Summing Symbols in Mutual Recurrences*

Berkeley R. Churchill[1] and Edmund A. Lamagna[2]

[1] University of California, Santa Barbara, CA, USA
Department of Mathematics
[2] University of Rhode Island, Kingston, RI, USA
Department of Computer Science and Statistics

Abstract. The problem of summing a set of mutual recurrence relations with constant coefficients is investigated. A method is presented for summing an order d system of the form $A(n) = \sum_{i=1}^{d} M_i A(n-i) + G(n)$, where $A, G : \mathbb{N} \to K^m$ and $M_1, \ldots, M_d \in M_m(K)$ for some field K and natural number m. The procedure expresses the sum $\sum_{i=0}^{n} A(i)$ in terms of $A(n), \ldots, A(n-d)$, initial conditions and sums of the inhomogeneous term $G(n)$.

1 Problem Statement

An important task in computer algebra systems is evaluating indefinite sums, that is computing values $s_n = \sum_{k=0}^{n} a_k$ where the a_k are some sequence depending only on k. Today many functions can be summed, in part due to the pioneering work of many researchers [3], [5], [7]. Nonetheless, there are still countless instances where we lack algorithms to sum particular expressions, or the algorithms that exist are inefficient or produce undesirable outputs.

One area of interest is summing recurrence relations. Summing any a_k is a special case of computing the value of A_n where $A_n = A_{n-1} + a_k$ and $A_0 = 0$. Recurrence relations arise frequently in algorithm analysis and numerical analysis of differential equations. The classical example is the Fibonacci sequence, defined as a function $f : \mathbb{N} \to \mathbb{N}$ [3] given by $F(n) = F(n-1) + F(n-2) \quad \forall n \geq 2$ with $F(0) = 0$, $F(1) = 1$. It is well known[4] that this sequence satisfies the property $\sum_{i=0}^{n} F_i = F_{n+2} - 1$.

This identity is nice because it presents the sum in terms of the original Fibonacci symbol. An even trickier situation is a system of linear recurrences, often referred to as mutual recurrences in the literature. Consider the following example: $A, B : \mathbb{N} \to \mathbb{Q}$ satisfy $A(n+2) - A(n+1) - A(n) - B(n) = 1$ and $-A(n) + B(n+2) - B(n+1) - B(n) = 1$ with $A(0) = B(0) = 0$ and $A(1) = B(1) = 1$. How could one write an algorithm that computes an identical expression for

* This work was supported in part by the National Science Foundation under a Research Experiences for Undergraduates (REU) grant, NSF Award No. 1004409.

[3] We use $\mathbb{N} = \{n \in \mathbb{Z} | n \geq 0\}$.

[4] $F(0) = F(2) - 1$ holds true. The proof is by induction: suppose $\sum_{i=0}^{n} F_i = F_{n+2} - 1$ and add F_{n+1} to both sides of the equation to verify the formula.

$\sum_{i=0}^{n} A(i)$ in terms of the symbols A and B themselves? How could this be generalized to deal with any such system? In this paper we present a procedure which can compute the sum in a closed form any time the inhomogeneous term can be summed repeatedly.

2 Related Work

The primary inspiration for this work is Ravenscroft and Lamagna's work on summation of a single linear recurrence. They provide an efficient algorithm to express the sum of a homogeneous recurrence $A(n) = \sum_{i=1}^{d} m_i A(n-i)$ in terms of $A(n-d), \ldots, A(n-1)$ by using a "summing factor" to deal with recurrences that initially appear to be degenerate. They also provide a technique that handles some inhomogeneous terms [8].

Several authors study summing C-finite sequences (those determined by recurrence relations with constant coefficients). Greene and Wilf provide an algorithm to sum a general form of products of C-finite sequences [4]. Kauers and Zimmermann study determining whether summation relationships exist between different C-finite sequences [6].

Work on P-finite sequences (those determined by recurrence relations with polynomial coefficients) has also been done. Abramov and van Hoeij discuss summing P-finite sequences in terms of the original coefficients [1]. Chyzak generalizes the works of Gosper [3] and Zeilberger [10] to sum P-finite sequences that are not hypergeometric [2]. Schneider extends Karr's approach [5] to P-finite sequences as well [9].

