



# MAGEMin framework

(Mineral Assemblage Gibbs Energy Minimization)


**MAGEMin** 

- MPI-parallel C code
- Point-wise minimization at given P-T-X





- Metapelite (White et al., 2014)
- Metabasite (Green et al., 2016)
- Igneous (Holland et al., 2018)
- Ultramafic (Evans & Forst, 2021)
- Mantle (Stixude & Lithgow-Bertelloni, 2010)

**MAGEMin\_C**




- Julia wrapper of the C code
- Flexible programming interface
- Database calibration
- Geodynamic coupling

**MAGEMinApp**

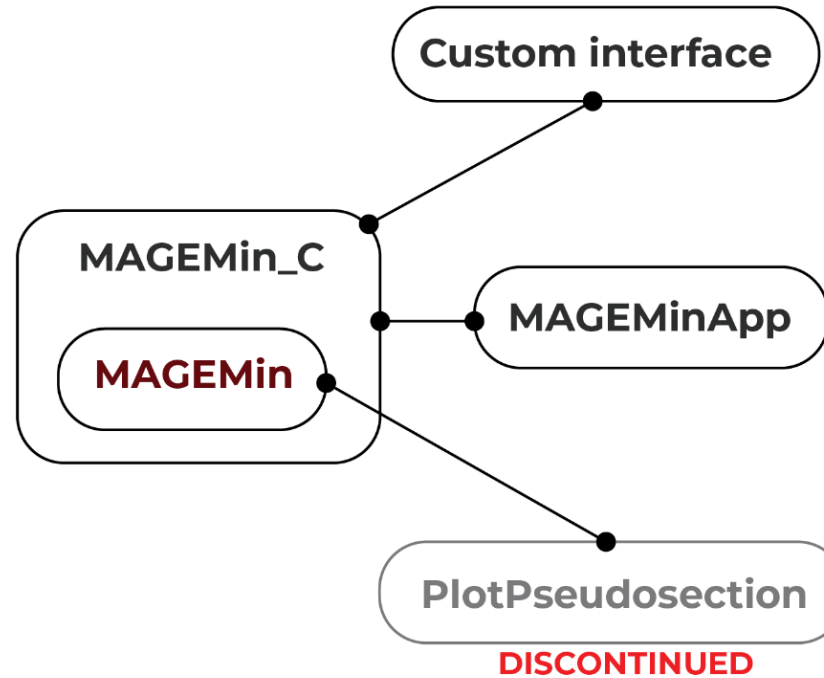
- Web browser app (graphic user interface)
- Parallel point-wise minimization
- PT, PX, TX and PT-X phase diagrams
- Auto labelling, contouring
- Fractional melting/crystallization paths

**PlotPseudosection**

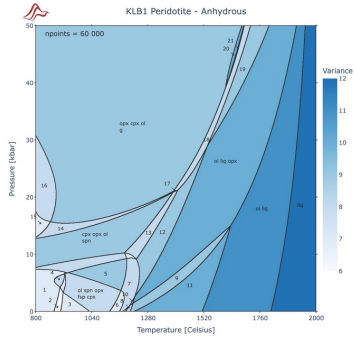


- Matlab app (Graphic user interface)
- Parallel point-wise minimization
- PT, PX, TX diagrams
- Contouring
- PT paths
- Trace element partitioning for mafic to ultramafic systems

E.G., GEODYNAMIC COUPLING



# MAGEMin github: ComputationalThermodynamics



nriel@uni-mainz.de

**ComputationalThermodynamics**  
Tools for thermodynamic computing  
14 followers · Germany

Popular repositories

- MAGEMin** (Public) · C · 63 stars · 15 forks  
The parallel Mineral Assemblage Gibbs Energy Minimization package
- MAGEMin\_C.jl** (Public) · C · 10 stars · 4 forks  
Julia interface to the MAGEMin C package
- MAGEMinApp.jl** (Public) · Julia · 5 stars · 2 forks  
Graphical User Interface for MAGEMin, which runs in your web-browser.
- SandBox** · Julia · 0 stars · 0 forks

README · GPL-3.0 license · Security

## MAGEMin\_C.jl

CI passing DOI [10.5281/zenodo.11217861](https://doi.org/10.5281/zenodo.11217861)

Julia interface to the MAGEMin C package, which performs thermodynamic equilibrium calculations. See the [MAGEMin](#) page for more details on the package & how to use it.

