

LaMEM short course

16-20 02 2026 Heidelberg

Nicolas Riel- nriel@uni-mainz.de

Needed softwares

- VS Code

<https://code.visualstudio.com/download>



- Julia

<https://julialang.org/downloads/>



- Paraview

<https://www.paraview.org/download/>



- (Co-pilot) - takes a few days for activation, very useful

<https://docs.github.com/en/copilot/managing-copilot/managing-copilot-as-an-individual-subscriber/managing-your-github-copilot-pro-subscription/getting-free-access-to-copilot-pro-as-a-student-teacher-or-maintainer>



WSL - Windows install

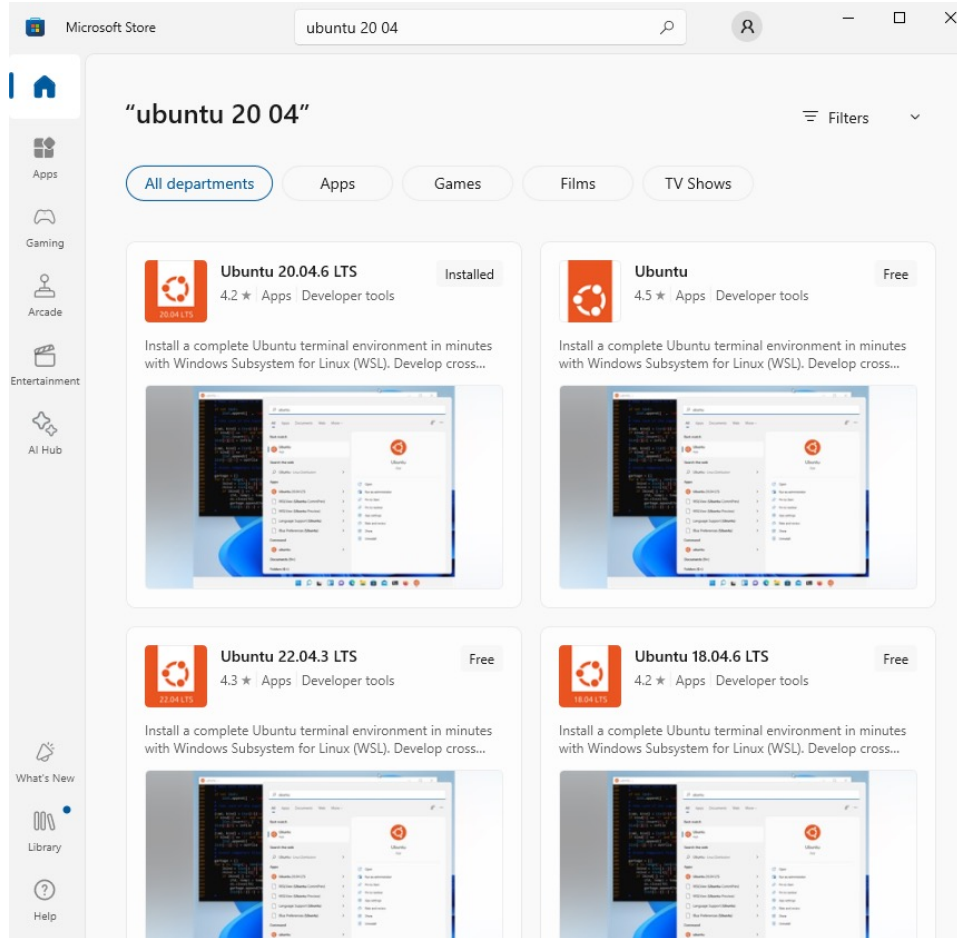
- We won't use windows...
- First install WSL (Windows Subsystem for Linux)

option 1: Microsoft Store
(option 2: PowerShell)

- *Linux/mac users can skip to slide 10*

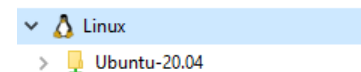
option 1

WSL - Windows install (linux/mac users can skip to slide 8)



- Open Microsoft store (start-up menu)
- Look for Ubuntu 20 04
- Install Ubuntu 20 04
- **Restart computer**
- A terminal will open and ask for setting up a Linux username and password to your Ubuntu

- A new folder in the explorer should appear:



(option 2) WSL - Windows install

Prerequisites

You must be running Windows 10 version 2004 and higher (Build 19041 and higher) or Windows 11 to use the commands below. If you are on earlier versions please see [the manual install page](#).

