関数 y は 1 変数関数とし、変数は x とする。関数 y は変数 x の関数であることを利用し $y(x)$ と書くこともある。

y の導関数は y' 又は $y'(x)$ と書く。

$$y'(x) = f(x, y), \quad y(x_0) = y_0 \tag{1}$$

導関数 y' が x と y の式 (e.g. $y' = x\cos y$) とし、定数 x_0 に対し定数 y_0 が初期値とする。

オイラー法 (前進オイラー法)

初期値 x_0, y_0 をスタート地点として h だけ増やした $x_1 = x_0 + h$ の地点の y_1 を $y_1 = y_0 + hf(x_0, y_0)$ として少しづつ x_i, y_i を求める。$y_i = y(x_i)$ というイメージで近似している。

$$x_i = x_{i-1} + h, \quad y_i = y_{i-1} + hf(x_{i-1}, y_{i-1}) \tag{2}$$

ここで求めた (x_i, y_i) の組を座標として順番に繋いだ折れ線グラフが $y = y(x)$ の近似したグラフとなる。

当然、x_0 からより離れた x_i について近似値 y_i と本来の値 $y(x_i)$ は離れていく傾向がある。

$f(x)$ の $x = a$ を中心としたテーラー展開は次のような式で表される。

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n \tag{3}$$

$$= \frac{f(a)}{0!}(x-a)^0 + \frac{f'(a)}{1!}(x-a)^1 + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \cdots \tag{4}$$

これを区間 Δx と初期値 x_0 で次のように置き換える。$\Delta x = x - a,\ x_0 = a$

$$f(x_0 + \Delta x) = f(x_0) + f'(x_0)\Delta x + \frac{f''(x_0)}{2!}\Delta x^2 + \frac{f'''(x_0)}{3!}\Delta x^3 + \cdots \tag{5}$$

上記式から初期値 x_0 と区間 h だけ増やした $x_1 = x_0 + h$ の $y(x)$ の値が分かる。x_0 は x_i と置き換えても同じ式であるので、x とした式は次のようになる。

$$y(x+h) = y(x) + \frac{h}{1!}y'(x) + \frac{h^2}{2!}y''(x) + \frac{h^3}{3!}y'''(x) + \cdots + \frac{h^n}{n!}y^{(n)}(x) + \cdots \tag{6}$$

オイラー法での近似はこの式の前から 2 つの項 (h について 1 次式) を用いて近似している。後半の項を利用しない近似の為にズレが大きくなりやすい。そこで、上記の式の項から利用する部分を増やすことにより精度を高めることが出来る。

ホイン法

h について 2 次の項までを取り出すと以下のようになる。

$$y(x+h) \approx y(x) + \frac{h}{1!}y'(x) + \frac{h^2}{2!}y''(x) \tag{7}$$

導関数の定義より極限を取らずに h が十分小さいとして近似を考える。

$$y''(x) = \lim_{h\to 0} \frac{y'(x+h) - y'(x)}{h} \quad \text{より} \quad y''(x) \approx \frac{y'(x+h) - y'(x)}{h} \tag{8}$$

$y''(x)$ を置き換えることで次の近似式が得られる。

$$y(x+h) \approx y(x) + \frac{h}{2}y'(x) + \frac{h}{2}y'(x+h) \tag{9}$$

この式を用いた近似を**ホイン法**という。

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x,y) \tag{10}$$

刻み幅を h とすると $x_{i+1} = x_i + h$ という関係がある。これに対してそれぞれの方法で y_i の値を近似する。

- **オイラー法**

$$y_{i+1} = y_i + hf(x_i, y_i) \tag{11}$$

- **ホイン法**　一部 (y_{i+1}) にオイラー法を用いる

$$\begin{aligned} y_{i+1} =& y_i + \frac{1}{2}\left(hf(x_i,y_i) + hf(x_{i+1},y_{i+1})\right) && (12)\\ =& y_i + \frac{1}{2}\left(hf(x_i,y_i) + hf(x_{i+1},y_i + hf(x_i,y_i))\right) && (13) \end{aligned}$$

- **ルンゲ-クッタ法 (4 次)**

$$\begin{aligned} y_{i+1} =& y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) && (14)\\ k_1 =& hf(x_i, y_i) && (15)\\ k_2 =& hf\left(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right) && (16)\\ k_3 =& hf\left(x_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right) && (17)\\ k_4 =& hf(x_i + h, y_i + k_3) && (18) \end{aligned}$$

$$\begin{cases} \dot{u} = u & (0 \leq t \leq 1) \\ u(0) = 1 \end{cases} \tag{19}$$

$\dot{u}$ は u の導関数であり、この微分方程式の解は $u = e^t = \exp(t)$ である。
次の 3 種類の方法で数値解 $\{U_n\}_{n=1}^N$ を求める。

1. **オイラー法**　ただし、$\Delta t = 0.0001$ $(N = 10000)$
2. **ホイン法**　ただし、$\Delta t = 0.01$ $(N = 100)$
3. 4 次の**ルンゲ-クッタ法**　ただし、$\Delta t = 0.1$ $(N = 10)$

