

SSI-MT000000

修士論文

システム情報科学コース修士論文スタイルファイルの使い方

山下 裕



HOKKAIDO
UNIVERSITY

2020年12月

北海道大学 大学院情報科学院
システム情報科学コース

本論文は北海道大学大学院情報科学院に
修士 (情報科学) 授与の要件として提出した修士論文である。

山下 裕

審査委員： 主査 A 教授
副査 B 教授
C 教授
D 准教授

Copyright © 2023 by Yuh Yamashita. All rights reserved.

システム情報科学コース修士論文スタイルファイルの使い方*

山下 裕

概要

これは、システム情報科学コースにおける修士論文用の \LaTeX スタイルファイルの使い方を説明したものである。

詳しくはこのサンプルのソースを見てほしい。

キーワード: \LaTeX , 修士論文, スタイルファイル

* 北海道大学 大学院情報科学院 システム情報科学コース 修士論文, SSI-MT000000, 2020 年 12 月 6 日.

Usage of the style file for master's thesis for Course of Systems Science and Informatics[†]

Yuh Yamashita

Abstract

This document describes the usage of the style file for master's thesis of Course of Systems Science and Informatics.

Read the source file of this document to understand the exact usage.

Keywords: \LaTeX , master's thesis, style file

[†] Master's Thesis, Course of Systems Science and Informatics, Graduate School of Information Science and Technology, Hokkaido University, SSI-MT000000, December 6, 2020.

目次

第 1 章	はじめに	1
第 2 章	設定部の書き方	3
2.1	本当の冒頭部分	3
2.2	プレアンブル	3
2.2.1	修士論文・博士論文の指定	5
2.2.2	論文タイトル	5
2.2.3	著者名	5
2.2.4	概要	6
2.2.5	学生番号	6
2.2.6	主査・副査	6
2.2.7	論文を最終提出した日付	6
2.2.8	著作権情報	6
2.2.9	その他	7
2.3	本文の始め	7
第 3 章	前提とする L ^A T _E X システム	9
3.1	大前提	9
3.2	日本語多ウェイトフォントの入手	10
3.3	許可される組み合わせについて	11
3.4	各処理系毎の最初の書き方	11
3.4.1	LuaL ^A T _E X+ ltjsbook	11
3.4.2	X _Y L ^A T _E X+ bxjsbook	12
3.4.3	upL ^A T _E X+ jsbook	13
3.4.4	(pdf)L ^A T _E X+ book	13
3.4.5	LuaL ^A T _E X+ book	14
3.5	アクセサン・ウムラウト他	15
3.6	ff, fi, fl, ffi, fll の合字	15
3.7	ギリシャ文字・キリル文字について	16

3.8	欧文文字領域のフォントについて	17
3.9	Unicode フォントの英語と日本語の切り替えについて	19
3.10	中国語などの表示について	20
3.11	ssithesis.sty のオプションについて	21
3.12	TeXLive のバージョンについて	21
第 4 章	その他	25
4.1	長い章・節タイトル	25
4.2	図について	25
4.3	数式について	26
4.3.1	align 環境	26
4.3.2	align*環境	27
4.3.3	equation + aligned 環境	27
4.3.4	align + split 環境	28
4.4	Times 系フォントの使い方	28
4.4.1	Lua \LaTeX の場合	28
4.4.2	X \LaTeX の場合	29
4.4.3	up \LaTeX , pdf \LaTeX の場合	29
4.5	Lua \LaTeX + unicode-math	29
4.6	定理環境	32
4.7	目次のレベル	33
4.8	参考文献について	33
第 5 章	英語論文	35
5.1	英語論文を書くには	35
5.2	プレアンブル	36
5.3	本文	37
5.4	参考文献の表題	37
	謝辞	39
	参考文献	41
付録 A	Appendix について	43
A.1	付録セクション	43

図目次

4.1	図を取り込んだ例	26
-----	--------------------	----

表 目 次

2.1	プリアンブルのコマンド一覧	4
3.1	オプション一覧	23
4.1	Unicode Math でのフォント指定 (<code>math-style=TeX,bold-style=TeX</code> の場合)	31
4.2	非 Unicode-Math でのフォント指定	32
4.3	定理環境一覧	33

第 1 章

はじめに

これは、システム情報科学専攻における修士論文用の \LaTeX スタイルファイルの使い方を説明したものである。未完成なところもあるが、将来的にはアブストラクトの `html` ファイルを自動生成するようにしたいと思っている。

このスタイルファイルは `book` 系列のクラスファイル — `jsbook.cls` (`up\LaTeX`) / `ltjsbook.cls` (`Lua\LaTeX`) / `bxjsbook.cls` (`Xq\LaTeX`) / `book.cls` (`pdf\LaTeX`/`Lua\LaTeX`; 英語論文) / `scrbook.cls` (`pdf\LaTeX`/`Lua\LaTeX`; 英語論文) — と共に使うことを前提としている。そのため、基本的には両面印刷で、必ず章は奇数ページから始まる。また、簡易製本機などの使用を前提としており、綴じる側のマージンが開く側のマージンよりも小さく取ってある。穴を開けて綴じる場合などは、`\oddsidemargin`, `\evensidemargin` などを調整するか、`equalmargin` オプションを指定して欲しい。

第 2 章

設定部の書き方

2.1 本当の冒頭部分

これに関しては、第 3 章にまとめてある。たとえば、TeXLive 2020/2021 で Lua^AT_EX を使う場合は

```
\documentclass[a4paper,11pt]{ltjsbook}
\def\logofilename{HuLogo-6.pdf}
\usepackage[colorlinks,hulogo]{ssithesis}
\usepackage[deluxe,haranoaji]{luatexja-preset}
```

等と書く。使う処理系やインストールしている日本語フォントによってここは大きく変わる可能性がある。

2.2 プリアンブル

以下は、このファイルのプリアンブルの一部を抜き出したものである。「本当の冒頭部分」に続く部分である。

```
\thesistype{\mastersthesis}
\title{システム情報科学コース修士論文スタイルファイルの使い方}
\etitle{Usage of the style file for master's thesis
for Course of Systems Science and Informatics}
\author{山下 裕}
\eauthor{Yuh Yamashita}
\jabstract{これは、システム情報科学コースにおける修士論文用の
\LaTeX スタイルファイルの使い方を説明したものである。 \par
詳しくはこのサンプルのソースを見てほしい。 }
\eabstract{This document describes the usage of the style file
```

表 2.1 プリアンブルのコマンド一覧

コマンド	意味
<code>\thesistype</code>	修士論文か博士論文か
<code>\title</code>	論文日本語タイトル
<code>\etitle</code>	論文英語タイトル
<code>\author</code>	日本語著者名
<code>\eauthor</code>	英語著者名
<code>\jabstract</code>	日本語アブストラクト
<code>\eabstract</code>	英語アブストラクト
<code>\jkeyword</code>	日本語キーワード列
<code>\ekeyword</code>	英語キーワード列
<code>\studentno</code>	学生番号
<code>\committeemember</code>	主査・副査
<code>\writtenyear</code>	提出年
<code>\writtenmonth</code>	提出月
<code>\writtenday</code>	提出日
<code>\thesiscopyright</code>	著作権情報

```

for master's thesis of Course of Systems Science and Informatics.\par
Read the source file of this document
to understand the exact usage.}
\jkeyword{\LaTeX, 修士論文, スタイルファイル}
\ekeyword{\LaTeX, master's thesis, style file}
\studentno{000000}
\committeemember{A 教授}{1}
\committeemember{B 教授}{2}
\committeemember{C 教授}{}
\committeemember{D 准教授}{}
\writtenyear{2020}
\writtenmonth{12}
\writtenday{6}
\thesiscopyright{Copyright © 2020 by Yuh Yamashita.
All rights reserved.}

```

以下、順番に説明する。

2.2.1 修士論文・博士論文の指定

`\thesistype{\masterthesis}`で修士論文を指定したことになる。博士論文の場合は、`\thesistype{\doctoralthesis}`のように書く。デフォルトで「修士(情報科学)」の修士論文である。たとえば、`\def\typeofdegree{工学}`とすれば「修士(工学)」になる。英語論文で「修士(工学)」の場合は、`\def\etypodegree{Engineering}`とする。なお、博士の場合は、「博士(情報科学)」であろうが「博士(工学)」であろうが、北海道大学での英語名は“Doctor of Philosophy”に固定されている(学位記の英語版ではそうなっている)。したがって、英語博士論文の場合は`\def\etypodegree`は指定しないほうがよい。

なお、日本語論文で表紙見返しに印字される「本論文は北海道大学…」の最後の句点を「。」から「.」に変えたければ、プレアンプルの`\thesistype{...}`の直前に

```
\makeatletter
\def\my@touten{.}
\makeatother
```

を挿入すること。

「論文」英語の英語表記は、修士論文は“Thesis”、博士論文は“Dissertation”になる。これを変えたければ、`\thesistype{...}`の後に、`\def\thesisordissertation{Thesis}`と書くことにより、強制的に“Thesis”にすることが可能である。ちなみに、アメリカでは博士論文は“Doctoral Dissertation”で、“Thesis”はそれよりも短い予備的研究を記述したものを意味する。一方、ヨーロッパでは逆で、博士論文は“Doctoral Thesis”で、“Dissertation”は修士論文に使われる。

2.2.2 論文タイトル

論文のタイトルを`\title{...}`のように書く。また、英語の論文タイトルを`\etitle{...}`のように書く。英語タイトルは英語アブストラクトに使われる。英語論文では、`\title{...}`は無視される。

2.2.3 著者名

日本語の著者名を`\author{...}`のように、英語の著者名を`\eauthor{...}`のように書く。英語論文では、`\author{...}`は無視される。

