más modernas, sí hay una diferencia, ya que las comillas dobles producen un objeto de tipo *string*. Para asegurar compatibilidad entre Octave y MATLAB, se recomienda usar comillas simples al trabajar con texto, salvo que se necesiten explícitamente objetos *string* en MATLAB.

Índice alfabético

Sobre las autoras y los autores

MARILINA CARENA. Doctora en Matemática y se desempeña como Investigadora Independiente del CONICET y Profesora Asociada en la Facultad de Ingeniería y Ciencias Hídricas (FICH) de la Universidad Nacional del Litoral (UNL).

ANIBAL CHICCO RUIZ. Doctor en Matemática y se desempeña como Profesor Adjunto de la Facultad de Ingeniería Química (FIQ) y Jefe de Trabajos Prácticos de la Facultad de Ingeniería y Ciencias Hídricas (FICH), ambas de la Universidad Nacional del Litoral (UNL). También es Investigador Asistente del CONICET, con lugar de trabajo en el Instituto de Matemática Aplicada del Litoral (IMAL).

LUCAS GENZELIS. Ingeniero en Informática y se desempeña como Jefe de Trabajos Prácticos en la Facultad de Ingeniería y Ciencias Hídricas (FICH) de la Universidad Nacional del Litoral (UNL). Forma parte de las cátedras vinculadas a Ecuaciones Diferenciales, así como de las asignaturas Cálculo II y III.

ELISABET HAYE. Magíster en Matemática y se desempeña como Profesora Titular en la Facultad de Ingeniería y Ciencias Hídricas (FICH) de la Universidad Nacional del Litoral (UNL), responsable de las asignaturas de grado vinculadas a Ecuaciones Diferenciales de esta facultad.

MÉTODOS NUMÉRICOS PARA ECUACIONES DIFERENCIALES ORDINARIAS

C Á T E D R A

Esta obra es una introducción a algunos de los métodos numéricos clásicos para la resolución de ecuaciones diferenciales ordinarias; combina fundamentos teóricos, ejemplos ilustrativos y una amplia colección de ejercicios. Los métodos seleccionados destacan por su relevancia histórica y conceptual, así como por su utilidad en aplicaciones concretas.

El enfoque adoptado responde a las necesidades habituales de una asignatura de ecuaciones diferenciales ordinarias orientada a carreras de ingeniería; el texto no pretende constituir un tratado de cálculo numérico, sino una introducción a aquellas herramientas numéricas que complementan de manera natural el estudio cualitativo y analítico de las ecuaciones diferenciales.

UNIVERSIDAD
NACIONAL DEL LITORAL