Ecuaciones diferenciales ordinarias

Curso de Física Computacional

M. en C. Gustavo Contreras Mayén

9 de abril de 2014

Contenido

Contenido

Contenido

Contenido

Contenido

Contenido

El principal inconveniente del método de series de Taylor es que requiere que se aplique de manera repetida la diferenciación a las variables dependientes.

Evaluar estas expresiones puede llegar a ser una tarea muy extendida y son, por tanto, propensas a errores y tediosas de calcular. Adicionalmente, existe el trabajo extra de codificar cada una de los derivadas.

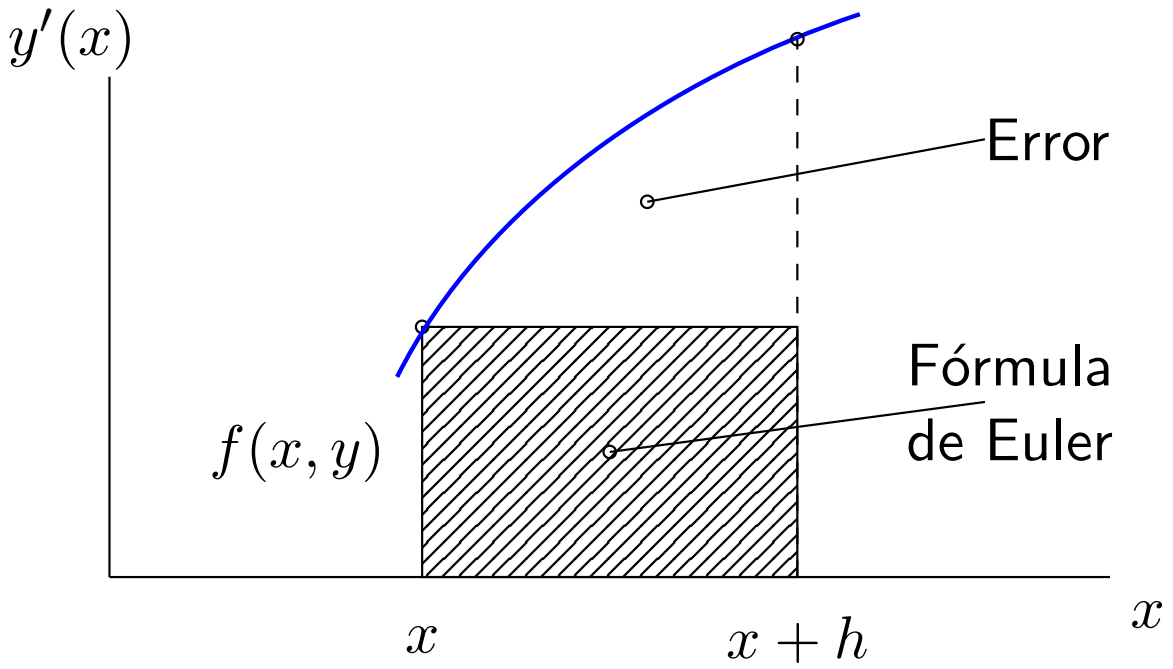
El objetivo de los métodos de Runge-Kutta (RK) es eliminar la necesidad de la diferenciación repetida de las ecuaciones diferenciales.

Dado que la diferenciación repetida no está implicada en la fórmula de integración en serie de Taylor de primer orden

$$\mathbf{y}(x+h) = \mathbf{y}(x) + \mathbf{y}'(x)h = \mathbf{y}(x) + \mathbf{F}(x, \mathbf{y})h$$

Se considera como el método Runge-Kutta de primer orden (RK1), también llamado, *método de Euler*. Como el error de truncamiento es bastante, no se usa en la práctica común.

Representación del método de Euler



Para simplificar, consideremos que tenemos una sola variable y, y que la ecuación diferencial es $y' = f(x, y)$.

El cambio en la solución y entre x y $x+h$ es

$$y(x+h) - y(h) = \int_x^{x+h} y' dx = \int_x^{x+h} f(x,y) dx$$

que es el área debajo de la gráfica de $y'(x)$.

La fórmula de Euler aproxima ésta área con el área del rectángulo sombreado. El área entre el rectángulo sombreado y la gráfica representa el error debido al truncamiento.

Se revisa claramente que el error de truncamiento es proporcional a la pendiente de la gráfica, esto es, proporcional a $y'(x)$

Contenido

Método de Runge-Kutta de segundo orden

Para obtener el método RK2, suponemos una fórmula de integración del tipo

$$\mathbf{y}(x+h) = \mathbf{y}(x) + c_0\mathbf{F}(x,\mathbf{y})h + c_1\mathbf{F}[x+ph, \mathbf{y}+qh\mathbf{F}(x,\mathbf{y})]h$$

e intentamos encontrar los parámetros c_0, c_1, p y q de tal forma que se parezca a la serie de Taylor

$$\begin{aligned}\mathbf{y}(x+h) &= \mathbf{y}(x) + \mathbf{y}'(x)h + \frac{1}{2!}\mathbf{y}''(x)h^2 + O(h^3) \\ &= \mathbf{y}(x) + \mathbf{F}(x,\mathbf{y})h + \frac{1}{2}\mathbf{F}'(x,\mathbf{y})h^2 + O(h^3)\end{aligned}$$

Notemos que

$$\mathbf{F}'(x,\mathbf{y}) = \frac{\partial \mathbf{F}}{\partial x} + \sum_{i=0}^{n-1} \frac{\partial \mathbf{F}}{\partial \mathbf{y}_i}\mathbf{y}_i' = \frac{\partial \mathbf{F}}{\partial x} + \sum_{i=0}^{n-1} \frac{\partial \mathbf{F}}{\partial \mathbf{y}_i}F_i(x,\mathbf{y})$$

donde n es el número de 1-EDO.

Entonces podemos escribir para $\mathbf{y}(x+h)$ como

$$\begin{aligned}\mathbf{y}(x+h) &= \mathbf{y}(x) + \mathbf{F}(x,\mathbf{y})h + \\ &= \frac{1}{2}\left(\frac{\partial \mathbf{F}}{\partial x} + \sum_{i=0}^{n-1} \frac{\partial \mathbf{F}}{\partial \mathbf{y}_i} F_i(x,\mathbf{y})\right) h^2 + O(h^3)\end{aligned}$$

Regresando a la ecuación inicial, re-escribimos el último término mediante una serie de Taylor en varias variables:

$$\mathbf{F}[x+ph, \mathbf{y}+qh\mathbf{F}(x,\mathbf{y})] = \mathbf{F}(x,\mathbf{y}) + \frac{\partial \mathbf{F}}{\partial x}ph + \\ + qh\sum_{i=1}^{n-1}\frac{\partial \mathbf{F}}{\partial y_i}F_i(x,\mathbf{y}) + O(h^2)$$

Por lo que la ecuación inicial, toma la forma:

$$\mathbf{y}(x+h) = \mathbf{y}(x) + (c_0 + c_1)\mathbf{F}(x,\mathbf{y})h + \\ + c_1 \left[\frac{\partial \mathbf{F}}{\partial x} ph + qh \sum_{i=1}^{n-1} \frac{\partial \mathbf{F}}{\partial y_i} F_i(x,\mathbf{y}) \right] h + O(h^3)$$

Para que las expresiones sean idénticas, se necesita que:

$$c_0 + c_1 = 1 \qquad c_1 p = \frac{1}{2} \qquad c_1 q = \frac{1}{2}$$

El conjunto anterior representa un sistema de tres ecuaciones y cuatro incógnitas, por lo que se asigna un valor a cualquiera de ellas.

Las opciones más comunes y sus nombres para los métodos son los siguientes:

$c_0 = 0$	$c_1 = 1$	$p = \frac{1}{2}$	$q = \frac{1}{2}$	Euler modificado
$c_0 = \frac{1}{2}$	$c_1 = \frac{1}{2}$	$p = 1$	$q = 1$	Heun
$c_0 = \frac{1}{3}$	$c_1 = \frac{2}{3}$	$p = \frac{3}{4}$	$q = \frac{3}{4}$	Ralston

Todas estas fórmulas son del tipo RK2, ninguna tiene una superioridad numérica con respecto a las otras.

