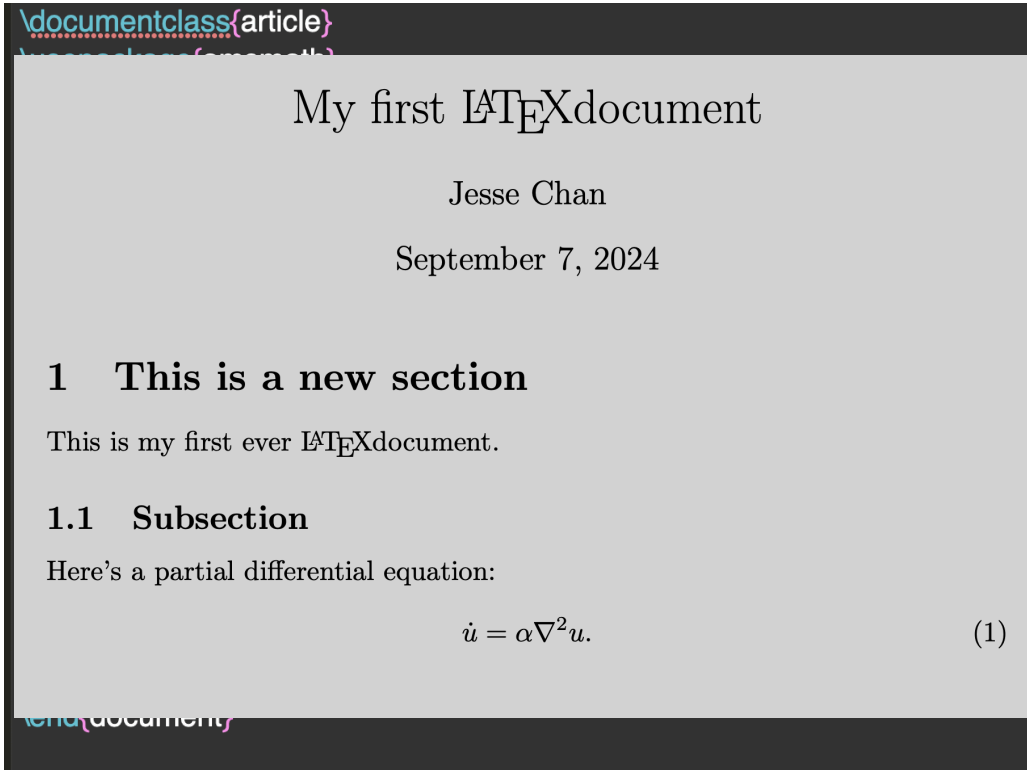
CMOR 420/520
Computational Science
LaTeX and numerical computing

Structure of a LaTeX document

- Preamble: load packages, define commands, …
- Document: sections, mathematics, figures, …
  - backslash “\” denotes a LaTeX command
- Compilation with “pdflatex”
  - “pdflatex simple.tex” creates both a pdf and some auxiliary files.



Preamble packages

- Using mathematical symbols and notations, tables, and images needs the correct packages
- Keep a template document that has all packages you may need in the preamble

```
\documentclass[10pt,letterpaper]{article}
\usepackage[letterpaper, margin=.75in]{geometry}
\usepackage[latin1]{inputenc}
\usepackage{float}
\usepackage{xypic}
\usepackage{graphicx, amsthm, amsmath, amssymb}
\graphicspath{{images/}{../images/}}
\usepackage{mathrsfs }
\usepackage{marginnote}
\usepackage{multicol}
\usepackage{subfiles}

\usepackage{tikz, pgf, calc}
\usetikzlibrary{arrows,matrix,positioning,fit,calc}
\usepackage[linewidth=1pt]{mdframed}
\usepackage{parskip}
\usepackage{cancel}
```

Preamble environments

- You can define environments to give a definition, Theorem, Lemma, etc.
- Keep a template document that has all environments you may need in the preamble

```
%% Theorems %%
\newtheorem{theorem}{Theorem}[section]
\theoremstyle{definition}
\newtheorem{definition}{Definition}[section]
\theoremstyle{definition}
\newtheorem{exmp}{Example}[section]
\newtheorem{prop}{Proposition}[section]
\newtheorem{cor}{Corollary}[section]
\newtheorem{lem}{Lemma}[section]
\newtheorem*{rem}{Remark}
```

LaTeX for mathematics

- Inline math (e.g., “We’ll set $x = 10$ here…”)
- Math “environments” (e.g., “\begin{equation}”, “\begin{align}”, “\begin{gather}”, etc)
  - \begin{align*} does not create equation labels.
- Can reference sections, equations, etc.

```
We can write math ``inline" like $\alpha \in \mathbb{R}$,
or using environments like
\begin{align}
        \beta \in \mathbb{C}^2.      \label{eq:beta}
\end{align}
and reference equations later like (\ref{eq:beta}).
```

We can write math “inline” like $\alpha \in \mathbb{R}$, or using environments like

$$\beta \in \mathbb{C}^2. \tag{1}$$

and reference equations later with (1).

