Numerical and Scientific Computing with Applications
David F. Gleich
CS 314, Purdue

January 17, 2014

Quiz! Survey due! Intro to Matlab

By the end of this class,

- *Know how to launch Julia and Juliabox (and what Jupyter is…)*
- *Evaluate basic expressions in Matlab*
- *Understand basic variables in Matlab.*
- *Work with matrices and vectors in Matlab.*

Next class

More Matlab

G&C - Chapter 2

Next next class

HW Due + Yet more Matlab

G&C – Chapter 2

Numerical computing languages

- Matlab
  - Octave [Open source Matlab clone]
  - Scilab [Open source Matlab clone]
- Numpy/Scipy (Python)
- Mathematica
- R [Statistics focused]
- Sage

These all use the same underlying computational libraries (BLAS/LAPACK, GMP, etc…)

Why Julia vs. Matlab vs. Python?

The class seems somewhat familiar with Matlab.

Julia has similar syntax, but much better programming abstractions.

- Can define multiple functions and glue code in one file!

Compared with Python, Julia has matrix support in the syntax.

- A*x means matrix-multiplication!
- 1-indexed like we do in class.

Julia downsides

- Still under development. Plotting is “weak” but good enough now.
- Lots of packages ☹

Using Julia

1. Use http://juliabox.com 🟢
2. Installing it on your own computer and getting IJulia/Jupyter. 🟦
3. Using Atom/UberJuno ◆◆
4. Using the console interface. ◆