3 Systems of Mutual Recurrences

Definition 1 (Mutual Recurrence). *Let K be a field, and $m, d \in \mathbb{Z}^+$. A system of mutual linear recurrence relations with constant coefficients on K of order d in m variables is a set of m functions $A_1(n), \ldots, A_m(n)$ mapping $\mathbb{N}$ into K satisfying*

$$\begin{pmatrix} A_1(n) \\ A_2(n) \\ \vdots \\ A_m(n) \end{pmatrix} = M_1 \begin{pmatrix} A_1(n-1) \\ A_2(n-1) \\ \vdots \\ A_m(n-1) \end{pmatrix} + \cdots + M_d \begin{pmatrix} A_1(n-d) \\ A_2(n-d) \\ \vdots \\ A_m(n-d) \end{pmatrix} + \begin{pmatrix} g_1(n) \\ g_2(n) \\ \vdots \\ g_m(n) \end{pmatrix}$$

for some $M_1, \ldots, M_d \in M_m(K)$ and $g_1, \ldots, g_m$ mapping $\mathbb{N} \to K$. Typically we will refer to this as a "mutual recurrence".

We call the vector containing the $g_i(n)$ the inhomogeneous term. If this inhomogeneous term is zero, the mutual recurrence is homogeneous. We call the values $\{A_i(j) : 1 \le i \le m, 0 \le j < d\}$ the *initial conditions* for the recurrence. The notation in this definition is used throughout the paper whenever a specific mutual recurrence is being considered.

Example 1. We will use the following example of a mutual recurrence to demonstrate computational procedures throughout the paper. For this example $m = 2$, so for convenience we use $A(n)$ to denote $A_1(n)$ and $B(n)$ to denote $A_2(n)$. $A(n) = 2A(n-1) + B(n-1)$, $B(n) = A(n-1) + 2B(n-2)$.

This may be written in the form stated in the definition as

$$\begin{pmatrix} A(n) \\ B(n) \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} A(n-1) \\ B(n-1) \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} A(n-2) \\ B(n-2) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Here the order is $d = 2$ and, coincidentally, $m = 2$ as well. □

4 Homogeneous Case

The homogeneous case can be reduced to a set of C-finite sequences easily and then solved with any of the existing methods, such as [1], [2], [4], [8]. To do so we write the system as follows:

$$(E-2)A(n) - B(n) = 0, \qquad (1)$$

$$(E^2-2)B(n) - EA(n) = 0. \qquad (2)$$

Here E is the shift operator defined as $Ef(n) = f(n+1)$. Regarding the above as a system of equations with coefficients in $\mathbb{Q}(E)$, we can find C-finite relations explicitly for A and B. In this case we could multiply (1) by (E^2-2) and add (2) leaving $(E^3 - 2E^2 - 3E + 4)A(n) = 0$. This demonstrates that A is C-finite and can be summed. $B(n)$ can be handled similarly. This approach works in general for homogeneous systems.

However, by combining an inhomogeneous term G, it is possible to construct a sequence that is not C-finite or P-finite. Yet so long as we can sum G using some algorithm, the method we present can still be used to sum the mutual recurrence.

5 Inhomogeneous Case

For the following discussion, fix a particular mutual recurrence and use the notation provided in the definition. Our goal is to compute $\sum_{j=0}^{n} A_i(j)$ for each $i \in \{1, \ldots, m\}$ and express the answer in terms of $\mathrm{Span}\{A_i(n-j) : 1 \le i \le m, 0 \le j \le d\}$, the initial conditions and possibly an inhomogeneous term. For any function $f : \mathbb{N} \to K$, define $S(f(n)) = \sum_{i=d}^{n} f(i)$. S will henceforth be known as the summation operator. Recursively define $S_i^j(n) = S(S_i^{j-1}(n))$ and $S_i^1(n) = S_i(n) = S(A_i(n))$. This operator corresponds to the notion of summing that allowed Ravenscroft and Lamagna to symbolically sum linear recurrences [8].

