Decoding of interleaved Reed-Solomon codes using improved power decoding

Puchinger, Sven; Rosenkilde, Johan Sebastian Heesemann

Published in:
Proceedings of 2017 IEEE International Symposium on Information Theory

Link to article, DOI:
10.1109/ISIT.2017.8006549

Publication date:
2017

Document Version
Peer reviewed version

Decoding of Interleaved Reed–Solomon Codes Using Improved Power Decoding

Sven Puchinger[1], Johan Rosenkilde né Nielsen[2]

[1]Institute of Communications Engineering, Ulm University, Ulm, Germany
[2]Department of Applied Mathematics & Computer Science, Technical University of Denmark, Lyngby, Denmark
Email: *sven.puchinger@uni-ulm.de*, *jsrn@jsrn.dk*

***Abstract*—We propose a new partial decoding algorithm for m-interleaved Reed–Solomon (IRS) codes that can decode, with high probability, a random error of relative weight $1-R^{\frac{m}{m+1}}$ at all code rates R, in time polynomial in the code length n. For $m>2$, this is an asymptotic improvement over the previous state-of-the-art for all rates, and the first improvement for $R>1/3$ in the last 20 years. The method combines collaborative decoding of IRS codes with power decoding up to the Johnson radius.**

***Index Terms*—Interleaved Reed–Solomon Codes, Collaborative Decoding, Power Decoding with Multiplicities**

I. Introduction

An m-interleaved Reed–Solomon (IRS) code is a direct sum of m Reed–Solomon (RS) codes of the same evaluation points. The relevant metric is to consider *burst errors*, where an error corrupts the same position in all m codewords. In this metric, IRS codes can be decoded far beyond half the minimum distance of the constituent RS codes with high probability, and the decoding problem has achieved a lot of attention in the last decades, e.g. [1]–[6]. All of these decoders are *partial*, in that they will fail for a few error patterns of any weight beyond half the minimum distance of the weakest constituent code.

Often *homogeneous* IRS codes are considered for simplicity, i.e. where the constituent RS codes all have the same rate R. For $R>1/3$ and $m>2$, the state of the art is still given by [1] with a relative decoding radius of $\frac{m}{m+1}(1-R)$, see Figure 1. This decoding radius was echoed in [3] with a better complexity by solving the classical *key equations* for the constituent RS code simultaneously.

Power decoding [7] is a partial decoding algorithm for RS codes that can decode beyond half the minimum distance for $R\leq 1/3$, achieving roughly the same decoding radius as the Sudan list-decoder [8]. The idea is to generate several linearly independent key equations from one received word by a non-linear operation. In [9], it was proposed to use power decoding in IRS codes to obtain several key equations from each constituent received word, thereby increasing the decoding radius for $R\leq 1/3$. [4] refined this approach by mixing the received words of the IRS code in the powering process.

Recently [10], an improved power decoding algorithm for RS codes was proposed which is able to decode up to the Johnson radius. The gain is obtained by introducing multiplicities, similar to the Guruswami–Sudan algorithm, resulting in more linearly independent key equations.

In this paper, we apply the improved power decoding of [10] to IRS codes using the ideas of [4], [9]. We obtain a system of key equations and show how to efficiently solve them. We argue that the algorithm will decode with high probability up to a relative distance $1-R^{\frac{m}{m+1}}$, and we support this using simulations for a wide variety of parameters. For $m>2$,[1] this is an asymptotical improvement for all rates R over the previous best [4], [5] (cf. Figure 1 for the case $m=3$).

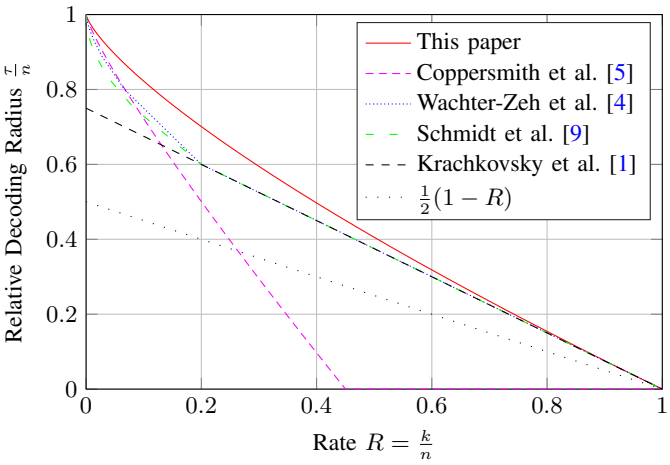

Figure 1. Comparison of Relative Decoding Radii for $m=3$.

The speed of our decoder depends on the sought performance: decoding up to relative distance $1-R^{\frac{m}{m+1}}-\varepsilon$, the complexity is $O^{\sim}(n(1/\varepsilon)^{m\omega+1})$, where $O^{\sim}$ means omission of logarithmic factors and ω is the exponent for matrix multiplication. This matches or improves upon all previous algorithms whenever the decoding radius is comparable.

The decoding algorithm has been implemented in SageMath v7.5 [11], and the source code can be downloaded from http://jsrn.dk/code-for-articles.

II. Preliminaries

Let q be a prime power and $\mathbb{F}_q$ a finite field of size q.

Definition 1: Let $\alpha_1,\dots,\alpha_n\in\mathbb{F}_q, k<n$. The corresponding *Reed–Solomon (RS) code* is the set

$$\mathcal{C}_{\mathrm{RS}}(n,k)=\{[f(\alpha_1),\dots,f(\alpha_n)]:f\in\mathbb{F}_q[x],\deg f<k\},$$

where we call f a *message polynomial* and the α_i's *evaluation points*. An *m-Interleaved Reed–Solomon (IRS) code* is the direct sum of m RS codes $\mathcal{C}_{\mathrm{RS}}(n,k_i)$ having the same evaluation points, i.e.,

$$\mathcal{C}_{\mathrm{IRS}}(n,k_1,\dots,k_m)=\left\{\begin{bmatrix}\boldsymbol{c}_1\\ \vdots\\ \boldsymbol{c}_m\end{bmatrix}:\boldsymbol{c}_i\in\mathcal{C}_{\mathrm{RS}}(n,k_i)\right\}\in\mathbb{F}_q^{m\times n}.$$

[1]In [6], an interpolation-based algorithm is proposed for $m=2$ that can achieve the same decoding radius as ours. It is claimed that it generalizes to $m>2$. However, no root-finding algorithm is given for the general case. See Section VII-A for more details.

