FAST MULTI-DIMENSIONAL SCATTERED DATA APPROXIMATION WITH NEUMANN BOUNDARY CONDITIONS

DENIS GRISHIN AND THOMAS STROHMER[†]

Abstract. An important problem in applications is the approximation of a function f from a finite set of randomly scattered data $f(x_j)$. A common and powerful approach is to construct a trigonometric least squares approximation based on the set of exponentials $\{e^{2\pi ikx}\}$. This leads to fast numerical algorithms, but suffers from disturbing boundary effects due to the underlying periodicity assumption on the data, an assumption that is rarely satisfied in practice. To overcome this drawback we impose Neumann boundary conditions on the data. This implies the use of cosine polynomials $\cos(\pi kx)$ as basis functions. We show that scattered data approximation using cosine polynomials leads to a least squares problem involving certain Toeplitz+Hankel matrices. We derive estimates on the condition number of these matrices. Unlike other Toeplitz+Hankel matrices, the Toeplitz+Hankel matrices arising in our context cannot be diagonalized by the discrete cosine transform, but they still allow a fast matrix-vector multiplication via DCT which gives rise to fast conjugate gradient type algorithms. We show how the results can be generalized to higher dimensions. Finally we demonstrate the performance of the proposed method by applying it to a two-dimensional geophysical scattered data problem.

Key words. Trigonometric approximation, nonuniform sampling, discrete cosine transform, Toeplitz+Hankel matrix, block Toeplitz+Hankel matrix, conjugate gradient method.

AMS subject classifications. 65T40, 42A10, 65D15, 65F10, 68U10.

1. Introduction.

An ubiquitous problem in mathematics and in applications is the reconstruction or approximation of a function f from its non-uniformly spaced sampling values $s_j = f(x_j)$. Without further knowledge about f this is an ill-posed problem, since the subspace of functions h with $h(x_j) = s_j$ has always infinite dimension. Moreover in practice we are given only a finite number of samples $\{s_j\}_{j=1}^r$, which makes a complete reconstruction of f in general impossible, so the best we can hope for is to compute a good approximation to f. Fortunately in many practical situations the functions under consideration are not arbitrary, but possess some smoothness properties. For instance physics often implies that f is bandlimited. In this and many other cases a linear combination of trigonometric basis functions $\{e^{2\pi ikx}\}_{k\in\mathbb{Z}}$ often provides a good approximation to f. Other powerful models for scattered data approximation are based on radial basis functions and on shift-invraint systems [19].

Least squares approximation using exponentials as basis functions provides a tool that is general enough to be useful in a variety of situations where smooth functions are involved, while the algebraic structure of the functions $e^{2\pi ikx}$ is rich enough to give rise to fast and robust numerical algorithms to compute the approximation, cf. e.g. [17, 4, 3].

Arguably the main drawback of approximation by exponentials is the underlying periodicity assumption about the function to be approximated. To be more precise, let f be a smooth continuous function and let $\{f(x_j)\}_{j=1}^r$ be samples of f taken at the points $x_1 < \cdots < x_r$. Without loss of generality we assume that $x_1 = 0$ and $x_r = 1$. We want to approximate f on the sampling interval $[x_1, x_r) = [0, 1)$ by a trigonometric polynomial $p(x) = \sum_{k=-M}^{M} c_k e^{2\pi ikx}$ with $M < r/2$. If

[*]The second author has been supported in part by NSF grant 9973373.

[†]Department of Mathematics, University of California, Davis, CA 95616-8633, USA; Correspondence: strohmer@math.ucdavis.edu.

$f(0) = f(1)$ we can safely conclude from Weierstrass' theorem that a trigonometric polynomial of low degree will give a good approximation to f on the interval $[0,1)$. However if $f(0) \neq f(1)$ then this difference is felt as discontinuity by the approximating polynomial p. In fact standard Fourier analysis tells us that the coefficients $\{c_k\}_{k\in\mathbb{Z}}$ of p will at best decay like $o(1/k)$, thus a large degree M is required to obtain a reasonable approximation to f on $[0,1)$. However since in practice only a finite number of samples is available we may not be able to choose M sufficiently large to obtain a satisfactory approximation to f.

A standard method to enforce periodicity of f on $[0,1)$ is to multiply f with a smooth "window function" w which decays rapidly to zero at the boundaries of the sampling interval. However such a procedure can considerably reduce the interval in which the approximation is in agreement with the "non-windowed" sampling values $f(x_j)$. We could also try to reduce the unpleasant behavior caused by the boundary effects by choosing the period N of p slightly larger than the length of the sampling interval. Nevertheless, if $|f(0) - f(1)|$ is large we still need a polynomial of large degree to obtain a reasonable approximation to f on $[0,1)$. We also note that boundary effects become worse with increasing dimension.

Instead of extending f (respectively its samples $f(x_j)$) periodically across the boundaries of the sampling interval, we can apply Neumann boundary conditions, i.e., a symmetric extension across the end points of the sampling interval. This has the big advantage that we avoid the discontinuity at the boundaries. The Fourier coefficients of a continuous (periodic) function decay at least like $o1/k$ and at best like $o1/k^2$. Thus loosely spoken, the decay is one order of magnitude faster than compared to a periodic extension. This faster decay implies that a lower polynomial degree should suffice to obtain a good trigonometric approximation.[1]

If we extend the sampling values $f(x_j)_{j=1}^{r}$ symmetrically across the boundaries we obtain a sampling sequence that is periodic on the interval $[0,2)$ and symmetric with respect to the midpoint 1. To adapt the trigonometric basis functions to this situation we have to replace the exponentials $\{e^{2\pi ikx}\}_{k\in\mathbb{Z}}$ by the basis functions $\{\cos(\pi kx)\}_{k\in\mathbb{N}}$. The functions $\cos(\pi kx)$ are symmetric around 1 and periodic with respect to the interval $[0,2)$. The advantage when using cosine polynomials instead of exponentials is obvious from the discussion above: we reduce disturbing boundary effects, which results in a better approximation of the original function.

In the case of trigonometric approximation based on exponentials it has been shown that the least squares approximation can be formulated as hermitian positive definite Toeplitz system [4]. Gröchenig has derived explicit bounds for the condition number of the Toeplitz matrix that allow to estimate the stability and convergence of the involved numerical algorithms [7, 4]. Moreover all steps to compute and solve the Toeplitz system can be done quickly by (nonuniform) FFT-based methods.

The crucial questions that we will investigate in this paper are: Does the least squares approximation problem using cosine polynomials also give rise to a linear system of equation whose matrix has a nice structure? Can we find fast and robust numerical algorithms to solve the least squares problem? Can we give a priori estimates on the condition number of the matrix? Can we generalize the algorithm easily to higher dimensions? How does our approach perform for real world problems? This paper is devoted to clarify these questions.

The rest of the paper is organized as follows. In Section 2 we analyze the least squares approximation problem using cosine polynomials. We show that the resulting matrix has a certain Toeplitz+Hankel structure and derive estimates on the condition number of this matrix. In Section 3 we present a fast algorithm to solve the least squares problem using the conjugate gradient method and the discrete cosine transform (DCT). The generalization to the multi-dimensional case is described in Section 4. Finally in Section 5 we demonstrate the performance of the proposed method by applying it to a scattered data problem arising in geophysics.

The idea of using Neumann boundary conditions instead of periodic boundary conditions has turned out to be very fruitful in the context of image deblurring problems. In fact, the research

[1]This is exactly the reason why the (old) JPEG image compression algorithm uses the DCT instead of the DFT.

presented in this paper was inspired by the article *A fast algorithm for deblurring models with Neumann boundary conditions* by Michael Ng, Raymond Chan, and W.C. Tang [13].

