

学位論文（修士）

ベイズ推定を用いたゆで卵の加熱時間の
最適化に関する研究

2025年2月

福井大学大学院工学研究科
安全社会基盤工学専攻原子力安全工学コース

ホゲ ホゲたろう

目次

第 0 章	このテンプレートの説明	5
0.1	修論の構成	5
0.1.1	LaTeX のエンジン	5
0.1.2	ファイル分割の詳細	6
0.1.3	参考文献の出力	6
0.2	LaTeX の各パッケージと使い方	7
0.2.1	zebra-goodies	7
0.2.2	cleverref	7
第 1 章	序論	8
1.1	本研究の背景	8
1.2	本研究の目的	8
第 1 章の参考文献		8
第 2 章	解析手法	9
2.1	解析条件	9
2.2	はじめに	9
2.3	直接法	9
2.3.1	ガウスの消去法	10
2.4	反復法	10
2.5	乱流解析	10
2.6	輻射伝熱	10
第 2 章の参考文献		10
第 3 章	データ同化	11
3.1	局所アンサンブル変換カルマンフィルタ	11
3.1.1	局所アンサンブル変換カルマンフィルタの理論	11
第 3 章の参考文献		12
第 4 章	解析結果	13

4.1	テスト 1	13
第 5 章	結論	14
付録 A	模式図の例	16

図目次

4.1	図のキャプション	13
-----	----------	----

表目次

2.1 実験条件	9
--------------------	---

第 0 章

このテンプレートの説明

0.1 修論の構成

修論、卒論とともに通常の学術論文と同じように

1. 序論（背景、既存の研究のレビューと目的）
2. 研究手法
3. 結果と考察
4. 結論
5. その他（参考文献、謝辞、付録等。ただし、参考文献は必須）

の順序で書きます。（実際には「その他」という章は普通作りません）それに対応して、このテンプレートでは TeX のソースファイルを

- introduction.tex
- solution.tex
- result.tex
- conclusion.tex

にしています。参考文献は章ごとに出力するようにしています。

TeX の場合、ソースファイルをどう分割しても特に問題ないので、手法や結果がながければ、さらに細かく分割したほうが見やすいです。

0.1.1 LaTeX のエンジン

このテンプレートは LuaLaTeX でコンパイルすることを想定しています。LuaLaTeX は、Unicode に対応しており、日本語の扱いが楽です。また、最近の TeX エンジンなので、今後も発展していくことが期待されます。TeX Live 2023 以降であれば、特に追加のインストールなしで使えます。LuaLaTeX 用の jsbook クラスとして、ltjsbook を使っています。

注意としては、lualatex の処理は pdflatex の処理よりも遅いです。大きな文書をコンパイルするときに

は、platex よりも時間がかかることがあります。下記のファイル分割を適宜使いながら、コンパイル時間を短縮してください。

0.1.2 ファイル分割の詳細

通常、ソースファイルを分割する場合、`\include \input` 命令を使います。ですが、ここでは `subfiles` パッケージを使っています。`subfiles` パッケージを使うと、分割した各ソースファイルを単独でコンパイルできるようになります。例えば、`introduction.tex` を単独でコンパイルすると、`introduction.tex` だけで完結した PDF が生成されます。これにより、部分的にコンパイルして確認したい場合に便利です。

`subfiles` でファイルを分割する場合、`Main` のソースファイルでは、`\subfile{ファイル名.tex}` とします。`\input` 等と異なり、拡張子の`.tex` まで必要です。分割したファイル側には、`\documentclass[./Main.tex]{subfiles}` をはじめに書きます。`Main.tex` が親ファイルであることを指定するためです。このテンプレートのファイルを確認してください。分割したファイル単独でコンパイルできるようになります。コード 1 に分割した TeX のソース例を示します。最後の `if` なんちゃらは、大本の `Main.tex` をコンパイルしたときには無視されます。

```
\documentclass[./Main.tex]{subfiles}
\graphicspath{{./Fig/}}
\begin{document}

% この間に本文を書く

\ifSubfilesClassLoaded{
\printbibliography[title=参考文献]
}{}

\end{document}
```