Install WSL command

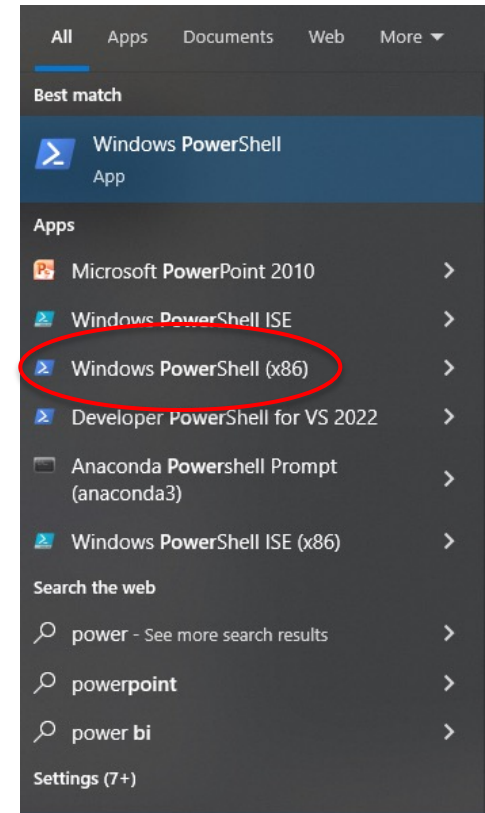
You can now install everything you need to run WSL with a single command. Open PowerShell or Windows Command Prompt in **administrator** mode by right-clicking and selecting "Run as administrator", enter the `wsl --install` command, then restart your machine.

PowerShell

Copy

```
wsl --install
```

<https://learn.microsoft.com/en-us/windows/wsl/install>



(option 2) WSL - Windows install

To install WSL2 on Windows using PowerShell, follow these steps:

1. Open PowerShell as Administrator.

2. Run the following command to enable the WSL feature:

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

3. Enable the Virtual Machine Platform feature:

```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

4. Set WSL2 as the default version:

```
wsl --set-default-version 2
```

5. Download and install the Linux kernel update package:

```
wsl.exe --install
```

6. To install Ubuntu 20.04 on WSL2, follow these steps. First run the following command to list available distributions:

