1 Introduction

Many problems in applied mathematics involve the *Gauss transform*

$$G_\delta f(x) = \int_\Gamma e^{-|x-y|^2/\delta} f(y)\, dy \qquad (\delta > 0) \tag{1}$$

of a function f defined on $\Gamma \subset \mathbf{R}^d$. The simplest example is the heat equation. The solution of the pure initial value problem

$$\begin{aligned} u_t(x,t) &= \Delta u(x,t) \qquad \text{for } t > 0 \\ u(x,0) &= f(x) \qquad \text{for } x \in \mathbf{R}^d \end{aligned}$$

is given by

$$u(x,t) = (4\pi t)^{-d/2}\, G_{4t} f(x)\,,$$

with Γ equal to the whole space. A similar transform, with Γ a lower-dimensional subset of $\mathbf{R}^d$, occurs when one solves any initial/boundary value problem for the heat equation by means of potential theory [1, 6, 9, 13, 14]. Other examples occur in vortex methods [3], tomography [11], and nonparametric statistics [7, 15]. Finally, a common analytical tool is *mollification*; one approximates an arbitrary function f by the family of smooth rapidly decreasing functions

$$f_\delta(x) = (\pi\delta)^{-d/2} G_\delta f(x)$$

which converge to f as $\delta \to 0$.

For numerical purposes, one must discretize $G_\delta f$. Given the values of f at a set of points $s_j \in \mathbf{R}^d$, one can approximate the integral (1) by means of a quadrature formula. A reasonable approximation to $G_\delta f$ might then take the form of a *discrete Gauss transform*

$$G(x) = \sum_{j=1}^{N} q_j\, e^{-|x-s_j|^2/\delta}\,, \tag{2}$$

where the coefficients q_j depend on the values $f(s_j)$ and the weights of the chosen quadrature formula. This paper will focus on the problem of evaluating the sum of Gaussians (2) as efficiently as possible. It will often be

convenient to speak of (2) as the Gaussian "field" due to sources of strengths q_j at the points s_j, evaluated at the "target" x.

Suppose now that we want to evaluate (2) directly from the definition at M targets $x = t_i$. In other words, we want to apply the rectangular matrix with entries

$$G_{ij} = e^{-|t_i - s_j|^2/\delta}$$

to the vector $q = (q_1, \ldots, q_N)^T$. This requires $O(NM)$ work, which grows rapidly as M and N increase, and makes large scale calculations prohibitively expensive.

In this paper, we present an algorithm for evaluating (2) at M points in $O(N + M)$ work. The constant in $O(N + M)$ depends only on the dimension d and the desired precision. The amount of memory required is also proportional to $N + M$, so that the algorithm is asymptotically optimal in terms of both work and storage. Furthermore, the sources and the targets can be placed anywhere; they need not be restricted to a regular grid. Even if the function f were given at N equispaced points and $G_\delta f$ evaluated at N equispaced points, fast convolution by means of the Fast Fourier transform (FFT) would require $O(N \log N)$ operations, whereas our algorithm requires only $O(N)$.

2 Hermite expansions

This section describes the properties of the Gaussian kernel and Hermite expansions which we will need. Good references for this material are [4, 5, 8] and particularly Hille's paper [10].

The Hermite polynomials $H_n(t)$ may be defined by the Rodrigues formula

$$H_n(t) = (-1)^n e^{t^2} D^n e^{-t^2} \qquad\qquad t \in \mathbf{R}$$

where $D = d/dt$. We will make use of this definition as well as the generating function for Hermite polynomials

$$e^{2ts - s^2} = \sum_{n=0}^{\infty} \frac{s^n}{n!} H_n(t) \ .$$

Multiplication of each side of the preceding expression by e^{-t^2} yields

$$e^{-(t-s)^2} = \sum_{n=0}^{\infty} \frac{s^n}{n!} h_n(t),$$

where the Hermite functions $h_n(t)$ are defined by

$$h_n(t) = e^{-t^2} H_n(t). \tag{3}$$

(Note that these are not the usual orthonormal Hermite functions; the definition here is the right one for this situation.) In practice, we will use a shifted and scaled version of this formula: for $s_0 \in \mathbf{R}$ and $\delta > 0$, we have

$$\begin{aligned} e^{-(t-s)^2/\delta} &= e^{-(t-s_0-(s-s_0))^2/\delta} \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \left(\frac{s-s_0}{\sqrt{\delta}} \right)^n h_n \left(\frac{t-s_0}{\sqrt{\delta}} \right) \\ &= e^{-(t-s_0)^2/\delta} \sum_{n=0}^{\infty} \frac{1}{n!} \left(\frac{s-s_0}{\sqrt{\delta}} \right)^n H_n \left(\frac{t-s_0}{\sqrt{\delta}} \right). \end{aligned}$$

This formula tells us how to evaluate the Gaussian field $e^{-(t-s)^2/\delta}$ at the target t due to the source at s, as an Hermite expansion centered at s_0. Thus we are shifting a Gaussian at s to a sum of Hermite polynomials times a Gaussian, all centered at s_0.

