Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 203 (2005) 274–286
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

The fast Gauss transform with complex parameters ☆

Fredrik Andersson [a,*], Gregory Beylkin [b]

[a] Centre for Mathematical Sciences, Lund University/LTH, Solvegatan 18, P.O. Box 118, SE-22100 Lund, Sweden
[b] Department of Applied Mathematics, University of Colorado, 526 UCB, Boulder, CO 80309-0526, USA

Received 17 March 2004; received in revised form 6 July 2004; accepted 8 July 2004
Available online 21 September 2004

Abstract

We construct a fast method, $\mathcal{O}(N \log N)$, for the computation of discrete Gauss transforms with complex parameters, capable of dealing with unequally spaced grid points. The method is based on Fourier techniques, and in particular it makes use of a modified unequally spaced fast Fourier transform algorithm, in combination with previously suggested divide and conquer strategies for ordinary fast Gauss transform methods.

Keywords: Gauss transform; Unequally spaced Fourier transforms; Fast algorithms; Chirped Gaussian

1. Introduction

The fast Gauss transform was introduced by Greengard and Strain in [1]. The discrete Gauss transform computes the contribution of a set of sources, each having an interaction profile in the form of a Gaussian, on a set of target points. The direct evaluation of the discrete Gauss transform is, generically, $\mathcal{O}(N^2)$ if the number of sources and targets is N. We note that in the case where the sources and targets coincide on an equally spaced grid, the discrete Gauss transform can be evaluated fast by means of the fast Fourier transform (FFT).

In [1] a combination of a divide and conquer strategy and the Hermite expansions is used to construct a fast algorithm for evaluating the discrete Gauss transform. Similar techniques was later used by Strain in [2]

☆ The research of G.B. was partially supported by NSF/ITR Grant DMS-0219326, DOE Grant DE-FG02-03ER25583, and DARPA/ARO Grant W911NF-04-1-0281.
* Corresponding author. Tel.: +46 46 222 09 21; fax: +46 46 222 4010.
E-mail address: fa@maths.lth.se (F. Andersson).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2004.07.020

to treat the generalized case of Gauss transforms with variable scales, an extension where the widths of the interaction Gaussians are either source or target dependent.

Fast algorithms for the discrete Gauss transform can be constructed in several ways. In [3] Greengard and Sun presented a new version based on Fourier techniques instead of the Hermite expansions. The divide and conquer strategy again plays the key role.

Another way of constructing fast Gauss transforms is by means of wavelet expansions using non-standard forms, cf. [4]. The complexity problems in the straightforward evaluation of the discrete Gauss transform arise when the interaction widths of the Gaussians are medium or large. In a wavelet representation issues of scales are dealt automatically due to built-in multi-resolution properties of the basis. Hence, since Gaussians are smooth non-oscillatory functions, the non-standard representation of the discrete Gauss transform is sparse.

In this paper, we construct fast Gauss transforms with complex parameters,where the interaction profiles are not pure Gaussians, but Gaussians with additional oscillation, $e^{-(a-ib)x^2}$. Our goal is fast evaluation of sums

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

with complex parameters $\alpha = a - ib$, $a > 0$, $b \in \mathbb{R}$, given coefficients q_j, source locations $s_j \in [-\frac{1}{2}, \frac{1}{2}]$ and target points $x_i \in [-\frac{1}{2}, \frac{1}{2}]$, $i = 1, \ldots, M$.

The motivation for doing this is partly for making existing tools more general, but mostly because there is an interest in fast evaluation of integrals of the form

$$Q(x) = \int q(s) e^{-\alpha|x-s|^2} \, ds. \tag{2}$$

Such integrals appear, for example, in the treatment of equations of mathematical physics by using the separated representation developed in [5].

In engineering literature the Gaussian functions with complex parameters are often referred to as *chirped* Gaussians. They appear in several applications, e.g., optics [6], communication theory [7] and atomic and molecular physics [8], motivating further the need for evaluating convolutions of the form (2). The fast Gauss transform with complex parameters is a tool for the fast evaluation of these integrals.

The algorithm that we construct uses a combination of the divide and conquer strategy and a modified, Gaussian based, unequally spaced FFT (USFFT) algorithm. A description of Gaussian based USFFT algorithms is given in Appendix A.

2. Invoking the USFFT machinery

The apparent difficulty in the fast evaluation of sums of Gaussians with complex parameters is the rapidly increasing oscillation of terms; the frequency of the oscillation increases quadratically with the distance from the center.

The problem, however, is avoided by splitting the real and imaginary part of the phase. As the first step towards fast evaluation of (1), we rewrite it as

$$G(x) = e^{ibx^2} \sum_j q_j e^{ibs_j^2} e^{-a|x-s_j|^2} e^{-i2bxs_j} = e^{ibx^2} \sum_j p_j e^{-a|x-s_j|^2} e^{-i2bxs_j}, \tag{3}$$

where $p_j = q_j e^{ibs_j^2}$. This relieves us from the problem of dealing with the quadratically increasing oscillations within the sum.

