Matrix structure, polynomial arithmetic, and erasure-resilient encoding/decoding*

Victor Y. Pan
Mathematics and Computer Science Department
Lehman College, CUNY, Bronx, NY 10468
vpan@alpha.lehman.cuny.edu

ABSTRACT

We exploit various matrix structures to decrease the running time and memory space of the known practical deterministic schemes for erasure-resilient encoding/decoding. Polynomial interpolation and multipoint evaluation enable both encoding and decoding in nearly linear time but the overhead constants are large (particularly, for interpolation), and more straightforward quadratic time algorithms prevail in practice. We propose faster algorithms. At the encoding stage, we decrease the running time per information packet from $C\log^2 r$, for a large constant C, or from r (for practical encoding) to $\log r$. For decoding, our improvement is by factors C and $N/\log N$, respectively, for the input of size N. Our computations do not involve polynomial interpolation. Multipoint polynomial evaluation is either also avoided or is confined to decoding.

1. INTRODUCTION

Computations with structured matrices are ominipresent in algebraic and numerical computations (see, e.g., [3], [4], [10], [21], [15], [16], [17], [22], [20], [13], and bibliography therein). The complexity of these computations is much lower than for the ones with general matrices because of the close correlation to polynomial computations (see [19], [3]).

In this paper, we apply computations with structured matrices to erasure-resilient codes and improve the known algorithms for encoding/decoding.

Erasure-resilient codes are highly important in practice and have been intensively studied, particularly in the last decade. The existent codes are quite advanced technically and incorporate several approaches, including various algebraic and combinatorial methods as well as some differential equation techniques. Because of practical importance of the problem, even minor progress in its solution is highly desirable, but the area is more limited technically and much less amenable to further progress than, say, error-correcting codes, so that one should be tempted to conclude that the intensive efforts spent here by the researchers have exhausted all the likely directions to further substantial improvements.

Our goal is to prove that this is not the case. That is, we found a novel way to improving the known scheme for erasure-resilient encoding/decoding. Our progress relies on manipulation with structured matrices. In particular, we exploit transformations among various classes of structured matrices in order to improve the known algorithms. The idea can be traced back to [18], but we propose its novel realizations.

Let us compare our approach to some previous works. The known techniques reduce the encoding to Trummer's celebrated problem of multiplication of an $(n-m)\times m$ Cauchy matrix C by a vector provided that m information packets and $n-m$ redundant packets are sent through the communication channel and a total of $n-m$ packets are erased. The decoding is reduced to solving a nonsingular linear system of k equations whose coefficient matrix is a submatrix of C and, therefore, is also a Cauchy matrix (cf. [14], [1], [26], [2] on this approach).

Trummer's problem can be also interpreted as the problem of multipoint evaluation of rational functions represented via their poles. Its solution is the basis for the solution of such important problems as many-body simulation in mechanics, computing the velocity field, computation of conformal mapping, and the evaluation of the Riemann zeta function. (Extensive bibliography on Trummer's problem can be traced from [21].) Simple straightforward solution of this problem as well as the solution of a Cauchy linear system has arithmetic cost $O(N^2)$, for the input size N.

Applications of the fast algorithms of [8] for Trummer's problem and [7] for solving a Cauchy linear system yield encoding/decoding at the computational cost $O(E(l)n/l + H(k) + I(k) + E(k))$. Here, $l = \min\{m, n-m\}$, k may vary between 0 and l, but in some typical practical computations k/n ranges between 0.4 and 0.5, and here and hereafter, we write $E(s) = O(M(s)\log s)$, $I(s) = O(M(s)\log s)$, $H(s) = O(M(s)\log s)$, and $M(s) = O((s\log s)\log\log s)$ to denote the computational cost of the evaluation of a polynomial of degree s at $s+1$ points, interpolation to such a polynomial, the recovery of its coefficients from its roots,

*Supported by NSF Grant CCR 9732206 and PSC-CUNY Award 669363

and multiplication of two polynomials of degree s, respectively (cf. [3]). (We measure the computational cost in terms of the number of field operations involved and will refer to them as *ops*.)

The overhead constants hidden in the above "O" notation are quite large, however, particularly for the interpolation, and in practice the computation goes by more straightforward algorithms that involve $O((n-m)m)$ and $O(km)$ ops at the stages of encoding and decoding, respectively [5]. More efficient codes can be devised by using randomization [11], but several major applications require deterministic codes.