We can also interchange t and s to write

$$e^{-(t-s)^2/\delta} = \sum_{n=0}^{\infty} \frac{1}{n!} h_n \left(\frac{s-t_0}{\sqrt{\delta}} \right) \left(\frac{t-t_0}{\sqrt{\delta}} \right)^n. \tag{4}$$

Looked at this way, the expansion expresses a Gaussian with target t as a Taylor series about a nearby target t_0; the coefficients of the Taylor series are the Hermite functions evaluated at t_0. Thus the same expansion serves as both a near-field (Taylor) and a far-field (Hermite) expansion. The final one-dimensional results we will need are the recurrence relation

$$h_{n+1}(t) = 2t\, h_n(t) - 2n\, h_{n-1}(t) \qquad t \in \mathbf{R},$$

for Hermite functions and Cramer's inequality for Hermite polynomials:

$$|H_n(t)| \leq K 2^{n/2} \sqrt{n!} e^{t^2/2}$$

where K is a numerical constant less than 1.09 in value. Cramer's inequality immediately implies the following useful bound for Hermite functions:

$$\frac{1}{n!}|h_n(t)| \leq K 2^{n/2} \frac{1}{\sqrt{n!}} e^{-t^2/2}.$$

We will also need the straightforward extensions of these facts to higher dimensions. Thus, let t and s lie in d-dimensional Euclidean space $\mathbf{R}^d$, and consider the Gaussian

$$e^{-|t-s|^2} = e^{-(t_1-s_1)^2-\ldots-(t_d-s_d)^2}.$$

We will find it convenient to adopt multiindex notation. A multiindex $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_d)$ is a d-tuple of nonnegative integers, playing the role of a multidimensional index. For any multiindex α and any $t \in \mathbf{R}^d$, we define

$$|\alpha| = \alpha_1 + \alpha_2 + \ldots + \alpha_d$$

$$\alpha! = \alpha_1! \alpha_2! \ldots \alpha_d!$$

$$t^\alpha = t_1^{\alpha_1} t_2^{\alpha_2} \ldots t_d^{\alpha_d}$$

$$D^\alpha = \partial_1^{\alpha_1} \partial_2^{\alpha_2} \ldots \partial_d^{\alpha_d}$$

where ∂_i is differentiation with respect to the ith coordinate in $\mathbf{R}^d$. If p is an integer, we say $\alpha \geq p$ if $\alpha_i \geq p$ for $1 \leq i \leq d$.

The multidimensional Hermite polynomials and Hermite functions are then defined by

$$H_\alpha(t) = H_{\alpha_1}(t_1) \ldots H_{\alpha_d}(t_d)$$

$$h_\alpha(t) = e^{-|t|^2} H_\alpha(t) = h_{\alpha_1}(t_1) \ldots h_{\alpha_d}(t_d) \tag{5}$$

where $|t|^2 = t_1^2 + \ldots + t_d^2$.

The Hermite expansion of a Gaussian in $\mathbf{R}^d$ is then simply

$$e^{-|t-s|^2} = \sum_{\alpha \geq 0} \frac{(t-s_0)^\alpha}{\alpha!} h_\alpha(s-s_0) \,. \tag{6}$$

Cramer's inequality generalizes to

$$\frac{1}{\alpha!}|h_\alpha(t)| \leq K\, e^{-|t|^2/2}\, 2^{|\alpha|/2}\, \frac{1}{\sqrt{\alpha!}}, \tag{7}$$

where K is less than $(1.09)^d$.

Finally, our algorithm will require the Taylor expansion of the Hermite function $h_\alpha(t)$ about an arbitrary point $t_0 \in \mathbf{R}^d$. Since h_α is defined by

$$\begin{aligned} h_\alpha(t) &= e^{-t^2} H_\alpha(t) \\ &= (-1)^{|\alpha|}\, D^\alpha\, e^{-t^2} \ , \end{aligned} \tag{8}$$

applying D^β gives immediately

$$D^\beta h_\alpha(t) = (-1)^{|\beta|} h_{\alpha+\beta}(t) \,. \tag{9}$$

Thus the Taylor series of h_α is

$$h_\alpha(t) = \sum_{\beta \geq 0} \frac{(t-t_0)^\beta}{\beta!}\, (-1)^{|\beta|} h_{\alpha+\beta}(t_0) \ . \tag{10}$$

We now present the three lemmas on which our algorithm relies. The first describes how to transform the field due to all sources in a box into a single rapidly converging Hermite expansion about the center of the box.

Lemma 2.1 *Let N_B sources s_j lie in a box B with center s_B and side length $r\sqrt{2\delta}$, with $r < 1$. Then the Gaussian field due to the sources in B,*

$$G(t) = \sum_{j=1}^{N_B} q_j\, e^{-|t-s_j|^2/\delta}, \tag{11}$$

is equal to a single Hermite expansion about s_B:

$$G(t) = \sum_{\alpha \geq 0} A_\alpha\, h_\alpha\left(\frac{t-s_B}{\sqrt{\delta}}\right) \,.$$

The coefficients A_α are given by

$$A_\alpha = \frac{1}{\alpha!} \sum_{j=1}^{N_B} q_j \left(\frac{s_j - s_B}{\sqrt{\delta}} \right)^\alpha . \tag{12}$$

The error $E_H(p)$ due to truncating the series after p^d terms satisfies the bound:

$$|E_H(p)| = |\sum_{\alpha \geq p} A_\alpha h_\alpha \left(\frac{t - s_B}{\sqrt{\delta}} \right)| \leq K Q_B \left(\frac{1}{p!} \right)^{d/2} \left(\frac{r^{p+1}}{1-r} \right)^d$$

where $Q_B = \sum |q_j|$ and $K = (1.09)^d$.