### Using the julia interface

First install julia. We recommend downloading the official binary from the [julia](#) webpage.

Next, install the `MAGEMin_C` package with:

```
julia> ]  
pkg> add MAGEMin_C
```

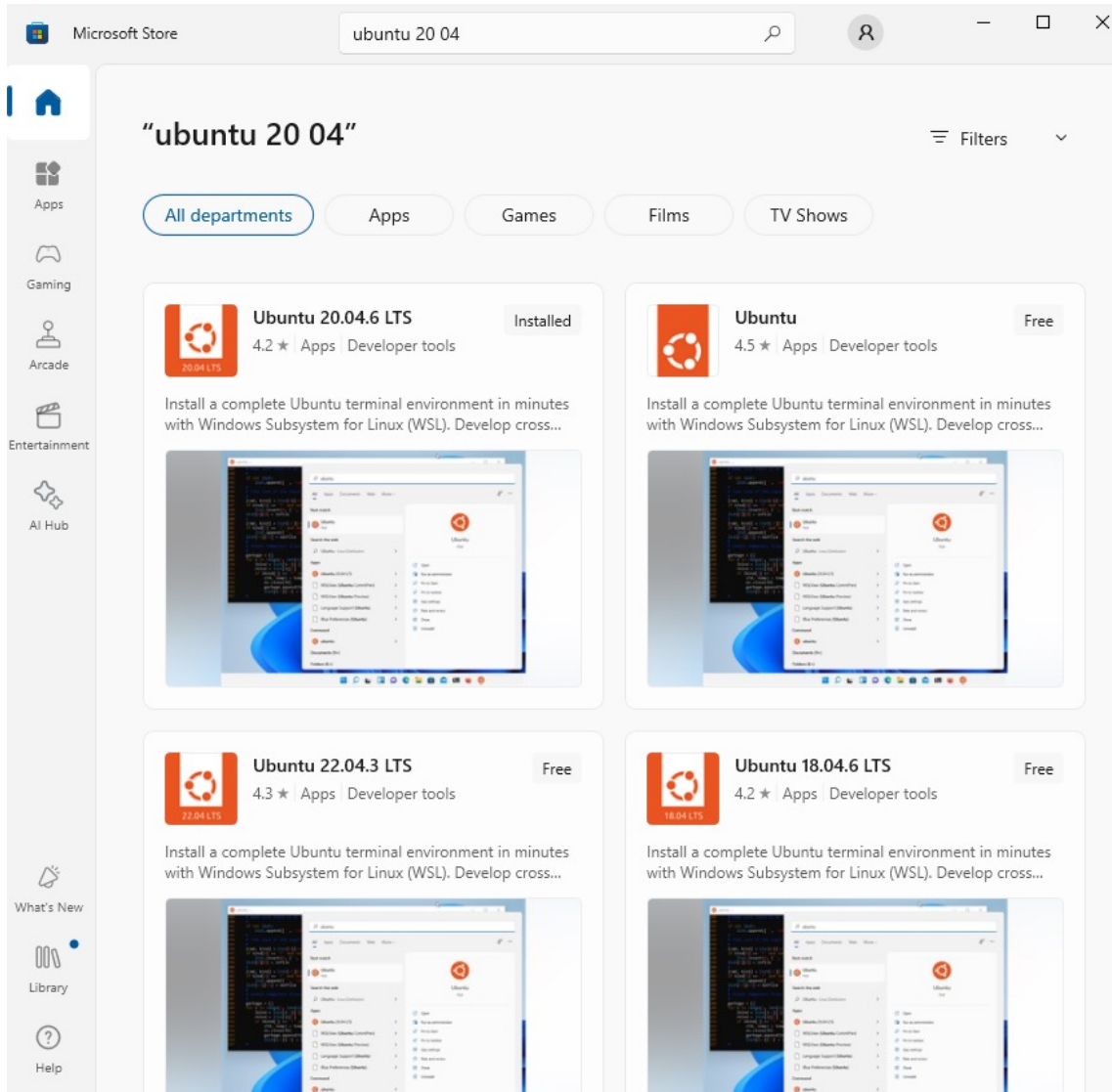


<https://github.com/ComputationalThermodynamics>

[https://github.com/ComputationalThermodynamics/MAGEMin\\_C](https://github.com/ComputationalThermodynamics/MAGEMin_C)

