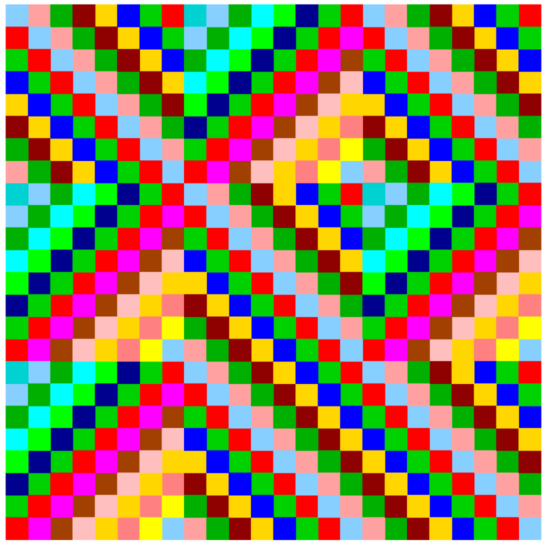
The Displacement Method in Coding Theory



M. Amin Shokrollahi

Lucent Technologies
Bell Labs Innovations

Joint work with Vadim Olshevsky

Motivation

The decoding of various classes of algebraic codes leads to the computation of a nonzero element in the kernel of a structured matrix.

Can we use the displacement method to perform this task more efficiently?

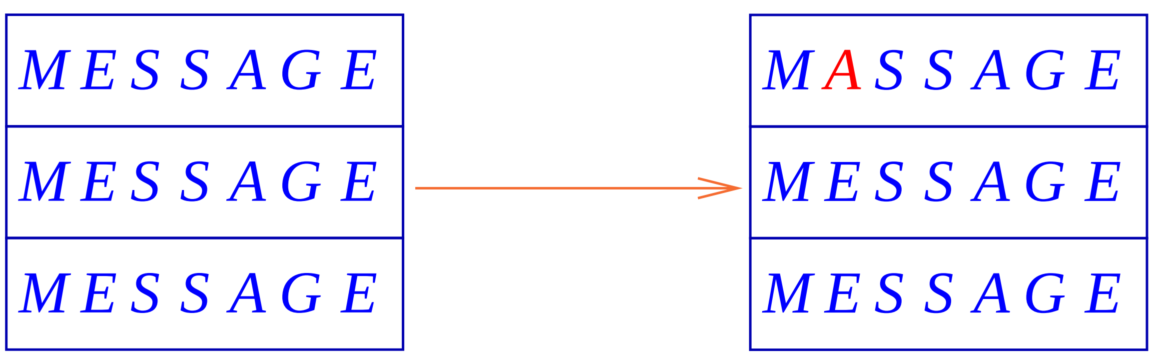
Yes, but with some restrictions.

Codes

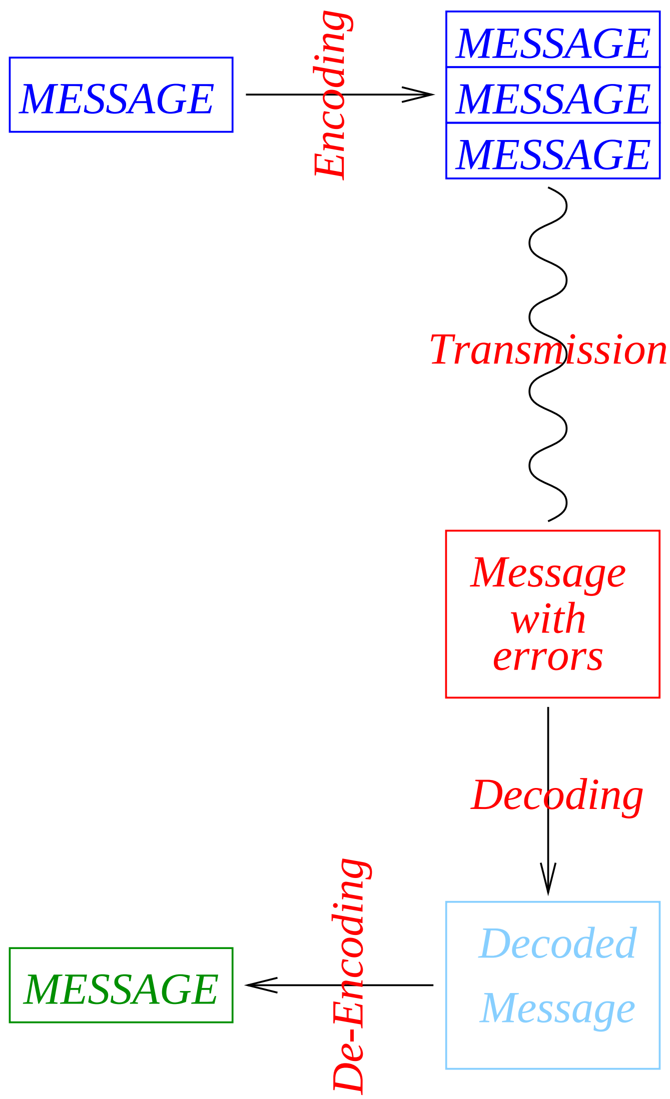
Codes are used when messages are to be transmitted over noisy channels.