Proof: Use (6) to expand each Gaussian in the sum (11) into a Hermite series about s_B and interchange the sums over α and j. The truncation error bound follows from Cramer's inequality (7) and the formula for the tail of a geometric series. □

The second lemma shows how to convert an Hermite expansion about s_B into a Taylor expansion about t_C. The Taylor series converges rapidly in a box of side $r\sqrt{2\delta}$ about t_C, with $r < 1$.

Lemma 2.2 *The Hermite expansion*

$$G(t) = \sum_{\alpha \geq 0} A_\alpha \, h_\alpha \left(\frac{t - s_B}{\sqrt{\delta}} \right) \tag{13}$$

has the following Taylor expansion, about an arbitrary point t_C:

$$G(t) = \sum_{\beta \geq 0} B_\beta \left(\frac{t - t_C}{\sqrt{\delta}} \right)^\beta . \tag{14}$$

The coefficients B_β are given by

$$B_\beta = \frac{(-1)^{|\beta|}}{\beta!} \sum_{\alpha \geq 0} A_\alpha \, h_{\alpha+\beta} \left(\frac{t_C - s_B}{\sqrt{\delta}} \right) . \tag{15}$$

If the A_α are given by (12) then the error $E_T(p)$ in truncating the Taylor series after p^d terms is bounded, in the box C with center t_C and side length $r\sqrt{2\delta}$, by

$$|E_T(p)| = |\sum_{\beta \geq p} B_\beta \left(\frac{t - t_C}{\sqrt{\delta}} \right)^\beta | \leq K Q_B \left(\frac{1}{p!} \right)^{d/2} \left(\frac{r^{p+1}}{1-r} \right)^d ,$$

if $r < 1$.

Proof: Each Hermite function in (13) can be expanded into a Taylor series by means of equation (10). The expansion (14) is then obtained by interchanging the order of summation. The truncation error bound is only a little more difficult: By the formula (12) for A_α, we have

$$\begin{aligned} B_\beta &= \frac{(-1)^{|\beta|}}{\beta!} \sum_{\alpha \geq 0} A_\alpha h_{\alpha+\beta}\left(\frac{s_B - t_C}{\sqrt{\delta}}\right) \\ &= \frac{(-1)^{|\beta|}}{\beta!} \sum_{\alpha \geq 0} \frac{1}{\alpha!} \sum_{j=1}^{N_B} q_j \left(\frac{s_j - s_B}{\sqrt{\delta}}\right)^\alpha h_{\alpha+\beta}\left(\frac{s_B - t_C}{\sqrt{\delta}}\right) \\ &= \frac{(-1)^{|\beta|}}{\beta!} \sum_{j=1}^{N_B} q_j \sum_{\alpha \geq 0} \frac{1}{\alpha!} \left(\frac{s_j - s_B}{\sqrt{\delta}}\right)^\alpha h_{\alpha+\beta}\left(\frac{s_B - t_C}{\sqrt{\delta}}\right) . \end{aligned}$$

But by (10), the inner sum is the Taylor expansion of $h_\beta((s_j - t_C)/\sqrt{\delta})$. Thus

$$B_\beta = \frac{(-1)^{|\beta|}}{\beta!} \sum_{j=1}^{N_B} q_j \, h_\beta\left(\frac{s_j - t_C}{\sqrt{\delta}}\right) , \tag{16}$$

and Cramer's inequality implies

$$|B_\beta| \leq \frac{1}{\beta!} K \, Q_B 2^{|\beta|/2} \sqrt{\beta!} \leq K \, Q_B \, \frac{2^{|\beta|/2}}{\sqrt{\beta!}} . \tag{17}$$

The truncation error bound follows, as in Lemma 2.1, from summing the tail of a geometric series. □

For our algorithm, we will need a variant of Lemma 2.2 in which the Hermite series is *truncated* before converting it to a Taylor series. This essentially means that in addition to truncating the Taylor series itself, we are also truncating the infinite sum expression (15) for the coefficients. Fortunately, however, the error due this approximation of the coefficients turns out to be much smaller than the truncation error of the Taylor series.

Lemma 2.3 *A truncated Hermite expansion*

$$G(t) = \sum_{\alpha \leq p} A_\alpha h_\alpha\left(\frac{t - s_B}{\sqrt{\delta}}\right)$$

has the following Taylor expansion about an arbitrary point t_C:

$$G(t) = \sum_{\beta \geq 0} C_\beta \left(\frac{t - t_C}{\sqrt{\delta}} \right)^\beta .$$

The coefficients C_β are given by

$$C_\beta = \frac{(-1)^{|\beta|}}{\beta!} \sum_{\alpha \leq p} A_\alpha \, h_{\alpha+\beta} \left(\frac{s_B - t_C}{\sqrt{\delta}} \right) . \tag{18}$$

If the A_α are given by (12) then the error $E_T(p)$ in truncating the Taylor series after p^d terms is bounded, in the box C with center t_C and side length $r\sqrt{2\delta}$, by

$$|E_T(p)| = | \sum_{\beta \geq p} C_\beta \left(\frac{t - t_C}{\sqrt{\delta}} \right)^\beta | \leq K' \, Q_B \left(\frac{1}{p!} \right)^{d/2} \left(\frac{r^{p+1}}{1-r} \right)^d ,$$

where $K' \leq K \left(1 + (p!)^{-d/2}\right) \leq 2K$ if $r \leq 1/2$.

