Symbolic-Numeric Sparse Polynomial Interpolation in Chebyshev Basis and Trigonometric Interpolation

Mark Giesbrecht\textsuperscript{1}, George Labahn\textsuperscript{1}, and Wen-shin Lee\textsuperscript{2}

\textsuperscript{1} School of Computer Science, University of Waterloo
Waterloo, Ontario N2L 3G1, Canada
{mg,glabahn}@uwaterloo.ca
http://www.scg.uwaterloo.ca/{mg,glabahn}

\textsuperscript{2} GALAAD, INRIA Sophia Antipolis
BP 93, 06902 Sophia Antipolis, France
wen-shin.lee@sophia.inria.fr
http://www.wen-shin.com

Abstract. We consider the problem of efficiently interpolating an “approximate” black-box polynomial \( p(x) \) that is sparse when represented in the Chebyshev basis. Our computations will be in a traditional floating-point environment, and their numerical sensitivity will be investigated. We also consider the related problem of interpolating a sparse linear combination of “approximate” trigonometric functions. The costs of all our algorithms will be sensitive to the sparsity of the output.

1 Introduction

A black-box polynomial \( p(x) \in \mathbb{R}[x] \) is a procedure that can output the value of \( p(\alpha) \) at any given input \( \alpha \in \mathbb{R} \). The traditional definition of a sparse polynomial is a sum of a small number of non-zero terms, where the terms are of the form \( cx^k \) for some constant \( c \in \mathbb{R} \) and exponent \( k \in \mathbb{Z}_{\geq 0} \). It is also reasonable to consider polynomials whose representations are sparse in other bases, such as the Chebyshev polynomials. Now, let \( T_k(x) \) denote the \( k \)th Chebyshev polynomial of the first kind:

\[
T_0(x) = 1, \quad T_1(x) = x, \quad T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x) \quad \text{for} \quad k \geq 2.
\]

Any polynomial \( p(x) \) can be represented in the Chebyshev basis as

\[
p(x) = \sum_{j=1}^{t} c_j T_{d_j}(x),
\]

where \( 0 \leq d_1 < d_2 < \cdots < d_t \) and \( c_1, \ldots, c_t \in \mathbb{R} \). Lakshman and Saunders \cite{10} give a sparse algorithm (using exact arithmetic in \( \mathbb{Q}[x] \)) which interpolates the Chebyshev representation of a black-box polynomial. Its cost (the number of black-box evaluations plus auxiliary field operations) is polynomial of the sparsity \( t \) in the Chebyshev representation.

In this paper, we consider the situation in which both the inputs and outputs of the black box for \( p(x) \) are precise only to a fixed precision. We give two approaches to solving the sparse Chebyshev interpolation problem. The first is a modification of the method of Lakshman and Saunders \cite{10}. The other is obtained by solving a generalized eigenvalue problem. Both approaches may be regarded as extensions of the symbolic-numeric sparse polynomial interpolation algorithms in the standard power basis presented in \cite{6}.

The trigonometric function \( \cos k\theta \) can be regarded as the \( k \)th Chebyshev polynomial in \( \cos \theta \). We also consider the related problem of efficiently interpolating a sparse linear combination of trigonometric functions \( \cos k\theta \) and \( \sin k\theta \). Thus, we seek to efficiently interpolate \( f \) as

\[
f(\theta) = \frac{A_0}{2} + \sum_{k=1}^{m} (A_k \cos k\theta + B_k \sin k\theta),
\]

in which many \( A_k \in \mathbb{R} \) and \( B_k \in \mathbb{R} \) are zero.
It is standard to interpolate $f$ on a uniform partition of $[0, 2\pi]$, and that $f(\theta)$ is interpolated from the points $(\phi_k, f(\phi_k))$, where $\phi_k = 2\pi k/n$ for some appropriately chosen $n$ (for an overview see [8], Section 9). The cost of such methods depends on the maximum number of terms in the target function (i.e., on $m$ in (2)). Typically $n = 2m$ or $n = 2m + 1$ and for $k = 0, \ldots, m$, so every $\sin k\theta$ and $\cos k\theta$ is interpolated, regardless how many of them have zero coefficients. Thus, these algorithms require time that is polynomial in $m$.

By a variant of Prony’s method [2, 14], the interpolation of a trigonometric function $f(x)$ can be sensitive to its sparsity, the number of non-zero $A_k, B_k$ in (2) [8, pp. 382-386]. By combining this with a connection between Prony’s method and Ben-Or/Tiwari sparse interpolation observed in [6], we exploit the progress in sparse Chebyshev polynomial interpolation and show how a sparse linear combination of trigonometric functions can be efficiently interpolated by solving a generalized eigenvalue problem.

### 2 Sparse Interpolation in the Chebyshev Basis

In this section we introduce a Prony-like algorithm for the interpolation of polynomials in the Chebyshev basis, which is derived in [10]. In Section 3 we examine the numerical sensitivity of this algorithm and present a simple modification which improves stability.

