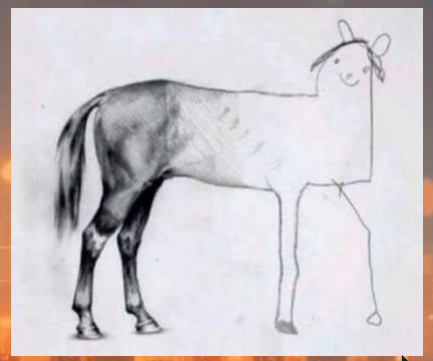
Best practices

(how to protect the future You,
who is always grumpier)



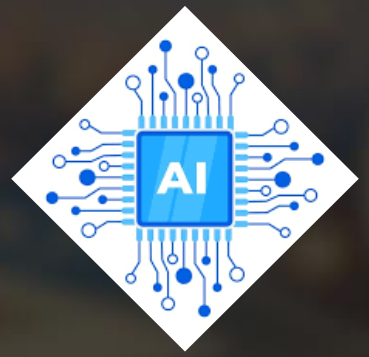
Slide number

But isn’t it obvious?

There are good reasons to care

- **Reproducibility:** future you / students / sceptical R2 rerun the same analysis and get the same results
- **Correctness:** catch bugs early, not when R2 does it
- **Maintainability:** update without global collapse
- **Collaboration:** reduce friction for students / collaborators / R2
- **Longevity:** code outlives memory (and PhD lifetimes)

- Version control
- Testing
- Automated code quality
- Types & Explicit Interfaces
- Documentation



Version control as Scientific Memory
(just use git)

- ❑Every change to code = a change to the scientific argument
- ❑Small, meaningful commits (what changed + why)
- ❑Feature branches + pull requests for non-trivial work
- ❑Tag the version used for published results
- ❑Track configuration files under version control

First time contributing to astropy/astropy?

Dismiss

If you know how to fix an issue, consider opening a pull request for it.
You can read this repository's contributing guidelines to learn how to open a good pull request.

Filters | is:pr is:open | Labels 116 | Milestones 3 | New pull request

150 Open ✓ 12,478 Closed | Author | Label | Projects | Milestones | Reviews | Assignee | Sort

- **BUG: Fix broadcasting bug in uniform() for ndim >= 2 inputs** uncertainty — 1
  #19308 opened 5 hours ago by yaochengchen • Review required • 1 task
- **coords: add comprehensive docstring to CIRS frame (fixes #18981)** coordinates Docs no-changelog-entry-needed — 1 — 5
  #19307 opened yesterday by Omiii-215 • Review required • 1 task • v8.0.0
- **coordinates: Fix binary I/O bug in TransformGraph.to_dot_graph** Bug Close? coordinates — 6
  #19302 opened last week by yaochengchen • Review required • v8.0.0
- **DOC: simplify instructions for generating new hash libraries** Affects-dev Docs no-changelog-entry-needed testing — 4
  #19297 opened last week by neutrinoceros • Review required • 1 task • v8.0.0
- **Better detect invalid extra bytes in FITS files** Bug io.fits Performance — 12
  #19296 opened last week by maxnoe • Review required • 1 task • v8.0.0
- **Fix Tabular models incorrectly reporting uses_quantity=True when unitless (#19293)** Bug modeling — 1 — 1
  #19295 opened last week by coder-jayp • Review required • v8.0.0
- **Support for fsspec filesystem configuration via fits.open** Bug Docs Extra CI io.fits utils — 1 — 31
  #19294 opened last week by bmorris3 • Approved • 1 task • v8.0.0

Testing protects against silent wrongness

- ❑ Unit tests for core logic and numerical routines
- ❑ Integration test for the full analysis pipeline on small data
- ❑ Analysis / simulation tests:
  - ❑ Regression for key scientific outputs (e.g. just optimised my integrator? It better not change the shape of my iron line!)
  - ❑ physical invariants (e.g. symmetry, conservation, limits)
  - ❑ assumption recovery (e.g. those fluxes should probably be non-negative)

Language notes

Python: pytest | C++: Catch2/GTest | Rust: built-in | Julia: Pkg.test | Fortran: pFUnit or black-box tests

