A SOLUTION TO CERTAIN POLYNOMIAL EQUATIONS WITH APPLICATIONS TO NONLINEAR FITTING

CHRIS CONNELL

ABSTRACT. We present a combinatorial method for solving a certain system of polynomial equations of Vandermonde type in $2N$ variables by reducing it to the problem of solving two special linear systems of size N and rooting a single univariate polynomial of degree N. Over $\mathbb{C}$, all solutions can be found with fixed precision using, up to polylogarithmic factors, $O(N^2)$ bitwise operations in the worst case. Furthermore, if the data is well conditioned, then this can be reduced to $O(N)$ bit operations, up to polylogarithmic factors. As an application, we show how this can be used to fit data to a complex exponential sum with N terms in the same, nearly optimal, time.

1. INTRODUCTION

In this paper we present a combinatorial method for solving the following system of polynomial equations in unknowns A_i and X_i where the elements F_i lie in a (commutative) integral domain $\mathbb{D}$.

$$(1) \quad \begin{cases} F_1 &= A_1 & + & A_2 & + & \cdots & + & A_N \\ F_2 &= A_1 X_1 & + & A_2 X_2 & + & \cdots & + & A_N X_N \\ \vdots & \vdots & & \vdots & & \ddots & & \vdots \\ F_{2N} &= A_1 X_1^{2N-1} & + & A_2 X_2^{2N-1} & + & \cdots & + & A_N X_N^{2N-1}. \end{cases}$$

Certainly, any polynomial system which is regular can be reduced to solving univariate polynomials by producing resultants via Sylvester's elimination method or other algorithms which produce a Gröbner basis. Simply applying the Weak Nullstellensatz gives a criteria for when solutions exist in terms of the associated ideal. The problem is that such general methods do not usually provide the most efficient procedures for accomplishing these tasks. For instance, one can easily check that for the simple case of $F_i = 1 + 1/i \in \mathbb{Q}$ in the above system, at least for $N \leq 5$, one must still root N polynomials with degrees $1, \ldots, N$ even once the Gröbner basis has been computed. This holds regardless of elimination order.

The reduction we present provides an explicit and computationally reasonable method for determining when solutions of this system exist and what they are. We will show that when solutions exist, they lie in an extension (of degree at most N) of the fraction field, $\mathbb{K}$, of the integral domain $\mathbb{D}$, although we could (almost) as easily look for solutions in any integral domain extending $\mathbb{D}$. We summarize the solution as follows.

Theorem I. *The problem of solving system* (1) *is equivalent to (in order) solving a certain linear Hankel system, factoring a univariate degree N polynomial, and solving a linear Vandermonde system. Moreover this reduction is given explicitly.*

More generally, for $i = 1, \ldots, N$ choose elements q_i and r_i from the set of exponents of $\mathbb{D}$ (i.e., those elements $y \in \mathbb{D}$ such that x^y is well defined and exists in $\mathbb{D}$ for all $x \in \mathbb{D}$). Then for any positive integer M we may consider the over/under-determined system of $\mathbb{D}$-rational equations below.

(2)
$$
\begin{cases}
F_1 & = & A_1 X_1^{r_1} & + & A_2 X_2^{r_2} & + & \cdots & + & A_N X_N^{r_N} \\
F_2 & = & A_1 X_1^{q_1 + r_1} & + & A_2 X_2^{q_2 + r_2} & + & \cdots & + & A_N X_N^{q_N + r_N} \\
\vdots & & \vdots & & \vdots & & \ddots & & \vdots \\
F_M & = & A_1 X_1^{q_1(M-1) + r_1} & + & A_2 X_2^{q_2(M-1) + r_2} & + & \cdots & + & A_N X_N^{q_N(M-1) + r_N}.
\end{cases}
$$

After proving the theorem for system (1) in Section 2 we will explain how to solve system (2) in like manner. One item of importance of these particular systems of equations arises from their connection to the problem of fitting a sum of real or complex exponentials. In fact solutions to the above systems result in an efficient method for finding the coefficients β_i and exponents α_i which provide a "best fit" approximation of a function $F(t)$, or discrete samples $F = \{(t_1, f_1), \ldots, (t_n, f_n)\}$, by a function of the form

$$
\widetilde{F}(t) = \sum_{i=1}^{N} \beta_i e^{\alpha_i t},
$$

for any choice of N. Here the β_i and the α_i depend only on the A_i and X_i in a relatively simple way. This will be explored in Section 3.

2. Solving the equations

For the first step in solving equations (1), notice that these equations are linearly dependent on the A_i. We can then easily eliminate the A_i from the first N equations. To do this, use the the first N equations of (1) to solve for the A_i,

(3)
$$
\begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_N \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ X_1 & X_2 & \cdots & X_N \\ \vdots & \vdots & \ddots & \vdots \\ X_1^{N-1} & X_2^{N-1} & \cdots & X_N^{N-1} \end{bmatrix} \cdot \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_N \end{bmatrix},
$$

and then substitute these into the last N equations of system (1). Recognizing the matrix above as the transpose of the Vandermonde matrix, we can formally compute the entries.

First we establish some notation. Let Alt be the elementary alternating polynomial in the variables $X_1, \ldots, X_N$. Explicitly,

$$
\mathrm{Alt} = \prod_{k=1}^{N} \prod_{j>k} (X_k - X_j).
$$

Similarly, let Alt_i be the elementary alternating polynomial in the variables $X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_N$. Let Sym_j be the jth elementary symmetric polynomial in the variables $X_1, \ldots, X_N$. Explicitly, for $1 \leq j \leq N$,

$$(4) \qquad \mathrm{Sym}_j = \sum_{l_1=1}^{N} \sum_{l_2=l_1+1}^{N} \cdots \sum_{l_j=l_{j-1}+1}^{N} \prod_{k=1}^{j} X_{l_k},$$

and $\mathrm{Sym}_0 = 1$.

Lastly, let $\mathrm{Sym}_{i,j}$ be the jth elementary symmetric polynomial in the variables $X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_N$. We will outline a proof to the following well-known lemma.

Lemma 2.1. *With the above notation, whenever $X_i \neq X_j$ for $j \neq i$, the A_i satisfy*

$$(5) \qquad A_i = \frac{(-1)^{N+i+1} \, \mathrm{Alt}_i}{\mathrm{Alt}} \sum_{j=1}^{N} (-1)^j \, \mathrm{Sym}_{i,N-j} \, F_j.$$

Proof. Consider the entries of the transposed Vandermonde matrix as formal monomials with coefficients in $\mathbb{Z}$. The minor $M_{i,j}$ of the transposed Vandermonde matrix corresponding to position (i, j) is an alternating polynomial in all of the variables except X_j. Every alternating polynomial F (except in characteristic 2) may be formally expressed as a product, $F = \mathrm{Alt}\, S$, of the elementary alternating polynomial Alt and some symmetric polynomial S. Hence $M_{i,j} = \mathrm{Alt}_j\, S_{i,j}$ where for each i and j, $S_{i,j}$ is symmetric in all of the variables except X_j. For any row i (resp. column j), the alternating sum of the $M_{i,j}$ in j (resp i) is Alt. One easily checks that this relationship is satisfied when $S_{i,j} = \mathrm{Sym}_{j,N-i}$ (uniquely for coefficients in $\mathbb{Z}$).

