# A Fast Algorithm for Spherical Grid Rotations and its Application to Singular Quadrature

Zydrunas Gimbutas* Shravan Veerapaneni†

September 11, 2013

**Abstract**

We present a fast and accurate algorithm for evaluating singular integral operators on smooth surfaces that are globally parametrized by spherical coordinates. Problems of this type arise, for example, in simulating Stokes flows with particulate suspensions and in multi-particle scattering calculations. For smooth surfaces, spherical harmonic expansions are commonly used for geometry representation and the evaluation of the singular integrals is carried out with a spectrally accurate quadrature rule on a set of rotated spherical grids. We propose a new algorithm that interpolates function values on the rotated spherical grids via hybrid nonuniform FFTs. The algorithm has a small complexity constant, and the cost of applying the quadrature rule is nearly-optimal $\mathcal{O}(p^4 \log p)$ for a spherical harmonic expansion of degree $p$.

**Keywords:** Spherical harmonics; boundary integral equations; singular quadrature; interpolation; non-uniform FFT.

**Mathematical Subject Classifications:** 45B05, 65R20, 65T40.

## 1 Introduction

Evaluating layer potentials is a common task that arises while solving boundary value problems via the classical potential theory. The *single-layer potential*, for instance, is given by

$$\mathcal{S}[f](\mathbf{x}) = \int_{\Gamma} G(\mathbf{x}, \mathbf{y}) f(\mathbf{y})\, d\Gamma(\mathbf{y}), \tag{1}$$

where $G$ is the Green's function (or fundamental solution), $f$ is the density and $\Gamma$ is the boundary. The Green's functions are known in closed analytic form for linear partial differential equations (PDEs) with constant coefficients, for example, $G(\mathbf{x}, \mathbf{y}) = 1/4\pi\|\mathbf{x} - \mathbf{y}\|$ for the Laplace equation.

*National Institute of Standards and Technology, Information Technology Laboratory, 325 Broadway, Mail Stop 891.01, Boulder, CO 80305. *email*: zydrunas.gimbutas@nist.gov. The work of this author was supported in part by the Air Force Office of Scientific Research under NSSEFF Program Award FA9550-10-1-0180 and in part by the National Science Foundation under grant DMS-0934733. Contribution of U.S. Government; not subject to copyright in the United States.

†Department of Mathematics, University of Michigan, Ann Arbor, MI 48109. *email*: shravan@umich.edu. The work of this author was supported by the National Science Foundation under grant DMS-1224656.

In this article, we are interested in fast and accurate algorithms for computing layer potentials when the boundary $\Gamma$ is smooth and topologically equivalent to a sphere. One particular application of interest is the numerical simulation of interfacial flows involving bubbles, vesicles or capsules [15]. For these problems, one needs to evaluate (1) on the surface of a deforming particle with the Stokesian fundamental solution and a geometry-dependent interfacial force. For example, in the case of bubble flows, $f = H\mathbf{n}$, where $H$ is the mean curvature and $\mathbf{n}$ is the unit normal to $\Gamma$ [15]. In multiple particle simulations, when a suitable fast algorithm such as the Fast Multipole Method (FMM) [9] is used to compute the nonlocal hydrodynamic interactions, evaluating (1) on individual particles dominates the computational cost of interfacial flow simulations, [16]. Even a modest acceleration in layer potential evaluation directly translates to faster simulations.

For a large class of interfacial problems, the deforming bodies are smooth. Spherical harmonic representations are particularly well suited in this setting, both for representing geometry and function densities, and have been widely used in different applications [5, 23, 25]. Such representations require relatively fewer degrees of freedom owing to their spectral accuracy. In collocation schemes, it is also convenient to sample geometry information and function densities on the suitably chosen set of discretization points. The standard discretization of the sphere is the "tensor product" of all pairs $(\theta_j, \phi_k)$ in the parametric domain, which are given by

$$\left\{\phi_k = \frac{2\pi k}{2p+2},\, k = 0, \ldots, 2p+1\right\} \quad \text{and} \quad \left\{\theta_j = \cos^{-1}(t_j),\, j = 0, \ldots p\right\}, \tag{2}$$

where $p$ is the approximation degree of the spherical harmonic expansion and $t_j$'s are the nodes of the $(p+1)$-point Gauss-Legendre quadrature on $[-1, 1]$. In this paper, we will refer to these discretization nodes as the *spherical Gaussian grid* or simply as the *spherical grid*. The forward and backward spherical harmonic transforms [18, 21] can be used to convert between the coefficients of spherical harmonic representations and the values of function densities on the spherical grid with $\mathcal{O}(p^3)$ work.

If the target point $\mathbf{x}$ in (1) coincides with either the north or the south pole, the integrand is non-singular in the parametric domain [1]. Using this fact, a quadrature rule has been developed in [7] based on rotating the coordinate system so that the north pole of the rotated spherical harmonic expansion is pointing to the target, then, an auxiliary spherical grid in the rotated coordinate system is used for a quadrature rule to evaluate the singular integral. The process is then repeated for a number of targets that are usually chosen to be the spherical grid points. The problem of finding all such auxiliary grids can be summarized as follows:

**Problem 1.** *Given the p-th order spherical harmonic representation of a function $f$ and a spherical grid of size $\mathcal{O}(p^2)$, compute the values of $f$ on $\mathcal{O}(p^2)$ auxiliary rotated grids whose north pole locations coincide with the original spherical grid points.*

There are a number of numerical techniques for solving this problem. For small values of $p$, a strategy is developed in [7] to rotate the spherical harmonic expansions with a computational cost of $\mathcal{O}(p^5)$. For a fixed geometry, the operators can be precomputed using $\mathcal{O}(p^5)$ work and $\mathcal{O}(p^4)$ storage [6], and the cost of applying the operators is $\mathcal{O}(p^4)$. For deforming geometries, the precomputation step is required at every time step so the total computational cost remains $\mathcal{O}(p^5)$.

For large values of $p$, a simple asymptotically optimal algorithm is to interpolate the input data given at $\mathcal{O}(p^2)$ points in the parametric domain to the $\mathcal{O}(p^4)$ auxiliary points. The nonuniform FFT (nuFFT) [3, 8, 11], can be used for this task which requires $\mathcal{O}((\sigma p)^2 \log(\sigma p) + |\log 1/\varepsilon|^2 p^4)$

work, where $\varepsilon$ is the required precision, and $\sigma$ is the oversampling parameter in the nuFFT algorithm. While the asymptotic complexity of this scheme is optimal, the $\mathcal{O}(|\log 1/\varepsilon|^2)$ scaling of the complexity constant limits its use in solving Problem 1 for practical problem sizes. For example, following [8], if the dilated Gaussian window function is used with the oversampling parameter $\sigma = 2$ and the spreading constant $m$ set to 12 to achieve 12 digits of accuracy, then the complexity constant is $(2m+1)^2 = 625$. This constant can further be reduced to 225 by using the Kaiser-Bessel window, [11, 12, 13], for the same choice of oversampling and precision parameters, but the scheme now requires expensive window function evaluations. In order to avoid that cost, one can precompute all the necessary quantities and store them in a table at the expense of extra storage. The storage costs scale like $\mathcal{O}(p^4)$, if the full or tensor product precomputation is used. As a workaround to potentially huge storage costs, a memory-efficient linear table lookup scheme can be used [11, 14]. The storage cost for this scheme scales like $\mathcal{O}(2K)$, where $K$ is a precision dependent parameter, with only minor performance loss.

