

## A fast Hermite transform<sup>☆</sup>

Gregory Leibon<sup>a</sup>, Daniel N. Rockmore<sup>a,b,\*</sup>, Wooram Park<sup>c</sup>, Robert Taintor<sup>a</sup>,  
Gregory S. Chirikjian<sup>c</sup>

<sup>a</sup> Department of Mathematics, Dartmouth College, Hanover, NH 03755, United States

<sup>b</sup> Department of Computer Science, Dartmouth College, Hanover, NH 03755, United States

<sup>c</sup> Department of Mechanical Engineering, The Johns Hopkins University, 3400 North Charles St., Baltimore, MD 21218, United States

### ARTICLE INFO

#### Keywords:

Generalized Fourier transform  
Hermite transform  
Orthogonal polynomial transform  
Three-term recurrence  
Tomographic reconstruction  
Protein structure

### ABSTRACT

We present algorithms for fast and stable approximation of the Hermite transform of a compactly supported function on the real line, attainable via an application of a fast algebraic algorithm for computing sums associated with a three-term relation. Trade-offs between approximation in bandlimit (in the Hermite sense), and size of the support region are addressed. Numerical experiments are presented that show the feasibility and utility of our approach. Generalizations to any family of orthogonal polynomials are outlined. Applications to various problems in tomographic reconstruction, including the determination of protein structure, are discussed.

© 2008 Elsevier B.V. All rights reserved.

### 1. Introduction

In this paper, we present a modification and application of an algebraic algorithm for the efficient computation of a certain class of discrete polynomial transforms. A preliminary version of this paper first appeared in [20]. In particular, we modify a theorem of Driscoll, Healy, and Rockmore [10] in order to find an efficient algorithm for the computation of a certain kind of discrete *Hermite transform* (see below) for the general purposes of tomography, and in particular, for accomplishing a key step in an algorithm for the efficient determination of protein structure from a so-called *tilt series* (cf. Section 6).

The classical two-dimensional tomography problem performs the reconstruction of a two-dimensional density from a collection of measured one-dimensional projections. Most computational approaches to the general problem use standard Fourier analysis and the Fast Fourier Transform (see e.g., [16] and the references therein). The availability of a fast algorithm, such as the FFT can drive the choice of the Fourier basis, but for some problems, it may be more natural to use another basis for  $L^2(\mathbb{R}^n)$ , making necessary the consideration of new fast algorithms in order to make these techniques feasible. One such case is the use of the basis of Hermite functions and their tensor products.

The *Hermite transform* expresses a function  $f(x) \in L^2(\mathbb{R})$  in the complete orthonormal  $L^2(\mathbb{R})$ -basis of *Hermite functions*  $\{\psi_n(x)\}_{n=0}^\infty$  where

$$\psi_n(x) = (h_n)^{-\frac{1}{2}} e^{-\frac{x^2}{2}} H_n(x) \quad (1)$$

and  $h_n = 2^n n! \sqrt{\pi}$  provided  $H_n(x)$  is the  $n$ th *Hermite polynomial*. The Hermite polynomials are determined by the three-term recurrence relation

<sup>☆</sup> This work is supported by NIH, grant number NIH R01GM075310-02.

\* Corresponding author at: Department of Mathematics, Dartmouth College, Hanover, NH 03775, United States.

E-mail addresses: [gleibon@gmail.com](mailto:gleibon@gmail.com) (G. Leibon), [rockmore@math.dartmouth.edu](mailto:rockmore@math.dartmouth.edu) (D.N. Rockmore), [wpark7@jhu.edu](mailto:wpark7@jhu.edu) (W. Park), [rtaintor@gmail.com](mailto:rtaintor@gmail.com) (R. Taintor), [gregc@jhu.edu](mailto:gregc@jhu.edu) (G.S. Chirikjian).

URL: <http://www.cs.dartmouth.edu/~rockmore> (D.N. Rockmore).

$$\mathbf{H}_{n+1}(x) = 2x\mathbf{H}_n(x) - 2n\mathbf{H}_{n-1}(x) \quad (2)$$

with initial conditions  $\mathbf{H}_{-1}(x) = 0$ ,  $\mathbf{H}_0(x) = 1$ .

Thus, any  $f \in L^2(\mathbb{R})$  can be represented as

$$f(x) = \sum_{n=0}^{\infty} \hat{f}(n) \psi_n(x)$$

where  $\hat{f}(n)$  is the  $n$ th Hermite coefficient and is defined by

$$\hat{f}(n) = \langle f, \psi_n \rangle = \int_{-\infty}^{\infty} f(x) \psi_n(x) dx. \quad (3)$$

The Hermite functions have the useful property that they are eigenfunctions for the Fourier transform (see [36] for various details of the Hermite functions). Moreover, in higher dimensions (in which case we mean the product of Hermite functions in the independent variables) the Hermite transform behaves well under rotation: given  $f \in L^2(\mathbb{R}^3)$  and a given  $R \in SO(3)$ , then if  $f$  is “Hermite-limited” (i.e., has a finite expansion in terms of Hermite functions), then  $Rf$  (defined by  $Rf(x) = f(R^{-1}x)$ ) is also Hermite-limited. This is effectively a consequence of the fact that the Hermite functions are also eigenfunctions for the Laplacian (in any number of dimensions), and that the Laplacian commutes with rotation. Thus, there is a relatively simple transformation between coefficients that depends only on  $R$  [30].

Our interest in the Hermite transform arises from this nice rotational invariance, as it applies to a particular problem in protein structure determination (cf. Section 6). In real applications such as this, there are many considerations. In particular, questions of discretization and then questions of efficient and reliable computation are paramount. Thus, we have two algorithmic goals:

- (1) *Analysis*: computation of a discrete Hermite transform, realized from a sampling of the real line that takes a finite collection of function values (sample values) and returns Hermite coefficients.
- (2) *Synthesis*: computation of the function values at a discrete set of points from a finite collection of Hermite coefficients.

Analysis and synthesis are sometimes also referred to as the “forward” and “inverse” transforms, respectively.

In this paper, we explore aspects of the related notions of forward and inverse discrete Hermite transforms. In Section 2, we derive efficient algorithms for the forward and inverse transforms:  $O(M \log^2 M)$  algorithms for computing both the forward and inverse discrete Hermite transform from a set of  $O(M)$  sample points. Note that in its extension to  $n$  dimensions (the cases of  $n = 2, 3$  provide the motivating applications for this work), this amounts to algebraic algorithms that take advantage of a three-term recurrence that require  $O(M^n \log^2 M)$  operations versus  $O(M^{2n})$  operations, if accomplished directly (i.e., “naively”). These complexity results are augmented by an investigation of numerical aspects of the problem. In Section 3, we take into account that the situation of interest is one in which the function  $f$  has compact support, whose range of definition is essentially arbitrary. We investigate some of the trade-offs that arise in the accompanying flexibility in the choice of range of integration. This is critical for the applications of interest in which the data is clearly limited in space, but does not come with a fixed range of support (i.e., there is no a priori fixed set of units that make it necessary to use one grid over another). Section 4 provides an error analysis of these numerical methods. We have implemented our algorithm (see [1] to obtain the software) and a discussion of some standard numerical experiments is given in Section 5. Finally, in Section 6 we sketch the applications which these efficient transforms were designed to facilitate. In particular, we close with some 2D experiments using simulated EM image data for a particular protein complex. The success of this last experiment is evidence for the utility of this approach.

## 2. Efficient transforms

Our concern is the efficient computation of a discrete Hermite transform for functions of compact support. If we absorb the (truncated) exponential factor into the function of interest, the assumption of compact support allows us to reinterpret the integrals (3). We then instead find ourselves in the situation in which we seek to express a function  $f \in L^2([a, b])$  in terms of a complete orthonormal basis constructed via any set of orthogonal polynomials, the precise set-up of which can be found in [5]. Thus, in this section, we present a method for efficiently transforming functions in  $L^2([a, b])$  into a complete orthonormal bases constructed via any orthogonal polynomials. We shall call such a process an *orthogonal polynomial transform* and the associated coefficients the *orthogonal transform coefficients*.

The orthogonal polynomial bases are of the form  $\{\psi_n\}_{n=0}^{\infty}$  where

$$\psi_n(x) = \sqrt{w(x)} p_n(x) \quad (4)$$

with  $w(x)$  the *weight function* and the  $\{p_n\}_{n=0}^{\infty}$  are a collection of *orthogonal polynomials* which we can view as determined by a recurrence relation

$$p_{n+1}(x) = (a_n x + b_n) p_n(x) - c_n p_{n-1}(x) \quad (5)$$

with initial conditions  $p_{-1}(x) = 0, p_0(x) = d_0$ . Hence,  $f \in L^2([a, b])$  can be represented as

$$f(x) = \sum_{n=0}^{\infty} \hat{f}(n) \psi_n(x)$$

where

$$\hat{f}(n) = \langle f, \psi_n \rangle = \int_a^b f(x) \psi_n(x) dx.$$

Notice we are using the convention where we have absorbed the normalizing constants into the polynomials. Our definition of approximate discrete Hermite transform extends immediately to this setting, and we now assume a Hermite basis in this absorbed form. In the absorbed form the normalized Hermite polynomials satisfy  $d_0 = \frac{1}{\pi^{1/4}}$  and the three-term relation

$$p_{n+1}(x) = x \sqrt{\frac{2}{n+1}} p_n(x) - \sqrt{\frac{n}{n+1}} p_{n-1}(x).$$

For most numerical applications, this absorbing of the constants is necessary to control overflow. A standard source for numerical approaches to this can be found in the LAPACK library.<sup>1</sup>

We will need to make use of the following theorem from [10]:

**Theorem 1.** Suppose we have a collection of numbers  $\{g_{m,l}\}_{l,m=0}^{M-1}$  that satisfy the three term relation

$$g_{m+1,l} = (a_m x_l + b_m) g_{m,l} + c_m g_{m-1,l},$$

then for any  $\{f_l\}_{l=0}^{M-1}$  we can compute the collection

$$\left\{ \sum_{l=0}^{M-1} f_l g_{m,l} \right\}_{m=0, \dots, M-1} \quad (6)$$

in  $O(M \log^2 M)$  operations via a linear straight-line algorithm. Furthermore, the stability of this computations can be improved if  $x_l$  is a linear function of  $l$ .

The proof of Theorem 1 takes advantage of the natural divide-and-conquer approach (or dynamic programming) implicit in the three-term recurrence – that is, a polynomial of a given degree can be expressed using linear combinations of suitably shifted lower degree polynomials. The linear nature of the relation enables this to be expressed in terms of a product of sparse, structured matrices so that the matrix-vector product implicit in the calculation of (6) can be written as a product of sparse structured matrices with an arbitrary vector input. This is akin to the factorization of the discrete Fourier transform matrix that is implicit in many of the FFTs (e.g., Cooley-Tukey) and its generalizations (see [23] for a discussion and references) and it gives the increased efficiency (and the straight-line algorithm). See [10] for details.