It becomes convenient to write a mutual recurrence as follows. In the following equation, the leftmost matrix is a block matrix, while the others are not. Check that the dimensions of the matrices are, from left to right, $m \times m(d+1)$, $m(d+1) \times 1$ and $m \times 1$, and that the following equation is identical to the definition provided earlier.

$$\begin{pmatrix} I & -M_1 & -M_2 & \cdots & -M_d \end{pmatrix} \begin{pmatrix} A_1(n) \\ A_2(n) \\ \vdots \\ A_m(n) \\ \hline \vdots \\ \hline A_1(n-d) \\ A_2(n-d) \\ \vdots \\ A_m(n-d) \end{pmatrix} = \begin{pmatrix} g_1(n) \\ g_2(n) \\ \vdots \\ g_m(n) \end{pmatrix}.$$

This equation can be manipulated using the standard three row operations along with the S operator without changing its validity. This means we may perform the following operations: (i) multiply rows by scalars, (ii) add any row times a scalar to another row, (iii) swap rows and (iv) apply the S operator to any row.

The S operator now deserves some attention. Let V be a vector space over K with basis $\beta = \{A_i(n-j) : 1 \leq i \leq m, 0 \leq j \leq d\} \cup \{S_i^j(n) : 1 \leq i \leq m, j \in \mathbb{Z}^+\} \cup \{1\}$. Throughout our work S will only be applied to functions of the form $v(n)+i(n)$ where $v \in V$ and $i(n)$ is an inhomogeneous term that depends only on n, and never on any function of the A_i. Therefore to understand S we need only understand how S acts on β; summing the inhomogeneous parts yields other inhomogeneous parts, and this may be accomplished via other methods [3], [5], [7].

From the definitions, we already know $S(S_i^j(n)) = S_i^{j+1}(n)$. The others are not hard to compute as $S(A_i(n-j)) = A_i(d-j)+A_i(d-j+1)+\cdots+A_i(n-j) = \sum_{k=d-j}^{d-1} A_i(k) + \sum_{k=d}^{n} A_i(k) - \sum_{k=n-j+1}^{n} A_i(k) = S_i(n) - \sum_{k=n-j+1}^{n} A_i(k) + \sum_{k=d-j}^{d-1} A_i(k)$ and $S(1) = 1+1+\cdots+1 = (n-d+1)$. Now consider applying S to a row of a matrix equation. By this we mean expanding the entire row of the equation, applying S to both the right and left hand sides, and then returning it to matrix form. Unless the row only contains an inhomogeneous term and nothing else (this does not occur in our procedure), applying S will create a new column including an $S_i^j(n)$. Therefore, we need to expand our matrix equation by t blocks of size $m \times m$. t will be assigned a definite value later. Here is the new matrix equation, now expanded:

$$\left(0 \cdots 0 \middle| I \; -M_1 \; \cdots \; -M_d\right) *$$
$$\left(S_1^t(n) \cdots S_m^t(n) \middle| \cdots \middle| S_1(n) \cdots S_m(n) \middle| A_1(n) \cdots A_m(n) \middle| \cdots \middle| A_1(n-d) \cdots A_m(n-d)\right)^t$$
$$= \begin{pmatrix} g_1(n) \\ g_2(n) \\ \vdots \\ g_m(n) \end{pmatrix}$$

For brevity we will notationally abbreviate equations of the above form as an augmented matrix like this,

$$\left(0\;0 \cdots 0 \middle| I \; -M_1 \; -M_2 \cdots -M_d \middle| G\right).$$

Here G denotes the column of inhomogeneous terms. In the following we will explicitly demonstrate the action of the summation operator on an augmented matrix.

Suppose we start with the following augmented matrix,

$$\left(0\;0 \cdots 0\; B_j\; B_{j-1} \cdots B_1 \middle| C_1\; C_2 \cdots C_k \middle| G\right)$$

where $B_1, \ldots, B_j$ and $C_1, \ldots, C_k$ are $m \times m$ matrices. In the expanded matrix equation, B_j is multiplied by the block matrix containing the $S_1^j(n), \ldots, S_m^j(n)$ as rows. A row of their product is therefore a linear combination of $S_i^j(n)$. Applying the summation operator creates an identical linear combination of $S_1^{j+1}(n), \ldots, S_m^{j+1}(n)$. The same logic applies to $B_{j-1}, \ldots, B_1$. In block matrix form, all the $B_1, \ldots, B_j$ appear to be shifted one block to the left after applying the summation operator. The result looks like

$$\left(0\;0 \cdots 0\; B_j\; B_{j-1} \cdots B_1 \; * \middle| * \; * \cdots * \middle| *\right).$$

To determine the block matrix represented by the leftmost asterisk, consider when $S_i(n)$ appears in the image of the S operator; it appears once for every occurrence of $A_i(n-j)$ (for any $j \in \mathbb{N}$) in the preimage. This implies $\sum_{i=1}^k C_i$ is the value of the leftmost asterisk.

