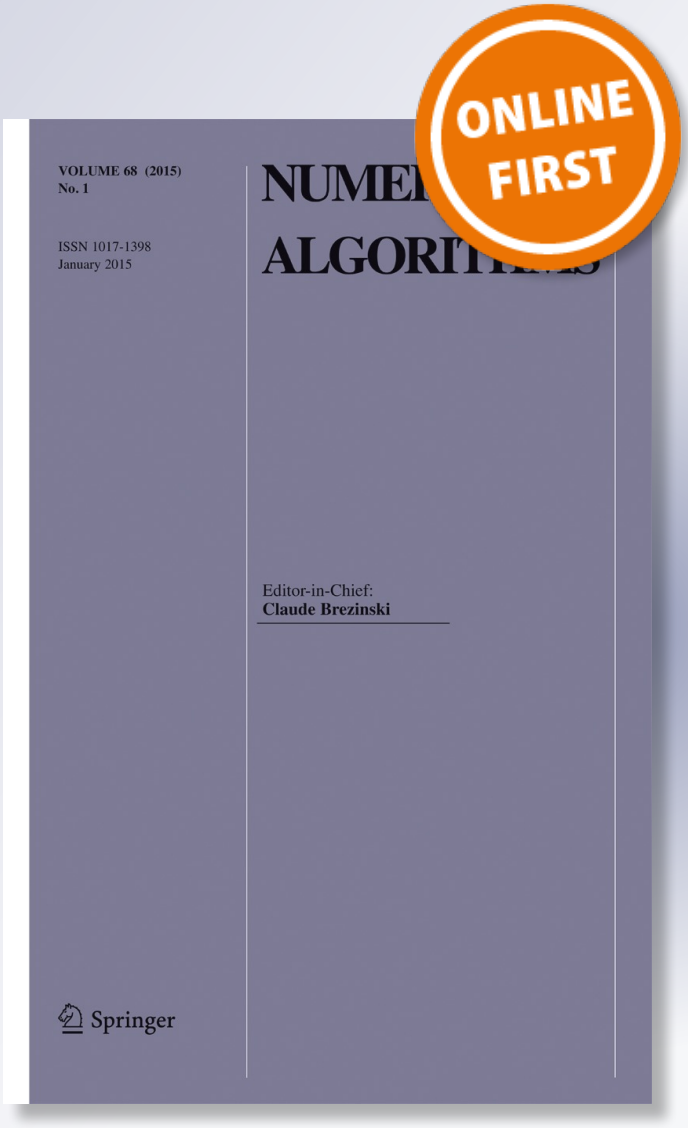
The sparse cardinal sine decomposition and its application for fast numerical convolution

François Alouges & Matthieu Aussal

Numerical Algorithms

ISSN 1017-1398

Numer Algor
DOI 10.1007/s11075-014-9953-6



Springer

Numer Algor
DOI 10.1007/s11075-014-9953-6

ORIGINAL PAPER

The sparse cardinal sine decomposition and its application for fast numerical convolution

François Alouges · Matthieu Aussal

Received: 23 July 2014 / Accepted: 23 December 2014

Abstract Fast convolution algorithms on unstructured grids have become a well established subject. Algorithms such as Fast Multipole Method (FMM), Adaptive Cross Approximation (ACA) or $\mathcal{H}$-matrices for instance are, by now, classical and reduce the complexity of the matrix-vector product from $O(N^2)$ to $O(N \log N)$ with a broad range of applications in e.g. electrostatics, magnetostatics, acoustics or electromagnetics. In this paper we describe a new algorithm of which we would like to explore the potential. Based on the Non Uniform FFT algorithm, it is at the same time simple, efficient and versatile.

Keywords Quadrature · Non uniform FFT · Fast convolution · Fast multipole method

Mathematics Subject Classifications (2010) 65T50 · 65Z05

1 Introduction

In many areas of computational physics appears the problem of solving a PDE via the convolution of a distribution with an explicit Green kernel. This is for instance the case in electrostatics, magnetostatics or gravitation where a Coulombian interaction is used. In order to fix the ideas we set the problem as computing for N given

F. Alouges (✉) · M. Aussal
CMAP - Ecole Polytechnique, Route de Saclay, 91128 Palaiseau Cedex, France
e-mail: francois.alouges@polytechnique.edu

M. Aussal
Digital Media Solutions, 45 Grande Allée du 12 Février 1934, 77186 Noisiel, France
e-mail: matthieu.aussal@gmail.com

Published online: 15 January 2015

punctual masses $(f_k)_{\{1\leq k\leq N\}}$ located at the points $(x_k)_{\{1\leq k\leq N\}}$ in $\mathbb{R}^3$, the Coulombian potential at the same points defined by

$$\forall k \in \{1, \cdots, N\},\ g_k = \sum_{l=1}^{N} \frac{f_l}{4\pi |x_k - x_l|}. \tag{1}$$

The previous problem is easily solved with the $O(N^2)$ algorithm by computing straightforwardly the potential generated by each source at each target. However for big values of N (e.g. a million or a billion), the above formula is too slow to be used. Techniques already exist to speed up this computation, the most famous of which being certainly the FMM (for Fast Multipole Method) developed by Greengard and coauthors (see the seminal book [11] for instance). We hereafter would like to present a different method that is based on a sparse Fourier representation.

2 Principle of the method

Introducing the $N \times N$ matrix A whose entries are $A_{kl} = \frac{1}{4\pi |x_k - x_l|}$, we recast the problem as storing A and computing the matrix-vector product $G = AF$, where $G = (g_1, \cdots, g_N)^t$ and $F = (f_1, \cdots, f_N)^t$. A fast algorithm is at hand if we are able to write A as a sum of a small number of rank one matrices[1]

$$A = \sum_n C_n R_n,$$

where C_n and R_n are respectively column and row vectors of size N. Indeed, the matrix vector product is computed in this case as

$$AF = \sum_n C_n R_n F = \sum_n C_n (R_n^t \cdot F), \tag{2}$$

and each term in the sum is computed in $O(N)$ operations.

Notice that such a decomposition is formally given by writing the Fourier transform of the kernel $\frac{1}{4\pi |x|}$ in $\mathbb{R}^3$

$$\mathcal{F}\left(\frac{1}{4\pi |x|}\right) = \frac{1}{|\xi|^2}.$$

Indeed, this enables us to write

$$\frac{1}{4\pi |x - y|} = \frac{1}{(2\pi)^3} \int_{\mathbb{R}^3} \frac{e^{i(x-y)\cdot\xi}}{|\xi|^2} d\xi,$$

[1] This is by the way the basis of the principle of $\mathcal{H}-$matrices, see [13].

while coming back to the problem (1), we deduce

$$\sum_{l=1}^{N} \frac{1}{4\pi\,|x_k - x_l|} f_l = \frac{1}{(2\pi)^3} \sum_{l=1}^{N} \left(\int_{\mathbb{R}^3} \frac{e^{i(x_k - x_l)\cdot\xi}}{|\xi|^2} d\xi \right) f_l$$
$$= \frac{1}{(2\pi)^3} \int_{\mathbb{R}^3} e^{ix_k\cdot\xi} \left(\sum_{l=1}^{N} \frac{e^{-ix_l\cdot\xi}}{|\xi|^2} f_l \right) d\xi. \tag{3}$$

This latter formula is exactly under the desired form (2) with

$$C_\xi = \frac{1}{(2\pi)^3}(e^{ix_1\cdot\xi}, \cdots, e^{ix_N\cdot\xi})^t \text{ and } R_\xi = \frac{(e^{-ix_1\cdot\xi}, \cdots, e^{-ix_N\cdot\xi})}{|\xi|^2},$$

except that the finite sum indexed by n is now replaced by an integral over $\xi \in \mathbb{R}^3$. We can thus produce a formula if we can derive a quadrature formula that approximates the integral. For efficiency reasons, it is also desirable to have the smallest possible number of terms in this quadrature formula.[2] In order to proceed, we pass in spherical coordinates, and further transform (3) into

$$g_k = \sum_{l=1}^{N} \frac{1}{4\pi|x_k - x_l|} f_l$$
$$= \frac{1}{(2\pi)^3} \int_0^{+\infty} \left(\int_{\mathbb{S}^2} e^{i\lambda x_k\cdot\xi} \left(\sum_{l=1}^{N} \frac{e^{-i\lambda x_l\cdot\xi}}{\lambda^2} f_l \right) \lambda^2 d\xi \right) d\lambda$$
$$= \frac{1}{(2\pi)^3} \int_0^{+\infty} \left(\int_{\mathbb{S}^2} e^{i\lambda x_k\cdot\xi} \left(\sum_{l=1}^{N} e^{-i\lambda x_l\cdot\xi} f_l \right) d\xi \right) d\lambda. \tag{4}$$