### 2.1. The Gauss quadrature transform

Functions of the form  $\sum_{n=0}^{M-1} f_n \psi_n$  (with  $\psi_n$  as in (4)) will be called *bandlimited* with *bandlimit*  $M$  (suppressing the actual orthogonal system when it is not a point of confusion). The functions with bandlimit  $M$  are precisely those functions in the form  $p(x) \sqrt{w(x)}$ , where  $p(x)$  is a polynomial of degree  $\leq M-1$ . When dealing with a function of the form  $q(x)w(x)$  with  $q(x)$  a polynomial of degree less than  $2M-1$ , the fundamental theorem of Gauss quadrature (see e.g., [7]) asserts that

$$\int_a^b q(x) w(x) dx = \sum_{k=0}^{M-1} q(x_k) w_k$$

where the  $\{x_k\}_{k=0}^{M-1}$  are the abscissae of the roots of  $p_M(x)$  and the  $\{w_k\}_{k=0}^{M-1}$  are the associated *weights*. A careful description of these quantities and how to compute them is presented in [32].

Thus, while in some cases the quadrature formula will be exact, more generally, a natural (possibly) approximate discrete transform is given by

$$\widehat{f_{g,M}}(n) = \widehat{f_g}(n) = \sum_{k=0}^{M-1} f(x_k) g_n(x_k). \quad (7)$$

Here, the subscript  $g$  indicates that this is a Gaussian quadrature summation with quadrature points dictated by the bandlimit  $M$  (which we will suppress at times). Also  $g_n(x_k) = \frac{w_k p_n(x_k)}{\sqrt{w(x_k)}}$ . The *Gauss Quadrature Transform* (of bandlimit  $M$ ) is then the collection of summations (inner products)

<sup>1</sup> See <http://www.netlib.org/lapack>.

$$\{\widehat{f_{g,M}}(n) | 0 \leq n < M\}.$$

In the particular case in which the  $p_n(x)$  are the (normalized) Hermite polynomials, we call this the *Hermite Gaussian Quadrature Transform* of bandlimit  $M$ .

For a function in the form  $f(x) = p(x)\sqrt{w(x)}$  with bandlimit  $M$  we have

$$\begin{aligned}\widehat{f_g}(n) &= \sum_{k=0}^{M-1} f(x_k) g_n(x_k) \\ &= \sum_{k=0}^{M-1} w_k p(x_k) p_n(x_k) \\ &= \int_a^b f(x) \psi_n(x) dx \\ &= \widehat{f}(n).\end{aligned}$$

Now we apply [Theorem 1](#) to obtain the following:

**Theorem 2.** *With the notation above (which assumes a family of orthogonal polynomials), the Gauss Quadrature Transform of bandlimit  $M$ ,  $\widehat{f_{g,M}}$  can be computed in  $O(M \log^2 M)$  operations. If  $f$  has bandlimit  $M$ , then (in exact arithmetic) this is the exact computation of the orthogonal transform coefficients of  $f$  in the basis of orthogonal polynomials of degree at most  $M$ . In particular, this holds for the Hermite Gauss Quadrature transform.*

The fact that we have used a linear straight-line algorithm for the discrete Hermite transform allows us to apply *Tellegen's principle* (see Theorem 13.20, Ch. 13 in [8]) in order to compute efficiently the transpose

$$\sum_{n=0}^{M-1} h(n) g_n(x_k) \quad 0 \leq k < M. \quad (8)$$

Thus, in the case in which  $h(n) = \widehat{f_g}(n)$  (8) becomes

$$\sum_{n=0}^{M-1} \widehat{f_g}(n) g_n(x_k) = \frac{w_k}{w(x_k)} \sum_{n=0}^{M-1} \widehat{f_g}(n) \psi_n(x_k) \quad (9)$$

which is precisely the evaluation of the function determined by the orthogonal transform coefficients computed by the Gauss Quadrature Transform (7), i.e., it is the *inverse (Gauss quadrature) transform* (evaluation) determined by the forward transform. Note that in the case of a function of bandlimit  $M$ , this is an actual inverse (when evaluated on the sample points).

After an  $O(M)$  modification of (9) by factoring out the ratio  $\frac{w_k}{w(x_k)}$ , we have our next result:

**Theorem 3.** *With the notation above (which assumes a family of orthogonal polynomials), the inverse Gauss quadrature transform (of order or bandlimit  $M$ ) can be computed in  $O(M \log^2 M)$  operations. In particular, this holds for the inverse Hermite Gauss Quadrature Transform.*

The gulf between existence and realization of a fast algorithm (as guaranteed by Tellegen's principle) can be quite large, for while Tellegen's principle is, in principle, constructive, implementation is a highly subtle issue (see e.g., [6]) and it is rarely relied upon for practical implementations. In [8] practical rules are developed to help ease the construction, but it is still in general a complicated task. In fact, we have implemented a different approximate inverse transform which we describe in Section 2.3.

**Remark.** It should be noted that in the image processing literature the Hermite transform refers not to our use of the Hermite transform, but rather to the use of the Hermite basis in the local processing of the data as introduced by Martens in [22]. It is unlikely that the discussion here will be relevant in this context, since the number of coefficients used in such a preprocessing step is rather small (for example in using the hierarchical Hermite transform in [21] the author chooses transforms of order at most 3).

## 2.2. The Newton–Cotes transform

Three practical problems arise when implementing the Hermite Gauss Quadrature Transform:

- (1) In applying the algorithm described in [10] to this case, an intermediate step is required which uses the power basis ( $e_\ell(x) = x^\ell$ ) evaluated at the abscissae  $x_k$ . For most choices of orthogonal polynomials this will likely to lead to an unstable computation, since the power basis is notoriously ill-conditioned.
- (2) Real data is unlikely to be in the form of evaluations at the roots of the orthogonal polynomials (indeed, it is quite often the case that it comes at equispaced points in a grid) so that some sort of interpolation step is required, again forcing information to be thrown away with a concomitant further level of approximation and introduction of error.

- (3) The bandlimited functions are very unlikely to represent accurately the functions underlying our real data, and the Gaussian quadrature method itself is well known to be a poor approximation when applied to functions outside the class of those that are bandlimited.

The above issues will be particularly relevant in the case in which one of the endpoints of  $(a, b)$  is infinite, and the data is naturally viewed as compactly supported in some region, as is the case with our image-like data. In this case, it is possible to deal simultaneously with all three of these problems by working instead with  $M_1$  equispaced points  $\vec{x}$  such that  $x_k - x_{k-1} = \frac{C}{M_1}$  for some constant  $C$ . Then the *Hermite Newton–Cotes Quadrature Transform* of bandlimit  $M$  and order  $M_1$  is the computation of

$$\widehat{f_{c,M_1}}(n) = \widehat{f_c}(n) = \sum_{k=0}^{M_1-1} f(x_k) c_n(x_k) \quad 0 \leq n < M, \quad (10)$$

where the subscript  $c$  indicates Newton–Cotes,  $M_1$  gives the number of sample points (suppressed when not confusing), and  $c_n(x_k) = \frac{c_k c}{M} \sqrt{w(x_k)} p_n(x_k)$  (the  $c_k$  depend on the integration technique). For example, the two-point formula gives the trapezoid rule with  $c_0 = 1/2 = c_{M_1-1}$  and  $c_k = 1$  otherwise, while the three-point formula (when  $M_1 = 2K$ ) with  $c_0 = 1/3 = c_{2K}$ ,  $c_{2l} = 2/3$  for  $l \neq 0, K$  and  $c_{2l+1} = 4/3$  gives Simpson's rule.

Again, as an application of [Theorem 1](#) we have

**Theorem 4.** *Let all notation be as above and let  $M_1 = O(M)$ . Assume that the  $p_k$  are a family of orthogonal polynomials. Then the Newton–Cotes Transform (for the family  $p_k$ ) of degree  $M$  can be computed in at most  $O(M \log^2 M)$  operations. In particular, this applies to the Hermite polynomials.*

Note that  $\widehat{f_c}(n)$  satisfies the criteria for stability stated in [Theorem 1](#). This is achieved by making use of the Chebyshev polynomial basis rather than power basis (see [\[10,14\]](#)). In this case we are working on a grid, allowing us to dodge concern (2). Finally, the third concern is alleviated since Newton–Cotes (especially the trapezoid rule) tends to be robust when applied to non-smooth data. We also have the freedom of using more information per orthogonal transform coefficient, by independently fixing the number of sample points  $M_1$  and the bandlimit  $M$ .

### 2.3. The Euler inverse transform

Here, we give an explicit approximate algorithm for an inverse transform. The approach presented here is fast but is restricted to the case of *classical orthogonal polynomials*, defined as those polynomials  $f(x)$  that satisfy a Sturm–Liouville differential equation of the form

$$Q(x)f'' + L(x)f' + \lambda(n)f = 0$$

(see page 164 of [\[5\]](#)). It also requires significant restrictions on the domain on which the function is compactly supported.

We proceed by using a slight generalization of the classic Euler forward method. This is used to approximate a first order derivative by forward differences. In our case, we wish to incorporate the information contained in both the first and second order derivatives (as described by the Sturm–Liouville differential equation), in order to approximate the values of the  $p_n(x)$  on a discretization  $\{x_0 < \dots < x_{M_1-1}\}$  of the support. Letting  $dx_i = x_i - x_{i-1}$ ,  $Q(x_i) = Q_i$ ,  $L(x_i) = L_i$ ,  $\alpha = 2 - \frac{dx_i L_i}{Q_i} - \frac{dx_i^2 \lambda_n}{Q_i}$ , and  $\beta = \left(-1 + \frac{dx_i L_i}{Q_i}\right)$  then a generalization of the Euler forward method applied to this equation yields

$$p_n(x_{i+1}) = \alpha p_n(x_i) + \beta p_n(x_{i-1}) + O(dx_i^3). \quad (11)$$

We assume here, that in the implied asymptotic, the  $dx_i$  are of the same order. By ignoring the error term, this relation allows us to recursively approximate the  $p_n(x)$  via a three-term recursion relation.

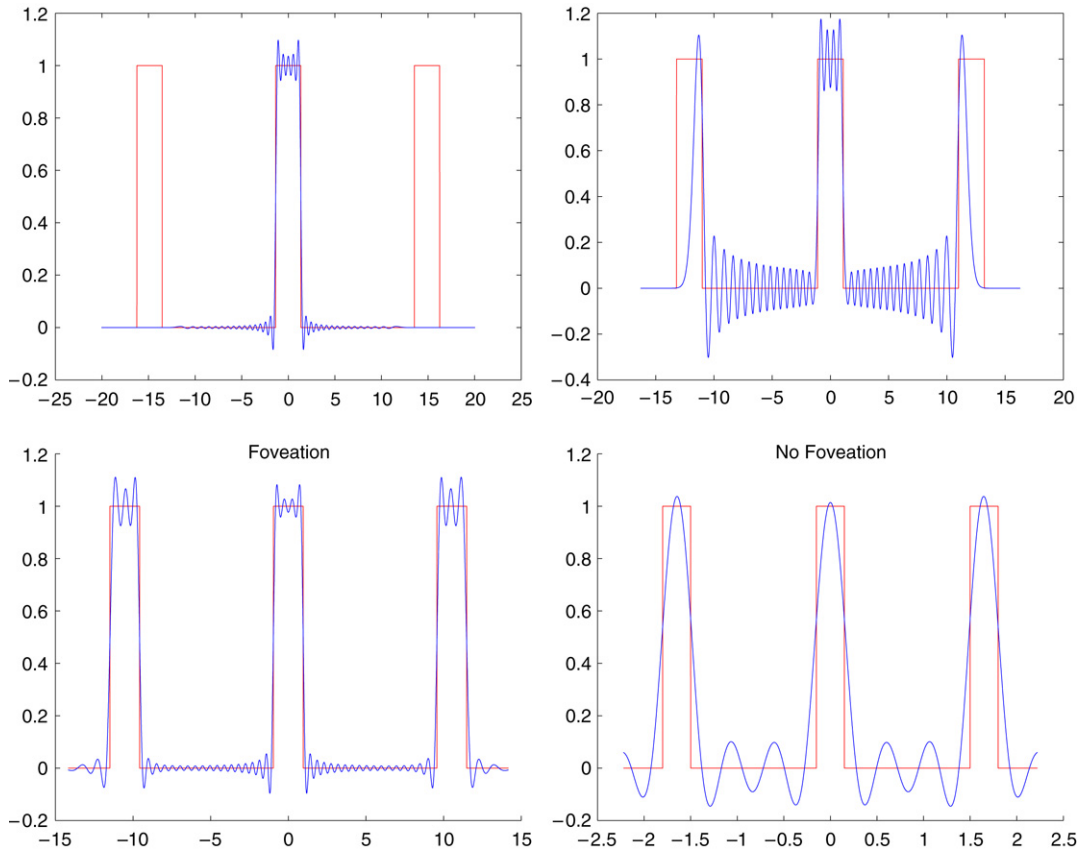
We now introduce the following notation. We let  $\pi_M(f)$  denote the projection  $\sum_{k=0}^{M-1} \widehat{f}(n) \psi_n$  for any  $f \in L^2(\mathbf{R})$ . I.e.,  $\pi_M$  represents the projection onto the subspace spanned by the first  $M$  elements in our orthonormal basis. When there is no worry for confusion, we let  $\widehat{f}_M$  denote the orthogonal transform coefficients of  $\pi_M(f)$ .

