

令和7年度

名古屋大学大学院情報学研究科  
知能システム学専攻  
入学試験問題 [専門]

令和6年8月7日

注意事項

Important Notices

1. 試験開始の合図があるまでは、この問題冊子を開いてはならない。  
Do not open this question booklet until the instruction to start the exam is given.
2. 試験終了まで退出できない。  
Do not leave the room until the end of the exam.
3. 外国人留学生の志願者は、日本語と日本語以外の1言語間の辞書1冊に限り使用してよい。  
電子辞書の使用は認めない。  
Foreign student applicants are allowed to use one dictionary between Japanese and one other language.  
Use of electronic dictionaries is not permitted.
4. 日本語または英語で解答すること。  
Answer in either Japanese or English.
5. 問題冊子、解答用紙3枚、草稿用紙3枚が配布されていることを確認すること。  
Confirm that you have received a question booklet, three answer sheets, and three draft sheets.
6. 問題は解析・線形代数、確率・統計、プログラミングの3科目がある。  
これらのすべてについて解答すること。解答する科目名が記載された解答用紙を使用すること。  
The exam questions cover three subjects: Calculus & Linear Algebra, Probability & Statistics, and Programming.  
Answer all of these subjects.  
Also, use the answer sheet that has the name of the corresponding subject written on it.
7. すべての解答用紙の所定の欄に受験番号を必ず記入すること。  
解答用紙に受験者の氏名を記入してはならない。  
Be sure to fill in your examinee number in the designated field in each answer sheet.  
Do not write your name on the answer sheets.
8. 解答用紙に書ききれない場合は、裏面を使用してもよい。  
ただし、裏面を使用した場合は、その旨、解答用紙表面右下に明記すること。  
If you cannot write all your answers on the answer sheet, you may use the back side.  
However, if you use the back side, clearly indicate this on the bottom right of the front side of the answer sheet.
9. 解答用紙は試験終了後に3枚とも提出すること。  
Submit all three answer sheets after the exam ends.
10. 問題冊子、草稿用紙は試験終了後に持ち帰ること。  
Take the question booklet and draft sheets with you after the exam.

# 解析・線形代数

(解の導出過程も書くこと)

[1] 次の微分方程式 (1) について、以下の問いに答えよ。

$$\frac{d^3x}{dt^3} - 3\frac{d^2x}{dt^2} - 6\frac{dx}{dt} + 8x = 0 \quad (1)$$

- (a)  $\mathbf{u}^T = \left(x, \frac{dx}{dt}, \frac{d^2x}{dt^2}\right)$  とすると、式 (1) は  $\frac{d\mathbf{u}}{dt} = \mathbf{A}\mathbf{u}$  と表せる。このとき、行列  $\mathbf{A}$  を求めよ。ただし、 $\mathbf{u}^T$  は  $\mathbf{u}$  の転置を表す。
- (b) 行列  $\mathbf{A}$  の固有値を求めよ。
- (c) 微分方程式 (1) を解け。

[2] 複素数 について、以下の問いに答えよ。

- (a) 複素数  $a_1 = \sqrt{3} + i$ ,  $a_2 = 2i$  に対して、 $\frac{a_1}{a_2}$  の絶対値と偏角を示せ。ただし、 $i$  は虚数単位を表す。
- (b) 3つの複素数  $z_1, z_2, z_3$  を複素平面上の点にそれぞれ対応させる。これらの点が正三角形をなすとき、次の式 (2) が成り立つことを示せ。

$$z_1^2 + z_2^2 + z_3^2 - z_1z_2 - z_2z_3 - z_3z_1 = 0 \quad (2)$$

[3] 次の立体  $V$  について、以下の問いに答えよ。

$$V = \left\{ (x, y, z) \in \mathbf{R}^3 \mid x^{\frac{2}{3}} + y^{\frac{2}{3}} + z^{\frac{2}{3}} \leq 1, x \geq 0, y \geq 0, z \geq 0 \right\}$$

$$\text{なお、} \int_0^{\frac{\pi}{2}} \sin^n x \, dx = \int_0^{\frac{\pi}{2}} \cos^n x \, dx = \begin{cases} \frac{(n-1)(n-3)\cdots 3 \cdot 1}{n(n-2)\cdots 4 \cdot 2} \cdot \frac{\pi}{2} & (n \geq 2 \text{ かつ } n \text{ は偶数}) \\ \frac{(n-1)(n-3)\cdots 4 \cdot 2}{n(n-2)\cdots 5 \cdot 3} & (n \geq 3 \text{ かつ } n \text{ は奇数}) \end{cases}$$

