Polar Grassmann Codes
Part 2

Luca Giuzzi

Università degli Studi di Brescia

ALCOMA15 — Kloster Banz 15–20 March 2015

Joint work with Ilaria Cardinali

Projective Orthogonal Grassmann Codes

- I. Cardinali, LG, "*Codes and Caps from Orthogonal Grassmannians*", Finite Fields Appl. **24** (2013), 148-169
- I. Cardinali, LG, A. Pasini, "*Line Polar Grassmann Codes of Orthogonal Type*", preprint (arXiv:1407.6149)
- I. Cardinali, LG, "*Enumerative Coding for Line Polar Grassmannians*", preprint (arXiv: 1412.5466)

Line Orthogonal Grassmann Codes

Theorem (I. Cardinali, LG, A. Pasini)

- *Codes and Caps from Orthogonal Grassmannians*, Finite Fields Appl. **24** (2013), 148-169
- *Line Polar Grassmann Codes of Orthogonal Type*, arXiv: 1407.6149

For q odd, the code $\mathcal{C}(\Delta_{n,2})$ has parameters

$$N = \frac{(q^{2n}-1)(q^{2n-2}-1)}{(q-1)(q^2-1)}, \qquad K = \binom{2n+1}{2}.$$

$$d_{\min} = q^{4n-5} - q^{3n-4}.$$

Requirements

Efficient algorithms for:

- Encoding
- Decoding
- Error correction

Requirements

Efficient algorithms for:

- Encoding
- Decoding
- Error correction

Remark

For $q = 3$, $n = 5$, $k = 2$

$$N = 24209680$$

For $q = 3$, $n = 7$, $k = 2$

$$N = 158866282120$$

Representing the lines

- Each line is represented by a $2 \times (2n + 1)$ matrix G in Reduced Row–Echelon Form (RREF[1])

[1] Also called Hermite normal form

Representing the lines

- Each line is represented by a $2 \times (2n+1)$ matrix G in Reduced Row–Echelon Form (RREF[1])

RREF

Matrix G such that:

- the first nonzero component in each row is 1
- the first nonzero component in the first row is to the left of the first nonzero component in the second row
- the entry above the first nonzero component in the second row is 0

[1] Also called Hermite normal form

RREF reduction

$$\begin{pmatrix} 0 & 3 & 9 & 0 & 3 \\ 2 & 6 & 6 & 0 & 0 \end{pmatrix}$$

RREF reduction

$$\begin{pmatrix} 2 & 6 & 6 & 0 & 0 \\ 0 & 3 & 9 & 0 & 3 \end{pmatrix}$$

RREF reduction

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

RREF reduction

$$\begin{pmatrix} 1 & 0 & -1 & 0 & -1 \\ 0 & 1 & 3 & 0 & 1 \end{pmatrix}$$

Representing the lines

Definition

For any $\ell \in \Delta_{n,2}$, we say that

$$L = \begin{pmatrix} a_1 & \dots & a_{2n+1} \\ b_1 & \dots & b_{2n+1} \end{pmatrix} = \begin{pmatrix} A \\ B \end{pmatrix}$$

represents ℓ if

1. $\ell = \langle A, B \rangle$
2. L is in RREF

Representing the lines

Definition

For any $\ell \in \Delta_{n,2}$, we say that

$$L = \begin{pmatrix} a_1 & \dots & a_{2n+1} \\ b_1 & \dots & b_{2n+1} \end{pmatrix} = \begin{pmatrix} A \\ B \end{pmatrix}$$

represents ℓ if

1. $\ell = \langle A, B \rangle$
2. L is in RREF

$$\mathcal{D}_{n,2} := \left\{ L_i = \begin{pmatrix} A_i \\ B_i \end{pmatrix} : \exists \ell_i \in \Delta_{n,2} \text{ represented by } L_i \right\}$$

Encoding (q odd)

- $V := V_{2n+1}$, $\mathcal{Q} := Q(2n, q)$
- $N = \#\Delta_{n,2} = \#\{$ totally singular lines in $\mathcal{Q}\}$; $K = 2n^2 + n$

Encoding (q odd)

- $V := V_{2n+1}$, $\mathcal{Q} := Q(2n,q)$
- $N = \#\Delta_{n,2} = \#\{$ totally singular lines in $\mathcal{Q}\}$;
  $K = 2n^2 + n$
- $\mathcal{D}_{n,2} := \left\{ L_i = \begin{pmatrix} A_i \\ B_i \end{pmatrix} : \exists \ell_i \in \Delta_{n,2} \text{ represented by } L_i \right\}$
- $\iota : \mathcal{D}_{n,2} \to \{0, \ldots, N-1\}$: bijection