Regardless of the precomputation strategy, the generic interpolation process in two dimensions requires $\mathcal{O}((2m+1)^2)$ multiplications per evaluation point, where $m$ is the spreading parameter that depends on the precision required and the choice of window function. Even for modest precision requirements of about 5 digits of accuracy, if the nearly optimal Kaiser-Bessel window function is used with the reasonable choice of the oversampling parameter $\sigma = 2$ and the spreading constant $m$ set to 3, the corresponding complexity constant is 49. This is sometimes called the curse of dimensionality [8]; without additional information about separability of interpolation directions in two dimensions, the complexity constant is proportional to a square of the spreading parameter.

**Key Idea.** Our algorithm is based on a simple geometric observation, that for a collection of regularly rotated tensor-product spherical grids, the $\mathcal{O}(p^4)$ points to be interpolated are located on $\mathcal{O}(p^3)$ latitudes while uniformly spaced and shifted by latitude-dependent phases. The interpolation is performed by applying the one dimensional nonuniform FFT along the meridians on a dense intermediary $\mathcal{O}(p^3) \times \mathcal{O}(p)$ tensor-product grid, which requires $\mathcal{O}(p(\sigma p)\log(\sigma p) + |\log 1/\varepsilon| p^4)$ work, then, by shifting the dense grid data to the rotated grids via standard one dimensional FFTs with $\mathcal{O}(p^4 \log p)$ work. It has a nearly optimal computational complexity of $\mathcal{O}(p^4 \log p)$ and, additionally, the complexity constant is significantly smaller, compared to the two-dimensional interpolation scheme, namely, we gain an extra factor of $\mathcal{O}(|\log 1/\varepsilon|)$. In particular, if the oversampling parameter $\sigma = 2$ and 12 digits of accuracy is desired, the complexity constant is 25 for the Gaussian window, and 15 for the Kaiser-Bessel window.

The paper is organized as follows. In Section 2, we first review spherical harmonic representations and spectral quadrature rules for singular integrals. Then, we present a standard numerical algorithm for rotating spherical transforms and conclude the section with an acceleration technique to improve its performance. In Section 3, a hybrid nuFFT based fast algorithm is developed for applying the spectral quadrature rule, followed by numerical results and a brief discussion of future work in Sections 4 and 5.

## 2 Mathematical Preliminaries

A spherical harmonic of degree $n$ and order $m$ is denoted by $Y_n^m$ and defined by

$$Y_n^m(\theta, \phi) = \sqrt{\frac{2n+1}{4\pi}} \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} P_n^{|m|}(\cos\theta)\, e^{im\phi}, \tag{3}$$

where $\phi$ is the azimuthal angle of the target point with respect to the $x$-axis, $\theta$ is the polar angle with respect to the $z$-axis, and $P_n^m$ are the associate Legendre functions, [2, 17]. Any scalar function $f$ on $\Gamma$ can be expanded in terms of spherical harmonics as

$$f(\theta,\phi) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} f_n^m \, Y_n^m(\theta,\phi), \tag{4}$$

where the coefficients $f_n^m$ are the moments of the expansion. For smooth functions, the finite-term approximation of (4) given by

$$f(\theta,\phi) \approx \sum_{n=0}^{p} \sum_{m=-n}^{n} f_n^m \, Y_n^m(\theta,\phi) \tag{5}$$

is superalgebraically convergent with $p$. Similarly, for a smooth geometry, the boundary $\Gamma$ can be conveniently described by a set of spherical harmonic coefficients $\mathbf{x}_n^m$, so that for all $\mathbf{x} \in \Gamma$, the coordinate functions $\mathbf{x}(\theta,\phi)$ are approximated as

$$\mathbf{x}(\theta,\phi) \approx \sum_{n=0}^{p} \sum_{m=-n}^{n} \mathbf{x}_n^m \, Y_n^m(\theta,\phi), \tag{6}$$

$$\theta \in [0,\pi], \quad \phi \in [0,2\pi].$$

The area element $W$, needed for evaluation of surface integrals, can be computed from (6) as

$$\mathbf{x}_\theta = \sum_{n=0}^{p} \sum_{m=-n}^{n} \mathbf{x}_n^m \, \frac{\partial}{\partial\theta} Y_n^m(\theta,\phi), \quad \mathbf{x}_\phi = \sum_{n=0}^{p} \sum_{m=-n}^{n} \mathbf{x}_n^m \, \frac{\partial}{\partial\phi} Y_n^m(\theta,\phi), \tag{7}$$

$$W(\theta,\phi) = |\mathbf{x}_\theta \times \mathbf{x}_\phi|. \tag{8}$$

### 2.1 Spherical Grid Based Quadrature Rules

The moments $f_n^m$ can be evaluated using the formula

$$f_n^m = \int_0^{\pi} \int_0^{2\pi} f(\theta,\phi) \overline{Y_n^m(\theta,\phi)} \sin\theta \, d\theta \, d\phi \tag{9}$$

and a suitable numerical integration scheme. A standard choice is to use the trapezoidal rule along $\phi$-direction and Gauss-Legendre quadrature along $\theta$-direction. For an approximation of degree $p$, the spherical grid has $2p+2$ equispaced nodes in the $\phi$-direction and $p+1$ nodes along the $\theta$-direction given by (2). The numerical integration scheme for smooth functions is then given by

$$\int_\Gamma f(\mathbf{y}) \, d\Gamma(\mathbf{y}) \approx \sum_{j=0}^{p} \sum_{k=0}^{2p+1} w_j f(\mathbf{y}(\theta_j,\phi_k)) \frac{W(\theta_j,\phi_k)}{\sin\theta_j}, \quad \text{where} \quad w_j = \frac{2\pi}{2p+2} \lambda_j \tag{10}$$

and $\lambda_j$'s are the Gauss-Legendre quadrature weights.

