

# 計算物理学II

## 第3回 LaTeXによる科学文書作成

秋山 進一郎

2025年10月17日

# 授業日の確認

- ・全10回

- ・第1回：10月3日（金）

- ・第2回：10月10日（金）

- ・第3回：10月17日（金）

- ・第4回：10月24日（金）★

- ・第5回：10月31日（金）

- ・第6回：11月14日（金）★

- ・第7回：11月21日（金）

- ・第8回：12月5日（金）★

- ・第9回：12月12日（金）

- ・第10回：12月19日（金）★

- ・★の付いた授業にてレポート課題を配布予定

- ・次回（10/24）は自習とします
- ・各自でレポート課題に取り組み、11/7までに提出して下さい

# 今日の授業の目標

- **LaTeXを使う環境を整える**
  - シェルスクリプトの利便性を理解する
  - VS codeの設定（授業中）
  - Overleafの利用（授業中 or 授業後）
- 科学文書作成の練習

# この授業のレポート課題について

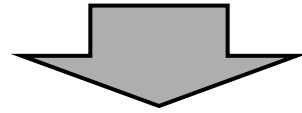
- 4回ともLaTeXで作成して提出してください
- **LaTeXの使用自体を課題の一部とします**
- 他の授業のレポート課題でもどんどん使ってみてください

# LaTeX入門のための簡単な手引き

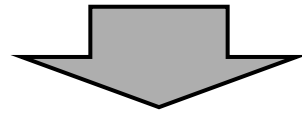
- **compphy2/intro\_latex/introduction\_latex.pdf**  
というドキュメントがあります
- **本日の講義に限らず，必要に応じて適宜参照してください**
- **ミニマルな情報のみなので，適宜ネット検索をしましょう**

# 毎回の授業の流れ

**“compphy2”に移動する**



**“git fetch”と“git merge”で演習資料を入手**



**“lecture\*”に移動し、  
“lecture\_material\_\*.pdf”を開いて演習開始**

# 本日の演習内容

- LaTeXを使う環境を整える
  - シェルスクリプトの活用
  - VS codeの設定
  - Overleafの利用（時間に余裕があれば）
- LaTeXを使ってみる
  - 数式の記述
  - 図の挿入

# LaTeXとは？

- L. Lamportによって開発された文書清書のためのソフトウェア
  - D. Knuthによって開発されたTeX（テック, テフ）を使いやすく拡張したもの
- LaTeXによる基本的な文書作成の流れ
  - ① .texファイルを作成, 編集する
  - ② .texファイルをplatexコマンドでコンパイルする
  - ③ .dviファイルが生成される
  - ④ dvipdfmxコマンドを使い, .dviファイルを.pdfファイルへ変換する
  - ⑤ 完成した.pdfファイルの確認, 閲覧



# Terminalからコンパイルしてみよう 1/2

- lecture3の中に, ex\_terminalというディレクトリを作る

```
akiyama@:~/compphys2/lecture3$ mkdir ex_terminal
```

- lecture3/src/main.texをex\_terminalへcpしましょう

```
akiyama@:~/compphys2/lecture3$ cd ex_terminal/  
akiyama@:~/compphys2/lecture3/ex_terminal$ cp -r ../src/main.tex ./
```

- main.texをlatexコマンドでコンパイルしてみる

```
akiyama@:~/compphys2/lecture3/ex_terminal$ latex main.tex
```

- lsでディレクトリ内を確認し, main.dviというファイルができていればOK

```
akiyama@:~/compphys2/lecture3/ex_terminal$ ls  
main.aux main.dvi main.log main.out main.tex
```

# Terminalからコンパイルしてみよう 2/2

- 続いて, dvipdfmxコマンドを使って, main.dviをmain.pdfに変換する

```
akiyama@~/compphys2/lecture3/ex_terminal$ dvipdfmx main.dvi
main.dvi -> main.pdf
[1]
4839 bytes written
```

- lsでディレクトリ内を確認し, main.pdfというファイルができたことを確認する

```
akiyama@~/compphys2/lecture3/ex_terminal$ ls
main.aux  main.dvi  main.log  main.out  main.pdf  main.tex
```

- main.pdfを開いてみましょう
  - 右のようなpdfファイルが見えれば成功です

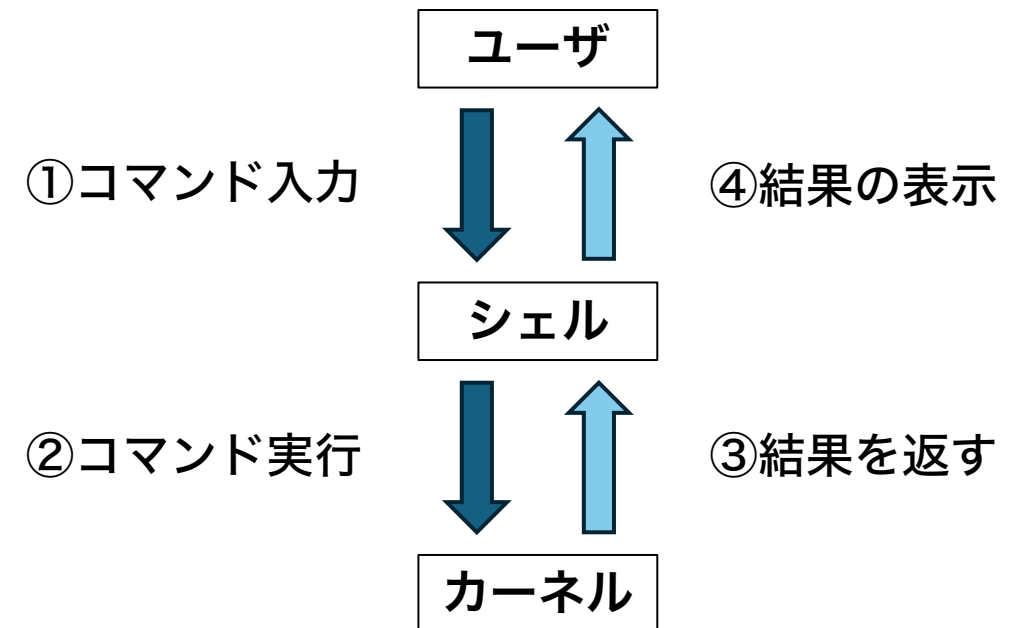


# より実践的なLaTeXの使い方

- 今やったように, terminalからtexファルをコンパイルし, pdfファイルを作るには複数のコマンドを立て続けに行う必要がある
  - 実際に文書作成を行う時は頻繁にコンパイルをするので面倒ですね
  - 文書の作成にはエディタ, コンパイルにはterminal, できた文書の確認にはビューアーを使うため, これらを行き来する必要がある → これも面倒ですね
- そこで, 以下のツールを使ってこれらの面倒を解消していきます
  - シェルスクリプト
  - VS code
  - Overleaf
- 以下, LaTeXによる文書作成を今後快適に行うための準備を進めていきます

