

# Sharpening your saw before cutting down the tree

## – Personal development environment (PDE)

Le Chen  
Auburn University



Graduate Student Seminar

Department of Mathematics & Statistics  
Auburn University

Feb. 15th 2023





# What is Vi/Vim?

- ▶ Text editor designed for speed and efficiency
- ▶ Built-in support for many programming languages and file types
- ▶ Highly customizable and extensible through plugins and configurations
- ▶ Available on most platforms (including Linux, macOS, and Windows)

# What is Vi/Vim?

- ▶ Text editor designed for speed and efficiency
- ▶ Built-in support for many programming languages and file types
- ▶ Highly customizable and extensible through plugins and configurations
- ▶ Available on most platforms (including Linux, macOS, and Windows)

# What is Vi/Vim?

- ▶ Text editor designed for speed and efficiency
- ▶ Built-in support for many programming languages and file types
- ▶ Highly customizable and extensible through plugins and configurations
- ▶ Available on most platforms (including Linux, macOS, and Windows)

# What is Vi/Vim?

- ▶ Text editor designed for speed and efficiency
- ▶ Built-in support for many programming languages and file types
- ▶ Highly customizable and extensible through plugins and configurations
- ▶ Available on most platforms (including Linux, macOS, and Windows)

# What is NeoVim?

- ▶ Fork of Vim with many additional features and improvements
- ▶ Decouples Vim from its legacy architecture to enable easier maintenance and extension
- ▶ Built-in support for features such as asynchronous plugins, better integration with terminals, and remote plugins

# What is NeoVim?

- ▶ Fork of Vim with many additional features and improvements
- ▶ Decouples Vim from its legacy architecture to enable easier maintenance and extension
- ▶ Built-in support for features such as asynchronous plugins, better integration with terminals, and remote plugins

# What is NeoVim?

- ▶ Fork of Vim with many additional features and improvements
- ▶ Decouples Vim from its legacy architecture to enable easier maintenance and extension
- ▶ Built-in support for features such as asynchronous plugins, better integration with terminals, and remote plugins

# History of Vim and NeoVim

- ▶ 1976 - vi is created by Bill Joy at UC Berkeley
- ▶ 1991 - Vim (Vi Improved) is released by Bram Moolenaar
- ▶ 2014 - NeoVim is forked from Vim by Thiago de Arruda and other developers
- ▶ 2015 - NeoVim 0.1.0 is released, featuring many new features and improvements over Vim
- ▶ 2023 - Latest release of Vim is version 8.2, and latest release of NeoVim is version 0.9.0

# History of Vim and NeoVim

- ▶ 1976 - vi is created by Bill Joy at UC Berkeley
- ▶ 1991 - Vim (Vi Improved) is released by Bram Moolenaar
- ▶ 2014 - NeoVim is forked from Vim by Thiago de Arruda and other developers
- ▶ 2015 - NeoVim 0.1.0 is released, featuring many new features and improvements over Vim
- ▶ 2023 - Latest release of Vim is version 8.2, and latest release of NeoVim is version 0.9.0

# History of Vim and NeoVim

- ▶ 1976 - vi is created by Bill Joy at UC Berkeley
- ▶ 1991 - Vim (Vi Improved) is released by Bram Moolenaar
- ▶ 2014 - NeoVim is forked from Vim by Thiago de Arruda and other developers
- ▶ 2015 - NeoVim 0.1.0 is released, featuring many new features and improvements over Vim
- ▶ 2023 - Latest release of Vim is version 8.2, and latest release of NeoVim is version 0.9.0

# History of Vim and NeoVim

- ▶ 1976 - vi is created by Bill Joy at UC Berkeley
- ▶ 1991 - Vim (Vi Improved) is released by Bram Moolenaar
- ▶ 2014 - NeoVim is forked from Vim by Thiago de Arruda and other developers
- ▶ 2015 - NeoVim 0.1.0 is released, featuring many new features and improvements over Vim
- ▶ 2023 - Latest release of Vim is version 8.2, and latest release of NeoVim is version 0.9.0

# History of Vim and NeoVim

- ▶ 1976 - vi is created by Bill Joy at UC Berkeley
- ▶ 1991 - Vim (Vi Improved) is released by Bram Moolenaar
- ▶ 2014 - NeoVim is forked from Vim by Thiago de Arruda and other developers
- ▶ 2015 - NeoVim 0.1.0 is released, featuring many new features and improvements over Vim
- ▶ 2023 - Latest release of Vim is version 8.2, and latest release of NeoVim is version 0.9.0

