II.2 Least Squares : Four Ways

Many applications lead to **unsolvable linear equations** $Ax = b$. It is ironic that this is such an important problem in linear algebra. We can't throw equations away, we need to produce a best solution $\widehat{x}$. **The least squares method chooses $\widehat{x}$ to make $||b - A\widehat{x}||^2$ as small as possible.** Minimizing that error means that its derivatives are zero : those are the *normal equations* $A^{\mathrm{T}}A\widehat{x} = A^{\mathrm{T}}b$. Their geometry will be in Figure II.2.

This section explains four ways to solve those important (and solvable !) equations :

1 The SVD of A leads to its **pseudoinverse** A^+. Then $\widehat{x} = A^+b$: One short formula.

2 $A^{\mathrm{T}}A\widehat{x} = A^{\mathrm{T}}b$ can be solved directly when A has **independent columns.**

3 The Gram-Schmidt idea produces **orthogonal columns** in Q. Then $A = QR$.

4 **Minimize** $||b - Ax||^2 + \delta^2||x||^2$. That penalty changes the normal equations to $(A^{\mathrm{T}}A + \delta^2 I)x_\delta = A^{\mathrm{T}}b$. Now the matrix is invertible and x_δ goes to $\widehat{x}$ as $\delta \to 0$.

$A^{\mathrm{T}}A$ has an attractive symmetry. But its size may be a problem. And its condition number—measuring the danger of unacceptable roundoff error—is the *square* of the condition number of A. In well-posed problems of moderate size we go ahead to solve the least squares equation $A^{\mathrm{T}}A\widehat{x} = A^{\mathrm{T}}b$, but in large or ill-posed problems we find another way.

We could orthogonalize the columns of A. We could use its SVD. For really large problems we sample the column space of A by simply multiplying Av for **random vectors** v. This seems to be the future for very big computations : a high probability of success.

First of all, please allow us to emphasize the importance of $A^{\mathrm{T}}A$ and $A^{\mathrm{T}}CA$. That matrix C is often a positive diagonal matrix. It gives stiffnesses or conductances or edge capacities or inverse variances $1/\sigma^2$—the constants from science or engineering or statistics that define our particular problem : the "weights" in weighted least squares.

Here is a sample of the appearances of $A^{\mathrm{T}}A$ and $A^{\mathrm{T}}CA$ in applied mathematics :

In mechanical engineering, $A^{\mathrm{T}}A$ (or $A^{\mathrm{T}}CA$) is the **stiffness matrix**

In circuit theory, $A^{\mathrm{T}}A$ (or $A^{\mathrm{T}}CA$) is the **conductance matrix**

In graph theory, $A^{\mathrm{T}}A$ (or $A^{\mathrm{T}}CA$) is the (weighted) **graph Laplacian**

In mathematics, $A^{\mathrm{T}}A$ is the **Gram matrix** : inner products of columns of A

In large problems, $A^{\mathrm{T}}A$ is expensive and often dangerous to compute. We avoid it if we can ! The Gram-Schmidt way replaces A by QR (orthogonal Q, triangular R). Then $A^{\mathrm{T}}A$ is the same as $R^{\mathrm{T}}Q^{\mathrm{T}}QR = R^{\mathrm{T}}R$. And the fundamental equation $A^{\mathrm{T}}A\widehat{x} = A^{\mathrm{T}}b$ becomes $R^{\mathrm{T}}R\widehat{x} = R^{\mathrm{T}}Q^{\mathrm{T}}b$. Finally this is $R\widehat{x} = Q^{\mathrm{T}}b$, safe to solve and fast too.

Thus $A^{\mathrm{T}}A$ and $A^{\mathrm{T}}CA$ are crucial matrices—but paradoxically, we try not to compute them. Orthogonal matrices and triangular matrices : Those are the good ones.

A^+ is the Pseudoinverse of A

I will first describe the pseudoinverse A^+ in words. If A is invertible then A^+ is A^{-1}. If A is m by n then A^+ is n by m. When A multiplies a vector $\boldsymbol{x}$ in its row space, this produces $A\boldsymbol{x}$ in the column space. Those two spaces have equal dimension r (the rank). Restricted to these spaces A is always invertible—and A^+ inverts A. **Thus $A^+A\boldsymbol{x} = \boldsymbol{x}$ exactly when $\boldsymbol{x}$ is in the row space.** And $AA^+\boldsymbol{b} = \boldsymbol{b}$ when $\boldsymbol{b}$ is in the column space.