Método de Euler modificado

Sustituimos los valores de los parámetros en la ecuación general para obtener:

$$\mathbf{y}(x+h) = \mathbf{y}(x) + \mathbf{F}\left[x + \frac{h}{2}, \mathbf{y} + \frac{h}{2}\mathbf{F}(x, \mathbf{y})\right] h$$

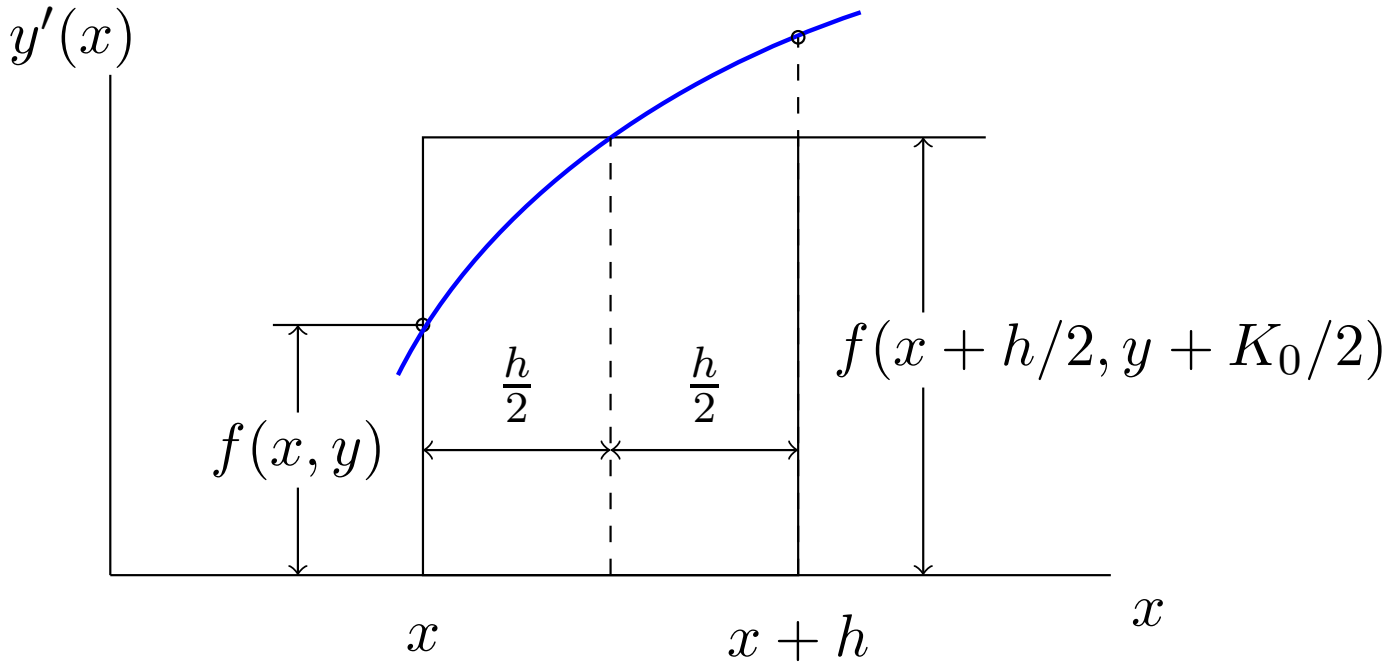
Esta fórmula de integración puede evaluarse convenientemente, siguiendo la siguiente secuencia de operaciones:

$$\mathbf{K}_0 = h\mathbf{F}(x, \mathbf{y})$$

$$\mathbf{K}_1 = h\mathbf{F}\left(x + \frac{h}{2}, \mathbf{y} + \frac{1}{2}\mathbf{K}_0\right)$$

$$\mathbf{y}(x+h) = \mathbf{y}(x) + \mathbf{K}_1$$

Representación del método de Euler modificado



Representación gráfica del método de Euler modificado para una EDO $y' = f(x, y)$.

El valor de $\mathbf{K}_0 = h\mathbf{F}(x, \mathbf{y})$ devuelve un estimado de y en el punto central para la fórmula de Euler

$$y(x + \frac{h}{2}) = y(x) + f(x, y)\frac{h}{2} = y(x) + \frac{K_0}{2}$$

La segunda ecuación $\mathbf{K}_1$, aproxima el área del bloque por el área K_1 del rectángulo. El error es proporcional a la curvatura y''' de la gráfica.

Ejemplo

Utiliza RK2 para integrar la siguiente EDO:

$$y' = \sin y \qquad\qquad y(0) = 1$$

de $x = 0$ a $x = 0.5$ en pasos de $h = 0.1$

Solución

Del problema tenemos que:

$$F(x, y) = \sin y$$

por lo que las fórmular canónicas de integración son

$$K_0 = hF(x, y) = 0.1 \sin y$$

$$K_1 = hF\left(x + \frac{h}{2}, y + \frac{1}{2}K_0\right) = 0.1 \sin\left(y + \frac{1}{2}K_0\right)$$

$$y(x + h) = y(x) + K_1$$

Como $y(0) = 1$, podemos integrar

$$K_0 = 0.1 \sin(1.0000) = 0.0841$$

$$K_1 = 0.1 \sin\left(1.0000 + \frac{0.0841}{2}\right) = 0.0863$$

$$y(0.1) = 1.0 + 0.0863 = 1.0863$$

Como $y(0) = 1$, podemos integrar

$$K_0 = 0.1 \sin(1.0000) = 0.0841$$

$$K_1 = 0.1 \sin\left(1.0000 + \frac{0.0841}{2}\right) = 0.0863$$

$$y(0.1) = 1.0 + 0.0863 = 1.0863$$

$$K_0 = 0.1 \sin(1.0863) = 0.0885$$

$$K_1 = 0.1 \sin\left(1.0.0863 + \frac{0.0885}{2}\right) = 0.0905$$

$$y(0.2) = 1.0863 + 0.0905 = 1.1768$$

y así, sucesivamente.

A manera de resumen, las cuentas se presentan en la siguiente tabla:

x	y	K_0	K_1
0.0	1.0000	0.0841	0.0863
0.1	1.0863	0.0885	0.0905
0.2	1.1768	0.0923	0.0940
0.3	1.2708	0.0955	0.0968
0.4	1.3676	0.0979	0.0988
0.5	1.4664		

La solución exacta (que podrían demostrar que cumple) es:

$$x(y) = ln(\csc y - \cot y) + 0.604582$$

que devuelve en $x(1.4664) = 0.5000)$

Sin embargo, es poco probable que esta precisión se mantenga mientras continuemos integrando, dado que los errores (tanto de truncamiento como de redondeo) se van acumulando, si tenemos un rango amplio de integración, se requiere de mejores fórmulas de integración.

Contenido

Método de Runge-Kutta de cuarto orden

El método de RK4 se obtiene de la serie de Taylor, de la misma forma como se obtuvo RK2; considerando que la derivación es un proceso largo y no tan instructivo, por lo que lo omitiremos.

La expresión final de la fórmula de integración también depende de la elección de los parámetros, es decir, no hay una única fórmula para RK4.

La expresión más popular se le conoce como método RK4, que requiere de las siguientes operaciones:

$$K_0 = h\mathbf{F}(x, \mathbf{y})$$

$$K_1 = h\mathbf{F}(x + \frac{h}{2}, \mathbf{y} + \frac{\mathbf{K}_0}{2})$$

$$K_2 = h\mathbf{F}(x + \frac{h}{2}, \mathbf{y} + \frac{\mathbf{K}_1}{2})$$

$$K_3 = h\mathbf{F}(x + h, \mathbf{y} + \mathbf{K}_2)$$

$$\mathbf{y}(x + h) = \mathbf{y}(x) + \frac{1}{6}(\mathbf{K}_0 + 2\mathbf{K}_1 + 2\mathbf{K}_2 + \mathbf{K}_3)$$

El principal inconveniente de este método es que no se presta para una estimación del error de truncamiento. Por lo tanto, tenemos que "adivinar" el tamaño del paso de integración h, o determinarlo por ensayo y error.