## Advantages of NeoVim

- ▶ Asynchronous plugins - Plugins can run in the background without blocking the editor
- ▶ Better terminal integration - Built-in terminal emulator for running shell commands
- ▶ Simplified configuration - NeoVim uses a Lua-based configuration system that is easier to read and write than Vim's configuration
- ▶ Improved performance - NeoVim's architecture allows it to be faster and more stable than Vim, especially with large files or complex operations

## Advantages of NeoVim

- ▶ Asynchronous plugins - Plugins can run in the background without blocking the editor
- ▶ Better terminal integration - Built-in terminal emulator for running shell commands
- ▶ Simplified configuration - NeoVim uses a Lua-based configuration system that is easier to read and write than Vim's configuration
- ▶ Improved performance - NeoVim's architecture allows it to be faster and more stable than Vim, especially with large files or complex operations

## Advantages of NeoVim

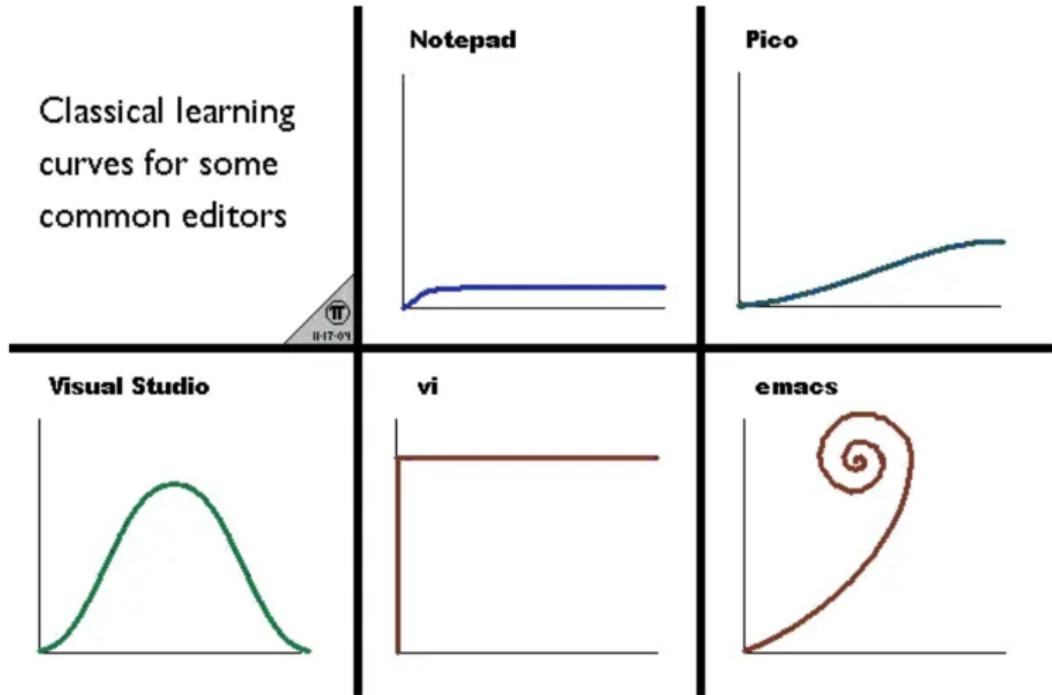
- ▶ Asynchronous plugins - Plugins can run in the background without blocking the editor
- ▶ Better terminal integration - Built-in terminal emulator for running shell commands
- ▶ Simplified configuration - NeoVim uses a Lua-based configuration system that is easier to read and write than Vim's configuration
- ▶ Improved performance - NeoVim's architecture allows it to be faster and more stable than Vim, especially with large files or complex operations

## Advantages of NeoVim

- ▶ Asynchronous plugins - Plugins can run in the background without blocking the editor
- ▶ Better terminal integration - Built-in terminal emulator for running shell commands
- ▶ Simplified configuration - NeoVim uses a Lua-based configuration system that is easier to read and write than Vim's configuration
- ▶ Improved performance - NeoVim's architecture allows it to be faster and more stable than Vim, especially with large files or complex operations

# Editor holy war

Classical learning  
curves for some  
common editors



<sup>1</sup>Image from <https://prajwollamichhane11.medium.com/vim-vs-emacs-the-editor-war-b63ecb12ea92>

## VIM

usable in just about  
any environment.

does one thing, well.



## EMACS

flexible, customizable, and  
packed with every feature  
known to man.