Encoding (q odd)

- $V := V_{2n+1}$, $\mathcal{Q} := Q(2n, q)$
- $N = \#\Delta_{n,2} = \#\{$ totally singular lines in $\mathcal{Q}\}$;
  $K = 2n^2 + n$
- $\mathcal{D}_{n,2} := \left\{ L_i = \begin{pmatrix} A_i \\ B_i \end{pmatrix} : \exists \ell_i \in \Delta_{n,2} \text{ represented by } L_i \right\}$
- $\iota : \mathcal{D}_{n,2} \to \{0, \ldots, N-1\}$: bijection
- $\mathbf{m} = (m_1, \ldots, m_K) \in (\bigwedge^2 V)^*$: message

Encoding (q odd)

- $V := V_{2n+1}$, $\mathcal{Q} := Q(2n, q)$
- $N = \#\Delta_{n,2} = \#\{$ totally singular lines in $\mathcal{Q}\}$;
  $K = 2n^2 + n$
- $\mathcal{D}_{n,2} := \left\{ L_i = \begin{pmatrix} A_i \\ B_i \end{pmatrix} : \exists \ell_i \in \Delta_{n,2} \text{ represented by } L_i \right\}$
- $\iota : \mathcal{D}_{n,2} \to \{0, \dots, N-1\}$: bijection
- $\mathbf{m} = (m_1, \dots, m_K) \in (\bigwedge^2 V)^*$: message
- $\mathbf{m} : \bigwedge^2 V \to \mathbb{F}_q$

Encoding (q odd)

- $V := V_{2n+1}$, $\mathcal{Q} := Q(2n, q)$
- $N = \#\Delta_{n,2} = \#\{$ totally singular lines in $\mathcal{Q}\}$; $K = 2n^2 + n$
- $\mathcal{D}_{n,2} := \left\{ L_i = \begin{pmatrix} A_i \\ B_i \end{pmatrix} : \exists \ell_i \in \Delta_{n,2} \text{ represented by } L_i \right\}$
- $\iota : \mathcal{D}_{n,2} \to \{0, \dots, N-1\}$: bijection
- $\mathbf{m} = (m_1, \dots, m_K) \in (\bigwedge^2 V)^*$: message
- $\mathbf{m} : \bigwedge^2 V \to \mathbb{F}_q$
- $\varphi_{\mathbf{m}}(x, y) := \mathbf{m}(x \wedge y)$: bilinear alternating form on V
- Define $\mathbf{c} = (c_i)_{i=1}^N$ where

$$c_i = \varphi_{\mathbf{m}}(A_i, B_i).$$

Encoding (q odd)/details

- Represent the form $\varphi_{\mathbf{m}}$ with $\mathbf{m} = (m_1, \dots, m_K)$ by means of the matrix

$$M = \begin{pmatrix} 0 & m_1 & m_2 & \dots & m_{n-1} \\ -m_1 & 0 & m_n & \dots & m_{2n-3} \\ \vdots & \ddots & \ddots & & \vdots \\ -m_{n-1} & -m_{2n-3} & \dots & -m_K & 0 \end{pmatrix}$$

Encoding (q odd)/details

- Represent the form $\varphi_{\mathbf{m}}$ with $\mathbf{m} = (m_1, \ldots, m_K)$ by means of the matrix

$$M = \begin{pmatrix} 0 & m_1 & m_2 & \ldots & m_{n-1} \\ -m_1 & 0 & m_n & \ldots & m_{2n-3} \\ \vdots & \ddots & \ddots & & \vdots \\ -m_{n-1} & -m_{2n-3} & \ldots & -m_K & 0 \end{pmatrix}$$

- Let $L_i := \begin{pmatrix} A_i \\ B_i \end{pmatrix} = \iota^{-1}(i)$
- Define

$$c_i := \varphi_{\mathbf{m}}(\ell_i) := A_i M B_i^T.$$

Enumerative coding

Take

- $\preceq$: total order on $\mathbb{F}_q^2$
- $\mathcal{D}_{n,2}$: representatives of the lines of $\Delta_{n,2}$ in RREF

Enumerative coding

Take

- $\preceq$: total order on $\mathbb{F}_q^2$
- $\mathcal{D}_{n,2}$: representatives of the lines of $\Delta_{n,2}$ in RREF
- $\forall t \leq 2n+1$, $G_1, \ldots, G_t \in \mathbb{F}_q^2$ define

$$n(G_1, \ldots, G_t) = \#\{G \in \mathcal{D}_{n,2} : G \text{ begins with } G_1, \ldots, G_t\}$$