Let  $\tilde{p}_n(x_k)$  denote the approximation to  $p_n(x_k)$  given in (11). Then using this, we can hope to approximate the inverse transform (evaluation) at any set of points  $\vec{x} = (x_0, \dots, x_{M_1})$ , from a set of  $M$  orthogonal transform coefficients, denoted as  $(\widehat{f}_M)^\vee(\vec{x})$  via the *Euler Inverse Transform* denoted  $(\widehat{f}_M)^\vee_e(\vec{x})$  defined by

$$(\widehat{f}_M)^\vee_e(x_k) = \sum_{n=0}^{M-1} \widehat{f}_M(n) e_{x_k}(n).$$

where the subscript  $e$  indicates the use of the Euler forward method (11) and  $e_{x_k}(n) = \sqrt{w(x_k)} \tilde{p}_n(x_k)$ .

Once again, we can apply [Theorem 1](#), this time to compute  $(\widehat{f}_M)^\vee_e(\vec{x})$ : assuming  $M_1 = O(M)$  we can compute  $(\widehat{f}_M)^\vee_e(\vec{x})$  in  $O(M \log^2 M)$  operations by first computing the sums  $\sum_{n=0}^{M-1} \widehat{f}_M(n) \tilde{p}_n(x_k)$  (using a fast transform as described by either [Theorem 2](#) or [4](#)) and then performing the multiplication by the  $\sqrt{w(x_k)}$  (requiring another  $O(M)$  operations). Thus, we have the following theorem.



**Fig. 1.** Here we approximate the function  $f(x)$  composed of three contiguous boxcars. Each of the graphs is a rescaled version  $f(cx)$ . For each choice of  $c$ , the projection  $\pi_{100}(f(cx))$  overlays the boxcars  $f(cx)$ . Notice the range of behaviors of the approximation.

**Theorem 5.** With the notation as above, assume that  $M_1 = O(M)$ . Then for any set of  $M_1$  sample points  $\vec{x}$ , we can compute the Euler (approximate) inverse transform  $(\hat{f}_M)_e(\vec{x})$  in  $O(M \log^2 M)$  operations.

Notice that we are in the stable case, provided that  $\lambda_n$  is a linear function of  $n$ . We postpone a discussion of convergence to the exact answer to Section 3.

### 3. Support intervals and template functions

For the purposes of applications, we are primarily interested in functions of compact support, defined on either  $\mathbb{R}^2$  or  $\mathbb{R}^3$  which via tensor products we reduce to the one-dimensional case. Since our interest is in the analysis of image or electron density data, a *realistic* class of functions would be ones which are compactly supported on an interval  $[-A, A]$  and piecewise smooth, with a bounded number of pieces. A much less realistic, but easy to handle class, would be the Hermite-bandlimited functions of bandlimit  $M$ .

In practice, the attractive analytic qualities of a bandlimited function are often overshadowed by the fact that realistic functions can be quite different from bandlimited functions, and serious choices must be made in deciding how to re-scale a realistic function so that it is best captured by a bandlimited function.

For example, in Fig. 1 we see how scale affects our view of a realistic function by examining  $\pi_{100}(f(cx))$  for various choices of  $c$ . The bottom right shows the basic function, a sequence of three boxcars that fit within the closed interval  $[-2.5, 2.5]$ . The three different scalings then fit this into  $[-15, 15]$ ,  $[-20, 20]$  and  $[-25, 25]$ . Notice how the degree of approximation varies. From our point of view, a good choice of  $c$  will correspond to a good choice of support interval  $[-A, A]$ .

One way to deal with this re-scaling problem, is to try and pick a support interval suited to the analysis. To help motivate the choices involved, it is useful to recall the behavior of the Hermite functions. As we see in Fig. 2,  $\psi_n$  has an interval  $[-S_n, S_n]$  where it is pseudoperiodic, and a complementary interval where  $\psi_n$  decreases monotonically and asymptotically as  $\frac{2^n}{\sqrt{h_n}} x^n e^{-x^2/2}$ . There are several ways to estimate  $S_n$ . Following [17], the last maximum  $S_{n,1} = \arg(\max(\psi_n))$  can be used, or as in [25], the beginning of the tail is another reasonable choice, giving  $S_{n,2,\epsilon} = \max(\arg(\psi_n) = \epsilon)$ . We choose to use the final inflection point of  $\psi_n$ ,  $S_n = \sqrt{2n+1}$ , since it is easy to remember, compute, and it satisfies  $S_{n,1} \leq S_n \leq S_{n,2,\epsilon}$  for any



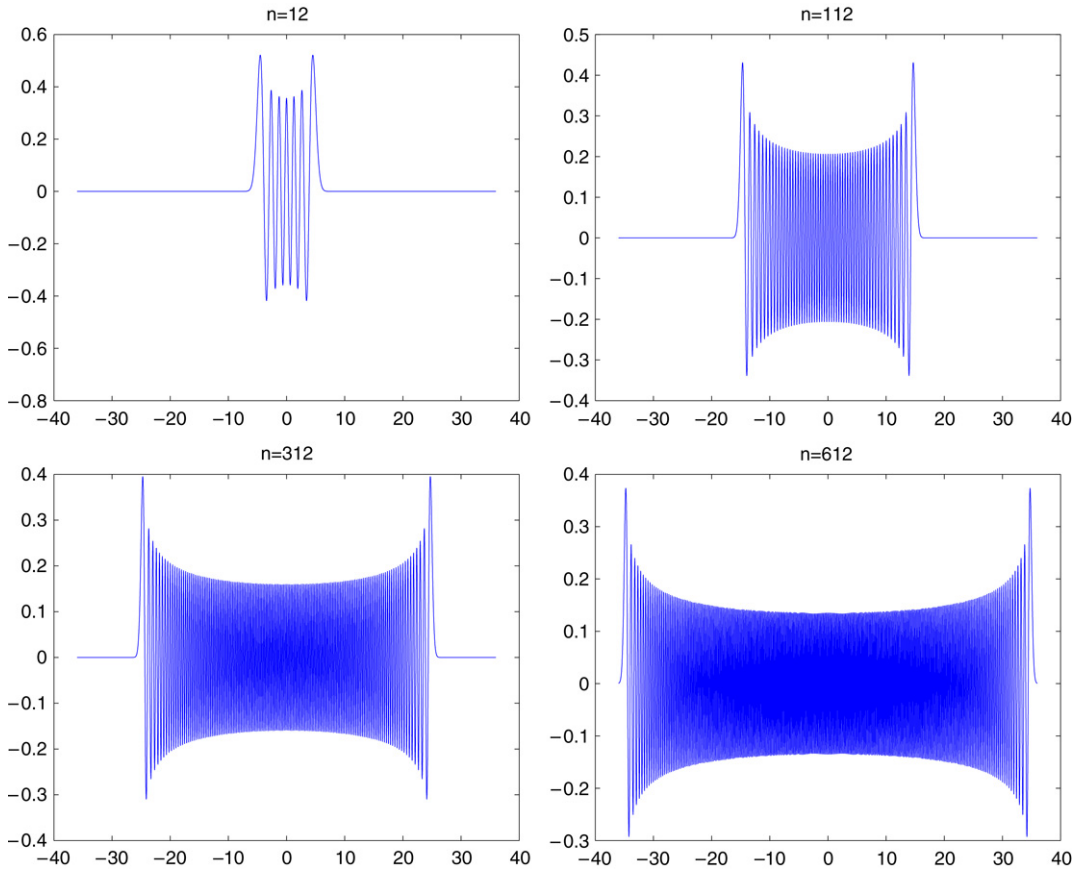


Fig. 2. Here we see  $\psi_n(x)$  with  $S_n \in \{5, 15, 25, 35\}$ .

reasonable choice of  $\epsilon$ . In Fig. 2, we see  $\psi_n$  for  $n = 12, 112, 312, 612$  with corresponding support intervals having endpoints  $S_n = 5, 15, 25, 35$  units from the origin.

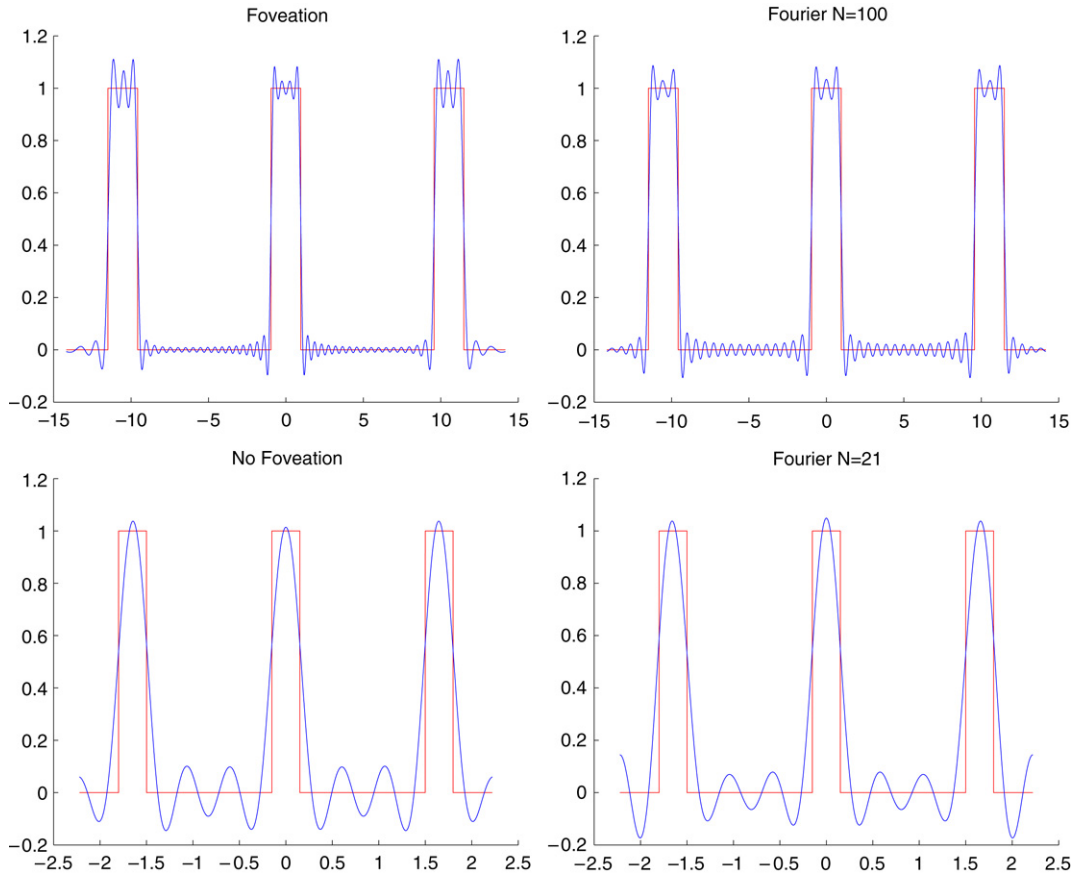
Thus, given a choice of bandlimit,  $M$ , a natural choice of  $A$  is  $A = S_{M-1}$ . Intervals of roughly this length have the advantage of using the vast majority of the information contained in the Hermite functions, but a potential liability is that this information is compressed near the origin, where all the Hermite functions can contribute to the approximation.

