# What the algorithm does with each matrix

The algorithm is also a test. Feed it a matrix and how it terminates classifies the matrix — completion means positive definite, and each way of failing identifies a different reason.

**5**
CASES

## THE ALGORITHM COMPLETES — 2

### 1 Symmetric positive definite

all $\lambda > 0$

$A = LL^{T}$, all $\ell_{ii} > 0$

The intended case. Every square root is taken of a strictly positive number, the diagonal of $L$ comes out positive, and the factorization is unique. Half the cost of **LU** because symmetry means only one triangle is computed.

### 2 Near-singular but definite

**smallest $\lambda$ near zero**

runs; $\ell_{ii}$ very small

Completes without complaint while the accuracy quietly degrades — small pivots amplify rounding into $L$. The failure is silent, so the smallest diagonal entry is worth monitoring; adding $\varepsilon I$ regularizes at the cost of solving a slightly different problem.

## THE ALGORITHM HALTS — AND THE HALT IS THE ANSWER — 3

### 3 Positive semi-definite

some $\lambda = 0$

zero appears on the diagonal of $L$

Rank-deficient rather than wrong. Pivoted Cholesky, $P^{T}AP = LL^{T}$, handles it by reordering so the zeros arrive last — the factorization exists but is no longer unique.

### 4 Symmetric indefinite

some $\lambda < 0$

negative under the square root

No real $L$ exists, and the algorithm discovers this the moment it tries. Use $LDL^{T}$, which permits negative entries in $D$ and so handles any symmetric matrix — at the price of no longer testing definiteness.

### 5 Not symmetric

fails before it starts

$A = LL^{T}$ is impossible

Any product $LL^{T}$ is symmetric by construction, so a non-symmetric $A$ has no such factorization whatever its eigenvalues are. This is a precondition rather than an outcome — check symmetry first and use LU if it fails.

This is why Cholesky is the standard positive-definiteness test. Computing eigenvalues to check they are all positive costs $O(n^{3})$ with an iterative method and gives approximate answers; running Cholesky costs $\frac{1}{3}n^{3}$, terminates in finitely many steps, and answers exactly — with the factorization in hand as a byproduct when the answer is yes.