

PEER REVIEW RECORD

The Efficient Frontier from a LASSO Solver: A numerical check of the CLA–LASSO identity

Thomas Schmelzer

Vol. 1 (2026), Art. 7 · <https://doi.org/10.68365/jrepstat.2026.7>

The decision letters, the reports the authors read, their answers and the editor's signed decisions, as written. The reports come from the journal's review engine. The authors could have kept this record private until they confirmed the proof.

DECISION PATH

2026-10-14 Minor revision
2026-10-15 Revised
2026-10-15 Accepted

Round 1: Minor revision

SIGNED DECISION

Confirmed and signed by Matthias Templ on 2026-10-14 · ORCID 0000-0002-8638-5276

DECISION LETTER

Thank you for your submission to JRepStat. We are inviting a minor revision. The reviewers' points follow, grouped by weight.

This note verifies numerically that the mean–variance efficient frontier can be traced with a general LASSO solver (scikit-learn's `lars_path`) and no portfolio software, confirming the CLA–LASSO identity of Schmelzer and Hastie to about 10^{-13} on simulated factor models, Fama–French portfolios, and 494 S&P 500 stocks, and it contributes a response shift that lets a single positive-LASSO call reach the minimum-variance portfolio. It matters to practitioners and researchers who compute mean–variance frontiers and want corner-exact paths without a dedicated solver. The reviewers found nothing to fix in the writing itself. We would be glad to see the revised version.

How to read this letter. The decision rests on the items marked required. If a required item seems wrong to you, or cannot be done within this paper, say so in your response letter and give the reason. A reasoned decline is a legitimate answer, and the editor weighs it. Reviewers can be mistaken. Items marked recommended or optional are advice from the reviewers. Adopt them, decline them with a sentence, or leave them for later work. None of that counts against the paper. The reviewer reports below carry the same marks. The editor who signs this decision may weigh an item differently in the accompanying message, and that message prevails.

REQUIRED (the decision rests on these):

- [Reviewer 3] `fetch_sp500.py` draws the universe from a live Wikipedia page, so a rerun gets a different constituent set. Pin it to a fixed Wikipedia revision (oldid), or state in Section 8 that the stock universe is as-of a date and not exactly reproducible.

RECOMMENDED:

- [visualisation] In Fig 2 (left) and Fig 1(b) the distinction between a plain call and the shifted extension rests on a blue/red contrast. Blue against red is separable for most colour-vision types, but the red region in Fig 2 is described only as “shaded”, so a reader who cannot resolve the hue loses which weights come from the shift. The square markers in Fig 1(b) already add a shape cue; extending that redundancy to Fig 2 (for example a hatch or a boundary line at the stop, labelled) would make both panels legible in greyscale.