Suppose that $p(x)$ is represented with respect to the Chebyshev basis as in (1). We define the polynomial

$$A(z) = \prod_{j=1}^{t}(z - T_{d_j}(a)) = T_t(z) + \lambda_{t-1}T_{t-1}(z) + \cdots + \lambda_0$$

for some $a > 1$. The polynomial $A(z)$ provides the linear relations between the evaluations of $p(x)$ at $T_k(a)$ (see [10]): for $\alpha_k = p(T_k(a))$,

$$\sum_{j=0}^{t-1} \lambda_j (\alpha_{j+1} + \alpha_{j-i}) = - (\alpha_{t+i} + \alpha_{t-i}) \text{ for } i \geq 0. \quad (3)$$

Relations in (3) give the following $t \times t$ symmetric Hankel-plus-Toeplitz system:

$$\begin{bmatrix}
2\alpha_0 & 2\alpha_1 & \ldots & 2\alpha_{t-1} \\
2\alpha_1 & \alpha_2 + \alpha_0 & \ldots & \alpha_t + \alpha_{t-2} \\
\vdots & \vdots & \ddots & \vdots \\
2\alpha_{t-1} & \alpha_t + \alpha_{t-2} & \ldots & \alpha_{2t-2} + \alpha_0
\end{bmatrix}_{A}
\begin{bmatrix}
\lambda_0 \\
\lambda_1 \\
\vdots \\
\lambda_{t-1}
\end{bmatrix} =
\begin{bmatrix}
2\alpha_t \\
\alpha_{t+1} + \alpha_{t-1} \\
\vdots \\
\alpha_{2t-1} + \alpha_1
\end{bmatrix} \quad (4)$$

By showing that $A$ is non-singular [10, Lemma 6], Lakshman and Saunders give a sparse polynomial interpolation algorithm in the Chebyshev basis for exact arithmetic.

**Algorithm: SparseChebyshevInterp** [10]

Given a black-box polynomial $p(x)$ and the number of non-zero terms $t$ of $p(x)$ in the Chebyshev basis, find $c_1, \ldots, c_t \in \mathbb{R}$ and $d_1, \ldots, d_t \in \mathbb{Z}_{\geq 0}$ such that $p(x) = \sum_{j=1}^{t} c_j T_{d_j}(x)$.

1. [Evaluate $p(T_k(a))$.] Choose $a > 1$, evaluate $\alpha_k = p(T_k(a))$ for $k = 0, 1, \ldots, 2t - 1$.
2. [Degrees $d_j$.]
   2.1 Solve the symmetric Hankel-plus-Toeplitz system in (4) to obtain $A(z)$.
   2.2 Obtain $T_{d_1}(a), \ldots, T_{d_t}(a)$ by finding all roots for $A(z) = 0$. The values of $d_j$ can be recovered from $T_{d_j}(a)$ for $1 \leq j \leq t$. 

© Giesbrecht, Labahn, and Lee.
(3) [Coefficients \(c_j\).] Coefficients \(c_j\) can be obtained by solving the transposed Vandermonde-like system:

\[
\begin{bmatrix}
1 & \ldots & 1 \\
T_{d_1}(T_0(a)) & \ldots & T_{d_1}(T_0(a)) \\
\vdots & \ddots & \vdots \\
T_{d_1}(T_{t-1}(a)) & \ldots & T_{d_1}(T_{t-1}(a))
\end{bmatrix}
\begin{bmatrix}
c_1 \\
c_2 \\
\vdots \\
c_t
\end{bmatrix}
= \begin{bmatrix}
p(T_0(a)) \\
p(T_1(a)) \\
\vdots \\
p(T_{t-1}(a))
\end{bmatrix}.
\]

In the algorithm \texttt{SparseChebyshevInterp}, the target polynomial is evaluated at \(T_k(a)\) for \(a > 1\) because \(T_k(a)\) are strictly monotonically increasing in \(k\) for any \(a > 1\). As a result, both the one-to-one correspondence between \(T_k(a)\) and \(k\) (for the recovery of each \(d_j\)) and the non-singularity of \(A\) in (4) are guaranteed [10]. However, it will be useful in our case to use a smaller value for \(a \in \mathbb{R}\). It is easily proven that for \(N \geq 2d_t\) and \(a = \cos(2\pi/N)\) we have \(T_0(a) > T_1(a) > \cdots > T_{d_t}(a)\). This can be used to establish that, should we choose such an \(a\) in Step 1 of \texttt{SparseChebyshevInterp}, the matrix \(A\) will be non-singular and the algorithm will work as specified. In what follows we will examine the numerical conditioning of \(A\), and the sensitivity of the entire algorithm.

### 3 Numerical Issues with \texttt{SparseChebyshevInterp}

When the sparse Chebyshev interpolation algorithm of the previous section is implemented directly in a floating-point environment, significant numeric errors may be encountered in the solving of the Hankel-plus-Toeplitz system, in finding the roots of the polynomial \(T\) in a floating-point environment, significant numeric errors may be encountered in the solving of the Hankel-plus-Toeplitz system. This problem is avoided when we choose \(a_1 > 1\), the difference among the powers of \(a\) can contribute detrimentally to the ill-conditioning of the Hankel-plus-Toeplitz system. This problem is avoided when we choose \(-1 < a < 1\).

We modify the algorithm to mitigate this ill-conditioning. In the first step, we choose \(a = \cos(2\pi/N)\), where \(N \geq 2d_t\) and \(d_t = \deg p\). Thus we assume \(\deg p\), or at least an upper bound for it, is supplied as part of the input. All other steps remain the same except they are now being computed in floating-point arithmetic.

**Algorithm: \texttt{FPSparseChebyshevInterp} (Step 1)**

Given a black-box polynomial \(p(x)\), the number of non-zero terms \(t\) of \(p(x)\) in the Chebyshev basis, and an upper bound \(D \geq \deg p\), this algorithm determines \(c_j\) and \(d_j\) such that \(\sum_{j=1}^{t} c_j T_{d_j}(x)\) interpolates \(p(x)\).