We propose alternative deterministic solution algorithms, which better exploit the Cauchy structure involved. This enables us to avoid interpolation completely and either to avoid multipoint polynomial evaluation too (in the case where the ratios k/n and $(m-k)/n$ are not small) or to limit it to the decoding stage (otherwise). The encoding stage can be performed in $O(M(l)n/l)$ ops or, alternatively, by using $O(r)+2DFT(r)$ ops provided that $r \geq \max\{m, n-m\}$ and the ground field allows us to perform the discrete Fourier transform at the r-th roots of 1 by using DFT(r) ops, which is $O(r \log r)$ if FFT can be applied. This improves the known encoding approaches by factor $\log r$. At the stage of decoding it is sufficient to use $O((E(k)+M(k))m/k)$ ops or, unless k/n is small, only $O((H(k)+M(k))m/k)$ ops. In all our estimates, at both encoding and decoding stages, we achieve substantial decrease of the overhead constants hidden in the "O" notation, versus the approach based on [7] and [8].

To summarize, we improve by factor $C \log r$, for a large constant C, the known asymptotically fast encoding methods (which are nonpractical because C is large) and by factor of order $r/\log r$ the practical encoding methods. (This reduces the record asymptotic time bound for encoding from order $\log^2 r$ to $O(\log r)$ per packet.) For decoding, our improvement is by factors C and $k/\log k$, respectively.

All our computations are performed in linear space, of n words of memory at the encoding stage and from $O(k)$ to n at the decoding stage.

Technically, we rely on the well known algebraic encoding/decoding algorithm of [14], [1], [26], [2], [5] (see Algorithm 3.1 and Remark 3.1 in our section 3), whose implementation in [5] is known as the best practical deterministic erasure-resilient code, but we propose novel versions of its basic blocks of structured matrix computations. There, we yield improvements that demonstrate the potential power of the application of structured matrices to encoding/decoding and promise to be practically useful. The algorithm allows its further practical improvements by using the XOR's computer words and tabulation (cf. [5]). Our computations are reduced to polynomial multiplication or FFT and, therefore, allow processor efficient parallel acceleration to yield the time bounds $O(\log m)$ and $O(\log^2 k)$ at the encoding and decoding stages, respectively.

Due to the well known importance of Trummer's problem, its slightly improved solution proposed in section 5 may be of some independent interest.

More specifically, our progress is achieved based on our novel combination of the Cauchy matrix structure, which ensures nonsingularity, with the Fourier type or Hankel structure, which support additional theoretical and practical speed up of the computations. Our exploitation of structured matrices for improving erasure-resilient codes is of distinct nature than the known applications of structured matrices to error-correcting codes (cf. [17], [20]). Indeed, these two coding areas are quite distinct, as well as the techniques involved there, in spite of some apparent similarities. For instance, the advantages of working in $GF(2^p)$ are typical for the error-correcting codes but turned out to be less important and overweighted by other considerations (such as nonsingularity of the Cauchy generator matrices) for the deterministic erasure-resilient codes. Furthermore, as we mentioned, the relative simplicity of the area of erasure-resilient codes leaves much less room for innovations and new progress.

We organize our paper as follows. After some definitions and preliminaries in the next section, we describe the basic encoding/decoding scheme in section 3, where we also show some choices for Cauchy matrix defining the code generator. We specify our computations with structured matrices for both encoding and decoding in section 4-6. In the same sections we estimate the computational cost of each stage.

Acknowledgement. The problem of improving the erasure-resilient encoding/decoding and reference [5] were brought to my attention by Marek Karpinski.

2. DEFINITIONS AND SOME PRELIMINARY RESULTS

Fix two integers m and n, $n > m > 0$, a field $\mathbb{F}$ (say, $GF[2]$ or $GF[2^L]$) and its n distinct elements $s_i, i = 0, \dots, n-m-1$, and $-t_j, j = 0, \dots, m-1$. Let I_s stand for the $s \times s$ identity matrix. Let $\mathbf{s} = (s_i)_{i=0}^{n-m-1} \in \mathbb{F}^{n-m}$, $\mathbf{t} = (t_j)_{j=0}^{m-1} \in \mathbb{F}^m$, $\mathbf{u} = (u_i)_{i=0}^{p-1} \in \mathbb{F}^p$, and $\mathbf{v} = (v_i)_{i=0}^{q-1} \in \mathbb{F}^q$ be some vectors of dimensions $n-m, m, p$ and q, respectively. $W = (w_{i,j})$ is a *Hankel matrix* if $w_{i,j} = w_{i-1,j+1}$ for all pairs (i,j). For a vector $\mathbf{r} = (r_i)_{i=0}^{k-1}$, let $H(\mathbf{r}) = (h_{i,j})$ be the $l \times l$ triangular Hankel matrix defined by its first column vector $\mathbf{r}$, $h_{i,j} = r_{i+j}$ for $i+j < l$, $h_{i,j} = 0$ for $i+j \geq l$, let $V(\mathbf{r}) = (r_i^j)_{i,j=0}^{k-1,l-1}$ be a $k \times l$ Vandermonde matrix, and write $V_k(\mathbf{r}) = V(\mathbf{r})$. For a vector or a matrix W, let W^T be its transpose. Let $C = C(\mathbf{u}, \mathbf{v}) = (\frac{1}{u_i - v_j})_{i=0,j=0}^{p-1,q-1}$ be a $p \times q$ *Cauchy matrix*, defined by two vectors $\mathbf{u}$ and $\mathbf{v}$, and let $G = G(\mathbf{s}, \mathbf{t}) = \begin{pmatrix} I_m \\ C(\mathbf{s}, \mathbf{t}) \end{pmatrix} \in \mathbb{F}^{n \times m}$ be the $n \times m$ *generator matrix*. $\mathrm{diag}(\mathbf{u}) = \mathrm{diag}(u_i)_{i=0}^{p-1}$ will be our notation for the $p \times p$ diagonal matrix with diagonal entries $u_0, \dots, u_{p-1}$; $h_{\mathbf{u}}(x)$ for the monic polynomial $\prod_{i=0}^{p-1}(x-u_i)$ of degree p in x, $h'_{\mathbf{u}}(x)$ for its derivative in x, and $|S|$ for the cardinality of a set S.