2.2.4 概要

日本語アブストラクトと英語のアブストラクトを、`\jabstract{…}`、`\eabstract{…}`のように書く。また、日本語キーワードと英語キーワードを、`\jkeyword{…}`、`\ekeyword{…}`のように書く。英語論文では、`\jabstract{…}`と`\jkeyword{…}`は無視される。日本語論文のクラスファイルはjsbook系列のものを使うので、その場合`\begin{abstract}…\end{abstract}`は`\chapter{…}`の直後に置いて各章の概要を示すのに使われることに注意。

2.2.5 学生番号

Paper ID を生成するのに必要。`\studentno{…}`のように学生番号を記述する。

2.2.6 主査・副査

主査・副査の先生の名前・肩書きを、`\committemember{氏名・肩書}{1 or 2 or 空白}`のように書く。英語論文では、先生の名前と肩書きも英語にすること。2番目の引数は、「1ならば主査, 2ならば副査, 空白なら何もつかない」を意味する。英語の「副査」の訳語が不明だったので、英語論文では副査はなにもつかない(主査とそれ以外の一般委員という扱い)。

2.2.7 論文を最終提出した日付

論文を最終提出した日付を、

```
\writtenyear{2021}
\writtenmonth{2}
\writtenday{3}
```

のように書く。

2.2.8 著作権情報

著作権表示を`\thesiscopyright{…}`で入力すると表紙見返し下部に印字される。省略すると、著作権表示は印字されない。著作権表示を希望するならば、“Copyright”の文字の有無よりも“©”が書かれているかどうか重要で、著作権者と著作権が発生する年を明記しなくてはならない。

2.2.9 その他

本文の偶数ページの左上には、論文のタイトルが書かれる。もし、論文タイトルが長すぎたり、`\title{ }` (英語論文の場合は`\etitle{ }`) で指定したものに改行が入っている場合は、プリアンブルに、

```
\def\titleshort{代わりの論文タイトル}
```

を入れる。これにより、本文の偶数ページの左上に代替文字列が表示される。また、旧組織 (研究科/専攻) の場合は、

```
\oldorganization
```

をプリアンブルに入れる。

2.3 本文の始め

本原稿の、`\begin{document}`以降の頭の部分は、

```
\frontmatter  
\maketitle  
\tableofcontents  
\listoffigures  
\listoftables  
\mainmatter
```

となっている。わからなければ、このまま書き写すのがよいだろう。もし、図が無ければ`\listoffigures` は不要である。また、表が無ければ`\listoftables` は不要である。

第 3 章

前提とする L^AT_EX システム

3.1 大前提

このスタイルファイルは、最近の L^AT_EX システムの発展に合わせて、古い環境での実行を意図的に排除している。以下にあげた項目は、その一部である。

1. e-T_EX 拡張必須。これは L^AT_EX2e 2017/01/01 以降なら満たされているはず。
2. Unicode (UTF-8 BOM 無し) 以外のエンコーディングは不可。最近の L^AT_EX では、西欧文字の場合でも Latin-1 (ISO/IEC 8859-1) のコードではなく、utf-8 前提となった。これに従い、Shift JIS などは排除される。そもそも、JIS 第 3, 第 4 水準漢字 (JIS の第 2 面) が (IBM 拡張領域に無理やり割り付けたものを除き) 表現できないこともあり、Shift JIS は使われなくなっている*¹。欧米でも、L^AT_EX とは全く別の動機 (Emoji 🌸 を使いたい!) で utf-8 化が進んでいた。TeX Live 2018 以降か否か、が utf-8 化されているかどうかのボーダーラインだと思われる。
3. 最近の pL^AT_EX でも unicode が一部使えるようにはしてあるが、BMP 内の文字しか使うことができず、また様々な問題があるので [3]、pL^AT_EX (≠ upL^AT_EX) は徐々に使われなくなってきた。その代わりに、(日本語文字に限り) BMP 外の unicode を使うことができる upL^AT_EX が主流となってきている。さらに、英語版の L^AT_EX は pdfL^AT_EX が前提となっていて .dvi ファイルを生成しないのがデフォルトである。それに合わせて日本語でも LuaL^AT_EX や XeL^AT_EX を使うようになってきた。LuaL^AT_EX+LuaT_EX-ja は、次期日本語 L^AT_EX の本命とされているので、本スタイルファイルでもこれを推薦環境とする。しかし、LuaT_EX-ja は「とにかく重い」+「日本語フォントによってはメモリの関係で 64bit 処理系が必要」なので、XeL^AT_EX や upL^AT_EX も本スタイルファイルでは使えるようになってきている。特に、Overleaf で使う場合は LuaT_EX-ja はタイムオーバーが続出し使い物にならない。

*¹ Shift_JIS-2004 という規格があるが、Windows で使われる cp932 の“NEC 特殊文字”、“IBM 拡張文字”とコード領域が衝突してしまっている。そのため、実際に Shift_JIS-2004 を採用しているシステムは見当たらないし、IANA の Character sets にも登録されていない。

4. 古い日本語のスタイルファイルは色々な問題があるので [3]、日本語論文では (lt/bx)js-book.cls を使うようにした。
5. 基本的には日本語フォントは**多ウェイトフォント**を使うようにした。**明朝体の太文字**と**ゴシック体**は別物で、**ゴシック体の太文字**も可能になっている。正常ならば、`\sffamily` や `\textsf{...}` でゴシックになるが、後述の TeXLive 2020 初期バージョンでは `\gtfamily` も加えたほうが良い。

3.2 日本語多ウェイトフォントの入手

TeXLive 2020 では原ノ味フォントを使うので、それはそのまま使えば良い。Mac ではヒラギノを使えば良い。ヒラギノを使う方法は web 上にたくさん載っているのを参考にされたい (TeXLive 2020 ならそれ向けの記事を参照する事)。Windows/Linux + TeXLive 2019 以前では、フォントのライセンスの関係もあるので、Google Noto フォントを使うのが良いと思われる。Google Noto フォントのサイト [4] に行き、“Noto Sans CJK JP” と “Noto Serif CJK JP” をダウンロード・展開し、otf ファイルを選択・右クリック・全てのユーザーに対してインストール、を選べばよい (Windows の場合)。“CJK JP” ではなく単に “Noto Sans/Serif JP” となっているフォントもあるが、それは**サブセットなのでインストールしないこと**。ただし、Noto フォントは Adobe-Identity-0 (AI0) という特殊なエンコーディングを採用していて、Adobe-Japan-1-{1...7}ではない。よって、やはりお勧めは原ノ味フォントということになる。

TeXLive で日本語フォントを選ぶ方法は、LuaTeX-ja や bxjsbook では L^AT_EX のソースファイル中に記述し、upL^AT_EX では `kanji-config-updmap-sys` コマンドを使う。

```
C:> kanji-config-updmap-sys status
Standby family : bizud
Standby family : haranoaji
Standby family : ipa
Standby family : ipaex
Standby family : kozuka
Standby family : ms
Standby family : noto (AI0)
Standby family : yu-win10
C:> kanji-config-updmap-sys --jis2004 haranoaji
```

`kanji-config-updmap-user` の設定の方が優先されるので、もし上記の設定が反映されていない場合は、`-sys` を `-user` に変えて実行してみると良い。

3.3 許可される組み合わせについて

本スタイルファイルで許可される、L^AT_EX の処理系とメインのクラスファイルの組み合わせは以下のとおりである。

LuaL^AT_EX+ ltjsbook 今、見ているファイルはこれで作成されている。高速な PC を持っているならこれがお勧め。

X_YL^AT_EX+ bxjsbook 次期 L^AT_EX システムの主導権は LuaL^AT_EX に移行してしまったが、日本語環境に限って言えば、これも有力である。

upL^AT_EX+ jsbook いったん dvi ファイルを作成する方法。otf ライブラリを読み込んでいるので、これでも多ウェイトフォントを扱うことはできる。

(pdf)L^AT_EX+ book 英語論文のみ。生の L^AT_EX を使う理由はないので、pdf_latex コマンドの方を使うべき。

(pdf)L^AT_EX+ scrbook 英語論文のみ。book.cls の代わりに、KOMA script の scrbook.cls を使うこともできる。生の L^AT_EX を使う理由はないので、pdf_latex コマンドの方を使うべき。

LuaL^AT_EX+ book 英語論文のみ。

LuaL^AT_EX+ scrbook 英語論文のみ。book.cls の代わりに、KOMA script の scrbook.cls を使うこともできる。

これまでの pL^AT_EX をやめる頃合いなので、pL^AT_EX はサポート範囲外とした。

3.4 各処理系毎の最初の書き方

3.4.1 LuaL^AT_EX+ ltjsbook

Noto フォントを使った場合のファイルの最初の部分は、以下のようになる。

```
\documentclass[a4paper,11pt]{ltjsbook}
\def\logofilename{HuLogo-6.pdf}
\usepackage[colorlinks,hulogo]{ssithesis}
\usepackage[deluxe,noto-otf]{luatexja-preset}
```

原ノ味フォントを使うのであれば、

```
\documentclass[a4paper,11pt]{ltjsbook}
\def\logofilename{HuLogo-6.pdf}
\usepackage[colorlinks,hulogo]{ssithesis}
\usepackage[deluxe,haranoaji]{luatexja-preset}
```