(1) [Evaluate \(p(T_k(a))\).] Choose \(a = \cos(2\pi/N)\), where \(N \geq 2D\), and evaluate \(\alpha_k = p(T_k(a))\) for \(k = 0, 1, \ldots, 2t - 1\).

In the remainder of this subsection we study the sensitivity of Steps 2.1, 2.2, and 3 in algorithm \texttt{FPSparseChebyshevInterp}.

#### 3.1 Solving the Hankel-plus-Toeplitz System

In general, if the target polynomial \(p(x)\) is of a high degree and \(p(x)\) is evaluated at \(T_k(a)\) for \(a > 1\), the difference among the powers of \(a\) can contribute detrimentally to the ill-conditioning of the Hankel-plus-Toeplitz system. This problem is avoided when we choose \(-1 < a < 1\).

To discuss the condition of the Hankel-plus-Toeplitz system \(A\), we consider its factorization as a product of a lower triangular matrix and a Vandermonde matrix. First note that the polynomial \(T_{kd}(x)\) can be expressed as a \(k\)th degree polynomial in \(T_d(x)\) with leading coefficient \(2^{k-1}\).

**Lemma 1.** For \(k \geq 1\),

\[
T_{kd}(x) = 2^{k-1}T_d(x) + \sum_{j=0}^{k-1} \gamma_j T_d^j(x)
\]

(see [9, Lemma 10]).
Based on Lemma 1, the Vandermonde-like $W$ in (5) can be factored as a product of a lower triangular matrix $L$ and a Vandermonde matrix $V$:

$$
W = \begin{bmatrix}
    1 & 0 & \cdots & 0 \\
    0 & 1 & \cdots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    \ast & \ast & \cdots & 2^{t-2}
\end{bmatrix}
\begin{bmatrix}
    1 & \cdots & 1 \\
    T_{d_1}(a) & \cdots & T_{d_t}(a) \\
    \vdots & \vdots & \ddots & \vdots \\
    T_{d_1}^{t-1}(a) & \cdots & T_{d_t}^{t-1}(a)
\end{bmatrix} = L \cdot V. \tag{7}
$$

With $D = \text{diag}(2c_1, \ldots, 2c_t)$, our factorization of $A$ follows (cf. [9, 10]):

$$
A = WDW^T = LVD(LV)^T = L(VDV^T)L^T. \tag{8}
$$

We take advantage of the factorization above to obtain upper and lower bounds for the condition number of $A$. Throughout this paper we will let $\|A\| = \|A\|_{\infty}$ (for any matrix $A$) be the infinity norms. All results stated will apply to any norm, up to an appropriate multiplicative constant.

**Theorem 1.** Let $A$ be the Hankel-plus-Toeplitz matrix in Step 2.1 of FPSparseChebyshevInterp. Then

$$
\min_j |2c_j| \cdot \|L\|^2 \|V\|^2 \leq \|A^{-1}\| \leq \|L^{-1}\|^2 \|V^{-1}\|^2 \min_j |2c_j|.
$$

**Proof.** Consider the factorization of $A$ in (8). Let $D_j$ be the matrix derived from $D$ by using 0 to replace $2c_j$ in the diagonal. Then the matrix $LVD_jV^T \cdot L^T$ is singular for $1 \leq j \leq t$, and we have

$$
\frac{1}{\|A^{-1}\|} = \min \{\|A - H\|, \text{ H singular} \}
\leq \min \{\|A - LVD_jV^T \cdot L^T\| \} \leq \|L\|^2 \cdot \|V\|^2 \cdot \min \{\|2c_j\| \}.
$$

For the upper bound, since $A^{-1} = (L^T)^{-1} (V^T)^{-1} D^{-1} V^{-1} L^{-1}$, we have

$$
\|A^{-1}\| \leq \|L^{-1}\|^2 \|V^{-1}\|^2 \|D^{-1}\| \leq \|L^{-1}\|^2 \|V^{-1}\|^2 \sum_{j=1}^{t} \|D^{-1}c_j\| \leq \|L^{-1}\|^2 \|V^{-1}\|^2 \cdot \max_j \frac{1}{|2c_j|}.
$$

It remains to find bounds for the norms of $L$ and $L^{-1}$, and $V$ and $V^{-1}$.

**Lemma 2.** For $L$ as in (7) we have

$$
\|L\| \leq t \cdot 2^{t-1} \quad \text{and} \quad \|L^{-1}\| \leq K := 9.
$$

**Proof.** We first bound the norm of $L$. Recall $T_0(x) = 1$ and the iterative relations

$$
T_{kd}(x) = T_k(T_d(x)) = 2T_d(x) \cdot T_{k-1}(T_d(x)) - T_{k-2}(T_d(x)) = 2T_d(x) \cdot T_{(k-1)d}(x) - T_{(k-2)d}(x).
$$

The coefficient $\gamma_j$ in (6) is either 0 or a $(t - 2)$th degree polynomial in 2, that is

$$
\gamma_j = \sum_{l=0}^{k-2} q_l 2^l \text{ with } q_l \in \{-1, 0, 1\}. \tag{9}
$$

Then $\|L\| \leq t \sum_{l=0}^{t-2} 2^l = t \cdot (2^t - 1)$.