This compression of information motivates the use of the Hermite functions in the study of *foveated* images. This is a non-uniform resolution of an image whose resolution is highest at certain point, called the *fovea*. In Fig. 1, we see that in the top two scaled versions, our bandlimited version of the image is extremely foveated, while it is not at all foveated in the version at the bottom right, and it is only slightly foveated in the bottom left where  $A = S_{M-1}$ .

Foveation can be an asset (e.g., [19]), but if a more uniform resolution is required, then it may be necessary to choose an interval shorter than  $S_{M-1}$ . In an extreme case, consider a fixed interval, i.e., in which  $A$  is independent of  $M$ . We choose to use the points within  $\pi$  standard deviations from the mean using the density function  $|\psi_0(x)|^2$ . Hence  $A = \frac{\pi}{\sqrt{2}}$ . Thinking of the Gaussian  $|\psi_0(x)|^2$  as our archetypical template function, we see that this density is indeed “almost” compactly supported on this interval and that any larger interval would be something of a waste from this vantage point. The exact choice of  $\pi$  was to highlight certain estimates and features, most notably that this Hermite transform on this interval compares with the Fourier transform with frequencies less than  $\sqrt{M}$  in absolute value. Hence, this partial series utilizes roughly  $2\sqrt{M} + 1$  terms of the Fourier expansion.

Fig. 3 provides a comparison for Fig. 1 with the results from using Fourier techniques, comparing the bottom two pictures in Fig. 1 with their corresponding Fourier projections to get a sense for the residual foveation. We now adopt the notation of  $\pi_M(f)$  to denote the projection  $\sum_{k=0}^{M-1} \hat{f}(n) \psi_n$  for any  $f \in L^2(\mathbf{R})$  and  $\pi_M^F(f)$  to denote the analogous Fourier projection. In the top two pictures in Fig. 3, we compare the Hermite projection  $\pi_{100}(f(c_1x))$  with the projection onto the first 100 coefficients of the Fourier basis  $\pi_{100}^F(f)$ , while the bottom two compares  $\pi_{100}(f(c_2x))$  with  $\pi_{21}^F(f)$ . Note that the various expansions do indeed contain comparable information.

In Fig. 4 we compare the Newton–Cotes and Gauss quadrature transform on an example.



**Fig. 3.** In the top row we compare  $\pi_{100}(f)$  and the Fourier projection  $\pi_{100}^F(f)$  when using  $A = S_{100}$ . In the bottom row, we compare  $\pi_{100}(f)$  and the Fourier projection  $\pi_{21}^F(f)$  when using  $A = \frac{\pi}{\sqrt{2}}$ .

#### 4. Convergence

In this section we discuss aspects of convergence and error estimation. Proofs of the two main analytic results (Theorems 6 and 7 can be found in [20]).

##### Error estimation

With respect to the Newton–Cotes method, we now see that we have the freedom of using more information per Hermite coefficient by independently fixing  $M_1$  (the number of sampling points) and  $M$  (the number of coefficients) required for the forward and inverse transforms, respectively. We can also estimate the error (see [20] for a proof).

**Theorem 6.** Let  $M_1$  denote the number of sample points, let  $M$  denote the bandlimit and let  $M = (1/C)M_1$ . For some fixed constant  $C$ , let  $A \leq S_M$ , and let

$$\|f(x)\|_{\infty,k} = \max_{m \leq k} \left\| \frac{d^m}{dx^m} f(x) \right\|_{\infty}.$$

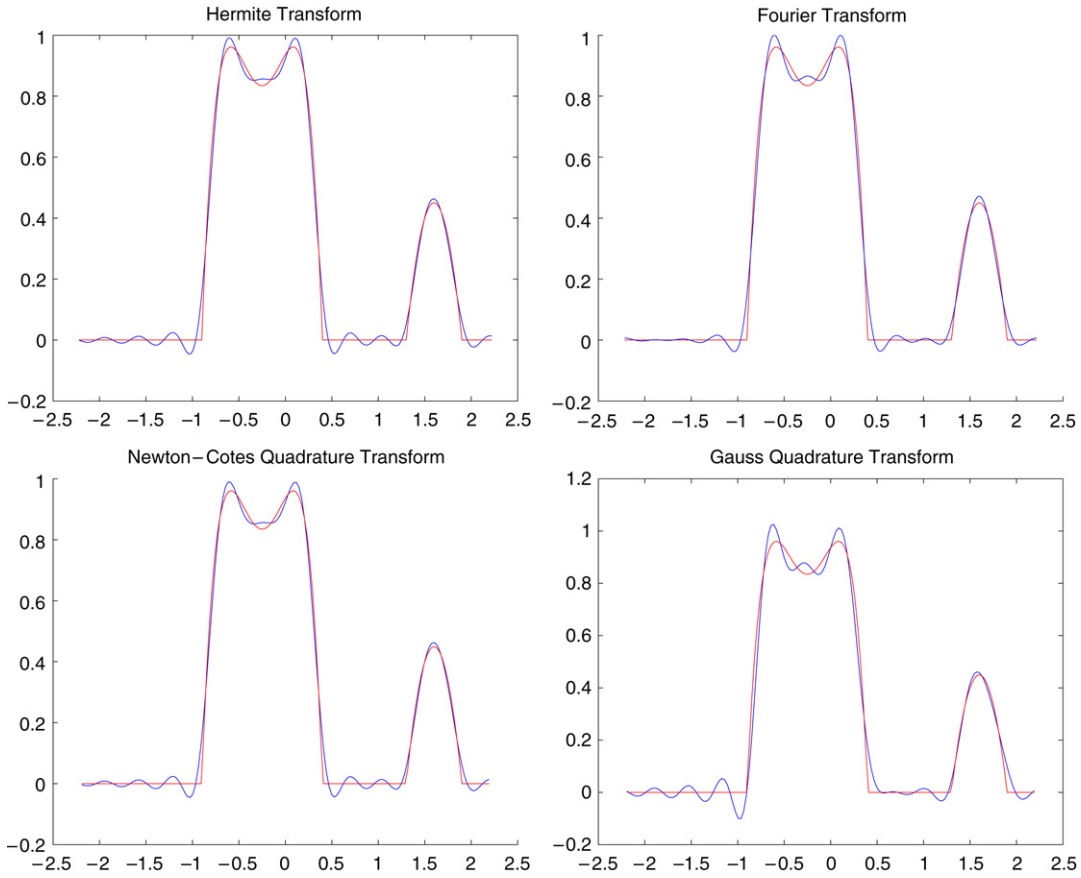
Then

$$\|\hat{f}_c - \pi_M f\|_{L_2(\mathbf{R})} \leq \frac{C_{\epsilon,k} \|f\|_{\infty,k,A}}{M^{(k-2)/2 + 1/12 - \epsilon}}$$

where  $\|f\|_{\infty,k,A}$  is the sup norm of the first  $k$  derivatives, restricted to the interval  $[-A, A]$ .

So we get convergence as  $M$  increases for any  $k$ -point Newton–Cotes method ( $k \geq 2$ ). Notice the convergence from the estimate derived for the trapezoid rule is only on the order of  $M^{-1/12+\epsilon}$ . In practice, it seems that the trapezoid rule does better than this (rather dramatic) overestimates suggests, even when  $M_1 = M$ . This is illustrated in Fig. 5.





**Fig. 4. Comparison of Hermite Gaussian Quadrature, Hermite Newton–Cotes, and Fourier projections.** Let  $M = M_1 = 150$ . Here, we compare  $\pi_M(f)$  in the upper left,  $\pi_{2\sqrt{M}+1}^F(f)$  in the upper right,  $(\hat{f}_E)$  in the lower right and  $(\hat{f}_C)$  in the lower left for  $M_1 = M = 150$  and  $A = \frac{\pi}{\sqrt{2}}$ . Notice if you intend to manipulate the transform, then the Newton–Cotes transforms tend to perform as well as the Gaussian–Quadrature transforms.

#### 4.1. Error analysis and numerical exploration of the Euler inverse transform

In Section 2.3, we gave a fast algorithm for an (approximate) Euler Inverse Transform (Theorem 5). Notice that we are in the stable case, provided that  $\lambda_n$  is a linear function of  $n$ . In Fig. 6, we compare this approximation with the true inverse, using a uniformly spaced set of points.

