Managing Academic Software Development

Dr Sam Mangham

Research Software Group

ORCID:0000-0001-7511-5652

University of Southampton

Who Am I

- Senior RSE @ University of Southampton
- Trustee @ Society of Research Software Engineering
- RSE @ Software Sustainability Institute
- Generalist, interdisciplinary RSE, training, community

UNIVERSITY OF Southampton

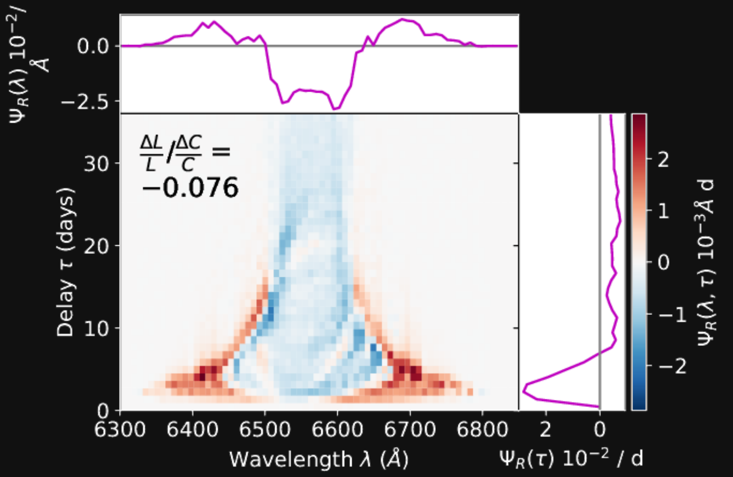
RSG Research Software Group

RSE SOCIETY OF RESEARCH SOFTWARE ENGINEERING

Software Sustainability Institute

Background

- PhD in Astrophysics
  - HPC monte carlo radiation transfer code for supermassive black holes
- Neutronics @ Culham Centre for Fusion Energy
  - HPC monte carlo radiation transfer code for fusion
- Both large legacy HPC codes!



Mangham et al, 2019, ESO/M. Kornmesser

Why?

Enterprise

- Often large teams
- Formal training
- Formal project management frameworks & staff
- Software is the product

Academic

- Small/single teams
- Large numbers of loose collaborators
- Limited training
- Ad-hoc management (by other researchers) or self-management
- Papers are the product

