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

Mark Giesbrecht and George Labahn
School of Computer Science, University of Waterloo
Waterloo, Ontario N2L 3G1, Canada

and
Wen-shin Lee
INRIA, GALAAD
BP 93, 06902 Sophia-Antipolis, France

October 12, 2005

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. As well, we 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. Let $T_k(x)$ denote the k-th Chebyshev polynomial of the first kind:

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

Any polynomial $p(x)$ can be written in the Chebyshev basis as

$$p(x) = \sum_{j=1}^{t} c_j T_{d_j}(x), \tag{1.1}$$

where $0 \leq d_1 < d_2 < \cdots < d_t$ and $c_1, \ldots, c_t \in \mathbb{R}$. Lakshman and Saunders [10] give an algorithm (using exact arithmetic in $\mathbb{Q}[x]$) which interpolates the Chebyshev representation of a black-box polynomial from a small number of evaluations. Its cost (the number of black-box evaluations plus auxiliary field operations) is polynomial in the sparsity t of 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 [10]. The other is obtained by solving a generalized eigenvalue problem. Both approaches may be regarded as generalizations of symbolic-numeric sparse polynomial interpolations in the standard power basis [9].

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

$$f(\theta) = \frac{A_0}{2} + \sum_{k=1}^{m} (A_k \cos k\theta + B_k \sin k\theta), \tag{1.2}$$

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 $f(\theta)$ 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 [7], Section 9). The cost of such methods depends on the maximum number of terms in the target function (i.e., on m in (1.2)). 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 polynomial in m.

By a variant of Prony's method [2, 14], the interpolation of a trigonometric function $f(x)$ can be sensitive to the sparsity, the number of non-zero A_k, B_k in (1.2) [7, pp. 382-386]. By combining this with a connection between Prony's method and Ben-Or/Tiwari sparse interpolation observed in [9], 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

(1.1). We define the polynomial

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

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

$$\sum_{j=0}^{t-1} \lambda_j(\alpha_{j+i} + \alpha_{|j-i|}) = -(\alpha_{t+i} + \alpha_{|t-i|}) \text{ for } i \geq 0. \tag{2.1}$$

Relations in (2.1) give the following $t \times t$ symmetric Hankel-plus-Toeplitz system:

$$\underbrace{\begin{bmatrix} 2\alpha_0 & 2\alpha_1 & \dots & 2\alpha_{t-1} \\ 2\alpha_1 & \alpha_2 + \alpha_0 & \dots & \alpha_t + \alpha_{t-2} \\ \vdots & \vdots & \ddots & \vdots \\ 2\alpha_{t-1} & \alpha_t + \alpha_{t-2} & \dots & \alpha_{2t-2} + \alpha_0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_{t-1} \end{bmatrix} = - \begin{bmatrix} 2\alpha_t \\ \alpha_{t+1} + \alpha_{t-1} \\ \vdots \\ \alpha_{2t-1} + \alpha_1 \end{bmatrix}. \tag{2.2}$$

By showing that $\mathcal{A}$ is non-singular [10, see Lemma 6], Lakshman and Saunders give a sparse polynomial interpolation algorithm in the Chebyshev basis using 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 $c_1, \dots, c_t \in \mathbb{R}$ and $d_1, \dots, d_t \in \mathbb{Z}_{\geq 0}$ 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 $k = 0, 1, \dots, 2t-1$.

(2) [Degrees d_j.]

(2.1) Solve the symmetric Hankel-plus-Toeplitz system in (2.2) to obtain $\Lambda(z)$.

(2.2) Find all roots of $\Lambda(z)$ to obtain $T_{d_1}(a), \dots, T_{d_t}(a)$. The values of d_j can be recovered from $T_{d_j}(a)$ for $1 \leq j \leq t$.