For the kth asterisk to the right of the separator, the number of $A_i(n-k)$ in the image of S is given by the negation of the number of $A_i(n-l)$ in the preimage, where $l > k$. This result is best stated as a lemma.

Lemma 1. *Given an augmented block matrix of the form*

$$\left(0\;0 \cdots 0\; B_j\; B_{j-1} \cdots B_1 \middle| C_1\; C_2 \cdots C_k \middle| G\right)$$

applying S to each row yields a new block matrix

$$\left(0\;0 \cdots 0\; B_j\; B_{j-1} \cdots B_1\; \textstyle\sum_{i=1}^k C_i \middle| -\sum_{i=2}^k C_i \; -\sum_{i=3}^k C_i \cdots -C_k \; 0 \middle| G'\right)$$

where G' is some column matrix of inhomogeneous functions.

Our goal is to solve for $S_1(n), \dots, S_m(n)$ in terms of the $A_i(n-j)$ and inhomogeneous terms. To accomplish this we attempt to use the four row operations to put the matrix into the form $\left(0\ 0 \cdots 0\ I \middle| * * \cdots * \middle| *\right)$ where the matrices marked by $*$ need not satisfy any conditions. At this point, each of $S_1(n), \dots, S_m(n)$ can be fully solved via back-substitution. The following example illustrates how this works in tandem with Lemma 1.

Example 2. In the context of Example 1, we will demonstrate how this can be used to evaluate the sums we desire. Begin by writing the mutual recurrence as an augmented matrix,

$$\left(0 \middle| I\ -M_1\ -M_2 \middle| G\right) = \left(\begin{array}{cc|cccccc|c} 0 & 0 & 1 & 0 & -2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & -2 & 0 \end{array}\right).$$

Apply the summation operator to both rows:

$$\left(\begin{array}{cc|cccccc|c} -1 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & -2A(1) - B(1) \\ -1 & -1 & 1 & 2 & 0 & 2 & 0 & 0 & -A(1) - 2B(1) - 2B(0) \end{array}\right)$$

Negate row 1 and add it to row 2:

$$\left(\begin{array}{cc|cccccc|c} -1 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & -2A(1) - B(1) \\ 0 & 0 & -1 & 1 & 0 & 2 & 0 & 0 & A(1) - B(1) - 2B(0) \end{array}\right)$$

Sum row 2 again:

$$\left(\begin{array}{cc|cccccc|c} -1 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & -2A(1) - B(1) \\ -1 & 3 & 0 & -2 & 0 & 0 & 0 & 0 & (A(1) - B(1) - 2B(0))(n-1) + 2B(1) \end{array}\right)$$

Negate row 1 and add it to row 2:

$$\left(\begin{array}{cc|cccccc|c} 1 & 1 & -2 & -1 & 0 & 0 & 0 & 0 & 2A(1) + B(1) \\ 0 & 4 & -2 & -3 & 0 & 0 & 0 & 0 & (A(1) - B(1) - 2B(0))(n-1) + 3B(1) + 2A(1) \end{array}\right)$$

Divide row two by 4 and subtract it from row 1:

$$\left(\begin{array}{cc|cccccc|c} 1 & 0 & -\frac{3}{2} & -\frac{1}{4} & 0 & 0 & 0 & 0 & -\frac{1}{4}(A(1) - B(1) - 2B(0))(n-1) + \frac{3}{2}A(1) + \frac{1}{4}B(1) \\ 0 & 1 & -\frac{1}{2} & -\frac{3}{4} & 0 & 0 & 0 & 0 & \frac{1}{4}(A(1) - B(1) - 2B(0))(n-1) + \frac{3}{4}B(1) + \frac{1}{2}A(1) \end{array}\right)$$

For completeness, observe that $\sum_{i=0}^{n} A(i) = S_A(n) + A(1) + A(0)$ and $\sum_{i=0}^{n} B(i) = S_B(n) + B(1) + B(0)$ and

$$\sum_{i=0}^{n} A(i) = \frac{3}{2}A(n) + \frac{1}{4}B(n) + \frac{1}{4}(A(1) - B(1) - 2B(0))n - \frac{3}{4}A(1) + \frac{1}{2}B(0) + A(0)$$
$$\sum_{i=0}^{n} B(i) = \frac{1}{2}A(n) + \frac{3}{4}B(n) - \frac{1}{4}(A(1) - B(1) - 2B(0))n - \frac{1}{4}A(1) + \frac{1}{2}B(0).$$

Now we are ready to state the main procedure which is similar to the approach of this example.

