

Tarea 4

Problema 1.

- a) Escriba una función que reciba como entrada $m \in \mathbb{N}$ y devuelva la matriz $A \in \mathbb{R}^{m \times m}$ dada por

$$A = \begin{bmatrix} 1 & & & & & 1 \\ -1 & 1 & & & & 1 \\ -1 & -1 & 1 & & & 1 \\ \vdots & \vdots & \ddots & \ddots & & \vdots \\ -1 & -1 & \dots & -1 & 1 & 1 \\ -1 & -1 & \dots & -1 & -1 & 1 \end{bmatrix},$$

donde las entradas de la diagonal y la última columna son 1, y las entradas debajo de la diagonal son -1 .

Solución Se adjunta la función en los documentos.

- b) Para $m = 60$, calcule el factor de crecimiento de A ,

$$\rho(A) = \frac{\max |u_{ij}|}{\max |a_{ij}|},$$

y la norma $\|A - LU\|_2$. Comente sobre los resultados. Puede usar el comando `lu`.

Solución: Se calcularon los valores usando **MATLAB** para obtener

$$\rho(A) = 5,76 \times 10^{17}$$

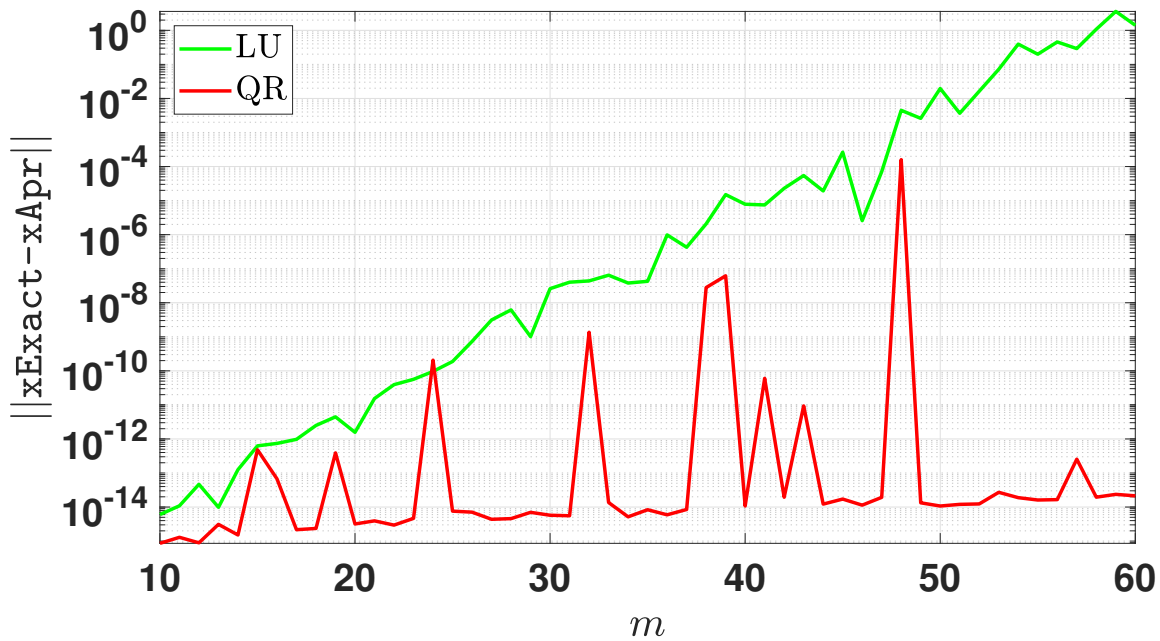
$$\|A - LU\|_2 = 72,1526$$

Se puede observar que el error obtenido en la aproximación LU de A es muy alto. Esto se debe precisamente al alto factor de crecimiento de A , (de hecho es la matriz con mayor factor de crecimiento en $\mathbb{R}^{60 \times 60}$).