Without coding

With coding



Encoding and Decoding



$$\text{Encoding}^{-1} \neq \text{Decoding!}$$

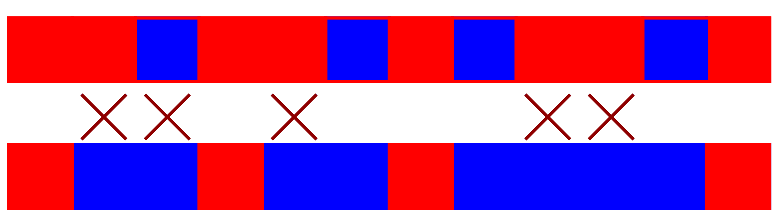
Linear codes

A k-dimensional subspace of $\mathbf{F}_q^n$ is called an $[n, k]_q$-code.

n is called the *block-length*, and k is called the *dimension* of the code.

The *Hamming weight* of a word x is the number of its nonzero entries.

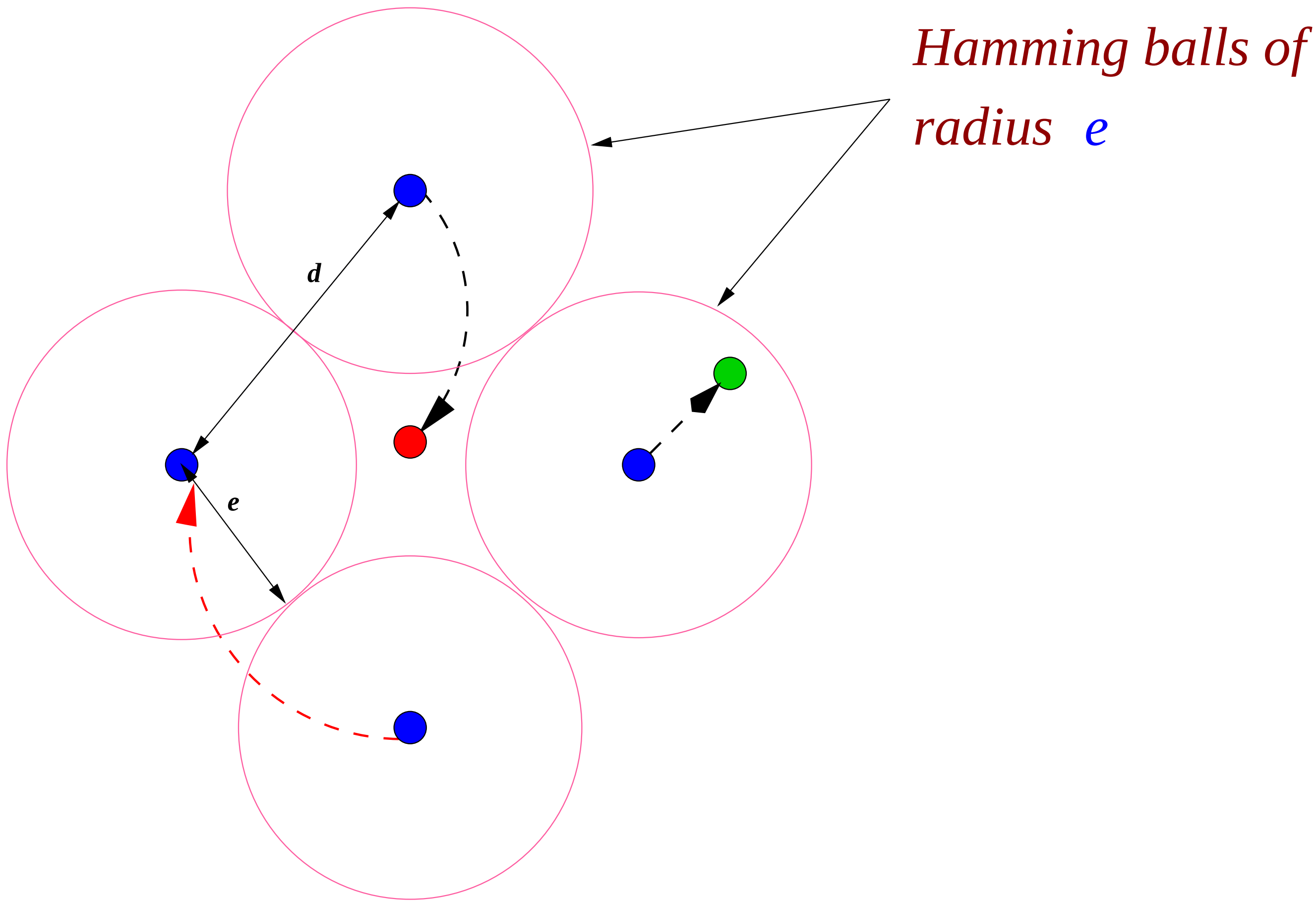
The *minimum distance* of a linear code is the minimum weight of a nonzero codeword, or the minimum distance between two distinct codewords.



