# Introduction to Julia

# Advantages

- Julia is fast
  - “Julia’s LLVM-based just-in-time (JIT) compiler combined with the language’s design allow it to approach and often match the performance of C.”
  - Where benchmark times are relative to C (i.e., C performance = 1)

| algorithms | Fortran | Julia | Python | R | Matlab | Octave | Mathe-matica | JavaScript | Go | LuaJIT | Java |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | gcc 5.1.1 | 0.4.0 | 3.4.3 | 3.2.2 | R2015b | 4.0.0 | 10.2.0 | V8 3.28.71.19 | go1.5 | gsl-shell 2.3.1 | 1.8.0_45 |
| fib | 0.7 | 2.11 | 77.76 | 533.52 | 26.89 | 9324.35 | 118.53 | 3.36 | 1.86 | 1.71 | 1.21 |
| parse_int | 5.05 | 1.45 | 17.02 | 45.73 | 802.52 | 9581.44 | 15.02 | 6.06 | 1.2 | 5.77 | 3.35 |
| quicksort | 1.31 | 1.15 | 32.89 | 264.54 | 4.92 | 1866.01 | 43.23 | 2.7 | 1.29 | 2.03 | 2.6 |
| mandel | 0.81 | 0.79 | 15.32 | 53.16 | 7.58 | 451.81 | 5.13 | 0.66 | 1.11 | 0.67 | 1.35 |
| pi_sum | 1 | 1 | 21.99 | 9.56 | 1 | 299.31 | 1.69 | 1.01 | 1 | 1 | 1 |
| rand_mat_stat | 1.45 | 1.66 | 17.93 | 14.56 | 14.52 | 30.93 | 5.95 | 2.3 | 2.96 | 3.27 | 3.92 |
| rand_mat_mul | 3.48 | 1.02 | 1.14 | 1.57 | 1.12 | 1.12 | 1.3 | 15.07 | 1.42 | 1.16 | 2.36 |

- Julia is designed for parallel computing

# Installation

- Julia only
  - Follow the instructions in https://julialang.org/downloads/

- Juno (Julia + Atom)
  - Recommended setup for the first time users
  - Juno is a powerful integrated development environment for programming in Julia
  - Follow the instructions in https://github.com/JunoLab/uber-juno/blob/master/setup.md

- ECCO Juila
  - On your ecco shell, type:
  - qsub -I -q interactive -l nodes=compute-0-1
  - module load julia
  - julia

# Other References

- Basic tutorials
  - https://julialang.org/learning/

- Documentation
  - https://docs.julialang.org/en/stable/index.html
  - https://en.wikibooks.org/wiki/Introducing_Julia/Getting_started