

コンピュータアニメーション特論

LaTeX の利用方法

九州工業大学 情報工学研究院 尾下真樹

1 はじめに

本資料では、LaTeX を使った文書作成方法を説明する。

LaTeX は文書作成システムの一つである。Microsoft Word などの他の多くの文書作成システムは、最終的に見える形で文書の編集を行う WYSIWYG (What You See Is What You Get) 方式のシステムとなっているが、LaTeX は、テキスト形式で文書のソースファイルを記述し、コンパイルを行うことで最終的に見える形の文書を生成する方式となっている。LaTeX には、最終的な文書を得るためにコンパイルの作業が必要があるが、文書の書式を気にせずに内容のみを記述することができ、全体で書式が統一された文書を簡単に作成できる、といったメリットがある。LaTeX は、このような特徴から、特に学术论文や技術文書の作成に広く使われている。授業のレポートや卒業論文・修士論文の作成にも使用する場合が多いので、LaTeX の使い方に慣れておくことが望ましい。

2 LaTeX の環境設定

LaTeX のソースファイルは、任意のテキストエディタを使って編集できるが、ソースファイルのコンパイルを行って文書を生成するためには、LaTeX のシステムをインストールする必要がある。以下、複数の利用方法を説明する。

2.1 Ubuntu 仮想環境での LaTeX の利用

九州工業大学情報基盤センターが提供する BYOD 端末用の Ubuntu 仮想環境には、LaTeX を利用するための環境が設定されているため、追加の作業を行わなくとも LaTeX を利用できる。

情報基盤センター ISC オンラインガイド: <https://onlineguide.isc.kyutech.ac.jp/guide2020/>

2023 年度以降の入学者は、九州工業大学情報基盤センターより、Windows の WSL を使った Ubuntu 仮想環境の利用方法が案内されているので、案内に従って環境設定を行う。

2022 年度以前の入学者は、VirtualBox を使った Ubuntu 仮想環境の利用方法が案内されているので、案内に従って環境設定を行う。2023 年度以降に新たに設定を行う場合は、上記の Windows の WSL を使った Ubuntu 仮想環境の設定を行っても構わない。

Ubuntu 仮想環境の設定や利用に関する問題は、授業の範囲外であるため、何らかの問題が生じた場合は、情報基盤センターの窓口にご相談して解決する。

2.2 Windows や Mac のネイティブ環境での LaTeX の利用

Windows や Mac のネイティブ環境で利用できる LaTeX システムを利用することもできる。現在は、TeX Live というディストリビューションが広く使われている。TeX Live をインストールすることで、ネイティブ環境で LaTeX を利用できる。

TeX Live の解説: <https://texwiki.texjp.org/?TeX%20Live>

TeX Live の設定や利用に関する問題は、授業の範囲外であり、情報基盤センターでも対応してもらうことは難しいため、問題が生じた場合には自力で解決する必要がある。自力で設定や利用を行うことが難しい場合は、上記の Ubuntu 仮想環境を利用することを推奨する。

2.3 オンラインでの LaTeX の利用

自分が使用するコンピュータに LaTeX のシステムをインストールしなくとも、ネットワークを通じて外部サーバの LaTeX システムを利用できる、OverLeaf のようなサービスもある。こちら、利用したい場合は、自身の責任で利用する。

OverLeaf: <https://ja.overleaf.com/>

3 LaTeX のコンパイル方法

Ubuntu 仮想環境や、Windows や Mac のネイティブ環境にインストールされている LaTeX 環境で、LaTeX ソースファイルをコンパイルする方法を説明する。ターミナルを開いて、以下のようなコマンドを実行することで、LaTeX ソースファイルをコンパイルして、PDF 形式の文書ファイルを作成できる。

1. ソースファイル (.tex) をコンパイルして dvi ファイルを生成する。

```
platex ファイル名.tex
```

2. dvi ファイルを PDF に変換する。

```
dvipdfmx ファイル名.dvi
```

ソースファイルから画像やプログラムリストを読み込んでいる場合は、(読み込み時にファイルが置かれているパスを設定していない場合は) ソースファイルと同じディレクトリにこれらのファイルが置かれている必要がある。

また、文書中にプログラムリストを挿入するために listing パッケージや jlisting パッケージを使用する場合、jlisting パッケージは標準的な LaTeX 環境には含まれないため、jlisting.sty ファイルを LaTeX のシステムディレクトリに追加するか、ソースファイルと同じディレクトリに置く必要がある。

何らかの原因でコンパイルが完了できなかった場合は、エラーの説明やエラーが発生した位置が表示されるので、これらの情報をもとに修正を行う。

4 LaTeX のソースファイルの書き方

LaTeX のソースファイルの基本的な書き方を説明する。

授業のレポートや卒業論文・修士論文では、元になるテンプレートのソースファイルが用意されているので、ソースファイルを見ながら書き方を理解して、内容を追加していくと良い。

4.1 全体の構成

LaTeX ソースファイルの全体の構成としては、冒頭に、文書の種類の定義、使用するパッケージの制限、タイトル・著者等の情報を記述した後に、本文が、`\begin{document}` ~ `\end{document}` に囲まれた中に記述する。

4.2 本文の記述方法

本文では、基本的には、ソースファイルに記述したテキストが、そのまま本文として出力される。

ただし、ソースファイル中のインデントや改行は無視されるので、改行を行いたい場合は、`\\` を付けるか、一行空行を空ける必要がある。

アンダーバー (`_`) やダブルクォート (`"`) や中括弧 (`{ }`) などの特別な記号は、そのまま記述しても表示されない (コンパイルエラーが出る) ので、各記号の前に `\` を付ける。