A *homogeneous* IRS code is an IRS code whose m constituent codes have the same dimension $k_i = k$.

For simplifying of notation and analysis, we only consider homogeneous IRS codes in this paper.

As a metric for errors, we consider the *burst error metric*: if $\boldsymbol{r} = \boldsymbol{c}+\boldsymbol{e} \in \mathbb{F}_q^{m\times n}$ is received with $\boldsymbol{c} \in \mathcal{C}_{\mathrm{IRS}}(n,k;m)$, then the *error positions* are the non-zero columns of $\boldsymbol{e}$, i.e. $\mathcal{E} = \bigcup_{i=1}^{m}\{j : e_{i,j} \neq 0\}$, and the *number of errors* is $|\mathcal{E}|$. Burst errors occur naturally, for instance, if we transmit with a symmetric q^m-ary channel using the isomorphism $\mathbb{F}_q^{m\times n} \simeq (\mathbb{F}_q^m)^n$.

For a vector $\boldsymbol{i} = (i_1,\ldots,i_m) \in \mathbb{Z}_{\geq 0}^m$ we define its *size* as $|\boldsymbol{i}| := \sum_t i_t$. We denote by $\preceq$ the product partial order on $\mathbb{Z}_{\geq 0}^m$, i.e. $\boldsymbol{i} \preceq \boldsymbol{j}$ whenever $i_t \leq j_t$ for all t. The number of vectors $\boldsymbol{i} \in \mathbb{Z}_{\geq 0}^m$ with $|\boldsymbol{i}| = \mu$ is given by $\binom{m+\mu-1}{\mu}$. We will use the following well-known relations.

Lemma 1: Let $m, t \in \mathbb{Z}_{>0}$. Then,

$$\sum_{\mu=0}^{t}\binom{m+\mu-1}{\mu} = \binom{m+t}{m}, \text{ and } \sum_{\mu=0}^{t-1}\mu\binom{m+\mu-1}{\mu} = r\binom{m+t-1}{m+1}.$$

We also introduce the following notational short-hands:

Definition 2: For $\boldsymbol{a} \in \mathbb{F}_q[x]^m$, and $\boldsymbol{i}, \boldsymbol{j} \in \mathbb{Z}_{\geq 0}^m$, we define

$$\boldsymbol{a}^{\boldsymbol{i}} := \prod_{t=1}^{m} a_t^{i_t}, \quad \binom{\boldsymbol{j}}{\boldsymbol{i}} := \prod_{t=1}^{m}\binom{j_t}{i_t}.$$

We then have a vectorised binomial theorem:

Lemma 2: Let $r \in \mathbb{Z}_{>0}$, $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{F}_q[x]^m$, and $\boldsymbol{j} \in \mathbb{Z}_{\geq 0}^m$. Then,

$$(\boldsymbol{a}+\boldsymbol{b})^{\boldsymbol{j}} = \sum_{\boldsymbol{i}\preceq\boldsymbol{j}}\binom{\boldsymbol{j}}{\boldsymbol{i}}\boldsymbol{a}^{\boldsymbol{i}}\boldsymbol{b}^{\boldsymbol{j}-\boldsymbol{i}}.$$

III. Key Equations

In the following, we develop "powered key equations" for IRS codes similar to [10]. As in [10], we use the formulation of Power decoding introduced in [12].

Consider a decoding instance, i.e. that $\boldsymbol{r} = \boldsymbol{c}+\boldsymbol{e} \in \mathbb{F}_q^{m\times n}$ has been received with $\boldsymbol{c} \in \mathcal{C}_{\mathrm{IRS}}(n,k;m)$ and $\mathcal{E}$ the corresponding error positions. Let $\boldsymbol{f} = (f_1,\ldots,f_m) \in \mathbb{F}_q[x]_{<k}^m$ be the message polynomials corresponding to $\boldsymbol{c}$.

Definition 3: The *error locator polynomial* is defined by

$$\Lambda := \textstyle\prod_{i\in\mathcal{E}}(x-\alpha_i).$$

Definition 4: For $t = 1,\ldots,m$, let $R_t \in \mathbb{F}_q[x]$ be the unique interpolation polynomial of degree $\deg R_t < n$ such that

$$R_t(\alpha_i) = r_{t,i} \quad \forall i = 1,\ldots,n.$$

We write $\boldsymbol{R} = (R_1,\ldots,R_m)$. Let $G \in \mathbb{F}_q[x]$ be given by

$$G := \textstyle\prod_{i=1}^{n}(x-\alpha_i).$$

Note that Λ is not known at the receiver, but $\boldsymbol{R}$ and G are. We can relate the polynomials as follows.

Lemma 3: Let G and $\boldsymbol{R}$ be as in Definition 4. Then G divides each element of $\Lambda(\boldsymbol{f}-\boldsymbol{R})$.

Proof: For all $i = 1,\ldots,m$ and $j = 1,\ldots,n$, we obtain

$$\big(\Lambda(f_i - R_i)\big)(\alpha_j) = \Lambda(\alpha_j)\cdot(-e_{i,j}) = \begin{cases} 0\cdot(-e_{i,j}), & j\in\mathcal{E} \\ \Lambda(\alpha_j)\cdot 0, & \text{else.}\end{cases}$$

Thus, $(x-\alpha_j) \mid \Lambda(f_i - R_i)$ and the claim follows. ∎

In light of the above lemma, let

$$\boldsymbol{\Omega} := \Lambda(\boldsymbol{f}-\boldsymbol{R})/G \in \mathbb{F}_q[x]^m.$$

Following nomenclature from RS decoding, the $\boldsymbol{\Omega}$ is known as *error evaluators*. Note that the entries of $\boldsymbol{\Omega}$ each have degree less than $|\mathcal{E}|$.

We can now state the set of key equations that we will use for decoding. It consists of $\binom{m+\ell}{m} - 1$ many relations between the unknown polynomials Λ, $\boldsymbol{f}$, and $\boldsymbol{\Omega}$.

