

# 情報科学演習レポート

実験課題名 L<sup>A</sup>T<sub>E</sub>X 実習課題

実験実施日 2025年9月28日, 11月30日

レポート提出日 2025年12月1日

学生番号 24516270

報告書提出者 上野能登

# 1 目的

本演習の目的は TeX のマクロパッケージである LATEX の仕組みと操作を演習課題を通して理解し、将来の卒業論文やレポート作成に活用できるスキルを身につけることである。

## 2 実験装置・環境

本演習でコンパイルに使用した環境は以下の通りである。ただし、LATEX ソースコードの執筆にはほかのコンピュータも使用した。

コンピュータ環境の構成を表 1 に、ソフトウェア環境の構成を表 2 に示す。

表 1: 本演習におけるハードウェア環境構成

項目	詳細
OS	Microsoft Windows 11 Home (64-bit)
CPU	AMD Ryzen 7 8845HS
GPU (単体)	NVIDIA GeForce RTX 4060 Laptop GPU
システムモデル	ASUS TUF Gaming A14 FA401UV_FA401UV
物理メモリ (RAM)	32 GB

表 2: 本演習におけるソフトウェア環境構成

項目	詳細
TeX ディストリビューション	<b>TeX Live 2025</b>
コンパイラ	pLaTeX (e-upTeX 3.141592653-p4.1.2-u2.00-250202-2.6)
TeX Live 2025 インストール元 (参考)	山形大学サーバー内のミラー
画像生成言語	<b>Python 3.13.9</b> (Windows App Store 版)
主要ライブラリ (NumPy)	<b>2.1.3</b>
主要ライブラリ (Matplotlib)	<b>3.10.0</b>

## 3 方法

### 3.1 実習 4.1

各記号の出力を確認するため 4\_1.tex に以下のコード：

-----

```
\documentclass[12pt,a4j]{jarticle}
```

```
\begin{document}
```

109 キーボードの第一列には数字が並んでいるが、shift キーを押しながらでは、順に、

```
\begin{center}
```

```

!!!!!"#$%&'(=)\textasciitilde\textbar
\end{center}
が入力される。(ちなみにこれは verbatim 環境を利用すると,
\begin{verbatim}
!!!!!"#$%&'(=)\~|
\end{verbatim}
という表示になる。)これが、101 キーボードでは,
\begin{center}
!!!!@\#"$%\textasciicircum\&*\(\)-+\textbar\textasciitilde
\end{center}
と異なる順に配置されている。
\end{document}

```

---

を入力した。その後、4\_1.tex をコンパイルし、その結果を画面上に表示させ、意図した結果が得られるか確認した。

## 3.2 実習 4.2

verbatim 環境や各箇条書きの環境、表組み環境の挙動の確認のため 4\_2.tex に以下のコード：

---

```

\documentclass[12pt,a4j]{jarticle}

\begin{document}
\section{実習 4.2}
\subsection{3.5 打ち込んだ通りに出力する}
\begin{verbatim}
#include<stdio.h>
int main()
{
    int i=2;
    printf("%d%d\n", i, i*i);
    return 0;
}
\end{verbatim}
ここで、\verb|\#include| はプリプロセッサへの命令で...また、\verb+%d+ で整数型の変数を表す。
\subsection{3.6 箇条書き}
上杉家の名君として次の二人が有名。
\begin{description}
\item[上杉謙信] 1530--1578 没。米沢上杉家の藩祖。名将と謳われる。

```

```
    \item[上杉鷹山] 1751--1822 没。第9代米沢藩主。中興の名君。  
  \end{description}
```

```
The latest CD by "PUR" consists of 4 songs with the title:  
 \begin{enumerate}  
   \item Ouverte "ure  
   \item Hold me tight  
   \item If you hear this Tango  
   \item Adventure land.  
 \end{enumerate}
```

