

第4章 演習課題

課題 1

サンプルプログラムをコンパイル・実行して動作を確認せよ。さらに、適宜修正してその実行結果を確認せよ。

課題 2

標準入力から2つの整数 (n, m とする) を読み込み、その大小を比較するプログラムを作成せよ。例えば $n = 1, m = 2$ なら以下のように `1 is smaller than 2` と表示する。

```
$/a.out
Input two integers:
1    # キーボード入力
2    # キーボード入力
      1 is smaller than      2
```

同様に $n = 2, m = 1$ なら `2 is larger than 1`, $n = m = 1$ なら `1 is equal to 1` などと表示するものとする。

課題 3

0° から 180° まで 10° 刻みの θ および, $\sin \theta, \cos \theta$ を標準出力に表示するプログラムを作成せよ (以下のように各 θ の値ごとに改行せよ)。またこの結果をリダイレクトを用いてファイルとして記録し, `gnuplot` を用いてこのファイルのデータと `gnuplot` に組み込みの三角関数を共に図示せよ。なお三角関数の引数はラジアン単位であることに注意せよ。プログラムの出力結果は以下のようなものとなる。

```
$ ./a.out
0.0000000000000000      0.0000000000000000      1.0000000000000000
10.0000000000000000    ...                      ...
20.0000000000000000    ...                      ...
...                     ...                      ...
180.0000000000000000    ...                      ...
```

なおリダイレクトでデータファイルを作成するには

```
$ ./a.out > data.dat
```

とすれば良い。 `gnuplot` では

```
> plot 'data.dat' using 1:2 w lp, sin(x/180*pi)
> replot 'data.dat' using 1:3 w lp, cos(x/180*pi)
```

などとすれば結果を確認出来る。(忘れていた人は復習せよ!)

課題4

標準入力から与えられた2つの整数 $m, n \geq 1$ の最大公約数を表示するプログラムを作成せよ。最大公約数を求めるには以下のアルゴリズム (ユークリッドの互除法) を用いるとよい。

- [1] m を n で割った余り r を求める。
- [2] もし $r = 0$ ならば n が最大公約数である。
- [3] もし $r \neq 0$ ならば, m に n を, n に r を代入して [1] に戻る (繰り返す)。

なお, 組み込み関数 `mod` を用いて

```
r = mod(m, n)
```

とすれば m を n で割った余りを r に代入することが出来る。

課題5

以下の級数計算により自然対数の底 e の近似値を求めるプログラムを作成せよ。

$$e \simeq \sum_{n=0}^N \frac{1}{n!}. \quad (0! = 1 \text{ に注意せよ})$$

ただし以下の条件を満たすこと。

- ▷ 級数の項数 N および許容誤差 ϵ を標準入力から読み込む。
- ▷ $N > 1$ でない場合および $0 < \epsilon < 1$ でない場合にはエラーメッセージを表示して終了する。
- ▷ 誤差が ϵ 以下になった時点か, $n = N$ まで計算した時点で級数計算を打ち切る。
- ▷ 最後に収束したかどうか, 最終的な項数 N , 真値, 近似値, 相対誤差を表示して終了する。

出力のイメージとしては例えば以下のようなものである。

```
$ ./a.out
10          # キーボード入力
1.0e-8      # キーボード入力
Did not converge !
N           :           10
Exact value : 2.7182818284590451
Approximated value : 2.7182818011463845
Error       : 1.00477663102110533E-008
```

課題6

標準入力から文字列 (英単語) を読み込み, それが `food`, `animal`, `vehicle`, `others` (それ以外) のいずれかを判定し, 表示するプログラムを作成せよ. ただし `exit` が入力されるまでプログラムは終了せず何度でも入力を受け付けるものとする. なお以下の英単語リスト以外のものは `others` と判断してよい: `apple`, `orange`, `banana`, `dog`, `cat`, `lion`, `car`, `airplane`, `motorcycle`.

実行結果は例えば以下のような出力になるだろう.

```
$ ./a.out
apple    # キーボード入力
        food
cat      # キーボード入力
        animal
car      # キーボード入力
        vehicle
dog      # キーボード入力
        animal
airplane # キーボード入力
        vehicle
bike     # キーボード入力
        others
exit     # キーボード入力
Now exit program...
```

課題7 †

以下の漸化式

$$p_{n+1} = p_n + \alpha p_n (1 - p_n)$$

で定義される数列 $p_n (n = 0, 1, \dots)$ を考える. 初期値 $p_0 = 0.9$ から数列を生成し, そのうち $n = 100, \dots, 200$ までを α の関数として $1 < \alpha < 3$ の範囲でプロットせよ. α を 10^{-3} 刻みで変えながらプロットすると結果は以下のようなになるだろう.

このような写像はロジスティック写像と呼ばれ, 非常に単純な式ながら一定の条件を満たすときにはカオスを生み出すことが知られている.

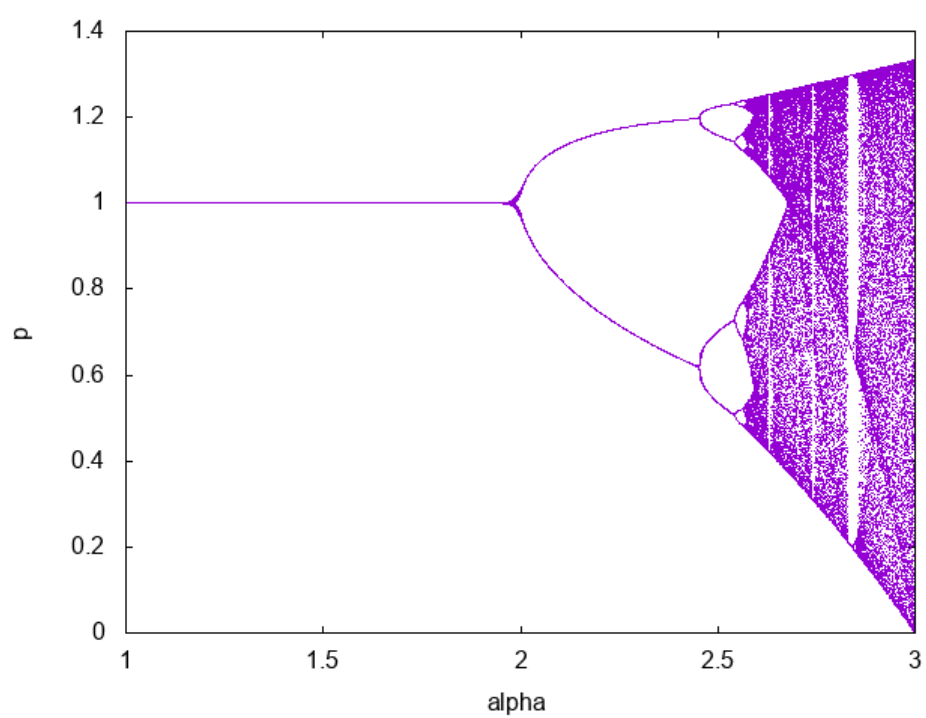


図 1: ロジスティック写像