Efficient List-Decoding of Reed-Solomon Codes with the Fundamental Iterative Algorithm

Alexander Zeh, Christian Gentner, Martin Bossert

Efficient List-Decoding of Reed-Solomon Codes with the Fundamental Iterative Algorithm

Alexander Zeh, Christian Gentner and Martin Bossert
Department of Telecommunications and Applied Information Theory
University of Ulm, Germany
{alexander.zeh,christian.gentner,martin.bossert}@uni-ulm.de

***Abstract*—In this paper we propose a new algorithm that solves the Guruswami-Sudan interpolation step for Reed-Solomon codes efficiently. It is a generalization of the Feng-Tzeng approach, the so-called Fundamental Iterative Algorithm. From the interpolation constraints of the Guruswami-Sudan principle it is well known that an improvement of the decoding radius can only be achieved, if the multiplicity parameter s is smaller than the list size l. The code length is n and our proposed algorithm has a complexity (without asymptotic assumptions) of $\mathcal{O}\left(ls^4n^2\right)$.**

***Index Terms*—Guruswami-Sudan algorithm, list decoding, Fundamental Iterative Algorithm (FIA), Reed-Solomon codes, Hankel matrices**

I. Introduction

A Reed-Solomon code $\mathcal{RS}(n,k,d)$ with arbitrary rate $R = k/n$ can be list-decoded with the Guruswami-Sudan (GS) principle (see [1], [2]). The received vector is interpolated with a bivariate polynomial $Q(x,y)$ and the y-roots of Q form the list of possible sent codewords.

The reformulation of this interpolation problem over a univariate polynomial ring was done by Roth and Ruckenstein [3] for the Sudan case (where the multiplicity is $s = 1$) and generalized to GS in [4]. This allows a syndrome-based decoding method and a representation of the set of linear equations as Hankel (or Töplitz) matrices. We generalize here the Feng-Tzeng [5] approach, the so-called Fundamental Iterative Algorithm (FIA), to the GS-case, where a block Hankel-matrix occurs.

In the next section we recall the basic properties of the original FIA and show how it can be used to solve the Key Equation (KE) for classical decoding of RS codes up to half the minimum distance. The original FIA [5] can be seen as special case of our algorithm. To get an idea of the extension, we summarize the GS-interpolation conditions in Section III. The Sudan Key Equation (SKE) of [3] and the Guruswami-Sudan Key Equation (GSKE) of [4] will be presented in Section IV. The set of linear homogeneous equations of the SKE can be represented as a row of Hankel matrices (where one Hankel matrix occurs for the classical KE). For the GSKE the matrix consists of many horizontally and vertically arranged Hankel matrices (so-called block Hankel-matrix). To adapt the FIA to GSKE, we first consider an intermediate step of one line of vertically arranged Hankel matrices (see Section V). The final algorithm is presented in Section VI and its complexity is analyzed. We conclude in Section VII. In the appendix we consider a simple example and illustrate the principle of our algorithm. (The reference [6] was mentioned by the reviewer, but in our opinion the connection to our algorithm solving the complete set of equations for the GS-algorithm is marginal.)

II. Key Equation, FIA and Hankel matrices

The so-called Fundamental Iterative Algorithm (FIA) (introduced in [5]) is used to find the minimal number of first columns of an arbitrary $\tau\times(\tau+1)$ matrix $\mathcal{S} = [\mathcal{S}_{i,j}]$ where $i = 0,\dots,\tau-1$ and $j = 0,\dots,\tau$ which are linearly dependent. The time complexity of the FIA for such a matrix is $\mathcal{O}\left(\tau^3\right)$ (comparable to the standard gaussian procedure). Nevertheless, if the FIA is applied to structured matrices the complexity can be reduced. Therefore, we recall the definition of the Hankel matrix in the following.

Definition 1 (Hankel matrix) *A $\tau\times(\tau+1)$ Hankel matrix $\mathcal{S}$ is a matrix where $\mathcal{S}_{i-1,j+1} = \mathcal{S}_{i,j}\ \forall\, i = 1,\dots,\tau-1$ and $j = 0,\dots,\tau-1$ holds.*

We remark that a $\tau\times(\tau+1)$ Hankel matrix consists of 2τ different elements. The FIA can be tailored to this Hankel structure and it finds the smallest integer c, such that the columns 0 through c of a $\tau\times(\tau+1)$ Hankel matrix $\mathcal{S}$ are linearly independent with time complexity $\mathcal{O}\left(\tau^2\right)$.