Research Institutes

- Somewhere in-between
- Vary with scale, focus, discipline

Outline

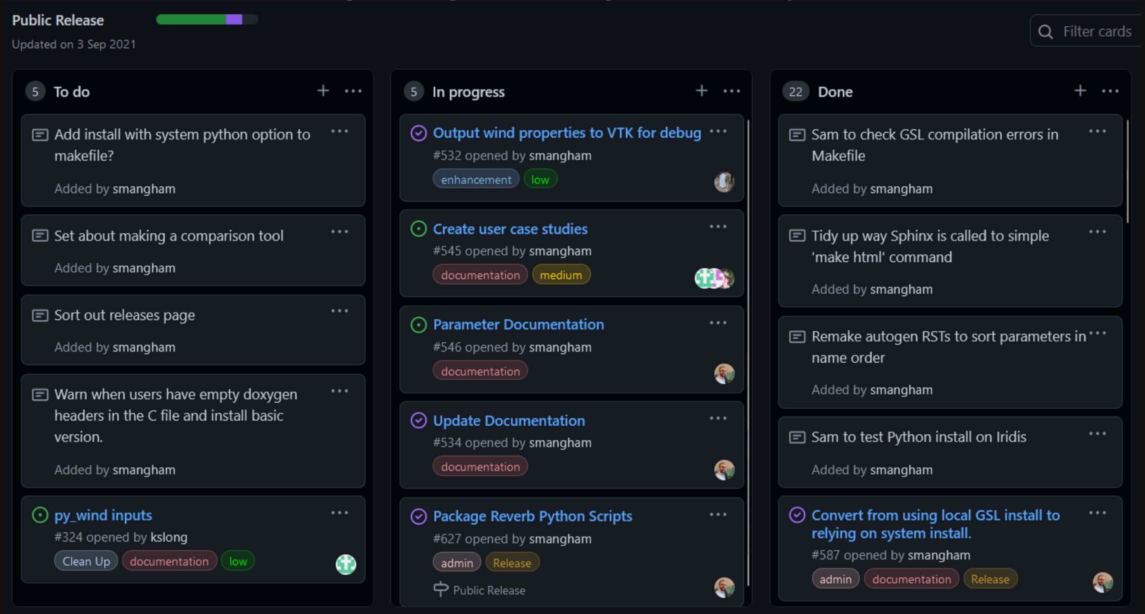
- Development
- Usage
- Publication

Managing Development

Project Boards

> *"Programmers tend to start coding right away. Sometimes this works."* **- Eric Larsen, 2018**

- Break a project into components
- Subdivide as you go!
- Track progress publicly



Public Release

Updated on 3 Sep 2021

5 To do

Add install with system python option to makefile?
Added by smangham

Set about making a comparison tool
Added by smangham

Sort out releases page
Added by smangham

Warn when users have empty doxygen headers in the C file and install basic version.
Added by smangham

py_wind inputs
#324 opened by kslong
Clean Up · documentation · low

5 In progress

Output wind properties to VTK for debug
#532 opened by smangham
enhancement · low

Create user case studies
#545 opened by smangham
documentation · medium

Parameter Documentation
#546 opened by smangham
documentation

Update Documentation
#534 opened by smangham
documentation

Package Reverb Python Scripts
#627 opened by smangham
admin · Release
Public Release

22 Done

Sam to check GSL compilation errors in Makefile
Added by smangham

Tidy up way Sphinx is called to simple 'make html' command
Added by smangham

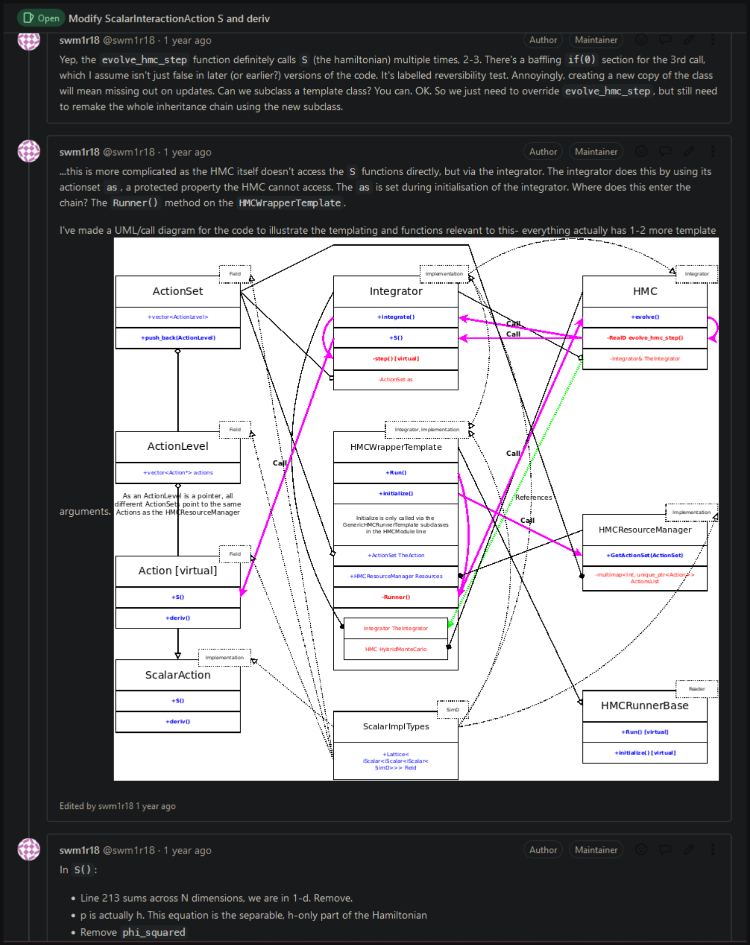
Remake autogen RSTs to sort parameters in name order
Added by smangham