```
wsl --list --online
```

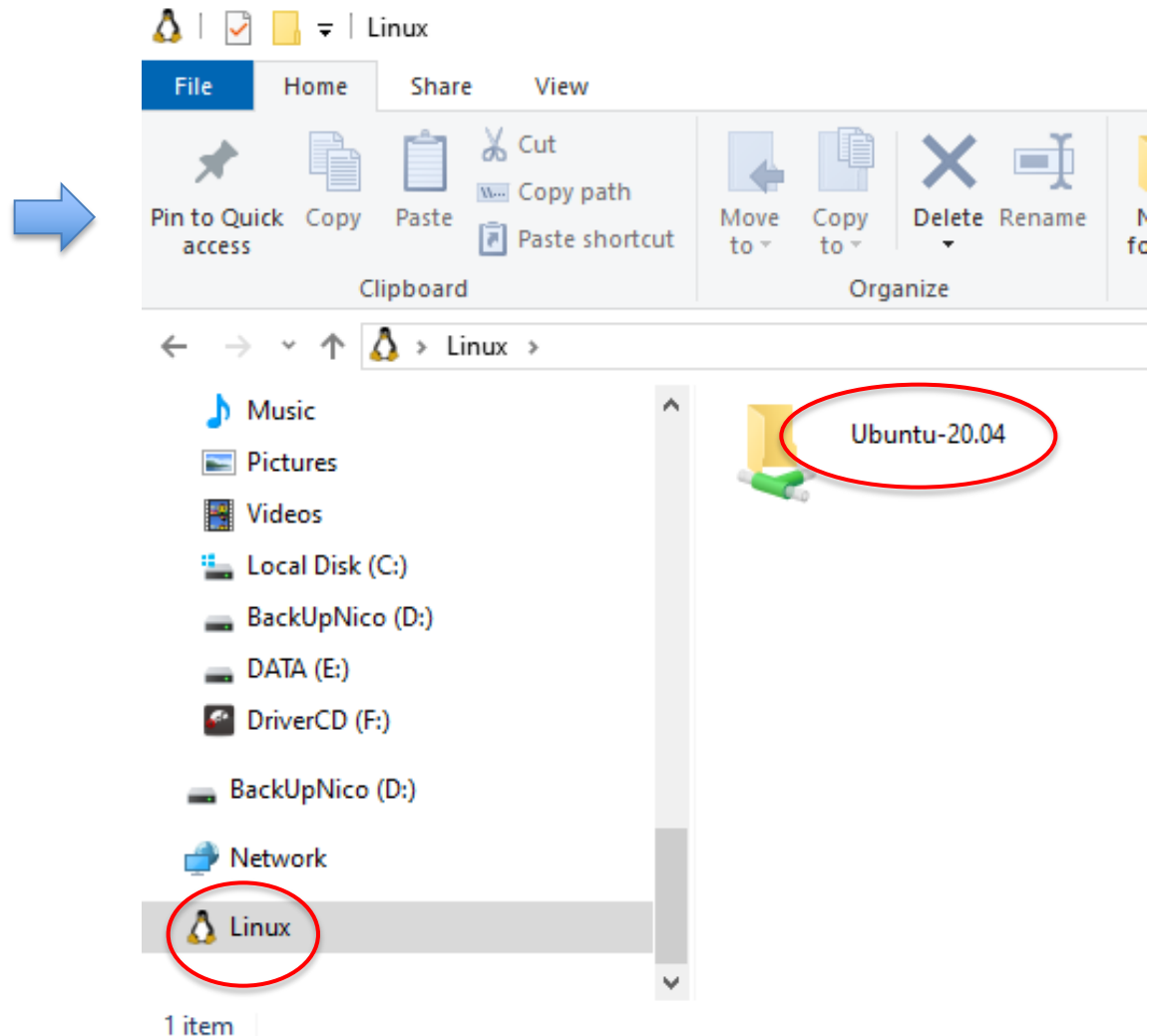
7. Install Ubuntu 20.04 by running:

```
wsl --install -d Ubuntu-20.04
```

8. Once the installation is complete, launch Ubuntu 20.04 from the Start menu and complete the initial setup. Note that username and password need to be provided for the linux

(option 2) WSL - Windows install

Once installed, the Linux folder can be easily accessed in the bottom left of the explorer



WSL - Debug

Search → Windows Features

Activate:

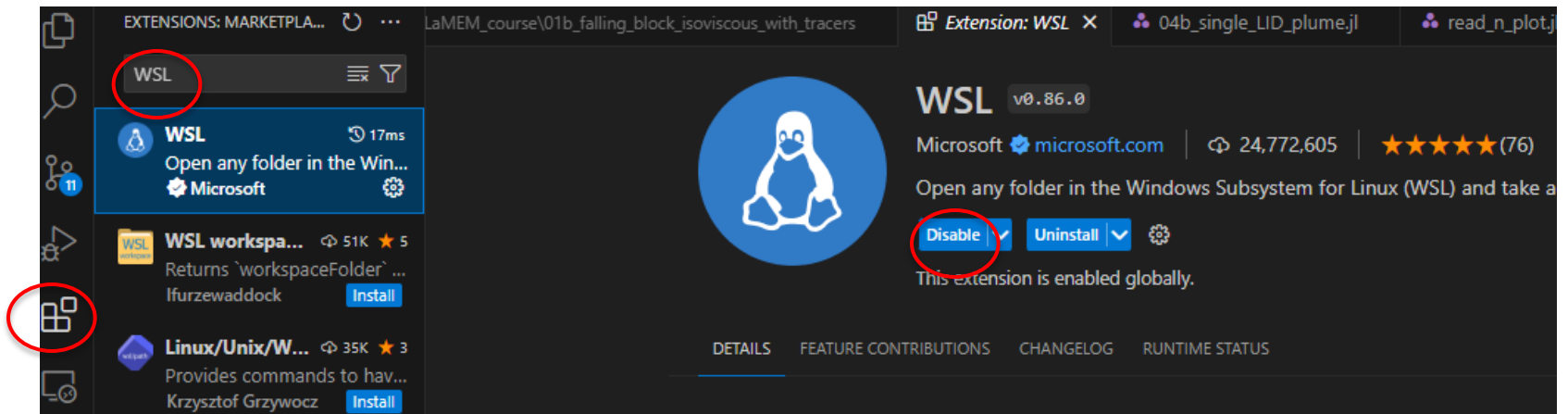
VM

Hyper-v

Sub system

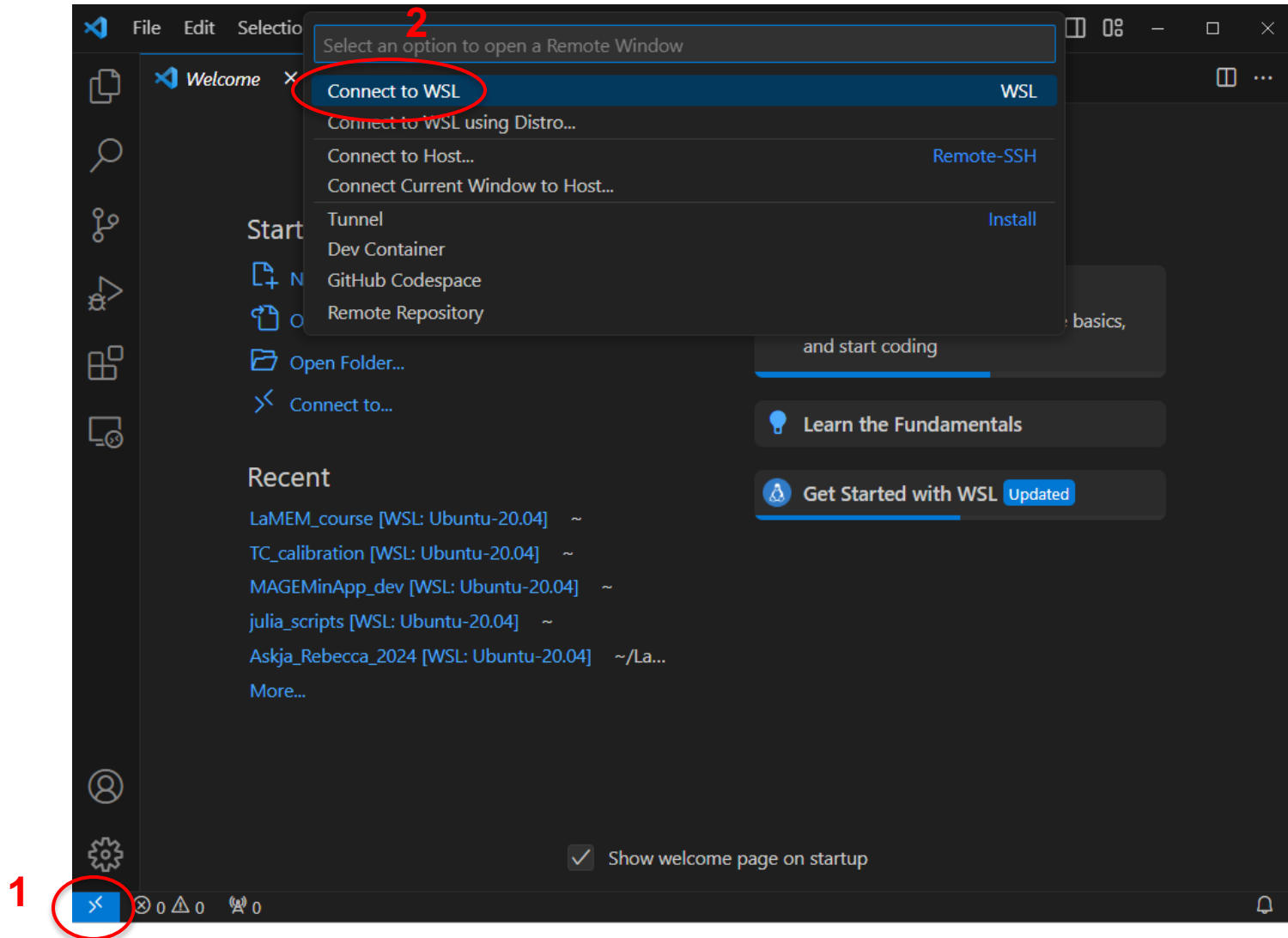
Setup VS-Code for WSL (Windows users)

- Add WSL plugin

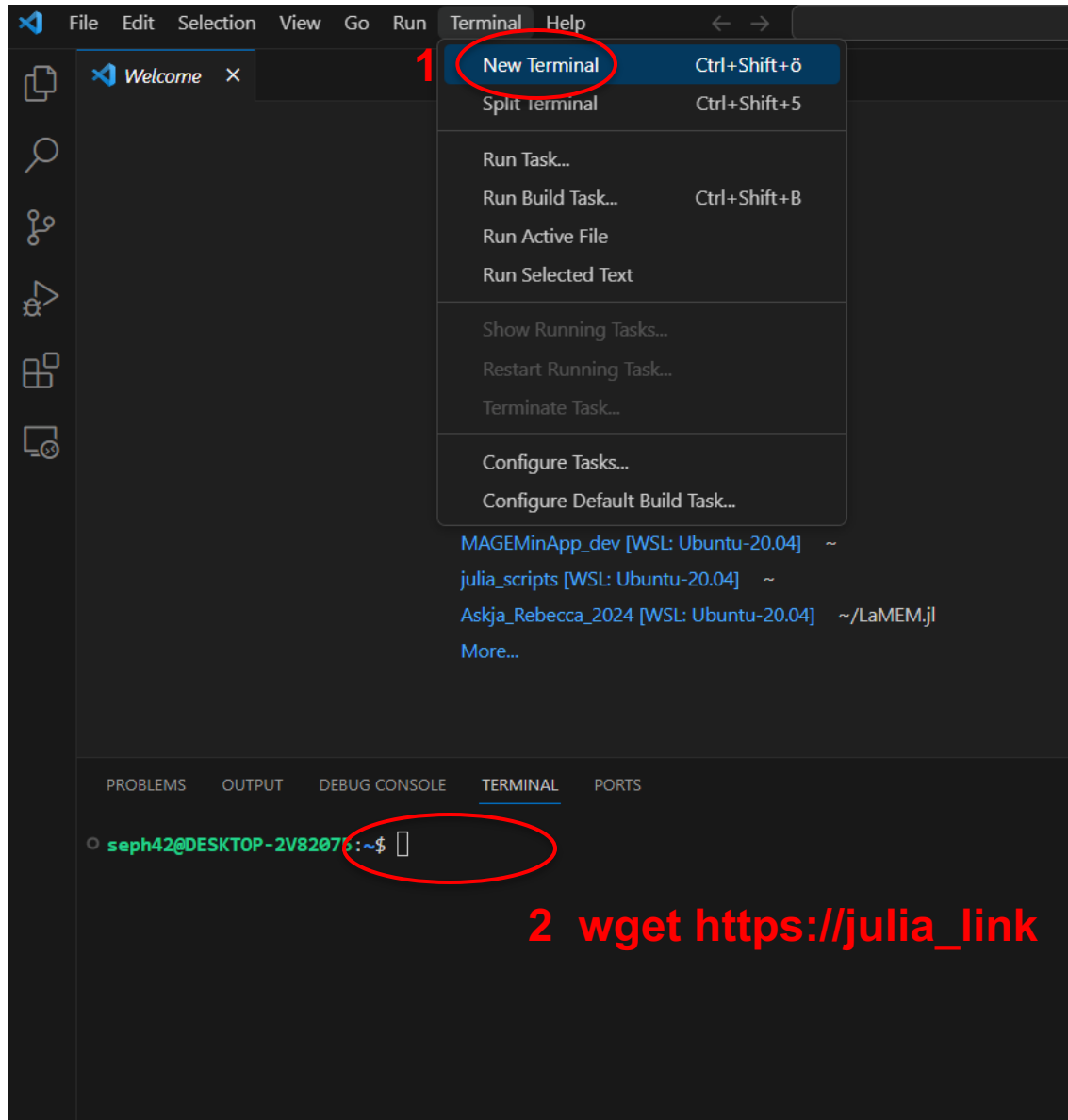


This operation allow to have linux working directly through vscode as a terminal

Connect VS-Code to WSL (Windows users)



Connect VS-Code to WSL (Windows users)



Linux / Mac



- Julia