(3) [Coefficients c_j.] Coefficients c_j can be obtained by solving the transposed Vandermonde-like system:

$$\underbrace{\begin{bmatrix} 1 & \dots & 1 \\ T_{d_1}(T_0(a)) & \dots & T_{d_t}(T_0(a)) \\ \vdots & \ddots & \vdots \\ T_{d_1}(T_{t-1}(a)) & \dots & T_{d_t}(T_{t-1}(a)) \end{bmatrix}}_{\mathcal{W}} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_t \end{bmatrix} = \begin{bmatrix} p(T_0(a)) \\ p(T_1(a)) \\ \vdots \\ p(T_{t-1}(a)) \end{bmatrix}. \tag{2.3}$$

In the algorithm SparseChebyshevInterp, the target polynomial is evaluated at $T_k(a)$ for $a > 1$ because the $T_k(a)$ 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 $\mathcal{A}$ in (2.2) are guaranteed [10]. 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 SparseChebyshevInterp, the matrix $\mathcal{A}$ will be non-singular and the algorithm will work as specified. In what follows we will examine the numerical conditioning of $\mathcal{A}$, and the stability of the entire algorithm.

3 Numerical issues with 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 $\Lambda(z)$, and in solving the Vandermonde-like system. That is, Steps 2.1, 2.2, and 3 in SparseChebyshevInterp.

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: `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 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 scales of powers of a can contribute detrimentally to the poor condition 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 $\mathcal{A}$, 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 kth degree polynomial in $T_d(x)$ with leading coefficient 2^{k-1}.

Lemma 3.1. *For $k \geq 1$,*

$$T_{kd}(x) = 2^{k-1} T_d^k(x) + \sum_{j=0}^{k-1} \gamma_j T_d^j(x) \tag{3.1}$$

See [8, Lemma 10].

Based on Lemma 3.1, the Vandermonde-like $\mathcal{W}$ in (2.3) can be factored as a product of a lower triangular matrix $\mathcal{L}$ and a Vandermonde matrix $\mathcal{V}$:

$$\mathcal{W} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & \vdots \\ * & * & 2 & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ * & * & * & \dots & 2^{t-2} \end{bmatrix} \begin{bmatrix} 1 & \dots & 1 \\ T_{d_1}(a) & \dots & T_{d_t}(a) \\ \vdots & \ddots & \vdots \\ T_{d_1}^{t-1}(a) & \dots & T_{d_t}^{t-1}(a) \end{bmatrix} = \mathcal{L}\mathcal{V}. \tag{3.2}$$

With $\mathcal{D} = \mathrm{diag}(2c_1, \dots, 2c_t)$, our factorization of $\mathcal{A}$ follows (cf. [8, 10]):

$$\mathcal{A} = \mathcal{W}\mathcal{D}\mathcal{W}^{\mathrm{Tr}} = \mathcal{L}\mathcal{V}\mathcal{D}(\mathcal{L}\mathcal{V})^{\mathrm{Tr}} = \mathcal{L}(\mathcal{V}\mathcal{D}\mathcal{V}^{\mathrm{Tr}})\mathcal{L}^{\mathrm{Tr}}. \tag{3.3}$$

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

Theorem 3.2. *Let $\mathcal{A}$ be the Hankel-plus-Toeplitz matrix generated from Step 2.1 of* `FPSparseChebyshevInterp`. *Then*

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

Proof. Consider the factorization of $\mathcal{A}$ in (3.3). Let D_j be the matrix derived from $\mathcal{D}$ by using 0 to replace $2c_j$ in the diagonal. Then the matrix $\mathcal{L}\mathcal{V}D_j\mathcal{V}^{\mathrm{Tr}}\mathcal{L}^{\mathrm{Tr}}$ is singular for $1 \leq j \leq t$, and we have

$$\begin{aligned} \frac{1}{\|\mathcal{A}^{-1}\|} &= \min\{\|\mathcal{A} - H\|,\ H \text{ singular }\} \\ &\leq \min\{\|\mathcal{A} - \mathcal{L}\mathcal{V}D_j\mathcal{V}^{\mathrm{Tr}}\mathcal{L}^{\mathrm{Tr}}\|\} \leq \|\mathcal{L}\|^2 \cdot \|\mathcal{V}\|^2 \cdot \min |2c_j|. \end{aligned}$$

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

$$\begin{aligned} \|\mathcal{A}^{-1}\| &\leq \|\mathcal{L}^{-1}\|^2 \|\mathcal{V}^{-1}\|^2 \|D^{-1}\| \\ &\leq \|\mathcal{L}^{-1}\|^2 \|\mathcal{V}^{-1}\|^2 \cdot \sum_{j=1}^{t} \|D^{-1}e_j\| \leq \|\mathcal{L}^{-1}\|^2 \cdot \|\mathcal{V}^{-1}\|^2 \cdot \max_j \frac{1}{|2c_j|}. \end{aligned}$$