Now, we turn our attention back to the singular layer potential (1) which can be written in the spherical coordinate basis as

$$\mathcal{S}[f](\mathbf{x}) = \int_0^{\pi} \int_0^{2\pi} G(\mathbf{x}, \mathbf{y}(\theta,\phi)) \, f(\mathbf{y}(\theta,\phi)) \, W(\theta,\phi) \, d\theta \, d\phi. \tag{11}$$

If the target $\mathbf{x}$ lies on $\Gamma$, the rule (10) is inefficient for computing layer potentials because the Green's functions arising from linear constant-coefficient elliptic PDEs have $1/r$ type of singularity:

$$G(\mathbf{x},\mathbf{y}) = \begin{cases} \frac{1}{4\pi\|\mathbf{x}-\mathbf{y}\|}, & \text{Laplace,} \\ \frac{1}{4\pi\|\mathbf{x}-\mathbf{y}\|} e^{ik\|\mathbf{x}-\mathbf{y}\|}, & \text{Helmholtz,} \\ \frac{1}{8\pi}\left[\frac{1}{\|\mathbf{x}-\mathbf{y}\|}\mathbf{I} + \frac{(\mathbf{x}-\mathbf{y})\otimes(\mathbf{x}-\mathbf{y})}{\|\mathbf{x}-\mathbf{y}\|^3}\right], & \text{Stokes.} \end{cases} \tag{12}$$

In the case of spherical grids, when $\mathbf{x}$ coincides with one of the poles where the area element $W$ is zero, however, the following theorem holds:

**Theorem 1.** (Quadrature rule for singular integrals at poles [6, 7, 23]). *For any smooth function $f$ defined on a $C^\infty$ surface $\Gamma$ globally parametrized by spherical coordinates $(\theta,\phi)$, the quadrature rule for computing the Laplace potential at the north-pole $\mathbf{x}(0,0)$ given by*

$$\frac{1}{4\pi}\int_\Gamma \frac{f(\mathbf{y})}{\|\mathbf{x}(0,0)-\mathbf{y}\|}\,d\Gamma(\mathbf{y}) \approx \frac{1}{4\pi}\sum_{j=0}^{p}\sum_{k=0}^{2p+1} \frac{w_j^s}{\|\mathbf{x}(0,0)-\mathbf{y}(\theta_j,\phi_k)\|}\, f(\mathbf{y}(\theta_j,\phi_k))\,\frac{W(\theta_j,\phi_k)}{\sin\theta_j}, \tag{13}$$

*where*

$$w_j^s = 8\pi\, w_j \sum_{n=0}^{p} \sin(\theta_j/2)\, P_n(\cos\theta_j)\,, \tag{14}$$

*is superalgebraically convergent with $p$.*

For an arbitrary target $\mathbf{x}\in\Gamma$, we can simply rotate the coordinate system so that $\mathbf{x}$ becomes a north pole and use Theorem 1 to evaluate $\mathcal{S}[f](\mathbf{x})$. Both the function $f$ and the coordinate functions $\mathbf{y}$ need to be interpolated to the new spherical grid locations in the rotated coordinate system. Since there are $(p+1)(2p+2)=\mathcal{O}(p^2)$ points in the spherical grid (2), a naïve interpolation would be $\mathcal{O}(p^4)$. In practical applications, the layer potentials need to be evaluated at each of the spherical grid points, giving rise to Problem 1 and requiring $\mathcal{O}(p^6)$ work. This cost can be reduced to $\mathcal{O}(p^5)$ by using properties of spherical harmonic representations, which we discuss next.

## 2.2 Rotation of Spherical Harmonic Expansions

Suppose now that the pole of the spherical harmonic expansion needs to be rotated into the spherical grid location $(\beta,\alpha)$. In the rotated system, the function $f$ defined by (5) can be expressed as

$$f(\theta',\phi') = \sum_{n=0}^{p}\sum_{m'=-n}^{n} f_n^{m'}(\alpha,\beta,\gamma)\, Y_n^{m'}(\theta',\phi'), \tag{15}$$

where $(\theta',\phi')$ denote the coordinates of the point $(\theta,\phi)$ in the rotated system,

$$f_n^{m'}(\alpha,\beta,\gamma) = \sum_{m=-n}^{n} D_n^{m',m}(\alpha,\beta,\gamma)\cdot f_n^m, \tag{16}$$

and the standard Euler angles $(\alpha,\beta,\gamma)$ define the rotation using the $z$-$y$-$z$ convention in a right-handed frame [2, 17]. The coefficients of the transformation (due to Wigner [24]) are given by

$$D_n^{m',m}(\alpha,\beta,\gamma) = e^{im\gamma}\, d_n^{m',m}(\beta)\, e^{im\alpha}, \tag{17}$$

where

$$d_n^{m',m}(\beta) = (-1)^{m'-m}[(n+m')!\,(n-m')!\,(n+m)!\,(n-m)!]^{1/2} \sum_s (-1)^s \frac{\left(\cos\frac{\beta}{2}\right)^{2(n-s)+m-m'} \left(\sin\frac{\beta}{2}\right)^{2s-m+m'}}{(n+m-s)!s!(m'-m+s)!(n-m'-s)!}, \tag{18}$$

with the range of $s$ determined by the condition that all factorials are non-negative.

In our context, we can fix $\gamma = 0$ without loss of generality because it does not affect the layer potential evaluation. The Euler angles corresponding to spherical grid locations $(\theta_j, \phi_k)$ defined in (2) are then given by $(\phi_k, \theta_j, 0)$, and the coefficients of the rotated spherical harmonic expansion $f_n^{m'}(\phi_k, \theta_j, 0)$ can be expressed as

$$f_n^{m'}(\phi_k, \theta_j, 0) = \sum_{m=-n}^{n} f_n^m \cdot d_n^{m',m}(\theta_j) \cdot e^{im\phi_k}. \tag{19}$$

For a fixed location $(\theta_j, \phi_k)$, the angle $\phi_k$ rotation about the $z$-axis in (17) is diagonal and requires only $\mathcal{O}(p^2)$ work. The angle $\theta_j$ rotation about the $y$-axis, however, requires $\mathcal{O}(p^3)$ work even if Wigner coefficients $\{d_n^{m',m}(\theta_j)\}$ were known. Since there are $\mathcal{O}(p^2)$ spherical grid locations, the total work to evaluate the coefficients of the rotated spherical harmonic expansions is $\mathcal{O}(p^5)$.

For a constant latitude $\theta_j$, the spherical grids are sampled uniformly. The evaluation of sums in (19) can be accelerated via the discrete FFT, since (19) becomes a discrete Fourier sum:

$$f_n^{m'}(\phi_k, \theta_j, 0) = \sum_{m=-n}^{n} f_n^m \cdot d_n^{m',m}(\theta_j) \cdot e^{i2\pi km/(2p+2)} \quad \text{for} \quad k = 0, \ldots, 2p+1. \tag{20}$$

The cost of evaluating sums (20) for one fixed latitude is $\mathcal{O}(p^3 \log p)$, and the total work for rotating a spherical harmonic expansion to all new pole locations is $\mathcal{O}(p^4 \log p)$.