The basis of our method thus consists in finding numerical quadratures for the integrals in λ and ξ. This would eventually permit us to approximate the above formula by

$$\sum_{l=1}^{N} \frac{1}{4\pi\,|x_k - x_l|} f_l \sim \sum_{s=1}^{S} \omega_s e^{i\lambda_s x_k\cdot\xi_s} \left(\sum_{l=1}^{N} e^{-i\lambda_s x_l\cdot\xi_s} f_l \right)$$
$$= \sum_{s=1}^{S} \omega_s e^{ix_k\cdot\zeta_s} \left(\sum_{l=1}^{N} e^{-ix_l\cdot\zeta_s} f_l \right),$$

having set $\zeta_s = \lambda_s\xi_s \in \lambda_s\,\mathbb{S}^2$. Afterwards, the convolution can be done in two passes :

- Compute $\eta_s = \sum_{l=1}^{N} e^{-ix_l\cdot\zeta_s} f_l$ from the $(f_l)_{1\le l\le N}$ for all $1 \le s \le S$;
- Compute $g_k = \sum_{s=1}^{S} e^{ix_k\cdot\zeta_s}(\omega_s\eta_s)$ with previously computed $(\eta_s)_{1\le s\le S}$.

[2] Notice that similar ideas are already used in [3] and [10], though in a different context.

Both steps can be realized using a (direct and transpose) type-3 Non Uniform FFT (NUFFT) algorithm [5, 17, 18]. We will be more explicit on this point later.

Afterall, the method is not surprising and simply relies on the fact that a convolution is a product in Fourier space. The difficulty however stands in the fact that one needs, to make the method of practical use, a good quadrature which is both precise and efficient (i.e. that possesses a small number of points).

It turns out that this is feasible to a very reasonable cost, and at the end gives a quite powerful numerical method that we have called *Sparse Cardinal Sine Decomposition* (SCSD). Maybe the originality of the present approach lies in the fact that, having written the integral in spherical coordinates permits us furthermore to decouple the approximation problem (in λ and ξ), and treat them separately. The aim of the paper is to give such a method and estimate its complexity through the following Proposition.

Proposition 1 *Assuming that the original points are uniformly distributed in a ball, the SCSD algorithm descirbed above has a complexity that scales as* $O((-\log(\varepsilon))^{3/2} N \log N)$. *If instead, the points are spread uniformly on a regular surface in* $\mathbb{R}^3$, *then the complexity is* $O((-\log(\varepsilon))^{6/5} N^{6/5} \log N)$, *where* ε *measures the accuracy to within the formula* (1) *is computed.*

The paper is divided as follows. Section 3 addresses the problem of finding a good quadrature in λ, by introducing a so-called sparse cardinal sine decomposition. Section 4 explain the spherical quadrature. Actually, a classical Gauss-Legendre quadrature is used whose parameters are thoroughly studied (see [12] for instance). Eventually Section 5 explains how to take into account the close interactions, while Section 6 details how to choose the parameters of the method. Other kernels are treated in Section 7 and numerical results are provided the last Section. In particular, we compare the timings obtained to compute the convolution using either a FMM routine provided by Leslie Greengard on CMCL website [20] or the abovementionned quadrature and NUFFT approach where the (type-3) NUFFT routine is also taken from the same website [19]. (Another implementation, which in particular uses the FFTW package, is also available in the NFFT project of D. Potts [22] but the type 3 NFFT does not yet have a MATLAB interface. Since all our numerical experiments are done using MATLAB, we did not compare our result against this library.)

3 The sparse cardinal sine decomposition

In order to decouple the problem in spherical coordinates we notice that the formula (4) can be rewritten under the form

$$\begin{aligned}\sum_{l=1}^{N} \frac{1}{4\pi|x_k - x_l|} f_l &= \frac{1}{(2\pi)^3} \int_0^{+\infty} \sum_{l=1}^{N} \left(\int_{\mathbb{S}^2} e^{i\lambda x_k \cdot \xi} (e^{-i\lambda x_l \cdot \xi} f_l) d\xi \right) d\lambda \\ &= \frac{1}{2\pi^2} \int_0^{+\infty} \left(\sum_{l=1}^{N} \operatorname{sinc}(\lambda|x_k - x_l|) f_l \right) d\lambda \,,\end{aligned}$$

using $\int_{\mathbb{S}^2} e^{ix\cdot\xi} d\xi = 4\pi\,\mathrm{sinc}(|x|)$, for all $x \in \mathbb{R}^3$. Here, sinc stands for the cardinal sine function

$$\mathrm{sinc}(r) = \frac{\sin(r)}{r}.$$

This simply reflects that

$$\forall r \in \mathbb{R},\ \frac{1}{4\pi r} = \frac{1}{2\pi^2}\int_0^{+\infty} \mathrm{sinc}(\lambda r)\, d\lambda. \tag{5}$$

As we have pointed out in the preceding section, the first step of the method consists in finding a discrete quadrature rule

$$\frac{1}{4\pi r} \sim \sum_p \alpha_p\, \mathrm{sinc}(\lambda_p r). \tag{6}$$

This is a decomposition of the Green kernel into a series of dilated sinc functions. Notice that in the case of the Laplace kernel, this decomposition is equivalently rewritten, setting $\beta_p = \frac{4\pi\alpha_p}{\lambda_p}$, as

$$1 \sim \sum_p \beta_p \sin(\lambda_p r),$$

and we seek for a formula that should be valid as uniformly as possible in r and as sparse as possible in p. An exact formula exists, though containing an infinite number of terms, since for all $r \in (0, \pi)$, one has

$$\sum_{p\geq 0} \frac{4}{\pi(2p+1)} \sin\big((2p+1)r\big) = 1. \tag{7}$$

Indeed, the preceding formula comes from the decomposition in Fourier series of the 2π-periodic square function defined on $(-\pi, \pi)$ by

$$h(r) = \begin{cases} 1 \text{ if } r \in (0, \pi), \\ -1 \text{ if } r \in (-\pi, 0). \end{cases}$$

Unfortunately, due to the discontinuity of the square function at 0, the convergence of the series (7) is slow and highly depends on the value of r. Indeed, the series only converges because of the oscillatory behavior of the sine, which occurs later if r is small. It is also very well known that, truncating[3] the series (7) leads to a Gibbs phenomenon. Very high oscillations occur near 0 and π while in the center of the interval, the approximation is much better.

Truncating the series does not lead to a good approximation, and it is well known [1, 14, 24] that much better results can be obtained by using the so-called $\sigma-$approximation. It consists in replacing the truncated series

$$1 \sim \sum_{p=0}^{P-1} \frac{4}{\pi(2p+1)} \sin\left((2p+1)r\right)$$

[3] Alternative strategies, e.g. using Padé (see after) approximations lead to very good approximations of the square function (see [25]). They are however difficult to use in our context and unfortunately seem useless.

by

$$1 \sim \sum_{p=0}^{P-1} \frac{4}{\pi(2p+1)} \left[\operatorname{sinc}\left(\frac{2p+1}{2P}\pi\right) \right]^k \sin((2p+1)r)$$

where the term $\operatorname{sinc}\left(\frac{2p+1}{2P}\pi\right)$ is the Lanczos' $\sigma-$factor. A power $k = 1$ is usually used but the $\sigma-$factor can also be squared or cubed ($k = 2$ or 3) in the most extreme cases. As a sake of illustration, we show in Fig. 1 the graphs obtained with $P = 5$ terms in the sum and $k = 0, 1, 2$. It is clear that the approximation is much better and the Gibbs phenomenon is significantly reduced. However, as one can see, the approximation can be very good in the center of the interval and quite bad at the extremities.