Proof: The proof is an application of the triangle inequality. Write C_β as the coefficient B_β from Lemma 2.2 plus the tail of a series

$$\begin{aligned}
C_\beta &= \frac{(-1)^{|\beta|}}{\beta!} \sum_{j=1}^{N_B} q_j \sum_{\alpha \leq p} \frac{1}{\alpha!} \left(\frac{s_j - s_B}{\sqrt{\delta}} \right)^\alpha h_{\alpha+\beta} \left(\frac{s_B - t_C}{\sqrt{\delta}} \right) \\
&= \frac{(-1)^{|\beta|}}{\beta!} \sum_{j=1}^{N_B} q_j \left(\sum_{\alpha \geq 0} - \sum_{\alpha > p} \right) \frac{1}{\alpha!} \left(\frac{s_j - s_B}{\sqrt{\delta}} \right)^\alpha h_{\alpha+\beta} \left(\frac{s_B - t_C}{\sqrt{\delta}} \right) \\
&= B_\beta - \frac{(-1)^{|\beta|}}{\beta!} \sum_{j=1}^{N_B} q_j \sum_{\alpha > p} \frac{1}{\alpha!} \left(\frac{s_j - s_B}{\sqrt{\delta}} \right)^\alpha h_{\alpha+\beta} \left(\frac{s_B - t_C}{\sqrt{\delta}} \right) \\
&= B_\beta + (C_\beta - B_\beta) .
\end{aligned}$$

Then

$$|E_T(p)| \leq | \sum_{\beta \geq p} B_\beta \left(\frac{t - t_C}{\sqrt{\delta}} \right)^\beta | + | \sum_{\beta \geq p} (C_\beta - B_\beta) \left(\frac{t - t_C}{\sqrt{\delta}} \right)^\beta | . \tag{19}$$

From Lemma 2.2, we know that the first sum is bounded by

$$K \, Q_B \left(\frac{1}{p!} \right)^{d/2} \left(\frac{r^{p+1}}{1-r} \right)^d .$$

Hence we need only bound the second sum. For this, we have

$$
\begin{aligned}
|\sum_{\beta \geq p} \quad (C_\beta - B_\beta) \quad & \left(\frac{t - t_C}{\sqrt{\delta}} \right)^\beta | \\
\leq \qquad & Q_B \sum_{\beta > p} | \left(\frac{t - t_C}{\sqrt{\delta}} \right)^\beta | \frac{1}{\beta!} \sum_{\alpha > p} \frac{1}{\alpha!} | \left(\frac{s_j - s_B}{\sqrt{\delta}} \right)^\alpha || h_{\alpha+\beta} \left(\frac{s_B - t_C}{\sqrt{\delta}} \right) | \\
\leq \qquad & K\, Q_B \sum_{\alpha > p} \sum_{\beta > p} \frac{r^{|\alpha|}}{\sqrt{\alpha!}} \sqrt{\frac{(\alpha+\beta)!}{\alpha!\beta!}} \frac{r^{|\beta|}}{\sqrt{\beta!}}
\end{aligned}
$$

But

$$\frac{(\alpha+\beta)!}{\alpha!\beta!} \leq 2^{|\alpha+\beta|},$$

so the lemma follows immediately. □

Remark 2.1. The alternate expression (16) for B_β which appears in the proof of Lemma 2.2 has a simple meaning. Rather than using Lemma 2.1 to accumulate all the Gaussians into a single Hermite expansion and then shifting it to t_C, we can use Lemma 2.3 to shift each Gaussian individually to t_C and add up the resulting Taylor coefficients. (A Gaussian is a one-term Hermite series, after all, and can therefore be shifted just like any other truncated Hermite series.) Thus, a Gaussian

$$G(t) = q e^{-|t - s_j|^2/\delta}$$

has the following Taylor expansion about t_C;

$$G(t) = \sum_{\beta \geq 0} B_\beta \left(\frac{t - t_C}{\sqrt{\delta}} \right)^\beta .$$

The coefficients B_β are given by

$$B_\beta = q \frac{(-1)^{|\beta|}}{\beta!} h_\beta \left(\frac{s_j - t_C}{\sqrt{\delta}} \right) \tag{20}$$

and the error in truncation after p^d terms is

$$|E_T(p)| = |\sum_{\beta \geq p} B_\beta \left(\frac{t - t_C}{\sqrt{\delta}} \right)^\beta | \leq K\,|q| \left(\frac{1}{p!} \right)^{d/2} \left(\frac{r^{p+1}}{1-r} \right)^d$$

for $r < 1$.

3 The fast Gauss transform

We now have the tools necessary to construct and analyze a fast algorithm for evaluating the discrete Gauss transform

$$G(t_i) = \sum_{j=1}^{N} q_j e^{-|t_i - s_j|^2/\delta} \tag{21}$$

for $1 \leq i \leq M$ in $O(M+N)$ work. By shifting the origin and rescaling δ if necessary, we can assume (as a convenient normalization) that the sources s_j and targets t_i all lie in the unit box $B_0 = [0,1]^d$.

The algorithm is based on subdividing B_0 into smaller boxes with sides of length $r\sqrt{2\delta}$ parallel to the axes, with a fixed $r \leq 1/2$. We can then assign each source s_j to the box B in which it lies and each target t_i to the box C where it lies. For the sake of clarity, we maintain a notational distinction between source boxes B and target boxes C even though they may be the same.