For a vector $\mathbf{v} = (v_j)_{j=0}^{q-1}$, a matrix $W = (w_{j,k})_{j=0,k=0}^{q-1,p-1}$, and two subsets, J of the set $\{0, \dots, q-1\}$ and K of the set $\{0, \dots, p-1\}$, let $\mathbf{v}_J$ be the subvector $(v_j)_{j \in J}$ of $\mathbf{v}$ and let $W_{J,K}$ be the submatrix $(w_{j,k})_{j \in J, k \in K}$ of W.

Hereafter, "ops" stands for "operations in the field $\mathbf{F}$", $M(k)$ ops suffice to multiply modulo x^k a pair of polynomials in

x,

$$M(k) \le 2k^2 - 2k + 1, \quad M(k) = O((k \log k) \log\log k), \quad (1)$$

$H(k)$ ops suffice to compute the coefficients of the polynomial $h_{\mathbf{w}}(x)$ for a given vector $\mathbf{w} \in \mathbb{F}^k$,

$$H(k) \le (k-1)^2, \quad H(k) = O(M(k) \log k), \quad (2)$$

$E(k,l)$ ops suffice to evaluate a polynomial of degree at most $k-1$ on a set of l points, $\{r_i\}, i = 0, \ldots, l-1$, which is equivalent to multiplication of the Vandermonde matrix $V(\mathbf{r})$ by the coefficient vector of this polynomial,

$$E(k,l) \le (2k-1)l, \quad E(k,l) = O((M(k) \log k)\lceil l/k \rceil), \quad (3)$$

and $F(k,l)$ ops suffice to perform such an evaluation where the points are of the form $b^i, i = 0, 1, \ldots, l-1; b \in \mathbb{F}$,

$$F(k,l) = O((k+l) \log(\min\{k,l\})) \quad (4)$$

(cf. [3]). Hereafter, we write $E(k,k) = E(k), F(k,k) = F(k)$. $F(k,l)$ and $F(k)$ represent the computational cost of *generalized discrete Fourier transform (generalized DFT)* [3, p.14]. For b being a primitive k-th root of 1, we arrive at the classical DFT, and then

$$F(k) \le 1.5k \log_2 k \quad (5)$$

if k is a power of 2. If the field $\mathbb{F}$ contains a primitive $2k$-th root of 1 for an integer s, then

$$M(k) \le 9k \log_2(2k) \quad (6)$$

(cf. [3]).

DEFINITION 2.1. *The minimum number of ops, required to multiply a fixed rectangular or square matrix W by a vector will be denoted by c_W.*

LEMMA 2.1. *$c_V \le c_W$ for any submatrix V of a matrix W.*

LEMMA 2.2. *Let V be a $k \times l$ block matrix with $g \times h$ blocks $V(i,j)$, $i = 0, \ldots, k-1$; $j = 0, \ldots, l-1$. Then $c_V \le \sum_{i,j} c_{V(i,j)} + (l-1)gk$.*

We will also use the following results, the first and the third of which are obvious:

LEMMA 2.3. *$c_{V_l(\mathbf{r})} \le E(k,l)$, if $\mathbf{r} \in \mathbb{F}^k$.*

LEMMA 2.4. *(Cf. [23].) $c_{V^T(\mathbf{r})} = c_{V(\mathbf{r})}$.*

LEMMA 2.5. *(Cf. [3].) $c_W \le M(k+2l-2)$ for a $k \times l$ Hankel matrix W; $c_W \le M(2k-1)$ for a $k \times k$ triangular Hankel matrix $W = H(\mathbf{r})$.*

3. ERASURE-RESILIENT ENCODING/DECODING SCHEME

The next scheme for erasure-resilient encoding/decoding involves a Cauchy matrix and relies on a Reed-Solomon systematic linear code with an m-packet message vector $\mathbf{v}$ and a generator matrix $G = G(\mathbf{s}, \mathbf{t})$ [14], [1], [26], [2]. Stage 1 represents encoding, stages 2-4 decoding. Practically, the scheme is applied to several vectors $\mathbf{v}$, which are processed successively or concurrently.

