Math 228A: Numerical Solution of Differential Equations

Lin Lin

https://github.com/lin-lin/math228a
(course materials)

https://bcourses.berkeley.edu/courses/1484390
(assignments and announcements)

This class is

- A course on numerical schemes for solving ODEs.
- A math course and therefore contains proofs.. (though we have a diverse audience)
- A course that requires coding (perhaps a significant amount depending on the perspective..)
- A journey that we explore together how to design integrators, how to analyze them, and how to solve them (for potentially large scale, nonlinear problems)

This class is NOT

- A course that involves a lot of “real-world examples”
- Suitable if you have not taken 128A or equivalent.
- A guarantee that I can show up on your qualification exam..

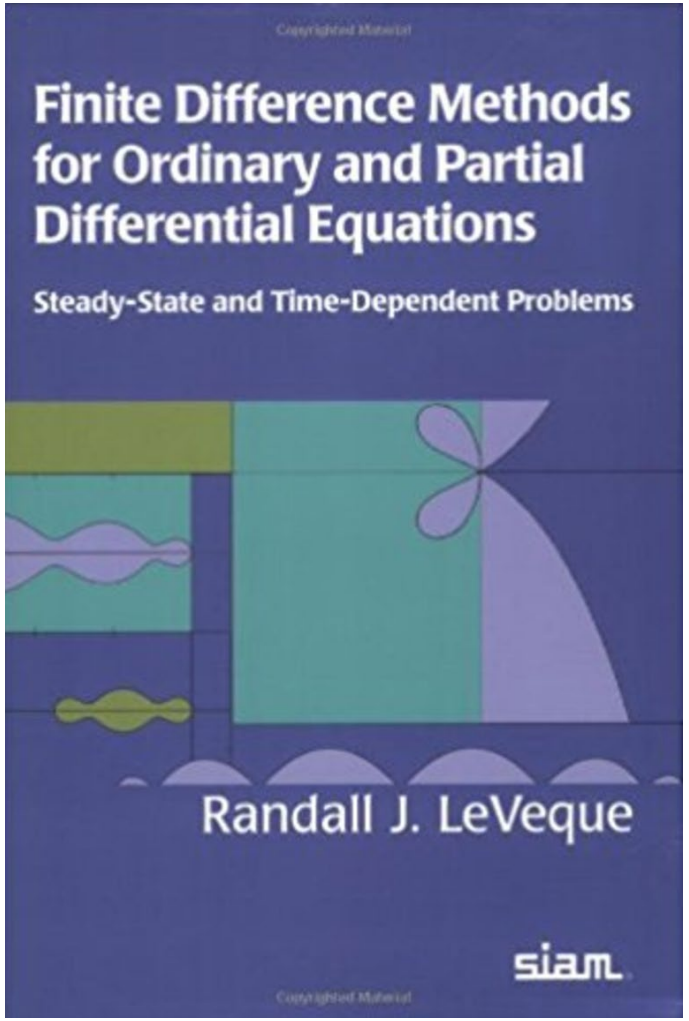
Caution:

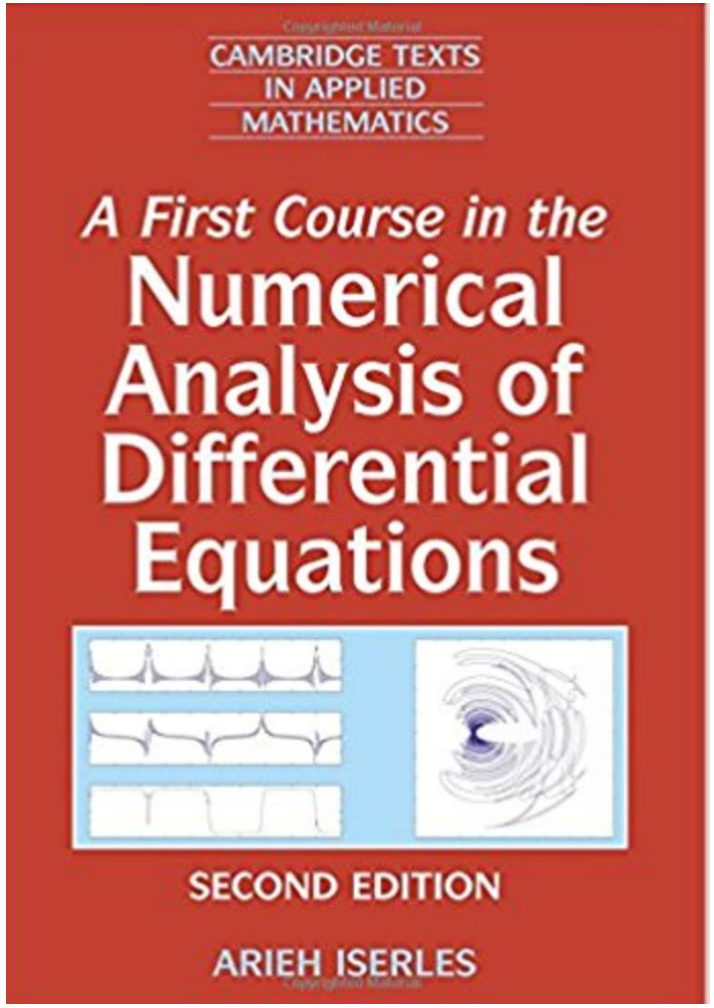
This class involves

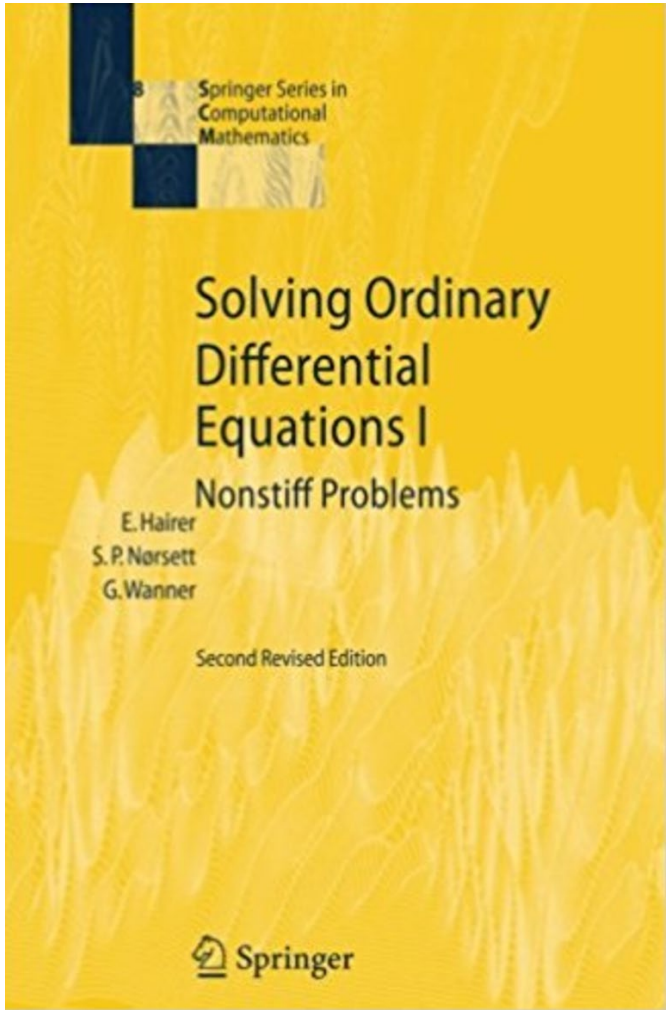
PROOF & Coding

& NOT many real world examples

We do not have a single text book!







Electronic version available for all three books are available via Berkeley Library!
(See bCourses for links)

I heard that “ODE is dead”. Why do I want to spend a semester on ODE?