In order to avoid the problems at the extremities, we consider the following approximation problem:

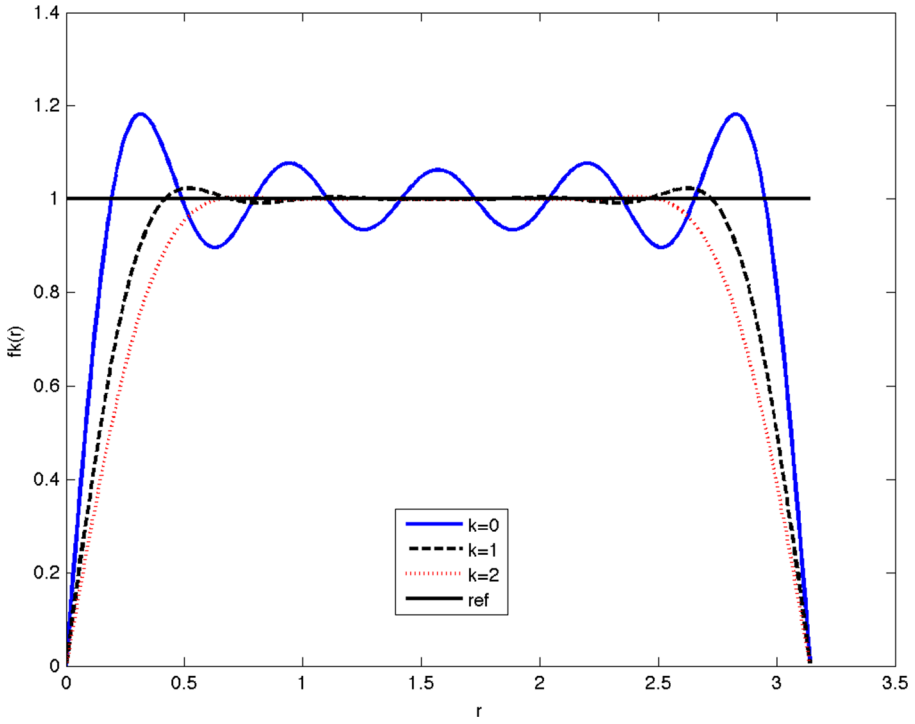


Fig. 1 The Gibbs phenomenon obtained with the convergence of the Fourier series of the square function on $[0, \pi]$. We give the approximations obtained by truncating the series after its 5th term and regularizing it by a $\sigma-$factor for $k = 0$ (dashed), $k = 1$ (plain) and $k = 2$ (dotted). The reference square is indicated as a horizontal line. The Gibbs phenomenon is clearly visible and is very much reduced as k increases

For a given $\rho \in \left(0, \frac{\pi}{2}\right)$, and P, find weights $(\beta_p)_{\{0 \le p \le P-1\}}$ such that the accuracy

$$\varepsilon = \sup_{r \in [\rho, \pi-\rho]} \left| \sum_{p=0}^{P-1} \beta_p \sin((2p+1)r) - 1 \right| \tag{8}$$

is as small as possible.

The treatment of small values for r is postponed to Section 5. It is expected that in order to maintain a given precision, P must be increased as ρ gets smaller. We propose to solve this problem via a least square approximation. This leads to the following natural algorithm

Algorithm 1 (least square approximation)

1. Given ρ, and P.
2. Compute $A = (A_{lp})_{1 \le l, p \le P}$ and $b = (b_l)_{1 \le l \le P}$ defined by
   $A_{lp} = \int_\rho^{\pi-\rho} \sin((2l-1)r) \sin\left((2p-1)r\right) dr$, and
   $b_l = \int_\rho^{\pi-\rho} \sin((2l-1)r) dr$.
3. Solve
   $A\beta = b$ where $\beta = (\beta_0, \cdots, \beta_{P-1})^t$.

Of course, both the matrix and the right-hand side can be computed analytically.

We have shown in Fig. 2 the approximation error on the interval $[\rho, \pi - \rho]$ obtained by the preceding algorithm compared to the ones obtained with the truncated Fourier series smoothed by the Lanczos $\sigma-$factor. It is noticeable that the approximation is better with our algorithm especially for small (or high) values of r, namely close to ρ or $\pi - \rho$. Moreover, the error is more uniformly distributed.

Eventually, one can translate the preceding formula to get an approximation for all $R \in [R_{\min}, R_{\max}]$ by choosing $\rho = \delta R_{\min}$, with $\delta = \frac{\pi}{R_{\min}+R_{\max}}$ (notice that then $\pi - \rho = \delta R_{\max}$). Setting $r = \delta R$ in the preceding formula we obtain

$$\varepsilon = \sup_{R \in [R_{\min}, R_{\max}]} \left| \sum_{p=0}^{P-1} \beta_p \sin((2p+1)\delta R) - 1 \right|,$$

and

$$\frac{1}{4\pi R} \sim \sum_{p=0}^{P-1} \frac{(2p+1)\delta\beta_p}{4\pi} \operatorname{sinc}((2p+1)\delta R), \tag{9}$$

which is exactly under the form (6) with $\lambda_p = (2p+1)\delta$, and $\alpha_p = \frac{\lambda_p \beta_p}{4\pi}$.

Remark that the accuracy ε of the method only relies on the parameters ρ and P, or conversely, for a given accuracy ε and ρ, one can find the smallest value for P.

Lemma 1 *Let $\rho \in (0, \pi/2)$. Construct for all $P \in \mathbb{N}$ the least square approximation h_P of h with the help of the Algorithm 1. Then, there exists $P_0 \in \mathbb{N}$ such that for all $P > P_0$, one has the estimation*

$$\|h - h_P\|_{L^2([\rho, \pi-\rho])} \le \frac{1 - \sin(\rho)}{\sqrt{\sin(\rho) P}} \left(\frac{1 - \sin(\rho)}{1 + \sin(\rho)} \right)^{P+1}.$$

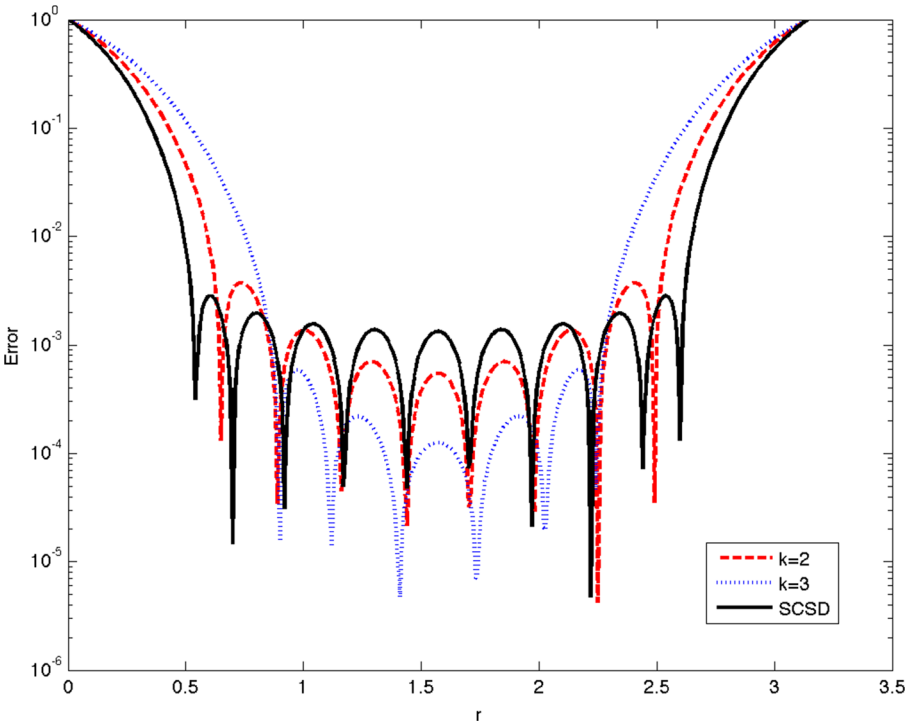


Fig. 2 Comparison of the errors for the approximation of the square function with Lanczos $\sigma-$approximation with k=2 (dashed), k=3 (dotted) and the SCSD (plain) all with 5 coefficients. In the SCSD, the error is more equilibrated on the whole interval and drops down quicker

Proof The proof is a quite direct consequence of the approximation of the sign function by polynomials. Indeed, let $\tilde{h}$ be the sign function defined on $[-1, 1]$ by

$$\tilde{h}(r) = \begin{cases} 1 \text{ if } r \in (0, 1), \\ -1 \text{ if } r \in (-1, 0). \end{cases}$$

□

We have obviously $h(r) = \tilde{h}(\sin(r)), \forall r \in \mathbb{R}$. We remark that the sought approximation $\sum_{p=0}^{P-1} \beta_p \sin((2p+1)r)$ can be rewritten as $\mathbb{P}_P(\sin(r))$ where $\mathbb{P}_P$ is a polynomial function of degree at most $2P-1$. Therefore, by a simple change of variable, approximating h with a truncated Fourier series amounts to approximate $\tilde{h}$ by a polynomial. It turns out that this problem has been widely studied in the literature, and even dates back to the 19th century [25] where the optimal approximation of $\tilde{h}$ with a Padé approximant is obtained. Quite strangely, the case of polynomial approximation has been treated much later in [7] (see also the references therein). In particular, it is shown that the best polynomial approximation $\mathbb{Q}_P$ of degree $2P-1$ of $\tilde{h}$ in L^∞ norm on $[-1, -a] \cup [a, 1]$ satisfies

$$\|\tilde{h} - \mathbb{Q}_P\|_{L^\infty([-1,-a]\cup[a,1])} \sim \frac{1-a}{\sqrt{\pi a P}} \left(\frac{1-a}{1+a}\right)^{P+1} \text{ as } P \to +\infty.$$