For each target box C, we need to evaluate the total field due to sources in all boxes B, at each target in C. Because the range of the Gaussian $e^{-|t-s|^2/\delta}$ is $O(\sqrt{\delta})$ and the boxes have side lengths $r\sqrt{2\delta}$, only a fixed number of source boxes B can contribute more than $Q\epsilon$ to the field in a given target box C, where $Q = \sum_{j=1}^{N} |q_j|$ and ϵ is a specified precision. Indeed, if we cut off the sum over all B after including the $(2n+1)^d$ nearest boxes to C, we incur an error bounded by $Qe^{-2r^2n^2}$. We can always choose n depending only on r and ϵ to make this less than $Q\epsilon$. For example, if $r = 1/2$ we get single precision accuracy relative to Q with $n = 6$ and double precision with $n = 8$.

Suppose now that we want to evaluate the field due to a box B with N_B sources at M_C targets in a box C. There are two ways in which B can transmit its influence, and two ways in which C can handle the information it receives. B can directly send out the strengths and centers of all N_B Gaussians located in B, or it can use Lemma 2.1 to collect them into a

single truncated Hermite expansion. C can then directly evaluate all fields (Gaussians or Hermite expansions) sent to it, at the M_C target locations in C, or it can use Lemma 2.3 to convert the fields sent to it into a single truncated Taylor expansion about the center t_C of C. Evaluation of this Taylor series then yields the total field at each target location.

Thus, there are four possible ways in which B can influence C.

1. N_B Gaussians $\rightarrow$ directly evaluated
2. N_B Gaussians $\rightarrow$ accumulated in Taylor series via (20)
3. Hermite series $\rightarrow$ directly evaluated
4. Hermite series $\rightarrow$ accumulated in Taylor series via (18)

A fast algorithm can be based on any one of the second through fourth alternatives, because they all decouple the number of sources from the number of targets. Methods 1 through 4 require the following work to evaluate $G(t)$ at M targets t_i.

1. The cost of evaluating N Gaussians at M points is of the order $O(NM)$.
2. Consider a fixed source box B. For each target box C within range, we must compute p^d Taylor series coefficients

$$C_\beta(B) = \frac{(-1)^{|\beta|}}{\beta!} \sum_{s_j \in B} q_j h_\beta \left(\frac{s_j - t_C}{\sqrt{\delta}} \right) \ . \tag{22}$$

Each coefficient requires $O(N_B)$ work to evaluate, resulting in a net cost of the order $O(p^d N_B)$. Since there are at most $(2n+1)^d$ boxes within range, the total work for forming all Taylor series is of the order $O((2n+1)^d p^d N)$. Now, for each target t_i, one must evaluate the p^d term Taylor series corresponding to the box C in which t_i lies. The total cost of this algorithm is, therefore,

$$O((2n+1)^d p^d N) + O(p^d M) \ .$$

3. In the third approach, we form a Hermite series for each box B and evaluate it at all targets. First, using Lemma 2.1, we write

$$\begin{aligned} G(t) &= \sum_B \sum_{s_j \in B} q_j e^{-|t-s_j|^2/\delta} \\ &= \sum_B \sum_{\alpha \geq 0} A_\alpha(B)\, h_\alpha\left(\frac{t-s_B}{\sqrt{\delta}}\right) + E_H(p) \end{aligned}$$

where $|E_H(p)| \leq Q\epsilon$ and

$$A_\alpha(B) = \frac{1}{\alpha!} \sum_{s_j \in B} q_j \left(\frac{s_j - s_B}{\sqrt{\delta}}\right)^\alpha \ . \tag{23}$$

To compute each $A_\alpha(B)$ costs $O(N_B)$ work, so forming all the Hermite expansions requires $O(p^d N)$ work. Evaluating at most $(2n+1)^d$ expansions at each target t_i costs $O((2n+1)^d p^d)$ work per target, so this approach results in a total work

$$O(p^d N) + O((2n+1)^d p^d M) \ .$$

4. Finally, suppose we accumulate all sources into truncated Hermite expansions and transform all Hermite expansions into Taylor expansions via Lemma 2.3. Thus, we approximate $G(t)$ in C by

$$\begin{aligned} G(t) &= \sum_B \sum_{s_j \in B} q_j e^{-|t-s_j|^2/\delta} \\ &= \sum_{\beta \leq p} C_\beta \left(\frac{t-t_C}{\sqrt{\delta}}\right)^\beta + E_T(p) + E_H(p) \end{aligned}$$

where $|E_H(p)| + |E_T(p)| \leq Q\epsilon$,

$$C_\beta = \frac{(-1)^{|\beta|}}{\beta!} \sum_B \sum_{\alpha \leq p} A_\alpha(B)\, h_{\alpha+\beta}\left(\frac{s_B - t_C}{\sqrt{\delta}}\right) \ , \tag{24}$$

and the coefficients $A_\alpha(B)$ are given by (23). As we saw under the third approach, it costs $O(p^d N)$ work to form all the Hermite expansions, i.e.

to compute the coefficients $A_\alpha(B)$ for $\alpha \leq p$ and all source boxes B. Because of the product form (5) of $h_{\alpha+\beta}$, the computation of the p^d coefficients C_β involves only $O(\,d\,p^{d+1})$ operations for each box B in range. Therefore, a total of $O((2n+1)^d\,d\,p^{d+1})$ work per target box C is required. Finally, evaluating the appropriate Taylor series for each target t_i requires $O(p^d M)$ work. Hence this algorithm has net CPU requirements of the order

$$O((2n+1)^d\,d\,p^{d+1}N_{box}) + O(p^d N) + O(p^d M)\ ,$$

where the number of boxes N_{box} is bounded by $min((r\sqrt{2\delta})^{-d/2}, M)$. Note that the factor $(2n+1)^d$ no longer multiplies either the $O(N)$ or $O(M)$ terms. The work is now decoupled into three parts; $O(p^d N)$ to form Hermite expansions, $O(p^d M)$ to evaluate Taylor series, and a constant term depending on the number of box-box interactions and the cost of transforming a Hermite expansion into a Taylor series.