Next we proceed by using the approach for constructing the USFFT details of which are outlined in Appendix A. We introduce (similar to (A.7) and (A.8) in Appendix A.1)

$$f(s) = \sum_j p_j \delta(s - s_j), \tag{4}$$

with the Fourier transform

$$\hat{f}(\sigma) = \sum_j p_j e^{-2\pi i \sigma s_j}. \tag{5}$$

Using the Fourier integral,

$$e^{-at^2} = \sqrt{\frac{\pi}{a}} \int_{-\infty}^{\infty} e^{-\frac{\omega^2}{4a}} e^{i\omega t} \, d\omega, \tag{6}$$

we rewrite Eq. (3) as

$$G(x) = e^{ibx^2} \sqrt{\frac{\pi}{a}} \sum_j p_j \int_{-\infty}^{\infty} e^{-\frac{\pi^2\omega^2}{a}} e^{2\pi i \omega(x - s_j)} e^{-i2bxs_j} \, d\omega. \tag{7}$$

Changing the order of summation and integration and splitting $e^{2\pi i\omega(x-s_j)}$, we have

$$G(x) = e^{ibx^2} \sqrt{\frac{\pi}{a}} \int_{-\infty}^{\infty} e^{-\frac{\pi^2\omega^2}{a} + 2\pi i\omega x} \sum_j p_j e^{-i(2bx + 2\pi\omega)s_j} \, d\omega = e^{ibx^2} \sqrt{\frac{\pi}{a}} \int_{-\infty}^{\infty} e^{-\frac{\pi^2\omega^2}{a} + 2\pi i\omega x} \hat{f}\left(\frac{bx}{\pi} + \omega\right) d\omega. \tag{8}$$

The decay of the Gaussian factor $e^{-\frac{\pi^2\omega^2}{a}}$ allows us to replace the integral over the real line in (8) by that over a finite interval. For accuracy ϵ, it is sufficient to evaluate the integral (8) over the interval (approximately) $|\omega| < \sqrt{-a \ln(\epsilon)}/\pi$. Since $|x| \leqslant \frac{1}{2}$, we need to construct an accurate approximation of $\hat{f}(\xi)$ in the interval

$$|\xi| < \frac{b}{2\pi} + \frac{\sqrt{-a \ln(\epsilon)}}{\pi}. \tag{9}$$

We make use of the approach discussed in some detail in Appendix A. Let us denote the Gaussian bell parameterized by $\lambda > 0$ as $\gamma_\lambda(x) = e^{-\lambda x^2}$, its Fourier transform as $\hat{\gamma}(\xi)$, and define (see also (A.4))

$$a_\lambda(\xi) = \sum_{l \in \mathbb{Z}} |\hat{\gamma}_\lambda(\xi + l)|^2. \tag{10}$$

Using construction in Algorithm 3, we approximate for $|\xi| \leqslant \frac{N}{2}$

$$\hat{f}(\xi) \approx \frac{1}{\sqrt{a_\lambda(\frac{\xi}{\nu N})}} \sum_n \hat{g}_n \gamma_\lambda(\nu\xi - n), \tag{11}$$

where g_n are defined in (A.26) and parameters N and ν are described in Appendix A. By choosing N such that the right-hand side of (9) is less than $N/2$, we substitute (11) into (8) and obtain the approximation

$$G(x) \approx \tilde{G}(x) = e^{ibx^2} \sqrt{\frac{\pi}{a}} \int_{-\infty}^{\infty} e^{-\frac{\pi^2\omega^2}{a} + 2\pi i\omega x} \sum_n \hat{g}_n \frac{\gamma_\lambda(\nu(\frac{bx}{\pi} + \omega) - n)}{\sqrt{a_\lambda(\frac{bx/\pi + \omega}{\nu N})}}. \tag{12}$$

Changing the order between the summation and integration in (12) yields

$$\tilde{G}(x) = e^{ibx^2} \sqrt{\frac{\pi}{a}} \sum_n \hat{g}_n \int_{-\infty}^{\infty} e^{-\frac{\pi^2\omega^2}{a} + 2\pi i\omega x} \frac{\gamma_\lambda(\nu(\frac{bx}{\pi} + \omega) - n)}{\sqrt{a_\lambda(\frac{bx/\pi + \omega}{\nu N})}} \, d\omega. \tag{13}$$

By using Remark 1 (see Appendix A), we approximate $\sqrt{a_\lambda(\xi)}$ for $|\xi| < \frac{1}{4}$ by a Gaussian. As a result, the integral in (13) is evaluated analytically; this makes our choice of Gaussians as interpolating functions a natural one. Namely, we have

$$\begin{aligned}
&\sqrt{\frac{\pi}{a}}\int_{-\infty}^{\infty}\exp\left(\frac{-\pi^2\omega^2}{a}+2\pi i\omega x-\lambda\left(v^2\left(\frac{bx}{\pi}+\omega\right)-n\right)^2+\frac{(bx+\pi\omega)^2}{(vN)^2\lambda}\right)d\omega\\
&\quad=\frac{vN\sqrt{\lambda}\pi}{\sqrt{\kappa}}\times\exp\left(\frac{-\lambda^2v^2N^2(n\pi-vbx)^2+\lambda n^2\pi^2a-x^2\pi^2(a\lambda v^2N^2-b^2)}{\kappa}\right)\\
&\qquad\times\exp\left(2i\frac{abx^2(\pi^2-\lambda^2v^4N^2)+n\pi ax\lambda^2v^3N^2}{\kappa}\right),
\end{aligned}$$