We now bound the norm of $L^{-1}$. The first two rows of matrix $L^{-1}$ are $[1, 0, \ldots, 0]$ and $[0, 1, 0, \ldots, 0]$. We consider the $k$th row in the $t \times t$ matrix $L^{-1}$:

$$
L^{-1} = \begin{bmatrix}
    [l_{1,1}^{-1}, \ldots, l_{1,k}^{-1}, 0, \ldots, 0] \\
    [l_{2,1}^{-1}, \ldots, l_{2,k}^{-1}, 0, \ldots, 0] \\
    \vdots \\
    [l_{t,1}^{-1}, \ldots, l_{t,k}^{-1}, 0, \ldots, 0]
\end{bmatrix}
$$
Lemma 4. Let $k$ no more than $k > 2$. Combining this fact with $l_{k,k}^{-1} = 1/2^{k-2}$, we have

$$l_{k,i}^{-1} \leq \frac{k-i}{2^{k-2}} \leq \frac{2^{k-i+1}}{2^{k-2}} = 2^{3-i} \text{ for } 1 \leq i < k,$$

and

$$\|L_k^{-1}\| \leq \sum_{i=1}^{k-1} 2^{3-i} + \frac{1}{2^{k-2}} \leq 3 + 1 \text{ for } 2 < k \leq t.$$

For $1 \leq j \leq t$, it is obvious that $\|\mathcal{V}\| \leq t$. For $\|\mathcal{V}^{-1}\|$ we have:

**Lemma 3.** Let $\mathcal{V}$ be the Vandermonde matrix in (7), then

$$\|\mathcal{V}^{-1}\| \leq \frac{2^{t-1}}{\min_k \prod_{j=1,j \neq k} |T_{d_k}(a) - T_{d_j}(a)|}.$$  \hfill (10)

**Proof.** See [3, Theorem 1] for $|T_{d_j}(a)| \leq 1$.

**Lemma 4.** Let $\mathcal{A}$ be the Hankel-plus-Toeplitz matrix in the Step 2.1 of FPSparseChebyshevInterp, then

$$\frac{1}{2^{2(t-1)} \cdot t^2 \cdot \min j |2c_j|} \leq \|\mathcal{A}^{-1}\| \leq \frac{K^2 \cdot t \cdot 2^{2(t-1)}}{\min_{j \neq k} |T_{d_k}(a) - T_{d_j}(a)|^2 \cdot \min j |2c_j|}.$$

Note that the previous lemmas also apply to other matrix norms, up to a suitable multiplicative constant.

### 3.2 Root Finding for the Polynomial $A(z) = 0$

We now consider Step 2.2 in algorithm FPSparseChebyshevInterp, in which we find the roots of the polynomial $A(z) = 0$, with coefficients obtained from solving the Hankel-plus-Toeplitz system $\mathcal{A}$ in Step 2.1.

Finding the roots of a polynomial is generally an ill-conditioned problem with respect to perturbations in the coefficients. However, for our polynomial $A(z) = \prod_{j=1}^{t} (z - T_{d_j}(a))$ with $a \in (-1, 1)$ all the roots $T_{d_j}(a)$ are in $(-1, 1)$ and the conditioning depends on the distribution of $T_{d_j}(a)$ in the interval (cf. [16]).

Let $\tilde{y}_j$ be a zero of $A(z) + \epsilon \Gamma(z)$, a perturbation of $A$, where $\Gamma(z) = \epsilon_1 z^t + \epsilon_{t-1} z^{t-1} + \cdots + \epsilon_0 \in \mathbb{R}[z]$, and $\epsilon > 0$ can be thought of as “small.” Then $\tilde{y}_j = T_{d_j}(a) + \sum_{k=1}^{t} \epsilon_k \epsilon_k^k \approx T_{d_j}(a) + \epsilon_1 \epsilon$ for some $\epsilon_1, \epsilon_2, \ldots$, and

$$A(T_{d_j}(a) + \epsilon_1 \epsilon) + \epsilon \Gamma(T_{d_j}(a) + \epsilon_1 \epsilon) = \sum_{k=0}^{t} \lambda_k (T_{d_j}(a) + \epsilon_1 \epsilon)^k + \epsilon \sum_{k=0}^{t} \epsilon_k (T_{d_j}(a) + \epsilon_1 \epsilon)^k \approx 0.$$  

Taking the Taylor expansion about the point $T_{d_j}(a)$ gives

$$\sum_{k=0}^{t} \frac{1}{k!} \Gamma^{(k)}(T_{d_j}(a)) \cdot (\epsilon_1 \epsilon)^k + \epsilon \sum_{k=0}^{t} \frac{1}{k!} \Gamma^{(k)}(T_{d_j}(a)) \cdot (\epsilon_1 \epsilon)^k \approx 0.$$  

Since $A(T_{d_j}(a)) = 0$ and $|T_{d_j}(a)| \leq 1$, and considering only the first order terms in $\epsilon$, we have $A^{(1)}(T_{d_j}(a)) \cdot \epsilon_1 \epsilon + \epsilon \Gamma(T_{d_j}(a)) \approx 0$ and so

$$|\epsilon_1| \approx \left| \frac{\Gamma(T_{d_j}(a))}{A^{(1)}(T_{d_j}(a))} \right| \leq \frac{\sum_{k=0}^{t} |\epsilon_k|}{\prod_{k \neq j} |T_{d_j}(a) - T_{d_k}(a)|}.$$  

Therefore,

$$|T_{d_j}(a) - \tilde{y}_j| < \frac{\epsilon \cdot \sum_{k=0}^{t} |\epsilon_k|}{\prod_{j \neq k} |T_{d_j}(a) - T_{d_k}(a)|} + O(\epsilon^2).$$

The size of $|\prod_{j \neq k} |T_{d_j}(a) - T_{d_k}(a)|$ is related to the condition of the Vandermonde system $\mathcal{V}$. 