である。

- (a)  $V$  を平面  $z = k$  により切断したときの断面を  $S$  とする。 $S$  の式を  $x = r \cos^3 t$ ,  $y = r \sin^3 t$  で変数変換するとき、 $r$  と  $t$  のとりうる値の範囲を求めよ。
- (b)  $S$  の面積を求めよ。
- (c)  $V$  の体積を求めよ。

## Translation of technical terms

微分方程式	differential equation	式	formula	行列	matrix
転置	transpose	固有値	eigenvalue	複素数	complex number
絶対値	absolute value	偏角	argument	虚数単位	imaginary unit
複素平面	complex number plane	点	point	正三角形	regular triangle
立体	solid	偶数	even number	奇数	odd number
平面	plane	断面	section	変数変換	change of variable
値	value	範囲	range	面積	area
体積	volume				

## 確率・統計

解の導出過程も書くこと。

[1] 次のように作られたコインが1つずつある。

コイン A : 表おもてが出る確率かくりつが  $1/3$ , 裏うらが出る確率が  $2/3$ .

コイン B : 表が出る確率が  $2/3$ , 裏が出る確率が  $1/3$ .

以下の問いに既約分数きやくぶんすうで答えよ。

(a) コイン A を 5 回投げるとき, 表が 2 回だけ出る確率を求めよ。

(b) コイン A を 4 回投げるとき, 表が 2 回以上連続して出る確率を求めよ。

(c) コインを 1 つ無作為むきくゐに選んで, 4 回投げて 1 回だけ表が出た場合, このコインがコイン A である確率を求めよ。

[2] 確率変数  $X$  は次の確率密度関数かくりつみつどかんすう  $f(x)$  を持つとする。ただし,  $k$  は定数ていすうとする。

$$f(x) = \begin{cases} kx(2-x) & (0 \leq x \leq 2) \\ 0 & (\text{otherwise}) \end{cases}$$

以下の問いに答えよ。

(a) 定数  $k$  を求めよ。

(b) 累積分布関数るいせきぶんぶんかんすう  $F(x)$  を求めよ。

(c)  $F(x) = 1/2$  および  $F(x) = 5/32$  のとき, それぞれに対応する  $x$  の値を求めよ。

(d) 確率変数  $Y$  を  $Y = 2X + 1$  と定義する。  $Y$  の確率密度関数  $g(y)$  を求めよ。

## Translation of technical terms

表: head

裏: tail

無作為: random

確率密度関数: probability density function

累積分布関数: cumulative distribution function

確率: probability

既約分数: irreducible fraction

確率変数: random variable

定数: constant

# プログラミング

整数を要素とする行列を扱う Python プログラムについて考える。2 ページ以降に掲載のプログラムは、20～48 行目の関数 `proc` を再帰的に呼び出しながら処理を行う。プログラム中の変数  $A$  は行列を扱う変数であり、以降の設定では行列  $A$  と記載する。また、 $A$  は行数  $M$ 、列数  $N$  からなる行列とし、以下で示す数式においては  $i$  行  $j$  列の要素を  $A_{i,j}$  ( $1 \leq i \leq M$ ,  $1 \leq j \leq N$ ) のように同じアルファベットのイタリック体に添え字を組み合わせて表記する。以下の問いに答えよ。ただし、行列  $A$  の内容を答える際は `[]` を省略しても構わない。

