

Computación científica LF-214

Profesor: Julio C. Marín

Departamento de Meteorología, Universidad de Valparaíso


Primer Semestre

- Ecs. diferenciales de primer orden con 1 variable
- Función odeint de scipy

Función odeint de scipy

docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.odeint.html

Getting Started Imported Latest Headlines Saramago Asistencia al cl... Library Genesis Other bookmarks

 SciPy.org

SciPy.org

Docs

SciPy v1.6.3 Reference Guide

Integration and ODEs (scipy.integrate)

index

modules

next

previous

scipy.integrate.odeint

scipy.integrate.odeint(func, y0, t, args=(), Dfun=None, col_deriv=0, full_output=0, ml=None, mu=None, rtol=None, atol=None, tcrit=None, h0=0.0, hmax=0.0, hmin=0.0, ixpr=0, mxstep=0, mxhnil=0, mxordn=12, mxords=5, printmessg=0, tfirst=False)

[\[source\]](#)

Integrate a system of ordinary differential equations.

Note:

For new code, use [scipy.integrate.solve_ivp](#) to solve a differential equation.

Solve a system of ordinary differential equations using lsoda from the FORTRAN library odepack.

Solves the initial value problem for stiff or non-stiff systems of first order ode-s:

```
dy/dt = func(y, t, ...) [or func(t, y, ...)]
```

where y can be a vector.

Previous topic

[scipy.integrate.OdeSolution_](#)

Next topic

[scipy.integrate.ode](#)

Quick search

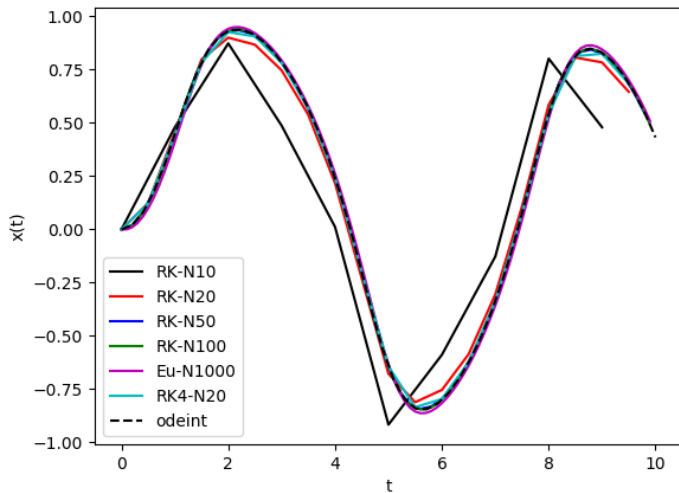
search

Mismo Ejemplo anterior

Resolver ODE $\frac{dx}{dt} = -x^3 + \sin t$ usando función odeint para $x(t=0) = 0$ en intervalo desde $a = 0$ a $b = 10$ en 50 iteraciones.

- `import numpy as np`
- `from scipy.integrate import odeint`
- `import matplotlib.pyplot as plt`
- `def model(x, t):`
 - `dxdt = -x**3 + np.sin(t)`
 - `return dxdt`
- `a, b = 0.0, 10.0 # tiempo inicial y final`
- `x0 = 0.0 # Condición inicial`
- `N = 50; h = (b-a)/N; t = np.arange(a, b, h)`
- `# Resolver ODE`
- `xp = odeint(model, x0, t)`

Mismo Ejemplo anterior



Ejercicio

Ejercicio: Resolver ODE $\frac{dy}{dt} = -ky$ para $k = 0.3$ entre 0 y 20 con la condición inicial $y(t = 0) = 5$

Ejercicio: Resolver ODE $\frac{dy}{dt} = -ky$ para $k = 0.3$ entre 0 y 20 con la condición inicial $y(t = 0) = 5$

- `import numpy as np`
- `from scipy.integrate import odeint`
- `import matplotlib.pyplot as plt`
- `# Función que entrega dy/dt`
- `def model(y, t):`
 - `k = 0.3`
 - `dydt = -k*y`
 - `return dydt`
- `y0 = 5.0 # Condición inicial`
- `t = np.linspace(0, 20, 50)`
- `yp = odeint(model, y0, t) # Resolver ODE`
- `plt.clf(); plt.plot(t, yp, '-k')`
- `plt.ylabel('y(t)'); plt.xlabel('t')`

Función odeint de scipy

- Se puede agregar un 4to argumento (args) a función odeint
 - Este 4to argumento es un tuple ()
 - Permite pasarle información a función 'model'
 - En ejemplo anterior supongamos que no conocemos k
-
- `def model(y, t, k):`
 - `dydt = -k*y`
 - `return dydt`
 - `y0 = 5.0 # Condición inicial`
 - `t = np.linspace(0, 20, 50)`
 - `k = 0.3`
 - `yp = odeint(model, y0, t, args=(k,)) # Resolver ODE`
 - `plt.clf(); plt.plot(t, yp, '-k')`
 - `plt.ylabel('y(t)'); plt.xlabel('t')`

Ejercicio

Ejercicio: Resolver ODE $\frac{dy}{dt} = -ky$ entre 0 y 20 con condición inicial $y(t = 0) = 5$ para los siguientes valores de $k = 0.1, 0.2$ y 0.5 . Plotear las 3 soluciones en el mismo plot

Ejercicio: Resolver ODE $\frac{dy}{dt} = -ky$ entre 0 y 20 con condición inicial $y(t = 0) = 5$ para los siguientes valores de $k = 0.1, 0.2$ y 0.5 . Plotear las 3 soluciones en el mismo plot

- `def model(y, t, k):`
- `dydt = -k*y`
- `return dydt`
- `y0 = 5.0; t = np.linspace(0,20,50)`
- `# Resolver ODE`
- `k = (0.1, 0.2, 0.5); lab_p = ('-k', '-b', '-r')`
- `plt.clf()`
- `for kk in range(len(k)):`
- `yp = odeint(model, y0, t, args=(k[kk],))`
- `plt.plot(t, yp, lab_p[kk])`
- `plt.ylabel('y(t)'); plt.xlabel('t')`
- `plt.legend(('k=0.1', 'k=0.2', 'k=0.5'), loc=1)`

Ejercicio