3.3 Solving the Vandermonde-like System

The coefficients $c_j$ in (1) can be recovered by solving the Vandermonde-like system in (5). Based on the factorization in (7) we have

$$\|W^{-1}\| = \|V^{-1}\|\|L^{-1}\| \leq K \cdot \|V^{-1}\|,$$

where $K = 9$ as in Lemma 2. The Vandermonde matrix $V$ has all its nodes $T_d(a)$ at real values. When the $T_d(a)$ are located symmetrically with respect to the origin, the lower bound for the condition number of such a $t \times t$ system $V$ grows exponentially in $t$. This happens, for example, when values of $T_d(a) = 1 - 2(j - 1)/(n - 1)$ for $j = 1, 2, \ldots, n$, are equidistant points between $-1$ and 1. This phenomenon also occurs when the $d_j$’s are evenly distributed between 0 and $m$, where $m = N/2$ when $N$ is even and $m = (N + 1)/2$ when $N$ is odd. If all the nodes in $V$ are positive, then it is known that condition number of $V$ is bounded from below by a constant times $2^t$ [4,1].

4 Sparse Chebyshev Interpolation Using Generalized Eigenvalues

An important variant of Prony’s method proposed by Golub, Milanfar, and Varah [7] combines solving the Hankel system and finding roots of the corresponding generating polynomial into a single generalized eigenvalue problem (see also [11]). The advantage of this reformulation is that there are well-established, numerically stable methods for the solving the generalized eigenvalue problem.

We can apply the generalized eigenvalue reformulation to the associated symmetric Hankel-plus-Toeplitz system in our method. As a result, Steps 2.1 and 2.2 in FPSparseChebyshevInterp can be combined into the procedure for solving a generalized eigenvalue problem.

From the Hankel-plus-Toeplitz system $A$ in (4), we define

$$A_1 = \begin{bmatrix}
2\alpha_1 & \alpha_0 & \ldots & \alpha_t & \alpha_{t-1} \\
2\alpha_2 & \alpha_1 & \ldots & \alpha_{t+1} & \alpha_{t-2} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
2\alpha_t & \alpha_{t+1} & \ldots & \alpha_{2t-1} & \alpha_1
\end{bmatrix},$$

(11)

and set $Z = \text{diag}(T_{d_1}(a), \ldots, T_{d_t}(a))$. Then for the Vandermonde-like system $W$ in (5) and $D = \text{diag}(2c_1, \ldots, 2c_t)$, we have $A = WDW^T$ and $\frac{1}{2}(A_1 + A_1) = WDZW^T$. The values $T_{d_1}(a), T_{d_2}(a), \ldots, T_{d_t}(a)$ are solutions for $z$ in the generalized eigenvalue system

$$\frac{1}{2}(A_1 + A_1)v = zAv.$$  

(13)

Algorithm: GEVSparseChebyshevInterp

Given a black-box polynomial $p(x)$, and the number of non-zero terms $t$ of $p(x)$ in the Chebyshev basis, determine $c_j$ and $d_j$ such that $\sum_{j=1}^t c_j T_{d_j}(x)$ interpolates $p(x)$.

1. [Evaluate $p(T_k(a))$.] Choose an appropriate $a$, evaluate $\alpha_k = p(T_k(a))$ for $k = 0, 1, \ldots, 2t - 1$.

2. [Degrees $d_j$.] Obtain $T_{d_j}(a)$ by solving the generalized eigenvalue system (13), $d_j$ can be recovered from values of $T_{d_j}(a)$.

3. [Coefficients $c_j$.] Compute coefficients $c_j$. 

4.1 Sensitivity of the Generalized Eigenvalue Problem

We can apply the analysis of the generalized eigenvalue problem in [7, 6] to our Hankel-plus-Toeplitz system in the Step 2 of GEVSparseChebyshevInterp. For a given eigenvalue \( z_j \) and the associated eigenvector \( \nu \), suppose

\[
\frac{1}{2}((A_j^t + A_j) + \epsilon(\tilde{A}_j^t + \tilde{A}_j))(\nu + \epsilon\nu^{(1)} + \cdots) = (z_j + \epsilon z_j^{(1)} + \cdots)(A + \epsilon \tilde{A})(\nu + \epsilon\nu^{(1)} + \cdots)
\]

is an \( \epsilon \)-perturbation of our eigenvalue problem. Looking only at first order errors gives

\[
\left(\frac{1}{2}(A_j^t + A_j) - z_j A\right)\nu^{(1)} = (z_j^{(1)} A + z_j \tilde{A} - \frac{1}{2}(A_j^t + A_j))\nu.
\]

Both \( \frac{1}{2}(A_j^t + A_j) \) and \( A \) are symmetric. As a result, \( \nu \) is a left and right eigenvector at the same time. The left side of (14) is cancelled by multiplication on the left by \( \nu \) giving

\[
z_j^{(1)} = \frac{\nu^t((\frac{1}{2}(A_j^t + A_j) - z_j A)\nu)}{\nu^t A\nu}.
\]