6 The Summation Procedure

Procedure 1 *Given a mutual recurrence and its corresponding augmented matrix, take its augmented matrix*

$$U = \left(0 \middle| I \; -M_1 \; -M_2 \; \cdots \; -M_d \middle| G \right).$$

For each $t \geq 0$ do the following starting with U: (1) Augment the matrix with $t+1$ blocks of $m \times m$ zero matrices on the left-hand side. (2) Duplicate each row of the matrix t times. The matrix has dimensions $(t+1)m \times (t+d+3)m$ and looks like:

$$\left(\begin{array}{ccc|cccccc|c} 0 & \cdots & 0 & I & -M_1 & -M_2 & \cdots & -M_d & G \\ 0 & \cdots & 0 & I & -M_1 & -M_2 & \cdots & -M_d & G \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & I & -M_1 & -M_2 & \cdots & -M_d & G \end{array} \right).$$

Number the block rows from top to bottom starting at 1. (3) Apply S to the $(t+2-i)^{th}$ block row i times for $1 \leq i \leq t+1$. (4) If placing this matrix in row-reduced echelon form results in some submatrix of the form

$$\left(0 \; 0 \cdots 0 \; I \middle| * \; * \cdots * \middle| * \right)$$

then stop. Back-substitute to solve for each $S_i(n)$ when $i \in \{1, \ldots, m\}$. Otherwise increment t by one and continue.

In the following we prove that this procedure terminates when $m = 1$ or when $d = 1$. By construction, if it terminates we are guaranteed that the solution is correct. That the procedure always terminates is left as a conjecture.

7 Analysis

Given integers m, d and matrices $M_1, \ldots, M_d \in M_m(K)$, define $M_0 = -I_m$ and

$$f(i) = \sum_{k_1=i}^{d} \sum_{k_2=k_1}^{d} \cdots \sum_{k_{i+1}=k_i}^{d} M_{k_{i+1}}.$$

On some occasions it is useful to consider a more general function of two variables,

$$f(i,j) = \sum_{k_1=j}^{d} \sum_{k_2=k_1}^{d} \cdots \sum_{k_{i+1}=k_i}^{d} \alpha_{k_i+1}.$$

For convenience when $i \geq 0$ we also define $f(-1, i) = M_i$. Notice that $f(i,i) = f(i)$ for all nonnegative integers i. We also have the identity

$$f(i,j) = \sum_{k=j}^{d} f(i-1,k) \tag{3}$$

which is easily checked from the definitions given. Typically m, d and $M_1, \ldots, M_d$ correspond to a particular mutual recurrence that should be clear from the context.

Lemma 2. *Procedure 1 terminates if the following statement holds: Given positive integers m, d, a field K and $M_1, \ldots, M_n \in M_m(K)$, $M_0 = -I_m$, there exists some $t \in \mathbb{N}$ such that the matrix*

$$\begin{pmatrix} f(0) & f(1) & f(2) & \cdots & f(t) \\ 0 & f(0) & f(1) & \cdots & f(t-1) \\ 0 & 0 & f(0) & \cdots & f(t-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & f(0) \end{pmatrix}$$

contains the vectors $(0,0,\ldots,0,\underbrace{1,0,\ldots,0}_{m}), (0,0,\ldots,0,\underbrace{0,1,\ldots,0}_{m}), \cdots,$ $(0,0,\ldots,0,\underbrace{0,0,\ldots,1}_{m})$ *in its rowspace. Note that each entry in this $tm \times tm$ matrix is an $m \times m$ block.*

Proof. The matrix presented above is the matrix derived from performing the four steps listed in the procedure for a particular value of t. If for some t the rowspace of this matrix contains these m standard basis vectors, then the $m \times m$ identity matrix can be formed as a submatrix from a linear combination of the rows, and hence the procedure terminates.

