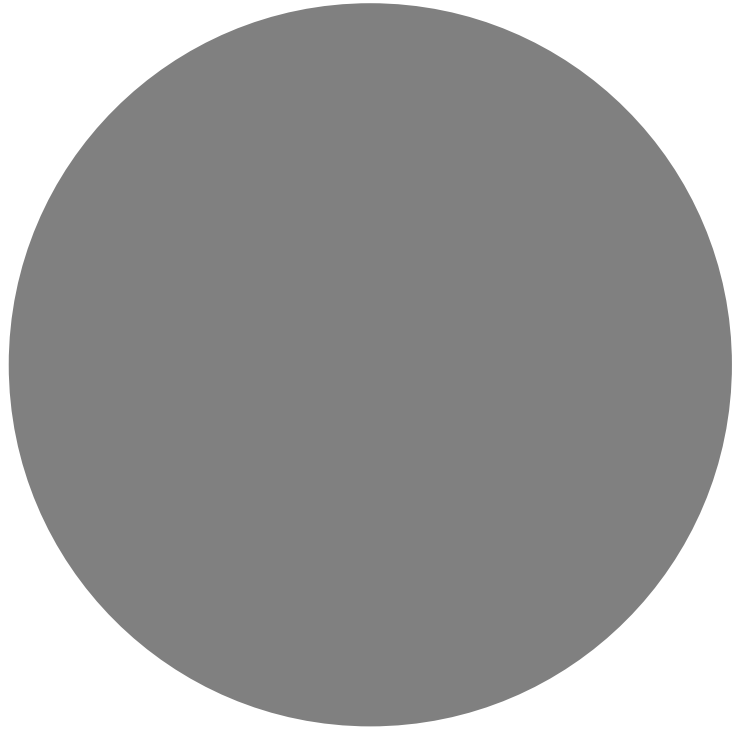
Runge-Kutta method



Equation and basic idea

$$\begin{cases} \frac{du}{dt} = f(u(t)) \\ u(0) = u_0 \end{cases}$$

Integrate both side of above, we have

$$u(t + \Delta t) - u(t) = \int_t^{t+\Delta t} f\big(u(\tau)\big) d\tau$$

We will introduce Euler method and Runge Kutta method according to the choice of discretization of right hand side.

Euler method

- Use rectangular approximation
- $\int_t^{t+\Delta t} f\big(u(\tau)\big)d\tau \cong \Delta t f(u(t))$
- Thus we got

$$u(t+\Delta t) = u(t) + \Delta t f(u(t))$$

2nd order Runge-Kutta

Use Trapezoidal rule (台形則)

$$\int_t^{t+\Delta t} f\big(u(\tau)\big)d\tau \cong \frac{\Delta t}{2}\{\ f(u(t) + f\big(u(t+\Delta t)\big)\ \ \}$$

Since we do not know $u(t+\Delta t)$, we use

$$u(t+\Delta t) = u(t) + \Delta t f(u(t))$$

Then we

$$f\big(u(t+\Delta t)\big) = f(\ \ u(t) + \Delta t f\big(u(t)\big)\ \ )$$

Eventually,

$$u(t+\Delta t) = u(t) + \frac{\Delta t}{2}\{\, f(u(t) + f\left(\ \ u(t) + \Delta t\ f\big(u(t)\big)\right)\}$$

4^{th} order Runge-Kutta

Use Simpson's rule (シンプソン則)

$$\int_t^{t+\Delta t} f\big(u(\tau)\big)d\tau \cong \frac{\Delta t}{6}\left\{\ f(u(t)+4f\left(u\left(t+\frac{1}{2}\Delta t\right)\right)+f(u+\Delta t)\ \right\}$$

Since we do not know $u\left(t+\frac{1}{2}\Delta t\right)$ and $u(t+\Delta t)$, we use

$u\left(t+\frac{1}{2}\Delta t\right)=\frac{1}{2}(u_1+u_2)$ where

$$u_1 = u(t)+\frac{\Delta t}{2}f(u(t))$$

$$u_2 = u(t)+\frac{\Delta t}{2}f(u_1)$$

Since we do not know $u(t+\Delta t)$, we use

$$u(t+\Delta t) = u(t) + \Delta t f(u(t))$$

$$f\big(u(t+\Delta t)\big) = f(\ (u_3\ \ )$$

where,

$$u_3 = u(t) + \Delta t \frac{\Delta t}{2} f(u_2)$$

Totally, we have

$$u(t+\Delta t) = u(t) + \frac{\Delta t}{2}\ \{\ f(u(t) + f\left(\ u(t) + \Delta t\ f\big(u(t)\big)\right)\}$$

Totally, we have

$$u(t+\Delta t) = u(t) + \frac{1}{6}\{k_0 + 2k_1 + 2k_2 + k_3\}$$

$$k_0 = \Delta t f(u(t))$$

$$k_1 = \Delta t f\left(u(t) + \frac{1}{2}k_0\right)$$

$$k_2 = \Delta t f\left(u(t) + \frac{1}{2}k_1\right)$$

$$k_3 = \Delta t f(u(t) + k_2)$$

```
from matplotlib import pyplot as plt
import numpy as np

def runge_kutta(f, t0, x0, v0, te, h):
    ts = np.arange(t0, te, h);
    xs = []
    vs = []
    x = x0
    v = v0
```

```
for t in ts:
        xs.append(x)
        vs.append(v)
        k1 = f(t, v, x);
        l1 = v

        k2 = f(t + h/2, v + h/2*k1, x + h/2*l1)
        l2 = v + h/2*k1

        k3 = f(t + h/2, v + h/2*k2, x + h/2*l2)
        l3 = v + h/2*k2

        k4 = f(t + h, v + h*k3, x + h*l3)
        l4 = v + h*k3

        v = v + (k1 + 2*k2 + 2*k3 + k4)/6*h
        x = x + (l1 + 2*l2 + 2*l3 + l4)/6*h

    return (ts, np.array(xs))
```

```
fig, ax = plt.subplots()

f = lambda t, v, x: -x
ts, xs = runge_kutta(f, 0, 10, 0, 100, 0.1)

ax.plot(ts, xs)
plt.show()
```