Once the coefficients of the rotated spherical harmonic expansions $f_n^{m'}(\phi_k, \theta_j, 0)$ are constructed, we can compute the function values on rotated grids via the Spherical Harmonic Transform (SHT) [18]:

$$f(\theta_{j'k';jk}, \phi_{j'k';jk}) = \sum_{n=0}^{p} \sum_{m'=-n}^{n} f_n^{m'}(\phi_k, \theta_j, 0)\, P_n^{m'}(\theta_{j'}) \cdot e^{im'\phi_{k'}} \tag{21}$$

$$= \sum_{m'=-p}^{p} \left[ \sum_{n=|m'|}^{p} f_n^{m'}(\phi_k, \theta_j, 0)\, P_n^{m'}(\theta_{j'}) \right] \cdot e^{im'\phi_{k'}}, \tag{22}$$

where $(\theta_{j'k';jk}, \phi_{j'k';jk})$ denotes the coordinates of the rotated spherical grid $(\theta_{j'}, \phi_{k'})$ with the pole location $(\theta_j, \phi_k)$, where

$$\theta_{j'k';jk} = \cos^{-1} z_{j'k';jk}, \quad \phi_{j'k';jk} = \tan^{-1} \frac{y_{j'k';jk}}{x_{j'k';jk}}, \tag{23}$$

and $x_{j'k';jk}$, $y_{j'k';jk}$, $z_{j'k';jk}$ are the Cartesian coordinates of the rotated spherical grids:

$$\begin{pmatrix} x_{j'k';jk} \\ y_{j'k';jk} \\ z_{j'k';jk} \end{pmatrix} = \begin{pmatrix} \cos\phi_k & -\sin\phi_k & 0 \\ \sin\phi_k & \cos\phi_k & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_j & 0 & -\sin\theta_j \\ 0 & 1 & 0 \\ \sin\theta_j & 0 & \cos\theta_j \end{pmatrix} \begin{pmatrix} \cos\phi_{k'}\sin\theta_{j'} \\ \sin\phi_{k'}\sin\theta_{j'} \\ \cos\theta_{j'} \end{pmatrix}. \tag{24}$$

The evaluation of (22) can be decomposed into the associate Legendre transform

$$g_{jk}^{m'}(\theta_{j'}) = \sum_{n=|m'|}^{p} f_n^{m'}(\phi_k, \theta_j, 0)\, P_n^{m'}(\theta_{j'}), \tag{25}$$

and the Fourier transform

$$f(\theta_{j'k';jk}, \phi_{j'k';jk}) = \sum_{m'=-p}^{p} g_{jk}^{m'}(\theta_{j'}) \cdot e^{im'\phi_{k'}}. \tag{26}$$

If the direct associate Legendre transform is used to evaluate (25), then the cost of evaluating the spherical transform is $\mathcal{O}(p^3)$ for one pole location and the total work of evaluating function values at all the rotated grids is $\mathcal{O}(p^5)$.

We introduce the following acronyms to facilitate further discussion:

- The direct rotation algorithm (19), followed by the direct SHT (22) will be referred to as the *direct spherical grid rotation* (**DSR**) algorithm (see also [6], [7]).
- The FFT-accelerated rotation algorithm (20), followed by the direct SHT (22) will be referred to as the *FFT-accelerated spherical grid rotation* (**FSR**) algorithm (see [20]).

The computational complexity of both algorithms is $\mathcal{O}(p^5)$. The cost of DSR is dominated by spherical harmonic expansion rotations, while the cost of FSR is dominated by the spherical transforms. It is worth mentioning, that the fast spherical transform algorithms [18, 21, 22] would lead to better asymptotic complexity of either $\mathcal{O}(p^4 \log^2 p)$ by using the Driscoll-Healy type algorithms [10, 13], or $\mathcal{O}(p^4 \log p)$ by using the Fast Multipole Method type schemes [19, 21, 22, 18] to perform the associated Legendre transforms. Since the fast spherical transform algorithms as a rule break-even with the direct SHT scheme only for very large values of $p > 100$ (see the review paper [4] for details), we have not incorporated these accelerations into our codes.

# 3 Rotation of Spherical Grids via nonuniform FFTs

In this section, we introduce a new algorithm for spherical grid rotations that avoids performing the spherical transforms and has a nearly optimal asymptotic complexity of $\mathcal{O}(p^4 \log p)$. We start by observing that the problem of evaluating values of $f$ on the rotated grids can be viewed as an interpolation problem on the sphere. In order to perform interpolation efficiently, $f$ can be expanded into a Fourier series in the extended parametric domain $\theta \in [0, 2\pi]$, $\phi \in [0, 2\pi]$, and evaluating values of the user-defined function on all grids $(\theta_{j'k';jk}, \phi_{j'k';jk})$ can then be done via the nonuniform FFT, [3, 8, 11].

## 3.1 Hybrid nonuniform FFT (hnuFFT)

Suppose now that we have rotated the pole of the spherical grid into $2p+2$ new constant latitude locations $(\theta_j, \phi_k)$, where $\theta_j$ is fixed and $\phi_k = 2\pi k/(2p+2)$ for $k = 0, \ldots, 2p+1$. It is easy to see from (23) and (24) that the $k$-th rotated grid can be obtained by rotating the grid corresponding to the pole $(\theta_j, \phi_0)$ about the z-axis by angle $\phi_k$, and that the coordinates are

$$\theta_{j'k';jk} = \theta_{j'k';j0} \quad \text{and} \quad \phi_{j'k';jk} = \phi_{j'k';j0} + \phi_k. \tag{27}$$

Note, that the rotated grids can be aligned on great circles by setting the latitude dependent phases $\phi_{j'k';j0}$ to zero (see Figure 1), effectively decoupling interpolation directions. This suggests a hybrid interpolation algorithm which can be informally described by using a simple geometrical interpretation. First, the function is interpolated on the great circles to a large number of points located on constant latitudes $\theta_{j'k';j0}$ via one dimensional nonuniform FFTs, leading to a much smaller associated complexity constant. Then, the interpolated points on great circles are shifted by latitude dependent phases $\phi_{j'k';j0}$ back to the final rotated grid locations.