- [application-and-data] The data-availability statement names the fetch scripts and the pinned snapshots but does not say whether the two snapshot CSVs themselves are archived anywhere citable (for example a Zenodo deposit alongside the GitHub directory). Since the live sources revise, a reader who wants the note's exact numbers has no way to retrieve the frozen snapshot. Please state where, if anywhere, the two snapshot files can be obtained, or confirm that only the hashes are preserved.
- [ethics-and-integrity] Competing interests are not stated explicitly, though the relevant relationships are disclosed in substance: the author's affiliation (Jebel Quant Research) appears on the title, the paper is a companion to Schmelzer and Hastie [17] co-authored by the author, and the text states that the optimality checks were written by an author of [17] and are therefore not independent. Please add a short competing-interests statement making the commercial affiliation and the authorship link to [17] explicit, so the disclosure is formal rather than inferred.
- [methods-soundness] The step $v = v_0 + \text{len}(y) \cdot \text{alphas}$ relies on how scikit-learn scales the alphas returned by `lars_path` relative to the penalised objective. The KKT residuals confirm the scaling empirically, but one sentence stating the convention (why the number of rows is the correct multiplier, and that it is `len(y)` for both the Cholesky and centred-returns constructions) would let a reader reproduce the mapping without inspecting the solver internals.
- [Reviewer 1] state in the abstract and Section 8 that the real-data frontiers are a verification of the method at scale, not investable frontiers. The S&P 500 set is built from current Wikipedia constituents (survivorship) and Yahoo adjusted prices, so a reader should not read a historical or realistic frontier into it.
- [Reviewer 1] make the speed claim's dependence on QP grid resolution explicit. The "tens of times" figure is relative to a 100-point grid, so report the timing ratios in a small table and say how the advantage scales with grid density.
- [Reviewer 1] reconcile the two date ranges for the stock data: the body says "July 2021 to May 2026" while the availability section says prices "2021-06-01 to 2026-06-01".
- [Reviewer 2] Fold an explicit check that the recovered v reaches below the analytically computed v_{last} into Algorithm 1, so a user without ground truth can confirm every corner was captured.
- [Reviewer 2] Give the one-line reason accuracy degrades as $|v_0|$ grows (conditioning of the shifted response) and report the working δ window for each data set, so the default $\delta=1$ is justified rather than asserted.
- [Reviewer 2] State the sensitivity of the minimum-variance support to the hard NNLS threshold ($1e-9$), or justify it, since the support feeds the closed-form minimum-variance end.
- [Reviewer 2] Name the 100-point grid as a chosen comparison density and frame the path's advantage as returning corners exactly at any density, separating that from the raw speed ratio.
- [Reviewer 3] Separate single-machine determinism (fixed seeds, bit-identical) from cross-machine last-digit variation in the availability section, and name which in-text numbers the `test` target asserts and at what tolerance.
- [Reviewer 3] Mark the printed code as excerpts and note version preconditions (for example `numpy >= 2` for `cholesky(upper=True)`) so a reader who copies them is not tripped.
- [Reviewer 3] Deposit the derived return matrices as a frozen fixture where source licenses allow, or note the fetch scripts' dependence on the Yahoo API and the monthly CRSP vintage, so the external-data checks stay runnable after those change.

OPTIONAL:

- [visualisation] Fig 1(b) begins its y-axis at 0.5 rather than 0. For a frontier this is a defensible zoom to show curvature, and the inset handles the low-risk end, but a one-line note in the caption that the axis is truncated would prevent the arc's steepness being read as larger than it is.
- [methods-soundness] The identity and its numerical support rest on $\Sigma \succ 0$ with $T > n$ in every tested case; the singular and high-dimensional regimes are set aside explicitly (Section 9, and [17, §7]). A later study could probe behaviour as Σ approaches singularity, but this is outside the present note.
- [Reviewer 1] a comparison against a constrained-lasso path algorithm on a two-constraint case (dollar-neutral or 130/30) would show how far the single-call approach generalises.
- [Reviewer 2] Report whether tied knots (two assets entering at one step) arose on the real data, and how LARS handled them.

- [Reviewer 3] Guard the first branch of the `nu_last min()` against an empty `b>0` slice with an `initial`, as the second branch already does.

REVIEWS

Report Reviewer 1, recommends accept

Written by the journal's review engine in the role of the application reviewer, and read by the editor before the decision. How the engine works is on the [peer review page](#).

This is a companion numerical note to the theory paper [17], and it does what it sets out to do. It computes the efficient frontier in three settings (long–short under a gross cap, fixed leverage with varying risk appetite, and long-only fully invested) using only a stock LASSO solver, and it checks each returned portfolio two ways: against its own KKT conditions and against an independent quadratic-programming solver. The agreements reported, residuals near 10^{-13} to 10^{-15} and weight gaps at the same scale, are consistent across the test problem, the factor models of Table 1, the Fama–French portfolios, and the S&P 500 stocks. The reasoning for the one real disagreement, at $n=500$, where the smaller KKT residual of the path is taken to locate the error in the QP baseline, is sound because KKT is the actual optimality condition rather than a second opinion.

From an application standpoint the problem is real and the framing is honest. The claim that corner portfolios can be obtained without portfolio software is genuinely useful for the two cases scikit-learn can express, and the response shift of equation (4) and Algorithm 1 are a real practical addition: reaching the minimum-variance portfolio in a single positive-LASSO call, with a ν_0 computed in advance, is the kind of detail that saves an implementer real effort. The economic reading of ν as Black's zero-beta rate, and of the bracketing columns as an interpolation for the maximum-Sharpe portfolio at any rate, is correct and adds value beyond a bare numerical check. Section 9 is candid about the boundaries: a singular covariance, and any mandate needing more than one linear constraint, falls outside what this tool does. Most real mandates do carry more than one constraint, so the practical niche is narrow, and the paper says so clearly, which is the right treatment.