Theorem 4 (Key Equations): Choose any $s, \ell \in \mathbb{Z}_{>0}$ with $s \leq \ell$. Let Λ, G, $\boldsymbol{R}$, $\boldsymbol{f}$, and $\boldsymbol{\Omega}$ be as above. Then, for all $\boldsymbol{j} \in \mathbb{Z}_{\geq 0}^m$ with $1 \leq |\boldsymbol{j}| \leq \ell$, we have

$$\Lambda^s\boldsymbol{f}^{\boldsymbol{j}} = \sum_{\boldsymbol{i}\preceq\boldsymbol{j}}\left[\Lambda^{s-|\boldsymbol{i}|}\boldsymbol{\Omega}^{\boldsymbol{i}}\right]\left[\binom{\boldsymbol{j}}{\boldsymbol{i}}\boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}}G^{|\boldsymbol{i}|}\right], \qquad |\boldsymbol{j}| < s$$

$$\Lambda^s\boldsymbol{f}^{\boldsymbol{j}} \equiv \sum_{\substack{\boldsymbol{i}\preceq\boldsymbol{j}\\ |\boldsymbol{i}|<s}}\left[\Lambda^{s-|\boldsymbol{i}|}\boldsymbol{\Omega}^{\boldsymbol{i}}\right]\left[\binom{\boldsymbol{j}}{\boldsymbol{i}}\boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}}G^{|\boldsymbol{i}|}\right] \mod G^s, \quad |\boldsymbol{j}| \geq s.$$

Proof: Using Lemma 2, we can write

$$\Lambda^s\boldsymbol{f}^{\boldsymbol{j}} = \Lambda^s\big((\boldsymbol{f}-\boldsymbol{R})+\boldsymbol{R}\big)^{\boldsymbol{j}} = \Lambda^s\sum_{\boldsymbol{i}\preceq\boldsymbol{j}}\binom{\boldsymbol{j}}{\boldsymbol{i}}(\boldsymbol{f}-\boldsymbol{R})^{\boldsymbol{i}}\boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}}.$$

For $|\boldsymbol{i}| < s$, we rewrite

$$\Lambda^s(\boldsymbol{f}-\boldsymbol{R})^{\boldsymbol{i}} = \Lambda^{s-|\boldsymbol{i}|}[\Lambda(\boldsymbol{f}-\boldsymbol{R})]^{\boldsymbol{i}} = \Lambda^{s-|\boldsymbol{i}|}\boldsymbol{\Omega}^{\boldsymbol{i}}G^{|\boldsymbol{i}|}.$$

If $|\boldsymbol{i}| \geq s$, we decompose $\boldsymbol{i} = \boldsymbol{i}_s + \boldsymbol{i}'$, where $\boldsymbol{i}_s, \boldsymbol{i}' \in \mathbb{Z}_{\geq 0}^m$ with $|\boldsymbol{i}_s| = s$. Using this notation, we obtain

$$\Lambda^s(\boldsymbol{f}-\boldsymbol{R})^{\boldsymbol{i}} = [\Lambda(\boldsymbol{f}-\boldsymbol{R})]^{\boldsymbol{i}_s}(\boldsymbol{f}-\boldsymbol{R})^{\boldsymbol{i}'} = G^s\boldsymbol{\Omega}^{\boldsymbol{i}_s}(\boldsymbol{f}-\boldsymbol{R})^{\boldsymbol{i}'},$$

so all summands $\boldsymbol{i}$ with $|\boldsymbol{i}| \geq s$ are zero modulo G^s. Hence, the equation and congruence above hold. ∎

We use the following abbreviations in the remaining sections.

$$\Psi_{\boldsymbol{j}} := \Lambda^s\boldsymbol{f}^{\boldsymbol{j}}, \ \Lambda_{\boldsymbol{i}} := \Lambda^{s-|\boldsymbol{i}|}\boldsymbol{\Omega}^{\boldsymbol{i}}, \text{ and } A_{\boldsymbol{i},\boldsymbol{j}} := \tbinom{\boldsymbol{j}}{\boldsymbol{i}}\boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}}G^{|\boldsymbol{i}|}.$$

IV. Solving the Key Equations

The key equations of Theorem 4 are non-linear relations between the unknowns $\Lambda, \boldsymbol{f}$ and $\boldsymbol{\Omega}$ and therefore a priori difficult to solve. The approach, as for classical key equation decoding, is to relax the non-linear relations to—seemingly much weaker—linear relations, and then hope that a minimal solution to these is the sought. The linear problem that we will relax into is a very general variant of Padé approximations:

Problem 5: [Simultaneous Hermite Padé approximation] Given $S_{i,j}, G_j \in \mathbb{F}_q[x]$ for all $i \in I, j \in J$ for index sets I, J as well as degree bounds $N_i, T_j \in \mathbb{Z}_{\geq 0}$, compute, if it exists, $\lambda_i, \psi_j \in \mathbb{F}_q[x]$, not all zero and such that

$$\sum_{i\in I}\lambda_i S_{i,j} \equiv \psi_j \mod G_j, \quad \forall j \in J\ ,$$

as well as $\deg\lambda_i < N_i$ and $\deg\psi_j < T_j$ for all $i \in I, j \in J$.

Such λ_i, ψ_j is a *solution* to the Padé approximation. If furthermore $\lambda_{\boldsymbol{0}} \neq 0$ and has minimal degree for a specific $\boldsymbol{0} \in I$, we say that it is a *minimal solution*.

Proposition 6: Let $\tau \geq \deg(\Lambda)$ and $\Lambda_{\boldsymbol{i}}, \Psi_{\boldsymbol{j}}$ and $A_{\boldsymbol{i},\boldsymbol{j}}$ be defined as in Section III. Then $\Lambda_{\boldsymbol{i}}, \Psi_{\boldsymbol{j}}$ is a solution to the simultaneous Hermite Padé approximation given by:

$$I = \{\boldsymbol{i} \in \mathbb{Z}_{\geq 0}^m, |\boldsymbol{i}| < s\} \qquad J = \{\boldsymbol{j} \in \mathbb{Z}_{\geq 0}^m, |\boldsymbol{j}| \leq \ell\}$$

$$S_{\boldsymbol{i},\boldsymbol{j}} = A_{\boldsymbol{i},\boldsymbol{j}} \qquad G_{\boldsymbol{j}} = \begin{cases} x^{\tau s+|\boldsymbol{j}|(n-1)+1} & |\boldsymbol{j}| < s \\ G^s & |\boldsymbol{j}| \geq s\end{cases}$$

$$N_{\boldsymbol{i}} = \tau s - |\boldsymbol{i}| + 1 \qquad T_{\boldsymbol{j}} = \tau s + |\boldsymbol{j}|(k-1) + 1\ ,$$

and leader index $\boldsymbol{0}$ (all-zero vector).