である。ssithesis のオプションに関しては 3.11 節を参照されたい。苦労すれば、カラー絵文字も出すことができる 🌈 🐼 🏠 👨 👩 🍷。日本語フォント設定を変えて実行すると、初回実行時の実行時間がものすごくかかってしまう。これは、フォントの情報を “*.lua” というファイルに書き込んで保存するためである。特に yu-win10 (游フォント) は、64bit 処理系の別途インストールが必須となる (TeXLive2023 以降は、はじめから 64bit 処理系)。なお、JIS2004 がデフォルトのフォントに jis2004 オプションを付けても warning が増えるだけなので止めておいたほうが良い。JIS1990 の字形が欲しい場合は別途フォントを定義できる。

```
\newfontfamily\jisninety[CJKShape=JIS1990]{HaranoAjiMincho-Regular}
```

これを使って

```
通常の「辻葛」。  
{\jisninety JIS90 字形の「辻葛」。}
```

のように書くと

```
通常の「辻葛」。  
JIS90 字形の「辻葛」。
```

のようになる。ただし、IVS (異体字セレクタ) 付きの文字を入力可能ならば^{*2}、それを使って JIS90 文字を異体字としてそのまま入力してしまうほうが、現状では簡単である^{*3}。また、IVS 付き文字を入れるのが困難な場合は、luatexja-otf パッケージを使い\CID{...}で CID コードを入れる手がある。

3.4.2 Xe_{La}TeX+ bxsbook

Xe_{La}TeX は Overleaf 専用として考えていて、余り気を入れてデバッグしていない。たとえば、Noto フォントを使う場合は、以下のようになる。

```
\documentclass[xelatex,a4paper,11pt,ja=standard,jafont=noto]{bxsbook}
\usepackage[colorlinks]{ssithesis}
```

^{*2} たとえば、Windows の MS IME では「設定」「全般」で「変換候補の一覧に含める文字セットを選択する」の箇所を「IVS を除く」から「すべて」に変更すると、よく使われる異体字に限り入力できるようになる。

^{*3} ただし、IVS で区別されているはずの異体字がエディタ上で同じ字体で表示されてしまう場合があるので注意。VS Code の最新版は IVS 付き文字を正しく表示するが、古い VS Code では正しく表示されなかった。少なくとも IVS を除去したりしないエディタを使う必要がある。Overleaf では IVS 付き文字は入力不可能である。また、Unicode 6.3 から SVS (漢字だけではなく汎用の異体字選択機構) による CJK 互換漢字エリアの異体字選択が可能となったが、LuaTeX-ja がこれに対応しているかどうか不明である。なお、字体はフォント依存である。Adobe-Japan1 でないフォント、たとえば中国語のフォントでは同じコードでも字体が異なる (というより、中国語は IVS の適用範囲外である)。同じ日本語のフォントであっても、Adobe-Japan1 なフォントと Hanyo-Denshi に基づいているフォント (IPAmj 明朝) の 2 種類あり対応する IVS が異なる。

原ノ味フォントの場合は `jafont=haranoaji` となる。これも基本的には多ウェイトフォントを使うことができる。また、X_YL^AT_EX+ `bxjsbook.cls` も IVS 付き文字を認識するので、JIS90 フォントは L^AT_EX ソースの文字レベルで指定してしまえばよい。X_YL^AT_EX の場合は CID コードを直接入力できないが、どうしても入力したい場合は [5] の記事を参考にすればよい。

3.4.3 upL^AT_EX+ jsbook

いったん dvi ファイルを作成する方法。

```
\documentclass[dvipdfmx,uplatex,a4paper,jis2004]{jsbook}
\def\logfile{HuLogo-6.pdf}
\usepackage[colorlinks,hulogo]{ssithesis}
```

のように書く。フォントは TeXLive の場合、`kanji-config-updmap-sys` などで指定する。`divpdfmx` を使うのであれば、

```
\documentclass[dvipdfmx,uplatex,a4paper,jis2004]{jsbook}
\def\logfile{HuLogo-6.pdf}
\usepackage[colorlinks,hulogo]{ssithesis}
\usepackage[haranoaji,unicode]{pxchfon}
```

のように、L^AT_EX ソース中にフォントを記述することができる。otf ライブラリを `deluxe` オプションで読み込んでいるので、この組み合わせでも多ウェイトフォントを扱うことができる。グローバルオプションの `jis2004` は、`ssithesis` が読み込んでいる otf ライブラリが参照すると共に、`pxchfon.sty` も参照する。ただし、この組み合わせでは IVS (U+E0100~U+E01EF (VS17~VS256)) を読み込めないなので、文字毎に JIS2004/JIS1990 を変えるためには otf ライブラリの `\CID{...}` で記述するしか方法がなさそうである。

3.4.4 (pdf)L^AT_EX+ (scr)book

英語論文専用の組み合わせである。素の L^AT_EX を使うことはほぼ無いと思われるので、pdfL^AT_EX を使う前提で記述する。たとえば、

```
\documentclass[a4paper,11pt,pdftex]{book}
\def\logfile{HuLogo-6.pdf}
\usepackage[colorlinks,hulogo]{ssithesis}
```

のように書けばよい。何らかの理由で素の L^AT_EX を使って dvi 経由にしたい場合は、例えば `divpdfmx` を使うとして、

```
\documentclass[a4paper,11pt,dvipdfmx]{book}
```

```
\def\logofilename{HuLogo-6.pdf}
\usepackage[colorlinks,hulogo]{ssithesis}
```

となる。KOMA script の `scrbook.cls` を `pdflatex` で使う場合は、

```
\documentclass[a4paper,pdftex,chapterprefix=on]{scrbook}
\def\logofilename{HuLogo-6.pdf}
\usepackage[colorlinks,hulogo]{ssithesis}
```

となる。KOMA script では 11pt が標準なので、この場合 global option ‘11pt’ は不要である。`chapterprefix=on` を入れないと、章の見出しがシンプルなものになってしまう。

実は、`hyperref` を日本語化する過程でそれ以外の様々な手直しが入っている。それにより、日本語論文用の 3 つの組み合わせにおいては、`hyperref` の引数を元に pdf の bookmark を作る際に、数式を適切に処理し、カーニングも除去するなど色々と便利になっている。その手直しがこの組み合わせでは入っていないので、例えば `\section{...}` に数式が入るとワーニングが出る。そのようなことを避けるために、

```
\section{\texorpdfstring{\mathcal{H}_{-\infty}}{H-inf} control}
```

のように、表示用と PDF の bookmark 用で別な文字列を与えると良い。`\section[...]{...}` の場合は、[...] の中で `\texorpdfstring` の記述をすること。

3.4.5 Lua \LaTeX + (scr)book

これも、**英語論文専用**の組み合わせである。

```
\documentclass[a4paper,11pt]{book}
\def\logofilename{HuLogo-6.pdf}
\usepackage[colorlinks,hulogo]{ssithesis}
```

と書いて Lua \LaTeX でタイプセットすればよい。KOMA script の `scrbook.cls` を使う場合は、

```
\documentclass[a4paper,chapterprefix=on]{scrbook}
\def\logofilename{HuLogo-6.pdf}
\usepackage[colorlinks,hulogo]{ssithesis}
```

とする。global option ‘11pt’ が無い理由、および `chapterprefix=on` としている理由は 3.4.4 の項に書いた理由と同じである。また、3.4.4 の項の `hyperref` に関する注意書きはこの組み合わせでも有効である。

`book.cls` と `scrbook.cls` では、セクション等の見出しのフォントが異なる。`book.cls` ではボールド体になるが、`scrbook.cls` ではサンセリフ体になる。

3.5 アクサン・ウムラウト他

“(pdf)L^AT_EX+(scr)book”の組み合わせ、および日本語関係を除く“upL^AT_EX+jsbook”の組み合わせでは、T_EX ソースファイルの読み込みは utf-8、本文出力のためのフォント文字コードは T1 (8bit) エンコーディングになっている^{*4}。この場合、Unicode フォントではないので、アクサンやウムラウトがついているコードは、T1 に入っている分だけそのまま PDF に出力される。たとえば、éë è は L^AT_EX ソースに utf-8 で書いたものがそのまま PDF に出るが、ŵ などは 2 文字からなる人工的な合字^{*5}になる。一方、LuaL^AT_EX や XeL^AT_EX ではそのまま 1 文字でタイプセットされる。また、XeL^AT_EX(LuaL^AT_EX も?) では、Combining Diacritical Marks (U+0300–U+036F) を使って unicode 上で合成することもできるが、対応しているフォントでなくてはならないし、既に unicode 上に豊富な文字コードが用意されている (3.8 節参照) ので、あえて U+0300–U+036F を使う理由は無いであろう。

出力側はともあれ、どの処理系でも入力側は utf-8 なので、今後は、`\”{e}` で `ë` を出したりするような手間が少なくなるであろう。ただ、upL^AT_EX+fontenc.sty (utf8 オプション) では、欧文文字 utf-8 全部が入力可能ではなく、「出力不可能と思われる文字コードをそもそも受け取らない」ようになっているので注意である^{*6}。これは、inputenc による欧文 utf8 処理は入り口だけであり、結局従来の 8 ビットを基本とする fontenc で扱うことができる文字だけ受け付ける設定となっているからである。また、なぜ欧文文字だけにこのようなことが起きるのかに関しては「upL^AT_EX で unicode 内部処理が可能なのは和文のみであり、欧文に関しては従来の 8 ビット欧文 L^AT_EX と同じ能力しか持たない」からである。

また注意すべき点として、処理系によっては BibL^AT_EX 関係の unicode 化が未だのケースもあり^{*7}、`*.bib` ファイルでは`\”{e}`と書いておくのが安全である。

upL^AT_EX, (pdf)L^AT_EX の場合は、T1 の他に、オプションで、LGR, T2A エンコーディングも読み込んでいる。切り替えは babel を使って行うが、それに関しては 3.7 節を参照されたい。