山形大学米沢キャンパス組織図：

```
\begin{itemize}  
   \item 工学部（6学科）  
     \begin{itemize}  
       \item 情報・エレクトロニクス学科  
       \item システム創成工学科  
       \item (他4学科略)  
     \end{itemize}  
   \item 大学院理工学研究科（6専攻）  
     \begin{itemize}  
       \item 数理情報システム専攻  
       \item 情報・エレクトロニクス専攻  
       \item (他4専攻略)  
     \end{itemize}  
 \end{itemize}
```

```
\subsection{3.7 中央寄せとスペーシング}  
 強調のため、ある単語または文を中心に置くには  
 \begin{center}  
   center 環境  
 \end{center}  
 を用いる。
```

```
\subsection{3.8 表組み}  
 \begin{center}  
   \begin{tabular}{r|lc}  
     \hline  
     国名 & 首都 & 通貨 \\ \hline  
     日本 & 東京 & 円 \\ \hline  
     イギリス & ロンドン & ポンド \\ \hline  
     アルゼンチン & ブエノスアイレス & ペソ \\ \hline  
   \end{tabular}  
 \end{center}
```

```

\hline
\end{tabular}
\end{center}

\begin{table}[h]
\caption{各国の首都と通貨}
\begin{center}
\begin{tabular}{r|lc}
\hline
国名 & 首都 & 通貨 \\
\hline
日本 & 東京 & 円 \\
イギリス & ロンドン & ポンド \\
アルゼンチン & ブエノスアイレス & ペソ \\
\hline
\end{tabular}
\end{center}
\end{table}

\end{document}

-----
を入力した。その後、4_2.tex をコンパイルし、その結果を画面上に表示させ、意図した結果が得られるか確認した。

```

### 3.3 実習 4.3(a)

description 環境を用いた箇条書きの挙動を調べるために、4\_3\_1.tex に以下のコード：

```

\documentclass[12pt,a4j]{jarticle}

\begin{document}
\begin{description}
\item[cat](\textit{pl}.cats)
\begin{enumerate}
\item \textbf{A cat} is a small, furry animal with a tail, whiskers, and sharp claws.
      Cats are often kept as pets.
\item \textbf{Cats} are lions, tigers, and other wild animals in the same family.
\item \textbf{See also} \textbf{Cheshire cat}, \textbf{fatcat}, \textbf{wildcat}.
\end{enumerate}
\end{description}

```

```
\end{description}
\end{document}
```

-----  
を入力した。その後、4\_3\_1.tex をコンパイルし、その結果を画面上に表示させ、意図した結果が得られるか確認した。

### 3.4 実習 4.3(b)

verbatim 環境を用いて 4.3(a) の  $\text{\TeX}$  ソースコードをそのまま出力するため 4\_3\_2.tex に以下のコード：

```
\documentclass[12pt,a4j]{jarticle}
```

```
\begin{document}
\begin{verbatim}
\documentclass[12pt,a4j]{jarticle}

\begin{document}
\begin{description}
\item[cat] A cat is a small, furry animal with a tail, whiskers, and sharp claws. Cats are often kept as pets.
\item[See also] \textbf{Cheshire cat}, \textbf{fatcat}, \textbf{wildcat}.
\end{description}
\end{document}
\end{verbatim}
\end{document}
```

-----  
を入力した。その後、4\_3\_2.tex をコンパイルし、その結果を画面上に表示させ、意図した結果が得られることを確認した。

### 3.5 実習 8.1

表の中に数式を出力した際の挙動を確認するため,8\_1.tex に以下のコード：

```

\documentclass[12pt,a4j]{jarticle}

\begin{document}
\begin{table}
\caption{増減表}
\begin{center}
\begin{tabular}{c|c|c|c|c|c}
\hline
 $x$  & ... &  $\frac{\beta+1-\sqrt{\beta^2-\beta+1}}{3}$  & ... &  $\frac{\beta+1+\sqrt{\beta^2-\beta+1}}{3}$  & ... \\
\hline
 $f'(x)$  & + & 0 & - & 0 & + \\
\hline
 $f(x)$  & \nearrow & 極大値 & \searrow & 極小値 & \nearrow \\
\hline
\end{tabular}
\end{center}
\end{table}
\end{document}

```

-----  
を入力した。その後、8\_1.tex をコンパイルし、その結果を画面上に表示させ、意図した結果が得られるか確認した。

### 3.6 実習 8.2

各数式環境の挙動を確認するため、8\_2.tex に以下のコード：

```

\documentclass[12pt,a4j]{jarticle}

\begin{document}
\begin{equation}
B := \int \int_{[0, \infty)^2} e^{-(x^2+y^2)} dx dy \label{eq:xyint}
\end{equation}
式 (\ref{eq:xyint}) に極座標変換  $x=r\cos\theta$ ,  $y=r\sin\theta$  を代入すると、
\begin{eqnarray}
B &= \int_{\theta=0}^{\pi/2} \int_{r=0}^{\infty} e^{-r^2} r dr d\theta \nonumber \\
&= \frac{\pi}{2}
\end{eqnarray}