Error correction

An $[n, k, d]_q$-code is an $[n, k]_q$-code of minimum distance d .

It is capable of correcting up to $e := (d - 1)/2$ errors.

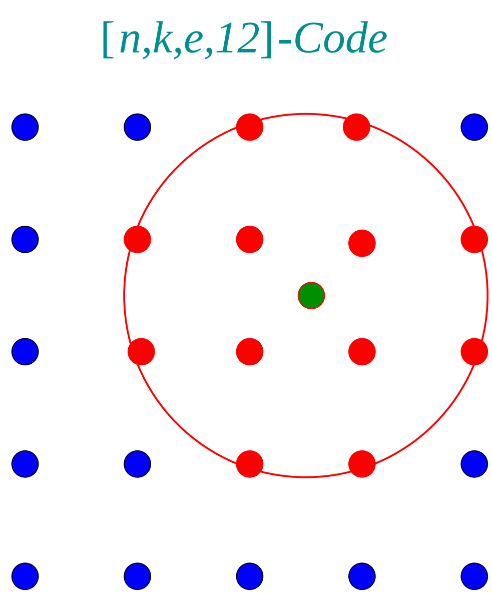


Beyond Error-correction Bound

An $[n, k, e, b]_q$-code is $[n, k]_q$-code such that any Hamming ball of radius e contains at most b codewords.

- $[n, k, d]_q$-code is $[n, k, n, q^k]_q$-code.
- $[n, k, d]_q$-code is $[n, k, (d-1)/2, 1]_q$-code.

Interested in: large e, small b, and efficient reconstruction algorithms.

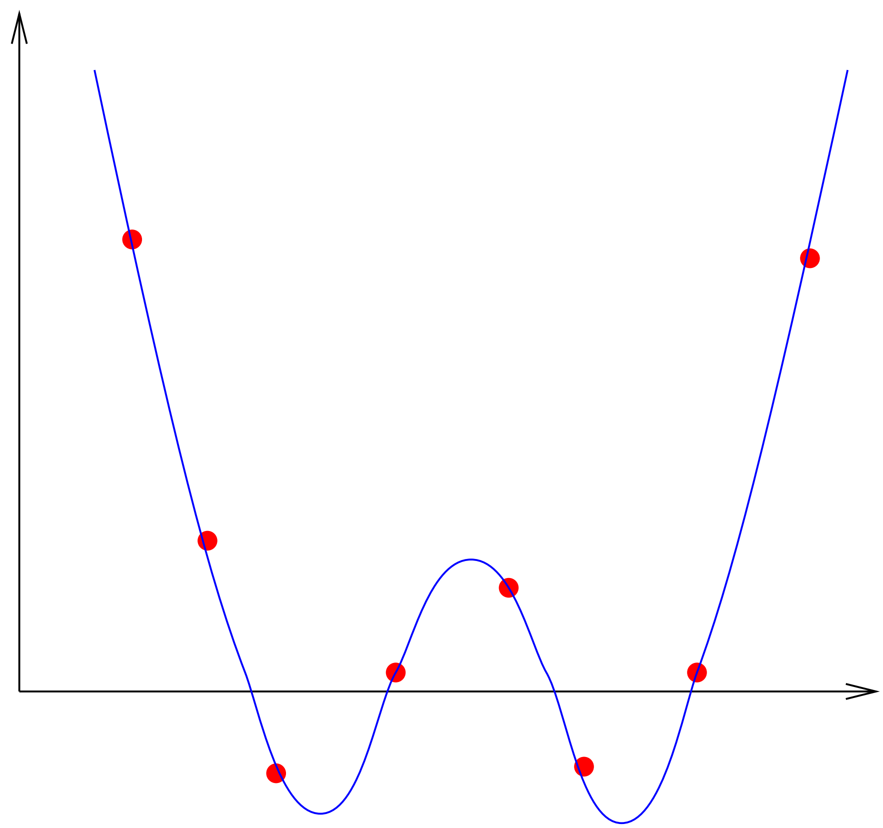


Reed-Solomon Codes

$x_1, \ldots, x_n$ distinct elements of $\mathbf{F}_q$, and $k \le n$.

$$\gamma\colon \mathbf{F}_q[X]_{<k} \to \mathbf{F}_q^n, \quad f \mapsto (f(x_1), \ldots, f(x_n)).$$

Image of γ is a linear code of block-length n over $\mathbf{F}_q$.
It is called a Reed-Solomon code or RS-code.



Parameters of Reed-Solomon Codes

Theorem. *Nonzero polynomial of degree m over a field can have at most m roots in the field.*

Implication:

above code has dimension k and minimum distance $n - k + 1$.

Encoding is "easy:" multiplication of a Vandermonde matrix with a vector.

Decoding?

Decoding

Decoding problem (after Welch-Berlekamp):

Let $(y_1, \ldots, y_n) \in \mathbf{F}_q^n$. Want polynomial $f \in \mathbf{F}_q[x]_{<k}$ such that at least $(n+k)/2$ of the values $f(x_1), \ldots, f(x_n)$ coincide with those of y_i.