2. Nonuniform sampling, cosine polynomials, and Toeplitz+Hankel matrices.

We start by defining the space $\boldsymbol{P}_M$ of cosine polynomials of maximal degree M as

$$\boldsymbol{P}_M = \left\{ p : p(x) = \frac{c_0}{\sqrt{2}} + \sum_{k=1}^{M} c_k \cos(\pi k x), c = \{c_k\}_{k=0}^{M} \in \mathbb{R}^{M+1} \right\}. \tag{2.1}$$

There are two reasons for the introduction of the $1/\sqrt{2}$-scaling factor of the coefficient c_0 in (2.1). The first reason is that we have the Parseval type identity

$$\|p\|_2^2 = \int_{-\infty}^{+\infty} |p(x)|^2 dx = \frac{c_0^2}{2} + \frac{1}{2}\sum_{k=1}^{M} c_k^2 = \frac{1}{2}\|c\|_2^2. \tag{2.2}$$

The second reason is increased stability of the numerical algorithms we are going to derive, as we will explain in the remark after Theorem 2.1.

Let us return to the approximation problem. Given sampling points[2] $\{x_j\}_{j=1}^r$ and sampling values $\{s_j\}_{j=1}^r$, we want to solve the least squares problem

$$\min_{p \in \boldsymbol{P}_M} \sum_{j=1}^{r} |p(x_j) - s_j|^2 w_j. \tag{2.3}$$

Here the $w_j > 0$ are weights which the user may choose at her convenience. Often the trivial choice $w_j = 1$ is sufficient. In other cases it is useful to choose the weights such that they compensate for irregularities in the sampling set, i.e., smaller weights are used in regions with high sampling density and larger weights in regions with few sampling points. In (2.3) we have assumed that the polynomial degree M is fixed. We will discuss the important question of how to determine the appropriate degree of the approximating polynomial in Section 3.

By defining the $r \times (M+1)$ Vandermonde-like matrix V via

$$V_{j,k} = \begin{cases} \frac{1}{\sqrt{2}}\sqrt{w_j}, & \text{for } k = 0; j = 1, \dots, r, \\ \sqrt{w_j}\cos(\pi k x_j), & \text{for } k = 1, \dots M; j = 1, \dots, r, \end{cases} \tag{2.4}$$

and setting $s^{(w)} = \{\sqrt{w_j} s_j\}_{j=1}^r$ we can reformulate the least squares problem (2.3) as

$$\min_{c \in \mathbb{R}^{M+1}} \|Vc - s^{(w)}\|_2^2. \tag{2.5}$$

It is well-known that the solution of (2.5) can be computed by solving the normal equations

$$V^T V c = V^T s^{(w)}. \tag{2.6}$$

Switching to the normal equations can lead to problems of numerical instability due to the squaring of the condition number of V. However, as we will see, the system matrix of the normal equations has a very nice algebraic structure that paves the way to fast numerical algorithms for solving (2.3). Thus to handle the trade-off between numerical stability and computational efficiency it is important to have an a priori estimate of the condition number of the matrix V. Such an estimate will aid us in the decision if we shall compute the least squares solution by a direct solution of the system $Vc = s^{(w)}$ or by switching to the system $V^T V c = V^T s^{(w)}$.

The following theorem provides both insight in the algebraic structure of $V^T V$ and an upper bound of the condition number of $V^T V$.

THEOREM 2.1. *Assume we are given nonuniformly spaced sampling points $\{x_j\}_{j=1}^r \in [0,1]$, sampling values $s = \{s_j\}_{j=1}^r$ and positive weights $\{w_j\}_{j=1}^r$. Define $A := V^T V$, where V is as in (2.4), and set $b = V^T s^{(w)}$. There holds:*
(i) The matrix A is a scaled Toeplitz+Hankel matrix of the form

$$A = D(T+H)D, \tag{2.7}$$

[2] Throughout the paper we will always assume that the sampling locations x_j are pairwise distinct.

where

$$T = \begin{bmatrix} a_0 & a_1 & \dots & a_{M-1} & a_M \\ a_1 & a_0 & \ddots & & a_{M-1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{M-1} & & \ddots & \ddots & a_1 \\ a_M & a_{M-1} & \dots & a_1 & a_0 \end{bmatrix}, \; H = \begin{bmatrix} a_0 & a_1 & \dots & a_{M-1} & a_M \\ a_1 & a_2 & \iddots & a_M & a_{M+1} \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ a_{M-1} & \iddots & \iddots & & a_{2M} \\ a_M & a_{M+1} & \dots & a_{2M} & a_{2M+1} \end{bmatrix}, \tag{2.8}$$

with

$$a_k = \frac{1}{2}\sum_{j=1}^{r} w_j \cos(\pi k x_j), \qquad k = 0, \dots, 2M+1, \tag{2.9}$$

and $D = \operatorname{diag}(\frac{1}{\sqrt{2}}, 1, \dots, 1)$.
(ii) If $M < r$ *then* A *is invertible and the coefficient vector* $c = \{c_k\}_{k=0}^{M}$ *of the cosine polynomial* $p \in \boldsymbol{P}_M$ *that solves* (2.3) *is given by*

$$c = A^{-1}b. \tag{2.10}$$

(iii) Define the weights w_j *by*

$$w_j = \frac{x_{j+1} - x_{j-1}}{2}, \qquad j = 1, \dots, r, \tag{2.11}$$

where we set $x_0 := -x_1, x_{r+1} := 2 - x_r$. *If*

$$\delta := \max_j |x_{j+1} - x_j| < \frac{1}{M} \tag{2.12}$$

then the condition number $\kappa(A)$ *is bounded by*

$$\kappa(A) \le \frac{(1+\delta M)^2}{(1-\delta M)^2}. \tag{2.13}$$

Proof. (i) Note that

$$A_{k,l} = (V^T V)_{k,l} = \varepsilon_{k,l} \sum_{j=1}^{r} w_j \cos(\pi l x_j)\cos(\pi k x_j), \qquad k, l = 0, \dots, M, \tag{2.14}$$

where

$$\varepsilon_{k,l} = \begin{cases} \frac{1}{2} & \text{if } k = 0 \text{ and } l = 0, \\ \frac{1}{\sqrt{2}} & \text{if } k = 0 \text{ or } l = 0,\, k \ne l, \\ 1 & \text{if } k > 0 \text{ and } l > 0. \end{cases} \tag{2.15}$$

The result follows now readily from a simple calculation by applying the formula

$$\cos(\alpha)\cos(\beta) = \cos(\alpha+\beta) + \cos(\alpha-\beta), \tag{2.16}$$

to (2.14) and using the fact that the entries of T and H satisfy $T_{k,l} = a_{k-l}$ and $H_{k,l} = a_{k+l}$ respectively.

(ii) The invertibility of A follows from the well-known fact that the Vandermonde-like matrix V has rank $M+1$ for mutually different points x_j (assuming $w_j \ne 0$). The rest follows from (2.6).

(iii) With the exception of a few minor modifications the proof of this part is similar to Gröchenig's elegant proof on the upper bound of the condition number of certain Toeplitz matrices, see [7]. However instead of confronting the reader with a patchwork of required modifications of Gröchenig's proof we prefer to present a complete proof.