LaTeX special symbols

- Easy to find references for special symbols, fonts, etc.
  - https://www.overleaf.com/learn/latex/List_of_Greek_letters_and_math_symbols
  - https://www.cmor-faculty.rice.edu/~heinken/latex/symbols.pdf

Preamble commands

- You can define 'commands' to simplify typing equations

```
\newcommand{\R}{\mathbb{R}}
\newcommand{\C}{\mathbb{C}}
\newcommand{\N}{\mathbb{N}}
\newcommand{\Q}{\mathbb{Q}}
\newcommand{\sumin}{\sum_{i=1}^n}

\newcommand{\dt}{\Delta t}
\newcommand{\ukone}{u^{k+1}}
\newcommand{\uk}{u^k}
\newcommand{\utkone}{\tilde{u}^{k+1}}
\newcommand{\utk}{\tilde{u}^k}
\newcommand{\uhkone}{\hat{u}^{k+1}}
\newcommand{\ekone}{e^{k+1}}
\newcommand{\ek}{e^k}
\newcommand{\etkone}{\tilde{e}^{k+1}}
\newcommand{\etk}{\tilde{e}^k}
\newcommand{\ehkone}{\hat{e}^{k+1}}
```

LaTeX “floating” environments

- Figures and tables are considered “floats” in LaTeX: they can be moved (floated) around by LaTeX freely
  - “\begin{figure}[h]” requests that the figure be placed **h**ere (**t**op, **b**ottom, top of next **p**age). [!h] strongly requests.
  - Works on all “float” environments
- “\includegraphics” can edit images (crop, change aspect ratio, rotate, …)
  - “\includegraphics” works without the “figure” environment too.

* Include multiple images in one figure by inserting “\includegraphics” more than once in the environment

```
\begin{figure}[h]
\centering
\includegraphics[width=.25\textwidth]{chan_headshot.jpg}
\caption{This is a photo of Jesse Chan.}
\label{fig:jc}
\end{figure}

I can reference Figure~\ref{fig:jc} this way.
```

Figure 1: This is a photo of Jesse Chan.

I can reference Figure 1 this way.

LaTeX tables and alignment

- Tables and environments like “\begin{align}” use the alignment character “&”
- Here, “\begin{tabular}” defines the actual table; “\begin{table}” just creates the float container for the “tabular” instance.
- Double backslash “\\” adds a new line in LaTeX

```
\begin{table}[htbp]
    \begin{tabular}{r|l|cc}
        1 & 2 & 3 & 4 \\ \hline
        5 & 6 & 7 & 8 \\
        9 & 10 & 11 & 12 \\
    \end{tabular}
\end{table}
```