En contraste, los llamados métodos adaptativos pueden evaluar el error de truncamiento en cada paso de integración y ajustar el valor de h en consecuencia, pero con un gran costo, computacionalmente hablando.

Algoritmo RK4

La función `integra` en este módulo, implementa el método RK4. El usuario deberá de proporcionar en `integra` la función `F(x,y)` que define el conjunto de 1-EDO $\mathbf{y}' = \mathbf{F}(x, \mathbf{y})$.

```
1  def integra(F,x,y,xAlto,h):
2
3      def rk_4(F,x,y,h):
4          K0 = h*F(x,y)
5          K1 = h*F(x + h/2.0, y + K0/2.0)
6          K2 = h*F(x + h/2.0, y + K1/2.0)
7          K3 = h*F(x + h, y + K2)
8          return (K0 + 2.0*K1 + 2.0*K2 + K3) /
               6.0
9
10     X = []
11     Y = []
12     X.append(x)
13     Y.append(y)
```

```
1        while x < xAlto:
2            h = min(h, xAlto - x)
3            y = y + rk_4(F,x,y,h)
4            x = x + h
5            X.append(x)
6            Y.append(y)
7
8        return array(X), array(Y)
```

Ejemplo

Resolver

$$y'' = -0.1y' - x \qquad y(0) = 0 \qquad y'(0) = 1$$

de $x = 0$ a $x = 2$ con incrementos de $h = 0.25$ mediante RK4.

Usando la notación $y_0 = y$ junto con $y_1 = y'$, podemos escribir un conjunto de 1-EDO como

$$\mathbf{y}' = \mathbf{F}(x, \mathbf{y}) = \begin{bmatrix} y_0' \\ y_1' \end{bmatrix} = \begin{bmatrix} y_1 \\ -0.1y_1 - x \end{bmatrix}$$

Código completo

```
1  def F(x,y):
2      F = zeros((2), dtype='float64')
3      F[0] = y[1]
4      F[1] = -0.1 * y[1]
5      return F
6
7  x=0.0
8  xAlto=2.0
9  y = array([0.0,1.0])
10 h=0.25
11 freq=1
12
13 X,Y = integra(F,x,y,xAlto,h)
14 imprimeSoln(X,Y,freq)
```

Resultado

x	y[0]	y[1]
0.0000e+00	0.0000e+00	1.0000e+00
2.5000e-01	2.4690e-01	9.7531e-01
5.0000e-01	4.8771e-01	9.5123e-01
7.5000e-01	7.2257e-01	9.2774e-01
1.0000e+00	9.5163e-01	9.0484e-01
1.2500e+00	1.1750e+00	8.8250e-01
1.5000e+00	1.3929e+00	8.6071e-01
1.7500e+00	1.6054e+00	8.3946e-01
2.0000e+00	1.8127e+00	8.1873e-01

Ejercicio 2

Usa RK4 para integrar

$$y' = 3y - 4e^{-x} \qquad y(0) = 1$$

desde $x = 0$ hasta $x = 10$ en pasos de $h = 0.1$. Comparar el resultado con la solución analítica $y = \exp(-x)$

Usaremos el programa anterior. Recordemos que `rk_4` supone que y es un arreglo, por lo que debemos de especificar la condición inicial como $y = \text{array}([1.0])$ y no $y = 1.0$

```
1  def F(x,y):
2      F = zeros((1), dtype='float64')
3      F[0] = 3.0 * y[0] - 4.0 * exp(-x)
4      return F
5
6
7  x = 0.0
8  xAlto = 10.0
9  y = array([1.0])
10 h = 0.1
11 freq = 20
12
13 X,Y = integra(F,x,y,xAlto,h)
14 imprimeSoln(X,Y,freq)
```

Resultado

x	y[0]
0.0000e+00	1.0000e+00
2.0000e+00	1.3250e-01
4.0000e+00	-1.1237e+00
6.0000e+00	-4.6056e+02
8.0000e+00	-1.8575e+05
1.0000e+01	-7.4912e+07

Pero ¿es correcto esto?

De acuerdo a la solución numérica, y debería de acercarse a cero conforme se incrementa x, pero el resultado nos muestra lo contrario: después de un incremento inicial, la magnitud de y se incrementa súbitamente.

Podemos explicar esto mirando de cerca la solución analítica.

La solución general de la EDO dada es

$$y = Ce^{3x} + e^{-x}$$

que puede verificarse por sustitución.

La condición inicial $y(0) = 1$ hace que $C = 0$, que para la solución del problema, es precisamente $y = exp(-x)$.

El problema en la solución numérica es el término dominante Ce^{3x}

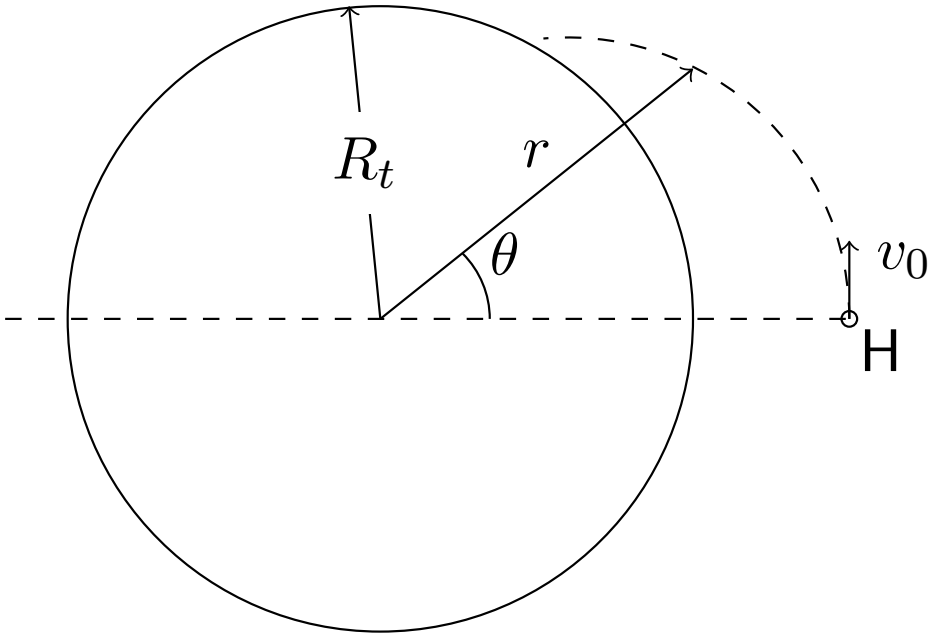
Supongamos que la condición inicial tiene un pequeño error ϵ, por lo que tenemos $y(0) = 1 + \epsilon$. Esto cambia la solución analítica por

$$y = \epsilon e^{3x} + e^{-x}$$

$$y = \epsilon e^{3x} + e^{-x}$$

Vemos que el término que contiene el error ϵ, se hace dominante conforme x aumenta. Dado que los errores son inherentes a las soluciones numéricas, tenemos el mismo efecto para cambios pequeños en las condiciones iniciales, por lo que concluimos que nuestra solución numérica es víctima de la *inestabilidad numérica*, debida a la sensibilidad de la solución a las condiciones iniciales.

Ejercicio más elaborado



Un satélite se lanza desde una altitud $H = 772$ km sobre el nivel del mar, con una velocidad inicial $v_0 = 6700\ m/s$ en la dirección que se muestra.