ALGORITHM 3.1. *(Erasure-resilient encoding/decoding).*

INPUT: a field $\mathbb{F}$, two integers m and n, $n \ge m$, an m-packet message $\mathbf{v}$, a pair of vectors $\mathbf{s} \in \mathbb{F}^{n-m}$ and $\mathbf{t} \in \mathbb{F}^m$ with n distinct components, the generator matrix $G = G(\mathbf{s}, \mathbf{t}) = \begin{pmatrix} I_m \\ C(\mathbf{s}, \mathbf{t}) \end{pmatrix} \in \mathbb{F}^{m \times m}$, a set $\{\rho_l = \rho_l(\mathbf{s}, \mathbf{t})\}_l$ of elements of $\mathbb{F}$ obtained by performing some arithmetic operations with the components of the vectors $\mathbf{s}$ and $\mathbf{t}$, and a subset Q of cardinality $|Q| = n - m$ in the set $\{0, \ldots, n-1\}$, (Q indexes the $n-m$ erased packets);

1. (*encoding an m-packet message* $\mathbf{v}$). Compute the vector $\mathbf{u} = G\mathbf{v}$.

2. Define the intersections, K and P, of the set Q with the sets $\{0, \ldots, m-1\}$ and $\{m, \ldots, n-1\}$, respectively; the complements of these intersections, $M = \{0, \ldots, m-1\} - K$ and $J = \{m, \ldots, n-1\} - P$; the subvectors $\mathbf{u}_K$ and $\mathbf{u}_M$ of $\mathbf{u}$, and the Cauchy matrices $C_{J,K} = C(\mathbf{p}, \mathbf{q}) \in \mathbb{F}^{k \times k}$ and $C_{J,M} = C(\mathbf{p}, \mathbf{r}) \in \mathbb{F}^{k \times (m-k)}$ (these are two submatrices of the matrix $C_{J,K \cup M} = C(\mathbf{p}, \mathbf{t})$, for $k = |J| = |K|$.

3. Compute the matrix $\tilde{G} = C_{J,K}^{-1}$.

4. Compute the vectors $\tilde{\mathbf{u}}_K = \mathbf{u}_K - C(\mathbf{p}, \mathbf{r})\mathbf{u}_M$ and $\mathbf{v}_K = \tilde{G}\tilde{\mathbf{u}}_K$.

OUTPUT: the vector $\mathbf{v}$ composed of its subvectors $\mathbf{v}_M = \mathbf{u}_M$ and $\mathbf{v}_K$.

REMARK 3.1. *The algorithm is the basis for the current best practical deterministic erasure-resilient encoding and decoding [5]. We refer the reader to the latter paper for many implementation details.*

The use of a Cauchy matrix $C(\mathbf{s}, \mathbf{t})$ is due to the next theorem, which also implies the correctness of Algorithms 3.1 [14].

THEOREM 3.1. *(Cf. [12].) Every square submatrix of the matrix $C = (\frac{1}{s_i - t_j})_{i,j}$ is nonsingular if all n values s_i, t_j are distinct.*

We allow cost-free precomputations with vectors $\mathbf{s}$ and $\mathbf{t}$ because these vectors are given once and for all messages sent through. (Note that the vectors $\mathbf{p}, \mathbf{q}$ and $\mathbf{r}$ are treated

differently - they are updated with each new message.) The choice of vectors $\mathbf{s}$ and $\mathbf{t}$ is in fact ours, and we will use such a choice to simplify the computations. Specifically, we will rely on two choices:

$$s_i = as^i, \quad t_j = bt^j, \quad a, b, s, t \in \mathbb{F}, \tag{7}$$

$$i = 0, 1 \ldots, n-m-1; j = 0, 1, \ldots, m-1,$$

$$s_{i+1} - s_i = t_j - t_{j+1} \text{ for all pairs } (i, j). \tag{8}$$

Equations (7) impose generalized Fourier structure on the matrices $V_l(\mathbf{s})$ and $V_l(\mathbf{t})$, which we will associate with the matrix $C(\mathbf{s}, \mathbf{t})$, and consequently, on their submatrices $V_l(\mathbf{p})$, $V_l(\mathbf{q})$, and $V_l(\mathbf{r})$ (cf. Lemma 2.1):

FACT 3.1. *Equations (7) imply that*

$$c_{V_l(\mathbf{p})} \le c_{V_l(\mathbf{s})} \le F(n-m, l) + n - m,$$
$$\max\{c_{V_l(\mathbf{r})}, c_{V_l(\mathbf{q})}\} \le c_{V_l(\mathbf{t})} \le F(m, l).$$

In the special case where $\mathbb{F}$ contains ω_n, a primitive n-th root of 1, we may satisfy (7) by choosing $a = 1, b = \omega_n^{n-m}, s = t = \omega_n$. In this case generalized DFT's supporting Fact 3.1 turn into classical DFT's.

