A MATRIX FORMULATION OF LOWER TRIANGULAR TOEPLITZ SYSTEMS SOLVERS

C. DI FIORE AND P. ZELLINI

Abstract. We describe a simple matrix formulation of methods for solving generic lower triangular Toeplitz systems of $n = b^s$ linear equations, where b is any positive integer. These methods can be applied in the computation of Bernoulli numbers.

1. Introduction

In [8] it is claimed that if $\mathbf{b}$ denotes the vector of all Bernoulli numbers, then for a suitable diagonal matrix D and a suitable vector $\mathbf{f}$, the vector $D\mathbf{b}$ satisfies the following lower triangular Toeplitz (ltT) linear system:

$$[\sum_{k=0}^{+\infty} \frac{2\alpha^{3k}}{(6k+2)!(2k+1)} Z^{3k}]\mathbf{x} = \mathbf{f}, \quad Z = \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ & 1 & 0 & \\ & & \cdot & \cdot \end{bmatrix} \tag{1}$$

($\alpha \in \mathbb{R}$). This system is called Ramanujan-Toeplitz because it is obtained from some linear equations involving Bernoulli numbers found in [20]. Note the sparse structure of such ltT system: the second, third, fifth, sixth, ninth, tenth, and so on, diagonals of the coefficient matrix are all null.

Now, it is possible to define a procedure $\mathcal{G}_3$ that exploits this special sparsity related to Bernoulli numbers, and computes the first $n = 3^s$ ($s \in \mathbb{N}$) components of the solution of (1) in $O(s3^s)$ arithmetic operations. $\mathcal{G}_3$ is obtained as a particular case of a more general procedure $\mathcal{G}_b$ able to compute the first $n = b^s$ entries of the solution of any (semi-infinite) ltT system in $O(sb^s)$ arithmetic operations. The procedure $\mathcal{G}_b$ is described in this paper in terms of simple matrix operations, which consist precisely in a finite (independent of s) number of ltT matrix by vector products of dimensions b, b^2, …, b^s .

Let us recall the main properties of ltT matrices. Let Z_n be the upper left $n \times n$ submatrix of the semi-infinite matrix Z in (1). Z_n is usually called *lower-shift* due to the effect that its multiplication by a vector $\mathbf{v} = [v_0\, v_1\, \cdots\, v_{n-1}]$ produces: $Z_n\mathbf{v} = [0\, v_0\, v_1\, \cdots\, v_{n-2}]^T$. Let $\mathcal{L}$

1991 *Mathematics Subject Classification.* 11B68, 11Y55, 15A06, 15A09, 15A24, 15B05, 15B99, 65F05.

Key words and phrases. Triangular Toeplitz matrices; Bernoulli numbers.

be the subspace of $\mathbb{C}^{n\times n}$ of those matrices which commute with Z_n. It is simple to observe that $\mathcal{L}$ is a matrix algebra closed under inversion, and that $A \in \mathcal{L}$ if and only if A is lower triangular and has a Toeplitz structure, i.e. $A_{i,j+1} = A_{i-1,j}$, $i = 2,\dots,n$, $j = 1,\dots,n-1$. This is also equivalent to say that $\mathcal{L} = \{p(Z_n) : p \text{ a polynomial}\}$[1], and therefore the space $\mathcal{L}$ of ltT matrices is a commutative matrix algebra.

As a consequence of the above arguments, the inverse A^{-1} of a matrix $A \in \mathcal{L}$ has a lower triangular Toeplitz (ltT) structure, and thus A^{-1} is completely determined as soon as its first column is known. It follows that a ltT linear system of n equations, $A\mathbf{x} = \mathbf{f}$ (such as the Ramanujan one in (1)), can be solved by the following two-phase procedure:

(i) Compute $\mathbf{z} \in \mathbb{C}^n$ such that $A\mathbf{z} = \mathbf{e}_1 = [1\,0\,\cdots\,0]^T$.
(ii) Multiply by $\mathbf{f}$ the ltT matrix with $\mathbf{z}$ as first column.

The procedure $\mathcal{G}_b$ described in this paper belongs to the above class of two-phase ltT systems solvers. So, in order to define it, one needs only to describe the algorithm $\mathcal{G}_b$ implementing phase (i). Essentially, in $\mathcal{G}_b$ first of all the ltT coefficient matrix of the system $A\mathbf{z} = \mathbf{e}_1$ is transformed, by a number $s = \log_b n$ of ltT left multiplications, into the identity matrix. This first part of the algorithm is a sort of Gaussian-elimination procedure where, at each step, $(b-1)/b$ of the remaining non null diagonals are nullified. Moreover, the dimension of the ltT transform used at the generic step is in fact b times smaller than in the previous step. Then, in the second part of the $\mathcal{G}_b$ algorithm, the same ltT transforms are applied in opposite order to $\mathbf{e}_1$, and since such process is particularly cheap when applied to vectors with a suitable sparse structure, such as $\mathbf{e}_1$, also this second part of the algorithm has a very low cost. Finally, $\mathcal{G}_b$ turns out to have a $O(sb^s)$ complexity, if $n = b^s$.

