



RESEARCH ARTICLE · APPLIED ANALYSIS

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

Thomas Schmelzer\*

Jebel Quant Research, Abu Dhabi

\*Corresponding author: thomas@jqrae

## ARTICLE INFO

### Keywords

efficient frontier  
critical line algorithm  
LASSO  
least angle regression  
portfolio optimisation  
reproducibility

### History

Received 14 Oct 2026  
Final version 17 Oct 2026  
Accepted 15 Oct 2026  
Version 1

### Handling editor

Matthias Templ

## ABSTRACT

Schmelzer and Hastie prove that Markowitz’s Critical Line Algorithm and the LASSO path trace the same curve. This companion note checks that result numerically with a stock LASSO solver, `lars_path` from `scikit-learn`, and no portfolio software. It covers long–short portfolios under a leverage cap, fixed leverage with varying risk appetite, and the classical long-only, fully invested frontier. Every corner the path returns meets its own optimality conditions to about  $10^{-13}$  and agrees with independent quadratic-programming solvers, on factor models of up to 500 assets, on the Fama–French size and book-to-market portfolios, and on five years of simulated daily returns of 500 stocks. A plain call stops the long-only path at the maximum–Sharpe portfolio; shifting the response once, by an amount computed in advance, lets a single call reach the minimum-variance portfolio. The script, its checks, a script that fetches the data, and an R cross-check are published, with a locked environment, so that anyone can repeat them.

## ✓ Reproduced verified 2026-10-15

## REPRODUCIBILITY RECORD

|             |                                                                                                                                                                                                                |                         |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| Result      | Reported results reproduced. All 42 counts and path values reproduce on macOS arm64 and Linux. Residuals near $10^{-13}$ differ in their last digits, as the article states.                                   | ✓ measured              |
| Entry point | <code>cd code/lasso-3e2853642989717c716fed9eca50a3873fa04dd3/preprints/lasso &amp;&amp; uv run python/make_figures.py test</code>                                                                              | ✓ executed              |
| Environment | Python 3.12 from <code>make_figures.py.lock</code> via <code>uv</code> , Linux aarch64 container and macOS arm64, R 4.6.1 with <code>lars</code> 1.3                                                           | ✓ measured              |
| Determinism | Seeded.                                                                                                                                                                                                        | declared                |
| Archive     | <code>the-efficient-frontier-from-a-lasso-solver-a-numerical-check-of-the-cla-lasso-identity-v1-repro.zip</code> · reviewed bundle SHA-256<br>0b9d144d369821d8ea3ba6d7d4f15aff50c8ff5e0bb0b206c7ef6136f1492983 | ✓ computed              |
| Code        | <code>https://github.com/Jebel-Quant/lasso</code>                                                                                                                                                              | declared                |
| Data        | Open.                                                                                                                                                                                                          | declared, route checked |

## 1 Introduction

As the trade-off between expected return and variance changes, the optimal portfolio moves along straight lines and changes direction only at finitely many *corner portfolios*, where an asset enters or leaves. Markowitz’s Critical Line Algorithm (CLA) (Markowitz, 1956) walks from one corner to the next. The LASSO (Tibshirani, 1996) behaves the same way as its penalty moves, and the LARS algorithm (Efron et al., 2004) walks its corners, which statisticians call knots. Schmelzer and Hastie (2026) prove that after

CITE AS Schmelzer, T. (2026). The Efficient Frontier from a LASSO Solver: A numerical check of the CLA–LASSO identity. Journal of Reproducible Statistics, 1, 7. <https://doi.org/10.68365/jrepmat.2026.7>

a change of variables the two paths coincide. This note checks that result numerically, by computing frontiers with a LASSO solver and testing them against independent solvers, in three settings:

1. long–short under a gross-exposure cap, where the plain LASSO path is the frontier (Section 5);
2. fixed leverage with varying risk appetite, where the same path is rescaled (Section 6);
3. long only and fully invested, where the nonnegative LASSO path, shifted once, covers the whole frontier in one call (Section 7, Algorithm 1).

The usual alternative is to solve a quadratic program on a grid of tilts. Each solve is only as accurate as its tolerance, and every corner between two grid points is missed. A path algorithm returns the corners themselves, and every other frontier portfolio is a linear interpolation between two of them.

## 2 Background

Markowitz (1952) called a portfolio efficient if no other has a higher expected return  $E$  without a higher variance  $V$ , or a lower  $V$  without a lower  $E$ . Traced by minimising  $\frac{1}{2}V - \lambda E$  over fully invested portfolios for  $\lambda \geq 0$ , they form the efficient frontier, with the minimum-variance portfolio at  $\lambda = 0$ . The CLA followed (Markowitz, 1956). Tobin (1958) and Sharpe (1964) moved the picture to the  $(\sigma, E)$  plane, where Sharpe located the dominant portfolio, the one with the highest Sharpe ratio, at the tangent from the riskless rate.

