

# Internship Report - Computational Materials

Hasitha S

B.E Materials Science and Engineering  
College of Engineering, Guindy

June 11, 2025

## 1 Introduction

This report summarizes what I learned during my internship in computational materials science, including Linux system commands, Python, SageMath, and LaTeX documentation using *Overleaf*.



---

## 2 Linux System Commands

I explored a wide range of Linux terminal commands and features relevant to computational tasks. These are grouped into several categories:

### 2.1 Basic File and Directory Commands

- `pwd` – Print the current working directory
- `cd` – Change the current directory
- `ls` – List the files and directories in the current location
- `mkdir` – Create a new directory
- `mv` - Move files or rename
- `rm` – Remove files or directories
- `ls` – lists folder contents
- `ls -l` – shows extended details (permissions, size, etc.)
- `ls -ld` – shows info about the directory itself, not its contents
- `ls -li` – includes inode numbers
- `ls -lid <directory>` – inode info for a specific directory
- `less` - to read files and quit
- `cat` - concatenate text to page
- `wc` - wordcount
- Handling of file and directory permissions using `chmod`
- Other tools like `cal`, `date`

## 2.2 File Description and System Info

- `file` – Determine the type of a file
- `stat` – Display detailed information about a file, including size and permissions
- `du` – Estimate file space usage (disk usage)
- `free` – Show available and used memory on the system
- `groups` – Show the groups a user belongs to
- `man` – View the manual page for a command
- `help` – Display help for shell built-in commands
- `which` – Show the full path of a command
- `whatis` – Give a one-line description of a command
- `apropos` – Search the manual page names and descriptions for a keyword
- `/proc` – Contains information about running processes and system details (like CPU and memory)
- `/sys` – Shows details about hardware devices and kernel settings

## 2.3 Shell Variables

- `echo` – Prints text or variable values to the terminal; use double quotes for spacing and special characters
- `$USER`, `$HOME`, `$HOSTNAME` – Environment variables that show your username, home directory, and machine name
- `\` – Escape character used to prevent interpretation (e.g., to avoid alias effects)
- `printenv`, `print` – Display a list of all environment variables
- `set` – Shows all shell variables and functions
- `$0` – Shows the name of the shell or script being executed
- `$$` – Displays the process ID (PID) of the current shell
- `ps` – Shows currently running processes
- `ps --forest` – Displays processes in a tree format to show parent-child relationships
- `ps -ef` – Lists every process on the system in full detail
- `ps -f`, `ps -e` – Show detailed info about running processes

## 2.4 Combining Commands and File Operations

- `;` – Used to run multiple commands in a row
- `()` – Creates a subshell where commands are run in a separate environment
- `$BASH_SUBSHELL` – Shows how many subshells deep the current shell is
- `&&` – Runs the second command only if the first one succeeds
- `||` – Runs the second command only if the first one fails
- **File Descriptors:**
  - `0` – `stdin`, standard input
  - `1` – `stdout`, standard output
  - `2` – `stderr`, standard error

- `command > filename` – Redirects output to a file (overwrites if file exists)
- `command >> filename` – Appends output to a file
- `command 2> filename` – Redirects error messages to a file
- `command > filename 2>&1` – Redirects both output and errors to the same file
- `< filename` – Takes input from a file instead of typing it manually
- `cat > filename` – Lets you type content into a file (use **Ctrl+D** to stop)
- `|` – Pipe operator; sends the output of one command as input to another
- `/dev/null` – Special file that discards everything sent to it (useful to ignore output or errors)
- `command | tee filename` – Displays the output and also writes it to one or more files.
- `diff file1 file2` – Compares two files and shows differences

## 2.5 Linux Process Management

- `coproc` – Launches a process in the background.
- `sleep 30 &` – Runs a process in the background using `&`.
- `fg` – Brings a background job to the foreground.
- `ctrl + c` – Terminates the current foreground process.
- `kill <PID>` – Ends a background process using its process ID.
- `jobs` – Lists current background jobs in the shell.
- `help` – Displays help for shell built-in commands (useful instead of `man`).
- `echo $-` – Shows current shell options/features.
- `bash -c "echo $-"` – Creates a new shell and displays its options.

## 2.6 Pattern Matching with grep and egrep

- `grep <pattern> <filename>` – Searches for a pattern in a file using basic regular expressions.
- `egrep` – Uses extended regular expressions (ERE) for more complex patterns.

### Common Pattern Matching Symbols:

- `.` : Matches any single character.
- `*` : Matches zero or more of the preceding character.
- `[ ]` : Matches any one character in the set or range.
- `[^]` : Matches any character not in the set.
- `^` : Matches beginning
- `$` : Matches ending
- `\` : Escapes special characters.
- `{n,m}` : Matches between `n` and `m` repetitions of the preceding character.
- `()` : Groups expressions for repeated matches.
- `+` : Matches one or more of the preceding pattern.
- `?` : Matches zero or one occurrence of the preceding pattern.
- `|` : Acts as OR between patterns.
- `\b` : Matches a word boundary.