数値解 $\{U_n\}_{n=1}^N$ と理論解 $u = \exp(t)$ との差を求めよ。

$$|U_n - \exp(t_n)| = |U_n - \exp(n \cdot \Delta t)| \tag{20}$$

これらを横軸が時間 t のグラフにまとめよ。

```
1  printf("%f  %.10f\n", i*DT, fabs(u-exp(i*DT)));
```

オイラー法

(19) より初期値は $t = 0$ のとき $u(0) = 1$ なので、$t_0 = 0, u_0 = 1$。
オイラー法の式を問の微分方程式の記号に合わせると次のように表せる。

$$u_{i+1} = u_i + \Delta t u_i \tag{21}$$

$\Delta t = 0.0001$ より t に 0.0001 が加算されるごとに次のように u_i が求められる。

$$u_{i+1} = (1 + \Delta t)u_i = 1.0001 \times u_i \tag{22}$$

この式を使って最初の方を求めると次のようになる。

$$(t_0, u_0) = (0, 1) \tag{23}$$
$$(t_1, u_1) = (t_0 + 0.0001, 1.0001 \times u_0) = (0.0001, 1.0001) \tag{24}$$
$$(t_2, u_2) = (t_1 + 0.0001, 1.0001 \times u_1) = (0.0002, 1.00020001) \tag{25}$$
$$(t_3, u_3) = (t_2 + 0.0001, 1.0001 \times u_2) = (0.0002, 1.000300030001) \tag{26}$$

これを繰り返し 10000 回行うと $0 \leq t \leq 1$ の間の値が求められる。

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main(void){
5
6      double DT=0.0001;
7      double u=1.0;
8
9      // 初期値表示
10     printf("t=%f u=%f exp(t)=%f 差分：%.10f\n", 0*DT, u, exp
         (0*DT), fabs(u - exp(0*DT) ));
11
12     // 近似値と理論値、その差分 表示
13     for (int i=1 ; i<=1000 ; i++){
14         u= u*(1+DT);
15         printf("t=%f u=%f exp(t)=%f 差分：%.10f\n", i*DT, u,
             exp(i*DT), fabs(u - exp(i*DT)));
16     }
17     return 0;
18 }
```

このコードは 1000 回になっているので、13 行目の `i<=1000` を `i<=10000` に変える必要がある。

ホイン法

ホイン法の式の記号を方程式の記号に合わせると次のようになる。

$$u_{i+1} = u_i + \frac{\Delta t}{2}\left(u_i + u_{i+1}\right) \tag{27}$$

右辺の u_{i+1} はオイラー法の式 (22) を使い置き換えると次のような式になる。

$$u_{i+1} = \left(1 + \Delta t + \frac{1}{2}\Delta t^2\right) u_i \tag{28}$$

$\Delta t = 0.01$ として $0 \leq t \leq 1$ の範囲で近似値を計算する。

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main(void){
5
6      double DT=0.01;
7      double u=1.0;
8
9      // 初期値表示
10     printf("t=%f u=%f exp(t)=%f 差分：%.10f\n", 0*DT, u, exp
         (0*DT), fabs(u - exp(0*DT) ));
11
12     // 近似値と理論値、その差分 表示
13     for (int i=1 ; i<=100 ; i++){
14         u= u*(1+DT+pow(DT,2)/2);
15         printf("t=%f u=%f exp(t)=%f 差分：%.10f\n", i*DT, u,
             exp(i*DT), fabs(u - exp(i*DT) ));
16     }
17     return 0;
18 }
```

ルンゲ-クッタ法 (4 次)

式の記号を置き換えると次のようになる。

$$u_{i+1} = u_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{29}$$

$$k_1 = \Delta t u_i \tag{30}$$

$$k_2 = \Delta t \left(u_i + \frac{k_1}{2} \right) \tag{31}$$

$$k_3 = \Delta t \left(u_i + \frac{k_2}{2} \right) \tag{32}$$

$$k_4 = \Delta t (u_i + k_3) \tag{33}$$

これをプログラムにすると次のようになる。

```c
#include <stdio.h>
#include <math.h>

int main(void){

    double DT=0.1;
    double u=1.0;
    double k1,k2,k3,k4;

    // 初期値表示
    printf("t=%f u=%f exp(t)=%f 差分：%.10f\n", 0*DT, u, exp
      (0*DT), fabs(u - exp(0*DT) ));

    // 近似値と理論値、その差分 表示
    for (int i=1 ; i<=10 ; i++){
        k1 = DT * u;
        k2 = DT * (u + k1/2);
        k3 = DT * (u + k2/2);
        k4 = DT * (u + k3);
        u= u + (k1 + 2*k2 + 2*k3 + k4)/6;
        printf("t=%f u=%f exp(t)=%f 差分：%.10f\n", i*DT, u,
          exp(i*DT), fabs(u - exp(i*DT) ));
    }
    return 0;
}
```