- c) Para cada $m=10:60$, defina `xExact = rand(m,1)` y calcule `b=A*xExact`. Resuelva el sistema `A*xApr = b` mediante la factorización LU y la factorización QR (Puede usar los comandos `lu` y `qr`). Obtenga una gráfica de $\|xExact - xApr\|_2$ en función de m . Justifique el comportamiento del gráfico.

Solución: Al implementar la solución en **MATLAB** se obtiene la gráfica 1.

Gráfica 1



Se puede observar una vez más, la dependencia del factor de crecimiento ρ del método LU , puesto que su error crece rápidamente conforme m crece. Mientras tanto, si bien el método QR presenta picos de error, no se tiene un crecimiento real que dependa de m , puesto que se demostró teóricamente que la estabilidad del método no depende del factor de crecimiento de la matriz. **Nota:** Estos picos se mostraron en todas las corridas del script, cabe destacar que las matrices que estamos evaluando son aleatorias

Problema 2

(Runge-Kutta adaptivo) En clase discutimos el método explícito de Runge-Kutta de cuarto orden, dado por el esquema

Esquema 1

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
<hr/>				
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

Para ese caso asumimos que h es constante en cada iteración. A continuación describimos un método adaptivo, en donde en cada paso se determina si el valor h debe ser aumentado o disminuido, según consideraciones locales de error. Deseamos resolver el problema

$$\begin{cases} y' = f(t, y), & t \in [0, T_{\text{máx}}] \\ y(t_0) = y_0 \end{cases}$$

Utilizando el método de Runge-Kutta explícito de orden cuatro, con *step size* variable.

El programa incluye tres parámetros: $h_{\text{máx}}$ (el máximo posible valor de h , por consideraciones de estabilidad), tol (la tolerancia prescrita) y h_0 (valor inicial de h).

En cada iteración k , el error local Δ se estima de la siguiente manera. Primero, se calcula la aproximación y_1 (en el tiempo $t_k + h$) mediante una iteración de Runge-Kutta con incremento h . Luego, se calcula una aproximación y_2 (en el tiempo $t_k + h$) utilizando dos iteraciones de Runge-Kutta con incremento $\frac{h}{2}$ cada una. Se toma así $\Delta = y_2 - y_1$.

El valor de h se modifica de la siguiente manera. Calcule

$$\tilde{h} = \begin{cases} Sh|\frac{tol}{\Delta}|^{0,20}, & si |\Delta| \leq tol \\ Sh|\frac{tol}{\Delta}|^{0,25}, & si |\Delta| > tol \end{cases}$$

donde S es un factor de *seguridad* cercano a 1; en nuestro caso tome $S = 0,9$. Luego, se toma $h_{new} = \min\{h_{m\acute{a}x}, \tilde{h}\}$. Finalmente, si $|\Delta| < tol$, se toma el valor $y_{k+1} = y_2 + \Delta/15$; en caso contrario se repite el ciclo. Note que en cualquier caso se continúa con el valor h_{new} .

- a) Escriba una función que implemente el método adaptivo descrito anteriormente. Las entradas deben ser $f, y_0, t_0, T_{m\acute{a}x}, h_0, h_{m\acute{a}x}, tol$. (Sugerencia: para calcular y_1 y y_2 puede utilizar la función discutida en clase que implementa el esquema 1).

Solución: Se adjuntó la función `RKadaptivo`

- b) Utilice la función anterior para el caso $f(t) = \log(t + 0,01) + 1/(t - 4)$, $y_0 = 0,8$, $t_0 = 0$, $T_{m\acute{a}x} = 3,99$. Tome $h_0 = 10^{-5}$, $h_{m\acute{a}x} = 10^{-2}$, $tol = 10^{-12}$. Dibuje h en función del tiempo (en una escala adecuada). Comente las variaciones observadas en las variaciones de h . Dibuje además el error $|y(t) - \tilde{y}(t)|$ en función de t .