```
curl -fsSL https://install.julialang.org | sh
```

<https://julialang.org/downloads/>

- Download Julia 1.10 (64 bit)

Current stable release: v1.10.0 (December 25, 2023)

Checksums for this release are available in both [SHA256](#) and [MD5](#) formats.

Platform	64-bit
Windows [help]	installer , portable
macOS x86 (Intel or Rosetta) [help]	.dmg , .tar.gz
macOS (Apple Silicon) [help]	.dmg , .tar.gz
Generic Linux on x86 [help]	glibc (GPG) , musl^[1] (GPG)
Generic Linux on ARM [help]	AArch64 (GPG)
Generic Linux on PowerPC [help]	little endian (GPG)
Generic FreeBSD on x86 [help]	.tar.gz
Source Tarball (GPG) Tarball with dependencies (GPG) GitHub	

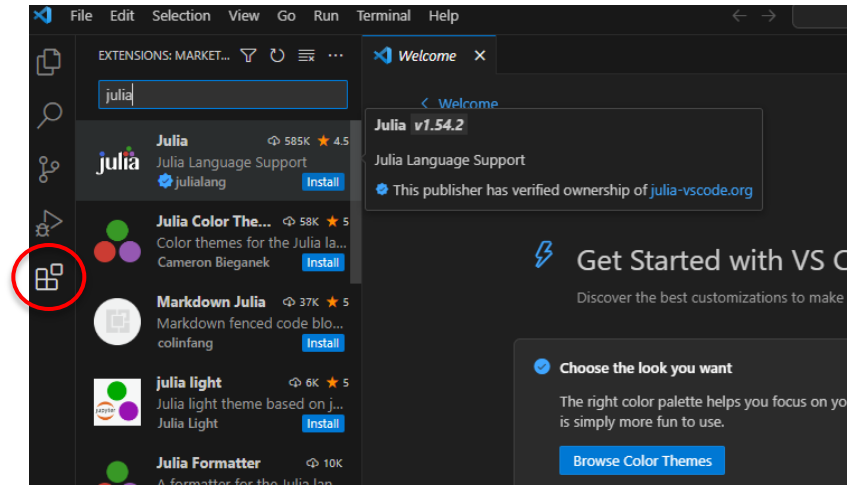
Almost everyone should be downloading and using the latest stable release.

wget http://full_link_(...)

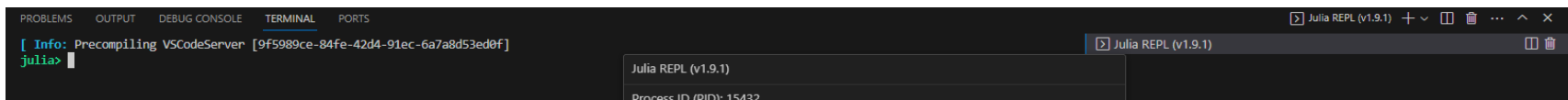
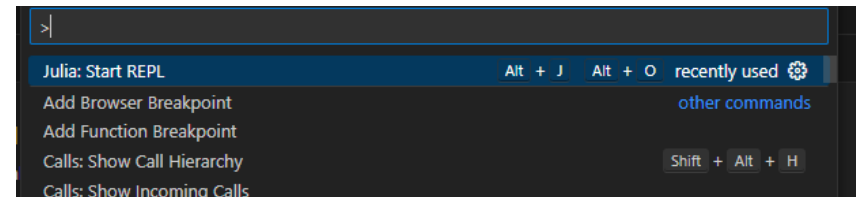
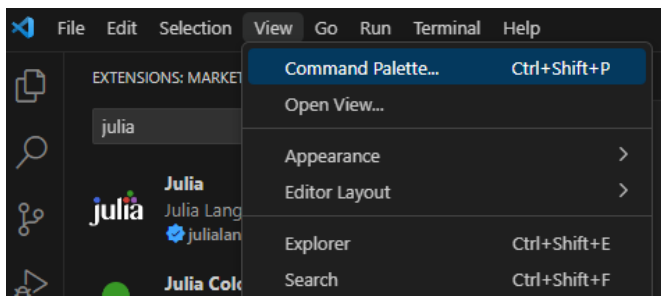
Setting up Julia working environment

Setup VS-Code for Julia

- Add Julia plugin



- Start Julia REPL (terminal)

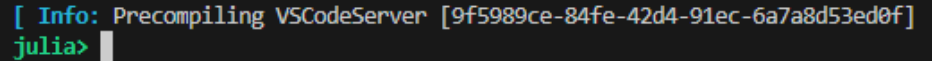


Julia REPL (read-eval-print loop)

→ Terminal or prompt pasting

- Julia terminal

computation space, execute scripts...

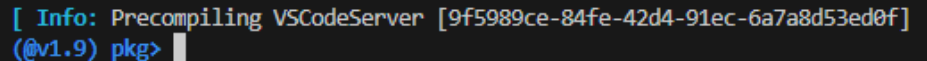


[Info: Precompiling VSCodeServer [9f5989ce-84fe-42d4-91ec-6a7a8d53ed0f]
julia> |

A screenshot of a terminal window showing the Julia REPL prompt. The prompt is 'julia>' followed by a cursor. Above the prompt, there is a status bar message: '[Info: Precompiling VSCodeServer [9f5989ce-84fe-42d4-91ec-6a7a8d53ed0f]'. A blue arrow points from the 'Julia terminal' bullet point to this screenshot.

-] package manager

add/update packages

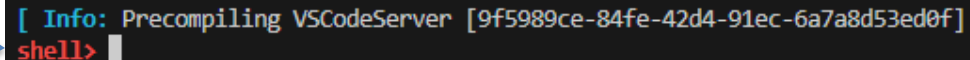