Recall that a matrix with entries in $\mathbb{D}$ is invertible in $\mathbb{D}$ if and only if its determinant is a unit in $\mathbb{D}$. If that happens, then the inverse matrix is the transpose of the matrix of minors with alternating signs divided by the determinant. It follows that the (i, j) entry of the inverse of the Vandemonde matrix is

$$\frac{(-1)^{N+i+1+j} \, \mathrm{Alt}_i}{\mathrm{Alt}} \, \mathrm{Sym}_{i,N-j}.$$

So for any fixed $X_1, \ldots, X_N$ belonging to a commutative domain $\mathbb{I}$, if $X_i \neq X_j$ for any two indices $i \neq j$, then $\mathrm{Alt} \neq 0$ and the Vandermonde matrix is invertible in $\mathbb{I}[\frac{1}{\mathrm{Alt}}]$. Thus the formula for the A_i is valid in this domain and hence in the fraction field for $\mathbb{I}$ as well. $\qquad \square$

Remark 2.2. One approach to finding solutions where some of the X_i, say $X_{N-r}, \ldots, X_N$, coincide is to make this substitution to begin with. In this case, system (1) reduces to the overdetermined system (2) with $r_i = 0$, $q_i = 1$, $M = N$ and N replaced by $N - r$. When solutions to this latter system exist, say in a field $\mathbf{k}/\mathbb{K}$, there is an r dimensional space (over $\mathbf{k}/\mathbb{K}$) of solutions to the $A_{N-r}, \ldots, A_N$ in system (1).

Substituting the A_i back into equations (1) then yields relations for the F_{k+N} in terms of symmetric polynomials in the X_i and the F_i. More specifically, if for each j and k, $C_{k,j}$ denotes the coefficients for F_j in the expression for F_{k+N}, then

the second half of the system (1) becomes

$$\text{(6)} \quad \begin{cases} F_{N+1} & = & C_{1,1}F_1 & + & C_{1,2}F_2 & + & \cdots & + & C_{1,N}F_N \\ F_{N+2} & = & C_{2,1}F_1 & + & C_{2,2}F_2 & + & \cdots & + & C_{2,N}F_N \\ \vdots & & \vdots & & \vdots & & \ddots & & \vdots \\ F_{2N} & = & C_{N,1}F_1 & + & C_{N,2}F_2 & + & \cdots & + & C_{N,N}F_N, \end{cases}$$

where we have

$$C_{k,j} = \frac{(-1)^{N+j+1}}{\text{Alt}} \sum_{i=1}^{N} (-1)^i \, \text{Alt}_i \, \text{Sym}_{i,N-j} \, X_i^{k+N-1}.$$

Now we will show that the $C_{k,j}$ satisfy a recursion relation.

Proposition 2.3. *The $C_{k,j}$ satisfy*

$$C_{k,j} = (-1)^{N+j} \, \text{Sym}_{N-j+1} \, C_{k-1,N} + C_{k-1,j-1},$$

with

$$C_{0,N} = 1, \quad C_{0,j} = 0 \quad \text{for} \quad 0 \le j < N, \quad \text{and} \quad C_{k,0} = 0 \quad \text{for all} \quad k \ge 0.$$

Proof. To verify this, we simply compute

$$C_{k,j} - (-1)^{N+j} \, \text{Sym}_{N-j+1} \, C_{k-1,N} - C_{k-1,j-1}$$

$$= \frac{(-1)^{N+j+1}}{\text{Alt}} \sum_{i=1}^{N} (-1)^i \, \text{Alt}_i \, \text{Sym}_{i,N-j} \, X_i^{k+N-1}$$

$$- (-1)^{N+j} \, \text{Sym}_{N-j+1} \frac{-1}{\text{Alt}} \sum_{i=1}^{N} (-1)^i \, \text{Alt}_i \, \text{Sym}_{i,0} \, X_i^{k+N-2}$$

$$- \frac{(-1)^{N+j}}{\text{Alt}} \sum_{i=1}^{N} (-1)^i \, \text{Alt}_i \, \text{Sym}_{i,N-j+1} \, X_i^{k+N-2}$$

$$= \frac{(-1)^{N+j+1}}{\text{Alt}} \sum_{i=1}^{N} \left(\text{Sym}_{i,N-j} \, X_i - \text{Sym}_{N-j+1} + \text{Sym}_{i,N-j+1} \right)(-1)^i X_i^{k+N-2} \text{Alt}_i$$

$$= 0$$

since the elementary symmetric polynomials satisfy

$$\text{(7)} \qquad \text{Sym}_{N-j+1} = \text{Sym}_{i,N-j} \, X_i + \text{Sym}_{i,N-j+1},$$

where we assign $\text{Sym}_{i,N} = 0$.

The base cases of the recursion may be verified as follows. To show $C_{0,N} = 1$, we verify by switching variables that the numerator of

$$C_{0,N} = \frac{-1}{\text{Alt}} \sum_{i=1}^{N} (-1)^i \, \text{Alt}_i \, X_i^{N-1}$$

is an alternating polynomial of total degree $\binom{N}{2}$ in the variables $X_1, \ldots, X_N$ and hence by the principal theorem of alternating polynomials it must be a constant multiple of Alt. To show that the constant is 1, we simply substitute the values $X_i = i$ to obtain

$$C_{0,N} = -\sum_{i=1}^{N} (-1)^i \frac{(-1)^{N-1} i^{N-1}}{i!(N-i)!} = \frac{(-1)^N}{N!} \sum_{i=1}^{N} \binom{N}{i} (-1)^i i^N = 1.$$

The last equality follows from recognizing the second to last expression as the Sterling number σ_N^N.

Similarly, for the $C_{0,j}$ with $j < N$ we expand recursively on j, using the relations (7), as follows

$$
\begin{aligned}
C_{0,j} &= \frac{(-1)^{N+j+1}}{\text{Alt}} \sum_{i=1}^{N} (-1)^i \, \text{Sym}_{i,N-j} \, \text{Alt}_i \, X_i^{N-1} \\
&= \frac{(-1)^{N+j+1}}{\text{Alt}} \left(\text{Sym}_{N-j+1} \sum_{i=1}^{N} (-1)^i \, \text{Alt}_i \, X_i^{N-2} \right. \\
&\qquad\qquad\qquad \left. - \sum_{i=1}^{N} (-1)^i \, \text{Sym}_{i,N-j+1} \, \text{Alt}_i \, X_i^{N-2} \right) \\
&\ \ \vdots \\
&= \frac{(-1)^{N+j+1}}{\text{Alt}} \left(\sum_{k=1}^{j} (-1)^{k+1} \, \text{Sym}_{N-j+k} \sum_{i=1}^{N} (-1)^i \, \text{Alt}_i \, X_i^{N-1-k} \right).
\end{aligned}
$$

However, we may observe by switching pairs of variables X_i and X_j for all $i \neq j$ that the sum

$$
\sum_{i=1}^{N} (-1)^i \, \text{Alt}_i \, X_i^{N-1-k}
$$

is an alternating polynomial. Formally considering the coefficients to lie in $\mathbb{Z}$, it must then be a product of a symmetric polynomial and Alt. However for $k \geq 1$, its degree is less than Alt so each sum is zero in $\mathbb{D}$.

Lastly, $C_{k,0} = 0$ since $\text{Sym}_{i,N} = 0$. $\qquad\qquad\qquad\qquad\qquad\qquad \square$

Using the above recurrence formula, the relationship between the X_i and F_i has been somewhat simplified. For instance, with $N = 3$, the system (6) becomes the following.

$$
\begin{bmatrix} F_4 \\ F_5 \\ F_6 \end{bmatrix} = \begin{bmatrix} \text{Sym}_3 & -\text{Sym}_2 & \text{Sym}_1 \\ \text{Sym}_1\,\text{Sym}_3 & \text{Sym}_3 - \text{Sym}_1\,\text{Sym}_2 & \text{Sym}_1^2 - \text{Sym}_2 \\ (\text{Sym}_1^2 - \text{Sym}_2)\,\text{Sym}_3 & \text{Sym}_2^2 - \text{Sym}_1^2\,\text{Sym}_2 + \text{Sym}_1\,\text{Sym}_3 & \text{Sym}_1^3 - 2\,\text{Sym}_1\,\text{Sym}_2 + \text{Sym}_3 \end{bmatrix} \cdot \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}
$$

The above proposition allows us to apply symmetric reduction to system (1) since the X_i variables only appear in symmetric polynomials. Next we will show that system (6) has a simpler form.