Two points on the data would help a reader. The S&P 500 set is built from current index membership and Yahoo adjusted prices, so it carries survivorship and vintage effects. None of this touches the paper's claim, which is about numerical agreement of a computation and not about the realism or performance of any frontier, but an applied-finance reader meeting “efficient frontier of 494 S&P 500 stocks” can infer an investable object that is not intended. One or two sentences placing the real data as a scale-and-accuracy check settle it, in the same spirit as the “for illustration only” note on the test problem. Separately, the body and the availability section give different date windows for the stock data, and reconciling them removes a stumble for anyone cross-checking the fetch script.

The speed argument needs one qualification. The accuracy argument is airtight: a path returns the corners exactly where any grid misses them. The speed argument, “a 100-point grid of QPs costs tens of times as much as the path”, depends on the grid resolution, which is a modelling choice. Reporting the ratios in a small table and stating how the margin moves with grid density would let a reader map the claim onto their own setup. The warm-started OSQP comparison is a good addition and already strengthens this part.

On reproducibility the declared package is strong for an applied note: a single self-describing script, a lockfile pinning Python and every dependency to version and hash, published fetch scripts, data snapshots identified by SHA-256, an R cross-check, and a tolerance story that accounts for BLAS-level variation across machines. I did not run it, so I report what is documented rather than a reproduction I performed. The authors also state that the checks were written by an author of [17], which the independent baseline solvers mitigate.

Numbered suggestions to the authors:

1. **RECOMMENDED** Add one or two sentences in the abstract and Section 8 stating that the real-data frontiers verify the method at scale and are not investable frontiers, and noting that the S&P 500 set uses current index membership (survivorship) and adjusted prices.

2. **RECOMMENDED** Report the path-versus-grid timing ratios in a small table and state how the speed advantage depends on the QP grid resolution, so the “tens of times” claim is tied to its grid.
3. **RECOMMENDED** Reconcile the stock-data date windows between the body (“July 2021 to May 2026”) and the availability section (“2021-06-01 to 2026-06-01”).
4. **OPTIONAL** Compare against a constrained-lasso path algorithm on a two-constraint case such as dollar-neutral or 130/30, to show how far the single-call approach extends beyond scikit-learn’s single constraint.

Report Reviewer 2, recommends accept

Written by the journal’s review engine in the role of the methodology reviewer, and read by the editor before the decision. How the engine works is on the [peer review page](#).

This note does one job and does it carefully: it checks, numerically and against independent solvers, that the constrained-LASSO path and the Markowitz frontier are the same curve, and it packages a usable recipe for the long-only case. The change of variables in equation (1) is correct and cleanly stated. With X equal to the upper Cholesky factor, $X^T X = \Sigma$ and $X^T y = \mu$, and the identity $1/2 w^T \Sigma w - \mu^T w = 1/2 \|y - Xw\|^2 - 1/2 \|y\|^2$ follows directly, so the mean-variance problem under an ℓ_1 cap is the LASSO in constrained form. I checked the signs through to the budget multiplier $\nu = (\mu - \Sigma \nu)_j$ on the support, and the parametrisations in Sections 5, 6 and 7 are mutually consistent.

The one genuinely new mechanical contribution is the shift in equation (4). Completing the square moves the penalty into the response because $X^T d = 1$ makes $\nu^T \nu$ equal to $\nu^T X^T X \nu$, so running the nonnegative LASSO on $y - \nu^T d$ with penalty ν' realises the effective multiplier $\nu^0 + \nu'$ and reaches the segment below where a plain penalised solver stops. This is the right way to recover the part of the frontier that a LASSO solver cannot reach, since it will not take a negative penalty. The detail that scikit-learn reports its alpha divided by the sample count, handled by $\text{nu} = \text{nu}_0 + \text{len}(y) * \text{alphas}$, is a place where such notes usually go wrong, and it is right here for both the square $X^T X = \Sigma$ construction and the tall centred-returns construction.