(1) Compute $g \in \mathbf{F}_q[x]_{<(n+k)/2}$ and $h \in \mathbf{F}_q[x]_{\leq(n-k)/2}$, not both zero, such that

$$\forall\, i = 1, \ldots, n: \qquad g(x_i) + y_i h(x_i) = 0.$$

(2) Then $f = -g/h$.

Let $H := g + fh$.

Then $\deg(H) < (n+k)/2$.

But: H has at least $(n+k)/2$ zeros!

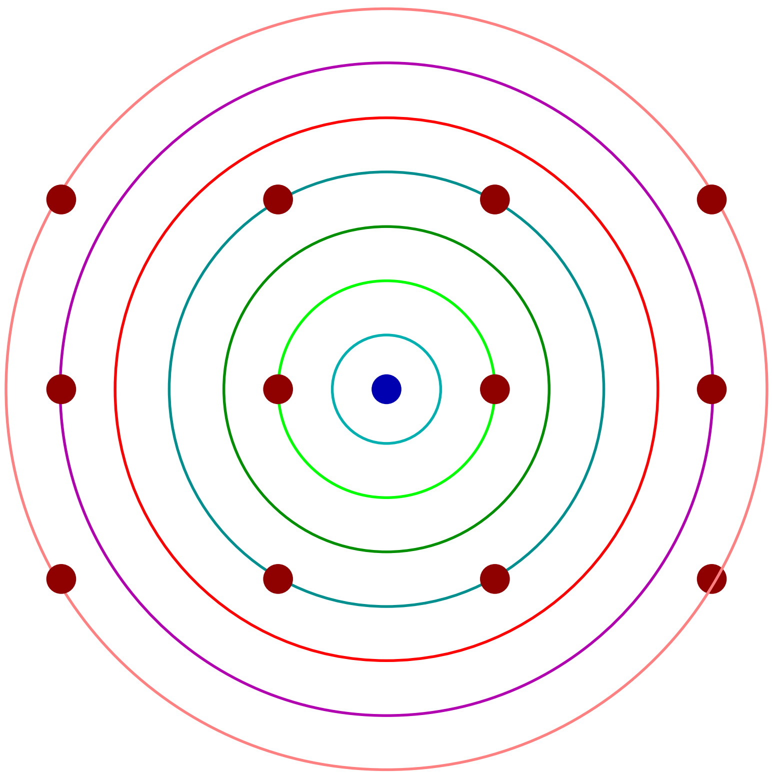
Example

$n = 6$ and $k = 4$:

$$\begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 & x_1^4 & y_1 & y_1x_1 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 & y_2 & y_2x_2 \\ 1 & x_3 & x_3^2 & x_3^3 & x_3^4 & y_3 & y_3x_3 \\ 1 & x_4 & x_4^2 & x_4^3 & x_4^4 & y_4 & y_4x_4 \\ 1 & x_5 & x_5^2 & x_5^3 & x_5^4 & y_5 & y_5x_5 \\ 1 & x_6 & x_6^2 & x_6^3 & x_6^4 & y_6 & y_6x_6 \end{pmatrix} \begin{pmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \\ g_4 \\ h_0 \\ h_2 \end{pmatrix} = 0.$$

List-Decoding of Reed-Solomon Codes

What if the number of errors is larger than $(n-k)/2$?



List-decoding (after M. Sudan):

Compute polynomials $h_0, \ldots, h_\ell$ with $\deg(h_i) \leq b - ik$ for an appropriate b, such that

$$\forall\, i = 1, \ldots, n: \quad h_0(x_i) + h_1(x_i)y_i + \cdots + h_\ell(x_i)y_i^\ell = 0.$$

The polynomials f with

$$h_0(x) + h_1(x)f(x) + \cdots + h_\ell(x)f(x)^\ell,$$

correspond to the possible codewords.

How to compute the h_i?

How to compute f?

Bivariate Interpolation

Given: Points $(x_1, y_1), \ldots, (x_n, y_n)$ over a field K, and integers $0 \le d_0 \le d_1 \le \cdots \le d_\ell$.

Want: Nontrivial Polynomial $H(x, y) := \sum_{i=0}^{\ell} h_i(x) y^i$ with $\deg(h_i) \le d_i$, such that $H(x_i, y_i) = 0$ for $i = 1, \ldots, n$.

Existence? f exists, if $\sum_{i=0}^{\ell}(d_i + 1) > n$.