The nullspace of A^+ is the nullspace of A^{T}. It contains the vectors $\boldsymbol{y}$ in $\mathbf{R}^m$ with $A^{\mathrm{T}}\boldsymbol{y} = \mathbf{0}$. Those vectors $\boldsymbol{y}$ are perpendicular to every $A\boldsymbol{x}$ in the column space. For these $\boldsymbol{y}$, we accept $\boldsymbol{x}^+ = A^+\boldsymbol{y} = \mathbf{0}$ as the best solution to the unsolvable equation $A\boldsymbol{x} = \boldsymbol{y}$. Altogether **$A^+$ inverts A where that is possible :**

$$\textbf{The pseudoinverse of } A = \begin{bmatrix} 2 & 0 \\ 0 & \mathbf{0} \end{bmatrix} \textbf{ is } A^+ = \begin{bmatrix} 1/2 & 0 \\ 0 & \mathbf{0} \end{bmatrix}.$$

The whole point is to produce a suitable "pseudoinverse" when A has no inverse.

Rule 1 If A has independent columns, then $A^+ = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}$ and so $A^+A = I$.
Rule 2 If A has independent rows, then $A^+ = A^{\mathrm{T}}(AA^{\mathrm{T}})^{-1}$ and so $AA^+ = I$.
Rule 3 A diagonal matrix Σ is inverted where possible—otherwise Σ^+ has zeros :

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \Sigma^+ = \begin{bmatrix} 1/\sigma_1 & 0 & 0 \\ 0 & 1/\sigma_2 & 0 \\ 0 & 0 & \mathbf{0} \\ 0 & 0 & 0 \end{bmatrix}$$

On the four subspaces
$\Sigma^+\Sigma = I \quad \Sigma\Sigma^+ = I$
$\Sigma^+\Sigma = \mathbf{0} \quad \Sigma\Sigma^+ = \mathbf{0}$

All matrices $\boxed{\textbf{The pseudoinverse of } A = U\Sigma V^{\mathrm{T}} \textbf{ is } A^+ = V\Sigma^+U^{\mathrm{T}}.}$ (1)

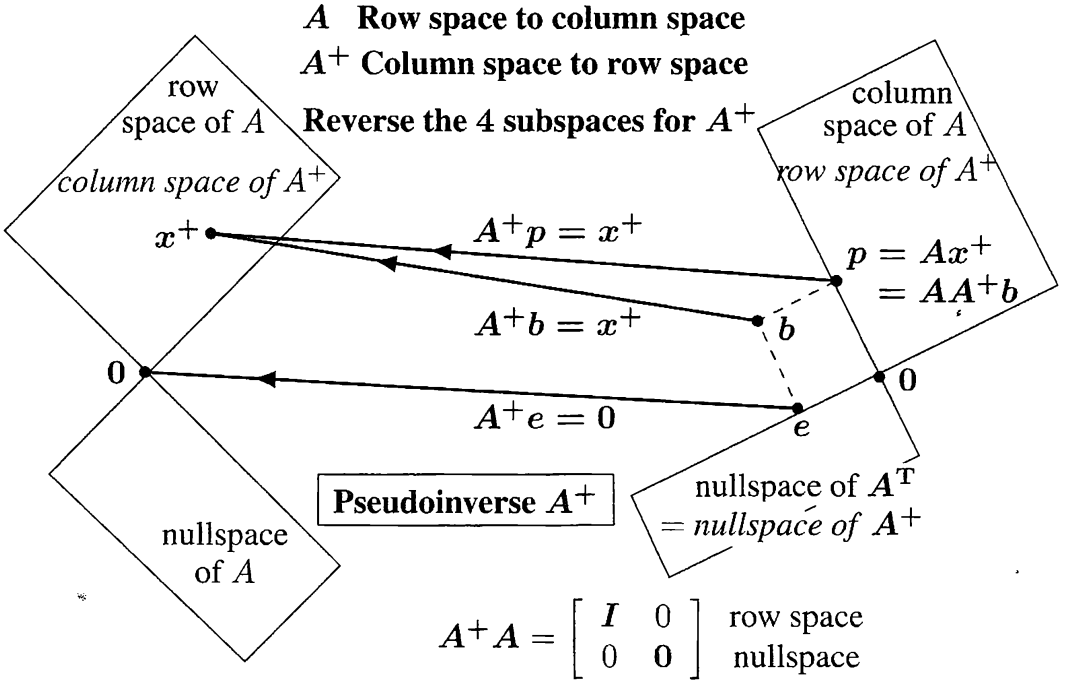

Figure II.1: Vectors $\boldsymbol{p} = A\boldsymbol{x}^+$ in the column space of A go back to $\boldsymbol{x}^+$ in the row space.

