

Lecture 6, Sci. Comp. for DPhil Students

Nick Trefethen, Thursday 31.10.19

Today

- II.3 QR factorization
- II.4 Computation of the QR factorization
- II.5 Linear least-squares

Handouts

- Quiz 4
- Householder's 4-page paper from 1958 on QR factorization
- m09_colorquiz.m, m10_QRfact.m, m11_leastsquares3.m

Announcements

- Assignment 2 is due on Tuesday.
- Read: Trefethen & Bau, chapters 7 & 11
- Please do Quiz 4

Just for fun, let's look at a game I had fun writing in MATLAB: `m09_colorquiz.m`

In MATLAB, colour images are processed beautifully via linear algebra. There's an Image Processing Toolbox.

II.3 QR factorisation

Recall from Tuesday: we often work with a matrix Q with ON columns. In the rectangular case there is no name for such matrices. Note that if Q is $m \times n$ with $m > n$ then

$$Q^T Q = I_n \quad \text{but} \quad Q Q^T \neq I_m$$

(illustrate with figures).

Let A be an matrix with n columns. A (reduced) **QR factorization** of A is a factorisation

$$\begin{pmatrix} | & | & | & | \\ | & | & | & | \\ | & | & | & | \\ | & | & | & | \\ | & | & | & | \\ | & | & | & | \\ | & | & | & | \end{pmatrix} = \begin{pmatrix} | & | & | & | \\ | & | & | & | \\ | & | & | & | \\ | & | & | & | \\ | & | & | & | \\ | & | & | & | \\ | & | & | & | \end{pmatrix} \begin{pmatrix} r_1 & r_{11} & \dots & r_{1n} \\ & r_{12} & & \\ & & r_{22} & : \\ & & & . \\ & & & . \\ & & & r_{nn} \end{pmatrix}$$

is a linear combination of the columns of A . If $A = QR$, then

$$b = QRx = Q(Rx)$$

or by multiplying on the left by Q^T , $Rx = Q^T b$.

Thus multiplying on the left by R takes us from a vector of coefficients for an expansion in columns of A to a vector of coefficients for an expansion in columns of Q . We talked about such things last lecture in the square case.

II.4 Computation of the QR factorisation

Given A , there are two approaches to computing Q and R such that $A = QR$.

(1) Gram-Schmidt orthogonalisation: triangular orthogonalisation

Construct vectors q_1, q_2, \dots in succession.

This amounts to multiplying A on the right by successive upper-triangular matrices until it has orthonormal columns:

$$AR_1R_2R_3\cdots = Q$$

“*triangular orthogonalisation*”: aim for Q and get R as a by-product.

Gram-Schmidt is a classical idea of great importance. On the other hand, as a linear algebra algorithm it is often not as good as the more accurate alternative:

(2) Householder triangularisation: orthogonal triangularisation

[Householder 1958 paper]

Construct columns r_1, r_2, \dots of R in succession.

This amounts to multiplying A on the left by successive orthogonal matrices until it has upper-triangular structure:

$$\cdots Q_3Q_2Q_1A = R \quad \text{expand on this a bit}$$

“*orthogonal triangularisation*”: aim for R and get Q as a by-product.

With rounding errors, this algorithm produces matrices Q whose columns are much more nearly orthogonal than with Gram-Schmidt.

For a one-off application like least-squares, this often does not matter.

It sometimes matters a good deal, however, for applications involving repeated calculation of QR factorisations, such as computing eigenvalues or singular values.

[m10_QRfact.m]

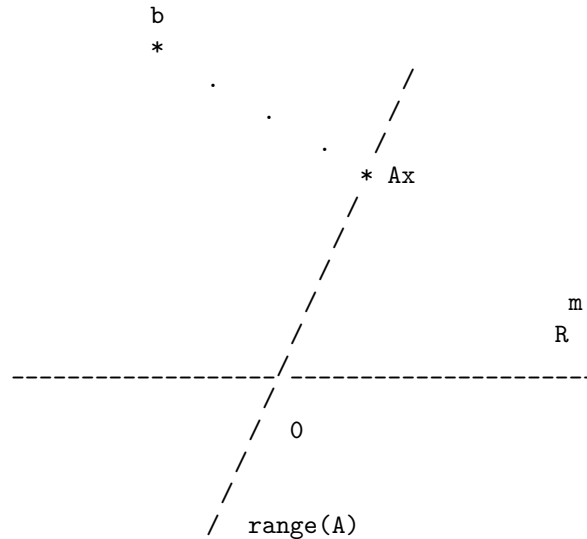
II.5 Linear least-squares

Let's return to the least-squares problem. These ideas were invented independently by Gauss and Legendre before 1800, leading to a big fight.

An expert in this subject is *Åke Björck* – see his book in our list.

Suppose A is $m \times n$ with $m > n$. A **least-squares solution** to $Ax = b$ is a vector x with minimal residual norm, i.e.,

$$\|Ax - b\| = \text{minimum} \quad (*)$$



It's clear geometrically, and easily proved, that $(*)$ is equivalent to the condition that the **residual** $b - Ax$ is orthogonal to $\text{range}(A)$.

That is, for each column a_j of A , $a_j^T(b - Ax) = 0$. Equivalently,

$$A^T(b - Ax) = 0, \quad \text{i.e.,} \quad A^T Ax = A^T b.$$

These are the **normal equations**.

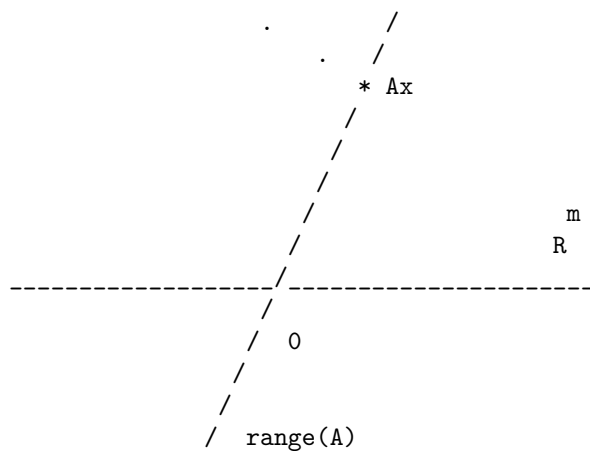
Theorem. $\text{rank}(A) = n \Leftrightarrow A^T A \text{ is SPD} \Leftrightarrow \text{solution to } (*) \text{ is unique}$

The normal equations are invaluable theoretically. They are also useful computationally. However, there is a more accurate “standard” method for least-squares problems of moderate size:

Least-squares via QR factorisation

Suppose $A = QR$ and we want to solve $Ax \approx b$.





Now Ax is the orthogonal projection of b onto $\text{range}(A)$. Thus it can be written

$$Ax = Q Q^T b$$

\swarrow \searrow
 orthogonal expansion
 basis coeffs

($Q Q^T$ is called a **projection matrix**.)

Multiplying on the left by Q^T gives $Q^T Q R x = Q^T b$, i.e.,

$$R x = Q^T b.$$

You can solve this system to get the least-squares solution vector x .

It's a triangular system – easily solved by back-substitution successively for x_n, x_{n-1}, \dots, x_1 .

[m11_leastquares3.m]