Sam to test Python install on Iridis
Added by smangham

Convert from using local GSL install to relying on system install.
#587 opened by smangham
admin · documentation · Release

Project Boards

- Document process on tasks
  - GitHub/GitLab etc. let you turn issues into lab books
- BUS FACTOR
  - Collaboration
  - Future You is a collaborator
  - Knowledge decays quickly



Open Modify ScalarInteractionAction S and deriv

swm1r18 @swm1r18 · 1 year ago Author Maintainer

Yep, the `evolve_hmc_step` function definitely calls `S` (the hamiltonian) multiple times, 2-3. There's a baffling `if(0)` section for the 3rd call, which I assume isn't just false in later (or earlier?) versions of the code. It's labelled reversibility test. Annoyingly, creating a new copy of the class will mean missing out on updates. Can we subclass a template class? You can. OK. So we just need to override `evolve_hmc_step`, but still need to remake the whole inheritance chain using the new subclass.

swm1r18 @swm1r18 · 1 year ago Author Maintainer

...this is more complicated as the HMC itself doesn't access the `S` functions directly, but via the integrator. The integrator does this by using its actionset `as`, a protected property the HMC cannot access. The `as` is set during initialisation of the integrator. Where does this enter the chain? The `Runner()` method on the `HMCWrapperTemplate`.

I've made a UML/call diagram for the code to illustrate the templating and functions relevant to this- everything actually has 1-2 more template arguments.

Edited by swm1r18 1 year ago

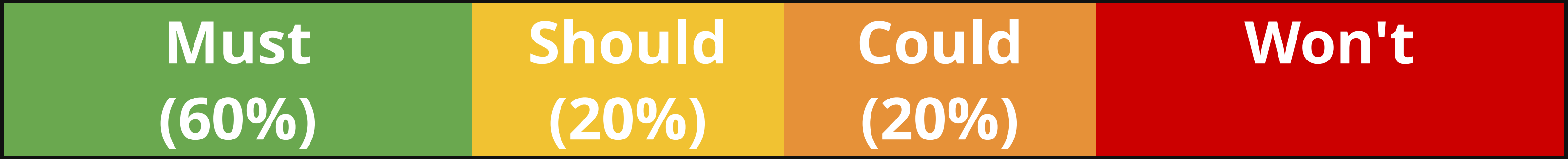
swm1r18 @swm1r18 · 1 year ago Author Maintainer

In `S()`:

- Line 213 sums across N dimensions, we are in 1-d. Remove.
- p is actually h. This equation is the separable, h-only part of the Hamiltonian
- Remove `phi_squared`

Prioritisation

- Time estimates
- MoSCoW



Must (60%)

Should (20%)

Could (20%)

Won't

- Consider and revise!