Proposition 2.4. *The system* (6) *may be expressed as the following linear system.*

$$
(8) \qquad \begin{bmatrix} F_{N+1} \\ F_{N+2} \\ \vdots \\ F_{2N} \end{bmatrix} = \begin{bmatrix} F_N & -F_{N-1} & \cdots & (-1)^{N+1} F_1 \\ F_{N+1} & -F_N & \cdots & (-1)^{N+1} F_2 \\ \vdots & \vdots & \ddots & \vdots \\ F_{2N-1} & -F_{2N-2} & \cdots & (-1)^{N+1} F_N \end{bmatrix} \cdot \begin{bmatrix} \text{Sym}_1 \\ \text{Sym}_2 \\ \vdots \\ \text{Sym}_N \end{bmatrix}
$$

Moreover, for any solution to system (1), *the elementary symmetric polynomials in the X_i satisfy the above linear relation.*

Proof. Repeatedly applying the recurrence relation from Proposition 2.3 on the kth equation of (6) for $k \geq 1$ we obtain

$$
\begin{aligned}
F_{N+k} &= \sum_{i=1}^{N} C_{k,i} F_i \\
&= (-1)^{N+1} F_1 \operatorname{Sym}_N C_{k-1,N} \\
&\quad + \sum_{i=2}^{N} F_i \left((-1)^{N+i} \operatorname{Sym}_{N-i+1} C_{k-1,N} + C_{k-1,i-1} \right) \\
&= C_{k-1,N} \underbrace{\sum_{i=1}^{N} (-1)^{N+i} F_i \operatorname{Sym}_{N-i+1}}_{F_{N+1}} + \sum_{i=2}^{N} F_i C_{k-1,i-1} \\
&= \sum_{i=2}^{N+1} F_i C_{k-1,i-1} \\
&= \sum_{i=1}^{N} F_{i+1} C_{k-1,i} \\
&= F_2 C_{k-1,1} + \sum_{i=2}^{N} F_{i+1} \left((-1)^{N+i} \operatorname{Sym}_{N-i+1} C_{k-2,N} + C_{k-2,i-1} \right) \\
&\vdots \quad \vdots \\
&= \sum_{i=1}^{N} F_{i+2} C_{k-2,i} \\
&\vdots \quad \vdots \\
&= \sum_{i=1}^{N} F_{i+k-1} C_{1,i} \\
&= \sum_{i=1}^{N} F_{i+k-1} (-1)^{N+i} \operatorname{Sym}_{N-i+1} \\
&= \sum_{i=1}^{N} F_{N-i+k} (-1)^{i+1} \operatorname{Sym}_i \, .
\end{aligned}
$$

Therefore these equations reduce to the N linear equations

$$
F_{N+k} = \sum_{i=1}^{N} F_{N-i+k} (-1)^{i+1} \operatorname{Sym}_i,
$$

as desired.

The above reductions also show that for any solution to system (1) the elementary symmetric polynomials in the X_i satisfy (8). This is immediate whenever the X_i are distinct. The coincidence of two or more of the X_i invalidates formula (5). Nevertheless, Proposition 2.3 implies, in particular, that the $C_{i,j}$ are polynomial in the X_i. It follows from continuity in the parameters that the elementary symmetric polynomials in the X_i still satisfy (8) in this case. $\qquad \square$

The system (8) may be easily solved for the Sym_i assuming that the system is nonsingular. Replacing Sym_i by $(-1)^{N-i} \operatorname{Sym}_{N-i+1}$ in system (8) and performing the same operations on the corresponding columns of the matrix yields the following

system.

$$
(9) \qquad
\begin{bmatrix} F_{N+1} \\ F_{N+2} \\ \vdots \\ F_{2N} \end{bmatrix}
=
\begin{bmatrix}
F_1 & F_2 & \cdots & F_N \\
F_2 & F_3 & \cdots & F_{N+1} \\
\vdots & \vdots & \ddots & \vdots \\
F_N & F_{N+1} & \cdots & F_{2N-1}
\end{bmatrix}
\cdot
\begin{bmatrix}
(-1)^{N-1}\operatorname{Sym}_N \\
(-1)^{N-2}\operatorname{Sym}_{N-1} \\
\vdots \\
\operatorname{Sym}_1
\end{bmatrix}
$$

This we recognize as a Hankel system which, like Vandermonde systems, can be solved much more efficiently than general systems as we will see in Section 2.2.

The original objective was to solve for the X_i in terms of the F_j. However, treating the Sym_i as known constants, the system of equations (4) in variables $X_1, \ldots, X_N$ has, for each solution, up to $N!$ redundant solutions corresponding to permutations of the X_i. We are only interested in obtaining all the solutions up to permutation. Hence we wish to reduce the degree of this symmetric system. The following lemma is classical for fields, but nevertheless we will present a short proof to highlight its applicability to commutative rings in general.

Lemma 2.5. *For elements S_i in any unital commutative ring R, consider the system of equations $S_i = \operatorname{Sym}_i$ for $i = 1, \ldots, N$, where Sym_i are given by (4). In any commutative ring R' containing R, a solution $X_1, \ldots, X_N$ to this system is also a complete set of roots in R' for the monic polynomial*

$$
(10) \qquad P(X) := X^N + \sum_{i=1}^{N} X^{N-i}(-1)^i \, S_i .
$$

Conversely every complete set of roots for the polynomial is also a solution to the system.

Proof. Recall that $\operatorname{Sym}_i = \sum_{1 \le l_1 < \cdots < l_i \le N} \prod_{m=1}^{i} X_{l_m}$. For any solution set $X_1, \ldots, X_N$ of the system $S_i = \operatorname{Sym}_i$ for $i = 1, \ldots, N$, substituting Sym_i for S_i into the equation for $P(X)$, we find

$$
P(X_k) = X_k^N + \sum_{i=1}^{N} (-1)^i X_k^{N-i} \operatorname{Sym}_i(X_1, \ldots, X_N).
$$

The $(-1)^N \operatorname{Sym}_N$ term cancels with the X_k term of $(-1)^{N-1} X_k^1 \operatorname{Sym}_{N-1}$ and the X_k^2 terms of this quantity cancel with the X_k^2 terms of $(-1)^{N-2} X_k^2 \operatorname{Sym}_{N-2}$, and so forth, until finally the remaining X_k^N term of $-X_k^{N-1} \operatorname{Sym}_1$ cancels X_k^N, showing that $P(X_k) = 0$.

Conversely, if $P(X) = \prod_{i=1}^{N} (X - X_i)$, then expanding this product shows that the coefficient $(-1)^k S_k$ of X^k is $(-1)^k \operatorname{Sym}_k$ as expected. In these verifications we only use commutativity of the ring operations. $\qquad\square$

Completing the proof of Theorem I. Now we have the prescription for solving the original system (1). Starting with known $F_i \in \mathbb{D}$, we use equation (8) to solve for the Sym_i which will be rational in the F_i whenever solutions exist. We then substitute $\operatorname{Sym}_i \in \mathbb{K}$ for the S_i in equation (10). For a complete factorization of the polynomial $P(X)$ we obtain the solution of the X_i in an extension field $\mathbf{k}$ of at most degree n over $\mathbb{K}$. Then we substitute these values into (3) and solve for the A_i which also live in $\mathbf{k}$ whenever solutions exist.