Assuming the perturbations are of the same size as the precise value, that is, \( \|\tilde{A}\| = \|A\| \) and \( \|\frac{1}{2}(A_j^t + A_j)\| = \|\frac{1}{2}(A_j^t + A_j)\| \), and \( \nu \) is normalized as a unit vector, then (15) gives the error bound

\[
\frac{\|\frac{1}{2}(A_j^t + A_j)\| + |z_j|\|A\|}{\|\nu^t A\nu\|}.
\]

Notice that the columns of \((W^t)^{-1}\) give both the right and left eigenvectors of (13). If \( z_j \) is the eigenvalue corresponding to the \( j \)th column of \((W^t)^{-1}\), that is \( v_j = (W^t)^{-1}e_j \) for (13), then \( 1/|\nu^t A\nu| \) can be reduced to

\[
\frac{1}{|\nu^t A\nu|} = \frac{|v_j^t \nu v_j|^2}{|v_j^t W^t W^t v_j|} = \frac{|v_j|^2}{|v_j^t W^t W^t v_j|} = \frac{(W^t)^{-1}e_j|^2}{|e_j|} \leq \frac{|\nu^t L^{-1}e_j|^2}{|e_j|} \leq \frac{K^2 \cdot \|V^{-1}\|^2}{|e_j|}.
\]

Based on their similar structures, we may assume \( \|\frac{1}{2}(A_j^t + A_j)\| = \|A\| \). If \( a \) is chosen such that \( T_{dz}(a) \leq 1 \), then \( \|z_j\| \leq 1 \) and

\[
||z_j^{(1)}|| \leq \frac{2K^2 \cdot \|A\| \cdot \|V^{-1}\|^2}{|e_j|}.
\]

4.2 Computing the Coefficients \( c_j \)

\( \tilde{W} \) and \( W \) can be obtained. The coefficients \( c_j \) can then be computed since \( D = \text{diag}(2c_1, \ldots, 2c_r) = W^{-1} A(W^t)^{-1} \). On the other hand, if the \( T_{dz}(a) \) are obtained as generalized eigenvalues by the QZ algorithm, then the computed eigenvectors \( v_j \) can be used:

\[
c_j = (v_j^t A v_j)(H_{j,1})^2,
\]

where \( H = M^{-1}, M = (W^t)^{-1}S \) has \( v_j \) as columns, and \( S \) is a diagonal scaling matrix. The diagonals of \( S \) can be computed by solving \( [S]_{j,j}H_{j,1} = 1 \) (see [7]).

Coefficients \( c_j \) can also be recovered by solving the associated Vandermonde-like system (5), as described in the previous section.

5 Trigonometric Interpolation

In this section we present new algorithms to interpolate an approximate black-box function \( f \) as a linear combination of trigonometric functions of different periods. That is, we wish to find a representation of \( f \) as

\[
f(\theta) = \frac{A_0}{2} + \sum_{k=1}^{m} (A_k \cos k\theta + B_k \sin k\theta),
\]

in which many of the \( A_k \in \mathbb{R} \) and \( B_k \in \mathbb{R} \) are zero, with a small number of evaluations of the black box. We exhibit algorithms whose costs are proportional to the number of non-zero terms (i.e., the sparsity) in \( f \) when represented as above, and discuss their numerical sensitivity.
5.1 Cosine Interpolation

Recall that the $k$th Chebyshev polynomial gives polynomial relations between $\cos k\theta$ and $\cos \theta$: $\cos k\theta = T_k(\cos \theta)$. For the problem of interpolating a sum of cosine functions

$$g(\theta) = \sum_{j=1}^{t} A_j \cos h_j \theta \quad \text{with } h_1 < h_2 < \cdots < h_t,$$

the sparse Chebyshev polynomial interpolation algorithms in Sections 3 and 4 can be transformed easily into sparse cosine interpolation algorithms by considering $g(\cos^{-1} a) = p(a)$ for $a = \cos \phi$, where $\phi = 2\pi/N$ and $N \geq 2h_t$. We have $-1 \leq a \leq 1$ because $a = \cos \phi$.

By modifying the algorithm FPSparseChebyshevInterp of the previous section, we obtain the algorithm SparseCosineInterp:

Algorithm: SparseCosineInterp

Given a black box for $g(\theta) = \sum_{j=1}^{t} A_j \cos h_j \theta$, the number of cosine terms $t$ of $g(\theta)$, and an upper bound $M$ for the maximum period of a term in $g(\theta)$, find $A_1, \ldots, A_t \in \mathbb{R}$ and $h_1 < \ldots < h_t \leq M$ such that $g(\theta) = \sum_{j=1}^{t} A_j \cos h_j \theta$.