# シェルスクリプトとは？

- ・ シェルが解釈可能なコマンドを複数集めて作ったファイルのこと
  - ・ シェルとは，ユーザからコマンド入力を受け付けるシステムのこと
  - ・ 入力されたコマンドはカーネル（計算機の心臓部分）で実行される（下図参照）
- ・ シェルスクリプトを予め書いておけば，そこに書かれた複数のコマンドが順番に実行される
  - ・ 同じコマンドを何回も入力しなくて済む



# LaTeX (latexmk) 用のスクリプトを準備する 1/2

- lecture3の中にあるsrcディレクトリにlatexmkrcというファイルが入っている
  - srcディレクトリに移動し, latexmkrcを一度眺めておこう
- l.3でplatexコマンドをいくつかのオプション付きで実行し, l.6で.dviから.pdfへの変換を行う
- その他の部分で後々のための設定をいくつか行なっているが, 今は理解しなくて問題ありません

```
home > akiyama > compphys2 > lecture3 > src > latexmkrc
1  #!/usr/bin/env perl
2
3  $latex = 'platex -synctex=1 -halt-on-error -file-line-error %0 %S';
4  $max_repeat = 5;
5  $bibtex = 'pbibtex %0 %S';
6  $biber = 'biber --bblencoding=utf8 -u -U --output_safechars %0 %S';
7  $makeindex = 'mendex %0 -o %D %S';
8  $dvi2pdf = 'dvi2pdf %0 -o %D %S';
9  $pdf_mode = 3;
10
11 $pvc_view_file_via_temporary = 0;
12 if ($^O eq 'linux') {
13     $dvi_previewer = "xdg-open %S";
14     $pdf_previewer = "xdg-open %S";
15 } elsif ($^O eq 'darwin') {
16     $dvi_previewer = "open %S";
17     $pdf_previewer = "open %S";
18 } else {
19     $dvi_previewer = "start %S";
20     $pdf_previewer = "start %S";
21 }
22
23 $clean_full_ext = "%R.synctex.gz"
24
```

# LaTeX (latexmk) 用のスクリプトを準備する 2/2

- latexmkrcを次のようにホームへコピーする

```
akiyama@:~/compphys2/lecture3/src$ cp ./latexmkrc ~/.latexmkrc
```

ピリオドに  
注意！

- lecture3へ戻り, ex\_latexmkというディレクトリを新しく作ろう
- lecture3/src/main.texをex\_latexmkへcpしましょう

```
akiyama@:~/compphys2/lecture3$ mkdir ex_latexmk  
akiyama@:~/compphys2/lecture3$ cd ex_latexmk/  
akiyama@:~/compphys2/lecture3/ex_latexmk$ cp ../src/main.tex ./
```

# latexmkで一発でコンパイルする

- lsを使って、ディレクトリ内にmain.texしかないことを確認する

```
akiyama@~/compphys2/lecture3/ex_latexmk$ ls  
main.tex
```

- 以下のコマンドを実行することで、先ほどホームに置いた.latexmkrcが実行される

```
akiyama@~/compphys2/lecture3/ex_latexmk$ latexmk main.tex
```

- lsで中を確認し、main.pdfが出来ていれば成功

```
akiyama@~/compphys2/lecture3/ex_latexmk$ ls  
main.aux main.dvi main.fdb_latexmk main.flx main.log main.out main.pdf main.synctex.gz main.tex
```

- main.pdfも開いて確認しておこう
- ここまで出来たら、一度lecture3ディレクトリに戻りましょう (cd ../)

# VS codeの設定 1/6

- ・スクリプトによりコンパイルの手間は大幅に削減された
- ・次はtexファイルを編集していく環境を整えよう
  - ・この授業ではVS codeを使うことにする
  - ・VS codeでは便利な拡張機能が使えるため
- ・以下のコマンドを実行してVS codeを立ち上げよう