where

$$\kappa=(\pi^2\lambda v^2N^2-\pi^2a+\lambda^2v^4aN^2). \tag{14}$$

Thus, we obtain

$$\begin{aligned}
\tilde{G}(x)&=\frac{vN\sqrt{\lambda}\pi}{\sqrt{\kappa}}\exp\left[\left(\frac{-\pi^2(a\lambda v^2N^2-b^2)+2iab(\pi^2-\lambda^2v^4N^2)}{\kappa}-ib\right)x^2\right]\\
&\quad\times\sum_n\tilde{\tilde{g}}_n\exp\left(-\frac{\lambda^2v^2N^2(n\pi-vbx)^2}{\kappa}\right)\exp\left(\frac{2in\pi ax\lambda^2v^3N^2}{\kappa}\right),
\end{aligned} \tag{15}$$

where

$$\tilde{\tilde{g}}_n=\tilde{g}_n\exp\left(-\frac{\lambda a\pi^2n^2}{\kappa}\right). \tag{16}$$

The structure of expressions in (3) and (15) is quite similar. Both contain a factor that depends on the target position x, and a sum of coefficients multiplied by a Gaussian and an oscillatory factor. The width of the Gaussian in (3) is proportional to $\frac{1}{\sqrt{a}}$, and in (15), to $\frac{\sqrt{\kappa}}{\lambda\pi vN}$. The dependence of the width of the Gaussian in (15) on $\sqrt{a}$ becomes more transparent by using

$$\frac{\sqrt{\kappa}}{\lambda\pi vN}=\sqrt{\frac{1}{\lambda}+a\left(\frac{v^2}{\pi^2}-\frac{1}{\lambda^2v^2N^2}\right)}. \tag{17}$$

Given typical choices of parameters, e.g., $v = 2$, and $\lambda = 0.14$ for double precision accuracy, and a sufficiently large N, we have

$$\frac{\sqrt{\kappa}}{\lambda\pi vN}\approx\frac{v}{\pi}\sqrt{a}. \tag{18}$$

It follows from (17) and (18) that the width of the Gaussian in (15) is (effectively) directly proportional $\sqrt{a}$, whereas in (3) it is inversely proportional. Thus, for a given accuracy, the number of terms in these sums for large values of a is low in (3) and high in (15) and vice versa.

3. Scaling and subdivision

From previous considerations it is clear that if a is either large or small the evaluation of (1) is straightforward. We note that using expressions in the frequency domain for large and medium values of a requires

relatively many terms in (15). We avoid this problem by dividing the source and target points into boxes proportional to the size of the interaction region, $\frac{1}{\sqrt{a}}$. When scaling such a box to the unit size $[-\frac{1}{2},\frac{1}{2}]$, the corresponding value of $\sqrt{a}$ is reduced by the same scaling factor. Hence, only values of a up to a certain number (depending on the desired precision) are needed to be treated.

More specifically, for a given accuracy ϵ, the points x affected by a source at s_j, are the ones for which

$$e^{-a(x-s_j)^2} \leqslant \epsilon,$$

or

$$|x - s_j| \leqslant \sqrt{\frac{-\ln \epsilon}{a}}.$$

Therefore, we divide the source points into K disjoint boxes C_k,

$$\left[-\frac{1}{2},\frac{1}{2}\right] \subseteq \bigcup_{k=1}^{K} C_k$$

of length

$$2\sqrt{\frac{-\ln \epsilon}{a}},$$

and centers c_k. For each k, define the corresponding box D_k for the target points with the center c_k and of twice the length of B_k. Let

$$\tilde{s}_j = \frac{s_j - c_m}{\sqrt{\frac{-\ln\epsilon}{a}}}, \quad \tilde{x}_i = \frac{x_i - c_m}{\sqrt{\frac{-\ln\epsilon}{a}}}, \quad \tilde{a} = a\left(\sqrt{\frac{-\ln\epsilon}{a}}\right)^2 = -4\ln\epsilon. \tag{19}$$

so that $\tilde{s}_j \in [-\frac{1}{2},\frac{1}{2}]$ and $\tilde{x}_i \in [-1,1]$ for $s_j \in C_k$ and $x_i \in D_k$. The contribution from the sources in C_k to targets in D_k is then computed via (15). Therefore, the complete transform (1) may now be computed by repeating this procedure for all k.

Since the parameter $a\sim$ in (19) is independent of a, the number of needed terms in (15) (see (17)) becomes

$$m = 2\left\lceil \sqrt{-\ln\epsilon\left(\frac{1}{\lambda} + \frac{-4\ln(\epsilon)v^2}{\pi^2}\right)}\right\rceil + 1,$$

where $\lceil x \rceil$ is the smallest integer larger than x. For double precision, $\lambda = 0.14$ and an oversampling factor $v = 2$,

$$m = 95, \tag{20}$$

i.e., about three times worse compared with the number of terms needed for the ordinary USFFT.

