Scientific Computing for Physicist

Jin-Guo Liu

Yu-Sheng Zhao

Jin-Guo Liu
Hong-Kong University of Science and Technology (Guangzhou)

Yu-Sheng Zhao
Hong-Kong University of Science and Technology (Guangzhou)

https://book.jinguo-group.science

Version: 2024-01-23

Contents

1 About

This is an example book, built via Books.jl[1] and is made possible by the Julia programming language (Bezanson et al., 2017) and pandoc[2].

[1] https://books.huijzer.xyz

[2] https://github.com/jgm/pandoc

2 Setup Julia

This setup guide is adapted with the mainland China users in mind. If you are not in mainland China, you may skip some steps.

2.1 Step 1: Installing Julia

For Linux/Mac users, please open a terminal and type the following command to install Julia[1] with juliaup[2]. `Juliaup` is a tool to manage Julia versions and installations. It allows you to install multiple versions of Julia and switch between them easily.

[1] https://julialang.org/
[2] https://github.com/JuliaLang/juliaup

```
curl -fsSL https://install.julialang.org | sh # Linux and macOS
```

For Windows users, please open execute the following command in a `cmd`,

```
winget install julia -s msstore # Windows
```

You can also install Juliaup directly from Windows Store[3].

[3] https://www.microsoft.com/store/apps/9NJNWW8PVKMN

2.2 For users suffering from the slow download speed

You may need to specify another server for installing Juliaup. To do so, execute the following command in your terminal before running the script above.

Linux and macOS

```
export JULIAUP_SERVER=https://mirror.nju.edu.cn/julia-releases/ # Linux & macOS
```

Windows

```
$env:JULIAUP_SERVER="https://mirror.nju.edu.cn/julia-releases/" # Windows
```

An alternative approach is downloading the corresponding Julia binary from the Nanjing university mirror website[4]. After installing the binary, please set the Julia binary path properly if you want to start a Julia REPL from a terminal, check this manual page[5] to learn more.

[4] https://mirror.nju.edu.cn/julia-releases/
[5] https://julialang.org/downloads/platform/

2.3 *Installing Julia*

To verify that Julia is installed, please open a **new** terminal and run the following command in your terminal. `bash julia` - It should start a Julia REPL(Read-Eval-Print-Loop) session like this - If you wish to install a specific version of Julia, please refer to the documentation[6].

[6] https://github.com/JuliaLang/juliaup

2.4 *Step 2: Package Management*

- `Julia` has a mature eco-system for scientific computing.
- `Pkg` is the built-in package manager for Julia.

- To enter the package manager, press `]` in the REPL.
- The environment is indicated by the `(@v1.9)`.
- To add a package, type `add <package name>`.
- To exit the package manager press `backspace` key
- Read More[7]

[7] https://pkgdocs.julialang.org/v1/managing-packages/

2.5 *Step 3. Configure the startup file and add* `Revise`

First create a new file `~/.julia/config/startup.jl` by executing the following commands

```
mkdir -r ~/.julia/config touch ~/.julia/config/startup.jl
```

You could open the file with your favourite editor and add the following content

```
ENV["JULIA_PKG_SERVER"] = "http://cn-southeast.pkg.juliacn.com/"
try
    using Revise
catch e
    @warn "fail to load Revise."
```

```
end
```

The contents in the startup file is executed immediately after you open a new Julia session.

Then you need to install Revise[8], which is an Julia package that can greatly improve the using experience of Julia. To install `Revise`, open Julia REPL and type

[8] https://github.com/timholy/Revise.jl

```
julia> using Pkg; Pkg.add("Revise")
```

If you don't know about `startup.jl` and where to find it, here[9] is a good place for further information.

[9] https://docs.julialang.org/en/v1/manual/command-line-interface/#Startup-file

2.6 *More Packages*

- You may find more Julia packages here[10].

[10] https://juliahub.com/

As a final step, please verify your Julia configuration by openning a Julia REPL and type

```
julia> versioninfo()
Julia Version 1.9.2
Commit e4ee485e909 (2023-07-05 09:39 UTC)
Platform Info:
  OS: macOS (arm64-apple-darwin22.4.0)
  CPU: 10 × Apple M2 Pro
  WORD_SIZE: 64
  LIBM: libopenlibm
  LLVM: libLLVM-14.0.6 (ORCJIT, apple-m1)
  Threads: 1 on 6 virtual cores
Environment:
  JULIA_NUM_THREADS = 1
  JULIA_PROJECT = @.
  JULIA_PKG_SERVER = http://cn-southeast.pkg.juliacn.com/
```

2.7 *Step 3. Download an editor: VSCode*

Install VSCode by downloading the correct binary for your platform from here[11]. Open VSCode and open the `Extensions` tab on the left side-bar of the window, search `Julia` and install the most popular extension. read more...[12]

[11] https://code.visualstudio.com/download

[12] https://github.com/julia-vscode/julia-vscode

You are ready to go, cheers!

2.8 A quick introduction to the Julia REPL

A Julia REPL has four modes,

1. Julian mode is the default mode that can interpret your Julia code.
2. Shell mode is the mode that you can run shell commands. Press `;` in the Julian mode and type

```
shell> date
Sun Nov  6 10:50:21 PM CST 2022
```

To return to the Julian mode, type the Backspace key.

3. Package mode is the mode that you can manage packages. Press `]` in the Julian mode and type

```
(@v1.8) pkg> st
Status `~/.julia/environments/v1.8/Project.toml`
  [295af30f] Revise v3.4.0
```

To return to the Julian mode, type the Backspace key.

4. Help mode is the mode that you can access the docstrings of functions. Press `?` in the Julian mode and type

```
help> sum
... docstring for sum ...
```

To return to the Julian mode, type the Backspace key.

read more...[13]

[13] https://docs.julialang.org/en/v1/stdlib/REPL/

3 The open source workflow

Here are some contents

3.1 Subsection

3.1.1 Subsubsection

Appendix

This is the appendix.

A	B	C
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6

Table 3.1: Example dataframe.

References

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98.