コード 1 `subfile` で分割した TeX のソース

0.1.3 参考文献の出力

参考文献は `biblatex` で出力しています。`preamble.tex` を確認してもらえば分かります。`biblatex` は `bibtex` より新しい機能なので、細かい指定が可能です。

`biblatex` の設定をしているのは `preamble.TeX` の

```
\usepackage[style=phys, biblabel=brackets, backend=biber, refsection=chapter]{biblatex}
```

の部分です。`style=phys` は物理学系の引用スタイル、`biblabel=brackets` は参考文献番号を角括弧で囲む指定、`backend=biber` は `biber` という参考文献処理プログラムを使う指定、`refsection=chapter` は章

ごとに参考文献を分ける指定です。refsection の指定を外せば、文章全体の参考文献リストになります。
bib ファイルの処理には biber というコマンドを使うので、自分でコンパイルするときには以下の流れになります。Main のファイルが Main.tex として、

```
lualatex Main.tex  
biber Main  
lualatex Main.tex  
lualatex Main.tex
```

の順でコンパイルします。最初の lualatex で aux ファイルが生成され、biber で参考文献データが処理され、その後の lualatex で参考文献が出力されます。2 回 lualatex を回すのは、参照番号などを正しく反映させるためです。

0.2 LaTeX の各パッケージと使い方

0.2.1 zebra-goodies

todo や comment を本文に挿入するためのパッケージ。TeX のソースに

```
\todo{ここに TODO を書く}  
\comment{XX について要確認}
```

という感じで、コマンドを入れると、PDF に赤字で TODO やコメントが挿入されます。

原稿が完成して、todo やコメントを一括で消したいときには

```
\usepackage[final]{zebra-goodies}
```

でコンパイルすれば、todo や comment が消えます。

0.2.2 cleverref

図番号・表番号を参照するのに通常は\ref{}を使いますが、\cref{}を使うと、自動的に「図」「表」などの接頭語が付きます。例えば、\cref{fig:example}とすると、「図 1」のように出力されます。

第 1 章

序論

1.1 本研究の背景

[1]

1.2 本研究の目的



[TODO 1: 目的について記載]

第1章の参考文献

- [1] X. Liu, S. Kelm, M. Kampili, G. V. Kumar, and H.-J. Allelein, “Monte carlo method with snbek nongray gas model for thermal radiation in containment flows”, Nuclear Engineering and Design **390**, 111689 (2022) (cit. on p. [8](#)).

第 2 章

解析手法

2.1 解析条件

表 2.1 に条件を示す。

表 2.1 実験条件

項目	内容
ケース 1	乱流
ケース 2	層流

2.2 はじめに

陰解法を用いた場合、行列の反転、すなわち連立一次方程式を解く必要がある。連立一次方程式の解法には直接法と反復法がある。直接法では厳密に解が求められるが、計算量が多い。反復法は反復計算により近似解を求める方法であり、係数行列の特性に応じて、高速に解を得られることがある。CFDでは反復法が多く用いられる。行列の特性によって収束の速度に違いが出ると考えられるので、各手法を適用し、最適なものを選択する。ここでは式 (2.1) で表される n 次元の連立一次方程式を解くことを考える。

$$Ax = b \tag{2.1}$$

A は $n \times n$ の係数行列、 x は $n \times 1$ の解ベクトル、 b は $n \times 1$ のベクトルを表す。

式 (2.1)

この章では主に文献 [1] を参考にした。

2.3 直接法

連立 1 次方程式を行列演算によって直接解く、直接法であるガウスの消去法について述べる。

2.3.1 ガウスの消去法

ガウスの消去法は、係数行列 A を一度上三角行列に変形し、それから単位行列 I を求めることで解を得る。

ガウスの消去法による演算量は $O(n^3)$ に比例して増大するため、次元の大きな連立方程式を解くのは非現実的となる。このため CFD では反復法が多く用いられる。

2.4 反復法

反復法では、初期値として適当な値を与え、真の解に収束する近似解を反復計算により求める。ただし、係数行列によっては反復計算が収束せず、解を得ることができない場合がある。反復解が収束するためには係数行列が対角優位であることが必要である。対角優位とは、行列の対角成分が他の成分に比べて、その絶対値が大きいような行列である。

$$|a_{ii}| > \sum_{i \neq j} |a_{ij}|, \quad (i = 1, \dots, n) \quad (2.2)$$

ここで a_{ij} は係数行列の要素。

2.5 乱流解析

文献 [2]