```

```

\uuuu&u=\frac{\pi}{2}\int_{0}^{\infty}
\uuuue^{-r^2}r\,dr\label{eq:rint}
\end{eqnarray}
となる. 式 (\ref{eq:rint}) の計算結果は
\[

B=\frac{\pi}{2}\int_0^\infty e^{-r^2}r\,dr=\frac{\pi}{4}
\]

であり, A=$\sqrt{B}$が満たされるので
\[

\frac{\sqrt{\pi}}{2}
\]

\end{document}

```

-----  
を入力した. その後, 8\_2.tex をコンパイルし, その結果を画面上に表示させ, 意図した結果が得られるか確認した.

### 3.7 実習 8.3

行列を表示させる際の挙動を確認するため, 8\_3.tex に以下のコード:

```

\documentclass[12pt,a4j]{jarticle}

\begin{document}
\[

\left(
\begin{array}{ccc}
\cos\theta\cos\zeta & -r\sin\theta\cos\zeta \\
-r\cos\theta\sin\zeta & \cos\theta\sin\zeta \\
\sin\theta\cos\zeta & r\sin\theta\cos\zeta \\
r\cos\theta\sin\zeta & \sin\theta\sin\zeta
\end{array}
\right)
\end{array}
\right)
\end{document}

```

-----  
を入力した. その後, 8\_3.tex をコンパイルし, その結果を画面上に表示させ, 意図した結果が得られるか確認した.

### 3.8 実習 8.4

複雑な数式や表を含んだ場合の TeX ソースコードと出力結果の関係を調べるために 8\_4.tex に以下のコード：

```
-----  
  
\documentclass[12pt,a4j]{jarticle}  
\usepackage[dvipdfmx]{graphicx}  
  
\begin{document}  
$z=x+yi$ を複素数とする。領域 $D$ は次の不等式：  
\[  
~~~~~D=\{z=x+yi \mid |z| \leq 1, |z-1| \geq 1,  
~~~~~|z+1| \geq 1, y > 0\}  
\]  
で定義されているとする。この時 $D$ の頂点を求めたい。  
$|z| = \sqrt{x^2+y^2}$ なので, $D$ の境界は  
\begin{eqnarray}  
~~~~~x^2+y^2=1 \label{eq:xyintA} \\  
~~~~~(x-1)^2+y^2=1 \label{eq:xyintB} \\  
~~~~~(x+1)^2+y^2=1 \label{eq:xyintC}  
\end{eqnarray}  
で与えられる。(\ref{eq:xyintA}) から (\ref{eq:xyintB}) を引くことにより,  
\[  
~~~~~2x+1=0  
\]  
を得る。すなわち $x=\frac{1}{2}$ となる。  
これを (\ref{eq:xyintA}) に代入すると  
\[  
~~~~~(\frac{1}{2})^2+y^2=\frac{1}{4}+y^2=1  
\]  
すなわち $y^2=\frac{3}{4}$ となる。$y>0$ なので,  
$y=\sqrt{\frac{3}{4}}$ を得る。同様に, (\ref{eq:xyintA}) から  
(\ref{eq:xyintC}) を引くことにより,  
\[  
~~~~~2x-1=0  
\]  
を得る。すなわち $x=-\frac{1}{2}$ となる。  
これを (\ref{eq:xyintA}) に代入して  
$\left(x, y\right)=\left(-\frac{1}{2}, \sqrt{\frac{3}{4}}\right)$ を得る。  
同様に, (\ref{eq:xyintB}) から (\ref{eq:xyintC}) を引くことにより,  
\[  
~~~~~4x=0
```

```
\]
を得る. これを (\ref{eq:xyintB}) に代入することにより,
 $(x, y) = (0, 0)$  となる.