<sup>2</sup>Image from <https://prajwollamichhane11.medium.com/vim-vs-emacs-the-editor-war-b63ecb12ea92>

# Basic Vim Commands

- ▶ Three modes: normal, insert, and visual
- ▶ Navigation: h, j, k, l, w, b, e, 0, \$, gg, G, f, F, t, T
- ▶ Editing: i, a, o, I, A, O, r, d, D, y, Y, p, P, u, x
- ▶ Saving and quitting: :w, :wq, :q, :q!, :w!, :wq!
  
- ▶ Text objects in VimTeX:
  - c** (change), **v** (visual select), **t** (toggle), **d** (delete)
  - i** (inside), **a** (around), **s** (surround)
  - e (environment), \$ (inline math), s (sentence), p (paragraph), w (word), W (word)
  
- ▶ Macros: q, @
- ▶ Snippets: tab
  
- ▶ Searching and replacing: /, ?, n, N, :s, :s/, g, gc

# Basic Vim Commands

- ▶ Three modes: normal, insert, and visual
- ▶ Navigation: h, j, k, l, w, b, e, 0, \$, gg, G, f, F, t, T
- ▶ Editing: i, a, o, I, A, O, r, d, D, y, Y, p, P, u, x
- ▶ Saving and quitting: :w, :wq, :q, :q!, :w!, :wq!
  
- ▶ Text objects in VimTeX:
  - c** (change), **v** (visual select), **t** (toggle), **d** (delete)
  - i** (inside), **a** (around), **s** (surround)
  - e (environment), \$ (inline math), s (sentence), p (paragraph), w (word), W (word)
  
- ▶ Macros: q, @
- ▶ Snippets: tab
  
- ▶ Searching and replacing: /, ?, n, N, :s, :s/, g, gc

# Basic Vim Commands

- ▶ Three modes: normal, insert, and visual
- ▶ Navigation: h, j, k, l, w, b, e, 0, \$, gg, G, f, F, t, T
- ▶ Editing: i, a, o, I, A, O, r, d, D, y, Y, p, P, u, x
- ▶ Saving and quitting: :w, :wq, :q, :q!, :w!, :wq!
  
- ▶ Text objects in VimTeX:
  - c** (change), **v** (visual select), **t** (toggle), **d** (delete)
  - i** (inside), **a** (around), **s** (surround)
  - e (environment), \$ (inline math), s (sentence), p (paragraph), w (word), W (word)
  
- ▶ Macros: q, @
- ▶ Snippets: tab
  
- ▶ Searching and replacing: /, ?, n, N, :s, :s/, g, gc

# Basic Vim Commands

- ▶ Three modes: normal, insert, and visual
- ▶ Navigation: h, j, k, l, w, b, e, 0, \$, gg, G, f, F, t, T
- ▶ Editing: i, a, o, I, A, O, r, d, D, y, Y, p, P, u, x
- ▶ Saving and quitting: :w, :wq, :q, :q!, :w!, :wq!
  
- ▶ Text objects in VimTeX:
  - c** (change), **v** (visual select), **t** (toggle), **d** (delete)
  - i** (inside), **a** (around), **s** (surround)
  - e (environment), \$ (inline math), s (sentence), p (paragraph), w (word), W (word)
  
- ▶ Macros: q, @
- ▶ Snippets: tab
  
- ▶ Searching and replacing: /, ?, n, N, :s, :s/, g, gc

# Basic Vim Commands

- ▶ Three modes: normal, insert, and visual
- ▶ Navigation: h, j, k, l, w, b, e, 0, \$, gg, G, f, F, t, T
- ▶ Editing: i, a, o, I, A, O, r, d, D, y, Y, p, P, u, x
- ▶ Saving and quitting: :w, :wq, :q, :q!, :w!, :wq!
  
- ▶ Text objects in VimTeX:
  - c** (change), **v** (visual select), **t** (toggle), **d** (delete)
  - i** (inside), **a** (around), **s** (surround)
  - e (environment), \$ (inline math), s (sentence), p (paragraph), w (word), W (word)
  
- ▶ Macros: q, @
- ▶ Snippets: tab
  
- ▶ Searching and replacing: /, ?, n, N, :s, :s/, g, gc

# Basic Vim Commands

- ▶ Three modes: normal, insert, and visual
- ▶ Navigation: h, j, k, l, w, b, e, 0, \$, gg, G, f, F, t, T
- ▶ Editing: i, a, o, I, A, O, r, d, D, y, Y, p, P, u, x
- ▶ Saving and quitting: :w, :wq, :q, :q!, :w!, :wq!
  
