

基礎制御工学および演習 (第 11 回)

	2015 年 12 月 8 日 実施
学籍番号 15FR999	氏名 かいとう れいこ

【1 次チェック】

回	チェック日	チェック者 (氏名)	判定
1-1	月 日	(TA, 受講生)	(合格、再提出)
1-2	月 日	(TA, 受講生)	(合格、再提出)
1-3	月 日	(TA, 受講生)	(合格、再提出)

【2 次チェック】

回	チェック日	チェック者 (氏名)	判定
2-1	月 日	shio	(合格、再提出)
2-2	月 日		(合格、再提出)

実施要領

[標準フロー]

1. 演習プリントを受け取ったら、まず学籍番号と氏名を記入する。
2. 演習時間内に全問解答し、副手または担当教員の指名する 1 次チェック担当者による 1 次チェックを受ける。
3. 全問正解状態になるまで 不正解問題の理解、答案修正、1 次チェックを繰り返す。
4. 1 次チェックが完了し 全問正解状態になったら 指導教員の 2 次チェックを受ける。
5. 2 次チェックが完了し「合格」印を受けると他の受講生の 1 次チェック担当に指名されることがある。
6. 2 次チェック完了答案は演習時間終了時に提出する。次回の演習開始時に返却される。
7. 演習時間内に 1 次および 2 次チェックが完了しなかった者については、別途指示する。
8. 解答済演習プリントは大切に保管し、学期末にポートフォリオとして整理し提出する。

[質問受付] 授業担当は汐月 (11016A 室)、井上 (11001A 室)。

担当副手は 根本 (岩)、*和久井 (汐)、満洲 (鈴)、*倉持 (汐)、千脇 (汐)、齋藤 (昌) です。

(岩) = 10424 室、(昌) = 10425 室、(汐) = 10426 室、(鈴) = 11003 室

1. 漸化式の反復計算によるプログラミング

以下のプログラムはフィボナッチ数を漸化式 $F_{n+2} = F_{n+1} + F_n$ をの繰返しで実行するプログラムである (fibonacci.mm)。これを改造してトリボナッチ数列 $F_{n+3} = F_{n+2} + F_{n+1} + F_n$, $n = 0, 1, 2, \dots$ を計算する。以下の問いに答えよ。

```
Func void main()
{
    Integer f1, f2, f3, fn, n, N=12;

    n=0; f1 = 0;
    printf("f%d = %d\n", n, f1);
    n=1; f2 = 1;
    printf("f%d = %d\n", n, f2);
    for(n=2; n<=N ;n++) {
        fn = f1 + f2 + f3;
        printf("f%d = %d\n", n, fn);
        f1 = f2;
        f2 = fn f3;
    } f3 = fn;
}
main
```

- (1) 変更・追加する内容を赤で示せ。
- (2) 学籍番号の下3桁を初期値 F_0, F_1, F_2 としてトリボナッチ数列 $F_0, \dots, F_{29}$ を計算し下表に記入せよ。
- (3) 同じ計算を電卓を用いて計算し下表に記入せよ。
- (4) (2)(3)の結果を比較し、値が異なる場合にはその理由を説明せよ。

[MaTX/PCによる計算結果]

0	1	2	3	4	5	6	7	8	9
0	1	1	2	4	7	13	24	44	81
10	11	12	13	14	15	16	17	18	19
149	274	504	927	1705	3136	5768	10609	19513	35890
20	21	22	23	24	25	26	27	28	29
66012	121415	223317	410744	755476	1389537	2555757	4700770	8646064	15902591

[電卓による計算結果]

0	1	2	3	4	5	6	7	8	9
0	1	1	2	4	7	13	24	44	81
10	11	12	13	14	15	16	17	18	19
149	274	504	927	1705	3136	5768	10609	19513	35890
20	21	22	23	24	25	26	27	28	29
66012	121415	223317	410744	755476	1389537	2555757	4700770	8646064	15902591

2. 行列の反復計算によるプログラミング

以下のプログラムはフィボナッチ数を計算する漸化式 $F_{n+2} = F_{n+1} + F_n$ を MaTX の行列演算機能を利用して実行するプログラムである (fibonacc2.mm)。これを改造してトリボナッチ数列 $F_{n+3} = F_{n+2} + F_{n+1} + F_n$, $n = 0, 1, 2, \dots$ を計算する。以下の問いに答えよ。

```
Func void main()
{
    Integer i, N;
    Matrix F, x;

    N = 12;
    F = [[0 1][1 1]]; [[0,1,0][0,0,1][1,1,1]];
    x = [0 1]; [0,0,1]';
    for(i=0; i<=N ;i++) {
        printf("f%d = %3.0f\n", i, x(1));
        x = F*x;
    }
}

main
```

- (1) 変更・追加する内容を赤で示せ。
- (2) 学籍番号の下3桁を初期値 F_2, F_1, F_0 としてトリボナッチ数列 $F_0, \dots, F_{29}$ を計算し下表に記入せよ。
- (3) 同じ計算を電卓を用いて計算し下表に記入せよ。
- (4) (2)(3)の結果を比較し、値が異なる場合にはその理由を説明せよ。

[MaTX/PC による計算結果]

[illegible]

[電卓による計算結果]

[illegible]

3. 円周率 π を計算する

円周率 π を計算する計算式を調査し、アルゴリズム（公式、級数展開表現）をひとつ決めて、

- (1) アルゴリズムを記述せよ。
- (2) MaTX で計算するプログラムを作成せよ。
- (3) 計算結果を記入せよ。

【参考1】 ライプニッツの公式

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1}$$

【参考2】 ガウス・ルジャンドルのアルゴリズム

初期条件: $a_0 = 1, b_0 = \frac{1}{\sqrt{2}}, t_0 = \frac{1}{4}, p_0 = 1$

反復式:
$$\begin{cases} a_{n+1} = (a_n + b_n)/2, & b_{n+1} = \sqrt{a_n b_n}, \\ t_{n+1} = t_n - p_n(a_n - a_{n+1})^2, & p_{n+1} = 2p_n \end{cases}$$

近似値: $\pi \approx \frac{(a_n + b_n)^2}{4t_n}$

※ Wikipedia(<https://ja.wikipedia.org/wiki/円周率>, 円周率の歴史など) や岩波数学公式II(森口、宇田川、一松著,1957) なども参照せよ。

[(1) 使用したアルゴリズム（公式、級数展開表現）]

ガウス・ルジャンドルのアルゴリズム (上の参考2)

[(2) FILE] () .mm

```
Func void main()
{
    Integer i, n, N;
    Real a0, b0, t0, p0;
    Real a1, b1, t1, p1;

    a0 = 1.0; b0 = 1/sqrt(2.0);
    t0 = 1/4.0; p0 = 1.0;
    for (N = 1; N <= 10; N++) {
        a1 = (a0 + b0)/2.0;
        b1 = sqrt(a0 * b0);
        t1 = t0 - p0 * (a0 - a1) * (a0 - a1);
        p1 = 2 * p0;
        printf("%2d, %15.8f\n",
            N, (a1 + b1) * (a1 + b1) / 4 / t1);
        a0 = a1; b0 = b1;
        t0 = t1; p0 = p1;
    }
    main
```

[(3) 計算結果]

0. (3.14057925)
1. (3.14159265)
2. (3.14159265)
3. ()
4. ()
5. ()
6. ()
7. ()
8. ()
9. ()
10. ()
11. ()
12. ()
13. ()
14. ()
15. ()
16. ()
17. ()
18. ()
19. ()
20. ()

↓
以下変化なし