Maintaining and Refactoring Legacy Code

Abraham R. Flores
Washington University in St. Louis

Software Infrastructure for Advanced Nuclear Physics Computing

June 20th 2024

Collaborators

Lorenzo Andreoli (ODU/JLab)
Maria Piarulli (WASHU)
Saori Pastore (WASHU)
Patrick Fasano (ANL)
Bob B. Wiringa (ANL)
Alessandro Lovato (ANL)

Work Supported by the 2021 Early Career Award No. DE-SC0022002 (M.P.)
Work Supported by the FRIB Theory Alliance Award No. DE-SC0013617 (M.P.)
Work Supported by DOE DE-SC0021027 (S.P.)

Why Invest More Resources

The Project Pipeline
(for code based projects)

An idea
(maybe new physics)

Implement the feature
into the code

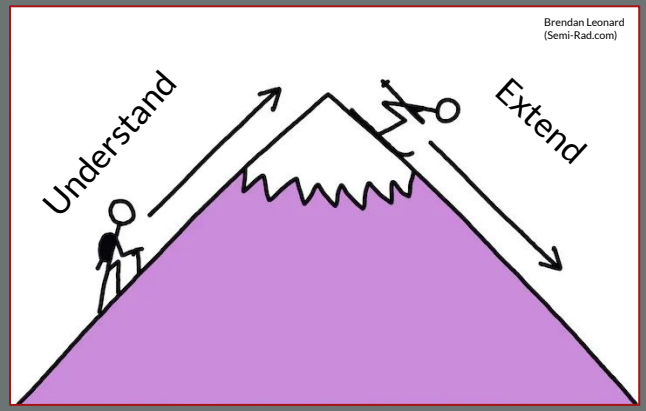
Debug : Test : Run

Analyze Results

Write Manuscript

Publish

Developing Code

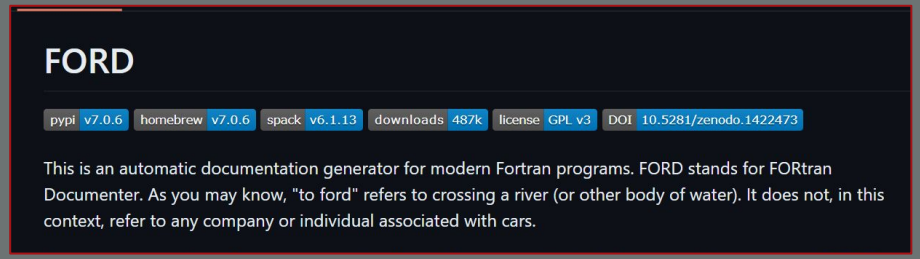


The Minimum

- Version Control
  - Tracks history
  - Merge changes across developers
  - Forks for other users with specific needs
- README
  - A description of what the code can do
  - A simple walkthrough of a using the code
- TESTS
  - How you know you have a bug with your current features

What Is Planning for QMC

- Automated Documentation
  - Comments to labeled wiki
  - Graph of connections
- Resources Dropbox
  - Overview of Language features
  - List of papers and recorded talks
  - Useful notes
- Templates
  - A base framework to extend the code
  - Job submissions
- Walkthroughs
  - compiling, running, and developing the code



How Do We Change Legacy Code?

You must be able to run the code on at least one machine.

1. Identify the Objective
   a. QMC on GPUs
   b. QMC for reactions
2. Generate Approval Tests
   a. Small Scope (output of a function)
   b. Large Scope (Evaluation of an operator at a specific point)
3. Change the code In Small Stages
   a. Undo if tests fail

Expectations for Refactored QMC

- Immense speedup (for A>10) from GPUs
  - Restructured the code for concurrent operations (after first refactoring)
- Generate any wave function bound or unbound at run-time
  - Needed for continuum calculations
- Code was split into smaller chunks that are easier to follow
  - Locality of behaviour
  - Trivial to deploy on any machine
  - Easier to connect physics to the code (better for students)
  - Common and Implicit were removed

Final Comments

VMC refactor is in its final stages

Code base was on the order of 30k lines of code

Difficult part is done for VMC, need to implement features and optimize

Approximate Total Time Refactoring: 6 Months @ 8 hours / day

GFMC refactor is planned be completed by the End of 2025

Thanks for Listening!

Work Supported by the 2021 Early Career Award No. DE-SC0022002 (M.P.)
Work Supported by the FRIB Theory Alliance Award No. DE-SC0013617 (M.P.)
Work Supported by DOE DE-SC0021027 (S.P.)

Extra Slides

Scale of Variational Monte Carlo

(Pieper 2007)	A	Pairs	Spin x Isospin	scale / ^{8}Be	
^{4}H	**4**	**6**	**16 x 4**	**0.002**	
^{5}He	**5**	**10**	**32 x 5**	**0.010**	← **Single Processor**
^{6}Li	6	13	64 x 5	0.036	
^{7}Li	7	21	128 x 14	0.33	
^{8}Be	8	28	256 x 14	1.0	
^{9}Be	9	36	512 x 42	8.7	
^{10}Be	10	43	1024 x 90	52.	
^{11}B	11	55	2048 x 132	200.	
^{12}C	12	66	4096 x 132	530.	← **Supercomputer**

Scale = A*Pairs*Spin*Isospin

*Isospin component has reduced impact.