Solución: Primero calculamos la solución general de la ecuación

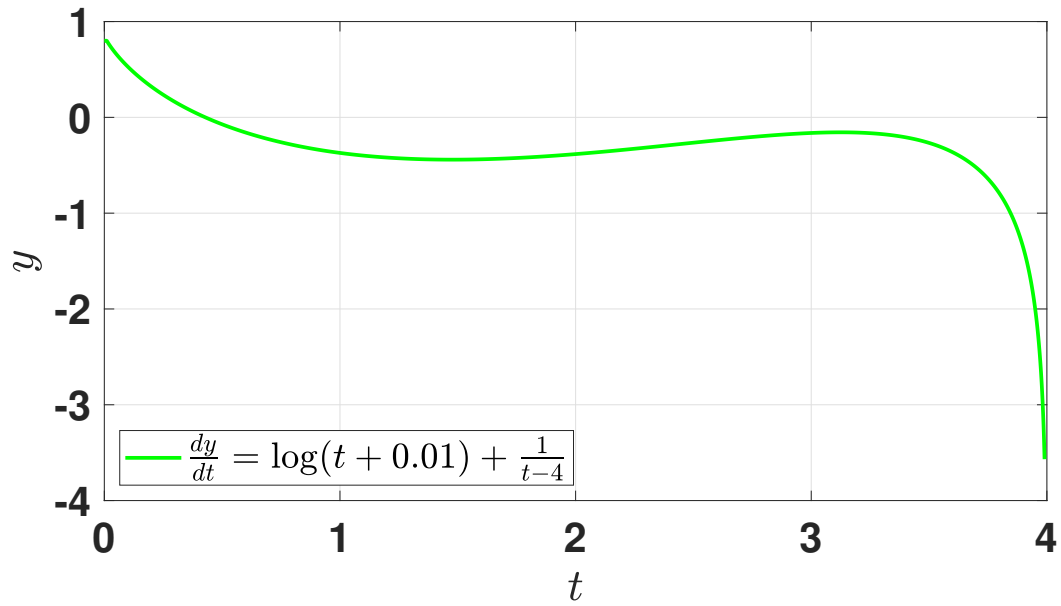
$$\begin{aligned} y' &= \log\left(1 + \frac{1}{100}\right) + \frac{1}{t-4} \\ \Rightarrow y &= \int \log\left(1 + \frac{1}{100}\right) + \frac{1}{t-4} dt \\ \Rightarrow y &= \left(t + \frac{1}{100}\right) \log\left(t + \frac{1}{100}\right) - t - \frac{1}{100} + \log|4-t| + C \end{aligned}$$

Y despejando la condición inicial $y(0) = 0,8$, se tiene que $C \approx -0,5302426593$.

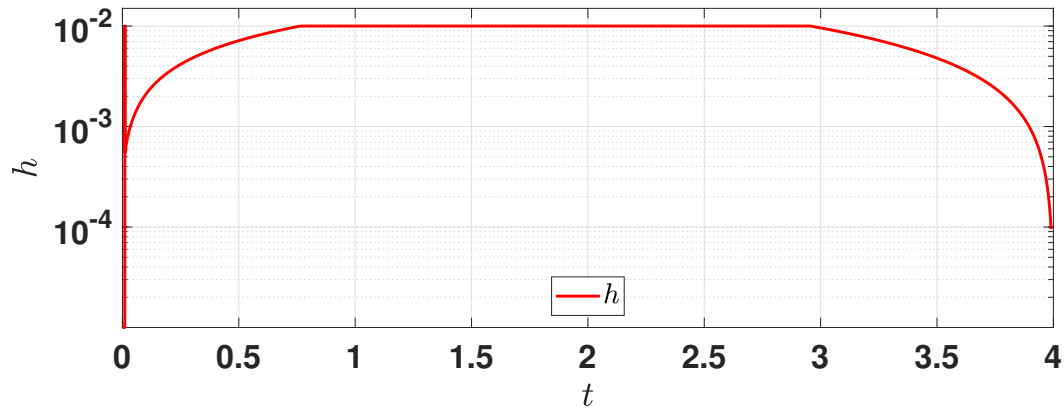
Luego, al aplicar la función del apartado a), se tienen las Gráficas 2,3 y 4. Las cuales corresponden a una gráfica de la solución a la ecuación (obtenida usando el método adaptivo), el tamaño de paso h , y el error entre la solución general y la aproximación, respectivamente.