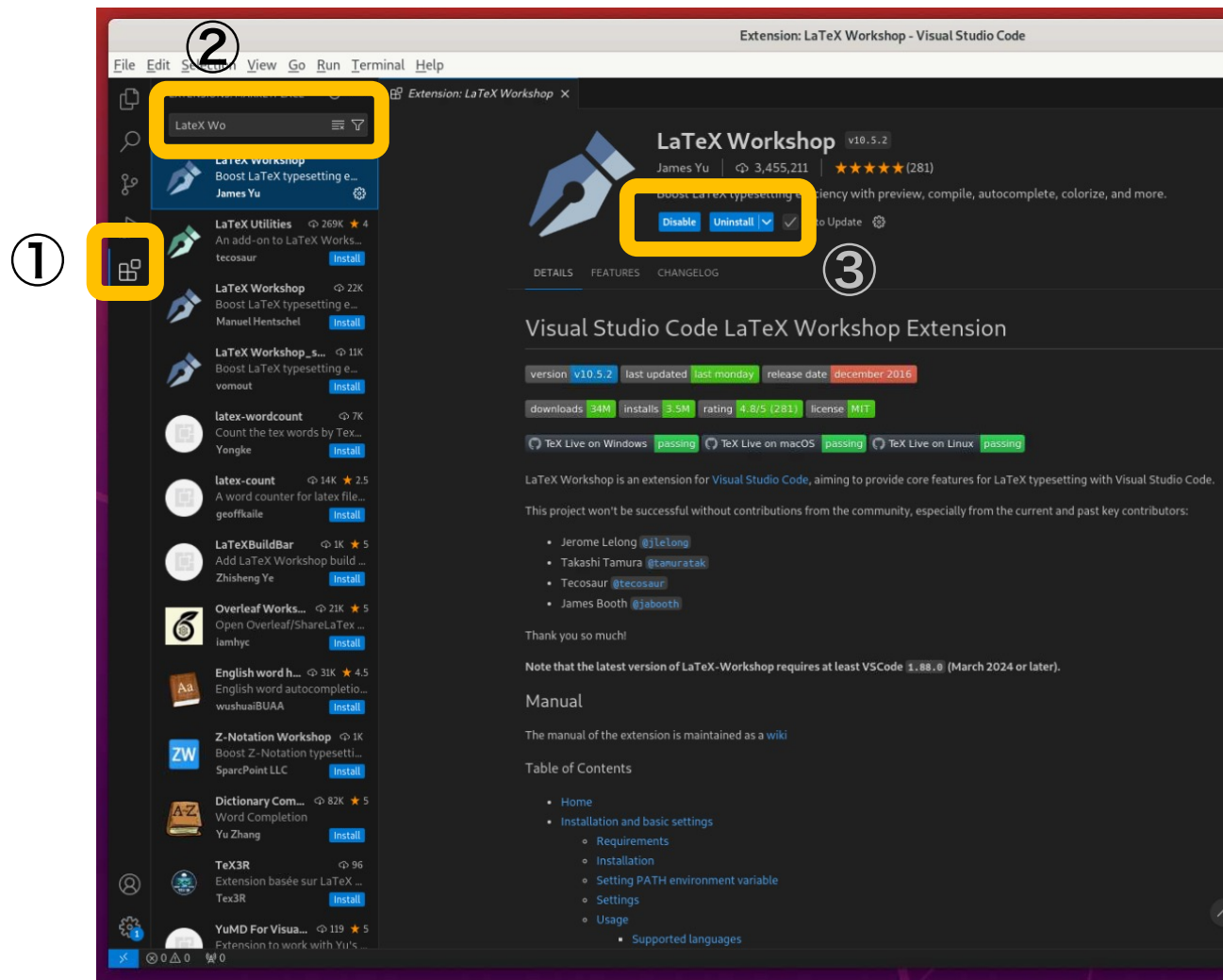
```
akiyama@:~/compphys2/lecture3$ code
```



# VS codeの設定 2/6

- 以下の手順で、拡張機能であるLaTeX Workshopをインストールする

- ① Extensionsをクリック
- ② 検索窓にlatexと入力し、候補に出るLatex Workshopを選択
- ③ 青いinstallボタンをクリック  
(右図ではinstall済となっている)



# VS codeの設定 3/6

- ここで一度terminalに戻ります
  - VS codeの設定は, settings.jsonというファイルを通して行います
  - setting.jsonはVS codeのアプリからも開くことができますが, ここでは練習も兼ねてterminalから行います
- すでに編集されたsettings.jsonがlecture3/srcの中に入っています
  - 配布されたsettings.jsonをVS codeが参照できるように, 以下の手順でcpします

# VS codeの設定 4/6

- ・ホームディレクトリへ移動し, `.config/Code/User`というディレクトリへ入ります
  - ・lsで中を見てみましょう
  - ・人によっては, すでに`settings.json`があるかもしれませんが, 問題ありません

```
akiyama@:~/compphys2/lecture3$ cd  
akiyama@:~$ cd .config/Code/User/  
akiyama@:~/.config/Code/User$ ls  
globalStorage  History  snippets  workspaceStorage
```

- ・ここに配布されている`settings.json`をcpしてきます

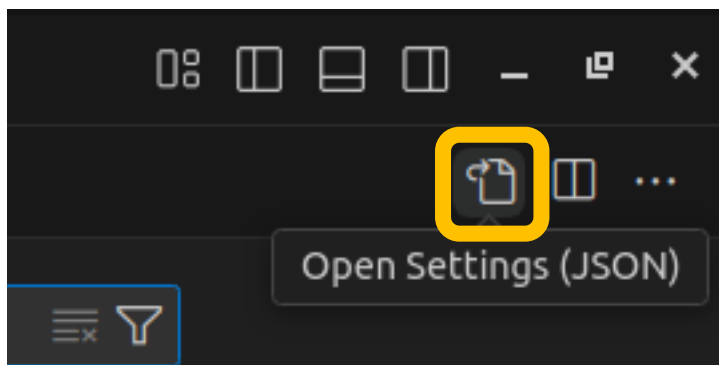
```
akiyama@:~/.config/Code/User$ cp ~/compphys2/lecture3/src/settings.json ./
```