The $\mathcal{G}_b$ algorithm is not totally new. In particular, when $b = 2$ it seems to be related to a known power series inverting algorithm, presented in [21]. However, the very natural and simple matrix formulation of the $\mathcal{G}_b$ algorithm seems to be absent in literature. In fact, usually ltT systems solvers are written in terms of polynomial arithmetics since they solve, instead of the ltT system directly, the equivalent problem of finding the reciprocal of a power series. We think that our approach could be a good simple alternative and suitable, since the problem of solving a ltT system is also a matrix problem. These reasons suggest to include the $\mathcal{G}_b$ procedure in the wide literature on low complexity ltT solvers [14], [18], [4], [3], [6], [22], [23], [24], [21], [17], [19], [12].

We conclude by noting that the algorithm $\mathcal{G}_b$ can be applied, for example, to find any pair of two consecutive Bernoulli numbers. In

[1] This is easy to prove, however it also follows from the fact that Z_n is non-derogatory, i.e. the characteristic and minimum polynomials of Z_n are equal (see [15])

fact, by a remark in [8], this is equivalent to solve a ltT system of type $A\mathbf{z} = \mathbf{e}_1$. Whereas the $\mathcal{G}_b$ two-phase procedure can be used to find the first n Bernoulli numbers, by solving the ltT systems stated in [8], for example (1). In particular, $\mathcal{G}_3$ turns out to be the most suitable procedure to solve the sparse ltT system (1). Actually, (1) can be interpreted as the output of the first step of $\mathcal{G}_3$ applied to a *full* ltT system (whose solution yields Bernoulli numbers). Further steps of $\mathcal{G}_3$ – each nullifying 2/3 of the remaining non null diagonals – yield sparser and sparser ltT systems satisfied by Bernoulli numbers, and finally yield n Bernoulli numbers for any arbitrary n.

2. An algorithm for the solution of ltT linear systems

Let b be a positive integer greater than 1. We now present an algorithm $\mathcal{G}_b$ of complexity $O(n \log_b n)$ for the computation of $\mathbf{x}$ such that $A\mathbf{x} = [1\,0 \cdots 0]^T$, where A is a $n \times n$ ltT matrix, with $[A]_{11} = 1$ (which of course is not restrictive) and $n = b^s$ for some $s \in \mathbb{N}$.

2.1. **Multiply a ltT matrix by a vector.** We shall see that the $\mathcal{G}_b$ algorithm needs a function that implements the product of a ltT $b^j \times b^j$ matrix by a vector. Such operation can be performed in at most $c_b j b^j$ arithmetic operations. The latter result follows immediately, for example, from some well known representations of a generic Toeplitz matrix $T = (t_{i-k})_{i,k=1}^m$, $m = b^j$, in terms of circulant matrices, and from the fact that circulants can be multiplied by a vector via fast discrete transforms (FFT) [7]. For the sake of completeness, such representations are displayed here below:

$$T = \{\mathcal{C}(\mathbf{a})\}_m, \ \mathbf{a}^T = [t_0 \ t_{-1} \ \cdots \ t_{-m+1} \ \mathbf{0}^T_{(b-2)m+1} \ t_{m-1} \ \cdots \ t_1]; \tag{2}$$

$$\begin{aligned} T &= \mathcal{C}(\mathbf{a}) + \mathcal{C}_{-1}(\mathbf{a}'), \\ a_i &= \frac{1}{2}(t_{-i+1} + t_{m-i+1}), \ a'_i = \frac{1}{2}(t_{-i+1} - t_{m-i+1}), \ i = 1, \dots, m \end{aligned} \tag{3}$$

($t_m = 0$). Here $\mathcal{C}(\mathbf{a})$ ($\mathcal{C}_{-1}(\mathbf{a})$) denotes the circulant (skew circulant) matrix with first row $\mathbf{a}^T$, and $\{\mathcal{C}(\mathbf{a})\}_m$ denotes the upper-left $m \times m$ submatrix of $\mathcal{C}(\mathbf{a})$.[2]

2.2. **Preliminary Lemmas.** Given a vector $\mathbf{v} = [v_0 \ v_1 \ v_2 \ \cdots]^T$, $v_i \in \mathbb{C}$ (briefly $\mathbf{v} \in \mathbb{C}^{\mathbb{N}}$), let $L(\mathbf{v})$ be the semi-infinite ltT matrix whose first

[2] Note that the two obvious procedures performing the multiplication ltT matrix $\times$ vector based on the representations (2), (3) of T hold for a generic (full) Toeplitz matrix T; so one guesses that better procedures may be introduced, ad hoc for the ltT case.

column is $\mathbf{v}$, i.e.

$$L(\mathbf{v}) = \sum_{k=0}^{+\infty} v_k Z^k, \quad Z = \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ & 1 & 0 & \\ & & \cdot & \cdot \end{bmatrix}.$$

Lemma 2.1 Let $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{C}^{\mathbb{N}}$. Then $L(\mathbf{a})L(\mathbf{b}) = L(\mathbf{c})$ if and only if $L(\mathbf{a})\mathbf{b} = \mathbf{c}$.