En la primera gráfica, se puede apreciar que al menos a grandes rasgos, la solución parece acercarse a la esperada (la solución general se puede introducir en cualquier graficador de funciones), lo cual es un buen indicador.

Gráfica 2

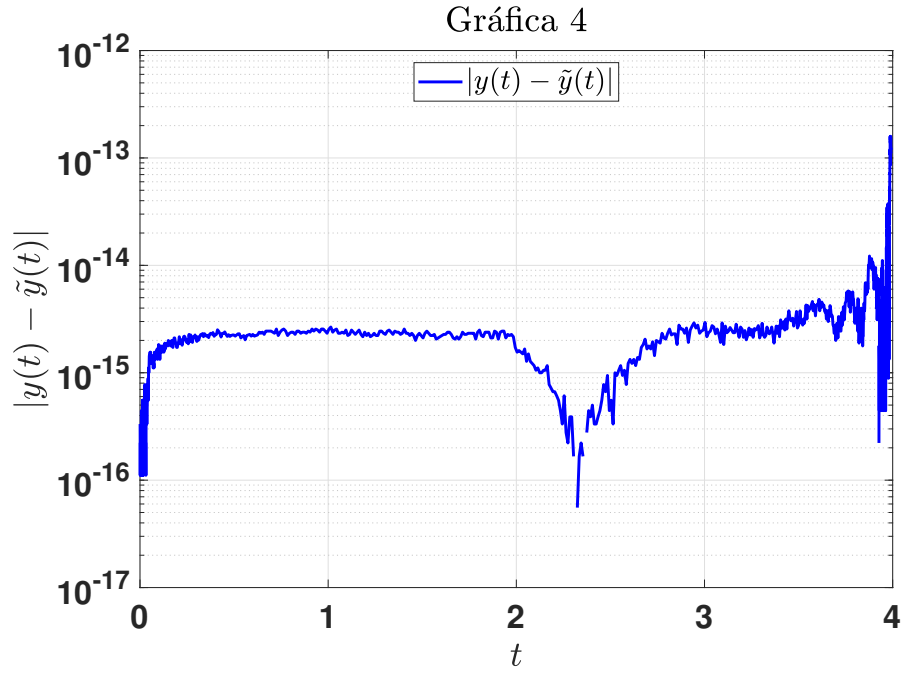


Gráfica 3



Con respecto a la Gráfica 3. Lo primero a notar es un abrupto cambio en la distancia de paso en las primeras iteraciones. Esto nos dice que nuestro paso inicial es demasiado pequeño para el método adaptivo, por lo cual se reajusta rápidamente. Posteriormente dicho valor se estabiliza durante gran parte de las iteraciones. Finalmente, llegando a la singularidad de la función en $t = 4$, se debe volver a ajustar la distancia de paso, debido al comportamiento anómalo de la función cerca de 4.

Finalmente, con respecto a la calidad de la aproximación, en la Gráfica 4 se puede notar que la aproximación es extremadamente precisa, incluso para contar con un número muy pequeño de iteraciones (934). Esto revela el poder de los métodos adaptivos, pues aseguran convergencia, y economía de cómputo.