The proof makes use of *Wirtinger's inequality* [9]: If $f \in \boldsymbol{L}^2(a,b)$ and either $f(a) = 0$ or $f(b) = 0$, then

$$\int_a^b |f(x)|^2\,dx \le \frac{4}{\pi^2}(b-a)^2 \int_a^b |f'(x)|^2\,dx. \tag{2.17}$$

We proceed with the proof of (iii). Let P be the orthogonal projection of $\boldsymbol{L}^2([0,1])$ onto P_M. Define the operator S by

$$Sp = P\Big(\sum_{j=1}^r p(x_j)\chi_j\Big). \tag{2.18}$$

Here $\chi_j(x)$ denotes the characteristic function of the interval $[y_{j-1}, y_j]$, where $y_j = \frac{x_{j+1}-x_j}{2}, j = 1,\dots,r$ with $x_0 = -x_1, x_{r+1} = 1 - x_r$.

We compute

$$\|p - Sp\|_2^2 = \|P\big(\sum_{j=1}^r (p - p(x_j))\chi_j\big)\|_2^2 \le \|\sum_{j=1}^r (p - p(x_j))\chi_j\|_2^2 =$$
$$\int_0^1 |\sum_{j=1}^r (p - p(x_j))\chi_j|^2 dx = \sum_{j=1}^r \int_{y_{j-1}}^{y_j} |p - p(x_j)|^2 dx. \tag{2.19}$$

We write

$$\int_{y_{j-1}}^{y_j} |p - p(x_j)|^2 dx = \int_{y_{j-1}}^{x_j} |p - p(x_j)|^2 dx + \int_{x_j}^{y_j} |p - p(x_j)|^2 dx,$$

and apply Wirtinger's inequality (2.17) to each of the integrals on the left-hand side. Since $|y_j - x_j| \le \delta/2$ and $|x_j - y_{j-1}| \le \delta/2$ we obtain

$$\sum_{j=1}^r \int_{y_{j-1}}^{y_j} |p - p(x_j)| dx \le \frac{\delta^2}{\pi^2} \sum_{j=1}^r \int_0^1 |p'(x)| dx = \frac{\delta^2}{\pi^2}\|p'\|_2^2. \tag{2.20}$$

Note that

$$p'(x) = \sum_{k=0}^M c_k \pi k \sin(\pi k x), = \sum_{k=1}^M c_k \pi k \sin(\pi k x). \tag{2.21}$$

Hence we have the Bernstein type inequality

$$\|p'\|_2^2 = \int_0^1 |\sum_{k=1}^M c_k \pi k \sin(\pi k x)|^2 dx$$
$$\le (\pi M)^2 \int_0^1 |\sum_{k=1}^M c_k \sin(\pi k x)|^2 dx \le (\pi M)^2 \|p\|_2^2. \tag{2.22}$$

Thus by combining (2.19), (2.20) and (2.22) we get

$$\|p - Sp\|_2^2 \le \delta^2 M^2 \|p\|_2^2. \tag{2.23}$$

Hence

$$\|I - S\|_{\mathrm{op}} \le \delta M, \tag{2.24}$$

and since $\delta < 1/M$ by assumption, we conclude that S is invertible and

$$\|S^{-1}\|_{\mathrm{op}} \le (1 - \delta M)^{-1}. \tag{2.25}$$

There holds

$$(1-\delta M)^2\|p\|_2^2 = (1-\delta M)^2\|S^{-1}Sp\|_{\rm op}^2 \leq$$
$$\leq (1-\delta M)^2\|S^{-1}\|_{\rm op}^2\|Sp\|_2^2 \leq \|Sp\|_2^2 \leq \sum_{j=1}^{r}|p(x_j)|^2 w_j\,. \tag{2.26}$$

Also

$$\sum_{j=1}^{r}|p(x_j)|^2 w_j \leq \|p-p+\sum_{j=1}^{r}p(x_j)\chi_j\|_2^2 \leq \big(\|p\|+\|p-\sum_{j=1}^{r}p(x_j)\chi_j\|_2\big)^2$$
$$\leq \big(\|p\|_2+\delta M\|p\|_2\big)^2 \leq (1+\delta M)^2\|p\|_2^2. \tag{2.27}$$

Thus

$$(1-\delta M)^2\|p\|_2^2 \leq \sum_{j=1}^{r}|p(x_j)|^2 w_j \leq (1+\delta M)^2\|p\|_2^2. \tag{2.28}$$

By definition we have for any $p \in \boldsymbol{P}_M$ with coefficient vector a

$$\langle Aa,a\rangle = \langle V^TVa,a\rangle = \langle Va,Va\rangle = \sum_{j=1}^{r}|p(x_j)|^2 w_j\,. \tag{2.29}$$

Using the relation $\|p\|_2^2 = \frac{1}{2}\|a\|_2^2$ we obtain

$$\frac{1}{2}(1-\delta M)^2\|a\|_2^2 \leq \langle Aa,a\rangle \leq \frac{1}{2}(1+\delta M)^2\|a\|_2^2, \tag{2.30}$$

and therefore

$$\kappa(A) \leq \frac{(1+\delta M)^2}{(1-\delta M)^2}.$$

❏

Remark: We briefly analyze the least squares problem (2.5) when using non-scaled cosine polynomials $\tilde{p}(x) = \sum_{k=0}^{M} c_k\cos(\pi k x)$. It is easy to see that the corresponding Vandermonde-like matrix $\tilde{V}$ satisfies

$$\tilde{V}D = V, \tag{2.31}$$

with D as in part (i) of Theorem 2.1 and V as in (2.4). Hence

$$\tilde{A} := \tilde{V}^T\tilde{V} = D^{-1}V^TVD^{-1}. \tag{2.32}$$

The estimates

$$\|\tilde{A}x\|_2 \leq \|D^{-1}\|_{\rm op}^2\|A\|_{\rm op}\|x\|_2 \leq 2\|A\|_{\rm op}\|x\|_2, \tag{2.33}$$

and

$$\|\tilde{A}^{-1}x\|_2 \leq \|D\|_{\rm op}^2\|A\|_{\rm op}\|x\|_2 \leq \|A\|_{\rm op}\|x\|_2, \tag{2.34}$$

imply that

$$\mathrm{cond}(\tilde{A}) \leq 2\,\mathrm{cond}(A). \tag{2.35}$$

Thus the condition number of $\tilde{A}$ can be twice as large as the condition number of A. This is why we prefer to use scaled cosine polynomials as defined in (2.1). The inequality (2.35) is sharp as can be seen from the following simple example. Let the sampling points x_j be equally spaced, and choose the weights w_j as in Theorem 2.1. In this case it is not difficult to see that

$$A = \frac{1}{2}I_{M+1}, \tag{2.36}$$

where I_{M+1} denotes the $(M+1)\times(M+1)$ identity matrix, whereas

$$\tilde{A} = \frac{1}{2}\begin{bmatrix} 2 & 0 & \dots & 0\\ 0 & 1 & \dots & 0\\ \vdots & & \ddots & \vdots\\ 0 & & & 1\end{bmatrix}. \tag{2.37}$$

Thus obviously $\mathrm{cond}(\tilde{A}) = 2\,\mathrm{cond}(A)$ in this case.

3. Fast computation of the least squares approximation. In this section we present a fast algorithm for solving the least squares problem (2.3). Our algorithm is based on the conjugate gradient method in connection with a fast matrix-vector multiplication involving the DCT. Before we proceed we briefly review some properties of the DCT-I. There are four types of the DCT, cf. [22]. For our purposes we will use the (scaled) DCT-I.

DEFINITION 3.1. *The* Type-I Discrete Cosine Transform matrix *(DCT-I for short) of size $n \times n$ is defined by*

$$[C_n]_{k,l} = \begin{cases} \frac{1}{\sqrt{2n-2}} \cos(\pi \frac{kl}{n-1}) & \text{if } k = 0 \text{ or } k = n-1, \\ \frac{2}{\sqrt{2n-2}} \cos(\pi \frac{kl}{n-1}) & \text{if } k = 1, \dots, n-2. \end{cases} \tag{3.1}$$

If the dimension of the matrix C_n is clear from the context we drop the subscript and simply write C instead.

