

しかし数値計算を使うと関数 $\theta(t)$ を使わずに、運動方程式から直接的に時刻 t におけるほぼ正確な角度を求めることができます。

このように解析的に解けない微分方程式でも数値計算により近似的に解を得ることができます。今回は常微分方程式の数値計算法の中でもアルゴリズムを理解しやすいオイラー法(Euler method) について学びます。

1.9.1 オイラー法 (Euler method)

オイラー法は次の漸化式を計算することで微分方程式の近似解を得るアルゴリズムです。

$$x_{k+1} = x_k + hf(x_k, t_k) \quad (1.15)$$

1 階常微分方程式

$$\frac{dx}{dt} = f(x(t), t) \quad (1.16)$$

を解くことを考えます。

数値的に解く方法として、区間を $t = t_0, t_1, t_2, \dots, t_k, \dots, t_n$ のように区切り、対応する x の数値 $x = x_0, x_1, x_2, \dots, x_k, \dots, x_n$ を求めてみましょう。もし微分方程式が

$$\frac{dx}{dt} = f(t) \quad (1.17)$$

のように未知関数 $x(t)$ を含まない式であれば、両辺を積分するだけで解が求まります。高校の数学で学んだ区分求積法を応用すれば簡単におおよその積分ができます。しかし式 (1.16) にこの方法は適用できません。

t のきざみ幅 Δt を h とすると ($\Delta t = h = t_{k+1} - t_k$)、式 (1.16) の左辺の導関数は次のように近似できます。

$$\left. \frac{dx}{dt} \right|_{t=t_k} \simeq \frac{\Delta x}{\Delta t} \Big|_{t=t_k} = \frac{x_{k+1} - x_k}{h} \quad (1.18)$$

この式を前進差分と呼びます。式 (1.16)、(1.18) より

$$x_{k+1} = x_k + hf(x_k, t_k) \quad (1.19)$$

の漸化式を得ます。

図 (1.4) のように、オイラー法をグラフを使って考えると直感的に理解しやすいと思います。 x_k, t_k における傾きにきざみ幅 h を掛けると Δx_{k+1} を得ます。それまで得ている x_k ($k = 0$ の場合は初期値 x_0) に Δx_{k+1} を足すことで x_{k+1} を得ます。この操作を繰り返すことで $x(t)$ を得ることができます。

また、図から明らかなように、 k を増やすたびに真の値との誤差が蓄積されていきます。一般的には $\lim_{h \rightarrow 0} h$ とすることで真の値となりますが、コンピュータでは有限の値しか扱えないことと、丸め誤差による誤差が大きくなっていくためにどうしてもオイラー法では誤差が大きくなります。

このように誤差の大きいオイラー法はあまり実用的ではありませんが、アルゴリズムを理解しやすいため常微分方程式の数値計算の概念を学ぶには最適な手法です。

1.9.2 1 階のオイラー法

では、微分方程式 $\frac{dx}{dt} = t$ をオイラー法を使って解いてみましょう。初期条件は $x(0) = 0$ で、 $t = 5$ の時点での解を求めます。

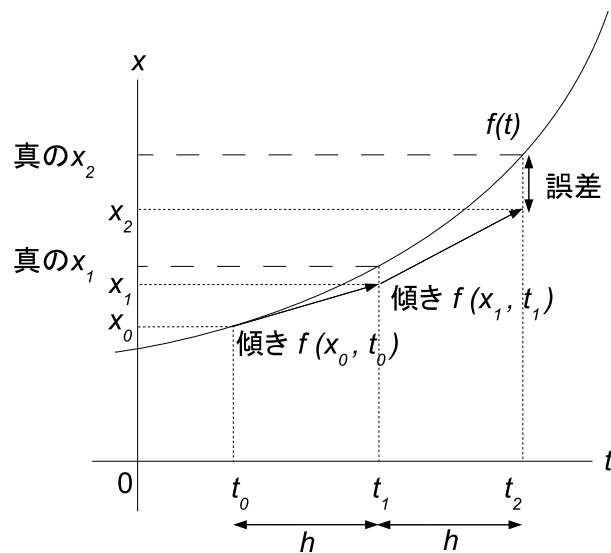


図 1.4 オイラー法

例題 22 オイラー法

example-22.c

```
#include <stdio.h>

int main(void)
{
    double x, t, h, t_last;
    x = 0;
    t = 0;
    h = 0.01;
    t_last = 5.0;

    printf("#t\tx\n");
    while (t <= t_last) {
        x = x + h * (t); /* 漸化式 */
        printf("%f\t%f\n", t, x);
        t = t + h;
    }
    printf("%f\t%f\n", t, x);
    return 0;
}
```

実行結果

#t	x
0.000000	0.000000
0.010000	0.000100
...	
5.000000	12.525000
5.010000	12.525000

次に微分方程式 $\frac{dx}{dt} = x(t)$ をオイラー法を使って解いてみましょう。初期条件は $x(0) = 1$ で、 $t = 5$ の時点での解

を求めます。

例題 23 オイラー法

example-23.c

```
#include <stdio.h>

int main(void)
{
    double x, t, h, t_last;
    x = 1;
    t = 0;
    h = 0.01;
    t_last = 5.0;

    printf("#t\tx\n");
    while (t <= t_last) {
        x = x + h * (x); /* 漸化式 */
        printf("%f\t%f\n", t, x);
        t = t + h;
    }
    printf("%f\t%f\n", t, x);
    return 0;
}
```

実行結果

#t	x
0.000000	1.010000
0.010000	1.020100
...	
5.000000	146.220500
5.010000	146.220500

1.9.3 課題

課題 19 微分方程式 $\frac{dx}{dt} = \cos(t)$ をオイラー法で解いてみよう。初期条件は $x(0) = 1$ で、 $t = 7$ とする。

課題 20 微分方程式 $\frac{dx}{dt} = x \cos(t)$ をオイラー法で解いてみよう。初期条件は $x(0) = 1$ で、 $t = 7$ とする。

課題 21 各例題課題の解析解とオイラー法で求めた数値解との誤差を比較してみよう。