$$H := h_{00} + h_{01}x + h_{02}x^2 + h_{10}y + h_{11}xy + h_{20}y^2.$$

We have

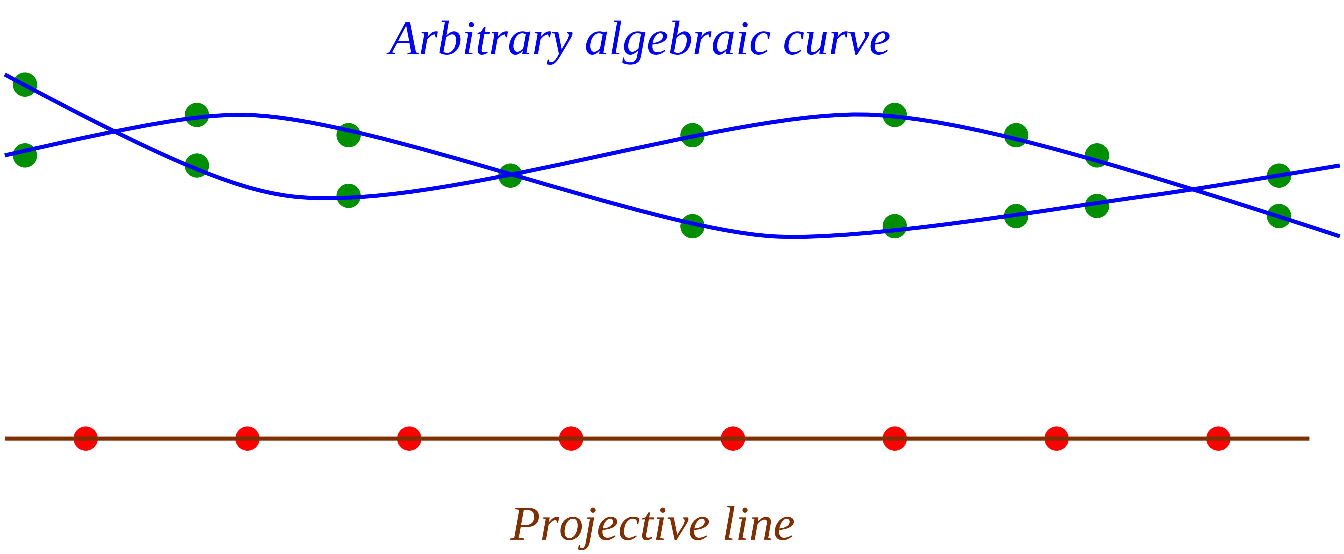
$$\begin{pmatrix} 1 & x_1 & x_1^2 & y_1 & y_1x_1 & y_1^2 \\ 1 & x_2 & x_2^2 & y_2 & y_2x_2 & y_2^2 \\ 1 & x_3 & x_3^2 & y_3 & y_3x_3 & y_3^2 \\ 1 & x_4 & x_4^2 & y_4 & y_4x_4 & y_4^2 \\ 1 & x_5 & x_5^2 & y_5 & y_5x_5 & y_5^2 \end{pmatrix} \begin{pmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{20} \end{pmatrix} = 0.$$

Algebraic Geometric Codes

Interpolation is done on algebraic curves rather than on the projective line.

List-decoding algorithms can be extended to this case (Shokrollahi-Wasserman).

For plane curves and certain low-dimensional projective models of curves, a part of the decoding process can be reduced to finding elements in the kernel of a structured matrix.



Example

Elliptic curve over $\mathbf{F}_5$:

$$Y^2 = X^3 - X.$$

Has points

$$\begin{array}{llll} P_1 = (0,0) & P_2 = (1,0) & P_3 = (4,0) & P_4 = (2,1) \\ P_5 = (2,4) & P_6 = (3,2) & P_7 = (3,3). & \end{array}$$

Let Q have projective coordinates $(0\colon 1\colon 0)$.

Then $L(3Q) = \langle 1, X, Y \rangle$ and C has generator matrix

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 4 & 2 & 2 & 3 & 3 \\ 0 & 0 & 0 & 1 & 4 & 2 & 3 \end{pmatrix}.$$

Dimension of the code is 3 and minimum distance is $4 = 7 - 3$.