- ・ここまでできたらVS codeの画面に戻ります

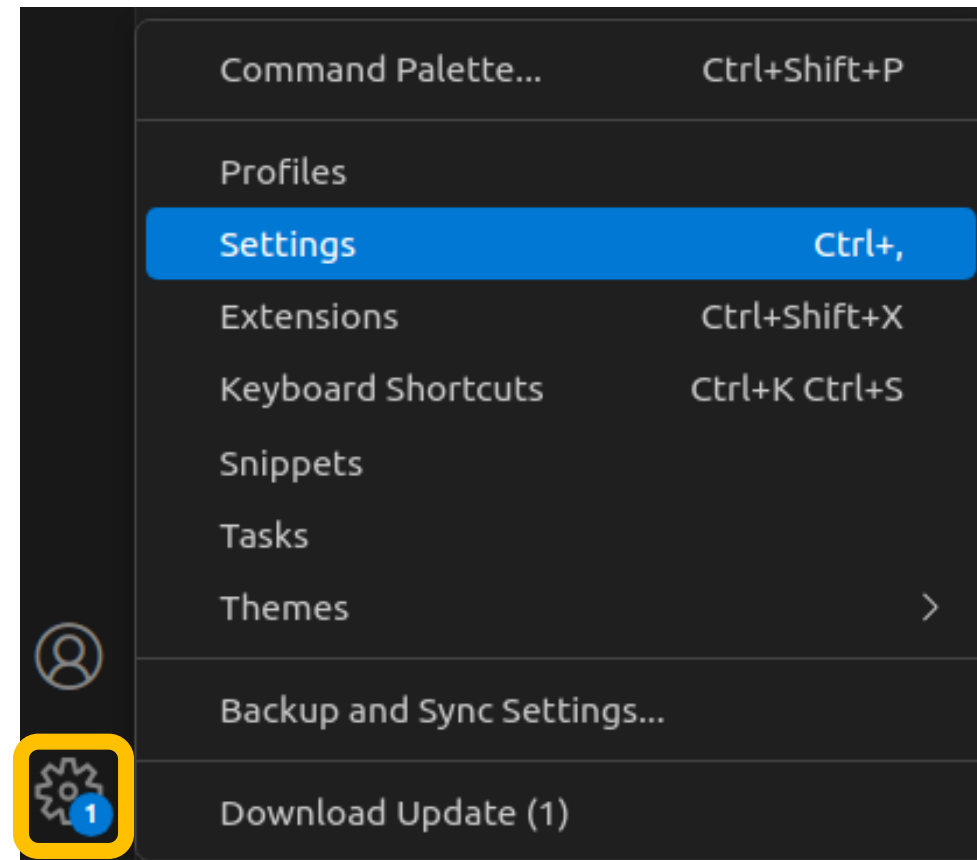
# VS codeの設定 5/6

- Terminalからcpしたsettings.jsonをVS codeから確認します

- ① VS codeの左下にある  マークを押し,  
“Settings”を選択 (右図参照)
- ② VS codeの右上に“Open Settings(JSON)”  
が表示されるので選択 (下図参照)



②



①

# VS codeの設定 6/6

- 左のようなjsonファイルが見えればOKです
  - 内容の詳細な理解は現時点では不要です
  - 「.jsonファイルを通して, LaTeXの環境を設定できる」ということを頭に入れておいてもらえればOKです
- ここまでできたら, 一度VS codeを終了します
  - これでVS codeの設定が完了しました



```
1  {
2      // TeX関連の設定 //
3      "[tex]": {
4          "editor.suggest.snippetsPreventQuickSuggestions": false,
5          "editor.tabSize": 2
6      },
7
8      "[latex]": {
9          "editor.suggest.snippetsPreventQuickSuggestions": false,
10         "editor.tabSize": 2
11     },
12
13     "[bibtex]": {
14         "editor.tabSize": 2
15     },
16
17     // LaTeX Workshopの設定 //
18     "latex-workshop.intellisense.package.enabled": true,
19
20     "latex-workshop.latex.clean.fileTypes": [
21         "*.aux",
22         "*.dvi",
23         "*.fdb_latexmk",
24         "*.fls",
25         "*.log",
26         "*.out",
27         "*.synctex.gz"
28     ],
29
30     "latex-workshop.latex.outDir": "output",
31
32     "latex-workshop.latex.recipes": [
33         {
34             "name": "latexmk",
35             "tools": [
36                 "latexmk"
37             ]
38         },
39     ],
40 }
```

# VS codeを使ってみよう 1/3

- それではVS codeを使ってLaTeXの編集, コンパイルをしてみましょう
- lecture3に戻り, ex\_vscodeというディレクトリを用意します
- 下のように, main.texをcpしてきましょう

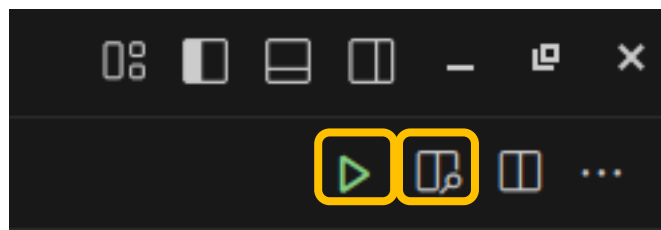
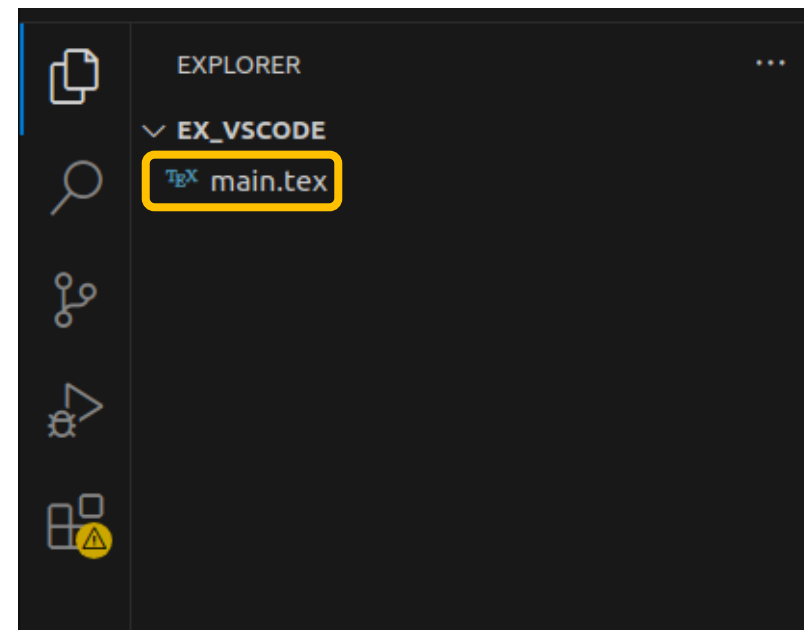
```
akiyama@~/compphys2/lecture3$ mkdir ex_vscode
akiyama@~/compphys2/lecture3$ cd ex_vscode/
akiyama@~/compphys2/lecture3/ex_vscode$ cp ../src/main.tex ./
akiyama@~/compphys2/lecture3/ex_vscode$ ls
main.tex
```

- このディレクトリごとVS codeで開きます

```
akiyama@~/compphys2/lecture3/ex_vscode$ code ./
```