Equations (8) impose Hankel structure on the matrix $C(\mathbf{s}, \mathbf{t})$:

FACT 3.2. *The matrix $C(\mathbf{s}, \mathbf{t}) = (\frac{1}{s_i - t_j})_{i,j}$ is a Hankel matrix if and only if equations (8) hold.*

In particular, we have $C(\mathbf{s}, \mathbf{t}) = (\frac{1}{i+j+a})_{i,j}$ if $s_i = i + a$, $t_j = -j$, for all pairs (i, j); this choice satisfies (8).

Our next objective is to specify the algorithms for the multiplication of the matrices $C(\mathbf{s}, \mathbf{t})$, $C(\mathbf{p}, \mathbf{r})$ and $C^{-1}(\mathbf{p}, \mathbf{q})$ by vectors and to estimate the computational complexity $c_{C(\mathbf{s},\mathbf{t})}$, $c_{C(\mathbf{p},\mathbf{r})}$ and $c_{C^{-1}(\mathbf{p},\mathbf{q})}$ of such operations. We will do this in the next three sections. This will complete our description and analysis of Algorithm 3.1, which consists of such three multiplications apart from k subtractions in the field $\mathbb{F}$ performed at its stage 4.

4. MULTIPLICATION OF THE INPUT MATRIX $C(\mathbf{S}, \mathbf{T})$ BY A VECTOR

The straightforward algorithm implies that

$$c_{C(\mathbf{s},\mathbf{t})} \le (m-n)(2m-1). \tag{9}$$

Based on (8), Fact 3.2 and Lemma 2.5 we obtain faster computation, by using

$$c_{C(\mathbf{s},\mathbf{t})} \le M(n+m-1). \tag{10}$$

ops. Let us next assume (7) and deduce that

$$c_{C(\mathbf{s},\mathbf{t})} \le F(m) + M(m) + F(n-m) + n. \tag{11}$$

Our algorithm supporting (11) relies on the following formulae [6], [3, p.174], [9]]:

$$C(\mathbf{s}, \mathbf{t}) = \left(\operatorname{diag}(h_{\mathbf{t}}(s_i))_{i=0}^{m-1}\right)^{-1} V(\mathbf{s}) V^{-1}(\mathbf{t}) \operatorname{diag}(h'_{\mathbf{t}}(t_j))_{j=0}^{n-m-1},$$

$$V^{-1}(\mathbf{t}) =$$

$$R_m T_f(\mathbf{h_t} + f\mathbf{e}^{(0)}) V^T(\mathbf{t}) \operatorname{diag}(h'_{\mathbf{t}}(t_i)(f - t_i^{n-m}))_{i=0}^{n-m-1})^{-1}, \tag{13}$$

where $f \neq 0$ is any element of $\mathbb{F}$, $\mathbf{e}^{(0)} = (1, 0, \ldots, 0)^T$ is the first coordinate vector, $\mathbf{h_t}$ is the coefficient vector of the polynomial $h_{\mathbf{t}}(x) - x^m$ of degree $m-1$, R_m is the $m \times m$ reversion matrix, $R_m = (r_{i,j})_{i,j=0}^{m-1}$, $r_{i,j} = 1$ if $i = m - 1 - j$, $r_{i,j} = 0$, otherwise, and $T_f(w)$ denotes the square *f-circulant matrix* with the first column $\mathbf{w} = (w_i)_i$, $T_f(w) = \sum_i w_i T_f^i$, $T_f = (t_{i,j})$, $t_{i,j} = 1$ if $i = j + 1$, $t_{i,j} = f$ if $i = 0, j = m - 1$; $t_{i,j} = 0$ otherwise.