The DCT-I matrix C satisfies $CC = I$. It is not unitary, but can be easily made unitary by appropriate scaling. For define the diagonal matrix $\tilde{D} = \mathrm{diag}([1, \sqrt{2}, \dots, \sqrt{2}, 1])$ and set $\tilde{C} = \tilde{D}{-}1C\tilde{D}$. Then it is easy to see that $\tilde{C}\tilde{C}^T = I$. In some cases it is more convenient to work with $\tilde{C}$ instead of C [11]. However the results presented in this paper can be more elegantly expressed when using the definition (3.1) of the DCT-I. Fast algorithms for computing Cx require 2.5 $\mathcal{O}(n \log n)$ operations if x is a vector of length $n+1$ and n is a power of two [25], cf. also [21, 1].

It is well-known that the DCT-I matrix diagonalizes certain Toeplitz+Hankel matrices [18, 11]. For let $T = \mathrm{toep}(a)$ be a symmetric Toeplitz matrix with first column $a = [a_0, a_1, \dots, a_n]^T$. We define the counter-identity matrix J by

$$J = \begin{bmatrix} 0 & & 1 \\ & \iddots & \\ 1 & & 0 \end{bmatrix}. \tag{3.2}$$

If

$$B = \mathrm{toep}(a) + J\,\mathrm{toep}(Ja) := T + H \tag{3.3}$$

(note that $J\,\mathrm{toep}(Ja)$ is a Hankel matrix that is symmetric with respect to the counter diagonal) then

$$C^T BC = \Sigma, \qquad \text{where } \Sigma \text{ is a diagonal matrix.} \tag{3.4}$$

An important consequence of this diagonalization property is that the multiplication of a matrix B of the form (3.3) with a vector x can be carried out in $\mathcal{O}(n \log n)$ operations via DCT-I [1], similar to the multiplication of a vector by a Toeplitz matrix which can be computed via FFT by embedding the Toeplitz matrix into a circulant matrix.

To be precise, assume we want to compute $y = Bx$ where $C^T BC = \Sigma$. There holds

$$y = Bx = C^T C^T BCCx = C^T \Sigma Cx. \tag{3.5}$$

Of course in a numerical implementation we would not compute the diagonal matrix Σ explicitly. Instead we proceed as follows. Let b be the first column of B, define the scaling matrix $D_1 = \mathrm{diag}(2, 1, \dots, 1, 2)$ and observe that $C = D_1^{-1} C^T D_1$. A simple calculation shows that $D_1^{-1}\Sigma = \sqrt{\frac{n-1}{2}}\,\mathrm{diag}(C^T b)$. Hence

$$y = Bx = C^T D_1 D_1^{-1} \Sigma Cx = \sqrt{\frac{n-1}{2}} C^T \,\mathrm{diag}(D_1 C^T b) D_1^{-1} C^T D_1 x,$$

and therefore

$$y = \sqrt{\frac{n-1}{2}} C^T \left[(C^T b) \circ (C^T D_1 x]) \right], \tag{3.6}$$

where the operation "$\circ$" denotes the pointwise product between vectors. Hence the product Bx can be computed by three DCT-I's in $\mathcal{O}(n \log n)$ operations.

Observe that the Toeplitz+Hankel part of the matrix $A = D(T+H)D$ in (2.7) of Theorem 2.1 is *not* of the form (3.3), since the first row and the last column of the Hankel matrix H in (2.8) have different entries. Thus A is *not* diagonalized by the DCT-I (or any other DCT). But we can embed the Toeplitz+Hankel part of A in a Toeplitz+Hankel matrix of the form (3.3), similar to the embedding of a Toeplitz matrix in a circulant matrix. To see this, let T and H be defined as in (2.8). We embed $T + H$ in the $(2M+1) \times (2M+1)$ augmented Toeplitz+Hankel matrix $T_{\text{aug}} + H_{\text{aug}}$, where

$$T_{\text{aug}} = \begin{bmatrix} a_0 & \cdots & a_M & a_{M+1} & \cdots & a_{2M} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ a_M & \ddots & \ddots & \ddots & \ddots & a_M \\ a_{M+1} & \ddots & \ddots & \ddots & \ddots & a_{M-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ a_{2M} & \cdots & a_{M+1} & a_M & \cdots & a_0 \end{bmatrix}, \tag{3.7}$$

$$H_{\text{aug}} = \begin{bmatrix} a_0 & \cdots & a_M & a_{M+1} & \cdots & a_{2M} \\ \vdots & ⋰ & ⋰ & ⋰ & ⋰ & \vdots \\ a_M & ⋰ & ⋰ & ⋰ & ⋰ & a_M \\ a_{M+1} & ⋰ & ⋰ & ⋰ & ⋰ & a_{M-1} \\ \vdots & ⋰ & ⋰ & ⋰ & ⋰ & \vdots \\ a_{2M} & \cdots & a_{M+1} & a_M & \cdots & a_0 \end{bmatrix}. \tag{3.8}$$

The matrix $T + H$ is the $(M+1) \times (M+1)$ principal leading submatrix of $T_{\text{aug}} + H_{\text{aug}}$.

Thus for a DCT-I based fast implementation of the matrix vector product Ax we proceed as follows. We write $y = Ax = D(T+H)Dx$ and define $x_{\text{aug}} := [(Dx)^T, 0, \ldots, 0]^T$. Compute $y_{\text{aug}} = A_{\text{aug}} x_{\text{aug}}$ according to (3.6). The vector y is then given by the first $M+1$ entries of y_{aug} multiplied by D.

In order to obtain augmented matrices whose size is $2^n + 1$ we can always insert as many zeros as necessary after a_{2M} in the first row of T_{aug} and H_{aug} without destroying the algebraic structure of the matrices. Thus the matrix vector multiplication Ax can always be carried out in $\mathcal{O}(M \log M)$. This zero-padding is similar to the zero-padding of the Toeplitz case (where the zeros are added in the middle of the first row).

Note that a direct computation of the entries of the matrix A and of the right hand side b will take $\mathcal{O}(Mr)$ operations. Thus, although we can solve the system $Ax = b$ in $\mathcal{O}(M \log M)$ operations, the computation of the entries of A and b will soon become the bottleneck for large scale problems. Fortunately there exist fast algorithms for computing sums of the form (2.9). In [14] Daniel Potts has developed fast algorithms for computing the DCT for nonuniformly spaced points. Like nonuniform FFT algorithms [15] a nonuniform DCT-I (NDCT for short) can be computed in $\mathcal{O}(\alpha M \log(\alpha M) + mr)$ operations, where α and m are constants. See [14] for details.

Based on the observations above, we propose the following fast algorithm for solving the least squares problem (2.3).

Algorithm 1 (Fast scattered data approximation using cosine polynomials).

Input: *Nonuniformly spaced sampling points* $\{x_j\}_{j=1}^r \in [0,1]$, *sampling values* $\{s_j\}_{j=1}^r$, *weights* $\{w_j\}_{j=1}^r$ *and user-defined points* $\{t_l\}_{l=0}^L \in [0,1]$.

Task: *Compute the coefficients of the cosine polynomial of degree M that solves* (2.3) *and evaluate the polynomial at the points* $\{t_l\}_{l=1}^L$.