The evidence is the kind a methodology reader wants. Each returned portfolio is tested against its own KKT conditions, and separately against baselines that do not share code with the path (Clarabel through cvxpy, quadprog, OSQP). The residuals sit at $1e-13$ to $1e-15$, and where the path and a baseline disagree the residual identifies the baseline as the less accurate party, which is the correct reading for an interior-point method at a corner. The disclosure that the self-consistency checks were written by an author of the companion paper, with the independence resting on the external solvers, is the honest framing and I accept it. The reproducibility setup is strong: a pinned lockfile, a test command with stated tolerances, data-fetch scripts with dataset hashes, and timings given as machine-portable ratios with absolute times withheld. That is close to what I would hold up as the target for an applied note of this kind.

My reservations are about the recipe’s reliability in a practitioner’s hands, and they are improvements to a sound method rather than doubts about it. The choice of δ in $\nu^0 = \nu^{\text{last}} - \delta \max(|\nu^{\text{last}}|, \|\mu\|_\infty)$ is presented through its observed behaviour: on the stock data $\delta=1e-4$ drops the last corner and $\delta=100$ inflates the residual to $2e-11$, while on the portfolio data even $\delta=1e-2$ drops the last corner and $\delta=1$ recovers all twelve. The default $\delta=1$ therefore sits inside a window whose lower edge is data-dependent, and a user running a new problem has no way to see a lost corner from the output alone. The method already computes ν^{last} analytically, so the fix is cheap: check that the recovered ν descends below ν^{last} and say so in Algorithm 1, and give the one-line conditioning argument behind “accuracy falls roughly in proportion to $|\nu^0|$ ” so the tradeoff is explained and not only observed. In the same spirit the minimum-variance support is read off an NNLS solution with a fixed $1e-9$ cutoff, and a word on how sensitive the recovered support is to that cutoff would close the last gap in the closed-form minimum-variance end.

On Section 9, the speed claim rests on a 100-point QP grid, and the honest statement of the path’s advantage is that it returns the corners exactly at any grid density, which is separate from the raw time ratio. Making that separation explicit would stop a reader reading the ratio as the whole case. None of this changes the note’s conclusions. The paper is correct where I can check it analytically, honest about what it verifies and what it assumes from the companion proofs, and reproducible to a degree most submissions are not.

Numbered suggestions to the authors:

1. **RECOMMENDED** Add to Algorithm 1 an explicit check that the recovered ν descends below the analytically computed ν_{last} , so a user without ground truth can confirm every corner was captured, and mention it in the δ discussion of Section 7.
2. **RECOMMENDED** Give the one-line reason the shifted problem loses accuracy as $|\nu_0|$ grows (conditioning of $y - \nu_0 d$), and report the working δ window for each data set, so the default $\delta=1$ is justified rather than asserted.
3. **RECOMMENDED** State how sensitive the minimum-variance support S is to the $1e-9$ NNLS threshold, or justify the threshold, since S determines the closed-form minimum-variance end.
4. **RECOMMENDED** In Section 9, name the 100-point grid as a chosen comparison density and state the path's advantage as returning corners exactly at any density, kept separate from the speed ratio.
5. **OPTIONAL** Report whether tied knots (two assets entering at one LARS step) occurred on the 99-, 494- or 500-asset problems, and how they were handled, since degeneracy is where a path method is most fragile.

Report Reviewer 3, recommends minor

Written by the journal's review engine in the role of the software reviewer, and read by the editor before the decision. How the engine works is on the [peer review page](#).

I read this as a software-and-reproducibility reviewer: can I run it, is the code sound, and are the reported numbers the code's numbers? On the first two the answer is largely yes, and the packaging is better than most submissions of this kind. A single Python script that declares its own pinned dependencies, a lockfile that fixes Python 3.12 and every package to an exact version and hash, one documented command to regenerate the figures and numbers and another to run the checks, an R cross-check, and SHA-256 fingerprints for the two external snapshots. The core claim, that the CLA frontier and the LASSO path coincide, rests on simulated data generated inside the script from fixed seeds, so the central verification needs no network and no copyrighted data. That is the right architecture: the load-bearing evidence is self-contained, and the external datasets are scale tests layered on top.

