# On the computation of nonnegative quadrature weights on the sphere

Manuel Gräf Stefan Kunis Daniel Potts

We compute quadrature weights for scattered nodes on the two-dimensional unit-sphere, which are exact for spherical polynomials of high degree $N$. Different algorithms are proposed and numerical examples show that we can compute nonnegative quadrature weights if approximately $4N^2/3$ well distributed nodes are used. We compare these results with theoretical statements which guarantee nonnegative quadrature weights. The proposed algorithms are based on fast spherical Fourier algorithms for arbitrary nodes which are publicly available. Numerical experiments are presented to demonstrate that we are able to compute quadrature weights for circa 1.5 million nodes which are exact for spherical polynomials up to $N = 1024$.

*Key words and phrases* : two-sphere, quadrature, nonequispaced fast spherical Fourier transform, NFFT, FFT

*2000 AMS Mathematics Subject Classification* : 65T99, 33C55, 42C10, 65T50

## 1 Introduction

Harmonic analysis on the sphere typically consists in the expansion of functions $f \in \mathrm{L}^2(\mathbb{S}^2)$ with respect to the orthonormal basis of spherical harmonics $Y_k^n$, $n \in \mathbb{N}_0$, $k = -n, \ldots, n$. The computation of the Fourier coefficients

$$a_k^n = \int_{\mathbb{S}^2} f(\boldsymbol{x}) \overline{Y_k^n(\boldsymbol{x})} \mathrm{d}\boldsymbol{x}, \tag{1.1}$$

can be approximated up to some finite degree $n \leq N \in \mathbb{N}_0$ by a quadrature rule

$$\tilde{a}_k^n = \sum_{i=1}^{M} w_i f(\boldsymbol{x}_i) \overline{Y_k^n(\boldsymbol{x}_i)} \tag{1.2}$$

Chemnitz University of Technology, Department of Mathematics, 09107 Chemnitz, Germany, {m.graef,kunis,potts}@mathematik.tu-chemnitz.de

with sampling nodes $\boldsymbol{x}_i \in \mathbb{S}^2$ and quadrature weights $w_i$, $i = 0, \ldots, M-1$. The corresponding synthesis computes function values from given expansion coefficients, i.e.,

$$f(\boldsymbol{x}_i) = \sum_{k=0}^{N} \sum_{n=-k}^{k} a_k^n Y_k^n(\boldsymbol{x}_i) \tag{1.3}$$

For both transforms (1.2) and (1.3) fast approximate algorithms of complexity $\mathcal{O}(N^2 \log^2 N + M)$ where suggested in [10] and [11], respectively. An implementation is public available from the NFFT homepage [8]. The aim of this paper is to compute nonnegative quadrature weights $w_i$, for arbitrary sampling nodes such that we can compute the spherical Fourier coefficients of $f$ in (1.3) by the quadrature (1.2). We propose algorithms based on iterative solvers making use of fast matrix vector multiplication.

Up to now, the following theoretical results are known. As a consequence of spherical Marcinkiewicz-Zygmund inequalities [13, 2, 9] a sufficient condition for the existence of nonnegative quadrature weights is proven in [13]. Furthermore, with the constants established in [9] we can guarantee nonnegative quadrature weights for $M$ best arranged sampling nodes and every polynomial degree $N < \sqrt{M}/1530$. These theoretical results are far from being optimal, and [12] suggests an orthonormalization procedure for the computation of quadrature weights to improve upon these results. In this note, we suggest simple and fast iterative methods to compute nonnegative quadrature weights for $M$ well distributed quadrature nodes and polynomial degree $N \approx \sqrt{3M/4}$.

The outline of this paper is as follows: In Section 2 we state an optimization problem to compute nonnegative quadrature weights for a given sampling set and polynomial degree. Lemma 2.1 gives a sufficient condition on the polynomial degree $N$ for the existence of nonnegative quadrature. Subsequently, we present three algorithms in Section 3 for solving the mentioned optimization problem and test these on several examples in Section 4.

## 2 Prerequisites

Let $\mathbb{S}^2 := \left\{\boldsymbol{x} \in \mathbb{R}^3 : \|\boldsymbol{x}\|_2 = 1\right\}$ denote the unit sphere and let $(\vartheta, \varphi)^\top \in [0, \pi] \times [0, 2\pi)$ with $\boldsymbol{x} = (\sin\vartheta\cos\varphi, \sin\vartheta\sin\varphi, \cos\vartheta)^\top$ be its parameterization. For finite sampling sets $\mathcal{X} = \left\{\boldsymbol{x}_i \in \mathbb{S}^2 : i = 0, \ldots, M-1\right\}$, we denote by

$$\delta_{\mathcal{X}} := 2 \max_{\boldsymbol{y}\in\mathbb{S}^2} \min_{\boldsymbol{x}\in\mathcal{X}} \arccos(\boldsymbol{x} \cdot \boldsymbol{y}) \tag{2.1}$$

their mesh norm. A good starting point on well and best distributed sampling nodes on the sphere $\mathbb{S}^2$ can be found in [15].