□

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

Lemma 3.3. *For $\mathcal{L}$ as in* (3.2) *we have*

$$\|\mathcal{L}\| \leq t \cdot 2^{t-1} \quad \text{and} \quad \|\mathcal{L}^{-1}\| \leq K := 17.$$

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

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

The coefficient γ_j in (3.1) is either 0 or a $(t-2)$-th degree polynomial in 2, that is

$$\gamma_j = \sum_{l=0}^{k-2} q_l 2^l \text{ with } q_l \in \{-1, 0, 1\}. \tag{3.4}$$

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

We now bound the norm of $\mathcal{L}^{-1}$. The first two rows of matrix $\mathcal{L}^{-1}$ are $[1, 0, \ldots, 0]$ and $[0, 1, 0, \ldots, 0]$. We consider the kth row in the $t \times t$ matrix $\mathcal{L}^{-1}$: $\mathcal{L}_k^{-1} = [l_{k,1}^{-1}, \ldots, l_{k,k}^{-1}, 0, \ldots, 0]$ for $k > 2$. Recall that the entries in the jth row of $\mathcal{L}$ are polynomials in the constant 2 of degree no more that $j-2$. Combining this fact with $l_{k,k}^{-1} = 1/2^{k-2}$, we have

$$l_{k,i}^{-1} \leq \frac{\sum_{j=0}^{k-i} 2^j}{2^{k-2}} \leq \frac{2^{k-i+1}}{2^{k-2}} = 2^{3-i} \text{ for } 1 \leq i < k,$$

and

$$\|\mathcal{L}_k^{-1}\| \leq \sum_{i=1}^{k-1} 2^{3-i} + \frac{1}{2^{k-2}} \leq 2^4 + 1 \text{ for } 2 < k \leq t.$$

□

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

Lemma 3.4. *Let $\mathcal{V}$ be the Vandermonde matrix in* (3.2)*, then*

$$\|\mathcal{V}^{-1}\| \leq \frac{2^{t-1}}{\min_k \prod_{j=1, j\neq k}^{t} |T_{d_k}(a) - T_{d_j}(a)|}. \tag{3.5}$$

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

Lemma 3.5. *Let $\mathcal{A}$ be the Hankel-plus-Toeplitz matrix generated from the Step 2.1 of* `FPSparseChebyshevInterp`*, then*

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

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

3.2 Root finding for the polynomial $\Lambda(z)$

We now consider Step 2.2 in algorithm `FPSparseChebyshevInterp`, in which we find the roots of the polynomial $\Lambda(z)$, 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 $\Lambda(z) = \prod_{j=1}^{t}(z - T_{d_j}(a))$ 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 $\Lambda(z) + \epsilon\Gamma(z)$, a perturbation of Λ, where $\Gamma(z) = \varepsilon_t z^t + \varepsilon_{t-1} z^{t-1} + \cdots + \varepsilon_0 \in \mathbb{R}[z]$, and $\epsilon > 0$ can be thought of as "small". Then $\tilde{y}_j = T_{d_j}(a) + \sum_{k=1}^{\infty} \zeta_k \epsilon^k \approx T_{d_j}(a) + \zeta_1\epsilon$ for some $\zeta_1, \zeta_2, \ldots$, and

$$\begin{aligned}
&\Lambda(T_{d_j}(a) + \zeta_1\epsilon) + \epsilon\Gamma(T_{d_j}(a) + \zeta_1\epsilon) \\
&\quad = \sum_{k=0}^{t} \lambda_k (T_{d_j}(a) + \zeta_1\epsilon)^k + \epsilon \sum_{k=0}^{t} \varepsilon_k (T_{d_j}(a) + \zeta_1\epsilon)^k \approx 0.
\end{aligned}$$