Enumerative coding

Take

- $\preceq$: total order on $\mathbb{F}_q^2$
- $\mathcal{D}_{n,2}$: representatives of the lines of $\Delta_{n,2}$ in RREF
- $\forall t \leq 2n+1$, $G_1, \dots, G_t \in \mathbb{F}_q^2$ define

$$n(G_1, \dots, G_t) = \#\{G \in \mathcal{D}_{n,2} : G \text{ begins with } G_1, \dots, G_t\}$$

- $\forall G = (G_1, \dots, G_{2n+1}) \in W$ let

$$\iota(G) := \sum_{j=1}^{2n+1} \sum_{X \prec G_j} n(G_1, \dots, G_{j-1}, X).$$

Prefix coding

Theorem (Cover)

●T.M. Cover, *Enumerative source encoding*, IEEE Trans. Information Theory, vol. **IT-19** (1973), 73–77

The function $\iota : \mathcal{D}_{n,2} \to \{0, \ldots, N-1\}$ *is a bijection*

Prefix coding

Theorem (Cover)

●T.M. Cover, *Enumerative source encoding*, IEEE Trans. Information Theory, vol. **IT-19** (1973), 73–77

The function $\iota : \mathcal{D}_{n,2} \to \{0, \dots, N-1\}$ *is a bijection*

Algorithm (decoding)

Input: $i \in \{0, \dots, N-1\}$
Output: $G \in \mathcal{D}_{n,2}$

$i_1 \leftarrow 1$
for $k = 1, \dots, 2n+1$ **do**

$$M \leftarrow \left\{ X : \begin{array}{l} \sum_{X \prec Y} n((G_1, \dots, G_{k-1}, X)) \leq i_k, \\ n((G_1, \dots, G_{k-1}, Y)) > 0 \end{array} \right\}$$

$G_k \leftarrow \max_{\prec} M$

$i_{k+1} \leftarrow i_k - \sum_{X \prec G_k} n((G_1, \dots, G_{k-1}, X))$

end for
return $G = (G_1, \dots G_k, \dots, G_{2n+1})$

Projective Grassmannians

- N. Silberstein and T. Etzion, *Enumerative coding for Grassmannian space*, IEEE Trans. Inform. Theory, **57** (2011), 365–374
- Y. Medvedeva, *Fast enumeration for Grassmannian space*, in "Problems of Redundancy in Information and Control Systems (RED), 2012 XIII International Symposium on". IEEE (2012), 48–52

Polar Grassmannians

Algorithm (I. Cardinali, LG)

It is possible to evaluate the function $\iota : \mathcal{D}_{n,2} \to \{0, \ldots, N-1\}$ as well as its inverse with complexity $O(n^3 q^2)$.

Polar Grassmannians

Algorithm (I. Cardinali, LG)

It is possible to evaluate the function $\iota : \mathcal{D}_{n,2} \to \{0, \dots, N-1\}$ as well as its inverse with complexity $O(n^3 q^2)$.

- We determine a function n_q for Cover's algorithm

Polar Grassmannians

Algorithm (I. Cardinali, LG)

It is possible to evaluate the function $\iota : \mathcal{D}_{n,2} \to \{0, \dots, N-1\}$ as well as its inverse with complexity $O(n^3q^2)$.

- We determine a function n_q for Cover's algorithm
- The result works also in even characteristic

Sketch of the algorithm

- Fix on V_{2n+1} the nondegenerate quadratic form

$$\mathfrak{q}_n(x) := x_1^2 + \sum_{i=1}^{n} x_{2i}x_{2i+1}$$

- Let

$$\mathfrak{b}_n(\mathbf{x},\mathbf{y}) = \begin{cases} x_1y_1 + \frac{1}{2}\sum_{i=1}^{n}(x_{2i}y_{2i+1} + y_{2i}x_{2i+1}) & \text{for } q \text{ odd} \\ \sum_{i=1}^{n}(x_{2i}y_{2i+1} + y_{2i}x_{2i+1}) & \text{for } q \text{ even} \end{cases}$$

Sketch of the algorithm/2

- Given (in RREF)

$$D = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_t \\ \beta_1 & \beta_2 & \dots & \beta_t \end{pmatrix} = \begin{pmatrix} A \\ B \end{pmatrix},$$

Sketch of the algorithm/2

- Given (in RREF)

$$D = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_t \\ \beta_1 & \beta_2 & \dots & \beta_t \end{pmatrix} = \begin{pmatrix} A \\ B \end{pmatrix},$$