Problema 3

Considere la ecuación diferencial dada por el sistema

$$\begin{cases} \frac{\partial^2 x}{\partial t^2} = x + 2\frac{\partial y}{\partial t} - u_c \frac{x+u}{((x+u)^2+y^2)^{3/2}} - u \frac{x-u_c}{((x-u_c)^2+y^2)^{3/2}} \\ \frac{\partial^2 y}{\partial t^2} = y - 2\frac{\partial x}{\partial t} - u_c \frac{y}{((x+u)^2+y^2)^{3/2}} - u \frac{y}{((x-u_c)^2+y^2)^{3/2}} \end{cases}$$

donde $x(t), y(t)$ son funciones $x, y : [0, \infty) \rightarrow \mathbb{R}$. Dicho sistema modela el movimiento bidimensional de un satélite, donde $(x(t), y(t))$ representa su ubicación en el tiempo t . En este caso $u = 0,012277471$ y $u_c = 1-u$ son constantes dadas. Considere además las condiciones iniciales de posición y velocidad en $t = 0$ dadas por $x(0) = 0,994, y(0) = 0, x'(0) = 0, y'(0) = -2,001585106$. Sabemos a priori que el movimiento tiene periodo $T = 17,0652166$.

- a) Haga $z = x', w = y'$ y obtenga un sistema de grado uno de ecuaciones diferenciales con cuatro ecuaciones de la forma $\mathbf{u}' = \mathbf{f}(t, \mathbf{u}(t)), \mathbf{u}(0) = \mathbf{u}_0$.

Solución: Llamemos:

$$F(t) = -u_c \frac{x+u}{((x+u)^2+y^2)^{3/2}} - u \frac{x-u_c}{((x-u_c)^2+y^2)^{3/2}}$$

$$G(t) = -u_c \frac{y}{((x+u)^2+y^2)^{3/2}} - u \frac{y}{((x-u_c)^2+y^2)^{3/2}}$$

Entonces haciendo el cambio de variable $z = x', w = y'$ Obtenemos el sistema

$$\begin{cases} x' = z \\ y' = w \\ z' = x + 2w + F(t) \\ w' = y - 2z + G(t) \end{cases}$$

Entonces tome

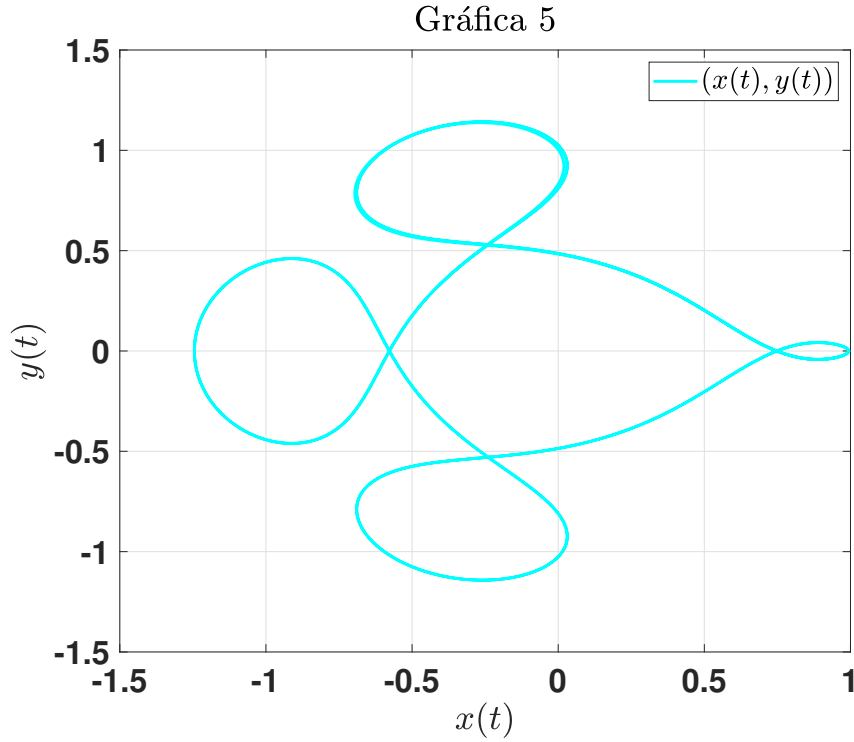
$$\mathbf{u} = (x, y, z, w)$$

$$\mathbf{f}(t, \mathbf{u}(t)) = \begin{pmatrix} z(t) \\ w(t) \\ x(t) + 2w(t) + F(t) \\ y(t) - 2z(t) + G(t) \end{pmatrix}$$

y $\mathbf{u}_0 = (x(0), y(0), x'(0), y'(0))$, para obtener la forma deseada.

