Compliance sheet — star-polyfit

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

1. Identity

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

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_polyfit-
0.1.0-py3-none-any.whl
```

FILE	SHA-256
star_polyfit-0.1.0-py3-none-any.whl	923f41d89a6a9b3ca518b4a8445acf05454 0994623add6ab24e1eb564067a425
star_polyfit.py	06546fdd6df439b32b38da7c47bacd00216 407af317f25f0d0e1784605f0f566
star_polyfit-0.1.0-source.zip	5f6e6e1fb9b23bf1a1d5ebb4e81cb6e6b8b ba8d4eee8fe05be44aac1178ff47c

5. What it does, and its limits

Least-squares polynomial of degree 0 to 10, weighted or not: the normal equations are formed and solved in exact rational arithmetic and each coefficient is rounded once, so conditioning costs no digit; exact evaluation and exact residual sum of squares; NIST Wampler1 reproduced exactly; every coefficient of 161 problems equal to SymPy's exact solution.

Exact coefficients are not a good model: a high-degree fit of noisy data oscillates, and the exact coefficients of a badly conditioned problem change a lot when the data change a little; the float nearest to each coefficient is still only a float, and evaluating the fitted polynomial in ordinary float arithmetic far from the origin can lose every digit (use `polyval`, or centre the data). No standard errors, no covariance matrix, no orthogonal polynomials, no automatic choice of the degree, no robust fitting. Exact arithmetic is slow: about a second for 2000 points at degree 10 on a laptop. "Nearest float" refers to the floats passed in, not to the decimal numbers you may have meant.

- `polyfit(xs, ys, degree, weights=None)` returns the coefficients, lowest power first, of the polynomial that minimises the sum of w (y - p(x))². `polyval(coefficients, x)` evaluates a polynomial; `residual_sum(xs, ys, coefficients, weights=None)` returns the weighted sum of squared residuals.
- The normal equations are formed and solved in exact rational arithmetic on the floats given, and each result is rounded once: the conditioning of the problem costs no digit. `polyval` and `residual_sum` are exact sums rounded once as well.
- Measured on 161 problems (120 well conditioned, 40 badly conditioned, and the NIST dataset Wampler1): every coefficient equals the exact rational solution of SymPy rounded to a float and the QR solution of mpmath at 120 digits. NumPy agrees within 1e-9 on the well-conditioned problems;

on the badly conditioned ones (x in [1000, 1001], x = 1e6 + k) its coefficients share no digit with the exact ones (the numbers are in the evidence file).

- Every function returns floats or raises `ValueError`: a degree that is not an integer from 0 to 10, data that are not lists or tuples of finite real numbers within 1e100 (booleans refused), different lengths, negative weights, fewer distinct x with a positive weight than degree + 1, a result too large for a float.

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_polyfit_0_1_0,
  author  = {Cavazzini, Andrea},
  title   = {{star-polyfit}: Least-squares polynomial, correctly rounded},
  version = {0.1.0},
  year    = {2026},
  url     = {https://andrea07072000.github.io/S.T.A.R.-/software/star_polyfit/},
  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