Proof. If $L(\mathbf{a})L(\mathbf{b}) = L(\mathbf{c})$, then the first column of $L(\mathbf{a})L(\mathbf{b})$ must be equal to the first column of $L(\mathbf{c})$, and these are the vectors $L(\mathbf{a})\mathbf{b}$ and $\mathbf{c}$, respectively. Conversely, assume that $L(\mathbf{a})\mathbf{b} = \mathbf{c}$ and consider the matrix $L(\mathbf{a})L(\mathbf{b})$. It is ltT, being a product of ltT matrices, and, by hypothesis, its first column, $L(\mathbf{a})\mathbf{b}$, coincides with the vector $\mathbf{c}$, which in turn is the first column of the ltT matrix $L(\mathbf{c})$. The thesis follows from the fact that ltT matrices are uniquely defined by their first columns. □

Remark. The above Lemma holds for any algebra of matrices uniquely defined by their first columns (or rows). A possible exhaustive list of these algebras can be deduced from a wide literature (see for instance [10], [2], [11], [5]).

Given a vector $\mathbf{v} = [v_0\, v_1\, v_2\, \cdots]^T \in \mathbb{C}^{\mathbb{N}}$, let E be the semi-infinite matrix with entries 0 or 1 which, applied to $\mathbf{v}$, has the effect of inserting $b-1$ zeros between two consecutive components of $\mathbf{v}$. Then

$$E = \begin{bmatrix} 1 & & & \\ \mathbf{0} & \mathbf{0} & & \\ 0 & 1 & & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \\ 0 & 0 & 1 & \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}, \ \mathbf{0} = \mathbf{0}_{b-1}, \ E^k = \begin{bmatrix} 1 & & & \\ \mathbf{0} & \mathbf{0} & & \\ 0 & 1 & & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \\ 0 & 0 & 1 & \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}, \ \mathbf{0} = \mathbf{0}_{b^k-1},$$

$$\mathbf{v} = \begin{bmatrix} 1 \\ v_1 \\ v_2 \\ \cdot \end{bmatrix}, \ E\mathbf{v} = \begin{bmatrix} 1 \\ \mathbf{0} \\ v_1 \\ \mathbf{0} \\ v_2 \\ \cdot \end{bmatrix}, \ L(E\mathbf{v}) = \begin{bmatrix} 1 & & & & & \\ \mathbf{0} & I & & & & \\ v_1 & \mathbf{0}^T & 1 & & & \\ \mathbf{0} & v_1 I & \mathbf{0} & I & & \\ v_2 & \mathbf{0}^T & v_1 & \mathbf{0}^T & 1 & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}, \ \mathbf{0} = \mathbf{0}_{b-1}.$$

Lemma 2.2 Let $\mathbf{u}, \mathbf{v} \in \mathbb{C}^{\mathbb{N}}$ with $u_0 = v_0 = 1$. Then $L(E\mathbf{u})E\mathbf{v} = EL(\mathbf{u})\mathbf{v}$, and, more in general, for each $k \in \mathbb{N}$, $L(E^k\mathbf{u})E^k\mathbf{v} = E^k L(\mathbf{u})\mathbf{v}$.

Proof. By inspecting the vectors $L(E\mathbf{u})E\mathbf{v}$ and $EL(\mathbf{u})\mathbf{v}$ one observes that they are equal. By multiplying E on the left of the identity $L(E\mathbf{u})E\mathbf{v} = EL(\mathbf{u})\mathbf{v}$ and using the same identity also for the vectors $E\mathbf{u}$ and $E\mathbf{v}$, in place of $\mathbf{u}$ and $\mathbf{v}$ respectively, one observes that it also holds $L(E^2\mathbf{u})E^2\mathbf{v} = E^2L(\mathbf{u})\mathbf{v}$. And so on. □

2.3. **The computation of the first column of the inverse of a $n \times n$ ltT matrix ($n = b^s$).** The $\mathcal{G}_b$ algorithm for the computation of $\mathbf{x}$ such that $A\mathbf{x} = \mathbf{e}_1$ consists of two parts. In the first one, some special ltT matrices are computed, whose successive left multiplication by the matrix A transforms A into the the identity. In the second part such matrices are successively left multiplied (in the opposite order) by the vector $\mathbf{e}_1$. The method appears as a kind of Gaussian elimination, where diagonals are nullified instead of columns. The overall cost $O(n \log_b n)$ comes from a double fact: at each step of the first part $(b-1)/b$ of the remaining non null diagonals are nullified by using ltT transforms of dimension b times smaller than in the previous step, and an analogous reduction of the dimensions of the ltT transforms involved is applied in the second part, by exploiting the particular sparse structure of $\mathbf{e}_1$.

The given $n \times n$ matrix A can be thought as the upper-left submatrix of a semi-infinite ltT matrix $L(\mathbf{a})$, whose first column is $[1\, a_1\, a_2\ \cdot\ a_{b^s-1}\, a_{b^s}\ \cdot\ ]^T \in \mathbb{C}^{\mathbb{N}}$. The two parts of the $\mathcal{G}_b$ algorithm are described here below. (More details, especially in the cases $b = 2, 3$, can be found in [9]).

