The Ubiquitous Reed-Solomon Codes

by Barry A. Cipra

Reprinted from *SIAM News*, Volume 26-1, January 1993

In this so-called Age of Information, no one need be reminded of the importance not only of speed but also of accuracy in the storage, retrieval, and transmission of data. It's more than a question of "Garbage In, Garbage Out." Machines do make errors, and their non-man-made mistakes can turn otherwise flawless programming into worthless, even dangerous, trash. Just as architects design buildings that will remain standing even through an earthquake, their computer counterparts have come up with sophisticated techniques capable of counteracting the digital manifestations of Murphy's Law.

What many might be unaware of, though, is the significance, in all this modern technology, of a five-page paper that appeared in 1960 in the *Journal of the Society for Industrial and Applied Mathematics*. The paper, "Polynomial Codes over Certain Finite Fields," by Irving S. Reed and Gustave Solomon, then staff members at MIT's Lincoln Laboratory, introduced ideas that form the core of current error-correcting techniques for everything from computer hard disk drives to CD players. Reed-Solomon codes (plus a lot of engineering wizardry, of course) made possible the stunning pictures of the outer planets sent back by Voyager II. They make it possible to scratch a compact disc and still enjoy the music. And in the not-too-distant future, they will enable the profitmongers of cable television to squeeze more than 500 channels into their systems, making a vast wasteland vaster yet.

"When you talk about CD players and digital audio tape and now digital television, and various other digital imaging systems that are coming--all of those need Reed-Solomon [codes] as an integral part of the system," says Robert McEliece, a coding theorist in the electrical engineering department at Caltech.

Why? Because digital information, virtually by definition, consists of strings of "bits"--0s and 1s--and a physical device, no matter how capably manufactured, may occasionally confuse the two. Voyager II, for example, was transmitting data at incredibly low power--barely a whisper--over tens of millions of miles. Disk drives pack data so densely that a read/write head can (almost) be excused if it can't tell where one bit stops and the next one (or zero) begins. Careful engineering can reduce the error rate to what may sound like a negligible level--the industry standard for hard disk drives is 1 in 10 billion--but given the volume of information processing done these days, that "negligible" level is an invitation to daily disaster. Error-correcting codes are a kind of safety net--mathematical insurance against the vagaries of an imperfect material world.

The key to error correction is redundancy. Indeed, the simplest error-correcting code is simply to repeat everything several times. If, for example, you anticipate no more than one error to occur in transmission, then repeating each bit three times and using "majority vote" at the receiving end will guarantee that the message is heard correctly (e.g., 111 000 011 111 will be correctly heard as 1011). In general, n errors can be compensated for by repeating things $2n + 1$ times.

But that kind of brute-force error correction would defeat the purpose of high-speed, high-density information processing. One would prefer an approach that adds only a few extra bits to a given message. Of course, as Mick Jagger reminds us, you can't always get what you want--but if you try, sometimes, you just might find you get what you need. The success of Reed-Solomon codes bears that out.

In 1960, the theory of error-correcting codes was only about a decade old. The basic theory of reliable digital communication had been set forth by Claude Shannon in the late 1940s. At the same time,

Richard Hamming introduced an elegant approach to single-error correction and double-error detection. Through the 1950s, a number of researchers began experimenting with a variety of error-correcting codes. But with their SIAM journal paper, McEliece says, Reed and Solomon "hit the jackpot."

The payoff was a coding system based on *groups* of bits--such as bytes--rather than individual 0s and 1s. That feature makes Reed-Solomon codes particularly good at dealing with "bursts" of errors: Six consecutive bit errors, for example, can affect at most two bytes. Thus, even a double-error-correction version of a Reed-Solomon code can provide a comfortable safety factor. (Current implementations of Reed-Solomon codes in CD technology are able to cope with error bursts as long as 4000 consecutive bits.)

Mathematically, Reed-Solomon codes are based on the arithmetic of finite fields. Indeed, the 1960 paper begins by defining a code as "a mapping from a vector space of dimension *m* over a finite field *K* into a vector space of higher dimension over the same field." Starting from a "message" *(a_0, a_1, . . ., a_{m-1})*, where each *a_k* is an element of the field *K*, a Reed-Solomon code produces *(P(0), P(g), P(g^2), . . ., P(g^{N-1}))*, where *N* is the number of elements in *K*, *g* is a generator of the (cyclic) group of nonzero elements in *K*, and *P(x)* is the polynomial *a_0 + a_1x + . . . + a_{m-1} x^{m-1}*. If *N* is greater than *m*, then the values of *P* overdetermine the polynomial, and the properties of finite fields guarantee that the coefficients of *P*--i.e., the original message--can be recovered from any *m* of the values.

Conceptually, the Reed-Solomon code specifies a polynomial by "plotting" a large number of points. And just as the eye can recognize and correct for a couple of "bad" points in what is otherwise clearly a smooth parabola, the Reed-Solomon code can spot incorrect values of P and still recover the original message. A modicum of combinatorial reasoning (and a bit of linear algebra) establishes that this approach can cope with up to *s* errors, as long as *m*, the message length, is strictly less than *N - 2s*.

In today's byte-sized world, for example, it might make sense to let *K* be the field of degree 8 over *Z_2*, so that each element of *K* corresponds to a single byte (in computerese, there are four bits to a nibble and two nibbles to a byte). In that case, *N = 2^8 = 256*, and hence messages up to 251 bytes long can be recovered even if two errors occur in transmitting the values *P(0), P(g), . . ., P(g^{255})*. That's a lot better than the 1255 bytes required by the say-everything-five-times approach.

Despite their advantages, Reed-Solomon codes did not go into use immediately--they had to wait for the hardware technology to catch up. "In 1960, there was no such thing as fast digital electronics"--at least not by today's standards, says McEliece. The Reed-Solomon paper "suggested some nice ways to process data, but nobody knew if it was practical or not, and in 1960 it probably wasn't practical."

But technology did catch up, and numerous researchers began to work on implementing the codes. One of the key individuals was Elwyn Berlekamp, a professor of electrical engineering at the University of California at Berkeley, who invented an efficient algorithm for decoding the Reed-Solomon code. Berlekamp's algorithm was used by Voyager II and is the basis for decoding in CD players. Many other bells and whistles (some of fundamental theoretic significance) have also been added. Compact discs, for example, use a version called cross-interleaved Reed-Solomon code, or CIRC.

Reed, now a professor of electrical engineering at the University of Southern California, is still working on problems in coding theory. Solomon, recently retired from the Hughes Aircraft Company, consults for the Jet Propulsion Laboratory. Reed was among the first to recognize the significance of abstract algebra as the basis for error-correcting codes.

"In hindsight it seems obvious," he told *SIAM News*. However, he added, "coding theory was not a subject when we published that paper." The two authors knew they had a nice result; they didn't know what impact the paper would have. Three decades later, the impact is clear. The vast array of applications, both current and pending, has settled the question of the practicality and significance of Reed-Solomon codes. "It's clear they're practical, because everybody's using them now," says

Berlekamp. Billions of dollars in modern technology depend on ideas that stem from Reed and Solomon’s original work. In short, says McEliece, "it’s been an extraordinarily influential paper."

(Barry A. Cipra is a mathematician and writer based in Northfield, Minnesota.)

SIAM
3600 University City Science Center
Philadelphia, PA 19104-2688, USA

Orders by electronic mail: service@siam.org
Orders by phone: +1 215/382-9800
Orders by phone: 1-800/447-7426 (USA only)
Fax: +1 215/386-7999
General email: siam@siam.org