なお、環境によっては、`\` (バックスラッシュ) は、`¥` (円マーク) として表示されるので、注意する。

4.3 章・節・小節

`\chapter{見出し}`, `\section{見出し}`, `\subsection{見出し}` を使って、章・節・小節を分けることができる。`\newpage` コマンドを使うと、改ページを行える。各章の前には、自動的に改ページが行われる。

4.4 箇条書き

`\begin { itemize } ~ \end { itemize }` の内側に、各項目を `\item ~` として記述することで、箇条書きにすることができる。箇条書きの内側に、さらに別の箇条書きを入れることで、階層化することもできる。

`\begin { enumerate } ~ \end { enumerate }` を使うと、番号付きの箇条書きになる。

`\begin { description } ~ \end { description }` を使い、各項目を `\item [見出し] ~` として記述することで、各見出しを説明するような、見出し付き箇条書きになる。

4.5 図の挿入

`\begin { figure } ~ \end { figure }` や `\includegraphics [オプション] { 画像ファイル名 }` を使用することで、画像を挿入できる。画像ファイルの形式は、EPS 形式にする。(PNG 形式などの他の形式の画像も読み込むことができるが、少し難しいため、EPS 形式を推奨する。) また、オプションで図の大きさを指定できる。図の大きさは、絶対的な大きさ (cm など) や、ページの表示幅に対する相対的な比率 (ページの表示幅の 0.5 倍など) で指定することができる。

`\caption { キャプション }` を使うことで、図のキャプションを設定できる。図が何を表しているのかが分かるように、適切なキャプションを設定する。

`\label { ラベル文字列 }` を使うことで、図を本文から参照できるようになる。本文中に `\ref { ラベル文字列 }` を記述すると、コンパイル時にその図の図番号に置き換えて出力される。全体で図を追加・削除して図番号が変わったときにも、正しい図番号で参照することができるため、図を参照するときにはラベルを使う方が良い。ただし、参照する全ての図に対して、個別のラベル文字列を設定する必要がある。

`\begin { center } ~ \end { center }` を使うことで図のセンタリングを行える。

`\fbox { ~ }` を使うことで図に枠を付けて表示することができる。

LaTeX では、図は自動的に配置されるため、図の細かい配置を気にする必要はない。図が後方に配置されて気になる場合は、`\clearpage` コマンドを使用すると、コマンドよりも上の図を全て出力した上で改ページを行うことができる。

4.6 プログラムリストの挿入

`listings` パッケージの機能を使うことで、プログラムリストなどのテキストファイルを挿入できる。本文中にプログラムリストを直接記述する書き方では、インデントや改行が無視されたり、記号が正しく表示されなかったりして、単純にプログラムリストを示すことは難しい。`listings` パッケージを使うことで、プログラムリストをその

まま挿入することができる。また、`jlisting` パッケージを使用することで、日本語のコメントを含むプログラムリストにも対応できる。

`listings`, `jlisting` パッケージを使用する場合は、ソースファイルの冒頭で、`\usepackage { listings,jlisting }` のように記述して、これらのパッケージの読み込みを行う。

プログラムリストを挿入する場所に、`\begin { lstlisting } [caption=キャプション] ~ \end { lstlisting }` の形式で、プログラムリストを記述する。

もしくは、`\lstinputlisting[caption=キャプション] { ファイル名 }` の形式で、同じディレクトリにある別のファイルを挿入することもできる。既存のプログラムのソースファイル全体を挿入する場合は、こちらの方法を使う方が良い。

4.7 整形済みテキストの挿入

SQL の実行結果などの整形済みテキストを本文中に含めるときには、`\begin { verbatim } ~ \end { verbatim }` を使用する。ただし、`verbatim` 環境を使うと、文字数が多い行はページの横幅を超えてしまうため、その場合、手作業で適当に改行を挿入してページ内に収まるように調整する。

4.8 参照

本文から、図やプログラムリストや章・節・小節を参照するときには、ラベルを使用することで、自動的に正しい番号を挿入することができる。

図の場合は `\{ figure } ~ \end { figure }` の内側、章・節・小節の場合は見出しの直後に、`\label { ラベル名 }` を追加することで、本文中で `\ref { ラベル名 }` とすると、そのラベルに対応する番号を自動的に挿入することができる。

プログラムリストの場合は、`\lstinputlisting[caption=キャプション,label=ラベル名]{ ファイル名 }` のようにして、ラベルを設定する。ラベル名には、参照する各要素を区別できるように、各要素に異なる文字列を設定する必要がある。

5 よくあるエラーと対処方法

LaTeX のコンパイル時にエラーが表示され、コンパイルを行うことができなかった場合は、エラーメッセージの説明を参照して、ソースファイルの問題を修正する。よくある間違いと対処方法は、以下の通り。

コンパイルに必要なファイルを読み込むことができない。

上記の「LaTeX コンパイル方法」の説明の通り、ソースファイルから参照している追加パッケージ、図、プログラムリストなどの全てのファイルを、ソースファイルと同じディレクトリに置く必要がある。図やプログラムリストなどへの参照を追加した場合は、それらのファイル名が正しいことを確認する。

本文中に `_` などの記号が含まれている。

`_` のような特殊文字は、LaTeX のコマンドの一部として扱われるため、そのまま本文中に記述することはできない。ソースファイル名を記述する場合など、本文中に `_` を使用したい場合は、前に `_` 記号を付けて、`_` のように記述する。

図やソースファイルを参照しているときに、参照番号が正しく更新されない。

参照番号が正しく反映されるようにするためには、複数回のコンパイルを行う必要がある。また、複数の図やソースファイルに同じラベル名を設定していると、正しい参照番号を表示することができないため、全ての図やソースファイルには、独立したラベル名を設定する必要がある。