Proof: That $\Lambda_{\boldsymbol{i}}, \Psi_{\boldsymbol{j}}$ satisfy the degree constraints follows from $\tau \geq \deg(\Lambda)$ and $\deg \boldsymbol{f}^{\boldsymbol{j}} \leq |\boldsymbol{j}|(k-1)$. For $|\boldsymbol{j}| \geq s$ the Padé approximation congruence is that of Theorem 4, and for $|\boldsymbol{j}| < s$, the congruence is satisfied since it is an equality in Theorem 4. ∎

Remark 7: The modulus $x^{\tau s+|\boldsymbol{j}|(n-1)+1}$ in Proposition 6 arise as $1 + \deg\left(\sum_{|\boldsymbol{i}|<s} \lambda_{\boldsymbol{i}} A_{\boldsymbol{i},\boldsymbol{j}}\right)$ for $|\boldsymbol{j}| < s$.

In other words, solving the simultaneous Hermite Padé approximation given by Proposition 6 will, up to a scalar multiple, recover $\Lambda, \boldsymbol{f}$ and $\boldsymbol{\Omega}$ if $\Psi_{\boldsymbol{j}}$ and $\Lambda_{\boldsymbol{i}}$ is the *minimal* solution. The decoding algorithm is built around exactly this expectation. Formally, the Power-IRS decoding algorithm is as follows:

1) Compute a *minimal* solution $(\lambda_{\boldsymbol{i}})_{\boldsymbol{i}}, (\psi_{\boldsymbol{j}})_{\boldsymbol{j}}$ to the simultaneous Hermite Padé approximation given by Proposition 6, minimising $\deg \lambda_{\boldsymbol{0}}$.
2) Let $f_t := \psi_{\boldsymbol{u}_t}/\lambda_{\boldsymbol{0}}$ for each $t = 1, \ldots, m$, where the $\boldsymbol{u}_t$ is the vector with 0 everywhere but a 1 at index t. If these are not all polynomials of degree less than k, declare fail.
3) If $\mathrm{wt}(\boldsymbol{r} - \boldsymbol{c}) \leq \tau$, return $\boldsymbol{c}$, where $\boldsymbol{c}$ is the codeword of the message polynomials $(f_1, \ldots, f_m)$ and $\mathrm{wt}(\boldsymbol{x})$ is the number of non-zero columns of $\boldsymbol{x}$. Otherwise declare fail.

We discuss in later sections when we can expect this algorithm to work. For now we turn to the complexity of the decoder. Solving variants of Padé approximation problems has a long and lustrous history in both coding theory and computer algebra. In particular, [13] gives the currently fastest algorithm for our case. The very general algorithm of [14] can also be used but is slightly slower. See also these papers for a discussion of the history of computing Padé approximations. We obtain:

Proposition 8: Given a homogeneous IRS code $\mathcal{C}_{\mathrm{IRS}}$ and $s, \ell \in \mathbb{Z}_{>0}$, the Power-IRS decoding algorithm can be performed in $O^{\sim}\left(ns\binom{m+\ell}{m}^{\omega-1}\binom{m+s-1}{m}\right)$ operations in $\mathbb{F}_q$.

Proof: The $\binom{m+\ell}{m}\binom{m+s-1}{m}$ values $A_{\boldsymbol{i},\boldsymbol{j}}$ can easily be computed in the target cost once we know $\boldsymbol{R}^{\boldsymbol{i}}$ mod G^s for all $\boldsymbol{i}$ with $|\boldsymbol{i}| \leq \ell$. These are computed using dynamic programming in time $O^{\sim}(sn)$. Solving the Padé approximation is the only other expensive step and has the target cost by using [13]. ∎

V. Decoding Performance

One cannot hope to improve upon single-RS decoding in the worst case scenario: the sent codeword and error could be the same in each constituent code, so the interleaving gives no new information. Similarly, list decoding IRS codes cannot go beyond list decoding RS codes.

As previous IRS decoders, Power-IRS sidesteps this issue by being a *partial decoder* whose capability is characterised by two concepts: 1) the number of errors we expect to decode, which we will call the "*decoding radius*"; and 2) the *probability of success* for a given number of errors, assuming uniformly random error patterns of a given weight.

Solving the simultaneous Hermite Padé approximation of Proposition 6 will lead to finding $\boldsymbol{\Lambda}, \boldsymbol{f}$ and $\boldsymbol{\Omega}$ if there are not other solutions to the approximation. In Section V-A we define the decoding radius as just below the number of errors when "generic" solutions are sure to appear, effectively ruling out that we find the special solution. In Section V-B we prove that success or failure is dictated only by the error, and not the sent codeword. We then turn to the success probability: we have no formal bounds, but we discuss what we believe are the main contributing factors and present a conjecture. The conjecture is backed by simulations in Section VI.

A. Decoding Radius

Lemma 9: Consider a simultaneous Hermite Padé approximation (Problem 5). The approximation is guaranteed to have at least two $\mathbb{F}_q$-linearly independent non-zero solutions whenever

$$\textstyle\sum_{i\in I} N_i > 1 + \sum_{j\in J}(\deg G_j - T_j)$$

Proof: The degree restrictions on the remainders ψ_j can be considered as $\mathbb{F}_q$-linear restrictions in the coefficients of the λ_i: that $\deg \psi_j < T_j$ implies that the coefficients to $x^{T_j}, \ldots, x^{\deg G_j - 1}$ are zero in the corresponding $\mathbb{F}_q[x]$-linear expression in the λ_i. This gives $\sum_{i\in I} N_i$ indeterminates and at least $\sum_{j\in J}(\deg G_j - T_j)$ restrictions over $\mathbb{F}_q$. ∎

Theorem 10: The simultaneous Hermite Padé approximation of Proposition 6 is guaranteed to have a solution which is $\mathbb{F}_q$-linearly independent from $\Lambda_{\boldsymbol{i}}, \Psi_{\boldsymbol{j}}$ whenever $\tau > \tau_{\mathrm{new}}$, where

$$\tau_{\mathrm{new}} := n\left[1 - \frac{s\binom{m+s-1}{m} - m\binom{m+s-1}{m+1}}{s\binom{m+\ell}{m}}\right] - \frac{m}{m+1}\frac{\ell}{s}(k-1) - \frac{1}{s}\left[1 - \frac{1}{\binom{m+\ell}{m}}\right]$$

Proof: See Appendix A. ∎

Following the discussion above, we call τ_{new} the *decoding radius* of the Power-IRS decoder. In the following asymptotic analyses, $O_{R,m}$ means that R and m are considered constants (i.e. hidden).