Conversely, given a solution X_i, A_i for $i = 1, \ldots, N$ of system (1), Proposition 2.4 states that the elementary symmetric polynomials in X_i form solve system (8)

and by Lemma 2.5 the X_i form a factorization of $P(X)$. It then follows that the A_i must satisfy (3). Therefore, if either of the two linear systems or the factorization problem fails to admit a solution, then system (1) does not admit solutions.   □

Solutions, when they exist, naturally live in the Galois extension $\mathbf{k}/\mathbb{K}$ which splits $P(X)$. The problem of looking further for solutions in $\mathbb{D} \subset \mathbf{k}/\mathbb{K}$ is then only as hard as the membership problem.

Since system (8) is essentially a Hankel system, much can be said about when solutions exist in terms of $F_1, \ldots, F_{2N}$ (see for instance [Gou97, ISL92, Las90]). Similarly, from equation (5) we see that solutions to the A_i exist whenever the X_i are distinct, or else whenever $X_i = X_j$ for two indices i, j, the values $F_1, \ldots, F_N$ satisfy $\sum_{l=1}^{N} (-1)^l \operatorname{Sym}_{k,N-l} F_l = 0$ for $k = i$ and $k = j$.

Let $\mathbf{k}$ be any field extending $\mathbb{K}$. The solution variety $V \subset \mathbf{k}^{2N}$ for system (1) is fibered in the following way. For each solution $\vec{s}$ in the affine subspace $S \subset \mathbf{k}^N$ of solutions to (8) we consider the (possibly empty) set $L_{\vec{s}}$ of all distinct (including permutations) complete factorizations of $P(X)$ with coefficients S_i taken from $\vec{s}$ in order. Each of these forms the coordinates for a point $\vec{x} \in \mathbf{k}^N$. For each $\vec{x} \in L_{\vec{s}}$ we have a (possibly empty) affine subspace $T_{\vec{x}} \subset \mathbf{k}^N$ of solutions to (3) for this factorization. Hence

$$V = \bigcup_{\vec{s} \in S} \bigcup_{\substack{\vec{x} \in L_{\vec{s}} \\ T_{\vec{x}} \neq \emptyset}} (\vec{x}, T_{\vec{x}}).$$

Since $T_{\vec{x}}$ is independent of the order of $\vec{x}$, the symmetric group S_N acts on V (not necessarily freely) by permuting the first N coordinates and the second N coordinates the same way. The quotient $\hat{V} = V/S_N$ is the reduced variety and we will say that system (1) is regular if $\hat{V}$ is exactly one point. Note that if system (1) is regular for any nontrivial field, then the system is regular for any field in which it admits a solution.

2.1. **The case of system** (2). We can relate the solutions of system (2) in a field $\mathbf{k}/\mathbb{K}$ to the solutions of the simplified system (2) where we set $q_i = 1$ and $r_i = 0$ for all $i = 1, \ldots, N$ as follows.

Consider a solution, A_i and X_i $(i = 1, \ldots, N)$, to the simplified system in a field $\mathbf{k}'/\mathbb{K}$. For each set of solutions x_i, $i = 1, \ldots, N$, in $\mathbf{k}/\mathbb{K}$ of the equations $X_i = x_i^{q_i}$, set $a_i = A_i x_i^{-r_i}$. Since r_i is an exponent, $a_i \in \mathbf{k}$. The quantities $X_i = x_i$ and $A_i = a_i$ then provide a solution to (2). Furthermore, all solutions in $\mathbf{k}$ arise in this way as $\mathbf{k}'$ is allowed to vary.

It remains to discuss the solution to system (1) in the case when the number of equations is $M \neq 2N$. Of course, when we have $M \leq N$, we could solve (2) for the A_i's in $\mathbb{K}$ for any choice of M distinct X_i's which in turn would be a solution to the original system. Therefore we may assume $M > N$.

Calculating $A_1, \ldots, A_N$ as before and substituting these into the second $M - N$ equations, we obtain coefficients $C_{k,j}$ as in (6) for $0 \leq k \leq M - N$ and $1 \leq j \leq N$. Proposition 2.3 still holds when $M > N$, and in fact it gives the formula for $C_{k,j}$ for all $k \geq 1$ and $1 \leq j \leq N$. Hence, we obtain the same system of equations,

$$F_{N+k} = \sum_{i=1}^{N} F_{N-i+k}(-1)^{i+1} \operatorname{Sym}_i,$$

for $k = 1, \ldots, M - N$.

Lastly in the under-determined case $M < 2N$, the following system

$$(11) \quad \begin{bmatrix} F_{N+1} \\ F_{N+2} \\ \vdots \\ F_M \end{bmatrix} = \begin{bmatrix} F_N & -F_{N-1} & \cdots & (-1)^{N+1}F_1 \\ F_{N+1} & -F_N & \cdots & (-1)^{N+1}F_2 \\ \vdots & \vdots & \ddots & \vdots \\ F_{M-1} & -F_{M-2} & \cdots & (-1)^{N+1}F_{M-N} \end{bmatrix} \cdot \begin{bmatrix} \mathrm{Sym}_1 \\ \mathrm{Sym}_2 \\ \vdots \\ \mathrm{Sym}_N \end{bmatrix}$$

can be solved (generically) for a $2N - M$ dimensional space of solutions which correspond to completing this system with variables $F_{M+1}, \ldots, F_{2N}$ to transform it into the Hankel system (9).

In the over-determined case $M > 2N$, system (11) may be restricted to any subsystem of size N each corresponding to rows n through $N + n - 1$ for any $n = 1, \ldots, M - 2N$. This way the system will remain essentially Hankel after relabelling. Once any of these systems is solved one may easily check the remaining equations for consistency.

In either case, for each solution to system (11) we obtain the X_i and A_i as before by solving (10) and (3), respectively.

2.2. Time complexity.

Now we consider the efficiency for the entire procedure on a random access machine. There are several different cases to consider. First we consider exact solutions in fields. Of course, in general, not much can be said since the fields themselves can be arbitrarily complicated for a computer to handle. That is, one arithmetic operation can take arbitrarily long, such as in an extension field over $\mathbb{Q}$ with an arbitrary large degree. The factorization problem is also very delicate in arbitrary fields. However, some of the most important cases are the finite and p-adic cases, F_p and $\mathbb{Q}_p$. Over these fields, both the Hankel system (9) and the Vandermonde system (3) can be solved with $O\left(N \log^2(N) + N \log p\right)$ arithmetic operations ([BDCMM01]). Factoring the polynomial $P(X)$ in $\mathbb{F}_p$ takes no worse than $O\left(N^2 \log p\right)$ operations (see [vzGP01, Sho95] for subquadratic algorithms when q is relatively small compared to n). For not much worse, one can work in extensions of $\mathbb{Q}_p$ (see [Pau01]).

For the case of approximate solutions in $\mathbb{C}$, we wish to find a solution to the system with accuracy 2^{-b}. We also want to consider an actual computer implementation, so we consider the time in bit operations instead of arithmetic operations. In the case of factorization, the latter usually masks an extra order of N bit operations due to the necessity of high precision.