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

$$\sum_{k=0}^{t} \frac{1}{k!}\Lambda^{(k)}(T_{d_j}(a)) \cdot (\zeta_1\epsilon)^k + \epsilon \sum_{k=0}^{t} \frac{1}{k!}\Gamma^{(k)}(T_{d_j}(a)) \cdot (\zeta_1\epsilon)^k \approx 0.$$

Since $\Lambda(T_{d_j}(a)) = 0$ and $|T_{d_j}(a)| \le 1$, and considering only the first order terms in ϵ, we have $\Lambda^{(1)}(T_{d_j}(a)) \cdot \zeta_1\epsilon + \epsilon\Gamma(T_{d_j}(a)) \approx 0$ and so

$$|\zeta_1| \approx \left| \frac{\Gamma(T_{d_j}(a))}{\Lambda^{(1)}(T_{d_j}(a))} \right| \le \frac{\sum_{k=0}^{t} |\varepsilon_k|}{|\prod_{k \ne j}(T_{d_j}(a) - T_{d_k}(a))|}.$$

Therefore,

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

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

3.3 Solving the Vandermonde-like system

The coefficients c_j in (1.1) can be recovered by solving the Vandermonde-like system in (2.3). Based on the factorization in (3.2) we have

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

where $K = 17$ as in Lemma 3.3. The Vandermonde matrix $\mathcal{V}$ has all its nodes $T_{d_j}(a)$ at real values. When the $T_{d_j}(a)$ are located symmetrically with respect to the origin, then the lower bound for the condition number of such a $t \times t$

system $\mathcal{V}$ grows exponentially in t. This happens, for example, when values of $T_{d_j}(a) = 1 - 2(j-1)/(n-1)$ 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 $\mathcal{V}$ are positive, then it is known that condition number of $\mathcal{V}$ 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 [6] combines solving the Hankel system and finding roots of the corresponding generating polynomial into into a single generalized eigenvalue problem (see also [11]). The advantage of this reformulation is that there are well-estalished, 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 $\mathcal{A}$ in (2.2), we define

$$\mathcal{A}_{\uparrow} = \begin{bmatrix} 2\alpha_1 & \alpha_2 + \alpha_0 & \ldots & \alpha_t + \alpha_{t-2} \\ 2\alpha_2 & \alpha_3 + \alpha_1 & \ldots & \alpha_{t+1} + \alpha_{t-3} \\ \vdots & \vdots & \ddots & \vdots \\ 2\alpha_t & \alpha_{t+1} + \alpha_{t-1} & \ldots & \alpha_{2t-1} + \alpha_1 \end{bmatrix}, \tag{4.1}$$

$$\mathcal{A}_{\downarrow} = \begin{bmatrix} 2\alpha_1 & \alpha_2 + \alpha_0 & \ldots & \alpha_t + \alpha_{t-2} \\ 2\alpha_0 & 2\alpha_1 & \ldots & 2\alpha_{t-1} \\ 2\alpha_1 & \alpha_2 + \alpha_0 & \ldots & \alpha_t + \alpha_{t-2} \\ \vdots & \vdots & & \vdots \\ 2\alpha_{t-2} & \alpha_{t-1} + \alpha_{t-3} & \ldots & \alpha_{2t-3} + \alpha_1 \end{bmatrix}, \tag{4.2}$$

and set $Z = \mathrm{diag}(T_{d_1}(a), \ldots, T_{d_t}(a))$. Then for the Vandermonde-like system $\mathcal{W}$ in (2.3) and $\mathcal{D} = \mathrm{diag}(2c_1, \ldots, 2c_t)$, we have $\mathcal{A} = \mathcal{W}\mathcal{D}\mathcal{W}^{\mathrm{Tr}}$ and $\frac{1}{2}(\mathcal{A}_{\uparrow} + \mathcal{A}_{\downarrow}) = \mathcal{W}\mathcal{D}Z\mathcal{W}^{\mathrm{Tr}}$. The values $T_{d_1}(a)$, $T_{d_2}(a)$, …, $T_{d_t}(a)$ are solutions for z in the generalized eigenvalue system

$$\frac{1}{2}(\mathcal{A}_{\uparrow} + \mathcal{A}_{\downarrow})v = z\mathcal{A}v. \tag{4.3}$$