This pseudoinverse A^+ (*sometimes written* $A^\dagger$ with a dagger instead of a plus sign) solves the least squares equation $A^TA\widehat{x} = A^Tb$ in one step. This page verifies that $x^+ = A^+b = V\Sigma^+U^Tb$ is best possible. At the end of this section, we look in more detail at A^+.

Question: The formula $A^+ = V\Sigma^+U^T$ uses the SVD. Is the SVD essential to find A^+? *Answer*: No, A^+ could also be computed directly from A by modifying the elimination steps that usually produce A^{-1}. However each step of arithmetic would have to be exact! You need to distinguish exact zeros from small nonzeros. That is the hard part of A^+.

The Least Squares Solution to $Ax = b$ is $x^+ = A^+b$

I have written x^+ instead of $\widehat{x}$ because the vector x^+ has two properties:

1 $x = x^+ = A^+b$ makes $\|\|b - Ax\|\|^2$ as small as possible.	**Least squares solution**
2 If another $\widehat{x}$ achieves that minimum then $\|\|x^+\\| < \|\|\widehat{x}\|\|$.	**Minimum norm solution**

$x^+ = A^+b$ is the minimum norm least squares solution. When A has independent columns and rank $r = n$, this is the only least squares solution. But if there are nonzero vectors x in the nullspace of A (so $r < n$), they can be added to x^+. The error $b - A(x^+ + x)$ is not affected when $Ax = 0$. But the length $\|\|x^+ + x\|\|^2$ will grow to $\|\|x^+\|\|^2 + \|\|x\|\|^2$. Those pieces are orthogonal: Row space $\perp$ nullspace.

So the minimum norm (shortest) solution of $A^TA\widehat{x} = A^Tb$ is $x^+ = A^+b$, **x^+ has a zero component in the nullspace of A.**

Example 1 The shortest least squares solution to $\begin{bmatrix} 3 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 8 \end{bmatrix}$ is x^+

$x^+ = A^+b = \begin{bmatrix} 1/3 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 6 \\ 8 \end{bmatrix} = \begin{bmatrix} \mathbf{2} \\ \mathbf{0} \end{bmatrix}$. All vectors $\begin{bmatrix} \mathbf{0} \\ \boldsymbol{x_2} \end{bmatrix}$ are in the nullspace of A.

All the vectors $\widehat{x} = \begin{bmatrix} \mathbf{2} \\ \boldsymbol{x_2} \end{bmatrix}$ minimize $\|\|b - A\widehat{x}\|\|^2 = 64$. But $x^+ = \begin{bmatrix} \mathbf{2} \\ \mathbf{0} \end{bmatrix}$ is shortest.

That example shows the least squares solutions when A is a diagonal matrix like Σ. To allow every matrix $U\Sigma V^T$, we have to account for the orthogonal matrices U and V. We can freely multiply by U^T without changing any lengths, because $U^TU = I$:

$$\textbf{Squared error} \qquad \|\|b - Ax\|\|^2 = \|\|b - U\Sigma V^Tx\|\|^2 = \|\|U^Tb - \Sigma V^Tx\|\|^2. \qquad (2)$$

Set $w = V^Tx$ to get $\|\|U^Tb - \Sigma w\|\|^2$. **The best w is Σ^+U^Tb.** And finally x^+ is A^+b:

$$w = V^Tx^+ = \Sigma^+U^Tb \text{ and } V^T = V^{-1} \text{ lead to } x^+ = \boldsymbol{V\Sigma^+U^Tb = A^+b.} \qquad (3)$$

The SVD solved the least squares problem in one step A^+b. The only question is the computational cost. Singular values and singular vectors cost more than elimination. The next two proposed solutions work directly with the linear equations $A^TA\widehat{x} = A^Tb$. This succeeds when A^TA is invertible—and then $\widehat{x}$ is the same as x^+.

When is A^TA Invertible ?