Both the Hankel system (9) and the the Vandermonde system (3) can be solved in $\mathbb{C}$ with $O\left(bN \log^2(N)\right)$ bit operations with storage $O(N)$ (see [Pan92]). In [Pan02] it was recently shown that the problem of approximate factorization in $\mathbb{C}$ requires

$$O\left((bN^2 \log bN \log^2 N \log \log bN)(\log^2 N + \log bN)\right)$$

bit operations for accuracy 2^{-b}. On the other hand, this can be reduced, up to polylogarithmic factors, down to $O(bN)$ bit operations assuming that the zeros of the polynomial are not too tightly clustered ([Pan02]). In particular, the entire time complexity for finding the X_i's and A_i's is dominated by the factoring problem. Hence up to polylog factors, it runs in $O(N^2)$ bit operations for fixed accuracy and in roughly $O(N)$ bit operations with well-conditioned data.

These running times are optimal up to polylogarithmic factors. In order to verify this, we begin by replacing Sym_i in (8) by the coefficient S_i of the monic polynomial

(10). Setting $F_1, \ldots, F_{N-1}$ all to 0 and $F_N = 1$, we may rewrite (8) as the following Toeplitz system.

$$
(12) \quad
\begin{bmatrix} S_1 \\ -S_2 \\ \vdots \\ (-1)^{N+1}S_N \end{bmatrix}
=
\begin{bmatrix}
1 & 0 & \cdots & 0 \\
-S_1 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
(-1)^{N-1}S_{N-1} & (-1)^{N-2}S_{N-2} & \cdots & 1
\end{bmatrix}
\cdot
\begin{bmatrix} F_{N+1} \\ F_{N+2} \\ \vdots \\ F_{2N} \end{bmatrix}
$$

This can be solved for the $F_{N+1}, \ldots, F_{2N}$ up to polylog factors in $O(N)$ time ([Pan92]). On the other hand, if the $X_1, \ldots, X_N$ belong to a solution of (1) for this choice of $F_1, \ldots, F_{2N}$, then we showed that the X_i represent all the roots of (10). Therefore solving (1) takes asymptotically at least as much time as a complete factorization. On the other hand, the stated $O(N^2)$ worst case time and $O(N)$ well-conditioned case time are optimal, up to polylog factors in N, for complete approximate factorization (see [Pan02]). The speed of this algorithm will become even more significant when we explore some applications below.

It is worth noting that the algorithm of [Pan02] is quite complicated and has not been implemented in any numerical package that we are aware of. Moreover, the widely used versions of the Jenkins-Traub factorization algorithm are still likely to be much faster than the near optimal algorithm for $N < 100$.

3. Applications

Here we present an application to a generic fitting problem. We will start with a general component minimization scheme before narrowing down to the special case of approximating by exponential sums. In what follows, let $\mathbb{K}$ be any complete field with respect to some norm.

Given a measure space (X, μ), where μ is a $\mathbb{K}$ valued measure, suppose that one has an arbitrary function $F \in L^2_{\text{loc}}(X, \mu)$ taking values in $\mathbb{K}$. We may assume that F is restricted to a region $S \subset X$, the sampling domain. We wish to find the best fit of F as a linear combination of N functions from the collection $\{f_\alpha\}_{\alpha \in \mathcal{A}} \subset L^2_{\text{loc}}(X, \mu)$ for some parameter space $\mathcal{A} \subset \mathbb{K}$ and natural number N. The approximating function to F will be

$$
(13) \qquad \widetilde{F}_N = \sum_{i=1}^{N} \beta_i f_{\alpha_i},
$$

for some choice of constants $\beta_i \in \mathbb{K}$ and parameter indices $\alpha_i \in \mathcal{A}$ such that $\widetilde{F}_N$ is closest to F in a certain sense that we will make precise shortly.

We need to find the best choice of coefficients β_i and parameters α_i for $\widetilde{F}_N$. To do this, we choose kernel functions $K_n \in L^2(S, \mu)$ based on the collection $\{f_\alpha\}_{\alpha \in \mathcal{A}}$ (but independent of α) such that for each integer $n = 0, \ldots, 2N - 1$,

$$
\int_S K_n(x) f_\alpha(x) d\mu(x) = G_S(\alpha) P_S(\alpha)^n,
$$

for functions G_S and P_S on $\mathcal{A}$. The subscript S is used to indicate that these functions, along with the kernel functions, may depend on the domain as well. Note that if $G_S(\alpha) = 0$ (respectively, $P_S(\alpha) = 0$), then f_α is orthogonal to all of the K_n for $n = 0, \ldots$ (respectively, for $n = 1, \ldots$).

At this point, we put the above into the framework of system (1). Setting $A_i = \beta_i G_S(\alpha_i)$ and $X_i = P_S(\alpha_i)$, we may easily verify that,

$$(14) \qquad \int_S K_n(x)\widetilde{F}_N(x)d\mu(x) = \sum_{i=1}^N A_i X_i^n.$$

As this suggests, we can set $F_i = \int_S K_{i-1}(x)F(x)d\mu(x)$ for $i = 1, \ldots, 2N$ and then solve the system (1) for the A_i, X_i. We must assume the F_i are finite, which holds independently of $K_{i-1} \in L^2(S)$ if $F \in L^2(S)$. We must further assume that $P_S(\mathcal{A})$ contains all possible solution values for the X_i as the $F_1, \ldots, F_{2N}$ range over all values produced from the class of functions F we wish to consider. For instance, this last condition is always satisfied if we choose $\mathcal{A}$ and the family $\{K_n\}$ so that $P_S(\mathcal{A}) = \mathbb{K}$.

Under these conditions, then at least one solution of $P_S(\alpha_i) = X_i$ exists with $\alpha_i \in \mathcal{A}$. If $A_i = 0$, then we simply remove the corresponding term from the sum (13); otherwise we set $\beta_i = \frac{A_i}{G_S(\alpha_i)}$. Substituting these values of α_i and β_i in (13) will produce a candidate minimizer, $\widetilde{F}_N$, with at most N terms.

Any such $\widetilde{F}_N$ will satisfy

$$(15) \qquad \int_S K_n(x)\left(F(x) - \widetilde{F}_N(x)\right)d\mu(x) = 0,$$

for $n = 0, \ldots, 2N - 1$. In other words, $\widetilde{F}_N - F$ lies in $\mathrm{Span}_{L^2}\{K_0, \ldots, K_{2N-1}\}^\perp$. Also note that whenever $F(x)$ is given by a linear combination from $\{f_\alpha\}_{\alpha \in \mathcal{A}}$ with at most N terms, then F will be included among the possibilities for $\widetilde{F}_N$.

For all $n \geq 2N$, we have from the definition of $C_{k,j}$ and the validity of Proposition 2.3 for all $k \geq 0$ that

$$(16) \qquad \int_S K_n(x)\widetilde{F}_N(x)d\mu(x) = \sum_{i=1}^N A_i X_i^n = \sum_{i=1}^N C_{n-N+1,i}F_i.$$

The $C_{n-N+1,i}$ are all homogeneous polynomials in the X_i of degree $n - i + 1$, but they may each have many terms. We will need to estimate the number of terms in $C_{k,j}$ since we have no priori control over the size of the A_i.