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 (4.3), 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 [6, 9] to our Hankel-plus-Toeplitz system in the Step 2 of `GEVSparseChebyshevInterp`. For a given eigenvalue z_j and the associated eigenvector ν, suppose

$$\begin{aligned}&\frac{1}{2}((\mathcal{A}_\uparrow + \mathcal{A}_\downarrow) + \epsilon(\hat{\mathcal{A}}_\uparrow + \hat{\mathcal{A}}_\downarrow))(\nu + \epsilon\nu^{(1)} + \cdots)\\ &\qquad = (z_j + \epsilon z_j^{(1)} + \cdots)(\mathcal{A} + \epsilon\hat{\mathcal{A}})(\nu + \epsilon\nu^{(1)} + \cdots)\end{aligned}$$

is an ϵ-perturbation of our eigenvalue problem. Looking only at first order errors gives

$$(\frac{1}{2}(\mathcal{A}_\uparrow + \mathcal{A}_\downarrow) - z_j\mathcal{A})\nu^{(1)} = (z_j^{(1)}\mathcal{A} + z_j\hat{\mathcal{A}} - \frac{1}{2}(\hat{\mathcal{A}}_\uparrow + \hat{\mathcal{A}}_\downarrow))\nu. \tag{4.4}$$

Note that both $\frac{1}{2}(\mathcal{A}_\uparrow + \mathcal{A}_\downarrow)$ and $\mathcal{A}$ are symmetric, therefore ν is a left and right eigenvector at the same time. The left side of (4.4) is cancelled by multiplication on the left by ν^{Tr} giving

$$z_j^{(1)} = \frac{\nu^{\mathrm{Tr}}(\frac{1}{2}(\hat{\mathcal{A}}_\uparrow + \hat{\mathcal{A}}_\downarrow) - z_j\hat{\mathcal{A}})\nu}{\nu^{\mathrm{Tr}}\mathcal{A}\nu}. \tag{4.5}$$

Assuming the perturbations are of the same size as the precise value, that is, $\|\hat{\mathcal{A}}\| = \|\mathcal{A}\|$ and $\|\frac{1}{2}(\hat{\mathcal{A}}_\uparrow + \hat{\mathcal{A}}_\downarrow)\| = \|\frac{1}{2}(\mathcal{A}_\uparrow + \mathcal{A}_\downarrow)\|$, and ν is normalized as a unit vector, then (4.5) gives the error bound

$$\frac{\|\frac{1}{2}(\mathcal{A}_\uparrow + \mathcal{A}_\downarrow)\| + |z_j|\|\mathcal{A}\|}{|\nu^{\mathrm{Tr}}\mathcal{A}\nu|}.$$

Notice that the columns of $(\mathcal{W}^{\mathrm{Tr}})^{-1}$ give both the right and left eigenvectors of (4.3). If z_j is the eigenvalue corresponding to the jth column of $(\mathcal{W}^{\mathrm{Tr}})^{-1}$, that is $v_j = (\mathcal{W}^{\mathrm{Tr}})^{-1}e_j$ for (4.3), then $1/|\nu^{\mathrm{Tr}}\mathcal{A}\nu|$ can be reduced to

$$\begin{aligned}\frac{1}{|\nu^{\mathrm{Tr}}\mathcal{A}\nu|} &= \frac{|v_j^{\mathrm{Tr}} \cdot v_j|^2}{|v_j^{\mathrm{Tr}}\mathcal{A}v_j|} = \frac{\|v_j\|^2}{|v_j^{\mathrm{Tr}} \cdot \mathcal{W} \cdot \mathcal{D} \cdot \mathcal{W}^{\mathrm{Tr}} \cdot v_j|} = \frac{\|(\mathcal{W}^{\mathrm{Tr}})^{-1}e_j\|^2}{|c_j|}\\ &\le \frac{\|\mathcal{V}^{-1}\|^2 \cdot \|\mathcal{L}^{-1}\|^2}{|c_j|} \le \frac{K^2 \cdot \|\mathcal{V}^{-1}\|^2}{|c_j|}.\end{aligned}$$