The LASSO fits least squares under  $\|\beta\|_1 \leq t$  and so shrinks and selects at once. Its path is piecewise linear in  $t$ . Osborne, Presnell and Turlach (2000) gave a homotopy for it and Efron et al. (2004) gave LARS, which also covers the positive LASSO. Later practice moved to coordinate descent on a grid of penalties (Friedman et al., 2010), which, like a grid of tilts, returns points rather than corners.

The two literatures developed for decades without citing each other (Schmelzer & Hastie, 2026, Section 5). The best-known bridge, Brodie et al. (2009), read a Markowitz problem with a gross-exposure penalty as a LASSO, but swept the penalty at a fixed target return rather than tracing the frontier. Fan, Zhang and Yu (2012) ran LARS along the gross-exposure cap of the minimum-variance portfolio. They removed the budget by regressing on a reference asset, which leaves an  $\ell_1$  constraint that is no longer the LASSO's, so their path is an approximation.

## 3 The test problem

The test problem is the seeded 20-asset example of Figure 1 in Schmelzer & Hastie (2026): a five-factor covariance  $\Sigma = \text{diag}(d) + U \text{diag}(\delta) U^\top$ , and  $\mu$  the sample mean of 50 days of returns simulated from it. Its units are arbitrary and its expected returns large against its risk, so Sharpe ratios reach 2.1 per period, far above anything real assets offer. It is for illustration only; Section 8 adds real portfolio returns and a larger universe on a daily scale. The companion script `python/make_figures.py` regenerates every figure and number in the note, and its tests check each against a tolerance. The statement on code and data availability at the end says where to find it.

## 4 From mean–variance to least squares

Let  $\Sigma > 0$  be the covariance of  $n$  assets and  $\mu$  their expected returns. With the Cholesky factor  $\Sigma = LL^\top$ , set  $X = L^\top$  and  $y = L^{-1}\mu$ , so that  $X^\top X = \Sigma$  and  $X^\top y = \mu$ . Then for every portfolio  $w$

$$\frac{1}{2} w^\top \Sigma w - \mu^\top w = \frac{1}{2} \|y - Xw\|_2^2 - \frac{1}{2} \|y\|_2^2, \quad (1)$$

and under any constraints the mean–variance problem is a least-squares problem.

```
X = np.linalg.cholesky(Sigma, upper=True) # X.T @ X == Sigma
y = np.linalg.lstsq(X.T, mu)[0]          # X.T @ y == mu
w = np.linalg.lstsq(X, y)[0]              # unconstrained optimum Sigma^{-1} mu
```

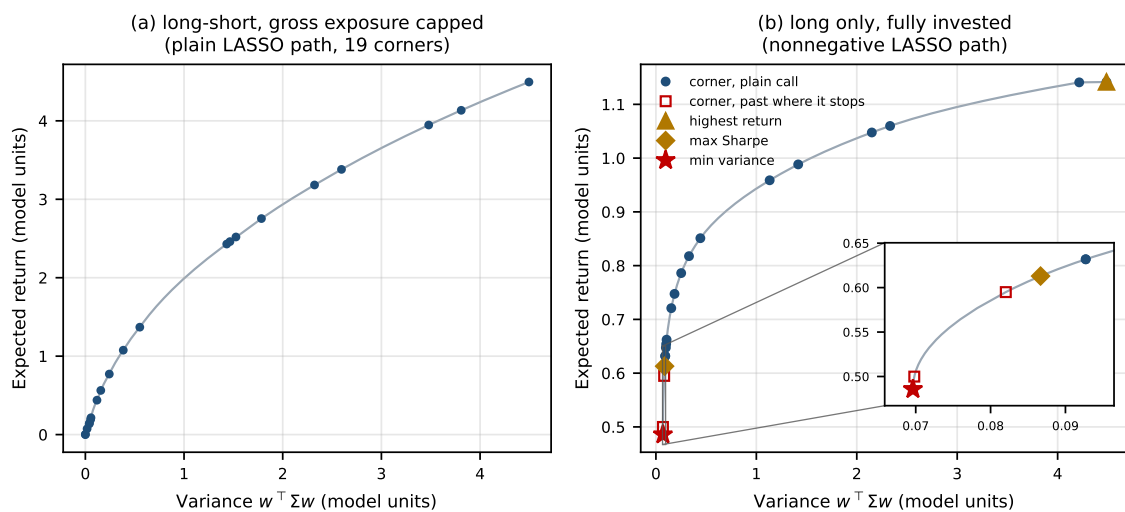


**Figure 1** Both frontiers of the test problem, computed with a LASSO solver. (a) Long-short under a gross-exposure cap. (b) Long-only, fully invested: blue corners come from a plain call, red squares are the two it misses (Section 7). A triangle marks the highest-return portfolio, a diamond the maximum-Sharpe and a star the minimum-variance portfolio; the inset magnifies the low-risk end.