2.6 輻射伝熱

[3]

第2章の参考文献

- [1] R. I. Issa, “Solution of the implicitly discretised fluid flow equations by operator-splitting”, Journal of Computational Physics **62**, 40–65 (1986) (cit. on p. 9).
- [2] 梶島岳夫, *乱流の数値シミュレーション*, 初版 (養賢堂, 1999) (cit. on p. 10).
- [3] M. F. Modest and S. Mazumder, *Radiative heat transfer* (Academic press, 2021) (cit. on p. 10).

第3章

データ同化

3.1 局所アンサンブル変換カルマンフィルタ

3.1.1 局所アンサンブル変換カルマンフィルタの理論

本研究では、データ同化手法の一つである局所アンサンブル変換カルマンフィルタ (Local Ensemble Transform Kalman Filter: LETKF) を適用した ([1])。

アンサンブル変換カルマンフィルタでは、 N 個のアンサンブルメンバーと呼ばれる複数の状態ベクトルからその流れ場の平均及び分散を求め、これに対して不確かさを持つ観測データを与えて、確率的にもっともらしい流れ場を推定する。以下でその手法を簡単に説明する。

N 個のアンサンブルメンバーを使用して、 m 次元の状態ベクトルの推定することを考える。時刻 t における予測値 $\mathbf{x}_t^{f(i)}$ は

$$\mathbf{x}_t^{f(i)} = \mathbf{M} \mathbf{x}_{t-1}^{a(i)} \quad (3.1)$$

と表される。ここで、上付き文字の f および a はそれぞれ予測 (forecast) と解析 (analysis) の値であることを示し、下付き文字の t は時刻を示す。また、 \mathbf{M} は状態ベクトルを発展させる関数、ここでは CFD の支配方程式のことを指す。予測値の平均と不確かさは以下に示すアンサンブル平均と誤差共分散行列で表される。

$$\overline{\mathbf{x}^f} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}^{f(i)} \quad (3.2)$$

$$\mathbf{P}^f = \frac{1}{N-1} \delta \mathbf{X}^f (\delta \mathbf{X}^f)^T \quad (3.3)$$

第3章の参考文献

- [1] B. R. Hunt, E. J. Kostelich, and I. Szunyogh, “Efficient data assimilation for spatiotemporal chaos: a local ensemble transform kalman filter”, *Physica D: Nonlinear Phenomena* **230**, 112–126 (2007) (cit. on p. 11).