For the quadrature formulae of consideration we underly the spaces of polynomials $\Pi_N$, $N \in \mathbb{N}_0$. Every $f_N \in \Pi_N$ has an orthonormal expansion (1.3) with respect to the spherical harmonics

$$Y_k^n(\vartheta, \varphi) := \sqrt{\frac{2n+1}{4\pi}} P_{|k|}^n(\cos\vartheta) \mathrm{e}^{\mathrm{i}k\varphi} \tag{2.2}$$

of degree $n = 0, \ldots, N$ and order $|k| \leq n$. For the readers convenience, the associated Legendre functions $P_k^n : [-1, 1] \to \mathbb{R}$ and the Legendre polynomials $P_n : [-1, 1] \to \mathbb{R}$ are given by

$$P_k^n(x) = \left(\frac{(n-k)!}{(n+k)!}\right)^{1/2} \left(1-x^2\right)^{k/2} \frac{\mathrm{d}^k}{\mathrm{d}x^k} P_n(x), \qquad n \in \mathbb{N}_0,\ k \le n,$$
$$P_n(x) := \frac{1}{2^n n!} \frac{\mathrm{d}^n}{\mathrm{d}x^n} \left(x^2-1\right)^n, \qquad n \in \mathbb{N}_0.$$

Then, to given points $\boldsymbol{x}_i \cong (\vartheta_i, \varphi_i)$, $i = 0, \ldots, M-1$, and a polynomial degree $N$ we ask for nonnegative quadrature weights $w_i \ge 0$ such that for all $f \in \Pi_N$ the equation

$$\int_{\mathbb{S}^2} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \sum_{i=0}^{M-1} w_i f(\boldsymbol{x}_i) \tag{2.3}$$

holds true. A sufficient condition on the existence of nonnegative quadrature weights follows from [6, 13, 9, 2].

**Lemma 2.1.** *For (sufficiently large) $M \in \mathbb{N}$ let $\mathcal{X}_M = \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{M-1}\} \subset \mathbb{S}^2$ be a set of sampling nodes with minimal mesh norm*

$$\delta_{\mathcal{X}_M} = \min_{\mathcal{X} \subset \mathbb{S}^2, |\mathcal{X}| = M} \delta_{\mathcal{X}}. \tag{2.4}$$

*Then, for polynomial degree*

$$N < \frac{\sqrt{M}}{1530} \tag{2.5}$$

*there exists nonnegative quadrature weights $\boldsymbol{w} = (w_0, \ldots, w_{M-1})$, $w_i \ge 0$, $i = 0, \ldots, M-1$, satisfying equation* (2.3)*.*

*Proof.* A minimal spherical covering $\mathcal{X}_M$ of cardinality $M$ has mesh norm

$$\left(\frac{32\pi}{3\sqrt{3}M}\right)^{\frac{1}{2}} - C_1 M^{-1} \le \delta_{\mathcal{X}_M} \le \left(\frac{32\pi}{3\sqrt{3}M}\right)^{\frac{1}{2}} + C_2 M^{-\frac{2}{3}} \tag{2.6}$$

for some constants $C_1, C_2 \ge 0$, cf. [6]. Hence, for sufficiently large $M$ we can bound the mesh norm by

$$\delta_{\mathcal{X}_M} \le \frac{5}{\sqrt{M}},$$

which yields in conjunction with (2.5) the inequality

$$2(153\delta_{\mathcal{X}_M} N) \le 1530 \frac{N}{\sqrt{M}} < 1. \tag{2.7}$$

Moreover, the proof of [9, Theorem 1] shows for $153N\delta_{\mathcal{X}_M} < 1$, some (Voronoi) weights $v_i > 0$, $i = 0, \ldots, M-1$, and $f \in \Pi_N$ the $\mathrm{L}^1$-Marcinkiewicz-Zygmund inequality

$$\left| \sum_{i=0}^{M-1} v_i |f(\boldsymbol{x}_i)| - \int_{\mathbb{S}^2} |f(\boldsymbol{x})| \mathrm{d}\boldsymbol{x} \right| \le 153N\delta_{\mathcal{X}_M}.$$

In conjunction with the arguments from the proof of [13, Theorem 4.1] this guarantees nonnegative quadrature weights under condition (2.7). ■

**Remark 2.2.** The condition $(N+1)^2 \le M$ is necessary for exact quadrature formulae (2.3) of degree $2N$, cf. [16, Lemma 2]. Since the bounds in (2.6) are asymptotically optimal, the pessimistic constant in (2.5) is due to the techniques employed in [13, 9, 2]. $\square$

Quadrature weights in more realistic settings are obtained as follows. Using the series expansion (1.3) with coefficients (1.1) and the relation $a_0^0 = \frac{1}{\sqrt{4\pi}} \int_{\mathbb{S}^2} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ we see that equation (2.3) is equivalent to the linear system of equations

$$\boldsymbol{Y}^* \boldsymbol{w} = \sqrt{4\pi}\boldsymbol{e}_0, \tag{2.8}$$

where $\boldsymbol{Y}$ is the nonequispaced spherical Fourier matrix

$$\boldsymbol{Y} := \left(Y_k^n(\boldsymbol{x}_i)\right)_{i=0,\ldots,M-1;\, n=0,\ldots,N,|k|\le n} \in \mathbb{C}^{M\times(N+1)^2},$$