Table 1 Number of terms needed to reach an accuracy $\varepsilon = 10^{-3}$ (defined in (8) on $[\rho, \pi - \rho]$ for the truncated Fourier series with σ-factor to the power k and the least square approximation (SCSD)

ρ	$k = 0$	$k = 1$	k=2	k=3	SCSD	Predicted P
0.5	656	25	11	9	7	7
0.1	>1000	121	54	44	37	34
0.01	>1000	>1000	539	432	362	345

The last column gives the prediction of the number of terms for the sparse approximation SCSD using the formula (10)

We therefore deduce, setting $a = \sin(\rho)$ that

$$\begin{aligned}\|h(r) - \mathbb{P}_P(\sin(r))\|_{L^2([\rho,\pi-\rho])} &\le \|h(r) - \mathbb{Q}_P(\sin(r))\|_{L^2([\rho,\pi-\rho])} \\ &\le \sqrt{(\pi - 2\rho)}\|h(r) - \mathbb{Q}_P(\sin(r))\|_{L^\infty([\rho,\pi-\rho])} \\ &\le \sqrt{(\pi - 2\rho)}\|\tilde{h}(r) - \mathbb{Q}_P\|_{L^\infty([\sin(\rho),1])} \\ &\sim \sqrt{1 - \frac{2\rho}{\pi}}\frac{1 - \sin(\rho)}{\sqrt{\sin(\rho)P}}\left(\frac{1 - \sin(\rho)}{1 + \sin(\rho)}\right)^{P+1}\end{aligned}$$

as $P \to \infty$. The result follows.

The preceding result is used in order to fix the degree P of the approximation, depending on the requested precision ε (in L^2 norm) and the lower bound ρ down to which the approximation is sought. We find that

$$P \sim -\frac{\log(\varepsilon)}{2\sin(\rho)}, \tag{10}$$

or in the limit $R_{\min} \ll R_{\max}$,

$$P \sim -\frac{\log(\varepsilon)}{2\pi}\frac{R_{\max}}{R_{\min}}. \tag{11}$$

We present in Table 1 and Table 2, a comparison of the least square approximation and the truncated Fourier series with the Lanczos' $\sigma-$factor with a power k ranging from 0 to 3. Different values have been chosen for ρ, and we indicate the smallest number of terms (i.e. the value of P) to reach a precision $\varepsilon = 10^{-3}$ and $\varepsilon = 10^{-6}$ respectively, for the four methods. In both cases, we evaluate the error ε by sampling

Table 2 Number of terms needed to reach an accuracy $\varepsilon = 10^{-6}$ (defined in (8) on $[\rho, \pi - \rho]$ for the truncated Fourier series with σ-factor to the power k and the least square approximation (SCSD)

ρ	$k = 0$	$k = 1$	k=2	k=3	SCSD	Predicted P
0.5	>1000	>1000	106	44	14	14
0.1	>1000	>1000	>1000	188	71	69
0.01	>1000	>1000	>1000	>1000	708	690

The last column gives the prediction of the number of terms for the sparse approximation SCSD using the formula (10)

the interval $[\rho, \pi - \rho]$ with 1000 equally spaced points and computing the difference given in (8) at those points.

It is seen that the least square approximation outperforms the Lanczos' σ-approximation to any order. Moreover, the prediction (10) for the number of terms to keep for a given accuracy is very effective.

4 Spherical quadratures

Once the preceding radial quadrature is known, that is to say we have a discrete approximation

$$1 \sim \sum_{p=0}^{P-1} \beta_p \sin(\lambda_p R)$$

valid for $R \in [R_{\min}, R_{\max}]$, with $R_{\min} > 0$, at a precision ε, we proceed to the second step, namely the spherical quadrature. Let $x \in \mathbb{R}^3$ such that $R = |x| \in [R_{\min}, R_{\max}]$, we already know that

$$\frac{\sin(\lambda_p |x|)}{\lambda_p |x|} = \frac{1}{4\pi} \int_{\mathbb{S}^2} \exp(i\lambda_p x \cdot \xi) d\xi. \tag{12}$$

We again need a spherical quadrature to produce a discrete version of the above integral.

Spherical quadrature is a long standing subject, and several have been studied in the literature. We refer the reader to the review article [12] for a comprehensive explanation of the state of the art concerning this question. In what follows, we have chosen to use the Gauss-Legendre quadrature which consists in composing unidimensional quadratures (in the vertical axis z and the polar horizontal angle θ on the sphere $\mathbb{S}^2$). Positive weights are assured but there is a clear drawback that the underlying set of points is much denser near the poles than near the equator. Other techniques have therefore been proposed like the Lebedev grids [16], or other cubature rules [9]. Although very tempting because they need a fewer number of quadrature points, both techniques are limited in the degree of precision and much less adaptive than the Gauss-legendre quadrature.

The Gauss-Legendre quadrature is simple. Given a number M, it consists in discretizing the set of azimuthal angles with $2M$ angles

$$\theta_k = k\frac{\pi}{M}, k = 0, \cdots, 2M - 1,$$

while the elevation angles $(\psi_l)_{0 \le l \le M-1}$ are such that $\cos(\psi_l)$ is a zero of the M^{th} Legendre polynomial $\mathbb{L}_M$. The complete set of points used in the quadrature is thus

$$\xi_{kl} = (\cos(\theta_k)\sin(\psi_l), \sin(\theta_k)\sin(\psi_l), \cos(\psi_l)) \text{ with } L_M(\cos(\psi_l)) = 0.$$

for all values $0 \leq k \leq 2M-1$ and $0 \leq l \leq M-1$. The corresponding weights $(\omega_{kl})_{0\leq k\leq 2M-1, 0\leq l\leq M-1}$ do not depend on k (i.e. $\omega_{kl} = \omega_l$), and are such that the one dimensional quadrature formula

$$\int_{-1}^{1} f(x)dx \sim \sum_{l=0}^{M-1} \frac{\omega_l}{2\pi} f(\cos(\psi_l))$$

is actually the classical one dimensional Gauss-Legendre quadrature which is exact for all f polynomial of degree less than or equal to $2M-1$. In particular they are all positive. Similarly, Gauss-Legendre quadratures on the sphere are known to integrate exactly all the spherical harmonics Y_{lm} of degree $l \leq 2M-1$, or equivalently the restriction on the sphere of any function $g \in \mathcal{P}_{2M-1}$, where we have denoted by $\mathcal{P}_J$ the set of polynomial in $\xi = (\xi_1, \xi_2, \xi_3) \in$ of degree less than or equal to J.

Classical theorems (see e.g. [12]) usually express the precision of the quadrature formula in terms of the Sobolev or Hölder regularity of f_X. they are therefore useless in our context, since the function on which we need an estimation is known and analytic on the sphere. Instead of this, we are more interested in the smallest value that one should choose for M in the quadrature to obtain a required precision in the formula uniformly in X.