第 4 章

解析結果

4.1 テスト 1

図 4.1 に示す。

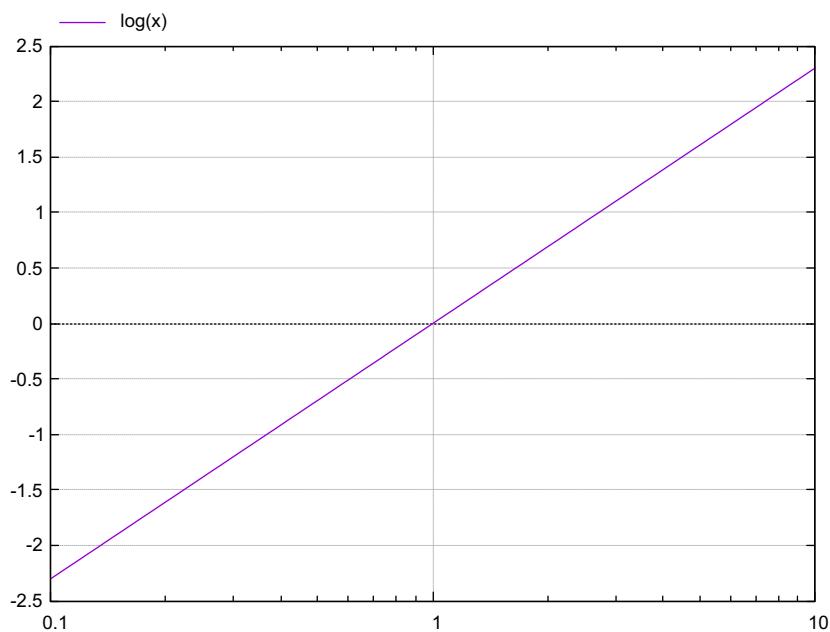


図 4.1 図のキャプション

図 4.1

第 5 章

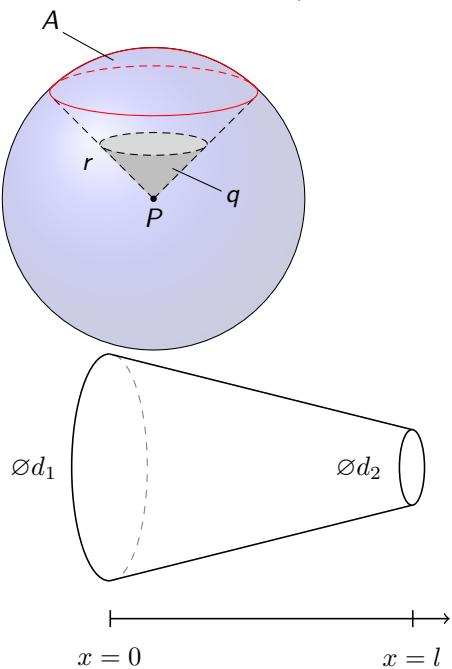
結論

謝辞

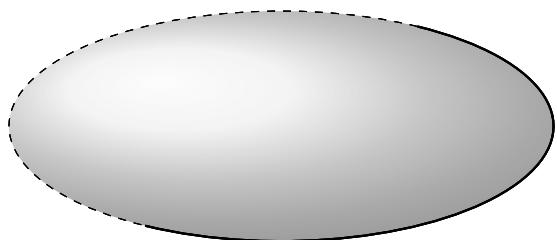
付録 A

模式図の例

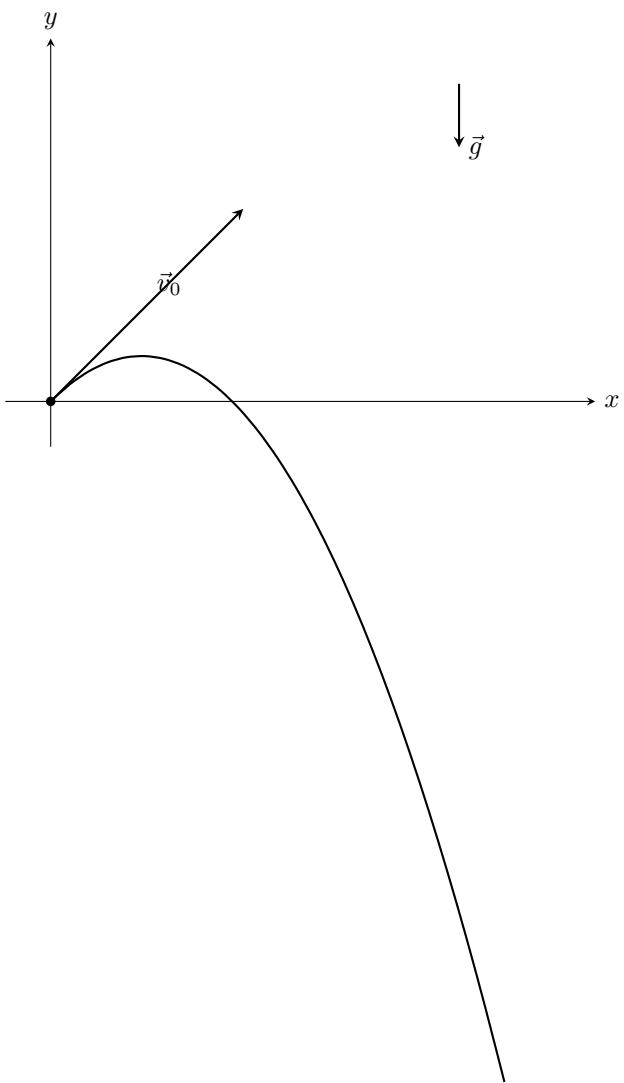
tikz というライブラリを使うと、コマンドを TeX のソースファイルに記述することで、図が書けます。以下の図がその例。一部、GPT で生成したものもあるので、若干変です。



N



S



.....

Zebra Notes

Type	Number
todo	1