以上により, 図1で与えられる$D$の各頂点は
 $x=0, \frac{1}{2} + \frac{\sqrt{3}}{2}i$ ,
 $-\frac{1}{2} + \frac{\sqrt{3}}{2}i$  の三点で与えられる.

\begin{figure}[h]
\centering
\includegraphics[width=6cm]{./pic.png}
\caption{D の領域}
\end{figure}
\end{document}
```

-----  
 を入力した. その後, 8\_4.tex をコンパイルし, その結果を画面上に表示させ, 意図した結果が得られるか確認した. ただし, 画像は circle.py に以下のコード:

```
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import numpy as np

# 図のセットアップ
fig, ax = plt.subplots(figsize=(6, 5))

# 1. 3つの円を描画
# 円の方程式: x^2+y^2=1, (x-1)^2+y^2=1, (x+1)^2+y^2=1
circle_center = patches.Circle((0, 0), 1, linewidth=1,
                               edgecolor='black',
                               facecolor='none', zorder=2)
circle_right = patches.Circle((1, 0), 1, linewidth=1,
                             edgecolor='black',
                             facecolor='none', zorder=2)
circle_left = patches.Circle((-1, 0), 1, linewidth=1,
                            edgecolor='black',
                            facecolor='none', zorder=2)

ax.add_patch(circle_center)
ax.add_patch(circle_right)
ax.add_patch(circle_left)

# 2. 領域 D (黒い部分) の塗りつぶし
# 3つの円の交点により x=-0.5から0.5まで
```

```

x=np.linspace(-0.5,0.5,500)

#上側の境界:中央の円
y_upper=np.sqrt(1-x**2)

#下側の境界:
#x<0:左の円の上側,x>=0:右の円の上側
y_lower=np.where(x<0,np.sqrt(1-(x+1)**2),
np.sqrt(1-(x-1)**2))

ax.fill_between(x,y_lower,y_upper,color='black',zorder=1)

#3.破線の描画(x=-0.5,x=0.5の位置)
y_intersect=np.sqrt(0.75)
ax.vlines(x=[-0.5,0.5],ymin=0,ymax=y_intersect,
colors='black',linestyles='dashed',
linewidth=1,zorder=2)

#4.軸と目盛りの設定
ax.spines['left'].set_position('zero')
ax.spines['bottom'].set_position('zero')
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')

ax.set_xlim(-2.5,2.5)
ax.set_ylim(-1.5,1.8)

xticks=[-2,-1,1,2]
yticks=[-1,1]
ax.set_xticks(xticks)
ax.set_yticks(yticks)
ax.set_xticklabels(xticks,fontsize=12)
ax.set_yticklabels(yticks,fontsize=12)

ax.tick_params(direction='inout',length=6,width=1,
colors='black')

ax.text(-0.2,-0.2,"0",fontsize=12,
ha='center',va='center')
ax.text(2.6,0,"x",fontsize=14,ha='left',va='center')
ax.text(0,1.9,"y",fontsize=14,ha='center',va='bottom')

ax.set_aspect('equal')

```

```

# 5. 軸の矢印を描画
ax.plot(1, 0, ">k", transform=ax.get_yaxis_transform(),
         clip_on=False)
ax.plot(0, 1, "^k", transform=ax.get_xaxis_transform(),
         clip_on=False)

plt.tight_layout()

# 画像保存
fig.savefig('pic.png',
             bbox_inches='tight', dpi=300)

```

-----  
を入力し, その後, インタプリタで実行し作成したものを使う.

## 4 演習結果

### 4.1 実習 4.1

「方法」で説明した実習 4.1 のソースコードをコンパイルすると, 画面上に以下のように表示された.

-----  
109 キーボードの第一列には数字が並んでいるが, shift キーを押しながらでは, 順に,

! " # \$ % & ' ( ) = ~ |

が入力される. (ちなみにこれは verbatim 環境を利用すると,

! " # \$ % & ' ( ) = ~ |

という表示になる.) これが, 101 キーボードでは,

! @ # \$ % ^& \* ( ) - + |~

と異なる順に配置されている. -----

# \$ % & など記号が TeX でもつ特別な意味ではなく, そのまま出力された.

### 4.2 実習 4.2

「方法」で説明した実習 4.2 のソースコードをコンパイルすると, 画面上に以下のように表示された.

### 3.5 打ち込んだ通りに出力する