- Dynamics occurs ubiquitously in scientific computing
- PDEs is not that different
  - You need to handle the ∂_t anyway
  - After spatial discretization, it just becomes a large ODE
- The linear algebra techniques (esp. related to implicit integrators) is largely transferable to PDEs
- Many equations are stiff (i.e. one type of being “difficult”)
- Many ODEs have special structures (e.g. Hamiltonian systems)

Neural network: make ODE cool again?

- Best paper NeurIPS 2018
- We might spend some time on this

Neural Ordinary Differential Equations

Ricky T. Q. Chen*, Yulia Rubanova*, Jesse Bettencourt*, David Duvenaud
University of Toronto, Vector Institute
{rtqichen, rubanova, jessebett, duvenaud}@cs.toronto.edu

Abstract

We introduce a new family of deep neural network models. Instead of specifying a discrete sequence of hidden layers, we parameterize the derivative of the hidden state using a neural network. The output of the network is computed using a black-box differential equation solver. These continuous-depth models have constant memory cost, adapt their evaluation strategy to each input, and can explicitly trade numerical precision for speed. We demonstrate these properties in continuous-depth residual networks and continuous-time latent variable models. We also construct continuous normalizing flows, a generative model that can train by maximum likelihood, without partitioning or ordering the data dimensions. For training, we show how to scalably backpropagate through any ODE solver, without access to its internal operations. This allows end-to-end training of ODEs within larger models.

So there is coding.. which programming language to use?