## 2.7 Command Line Editors

- ED

- `ed filename` : Opens the file
- `P` : Enable prompt.
- `2` : Refers to second line, `.` current line, `$` last line.
- `d` : Delete line.
- `f` : Display file name.
- `%` : Refers to all lines.
- `+` / `-` : Next or previous line.
- `;` : Range from current line to end.
- `/RE/` : Search using regular expression.
- `,p` : Print entire content.
- `a` : Append text (exit with single `.` on a new line).
- `s/old/new/` : Substitute text.
- `2,3j` : Join lines 2 and 3.
- `m1` : Move current line to first position.
- `%s/(.*)/PREFIX\1/` : Add prefix to all lines.
- `!command` : Execute shell command.
- `e filename` : Edit another file.
- `r filename` : Read file contents into buffer.
- `r !command` : Read command output into buffer.
- `w filename` : Write buffer to file.
- `q` : Quit.

- VI

- **Modes:**
    - \* **Command mode**
    - \* **Insert mode**
      - `i` : Insert at cursor.
      - `o` : Insert new line below.
      - `a` : Append after cursor.
    - \* **Ex mode**
      - `:w` : Write file.
      - `:q` : Quit.
      - `:wq` or `:x` : Save and quit.
      - `:q!` : Quit without saving.
      - `h j k l` : Move cursor (left, down, up, right).
      - `:1s/old/new/` : Substitute in first line.
      - `r` : Replace single character.
      - `R` : Replace continuously until `Esc`
-

## 3 Python

I explored various practical aspects of Python. I have organized my work into Jupyter notebooks, each covering a specific topic. These notebooks are hosted on my GitHub repository and can be accessed through the links provided below for each section.

- **Basics of Python**  
<https://github.com/Hasitha1405/computational-materials-internship/blob/main/Python/1.Basics.ipynb>
  - **2D Visualization of Data**  
<https://github.com/Hasitha1405/computational-materials-internship/blob/main/Python/2.2D%20visualization%20of%20data.ipynb>
  - **Arrays from numpy, fitting polynomials and user defined functions to data**  
<https://github.com/Hasitha1405/computational-materials-internship/blob/main/Python/3.Arrays%20from%20numpy%2C%20fitting%20polynomials%20and%20user%20defined%20functions%20to%20data.ipynb>
  - **Binning of large arrays, Parametric plots**  
<https://github.com/Hasitha1405/computational-materials-internship/blob/main/Python/4.Binning%20of%20large%20arrays%2C%20Parametric%20plots.ipynb>
  - **Linear Algebra, Array representation**  
<https://github.com/Hasitha1405/computational-materials-internship/blob/main/Python/5.Linear%20Algebra%2C%20Array%20representation.ipynb>
- 

## 4 SageMath

In addition to Python, I also worked with SageMath—an open-source mathematics software system, especially useful for symbolic computation, algebra, calculus, number theory, and data visualization.

I used SageMath primarily through Jupyter notebooks to perform mathematical operations and visualize results. Relevant SageMath notebooks are also linked below for reference.

- **Introduction to symbolic computation using sage expressions**  
<https://github.com/Hasitha1405/computational-materials-internship/blob/main/SageMath/1.Introduction%20to%20symbolic%20computation%20using%20sage%20expressions.ipynb>
- **Manipulation of symbolic expressions**  
<https://github.com/Hasitha1405/computational-materials-internship/blob/main/SageMath/2.Manipulation%20of%20symbolic%20expressions.ipynb>
- **Symbolic differentiation**  
<https://github.com/Hasitha1405/computational-materials-internship/blob/main/SageMath/3.Symbolic%20differentiation.ipynb>
- **Symbolic Integration**  
<https://github.com/Hasitha1405/computational-materials-internship/blob/main/SageMath/4.Symbolic%20Integration.ipynb>
- **LaTeX symbols**  
<https://github.com/Hasitha1405/computational-materials-internship/blob/main/SageMath/5.LaTeX%20symbols.ipynb>
- **Integers**  
<https://github.com/Hasitha1405/computational-materials-internship/blob/main/SageMath/6.Integer%20s.ipynb>
- **Real Number**  
<https://github.com/Hasitha1405/computational-materials-internship/blob/main/SageMath/7.Real%20Numbers.ipynb>

- **Rational Number**

<https://github.com/Hasitha1405/computational-materials-internship/blob/main/SageMath/8.Rational%20Numbers.ipynb>

- **Complex numbers**

<https://github.com/Hasitha1405/computational-materials-internship/blob/main/SageMath/9.Complex%20numbers.ipynb>

---

## 5 LaTeX Documentation

I also learned to use  $\text{\LaTeX}$ , a powerful typesetting system commonly used in academia for writing technical and scientific documents.

I used  $\text{\LaTeX}$  to compile and format this report and became familiar with its key features such as:

- Creating sections, subsections, and structured content
  - Formatting text (bold, italics, underline)
  - Adding bullet points and numbered lists
  - Writing and aligning mathematical expressions
  - Inserting images and controlling their size and position
  - Adding hyperlinks and URLs using the `hyperref` package
  - Referencing sections, figures, and equations
  - Altering line spacing and adjusting margins using packages like `geometry`
  - Customizing page layout, borders, and alignment
  - Handling special characters and escape sequences
- 

## 6 Conclusion

I gained practical exposure to a wide range of tools essential for computational materials science.

I explored the Linux command line for efficient system navigation, created and executed Bash scripts, worked with Python for data visualization and analysis, and used SageMath for symbolic and numerical computation. In addition, I learned to document my work professionally using  $\text{\LaTeX}$ .

This experience has strengthened my technical foundation and improved my ability to approach problems systematically using computational tools.