Theorem 11: Let $(\ell_i, s_i) = (i, \lfloor \gamma i \rfloor + 1)$ for $i \in \mathbb{Z}_{>0}$, where $\gamma = \sqrt[m+1]{\frac{k-1}{n}}$. Then,

$$\tau_{\mathrm{new}}(\ell_i, s_i) = n\Big(1 - \big(\tfrac{k-1}{n}\big)^{\frac{m}{m+1}} - O_{R,m}\big(\tfrac{1}{i}\big)\Big).$$

Proof: See Appendix A. ∎

Corollary 12: For a fixed code and any constant $\varepsilon > 0$, we can choose $s, \ell \in O_{R,m}(1/\varepsilon)$ such that $\tau_{\mathrm{new}} \geq n(1 - R^{\frac{m}{m+1}} - \varepsilon)$ where $R = \frac{k}{n}$.

In this case, the algorithm has complexity $O^{\sim}\big(n(1/\varepsilon)^{m\omega+1}\big)$.

B. Expecting Successful Decoding

Now we turn to the probability that Power-IRS will succeed. We start with the observation that decoding success is independent of the sent codeword.

Theorem 13: The success of decoding $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$ depends only on the error $\boldsymbol{e}$.

Proof: The proof closely resembles the one of [10, Proposition 6]. It suffices to show that if decoding $\boldsymbol{r}$ fails, then decoding $\boldsymbol{r} + \hat{\boldsymbol{c}}$ for any $\hat{\boldsymbol{c}} \in \mathcal{C}_{\mathrm{IRS}}$ also fails. Let $\hat{\boldsymbol{c}}$ correspond to the message polynomial $\hat{\boldsymbol{f}}$. If decoding $\boldsymbol{r}$ fails, there is a solution $\lambda_{\boldsymbol{i}}, \psi_{\boldsymbol{j}}$ to the Padé approximation of Proposition 6 for received word $\boldsymbol{r}$ with $\lambda_{\boldsymbol{0}} \neq \Lambda^s$ and $\deg \lambda_{\boldsymbol{0}} \leq \deg \Lambda^s$. We claim that $\hat{\psi}_{\boldsymbol{j}} := \sum_{\boldsymbol{i}\preceq\boldsymbol{j}} \binom{\boldsymbol{j}}{\boldsymbol{i}} \hat{\boldsymbol{f}}^{\boldsymbol{i}} \psi_{\boldsymbol{j}-\boldsymbol{i}}$ and $\hat{\lambda}_{\boldsymbol{i}} := \lambda_{\boldsymbol{i}}$ is a solution to the approximation problem for received word $\boldsymbol{r} + \hat{\boldsymbol{c}}$, so that decoding $\boldsymbol{r} + \hat{\boldsymbol{c}}$ also fails. Let $\hat{\boldsymbol{R}} := \boldsymbol{R} + \hat{\boldsymbol{f}}$. Then the required congruences are satisfied since

$$\begin{aligned}
\sum_{\substack{\boldsymbol{i}\preceq\boldsymbol{j}\\ |\boldsymbol{i}|<s}} \hat{\lambda}_{\boldsymbol{i}}\binom{\boldsymbol{j}}{\boldsymbol{i}}\hat{\boldsymbol{R}}^{\boldsymbol{j}-\boldsymbol{i}}G^{|\boldsymbol{i}|} &= \sum_{\substack{\boldsymbol{i}\preceq\boldsymbol{j}\\ |\boldsymbol{i}|<s}} \sum_{\boldsymbol{h}\preceq\boldsymbol{j}-\boldsymbol{i}} \lambda_{\boldsymbol{i}} \underbrace{\binom{\boldsymbol{j}}{\boldsymbol{i}}\binom{\boldsymbol{j}-\boldsymbol{i}}{\boldsymbol{h}}}_{=\binom{\boldsymbol{j}}{\boldsymbol{h}}\binom{\boldsymbol{j}-\boldsymbol{h}}{\boldsymbol{i}}} \boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}-\boldsymbol{h}}\hat{\boldsymbol{f}}^{\boldsymbol{h}}G^{|\boldsymbol{i}|} \\
&= \sum_{\boldsymbol{h}\preceq\boldsymbol{j}} \binom{\boldsymbol{j}}{\boldsymbol{h}}\hat{\boldsymbol{f}}^{\boldsymbol{h}} \sum_{\substack{\boldsymbol{i}\preceq\boldsymbol{j}-\boldsymbol{h}\\ |\boldsymbol{i}|<s}} \lambda_{\boldsymbol{i}}\binom{\boldsymbol{j}-\boldsymbol{h}}{\boldsymbol{i}}\boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{h}-\boldsymbol{i}}G^{|\boldsymbol{i}|} \\
&\equiv \sum_{\boldsymbol{h}\preceq\boldsymbol{j}} \binom{\boldsymbol{j}}{\boldsymbol{h}}\hat{\boldsymbol{f}}^{\boldsymbol{h}}\psi_{\boldsymbol{j}-\boldsymbol{h}} = \hat{\psi}_{\boldsymbol{j}} \mod G_{\boldsymbol{j}}
\end{aligned}$$

Also, $\deg \hat{\psi}_{\boldsymbol{j}} \leq \min_{\boldsymbol{h}}\{\deg \psi_{\boldsymbol{j}-\boldsymbol{h}} + |\boldsymbol{h}|(k-1)\} \leq \tau s + |\boldsymbol{j}|(k-1) = T_{\boldsymbol{j}}$. ∎

Turning to the probability of success. There are two mechanisms we should expect cause Power-IRS to fail on reception of $\boldsymbol{r} = \boldsymbol{e} + \boldsymbol{c}$: 1) the simultaneous Hermite Padé approximation has a "spurious", non-coding theoretic related solution; and 2) there is a codeword $\boldsymbol{c}' \neq \boldsymbol{c}$ with $\mathrm{wt}(\boldsymbol{r} - \boldsymbol{c}') \leq \mathrm{wt}(\boldsymbol{r} - \boldsymbol{c})$. The former probability can be characterised for "random" Padé approximations (i.e. random $S_{i,j}$ polynomials), and the latter can easily be estimated using classical coding theory. Both are exponentially decaying in the value $(\tau_{\mathrm{new}} - \varepsilon)$, where ε is the number of errors. We expected that these were essentially the only contributing factors to decoding failure; our simulations indicate that the observed failure rate are often, but not always, very close to the union bound of the above two probabilities.