The Displacement Method

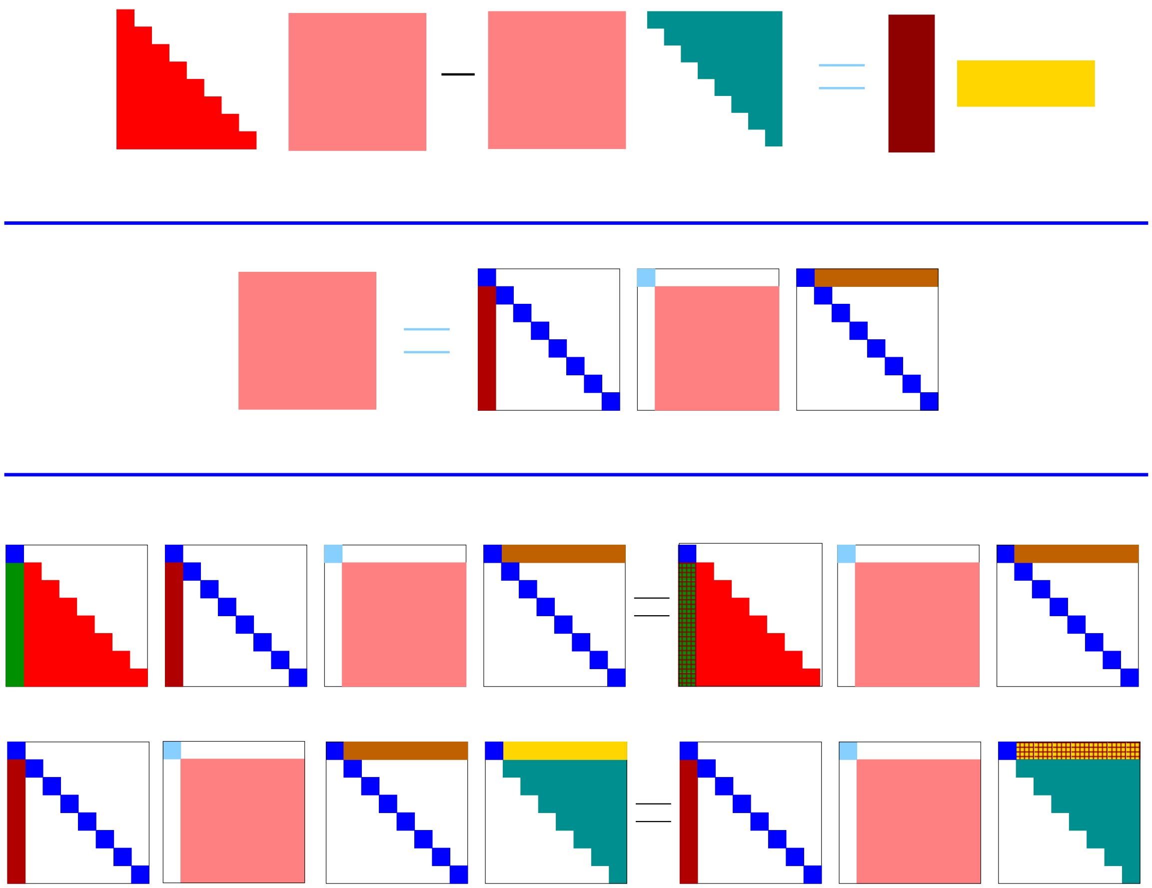
We can use the method of displacement to find non-trivial elements in the kernels of the above structured matrices.

The displacement method is not new: Morf, Kailath-Kung-Morf, Friedlander-Morf-Kailath-Ljung, Bitmead-Anderson, Heinig-Rost, Lev-Ari, Chun, Bruckstein, Sayed, Koltracht, Gohberg, Bin-Pan, Sahnovich, Dym, ...

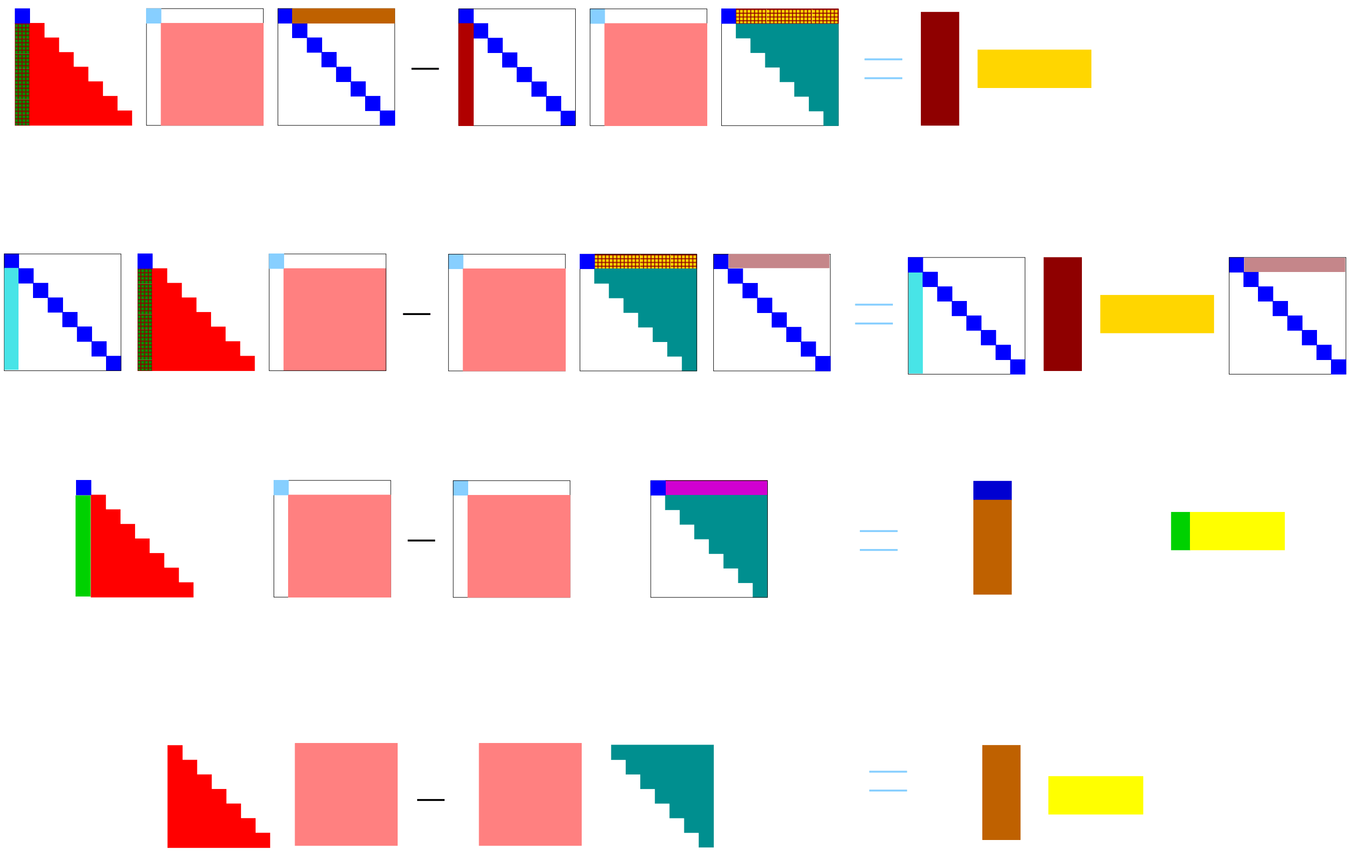
It has been used in many areas Control theory, interpolation, Numerical Mathematics, Signal Processing,...