- b) Resuelva el sistema obtenido mediante el algoritmo adaptivo de la pregunta 3. Utilice $T_{\text{máx}} = 40, h_0 = 10^{-5}, h_{\text{máx}} = 10^{-2}, \text{tol} = 10^{-6}$. Grafique la curva paramétrica $\{(x(t), y(t)), t \in [0, T_{\text{máx}}]\}$. Compare los valores $(x(0), y(0)), (x(T), y(T))$ y $(x(2T), y(2T))$. Comente sus resultados.

Solución: Al implementar el algoritmo para el problema de aplicación, se obtiene la figura 5, que describe la trayectoria



Se puede observar que la aproximación es bastante buena, numéricamente al evaluar en los periodos se obtuvo que

$X(0) = 0,9940000000000000$	$Y(0) = 0$
$X(T) = 0,993208458224920$	$Y(T) = -0,004486955982549$
$X(2T) = 0,994106192301190$	$Y(2T) = -1,272527371227908 \times 10^{-4}$

Por lo que los resultados son estables conforme avanza t .

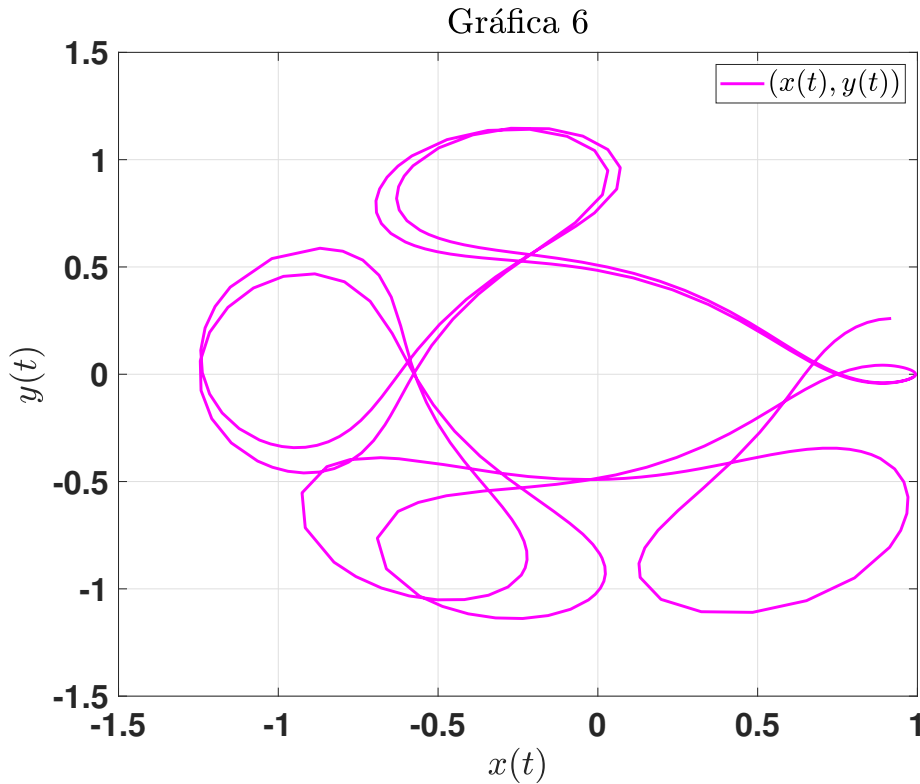
Nota: Esta corrida tomó alrededor de 4100 iteraciones.

c) Resuelva ahora el sistema mediante el comando

$$[t,u] = \text{ode113}(f,[0,40],u0).$$

Grafique la curva paramétrica $\{(x(t),y(t)), t \in [0, T_{\text{máx}}]\}$. ¿Qué observa en este caso?