Here, we want to point out that this algorithm can be used to solve the classical Key Equation (KE) for RS-codes, where the decoding radius $\tau \le \lfloor\frac{n-k}{2}\rfloor$. The classical KE is:

$$\Lambda(x)\cdot S(x) \equiv \Omega(x) \mod x^{n-k}, \tag{1}$$

where $\deg\Omega(x) < n-k-\tau$ and $\Lambda(x) = \sum_{i=0}^{n-k-\tau}\Lambda_i x^i$ is the error-locator polynomial. Representing the τ homogeneous linear equations gives us the Hankel structure of the matrix coming from the polynomial multiplication of the previous equation. It is well-known that Massey's algorithm [7] solves the KE with complexity $\mathcal{O}\left(\tau^2\right)$ and that this problem is equivalent to the problem of finding the the shortest linear shift register that generates a zero sequence when the inputs are the coefficients of the syndrome polynomial $S(x)$.

III. Guruswami-Sudan Principle

The Guruswami-Sudan principle [2] is recalled shortly. Let $\{\alpha_1,\dots,\alpha_n\}$ be the support of a Reed-Solomon code $\mathcal{RS}(n,k,d)$, where all the $\alpha_i \in \mathbb{F}_q$ are distinct. Let k be

the dimension and $d = n - k + 1$ the minimum distance of the RS code under consideration and a codeword $(c_1, \ldots, c_n)$ is defined by $c_i = f(\alpha_i)\ \forall\ i = 1, \ldots, n$. The received word is denoted by $\mathbf{r} = (r_1, \ldots, r_n)$ and is the addition of the codeword and the error. The number of errors that can be corrected is denoted by τ. The parameter s is the order of multiplicity of the bivariate interpolation polynomial in the GS-algorithm. Then the GS-polynomial $Q(x, y)$ has to fulfill the following three conditions:

① $Q(x, y) \neq 0$;
② $Q(x, y) = \sum_{t=0}^{l} Q^{(t)}(x) y^t = \sum_{t=0}^{l} \sum_{i}^{N_t - 1} q_{t,i} x^i y^t$, where $\deg Q^{(t)}(x) < N_t$ with $N_t = s(n-\tau) - t(k-1)$;
③ $\mathsf{mult}(Q(x, y), (\alpha_i, r_i)) \geq s$, $i = 1, \ldots, n$.

We recall that the condition ③ is the multiplicity condition defined as follows: Let $Q(x, y) = Q^{(0)} + Q^{(1)} + \cdots + Q^{(i)} + \cdots$ be given, where $Q^{(i)}$ is homogeneous of degree i. The multiplicity of Q at the point $(0, 0)$ (here denoted with $\mathsf{mult}(Q(x, y), (0, 0))$ is the smallest i such that $Q_i \neq 0$ and the multiplicity of Q at the point (α_i, β_i) is the multiplicity at $(0, 0)$ of the polynomial $Q(x + \alpha_i, y + \beta_i)$.
Then for all $f(x)$, corresponding to codewords c such that Hamming distance $d(f, c) \leq \tau$, it holds, that $Q(x, f(x)) = 0$. It is noted, that $s = 1$ in the case of Sudan.
To determine the number of correctable errors, the following conditions (see [2]) have to be satisfied:

$$s\tau \leq sn - (m+1) - l(k-1), \tag{2}$$

where m is a nonnegative integer such that $(m+1)(l+1) + (k-1)\binom{l+1}{2} > \binom{s+1}{2} n$. The degree of $Q^{(t)}$ can also be expressed by $N_t = m + 1 + (l - t)(k - 1)$. We remark that we search the smallest m which satisfies the above condition. So we can bound the number of unknowns (coefficients $q_{t,i}$ of the polynomials $Q^{(t)}$) by:

$$\binom{s+1}{2} n < \sum_{i=0}^{l} N_t < \binom{s+1}{2} n + l. \tag{3}$$

We will use this bound to determine the complexity of our algorithm in Section VI.
In Table I we consider $\mathcal{RS}(16, 4, 13)$ code over $F_{17} = GF(17)$. Classical decoding permits to correct six errors (see Table I, denoted with $s = 0$). Sudan's algorithm increases the decoding radius to $\tau = 7$ already. If we apply the modified interpolation conditions of Guruswami-Sudan (for $s = 2$), we can correct eight errors and the corresponding list size is $l = 4$. McEliece [8] has shown that $\bar{l}(\tau)$ is slightly less than

TABLE I
EXAMPLE FOR $\mathcal{RS}(16, 4, 13)$ OVER $GF(17)$.