- ▶ Text objects in VimTeX:
  - c** (change), **v** (visual select), **t** (toggle), **d** (delete)
  - i** (inside), **a** (around), **s** (surround)
  - e (environment), \$ (inline math), s (sentence), p (paragraph), w (word), W (word)
  
- ▶ Macros: q, @
- ▶ Snippets: tab
  
- ▶ Searching and replacing: /, ?, n, N, :s, :s/, g, gc

# Basic Vim Commands

- ▶ Three modes: normal, insert, and visual
- ▶ Navigation: h, j, k, l, w, b, e, 0, \$, gg, G, f, F, t, T
- ▶ Editing: i, a, o, I, A, O, r, d, D, y, Y, p, P, u, x
- ▶ Saving and quitting: :w, :wq, :q, :q!, :w!, :wq!
  
- ▶ Text objects in VimTeX:
  - c** (change), **v** (visual select), **t** (toggle), **d** (delete)
  - i** (inside), **a** (around), **s** (surround)
  - e (environment), \$ (inline math), s (sentence), p (paragraph), w (word), W (word)
  
- ▶ Macros: q, @
- ▶ Snippets: tab
  
- ▶ Searching and replacing: /, ?, n, N, :s, :s/, g, gc

# Basic Vim Commands

- ▶ Three modes: normal, insert, and visual
- ▶ Navigation: h, j, k, l, w, b, e, 0, \$, gg, G, f, F, t, T
- ▶ Editing: i, a, o, I, A, O, r, d, D, y, Y, p, P, u, x
- ▶ Saving and quitting: :w, :wq, :q, :q!, :w!, :wq!
  
- ▶ Text objects in VimTeX:
  - c** (change), **v** (visual select), **t** (toggle), **d** (delete)
  - i** (inside), **a** (around), **s** (surround)
  - e (environment), \$ (inline math), s (sentence), p (paragraph), w (word), W (word)
  
- ▶ Macros: q, @
- ▶ Snippets: tab
  
- ▶ Searching and replacing: /, ?, n, N, :s, :s/, g, gc

# Basic Vim Commands

- ▶ Three modes: normal, insert, and visual
- ▶ Navigation: h, j, k, l, w, b, e, 0, \$, gg, G, f, F, t, T
- ▶ Editing: i, a, o, I, A, O, r, d, D, y, Y, p, P, u, x
- ▶ Saving and quitting: :w, :wq, :q, :q!, :w!, :wq!
  
- ▶ Text objects in VimTeX:
  - c** (change), **v** (visual select), **t** (toggle), **d** (delete)
  - i** (inside), **a** (around), **s** (surround)
  - e (environment), \$ (inline math), s (sentence), p (paragraph), w (word), W (word)
  
- ▶ Macros: q, @
- ▶ Snippets: tab
  
- ▶ Searching and replacing: /, ?, n, N, :s, :s/, g, gc

# Basic Vim Commands

- ▶ Three modes: normal, insert, and visual
- ▶ Navigation: h, j, k, l, w, b, e, 0, \$, gg, G, f, F, t, T
- ▶ Editing: i, a, o, I, A, O, r, d, D, y, Y, p, P, u, x
- ▶ Saving and quitting: :w, :wq, :q, :q!, :w!, :wq!
  
- ▶ Text objects in VimTeX:
  - c** (change), **v** (visual select), **t** (toggle), **d** (delete)
  - i** (inside), **a** (around), **s** (surround)
  - e (environment), \$ (inline math), s (sentence), p (paragraph), w (word), W (word)
  
- ▶ Macros: q, @
- ▶ Snippets: tab
  
- ▶ Searching and replacing: /, ?, n, N, :s, :s/, g, gc

# Basic Vim Commands

- ▶ Three modes: normal, insert, and visual
- ▶ Navigation: h, j, k, l, w, b, e, 0, \$, gg, G, f, F, t, T
- ▶ Editing: i, a, o, I, A, O, r, d, D, y, Y, p, P, u, x
- ▶ Saving and quitting: :w, :wq, :q, :q!, :w!, :wq!
  