This important question needs more investigation. For now, based on the observed failure rates, and the formal bounds on success probability for Power decoding RS codes, we conjecture that the probability that Power-IRS succeeds on an error $\boldsymbol{e}$ with ε non-zero columns can be lower bounded by $1 - q^{-b(\tau_{\mathrm{new}} - \varepsilon)}$ for some constant $b > 1$ that depends on the code and decoder parameters n, k, m, ℓ, and s.

VI. Simulations

We implemented the decoding algorithm in SageMath v7.5 [11] and observed the decoding behaviour on random errors for a range of parameters. The source code can be downloaded from http://jsrn.dk/code-for-articles.

Some of the results are outlined in Table I. For instance, the $\mathcal{C}_{\mathrm{IRS}}(257, 86; 2)$ code over $\mathbb{F}_{257}$ can be decoded by previous algorithms up to 114 errors. Our algorithm is able to decode up to 124 errors by choosing $(\ell, s) = (4, 3)$ and we observed failure in $\approx 1.1 \cdot 10^{-5}$ fraction of the trials.

Over $\mathbb{F}_{17}$, we can decode up to 13 errors with an observed failure rate of $\approx 1.9 \cdot 10^{-3}$ using our decoder for the code $\mathcal{C}_{\mathrm{IRS}}(17, 3; 5)$ with parameters $(5, 3)$ (instead of 11 using [4]). Note that this is very close to $n - k = 14$.

We also compared our decoder to the example in [4, Table 1]: An $\mathcal{C}_{\mathrm{IRS}}(16, 2; 3)$ code over $\mathbb{F}_{17}$. The decoder in [4] can decode up to 12 errors with observed failure rate $6.2 \cdot 10^{-3}$. By choosing the parameters of our decoder $(3, 2)$, we observe a lower failure rate $9.1 \cdot 10^{-5}$ for the same number of errors. By further increasing the parameters, we decode one more error with $\approx 90\%$ probability.

In general, the observed failure rate $\hat{\mathrm{P}}_{\mathrm{fail}}$ rapidly decreases with the number of errors, backing up the conjecture of the previous section. Also, whenever the number of errors was above the radius, decoding always failed.

Table I
Simulation results. Code parameters q, n, k, m , Decoder parameters (ℓ, s), Number of samples N. Observed Failure Rate $\hat{\mathrm{P}}_{\mathrm{fail}}$. For comparison: Decoding radii τ_{WZB} of [4] and τ_{KL} of [1] for the chosen parameters.

q	n	k	m	(ℓ, s)	τ_{new}	$\hat{\mathrm{P}}_{\mathrm{fail}}$	τ_{WZB}	τ_{KL}	N
257	257	86	2	$(3, 2)$	120	0	114	114	10^6
				$(4, 3)$	124	$1.1 \cdot 10^{-5}$	114	114	10^6
43	43	18	2	$(4, 3)$	18	$5.4 \cdot 10^{-4}$	16	16	10^6
17	17	3	2	$(3, 2)$	11	$1.9 \cdot 10^{-3}$	10	9	10^6
			4	$(4, 3)$	13	$2.8 \cdot 10^{-2}$	10	9	10^4
			5	$(5, 3)$	13	$1.9 \cdot 10^{-3}$	10	9	10^4
17	16	2	3	$(3, 2)$	12	$9.1 \cdot 10^{-5}$	12	10	10^6
				$(6, 3)$	13	$1.0 \cdot 10^{-1}$	12	10	10^4
16	16	3	3	$(2, 1)$	10	0	10	9	10^6
				$(3, 2)$	11	$2.1 \cdot 10^{-5}$	10	9	10^6

VII. Comparison to Related Work

We now compare the asymptotic decoding performance of Power-IRS to other IRS decoders and Folded RS codes.

A. Other Decoding Algorithms for Interleaved RS Codes

Using the strict generalized Bernoulli inequality, we obtain

$$\tfrac{m}{m+1}(1 - R) < 1 - R^{\frac{m}{m+1}}$$

for all rates $R \in (0, 1)$ and interleaving degrees $m \geq 1$. Hence, our algorithm improves upon the algorithms in [1], [3] at all rates. For $R > \frac{1}{3}$ and $m > 2$, this is the first improvement since 1997 [1]. For $R < \frac{1}{3}$, the previous best are the CS [5] and WZB [4] algorithms (the WZB algorithm strictly improves upon the KL [1] and SSB [9] algorithm). The relative decoding radius of the CS algorithm is $\frac{\tau_{\mathrm{CS}}}{n} \to 1 - R^{\frac{m}{m+1}} - R$ $(n \to \infty)$ [5, Theorem 1], which is strictly smaller than our $1 - R^{\frac{m}{m+1}}$. The WZB algorithm is a special case of Power-IRS where $s = 1$. As demonstrated in Figure 1 on the first page for $m = 3$, the additional degree of freedom of s greatly increases the decoding capability. Note also that for $m = 1$, Power-IRS reduces to the algorithm in [10] and decodes to the Johnson radius as in [15].

In his PhD thesis [6, Chapter 2], Parvaresh described a decoding algorithm for interleaved RS codes based on multivariate interpolation, which can be seen as a generalization of the Guruswami–Sudan algorithm. The existence of a suitable $(m+1)$-variate interpolation polynomial is guaranteed if the relative number of errors is smaller than $1 - R^{\frac{m}{m+1}}$. However, the root-finding step is only described for the case $m = 2$ (cf. [6, Section 2.5]). To the best of our knowledge, no subsequent work on this topic was published by the author, so the case $m > 2$ remains an open problem.