Multiplicity s	**Radius** τ	**List Size** $l(s)$	$\bar{l}(\tau)$
0	6	1	$3.36183098 \times 10^{-4}$
1	7	2	$7.280428277 \times 10^{-3}$
2	8	4	$1.24464565671 \times 10^{-1}$
28	9	64	1.449716376

a rigorous bound on the average number of codewords on the list, which were not sent (and within the decoding radius τ from the received word and).
So for $s = 28$ we have in average ≈ 2.45 codewords on the list ($\bar{l}(\tau)$ and the sent codeword). We refer to this example in the annexe, where we apply our new algorithm to the GS-case with multiplicity $s = 2$.

IV. List Decoding and FIA

A. Overview

In this section we provide the link between the list-decoding principle of Guruswami-Sudan for Reed-Solomon codes and the FIA (as mentioned in Section II for Hankel matrices). We first consider the Sudan case (where the interpolation multiplicity is $s = 1$) and show how the corresponding system of linear equations can be solved efficiently. If the SKE represented in matrix form, we obtain a horizontal line of $l+1$ (without the original reduction of [3]) Hankel matrices. In the GSKE case the matrix form of the set of linear equations gives us a matrix of $(l + 1) \times s$ Hankel matrices.

B. Sudan case

Sudan's original approach [1] was reformulated to the SKE in [3] and [9]. In the following we skip the reduction step (where the $Q^{(0)}$ with degree smaller than $N_0 = n - \tau$ can be interpolated) and present the full system of n linear homogeneous equations. The following lemma gives the basic idea of [3].

Lemma 1 *Let $Q(x, y) = \sum_{t=0}^{l} Q^{(t)}(x) y^t$ be the Sudan interpolation polynomial that satisfies the conditions ①-③ for $s = 1$ and let $R(x)$, such that $R(\alpha_i) = r_i\ \forall\ i = 1, \ldots, n$. Furthermore, let $G(x) = \prod_{j=1}^{n} (x - \alpha_i)$. Then $Q(x, y)$ satisfies condition ③, if and only if there exist a polynomial $B(x)$ over F for which*

$$Q(x, R(x)) = B(x) \cdot G(x), \tag{4}$$

where $\deg B(x) < l(n - k) - \tau$.

Using condition ② and dividing by $G(x)$ gives us n equations on the unknowns.

$$\sum_{t=0}^{l} Q^{(t)}(x) \frac{R(x)^t}{G(x)} = B(x) \tag{5}$$

We write $Q(x, y)$ as a vector $\mathbf{Q}$:

$$\mathbf{Q} = \begin{pmatrix} \mathcal{Q}^{(0)} & \mathcal{Q}^{(1)} & \ldots & \mathcal{Q}^{(l)} \end{pmatrix}^T,$$

where $\mathcal{Q}^{(t)} = (\mathcal{Q}_0^{(t)}, \mathcal{Q}_1^{(t)}, \ldots, \mathcal{Q}_{N_t - 1}^{(t)})^T$. Equivalent to this representation the syndrome polynomials $S^{(t)}(x)$ lead to $l+1$ Hankel matrices $\mathcal{S}^{(t)} = [\mathcal{S}_{i,c}^{(t)}]_{i,c}\ \forall\, t = 0, \ldots, l$.
In comparison to the original approach of Roth and Ruckenstein [3], the number of rows of the matrices $\mathbf{S} = (\mathcal{S}^{(0)}\ \mathcal{S}^{(1)}\ \cdots\ \mathcal{S}^{(l)})$ is n (not τ) and the number of columns is N_t for all $t = 0, \ldots, l$ (for the definition see [3]). Finally, we obtain the following matrix representation for the non-reduced SKE:

$$\begin{pmatrix} \mathcal{S}^{(0)} & \mathcal{S}^{(1)} & \cdots & \mathcal{S}^{(l)} \end{pmatrix} \cdot \mathbf{Q} = \mathbf{0}. \tag{6}$$

The adaption of the FIA for horizontally arranged Hankel matrices in [3] and [9] is done by a re-ordering of the columns that correspond to the weighted degree of the interpolation polynomial $Q(x,y)$. Here, we will denote this as horizontal ordering $\prec_H$ of the columns (denoted by t) of $\mathcal{S}^{(i)}$ and it will be used in our algorithm in Section VI. It is defined as $(i,t) \prec_H (i',t')$ if and only if:

$$\begin{gathered} t + i(k-1) < t' + i'(k-1) \\ \text{or} \\ t + i(k-1) = t' + i'(k-1) \text{ and } i < i'. \end{gathered} \tag{7}$$