El conjunto de EDO que describen el movimiento del satélite son:

$$\ddot{r} = r\dot{\theta}^2 - \frac{GM_t}{r^2} \qquad \ddot{\theta} = -\frac{2\dot{r}\dot{\theta}}{r}$$

donde r y θ son las coordenadas polares del satélite.

Las constantes involucradas en las expresiones, son:

$G = 6.672 \times 10^{-11}\text{m}^3\text{kg}^{-1}\text{s}^2$

$M_t = 5.9742 \times 10^{24}$kg, Masa de la Tierra

$R_e = 6378.14$km, radio de la Tierra al nivel del mar

Problema a resolver

1. Obtén el conjunto de 1-EDO y las condiciones iniciales del problema, de la forma $\dot{\mathbf{y}} = \mathbf{F}(x, \mathbf{y})$, $\mathbf{y}(0) = \mathbf{b}$.
2. Con RK4 integra las 1-EDO en el tiempo en que se lanza el satélite y choca en su regreso a la Tierra.
3. Calcula el lugar del impacto, con θ.

Inciso 1)

Tenemos que

$$GM_t = (6.672 \times 10^{-11})(5.9742 \times 10^{24}) = \\ = 3.9860 \times 10^{14} \mathrm{m}^3\mathrm{s}^2$$

Ahora hacemos

$$\mathbf{y} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} r \\ \dot{r} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

Por lo que el conjunto equivalente de 1-EDO, resulta ser

$$\dot{\mathbf{y}} = \begin{bmatrix} \dot{y}_0 \\ \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_0 y_3^2 - 3.9860 \times 10^{14}/y_0^2 \\ y_3 \\ -2y_1 y_3/y_0 \end{bmatrix}$$

Las condiciones iniciales resultan

$$r(0) = R_t + H = (6378.14 + 772) \times 10^3 = 7.15014 \times 10^6 \text{ m}$$
$$\dot{r}(0) = 0$$
$$\theta(0) = 0$$
$$\dot{\theta}(0) = \frac{v_0}{r(0)} = \frac{6700}{7.15014 \times 10^6} = 0.937045 \times 10^{-3} \text{ rad/s}$$

Por tanto

$$\mathbf{y}(0) = \begin{bmatrix} 7.15014 \times 10^6 \\ 0 \\ 0 \\ 0.937045 \times 10^{-3} \end{bmatrix}$$

Inciso 2)

Usaremos lo que ya hemos construido con el programa `integra`, para continuar con la notación, la variable independiente t, la escribimos como x.

El período de integración, es decir, el valor de `xAlto` cuando el satélite choca con la Tierra, hay que estimarlo tentativamente.

```
1  def F(x,y):
2      F = zeros((4), dtype='float64')
3      F[0] = y[1]
4      F[1] = y[0]*(y[3]**2) - 3.9860e14 / (y
           [0]**2)
5      F[2] = y[3]
6      F[3] = -2.0*y[1]*y[3]/y[0]
7      return F
8
9  x = 0.0
10 xAlto =
11 y = array([7.1514e6, 0.0 , 0.0, 0.937045e-3])
12 h = 50.0
13 freq = 2
14 X,Y = integra(F,x,y,xAlto,h)
15 imprimeSoln(X,Y,freq)
```

Resultados

x	y[0]	y[1]	y[2]	y[3]
0.0000e+00	7.1514e+06	0.0000e+00	0.0000e+00	9.3704e-04
1.0000e+02	7.1438e+06	-1.5134e+02	9.3771e-02	9.3903e-04
2.0000e+02	7.1212e+06	-3.0199e+02	1.8794e-01	9.4502e-04
3.0000e+02	7.0835e+06	-4.5121e+02	2.8291e-01	9.5510e-04
4.0000e+02	7.0310e+06	-5.9819e+02	3.7910e-01	9.6942e-04
5.0000e+02	6.9639e+06	-7.4201e+02	4.7694e-01	9.8817e-04
6.0000e+02	6.8827e+06	-8.8159e+02	5.7689e-01	1.0116e-03
7.0000e+02	6.7878e+06	-1.0156e+03	6.7944e-01	1.0401e-03
8.0000e+02	6.6798e+06	-1.1426e+03	7.8510e-01	1.0740e-03
9.0000e+02	6.5596e+06	-1.2605e+03	8.9443e-01	1.1138e-03
1.0000e+03	6.4281e+06	-1.3671e+03	1.0081e+00	1.1598e-03
1.1000e+03	6.2866e+06	-1.4595e+03	1.1266e+00	1.2126e-03
1.2000e+03	6.1368e+06	-1.5343e+03	1.2508e+00	1.2725e-03

El satélite choca con la Tierra cuando r es igual a $R_t = 6.37814 \times 10^6$ m. De los resultados, vemos que esto ocurre entre el tiempo $t = 1000$ y 1100 segundos.

Un valor de t más preciso, lo podemos obtener mediante una interpolación polinomial, pero si no queremos una precisión alta, con una interpolación lineal, bastará.

Hacemos $1000 + \Delta t$ el tiempo para el impacto, por lo que escribimos

$$r(1000 + \Delta t) = R_t$$

Desarrollando r con dos términos de la serie de Taylor, tenemos que

$$r(1000) + \dot{r}(1000)\Delta t = R_t$$

$$6.4250 \times 10^6 + (-1.3708 \times 10^3)\Delta t = 6378.14 \times 10^3$$

que al despejar,

$$\Delta t = 34.184 \text{ s}$$

Por lo que el tiempo de impacto es 1034.25 segundos.

Inciso 3)

La coordenada θ del impacto, la podemos calcular de una manera similar, para ello desarrollamos dos términos de la serie de Taylor:

$$\begin{aligned}\theta(1000+\Delta t) &= \theta(1000)+\dot{\theta}(1000)\Delta t \\ &= 1.0083+(1.1605\times 10^{3})(34.184) \\ &= 1.0489 \text{ rad} = 60.00^{\circ}\end{aligned}$$

Contenido

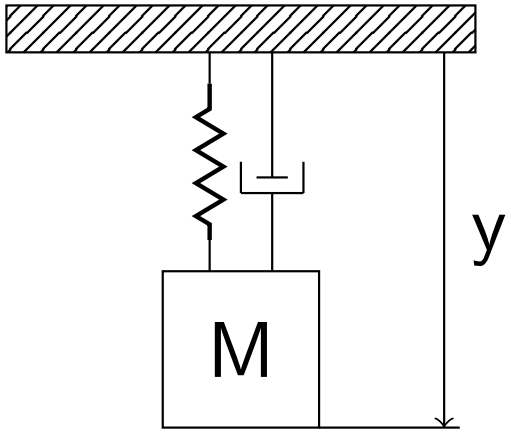
Ejercicio

Una masa $M = 0.5$ kg se une al extremo inferior de un resorte sin masa. El extremo superior se fija a una pared en reposo. La masa experimenta una resistencia $R = -Bdy/dt$ debida al aire, donde B es una constante de amortiguamiento. La ecuación de movimiento es:

$$M\frac{d^2}{dt^2}y + B\frac{d}{dt}y + ky = 0 \qquad y(0) = 1, y'(0) = 0$$

donde $k = 100\ k/s^2$ y $B = 10\ k/s$

Sistema masa-resorte

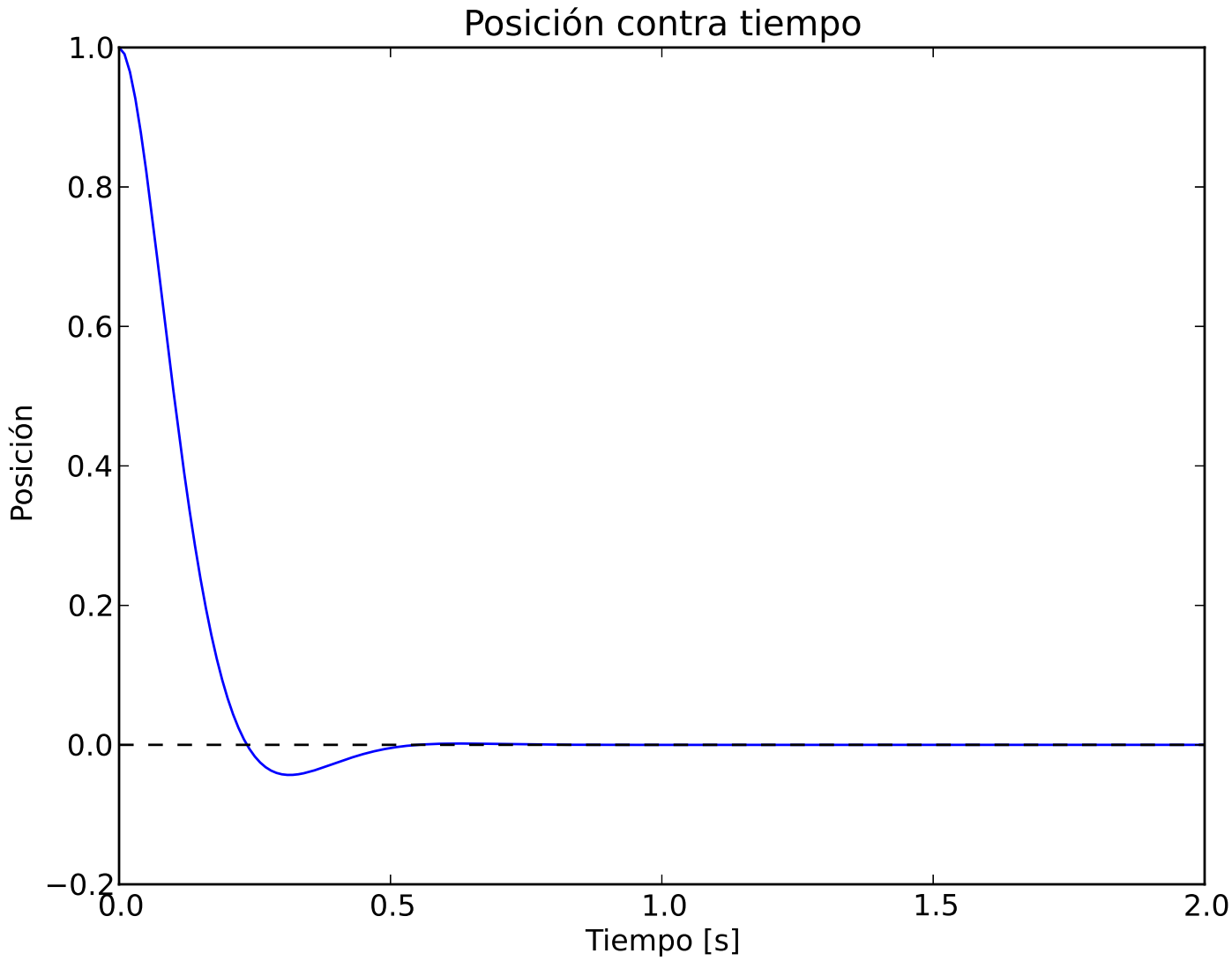


$$M\frac{d^2}{dt^2}y + B\frac{d}{dt}y + ky = 0 \qquad y(0) = 1, y'(0) = 0$$

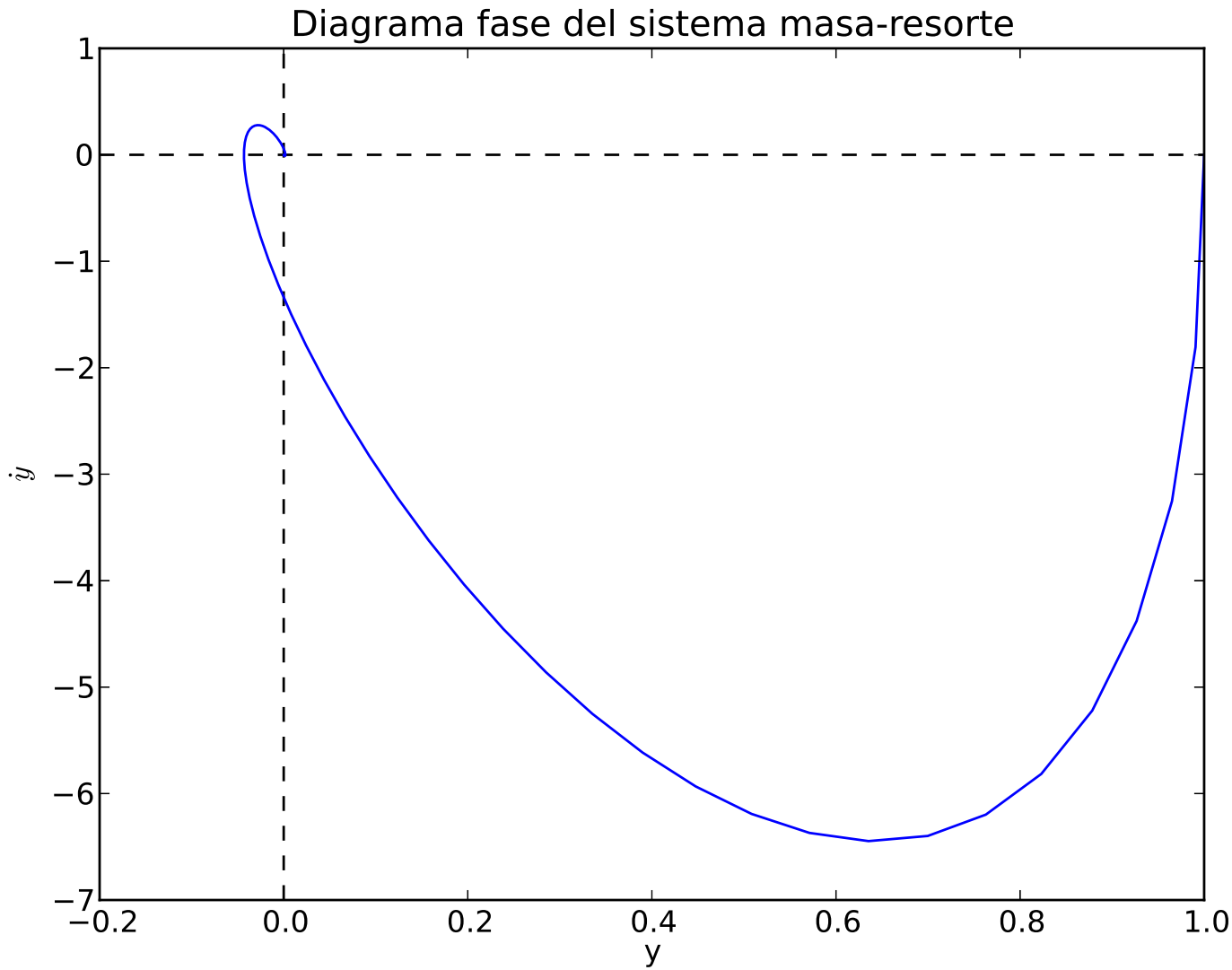
donde $k = 100\ k/s^2$ y $B = 10\ k/s$.

Calcula $y(t)$ para $0 < t < 2$, con $h = 0.001$, ¿qué pasa si $B = 0$?

