DECOMPOSITIONS · QR

Four routes to the same factorization

All four produce $A = QR$ and none is a variant of the others in output — they differ only in how they get there and how much accuracy survives the trip. The split below is by numerical standing, which is the column that decides in practice.

4

METHODS

ORTHOGONALIZATION — BUILD Q A COLUMN AT A TIME

2

1 Classical Gram–Schmidt

unstable — loses orthogonality

$$\mathbf{q}_j = \mathbf{a}_j - \sum_{i<j} \operatorname{proj}_{\mathbf{q}_i}(\mathbf{a}_j)$$

Each column is orthogonalized against the **original** earlier columns. In exact arithmetic this is correct; in floating point the accumulated rounding means $Q^T Q$ drifts measurably from I, and badly when the columns are nearly dependent. Worth knowing as the definition, not as a method.

2 Modified Gram–Schmidt

better, still not backward stable

subtract each projection immediately

The same arithmetic reassociated: update the remaining columns in place after each subtraction rather than accumulating. Substantially more accurate than the classical form for no extra cost, which is why it is the version to use if **Gram–Schmidt** is used at all.

TRIANGULARIZATION — ZERO OUT A INSTEAD

2

3 Householder reflections

backward stable — the default

$$H = I - 2\mathbf{v}\mathbf{v}^T / \mathbf{v}^T\mathbf{v}$$

One reflection per column zeros everything below the pivot at once. Q arrives as a product of reflections rather than being built directly, which is what keeps it orthogonal to machine precision. This is what a numerical library calls unless told otherwise.

4 Givens rotations

backward stable; local

a plane rotation per entry

Zeros one entry at a time instead of a whole column. Slower on a dense matrix, but each rotation touches only two rows — so a sparse matrix stays sparse, and the work can be parallelised. The reason to choose it is structure, not speed.

Two of these are used and two are taught. Gram–Schmidt makes the connection between orthogonalization and QR visible, which is why it appears in every course; Householder and Givens are what libraries actually call, because backward stability is the property that matters once the matrix is anything other than well behaved.