FACT 4.1. *(Cf. [3].) $M(m)$ ops are sufficient to multiply an f-circulant $m \times m$ matrix by a vector.*

To arrive at (11), we fix $f \in \mathbb{F}$ and precompute the vector $\mathbf{h_t} + f\mathbf{e}^{(0)}$ and the diagonal matrices $\operatorname{diag}(h_{\mathbf{t}}(s_i))_{i=0}^{m-1}$ and $\operatorname{diag}(f - t_i^{n-m})_{i=0}^{n-m-1}$, whose entries only depend on the vectors $\mathbf{s}$ and $\mathbf{t}$. (Such a precomputation is cost-free by our assumption.) Then, by (12), (13), it remains to multiply by vectors the following matrices: $\operatorname{diag}(f - t_i^{n-m})_{i=0}^{n-m-1}$ (by using $n-m$ ops), $V^T(\mathbf{t})$ (by using $c_{V(\mathbf{t})}$ ops, by Lemma 2.4), $c_{V(\mathbf{t})} \le F(m)$ (by Fact 3.1), $T_f(\mathbf{h_t} + f\mathbf{e}^{(0)})$ ($M(m)$ ops, by Fact 4.1), R_m (no ops involved), $V(\mathbf{s})$ ($c_{V(\mathbf{s})} \le F(n-m)$ (by Fact 3.1), and $\operatorname{diag}(h_{\mathbf{t}}(s_i))_{i=0}^{m-1}$ (m ops). Summarizing, we obtain (11). The choice among the three bounds (9), (10) or (11) depends on the input values m and n and is fixed together with the generator matrix.

5. MULTIPLICATION OF A SUBMATRIX $C(\mathbf{P}, \mathbf{R})$ BY A VECTOR

The straightforward algorithm implies that

$$c_{C(\mathbf{p},\mathbf{r})} \le (2m - 2k - 1)k. \tag{14}$$

On the other hand, $C(\mathbf{p}, \mathbf{r})$ is a submatrix of the matrix $C(\mathbf{s}, \mathbf{t})$, so by Lemma 2.1,

$$c_{C(\mathbf{p},\mathbf{r})} \le c_{C(\mathbf{s},\mathbf{t})}, \tag{15}$$

and the multiplication of the matrix $C(\mathbf{p}, \mathbf{r})$ by a vector can be performed by the algorithms of the previous sections. Let us recall two other algorithms. The first of them relies on

FACT 5.1. *(Cf. [25], [6].)*

$$C(\mathbf{p}, \mathbf{r}) = \left(\operatorname{diag}(h_{\mathbf{r}}(p_i))_{i=0}^{k-1}\right)^{-1} V(\mathbf{p}) H(\mathbf{r}) V^T(\mathbf{r}).$$

Now, to multiply the matrix $C(\mathbf{p}, \mathbf{r})$ by a vector, we may first compute the coefficients of the polynomial $h_{\mathbf{r}}(x)$ (by using $H(m-k)$ ops), then its values $h_{\mathbf{r}}(p_i)$ for $i = 0, 1, \ldots, k-1$ ($c_{V_{m-k}(\mathbf{p})}$ ops), and then successively multiply by vectors the matrices $V^T(\mathbf{r})$, $H(\mathbf{r})$, $V(\mathbf{p})$, and $(\operatorname{diag}(h_{\mathbf{r}}(p_i))_{i=0}^{k-1})^{-1}$ (k divisions). By Lemmas 2.4 and 2.5 and Fact 3.1, the overall cost of these computations is bounded by

$$\begin{aligned} c_{C(\mathbf{p},\mathbf{r})} \le & k + H(m-k) + M(2m-2k-1) \\ & + c_{V_{m-k}(\mathbf{p})} + c_{V(\mathbf{p})} + c_{V(\mathbf{r})}, \end{aligned} \tag{16}$$

where

$$c_{V_{m-k}(\mathbf{p})} \le \min\{E(m-k+1,k), F(n-m,m-k)\}, \quad (17)$$

$$c_{V(\mathbf{p})} \le \min\{E(k), F(n-m,k)\}, \quad (18)$$

$$c_{V(\mathbf{r})} \le \min\{E(m-k), F(m,m-k)\}, \quad (19)$$

and all values $E(l,q)$ satisfy (3).

Finally, an alternative algorithm [16] computes the product

$$C(\mathbf{p},\mathbf{r})\mathbf{v} = \Big(\sum_{j=0}^{m-k-1} v_j/(p_i - r_j)\Big)_{i=0}^{k-1}$$

by first computing the coefficients of two polynomials $n(x)$ and $d(x)$ such that

$$n(x)/d(x) = \sum_{j=0}^{m-k-1} v_j/(x-r_j), \quad \deg n(x) \le \deg d(x) = m-k,$$

and then computing the values $u(x)/d(x)$ at the points $x = p_i, i = 0,1,\dots,k-1$. The computation of the coefficients is by recursive summation of partial fractions, which starts with the pairs $v_{2j-1}/(x-r_{2j-1})$ and $v_{2j}/(x-r_{2j})$. This stage involves

$$\tilde{c}_{m-k} \le 3\sum_{i=1}^{s} 2^{s-i} M(2^i) = O(M(m-k)\log(m-k)) \quad (20)$$

ops for $s = \lceil \log_2(m-k) \rceil$. The subsequent computation of the k values $u(p_i)/d(p_i)$ involves $k + 2c_{V_{m-k}(\mathbf{p})}$ ops, and we obtain that

$$c_{C(\mathbf{p},\mathbf{r})} \le \tilde{c}_{m-k} + k + 2c_{V_{m-k}(\mathbf{p})} \quad (21)$$

for $\tilde{c}_{m-k}$ and $c_{V_{m-k}(\mathbf{p})}$ bounded according to (20) and (17).