Observe that by iterating the recurrence relation in Proposition 2.3 we obtain

$$C_{k,N} = \sum_{i=1}^{\min\{k,N\}} (-1)^{i+1}\,\mathrm{Sym}_i\,C_{k-i,N}.$$

By induction on k, we obtain that $C_{k,N}$ is precisely the sum of all distinct monic monomials of total degree k in the variables $X_1, \ldots, X_N$. For $j \neq N$ it is not as apparent how to describe $C_{k,j}$. Nevertheless, by iterating the recurrence relation twice, we obtain

$$C_{k,j} = (-1)^{N-j}[\mathrm{Sym}_{N-j+1}\,C_{k-1,N} - \mathrm{Sym}_{N-j+2}\,C_{k-2,N}] + C_{k-2,j-2}.$$

Since Sym_i consists of the sum of all products of i distinct elements from $X_1, \ldots, X_N$, it follows from the description of $C_{k,N}$ that every term in the expansion of $\mathrm{Sym}_{N-j+2}\,C_{k-2,N}$ is represented in $\mathrm{Sym}_{N-j+1}\,C_{k-1,N}$. Since $C_{1,j} = (-1)^{N-j}S_{N-j+1}$, by induction each term in $C_{k,j}$ has the same sign $(-1)^{N-j}$.

Therefore, using that Sym_i has $\binom{N}{i}$ terms, the recurrence relation for the sum, $c_{k,j}$, of the absolute value of the coefficients of all terms in $C_{k,j}$ is

$$c_{k,j} = \binom{N}{N-j+1} c_{k-1,N} - c_{k-1,j-1}$$

with $c_{0,N} = 1$ and $c_{0,j} = 0$ for $j = 1, \ldots, N-1$. It is straightforward to verify that this has solution $c_{k,j} = \frac{(N-1+k)!}{(N-j+k)\,(k-1)!\,(j-1)!\,(N-j)!}$. This is a polynomial in k of degree $N-1$. Moreover, estimating each factorial term using Sterling's formula shows that for each $j \in \{1, \ldots, N\}$, the quantity $N\,2^{-k}\,e^{-\frac{3}{2}k}\,c_{k,j}$ tends to 0 as either k or N tends to ∞, so long as $k \geq N$.

Therefore, if we assume that the sequence $\{F_i\}$ tends to 0 and that for some $\epsilon > 0$, $|P_S(\alpha)| \leq \frac{e^{-3/2}}{2} - \epsilon$ for all $\alpha \in \mathcal{A}$, then for all $n > 2N$,

$$\left| \int_S K_n(x) \left(F(x) - \widetilde{F}_N(x) \right) d\mu(x) \right| \leq |F_n| + \left| \sum_{i=1}^{N} (2e^{3/2})^{-n+i-1} c_{n-N+1,i} F_i \right|$$

$$\leq |F_n| + Ce^{-\epsilon(n-N)},$$

where $C > 0$ is a constant depending only on the largest F_i. Note that the right-hand side tends to 0 as either n or N tends to infinity since $n - N > N$ by assumption.

Now assume that $F \in L^2(S)$, so that the F_i are square summable. Moreover, suppose the sequence $\{K_n\}$ contains a nondegenerate basis of $L^2(S)$, by which we mean there is a constant $C' > 0$ such that for all $f \in L^2(S)$,

$$\sum_{n \geq 0} \langle K_n, f \rangle_{L^2(S)}^2 \geq C' \|f\|_{L^2(S)}^2.$$

In this case,

$$\|F - \widetilde{F}_N\|_{L^2(S)}^2 \leq \sum_{n \geq 0} \left\langle K_n, F - \tilde{F}_N \right\rangle_{L^2(S)}^2 \leq C' \sum_{n > 2N} \left(|F_n| + Ce^{-\epsilon(n-N)} \right)^2.$$

Hence, $\|F - \widetilde{F}_N\|_{L^2(S)}$ tends to 0 as N tends to infinity. If we only require that $X_i < 1$ for $i = 1, \ldots, N$, then we still may conclude from equation (14) that each $\tilde{F}_N$ belongs to $L^2(S)$ for each N.

Let $H_{F,N}$ denote the Hankel matrix in system (9). Fix N and suppose F satisfies the condition that for $i = N+1, \ldots, 2N$, we have $|F_i| < \frac{1}{CN}$ where C is any upper bound for $\|H_{F,N}^{-1}\|$. Then by system (9) the norm of the coefficients of the polynomial (10) are all less than $\frac{1}{N}$ and therefore the roots X_i are all less than 1. In this case, if the $\{K_n\}$ contain a nondegenerate basis, then $\tilde{F}_N \in L^2(S)$.

We point out that with obvious modifications, the above analysis applies to the case when the K_n represent distributions instead of functions. Now we will focus on applying these ideas to the concrete example of exponentials.

4. Fitting exponentials

In terms of the context from the previous section, we set $\mathbb{K} = \mathbb{C}$ and consider the collection of functions $f_\alpha(x) = e^{-\alpha x}$ with $\alpha \in \mathcal{A} = \mathbb{C}$ and any $S \subset \mathbb{C}$. We may parameterize S by the variable t. In short we will fit an equally spaced complex time series $F_1, \ldots, F_{2N}$ to a sum of complex exponentials $F(t) = \sum_{j=1}^{N} A_j e^{\alpha_j t}$.

We first assume that for $j = 1, \ldots, 2N$, the F_j represent given approximations to $F(t_j)$ where $t_j = j/q + r$ for fixed constants q and r. In terms of the previous section, this corresponds to taking $K_j = \delta_{t_j}$ for $j = 1, \ldots, 2N$, where δ_{t_j} is a Dirac delta at t_j. We can then solve for each A_j and each e^{α_j} simply by using our algorithm to solve the corresponding version of (2) with $q_i = q$ and $r_i = r$.

Example 4.1. Let $F(t)$ be a series of N exponential functions each of whose coefficients and exponents are randomly chosen using a uniform distribution over the square in $\mathbb{C}$ with opposite corners $\pm(1 + i)$. To keep the exponentials from becoming too large or small, we set $q = 2N$ and $r = 0$ so that $F_j = F\left(\frac{j}{2N}\right)$ for $j = 0, \ldots, 2N - 1$.

We implemented our algorithm in the *Mathematica* computer algebra system, version 4.1, on a 900 MHz Pentium III machine equipped with 400 megabytes of RAM. Since the purpose of this example is to demonstrate the viability of the algorithm for relatively small N, we did not attempt to implement the optimal algorithm. Instead, we chose to use *Mathematica's* built-in linear solver, which has computational complexity $O(N^3)$, for the two linear systems. We also used *Mathematica's* implementation of the suboptimal Jenkins-Traub algorithm for the polynomial factorization.

Nevertheless, this implementation could still recover the original $F(t)$ with an accuracy of 10^{-9} from the F_i in approximately 2.6 seconds of CPU time for $N = 100$. For $N = 1000$ the entire algorithm takes approximately 1100 seconds. The single most time consuming step is the rooting of the polynomial which was accomplished in 504 seconds of CPU time.

By contrast, *Mathematica's* nonlinear regression routine, based on the Levenberg-Marquart algorithm, expended 15 seconds of CPU time on the same machine to fit an exponential sum $F(t)$ with $N = 100$, but restricted to real exponents and coefficients. Similarly, the case with $N = 1000$ exhausted 1239 seconds of CPU time. For some reason, the actual time to complete this fit was more than 12 times as long, even without any disk paging.

For many typical numerical applications, the F_i may represent inaccurate or unevenly spaced data. For this situation we may want to rely on averaging techniques. For instance, when $S = [0, \infty)$ and we are modelling decaying exponentials, i.e., $\mathrm{Re}(\alpha_i) < 0$, then we can use our method when the Laplace transform of K_n has the correct form. Table 1 is a partial list collected from [IK77] for possible choices of $K_n(x)$ along with the corresponding G_S and P_S which fit into the framework of Section 3. For instance, choosing $K_n(x) = \frac{x^{n-1}}{\Gamma(n)}$ leads to equating the statistical moments of $\widetilde{F}_N$ with those of F; other choices lead to different notions of approximation.