1. [Evaluate $g(k\phi)$.] Choose $\phi = 2\pi/N$ for $N \geq 2M$, evaluate $\alpha_k = g(k\phi)$ for $k = 0, 1, \ldots, 2t - 1$.
2. [Find periods $h_j$.]
   (2.1) Solve the symmetric Hankel-plus-Toeplitz system in equation (4).
   (2.2) Find roots of $A(z)$ to obtain $\cos h_j \phi$. The $h_j$ can then be recovered from values of $\cos h_j \phi$.
3. [Find coefficients $A_j$.] Determine coefficients $A_1, \ldots, A_t$ by solving the transposed Vandermonde-like system

$$\begin{bmatrix}
1 \\
\cos(h_1 \phi) & \cdots & \cos(h_t \phi) \\
\vdots & \ddots & \vdots \\
\cos(h_1(t-1) \phi) & \cdots & \cos(h_t(t-1) \phi)
\end{bmatrix}
\begin{bmatrix}
A_1 \\
A_2 \\
\vdots \\
A_t
\end{bmatrix}
= \begin{bmatrix}
g(0) \\
g(\phi) \\
\vdots \\
g((t-1)\phi)
\end{bmatrix}. \tag{17}
$$

As in Section 4, we can combine the explicit formation of $A(z)$ and finding its roots into a single generalized eigenvalue problem as in GEVSparseChebyshevInterp. This should improve the numerical stability of the algorithm. We replace Steps 2.1 and 2.2 in the above algorithm, as follows:

Algorithm: GEVSparseCosineInterp

Given a black box $g(\theta) = \sum_{j=1}^{t} A_j \cos h_j \theta$, the number of cosine terms $t$ of $g(\theta)$, and an upper bound $M$ for the maximum period of a term in $g(\theta)$, find $A_1, \ldots, A_t \in \mathbb{R}$ and $h_1 < \ldots < h_t \leq M$ such that $g(\theta) = \sum_{j=1}^{t} A_j \cos h_j \theta$.

1. [Evaluate $g(k\phi)$.] Choose $\phi = 2\pi/N$ for $N \geq 2M$, evaluate $\alpha_k = g(k\phi)$ for $k = 0, 1, \ldots, 2t - 1$.
2. [Find periods $h_j$.] Obtain $\cos h_j \phi$ via solving the associated generalized eigenvalue system as in (13). The periods $h_j$ can be recovered from values of $\cos h_j \phi$.
3. [Find coefficients $A_j$.] Compute coefficients $A_j$.

5.2 Sparse Interpolation for Trigonometric Functions

We now consider the interpolation of a sparse linear combination of sine and cosine functions:

$$f(\theta) = \sum_{j=1}^{t_1} A_j \cos h_j \theta + \sum_{j=1}^{t_2} B_j \sin k_j \theta. \tag{18}$$
When \( t_1 = t_2 \) and \( h_j = k_j \), \( f(\theta) \) can interpolated through a variant of Prony’s method that requires \( 3t_1 \) evaluations [8, pp. 382-386], though Hildebrand notes that this algorithm suffers from the numerical instability generally associated with Prony’s method.

Alternatively, the interpolation of \( f(\theta) \) can be transformed into the problem of finding an associated phase polynomial that is a sum of exponential functions (see, e.g., [15]). Let \( N \) be chosen as either \( 2m + 1 \) (odd) or \( 2m \) (even), and \( \phi_\ell = 2\pi \ell / N \) for \( \ell = 0, \ldots, N - 1 \) over the interval \([0, 2\pi]\). It is easily derived that for \( \ell = 0, \ldots, N - 1 \),
\[
\cos h_j \phi_\ell = \frac{e^{h_j \phi_\ell} + e^{(N-h_j)\phi_\ell}}{2} \quad \text{and} \quad \sin k_j \phi_\ell = \frac{e^{k_j \phi_\ell} - e^{(N-k_j)\phi_\ell}}{2i}.
\]

The phase polynomial \( p(\theta) \) for \( f(\theta) \) is defined by
\[
p(\theta) = \sum_{\ell=0}^{N-1} \beta_\ell e^{\ell i \theta},
\]
with coefficients \( \beta_\ell \) as follows:

- If \( N = 2m + 1 \), then for \( k = 1, \ldots, m \),
  \[
  \beta_0 = \frac{A_0}{2}, \beta_k = \frac{1}{2}(A_k - iB_k), \quad \beta_{N-k} = \frac{1}{2}(A_k + iB_k).
  \]
- If \( N = 2m \), then \( \beta_m = A_m/2 \), and for \( k = 1, \ldots, m - 1 \),
  \[
  \beta_0 = \frac{A_0}{2}, \beta_k = \frac{1}{2}(A_k - iB_k), \quad \beta_{N-k} = \frac{1}{2}(A_k + iB_k).
  \]

While \( p(\theta) = f(\theta) \) need not to hold everywhere, by definition \( p(\phi_\ell) = f(\phi_\ell) \), and \( p(\theta) \) can be interpolated from \( f(\phi_0), f(\phi_1), \ldots, f(\phi_{N-1}) \). Once the phase polynomial \( p(\theta) \) is found, coefficients \( A_j \) and \( B_j \) in \( f(\theta) \) can be recovered according to their relations with \( \beta_\ell \).

We notice that the phase polynomial can be interpolated through sparse methods that are similar to sparse polynomial interpolation on the unit circle [6]. Hence, the generalized eigenvalue approach can be used in the interpolation of the phase polynomial \( p(\theta) \) and the corresponding trigonometric function \( f(\theta) \). This provides considerable numerical stability. We note that it does so at the expense of moving to computations over the complex numbers.