Based on their similar structures, we may assume $\|\frac{1}{2}(\mathcal{A}_\uparrow + \mathcal{A}_\downarrow)\| = \|\mathcal{A}\|$. If a is chosen such that $T_{d_j}(a) \le 1$, then $\|z_j\| \le 1$ and

$$\|z_j^{(1)}\| \le \frac{K^2 \cdot \|\mathcal{A}\| \cdot \|\mathcal{V}^{-1}\|^2}{|c_j|}. \tag{4.6}$$

4.2 Computing the coefficients c_j

From the computed $T_{d_j}(a)$, both $\mathcal{W}$ and $\mathcal{W}^{-1}$ can be obtained. The coefficients c_j can then be computed since $\mathcal{D} = \text{diag}(2c_1, \ldots, 2c_t) = \mathcal{W}^{-1}\mathcal{A}(\mathcal{W}^{\text{Tr}})^{-1}$. On the other hand, if the $T_{d_j}(a)$ are obtained as generalized eigenvalues by the QZ algorithm, then the computed eigenvectors ν_j can be used:

$$c_j = (\nu_j^{\text{Tr}}\mathcal{A}\nu_j)(H_{j,1})^2,$$

where $H = M^{-1}$, $M = (\mathcal{W}^{\text{Tr}})^{-1}S$ has ν_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 [6]).

Coefficients c_j can also be recovered by solving the associated Vandermonde-like system (2.3), 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 stability.

5.1 Cosine interpolation

Recall that the kth 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 consine 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 `SparseConsineInterp`:

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 $N \geq 2M$, and 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 (2.2).

(2.2) Find roots of $\Lambda(z)$ to obtains $\cos h_j\phi$. 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

$$\underbrace{\begin{bmatrix} 1 & \ldots & 1 \\ \cos(h_1\phi) & \ldots & \cos(h_t\phi) \\ \vdots & \ddots & \vdots \\ \cos(h_1(t-1)\phi) & \ldots & \cos(h_t(t-1)\phi) \end{bmatrix}}_{\mathcal{W}} \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_t \end{bmatrix} = \begin{bmatrix} g(0) \\ g(\phi) \\ \vdots \\ g((t-1)\phi) \end{bmatrix}. \quad (5.1)$$

As in Section 4, we can combine the explicit formation of $\Lambda(z)$ 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 (4.3). 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:

$$f(\theta) = \underbrace{\sum_{j=1}^{t_1} A_j \cos h_j\theta}_{g_1(\theta)} + \underbrace{\sum_{j=1}^{t_2} B_j \sin k_j\theta}_{g_2(\theta)} . \tag{5.2}$$

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 [7, 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 $\ell = 0, \ldots, N-1$,

$$\cos h_j\phi_\ell = \frac{e^{h_j i\phi_\ell} + e^{(N-h_j)i\phi_\ell}}{2} \quad \text{and} \quad \sin k_j\phi_\ell = \frac{e^{k_j i\phi_\ell} - e^{(N-k_j)i\phi_\ell}}{2i}.$$

The phase polynomial $p(\theta)$ for $f(\theta)$ is defined by

$$p(\theta) = \sum_{\ell=0}^{N-1} \beta_\ell e^{\ell i\theta}, \tag{5.3}$$

with coefficients β_ℓ 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 β_ℓ.

We notice that the phase polynomial can be interpolated through sparse methods that are similar to sparse polynomial interpolation on the unit circle [9]. 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 compuations 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 (5.2) for $1 \leq k \leq t_1$, `SparseCosineInterp` or `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

$$\begin{bmatrix} 1 & \dots & 1 \\ \sin(k_1\phi) & \dots & \sin(k_{t_2}\phi) \\ \vdots & \ddots & \vdots \\ \sin(k_1(t_2-1)\phi) & \dots & \sin(k_{t_2}(t_2-1)\phi) \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_{t_2} \end{bmatrix} = \begin{bmatrix} g_2(0) \\ g_2(\phi) \\ \vdots \\ g_2((t_2-1)\phi) \end{bmatrix}. \quad (5.4)$$

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 [9]. 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:

$$f(\theta_1, \dots, \theta_n) = \sum_{j=1}^{t_1} A_j \cos(h_{1,j}\theta_1 + \dots + h_{n,j}\theta_n) + \sum_{j=1}^{t_2} B_j \sin(k_{1,j}\theta_1 + \dots + k_{n,j}\theta_n),$$

when $h_{i,j}$ and $k_{i,j}$ are all integers.