$\boldsymbol{e}_0$ is the unit vector

$$\boldsymbol{e}_0 := (1,0,\ldots,0)^\top \in \mathbb{R}^{(N+1)^2}$$

and $\boldsymbol{w}$ is the weight vector

$$\boldsymbol{w} := (w_i)_{i=0,\ldots,M-1} \in \mathbb{R}^M.$$

Since we cannot say for which $N$ the system (2.8) is solvable, except for $N$ satisfying Lemma 2.1, we propose the following convex optimization problem

$$\min \left\| \boldsymbol{Y}^* \boldsymbol{w} - \sqrt{4\pi}\boldsymbol{e}_0 \right\|_2 \quad \text{subject to} \quad \boldsymbol{w} \ge \boldsymbol{0}. \tag{2.9}$$

# 3 Iterative solver

In order to use tools for real convex optimization we introduce the real convex optimization problem

$$\min \left\| \boldsymbol{A} \boldsymbol{w} - \sqrt{4\pi}\boldsymbol{e}_0 \right\|_2 \quad \text{subject to} \quad \boldsymbol{w} \ge \boldsymbol{0} \tag{3.1}$$

with

$$\begin{aligned}
\boldsymbol{A} &:= \begin{pmatrix} \boldsymbol{A}_1^\top \\ \boldsymbol{A}_2^\top \end{pmatrix} \in \mathbb{R}^{(N+1)^2\times M}, \\
\boldsymbol{A}_1 &:= \operatorname{Re}\left(Y_k^n(\boldsymbol{x}_i)\right)_{i=0,\ldots,M-1;\, n=0,\ldots,N,0\le k\le n} \in \mathbb{R}^{M\times\frac{(N+1)(N+2)}{2}}, \\
\boldsymbol{A}_2 &:= \operatorname{Im}\left(Y_k^n(\boldsymbol{x}_i)\right)_{i=0,\ldots,M-1;\, n=1,\ldots,N,-n\le k\le -1} \in \mathbb{R}^{M\times\frac{N(N+1)}{2}}.
\end{aligned}$$

This problem is equivalent to problem (2.9) due to the representation (2.2) of the spherical harmonics

$$Y_k^n(\vartheta,\varphi) = \sqrt{\frac{2n+1}{4\pi}} P_{|k|}^n(\cos\vartheta)(\cos(k\varphi) + \mathrm{i}\sin(k\varphi)).$$

Recently, fast approximate algorithms for the matrix times vector multiplication with the nonequispaced spherical Fourier matrix $\boldsymbol{Y}$ and its adjoint $\boldsymbol{Y}^*$ have been proposed in [11, 10]. For an implementation see [8]. Thus we immediately obtain a fast algorithm for the matrix times vector multiplication $\boldsymbol{v} := \boldsymbol{A}\boldsymbol{w} \in \mathbb{R}^{(N+1)^2}$ based on the adjoint spherical Fourier transform by computing $\tilde{\boldsymbol{v}} := \boldsymbol{Y}^*\boldsymbol{w}$ and setting

$$\boldsymbol{v} = (v_k^n)_{n=0,\dots,N,-n\le k\le n} \qquad \text{with} \qquad v_k^n := \begin{cases} \mathrm{Re}(\tilde{v}_k^n) & \text{for } k \ge 0, \\ \mathrm{Im}(\tilde{v}_k^n) & \text{for } k < 0. \end{cases}$$

Similar we compute the matrix times vector multiplication $\boldsymbol{w} := \boldsymbol{A}^\top \boldsymbol{v} \in \mathbb{R}^M$ again with the spherical Fourier transform

$$\boldsymbol{w} = \mathrm{Re}\,(\boldsymbol{Y}\tilde{\boldsymbol{v}}) \qquad \text{after setting} \qquad \tilde{v}_k^n := \begin{cases} v_k^n & \text{for } k \ge 0, \\ \mathrm{i}v_k^n & \text{for } k < 0. \end{cases}$$

In both cases the arithmetical complexity is $\mathcal{O}\left(N^2 \log^2 N + M\right)$.

In the following we present three methods used to solve problem (3.1). All of them make use of the fast matrix vector multiplication proposed above. The first and second are adapted standard algorithms for convex optimization whereas the third is a software package for bound-constrained least-square problems [4].