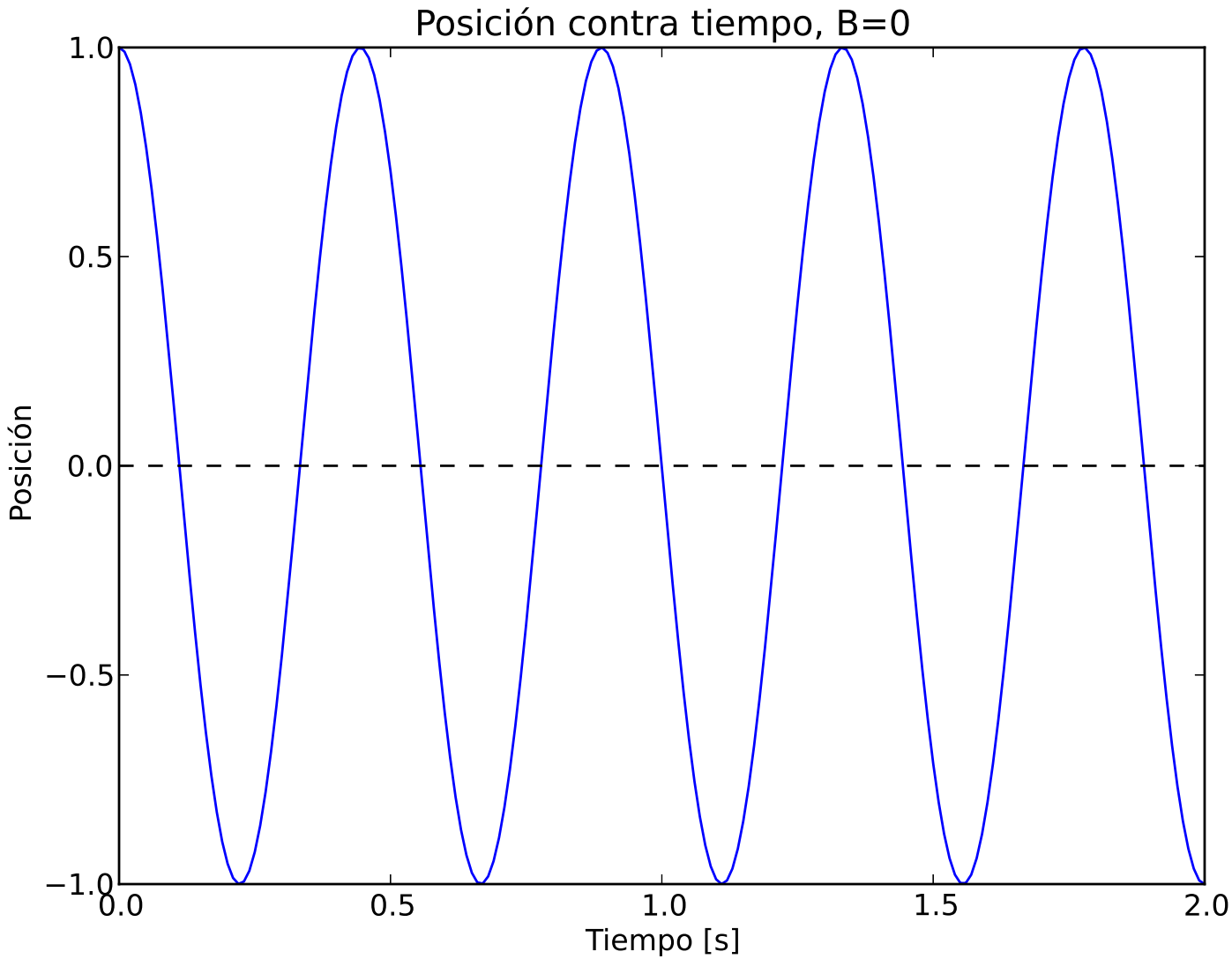
Solución gráfica



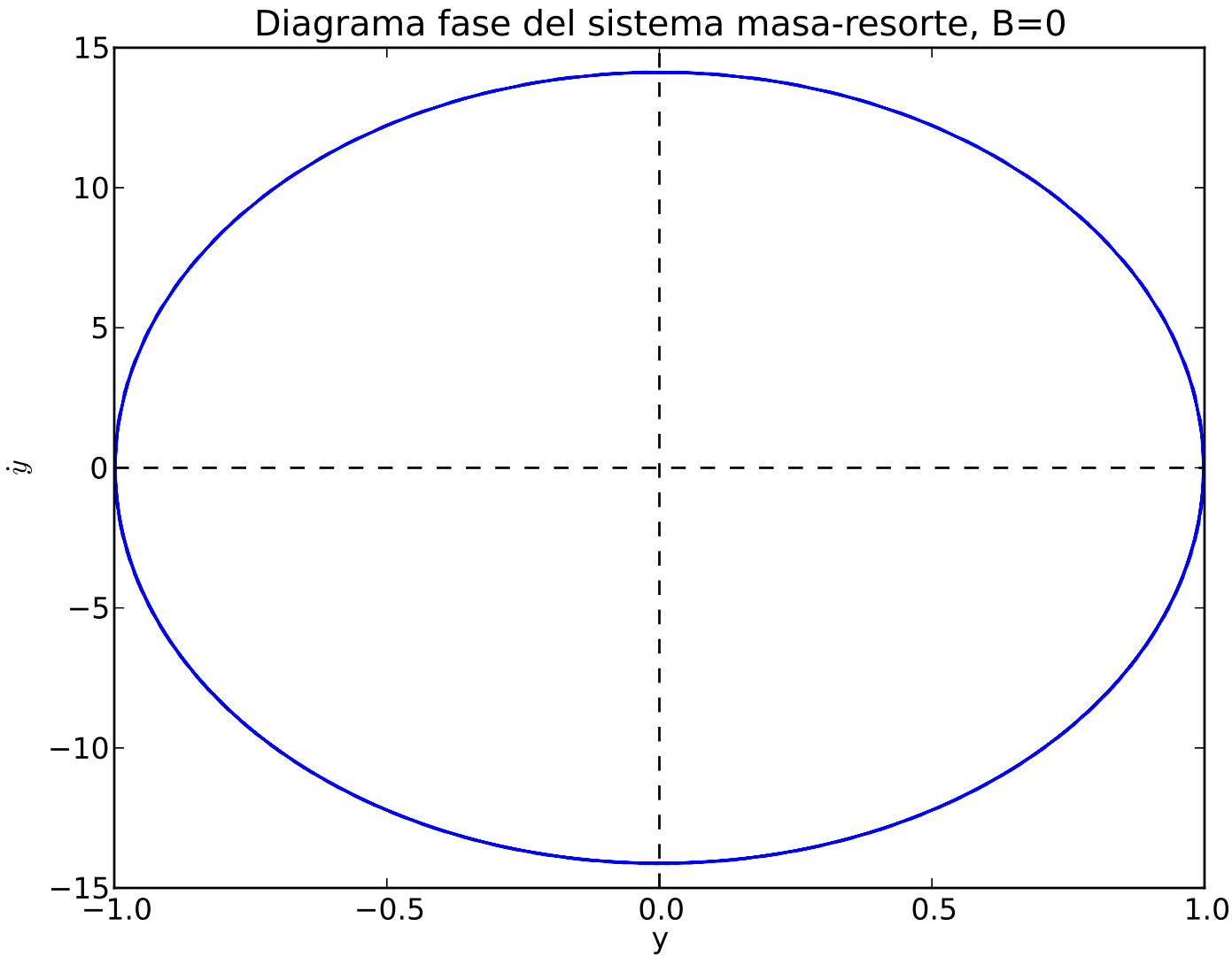
Solución gráfica



Solución gráfica



Solución gráfica



Ejercicio

Una pieza metálica con una masa de 0.1 kg a $200°$ C (473 K), se coloca en cierto momento en un cuarto cuya temperatura es $25°$ C, en donde la pieza está sujeta al proceso de enfriamiento por convección natural y transferencia de calor por radiación.