The complexity of the algorithm is then

$$\mathcal{O}\left(K\left(\frac{N}{K}\log\left(\frac{N}{K}\right) + m\frac{2M}{K}\right)\right) = \mathcal{O}\left(N\log\left(\frac{N}{K}\right) + 2mM\right).$$

3.1. A note on variable scales

In [2] Strain generalized the fast Gauss transform for variable scales. Specifically, sums of the form

$$S(x_i) = \sum_{j=1}^{N} q_j e^{-a_j|x_i - s_j|^2} \tag{21}$$

and

$$T(x_i) = \sum_{j=1}^{N} q_j e^{-a_i|x_i - s_j|^2} \tag{22}$$

were studied. These are referred to as the Gauss transforms with source and target dependent scales, respectively. Our method for treating the Gauss transforms with complex parameters can be directly applied for computing (22), but, unfortunately, not for (21). The evaluation of target dependent sums

$$T(x_i) = \sum_{j=1}^{N} q_j e^{-\alpha_i|x_i - s_j|^2}, \tag{23}$$

where $\alpha_i = a_i - ib_i$, requires only replacing necessary parameters in (15). If the parameter α_j is source dependent, then such simple generalization is not available.

4. Numerical experiments

We present numerical experiments to demonstrate the performance of our algorithms. The code written in C was tested on a Pentium 4 PC running LINUX. We used FFT from the FFTW package, and the code was compiled using gcc with the -O3 option. The sources and targets were randomly placed (with the uniform distribution) in $[-\frac{1}{2}, \frac{1}{2}]$ and the coefficients q_j were uniformly distributed in the complex box $[-\frac{1}{2}, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]$. The evaluation times and the errors are shown in Table 1 for both the direct and the fast methods.

The calculations were performed in double precision and with the parameter $\lambda = 0.14$, and the oversampling factor $\nu = 2$. The error was computed in ℓ^∞ norm.

The results displayed in Table 1 clearly demonstrate $\mathcal{O}(N \log N)$ complexity of the new algorithm. For speed comparison, we provide timings of two versions of the direct algorithm; one with the straightforward explicit evaluation of the Gaussians, and another, with precalculated matrix, where the direct algorithm

Table 1
Timing of the fast Gauss transform for $a = 138$ and $b = 100$

N	Error ℓ^∞	Timings in seconds			
		Fast GT	FFT	Direct Alg.	Matrix–vect. mult.
64	2.4e − 12	3.9e − 04	9.4e − 06	4.9e − 03	4.2e − 05
128	7.2e − 13	7.6e − 04	2.3e − 05	1.8e − 02	1.8e − 04
256	5.8e − 13	1.5e − 03	4.8e − 05	6.8e − 02	1.0e − 03
512	8.6e − 12	2.9e − 03	1.0e − 04	2.6e − 01	3.1e − 03
1024	1.4e − 12	6.0e − 03	2.2e − 04	1.2e + 00	1.3e − 02
2048	6.5e − 12	1.5e − 02	1.0e − 03	4.1e + 00	5.0e − 02
4096	2.6e − 12	2.7e − 02	2.9e − 03	1.7e + 01	1.9e − 01
8192	6.6e − 12	6.9e − 02	1.1e − 02	6.7e + 01	3.8e + 00
16384	6.9e − 12	1.4e − 01	2.8e − 02	2.6e + 02	*
32768	5.4e − 12	2.8e − 01	6.1e − 02	1.1e + 03	*

The time is not displayed in (*) as it was strongly affected by the limited size of RAM.

amounts to the matrix–vector multiplication. We also provide timings of the FFT algorithm for comparison.

We see that the fast algorithm is faster than the explicit direct method even for the low number of sources and targets. The break-even point between the fast algorithm and the matrix–vector multiplication is at about $N = 512$. Measured in units of FFTs, the cost of the new algorithm varies from about 30 FFTs for $N = 128$ to about six FFTs for $N = 8192$.

For the calculations in Table 1 the scaling and subdivision routines described in Section 3, are not employed. However, each partition obtained by the subdivision scheme can be dealt with by calculations such as displayed in Table 1, with $a < 138$. Therefore, Table 1 displays upper bounds on the necessary computation time needed to treat each partition obtained from the subdivision scheme.

5. Conclusions

We have presented a fast algorithm for the calculation of sums of the type (1) in $\mathcal{O}(N\log(N) + M)$ time. As a generalization of previous methods for evaluating the discrete Gauss transforms, our method is designed for the Gauss transforms with complex parameters. Our method is based on a version of the USFFT with the Gaussian interpolating functions. Our method can also handle the generalized Gauss transforms of the form (22), where the parameters are target dependent.

Appendix A. Gaussian based unequally spaced FFT

A.1. Preliminary considerations

The unequally spaced FFT (USFFT) algorithms are designed for the fast evaluation of trigonometric sums

$$\sum_{l=1}^{N} f_l e^{-2\pi i x_l \xi}, \quad |\xi| \leqslant \frac{N}{2}, \tag{A.1}$$

given points x_l and coefficients f_l. One of the approaches uses interpolation techniques by convolving with a "bell" having a relatively small effective support. In [9] and [10] the Gaussians and the B-splines were used as such "bells". The B-splines perform slightly better than the Gaussian bells since, for a given precision, they have a slightly smaller effective support.