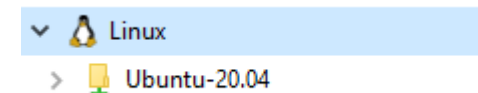
<https://github.com/ComputationalThermodynamics/MAGEMinApp>

# MAGEMinApp installation (Windows with WSL)



- 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:



# Visual Studio Code installation

- <https://code.visualstudio.com/>
- Download and install

The image shows the Visual Studio Code website and a running instance of the editor. The website header includes the Visual Studio Code logo, navigation links (Docs, Updates, Blog, API, Extensions, FAQ, Learn), a search bar, and a 'Download' button. A banner below the header states 'Version 1.91 is now available! Read about the new features and fixes from June.' The main content area features the text 'Code editing. Redefined.' and 'Free. Built on open source. Runs everywhere.' Below this is a 'Download for Windows' button with a dropdown arrow, and a link for 'Web, Insiders edition, or other platforms'. A disclaimer at the bottom of the website reads: 'By using VS Code, you agree to its license and privacy statement.'

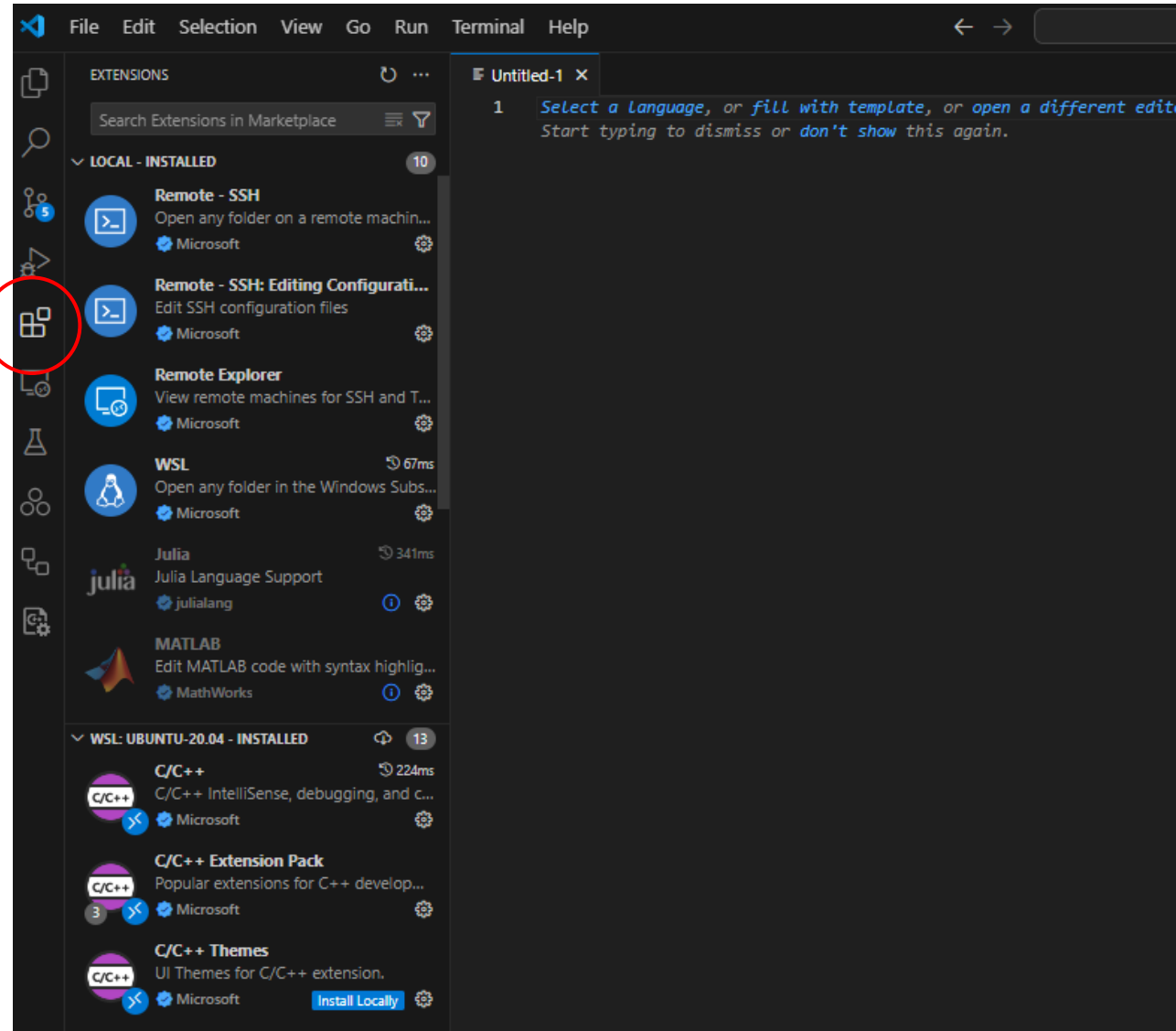
The screenshot of the Visual Studio Code editor shows the 'EXTENSIONS: MARKETPLACE' sidebar on the left, listing various extensions such as Python, GitLens, C/C++, ESLint, Debugger for Chrome, Language Support for Java, vscode-icons, Vetur, and C#. The main editor area displays a JavaScript file named 'serviceWorker.js' with the following code:

```
src > JS serviceWorker.js > register > window.addEventListener('load') callback
39
40
41 // Add some additional logging to localhost, p...
42 // service worker/PWA documentation.
43 navigator.serviceWorker.ready.then(() => {
44
45   product
46   productSub
47   removeSiteSpecificTrackingException
48   removeWebWideTrackingException
49   requestMediaKeySystemAccess
50   sendBeacon
51   serviceWorker (property) Navigator.serviceWorke...
52   storage
53   storeSiteSpecificTrackingException
54   storeWebWideTrackingException
55   userAgent
56   vendor
57
58 function registerValidSW(swUrl, config) {
59   navigator.serviceWorker
60     .register(swUrl)
61     .then(registration => {
```