The invertibility (or not) of the matrix A^TA is an important question with a nice answer:

A^TA is invertible exactly when A has independent columns. If $Ax = 0$ then $x = 0$

Always A and A^TA have the same nullspace ! This is because $A^TAx = 0$ always leads to $x^TA^TAx = 0$. This is $||Ax||^2 = 0$. Then $Ax = 0$ and x is in $\mathbf{N}(A)$. For all matrices:

$$\mathbf{N}(A^TA) = \mathbf{N}(A) \text{ and } \mathbf{C}(AA^T) = \mathbf{C}(A) \text{ and } \mathbf{rank}\,(A^TA) = \mathbf{rank}\,(AA^T) = \mathbf{rank}\,(A)$$

We now go forward when A^TA is invertible to solve the normal equations $A^TA\widehat{x} = A^Tb$.

The Normal Equations $A^TA\widehat{x} = A^Tb$

Figure II.2 shows a picture of the least squares problem and its solution. The problem is that b is not in the column space of A, so $Ax = b$ has no solution. The best vector $p = A\widehat{x}$ is a projection. **We project b onto the column space of A.** The vectors $\widehat{x}$ and $p = A\widehat{x}$ come from solving a famous system of linear equations: $A^TA\widehat{x} = A^Tb$. To invert A^TA, we need to know that A has independent columns.

The picture shows the all-important right triangle with sides b, p, and e.

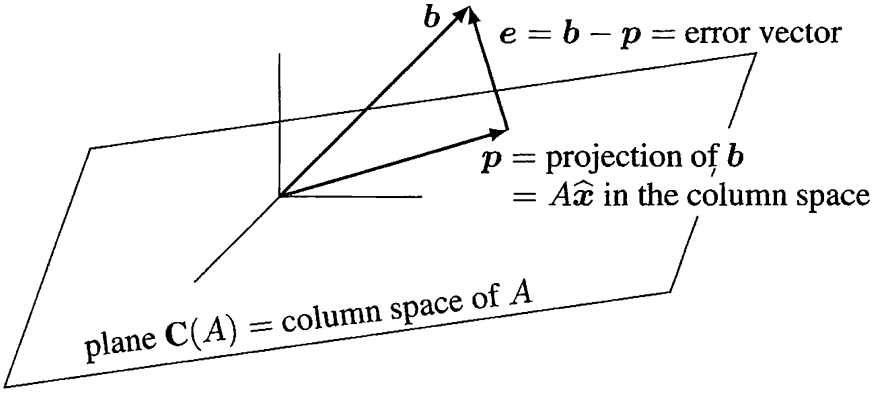

Figure II.2: The projection $p = A\widehat{x}$ is the point in the column space that is closest to b.

Everybody understands that e is perpendicular to the plane (the column space of A). This says that $b - p = b - A\widehat{x}$ is perpendicular to all vectors Ax in the column space:

$$(Ax)^T(b - A\widehat{x}) = x^TA^T(b - A\widehat{x}) = 0 \text{ for all } x \text{ forces } A^T(b - A\widehat{x}) = \mathbf{0}. \quad (4)$$

Everything comes from that last equation, when we write it as $A^TA\,\widehat{x} = A^Tb$.

Normal equation for $\widehat{x}$	$A^TA\,\widehat{x} = A^Tb$	(5)
Least squares solution to $Ax = b$	$\widehat{x} = (A^TA)^{-1}A^Tb$	(6)
Projection of b onto the column space of A	$p = A\,\widehat{x} = A(A^TA)^{-1}A^Tb$	(7)
Projection matrix that multiplies b to give p	$P = A(A^TA)^{-1}A^T$	(8)

A now has independent columns: $r = n$. That makes $A^{\mathrm{T}}A$ positive definite and invertible. We could check that our $\widehat{x}$ is the same vector $x^+ = A^+b$ that came from the pseudoinverse. There are no other $\widehat{x}$'s because the rank is assumed to be $r = n$. The nullspace of A only contains the zero vector.

Projection matrices have the special property $\boldsymbol{P^2 = P}$. When we project a second time, the projection p stays exactly the same. Use equation (8) for P:

$$P^2 = A\,(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}A\,(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}} = A\,(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}} = P. \tag{9}$$

The Third Way to Compute $\widehat{x}$: Gram-Schmidt

The columns of A are still assumed to be independent: $r = n$. But they are not assumed to be orthogonal! Then $A^{\mathrm{T}}A$ is not a diagonal matrix and solving $A^{\mathrm{T}}A\widehat{x} = A^{\mathrm{T}}b$ needs work. Our third approach **orthogonalizes the columns of A**, and then $\widehat{x}$ is easy to find.