Thus, we really have four algorithms for evaluating $G(t)$, three of which are asymptotically optimal. We can try to minimize the constants in the work estimate by varying the choice of algorithm from box to box. Clearly an optimal strategy for this choice is global, but a reasonable strategy can be constructed in which each box decides independently what action to take. For this purpose, let N_B sources in a box B be within range of M_C targets in a box C. Choose cutoff parameters N_F and M_L. Then

1. if $N_B < N_F$ then B sends out N_B Gaussians.
2. if $N_B \geq N_F$ then B sends out a Hermite expansion.
3. if $M_C < M_L$ then C evaluates all fields sent to it immediately.
4. if $M_C \geq M_L$ then C transforms all fields sent to it into Taylor series, accumulates the coefficients, and only then evaluates the Taylor series.

The work in this algorithm can be broken down as follows:

1.
$$\sum_{N_B \geq N_F} O(p^d N_B)$$
to evaluate Hermite expansions,

2.
$$+ \sum_{M_C < M_L} \sum_{N_B < N_F} O(N_B M_C)$$
to evaluate Gaussians,

3.
$$+ \sum_{M_C < M_L} \sum_{N_B \geq N_F} O(p^d M_C)$$
to evaluate Hermite expansions,

4.
$$+ \sum_{M_C \geq M_L} \sum_{N_B < N_F} O(p^d N_B)$$
to transform Gaussians into Taylor series,

5.
$$+ \sum_{M_C \geq M_L} \sum_{N_B \geq N_F} O(d\, p^{d+1})$$
to transform Hermite series into Taylor series,

6.
$$+ \sum_{M_C \geq M_L} O(p^d M_C)$$
to evaluate Taylor series.

Clearly we can achieve a rough balance of work by taking $N_F = O(p^{d-1})$ and $M_L = O(p^{d-1})$. The total work then has the form $O(p^d N) + O(d\, p^{d+1}(2n+1)^d min((r\sqrt{2\delta})^{-d/2}, M)) + O(p^d M)$. This is linear in N and M, with a constant depending only on the precision. The complexity estimate is similar to the fourth algorithm above, but the advantage here is that when there are only a few particles in a box, the overhead associated with transformation of Hermite series to Taylor series is avoided.

4 Formal Description of the Algorithm

In this section, we describe the fast Gauss transform in a more procedural form.

Algorithm

Comment [Choose the largest $r \leq 1/2$ such that $1/r\sqrt{2\delta}$ is an integer N_{side}. Subdivide the unit box into N_{side}^d boxes. Choose the number n of boxes to go out in each direction based on r and the desired precision ϵ. Each source sends to $(2n+1)^d$ boxes. Choose the number of terms p^d based on r and ϵ. Choose the cutoffs N_F and M_L.]

Step 1.

Assign sources and targets to boxes. Determine number of boxes containing more than M_L targets. For each such box, allocate storage for a Taylor series with p^d terms and initialize to zero.

Step 2.

Comment [Loop through boxes, computing interactions between sources in box and targets within range n. For each pair of source and target boxes, one of the four options summarized on p. 10 is used.]

do $i = 1, ..., N_{side}^d$
  N_B = number of sources in i^{th} box B.
  Form the *interaction list* of $(2n+1)^d$ target boxes C within range of B.
  if $N_B \leq N_F$ **then**
    do $j = 1, ..., (2n+1)^d$
      M_C = number of targets in j^{th} box C in *interaction list*.
      if $M_C \leq M_L$ **then**
        Compute source/target interactions by direct evaluation of Gaussians.
      else
        Convert each of the N_B sources into a Taylor series about the center of box C via equation (20) and add to Taylor series for box C.
      end if
    end do
  else $(N_B > N_F)$
    Form Hermite expansion about center of box B due to N_B sources via equation (20).
    do $j = 1, ..., (2n+1)^d$
      M_C = number of targets in j^{th} box C.
      if $M_C \leq M_L$ **then**

Evaluate Hermite expansion at each target location and add to accumulated potential.
else
Convert Hermite expansion into a Taylor series about the center of box C by means of equation (12) and add to Taylor series for box C.
end if
end do
end if
end do

Step 3.

Comment [Loop through boxes evaluating Taylor series for boxes with more than M_L targets.]

do $i = 1, ..., N_{side}^d$
M_C = number of targets in i^{th} box C.
if $M_C > M_L$ **then**
Evaluate Taylor series for box C at each of the M_C target positions to obtain the desired potential.
end if
end do

5 Numerical Experiments

In this section, we present the results of numerical experiments with the fast Gauss transform and demonstrate dramatic speedups over the direct calculation for realistic problems. A two-dimensional version of the algorithm was programmed in Fortran and run on a Sun-4 workstation, using up to 100,000 sources and targets and δ lying in the range .0001 to 1.0.