**Example 1.** The Hermite polynomials are classical with respect to  $L(x) = -2x$  and  $Q(x) = 1$ , and  $\lambda(n) = 2n$ . Since  $\lambda_n$  is a linear function of  $n$ , we are in the stable situation. In the case where  $A_M = A_0$  the above procedure works well. When we utilize the Euler transform, we actually perform it twice, once in the forward direction starting at  $x_1 = 0$  using  $M_1/2$  steps and then in the reverse direction using  $M_1/2$  steps. To avoid overflow problems we need to approximate the initial conditions using Stirling's formula. Namely the initial value is

$$\frac{H_n(0)}{\sqrt{h_n}} = \begin{cases} \frac{(-1)^{n/2}n!}{\pi^{1/4}(n/2)!\sqrt{n!2^{n/2}}} & n \text{ even} \\ 0 & n \text{ odd} \end{cases} \approx \begin{cases} \left( \frac{(-1)^{n/2}2^{1/4}}{\sqrt{\pi}} \right) \frac{1}{n^{1/4}} & n \text{ even} \\ 0 & n \text{ odd} \end{cases}$$

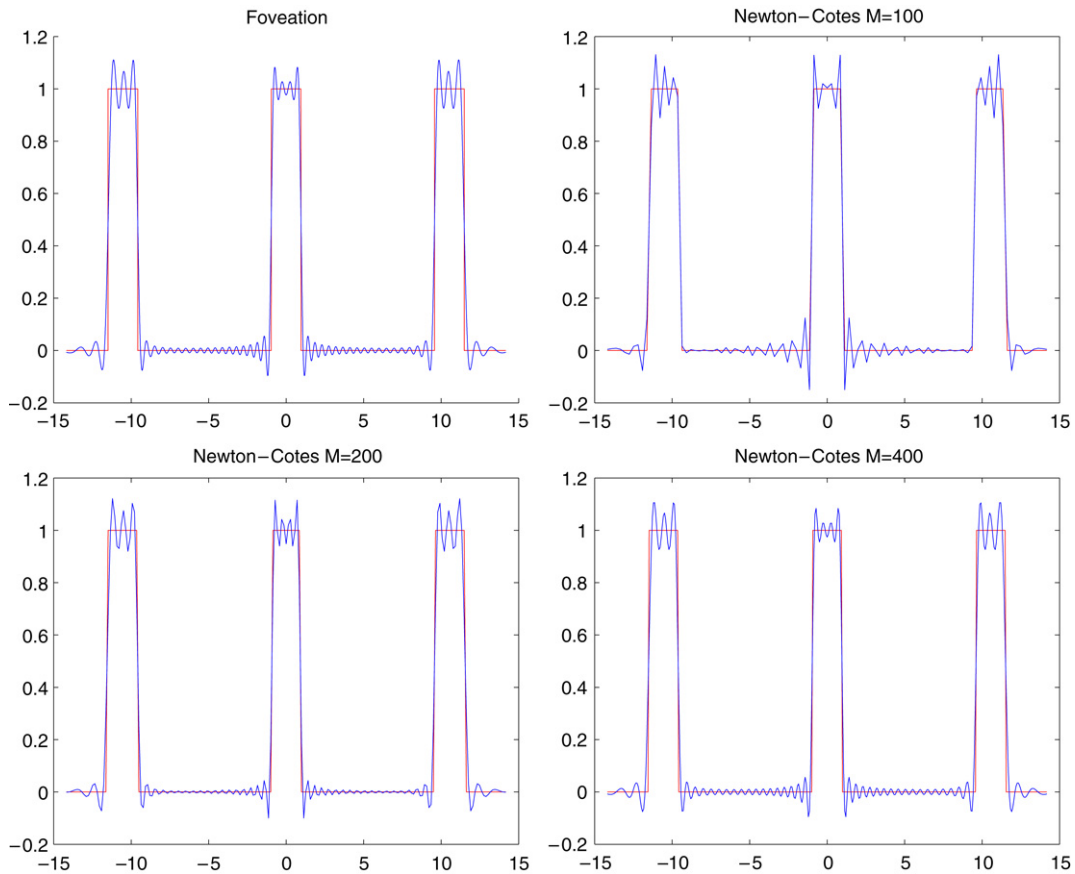
and the initial derivative is

$$\frac{\frac{H_n}{dx}(0)}{\sqrt{h_n}} \approx \begin{cases} \left( \frac{(-1)^{(n+1)/2}2^{3/4}}{\sqrt{\pi}} \right) n^{1/4} & n \text{ odd} \\ 0 & n \text{ even.} \end{cases}$$

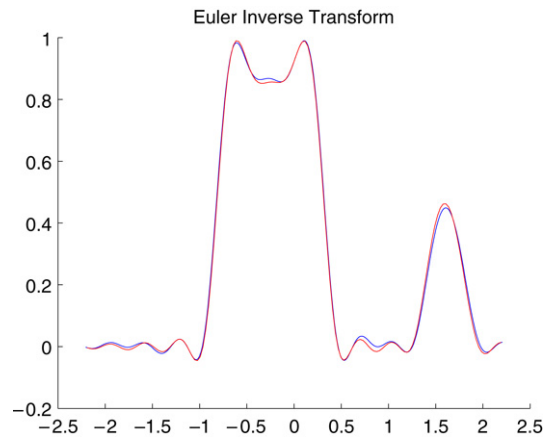
Although Stirling's formula is only an asymptotic approximation, it is nevertheless quite good in general. Higher order approximations could be used if necessary.

A slight improvement in the convergence can be obtained by using the three-term relation, derived from the fact that the  $\psi_n$  satisfy the second order equation

$$\psi_n'' + (\lambda(n) + 1 + x^2)\psi_n = 0.$$

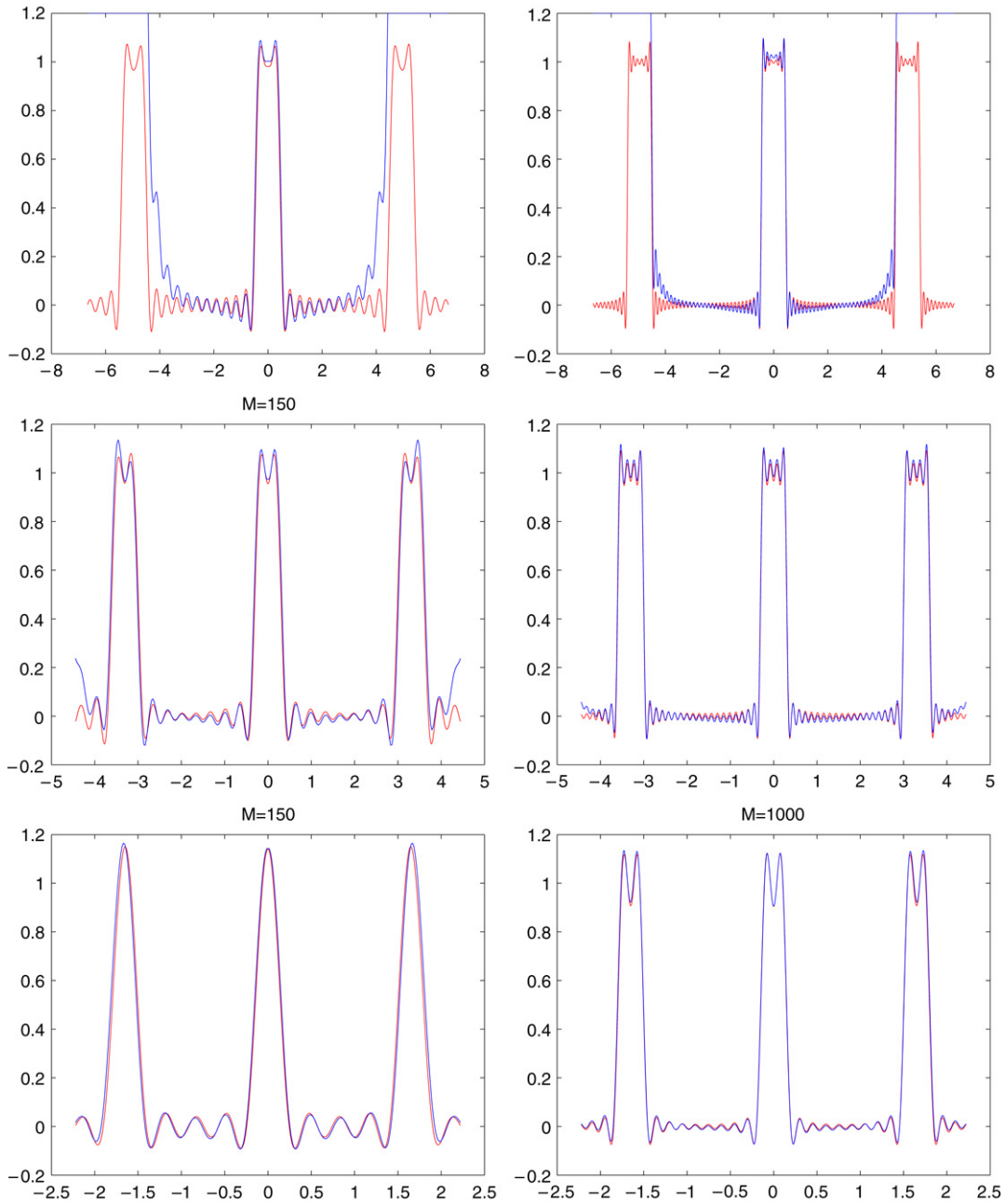


**Fig. 5.** Letting  $M = 100$  and  $A = S_M$ , here we compare  $\pi_M(f)$  in the upper left to the trapezoid rule approximations  $(\hat{f}_c)_e$  for  $M_1 = 100, 200, 400$  in order to get some sense for the improvement obtained by making  $M_1$  larger relative to  $M$ .



**Fig. 6.** Here we compare  $(\hat{f}_c)_e$  in red with  $(\hat{f}_c)$  in blue with  $M = M_1/2 = 150$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Using a fixed constant  $A$ , this approach works well, and below we see that it converges as  $M_1 = CM$  is increased. However the Euler approximation is not robust as the interval  $A$  is lengthened, and even for moderate  $n$  the Euler approximation runs into serious problems when we approximate  $\psi_n(x)$  after the point of inflection at  $\sqrt{2n+1}$ . This puts serious restrictions on those  $A$  which can be used safely. This, in turn, will affect the accuracy of this method. In general, making  $M_1$  larger than  $M$  is an effective way of improving the convergence rate. But even in the worst case scenario where  $M = M_1$ , using the results of [18], we can prove the following convergence (see [20] for a proof):



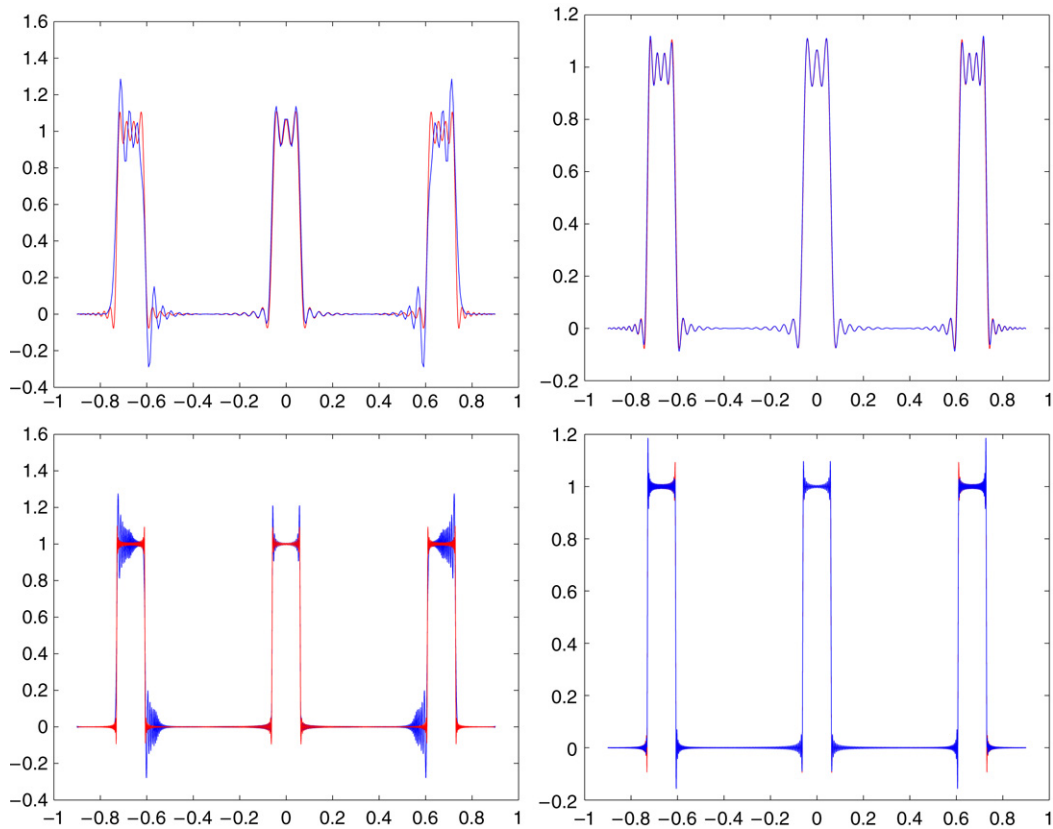
**Fig. 7.** Here we compare  $(\hat{f}_M)_e$  and  $\pi_M(f)$  for three different intervals  $3\frac{\pi}{2}$ ,  $2\frac{\pi}{2}$ , and  $\frac{\pi}{2}$  (in the three rows) and for two different values of  $M = (1/2)M_1$  in the columns ( $M = 150$  and  $M = 1000$ ). In red is  $\pi_M(f)$ . (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

**Theorem 7.** With the notation as above,

$$|(\hat{f}_M)_e(x_k) - \pi_M f(x_k)| \leq \frac{C_\epsilon \|f\|_{L^2(\mathbf{R})}}{M^{1/4-\epsilon}}$$

for a fixed interval.