The last line solves the normal equations  $X^T X w = X^T y$ , which are  $\Sigma w = \mu$ . With constraints, the optimality conditions become normal equations restricted to the assets held, a linear system in the trade-off parameter. The weights stay linear until the holdings change, and those changes are the corners. The substitution is a change of variables, not a statistical model:  $X$  has one row per asset, and  $y$  is not data (Schmelzer & Hastie, 2026, Section 4).

Only  $X^T X$  and  $X^T y$  matter, so any square root of  $\Sigma$  will do. If  $\Sigma$  is a sample covariance from a  $T \times n$  return matrix  $R$  with  $T > n$ , the centred returns  $X = (R - \mathbf{1}\bar{r}^T)/\sqrt{T-1}$  work, with  $y$  from `ltsq`. For a factor model, stacking  $\text{diag}(d)^{1/2}$  over  $\text{diag}(\delta)^{1/2} U^T$  gives an  $(n+k) \times n$  matrix with no factorisation at all, and  $y = (\text{diag}(d)^{-1/2} \mu, 0)$ . Both reproduce the Cholesky path to rounding.

## 5 Long-short with a leverage cap

**Parameters.**  $\lambda$ : the tilt, or risk appetite.  $c$ : the gross-exposure cap, and  $\kappa$ : its multiplier, the LASSO penalty of Sections 5 and 6.  $t = \mathbf{1}^T v = 1/\lambda$ : the budget of the rescaled long-only problem, and  $v$ : its multiplier, the LASSO penalty of Section 7.

With the gross exposure  $\|w\|_1$  capped at  $c$ ,

$$\min_w \frac{1}{2} w^T \Sigma w - \mu^T w \quad \text{s.t.} \quad \|w\|_1 \leq c \quad (2)$$

is, by (1), the LASSO in constrained form, the observation behind the sparse portfolios of Brodie et al. (2009). Its penalised form is  $\min_{\beta} \frac{1}{2} \|y - X\beta\|_2^2 + \kappa \|\beta\|_1$ , where  $\kappa \geq 0$  is the multiplier of the cap and  $c = \|\beta(\kappa)\|_1$ . So the LASSO path is the frontier of (2) (Schmelzer & Hastie, 2026, Theorem 1), and `scikit-learn` (Pedregosa et al., 2011) computes it:

```
_, _, B = lars_path(X, y, method="lasso") # columns: start, corners, end
_, _, B = lars_path_gram(Xy=mu, Gram=Sigma, n_samples=len(mu), method="lasso")
```

The second line gives the same path from  $(\Sigma, \mu)$  alone. In R the `lars` package (Hastie & Efron, 2022) traces it too:

```
library(lars)
X <- chol(Sigma) # t(X) %*% X == Sigma
y <- backsolve(X, mu, transpose = TRUE) # t(X) %*% y == mu
```

```
fit <- lars(X, y, type = "lasso", normalize = FALSE, intercept = FALSE)
B <- t(coef(fit))                                # columns: start, corners, end
```

It has no positivity constraint, so the long-only path of Section 7 needs `scikit-learn`. On the test problem the path has 21 columns: the empty portfolio at the start, 19 corners, each adding one asset, and the end. The path starts at  $w = 0$ , which is optimal while  $\kappa \geq \max_j |\mu_j|$ , and ends at  $\kappa = 0$ , where the cap stops binding at  $w^\star = \Sigma^{-1}\mu$  with  $\|w^\star\|_1 = 7.765$ . That end maximises  $\mu^\top w - \frac{1}{2}w^\top \Sigma w$ , the second-order approximation to expected log growth, so it is the Kelly portfolio (Kelly, 1956). Since  $\Sigma^{-1}\mu$  also maximises the scale-free Sharpe ratio, it points towards the maximum-Sharpe portfolio. On the test problem the Sharpe ratio rises from corner to corner, from 0.539 at the first to  $\sqrt{\mu^\top \Sigma^{-1} \mu} = 2.120$  at the end, although nothing shown here says it must. Neither end is the minimum-variance portfolio, which needs the budget  $\mathbf{1}^\top w = 1$  that (2) lacks.

## 6 Fixed leverage, varying risk appetite

A mandate usually fixes the leverage, say  $c = 2$ , and varies risk appetite through a tilt  $\lambda > 0$ . Substituting  $w = \lambda v$  in  $\min \frac{1}{2}w^\top \Sigma w - \lambda \mu^\top w$  subject to  $\|w\|_1 \leq c$  gives (2) with cap  $c/\lambda$ . The same path therefore serves: a column  $\beta$  with  $t = \|\beta\|_1$ , scaled to  $w = (c/t)\beta$ , is optimal at tilt  $\lambda = c/t$ .

```
t = np.abs(B[:, 1:]).sum(0)                # gross exposure of each column
W, tilts = B[:, 1:] * (c / t), c / t      # the same portfolios at leverage c
```

At  $c = 2$  the columns cover tilts from 745.6 down to 0.2576. Below that the cap is slack and the solution  $\lambda \Sigma^{-1}\mu$  shrinks to zero, so the path covers every tilt.