[Info: Precompiling VSCodeServer [9f5989ce-84fe-42d4-91ec-6a7a8d53ed0f]
(@v1.9) pkg> |

A screenshot of a terminal window showing the Julia package manager prompt. The prompt is '(@v1.9) pkg>' followed by a cursor. Above the prompt, there is a status bar message: '[Info: Precompiling VSCodeServer [9f5989ce-84fe-42d4-91ec-6a7a8d53ed0f]'. A blue arrow points from the 'package manager' bullet point to this screenshot.

- ; shell

Changing directories

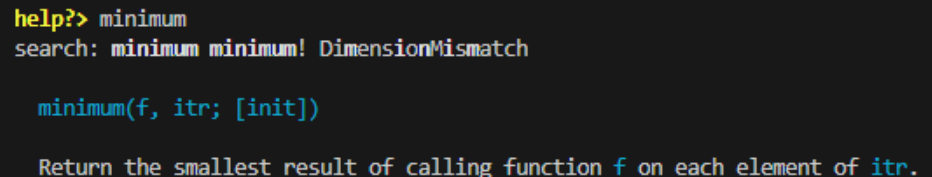


[Info: Precompiling VSCodeServer [9f5989ce-84fe-42d4-91ec-6a7a8d53ed0f]
shell> |

A screenshot of a terminal window showing the Julia shell prompt. The prompt is 'shell>' followed by a cursor. Above the prompt, there is a status bar message: '[Info: Precompiling VSCodeServer [9f5989ce-84fe-42d4-91ec-6a7a8d53ed0f]'. A blue arrow points from the 'shell' bullet point to this screenshot.

- ? help

provide help with functions



help?> minimum
search: minimum minimum! DimensionMismatch

minimum(f, itr; [init])

Return the smallest result of calling function f on each element of itr.

A screenshot of a terminal window showing the Julia help prompt. The prompt is 'help?>' followed by the word 'minimum'. Below the prompt, there is a search result: 'search: minimum minimum! DimensionMismatch'. Below that, there is a function signature: 'minimum(f, itr; [init])'. Below that, there is a description: 'Return the smallest result of calling function f on each element of itr.'. A blue arrow points from the '? help' bullet point to this screenshot.

- Backspace

back to julia terminal

*Note that on Windows the shell is bugged,
navigate through directory using terminal → cd("path"), pwd()*

Add LaMEM.jl (Julia wrapper)

-] add LaMEM