The time complexity of the FIA for this case ($l+1$ horizontally arranged Hankel matrices) and for n equations is $\mathcal{O}\left(ln^2\right)$. We remark that in [3] the SKE was called Extended Key Equation (EKE) and that the complexity of the FIA-based algorithm can be reduced by omitting the $Q^{(0)}$ to τ equations. They applied the modified FIA to the reduced set and interpolate the missing $Q^{(0)}$ with $Q_0(\alpha_i) = -\sum_{t=1}^{l} Q^{(t)}(\alpha_i) r_i^t$ for all $i = 1, \ldots, n$.

C. Guruswami-Sudan case

We also consider for the Guruswami-Sudan [2] the complete set of $\binom{s+1}{2} n$ homogeneous equations without any reduction. This comes from the following lemma (proved in [4]):

Lemma 2 *Let $Q(x,y) = \sum_{t=0}^{l} Q^{(t)}(x) y^t$ be the Guruswami-Sudan interpolation polynomial that satisfies the conditions ①-③ for $s > 1$ and let $R(x)$, such that $R(\alpha_i) = r_i \; \forall i = 1, \ldots, n$. Furthermore let $G(x) = \prod_{j=1}^{n}(x - \alpha_i)$. Then $Q(x,y)$ satisfies condition ③, if and only if there exist s polynomials $B^{(b)}(x) \; \forall \, b = 0, \ldots, s-1$ over F for which*

$$Q^{[b]}(x, R(x)) = B^{(b)}(x) \cdot G(x)^{s-b}, \tag{8}$$

where $\deg B^{(b)}(x) < l(n-k) - s\tau + b$.

We remark that $Q^{[b]}$ denotes the b-th Hasse derivative of the bivariate polynomial $Q(x,y)$ with respect to the variable y.
It can be shown that the Equation (8) leads to a linear system $\mathbf{S} \cdot \mathbf{Q} = \mathbf{0}$, where the syndrome matrix $\mathbf{S}$ has the following form (for details see [4]):

$$\begin{pmatrix} \mathcal{S}^{(0,0)} & \mathcal{S}^{(0,1)} & \ldots & \ldots & \ldots & \mathcal{S}^{(0,l)} \\ 0 & \mathcal{S}^{(1,1)} & \ldots & \ldots & \ldots & \mathcal{S}^{(1,l)} \\ \vdots & & \ddots & & \vdots & \\ 0 & \ldots & 0 & \mathcal{S}^{(s-1,s-1)} & \ldots & \mathcal{S}^{(s-1,l)} \end{pmatrix} \cdot \tag{9}$$

All matrices depend on the received vector $\mathbf{r}$ except the ones on the diagonal $\mathcal{S}^{(i,i)} \;\; \forall \;\; i = 0, \ldots, s-1$. To solve the GSKE we adapt the FIA first for a single line of vertically arranged Hankel matrices (see next section) and then combine this algorithm with the idea of Roth and Ruckenstein to form an algorithm for a block Hankel matrix.
In [4] the system (9) was reduced by omitting some $\mathcal{S}^{(i,i)}$. For the sake of simplicity, we outline our algorithm for the non-reduced system of equations. Nevertheless, the algorithm is also applicable to the reduced system, where also a block Hankel matrix occurs.

V. Intermediate Step: Vertical Arrangement of Hankel matrices

In this section we derive an algorithm for determing the column vector $\mathbf{T}$ of the equation $\mathbf{S} \cdot \mathbf{T} = \mathbf{0}$ if $\mathbf{S}$ is matrix where s syndrome matrices are arranged vertically.

$$\mathbf{S} = \begin{pmatrix} \mathcal{S}^{(0)} & \mathcal{S}^{(1)} & \ldots & \mathcal{S}^{(s-1)} \end{pmatrix}^T$$

Each matrix $\mathcal{S}^{(i)}$ has a Hankel structure and consists of $(s-i) \cdot n$ rows and N columns.

Algorithm 1: Multiple Hankel matrices aranged vertically

Input: Polynomials $S^{(i)}(x)$, where $i = 0, \ldots, (s-1)$
Output: Polynomial $T(x)$

Data structures:
Column pointer ψ, row pointer (ϑ, κ)
Arrays A and D indexed with the row pointer (ϑ, κ)
Variable $\Delta \in F$, variable *compute* $\in \{\text{TRUE}, \text{FALSE}\}$.