Lemma 2 *Let $X \in \mathbb{R}^3$, and f the function*

$$\begin{aligned} f_X : \mathbb{S}^2 &\to \mathbb{C} \\ \xi &\mapsto \exp(iX \cdot \xi). \end{aligned}$$

Then, we have, using the Gauss-Legendre quadrature, and provided $|X| < 2M+1$,

$$\left| \int_{\mathbb{S}^2} f_X(\xi)d\xi - \sum_{l=0}^{M-1} \sum_{k=0}^{2M-1} \omega_l f_X(\xi_{kl}) \right| \leq 8\pi \frac{|X|^{2M}}{(2M)!} \frac{2M+1}{2M+1-|X|}.$$

Proof For any function g, we set $I(g) = \int_{\mathbb{S}^2} g(\xi)d\xi$, the exact integral and $Q(g) = \sum_{l=0}^{M-1} \sum_{k=0}^{2M-1} \omega_l g(\xi_{kl})$ its approximation. We remark first that since for any $g \in \mathcal{P}_{2M-1}$, $I(g) = Q(g)$, we get the estimate

$$\begin{aligned} |I(f_X) - Q(f_X)| &\leq |I(f_X - g) - Q(f_X - g)| \\ &\leq \left(4\pi + \sum_{l=0}^{M-1} 2M|\omega_l| \right) \|f_X - g\|_\infty \\ &\leq 8\pi \|f_X - g\|_\infty, \forall g \in \mathcal{P}_{2M-1}, \end{aligned}$$

since the weights are all positive and the constant function $1 \in \mathcal{P}_{2M-1}$ leads to $Q(1) = 4\pi$. □

Now, expanding f_X in Taylor series, we have

$$f_X(\xi) = \sum_{m=0}^{\infty} \frac{(iX \cdot \xi)^m}{m!},$$

while taking $g(\xi) = \sum_{m=0}^{2M-1} \frac{(iX\cdot\xi)^m}{m!}$, in the preceding estimate, we obtain

$$\begin{aligned} |I(f_X) - Q(f_X)| &\leq 8\pi \|f_X - g\|_\infty \\ &\leq 8\pi \sup_{\xi\in\mathbb{S}^2} \sum_{m=2M}^{\infty} \left| \frac{(iX\cdot\xi)^m}{m!} \right| \\ &\leq 8\pi \frac{|X|^{2M}}{(2M)!} \frac{2M+1}{2M+1-|X|} \end{aligned}$$

if $2M + 1 > |X|$.

However a much better estimate can be obtained if X is vertical, that is to say $X = (0, 0, |X|)^t$. Indeed, one has the following lemma.

Lemma 3 *Let $\lambda > 0$ and with the preceding notation $f_{\lambda e_3} : \xi \to \exp(i\lambda\xi_3)$. Then one has*

$$|I(f_{\lambda e_3}) - Q(f_{\lambda e_3})| \leq \frac{2(2\lambda)^{2M}(M!)^4}{(2M+1)[(2M)!]^3}.$$

Proof First, we remark that the set of Gauss-Legendre quadrature points being symmetric with respect to the origin on the sphere, one can replace $f_{\lambda e_3}$ with its even part, namely $g_{\lambda e_3} : \xi \to \cos(\lambda\xi_3)$. Then, we notice that $f_{\lambda e_3}$ is also constant on the sphere along the horizontal circles. The integration along those circles is therefore exact and one thus deduces

$$\begin{aligned} |I(f_{\lambda e_3}) - Q(f_{\lambda e_3})| &= |I(g_{\lambda e_3}) - Q(g_{\lambda e_3})| \\ &= \left| \int_{-1}^{1} g_{\lambda e_3}(z)dz - \sum_{l=0}^{M-1} \omega_l g_{\lambda e_3}(\cos(\psi_l)) \right| \end{aligned}$$

which is nothing but the error of the 1D Gauss-Legendre quadrature of $g_{\lambda e_3}(z)$. Noticing that $g_{\lambda e_3}$ is an analytic function in a neighborhood of the segment $[-1, 1]$ in $\mathbb{C}$, a classical result (see e.g. [15] p. 146) applies which gives

$$|I(f_{\lambda e_3}) - Q(f_{\lambda e_3})| \leq \frac{2^{(2M+1)}(M!)^4}{(2M+1)[(2M)!]^3} \|g_{\lambda e_3}^{(2M)}\|_\infty$$

which gives the result. □

It is not difficult to see that Lemma 3 gives a much better estimation than Lemma 2. However, this estimation only holds for vertical X. Nevertheless, we have observed that such vectors provide the worst possible error for the Gaussian quadrature among vectors of the same magnitude. As a matter of fact, we report in Fig. 3 the quantity $|I(f_X) - Q(f_X)|$ for vectors X of the same magnitude and different azimuthal angles. It is clearly seen that the maximum error is obtained for azimuthal angles $\pm\pi/2$ i.e. vertical vectors X. We express this result as the following conjecture.

Conjecture 1 *We have*

$$\forall\lambda > 0\,, \quad \max_{X\in\lambda\mathbb{S}^2} |I(f_X) - Q(f_X)| = \left|I(f_{\lambda e_3}) - Q(f_{\lambda e_3})\right|$$

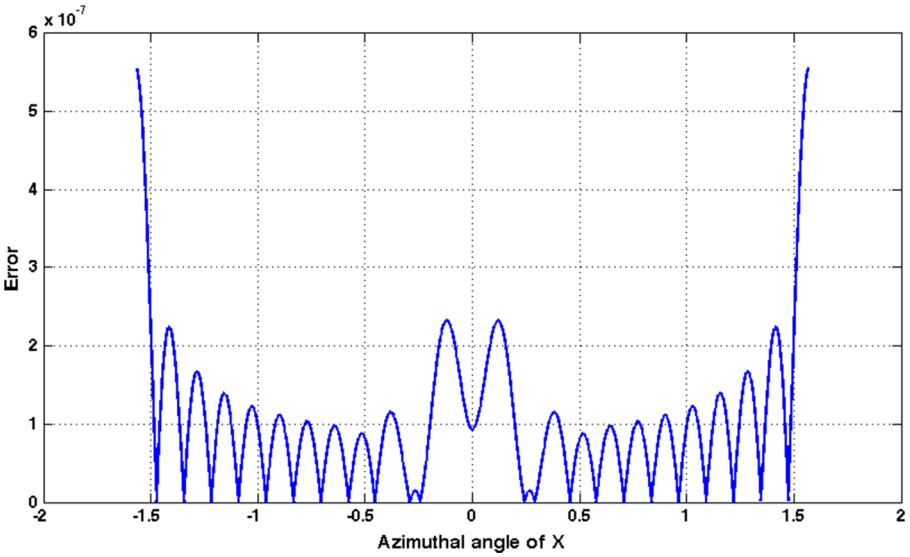


Fig. 3 A typical graph of the error $|I(f_X) - Q(f_X)|$ for vectors X of different azimuthal angles and magnitude 10. We have chosen a number of quadrature points of the Gauss-Legendre quadrature $2M \times M$ with $M = 12$

Although we have not yet been able to prove this result, we use the result obtained in Lemma 3 in the sequel for the construction of the whole sparse grid.

As before, we use Lemma 3. in order to compute the minimal value for M needed to achieve a desired accuracy ε'. Since, in view of (12), the maximum value for $|X|$ depends on p and is equal to $\lambda_p R_{\max} \leq (2p+1)\pi$, the corresponding degree M_p to which the quadrature should be exact can be estimated in order to satisfy

$$\frac{2((4p+2)\pi)^{2M}(M!)^4}{(2M+1)[(2M)!]^3} \leq \varepsilon'_p, \tag{13}$$

which, using Stirling formula leads to

$$M_p \sim \frac{(2p+1)\pi e}{4} - \frac{1}{2}\log\left(\frac{2\varepsilon'_p}{\sqrt{\pi}}\right). \tag{14}$$

We further rescale the quadrature points by setting $\zeta_{kl}^p = (2p+1)\delta\xi_{kl} = \frac{(2p+1)\pi\xi_{kl}}{R_{\min}+R_{\max}}$ and call the corresponding weights ω_l^p. Notice that we have emphasized the dependence in p, and that the points ζ_{kl}^p now discretize the sphere $\lambda_p\mathbb{S}^2$. We thus obtain a quadrature formula as

$$\frac{\sin(\lambda_p|x|)}{\lambda_p|x|} \sim \frac{1}{4\pi}\sum_{l=0}^{M_p-1}\sum_{k=0}^{2M_p-1}\omega_l^p \exp\left(ix\cdot\zeta_{kl}^p\right) \tag{15}$$

which, if M_p satisfies (14), is accurate to within an error ε'_p uniformly for all $x \in [R_{\min}, R_{\max}]$.

5 Close interactions

The formula (9) is only valid for $|x| \in [R_{\min}, R_{\max}]$. Therefore, interactions between neighboring points (pairs of points x_k and x_l such that $|x_k - x_l| \leq R_{\min}$ or equivalently such that $\delta|x_k - x_l| \leq \rho$) are wrongly computed, and need a fix. We thus correct the preceding computation with the (sparse) matrix multiplication

$$\forall k \in \{1, \cdots, N\}, \quad \tilde{g}_k = \sum_{l=1}^{N} B_{kl} f_l$$

where

$$B_{kl} = \frac{1}{4\pi|x_k - x_l|} - \sum_{p=0}^{P-1} \alpha_p \sum_{m=0}^{M_p-1} \sum_{n=0}^{2M_p-1} \omega_l^p \exp\left(i(x_k - x_l) \cdot \zeta_{mn}^p\right).$$

Fortunately, those interaction are only local and can be computed with a sparse matrix multiplication.