Step 1: *Compute the first column of A in* (2.7) *and the right hand side $b = V^T s^{(w)}$ via NDCT. This takes $\mathcal{O}(\alpha M \log(\alpha M) + mr)$ operations, where α and m are (small) constants.*

Step 2: *Solve $Ac = b$ iteratively by the conjugate gradient method. Using fast matrix-vector multiplication this can be done in $\mathcal{O}(M \log M)$ operations per iteration.*

Step 3: *Evaluate* $p(x) = \frac{c_0}{\sqrt{2}} + \sum_{k=1}^{M} c_k \cos(\pi k x)$ *at the points* $\{t_l\}_{l=0}^{L}$. *If* $t_l = l/L$ *and* $L = 2^n$ *for some* $n \in \mathbb{N}$, *then this can be done by a DCT in* $\mathcal{O}(L \log L)$ *operations. If* $L \neq 2^n$ *we can use a fast radix-p DCT, see [21]. If the* t_l *are nonuniformly spaced we use a NDCT to compute* $\{p(t_l)\}_{l=1}^{L}$.
Output: *Least squares approximating polynomial* p *of degree* M, *evaluated at the points* $\{t_l\}_{l=1}^{L}$.

REMARK: If the sampling set satisfies the maximal gap condition (2.12) and the weights are chosen according to (2.11) we can utilize the bound on $\kappa(A)$ in (2.13) of Theorem 2.1 to estimate the rate of CG using the standard formula [5]

$$\|c^{(n)} - c\|_2 \leq 2\kappa(A) \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \|c^{(0)} - c\|_2, \tag{3.9}$$

where $c^{(n)}$ denotes the solution after the n-th iteration of CG applied to $Ac = b$.

If the condition number of A is large (whether or not the maximal gap condition is satisfied) it may be better to solve the least squares problem (2.3) $Vc = b$ without explicitly establishing the normal equations. One can resort to "non-symmetric" versions of CG such as GMRES or LSQR, cf. [5]. Since the NDCT provides a fast way to carry out the multiplication of the matrix V with a vector we still obtain a fast algorithm. However the computational costs are in general larger than those for Algorithm 1 since a NDCT is more expensive than a DCT and the NDCT has to be applied in each iteration, whereas in Algorithm 1 it has to be applied only in the initial stage of the algorithm.

If the matrix A is ill-conditioned due to large gaps in the sampling set one might be tempted to apply one of the cosine-transform based preconditioners to improve the situation. However preconditioners cannot significantly improve the stability in this case. This can be shown in a similar way as it is done in Section 4.2 of [24] for trigonometric approximation using exponentials.

There exist fast direct methods to solve Toeplitz+Hankel systems (not all of them apply to our situation though), see [12] and in particular the work of Heinig [11, 10] . But many of these solvers require that the matrix dimension is a power of two. It is possible to overcome this severe constraint, however at the cost of a more involved algorithm. As we have seen for the conjugate gradient iterations the initial size of the matrix does not play a major role, since when constructing the augmented matrix we can always insert the appropriate number of zeros to get a size of a power of two. Furthermore, if the set of sampling points is a jittered version of a set of regularly spaced points, standard perturbation theory implies that the eigenvalues of A will be clustered around 1. Thus CG will converge in very few iterations. Direct solvers cannot take advantake of such sitations.

3.1. Multilevel scattered data approximation. The reader may have noticed that we have tacitly assumed that the polynomial degree M is given a priori. Although this is a common assumption in polynomial approximation it is not justified in many applications. In fact, the appropriate choice of M has a major influence on the usefulness of the resulting approximating polynomial, cf. [23]. In [20] Otmar Scherzer and the second author have developed a multilevel scheme that automatically adapts to the solution of the optimal "level", i.e., the optimal polynomial degree in our case. This multilevel algorithm applies to our approximation method without modification.

In a nutshell the multilevel version of Algorithm 1 works as follows, for details we refer to [20, 8]. We start at the first *level* with an initial choice for the approximating polynomial (e.g., $M_0 = 1$) and apply Algorithm 1. We stop the CG iterations when a specific stopping criterion is satisfied and obtain the approximation p_1, say. Then we proceed to the next *level* by choosing a degree $M_1 > M_0$ (e.g., $M_1 = M_0 + 1$). We use the approximation p_1 from the previous level as initial guess for the solution at the new level and apply Algorithm 1. We proceed through increasing levels until at the k-th level the approximating polynomial p_k satisfies the *discrepancy principle*

$$\sum_{j=1}^{r} |p_k(x_j) - s_j| w_j \leq \varepsilon \sum_{j=1}^{r} |s_j|^2 w_j, \tag{3.10}$$

where ε is a parameter related to the accuracy of the given data s_j.

A fast $\mathcal{O}(M \log M)$ implementation of the multi-level scheme for cosine polynomials can be derived in a similar way as it is done for the exponentials, see Algorithm 2 in Section 5.1 of [8]. An crucial observation thereby is that the scaled Toeplitz+Hankel matrix A_M associated with the least squares problem (2.5) for degree M is related to the matrix A_{M+1} associated with the least squares problem (2.5) in a nice way. Namely, A_M is the principal leading submatrix of A_{M+1}.

REMARK: Finding the optimal level for the approximating function is a common and important problem in scattered data approximation. When using radial basis functions or shift-invariant systems as model one has to deal with the trade-off between accuracy and stability when determining the width of the basis functions, cf. e.g.[19]. The multi-level idea provides a natural framework to handle this trade-off.

4. Two-dimensional scattered data approximation. Many of the results of the previous sections can be extended to arbitrary dimensions. For the sake of simplicity of notation we will focus mainly on the two-dimensional case.

We are given sampling values $s = \{s_j\}_{j=1}^r$ and randomly spaced sampling points $\{(x_j, y_j)\}_{j=1}^r$. Without loss of generality we assume that $(x_j, y_j) \in [0,1] \times [0,1]$, otherwise we can always renormalize the sampling points accordingly.

The space $\boldsymbol{P}_{M_x M_y}$ consists of two-dimensional cosine polynomials p of degree $M_x M_y$ defined by

$$p(x, y) = \frac{c_{0,0}}{\sqrt{2}} + \sum_{\substack{k=0 \\ \max\{k,l\}>0}}^{M_x} \sum_{l=0}^{M_y} c_{k,l} \cos(\pi k x) \cos(\pi l y), \tag{4.1}$$

with real-valued coefficients $c_{k,l}$.

Analogous to the one-dimensional scattered data problem we want to find the $p \in \boldsymbol{P}_{M_x M_y}$ that solves

$$\min \sum_{j=1}^{r} |p(x_j, y_j) - s_j|^2 w_j. \tag{4.2}$$

We define the block matrix V by

$$V = \begin{bmatrix} V^{(0)} & V^{(1)} & \dots & V^{(M_y)} \end{bmatrix}, \tag{4.3}$$

$$\text{with } V_{j,k}^{(l)} = \varepsilon_{k,l} \sqrt{w_j} \cos(\pi k x_j) \cos(\pi l y_j), \quad j = 1, \dots, r, \tag{4.4}$$

$$\text{where } \varepsilon_{k,l} = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } k = 0 \text{ and } l = 0, \\ 1 & \text{if } k = 0, \dots, M_x; l = 0, \dots, M_y; \max\{k, l\} > 0. \end{cases} \tag{4.5}$$

By stacking the columns of c and with a slight abuse of notation we can rewrite (4.2) as

$$\min \|Vc - s^{(w)}\|, \tag{4.6}$$

where $s^{(w)} = \{\sqrt{w_j} s_j\}_{j=1}^r$.

Similar to the 1-D case, we can solve (4.6) by switching to the normal equations. The next theorem describes the algebraic structure of the system matrix of the normal equations.