- define

$$\widehat{D} := \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_t & x_{t+1} & \dots x_{2n+1} \\ \beta_1 & \beta_2 & \dots & \beta_t & y_{t+1} & \dots y_{2n+1} \end{pmatrix} = \begin{pmatrix} \widehat{A} \\ \widehat{B} \end{pmatrix}.$$

Sketch of the algorithm/2

- Given (in RREF)

$$D = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_t \\ \beta_1 & \beta_2 & \dots & \beta_t \end{pmatrix} = \begin{pmatrix} A \\ B \end{pmatrix},$$

- define

$$\widehat{D} := \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_t & x_{t+1} & \dots x_{2n+1} \\ \beta_1 & \beta_2 & \dots & \beta_t & y_{t+1} & \dots y_{2n+1} \end{pmatrix} = \begin{pmatrix} \widehat{A} \\ \widehat{B} \end{pmatrix}.$$

- $n_{\mathfrak{q}}(D)$ is the number of RREF solutions of

$$\begin{cases} \mathfrak{q}_n(\widehat{A}) = 0 \\ \mathfrak{q}_n(\widehat{B}) = 0 \\ \mathfrak{b}_n(\widehat{A}, \widehat{B}) = 0 \end{cases} \quad (1)$$

Example: $q = 5$, $n = 5$

$$q = 5, \qquad n = 5$$

Example: $q = 5$, $n = 5$

$$q = 5, \qquad n = 5$$

- $n_q(\emptyset) = 39736324056 = N$

Example: $q = 5$, $n = 5$

$$q = 5, \qquad n = 5$$

- $n_q(\emptyset) = 39736324056 = N$
- $n_q\begin{pmatrix}0\\0\end{pmatrix} = 1601656056, \qquad n_q\begin{pmatrix}1\\0\end{pmatrix} = 38134668000.$

Example: $q = 5$, $n = 5$

$$q = 5, \qquad n = 5$$

- $n_q(\emptyset) = 39736324056 = N$
- $n_q\begin{pmatrix}0\\0\end{pmatrix} = 1601656056, \qquad n_q\begin{pmatrix}1\\0\end{pmatrix} = 38134668000.$
- $n_q\begin{pmatrix}0&0\\0&0\end{pmatrix} = 66031056, \qquad n_q\begin{pmatrix}0&1\\0&0\end{pmatrix} = 1535625000$
- $n_q\begin{pmatrix}1&0\\0&0\end{pmatrix} = 1523418000, \qquad n_q\begin{pmatrix}1&0\\0&1\end{pmatrix} = 30468750000.$

Example: $q = 5$, $n = 5$

$$q = 5, \qquad n = 5$$

- $n_q(\emptyset) = 39736324056 = N$
- $n_q\begin{pmatrix}0\\0\end{pmatrix} = 1601656056, \qquad n_q\begin{pmatrix}1\\0\end{pmatrix} = 38134668000.$
- $n_q\begin{pmatrix}0&0\\0&0\end{pmatrix} = 66031056, \qquad n_q\begin{pmatrix}0&1\\0&0\end{pmatrix} = 1535625000$
- $n_q\begin{pmatrix}1&0\\0&0\end{pmatrix} = 1523418000, \qquad n_q\begin{pmatrix}1&0\\0&1\end{pmatrix} = 30468750000.$
- $n_q\begin{pmatrix}1&0&1\\0&1&1\end{pmatrix} = 12285 \cdot 10^5, \qquad n_q\begin{pmatrix}1&0&1\\0&1&2\end{pmatrix} = 12090 \cdot 10^5$

Example: $q = 5$, $n = 5$

$$\iota(0) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \leftrightarrow \ell_0$$

Example: $q = 5$, $n = 5$

$$\iota(0) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \leftrightarrow \ell_0$$

$$\iota(1) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \leftrightarrow \ell_1$$

Example: $q = 5$, $n = 5$

$$\iota(0) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \leftrightarrow \ell_0$$

$$\iota(1) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \leftrightarrow \ell_1$$

$$\vdots$$

Example: $q = 5$, $n = 5$

$$\iota(0) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \leftrightarrow \ell_0$$

$$\iota(1) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \leftrightarrow \ell_1$$

$$\vdots$$

$$\iota(N-1) = \begin{pmatrix} 1 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 3 \end{pmatrix} \leftrightarrow \ell_{N-1}$$