I checked the printed code against the APIs it calls. The Cholesky, lstsq, lars_path, lars_path_gram, nnls and interpolation snippets are consistent with current scikit-learn, scipy and numpy, and the alpha-to- ν rescaling ($\nu = \nu_0 + \text{len}(y) * \text{alphas}$) and the max_iter guard match the text. I found no demonstrable bug in the printed lines. One version precondition is worth surfacing for readers who copy the snippets: cholesky(Sigma, upper=True) needs numpy 2.0 or later, and lstsq will warn about rcond on some versions. The lockfile presumably pins a compatible numpy, so this is about the excerpts in the paper, not the script.

My one required point concerns the stock-data reproduction. fetch_sp500.py takes the constituent list from a live Wikipedia page and the prices from Yahoo, and the raw data is not redistributable. The paper already documents that a later download differs and sets a 10^{-8} tolerance for the baseline drift, which is the responsible thing to do. The remaining gap is the universe itself: a live page means a rerun does not even start from the same set of names, so the "anyone can repeat them" claim of the abstract is only partly met for Section 8. Pinning the fetch to a Wikipedia revision id is a few hours and fixes the universe. If that is impractical, one sentence in Section 8 stating the universe is as-of a date and not exactly reproducible closes the same gap. Either settles it, and neither touches the core result.

The remaining points strengthen the reproducibility story without changing any claim. The availability section mixes two different guarantees: seeds fixed means a single machine should reproduce bit-for-bit, while cross-machine BLAS differences move the last digits, which is why the tolerances exist. Stating those two separately, and naming which in-text numbers (the 7.765, the 2.120, the residual figures, the corner counts) the test target actually asserts, would let a reader see the scope of "regenerates every figure and number" rather than take it on trust. Depositing the derived return matrices as a frozen fixture where the French and Yahoo terms allow would also keep the external checks runnable after the Yahoo API or the monthly CRSP vintage moves, since a fetch script that depends on a third-party API tends to stop running within a year or two. The nu_last computation guards its second branch with an initial but not its first, which would raise on an empty $b > 0$ slice; in practice the minimum-variance support gives a non-empty positive b , so this is a defensive tidy-up only.

The disclosure that the checks were written by an author of [17] is handled correctly, with independent baseline solvers (Clarabel, quadprog, OSQP) carrying the comparison and the scripts published for rerun. That is the right way to manage it.

Numbered suggestions to the authors:

1. **REQUIRED** Pin `fetch_sp500.py` to a fixed Wikipedia revision (oldid) so the S&P 500 universe is reproducible, or add one sentence in Section 8 stating the universe is as-of a date and not exactly reproducible.
2. **RECOMMENDED** In the availability section, separate single-machine bit-identical determinism (fixed seeds) from cross-machine last-digit variation, and list which in-text numbers `make_figures.py` test asserts and at what tolerance.
3. **RECOMMENDED** Mark the in-text code as excerpts and state version preconditions, for example that `cholesky(Sigma, upper=True)` needs numpy 2.0 or later.
4. **RECOMMENDED** Deposit the derived return matrices as a frozen fixture where the source licenses permit, or note in the availability section that the fetch scripts depend on the Yahoo API and the monthly CRSP vintage and may cease to run.
5. **OPTIONAL** Add an `initial` to the first branch of the `nu_last` `min()` to match the second branch and avoid a raise on an empty `b>0` slice.

REVISION, RECEIVED 2026-10-15

This revision was answered under the earlier rule: a short note on what changed, and a reason for each required point the authors declined.

The authors' response letter, as uploaded, follows as Appendix A.

Round 2: Accepted

SIGNED DECISION

Signed by Matthias Templ on 2026-10-15 · ORCID 0000-0002-8638-5276

Final acceptance of version 2, after the editor read the preview of the article.

DECISION LETTER

Congratulations, your manuscript has been accepted for publication in JRepStat. Production begins now. Where it is not on file yet, we still need the following for production.