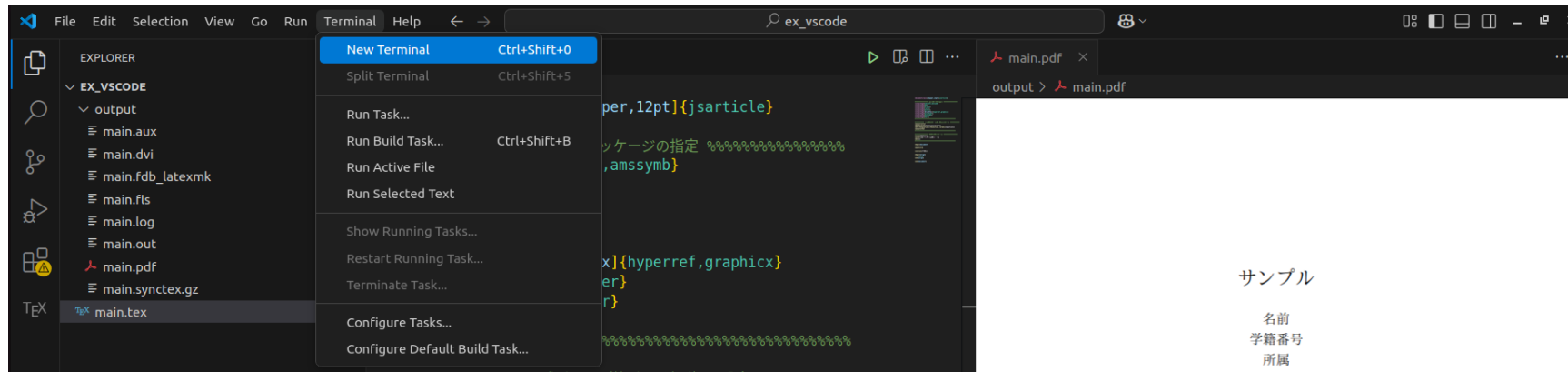
# VS codeを使ってみよう 2/3

- 左のExplorerからディレクトリの中を見ることができます
  - main.texをクリックすると、編集画面で開きます
  - texファイルを開いた状態で、Ctrl+Alt+Bを押すとコンパイルされます ⇒ やってみましょう
    - outputというディレクトリができたなら成功です Explorerを確認しましょう
    - コンパイルされたpdfファイルもこの中にあります
  - Ctrl+Alt+Vを押すと右にプレビューが表示されます
  - これらのコマンドの代わりに、右上のボタンを使うこともできます

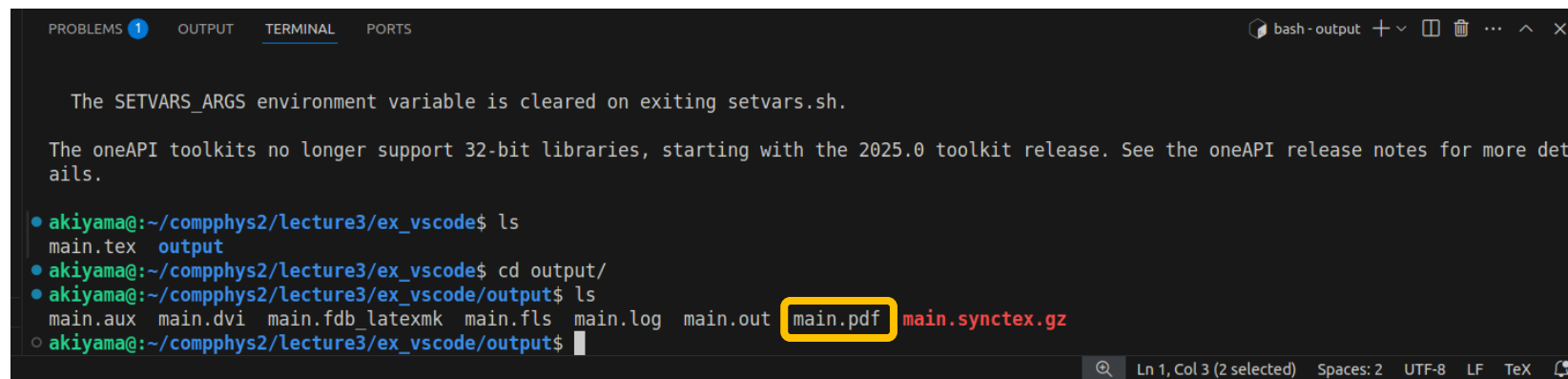


# VS codeを使ってみよう 3/3

- VS codeからTerminalを開くことができる
  - 左上の「Terminal」 → 「New Terminal」を押すかCtrl+Shift+Oを押す