```
#include <stdio.h>
int main()
{
    int i=2;
    printf("%d %d\n", i, i*i);
    return 0;
}
```

ここで, #include はプリプロセッサへの命令で... また, %d で整数型の変数を表す.

### 3.6 箇条書き

上杉家の名君として次の二人が有名.

上杉謙信 1530 – 1578 没. 米沢上杉家の藩祖. 名将と謳われる.

上杉鷹山 1751 – 1822 没. 第9代米沢藩主. 中興の名君.

The latest CD by "PUR" consists of 4 songs with the title:

1. Ouvert üre
2. Hold me tight
3. If you hear this Tango
4. Adventure land.

山形大学米沢キャンパス組織図:

- 工学部 (6 学科)
  - 情報・エレクトロニクス学科
  - システム創成工学科
  - (他4学科略)
- 大学院理工学研究科 (6 専攻)
  - 数理情報システム専攻
  - 情報・エレクトロニクス専攻
  - (他4専攻略)

### 3.7 中央寄せとスペーシング

強調のため, ある単語または文を中心に置くには

center 環境

を用いる.

### 3.8 表組み

国名	首都	通貨
日本	東京	円
イギリス	ロンドン	ポンド
アルゼンチン	ブエノスアイレス	ペソ

表 3: 各国の首都と通貨

国名	首都	通貨
日本	東京	円
イギリス	ロンドン	ポンド
アルゼンチン	ブエノスアイレス	ペソ

-----  
verbatim 環境では内部に入力した C のソースコードがそのまま出力された. また,description 環境を使うことで箇条書きと各項目の説明として出力された. enumerate 環境や itemize 環境でも箇条書きが出力された.tabular,table 環境では表が出力され,table 環境では表番号とキャプションも出力された.

### 4.3 実習 4.3(a)

「方法」で説明した実習 4.3(a) のソースコードをコンパイルすると, 画面上に以下のように表示された.

-----  
**cat** ( *pl .cats* )

1. **A cat** is a small, furry animal with a tail, whiskers, and sharp claws. Cats are often kept as pets.
2. **Cats** are lions, tigers, and other wild animals in the same family.
3. *See also Cheshire cat, fatcat, wildcat.*

-----  
description 環境によって各項目が箇条書きの形で出力され,\textbf を用いることで太字に,\textit を用いることで斜体として出力された.

### 4.4 実習 4.3(b)

「方法」で説明した実習 4.3(b) のソースコードをコンパイルすると, 画面上に以下のように表示された.