Part I. Set $\mathbf{a}^{(0)} := \mathbf{a}$, and find $\hat{\mathbf{a}}^{(0)}$, $\mathbf{a}^{(1)} \in \mathbb{C}^{\mathbb{N}}$ such that

$$L(\mathbf{a}^{(0)})\hat{\mathbf{a}}^{(0)} = E\mathbf{a}^{(1)} = \begin{bmatrix} 1 \\ \mathbf{0} \\ a_1^{(1)} \\ \cdot \end{bmatrix},\ \mathbf{0} = \mathbf{0}_{b-1}\ \Big(\text{thus}\ \ L(\hat{\mathbf{a}}^{(0)})L(\mathbf{a}^{(0)}) = L(E\mathbf{a}^{(1)})\ \Big).$$

Then, for $k = 2, \dots, s$ find $\hat{\mathbf{a}}^{(k-1)}$, $\mathbf{a}^{(k)} \in \mathbb{C}^{\mathbb{N}}$ such that

$$L(\mathbf{a}^{(k-1)})\hat{\mathbf{a}}^{(k-1)} = E\mathbf{a}^{(k)} = \begin{bmatrix} 1 \\ \mathbf{0} \\ a_1^{(k)} \\ \cdot \end{bmatrix},\ \mathbf{0} = \mathbf{0}_{b-1}$$

$$\Big(\text{thus}\ \ L(E^{k-1}\mathbf{a}^{(k-1)})E^{k-1}\hat{\mathbf{a}}^{(k-1)} = E^k\mathbf{a}^{(k)} = \begin{bmatrix} 1 \\ \mathbf{0} \\ a_1^{(k)} \\ \cdot \end{bmatrix},\ \mathbf{0} = \mathbf{0}_{b^k-1},$$

$$\text{and}\quad L(E^{k-1}\hat{\mathbf{a}}^{(k-1)})L(E^{k-1}\mathbf{a}^{(k-1)}) = L(E^k\mathbf{a}^{(k)})\ \Big).$$

After the above s steps, we obtain the following identity

(4)
$$L(E^{s-1}\hat{\mathbf{a}}^{(s-1)})L(E^{s-2}\hat{\mathbf{a}}^{(s-2)})\ \cdots\ L(E\hat{\mathbf{a}}^{(1)})L(\hat{\mathbf{a}}^{(0)})L(\mathbf{a}^{(0)}) = L(E^s\mathbf{a}^{(s)})$$

where the upper left $b^s \times b^s$ submatrices of $L(\mathbf{a}^{(0)})$ and of $L(E^s\mathbf{a}^{(s)})$ are, respectively, the initial ltT matrix A and the identity matrix, i.e.

$$L(\mathbf{a}^{(0)}) = \left[\begin{array}{ccc|cc} & A & & \mathbf{0} & O \\ \hline a_{b^s} & \cdots & a_1 & 1 & \mathbf{0}^T \\ & \cdot & & \cdot & \cdots \end{array}\right], \quad L(E^s\mathbf{a}^{(s)}) = \left[\begin{array}{c|cc} I_{b^s} & \mathbf{0} & O \\ \hline a_1^{(s)}\mathbf{e}_1^T & 1 & \mathbf{0}^T \\ \cdot & \cdot & \cdots \end{array}\right].$$

Part II. By (4), any system of type $L(\mathbf{a}^{(0)})\mathbf{z} = E^{s-1}\mathbf{v}$, $\mathbf{v} \in \mathbb{C}^{\mathbb{N}}$, is equivalent to the following linear system

$$\begin{aligned} L(E^s\mathbf{a}^{(s)})\mathbf{z} &= L(\hat{\mathbf{a}}^{(0)})L(E\hat{\mathbf{a}}^{(1)}) \cdots L(E^{s-2}\hat{\mathbf{a}}^{(s-2)})L(E^{s-1}\hat{\mathbf{a}}^{(s-1)})E^{s-1}\mathbf{v} \\ &= L(\hat{\mathbf{a}}^{(0)})EL(\hat{\mathbf{a}}^{(1)})E \cdots EL(\hat{\mathbf{a}}^{(s-2)})EL(\hat{\mathbf{a}}^{(s-1)})\mathbf{v}. \end{aligned}$$

The matrices involved in the vector on the right hand side are lower triangular. Moreover, the upper left square submatrices of E of dimensions $b^j \times b^j$, $j = s, \dots, 2$, have their last $b^{j-1}(b-1)$ columns null. As a consequence of these two observations, any vector $\{\mathbf{z}\}_n$, $n = b^s$, such that

$$A\{\mathbf{z}\}_n = \{L(\mathbf{a})\}_n\{\mathbf{z}\}_n = \begin{bmatrix} v_0 \\ \mathbf{0} \\ v_1 \\ \mathbf{0} \\ \cdot \\ v_{b-1} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{0} = \mathbf{0}_{b^{s-1}-1}$$