Unlike the better known algorithm of [8], our algorithms for Trummer's problem of multiplication of a Cauchy matrix $C(\mathbf{p},\mathbf{r})$ by a vector avoid interpolation.

The choice among the algorithms of this section depends on the values n, m and k, that is, may vary with the number of erased packets.

6. SOLUTION OF A CAUCHY LINEAR SYSTEM OF EQUATIONS

We will consider two approaches. One of them relies on an extension of (12) and (13):

$$C^{-1}(\mathbf{p},\mathbf{q}) = \Big(\mathrm{diag}(h'_{\mathbf{q}}(q_i))_{j=0}^{n-m-1}\Big)^{-1} V(\mathbf{q})V^{-1}(\mathbf{p})\,\mathrm{diag}(h_{\mathbf{q}}(p_i))_{i=0}^{k-1}, \quad (22)$$

$$V^{-1}(\mathbf{p}) = R_k T_f(\mathbf{h_p} + f\mathbf{e}^{(0)})V^T(\mathbf{p})\,\mathrm{diag}(h'_{\mathbf{p}}(p_i)(f-p_i^k))_{i=0}^{k-1}). \quad (23)$$

The algorithm based on (22), (23) is similar to the one supporting (11), except that now we have to include the cost of computing the entries of the diagonal matrices because they depend on the set Q (besides the vectors $\mathbf{s}$ and $\mathbf{t}$) and, therefore, generally change with each new message sent. The algorithm supports the following cost bound:

$$c_{C^{-1}(\mathbf{p},\mathbf{q})} \le 2H(k) + 5k + 1 + M(k) + kP(k) + 2c_{V(\mathbf{p})} + 3c_{V(\mathbf{q})} \quad (24)$$

provided that $c_{V(\mathbf{p})}$ is bounded according to (18),

$$c_{V(\mathbf{q})} \le \min\{E(k), F(m,k)\}, \quad (25)$$

and each power $p_i^{(k)}$ can be computed in $P(k)$ ops,

$$P(k) < 2\lceil \log_2 k \rceil. \quad (26)$$

An alternative approach relies on

THEOREM 6.1. *(Cf. [24], [6]). For vectors $\mathbf{p},\mathbf{q}$ filled with $2k$ distinct coordinates, we have*

$$C^{-1}(\mathbf{p},\mathbf{q}) = D_0 C(\mathbf{q},\mathbf{p}) D_1,$$

where

$$D_0 = \mathrm{diag}\left(\frac{h_{\mathbf{p}}(q_i)}{h'_{\mathbf{q}}(q_i)}\right)_{i=0}^{k-1}, \quad D_1 = \mathrm{diag}\left(\frac{h_{\mathbf{q}}(p_j)}{h'_{\mathbf{p}}(p_j)}\right)_{j=0}^{k-1}.$$

In this approach, we evaluate the diagonal matrices D_0 and D_1 and multiply them and the matrix $C(\mathbf{p},\mathbf{q})$ by vectors. This leads us to the cost bound

$$c_{C^{-1}(\mathbf{p},\mathbf{q})} \le 2H(k) + 6k + 2c_{V(\mathbf{p})} + 2c_{V(\mathbf{q})} + c_{C(\mathbf{q},\mathbf{p})}. \quad (27)$$

The terms $c_{V(\mathbf{p})}$ and $c_{V(\mathbf{q})}$ are bounded according to (18) and (25), and we may extend the algorithm of section 5 to estimate $c_{C(\mathbf{q},\mathbf{p})}$. The choice between the two algorithms supporting (24)-(26) and (27) depends on the values n, m and k, that is, changes for each new message sent. The bound of (24) is smaller for large k, but the bound of (27) is smaller for smaller k. In particular, by using the straightforward evaluation of the coefficients of the polynomials $h_{\mathbf{p}}(x)$ and $h_{\mathbf{q}}(x)$, we extend (27) to yield

$$c_{C^{-1}(\mathbf{p},\mathbf{q})} \le 2(k-1)^2 + 6k + 5(2k-1)k = 12k^2 - 3k + 2.$$

7. REFERENCES

[1] A. Albanese, J. Blömer, J. Edmonds, M. Luby, M. Sudan, Priority Encoding Transmission, *Proc. 35th Ann. Symp. on Foundations of Computer Science (FOCS)*, 604-613, IEEE Computer Society Press, 1994.