THEOREM 4.1. *Let V be as defined in* (4.3)-(4.4). *Then the matrix $A := V^T V$ is a scaled block*

Toeplitz+Hankel matrix of the form $A = D(T+H)D$ with

$$T = \begin{bmatrix} A^{(0)} & A^{(1)} & \dots & A^{(M_y-1)} & A^{(M_y)} \\ A^{(1)} & A^{(0)} & \ddots & & A^{(M_y-1)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ A^{(M_y)} & \dots & \dots & \dots & A^{(0)} \end{bmatrix}, \tag{4.7}$$

$$H = \begin{bmatrix} A^{(0)} & A^{(1)} & \dots & A^{(M_y-1)} & A^{(M_y)} \\ A^{(1)} & A^{(2)} & \iddots & \iddots & A^{(M_y+1)} \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & & A^{(2M_y-1)} \\ A^{(M_y)} & \dots & \dots & A^{(2M_y-1)} & A^{(2M_y)} \end{bmatrix}, \tag{4.8}$$

where each block $A^{(k)}, k = 0, \dots, 2M_y$ is an $(M_x+1)\times(M_x+1)$ matrix of the form $A^{(k)} = T^{(k)}+H^{(k)}$ with $T(k)$ and $H^{(k)}$ as in (2.8) *and* $D = \operatorname{diag}(\frac{1}{\sqrt{2}}, 1, \dots, 1)$.

Proof. It follows from (4.3) and (4.4) that

$$A_{l,l',k,k'} = \varepsilon_{k,l}\varepsilon_{k',l'} \sum_{j=1}^{r} w_j \Big(\cos(\pi(l+l')x_j)\cos(\pi(k+k')y_j) + \cos(\pi(l-l')x_j)\cos(\pi(k+k')y_j) + \\ + \cos(\pi(l+l')x_j)\cos(\pi(k-k')y_j) + \cos(\pi(l-l')x_j)\cos(\pi(k-k')y_j)\Big). \tag{4.9}$$

Here the indices l, l' refer to the (l, l')-th block of A and the indices k, k' refer to the element in the k-th row and k'-th column in a certain block.

Now we consider the entries of A for fixed l and l'. Using formula (2.16) we calculate

$$A_{l,l',k,k'} = \varepsilon_{k,l}\varepsilon_{k',l'} \sum_{j=1}^{r} w_j \Big(c_1[\cos(\pi(k+k')y_j) + \cos(\pi(k-k')y_j)] + \\ + c_2[\cos(\pi(k+k')y_j) + \cos(\pi(k-k')y_j)]\Big), \qquad k, k' = 0, \dots, M_x, \tag{4.10}$$

where the constants c_1 and c_2 are given by $c_1 := \cos(\pi(l+l')x_j)$, $c_2 := \cos(\pi(l-l')x_j)$. Thus the (l, l')-th block of A is indeed of the form (2.7).

By repeating this step with reversed roles for k, k' and l, l' we see that the "global" structure of A is of the form (4.8). ❏

In order to utilize the block Toeplitz+Hankel structure of the normal equations we have to extend the fact that the DCT-I diagonalizes certain Toeplitz+Hankel matrices to the case of block Toeplitz+Hankel matrices.

We need some preparation before we proceed. Let B be a block matrix of the form

$$B = \begin{bmatrix} B^{(0,0)} & \dots & B^{(0,n-1)} \\ \vdots & & \vdots \\ B^{(n-1,0)} & \dots & B^{(n-1,n-1)} \end{bmatrix} \tag{4.11}$$

where the blocks $B^{(k,l)}$ are matrices of size $m \times m$. For such block matrices we define the *mod-m permutation matrix* $\Pi_{m,n}$ via

$$[\Pi_{m,n} B \Pi_{m,n}^T]_{i,j;k,l} = B_{k,l;i,j}, \qquad 0 \le i, j \le m-1, 0 \le k, l \le n-1. \tag{4.12}$$

In words, the (i, j)-th entry of the (k, l)-th block of B is permuted to the (k, l)-th entry of the (i, j)-th block. We have $\Pi_{m,n} = \Pi_{n,m}^T$, see [25].

DEFINITION 4.2. *The* two-dimensional type-I Discrete Cosine Transform *of an* $m \times n$ *signal* x *is given by*

$$(Cx)_{i,j} = \frac{\varepsilon_{i,j}}{\sqrt{2m-2}\sqrt{2n-2}} \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} x_{k,l} \cos\big(\pi \frac{ik}{m-1}\big) \cos\big(\pi \frac{jl}{n-1}\big), \tag{4.13}$$

$$i = 0, \dots, m-1; j = 0, \dots, n-1, \tag{4.14}$$

where

$$\varepsilon_{i,j} = \begin{cases} 1 & \text{if } i \in \{0, m-1\} \text{ and } j \in \{0, n-1\}, \\ 2 & \text{if } i \in \{0, m-1\} \text{ and } j \notin \{0, n-1\}, \\ 2 & \text{if } i \notin \{0, m-1\} \text{ and } j \in \{0, n-1\}, \\ 4 & \text{if } i = 1, \dots, m-2 \text{ and } j = 1, \dots, n-2. \end{cases}$$

The two-dimensional DCT-I can be represented by the $mn \times mn$ matrix $C_m \otimes C_n$ where the matrices C_m and C_n represent one-dimensional DCT-I's as in definition 3.1 and $\otimes$ denotes the usual Kronecker product.

Similar to the 1-D DCT-I the 2-D DCT-I diagonalizes certain block Toeplitz+Hankel matrices.

THEOREM 4.3. *A matrix* B *is diagonalized by a two-dimensional DCT-I if and only if* B *is of the form*

$$B = \begin{bmatrix} B^{(0)} & B^{(1)} & \dots & B^{(n-2)} & B^{(n-1)} \\ B^{(1)} & B^{(0)} & \ddots & & B^{(n-2)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ B^{(n-1)} & \dots & \dots & \dots & B^{(0)} \end{bmatrix} + \begin{bmatrix} B^{(0)} & B^{(1)} & \dots & B^{(n-2)} & B^{(n-1)} \\ B^{(1)} & B^{(2)} & \iddots & \iddots & B^{(n-2)} \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & & \vdots \\ B^{(n-1)} & \dots & \dots & \dots & B^{(0)} \end{bmatrix}, \tag{4.16}$$

where each block $B^{(k)}, k = 0, \dots, n-1$ *is a* $m \times m$ *Toeplitz+Hankel matrix of the form* (3.3).

Proof. The proof is similar to the proof of Theorem 3.3 in [13] and uses basic properties of the Kronecker product $\otimes$. Let B be a block Toeplitz+Hankel matrix as in the assumption of the theorem. We have to show that B is diagonalized by the two-dimensional DCT-I $C = C_n \otimes C_m$. Note that each block $B^{(k)}$ of B can be diagonalized by a one-dimensional DCT-I C_m, i.e., $C_m^T B^{(k)} C_m = \Lambda^{(k)}$, $k = 0, \dots n-1$, where the $\Lambda^{(k)}$ are $m \times m$ diagonal matrices. Since $C_n \otimes C_m = (C_n \otimes I_m)(I_n \otimes C_m)$ it follows that

$$(C_n \otimes C_m)^T B (C_n \otimes C_m) = (C_n^T \otimes I_m)(I_n \otimes C_m^T) B (I_n \otimes C_m)(C_n \otimes I_m) = (C_n^T \otimes I_m) \Lambda (C_n \otimes I_m), \tag{4.17}$$

where

$$\Lambda = \begin{bmatrix} \Lambda^{(0)} & \Lambda^{(1)} & \dots & \Lambda^{(n-2)} & \Lambda^{(n-1)} \\ \Lambda^{(1)} & \Lambda^{(0)} & \ddots & & \Lambda^{(n-2)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \Lambda^{(n-1)} & \dots & \dots & \dots & \Lambda^{(0)} \end{bmatrix} + \begin{bmatrix} \Lambda^{(0)} & \Lambda^{(1)} & \dots & \Lambda^{(n-2)} & \Lambda^{(n-1)} \\ \Lambda^{(1)} & \Lambda^{(2)} & \iddots & \iddots & \Lambda^{(n-2)} \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & & \vdots \\ \Lambda^{(n-1)} & \dots & \dots & \dots & \Lambda^{(0)} \end{bmatrix}. \tag{4.18}$$