- ▶ Text objects in VimTeX:
  - c** (change), **v** (visual select), **t** (toggle), **d** (delete)
  - i** (inside), **a** (around), **s** (surround)
  - e (environment), \$ (inline math), s (sentence), p (paragraph), w (word), W (word)
  
- ▶ Macros: q, @
- ▶ Snippets: tab
  
- ▶ Searching and replacing: /, ?, n, N, :s, :s/, g, gc

# Basic Vim Commands

- ▶ Three modes: normal, insert, and visual
- ▶ Navigation: h, j, k, l, w, b, e, 0, \$, gg, G, f, F, t, T
- ▶ Editing: i, a, o, I, A, O, r, d, D, y, Y, p, P, u, x
- ▶ Saving and quitting: :w, :wq, :q, :q!, :w!, :wq!
  
- ▶ Text objects in VimTeX:
  - c** (change), **v** (visual select), **t** (toggle), **d** (delete)
  - i** (inside), **a** (around), **s** (surround)
  - e (environment), \$ (inline math), s (sentence), p (paragraph), w (word), W (word)
  
- ▶ Macros: q, @
- ▶ Snippets: tab
  
- ▶ Searching and replacing: /, ?, n, N, :s, :s/, g, gc

# Basic Vim Commands

- ▶ Three modes: normal, insert, and visual
- ▶ Navigation: h, j, k, l, w, b, e, 0, \$, gg, G, f, F, t, T
- ▶ Editing: i, a, o, I, A, O, r, d, D, y, Y, p, P, u, x
- ▶ Saving and quitting: :w, :wq, :q, :q!, :w!, :wq!
  
- ▶ Text objects in VimTeX:
  - c** (change), **v** (visual select), **t** (toggle), **d** (delete)
  - i** (inside), **a** (around), **s** (surround)
  - e (environment), \$ (inline math), s (sentence), p (paragraph), w (word), W (word)
  
- ▶ Macros: q, @
- ▶ Snippets: tab
  
- ▶ Searching and replacing: /, ?, n, N, :s, :s/, g, gc

# Basic Vim Commands

- ▶ Three modes: normal, insert, and visual
- ▶ Navigation: h, j, k, l, w, b, e, 0, \$, gg, G, f, F, t, T
- ▶ Editing: i, a, o, I, A, O, r, d, D, y, Y, p, P, u, x
- ▶ Saving and quitting: :w, :wq, :q, :q!, :w!, :wq!
  
- ▶ Text objects in VimTeX:
  - c** (change), **v** (visual select), **t** (toggle), **d** (delete)
  - i** (inside), **a** (around), **s** (surround)
  - e (environment), \$ (inline math), s (sentence), p (paragraph), w (word), W (word)
  
- ▶ Macros: q, @
- ▶ Snippets: tab
  
- ▶ Searching and replacing: /, ?, n, N, :s, :s/, g, gc

## Interview with a VIM Enthusiast

<https://www.youtube.com/watch?v=9n1dtmzqnCU&t=141s>

## Introduction to vi/vim/nvim

### Using git/github to track your progress

What is Git?

Basic Git Commands

What is GitHub?

Getting Started with Git and GitHub

Conclusion

Chatgpt

Gallery generated by Midjourney

Acknowledgement

## What is Git?

- ▶ Distributed version control system
- ▶ Tracks changes to code over time
- ▶ Enables collaboration and easy rollback

## What is Git?

- ▶ Distributed version control system
- ▶ Tracks changes to code over time
- ▶ Enables collaboration and easy rollback

## What is Git?

- ▶ Distributed version control system
- ▶ Tracks changes to code over time
- ▶ Enables collaboration and easy rollback

# Basic Git Commands

- ▶ git init
- ▶ git add
- ▶ git commit
- ▶ git push
- ▶ git pull
- ▶ git clone

# Basic Git Commands

- ▶ git init
- ▶ git add
- ▶ git commit
- ▶ git push
- ▶ git pull
- ▶ git clone

# Basic Git Commands

- ▶ git init
- ▶ git add
- ▶ git commit
- ▶ git push
- ▶ git pull
- ▶ git clone

# Basic Git Commands

- ▶ git init
- ▶ git add
- ▶ git commit
- ▶ git push
- ▶ git pull
- ▶ git clone

# Basic Git Commands

- ▶ git init
- ▶ git add
- ▶ git commit
- ▶ git push
- ▶ git pull
- ▶ git clone

# Basic Git Commands

- ▶ git init
- ▶ git add
- ▶ git commit
- ▶ git push
- ▶ git pull
- ▶ git clone

# What is GitHub?

- ▶ Web-based platform for Git repositories
- ▶ Provides collaboration features (e.g., pull requests)
- ▶ Includes project management tools (e.g., issues, milestones)

# What is GitHub?

- ▶ Web-based platform for Git repositories
- ▶ Provides collaboration features (e.g., pull requests)
- ▶ Includes project management tools (e.g., issues, milestones)