Its use in Coding Theory seems to be novel.

The Displacement Method



The Displacement Method



Example: Decoding RS-Codes

$$\begin{pmatrix} \frac{1}{x_1} & & & & \\ & \frac{1}{x_2} & & & \\ & & \frac{1}{x_3} & & \\ & & & \frac{1}{x_4} & \\ & & & & \frac{1}{x_5} \end{pmatrix} \begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 & x_1^4 & y_1 & y_1x_1 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 & y_2 & y_2x_2 \\ 1 & x_3 & x_3^2 & x_3^3 & x_3^4 & y_3 & y_3x_3 \\ 1 & x_4 & x_4^2 & x_4^3 & x_4^4 & y_4 & y_4x_4 \\ 1 & x_5 & x_5^2 & x_5^3 & x_5^4 & y_5 & y_5x_5 \\ 1 & x_6 & x_6^2 & x_6^3 & x_6^4 & y_6 & y_6x_6 \end{pmatrix}$$

$$- \begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 & x_1^4 & y_1 & y_1x_1 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 & y_2 & y_2x_2 \\ 1 & x_3 & x_3^2 & x_3^3 & x_3^4 & y_3 & y_3x_3 \\ 1 & x_4 & x_4^2 & x_4^3 & x_4^4 & y_4 & y_4x_4 \\ 1 & x_5 & x_5^2 & x_5^3 & x_5^4 & y_5 & y_5x_5 \\ 1 & x_6 & x_6^2 & x_6^3 & x_6^4 & y_6 & y_6x_6 \end{pmatrix} \left(\begin{array}{ccccc|cc} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

$$= \begin{pmatrix} 1/x_1 & y_1/x_1 \\ 1/x_2 & y_2/x_2 \\ 1/x_3 & y_3/x_3 \\ 1/x_4 & y_4/x_4 \\ 1/x_5 & y_5/x_5 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Example: Bivariate Interpolation

$$
\begin{pmatrix}
\frac{1}{x_1} & & & & \\
& \frac{1}{x_2} & & & \\
& & \frac{1}{x_3} & & \\
& & & \frac{1}{x_4} & \\
& & & & \frac{1}{x_5}
\end{pmatrix}
\begin{pmatrix}
1 & x_1 & x_1^2 & y_1 & y_1x_1 & y_1^2 \\
1 & x_2 & x_2^2 & y_2 & y_2x_2 & y_2^2 \\
1 & x_3 & x_3^2 & y_3 & y_3x_3 & y_3^2 \\
1 & x_4 & x_4^2 & y_4 & y_4x_4 & y_4^2 \\
1 & x_5 & x_5^2 & y_5 & y_5x_5 & y_5^2
\end{pmatrix}
$$

$$
-\begin{pmatrix}
1 & x_1 & x_1^2 & y_1 & y_1x_1 & y_1^2 \\
1 & x_2 & x_2^2 & y_2 & y_2x_2 & y_2^2 \\
1 & x_3 & x_3^2 & y_3 & y_3x_3 & y_3^2 \\
1 & x_4 & x_4^2 & y_4 & y_4x_4 & y_4^2 \\
1 & x_5 & x_5^2 & y_5 & y_5x_5 & y_5^2
\end{pmatrix}
\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

$$
=\begin{pmatrix}
1/x_1 & y_1/x_1 - x_1^2 & y_1^2/{}_1 - y_1x_1 \\
1/x_2 & y_2/x_2 - x_2^2 & y_2^2/{}_2 - y_2x_2 \\
1/x_3 & y_3/x_3 - x_3^2 & y_3^2/{}_3 - y_3x_3 \\
1/x_4 & y_4/x_4 - x_4^2 & y_4^2/{}_4 - y_4x_4 \\
1/x_5 & y_5/x_5 - x_5^2 & y_5^2/{}_5 - y_5x_5
\end{pmatrix}
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}.
$$

Bivariate Interpolation

The Displacement method gives an algorithm for the LU-decomposition with running time

$$O(n^2\ell)$$

instead of $O(n^3)$ witht Gaussian elimination.