- VS codeのTerminalからもpdfファイルができていることを確認してみましょう





# これでVS codeの準備は完了

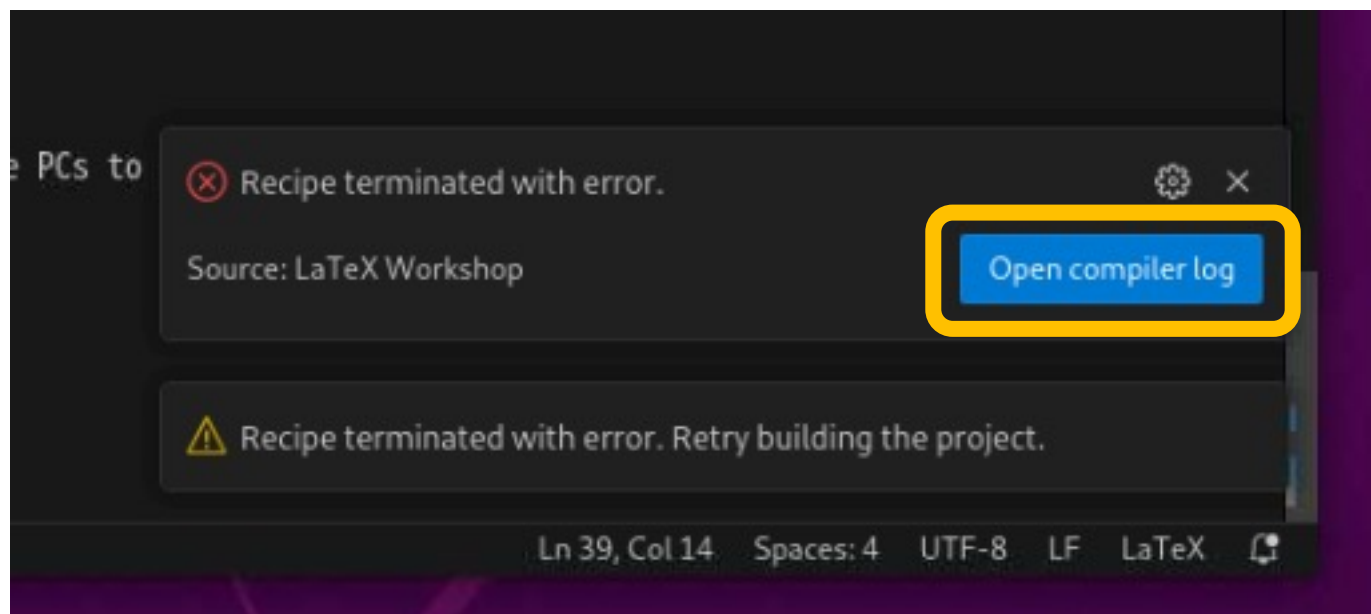
- 以上の準備は一回だけやればよく, 今後はVS codeを使ってtexファイルの編集だけに注力することができます
- それでは, VS codeを使ってLaTeXによる文書作成を実際に進めていきましょう

# 本日の演習内容

- LaTeXを使う環境を整える
  - シェルスクリプトの活用
  - VS codeの設定
  - Overleafの利用（時間に余裕があれば）
- LaTeXを使ってみる
  - 数式の記述
  - 図の挿入

# 以降の演習について

- 今開いているVS codeを使っていきます
- texファイルを編集したら、こまめにCtrl+Alt+Bでコンパイルしましょう
- コンパイルエラーになると、それを知らせるウィンドウが右下に現れます
  - 「Open compiler log」を押して、メッセージを見ましょう
- エラー箇所の特定につながります



# 文書の表題を設定しよう

- ・タイトル, 名前, 学籍番号, 所属, 日付を入力し, **main.texを上書き保存**しましょう
- ・ここでは仮に「計算物理学2第3回」としました
- ・日付はハードコーディングせずに, `\today`を使いましょう
- ・もし日本語入力する方法が分からなければ, [ここ](#) を参照

保存せずにコンパイルすると  
編集内容が反映されないので注意

```

22  %%%%%%%%%%%%%% 文書の表題を設定 %%%%%%%%%%%%%%
23  \title{計算物理学2第3回}
24  \author{
25      秋山進一郎\\
26      000000000\\
27      筑波大学計算科学研究センター
28  }
29  \date{\today}
30  %%%%%%%%%%%%%%
```

計算物理学 2 第 3 回

秋山進一郎

000000000

筑波大学計算科学研究センター

2025 年 10 月 16 日

1 練習 1

$$f(x) = x$$

(1.1)

# 数式を入力しよう

- 例として,  $f(x)=x$  という式が入力してあります
- 配布資料「LaTeX入門」を適宜参照し, 以下の二つに取り組んでみてください
  - 式(1.1)の下に $f(x)=\sin x$ を出力させましょう
  - さらにその下に $\sin x$ の原点周りでのTaylor展開を書いてみましょう

## 1 練習 1

$$f(x) = x \quad (1.1)$$

$$f(x) = \sin x \quad (1.2)$$

$$\sin x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!} \quad (1.3)$$

# 図を挿入してみよう 1/2

- 第2回の演習で作成した図を挿入してみましょう
- 今いるex\_vscodeディレクトリの中にfigという名前のディレクトリを作りましょう
  - VS codeで開いているTerminalを使ってみましょう
- 作成したfigディレクトリに, 前回作成した図をcpしてきましょう
  - 3行目の“\*pdf”は「ファイル名の最後がpdfとなっているすべてのファイル」という意味になります

```
• akiyama@~/compphys2/lecture3/ex_vscode$ mkdir fig
• akiyama@~/compphys2/lecture3/ex_vscode$ cd fig/
• akiyama@~/compphys2/lecture3/ex_vscode/fig$ cp ../../../../lecture2/ex_matplotlib/*pdf ./
• akiyama@~/compphys2/lecture3/ex_vscode/fig$ ls
functions.pdf  noisy_data.pdf
○ akiyama@~/compphys2/lecture3/ex_vscode/fig$
```

## 図を挿入してみよう 2/2

- texファイル上に新しいsectionを作しましょう
  - `\section{練習2}`などとしましょう
- 「LaTeX入門」の6節を見ながら、先ほどコピーしてきた図を挿入しましょう
  - `\caption{}`に図の説明を記述してみましょう
  - コンパイルして図が表示されましたか？

# 特殊文字を含む数式を書いてみよう

- texファイル上に新しいsectionを作しましょう
- `\section{練習3}`などとしましょう
- 「LaTeX入門」を適宜参照しながら，下のような文章とともにSchrödinger方程式を書いてみましょう（öは`\"o`）

## 3 練習 3

量子力学で基本となる方程式は，Schrödinger 方程式と呼ばれ，

$$i\hbar \frac{\partial \Psi}{\partial t} = -\frac{\hbar^2}{2m} \nabla^2 \Psi + V\Psi \quad (3.1)$$