Suponemos que la distribución de temperatura es uniforme en la pieza, la ecuación de T vs t es:

$$\frac{dT}{dt} = \frac{A}{(\rho c v)}[\epsilon\sigma(297^4 - T^4) + h_c(297 - T)]$$

con $T(0) = 473$ donde T es la temperatura en grados Kelvin y las constantes son:

$\rho =$ 300 $\frac{k}{m^3}$ – densidad del metal

$v =$ 0.001 m^3 – volumen del metal

$A =$ 0.25 m^2 – área de la superficie del metal

$c =$ 900 $\frac{J}{kK}$ – calor específico del metal

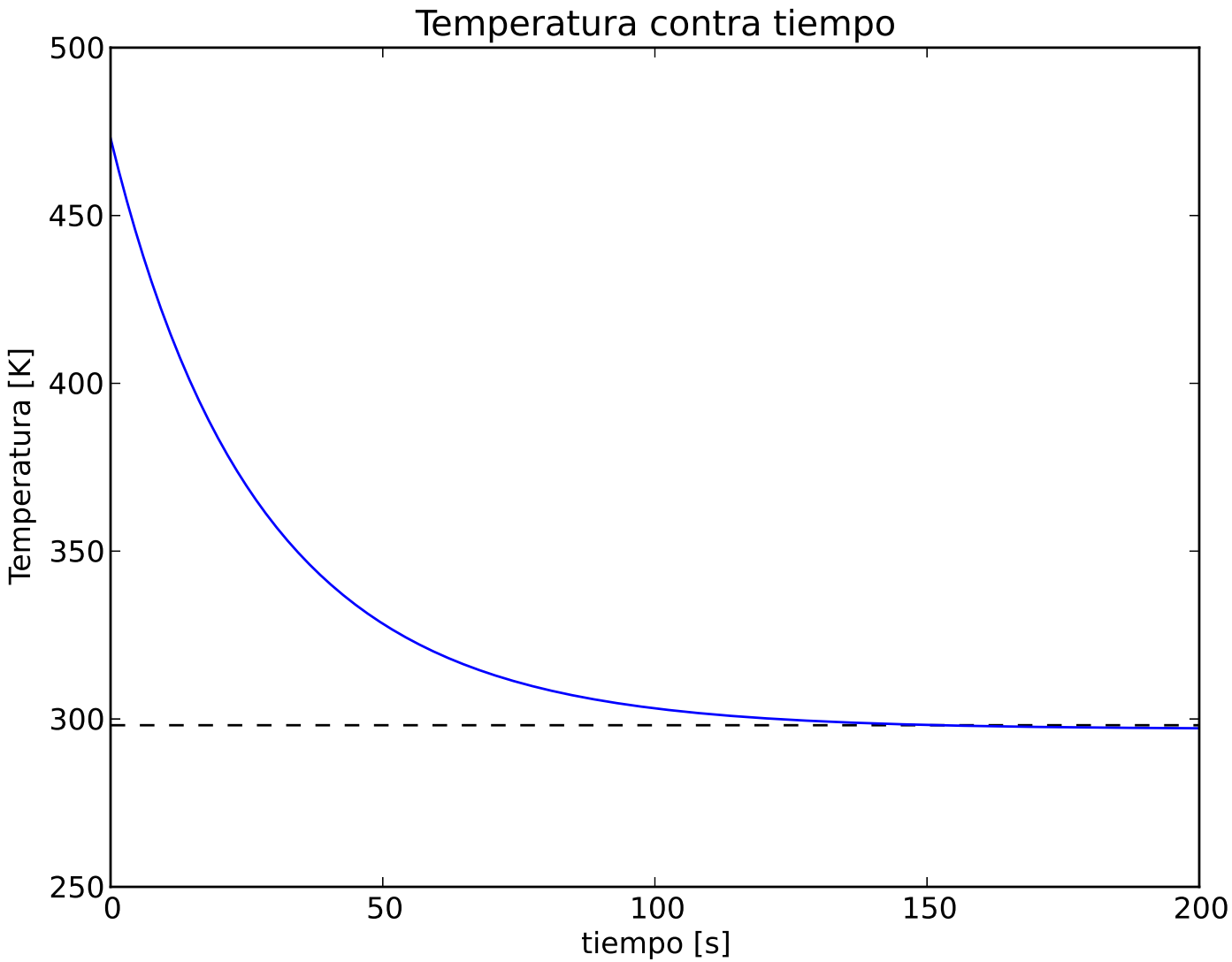
$h_c =$ 30 $\frac{J}{m^2K}$ coeficiente de transferencia de calor

$\epsilon =$ 0.8 – emisividad del metal

$\sigma =$ 5.67 $\times 10^{-8}$ $\frac{w}{m^2K^2}$ cte Stefan-Boltzamann

Calcular el valor de T para el intervalo $0 < t < 200$ segundos, usa $h = 0.25$

Solución gráfica



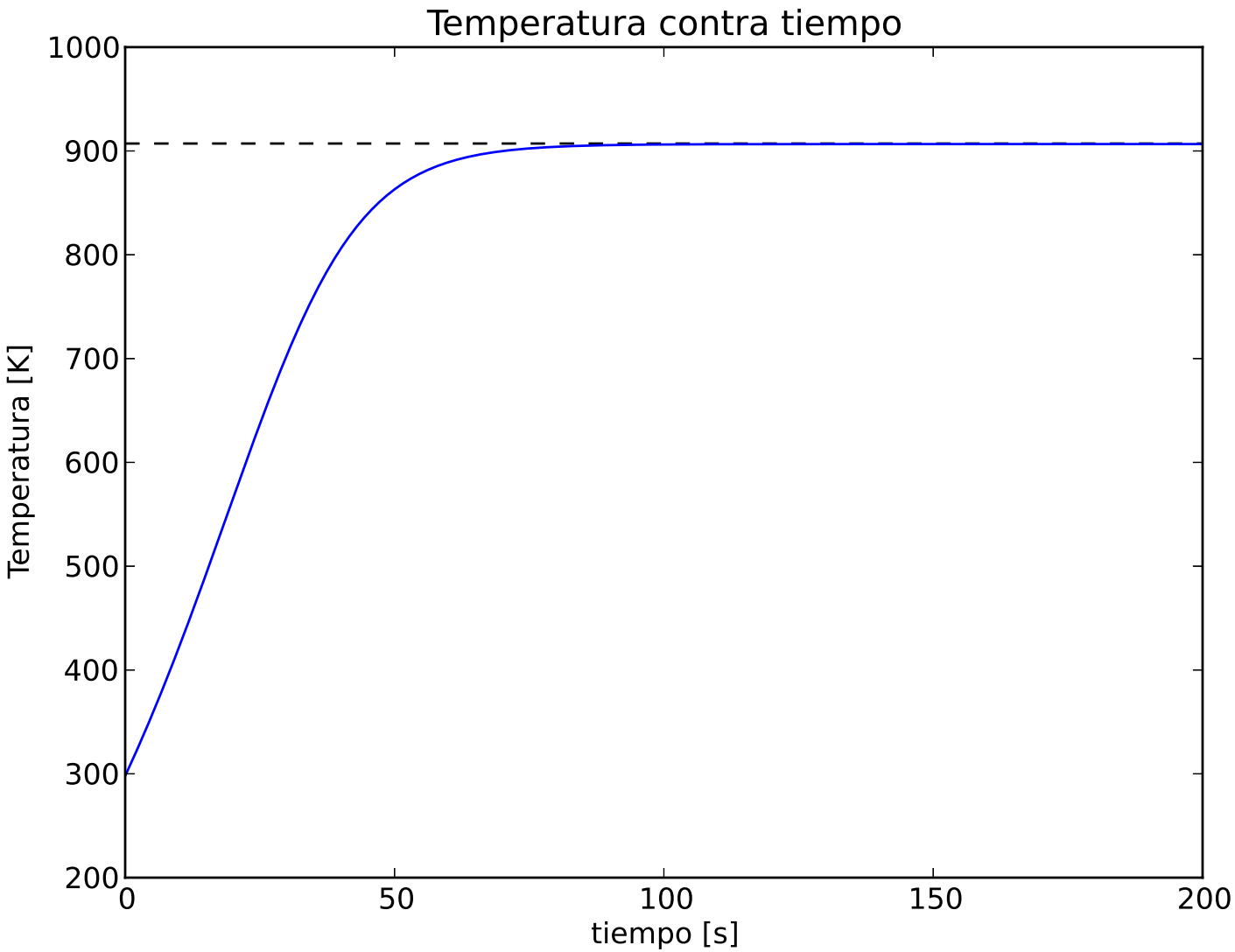
Ejercicio adicional

Con la misma pieza metálica del ejercicio anterior, ahora consideremos que la T inicial es de 25° y se calienta internamente de forma eléctrica a razón de $q = 3000$ W. La ecuación de temperatura es:

$$\begin{aligned}\frac{dT}{dt} &= \frac{1}{\rho c v}\left[q - \epsilon\sigma A\left(T^4 - 298^4\right) - h_c A\left(T - 298\right)\right], \\ T(0) &= 298\end{aligned}$$

Calcular la temperatura de la pieza hasta $t =$ minutos, usando RK4, con $h = 0.1$ minutos.