We examined the cost of the fast algorithm as compared to the direct evaluation of all the Gaussians, as N and M increased. Two distributions of targets and sources were tried, uniformly distributed in the unit box and equally spaced on a circle, and several values of δ were used. For the uniform distribution, strengths were uniformly distributed between -1 and 1. For the circle, the strengths were specified to be $\cos(\theta)$, where θ is the angle. We

asked for an error relative to the total charge Q of $\epsilon = 10^{-6}$, which required $p^d = 8^2$ terms in the Hermite and Taylor expansions. The results are given in Tables 1-4.

Table 1: Table of cost and errors for $\delta = 1.0$, with targets and sources distributed uniformly in the unit box. CPU times are given in seconds for the fast and direct algorithms. Times for the direct algorithm were estimated by evaluating $G(t)$ at 100 targets and extrapolating.

Case	$N = M$	Fast	Direct	Error/Q
1	100	0.42	0.59	.627E-08
2	200	0.62	2.3	.306E-08
3	400	1.1	9.7	.175E-08
4	800	1.8	38	.157E-08
5	1600	3.4	150	.126E-08
6	3200	6.5	601	.135E-08
7	6400	12.8	2407	.114E-08
8	12800	26.0	9702	.563E-09
9	25600	51.9	38790	.563E-09
10	51200	103	155550	.337E-09
11	102400	205	622780	.237E-09

With 102,400 sources and targets equispaced on a circle and $\delta = .01$, the fast algorithm is more than 3000 times faster than the direct calculation. Typically the performance of the fast algorithm improves when the source distribution is spatially non-uniform, as it is in many practical problems. There are then more particles in each of a smaller number of occupied boxes, reducing overhead costs.

Storage requirements for the fast algorithm are reasonable as long as N and M are large and δ is not too small. For extremely small δ, one should modify the algorithm to make use of just those boxes containing targets or sources. The interaction list for each source box can then be formed by means of an adaptive tree structure, as is done in the fast multipole method [2]. This would avoid having to loop through $N_{side}^d = (2r^2\delta)^{-d/2}$ (largely empty) boxes.

Table 2: Table of cost and errors for $\delta = 0.01$, with targets and sources distributed uniformly in the unit box.

Case	$N = M$	Fast	Direct	Error/Q
1	100	0.65	0.70	.115E-08
2	200	1.840	2.76	.616E-09
3	400	5.8	10.9	.478E-09
4	800	20.6	43.6	.291E-08
5	1600	115	174	.274E-08
6	3200	349	697	.443E-08
7	6400	344	2792	.249E-08
8	12800	353	11173	.177E-08
9	25600	383	44650	.144E-08
10	51200	431	179120	.120E-08
11	102400	538	716760	.501E-09

Table 3: Table of cost and errors for $\delta = 0.01$, with targets and sources spaced uniformly on a circle.

Case	$N = M$	Fast	Direct	Error/Q
1	100	3.25	0.67	.479E-09
2	200	4.93	2.70	.489E-09
3	400	10.9	10.8	.182E-06
4	800	12.7	42.8	.191E-06
5	1600	14.5	172	.203E-06
6	3200	18.5	684	.204E-06
7	6400	26.4	2749	.201E-06
8	12800	38.7	11003	.177E-06
9	25600	65.1	43955	.172E-06
10	51200	116	176300	.170E-06
11	102400	219	705650	.170E-06

Table 4: Table of cost and errors for $\delta = 0.0001$, with targets and sources spaced uniformly on a circle.

Case	$N = M$	Fast	Direct	Error/Q
1	100	4.61	0.68	.539E-10
2	200	5.31	2.72	.270E-10
3	400	6.24	10.9	.140E-10
4	800	8.71	42.9	.834E-11
5	1600	19.6	172	.298E-10
6	3200	63.9	690	.799E-08
7	6400	79.0	2785	.469E-08
8	12800	96.9	11132	.217E-09
9	25600	127	44375	.757E-10
10	51200	179	178120	.417E-10
11	102400	287	713830	.385E-10

6 Generalizations

The fast Gauss transform generalizes immediately to sums of the form

$$(-1)^{|\gamma|} D^\gamma G(t) = \sum_{j=1}^{N} q_j h_\gamma (t - s_j), \tag{25}$$

convolutions with a fixed Hermite function. One need only apply D^γ to all the formulas presented above and use the formula (9) relating derivatives of Hermite functions. An arbitrary multivariable polynomial $P(s)$ can be expressed as a sum of Hermite polynomials, so we can use our algorithm to evaluate sums of the form

$$\sum_{j=1}^{N} q_j P\left(\frac{t - s_j}{\sqrt{\delta}}\right) e^{-|t-s_j|^2/\delta}$$

in optimal time. As an extension of this remark, we can evaluate any convolution sum

$$K * f\,(t) = \sum_{j=1}^{N} f(s_j) K(t - s_j) \tag{26}$$

for which the kernel K has a rapidly converging Hermite series. We approximate K to within ϵ by a q^d-term truncated Hermite expansion and apply the

fast algorithm of this paper to carry out each convolution with an Hermite function. This would cost $O((pq)^d(N+M))$ to evaluate (26) at M points. A better approach, however, would be to modify the algorithm so as to create, for each box, a single $(p+q)^d$-term far-field expansion which includes the effect of Hermite functions of indices up to q. The modified algorithm would evaluate (26) at M points in $O((p+q)^d(N+M))$ work; the constant $p+q$ depends only on the precision required and the smoothness of K.