# What is GitHub?

- ▶ Web-based platform for Git repositories
- ▶ Provides collaboration features (e.g., pull requests)
- ▶ Includes project management tools (e.g., issues, milestones)

# Getting Started with Git and GitHub

- ▶ Install Git on your computer
- ▶ Create a GitHub account
- ▶ Create a new repository on GitHub
- ▶ Clone the repository to your local machine
- ▶ Add files and commit changes
- ▶ Push changes to the remote repository

# Getting Started with Git and GitHub

- ▶ Install Git on your computer
- ▶ Create a GitHub account
- ▶ Create a new repository on GitHub
- ▶ Clone the repository to your local machine
- ▶ Add files and commit changes
- ▶ Push changes to the remote repository

# Getting Started with Git and GitHub

- ▶ Install Git on your computer
- ▶ Create a GitHub account
- ▶ Create a new repository on GitHub
- ▶ Clone the repository to your local machine
- ▶ Add files and commit changes
- ▶ Push changes to the remote repository

# Getting Started with Git and GitHub

- ▶ Install Git on your computer
- ▶ Create a GitHub account
- ▶ Create a new repository on GitHub
- ▶ Clone the repository to your local machine
- ▶ Add files and commit changes
- ▶ Push changes to the remote repository

# Getting Started with Git and GitHub

- ▶ Install Git on your computer
- ▶ Create a GitHub account
- ▶ Create a new repository on GitHub
- ▶ Clone the repository to your local machine
- ▶ Add files and commit changes
- ▶ Push changes to the remote repository

# Getting Started with Git and GitHub

- ▶ Install Git on your computer
- ▶ Create a GitHub account
- ▶ Create a new repository on GitHub
- ▶ Clone the repository to your local machine
- ▶ Add files and commit changes
- ▶ Push changes to the remote repository

# Conclusion

- ▶ Git and GitHub are powerful tools for managing code and collaborating with others
- ▶ By using Git and GitHub, you can easily keep track of changes to your code and work more efficiently with others
- ▶ Practice using Git and GitHub regularly to become proficient in using these tools

# Conclusion

- ▶ Git and GitHub are powerful tools for managing code and collaborating with others
- ▶ By using Git and GitHub, you can easily keep track of changes to your code and work more efficiently with others
- ▶ Practice using Git and GitHub regularly to become proficient in using these tools

# Conclusion

- ▶ Git and GitHub are powerful tools for managing code and collaborating with others
- ▶ By using Git and GitHub, you can easily keep track of changes to your code and work more efficiently with others
- ▶ Practice using Git and GitHub regularly to become proficient in using these tools



What is it good at?

1. Write codes (all languages) – Work automation.
2. Write essay, poem, etc. – for fun.
3. Revise general purposed text.

What is it not good at?

1. Mathematically intensive tasks
2. Literature review (be really careful)

What is it good at?

1. Write codes (all languages) – Work automation.
2. Write essay, poem, etc. – for fun.
3. Revise general purposed text.

What is it not good at?

1. Mathematically intensive tasks
2. Literature review (be really careful)

What is it good at?

1. Write codes (all languages) – Work automation.
2. Write essay, poem, etc. – for fun.
3. Revise general purposed text.

What is it not good at?

1. Mathematically intensive tasks
2. Literature review (be really careful)

What is it good at?

1. Write codes (all languages) – Work automation.
2. Write essay, poem, etc. – for fun.
3. Revise general purposed text.

What is it not good at?

1. Mathematically intensive tasks
2. Literature review (be really careful)

What is it good at?

1. Write codes (all languages) – Work automation.
2. Write essay, poem, etc. – for fun.
3. Revise general purposed text.

What is it not good at?

1. Mathematically intensive tasks
2. Literature review (be really careful)

## Introduction to vi/vim/nvim

### Using git/github to track your progress

What is Git?

Basic Git Commands

What is GitHub?

Getting Started with Git and GitHub

Conclusion

Chatgpt

Gallery generated by Midjourney

Acknowledgement

Some fun images generated by Midjourney  
for this talk.

Enjoy !

Illustrate the idea If you sharpen the saw you would...



Illustrate the idea If you sharpen the saw you would...



Illustrate the idea If you sharpen the saw you can cut...



Two guys are lumbering one with advanced tool doing...



# Illustrate the idea of Time taken for sharpening...



Show sharp tools are superior than dull tools...



Illustrate An artisan must first sharpen his tools if...