However, using the Gaussian bells is a natural choice for the purposes of this paper, since integrals containing only Gaussians are evaluated analytically. We note that the error estimates presented in [9] are rather pessimistic and develop more accurate estimates.

Let us denote the Gaussian bell parameterized by $\lambda > 0$ as

$$\gamma_\lambda(x) = e^{-\lambda x^2}, \tag{A.2}$$

with the Fourier transform

$$\hat{\gamma}_\lambda(\xi) = \int_{-\infty}^{\infty} e^{-2\pi i x \xi} \gamma_\lambda(x)\, dx = \sqrt{\frac{\pi}{\lambda}} e^{-\frac{(\pi\xi)^2}{\lambda}}. \tag{A.3}$$

Starting from this, we want to construct an interpolating function (something reminiscent of the sinc-function used for ideal interpolation). The interpolation is then achieved by convolving data with the interpolating function. Generally, in order to obtain high accuracy, it is necessary for the interpolating function

to have a rather large effective support. The key idea here is to split such an interpolating function into two parts, one with a relatively small support, applied directly as a convolution, and one complementary part, applied multiplicatively in the Fourier domain.

More specifically, let

$$a_\lambda(\xi) = \sum_{l\in\mathbb{Z}} |\,\hat{\gamma}_\lambda(\xi + l)|^2, \tag{A.4}$$

and define the interpolating function φ (in the Fourier domain) by

$$\hat{\varphi}(\xi) = \frac{\hat{\gamma}_\lambda(\xi)}{\sqrt{a_\lambda(\xi)}}. \tag{A.5}$$

As is shown below, this function fulfills the desired interpolating properties.

Lemma 1. *The interpolating function $\hat{\varphi}(\xi)$ defined in (A.5) is a real and even function. Moreover, it is strictly decreasing for $\xi > 0$, and $\hat{\varphi}(\xi) \leqslant 1$.*

Proof. From its definition it is clear that $\hat{\varphi}$ is real and even. For the monotonicity property, the straightforward calculation gives

$$\frac{\mathrm{d}}{\mathrm{d}\xi}\hat{\varphi}(\xi) = \frac{2\pi^2 \mathrm{e}^{-\frac{\pi^2\xi^2}{\lambda}} \sum_{k\in\mathbb{Z}} k \mathrm{e}^{-\frac{2\pi^2(\xi+k)^2}{\lambda}}}{\left(\sum_{k\in\mathbb{Z}} \mathrm{e}^{-\frac{2\pi^2(\xi+k)^2}{\lambda}}\right)^{\frac{3}{2}}}. \tag{A.6}$$

The sign of the right-hand side of (A.6) depends only on the sign of

$$\sum_{k\in\mathbb{Z}} k \mathrm{e}^{-\frac{2\pi^2(\xi+k)^2}{\lambda}}.$$

Since the center of the Gaussian is shifted left for $\xi > 0$, this sign is negative, and hence $\hat{\varphi}(\xi)$ is strictly decreasing for $\xi > 0$. The bound then follows immediately from the fact that $\hat{\varphi}(0) = 1$. □

Let us turn our attention back to sums of the form (A.1). In the discussion below, we assume that $-\frac{1}{2} \leqslant x_l \leqslant \frac{1}{2}$, $l = 1, \ldots, N$, and that the coefficients are zero in a narrow neighborhood of the boundary of $[-\frac{1}{2}, \frac{1}{2}]$. This assumption is made for simplicity and may be avoided by using periodically extended functions, cf. [10].

Let us write

$$f(x) = \sum_{l=1}^{N} f_l \delta(x - x_l), \tag{A.7}$$

so that the sum in (A.1) is the Fourier transform of f,

$$\hat{f}(\xi) = \sum_{l=1}^{N} f_l \mathrm{e}^{-2\pi \mathrm{i} x_l \xi}. \tag{A.8}$$

We define the projection operator $\mathscr{P}_k^v$ by the convolution

$$\mathscr{P}_k^v f = \int_{-\infty}^{\infty} f(x)\gamma_\lambda(vN_x - k)\,\mathrm{d}x. \tag{A.9}$$

The parameter v, with the typical value $v = 2$, is an oversampling factor.

Application of Parseval's formula to (A.9) gives

$$\mathscr{P}_k^v f = \int_{-\infty}^{\infty} \mathrm{e}^{2\pi i k\xi} \hat{f}(vN\xi)\hat{\gamma}_\lambda(\xi)\,\mathrm{d}\xi, \tag{A.10}$$

which, in turn, is written as

$$\mathscr{P}_k^v f = \int_{-\frac{1}{2}}^{\frac{1}{2}} \mathrm{e}^{2\pi i k\xi} \sum_{l\in\mathbb{Z}} \hat{f}(vN(\xi+l))\hat{\gamma}_\lambda(\xi+l)\,\mathrm{d}\xi. \tag{A.11}$$

We split the integral over $\mathbb{R}$ in (A.10) into a sum over intervals in (A.11), so that for a proper choice of λ, and $|\xi|$ sufficiently small, $\hat{\gamma}_\lambda(\xi)$ can be kept below desired computational precision, whenever $l \neq 0$, thereby allowing us to neglect those terms.

