

1 Lecture 0: Running Julia code

Most of the material in this module will be illustrated in the [Julia programming language](#). Programming in Julia will not be examined but you might find it useful (and fun) for the project (see the module guide) or for computational experimentation with the material in this module.

1.1 Instructions

1. Download Julia at <https://julialang.org/downloads/> and install.
2. Run Julia. This should open the Julia REPL.
3. Install the required packages by typing (in the Julia REPL, with patience)

```
import Pkg
using Pkg
Pkg.add("ApproxFun")
Pkg.add("Plots")
Pkg.add("GR")
Pkg.add("Plotly")
Pkg.add("PlotlyJS")
Pkg.add("Interact")
Pkg.add("IJulia")
Pkg.add("DifferentialEquations")
Pkg.add("ComplexPhasePortrait")
Pkg.add("OscillatoryIntegrals")
Pkg.add("SpecialFunctions")
```

4. Open a Jupyter notebook by typing:

```
using IJulia
notebook()
```

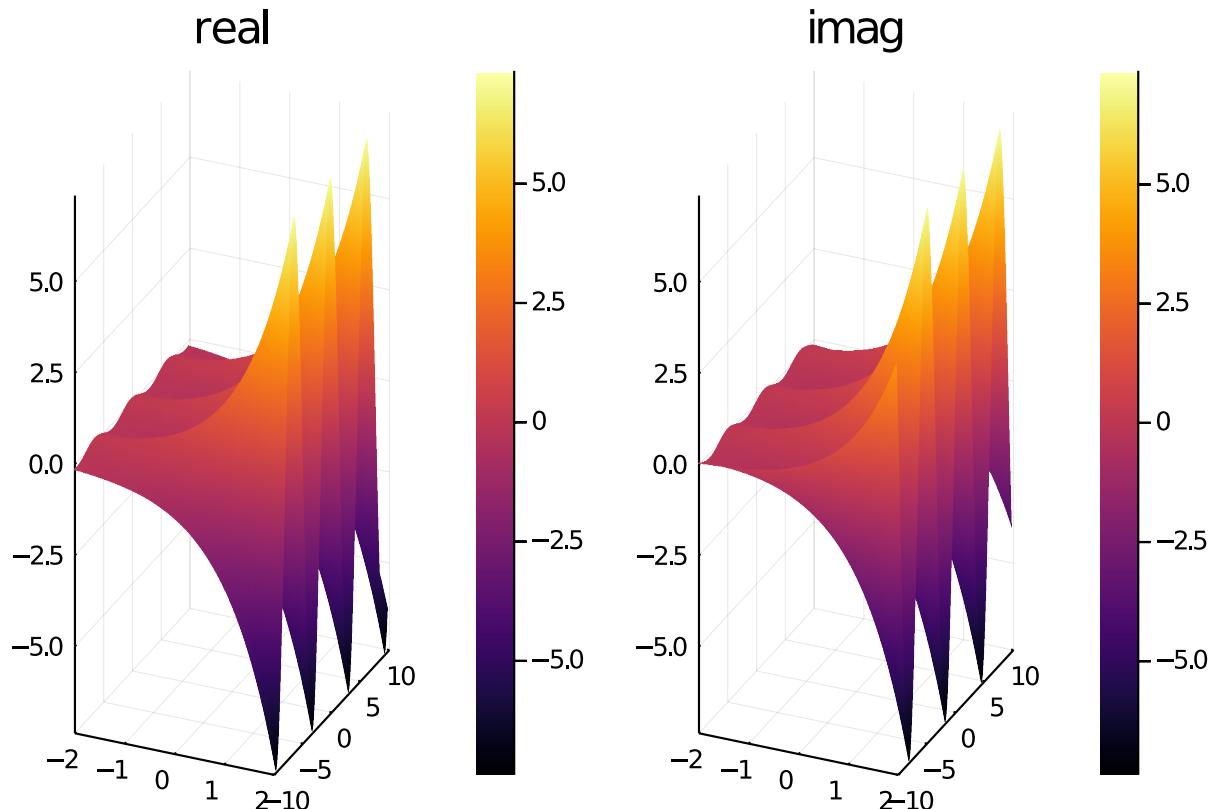
The first time you run `notebook()`, it will prompt you for whether it should install Jupyter. Hit enter to have it use the Conda.jl package to install a minimal Python+Jupyter distribution (via Miniconda) that is private to Julia (not in your PATH). If you need more detailed information on opening a Jupyter notebook, see the [IJulia page](#).

5. In the top right of the tab that has been opened in your browser, click on `New` and then click on your version of Julia, e.g., Julia 1.5.3
6. Let's take the Julia code in the notes for Lecture 1 as an example. In the first cell of the Jupyter notebook, type

```
using Plots, ComplexPhasePortrait, SpecialFunctions
gr();
```

Note that 'gr()' only needs to be included the first time you type [Using Plots](#). That's why the command 'gr()' is not in the lecture notes. To run the cell, press Ctrl+Enter. Wait patiently for the packages to precompile. Then type, in a new cell,

```
f = z -> exp(z)
u = z -> real(f(z))
v = z -> imag(f(z))
# set up plotting grid
xx = range(-2 ; stop=2, length=100)
yy = range(-10; stop=10, length=100)
plot(surface(xx, yy, u.(xx' .+ im.*yy); title="real"),
surface(xx, yy, v.(xx' .+ im.*yy); title="imag"))
```



You can also include text and LaTeX code in Jupyter notebooks. Alternatives to Jupyter notebooks include [Juno](#) and [Pluto](#).