If f is interpolated through its associated phase polynomial the method developed in [9] can be directly implemented. Here we apply a similar strategy for interpolating a sum of cosine functions:

$$g(\theta_1, \dots, \theta_n) = \sum_{j=1}^{t} A_{h_j} \cos(h_{1,j}\theta_1 + \dots + h_{n,j}\theta_n),$$

with $h_{1,j} \leq m_1, \dots, h_{n,j} \leq m_n$.

Let $p_1, \dots, p_n \in \mathbb{Z}_{\geq 0}$ 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 $g(\omega_1, \dots, \omega_n)$, each term $\cos(h_{1,j}\theta_1 + \dots + h_{n,j}\theta_n)$ is mapped to value $\cos(h_{1,j}2\pi/p_1 + \dots + h_{n,j}2\pi/p_n) = \cos(h_j 2\pi/m)$. The period for each variable

$(h_{j_1}, \ldots, h_{j_n}) \in \mathbb{Z}^n_{\geq 0}$ can be uniquely determined by the Chinese remainder algorithm (cf. [5]). That is, $d_j \bmod p_k \equiv d_{j_k}$ for $1 \leq k \leq n$, and

$$h_j = h_{j_1} \cdot \left(\frac{m}{p_1}\right) + \cdots + h_{j_n} \cdot \left(\frac{m}{p_n}\right). \tag{5.5}$$

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 studying further the numerical sensitivity, especially in the situation when only an inexact upper is supplied for the number of terms in the input.

References

[1] Beckermann, B. The condition number of real Vandermonde, Krylov and positive definite Hankel matrices. *Numeriche Mathematik 85* (2000), 553–577.

[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 Mathematik 4* (1962), 117–123.

[4] Gautschi, W., and Inglese, G. Lower bounds for the condition numbers of Vandermonde matrices. *Numerische Mathematik 52* (1988), 241–250.

[5] Geddes, K. O., Czapor, S. R., and Labahn, G. *Algorithms for Computer Algebra.* Kluwer Academic Publ., Boston, Massachusetts, USA, 1992.

[6] Golub, G. H., Milanfar, P., and Varah, J. A stable numerical method for inverting shape from moments. *SIAM J. Sci. Comput. 21*, 4 (1999), 1222–1243.

[7] Hildebrand, F. B. *Introduction to numerical analysis.* McGraw-Hill Book Co., New York, 1956.

[1] Maple code is available at http://scg.uwaterloo.ca/ ws2lee/software/sparsechebysev.

[8] Kaltofen, E., and Lee, W.-s. Early termination in sparse interpolation algorithms. *J. Symbolic Comput. 36*, 3–4 (2003), 365–400.

[9] Labahn, G., Giesbrecht, M., and Lee, W.-s. Symbolic-numeric sparse interpolation of multivariate polynomials. To be submitted, 2004.

[10] Lakshman Y. N., and Saunders, B. D. Sparse polynomial interpolation in non-standard bases. *SIAM J. Comput. 24*, 2 (1995), 387–397.

[11] Luk, F. T., and Vandevoorde, D. Decomposing a signal into a sum of exponentials. In *Interative Methods in Scientific Computing* (1997), pp. 329–357.

[12] Mallat, S. *A wavelet tour of signal processing*, 2 ed. Academic Press, London, 1998.

[13] Mansour, Y. Randomized approxmation and interpolation of sparse polynomials. *SIAM Journal on Computing 24*, 2 (1995), 357–368.

[14] Prony, R. Essai expérimental et analytique sur les lois de la Dilatabilité des fluides élastique et sur celles de la Force expansive de la vapeur de l'eau et de la vapeur de l'alkool, à différentes températures. *J. de l'École Polytechnique 1* (Floréal et Prairial III (1795)), 24–76. R. Prony is Gaspard(-Clair-François-Marie) Riche, baron de Prony.

[15] Stoer, J., and Bulirsch, R. *Introduction to numerical analysis*, 2 ed. New York, 1992.

[16] Wilkinson, J. H. *Rounding errors in algebraic processes.* Prentice-Hall, Englewood Cliffs, N.J., 1963.