Solución: Al implementar el algoritmo de MATLAB para el problema de aplicación, se obtiene la figura 6, que describe la trayectoria



Esta vez se tiene

$X(0) = 0,9940000000000000$	$Y(0) = 0$
$X(T) = 0,988300480980110$	$Y(T) = 0,012190987212319$
$X(2T) = 0,654532855410150$	$Y(2T) = -0,349846602949540$

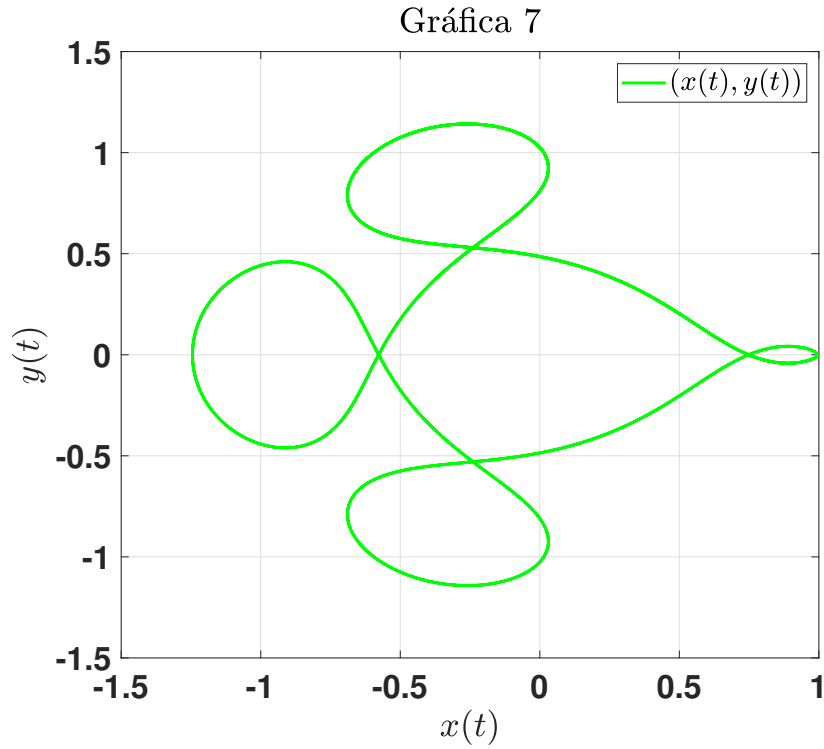
De donde podemos ver que se presenta un problema de estabilidad de la solución, después del segundo periodo. Sin embargo es importante destacar que este método requiere **muchas menos iteraciones** que el anterior, pues alcanzó resultados parecidos (por lo menos al inicio), con cerca del 10% de las iteraciones necesarias en el anterior.

d) Defina el vector de opciones

$$\text{options} = \text{odeset}('RelTol',1e-10,'Stats','on','AbsTol',1e-10).$$

Ejecute ahora el comando $[t,u] = \text{ode113}(f,[0,40],u0,\text{options})$. Grafique la curva paramétrica $\{(x(t),y(t)), t \in [0, T_{\text{máx}}]\}$. ¿Qué observa en este caso?

Solución: En este caso, modificando las opciones del código interno de MATLAB se obtiene la Gráfica 7.



En la cual se puede ver una solución completamente estable, la cual respeta el periodo T que se esperaba. Numéricamente se obtuvo que

$X(0) = 0,994000000000000$	$Y(0) = 0$
$X(T) = 0,993996136853638$	$Y(T) = -2,864376096295169 \times 10^{-7}$
$X(2T) = 0,993996136853638$	$Y(2T) = 4,696898262576229 \times 10^{-5}$

Por lo cual se puede concluir que este método es óptimo para este tipo de problemas, puesto que no necesitó tantas iteraciones (necesitó 1915), como el programado a mano (que aún así ofrecía beneficios de cómputo). Además, con pocas iteraciones, ofreció aproximaciones precisas y estables a nuestro problema de valores iniciales.