The only thing left to check is that performing the steps of the procedure results in a matrix in the above form. After completing step 2 in the procedure, the matrix has tm rows and is

$$\left(\begin{array}{cccc|ccccc|c} 0 & 0 & \cdots & 0 & I & -f(-1,1) & -f(-1,2) & \cdots & -f(-1,d) & G \\ 0 & 0 & \cdots & 0 & I & -f(-1,1) & -f(-1,2) & \cdots & -f(-1,d) & G \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & I & -f(-1,1) & -f(-1,2) & \cdots & -f(-1,d) & G \end{array}\right).$$

We claim that applying the summation operator $j+1$ times to any row will result in the row appearing as

$$\left(0 \cdots 0 \; f(0) \; f(1) \cdots f(j) \middle| f(j,j+1) \; f(j,j+2) \cdots f(j,d) \; 0 \cdots 0 \middle| G' \right).$$

Using Lemma 1 this is easy to verify inductively. Assuming that after $j+1$ summations the row takes the above form, summing once more obtains the form

$$\left(0 \cdots 0\; f(0) \cdots f(j)\; \sum_{k=1}^{d} f(j,j+k) \middle| -\sum_{k=2}^{d} f(j,j+k) \cdots -\sum_{k=d}^{d} f(j,j+k)\; 0 \cdots 0 \middle| G' \right).$$

From the identity (3) it follows that this row equals

$$\left(0 \cdots 0\; f(0) \cdots f(j)\; f(j+1) \middle| -f(j+1,j+2) \cdots -f(j+1,d)\; 0 \cdots 0 \middle| G' \right).$$

Therefore, after completing all of step 4, the matrix appears as

$$\left(\begin{array}{ccccc|cccc|c} f(0) & f(1) & f(2) & \cdots & f(t) & * & * & \cdots & * & * \\ 0 & f(0) & f(1) & \cdots & f(t-1) & * & * & \cdots & * & * \\ 0 & 0 & f(0) & \cdots & f(t-2) & * & * & \cdots & * & * \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & f(0) & * & * & \cdots & * & * \end{array}\right).$$

For now we are only concerned with the left-hand portion of this matrix.

If this matrix contains the necessary basis vectors in its rowspace, then the identity matrix must exist as a submatrix of the left-hand block. If no such t exists then the procedure never terminates. □

Corollary 1. *If $f(0) = M_1 + M_2 + \cdots + M_d - I$ is nonsingular, then the procedure terminates when $t = 1$.*

Lemma 3. *Let $p(x) = \alpha_d x^d + \cdots + \alpha_1 x + \alpha_0$ with $\alpha_0, \ldots, \alpha_d \in K$. Then $(x-1)^t$ divides $p(x)$ if and only if $f(0) = f(1) = \cdots = f(t-1) = 0$.*

Proof. We use the function f taking $M_i = \alpha_i$. By induction we show that $\frac{1}{(x-1)^t}\sum_{i=0}^{d} f(-1,i)x^i = \sum_{i=t}^{d} f(t-1,i)x^{i-t} + \sum_{i=1}^{t} \frac{f(t-i)}{(x-1)^i}$ for $t \geq 0$. When $t = 0$ this is just the polynomial we start with. The key to the inductive step is the observation that for a generic polynomial $g(x) = b_n x^n + \cdots + b_1 x + b_0$ the quotient $g(x)/(x-1)$ can be computed by summing the coefficients, as is done in the process of "synthetic division". Specifically, $g(x)/(x-1) = b_n x^{n-1} + (b_n + b_{n-1})x^{n-2} + \cdots + (b_n + \cdots + b_1) + (b_n + \cdots + b_1 + b_0)/(x-1)$. Using this technique, divide each side of the inductive goal by $(x-1)$. $1/(x-1)^{t+1}\sum_{i=0}^{d} f(-1,i)x^i = \sum_{i=t+1}^{d}\sum_{j=i}^{d} f(t-1,j)x^{i-(t+1)} + \sum_{j=t}^{d} f(t-1,j)/(x-1) + \sum_{i=1}^{t}\frac{f(t-i)}{(x-1)^{i+1}} = \sum_{i=t+1}^{d} f(t,i)x^{i-(t+1)} + f(t)/(x-1) + \sum_{i=2}^{t+1}\frac{f(t+1-i)}{(x-1)^i} = \sum_{i=t+1}^{d} f(t,i)x^{i-(t+1)} + \sum_{i=1}^{t+1}\frac{f(t+1-i)}{(x-1)^i}$. This completes the induction. Notice that now $(x-1)^t$ divides $p(x)$ evenly if and only if $f(0) = f(1) = \cdots = f(t-1)$. □