You could say: The work is now in producing orthogonal (even orthonormal) columns. Exactly true. The operation count is actually doubled compared to $A^{\mathrm{T}}A\widehat{x} = A^{\mathrm{T}}b$, but orthogonal vectors provide numerical stability. Stability becomes important when $A^{\mathrm{T}}A$ is nearly singular. The **condition number** of $A^{\mathrm{T}}A$ is its norm $||A^{\mathrm{T}}A||$ times $||(A^{\mathrm{T}}A)^{-1}||$. When this number σ_1^2/σ_n^2 is large, it is wise to orthogonalize the columns of A in advance. Then work with an orthogonal matrix Q.

The condition number of Q is $||Q||$ times $||Q^{-1}||$. *Those norms equal* 1*: best possible.*

Here is the famous Gram-Schmidt idea, starting with A and ending with Q. Independent columns $a_1, \ldots, a_n$ lead to orthonormal $q_1, \ldots, q_n$. This is a fundamental computation in linear algebra. The first step is $q_1 = a_1/||a_1||$. That is a unit vector: $||q_1|| = 1$. Then subtract from a_2 its component in the q_1 direction:

Gram-Schmidt step			
	Orthogonalize	$A_2 = a_2 - (a_2^{\mathrm{T}}\,q_1)\,q_1$	(10)
	Normalize	$q_2 = A_2/||A_2||$	(11)

Subtracting that component $(a_2^{\mathrm{T}}\,q_1)\,q_1$ produced the vector A_2 orthogonal to q_1:

$$(a_2 - (a_2^{\mathrm{T}}\,q_1)\,q_1)^{\mathrm{T}}q_1 = a_2^{\mathrm{T}}\,q_1 - a_2^{\mathrm{T}}\,q_1 = 0 \text{ since } q_1^{\mathrm{T}}\,q_1 = 1.$$

The algorithm goes onward to a_3 and A_3 and q_3, normalizing each time to make $||q|| = 1$. Subtracting the components of a_3 along q_1 and q_2 leaves A_3:

$$\textbf{Orthogonalize}\quad A_3 = a_3 - (a_3^{\mathrm{T}}\,q_1)\,q_1 - (a_3^{\mathrm{T}}\,q_2)\,q_2 \qquad \textbf{Normalize}\quad q_3 = \frac{A_3}{||A_3||} \tag{12}$$

$$A_3^{\mathrm{T}}q_1 = A_3^{\mathrm{T}}q_2 = 0 \quad \text{and} \quad ||q_3|| = 1$$

Each q_k is a combination of a_1 to a_k. Then each a_k is a combination of q_1 to q_k.

$$\textbf{a's from q's} \qquad \begin{aligned} a_1 &= ||a_1||\,q_1 \\ a_2 &= (a_2^{\mathrm{T}}\,q_1)\,q_1 + ||A_2||\,q_2 \\ a_3 &= (a_3^{\mathrm{T}}\,q_1)\,q_1 + (a_3^{\mathrm{T}}\,q_2)\,q_2 + ||A_3||\,q_3 \end{aligned} \tag{13}$$

Those equations tell us that **the matrix $R = Q^{\mathrm{T}}A$ with $r_{ij} = q_i^{\mathrm{T}}a_j$ is upper triangular:**

$$\begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix} \text{ is } A = QR. \quad (14)$$

Gram-Schmidt produces orthonormal q's from independent a's. Then $A = QR$.

If $A = QR$ then $R = Q^{\mathrm{T}}A =$ inner products of q's with a's! Later a's are not involved in earlier q's, so R is triangular. And certainly $A^{\mathrm{T}}A = R^{\mathrm{T}}Q^{\mathrm{T}}QR = R^{\mathrm{T}}R$:

The least squares solution to $Ax = b$ is $\widehat{x} = R^{-1}Q^{\mathrm{T}}b$.

The MATLAB command is $[Q,\ R] = qr(A)$. Every $r_{ij} = q_i^{\mathrm{T}}a_j$ because $R = Q^{\mathrm{T}}A$. The vector $\widehat{x} = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b$ is $(R^{\mathrm{T}}R)^{-1}R^{\mathrm{T}}Q^{\mathrm{T}}b$. This is exactly $\widehat{x} = R^{-1}Q^{\mathrm{T}}b$.

Gram-Schmidt with Column Pivoting

That straightforward description of Gram-Schmidt worked with the columns of A in their original order $a_1, a_2, a_3, \ldots$ This could be dangerous! We could never live with a code for elimination that didn't allow row exchanges. Then roundoff error could wipe us out.