Solución gráfica



Contenido

Usando Python para resolver las EDO

Un sistema de EDOs es usualmente formulado en forma estándar antes de ser resuelto numéricamente con Python. La forma estándar es:

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

donde

$$y = [y_1(t), y_2(t), \ldots, y_n(t)]$$

y f es una función que determina las derivadas de la función $y_i(t)$.

Para resolver la EDO necesitamos conocer la función f y una condición inicial, $y(0)$.

Nótese que EDOs de orden superior siempre pueden ser escritas en esta forma introduciendo nuevas variables para las derivadas intermedias.

Una vez definida la función f y el arreglo y_0, podemos usar la función `odeint`:

$$y_t = \texttt{odeint}(f, y_0, t)$$

donde t es un arreglo con las coordenadas temporales para las que se resolverá el sistema de EDOs. El resultado y_t es un arreglo con una línea para cada punto de tiempo t, y donde cada columna corresponde a una solución $y_i(t)$ para ese tiempo.

Ejemplo

La ecuación de movimiento para el oscilador amortiguado es:

$$\frac{d^2x}{dt^2} + 2\zeta\omega_0\frac{dx}{dt} + \omega_0^2 x = 0$$

donde x es la posición del oscilador, ω_0 la frecuencia, y ζ es el factor de amortiguamiento.

Para escribir esta 2-EDO en la forma estándar, introducimos $p = dx/dt$

$$\frac{dp}{dt} = -2\zeta\omega_0 p - \omega^2 x$$
$$\frac{dx}{dt} = p$$

En la implementación de este ejemplo agregaremos algunos argumentos extras a la función del lado derecho de la EDO, en lugar de usar variables globales. Como consecuencia de los argumentos extra, necesitamos pasar un argumento clave `args` a la función `odeint`

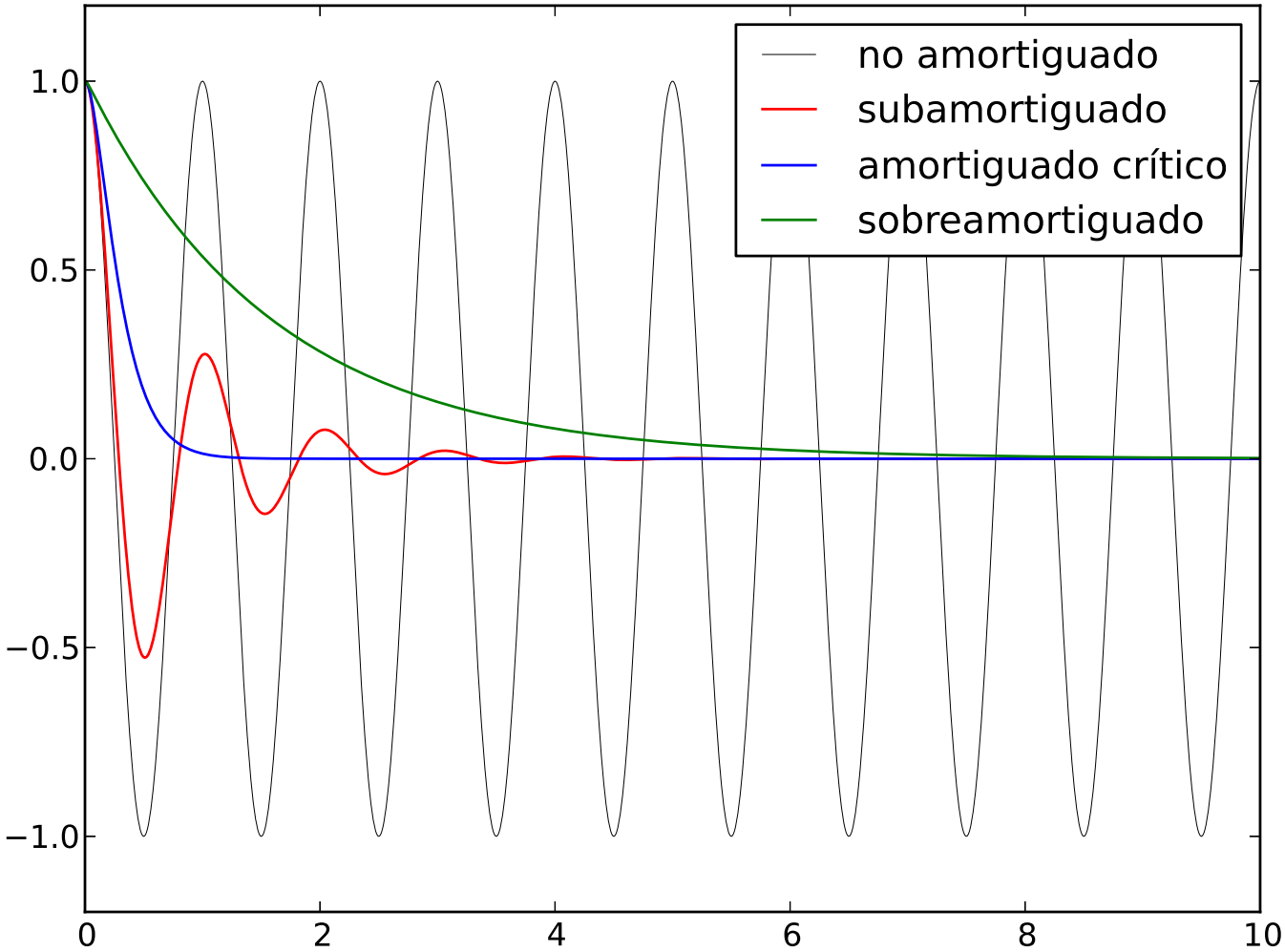
Código para resolver el problema

```
1 from scipy.integrate import odeint
2 from numpy import linspace
3 import matplotlib.pyplot as plt
4
5 def F(y, t, zeta, w0):
6     F = zeros((2), dtype='float64')
7     F[0] = y[1]
8     F[1] = -2 * zeta * w0 * y[1] - w0**2 * y
          [0]
9     return F
```

```
1
2  y0 =array([1.0, 0.0])
3
4  t = linspace(0, 10, 1000)
5  w0 = 2*pi*1.0
6
7
8  y1 = odeint(dy, y0, t, args=(0.0, w0))
9  y2 = odeint(dy, y0, t, args=(0.2, w0))
10 y3 = odeint(dy, y0, t, args=(1.0, w0))
11 y4 = odeint(dy, y0, t, args=(5.0, w0))
```

```
1 plt.axis([0,10,-1.2,1.2])
2 plt.plot(t, y1[:,0], 'k', label="no
    amortiguado", linewidth=0.25)
3 plt.plot(t, y2[:,0], 'r', label="
    subamortiguado")
4 plt.plot(t, y3[:,0], 'b', label="amortiguado
    critico")
5 plt.plot(t, y4[:,0], 'g', label="
    sobreamortiguado")
6 plt.legend()
7 plt.show()
```

Resultado


no amortiguado
subamortiguado
amortiguado crítico
sobreamortiguado
1.0
0.5
0.0
−0.5
−1.0
0
2
4
6
8
10