3.6 ff, fi, fl, ffi, ffl の合字

たまに問題になるケースは、ff, fi, fl, ffi, ffl などの合字に関するものである。

昔の L^AT_EX でタイプセットされた合字で出力された PDF から、コピー・ペーストすると、ff, fi, fl, ffi, ffl などの合字 (Unicode でいえば U+FB00~U+FB04) がソースコード中に入ってしまう

^{*4} さすがに OT1 (7-bit) ではない。

^{*5} upL^AT_EX で文字 ‘ŵ’ (Unicode 1 文字) を使い、PDF からテキストエディタにコピー・ペーストすればわかる

^{*6} newunicodechar パッケージや\DeclareUnicodeCharacter マクロを使う手はある。

^{*7} LuaL^AT_EX などを使っても BibL^AT_EX と組み合わせるとこの問題が表出する。今後は BibL^AT_EX ではなく biblatex.sty + biber を使いましょう、といわれるようになるかもしれない。実は utf8 が bib に含まれるほとんどのケースでは、pbibtex -kanji=utf8 を使えば良いが、それでうまく動作する*.bib ファイルを外部のジャーナル編集者等に送ってはいけない。

うことがある*8。

Latin Modern Unicode	U+FB00~U+FB04	ff, fi, fl, ffi, ffl
Latin Modern Unicode	通常アルファベット	ff, fi, fl, ffi, ffl

Unicode 1文字で ff と書いても、f+f と書いても、PDF の出来上がりは同じくタイプセットされる。その点に関しては全く問題ない。しかし、 \LaTeX のソースで U+FB00~U+FB04 を使うと、文字列検索やスペルチェックを掛けたときにトラブルの元になるし、雑誌論文への投稿時に web のアブストラクト欄にコピー・ペーストで入力したものが文字化けしたりする。したがって、「 \LaTeX のソースでは、U+FB00~U+FB04 を使わない」ようにすべきだと思う。なお、上記は Lua \LaTeX , Xe \LaTeX の時の話である。up \LaTeX ではそもそも U+FB00~U+FB04 を読み込めない。

また、これに関連した話題が、3.8 節、3.9 節にて述べられているので参照されたい。

3.7 ギリシャ文字・キリル文字について

本スタイルファイルを使わずに「標準と思われる設定」でギリシャ文字・キリル文字を書くとき、エラーになるか、あるいは日本語フォント系の文字が出力される。日本語フォント系のギリシャ文字・キリル文字のグリフは間延びしてほぼ使い道がない*9。そこで、非数式部分でギリシャ文字・キリル文字を使った場合、欧文フォント系のグリフを使うようにした。ただし、以下の理由でそれらを実際に使うには一工夫必要になる。

- Latin Modern フォント (unicode) にはギリシャ文字・キリル文字のグリフがない。
- フォントが T1 (8bit) にマップされる場合 (pdf \LaTeX , \LaTeX , up \LaTeX) はそもそも T1 にギリシャ文字・キリル文字の割り当てが無い。

最初の問題点が Lua \LaTeX と Xe \LaTeX の場合、2 番目の問題点が up \LaTeX , pdf \LaTeX の場合に該当する。(Scr)book.cls を使う英語論文でも、ギリシャ文字・キリル文字を使うと上記のどちらかが問題になる (Lua \LaTeX を使うか (pdf) \LaTeX を使うかによる)。

まず、Lua \LaTeX と Xe \LaTeX の場合について述べる。Latin Modern には unicode においてもギリシャ語フォントが含まれていないようで、また、自動的に CM Unicode になったりもしない。よって、Lua \LaTeX , Xe \LaTeX の場合は、ギリシャ文字とキリル文字の入った TrueType フォントを別途指定する必要がある。例えばプリアンブルで

*8 最近の \LaTeX で作られたこの PDF では、「表示」は合字フォント 1 文字だが、「テキストにコピー・ペースト」すると元のアルファベットに分解される。具体的には、PDF 中に ff と次の文字の間に見えない細い空白文字のようなものがあり、[本体の文字 ff に ‘f’] + [空白文字に ‘f’] のようにテキストが割り当てられている。

*9 アスキーアートを除く (°Д°) /。p y c c k i i y z y k ⇒русский язык としたい。なお、up \LaTeX のときは、本スタイルファイル内で \usepackage[prefernoncjk,ccv+]{pxcjkcat} 相当の動作をしてから \usepackage[LGR,T2A,T1]{fontenc} としており、その場合、後から \cjkcategory で変えてもキリル文字部分は日本語扱いにはならなかった。


```
\newfontfamily\cmuserif{CMU Serif}
```

としておいて、

```
{\cmuserif Ελληνικά} ⇒ Ελληνικά
{\cmuserif русский язык} ⇒ русский язык
```

とする。

upL^AT_EX や pdfL^AT_EX の場合、T1 encoding は 8 ビットしかないの(昔の OT1 は 7 ビットなのでもっとひどかった)、色々出そうと思うと切り替え作業が必要となる。ssithesis.sty 中に fontenc に LGR, T2A はメインじゃないエンコーディングとして登録済である。よって、babel を使って、フォントエンコーディングを LGR や T2A に一時的に変更すれば、ギリシャ文字・キリル文字も出せる。

```
\foreignlanguage{greek}{Ελληνικά} ⇒ Ελληνικά
\foreignlanguage{russian}{русский язык} ⇒ русский язык
```

Latin Modern から CM フォントに自動的に置き換わる^{*10}。New TX fonts に変えた場合も、キリル文字フォント・(数式じゃない)ギリシャ文字の両方が CM フォントに置き換わる。これは、\usepackage{lmodern}と書いたときに T1 エンコーディングだけを入れ替えているので、T2A や LGR に関してはデフォルトの CM フォントになるからである。CM フォントを嫌うのであれば、DejaVu フォントが T2*, LGR もサポートしている(らしい)ので、好みがあれば\usepackage{dejavu}として使う方法があるが、数式以外の英文フォント系が全て置き換わる。

3.8 欧文文字領域のフォントについて

以下は、ASCII コードより先の欧文フォント領域 U+0080~U+036F および U+1E00~U+1EFF の文字である^{*11}。Unicode font の話なので、LuaL^AT_EX あるいは XeL^AT_EX を使うときだけ関係する。赤文字は Latin Modern にはなく CM unicode にある文字、水色はどちらにも無い文字で本 PDF では Times New Roman で表示されているか、あるいは正常なグリフが表示されていない。緑文字はドイツ語のエス・ツェットの大文字(シャーフェス・エス; SS)で本来は“ß”(Times New Roman で表示)であるが LuaL^AT_EX 上の Latin Modern では SS と印字される。これらの色分けは、LuaL^AT_EX で私が試してみた結果なので、正式な文書から引用したものは無く、また、XeL^AT_EX の場合と若干結果が異なっている事を断っておく。

U+0080—U+00FF Latin-1 Supplement

```
ıçŁŒŸŹšˆ©ª«¬(®¯°±²³´µ¶·¸¹º»¼½¾ÀÁÂÃÄÅÆÇÈÉÊËÌÍ
```

^{*10} キリル文字フォントの時だけワーニングが出る。

^{*11} U + 3670~U+1DFF は各種言語の文字・発音記号などである。

3.9 Unicode フォントの英語と日本語の切り替えについて

本節も unicode font の話なので、LuaL^AT_EX あるいは XeL^AT_EX を使うときときだけ関係する。また、(scr)book.cls で英語論文を書くときは CJK フォントという概念は無いので本節は無視して良い (全て欧文フォント扱い)。Unicode の U+2070~U+2BFF のあたりのコードに対し、日本語フォントが使われるか英語フォントが使われるかは処理系依存である。LuaL^AT_EX+LuaT_EX-ja の場合^{*12}や XeL^AT_EX と共に使う bxjsbook.cls では英語フォントを使うようになっている。ただし、L^AT_EX システムで使われるフォントでは、ほとんどこの辺りをカバーしていない。そのような場合、LuaL^AT_EX, XeL^AT_EX の場合は次のような方策をとると良い。

1. その文字が入っている Truetype / OpenType フォントを使う。例えば、Windows では

```
\newfontfamily\arialuni{Arial Unicode MS}% 持っている場合
\newfontfamily\timesnew{Times New Roman}%
```

としておいて、Times New Roman を使いたいなら

```
{\timesnew 0}
```

とすればよい (結果: 0)。U+2070~U+2BFF のあたりしか調べていないが、Times New Roman より Arial Unicode MS の方が対応文字が若干多いようである。しかし、Microsoft による Arial Unicode MS のフリーダウンロードサービスは終了している。

2. 日本語フォントと強制的にみなす。たとえば、「➡」は haranoaji フォントに含まれているので日本語フォントで出力するためには、LuaL^AT_EX では、\ltjjachar「➡」とする。「」はバッククォートなので注意。XeL^AT_EX の場合は、\jachar{➡}とする。ただし、このコード領域における haranoaji フォントの対応文字は極めて少ない^{*13}

LuaL^AT_EX で、逆に日本語フォント領域にある文字を欧文フォントで出したい場合は、\ltjalchar「※」と書くと ※ となる。XeL^AT_EX の場合に相当するコマンドは、{\XeTeXcharclass「…」=0 …}である (「…」→「…」となる)。

















































なお、日本語文字と欧文文字の間には四分アキといわれる空白 (グルー) が挿入されるので、日本語フォントと欧文フォントの切り替えでは、グリフの見た目の変更だけではない違いがでてくる^{*14}。

なお、LuaT_EX-ja で日本語フォント (CJK) とみなされている以下のコードが、XeL^AT_EX(bxjsbook) では欧文フォントとされているので注意。これは、あくまで例であって、他にも扱いが異なる文字コードがあるかもしれない。

