This has been installed.

Appendix B: A Proof of Fact 4 (2.3.8)

Here is what we want to prove:

Fact 4: If **c** is any code word of the cyclic code generated by g(x), then the infinite sequence

$$\{o_i\} = \{\mathbf{c},\mathbf{c},\mathbf{c},\mathbf{c} \ldots\}$$

is a solution of the difference equation (2.3.2), which equation says

$$\sum_{j=0}^{k} h_j \, o_{m+j} = 0 \qquad m = 0,1,2.... \quad . \qquad (2.3.2)$$

The g(x) here is what one gets doing the division g(x) = (x^n-1)/h(x), and we assume that such an n exists such that such an even division exists (this assumption is discussed below).

Our proof is broken down into six short parts (1) through (6).

(1) GA Chapter 7 discusses the notion of a linear block code associated with a matrix G. One applies the non-square matrix G to a data vector **d** having k components and obtains thereby a code vector **c** having n components with n > k. If the vectors are row vectors, one writes **c** = **d**G.

Associated with matrix G is another matrix H called the parity check matrix for code G and one finds that, for any code word **c,** $\mathbf{c}H^T = \mathbf{0}$. One can use matrix H then to "check" to see whether **c** really is a code word of code G by seeing if $\mathbf{c}H^T = \mathbf{0}$ or not. One can show that the matrices G and H satisfy

$$GH^T = 0 \qquad \text{and} \qquad HG^T = 0 \qquad (B.1)$$

where each 0 is an all-zeros matrix of an appropriate size.

It turns out that the matrix H can itself be used to define another linear block code which is known as the dual code to code G. One applies the matrix H to a data vector **d'** having n-k components and obtains thereby a code vector **c'** having n components. If the vectors are row vectors, one writes **c'** = **d'**H.

It then turns out that the **c'** • **c** = 0 for any code word **c** of code G and any code word **c'** of code H. The proof is simple:

$$\mathbf{c}' \bullet \mathbf{c} = \mathbf{c}'\mathbf{c}^T = (\mathbf{d}'H)(\mathbf{d}G)^T = (\mathbf{d}'H)G^T\mathbf{d}^T = \mathbf{d}'(HG^T)\mathbf{d}^T = \mathbf{d}'(0)\mathbf{d}^T = 0 \qquad (B.2)$$

(2) GA Chapter 8 then discusses cyclic codes which have a close relationship to the block codes. The k components of the data vector **d** are now the coefficients of a polynomial d(x). The code words are generated now as the coefficients of polynomial c(x) which is obtained from polynomial multiplication,

$$c(x) = d(x)\, g(x). \qquad (B.3)$$

Here g(x) is a polynomial of degree n-k which is called the generator of the cyclic code. Associated with this cyclic code is a dual cyclic code where we have

$$c'(x) = d'(x)\, h(x) \qquad (B.4)$$

where the n-k components of the data polynomial d'(x) are the components of the data vector **d'** mentioned above. Polynomial h(x) is the generator of the dual code and has degree k. The relationship between g(x) and h(x) is this

$$g(x)h(x) = (x^n - 1) \ . \qquad (B.5)$$

Now the coefficients of c(x) which we call c_i are not necessarily the same as the c_i in part (1) above, but they are nonetheless the coefficients of some code word in the space of code words. The same comment applies to c'(x) and the c'_i.

(3) In (2.3.6) we *assumed* that there existed an integer n such that g(x)h(x) = $(x^n - 1)$ where h(x) was the polynomial used to construct the polynomial representation $GF(q=p^k) = R / (h(x))$ in (2.2.6). If h(x) is monic and irreducible with respect to GF(p) (sometimes we say "under GF(p)"), we know from (2.2.10) that h(x) is a minimum polynomial for some element α of GF(q). From the explanation given in the Proof of (2.2.33), we know that such an h(x) divides evenly into $(x^{q-1} - 1)$. We noted that there might be some n smaller than q-1 for which h(x) divides evenly into (x^n-1). Whatever the smallest n is, we just call it n. It might be q-1 if nothing smaller works. So, given our irreducible monic h(x) of Chapter 2, we know for sure that there exists a g(x) such that g(x)h(x) = $(x^n - 1)$. So we use that exact g(x) to define the cyclic code described in part 2 above, and then h(x) is our h(x) of interest in Chapter 2, it is not just some h(x) pulled out of a hat.