(for example, the vector $A^{-1}\mathbf{e}_1$ we are looking for), can be represented as follows

$$\begin{aligned} \{\mathbf{z}\}_n &= \{L(\hat{\mathbf{a}}^{(0)})\}_n\{E\}_n\{L(\hat{\mathbf{a}}^{(1)})\}_n\{E\}_n \\ &\quad \cdots \{E\}_n\{L(\hat{\mathbf{a}}^{(s-2)})\}_n\{E\}_n\{L(\hat{\mathbf{a}}^{(s-1)})\}_n\{\mathbf{v}\}_n \\ &= \{L(\hat{\mathbf{a}}^{(0)})\}_n\{E\}_{n,\frac{n}{b}}\{L(\hat{\mathbf{a}}^{(1)})\}_{\frac{n}{b}}\{E\}_{\frac{n}{b},\frac{n}{b^2}} \\ &\quad \cdots \{E\}_{b^3,b^2}\{L(\hat{\mathbf{a}}^{(s-2)})\}_{b^2}\{E\}_{b^2,b}\{L(\hat{\mathbf{a}}^{(s-1)})\}_b\{\mathbf{v}\}_b, \end{aligned} \tag{5}$$

where $\{M\}_{j,k}$ ($\{M\}_k$) denotes the $j \times k$ ($k \times k$) upper left submatrix of M. The latter formula allows to compute $\{\mathbf{z}\}_n$ efficiently.

Let us resume and count the operations required to implement the $\mathcal{G}_b$ algorithm. In the following, n is equal to b^s and $\mathbf{0}$ denotes $\mathbf{0}_{b-1}$. In the first part, for $k = 1, \dots, s$ one has to compute, by performing $\varphi_{n/b^{k-1}}$ arithmetic operations, the vectors $\{\hat{\mathbf{a}}^{(k-1)}\}_{n/b^{k-1}}$ and $\{\mathbf{a}^{(k)}\}_{n/b^k}$, i.e. scalars $\hat{a}_i^{(k-1)}$ and $a_i^{(k)}$ such that

$$\begin{bmatrix} 1 & & & & \\ a_1^{(k-1)} & 1 & & & \\ a_2^{(k-1)} & a_1^{(k-1)} & 1 & & \\ \cdot & \cdot & \cdot & \cdot & \\ a_{\frac{n}{b^{k-1}}-1}^{(k-1)} & \cdot & a_2^{(k-1)} & a_1^{(k-1)} & 1 \end{bmatrix} \begin{bmatrix} 1 \\ \hat{a}_1^{(k-1)} \\ \hat{a}_2^{(k-1)} \\ \cdot \\ \hat{a}_{\frac{n}{b^{k-1}}-1}^{(k-1)} \end{bmatrix} = \begin{bmatrix} 1 \\ \mathbf{0} \\ a_1^{(k)} \\ \mathbf{0} \\ \cdot \\ a_{\frac{n}{b^k}-1}^{(k)} \\ \mathbf{0} \end{bmatrix},$$

$k = 1, \ldots, s$ (for $k = s$ only the $\hat{a}_i^{(k-1)}$ needs to be computed). Note that, once $\{\hat{\mathbf{a}}^{(k-1)}\}_{n/b^{k-1}}$ is available, the vector $\{\mathbf{a}^{(k)}\}_{n/b^k}$ can be computed by a $n/b^{k-1} \times n/b^{k-1}$ ltT by vector product.[3]

In the second part, when applying (5), one has to compute the $b \times b$ ltT by vector product $\{L(\hat{\mathbf{a}}^{(s-1)})\}_b\{\mathbf{v}\}_b$ (which requires no operation if $v_0 = 1$, $v_i = 0$, $i \geq 1$), and $b^j \times b^j$ ltT by vector products of type

$$\{L(\hat{\mathbf{a}}^{(s-j)})\}_{b^j} \begin{bmatrix} 1 \\ \mathbf{0} \\ \bullet_1 \\ \mathbf{0} \\ \cdot \\ \bullet_{b^{j-1}-1} \\ \mathbf{0} \end{bmatrix}, \quad j = 2, \ldots, s-1, s \quad (\bullet_k \in \mathbb{C}).$$

Now, we already know that the number of arithmetic operations required by a $b^j \times b^j$ ltT matrix by vector product is bounded by $c_b j b^j$. Assume moreover that

(*) *the number of arithmetic operations required to compute* $\{\hat{\mathbf{a}}^{(s-j)}\}_{b^j}$ *is bounded by* $\hat{c}_b j b^j$ *for some constant* $\hat{c}_b$ ($j = s, \ldots, 2, 1$).

Then the number φ_{b^j} is bounded by $(\hat{c}_b + c_b) j b^j$, $j = s, \ldots, 2, 1$, and we can conclude that the total cost of the $\mathcal{G}_b$ algorithm is smaller than $(\hat{c}_b + 2c_b) \sum_{j=1}^{s} j b^j = O(s b^s) = O(n \log_b n)$. In particular, if $v_0 = 1$, $v_i = 0$, $i \geq 1$, such amount of operations yields the first column of A^{-1}, and therefore the $\mathcal{G}_b$ algorithm yields a two-phase ltT linear system solver $\mathcal{G}_b$ of complexity $O(n \log_b n)$ (see (i),(ii) in the Introduction).