Prioritisation

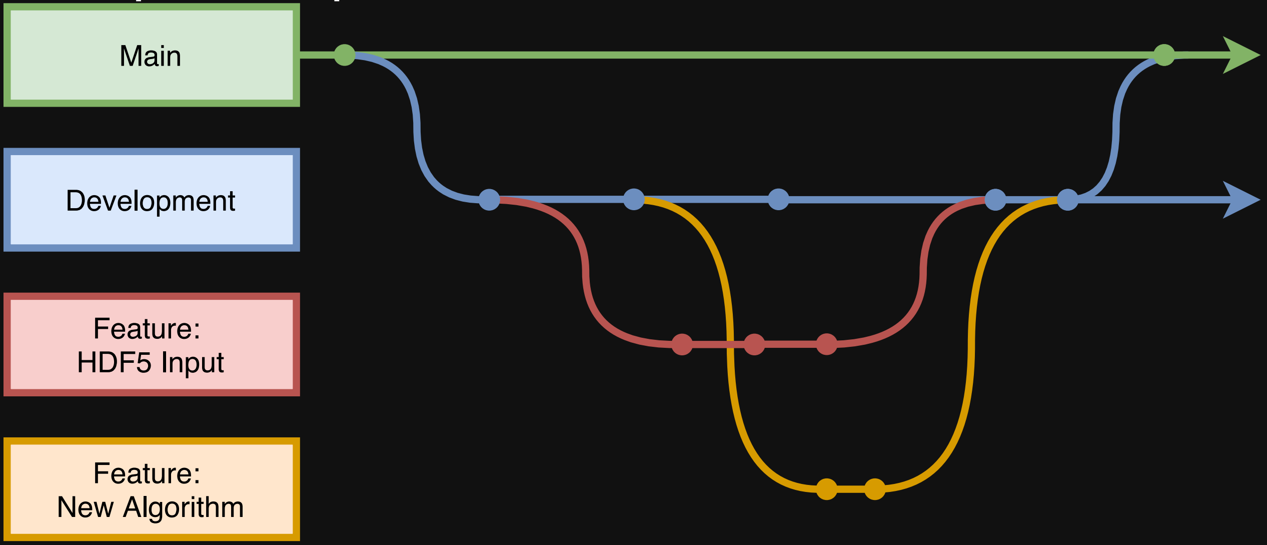
- Won'ts aren't forever
- Typical won'ts
    - Future research avenues
    - Features you don't need right now
    - Bugs that don't stop work
- Acknowledge them publicly
    - Help others plan around you
- Leave time for testing & documentation!

Version Control

- Protection against disaster
- Test and verify changes are *intended*
- Avoid having to rerun entire papers' worth of analysis to avoid version mismatches

Branching Workflows

- New branches for new features
  - Link branches to tasks
  - Easy to parallelise work
  - Easy to switch to working on another feature
- Regularly merge branches back to development!
  - Otherwise each developer ends up with a divergent version
- Review pull requests



Write Sustainable Code

- Proactively avoid technical debt
- Share and collaborate more easily
  - No code worth writing is disposable!
- Write for collaborators and community
- Can't reproduce results if the code isn't sustainable
  - HPC-BP talk on this

Write Readable Code

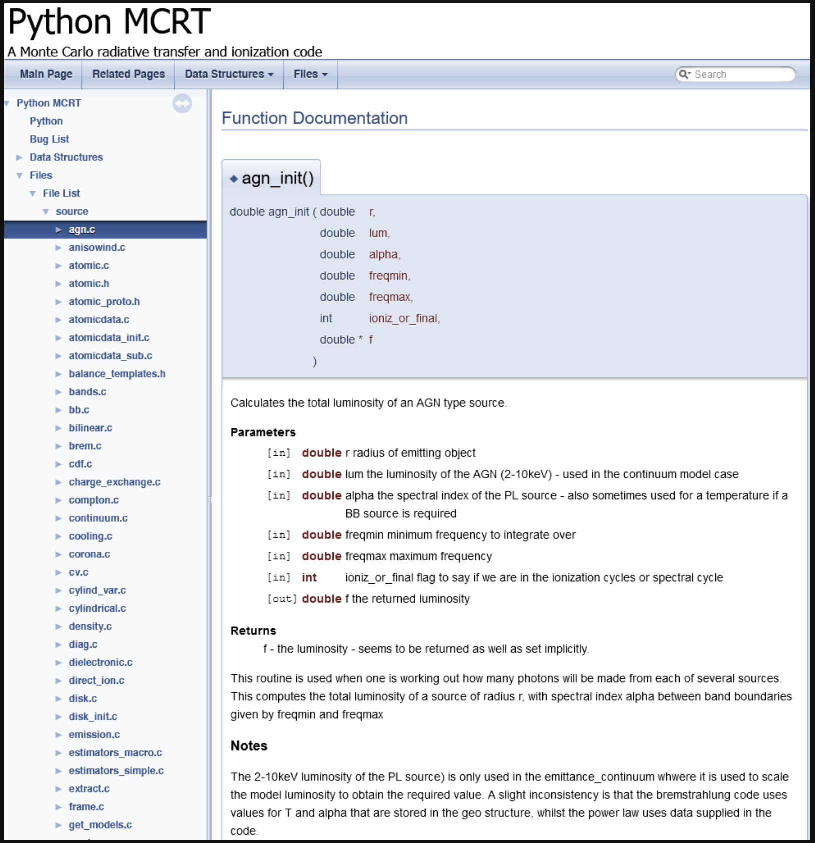
- Easier onboarding
- Follow community standards
    - E.g. PEP 8 for Python
        - pylint, flake8
    - E.g. C++ Core Guidelines, LLVM for C++
        - clang-tidy
    - Pick a style and stick to it!