We compute

$$\Pi_{m,n} \Lambda \Pi_{m,n}^T = \tilde{B} = \begin{bmatrix} \tilde{B}^{(0)} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \tilde{B}^{(1)} & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \tilde{B}^{(m-1)} \end{bmatrix}, \tag{4.19}$$

where $\mathbf{0}$ is an $n \times n$ zero matrix. It follows from (4.18) that each $\tilde{B}^{(k)}, k = 0, \ldots, m-1$ is an $n \times n$ Toeplitz+Hankel matrix of the form (3.3). Therefore $C_n^T \tilde{B}^{(k)} C_n = \tilde{\Lambda}^{(k)}, k = 0, \ldots, m-1$.

Since $\Pi_{m,n}(C_n^T \otimes I_m)\Pi_{m,n}^T = I_m \otimes C_n^T$ (e.g., see [25]) we have

$$\begin{aligned}
(C_n^T \otimes I_m)\Lambda(C_n \otimes I_m) &= \Pi_{m,n}\Pi_{m,n}^T(C_n^T \otimes I_m)\Pi_{m,n}^T \Pi_{m,n}\Lambda\Pi_{m,n}^T \Pi_{m,n}(C_n \otimes I_m)\Pi_{m,n}\Pi_{m,n}^T \\
&= \Pi_{m,n}^T(I_m \otimes C_n^T)\tilde{B}(I_m \otimes C_n^T)\Pi_{m,n} \\
&= \Pi_{m,n}^T \tilde{\Lambda}\Pi_{m,n},
\end{aligned} \tag{4.20}$$

where $\tilde{\Lambda}$ is a block diagonal matrix with diagonal blocks $\tilde{\Lambda}^{(k)}$. Thus $\tilde{\Lambda}$ is a diagonal matrix. It follows from the definition of $\Pi_{m,n}$ that $\Pi_{m,n}^T \tilde{\Lambda}\Pi_{m,n}$ is then also a diagonal matrix.

The opposite direction follows from the fact that $CC = I$. ❑

The matrix A associated with the least squares problem (4.6) is *not* diagonalized by the 2-D DCT-I. But analogous to the 1-D case, A can be embedded into a block Toeplitz+Hankel matrix that is diagonalized by the 2-D DCT-I. Thus similar to the 1-D case the matrix-vector multiplication Ax can be carried out in $\mathcal{O}(M_x M_y \log M_x M_y)$ operations.

We leave it to the reader to extend Theorems 4.1 and 4.3 and the fast matrix-vector multiplication to dimensions larger than two. Since the NDCT can also be generalized to two and higher dimensions we have a fast numerical algorithm for computing the least squares approximation using cosine polynomials in multiple dimensions in the same way as it is outlined in Algorithm 1.

REMARK: There is one notable difficulty that arises when considering the scattered data approximation problem in higher dimensions. In the 1-D case a sufficient condition for invertibility of the matrix A is that the polynomial degree M is smaller than the number of samples r. This is an immediate consequence of the fundamental theorem of algebra. Unfortunately the fundamental theorem of algebra does not extend to the multi-dimensional case. It is obvious that a necessary condition for the existence of A^{-1} is $M < r$. However this condition is no longer sufficient, since the sampling points need not be appropriately distributed. In higher dimensions, the zero set of a polynomial is an algebraic curve or an algebraic surface. For A to be invertible, the samples must not be contained in any algebraic surface. It is an open problem to efficiently characterize all sampling sets that yield an invertible matrix A.

It is still possible to obtain conditions that guarantee the existence of A^{-1} as well as to derive estimates for the condition number of A in the multi-dimensional case. This can be done for instance by adapting the approach in Section 4.3 of [6] to our situation. However the estimates are no longer sharp and get worse with increasing dimension. We do not pursue this direction here.

5. Numerical experiments: An example from geophysics. We demonstrate the performance of the proposed algorithm by applying it to a scattered data problem from geophysics. Exploration geophysics relies on measurements of the Earth's physical properties like the magnetic or gravitational field, with the goal of detecting anomalies which reveal underlying geological features. In geophysical practice, it is essentially impossible to gather data in a form that allows direct interpretation. Geoscientists, used to look at their measurements on maps or profiles and aim at further processing, need a representation of the originally irregularly spaced (scattered) data points on a regular grid. The reconstruction or approximation of potential fields on regular grids from scattered data is thus one of the first and crucial steps in the analysis of geophysical data.

As test example we use a synthetic anomaly f that represents the gravitational acceleration caused by an ensemble of buried rectangular boxes of different size, depth, and density contrast, see Fig. 5(a). This example has also been used in [16]. We sample this function at 496 randomly spaced points (x_j, y_j) in the interval $[0,1] \times [0,1]$. Since in practice measurements are always contaminated by noise we add white Gaussian noise in the amount of 5% of the ℓ^2-norm of the samples $f(x_j, y_j)$. We want to reconstruct the function on a regular grid Γ consisting of the grid points $\{(k/150, l/150)\}_{k,l=0}^{150}$.

In order to demonstrate the advantage of using Neumann boundary conditions over periodic boundary conditions we compare the proposed algorithm to the so-called ACT method [4, 8]. The latter has become a main ingredient for several approximation methods in geophysics [16, 2]. We also include in the comparison the approximation obtained by cubic spline interpolation, which we computed via the MATLAB function `griddata` using the option 'cubic'.

For the two methods using trigonometric approximation we use the same number of coefficients for the approximating polynomial. We use a total of 11 coefficients in the x-coordinate and the same number in the y-coordinate, resulting in approximating polynomials of degree 121 for both methods.

Since we know the original anomaly f we can compute the error between the approximation f_a and f via $e(f_a) = \|f(\Gamma) - f_a(\Gamma)\|_2/\|f(\Gamma)\|$ on the grid Γ. The proposed method gives an error of 0.029, the ACT method yields approximation error 0.072, and the approximation computed via cubic splines returns an error of 0.045. The approximation computed by the proposed method is appealing both from a visual and from an approximation error viewpoint.

The significantly larger error of ACT is only due to boundary effects. We note that there are several ways to improve the performance of the ACT method, see [16], which makes it indeed a powerful approximation method in geophysics [16, 2]. Since all these modifications can also be applied to the proposed method we expect that the proposed (modified) algorithm will still be significantly better than the modified ACT method.

The results of this experiment do not mean that the proposed method always performs better than the other two methods. Furthermore, a detailed comparison of various scattered data approximation methods would have to include other standard methods such as approximation by radial basis functions. Such a comparison is beyond the scope of this paper.

Acknowledgment. T.S. wants to thank Raymond Chan and Michael Ng for their kind invitation to Hongkong in December 2000. The initial steps to this research were carried out during this wonderful and inspiring visit.

REFERENCES

[1] G. Baszenski and M. Tasche, *Fast polynomial multiplication and convolutions related to the discrete cosine transform*, Linear Algebra Appl., 252 (1997), pp. 1–25.