In the next subsection 2.4 we will see that the assumption (*) is satisfied for $b = 2, 3$, and, by Proposition 2.3, for any larger value of b; it is enough to choose a larger constant $\hat{c}_b$. For example, for $b = 2$ it is immediate to see that a vector $\hat{\mathbf{a}}$ such that $L(\mathbf{a})\hat{\mathbf{a}} = E\mathbf{a}^{(1)}$ is available

[3] Such product can be in fact replaced with a number b of $n/b^k \times n/b^k$ ltT by vector products. This assertion, whose proof is left to the reader, is true for any $k = 1, \ldots, s-1$.

with no computations. To this aim it is sufficient to observe that

(6)

$$L(\mathbf{a})\big(\mathbf{e}_1+\sum_{i=1}^{+\infty}(-1)^i a_i\mathbf{e}_{i+1}\big) = \mathbf{e}_1+\sum_{i=1}^{+\infty}\delta_{i=0 \bmod 2}\big(2a_i+\sum_{j=1}^{i-1}(-1)^j a_j a_{i-j}\big)\mathbf{e}_{i+1}.$$

Thus $\hat{c}_2 = 0$.

2.4. **Observations on the algorithm's core.** Given the vector $\mathbf{a} \in \mathbb{C}^{\mathbb{N}}$, the problem of the computation of $\hat{\mathbf{a}} \in \mathbb{C}^{\mathbb{N}}$ such that $L(\mathbf{a})\hat{\mathbf{a}} = E\mathbf{a}^{(1)}$, for some $\mathbf{a}^{(1)} \in \mathbb{C}^{\mathbb{N}}$ can also be seen as a polynomial arithmetic problem. In fact, due to Lemma 2.1, the identity $L(\mathbf{a})\hat{\mathbf{a}} = E\mathbf{a}^{(1)}$ is equivalent to the equality $L(\mathbf{a})L(\hat{\mathbf{a}}) = L(E\mathbf{a}^{(1)})$, i.e.

$$(\sum_{k=0}^{+\infty} a_k Z^k)(\sum_{k=0}^{+\infty}\hat{a}_k Z^k) = \sum_{k=0}^{+\infty} a_k^{(1)} Z^{bk}.$$

Therefore the polynomial arithmetic problem can be stated as follows.

Given $a(z) = \sum_{k=0}^{+\infty} a_k z^k$, find a power series $\hat{a}(z) = \sum_{k=0}^{+\infty}\hat{a}_k z^k$ such that

(7) $$\hat{a}(z)a(z) = a_0^{(1)} + a_1^{(1)}z^b + a_2^{(1)}z^{2b} + \ldots =: a^{(1)}(z^b)$$

for some coefficients $a_i^{(1)}$.

It is possible to describe explicitly a power series $\hat{a}(z)$ that realizes such transformation (7), in fact the following result holds

Proposition 2.3 Let t is a b-th principal root of the unity, i.e. $t \in \mathbb{C}$, $t^b = 1$, $t^i \neq 1$ for $0 < i < b$. Then $\hat{a}(z) = a(zt)a(zt^2)\cdots a(zt^{b-1})$ satisfies the identity (7). Moreover, if the coefficients of a are real, then the coefficients of $\hat{a}$ are real.

Proof. Observe that a power series $p(z) = \sum_{k=0}^{+\infty} a_k z^k$ is equal to $q(z^b)$ for some power series q (i.e. p is a power series in z^b) if and only if $p(tz) = p(z)$. In fact, assume that $p(z) = q(z^b)$. Then $p(tz) = q((tz)^b) = q(t^b z^b) = q(z^b) = p(z)$, thus $p(tz) = p(z)$. Viceversa, if $p(tz) = p(z)$, then

$$\sum_{k=0}^{+\infty} a_k z^k = \sum_{k=0}^{+\infty} a_k (tz)^k,$$

but this equality implies $a_k = 0$ for all k, $t^k \neq 1$, i.e. for all k which are not divisible by b. In other words, p is a power series in z^b (there exists q s.t. $p(z) = q(z^b)$).

As a consequence of this result, the required equality $\hat{a}(z)a(z) = a^{(1)}(z^b)$ holds for some power series $a^{(1)}$ if and only if $\hat{a}(tz)a(tz) = \hat{a}(z)a(z)$. Now the thesis follows because the latter identity is obviously verified when $\hat{a}(z) = a(zt)a(zt^2)\cdots a(zt^{b-1})$. $\square$

Let us consider some corollaries of Proposition 2.3. For $b = 2$ we have $\hat{a}(z) = a(-z)$, that is we regain the result (6). It is clear that $a(-z)a(z) = a_0^{(1)} + a_1^{(1)}z^2 + a_2^{(1)}z^4 + \dots$ (compare with the Graeffe root-squaring method [16], [13], [21], [1]). In this case the coefficients of $\hat{a}$ are available with no computations, we only need to compute the new coefficients $a_i^{(1)}$. Thus (*) is true with $\hat{c}_2 = 0$.