```

\documentclass[12pt,a4j]{jarticle}

\begin{document}
\begin{description}
\item[cat] ( \textit{pl} ) .cats
\begin{enumerate}
\item \textbf{A cat} is a small, furry animal with a tail, whiskers, and sharp claws. Cats are often kept as pets.
\item \textbf{Cats} are lions, tigers, and other wild animals in the same family.
\item \textit{See also} \textbf{Cheshire cat, fatcat, wildcat.}
\end{enumerate}
\end{description}
\end{document}
-----
```

verbatim 環境を用いることで,4.3(a) を出力するための T<sub>E</sub>X コードがそのまま出力された.

## 4.5 実習 8.1

「方法」で説明した実習 8.1 のソースコードをコンパイルすると, 画面上に以下のように表示された.

表 4: 増減表

$x$	...	$\frac{\beta+1-\sqrt{\beta^2-\beta+1}}{3}$	...	$\frac{\beta+1+\sqrt{\beta^2-\beta+1}}{3}$	...
$f'(x)$	+	0	-	0	+
$f(x)$	$\nearrow$	極大値	$\searrow$	極小値	$\nearrow$

-----  
表示結果より,tabular 環境と \$\$ を組合わせることで表内に数式が出力されることを確認した.

## 4.6 実習 8.2

「方法」で説明した実習 8.2 のソースコードをコンパイルすると, 画面上に以下のように表示された.

$$B := \int \int_{[0,\infty)^2} e^{-(x^2+y^2)} dx dy \quad (1)$$

式(1)に極座標変換  $x = r \cos \theta$ ,  $y = r \sin \theta$  を代入すると,

$$\begin{aligned} B &= \int_{r=0}^{\infty} \int_{\theta=0}^{\frac{\pi}{2}} e^{-r^2} r dr d\theta \\ &= \frac{\pi}{2} \int_0^{\infty} e^{-r^2} r dr \end{aligned} \quad (2)$$

となる. 式(2)の計算結果は

$$B = \frac{\pi}{2} \int_0^{\infty} e^{-r^2} r dr = \frac{\pi}{4}$$

であり,  $A = \sqrt{B}$  が満たされるので

$$\frac{\sqrt{\pi}}{2}$$

---

\[を用いた数式環境では数式が出力され, さらに eqnarray 環境, equation 環境では数式の番号が出力された.

## 4.7 実習 8.3

「方法」で説明した実習 8.3 のソースコードをコンパイルすると, 画面上に以下のように表示された.

---

$$\begin{pmatrix} \cos \theta \cos \zeta & -r \sin \theta \cos \zeta & -r \cos \theta \sin \zeta \\ \cos \theta \sin \zeta & -r \sin \theta \sin \zeta & r \cos \theta \cos \zeta \\ \sin \theta & r \cos \theta & 0 \end{pmatrix}$$


---

表示結果より, array 環境と伸縮するかっこを組合わせることで行列が出力されることを確認した.

## 4.8 実習 8.4

「方法」で説明した実習 8.4 のソースコードをコンパイルすると, 画面上に以下のように表示された.

---

$z = x + yi$  を複素数とする. 領域  $D$  は次の不等式:

$$D = \{z = x + yi \mid |z| \leq 1, |z - 1| \geq 1, |z + 1| \geq 1, y > 0\}$$

で定義されているとする. この時  $D$  の頂点を求めたい.  $|z| = \sqrt{x^2 + y^2}$  なので,  $D$  の境界は

$$x^2 + y^2 = 1 \quad (3)$$

$$(x - 1)^2 + y^2 = 1 \quad (4)$$

$$(x + 1)^2 + y^2 = 1 \quad (5)$$

で与えられる.(3)から(4)を引くことにより,

$$2x + 1 = 0$$

を得る. すなわち  $x = \frac{1}{2}$  となる. これを (3) に代入すると

$$\left(\frac{1}{2}\right)^2 + y^2 = \frac{1}{4} + y^2 = 1$$

すなわち  $y^2 = \frac{3}{4}$  となる.  $y > 0$  ので,  $y = \frac{\sqrt{3}}{2}$  を得る. 同様に,(3) から (5) を引くことにより,

$$2x + 1 = 0$$

を得る. すなわち  $x = -\frac{1}{2}$  となる. これを (3) に代入して  $(x, y) = \left(-\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$  を得る. 同様に,(4) から (5) を引くことにより,

$$4x = 0$$

を得る. これを (4) に代入することにより,  $(x, y) = (0, 0)$  となる. 以上により, 図 1 で与えられる  $D$  の各頂点は  $x = 0, \frac{1}{2} + \frac{\sqrt{3}}{2}i, -\frac{1}{2} + \frac{\sqrt{3}}{2}i$  の三点で与えられる.

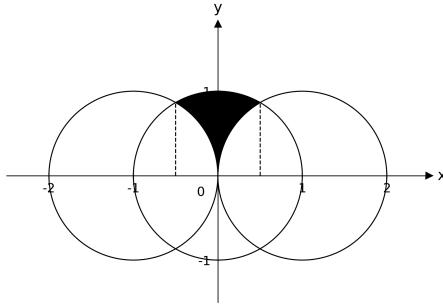


図 1:  $D$  の領域

---

以上の出力結果より, 数式環境を用いることで, 領域の定義や複雑な計算過程が意図通りに数式として出力されていることが確認できた. また, figure 環境を用いることで図表が挿入された形で表示された.

## 5 考察

本演習では, 基本的に想定通りの実行結果を得ることができた. 一方で表の作成においてキャプションの位置や配置指定 (`[h]` など) が意図通りに反映されない問題に直面した. そこで当初, tabular 環境を用いることで, 解決しようとしたが, `\caption` を入力したところエラーになってしまった. 調査の結果, tabular 環境はあくまで表の「格子」を作る機能であり, キャプションや配置場所を管理するのはその外側にある table 環境(浮動体)の役割であることが分かった. このことから, LaTeXにおいては「中身」と「配置枠」の役割分担が明確になされており, 適切な環境の入れ子構造を意識する必要があると考察できる. また, この構造を理解することは, 図 (figure 環境) を扱う際にも応用可能であると考えられる.

## 参考文献

- [1] 情報・エレクトロニクス学科（情報・知能コース）編, 「 $\text{\LaTeX}$  による文書作成入門」, [https://ecsylms1.kj.yamagata-u.ac.jp/webclass/course.php/2452513/manage/?acs\\_=7fdb2584](https://ecsylms1.kj.yamagata-u.ac.jp/webclass/course.php/2452513/manage/?acs_=7fdb2584), 2025.