Notice also that this result is achieved for a fixed interval, and the constant  $C$  depends rather dramatically on the choice of interval. In Fig. 7, we look at a few examples using different support intervals. For each example, we show the dependency on  $M = (1/2)M_1$ , using  $M = 150$  in the first column and  $M = 1000$  in the second.



**Fig. 8.** Here we compare  $(\hat{f}_M)_e$  and  $\pi_M(f)$  in the Legendre case. In red is  $\pi_M(f)$ . The top left is  $150 = M = M_1/2$  while in the top right we multiply this  $M_1$  by 5. In the bottom left is  $1000 = M = M_1/2$  while in the bottom right we multiply this  $M_1$  by 5. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In Fig. 6, we compare  $(\hat{f}_M)_e(\tilde{x})$  and  $(\hat{f}_N)_e$  to see what we found to be a typical phenomena. In comparison with difference between the  $f$  and  $(\hat{f}_M)_e(\tilde{x})$ , the difference between  $(\hat{f}_N)_e$  and  $(\hat{f}_N)_e(\tilde{x})$  is small, allowing us in practice to sensibly use the Euler Inverse Transform.

**Example 2.** In the noncompact (Hermite-like) case, the accuracy of this approximation is very dependent on the choice of interval. To give a sense of the extent to which the non-compactness of the interval plays a role, in Fig. 8 we apply these techniques to the Legendre polynomials which satisfy the Sturm–Liouville equation for  $L(x) = -2x$ ,  $Q(x) = 1 - x^2$ , and  $\lambda(n) = n(n+1)$  and live on the compact interval  $[-1, 1]$ . Using the techniques presented here for estimating the error, one can only demonstrate that the convergence is in distribution if  $M = CM_1$ . Hence, one is forced to increase  $M_1$  relative to  $M$  to improve the convergence. We witness this in Fig. 8.

**Remark.** The techniques described in this section are easily modified to give an efficient algorithm for the computation of expansions in terms of associated Laguerre polynomials (see e.g., [2]). These expansions arise in many areas, but are of particular interest for their use in the description of functions defined on the two-dimensional polar grid (see e.g., [24]). Considerations of compact support are similarly important for applications to image processing, and an analysis such as appears in this paper could be readily adapted to this case.

## 5. Timing and stability experiments

Using the Fast Hermite Newton–Cotes algorithm indicated above, we implemented a C program that performs a fast Newton–Cotes Hermite Transform [1] (denoted in Table 1 as FHT). The software is freely available [1]. In this section we report the results for the efficiency and stability of this implementation.

As discussed above, we have a freedom of choice in the description of the data (i.e., on which square  $[-A, A] \times [-A, A]$  the data live). Thus, for a given bandlimit  $M$ , we let  $A = \sqrt{2M}$  (recall Section 3). Furthermore, let  $n$  denote the number of sample points and define our sampling grid  $(x_i, x_j)$  by

$$x_i = 2A(i/n - 1/2), \quad (12)$$

for  $i = 0, \dots, n$ .

**Table 1**Stability of computation of  $\|H^\dagger H - I\|_2$  for a range of realistic bandlimit (powers of two)

$M$	$n = 10M$	$\ H^\dagger H - I\ _2$	Std. dev.
4	20	3.43E–08	2.57684E–08
8	40	1.00E–08	1.22054E–08
16	80	1.15E–09	1.78611E–09
32	160	3.92E–12	4.12421E–12
64	320	3.73E–12	4.11435E–12
128	640	7.63E–10	1.27733E–09

Note that the calculation of  $H^\dagger$  requires some matrix multiplies as well as the calculation of  $B^{-1}$ . These were all accomplished using the standard matrix arithmetic calls in Matlab. All calculations were performed on a Dell Latitude D800 (Specs: 1.79 GHz Processor, 512 Mb Ram, Windows XP Professional).

Since we are using scaled and “windowed”  $\psi_k(x)$  (i.e., constrained to the interval  $[-A, A]$ ) Hermite functions, the  $\psi_k$  are no longer orthonormal, but are nearly orthonormal.

Let us make some definitions. Let  $P$  denote the  $M \times (n + 1)$  matrix given by the first  $M$  sampled Hermite polynomials,

$$P_{i,j} = p_i(x_j)$$

with  $x_j$  defined as above. Then given a sampled function  $f = (f(x_0), \dots, f(x_n))^t$  the (discrete) Hermite transform is defined as

$$Hf = (dx)PEf \quad (13)$$

where  $E$  is the  $n \times n$  diagonal matrix of sampled exponentials

$$E_{j,j} = e^{-(x_j)^2/2}$$

and  $dx = (2A)/n$ .

We have described a fast algorithm for the computation of  $Pf$ , which then gives a similarly efficient algorithm for the computation of  $Hf$ . In this section, we will examine both the efficiency and stability of this algorithm.

For the stability, we first show that, as suggested, the windowed, scaled, and sampled versions of the Hermite functions  $\psi_k$  do in fact retain their orthonormality. Let  $B = (\frac{1}{dx})HH^T$ . By the orthogonality of the Hermite polynomials,  $B$  will be invertible. From this it then follows directly that

$$H^\dagger = \left( \frac{1}{dx} \right) H^T B^{-1}$$

is the  $(n + 1) \times M$  pseudoinverse of  $H$ . Table 1 shows that for a range of realistic bandlimit ( $M$ ) and comparable number of sample points ( $10M$ )  $H^\dagger$  does, in fact, give a very good approximation to the pseudoinverse.

Table 1 indicates that the following is then a reasonable test of stability and efficiency:

- Do (for fixed  $N$  and  $n$ )
  - . Step 1 (synthesis): Compute  $N$  “random” Hermite coefficients and synthesize a “random” bandlimited  $f$ .
  - . Step 2: Compute  $\|H^\dagger \text{Fast}(H)f - f\|_2$ .
- Until done.

The random coefficients were computed according to a  $N(0, 1)$  distribution, with the  $k$ th Hermite coefficient weighted by  $\frac{1}{k^2}$  to reflect twice differentiability. Once finished, compute the average error as well as the average time for the calculations of  $\text{Fast}(H)f$ . Table 2 shows the results of this experiment.

There are a few things to say about Table 2:

- (1) For  $M < 64$  the results are decidedly stable. At  $M = 64$  the results begin to falter, but are still passable. The trend of increasing error in fact continues to the point that for  $M > 64$  the computation is very unstable. For  $N > 16$  the fast algorithm begins to beat the naïve calculation by a significant amount over many trials. It should also be noted that the timing results improve with increasing  $n$ , due to the efficiency of the Chebyshev step, which transforms the sample points into function space.
- (2) The results we show here are for a one-dimensional transform. Image processing would occur on a two-dimensional grid, so the timings would be squared, indicating an even greater advantage in using the fast transform.
- (3) The program currently works for powers of two. This limitation can be overcome with some simple adaptations according to [10].

**Table 2**

Computation of stability and timings for  $\text{Fast}(H)$  for several realistic problem sizes

$M$	$n$	$L^2$ -Error	St.dev.	FHT time	Naive time
4	20	3.43E–08	2.58E–08	0.1	0.1
8	40	1.00E–08	1.22E–08	0.2	0.3
16	80	1.15E–09	1.79E–09	0.5	1.2
32	160	1.15E–10	7.50E–11	1.6	4.7
64	320	1.98E–04	1.97E–04	5.11	18.12

$L^2$ -error means the average of  $\|H^\dagger \text{Fast}(H)f - f\|_2$  for 100 randomly generated bandlimited functions  $f$  (of bandlimit  $N$ ). This is reported in column 3. Column 4 contains the standard deviation of these numbers. Columns 4 and 5 show the average time in milliseconds for the FHT and direct matrix vector multiply  $Hf$  respectively. The timing results are processor times in tens of milliseconds (hence the recording of a time of 0 s for the small problem sizes) as estimated by the clock function included with the gnu C compiler. All calculations were performed on a Dell Latitude D800.

- (4) The algorithm described in [10] prescribes an initial fast transform onto the Chebyshev polynomials, followed by a slick manipulation of the three-term recurrence relations of both the Chebyshev polynomials and the Hermite polynomials. We decided to perform this initial step naively, due to stability issues with known fast Chebyshev transforms. As a result, we cannot claim that our program performs asymptotically better in this initial step. The Chebyshev transform, however, was so fast and simple to perform, that for useful values of  $M$  the naïve algorithm was perfectly acceptable.
- (5) We only show results up to  $M = 64$  even though in Table 1 the calculation of the pseudoinverse is stable even for  $M = 128$  (and presumably beyond). This reflects the fact that for  $M = 128$ , the calculation for Table 2 was highly unstable (even if very fast!). It is possible that (among a variety of possibilities) our naïve linear scaling of the number sample points may be a source of inaccuracy. It would be of interest to investigate methods for improving the range of stable calculation, and this is a current project of interest.
- (6) The timing results do not include any precomputation involved in the fast algorithm. In a practical implementation (including ours), the results of this step would be saved to the hard disk and brought up each time the program is run. The program would then save these results for successive uses of the fast algorithm. Given many uses of the algorithm, this time would be negligible.

## 6. Potential applications: Medical imaging and cryo-electron microscopy

Multidimensional bandlimited Hermite function expansions are a promising tool in a number of potential applications. Their benefits relative to other bases follow from the properties that:

- (1) A rotated version of a multidimensional bandlimited Hermite function expansion will be of the same form, and with the same bandlimit, as the original unrotated version of the function [27,30]; and
- (2) In the bandlimited case, data sampled on a polar/spherical grid can be converted into data sampled on a Cartesian coordinate system exactly without loss of information [28].

Such situations arise in medical imaging and cryo-electron microscopy. In both of these applications, projections of a density are obtained and one desires to reconstruct the original density from a finite number of these projections. These applications are described in detail below.

### 6.1. Overview of medical imaging and cryo-electron microscopy

In the case of medical imaging, the direction along which projections are taken is known, and methods for reconstruction (using a technique called filtered backprojection) are well established [16]. This method requires interpolation of data from a polar grid in Fourier space onto a Cartesian grid, so that FFT methods can be used. This interpolation step, however, is an approximate one that can lead to loss of information. Using an alternative basis, such as the Hermite functions, has the potential to reduce this loss. A limiting factor is the existence of fast algorithms, such as those discussed in this paper.