で与えられる．ここで， $\hbar$ は

$$\hbar = \frac{h}{2\pi} = 1.054572 \times 10^{-34} \text{Js} \quad (3.2)$$

である．



## ここまで終わったら

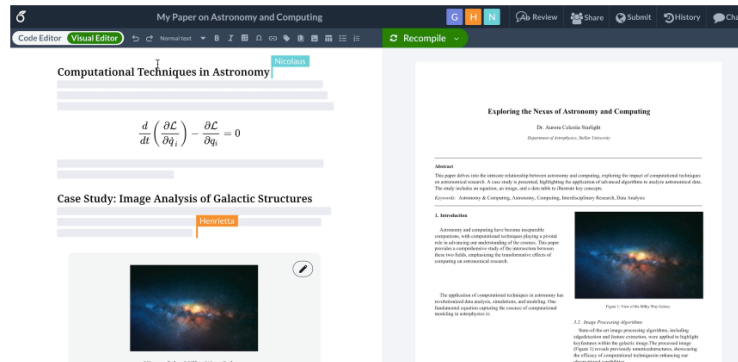
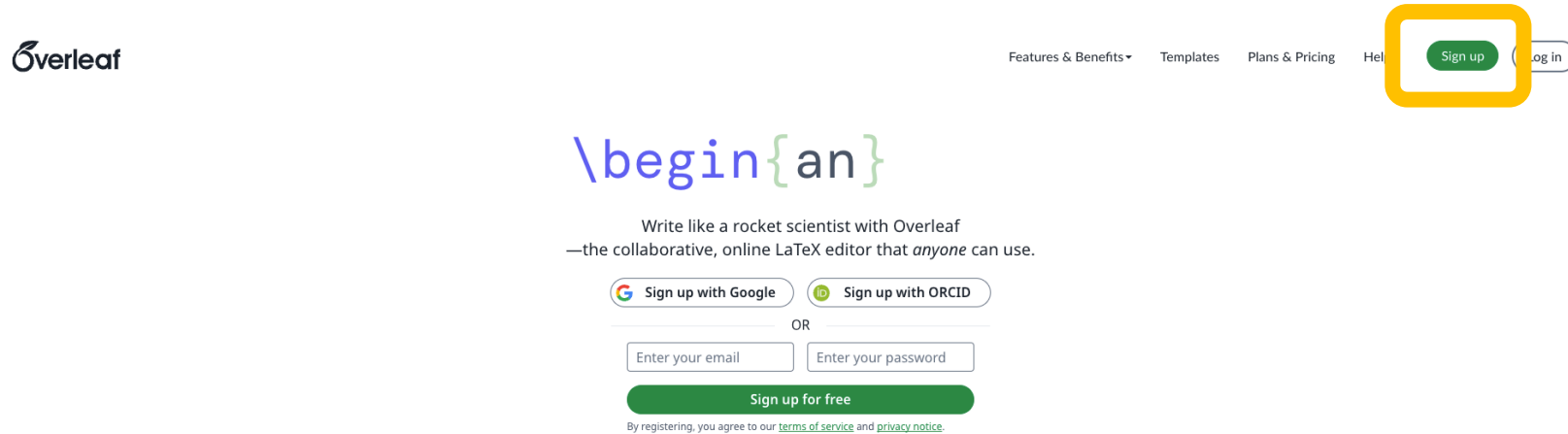
- お疲れ様でした
- 時間に余裕のある人は、以降のページに従い、Overleafも使ってみましょう

# Overleafとは？

- LaTeX専用のオンラインエディタ
  - 環境構築が不要
  - クラウドサービスなので、ネット環境さえあればどこからでも作業可能
  - クラウドサービスなので、自分のパソコンが壊れてもデータは残る
  - Grammarlyが使える
  - 共同編集者を追加できる（無料プランでは1人まで）
  - 無料プランの場合、図の量が増え過ぎるとコンパイルが遅くなるので注意
- 授業時間外ではリモートデスクトップでVS codeを使ってもいいが、素直にブラウザからOverleafを使う方がラクだと思います

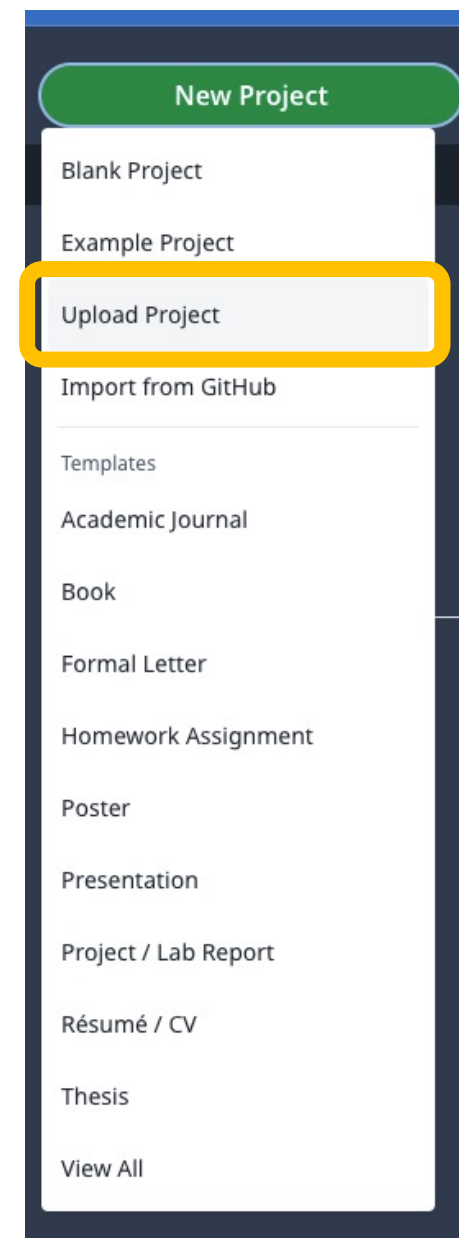
# Overleafのアカウントを作る

- [Overleaf](#)にアクセスし, sign upをクリック
- メールアドレス, パスワードの登録をする
- 初めて登録する人は無料プランを選び, 登録手続きを進めましょう



