Compliance sheet — star-quaternion

Everything a supplier qualification, a legal review or an export check usually asks about star-quaternion 0.1.0, in one document.

1. Identity

Document	STAR-PKG-STAR-QUATERNION-EN · v2.3 · 2026-11-14
Package	star-quaternion
Version	0.1.0
Capability	star.attitude.quaternion
Maturity	Released & Verified (RELEASED)
Python	>=3.9
Source commit	dc09f61d
Source	https://github.com/Andrea07072000/S.T.A.R.-/tree/main/packages/star_quaternion

2. Licence and obligations

Apache License 2.0. You may use, copy, modify and distribute it, also commercially, and you receive a patent licence from the contributors. When you redistribute it: keep the LICENSE file and the copyright notices (and a NOTICE file, if a future version adds one), and state in modified files that you changed them. You may not use the S.T.A.R. name to endorse your product.

3. What it needs at run time

Nothing beyond the Python standard library: it works offline.

4. Integrity

SHA-256 fingerprints of the files published on this website. Check a downloaded file with:

```
python -c "import
hashlib,sys;print(hashlib.sha256(open(sys.argv[1],'rb').read()).hexdigest())" star_quaternion-
0.1.0-py3-none-any.whl
```

FILE	SHA-256
star_quaternion-0.1.0-py3-none-any.whl	c66e06142eacc3ad6ba850d472d43bda045b63fc3a9a8f337447f2d3ec0b6910
star_quaternion.py	df2c68812a8624679517765a7f2647c84156f28bd6ff20137699a75be2dcdc78
star_quaternion-0.1.0-source.zip	f7f47443a67aa906716f65caad256d4b785659065251fff643af71acae9df99e

5. What it does, and its limits

Unit quaternion algebra, rotation matrices and axis-angle with a fixed, tested convention (scalar first, Hamilton, active rotation); agrees with NAIF SPICE and SciPy to 9e-16 and refuses non-unit quaternions and non-rotation matrices instead of normalising them silently.

No Euler angles (twelve conventions: a separate, explicit interface is needed), no interpolation, no attitude determination. Frames are the caller's: the functions rotate vectors, they do not know what the frames are called.

- Convention: quaternion (w, x, y, z) with the scalar first, Hamilton product, ACTIVE rotation; `to_dcm(q)` is the matrix R with `rotate(q, v) = R v`; `multiply(a, b)` applies b first, then a. This is the convention of NAIF SPICE (`q2m`, `m2q`, `qxq`); SciPy stores the scalar last.
- `normalize`, `conjugate`, `multiply`, `rotate`, `to_dcm`, `from_dcm`, `from_axis_angle`, `to_axis_angle`, `angle_between_deg`. `from_dcm` uses the largest-component method (exact on half-turns, where the trace method divides by zero) and returns the quaternion with w >= 0. Angles are computed with atan2: a rotation of 1e-12 degrees keeps its value.
- Measured against SPICE and SciPy on 411 cases (random rotations, identity, half-turns, 1e-9 degree rotations): matrix, matrix-to-quaternion, product, rotated vector and angle agree within 9e-16.
- Every function returns finite floats or raises `ValueError`: wrong shape, non-numeric, boolean, NaN, infinite or |value| >= 1e150 input; a zero quaternion or axis; a quaternion whose norm is further than 1e-6 from 1 where a unit quaternion is required (it is refused, not silently normalised); a matrix that is not orthonormal to 1e-9 or is a reflection.

6. Export, use and certification

The package is published and freely available. That usually takes software out of export-control lists, but not every use of it: read the page on space activities and export control before using it in a controlled programme. It is not certified or qualified for flight; the evidence describes what was tested and how.

7. Citation and bill of materials

```
@software{cavazzini_star_quaternion_0_1_0,
  author  = {Cavazzini, Andrea},
  title   = {{star-quaternion}: Rotations with one stated convention},
  version = {0.1.0},
  year    = {2026},
  url     = {https://andrea07072000.github.io/S.T.A.R.-/software/star_quaternion/},
  note    = {Apache-2.0. Source: https://github.com/Andrea07072000/S.T.A.R.-, commit dc09f61d}
}
```

CITATION.cff · BibTeX · SBOM CycloneDX 1.5 · SBOM SPDX 2.3

8. Warranty and liability

Provided "as is", without warranties or conditions of any kind (Apache License 2.0, sections 7 and 8), to the extent the law allows. This never excludes liability that the law does not allow to exclude.

Related: space activities and export control · acceptable use · evidence and limits · licence texts