Initialize:
Reset arrays A and D to zero $(\vartheta, \kappa) \leftarrow (0,0)$; $\rho \leftarrow 0$
compute $\leftarrow$ FALSE

```
1  while (ϑ, κ) < (s, n) do
2      if compute then
3          Δ ← ⟨x^κ · T(x), S^(ϑ)(x)⟩
4      else
5          if κ < 1 and ϑ = 0 then
6              T(x) ← x^ψ
7              Δ ← S_ψ^(ϑ)
8              (ϑ, κ)_V ← (0, 0)
9          else
10             T(x) ← x · T(x)
11             if κ = 0 then
12                 (ϑ, κ) ← (ϑ − 1, n)
13                 Δ ← 0
14             end
15             κ ← κ − 1
16         end
17         compute ← TRUE
18     end
19     if Δ = 0 or D[ϑ][κ] ≠ 0 then
20         if Δ ≠ 0 then
21             T(x) ← T(x) − (Δ / D[ϑ][κ]) · A[ϑ][κ](x)
22         end
23         (ϑ, κ) ≺ (ϑ, κ)
24     else
25         A[ϑ][κ](x) ← T(x); D[ϑ][κ] ← Δ
26         ψ ← ψ + 1
27         compute ← FALSE
28     end
29 end
```

Let $\prec_V$ denote the order over the set of pairs $\{(i,t) | i \in \{1, \ldots, s\}, t \in \mathbb{N}\}$, where $(i,t) \prec_V (i',t')$ if and only if:

$$\begin{gathered} t + i \cdot n < t' + i' \cdot n \\ \text{or} \\ t + i \cdot n = t' + i' \cdot s \text{ and } i < i' \end{gathered} \tag{10}$$

The rows of $\mathbf{S}$ are indexed with (ϑ, κ) where $0 \le \vartheta \le (s-1)$ and $0 \le \kappa < (s - \vartheta) \cdot n$. We rearrange the rows from top to bottom of the matrix $\mathbf{S}$ to the order $\prec_V$ on their indexes. A row in $\mathbf{S}$ is indexed with (ϑ, κ). For $s = 2$ we get the

following structure of the matrix $\mathbf{S}$:

$$\mathbf{S} = \left(\begin{array}{ccccc} \mathcal{S}_0^{(0)} & \mathcal{S}_1^{(0)} & \mathcal{S}_2^{(0)} & \cdots & \mathcal{S}_{N-1}^{(0)} \\ \mathcal{S}_1^{(0)} & \mathcal{S}_2^{(0)} & \mathcal{S}_3^{(0)} & \cdots & \mathcal{S}_N^{(0)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathcal{S}_n^{(0)} & \mathcal{S}_{n+1}^{(0)} & \mathcal{S}_{n+2}^{(0)} & \cdots & \mathcal{S}_{N+n-1}^{(0)} \\ \hline \mathcal{S}_0^{(1)} & \mathcal{S}_1^{(1)} & \mathcal{S}_2^{(1)} & \cdots & \mathcal{S}_{N-1}^{(1)} \\ \mathcal{S}_{n+1}^{(0)} & \mathcal{S}_{n+2}^{(0)} & \mathcal{S}_{n+3}^{(0)} & \cdots & \mathcal{S}_{N+n}^{(0)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathcal{S}_{2n-1}^{(0)} & \mathcal{S}_{2n}^{(0)} & \mathcal{S}_{2n+1}^{(0)} & \cdots & \mathcal{S}_{2n+N-1}^{(0)} \\ \mathcal{S}_{n-1}^{(1)} & \mathcal{S}_n^{(1)} & \mathcal{S}_{n+1}^{(1)} & \cdots & \mathcal{S}_{n+N-1}^{(1)} \end{array}\right) \quad (11)$$