Write Readable Code

- Descriptive variable names
  - Minimise potential for collision!
  - Not 'c', 'e', 'hb'
- Code completion & IDEs
  - CLion, PyCharm, Visual Studio Code
- Modular code
  - You **will have to** refactor!
  - You can't predict your code's future

Document Your Code

- Bus factor again
- Optionally: Document then design
  - Test-driven development
- Automated tools
  - Sphinx
  - Doxygen
- Automatic hosting
  - ReadTheDocs for Sphinx
  - CodeDocs.xyz for Doxygen
- Call graph generation
- docs-like-code



Python MCRT

A Monte Carlo radiative transfer and ionization code

- Python MCRT
  - Python
  - Bug List
  - Data Structures
  - Files
    - File List
      - source
        - agn.c
        - anisowind.c
        - atomic.c
        - atomic.h
        - atomic_proto.h
        - atomicdata.c
        - atomicdata_init.c
        - atomicdata_sub.c
        - balance_templates.h
        - bands.c
        - bb.c
        - bilinear.c
        - brem.c
        - cdf.c
        - charge_exchange.c
        - compton.c
        - continuum.c
        - cooling.c
        - corona.c
        - cv.c
        - cylind_var.c
        - cylindrical.c
        - density.c
        - diag.c
        - dielectronic.c
        - direct_ion.c
        - disk.c
        - disk_init.c
        - emission.c
        - estimators_macro.c
        - estimators_simple.c
        - extract.c
        - frame.c
        - get_models.c

Function Documentation

◆ agn_init()

```
double agn_init ( double   r,
                  double   lum,
                  double   alpha,
                  double   freqmin,
                  double   freqmax,
                  int      ioniz_or_final,
                  double * f
                )
```

Calculates the total luminosity of an AGN type source.

Parameters

[in]	double	r radius of emitting object
[in]	double	lum the luminosity of the AGN (2-10keV) - used in the continuum model case
[in]	double	alpha the spectral index of the PL source - also sometimes used for a temperature if a BB source is required
[in]	double	freqmin minimum frequency to integrate over
[in]	double	freqmax maximum frequency
[in]	int	ioniz_or_final flag to say if we are in the ionization cycles or spectral cycle
[out]	double	f the returned luminosity

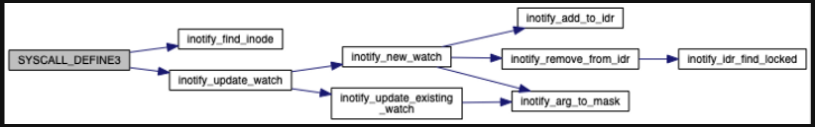
Returns

f - the luminosity - seems to be returned as well as set implicitly.

This routine is used when one is working out how many photons will be made from each of several sources. This computes the total luminosity of a source of radius r, with spectral index alpha between band boundaries given by freqmin and freqmax

Notes

The 2-10keV luminosity of the PL source) is only used in the emittance_continuum whwere it is used to scale the model luminosity to obtain the required value. A slight inconsistency is that the bremstrahlung code uses values for T and alpha that are stored in the geo structure, whilst the power law uses data supplied in the code.