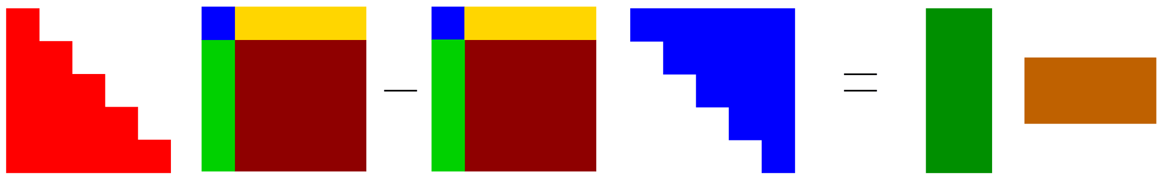
From the LU-decomposition one can determine H in time $O(n^2)$.

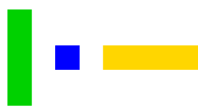
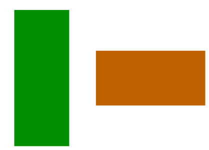
Extensions

- Hermite interpolation and list decoding.
- Algebraic geometric codes
- space efficient algorithms
- Parallel algorithms

Parallel Algorithms

Let



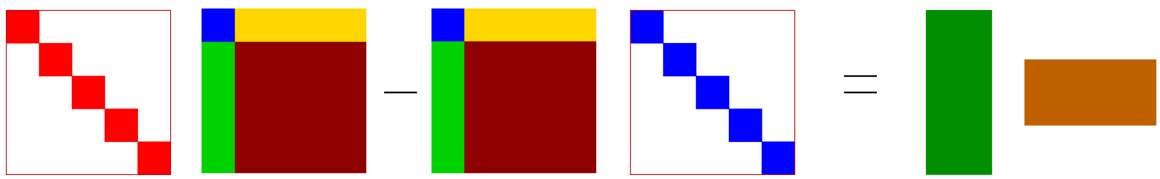
(1) Compute  through the generators  in parallel.

(2) Fill the first column of L and the first row of U.

(3) Compute generators of the Schur-complement and go back to (1). (Parallelizable.)

Parallel Algorithms

Need



to be able to perform first step in constant time on $O(n)$ processors.

This gives a parallel algorithm with running time

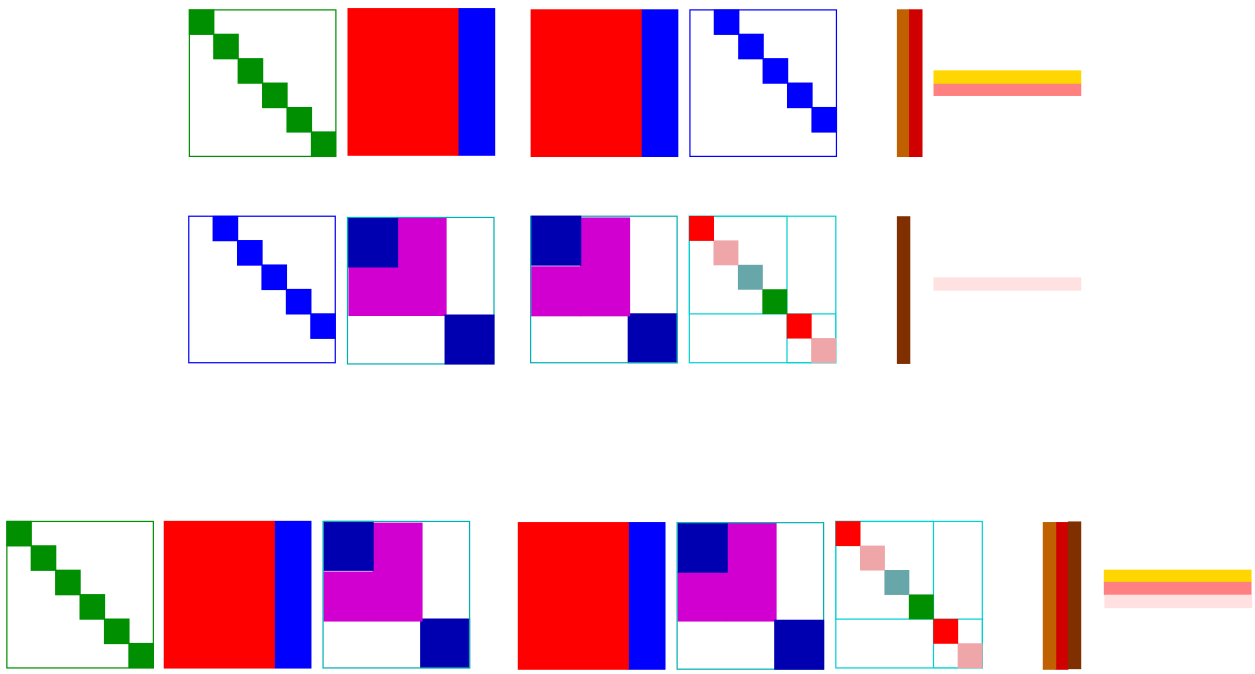
$$O(\ell n)$$

on

$$O(n)$$

processors.

WB: Parallel



Applications

- Soft decision decoding
- Biometric authentication
- Breaking block ciphers