```
[ Info: Precompiling VSCodeServer [9f5989ce-84fe-42d4-91ec-6a7a8d53ed0f]  
(@v1.9) pkg> add LaMEM
```



- Test LaMEM

```
Installed XML2_jll ————— v2.11.5+0  
Installed GeoInterface ————— v1.3.2  
Installed GeophysicalModelGenerator — v0.5.5  
Downloaded artifact: XML2  
Updating `C:\Users\Seph\.julia\environments\v1.9\Project.toml`  
.  
.  
.
```

(takes several minutes)

```
(@v1.9) pkg> test LaMEM
```



```
-----  
Test Summary: | Pass Total Time  
run LaMEM | 6 6 1m20.8s  
Test Summary: | Pass Total Time  
read LaMEM output | 10 10 4.6s  
No partitioning file required for 1 core model setup  
Test Summary: | Pass Total Time  
run lamem mode save grid test | 2 2 0.3s  
Testing LaMEM tests passed  
  
(@v1.9) pkg>
```

All tests should pass!

Add other packages

-] add GeophysicalModelGenerator ➤ Creates 3D input for LaMEM
-] add GeoParams ➤ Set of tools
-] add GMT ➤ Import topography
-] add PlotlyJS ➤ Plotting routine

Add other packages in a local environment

➤ For phase diagrams

Create a MAGEMinApp directory

```
mkdir MAGEMinApp
cd MAGEMinApp
julia
julia> ]
pkg> activate .
add MAGEMinApp
```

➤ For landscape evolution

Create a FastScape directory

```
mkdir FastScape
cd FastScape
julia
julia> ]
pkg> activate .
add
```

<https://github.com/boriskaus/FastScape.jl>

Note:

- Every time you want to use FastScape (if you closed the Julia terminal) you need to open a terminal, change path to FastScape then execute Step 2.

Setup overview



Visual Studio Code

- Terminal: launch Julia, change directory...
 - File → open folder
 - Shell → cd ../
- Create modify scripts, input files for LaMEM/FastScape (*.jl)
 - File → new file → save as *.jl
- Perform simulations



- Visualize simulation results

Windows users

WSL – Ubuntu

Access through shortcut:
Explorer right click on
Ubuntu → home/name or root
Send shortcut to Desktop