The terminal window at the bottom shows the command '1: node' and the output: 'You can now view create-react-app in the browser. Local: http://localhost:3000/ On Your Network: http://10.211.55.3:3000/ Note that the development build is not optimized.'

# Visual Studio Code installation

- Install WSL plugin



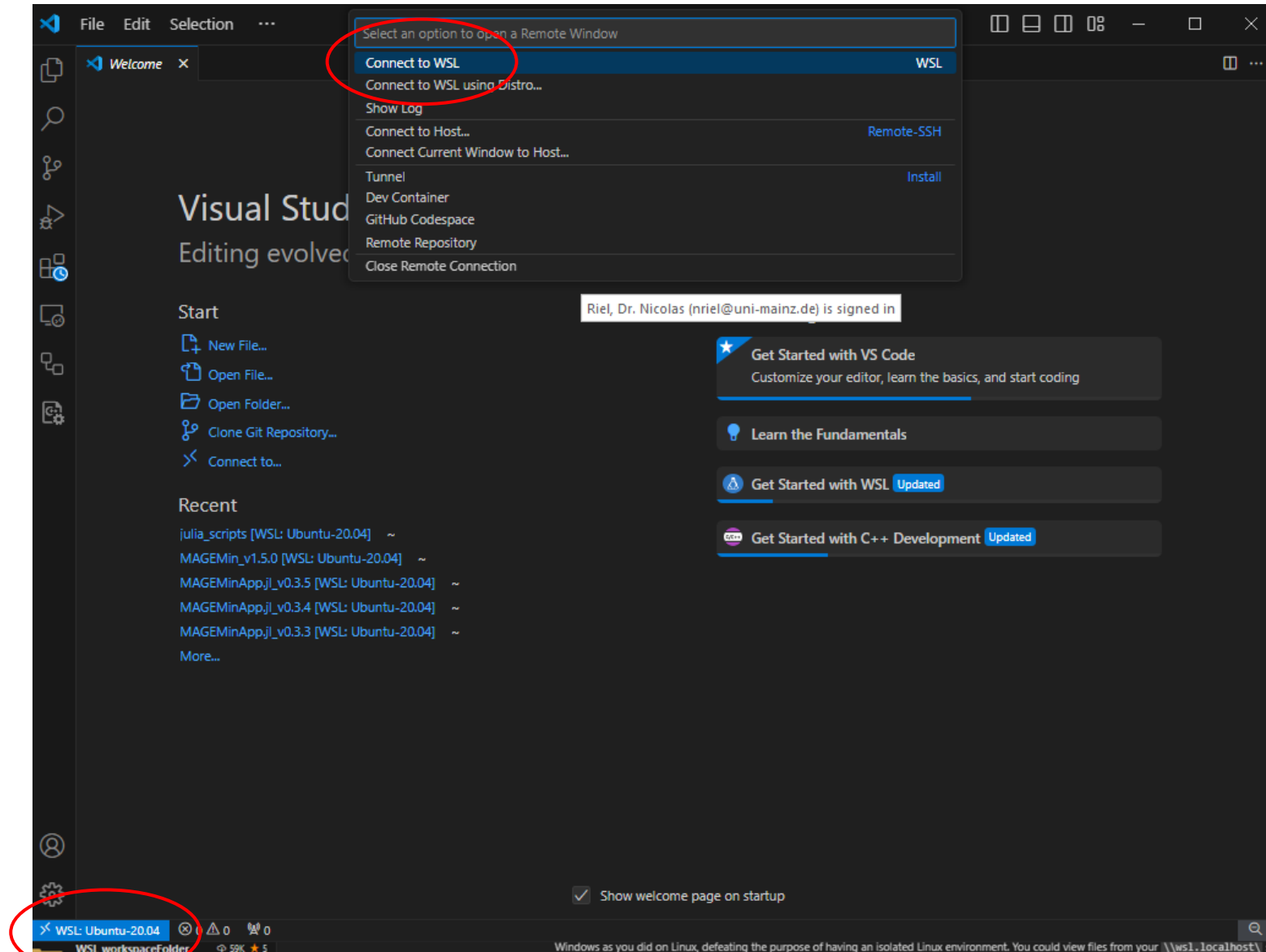
# Visual Studio Code installation

- Install WSL plugin

The screenshot shows the Visual Studio Code Extensions Marketplace interface. On the left, a list of extensions is displayed, with 'WSL' by Microsoft at the top. The main panel shows the details for the 'WSL' extension (version 0.88.2) by Microsoft. The extension description reads: 'Open any folder in the Windows Subsystem for Linux (WSL) and take advantage of Visual Studio Code's full feature set.' Below the description, there are 'Disable' and 'Uninstall' buttons. The 'Uninstall' button is circled in red. The page also includes sections for 'Visual Studio Code WSL', 'Why do I need the WSL extension?', 'Why WSL?', and 'Why the WSL extension in VS Code?'. The 'Uninstall' button is highlighted with a red circle.