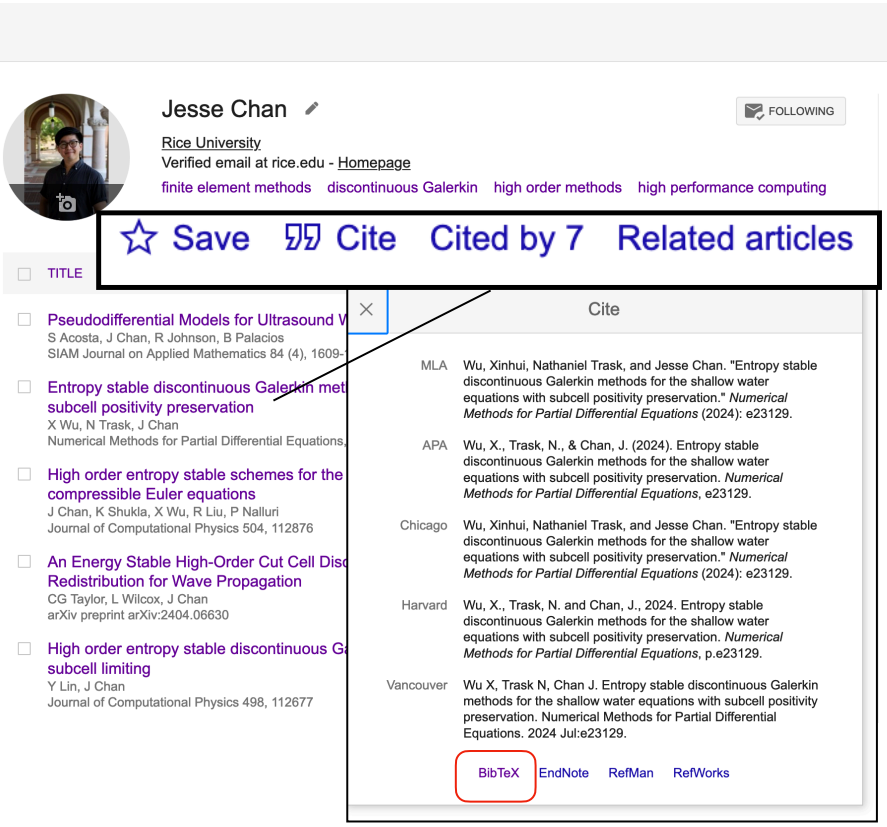
1	2	3	4
5	6	7	8
9	10	11	12

* Include multiple tables in one figure/table by calling “\tabular” more than once in the environment

Bibliographies and Bibtex

- Easy to grab citations from Google Scholar or any journal article
- Ref via \cite{wu2024entropy}

```
@article{wu2024entropy,
  title={Entropy stable discontinuous
Galerkin methods for …},
  author={Wu, Xinhui and Trask,
Nathaniel and Chan, Jesse},
  journal={Numerical Methods for PDEs},
  pages={e23129},
  publisher={Wiley Online Library}
}
```



Bibliographies and Bibtex

- Include references in a .bib file in the same directory.
- Add \bibliography{reference file name} and cite away
- Can customize the style of bibliography entries (MLA, Chicago, APA, etc)
- Process: compile LaTeX, then compile BibTex, then recompile LaTeX again.

```
I can reference a paper via \cite{wu2024entropy}.

\bibliographystyle{plain}
\bibliography{references.bib}

\end{document}
```

I can reference a paper via [1].

References

[1] Xinhui Wu, Nathaniel Trask, and Jesse Chan. Entropy stable discontinuous galerkin methods for the shallow water equations with subcell positivity preservation. *Numerical Methods for Partial Differential Equations*, page e23129.

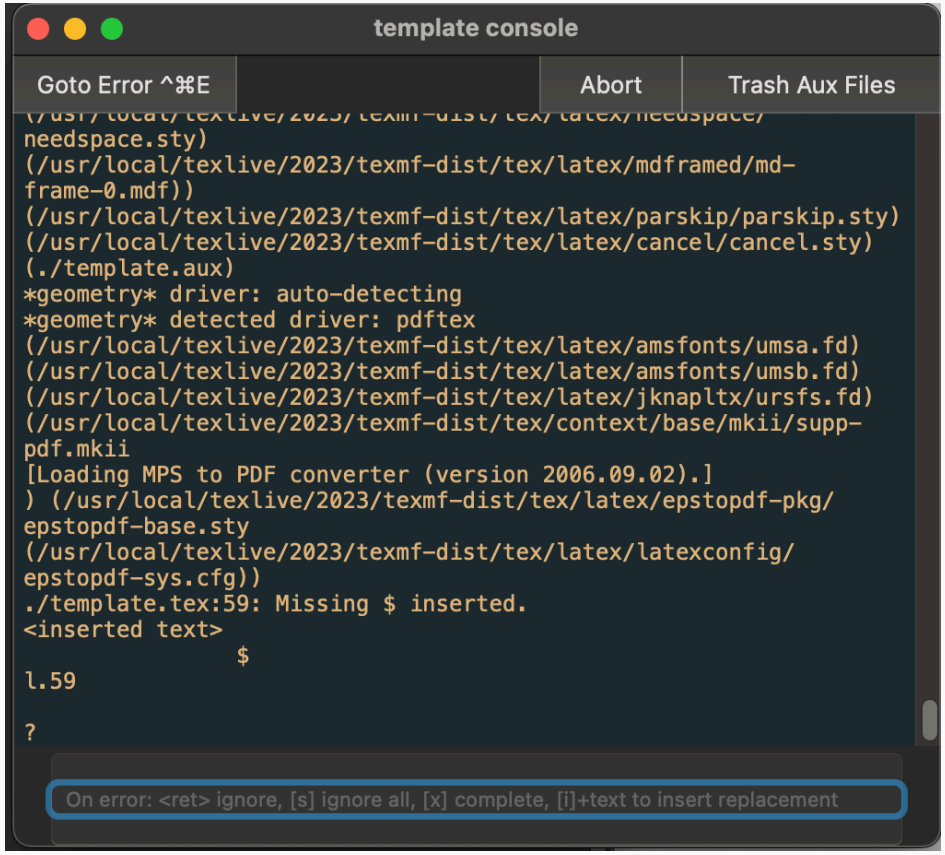
More about Bibtex

- Bibtex will create bibliography entries for only what is cited in the paper
- You can have a .bib file with hundreds of entries but a bibliography with just those you want/need!