Files

main

Go to file t

- .github
- docs
- equinox
- imgs
- tests
- .editorconfig
- .gitignore
- .pre-commit-config.yaml
- CONTRIBUTING.md
- LICENSE
- README.md
- mkdocs.yml
- pyproject.toml

equinox / tests /

Add file

nstarman feat(nn): ScanOverMLP (#1179) 48453cc · last week History

Name	Last commit message	Last commit date
..		
__init__.py	Tidied helper into a relative import (#133)	4 years ago
conftest.py	Enable sharding tests	5 months ago
helpers.py	Make tests pass on nvidia gpu (#1146)	3 months ago
test_abstract.py	New cleaner Module implementation.	7 months ago
test_ad.py	Run pyupgrade 3.10	11 months ago
test_caches.py	Switch to ruff-format and ruff for ipynb	3 years ago
test_callback.py	Switch to ruff-format and ruff for ipynb	3 years ago
test_checkpoint.py	Switch to ruff-format and ruff for ipynb	3 years ago
test_closure_to_pytree.py	Now using strict rank and dtype promotions when testing.	3 years ago
test_debug.py	Support complex dtypes in networks (#765)	2 years ago
test_enum.py	Fixing representations of vmapped enumerations (#1102)	5 months ago
test_errors.py	Fixed errors after non-errors. (#1156)	last month
test_eval_shape.py	Switch to ruff-format and ruff for ipynb	3 years ago

Automated code quality

Continuous Integration = A Robot Co-Author

- ❑Prevents broken code from reaching main (or software daylight)
- ❑Every push runs in a clean environment:
  - ❑dependencies installed from scratch
  - ❑Builds the project (if needed)
- ❑Runs tests, linters, type checks (formatting time is better spent over coffee)

Files

master

Go to file t

.devcontainer
.github
workflows
LabelCheck.yml
PrAssignee.yml
Typos.yml
Whitespace.yml
backport-label-audit.yml
cffconvert.yml
CODEOWNERS
dependabot.yml
.vscode
Compiler
JuliaLowering
JuliaSyntax
base
cli
contrib

julia / .github / workflows / PrAssignee.yml

View Runs

dependabot[bot] Bump actions/github-script from 7.0.1 to 8.0.0 (#59713) ba9f47b · 5 months ago History

Code Blame 210 lines (189 loc) · 9.96 KB · Raw

```yaml
name: PR Assignee
on:
  # Important security note: Do NOT use `actions/checkout`
  # or any other method for checking out the pull request's source code.
  # This is because the pull request's source code is untrusted, but the
  # GITHUB_TOKEN has write permissions (because of the `on: pull_request_target` event).
  #
  # Quoting from the GitHub Docs:
  # > For workflows that are triggered by the pull_request_target event, the GITHUB_TOKEN is granted
  # > read/write repository permission unless the permissions key is specified and the workflow can access secrets,
  # > even when it is triggered from a fork.
  # >
  # > Although the workflow runs in the context of the base of the pull request,
  # > you should make sure that you do not check out, build, or run untrusted code from the pull request with this event.
  #
  # Source: https://docs.github.com/en/actions/writing-workflows/choosing-when-your-workflow-runs/events-that-trigger-workflows#pull_request
  #
  # See also: https://securitylab.github.com/resources/github-actions-preventing-pwn-requests/
  pull_request_target:
    types: [opened, reopened, ready_for_review]

# Permissions for the `GITHUB_TOKEN`:
permissions:
  pull-requests: write # Needed in order to assign a user as the PR assignee

jobs:
  pr-assignee:
    runs-on: ubuntu-latest
```

Types Encode Assumptions

- ❑Clear function signatures reduce ambiguity
- ❑Encode units, shapes, and expected structures
- ❑Let the compiler/type checker catch mistakes early

Language notes

Python: type annotations + mypy/pyright

C++/Rust: use of strong types / don’t make everything a ‘double’

Julia: ensure type stability

Fortran: implicit none, explicit interfaces

Files
main
Go to file t
tests
xtactic
__init__.py
_energy.py
_spectra.py
_tables.py
_workflow.py
.gitignore
.pre-commit-config.yaml
LICENSE
README.md
pyproject.toml

xtactic / xtactic / _energy.py

joanna-pk added sliding window averaging for xspec-like rebinning of multiplica... 344feaf · last week History