Numerous methods exist for the approximate interconversion between Cartesian and polar data. Applications for this conversion include tomography and radar. To our knowledge, the only “exact” method for performing this interconversion is that presented in [28]. The exactness of the interconversion relies on the Cartesian data being bandlimited in the Hermite sense, which corresponds to a bandlimited Laguerre–Fourier expansion in polar coordinates. The interconversion problem is different from the related problem of performing fast transforms in polar coordinates, such as those presented in [3,12,31]. In other words, it is one problem to convert losslessly between bandlimited expansions in two different curvilinear coordinate systems, and it is another problem to derive fast transforms in each of these coordinate systems. Both problems are important, but they are not the same.



Hermite-function expansions in cryo-electron microscopy (cryo EM) potentially can be applied to tomographic reconstruction. In cryo EM, the structure of large protein systems, such as ribosomes or viral capsids are investigated via the use of an electron microscope, which passes an electron beam through the sample, effectively integrating the electron density along the beam line.<sup>2</sup> Sometimes, multiple beams can be passed through each protein structure at different angles, generating what is called a *tilt series*. In this case, reconstruction amounts to precisely the same inverse problem as in tomographic imaging.

Unfortunately, acquisition of the tilt series via the electron beam can actually cause damage to the sample, resulting in a degradation of the data that is collected. It is for this reason that a technique called *single-particle cryo EM* is sometimes used in place of a tilt series. In this method, many copies of the molecular complex of interest are frozen in a thin film of vitreous ice, resulting in a planar suspension of essentially identical copies, arranged at randomized orientations. The resulting two-dimensional film of ice and protein is called a micrograph. Then, using electron beams that pass orthogonally through the plane of the micrograph, a single planar projection is obtained per copy of the molecular complex. The goal is to obtain the electron density from the resulting projection data, the directions of which are random and unknown with respect to each molecular complex. This is a problem that has been studied intensively by electron microscopists and others over the past two decades (see e.g., [15,26]), and for which there are still many open issues. Some effective algorithms have been developed over the years, including contributions from van Heel's group [34,35], and the widely used algorithm of Baker and Cheng [4].

Multidimensional Hermite expansions turn out to be very attractive for the cryo EM problem. One reason, is that, unlike the Fourier basis, the rotation information is isolated from the basis elements in a finite number of coefficients [27,30]. This may help in rapid determination of the unknown projection directions, thereby efficiently solving the reconstruction problem [29]. More details of how fast Hermite expansions may assist in this problem are discussed in the following subsection.

## 6.2. Hermite expansions and the mathematics of tomographic reconstruction

Here, we provide a mathematical formulation of density reconstruction from a collection of measured projections. These projections are essentially integrals of the density along directions which may or may not be known a priori. The issue of whether or not the projection direction is known is the major difference between the medical imaging and single-particle cryo-EM problems.

A simplified version of both reconstruction problems is stated as follows: find the unknown function  $f(\mathbf{x})$ , where  $\mathbf{x} \in \mathbb{R}^3$ , from the measurements

$$g_i(x_1, x_2) = \int_{-\infty}^{\infty} f(R_i^T \mathbf{x}) dx_3 \quad (14)$$

for  $i = 1, \dots, M$  where each rotation  $R_i \in SO(3)$  is either known or unknown.

If each  $R_i$  were known in advance, this would be nothing more than a classical tomography problem. However, when all the rotations  $R_i$  are unknown, the reconstruction of  $f(\mathbf{x})$  from the collection of projections  $\{g_i(x_1, x_2) \mid i = 1, \dots, M\}$  is known to be a hard problem (particularly when the biomolecular structure under investigation does not share features with known structures).

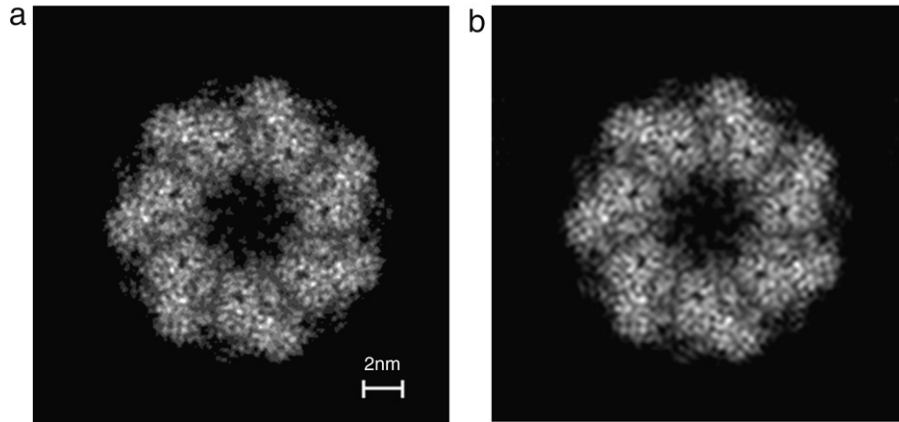
Given an initial guess  $\tilde{f}(\mathbf{x}; \mathbf{c})$ , which is a bandlimited expansion defined by the coefficients  $\mathbf{c}$ , and represents an a priori best estimate of  $f(\mathbf{x})$ , one can compute the cost function

$$C(\mathbf{c}, \tilde{R}_1, \dots, \tilde{R}_M) = \sum_{i=1}^M \int_{x_1} \int_{x_2} \left| g_i(x_1, x_2) - \int_{x_3} \tilde{f}(\tilde{R}_i^T \mathbf{x}, \mathbf{c}) dx_3 \right|^2 dx_2 dx_1. \quad (15)$$

In the case when the projection directions are not known,  $\{\tilde{R}_i\}$  is an initial guess for the set of projection orientations. This can be chosen by sampling from the uniform distribution on  $SO(3)$ . Then, the best match of projection direction to each experimentally measured projection is determined by minimizing over all possible permutations on  $M$  indices,  $\Pi_M$ . One seeks to reduce the value of  $C(\mathbf{c}, \tilde{R}_1, \dots, \tilde{R}_M)$  by updating guesses of  $M$  orientations and the parameters  $\mathbf{c}$  that define the best estimate of  $f(\mathbf{x})$ . A natural choice for the form of  $\tilde{f}(\mathbf{x}; \mathbf{c})$  is a multi-dimensional Hermite expansion, for reasons discussed previously, and used in [29].

The fast Hermite transform enters into the formulation at the very first stage of this procedure, where the experimental data  $g_{\pi(i)}(x_1, x_2)$  are expressed as a bandlimited multidimensional Hermite expansion. The extraction of Hermite function coefficients from the data and the use of the Parseval theorem for Hermite expansions (together with the property that rotations of multi-dimensional Hermite expansions maintain their original bandlimit), then means that (15) can be dealt with purely in the spectral domain.

<sup>2</sup> A more realistic version of the problem needs to take into account things like aperture-dependent phase and gain effects, that are specific to the particular electron microscope being used [9,11,33,13], but when such effects are known a priori the problem effectively can be formulated as stated above.



**Fig. 9.** (a) A simulated EM image of GroEL/GroES (PDB Code: 1AON); (b) Resulting image after applying Hermite forward and inverse transform to (a). The simulated image (left) was obtained by assigning a 3D Gaussian kernel (standard deviation = 1.25 pixel) to each  $C\alpha$ , and then projecting the resulting density onto the  $x, y$ -plane. The image size is  $257 \times 257$ , and the bandlimit for each axis is 256.

**Table 3**  
Computation time for different sizes of test images

Image size	$33 \times 33$	$65 \times 65$	$129 \times 129$	$257 \times 257$
Bandlimit	$32 \times 32$	$64 \times 64$	$128 \times 128$	$256 \times 256$
FHT time (s)	0.078	0.313	1.464	7.157
Naive time (s)	0.025	0.218	1.682	13.984
FHT error	0.2573	0.1217	0.1418	0.1034
Naive error	0.2573	0.1216	0.1417	0.1027
Support interval	$[-8 \ 8]$	$[-8 \ 8]$	$[-7.92 \ 7.92]$	$[-6.4 \ 6.4]$

The third and fourth rows show the computation time for the fast Hermite transform algorithm and the naive implementation of the 2D version of (10), respectively. The fifth and sixth rows show the error between the original image and the resulting image after the forward and inverse Hermite transforms are applied. The last row shows the support intervals.

### 6.3. Hermite expansions of 2D projected images in cryo electron microscopy

The numerical experiments in Section 5 are accomplished in the 1D setting. In this section, we include some further experiments, this time in 2D, that are more directly relevant to the protein structure problem that originally inspired this work. They also provide a nice example of the various tradeoffs that can occur among choices of bandlimit, sampling, and interval of definition (see Section 3).

Recall that the first step in solving the reconstruction problem in (14), is to apply the Hermite transform to the projected 2D image,  $g_i(x_1, x_2)$ . We present here numerical examples of the 2D Hermite transform of simulated cryo EM images.

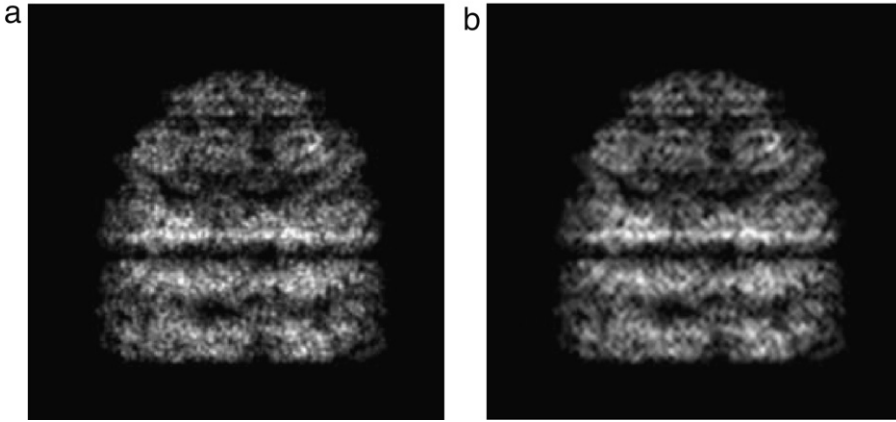
Fig. 9 shows the first example. A simulated EM image of the GroEL/GroES protein complex (PDB Code: 1AON) was obtained by assigning a 3D Gaussian kernel to each  $C\alpha$  atom in the complex, and then projecting the result to the  $x, y$ -plane. The standard deviation of the kernel corresponds to 1.25 pixel lengths in the image in this particular example, which is equivalent to 1 Å in the real scale of the GroEL/GroES protein complex. Projection from an orthogonal direction results in the  $x, z$  projection shown in Fig. 10. Figs. 9 and 10 can be considered to be the top and the front projections, respectively.

In these two examples, we were able to use a bandlimit that was almost the same as the sample size, without introducing computational instabilities. This was possible by manipulating the support intervals that we discussed in Section 3. Too large an interval leads to aliasing problems and instability, and too small an interval causes considerable loss of information in the forward Hermite transform process. The optimal value for the support interval can be determined from a series of trials. Since the Hermite transforms in Figs. 9 and 10 will be merged to one 3D density, it is important to apply the same support interval for both cases.

To test the computation time with respect to the size of images, we prepared images with the following different sizes:  $33 \times 33$ ,  $65 \times 65$  and  $129 \times 129$  and  $257 \times 257$ . The  $257 \times 257$  image is shown in Fig. 9 and the other images were obtained by reducing the largest image. The numerical results are summarized in Table 3.