Example: $q = 5$, $n = 5$

$$\iota(0) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \leftrightarrow \ell_0$$

$$\iota(1) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \leftrightarrow \ell_1$$

$$\vdots$$

$$\iota(N-1) = \begin{pmatrix} 1 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 3 \end{pmatrix} \leftrightarrow \ell_{N-1}$$

$$\iota(N) = \begin{pmatrix} 1 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 4 \end{pmatrix} \leftrightarrow \ell_N$$

Decoding

Received vector:

$$\mathbf{r} := (r_1, \ldots, r_N)$$

Decoding:

- Select some independent components r_i for $i \in \mathbb{I}$
- Suppose $\ell_i = \langle A_i, B_i \rangle$ with $L_i = \begin{pmatrix} A_i \\ B_i \end{pmatrix}$
- Determine $\varphi : \bigwedge^2 V \to \mathbb{F}$ such that

$$\varphi(A_i \wedge B_i) = r_i$$

- Reconstruct the antisymmetric matrix M of ϕ
- Check the behavior on the remaining components

Error correction

Received vector:

$$\mathbf{r} = (r_1, \ldots, r_N)$$

- We expect $\mathbf{r}$ to contain only *some* components of a codeword (plenty of erasures)
- We need *local correction* algorithms
- We want to be able to recover a given component r_i without using most of the remaining components

Error correction

Received vector:

$$\mathbf{r} = (r_1, \ldots, r_N)$$

Correction:

- Fix a component $i \in \{0, \ldots, N\}$

Error correction

Received vector:

$$\mathbf{r} = (r_1, \ldots, r_N)$$

Correction:

- Fix a component $i \in \{0, \ldots, N\}$
- Determine $\ell = \langle \iota^{-1}(i) \rangle$

Error correction

Received vector:

$$\mathbf{r} = (r_1, \dots, r_N)$$

Correction:

- Fix a component $i \in \{0, \dots, N\}$
- Determine $\ell = \langle \iota^{-1}(i) \rangle$
- Let
$$\Sigma_i := \{\pi : \ell \subseteq \pi \subseteq \mathcal{Q}, \dim \pi = 3\}$$
- Choose $\pi \in \Sigma_i$

Error Correction

- for any s, t lines of π with $s, t \neq \ell$ determine an alternating form ψ_{st} defined on π such that

$$\psi_{st}(s) := \psi_{st}(S_1 \wedge S_2) = r_{\iota(s)}$$

$$\psi_{st}(t) := \psi_{st}(T_1 \wedge T_2) = r_{\iota(t)}$$

Error Correction

- for any s, t lines of π with $s, t \neq \ell$ determine an alternating form ψ_{st} defined on π such that

$$\psi_{st}(s) := \psi_{st}(S_1 \wedge S_2) = r_{\iota(s)}$$

$$\psi_{st}(t) := \psi_{st}(T_1 \wedge T_2) = r_{\iota(t)}$$

- ψ_{st} is represented by a (singular) 3×3 antisymmetric matrix
- compute $\psi_{st}(\ell) := \psi_{st}(L_1 \wedge L_2)$

Error Correction

- for any s, t lines of π with $s, t \neq \ell$ determine an alternating form ψ_{st} defined on π such that

$$\psi_{st}(s) := \psi_{st}(S_1 \wedge S_2) = r_{\iota(s)}$$

$$\psi_{st}(t) := \psi_{st}(T_1 \wedge T_2) = r_{\iota(t)}$$

- ψ_{st} is represented by a (singular) 3×3 antisymmetric matrix
- compute $\psi_{st}(\ell) := \psi_{st}(L_1 \wedge L_2)$
- if $\psi_{st}(\ell) = r_i$, then accept r_i;
- otherwise, choose any two other lines $s', t' \subseteq \pi$
- choose the majority value for r_i

Conclusions

Remark

- *The strategies for encoding, error correction and decoding work for all Line Grassmann codes (Projective, Orthogonal, Symplectic, Hermitian)*

Conclusions

Remark

- *The strategies for encoding, error correction and decoding work for all Line Grassmann codes (Projective, Orthogonal, Symplectic, Hermitian)*
- *We need efficient enumerators for the points*

Conclusions

Remark

- *The strategies for encoding, error correction and decoding work for all Line Grassmann codes (Projective, Orthogonal, Symplectic, Hermitian)*
- *We need efficient enumerators for the points*
- *Enumerators for* projective *Grassmannians are more efficient than those for the* polar *cases*

Future developments

- Extend the index coding algorithm for $k > 2$
- Index coding for $\left\{ \begin{array}{c} \text{Symplectic} \\ \text{Hermitian} \end{array} \right\}$ Grassmann codes
- Better bounds on the complexity of error correction
- Improved local error correction and decoding