Call graph, Christina Jacob, 2020

Test-Driven Development

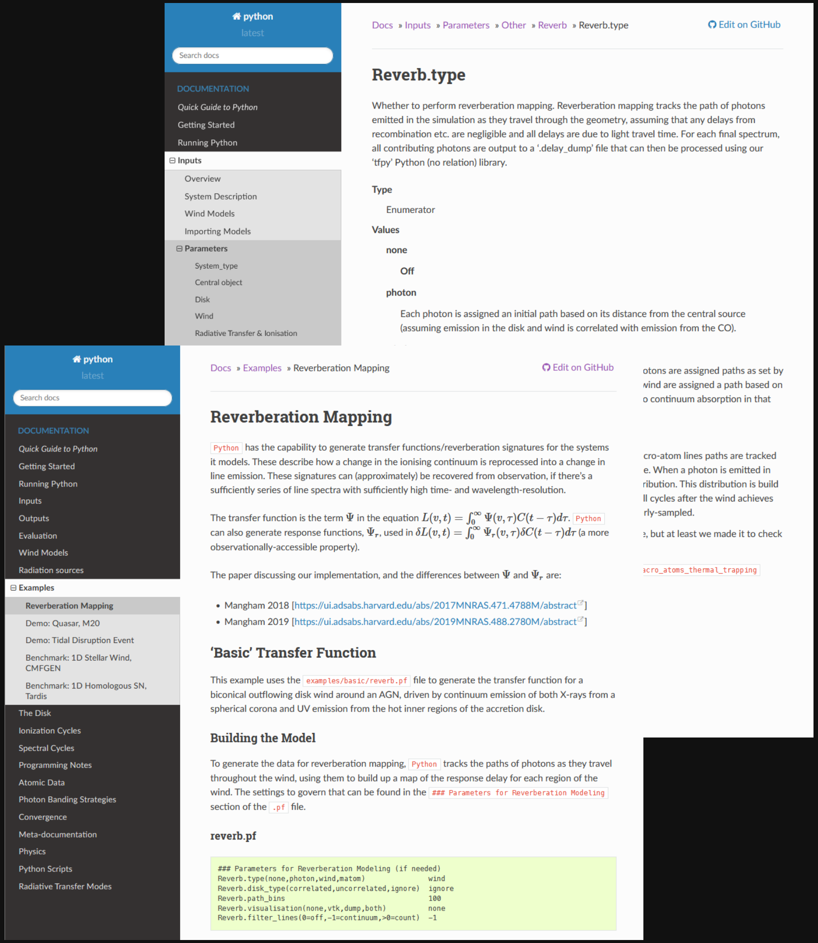
- Continuous Integration
- Many more detailed talks on this!

Questions?

Managing Usage

Public Documentation

- Easy onboarding
- Quick reference for yourself
- Online documentation platforms
  - ReadTheDocs again
  - GitHub Pages
  - GitHub wikis



Reverb.type

Whether to perform reverberation mapping. Reverberation mapping tracks the path of photons emitted in the simulation as they travel through the geometry, assuming that any delays from recombination etc. are negligible and all delays are due to light travel time. For each final spectrum, all contributing photons are output to a '.delay_dump' file that can then be processed using our 'tfpy' Python (no relation) library.

Type

Enumerator

Values

none

Off

photon

Each photon is assigned an initial path based on its distance from the central source (assuming emission in the disk and wind is correlated with emission from the CO).

otons are assigned paths as set by
wind are assigned a path based on
o continuum absorption in that

cro-atom lines paths are tracked
e. When a photon is emitted in
ribution. This distribution is build
ll cycles after the wind achieves
rly-sampled.

e, but at least we made it to check

acro_atoms_thermal_trapping

Reverberation Mapping

`Python` has the capability to generate transfer functions/reverberation signatures for the systems it models. These describe how a change in the ionising continuum is reprocessed into a change in line emission. These signatures can (approximately) be recovered from observation, if there's a sufficiently series of line spectra with sufficiently high time- and wavelength-resolution.

The transfer function is the term Ψ in the equation $L(v,t)=\int_0^\infty \Psi(v,\tau)C(t-\tau)d\tau$. `Python` can also generate response functions, Ψ_r, used in $\delta L(v,t)=\int_0^\infty \Psi_r(v,\tau)\delta C(t-\tau)d\tau$ (a more observationally-accessible property).