For $b = 3$ we have $\hat{a}(z) = a(zt)a(zt^2)$, $t = e^{\mathbf{i}\frac{2\pi}{3}}$. By Proposition 2.3 the equalities $a(z)a(zt)a(zt^2) = a_0^{(1)} + a_1^{(1)}z^3 + a_2^{(1)}z^6 + \dots$ and

$$L(\hat{\mathbf{a}})L(\mathbf{a}) = L(E\mathbf{a}^{(1)}), \quad E = \begin{bmatrix} 1 & & & \\ 0 & & & \\ 0 & & & \\ 0 & 1 & & \\ 0 & 0 & & \\ 0 & 0 & & \\ 0 & 0 & 1 & \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}, \tag{8}$$

hold for some vector $\mathbf{a}^{(1)}$, and the coefficients of $\hat{a}(z) = a(zt)a(zt^2)$ are real, provided that the coefficients of a are real. This time, the coefficients of $\hat{a}$ are not easily readable from the coefficients of a. In order to represent and calculate the coefficients of $\hat{a}$, observe that the polynomial equality $\hat{a}(z) = a(zt)a(zt^2)$ is equivalent to the matrix identity $L(\hat{\mathbf{a}}) = L(\mathbf{p})L(\mathbf{q})$, where $\mathbf{p}$ and $\mathbf{q}$ are the vectors of the coefficients of the power series $a(zt)$ and $a(zt^2)$, respectively. Therefore we obtain the formula

$$\hat{\mathbf{a}} = L(\mathbf{p})\mathbf{q}, \quad p_i = a_i t^i, \; q_i = a_i t^{2i}, \tag{9}$$

which, taking into account that $t = e^{\mathbf{i}\frac{2\pi}{3}}$, becomes

$$\hat{\mathbf{a}} = L(\hat{\mathbf{p}})\hat{\mathbf{p}} + L(\hat{\mathbf{q}})\hat{\mathbf{q}}, \tag{10}$$

$$\hat{p}_i = \begin{cases} a_i & i = 3j \\ -\frac{1}{2}a_i & i \neq 3j \end{cases}, \quad \hat{q}_i = \begin{cases} 0 & i = 3j \\ -\sqrt{3}a_i/2 & i = 3j+1 \\ \sqrt{3}a_i/2 & i = 3j+2 \end{cases}, \; j = 0, 1, 2, \dots$$

Note that (10) shows that the $\hat{a}_i$ are real if the a_i are real. So, by (9), a vector $\hat{\mathbf{a}}$ satisfying (8) is computable by one ltT matrix-vector product, and therefore (choose $b = 3$ in (2), (3)) the first 3^j entries of such vector can be obtained with $c_3 j 3^j$ arithmetic operations, i.e. (*) is true with $\hat{c}_3 = c_3$.

Thus, if A is a $3^s \times 3^s$ ltT matrix, then the first column of A^{-1} (or, equivalently, the solution of any system $A\mathbf{x} = \mathbf{f}$) can be computed by the $\mathcal{G}_3$ algorithm with $O(s3^s)$ arithmetic operations. Just set $b = 3$ in Part I and II of the $\mathcal{G}_b$ algorithm (see the previous section and, for more details, [9]). Note that, in the $\mathcal{G}_3$ algorithm, the ltT matrix A is transformed into the identity by s steps, each consisting in nullifying 2/3 of the remaining non null diagonals. It follows that $\mathcal{G}_3$ must be preferred to any other $\mathcal{G}_b$ algorithm (and, in particular, to $\mathcal{G}_2$) in solving the Ramanujan ltT system (1), where the coefficient matrix has two null diagonals which alternate the non null ones. In fact, $\mathcal{G}_3$ continues in the most natural way the work done to obtain (1) from a full ltT system.

Finally, in the general case where E in equality $L(\mathbf{a})\hat{\mathbf{a}} = E\mathbf{a}^{(1)}$ inserts $b-1$ zeros between two consecutive components of $\mathbf{a}^{(1)}$, one can easily prove that the first b^j entries of $\hat{\mathbf{a}}$ can be obtained in no more than $(b-2)c_b j b^j$ arithmetic operations, i.e. (*) is true with $\hat{c}_b = (b-2)c_b$. The proof of this fact is left to the reader.

Remark. There are infinite possible choices of $\hat{\mathbf{a}}$ for which (8) holds for some vector $\mathbf{a}^{(1)}$. Looking for the simplest one, i.e. for *those* $\hat{a}_i$ *which are such that* $(L(\mathbf{a})\hat{\mathbf{a}})_i = 0$*,* $i = 2, 3, 5, 6, 8, 9, \ldots$*, and, in the same time, have the simplest expression in terms of the* a_i, we have obtained the following formula for such "optimal" $\hat{a}_i$:

(11)

$$\hat{a}_i^{opt} = -\sum_{r=0}^{\lfloor \frac{i-1}{2} \rfloor} a_r a_{i-r} + \delta_{i=0 \bmod 2} a_{\frac{i}{2}}^2 + 3 \begin{cases} \sum_{s \geq \frac{3-i}{6}}^{0} a_{\frac{i-3}{2}+3s} a_{\frac{i+3}{2}-3s} & i \text{ odd} \\ \sum_{s \geq \frac{6-i}{6}}^{0} a_{\frac{i-6}{2}+3s} a_{\frac{i+6}{2}-3s} & i \text{ even} \end{cases} .$$

Now, our optimal $\hat{a}_i^{opt}$ turn out to be equal to the $\hat{a}_i$ obtained by setting $b = 3$ in Proposition 2.3, i.e. to the $\hat{a}_i$ defined by (9)-(10). Thus the statement of Proposition 2.3 may be completed with the assertion – not yet proved for any n – that the power series $\hat{a}(z)$ proposed as solution of problem (7) is "optimal".