## 7 Long only, fully invested

The classical frontier requires  $w \geq 0$  and  $\mathbf{1}^\top w = 1$ . Substituting  $w = \lambda v$  and using (1) turns the tilted problem into

$$\min_v \frac{1}{2} \|Xv - y\|_2^2 \quad \text{s.t.} \quad \mathbf{1}^\top v = t, v \geq 0, \quad t = 1/\lambda, \quad (3)$$

with  $w = v/t$  (Schmelzer & Hastie, 2026, Proposition 3). For  $v \geq 0$  the budget is  $\|v\|_1$ , so (3) is the nonnegative LASSO, whose penalty  $\nu$  is the multiplier of the budget. `lars_path(..., positive=True)` traces it, and each column divided by its sum is a corner portfolio at tilt  $1/\mathbf{1}^\top v$ .

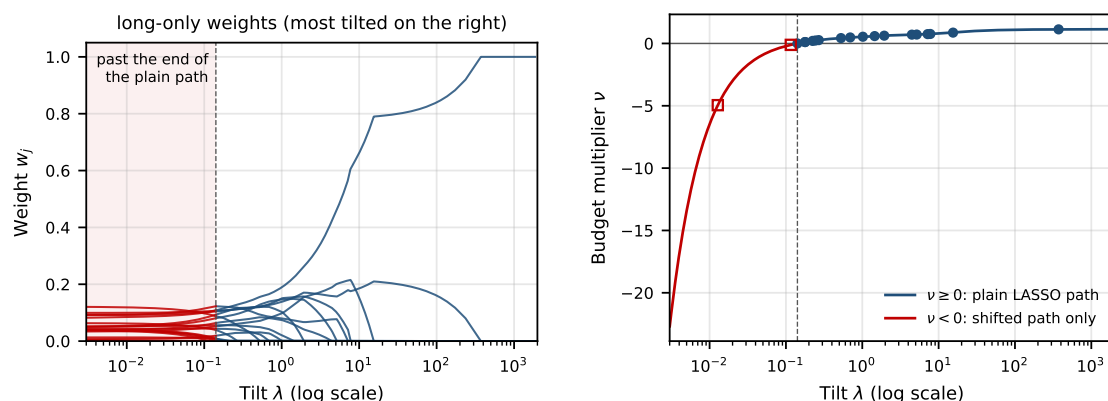


**Figure 2** The long-only path of the test problem against the tilt  $\lambda$ . *Left*: the weights; blue where a plain call supplies them, red (shaded) where it does not. *Right*: the budget multiplier  $\nu = (\mu - \Sigma v)_j$ ,  $j$  on the support. A plain call covers  $\nu \geq 0$  and ends at  $\nu = 0$ ,  $\lambda = 0.1414$  (dashed); the shift (4) reaches the two further corners (squares) where  $\nu < 0$ .

**The stop.** Called on  $(X, y)$  the path runs from  $\nu = \max_j \mu_j$  down to  $\nu = 0$  and stops there, because a LASSO solver takes no negative penalty. (If every  $\mu_j \leq 0$  it returns only the origin.) On the test problem that gives 17 corners, down to tilt 0.1415. The stopping point  $\nu_0 = \arg \min_{\nu \geq 0} \frac{1}{2} \nu^\top \Sigma \nu - \mu^\top \nu$  is the end of Section 5 with long-only weights. Writing  $\nu = s w$  with  $\mathbf{1}^\top w = 1$  and maximising over the scale  $s \geq 0$  first leaves  $\frac{1}{2} (\mu^\top w)^2 / w^\top \Sigma w$ , half the squared Sharpe ratio, so  $\nu_0 / \mathbf{1}^\top \nu_0$  is the long-only maximum-Sharpe portfolio, at tilt 0.1414. The frontier continues beyond it: the budget still binds while  $\nu$  turns negative (Figure 2), which a constraint allows and a penalty does not (Schmelzer & Hastie, 2026, Section 3.4). The stop is a property of the penalised form that LASSO solvers use.

**The shift.** Completing the square once more moves the penalty into the response:

$$\frac{1}{2} \|X\nu - y\|_2^2 + \nu \mathbf{1}^\top \nu = \frac{1}{2} \|X\nu - (y - \nu d)\|_2^2 + \text{const}, \quad X^\top d = \mathbf{1}. \quad (4)$$