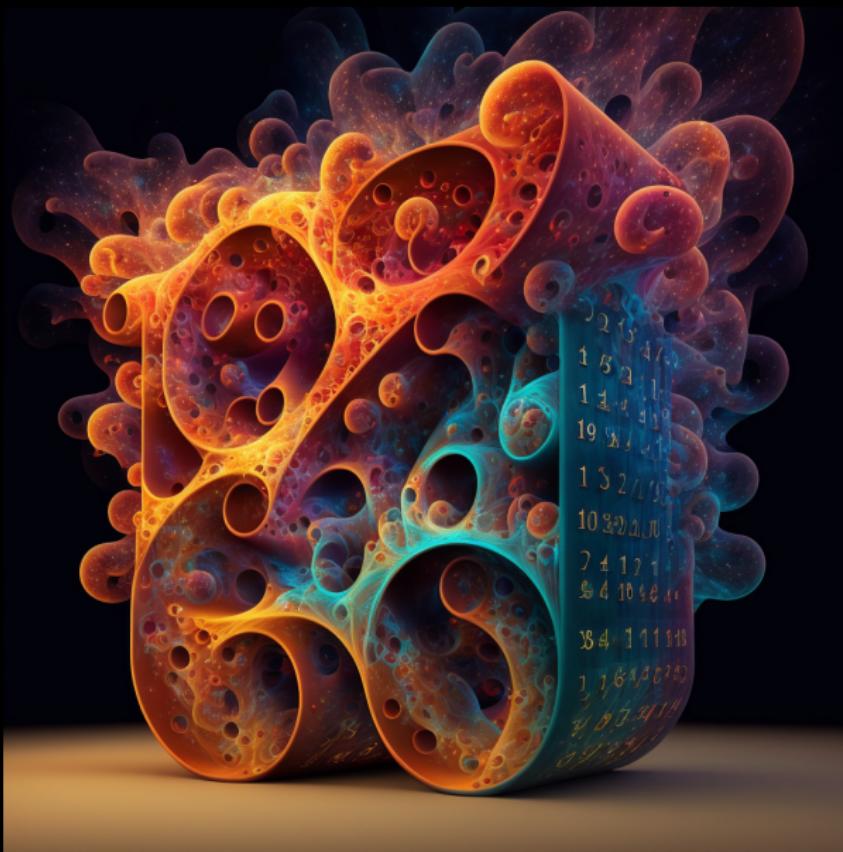


# Chaotic stochastic heat flow on a torus...

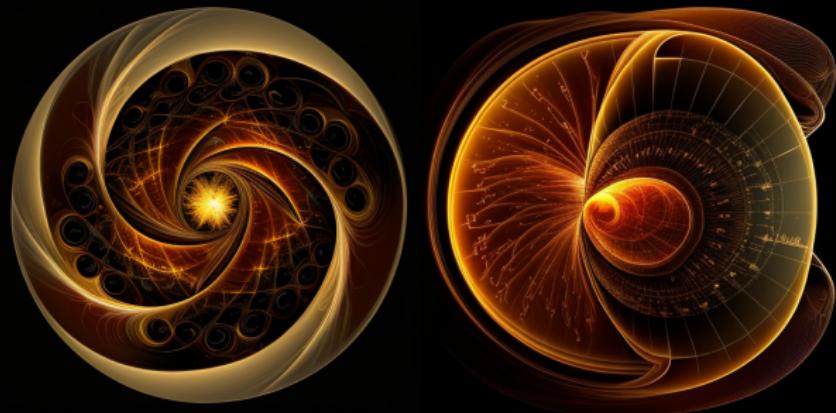




# Stochastic heat equation...



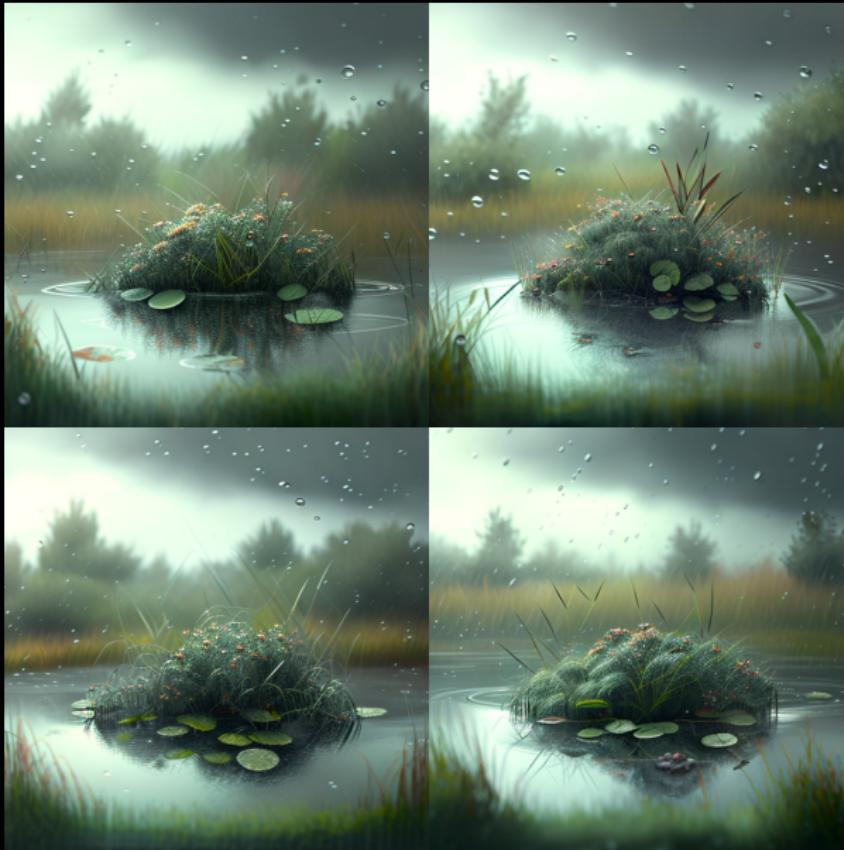
# Intermittency parabolic Anderson model Brownian motion...



A gitar placed in the sand storm in a desert show.png



Show rain drops on a small pond in a foggy raining day...



Lightning clouds dramatic light zigzag highly detailed...



Mushrooms in clusters in a forest with sun shades...



Was it nuclear explosion or cosmic forces beyond comprehension...



Standing on the planet Mars looking at home the earth...



A homesick guy looking back to earth from Mars...



A person in front of the whole universe galaxies...



Cosmic forces beyond comprehension...



# Most famous mathematicians and physicist in the history...



## Introduction to vi/vim/nvim

### Using git/github to track your progress

What is Git?

Basic Git Commands

What is GitHub?

Getting Started with Git and GitHub

Conclusion

Chatgpt

Gallery generated by Midjourney

Acknowledgement

- Le: *Write a sonnet for my math graduate students who attended my seminar talk about the using vim. git, github, and other tools to improve the workflow.*
- Chatgpt:

Oh, math graduate students, heed my call,  
For I bring tools to make your work sublime,  
With vim and git, you'll never fall,  
And on Github, your code will always shine.

Gone are the days of clunky text editors,  
With vim's shortcuts, your fingers will dance,  
No more worries of losing all your data,  
For git's version control will give you a chance.

And on Github, your code will come alive,  
With collaboration at your fingertips,  
Your workflow will be streamlined and thrive,  
