

## 第 15 週目 追加問題

解答は答えだけでなく、途中経過も記述すること（プログラム作成の場合はこの限りでない）。解答用紙が足りない場合は別途解答用紙を印刷し、ホッチキス止めしたものを提出すること。その際は、それぞれの解答用紙に氏名、学籍番号を記載すること。

### [問 1]

次ページに 4 次のルンゲ・クッタ法を利用して、

$$\frac{dy}{dt} = 1 - y^2$$

を解くプログラムを示す。プログラムでは、最終結果のみ表示している。このプログラムを修正し、実行時間を 20%程度削減せよ。ただし、以下の条件のもとでプログラムを修正する。

- 表示は最終結果のみでよい。
- アルゴリズムは、4 次のルンゲ・クッタを使用する。
- 微分方程式は、上記の形式を対象とすることを前提にしてよい（関数の書式は変更してよい）。

実行時間は PC の状況によって多少変動するので、数回実行して平均を確認すること。PC の性能によっては、次のページのプログラムでは、実行時間が非常に小さい場合がある。その場合には、計算回数数を 10 倍、100 倍と変更してみよう。

(ヒント) 掛け算／割り算の回数を減らすことを考えてみよう。

4 次のルンゲ・クッタ法で微分方程式を解く (計算時間を計測)

```
#include <stdio.h>
#include <time.h>

// この例では  $y' = 1 - y^2$  を解いている
double f(double x, double y)
{
    return 1-y*y;
}

int main(void)
{
    clock_t start;
    clock_t end;
    int i;
    double x = 0;
    double y = 0;
    double h = 0.01; // ステップ幅
    long n = 100000000; // 計算回数
    double x0 = 0;
    double y0 = 0;
    double f1;
    double f2;
    double f3;
    double f4;

    start = clock();

    for(i=1;i<=n;i++){
        f1 = h * f(x, y);
        f2 = h * f(x+h/2, y+f1/2.0);
        f3 = h * f(x+h/2, y+f2/2.0);
        f4 = h * f(x+h, y+f3);
        y = y + (f1 + 2*f2 + 2*f3 + f4)/6.0;
        x = x0 + h * i;
    }

    printf("%f: %f\n", x, y);

    end = clock();

    printf( "処理時間:%d[ms]\n", end - start );

    return 0;
}
```