Cohn and Heninger [16] proposed an algorithm for noisy multi-polynomial reconstruction, which is closely related to decoding IRS codes. More precisely, [16, Theorem 3] implies that decoding is successful if the fraction of errors is less than $1 - R^{\frac{m}{m+1}}$, given that certain polynomials, which depend on the algorithm's input, are algebraically independent. The paper does not state under which conditions this so-called algebraic independence hypothesis holds. In future work, the performance of this decoder must be evaluated more thoroughly. In any case, our algorithm seems to result in a much smaller complexity due to the non-existence of the heavy root-finding step, which requires resultant or Gröbner-basis computations in [16].

B. Folded RS Codes

m-folded Reed–Solomon codes are very similar to IRS codes and are interesting since they allow list-decoding beyond the Johnson radius [17, Theorem 4.4]: for any $\varepsilon > 0$, there is a family of m-folded RS codes of rate R for which a fraction of $1 - R - \varepsilon$ errors can be list-decoded, where $m \in O(1/\varepsilon^2)$.

In comparison, our results indicate that for any $\varepsilon > 0$, any m-interleaved RS code of rate R with $m = \frac{\log(1/(R+\varepsilon))}{\log(1+\varepsilon/R)} \leq \frac{\log(1/R)}{\varepsilon/R} \in O(1/\varepsilon)$ can correct a fraction of

$$1 - R^{\frac{m}{m+1}} = 1 - R^{\frac{\log(R+\varepsilon)}{\log(R)}} = 1 - R - \varepsilon$$

errors. Hence the decoding radius of Power-IRS convergences much faster to capacity $1-R$ with respect to the interleaving degree (equivalently the field size q^m). It should be kept in mind that the algorithm in [17] is a list decoder, so it guarantees to return a list of codewords containing the sent codeword, which is a much stronger guarantee than Power-IRS.

Another related capacity-achieving construction is univariate multiplicity codes [18] (also called Derivative codes) which has the same convergence rate $m \approx O(1/\varepsilon^2)$ as Folded RS codes.

VIII. Conclusion

We have introduced a new partial decoding algorithm for m-Interleaved Reed–Solomon codes that can decode, with seemingly high probability, up to $n(1 - R^{\frac{m}{m+1}} - \varepsilon)$ errors with complexity quasi-linear in n and polynomial in $1/\varepsilon$. Our decoding radius is motivated by essentially assuming that the underlying linear system is random, but simulations on a variety of parameters strongly back up this value. Formally bounding the success probability remains an open problem.

Acknowledgement

We would like to thank Hannes Bartz for making us aware of [6] and Vincent Neiger for pointing us at [16].

References

[1] V. Y. Krachkovsky and Y. X. Lee, "Decoding for Iterative Reed–Solomon Coding Schemes," *IEEE Trans. Magn.*, vol. 33, no. 5, pp. 2740–2742, 1997.

[2] D. Bleichenbacher, A. Kiayias, and M. Yung, "Decoding of Interleaved Reed Solomon Codes Over Noisy Data," in *ICALP*. Springer, 2003, pp. 97–108.

[3] G. Schmidt, V. R. Sidorenko, and M. Bossert, "Collaborative Decoding of Interleaved Reed–Solomon Codes and Concatenated Code Designs," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 2991–3012, 2009.

[4] A. Wachter-Zeh, A. Zeh, and M. Bossert, "Decoding Interleaved Reed–Solomon Codes Beyond Their Joint Error-Correcting Capability," *Designs, Codes and Cryptography*, vol. 71, no. 2, pp. 261–281, 2014.

[5] D. Coppersmith and M. Sudan, "Reconstructing Curves in Three (and Higher) Dimensional Space from Noisy Data," in *ACM STOC*, 2003.

[6] F. Parvaresh, "Algebraic List-Decoding of Error-Correcting Codes," Ph.D. dissertation, University of California, San Diego, 2007.

[7] G. Schmidt, V. R. Sidorenko, and M. Bossert, "Syndrome Decoding of Reed–Solomon Codes Beyond Half the Minimum Distance Based on Shift-Register Synthesis," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5245–5252, 2010.

[8] M. Sudan, "Decoding of Reed–Solomon Codes Beyond the Error-Correction Bound," *J. Complexity*, vol. 13, no. 1, pp. 180–193, 1997.

[9] G. Schmidt, V. Sidorenko, and M. Bossert, "Enhancing the Correcting Radius of Interleaved Reed-Solomon Decoding using Syndrome Extension Techniques," in *IEEE ISIT*, 2007, pp. 1341–1345.

[10] J. S. Nielsen, "Power Decoding Reed–Solomon Codes Up to the Johnson Radius," *Submitted to Advances in Mathematics of Communications, arXiv preprint:1505.02111*, 2015.

[11] W. A. Stein *et al.*, "SageMath Software," http://www.sagemath.org.

[12] J. S. R. Nielsen, "Power Decoding of Reed–Solomon Codes Revisited," in *Proc. ICMCTA4*, Sep. 2014.

[13] J. Rosenkilde né Nielsen and A. Storjohann, "Algorithms for Simultaneous Hermite Padé Approximations," *In preparation. Extended version of [19].*

[14] C.-P. Jeannerod, V. Neiger, E. Schost, and G. Villard, "Fast Computation of Minimal Interpolation Bases in Popov Form for Arbitrary Shifts," in *ACM ISSAC*, 2016.

[15] V. Guruswami and M. Sudan, "Improved Decoding of Reed-Solomon and Algebraic-Geometric Codes," in *IEEE FOCS*, 1998, pp. 28–37.

[16] H. Cohn and N. Heninger, "Approximate Common Divisors via Lattices," *The Open Book Series*, vol. 1, no. 1, pp. 271–293, 2013.

[17] V. Guruswami and A. Rudra, "Explicit Codes Achieving List Decoding Capacity: Error-Correction With Optimal Redundancy," *IEEE Trans. Inf. Theory*, vol. 54, no. 1, pp. 135–150, 2008.

[18] S. Kopparty, "List-Decoding Multiplicity Codes," in *Electronic Colloquium on Computational Complexity*, vol. 19, 2012, p. 2.

[19] J. Rosenkilde né Nielsen and A. Storjohann, "Algorithms for Simultaneous Padé Approximations," in *ACM ISSAC*, 2016.