Specifically, since the integral of (A.11) represents the coefficients of the Fourier series, it follows that for

$$F(\xi) = \sum_{k\in\mathbb{Z}} \left(\mathscr{P}_k^v f\right)\mathrm{e}^{-2\pi i k\xi}, \tag{A.12}$$

it holds that

$$F(\xi) = \sum_{l\in\mathbb{Z}} \hat{f}(vN(\xi+l))\hat{\gamma}_\lambda(\xi+l). \tag{A.13}$$

Dividing both sides of (A.13) by $\sqrt{a_\lambda(\xi)}$ then gives

$$\frac{F(\xi)}{\sqrt{a_\lambda(\xi)}} = \sum_{l\in\mathbb{Z}} \hat{f}(vN(\xi+l))\hat{\varphi}(\xi+l),$$

since $a_\lambda(\xi)$ is periodic with the period 1. From Lemma 1 it follows that

$$\sup_{|\xi|\leqslant\alpha} \hat{\varphi}(\xi+l) = \hat{\varphi}(|\,l\,|-\alpha),$$

and that

$$\sup_{|\xi|\leqslant\alpha} 1-\hat{\varphi}(\xi) = 1-\hat{\varphi}(\alpha),$$

Hence, by introducing

$$C_{\hat{f}}(l,\alpha) = \sup_{|\xi|<\alpha} |\,\hat{f}(vN(\xi+l))\,|,$$

we estimate

$$\left|\frac{F(\xi)}{\sqrt{a_\lambda(\xi)}} - \hat{f}(vN(\xi))\right| \leqslant (1-\hat{\varphi}(\alpha))C_{\hat{f}}(0,\alpha) + \sum_{l\in\mathbb{Z}\setminus\{0\}} C_{\hat{f}}(l,\alpha)\hat{\varphi}(|\,l\,|-\alpha)$$

for $|\xi| < \alpha$. We summarize this result as

Theorem 1. *Let E_∞ be the error in approximating the Fourier transform of the compactly supported function f by $\frac{F(\xi)}{\sqrt{a_\lambda(\xi)}}$, with F as defined in (A.12) and a by (A.4), i.e.*

$$E_\infty = \frac{\sup_{|\xi|<\alpha} |\,\frac{F(\xi)}{\sqrt{a_\lambda(\xi)}} - \hat{f}(vN\xi)\,|}{\sup_{|\xi|<\alpha} |\,\hat{f}(vN\xi)\,|}.$$

Then

$$E_\infty \leqslant 1 - \hat{\varphi}(\alpha) + \frac{1}{C_{\hat{f}}(0,\alpha)} \sum_{l \in \mathbb{Z}\setminus\{0\}} C_{\hat{f}}(l,\alpha)\hat{\varphi}(|\, l \,| - \alpha).$$

Due to rapid decay of $\hat{\varphi}$ away from the origin, the dominant part in the error estimate is $1 - \hat{\varphi}(\alpha)$ and the terms corresponding to $l = \pm 1$.

Since we are interested in calculating $\hat{f}(\xi)$ for $|\, \xi \,| \leqslant \frac{N}{2}$, the parameter α above should be chosen as $\alpha = \frac{1}{2v}$. For the oversampling factor $v = 2$, and $|\, \xi \,| < \frac{1}{4}$, $\hat{\varphi}(\xi)$ is equal to one within the double precision and likewise, $\hat{\varphi}(\xi \pm 1)$ are zero within the double precision, provided $\lambda < 0.14$. For $\lambda = 0.14$ the numerical support of γ_λ is contained in $[-16, 16]$. Hence, each projection (A.9) takes on average addition of 33 terms, which should be compared to 23 if the B-splines are used, cf. [10].

Remark 1. For oversampling factors $v \geqslant 2$, the accurate evaluation of $a_\lambda(\xi)$ is only needed for $|\, \xi \,| \leqslant \frac{1}{4}$. In this interval the calculation of a_λ simplifies significantly. The fraction

$$\frac{\gamma_\lambda(\xi)}{\gamma_\lambda(\xi + 1)}$$

for $|\, \xi \,| < \frac{1}{2v}$ reaches its maxima at $|\, \xi \,| = \frac{1}{2v}$. For $\lambda = 0.14$ (double precision) and $v = 2$, its size is 5×10^{-16} and, thus, all the terms with $l \neq 0$ in (A.4) can be neglected. Therefore, for practical calculations we use

$$\sqrt{a_\lambda(\xi)} \approx \sqrt{\frac{\pi}{\lambda}} \mathrm{e}^{-\frac{\pi^2 \xi^2}{\lambda}}.$$

A.2. Algorithms

We consider three cases for the evaluation of

$$\hat{f}(\xi_j) = \sum_{l=1}^{N} f_l \mathrm{e}^{-2\pi \mathrm{i} x_l \xi_j}, \tag{A.14}$$

namely,

(1) Equally spaced ξ_j and unequally spaced x_l.
(2) Unequally space ξ_j and equally spaced x_l.
(3) Both ξ_j and x_l are unequally spaced.

A.2.1. Unequally spaced FFT

The first case is treated by the scheme outlined above, since the sum (A.12) can be computed by the FFT.