Here $J_n(t)$ is the nth Bessel J function, $L_n(t)$ is the nth Laguerre polynomial, and $H_n(t)$ is the nth Hermite polynomial. We see from Table 1 that for these kernel functions either α is uniquely determined from $P_S(\alpha)$ for α on $\mathbb{C}$ (with a given branch cut), a half plane ($P_S = \frac{2p}{\alpha^2 + p^2}$), or a strip (in the exponential cases).

For a given choice of the K_n it takes at most $O(Nm)$ time to calculate the integrals F_i for $i = 0, \ldots, 2N - 1$ when $F(t)$ has been sampled at m points. Hence the entire algorithm to compute a fit $\widetilde{F}_N$ is, in the worst case, $O(N^2 \log(N) + Nm)$ for a fixed precision. Note that for some choices of K_n the integration can be done in $O(N + m)$ time.

Table 1.

Choices of Kernel Functions for $f_\alpha(t) = e^{-\alpha t}$		
$K_n(t)$	$G_S(\alpha)$	$P_S(\alpha)$
$\begin{cases} 0 & (0 \le t < n) \\ 1 & (n \le t) \end{cases}$	$\frac{1}{\alpha}$	$e^{-\alpha}$
$e^{-pt}\frac{t^{n-1}}{\Gamma(n)}$	1	$\frac{1}{\alpha+p}$
$\frac{1}{\sqrt{\pi t}}e^{-n^2/(4t)}$	$\frac{1}{\sqrt{\alpha}}$	$e^{-\sqrt{\alpha}}$
$\frac{1}{\sqrt{n}}\sin(\sqrt{nt})$	$\sqrt{\frac{\pi}{4\alpha^3}}$	$e^{-\frac{1}{4\alpha}}$
$\frac{1}{\sqrt{t}}\cos(\sqrt{nt})$	$\sqrt{\frac{\pi}{\alpha}}$	$e^{-\frac{1}{4\alpha}}$
$J_n(t)$	$\frac{1}{\sqrt{1+\alpha^2}}$	$\sqrt{1+\alpha^2}-\alpha$
$\frac{n}{t}J_n(pt)$	1	$\frac{1}{p}\left(\sqrt{p^2+\alpha^2}-\alpha\right)$
$\frac{t^n}{\Gamma(n+1/2)}J_n(pt)$	$\frac{1}{\sqrt{\pi(p^2+\alpha^2)}}$	$\frac{2p}{\alpha^2+p^2}$
$\left(\frac{t}{n}\right)^{p/2}J_p(\sqrt{nt})$	$2^{-p}\alpha^{-(p+1)}$	$e^{-\frac{1}{4\alpha}}$
$\frac{t^p\Gamma(n-p+1)}{n!}L_n^{(p)}(T)$	$\frac{1}{\alpha(\alpha-1)^p}$	$\frac{\alpha-1}{\alpha}$
$\frac{1}{(2n+1)!!}H_{2n+1}(\sqrt{t})$	$\sqrt{\frac{\pi}{2\alpha^3}}$	$\frac{1-\alpha}{\alpha}$
$\frac{1}{\sqrt{t}(2n-1)!!}H_{2n}(\sqrt{t})$	$\sqrt{\frac{\pi}{\alpha}}$	$\frac{1-\alpha}{\alpha}$

Appendix A. Other noniterative methods in special cases

Here we report on a couple of previously existing methods to handle special cases of the problem of fitting exponential series. The more well-known techniques of nonlinear regression and the Prony method have already been well analyzed in the literature, so we will not discuss them here. Though very effective for small N these inherently iterative methods suffer from being asymptotically inefficient.

The special cases presented here have the potential to run in $O(N \log N)$ operations for well-behaved inputs and are both simple and elegant. Unfortunately, they only recover exponents α_j lying in a particular one real dimensional subset of $\mathbb{C}$. Moreover, their numerical deficiencies make them for the most part impractical.

The general scheme is to find an appropriate operator $\Box$ such that $\Box e^{\alpha t} = \delta(t-\alpha)$ where $\delta(t)$ is the Dirac delta distribution centered at 0. Thus if $F(t) = \sum_{j=1}^N A_j e^{\alpha_j t}$, we may express this as

$$F(t) = \int_{-\infty}^{\infty} f(s)e^{st}ds$$

for

$$f(s) = \sum_{j=1}^N A_j \delta(s-\alpha_j),$$

and $\Box F(t) = f(t)$.

When applying discrete versions of such an operator, the inherent numerical inaccuracies do not always make them a good choice for recovery of a best fit approximation, especially in the presence of noise. However, they can at least give a practical means of finding the number of terms, N, in the sum, the hope being that the number of peaks N of a numerical realization of $\Box F(t)$ may be discernable even when their relative heights and positions are not.

In the first case, using the notation from Section 3, if $S = \mathbb{R}$ and $\alpha_j \in \mathbb{R} \cdot i$, then we could simply take the Fourier transform of F to obtain a sum of Dirac measures centered at the $-i \cdot \alpha_j$ with weights A_j. Numerically, this can be accomplished using a nonuniform fast Fourier transform algorithm and taking A_j to be the total mass of each peak which will be centered approximately at $-i \cdot \alpha_j$. This can be done in $O(N \log N)$ time for fixed precision assuming the sampling is less than a fixed multiple of N ([DR93] and see [NL99, Bey95, CFR84] for consequent improvements). However the constants in the time dependency may be large due to the need to resolve the peaks enough to accurately estimate the A_i. We also needed to assume that $F(t)$ can be sampled at an appropriate set of real values of t.

Similarly, when $\alpha_j \in R$ with $\alpha_j < 0$ we could take the inverse Laplace transform

$$f(s) = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} F(t)e^{st}dt \quad \text{for any} \quad \sigma > 0$$

to obtain a sum of Dirac measures centered at $-\alpha_j$ with weights A_j. Numerically this can be done using an algorithm based on the fast Fourier transform (FFT) when F has been sampled along the line $\sigma + i\mathbb{R}$. Otherwise, if F is sampled only for positive real values (which one might commonly expect in practice), then one may use the inversion formula arising from the Mellin transform,

$$f(s) = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} \frac{s^{-p}}{\Gamma(1-p)} \left(\int_0^\infty F(t)t^{-p}dt \right) dp.$$

Here $\Gamma(x)$ is the Euler gamma function and the function F must also satisfy certain analytic conditions (see [AR00]) in order for the integrals to converge. It seems that the general method described in [Str92] for fast evaluation of discrete transforms with sufficiently decaying kernels, when applied to the discrete application of the above transform given in [AR00], would give a means of computing this numerically in $O(N \log N)$ time.

The Gardner transform ([GGLM59]) provides an efficient alternate approach to inverting the Laplace transform in this case when $F(t)$ is sampled at real values only. Its advantage is that it can be simply evaluated from three FFT's.