^{*12} デフォルトは日本語フォントだが本スタイルファイル内で切り替えている

^{*13} それでも Latin Modern よりは多い。

^{*14} 「あ※あ」と「あ※あ」を比較。

		(U+2002) En space,			(U+2010) Hyphen,
		(U+2011) Non-breaking hyphen,			(U+2016) Double vertical line,
		(U+201A) Single low-9 quotation mark,			
		(U+201E) Double low-9 quotation mark,			(U+2020) Dagger,
		(U+2021) Double dagger,			(U+2022) Bullet,
		(U+2030) Per mille sign,			(U+2032) Prime,
		(U+2033) Double prime,			(U+2039) Single left-pointing angle quot.,
		(U+203A) Single right-pointing angle quot.,			
		(U+203B) Reference mark,			
		(U+203C) Double exclamation mark,			(U+203E) Overline,
		(U+203F) Undertie,			(U+2042) Asterism,
		(U+2044) Fraction slash,			(U+2047) Double question mark,
		(U+2048) Question exclamation mark,			
		(U+2049) Exclamation question mark,			
		(U+2051) Two asterisks aligned vertically			

また、両方ともに“Em dash”, “En dash”, “Horizontal bar”, “Left single quotation mark”, “Right single quotation mark”, “Left double quotation mark”, “Right double quotation mark”, “Horizontal ellipsis (和文だと三点リーダ)”はCJK扱いである。これは、「欧文の文脈においては、相当する \LaTeX の記法(-, -, ‘ ’, “ ”, \ldots,...)の方を使いなさい」ということだと思う。

3.10 中国語などの表示について

中国語などの文字を出したい場合があるかもしれない。特に、名前を簡体字で出したいという要望があるかもしれない。以下では、簡体字の場合だけ示すが、繁体字やハングル文字も同様のはずである。ここでは、 \TeX -Liveに含まれている、Fandolフォントを使うこととする。なお、(scr)book.clsを使い英語論文を書く場合は、本節の内容は一部正しくない^{*15}ので注意。

Lua \LaTeX の場合にプリアンブルに

```
\newfontfamily\gbsong[BoldFont=FandolSong-Bold.otf]%
  {FandolSong-Regular.otf}
\newfontfamily\gbhei[BoldFont=FandolHei-Bold.otf]%
  {FandolHei-Regular.otf}
\newfontfamily\gbfang{FandolFang-Regular.otf}
\newfontfamily\gbkai{FandolKai-Regular.otf}
```

^{*15} newfontfamily を newfontfamilyy にすればおそらく大丈夫。

と書く。X_YL^AT_EX の場合は、同じくプリアンブルで

```
\defaultCJKfontfeatures{Scale=0.924715}
\newCJKfontfamily\gbsong[BoldFont=FandolSong-Bold.otf]%
    {FandolSong-Regular.otf}
\newCJKfontfamily\gbhei[BoldFont=FandolHei-Bold.otf]%
    {FandolHei-Regular.otf}
\newCJKfontfamily\gbfang{FandolFang-Regular.otf}
\newCJKfontfamily\gbkai{FandolKai-Regular.otf}
```

のように宣言する。LuaL^AT_EX, X_YL^AT_EX の場合は、本文中で、`{\gbsong 简体中文}`とすれば「简体中文」となる。一方、`{\gbhei 简体中文 (黒体)}`とすれば「简体中文 (黒体)」となる。また、`{\gbfang 简体中文}`および`{\gbkai 简体中文}`とすれば、それぞれ「简体中文」および「简体中文」となる。

一方、upL^AT_EX の場合、dvipdfmx を使うと仮定し、プリアンブルで

```
\usepackage[haranoaji,unicode]{pxchfon}
\setchineseminchfont[0]{FandolSong-Regular.otf}
\setchinese Gothicfont[0]{FandolHei-Regular.otf}
```

などを書いておく。すると、`\foreignlanguage{schinese}{简体中文}`とすれば「简体中文」となる。また、`\foreignlanguage{schinese}{\gtfamily 简体中文 (黒体)}`とすれば「简体中文 (黒体)」となる。

3.11 ssithesis.sty のオプションについて

よく使うグローバルオプション(クラスファイル名の前の [...] の中)と ssithesis.sty で用いられるオプションを表 3.1 にまとめた。graphicx, xcolor, hyperref を読み込んでいるので、それらに関するオプションはグローバルオプションのところに書く。

3.12 TeXLive のバージョンについて

2021 年 3 月 27 日現在、TeXLive 2020 のツリーは凍結され、TeXLive 2021 のリリース待ちになっている。今後も TeXLive は新しいものに順次更新し、また TeX Live Manager (tlmgr) でパッチを定期的に当てていくことをお勧めする。原ノ味フォントがデフォルトフォントという理由だけでも TeXLive 2020(以降) を使うべきだと思う。そのためには、設定変更や外部パッケージ・フォントの導入は texmf-local フォルダ以下にすべきで、TeXLive を更新してもこれらが引き継がれるようにしたほうがよい。

なお、TeXLive 2020 で欧文フォントスキームの変更があり、フォントのウェイト(太さ)は x

軸と y 軸で別管理になった。その変更に対する (lt/bx)jsarticle 側の変更が TeXLive 2020 の当初版では間に合っていないので、ワーニングが大量に出る。導入後に tlmgr を使って、TeXLive 2020 のアップデートが必須である。

表 3.1 オプション一覧

Global Options	説明
a4paper	A4 用紙を指定. 本スタイルファイルでは必須
11pt	文字の大きさを 11pt にする. ここしばらくは 11pt 文字で書くのが本コースの標準であったので, 今回もそれに倣う. scrbook.cls を使う場合は不要
uplatex	upL ^A T _E X+ jsbook.cls の組み合わせでは必須
xelatex	X _Ǝ L ^A T _E X+ bxjsbook.cls の組み合わせでは必須
english	(lt/bx)jsbook.cls で英語論文を書くときはこれを指定
pdftex	英語論文において, pdfL ^A T _E X+ (scr)book.cls の組み合わせではこれを書く. graphicx や xcolor に正しいデバイスオプションを伝える ^{a)}
oneside	章の切れ目に空白ページが挿入されないようにする
vartwoside	偶奇ページのヘッダを同じにする
jis2004	upL ^A T _E X と共に使うときに, JIS2004 字形を指定する. 古い JIS1990 を使うのでなければ, uplatex コマンドのときは必ず指定すると考えても問題ない
chapterprefix=on	scrbook.cls を使う場合は, これを定義する.
Local Options	説明
colorlinks	Hyperref のリンクに色を付ける. 目次の各項目にも色が付くので, 提出版ではこれを指定しない方がよいかも
nomodern	\usepackage{lmodern}相当の動作をしないようになる
nofontspec	LuaL ^A T _E X, X _Ǝ L ^A T _E X の下では, \usepackage[no-math]{fontspec}相当の動作をするが, それを抑制する ^{b)}
noamssymb	amssymb.sty を読み込まない. Unicode-Math を使う時に指定すると良いかもしれない。
nodeluxe	upL ^A T _E X+ jsbook.cls の組み合わせでは otf.sty を読み込むが, そのときのオプションに deluxe を付けない. 何らかの理由で多ウェイトフォントを使わない時はこれを指定.
equalmargin	左右のマージンを等しくする
hulogo	表紙に北海道大学ロゴを表示する. ロゴのファイル名は \def\logofilename{HuLogo-6.pdf}のように定義する.

a) 今は、graphicx, xcolor, hyperref の前にデバイスオプションを書いたり “*.def” ファイルを書いたりするのは非推奨で、global option に書くのが推薦。

b) LuaL^AT_EX 使用時には、luatexja-preset.sty 実行時に fontspec.sty をオプション無しで実行する (オプションは先に書かれた方が優先)。

第 4 章

その他

4.1 長い章・節タイトル

`\chapter` には{ }の前に [] で、目次に載せる項目を変更させる方法がある。これにより、長い章のタイトルが入った目次を変更できる。

また、奇数ページ左上のヘッダ部分の文字列を変更するには、`\chaptermark{変更後文字列}` というコマンドを `\chapter{ }` コマンドの後に入れると良い。同様に、奇数ページ右上の文字列は、`\section{ }` の後に、`\sectionmark{ }` コマンドで変えることができる。

4.2 図について

このスタイルファイルは、

```
\usepackage{amsmath,amssymb}
\usepackage[nosetpagesize]{graphicx}
\usepackage{xcolor}
```

相当が実行されている。図は `graphicx` パッケージを使い出力してほしい。その際、長さを示す単位として、`pt`, `mm`, `cm`, `in` の代わりに `truept`, `truemm`, `truecm`, `truein` を使ってほしい。これは、`js` スタイルファイルに `10pt` 以外のものを使用すると、「ちょっと小さな用紙に `10pt` で版を作る」⇒「拡大して表示」の手順をとるからである。通常の `pt`, `mm`, `cm`, `in` は拡大前の長さを示している。

なお、`EPS` ではなく、`PDF` を読み込むように今後なっていくと思われる。Adobe Illustrator を使っているならば、アートボードをオブジェクト全体に合わせて、別名で `PDF` にセーブ (編集用には元のファイルを保持) し、読み込むと良い。別名でセーブ時、Illustrator の編集機能を保持しないで、`bookmark` やサムネールも全部取ってファイルサイズを小さくすべきである。アートボードをオブジェクト全体に合わせる作業をすると、`Bounding Box` をオブジェクト全体に合わせてくれる。また、`TeXLive` に含まれている `pdfcrop` というコマンドを使って `PDF`