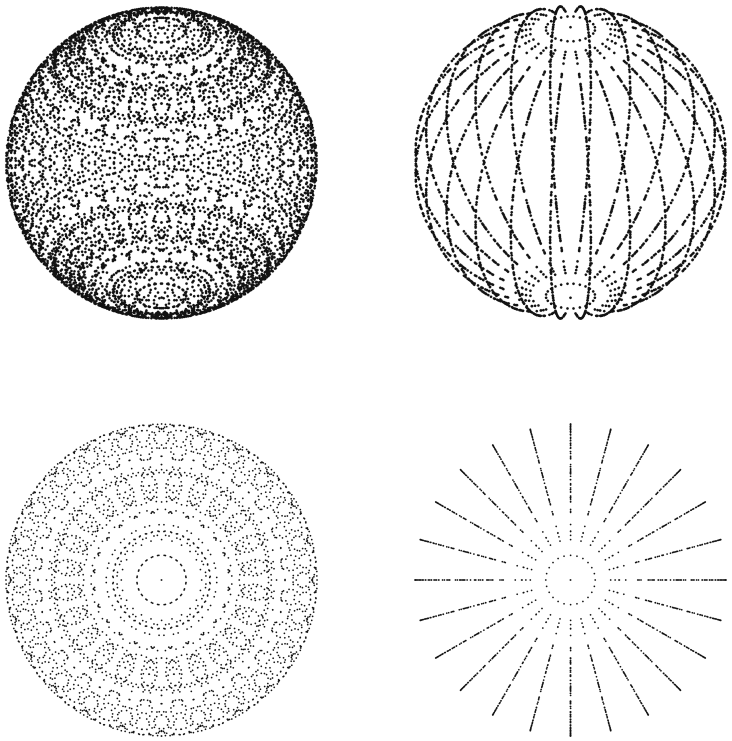


Figure 1: The two panels on the left depict the discrete points belonging to the union of $2p + 2$ spherical grids for a fixed latitude (side and top view), $p = 12$. By setting the latitude dependent phases $\phi_{j'k';j0}$ to zero, these seemingly random points can be aligned on the great circles, as depicted on the right panels.

**Algorithm: Rotation of spherical grids via hybrid nonuniform FFT (hnuFFT)**

Given the $p$-th order spherical harmonic representation of a function $f$ described by the spherical harmonic expansion coefficients $f_n^m$

$$f(\theta, \phi) = \sum_{n=0}^{p} \sum_{m=-n}^{n} f_n^m \, Y_n^m(\theta, \phi), \tag{28}$$

and a spherical grid $(\theta_j, \phi_k)$, evaluate the values of $f(\theta_{j'k';jk}, \phi_{j'k';jk})$ on all auxiliary rotated grids

whose north pole locations coincide with the original spherical grid points, where $(\theta_{j'k';jk}, \phi_{j'k';jk})$ denotes the coordinates of the rotated spherical grid $(\theta_{j'}, \phi_{k'})$ with the pole location $(\theta_j, \phi_k)$.

1. *Evaluate the Fourier coefficients:*

   (a) Form the auxiliary doubly periodic spherical Chebyshev grid $(\theta_j^D, \phi_k^D)$ with $2p+2$ equispaced nodes in the $\phi$-direction $\{\phi_k^D = \frac{2\pi k}{2p+2}\}_{k=0}^{2p+1}$ and $2p+2$ equispaced nodes in the $\theta$-direction $\{\theta_j^D = \frac{(2j+1)\pi}{2p+2}\}_{j=0}^{2p+1}$.

   (b) Extend $f$ into the doubly periodic parametric domain $\theta \in [0, 2\pi], \phi \in [0, 2\pi]$ by using the relation for smooth functions on great circles

$$f^P(\theta, \phi) = \begin{cases} f(\theta, \phi), & \text{if } 0 \le \theta \le \pi, \\ f(2\pi - \theta, \phi + \pi), & \text{if } \pi \le \theta \le 2\pi, \end{cases} \tag{29}$$

   and evaluate the values $f^P(\theta_j^D, \phi_k^D)$ from the spherical harmonic expansion coefficients.

   (c) Compute the Fourier coefficients $\hat{f}_{nm}$ in the original coordinate system:

$$\hat{f}_{nm} = \sum_{j=0}^{2p+1} \sum_{k=0}^{2p+1} f^P(\theta_j^D, \phi_k^D) \cdot e^{-in\theta_j^D} \cdot e^{-im\phi_k^D}, \tag{30}$$

   for $n = 0, \ldots, 2p+1$, $m = 0, \ldots, 2p+1$.

2. *Evaluate the sums:*

$$f(\theta_{j'k';jk}, \phi_{j'k';jk}) = \frac{1}{(2p+2)^2} \sum_{n=0}^{2p+1} \sum_{m=0}^{2p+1} \hat{f}_{nm} \cdot e^{in\theta_{j'k';j0}} \cdot e^{im\phi_{j'k';j0}} \cdot e^{im\phi_k}. \tag{31}$$

   (a) Apply the nonuniform adjoint FFT with respect to $\theta$:

$$g_m(\theta_{j'k';j0}) = \frac{1}{2p+2} \sum_{n=0}^{2p+1} \hat{f}_{nm} \cdot e^{in\theta_{j'k';j0}}. \tag{32}$$

   (b) Apply the regular inverse FFT with respect to $\phi$:

$$f(\theta_{j'k';jk}, \phi_{j'k';jk}) = \frac{1}{2p+2} \sum_{m=0}^{2p+1} g_m(\theta_{j'k';j0}) \cdot e^{im\phi_{j'k';j0}} \cdot e^{im\phi_k}. \tag{33}$$

---

The arithmetic complexity of this algorithm is $\mathcal{O}(p^2 \log p + M_{\text{total}} + p^4 \log p)$, where the number of points $M_{\text{total}} = \mathcal{O}(p^4)$, therefore, the total work is $\mathcal{O}(p^4 \log p)$ for all rotated spherical grids. For one fixed latitude $\theta_j$, the computational cost is $\mathcal{O}(p^3 \log p)$ while using $\mathcal{O}(p^3)$ intermediate storage.

In our implementation, we have also incorporated the following optimizations:

1. For real-valued functions, the Fourier coefficients have complex conjugate symmetries. Therefore, only half of the coefficients $g_m(\theta_{j'k';j0})$ need to be evaluated during Step 2a for each fixed polar angle location $\theta_{j'k';j0}$, and the complex-valued FFT can be replaced with the real-valued FFT during Step 2b.

2. For better regular FFT performance, the number of angular discretization nodes on the spherical grid can be set to the next nearest integer $N_p \geq 2p+2$ that is a multiple of either 2, 3, or 5. The second part of the algorithm can then be trivially adjusted to handle the modified Gaussian spherical grids

$$\left\{\phi_k = \frac{2\pi k}{N_p},\, k = 0, \ldots, N_p - 1\right\} \quad \text{and} \quad \left\{\theta_j = \cos^{-1}(t_j),\, j = 0, \ldots p\right\}, \tag{34}$$

where $t_j$'s are the $(p+1)$-point Gauss-Legendre quadrature nodes, by applying the regular inverse FFT with respect to $\phi$ to the oversampled angular grid

$$f(\theta_{j'k';jk}, \phi_{j'k';jk}) = \frac{1}{N_p} \sum_{m=0}^{N_p-1} g_m(\theta_{j'k';j0}) \cdot e^{im\phi_{j'k';j0}} \cdot e^{im\phi_k}. \tag{35}$$

3. Finally, we note that nearly half of the following values ($z$-components of the rotated spherical grid),

$$z_{j'k';j0} = \sin\theta_j \cos\phi_{k'} \sin\theta_{j'} + \cos\theta_j \cos\theta_{j'}, \tag{36}$$

are duplicate because of the symmetry in $\cos\phi_{k'}$ when evaluated on the regular spherical grid. Therefore, the sums $g_m(\theta_{j'k';j0})$ in (32) need to be evaluated at half of the polar angle locations only.