Appendix

A. Technical Proofs

Proof of Theorem 10: By Lemma 9, we are ensured a solution different from $\Lambda_{\boldsymbol{i}}, \Psi_{\boldsymbol{j}}$ if $\sum_{|\boldsymbol{i}|<s} N_{\boldsymbol{i}} > \sum_{\boldsymbol{j}\le\ell}(\deg G_{\boldsymbol{j}} - T_{\boldsymbol{j}}) + 1$, where $N_{\boldsymbol{i}}, T_{\boldsymbol{j}}, G_{\boldsymbol{j}}$ are as in Proposition 6. We get

$$\sum_{|\boldsymbol{i}|<s} N_{\boldsymbol{i}} = \sum_{|\boldsymbol{i}|<s} (\tau s - |\boldsymbol{i}| + 1) = (\tau s + 1)\tbinom{m+s-1}{m} - m\tbinom{m+s-1}{m+1},$$

$$\sum_{|\boldsymbol{j}|\le\ell} (\deg G_{\boldsymbol{j}} - T_{\boldsymbol{j}}) =$$

$$\sum_{|\boldsymbol{j}|<s} \left(|\boldsymbol{j}|(n-k)\right) + \sum_{s\le|\boldsymbol{j}|\le\ell} (s(n-\tau) - |\boldsymbol{j}|(k-1) - 1) =$$

$$(n-k)m\tbinom{m+s-1}{m+1} + (s(n-\tau)-1)\left(\tbinom{m+\ell}{m} - \tbinom{m+s-1}{m}\right)$$
$$- (k-1)\left(m\tbinom{m+\ell}{m+1} - m\tbinom{m+s-1}{m+1}\right).$$

The inequality becomes:

$$s\tau\tbinom{m+\ell}{m} > n\left[m\tbinom{m+s-1}{m+1} + s\left(\tbinom{m+\ell}{m} - \tbinom{m+s-1}{m}\right)\right]$$
$$- (k-1)m\tbinom{m+\ell-1}{m+1} - \tbinom{m+\ell}{m} + 1.$$

■

Lemma 14: Let $m \in \mathbb{Z}_{>0}$ and $\gamma \in (0,1)$ be fixed. Then for $i \in \mathbb{Z}_{>0}$ going to infinity, we have

$$\frac{\binom{m+\lfloor\gamma i\rfloor}{m}}{\binom{m+i}{m}} = \gamma^m + O_{\gamma,m}\left(\tfrac{1}{i}\right).$$

Proof: Using Stirling's formula $(*)$ and

$$1 \le \sqrt{\tfrac{(m+\gamma i)i}{(m+i)\gamma i}} = \sqrt{1 + \tfrac{(1-\gamma)m}{\gamma i}} \le 1 + \tfrac{(1-\gamma)m}{\gamma i}, \tag{1}$$

$$\left|e^x - \left(1 + \tfrac{x}{i}\right)^i\right| = O\left(\tfrac{1}{i}\right) \quad \forall x \in \mathbb{R}, \tag{2}$$

$$\left(\tfrac{m+\gamma i}{m+i}\right)^m = \gamma^m + \sum_{j=1}^{m} \tbinom{m}{j}\gamma^{m-j}\left(\tfrac{(1-\gamma)m}{m+i}\right)^j = \gamma^m + O_{\gamma,m}\left(\tfrac{1}{i}\right), \tag{3}$$

we obtain

$$\frac{\binom{m+\lfloor\gamma i\rfloor}{m}}{\binom{m+i}{m}} = \frac{(m+\lfloor\gamma i\rfloor)!i!}{\lfloor\gamma i\rfloor!(m+i)!} \overset{(*)}{=} \overbrace{\sqrt{\tfrac{(m+\gamma i)i}{(m+i)\gamma i}}}^{\overset{(1)}{=}1+O(\frac{1}{i})} \cdot \frac{(m+\gamma i)^{m+\gamma i} i^i}{(\gamma i)^{\gamma i}(m+i)^{m+i}}$$
$$\cdot \underbrace{e^{(m+\gamma i)+i-\gamma i-(m+i)}}_{=1} + O\left(\tfrac{1}{i}\right)$$
$$= \underbrace{\left(\tfrac{m+\gamma i}{m+i}\right)^m}_{\underset{(3)}{=}\gamma^m+O(\frac{1}{i})} \cdot \underbrace{\tfrac{(m+\gamma i)^{\gamma i}}{(\gamma i)^{\gamma i}}}_{\underset{(2)}{=}e^m+O(\frac{1}{i})} \cdot \underbrace{\tfrac{(i)^i}{(m+i)^i}}_{\underset{(2)}{=}e^{-m}+O(\frac{1}{i})} + O\left(\tfrac{1}{i}\right) = \gamma^m + O\left(\tfrac{1}{i}\right),$$

which proves the claim. ■

Proof of Theorem 11: We choose (ℓ_i, s_i) as in the theorem statement. Using Lemma 14, we obtain

$$\tfrac{\tau_{\text{new}}}{n} = 1 - \Big[1 + \underbrace{\left(1 - \tfrac{1}{s_i}\right)}_{=1-O(\frac{1}{i})} \tfrac{m}{m+1}\Big] \underbrace{\frac{\binom{m+s_i-1}{m}}{\binom{m+\ell_i}{m}}}_{\frac{\binom{m+\lfloor\gamma i\rfloor}{m}}{\binom{m+i}{m}} = \gamma^m + O(\frac{1}{i})}$$
$$- \tfrac{m}{m+1} \underbrace{\tfrac{\ell_i}{s_i}}_{\substack{=\gamma^{-1}\\ +O(\frac{1}{i})}} \tfrac{k-1}{n} - \underbrace{\tfrac{1}{s_i}}_{=O(\frac{1}{i})} \underbrace{\left[1 - \tfrac{1}{\binom{m+\ell}{m}}\right]}_{=1-O(\frac{1}{i})}$$
$$= 1 + \tfrac{m}{m+1}\underbrace{\left(\gamma^m - \gamma^{-1}\tfrac{k-1}{n}\right)}_{=0} - \gamma^m - O\left(\tfrac{1}{i}\right)$$
$$= 1 - \left(\tfrac{k-1}{n}\right)^{\frac{m}{m+1}} - O\left(\tfrac{1}{i}\right),$$

which proves the claim. ■