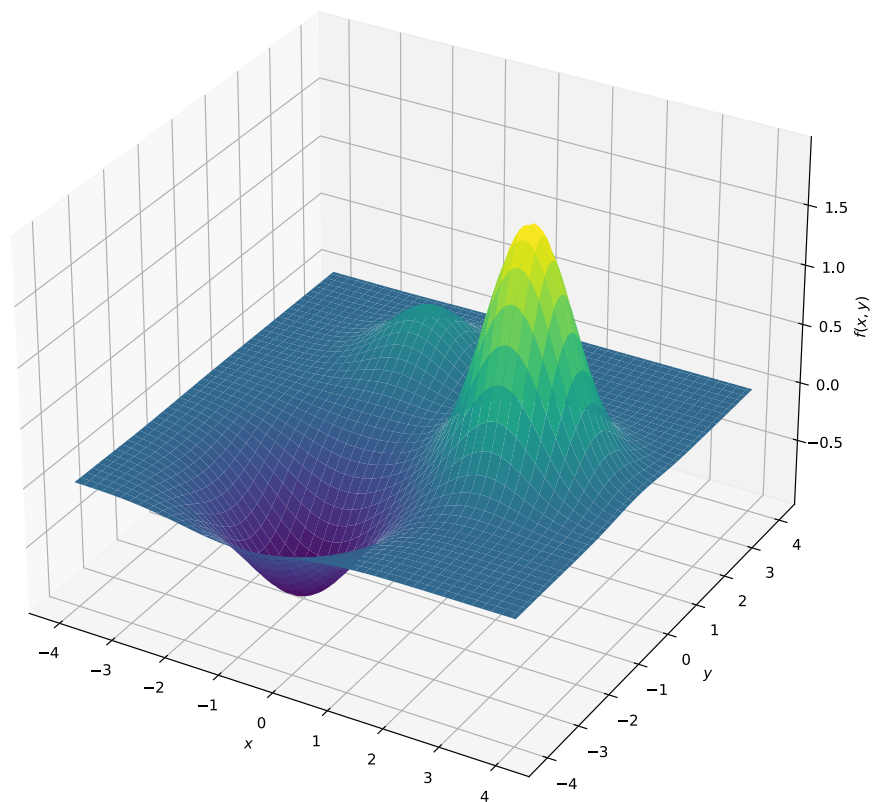


図 4.1 図を取り込んだ例

の図の余白をトリミングする方法もあるが、ソフトが生成する図にはしばしば「見えないページ枠」が含まれていてそれに沿ってトリミングしてしまうことがあるので注意。この作業が面倒な場合は、`mediabb.sty` を読んでおく方法があるらしいが、本スタイルファイル作成者は未検証である。

4.3 数式について

`AMSTEX` を読み込んである。`align` 環境, `align*`環境, `alignat` 環境, `alignat*`環境, `gather` 環境, `aligned` 環境, `split` 環境を使い分けよう。以下に例を示す。式番号の付き方がそれぞれ違っていることに注意されたい。なお、以下の例において、`unicode-math` 使用時は `\boldsymbol` の代わりに `\symbfit` を使うこと。

4.3.1 `align` 環境

```
\begin{align}
```

```

& \dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu} \\
& \boldsymbol{y} = \boldsymbol{Cx} \\
& \boldsymbol{A} = \boldsymbol{A}_{\mathrm{orig}} + \boldsymbol{BF}
\end{align}

```

$$\dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu} \quad (4.1)$$

$$\boldsymbol{y} = \boldsymbol{Cx} \quad (4.2)$$

$$\boldsymbol{A} = \boldsymbol{A}_{\mathrm{orig}} + \boldsymbol{BF} \quad (4.3)$$

4.3.2 align*環境

```

\begin{align*}
& \dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu} \\
& \boldsymbol{y} = \boldsymbol{Cx} \\
& \boldsymbol{A} = \boldsymbol{A}_{\mathrm{orig}} + \boldsymbol{BF}
\end{align*}

```

$$\dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu}$$

$$\boldsymbol{y} = \boldsymbol{Cx}$$

$$\boldsymbol{A} = \boldsymbol{A}_{\mathrm{orig}} + \boldsymbol{BF}$$

4.3.3 equation + aligned 環境

```

\begin{equation}
\begin{aligned}
& \dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu} \\
& \boldsymbol{y} = \boldsymbol{Cx} \\
& \boldsymbol{A} = \boldsymbol{A}_{\mathrm{orig}} + \boldsymbol{BF}
\end{aligned}
\end{equation}

```

$$\dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu}$$

$$\boldsymbol{y} = \boldsymbol{Cx} \quad (4.4)$$

$$\boldsymbol{A} = \boldsymbol{A}_{\mathrm{orig}} + \boldsymbol{BF}$$

equation 環境と align 環境では上下のスペーシングが異なることがわかる。 \mathcal{AMS} の align

系は、段落の最初の行に `display` 数式を置いても (英文で普通書いていけばありえないが...) 上部にスペースが空く*¹。

4.3.4 align + split 環境

```
\begin{align}
  \begin{split}
    & \dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu} \\
    & \boldsymbol{y} = \boldsymbol{Cx}
  \end{split}
\end{align}
\\
& \boldsymbol{A} = \boldsymbol{A}_{\mathrm{orig}} + \boldsymbol{BF}
```

$$\dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu} \quad (4.5)$$

$$\boldsymbol{y} = \boldsymbol{Cx} \quad (4.6)$$

$$\boldsymbol{A} = \boldsymbol{A}_{\mathrm{orig}} + \boldsymbol{BF} \quad (4.6)$$

4.4 Times 系フォントの使い方

4.4.1 Lua \LaTeX の場合

Lua \LaTeX の場合は Times 系にするときは、数式は `newtxfont` でそれ以外の欧文文字は `truetype/opentype` フォントに変えたほうが簡単である。たとえば、

```
\documentclass[a4paper,11pt]{ltjsbook}
\def\logofilename{HuLogo-6.pdf}
\usepackage[colorlinks,nolmodern,hulogo]{ssithesis}
\usepackage[varg]{newtxmath}
\setmainfont[Ligatures=TeX]{TeX Gyre Termes}
\setsansfont[Ligatures=TeX]{TeX Gyre Heros}
\setmonofont{Nimbus Mono}%TeX Gyre Cursor は細すぎ
\usepackage[deluxe,haranoaji]{luatexja-preset}
```

のように書く。TeXLive 2019 までの Lua \LaTeX + `newtxmath` の組み合わせでは、アクセント 2 階建て (\dot{x} など) のときに変になってしまっていたが、TeXLive 2020 からまともになっている。TeXLive 2019 のときは `newtxmath` ではなく `\usepackage[varg]{txfonts}` を使うと良い。ま

*¹ よって、Beamer で `align` 系を使うと縦方向の空白が空く傾向がある。

た、Unicode Math を使うことも検討すべきである (4.5 参照)。

4.4.2 X \LaTeX の場合

たとえば、

```
\documentclass[xelatex,a4paper,11pt,ja=standard,jafont=haranoaji]{bxjsbook}
\def\logofilename{HuLogo-6.pdf}
\usepackage[colorlinks,nolmodern,hulogo]{ssithesis}
\usepackage[varg]{newtxmath}
\setmainfont[Ligatures=TeX]{TeX Gyre Termes}
\setsansfont[Ligatures=TeX]{TeX Gyre Heros}
\setmonofont{Nimbus Mono}
```

のように書く。私の環境では Nimbus Mono の代わりに zco Light が使われるのが不思議だが...

4.4.3 up \LaTeX , pdf \LaTeX の場合

Truetype/opentype を使うことは最初から諦めるので、ここは単純に、

```
\usepackage[colorlinks,nolmodern,hulogo]{ssithesis}
\usepackage[defaultsup]{newtxtext}
\usepackage[varg]{newtxmath}
```

だけで良い。

4.5 Lua \LaTeX + unicode-math

Lua \LaTeX /X \LaTeX の場合、本スタイルファイルでは `\usepackage[no-math]{fontspec}` として `fontspec.sty` を読み込んでいる (`nofontspec` を指定しない場合) ので、数式モードの中は 7bit のフォントエンコーディングのままである。[no-math] を取り除いただけの状態では、様々なパッケージと衝突する割には数式中で unicode フォントを使えるわけでもなく^{*2}、中途半端な状態になる。もし、数式中も unicode フォントを使いたいのであれば、例えば以下のようになるとよい。

```
\usepackage{fontspec}
\usepackage[colorlinks,nolmodern,nofontspec,noamssymb,hulogo]{ssithesis}
\usepackage[math-style=TeX,bold-style=TeX]{unicode-math}
\setmainfont[Ligatures=TeX]{Latin Modern Roman}
```

^{*2} メリットとしては `\mathrm{}` 中でアクセント付き文字が T1 でタイプセットされるくらいしかない。

```

\setsansfont[Ligatures=TeX]{Latin Modern Sans}
\setmonofont{Latin Modern Mono}
\setmathfont{Latin Modern Math}
\setmathfont{XITS Math}[range={scr,bfscr}]

```

また、Times 系の場合の例は以下の通りである。

```

\usepackage{fontspec}
\usepackage[colorlinks,nolmodern,nofontspec,noamssymb,hulogo]{ssithesis}
\usepackage[math-style=TeX,bold-style=TeX]{unicode-math}
\setmainfont[Ligatures=TeX]{XITS}
\setsansfont[Ligatures=TeX]{TeX Gyre Heros}
\setmonofont[Ligatures=TeX]{Inconsolatazi4}
\setmathfont{XITS Math}
\setmathfont{XITS Math}[range={cal,bfcal},StylisticSet=1]