## 3.1 Conjugate gradient method

To minimize the norm $\left\|\boldsymbol{A}\boldsymbol{w} - \sqrt{4\pi}\boldsymbol{e}_0\right\|_2$ we consider the CGNR method. The drawback of the CGNR method is that it does not take the constraints $\boldsymbol{w} \ge \boldsymbol{0}$ into account, so that we cannot guarantee the nonnegativity of the solution. But we will see in the numerical tests in Section 4 that for well conditioned problems, i.e, the sampling sets are uniform distributed, we get a nonnegative solution $\boldsymbol{w}$ by the CGNR method for problem (3.1) with polynomial degrees close to the maximal degree of exactness. For ill conditioned problems we propose a reiterated CGNR method. Here, we start with the CGNR method on the whole $\mathbb{R}^M$ and restart the CGNR method after deleting the nodes with negative weights. We repeat this until all remaining weights $w_i$ are nonnegative or all nodes are deleted. Furthermore, the reiterated CGNR method terminates the inner iteration besides the trivial termination conditions, if for the current residual vector $\boldsymbol{r}_l := \boldsymbol{A}\boldsymbol{w}_l - \sqrt{4\pi}\boldsymbol{e}_0$ the inequality $\|\boldsymbol{A}^*\boldsymbol{r}_l\|_2/\|\boldsymbol{A}^*\boldsymbol{b}\|_2 < 0.001\|\boldsymbol{r}_l\|_2/\|\boldsymbol{b}\|_2$ holds true or if the residual $\|\boldsymbol{r}_l\|_2 \,/\, \left\|\sqrt{4\pi}\boldsymbol{e}_0\right\|_2$ after 50 steps does not get lower.

## 3.2 Infeasible interior point method

Interior point methods are commonly used to solve linear optimization problems [14, 17]. Here we consider an adapted infeasible interior point method (IIPM) based on a damped Newton iteration applied to the following nonlinear system of equations

$$\boldsymbol{A}\boldsymbol{w} - \sqrt{4\pi}\boldsymbol{e}_0 = \boldsymbol{0}, \quad \boldsymbol{0} < \boldsymbol{w} \in \mathbb{R}^M, \tag{3.2}$$

$$\boldsymbol{A}^\top \boldsymbol{y} + \nabla b(\boldsymbol{w}) = \boldsymbol{0} \tag{3.3}$$

with the barrier function $b(\boldsymbol{w}) := -\sum_{j=0}^{M-1} \ln(w_j)$ and dual variables $\boldsymbol{y} \in \mathbb{R}^{(N+1)^2}$. For $\mu = 1$ and $\boldsymbol{c} = \boldsymbol{0}$, these are the saddle-point conditions of the Lagrange function $L_\mu(\boldsymbol{w}, \boldsymbol{y}) := \boldsymbol{c}^\top\boldsymbol{w} + (\boldsymbol{A}\boldsymbol{w} - \boldsymbol{b})^\top\boldsymbol{y} + \mu b(\boldsymbol{w})$ of the minimization problem

$$\min(\boldsymbol{c}^\top\boldsymbol{w} + \mu b(\boldsymbol{w})) \;\text{ subject to }\; \boldsymbol{A}\boldsymbol{w} = \sqrt{4\pi}\boldsymbol{e}_0,\; \boldsymbol{0} < \boldsymbol{w},$$

which motivates the primal-dual interior point methods. In our special case we are only interested in nonnegative solutions $\boldsymbol{w}$ and do not care about the dual equations (3.3). The challenge to use interior point methods with the CG method is that the condition number of the linear system $\boldsymbol{A}\boldsymbol{D}\boldsymbol{A}^T\Delta\boldsymbol{y} = \boldsymbol{v}$ in Algorithm 1 increases fast near the boundary of the positive cone $\mathbb{R}_+^M$ where $\boldsymbol{D}$ denotes the inverse of the Hessian matrix

$$\boldsymbol{D} := \operatorname{Hess}\left(b(\boldsymbol{w})\right)^{-1} := \left(\frac{\partial^2 b(\boldsymbol{w})}{\partial w_i \partial w_j}\right)^{-1}_{i,j=0,\ldots,M-1} = \operatorname{diag}\left(w_i^2\right)_{i=0,\ldots,M-1}.$$

For the numerical tests we terminate the adapted IIPM, if in the outer iteration the current

**Algorithm 1** adapted IIPM

**Input:** polynomial degree $N \in \mathbb{N}$, number of nodes $M \in \mathbb{N}$
sampling nodes $\boldsymbol{x}_j \in \mathbb{S}^2$, $j = 1, \ldots, M$
accuracy $\varepsilon > 0$, limit of iterations $L_{\max} \in \mathbb{N}$, damping parameter $\varrho \in (0,1)$

Initialize $\boldsymbol{w}_0 := (\frac{\sqrt{4\pi}}{M}, \ldots, \frac{\sqrt{4\pi}}{M})^\top \in \mathbb{R}^M$, $\boldsymbol{y}_0 := (0, \ldots, 0)^\top \in \mathbb{R}^{(N+1)^2}$ and $l := 0$
**while** $\varepsilon < \frac{\left\|\boldsymbol{A}\boldsymbol{w}_l - \sqrt{4\pi}\boldsymbol{e}_0\right\|_2}{\left\|\sqrt{4\pi}\boldsymbol{e}_0\right\|_2}$ and $l < L_{\max}$ **do**
  $\boldsymbol{D}_l := \operatorname{diag}(w_0^2, \ldots, w_{M-1}^2)$
  $\nabla b(\boldsymbol{w}_l) := -(w_0^{-1}, \ldots, w_{M-1}^{-1})$
  Solve $\boldsymbol{A}\boldsymbol{D}_l\boldsymbol{A}^\top\Delta\boldsymbol{y}_l = \boldsymbol{A}\boldsymbol{w}_l - \sqrt{4\pi}\boldsymbol{e}_0 - \boldsymbol{A}\boldsymbol{D}_l(\boldsymbol{A}^\top\boldsymbol{y}_l + \nabla b(\boldsymbol{w}_l))$ for $\Delta\boldsymbol{y}_l$ with the CG method
  $\Delta\boldsymbol{w}_l = -\boldsymbol{D}_l(\boldsymbol{A}^\top\Delta\boldsymbol{y}_l + \nabla b(\boldsymbol{w}_l) + \boldsymbol{A}^\top\boldsymbol{y}_l)$
  Compute the maximal $\tau_{\max,l}$ such that $\boldsymbol{w}_l + \tau_{\max,l}\cdot\Delta\boldsymbol{w}_l > 0$
  $\tau_l := \min(1, \varrho\cdot\tau_{\max,l})$
  $\boldsymbol{w}_{l+1} := \boldsymbol{w}_l + \tau_l\cdot\Delta\boldsymbol{w}_l$
  $\boldsymbol{y}_{l+1} := \boldsymbol{y}_l + \tau_l\cdot\Delta\boldsymbol{y}_l$
  $l := l+1$
**end while**