On the other hand, taking advantage of the facts that \( g_1(\theta) \) is odd and \( g_2(\theta) \) is even (cf. [13]), when \( t_1 = t_2, h_j = k_j, \) and \( A_j \neq 0 \) in (18) for \( 1 \leq k \leq t_1 \), either \texttt{SparseCosineInterp} or \texttt{GEVSparseCosineInterp} can be used to interpolate the cosine component of \( f(\theta) \) from the following evaluation:
\[
g_1(\theta) = \frac{1}{2}(f(\theta) + f(-\theta)).
\]

Once \( g_1(\theta) \) is interpolated, the \( k_j \) in \( g_2(\theta) \) are also recovered. The coefficients \( B_j \) can be computed by solving
\[
\begin{bmatrix}
1 & \ldots & 1 \\
\sin(k_1 \phi) & \ldots & \sin(k_{t_2} \phi) \\
\vdots & \ddots & \vdots \\
\sin(k_1(t_2 - 1) \phi) & \ldots & \sin(k_{t_2}(t_2 - 1) \phi)
\end{bmatrix}
\begin{bmatrix}
B_1 \\
B_2 \\
\vdots \\
B_{t_2}
\end{bmatrix}
= 
\begin{bmatrix}
g_2(0) \\
g_2(\phi) \\
\vdots \\
g_2((t_2 - 1) \phi)
\end{bmatrix}.
\]

The values for \( g_2(j \cdot \phi) \) are obtained from evaluating:
\[
g_2(\theta) = \frac{1}{2}(f(\theta) - f(-\theta)).
\]

### 5.3 Multivariate Case

We can also extend this trigonometric interpolation to the multivariate case. For sparse polynomial interpolation, multivariate methods for floating-point arithmetic are developed in [6]. This may be
applicable to multi-dimensional Fourier series, as it applies to image processing [12]. We consider
the following case of multivariate trigonometric interpolation:
\[
f(\theta_1, \ldots, \theta_n) = \sum_{j=1}^{t_1} A_j \cos(h_{1,j} \theta_1 + \ldots + h_{n,j} \theta_n) + \sum_{j=1}^{t_2} B_j \sin(k_{1,j} \theta_1 + \ldots + k_{n,j} \theta_n),
\]
when \(h_{i,j}\) and \(k_{i,j}\) are all integers.

If \(f\) is interpolated through its associated phase polynomial, the multivariate method developed
in [6] can be directly implemented. Here we apply a similar strategy for interpolating a sum of
cosine functions:
\[
g(\theta_1, \ldots, \theta_n) = \sum_{j=1}^{t} A_{h_j} \cos(h_{1,j} \theta_1 + \ldots + h_{n,j} \theta_n),
\]
with \(h_{1,j} \leq m_1, \ldots, h_{n,j} \leq m_n\).

Let \(p_1, \ldots, p_n \in \mathbb{Z}_{\geq 0}\) be pairwise relatively prime and \(p_j > m_j\) for \(1 \leq k \leq n\). Consider
interpolating \(g\) at \(\omega_k = 2\pi/p_k\). Set \(m = p_1 \cdots p_n\) and \(\omega = 2\pi/m\), then \(\omega_k = m/p_k\) for \(1 \leq k \leq n\).

In \(g(\omega_1, \ldots, \omega_n)\), each term \(\cos(h_{1,j} \theta_1 + \ldots + h_{n,j} \theta_n)\) is mapped to value \(\cos(h_{1,j} 2\pi/p_1 + \ldots + h_{n,j} 2\pi/p_n) = \cos(h_{1,j} 2\pi/m)\). The period for each variable \((h_{j_1}, \ldots, h_{j_k}) \in \mathbb{Z}_{\geq 0}^n\) can be uniquely
determined by the Chinese remainder algorithm (cf. [5]). That is, \(d_j \mod p_k \equiv d_{j_k}\) for \(1 \leq k \leq n\), and
\[
h_j = h_{j_1} \cdot \left(\frac{m}{p_1}\right) + \cdots + h_{j_n} \cdot \left(\frac{m}{p_n}\right). \tag{21}
\]

6 Conclusions and Future Works

We develop sparse Chebyshev interpolation algorithms in floating-point arithmetic. We give
formulations based on a Prony-like root-finding method, and on a more numerically stable generalized
eigenvalue approach. Based on the relations between cosine functions and Chebyshev polynomials,
we extend these interpolation results to sparse trigonometric functions. We also show how these
can be improved numerically through the use of generalized eigenvalue solvers. Finally, we give a
method for a sparse, multivariate trigonometric interpolation.

We have implemented FPSparsChebyshevInterp and GEVSparseChebyshevInterp in Maple\(^1\).
Currently we are conducting extensive testing and numerical experiments. We are further studying
the numerical sensitivity, especially in the situation when only an inexact upper is supplied for the
number of terms in the input, and investigating connections to Fourier series.

References

1. BECKERMAN, B. The condition number of real Vandermonde, Krylov and positive definite Hankel
2. BREZINSKI, C. History of Continued Fractions and Padé Approximants. Springer Verlag, Heidelberg,
Germany, 1991.
3. GAUTSCHI, W. On inverses of Vandermonde and confluent Vandermonde matrices. Numerische
4. GAUTSCHI, W., and INGLESE, G. Lower bounds for the condition numbers of Vandermonde matrices.
Publ., Boston, Massachusetts, USA, 1992.
6. GIESBRECHT, M., LABAHN, G., and LEE, W.-S. Symbolic-numeric sparse interpolation of multivariate
polynomials. In Proc. Ninth Rhine Workshop on Computer Algebra (RWCA’04), University of
7. GOLUB, G. H., MILANFAR, P., and VARAH, J. A stable numerical method for inverting shape from

\(^1\) Maple code is available at http://scg.uwaterloo.ca/~ws2lee/software/sparsechebyshev.