```

これらは、Lua \LaTeX の場合しか検証していないが、Xe \LaTeX の場合も同様だと思われる。なお、Unicode のコード表の上ではスクリプト体とカリグラフィ体の区別がない。そのため、Latin Modern Math フォントにはスクリプト体のフォントがなく、そのままではカリグラフィ体が代わりに出力されてしまうので、上記の非 Times 系の例ではスクリプト体だけ別フォントに置き換えている。また、Times 系の例では XITS Math フォントを使っているが、XITS Math フォントのデフォルトではカリグラフィ体を指定してもスクリプト体になる。上記の例では、カリグラフィ体の時は当該フォントの異体字を使うようにしており、きちんとカリグラフィ体になる。

`unicode-math` の場合は、`\mathbf` などの数式中テキストのフォント切り替えと他の数式部分のフォント切り替えコマンドが分離されている。通常の数式部分のフォント切り替えは、`\symup`, `\symbfup`, `\symit`, `\symbit`, `\symsup`, `\symsfit`, `\symbfsup`, `\symbfsfit`, `\symtt`, `\symbb`, `\symbbit`, `\symscr`, `\symbfscr`, `\symfrac`, `\symbffrak`, `\symcal`, `\symbfcal` などを使う。表 4.1 を参考にされたい。`\math***`形式のコマンドは主に式中の非数式部分にて使われ、その場合はリガチャ(合字)がされる。よって、`\mathbfup{fit}`と`\symbfup{fit}`の結果は、“**fit**”と“**fit**”のように異なる。Unicode Math の通常の式に対しては、`\sym***`形式のコマンドを使うべきである。

`math-style=TeX` オプションのときの通常の(無指定の場合の)ギリシャ文字大文字は立体になる。これを斜体にしたければ、`\symit{\Gamma}`などとするか、オプションを`math-style=ISO`のように変えるべきである。また、`bold-style=TeX` オプションのとき、`\symbf` は、ギリシャ文字小文字を除いてほぼ`\symbfup` に等しい。一方、`bold-style=ISO` オプションのとき、`\symbf` は、`\symbit` に等しい。

特に注意すべきなのは、`unicode-math` のときは `amssymb.sty` を読み込んではいけないことである。そのため上記の例では `noamssymb` オプションを指定している。`amssymb.sty` を読み込ま

表 4.1 Unicode Math でのフォント指定 (math-style=TeX, bold-style=TeX の場合)

	通常数式中	式中の非数式部分	例 ($\backslash\text{sym}***$ のとき)
通常			$abcABC\alpha\beta\Xi\Gamma$
立体	$\backslash\text{symup}$	$\backslash\text{mathup}$	$abcABC\alpha\beta\Xi\Gamma$
立体太文字	$\backslash\text{symbfup}$	$\backslash\text{mathbfup}$	$abcABC\alpha\beta\Xi\Gamma$
	($\approx \backslash\text{symbf}$)	($\approx \backslash\text{mathbf}$)	$abcABC\alpha\beta\Xi\Gamma$
斜体	$\backslash\text{symit}$	$\backslash\text{mathit}$	<i>$abcABC\alpha\beta\Xi\Gamma$</i>
斜体太文字	$\backslash\text{symbfifit}$	$\backslash\text{mathbfifit}$	<i>$abcABC\alpha\beta\Xi\Gamma$</i>
Sans Serif	$\backslash\text{symsfup}$	$\backslash\text{mathsfup}$	$abcABC\alpha\beta\Xi\Gamma$
Sans Serif 斜体	$\backslash\text{symsfit}$	$\backslash\text{mathsfifit}$	<i>$abcABC\alpha\beta\Xi\Gamma$</i>
Sans Serif 太文字	$\backslash\text{symbfsfup}$	$\backslash\text{mathbfsfup}$	$abcABC\alpha\beta\Xi\Gamma$
	($= \backslash\text{symbfsf}$)	($= \backslash\text{mathbfsf}$)	$abcABC\alpha\beta\Xi\Gamma$
Sans Serif 太文字斜体	$\backslash\text{symbfsfit}$	$\backslash\text{mathbfsfit}$	<i>$abcABC\alpha\beta\Xi\Gamma$</i>
タイプライター文字	$\backslash\text{symtt}$	$\backslash\text{mathtt}$	$abcABC\alpha\beta\Xi\Gamma$
黒板太字	$\backslash\text{sybbb}$	$\backslash\text{mathbbb}$	$abcABC\alpha\beta\Xi\Gamma$
黒板太字斜体 (Ddeij のみ)	$\backslash\text{sybbbifit}$	$\backslash\text{mathbbbifit}$	<i>$Ddeij$</i>
花文字 (script 体)	$\backslash\text{symscr}$	$\backslash\text{mathscr}$	<i>$abcABC\alpha\beta\Xi\Gamma$</i>
花文字太文字	$\backslash\text{symbfscr}$	$\backslash\text{mathbfscr}$	<i>$abcABC\alpha\beta\Xi\Gamma$</i>
Fraktur	$\backslash\text{symfrac}$	$\backslash\text{mathfrac}$	$abc\mathfrak{ABC}\alpha\beta\Xi\Gamma$
Fraktur 太文字	$\backslash\text{symbffrak}$	$\backslash\text{mathbffrak}$	$abc\mathfrak{ABC}\alpha\beta\Xi\Gamma$
カリグラフ	$\backslash\text{symcal}$	$\backslash\text{mathcal}$	$abc\mathcal{ABC}\alpha\beta\Xi\Gamma$
カリグラフ太文字	$\backslash\text{symbfcal}$	$\backslash\text{mathbfcal}$	$abc\mathcal{ABC}\alpha\beta\Xi\Gamma$

なくても unicode-math では amssymb.sty で定義されている記号や Fraktur($\backslash\text{symfrac}$) や花文字 (script 体; $\backslash\text{symscr}$) も相当する unicode 文字を使って最初から使える状態にある。

非 unicode-math の場合 (表 4.2 参照^{*3}) の場合と比べても、フォント選択の自由度が大きくなっていることがわかる。表 4.2 の非 unicode-math の場合のコマンドを unicode-math の時にも使えることがあるが、できるだけ unicode-math では $\backslash\text{sym}***\{\dots\}$ 形式のものを使うようにしよう。また、Unicode-math で作った PDF から MS-Word に数式をコピー・ペーストした場合、従来のように判別不能な文字列にはならず、ある程度数式の形を保ったままコピーされる。MS-Word の数式エディタ中にコピー・ペーストして、添字などの処理を加えればちゃんとした数式になることが多い。一方、unicode-math で作った PDF を Adobe Illustrator で読み込むと、数式部分はアウトライン化した図形として処理される (数字文字^{*4}を除く)。これを

^{*3} 非 unicode-math 時にスクリプト体数式フォントを使う場合は $\backslash\text{usepackage}\{\text{mathrsfs}\}$ とする。

^{*4} これらを処理するために、Latin Modern Math などの otf フォントを Windows から見えるようにして、Illustrator で読み込むときに「フォント置き換え」をするとよい。

表 4.2 非 Unicode-Math でのフォント指定

	コマンド	例
通常		$abcABC\alpha\beta\Xi\Gamma$
立体	<code>\mathrm</code>	$abcABC\alpha\beta\Xi\Gamma$
斜体	<code>\mathit</code>	$abcABC\alpha\beta\Xi\Gamma$
立体ボールド	<code>\mathbf</code>	$\mathbf{abcABC\alpha\beta\Xi\Gamma}$
斜体ボールド	<code>\boldsymbol</code>	$\boldsymbol{abcABC\alpha\beta\Xi\Gamma}$
Sans Serif (ラテン文字のみ)	<code>\mathsf</code>	$abcABC$
タイプライター文字 (ラテン文字のみ)	<code>\mathtt</code>	$abcABC$
黒板太字 (ラテン大文字のみ)	<code>\mathbb</code>	\mathbb{ABC}
花文字 (script 体; ラテン大文字のみ)	<code>\mathscr</code>	\mathscr{ABC}
Fraktur (ラテン文字のみ)	<code>\mathfrak</code>	$\mathfrak{abc\mathbb{A}\mathbb{B}\mathbb{C}}$
カリグラフィ (ラテン大文字のみ)	<code>\mathcal</code>	\mathcal{ABC}

Illustrator で処理しているグラフ等にコピーすれば、図と本文で同じ見た目の数式を表示できる (PSTricks は Illustrator 作成 EPS では使えないので...)*⁵。

4.6 定理環境

$\mathcal{A}\mathcal{M}\mathcal{S}$ の `amsthm` を読み込んでいる。`amsthm` を使っているので、`proof` 環境も自動的に定義されるが、日本語環境では「証明」と日本語で書かれる。日本語論文において、「斜体になる」等の不都合が起きないようにしてある。表 4.3 にスタイルファイル内で用意した定理環境の一覧をまとめてある。

たとえば、

```
\begin{thm}
一点可縮でない多様体上で定義された滑らかなシステムのある平衡点を
滑らかな状態フィードバックで大域的漸近安定化することは不可能である。
\end{thm}
\begin{proof}
リアプノフ関数のレベル面は閉ループ系の流れに沿って変移レトラクトと
なるので、サブレベル集合は原点に滑らかに変形できないとおかしい。(大雑把すぎ)
\end{proof}
```

と書けば、

*⁵ 簡単な図なら TikZ で描いてしまうのが $\text{L}\text{T}\text{E}\text{X}$ 界隈の流行であるが、個人的感想を言えばやはり面倒である。

表 4.3 定理環境一覧

環境名	日本語見出し	英語見出し
thm	定理	Theorem
lem	補助定理	Lemma
prop	補題	Proposition
cor	系	Corollary
axiom	公理	Axiom
defi	定義	Definition
assum	仮定	Assumption
conj	予想	Conjecture
exam	例題	Example
rem	注意	Remark

定理 1 一点可縮でない多様体上で定義された滑らかなシステムのある平衡点を滑らかな状態フィードバックで大域的漸近安定化することは不可能である。

証明 リアプノフ関数のレベル面は閉ループ系の流れに沿って変移レトラクトとなるので、サブレベル集合は原点に滑らかに変形できないとおかしい。(大雑把すぎ) □

となる。

4.7 目次のレベル

目次を subsection まで出す場合は、

```
\setcounter{tocdepth}{2}
```

をプリアンブルに入れるとよい。デフォルトは section までである。

4.8 参考文献について

本スタイルファイルでは参考文献スタイルに関しては何も定めてはいない。BibTeX 等を使うのであれば、各々の専門分野で良く用いられている好きなスタイルを採用すればよい。

最近では BibTeX ではなく biblatex.sty を使う方法が徐々に普及し始めていて、自由に unicode が使える。本スタイルファイルでもこれを標準採用しようと考えたが、結局は何もしなかった。これは以下の理由による。

- 日本語対応が不足 (無理やり使う方法は [6] 参照. 2018 年くらいからの biblatex.sty で使うことができる.)