We can reformulate the equation $\mathbf{S} \cdot \mathbf{T} = \mathbf{0}$ to a polynomial structure and get

$$\sum_{i=0}^{N-1} \left(\sum_{\kappa=0}^{((s-\vartheta)\cdot n-1)} T_i \cdot S_{\kappa+i}^{(\vartheta)} \right) = 0, \qquad 0 \le \vartheta \le (s-1) \quad (12)$$

or written with the inner product:

$$\langle x^\kappa \cdot T(x), S^{(\vartheta)}(x) \rangle = 0 \quad (13)$$

where $0 \le \vartheta \le (s-1)$ and $0 \le \kappa < (s-\vartheta) \cdot n$. Due to each matrix $\mathcal{S}^{(i)}$ has a Hankel structure we reformulate the FIA algorithm for Hankel structure as described in [5] to fit the particular structure of the Matrix $\mathbf{S}$. Algorithm 1 presents an algorithm for solving this particular structure. The algorithm allocates two arrays each indexed with the row pointer (ϑ, κ), array $A[\vartheta][\kappa]$ for buffering the polynomials and array $D[\vartheta][\kappa]$ for buffering the discrepancy values. The iteration of the main loop starts with a computation of the discrepancy Δ in Line 2. Line 21 updates the polynomial as described in [5]. Respectively to Equation (10) the row pointer is incremented in Line 23 after the polynomial was updated or the discrepancy was zero.

As it was the case with Hankel matrices, the idea is to start off a new column in $\mathbf{S}$ with a clever choice of an initial value of $T(x)$. Specifically suppose we are calculating a discrepancy at column ψ and row (ϑ, κ) where $\vartheta > 0$. We write this polynomial in the array $A[\vartheta][\kappa](x)$ and the current discrepancy in $D[\vartheta][\kappa]$ and increment the column pointer to $\psi + 1$. Like with Hankel matrices we select for our new polynomial $T(x) = x \cdot A[\vartheta][\kappa](x)$ when starting the column $\psi + 1$ and take the same discrepancy. Clearly we have

$$\langle x^i \cdot T(x), S^{(\vartheta)}(x) \rangle = \langle x^{i+1} \cdot A[\vartheta][\kappa](x), S^{(\vartheta)}(x) \rangle \quad (14)$$

for every $i \ge 1$. The initial value of $T(x)$ at column $\psi + 1$ already satisfies $\langle x^i \cdot T(x), S^{(j)}(x) \rangle = 0$ for all $(i, j) \prec (\vartheta, \kappa)$, which means that the discrepancy values are zero for all $(i, j) \prec (\vartheta, \kappa)$. Thus, we can start examining row $(\vartheta, \kappa - 1)$ in column $\psi + 1$ (see Lines 5 to 16).

The algorithm has one exception in Line 11, if we are calculating a discrepancy for $\vartheta \ge 1$ and $\kappa = 0$ we cannot decrement κ and we have to use the previous used ϑ, κ.

VI. A FIA for a block Hankel matrix

A. Principle

In this section we derive an algorithm for determing the $Q(x, y)$ of the GSKE. In this case we have a syndrome matrix as defined in Equation (9) which consists of multiple Hankel matrices. The columns of the matrix $\mathbf{S}$ are indexed with (ν, μ), where $0 \le \nu \le l$ and $0 \le \mu < N_\nu$ and the rows are indexed with (ϑ, κ), where $0 \le \vartheta \le (s-1)$ and $0 \le \kappa < (s-\vartheta) \cdot n$. We rearrange the columns from left to right with respect to the order $\prec_H$ on their indexes and the rows from top to bottom with respect to the order $\prec_V$ on their indexes. We can generalize Equation (13) for two bivariate polynomials to

$$\langle x^\kappa \cdot T(x, y), S^{(\vartheta)}(x, y) \rangle = 0, \quad (15)$$

where $0 \le \vartheta \le (s-1)$ and $0 \le \kappa < (s-\vartheta) \cdot n$.

Algorithm 2: Solving the GSKE

Input: Biv. Polynomials $S^{(i)}(x, y)$, $i = 0, \ldots, (s-1)$
Output: Biv. Polynomial $T(x, y)$

Data structures:
Column pointer (ν, μ), row pointer (ϑ, κ)
Arrays A and D indexed with the row pointer (ϑ, κ)
Array R for buffering the row pointer (ϑ, κ)
Variable $\Delta \in F$, variable *compute* $\in \{\text{TRUE}, \text{FALSE}\}$.