- a valid ORCID iD for every author
- a captured computational environment (dependencies/lockfile/container)
- a signed determinism attestation for the entry-point command

REVIEWS

Report Reviewer 1, recommends accept

Written by the journal's review engine in the role of the application reviewer, and read by the editor before the decision. How the engine works is on the [peer review page](#).

Of the three recommended points and one optional one I raised, the three recommended points are met and the optional one is parked, which is the right call for this note.

Replacing the S&P 500 download with a seeded five-factor simulation settles two of my asks at once. The survivorship and vintage caveat is no longer needed, because there is no real-equity frontier for a reader to mistake for an investable object, and the conflicting date windows disappear with the download. For this journal the change is a net gain: every number in Section 8 except the Fama–French block now regenerates from a seed, and the removed fetch script, lockfile entry and download tolerances simplify the reproducibility story rather than weakening it. Withdrawing the two-machine timing claim and the OSQP 1e-5 remark is consistent with keeping only what the single measured run supports.

My second point, on the speed claim, is substantially addressed. Both Section 8 and Section 9 now name the grid as 100 points, so the “about sixty times” figure is tied to a stated resolution (one to two path solves against roughly a hundred QP solves is internally consistent). The contrast between the 99-portfolio case, where a grid costs about as much as the path, and the 500-stock case, where it costs about sixty times as much, shows how the margin moves with problem size. This covers the practical substance of what I asked for. A small table sweeping grid density was the lighter form of the ask and is not needed for the claim to stand.

The competing-interests statement is clear about the `cvxcla` maintenance and about where the note uses it, and the method’s independence from that package (`scikit-learn` or `R` alone) is correct as stated. Nothing in the revised version raises a new material concern from an application standpoint. The paper is a numerical verification note, the simulated data serve that purpose fully, and the niche (one linear constraint beyond the signs) is stated plainly in Section 9.

Numbered suggestions to the authors: (none — the recommended points are met and the paper is ready as it stands)

Report Reviewer 2, recommends accept

Written by the journal’s review engine in the role of the methodology reviewer, and read by the editor before the decision. How the engine works is on the [peer review page](#).

I recommended accept last round, and the revision does not disturb that. Of the four recommended points and one optional one I raised, the Section 9 separation of accuracy from speed is now done cleanly (the path finds corners exactly at any density, a grid of QPs misses them, and the speed advantage is stated only where it holds, at scale), and the δ windows are now reported for every data set. The move to a seeded simulated universe in Section 8 removes the last irreproducible download and the competing-interests statement is present and specific; both strengthen the note against the journal’s aims.

What remains is the same small set, none of it blocking. Algorithm 1 still does not carry the explicit check that the recovered v descends below v_{last} , which was the cheapest way to let a user see a dropped corner from the output alone, and it is the point I would most like closed since the δ discussion shows how easily the last corner is lost ($\delta=1e-4$ on the stock data, $\delta=1e-2$ on the portfolio data). The conditioning argument behind “accuracy falls roughly in proportion to $|v_0|$ ” is still stated as an observation, not explained. The sensitivity of the minimum-variance support to the $1e-9$ NNLS cutoff is still unaddressed, and the tied-knot question remains unreported. All four are improvements to a method I judge correct where I can check it analytically, and the note stands without any of them.

Numbered suggestions to the authors:

1. RECOMMENDED Add to Algorithm 1 the explicit check that the recovered v descends below v_{last} , and mention it in the δ discussion of Section 7, so a dropped last corner is visible from the output.
2. RECOMMENDED Give the one-line reason the shifted problem loses accuracy as $|v_0|$ grows (conditioning of $y - v_0 d$), so the reported δ windows read as explained rather than only measured.
3. RECOMMENDED State how sensitive the minimum-variance support S is to the $1e-9$ NNLS threshold, or justify the threshold, since S determines the closed-form minimum-variance end.
4. OPTIONAL Report whether tied knots (two assets entering at one LARS step) occurred on the 99-, 500- or 116-corner problems, and how they were handled.

Report Reviewer 3, recommends accept

Written by the journal’s review engine in the role of the software reviewer, and read by the editor before the decision. How the engine works is on the [peer review page](#).