If the nonnegative LASSO runs on  $y - \nu_0 d$  for some  $\nu_0 < 0$ , its penalty  $\nu' \geq 0$  acts as  $\nu = \nu_0 + \nu'$  on the original problem, and one path covers every  $\nu \geq \nu_0$ . For the Cholesky factor  $d = (X^\top)^{-1} \mathbf{1}$ ; for a tall  $X$  least squares returns the minimum-norm solution. `lars_path` reports  $\nu'$  divided by the number of rows, and it stops after 500 steps unless told otherwise, which a large problem needs:

```
d = np.linalg.lstsq(X.T, np.ones(len(mu)))[0] # solves X.T @ d == 1
alphas, _, V = lars_path(X, y - nu0 * d, method="lasso", positive=True,
                        max_iter=10 * len(mu))
nu = nu0 + len(y) * alphas # budget multiplier per column
corners = V[:, 1:-1] # drop the origin and the end
W, tilts = corners / corners.sum(0), 1 / corners.sum(0)
```

**Choosing  $\nu_0$ .** Any  $\nu_0$  below the last corner works, and the last corner is known in advance. Beyond it the portfolio holds the minimum-variance support  $S$ , which is the support of  $\arg \min_{u \geq 0} \|Xu - d\|_2$  (Lawson & Hanson, 1974). There  $\nu_S = a - \nu b$  with  $a = \Sigma_{SS}^{-1} \mu_S$  and  $b = \Sigma_{SS}^{-1} \mathbf{1}$ , and the segment stays optimal while  $\nu < a_i/b_i$  for  $b_i > 0$  and, for  $j \notin S$  with  $\Sigma_{jS} b > 1$ , while  $\nu \leq (\mu_j - \Sigma_{jS} a) / (1 - \Sigma_{jS} b)$ . The smallest bound is the last corner  $\nu_{\text{last}}$ :

```
S = nnls(X, d)[0] > 1e-9 # minimum-variance support
a = np.linalg.solve(Sigma[np.ix_(S, S)], mu[S])
b = np.linalg.solve(Sigma[np.ix_(S, S)], np.ones(S.sum()))
h = 1 - Sigma[np.ix_(~S, S)] @ b
g = mu[~S] - Sigma[np.ix_(~S, S)] @ a
nu_last = min((a[b > 0] / b[b > 0]).min(),
              (g[h < 0] / h[h < 0]).min(initial=np.inf))
nu0 = nu_last - max(abs(nu_last), np.abs(mu).max())
```

A fixed  $\nu_0$  would not work, because  $\nu_{\text{last}}$  varies widely: it is  $-4.95$  on the test problem,  $-11.9$  on one of the models of Table 1,  $-0.190$  on the stock data and  $-0.0118$  on the portfolio data of Section 8, where  $\mu$  is small. The margin  $\delta$  in  $\nu_0 = \nu_{\text{last}} - \delta \max(|\nu_{\text{last}}|, \|\mu\|_\infty)$  has to be large enough that the end of the path does not merge with the last corner, and small enough not to cost accuracy, which falls roughly in proportion to  $|\nu_0|$ . On the stock data  $\delta = 10^{-4}$  loses the last corner; from  $\delta = 10^{-2}$  to 1 all 116 are found with KKT residuals below  $5 \times 10^{-13}$ , and at  $\delta = 100$  the residual has grown to  $3 \times 10^{-11}$ . On the portfolio data even  $\delta = 10^{-2}$  loses the last corner, and  $\delta = 1$  finds all 12. The code takes  $\delta = 1$ . The minimum-variance portfolio itself is the limit  $\nu \rightarrow -\infty$ ; on the final segment it is  $w_S \propto \Sigma_{SS}^{-1} \mathbf{1}$ .

**What  $\nu$  means.** By (4) the point with multiplier  $\nu$  solves the same problem with  $\mu$  replaced by  $\mu - \nu \mathbf{1}$ , so it is the long-only maximum-Sharpe portfolio at risk-free rate  $r = \nu$ , Black's zero-beta rate (Black, 1972), a classical result. What the LASSO adds is that the rate is the path's own parameter, so for  $\nu_0 < r < \max_j \mu_j$  the maximum-Sharpe portfolio is an interpolation between the two columns that bracket  $r$ :

```
assert nu[-1] < r < nu[0] # the path covers the rate r
k = np.argmax(nu < r) # first column past r
s = (nu[k - 1] - r) / (nu[k - 1] - nu[k])
```

```

w_r = V[:, k - 1] + s * (V[:, k] - V[:, k - 1])
w_r /= w_r.sum() # max-Sharpe portfolio at r

```

**Weights versus frontier.** Between corners the weights are linear in the tilt  $\lambda$  (or, in Section 5, in the cap  $c$ ) and the variance is quadratic, so each piece of the frontier is an arc. Traced by the tilt  $\lambda$  at fixed constraints (Sections 6 and 7), the frontier has no kinks, since its slope  $dE/dV = 1/(2\lambda)$  varies continuously. The corners then show only in the weights (Figure 2). The long-short path of Section 5 is traced by the cap instead and does kink (Figure 1(a)).