If the coefficients f_l are zero for points x_l within a (narrow) neighborhood of the boundary of $[-\frac{1}{2}, \frac{1}{2}]$, i.e., the interaction region of $\gamma_\lambda(vN_x)$, it follows that $\mathscr{P}_k^v$ is close to zero for $|\, k \,| \geqslant \frac{vN}{2}$. Hence, the infinite sum (A.12) can be replaced by a sum with terms indexed by $|\, k \,| < \frac{vN}{2}$.

Assuming $\xi_j = \frac{j}{N} - \frac{v}{2}$, $j = 0, \ldots, vN - 1$, the evaluation of $\hat{f}(\xi_j)$ for $j = 0, \ldots, N - 1$ uses

Algorithm 1.

(1) Compute

$$\mathscr{P}_k^v f = \sum_{l=1}^{N} f_l \gamma_\lambda(vNx_l - k). \tag{A.15}$$

(2) Compute the sum

$$F(\xi_j) = \sum_{k=-\frac{\nu N}{2}}^{\frac{\nu N}{2}-1} \mathscr{P}_k^{\nu} f \mathrm{e}^{-2\pi i k \xi_j} \tag{A.16}$$

by FFT.

(3) Divide $F(\xi_j)$ by $\sqrt{a_\lambda(\xi_j)}$.

A.2.2. Fast evaluation of the Fourier series at unequally spaced points

We use a duality argument to construct an algorithm for the case where the function samples are known at uniformly distributed points, and the evaluation is sought at unequally spaced points.

Let us define operator T as

$$Tf(\xi) = \frac{F(\xi/v)}{\sqrt{a_\lambda(\xi/v)}},$$

with $F(\xi)$ defined by (A.12) and $a_\lambda(\xi)$ by (A.4). Consider now the problem of evaluating sums

$$\hat{g}(\xi) = \sum_{l=-\frac{N}{2}}^{\frac{N}{2}-1} g_l \mathrm{e}^{\frac{-2\pi i \xi l}{N}},$$

at unequally spaced points $|\xi_j| \leqslant \frac{N}{2}$. Let us consider $f \in C_0^\infty$ with supp $(\hat{f}) \subset (-\frac{N}{2}, \frac{N}{2})$ and $||f||_\infty = 1$, and

$$g(x) = \sum_{l=-\frac{N}{2}}^{\frac{N}{2}-1} g_l \delta\left(x - \frac{1}{N}\right).$$

Since g is supported in $[-\frac{1}{2}, \frac{1}{2}]$, it follows from Theorem 1 and the conditions on f, that for $|x| \leqslant \frac{1}{2}$

$$|\langle g(x), \hat{f}(Nx) - Tf(x)\rangle| \leqslant \epsilon \sum_{l=-\frac{N}{2}}^{\frac{N}{2}-1} |g_l|,$$

where $\langle\cdot,\cdot\rangle$ denotes the usual inner product in $L_2(\mathbb{R})$. Now, since

$$\langle g(x), \hat{f}(Nx)\rangle = \langle \hat{g}(N\xi), f(\xi)\rangle, \tag{A.17}$$

an approximation of $\hat{g}(\xi)$ can be obtained by applying the adjoint operator T^* to $g(x)$. We write

$$\langle g, Tf\rangle = \int_{-\infty}^{\infty} \sum_{l=-\frac{N}{2}}^{\frac{N}{2}-1} \frac{g_l}{\sqrt{a_\lambda\left(\frac{\xi}{v}\right)}} \delta\left(\xi - \frac{l}{N}\right) \bar{F}\left(\frac{\xi}{v}\right) \mathrm{d}\xi, \tag{A.18}$$

and define the operator $\tilde{\mathscr{P}}^{\nu} f$ on test functions f as

$$\tilde{\mathscr{P}}^{\nu} f(x) = \sum_{k\in\mathbb{Z}} \int_{-\infty}^{\infty} f(\tilde{x}) \gamma_\lambda(\nu N\tilde{x} - x)\, \mathrm{d}\tilde{x} \delta(x - k). \tag{A.19}$$

The Fourier transform of (A.19) is then equal to $F(\xi)$ in (A.12) and, hence, the Parseval's identity applied to (A.18) gives

$$\langle g, Tf\rangle = \int_{-\infty}^{\infty} \hat{G}\left(\frac{\xi}{v}\right) \tilde{\mathscr{P}}^{v} f(\xi)\,\mathrm{d}\xi, \tag{A.20}$$

where

$$\hat{G}(\xi) = \sum_{l=-\frac{N}{2}}^{\frac{N}{2}-1} \frac{g_l}{\sqrt{a_\lambda\left(\frac{l}{vN}\right)}} \mathrm{e}^{-\frac{2\pi i l\xi}{N}}. \tag{A.21}$$

Inserting (A.19) into (A.20) finally yields

$$\langle g, Tf\rangle = \int_{-\infty}^{\infty} \sum_{k\in\mathbb{Z}} \hat{G}\left(\frac{k}{v}\right) \gamma_\lambda(vN\tilde{x} - k)\bar{f}(\tilde{x})\,\mathrm{d}\tilde{x} = \langle \hat{g}_{\mathrm{app}}(N\xi), f\rangle, \tag{A.22}$$

where

$$\hat{g}_{\mathrm{app}}(\xi) = \sum_{k\in\mathbb{Z}} \hat{G}\left(\frac{k}{v}\right) \gamma_\lambda(v\xi - k). \tag{A.23}$$