**Output:** quadrature weights $\boldsymbol{w} > \boldsymbol{0}$

residual $\left\|\boldsymbol{A}\boldsymbol{w}_l - \sqrt{4\pi}\boldsymbol{e}_0\right\|_2 / \left\|\sqrt{4\pi}\boldsymbol{e}_0\right\|_2$ is greater than the previous residual or if it is lower than the accuracy of the solution obtained in the CG method. The CG method stops, if the residual after 50 steps does not get lower.

### 3.3 Bound-constrained least-squares formulation

For solving bound-constrained least-squares problems (BCLS) of the more general form

$$\min_{\boldsymbol{l}\le\boldsymbol{w}\le\boldsymbol{u}} \frac{1}{2}\left\|\boldsymbol{A}\boldsymbol{w} - \sqrt{4\pi}\boldsymbol{e}_0\right\|_2^2 + \frac{1}{2}\mu\left\|\boldsymbol{w}\right\|_2 + \boldsymbol{c}^\top\boldsymbol{w}$$

where the vectors $\boldsymbol{l}, \boldsymbol{u}$ are bounds for the solution $\boldsymbol{w}$ and $\mu \in [0,\infty)$ is a regularization parameter, we use the BCLS software package [4]. The used algorithm is based on iterative solvers, which makes it suitable for our considerations. To obtain the optimization problem (3.1) we set $\boldsymbol{l} = \boldsymbol{0} \in \mathbb{R}^M$, $\boldsymbol{u} = (\sqrt{4\pi}, \ldots, \sqrt{4\pi})^T \in \mathbb{R}^M$, $\mu = 0$ and $\boldsymbol{c} = \boldsymbol{0} \in \mathbb{R}^M$. Furthermore, we set the flags BCLS_PROJ_SEARCH_EXACT and BCLS_NEWTON_STEP_CGLS.

# 4 Numerical results

The methods were tested on an Intel Pentium 4 with 3.2 GHz and 1 GB memory and a standard 32 Bit Linux using the nfft3 library [8]. For the nonequispaced fast spherical Fourier transform (nfsft) we set the flag PRE_PSI, the threshold parameter $\kappa = 1000$ and the cutoff parameter $m = 9$. Furthermore, we set the requested accuracy to $\varepsilon =$1e-15. Because of this high precision the algorithms terminate in the most cases before this accuracy could be achieved.

We consider four different examples of sets of sampling nodes on the sphere, which are almost uniformly distributed, as well as a random set of nodes on the sphere and furthermore a special grid based on a tensor product, where the polynomial degree of exactness is already known. In the Tables 4.1 - 4.6 we illustrate the numerical results. We list the final residual $\left\|\boldsymbol{A}\boldsymbol{w} - \sqrt{4\pi}\boldsymbol{e}_0\right\|_2 / \left\|\sqrt{4\pi}\boldsymbol{e}_0\right\|_2$, the number of outer and the average number of inner iterations according to the used methods. Due to the lack of further termination conditions in the BCLS package we give a limit of maximal iterations, which depends on the examples, and mark this intervention by *.

We start with grids where the quadrature weights are known analytically.

**Example 4.1.** The Gauss-Legendre quadrature grid $\mathcal{X}_S^{\mathrm{G}}$ of size $S \in \mathbb{N}_0$ is the Cartesian product

$$\mathcal{X}_S^{\mathrm{G}} := \left\{\vartheta_j^{\mathrm{G}} \;:\; j = 0, \ldots, S\right\} \times \left\{\frac{k\pi}{S+1} \;:\; k = 0, \ldots, 2S+1\right\}$$

with equispaced latitudinal nodes. For the longitudinal direction we use the Gauss-Legendre quadrature with nodes $\vartheta_j^{\mathrm{G}}$. The corresponding weights $w_j^{\mathrm{G}}$ can be obtained as the unique solution of an eigenvalue problem, see [1, pp. 95]. The weights for the entire quadrature formula are then given by

$$W_S^G := \left\{w_{j,k}^{\mathrm{G}} := \frac{2\pi}{2S+2} w_j^{\mathrm{G}} \;:\; j = 0, \ldots, S;\; k = 0, \ldots, 2S+1\right\}.$$