# 4 Numerical Results and Discussion

In this section, we analyze the performance of all the spherical grid rotation algorithms discussed so far and an application of the quadrature rule (13) to a particular problem. All tests were performed on a laptop with an Intel Core i7 2.2GHz processor and 32GB RAM using double precision arithmetic.

Tables 1–2 report the results of applying the algorithms to real-valued functions $f$ defined in the parametric domain $\theta \in [0, \pi]$, $\phi \in [0, 2\pi]$ by the formula

$$f(\theta, \phi) = \sum_{n=0}^{p} \sum_{m=-n}^{n} c_{n,m}\, P_n^m(\cos\theta) \cdot e^{im\phi}, \tag{37}$$

where $p$ is the degree of the spherical harmonic expansion, and $c_{n,m}$ are randomly generated complex numbers such that their real and imaginary parts are in the interval $[-1, 1]$, and $c_{n,-m} = \overline{c_{n,m}}$, for $m > 0$.

**Comparing the performance of the rotation algorithms.** In Table 1, we show the setup parameters and precomputation times. The second and third columns contain the number of angular discretization nodes and the total number of grid points used in the tests, respectively.

We used the nonuniform fast spherical transform scheme (available as a part of the publicly available `NFFT3` library [11], version 3.2.3) to test the nonuniform FFT interpolation in two dimensions algorithm. The default library flags were used that correspond to the Kaiser-Bessel window for the nonuniform FFT in two dimensions with the spreading constant set to 6 and the oversampling parameter $\sigma = 2$, yielding approximately 10 digits of accuracy, with either tensor product based precomputation (NFSFT1) or the memory-efficient linear table lookup scheme (NFSFT2).

The spreading window function values for all target grid locations on the sphere were precomputed and stored in a tensor product lookup table to accelerate the subsequent calls to the NFSFT1 scheme. For the FSR and DSR schemes, we have precomputed and stored the elements of Wigner rotation matrices for all latitudes of the original spherical grid. The fourth and fifth columns contain the times to precompute the required quantities for the NFSFT1 and the DSR/FSR schemes, respectively. No attempt has been made to accelerate the above precomputation steps. We do not list the precomputation times for the NFSFT2 scheme, since the corresponding linear lookup table depends only on a preset precision parameter and can be used for arbitrary grid configurations.

From this table, it is clear that the two-dimensional interpolation scheme NFSFT1 has large precomputation and storage costs, proportional to $\mathcal{O}(p^4)$. In fact, for values of $p$ greater that 60, the NFFT3 library failed to allocate memory required for the storage tables. For such cases, we only list an estimate by calling the initialization routine for a set of grids corresponding to each constant latitude separately. The precomputation and storage costs for the DSR and FSR schemes are also proportional to $\mathcal{O}(p^4)$ but with a much smaller constant.

In order to eliminate the precomputation and storage cost for the hnuFFT scheme, we used the dilated Gaussian window with the spreading constant set to 12 with the oversampling parameter $\sigma = 2$ to yield 12 digits of accuracy. The hnuFFT scheme was subsequently used without any precomputation and, therefore, the initialization timings are omitted from Table 1.

In Table 2, we compare the performance of the new algorithm with the reference algorithms. In the third and forth columns, we report the timings for rotations using the NFSFT1 and NFSFT2 schemes, respectively. The timings were performed using the precomputed tables. For values of $p$ greater that 60, the tensor product lookup table was not available for the NFSFT1 scheme due to memory limitation reasons. For such values, we only give a timing estimate by calling the routine for a set of grids corresponding constant latitudes separately and using partially precomputed lookup tables. The NFSFT2 scheme is slightly slower due to an additional cost during the linear interpolation step. While the asymptotic behavior is nearly-optimal, the CPU timings for both schemes are higher compared to other algorithms because of the relatively large complexity constants.

Despite the higher asymptotic complexity, the DSR algorithm performs significantly better than 2D interpolation for all values $p$ reported here. The FSR algorithm naturally is faster than the DSR scheme since it accelerates one of the intermediate steps. In fact, it outperforms all the other algorithms for small values of $p$. For $p > 36$, the hybrid nonuniform FFT (hnuFFT) scheme has the lowest CPU timings.

For single precision calculations, it is possible to reduce the NFSFT scheme timings by approximately a factor of 2 by setting the Kaiser-Bessel window spreading constant to 4 (for a 7 digit accurate scheme), or by approximately a factor of 3 by setting the spreading constant to 3 (for a 5 digit accurate scheme). The savings are still not sufficient for the NFSFT to be competitive with the new scheme.

**Layer potential computation.** Next, we investigate the performance of the new rotation algorithm in computing the single layer Stokes potential and report the results in Figure 2. Consider the surface $\Gamma$ shown in Figure 2 whose coordinate functions $\mathbf{x}(\theta, \phi)$ are given by

| p | $N_p$ | $M$ | NFSFT1 | DSR/FSR |
|---|---|---|---|---|
| 12 | 30 | 390 | 0.35 | 0.0031 |
| 24 | 50 | 1250 | 3.58 | 0.040 |
| 36 | 80 | 2960 | 20.46 | 0.21 |
| 48 | 100 | 4900 | 56.08 | 0.73 |
| 60 | 128 | 7808 | 142.4 | 2.29 |
| 72 | 150 | 10950 | (280.1) | 4.05 |
| 84 | 180 | 15300 | (547.0) | 7.16 |
| 96 | 200 | 19400 | (879.5) | 11.94 |
| 108 | 240 | 26160 | (1599) | 25.03 |

Table 1: *Precomputation times (in seconds) for the grid rotation algorithms as a function of the order of spherical harmonic approximation p. $N_p$ is the number of discretization nodes in $\phi$ direction. $M = (p+1)N_p$ is the total number of discretization points. The figures in parentheses are estimates used when the memory required by the NFSFT1 scheme is excessive.*

$$\mathbf{x}(\theta,\phi) = \begin{bmatrix} \sin\theta\cos\phi + \frac{3}{10}\sin\left(\frac{9\pi}{4}\cos\theta\right) \\ \sin\theta\sin\phi + \frac{1}{2}\cos\left(\frac{9\pi}{4}\cos\theta\right) \\ \cos\theta \end{bmatrix}, \quad \theta\in[0,\pi], \quad \phi\in[0,2\pi]. \tag{38}$$

