数値解析レポート No.4

32 番 平田 蓮

1 連立微分方程式

$$
\begin{aligned}
f(x, y; t) &= \frac{dx}{dt} = x + 6y + t - 10 \\
g(x, y; t) &= \frac{dy}{dt} = x + t - 3
\end{aligned}
$$

とすると、x, y の値をオイラー法のように解くことができる。具体的には以下の式を実装すれば良い。

$$
\begin{aligned}
x_{i+1} &= x_i + f(x_i, y_i; t)dt \\
y_{i+1} &= y_i + g(x_i, y_i; t)dt
\end{aligned}
$$

リスト 1 に二連立微分方程式を解く関数 `sim_euler()` を示す。

リスト 1 sim_euler.c

```
#define N 10000

void sim_euler(
    double (*f)(
        double x, double y, double t
    ),
    double (*g)(
        double x, double y, double t
    ),
    double x[N],
    double y[N],
    double l,
    double r,
    int n,
    int debug
) {
    double t = 0, dt = (l + r) / (double)n;
    int i;

    for (i = 1; i <= n; i++, t += dt) {
```

```
        if (debug) {
            printf("t: %4.4f, x: %4.4f, y: %4.4f\n", t, x[i - 1], y[i - 1]);
        }

        x[i] = x[i - 1] + f(x[i - 1], y[i - 1], t) * dt;
        y[i] = y[i - 1] + g(x[i - 1], y[i - 1], t) * dt;
    }

    if (debug) {
        printf("t: %4.4f, x: %4.4f, y: %4.4f\n", t, x[n], y[n]);
    }
}
```

この関数を用いて x, y の解を求めたものを図 1, 2 に示す。

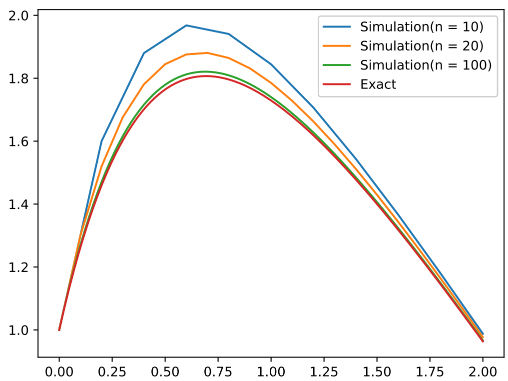

図 1　x の解

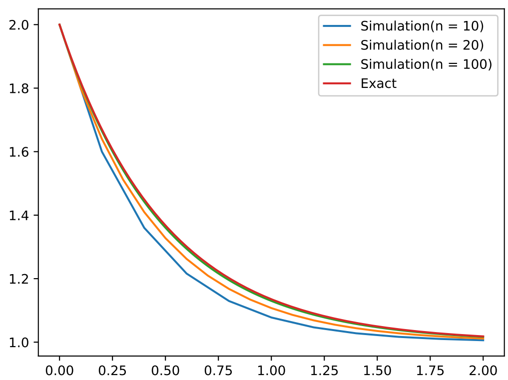

図 2　y の解

ステップ幅を狭めるほど正確な近似ができていることがわかる。

2　高次微分方程式

$\dfrac{d^2y}{dx^2} + 5\dfrac{dy}{dx} - 6y = 0$ について、$\dfrac{dy}{dx} = z$ とすることで, 以下のように二次連立微分方程式とできる。

$$\frac{dz}{dx} = 6y - 5z$$
$$\frac{dy}{dx} = z$$

これは前節と同じように解くことができる。y の解を図 3 に示す。

分割数 $n = 100$ のとき、$x = 1$ に対して $y \approx 2.72$ である。

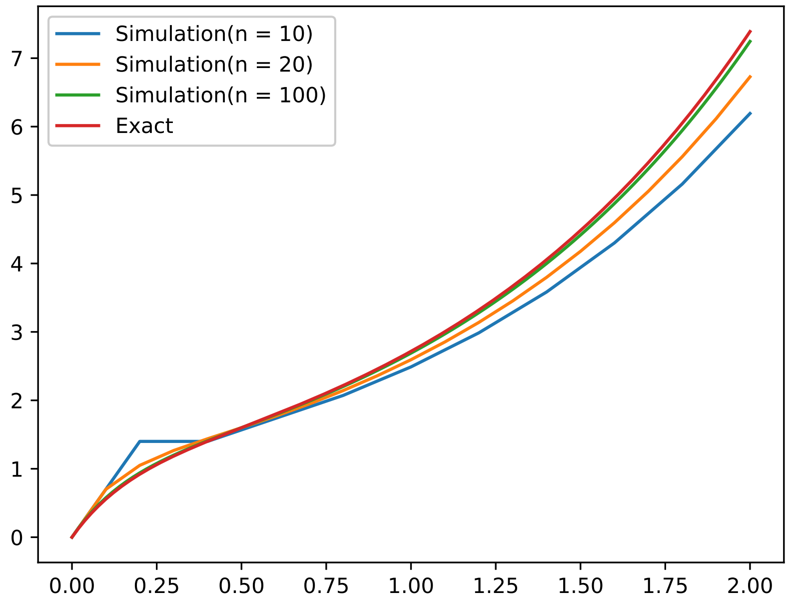

図 3　y の解

3　考察

どちらの微分方程式も、オイラー法の式を使い同様に解いたが、ホイン法の漸化式を使うと精度が高まることが考えられる。

また、今回作成したプログラムは二連立にしか対応していないが、任意個数の引数を受け取る関数を作ることで n 連立に対応させられると考えられる。