**Algorithm 1: the long-only frontier from `lars_path`.**

1. Build  $X, y$  with  $X^T X = \Sigma$ ,  $X^T y = \mu$ , or use `lars_path_gram` with  $Xy = \mu - \nu_0 \mathbf{1}$ .
2. Solve  $X^T d = \mathbf{1}$ , compute  $\nu_{\text{last}}$  and set  $\nu_0$  as above.
3. Run the positive LASSO on  $y - \nu_0 d$  with `max_iter` well above  $n$ , and recover  $\nu = \nu_0 + n_{\text{rows}} \alpha$ .
4. Drop the first and last columns; each remaining  $v/\mathbf{1}^T v$  is a corner, at tilt  $1/\mathbf{1}^T v$ .
5. Take the minimum-variance end as  $w_S \propto \Sigma_S^{-1} \mathbf{1}$ , and any other point by interpolation, in  $1/\lambda$  for a tilt or in  $\nu$  for a rate.

## 8 Numerical evidence

Two kinds of evidence are reported. The first needs no second solver: each portfolio the path returns is checked against its own optimality conditions. For the cap with multiplier  $\kappa$  these are  $(\mu - \Sigma w)_j = \kappa \text{sign}(w_j)$  on the support and  $|(\mu - \Sigma w)_j| \leq \kappa$  off it; for the long-only problem at tilt  $\lambda$  they are  $(\Sigma w - \lambda \mu)_j = \gamma$  on the support and  $\geq \gamma$  off it, with  $\mathbf{1}^T w = 1$  and  $w \geq 0$ . The KKT residual is the worst violation, divided by  $\|\mu\|_\infty$  or by the size of the gradient. The second compares with baselines that are independent of the path. The long-short problems go to Clarabel, an interior-point solver, through `cvxpy`; the long-only problem goes to `quadprog`, the active-set method of Goldfarb and Idnani (1983).

*Long-short.* Every column has a KKT residual below  $2.6 \times 10^{-15}$ , and below  $2.9 \times 10^{-15}$  once rescaled to fixed leverage. At each column's cap Clarabel matches the path to  $3.0 \times 10^{-7}$  at the corners and  $1.4 \times 10^{-11}$  inside segments. The gap at the corners is the baseline's, since interior-point methods are least accurate where the holdings change. At fixed leverage the figures are  $6.2 \times 10^{-6}$  and  $1.9 \times 10^{-11}$ .

*Long only.* The shifted path finds 19 corners on the test problem, the 17 of a plain call and two more at tilts 0.1158 and 0.0128. They match the QP to  $6.0 \times 10^{-13}$ , points inside segments to  $7.9 \times 10^{-14}$ , the minimum-variance end to  $1.0 \times 10^{-16}$ , and the maximum-Sharpe portfolio at  $r = 0$  to  $3.4 \times 10^{-15}$ . The corners and the minimum-variance end have KKT residuals below  $3.6 \times 10^{-14}$ . Table 1 repeats both checks on other sizes and seeds. Where the two disagree, at  $n = 500$ , the residual shows that the gap of  $6.7 \times 10^{-11}$  is the QP's. The number of corners need not equal the number of assets, because some assets leave and re-enter.

**Table 1** Algorithm 1 on five-factor models, three seeds each: the worst KKT residual over the corners and the minimum-variance end, the largest  $\ell_\infty$  weight discrepancy against an independent active-set QP at the same tilt, and the same discrepancy at the minimum-variance end.

| $n$ | seeds | corners | $\nu_{\text{last}}$ | KKT residual          | worst corner          | min.-variance end     |
|-----|-------|---------|---------------------|-----------------------|-----------------------|-----------------------|
| 20  | 3     | 19      | −0.57 to 0.12       | $1.1 \times 10^{-14}$ | $8.6 \times 10^{-14}$ | $2.4 \times 10^{-16}$ |
| 50  | 3     | 49      | −11.9 to −0.74      | $1.0 \times 10^{-13}$ | $1.5 \times 10^{-13}$ | $1.7 \times 10^{-16}$ |
| 100 | 3     | 99–103  | −0.56 to −0.38      | $2.5 \times 10^{-14}$ | $1.1 \times 10^{-11}$ | $3.3 \times 10^{-16}$ |
| 500 | 3     | 503     | −0.45 to −0.26      | $2.1 \times 10^{-13}$ | $6.7 \times 10^{-11}$ | $4.3 \times 10^{-16}$ |

*Portfolio returns.* The daily value-weighted returns of the 100 portfolios that Fama and French form on size and book-to-market (French, n.d.), from January 2016 to December 2025, are free to download from their Data Library, and a script published with the code fetches them. One portfolio misses 250 days and is dropped, which leaves 99 portfolios over 2514 days. With the centred returns as  $X$  the last corner sits at  $\nu_{\text{last}} = -0.0118$ , eleven times  $\max_j |\mu_j|$  below zero. Algorithm 1 returns all 12 corners. They match the QP to  $2.1 \times 10^{-13}$  and have KKT residuals below  $3.9 \times 10^{-14}$ , and the minimum-variance portfolio,

which holds 7 of the portfolios, matches to  $1.6 \times 10^{-15}$ . Diversified portfolios are highly correlated, so the frontier turns only 12 times and a grid of QPs is cheap here.

