VECTOR

A new SymPy backend for Vector: uniting experimental and theoretical physicists

Jim Pivarski[1], Saransh Chopra[1 2]

[1] Princeton University [2] University of Delhi

Support for this work was provided by NSF cooperative agreements OAC-1836650 and PHY-2323298 (IRIS-HEP).

Introduction

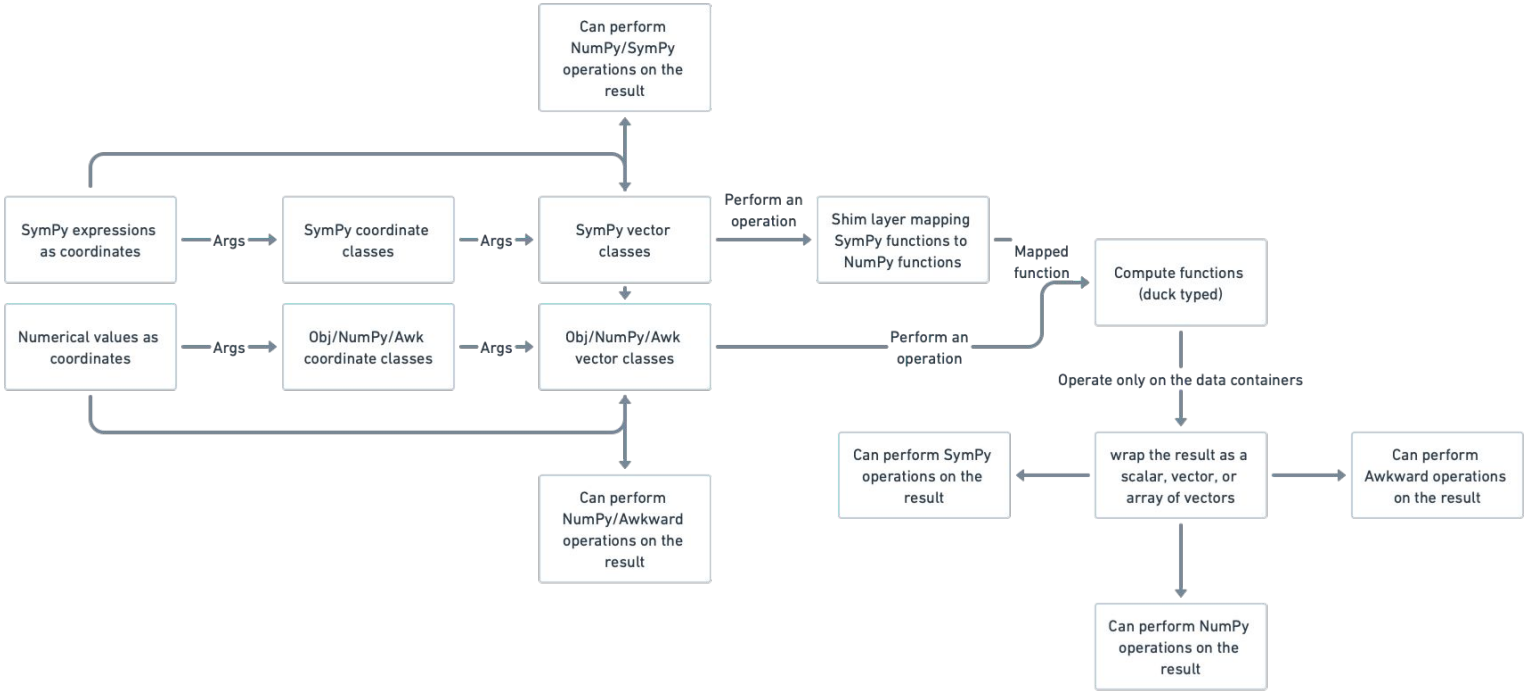
- Vector is a Python 3.8+ library (Python 3.6 and 3.7 supported till v0.9.0 and v1.0.0, respectively) for 2D, 3D, and Lorentz vectors, especially arrays of vectors, to solve common physics problems in a NumPy-like way.
- Vector can perform numerical computation using pure Python, NumPy, and Awkward Arrays; hence, it is used by experimental physicists in their analysis pipelines.
- Vector also supports Dask, Jax, and Numba on Awkward Array of vectors

Motivation

- Vector (and most of the Scikit-HEP ecosystem) is at the moment meant to be used exclusively by experimental physicists.
- Vector's compute functions were written to operate only on data containers, which is tested using uncompyle6 in the CI.
- Adding SymPy as a new backend would kill 2 birds with 1 stone -
  - along with experimental physicists using vector for numerical computations, the SymPy backend will enable theoretical physicists to utilize the library for symbolic computations.
  - allow the developers to get rid of uncompyle6 when Python 3.8 reaches EOL; SymPy tests will ensure that the compute functions run only on data containers

Example

Working


Can perform NumPy/SymPy operations on the result
SymPy expressions as coordinates
Args
SymPy coordinate classes
Args
SymPy vector classes
Perform an operation
Shim layer mapping SymPy functions to NumPy functions
Mapped function
Compute functions (duck typed)
Numerical values as coordinates
Args
Obj/NumPy/Awk coordinate classes
Args
Obj/NumPy/Awk vector classes
Perform an operation
Operate only on the data containers
Can perform SymPy operations on the result
wrap the result as a scalar, vector, or array of vectors
Can perform Awkward operations on the result
Can perform NumPy/Awkward operations on the result
Can perform NumPy operations on the result

Caveats

Operations on SymPy vectors are only 100% compatible with numeric vectors (Python, NumPy, and Awkward backends) if the vectors are positive time-like, that is, if t**2 > x**2 + y**2 + z**2.

The space-like and negative time-like cases have different sign conventions; hence, to make SymPy's simplification to work, these sign conventions are ignored in the shim layer.

Other major developments

- NumPy 2.0 support
- dask-awkward support in vector constructors
- Jax support through Awkward Arrays
- Coffea is switching to vector internally
- Better sub-classing support for Awkward mixins
- A uniform and a strict promotion/demotion scheme for geometric coordinates
- Momentum support for transformation methods
- …
- Multiple other QOL fixes and features (development spearheaded by physicists' request)

Vector had 5 new releases this year - v1.2, v1.3, v1.3.1, v1.4, v1.4.1 (changelog).

Cite vector!

```
@software{Schreiner_vector,
author = {Schreiner, Henry and Pivarski, Jim and Chopra, Saransh},
doi = {10.5281/zenodo.5942082},
license = {BSD-3-Clause},
title = {{vector}},
url = {https://github.com/scikit-hep/vector}
}
```

Thank you!