[2] A. Duijndam, M. Schonewille, and C. Hindriks, *Reconstruction of band-limited data irregularly sampled along one spatial direction*, Geophysics, 64 (1999), pp. 524–538.

[3] H. Fassbender, *On numerical methods for discrete least-squares approximation by trigonometric polynomials*, Math. Comp., 66 (1997), pp. 719–741.

[4] H. G. Feichtinger, K. Gröchenig, and T. Strohmer, *Efficient numerical methods in non-uniform sampling theory*, Numerische Mathematik, 69 (1995), pp. 423–440.

[5] G. Golub and C. van Loan, *Matrix Computations*, Johns Hopkins, Baltimore, third ed., 1996.

[6] K. Gröchenig, *Reconstruction algorithms in irregular sampling*, Math. Comp., 59 (1992), pp. 181–194.

[7] K. Gröchenig, *A discrete theory of irregular sampling*, Lin. Alg. and Appl., 193 (1993), pp. 129–150.

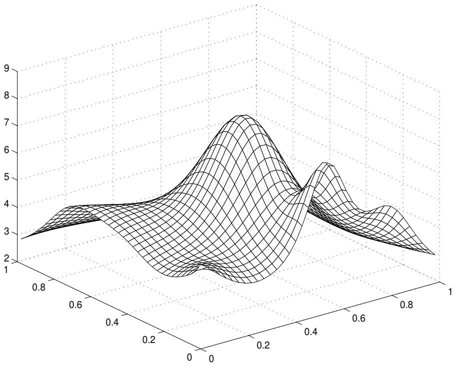
[8] K. Gröchenig and T. Strohmer, *Numerical and theoretical aspects of non-uniform sampling of band-limited images*, in Theory and Practice of Nonuniform Sampling, F. Marvasti, ed., Kluwer/Plenum, 2001.

[9] G. H. Hardy, J. E. Littlewood, and G. Pólya, *Inequalities*, Cambridge University Press, Cambridge, 1952.

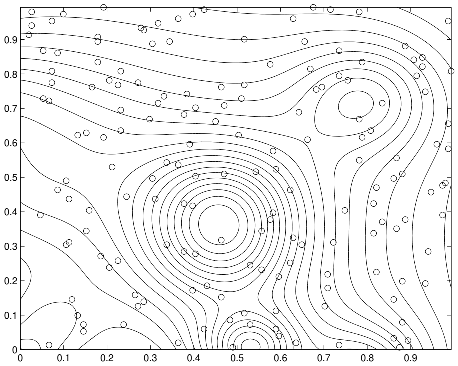
[10] G. Heinig, *Chebyshev-Hankel matrices and the splitting approach for centrosymmetric Toeplitz-plus-Hankel matrices*, Linear Algebra Appl., 327 (2001), pp. 181–196.

[11] G. Heinig and K. Rost, *Representation of Toeplitz-plus-Hankel matrices using trigonometric transformations with applications to fast matrix-vector multiplication*, Linear Algebra Appl., 275–276 (1998), pp. 225–248.

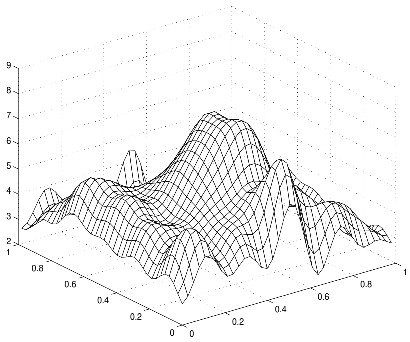
[12] T. Kailath and A. Sayed, *Fast Reliable Algorithms for Matrices with Structure*, SIAM, Philadelphia. PA, 1999.



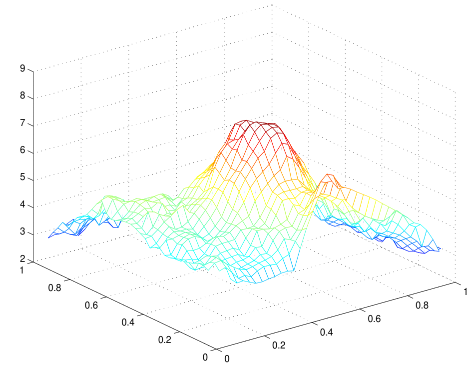
(a) Synthetic gravity anomaly, gravity is in mGal.



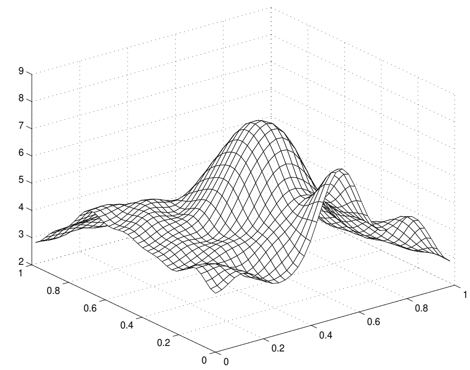
(b) Contour plot of gravity anomaly; sampling locations are marked as "o".



(c) ACT method, error = 0.072



(d) Cubic splines, error = 0.045



(e) Proposed method, error = 0.029

FIG. 5.1. *Approximation of gravitational anomaly from noisy scattered data (5% noise) by proposed method and comparison to standard algorithms.*

[13] M. K. Ng, R. H. Chan, and W.-C. Tang, *A fast algorithm for deblurring models with Neumann boundary conditions*, SIAM J. Sci. Comput., 21 (1999), pp. 851–866 (electronic).

[14] D. Potts, *Fast algorithms for discrete polynomial transforms on arbirtrary grids*, 2001. preprint.

[15] D. Potts, G. Steidl, and M. Tasche, *Fast Fourier transforms for nonequispaced data: a tutorial*, in Modern Sampling Theory: Mathematics and Applications, J. Benedetto and P. Ferreira, eds., Birkhäuser, 2001, pp. 247–270.

[16] M. Rauth and T. Strohmer, *Smooth approximation of potential fields from noisy scattered data*, Geophysics, 63 (1998), pp. 85–94.

[17] L. Reichel, G. Ammar, and W. Gragg, *Discrete least squares approximation by trigonometric polynomials*, Math. Comp., 57 (1991), pp. 273–289.

[18] V. Sanchez, P. Garcia, A. Peinado, J. Segura, and A. Rubio, *Diagonalizing properties of the discrete cosine transform*, IEEE Trans. Sig. Proc., 43 (1995), pp. 2631–2641.

[19] R. Schaback, *Multivariate interpolation and approximation by translates of a basis function*, in Approximation theory VIII, Vol. 1 (College Station, TX, 1995), World Sci. Publishing, River Edge, NJ, 1995, pp. 491–514.

[20] O. Scherzer and T. Strohmer, *A multi–level algorithm for the solution of moment problems*, Num.Funct.Anal.Opt., 19 (1998), pp. 353–375.

[21] G. Steidl, *Fast radix-p discrete cosine transform*, Appl. Algebra Engrg. Comm. Comput., 3 (1992), pp. 39–46.

[22] G. Strang, *The discrete cosine transform*, SIAM Rev., 41 (1999), pp. 135–147 (electronic).

[23] T. Strohmer, *A Levinson-Galerkin algorithm for regularized trigonometric approximation*, SIAM J. Sci. Comp., 22 (2000), pp. 1160–1183.

[24] ———, *Numerical analysis of the non-uniform sampling problem*, J. Comp. Appl. Math., 122 (2000), pp. 297–316.

[25] C. van Loan, *Computational Frameworks for the Fast Fourier Transform*, SIAM, 1992.