Acknowledgements. Some of the contents of this work have been the subject of a communication held at the meeting "Due Giorni di Algebra Lineare Numerica 2012" (Genova, 16–17 Febbraio 2012; speaker: Carmine Di Fiore). See www.dima.unige.it/∼dibenede/2gg/home.html. The $\mathcal{G}_b$ algorithm and its possible application in the computation of Bernoulli numbers have been taught to the students of the Rome-Moscow schools "Matrix Methods and Applied Linear Algebra" 2012 and 2014 [9]. We are also indebted to Carlo Pagano and Alessandro Spadoni for their useful comments on this paper.

References

[1] E. H. Bareiss, Resultant procedure and the mechanization of the Graeffe process, Journal of the ACM, 7 (1960), pp.346–386

[2] R. Bevilacqua, C. Di Fiore, P. Zellini, h-Space structure in matrix displacement formulas, *Calcolo*, 33 (1996), pp.11–35

[3] D. Bini, V. Pan, Polynomial division and its computational complexity, *J. Complexity*, 2 (1986), pp.179–203

[4] D. Bini, V. Pan, Polynomial and Matrix Computations, Volume 1, Fundamental Algorithms, Birkhäuser, Boston, MA, 1994.

[5] A. Bortoletti, C. Di Fiore, On a set of matrix algebras related to discrete Hartley-type transforms, *Linear Algebra Appl.*, 366 (2003), pp.65–85

[6] D. Commenges, M. Monsion, Fast inversion of triangular Toeplitz matrices, *IEEE Trans. Automat. Control*, AC-29 (1984), pp.250–251

[7] P. J. Davis, *Circulant matrices*, Wiley, New York, 1979

[8] C. Di Fiore, F. Tudisco, P. Zellini, *Lower triangular Toeplitz-Ramanujan systems whose solution yields the Bernoulli numbers*, submitted for publication

[9] C. Di Fiore et al, Bernoulli, Ramanujan, Toeplitz and triangular matrices, http://www.mat.uniroma2.it/∼difiore/perGenova, genova, Bernoulli, RomeMoscow2014

[10] C. Di Fiore, P. Zellini, Matrix decompositions using displacement rank and classes of commutative matrix algebras, *Linear Algebra Appl.*, 229 (1995), pp.49–99

[11] C. Di Fiore, P. Zellini, Matrix algebras in optimal preconditioning, *Linear Algebra Appl.*, 335 (2001), pp.1–54

[12] D. Harvey, Faster algorithms for the square root and reciprocal of power series, *Mathematics of Computation*, 80 (2011), pp.387–394

[13] F. B. Hildebrand, Introduction to Numerical Analysis, McGraw-Hill, 2nd edition, 1974

[14] H. T. Kung, On computing reciprocals of power series, *Numerische Mathematik*, 22 (1974), pp.341–348

[15] P. Lancaster, M. Tismenetsky, *The Theory of Matrices, second edition with applications*, Computer Science and Applied Mathematics, Academic Press, Orlando, Florida, 1985 (pp. 416–420)

[16] D. H. Lehmer, The Graeffe process as applied to power series, *Math. Comp.*, 1 (1945), pp.377–383; Corrigendum: *Math. Comp.*, 3 (1948), 227

[17] F. R. Lin, W. K. Ching, M. K. Ng, Fast inversion of triangular Toeplitz matrices, *Theoretical Computer Science*, 315 (2004), pp.511–523

[18] M. Morf, Doubling algorithms for Toeplitz and related equations, Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '80 (April 1980, Stanford Univ., California), 5, pp.954–959

[19] B. J. Murthy, Acceleration of the inversion of triangular Toeplitz matrices and polynomial division, in *Computer Algebra in Scientific Computing*, Lecture Notes in Computer Science, 6885, Springer, 2011, pp.321–332.

[20] S. Ramanujan, Some properties of Bernoulli numbers (J. Indian Math. Soc. 3 (1911), 219–234), in *Collected Papers of Srinivasa Ramanujan*, (Edited by G. H. Hardy, P. V. Seshu Aiyar, and B. M. Wilson), New York, Chelsea, 1962

[21] A. Schönhage, Variations on computing reciprocals of power series. *Inform. Process. Lett.*, 74 (2000), pp.41–46

[22] W. F. Trench, Explicit inversion formulas for Toeplitz band matrices, *SIAM J. Alg. Disc. Meth.*, 6 (1985), pp.546–554

[23] W. F. Trench, A note on solving nearly triangular Toeplitz systems, *Linear Algebra Appl.*, 93 (1987), pp.57–65

[24] W. F. Trench, Inverses of lower triangular Toeplitz matrices, http://ramanujan.math.trinity.edu/wtrench/research/papers/TN-6.pdf

Dipartimento di Matematica, Università di Roma "Tor Vergata", Via della ricerca scientifica, 00133, Rome, Italy

E-mail address: `difiore@mat.uniroma2.it`