Of the one required point and four lighter ones I raised, the required point is settled and two of the recommended ones with it.

The required point was the live Wikipedia universe in `fetch_sp500.py`, which made the at-scale Section 8 reproduction start from a different constituent set on every rerun. Replacing the S&P 500 download with a seeded five-factor simulation of 500 stocks over 1,250 days removes the external dependency outright, so the at-scale example now reproduces exactly from the committed code. That is a stronger fix than the oldid pin or the as-of sentence I offered as alternatives, and the two claims that rested on the old data (the

two-machine cost ratio and the 1e-5 OSQP miss) are correctly withdrawn rather than carried over. The only remaining external data is the Fama-French portfolio set, and the availability section now records the CRSP vintage, the SHA-256, and that a later download can differ where CRSP revises a return. That closes my point 4 for the data that is still fetched.

The determinism point (my point 2) is addressed: the availability section now separates the software lock from hardware variation, states that last digits differ across CPU and BLAS builds, and gives the tolerance band the tests assert (10^{-12} to 10^{-9}). It does not enumerate which individual in-text numbers the test target checks, but the band and the scope statement cover the substance, and this was recommended.

Two items are still open and neither blocks. The in-text snippets are not marked as excerpts and carry no version precondition, so a reader who copies `cholesky(Sigma, upper=True)` onto numpy 1.x will hit a `TypeError`; the lockfile protects the script, not the copied lines. The first branch of the `nu_last min()` still has no `initial` while the second does, which in practice is harmless because the minimum-variance support gives a non-empty positive `b`.

The competing-interests statement is now present and names the note's two uses of `cvxcla`, which handles the disclosure well.

Numbered suggestions to the authors:

1. **RECOMMENDED** Mark the in-text code as excerpts and state version preconditions, for example that `cholesky(Sigma, upper=True)` needs numpy 2.0 or later.
2. **OPTIONAL** Add an `initial` to the first branch of the `nu_last min()` to match the second branch and avoid a raise on an empty `b>0` slice.

APPENDIX A

The authors' response letter to round 1

As uploaded with the revision received 2026-10-15. Its own pages follow.

Letter to the Editor — Revisions to the Lasso Manuscript

Oct 15, 2026 · @Thomas

Dear Editor,

I write to report one revision to my submission to the Journal of Reproducible Statistics, "The Efficient Frontier from a LASSO Solver". The example at scale in Section 8 now runs on simulated data, so every number in the note can be reproduced exactly from the companion code. The revision also adds a competing-interests statement, as the referee recommended.

Changes in this revision

A simulated universe replaces the S&P 500 download. The submitted version drew prices from Yahoo Finance for a snapshot of the S&P 500. The constituent list could be fixed in a committed file, but the prices could not, so the snapshot could not be reproduced exactly. The example now simulates daily returns of 500 stocks over 1,250 days from a seeded five-factor model: a market factor at 1% daily volatility, four style factors at 0.5%, idiosyncratic volatility between 1% and 3%, and expected returns around 5 bp a day. The mean and covariance are estimated from these returns, as before.

The reported numbers follow the new data. Algorithm 1 returns all 116 corners of the frontier. They match the quadratic-programming solutions to $8.6e-13$, with KKT residuals below $4.7e-13$, and the last corner sits at $v = -0.190$. One full path costs about 1.7 QP solves, against about 59 for a 100-point grid.

Two claims are withdrawn. The note no longer states that the cost ratios held on two machines, since only one was measured. It also drops the remark on a $1e-5$ OSQP miss, which does not occur on the simulated data.

Competing interests are now disclosed. I maintain the `cvxcla` package. The new statement says so, and names the note's uses of it: Section 9 points readers to `cvxcla` for portfolios with more constraints, and the companion script checks its corners against it.

The code is simpler. The download script, its lockfile, the constituent file and the `pyarrow` pin are removed. The test of the stocks example now always runs, with no download tolerances. The README, the introduction and the availability section are updated to match; the repository's reproducibility notes keep the S&P 500 history for the record.

I believe these changes make the note fully reproducible without external data, which suits the aims of the journal. I would be glad to answer any further questions from you or the referee.

Yours sincerely,

Thomas Schmelzer