The paper discussing our implementation, and the differences between Ψ and Ψ_r are:

- Mangham 2018 [https://ui.adsabs.harvard.edu/abs/2017MNRAS.471.4788M/abstract]
- Mangham 2019 [https://ui.adsabs.harvard.edu/abs/2019MNRAS.488.2780M/abstract]

'Basic' Transfer Function

This example uses the `examples/basic/reverb.pf` file to generate the transfer function for a biconical outflowing disk wind around an AGN, driven by continuum emission of both X-rays from a spherical corona and UV emission from the hot inner regions of the accretion disk.

Building the Model

To generate the data for reverberation mapping, `Python` tracks the paths of photons as they travel throughout the wind, using them to build up a map of the response delay for each region of the wind. The settings to govern that can be found in the `### Parameters for Reverberation Modeling` section of the `.pf` file.

reverb.pf

```
### Parameters for Reverberation Modeling (if needed)
Reverb.type(none,photon,wind,matom)                  wind
Reverb.disk_type(correlated,uncorrelated,ignore)  ignore
Reverb.path_bins                                  100
Reverb.visualisation(none,vtk,dump,both)          none
Reverb.filter_lines(0=off,-1=continuum,>0=count)  -1
```

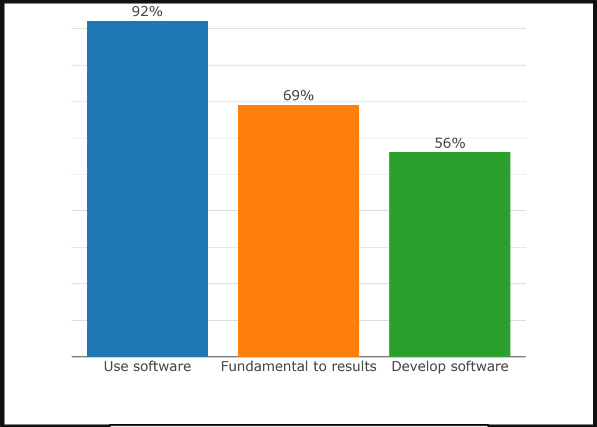
Public Issues

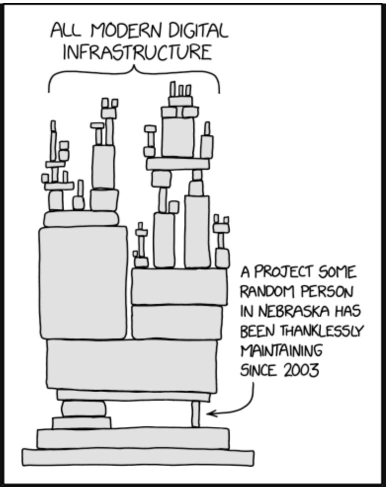
- Facilitate problem solving
  - Searchable if possible!
- Own up to the code's limitations
  - Benefits far outweigh embarassment!
- Issues are a dialogue with your users
  - Even non-issues!
  - Structure it with issue templates

Questions?

