

Дифференциальные уравнения

(функция `integrate.odeint`)

Ф.Я.Халили

МГУ, физический факультет

29 апреля 2008 г.

Функция odeint

```
odeint( $f(x, y)$ ,  $x_0$ , <вектор  $x$ >)
```

Эта функция находится в модуле `scipy.integrate` и, соответственно, загружается командой

```
from scipy.integrate import *   или  
from scipy.integrate import odeint
```

Для ее использования уравнение должно быть приведено к *стандартному виду*:

$$\frac{dy}{dx} = f(x, y),$$

где y и f могут быть векторами, что позволяет решать системы уравнений и уравнения порядка выше первого.

Моделирование сухого трения

В качестве примера рассмотрим систему с сухим трением:

$$\frac{dv}{dt} = -K \operatorname{sign}(v) + F(t),$$

где v – скорость тела, $F(t)$ – внешняя силы и K – коэффициент трения.

Чтобы не использовать разрывную функцию sign , заменим ее арктангенсом:

$$\frac{dv}{dt} = -\frac{2K}{\pi} \arctan(\alpha v) + F(t),$$

где α – большой коэффициент.

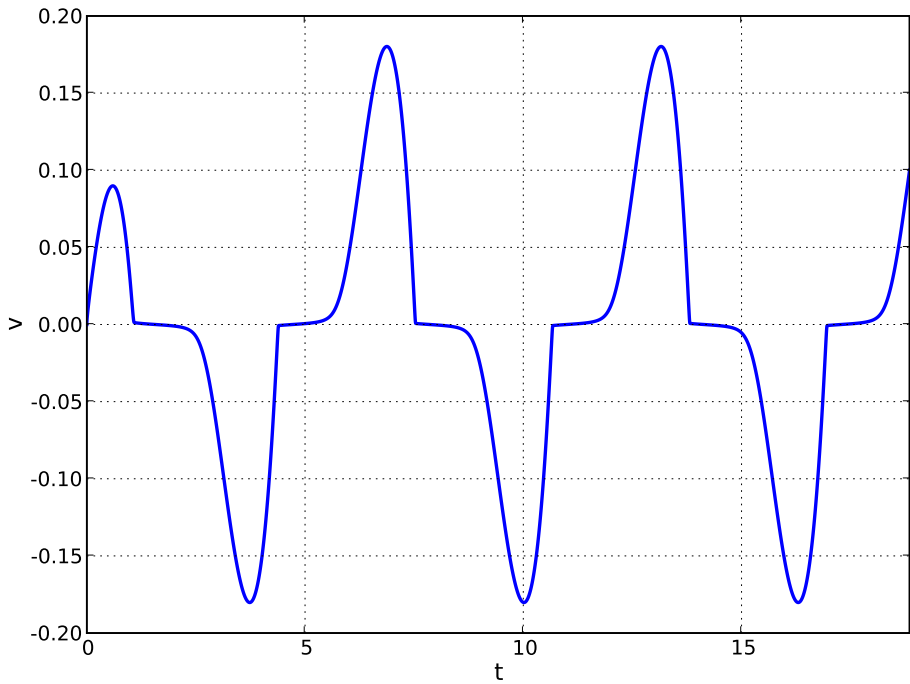
Моделирование сухого трения

```
from pylab import *
from scipy.integrate import *

t = linspace(0,6*pi,10000)

def f(v,t):
    return -(2/pi)*arctan(1000*v)+1.2*cos(t)

v = odeint(f,0,t)
plot(t,v,lw=2)
xlabel('t')
ylabel('v')
xlim(0,6*pi)
grid()
savefig('example9.pdf')
```



Нелинейный осциллятор

В качестве примера уравнения второго порядка, рассмотрим уравнение движения нелинейного осциллятора:

$$\frac{d^2x}{dt^2} + \omega^2(1 - \xi x^2)x = F \cos(\omega t) ,$$

где ξ – фактор нелинейности. Разделим уравнение на два первого порядка:

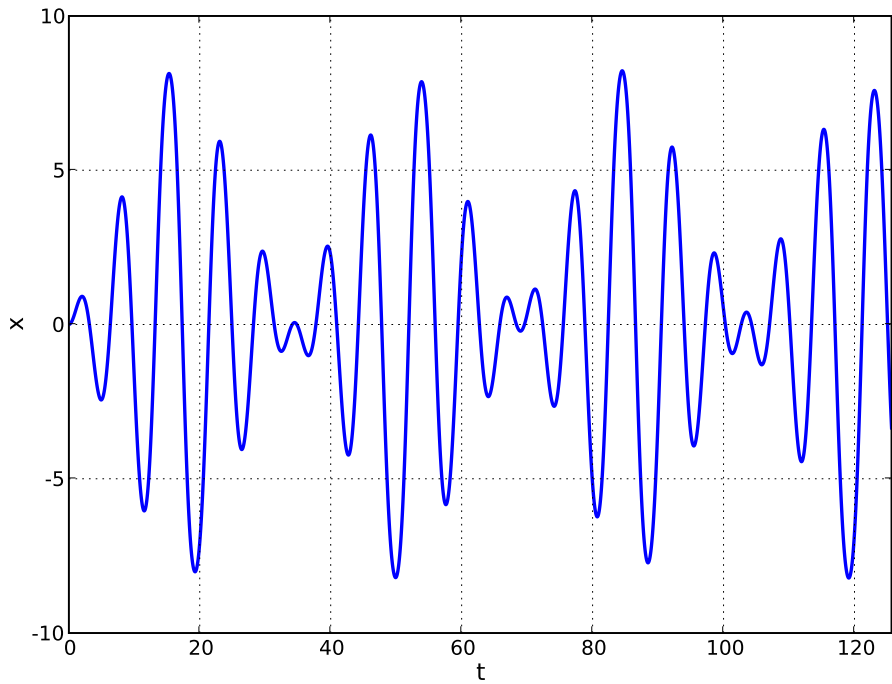
$$\begin{aligned} \frac{dx}{dt} &= v , \\ \frac{dv}{dt} + \omega^2(1 - \xi x^2)x &= F \cos(\omega t) , \end{aligned}$$

(v – скорость) и запишем в стандартном виде:

$$\frac{d}{dt} \begin{pmatrix} x \\ v \end{pmatrix} = \begin{pmatrix} v \\ -\omega^2(1 - \xi x^2)x + F \cos(\omega t) \end{pmatrix} .$$

Нелинейный осциллятор

```
from pylab import *
from scipy.integrate import *
t = linspace(0,40*pi,10000)
def f(y,t):
    x,v = y
    return [v, -(1-x**2/100)*x+cos(t)]
result = odeint(f,[0,0],t)
x = result[:,0]
v = result[:,1]
plot(t,x,lw=2)
xlabel('t')
ylabel('x')
xlim(0,40*pi)
grid()
savefig('example10.pdf')
```



Фазовый портрет

Можно также построить фазовый портрет системы:

```
rc('figure',figsize=(6.25,6))  
figure()  
plot(x,v,lw=2)  
xlabel('x')  
ylabel('v')  
xlim(-10,10)  
ylim(-10,10)  
grid()  
savefig('example10a.pdf')
```