We remark that the methodology presented here is very close to an Ewald summation method, which is also called P^3M or $SPME$ in the literature [2]. Named after Paul Ewald [8], this method indeed splits the interactions into two parts, one which is local in space (the close interactions in our case) and the long range interactions which are treated in Fourier space. However, this method is usually only used for periodic problems. Notice that a generalization of the Ewald summation method exists for the free space (the so-called *fast summation method* [21, 23]), but there the Fourier grid is equispaced. We here instead take advantage of the radial structure of the Fourier transform of the kernel to propose a radial quadrature on spheres. We believe that this is not only more adapted to the singularity of the Fourier transform (this is especially true for Helmholtz kernel), but also permits to have a sparser quadrature formula which, at the end, leads to an efficient method. This is therefore the heart of the proposed methodology.

6 Summary of the method. Fixing the parameters

We summarize the method in the following algorithms. Algorithm 2 below explains the initialization step which merely consists in finding the global quadrature formula for a given precision and parameters $R_{\min}$ and $R_{\max}$. The Algorithm 3 given after provides the matrix vector product, using the preceding quadrature formula.

In order to evaluate the complexity of the method, we remark that steps 2. and 3. of Algorithm 3 can be realized using a Type 3 NUFFT. The complexity of the NUFFT is known to be $O(M \log M)$, where $M = \max(N, N_\zeta)$, where N_ζ is the number of quadrature points in the Fourier quadrature. The remaining of this Section thus consists in estimating N_ζ in terms of N and showing Proposition 1. Before proceeding, we estimate the error that one can make in the spherical quadratures which, when used in conjunction with the SCSD quadrature formula (9), will give a relative error of the order of ε, at least for all $R \in [R_{\min}, R_{\max}]$. In that aim, we use the following Lemma.

Algorithm 2 (Precomputation)

1. Given $(x_k)_{1\le k\le N}$, compute $R_{\max}$ and choose $R_{\min}$. Prescribe an accuracy ε, and set $\delta = \frac{\pi}{R_{\min}+R_{\max}}$.
2. Compute the Sparse Cardinal Sine Decomposition (Algorithm 2), using $\rho = \delta R_{\min}$, and $P = -\dfrac{\log(\varepsilon) R_{\min}}{2R_{\max}}$.
3. For all $p \in \{0, \cdots, P-1\}$, compute the Gauss-Legendre quadrature $(\omega_l^p, \zeta_{kl}^p)$.
4. Compute the sparse matrix B_{kl} to handle local interactions.

Algorithm 3 (Matrix vector product)

1. Given a set of values $(f_j)_{1\le j\le N}$, associated to the $(x_j)_{1\le j\le N}$.
2. Compute $\eta_{kl}^p = \displaystyle\sum_{j=1}^{N} \exp(-i x_j \cdot \zeta_{kl}^p) f_j$.
3. Compute $g_j^1 = \displaystyle\sum_{p,l,k} \exp(i x_j \cdot \zeta_{kl}^p) \alpha_p \omega_l^p \eta_{kl}^p$.
4. Compute $g_j^2 = \displaystyle\sum_{k=1}^{N} B_{jk} f_k$.
5. Compute $g_j = g_j^1 + g_j^2$.

Lemma 4 *Assume ε'_p in (13) is chosen as $\varepsilon'_p = \frac{\varepsilon}{4\pi R_{\max} P|\alpha_p|}$, where P is the number of terms of the Sparse Cardinal Sine Decomposition and the conjecture 1 is true, then the global quadrature ensures*

$$\sup_{|X|\in[R_{\min},R_{\max}]} 4\pi|X| \left| \frac{1}{4\pi|X|} - \sum_{p,k,l} \alpha_p \omega_l^p \exp(iX\cdot\zeta_{kl}^p) \right| \le 2\varepsilon.$$

Proof In view of the preceding Lemmas, we have

$$\begin{aligned}
&\left| 1 - 4\pi|X| \sum_{p,k,l} \alpha_p \omega_l^p \exp(iX\cdot\zeta_{kl}^p) \right| \\
&\quad\le \left| 1 - \sum_p \beta_p \sin(\lambda_p|X|) \right| + 4\pi|X| \sum_p |\alpha_p| \left| \operatorname{sinc}(\lambda_p|X|) - \sum_{kl} \omega_l^p \exp(iX\cdot\zeta_{kl}^p) \right| \\
&\quad\le \varepsilon + \sum_{p=0}^{P-1} 4\pi R_{\max}|\alpha_p|\varepsilon'_p \\
&\quad\le 2\varepsilon
\end{aligned}$$

if $\varepsilon'_p = \dfrac{\varepsilon}{4\pi R_{\max} P|\alpha_p|}$. Notice that $\beta_p \sim \frac{1}{2p+1}$ so that $\alpha_p = \frac{\lambda_p \beta_p}{4\pi} \sim \frac{\delta}{4\pi}$ and we deduce

$$\varepsilon'_p \sim \frac{\varepsilon}{P\pi} = -\frac{2\varepsilon}{\log(\varepsilon)} \frac{R_{\min}}{R_{\max}}.$$

□

Eventually, we may furthermore estimate the total number of points N_ζ of this quadrature by

$$\begin{aligned} N_\zeta &= \sum_{p=0}^{P-1} 2N_p^2 \\ &= \sum_{p=0}^{P-1} 2\left(\frac{(2p+1)\pi e}{4} - \frac{1}{2}\log\left(\frac{2\varepsilon'_p}{\sqrt{\pi}}\right)\right)^2 \\ &\sim \pi^2 e^2 \frac{P^3}{6} \\ &\sim \frac{e^2}{48\pi}\left(-\log(\varepsilon)\frac{R_{\max}}{R_{\min}}\right)^3 \end{aligned} \tag{16}$$

using (11).

We are now in a position to prove Proposition 1. Notice that the preceding error estimate relies on conjecture 1 but the number of points chosen in the method does not (only the precision does). Therefore the proposition follows from counting the number of short and long range interactions with the preceding methods.

Proof (of Proposition 1) Call $R_{\max}$ the diameter of the ball in which the points $(x_l)_{1\le l\le N}$ are uniformly distributed. Their density is thus given by

$$\rho = \frac{6}{\pi}\frac{N}{R_{\max}^3}.$$

The number of points contained in a ball of radius $R_{\min}$ is thus of the order of

$$\frac{4}{3}\pi R_{\min}^3 \rho = 8N\left(\frac{R_{\min}}{R_{\max}}\right)^3,$$

which gives the number of close interactions and correspondingly the number of terms in the correction matrix B as $8N^2\left(\frac{R_{\min}}{R_{\max}}\right)^3$.

The complexity of the method is thus governed by the number of points in the ζ mesh in the Fourier variable, given by formula (16), and the number of close interactions. Setting

$$R_{\min} = R_{\max}\left(\frac{e^2}{48\pi}\frac{(-\log(\varepsilon))^3}{8N^2}\right)^{1/6},$$

we get a total number of close interactions equal to the number of points N_ζ in the ζ grid

$$8N^2\left(\frac{R_{\min}}{R_{\max}}\right)^3 = \frac{e^2}{48\pi}\left(-\log(\varepsilon)\frac{R_{\max}}{R_{\min}}\right)^3 = \left(\frac{e^2}{6\pi}\right)^{1/2}(-\log(\varepsilon))^{3/2}N\,.$$

This clearly indicates that, in that case the complexity of the whole algorithm is $O(N\log N)$, the logarithmic term being due to the NUFFT [6].

When the points are uniformly distributed on a surface in 3D, the number of close interactions becomes $4N^2\left(\frac{R_{\min}}{R_{\max}}\right)^2$, and choosing now

$$R_{\min} = R_{\max}\left(\frac{e^2}{48\pi}\frac{(-\log(\varepsilon))^3}{4N^2}\right)^{1/5},$$

we obtain

$$4N^2\left(\frac{R_{\min}}{R_{\max}}\right)^2 = \frac{e^2}{48\pi}\left(-\log(\varepsilon)\frac{R_{\max}}{R_{\min}}\right)^3 = \left(\frac{e^2}{6\pi}\right)^{2/5}(-\log(\varepsilon))^{6/5}N^{6/5},$$

which now becomes only very slightly super-linear in N, giving an overall complexity $O(N^{6/5}\log(N))$. □

7 Generalization to other kernels

The preceding strategy can be of applied to other kernel than the Laplace one. Indeed, we see here that most – if not all – of the classically used kernels can be handled by using the formula given above. We split the following discussion into two parts: the non-oscillatory and the oscillatory Green kernels.

