RREF and all that. Codes/SKD/2019

What is RREF? A matrix is in RREF if:

1. The first non-zero entry in any row is 1. We call these 'leading 1s' (or 'pivots').
2. If a column contains a leading 1, then every other entry in that column is 0.
3. Each leading 1 is to the right of any leading 1 in a higher row.
4. Any row of zeros is below all the rows with non-zero entries.

So, typically, a matrix in RREF looks something like this:

$$\begin{pmatrix} 1 & * & 0 & 0 & * & * & 0 & * \\ 0 & 0 & 1 & 0 & * & * & 0 & * \\ 0 & 0 & 0 & 1 & * & * & 0 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$, where the $*$s can be anything (including 0 or 1).

A matrix in RREF could also have some extra columns of 0s at the left. (If it is a 'portrait' matrix, it *must* have one or more rows of 0s at the bottom.)

How do we get there? To "put a matrix into RREF" we use elementary row operations - EROs. These are:

$P_{i,j}$: Swap row i with row j. Using these we can permute the rows however we wish.

M_λ: Multiply a row by some $\lambda \neq 0$ from the field we are working in.

$A_{i,j}(\lambda)$: Add $\lambda\times$row i to row j.

Although we never used this term in the Codes module, for the rest of this document I will say that two matrices M_1 and M_2 are *row-equivalent* if you can turn one into the other using only EROs.

Is RREF unique? Given a matrix, there are usually many different sequences of EROs you could use to get it into RREF. (That's why it's such a pain to mark these questions, and it helps if people write down the P, M or A for the EROs they use!)

But the final RREF matrix is unique. In other words, any matrix is row-equivalent to exactly one matrix in RREF. This is useful if you want to see whether two matrices are row-equivalent: just find the RREF for each of them. If it's the same, they are row-equivalent; if not, not.

Are there other, related forms for a matrix? Yes.

Echelon form This requires:

1. The first non-zero entry in any row is to the right of the first non-zero entry in any higher row.
2. Any row of zeros is below all the rows with non-zero entries.

So, typically, a matrix in echelon form looks something like this:

$$\begin{pmatrix} \dagger & * & * & * & * & * & * & * \\ 0 & 0 & \dagger & * & * & * & * & * \\ 0 & 0 & 0 & \dagger & * & * & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & \dagger & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \text{ where } *\text{s can be anything, but } \dagger\text{s are non-zero.}$$

Clearly, RREF $\Rightarrow$ echelon form, but not the other way. Echelon form is similar to upper-triangular form, but makes sense also for non-square matrices. When we use matrices to solve systems of linear equations, we can choose whether to go all the way to RREF, or stop at echelon form, before 'substituting back'.

Standard Form $(I|A)$ Clearly, this standard form $\Rightarrow$ RREF, but not the other way. It follows from the uniqueness of RREF that, if matrix M has RREF which is *not* in standard form, then M is not row-equivalent to any matrix in standard form.

What do R, R, E and F stand for, anyway? It doesn't really matter, once we have the definition above, but I have seen two answers to this:

Row-Reduced Echelon Form - because the process of using ERO's can be called 'row-reduction', and we do get a (special!) echelon form.

Reduced Row-Echelon Form - because the echelon form discussed above can also be called 'row-echelon form'. We have achieved that but also 'reduced' the matrix further: extra ones and zeros!

What are the implications for codes? If two matrices are row-equivalent, then the span of each set of rows is the same. This means that, if the generator-matrices of two codes are row-equivalent, then the two codes are in fact the same set of vectors. However, two different generator-matrices for the same code will encode the same message as different codewords. (Try an example...). So when you come to channel-decode the codeword back to the message, you must use the generator-matrix that was used for encoding.

Similarly, doing EROs to a check-matrix correponds to manipulating the equations which codewords must satisfy, in such a way that the set of solutions remains the same. So if the check-matrices of two codes are row-equivalent, then the two codes are in fact the same. But in syndrome-decoding we must use the same check-matrix to make the table, and to find the syndrome of the received word.

Finally, note that the $G \leftrightarrow H$ algorithm, which streamlines the process of finding the solutions to equations, *requires a matrix in RREF*. Echelon form is not enough!

How does equivalence of codes relate to row-equivalence of matrices?

Given a generator- or check-matrix for a code, we can make the matrix for an equivalent code by using *column* operations. One could describe these as ECOs - but we only allow two kinds:

$CP_{i,j}$: Swap column i with column j. Using these we can permute the columns however we wish.

CM_λ: Multiply a column by some $\lambda \neq 0$ from the field we are working in.

So, roughly speaking, we could say that row-equivalence of matrices gives you the *same* code, but 'column-equivalence' of matrices only gives you an *equivalent* code.

Note that we do *not* use these column operations when putting a matrix into RREF. Once you have it in RREF, you might then use column operations to get the matrix to standard form - but then you don't have the same code any more, only an equivalent one.