Code Blame 333 lines (264 loc) · 10.9 KB Raw

```python
# Energy-grid related utility functions
from pathlib import Path

import numpy as np
from astropy.io import fits
from jaxtyping import Array, Float


def energy_grid_from_table_model(table_model_path: Path) -> Float[Array, " N"]:
    """Extract energy grid from a table model fits file
    ENERG_LO and ENERG_HI columns"""

    with fits.open(table_model_path) as hdulist:
        table = hdulist[2].data
        energ_lo = np.asarray(table["ENERG_LO"])
        energ_hi = np.asarray(table["ENERG_HI"])

    return np.append(energ_lo, energ_hi[-1])


def _get_widths_and_centers(
    edges: Float[Array, " N"],
) -> tuple[Float[Array, " N-1"], Float[Array, " N-1"]]:
    """Get the widths and centers of energy bins from their edges.
    Note: there is no unit conversion in the process.

    Parameters
```

Documentation

- ❑ Docstrings: what, inputs, outputs, units, assumptions
- ❑ Comments explain why, not what
- ❑ README: how to run and reproduce results
- ❑ Document data sources and preprocessing
- ❑ (ML / Stats / anything requiring randomisation): record seeds, parameters, and known limitations

cp2k

Public

Watch 36 Fork 450 Star 1.1k

master 29 Branches 25 Tags Go to file t Add file Code

Growl1234 and mkrack Along with minor fixes to make_cp2k.sh 0ada215 · 25 minutes ago 23,127 Commits

.fmf	fedora: Optimize builds (#3268)	2 years ago
benchmarks	ELPA: turn section PRINT_ELPA into a lone keyword ELPA...	last month
ci	Update CI docker files and adapt make_cp2k script	last week
cmake	Add DBCSR cmake configuration check. Apply only to DB...	3 days ago
data	Fix typos with Cs in BASIS_ccGRB_UZH (#4698)	last month
docs	Add DBCSR cmake configuration check. Apply only to DB...	3 days ago
src	fix for partial occupation numbers from tblite (#4857)	17 hours ago
tests	fix for partial occupation numbers from tblite (#4857)	17 hours ago
tools	Along with minor fixes to make_cp2k.sh	25 minutes ago
.dockerignore	Docker: Migrate coverage test to CMake	7 months ago
.git-blame-ignore-revs	Add 2b6ea5 to .git-blame-ignore-revs	3 years ago
.gitattributes	add Github Action workflow to create release tarball	7 years ago
.gitignore	DLA-Future Documentation	2 years ago

About

Quantum chemistry and solid state physics software package

www.cp2k.org

simulation hpc quantum-chemistry material-science

Readme
GPL-2.0 license
Activity
Custom properties
1.1k stars
36 watching
450 forks
Report repository

Releases 25

CP2K v2026.1 Latest
on Jan 6

+ 24 releases

☑ One Good Tutorial

Are you setting up documentation for new piece of scientific software? If so, make sure to include **One Good Tutorial**.

But you don't need much more than that. According to the **One Good Tutorial software documentation checklist** (version 1), your documentation is good enough if it contains these nine major items:

- [ ] 🗺️ **Synopsis:** 1–3 sentence summary of your project
- [ ] 📄 **Tutorial:** ✨ Show people what your software can do! ✨
- [ ] 👩‍💼 **Contact Information:** How to ask a human about your software
- [ ] 🚀 **Install Instructions:** How to install your software
- [ ] 📜 **Citation Instructions:** How to cite your software
- [ ] 🙌 **Contribution Statement:** How users can contribute to your project
- [ ] 📚 **Reference Material:** Precise specifications of APIs, etc.
- [ ] ⚖️ **Licensing Statement:** The legal status of your code
- [ ] 🙏 **Acknowledgments:** Credit your funders

onegoodtutorial.org

LLMs are great*

DOs

- ❑ boilerplate and scaffolding: tests / setting up CI / generating docstrings (when type annotations and comments are in place!) / refactoring repetitive code
- ❑ Explain someone else's janky code
- ❑ Improve clarity:
  - ❑ rewrite confusing code
  - ❑ add type hints
- ❑ Keep prompts precise and constrained
  - ❑ include function signature
  - ❑ specify shapes/units
  - ❑ specify performance constraints
- ❑ Treat generated code as a draft, not truth

DON'Ts

- ❑ Don't paste unverified physics into production code
- ❑ Don't assume correct units or conventions
- ❑ Don't let LLMs choose scientific defaults silently
- ❑ Don't skip tests because "it looks right"