The single layer Stokes potential $\mathcal{S}[\mathbf{f}](\mathbf{x})$ gives the velocity field induced by an interfacial force $\mathbf{f}(\mathbf{x})$ when the exterior and the interior of the interface $\Gamma$ is filled by same Stokesian fluid [15]. Let $H$ be the mean curvature of $\Gamma$ and $\mathbf{n}$ be its unit normal. We set $\mathbf{f}(\mathbf{x}) = H(\mathbf{x})\mathbf{n}(\mathbf{x})$ which corresponds to the interfacial force on a bubble with unit surface tension. Our goal is to evaluate $\mathcal{S}[\mathbf{f}](\mathbf{x})$ on the interface at all the $M$ spherical grid locations (2). When the evaluation point is a north pole, we can simply use the quadrature rule of Theorem 1 and write [23]:

$$\mathcal{S}[\mathbf{f}](\mathbf{x}(0,0)) = \sum_{j=0}^{p}\sum_{k=0}^{N_p-1} w_j^s G(\mathbf{x}(0,0),\mathbf{y}(\theta_j,\phi_k)))\mathbf{f}(\mathbf{y}(\theta_j,\phi_k))\frac{W(\theta_j,\phi_k)}{\sin\theta_j}, \tag{39}$$

where

$$G(\mathbf{x},\mathbf{y}) = \frac{1}{8\pi}\left[\frac{1}{\|\mathbf{x}-\mathbf{y}\|}I + \frac{(\mathbf{x}-\mathbf{y})\otimes(\mathbf{x}-\mathbf{y})}{\|\mathbf{x}-\mathbf{y}\|^3}\right] \tag{40}$$

is the free space Green's function for the Stokes equations. For other target locations, similar to the Laplace potential, we evaluate $\mathcal{S}[\mathbf{f}]$ by rotating the coordinate system so that the target becomes a north pole in the new coordinate system and then applying (39). A total of six functions need to be rotated to the new coordinate system – the three components of $\mathbf{y}$ and the three components of $\mathbf{f}(\mathbf{y})W/\sin\theta$. Therefore, a total of $6M$ rotations have to be calculated.

From Figure 2, it is clear that the spherical grid rotations dominate the total cost of the single layer potential computation, and the CPU timings scale proportionally to the cost of the hnuFFT rotation scheme, namely, as $\mathcal{O}(p^4\log p)$. In Figure 3, we report the relative errors in computing $\mathcal{S}[\mathbf{f}](\mathbf{x})$ corresponding to two geometries that arise in physical applications. The reference values have been generated using finer discretizations.

| $p$ | $M$ | NFSFT1 $\mathcal{O}(p^4)$ | NFSFT2 $\mathcal{O}(p^4)$ | DSR $\mathcal{O}(p^5)$ | FSR $\mathcal{O}(p^5)$ | hnuFFT $\mathcal{O}(p^4 \log p)$ |
|---|---|---|---|---|---|---|
| 12 | 390 | 0.046 | 0.13 | 0.0025 | 0.0023 | 0.0043 |
| 24 | 1250 | 0.45 | 1.34 | 0.037 | 0.030 | 0.036 |
| 36 | 2960 | 2.69 | 7.41 | 0.27 | 0.20 | 0.20 |
| 48 | 4900 | 7.19 | 19.9 | 0.99 | 0.66 | 0.58 |
| 60 | 7808 | 17.8 | 44.8 | 3.08 | 2.04 | 1.80 |
| 72 | 10950 | (35.5) | 88.7 | 7.75 | 4.62 | 3.53 |
| 84 | 15300 | (69.1) | 207.1 | 21.45 | 9.88 | 6.03 |
| 96 | 19400 | (115.9) | 278.5 | 46.31 | 18.47 | 10.11 |
| 108 | 26160 | (207.8) | 513.5 | 100.03 | 42.01 | 19.95 |

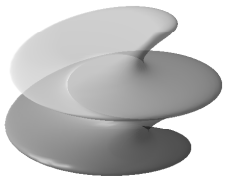
Table 2: *CPU times (in seconds) for the grid rotation algorithms as a function of the order of spherical harmonic approximation $p$. $M$ is the number of discretization points. The precision was set to yield 12 digits for the hnuFFT scheme. The figures in parentheses are estimates used when the memory required by the NFSFT1 scheme is excessive.*

# 5 Conclusion

We introduced a new algorithm for accelerating the computation of singular integrals on surfaces that are globally parametrized by spherical coordinates. The complexity of the new algorithm is $\mathcal{O}(p^4 \log p)$, with a small complexity constant and $\mathcal{O}(p^3)$ intermediate storage requirements. It is worth mentioning, that while the FFT acceleration step in the FSR scheme has been widely used in the context of molecular replacement simulations [20], we are not aware of its use in the context of singular quadrature. For small values of $p$, the FSR scheme can be used as an alternative to our hybrid algorithm. Currently, we are investigating extensions to the spherical grid based quadrature rules to enable hyper-singular integral evaluations.

# 6 Acknowledgments

We would like to thank the anonymous referees for their comments and valuable suggestions that substantially improved this paper.



| $p$ | $T_{rotation}$ | $T_{kernel}$ | $T_{total}$ | $E_2$ |
|---|---|---|---|---|
| 12 | 0.0312 | 0.0015 | 0.0327 | $8.4 \times 10^{-3}$ |
| 24 | 0.238 | 0.014 | 0.252 | $6.2 \times 10^{-4}$ |
| 36 | 1.30 | 0.08 | 1.38 | $1.6 \times 10^{-5}$ |
| 48 | 3.54 | 0.27 | 3.81 | $4.0 \times 10^{-7}$ |
| 60 | 10.87 | 0.59 | 11.46 | $3.9 \times 10^{-8}$ |
| 72 | 21.31 | 1.19 | 22.50 | $3.2 \times 10^{-9}$ |
| 84 | 36.38 | 2.25 | 38.63 | $3.3 \times 10^{-10}$ |
| 96 | 60.88 | 3.81 | 64.69 | $4.5 \times 10^{-12}$ |
| 108 | 120.64 | 6.25 | 126.89 | $1.5 \times 10^{-12}$ |

Figure 2: *CPU times (in seconds) for computing the Stokes single layer potential $T_{total}$, the time spent in spherical grid rotations $T_{rotation}$ via the hnuFFT scheme, the time for the kernel evaluations $T_{kernel}$, and the relative errors ($E_2$). For every evaluation point on the boundary, six rotations are required corresponding to three coordinate functions and three components of the interfacial force. Observe that, for every $p$, $T_{rotation}$ makes up a significant fraction (more than 90%) of the total CPU time. Hence, even a modest acceleration in performing spherical grid rotations will translate to faster Stokes flow simulations.*

## References


[1] K. E. Atkinson, *The Numerical Solution of Integral Equations of the Second Kind*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 1997.