Examples of convolution kernels K with rapidly converging Hermite series include any smooth function which decays at infinity faster than any power; in particular, any smooth function with compact support.

One application of this generalization is to the problem of evaluating the *continuous* Gauss transform (1), rather than the discrete sum of Gaussians. Evaluation of the continuous Gauss transform with an order of accuracy independent of δ, as would be required to evaluate the mollification of a nonsmooth function, seems to require the use of product integration. In other words, one replaces the density f with a piecewise polynomial and evaluates the resulting integrals exactly. This gives a weighted sum of values f_j which cannot be evaluated by the Gauss transform, because the integral of a Gaussian over an element is no longer a Gaussian. However, the result can be expanded in a rapidly converging Hermite series of the form (26), and this sum can be evaluated by the generalized Gauss transform just described.

However, if one really wants to evaluate (1) accurately, product integration followed by Hermite expansion is unnecessarily troublesome. A more straightforward approach is to use the expansion on which our algorithm is based to create a "semi-continuous Gauss transform." Rather than discretizing the integral and forming the discrete moments due to the sources in each box, one simply forms the continuous moments due to the sources in each box. This gives a far more accurate Hermite expansion which can then be manipulated just as in the standard Gauss transform algorithm.

Consider, for example, the problem of evaluating

$$G\mu(t) = \int_\Gamma e^{-|t-s|^2}\mu(s)ds$$

where Γ is a hypersurface in $\mathbf{R}^d$. The semi-continuous Gauss transform can be described by the equation

$$G\mu(t) = \sum_B \int_{\Gamma\cap B} \sum_\alpha \frac{1}{\alpha!}\left(\frac{s-s_B}{\sqrt{\delta}}\right)^\alpha h_\alpha\left(\frac{t-s_B}{\sqrt{\delta}}\right)\mu(s)ds \tag{27}$$

$$= \sum_B \sum_\alpha h_\alpha\left(\frac{t-s_B}{\sqrt{\delta}}\right) M_\alpha. \tag{28}$$

Here the moments M_α can be very easily evaluated to high accuracy. We then have the Hermite expansion of $G\mu(t)$, and can manipulate it just as any other Hermite expansion. The utility of this algorithm is obvious; we are currently applying it to other problems of applied mathematics.

7 Conclusions

We have presented a "fast Gauss transform" algorithm for evaluating the sums

$$G(t_i) = \sum_{j=1}^N q_j e^{-|t_i-s_j|^2/\delta} \qquad i = 1,\ldots,M \tag{29}$$

for $1 \le i \le M$ in $O(M+N)$ work. Direct evaluation would require $O(NM)$ work in general, so this is a substantial improvement in computational complexity. In order to evaluate the sum of 100,000 Gaussians at 100,000 points, for example, the fast algorithm requires about four minutes of CPU time on a Sun-4, while direct evaluation would take more than a *week*. There are many fields of applied mathematics where such an algorithm will be a useful tool.

References

[1] R. Brown, *Layer Potentials and Boundary Value Problems for the Heat Equation on Lipschitz Cylinders*, Ph.D. Thesis, University of Minnesota, 1987.

[2] J. Carrier, L. Greengard, and V. Rokhlin, *A Fast Adaptive Multipole Algorithm for Particle Simulations*, Siam J. Sci. Stat. Comput., 9 (1988), pp. 669-686.

[3] G.H. Cottet, S. Mas-Gallic, and P.A. Raviart, *Vortex Methods for the Incompressible Euler and Navier-Stokes Equations*, Proceedings of the Workshop on Computational Fluid Dynamics and Reacting Gas Flows, Institute for Mathematics and its Applications, Minneapolis, Minnesota, September 1986.

[4] H. Dym and H. P. McKean, *Fourier Series and Integrals*, Academic Press, San Diego, 1972.

[5] A. Erdelyi, et. al. *Higher Transcendental Functions*, vol. II, McGraw-Hill, New York, 1953.

[6] A. Friedman, *Partial Differential Equations of Parabolic Type*, Prentice-Hall, New Jersey, 1964.

[7] S. Geman and C. Hwang, *Nonparametric Maximum Likelihood Estimation by the Method of Sieves*, Ann. Statist., 10 (1982), pp. 401-414.

[8] I. S. Gradshteyn and I. M. Ryzhik, *Tables of Integrals, Series, and Products*, Academic Press, New York, 1980.

[9] L. Greengard and J. Strain, *A Fast Algorithm for the Evaluation of Heat Potentials*, Yale University Department of Computer Science Research Report YALEU/DCS/RR-700, May 1989.

[10] E. Hille, *A Class of Reciprocal Functions*, Ann. Math. 27(1926), pp. 427-464.

[11] N. Lerner, *Wick-Wigner Functions and Tomographic Methods*, preprint,1989.

[12] A. McIntyre, *Boundary Integral Solutions of the Heat Equation*, Math. Comp., 46 (1986), pp. 71-79.

[13] P. J. Noon, *The Single Layer Heat Potential and Galerkin Boundary Element Methods for the Heat Equation*, Ph.D. Thesis, University of Maryland, 1988 .

[14] W. Pogorzelski, *Integral Equations and Their Applications*, Pergamon Press, Oxford, 1966.

[15] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, London, 1986.