Initialize:
Reset arrays A, D and C to zero
$(\nu, \mu) \leftarrow (0, 0)$ and $(\vartheta, \kappa) \leftarrow (0, 0)$
compute $\leftarrow$ FALSE

```
1  while (ϑ,κ) < (s,n) do
2      if compute then
3          Δ ← ⟨x^κ · T(x,y), S^(ϑ)(x,y)⟩
4      else
5          if R[ν] < 1 then
6              T(x,y) ← y^ν · x^μ
7              Δ ← S_μ^(0,ν)
8              (ϑ,κ) ← (0,0)
9          else
10             T(x,y) ← x · A[R[ν]](x,y)
11             Δ ← D[R[ν]]
12             (ϑ,κ) ← R[ν]
13             if κ = 0 then
14                 (ϑ,κ) ← (ϑ − 1, n)
15                 Δ ← 0
16             end
17             κ ← κ − 1
18         end
19         compute ← TRUE
20     end
21     if Δ = 0 or D[ϑ][κ] ≠ 0 then
22         if Δ ≠ 0 then
23             T(x,y) ← T(x,y) − Δ/D[ϑ][κ] · A[ϑ][κ](x,y)
24         end
25         (ϑ,κ) ≺_V (ϑ,κ)
26     else
27         A[ϑ][κ](x,y) ← T(x,y)
28         D[ϑ][κ] ← Δ
29         R[ν] ← (ϑ,κ)
30         compute ← FALSE
31         (ν,μ) ≺_H (ν,μ)
32     end
33 end
```

Algorithm 2 solves this equation by combining the algorithm for the SKE defined in [3] and Algorithm 1. The iteration

of this algorithm increments the row pointer in Line 25 with respect to Equation (10) and the column pointer in Line 31 with respect to Equation (7). Line 3 presents the computation of the discrepancy as described in Equation (15). If the discrepancy is zero for all $0 \leq \vartheta \leq (s-1)$ and $0 \leq \kappa \leq (s-\vartheta) \cdot n - 1$ we fullfill this equation.

B. Complexity Analysis

The Algorithm 2 is tailored for a block Hankel matrix, where each submatrix $\mathcal{S}^{(b,t)}$ is a $((s-b) \cdot n) \times N_t$ Hankel matrix for all $b = 0, \ldots, s-1$ and $t = 0, \ldots, l$.

Proposition 1 *The time complexity of Algorithm 2 is* $\mathcal{O}\left(ls^4n^2\right)$.

Proof: By Equation (3), one iteration of the main loop of Algorithm 2 has time complexity $\mathcal{O}\left(s^2n\right)$.
For bounding the iterations of the main loop, we observe $\nu = t$ and $\vartheta = j$. For every $\nu = t$ we can increase μ at most N_t times. Therefore, κ can be decreased also at most N_t times. For every $\vartheta = j$ the initial value of κ is 0 and its final value cannot exceed $(s-j) \cdot n - 1$. It follows that the number of iterations cannot increase $(s-j) \cdot n - 1 + N_t$. Hence, the number of iterations with $\nu = t$ and $\vartheta = j$ is at most $(s-j) \cdot n - 1 + 2N_t$. Summing all over t and j, the number of iterations of the main loop is at most $\sum_{t=0}^{l} \left(\sum_{j=0}^{s-1} \left((s-j) \cdot n - 1 + 2N_t \right) \right) = \sum_{t=0}^{l} \left(\binom{s+1}{2} n + 2sN_t \right)$, which equals to the complexity $\mathcal{O}\left(l \cdot s^2 n + s \cdot s^2 n\right)$. With $l > s$ in case of Guruswami-Sudan we get the complexity of $\mathcal{O}\left(ls^2n\right)$. Thus, the overall time complexity of the algorithm is at most $\mathcal{O}\left(ls^4n^2\right)$. ■

VII. Conclusion

We proposed a generalization of the Fundamental Iterative Algorithm from Feng and Tzeng to a block Hankel matrix. The motivation was a fast realization of the interpolation problem of the Guruswami-Sudan principle, where the block Hankel structure of the set of linear homogeneous equations comes from the reformulation over a univariate polynomial ring (Key Equation).

The authors would like to thank Daniel Augot and Vladimir Sidorenko for the various fruitful discussions.

Appendix
Generalized FIA for the GSKE

The $\mathcal{RS}(16, 4, 13)$ code defined over $GF(17)$ with multiplicity $s = 2$ and the corresponding list size $l = 4$ is considered. The corresponding set of homogeneous equations coming from Equation (9) is a matrix 2×5 block Hankel matrix $\mathbf{S}$ with $3n = 48$ rows and $\sum_{t=0}^{l} N_t = 50$ columns:

$$\mathbf{S} = \begin{pmatrix} \mathcal{S}^{(0,0)} & \mathcal{S}^{(0,1)} & \mathcal{S}^{(0,2)} & \mathcal{S}^{(0,3)} & \mathcal{S}^{(0,4)} \\ 0 & \mathcal{S}^{(1,1)} & \mathcal{S}^{(1,2)} & \mathcal{S}^{(1,3)} & \mathcal{S}^{(1,4)} \end{pmatrix}.$$