The number of nodes is $M = 2S^2 + 4S + 2$ and the quadrature formula is exact for polynomials with degree $N \leq 2S + 1$. We choose $S = 48$, i.e., $M = 4802$ and compute the weights for the polynomial degree $N = 2S + 1 = 97$. It is remarkable that we obtain the Gauss-Legendre weights for $N = 97$, even though the system of equations in (3.1) is overdetermined. Furthermore we test our algorithms for the case $N = 98$ where all algorithms fail in finding nonnegative quadrature weights, cf. Table 4.1. □

**Example 4.2.** A set of $M$ sampling nodes is called a spherical $t$-design if the integral of any polynomial of degree at most $t$ over the sphere $\mathbb{S}^2$ is equal to the average value of the polynomial over the set of $M$ nodes. For the tests we choose some $t$-designs with minimal cardinality from [7]. We remark that in these cases the system of equations (3.1) is overdetermined, too. Table 4.2 shows the results obtained using the CGNR method, all other methods work as well. □

| algorithm | $N$ | residual | iterations outer | iterations inner | time |
|---|---|---|---|---|---|
| reiterated CGNR | 97 | 2.134744e-14 | 1 | 38.0 | 29s |
| adapted IIPM | 97 | 2.148676e-14 | 3 | 78.3 | 163s |
| BCLS | 97 | 1.647701e-07 | 7* | 5.0 | 26s |
| reiterated CGNR | 98 | 7.906984e-01 | 2 | 10.5 | 15s |
| adapted IIPM | 98 | 9.199394e-01 | 1 | 51.0 | 34s |
| BCLS | 98 | 7.906984e-01 | 4* | 25.0 | 44s |

Table 4.1: Results for the Gauss-Legendre grid of size $M = 4802$, i.e. $S = 48$, and requested polynomial degree of exactness $N$.

| set of sampling nodes | $M$ | $N$ | residual | iterations | time |
|---|---|---|---|---|---|
| vertices of the regular octahedron | 6 | 3 | 2.944461e-16 | 1 | 0s |
| vertices of the regular icosahedron | 12 | 5 | 1.749046e-15 | 2 | 0s |
| 13-design | 94 | 13 | 1.506909e-11 | 10 | 0s |
| 17-design | 156 | 17 | 6.542798e-07 | 10 | 0s |
| 21-design | 240 | 21 | 6.524231e-06 | 9 | 0s |

Table 4.2: Results for spherical $t$-design of size $M$ and requested polynomial degree of exactness $N$.

**Example 4.3.** The HEALPix grid $\mathcal{X}_S^{\mathrm{H}}$ is an equal area partitioning scheme on the sphere and has importance in several applications, e.g. for cosmic microwave background data analysis [5]. It comprises $M = 12S^2$ nodes and is given explicitly by

$$\mathcal{N}_S := \left\{ \left( \arccos\left(1 - \frac{k^2}{3S^2}\right), \frac{\pi\left(n + \frac{1}{2}\right)}{2k} \right) \,:\, k = 1, \ldots, S-1;\, n = 0, \ldots, 4k-1 \right\},$$

$$\mathcal{E}_S := \left\{ \left( \arccos\left(\frac{2(2S-k)}{3S}\right), \frac{\pi\left(n + \frac{\delta_{0,(k+S) \bmod 2}}{2}\right)}{2S} \right) \,:\, k = S, \ldots, 3S;\, n = 0, \ldots, 4S-1 \right\},$$

$$\mathcal{S}_S := \left\{ \left( \arccos\left(-\left(1 - \frac{k^2}{3S^2}\right)\right), \frac{\pi\left(n + \frac{1}{2}\right)}{2k} \right) \,:\, k = 1, \ldots, S-1;\, n = 0, \ldots, 4k-1 \right\},$$

$$\mathcal{X}_S^{\mathrm{H}} := \mathcal{N}_S \cup \mathcal{E}_S \cup \mathcal{S}_S.$$

A drawback is that HEALPix grids lack an exact integration scheme with easily computable weights. Test results for the proposed algorithms applied to the HEALPix grids are shown in Table 4.3. It is remarkable that the error achieved by the BCLS package keeps under 1e-8 for the HEALPix grid of size $S = 20$ and the polynomial degrees $N = 61, 62, 63$. The error deteriorate from 1e-14 for degree $N = 61$ to 1e-5 for degree $N = 62, 63$ using the other methods.

By means of our implementation we are able to compute quadrature weights for HEALPix grids up to degree of exactness $N = 1024$ and higher, since these algorithms do not store the whole Fourier matrix $\boldsymbol{Y}$ into memory. Some test results are shown in Table 4.4. These tests were run on a Intel Xeon with 3GHz and 16GB and a 64-bit Linux.