- Bibtex will create a .bbl file that is essentially the bibliography
  - This file can be edited directly if desired

Error tracing

- Latex produces a console at compilation that reads through the document
- If there is error, the console stops and tells you (around) where the error is and what the issue is



Exercise

- Download the latex template document from Canvas
- Give it a title, author, and write the following

This is my first latex document. For $x \in \mathbb{R}$ and $y \in \mathbb{R}$ we have that

$$x + y \in \mathbb{R}.$$

I can make a table of values

1	2	3
4	5	6
7	8	9

Overleaf

- Nowadays, most people use LaTeX via Overleaf.
- Overleaf has great resources for learning LaTeX.
- Note that Rice provides Overleaf professional! See https://www.overleaf.com/edu/rice
- I still prefer using LaTeX locally; it's faster, more responsive, and works offline.

Numerical computing concepts

- IEEE 754: standardized behavior (1985, 2008) to address portability issues across computer architectures.
- Several concepts are shared between C, C++, and Julia (and any programming language suitable for numerical computing)

Data types in scientific computing

- For numerical computing, it can help to know the difference between different data types (e.g., integer, float, double, long int, etc) and what distinguishes them from each other
  - Integers and decimal (float, double) behave very differently!
- To start, numbers are represented in binary in a computer.
  - If we represent an integer in binary, 0110 corresponds to

$$5 = 0 * (2^0) + 1 * (2^1) + 1 * (2^2) + 0 * (2^3)$$

Binary representations of integers

- 5 = 0110 = 0 * (2^0) + 1 * (2^1) + 1 * (2^2) + 0 * (2^3)
  - Binary 0/1 digits are coefficients in a binary expansion
  - We use 4 “bits” here; 64-bit integers are most common, but 8-bit, 16-bit, and 32-bit integers are all used.
- What does this expansion imply about the numbers that can be represented in binary?

* They are finite! Only a finite amount of numbers can be expressed as 64 bits

* IEEE 64 bit representation has 8-bits for an exponent to allow for representation of more numbers

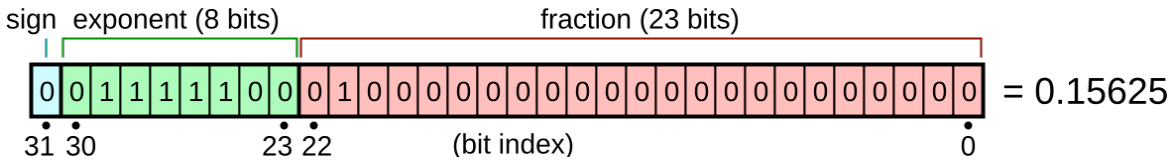
Unsigned vs signed integers

- You can have both “unsigned integer” or “signed integer” types (e.g., “unsigned int” vs “int”)
  - Both have the same bit representation, but can yield different results because they are *interpreted* differently
  - For signed integers: one bit determines the sign.
- Advantages of unsigned integers: represent larger values, provides additional information about what the variable is for.

What about decimal numbers?

- Easy to represent integers as binary; what about decimal?
  - Fixed point representation: represent digits before and after a decimal point (fixed number of representable digits)
  - Floating point representation: represent a number as a coefficient and exponent. Higher *relative* accuracy.

$$12.345 = \underbrace{12345}_{\text{significand}} \times \underbrace{10^{\overbrace{-3}^{\text{exponent}}}}_{\text{base}}$$



https://en.wikipedia.org/wiki/Floating-point_arithmetic

Consequences of floating point representations

- Rounding: the sum of two floating point numbers may not be representable as another floating point number, so the result is *rounded* to the nearest representable number.
- Checking equality of floating point numbers is unreliable!

```
julia> sin(1.0 * pi) == 0.0
false
```

- Catastrophic cancellation: subtracting large numbers from each other can result in large relative errors due to rounding.
- Overflow: results in either NaNs/Infs.

Arrays

- Arrays include vectors, matrices, tensors, …
- Typically represents a list of numbers of one type (e.g., an array typically doesn't have both 32-bit and 64-bit numbers)
- What's the difference between arrays and other containers like Dictionaries in Python? **Speed.**
  - Arrays are very *low-overhead* containers, which make them important for efficient programs.