7.1 Non-oscillatory Green kernels

We put in this class kernels that are coming from equations as various as the equation of magnetostatics, those for linear elasticity and Stokes equations. Those equations are currently used with a very wide range of applications in particular in solid and fluid mechanics.

Les us start with the Stokes problem, which models a fluid flow at low Reynolds number

$$\begin{cases} -\mu\Delta u + \nabla p = f & \text{in } \mathbb{R}^3, \\ \operatorname{div} u = 0 & \text{in } \mathbb{R}^3. \end{cases}$$

Here, f is the force density and $(u,\, p)$ are respectively the fluid velocity and pressure. The parameter μ stands for the viscosity of the fluid. In the free space, this system can be solved by using Fourier variables. Indeed, one has $\hat{u}(\xi) = \hat{\mathcal{S}}(\xi)\hat{f}(\xi)$ where

$$\hat{\mathcal{S}}(\xi) = \frac{1}{\mu|\xi|^2}\left(\mathrm{Id} - \frac{\xi\otimes\xi}{|\xi|^2}\right),$$

which going back to real space leads to the following formula for the so-called Stokeslet $\mathcal{S}$

$$\mathcal{S}(x) = \frac{1}{8\pi\mu|x|}\left(\mathrm{Id} + \frac{x \otimes x}{|x|^2}\right).$$

As for the magnetostatic case, solving Stokes equations in the free space, namely computing for a given force distribution $(f_j)_{1\leq j\leq N}$ at the points $(x_j)_{1\leq j\leq N}$ the velocities

$$\forall i \in \{1, \cdots, N\}, u_i = \sum_{j=1}^{N} \mathcal{S}(x_i - x_j) f_j$$

can be done by the following procedure

Algorithm 4

1. Compute $\hat{f}(\zeta_s)$ using the NUFFT.
2. Compute $\frac{1}{8\pi\mu}\left(\mathrm{Id} - \frac{\zeta_s \otimes \zeta_s}{|\zeta_s|^2}\right)\hat{f}(\zeta_s)$ for all s.
3. Multiply by the SCSD weights computed from the Laplace formula.
4. Go back to real space by the (inverse) NUFFT.

Eventually, the case of linear elasticity follows the same procedure. Namely, the equation becomes now

$$-\mu\Delta u - (\mu + \lambda)\nabla \operatorname{div}(u) = f \text{ in } \mathbb{R}^3,$$

where μ and λ are the Lamé coefficients of a material, u is its deformation, and f the applied force density. As before, taking the Fourier transform of this equation leads to $\hat{u}(\zeta) = \hat{\mathcal{E}}(\zeta)\hat{f}(\zeta)$ where

$$\hat{\mathcal{E}}(\zeta) = \frac{1}{\mu|\zeta|^2}\left(\mathrm{Id} - \frac{\mu + \lambda}{2\mu + \lambda}\frac{\zeta \otimes \zeta}{|\zeta|^2}\right),$$

and the modification that one needs to bring to Algorithm 2 is straightforward.

7.2 Oscillatory Green kernels

The classical oscillatory Green kernel that people use in practice quite often is

$$G(x) = \frac{\exp(ik|x|)}{4\pi|x|}.$$

It is well-known that G is the fundamental solution to the Helmholtz equation together with Sommerfeld radiation condition at ∞

$$\begin{cases} \Delta G + k^2 G = \delta_0 \ \text{ in } \mathbb{R}^3, \\ \frac{\partial G}{\partial R} - ikG = O\left(\frac{1}{R^2}\right) \ \text{ as } R = |x| \to \infty, \end{cases}$$

where δ_0 stands for the Dirac mass centered at 0. The function G plays a prominent role in acoustics and electromagnetics when solving the 3D diffraction problem of an object in the free space. In order to use the method detailed in this paper, we need to

obtain a sinc decomposition of the radial function $g(R) = \frac{\exp(ikR)}{4\pi R}$, or equivalently to write

$$\exp(ikR) \sim \sum_p \gamma_p \sin(\lambda_p R)$$

for weights $(\gamma_p)_{0\leq p\leq P-1}$ and scaling factors $(\lambda_p)_{0\leq p\leq P-1}$ to be determined. Again, we expect the preceding formula to be valid as uniformly as possible as R ranges between $R_{\min}$ and $R_{\max}$ (values $R \leq R_{\min}$ correspond to local interactions and are handled as for Coulombian interactions by considering a sparse matrix).

Obviously, the imaginary part of the kernel is already a sine, and we therefore need only to concentrate on the real part, namely the cosine function. Notice that if the $(\lambda_p)_{0\leq p\leq P-1}$ were known, we could compute the weights by the same least-square procedure that we explained before.

Strangely, a formula can be obtained from the preceding one. Indeed, let's consider the preceding decomposition in Fourier series of the square function

$$1 \sim \sum_{p=0}^{P-1} \beta_p \sin\big((2p+1)\delta R\big).$$

valid for $R \in [R_{\min}, R_{\max}]$, with $\delta = \frac{\pi}{R_{\min}+R_{\max}}$. We multiply both sides by $\cos(kR)$ to obtain

$$\cos(kR) \sim \sum_{p=0}^{P-1} \beta_p \sin\big((2p+1)\delta R\big)\cos(kR),$$

while using $\sin\big((2p+1)\delta R\big)\cos(kR) = \frac{1}{2}\sin\big((k+(2p+1)\delta)R\big) - \frac{1}{2}\sin\big((k-(2p+1)\delta)R\big)$, gives

$$\cos(kR) \sim \sum_{p=0}^{P-1} \frac{\beta_p}{2}(\sin\big((k+(2p+1)\delta)R\big) - \sin\big((k-(2p+1)\delta)R\big),$$

which is under the desired form, and provides us with the right values for the sequence $(\lambda_p)_p$. We thus obtain this striking result that any sinc decomposition of the square function (that correspond to Laplace equation) gives instantaneously a suitable approximation for the Helmholtz kernel and consequently a fast method. It is needless to say that this is not the case for the FMM for instance where having written a FMM approximation for the Laplace equation does not give any hint about the writing of a similar method for the Helmholtz equation. To finish, notice that we just obtain twice as many values as before, clustered around the frequency k of the problem.

8 Numerical results

A MATLAB implementation has been developed and compared to the direct call to the FMM Laplace and Helmholtz routines provided on the NYU site. Our method, instead, relies on the NUFFT routine also available on the same website and the sinc

Table 3 Comparison between the SCSD and the FMM for the Laplace operator, for a number of points ranging from 10^3 to 10^6 uniformly randomly distributed in a cube of diameter $R_{\max}$ and $R_{\min} = 1$

N	$R_{\min}$	$R_{\max}$	N_ζ	T_P (s)	T_{MV} SCSD (s)	T_{MV} FMM (s)
10^3	1	10	$1.2 \cdot 10^4$	0.23	0.060	0.026
10^4	1	22	$5.9 \cdot 10^4$	0.27	0.27	0.31
10^5	1	47	$3.5 \cdot 10^5$	2.0	4.4	7.8
10^6	1	100	$2.3 \cdot 10^6$	37	28	75

The columns report respectively the total number N_ζ of quadrature points in Fourier space, the time T_P for the precomputation of the SCSD, the time T_{MV} for a matrix-vector product with the SCSD and eventually the time for the FMM. For both methods an error $\varepsilon = 10^{-3}$ is chosen

decompostition that we have exposed. Besides the NUFFT kernel routine, everything else is written in MATLAB. We give in Table 3 (for the Laplace kernel) and Table 4 (for the Helmholtz kernel) the performances in terms of accuracy and CPU-time obtained by both methods on a set of points ranging from $N = 10^3$ to $N = 10^6$. The points are uniformly randomly distributed in a cube of diameter R_{max}. In both cases, the first column gives the preprocessing time (for computing the integration points and weights), and the second row gives the time for a matrix-vector product (mainly the time spent in the NUFFT). Third column gives the total time that needs to be compared to the time of the FMM routine (fourth column).

It is noteworthy to say that although the total time of the SCSD algorithm ($T_P + T_{MV}$) is very comparable to the FMM and scale identically. We also present in Fig. 4 curves that show the scalings of the CPU-time for the three methods ($O(N^2)$ Direct product, FMM, SCSD) in terms of the number of points, for Laplace and Helmholtz kernels.

The computer used for this test is a laptop cadenced to 1.6 GHz and possessing 8 GB of memory. For a more fair comparison, both the FMM and the NUFFT routines have been recompiled to run on a single thread and MATLAB is also run on a single thread. Parallelization strategies of the method will be considered elsewhere.