- 21 行目のコードが初めて実行された際に標準出力に表示される行列  $A$  の内容を答えよ。
- 27 行目は、行列  $A$  に対して、 $i$  行と  $s$  行の要素をすべて入れ替える処理である。`ア`、`イ` に入れるべき適切なコードを答えよ。
- 28 行目は、行列  $A$  に対して、 $j$  列と  $s$  列の要素をすべて入れ替える処理である。`ウ`、`エ` に入れるべき適切なコードを答えよ。
- 31～32 行目は以下の処理をプログラムとして実装したものである。ここで、 $s$  と  $r$  はプログラム中の変数にそれぞれ対応する。また、 $\leftarrow$  は代入を表し、 $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す。 $(n = 1, \dots, N)$  は、 $n$  の値を 1 から  $N$  まで変化させて処理を行うことを表す。`オ`～`キ` に入れるべき適切なコードを答えよ。

$$k \leftarrow \left\lfloor \frac{A_{r,s}}{A_{s,s}} \right\rfloor$$
$$A_{r,n} \leftarrow A_{r,n} - A_{s,n} \times k \quad (n = 1, \dots, N)$$

- 35～36 行目は以下の処理をプログラムとして実装したものである。ここで、 $s$  と  $c$  はプログラム中の変数にそれぞれ対応する。`ク`～`コ` に入れるべき適切なコードを答えよ。

$$k \leftarrow \left\lfloor \frac{A_{s,c}}{A_{s,s}} \right\rfloor$$
$$A_{m,c} \leftarrow A_{m,c} - A_{s,c} \times k \quad (m = 1, \dots, M)$$

- 21 行目のコードが 2 度目に実行された際に標準出力に表示される行列  $A$  の内容を答えよ。
- プログラムが最後まで実行される間に標準出力に表示される行列  $A$  の内容をすべて答えよ。
- 38 行目に指定されている条件の `True` を次式に変更したい（式中の  $s$  はプログラム中の変数に対応する）。ただし、 $|x|$  は  $x$  の絶対値を表す。

$$\sum_{m=s+1}^M |A_{m,s}| + \sum_{n=s+1}^N |A_{s,n}| = 0$$

このとき、38 行目を以下のように変更するとすると、`サ` に入れるべき適切なコードを答えよ。

```
if サ:
```

また、変更後のプログラムを最後まで実行したときに標準出力に表示される行列  $A$  の内容をすべて答えよ。

## Translation of technical terms

整数 integer 行列 matrix 関数 function 再帰的 recursive  
変数 variable 行 row 列 column 条件 condition

## プログラム

```
1 import numpy as np
2
3 def search(A, s):
4     R = np.abs(A[s:,s:])
5     idx, rmin = (-1,-1), 1000000000
6     for r in range(R.shape[0]):
7         for c in range(R.shape[1]):
8             if R[r,c] > 0 and R[r,c] < rmin:
9                 idx, rmin = (s+r,s+c), R[r,c]
10    return idx
11
12 def check(A, s):
13    tmp = A[s+1:,s+1:]
14    for r in range(tmp.shape[0]):
15        for c in range(tmp.shape[1]):
16            if tmp[r, c] % A[s, s] != 0:
17                return (s+r+1, s+c+1, tmp[r, c] // A[s, s])
18    return (-1, 0, 0)
19
20 def proc(A, s):
21    print(A)
22
23    i, j = search(A, s)
24    if i < 0:
25        return A
26
27    A[  ] = A[  ]
28    A[  ] = A[  ]
29
30    for r in range(s+1, A.shape[0]):
31        k = 
32        A[  ] -= 
33
34    for c in range(s+1, A.shape[1]):
35        k = 
36        A[  ] -= 
37
38    if True:
39        rr, cc, qq = check(A, s)
40        if rr >= 0:
41            A[rr, :] += A[s, :]
42            A[:, cc] -= qq * A[:, s]
43    else:
```

```
44         if A[s, s] < 0:
45             A[s, :] *= -1
46         return proc(A, s+1)
47
48     return proc(A, s)
49
50 A = np.array([3, 1, 3, 1, -1, 3, 3, 3, 0], dtype=int).reshape(3,3)
51 A = proc(A, 0)
```