*Stock returns.* The second data set tests the method at scale. It is simulated, so it needs no download and every rerun reproduces it: daily returns of 500 stocks over 1250 days, about five years, from a five-factor model with a market factor of 1% daily volatility, four style factors of 0.5%, idiosyncratic volatility between 1% and 3%, and expected returns around five basis points a day. Unlike the test problem,  $\mu$  and  $\Sigma$  are both estimated from the returns. With the centred returns as  $X$  the last corner sits at  $v_{\text{last}} = -0.190$ , about sixty times  $\max_j |\mu_j|$  below zero. Algorithm 1 returns all 116 corners in the time of about one QP solve, so a 100-point grid of QPs costs 75 to 95 times as much as the path and still misses the corners. Warm-starting narrows the gap without closing it: OSQP (Stellato et al., 2020), set up once so that its factorisation is reused, with each tilt changing only the linear term and each solve starting from the last, needs a sixth to a third of the cold grid's time at tolerance  $10^{-6}$  and agrees with the cold QP to  $3.7 \times 10^{-12}$  at every grid point. Absolute times depend on the machine and are not reported. Every corner has a KKT residual below  $4.7 \times 10^{-13}$  and matches its QP to  $8.6 \times 10^{-13}$ , and the minimum-variance portfolio, which holds 63 stocks, matches to  $5.8 \times 10^{-16}$ .

## 9 When to use it

The path beats a grid of QP solves, cold- or warm-started, on accuracy, since it finds the corners exactly. On speed it wins at scale, on the 500 simulated stocks; on a problem as small as the Fama–French portfolios its lead shrinks to a factor of three or four. LARS is in R and `scikit-learn`, the positive LASSO only in `scikit-learn`, so corner portfolios need no portfolio software, and one call shows that the knots of a LASSO path and the turning points of a frontier are the same points. It is the wrong tool when  $\Sigma$  is singular, where the identity is not known to survive (Schmelzer & Hastie, 2026, Section 7), or when the portfolio needs more than one linear constraint. The LASSO that `scikit-learn` solves has exactly one besides the signs,  $\|\beta\|_1 \leq t$ : the gross-exposure cap of Sections 5 and 6, or, with `positive=True`, the budget  $\mathbf{1}^\top w = 1$  of Section 7. The signs are either all free or all nonnegative. A dollar-neutral book adds the equality  $\mathbf{1}^\top \beta = 0$ , and a 130/30 book needs the budget and the gross cap at once. Path algorithms for a LASSO with general linear constraints exist (Gaines et al., 2018; James et al., 2020), but not in `scikit-learn`. The CLA takes such constraints and bounds on the weights directly, and `cvxcla` (Schmelzer, 2026) implements it. A risk budget  $w^\top \Sigma w \leq \sigma^2$  has no path of its own, but along each segment the variance is quadratic in the parameter, so the portfolio at a given  $\sigma$  comes from one scalar equation.

The numbers above are evidence for Theorem 1 and Proposition 3 of Schmelzer & Hastie (2026); the proofs are in that paper. Each check tests a prediction of one of the two results. Theorem 1 says the LASSO path is the frontier under a gross-exposure cap: every column meets the optimality conditions of its cap and matches Clarabel there. Proposition 3 says the nonnegative LASSO is the long-only frontier: every corner, and every point between corners, matches an active-set QP at the same tilt. Both make the frontier piecewise linear in its parameter, so a path finds its corners exactly where a grid of QP solves, cold- or warm-started, misses them. The checks were written by an author of Schmelzer & Hastie (2026), so they are not independent in the usual sense. The baseline solvers are, and the script and its tolerances are published so that anyone can rerun them. For practical use the note adds three details: the shift, a  $v_0$  computed in advance, and a `max_iter` large enough for big problems.

## Data and Code Availability

The software is in the directory `preprints/lasso/` of <https://github.com/Jebel-Quant/lasso>, under the MIT licence. `python/make_figures.py` regenerates every figure and number in the note; `uv run python/make_figures.py test` runs its checks. It is a single Python script that declares its own pinned dependencies, and `python/make_figures.py.lock` next to it fixes Python 3.12 and every package it pulls in, to the exact version and hash. The R snippet of Section 5 is `R/long_short.R`, with a check against `lars_path`.

The lockfile fixes the software, not the hardware. Residuals and gaps near  $10^{-13}$  come out with different last digits on another CPU or BLAS build, so a rerun will not match the printed values digit for digit. The tests therefore check them against tolerances between  $10^{-12}$  and  $10^{-9}$ . Costs are given only as ratios, which vary far less between machines than absolute times; `uv run python/make_figures.py` timings measures them.