[2] N. Alon, J. Edmonds, M. Luby, Linear Time Erasure Codes with Nearly Optimal Recovery, *Proc. 36th Ann. Symp. on Foundations of Computer Science (FOCS)*, 512-519, IEEE Computer Society Press, 1995.

[3] D. Bini, V.Y. Pan, *Polynomial and Matrix Computations, Volume 1: Fundamental Algorithms*, Birkhäuser, Boston, 1994.

[4] D. Bini, V.Y. Pan, *Polynomial and Matrix Computations, Volume 2: Fundamental and Practical Algorithms*, Birkhäuser, Boston, 2000.

[5] J. Blömer, M. Kalfane, R. Karp, M. Karpinski, M. Luby, D. Zuckerman, An XOR-Based Erasure-Resilient Coding Scheme, Technical Report TR-95-48, *International Computer Science Institute*, Berkeley, California, 1995.

[6] T. Fink, G. Heinig, K. Rost, An Inversion Formula and Fast Algorithms for Cauchy-Vandermonde Matrices, *Linear Algebra Appl.*, **183**, 179-191, 1993.

[7] N. Gastinel, Inversion d'une Matrice Generalisant la Matrice de Hilbert, *Chiffres*, **3**, 149-152, 1960.

[8] A. Gerasoulis, A Fast Algorithm for the Multiplication of Generalized Hilbert Matrices with Vectors, *Math. Comp.*, **50, 181**, 179-188, 1987.

[9] I. Gohberg, V. Olshevsky, Complexity of Multiplication with Vectors for Structured Matrices, *Linear Algebra Appl.*, **202**, 163-192, 1994.

[10] T. Kailath, A. Sayed, *Fast Reliable Algorithms for Matrices with Structure*, SIAM Publications, Philadelphia, 1999.

[11] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, V. Stemann, Practical Loss-Resilient Codes, *Proc. 29th Ann. Symp. on Theory of Computing (STOC'97)*, ACM Press, New York, 1997.

[12] L. Mirsky, *An Introduction to Linear Algebra*, Dover, New York, 1982.

[13] B. Mourrain, V. Y. Pan, Multivariate Polynomials, Duality and Structured Matrices, *J. of Complexity*, **16, 1**, 2000.

[14] F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, New York, 1977.

[15] V. Olshevsky, V. Y. Pan, A Unified Superfast Algorithm for Boundary Rational Tangential Interpolation Problem and for Inversion and Factorization of Dense Structured Matrices, *Proc. 39th Annual IEEE Symposium on Foundations of Computer Science,* 192-201, IEEE Computer Society Press, 1998.

[16] V. Olshevsky, V. Y. Pan, Polynomial and Rational Interpolation and Multipoint Evaluation (with Structured Matrices), *Proc. 26th Intern. Colloquium on Automata, Languages and Programming* (ICALP'99), **1644**, 585-594, Springer's LNCS, Berlin, 1999.

[17] V. Olshevsky, M. A. Shokrollahi, A Displacement Approach to Efficient Decoding of Algebraic-Geometric Codes, *Proc. 31st Ann. Symp. on Theory of Computing*, 235-244, ACM Press, New York, May 1999.

[18] V. Y. Pan, On Computations with Dense Structured Matrices, *Proc. ACM-SIGSAM Intern. Symp. on Symbolic and Alg. Comp.*, 34-42, ACM Press, New York, 1989, and *Math. of Computation.*, **55, 191**, 179-190, 1990.

[19] V. Y. Pan, Complexity of Computations with Matrices and Polynomials, *SIAM Review*, **34, 2**, 225-262, 1992.

[20] V. Y. Pan, Nearly Optimal Computations with Structured Matrices, *Proc. 11th Ann. ACM-SIAM Symp. on Discrete Algorithms, SODA'2000)*, 953-962, ACM Press, New York, and SIAM Publications, Philadelphia, 2000.

[21] V. Y. Pan, M. AbuTabanjeh, Z. Chen, E. Landowne, A. Sadikou, New Transformations of Cauchy Matrices and Trummer's Problem, *Computer and Math. (with Applics.)*, **35, 12**, 1-5, 1998.

[22] V. Y. Pan, Z. Chen, The Complexity of the Matrix Eigenproblem, *Proc. 31st Annual ACM Symp. on Theory of Computing*, 507-516, ACM Press, New York, 1999.

[23] P. Penfield Jr., R. Spencer, S. Duinker, *Tellegen's Theorem and Electrical Networks,* MIT Press, Cambridge, Massachusetts, 1970.

[24] M. O. Rabin, Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance, *J. ACM*, **36, 2**, 335-348, 1989.

[25] J. F. Traub, Associated Polynomials and Uniform Methods for the Solution of Linear Problems, *SIAM Review*, **8**, 277-301, 1966.

[26] S. B. Wicker, V. Bhargava, *Reed-Solomon Codes and Their Applications*, IEEE Press, New York, 1994.