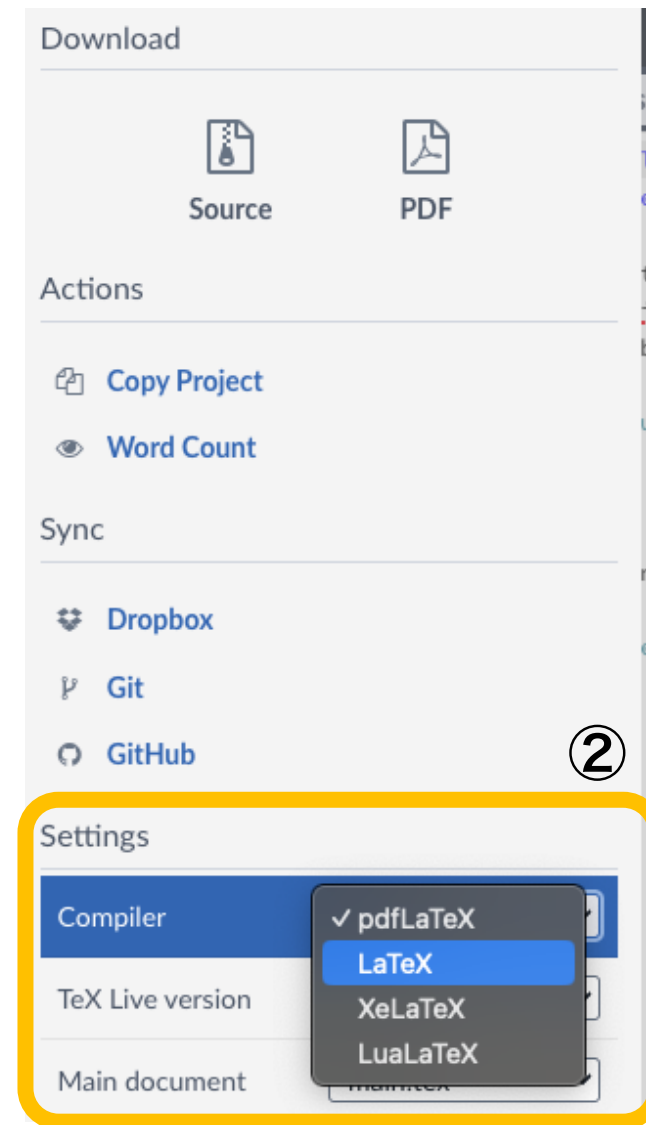
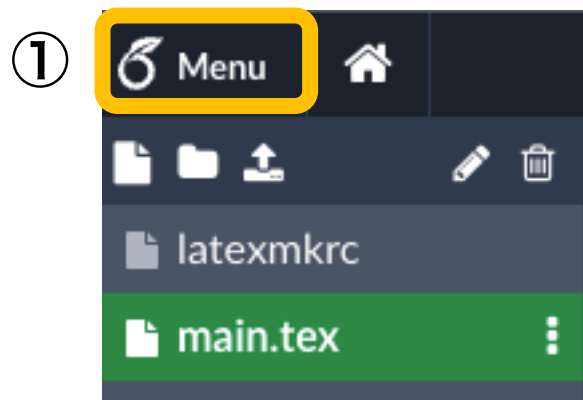
# OverleafでNew Projectを作る

- Overleafにlog inしたらNew Projectを作ってみよう
  - 作り方は色々あるが、ここではUpload Projectを選ぶ
  - lecture3/srcにある“template\_overleaf.zip”をアップロード



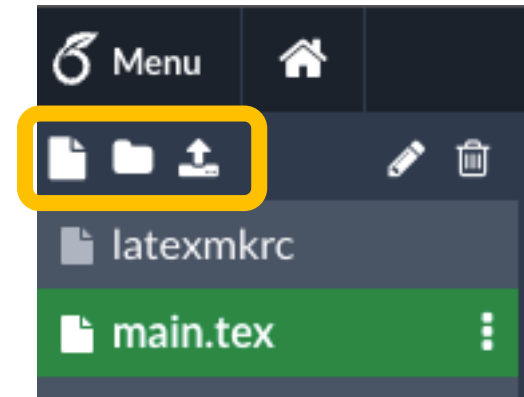
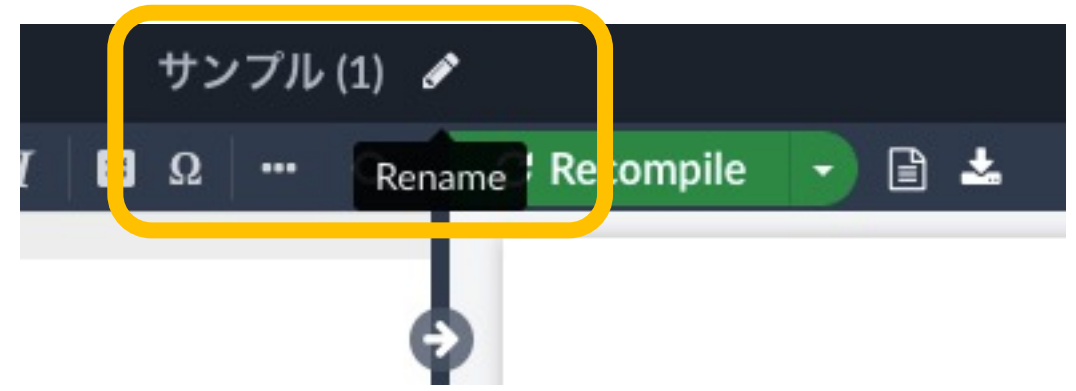
# Overleafで日本語を使えるようにする

- このままでは日本語が使えない
- 設定の変更手順
  - ① 左上の「Menu」を押す
  - ② 「Compiler」を「LaTeX」へ変更
- これで日本語が使える
- 日本語を使うためにはlatexmkrcとcompilerの変更が必要
  - 今回は、こちらで準備したlatexmkrcをtemplate\_overleaf.zipに入れておいた
- 日本語を使いたい場合、新しいProjectを作るとにlatexmkrcとcompilerの変更が必要となる点に注意



# Overleafの使い方

- Ctrl+Sでコンパイルと保存が実行される
- Project名の変更は画面中央上から
- 「Menu」の下ボタンから新規ディレクトリ、ファイルの作成、ファイルのアップロードが可能
- 保存していない状態でブラウザを離れようとするとき保存するか聞いてくるのでよく確認すること



## 参考サイト

- TeX Wiki [latexmk](#) [Visual Studio Code/LaTeX](#)