常 Ordinary Differential Equation

数値計算による微分方程式解法(python版)

cc by Shigeto R. Nishitani

- /Users/bob/Desktop/maple_ode/python_ode.ipynb
- origin git@github.com:daddygongon/maple_ode.git (fetch)

Table of Contents

Euler法による落下運動 ¶

重力場中の運動

重力場中のボールの落下を考えて，１軸で考えた運動方程式を立てます．

$$\begin{cases} v = \dfrac{dx}{dt} \\ a = \dfrac{d^2x}{dt^2} \end{cases}$$

mass gravity Force

質量をm, 重力加速度をgとして，働く力が$F = -mg$であるとすると，ニュートンの運動方程式$F = ma$は，

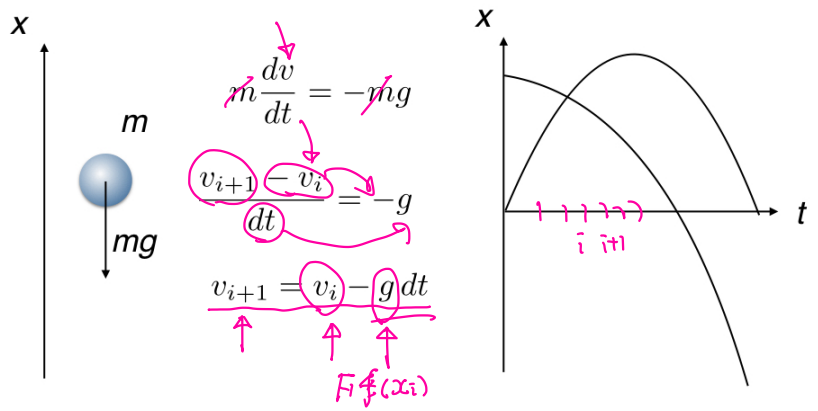
$$-mg = m\frac{d^2x}{dt^2}$$

となります．

力 ←|→ 自分の

Free gravitation fall

速度 velocity



Euler法

1次の微分方程式の一般形は

$$\frac{dx}{dt} = f(x,t)$$

と書けます．この微分方程式を簡単な近似から求めるオイラー法を示します． $x(t+\delta t)$をテイラー級数展開すると，

$$x(t+\delta t) \simeq x(t) + \frac{dx}{dt}\delta t$$

となります．これらを代入すると，計算アルゴリズムはつぎのようになります．

$$x_{i+1} = x_i + f_i\,\delta t$$

ここで，f_iは点(x_i, t_i)における関数の値です．このアルゴリズムを適用して，$t+\delta t$の座標x_{i+1}を一つ前の時間の座標x_iから導くことができます．これを重力場中の運動方程式に適用します．

重力場中の運動をEuler法で解いたら

Euler法は一階の微分方程式に対する定式化をしています．ところが，重力場中の運動は2階の微分方程式です．このようなときには媒介変数を導入して1次連立方程式に置き直します．

媒介変数として速度vを使って，2階の運動方程式

$$\frac{d^2x}{dt^2} = -g$$

が，1階の連立方程式

$$\frac{dv}{dt} = -g$$
$$\frac{dx}{dt} = v$$

で置き換えられると考えることに相当します．アルゴリズムにすると，

$$v_{i+1} = v_i - g\,dt$$
$$x_{i+1} = x_i + v_i\,dt$$

なる連立方程式を解くことに置き換わります．これをpythonで関数にして，さらに計算結果を表示させてみます．

In [1]:
```
def euler(x0, v0):
  v1 = v0 - g * dt
  x1 = x0 + v0 * dt
  return x1, v1
```

Eulerはx_i, v_iを受け取って，先ほど導いた簡単な計算によって，v_i+1, x_i+1を順次計算して返します．結果は，

In [2]:
```
import matplotlib.pyplot as plt

def my_plot(xx, vv, tt):
    plt.plot(tt, xx, color = 'b', linestyle='--',label="height")
    plt.plot(tt, vv, color = 'r', label="velocity")
    plt.legend()
    plt.xlabel('time')
    plt.ylabel('height and velocity')
    plt.grid()
    plt.show()
```

In [14]:
```
g, dt =9.8, 0.01
# g, dt =9.8, 0.1
# tt,xx,vv=[0.0],[0.0],[9.8] #for compare rain_drop
tt,xx,vv=[0.0],[0.0],[9.80]

t = 0.0

for i in range(0,220): #250 for compare rain_drop
  t += dt
  x, v = euler(xx[-1],vv[-1])
  tt.append(t)
  xx.append(x)
  vv.append(v)


my_plot(xx, vv, tt)
```