Table 4 Comparison between the SCSD and the FMM for the Helmholtz operator, for a number of points ranging from 10^3 to 10^6 uniformly randomly distributed in a cube of diameter $R_{\max}$ and $R_{\min} = 1$

N	$R_{\min}$	$R_{\max}$	N_ζ	T_P (s)	T_{MV} SCSD (s)	T_{MV} FMM (s)
10^3	1	10	$1.3 \cdot 10^4$	0.22	0.060	0.093
10^4	1	22	$7.8 \cdot 10^4$	0.29	0.36	1.5
10^5	1	47	$4.8 \cdot 10^5$	2.3	2.9	38
10^6	1	100	$3.1 \cdot 10^6$	39	36	475

The columns report respectively the total number N_ζ of quadrature points in Fourier space, the time T_P for the precomputation of the SCSD, the time T_{MV} for a matrix-vector product with the SCSD and eventually the time for the FMM. For both methods an error $\varepsilon = 10^{-3}$ is chosen

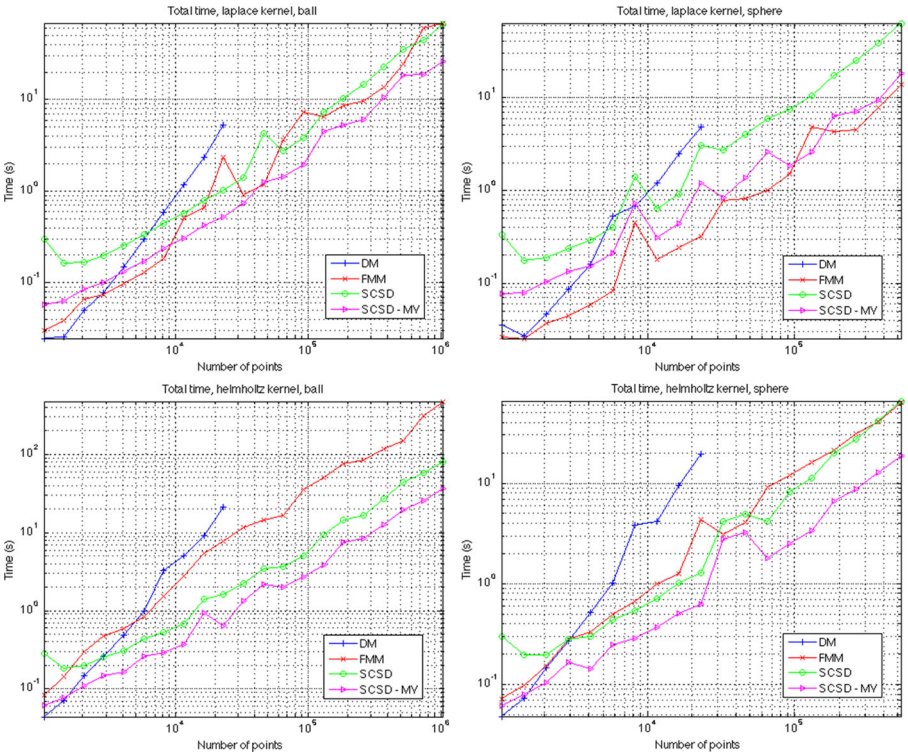


Fig. 4 Total time for a convolution with Laplace (top row) and Helmholtz (bottom row) kernels in 3D using the Direct Method (blue), the FMM (red) and the SCSD (green). Also reported is the time for the SCSD without the time for the precomputations (purple). The same error $\varepsilon = 10^{-3}$ is used for the FMM and the SCSD and a number of points between 10^3 and 10^6 is used uniformly distributed in a ball (left) and on a sphere (right). Timings for the FMM and the SCSD are very similar

9 Conclusion

We have presented an original alternative to the FMM for the computation of the coulombian interaction between punctual masses. Although the method is much simpler than the FMM, it turns out to be comparably fast, and only needs a simple NUFFT routine to apply. The heart of the method consists in using a spherical Gaussian quadrature in the Fourier space which is developed in two steps: first a sparse radial quadrature is computed to high accuracy and then spherical quadrature are obtained with simple Gauss-Legendre meshes. The analysis of the complexity of the method shows a very good behavior, comparable to the complexity of the FMM and extensions to classical kernels, in particular oscillatory ones, are given.

Acknowledgments Both the FMM and the NUFFT codes have been taken on the CMCL website [4, 19, 20]. We gratefully thank Leslie Greengard for having provided the community with such tools. We also thank the unknown referee who has indicated a few referennces, especially [23] that first escaped to our notice.

This research was partially supported by the iCODE institute, research project of the Idex Paris-Saclay.

References

1. Acton, F.S.: Numerical Methods That Work. Math. Assoc. Amer., Washington D.C. (1990)
2. Arnold, A., Fahrenberger, F., Holm, C., Lenz, O., Bolten, M., Dachsel, H., Halver, R., Kabadshow, I., Gähler, F., Heber, F., Iseringhausen, J., Hofmann, M., Pippig, M., Potts, D., Sutmann, G.: Comparison of scalable fast methods for long-range interactions. Phys. Rev. E **88**, 063308 (2013)
3. Chaillat, S., Bonnet, M.: A new fast multipole formulation for the elastodynamic half-space green's tensor. J. Comput. Phys. **256**, 787–808 (2014)
4. See http://www.cims.nyu.edu/cmcl/cmcl.html
5. Dutt, A., Rokhlin, V.: Fast Fourier transforms for nonequispaced data. SIAM J. Sci. Comput., 14 (1993)
6. Elbel, B., Steidl, G.: Fast fourier transforms for nonequispaced data. In: Chui, C.K., Schumaker, L.L. (eds.): Approximation Theory IX, pp. 39–46. Vanderbilt University Press, Nashville (1998)
7. Eremenko, A., Yuditskii, P.: Uniform approximation of sgn(x) by polynomials and entire functions. J. Anal. Math. **101**, 313–324 (2007)
8. Ewald, P.: Dir Berechnung optischer und elektrostatischer Gitterpotentiale. In Ann. Phys. **369**, 253–287 (1921)
9. Fliege, J., Maier, U.: A two-stage approach for computing cubature formulae for the sphere. In Mathematik 139T, Universitat Dortmund, Fachbereich Mathematik, Universitat Dortmund, 44221 (1996)
10. Greengard, L., Lin, P.: Spectral approximation of the free-space heat kernel. Appl. Comput. Harmonic Anal. **9**, 83–97 (2000)
11. Greengard, L.: The rapid evaluation of potential fields in particle systems. MIT Press (1988)
12. Hesse, K., Sloan, I.H., Womersley, R.S.: Numerical integration on the sphere. In: Handbook of Geomathematics, pp. 1185–1219. Springer (2010)
13. Hackbusch, W.: Hierarchische Matrizen. Springer (2009)
14. Hamming, R.W. Numerical Methods for Scientists and Engineers, 2nd Ed. Dover, New York (1986)
15. Kahaner, D., Moler, C., Nash, S.: Numerical methods and software. Prentice-Hall (1989)
16. Lebedev, V.I., Laikov, D.N.: A quadrature formula for the sphere of the 131st algebraic order of accuracy. Dokl. Math. **59**(3), 477–481 (1999)
17. Lee, J.-Y., Greengard, L.: Accelerating the nonuniform fast Fourier transform. SIAM Rev. **48**, 443–454 (2004)
18. Lee, J.-Y., Greengard, L.: The type 3 nonuniform fft and its application. J. Comput. Phys. **206**, 1–5 (2005)
19. See http://www.cims.nyu.edu/cmcl/nufft/nufft.html
20. See http://www.cims.nyu.edu/cmcl/fmm3dlib/fmm3dlib.html
21. Pippig, M., Potts, D.: Particle simulation based on nonequispaced fast Fourier transforms. In: Fast Methods for Long-Range Interactions in Complex Systems, IAS-Series, pp. 131–158. Forschungszentrum Jülich (2011)
22. See http://www-user.tu-chemnitz.de/~potts/nfft/
23. Potts, D., Steidl, G., Nieslony, A.: Fast convolution with radial kernels at nonequispaced knots. Numer. Math. **98**, 329–351 (2004)
24. Weisstein, E.W.: Lanczos σ factor. Mathworld-A Wolfram Web Resource. http://mathworld.wolfram.com/LanczosSigmaFactor.html
25. Zolotarev, E.I.: Application of the elliptic functions to questions on functions of least and most deviation from zero. Bull. de l'Académie de Sci. de St.-Petersbourg **24**(3) (1878)