And your projects will be free from any slips.

So let us raise our glasses high,  
To vim, git, and Github, our allies.

- Le: *Write a sonnet for my math graduate students who attended my seminar talk about the using vim. git, github, and other tools to improve the workflow.*
- Chatgpt:

Oh, math graduate students, heed my call,  
For I bring tools to make your work sublime,  
With vim and git, you'll never fall,  
And on Github, your code will always shine.

Gone are the days of clunky text editors,  
With vim's shortcuts, your fingers will dance,  
No more worries of losing all your data,  
For git's version control will give you a chance.

And on Github, your code will come alive,  
With collaboration at your fingertips,  
Your workflow will be streamlined and thrive,  
And your projects will be free from any slips.

So let us raise our glasses high,  
To vim, git, and Github, our allies.

- Le: *Write a sonnet for my math graduate students who attended my seminar talk about the using vim, git, github, and other tools to improve the workflow.*
- Chatgpt:

Oh, math graduate students, heed my call,  
For I bring tools to make your work sublime,  
With vim and git, you'll never fall,  
And on Github, your code will always shine.

Gone are the days of clunky text editors,  
With vim's shortcuts, your fingers will dance,  
No more worries of losing all your data,  
For git's version control will give you a chance.

And on Github, your code will come alive,  
With collaboration at your fingertips,  
Your workflow will be streamlined and thrive,  
And your projects will be free from any slips.

So let us raise our glasses high,  
To vim, git, and Github, our allies.

## Acknowledgement

- ▶ Chatgpt
- ▶ Github Copilot
- ▶ Midjourney