Compiled language

- C
- C++
- FORTRAN
- …

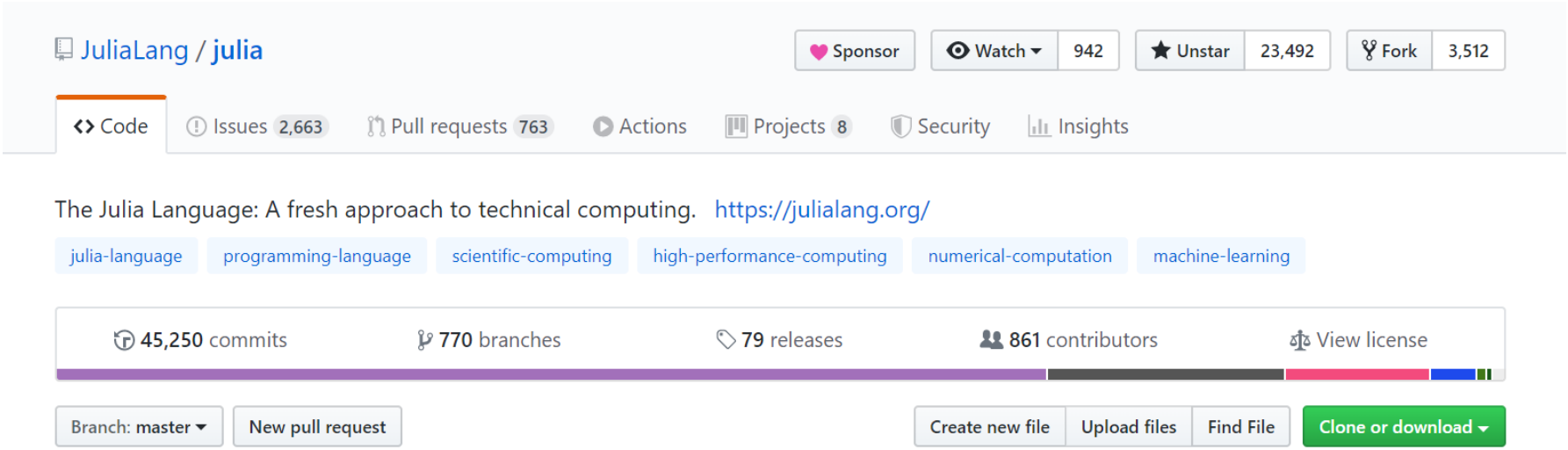
Interpreted language

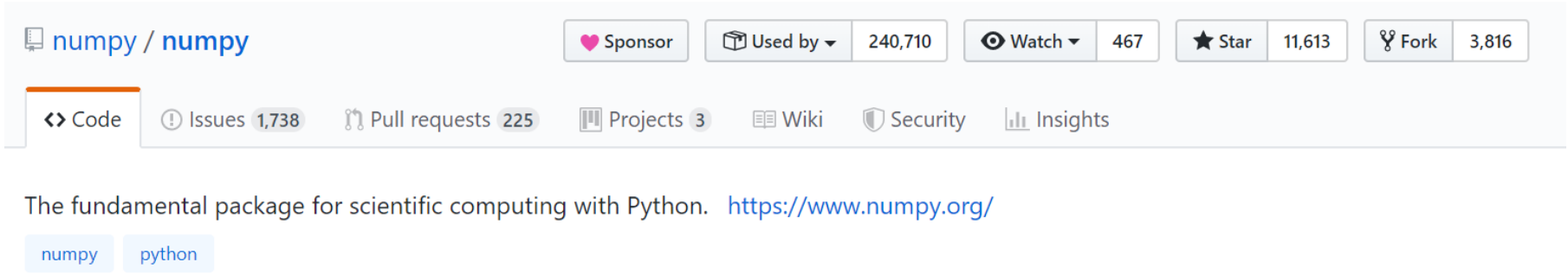
- MATLAB
- Python
- Julia: Promising newcomer
- …

Bezanson et al, Julia: A Fresh Approach to Numerical Computing, *SIAM Rev.*, 59(1), 65–98. 2017

J. H. Wilkinson Prize for Numerical Software, 2019

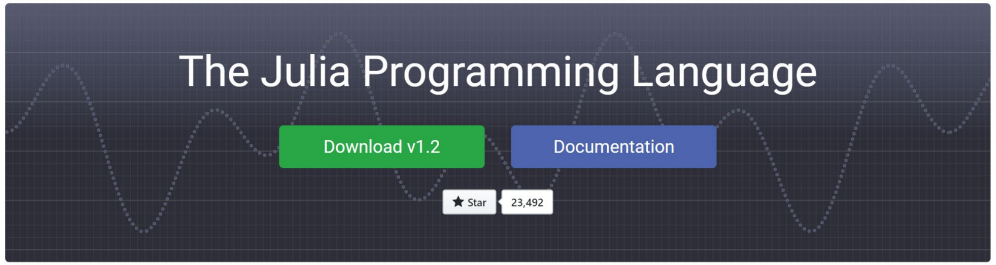
Julia





Web clip: 8/28/2019

Julia: https://julialang.org/ (Use Julia 1.2)



Current stable release: v1.2.0

Checksums for this release are available in both MD5 and SHA256 formats.

Windows (.exe) [help]	32-bit	64-bit	
	Windows 7/Windows Server 2012 users also require Windows Management Framework 3.0 or later. It is recommended that users on these legacy Windows systems install and use a terminal besides *cmd.exe* since the default terminal application has known issues which affect its usability with Julia and other libuv-based cross-platform software.		
macOS 10.8+ (.dmg) [help]		64-bit	
Generic Linux Binaries for x86 [help]	32-bit (GPG)	64-bit (GPG)	
Generic Linux Binaries for ARM [help]	32-bit (ARMv7-a hard float) (GPG)	64-bit (AArch64) (GPG)	
Generic FreeBSD Binaries for x86 [help]		64-bit (GPG)	
Source	Tarball (GPG)	Tarball with dependencies (GPG)	GitHub

Julia 1.2.0 is the only allowed Julia version in this class
Julia 1.2.0 is the only allowed Julia version in this class
Julia 1.2.0 is the only allowed Julia version in this class

Usage of other programming languages (Matlab/Python/..): please consult GSI.