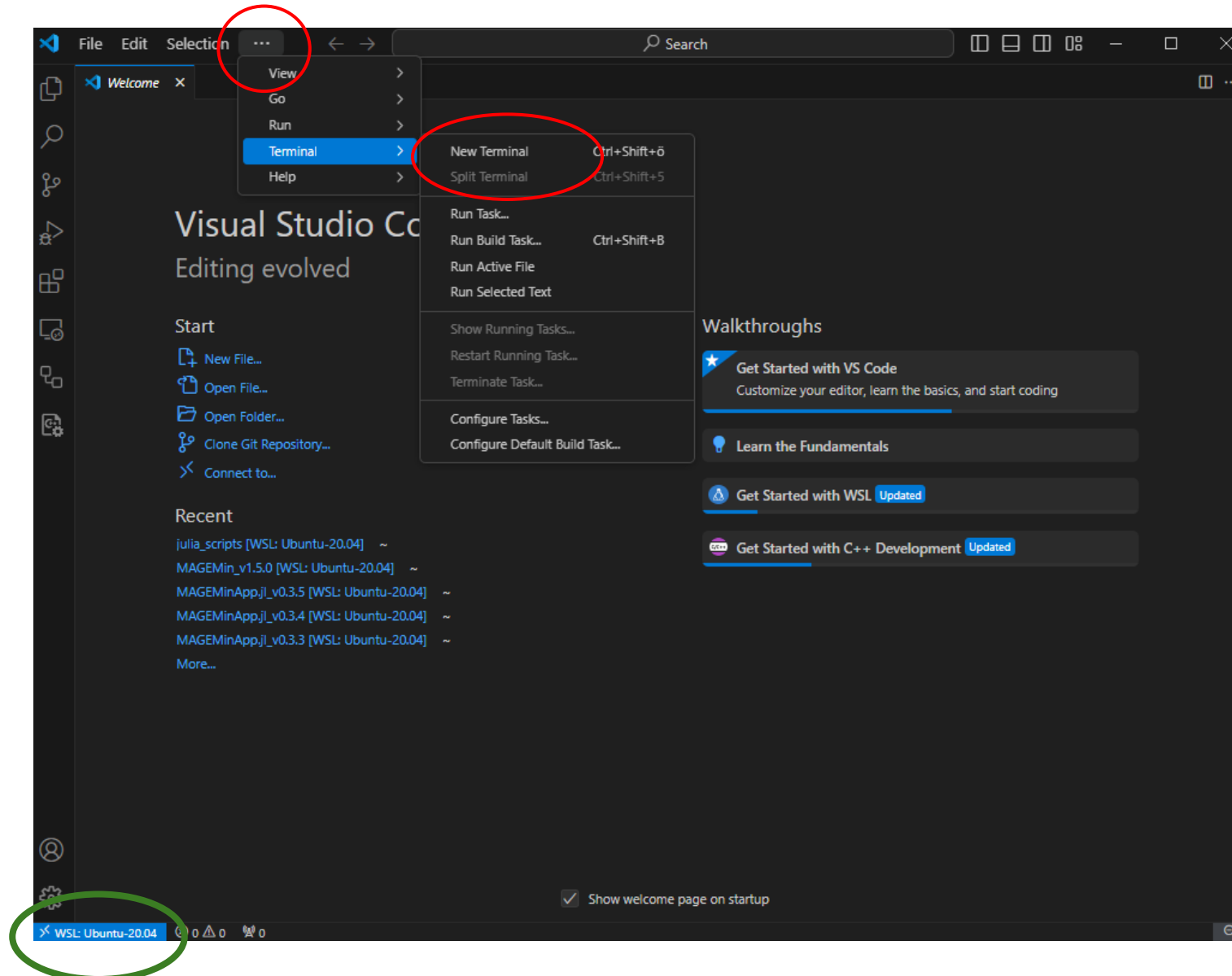
# Visual Studio Code installation

- Connect vscode to WSL



# Visual Studio Code installation

- Open a new terminal





# Julia installation

- Get the command to download Julia:  
`curl -fsSL https://install.julialang.org | sh`

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



[Download](#)

[Documentation](#)

[Learn](#)

[Blog](#)

[Community](#)

[Contribute](#)

[JSOC](#)

## Install julia

Install the latest Julia version ([v1.10.4](#) June 4, 2024) by running this in your terminal:

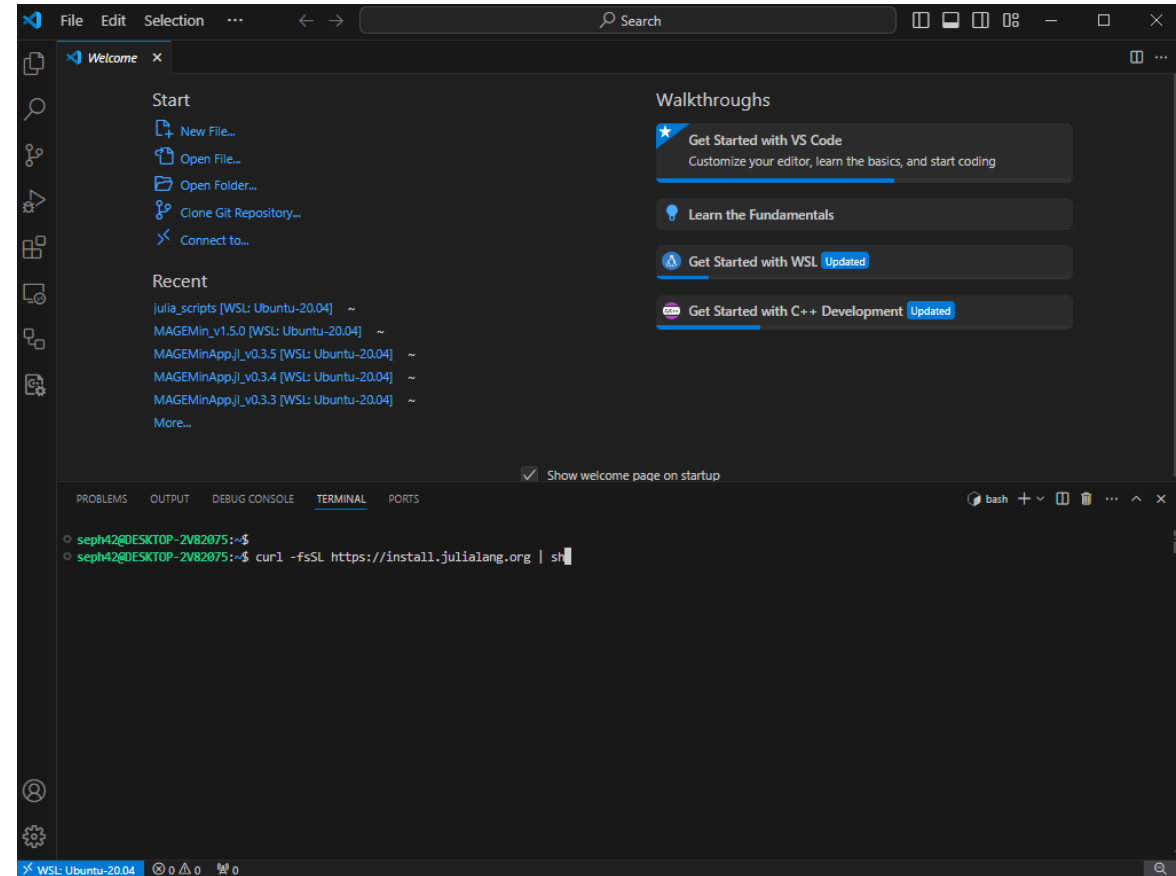
```
$ curl -fsSL https://install.julialang.org | sh
```

For Windows instructions [click here](#)

Once installed `julia` will be available via the command line interface.

This will install the `Juliaup` installation manager, which will automatically install julia and help keep it up to date. The command is `juliaup --help`.

Please star us [on GitHub](#). If you use Julia in your research, please [cite us](#). If possible, do consider [sponsoring us](#).



Please do not use the version of "Julia" shipped by unix package managers

Many unix package managers ship broken and/or significantly out of date versions of Julia. Please use `juliaup` or download the c

- Execute the command in the terminal



# Launch MAGEMinApp

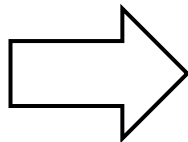
- In the Julia terminal, type:  
‘using MAGEMinApp’  
Then  
‘App()’
- The following text will be displayed in the terminal:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
seph42@DESKTOP-2V82075:~$ julia -t 6
Documentation: https://docs.julialang.org
Type "?" for help, "]" for Pkg help.
Version 1.10.0 (2023-12-25)
Official https://julialang.org/ release

(@v1.10) pkg> add MAGEMinApp
Resolving package versions...
No Changes to `~/julia/environments/v1.10/Project.toml`
No Changes to `~/julia/environments/v1.10/Manifest.toml`

julia> using MAGEMinApp
Using libMAGEMin.dylib from MAGEMin_jll

julia> App()
Info: Listening on: 127.0.0.1:8050, thread id: 1
```



- Copy and paste the address in your web-browser:

Reminder: Don't forget to properly cite the references used to create the diagrams by using the 'export references' buttons!

An example of file providing bulk-rock compositions is given in the 'examples' folder. The structure of the file should comply with the following structure:  
title:String; comments:String; db:String; sysUnit:String; oxide:Vector{String}; fac:Vector{PhaseID}

oxide	mol_fraction
SiO2	38.494
Al2O3	1.776
CaO	2.824
HgO	50.566
FeO	5.886
K2O	0.01
Na2O	0.25
TiO2	0.1
O	0.096
Cr2O3	0.109
H2O	0