Figure 1 shows the process of the row and column pointers of $\mathcal{S}^{(\vartheta,0)} \ldots \mathcal{S}^{(\vartheta,4)}$ with $0 \leq \vartheta \leq 1$ of the Algorithm 2. The ordering $\prec_H$ is identifiable, when we look at the vertical distances between two points of a graph. The horizontal distance for example between the two column pointers $(\nu = 0, \mu = 2)$ and $(\nu = 0, \mu = 3)$ in graph $\mathcal{S}^{(\vartheta,0)}$ is 1 whereas the distance increases to 2 between the column pointers $(\nu = 0, \mu = 5)$ and $(\nu = 0, \mu = 6)$ in the same graph. When we look at the vertical distances between two points, we can recognize the ordering $\prec_V$. In the rows 1 to 16 the vertical distance is 1 as marked in graph $\mathcal{S}^{(\vartheta,1)}$ with the row pointers $(\vartheta = 0, \kappa = 11)$ and $(\vartheta = 0, \kappa = 12)$. Between row 16 and 47 the vertical distance increases from 1 to 2 as marked in the same graph with the row pointers $(\vartheta = 1, \kappa = 9)$ and $(\vartheta = 0, \kappa = 10)$.

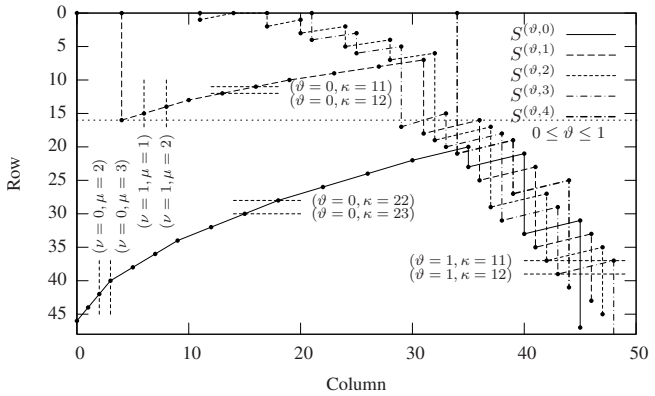

Fig. 1. Illustration of the column and row pointer of the generalized FIA (Algorithm 2) applied to an 2×5 block Hankel matrix.

References

[1] M. Sudan, "Decoding of Reed Solomon codes beyond the error-correction bound," *Journal of Complexity*, vol. 13, no. 1, pp. 180–193, 1997. [Online]. Available: http://dx.doi.org/http://dx.doi.org/10.1006/jcom.1997.0439

[2] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometry codes," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 1757–1767, 1999. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=782097

[3] R. M. Roth and G. Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance," *Information Theory, IEEE Transactions on*, vol. 46, no. 1, pp. 246–257, 2000. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=817522

[4] D. Augot and A. Zeh, "On the Roth and Ruckenstein Equations for the Guruswami-Sudan Algorithm," in *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, 2008, pp. 2620–2624. [Online]. Available: http://dx.doi.org/10.1109/ISIT.2008.4595466

[5] G. L. Feng and K. K. Tzeng, "A generalization of the Berlekamp-Massey algorithm for multisequence shift-register synthesis with applications to decoding cyclic codes," *Information Theory, IEEE Transactions on*, vol. 37, no. 5, pp. 1274–1287, 1991. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=133246

[6] R. Koetter and A. Vardy, "A complexity reducing transformation in algebraic list decoding of reed-solomon codes," in *Information Theory Workshop, 2003. Proceedings. 2003 IEEE*, 2003, pp. 10–13. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1216682

[7] J. Massey, "Shift-register synthesis and BCH decoding," *Information Theory, IEEE Transactions on*, vol. 15, no. 1, pp. 122–127, 1969. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1054260

[8] R. J. Mceliece, "The Guruswami–Sudan Decoding Algorithm for Reed–Solomon Codes," *Interplanetary Network Progress Report*, vol. 153, pp. 1–60, January 2003. [Online]. Available: http://adsabs.harvard.edu/cgi-bin/nph-bib_query?bibcode=2003IPNPR.153Q...1M

[9] G. Ruckenstein, "Error decoding strategies for algebraic codes," Ph.D. dissertation, Technion, 2001. [Online]. Available: http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-info.cgi/2001/PHD/PHD-2001-01