The term “Naive” in Table 3 means the naive implementation of the 2D version of (10). Explicitly, the 2D Hermite transform can be written from (10) as

$$\hat{f}(m, n) = \sum_{k=0}^{M_1-1} \sum_{l=0}^{M_1-1} f(x_k, x_l) c_m(x_k) c_n(x_l).$$



**Fig. 10.** (a) A simulated EM image of GroEL/GroES due to projections in the  $x$ - $z$  plane (PDB Code: 1AON); (b) Resulting image after applying Hermite forward and inverse transform to (a). The image size is  $257 \times 257$ , and the bandlimit for each axis is 256.

Since the variables,  $x_k$  and  $x_l$  are separated in the function  $c(x)$ , this can be implemented with an  $\mathcal{O}(M_1^3)$  algorithm. The actual computation time of the fast algorithm and this naive algorithm are reported in the third and fourth rows in Table 3, respectively. These trials were done on a different platform than the other results, and so the absolute times are not necessarily comparable, but their ratios are.

The fast Hermite transform shows faster performance in the relatively large images. The fifth and sixth rows show the normalized least squares error (NLSE) between the original image and the resulting image after the forward and inverse Hermite transform. This error can be called “filtering error”, since the original image is not bandlimited in the Hermite sense. The NLSE of the two images,  $R(m, n)$  and  $I(m, n)$  is defined as

$$\text{NLSE} = \sqrt{\frac{\sum_{m=1}^N \sum_{n=1}^N [R(m, n) - I(m, n)]^2}{\sum_{m=1}^N \sum_{n=1}^N [R(m, n)]^2}}.$$

Notice that the filtering errors are similar in both FHT and naive algorithm. This means that our fast algorithm for the Hermite transform pair achieved the speed benefit, without sacrificing numerical accuracy. The last row reports the size of the optimal support interval that we found by trial and error.

Hermite functions are good to represent not only the projected data, but also the 3D density. The 3D Hermite expansion is given as

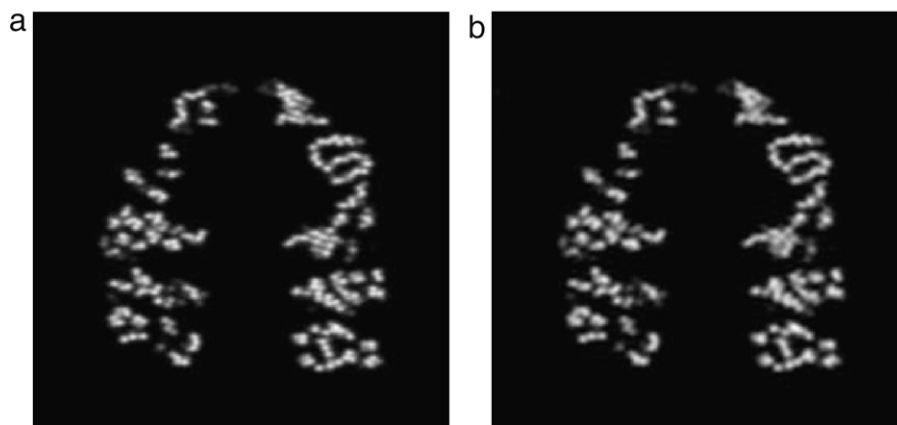
$$f_{3D}(x, y, z) = \sum_{i=0}^M \sum_{j=0}^M \sum_{k=0}^M a_{i,j,k} h_i(x) h_j(y) h_k(z)$$

where  $a_{i,j,k}$  is the Hermite coefficients and  $h(x)$  is the Hermite function. For a fixed value of  $z$ , the 2D slice of the 3D function is given as

$$f_{2D}(x, y) = \sum_{i=0}^M \sum_{j=0}^M b_{i,j} h_i(x) h_j(y),$$

where  $b_{i,j} = \sum_{k=0}^M a_{i,j,k} h_k(z)$ . (In contrast, the projection of  $f_{3D}(x, y, z)$  onto the plane  $x$ - $y$  plane would give a 2D Hermite expansion with coefficients of the form  $b_{i,j} = a_{i,j,0}$ ).

This implies that the application of the Hermite transform to 3D functions can be tested with 2D slices of the original 3D function. An example is shown in Fig. 11. Recall that Figs. 9 and 10 are projections onto the  $x$ - $y$  plane and the  $x$ - $z$  plane, respectively. Fig. 11 shows a “slice” in the  $x$ - $z$  plane passing through the geometric center of the GroEL/GroES complex. The figure also shows the Hermite-filtered version of this image. This image slice was obtained by assigning a 3D Gaussian kernel to each  $C_\alpha$  and sampling on the slice plane. For visualization purposes, we used the higher standard deviation (1.625 Å) for the Gaussian kernel than in the previous examples, because the  $C_\alpha$  atoms are more sparsely spaced in slices than in projections.



**Fig. 11.** Hermite transform with sliced image of Groel/Groes. (a) A simulated slice of electron density (b) Image resulting from applying the Hermite forward and inverse transform to (a). The image size is  $257 \times 257$ , and the bandlimit for each axis is 256. The filtering error is 9%.

## References

- [1] <http://www.cs.dartmouth.edu/~rockmore/FHT.htm>. This program is free software: You can redistribute it and/or modify under the terms of the Gnu General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version..
- [2] G. Arfken, *Mathematical Methods for Physicists*, 3rd ed., Academic Press, Orlando, FL, 1985.
- [3] A. Averbuch, R.R. Coifman, D.L. Donoho, M. Elad, M. Israeli, Fast and accurate polar Fourier transform, *Appl. Computat. Harmonic Anal.* 21 (2006) 145–167.
- [4] T.S. Baker, R.H. Cheng, A model-based approach for determining orientations of biological macromolecules imaged by cryoelectron microscopy, *J. Struct. Biol.* 116 (1996) 120–130.
- [5] H. Bateman, *Higher Transcendental Functions*, vol. 2, McGraw-Hill, New York, 1953 (California Institute of Technology: Bateman Manuscript Project).
- [6] A. Bostan, G. Lecerf, E. Schost, Tellegen's principle into practice, in: *Proceedings ISSAC'03*, ACM Press, 2003, pp. 37–44.
- [7] R. Bulirsch, J. Stoer, *Introduction to Numerical Analysis*, Springer-Verlag, Berlin, 2002.
- [8] P. Bürgisser, M. Clausen, M.A. Shokrollahi, *Algebraic Complexity Theory*, Springer-Verlag, Berlin, 1996.
- [9] P.C. Doerschuk, J.E. Johnson, Ab initio reconstruction and experimental design for cryo electron microscopy, *IEEE Trans. Information Theory* 46 (5) (2000) 1714–1729.
- [10] J.R. Driscoll, D. Healy, D. Rockmore, Fast discrete polynomial transforms with applications to data analysis for distance transitive graphs, *SIAM J. Comput.* 26 (4) (1997) 1066–1099.
- [11] H.P. Erickson, The Fourier transform of an electron micrograph – first order and second order theory of image formation, in: R. Barer, V.E. Cosslett (Eds.), *Advances in Optical and Electron Microscopy*, vol. 5, Academic Press, 1973, pp. 163–199.
- [12] K. Fourmont, Non-equispaced fast Fourier transforms with applications to tomography, *J. Fourier Anal. Appl.* 9 (5) (2003) 431–450.
- [13] J. Frank, *Three-Dimensional Electron Microscopy of Macromolecular Assemblies*, Academic Press, San Diego, CA, 1996.
- [14] W. Gautschi, *Orthogonal polynomials – constructive theory and applications*, *J. Comput. Appl. Math.* 12–13 (1985) 61–76.
- [15] T.A. Jones, J.-Y. Zou, S.W. Cowan, M. Kjeldgaard, Improved methods for building protein models in electron density maps and the location of errors in these models, *Acta Crystallogr. A* 47 (1991) 110–119.
- [16] A.C. Kak, M. Slaney, *Principles of Computerized Tomographic Imaging*, Society of Industrial and Applied Mathematics, Philadelphia, 2001.
- [17] D. Kortchagine, A. Krylov, Image database retrieval by fast Hermite projection method, in: *Proceedings of Graphicon 2005*, Moscow, 2005.
- [18] I. Krasikov, New bounds on the Hermite polynomials, <http://front.math.ucdavis.edu/math.CA/0401310>, 2005.
- [19] A. Krylov, D. Kortchagine, Hermite foveation, in: *Proceedings of Graphicon 2004*, Moscow, 2004, pp. 166–169.
- [20] G. Leibon, D. Rockmore, G. Chirikjian, A fast Hermite transform with applications to protein structure determination, in: *Proceedings of 2007 International Workshop on Symbolic-Numeric Computation*, ACM, 2007, pp. 117–124.
- [21] J.-B. Martens, The Hermite transform – applications, *Proc. IEEE Trans. Acoustics Speech Signal Process.* 38 (1990) 1607–1618.
- [22] J.-B. Martens, The Hermite transform – theory, *IEEE Trans. Acoustics Speech Signal Process.* 38 (1990) 1595–1606.
- [23] D. Maslen, D. Rockmore, Separation of variables and the computation of Fourier transforms on finite groups, I, *J. Amer. Math. Soc.* 10 (1) (1997) 169–214.
- [24] R. Massey, A. Refregier, Polar shapelets, *Mon. Not. R. Astro. Soc.* 36 (2005) 197–210.
- [25] M. Najafi, A. Krylov, D. Kortchagine, Image deblocking with 2-D Hermite transform, in: *Proceedings of GraphiCon 2003*, Moscow, 2003, pp. 180–183.
- [26] T.J. Oldfield, High-resolution crystallographic map interpretation, *Acta Crystallogr. Section D-Biol. Crystallogr.* 58 (2002) 963–967 (Part 6 Special Issue 2).
- [27] W. Park, *Inverse problems in structural biology and flexible needle steering*, Ph.D. Dissertation, Dept. of Mechanical Engineering, JHU, January 2008.
- [28] W. Park, G.S. Chirikjian, Interconversion between truncated cartesian and polar expansions of images, *IEEE Trans. Image Processing* 16 (8) (2007) 1946–1955.
- [29] W. Park, G.S. Chirikjian, Tomographic reconstruction of band-limited hermite expansions, in: *SPIE Medical Imaging*, San Diego, CA, 2008, in: *Proc. of SPIE*, vol. 6913, 2008.
- [30] W. Park, G. Leibon, D.N. Rockmore, G.S. Chirikjian, Lossless image rotation using band-limited Hermite expansions (in review).
- [31] D. Potts, G. Steidl, New Fourier reconstruction algorithms for computerized tomography, *Proc. SPIE* 4119 (1) (2000) 13–23.
- [32] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes in C*, 2nd ed., Cambridge University Press, 1995.
- [33] O. Scherzer, The theoretical resolution limit of the electron microscope, *J. Appl. Phys.* 20 (1949) 20–29.
- [34] M. van Heel, IMAGIC and its results, *Ultramicroscopy* 4 (1979) 117.
- [35] M. van Heel, G. Harauz, E.V. Orlova, A new generation of the IMAGIC image processing system, *J. Struct. Biol.* 116 (1996) 17–24.
- [36] N.J. Vilenkin, A.U. Klimyk, *Representation of Lie Groups and Special Functions*, Kluwer Academic Publ., Dordrecht, Holland, 1991.