The test problem, the factor models of Table 1 and the stock returns of Section 8 are simulated inside the script from fixed seeds, so they need no data. The portfolio returns of Section 8 come from Kenneth French's Data Library (French, n.d.) and, being copyrighted, are not redistributed. `python/fetch_french.py` downloads them and writes `data/french100_daily.csv`. The note used the CRSP vintage of August 2026, SHA-256 98298ef6...3c43. The library rebuilds its history each month from CRSP, so a later download can differ where CRSP revises a return.

## Competing interests

The author works at Jebel Quant Research, which hosts the code of this note, and maintains `cvxcla`, an open-source Python implementation of the Critical Line Algorithm. The note's method needs neither: it runs on `scikit-learn` or R alone. Section 9 points to `cvxcla` for portfolios with more constraints, and the companion script checks its corners against it. The author declares no other competing interests.

## Acknowledgements

I thank Trevor Hastie for many helpful discussions.

## References

- Black, F. (1972). Capital market equilibrium with restricted borrowing. *The Journal of Business*, 45(3), 444–455. <https://doi.org/10.1086/295472>
- Brodie, J., Daubechies, I., De Mol, C., Giannone, D., & Loris, I. (2009). Sparse and stable Markowitz portfolios. *Proceedings of the National Academy of Sciences*, 106(30), 12267–12272. <https://doi.org/10.1073/pnas.0904287106>
- Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, 32(2), 407–499. <https://doi.org/10.1214/009053604000000067>
- Fan, J., Zhang, J., & Yu, K. (2012). Vast Portfolio Selection With Gross-Exposure Constraints. *Journal of the American Statistical Association*, 107(498), 592–606. <https://doi.org/10.1080/01621459.2012.682825>
- French, K. R. (n.d.). *Data Library: 100 portfolios formed on size and book-to-market, daily*. Retrieved October 2026, from [https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data\\_library.html](https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html)
- Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1), 1–22. <https://doi.org/10.18637/jss.v033.i01>
- Gaines, B. R., Kim, J., & Zhou, H. (2018). Algorithms for Fitting the Constrained Lasso. *Journal of Computational and Graphical Statistics*, 27(4), 861–871. <https://doi.org/10.1080/10618600.2018.1473777>
- Goldfarb, D., & Idnani, A. (1983). A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming*, 27(1), 1–33. <https://doi.org/10.1007/bf02591962>
- Hastie, T., & Efron, B. (2022). *lars: least angle regression, lasso and forward stagewise* [R package version 1.3]. <https://doi.org/10.32614/CRAN.package.lars>
- James, G. M., Paulson, C., & Rusmevichientong, P. (2020). Penalized and Constrained Optimization: An Application to High-Dimensional Website Advertising. *Journal of the American Statistical Association*, 115(529), 107–122. <https://doi.org/10.1080/01621459.2019.1609970>
- Kelly, J. L., Jr. (1956). A New Interpretation of Information Rate. *Bell System Technical Journal*, 35(4), 917–926. <https://doi.org/10.1002/j.1538-7305.1956.tb03809.x>
- Lawson, C. L., & Hanson, R. J. (1974). *Solving Least Squares Problems*. Prentice-Hall.
- Markowitz, H. M. (1952). Portfolio selection. *The Journal of Finance*, 7(1), 77–91. <https://doi.org/10.1111/j.1540-6261.1952.tb01525.x>
- Markowitz, H. M. (1956). The optimization of a quadratic function subject to linear constraints. *Naval Research Logistics Quarterly*, 3(1–2), 111–133. <https://doi.org/10.1002/nav.3800030110>

- Osborne, M. R., Presnell, B., & Turlach, B. A. (2000). A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20(3), 389–403. <https://doi.org/10.1093/imanum/20.3.389>
- Pedregosa, F. et al. (2011). Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Schmelzer, T. (2026). *cvxcla: the Critical Line Algorithm with an operator interface in Python*. SSRN preprint 7583179. <https://doi.org/10.2139/ssrn.7583179>
- Schmelzer, T., & Hastie, T. (2026). *The Critical Line Algorithm and the Constrained LASSO: One Curve, Two Literatures*. arXiv. <https://doi.org/10.48550/ARXIV.2609.25704>
- Sharpe, W. F. (1964). Capital asset prices: a theory of market equilibrium under conditions of risk. *The Journal of Finance*, 19(3), 425–442. <https://doi.org/10.2307/2977928>
- Stellato, B., Banjac, G., Goulart, P., Bemporad, A., & Boyd, S. (2020). OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4), 637–672. <https://doi.org/10.1007/s12532-020-00179-2>
- Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1), 267–288. <https://doi.org/10.1111/j.2517-6161.1996.tb02080.x>
- Tobin, J. (1958). Liquidity preference as behavior towards risk. *The Review of Economic Studies*, 25(2), 65–86. <https://doi.org/10.2307/2296205>

---

COLOPHON Typeset with Typst 0.15.0 from the author's LaTeX source. Template jrepstat.typ at commit a79b6ff. Set in STIX Two Text, STIX Two Math and JetBrains Mono. Built 22 October 2026.