Note that values of $\hat{G}$ are needed at points $\frac{k}{v}$. To obtain this (oversampled) data, let

$$\tilde{g}_l = \begin{cases} \frac{g_l}{\sqrt{a_\lambda\left(\frac{l}{vN}\right)}}, & -\frac{N}{2} \leqslant l \leqslant \frac{N}{2} - 1; \\ 0, & \text{otherwise.} \end{cases} \tag{A.24}$$

Applying FFT to the sequence $\{\tilde{g}_l\}_{l=-\frac{vN}{2}}^{\frac{vN}{2}-1}$ gives the desired values $\{\hat{G}(\frac{k}{v})\}_k$.

We summarize these considerations for evaluation of the Fourier series at unequally spaced as

Algorithm 2.

(1) Calculate $\{\tilde{g}_l\}$ as defined by (A.24).
(2) Apply FFT to obtain $\hat{G}(\frac{k}{v})$.
(3) Calculate $\hat{g}_{\mathrm{app}}(\xi)$ by the sum in (A.23).

A.2.3. Evaluation of unequally spaced FFT at unequally spaced points

We use a combination of the two previous methods for the fast evaluation of (A.14), where both the spatial points $x_1 \in [-\frac{1}{2}, \frac{1}{2}]$ and frequencies $\xi_j \in [-\frac{N}{2}, \frac{N}{2}]$ are unequally spaced. This combination is constructed by using an intermediate equally spaced samples:

Algorithm 3.

(1) Calculate

$$\mathscr{P}_k^v f = \sum_l f_l \gamma_\lambda\left(vN\left(x_1 - \frac{k}{vN}\right)\right), \quad k = -\frac{v^2N}{2}, \ldots, \frac{v^2N}{2} - 1 \tag{A.25}$$

(by including only numerically significant terms in the sum) to obtain an equispaced representation on $[-\frac{v}{2}, \frac{v}{2}]$.

(2) Divide by the orthogonalization factor $\sqrt{a_\lambda}$,

$$g_k = \frac{\mathscr{P}_k^v f}{\sqrt{a_\lambda\left(\frac{k}{v^2N}\right)}}$$

(3) Apply FFT for the evaluation of

$$\hat{g}_n = \sum_{k=-\frac{v^2N}{2}}^{\frac{v^2N}{2}} g_k e^{-\frac{2\pi i k n}{v^2 N}}, \quad n = -\frac{v^2N}{2}, \ldots, \frac{v^2N}{2} - 1 \tag{A.26}$$

(4) Evaluate the sums (including only numerically significant terms)

$$\hat{h}(\xi_j) = \sum_n \hat{g}_n \gamma_\lambda(v\xi_j - n). \tag{A.27}$$

(5) Divide by the orthogonalization factor $\sqrt{a_\lambda}$ to obtain

$$\hat{f}_{\text{app}}(\xi_j) = \frac{\hat{h}(\xi_j)}{\sqrt{a_\lambda\left(\frac{\xi_j}{vN}\right)}} \tag{A.28}$$

as an approximation of $\hat{f}(\xi)$.

References

[1] L. Greengard, J. Strain, The fast Gauss transform, SIAM, J. Sci. Stat. Comput. 12 (1) (1991) 79–94.
[2] J. Strain, The fast Gauss transform with variable scales, SIAM J. Sci. Statist. Comput. 12 (5) (1991) 1131–1139.
[3] L. Greengard, X. Sun, A new version of the fast Gauss transform, in: Proceedings of the International Congress of Mathematicians, vol. III (Berlin, 1998), No. extra vol. III, 1998, pp. 575–584 (electronic).
[4] G. Beylkin, R. Coifman, V. Rokhlin, Fast wavelet transforms and numerical algorithms I, Comm. Pure Appl. Math. 44 (1991) 141–183, Yale University, Technical Report YALEU/DCS/RR-696, August 1989.
[5] G. Beylkin, M.J. Mohlenkamp, Numerical operator calculus in higher dimensions, Proc. Natl. Acad. Sci. USA 99 (16) (2002) 10246–10251. Available from: <www.pnas.org/cgi/content/abstract/112329799v1> .
[6] D. Andersson, Variational approach to nonlinear pulse propagation in optical fibers, Phys. Rev. A 27 (6) (1983) 3135–3145.
[7] T. Strohmer, R.W. Heath Jr., Grassmannian frames with applications to coding and communication, Appl. Comput. Harmon. Anal. 14 (3) (2003) 257–275.
[8] T.K. Rebane, P.V. Vitushinski, Complex Gaussian functions for four-particle systems, Opt. Spectrosc. 92 (1) (2002) 17–19.
[9] A. Dutt, V. Rokhlin, Fast Fourier transforms for nonequispaced data, SIAM J. Sci. Comput. 14 (6) (1993) 1368–1393.
[10] G. Beylkin, On the fast Fourier transform of functions with singularities, Appl. Comput. Harmon. Anal. 2 (4) (1995) 363–381.