Theorem 2. *The procedure terminates when $m = 1$. Moreover, if $(x-1)^p || M_d x^d + M_{d-1}x^{d-1} + \cdots + M_1 x + M_0$, then the procedure terminates when $t = p$.* [5]

[5] When $m = 1$, each $M_i \in K$

Proof. For some nonnegative integer p we have that $(x-1)^p || M_d x^d + M_{d-1}x^{d-1} + \cdots + M_1 x + M_0$. By lemma 3 we know that $f(0) = f(1) = \cdots = f(p-1) = 0$, and that $f(p) \neq 0$. By lemma 2 we know this implies the procedure terminates since all the entries in the matrix are zero except the top-right entry. □

Theorem 3. *Procedure 1 terminates with the correct answer when $d = 1$.*

Proof. The approach is to take the mutual recurrence of order 1 and explicitly show that the condition in lemma 2 is satisfied. Let $A = M_1$. The goal is to show that for some $t \geq 0$ the following matrix of dimension $(t+1)m \times (t+1)m$

$$Z = \begin{pmatrix} A-I & -A & 0 & 0 & \cdots & 0 \\ 0 & A-I & -A & 0 & \cdots & 0 \\ 0 & 0 & A-I & -A & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & A-I \end{pmatrix}$$

can be row-reduced into a matrix containing I as a submatrix. The easy case is when A is nilpotent. Here, $|A - I| \neq 0$, so row reducing the matrix with $t = 0$ yields the identity matrix. For the remainder of the cases we will first discuss some facts about A and polynomials involving A, and then proceed to the row-reduction.

Let $\mu_A(x)$ be the minimal polynomial of A, that is the polynomial of least degree such that $\mu_A(A) = 0$. We regard $\mu_A \in K[x]$ and can write $\mu_A(x) = \mu_k x^k + \mu_{k-1}x^{k-1} + \cdots \mu_1 x + \mu_0$ for some $\mu_i \in K$, but define for any $X \in M_m(K)$, $\mu(X) = \mu_k X^k + \cdots + \mu_1 X + \mu_0 I$. Let t be the greatest nonnegative integer such that $(x-1)^t$ divides $\mu(x)$ and write $\mu(x) = (x-1)^t q(x)$ for some $q(x) \in K[x]$. We define $q(X)$ for a matrix X the same way we did for μ_A. Notice that this allows us to perform the division algorithm with polynomials over matrices, in the sense that for any polynomial $g \in K[x]$ there exist polynomials $s, r \in K[x]$ such that $q(X) = g(X)s(X) + r(X)$ where the degree of r is less than the degree of g. This works because sums and products of a single matrix X freely commute with each other.

Label the block rows of the matrix from top to bottom starting at 1. For $1 \leq i \leq t$ multiply the row by $(A-I)^{t-i}A^{i-1}q(A)$. In the first block row the leftmost block entry becomes $\mu(A) = (A-I)^t q(A) = 0$. For all subsequent rows $i \leq t$, the leftmost non-zero block entry equals the entry above it; namely the leftmost non-zero block in row i is $(A-I)^{t-i+1}A^{i-1}q(A)$. In row t, the rightmost block entry is $-A^t q(A)$ and the entry below it is still $A - I$. At this point row reducing the matrix only leaves all the entries of the matrix nonzero except for these two, so

$$Z \sim \begin{pmatrix} 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & A^t q(A) \\ 0 & \cdots & 0 & A-I \end{pmatrix}.$$

Using the division algorithm, write $q(A) = (A-I)s(A)+\alpha A^p$ for some $\alpha \in K$, $p \in \mathbb{N}$ and $s \in K[x]$. Notice that $\alpha \neq 0$, for if it were not then $(x-1)|q(x)$ which would contradict its definition. Multiply the bottom row by $s(A)A^t$ and subtract from the row above. Divide the above row by α. This leaves the matrix

$$Z \sim \begin{pmatrix} 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & A^{p+t} \\ 0 & \cdots & 0 & A-I \end{pmatrix}.$$