Furthermore Figure 4.1 shows, that by using $M \approx \frac{4}{3}(N+1)^2$ nodes, we are able to integrate polynomials up to degree $N$ with relative error around 1e-12, where we used the CGNR method and set the accuracy to $\varepsilon$ =1e-12. $\square$

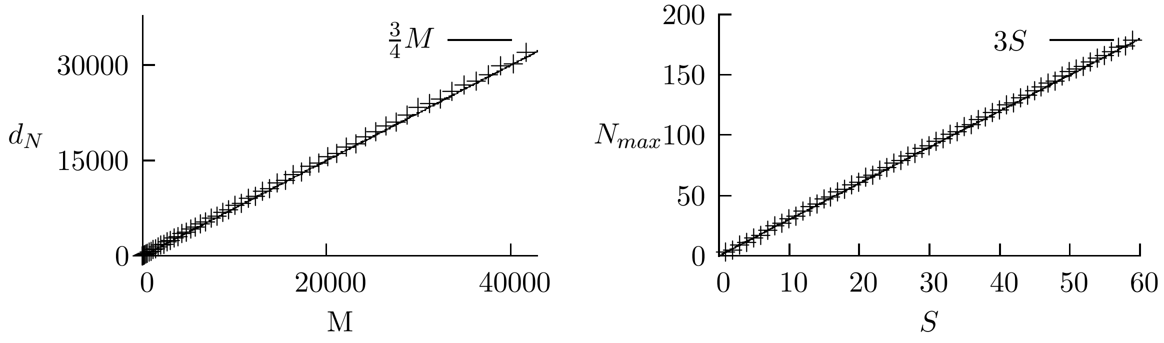


Figure 4.1: It seems there is a linear relation between the dimension $d_N = (N+1)^2$ of the polynomial space $\Pi_N$ and the maximal degree of exactness $N$ with accuracy $\left\|\boldsymbol{A}\boldsymbol{w} - \sqrt{4\pi}\boldsymbol{e}_0\right\|_2 / \left\|\sqrt{4\pi}\boldsymbol{e}_0\right\|_2 <$ 1e-12 for HEALPix grids of size $M = 12S^2$.

| algorithm | $N$ | residual | iterations outer | iterations inner | time |
|---|---|---|---|---|---|
| reiterated CGNR | 61 | 5.515829e-15 | 1 | 48.0 | 21s |
| adapted IIPM | 61 | 4.020338e-15 | 3 | 134.3 | 200s |
| BCLS | 61 | 3.192499e-10 | 5* | 11.8 | 23s |
| reiterated CGNR | 62 | 1.290948e-06 | 1 | 119.0 | 49s |
| adapted IIPM | 62 | 2.168341e-06 | 1 | 88.0 | 33s |
| BCLS | 62 | 1.618963e-10 | 5* | 99.2 | 154s |
| reiterated CGNR | 63 | 1.290948e-06 | 1 | 125.0 | 50s |
| adapted IIPM | 63 | 2.168341e-06 | 1 | 88.0 | 33s |
| BCLS | 63 | 1.165060e-09 | 5* | 99.8 | 137s |

Table 4.3: Results for the HEALPix grid of size $M = 4800$, i.e., $S = 20$, and requested polynomial degree of exactness $N$.

**Example 4.4.** The Reuter grid $\mathcal{X}_S^{\mathrm{E}}$ is a so-called equidistribution grid of size $S \in \mathbb{N}$ for which in the limit $S \to \infty$ the average value of a continuous function over the $S$ sampling nodes approaches the integral of this function over the sphere $\mathbb{S}^2$, see [3, Chapter 7]. We took the

| $M$ | $N$ | residual | total iterations | time |
|---|---|---|---|---|
| 367500 | 512 | 3.242945e-13 | 2883 | 31923s |
| 410700 | 512 | 1.918440e-14 | 173 | 2089s |
| 1687500 | 1024 | 1.088695e-14 | 93 | 5825s |

Table 4.4: Results for HEALpix grids of size $M$ and requested polynomial degree of exactness $N$ using the reiterated CGNR method.

ensemble [3, Example 7.1.9] with nodes given by

$$\mathcal{X}_S^{\mathrm{E}} := \{x_{0,0} = (0,0), x_{S,0} = (\pi, 0)\} \cup \bigcup_{j=1}^{S-1} \left\{ x_{j,k} = \left( \frac{j\pi}{S}, \left(k - \frac{1}{2}\right) \left(\frac{2\pi}{S_j}\right) \right) k = 1, \ldots, S_j \right\},$$

$$S_j := \begin{cases} 1 & \text{, if } j = 0 \text{ or } j = S, \\ \left\lfloor 2\pi / \arccos \left( \left( \cos \frac{\pi}{S} - \cos^2 \frac{j\pi}{S} \right) / \sin^2 \frac{j\pi}{S} \right) \right\rfloor & \text{, if } 0 < j < S. \end{cases}$$