Managing Release

Release Your Software




ALL MODERN DIGITAL INFRASTRUCTURE
A PROJECT SOME RANDOM PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

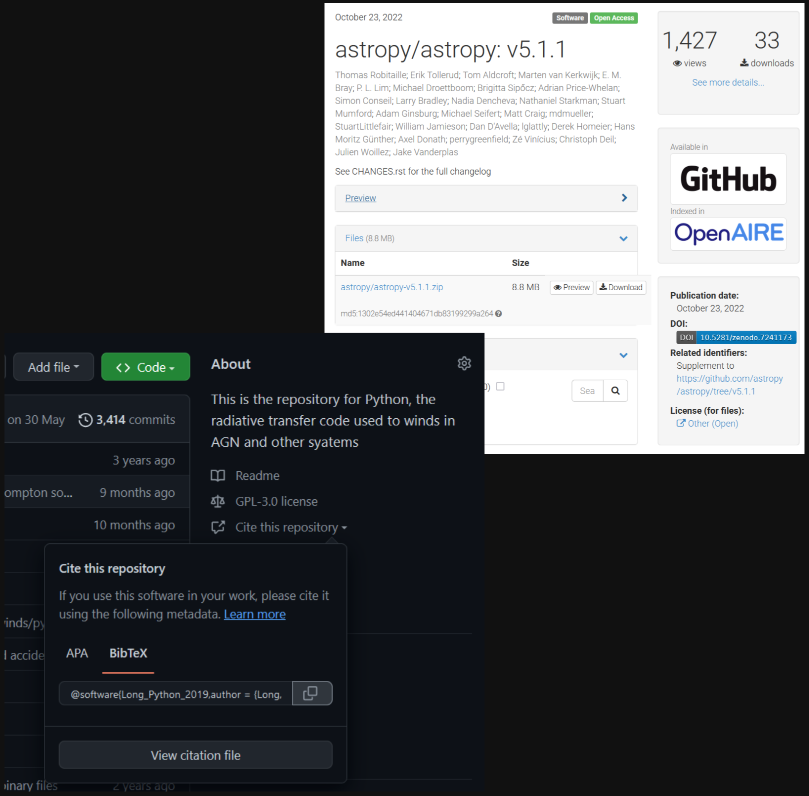
- Majority of research relies on software
- Much is paperware
- Public release is required for reproducibility!

S.J.Hettrick et al, Software in Research Survey, 2014; DOI:10.5281/zenodo.1183562

Randall Munroe, XKCD - Dependency

Structured Releases

- GitHub Releases
- Citation.CFF
- Zenodo
  - Provides citeable DOIs
- Include *all* info
  - Library versions
  - Compiler versions
  - Compiler flags



October 23, 2022

Software Open Access

astropy/astropy: v5.1.1

Thomas Robitaille; Erik Tollerud; Tom Aldcroft; Marten van Kerkwijk; E. M. Bray; P. L. Lim; Michael Droettboom; Brigitta Sipőcz; Adrian Price-Whelan; Simon Conseil; Larry Bradley; Nadia Dencheva; Nathaniel Starkman; Stuart Mumford; Adam Ginsburg; Michael Seifert; Matt Craig; mdmueller; StuartLittlefair; William Jamieson; Dan D'Avella; Iglattly; Derek Homeier; Hans Moritz Günther; Axel Donath; perrygreenfield; Zé Vinícius; Christoph Deil; Julien Woillez; Jake Vanderplas

See CHANGES.rst for the full changelog

Preview

Files (8.8 MB)

Name	Size	
astropy/astropy-v5.1.1.zip md5:1302e54ed441404671db83199299a264	8.8 MB	Preview Download

1,427 views

33 downloads

See more details...

Available in GitHub

Indexed in OpenAIRE

Publication date: October 23, 2022

DOI: DOI 10.5281/zenodo.7241173

Related identifiers: Supplement to https://github.com/astropy/astropy/tree/v5.1.1

License (for files): Other (Open)

About

This is the repository for Python, the radiative transfer code used to winds in AGN and other syatems

Readme

GPL-3.0 license

Cite this repository ▾

Cite this repository

If you use this software in your work, please cite it using the following metadata. Learn more

APA BibTeX

```
@software{Long_Python_2019,author = {Long,
```

View citation file

Software Licenses

- Previous HPC-BP talk
- No License
  - Automatically copyrighted
  - No rights for others to do *anything*
- Open-Source
  - Copyleft (e.g. GPL3)
  - Permissive (e.g. MIT)
- Proprietary License
  - Lawyers are expensive
- choosealicense

Thank you for your time

⭐ s.mangham@soton.ac.uk / rsg.soton.ac.uk ⭐

s.mangham@society-rse.org / society-rse.org

s.mangham@software.ac.uk / software.ac.uk

Advertisements

RSE SOCIETY OF RESEARCH SOFTWARE ENGINEERING

bit.ly/SocRSE-Mentoring-2022

RSG Research Software Group

rsg.southampton.ac.uk/jobs

Any questions?