[2] L. C. Biedenharn and J. D. Louck, *Angular Momentum in Quantum Physics: Theory and Application*, Addison-Wesley, Reading, Mass., 1981.

[3] A. Dutt and V. Rokhlin, *Fast Fourier Transforms for Nonequispaced Data*, SIAM J. Sci. Comput., 14 (1993), pp. 1368–1393.

[4] J. B. Drake, P. Worley and E. D'Azevedo, *Spherical Harmonic Transform Algorithms*, ACM Trans. Math. Softw., 35 (2008).

[5] J. B. Freund and M. M. Orescanin, *Cellular flow in a small blood vessel*, J. Fluid Mech., 671 (2011), pp. 466–490.

[6] M. Ganesh and I. G. Graham, *A High-Order Algorithm for Obstacle Scattering in Three Dimensions*, J. Comput. Phys., 198 (2004), pp. 211–242.

[7] I. G. Graham and I. H. Sloan, *Fully Discrete Spectral Boundary Integral Methods for Helmholtz Problems on Smooth Closed Surfaces in $R^3$*, Numer. Math., 92 (2002), pp. 289–323.

[8] L. Greengard and J.-Y. Lee, *Accelerating the Nonuniform Fast Fourier Transform*, SIAM Rev., 46 (2004), pp. 443–454.

[9] L. Greengard and V. Rokhlin, *A Fast Algorithm for Particle Simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.

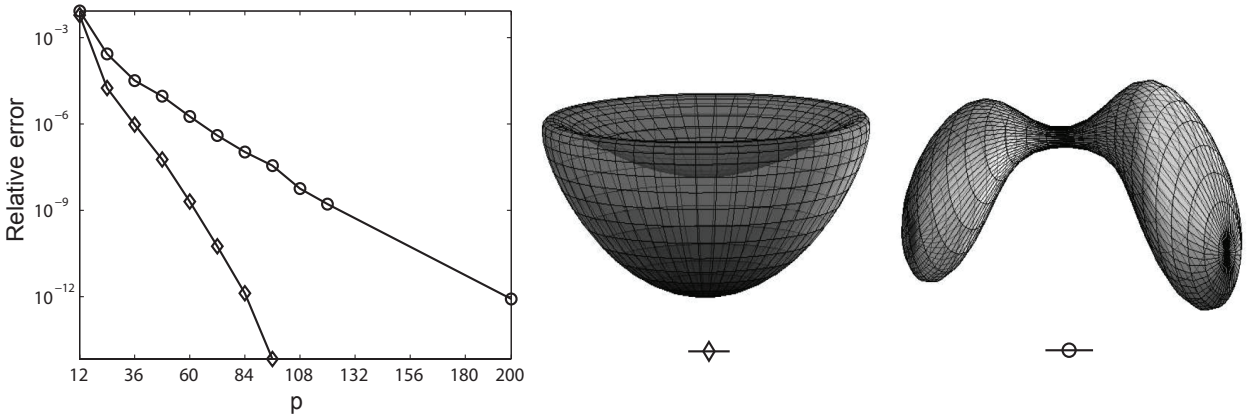



Figure 3: Relative errors in computing the Stokes single layer potential on two geometries shown on the right. The first parachute-shaped boundary is typically observed when a bubble rises in gravity. The neck formation in the second shape is typical of a soft-particle such as a capsule or red blood cell squeezing through a constricted vessel. In both cases, the errors decay superalgebraically.


[10] D. M. Healy Jr., D. N. Rockmore, P. J. Kostelec, and S. Moore, *FFTs for the 2-sphere: Improvements and Variations*, J. Fourier Anal. Appl., 9 (2003), pp. 341-385.

[11] J. Keiner, S. Kunis and D. Potts, *Using NFFT 3 - a Software Library for Various Nonequispaced Fast Fourier Transforms*, ACM Trans. Math. Software, 36 (2009), pp. 1-30.

[12] J. Keiner and D. Potts, *Fast Evaluation of Quadrature Formulae on the Sphere*, Math. Comput., 77 (2008), pp. 397–419.

[13] S. Kunis and D. Potts, *Fast Spherical Fourier Algorithm*, Math. Comput., 161 (2003), pp. 75–98.

[14] S. Kunis and D. Potts, *Time and Memory Requirements of the Nonequispaced FFT*, Sampl. Theory Signal Image Process. 7 (2008), pp. 77–100.

[15] C. Pozrikidis, *Interfacial Dynamics for Stokes Flow*, J. Comput. Phys., 169 (2001), pp. 250–301.

[16] A. Rahimian, I. Lashuk, S. Veerapaneni, A. Chandramowlishwaran, D. Malhotra, L. Moon, R. Sampath, A. Shringarpure, J. Vetter, R. Vuduc, D. Zorin, and G. Biros. *Petascale Direct Numerical Simulation of Blood Flow on 200k Cores and Heterogeneous Architectures*, In Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, 2010, pp. 1–11.

[17] M. E. Rose, *Elementary Theory of Angular Momentum*, John Wiley & Sons, Inc., 1957.

[18] D. S. Seljebotn, *Wavemoth-Fast Spherical Harmonic Transforms by Butterfly Matrix Compression*, Astrophys. J. Suppl. Ser., 199 (2012), pp. 1–5.

[19] R. Suda and M. Takami, *A fast spherical harmonic transform algorithm*, Math. Comp. 7 (2002), pp. 703-715.

[20] S. Trapani and J. Navaza, *AMoRe: Classical and Modern*, Acta Crystallogr., D64 (2008), pp. 11–16.

[21] M. Tygert, *Fast Algorithms for Spherical Harmonic Expansions, II*, J. Comput. Phys., 227 (2008), pp. 4260-4279.

[22] M. Tygert, *Fast Algorithms for Spherical Harmonic Expansions, III*, J. Comput. Phys., 229 (2010), pp. 6181–6192.

[23] S. K. Veerapaneni, A. Rahimian, G. Biros, and D. Zorin, *A Fast Algorithm for Simulating Vesicle Flows in Three Dimensions*, J. Comput. Phys., 230 (2011), pp. 5610–5634.

[24] E. P. Wigner, *Gruppentheorie und ihre Anwendungen auf die Quantenmechanik der Atomspektren*, Vieweg Verlag, Braunschweig, 1931.

[25] H. Zhao, A. H. G. Isfahani, L. N. Olson, and J. B. Freund, *A Spectral Boundary Integral Method for Flowing Blood Cells*, J. Comput. Phys., 229 (2010), pp. 3726–3744.