Finally multiply the bottom row of the matrix by $(A^{p+t-1}+A^{p+t-2}+\cdots+A^1)$ and subtract from the top row. This leaves the entry $A^{p+t}-(A^{p+t-1}+A^{p+t-2}+\cdots+A^1)(A-I) = A$ in that row. Negate the bottom row and add the other row to derive I in the bottom right corner of the matrix. □

Corollary 2. *For $m = 1$ and $d = 1$ the runtime is bounded by $O(m^3(t+d)^3)$.*

For both cases, we can use the above theorems to provide an explicit value for $t \leq m$. The two longest steps in the procedure are applying the summation operator and computing the row reduction. Via lemma 1 it takes $O(m^2(t+d^2))$ time to sum one block row of the matrix in addition to the time it takes to compute the sum of the inhomogeneous term. This must be performed only m times if the results from row i are used to compute row $i+1$. Therefore step 2 takes time $O(m^3(t+d^2))$. A naive implementation of step 3 using Gauss-Jordan elimination will require $O(m^3(t+d)^3)$ time. Thus the third step dominates and the asymptotic run time of the entire procedure is $O(m^3(t+d)^3)$ in addition to the time required to sum inhomogeneous parts. □

Notice that for both the $m = 1$ case and the $d = 1$ case there exists a polynomial $p(x)$ such that if q is the greatest integer where $(x-1)^q|p(x)$ then t is bounded above by q. This is reminiscent of Ranvenscroft and Lamagna's result in [8] where this power of t is used to derive a "summing factor" to sum linear recurrences. This suggests a more general phenomena of a minimal polynomial for a mutual recurrence that can be subject of further work. It also suggests that understanding this minimal polynomial will yield more results about the efficiency and termination of the procedure.

8 Conclusion

The procedure presented in this paper provides a way to compute symbolic sums for both homogeneous and inhomogeneous mutual recurrences. It is an improvement over Ravenscroft and Lamagna's [8] because it generalizes to mutual recurrences and provably works when $m = 1, d = 1$ or $|\sum_{i=1}^{d} M_i - I| = 0$. It also extends the current work on P-finite and C-finite summation. Given a sequence a_n that is not P-finite but can be summed via other methods, procedure 1 will sum mutual recurrences where the a_i appear as inhomogeneous terms.

We believe that generalizing the proof of theorems 2 and 3 will be difficult for two reasons. First, there is not a well-known theory of a characteristic polynomial or minimal polynomial for a mutual recurrence that satisfies our needs, or a theorem like Cayley-Hamilton which extends to multiple matrices. Secondly, the ring generated by two $m \times m$ matrices is not necessarily commutative. When there is only one matrix this ring commutes, which allowed many of the steps in the proof of theorem 3.

References

1. Abramov, S.A., van Hoeij, M.: Desingularization of linear difference operators with polynomial coefficients. In: Dooley, S. (ed.) International Symposium on Symbolic and Algebraic Computation. pp. 269–275. ACM, New York, NY, USA (1999)
2. Chyzak, F.: An extension of Zeilberger's fast algorithm to general holonomic functions. Discrete Mathematics 217(1-3), 115–134 (April 2000)
3. Gosper, Jr., R.W.: Decision procedure for indefinite hypergeometric summation. In: Proceedings of the National Aacademy of Sciences USA. vol. 75, pp. 40–42 (January 1978)
4. Greene, C., Wilf, H.S.: Closed form summation of C-finite sequences. Transactions of the American Mathematical Society 359, 1161–1189 (2007)
5. Karr, M.: Summation in finite terms. Journal of the ACM 28(2), 305–350 (April 1981)
6. Kauers, M., Zimmermann, B.: Computing the algebraic relations of C-finite sequences and multisequences. Journal of Symbolic Computation 43(11), 787–803 (2008)
7. Moenck, R.: On computing closed forms for summation. In: Proceedings of the MACSYMA User's Conference. pp. 225–236 (1997)
8. Ravenscroft, Jr., R.A., Lamagna, E.A.: Summation of linear recurrence sequences. In: Milestones in Computer Algebra. pp. 125–132 (2008)
9. Schneider, C.: A new sigma approach to multi-summation. Advances in Applied Mathematics 34(4), 740–767 (May 2005)
10. Zeilberger, D.: A fast algorithm for proving terminating hypergeometric series identities. Discrete Mathematics 80(2), 207–211 (March 1990)