- 従来の bib スタイルファイルが使えない。これは各 bibitem の内容が biblatex の $\text{T}_{\text{E}}\text{X}$ マクロでフォーマットされる^{*6}からで、biblatex 用に書かれたスタイルを使う必要がある。よって、biblatex.sty 用に unicode で書いた .bib ファイルを使いまわす際にトラブルが起きる可能性がゼロではない (up $\text{B}_{\text{I}}\text{T}_{\text{E}}\text{X}$ が使い物になればそんなことを気にしなくても良いかもしれないが...)。

各 Journal の bib スタイルが $\text{B}_{\text{I}}\text{T}_{\text{E}}\text{X}$ と biblatex 両方に提供されるようになれば、改めて本スタイルファイルでも考えたいと思っている。

学位論文著者が biblatex を使うことを否定しているわけではなく (むしろどんどん使えば良いと思っている)、各自で自由に書けばよい。その際のスタイルは IEEE 形式

```
\usepackage[backend=biber,style=ieee]{biblatex}
```

を推薦としたい (少なくとも TeXLive 2020 では biblatex-ieee パッケージは入っている)。標準の style=numeric は、工学でよく使われている形式とはちょっと違う気がする。 $\text{B}_{\text{I}}\text{T}_{\text{E}}\text{X}$ 互換の style=trad-plain でも良いかもしれない。参考文献を出力する箇所では `\printbibliography[title=参考文献]` と書く ($\text{B}_{\text{I}}\text{T}_{\text{E}}\text{X}$ とは異なる)。title=`\bibname` でも良いかもしれない。

$\text{B}_{\text{I}}\text{T}_{\text{E}}\text{X}$ や (u)p $\text{B}_{\text{I}}\text{T}_{\text{E}}\text{X}$ のコードがメンテナンスされずに放置されている現状を考えると、biblatex.sty への移行は将来的には避けられない気がしている。

^{*6} バックエンドで使う bibtex や biber は文献のソートだけを行っている。

第 5 章

英語論文

5.1 英語論文を書くには

英語論文を書くには 2 つの方法がある。1 つは、日本語論文で許されている 3 つの組み合わせにおいて、グローバルオプションに `english` と書くことである。例えば、Lua \LaTeX の場合、

```
\documentclass[a4paper,11pt,english]{ltjsbook}
\usepackage[colorlinks,hulogo]{ssithesis}
```

とする。日本語文字を一切使わないのであれば、`luatexja-preset` の読み込みも不要である^{*1}。こちらのほうは、まだテストが完全ではないので不具合があれば申し出てほしい。ただし、“`english`” オプションで英語論文にする方法では、日本語論文の場合と同様に章や節の見出しが **boldface** にならずに **Sans Serif** になる。

もう 1 つは、(pdf) \LaTeX あるいは Lua \LaTeX と `book.cls` / `scrbook.cls` を使う方法で、サブセクション 3.4.4, 3.4.5 で説明したやり方である。pdf \LaTeX + `book.cls` を使う場合は、

```
\documentclass[a4paper,11pt,pdftex]{book}
\usepackage[colorlinks,hulogo]{ssithesis}
```

となり、Lua \LaTeX + `book.cls` を使う場合は、

```
\documentclass[a4paper,11pt]{book}
\usepackage[colorlinks,hulogo]{ssithesis}
```

となる。また、pdf \LaTeX + `scrbook.cls` を使う場合は、

```
\documentclass[a4paper,pdftex,chapterprefix=on]{scrbook}
\usepackage[colorlinks,hulogo]{ssithesis}
```

^{*1} 逆に言えば、`luatexja-preset` を読み込めば、日本語文字も自由に入れることが可能である。

となり、Lua \LaTeX + scrbook.cls を使う場合は、

```
\documentclass[a4paper,chapterprefix=on]{scrbook}
\usepackage[colorlinks,hulogo]{ssithesis}
```

となる。

(pdf) \LaTeX と (scr)book.cls の組み合わせにおいて一部に日本語文字を入れる必要がある場合は、CJKutf8.sty パッケージを利用すると良いかもしれない。Lua \LaTeX と (scr)book.cls の組み合わせの場合で、ちょっとだけ日本語フォントを使用したい場合は font 定義で日本語フォントを直接指定してしまえば可能である。たとえば、プリアンブルで

```
\newfontfamily\haranoajimin{Harano Aji Mincho}
```

とし、{\haranoajimin あいうえお}とすればよい。ただし、その場合は日本語としての禁則処理や日本語文字と欧文文字間のグルー挿入などは一切なされないので注意。

5.2 プリアンブル

```
\thesistype{\mastersthesis}
\title{}
\etitle{Usage of the style file for master's thesis
for Course of Systems Science and Informatics}
\author{}
\eauthor{Yuh Yamashita}
\jabstract{}
\eabstract{....}
\jkeyword{}
\keyword{\LaTeX, master's thesis, style file}
\studentno{000000}
\committeemember{Professor John Doe}{1}
\committeemember{Professor William William}{2}
\committeemember{Professor Taro Hokudai}{}
\committeemember{Associate Professor Hanako Hokkai}{}
\writtenyear{2020}
\writtenmonth{12}
\writtenday{6}
```

のように書く。title, author, jabstract, jkeyword は参照されない。2 ページ目 (表紙の見返しページ) も英語の内容になる。

5.3 本文

本文の書き方は、日本語と同様である。ただし、

第〇部	⇒	Part 〇.
第〇章	⇒	Chapter 〇.
目次	⇒	Contents
図目次	⇒	List of Figures
表目次	⇒	List of Tables
参考文献	⇒	References
謝辞	⇒	Acknowledgements
付録	⇒	Appendix
索引(未検証)	⇒	Index
図(キャプション中)	⇒	Figure
表(キャプション中)	⇒	Table

となる。表 4.3 も参照されたい。また、日本語論文ではゴシック体・サンセリフ体が使われるところでは、`book.cls` では通常のセリフ体のボールドが使われる。英語論文でサンセリフ体のセクション見出し等を使う場合は、`scrbook.cls` の方を使うと良い。`book.cls` では、章・節の最初の段落の字下げはされない (`book.cls` の標準: 変えたい場合は `\usepackage{indentfirst}` とすること) が、`scrbook.cls` ではどの段落も 1 行目字下げが行われる。本スタイルファイルでは、`fancyhdr.sty` を読み込んでいるが、小文字 → 大文字の変換はしないようになっている*2。書いたままの内容が奇数ページヘッダに出る。

5.4 参考文献の表題

現状では、参考文献の表題は“**References**”である。これを“**Bibliography**”としたいときは、`LuaLaTeX`, `X2LaTeX` の場合、

```
\renewcommand{\bibname}{Bibliography}
```

(`pdfLaTeX`) の場合は、

```
\setlocalecaption{english}{bib}{Bibliography}
```

とブリアンプルに書く。`References` (あるいは `Reference List`) と書いた場合は、本文中に `\cite{...}` で引用したものだけリストに掲載すべきである。`Bibliography` と書いた場合は、著者が参考にした文献で本文中で引用していないものもリストに掲載できる、とされている。ど

*2 この点は、旧スタイルファイルと異なる

ちらを採用するのかに関しては、主査の意見をよく聞くべきである。

謝辞

謝辞は、\acknowledgements の後を書く。番号は付かないが、Chapter 扱いである。

参考文献

- [1] 磯崎秀樹: \LaTeX 自由自在, サイエンス社, 1992.
- [2] 奥村晴彦: \LaTeX 美文書作成入門, 技術評論社, 1991.
- [3] 某 ZR (@zr_tex8r): \LaTeX の「アレなデフォルト」傾向と対策, [https://qiita.com/zr_text8r/items/297154ca924749e62471](https://qiita.com/zr_tex8r/items/297154ca924749e62471).
- [4] Google Noto Fonts, <https://www.google.com/get/noto/>.
- [5] Insert characters via CID in XeTeX, <https://tex.stackexchange.com/questions/482446/insert-characters-via-cid-in-xetex>.
- [6] Takahiro Suzuki: BibLatex にて日本語文献を引用する方法 (ya-mat 氏作成記事), <http://suzukitakahiro.sakura.ne.jp/tech3/>.

付録 A

Appendix について

付録は、 \LaTeX ファイルの終わりに、

```
\appendix  
\chapter{1 番目の付録のタイトル}  
...  
\chapter{2 番目の付録のタイトル}  
...
```

のように書く。参考文献や謝辞の後に書くことに注意。

A.1 付録セクション

付録に節をつけることも可能。

$$e^x = \sum_{i=0}^{\infty} \frac{1}{i!} x^i \tag{A.1}$$

式番号もちゃんと処理される。