(4) Certainly d'(x) = 1 is a legal data word for the cyclic dual code and if we use it in (B.4) we obtain the legal dual code word c'(x) = h(x) . We can then write this equation c'(x) = h(x) as

$$c'_0 + c'_1x + c'_2x^2 + \ldots c'_kx^k + \ldots + c'_{n-1}x^{n-1} = h_0 + h_1x + h_2x^2 + \ldots h_kx^k + 0 + \ldots + 0x^{n-1}$$

We then write a vector equality for the coefficients,

$$\mathbf{c'} = \mathbf{h} \qquad \text{where} \qquad \mathbf{h} = (h_0,h_1,\ldots\ldots h_k,0,0\ldots0) \qquad (B.6)$$

where we fill out the end of vector **h** with a set of n-k-1 zeros so it then has n components to match **c'**. We then apply (B.2) to conclude that, for any code word **c** of the cyclic code generated by g(x),

$$\mathbf{h} \bullet \mathbf{c} = 0 \ . \qquad (B.7)$$

If we write this out as a summation, we get

$$\sum_{j=0}^{k} h_j c_j = 0 \qquad \text{(B.8)}$$

where the numbers $\{c_0, c_1, c_2, \ldots c_k\}$ are the first k+1 coefficients of **c**.

We have just proved in (B.8) this fact:

Fact : The dot product of the coefficients of h(x) with the first k+1 coefficients of any code word is 0.

(5) What happens if we dot h with *any* k+1 consecutive coefficients of a code word **c**? We know that there is some other code word **C** for which these k+1 consecutive coefficients are the *first* k+1 coefficients. We know this because we know that all cyclic permutations (rotations) of code words are also code words. Thus we have proved that :

Fact: The dot product of the coefficients of h(x) with *any* consecutive k+1 coefficients of any code word gives 0.

For example if k = 3 we might have this situation where (B.8) includes the products shown here as pairs of numbers one above the other :

c_0	c_1	c_2	c_3	c_4	c_5	.....	c_{n-1}
		h_0	h_1	h_2	h_3		

But (B.8) will also be true for any alignment which wraps the last coefficient like this one.

c_0	c_1	c_2	c_3	c_4	c_5	.....	c_{n-1}
h_1	h_2	h_3					h_0

Thus, when we say "consecutive coefficients" we would include a sequence like this

.... c_{n-2} c_{n-1} c_0 c_1 c_2.....

(6) Now consider the difference equation (2.3.2),

$$\sum_{j=0}^{k} h_j \; o_{m+j} = 0 \qquad m = 0,1,2.... \qquad \textit{(2.3.2)} \qquad \text{(B.9)}$$

Picture the sequence $\{o_i\}$ as an infinite row of numbers. Picture the sequence $\{h_j\}$ as a finite row of numbers. Place the row $\{h_j\}$ anywhere under the infinite row $\{o_i\}$, then take the sum of the product of the abutting numbers as done above. This is then the left side of (B.9), and the result is supposed to be zero. Varying m means varying the position of the row $\{h_j\}$. Here is a picture drawn for k=3 and m=5:

o_0	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}	o_{11}	...

$h_0 \quad h_1 \quad h_2 \quad h_3$

Now consider our *candidate* solution of the difference equation:

$$\{o_i\} = \{\mathbf{c}, \mathbf{c}, \mathbf{c}, \mathbf{c} \ldots\} \tag{B.10}$$

Consider the above process. No matter where we position $\{h_j\}$, we will be dotting **h** with a "consecutive" set of k+1 coefficients of the code word **c**. Suppose k = 3 and n = 5 (the size of the code words). The for (B.9) with m = 4 we would have

$$\begin{array}{ccccccccccccc} c_0 & c_1 & c_2 & c_3 & c_4 & c_0 & c_1 & c_2 & c_3 & c_4 & c_0 & c_1 & \ldots \\ & & & & h_0 & h_1 & h_2 & h_3 & & & & & \end{array}$$

No matter where we put our h_i row (that is, for any m > 0), equation (B.9) is satisfied. Therefore, the sequence of (B.10) is a solution of the difference equation (B.9) and that is what we set out to prove.