Similarly, each step of Gram-Schmidt should begin with a new column that is as independent as possible of the columns already processed. **We need column exchanges to pick the largest remaining column. Change the order of columns as we go.**

To choose correctly from the remaining columns of A, we make a simple change in Gram-Schmidt:

Old Accept column a_j as next. Subtract its components in the directions q_1 to q_{j-1}

New When q_{j-1} is found, subtract the q_{j-1} component from **all remaining columns**

This might look like more work, but it's not. Sooner or later we had to remove $(a_i^{\mathrm{T}}q_{j-1})q_{j-1}$ from each remaining column a_i. Now we do it sooner—as soon as we know q_{j-1}. Then we have a free choice of the next column to work with, and we choose the largest.

Elimination Row exchanges on A left us with $PA = LU$ (permutation matrix P)
Gram-Schmidt Column exchanges leave us with $AP = QR$ (permutation matrix P)

Here is the situation after $j-1$ Gram-Schmidt steps with column pivoting. We have $j-1$ orthogonal unit vectors q_1 to q_{j-1} in the columns of a matrix Q_{j-1}. We have the square matrix R_{j-1} that combines those columns of Q_{j-1} to produce $j-1$ columns of A. *They might not be the* first $j-1$ *columns of* A—we are optimizing the column order. All the remaining columns of A have been *orthogonalized* against the vectors q_1 to q_{j-1}.

Step j. Choose the largest of the remaining columns of A. Normalize it to length 1.

This is q_j. Then from each of the $n-j$ vectors still waiting to be chosen, subtract the component in the direction of this latest q_j. Ready now for step $j+1$.

We will follow Gunnar Martinsson's 2016 course notes for APPM 5720, to express step j in pseudocode. The original A is A_0 and the matrices Q_0 and R_0 are empty.

Step j is the following loop, which starts with A_{j-1} and ends at A_j. The code stops after j reaches $\min(m, n)$. **Here is column pivoting in Gram-Schmidt**:

i = argmax $||A_{j-1}(:, \ell)||$ finds the largest column not yet chosen for the basis

$\boldsymbol{q}_j = A_{j-1}(:, i)/||A_{j-1}(:, i)||$ normalizes that column to give the new unit vector $\boldsymbol{q}_j$

$Q_j = \begin{bmatrix} Q_{j-1} & \boldsymbol{q}_j \end{bmatrix}$ updates Q_{j-1} with the new orthogonal unit vector $\boldsymbol{q}_j$

$\boldsymbol{r}_j = \boldsymbol{q}_j^{\mathrm{T}} A_{j-1}$ finds the row of inner products of $\boldsymbol{q}_j$ with remaining columns of $\boldsymbol{A}$

$R_j = \begin{bmatrix} R_{j-1} \\ \boldsymbol{r}_j \end{bmatrix}$ updates R_{j-1} with the new row of inner products

$A_j = A_{j-1} - \boldsymbol{q}_j \boldsymbol{r}_j$ subtracts the new rank-one piece from each column to give A_j

When this loop ends, we have Q and R and $A = QR$. This R is a *permutation* of an upper triangular matrix. (It will be upper triangular if the largest columns in Step 1 come first, so each $i = j$.) The actual output can be an upper triangular matrix plus a vector with the numbers $1, \ldots, n$ in the permutation order we need to know, to construct R.

In practice, this QR algorithm with pivoting is made safer by *reorthonormalizing*:

$$\boldsymbol{q}_j = \boldsymbol{q}_j - Q_{j-1}(Q_{j-1}^{\mathrm{T}} \boldsymbol{q}_j)$$

$$\boldsymbol{q}_j = \boldsymbol{q}_j \,/\, ||\boldsymbol{q}_j|| \quad \textbf{(to make sure!)}$$

There is a similar reordering for "QR with Householder matrices" to reduce roundoff error. You have seen the essential point of pivoting: **good columns come first.**

Question: Both Q from Gram-Schmidt and U from the SVD contain an orthonormal basis for the column space C(A). *Are they likely to be the same basis*?

Answer: No, they are not the same. The columns of U are eigenvectors of AA^{T}. You cannot find eigenvectors (or eigenvalues) in a finite number of exact "arithmetic" steps for matrices of size $n > 4$. The equation $\det(A - \lambda I) = 0$ will be 5th degree or higher: *No formula can exist for the roots λ of a 5th degree equation (Abel).* Gram-Schmidt just requires inner products and square roots so Q must be different from U.

In the past, computing a nearly accurate eigenvalue took a much larger multiple of n^3 floating-point operations than elimination or Gram-Schmidt. That is not true now.