An upper bound for the number of nodes is $M \leq 2 + \frac{4}{\pi} S^2$. Table 4.5 shows us similar results for the Reuter grid as for the HEALPix grids. It seems that for size $S = 61$ the maximal degree of exactness is 60. Additionally, we observe the same behavior for the spiral nodes given in [15]. We have to choose also $M \approx \frac{4}{3}(N+1)^2$ nodes in oder to have degree $N$ of numerical exactness. □

| algorithm | $N$ | residual | iterations outer | iterations inner | time |
|---|---|---|---|---|---|
| reiterated CGNR | 60 | 2.277367e-15 | 1 | 32.0 | 13s |
| adapted IIPM | 60 | 1.302229e-14 | 3 | 80.0 | 85s |
| BCLS | 60 | 3.860017e-09 | 5* | 4.4 | 10s |
| reiterated CGNR | 61 | 3.247092e-06 | 1 | 37.0 | 18s |
| adapted IIPM | 61 | 9.272831e-06 | 1 | 66.0 | 27s |
| BCLS | 61 | 3.247492e-06 | 5* | 10.6 | 19s |
| reiterated CGNR | 62 | 1.562653e-05 | 1 | 78.0 | 34s |
| adapted IIPM | 62 | 5.206605e-05 | 1 | 114.0 | 58s |
| BCLS | 62 | 1.562656e-05 | 5* | 19.6 | 31s |

Table 4.5: Results for the Reuter grid of size $M = 4768$, i.e., $S = 61$, and requested polynomial degree of exactness $N$.

**Example 4.5.** Finally we consider a uniform random distribution over the sphere. We see in Table 4.6 that it is a challenge to compute nonnegative quadrature weights for random nodes with iterative solvers for sufficient high polynomial degree and accuracy. □

| algorithm | $N$ | residual | iterations | | time |
|---|---|---|---|---|---|
| | | | outer | inner | |
| reiterated CGNR | 41 | 7.727673e-15 | 7 | 275.3 | 417s |
| adapted IIPM | 41 | 1.654468e-12 | 7 | 496.6 | 1134s |
| BCLS | 41 | 1.848869e-02 | 445* | 44.9 | 5210s |
| reiterated CGNR | 42 | 1.133570e-14 | 8 | 494.0 | 823s |
| adapted IIPM | 42 | 7.014779e-04 | 9 | 713.6 | 1769s |
| reiterated CGNR | 43 | 8.915856e-02 | 9 | 778.1 | 1758s |
| adapted IIPM | 43 | 1.604574e-01 | 7 | 467.7 | 903s |

Table 4.6: Results for a random distribution with $M = 4800$ nodes and requested polynomial degree of exactness $N$.

# 5 Conclusions

In this paper we compared three algorithms for computing nonnegative quadrature weights at several distributions of sampling nodes. It turns out that for well distributed sampling nodes the CGNR method suffices to compute nonnegative quadrature weights and is practical in a reiterated variant also for random distributions. Furthermore, combined with the nonequispaced fast spherical Fourier transform [8] we are able to compute quadrature weights for suitable sets of sampling nodes up to polynomial degree $N = 1024$.

# References

[1] P. J. Davis and P. Rabinowitz. *Methods of Numerical Integration (Second Edition)*. Academic Press Inc., 1984.

[2] F. Filbir and W. Themistoclakis. Polynomial approximation on the sphere using scattered data. *Math. Nachr.*, 281:650 – 668, 2008.

[3] W. Freeden, T. Gervens, and M. Schreiner. *Constructive Approximation on the Sphere*. Oxford University Press, Oxford, 1998.

[4] M. P. Friedlander. BCLS 0.1, C software package, 2007.

[5] K. M. Górski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, and M. Bartelmann. Healpix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere. *The Astrophysical Journal*, 622:759–771, 2005.

[6] W. Habicht and B. L. van der Waerden. Lagerung von Punkten auf der Kugel. *Math. Ann.*, 123:223 – 234, 1951.

[7] R. H. Hardin and N. J. A. Sloane. Spherical Designs , 2002.

[8] J. Keiner, S. Kunis, and D. Potts. NFFT 3.0, C subroutine library, 2006.

[9] J. Keiner, S. Kunis, and D. Potts. Efficient reconstruction of functions on the sphere from scattered data. *J. Fourier Anal. Appl.*, 13:435 – 458, 2007.

[10] J. Keiner and D. Potts. Fast evaluation of quadrature formulae on the sphere. *Math. Comput.*, 77:397 – 419, 2008.

[11] S. Kunis and D. Potts. Fast spherical Fourier algorithms. *J. Comput. Appl. Math.*, 161:75 – 98, 2003.

[12] Q. T. Le Gia and H. N. Mhaskar. Quadrature formulas and localized linear polynomial operators on the sphere. *Preprint*, 2007.

[13] H. N. Mhaskar, F. J. Narcowich, and J. D. Ward. Spherical Marcinkiewicz-Zygmund inequalities and positive quadrature. *Math. Comput.*, 70:1113 – 1130, 2001. Corrigendum on the positivity of the quadrature weights in 71:453 – 454, 2002.

[14] J. Nocedal and S. J. Wright. *Numerical optimization.* Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.

[15] E. B. Saff and A. B. J. Kuijlaars. Distributing many points on a sphere. *Math. Intelligencer*, 19:5 – 11, 1997.

[16] I. H. Sloan. Polynomial interpolation and hyperinterpolation over general regions. *J. Approx. Theory*, 83:238 – 254, 1995.

[17] R. J. Vanderbei. *Linear programming.* International Series in Operations Research & Management Science, 114. Springer, New York, third edition, 2008.