To solve for $f(s)$ in terms of $F(t)$, we make it a convolution in order to conveniently take its Fourier transform. To do so, we make a transformation of variables in the Laplace transform. From the equation for $F(t)$ above under the assumption $\alpha_j < 0$, we recover,

$$e^t F(e^t) = \int_{-\infty}^0 f(s)e^t e^{se^t} ds.$$

Then making a change of variables from s to $-\beta e^{-x}$ (for any positive constant β) gives us

$$e^t F(e^t) = \int_{-\infty}^\infty f\left(-\beta e^{-x}\right) e^{-\beta e^{t-x}} \beta e^{t-x} dx.$$

We observe that the integral is a convolution with $K(z) = \beta e^{-\beta e^z} e^z$. Taking the Fourier transform of both sides then gives

$$\mathcal{F}_{t \to \omega}\left(e^t F\left(e^t\right)\right) = \mathcal{F}_{x \to \omega}\left(f\left(-\beta e^{-x}\right)\right)\mathcal{F}_{x \to \omega}\left(K(x)\right).$$

For the purpose of numerical stability, we multiply the left-hand side by a gaussian factor $e^{-(\frac{\omega}{\nu})^2}$ for some constant ν. Since this is equivalent to convolving $F(t)$ by the same factor at the outset, it acts as a low pass filter which cuts out baseline ripples which are far away from the central peak of $\mathcal{F}_{t \to \omega}\left(e^t F\left(e^t\right)\right)$ (see [SCSB76]). Good values of the constant ν vary, but often taking ν to be about half of the length of the interval on which the Fourier transform of F is defined works well.

The final result is

$$f\left(-\beta e^{-x}\right) = \mathcal{F}_{\omega \to x}^{-1}\left(\frac{\mathcal{F}_{t \to \omega}\left(e^t F\left(e^t\right)\right)e^{-(\frac{\omega}{\nu})^2}}{\mathcal{F}_{x \to \omega}\left(K(x)\right)}\right).$$

The purpose of the constant β is to shift the peak of $K(z)$ by a factor of $\log(\beta)$ in order to center it in the domain of $e^t F(e^t)$ so that it goes to zero on either edge of this window. This reduces numerical inaccuracies ([SCSB76]).

The numerical problems with applying this transform mostly stem from the constraint on the domain of the data. For example, the maximum of $e^t F(e^t)$ where $F(t)$ is a single exponential $e^{\alpha t}$ occurs at $t = \log \frac{-1}{\alpha}$ and decays very rapidly from that point. Therefore the domain of $F(t)$ should include points nearby to the points $\frac{-1}{\alpha_j}$ in order to obtain an accurate sampling. Clearly, integrating $e^t F(e^t)$ accurately from sampled data can be problematic since the domain of F is compressed so rapidly.

Since $\mathcal{F}_{x \to \omega}\left(K(x)\right)$ appears in the denominator in the Gardner transform, there are inaccuracies introduced whenever it decays on the portion of the domain of the numerator representing significant data. By examining Laplace transforms of $K(x)$ on the positive and negative imaginary axes, we find that

$$\mathcal{F}_{x \to \omega}\left(K(x)\right) = \beta^{-i2\pi s}\Gamma(1 + i2\pi s).$$

Therefore it is also important to choose β so that the domain where this function has small norm does not overlap with the domain of the data window used in the computation of $\mathcal{F}_{t \to \omega}\left(e^t F\left(e^t\right)\right)$. Since this later function is modelled by

$$\sum_{j=1}^{N} A_j(-\alpha_j)^{-(1+i2\pi s)}\Gamma(1 + i2\pi s),$$

this may be impossible if the α_j are spread apart.

In practice, we found that when applying this algorithm with machine precision of 10^{-16}, the number of peaks accurately corresponded to the number of exponential terms N except when the α_j were separated by less than 10^{-3}. Even when choosing optimal values of β and ν with fewer than six α_j uniformly distributed between 0 and 1, the precision of the α_j was usually less than 10^{-3}. On the other hand, even in cases when the peaks about the α_j are completely separated, it is difficult to estimate their mass to within a factor of 1 of the value of the original A_i. Due to the extensive overlap of peaks when two α_j are reasonably close, it is difficult to make any reasonable estimate at all.

Acknowledgments

References

[AR00]   R. G. Airapetyan and A. G. Ramm. Numerical inversion of the Laplace transform from the real axis. *J. Math. Anal. Appl.*, 248(2):572–587, 2000. MR2001g:65171

[BDCMM01] D. Bini, G. M. Del Corso, G. Manzini, and L. Margara. Inversion of circulant matrices over Z_m. *Math. Comp.*, 70(235):1169–1182, 2001. MR2001j:65047

[Bey95]   G. Beylkin. On the fast Fourier transform of functions with singularities. *Appl. Comput. Harmon. Anal.*, 2(4):363–381, 1995. MR96i:65122

[CFR84]   A. B. Cain, J. H. Ferziger, and W. C. Reynolds. Discrete orthogonal function expansions for nonuniform grids using the fast Fourier transform. *J. Comput. Phys.*, 56(2):272–286, 1984. MR86e:65170

[DR93]   A. Dutt and V. Rokhlin. Fast Fourier transforms for nonequispaced data. *SIAM J. Sci. Comput.*, 14(6):1368–1393, 1993. MR95d:65114

[GGLM59]   D. G. Gardner, J. C. Gardner, G. Laush, and W. W. Meinke. Method for the analysis of multicomponent experimental decay curves. *J. Chem. Phys.*, 31:978–986, 1959.

[Gou97]   M. C. Gouveia. Regular Hankel matrices over integral domains. *Linear Algebra Appl.*, 255:335–347, 1997. MR98c:15070

[IK77]   S. Iyanaga and Y. Kawada, editors. *Encyclopedic dictionary of mathematics. Vol. II.* MIT Press, Cambridge, Mass., Japanese edition, 1977. Networks to zeta functions, Translation reviewed by Kenneth O. May. MR81d:00003b

[ISL92]   A. Iserles, E. B. Saff, and Xiaoyan Liu. On zeros of Hankel determinants with iterated polynomial entries. *IMA J. Numer. Anal.*, 12(3):387–403, 1992. IMA Conference on Dynamics of Numerics and Numerics of Dynamics (Bristol, 1990). MR93m:58093

[Las90]   Alain Lascoux. Inversion des matrices de Hankel. *Linear Algebra Appl.*, 129:77–102, 1990. MR91d:15064

[NL99]   Nhu Nguyen and Qing Huo Liu. The regular Fourier matrices and nonuniform fast Fourier transforms. *SIAM J. Sci. Comput.*, 21(1):283–293 (electronic), 1999. MR2000j:65130

[Pan92]   V. Y. Pan. Complexity of computations with matrices and polynomials. *SIAM Rev.*, 34(2):225–262, 1992. MR93g:65179

[Pan02]   Victor Y. Pan. Univariate polynomials: nearly optimal algorithms for numerical factorization and root-finding. *J. Symbolic Comput.*, 33(5):701–733, 2002. Computer algebra (London, ON, 2001). MR2003f:13030

[Pau01]   Sebastian Pauli. Factoring polynomials over local fields. *J. Symbolic Comput.*, 32(5):533–547, 2001. MR2002h:13038

[SCSB76]   M. R. Smith, S. Cohn-Sfetcu, and H. A. Buckmaster. Decomposition of multicomponent exponential decays by spectral analytic techniques. *Technometrics*, 18(4):467–482, 1976. MR55:14397

[Sho95]   Victor Shoup. A new polynomial factorization algorithm and its implementation. *J. Symbolic Comput.*, 20(4):363–397, 1995. MR97d:12011

[Str92]   John Strain. A fast Laplace transform based on Laguerre functions. *Math. Comp.*, 58(197):275–283, 1992. MR92e:44001

[vzGP01]   Joachim von zur Gathen and Daniel Panario. Factoring polynomials over finite fields: a survey. *J. Symbolic Comput.*, 31(1-2):3–17, 2001. Computational algebra and number theory (Milwaukee, WI, 1996). MR2001k:11253

Department of Mathematics, University of Chicago, 5734 S. University Avenue, Chicago, Illinois 60637

Current address: Department of Mathematics, Rawles Hall, Indiana University, Bloomington, Indiana 47401

E-mail address: `cconnell@math.uchicago.edu`